AI in Healthcare. Hype or Help?

1 Breaking the (A)Ice

1.1 Welcome to module 1

Welcome to the first module of this MOOC, where we will break the (A)Ice. In this opening module, we lay the groundwork for a holistic understanding of AI in healthcare. We will embark on a journey that takes us through the past, present, and future of AI's role in healthcare, shedding light on both its potential benefits and the challenges it presents.

Key Focus Areas

- **Added value of AI in healthcare**: We begin by unravelling the value that AI can bring to healthcare. AI can be used in applications ranging from improving patient diagnosis to enabling groundbreaking medical discoveries. As such, we will explore how AI can impact different domains in the healthcare landscape.
- **Definition of AI**: Before delving deeper, we need to clarify: what exactly is Artificial Intelligence? We will break down AI into its fundamental components, ensuring a solid foundation for further exploration.
- **Timeline of AI**: To appreciate the context in which AI operates in healthcare, we will embark on a historical journey through the evolution of AI. We will trace its origins, key milestones, and pivotal moments that have shaped its current prominence.
- **Digital transformation in healthcare**: The integration of AI is a cornerstone of the ongoing digital transformation in healthcare. We will examine how the digital transformation of healthcare systems could contribute to achieving greater efficiency and improved patient care.
- **The good, the bad and the ugly of AI**: AI in healthcare is a double-edged sword. While it offers tremendous promise, it also raises ethical and practical concerns. We will navigate this complex terrain, discussing both the positive impacts and putting the finger on some potential hurdles for deployment of AI.

Why This Module Matters

Understanding the topics covered in this module is essential for anyone interested in the intersection of AI and healthcare. AI has the potential to revolutionize the way we deliver and receive healthcare services, but its implementation must be approached with a nuanced perspective. By the end of this module, you will understand the added value of AI in healthcare, comprehend what AI truly entails, grasp its historical context, and discern the many facets of its impact.

Learning goals

- Identify the different impact domains of AI in healthcare.
- Recognize the importance of learning the fundamentals of clinical machine learning for all stakeholders in the healthcare ecosystem.
- Gain knowledge about the origins of machine learning in healthcare.
- Understand the context and principles of common terms and definitions in machine learning.
- Define wording in the fields of machine learning, data science, and artificial intelligence.
- Start recognizing the challenges and limitations to machine learning approach in healthcare.

- Understand the first principles for designing machine learning applications for healthcare.

1.2 Added value of AI in healthcare

1.2.1 Scope of healthcare

In this first subsection of Module 1, we will discuss the **added value of AI in healthcare**. We will explore how AI can impact different domains in the healthcare landscape. But before we can do that, let's understand the scope of healthcare first.

The **scope of healthcare** is broad and encompasses a wide range of activities, services, and areas related to the maintenance and improvement of an individual's health.

Starting with the aspect of "**care**" in healthcare, let's look at several types of care. Instead of simply providing you with the definitions, try to actively recognize the meaning behind each type of care in the exercise below.

Drag each word next to its corresponding definition.

- Preventive Care: Vaccinations, regular check-ups, screenings, and lifestyle modifications aimed at avoiding illness and promoting overall health.
- Primary Care: Family doctors, internists, paediatricians are often the first point of contact for patients. They provide comprehensive healthcare services, manage chronic conditions, and refer patients to specialists when necessary.
- Specialty Care: Medical specialists, such as cardiologists, oncologists, and neurologists, focus on specific areas of medicine and provide expert care for complex or specific health conditions.
- Emergency Care: Hospitals and emergency rooms provide critical care for acute and life-threatening conditions, injuries and emergencies.
- Surgical Care: Surgeons perform various types of surgeries to treat injuries, diseases, and conditions, ranging from routine procedures to complex surgeries.

Now focusing on the other aspect of "**health**" in healthcare, can you also connect the definitions of health in the exercise below?

Drag each word next to its corresponding definition.

- Public Health: Health initiatives and organisations work to prevent the spread of diseases, promote health education, and improve the overall health of communities and populations.
- Mental Health and Behavioural Health: Services for mental conditions, substance abuse treatment, and therapy to address psychological and emotional well-being.
- Dental and Oral Health: Dentists and oral health professionals provide care for the teeth, gums and oral cavity.
- Global Health: Addressing health issues and providing healthcare services on a global scale, including efforts to combat infectious diseases, improve healthcare infrastructure in underserved regions, and promote health equity worldwide.

Different healthcare activities can also be defined based on the location of the patient requiring healthcare. For example, ambulance and emergency room services, in-hospital stays, care homes or nursing homes, at-home care for chronic illnesses, and more. Additionally, many other activities are closely linked to healthcare, such as pharmaceuticals and life sciences, rehabilitation and physical therapy, health insurance and administration, telemedicine, and more. The scope of healthcare is simply overwhelming and we ourselves are very interested in what your activity is in healthcare.

1.2.2 Benefits of AI in healthcare

Artificial intelligence is rapidly transforming the healthcare industry by levering its capability in analysing large amounts of medical data to identify patterns and trends that may not be visible to human experts. Therefore, AI proves invaluable in tailoring healthcare to individual patients, factoring in genetics, lifestyle, and environmental factors. In addition, it streamlines various tasks, from automating data entry and image analysis to appointment scheduling, record management, and referral processes. Here are some specific examples of **how AI is being used in healthcare today**:

- In **radiology**, AI is being used to develop tools that can automatically detect cancer and other diseases in medical images.
- In **drug discovery**, AI is being used to identify new targets for drugs and to design new drugs more quickly and efficiently.
- AI is being used to develop tools that can select the best **treatment** based on the patient.
- In primary care, AI is being used to develop **chatbots** that can provide first point of contact or answer frequently asked questions.
- In **surgery**, AI is being used to develop robots that can perform surgery more precisely and safely than human surgeons.
- AI can assist in the early detection of **breast cancer** through the analysis of mammogram images.
- AI can be utilized to optimize various aspects of the **supply chain** in hospitals and healthcare organizations, e.g., demand forecasting, inventory management, route optimization, and many more.

It is clear from these limited examples already that the impact of AI in healthcare is both substantial and diverse. Therefore, current, and future professionals and scholars, such as yourself, will face the challenge of adopting AI correctly in healthcare. This MOOC aims to empower you as a healthcare professional or scholar with valuable insights and critical knowledge about the era of AI in healthcare.

We will start with laying out some AI foundations, and all the aspects covered in this module will later be connected to various concrete use cases.

We do not aim to make an exact categorization, but we can broadly identify **seven impact domains of AI in healthcare**, which we will briefly illustrate in the next section.

1.2.3 Impact domains of AI in healthcare

- **Medical diagnosis** is the process by which healthcare professionals identify and determine the nature of a patient's health condition or disease based on symptoms, medical history, physical examinations, and often, diagnostic tests or medical imaging.
- **Medical prognosis** refers to the prediction or forecast of the expected course and outcome of a patient's health condition or disease. It provides information about the likely progression, recovery, or outcome of the condition.
- **Medical treatment** refers to the use of medical interventions, procedures, medications, or therapies by healthcare professionals to address and manage health conditions, injuries, or diseases in patients with the goal of improving their health and well-being.
- **Medical therapy** typically involves a systematic and structured approach to treat physical or mental health issues. It often includes counselling, rehabilitation, or other interventions to help individuals recover from injuries, manage chronic conditions, or address psychological and emotional well-being.

- **Medical screening** involves the systematic examination of individuals who may not show symptoms of a specific disease or condition but are at risk or within a certain age range for early detection. Its purpose is to identify potential health issues in their early stages, enabling timely intervention and treatment. Screening tests are typically applied to a broad population to reduce the impact of diseases by detecting them before symptoms manifest.
- **Medical prevention** refers to a set of proactive measures and interventions aimed at avoiding the occurrence or progression of diseases and health conditions. It encompasses strategies such as vaccinations, lifestyle modifications, early detection, and health education to reduce the risk of illness and promote overall health and well-being.
- **Personalized medicine**, also known as precision medicine, is an approach to medical treatment and healthcare that tailors medical decisions, interventions, and therapies to individual patients based on their unique genetic, genomic, clinical, and environmental characteristics. The goal is to provide more effective and personalized healthcare by considering each patient's specific factors and needs, ultimately improving treatment outcomes and reducing adverse effects.
- **Evidence-based medicine** is an approach to medical practice that emphasizes the use of the best available scientific evidence, alongside clinical expertise and patient values and preferences, to make informed decisions about patient care. It involves systematically reviewing and applying research findings and clinical studies.
- A **medical organization** is a structured entity, such as a hospital, clinic, healthcare network, or medical association, dedicated to providing healthcare services, conducting medical research, or supporting healthcare professionals in the delivery of care.
- **Medical logistics** refers to the planning, coordination, and management of the procurement, storage, transportation, and distribution of medical supplies, equipment, and resources within healthcare systems and organizations.
- **Medical assistance** refers to the professional help, care, or support provided by healthcare practitioners, such as doctors, nurses, or paramedics, to individuals who require medical attention or treatment for their health conditions, injuries, or other healthcare needs.
- **Medical interaction** refers to the communication and engagement between healthcare providers and patients during the provision of healthcare services.
- **Medical research** refers to the systematic investigation and study of health-related topics, diseases, treatments, and medical advancements with the aim of improving our understanding of the human body, diagnosing and treating diseases, and enhancing overall healthcare.
- **Medical education** is the process of acquiring knowledge, skills, and competencies related to the practice of medicine. It encompasses both theoretical learning and practical clinical training to ensure that medical professionals are well-prepared to provide quality healthcare services.

1.3 What is AI?

1.3.1 What is AI?

Introducing this new subsection in the module, we will answer the important question of granny Vivian:

Aisha recently enrolled in the MOOC "AI in Healthcare. Hype or Help?", just like you. Excited as she is, Aisha tells granny Vivian all about it. Vivian has a clear understanding of the "healthcare" part of the course, but she cannot grasp the true meaning behind "AI".

"**What is AI**?" Vivian asks sceptically.

"AI stands for Artificial Intelligence" Aisha replies.

"Okay, but what does Artificial Intelligence actually mean?" Vivian retorts. Good question... Can someone explain this to her?

We learned that **AI gives computers the ability to learn** without being explicitly programmed. AI is definitely not magic. AI learns to unravel patterns in data.

In this MOOC, we will discuss extensively how AI "learns", i.e. how the algorithm is optimized for a certain task. The way it learns can take various forms but is very different from how a child learns. At the end of this course, it should be very clear that the core to AI relates to optimizing a predefined algorithm by **feeding enough data of good quality to the algorithm**. By learning from data, AI allows to excel at a particular pre-defined task.

It is critically important to **begin with an informed question**. This question may be derived from the medical literature or from personal experience, but one has to define a unique question, i.e. which defines the output of your future model and envisions the related actions. Especially in medicine, machine learning is best understood as a means to an end that has consequences.

Let's continue to the next page for several important definitions concerning AI.

1.3.2 Some definitions: AI and machine learning

The modern terms of machine learning and artificial intelligence were coined in the 1950s and '60s, in regard to the theory that machines could be made to simulate learning or any other feature of intelligence. Both terms are often used interchangeably.

The term "machine learning" is often used by scientists or data-science practitioners.

The term "AI" is often used for marketing purposes or for communicating to the public.

Let's make the distinction clear between these terms:

Artificial intelligence (AI) refers broadly to the development of computer systems that can perform tasks that typically require human intelligence. These tasks include problem-solving, learning, understanding natural language, speech recognition and visual perception, among others.
Machine learning is a subset of AI. It is a family of statistical and mathematical modelling techniques that uses a variety of approaches to automatically learn and improve the prediction of a target state, without explicit programming. Machine learning relies heavily on pattern recognition and the theory that computers can learn useful relationships in data towards an output, without being explicitly programmed. "Learning" in machine learning is the reference of the desire to create a model that can achieve an objective through experience (or exposure to data) with little to no external (human) assistance.
Data science is a multidisciplinary field that involves extracting insights and knowledge from structured and unstructured data. It combines techniques from statistics, mathematics, computer science, and domain expertise to analyse and interpret complex data sets.

From simple sets of rules to machine learning

In the earliest days, automated algorithms tried to mimic human intelligence by implementing a set of **rules**. Such rules were rarely derived from data, but rather predefined by experts. For example (click on the box below):

Example COPD diagnosis (1)

Identifying chronic obstructive pulmonary disease (COPD) is still often based on the ratio of 2 spirometer values: forced vital capacity (FVC) / first second of forced expiration (FEV1) > 0.7. This is a clear rule, leading to a simple diagnostic approach. It is hard to claim that this algorithm is "intelligent". Furthermore, this rule ignores age-dependency of lung parameters and interaction with other potentially relevant variables.

What is COPD? COPD is a progressive respiratory condition characterized by persistent airflow limitation due to chronic bronchitis and/or emphysema. It is commonly caused by long-term exposure to irritating gases or particulate matter, such as seen in cigarette smoke. COPD leads to difficulty in breathing, coughing, and increased susceptibility to respiratory infections.

Heuristic rule-based approach

As such, we can define a heuristic rule-based approach in AI when:

- The computer programmer knows what the input and output looks like and assumes a relation between input and output.
- The computer **programmer implements a function** that processes the input and produces an output based on this predefined knowledge.
- The decision rules and decision thresholds are based on a manual effort to **explicitly define the rules**. This approach can be seen as implementing the existing human intelligence.

However, rule-based systems face challenges in handling complex, dynamic, or large-scale problems. The manual creation and maintenance of extensive rule sets is impractical as problems grow in complexity. This is where machine learning comes in.

Machine learning

Formally, machine learning:

- Starts from a set of **predefined features**, which are the known inputs, and aims to map this to a predefined output.
- The function that maps inputs to outputs is assumed to be too complex to code manually.
- In machine learning, the **computer learns the function** that maps inputs to outputs.
- Instead of relying on a computer programmer to come up with the rules of the function, we instead optimize (a limited set of) **parameters** in a generic function to learn the optimal input-output relationship based on available data. This is called "training" the statistical model on available **"training" data**.
- The machine learning algorithms use mathematical formulations to represent models and strive to learn parameters in these formulations, by tracking them back from the training data.

Example COPD diagnosis (2)

If we would extend the COPD diagnostic algorithm into a machine learning based approach, we would aim to collect a large set of parameters (e.g. FVC, FEV1, age, sex, symptoms, …) together with an expert diagnosis (patient has COPD or patient does not have COPD). In such setting, we still know the inputs and the outputs, but we aim to design optimal decision rules and decision thresholds from the data that is available.

You can already appreciate that the machine learning approach might detect more complex relations between inputs (than a simple ratio of two parameters) in order to make the diagnosis. Machine learning might outperform heuristic learning when the data used for deriving rules is representative for

the problem at hand. (Disclaimer: assessing representativeness is non-trivial and will be further investigated in the future section on cross-validation and testing.)

Deep learning

Then, what is deep learning?

- It is a subset of machine learning but goes one step further and aims to map inputs to outputs **without defining features** explicitly.
- It often starts from raw data, with little or no preprocessing.
- As such, it is highly dimensional and has a high number of parameters to optimize.
- And as such, it also requires a much larger amount of training data as it both learns the features and the relationship between input and output at the same time.
- Deep learning largely drops the explicit mathematical formulations and rather relies on **overparameterization**.

Example COPD diagnosis (3)

So, using deep learning for COPD classification could start from a very large set of spirometry curves (which are time series rather than discrete spirometry features) and ground truth labels (COPD or no COPD), and learn patterns from this data.

Success in heuristic learning, machine learning and deep learning can be defined as creating an accurate and reproducible model for the given task. As data is used for optimizing parameters in machine learning and deep learning, a result can only be considered sufficiently good when it is validated on an independent dataset and has sufficiently good performance on such an independent dataset.

1.4 Back to the future – Timeline of AI

1.4.1 Brief history of AI

Aisha had been talking excitedly to Vivian about her new course "AI in Healthcare. Hype or Help?".

"Aisha, dear, I'm happy that you finally explained to me what AI is, because it seems like everyone's talking about it, and I can't keep up with all these new inventions." Vivian says with a sigh.

Aisha chuckles. "Oh, Grandma, AI actually isn't as new as you think. The concept goes back decades." Vivian raises an eyebrow. "Decades? Really? I thought this was some modern wizardry."

Aisha: "No, it's been around for quite a while. The term 'artificial intelligence' was coined in the 1950s. And now, it's making a big impact, also in healthcare."

In this subsection, we will embark on a historical journey through the evolution of AI, in order to fully appreciate the context in which AI operates in healthcare today.

The emergence of AI cannot be retracted to a single moment in history. As is often the case, the data science field has experienced multiple **AI summers**, periods of increasing "belief", and **AI winters** of "disbelief", during which scepticism took over.

The concept of using computers to simulate intelligent behaviour and critical thinking, was first described by Alan Turing in 1950. In his paper "Computing Machinery and Intelligence", Turing described a simple test, which later became known as the **"Turing test"**, to determine whether computers were capable of human intelligence. This test involves a human judge interacting with both a machine and a human without knowing which is which. If the judge cannot distinguish between the

two based on their responses, the machine is said to have passed the Turing test, demonstrating a level of artificial intelligence comparable to human intelligence in conversation.

But even long before Turing, all sorts of **"automations"** had been developed that would appear to be intelligent: e.g., automatic yes-or-no nodding statues; the mysterious Antikythera mechanism, what may have been the first analogue computer, was created in Greece; a first real instance of a chess computer would appear in 1912, with an automaton named El Ajedrecista; and many more.

The current "summer" of new AI developments is deeply rooted in the exponential progress of science and technology across various disciplines. The past centuries have created the enabling conditions that allow AI to rapidly evolve into a new systems technology today.

Some examples of AI highlights (The History of Artificial Intelligence: Complete AI Timeline (techtarget.com)):

- 1950 – Turing Test: Alan Turing published "Computing Machinery and Intelligence," introducing the Turing test and opening the doors to what would be known as AI.
- 1952 – Samuel Checkers-Playing Program: Arthur Samuel developed Samuel Checkers-Playing Program, the world's first program to play games that was self-learning.
- 1966 – ELIZA: Joseph Weizenbaum created Eliza, one of the more celebrated computer programs of all time, capable of engaging in conversations with humans and making them believe the software had humanlike emotions.
- 1985 – Bayesian networks casual analysis: Judea Pearl introduced Bayesian networks causal analysis, which provides statistical techniques for representing uncertainty in computers.
- 1997 – Computer defeats world chess champion: IBM's Deep Blue defeated Garry Kasparov in a historic chess rematch, the first defeat of a reigning world chess champion by a computer under tournament conditions.
- 2011 – Siri (by Apple): Apple released Siri, a voice-powered personal assistant that can generate responses and take actions in response to voice requests.
- 2014 – DeepFace facial recognition: Facebook developed the deep learning facial recognition system DeepFace, which identifies human faces in digital images with near-human accuracy.
- 2022 – ChatGPT: OpenAI released ChatGPT in November to provide a chat-based interface to its GPT-3.5 LLM.

Finally, **why is AI impacting healthcare now**?

The current summer in AI in general is driven by more computational power, better algorithms and more available data. This last aspect is why AI is also transforming healthcare: there are large datasets being collected from patients, and nowadays they are stored in a digital format, as you will see on the next page.

1.5 The latest healthcare disruption. Take it or leave (it)

1.5.1 Digital transformation in healthcare

The scope of healthcare is continually evolving due to advancements in medical science and technology. One major driver of change in healthcare is the **digital transformation**, which refers to the process of integrating digital technologies and strategies (e.g., cloud databases and computing, data analytics, sensing and measuring devices, automation, and more) into various aspects of healthcare.

This shift is changing the way healthcare is delivered and is driven by a number of factors: the increasing availability of digital technologies, the growing demand for patient-centred care, and the need to improve efficiency and reduce costs.

The digital transformation of healthcare is still in its early stages, but by making healthcare more efficient, affordable, and accessible, its digital transformation can help to improve healthcare overall. Some of the **key areas** where the digital transformation is already having an impact on healthcare, include:

- **Electronic health records (EHRs)**: EHRs are digital systems that store patient medical records. They can help to improve communication among healthcare providers, reduce medical errors, and improve patient care.
- **Digital imaging**: Digital imaging such as X-rays, CT scans, and MRIs, are used to create images of the body's internal structures. These images can be used to diagnose diseases, plan treatment, and monitor patient progress. With digital imaging, compared to their analogue counterpart, improvements in image quality, acquisition time, reduced radiation exposure and lower costs can be obtained.
- **Telemedicine**: Telemedicine is the use of telecommunications technology to provide healthcare services remotely. It can be used to provide care to patients in rural areas or to those who are unable to travel to a healthcare facility.
- **Wearable devices**: Wearable devices are devices that can be worn on the body to track health data such as heart rate, blood pressure and sleep patterns. This data can be used to monitor patient health and to provide personalized care.
- **Virtual reality (VR)**: VR is used to train healthcare professionals, to provide therapy for patients with mental health conditions, and to create immersive experiences that can help patients better understand their condition.
- **Artificial intelligence (AI)**: Last but certainly not the least, AI is used to develop new healthcare tools, by harvesting all the digital data with the aim to improve the efficiency of healthcare operations, and to personalize patient care. With input from EHRs, digital images, and data from wearable devices, AI has become a critical tool in extracting value from the ongoing digital transformation in healthcare.

So, AI is a huge part of the digital transformation, but we must note that AI in healthcare is a double-edged sword. While it offers tremendous promise, it also raises ethical and practical concerns. We will navigate this complex terrain in the next subsection.

1.6 The good, the bad and the ugly of AI

1.6.1 What's the "good"?

In this new subsection, we will uncover **the good, the bad, and the ugly of AI**.

First of all,... What's the "**good**"? It is clear how AI could bring healthcare advantages; that is why you are here. Many use cases have demonstrated and will further demonstrate:

- **That accuracy and efficiency in diagnosis can be improved**: AI can help healthcare providers to more accurately and efficiently diagnose diseases and conditions by analysing heterogenous and large quantities of patient data, such as medical images, lab results, and electronic health records. As an example, AI algorithms have been developed to diagnose skin cancer with accuracy equivalent to that of dermatologists, and to detect breast cancer in mammography with accuracy superior to that of human radiologists. Similarly, it has been shown that

pulmonologists using AI for lung disease diagnosis outperform pulmonologists not relying on AI support.

- **Personalizing treatment**: AI can objectively personalize treatment plans for individual patients based on their unique medical history and genetic profile. This leads to more effective treatments, less side effects and better outcomes.
- **Predicting and preventing disease**: AI can be used to analyse large datasets of patient information to screen routine data for early markers of risk and predict the likelihood of disease outbreaks and pandemic evolution (think for example of COVID-19).
- **Streamlining administrative tasks**: AI will further streamline administrative tasks such as appointment scheduling, billing, and medical coding. For example, AI-powered virtual assistants can schedule appointments for patients by accessing electronic medical records (EMRs) and checking availability in real time. This way, reducing wait times and enabling more effective scheduling saves time for healthcare workers, and enhances the patient's experience.
- **Supporting drug discovery and development**: AI can be used to analyse large datasets of genetic and molecular information to identify new drug targets, and to predict the effectiveness of new drugs.

1.6.2 What's the "bad"?

Now you know in which way AI could be "good" in the healthcare domain. But what could be "**bad**" about AI in healthcare?

- **Reliance on technology**: While AI has the potential to improve the speed and accuracy of healthcare decision-making, it can also lead to a lack of critical thinking and reduction of diagnostic skills among healthcare providers.
- **Potential for errors**: Like any technology, AI systems are not perfect and can make errors. If these errors go undetected or if they are not corrected in a timely manner, they could lead to serious consequences for patients.
- **Trust & explainability**: We will need a lot of time to trust an autonomous car, to see how it reacts in situations we are familiar with, and also in situations of emergency. Consequently, it will take even more time not only for patients, but also for medical professionals, to trust AI with making medical diagnoses, supporting medical decision-making or designing new drugs. We will devote a section in this course to explainability, as a way to increase trust in the decisions of AI.
- **Risk for bias**: AI feeds on data. Implicitly, one would assume data is of high quality and representative for the whole population, but blindly assuming so, could end badly. If data is not of high quality or not fully representative, systematic errors might creep in, which might go unnoticed as AI is a black box tool. In practice, only when AI has access to high-quality data, it can excel at tasks. Achieving high-quality data requires time-consuming and monotonous work by medical professionals who act as data annotators. The dedicated contribution of data annotators is of crucial importance for the benefit of implementing AI in the healthcare setting.

Let's get into a real example depicting bias. The goal of an AI application was to distinguish between patients with pneumonia and patients without pneumonia, based on X-ray images. The training data consisted of data from one hospital (A) with almost no pneumonia, and a hospital (B) with a high prevalence of pneumonia. As both hospitals used different X-ray machines, the easiest solution for the AI was to make a decision based on the machine type, rather than on the presence of lung abnormalities, which defeated the purpose.

1.6.3 What's the "ugly"?

Finally, what's the "**ugly**" when it comes to AI in healthcare? Is artificial intelligence ethical? What are the moral questions to resolve?

Ethical concerns can relate to data privacy, discrimination, informed consent, and who is responsible for the AI decision.

- **Bias might lead to discrimination**: AI algorithms are only as good as the data they are trained on, and if that data is biased or incomplete, the AI system can produce discriminatory or unfair results. For example, if an AI system is trained on data from one race only, results might be completely unreliable when applied to data from a different race. Such racial biases could lead to unacceptable discriminatory healthcare decisions. In an extreme case, the informed AI developer might exploit data bias to induce discriminatory or manipulative results in certain situations.
- **Privacy and security concerns**: AI systems require large amounts of data to function effectively, but where is the data coming from? Using data for training AI, raises concerns about patient privacy and the security of personal health information. To protect patient privacy and stop data breaches, it is essential that patient data is managed safely and properly. But there is always a risk that this data could be used for unauthorized purposes, or that it would fall into the wrong hands.
- **Safety and effectiveness**: It is important to ensure that AI systems are safe and effective. This requires rigorous testing and evaluation from a technical point of view, as well as ongoing monitoring to ensure that the AI system is performing as intended. This goes without saying for any technology. But who is responsible and liable for the AI decisions?
- **Legal issues and liability**: What if a deep learning algorithm misses a diagnosis: the doctor accepts the judgment, and the patient suffers from the consequences? What if an autonomous surgical robot injures a patient during a procedure? It is an ongoing debate about who will be held liable in the future when robots and AI, acting autonomously, harm patients. Current consensus states that the professional is open to liability if he or she used the tool in a situation outside the scope of its regulatory approval, or misused it, or applied it despite significant professional doubts of the validity of the evidence surrounding the tool, or with knowledge of the toolmaker obfuscating negative facts. In any other cases, liability falls back on the creators and the companies behind them.

Lastly, there is often the **concern that AI might completely replace human decision-making**, which might result in a loss of the personal touch in patient care. It is crucial to emphasize that in our opinion, AI should not be used to replace healthcare personnel, but rather as a tool to support and improve the quality of care.

2 First AId kit

2.1 Welcome to Module 2

Welcome in Module 2, where we transition from the broad overview presented in Module 1 to a more technically oriented perspective. We embark on a journey taking us from the fundamental mathematics and algorithms that power AI to the practical aspects of software development. We will explore the intricacies of knowledge representation, AI agents, and the important role of probability and uncertainty in healthcare AI. Additionally, we will delve into the realm of data-driven AI, focusing on data exploration and visualization techniques that enable informed decision-making in healthcare.

Key Focus Areas

- **From algebra to algorithms**: Let's start with a bridge between mathematical fundamentals and AI algorithms. Understanding the mathematical foundations of AI is crucial for grasping the logic behind AI models and their applications in healthcare.
- **Engineering cycle of software development**: Successful AI implementation often follows the engineering cycle of software development. We will explore this iterative process, emphasizing its importance in healthcare AI projects.
- **Knowledge representation and reasoning**: Knowledge is at the heart of AI systems. Let's delve into knowledge representation techniques and how AI systems reason and make informed decisions based on this knowledge.
- **AI agents**: AI agents are the actors that interact with and make decisions in the healthcare environment. We will examine the architecture and functions of these agents in healthcare scenarios.
- **Probability and uncertainty in healthcare**: Healthcare is inherently uncertain, and understanding how AI deals with probability and uncertainty is vital. We will explore probabilistic models and their applications.
- **Data-driven AI**: Data is the lifeblood of AI in healthcare. We will discuss the concept of data-driven AI and why it is pivotal in making healthcare decisions more evidence-based and precise.
- Data visualization and exploration: Effective data exploration and visualization are key to extracting insights from healthcare data. We will delve into techniques and tools that enable healthcare professionals to harness the power of data.

Why This Module Matters

Module 2 serves as the foundation for the technical aspects of AI in healthcare. Understanding the mathematical principles, software development processes, and knowledge representation techniques is essential for creating AI solutions that are both reliable and effective in healthcare settings. Moreover, grasping the role of data, probability, and uncertainty, along with data exploration and visualization, empowers healthcare professionals to harness AI's potential for improved diagnosis, treatment, and patient care.

Learning goals

- Identify the seven stages of algorithmic and software development.
- Explain the integration of AI agents into different environments and link these to different levels of AI.
- Outline an AI knowledge cycle and illustrate by example the types of knowledge used in AI.

- Explain the task of planning and scheduling and identify planning tasks in given or self-proposed applications.
- Understand deterministic versus probabilistic AI-agents and environments and identify sources of uncertainty in given or self-proposed applications.
- Explain the difference between data-driven AI systems and models with expert systems and physics-based models.

2.2 Unravelling the AI mystery: from algebra to algorithms

2.2.1 The beginning: algorithms and software

Dr. Zarah returned home from a long day at her clinic, feeling intrigued and curious. A team of researchers, who are developing a groundbreaking tool to aid her in diagnosing complex medical cases, had approached her to collaborate. Yet, the language they used, with terms like "algorithms" and "software," felt foreign and disconnected from her daily practice.

During dinner, she had to ask the “expert” Aisha. -

"Why do we need all this talk about algorithms?" Zarah wondered aloud, waiting for Aisha to answer.

Aisha - "Well, let's start with the basics and the definition of the word algorithm. The word "**algorithm**" has its origins in the name of a Persian mathematician. The Latin translation of his name, "Algoritmi," was used to refer to him and his work. Over time, this Latin term evolved into the word "algorithm" in English.”

Zarah nodded, now intrigued by the historical connection. "So, an algorithm is related to mathematics?"

Aisha smiled, "Indeed, algorithms are related to mathematics and problem-solving. An algorithm is like a set of instructions or a recipe that a computer can follow to perform a specific task. In your medical world, it's comparable to a clinical protocol you might follow for diagnosing a patient's condition. It's a structured way to solve a problem."

Dr. Zarah nodded, slowly starting to see the connection. "So, are you saying that AI is like a digital doctor that follows a set of mathematical instructions to help me with diagnoses?"

Aisha smiled, "Exactly! AI is the technology that enables machines to mimic human intelligence. It's like having a virtual medical colleague that can process vast amounts of data, analyse them, and provide insights to assist in medical decision-making. Now, imagine building this digital colleague. It's a bit like developing a new medical tool, and it involves a process called **software development**. When you group multiple algorithms together and package them into a computer program, you get what we call **software**. Think of it as a toolbox filled with different tools (algorithms) that can be used to accomplish various tasks. This software can range from simple applications like word processors to complex systems like AI-powered medical diagnostic tools. **AI** is the realm where algorithms and software merge to create systems that can think and learn like humans. These AI systems use algorithms to process data, learn from it, and make decisions or predictions based on what they've learned.”

2.3 From idea to implementation

2.3.1 From idea to implementation

In this new subsection of the module, we will explore the **stages of software engineering** within the context of healthcare, tracing the path **from concept to implementation**, where lines of code become the building blocks of transformative medical solutions.

Simultaneously, we will delve into the **world of AI**, unravelling the iterative **cycle of knowledge** acquisition, refinement and application, as it empowers the healthcare sector with intelligent tools that learn, adapt and enhance patient care.

The future of technology and innovation is shaped by the interconnectedness of traditional software development and AI-driven knowledge cycles. This synergy can be harnessed for transformative advancements in healthcare.

2.3.2 The story of adopting Hachiko

As an introduction to the seven stages of software engineering, we will use the story of Hachiko's adoption in the family as an illustrative example.

Everyone in the family agrees that Hachiko is the most fun member to play with. Nowadays, nobody can imagine their lives being without this cute robot dog anymore; especially Noah, who had been dreaming of getting a dog for many years beforehand. It was actually not an easy journey to reach the point of adopting Hachiko.

Dreaming of getting a dog

Before the family welcomed Hachiko to their lives a few years ago, Noah constantly complained about getting a dog, but Eric and granny Vivian were against it. Dogs are nice, but none of the family members had the time to properly take care of a pet. Furthermore, baby Jack suffers from pet allergies. Consequently, it did not look like they could welcome a dog into the house.

But one day, Aisha found a leaflet of a company that builds robot pets. She offered the idea to the family as an alternative to getting a real dog. Of course Noah was ecstatic with the idea, Zarah did not see a problem with it, and even Eric and Vivian agreed that this was a nice compromise. So the family decided to go forward with this idea.

Prototype Rex

Despite not adopting a real dog, getting a robot pet was still a big project that required thorough decision-making. According to the leaflet of the company, they offered personalized versions of different animals, meaning that they could create the animal based on the detailed instructions of the owner. So the family had to decide on what exactly they wanted in their new pet friend.

After many discussions, the family agreed on getting an **AI-powered robot dog that can mimic the behaviour of a real dog, respond to commands, and learn from its environment**. His name would be **Rex**.

As soon as the family went to order the robot dog on the website, they realized that more decisions had to be made than anticipated, as they were required to select distinct functions. So the family defined their dog requirements more precisely:

- Must have **voice recognition** for commands.

- Should have **sensors** for environmental awareness.
- Should be able to perform basic dog-like **movements** and responses.
- Must have a **learning algorithm for adapting** to new commands and behaviours.

The company also offered the possibility of sending a **prototype/beta version** of the dog first, which could then be progressively optimized based on the needs of the family. The family placed their order. After a couple of days, Rex arrived. Everyone in the family was happy to welcome Rex.

But after a few hours, the family found out that **Rex had many problems**. The robot was designed to hear and understand commands in German, with the manual also being in German and not in English. Moreover, they soon realized that Rex was quite clumsy... He was often bumping into the couch or the walls, which pointed out that the face recognition was not working properly.

Therefore, the family had to send Rex back to the company, with the mention of all the issues that they noticed and any additional functionalities that they wanted. The company promised to develop a new and improved robot based on prototype Rex.

The road to Hachiko

This time, the company took more time to implement all the requested changes. During the development process, they regularly send update videos to the family, showing the different tests that the new dog was going through:

- Test the **voice recognition** system with various commands in **English**.
- Evaluate **sensor** accuracy in **different environments**.
- Assess the robot dog's ability to **learn and adapt to new commands**.

This process made the family feel like they were watching the training of a real dog.

After almost a month, the new robot dog arrived in the house. Noah gave him the name **Hachiko**, after he watched the movie about the legendary Japanese dog.

The last task for the family was to read Hachiko's manual in detail to learn how they could teach Hachiko new tricks. The manual also included the information on when and how Hachiko should be sent to his "vet" for repair.

Hachiko's adoption story illustrates the process of software development:

- **Planning**: Deciding on buying a robot dog.
- **Requirements**: Setting the initial features of the robot dog.
- **Design and prototyping**: Waiting for Rex and arrival of Rex
- **Software development**: Waiting period to implement the requested changes from Rex to Hachiko.
- **Testing**: Videos of Hachiko performing tasks during training.
- **Deployment**: Arrival of Hachiko.
- **Operations and maintenance**: Manual for new tricks and vet.

2.3.3 Seven stages of software engineering

Similar to Hachiko's adoption story, **software development** is a structured process of creating, designing, testing, and maintaining software applications or programs. It involves a series of well-defined steps and methodologies to turn an idea or set of requirements into a functional software product that meets specific needs.

Just as a skilled healthcare professional meticulously diagnoses, plans, and administers care, the software development **stages guide the creation of applications** designed to streamline healthcare processes, enhance patient care, and drive medical research. The different stages are vital in software development to ensure a structured and systematic approach that aligns technology with the complex and critical needs of the (healthcare) industry. They provide a **roadmap** for designing, building, and maintaining (healthcare) software solutions that are accurate, reliable and tailored to the specific requirements of healthcare providers and patients.

- **Stage 1 – Planning**: In the planning stage, you lay the foundation for your AI project. We define its purpose and what medical problems it will help solve. It is like devising a blueprint for a new hospital wing. You define the project's scope, objectives, and requirements. For instance, if you want to create an AI system to predict disease outbreaks, you would plan the data sources, target diseases and desired outcomes.
- **Stage 2 – Requirements**: Requirements gathering is like the process of understanding the needs of the patients. In healthcare AI, this means defining what the AI system should do. For example, if you're building a diagnostic tool, the requirements would include what types of medical data it needs to analyse and what diseases it should detect.
- **Stage 3 – Design and Prototyping**: Prototyping is like conducting clinical trials or providing a prototype diagnosis. You create a smaller, experimental version of your AI system to test its capabilities. It's like testing a new medication on a small group of patients before widespread use. We design and build a basic version of the AI tool to see if it can work and how it would fit into your workflow.
- **Stage 4 – Software Development**: In the development stage you build the actual AI system. It's like constructing a state-of-the-art surgical suite in a hospital. In this step you implement the models by writing the code, designing the algorithms, and creating the AI models that will analyse medical data. It's like teaching the models medical expertise.
- **Stage 5 – Training and Testing**: Training and testing are like medical residency and board exams for the designed AI system. During training, the AI model learns from vast datasets, much like a medical resident learns from real patient cases. Testing ensures that the diagnoses and predictions provided by the AI model are accurate and reliable, just like passing board exams proves a doctor's competence.
- **Stage 6 – Deployment**: Deployment is like the moment when you introduce a new treatment or medical device into the hospital. We release the AI tool to be used in real medical scenarios. You integrate the AI tool with electronic health records and workflows, so healthcare providers can use it seamlessly for diagnosing diseases and planning treatments.
- **Stage 7 – Operation and Maintenance**: Lastly comes the operation of the AI tool in real-life during which maintenance is essential. Just as hospitals require continuous maintenance and upgrades or just like the regular follow-up with patients, AI systems need regular updates to stay current with medical knowledge, data, and technology. We update it, fix any issues, and ensure it keeps improving.

In conclusion, we laid the groundwork by unveiling the seven stages that govern the orchestration of software development – a methodology that has proven its worth across countless industries, including healthcare. Yet, as technology's relentless march propels us into a future teeming with possibilities, it becomes evident that our narrative is far from complete. Next up, let's explore the **AI knowledge cycle**, a pivotal concept that transcends the boundaries of traditional software development.

2.3.4 AI knowledge cycle

The AI knowledge cycle, an instrumental guide in harnessing the untapped potential of AI, promises not just to augment our capabilities, but to revolutionize the very essence of healthcare and its delivery.

- In essence, the **seven stages of software development** serve as the foundation;
- while the **AI knowledge cycle** provides the **specialized processes and capabilities** required to leverage AI in healthcare effectively.

By combining these frameworks, healthcare organizations can develop AI-powered solutions that enhance patient care, diagnosis, treatment planning, and medical research while adhering to industry best practices for software development.

The AI knowledge cycle consists of the following parts:

- **Perception – Data gathering**: Perception is like a doctor's patient data collection. AI gathers data from various healthcare sources—patient records, medical images, research papers—much like you collect patient histories, lab results, and imaging scans.
- **Learning – Data analysis and pattern recognition**: Learning is where AI becomes a medical detective. AI algorithms analyse this data, looking for patterns and insights. They learn from this data. It is similar to the process followed by doctors to analyse patient symptoms and test results to make diagnoses.
- **Knowledge representation – Structuring insights**: Knowledge representation is like organizing medical knowledge into a patient's chart to make diagnosis easier. AI structures what is learned into a format they can work with, creating a clear, organized understanding of medical concepts and relationships. Imagine organizing your patient records to make diagnosis easier.
- **Reasoning – Drawing medical conclusions**: Reasoning is where AI makes medical decisions. AI systems use their organized knowledge to make decisions and predictions. In healthcare, this could mean diagnosing diseases, predicting patient outcomes, or suggesting treatment plans. It is a process similar to a doctor's clinical reasoning. AI uses structured knowledge to draw conclusions, make diagnoses, and recommend treatments.
- **Planning – Developing strategies**: Planning is like creating a treatment plan. AI formulates strategies based on what is learned and reasoned, much like doctors create treatment plans for patients.
- **Education – Continuous learning**: And finally, education is an ongoing process. AI continues to learn, adapt, and update its knowledge, much like doctors continuously educate themselves with the latest medical research. This iterative process ensures that AI systems stay current with the latest medical advancements.

So, the seven stages of software development build the AI system, while the AI knowledge cycle powers it with continuous learning and improvement. By understanding and leveraging both, you can harness the full potential of AI in healthcare, improving patient care and advancing medical knowledge.

- In the early stages of AI development, understanding the **seven stages of software engineering** is crucial to plan, design, and build the AI system effectively.
- As the AI system becomes operational, the **AI knowledge cycle** becomes increasingly important to ensure it can adapt, learn, and make informed decisions based on evolving data and knowledge.

Ultimately, a successful AI project will need to integrate both approaches, as they complement each other to create AI systems that are not only well-structured and robust, but also capable of continuous improvement and adaptation.

2.4 Knowledge representation and reasoning

2.4.1 Knowledge representation and reasoning

In the realm of healthcare, the utilization of AI holds immense promise for enhancing patient care, diagnosis, treatment planning, and medical research. At the core of AI's potential in healthcare lies the ability to effectively **represent and reason with medical knowledge**.

In this new subsection of the module, we will delve into the crucial concepts of knowledge representation and reasoning within the context of healthcare, while also examining the various types of medical knowledge involved. In this exploration, we will focus on the foundational stages of knowledge representation and reasoning, which are crucial for AI's role in healthcare.

2.4.2 Types of medical knowledge

Medical knowledge in AI is multifaceted, encompassing various types of information that are indispensable for healthcare applications. Some different types of information are:

- **Declarative knowledge (Object facts)**: Declarative medical knowledge represents factual information about diseases, treatments and patient data. It comprises statements that describe medical facts and relationships among medical objects. For example: "Hypertension is a condition characterized by high blood pressure." Declarative medical knowledge forms the foundation for medical reasoning and decision-making, allowing AI systems to understand and work with concrete medical facts.
- **Heuristic knowledge (Rule of thumb)**: Heuristic medical knowledge consists of practical guidelines and rules of thumb for medical decision-making. These rules are often derived from clinical expertise and serve as shortcuts for diagnosing and treating patients. For example: "In cases of severe chest pain, consider it a potential symptom of a heart attack and act promptly." Heuristic medical knowledge empowers AI systems to make informed clinical decisions, especially in situations where complete patient data may not be available.
- **Structural knowledge (Relationship between medical concepts)**: Structural medical knowledge illustrates the relationships and connections between medical concepts, diseases, treatments and patient characteristics. It provides context and meaning to medical data by showcasing how various medical pieces of information relate to one another. For example: "Diabetes is a chronic metabolic disorder characterized by elevated blood sugar levels." Structural medical knowledge allows AI systems to navigate complex medical scenarios by recognizing patterns and dependencies in the data.
- **Procedural knowledge (Medical procedures and protocols)**: Procedural medical knowledge outlines the step-by-step procedures and protocols for conducting medical tests, surgeries and treatments. It guides AI systems in executing medical tasks effectively and safely. For example: "To diagnose diabetes, perform a fasting blood glucose test and an oral glucose tolerance test." Procedural medical knowledge equips AI systems with the ability to perform medical tasks and interventions accurately.
- **Meta knowledge (Knowledge about medical knowledge)**: Meta knowledge, or metacognition, refers to an AI system's awareness and understanding of its own medical knowledge and decision-making processes. It enables self-assessment, reflection and adaptive strategies

based on the context and the quality of medical information available. For example: "I am highly confident in my ability to diagnose common cold symptoms, but I need additional input from a specialist for complex cases." Meta knowledge enhances the self-awareness and adaptability of AI systems, allowing them to make more informed decisions in medical contexts.

Knowledge representation and reasoning are the cornerstones of AI-driven healthcare transformation. By structuring and reasoning with various types of medical knowledge, AI systems can assist healthcare professionals in diagnosing diseases, developing treatment plans, and delivering personalized care. This not only improves patient outcomes, but also contributes to the advancement of medical research and healthcare practices. In the ever-evolving field of healthcare, knowledge representation and reasoning remain pivotal in harnessing the full potential of AI for the benefit of patients worldwide.

2.4.3 Connection of knowledge to intelligence

In the realm of healthcare, knowledge about the real world is a critical component of both human intelligence and the development of artificial intelligence. **Knowledge serves as a cornerstone for demonstrating intelligent behaviour in AI systems** designed for healthcare. Just as a healthcare professional relies on their medical knowledge and expertise to make informed decisions, AI agents in healthcare must possess relevant knowledge to act effectively.

Consider a scenario in which an AI application encounters a complex medical case with intricate symptoms and diagnostic possibilities. Much like a person faced with a language they do not understand, the AI agent's ability to respond appropriately depends on its "knowledge" of medical concepts, procedures and prior cases.

In essence, the relationship between knowledge and intelligence in healthcare AI is similar to a healthcare provider's ability to diagnose and treat patients. **Without a foundation of medical knowledge, both human and AI would struggle to make meaningful decisions** in the healthcare domain. Therefore, knowledge forms the bedrock upon which the intelligent behaviour of healthcare AI agents is built, enabling them to analyse data, assist in diagnoses, recommend treatments, and ultimately contribute to better patient outcomes.

2.4.4 Common methods for reasoning

Reasoning with knowledge in healthcare involves using available information, data, and medical expertise to make informed decisions and draw meaningful conclusions. Several common **methods for reasoning with knowledge in healthcare** include:

- **Rule-based reasoning** involves using a set of if-then rules to make decisions or draw conclusions. These rules are typically created by domain experts and are used to model specific medical knowledge. In healthcare, rule-based reasoning can be used for diagnostic decision support. For instance, if a patient presents with a fever, cough, and chest pain, a rule-based system may suggest that they could have pneumonia and recommend further tests or treatment options.
- **Probabilistic reasoning** involves modelling uncertainty and probability in healthcare decisions. Bayesian networks, for example, are used to represent and update probabilities based on new evidence. More information about reasoning with uncertainty will be provided in one of the following subsections. In cancer diagnosis, probabilistic reasoning can help assess the likelihood of a patient progressing to a metastatic state based on symptoms, family history, and test results.

- **Semantic reasoning** involves understanding the meaning and relationships between medical concepts and terms. It uses ontologies and knowledge graphs to represent and reason about medical knowledge. Semantic reasoning can be used to link various medical concepts, such as connecting symptoms to diseases, medications to side effects, or genes to diseases, allowing for more comprehensive knowledge representation and decision support.
- **Temporal reasoning** considers the timing and sequence of events. It is crucial for tasks like monitoring disease progression, tracking medication adherence, and predicting patient outcomes over time. In diabetes management, temporal reasoning can help healthcare providers understand how a patient's blood glucose levels change over weeks or months, allowing for personalized treatment adjustments.
- **Causal reasoning** explores cause-and-effect relationships in healthcare data. It aims to understand how various factors contribute to health outcomes. In epidemiology, causal reasoning is used to identify the factors contributing to the spread of diseases, such as investigating how certain behaviours or environmental conditions lead to disease transmission.
- **Deductive reasoning** involves drawing specific conclusions from general principles or premises. It is a top-down approach where logical rules or established facts are applied to reach a specific conclusion. In a deductive approach, if it is known that a patient has a specific genetic mutation that is associated with a rare disease, and the patient exhibits symptoms consistent with that disease, deductive reasoning would conclude that the patient likely has that rare disease.
- **Inductive reasoning** involves generalizing based on specific observations. It is a bottom-up approach where patterns are identified in data, and general principles or hypotheses are formulated from these observations. In an inductive approach, by analysing a large dataset of patient records, one may observe that a particular medication is consistently effective in reducing blood pressure. From this data, the general hypothesis can be induced that this medication is a reliable treatment for hypertension.
- **Abductive reasoning** is the process of generating the best possible explanation for a given set of observations or evidence. It involves inferring the most likely cause or explanation from incomplete information. In abductive reasoning, when a patient presents with multiple symptoms, the clinician may generate hypotheses about the underlying cause by considering various factors and their likelihood. It assists in forming a differential diagnosis by proposing the most plausible explanations.
- **Case-based reasoning** involves solving new problems by recalling and adapting solutions from previous, similar cases stored in a knowledge base. It relies on the similarity between the current problem and past cases. In case-based reasoning, when faced with a complex diagnosis, a clinician or AI system retrieves and adapts solutions from past cases with similar patient profiles, symptoms, treatments, and outcomes to suggest a course of action or treatment plan.

These reasoning methods in healthcare encompass various approaches to problem-solving and decision-making. The **choice of method** depends on the nature of the problem, the available data and evidence, and the level of certainty required in the decision-making process.

These methods for reasoning with knowledge in healthcare are often used in combination to address complex medical challenges, improve patient care, and support healthcare professionals in making more accurate and evidence-based decisions. They leverage the wealth of data and medical expertise available to provide valuable insights and recommendations for diagnosis, treatment, and disease prevention.

Now that we learned about knowledge representation and reasoning, we will move on to the actors that interact with and make decisions in the healthcare environment.

2.5 AI agents

2.5.1 What is an AI agent?

In this new subsection of the module, we will learn about AI agents.

An **AI agent** is a software or hardware system that operates autonomously or semi-autonomously to perform tasks or make decisions based on data, algorithms, and predefined rules. AI agents are designed to mimic human-like intelligence, and they can range from relatively simple rule-based systems to highly complex machine learning models.

AI agents can take **various forms**, including chatbots, virtual assistants, recommendation systems, autonomous vehicles, industrial robots, AI pet robots (like Hachiko) and more. They are integrated into a wide range of applications across industries, from healthcare and finance to entertainment and transportation, to automate tasks, provide decision support, and enhance user experiences. The specific capabilities and characteristics of an AI agent can vary widely depending on its intended purpose and design.

Here are some key characteristics that define an AI agent:

- **Autonomy**: AI agents can operate without continuous human intervention. They are capable of making decisions and taking actions based on their programming or learned behaviours. Many AI agents can perceive and interpret their environment. This can include processing sensory data like images, text, or sensor readings to understand their surroundings.
- **Reasoning**: AI agents often have reasoning capabilities that allow them to analyse data, draw conclusions, and make logical decisions. This can involve symbolic reasoning, probabilistic reasoning, or machine learning-based reasoning.
- **Learning**: AI agents are capable of learning from data or experiences. Machine learning and deep learning techniques enable AI agents to improve their performance over time through training on datasets or by interacting with their environment.
- **Interaction**: AI agents can interact with users or other systems through various interfaces. This interaction can be in the form of natural language understanding and generation, speech recognition, or other communication methods.
- **Action**: AI agents are capable of taking actions or providing recommendations based on their analysis and decision-making processes. These actions can be physical (e.g., controlling a robot) or virtual (e.g., providing recommendations in an app).
- **Adaptability**: AI agents can adapt to changing conditions and environments. They can adjust their behaviour or decision-making processes to achieve their objectives in different situations.
- Domain-specific or general-purpose: AI agents can be specialized for specific tasks or domains (e.g., medical diagnosis, game playing, language translation) or designed to be more general-purpose and versatile (e.g., virtual assistants).
- **Feedback loop**: Many AI agents operate in a feedback loop, where they receive feedback from their actions or decisions and use this feedback to improve their performance in subsequent tasks.

With these characteristics, AI agents are able to play a pivotal role in healthcare by augmenting medical professionals' abilities, assisting in diagnosis, treatment planning, and patient monitoring.

2.5.2 AI agents and their environment

In AI, the interaction between an AI agent and its environment is a fundamental concept. An agent is any entity capable of perceiving its environment through sensors and acting upon it through actuators.

The **environment** represents the external world or system that the AI agent interacts with. It is the context in which the agent operates, and it can vary widely in terms of complexity, dynamics, and characteristics. The AI agent's goal is to select actions based on its perception of the environment to maximize some notion of cumulative reward or achieve specific objectives.

An **environment** refers to the external surroundings or context in which an AI agent operates. In healthcare, the environment may consist of **patient data, medical sensors, hospital equipment and medical records**. An AI agent operating in this environment may make decisions related to patient care, diagnosis, or treatment. Different types of environments are the following:

Static environment: An environment that does not change while the agent is deliberating. Static environments remain relatively stable and do not undergo frequent or significant changes. For example, a chessboard remains static during a player's turn. In a pathology laboratory, the environment is relatively static, because the fundamental processes and procedures for analysing specimens remain consistent over time. This stability and standardization are essential for maintaining the accuracy and reliability of diagnostic results.

Dynamic environment: An environment that can change even if the agent doesn't take any action. For example, a traffic system with moving vehicles is dynamic. In the healthcare domain, an emergency room is a dynamic environment where patients arrive with various medical conditions, and their conditions can change rapidly.

Deterministic environment: An environment where the next state is completely determined by the current state and the agent's actions. Chess is an example, as each move leads to a predictable outcome. In the healthcare domain, a pharmacy dispensing system that follows strict protocols for medication distribution is a deterministic environment, or the controlled administration of anaesthesia during surgery can be considered deterministic when the process strictly adheres to protocols.

Stochastic environment: An environment with randomness involved, where the same action in the same state might lead to different outcomes. Weather forecasting is an example, as it involves probabilistic elements. In the context of healthcare, disease outbreaks and epidemiological models often involve stochastic elements due to factors like population interactions and transmission rates.

AI agents are designed and adapted to suit various combinations of these environmental properties, ensuring their effectiveness and robustness in real-world applications where conditions can be diverse and unpredictable.

The **structure of an AI agent** can vary depending on its complexity, purpose, and the specific architecture or framework used for its design. However, AI agents generally consist of several key components and modules. The main modules that all AI agents have are the **perception**, the **knowledge base**, the **decision-making** and the **action** that an agent takes.

The **perception** module of an AI agent is the part of an AI system that is responsible for sensing and interpreting information from the environment. This involves collecting data through various sensors or input sources, such as cameras, microphones, or other sensors.

For instance, wearable devices, medical sensors and imaging technologies can provide information about a patient's vital signs, medical history, and current condition.

Another important aspect for AI agents is the **knowledge base**, which represents the repository of information that the AI agent has access to. This information may include facts, rules and models about the world, as well as the system's own internal state. The knowledge base is used to store and retrieve information that the AI agent needs to make informed decisions. In some AI systems, the knowledge base is pre-defined and programmed by developers, while in others, it may be learned and updated over time through machine learning.

The knowledge base in healthcare would contain information about medical conditions, treatment protocols, patient histories, and other relevant healthcare data. This knowledge base may be populated with established medical knowledge and can also be updated with new information and research findings

The **decision-making** component of an AI agent is responsible for analysing the information gathered from the perception module and the knowledge base to make decisions or take actions. This involves reasoning about the current state of the environment, considering goals or objectives, and determining the best course of action. Decision-making can be rule-based, involving predefined logic, or it can be learned through machine learning algorithms. Reinforcement learning (you will learn more about this in Module 8), for example, is a type of machine learning used in decision-making where the agent learns by trial and error.

The decision-making component in healthcare applications would analyse data from the perception module and information stored in the knowledge base to make decisions about diagnosis, treatment plans, and patient care. Machine learning algorithms could be used to assist in decision-making by learning patterns from vast amounts of medical data.

Last but not least, the **action** component is responsible for executing the decisions made by the decision-making module. This involves translating the high-level decisions into specific actions that the AI agent can take in the environment.

For instance, in a robotic surgery system, the action component might control motors and actuators to move the robot in a certain direction or perform a specific task. The action component in healthcare would also involve implementing the decisions made by the system. This could include generating alerts for healthcare providers, suggesting treatment plans, and coordinating actions such as medication administration, surgery, or other medical interventions.

So you just learned about the modules of an AI agent. Differences in these modules define different types of AI agents and their integration in healthcare.

2.5.3 Types of AI agents and their integration in healthcare

Different types of AI agents can be structured to perform specific functions in healthcare settings, so let's explore them.

AI agent types can be distinguished by differences in their modules, except for the perception module. The latter is common for all types of agents, and receives input data from sensors (e.g., vital signs of a patient).

Discover different types of agents and their integration in healthcare below:

Simple reflex agent

- Knowledge base: Contains condition-action rules (e.g., if temperature is high, administer antipyretic).

- Decision-making: Matches current input to rules and selects actions.
- Action: Executes predefined actions based on rules.

Example: A temperature-monitoring system in a hospital uses a simple reflex agent to administer medication when a patient's fever exceeds a certain threshold.

Model-based reflex agent

- Knowledge base: Stores an internal model of the environment (e.g., patient condition model).
- Decision-making: Uses the model to predict outcomes and selects actions.
- Action: Performs actions based on predictions.

Example: An asthma management system uses a model-based reflex agent to predict asthma exacerbations based on patient history and environmental data, adjusting medication accordingly.

Goal-based agent

- Knowledge base: Contains goals and a world model (e.g., treatment goals and patient health model).
- Decision-making: Plans and selects actions to achieve goals.
- Action: Executes actions to progress toward or achieve goals.

Example: A care plan recommendation system sets goals for managing a patient's chronic condition, creating a plan and suggesting interventions to reach those goals.

Utility-based agent

- Knowledge base: Stores utility functions (e.g., patient well-being) and preferences.
- Decision-making: Calculates expected utilities for different actions and selects the one with the highest utility.
- Action: Executes the action with the highest expected utility.

Example: An ICU ventilator adjusts settings based on a utility-based agent that maximizes patient comfort and oxygenation while minimizing lung damage.

Learning agent

- Knowledge base: Initially empty or with basic rules; learns from experience.
- Decision-making: Uses learned patterns and models for decision-making.
- Action: Adjusts actions based on past experiences and learning.

Example: A machine learning-based system learns to predict patient readmissions by analysing historical patient data and continually updating its predictive model.

Multi-agent systems (ensemble of agents)

- Perception: Each agent perceives its environment.
- Communication: Agents communicate and share information.
- Knowledge base: Agents have their own knowledge and may share some information.
- Decision-making: Agents make decisions individually or collaboratively.
- Action: Execute actions based on individual or collective decisions.

Example: In a hospital, a multi-agent AI system can be deployed to improve clinical care coordination and patient management. This system consists of multiple AI agents, each serving a specific role in patient care and administration.

These architectures illustrate how different types of AI agents can be structured to achieve specific actions in healthcare, ranging from basic reflex responses to complex goal-driven decision-making and collaborative multi-agent interactions.

In this subsection, you learned about AI agents, the actors that interact with and make decisions in the healthcare environment. We examined the architecture and functions of these agents in healthcare scenarios. Unfortunately, the healthcare setting is inherently uncertain. So understanding how AI deals with probability and uncertainty is vital. Let's go to the next subsection to learn about **probability and uncertainty in healthcare AI.**

2.6 Probability and uncertainty in healthcare AI

2.6.1 Probability and uncertainty in healthcare AI

While cleaning up after dinner, Zarah was pondering about the proposition of the researchers who had approached her to collaborate. These researchers were developing an AI tool to aid her in diagnosing complex medical cases, which would be groundbreaking. But Zarah felt uneasy about something, and decided to open up to her husband since he also works in the medical field.

"Eric," she began, her voice tinged with uncertainty, "do you ever wonder if AI in healthcare can handle the unpredictability we face every day?"

Eric looked up from sipping his wine, his gaze meeting hers. "You mean like how we never know what could happen with each patient?"

Zarah nodded, her brow furrowing. "Exactly. Like when symptoms don't fit neatly into a diagnosis or when treatments don't go as planned. How can AI compensate for that uncertainty?"

That is a great question, Zarah... Let's explore how AI in healthcare handles uncertainty and probability in this new subsection.

Within the context of AI in healthcare, the practical implementation of AI algorithms often encounters **complex real-world scenarios** characterized by **ambiguity and uncertainty**. Instead of having access to perfect information, AI systems must contend with a myriad of unknown variables, ranging from missing data to potential instances of deliberate deception.

Consider the case of applying **AI in healthcare diagnostics**. The primary objective is to provide efficient and accurate assessments of patients' medical conditions. However, the real world introduces numerous unpredictable factors, such as unexpected fluctuations in a patient's vital signs, sudden onset of symptoms, or even inaccuracies in medical data due to noisy sensors or data entry errors. In healthcare, AI systems rely on a diverse array of sensors and data sources, including medical imaging and patient records. These sensors are inherently imperfect, generating data that contains inherent errors and inaccuracies, often referred to as "noise". Consequently, it's not uncommon for one sensor to provide conflicting information compared to another, which necessitates resolving these discrepancies without prematurely halting the diagnostic process due to minor fluctuations or inconsistencies.

One of the key reasons why modern AI methods are effective in real-world healthcare applications, in contrast to earlier approaches from the 1960s to the 1980s, is their capacity to **handle uncertainty**

effectively. Throughout the history of AI, various paradigms emerged for managing uncertain and imprecise information. Fuzzy logic, for instance, was once considered a viable approach, particularly in consumer applications like washing machines, where it could adapt the wash cycle based on the degree of dirtiness rather than just clean or dirty. However, **probability has emerged as the predominant method for reasoning under uncertainty**, and nearly all contemporary AI applications incorporate probabilistic elements.

2.6.2 Significance of probability

Probability finds extensive applications in healthcare, extending beyond its traditional role in games of chance. In healthcare contexts, **probability quantifies and compares risks** in daily life. For instance, it can estimate the likelihood of side effects of a specific medication, forecast a disease outbreak, or assess the potential for AI to automate tasks such as identifying fractured bones in X-ray images.

The most important insight regarding probability is not its mathematical intricacies, but rather the conceptual ability to quantify uncertainty, treating it as a measurable quantity. This approach allows us to compare uncertainties and, in some cases, measure them. While **measuring probabilities** can be challenging and may require extensive data, systematic data collection enables us to evaluate probabilistic statements rigorously. This means that we can engage in rational discussions and make informed decisions, even in the presence of uncertainty.

Probability plays a vital role in healthcare AI, and generally in all AI applications, when it comes to quantifying these uncertainties. It allows the AI system to provide not only a diagnosis but also a **measure of confidence or probability associated with that diagnosis**. This probability can help healthcare professionals make informed decisions about treatment and follow-up examinations.

This ability to quantify uncertainty is pivotal, especially in matters like vaccine development and public policy decisions. Before a vaccine is approved for use, **rigorous clinical testing quantifies both its benefits and risks**, allowing for an informed evaluation of whether the benefits outweigh the potential drawbacks.

Recognizing that uncertainty can be quantified is crucial, as it ensures that **discussions and decisions are rational and evidence-based**. Failure to quantify uncertainty can hinder rational discourse and lead to misguided decisions. For instance, fearing potential side effects without quantifying their likelihood may discourage vaccination against life-threatening diseases. This understanding of uncertainty and probability is highly applicable in various real-world scenarios, including those within the medical field. Medical professionals, judges in legal proceedings, and investors frequently encounter uncertain information and must make sound decisions based on probabilistic assessments. In the context of this AI course, we will explore how probability can automate reasoning under uncertainty in healthcare.

Example: Consider a medical diagnostic AI system designed to assist radiologists in interpreting X-ray images for fractures. The primary goal of this AI system is to accurately detect fractures and provide a diagnosis. However, in real-world healthcare scenarios, uncertainties abound. For instance, when analysing an X-ray, the AI system relies on various sensors and algorithms to identify potential fractures. These sensors may include high-resolution imaging devices and advanced image processing software. However, the data obtained from these sensors is never entirely flawless and may contain noise, artifacts, or variations due to factors like patient positioning or equipment calibration.

In this context, the AI system needs to deal with the uncertainty of whether what it detects as a fracture is indeed a true positive or a false positive. Additionally, it must consider the possibility of missing a

genuine fracture, leading to a false negative result. This uncertainty stems from the inherent imperfections and variability in the data.

We can conclude that probability is of great significance in healthcare AI. Quantifying probability is essential for assessing the likelihood of medical outcomes and uncertainties. Let's go to the next page to see how we can quantify probability.

2.6.3 Quantifying probability

In AI-driven (healthcare) applications, **quantifying probability is essential for assessing the likelihood of various (medical) outcomes and uncertainties**. This quantification helps AI systems make informed decisions, predict patient outcomes, and provide recommendations to medical professionals.

In order to quantify probability, you should understand the following **concepts and metrics**:

Expectation (Mean): represents the average value of a dataset. It is calculated by summing all values in a dataset and dividing by the total number of values. It provides a measure of central tendency and represents the typical or average value in the dataset. The mean is sensitive to extreme outliers.

The mean is commonly used when you want to find the average or typical value in a dataset. It's appropriate for data that is approximately symmetrically distributed and not heavily influenced by outliers. For example, it's useful for calculating the average patient age in a hospital or the average test score in a class.

Median: is the middle value in a dataset when the values are arranged in ascending or descending order. If there's an even number of values, it's the average of the two middle values. The median is a robust measure of central tendency that is not affected by extreme outliers. It represents the "typical" value that separates the dataset into two equal halves.

The median is preferred when dealing with data that may contain outliers or is skewed. It provides a more robust measure of central tendency in such cases. For instance, the median is commonly used in healthcare to represent the "typical" patient's length of stay in a hospital. Hospital stays can vary significantly, with some patients having very short stays and others much longer. Using the median length of stay provides a robust measure of the central tendency in this context, ensuring that extreme outliers (such as patients with unusually long stays) do not unduly influence the measure.

Variance and standard deviation: Variance is often used when you want to understand the extent to which individual data points deviate from the mean (average) of the dataset. It quantifies the average of the squared differences between each data point and the mean. Standard deviation is used when you want a more interpretable measure of the spread or dispersion of data. It is the square root of the variance and is expressed in the same units as the original data. It quantifies the spread of data and is particularly useful for understanding the variability in a dataset.

Another important concept to understand for quantifying probability in healthcare, is the following:

Conditional probability: is the probability of an event occurring given that another event has already occurred. It is denoted as $P(A|B)$, where A and B are events. Conditional probability is essential for modelling dependencies between variables, such as the probability of a patient having a particular disease given their medical history.

Example: Consider the probability of a patient having Parkinson's Disease (P), given that the patient is male (M). It is known that the prevalence of Parkinson's for men is at least 1.5 times higher than for women. The conditional probability in this case is represented as P(P|M).

To understand the importance of conditional probability, let's do the following exercises as an example.

Exercise 1

A patient with Parkinson's Disease comes to our hospital in Belgium. The origin of the patient is one of the following 4 countries: Qatar, Armenia, Georgia or Bahrain.

Which country is the most probable country of origin of our patient? For each country, calculate the probability for it being the country of origin. The only information you have, is the total population in each country. For this exercise, you may assume that the prevalence of Parkinson's Disease is the same for each country. The probability will be defined based on the total population.

Total population:

Qatar: 2.7 million → P(Q) = 2.7/10.8 = 0.25

Armenia 2.8 million → P(A) = 2.8/10.8 = 0.26

Georgia 3.7 million → P(G) = 3.7/10.8 = 0.34

Bahrain: 1.6 million → P(B) = 1.6/10.8 = 0.15

➔ Georgia

Exercise 2

Now given that the sex of the patient is male, **what will be the conditional probability P(P|M) of each country?** Select the most probable country, given the following information:

Percentage of males in each country:

Qatar = 72% → P(Q|M) = (0.72*2.7)/5.88 = 0.32

Armenia = 45% → P(A|M) = (0.45*2.8)/5.88 = 0.21

Georgia = 47% → P(G|M) = (0.47*3.7)/5.88 = 0.30

Bahrain = 62% → P(B|M) = (0.62*1.6)/5.88 = 0.17

➔ Qatar

Joint distribution: is a probability distribution that describes the probabilities of multiple random variables occurring simultaneously. It provides information about the co-occurrence of events involving these variables. For example, in healthcare, a joint distribution can represent the probabilities of having multiple medical conditions simultaneously.

Example: In healthcare, a joint distribution can represent the probabilities of a patient having both diabetes (D) and high blood pressure (HBP) simultaneously. P (D,HBP|) would describe the joint distribution.

Maximum likelihood estimation (MLE): is a method for estimating the parameters of a probability distribution that are most likely to have generated observed data. It is commonly used for fitting probability distributions to empirical data, which can be useful in healthcare research.

Example: MLE is used to estimate parameters in a probability distribution. In healthcare, it can be applied to estimate the parameters of a distribution to model the survival times of patients.

Hypothesis testing: is a statistical method used to assess the validity of a hypothesis based on observed data. In healthcare, this can involve testing the effectiveness of a new treatment compared to a control group or evaluating the significance of a diagnostic test.

Example: Hypothesis testing can be used to determine if a new drug treatment significantly reduces blood pressure compared to a placebo. The null hypothesis (H0) might be that there is no difference, while the alternative hypothesis (Ha) is that there is a significant difference.

Confidence interval: is a range of values constructed from sample data that is likely to contain the true value of a population parameter with a certain level of confidence. Confidence intervals are used in healthcare to express the uncertainty surrounding estimates, such as the mean blood pressure of a patient population.

Example: In healthcare, a 95% confidence interval for the mean cholesterol level in a population might be [180, 200] mg/dL. This interval indicates our level of confidence in the true population mean cholesterol level.

All of these concepts and metrics that we discussed, are foundational in AI and statistics for **quantifying uncertainty, making predictions, and drawing conclusions from data**. In the context of healthcare AI, they play a crucial role in developing accurate models for medical diagnosis, treatment planning, and patient care.

2.6.4 Bayes' rule

Finally, we will introduce a fundamental concept in probability theory: **Bayes' rule**, also known as Bayes' theorem. It describes the probability of an event, based on prior knowledge of conditions that might be related to the event. It is useful in the field of healthcare, where making accurate medical diagnoses and treatment decisions involves dealing with uncertainty and the need to integrate various pieces of evidence.

Bayes' rule, named after the 18th-century statistician Thomas Bayes, is a fundamental concept that plays a crucial role in medical decision-making. It allows healthcare professionals and AI systems to update beliefs and make informed judgments based on both prior knowledge and new evidence. In the realm of AI-driven healthcare, Bayes' rule emerges as a fundamental and powerful **formula for processing uncertain information and combining conflicting evidence**. This concept, though rooted in probability calculus, holds immense practical value across various medical applications.

Before delving into the practical application, it is essential to grasp some **key terminology** associated with Bayes' rule:

- **Prior and posterior odds**: Bayes' rule operates by transforming prior odds, representing our initial assessment of the likelihood of an event, into posterior odds. Posterior odds reflect the updated probabilities after considering new information. In healthcare, this concept allows us to refine our understanding of a patient's condition as more data becomes available.
- **Likelihood ratio**: The likelihood ratio is a statistical measure that quantifies how much more likely an observation or test result is under one hypothesis (event A) compared to another hypothesis (event B). In the context of Bayes' rule and probability theory, it is a fundamental concept used to update probabilities and make informed decisions based on evidence. The

likelihood ratio is the probability of the observation in case the event of interest, divided by the probability of the observation in case of no event.

Bayes' rule is computed by the following function:

Posterior odds = Likelihood ratio × Prior odds

$P(A|B) = P(B|A) \times P(A) / P(B)$

P(A|B) is the probability of event A occurring, given that event B has occurred.

P(B|A) is the probability of event B occurring, given that event A has occurred.

P(A) and P(B) are the probabilities of events A and B occurring independently of each other.

To illustrate the potency of Bayes' rule, let's embark on simple **medical diagnosis scenarios**. The following examples will emphasize how our intuitive judgments often struggle to navigate conflicting evidence, underscoring the need for a systematic approach.

Example 1

Suppose a patient undergoes a diagnostic test for a rare disease, let's call it Disease X. The test is highly sensitive, meaning it rarely gives false negatives (i.e., if you have the disease, the test is likely to detect it). However, it is not very specific, meaning it can sometimes produce false positives (i.e., indicating the disease when it's not present).

- **Prior probability (Prior odds)**: Before the test results are available, we estimate the patient's probability of having Disease X based on their medical history, symptoms, and risk factors. Let's say the prior probability is 10%.
- **Likelihood ratio**: We know from medical research that the likelihood ratio of the test is 9, meaning it is nine times more likely to be positive if the patient has the disease than if they don't.
- **Posterior probability (Posterior odds)**: After conducting the test and obtaining a positive result, we want to update our estimate of the patient's probability of having Disease X. Bayes' Rule allows us to calculate this new probability.

Using Bayes' Rule:

- Posterior odds = Likelihood ratio × Prior odds
- Posterior odds = 9 (Likelihood ratio) × 10% (Prior odds) = 90%

This means that after a positive test result, the patient's probability of having Disease X increases from 10% to 90%. The integration of prior information with the test result is facilitated by Bayes' rule, aiding healthcare professionals in making more accurate diagnoses.

Example 2

A test for HIV gives a positive result 98% of the time if someone carries the virus. However, there is a 3% chance that the test will be falsely positive if someone does not carry the virus. If we know that the prevalence of HIV in Sub-Saharan countries in adults is 5%, what is the **probability of someone carrying HIV when they test positive?**

P(carrying HIV given test is positive) = P(test is positive given carrying HIV) x P(carrying HIV) / P(test is positive) = 0.98 x 0.05 / (0.98 x 0.05 + 0.03 x 0.95) = 63.22%,

where the **denominator** is derived using the following probability equivalence:

$P(B) = P(B|A) \times P(A) + P(B|\text{no } A) \times P(\text{no } A)$

where **B** is the event of the test being positive and **A** the event of carrying the virus.

In healthcare, Bayes' rule facilitates evidence-based decision-making by seamlessly incorporating new data and prior beliefs. Whether it's medical testing, treatment evaluation, or disease prediction, Bayes' rule empowers healthcare professionals and AI systems to navigate uncertainty and provide more accurate and personalized care.

2.7 Data-driven AI

2.7.1 Data-driven AI in healthcare

Data is the lifeblood of AI in healthcare. In this new subsection, we will discuss the concept of **data-driven AI and why it is pivotal in making healthcare decisions more evidence-based and precise**. From personalized diagnostics to predictive analytics, get ready to explore the intersection of data and healthcare.

In the 21st century, we find ourselves immersed in an era characterized by an overwhelming **surge in data creation**. Globally, the scale of this digital tidal wave is staggering. Recent statistics indicate that the total volume of digital data is doubling every two years, with the world set to generate an estimated 175 zettabytes of data by 2025. This explosion encompasses many sources, ranging from social media interactions and online transactions to wearables and smartphones, which provide data every second.

This surge in data is not a mere abstract phenomenon, but has transformative implications across various sectors. Among these, the healthcare industry stands out as a crucible for change, experiencing a profound metamorphosis from paper-based record-keeping to a digitized landscape, replete with Electronic Health Records (EHRs) and cutting-edge wearables.

Healthcare data sources

Healthcare data can come from a variety of sources. Are you familiar with healthcare data sources?

The combination of diverse healthcare data sources has orchestrated a paradigm shift in the landscape of healthcare.

- **Wearable devices**, from smartwatches to fitness trackers, have empowered individuals with real-time health insights, enabling proactive management of well-being.
- **Electronic Health Records (EHRs)** have streamlined information exchange among healthcare providers, enhancing care coordination and patient outcomes.
- **Genomic data**, once a scientific frontier, is now a cornerstone for personalized medicine, tailoring treatment plans to individual genetic profiles.
- **Smart home devices**, coupled with environmental sensors, contribute valuable data for understanding and addressing health-related environmental factors.
- **Social media** and the sharing of health experiences online have fostered a sense of community and awareness.
- **Medical devices and fitness equipment** have transformed homes into hubs for continuous health monitoring and preventive care.

Collectively, these data sources mark a transformative era, where technology-driven insights propel healthcare into a personalized, proactive, and interconnected realm, significantly impacting the way individuals manage and professionals deliver healthcare services.

Battle against data overload

The **healthcare data revolution** is epitomized by the migration from analogue to digital platforms. According to a report by the World Health Organization (WHO), over 95% of member states now have some form of **digital health information system** in place. This transition has facilitated not only the efficient exchange of health information, but also the creation of an expansive reservoir of data, encompassing diverse facets of patient health, medical histories, and treatment outcomes.

However, the **sheer volume and complexity** of healthcare data pose a formidable challenge for professionals in the field. Doctors and nurses, tasked with navigating this intricate web of information, find themselves on the front lines of a battle against data overload. Traditional methods of data analysis, reliant on **manual processing**, are increasingly untenable in the face of the exponential growth in healthcare data.

Healthcare professionals, despite their expertise, are grappling with the overwhelming task of keeping up with the expanding amount of patient data, medical literature, and the continuous stream of emerging research. A study published in the Journal of the American Medical Association (JAMA) found that physicians spend, on average, only **27% of their time on direct patient care, with the rest dedicated to administrative tasks**, including data management. The consequences of this data deluge are profound. Doctors risk being buried under a mountain of information, leading to potential delays in diagnosis, compromised decision-making, and suboptimal patient care. Recognizing these challenges, the healthcare industry is turning to a powerful ally: **Artificial Intelligence**.

2.7.2 AI: A beacon in the data storm

Artificial Intelligence emerges as a beacon in the storm of healthcare data overload. Its capacity to process vast datasets at unparalleled speeds and extract nuanced insights, positions AI as a transformative force. Machine learning and deep learning algorithms empower AI systems to **identify intricate correlations within healthcare datasets**. This enables predictive analytics, personalized treatment plans, and early interventions, enhancing the overall efficiency of healthcare delivery. Through this collaboration between human expertise and AI capabilities, the healthcare industry can navigate the complex seas of data more effectively.

Beyond being a challenge, the explosion of healthcare data represents an unparalleled opportunity for **AI innovation**. The massive quantity and diversity of available data serve as the raw material for training and refining AI algorithms. The continuous influx of data, when harnessed by AI, contributes to the **refinement of algorithms**, leading to improved accuracy in diagnoses, treatment recommendations, and overall healthcare outcomes. This synergy between data quantity and AI innovation creates a dynamic feedback loop, propelling advancements in medical research, drug development, and the delivery of personalized medicine.

Data-driven AI stands at the forefront of technological innovation, heralding a new era in problem-solving and decision-making. At its essence, data-driven AI is a paradigm where intelligent systems learn and evolve by ingesting vast amounts of data, extracting patterns, and making predictions or decisions without explicit programming.

- The approach of data-driven AI relies on the synergistic relationship between **advanced algorithms** and the **abundance of diverse datasets**. In healthcare for example, data-driven AI

can analyse electronic health records, genomic information, and real-time patient data to provide personalized treatment plans and predictive insights.

- The driving force behind data-driven AI is its ability to continuously **learn and adapt**, refining its understanding and performance over time as it encounters new information.

As we navigate an increasingly complex and interconnected world, data-driven AI emerges as a powerful tool capable of unlocking transformative solutions across various domains, from healthcare and finance to education and beyond.

Methodologies in data-driven AI

Data-driven AI encompasses various approaches, among which **deep learning** and **feature-based machine learning** stand out as distinctive methodologies.

In the realm of AI, **machine learning** serves as the overarching paradigm, a dynamic field where systems learn from data to improve their performance on a task without being explicitly programmed.

Feature-based machine learning involves the extraction of relevant features from the input data, which are then used to train models to make predictions or decisions. This approach often requires human experts to identify and define these features.
Deep learning, a subset of machine learning, involves neural networks with multiple layers (deep neural networks). Unlike traditional feature-based methods, deep learning algorithms automatically learn hierarchical representations of data, eliminating the need for manual feature engineering.

In healthcare for instance, feature-based machine learning might rely on predefined health indicators, while deep learning can autonomously identify intricate patterns in medical imaging data.

As we delve into the landscape of AI, understanding the nuances between feature-based machine learning and deep learning becomes pivotal for harnessing the full potential of data-driven solutions across diverse domains. You will learn more about feature-based machine learning methods and approaches in Module 4, and we will delve into more details about neural networks in Module 8. For both approaches you must keep in mind that the most important part is the availability of data from which the model used can learn.

Tasks in data-driven AI

Data-driven AI encompasses two fundamental **tasks**: regression and classification.

Regression task: In regression, the objective is to predict continuous numerical values based on input features, hereby creating models that capture intricate relationships within data, such as forecasting disease outbreaks or estimating patient recovery times.
Classification task: Classification involves assigning predefined labels or categories to input data, categorizing instances into distinct classes. Applications range from medical image segmentation, where distinct types of tissue are identified and assigned specific labels, to medical diagnostics, where patients are classified into different disease categories based on test results.

These tasks lie at the core of data-driven AI, enabling machines to not only understand complex patterns and relationships within datasets, but also to make informed predictions and decisions across diverse domains. Go to the next page to practise distinguishing between regression and classification tasks.

2.7.3 Training, testing and validation

Training, testing, and validation are crucial steps in the development and evaluation of data-driven AI models. Without proper training, a model may not learn the underlying patterns in the data. Without testing, it's challenging to assess how well the model will perform in real-world scenarios. And without validation, it's difficult to optimize the model's performance and prevent issues like overfitting.

Training and testing in feature-based machine learning

In feature-based machine learning, the process involves selecting relevant features to build predictive models.

- The **training** phase consists of feeding historical data into the model, allowing it to discern patterns and relationships. However, the true measure of a model's prowess lies in its ability to generalize to new, unseen data.
- This is where the **testing** phase comes into play. A separate set of data, not seen during training, is used to evaluate the model's performance. The goal is to ensure that the model is not merely memorizing the training data (overfitting), but that it can accurately predict outcomes for novel instances.

Validation to enhance reliability

Enter the **validation** set, a key player in ensuring model robustness. It serves as a middle ground between training and testing, aiding in fine-tuning model parameters. By tweaking the model based on performance feedback from the validation set, we enhance its ability to generalize to diverse datasets.

The iterative process of training, validation, and testing ensures that the model evolves into a reliable tool for making predictions in real-world scenarios.

Deep learning's autonomous learning journey

In the realm of deep learning, the narrative takes a fascinating turn. Deep neural networks autonomously learn intricate representations from raw data, obviating the need for explicit feature engineering.

- The **training** phase involves exposing the neural network to diverse health data, enabling it to automatically extract hierarchical features.
- **Testing**, much like in feature-based machine learning, assesses the model's performance on unseen data. However, the abundance of parameters in deep neural networks calls for careful monitoring to prevent overfitting.

The pivotal role of data

At the core of both machine learning approaches lies the cornerstone of success: **data**. The quality, quantity, and diversity of data directly impact the models' ability to learn and generalize. A shortage of data can lead to poor model performance, while an excess can present challenges related to computational resources and overfitting.

As we usually say in AI: "**Garbage in – Garbage out**", so if the quality or quantity of the data is not adequate or the data is not helpful for the question at hand, then the model trained will also be poor.

2.8 Data visualization and exploration

2.8.1 Overview of health data

In this final subsection of Module 2, we will discuss **data visualization and exploration**, as these are key to extracting insights from healthcare data. We will delve into techniques and tools that enable healthcare professionals to harness the power of data. But first, we start with an overview of health data.

Sources of health data

In the realm of healthcare, data acts as the cornerstone upon which AI systems are built and refined. As we explained before, this data comes from a myriad of **sources**, each contributing valuable insights into various aspects of patient care and health outcomes.

- **Electronic Health Records (EHRs)** stand out for their comprehensive documentation of patient encounters, treatments and outcomes, serving as a primary data source for AI models aimed at improving diagnostic accuracy and patient care.
- **Wearable devices and mobile health applications** offer real-time data on patients' physiological parameters, enabling AI systems to monitor health trends and predict potential health events with remarkable precision.
- **Genomics and biobanks** provide a wealth of information that, when analysed through AI, can lead to groundbreaking advances in personalized medicine, tailoring treatments to the genetic makeup of individual patients.
- **Administrative and claims data**, though more bureaucratic, offer a macroscopic view of healthcare delivery and utilization, aiding in the optimization of healthcare services and policy planning through AI-driven analyses.

The significance of healthcare data in the context of AI cannot be overstated. It not only fuels the development of algorithms capable of diagnosing diseases with greater accuracy than ever before; it also enables the creation of predictive models that can anticipate health issues before they manifest, thereby shifting the focus from treatment to prevention.

As AI continues to evolve, the integration of diverse healthcare data sources will undoubtedly lead to more innovative solutions, transforming the healthcare landscape and enhancing the quality of life for patients around the world.

Categories of health data

There are different categories of health data (and data in general). We will discuss the following 3 main categories and explain their connection to AI.

Structured vs Unstructured data

- Structured data
    - Highly organized and easily searchable in databases.
    - Adheres to a specific format, making it easily analysable by algorithms.
    - Facilitates efficient data retrieval, analysis and reporting.
    - Crucial for clinical decision support systems, billing and health informatics.
- Unstructured data
    - Not organized in a predefined manner, making it more difficult to collect, process, and analyse.

    - Contains rich, detailed information that can provide deeper insights into patient conditions, treatment plans and outcomes.
    - Before using unstructured data for any statistical analysis, we must use advanced techniques like natural language processing (NLP) to extract valuable information, which is then structured prior to any further analysis

Quantitative vs Qualitative data

- Quantitative data
    - Numerical and can be measured and quantified.
    - Allows for statistical analysis and objective comparisons.
    - Essential for clinical research, epidemiological studies, and evidence-based medicine.
    - Enables the measurement of outcomes and the assessment of treatment effectiveness.
- Qualitative data
    - Descriptive and not numerical.
    - Provides context and understanding of the patient's experience and healthcare processes.
    - Offers insights into patient preferences, beliefs, and behaviours.
    - Valuable for improving patient-centred care, healthcare service delivery, and policy development.
    - Categorical variables fall into this category because they describe qualities or characteristics of data and are used to group data into categories or distinct groups based on shared features. Categorical variables can be further divided into:
        - Nominal variables: These are categories without any intrinsic ordering. Examples include blood type (A, B, AB, O), gender (male, female, other), or marital status (single, married, divorced).
        - Ordinal variables: These have a set order or ranking, but the intervals between the categories are not necessarily equal. Examples include stages of cancer (Stage I, II, III, IV), education level (high school, bachelor's, master's, doctorate), or Likert scale responses in surveys (strongly disagree, disagree, neutral, agree, strongly agree).

Tabular data vs Signals vs Images

- Tabular data
    - Structured in rows and columns like spreadsheets
    - Encompasses both categorical and numerical variables, such as patient demographics, diagnosis codes, lab results, and treatment information.
    - Highly amenable to statistical analysis and machine learning, making it indispensable for patient demographic analysis, epidemiological studies, and predictive modelling in healthcare.
- Signal data
    - Consists of continuous or discrete data points collected over time, reflecting physiological changes.
    - Includes data from electrocardiograms (ECG), electroencephalograms (EEG), and other modalities or wearable devices, capturing the dynamic nature of vital signs of our body functions.

    - Pivotal for real-time patient monitoring, diagnosing conditions like cardiac arrhythmias or neurological disorders, and understanding lifestyle patterns through physical activity analysis.
- Imaging data
    - Generated by medical imaging technologies.
    - Offers visual representations of the body's internal structures, providing high-dimensional spatial information crucial for disease diagnosis and treatment planning. This includes X-rays, MRI, CT scans, and ultrasounds (among others), which are analysed using advanced computer vision techniques to identify diseases, guide surgical procedures, and monitor treatment responses.

Each data type, from the structured records of tabular data to the dynamic physiological insights of signal data and the detailed anatomical views provided by imaging data, plays a vital role in enhancing patient care and advancing medical research through technology.

All these different categories bear their own characteristics, limitations and challenges which dictate the use of specific models, both for the preprocessing as well as for their analysis. Go to the next page to practise distinguishing between these different categories of health data.

2.8.2 Challenges and opportunities with data

In the healthcare sector, the growing use of data-driven technologies, especially AI, brings a host of challenges and opportunities, particularly in the realms of **data privacy, interoperability, and big data analytics.**

- **Data privacy** stands as a paramount concern, as healthcare data includes sensitive patient information that must be protected from unauthorized access and breaches. Regulations like HIPAA in the United States and GDPR in Europe set strict guidelines for handling patient data, but the increasing sophistication of cyber threats and the complexity of AI systems pose ongoing risks. Ensuring the privacy and security of healthcare data, while leveraging AI for advanced analytics and personalized medicine, remains a delicate balance, necessitating robust encryption methods, secure data-sharing protocols, and stringent compliance with legal frameworks. You will learn more about the challenges related to ethical and regulatory aspects of AI in healthcare in Module 10.

The integration of AI in healthcare also brings to the forefront the technical challenges associated with the nature of healthcare data itself, such as **heterogeneity, noise, and missing values**.

- Healthcare datasets often comprise a mix of structured and unstructured data, ranging from numerical lab results and imaging data to free-text clinical notes, which vary greatly in format, quality, and context. This **heterogeneity** requires sophisticated preprocessing and normalization techniques to ensure that data from disparate sources can be effectively integrated and analysed by AI systems.
- Healthcare data is frequently plagued by **noise**—irrelevant or erroneous information that can obscure meaningful patterns and lead to inaccurate analyses. Identifying and filtering out this noise without losing valuable data is a critical step in the data preparation process.
- **Missing values** are a common issue in healthcare datasets, resulting from non-recorded observations, lost data, or inconsistencies in data collection practices. Handling missing data appropriately is crucial, as improper imputation strategies can introduce bias or distort the underlying data distribution, significantly impacting the performance and reliability of AI models.

These challenges necessitate advanced data engineering and AI techniques to prepare, cleanse, and standardize healthcare data, laying a solid foundation for robust and accurate AI-driven analysis and decision-making.

Conversely, the opportunities presented by AI in addressing these challenges are immense. In this subsection, we will discuss several ways to deal with these challenges.

2.8.3 Data exploration techniques: Descriptive statistics

Before diving into the complexities of AI algorithms and model building, it is imperative to understand the raw material we work with - the data. **Data exploration** represents this crucial first step, allowing us to sift through vast amounts of healthcare information to uncover initial insights, patterns, and crucially, the questions we need to ask. It is similar to laying the groundwork for a building; the strength and reliability of the structure (our AI models) depend significantly on the quality of the foundation.

The path of data exploration in healthcare is swarming with challenges, from the sheer volume and variety of data to concerns around privacy and data quality. Yet, it is within these challenges that opportunities arise. By employing sophisticated data exploration techniques, we can begin to unlock the stories hidden within the data, stories that can guide clinical decisions, inform public health policies, and drive forward the frontiers of medical research.

We will explore the following techniques for data exploration in this subsection: **descriptive statistics and data visualization.**

Descriptive statistics provide a powerful toolkit for summarizing and understanding the vast and complex datasets typical in healthcare. By applying these statistical measures, we can gain initial insights into patient populations, treatment outcomes, and other critical healthcare variables. Descriptive statistics include the following measures:

Measures of descriptive statistics

Measures of central tendency

- **Mean** (see quantifying probability)
- **Median**: (see quantifying probability) The median is particularly useful in skewed distributions, providing a more accurate representation of the dataset's central tendency, such as the median length of hospital stays.
- **Mode**: The mode is the most frequently occurring value in a dataset. In healthcare, the mode can highlight the most common diagnosis in a patient dataset or the most frequently prescribed medication.

Measures of dispersion

- **Range**: The range is the difference between the highest and lowest values in a dataset. The range can give an idea of the spread of values, such as the range of heart rates observed in a patient group during a stress test.
- **Interquartile range (IQR)**: Measures the spread of the middle 50% of values. The IQR is particularly useful in understanding the variability in patient responses to a treatment, minimizing the impact of extreme outliers.
- **Variance and standard deviation**: Variance measures the average of the squared differences from the mean, while the standard deviation is the square root of the variance. Both provide insights into the variability of a dataset. In healthcare, understanding the variance and standard

deviation of patient outcomes can indicate the consistency of treatment effects (see quantifying probability).

Distribution analysis

- **Skewness**: Indicates the degree of asymmetry of a distribution around its mean. In healthcare, analysing skewness can help in understanding the distribution of clinical measurement data, such as blood glucose levels.
- **Kurtosis**: Measures the "tailedness" of the distribution. High kurtosis in healthcare data might indicate the presence of outliers, such as extremely high hospital charges or rare adverse reactions to a medication.

How can descriptive statistics be used?

- Descriptive statistics help in **identifying which variables** may have the most significant impact on the outcomes of interest. For example, mean and median values can highlight key variables that warrant closer examination or inclusion as features in a model.
- Measures of central tendency and dispersion provide a snapshot of the data's distribution, crucial for **selecting appropriate AI algorithms**. For instance, data skewed towards older age groups might influence the choice of algorithms in predicting healthcare outcomes for geriatric care.
- Descriptive analysis can **reveal outliers or anomalies** that could potentially skew the results of predictive models. Early detection allows for the cleaning and preprocessing of data, ensuring that models are trained on accurate and representative data.

Descriptive statistics serve as the bedrock of data analysis in healthcare, offering a gateway to understanding the complex narratives within healthcare datasets. As we progress further into data visualization techniques, these statistical measures will continue to play a crucial role in interpreting visual data representations and in laying the groundwork for predictive analytics and AI modelling in healthcare.

2.8.4 Data exploration techniques: Data visualization

Another way of exploring data is by data visualization. Applying **data visualization techniques** in the context of AI for healthcare, allows us to uncover patterns and insights that descriptive statistics alone might not fully reveal. Visualizations can make complex data more accessible and understandable, which is crucial in healthcare where data-driven decisions can have significant impacts on patient outcomes.

Data visualization is a compelling way to communicate healthcare data insights. Effective visualizations can highlight hidden patterns, compare treatment outcomes, and track disease progression over time, making them indispensable in both clinical and research settings. The most commonly used visualization tools are the following:

- **Bar charts and histograms**: Ideal for comparing categorical data (bar charts) and distributions of continuous data (histograms). For example, bar charts can compare the number of patients with different conditions across clinics, while histograms might display the distribution of patient ages within a study population.
- **Line graphs**: Perfect for illustrating trends over time, such as the progression of average blood glucose levels in a diabetic patient cohort or tracking the number of new cases during an epidemic.

- **Scatter plots**: Useful for exploring relationships between two variables, such as the correlation between BMI and blood pressure among patients. Scatter plots can also help in identifying clusters or patterns that might indicate subgroups within the data.
- **Box plots**: Provide a visual summary of the distribution of data, highlighting the median, quartiles, and outliers. In healthcare, box plots can be used to compare the distribution of a key metric, like cholesterol levels, across different patient groups.
- **Time series plots**: Essential for data that is sequential and time-dependent, such as tracking the spread of a disease over time or the effect of a new treatment across several months.
- **Heat maps**: Ideal for visualizing complex data matrices, like gene expression data or the correlation matrix of clinical variables. Heat maps can help identify areas of high activity or significant relationships between variables.

What is the correlation matrix?

Correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. The value is in the range of -1 to 1. A correlation of 1 means a perfect positive relationship, -1 means a perfect negative relationship, and 0 means no relationship.

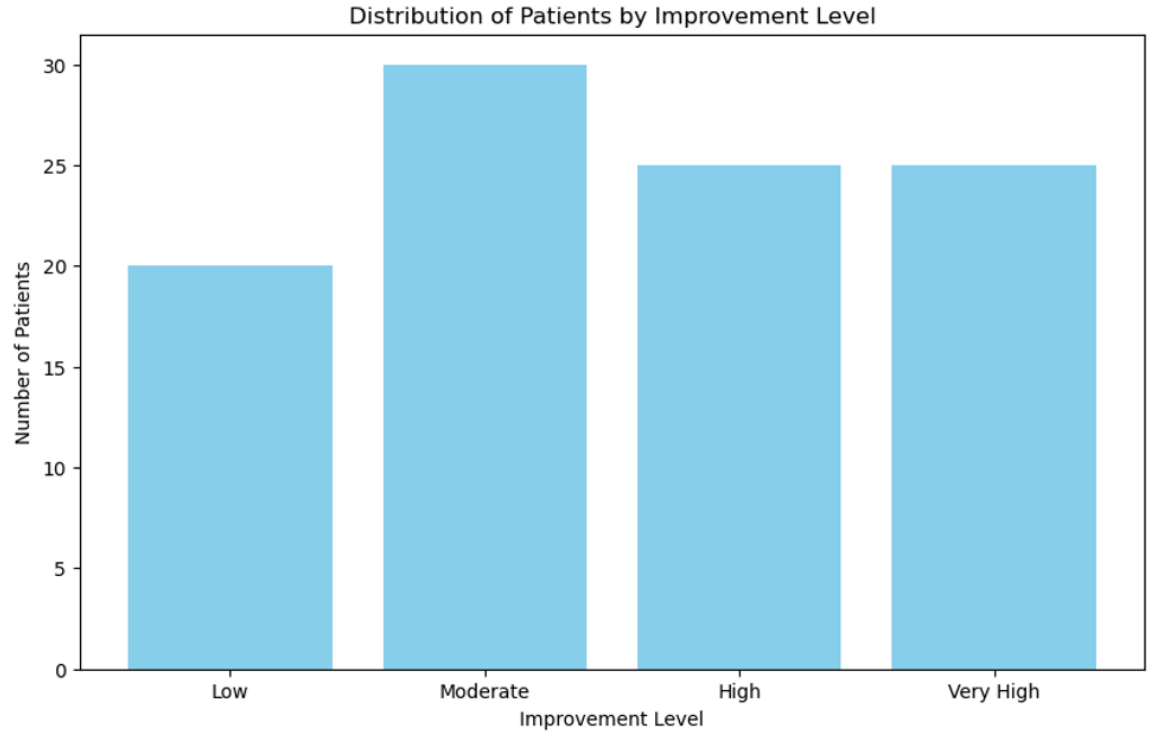
- The diagonal running from the top left to the bottom right represents the correlation of each variable with itself, which is always 1.
- Symmetry: The matrix is symmetrical around the diagonal because the correlation between variable A and variable B is the same as the correlation between variable B and variable A.

Data visualization acts as a bridge between raw data and actionable insights in healthcare, transforming complex datasets into intuitive and informative visuals. As we delve deeper into AI applications in healthcare, these visualization techniques not only aid in initial data exploration, but also play a crucial role in interpreting model outputs, explaining predictions to healthcare professionals, and ultimately guiding clinical decisions.

We will now explore an example with different visualizations. Let's consider a hypothetical study on a **diabetes management program**. This program monitors patients' blood glucose levels, medication adherence, and lifestyle factors over time, aiming to assess the program's effectiveness and to identify areas for improvement.

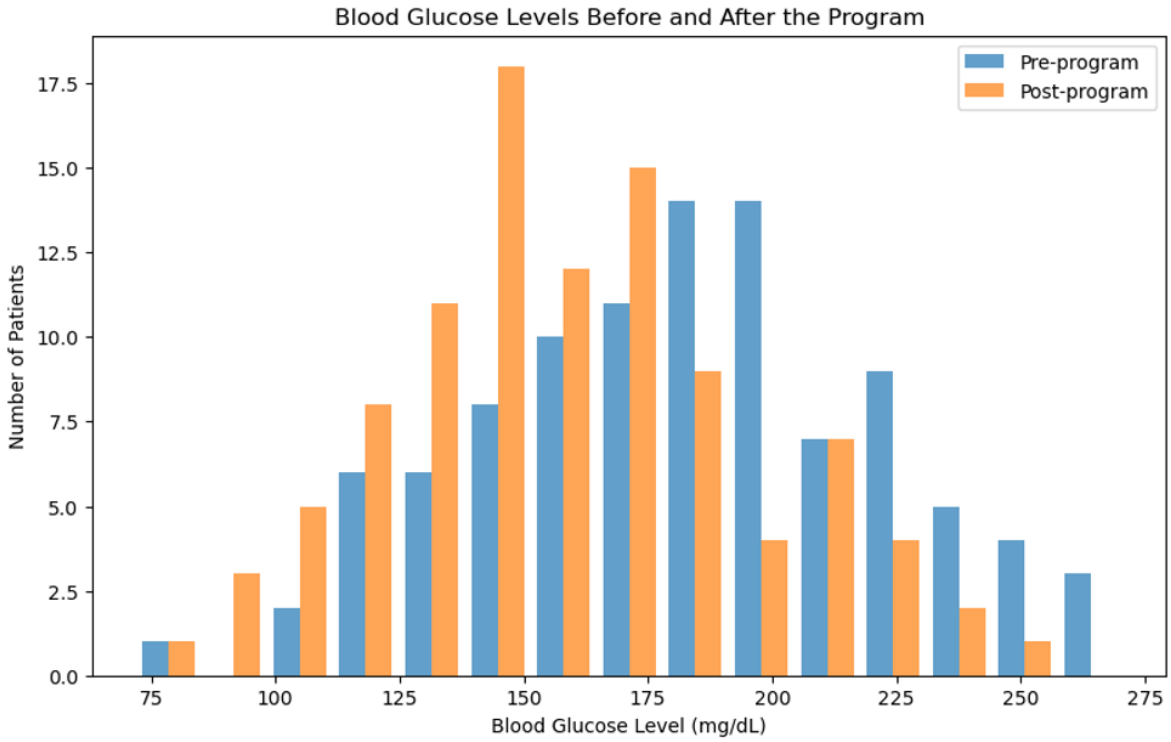
In this hypothetical scenario, we want to categorize the patients into four groups of **improvement level** ("low", "medium", "high" and "very high") after receiving medication.

Bar chart



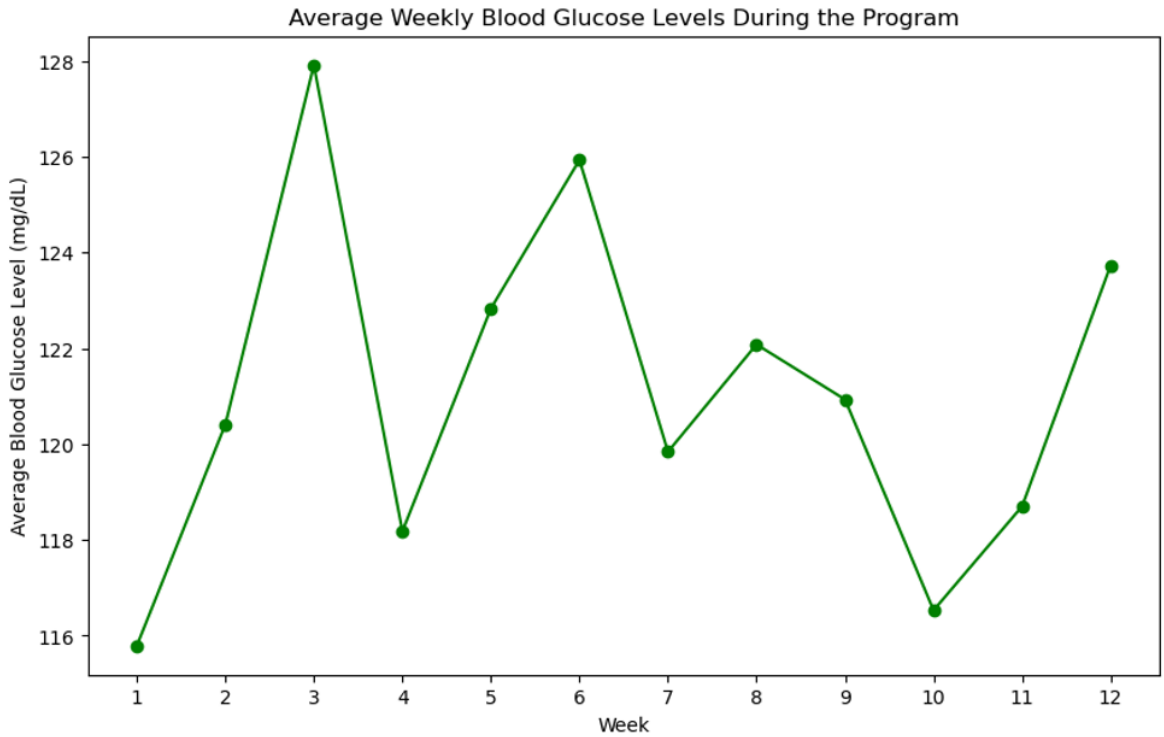
This **bar chart** can be used to visualize patients of the different categories based on their overall improvement level, combining factors like blood glucose reduction, medication adherence, and lifestyle score improvements.

Histogram



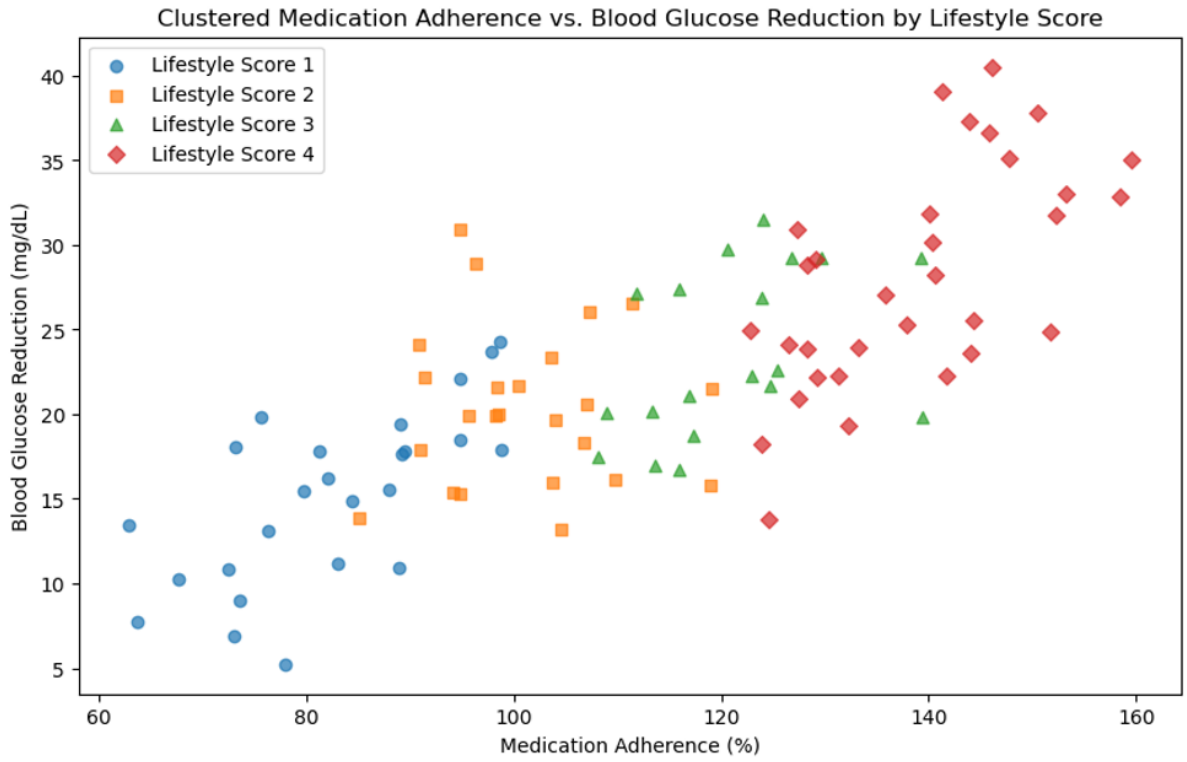
This **histogram** is employed to compare the distribution of the patients' blood glucose levels before and after participating in the diabetes management program.

Line graph



This **line graph** tracks the progression of average blood glucose levels among participants throughout the 12-week program.

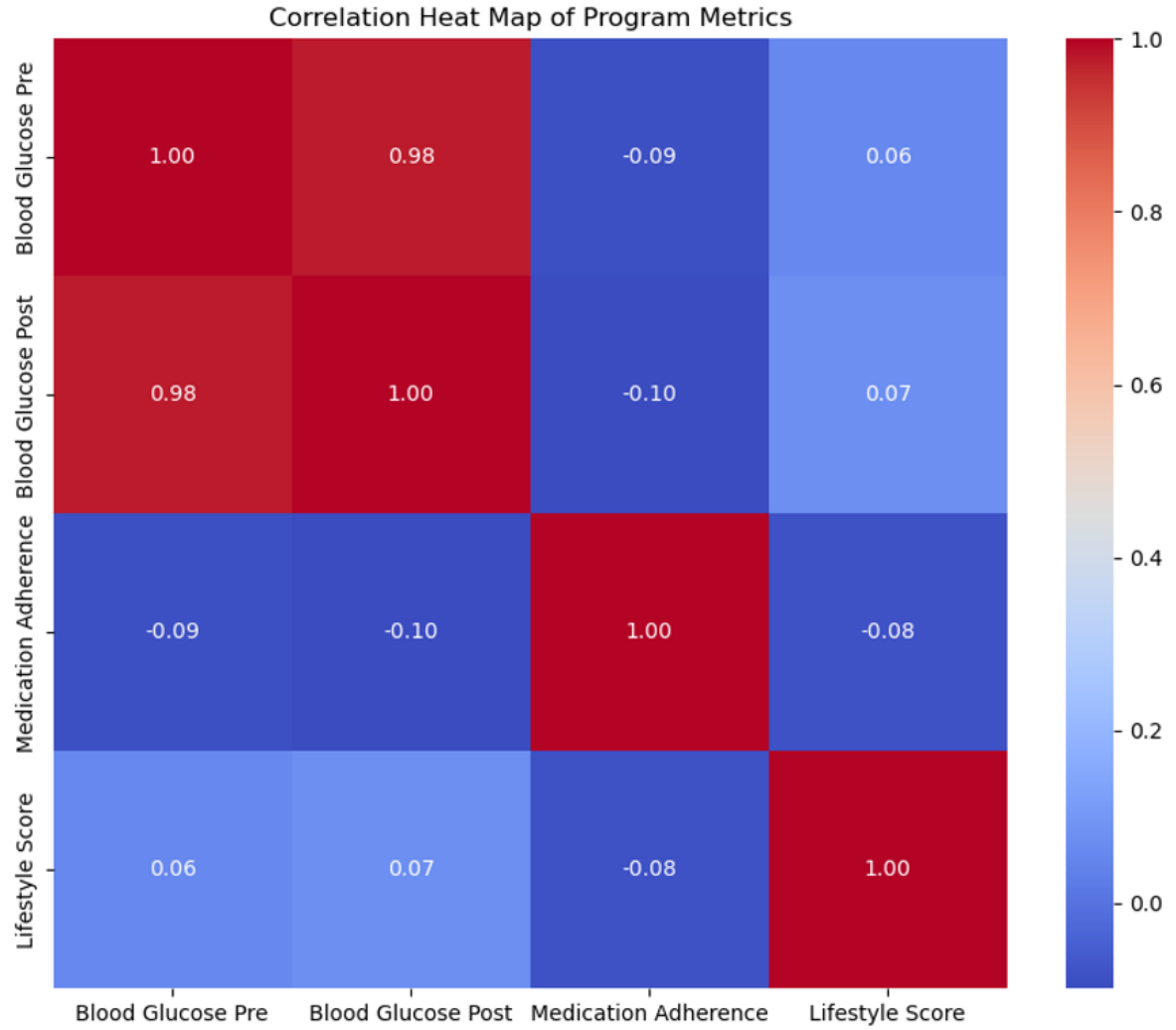
Scatter plot



In this **scatter plot**, each point represents a patient, with their position indicating their medication adherence and the reduction in their blood glucose levels. The colour of the points corresponds to the patient's lifestyle improvement score, with each colour representing a different score (1 to 4).

Clusters within the scatter plot may indicate patterns where patients with similar lifestyle scores have similar relationships between medication adherence and blood glucose reduction. This visualization helps to identify if certain lifestyle changes (reflected in the lifestyle scores) are associated with more pronounced reductions in blood glucose levels, particularly in conjunction with medication adherence. **By observing the clustering of points, healthcare professionals can infer potential relationships between lifestyle improvements and treatment outcomes**, guiding more personalized and effective diabetes management strategies. You can see in this example that subjects with Lifestyle Score 1 significantly differ from subjects with Lifestyle Score 4 concerning the medication adherence. This visualization technique can help in identifying patterns that suggest the effectiveness of lifestyle changes combined with medication adherence on blood glucose reduction.

Heat map - Correlation matrix



A **heat map** is a data visualization technique that uses colour coding to represent different values in a matrix. In our case, the heat map displays the **correlation matrix** of the program metrics, providing an intuitive and visual representation of the relationships between variables.

A strong positive correlation between 'Blood Glucose Pre' and 'Blood Glucose Post' metrics can be observed, because post-program levels should still be somewhat influenced by pre-program levels, despite expected reductions due to the management program.

The heat map is a powerful tool for quickly visualizing complex relationships in a dataset. In the context of our diabetes management program, it helps stakeholders quickly understand how different aspects of the program and patient behaviours are interrelated, aiding in the identification of key factors that contribute to the program's effectiveness. **By interpreting the colours and their intensities, healthcare professionals can make informed decisions to enhance the program's outcomes**, focusing on areas with the strongest correlations to patient improvements.

In healthcare data analysis, outliers can indicate important clinical findings or errors. Excluding outliers without investigation can lead to loss of valuable insights, while including them might skew the analysis. It is crucial to evaluate outliers on a case-by-case basis, considering the clinical context.

2.8.5 Data cleaning and preprocessing

Data cleaning and preprocessing are critical steps in the development of AI applications in healthcare, ensuring the accuracy, reliability, and validity of the models developed. The peculiarities of healthcare data, such as its heterogeneity, high dimensionality and the critical nature of accuracy, make these steps particularly challenging and crucial.

Identifying & correcting outliers

Outliers can significantly skew the results of data analysis and model training, leading to inaccurate predictions. Techniques:

- **Statistical methods**: Z-score, IQR (Interquartile Range) method for identifying outliers based on statistical metrics.
- **Visualization**: Box plots and scatter plots can help visually identify outliers.
- **Domain-specific thresholds**: Setting clinical relevance thresholds based on domain knowledge to identify and manage outliers.

Data consistency checks

Ensuring that the data across different sources or within the same dataset follows consistent formats and standards, especially critical in healthcare due to diverse data sources. Techniques:

- **Regular expressions**: For validating data formats (e.g., date and time stamps, medical codes).
- **Cross-field validation**: Ensuring logical relationships between fields (e.g., age and date of birth).

Handling missing values

Missing data is a common issue in healthcare datasets due to various reasons such as non-responses in surveys, errors in data entry, or omission of information in clinical records. Techniques:

- **Deletion**: Removing records with missing values, suitable for datasets with minimal missing data.
- **Imputation**: Substituting missing values with statistical estimates like mean, median (for numerical data), or mode (for categorical data). In order to substitute the missing values, we can also perform imputation based on closest neighbours, considering the similarity between observations to impute missing values.

Example: Let's consider the following example that includes patient cholesterol levels and BMI (Body Mass Index) with missing values. We will demonstrate how to perform both simple mean imputation and imputation based on closest neighbours.

Patient ID	Cholesterol (mg/dL)	BMI
1	200	22
2	Missing	25
3	180	Missing
4	220	24
5	Missing	21

Mean imputation

1. Calculate the mean of the observed values for each variable.
    1. Mean Cholesterol = (200 + 180 + 220) / 3 = 200 mg/dL
    2. Mean BMI = (22 + 25 + 24 + 21) / 4 = 23
2. Replace the "missing" values with the calculated mean.
    1. For Patient 2, replace "missing" cholesterol with 200 mg/dL.
    2. For Patient 3, replace "missing" BMI with 23.

Imputation based on closest neighbours

For this example, let's assume we use the 1-Nearest Neighbour approach based on the available variable.

1. For Patient 2 (missing cholesterol):
    1. The closest neighbour is determined based on the BMI value. Since Patient 2 has a BMI of 25, the closest neighbour is Patient 4 with a BMI of 24.
    2. Impute Patient 2's "missing" cholesterol value with Patient 4's cholesterol value (220 mg/dL).
2. For Patient 3 (missing BMI):
    1. The closest neighbour is based on the cholesterol level. Patient 3's cholesterol is 180 mg/dL, making Patient 1 (with cholesterol of 200 mg/dL) the closest neighbour.
    2. Impute Patient 3's missing BMI with Patient 1's BMI (22).

2.8.6 Data cleaning and preprocessing: Medical images and signals

The preprocessing methods explained until now are mainly used for tabular data (some of them can be used for signals or images, but not commonly). Data preprocessing is a crucial step in the analysis of **medical images and signals**, aiming to enhance the quality of the data and to make it more suitable for further processing or analysis. This step involves several techniques tailored to address the unique challenges presented by medical imaging and signal data.

Medical images preprocessing methods

- **Image enhancement**:
    - **Contrast** adjustment: Improves the visibility of features in an image by adjusting its contrast levels. Techniques like histogram equalization or contrast stretching are commonly used.
    - **Noise reduction**: Medical images, especially those from modalities like MRI or CT, can be affected by various types of noise various filters are applied to reduce noise while preserving important details.
- **Normalization**: Ensures consistency across images by standardizing their intensity values. This is particularly important in datasets acquired from different scanners or modalities, where intensity values for the same tissue can vary.

- **Registration**: Aligns two or more images into a common coordinate system, facilitating their comparison or integration. This is essential in studies where changes over time are monitored, or when images from different modalities need to be compared or combined.

Signal preprocessing methods

- **Filtering**:
    - **LowPass filtering**: Used to attenuate (reduce) the high-frequency components of a signal (e.g., muscle movement).
    - **HighPass filtering**: Used to attenuate (reduce) the low-frequency components of a signal (e.g., eye movement in EEG).
    - **Bandpass filtering**: Used to retain frequencies within a specific range while attenuating frequencies outside this range, useful in ECG or EEG signal processing to remove noise or artifacts.
- **Normalization**: Similar to image normalization, signal normalization adjusts the amplitude of signals to a common scale, facilitating comparison and analysis.
- **Artifact removal**: Identifying and removing non-physiological components from signals, such as muscle artifacts in EEG or motion artifacts in wearable sensor data.

Preprocessing in medical images and signals is tailored to address the specific challenges and characteristics of each data type. The choice of preprocessing methods depends on the nature of the data, the artifacts present, and the ultimate goal of the analysis or application. Proper preprocessing not only improves the quality of the data, but also enhances the performance and reliability of subsequent analysis, including AI models.

2.8.7 Data cleaning and preprocessing: Data harmonization

Data harmonization is a critical process in the healthcare sector, especially when dealing with data originating from multiple hospitals, to ensure consistency, interoperability, and the reliability of clinical insights derived from such data.

Electronic Health Records (EHRs), medical signals and images often vary significantly across institutions due to differences in data capture methodologies, technologies used, clinical protocols, and even regional healthcare regulations. This variability can lead to discrepancies in data formats, terminologies, and measurement scales, making it challenging to aggregate, compare or analyze data at a larger scale. For instance, one hospital might use a different set of codes or terminologies to describe medical procedures or diagnoses in EHRs compared to another, or they might employ different imaging protocols and resolutions, affecting the comparability of medical images. Similarly, medical signals such as ECGs or EEGs could be recorded with varying equipment settings and data formats, further complicating cross-institutional studies.

Harmonizing this data involves **standardizing** data formats, terminologies and protocols, and sometimes even **transforming** data values to a common scale, ensuring that datasets from different sources can be integrated accurately and meaningfully. This harmonization is crucial for enabling large-scale epidemiological studies, multi-centre clinical trials, and the development of robust AI models that require diverse and comprehensive datasets to improve their accuracy and generalizability. Without such efforts in data harmonization, the potential for healthcare data to inform clinical decision-making, public health policies, and personalized medicine approaches would be severely limited, underscoring the importance of this process in leveraging healthcare data to its fullest potential. In an attempt to harmonize the datasets, many of the methods explained in this subsection are being employed.

The **European Health Data Space (EHDS)** is an EU initiative designed to enhance the sharing and utilization of healthcare data across Europe, promoting better healthcare outcomes, innovation, and ensuring data privacy and security.

The integration and interoperability of diverse healthcare data, including EHRs, necessitate common data models like OMOP. These models standardize data across different sources, enabling seamless exchange and analysis. This standardization is crucial for the EHDS's goal of unlocking the full potential of health data for research, clinical care and policy-making, ensuring that healthcare data is accessible, usable, and beneficial across borders.

3 Inside the AI engine: Search and optimization

3.1 Welcome to Module 3

Welcome to Module 3! This module is focused on identifying search problems, and understanding common search and optimization algorithms in AI to solve and optimize these problems. Search problems present themselves in a wide range of applications across various fields, including logistics, operations and planning, and this in many domains, including healthcare, where efficient resource allocation and decision-making are crucial.

Key Focus Areas

- We define the three classic combinatorial optimization problems with examples in healthcare: the vehicle routing problem, the knapsack problem, and the nurse scheduling problem.
- We discuss additional search problems relevant to healthcare such as information retrieval and (game) strategy optimization.
- We show how most search and optimization problems can be formulated as constrained satisfaction and optimization problems.
- We explain search-graphs and search-trees, which are practical data representations to explore exponentially large search spaces.
- We make a distinction between uninformed and informed search algorithms, and describe the common search algorithms: Depth-first Search, Breadth-first Search, Hill Climbing Search, and Beam Search.
- We move from search to optimal search and explain the common A* algorithm.
- Finally, we introduce other optimization problems and the use of gradient descent and genetic algorithms as very general and practical techniques.

Why This Module Matters

This module equips you with essential knowledge to solve and optimize complex search problems, particularly within the healthcare domain, making it indispensable for efficient resource allocation and decision-making.

Learning goals

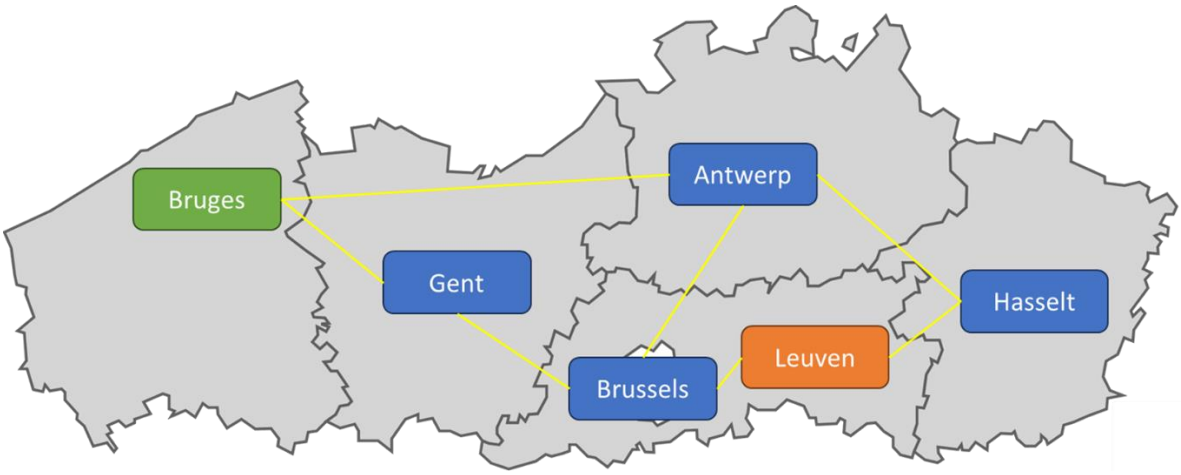
- Develop the ability to recognize various search problems within diverse healthcare scenarios.
- Acquire the skill to articulate a search problem in healthcare, representing it as a constrained satisfaction or optimization challenge involving variables, domains, constraints, and an objective function.
- Be proficient in distinguishing between uninformed, informed, and optimal search strategies. Also be able to engage in discussions and comparisons of commonly employed algorithms in terms of their completeness, convergence, optimality, time efficiency, and space complexity.
- Gain insight into the contrast between optimization and optimal search. Also elucidate algorithms using gradient descent and optimization techniques inspired by evolutionary processes.

3.2 The challenge of PharmaLog

Eric diligently logs his daily inventory and customer special orders at his pharmacy. To maintain a well-stocked inventory and fulfil unique customer requests not readily available in his pharmacy, Eric works with **PharmaLog**, a logistics company servicing pharmacies and hospitals across the Flanders region in

Belgium. PharmaLog's operational approach is straightforward: place an order before 10:00 AM, and receive your delivery on the same day! PharmaLog is also a progressive company and has started to use electrically powered **self-driving delivery vehicles (S-DV)**.

Despite Belgium's compact size and Flanders covering just half of it (Fig. 3.1 - Tip: consider copying and pasting this map into a separate document for convenient reference, as some exercises may require its use), PharmaLog's commitment of same-day deliveries necessitates meticulous planning of delivery routes. Specifically, the **warehouse of PharmaLog is located in Bruges, and Eric's pharmacy is located in Leuven**. The S-DV needs to plan its route from Bruges to Leuven, taking the available roads (yellow lines in Fig. 3.1) between the cities in Flanders and Brussels.



In planning its way to Eric's pharmacy, the PharmaLog's S-DV has multiple routing options, and **finding a suitable route is a classic example of a search problem**:

> In AI, a **search problem** refers to a task or challenge where an AI-agent or computer program needs to find a solution by exploring a set of possible states or actions (search space) in a systematic manner (search algorithm). It consists of defining an initial state, a goal state, a set of actions or operators that can transition the system from one state to another, and a search algorithm to find a sequence of actions leading from the initial state to the goal state.

In PharmaLog's search problem, the S-DV is the AI-agent, and the roads and cities constitute the environment through which the S-DV navigates and takes actions. The **initial state** is defined as the location of the warehouse in Bruges, and the **goal state** is the location of Eric's pharmacy in Leuven. The primary objective is to determine a sequence of actions or states, which translates to driving on a series of road segments, allowing the DV to transition from the initial state to the goal state. Try this for yourself, by connecting different cities across the different states listed, so that you start in Bruges and arrive in Leuven.

This task is relatively straightforward, given the small geographical area of Flanders. The **challenge** lies in scaling up the problem into larger areas and using thousands of different roads as well as in optimizing for various factors, such as time and cost. For instance, each road in Flanders incurs a specific cost, which corresponds to the electricity consumed by the S-DV during travel. Additionally, the roads vary in their susceptibility to heavy traffic or road works/blocks, which can result in potential time delays. Therefore, this search problem involves considering both sta**tic factors (road cost) and dynamic factors (road time)** within the environment of the AI-agent to make efficient and effective decisions for route planning.

Other criteria can also change. For example, it can be interesting to find the cheapest and therefore shortest route to Eric's pharmacy. However, one day, Eric makes an urgent delivery request, willing to pay extra for the service, and the S-DV needs to find the fastest route, taking peak-traffic into account. On another day, customers from each city including Eric make a standard delivery request, and the S-DV tries to plan its route to Eric's pharmacy, ensuring it has made deliveries to all the cities on the same day and this according to the cheapest overall route. Finally, PharmaLog is planning to expand, and instead of one S-DV, they purchased 2 more S-DVs. However, each of these S-DVs is limited in the amount of medication it can transport. It is clear that all these scenarios require **different objectives to be optimized**, and in the last case, also considering **multiple AI-agents**.

In a utopic version of Flanders, without traffic and where the distance between cities is the same, try to connect the cities as before, but now make sure you visit each city exactly once.

3.3 Textbook search problems

Now that you got introduced to what a search problem is, let's look at the following textbook search problems.

Vehicle Routing Problem

PharmaLog's challenge is a typical example of the vehicle routing problem, which is one of the three classical **combinatorial optimization problems (COP)** in AI. In COPs, the goal is to **find the best solution from a finite set of discrete choices** (all possible routes from Bruges to Leuven), typically involving a combination of elements (one route from Bruges to Leuven is a combination of cities visited, therefore cities = elements). The goal is to optimize an objective function (e.g., shortest, fastest, or complete route) while satisfying certain constraints (e.g., roadblocks, traffic, time-management, number of S-DVs etc.). Combinatorial optimization problems are characterized by the need to search through a large, often exponential, number of possible solutions to identify the optimal or close-to-optimal one. COPs have a wide range of applications in various fields, including logistics, scheduling, network design, and many other domains where efficient resource allocation and decision-making are crucial.

Knapsack Problem

The second classical COP is known as the knapsack problem, which is defined as follows: You are given a set of items, each with a weight and a value, and you have to **determine the most valuable combination of items to include in a knapsack with a limited weight capacity**. The goal is to maximize the total value of the items in the knapsack while not exceeding its weight limit.

For example, consider a scenario in healthcare supply chain management where a hospital has a limited budget and a storage room with a limited capacity for medical supplies. The hospital needs to purchase medical equipment and supplies to ensure it can provide quality care to patients. Each medical item has a cost (budget constraint) and takes up space in the storage room (capacity constraint). Additionally, each item has a different level of importance or utility (value) to the hospital, depending on its medical necessity or patient demand. The hospital faces the knapsack problem when deciding which medical items to purchase within its budget while ensuring they fit in the available storage space. The goal is to select a combination of medical supplies that maximizes the overall medical benefit (value) to the hospital while staying within budget and not exceeding storage capacity.

Nurse Scheduling Problem

The third classical combinatorial optimization problem is known as the nurse scheduling problem. Here the overall objective of the hospital is to **minimize labour costs, ensure adequate coverage of the hospital nursing floor, and consider nurse preferences**. However, there are several constraints to

consider: Each nurse has specific availability, skill sets, and maximum working hours per week. Shifts must be adequately staffed to meet patient demand and safety requirements. Some nurses may have preferences for particular shifts or days off. Compliance with labour laws and regulations is also important, such as maximum consecutive work hours. The goal is to create a schedule that efficiently utilizes the available nursing staff, minimizes labour costs (e.g., overtime pay), maintains quality patient care, and satisfies the preferences and constraints of the nurses and the healthcare facility.

Information Retrieval Problem

Aside from combinatorial optimization problems, search problems also present themselves in information retrieval problems like finding the minimum or maximum vital sign measurements (e.g., blood pressure, heart rate, temperature) or searching and retrieving electronic health records for patients with a specific medical condition or criteria. For instance, a medical researcher is conducting a study on patients with a rare genetic disorder called "XYZ Syndrome." The researcher needs to identify and collect medical records of patients who meet specific inclusion criteria for the study.

Strategy Problem

Other well-known search problems exist in strategy optimization, particularly in two-player games like chess. The key distinction here to the scenarios listed above, is that the **conditions continually evolve** as each player makes their move. In essence, the search for solutions in these games often revolves around predicting and selecting the optimal move at each stage of the game, aiming to secure a victory.

For example, during the past pandemic, in a hospital's intensive care unit (ICU), healthcare providers needed to continuously monitor and adjust treatment plans for critically ill covid-patients. The objective was to optimize treatment decisions and resource allocation to maximize patient outcomes, while considering various constraints and uncertainties. The goal in this search problem is to find an optimal treatment plan for each patient, considering the evolving medical conditions, available resources, and regulatory requirements. Similar to chess, this problem involves making a sequence of strategic decisions to maximize patient outcomes while dealing with dynamic and evolving conditions. Solving this search problem efficiently can lead to improved patient care in critical care settings.

A significant milestone in AI was when a computer defeated a human in a chess game, which was in 1997 when IBM's Deep Blue defeated World Chess Champion Garry Kasparov. This historic event marked a major advancement in AI, demonstrating the capability of computers to excel in complex strategy games like chess. Back then, it was the reason for a euphoric belief and boost in AI. The ability to solve combinatorial optimization problems along with more general search problems, led to a high increase of automation and optimization of labour and cost throughout the complete industry, not just healthcare. It is interesting to know that these search problems still exist today, and that the search algorithms from the 90s are still efficient in solving them. However, in contrast to the hype of AI today, these search algorithms do not learn from data, but merely provide an efficient way in exploring an exponential search space which is explained in the next few units.

3.4 Constrained satisfaction and optimization problems

Many, if not all, search problems can be formally defined as a **Constraint Satisfaction Problem (CSP)**. Here the goal is to find a consistent assignment of values to a set of variables, with respective variable domains (allowed range of values) and subject to a set of constraints that specify allowable combinations of values for those variables.

Defining a search problem as a CSP provides a systematic and structured approach to problem-solving, making it easier to analyse, model, and solve a wide range of complex problems across different domains. It helps in defining the problem space, including **variables, domains, and**

constraints, which makes it easier to understand and communicate problem requirements. In most cases, CSPs also include an **objective function** that needs to be minimized or maximized while satisfying the constraints. This is common in optimization problems derived from CSPs, known as Co**nstraint Optimization Problems**.

Let's see how we can translate most of the search problems from the previous unit into a constrained satisfaction or optimization problem. You will see that such problem formulations really capture a broad range of search problems.

The nurse scheduling problem as a CSP

- **Variables**: Each nurse is represented as a variable, and the values represent the shifts they can work (e.g., day shift, night shift).
- **Domains**: The domain for each nurse's variable consists of the available shifts they can work.
- **Constraints**: Constraints include ensuring that each shift is adequately staffed, adhering to labour laws (e.g., maximum consecutive work hours), and accommodating nurse preferences (e.g., granting specific days off).
- **Objective Function**: If needed, an objective function can be included to optimize scheduling objectives, such as minimizing overtime, maximizing nurse preferences, or evenly distributing shifts among nurses.

The knapsack problem as a CSP

- **Variables**: In the knapsack CSP, the variables represent the decision of whether or not to include each item in the knapsack. For example, for a set of items {A, B, C}, you have variables X_A, X_B, and X_C, where X_A = 1 if item A is included, and X_A = 0 if it is not.
- **Domains**: The domain for each variable is typically {0, 1}, indicating that you can either include an item (1) or not include it (0) in the knapsack.
- **Constraints**: The main constraint in the Knapsack CSP is the weight constraint. The sum of the products of the selected items' weights and their corresponding decision variables (X_A * weight_A + X_B * weight_B + X_C * weight_C) must not exceed the knapsack's weight capacity. This constraint ensures that the total weight of the items in the knapsack does not exceed the limit.
- **Objective Function**: The objective is to maximize the total value of the selected items. This is represented as an objective function that depends on the binary variables (representing items) and their corresponding values. The goal is to find a combination of items that maximizes the total value while staying within the knapsack capacity.

The vehicle routing problem as a CSP

- **Variables**: The variables represent the routes or sequences of visits made by each vehicle. Each variable corresponds to a specific vehicle and specifies the order in which customers or locations are visited by that vehicle.
- **Domains**: The domain for each variable consists of all possible permutations or sequences of customers or locations that the corresponding vehicle can visit. These permutations define the order in which the vehicle serves its customers or delivers goods.
- **Constraints**: The vehicle routing problem CSP includes several constraints, such as:
    - Capacity Constraint: Each vehicle has a limited capacity.
    - Visit Constraints: Each customer/location must be visited exactly once by one vehicle.
    - Time constraints: Each customer/location has specific time windows during which they can be visited.

    - Distance constraints: The total distance or time travelled by each vehicle should not exceed predefined limits.
- **Objective Function**: The objective is typically to minimize the total distance travelled by all vehicles or minimize the total travel time. This is represented as an objective function that depends on the binary variables (representing visits to customers) and the distances or times between customers and depots.

The treatment planning of critically ill covid-patients as a CSP

- **Variables**: Each patient and treatment option is represented as a variable, and the values represent the specific treatment or intervention chosen for the patient at each time step.
- **Domains**: The domain for each variable includes the available treatment options, dosages, and timing for each patient.
- **Constraints**: Constraints include factors such as:
    - Availability of medical equipment and personnel (e.g., ventilators, respiratory devices, trained staff).
    - Patient-specific medical conditions and responses to treatments.
    - Regulatory requirements and guidelines for medical interventions.
    - Limited resources, including the number of available ICU beds and staff capacity.
    - Uncertainty in patient conditions and potential complications.
- **Objective Function**: The objective function is to find an optimal treatment plan for each patient, considering the evolving medical conditions, available resources, and regulatory requirements.

The patient selection for XYZ syndrome study as CSP

- **Variables**: Each variable represents a patient's eligibility for the study. For example, X_A represents Patient A's eligibility, where X_A = 1 indicates eligibility, and X_A = 0 indicates ineligibility.
- **Domains**: The domain for each variable is binary: {0, 1}, representing ineligibility (0) or eligibility (1) for the XYZ Syndrome study.
- **Constraints**: The CSP includes constraints that define the specific inclusion criteria for the study, such as:
    - Age Constraint: Patients must be between the ages of 18 and 60. This is represented as a constraint on the age variable, e.g., ($18 \leq Age\_A \leq 60$) for Patient A.
    - Diagnosis Constraint: Patients must have a confirmed diagnosis of XYZ Syndrome, represented as a constraint that requires a patient's diagnosis (Diagnosis_A) to be "XYZ Syndrome."
    - Consent Constraint: Patients must have provided consent to participate in the study, represented as a constraint that requires a patient's consent status (Consent_A) to be "Given."
- **Objective Function (optional)**: The objective is to find a consistent assignment of eligibility variables that satisfies all constraints. Therefor an explicit objective function is not required, as long as all the constraints are listed and satisfied.

3.5 Search space, graph, and tree

Before we start with exploring and explaining existing AI algorithms to solve search problems, let's have a look again at the definition of a search problem:

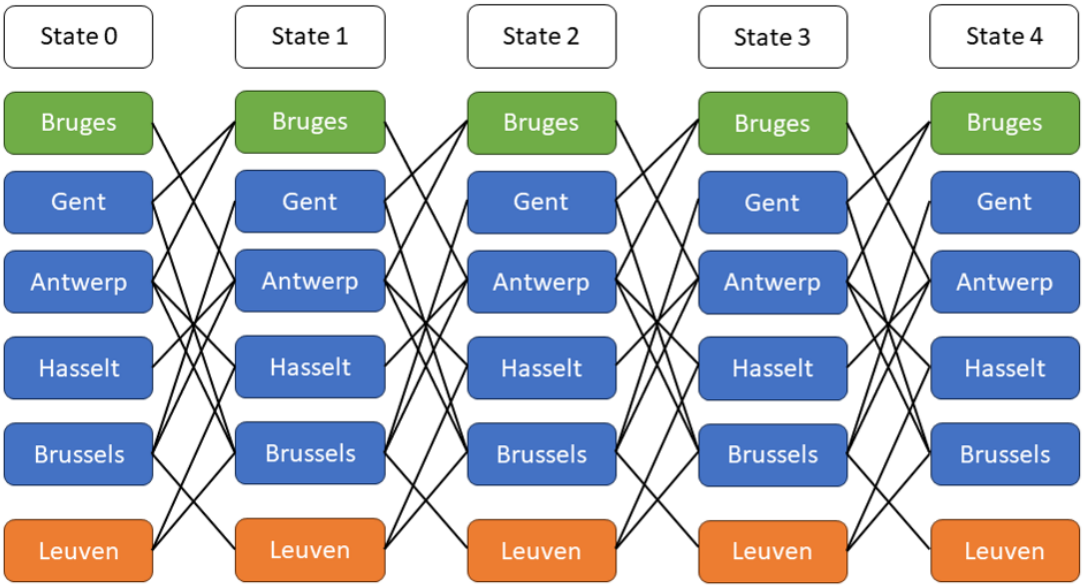
In AI, a **search problem** refers to a task or challenge where an AI-agent or computer program needs to find a solution by exploring a set of possible states or actions (search space) in a systematic manner (search algorithm). It consists of defining an initial state, a goal state, a set of actions or operators that can transition the system from one state to another, and a search algorithm to find a sequence of actions leading from the initial state to the goal state.

Search space

In the definition of a search problem, the **search space** is a fundamental concept intimately connected to the task at hand. It represents the complete set of potential states or configurations that can be investigated while attempting to find a solution. This encompasses all the available options or decisions that can be taken to reach the desired objective. The task of a search algorithm is to methodically explore and assess different states or solutions until an optimal or satisfactory one is identified. It is important to note that the size and structure of the search space has a significant impact on the complexity and efficiency of solving a given search problem.

Let's consider PharmaLog's challenge in Flanders and explore the search space and how its complexity evolves over different states:

- State 0 solutions: In this state, theoretically, any of our cities can serve as a starting point, provided there are no restrictions dictating that it must begin in Bruges. Hence, each city qualifies as a solution for this particular state.
- In State 1, considering each solution from State 0 and accounting for our Flanders map with the available roads, there emerge various potential State 1 solutions. These solutions manifest as pairs of interconnected cities.
- Considering the possible pairs of cities as State 1 solutions, and accounting for available roads, State 2 solutions comprise of all sets of three cities connected. This can be extended towards State 3,4,5... One can see that the amount of solutions from one state to another increases drastically.



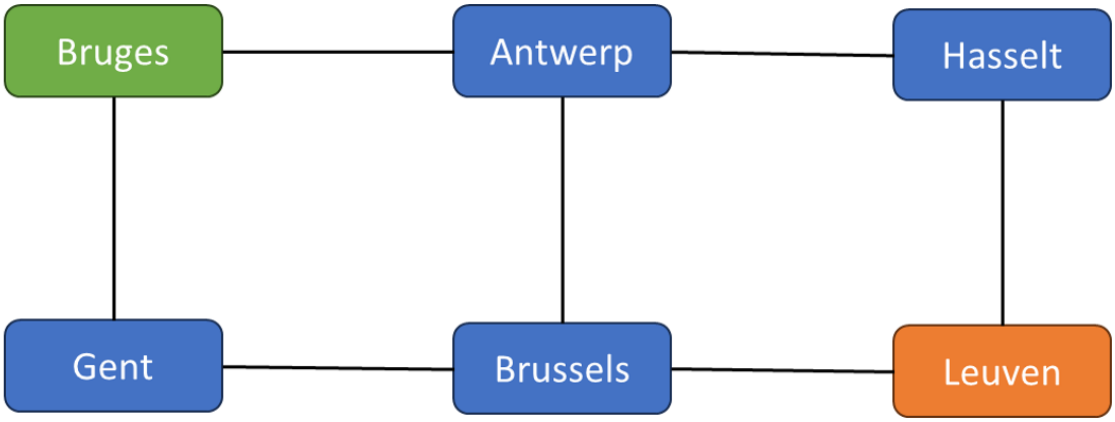
Even in this relatively simple search problem, it becomes evident that the **search space grows exponentially as the number of states or actions increases**.

Using the analogy of "all roads leading to Rome," it becomes apparent that there can be numerous solutions for going from Bruges to Leuven. When no additional information or constraints are provided, all of these solutions are considered equally viable. This highlights the vastness of the possibilities within the search space and underscores the need for effective search algorithms or strategies to navigate and find a suitable solution efficiently.

A good strategy is to define an efficient data representation of the search space, such as a search graph or search tree. Certainly not the one in the image carousel above, since you would probably agree that using this visualization makes it quickly become too complicated to find an effective route. Both search trees and search graphs are used to visualize and navigate through the search space of a problem, but they differ in whether revisiting states is allowed (search graph) or not (search tree).

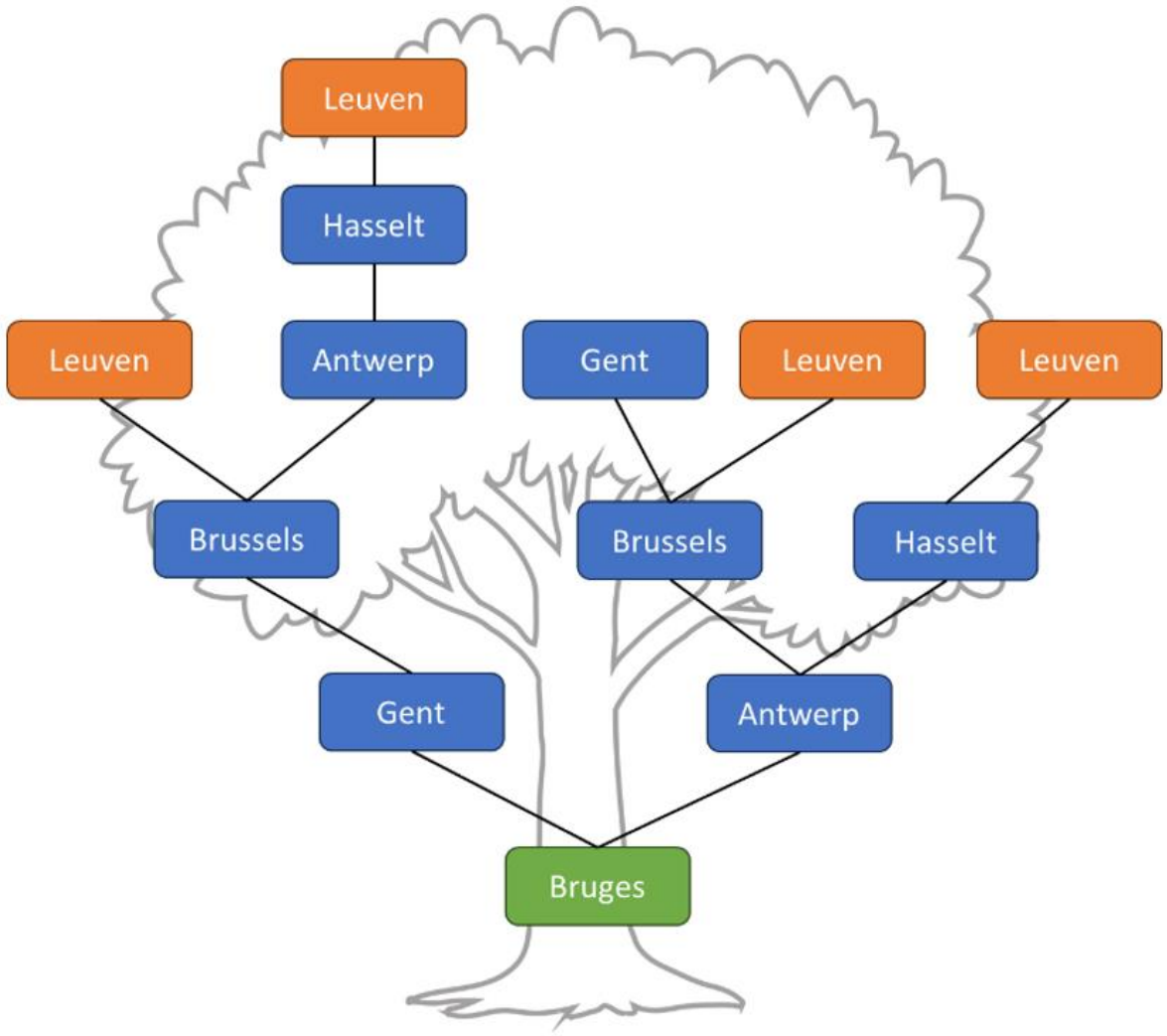
Search graph

Let's first look at the **search graph** representation of PharmaLog's challenge, which is a general representation of the search process where states or configurations are depicted as nodes, and the transitions or actions between these states are depicted as edges. They are particularly useful when dealing with problems where states can be revisited or when there are multiple paths to the same state.



Search tree

Now, let's explore PharmaLog's challenge using the concept of a **search tree**. A search tree provides a hierarchical visualization of the potential states or configurations during the search process. It begins with an initial state as its root node and extends into branches as different actions or decisions are made. Each node within the tree represents a distinct state, and the edges connecting these nodes symbolize the actions or transitions between states. Notably, in a search tree, a specific node is designated as the root node, and the revisitation of the same state is typically prohibited, resulting in a tree-like structure. This characteristic sets it apart from a search graph, making search trees less versatile but suitable for algorithms like depth-first search and breadth-first search (see next subsection on Search algorithms).



Both the search graph and search tree serve as valuable visual representations of the search problem. In the case of PharmaLog's relatively straightforward search problem, it becomes apparent to human observers that multiple solutions exist for traveling from Bruges to Leuven. The search tree, in particular, is well-suited for addressing the question of how to go from one specific node to another. Revisiting the questions from previously in this module:

Utilizing the search tree, we can easily pinpoint the various distinct options for reaching Leuven with the fewest number of actions or for visiting each city in parallel. This method of employing a search graph or tree serves to enhance not only human comprehension, but also the AI agent's capacity to efficiently seek solutions, a topic we will explore in more detail in the forthcoming section. **Given a search graph or search tree, it thus becomes easier to define different search strategies or algorithms.**

A **search algorithm** navigates through a graph or tree by following a set of rules. The choice of rules, or algorithms, influences how the search for solutions unfolds. Typically, all algorithms maintain a record of visited nodes and those yet to be explored (stored in memory). There are variations in the performance and outcomes of these algorithms. While some ensure finding a solution, others do not provide such guarantees. Certain algorithms not only promise a solution, but strive to find the best one based on an objective function. Speed and memory usage vary among algorithms; some are quick and memory-efficient, while others are slow or demand significant memory resources. Additionally, some algorithms may encounter issues like getting stuck in loops, repeatedly visiting the same nodes without reaching a solution.

Therefore, when evaluating a particular search algorithm or comparing different search algorithms, several key criteria can be considered to assess its performance and effectiveness in solving a problem. These criteria help determine how well the algorithm explores the search space/graph or tree and finds solutions. Here are some common **evaluation criteria for search algorithms**:

- **Completeness**: Does the algorithm guarantee finding a solution if one exists in the search space? A complete algorithm should find a solution whenever it exists.
- **Optimality**: Does the algorithm guarantee finding the optimal solution, i.e., the best possible solution among all alternatives? An optimal algorithm should find the best solution based on a defined objective or cost function.
- **Time Complexity**: What is the algorithm's time complexity? This measures how long it takes to find a solution. Ideally, it should have a reasonable time complexity for the problem at hand.
- **Space Complexity**: What is the algorithm's space complexity? This measures the amount of memory or storage required during the search process. A good algorithm should be memory-efficient.
- **Convergence**: Does the algorithm eventually terminate and return a solution, or does it risk running indefinitely without producing a result?

Now, let's move on to the next subsection to dive deeper into search algorithms.

3.6 Search algorithms

3.6.1 Search algorithms

In this new subsection of the module, we will delve into search algorithms. **Search algorithms** are fundamental techniques to solve search problems that involve exploring a vast space of possible solutions. These algorithms are designed to systematically traverse a tree-like or a graph-like structure that represents the potential states or configurations of a problem, ultimately leading to the discovery of a desirable solution.

There are two broad categories of search algorithms: uninformed and informed algorithms. The key distinction between these two lies in how they make decisions while navigating the search tree.

Uninformed algorithms, also known as blind search algorithms, operate without specific knowledge about the problem domain. They explore the search space methodically without considering the characteristics or qualities of the states they encounter.
Informed algorithms, also known as heuristic algorithms, leverage domain-specific knowledge or heuristics to make more informed decisions during the search. They evaluate the states in the search tree or graph based on heuristic functions that estimate the desirability of each state.

3.6.2 Uninformed algorithms: Depth-first search

Uninformed search algorithms are used when there is no additional information available to guide the search, e.g., in the Utopic version of Flanders. These algorithms systematically traverse the search space in a manner that is not influenced by domain-specific knowledge or heuristics. They typically employ a simple strategy, such as exploring all possible paths or nodes in a sequential manner, to find a solution. Common examples include Depth-First Search (DFS) and Breadth-First Search (BFS).

A **Depth-First Search (DFS)** explores a search-tree by traversing as deeply as possible along a branch before backtracking (going back in the direction of the root node). Here's a brief explanation of how DFS works:

1. **Start at the root node**: DFS begins at the root node (or starting point) of the tree or graph.
2. **Explore as deep as possible**: It explores a branch of the tree as deeply as it can, visiting the first unvisited node it encounters (from left-to-right). This means it goes as far down a branch as possible before backtracking.
3. **Backtrack when necessary**: When it reaches a dead-end or a node with no unvisited neighbours, DFS backtracks to the previous node and explores other unvisited branches.
4. **Repeat until the goal is found**: DFS continues this process, recursively exploring branches until it finds the goal node, then it stops.

Let's see how this applies to the following search-tree of PharmaLog's Challenge:

- Step 1: We start in the root node, which is the starting point at the Warehouse in Bruges.
- Step 2: Taking the left node, as the first node unvisited, we go as deep as we can into the tree, until we reach a death end. In this case all roads from Gent will take us to already visited cities.
- Step 3: We backtrack from Gent to Brussels
- Step 4: From Brussels we take the next unvisited node towards Leuven, and here the algorithm stops because we have reached our destination.

A DFS is implemented using a stack data structure to keep track of the nodes to visit. Therefore, it is **simple and memory-efficient** (you only need to store the nodes that need to be explored).

Stack Data Structure: A stack is a linear data structure that follows the Last-In, First-Out (LIFO) principle, where elements are added and removed from the top, allowing only the most recently added item to be accessed at any given time

3.6.3 Uninformed algorithms: Breadth-first search

An alternative search algorithm is the **Breadth-First search algorithm (BFS)**, which explores a tree data structure in a breadth first motion, i.e., it systematically explores all the nodes at the current depth level before moving on to the next level. Here's a brief description of how BFS works:

1. **Start at the root node**: BFS begins at the root node (or starting point) of the tree.
2. **Explore all neighbours at current level**: It explores all the neighbouring nodes at the current depth level before moving on to nodes at the next level. This ensures that it visits nodes closer to the root before going deeper into the structure.
3. **Maintain a queue**: BFS uses a queue data structure to keep track of the nodes to visit. Nodes are added to the queue as they are discovered and removed in a first-in, first-out (FIFO) manner, ensuring that nodes at the same level are processed in the order they were encountered.
4. **Continue until the goal is found**: BFS continues this process, level by level, until it finds the goal node.

Queue Data Structure: A queue is a linear data structure that follows the First-In, First-Out (FIFO) principle, where elements are added to the rear and removed from the front, ensuring that the oldest item is processed first.

Let's see how this applies to the following search-tree of PharmaLog's Challenge:

- Step1. The algorithm starts at the start node, which is the warehouse in Bruges.
- Step 2. On the next level, from left to right, first Antwerp is visited.
- Step 3. Then Gent is visited, before moving to the next level.
- Step 4. On the second level, again from left to right, we visit Brussels.

- Step 5. Then we visit Hasselt.
- Step 6. And again Brussels (via a different path).
- Step 7. Moving on to the third level, we first visit Gent.
- Step 8. And then finally we reach our goal node for the first time, so the algorithm can stop.

BFS guarantees that it will find the shortest path (minimum number of actions) to a goal node in an uninformed search tree, making it useful for an optimal solution in terms of shortest path. However, it can consume more memory compared to DFS, because it needs to maintain a queue with nodes at each level. In other words, for **extremely large search trees, this algorithm is often impractical.**

Many other (uninformed) search strategies and extensions exist. E.g., a **stochastic search**, is a group of search techniques that incorporates randomness or probabilistic elements into the search process. Instead of following a deterministic path, a stochastic search algorithm introduces randomness when making decisions about which nodes to explore in the search tree.

Another example is a **bidirectional search** algorithm, which is a group of techniques to simultaneously explore the search space from both the start and goal states. It starts with two separate searches, one from the initial state and one from the goal state, and these searches progress towards each other. The algorithm terminates when the two searches meet in the middle or when a solution is found, often resulting in more efficient searches for certain problems.

3.6.4 Informed algorithms

Next to uninformed search algorithms, informed search algorithms also exist.

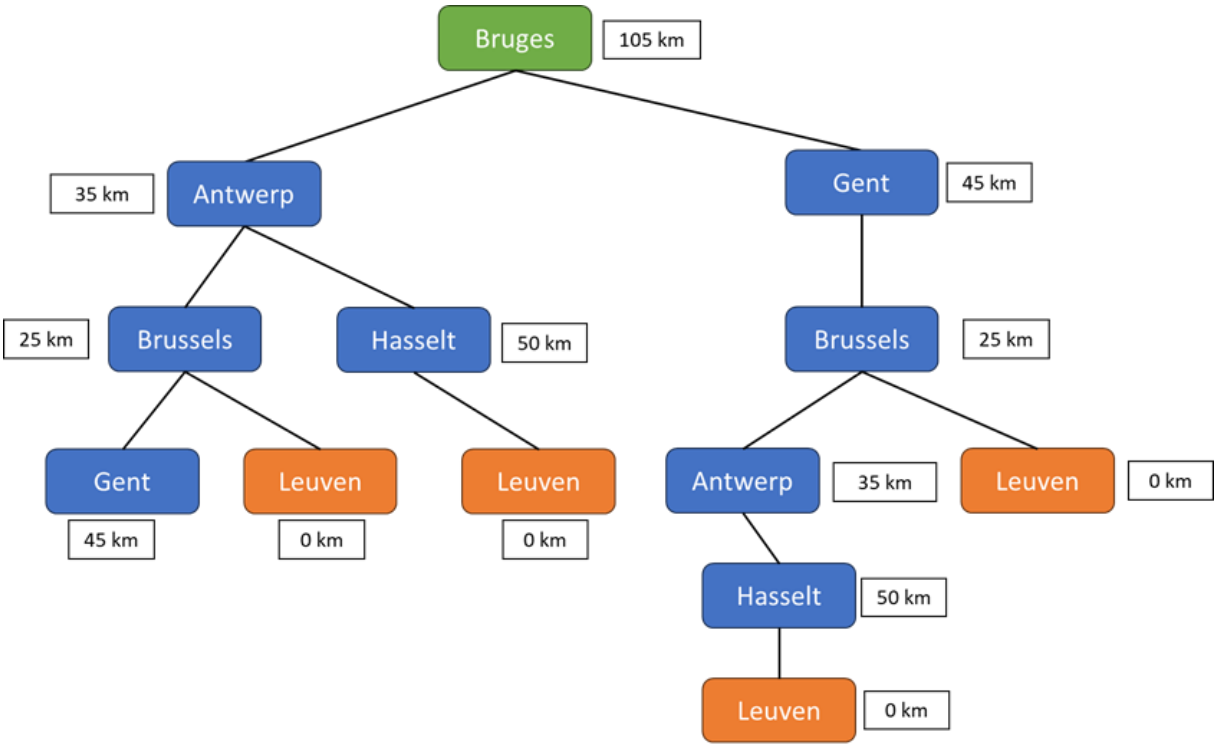
Informed search algorithms, known as heuristic search methods, prioritize paths that appear more promising, potentially resulting in faster convergence and enhanced solutions. To achieve this, they rely on problem-specific information, either provided or estimated at each state, to determine the most promising next state. In essence, a heuristic function is employed to express this problem-specific knowledge by quantifying the 'quality' of each state.

For instance, in PharmaLog's challenge, we can use the straight-line distance from each city to Leuven as the end-goal to gauge the quality of each city as an intermediate state for the S-DV.

Hill climbing search algorithm

The provided or estimated heuristic can assist in navigating the search tree, as seen in the application of the **hill climbing search** algorithm. This algorithm effectively combines elements of depth-first search with heuristic values. Notably, it prioritizes selecting the node with the most favourable heuristic value, rather than blindly advancing down the tree.

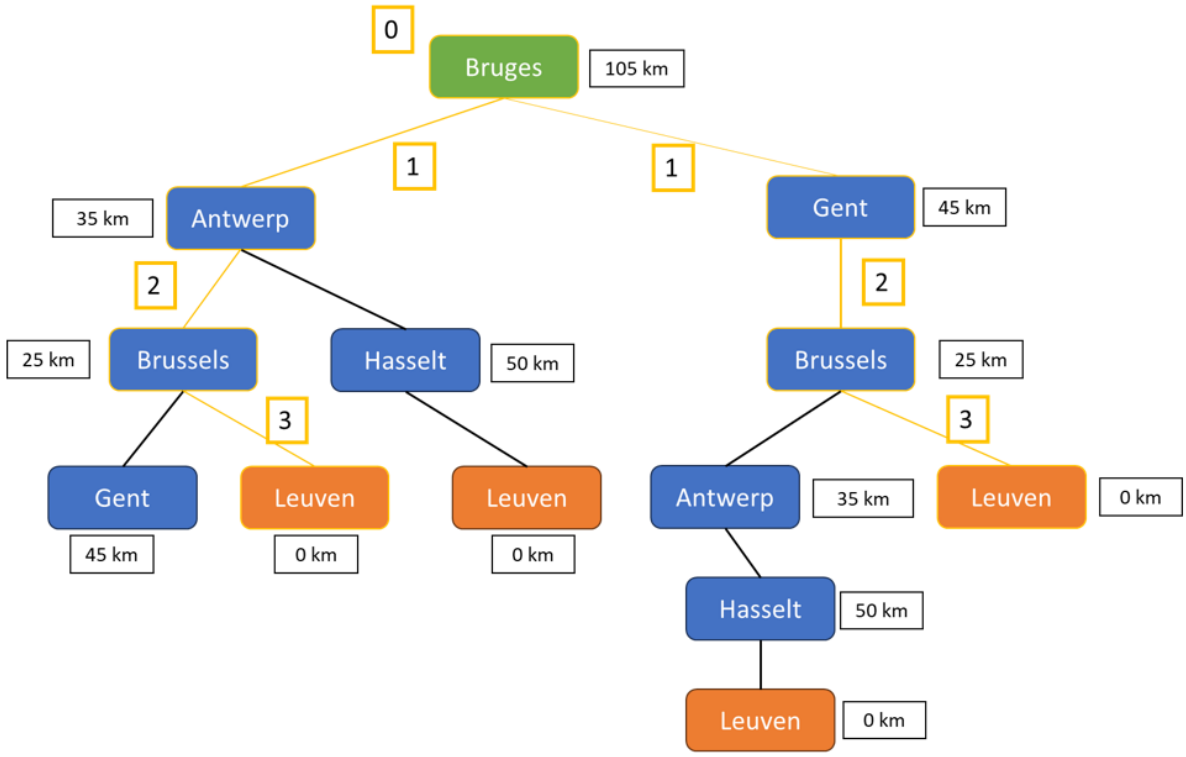
Given the heuristic-annotated search tree below, what path from Bruges to Leuven would be identified using this method? Reflect upon the answer.



Beam search algorithm

An alternative approach is the **beam search** algorithm, which constrains the breadth-first search by restricting the number of nodes explored at each level, referred to as the beamwidth. It prioritizes exploring a predetermined set of the most promising nodes based on the heuristic values at each new level as shown in the example here:

When the beam search algorithm restricts the number of selected nodes at each level to one (beamwidth = 1), it effectively transforms into the hill climbing algorithm without backtracking. This also illustrates a **local search** approach, where only a single path is retained and stored in memory. We will explore local search algorithms in more detail later.

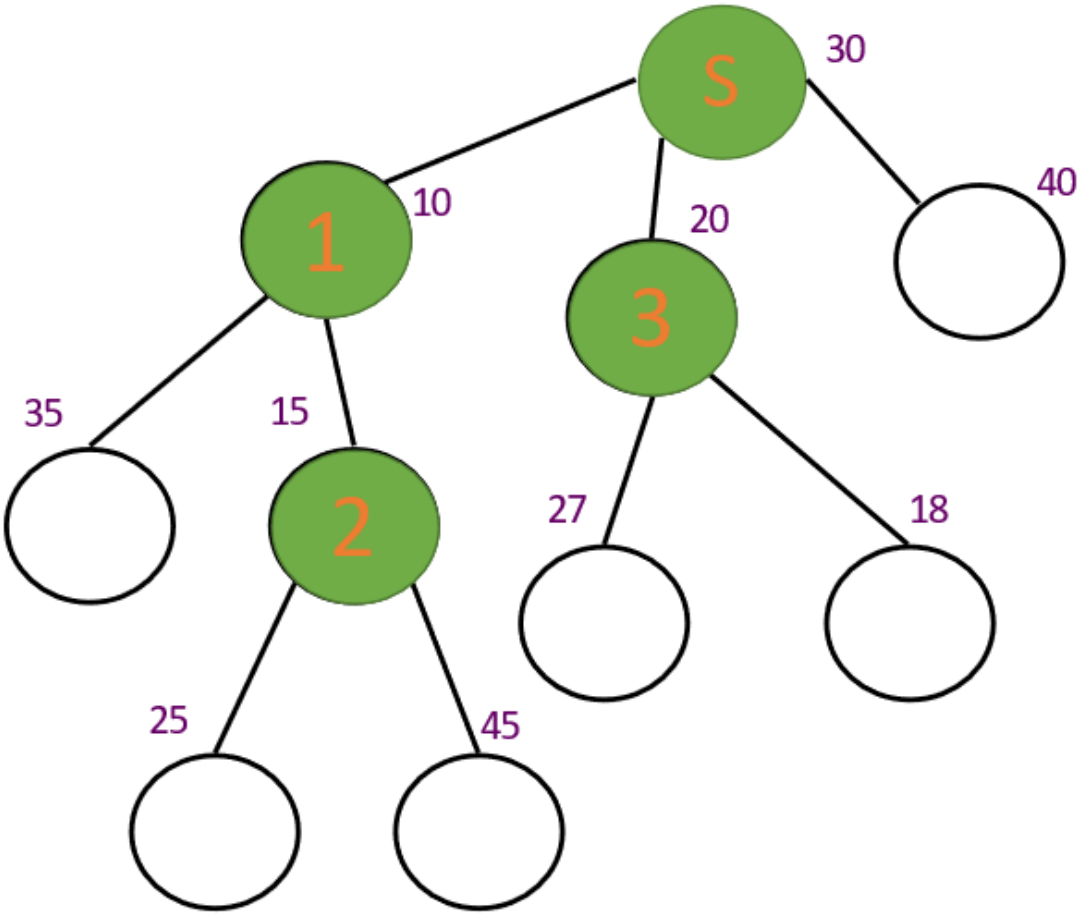


Greedy search algorithm

Another commonly employed informed search algorithm is the **greedy or heuristic best-first search**. Like hill climbing and beam search, it relies on a heuristic to determine the next best move. However, unlike them, it does not strictly follow either depth-first or breadth-first patterns; it can evolve in all directions. Throughout the search, all visited nodes are monitored, and at each step, the node with the most favourable heuristic value is selected, this can be a node deeper down in the tree or a node closer to the root node but moving more horizontally in the tree.

For the relatively simple PharmaLog Challenge, the resulting path would be the same as the hill climbing path (not the search strategy) starting at Bruges and going to Leuven, passing Antwerp and Brussels first:

- We start in the root node. Look at its children nodes and select the node with the best heuristic to expand to go to the next situation. The lowest value (10) is the best heuristic, hence this node will be explored first.
- The next best heuristic, from all heuristics visible and unvisited is 15. So we expand that node next.
- Similar to the steps before, the best unvisited heuristic now is 20.
- This node is not positioned deeper within the tree, but to the right and closer to the root node. The greedy search algorithm exhibits less rigidity in its choice of direction, as it can advance vertically and horizontally in subsequent steps, without being strictly bound to depth-first or breadth-first approach.

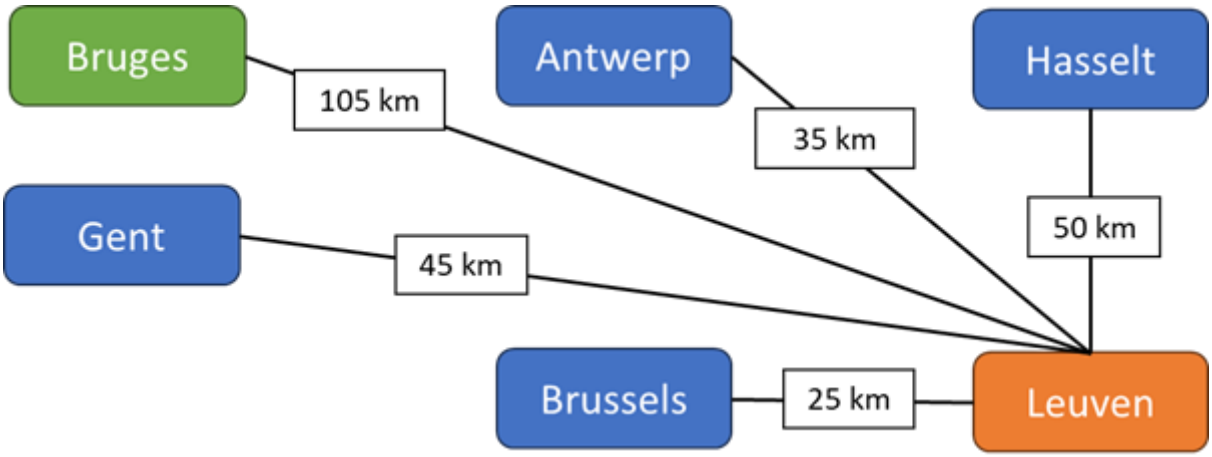


3.7 Optimal search algorithms

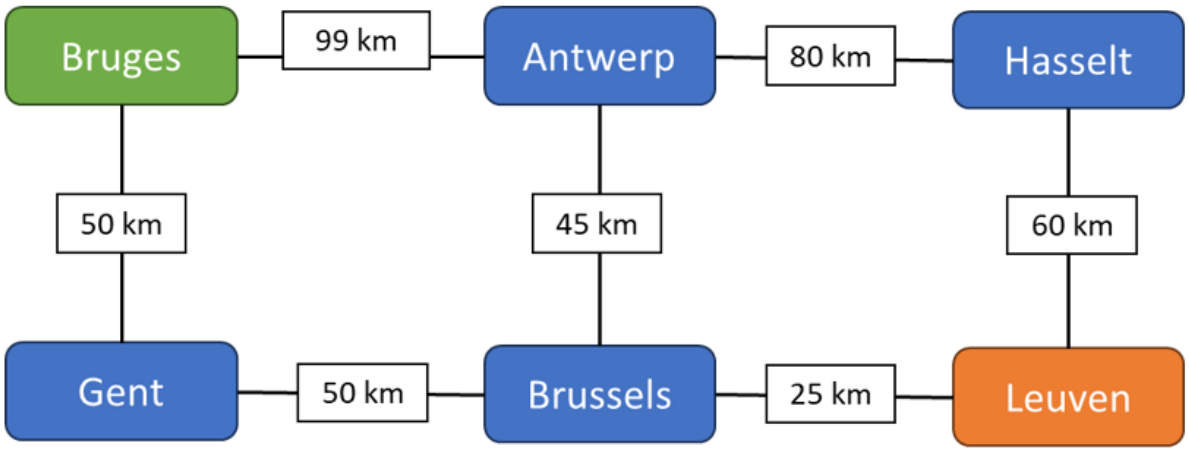
Until now, we have been talking about search algorithms, which are tools for finding good solutions efficiently. But now, let's focus on **optimal search algorithms**. These are techniques specifically designed to find the best solution among several options by adjusting a particular goal, like minimizing costs or maximizing benefits.

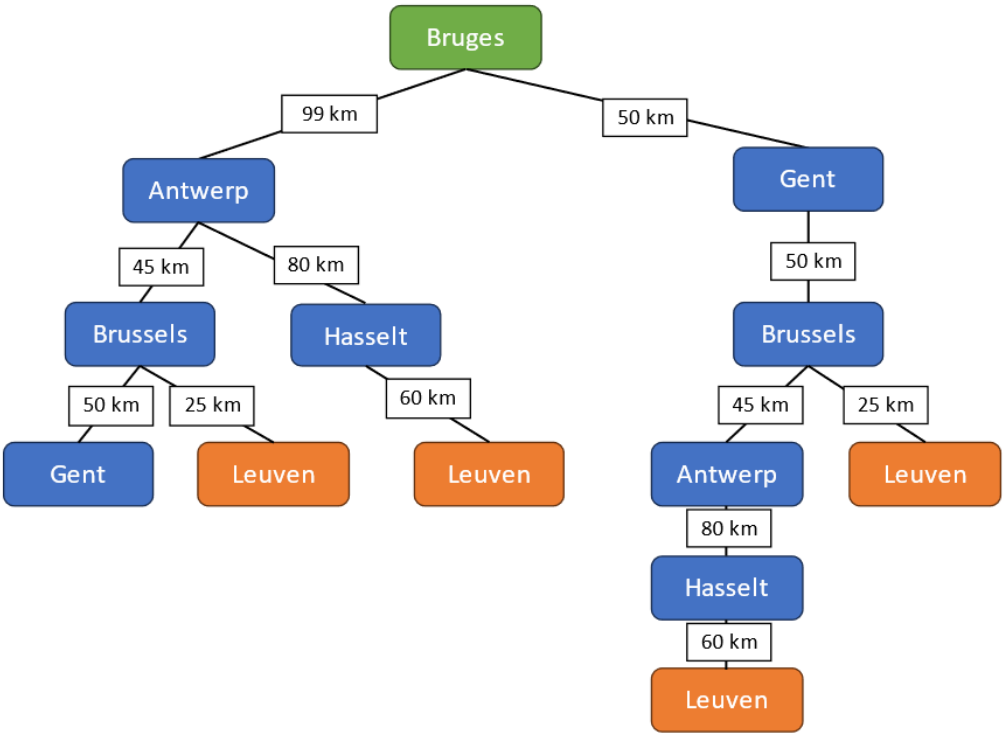
Interestingly, optimal search algorithms use the basic principles of search algorithms to explore the available options. However, what makes optimization special is that it goes beyond just exploring; it carefully examines additional information that is either estimated, or given to improve the quality of the final solution. In short, **optimization algorithms are built not just to find a solution, but to find the very best one considering the problem's constraints and goals.**

Both informed and optimal search methods rely on supplementary data, but they differ in the nature of this information. In an informed search, this supplementary data often takes the form of heuristics or estimations. For instance, in the PharmaLog Challenge, it involved estimating the straight-line distance between cities in Flanders and Leuven:



In contrast, an **optimal search** utilizes provided information to evaluate solutions and to determine the best possible solution. For instance, in the PharmaLog Challenge, this involves using the actual road distances between each pair of cities:





Minimal cost search algorithm

Given these distances, a typical objective to optimize is finding the shortest path in terms of total distance travelled (in contrast to number of states or nodes transitioned). As a first attempt, let's use the **minimal cost search algorithm** given the current state or node. The next node is determined by adding the node with the smallest cost from the current node. In other words, it does not look back or forward, only to the roads connected to the current node and their distance. Each time it will take the shortest road.

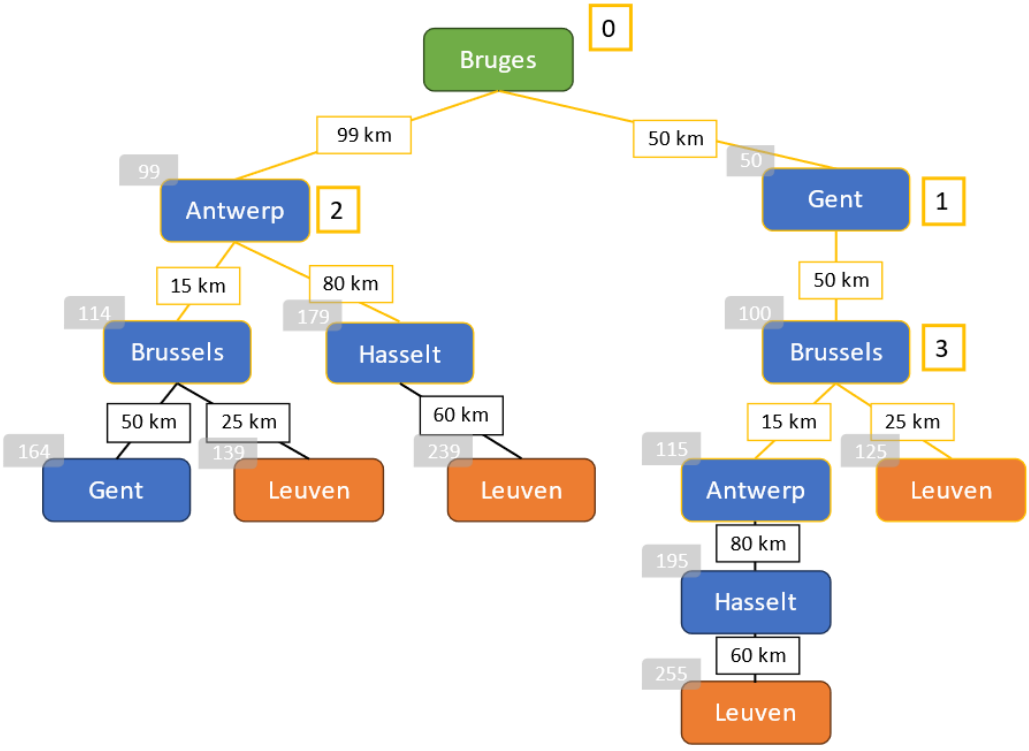
Given the search tree above, **what is the path** identified with this algorithm? Hint: its behaviour of exploring the tree is similar to the hill climbing algorithm.

Uniform cost search algorithm

A notable and practical adaptation of the minimal cost search is the **uniform cost search**. This algorithm, at each step, chooses to expand the node with the lowest accumulated cost. In essence, it

considers not only the distance of the upcoming path, but rather the distance already covered to the current node when making its selection. The application of this search algorithm to PharmaLog's challenge is shown here:

- Before we start, for each node the accumulated distance is computed from the previous node, shown in the grey boxes.
- Starting from Bruges, we explore the two nodes connected to it (children nodes).
- The accumulated distance in Gent is better than the accumulated distance in Antwerp from Bruges. Therefore, the node of Gent is explored next towards Brussels.
- Now the accumulated distance to Antwerp is the smallest compared to the accumulated distance to Brussels from Gent, and therefore we next explore the path from Antwerp towards its children nodes, Brussels and Hasselt.
- In the third step, the roads from Brussels to Antwerp and from Brussels to Leuven are explored, and in doing so the algorithm terminates, because we have reached our destination in Leuven.



A algorithm*

It is evident that while it represents an enhancement over the minimal cost algorithm, there is still room for further improvement. One notable advancement is the widely recognized **A* algorithm**, which essentially extends the uniform cost algorithm by incorporating three crucial enhancements:

1. The branch and bound strategy
2. The path deletion strategy
3. The inclusion of heuristic underestimates

These help streamline the search process and optimize the use of resources in algorithms like A* by focusing on the most promising paths.

Branch and bound strategy

The branch and bound strategy is a technique used in optimal search algorithms to efficiently explore and prune branches of the search tree. It involves breaking down the problem into smaller subproblems (branching) and using a bound or estimation to determine if a particular branch can lead to a better solution than the current best solution found so far. If a branch is deemed incapable of improving upon the current best solution, it is pruned, reducing the search space and improving efficiency.

Path deletion strategy

The path deletion strategy is a technique that identifies instances where the same node is reached using multiple paths, akin to the saying "all roads lead to Rome." In such cases, it selects and retains only the path to that node with the lowest accumulated cost, discarding the others.

Heuristic underestimates

Heuristic underestimates involves using a function that provides a lower-bound estimate of the cost from the current node to the goal. This heuristic guides the search by prioritizing nodes that are expected to lead to better solutions, effectively reducing the search space. A* combines the actual cost to reach a node with this heuristic estimate to make informed decisions, always selecting nodes that seem most promising in terms of reaching the goal efficiently. When the heuristic is admissible (never overestimates the true cost), A* is guaranteed to find an optimal solution.

3.8 Local search algorithms

In numerous optimal search problems, the specific path taken holds no significance; only the (optimal) attainment of the goal state matters. In these scenarios, iterative improvement algorithms prove valuable. These algorithms maintain a single "current" state and continually (over multiple iterations) attempt to enhance it. This process of **iteratively refining a single state is known as a local search**. Remember that in the unit on informed search, we've already encountered two examples of local search strategies: the beam search algorithm with a beamwidth of 1 and the hill climbing algorithm without backtracking.

Local search algorithms involve **probing the vicinity of a given solution**, aiming to transition to a neighbouring solution that enhances the heuristic or objective function, depending on whether it's an informed or optimal search, respectively. In contrast to global search algorithms, local search operates with limited knowledge of the entire search space. Think of it like a person hiking in misty mountains – they can't see the complete landscape but can locally assess what's uphill and what's downhill.

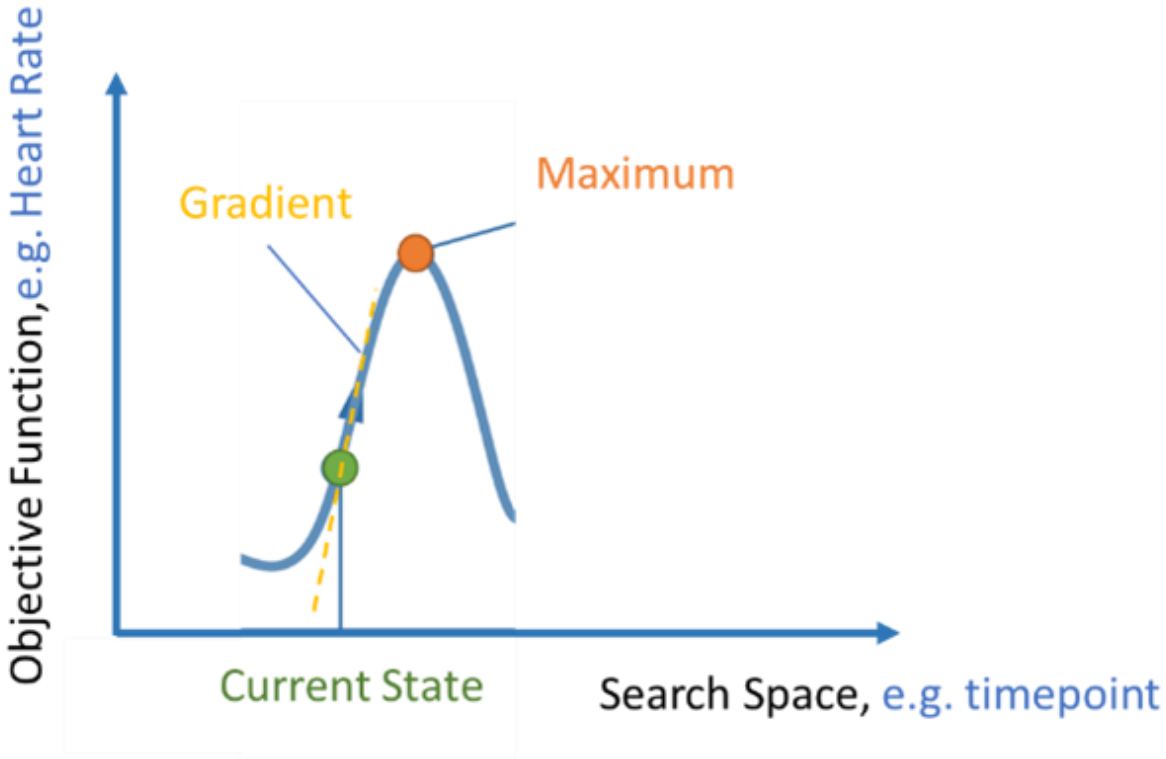
Local search algorithms have a **constant space complexity**, which means they maintain a consistent memory footprint throughout their execution. In other words, they eliminate the need to map out the complete search space, graph or search tree, which offers several advantages. In terms of efficiency, local search algorithms frequently surpass global search algorithms in computational time and memory utilization. They concentrate their efforts on a limited portion of the search space, making them especially suitable for tackling extensive or intricate search problems. In regard to scalability, local search can adeptly address problems with high-dimensional search spaces, where examining the entire space would be unmanageable. In fact, many **machine and deep learning** algorithms, which we will discover in the next modules depend on local searches!

Managing discrete and continuous search spaces

- In cases where search problems involve **discrete** search spaces, such as the vehicle routing, knapsack, and nurse scheduling problems, the go-to local search algorithm is the **hill climbing** algorithm.

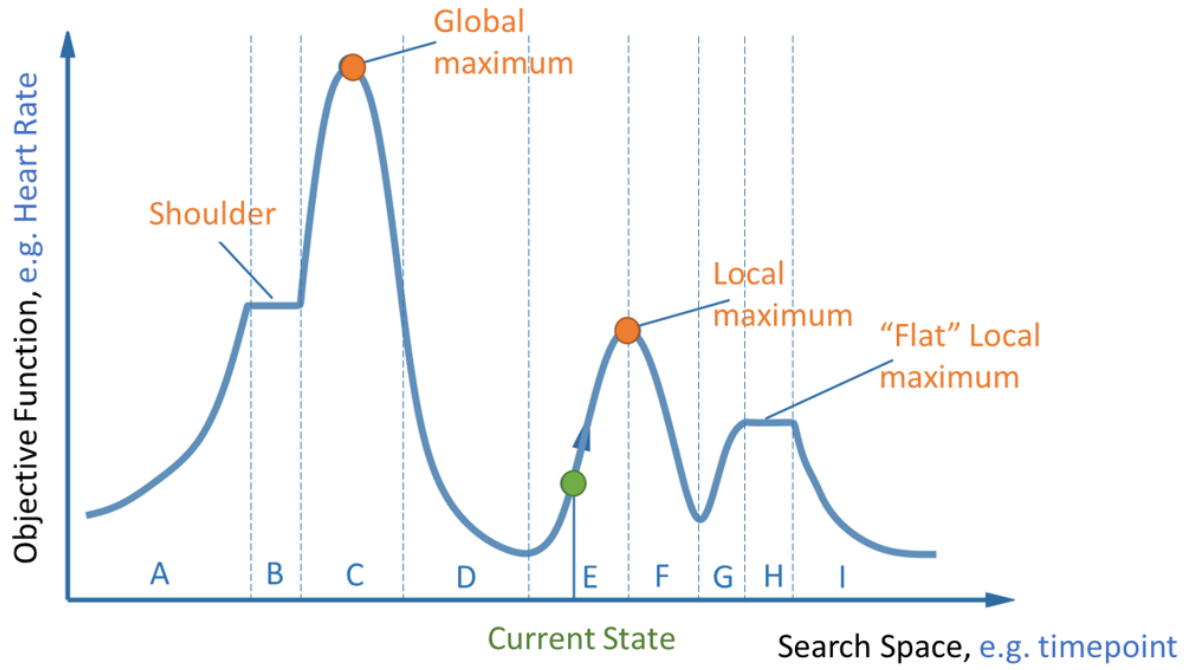
- When dealing with **continuous** search spaces (e.g., extracting the maximum heart rate from continuous heartrate monitoring) the go-to local search algorithm is the **gradient descent** (or ascent) algorithm. This technique relies on assessing the gradient of the objective function at the current state to determine the most promising direction for enhancing the current state.

Example: Assume we are tracking a person's heartrate using a wearable fitness and health tracking device, over a short period of time, leading to the hill-shaped "**optimization landscape**. The goal is to identify the highest heart rate within the total signal measured. The horizontal axis signifies the search space, encompassing all time points from the signal's beginning to its end (where we stopped the measurements). In this specific instance, it constitutes a one-dimensional space, with the current state or time point (green point) indicated on this axis. The vertical axis represents the recorded heart rate or, in our case, the value of the objective function for different states. With the gradient descent algorithm, the current state is adjusted by progressing in the direction of the current state's objective function gradient (yellow dashed line). In our pursuit of the maximum heart rate, this entails moving the current state upwards along the time point axis. If our objective were to find the minimum heart rate, the movement would occur in the opposite, descending direction. In summary, the gradient determines the direction of change. What is to be fine-tuned, is how much or how big of a step the current state is moved along this direction.



Local versus global search

It is essential to recognize that global search algorithms, such as depth-first and breadth-first search, possess their own advantages. They excel in guaranteeing the discovery of global optima and providing a comprehensive exploration of the entire search space. To understand the main limitation of a local versus global search strategy consider the following optimization landscape (e.g., after tracking the heart rate for a full day, instead of a brief period of time):



Challenges for local search

- **Local maximum**: While the local maximum may serve as a reasonably good solution to the optimization problem at hand, it should be noted that it is not equivalent to the global, and thus superior, optimal solution. The local maximum can mislead the local search algorithm into presuming it has found the optimal solution, as, when exploring only nearby states, the current state when situated in the local maximum appears to be the best option. Remember, when the current state is too far away from the global maximum, it cannot see or is blind to the existence of the global maximum.
- **Plateaus**: Another perplexing scenario for a local search algorithm arises when plateaus (e.g., the shoulder and the flat local maximum) exist in the objective function. In such cases, the current state and its nearby states all yield identical objective values and are equally valid solutions to the optimization problem. During this phase, the algorithm lacks the information or direction needed to determine the next best step.

Solutions for local search

In both the real world and the realm of healthcare, many optimization challenges exhibit multiple local maxima and plateaus. This characteristic implies that, despite the operational efficiency of local search algorithms, their utility is inherently limited. Nevertheless, AI developers have demonstrated remarkable creativity, devising numerous solutions over the past decades. Among these solutions, we can highlight three notable examples: the utilization of **simulated annealing, local beam searches, and genetic algorithms**. You can delve deeper into these approaches in the different tabs below:

Simulated Annealing

Simulated annealing is a technique inspired by the **annealing process** in metallurgy in combination with **random walk optimization**. It starts with an initial solution and iteratively explores neighbouring solutions while randomly allowing for occasional moves to worse solutions. These random movements are akin to a random walk optimization, which is a simple and basic optimization technique where the current state is updated by making random steps and therefore not following the gradient in the search space to explore potential solutions. This stochastic approach helps escape local optima and find better

solutions, making it a valuable tool in solving optimization problems with complex and irregular landscapes. However, with randomness alone, it is hard to finally converge to a final solution. Therefore, the key idea in simulated annealing, is to simulate the annealing of a material, so that the system explores the solution space with higher randomness initially (akin to heating) and gradually reduces the randomness over time (cooling) to converge towards an optimal solution following the gradient information.

Local Beam Search

A local beam search algorithm follows a similar concept to the global beam search algorithm we previously discussed, which involves **tracking a predefined number of paths**. In essence, it entails the use of multiple copies (e.g., **K-beam search** with K copies) of a local search algorithm, each initialized randomly. During each iteration, exploration of all successor states from the K current states occurs. Subsequently, the top K among these successor states are chosen as the new current states. It's important to note that this differs from running K local searches in parallel because communication occurs between the various searches as they compare all successor states, akin to saying, "Come over here, the grass is greener!"

Genetic Algorithms

Genetic algorithms are based on the principles of evolution, particularly genetic inheritance, and natural selection. Genetic algorithms track many states, often numbering more than 100 or even 1000, within a population. Each state in this population represents a potential solution to an optimization problem, whether constrained or unconstrained. The fundamental concept behind genetic algorithms is to evolve this population into new generations, with the aim of creating better solutions by combining the states within the current population. This process unfolds as follows:

- **Initialization**: Begin by creating an initial population of solutions, typically by randomly selecting, for instance, 100 states.
- **Fitness Evaluation**: Each of these states is tested as a potential solution and assigned a fitness score, indicating how good the solution to our problem really is.
- **Selection**: Following the principle of “survival of the fittest”, all states are ranked based on their fitness scores, and the fittest states are chosen as parents to generate new (offspring) states.
- **Reproduction**: Offspring states are created using a process called crossover, where the offspring state inherits values from both parent states for different variables in the optimization problem.
- **Mutation**: Additionally, to introduce novelty and diversity, individual values of variables can be changed at random (mutated). This mutation process enhances the diversity within the population and enables it to explore various parts of the optimization landscape.
- **New Generation**: The offspring states collectively form the new set of solutions and become the current population for the cycle to repeat itself.

Similar to the strategy used in simulated annealing, the mutation rate, which introduces an element of randomness, can be regulated over subsequent iterations of the algorithm. This adaptation of randomness ensures a balanced exploration-exploitation trade-off in the search for optimal solutions.

3.8.2 Outro

Search problems are ubiquitous in various application domains, including healthcare, as highlighted in this module. The introduction of AI solutions to these challenges marked a significant turning point, generating considerable excitement in the 1990s following the historic victory of a computer over a

human in a game of chess. The proliferation of search algorithms discussed in this module has played a pivotal role in automating numerous fields such as logistics, operations, and planning, where efficient resource allocation and decision-making are paramount.

In the context of AI agents, as explored in Module 2, search algorithms typically function as goal-driven agents, striving to plan and select actions to attain or optimize specific objectives. Despite their efficiency, they face certain limitations, particularly when confronted with dynamic changes in the environment or limited knowledge about the problem at hand. Dynamic alterations can impede the convergence of search algorithms, while a lack of comprehensive knowledge about the problem can hinder the accurate definition of variables, domains, and constraints. This, in turn, may prevent the execution of an optimal search or result in suboptimal solutions in practical applications.

Particularly in cases where the problem lacks a well-defined structure, it becomes pertinent to explore more sophisticated forms of AI, such as Learning Agents, which possess the capacity for learning, as we will delve into in the upcoming module.

4 Inside the AI engine: Learning from data

4.1 Welcome to Module 4

Welcome to the "Inside the AI engine" module about learning from data, where we delve into a journey to understand how cutting-edge AI technologies are revolutionizing the way we approach healthcare and the way the "engine of AI" really works.

Key Focus Areas

- **Supervised and unsupervised learning**: We'll start by demystifying the core concepts of supervised and unsupervised learning. Discover how these learning paradigms underpin many AI-driven solutions in healthcare, enabling us to make predictions, classify patients, and uncover hidden patterns in medical data.
- **Classification, regression, and clustering**: Dive deep into the diverse methods used for classification, regression, and clustering in healthcare. Explore real-world examples of how these techniques are utilized to identify diseases, predict patient outcomes, and group similar medical cases.
- **Model evaluation and selection metrics**: In AI for healthcare, model performance is paramount. Learn about the essential metrics used to assess the accuracy, precision, recall, and F1-score of machine learning models. Discover how these metrics guide the selection of the most suitable models for specific healthcare tasks.
- **Reinforcement learning and semi-supervised learning**: As we move forward, we'll explore advanced AI techniques like reinforcement learning and semi-supervised learning. Understand how these methods are being leveraged to optimize treatment plans, automate medical procedures, and make the most of a limited amount of labelled data.

Why This Module Matters

Healthcare is an area where AI's potential to make a difference is undeniable. Whether it's diagnosing diseases earlier, personalizing treatment plans, or streamlining administrative processes, AI has the power to transform healthcare delivery. By the end of this module, you will not only grasp the core concepts of machine learning, but also appreciate their vital role in shaping the future of healthcare.

Learning goals

- Learn and understand what supervised and unsupervised learning is and what their differences are. Distinguish between relevant examples of these two types of learning.
- Within the paradigm of supervised learning, describe the type of input data required and output data expected, and make the distinction between classification and regression. For both of these tasks, acquire basic knowledge about standard algorithms and restate how to evaluate these.
- Within the paradigm of unsupervised learning, describe the type of input data required and output data expected for the concepts of clustering, manifold learning, latent component analysis, novelty and outlier detection, covariance and density estimation. Explain the purpose of these concepts using application examples and standard algorithms.
- Learn how to select an algorithm to be used both for supervised and unsupervised learning, and evaluate the expected outcome.
- Gain knowledge about weak supervision and discriminative vs generative modelling in the context of supervised and unsupervised learning.

- Understand the importance of proper data management in machine learning, and restate factors influencing algorithmic choice.
- Learn to identify a machine learning problem in given or self-proposed applications, and identify an appropriate machine learning paradigm and algorithm to solve it.

4.2 Unsupervised learning

4.2.1 Clusters and unsupervised learning

As you may have noticed, there are different options for placing the medication on the shelves. You and your peers have most often proposed one of the following options for the division:

- Pills vs other
- Smaller dose medication vs bigger size
- Grey-scale medication vs colourful
- A split into 2 groups based on alphabetical order
- Painkillers vs other
- QD ("quaque die" - once per day) vs other

And there are even more possibilities. Since no other criterion was provided, all the categorisations above could have achieved the purpose. The categorisation of the medication into several groups or **clusters** without the provision of any training data, or any other input on how to create those clusters, is an example of **unsupervised learning** in AI.

Unsupervised learning is a type of machine learning where the algorithm learns patterns from input data without explicit supervision or labelled outputs. Instead, it seeks to find inherent structures or relationships within the data, such as clusters or associations, to uncover insights or make predictions.

Unsupervised classification is a type of AI algorithm used to group similar data points or objects based on their intrinsic properties and similarities, **without the need for pre-existing labels or categories**. It can be used to explore available data in the absence of instructions about the optimal categorisation. The algorithm identifies patterns and similarities in the data, and **clusters similar data points together based on their attributes or features**. Unsupervised classification is selected when we have **limited amounts or absence of available data**.

Defining features

In the exercise of sorting medication, you experienced that the first step is to define the features for the available data. The **features** are the characteristics or properties that define the data points, such as the form of the medication (e.g., pill, liquid, etc.), the size, the use of the medication or even the colour. In unsupervised classification, the choice and selection of features is critical, since it will have a big influence on how the algorithm will cluster the data points as it mainly relies on similarity. In the exercise, the selected feature in one of the categorisations was the form of medication (pills vs other), which is a categorical feature.

- **Categorical** features represent discrete, distinct categories or labels, such as colours or types of animals. They do not have a natural order or mathematical meaning.
- **Numerical** (continuous) features, on the other hand, are numeric values that can take any real number within a range and often represent measurements like age or temperature. They have a meaningful order and can be subjected to mathematical operations.

Clustering

In the categorisation of grey-scale vs colourful medication, the features are defined as the colours of the medication. This means that we represent medication solely based on its colour. Each medication will then have a unique correspondence to a point in the **three-dimensional colour space**, where the x, y, and z coordinates correspond to the amounts of red, green, and blue (RGB) in each medication's colour. For example, a medication with colour (128, 64, 192) would be represented as a point (128, 64, 192) in the colour space. So indirectly you selected a data space with higher dimensionality.

In the medical logistics exercise, you did not only define the characteristics for the data, but you also provided a **separation criterion** that you used for grouping medication.

Separation criterion is a measure used to evaluate the degree of distinction or segregation between different groups or clusters within a dataset. It quantifies how well the data points within each group are separated from each other and how distinct the groups are from one another.

Defining such separation can be done formally with an AI algorithm; here with unsupervised clustering. The **unsupervised clustering algorithm** can group the medication based on their similarities in the RGB space for instance. The algorithm will identify clusters of medication with similar colour properties and group them together. In general, the algorithm will aim to group the data based on some optimality criterion. The final grouping will be based on the distribution of the data and will define the boundary between the groups. Numerically, optimality will be based on the definition of a metric that quantifies the distance between different data points and identifies which data points are "close to each other" in the data representation space and which ones are "far from each other". Close examples will end up in the same cluster.

Assigning labels

Finally, we were able to visually inspect the resulting clusters and **assign labels** to them based on our interpretation of the data. For example, we were able to label the clusters as red, blue, green, yellow, etc., or assign more descriptive labels, such as grey-scale or colourful, etc. In the use case of medication, classification labels could be given such as painkiller or other purpose medication, or QD ("quaque die" - once per day) and medication with higher number of dosages per day.

In conclusion, unsupervised classification is a powerful tool for **data exploration**. It allows to group similar data points based on their intrinsic properties and similarities, without the need for pre-existing labels or categories. The choice and selection of features and the distance metric are critical for the algorithm's performance, and the resulting clusters can provide insights into the underlying structure and patterns of the data. Especially in healthcare, this enables us to make predictions, classify patients, and uncover hidden patterns in medical data.

Note that we specified the number of clusters to be two. However, a similar reasoning would still apply when more shelves would become available. For example, with four shelves, one could aim to group the available data in four groups. The number of shelves in this example is considered to be a hyperparameter. The definition and importance of hyperparameters will be discussed later on in the current module.

Let's continue to the next paragraph to learn about the popular unsupervised learning algorithm called **k-means clustering**, used for partitioning a dataset into a predefined number of clusters.

4.2.2 K-means clustering

Now that you learned about unsupervised learning, we are introducing **k-means clustering**, a popular unsupervised learning algorithm used for partitioning a dataset into a predefined number of clusters. The k-means algorithm is a powerful tool for clustering data into distinct groups based on similarity, and it is widely used in various **applications in healthcare**, including patient segmentation, disease subtyping, and healthcare recourse allocation. Overall, k-means clustering can provide valuable insights and facilitate decision-making processes in healthcare, leading to more effective patient management, resource allocation, and research advancements.

K-means clustering is an unsupervised clustering algorithm that partitions a dataset into k clusters. K is a pre-defined parameter chosen by the user (the hyperparameter) and represents the number of separate groups into which the data will be split. The algorithm aims to group similar data points together and minimize the variance within each cluster. It can be used to identify groups in the data without prior labels.

K-means relies on the idea that a cluster can be well characterized by a mean value of all datapoints in the cluster in the feature space. These mean values are usually called **centroids (or seeds)** and serve as the prototype of their cluster. At every iteration, the algorithm assigns each data point to the nearest cluster centroid, which is the most similar prototype, and afterwards recalculates the centroid of each updated cluster based on the new assignments. This process continues until convergence is achieved, i.e., the assignments and centroids no longer change significantly.

We structure the k-means algorithm into three main steps: initialization, optimization, and convergence/result.

1. Initialization
    1. Choose the number of clusters (k) you want to create.
    2. Initialize k cluster centroids (also called seeds) randomly. These centroids are typically chosen from the data points themselves or based on some heuristic.
    3. Assign each data point to the nearest centroid. This assignment is based on a distance metric, commonly Euclidean distance, where each point belongs to the cluster represented by its closest centroid.
2. Optimization
    1. Compute the mean of all data points assigned to each centroid. This mean becomes the new centroid for that cluster.
    2. Repeat the assignment and centroid update steps iteratively until one of the convergence criteria is met, such as a maximum number of iterations or when the centroids no longer change significantly between iterations.
    3. The optimization step aims to minimize the total within-cluster variance, which is typically calculated as the sum of squared distances between data points and their assigned cluster centroids. This process tries to find the best cluster centroids that minimize the variance within each cluster.
3. Convergence/Result
    1. When the algorithm converges (i.e., the centroids no longer change significantly or the maximum number of iterations is reached), it produces a final clustering result.
    2. Each data point is assigned to one of the k clusters based on the nearest centroid.
    3. The final cluster centroids represent the centre of each cluster.
    4. The algorithm outputs the cluster assignments for each data point and the final centroids.

It is important to note that k-means is sensitive to the initial placement of centroids, and **different initializations can lead to different results**. To mitigate this, researchers often run the algorithm multiple times with different initializations and select the best result based on a chosen evaluation metric (more about evaluation metrics will follow later in this module).

As no labels are used to cluster the data, no direct interpretation can be given to the different clusters. However, you can aim to qualitatively interpret the meaning of every cluster by examining the feature values of the centroids (central point of the class). With expert domain knowledge, **labels can be assigned based on subjective interpretation of the data**. For example, we can label the clusters as red, yellow, grey, or assign even more descriptive labels, such as pastel, bright, dark, etc.

To apply the k-means algorithm to our medication example, we first choose the value of k, which corresponds to the number of clusters we want to form. Let's go back to the example where we represent each medication as a point in a three-dimensional colour space, where the x, y, and z coordinates correspond to the amounts of red, green, and blue (RGB) in each medication's colour. For example, a medication with colour (128, 64, 192) would be represented as a point (128, 64, 192) in the colour space. K-means can help categorize medications based on their RGB colour values, allowing us to gain insights into the underlying structure and patterns of the data.

In conclusion, the k-means algorithm is a powerful tool for unsupervised clustering of data based on similarities in their properties. You can appreciate that the k-means method is a straightforward procedure. Most crucial is the choice of the number of clusters, which can be explored as a hyperparameter. More information about the selection of such hyperparameters will follow later in the current module.

As can be seen in the example of sorting medication and the different solutions proposed by your peers, distinctive features of the medication can play a significant role in the decision of how to perform the clustering. All those features constitute a high-dimensional space, and it is not possible to visualize more than three dimensional data in a single plot. In that case, you might consider **dimensionality reduction**, as introduced on the next page.

4.2.3 Dimensionality reduction

Healthcare datasets often contain a large number of features, such as patient demographics, clinical measurements, genetic data, medical imaging, etc, which constitutes a high-dimensional space. This makes data analysis, visualization and interpretation difficult. Here is where **dimensionality reduction** comes in. It plays a vital role in extracting meaningful information, improving computational efficiency, facilitating visualization and interpretation, and advancing research and clinical applications in healthcare.

Dimensionality reduction is the process of reducing the number of input variables or features in a dataset while preserving most of the relevant information. It aims to simplify complex datasets by transforming them into a lower-dimensional space, making it easier to analyse, visualize, and interpret the data.

As datasets become more complex and contain more variables, dimensionality reduction methods help streamline data representation and visualization, enabling more efficient computation and sometimes even improve model performance. By **condensing the data into a lower-dimensional space**, these methods not only enhance the understanding of patterns and relationships within the data, but also contribute to faster training and inference, making dimensionality reduction an indispensable tool for handling high-dimensional data in AI applications.

Dimensionality reduction is particularly valuable in scenarios where the **number of features is large compared to the number of samples**, such as in genomic data, image datasets, and text corpora. By reducing the dimensionality, AI systems can become more interpretable, computationally efficient, and perform better in downstream tasks like classification, clustering, and anomaly detection.

From 3D to 2D

Consider a scenario where you take a picture of two people. In this scenario, it indirectly comes down to transforming a three-dimensional (3D) environment into a two-dimensional (2D) depiction.

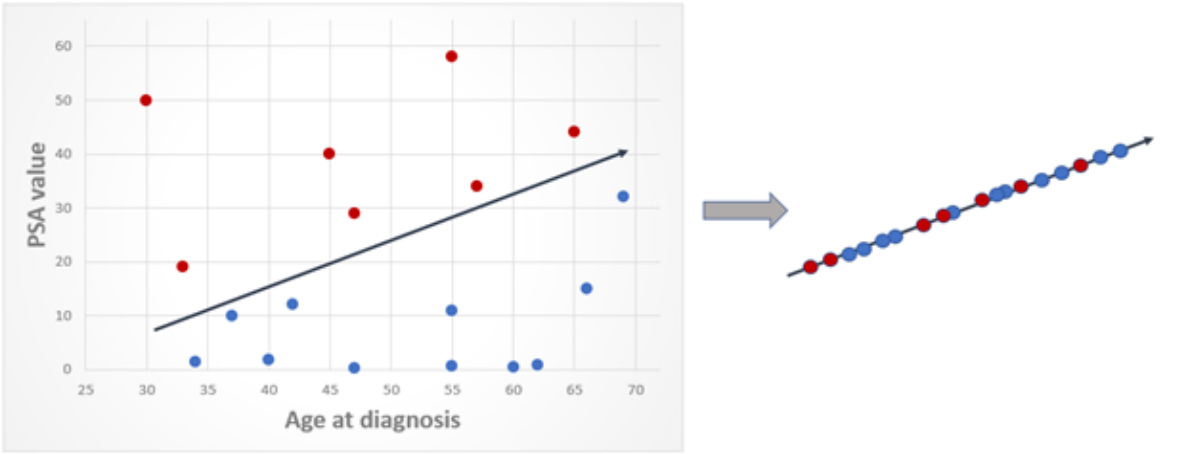
Some explicit 3D information will get lost, like the distance between the person in the foreground and the other person standing more in the background. However, you can still estimate the actual distance between them, because the person in the background appears smaller than the one in the foreground, thereby encoding the 3D information into the 2D image, though not entirely. Hence, the 3D space has been transformed to 2D space, retaining most of the information that is needed.

Dimensionality reduction & information loss

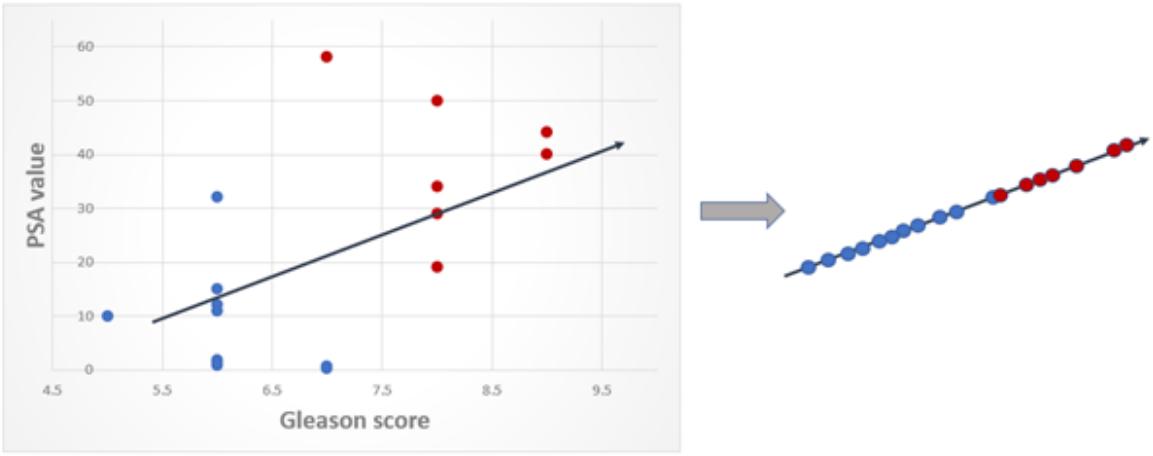
It is essential to strike a **good balance between dimensionality reduction and information loss**. While reducing dimensions, there is a possibility of losing some relevant information that may be crucial for certain applications. Hence, selecting an appropriate dimensionality reduction method and tuning its parameters carefully, are vital considerations for successfully applying this technique in AI.

Example: We want to classify the patients with prostate cancer that will progress into metastatic state. The patients that will **not evolve into metastatic state** are represented with the **blue** bullets in the figure below, while those that will have **metastasis** are represented with a **red** bullet.

Initially, we try to cluster the two cohorts based on the age of the patient at diagnosis of prostate cancer and the prostate-specific antigen (PSA). As you can see in the figure, when we move from the 2D space into the 1D space (by simple projection to the principal component, which will be further explained in the next section), the ability to distinguish (cluster) the two patient populations is lost, since valuable information is lost. The classification is not possible in 1D space. But in many cases, the simplification achieved by dimensionality reduction outweighs the loss in information, and the loss can be partly or fully reconstructed.



In the same example, if the available features "age" and "PSA" were changed to "Gleason score" (= grading system depicting aggressiveness of prostate cancer) and "PSA", we can see in the figure below that the dimensionality reduction helps to separate the two groups:



We will discuss various dimensionality reduction approaches in the next paragraphs: Principal Component Analysis (PCA), t-distributed Stochastic Neighbour Embedding (t-SNE), and manifold learning approaches in general. In the healthcare domain specifically, these approaches play a vital role in extracting meaningful information, improving computational efficiency, facilitating visualization and interpretation, and advancing research and clinical applications.

4.2.4 Dimensionality reduction: Principal Component Analysis

Principal Component Analysis (PCA) is a technique used for dimensionality reduction that transforms high-dimensional data into a lower-dimensional space while preserving the variance of the data. Variance is a statistical measure showing how spread the variables of a dataset are.

PCA is frequently utilized to streamline complex data, filter out noise, and detect latent variables that may not be directly measured. PCA enables us to identify a smaller set of features that can represent the original dataset in a compressed form, while retaining a certain level of variance based on the number of new features selected. The transformation is designed to maximize the variance of the first (or main) principal component, which is the feature or component that captures as much variability in the data as possible. Subsequent orthogonal components (unrelated to each other) are then chosen to maximize the remaining variance in the data, in descending order of importance.

4.2.5 Dimensionality reduction: Manifold learning and t-sne

Manifold learning is a dimensionality reduction technique that aims to preserve the underlying structure of data when reducing its dimensions. It identifies a lower-dimensional manifold (a curved subspace) embedded within the higher-dimensional data space.

t-distributed stochastic neighbour embedding (t-SNE) is a specific manifold learning algorithm commonly used for visualizing high-dimensional data by capturing local relationships and representing them in a lower-dimensional space, often used in data visualization and clustering tasks. It works by modelling the high-dimensional data as a probability distribution and then modelling the low-dimensional representation as another probability distribution. The goal is to minimize the divergence between the two probability distributions. This results in a low-dimensional representation of the high-dimensional data that preserves its structure.

Manifold learning approaches, contrary to PCA, are **non-linear methods**, hence, they can handle better data with non-linear relationships and complex structures. Furthermore, PCA focuses on retaining the maximum variance in the data when reducing dimensions. It tends to capture the directions of highest

variability in the data. On the other hand, manifold learning approaches aim to **preserve the local relationships between data points**, ensuring that nearby points in the high-dimensional space remain close to each other in the reduced-dimensional space, contrary to PCA which aims to maximize variance by preserving large pairwise distances between data points.

Using t-SNE on the example of sorting medication, we could represent the colours and types/use of medication in a two-dimensional space by minimizing the divergence between the high-dimensional and low-dimensional probability distributions. This would allow us to visualize the different clusters of medication based on their colours and types in a more intuitive and understandable way. For example, we might expect medication with similar colours and type to be grouped together in the low-dimensional space.

Now that we learned all about unsupervised learning, let us move on to the next subsection about supervised learning.

4.3 Supervised learning

4.3.1 Supervised learning

Based on what your peers selected in the previous exercise, the categorizations proposed by unsupervised learning are now reduced to the following:

- Two groups based on alphabetical order
- Pills vs other
- Grey-scale vs colourful medication

As you can note, based on the labelled (pre-categorized) examples (data points), the possible categorizations became more limited. You can already intuitively anticipate that the more labelled examples would be given, the more we will converge to a separation in two unique classes. This type of classification for which we provide **labelled data points** is called **supervised learning**.

The supervised learning model is referred to as "supervised", because it resembles a **teacher-student** relationship, where the algorithm (student) learns from the training dataset (teacher) by making predictions and receiving corrections from the teacher. Supervised learning empowers healthcare providers with data-driven insights, enhances patient care, and contributes to advancements in medical research and innovation.

From input to output

Supervised learning is a type of machine learning where an algorithm is trained to **predict an output, based on labelled input examples**. Supervised learning is a prevalent technique in practical machine learning for biomedical applications. The approach involves using an algorithm to learn the **mapping function** from the input variables (characteristics, X) to the output variable (Y). The goal is to create an accurate approximation of the mapping function, so that when new input data are presented, the algorithm can predict the output variable (Y). The mapping function enables the algorithm to transform the input data into new representation spaces that define decision boundaries, allowing it to separate existing classes. Decision boundary is the boundary between the different classes, based on which decision is made about the class of the data point.

Training and testing

The process of supervised learning involves two stages: training and testing. During the **training** stage, the algorithm is fed a set of labelled examples and adjusts its internal parameters to minimize the difference between the predicted output and the true output. The goal is to find a function that can

accurately predict the output for new, unseen inputs. Once the algorithm has been trained, it is evaluated on a separate set of labelled examples not used during training. This **testing** stage allows us to measure the algorithm's performance and to assess its ability to generalize to new data.

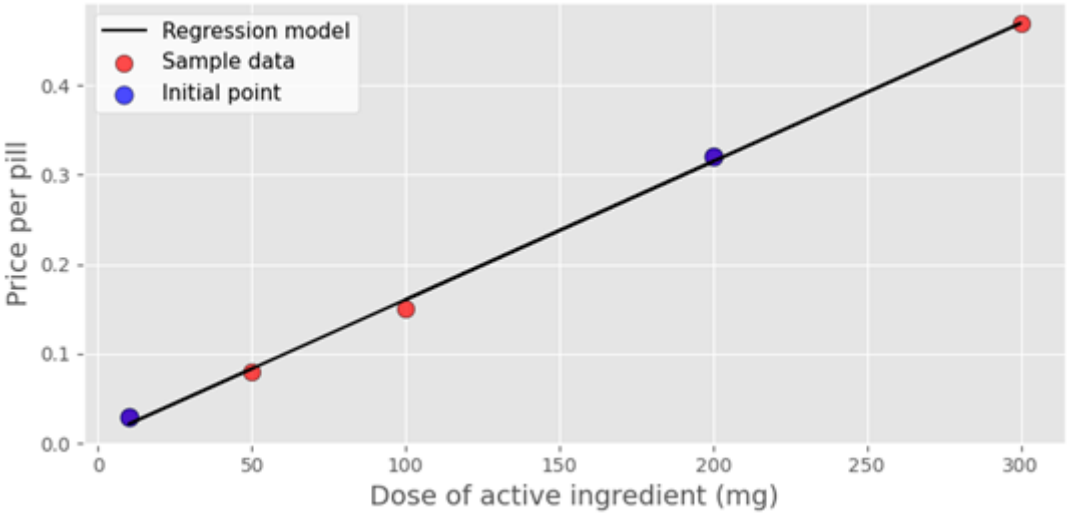
Regression and classification

Supervised learning can be further divided into two categories: regression and classification.

- In **regression**, the output variable is continuous, meaning that it can take on any value within a certain range. The task of a regression problem is to estimate relationships among the continuous variables. For example, one could estimate the price for all medications based on a set of given characteristics, predict the length of stay of a patient in a hospital (Y) based on his/her age (X_1) and the amount of blood lost during the surgery (X_2) etc.
- In **classification**, the output variable is categorical, meaning that it falls into one of several predefined classes and the algorithm tries to estimate the accurate output label for each input datapoint. This corresponds to identifying the decision boundary between the different classes. Several classification tasks are being faced in everyday clinical routine, such as: classification of tumours into malignant or benign, of classification of EEG segments into healthy or epilepsy, etc.
    - When a classification task needs to distinguish between two different classes (as the examples we mentioned), then the classification is called **binary**.
    - In cases where we need to classify the data points in more than two classes, the classification problem is referred to as a **multi-class** classification problem. Multi-class classifications are usually more challenging to solve than binary classifications.

There are several algorithms that can be used for supervised machine learning, such as linear and logistic regression, k-nearest neighbours, linear discriminant analysis, decision trees (and random forests), support vector machines and neural networks to name a few. We will explain the use and function of those methods in the following sections of the current and next module.

4.3.2 Regression

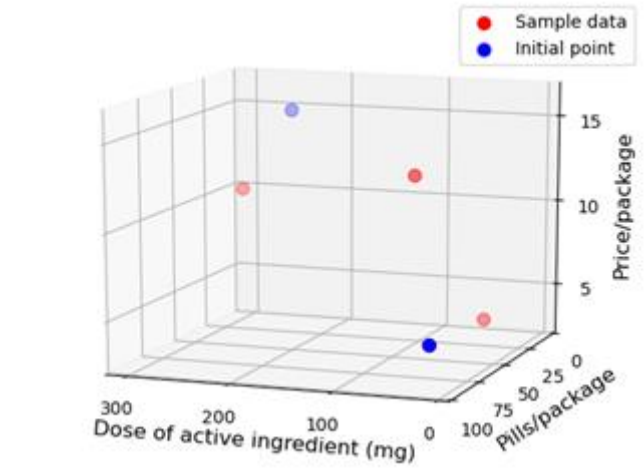


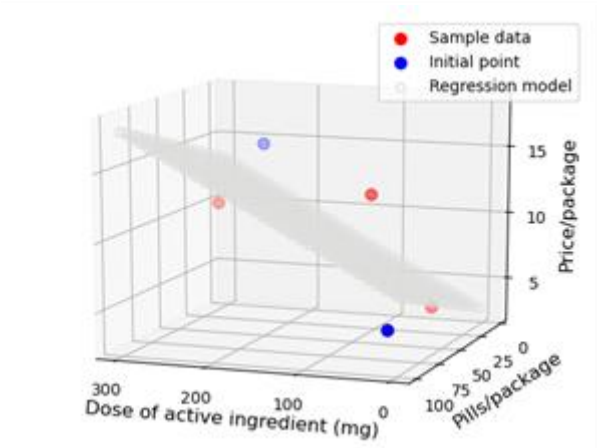
As you can see in the figure above, we placed the mean price predictions of your peers on a cartesian system, with the X- axis representing the concentration of the active ingredient of each pill (mg) of the medications, and the Y-axis representing the prediction for the weight of the medication. As you can note, even unconsciously, the estimations you made were based on the price of the medication you

had as input (labelled data points) and the concentration of the active ingredient that was available for every medication. Hence, you have fulfilled your first regression task.

Regression is a supervised learning algorithm, used to predict a continuous output variable based on one or more input variables (independent variables).

In the example of estimating the price of medication based on the number of pills, regression could be used to develop a model that can predict the price of a medication given its number of pills. To do this, we would first collect data on the number of pills for each medication and its price. We would then use this data to train a regression model, which would learn the relationship between the number of medication doses and its price. In this example, a single variable is used to predict the value of a numerical dependent variable (such as the price in our example), then such a linear regression algorithm is called **simple linear regression**.





Similarly to the previous task, we have placed the mean predictions of your peer students in a 3D graph. Now we have added the number of pills per package on the Z axis. As you can note, the predictions have also been affected from this input variable (feature). It makes sense that when you buy bigger packages, which include more pills, the price per pill drops. As you probably know, other factors can influence the price of medication, such as the use of the medication, the number of doses/pills per package, the manufacturer and others. Hence, when more than one independent variable is used to predict the value of a numerical dependent variable, then such a linear regression algorithm is called **multiple linear regression**. In the right panel of the figure above (b) we fitted such a multiple linear regression based on your estimations. The more we increase the number of data points for which you

know the price, the more information you will have, resulting in less noise (inaccuracy) in the estimations.

There are several types of regression algorithms that can be used, depending on the complexity of the relationship between the input and output variables. In this case, a simple linear regression model could be used, which assumes that there is a linear relationship between the dose of the active ingredient (mg) of the medication and its price. The **linear regression model** would then use the following equation:

Price = slope * dose + constant

In this equation, the constant and slope (which is usually called weight or coefficient of the input feature) are parameters learnt by the algorithm during training. Once the regression model is trained, it can be used to predict the price of one pill of new medications based on the quantity of the dose.

In the healthcare domain, linear regression has been used for several applications such as examining the relationship between risk factors and disease, predicting readmission rates, forecasting hospital bed occupancy, predicting medical costs and others.

In linear regression we assume that the relationship between the variable of interest (weight) and the dependent variable (size) is linear. We can extend this to multiple input variables. Suppose we want to predict the risk of cardiovascular disease (CVD) based on a patient's age and blood pressure. We have a dataset of patients with known ages, blood pressures, and whether or not they have CVD and the time to getting CVD. We can start by fitting a multiple linear regression model:

CVD risk = b1 (age) + b2 (blood pressure) + constant

By bx we represent the coefficients (weights) of each feature and still assume a linear relationship. However, the relationship between age and CVD risk might not be linear. Instead, it might have non-linear dependency. For example, CVD risk may increase more rapidly with age for older patients. In that case, we might try to fit a **non-linear curve** to our **polynomial regression model**:

CVD risk = constant + b1 (age) + b2 (age^2) + b3 (blood pressure)

This model allows for a curved relationship between age and CVD risk, which may provide a better fit to the data than a simple linear model. Polynomial regression involves fitting a polynomial equation to the data, rather than a straight line. This can be useful when the relationship between the independent and dependent variables is not linear. You might wonder why we chose this polynomial model. The answer is “we don't know if this is the best”. In practice we can try various models, and see which models provide a good model fit based on the available data. One can even consider the model choice a hyperparameter (see later subsection).

Other types of regression

Ridge regression

This is a type of linear regression that is used when there is multicollinearity (high correlation) among the independent variables. It adds a penalty term to the regression equation, which helps to reduce the impact of multicollinearity.

In the example above about predicting CVD, different features can be added, other than age and blood pressure, such as the BMI of the subject, cholesterol level, glucose level, smoking status, physical activity, and dietary habits. As you probably know, many of those features are highly correlated. For example, the cholesterol level and the glucose level of the subjects are correlated with the BMI,

physical activity can also be correlated with the smoking status and dietary habits and so on. To decrease the influence of specific features that have minor contribution to the outcome, it shrinks their respective coefficients by adding a penalty term.

Lasso regression

This is like ridge regression, but it adds a different type of penalty term that encourages some of the coefficients in the regression equation to be exactly zero. This can help to identify the most important independent variables and simplify the model.

In the example of predicting CVD, assume that we add in our model some features like number of siblings or PSA (prostate specific antigen) that do not have any impact in the outcome of interest. Lasso regression by applying a penalty, sets their coefficients to exactly zero, in contrast to ridge regression which decreases the coefficients as much as possible.

Elastic net regression

This is a combination of ridge and lasso regression, which adds both penalty terms to the regression equation. It can help to overcome some of the limitations of each method and provide a more robust model.

Overall, the choice of the type of regression model to use depends on the specific research question and the underlying assumed relationship between the variables. The use of a more complex model like elastic-net regression that needs to optimize two different penalty terms is also more expensive in terms of computational power. In healthcare, it is common to encounter non-linear relationships between variables, so it is important to consider a range of regression models to identify the best fit.

4.3.3 Classification

Next to regression, another category within supervised learning is classification.

Supervised classification is a machine learning task where the algorithm learns from labelled data to predict the category or class of new unseen data.

Supervised classification is a fundamental and probably the most prevalent task in AI, falling under the broader umbrella of supervised learning. The goal of supervised classification is to build a model that can **predict the class or category of a target variable based on a set of input features** (like the exercise where medication had to be sorted based on a few labelled examples). In this type of learning, we have a labelled dataset, where each data point is associated with a specific class label. The learning algorithm analyses these labelled examples to identify patterns and relationships between the input features and the target variable. Once the model is trained on the labelled data, it can be used to make predictions on new, unlabelled data, assigning the most likely class label to each instance.

Supervised classification plays a vital role in various healthcare applications, enabling accurate and efficient diagnosis and prediction. In the context of healthcare, there are several types of supervised classification tasks, many of which you will see in the upcoming "AI in action" modules. Usually, the models that are used for classification (and regression task) are separated into handcrafted feature-based models and neural networks. This distinction is not so important but is rather the result of historical research into classification models.

On the next pages in this subsection we will provide an overview of the most common handcrafted feature-based models used in healthcare applications:

- Logistic regression
- Linear discriminant analysis
- Decision trees and random forests
- Support vector machines

4.3.4 Classification: Logistic regression

Logistic regression is a type of supervised classification algorithm used for binary classification tasks. It models the probability of a binary outcome based on one or more predictor variables. The output of logistic regression is a probability score between 0 and 1, which is then mapped to a discrete class label (e.g., 0 or 1) based on a chosen threshold.

Logistic regression is a generalized linear model used for **binary** classification problems, where the target variable (dependent variable) can take one of two possible classes or categories (e.g., 0 or 1, yes or no). Despite its name, which can be misleading, logistic regression is a **classification** algorithm and not a regression algorithm. The goal of logistic regression is to **model the probability of the binary outcome** as a function of one or more predictor variables (independent variables). It uses the logistic function (sigmoid function) to map the predicted values to probabilities, which ensures that the predicted probabilities are between 0 and 1.

To make predictions, a **threshold** (usually 0.5) is chosen, and if the predicted probability is greater than the threshold, the data point is classified as 1; otherwise, it is classified as 0.

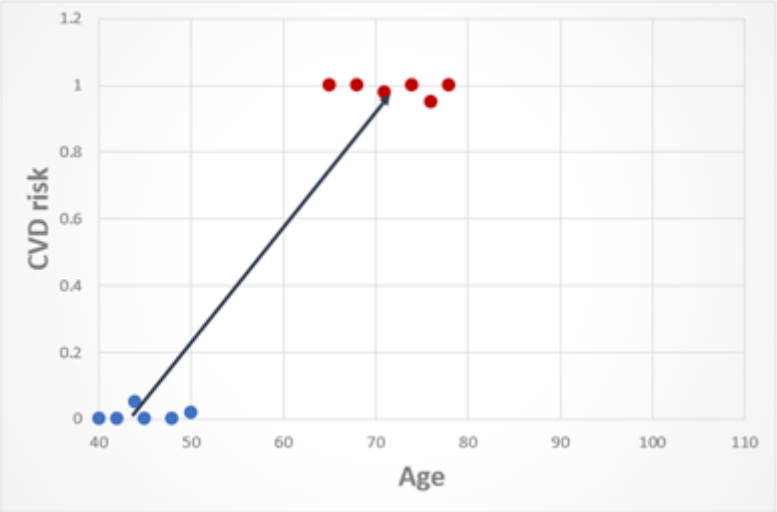
As mentioned before, the main difference between logistic regression and linear regression lies in their use cases and the nature of the dependent variable they handle. Linear regression is a method for regression, hence, it predicts a continuous outcome and fits a linear equation to the data, trying to minimize the errors between the predicted values and the actual values. In contrast, logistic regression is specifically designed for binary classification. It models the probability of belonging to a particular class and uses a nonlinear transformation (the logistic function) to ensure that the predicted probabilities are within the valid range of 0 to 1. So let us recall the basic function that we used for the estimation of cardiovascular disease (CVD). For simplicity we will assume that the risk of CVD is only based on blood pressure, hence, we have the following linear regression function for CVD:

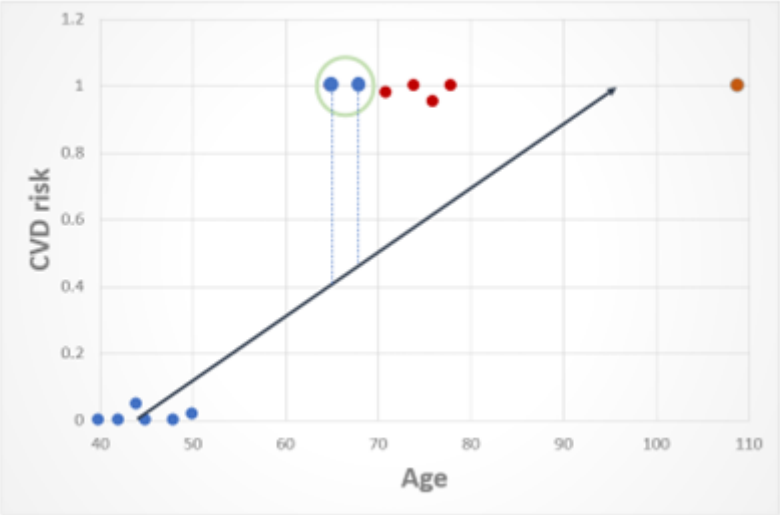
CVD risk = b1 (blood pressure) + constant

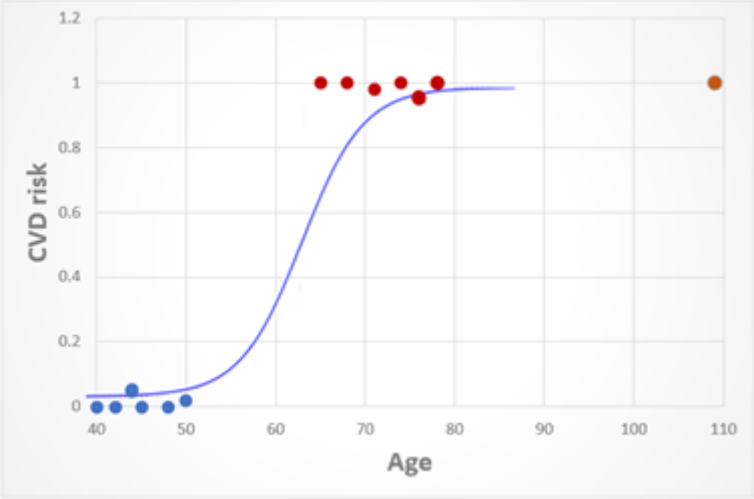
The risk score is a continuous variable. As we mentioned, logistic regression tackles discrete variables (0 and 1) so instead of providing a risk, we will try to identify which patients belong to the cluster of CVD, and which are the subjects that belong to the healthy cluster. We assume that every patient with risk above 0.5 (above 50%) will belong to the class 1, and all the subjects below 0.5 belong to the healthy class. The cut-off of 0.5 is our decision boundary and can potentially be optimized, but since it is the mean of the 0-1 boundary, it usually acts as a good threshold. Under those assumptions, could we adapt the use of linear regression in order to solve a classification task?

In the example in figure A below, when our dataset does not have any outlier the **linear** regression can fit a line that correctly distinguishes all the red-class (CVD patients) points correctly, since they have a value above 0.5 (perpendicular line from the point towards the regression line). In the case that we have an outlier in our dataset (figure B), linear regression will try to fit a new line that will be distanced equally from all the points. This will result in two of our points (inside the circle in figure B) to be misclassified since the CVD risk will be below 0.5 (around 0.4 – dotted perpendicular line to regression line). Hence, we showed that even under certain assumptions, linear regression is very susceptible to outliers. A new line must be used that “resembles” the behaviour of the linear regression line, but that

has a minimum and maximum to 0 and 1, respectively, and is not heavily influenced by the outliers. This line is the **sigmoid** function (figure C) or **logistic** regression function that is used for logistic regression.







4.3.5 Classification: Linear discriminant analysis

Linear discriminant analysis (LDA) is a dimensionality reduction and classification technique commonly used in the field of pattern recognition and machine learning. LDA is designed to find the linear combination of features that best separates or discriminates between two or more classes in a dataset. It is particularly useful for problems involving multi-class classification.

The main goal of LDA is to **project the original data into a lower-dimensional space while maximizing the separation between classes**. It does this by maximizing the distance between the means of the classes (inter-class distance), while minimizing the variance within each class (intra-class distance). This means that LDA aims to create a projection where data points from the same class are tightly clustered together, and different classes are well-separated.

A nice property of LDA is that it is the optimal classifier if these assumptions about the data are met:

- Data in both classes are Gaussian distributed
- Data in both classes have equal covariance matrices
- Data have known covariance matrices

Covariance matrix is a square matrix that summarizes the covariance relationships between multiple variables. In statistics and probability theory, covariance measures how much two random variables change together. The covariance matrix provides a compact way to represent these relationships for a set of variables, with each element representing the covariance between two specific variables.

4.3.6 Classification: Decision Trees and Random Forests

Decision trees are non-linear and non-parametric supervised learning algorithms that can be used for both classification and regression tasks. They work by recursively splitting the data into subsets based on the values of different features, effectively creating a tree-like structure. Each internal node of the tree represents a decision based on a specific feature, and each leaf node represents a class label (for classification) or a numeric value (for regression).

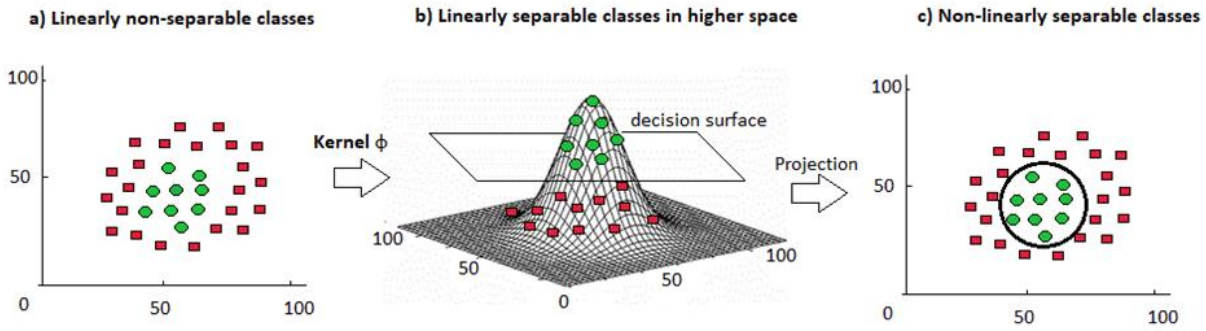
Random forests is an ensemble learning method that leverages the power of decision trees for improved predictive accuracy and reduced overfitting. The fundamental idea behind Random Forests is to create multiple decision trees, each based on a random subset of the original data and a random subset of the features.

4.3.7 Classification: Support vector machines and others

Support vector machines (SVM) are powerful and versatile supervised learning algorithms used for both classification and regression tasks. In the context of classification, SVMs are mainly used for binary classification problems, where the goal is to separate data points into two classes based on their feature values. SVMs are effective for problems with clear class boundaries and are well-suited for high-dimensional feature spaces.

The main idea behind SVMs is to find the **hyperplane** (a decision boundary in a higher-dimensional space) that best separates the data points belonging to different classes. The optimal hyperplane is the one that maximizes the margin, which is the distance between the hyperplane and the nearest data points of each class. These data points, called support vectors, are critical for determining the optimal hyperplane.

The initial algorithm of SVM assumes that the data are **linearly** separable. In real case scenarios, this assumption is not always valid, due to non-linearities or noise in the underlying data generation process. Non-linear versions of the SVM algorithm are also available. The most common approach for classifying **non-linear** data with the use of SVMs is through the adoption of **kernels** and the mapping of the data into a higher dimensional space, where they can be linearly classified (see figure below). The mapping of the original feature space into some higher-dimensional feature space, where the training set is separable, must be done in a way that any coherence information between the data points will be preserved. To map the data into a higher separable dimensional space, kernel functions are used. There are different types of kernel functions that can be used, such as linear, Gaussian, radial basis, polynomial, and others. The selection of the suitable kernel function is problem- and data-dependent and is usually selected based on trial and error.



SVMs have proven to be effective in many **healthcare applications**, such as text classification, image recognition, event detection in signals and bioinformatics. They are especially popular in problems where the number of features is large, and the decision boundary is complex. SVMs are well-regarded for their ability to handle high-dimensional data and their capacity to generalize well to new data points. However, their training time can be computationally expensive for large datasets and selecting the right kernel function and tuning hyperparameters are important for achieving optimal performance.

Other feature-based methods

For your information, other commonly used feature-based methods other than those analysed in the previous units are:

- **K-nearest neighbours (KNN)** is a simple and intuitive algorithm that classifies data points based on the class labels of their k (hyperparameter) nearest neighbours in the feature space. It can be used for both classification and regression tasks.
- **Gradient boosting** is an ensemble learning method that builds multiple weak learners (typically decision trees) sequentially, each one trying to correct the errors of its predecessor. It combines the predictions of these weak learners to form a strong overall model.
- **Naive Bayes classifier** is a probabilistic classification method based on Bayes' theorem. It assumes that features are conditionally independent given the class label and is especially useful for text classification tasks.

This concludes the subsection about supervised learning. Continue to the next subsection to learn about the importance of evaluating the performance of machine learning models. Evaluation is a crucial step to ensure that the model is effective and accurate.

4.4 Evaluation of the performance

4.4.1 Evaluation

In this new subsection of the module, we will delve into **evaluation**. Evaluating the performance of machine learning models is of utmost importance to ensure that the model is effective and accurate. Obtaining good evaluation performance on new, unseen data will allow us to make claims about generalizability. Model evaluation helps in selecting the best model among different algorithms, tuning hyperparameters, and identifying potential issues or shortcomings in the model's performance.

There are several evaluation metrics and techniques to assess the performance of machine learning models. The choice of evaluation metric depends on the **type of task** (classification, regression, clustering, etc.) and the **specific objectives** of the model. For example, in a classification task connected to a decision of an intervention that will be performed (which is invasive and hence of higher risk), you might be interested in minimizing the False Positives (FPs). In a use-case like seizure detection where neurologists are interested in not losing a single seizure, the maximization of True Positives (TPs) is of prime interest. Hence, in both cases we would consider using the metrics that consider FPs and TPs.

4.4.2 Confusion matrix and classification metrics

Confusion matrix is a fundamental tool for evaluating the performance of a classification model. It provides a detailed breakdown of the model's predictions and their correspondence with the actual class labels. The confusion matrix is particularly useful for binary classification tasks, where there are two possible classes (e.g., positive, and negative).

In a binary classification scenario, the confusion matrix is a 2x2 matrix...

Total segments	Predicted positive (1)	Predicted negative (0)
Actually positive (1)	(TP)	(FN)
Actually negative (0)	(FP)	(TN)

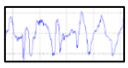
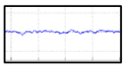
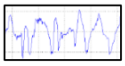
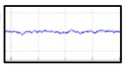
...whereby

- **TP =** the number of instances correctly predicted as the positive class (true positives).
- **TN =** the number of instances correctly predicted as the negative class (true negatives).
- **FP =** the number of instances incorrectly predicted as the positive class (false positives).
- **FN =** the number of instances incorrectly predicted as the negative class (false negatives).

Based on these values, various classification metrics can be derived:

- **Accuracy**: The proportion of correct predictions out of the total number of predictions given by (TP + TN) / (total samples).
- **Precision (Positive predictive value)**: The proportion of true positive predictions out of the total positive predictions, given by TP / (TP + FP).
- **Sensitivity (Recall, True positive rate)**: The proportion of true positive predictions out of the total actual positive instances, given by TP / (TP + FN).
- **Specificity (True negative rate)**: The proportion of true negative predictions out of the total actual negative instances, given by TN / (TN + FP).
- **False positive rate (False alarm rate)**: The proportion of false negatives predictions out of the total actual negative instances, given by FN/ (TN + FP).
- **F1-score**: The harmonic meaning of precision and recall, which balances the trade-off between precision and recall. It is given by 2 * (Precision * Recall) / (Precision + Recall).

Let us consider an example, in which we want to classify 1000 different 1-second EEG segments being either rest-EEG or epileptic EEG. We assign the positive class (1) to the class containing the epileptic segments and the negative class (0) to the class containing background rest-EEG. Since the classification of epileptic segments is a highly unbalanced problem (more segments of a daily EEG recordings belong to rest-EEG and much less to the epileptic-EEG) we assume that the 950 out of 1000 segments belong to the negative class (the unbalance is even larger, with epileptic segments accounting for less than 1%). The confusion matrix of a possible classification task takes the following form:

Total segments	**Predicted Positive (1)** (epilepsy) 	**Predicted Negative (0)** (rest-EEG) 
Actually Positive (1) (epilepsy) 	40 **(TP)**	10 **(FN)**
Actually Negative (0) (rest-EEG) 	90 **(FP)**	860 **(TN)**

In this example, **TN** is any EEG segment without epileptic spikes classified as rest-EEG, **FN** an EEG segment with epileptic spikes classified as rest-EEG. **FP** an EEG segment without epileptic spikes is classified as epileptic spike EEG and **TP** an EEG segment with epileptic spikes is classified as a segment with epileptic spike(s).

Based on our confusion matrix we can derive the previously defined metrics:

- **Accuracy**: (40 + 860) / 1000 = 0.9
- **Precision**: 40 / (40 + 90) = 0.3
- **Sensitivity**: 40 / (40 + 10) = 0.8
- **Specificity**: 860 / (860 + 90) = 0.91
- **False positive rate**: 90 / (860 + 90) = 0.1
- **F1-score**: 2 * 0.3 * 0.8 / (0.3 + 0.8) = 0.436

4.4.3 ROC curve

Decision thresholds

In classification tasks, **thresholds** play a crucial role in determining the **point at which the continuous output of a model**, such as the predicted probability in logistic regression, **is categorized into discrete classes.**

For instance, in a binary classification problem, a common threshold is 0.5, where predicted probabilities above this value are classified as one class, and those below as another (see again logistic regression where we also discussed the selection of the threshold). Adjusting this threshold can significantly impact the model's sensitivity and specificity, influencing the balance between identifying true positives and avoiding false positives. The Area Under the ROC Curve (AUC) provides a comprehensive measure of a model's discriminative ability across all possible thresholds, offering insights beyond what can be gleaned from true positives and false positives at a single threshold.

ROC curve

The **Receiver Operating Characteristic (ROC) curve** is a pivotal tool in evaluating the performance of classification models in artificial intelligence, particularly in distinguishing between different classes, such as patients with and without cardiovascular disease (CVD) based on biomarkers like age and BMI.

To construct a ROC curve, one must calculate the **True Positive Rate (TPR)** and the **False Positive Rate (FPR)** across various decision thresholds, which could range from 0.00 to 1.00 in a logistic regression

model, for instance. On the ROC curve, the FPR is plotted on the x-axis while the TPR is plotted on the y-axis, effectively summarizing the trade-offs between true positive identifications and false alarms as the classification threshold varies. The curve thus provides a visual representation of a model's ability to discern between classes under various conditions.

AUC

The **Area Under the ROC Curve (AUC)** is an integral metric in evaluating classification models because it provides a comprehensive measure of a model's ability to discriminate between classes across all possible thresholds. While **True Positives (TPs) and False Positives (FPs)** offer valuable insights into a model's performance at a specific threshold, they do not encapsulate the model's overall effectiveness. Here's why relying solely on TPs and FPs might not be adequate and the necessity of the AUC:

1. **Threshold dependency**: TPs and FPs are highly dependent on the chosen threshold for classification. A model might perform exceptionally well at one threshold but poorly at another. This threshold-specific performance doesn't offer a holistic view of the model's discriminative power.
2. **Imbalanced classes**: In datasets where one class significantly outnumbers the other, using TPs and FPs can be misleading. For instance, in a medical dataset with a small proportion of positive (disease) cases, a model might appear to perform well by predominantly predicting the majority class (no disease), leading to low FPs but also low TPs, masking its inability to identify the minority class effectively.
3. **Trade-off between sensitivity and specificity**: TPs (contributing to sensitivity or recall) and FPs (affecting specificity) represent two aspects of a model's performance that are often in trade-off. A model might be tuned to increase TPs, which might also increase FPs, decreasing specificity. Evaluating models based solely on TPs and FPs doesn't adequately address this trade-off.
4. **Comprehensive evaluation**: The AUC provides a singular value that represents the model's ability to discriminate between classes across all thresholds, offering a more comprehensive evaluation. A higher AUC indicates that the model has a high true positive rate across various false positive rates, highlighting its robustness and reliability.
5. **Model comparison**: AUC allows for an intuitive comparison between models. Since it summarizes the model's performance across all thresholds, it provides a clear criterion for comparing different models' effectiveness in classifying instances, making it easier to select the best model for a given task.

In summary, while TPs and FPs are crucial for understanding a model's performance at specific thresholds, the AUC offers a more holistic and threshold-independent measure of a model's ability to distinguish between classes, making it an indispensable metric in model evaluation and comparison.

4.4.4 Regression metrics

Regression metrics play a vital role in evaluating the performance of AI models, offering a systematic way to measure the predictive accuracy and reliability of these models in various real-world applications. Regression metrics include the following:

- **Mean absolute error (MAE):** This metric calculates the average absolute difference between predicted and actual values. It is less sensitive to outliers compared to some other metrics.

Formula: MAE = (1/n) * Σ|actual - predicted| , where Σ indicates summation

- **Mean squared error (MSE):** MSE measures the average of the squared differences between predicted and actual values. It penalizes larger errors more heavily.

Formula: MSE = (1/n) * Σ(actual - predicted)²

- **Root mean squared error (RMSE):** RMSE is the square root of MSE, which gives a measure of error in the same unit as the target variable. It is easier to interpret than MSE.

Formula: RMSE = √(MSE)

- **R-squared (R^2):** R-squared measures the proportion of the variance in the dependent variable that is explained by the independent variables. It ranges from 0 to 1, with higher values indicating a better fit.

Formula: $R^2 = 1 - (SSR/SST)$, where SSR is the sum of squared residuals and SST is the total sum of squares.

- **Mean absolute percentage error (MAPE):** MAPE calculates the average percentage difference between predicted and actual values, making it useful for revealing percentage-based errors.

Formula: MAPE = (1/n) * Σ(|(actual - predicted)/actual|) * 100

4.4.5 Clustering metrics

Clustering metrics are essential evaluation tools used to quantify the quality and effectiveness of clustering algorithms:

- **Silhouette score**: Silhouette score measures how similar an object is to its own cluster compared to other clusters. It ranges from -1 to 1, with higher values indicating better cluster separation.
- **Davies-Bouldin index**: This index quantifies the average similarity between each cluster and its most similar cluster. Lower values indicate better clustering.
- **Calinski-Harabasz index (Variance ratio criterion)**: Also known as the variance ratio criterion, it measures the ratio of between-cluster variance to within-cluster variance. Higher values suggest better-defined clusters.
- **Dunn index**: Dunn Index evaluates the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance. A higher Dunn index indicates better clustering.
- **Adjusted Rand index (ARI)**: ARI measures the similarity between the true labels and the predicted clusters, while correcting for chance. It ranges from -1 to 1, with higher values indicating better clustering.
- **Normalized mutual information (NMI)**: NMI measures the mutual information between true labels and predicted clusters while normalizing for cluster and label imbalance. It ranges from 0 to 1, with higher values indicating better clustering.
- **Jaccard index**: Jaccard Index calculates the similarity between two sets by dividing the size of their intersection by the size of their union. It can be used to measure the similarity of two clusterings.

In conclusion, it is essential to evaluate the performance of machine learning models using appropriate metrics and techniques to make informed decisions about model selection, hyperparameter tuning, and potential improvements.

Additionally, evaluating models on representative and unbiased datasets is crucial to ensure their reliability and applicability in real-world scenarios. Regularly monitoring and updating model

performance are also important to adapt to changes in the data distribution and maintain optimal performance over time.

Next, it is important to have proper feature and model selection. Let's learn about it in the next subsection.

4.5 Feature and model selection

4.5.1 Feature selection

Welcome to this subsection about feature and model selection. First of all, **feature selection** is the process of isolating the most consistent and relevant features to use in model construction. How can one achieve proper feature selection? That's what Aisha wonders too...

After following the current module, Aisha thinks of employing an AI solution for predicting diabetes. She wants to develop a machine learning model to predict the likelihood of a patient developing diabetes. While looking for appropriate data to work with, she came across large available open-source datasets that include real-world data from clinical routine checks, containing a wide range of patient information, including demographic details, medical history, laboratory test results, and imaging data.

In this scenario, **not all features may be equally informative or relevant** for predicting the onset of diabetes. Some features, like age, blood glucose levels, and family history of diabetes, might have a direct and significant impact on the prediction. On the other hand, certain features, such as the colour of a patient's eyes or unrelated medical conditions, may not contribute much to the predictive accuracy.

Without proper feature selection, the model might consider all available features, including the less relevant ones, leading to several **issues**:

- **Increased computational complexity**: Including irrelevant features can result in a more complex model, requiring additional computational resources for training and inference.
- **Overfitting**: The model may learn noise from irrelevant features, causing it to perform poorly on new, unseen data.
- **Interpretability**: A model with too many features becomes harder to interpret, making it challenging for healthcare professionals to understand the factors influencing predictions.

By employing feature selection techniques, researchers and data scientists can identify and retain only the most pertinent features for predicting diabetes. This not only improves the efficiency of the model, but also enhances its interpretability and generalization to new patient data. Features like fasting blood glucose levels, age, and family history, identified through feature selection, would likely play a more prominent role in predicting diabetes, leading to a more accurate and clinically useful model for healthcare practitioners.

Curse of dimensionality

The **curse of dimensionality** is a phenomenon associated with challenges posed by high-dimensional data. As the number of features increases, the data's dimensionality rises, causing the **data to become exponentially sparse** in these higher-dimensional spaces. To accurately represent the data distribution in such expansive spaces, an exponential increase in the number of samples would be required, often impractical given a fixed number of available samples. Additionally, the **sorting or classification of data becomes increasingly challenging** as dimensionality grows, with data points appearing more distant in higher-dimensional spaces.

Sparsity is a concept that describes situations in which the majority of elements in a matrix or dataset are zeros. This property is significant in various computational and mathematical fields (and AI), because it implies that the information contained within the dataset is concentrated in a relatively small number of elements. Sparsity is leveraged in numerous applications, such as signal processing, where sparse representations can lead to more efficient storage and computation, and in AI, where it can improve the performance of algorithms by reducing overfitting and enhancing interpretability.

Feature selection

Feature selection is a critical process in machine learning, involving the identification and inclusion of the **most impactful features that significantly contribute to predicting the desired output variable**. Whether performed automatically or manually, this task is complex and plays a pivotal role in the overall performance of a machine learning model.

The rationale behind feature selection lies in the abundance of available data features. While datasets often encompass a plethora of features, not all are relevant to solving the given problem. The indiscriminate use of all features can escalate the computational cost, as a greater number of features demands more intricate models. Moreover, retaining irrelevant features introduces noise to the data, offering no additional information and adversely affecting model performance. The "curse of dimensionality" exacerbates this issue.

To tackle these challenges, **feature selection methods** have been developed. Usually we separate the methods that are being employed for feature selection into 3 big families.

- **Filter methods**: Filter methods evaluate features based on their intrinsic properties, independent of any machine learning algorithm. Metrics like correlation coefficients and mutual information are commonly used. These methods are computationally efficient, but might overlook feature dependencies. Data exploration tools (see Module 2) are commonly used as a selection tool for the filter methods and in order to better understand which features are redundant or non-discriminative (in a classification task).
- **Wrapper methods:** Wrapper methods assess subsets of features based on the performance of a specific model. Wrapper methods for feature selection use a trial-and-error process to search for the most effective subset of features by evaluating the performance of a specific machine learning model with different combinations of features. They repeatedly train models with various feature sets and select the combination that yields the best model performance. While potentially more accurate, they are computationally intensive. The evaluation of feature subsets is based directly on the performance of a specific model, making wrapper methods highly dependent on the chosen model. This can be advantageous when optimizing for a particular model but limits the generalizability of the selected feature.
- **Embedded methods**: Embedded methods perform feature selection as part of the model training process. Two notable techniques are "Minimum Redundancy Maximum Relevance" and "LASSO" (least absolute shrinkage and selection operator). Minimum Redundancy Maximum Relevance focuses on selecting features that are both highly informative and minimally redundant, promoting an efficient representation of the dataset. LASSO employs regularization to encourage sparsity in the feature space (penalty term to reduce feature coefficients to zero), effectively zeroing out less influential features. Each of these techniques contributes to the optimization of feature sets, enhancing model performance and addressing the intricate issues associated with high-dimensional data. Techniques like LASSO regression effectively perform feature selection while the model is being trained.

4.5.2 Model selection

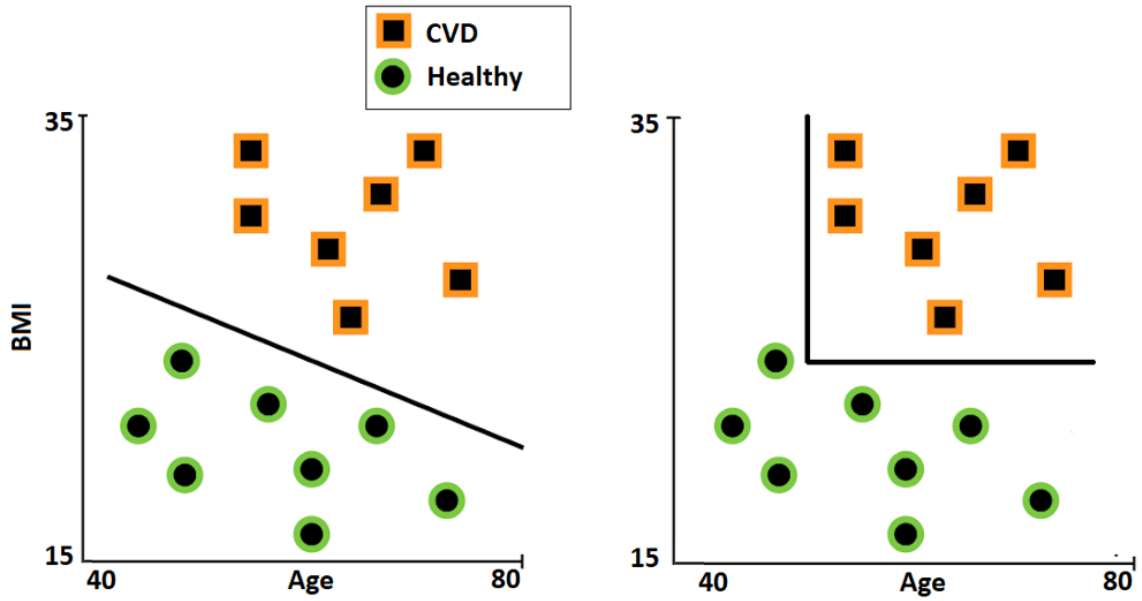
Now that we discussed feature selection, let's explore model selection more closely. Selecting an optimal AI model for a specific use case involves a nuanced process:

1. It begins with a **clear definition of the problem and the establishment of measurable objectives** (the most important step).
2. Then a variety of **candidate models are explored**, ranging from feature-based machine learning algorithms to neural networks (we will see more details in Module 8), each assessed against criteria such as accuracy, interpretability, and computational efficiency. Considerations of scalability, integration capability, and the model's ability to provide actionable insights play a crucial role in the final selection, especially in the domain of healthcare.
3. This iterative process, which might circle back to refine objectives or re-evaluate model choices based on initial findings, culminates in the deployment of a model that not only **addresses the use case's requirements but also adapts to evolving data and operational landscapes**, ensuring sustained relevance and interpretability.

After a machine learning model is trained, it becomes adept at making predictions on the **training data**. However, the true goal transcends mere performance on this training set. For practical deployment, the model must accurately predict outcomes on novel data it hasn't encountered during its training phase. Essentially, a model's value is determined by its ability to generalize, meaning its capacity to apply learned insights to fresh, unseen data, assuming this new data stems from the same distribution as the training set. To evaluate a model's prowess in generalization, we customarily reserve a segment of the available dataset as a **test set**, rather than employing the entire dataset for training. This test set, which remains untouched during the training phase, serves to gauge the model's effectiveness in predicting new instances. Ensuring that the test data is both separate from and representative of the training data is crucial for an accurate appraisal of model performance.

Example: Cardiovascular disease (CVD)

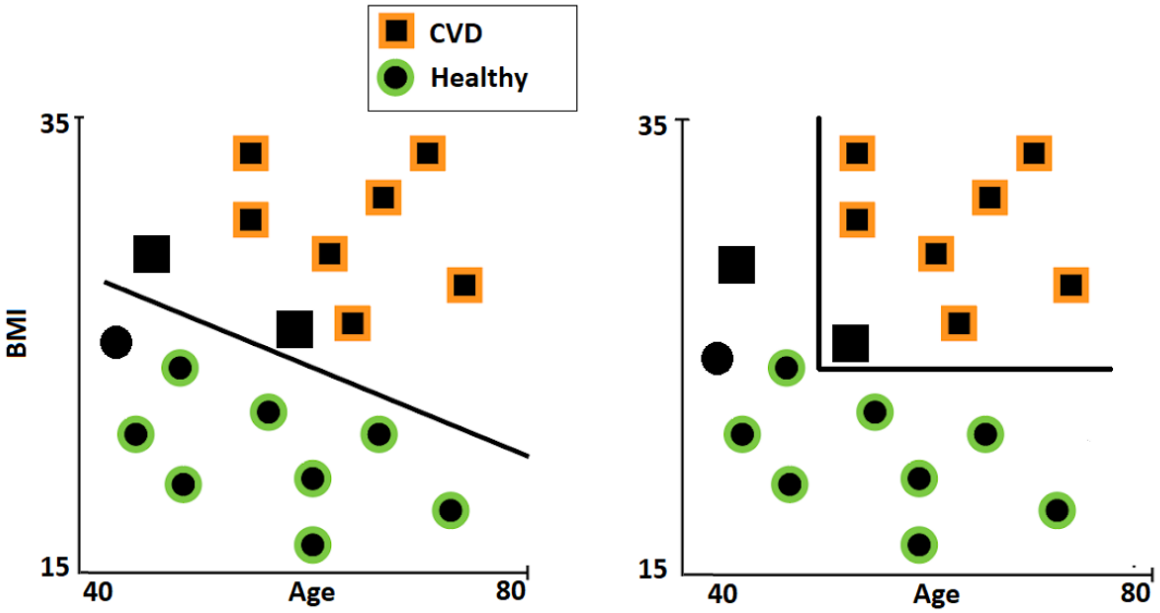
We have two models that classify CVD patients from healthy subjects. In the figure below, we can see the data based on the models they were trained with and their decision boundaries. In the left panel we can view the decision boundary of a **logistic regression model**, and on the right panel this of a **random forest**.



When assessing a machine learning strategy for a given dataset, we initiate by dividing the dataset into two subsets: one for **training** and the other for **testing**. This process encompasses two primary actions:

- **Model training**: Here, the model is exposed to the training data, fine-tuning its parameters to better align with the observed data patterns.
- **Model evaluation**: With the model parameters now set, we proceed to test its predictive accuracy on the test set, employing specific performance metrics for evaluation.

Following the previous CVD example, you can now see in the figure below that there are 3 new data samples as a test set (non-coloured items). Logistic regression is on the left, and random forests is on the right.



4.5.3 Model selection: Cross validation

Selecting suitable training and testing data is vital to ensure the model trains on a dataset that mirrors the overall dataset's characteristics. Similarly, the testing dataset must be representative and balanced to avoid skewed performance evaluations. For instance, a testing set clustered around the decision boundary might undervalue the model's efficacy, while a set composed solely of one class only tests the model's discriminatory ability for that class, neglecting its performance on others.

Cross-validation offers a robust solution to these selection dilemmas by systematically rotating the role of each data segment as the test set within a partitioned dataset. In this approach, often characterized by an arbitrary number of partitions, say 'x', the model undergoes training 'x' times. During each iteration, 'x-1' partitions serve as training data, with the remaining partition used for testing. This cycle ensures each data portion is utilized as a test set precisely once, thereby furnishing an aggregate measure of the model's average performance across the entire dataset.

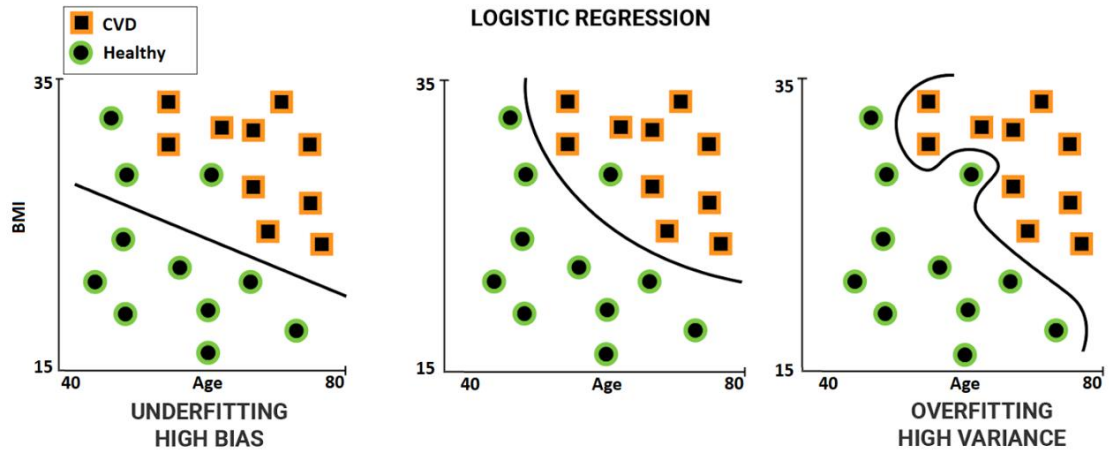
4.5.4 Overfitting

Overfitting occurs when a machine learning model learns the training data too well, capturing noise or random fluctuations rather than the underlying pattern, which leads to poor performance on unseen data.

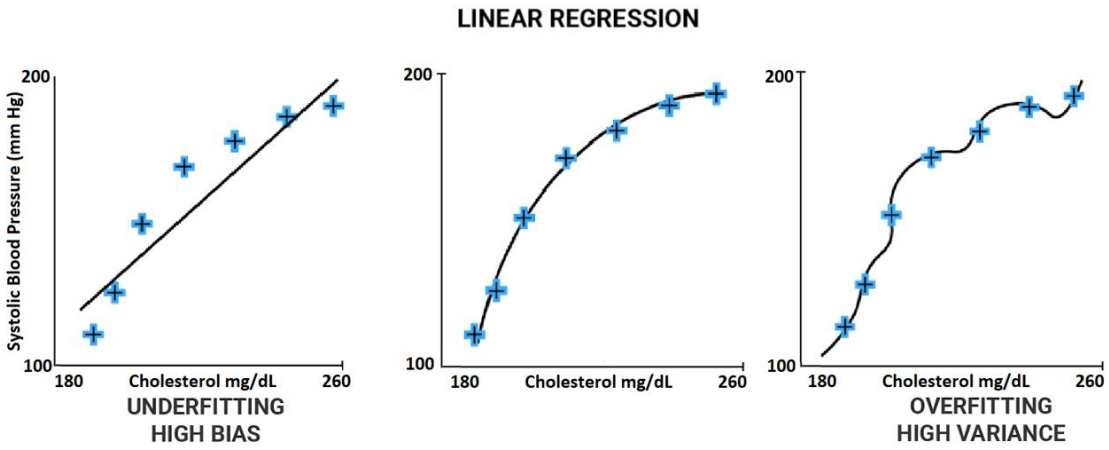
As we discussed, it is essential to evaluate the performance of machine learning models using appropriate metrics and techniques to make informed decisions about model selection, hyperparameter tuning, and potential improvements. Additionally, evaluating models on representative and unbiased datasets is crucial to ensure their reliability and applicability in real-world scenarios. Regularly monitoring and updating model performance are also important to adapt to changes in the data distribution and maintain optimal performance over time.

Let's consider the task of classifying subjects to those having cardiovascular disease (CVD) and the healthy population by using two known biomarkers: age and BMI. The classes are closely intertwined, reflecting the reality that there's no clear-cut threshold where these biomarkers suddenly indicate a high risk. The diagram provided below illustrates **overfitting in logistic regression**, specifically in models that classify patients with CVD based on age and BMI.

- On the left, the model exhibits **underfitting**, failing to adequately capture the relationship between age, BMI, and CVD risk.
- The centre diagram represents an **optimal model** that accurately classifies patients by learning the true patterns of CVD risk from age and BMI without being swayed by data noise.
- The right diagram, however, shows a model that **overfits** by overly conforming to the training data, including its anomalies or 'noise.' As a result, this overfitted model is less effective at predicting CVD in new patients compared to the balanced model shown in the centre.



Overfitting may also occur in **linear regression** tasks. Let us consider a similar regression task where we are trying to develop a regression model that predicts the risk score for cardiovascular disease using systolic blood pressure and total cholesterol levels as predictors.



4.6 Hyperparameters

When the number of possible clusters or categories increased to 3, your peers gave the following options for dividing all medication:

- Pills vs cream vs sirup
- Three alphabetical groups
- Size small vs medium vs big (mg)

If you compare to the sorting exercise with two shelves, you can note that when the number of possible categories increases, the possible categorizations/classifications decrease and the whole classification becomes more difficult. The **number of clusters is considered as a hyperparameter that affects the classification result**. There is no optimal way to select this hyperparameter, so in practice, different values can be explored, and the "optimal" grouping can be retained. Note that "optimal" can be subjective and is often based on domain knowledge, e.g., you know that there are 4 different tumour stages so in a tumour classification problem it might be wise to use 4 different classes. The first consideration is determining the number of clusters. In some cases, the research question may make it clear how many clusters are needed. However, in other situations, the optimal number of clusters is unknown.

Hyperparameter is a parameter that is not learned from the data, but is set prior to training a model. Hyperparameters control various aspects of the learning process and model architecture, influencing how the model learns from the training data. These parameters are essential for configuring and optimizing machine learning algorithms to achieve the best possible performance.

Some key characteristics of hyperparameters:

- **Not learned from data**: Unlike model parameters (weights and biases) that are learned during the training process to make predictions, hyperparameters are set by the data scientist or machine learning engineer before training begins.
- **Influence model behaviour**: Hyperparameters can significantly impact how a machine learning model behaves, including its generalization ability, convergence speed, and capacity to fit the training data.
- **Tuning required**: To find the best combination of hyperparameters for a given problem, practitioners often perform hyperparameter tuning or optimization. This involves systematically trying different values or ranges for hyperparameters to identify the most effective settings.

Commonly used hyperparameters

Some examples of commonly used hyperparameters in various machine learning algorithms are:

- **Learning rate**: A hyperparameter in gradient-based optimization algorithms like stochastic gradient descent (SGD), which determines the step size for updating model parameters during training. More information about learning rate will be provided in the section of Neural Network.
- **Number of hidden layers and neurons**: In neural networks, hyperparameters like the number of hidden layers and the number of neurons in each layer control the architecture of the network. More information about those hyperparameters will be provided later in the section of Neural Network.
- **Kernel type and kernel parameters**: In support vector machines (SVMs), the choice of kernel function (e.g., linear, polynomial, radial basis function) and associated kernel parameters are hyperparameters.
- **Number of trees and tree depth**: In ensemble methods like random forests and gradient boosting, hyperparameters control the number of trees in the ensemble and the maximum depth of each tree.
- **Number of clusters (k)**: As we saw previously in the section about k-means, the choice of the numbers of the clusters is considered a hyperparameter.

Hyperparameter selection

Hyperparameter selection in AI is a critical step in the development of machine learning models, as it significantly influences their performance and efficacy. Hyperparameters are the configuration settings used to structure the model, such as the learning rate in gradient descent, the depth of trees in a random forest, or the number of hidden layers in a neural network. Unlike model parameters, which are learned during training, hyperparameters are set prior to the learning process and remain constant during model training. The necessity of **tuning hyperparameters** arises from the fact that different hyperparameter values can lead to vastly different model behaviours. Proper tuning can enhance the model's ability to generalize from the training data to unseen data, optimizing performance metrics such as accuracy, precision, and recall. However, this tuning process involves a delicate balance. Over-tuning on the training set can lead to overfitting, where the model becomes too tailored to the training

data, losing its ability to generalize well to new data. Finding the optimal set of hyperparameters often involves experimentation and iterative tuning. Techniques like grid search, random search, and Bayesian optimization are commonly used to systematically search for the best hyperparameter values in a given range or space.

This is where the **validation set** comes into play. It acts as a proxy for the test set, allowing for the evaluation of hyperparameter efficacy without compromising the integrity of the test set, which should remain untouched until the final model evaluation. The validation set, typically a subset of the training data, is used to assess model performance under various hyperparameter settings. Techniques like cross-validation further enhance the robustness of this process by systematically rotating different subsets of the data as the validation set, ensuring a comprehensive evaluation across the entire training dataset.

The key reason for reserving the test set until after hyperparameters have been tuned, is to maintain an unbiased assessment of the model's performance on completely unseen data. The test set serves as the final arbiter of model generalization, providing an objective measure of how well the tuned model is expected to perform in real-world applications. By rigorously tuning hyperparameters on the validation set and reserving the test set for the final evaluation, practitioners can ensure both the effectiveness and the integrity of the model's predictive capabilities.

More specifically in the case of the **k-means algorithm**, in order to tune the number of clusters k , it is common to run the k-means algorithm for a range of different k values and monitor the clustering outcome, typically by examining the within-cluster distance metric. By plotting k against the within-cluster distance, we can identify the optimal k at the "elbow point" (see figure below) where the rate of decrease slows down. We select this "elbow" point since it represents a point of diminishing returns where increases in k lead to smaller gains in model improvement, indicating an optimal balance between cluster compactness and the number of clusters.

It is also important to monitor whether the algorithm starts creating groups with no actual meaning. In our case for example, more than 5 clusters would arise. It is highly likely that they do not have any actual meaning.

Another factor to consider is the **initialization of the centroids**, as the final solution can heavily depend on this. Random centroid initialization may lead to different outcomes, so it is advised to assess multiple runs of the algorithm. However, there are other initialization schemes, which can avoid local optima and speed up the iterative process. For example, the first centroid can be random and subsequent centroids are chosen by the model based on a probability proportional to the distance from the closest existing centroid. This ensures that the centroids are as far apart as possible, covering the complete data space.

4.7 Advanced learning strategies

4.7.1 Semi-supervised learning

In this last subsection of the module, we will explore several **advanced learning strategies**. Firstly, let's start with semi-supervised learning.

Semi-supervised learning is a machine learning paradigm where models are trained on a combination of labelled data and unlabelled data. The goal is to leverage both types of data to improve the model's performance, especially when labelled data is scarce or expensive to obtain.

Semi-supervised learning is a machine learning approach that harnesses the power of both **labelled** and **unlabelled** data to train models. While supervised learning relies solely on labelled examples for training, and unsupervised learning explores patterns within unlabelled data, semi-supervised learning bridges the gap by effectively combining these two types of data. It represents a valuable paradigm in situations where obtaining ample labelled data is challenging, costly, or impractical. In a semi-supervised learning setup, a portion of the training data is labelled, meaning each example has a corresponding target or output. The remaining data is left unlabelled, lacking explicit annotations. The aim is to leverage the knowledge contained in the labelled examples to enhance the model's ability to make predictions on the unlabelled data. Semi-supervised learning serves as a bridge between the realms of supervised and unsupervised learning, offering innovative solutions to address their unique challenges.

In contrast to pure unsupervised learning, which primarily seeks to uncover hidden patterns within unlabelled data, semi-supervised learning is a versatile technique capable of addressing a **wide array of machine learning problems**. These encompass classification, regression, clustering, and association tasks, making it applicable to a diverse range of real-world scenarios.

What sets semi-supervised learning apart from traditional supervised learning is its ability to extract valuable insights from relatively **small quantities of labelled data** while efficiently utilizing **vast volumes of unlabelled data**. This duality is particularly advantageous for several reasons. First, it significantly reduces the need for manual annotation, which is both time-consuming and costly. By making the most of the readily available unlabelled data, semi-supervised learning alleviates the burdensome process of acquiring extensive labelled datasets. This cost-effective advantage is particularly significant in domains where obtaining high-quality labelled data can be a formidable challenge.

Additionally, semi-supervised learning **minimizes data preparation time**, as the emphasis is on enhancing the model's performance through iterative exposure to unlabelled data. This not only streamlines the model development process but also allows for quicker adaptation to real-world complexities. One of the fundamental benefits of semi-supervised learning is that it allows models to generalize more effectively. While labelled data provides clear insights into how certain inputs relate to desired outputs, unlabelled data introduces diversity and real-world complexity. By learning from both types of data, models become more robust, adapting to a wider range of situations and object variations. This enhanced generalization is particularly valuable in scenarios where the model encounters previously unseen examples or faces variations in the data distribution.

In summary, semi-supervised learning offers an innovative and pragmatic approach by effectively combining the strengths of both supervised and unsupervised learning techniques. It's a versatile methodology suitable for various machine learning problems, while simultaneously reducing the overhead associated with manual annotation and expediting the data preparation process.

Active learning

Active Learning is a technique within the realm of semi-supervised learning that emphasizes a strategic and iterative process for **selecting the most informative or uncertain data points** from the unlabelled dataset. These selected data points are then **manually labelled** and added to the training data, making the model more effective. This approach, situated within semi-supervised learning, is particularly well-suited for scenarios where acquiring extensive labelled data is resource-intensive or time-consuming. By actively selecting and labelling the most relevant data points, Active Learning aims to minimize the labelling effort required to train a robust and accurate machine learning model. This not only makes it

a cost-effective method but also ensures that the model learns efficiently from a limited set of labelled examples, effectively bridging the gap between fully supervised and unsupervised learning paradigms.

In an active learning scenario, the model initially trains on a small set of labelled data, where each data point is paired with its corresponding target. However, instead of passively learning from this labelled dataset, active learning takes a proactive approach. It systematically selects data points from the unlabelled pool that are deemed most informative or uncertain, aiming to maximize the learning benefit with minimal labelling effort.

Example: Consider a scenario in **seizure detection with wearables**. The continuous monitoring of the patients with wearable EEG results in a big amount of data. The neurologists could annotate the whole amount. The active learning process involves identifying the instances where the model is uncertain or less confident in its predictions. This uncertainty can be based on various factors, such as subtle changes in physiological signals or unusual patterns of movement. Once the model flags uncertain segments of data, the wearer, or a medical professional overseeing the data collection, can review these segments to confirm or correct the model's predictions. This feedback loop allows for real-time improvement of the model's performance.

Active learning intervenes in this context. Rather than labelling all images manually, which would be impractical, the active learning process initiates with a small set of labelled images. These initial labels can correspond to common and well-understood cases. The model is then tasked with identifying the most informative or challenging images within the larger pool of unlabelled data. For instance, the model might select an X-ray image with a subtle anomaly that could be indicative of a rare condition. This is a data point that is expected to enhance the model's ability to identify more difficult EEG segments. By focusing on the most informative cases and labelling them, the model becomes increasingly proficient in recognizing a wide range of conditions, including those it has not been explicitly trained on.

The iterative nature of active learning ensures that the model continuously refines its understanding, adapting to evolving clinical scenarios. As more labelled data is incorporated, the model's diagnostic accuracy and generalization improve, ultimately benefiting patient care by facilitating earlier and more accurate disease detection.

In summary, active learning, as a form of semi-supervised learning, introduces an intelligent and cost-effective approach to machine learning in healthcare. Its ability to proactively select the most informative data points from vast pools of unlabelled data streamlines the model development process while significantly reducing the labelling effort. This not only accelerates the model's performance improvement but also ensures that it can adapt to the dynamic and intricate landscape of medical data, making it a valuable asset in the healthcare industry.

4.7.2 Weak supervision

Weak supervision is a machine learning paradigm where instead of relying solely on fully labelled data, various forms of noisy, incomplete, or imprecise supervision signals are utilized to train models.

Weak supervision is a strategic approach that **addresses the challenges associated with labelling data**, which can often be expensive, time-consuming, and prone to inaccuracies, similarly to semi-supervised machine learning. Weak supervision, as the name suggests, acknowledges that the labels provided for training data may not be perfect or complete, and it seeks to make the most of this less-than-ideal supervision.

In traditional supervised learning, models are trained on well-annotated, high-quality labelled data. However, in many real-world scenarios, obtaining such pristine labelled data is a formidable task. This is where weak supervision comes into play. Weak supervision leverages **sources of (weak) labels** that are less reliable or less comprehensive than traditional, expert-curated labels. These sources can include:

- **Noisy labels** are labels that contain errors or inaccuracies. These errors can be introduced by annotators, labelling tools, or external factors. For example, in a dataset of images, some images may be incorrectly labelled with the wrong category.
- **Incomplete labels** indicate that only a portion of the data has been labelled, and some data points are left unlabelled. This could occur in situations where it's not feasible to label every data point. For example, in a sentiment analysis task, only some customer reviews may be labelled for sentiment, while others remain unlabelled.
- **Partial labels** provide information about one aspect of the data but not the full information. For example, in a medical image dataset, some images may be labelled with the presence or absence of a particular symptom, but other symptoms or conditions are not labelled.
- **Ambiguous labels** indicate that the true label is unclear or could correspond to multiple possible categories. For example, in medical text-classification different symptoms classification may be labelled with multiple topics, making it ambiguous as to which topic is the most relevant. For example, a note mentioning "chest pain" might be ambiguously labelled with both "angina" and "heart attack" as potential conditions.
- **Heuristic labels** are generated using rules, heuristics, or domain knowledge rather than manual human annotation. Suppose you're working with a dataset of clinical notes, and a rule-based system is used to automatically label notes based on the presence of specific keywords, medical terminology and context. These rules might involve looking for specific keywords, medical codes (e.g., ICD-10 codes), or phrases associated with diseases. For example, you might have rules like "if the text mentions 'diabetes' or 'hyperglycaemia,' label it as 'Diabetes Mellitus.'
- **Proxy labels** are labels that indirectly represent the true labels. For example, in a dataset of wearable device data monitoring patients' activity and vital signs, the presence of intense physical activity may be used as a proxy label for estimating stress levels, as stress is often associated with increased heart rate and activity.

Weak labels are often used in weak supervision or weakly supervised learning settings, where the goal is to make the most of the available, albeit imperfect, labelling sources. The challenge is to develop machine learning models that can effectively learn from and generalize based on these less reliable labels. Techniques like active learning, self-training, and multi-instance learning are employed to handle and improve models trained with weak labels. These approaches aim to leverage weak labels to the fullest extent possible while maintaining acceptable performance and minimizing the need for strong, manually verified labels. By aggregating information from multiple, albeit weak, sources of supervision, the weak supervision framework trains more accurate models to predict different outcomes of interest. While the individual sources may be prone to inaccuracies or have limitations, the combined knowledge provides a more comprehensive view, allowing the model to make more informed predictions.

The key distinction between weak supervision and semi-supervised learning lies in the nature of the data used for training. Weak supervision does not rely on a combination of labelled and unlabelled data, as is the case in semi-supervised learning. Instead, it focuses on **making the most of weak or noisy labels** from various sources. In essence, weak supervision places greater emphasis on coping with

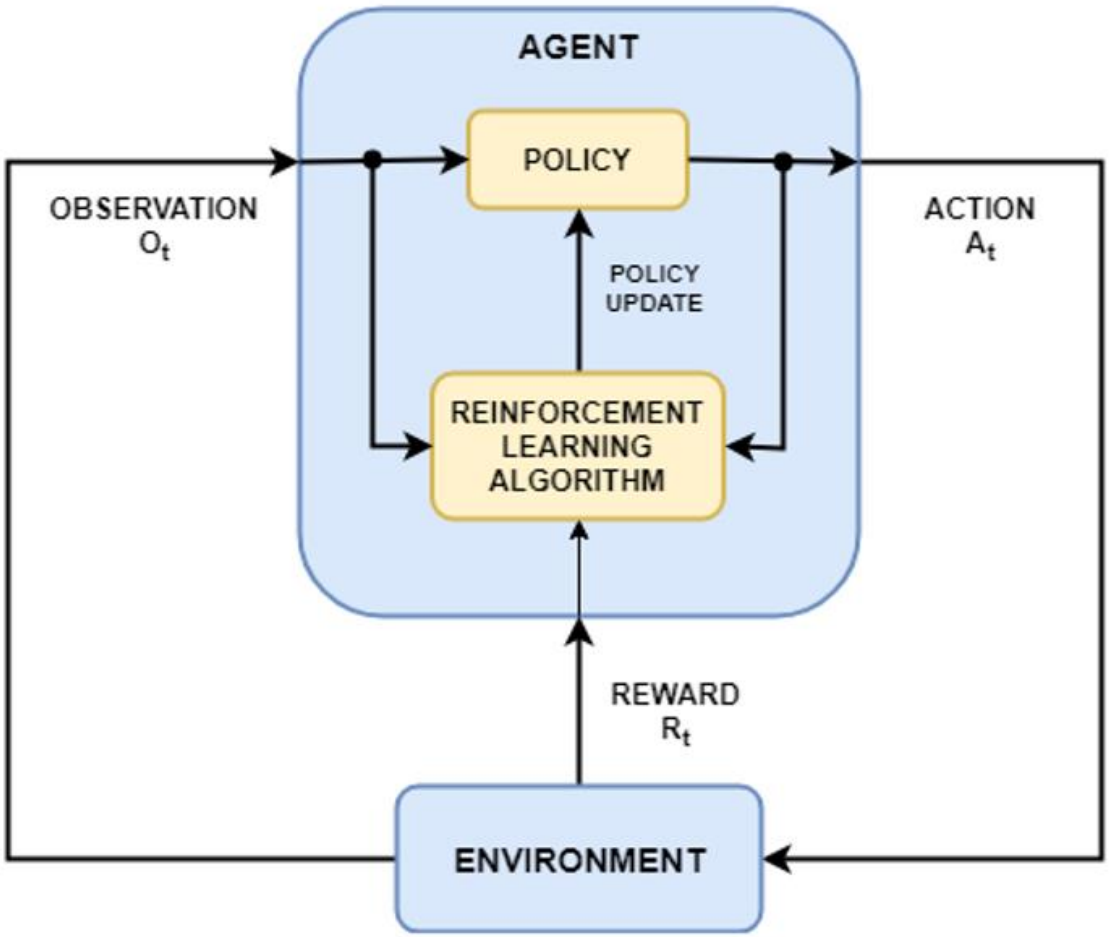
imperfect labels, seeking to optimize the knowledge derived from multiple weak sources. Semi-supervised learning, on the other hand, aims to leverage the abundant but unlabelled data to enhance the model's performance by learning from both types of data.

4.7.3 Reinforcement learning

Reinforcement learning (RL) is a subset of machine learning that has gained significant attention and relevance across a wide range of domains, including healthcare. At its core, RL focuses on **decision-making processes in dynamic environments**, where an agent learns to make a sequence of actions in order to achieve specific objectives. Unlike supervised learning, where algorithms are trained on labelled data, and unsupervised learning, which discovers hidden patterns in data, RL relies on a **trial-and-error** approach, which makes it particularly well-suited for scenarios where optimal decisions are learned through experience and interaction.

Reinforcement learning is characterized by several fundamental components:

- **Agent**: The decision-maker or learner in the environment. In healthcare, this agent can be a robot, an AI model, or a medical practitioner making treatment decisions.
- **Environment**: The external system with which the agent interacts. In healthcare, the environment includes the patient's health condition, medical resources, and various contextual factors. For more information about “agents” and “environments” have a look in Module 2.
- **State (S)**: A representation of the current situation or configuration of the environment. In healthcare, the state could encapsulate patient data, symptoms, and available treatments.
- **Action (A)**: The set of actions that the agent can take to influence the environment. In a healthcare context, actions might include prescribing medications, recommending treatments, or scheduling diagnostic tests.
- **Policy (P)**: The strategy or mapping from states to actions, which guides the agent's decision-making process. In healthcare, a policy defines the treatment plan or intervention strategy.
- **Reward (R)**: A numerical signal that indicates the immediate benefit or cost associated with an action taken by the agent. In healthcare, rewards can be based on patient outcomes, treatment costs, or patient satisfaction.
- **Value Function (V)**: The expected cumulative reward that an agent can achieve from a given state following a specific policy. It helps the agent evaluate the desirability of different states.



Reinforcement learning in healthcare

Consider the application of RL in **personalizing the treatment** of chronic diseases, such as **diabetes**. Managing diabetes requires continuous monitoring, medication adjustments, and lifestyle modifications, which makes it an ideal domain for RL. Diabetes management is a complex and dynamic process, where blood glucose levels fluctuate due to various factors such as diet, activity level, stress, and medication efficacy.

This is where the RL agent comes into play. Imagine a **diabetes management app** on the smartphone of the patient, which serves as the agent. The environment it interacts with the patient's body and the various factors affecting his/her health. The app's RL algorithm considers the current state – including recent meals, blood glucose level, and the time since last medication dose. The agent has a set of actions at its disposal. These actions include recommending insulin dosage adjustments, suggesting dietary choices, proposing activity levels, or maintaining the current treatment plan. The RL algorithm's policy guides the app in deciding which action to take in response to current state.

Now, the crucial element in this system is the **reward mechanism**. After each action is taken, the app receives a reward signal based on the impact of that action. If blood glucose levels stay within the target range and no adverse effects are experienced, the app receives a positive reward. However, if blood glucose levels deviate significantly from the desired range or side effects are encountered, the app receives a negative reward. This reward system serves as the feedback loop that helps the RL agent learn and adapt its policy over time. As the agent continually interacts with health data and observes the outcomes of its recommendations, it strives to maximize the cumulative rewards it receives.

Essentially, the app aims to provide to the patient personalized treatment strategies that optimize blood glucose control and overall health.

Reinforcement learning in healthcare, exemplified by this diabetes management system, promises personalized and adaptive care for patients. It can help patients maintain their health with minimal disruptions and a higher quality of life. While RL in healthcare holds great promise, it also faces challenges like ethical considerations, interpretability, and ensuring patient safety. Nevertheless, it is a testament to how AI can enhance healthcare by creating learning systems that adapt and optimize care for individuals.

Reinforcement learning in healthcare offers numerous potential benefits. It can lead to personalized treatment plans, efficient resource allocation, and improved patient outcomes. However, it also presents challenges related to the interpretability of learned policies, ethical concerns, and the need for extensive real-world testing to ensure safety and effectiveness.

5 AI in action: CSI Leuven- Bloodstain pattern analysis

5.1 Welcome to Module 5

Welcome to Module 5. This is the first of several "**AI in action**" modules, where we embark on a fascinating journey through real-world applications of AI in healthcare. In the "AI in action" modules, we shift our focus from theoretical concepts to practical use cases, allowing clinical and technical experts to share their insights and experiences with AI applications in the healthcare field. Through a series of illuminating use cases over the next modules, we will explore how AI is making a tangible difference in healthcare.

Why the "AI in action" modules matter

These modules about real-life use cases are particularly significant as they bridge the gap between theory and real-world impact. While understanding the theoretical foundations of AI is crucial, seeing how these principles translate into practical solutions is equally vital. By delving into the experiences and insights of experts who have developed or utilized AI applications in healthcare, you will gain a nuanced understanding of the transformative potential of AI in the field. Each use case offers a unique perspective on the practical challenges and opportunities that AI brings to healthcare.

This module is about bloodstain pattern analysis. Discover how AI can be applied in forensic investigations through interview clips with an expert in this field.

Learning goals

- Experience the contribution and limitations of AI in real-world applications for the case study of bloodstain pattern analysis.
- Explain the physical aspects leading to the ability of estimating a point of origin from bloodstain shape, and multiple bloodstain trajectories.
- Restate the step by step execution of an active shape model, as a bloodstain segmentation technique.
- Compare and explain the errors of a regression model built to estimate impact angle from bloodstain shape.
- Identify optimization problems in the context of bloodstain pattern analysis.
- Describe broader implications and limitations in this use case.

5.2 Context and (clinical) societal impact

In the cozy living room of the “AI” family household, a typical Sunday evening transforms into a family ritual of shared entertainment. Gathered around the television, each member selects their favourite spot on the couch, ready for an evening of Netflix indulgence. Thanks to the family's shared Netflix profile, tailored to their collective preferences, the recommendation system expertly curates a list of detective shows reminiscent of the popular **Crime Scene Investigation (CSI)** series. As the screen comes alive with gripping crime scenes and intriguing plot twists, the family find themselves captivated by the world of forensic investigations. Thanks to the Netflix recommender system, they recently stumbled upon "Dexter," a detective series that delves into the intricacies of **bloodstain analysis**.

Recommendation system is an AI algorithm designed to predict and suggest items of interest to users based on their preferences, behaviours, or historical interactions within a platform. It helps users discover content they are likely to enjoy, enhancing their overall experience. In the context of Netflix, the recommendation system analyses a user's viewing history, ratings, and other behaviours to

generate personalized suggestions. For instance, if a viewer frequently watches detective shows like CSI, the system will recommend similar crime dramas, such as "Dexter," creating a tailored and engaging content experience for the user.

"**Dexter**" is a television series that revolves around Dexter Morgan, a forensic blood and bloodstain pattern analyst for the Miami Metro Police Department by day and a vigilante serial killer by night. Dexter's unique code compels him to target criminals who have escaped justice, ensuring that they face the consequences of their actions. As he navigates his dual identity, the show explores themes of morality, justice, and the complexities of human nature. "Dexter" gained acclaim for its intense storytelling, psychological depth, and the portrayal of its morally ambiguous protagonist.

In this evening's episode, Dexter Morgan is analysing an **impact pattern**, a distinct pattern of bloodstains as shown in the figure on the right. Such a pattern is typically caused by the forceful contact between a blood source and a target surface. This can occur when a person is struck or injured, causing blood to be expelled and create distinct patterns on surrounding surfaces (e.g. the walls in a room). The size, shape, and distribution of these stains provide forensic analysts with valuable information about the nature and dynamics of the impact, aiding in the reconstruction of events at a crime scene. Impact bloodstain patterns are crucial elements in forensic investigations, helping investigators understand the sequence of actions and contributing to the overall analysis of a violent incident.

Stringing or **trajectory analysis** is a common technique used to determine the point of origin of impact patterns. This involves using strings or rods to connect blood droplets within a pattern, helping forensic analysts identify the location from which the blood was projected. Specifically, as shown in the figure below, by analysing the elliptic shape of individual blood stains, and the fan shaped, radial pattern of the blood stains in group, we can estimate the point of origin of the blood and thus the location where potentially the victim was struck by a blunt object.

Witnessing Dexter's detailed examination of the impact pattern, it becomes evident that this process demands considerable expertise, manual effort, and consumes a significant amount of time. Consequently, Aisha contemplates the potential for AI to streamline these labour-intensive and potentially perilous tasks, sparking her interest in the prospect of **AI assisting in forensic investigations**. Fortunately for Aisha, we have our CSI Leuven expert, Philip Joris, available to elucidate the role of AI in assisting with these intricate blood investigations.

"Hemo what? HemoVision, which is loosely translated as seeing blood." It turns out our specialist has an AI-based solution for analysing impact patterns. Let's delve into the data, methodologies, and challenges associated with this AI system.

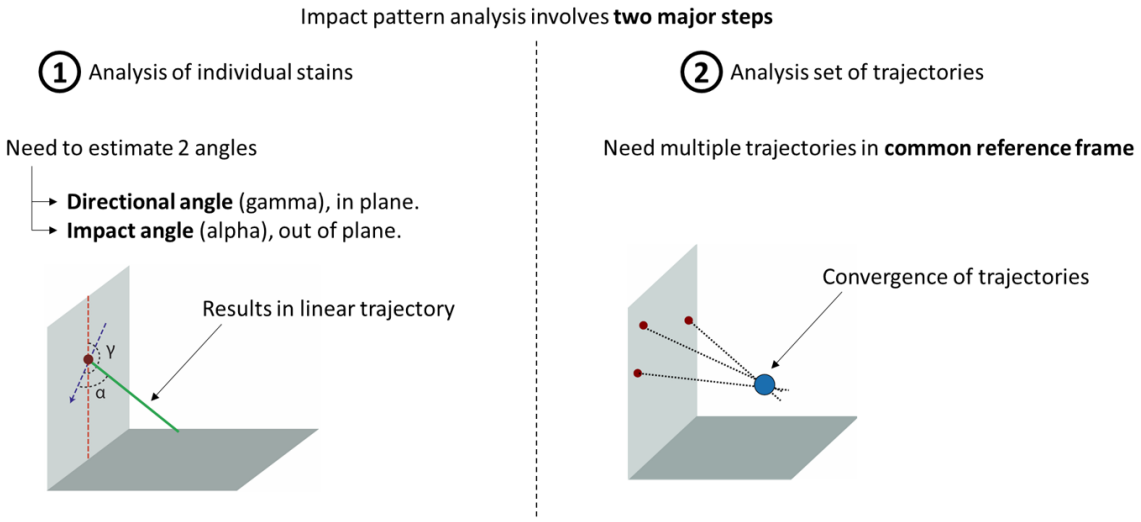
5.3 Data

Spatter patterns form when force is applied to liquid blood, creating droplets with various shapes based on speed and angle. **Impact pattern analysis** involves two major steps.

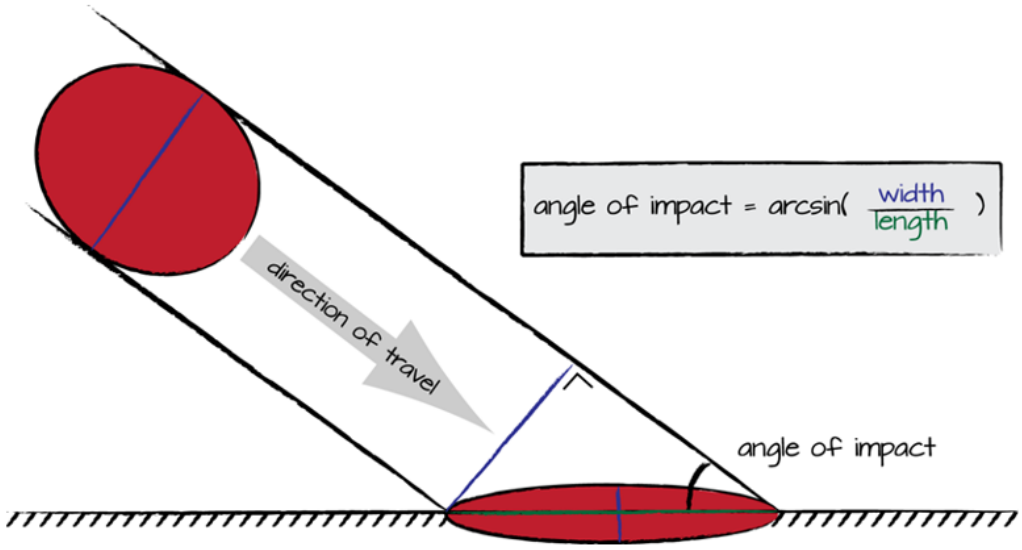
1. The first, **examining individual stains**, aims to estimate two angles (the directional angle and the impact angle) related to the linear trajectory of a blood droplet before impact.
2. The second step involves **estimation of the area of origin** by analysing sets of trajectories, combining linear paths of multiple droplets.

This latter step is an optimization problem, where no learning data is used. However, in the former step, image data (2D photography) from individual bloodstains enables the estimation of the necessary

angles and therefore trajectories as input to the second step. Therefore, the data to focus and work on, are 2D images of bloodstains.



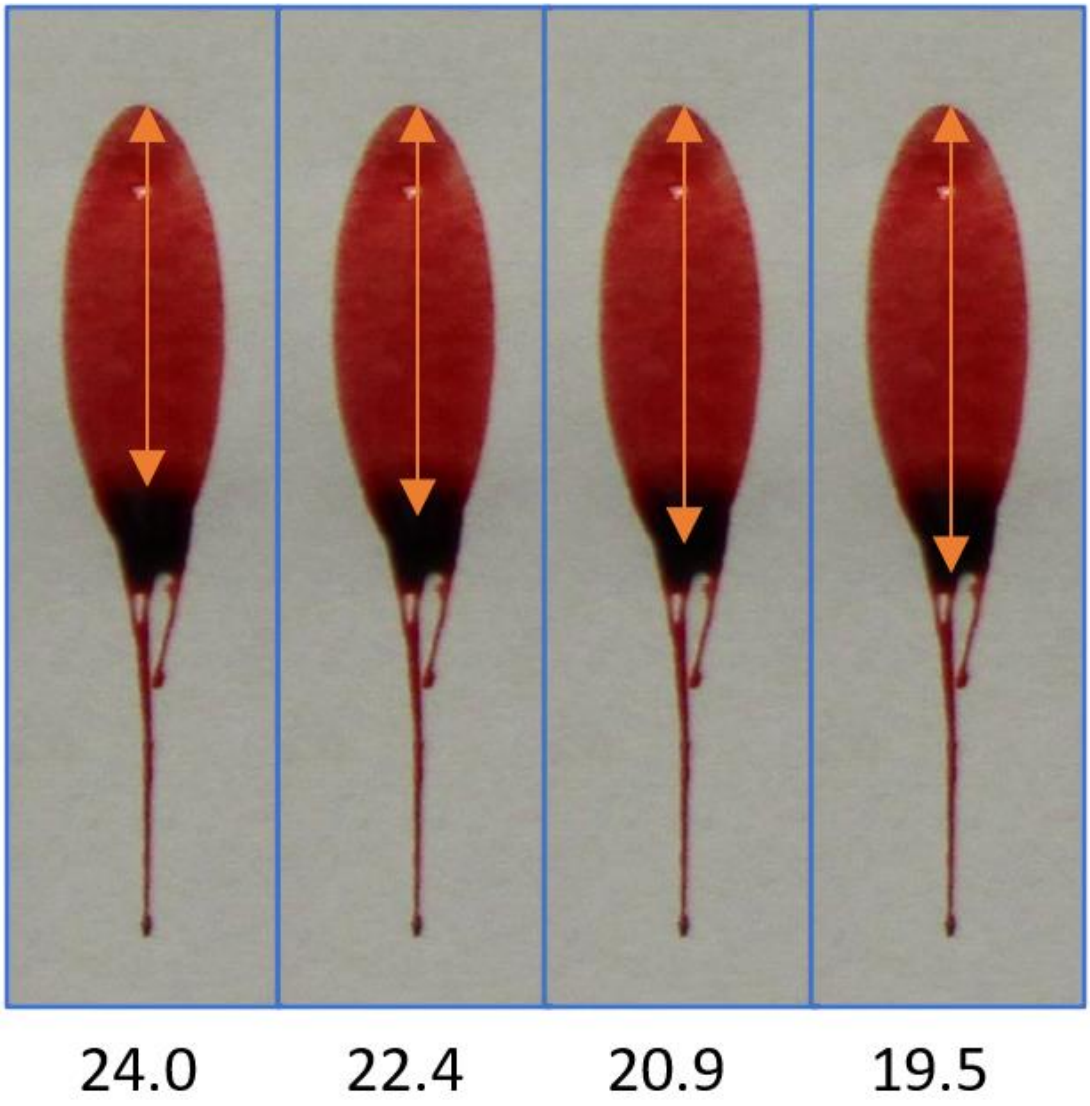
For the first angle, or **directional angle**, typically a vertical line (see red dotted line, figure above) is taken as a reference on the wall. Then the angle the bloodstain makes with this line is recorded as the directional (gamma) angle. For the second angle, or **impact (alpha) angle**, analysts look at the bloodstain's shape, because this indicates the angle at which the impact took place as seen in the figure below:



Determining the impact angle is crucial, and ideally, bloodstains would assume perfect elliptical shapes post-impact, simplifying the computation of the impact angle. However, **real-life** situations are seldom ideal and often involve noise. Similarly, bloodstains rarely form perfect ellipses, as evident in the provided examples above and below.

- Notably, the blood distribution within the stain is uneven, creating **colour gradients**.
- Additionally, stains exhibit **flairs and tails** depending on the impact angle.

Due to these imperfections, it becomes difficult to pinpoint the length and the width of the ellipse from which the impact angle can be computed. In the example below, you can see how small changes in the ellipse length affect the impact angle estimated. When faced with these imperfect bloodstain images as data for extracting the impact angle, it will be interesting to see how Philip's AI based method addresses these imperfections.



5.4 AI in action

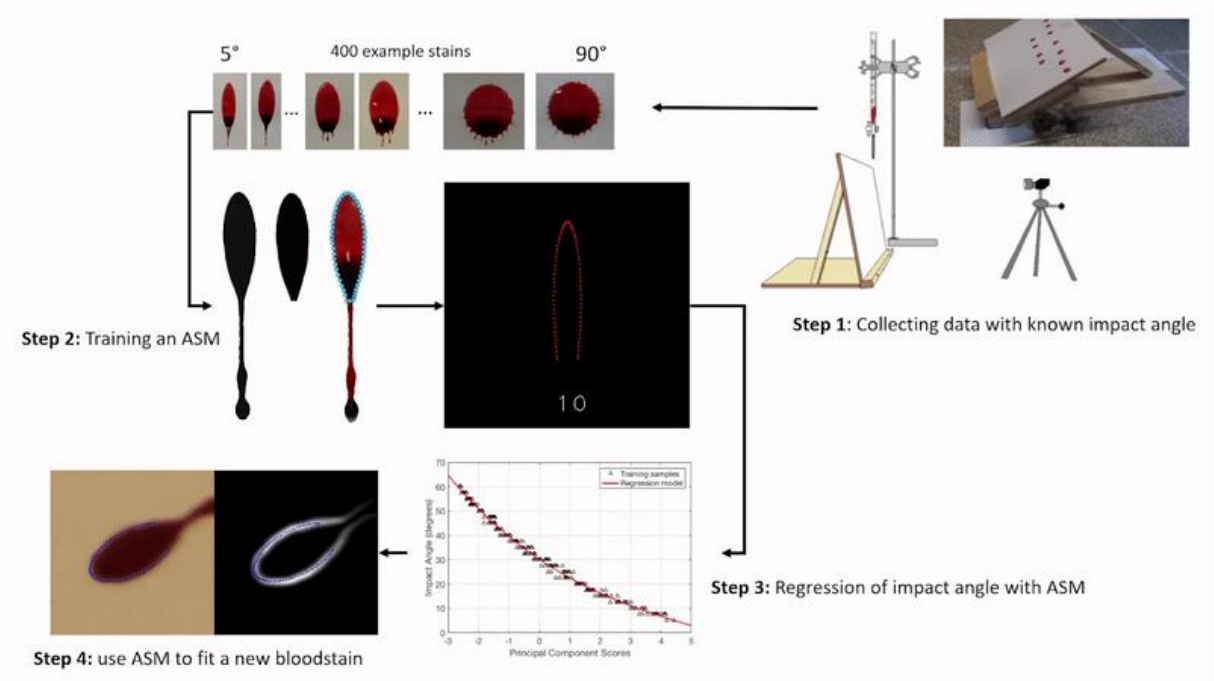
Part one: Individual bloodstain analysis

As mentioned earlier, the analysis of impact patterns involves two primary steps. Focusing on the initial step, which is the analysis of individual bloodstains and the estimation of their trajectory, Philip utilizes an **active shape model (ASM)** in conjunction with a polynomial regression. In essence, instead of assuming a heuristic elliptic model, which only works on theoretical data, Philip aims to learn bloodstain shape (ASM) and its relationship to the impact angle (regression) directly from the image data.

Active shape model (ASM) is a statistical model used in image analysis to capture the variability and shape characteristics of objects within images. It operates by iteratively adjusting a shape model to fit

the target object in an image. The model learns from a training set, allowing it to generalize and adapt to variations in shape and appearance.

In the context of impact pattern analysis, Philip employs an active shape model to enhance the accuracy of estimating trajectories by iteratively refining the model to align with the shapes of bloodstains, which involves the following four steps: 1) collection of training data, 2) extraction of shapes from images and training the ASM, 3) learning the relationship of shape coded in the ASM and the impact angle, and 4) aligning the ASM with new bloodstain images for inferring the impact angle. Let's explore each step separately.

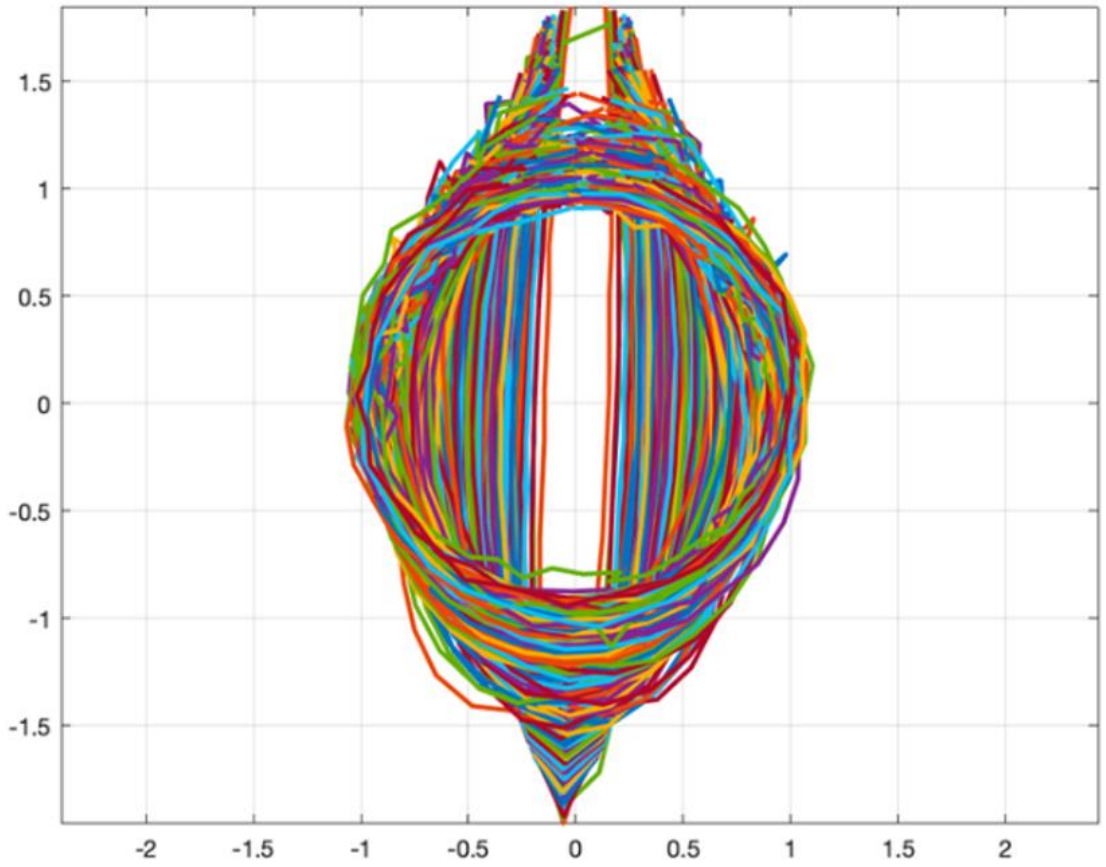


Step 1: Collection of training data

In the first step, Philip sets up a kind of stand with two wooden plates and a bar in between. The plates can be adjusted to different angles, and he places paper on them to catch blood droplets. He then uses a pipette filled with pig's blood to make the droplets, keeping the pipette at a fixed height. This way, he makes sure to get different bloodstains with known angles. Philip records impact angles from 5° to 90°, measuring them with a tool called a goniometer. For each angle, he creates 12 stains. After removing some imperfect stains, he ends up with a collection of 400 bloodstains. He takes photos of them right away to avoid them drying out or changing shape. He uses a Canon EOS D600 camera, making sure it is straight to the paper to get accurate pictures. This whole setup helps Philip understand how blood behaves when it splashes from different angles.

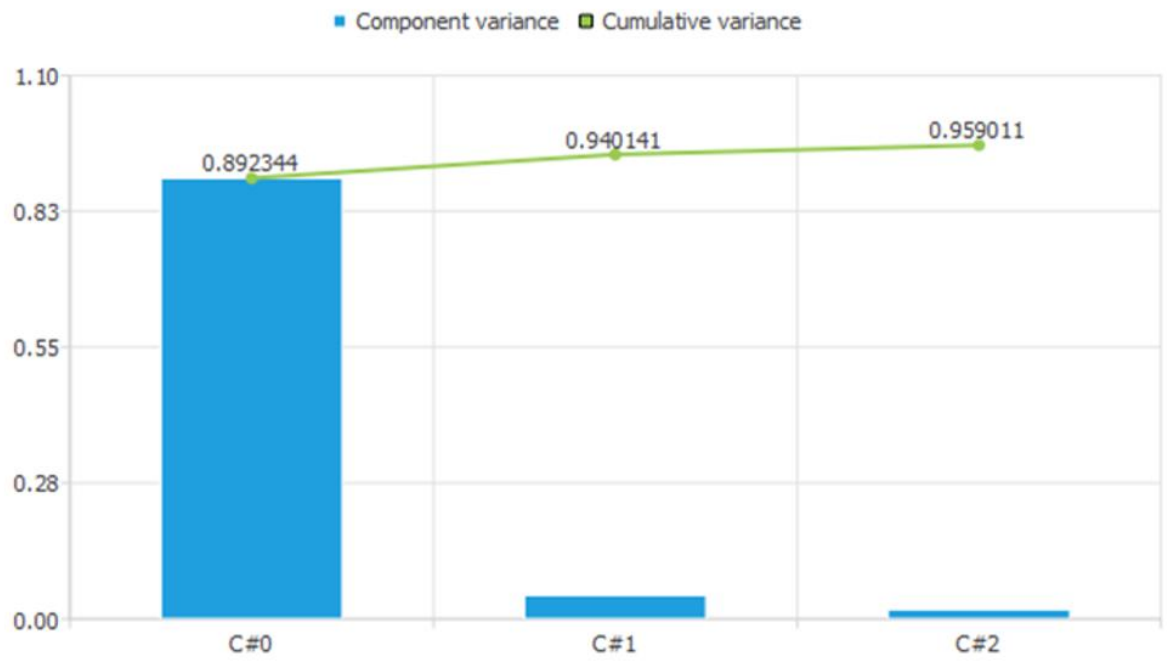
Step 2: Extraction of shapes from images & training the ASM

In the second step, Philip trains the active shape model using the gathered training data. In this process, he concentrates on outlining each stain in his collection, focusing on the contour that holds crucial shape information. This step is called **image segmentation**, where the bloodstain is separated into the foreground (black) and the rest of the image into the background (white), resulting in a black and white image instead of a coloured one. In this simplified form, the contour or edge of the bloodstain can be easily extracted based on the transition from black to white or vice versa. The contour is then sampled with 100 evenly spaced points, and this process is repeated for all the 400 bloodstains in the training dataset. The following figure shows all the bloodstain contours superimposed onto each other:



With the available contours and their corresponding impact angles, it becomes feasible to develop a regression model predicting impact angle based on contour shape. However, a single contour is high-dimensional, with each comprising 100 points characterized by 2 coordinates (x and y, denoting locations in the XY plane).

Following your correct answer of dimensionality reduction, Philip transforms each shape, consisting of 100 two-dimensional points, into a row-vector of size 200 by appending the y-coordinates to the x-coordinates. Subsequently, he applied Principal Component Analysis (PCA) to identify and rank the principal modes of variation in this 200-dimensional space based on their significance. From the figure below, we can see that the **primary component** is already responsible for 89% of the dataset's variation.



We can see that this primary component signifies the **widening and narrowing** of the stains, very much following the shape variations observed when changing the impact angle in the controlled experiments.

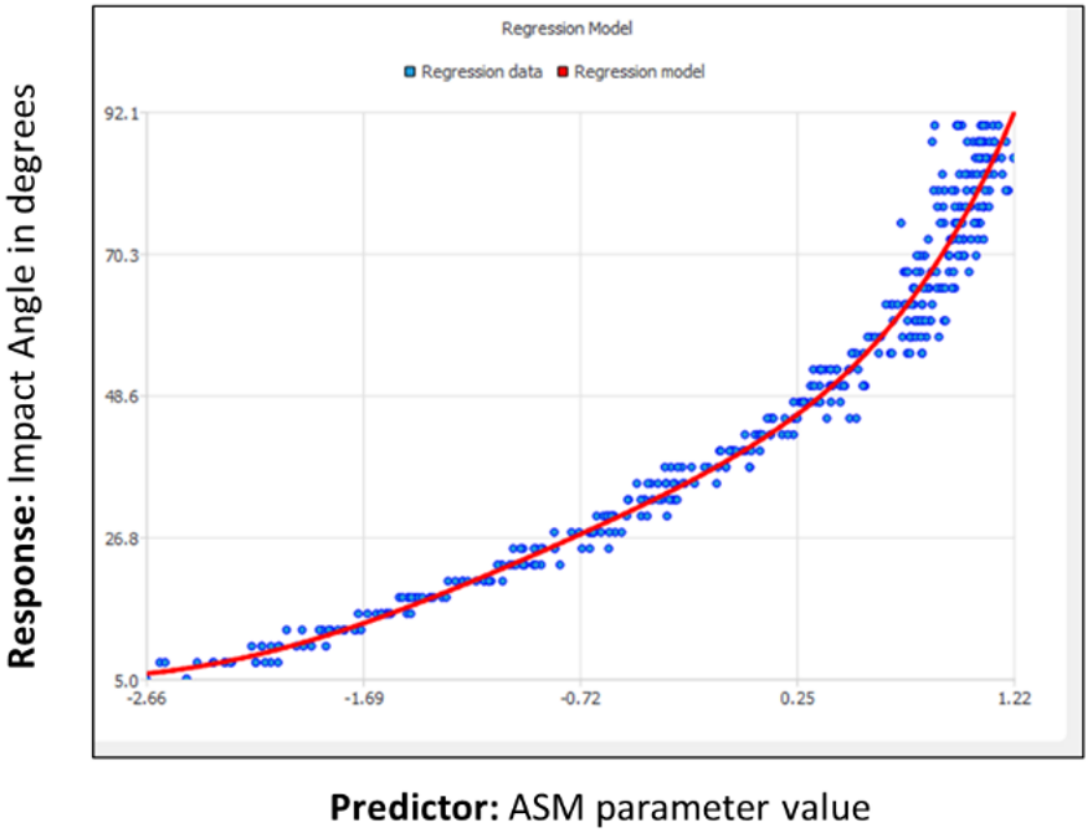
Due to the significant variance explained by the first mode, Philip can safely assume that this component would contain enough information to accurately represent any stain by moving along its principal axis. Since a single principal component is utilized, the model has only one parameter to vary its shape. In other words, with the use of PCA, Philip was able to reduce the dimensionality of his data problem with a factor of 200.

The resulting PCA model constitutes the **Active Shape Model (ASM)**, where each bloodstain contour in the dataset is now expressed with a 1-dimensional ASM parameter value. By assigning a specific value to the ASM parameter, a new bloodstain shape, represented as a contour with 100 connected equidistant 2D landmarks, can be generated. Furthermore, examining the distribution of ASM parameter values in the training data allows for setting boundaries within which the value for generating new shapes must reside, ensuring the creation of plausible shapes. The term "active" in ASM reflects the capability to actively encode and generate new shapes from this lower-dimensional (1-dimensional) representation.

Step 3: Learning the relationship of shape coded in the ASM and the impact angle

In the typical (non-learning based) method of fitting ellipses, we use the ratio of the minor to the major axis to predict the impact angle for a given bloodstain. However, because the ASM Philip obtained is not limited to an elliptical shape (see previous figure), he needs to use a different feature than the ratio of ellipse axis that correlates with the impact angle. As his ASM is controlled by a single parameter (the reconstruction score of a bloodstain's shape along the first principal component), Philip learns a regression to connect this score to the impact angle. For this task, he opts for a **third-order polynomial**, illustrated in the figure below. You can observe the impact angles for all stains plotted against their reconstruction parameter along the first principal component, with the regression curve in red drawn on top. It can also be observed that the relationship between the ASM parameter and impact angle is

curved. This indicates a non-linear relationship between both. In the case the relationship was linear, then one would have observed a straight line instead of a curved line connecting both variables.

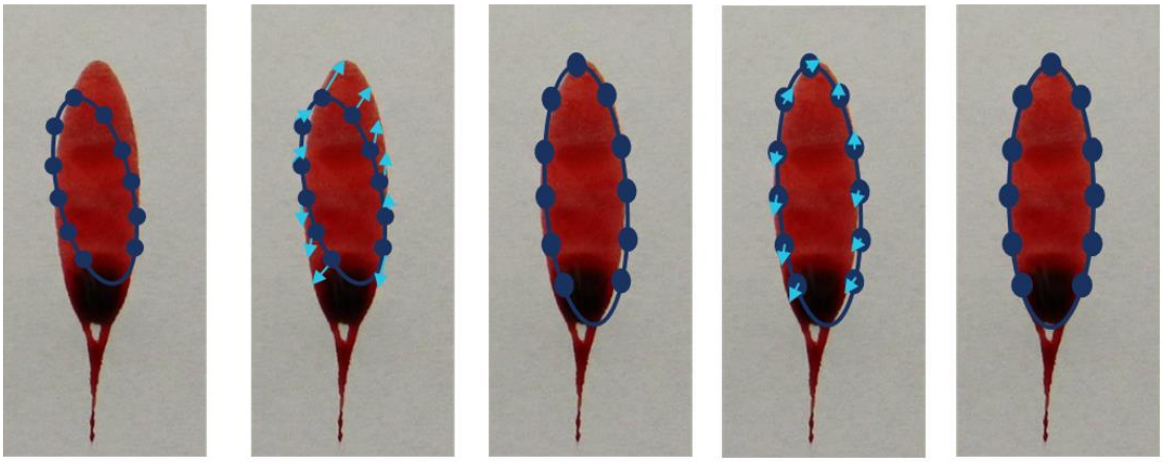


The integration of this regression with the ASM signifies the completion of the bloodstain analysis AI model, enabling Philip to approximate stains by adjusting the ASM along its first principal component. The third-order polynomial becomes instrumental in estimating the impact angle based on the obtained reconstruction parameter.

Step 4: Aligning the ASM with new bloodstain images for inferring the impact angle

In the fourth step, when estimating the impact angle of a new bloodstain image at a crime scene, the goal is to initially fit the Active Shape Model (ASM) model onto the image and then deduce the impact angle from this fit. The main task in the fourth step then becomes: how can we effectively fit the ASM to a new bloodstain image? The solution to this is given by techniques from Module 3: search and optimization.

The process of fitting the ASM to a bloodstain image unfolds iteratively, just like a **local search algorithm**. Following the figure below, the iteration involves a continuous cycle of **updating the current shape of the ASM**, depicted as a darker blue contour with equidistant points, and **determining how each point on the contour should move**, indicated by light blue arrows, to align with the edge of the bloodstain in the image.



The sequence progresses through various steps (see images above from left to right):

1. Initiate a shape from the ASM onto the image
2. Adjust each point on the shape to correspond to the bloodstain edge
3. Update the pose and shape of the ASM
4. Check if additional updates are needed
5. Achieve the optimal fit where the ASM aligns seamlessly with the edge of the bloodstain

Once a perfect alignment is achieved, the optimization process can be concluded. Since this constitutes an **optimization problem**, try and define this problem in terms of its variables, domains, constraints, and objective function. To simplify things slightly, you can ignore the pose correction (which involves a 2D rotation and translation of the current shape) that happens alongside updating the current shape in the ASM.

One possible formulation is as follows:

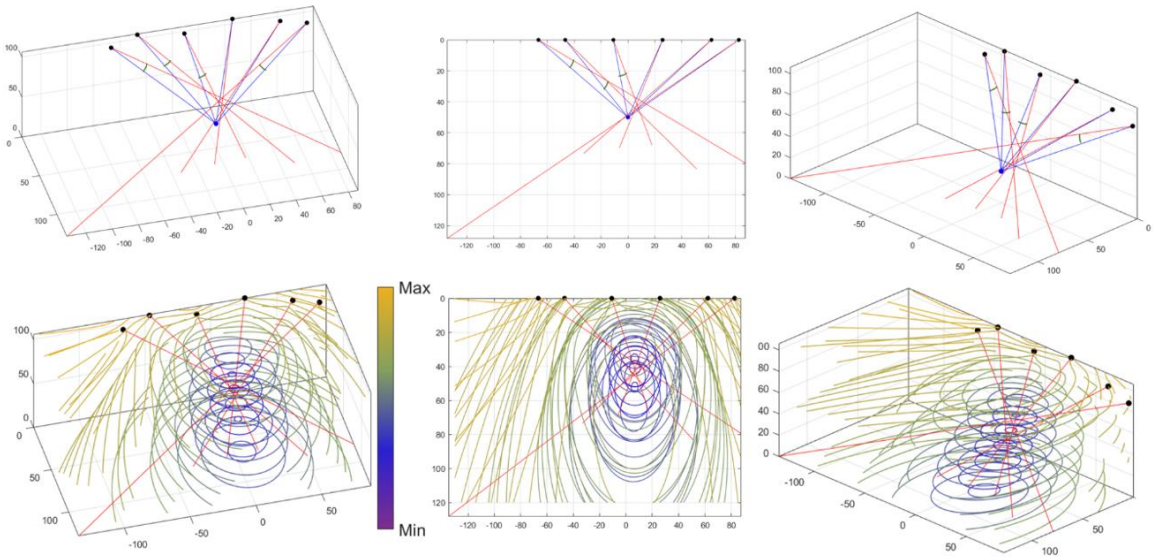
- **Variable**: ***s*** a 1-dimensional variable representing the ASM parameter.
- **Domain**: ***s*** takes continuous values as the ASM parameter.
- **Constraint**: The values of ***s*** are constrained within the range of minimum and maximum reconstruction scores observed in the ASM training data. This constraint ensures adherence to plausible bloodstain shapes.
- **Objective function**: Minimize the discrepancy between the current shape and the desired shape that seamlessly fits onto the edge of the bloodstain in the image. The discrepancy is quantified as the distance from each point on the current shape to the edge or boundary of the bloodstain in the image.

To summarize the initial phase of the pipeline, Philip's AI methodology for analysing individual bloodstains integrates a statistical shape model and a third-order polynomial. This connection links the statistical shape descriptor to impact angles. Through this data-driven approach, variations in stain shapes are learned rather than derived theoretically. The regression component facilitates the implicit modelling of (non-linear) physical interactions in the stain formation process, particularly addressing the limitation that physical interactions often prevent bloodstains from forming perfect elliptical shapes.

Part two: Area of origin (AO) estimation

In the context of an impact pattern, area of origin (AO) estimation endeavours to **determine the 3D location(s)** of the impact(s) responsible for creating the given pattern. This involves, the previous part,

or analysing individual impact stains to extract their impact (alpha, α) and directional (gamma, γ) angles. Utilizing these angles, visual representations of estimated stain trajectories can be generated, as illustrated in the figure below. Each trajectory (red lines) is characterized by a starting point (the stain's coordinate) and a direction. By assessing the convergence (the point in which they come together) of the estimated trajectories, the AO for the impact pattern can be ascertained.



In the figure above, we observe an illustrative crime scene analysis involving six distinct bloodstains, each yielding individual trajectories represented by red lines. These trajectories visually converge to a potential AO. To assess a potential AO (depicted as a blue point), we connect this potential AO with each bloodstain, forming blue lines. Ideally, if the 3D point accurately represents the AO, the blue lines should coincide with the red lines, resulting in zero angles between them. Therefore, by measuring these angles and summing them up, we quantify how closely the 3D point approximates the actual AO. The optimal AO is determined by identifying the 3D point where the sum of these angles is minimal. Mapping these values across the entire 3D search space, as shown in the bottom row of the figure, reveals a distinct landscape highlighting the approximate AO (dark blue contours).

This representation reflects an optimization landscape, and the search for the AO is in fact again an **optimization problem**. To formalize this process, we can define variables, domains, constraints (if any), and the objective function. Try to do this for yourself.

One possible formulation is as follows:

- **Variables**: ***p*** = (x,y,z), a 3D point representing a potential AO.
- **Domains**: ***p*** = (x,y,z), can take continuous values within the 3D search space.
- **Constraints**: None in the initial formulation. However, it is implicit that ***p*** should be within a physically plausible range, e.g. within the room of the crime scene.
- **Objective function**: Minimize the sum of angles between the blue lines (connecting ***p*** with each bloodstain) and the corresponding red lines (representing the estimated trajectories of individual bloodstains).

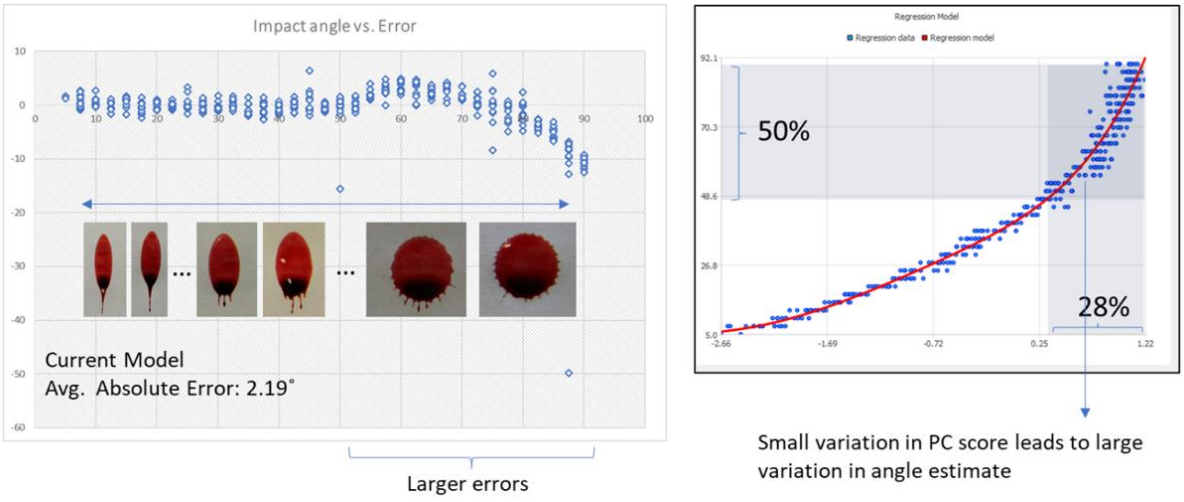
5.5 Evaluating AI

Leave-One-Out validation

Philip clarifies that the two components of the method outlined in the preceding section are assessed independently and through distinct evaluations. The initial component adopts a conventional machine learning approach, where a model is derived from data. In this phase, Philip employs a **Leave-One-Out (LOO)** strategy to assess the model's capability to accurately estimate an impact angle based on the image of a single bloodstain.

These techniques help ensure that a machine learning model generalizes well to new, unseen data and does not overfit to the training set. The choice of the validation technique depends on the characteristics of the dataset and the specific requirements of the problem at hand. In Philip's case, the LOO strategy is advantageous, because it provides a strong estimate of performance, especially for smaller datasets. This is because each data point serves as a test case while the model is trained on the rest. Since the computational workload for training and fitting an ASM is manageable, LOO is the best choice for Philip. However, for models with high computational demands, like many of the current neural network techniques, simpler strategies like a single train-test split or cross-validation with fewer folds may be better options. This is simply because, it otherwise becomes computationally impractical to train a new model, each time one data point is left out as test case.

In the figure below, you can see the **error results for angle estimations** using Philip's model. The overall mean absolute error is impressively low at 2.19˚. However, it is noticeable that this error varies across different impact angles. In other words, the error is not homogenous across the range of possible impact angles. Smaller impact angles, where bloodstain shapes are more elliptic and distinct, show more accurate estimations. Conversely, as impact angles approach 90 degrees and bloodstains become very round, it becomes challenging to precisely estimate the impact angle, resulting in larger errors.



Looking at the results on the left side of the figure below, the ideal scenario is for the dots to align along the zero line, indicating that the estimated and ground truth angles are the same, resulting in zero error. However, for larger impact angles, the estimated angles are consistently underestimated, causing the dots to shift below the zero line.

These errors introduce uncertainty into the impact angle estimations. Philip addresses this uncertainty in a unique way by explicitly incorporating it into the construction of trajectory lines for a bloodstain. In simple terms, through his validation experiment, he learns the variation/spread in error (mo**del uncertainty**) specific to each impact angle, recognizing that errors differ for various impact angles. E.g.

in the figure above, the uncertainty for an impact angle of 20 degrees is plus or minus 4 degrees. For an impact angle of 60 degrees the uncertainty increases to plus minus 15 degrees. Instead of generating a single trajectory line based on the most likely angle, he creates multiple trajectory lines based on different impact angles, all falling within the error spread or uncertainty of his model. In essence, for each bloodstain, a bundle of equally possible, but slightly different, trajectory lines is generated.

Doing this, is a form of **error propagation**: where the uncertainty in measurements or estimates is explicitly considered and incorporated into subsequent analyses or models. In this case, Philip is accounting for the variation in error specific to different impact angles, and by generating multiple trajectory lines within the uncertainty range, he is propagating the potential errors in the impact angle estimations to the next part of the pipeline or the estimation of the area of origin. This helps to provide a more comprehensive understanding of the potential variability in the results.

Impact pattern validation study

To evaluate the system's capability in estimating an area of origin, Philip creates **mock-up crime scenes** and assesses the performance of the proposed approach. The generation of each crime scene follows a consistent procedure. A stool is positioned at a random distance from the target wall(s), and a small piece of plastic is placed on top. A puddle of blood is then carefully collected on the plastic. Philip documents the three-dimensional location of the blood puddle in relation to the target wall before subjecting it to high-speed impact. Then the impact spatter is generated by impacting the blood with a hammer. Photographs, captured with a Canon EOS D600, are then analysed using the software of Philip. In the figure below, you can see a few screenshots of the mock-up crime scenes. In total, 10 such scenes were created.

Through HemoVision, a **34% improvement in AO estimation** is evident compared to a manual investigation conducted by a blood spatter expert like Dexter Morgan. Additionally, a **13% enhancement is observed compared to other state-of-the-art software** utilizing the ellipse model. These findings underscore the effectiveness of the pipeline. Beyond accuracy, the proposed method offers several practical advantages over traditional and alternative virtual approaches. HemoVision is **fully automated**, leading to a drastic reduction in the time needed to investigate a crime scene. This not only minimizes the risk of scene contamination but also reduces the potential biohazard risk, as

less time is spent at the crime scene. Furthermore, the virtual analysis can be **repeated infinitely** since a digital record of the crime scene is created. In contrast, a manual approach can only be executed before a scene is released and cleaned up.

5.6 Challenges

The primary hurdle identified involves addressing **scepticism and inherent resistance** in adopting the technology. To tackle this, a rigorous testing protocol for the AI model is needed, with the results disseminated across multiple channels, including scientific publications and conferences. This approach is not limited to impact spatter analysis but is deemed crucial for the broader field of science and technology in forensic science. Legal acceptance of a scientific method, following standards such as **Frye** and **Daubert**, becomes particularly vital in ensuring credibility and recognition within the legal framework.

The **Frye Standard**, originating from the 1923 case Frye v. United States, dictates that scientific evidence is admissible in court only if it has gained general acceptance in the relevant scientific community. This standard places emphasis on the widespread recognition and acceptance of a scientific technique. In contrast, the **Daubert Standard**, established in the 1993 case Daubert v. Merrell Dow Pharmaceuticals, introduced a more flexible approach. It mandates that scientific evidence must be both relevant and reliable, with judges serving as gatekeepers. The Daubert Standard considers factors such as testability, peer review, error rates, and general acceptance, giving judges greater discretion. While the federal courts shifted to Daubert, some states still adhere to Frye or have variations, impacting the admissibility of forensic evidence based on the chosen standard.

Another challenge that has been recognized is the potential **overreliance on the AI system**. It is crucial to emphasize that an expert is still required to assess and determine which spatter patterns to analyse and which bloodstains are suitable for estimating their impact angles. In simpler terms, if less reliable data from the crime scene is chosen for analysis, the resulting outcomes are also likely to be less reliable. In the words of Philip, the principle applies: "**garbage in - is garbage out**."

5.7 Future perspectives

The incorporation of AI technology in the field of forensic science remains relatively limited, with existing tools serving primarily as aids for criminal investigators in piecing together the events of a crime. Achieving widespread adoption and establishing these technologies as integral components of forensic practices will necessitate dedicated efforts over time. This undertaking demands **significant investments, ongoing research,** and **focused development efforts** to enhance the capabilities of AI tools and ensure their seamless integration into forensic investigations. The path forward involves overcoming challenges and gradually transforming these technologies into indispensable assets within the realm of forensic science.

Looking ahead more specifically for spatter impact analysis, HemoVision aims to improve its impact analysis by better integrating it with the crime scene. This could involve using technologies like 3D room scanners or augmented reality. In the video below, you will get a glimpse of how **augmented reality** can help visualize and understand results. Basically, a video of the crime scene is recorded and enhanced with the **virtual bloodstain pattern analysis** created by HemoVision.

In response to the question "will AI replace future forensic experts?", Philip holds reservations about the prospect of AI completely replacing the role of experts. While the idea is theoretically plausible, the inherent difficulty lies in the AI system's ability to comprehend the broader context of a crime scene and its evidence. Each crime scene presents a distinctive narrative, and the **interpretation of events**

based on evidence is likely to remain a task driven by human insight (and potentially even human emotions). The forensic expert's role remains pivotal in the nuanced interpretation of results and seamless integration with other pieces of evidence, playing a critical role in preventing miscarriages of justice.

Despite the advancements in AI, the human touch in forensic analysis appears indispensable for ensuring comprehensive and accurate investigations.

6 AI in action: Epilepsy detection

6.1 Welcome to Module 6

Welcome to Module 6. This is one of the "AI in action" modules, where we shift our focus from theoretical concepts to practical use cases, allowing clinical and technical experts to share their insights and experiences with AI applications in the healthcare field.

Why the "AI in action" modules matter

These modules about real-life use cases are significant as they bridge the gap between theory and real-world impact. By delving into the insights of experts who have developed or utilized AI applications in healthcare, you will gain a nuanced understanding of the transformative potential of AI in the field. Each use case offers a unique perspective on the practical challenges, risks and opportunities that AI brings to healthcare.

This module is about epilepsy detection with wearable sensors. Discover how AI can be applied in epileptic seizure detection through interview clips with clinical and technical experts in this field.

Learning goals

- Examine case studies and real-world applications where AI has been successfully integrated into the clinical practice, and in this use case: epilepsy management.
- Recognize the importance of accurate and timely detection of epileptic seizures and the role of AI in addressing these challenges.
- Understand the concept of handcrafted features and their significance in the context of epilepsy data analysis.
- Investigate the use of wearable technologies for continuous monitoring of patients with epilepsy and the role of AI in processing and interpreting the data collected from these devices.
- Be introduced to the challenges that the analysis of health data of low quality collected outside of the hospital environment have for AI (but also for clinicians).

6.2 Context and clinical impact

What is epilepsy?

Epilepsy is a neurological disorder characterized by recurrent and unpredictable **seizures**, which are abnormal bursts of electrical activity in the brain. These seizures can vary widely in their presentation and severity, but they all result from disruptions in the normal functioning of neuronal synapses, which are the connections between nerve cells (neurons) in the brain. In epilepsy, there is a disturbance in the balance of excitatory and inhibitory neurotransmission in neuronal synapses. This imbalance can result in the brain becoming overly excitable, making it more prone to generating abnormal electrical activity and seizures.

There are different types of seizures associated with epilepsy. Noah experienced absence and myoclonic seizures:

Absence seizures (Petit mal seizures)

- **Characteristics**: Absence seizures are characterized by a sudden and brief loss of consciousness or awareness. During an absence seizure, the person may appear to be staring blankly into space, with no purposeful movements.

- **Duration**: These seizures are very brief, typically lasting only a few seconds, and are often followed by an immediate return to normal consciousness.
- **Onset**: Absence seizures usually onset in childhood and are more common in children than in adults. They may be mistaken for daydreaming or inattention.

Myoclonic seizures

- **Characteristics**: Myoclonic seizures are characterized by sudden, brief, shock-like muscle contractions or jerks. These jerks can occur in various parts of the body and may affect one side or both sides simultaneously.
- **Consciousness**: Typically, individuals remain conscious during myoclonic seizures. They are often aware of the jerking movements.
- **Onset**: Myoclonic seizures can have various triggers, such as sleep deprivation, flashing lights, or specific medications. They can occur at any age and are associated with conditions like juvenile myoclonic epilepsy (JME).

Treatment of epilepsy

Treatment options for epilepsy may include medication, lifestyle modifications and, in some cases, surgery to control seizures and improve quality of life for those affected by this condition. To optimize the treatment, it is important to know **when exactly a seizure happens** and get an **accurate detection of the events**. This is challenging, like how Noah and his family did not notice that him blanking out was a seizure, which is why they could not register it. In this use case, we will discuss a useful tool for someone like Noah: epileptic seizure detection with wearable devices.

In this use case, we will explore the detection of epileptic seizures with wearable devices.

6.3 Data

In epilepsy, caregivers strive to mitigate risks and enhance the well-being of patients by facilitating rapid diagnoses and appropriate treatment. Nevertheless, the available diagnostic tools for long-term monitoring of treatment response in clinical trials or outpatient settings are limited. Presently, the assessment of seizures relies on costly in-hospital **video-electroencephalography (vEEG)**, short-term (video) EEG monitoring at home, or the use of **seizure diaries**, with the latter being the primary tool. Notably, seizure diaries suffer from low sensitivity and has been reported that patients can just report half of their seizures during vEEG monitoring in the case of focal seizures (Vandecasteele et al., 2020) and less than 10% in the case of absence seizures. Consequently, precise seizure detection and counting are of paramount importance for follow-up care outside specialized environments.

For an extended period, AI models have been employed to analyse comprehensive full-scalp EEG data and construct classifiers for seizure detection. Researchers have explored various methodologies, including feature-based and deep learning approaches. More recently, we**arable seizure detection devices** have emerged for the identification of focal seizures within hospital settings, utilizing machine learning methodologies. In the current use case, we will exhibit the employment of different AI tools for the detection of epileptic events with wearables.

The results presented in this use case are mainly part of Seizeit2 project, which is a large multicentre trial with a focus on clinical validation of a wearable device in people with typical absence, focal impaired awareness, and generalized tonic–clonic seizures (EIT Health: SeizeIT2; clinicaltrials.gov: NCT04284072). The wearable device used is the CE-marked device, **Sensor Dot (SD; Byteflies)**. SD is a discrete, user-friendly wearable that makes use of two behind-the-ear channels to detect seizures. This

device was developed during SeizeIT1 (2016–2019). Written informed consent was obtained from every participant. The ethical committees of all participating centres approved this study.

Data pre-processing and harmonization

In this project, **full-scalp and behind-the-ear (bte) EEG data** from epileptic patients was recorded.

The data from each modality is catalogued with separate devices. The behind-the-ear (bte) EEG data is recorded with the wearable EEG Sensor Dot, from Byteflies. The two devices have slightly different clock speeds, causing time shifts and **misalignment** with the data recorded with the hospital equipment. This becomes an issue since the seizure annotations are done on the full-scalp EEG data. If the data from the two modalities is not aligned in time, the seizure labels will be wrongfully marked on the wearable EEG.

Alignment

The first step in **pre-processing** the data is to **align** the two modalities. For this, we perform a correlation analysis between the bte-EEG channels and a simulated behind-the-ear montage from the full-scalp EEG. For this, we take electrodes close to the behind-the-ear location, mainly T4 and T6 for the right-side channel and T3 and T5 for the left. Consequently, we calculate the cross-correlation between the corresponding channels in the wearable and simulated bte electrodes. The time shift between modalities is the lag in which the cross correlation is highest. Since the time shift is not constant, the data is segmented into one-hour segments. The wearable data is then morphed (squeezed or stretched) to fit the full-scalp data.

Filtering

Another important step in pre-processing is **filtering** the EEG data. We employ a **high-pass filter** with a cut-off frequency of 0.5 Hz to remove the baseline drift, a low frequency noise associated with electrode movement. Additionally, a **low-pass filter** is used to remove high frequency noise. It is known that seizures can be identified in the low frequency range (3-5 Hz), and removing noise that does not contain valuable information for seizure detection can help AI models to classify the data.

Data quality, artifact removal and denoising

There are many sources for **artifacts** when recording biomedical signals. Typical sources of noise are **power line interference** (in the case of data measured with hospital equipment), **electrode movement and impedance changes**.

When considering the difference in hardware between the hospital equipment and the wearable sensors, it is expected that the lower quality of the electrical components affects the signal measured with the wearable device. Most of the baseline drifts and power line interferences are filtered in the pre-processing stage. The presence of artifacts, either caused by capturing muscle activity or abrupt movements causing impedance changes in the electrodes, is harder to remove.

A simple strategy to remove high amplitude artifacts, is to calculate the root mean square amplitude of the data segments, and define hard thresholds for excluding artefactual data. Other ways of removing artifacts include estimating the signal-to-noise ratio, where the ratio of the high and low frequency content of the signal is obtained, and segments with low ratio are removed. It is, however, inevitable to remove all artifacts. In AI, it is important to build robust models that can deal with artefactual data, and produce correct predictions.

Data normalization

Depending on the AI methodology that is used, data normalization may have different impacts.

- In some **feature-based** approaches, it is common to normalize the features instead of the data, ensuring that no single feature disproportionately impacts the results.
- In **neural networks or deep learning models**, the data can be normalized by subtracting the mean to the data and dividing it by the standard deviation, producing a 0-mean signal with standard deviation of 1. This helps the model to converge to the optimal solution.

6.4 AI in action

Selection of the appropriate machine learning model

In AI applications, the model choice depends on many factors, and it is not known which methodology is best. There are advantages and disadvantages to each specific model type. It is possible to divide machine learning methods into two different general groups: **feature-based** and **deep learning frameworks**. The two differ in the feature extraction. The first employs manually engineered features that are fed into a selected classifier. The latter relies on automatically extracting features from the data itself, where the features are learned by the model during training, depending on the classification output (you will learn more about deep learning frameworks in Module 8).

In the experiments of this use case, the best performance was achieved by employing a **feature-based Support Vector Machine (SVM)**, mainly due to the lack of multiple patients (big data). Multiple features are extracted from the two bte-EEG channels. Both time and frequency domain features are extracted, such as number of zero crossings, root mean square amplitude, signal power and sub-band power (Vandecasteele et al., 2020; Swinnen et al., 2021). After normalizing the features, a log-transformation is applied to particular features. Transformations can sometimes help differentiate the different classes in the data. The features are then fed onto the classifier, in this case an SVM. This model maps the data points to a space to maximize the width of the separation (hyperplane) between different classes (see Module 4 for more information).

Feature-based methods have the disadvantage of requiring more computational power to extract the features when compared to deep learning models. Additionally, the features are predefined and manually selected. However, deep learning approaches require large amounts of data to obtain good classification performances. In the epilepsy use case, obtaining seizure data is hard since these are rare events. If we consider the wearable scenario, it is also possible that many seizures that are recorded with the full-scalp EEG are not visible in the bte-EEG due to the unconventional electrode placement. To date, the best methods for seizure detection have been feature-based methods, mainly due to the lack of large open access datasets, which will facilitate the use of neural network-based approaches.

Training of the models (splitting, validation)

The training routine of the model is done in a **leave-one-out cross validation**. The data of one patient is left out as a test set, and the remaining data is used to train the model. In the epilepsy use case, when the dataset contains continuous recordings, the imbalance between classes (seizure and background EEG) tends to be very high. Seizure events are rare and are not predominant in the data. High imbalance can lead to lower performance, since the impact of the minority class will be minimal in the model's training. To solve this, it is possible to sample the training data segments, either by up-sampling the minority class (seizures) or by down-sampling the majority class (background EEG). In current applications, the seizure data segments are up-sampled by overlapping the data segments, meaning that two consecutive segments will contain 75% of common data. The majority class is down-sampled by taking 1 minute long non-overlapping segments every 15 minutes of recording.

It was then noticed that this algorithm suffered from **stability issues**, as the standard deviations of the sensitivity and the false alarm rate were high, mainly arising from the random selection of background

samples for balancing the classes. Therefore, different undersampling approaches were exploited, finally opting for the use of an adapted version of a cluster-based undersampling. The number of background samples was selected after performing a clustering with k-means (for a detailed analysis of k-means, see Module 4). So, all the background samples are initially clustered, and then a specific amount of background samples is picked from each cluster. This way, you do not select random background samples, instead selecting "representative" samples by picking them from all of the k different classes of background samples.

Hyperparameter tuning

The SVM is initially tuned via grid search. In this method, the hyperparameters are tested by defining specific values for each hyperparameter, and training and testing the model for each combination of parameters. The hyperparameters that produce the best classification are chosen to be the optimal ones. Contrary to the training routine, the hyperparameter tuning is done in a 5-fold cross-validation, where the data is divided into 5 portions: 4 are used to train the model and 1 is used to evaluate. This avoids overfitting the hyperparameters to specific patients in the dataset.

Multimodality

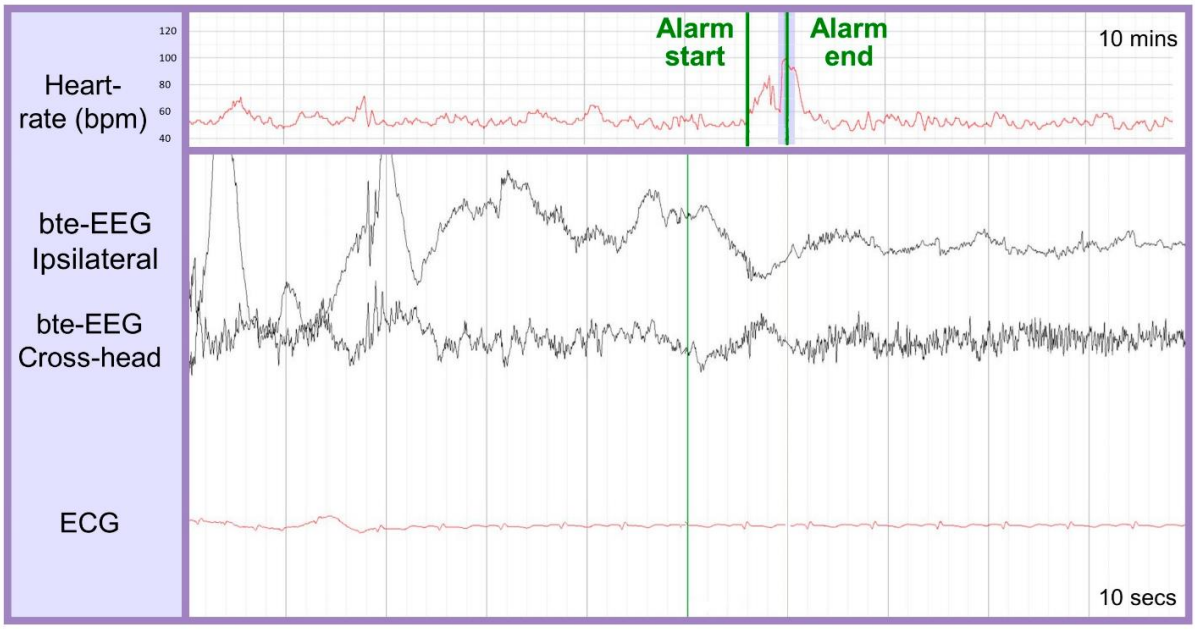
In the datasets of this use case, the wearable device measures not only **bte-EEG**, but also **ECG** and **EMG**. It is known that in some types of epilepsy, several physiological manifestations can occur other than in the brain. For example, in patients with motor seizures, the muscles are usually stiffened and/or they convulse. This can be measured with EMG. In other cases, the autonomic nervous system, in particular the cardiovascular system, can suffer pathological changes. In some individuals, the heart rate increases due to seizures, which can be measured with ECG. Having multiple sources of physiological changes induced by seizures, can help seizure detection.

The integration of **multiple modalities** can follow different strategies, such as:

- Early integration: where the data from each modality is merged before feeding the model.
- Late integration: where an individual model is developed for each modality and each output is merged in the final classification steps.

Furthermore, we have exploited the help of multimodality in the context of **absence seizures** for the detection of **false alarms**. Absence seizures usually occur when the patient is "frozen" (as you have seen in the introductory example with Noah), hence, lack of motor activity is expected. An alarm which occurs simultaneously with high activity of the patient is likely to be a false alarm in the case of absence seizures. Hence, data from accelerometers and gyroscope can be used to reduce false alarms (Chatzichristos et al., 2022).

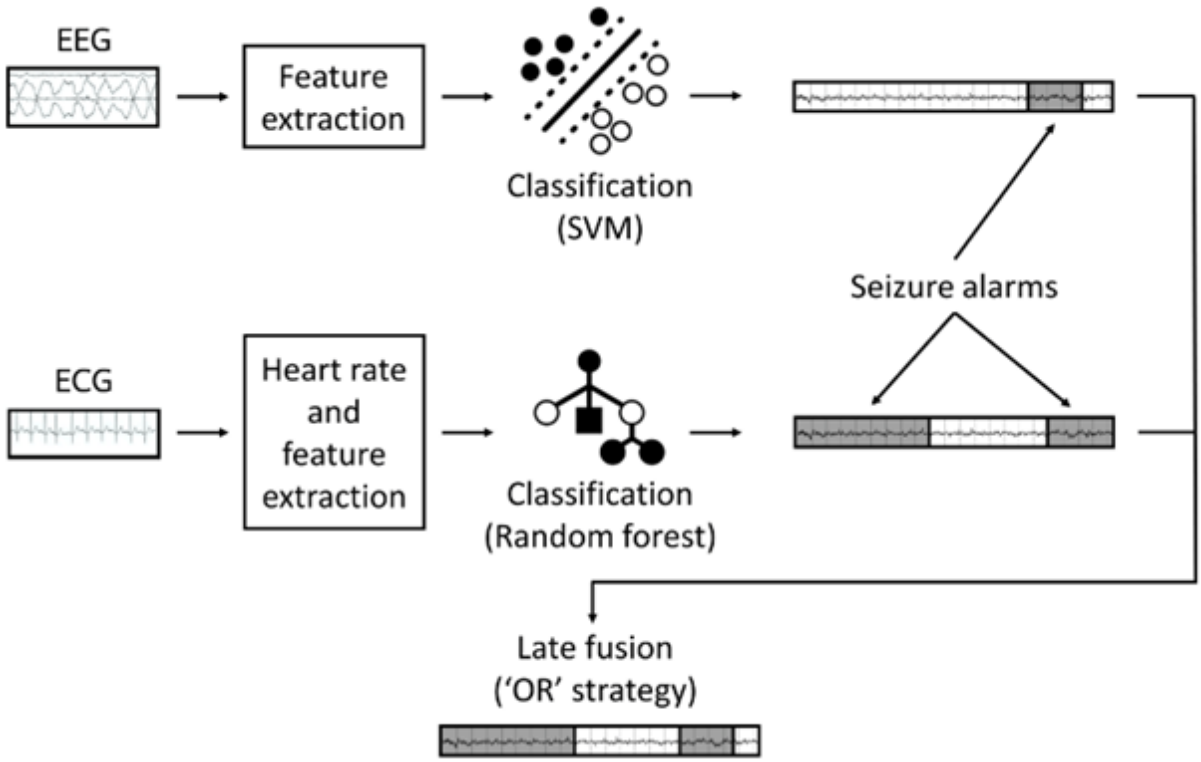
A known issue in multimodal seizure detection is the **misalignment of the seizure effects** in the different signals. For example, the heart rate increase tends to manifest up to a minute after the electrophysiological effects in the brain stop. When developing multimodal algorithms for seizure detection, it is important to take this into account. It is also important to adapt the detection algorithms for each modality (when using separate models). Typical EEG models take as input data segments of a couple of seconds, since seizures produce rapid periodical changes (around 3 Hz) in the signal. The heart rate increase usually is noticeable when looking at a larger segment size. The ECG model should take larger segments as input in order to identify the changes in the signal.



In the figure above, you can see an example of a seizure that was missed from neurologists when they checked only EEG. The change in the heart-rate aids in the detection of the seizure from the multimodal AI algorithm. You can note that there is a delay between the seizure in EEG and ECG, for which the algorithm must account for. In EEG-based AI applications, fusion methods are employed to integrate information from multiple EEG sources or features. There are three primary fusion approaches: early fusion, mid fusion, and late fusion.

- **Early fusion** combines raw signals or features from different sources before any further processing, enabling the AI model to learn directly from the integrated data.
- **Mid fusion** involves processing data from multiple modalities separately, and then combining the intermediate representations or feature vectors at a mid-level before feeding them into the AI model.
- **Late fusion** processes data independently from different modalities, often employing distinct AI models, and combines their outputs or predictions after the individual processing stages (Chatzichristos et al., 2022).

Late fusion is particularly advantageous when the different modalities provide unique and complementary information about the underlying event. By combining the outputs or predictions of multiple AI models trained on these sources, late fusion enhances the overall system's performance, accuracy, and robustness. In this use case, a late fusion with an “Or” is employed. This means that an event is considered a seizure if it is considered a seizure either with the use of EEG OR with the use of ECG. Hence, the two modalities are analysed independently from each other (late fusion). Late fusion approaches are more appropriate when there are misalignments between the signals (as in our case as explained above) or other mismatches between the different signals that will be analysed (e.g., different resolutions, time shifts, etc.). In the following figure you can observe the late fusion model that has been employed in Bhagubai et al. (2023) and that improved the accuracy of the detected seizures significantly:

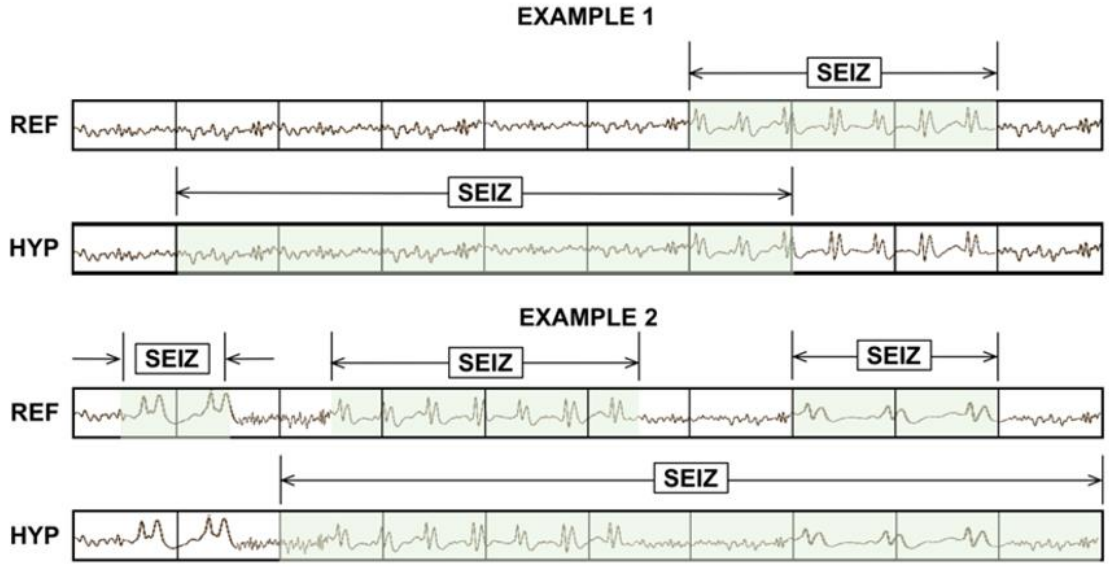


6.5 Evaluating AI

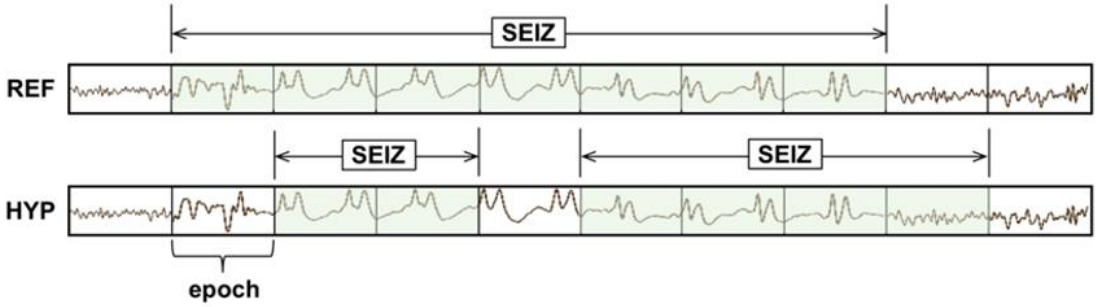
Ground truth and evaluation of AI performance

Supervised models require **labels** for identifying the different classes. In this use case, we resort to the seizure annotations produced by experts on the vEEG data. The evaluation of the model is done by comparing the classification output of the model of each data segment to the corresponding label (seizure or non-seizure) in the **ground-truth**. In time-series classification, metrics are calculated by directly comparing every data segment (or **epoch**) of the hypothesis directly with the correspondent ground-truth epoch. However, other methodologies can be used, considering the use case and the end-application of the seizure detection framework. The main objective of such models is to aid clinicians and decrease the review time of large amounts of data. The exact start and end-time of the seizure is not as important, since clinicians can review the alarms that were produced by the algorithm. An epoch-based evaluation can penalize the model's usability.

Another method for evaluating the performance in this use case, is the **any-overlap method**, where the hypothesis and ground-truth are compared at an event level. We consider a correct seizure alarm if the start and end time of the alarm produced by the model overlaps with the start and end of the ground-truth event. If the alarm does not overlap, it is counted as a false positive. Examples from the cases above can be seen in the following figure (Ziyabari et al., 2021):



OVLP scoring is very permissive about the degree of overlap between the reference and hypothesis. The TP score for example 1 is *1* with no false alarms. In example 2, the system detects *2* out of *3* seizure events, so the TP and FN scores are *2* and *1* respectively.



EPOCH scoring directly measures the similarity of the time-aligned annotations. TP, FN and FP are *5*, *2* and *1* respectively. TPs are considered for epoch 3, 4 and 6-9. Epochs 2 and 5 are missed (counted as FNs). Epoch 9 adds an FP.

Improvement of labels by AI

In this use case, the objective is to detect seizures based on wearable data. The ground-truth annotations are made on the vEEG system. Wearable EEG setups are developed with the intention of causing minimal obtrusion to the patients. In this sense, the number of electrodes used is significantly reduced, and their placement is usually on non-standard locations of the scalp. This can be a cause for **seizure patterns not being visible in the wearable channels.**

Including data segments that are labelled as seizure but do not contain seizure patterns can dampen the performance of the models. An alternative to this is to present wearable EEG data to clinicians and obtain annotations done in this modality. However, this is time consuming, and clinicians do not specialize in annotating in non-standard EEG montages. Another approach is to automatically correct seizure labels. One method is self-confidence-based annotation correction, where the objective is to estimate the joint distribution between the estimated labels and the true labels and prune the true labels (Zhang et al., 2022).

Relevant metrics

Since the AI system is a monitoring tool for clinicians, the most important evaluation metric is the model's **sensitivity**. It is more valuable to detect the highest number of seizures and disregard the false positives, since clinicians will manually go over the alarms. The sensitivity is simply the number of true

positives (alarms that are aligned with the ground-truth) divided by the number of true positives and true negatives (all correct classification of both classes). See Module 4 for more information.

In highly unbalanced use cases, such as seizure detection, **False Alarm (FA) rate** is commonly used since the specificity usually yields misleading high values due to the large number of true negatives. The false alarm rate is simply the number of false alarms divided by the total duration of data and normalized to one hour (FA/hour) or one day (FA/24 hours).

What is the gain for doctors?

The use of wearable EEG devices along with AI-based automatic seizure detection can offer several benefits to neurologists:

- **Improved efficiency**: Neurologists can efficiently monitor and assess a patient's EEG data remotely, reducing the need for frequent in-person visits. This saves time and resources for both the neurologist and the patient.
- **Timely intervention**: Automatic seizure detection can provide real-time alerts to neurologists when a seizure occurs, enabling them to intervene promptly and adjust treatment plans as needed.
- **Enhanced data**: Wearable EEG devices can provide continuous data over extended periods, offering a more comprehensive view of a patient's brain activity. This data can help neurologists make more informed decisions regarding treatment adjustments. The manual analysis of this large amount of data per patient is almost impossible to be performed from the neurologists.
- **Long-term monitoring**: The combination of wearables and AI allows for long-term monitoring, which is particularly valuable in assessing the effectiveness of treatment plans over time and adjusting them accordingly.
- **Data trends and insights**: AI algorithms can analyse large volumes of EEG data, identifying trends and patterns that may not be immediately apparent to human observers. Furthermore, the decrease in time needed for a neurologists to even validate and check all the alarms provided by an automated AI pipeline is tremendous compared to the full check of the vEEG files. As it has been reported in Swinnen et al. (2021) the average time to review a 24-hour, algorithm-labelled SD EEG file was 5–10 min, in comparison to 1–2 h for full EEG review without automated annotations. This can aid neurologists in making more accurate diagnoses and treatment recommendations.

In summary, the gain for neurologists lies in increased efficiency, timely intervention, access to high-quality data, long-term monitoring capabilities, and AI-driven insights that can lead to better patient care and treatment outcomes.

6.6 Challenges

Privacy of AI

The integration of wearable EEG devices with AI-based automatic seizure detection raises some important **privacy concerns**. These concerns revolve around the sensitivity of the health data collected, data security vulnerabilities, ownership and consent issues, data sharing with third parties, data retention policies, challenges in de-identifying or anonymizing data, varying legal and ethical frameworks, and the risk of data breaches.

It is imperative for stakeholders in healthcare, including providers, device manufacturers, and AI developers, to establish stringent privacy and security protocols, obtain informed patient consent, and comply with relevant data protection regulations to address these privacy concerns effectively while

maintaining trust with patients. Ensuring the security of the data collected by wearable EEG devices is crucial. Any vulnerabilities in data storage or transmission could be exploited by malicious actors.

Barriers to the patients with use of AI

There can be different barriers for patients when using wearable EEG devices for seizure monitoring:

- **Stigmatization**: Patients may fear being stigmatized due to visible or noticeable wearable devices, potentially leading to social discomfort or discrimination. Hence, the main requirement of the patients for using a wearable EEG is to be as invisible as possible. Following the views of the patients, the devices need to be "as 'normal' as possible," "non-stigmatizing" and "non-intrusive" (Bruno et al., 2018)
- **Quality of treatment**: Overreliance on wearable devices and AI for diagnosis and treatment decisions, without regular in-person visits to healthcare professionals, could potentially lead to a lower quality of care, missing important clinical nuances.
- **Limited interaction with doctors**: Reduced in-person visits may result in less interaction with healthcare providers, potentially impacting the doctor-patient relationship and the ability to address holistic healthcare needs.
- **Technological barriers**: Patients with limited access to or familiarity with technology may face challenges in using wearable EEG devices effectively, creating a digital divide in healthcare access. Furthermore, the continuous monitoring under specific conditions (e.g., increased temperature) is not yet viable due to side-effects of the wearables (e.g., itching) or increased noise (e.g., sweating).
- **Privacy concerns**: Patients may worry about the privacy of their health data, leading to hesitance in using wearable devices, particularly if they believe their data could be misused or compromised.
- **Reliability and false alarms**: Wearable devices may not always provide accurate seizure detection, due to increased noise or artifacts, leading to false alarms or missed events, which can be frustrating and potentially affect patient compliance.

To mitigate these barriers, it is essential for healthcare providers to offer comprehensive patient education, ensure that wearable devices are integrated into a broader care plan, address privacy concerns, and maintain regular communication between patients and healthcare professionals to provide a balanced approach to seizure management that leverages both technology and traditional medical care.

Quality of data in the wild

The clinical adoption of wearable EEG remains challenging, since the first-ever phase-4 clinical study was conducted, proving the feasibility and comfort of the device in a home environment. Furthermore, there are challenges regarding the signal quality and the time needed by the neurologists to annotate the data. Hence, research for automated seizure detection algorithms, tailored for wearable EEG, is still in its infancy.

If someone tries to use the same algorithm for capturing seizures with the same wearables within a hospital environment and "in the wild" (in everyday life), he/she will face significant challenges. For example, the novel automated absence detection algorithm, reported in Swinnen et al. (2021), achieved a mean sensitivity of 0.983 in the detection of absence seizures, with mean False Alarm (FA) per hour equal to 0.9138. The use of the automated algorithm reduced the review time of a 24-hour recording from 1 to 2 h to around 5–10 min for the neurologists. Nevertheless, the reported results were obtained from data acquired within the hospital environment. We argued that the first phase of the recording of each patient will consist of routine monitoring in the Epilepsy Monitoring Unit (EMU)

after which this data can be used to train the algorithm. The obtained classifier can be used to detect seizures from the data acquired when the patient wears the Sensor Dot (SD) at home.

As observed from the data obtained from the patients in the home environment with SD (Chatzichristos et al., 2022), the performance of the automated seizure annotation was much lower, compared to the performance of the algorithm trained and tested with data obtained in the hospital, especially in terms of FAs. A significant number of artifacts decreases the performance of the algorithm. Some of the artifacts (e.g., running artifact) are not be present in the data in the hospital environment, hence, the algorithm cannot recognize them.

6.7 Future perspectives

Finally, what are the prospects of applying AI for epileptic seizure detection in the future? Let's hear what our experts expect or wish for in the future:

Ask the technical expert - Future perspectives

The best-case scenario, let's say, would be to have a perfect epileptic detector that would work with wearables so that all the patients could take a wearable that also complies with their specifications, let's say, because all the patients do not like to have something visible so we need to create something that is very miniature, not adding any bias or stigma to the patients. So it's really important also to take into consideration the likes of the patients. Having every patient wearing a wearable and monitoring them 24 hours, it might sound like a Big Brother scenario, but still it will provide the best solution for the AI. If we have a lot of data, and a lot of good data, AI can do really good work. So in the best-case scenario, the patients are monitored 24 hours and we provide alarms, real-time alarms to the doctors. And even, I mean it's still undergoing research, it might be possible to predict some of the seizures even before happening so a couple of seconds before happening, and then let the patient know, let the family of the patient know that he or she can sit and or not make some activity.

AI cannot replace the doctor. That's my strong belief. I believe that AI can help a lot the doctor, be like something like the digital assistant of a doctor or digital colleague of the doctor, especially when it comes in cases where we have a lot of data that is impossible and intractable to be analysed by a single doctor or even a couple of doctors, then AI can help. So AI would always need good input from the doctor. So AI cannot do miracles. Always we say “if rubbish comes in, rubbish goes out.” So we always would need very good input from the doctors, very good labels, very good interpretation of what is happening. AI does not know physiology and does not know the patient himself or herself. So we always need the help of a doctor. But as I said before, the doctors will need AI more and more as data will become bigger in size. It's inevitable that they will need some help. And AI is here to stay, let's say, and help them in this field when they cannot handle the vast amount of data.

Ask the clinical expert - Future perspectives

I don't think that AI will be able to replace clinical experts. You really need somebody who can look at it at all aspects of a given problem, but AI certainly can assist the clinician in looking at the huge data files and also detecting subtle anomalies which the clinician might have overlooked. So I see AI application tools as an assistant to the clinician who is the expert. And it will be able to really refine clinical expertise, speed up a diagnosis and management, but it's not going to replace a clinician.

7 AI in action: Rheumatic heart disease detection

7.1 Welcome to Module 7

Welcome to Module 7. This is one of the "**AI in action**" modules, where we shift our focus from theoretical concepts to practical use cases, allowing clinical and technical experts to share their insights and experiences with AI applications in the healthcare field.

Why the "AI in action" modules matter

These modules about real-life use cases are significant as they bridge the gap between theory and real-world impact. By delving into the insights of experts who have developed or utilized AI applications in healthcare, you will gain a nuanced understanding of the transformative potential of AI in the field.

Module 7 in particular, offers the use case of applying AI in the detection of rheumatic heart disease, especially in the Global South. Discover how AI can be used for latent rheumatic heart disease detection, through interview clips with an expert in this field.

Learning goals

- Examine case studies and real-world applications where AI has been successfully integrated into clinical practice, and in this use case: Rheumatic Heart Disease (RHD).
- Understand the importance of data preprocessing and denoising for ECG analysis: Learners will be taught techniques for artifact removal and noise reduction in ECG signals, using methods such as linear or non-linear filters and wavelet transforms, to prepare data for accurate analysis.
- Comprehend the importance of early detection and screening in Global South: Participants will learn about the significance of early detection of RHD through optional screening mechanisms and the role of public awareness in preventing the disease's progression.
- Understand how to extract and select significant features from ECG data for AI analysis, focusing on both time-domain and frequency-domain characteristics.
- Describe broader implications and limitations in this use case.

7.2 Context and clinical impact

Noah woke up with a scratchy throat, his voice barely a whisper as he called out for his mom. Zarah's motherly instincts kicked into doctor mode as she examined him. "Looks like you've got a sore throat, Noah. Let's take a closer look." After examination, Zarah's suspicions were confirmed: Noah had strep throat. She immediately went to the pharmacy, returning with a bottle of antibiotics. "Here is the medication you need. You'll be feeling better in no time."

Noah sighed in relief, grateful for his mother's expertise. But Zarah's expression turned serious as she handed him the medicine. "Not everyone is as fortunate as you, Noah," she said solemnly. "Strep throat infections can lead to serious complications if left untreated, especially in places where access to healthcare is limited."

She explained to him how untreated strep infections could progress to acute rheumatic fever and ultimately rheumatic heart disease, particularly affecting those in low-income countries and disadvantaged communities.

Noah listened intently, understanding the importance of prompt treatment and the disparities in healthcare around the world. As he swallowed the first dose of medicine, he silently vowed to appreciate his mother's care and to never take his health for granted.

Strep throat infections are often caused by Group A Streptococcus (GAS) bacteria, and are commonly manifested as a sore throat in children and adults. Untreated or insufficiently treated strep infection leads to an acute rheumatic fever (ARF), which in the case of cardiac tissues, can progress to **rheumatic heart disease**.

Rheumatic heart disease (RHD) is a condition that results from rheumatic fever, which is caused by untreated streptococcal infections. It involves damage to the heart valves due to inflammation, leading to symptoms like heart murmurs, chest pain, and shortness of breath. Long-term complications can include heart failure and increased risk of stroke.

RHD can be entirely prevented and treated, but it disproportionately affects low-income countries in the Global South and socioeconomically disadvantaged groups. Epidemiological studies among at-risk countries, such as Ethiopia, have shown that chronic rheumatic valvular heart disease is one of the most frequently diagnosed cardiovascular conditions among patients attending cardiology clinics (Asmare, et al., 2021). Valvular heart disease accounts for 40% of the annual cases of cardiac-related diseases, and a striking 80% of these cases are associated with chronic RHD.

This high prevalence of RHD in the Global South is due to many determinants including lack of disease awareness, limited medical resources and physicians, malnutrition, economic burden, etc. Primarily lack of public awareness regarding the connection between GAS bacterial infection and its progression to result in RHD, deters individuals with sore throats from seeking immediate medical care. Therefore, an **optional screening mechanism** is crucial in reducing the burden of RHD in affected populations, emphasizing the **importance of early detection** and timely intervention to combat this debilitating disease. Additionally, increased efforts to raise awareness, improve access to healthcare, and rigorous preventive policies are of utmost importance.

Detecting and diagnosing RHD in the Global South can be challenging due to limited healthcare resources, while early diagnosis is crucial for timely intervention and management. Our expert makes use of AI to help alleviating this problem.

7.3 Data

Research ethics approval

Research ethics and approval of the planned protocol are crucial before starting a study. In case of a multicentric study that involves different parties or countries, like the current use case, the protocol must be approved by all necessary ethics review committees. The followed procedures are similar to the ones for non-AI-based clinical trials for studies that involve the development of AI models.

- **General data protection regulation (GDPR)**: All the sensitive patient data needs to be kept private and secure. GDPR approval should be obtained prior to applying for ethics committee (EC). In this application, the researchers detail tools and methods to be used for data protection and privacy with risk mitigation plans written explicitly.
- **Data management plan (DMP)**: DMP should be stated clearly in the beginning of the study. The research is expected to mention the datatypes, size, format, storage media, related costs if any, etc. in the plan.

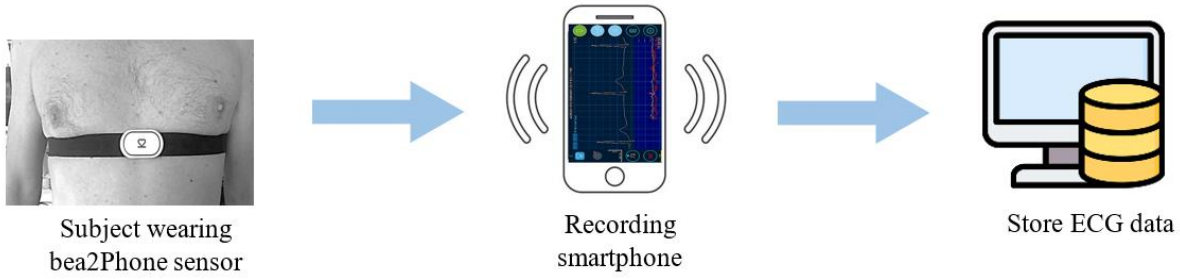
The current use case, "Symptomatic **rheumatic heart disease (RHD)** detection from **electrocardiogram (ECG)** recordings" has no approval from EC UZ/KU Leuven. However, it underwent research ethics regulations at the study site. The consents were obtained from participants during the trial. The dataset has been used in the following publications: Asmare, et al. (2021) and Asmare, et al. (2023).

Data acquisition

The RHD ECG data was collected at Tikur Anbessa Referral Teaching Hospital (TASH) and Cardiac Centre, Ethiopia. The participants were symptomatic, and diagnosis decisions were made by cardiologists using GE Vivid-9 echocardiograph machine. These decisions by the cardiologist were used as a ground truth for the ECG data labels of each participant. The data distribution per age and gender is shown in the table below.

Variables	RHD positive (n=124)	Normals (n=46)
Gender	74 (F) 50 (M)	15 (F) 31 (M)
Age	22.9 ± 8.9y	14.4 ± 9.4y

The ECG data recording procedure can be seen in the figure below. First, the participants were asked to wear a bea2phone single lead ECG sensor. Then, a recording app was installed to access and store the files in the memory. After connecting the phone with the sensor via Bluetooth, the sensor sends the ECG signals to the recording smartphone. Finally, the signals were extracted from the phone and stored in a computer for further analysis.



Data preprocessing and denoising

Artefacts in ECG and wearable ECG sensors are more susceptible to noises than standard wired ECG recording devices. These **artifacts** are commonly caused by movement or improper placement of contact electrode leads, motion artifacts, muscle contractions, and power line interferences.

The presence of artifacts constitutes the **preprocessing** of the ECG signal as an important prerequisite before analysing the underlying diagnostic markers. Studies have shown that removal of artifacts from ECG can be done using different techniques including linear or non-linear filters, wavelet transforms, empirical mode decomposition (EMD), deep learning models, etc. Here are the **denoising** techniques used for the acquired signal in this case:

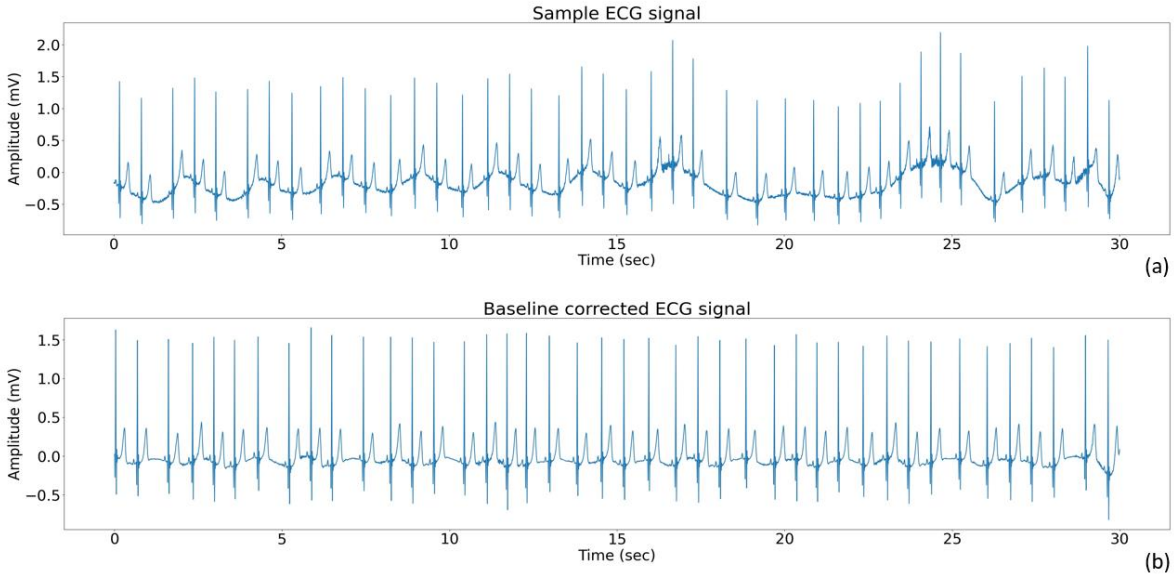
Baseline wander noise

Baseline wander (BW) noise can be easily observable at first sight when looking at the signal. It is a low frequency noise that occurs due to breathing. The BW noise often lies below 1Hz, typically less than 0.5Hz (J. A. Van Alste, et al., 1985, Sörnmo L. and Pablo L., 2005). Removal of this noise is crucial, particularly in an ambulatory or exercise electrocardiography as the chest walls move in higher intensity. The baseline wander noise is around the lowest frequencies of an ECG signal. A high-pass filter can be designed in order to remove it. The decision of cut-off frequency depends on the experimental setup, although it is often recommended to apply filtering from 0.5Hz to 0.8Hz (J. A. Van

Alste, et al., 1985). It is worth noting that the diagnostic frequency contents of the signal are not affected. For our use case, a Butterworth high-pass filter was used (more information in Module 2) with a cut-off frequency of 0.5Hz, and stopband attenuation of -60dB.

Cut-off frequency is the point in this spectrum where the filter starts to take effect. Frequencies below this point will pass through the filter without being altered, and frequencies above it will start to be reduced in strength.
Stopband attenuation refers to how much a filter reduces the strength or amplitude of frequencies within the stopband. The stopband is a range of frequencies that the filter is designed to block or significantly weaken. In other words, stopband attenuation measures the effectiveness of a filter in silencing or damping the frequencies that are not wanted. The higher the stopband attenuation, the more those unwanted frequencies are suppressed, meaning the filter is more effective at keeping those frequencies out of the output signal.

The obtained results are shown in the figure below for the original noisy ECG signal (a) and its corresponding filtered signal (b).

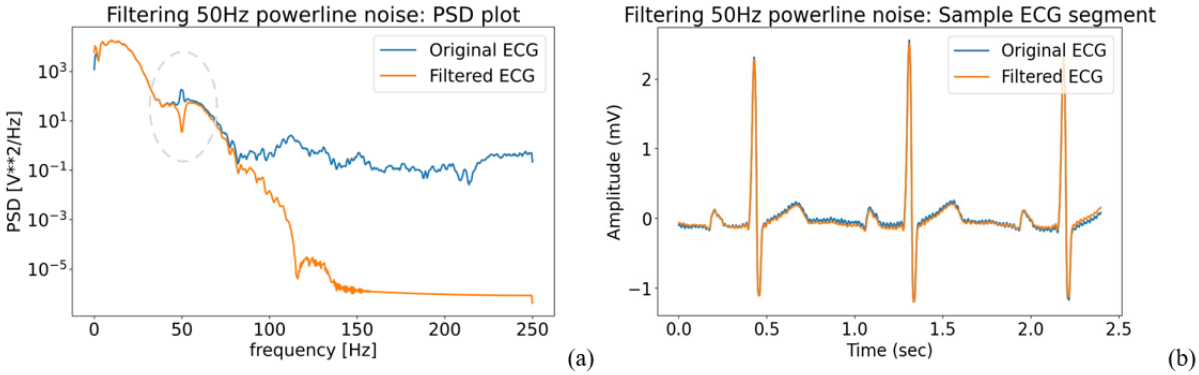


Power line noise

The non-stationary frequency generated by power mains affects the acquired signals. This frequency lies at either 50Hz or 60Hz in the USA. Since it lies at this particular frequency, a narrow band notch filter can be used to remove such noise. As can be seen in the figure below (a), the power spectral density plot of the signal shows a large peaked spectral energy at 50Hz indicating that there is a powerline interference at the corresponding frequency. Applying a IIR notch filter at 50Hz attenuated PLI noise. A better way to look at the effect of the filtering is to plot the signal's power spectral density as shown in the figure in (b).

Power spectral density (PSD) is a measure used in signal processing that describes how the power (or intensity) of a signal is distributed across different frequencies. Think of it as a way to break down a signal, like a piece of music or any time-varying signal, into its constituent frequencies, and to show how much power each of those frequencies carries.

When you look at a PSD plot, you're essentially seeing a map that tells you which frequencies are dominant in your signal and how much energy they have. For instance, in a piece of music, the PSD could show you how much of the song's energy is in the bass frequencies versus the treble frequencies. PSD is important because it helps engineers and scientists analyse signals in various fields, such as telecommunications, seismology, and audio engineering, to understand better and process these signals according to their frequency content. It's a fundamental tool for identifying patterns, filtering unwanted noise, and optimizing signal transmission or storage.



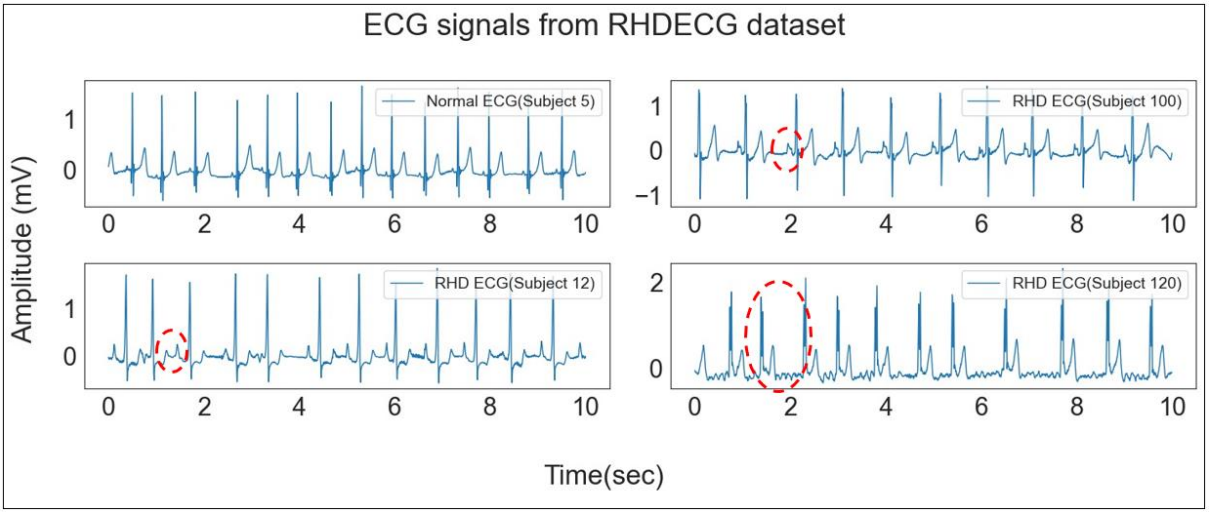
Muscle tremor artifacts

Artifacts due to muscle movements of the subject are one of the most common noises in ECG acquisition. The frequency range of this noise is wide, greater than 20Hz-200Hz (Drake J.D. and Callaghan J., 2005). A low-pass filter with an upper cut-off frequency up to 40 Hz is often used. Let us modify the Butterworth filter to be a bandpass filter to incorporate filtering of both BW and motion artifacts. Thus, we add a high pass cut-off frequency at 100Hz considering the age of participants in this use case and based on visual inspection of the signal. Continue to the next page for an exercise.

7.4 AI in action

Feature extraction

Extracting features from the acquired data requires careful analysis of underlying representative characteristics from the signals. The features also determine the performance of the AI model. In this study, since the participants were already having clinical symptoms of cardiovascular disease, the **ECG waveform of RHD patients manifests various morphological changes** depending on the deteriorated sub-valvular location, as illustrated in the figure below.



These morphological changes often result in fibrillatory waves, flattered waves, bifid and/or wide P-waves, biphasic T-waves, notched QRS complex, peaked tall T-waves, and other morphological deformations. Hence, the features can be extracted both from the time and frequency domain.

- In the **time** domain of the ECG signal, only five commonly used heart rate variability related features were calculated. These are the root mean of the squared successive differences between adjacent RR intervals (RMSSD), the standard deviation of the successive differences between RR intervals (SDSD), median absolute deviation of the RR intervals (MADRR), the 20th percentile of the RR intervals (Prc20RR), and proportion of RR intervals greater than 50ms (pRR50).
- In **frequency** domain, the input ECG signal is decomposed into wavelet features in 6 different levels. Then, relative percentages of wavelet energies at each level were calculated. The resultant feature vector for each ECG is 12 (five from time domain and seven frequency domain).

Wavelet analysis is a mathematical technique used to break down complex signals into simpler, manageable chunks of information across different scales or resolutions. It's somewhat similar to a musical score being analysed in terms of its various notes and rhythms, but instead of music, wavelet analysis deals with signals - anything from seismic data, to heart rates (as is in our case).

The core idea behind wavelet analysis is to represent a signal with wavelets - small waves that are localized in time. These wavelets can vary in scale (size), allowing them to capture both the fine details and the broad trends within a signal. This capability makes wavelet analysis particularly powerful for analysing signals that have non-stationary or transient characteristics - where the signal's properties change over time.

The process of decomposing a signal into different levels (broader trends or the overall shape of the signal) is often performed using a technique called the discrete wavelet transform (DWT). DWT applies filters to the signal to separate it into high and low frequency components. The low-frequency components (levels) are then further decomposed into even lower frequency approximations and higher frequency details, and this process is repeated multiple times to achieve multiple levels of analysis.

One can have various other features related to heart rate variability, such as in frequencies (very low, low, or high frequency), geometric features, or Poincare features. Often these features need to be computed from 24-hours duration ambulatory ECGs. However, the length of clean ECG segments from acquired recordings is short, thereby limiting us to use the above mentioned features for classification.

Feature normalization

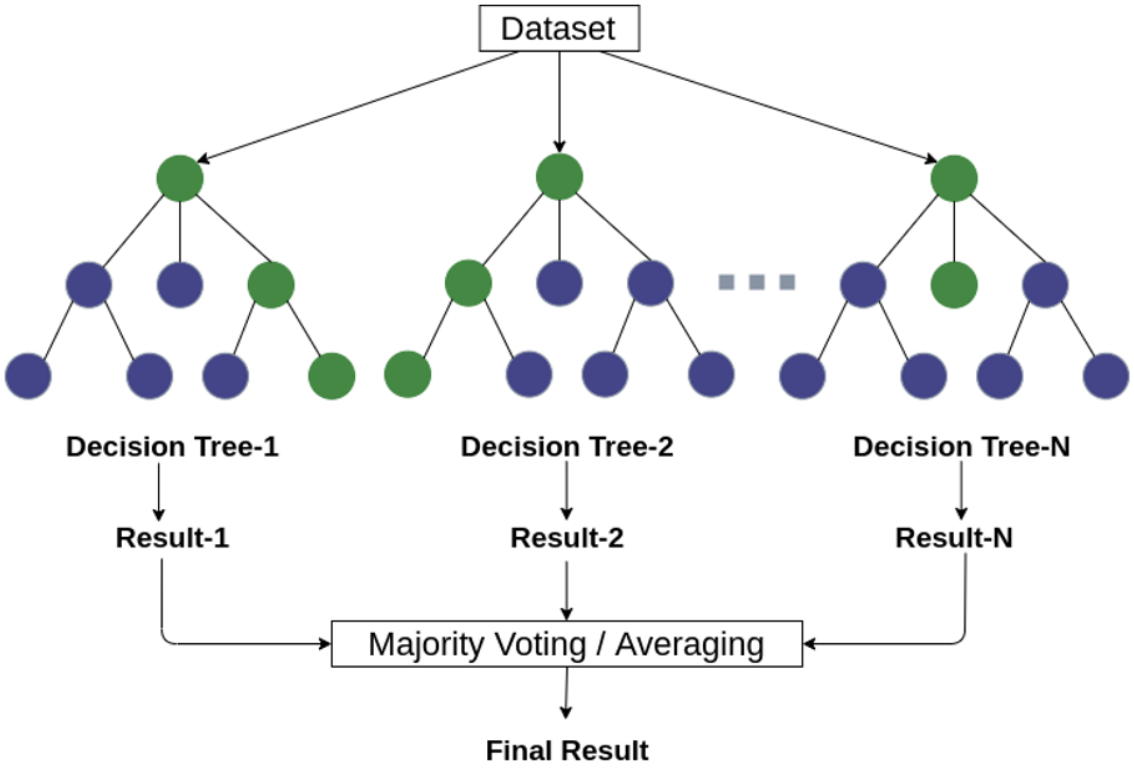
Data normalization is an important step in machine learning. The features to be fed to the model need to be of equal importance for the classifier during training. It can be understood from the relative percentage computation of wavelet energy features that they are in [0,1] scale, however, the time domain features are not in this scale. A first approach can be **converting the units from seconds to milliseconds**. A second approach is to **scale time domain features in the same range with energy features**. For our features, we converted the time units to milliseconds as shown in the table below. It is worth noting that data can also be standardized with zero mean and unit variance, thus the scales are similar for all features.

ECG_Id	RR_RMSSD	RR_SDSD	RR_MAD	RR_Prc20	RR_p50ms
1	0.034238	0.034648	0.068200	0.5280	0.012500
2	0.033894	0.035030	0.023722	0.5600	0.006250

3	0.191575	0.200890	0.106747	0.5872	0.058333
4	0.164881	0.172666	0.100817	0.7240	0.07272
...	...	...	...	...	...

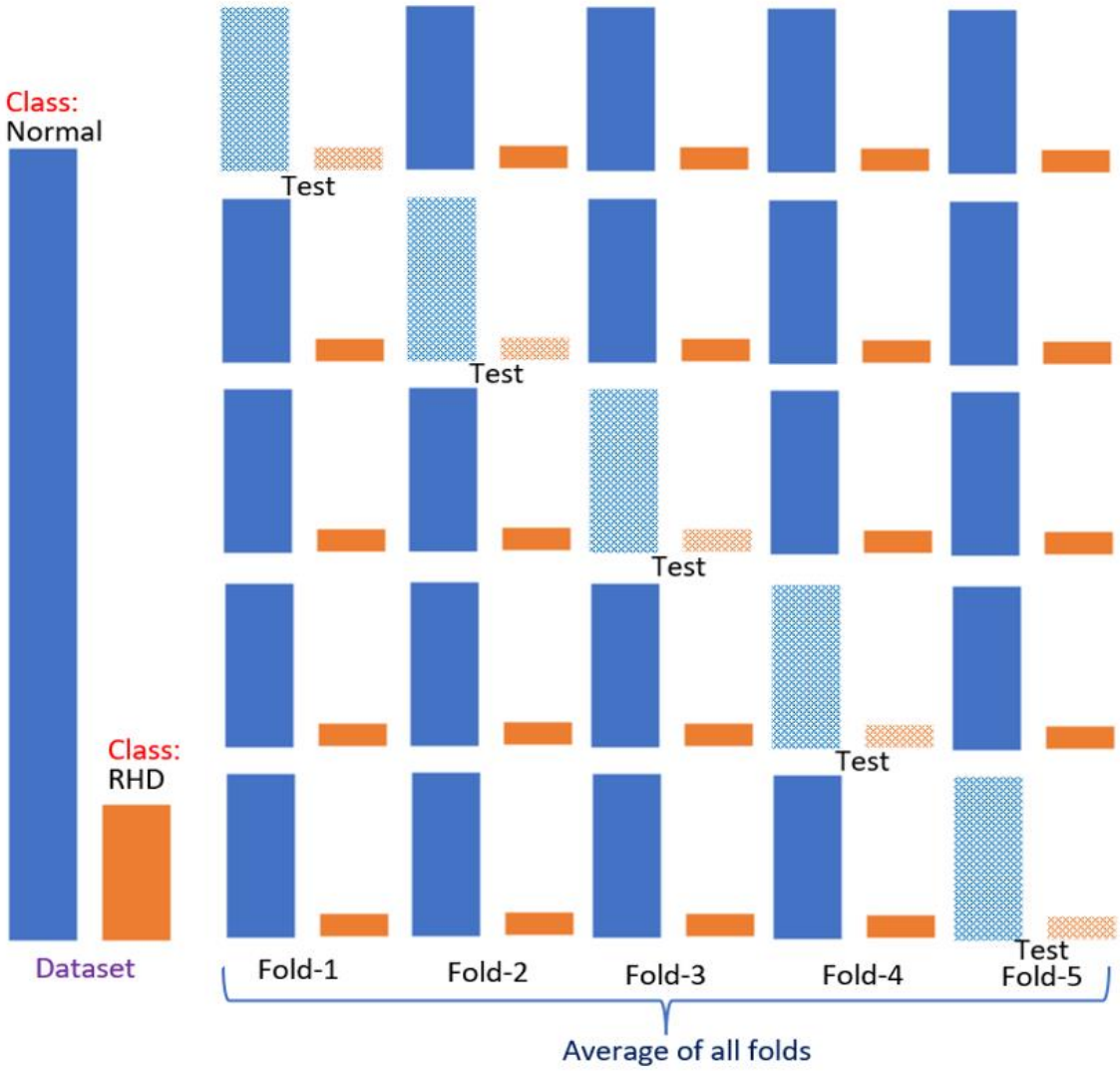
Selection of the appropriate ML model

There are various determinants for the choice of learning model. Considering the nature of input data samples and the expected output, we opted for the use of **random forests**. The model is expected to learn distribution patterns for a binary classification from the extracted features, not from the raw signal (see figure below). In our model, the features from each 10 second ECG sample were input to the tree-based decision makers. Decisions at each node were aggregated by either voting or taking the average to make the final class probability.



Training of the models (splitting, validation)

Once an appropriate model is chosen, then the data needs to be split into **training** and **test** sets (often training, validation and test sets). The test set should always be unseen data that we expect to be a representative of the real-world scenario. Data splitting could be done either by splitting the data based on a fixed ratio, or by using a **k-fold cross validation or leave-one-subject-out (LOSO)** approach. The choice merely depends on the classification problem at hand. In practice, the fixed ratios for training and test sets are 70% and 30% percents of the entire dataset respectively. However, in case of k-fold cross validation, which is the case in this example, the data need to be spliced sequentially into train-test sets k-times, as shown in the following figure:



Since an ECG varies per subject, localized affected area of the heart, and disease progression level; the LOSO approach would be preferable. We will use 10-fold cross validation, thereby ten rounds of training and validating. In each fold, one chunk is used as a test set, and the remaining is used for the training set. For instance, in the first fold the data is split into ten slices, of which the first slice is used for test set and the remaining nine chunks are used to train the model. From each ECG sample, a 10-second length segment was randomly taken for evaluation of the model.

Hyperparameter tuning

Hyperparameters sometimes hinder model performance by large margins. Plotting model learning curves during training is important to visually see its performance. However, we often provide a set of bounded parameter values from which the model needs to select. During training, the model is evaluated for different possible combinations of parameters in the search space that can yield best performance results. Let us define a search space for hyperparameters of our model, as shown in the table below. We train the model in 10-folds and consider the average as a final result.

SNo	Hyperparameter	Search values bound
1	Number of trees	{50, 500}
2	Maximum depth of trees	{4, 10}
3	Number of features	{5, 12}

7.5 Evaluating AI

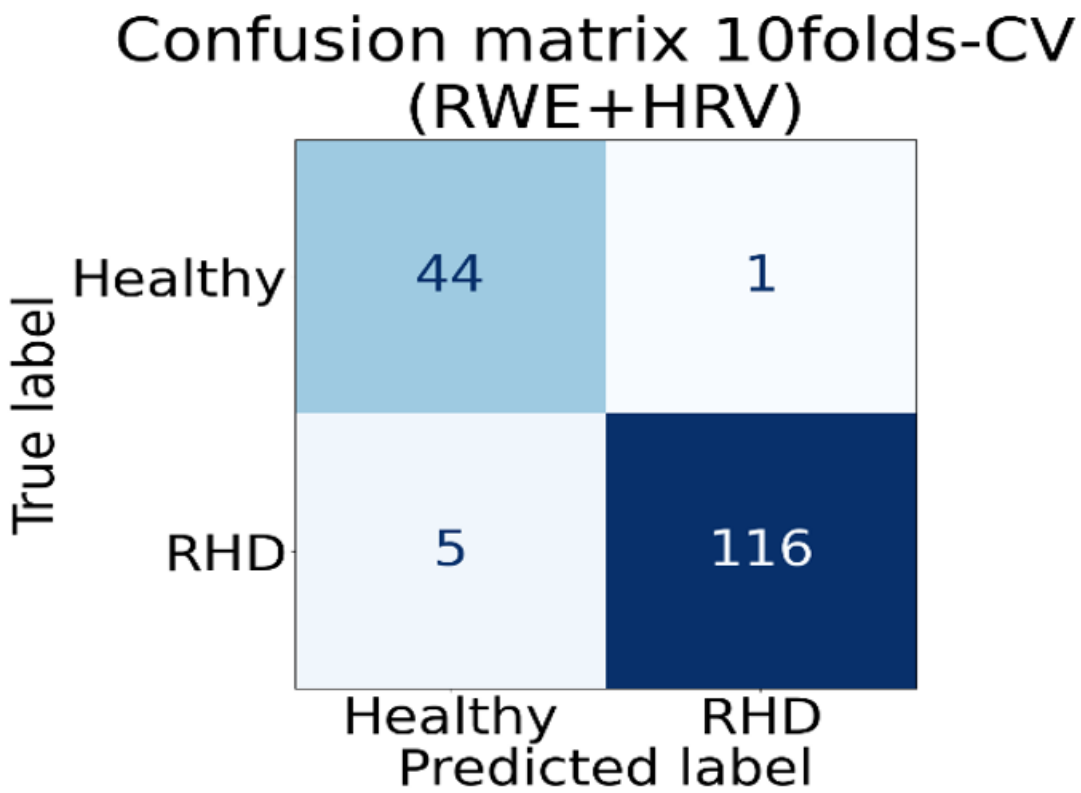
Dataset labelling and performance evaluation

The collected dataset is labelled as either of these two classes:

- Healthy subjects who had normal sinus rhythm (**normal**)
- RHD positive subjects (**RHD**)

To evaluate the performance of the model, appropriate evaluation metrics should be defined. Since the use case at hand aims to detect the disease from patients who are symptomatic and have been referred for an echo check-up at the cardiac clinic, our model should focus on correctly classifying subjects as "normal" or "RHD". This means that **accuracy** is an appropriate metric for this setting. It is worth noting here that if the classes were highly imbalanced, then accuracy would not have been a good measure, thereby the balance between precision and recall (**F1-score**) needs to be monitored. Nonetheless, despite the number of healthy subjects having normal sinus is less than those subjects diagnosed with RHD, the dataset is not significantly skewed to the minority class. The ratio of records "Healthy:RHD" is 1:2 and per subjects is 45:121. It must be noted that the detection scenario is in a referral hospital setting, hence, the balance of the classes is not the same as in the general population. Additional metrics such as sensitivity, specificity and F1-score can also be computed for model evaluation.

Now the model can be trained, and its average 10-fold group stratified cross validation results can be obtained using the following **confusion matrix**:



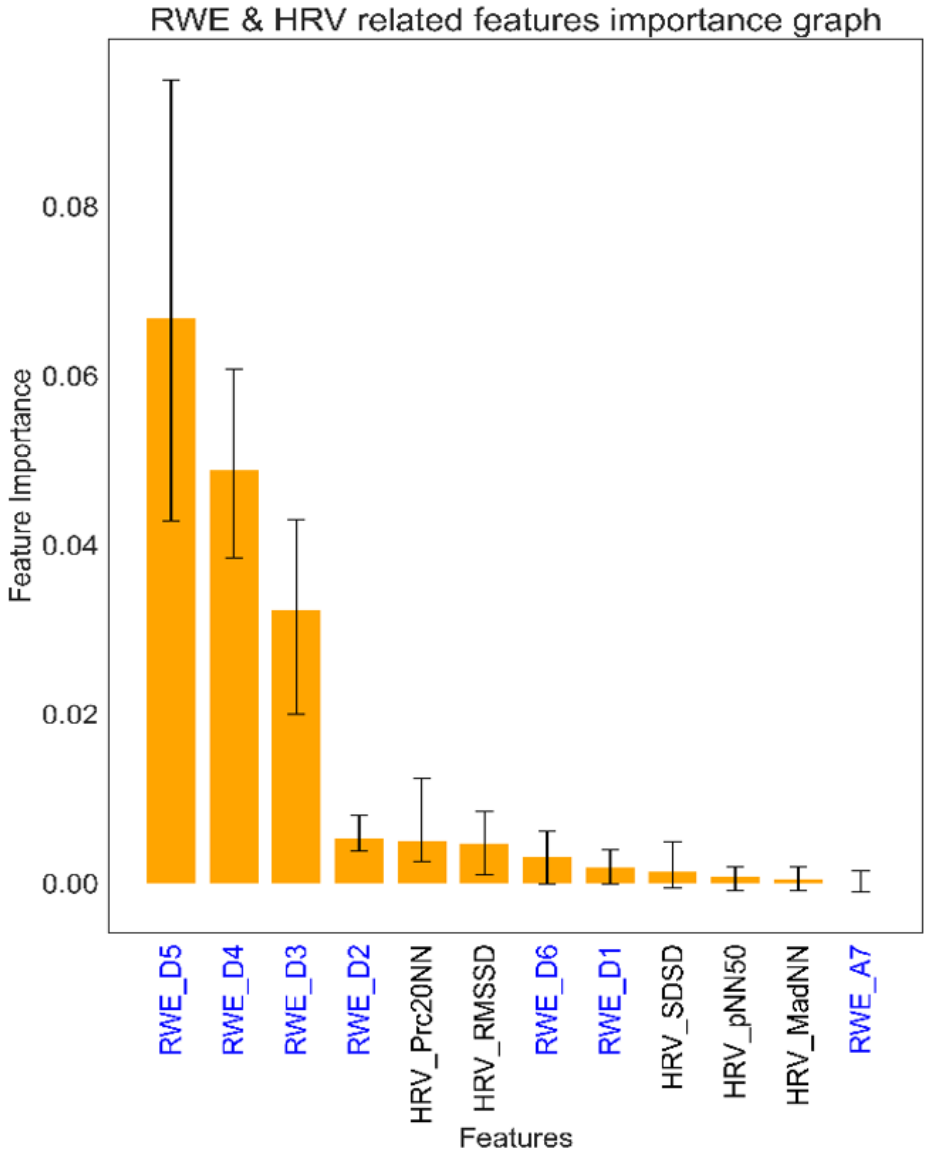
From the confusion matrix results, calculating performance metrics of the model yields as follows. The standard deviation (±std) was computed from the results obtained in 10 folds.

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP} = \frac{44 + 116}{44 + 5 + 116 + 1} = 96.04 \pm 0.04\%$$

$$F1 - score = \frac{TP}{TP + (FN + FP)/2} = \frac{44}{44 + 1} = 97.5 \pm 0.02\%$$

In summary, the results show that the model can detect 96.4% of cases from the total participants ECG data missing 5 subjects as false negatives and 1 subject as false positive. Such light-weight models that have a limited number of parameters to learn, may help the decision process in resource-limited cardiac wards while diagnosing RHD.

To further analyse comparative contribution of features to classify "RHD" ECGs from "normal" sinus rhythms, we can plot a **feature importance graph** as shown in the figure below. Frequency-based features at level 5 to level 2 (RWE_D5, RWE_D4, RWE_3 and RWE_2) were more important for the model to make classification decision than the others.



7.6 Challenges

Privacy of AI

AI systems, such as this research into detecting RHD, necessitate the collection of large amounts of data, often comprising of **sensitive information** like name, age, sex, address, and diagnostic information. While the strict data management plan ensures that this personal information remains inaccessible to the public and is appropriately anonymized upon preprocessing, the concerns about data privacy remain significant.

Consider potential scenarios where anonymized data gets inadvertently correlated with other datasets, revealing patient identities, and leading to consequences like insurance issues, school discrimination, or societal biases. Additionally, the imminent threat of security breaches, where malicious individuals could access data, underscores the importance of not only adhering to but continuously updating and improving data privacy standards in line with evolving technological landscapes.

Barriers to the patients with use of AI

The use of a chest strap for ECG data collection may bring some challenges. Some subjects may find wearing the **strap uncomfortable** leading to hesitancy in participation. The use of wearables might also add to **stigma** to patients, especially in those of younger age.

Furthermore, the perception of AI-driven tools in RHD screening could introduce **concerns about accuracy**, especially in community setups. It is important to ensure that individuals understand the tool's purpose and accuracy to increase trust and maximize participation. There could also be instances where the readings might generate false positives causing undue panic or unnecessary medical investigations. The same is true for false negatives where actual ill subjects might not go to medical centres for treatment.

Data Quality

The effectiveness of the AI tool hinges significantly on the quality of the collected ECG data. **Real-world** or "in the wild" data collection, especially in diverse community settings, can introduce inconsistencies in the data quality. Factors like **incorrect strap placement, movement artifacts**, or even other instrument interferences can impact the reliability of the readings, thus potentially influencing the tool's detection accuracy. Real-world settings often present scenarios that might not have been accounted for during the data collection phase. For instance, a subject might have another type of cardiovascular problem, leading to ECG variations not typically associated with RHD. Similarly, a non-medically trained person might not properly position the chest strap, leading to erroneous data which could significantly affect the model's predictions.

Importantly, for the AI model to be robust and accurate, it requires a large amount of **diverse data**. While collecting data from RHD subjects and healthy controls is a start, the model's effectiveness might benefit from augmentation strategies or additional data sources to enhance its learning and improve its generalization to varied populations. In neglected diseases like RHD, which are specific in geographic location (Global South), gathering extensive data is always challenging. Consider an analogous situation in the field of rare diseases. The rarity often leads to a limited dataset, making it hard to train a comprehensive AI model. While data augmentation, like generating synthetic ECG readings, can assist, it doesn't replace the authenticity of real patient data, leading to potential inaccuracies.

Heterogeneity of data

Given the **demographic diversity**, such as in age, sex, BMI, and other variables, the AI model should be trained to handle and interpret the heterogeneity within the data. Differences in heart rhythms across

demographics or even within the same demographic under **different conditions** can influence RHD detection. Addressing this variability is crucial for the tool's success in real-world applications. Variability poses another challenge. Two subjects from different sex, with different lifestyles, might present varied ECG patterns for the same severity of RHD. If the AI tool is predominantly trained on data from a specific demographic, its accuracy might reduce when used in a different demographic setting, leading to potential misdiagnoses.

7.7 Future perspectives

Challenges of the future

As the AI tool for RHD detection progresses, it will inevitably face future challenges. These may include keeping the model updated with the **latest research findings** on RHD, ensuring its scalability for **broader community use**, and addressing potential **biases** that could emerge from diverse population screenings. Furthermore, continuous validation in **different setups and populations** will be vital to maintain its efficacy.

As the AI tool gains popularity, imagine it getting **integrated with various wearable devices**, from smartwatches to fitness bands. The diversity in device hardware and software could introduce inconsistencies in data collection. In another scenario, with the evolution of medical understanding of RHD, the AI model would need continuous updates, requiring consistent retraining and validation against new data.

Can a model replace the doctors?

While AI models, like the proposed RHD screening tool, show a promise in aiding medical screening, they cannot wholly replace the expertise and **holistic care** provided by physicians. Such models can act as essential screening tools, especially in community mass screening programs, to identify potential RHD cases. However, the final diagnosis, treatment planning, and patient care should always involve medical professionals who can consider a wider array of factors beyond the AI's scope.

To elaborate this further, using AI for preliminary screening can be of analogy to automated blood pressure monitors. While they offer a quick way to check blood pressure, a consistently high reading always warrants a visit to the doctor. Similarly, while the AI tool can detect potential RHD cases, it lacks the holistic understanding of human physiology, patient history, and other subtle signs that doctors consider. For example, a doctor might correlate a patient's symptoms with family history, past bacterial infections, and other diseases to derive a diagnosis, something that an AI model, as advanced as it might be, currently cannot achieve comprehensively.

8 Inside the AI engine: Artificial neural networks

8.1 Welcome to Module 8

Welcome to Module 8 of this course. In this module, we delve into the world of **artificial neural networks (ANNs)**, and explore advanced concepts pivotal to their application in healthcare. Let's embark on this journey to unlock the potential of ANNs in transforming patient care and medical research.

Key Focus Areas

- **Artificial neuron and its networks**: We will explore artificial neurons, the basic computational units of ANNs. Then we will dive into the architecture and functions of neural networks, different classes of ANNs, and the mechanisms of information propagation through these networks.
- **Transfer learning**: Explore how pre-trained neural network models can be adapted and fine-tuned for specific healthcare tasks, enabling efficient utilization of existing knowledge and data.
- **Model-centric vs. data-centric AI**: We will discuss the pros and cons of model-centric and data-centric approaches to AI in healthcare.

Why This Module Matters

ANNs are a fundamental component in AI for healthcare due to their ability to mimic the human brain's functioning. ANNs enable machines to recognize complex patterns and relationships within vast amounts of healthcare data, such as medical images, patient records, and genetic information. By leveraging ANNs, healthcare professionals can develop predictive models for disease diagnosis, treatment planning, and prognosis prediction. In this module, you will explore how ANNs can equip healthcare systems with the capability to enhance accuracy, efficiency, and personalized care delivery, ultimately leading to improved patient outcomes and resource optimization.

Learning goals

- Outline the metaphoric connection of artificial neurons and neural networks with the human brain.
- Restate the flexibility of artificial neural networks as universal function approximators and the procedure of training a neural network.
- Identify different layers and operations in neural networks and be able to select those tailored to distinct input data types.
- Learn different neural network architectures and link them with standard principles and algorithms for supervised and unsupervised learning.
- Understand in what way neural networks are data-hungry, and explain how transfer learning, data augmentation, and data-centric AI help to lower the data burden.
- Discuss the lack of transparency in neural networks, present the necessity for "explainable AI" and describe contemporary techniques for "explainable AI". In addition, and in relation to the data burden and privacy concerns, learn technical solutions in training neural networks for the privacy-conserving use of multi-centre data.
- Understand the reason of the domination of artificial neural networks in machine learning and AI today, and assess its impact for the future.

8.2 What's the buzz about

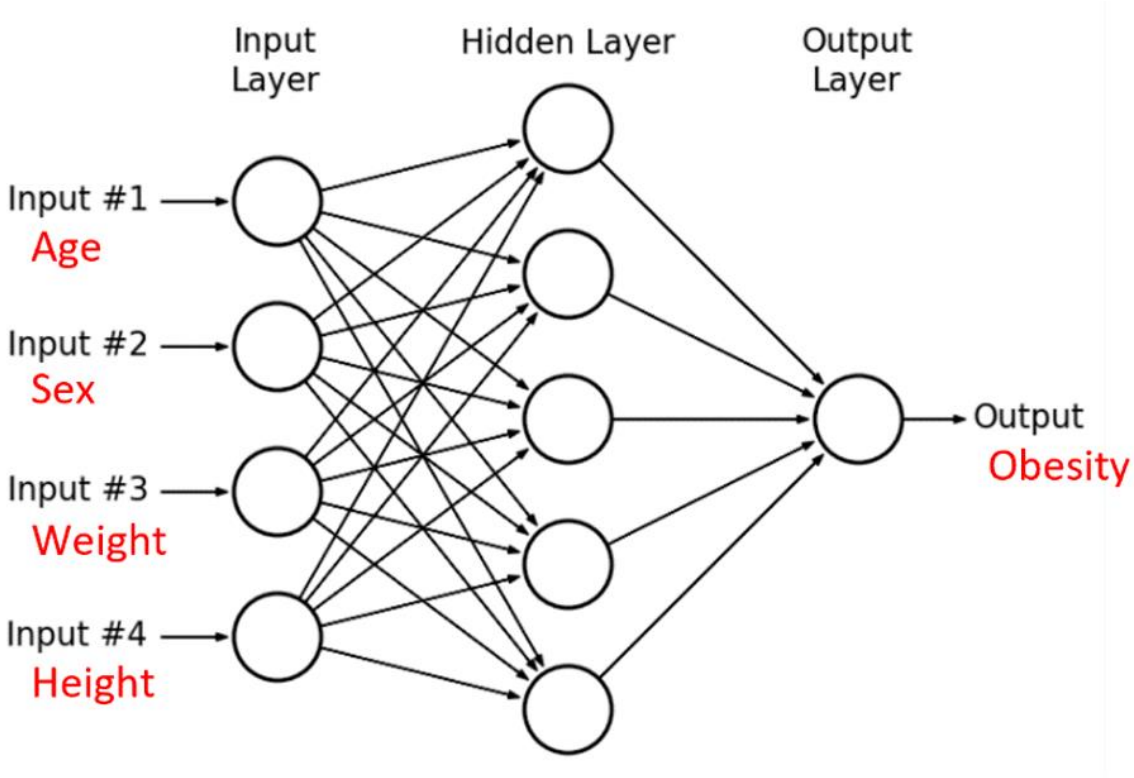
8.2.1 Artificial neural what?

What's the buzz about artificial neural networks? Let's explore it in this subsection of the module.

"Artificial neural networks" or "deep learning" are terms which may ring a bell. So what exactly are these? And why are these techniques so popular in AI?

Artificial neural networks (ANN) are a type of machine learning that loosely simulates the functions of biological neuronal networks. ANNs are composed of artificial neurons, which are aggregated into layers, processing information step by step. By training ANNs, we can learn complex structures and features from large quantities of data.

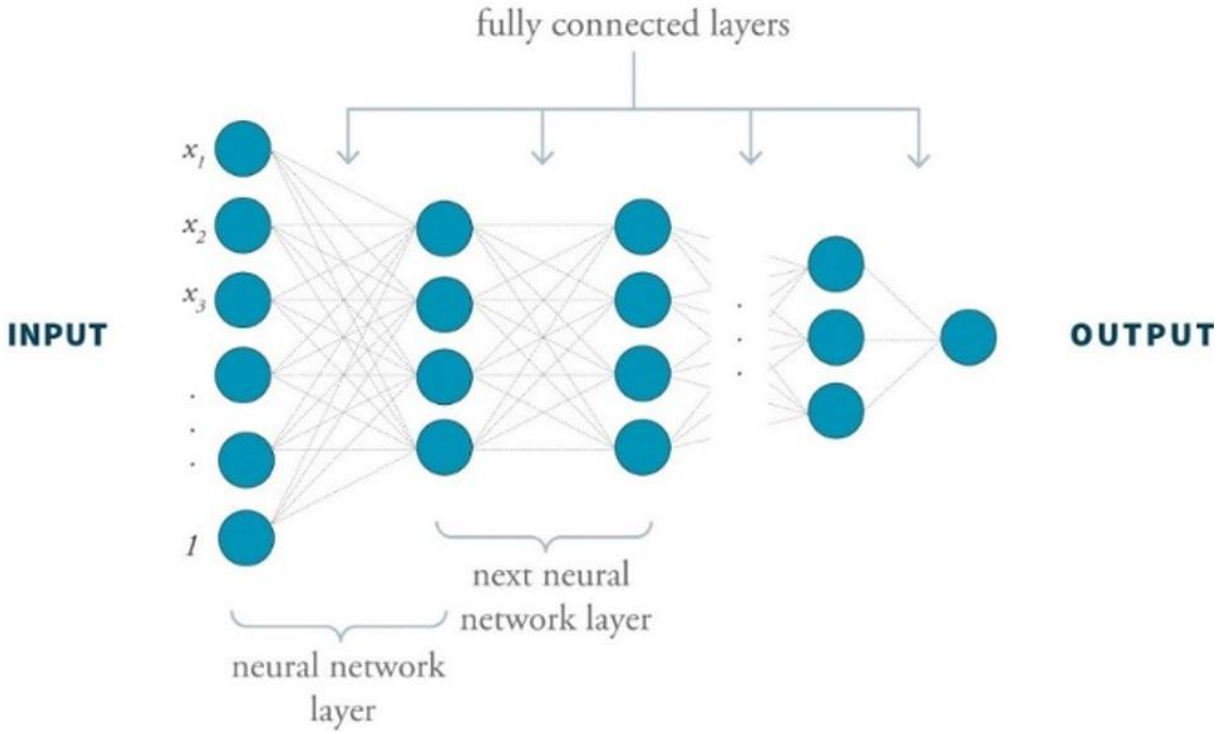
Take a look at the figure below, where the goal would be to detect presence of obesity. You might use a classifier that was introduced previously (e.g. SVM in Module 4) for solving this task, or you might use an ANN. The principle is the same: you aim to optimally estimate the parameters of the model based on available labelled training data.



In most traditional machine learning methods introduced in Module 4, the number of parameters is limited to the number of input features. With ANNs, there are many more parameters to be estimated, and as such the model can estimate more complex relationships. On the other hand, in order to reliably estimate all those parameters, we will need more training data for obtaining a robust model.

Deep learning (DL) is a term used to describe algorithms that learn in a purely data-driven manner with minimal interference of humans. This is achieved by learning the important features from data and the mapping from those important features to the output that we want. Deep learning models are machine learning models that organize parameters into hierarchical layers (see figure below). ANNs are the technology behind deep learning. "Deep" refers to the ability to add many hidden layers. By building

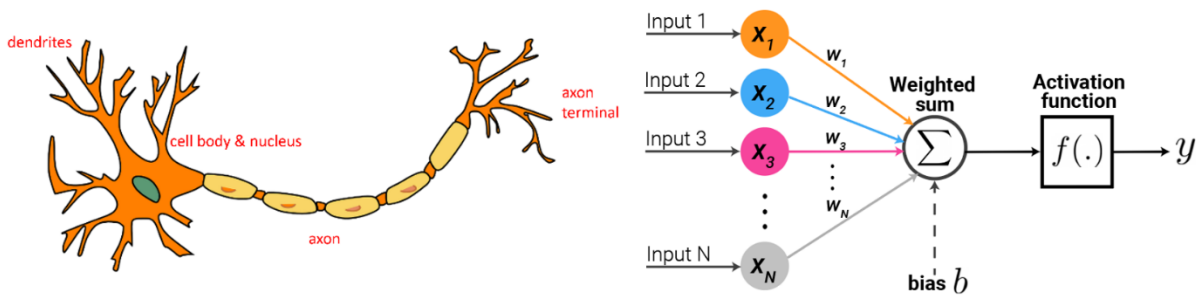
"deeper" networks with more layers, we can learn increasingly complex representations and information from "raw" input data.



For example, we can task a **neural network to find cardiac arrhythmia in EEG data**. In "classical" machine learning algorithms, an engineer would have to design ways to extract useful features from the EEG data. They would then train a machine learning algorithm that classifies segments of the signal as seizure or non-seizure, using these features as inputs. In the case of ANNs, we can directly feed the neural network with the raw EEG data and the seizure labels. The ANN learns by itself how to extract the relevant features and perform classification. The major advantage of using ANNs is that it saves time in finding suitable hand-crafted features. The disadvantage is that because of this, we humans do not know based on what knowledge the algorithm makes decisions. We call this the "**black-box" problem** of deep learning.

8.2.2 The artificial neuron

Why would we build artificial neural networks? The brain is a remarkably intricate information processing system capable of a vast array of intelligent actions. It thus provides a logical source of inspiration when we try to construct artificially intelligent systems. By simulating the underlying "subsymbolic" data processing at the neuronal level, we can try to recreate intelligence.



Given the comparison of a biological and an artificial neuron in the figure above, **can you identify the "dendrites" in the artificial neuron?** What about the cell body, and the axon terminal?

The structure of a neural network is an abstraction of how the brain works. This means that several inputs are processed by a neuron and the neuron will only "fire" (action potential analogy) whenever the right inputs are activated. However, this is where the analogy ends: internal workings of biological neurons are typically ignored, and the artificial neurons are much less complex than their natural counterparts.

In a mathematical way, a single artificial neuron is connected to multiple inputs x1...N, and a weight w1...N is associated with every input connection (see figure above). As such, a linear combination is calculated from the inputs, with only multiplication operators and a summation. A bias is added and the resulting value is fed into an activation function f, to form the output y. A single artificial neuron is also the simplest possible ANN, called a "single-layer perceptron". Its mathematical formulation is described as follows:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

On the next page, we will zoom into the detail of the activation function f. You might want to remind yourself about the logistic regression function, that transforms input data into categorically labelled data with an S shape.

8.2.3 Activation function

Activation function is typically a non-linear function. It describes whether or not the neuron is "activated", i.e. whether or not it outputs a signal. If a linear or no-activation function is used, the output y is just a linear combination of the inputs. The activation function is a critical component of neural networks and plays a fundamental role in their ability to model complex, non-linear relationships in data. Its importance can hardly be overstated, as it greatly influences the network's capacity to learn and make predictions.

Why do we need an activation function and which characteristics should it have?

To understand the necessity of it, we will go over an example together. Let's assume a very simplistic dataset where the input is the normalized blood pressure (mean = 0), and the output is a scale showing the danger that a patient might be in. We want to build a model for which we provide the blood pressure, which then returns the level of danger. For this cause, we have the following dataset:

Input (x)	Target (y)
-2	4
-1	1
0	0
1	1
2	4

If we would not consider an activation function, then our model would essentially be just a linear model. A linear model would mean that the relationships we can model should be of the form y = w*x + b.

We can see from the dataset table that for x = 0 or x = 1, then y = x, which would mean that w = 1 and b = 0. However, that solution does not work for the rest of our data. So, we need to find suitable

parameters with all data. If we would use a neural network without activation functions for this problem, the difference between estimate target and real target labels will always remain quite high and the network's performance quite poor. We realize that the solution is y = x^2. This function cannot be produced by a linear model. By adding the activation function to our neural network, we allow it to make such **more complex pattern discoveries**.

Activation functions introduce **non-linearity** to the network. Without non-linearity, a neural network would be reduced to a linear model, and it would be limited in its ability to capture and represent complex patterns in data. Non-linear activation functions enable the network to approximate a wide range of functions, making it suitable for a broader array of tasks.

Choosing an activation function

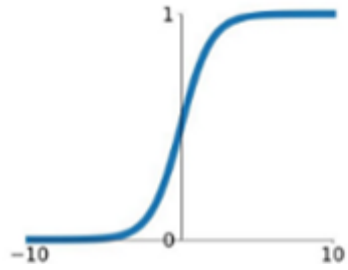
The **choice of activation function** impacts the expressive power of a neural network. Different activation functions have different properties, such as saturation (the range in which they remain nearly constant), and this influences the network's ability to model data. Some activation functions, like the **rectified linear unit (ReLU)**, have become popular, because they allow for efficient training. Different tasks and data may require different activation functions. For instance, the **sigmoid** or hyperbolic tangent (tanh) activation functions are often used in the output layer of binary classification problems. In contrast, the softmax function is suitable for multi-class classification. The ability to choose and customize activation functions makes neural networks adaptable to various problems.

The concept of activation functions in artificial neural networks was inspired by the way neurons work in the human brain. In **biological neurons**, activation functions, such as firing or not firing, play a fundamental role in signal processing. This bio-inspired design has led to the development of neural networks that can emulate certain aspects of human cognition, contributing to their success in tasks like image recognition and natural language processing.

In summary, the importance of activation functions in neural networks cannot be understated. They enable non-linearity, support gradient-based training, influence the network's capacity, allow for versatility in tackling various tasks, draw inspiration from biology, and contribute to regularization. Choosing the right activation function is a crucial step in designing effective neural networks, and it often requires careful consideration and experimentation to achieve the best results for a specific problem.

Let's discuss the following two activation functions:

Sigmoid activation function: In early neural networks, the sigmoid activation function was commonly used. The sigmoid has an S-shaped curve and is centred around x = 0. This made it suitable for outputting values between 0 and 1, which can represent probabilities in binary classification tasks. This property makes It particularly suitable for the output layers, where it is still mainly used. In the rest of the network two problems prevent it from being the default choice. The first is that it's a complicated function and so is its gradient, which makes it more computationally heavy especially when that function is used some million times in current big neural networks. The second issue is that it pushes both the input and the gradient to be of smaller values (e.g. if x = 100 then sigmoid(x) = 1). This also makes the gradients smaller and prevents the network from learning much. We will dive into this in the following section. More information about the sigmoid function was previously provided in the section of logistic regression.

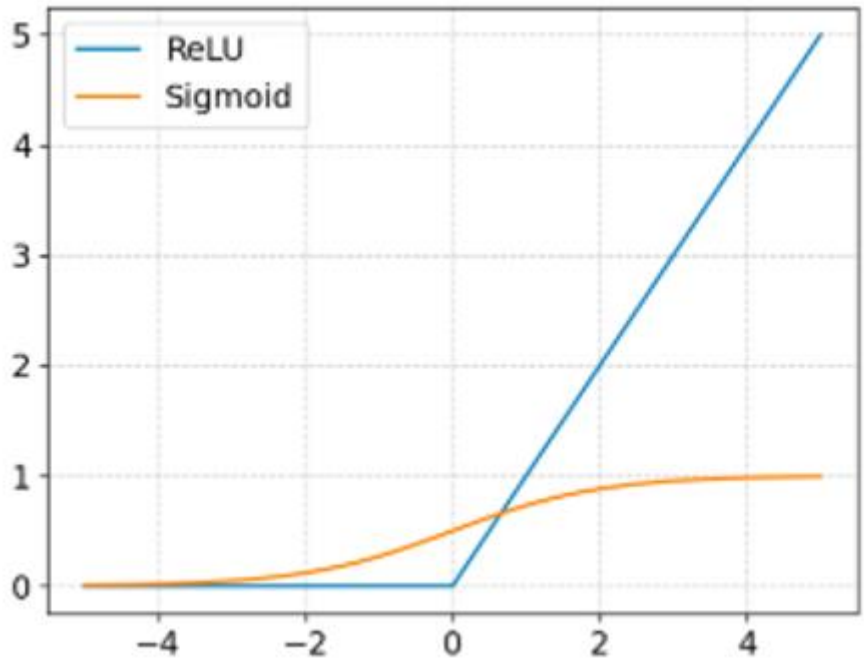


ReLU, which stands for **rectified linear unit**, is a popular activation function used in artificial neural networks, particularly in deep learning models. It is a simple yet effective mathematical function that introduces non-linearity to the network. To overcome both problems of sigmoid, ReLU was proposed. ReLU is simpler, namely if input is positive then output = input otherwise output = 0. Its gradients function is also very fast to be calculated, although it is not defined around the point 0. ReLU allows large values of input to maintain that property. At the same time, it makes a significant amount of output equal to zero, bringing further sparsity to the network helping both in faster computations and in the learning.

The ReLU activation function is defined as follows:

$f(x) = \max(0, x)$

In other words, it takes an input value x and returns x if x is positive or zero, and it returns zero if x is negative. Visually, the ReLU function looks like a piecewise linear function with a slope of 1 for positive values and zero for negative values. This makes it computationally efficient and easy to implement.



Here are several key characteristics and benefits of the ReLU activation function:

- **Non-linearity**: While ReLU is linear for positive values, it introduces non-linearity by setting negative values to zero. This non-linearity is crucial for neural networks to learn and represent complex, nonlinear relationships in data.

- **Sparsity**: ReLU encourages sparsity in the network, meaning that only a subset of neurons is activated for a given input. This sparsity can lead to more efficient and less redundant representations, which can be beneficial for both memory and computation.
- **Ease of optimization**: The gradient of the ReLU function is straightforward to compute. This makes training deep neural networks using ReLU much more manageable than some other activation functions (like the sigmoid or tanh), as it helps mitigate the vanishing gradient problem. We will come back on the importance of the gradients in the next section.
- **Efficiency**: ReLU is computationally efficient, as it involves only simple mathematical operations. This efficiency makes it suitable for training large and deep neural networks.

In practice, ReLU has become the default choice for many neural network architectures, especially in deep convolutional neural networks (CNNs) used for computer vision tasks. It has played a significant role in the success of deep learning and has become a fundamental building block in modern neural networks. Looking at the figure above, we can observe that the Sigmoid function tends to keep values close to 1 for high input numbers, limiting the output range. On the other hand, the ReLU activation function is 0 for input values less than or equal to 0, encouraging sparser solutions by selectively turning off certain neurons in neural networks.

We have now described a single neuron. ANNs are networks composed of many such neurons. The output of a neuron in turn serves as the input for the neurons in the next layer. Multiple neurons can thus be stacked on top of each other, and their outputs can be combined into a single output node to form a multi-layer perceptron neural network. The stacked neurons are called the hidden layer. An ANN could contain multiple hidden layers to extract more complex information. Let's learn about it in the next subsection.

8.3 The neuron and its networks

8.3.1 Multi-layer perceptron

In this new subsection of the module, we will delve into the **neuron and its networks**.

Firstly, let's discuss the concept of multi-layer perceptrons.

Multi-layer perceptron (MLP) is a type of artificial neural network (ANN) composed of multiple layers of nodes, with each layer fully connected to the next one. It consists of an input layer, one or more hidden layers, and an output layer. MLPs are capable of learning complex patterns in data and are widely used for tasks such as classification and regression in machine learning.

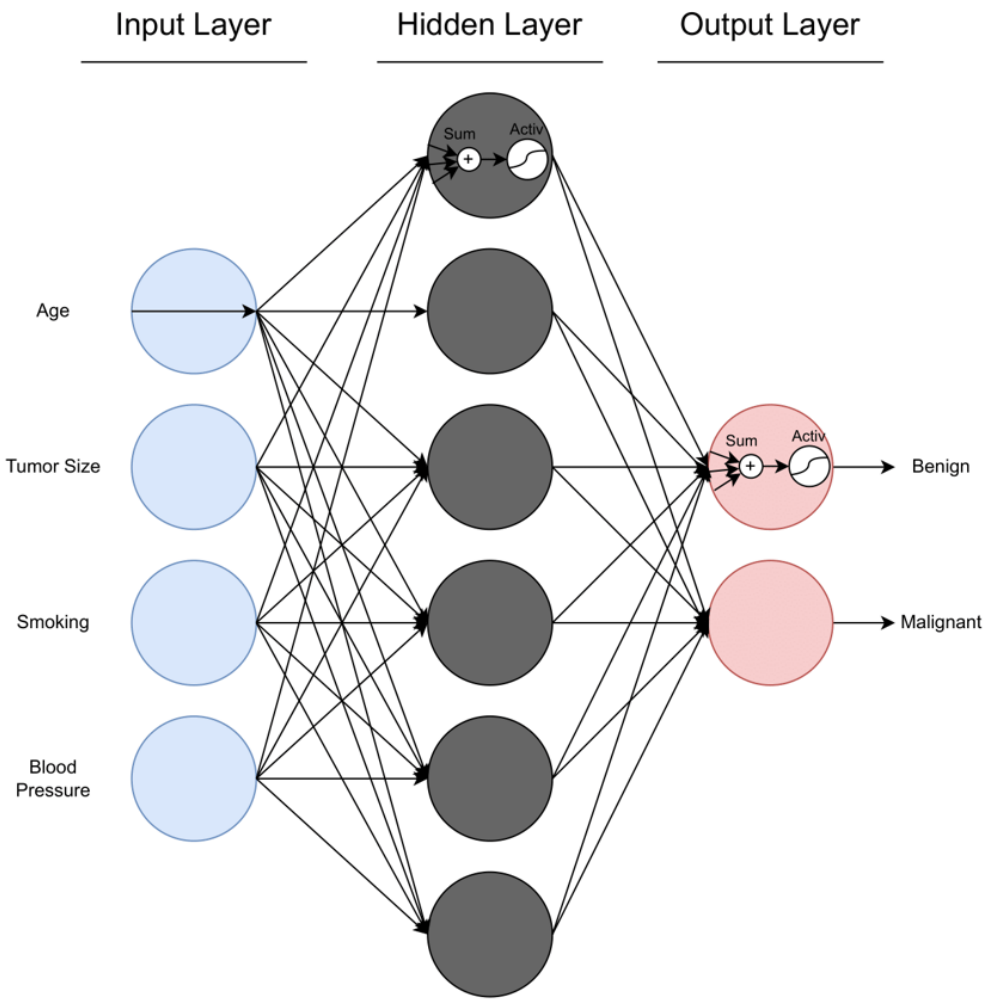
An MLP is a versatile and powerful ANN architecture that builds upon the foundational concept of a single perceptron. Unlike a single-layer perceptron, an MLP consists of **multiple layers of interconnected "neurons"**. These layers work together to process and understand complex data. According to the Universal Approximation theorem, an MLP can estimate almost any continuous function with finite data. These functions could be e.g. the translation of a patient record to possible disease, the suggestion of treatment based on patients' examination and possibly any combination of input output provided.

However, there is a catch – you might **need a lot of data and many perceptrons** in those MLPs to get the function right. This makes it particularly challenging for the learning process to discover the perfect solution. Also, gathering a lot of data is an expensive and difficult process, especially in the medical field.

In the context of healthcare, an MLP is an advanced type of ANN. Think of it as a sophisticated version of the perceptron, a building block for AI. Unlike a single-layer perceptron, an MLP consists of multiple layers of interconnected neurons. These layers work together to process and understand complex medical data.

Each MLP consists of three categories of layers according to the position of the layer:

- **Input layer**: Think of the input layer as the entry point for data. Each neuron in the input layer represents one piece of data, one feature for each particular datapoint. For example, one can show a specific symptom, another a specific medical history (comorbidity) and another a test result.
- **Hidden layers**: In these layers, input data are processed to derive their combination that would reveal the best output results. Each hidden layer neuron is a perceptron. Patterns between the input and the output are created, for example correlations among symptoms and diseases. The more hidden layers, the more complex relationships the model can represent.
- **Output layer**: The output layer provides the final results. It also has perceptron neurons but this time, a specific amount regarding the target we want to achieve. If the target is to predict one number, then it's just one neuron. If it's to predict 10 classes, then it's 10 perceptrons.



8.3.2 Training

Having discussed the basics of neural networks, we will now explore how these models actually learn from data. We will cover the **training process**, evaluate how well the models perform, and discover ways to make them generalize better.

For instance, let's consider a task where we need to **sort biopsy samples into categories**. This is a classification task. Imagine that we have 1000 tumour biopsy images from different patients, for which we already know whether the tumour was benign or malignant. Our aim is to build a model that, based on these 1000 images, can predict the category for future patients. We will assume that our model takes an image as input, and gives back the probabilities for each of the categories. The model consists of a predefined structure (e.g., MLP) with some parameters that need to be trained.

The first step would be to split our data. We need three groups of data, each used for a different purpose:

1. **Training set**: this is used to adjust the parameters of the model so that the model gives the most accurate predictions.
2. **Validation set**: this set is exploited to decide the values of the hyperparameters of the model and when the model generalizes well. The hyperparameters can be several choices we make when designing the model and are constant during a single training, for example the number of layers of a model. Deciding these values on our validation set prevents the model from selecting the combination that memorizes the training set and hence overfits.
3. **Test set**: this is the one that will give us an unbiased estimation for our model performance.

A few remarks about the split of sets are to be made:

- It is important that each set is as independent to the others as possible. For example, there should not be biopsies of the same patient in more than one set. That would lead to exaggerating the abilities of your model and hence not perform so well on new data.
- Moreover, a good practice is to avoid testing your models very frequently on the test set, since the choices for the model are then based on the test set and are considered biased.
- For the split itself, we usually choose the bigger part to be the training set, while we frequently choose equal size for validation and test set. For example, a typical 70-15-15% is used. Though in case the dataset is too small, we should think about k-fold cross validation. For more information on the training and approaches for validation and data splitting, see Module 4.

Training the model parameters

Now that we have introduced how we can build a basic deep learning model, we can also provide a high level overview on how we will train the model parameters:

1. First, rather than having a split of our data in train and test data, we will split our data into train, validation, and test datasets
2. We will use the training dataset:
    1. And make a prediction based on the sample's features with the current model parameters.
    2. We will compute the loss between the model's prediction and the sample's label. The loss is a numerical value representing how far the prediction is from the label. Low loss is good, and high loss is bad.
    3. The model will then update its parameters in a way that will reduce the loss it produces the next time it sees that same sample. This is known as an optimization step.

3. As the model will not be fully learned (or have converged using the professional jargon), we will iterate the previous steps. From time to time, for example after we have taken a pass through all our training dataset, we can evaluate our model on a **validation** set:
    1. In this phase, we assess if the parameters the model has learned, produce accurate predictions on data that it has not yet observed, in other words the validation set.
    2. The model does not learn from these samples, because we do not execute the optimization step during this phase. Without the optimization step, the model cannot update its parameters, which in turn prevents learning.
    3. The validation set is a measure of how the model will do "in the real world". We save a version of the model if it gives us the best validation performance we have seen so far.
4. We have now mentioned a critical component for optimizing the model, which we will elaborate further on in the next paragraphs: the choice of loss function, and we will continue with a small introduction to optimization theory.

8.3.3 Loss functions

On the previous page, so far we've only talked vaguely about the goal that our model of the biopsy example should achieve. It should predict accurately whether the biopsy shows a benign or malignant tumour. However, accuracy ignores the probability estimation the model returns for the biopsy image to be in one or the other classes. Usually an **entropy function** is used which, when minimized, pushes the probability of the correct class to be as close as possible to 1 (or 100%) while reducing the rest to 0. This function is called the **loss function**.

Loss function is a mathematical function that measures the difference between the predicted values of a model and the actual values of the data it's trained on. It quantifies how well the model is performing by providing a measure of the error or discrepancy between predictions and ground truth. The goal during training is typically to minimize this loss function, thereby improving the model's ability to make accurate predictions.

The loss function differs for different tasks such as classification or regression, and is used to assess how well a model with specific parameters performs. Overall, changing (updating / improving) the parameters of the model to minimize the loss, produces a model that is capable of predicting the class of new datapoints better.

Multiple loss functions exist, and some are better adapted for some problem types than others. Some factors to consider:

- What kind of problem are you solving?
- Are all datapoints equally important or should we pay less attention to outliers?
- Are the number of samples we have in each category roughly equal? (for classification)

Mean squared error

A loss function you should be familiar with is **mean squared error (MSE)**, which we also used for regression analysis. To calculate the MSE, you take the difference between the model predictions and the true label, which is also known as the ground truth, square it, and average it out across the whole dataset. Squaring gets rid of the sign (+/-) of the difference between the prediction and the ground truth and emphasizes outliers.

Cross entropy loss

One of the most common loss functions in classification with deep learning is the **cross entropy loss**. It estimates the difference in the probabilities of the predictions. The function is simple: it sums the negative log of the model's predicted probability for the ground truth class. Because probabilities are between 0 and 1, the log value is some negative number.

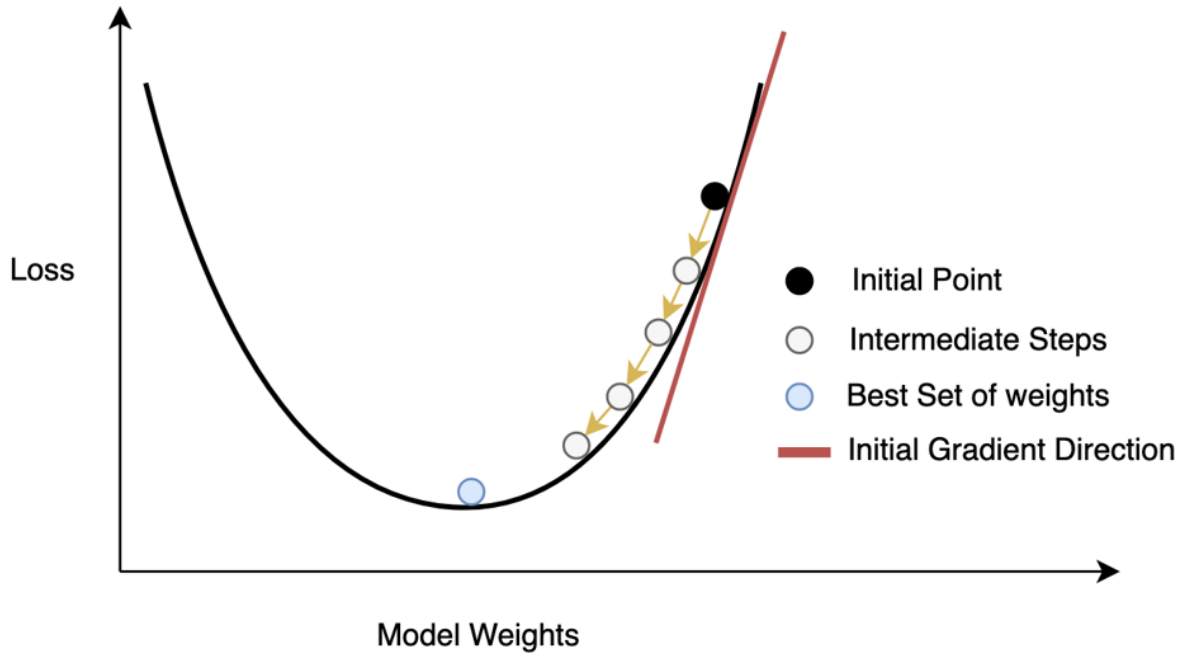
Why is this a good classifier?

- The log of a value that is close to 0 is a large negative number. Because we are using the negative log, this flips to being a large positive number.
- The negative log of a value that is close to 1 is close to 0.
- These dynamics are in line with what we need from a good classification loss function. In order to achieve a low loss, a classifier will have to produce probabilities for the ground truth class that are close to 1.

8.3.4 Gradient descent and optimization algorithms

Gradient descent (GD) is an optimization algorithm used in machine learning to find the optimal parameters (weights and biases) of a model that minimizes a given loss function. It is a fundamental technique in training neural networks, including multi-layer perceptrons (MLP). GD is a process of iteratively updating model parameters to minimize the loss function. It uses the gradient of the loss with respect to the parameters.

The **gradient** is essentially a vector that points in the direction of the steepest increase of the loss function (see figure below). By taking small steps in the opposite direction of the gradient, GD aims to reach a minimum of the loss function. The magnitude of that step is defined by one of the network's hyperparameters, the **learning rate**. This hyperparameter is among the most important ones, since usually the networks are quite sensitive to it. Increasing its value might lead us to diverge from the solution.



In the figure above, the black dot shows the initial set of weights, which we randomly picked. Based on the gradient direction, we choose how to update our parameters and move to the next intermediate step. Gradients offer this information about the local direction that the loss function is minimizing. This process iterates for every intermediate step until the model converges to the best set of weights.

8.3.5 Backpropagation

In this unit we will explore how we calculate the gradients that give us the weight updates of a neural network.

Backpropagation is a key algorithm used in training artificial neural networks. It involves calculating the gradient of the loss function with respect to the weights of the network, which allows for adjusting the weights in a direction that minimizes the loss. Backpropagation is a two-way process that is based on the chain rule trick that helps us avoid repeating many computational burdens.
Chain rule is a fundamental concept in calculus that plays a pivotal role in the calculation of gradients (derivatives) in neural networks. In a neural network with multiple layers and interconnected neurons, the chain rule allows to "chain" together the derivatives of each layer to compute efficiently the overall gradient of the loss with respect to the model's parameters. It would be computationally impossible to calculate each gradient on its own, while with the chain rule, we can partition the calculation and leverage the many repeating parts in the calculation of the gradients of each parameter.

To give an example, if we have an MLP with the loss, then for each weight we have its gradient being:

gradient = output gradient * input

We need to be careful that in the output gradient, we also have the gradient of the activation function. So now you understand better that difficult calculations of gradients mean computationally heavy training.

In the previous formula, we can see that it is important to know both the input and output gradient for the calculation of a parameter gradient. Current tools for **neural network training**, retain the input and output values of each layer in the memory so that we don't need to re-compute them. To understand how they do it, we need to explain the two (forward and backward) phases of the network passing that facilitate the training, and also the further steps taken:

1. Forward pass

The training process begins by giving the neural network some input data and calculating some predictions for them. This is what we describe as a forward pass. Unless mentioned, the network parameters are random so initially predictions are also random, and on every iteration we expect them to improve. During the forward pass, we calculate and save all the input values to each of the network layers.

2. Error calculation

After making a prediction, the network calculates how far off its prediction was from the actual correct answer. We have already described that many loss functions can play this role depending on our final goal.

3. Backward pass

Due to the chain rule formula, we need the gradient of the output to calculate the gradient of the current layer. We can then start backwards and calculate the gradient of the last layer that needs only

the gradient of the loss, then the one before etc. This process is called backward pass. When this is done for all the layers in the network, we have achieved calculating the gradients for each parameter.

4. Weight updates

With the gradients calculated, the network updates its weights using an optimization algorithm (e.g., gradient descent).

new weight = previous weight - a * gradient

The goal is to adjust the weights in a way that minimizes the error, making the network's predictions more accurate. The parameter a is called the learning rate or step size.

5. Iterative process

Steps 1 to 4 are repeated many times over multiple iterations (epochs) on the training data. With each iteration, the network fine-tunes its weights, gradually reducing the error.

8.3.6 Gradient updates in practice

Gradient descent (GD) in its basic form computes gradients by considering the entire dataset for each parameter update. However, in practice calculating a gradient for each of the datapoints and deciding our weight update based on all of them, would be quite computationally impractical and leads to slow convergence.

Therefore, a shortcut was suggested as a solution where we decide each step on a subset of the whole data. This subset is called a **batch**. Each update weight is based on the loss of each of the datapoints consisted in a batch. Under current hardware, the calculations for each datapoint within a batch are done concurrently. This approximation of the total dataset gradient through the batches is called **stochastic gradient descent (SGD)**. By using the smaller subsets, the updates are more frequent, and each update carries noise due to the smaller sample size. This noise has shown in practice that it can actually help the optimization process.

So, in essence, GD needs to be in batches for training complex models like neural networks on extensive datasets.

8.3.7 Convergence

Convergence: A model is considered to have converged when it has reached a state where further training is unlikely to significantly improve its performance on the training dataset. Convergence implies that the model has learned the underlying patterns and relationships within the training data to a satisfactory degree.

To verify that your model is converging, is to observe several indications in your training set:

- The **loss function**, which quantifies the difference between the model's predictions and the actual target values, should stabilize. In most cases, it should decrease and eventually reach a plateau. When the loss stabilizes, it suggests that the model is no longer making significant improvements.
- **Performance metrics**, such as accuracy, should also stabilize or show diminishing improvements. These metrics indicate how well the model is performing on the training data.
- Another indication can be, when the **gradients** of the loss function with respect to the model's parameters (weights and biases) become small. Small gradients suggest that the model is close to a local minimum of the loss function.

- Finally, running the training process multiple times with the same hyperparameters and different random parameters should result in similar or nearly **identical outcomes**. This indicates that the model consistently converges to a similar state with regards to the training set.

Even if our model converges on the training set, it might still be suboptimal to what it could achieve. There could be overfitting, where the model memorizes the data without actually learning much about them. Avoiding such a condition would require the exploitation of the validation set during training. We validate our model at the end of every epoch, and we keep the model of the epoch that performed the best in the validation dataset. If the model does not improve on the validation for some number of epochs, then we can consider it converged and stop the training. This process is called **early stopping**.

8.3.8 Hyperparameters for neural networks

Hyperparameters are settings or configurations that are not learned from the data, but are essential to control the learning process of a machine learning model. These parameters influence the model's behaviour, performance, and convergence during training.

In deep learning, there are many more hyperparameters that need to be estimated. Some common hyperparameters include le**arning rate, batch size, the number of hidden layers in a neural network, and the number of epochs**. The learning rate governs the step size in gradient-based optimization, and update steps measure the progress of training. Epochs determine how many times the entire dataset is used during training. Tuning these hyperparameters effectively is crucial for achieving optimal model performance.

We described the importance of the tuning of hyperparameters in Module 4.

Hyperparameter tuning is the process of finding the best combination of hyperparameters for the machine learning model. It is usually done using the separate set of the dataset: the validation set. There are many ways to decide the range of values for each hyperparameter. In practice, a small grid search usually achieves satisfying results. This means that a random subset of possible hyperparameter combinations is chosen, and models are optimized. We then keep the hyperparameter combination with the best performance on the validation set.

8.3.9 Overfitting and regularization

In Module 4, we learned about overfitting, for which early stopping can be a way to avoid it. In the case of neural networks, there are many more parameters to be estimated than with classical machine learning techniques. As such, the **risk of overfitting is larger**. Fortunately, there are ways to prevent a network from memorizing the training datapoints and learning irrelevant noise:

Regularization in AI is like a teacher guiding a student to focus on the essentials and to not overcomplicate things. In the context of machine learning and neural networks, when a model learns from data, it might get too fixated on the training details, including noise and outliers, which don't apply to other situations. Regularization techniques are the methods used to help the model generalize better to new, unseen data, rather than memorizing the training data. It's like adding constraints or penalties to the learning process to keep the model simple and focused on the main patterns, preventing it from getting distracted by the noise or making it overly complex. This way, the model can perform better on new data, not just the examples it was trained on.

Regularization techniques come in various forms. They can be additional terms to our loss function, normalization techniques in between the layers, or mechanisms that introduce challenges to the model's learning process.

Weight penalization

One key contributing factor to overfitting is the presence of large weight values within the network. When neural networks have large weight values, they can assign an undue amount of importance to individual features or data points during training. Essentially, the model becomes hyper-focused on specific details within the training data, even capturing noise and irrelevant patterns. This hyper-specialization causes the neural network to perform exceptionally well on the training data but poorly on new, unseen data.

Regularization techniques, such as L1 and L2 regularization, address this issue by **penalizing large weight values**. L1 regularization adds the absolute values of the weights to the loss function, promoting sparsity in the model. L2 regularization adds the squared values of the weights, encouraging smaller weight magnitudes. By encouraging smaller and more balanced weight magnitudes, these techniques help neural networks generalize better to unseen data and reduce the risk of overfitting. Therefore, the careful management of weight values is essential in ensuring that neural networks in medical applications provide reliable and accurate results.

Dropout

Another way to avoid such effects from single features is to randomly **mute them during training**. This method is called **dropout**.

During the training phase, dropout **randomly deactivates a subset of neurons** (or units) in a neural network layer at each iteration of the training process. The "dropout rate" specifies the probability that any given neuron is omitted from the computation during a training iteration. This rate is typically set between 20% and 50%. By randomly dropping out neurons, the network is forced to learn more robust features that are useful in conjunction with many different random subsets of the other neurons. This prevents the network from relying too much on any single neuron and helps it to generalize better to unseen data.

The process of dropout can be interpreted as training a large number of thin neural networks with shared weights, and averaging their predictions at test time. Each training iteration, a different "thinned" network is trained with a random selection of neurons being omitted. At test time, dropout is typically not applied; instead, the weights learned by the neurons are scaled by the dropout rate to compensate for the fact that more neurons are active than during training.

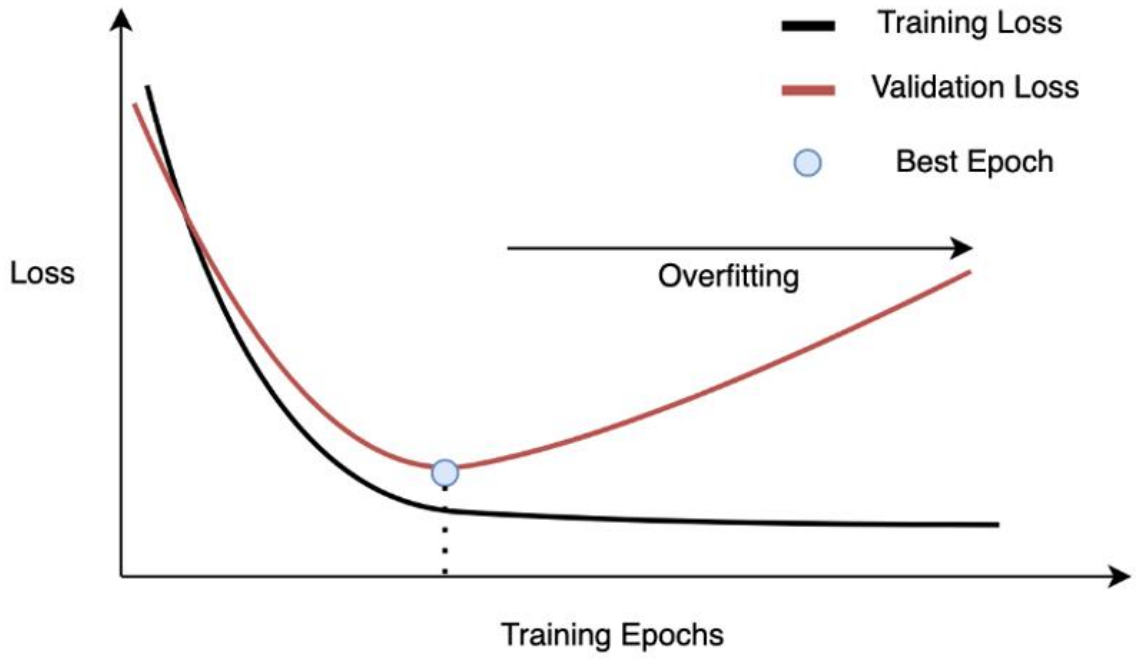
Dropout has been shown to be an effective and simple technique to improve the generalization of neural networks, especially in deep learning models where overfitting is a significant concern. It's widely used in various kinds of neural networks.

8.3.10 Visualisations and monitoring

Finally, the last method that we need to mention is the **debugging capabilities** we have through the different **visualization and monitoring tools**.

- These tools provide insights into the **model's performance during training** and can help identify signs of overfitting in real-time. By closely monitoring key indicators like training and validation loss, as well as performance metrics such as accuracy, you can catch overfitting as it begins to occur.

- Moreover, visualization tools allow you to **explore the internal representations** of the model, inspect the activations of individual neurons, and visualize feature maps in convolutional neural networks (CNNs). These capabilities provide valuable diagnostic information. If, for example, you notice that specific neurons are consistently firing for a particular class in your neural network, it might be an indication of overfitting to the training data. Here, we will not dive deeper in these methods of internal representations, but we mention them for completeness. We will learn more about explainable AI in Module 10.



In the provided figure above, we observe a line graph representing the **training and validation loss** as a function of the number of training epochs. The vertical axis measures the loss, which quantifies the difference between the model's predictions and the actual target values, while the horizontal axis indicates the progression of training over a series of epochs. Initially, both the training and validation loss curves exhibit a consistent and gradual decrease, reflecting the model's successful learning process. This decline is an encouraging sign, suggesting that the model is becoming more proficient at fitting the training data.

However, as the training progresses, a significant development emerges: a noticeable divergence between the training and validation loss curves. While the training loss continues its downward trend, the validation loss, depicted by another curve, begins to increase. This divergence indicates that the model, while excelling in minimizing errors on the training data, is encountering difficulties when exposed to new, unseen data. The rising validation loss suggests that the model's generalization capability is diminishing, as it's becoming overly specialized in remembering the training data. Early stopping then would help us choose the model in the state that produces the best validation results, and therefore would reduce the effect of overfitting.

Now that we have discussed the neuron and its networks, let's move onto the next subsection.

8.4 Operations, layers and architectures

8.4.1 Convolutional neural networks

In this new subsection of the module, we will dive deeper into **operations, layers and architectures**. Let's start with discussing convolutional neural networks.

Convolutional neural networks, commonly referred to as CNNs or ConvNets, are a specialized type of artificial neural network designed for processing grid-like data, such as images and video frames. They utilize a specialized architecture with convolutional layers, which apply filters to input data, enabling the network to automatically learn hierarchical patterns and features.

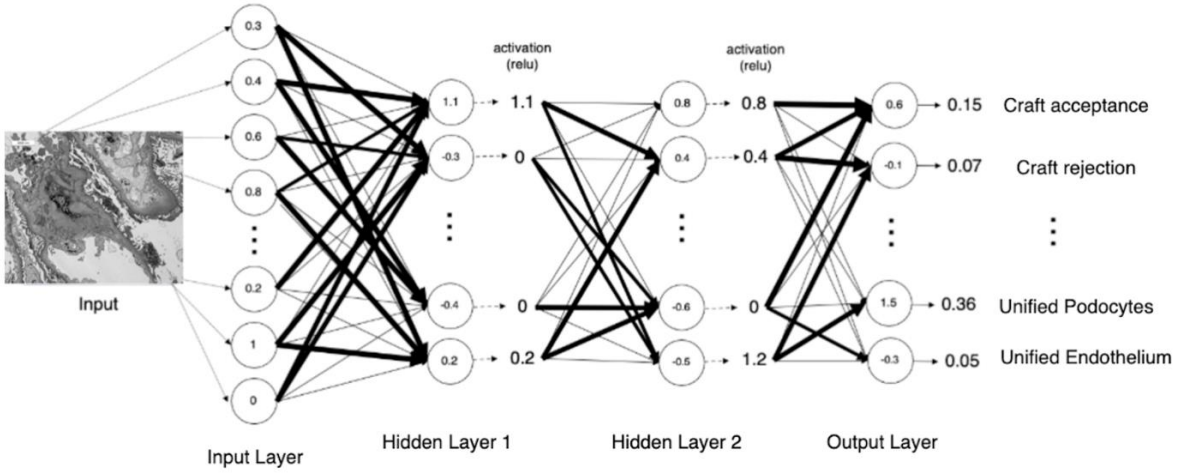
CNNs have revolutionized various fields, including computer vision, and are extensively used in healthcare for tasks like medical image analysis and diagnosis.

CNN versus MLP

For large input data, such as high-resolution images in medical imaging or video frames, using an **multilayer perceptron (MLP)** would lead to an explosion in the number of parameters. This can result in excessive computational requirements and a higher risk of overfitting. **CNNs handle large inputs more efficiently** by processing them in small, manageable chunks. The following example signifies the **problem of MLP with images**:

In images, each pixel can be considered as one variable, one feature that can take values from 0 to 255. Placing a weight on each pixel would immensely increase the number of parameters. Some calculations that can be done quickly to understand the scalability of this, are the following:

Assume that we work with high definition images 1024 x 1024 pixels, and our MLP has two layers with hidden dimension being 64. Then we can understand that the parameters of the first layer would be 1024 x 1024 x 64 $\simeq$ 67 million parameters. That is probably already a lot to fit in most common graphics processing units (GPUs). The problem becomes way bigger if we think about 4K images (4096x4096 pixels) and MLPs with more layers.



In the example shown in the figure above, a training image data is fed as input to the input layer. The dimensions of each image are 32×32 pixels. The numbers at every unit in the two hidden layers represent the summation of weights and input from all the units in the previous layer. There are 200 units in first hidden layer and 150 units in another hidden layer. This is why the hidden layers are called dense.

Dense layer, also known as a fully connected layer, is a specific type of hidden layer where each neuron is connected to every neuron in the previous and next layer. It's one of the most common types of layers used in neural networks.

Dense layers are typically used to change the dimensions of your vector space and can learn patterns among the input features. They are fundamental in networks dealing with non-sequential data (though they can be used in sequential models too, often before the final output).

Convolution filter

In computer vision and signal processing, the problem of input data overload has been faced with filters. **Filters** are usually pre-specified patterns that are passed through the data, thereby revealing whether that pattern is present or not. They are shifted through the data to show where the pattern appears and how similar it is.

Now we understand that the problem of large input dimensionality is faced by applying the same filter, sliding it through the whole data. This choice grants one of the benefits of CNNs in comparison with MLPs: translation invariance.

Translation invariance refers to the network's ability to recognize patterns or features in the input data regardless of their exact position or location.

To illustrate, if a specific feature is present in one part of the input, the CNN can detect and identify the same feature in a different part of the input without being overly sensitive to its precise location. In contrast, MLPs lack this inherent ability. In an MLP, each group of neurons focuses on pixels from distinct regions. If a pattern appears in both for example the top right corner and the bottom left corner of the input, MLP neurons would need to learn it independently. Therefore, If the same pattern would appear later in a new location then the network wouldn't be able to recognize it.

Another question that arises, is: **how many of such filters can we apply**? How many are enough in case we have many distinctive patterns? Indeed, in most real-world problems the number of patterns is so large that they cannot be applied directly. Imagine how many filters we would need to perform face recognition on a large population, especially if we consider where the person looks or how far the face is from the camera. This issue is solved similarly as before in MLPs by creating networks with more layers. In the first layers of a CNN, simple patterns are detected such as lines, edges or corners and as layers progress more distinct patterns appear. This property is called **hierarchical features**.

8.4.2 Pooling

Previously, we have described how sharing the parameters along the spatial dimensions, can reduce the number of parameters. Now, let's delve into the outcomes of these processes. The output closely mirrors the original image's size, meaning that integrating information from distant image regions might necessitate either extensive kernel sizes or numerous layers. This way, features processed by filters from opposite ends of an image can converge.

An effective strategy to address this challenge, is **pooling layers**.

Pooling is a technique used in convolutional neural networks (CNNs) to reduce the spatial dimensions of the feature maps generated by previous convolutional layers. By systematically subsampling the feature maps, pooling helps to manage the computational load and reduce further the number of parameters, while also addressing the risk of overfitting.

Pooling layers are crucial in **medical image analysis**, as they help compress the spatial dimensions of feature maps produced by preceding convolutional layers. Through strategic subsampling, pooling significantly cuts down on computational demands and further minimizes parameter count, which is pivotal in reducing the likelihood of model overfitting.

This is particularly vital in healthcare, where the accuracy and generalizability of models can directly impact diagnostic outcomes. Pooling achieves this by preserving only the most vital information within a feature map, selecting either the highest value (Max-Pooling) or the average value (Average-Pooling) from each segment, often a 2x2 or 3x3 window. The most important information within a feature map often relates to characteristics that can help in identifying, diagnosing, and understanding diseases or conditions. These characteristics vary depending on the specific application (e.g., detecting tumours in MRI scans, identifying fractures in X-rays, or spotting lesions in dermatological images).

This method of down sampling not only bolsters computational efficiency but also strengthens the model's ability to identify relevant features irrespective of their precise location in the image. Such translation invariance is especially **beneficial in healthcare applications**, where the ability of CNNs to recognize pathological features or anomalies across varied imaging conditions and positions, can support more accurate and reliable diagnoses.

Non-linear operators and regularization

All the non-linear operators described previously can be applied without further modification to CNNs. It is substantial to include a non-linear operator, e.g. ReLU or sigmoid, to make sure that we can extract a non-linear combination from the input features/pixels. Batch normalization, weight decay and dropout can be directly used in CNNs as well.

Overfit on CNN

Parameter sharing, pooling operations and local feature extraction in CNNs play a pivotal role in mitigating overfitting. By **sharing the parameters** along the spatial dimensions across different regions of the input data, and by aggregating information from neighbourhood areas, we discourage the network from memorizing noise or irrelevant details. Moreover, small filter kernels promote **local feature extraction**, capturing meaningful patterns and structures in small, localized regions of the input, which contributes to their ability to learn hierarchical representations of data. Together, these techniques make CNNs particularly robust and well-suited for complex tasks like image recognition, where overfitting can be a significant concern.

8.4.3 Recurrent neural networks

Another class of neural networks that handles primarily sequential data, is **recurrent neural networks (RNNs)**. Unlike feedforward neural networks (like MLPs), RNNs possess a unique capability to capture temporal dependencies and context within data.

Recurrent neural networks (RNNs) are a type of artificial neural network designed to process sequential data by maintaining internal memory. They are capable of capturing temporal dependencies in data.

RNNs are inspired from the way that the brain recalls previous memories to operate on a current situation. RNNs keep an **internal memory** that writes down information from already passed iterations. RNNs iterate over a known-structure of sequence, for example in natural language processing (NLP) words-sentences, or in biomedical time-series EEG, ECG or examination information about a sequence of patient visits. Over these iterations, a part of the network processes the current step, and another

part divides which information should be kept in memory and which of the memory should be discarded.

Challenges

RNNs have traditionally been the go-to architecture for processing sequential data, given their inherent design to remember previous inputs in the sequence. This capability has made them highly effective for applications like next-word prediction in smartphones or voice recognition in assistants. Despite their widespread use, RNNs grapple with certain limitations, particularly in **handling long sequences**. The issue arises from their tendency to forget details about inputs received many steps earlier in the sequence, a consequence of continuously updating their internal memory to incorporate new information.

A noteworthy example that illustrates how these challenges can be addressed in a specific domain, is the application of sleep staging. **Sleep staging**, the process of classifying the stages of sleep based on physiological signals, involves analysing long sequences of biomedical signal data. RNNs usually fail to achieve good scores in sleep staging. Specialized network architectures that do not only leverage the temporal continuity in sleep data, but also incorporate mechanisms to effectively retain and utilize information over long sequences, demonstrate enhanced ability to remember significant patterns across the entire sleep cycle, improving the accuracy of sleep stage classification compared to traditional RNNs.

The success of such models in sleep staging highlights a tailored solution to the inherent memory limitations of RNNs, showcasing the potential of specialized architectures in overcoming the challenges of long-sequence processing. This example underscores the importance of developing and applying model architectures that are specifically designed to address the unique characteristics and requirements of the task at hand, particularly in complex and critical fields like healthcare.

8.4.4 Transformers

Transformers are yet another class of neural networks for sequential processing. They can better handle some of the issues of RNNs. Instead of processing each sequence step after the previous one, transformers process them all simultaneously.

They achieve this processing by first mixing all the sequence steps information, and then by using the same network on each sequence part. For this mixing of sequence information, transformers employ an **attention mechanism** that enables them to selectively focus on relevant parts of the input data. Picture a spotlight that automatically illuminates the most pertinent information, facilitating contextual understanding.

Transformers versus RNNs

What sets transformers apart, is their ability to **process all elements of the input simultaneously**, as opposed to RNNs, which move sequentially through the data. This parallelism makes transformers remarkably efficient, akin to reading an entire book's pages at once, instantly comprehending connections between them. This parallel processing is what has given the recent outbreaks in natural language processing, since processing large corpus of data is now possible and training data can increase up to "the whole internet" level.

Challenges

Still, we should not think of transformers as the ultimate models, since they come with their own limitations. While transformers offer immense power, they require **substantial computational resources**, which may limit their accessibility for smaller-scale projects. That is primarily the reason

why transformers haven't been the default model on most biomedical applications. Moreover, they share the same **limitations on explainability, uncertainty and fairness** with all the previous models which prevent them from being used on crucial applications.

8.4.5 Generative neural networks

Generative models are yet another class of models. Their primary goal is to understand the underlying process that generates the data and mimic it. All the aforementioned models (MLP, CNNs, RNNs and transformers) can be used within the generative networks. The output of the network here is not a prediction, but rather a new datapoint that is generated from the network itself.

Generative models are used when **ChatGPT** generates an answer for your question, or when **Dall-e** creates an image out of your description.

We will briefly mention important categories of these models without diving deeper into the details:

- **Encoder-decoder models**: These are models with two parts where the encoder compresses the input into a small description, and that description is then used by the decoder to generate the same datapoint again. This way the model learns to describe the data points, and hence any new description could use the decoder to generate a new image.
- **Generative adversarial networks (GANs)**: This model again uses two parts: the generator and the discriminator. Their role is not cooperative as in the first group but rather competitive. The generator creates new fake datapoints, and the discriminator needs to find out whether they are true or not. Training both of them together gives a model that generates realistic images that they can trick the model of discriminator.
- **Diffusion models**: In this category the logic of denoiser is applied to learn the underlying data distribution. The model usually is of the form of encoder-decoder while this time the input is noisy data (e.g., image) and the output an effort to remove the noise. Each datapoint is enhanced gradually with some noise, and the model tries to denoise it step by step. Through multiple objectives of denoising, these models achieve so far the best results into understanding the logic behind the data and hence generating new.

Importance of generative models in biomedical research

Generative models represent a compelling avenue for exploration in biomedical research. While we've provided a glimpse into some existing approaches, we believe that they will be one of the key points in AI for the biomedical field. These models can address data scarcity by generating new datapoints, tackle privacy concerns by removing the biometric identity of patients data, and get a grip on the data completion challenge by imputing the missing values. Generative models have already proved being useful by proposing new therapies, or by unravelling protein structures for new drug discovery (Alphafold). Their ability to create, simulate, and enhance data offers exciting possibilities for researchers and practitioners in the field.

8.5 Transfer learning

8.5.1 Transfer learning

In this new subsection, we will discover the concept of **transfer learning**.

Transfer learning is a machine learning technique where knowledge gained from training one model on a specific task is applied to a related but different task. It involves reusing pre-trained models or

their learned representations to accelerate training or improve performance on new tasks with limited data.

Why?

Deep ANNs are great at performing complex tasks, outperforming classical machine learning approaches. However, a big drawback is that they need huge amounts of data to train on. So how can we exploit ANNs when we do not have unlimited data, which is often the case in a medical context? Transfer learning is a popular way to **decrease the reliance on large amounts of data**. Transfer learning is essentially reusing a pre-trained model on a new problem.

Transfer learning has several benefits, but the main advantages are **saving training time, better performance** of neural networks (in most cases), and not needing a lot of data.

What?

In transfer learning, the kno**wledge of an already trained machine learning model is applied to a different but related problem**. For example, if you trained a simple classifier to segment the hypothalamus in a brain image, you could use the knowledge that the model gained during its training to segment a tumour in similar brain images.

With transfer learning, we basically try to exploit what has been learned in one task to improve generalization in another. What a neural network has learned is captured in the weights that determine how each neuron processes inputs to compute outputs. So those weights are transferred, meaning that the network uses the weights learned in "task A" as a starting point when it learns a new "task B".

The general idea is to use the knowledge a model has learned from a task with a lot of available labelled training data in a new task that doesn't have much data. Instead of starting the learning process from scratch, we start with patterns learned from solving a related task.

How?

Remember how ANNs can be composed of multiple layers? In transfer learning, we can choose how much we want to adapt the pre-trained neural network in the new task. If "dataset B" is very similar to "dataset A", we may not want to change the ANN too much. We can then decide to **"fix" some layers** of the pre-trained ANN, meaning we do not update the weights of those neurons anymore when training for the new task. Whether we want to **fix some weights or fine-tune all the weights**, depends on the relative size of dataset A and B and on the similarity between the tasks. When some weights are kept fixed, these are usually the ones in the earliest layers. These layers learn the most basic and rudimentary features, which are often useful across different datasets.

It is also possible to **remove** part of the pre-trained network and replace it by some other layers which may be better suited for the new task. For example, the size of the output layer in a classification problem depends on how many classes there are. If we transfer knowledge between two tasks with different amount of output classes, then we can remove the output layer and replace it by a new, randomly initialized head.

8.5.2 Transfer learning in medical applications

Medical datasets are often quite small for deep learning purposes. The core problem is that acquiring and annotating medical data is expensive, because of expensive equipment needed to acquire data and experts needed to annotate the data. Even pre-training an ANN to detect objects in non-medical image datasets, can help to perform better at a totally different task on medical images.

Another use case for transfer learning is learning how to detect certain health markers in data captured by wearable devices. It is typically hard to obtain large labelled datasets with a wearable device, due to the difficulty of annotating these data for clinicians. An approach could be to train an ANN on a dataset with similar data (e.g., epilepsy detection on standard EEG) and then start from the pre-trained model when learning to perform a task on the wearable data (e.g., epilepsy detection on wearable behind-the-ear EEG).

The difference between "dataset A" and "dataset B" may take many forms. There could be a **difference between the acquired data**, like with wearable and hospital-based acquisitions or with CT scanners from two different companies. The discrepancy may be even larger, so transfer learning may be performed between different modalities as well, e.g. between CT scans and MRI images. The difference between two datasets may also come from the fact that the task differs (like annotating sleep or annotating epileptic seizures in EEG data).

8.6 Model-centric vs data-centric AI

8.6.1 Improve the AI's performance

In this new subsection, we will explore the differences between **model-centric vs data-centric AI**. Let's start with the following reflection exercise:

Reflection exercise - Should we work on the data or on the AI models? Which is more important?

Imagine you are a medical researcher studying a specific disease, and you have collected data from various patients, including their medical history, genetic information, and test results. Your goal is to use AI to predict disease progression. Now, here is a question:

Where would you focus your efforts? Why?

- Improving the quality of the data you've collected.
- Customizing a highly complex AI model for analysis.

In this scenario, you might argue that focusing on the **quality of the data** is the most important aspect. Good quality data will ensure that the AI model's predictions are meaningful and accurate. Even the most sophisticated AI model cannot make accurate predictions if the input data is noisy or incomplete.

A general rule when we deal with AI, is that AI models can only be as good as the data is. The more our data is good at clearly representing a problem, the more an AI model will be effective in solving the problem. But on the other hand, if we get a **simple AI model**, it will be likely able to solve only **simple problems**.

So, should we focus our work on acquiring the best data as possible or on customizing our AI model as best as we can to improve its performances?

It is not easy to answer this question. For instance, we should consider that in several real-world applications, it is not possible to change or improve the data acquisition mechanism, e.g. when data has been collected through standardized or very expensive systems or machinery. Furthermore, AI systems must often be installed on low-resources devices, such as wearable sensors, which are not capable to load complex AI models.

When we focus on systematically engineering our **data** to clearly represent the target problem, and hence to build better AI systems, we are exploiting a paradigm that we call **data-centric AI**.

When we focus instead on producing the best possible **AI model** for a given dataset, e.g., considering different AI architectures, leveraging different training techniques, or performing hyperparameters optimization, we are instead exploiting a **model-centric AI** paradigm.

Let's dive into data- and model-centric AI on the next pages.

8.6.2 Data-centric AI

Imagine you are working on an AI system for detecting **diabetic retinopathy** in medical images, a crucial task in ophthalmology. You have a dataset of retinal images. The images vary in resolution, brightness, and quality. Some contain only a portion of the retina, while others show the entire retina.

In a data-centric AI paradigm, we perform several steps to **elaborate the dataset** to produce a new version of itself, in which it is easier to solve the target problem. For this reason, we employ **simple AI models** to solve this task, such as shallow classifiers or lazy learners.

By comparing the gain in accuracy that we obtained using the same AI model on the old and the new version of the dataset, we get an idea on how good our approach is in processing the dataset and extracting the target information from it.

Techniques in data-centric AI

- **Data preprocessing**: Cleaning, filtering, and transforming raw data to make it more suitable for machine learning.
- **Feature engineering**: Creating relevant features that enhance the AI's understanding of the problem.
- **Data augmentation**: Generating additional training data by applying various transformations to the existing dataset.
- **Data quality assurance**: Ensuring data accuracy, consistency, and reliability.

Since we are **extracting specific information from the data**, and we are using only that information to solve the target problem (thanks to simple AI models), our system usually has better generalization capabilities. Indeed, our model relies only on the information extracted to solve the target problem, and since this information is typically covered by domain knowledge, our model can better generalize in a **real-world** scenario. For the same reasons, data-centric AI models are less prone to overfitting.

Let’s do a trivial example on this: imagine building an AI system to classify eye-images. You want to distinguish which one contains blue, green, or brown eyes. One approach is to insert all images as input to the AI model, and let it build the connection between the input image and the classification outcome autonomously. A data-centric AI paradigm instead should extract the eye-colour information from the image, thanks to image preprocessing and features extraction, and then use a simple classifier to associate each colour extracted to a target class.

Since we know that the target problem only depends on this feature that we can extract from the images, our model will have a **high generalization capability**. In the previous case instead, we do not know how the model makes its prediction, because it is considering the overall image, and hence we do not know if it will rely on specific features that this dataset shares, which is not replicable in real-world.

Drawbacks

- **Data availability**: Obtaining high-quality and diverse data can be challenging, limiting the applicability of data-centric AI.

- **Expertise required**: Data-centric AI requires expertise in data preprocessing, feature engineering, and domain knowledge, which is inaccessible in several applications (e.g., healthcare)
- **Data biases**: If we are not careful, data-centric approaches can perpetuate biases present in the data, leading to biased AI models.

In conclusion, data-centric AI is a **powerful** paradigm to build robust AI systems, which can be easily deployed also among low-resource devices, thanks to the usage of simple and light AI models. On the other hand, data-centric AI relies on the assumption that we know which **specific features** we need to extract from the data to solve the target problem, and that we can implement a feature engineering algorithm to extract such features. What can we do in these cases?

8.6.3 Model-centric AI

Imagine you are working on an AI system for **detecting lung cancer from X-ray images**, which is a critical task in healthcare. You have already deployed a dense neural network (DNN) to classify the X-ray images as either showing signs of lung cancer or not.

In a **model-centric AI** paradigm, given a fixed dataset, we must identify the best model possible to solve the target problem. This procedure includes identifying the **best model architecture** and selecting the **best parameters** of this architecture. This results in a long trial and error procedure in which all the performances of the identified models are compared to select the best one.

Can we avoid this procedure? No. Several algorithms have been proposed to optimize the choice of the best model and parameters, but there is not a strong mathematical foundation when we perform optimization with categorical parameters (such as in model-centric AI). Even if we use such algorithms only with the non-categorical parameters, the global convergence to the best model (which includes categorical and non-categorical parameters) is not assured.

Techniques in model-centric AI

- **Model selection**: Choosing the appropriate machine learning architecture and algorithms for a specific problem.
- **Hyperparameter tuning**: Optimizing model hyperparameters to fine-tune model performance.
- **Ensembling**: Combining multiple models to improve predictive accuracy.
- **Transfer learning**: Adapting pre-trained models to new tasks by updating their weights.

Model-centric AI is a very popular and effective paradigm in recent AI literature. Why? Because it makes the problem of improving the performances of the system almost transparent (i.e. independent) with respect to the specific data that the model is taking as input. This way, AI-engineers can develop their models on their own, which reduces the need to interact with domain-experts to validate their decisions. In data-centric AI instead, engineers must be closely involved in the data acquisition process, to be aware of what specific data they are using, and they must be in touch with domain experts to understand which specific features to extract from that data.

Also, in healthcare we sometimes do not know which specific features the AI should leverage to solve the target problem. Hence, it is **difficult to deploy an AI system using a data-centric paradigm**. For example, in epileptic seizure detection from wearable-EEG, medical doctors typically perform the annotations using videos of the subjects, and are not able to perform the same task from the wearable sensor. Which features should we extract in this case?

8.7 Understand the domination

8.7.1 Why are neural networks dominating the literature and AI architectures?

In this final subsection of the module, we will try to **understand the domination**. Bringing together all the considerations that we did about model-centric AI paradigm, this lets us understand why neu**ral networks (NNs)** approaches are dominating other AI architectures in the literature nowadays.

Neural networks are hierarchical systems composed by **layers of neurons**. Each layer processes the data, and puts it in input for the next layer until we have the final classification layer in which the network makes its decision. In such a way, we can see each layer processing as an automatic feature engineering step. If we have a NN with n-layers, we could say that during the firsts (n-1)-layers the network is extracting some features from the input data, while during the last layer the network is classifying those extracted features. Considering this analogy with data-centric paradigms, we can understand that **NNs learn autonomously which features to extract from the input data**, and how to solve the target problem thanks to those extracted features.

Other machine learning approaches, such as shallow classifiers or lazy learners are not able to perform such data processing. Therefore, they always need some additional step which involves features engineering to leverage the input data. On the other hand, NNs are plug-and-play, meaning that they just need a good amount of training data, which will make them ready to be trained and deployed.

Since NNs do not rely on features engineering, they can be smartly applied to all the scenarios where we do **not know which specific information should be extracted** from the data to solve the target problem.

Drawbacks

- **Overfitting risk**: Complex models can overfit to the training data, resulting in poor generalization to new data.
- **Computational intensity**: Developing, training, and optimizing complex models can be computationally expensive and may not be suitable for low-resource environments.
- **Dependency on data quality**: Model-centric approaches heavily depend on the quality of the training data. If the data is inadequate, model-centric AI may yield suboptimal results.

One of the primary challenges in model-centric AI is ensuring that the models developed are interpretable and explainable, especially in the medical domain where decisions can have critical consequences. In model-centric AI, we do not want to rely on specific features extracted from the data, so the models developed with this paradigm are opaque and difficult to understand. We call them **black boxes**, and we will talk about them in a few chapters.

8.7.2 Explainability

As you already saw, AI systems represent an interesting opportunity for healthcare. According to a recent study by MedTech Europe, integrating AI in healthcare could save around 200 billion euros per year and save approximately 400,000 lives annually. Other than some technological and infrastructural problems related to this task (cybersecurity, distributed healthcare records etc…) one of the most important problems is the so-called **trust problem**, which is significantly slowing the integration of AI in healthcare.

Indeed, new advanced AI architectures have been developed recently and are available to be used in multiple healthcare tasks. Despite their promising performances, their deployment for actual clinical use is challenging from clinicians' perspectives due to the lack of trust in these systems. One of the

major aspects of the trust problem is that the results provided by AI systems appear to be **too good to be true**, and their decision-making mechanism is usually **opaque and not easy to understand**.

Turning back to when we discussed model-centric AI, you may remember that the clinicians' scepticism towards AI has a strong theoretical and technological foundation. Indeed, when exploiting a model-centric AI paradigm, AI models are designed and trained to maximize their recognition performances. By doing this, the models learn autonomously which features to extract from the data, while the **underlying process is completely hidden**, even for the engineers who made it. For this reason, we call those AI models **black boxes**.

Explainable AI (XAI), or Interpretable AI, refers to an AI system which provides insights (i.e. explanations) about the decision making mechanism exploited by an AI system. Those insights should make the overall AI system comprehensible and interpretable from a human point of view.

Addressing the perception of the black box of AI by introducing explainable AI, would allow these systems to overcome the problem of trust in medical scenarios.

Why do we need explainability?

- **Patient perspective**: The consequences of medical AI errors can be life-altering, making it crucial to ensure that AI systems provide transparent, understandable, and reliable insights.
- **Regulatory compliance**: Healthcare is heavily regulated, with legal and ethical obligations regarding patient data privacy, consent, and fairness. Explainability is essential to comply with these regulations and ensure ethical AI practices in healthcare.
- **Medical decision support**: Healthcare professionals need to make informed decisions based on AI recommendations. Explainability enables doctors to understand the reasoning behind AI-generated predictions, which facilitates collaboration between AI and human expertise.
- **Bias and fairness**: Explainable AI can reveal and mitigate biases that are present in the data and model. This ensures that AI systems are fair and do not discriminate against different patient groups.
- **Education and research**: Explainable AI provides a means for medical professionals and researchers to gain insight into how AI systems arrive at specific conclusions. This knowledge can be invaluable for advancing medical knowledge and improving AI models.

In model centric AI paradigms, AI models, such as neural networks, are trained to solve tasks by analysing raw data, which they can process inside their complex structure. This can lead to biases in those models. **Biases** are rules that our model learns by leveraging strange correlations which are present in our dataset, but the rules are not replicable in the real world.

Handling biases can be challenging. Typically, we address them while performing data acquisition, by controlling the experimental environment. This is not always trivial, and biases often represent conditions that we were not aware of while acquiring the data. Hence, we were not able to avoid them.

Let's get into a real example depicting bias.

The goal is to distinguish patients with or without pneumonia from CT images coming from 3 different hospitals: hospital A, B and C. The engineers first trained their convolutional neural network (CNN) model using data from hospital A and B, and then they tested it in two different setups: with data coming from hospital A and B, and with data coming from hospital A, B and C. This way, they are testing the generalization capabilities of their CNN. What they get, is an accuracy drop from 73% (A, B) to 24% (A, B, C).

In this scenario the dataset consisting of the hospital C images represents our “real world” simulation case. As we already stated, biases are rules that our model learn from the data which help to solve the problem in the training phase, but the rules are not replicable in the real world (hospital C).

So why do we get this huge drop in accuracy when including hospital C? Because our model is biased.

How can we understand what the bias consists of? Which is the wrong rule that our model learnt from the data? To answer this question, we can employ Explainable AI.

When using CNN, we can exploit XAI algorithms to understand the activation map of the input images. In such activation map we highlight the regions of the image which are more relevant for the AI model during its decision-making. By doing this, we noticed that our CNN is pushing all the attention not on the part of the image related to the lung, but to the one containing the metal token used to acquire the CT image.

Why? To answer this question, we need to consider how we collected the dataset of hospitals A and B.

First, since the data has been acquired from two different hospitals, the CT scanners of these hospitals are different.

Then dataset A consisted of 42.396 images, while dataset B consisted of 112.120 images, so more than double the amount of dataset A. But the proportion of pneumonia positive samples in dataset A is 34.2% (positive rate), while in dataset B this is only 1.2%.

In such a situation, being able to recognize the hospitals rather than the pneumonia, means being able to correctly predict (i.e. output no pneumonia) the 71.7% of the results. This is close to the 73% accuracy obtained in their result.

Hence, it is reasonable that our CNN is biased in this setup, i.e., the easiest solution for the AI was to decide based on the scanner type (and hence identify the hospitals A and B), rather than on the presence of lung abnormalities. On the other hand, when we try to test the CNN on a novel dataset (C) that condition is not replicable and our accuracy falls.

9 Getting practical with AI. Tips & tricks

9.1 Welcome to Module 9

Welcome to Module 9, a pivotal stage in our course where we shift gears from theoretical foundations to the practical applicability of AI. In this module, you will be provided with the essential tips & tricks needed to navigate the practical challenges of AI implementation in healthcare

Key Focus Areas

- Interdisciplinary expertise
- Explainable AI tools
- Generalizability & amount of data
- Simple vs complex methods
- Model-driven vs data-driven AI
- Independence of training & testing

Why This Module Matters

This module bridges the gap between theory and application, equipping you with the tools and insights necessary to thrive in the real-world implementation of AI in the medical field. As the digital landscape continues to evolve, the ability to navigate the interdisciplinary and practical aspects of AI in healthcare becomes increasingly valuable.

Learning goals

- Recognize the difference between correlative and causative machine learning models in healthcare.
- Learn the best approach for handling data in clinical machine learning applications including common challenges like noise, missing data and class imbalance.
- Understand how dynamic medical practice and discontinuous timelines impact clinical machine learning application development.
- Discuss trade-off between model performance and interpretability.
- Understand how to improve upon an existing model, taking into account data quantity.
- Understand "confidence" in the model.

9.2 Tip 1: Interdisciplinary expertise

9.2.1 Aisha develops a predictive model for diabetes

As Aisha was finalizing the modules of this course you're following, she got excited to get practical. She decided to start developing a predictive model, eager to make a significant impact on diabetes management, which would benefit her diabetic grandmother Vivian. However, in the midst of coding, the engineering enthusiasm overshadowed the need for comprehensive medical input. The scientific question at hand was **predicting blood sugar fluctuations** in diabetic patients, a critical aspect of managing the disease.

Aisha started building her predictive model solely based on a public diabetes dataset that she found on the internet. As the project neared completion, excitement turned to disappointment. The algorithm, although technically sound, failed to deliver accurate predictions for her grandmother.

When Zarah returned home, she saw that Aisha was disappointed. After an initial explanation of the issue, Zarah started reviewing the results. She immediately identified the flaw: the model did not account for the diverse range of diabetic patient profiles, neglecting crucial nuances in their treatment plans and dietary restrictions. Grandma Vivian recently changed her diet to vegetarian, and this was not included at all in the model.

It became evident that the oversimplified approach had severe consequences. Patients with specific dietary restrictions or unique medication responses were not considered, leading to unreliable predictions that could potentially harm rather than aid in diabetes management.

The failure served as a lesson. Aisha realised the **importance of collaboration** and the indi**spensable role of doctors** and medical insights in shaping the algorithm.

9.2.2 Tip 1: Interdisciplinary expertise

AI in healthcare applications can only become successful if it is the result of interdisciplinary expertise, so you need both clinical knowledge on the topic, and practical (technical) AI development experience.

This first tip might seem trivial and straightforward to you, but it actually reflates to an underrated aspect of research into AI in healthcare.

AI is no magic, and will only give you an added value when you understand the problem and the problem context. It may be tempting to dive directly into some computational tools, but before starting this, you must clearly define your **scientific question**. Be clear on your aim, do a literature search on both the medical topic, and the algorithmic tools that might support you, and already identify any potential pitfalls.

At the same time, if a tech-savvy person thinks they can design a new method on a public medical dataset, they fail to acknowledge that there is a **medical context** behind it and that the public dataset might be full of bias.

In this sense, it is also useful to reflect on the difference between **correlation** and **causation**:

- Machine learning methods learn associations between inputs and outputs based on whatever signal it can extract from the data. The fact that there is a relationship, does not mean there is a causal relationship. There might be confounding factors, which cause a model to fit the data based on useless correlations. If you want to estimate strong causal effects, you will need to use causal machine learning approaches (as will be discussed in the use case in Module 14).
- The biggest challenge in general machine learning might even be, when a model exploits unexpected confounders that have no relevance to the task. It is clear that this can severely impair or invalidate the model's ability to generalize to new datasets (and generalization is always the most important aspect, see tip 2 in the next subsection).

9.2.3 Examples – Need for interdisciplinary expertise

Example 1: Chest X-rays

A simple example is the diagnosis of pneumonia based on chest X-rays, which we discussed in Module 1. An AI model with high accuracy exploited **non-medical cues** to estimate disease.

Example 2: Death risk estimation

Another example relates to estimating death risk in pneumonia cases: A high performing model could use patient information to identify their risk of dying from pneumonia. Upon detailed investigation, it

was found that the model was heavily relying on a correlation between **asthma and good patient prognosis** in the data.

- In this case the model was not wrong in identifying the correlation between asthma and good patient outcomes.
- However, upon inspection doctors realized that the correlation between asthma and good patient prognosis was the direct result of a hospital policy to admit and aggressively treat asthmatic patients with pneumonia.
- The mistake would be to assume the model's prediction meant that having asthma causes a good outcome for pneumonia patients.

So these examples are just to illustrate how important it is to **consider the whole medical context**, as AI can otherwise consider spurious facts and lead to useless models in a clinical context.

Sometimes medically irrelevant correlations can still be useful - the best way for this is to reconfigure the model's application context, and framing the context of the model applications is something you should spend a lot more time on.

In summary, supervised machine learning models will look at the data to solve the problem, and as such, they might find a model that is not ideal from the clinical point of view, by unravelling less relevant correlations.

Always keep in mind, that models lack **contextual knowledge and general common sense**. That is why you need **close collaboration between domain experts** to help develop, evaluate and deploy models designed by data geeks.

9.3 Tip 2: Explainable AI tools

To minimize the risk of deploying wrong models, probe your model with all the tools you have. A good tool for the developer is to use explainable AI tools for checking to what extend the model works and what correlations the model has learned.

Models often tend to be like a "**black box**". The black box metaphor refers to a system for which we can only observe the inputs and outputs, but not the internal workings. However, we would love to understand how the model comes to a prediction. In this sense, there are "interpretable" or "explainable" approaches in AI that try to shed light into the model.

Explainable AI is a whole field by itself. There are two distinct "flavours" of machine learning model explainability: intrinsic and post-hoc interpretability.

- **Intrinsic interpretability** is simply referring to models, often simple models, that are self-explanatory from the start.
- **Post-hoc interpretability** is used to understand decisions by complex models that do not have prescriptive declarative knowledge representations or features. When using features in the model, SHAP values are often shown. SHAP values measure how much each feature (such as income, age, credit score, etc.) contributes to the model's prediction. SHAP values can help you see which features are most important for the model and how they affect the outcome. For deep neural networks, saliency maps are often used as post-hoc explanations, which highlight which part of the input matters most to the model in making its prediction.

In general, there is a trade-off between more complex models that might obtain a better performance, and models with fewer features that are easier to visualize and understand. Choosing between performance and interpretability is not easy, and often the choice comes down to trust.

9.4 Tip 3: Generalizability & amount of data

Generalizability

The most important aspect of AI is that the model works in real life; that the model works on data not previously seen before. Therefore, the setting of your training dataset should resemble the setting where you will aim to deploy it.

Can you think of factors that might bias your model? For example, if you train your model on patients older than 40, because all your patients are older than 40, you might get worried about model performance on data from patients below 40. But typically, you do not notice such problems, because **you cannot know all sources of bias upfront**. And you cannot know how representative your training data is. Post-hoc, once you deploy the method, you can start measuring such discrepancies between training data and real world data. And eventually, you might consider to **retrain models** to reduce such sources of bias. But always be wary of hidden data subgroups not reflected in your training / test set.

Be very wary of **data leakage from testing to training dataset** during experimentation and evaluation. Splitting your data in training and testing datasets is the first thing you do. You cannot explore your testing data, or derive a normalization strategy based on all your data. You cannot check the result of preprocessing your data on the test set. And as soon as you evaluate your model once on the testing dataset, you cannot compare it to a different model evaluated on that test set. This happens very often in practice, but if you compare model A to model B (and model C, D, …) on your testing dataset, and you decide that model A outperforms model B, the only thing you can conclude is that this is the best model for that dataset. If you want a fair assessment, you need to freeze the whole analysis pipeline and apply the final model to an unseen dataset. Performance obtained in that way can be a decent estimate.

Amount of data

How much data do you need to train the model? Evaluate the influence of the amount of data, and if you obtain similar model performance with less data, you can conclude that you have enough data.

Typically, as you increase the size of your dataset, performance grows accordingly and eventually reaches a **plateau**. This plateau can vary depending on the complexity of the algorithm and the problem. But sometimes your dataset size will be limited by the number of patients you have in your database, or to how expensive or cumbersome it is to extract those data.

And **the more explanatory variables you aim to include, the more data samples you will need**. There are no theoretical guarantees to how much data you need, and very few good rules of thumb. You might have heard of the "1 in 10" rule that suggests the need for at least 10 examples of each label class, which is probably relevant for simple methods; or that you need at least 10 examples for each feature you include as explanatory variable.

Though more data is often good, naively adding more data may not be helpful, and in some cases it could **decrease performance**; especially in a dynamic field such as healthcare, as historical data might contain various changes in parameter values, clinical habits, patient populations etc. For example, Stanford research that used retrospective EMR datasets of varying size, found that a small dataset covering about 2000 patients and one month of the most recent data, was more effective in the final

performance of a machine learning prediction model than a much larger dataset composed of data collected over a 1 year period.

So, it is best to evaluate the influence of the amount of data, and if you obtain similar model performance with less data, you can conclude that you have enough data.

9.5 Tip 4: Simple vs complex methods

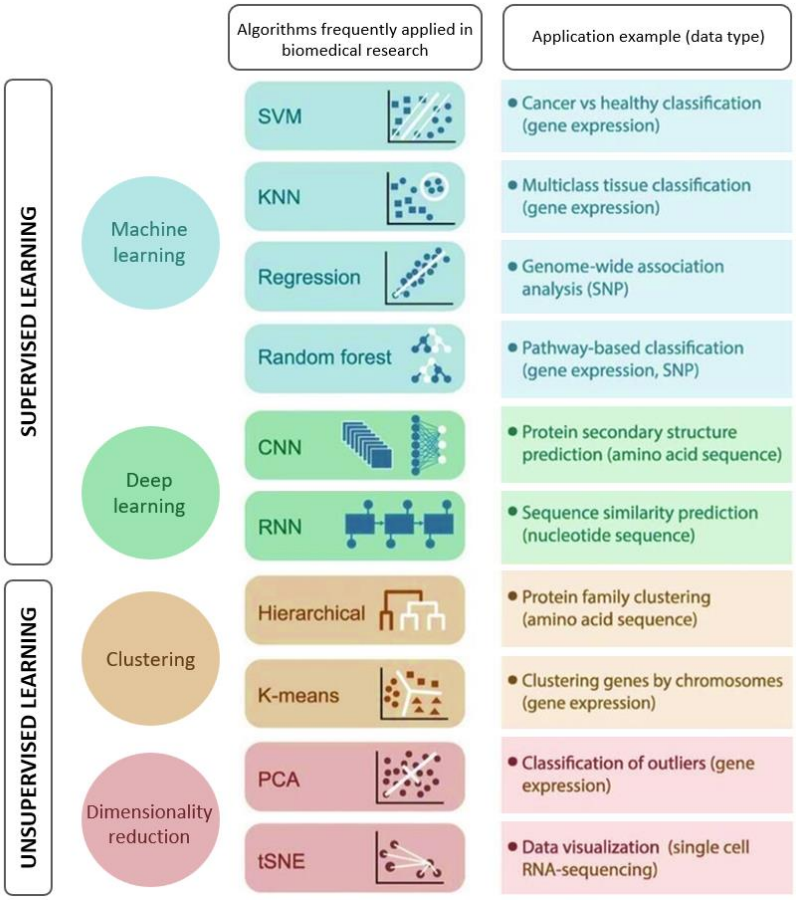
Start with the simplest methods, and use complex methods only if it is necessary.

Especially with the spread of machine learning and the buzz around deep learning, one might be tempted to use more complex methods just because they are more trendy or in fashion. Instead, we recommend to start any analysis by utilizing simple algorithms, chosen among the traditional ones. They are as good as complex ones after all.

If the **traditional algorithms** (with less parameters) are sufficient to solve your problem, just stick with them. Only if performance with the less complex technique is unsatisfactory, move to more complex methods, such as deep learning. Using simple methods will give you the chance to keep everything under control and evaluate results generated in an interpretable way, which might be helpful when other healthcare professionals will use your results.

Making a **more complex model** or tuning more hyperparameters to increase performance, can only improve the model in limited ways and can also run the risk of overfitting. So unless the performance of the model is very close to the goal, the best next step is probably still **acquiring more data**.

If you are still confused between supervised and unsupervised learning, or which method to use, you can leverage this guide designed to help you understand the differences and applications of each:



Abbreviations: SVM, support vector machines; KNN, K-nearest neighbours; CNN, convolutional neural networks; RNN, recurrent neural networks; PCA, principal component analysis; t-SNE, t-distributed stochastic neighbour embedding.

9.6 Tip 5: Model-driven vs data-driven AI

"Garbage in, garbage out!" - bad data will result in bad models. We want high quality data. Be sceptical of your data, your model, and any metric numbers. Dig into the data and the labels. Only with good data and good labels, your model can become useful.

No matter how sophisticated the machine learning algorithm or the data engineering techniques are, **bad data will result in bad models**. The choice of data and problem to solve is infinitely more important than the algorithm or the model (see Module 8 where we discussed model-centric vs data-centric AI).

The assessment of, and methodology to create a high-quality dataset are not standardized. As such, sometimes learning how to **improve the data** to have better model performance is a relevant exercise.

It is also important to be clear how reliable your **ground truth labelling** is.

- Labels like mortality have a relatively straightforward relation to readily available determinations of ground truth.
- With other labels, like pneumonia, it can be much more difficult to codify ground truth, as that truth may only be expressed in clinical and medical imaging data, which can be hard to mechanistically interpret, and can be fraught with inaccuracies as well as confounding information.
- Is there even a ground truth between experts? Are you aware of the inter-rater disagreement for your problem at hand?
- Look out for labels (like with COPD patients) that rely on numerical cut-offs that change over time in medical practice, or those that vary by age in terms of the upper and lower bounds. For these labels, the consideration of data "shelf life" and treatment changes is very important.

We can expect that our labels will not be 100% accurate compared to a ground truth, thus we need to find ways to estimate and understand our **label noise**. Label noise is inevitable and even very noisy data can train a very good model. You can better deal with label noise when you have large datasets.

So in conclusion, be sceptical of your data, your model, and any metric numbers no matter if they are bad and especially if they are good. Dig into the data and the labels. Only with good data and good labels, your model can become useful.

- Really try and explain why the model fails in certain circumstances.
- Work toward acquiring external datasets if it makes sense for the prediction-output pairing, provided you can line up the labels and data types, because it is an incredibly useful way to get a true sense of your model performance.
- Ensure that the test set is as free from noise as possible, is independent, and is truly providing all the information needed to make decisions on performance.

9.7 Tip 6: Independence of training & testing

Testing and training must be completely independent. If the model indirectly learns some information of the test data during training, then testing won't give an independent performance measure anymore.

It is easy to artificially boost accuracies by violating the independence assumption. Be VERY cautious to **avoid information leakage**, i.e., accidental use of information from the test set during training.

Examples of information leakage

Normalizing all the data, then splitting into training and test sets

When the whole dataset is normalized, including test and training data, the normalized training data will contain some information of the test data. This means that the independence of training data and test data is violated. Instead, **normalize training and test data separately after splitting**.

Using data from the same patient in training and testing

Data from the same patient is very likely to be correlated, hence, this is a violation of the independence of training data and test data.

Training a model, evaluation on test data, iterating

In order to obtain better results, we might adopt a training strategy where we periodically evaluate the model on the test data, and then continue training. At the end of training, we can then choose the model that performed best on the test data as our final model. This results in choosing a model biased towards the test data, and therefore, the test performance is not an independent performance measure.

Note: what we can do to adopt a strategy like this without information leakage, is to separate another portion of the dataset as the **"validation set"**, which can be used during training to choose the best model. The test set is then set aside and not used during training, while the validation set is used to periodically test the model and choose the best one.

Some other common mistakes:

Preprocessing methods that use statistics of the whole data set, or normalization of features

This fits into the category of information leakage. Any parameters of preprocessing must be estimated solely using the statistics of the training data, and not of the test data. For example, it is inappropriate to first perform independent component analysis (ICA) on the entire data, then select the best components and extract features from these components as input to a classifier, and finally evaluate the classifiers' performance using cross-validation. Although the bias induced by unsupervised preprocessing techniques is usually rather small, it can result in an improper model selection and overoptimistic results. A more severe mistake would be to use a preprocessing method which uses class label information, and perform this method on the whole data set before training.

Loss function not appropriate

Different problems require different loss functions. For example, not all performance metrics are appropriate when classes are unbalanced (i.e. when the amount of samples per class is not approximately equal). If 85% of the dataset belongs to class 1 and 15% to class 2, an uninformed model that classifies everything as class 1 achieves an 85% accuracy. Therefore, accuracy is not a suitable performance measure in this case. A better option would be a normalized error that assigns higher weights to underrepresented classes.

Outliers are rejected from the whole dataset (resulting in a simplified test set)

Removing outliers from a training dataset is always allowed, however, rejecting them from the test dataset is a bit more problematic. Indeed, by rejecting outliers, the test set is being simplified, and this

improves the testing accuracy. A method which obtains a similar accuracy but without rejecting outliers, actually performs better.

Features are selected on the whole dataset, including trials that are used later in the test set

This mistake is again an example of information leakage. **Feature selection is a part of training** an algorithm, and should therefore be performed solely based on the training set.

Selecting hyperparameters by cross-validation on the whole data set and reporting the performance for the selected values

Many machine learning methods have hyperparameters that need to be optimized, e.g. the number of neighbours in k-nearest neighbours (KNN). Often, such a hyperparameter is optimized by performing cross-validation on the whole dataset, repeating this for different choices of the hyperparameter, and then choosing the hyperparameter with the best performance from those cross-validations. In this case, the performance obtained with the best hyperparameter choice is a biased performance metric, as it has been directly minimized by the model selection: we optimized the hyperparameter over this cross-validation performance. In this case, an extra independent test set is needed to get a realistic performance measure.

Non-stationarity of the data disregarded

This is a problem related to the distribution of training and test data. Training and test data are assumed to be identically distributed. However, when we deal with biomedical signals, data are not always stationary over time. E.g., the characteristics of brain signals and, in particular, the feature distributions often change slowly with time. Therefore, a model fitted to data from the beginning of an experiment may not generalize well on data towards the end of the same experiment. When we use the end of the signals as test data, and the beginning as training data, the test error might be very large. When we use cross-validation, with training and test samples from the whole length of the signal, the non-stationarity of the signal is accounted for in the model. In this case, we might overlook the non-stationarity altogether.

In order to check for non-stationarities, it is therefore advisable to check how the performance with a chronological training/test split compares to the cross-validation performance. If the latter is much better than the former, this may be caused by non-stationarity of the data.

This concludes Module 9. We hope that you could appreciate the important tips & tricks needed to navigate the practical challenges of AI implementation in healthcare.

10 How to coexist with AI

10.1 Welcome to Module 10

Welcome to Module 10, where we embark on an exploration of coexisting with AI in an ethical, safe, and sustainable manner. This module delves into the critical aspects of ethics, trustworthiness, safety, and societal implications that underpin AI integration in healthcare. We will uncover the principles and practices necessary to ensure the responsible development and deployment of AI technologies in the medical sector.

Key Focus Areas

- **Ethics and trustworthy AI**: We will dive into the foundational principles of ethics in AI, exploring the importance of AI systems being designed and utilized in ways that are fair, transparent, and respectful of human values and rights. Trustworthiness is key to gaining public acceptance.
- **Utilitarianism**: We will consider the ethical theory of utilitarianism, which seeks to maximize overall societal happiness and well-being, and how it can guide AI development and decision-making.
- **Technical robustness and safety**: Learn about the technical aspects of AI safety, including methods for ensuring robustness, fault tolerance, and the minimization of unintended harm.
- **Privacy and data governance**: Explore the critical issues surrounding data privacy and governance, including the ethical handling of personal data and the establishment of safeguards against misuse.
- **Transparency, fairness, and bias**: We will examine the principles of transparency in AI, the challenges of fairness and bias in algorithms, and the techniques for addressing these issues to ensure equitable outcomes.
- **Sustainability and societal impacts**: Understand the environmental impact of AI and the broader societal implications, including the potential for job displacement and the need to create sustainable AI solutions.
- **Accountability and human oversight**: Delve into the mechanisms for ensuring accountability in AI systems, including the role of human oversight and the establishment of responsible decision-making processes.

Why This Module Matters

As AI continues to evolve and integrate into the healthcare sector, addressing the ethical, safety, and societal impact challenges is of paramount importance. Ensuring AI technologies are developed and used in a manner that respects human rights, safeguards privacy, and mitigates potential harms is crucial for fostering trust and acceptance. Moreover, achieving sustainability in AI development and understanding its broader societal impacts is essential for shaping a future where humans and AI coexist harmoniously, especially in the context of healthcare.

Learning goals

- Give a general definition of the terms, "sustainability", "ethics and legislation", and illustrate these definitions with examples (in the broad sense, not necessarily related to AI).
- Describe responsible, sustainable and trustworthy AI, and explain why these are relevant for the success of AI in the future.
- Explain the main components of trustworthy AI.

- Explain the four ethical principles that reflect the foundations of trustworthy AI (respect for human autonomy, prevention of harm, fairness, explicability), and address the tensions between them.
- Learn how to propose and explain requirements that support the implementation of trustworthy AI throughout the AI system's life cycle (human agency and oversight, technical robustness and safety, privacy and data governance, transparency, diversity, non-discrimination and fairness, societal and environmental wellbeing, accountability).

10.2 Ethics and trustworthy AI

10.2.1 Grandma's dilemma: privacy and accountability in AI in healthcare

In this first subsection of the module, we will delve into **ethics and trustworthy AI**. Let's start with the following introduction story...

The matriarch of the family, Vivian, is a kind-hearted and wise woman. She has seen the world change in many ways during her lifetime, and now she found herself pondering the implications of AI in healthcare. One evening, as the family gathered around the dinner table, she voiced her concerns.

"You know," she began, "I've been thinking about this AI that knows so much about our health. It's quite impressive, but I can't help but wonder about our privacy. What happens to all the data we're sharing with it?"

The family exchanged glances, realizing they hadn't thought much about this aspect of their AI companion.

Aisha took the floor trying to "defend" AI. "Well, grandma, the AI uses our data to make accurate health predictions and recommendations. But as far as I know, it's designed to keep our information secure. It's all anonymized, and the engineers who implement all the software must re-assure strict data protection."

Grandma Vivian nodded thoughtfully. "That's reassuring, but what if it makes a mistake? Who will be responsible if it prescribes the wrong treatment or gives us incorrect advice? Can we trust it blindly?"

Her questions hung in the air, and the room fell silent for a moment. The family realized that they had never discussed the issue of accountability in AI in healthcare before.

Dr Zarah, chimed in, "You bring up a valid point. While AI can provide valuable insights, it's not infallible. There must be mechanisms in place to hold someone accountable if things go wrong. We should do some research to understand the legal and ethical framework surrounding AI in healthcare."

As they continued their dinner conversation, they made a pact to learn more about the ethics of AI in healthcare. Vivians concerns had sparked a crucial discussion within the family, setting them on a path to better understand the complexities of data privacy and accountability in the world of AI.

Little did they know that this would be just the beginning of their journey into exploring the multifaceted world of ethics in AI, as they sought to ensure that their family's health and well-being remained in safe and responsible hands. After following this module of the course, Aisha will be able to enlighten them with more information about this hot topic.

10.2.2 Evaluating personal values

Values are deeply personal and often shaped by our upbringing, culture, and life experiences. These **values guide our ethical decisions and behaviour**. However, even among humans, defining and

prioritizing these values can be a complex and subjective task. This exercise was designed to highlight the intricacies of this process and to shed light on the challenges of applying it to AI. After completing the exercise, you may find that your selected values reflect your personal beliefs and priorities. However, you might also notice that different individuals have their own unique sets of values, and the importance they assign to these values can vary widely.

This exercise illustrates the **challenge of defining ethics and personal values for AI**. When programming AI systems to make ethical decisions, developers must grapple with the vast diversity of human values and the subjectivity inherent in this process. The **difficulty in establishing a universal set of ethical guidelines for AI** becomes apparent, as there is no single "right" answer that applies to everyone. As we move forward in the development of AI, it becomes crucial to recognize the limitations and complexities of this task and to foster ongoing discussions and ethical considerations to create AI systems that align with our collective values while respecting the individuality of human perspectives.

10.2.3 Trustworthy AI

Trustworthy AI is a fundamental concept defined and regulated by the **European Union (EU)** as part of their framework for the development and deployment of AI. In the context of healthcare, ensuring AI systems are trustworthy is of paramount importance as they can impact patient outcomes, safety, and healthcare decision-making.

Trustworthy AI in healthcare encompasses several key elements, each of which plays a crucial role in assuring the reliability and ethical use of AI technologies. Dive deeper into each key element below:

Human agency and oversight

Trustworthy AI in healthcare places humans in control and ensures that AI systems are designed to augment human decision-making, rather than replace it. It requires transparency in AI processes to allow healthcare professionals and patients to understand how decisions are made.

Example: An AI-based diagnostic tool used in healthcare provides explanations for its recommendations, empowering physicians to make informed decisions about patient care. It does not make autonomous medical decisions but assists medical professionals by providing valuable insights.

Technical robustness and safety

This element emphasizes the reliability and safety of AI systems in healthcare. AI applications must be thoroughly tested and designed to minimize risks and vulnerabilities.

Example: An AI-driven surgical robot in healthcare is equipped with multiple layers of safety features, such as redundancy in critical components and real-time monitoring by skilled surgeons to prevent unintended harm to patients.

Privacy and data governance

Trustworthy AI in healthcare ensures that patient data is handled with the utmost care, adhering to data protection regulations like GDPR. It also promotes the use of anonymized and encrypted data.

Example: A telehealth AI platform used for remote consultations in healthcare employs end-to-end encryption to protect sensitive patient information, and data is stored and processed in compliance with privacy laws.

Transparency

Transparency is vital for trust in healthcare AI. Users should have access to information about AI system capabilities and limitations, as well as understanding the rationale behind generated recommendations.

Example: A medication recommendation system in healthcare shall display the sources of data used and the algorithms applied to provide information to patients, so they can make informed choices about their treatments.

Diversity, non-discrimination, and fairness

AI systems should be designed and trained to avoid bias and discrimination in healthcare, ensuring equitable outcomes across different patient groups.

Example: An AI-based system for triaging patients in emergency departments must be regularly audited for fairness, and adjustments are made to the algorithm to avoid bias in patient prioritization based on factors such as race, gender, or age.

Societal and environmental well-being

Trustworthy AI should consider the broader impact on society and the environment. In healthcare, this means ensuring AI applications contribute positively to patient outcomes and healthcare system efficiency.

Example: An AI-based system for optimizing hospital resource allocation in healthcare helps reduce patient waiting times, lowering stress and improving the overall patient experience.

Accountability

Accountability is key to ensure that those responsible for AI systems in healthcare can be identified and held liable for their actions.

Example: The development team behind a clinical decision support AI system in healthcare maintains a transparent record of their involvement in the system's development, allowing for accountability in case of errors or adverse events.

These key elements of trustworthy AI, as defined by EU regulations, serve as a foundation for promoting ethical, reliable, and responsible AI development and deployment in healthcare and other domains. Adhering to these principles helps ensure that AI systems enhance healthcare outcomes, protect patients, and uphold the highest ethical standards.

10.2.4 Recommendations on the ethics of AI

The **"Recommendation on the Ethics of Artificial Intelligence" by UNESCO**, adopted in November 2021, provides a comprehensive framework for the ethical development and deployment of artificial intelligence. It emphasizes the importance of aligning AI with human values and fostering AI that benefits humanity.

These recommendations can be effectively applied in healthcare to ensure that AI technologies in the medical field prioritize patient well-being, ethical considerations, and societal values. They are built upon four core values and ten core principles, laying out a human-rights centred approach to the ethics of AI.

Discover the core values and principles with examples by revealing them below:

Four core values

1. Human dignity

In healthcare, upholding human dignity means respecting the intrinsic worth of each patient. AI technologies must be designed to enhance, not diminish, the dignity of patients.

Example: An AI-driven end-of-life care system in healthcare must allow patients to express their end-of-life preferences and ensure those wishes are honoured with the utmost respect.

2. Non-discrimination

Non-discrimination is a cornerstone of healthcare ethics. AI in healthcare must be developed to avoid bias and discrimination, ensuring equal access and quality of care for all patients.

Example: An AI-powered health monitoring system must be designed to provide equal and accurate healthcare advice to patients regardless of their gender, race, or socio-economic status.

3. Autonomy

In healthcare, AI should empower patients and healthcare professionals to make informed decisions. It should provide information, not dictate outcomes, preserving individual autonomy.

Example: A medical chatbot in healthcare offers personalized health information and treatment options, respecting the patient's autonomy by allowing them to choose their preferred course of action.

4. Transparency and accountability

Transparency ensures patients and healthcare professionals understand AI processes and decision-making. Accountability holds developers and healthcare providers responsible for AI systems' actions and consequences.

Example: A medical AI platform must maintain a transparent record of all treatment recommendations, ensuring that healthcare providers can be held accountable for any unforeseen issues.

Ten core principles

1. Human rights and human dignity

In healthcare, AI must respect the fundamental human right to healthcare and the dignity of the individual.

Example: A telemedicine platform ensures patients receive equitable access to medical consultations, regardless of their geographical location, respecting their human rights and dignity.

2. Non-discrimination

AI in healthcare should be free from bias, ensuring that every patient, regardless of background, receives fair and equal treatment.

Example: An AI-based system for organ transplantation matching ensures that allocation decisions are made without bias, respecting the principle of non-discrimination.

3. Autonomy and individual decision-making

AI should provide patients with the information and tools to make informed decisions about their healthcare.

Example: A fertility treatment AI tool offers patients detailed information about treatment options, enabling them to make autonomous decisions regarding their reproductive health.

4. Beneficence and non-maleficence

AI in healthcare should aim to do good and prevent harm. It should enhance patient care and safety.

Example: An AI-driven medication reminder system ensures patients take their medications correctly, promoting beneficence and non-maleficence by preventing medication errors.

5. Justice

In healthcare, AI should promote fairness and equitable access to healthcare resources, including medical interventions and treatments.

Example: An AI-assisted healthcare triage system ensures that patients with the most urgent needs receive prompt medical attention, reflecting the principle of justice.

6. Privacy and data protection

Healthcare AI must handle patient data with the utmost care and respect privacy regulations.

Example: A mental health AI chatbot ensures that patient conversations are kept confidential, adhering to data protection principles.

7. Transparency

Transparency in healthcare AI means providing patients and healthcare providers with insights into the AI's decision-making process.

Example: A cancer diagnostic AI system offers explanations for its recommendations, giving healthcare professionals and patients insight into its decision, adhering to the transparency principle.

8. Accountability

Developers and healthcare providers must be accountable for the ethical use and consequences of AI in healthcare.

Example: The development team behind an AI-powered surgery assistant ensures that their actions and decisions are accountable through proper documentation and oversight.

9. Access to information and knowledge

AI should provide equitable access to healthcare information and knowledge, empowering patients and healthcare professionals.

Example: An AI-powered medical library provides healthcare professionals in underserved areas with the same access to the latest medical research as their urban counterparts, reflecting the principle of equitable access.

10. Education and capacity building

Education and training in the use of AI in healthcare are essential to ensure that all healthcare professionals can leverage these tools effectively and ethically.

Example: A healthcare AI platform offers continuous training to healthcare professionals, ensuring they are well-equipped to use AI to enhance patient care and safety.

UNESCO's four core values and ten core principles provide a robust **human-rights centred approach to the ethics of AI in healthcare**. By following these values and principles, healthcare stakeholders can ensure that AI technologies prioritize human rights, ethical considerations, and the well-being of patients, while upholding the highest standards of medical ethics.

10.3 Utilitarianism

10.3.1 Baby's juice dilemma: A lesson in utilitarianism

In this new subsection of the module, we will discuss **utilitarianism**. Firstly, let's begin with the following introduction story...

In the bustling "AI" family household, ethical discussions were not limited to the realm of AI in healthcare; they extended into everyday life, initiated by the concerns of grandma Vivian. One morning, a scenario unfolded that provided the perfect opportunity to introduce the concept of utilitarianism to the family.

Little Jack toddled into the kitchen, filled with wonder. "Juice! Juice!" he gleefully exclaimed. Zarah, always eager to keep her little one happy, smiled and fetched a glass of freshly squeezed orange juice for him. Jack took his first sip, his tiny face lighting up with delight. "More, more!" he giggled, extending his glass for a refill. Zarah hesitated, but eventually obliged, filling his glass once again.

Jack continued to drink, his happiness evident in every sip. But as the glass emptied for the second time, something changed. He started to squirm uncomfortably and clutch his tummy. "Owie!" he whimpered, tears welling up in his eyes.

Zarah quickly realized that perhaps Jack had overindulged. The family gathered around as Jack's discomfort grew.

Eric, who had a penchant for philosophy, decided it was time to turn this into a learning moment. "Jack," he began, "do you remember when you first had that juice? You were so happy, and it tasted delicious, didn't it?"

Jack nodded, his lower lip still quivering. Eric continued, "But now, after having so much juice, you're in pain, and it's not making you happy anymore, is it?"

Jack shook his head, his tears beginning to dry. Eric smiled gently and turned to the rest of the family. "This is a simple lesson in utilitarianism. It's the idea that we should aim to maximize overall happiness and minimize suffering. When Jack first had the juice, it brought him happiness, so it was a good thing. But when he kept drinking even after his belly started hurting, it turned into a bad thing because it caused suffering."

Grandma Vivian nodded, impressed by the way Eric had made a complex philosophical concept so relatable. "So, it's like saying that an action is right if it brings more happiness than suffering?"

Eric nodded. "Exactly. And in Jack's case, it teaches us to be mindful of our actions and their consequences, just like we need to be careful about how much juice we give him."

The family shared a moment of understanding, realizing that ethical principles like utilitarianism could be applied not only in philosophy but also in their everyday lives. But does it also apply for AI? In this subsection, let's learn!

10.3.2 Marginal utility

Marginal utility is a fundamental concept in **economics** that refers to the additional satisfaction or benefit (utility) a person derives from consuming one more unit of a good or service. It helps economists and individuals understand how people make decisions about consuming goods and services. In the context of **AI applications**, marginal utility can be thought of in terms of the additional value or benefit gained from using AI for a particular purpose.

Diminishing marginal utility

Diminishing marginal utility is an economic concept that describes the idea that the additional satisfaction or benefit (utility) gained from consuming an additional unit of a good or service decreases as you consume more of it. In other words, **the more you have of something, the less additional satisfaction you get** from each additional unit. This concept is often associated with the law of diminishing marginal utility, which suggests that as people consume more of a product or service, the extra benefit they derive from each additional unit decreases. Similar to the economic concept, in AI, there may be diminishing marginal utility as you apply AI to solve a problem. For example, the first AI-driven automation of a manual task within a business process may provide substantial efficiency gains and value. However, **as more AI automation is implemented, the additional value gained from each new AI application may diminish.**

In the business context, decision-makers need to consider marginal utility when allocating resources to AI projects. They should invest in AI applications that offer the highest additional value relative to the cost and effort required. The concept of diminishing marginal utility in AI underscores the importance of a balanced AI strategy that takes into account both the initial benefits of AI adoption and the long-term return on investment.

Example in healthcare

Consider a patient who is experiencing pain. If they receive a single dose of pain medication, the relief they experience from that first dose might be quite significant, resulting in a high level of utility. However, if they continue to take more and more doses of the same medication, the additional relief they experience from each subsequent dose is likely to decrease. This is a real-world example of diminishing marginal utility, as the utility of each additional dose of medication diminishes as the patient takes more.

The Monster of Utility

Robert Nozick was a philosopher who introduced the concept of the "**Monster of Utility**" as a thought experiment to challenge the idea that maximizing utility (happiness, well-being) should be the sole goal of ethical and political decision-making. The "Monster of Utility" is a hypothetical scenario in which a **machine or system can directly stimulate people's brains to make them feel immensely happy**, eliminating all other concerns, needs, or desires in their lives. Nozick argues that even if this machine could provide the maximum possible utility, people may not choose to use it because they value other aspects of their lives, such as autonomy, choice, and experiences that go beyond simple pleasure.

Example in healthcare

In the context of healthcare, the "Monster of Utility" concept challenges the idea of prioritizing only the maximization of patient well-being without considering other values. For instance, if a healthcare system focused solely on maximizing patients' physical well-being, it might advocate for the use of invasive, uncomfortable, or risky medical procedures without taking into account the patients' preferences for quality of life, comfort, and autonomy. Nozick's thought experiment reminds us that

people have diverse values and desires, and their well-being is not solely determined by the maximization of happiness but also by individual autonomy and the ability to make choices about their own healthcare.

AI applications should not solely focus on maximizing utility. Instead, they should consider the broader values and choices that individuals or societies prioritize. For example, an AI used in healthcare should respect patients' autonomy and preferences even when maximizing health outcomes. Nozick's "Monster of Utility" thought experiment reminds us that people have diverse values and desires. AI should respect individual autonomy and the ability to make choices about their use. For instance, in recommendation algorithms, AI should provide diverse options rather than pushing users into a narrow utility-maximizing path. The concept highlights the importance of ethical considerations in AI design. AI applications should be developed with ethical principles in mind, respecting individuals' rights, values, and autonomy. In AI applications, there is often a need to **balance utility with values**. For instance, a content recommendation algorithm should not prioritize utility (maximizing user engagement) at the expense of values like diversity, privacy, and user choice.

In summary, **diminishing marginal utility** is an economic concept describing the decreasing additional satisfaction from consuming more of a good or service. The "**Monster of Utility**" thought experiment by Robert Nozick challenges the idea of prioritizing happiness as the sole goal of ethical and political decisions, emphasizing the importance of individual autonomy and choice, even in healthcare. Marginal utility and the "Monster of Utility" concept can be applied to AI applications. While marginal utility helps in evaluating the additional benefits of AI, the "Monster of Utility" reminds us that AI should respect values and choices beyond utility, particularly in areas where ethical considerations and individual preferences are significant, such as healthcare, content recommendation, or autonomous decision-making systems.

10.3.3 How good is good? How can we quantify utility in the context of AI?

Defining "good" in the context of AI and quantifying the utility of an AI application can be complex, as it depends on the specific goals, tasks, and ethical considerations associated with that AI system. Here are some key considerations for understanding **what is "good" and quantifying utility in AI**:

One common measure of a good AI application is its **performance and accuracy** in achieving its intended task. For example, in a medical diagnostic AI system, "good" may mean high accuracy in diagnosing diseases. The **efficiency** of an AI system can also be a measure of its goodness. A "good" AI application may process tasks quickly and with minimal resource consumption. But apart from the metrics of the performance of the AI models there are other concepts that must be quantified to assess how “good” an AI application is.

User satisfaction, ethical considerations, value alignment and adaptability are some of the concepts that must be considered too. The subjective experience of users can define what is "good." In AI interfaces or chatbots, a "good" application should be user-friendly, providing an excellent user experience. Furthermore, a "good" AI application should adhere to ethical principles and not cause harm or violate human rights. For example, a good AI system in law enforcement should respect privacy and avoid bias. In some cases, what is considered "good" depends also on how well the AI aligns with human values and intentions. An AI-driven recommendation system should suggest items or content that align with the user's preferences. Finally, a good AI system must be adaptive and capable of learning from data to improve its performance over time. An AI application that continuously refines its recommendations based on user feedback can be considered "good."

Quantifying the utility of an AI application involves measuring its **value, effectiveness, and impact**. The specific metrics used can vary depending on the application, as mentioned also in Module 4. **Accuracy** metrics can be one of the main means of quantifying utility, but utility can also be quantified through **cost-benefit analysis**. This involves comparing the costs of implementing and maintaining the AI system with the benefits it provides in terms of increased revenue, reduced costs, or improved efficiency. Collecting **feedback from users** can help measure the utility of an AI application. Surveys, user ratings, and Net Promoter Scores (NPS) can provide insights into user satisfaction and perceived value. In healthcare, utility can also be quantified by assessing different **health outcomes**. For instance, in a medical AI system, utility may be measured by reduced mortality rates, shorter hospital stays, or improved patient quality of life.

Utility can also involve **ethical assessments**, ensuring that the AI application respects principles like privacy, fairness, and transparency. You can use specific ethical frameworks to evaluate utility in this context. For commercial AI applications, utility may be measured by **market share, revenue generated, or return on investment (ROI)**. As we will explain later in this module, **environmental sustainability** must also be considered. For example, utility may be quantified by the reduction in carbon emissions or other environmental benefits achieved by AI applications.

Quantifying utility is not always straightforward, as it often involves trade-offs between various factors. The choice of metrics and methods should align with the specific goals and values associated with the AI application in question. It is important to consider both quantitative and qualitative measures to fully assess utility in AI.

10.4 Technical robustness and safety

10.4.1 Grandma’s glucose levels: Importance of technical robustness and safety

In this new subsection of the module, we will discuss **technical robustness and safety**. Let's start with the following story...

In the "AI" family's journey with AI in healthcare, there came a pivotal moment that underscored the significance of technical robustness and safety. It was a day they would never forget.

One morning, an alarming incident occurred. Grandma Vivian, who recently started relying on an AI healthcare companion for managing her diabetes, received a startling recommendation. The AI system advised her to take an unusually high dose of insulin, far beyond her prescribed limit. Alarmed by the message, Vivian hesitated and decided to consult Zarah and Eric. With their experience in healthcare, they both advised her against it, and they quickly reached out to the AI healthcare provider to report the issue.

A couple of days later, Zarah informed the family that an investigation was launched, and it was discovered that a software glitch had caused the AI to misinterpret Vivian's glucose levels, leading to the erroneous recommendation.

The family were shaken by this incident and Vivan seemed to be losing her trust in AI. It was a stark reminder that even the most advanced AI systems were not infallible. They realized that without the strict adherence to technical robustness and safety protocols, such errors could have catastrophic consequences.

Aisha said, "This incident highlights why technical robustness and safety are paramount. AI can bring tremendous benefits, but we must remain vigilant and ensure these systems are thoroughly tested and monitored to prevent any harm."

Zarah, who had seen the importance of precision in healthcare firsthand, nodded in agreement. "Indeed. Just like in medicine, where precision is crucial, AI systems must operate flawlessly to avoid putting patients at risk."

The family shared their story with others in their community, advocating for the rigorous development and testing of AI healthcare systems to prevent similar incidents from occurring in the future.

As grandma Vivian continued her diabetes management with a renewed sense of caution, the family knew that these principles are not just buzzwords. These are the bedrock of trust and reliability in the world of AI. Vivian summarized their discussion. "Technical robustness and safety are like the foundation of trust in AI healthcare. Just as we trust the beams and pillars of our house to keep it standing, we must trust the reliability and safety of our AI to keep us healthy."

10.4.2 Accuracy metrics and overfitting

Accuracy metrics and overfitting are crucial considerations for the technical robustness of AI algorithms (in healthcare) due to the critical nature of (medical) decisions and the need for reliable, trustworthy AI solutions.

Accuracy metrics

Accurate performance metrics are vital, because they directly impact various (patient) outcomes and (medical) decision-making. In healthcare, even small errors can have significant consequences. Therefore, algorithms must be designed to maximize their ability to provide correct results. By using appropriate accuracy metrics, AI developers can **gauge the reliability** of their models and **fine-tune** them to meet high standards. For example, in a diagnostic AI system, high accuracy ensures that patients receive the correct diagnosis and treatment recommendations, minimizing misdiagnoses and reducing the risk of harmful consequences. (For more information related to accuracy metrics you can refer to Module 4.)

Accuracy metrics are pivotal for **trustworthy AI** in healthcare. They serve as a cornerstone for ensuring that AI algorithms deliver reliable and high-quality results. In healthcare, where the consequences of inaccuracies can be life-altering, or even life-threatening, these metrics provide a measure of how well the AI system performs. High accuracy is not only about achieving correct results but also building trust among healthcare professionals and patients. The transparency in performance, as indicated by accuracy metrics, enhances the understanding and acceptance of AI tools, reinforcing their trustworthiness.

Overfitting

Overfitting is a common challenge in AI algorithm development, especially in healthcare (as we discussed in Module 4). Overfitting occurs when an algorithm is excessively tuned to the training data, capturing noise rather than the underlying patterns.

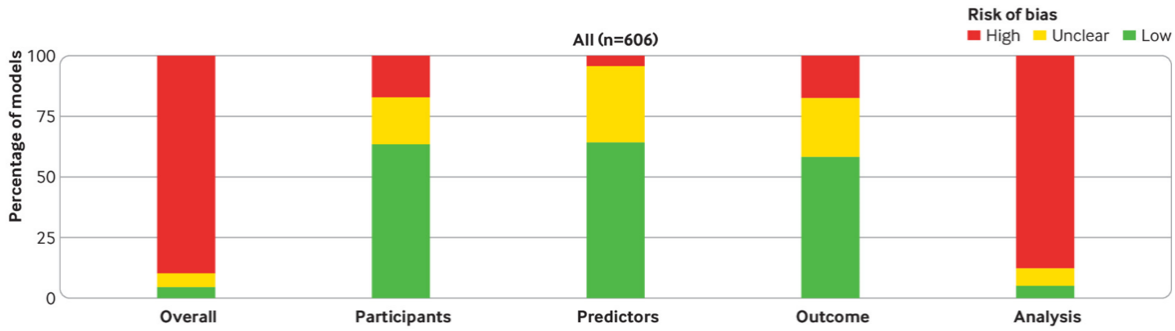
In healthcare, overfitting can be dangerous, as it can lead to incorrect diagnoses or treatment recommendations. Addressing overfitting is essential for ensuring the generalizability of AI models to real-world patient cases. Developers must use appropriate techniques like regularization, cross-validation, and robust training data to mitigate overfitting risks. Addressing overfitting is another indispensable aspect of trustworthy AI in healthcare. Overfitting not only undermines the generalizability and reliability of AI models, but also introduces the risk of making erroneous and potentially harmful recommendations. Trustworthy AI requires a commitment to the robustness of algorithms, which means they should not merely excel in training on specific data, but also demonstrate the ability to make accurate and unbiased predictions in real-world clinical scenarios. **Mitigating**

overfitting ensures that AI models perform consistently, even when faced with diverse patient cases and data, thereby upholding the trustworthiness of AI systems used in healthcare.

In healthcare, the **consequences of inaccuracy and overfitting can be severe**, including misdiagnoses, incorrect treatments, and potential harm to patients. Therefore, focusing on accuracy metrics and guarding against overfitting is a priority to enhance the technical robustness of AI algorithms in the healthcare domain. By doing so, AI systems can better serve as valuable tools to assist healthcare professionals in delivering accurate, reliable, and safe care to patients.

Let us deep dive in a specific use-case. **Prediction models for COVID-19** proliferated rapidly in the academic realm, aiming to aid medical decision-making during the pandemic. However, a significant concern emerged regarding the trustworthiness and reliability of these models. The majority of published prediction model studies suffered from inad**equate reporting and a high susceptibility to bias**, which likely inflated the reported predictive performance. Models created with a low risk of bias need thorough validation before they can be considered for clinical implementation. Failure to do so carries the risk of unreliable predictions, which could lead to more harm than benefit in guiding critical clinical decisions.

One of the primary challenges is the insufficient sample size in many studies, as seen in the figure below. This issue was observed in 67% of cases. Inadequate sample sizes, combined with a limited number of events, increased the risk of overfitting, especially when complex modelling techniques were applied. Failure to adequately account for overfitting or optimism was also prevalent. As can be seen in the figure below, the most of the bias in prediction models arise from the analysis phase (overfeeding). We will come back to other types of bias, such as selection of participants, later in this module.



Notably, as mentioned in the review study by Wynants et al., a significant portion of models (16%) received neither internal nor external validation. Even when internal validation was conducted, it was sometimes improperly executed, potentially inflating the models' performance statistics.

Moreover, evaluation of discrimination and calibration, crucial for reliable predictions, was often incomplete or conducted using inappropriate statistical methods. Calibration, which assesses how closely predicted outcomes align with actual outcomes, was only examined in a minority of cases. Many models inadequately handled missing data, with a substantial portion resorting to complete case analysis or failing to specify their approach to missing data.

In summary, trustworthiness and reliability in AI-driven prediction models for COVID-19 were compromised due to issues such as insufficient sample sizes, overfitting, incomplete validation, and suboptimal handling of missing data. This raised even concerns about the potential harm associated with relying on these models for crucial clinical decisions during the pandemic. Adherence to

methodological guidelines, rigorous validation processes and measurements against overfeeding are crucial to ensure the credibility of AI-based predictions in healthcare.

10.4.3 Reproducibility

Reproducibility in AI refers to the capacity to replicate and verify AI outcomes and methods. It necessitates the open sharing of code, data, and methodologies to enable others to independently recreate AI models and their results. Reproducibility in AI, particularly in the healthcare domain, is critical for enhancing trustworthiness and ensuring that AI systems are reliable and credible. It can be divided into three key aspects: **technical, statistical, and conceptual reproducibility**.

Technical reproducibility

Technical reproducibility focuses on the ability to replicate the implementation and execution of AI models, algorithms, and experiments. It involves sharing the code, software, and computational environment used to develop AI systems, as well as the data used to train the model. Technical reproducibility ensures that other researchers and practitioners can recreate the same AI model and experiments. This allows for independent validation, verification, and improvement of AI systems. Without technical reproducibility, it becomes challenging to trust AI models and their results, as the implementation details may be hidden, making it impossible to identify and rectify errors or biases.

Statistical reproducibility

Statistical reproducibility in the context of AI pertains to the ability to maintain the validity of a result when subjected to resampled conditions that may produce slightly different numerical outcomes but should not significantly alter the overall claimed result. For instance, in AI, this means that if an algorithm is trained multiple times on the same data but with different initializations or random subsamples, the reported results should exhibit statistical equivalence even if they are not precisely identical. This concept shares similarities with "internal validity," which is commonly applied in social science research. It essentially examines the extent to which the AI model's performance remains consistent under variations, ensuring that it doesn't rely on specific initial conditions or data subsamples. Statistical reproducibility is about ensuring the robustness of results despite minor numerical variations.

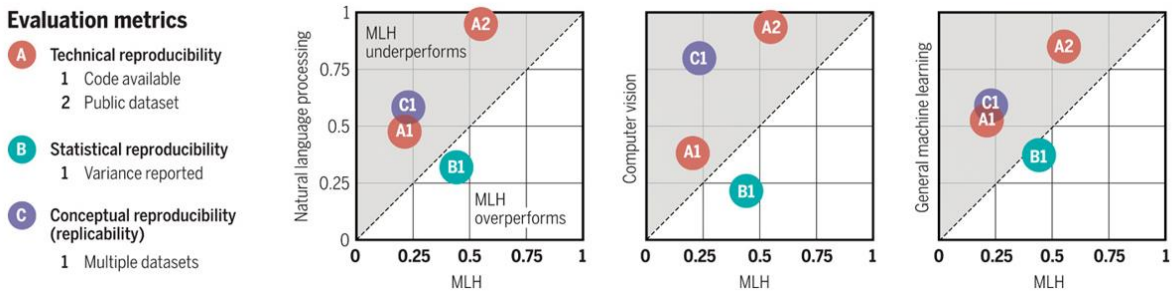
Conceptual reproducibility

Conceptual reproducibility, also known as "replicability," addresses how effectively desired results can be reproduced under conditions that align with the high-level, conceptual description of the intended effect or outcome. This means that beyond statistical equivalence, the core concept or principle behind the AI model's functionality remains consistent across different conditions. In essence, conceptual reproducibility emphasizes the capacity to achieve the same high-level outcome or effect even in settings that mirror the overarching concept, irrespective of specific numerical variations. This concept bears a resemblance to "external validity," as it examines how well an AI model's principles and effects can be generalized and applied to real-world situations. It underscores the broader reliability of AI systems, ensuring that they work as intended in various contexts, beyond numerical precision.

In the healthcare domain, where the consequences of AI errors can be life-altering, reproducibility is a fundamental principle for building trust in AI systems and ensuring their safe and effective use in clinical practice. As it has been reported by several studies, **AI in healthcare is haunted by reproducibility issues**, even compared to other fields where AI is being used.

In the figure below, four different reproducibility metrics (A, B, and C) are depicted for evaluating scientific papers with machine learning (ML) applications in four distinct domains: machine learning in health (MLH), natural language processing, computer vision, and general machine learning. Presented

is the fraction of papers in a given subspecialty (y axis) versus those in MLH (x axis) that release their code (A1), release their data (A2), report their variance (B1), and leverage multiple datasets (C1). As can be noted, MLH lags other subfields of machine learning on all measures of reproducibility apart from inclusion of proper statistical variance. The main issue of reproducibility for MLH is the open access to datasets, which indirectly affects also metric C. The lack of available public datasets is a well-known issue and is related to privacy restrictions due to the nature of health data, which will be analysed in the following subsection.



10.5 Privacy and data governance

10.5.1 Data privacy in the age of AI

Introducing this new subsection on **privacy and data governance**, let's begin with the following story...

Eric, a dedicated pharmacist, had become well aware of the power of AI in improving patient care and medication management. However, as he watched the healthcare landscape evolve, he couldn't help but harbour concerns about data privacy. One evening, after a long day at the pharmacy, Eric sat down at the living room with a furrowed brow. The family gathered around, sensing that something was weighing heavily on his mind.

"Dad, what's bothering you?" asked Aisha. Eric took a deep breath and began, "You see, I've been thinking a lot about the AI program the pharmacy is planning to install. It's designed to help us manage medication orders more efficiently and prevent harmful drug interactions. But there's something that's been bothering me—data privacy."

The family exchanged curious glances, sensing the gravity of Eric's concerns. Grandma Vivian nodded understandingly. "You're worried about the information the pharmaceutical company might have access to, aren't you?"

Eric nodded, relieved that someone understood. "Exactly. If we use this AI program, it will have access to data about our clients' medications and health conditions. What if the pharmaceutical company or anyone else misuses or mishandles that sensitive information? Our clients trust us with their healthcare needs, and we have a responsibility to protect their privacy."

Zarah added, "Data privacy is not just about following regulations; it's about respecting the trust our clients place in us. We must ensure that their personal information remains confidential and secure."

Aisha chimed in, "So, dad, what can we do to protect your clients' data?". Eric smiled "Well, we can start by carefully reviewing the data privacy policies of the AI program and the pharmaceutical company. We need to ensure that they have robust security measures in place to safeguard patient information. And if we find any red flags, we should voice our concerns."

As the family discussed the issue further, they realized that data privacy was a critical aspect of implementing AI in healthcare. It wasn't just a matter of convenience and efficiency; it was about preserving the sanctity of the doctor-patient relationship and the pharmacist-client trust. They also decided to advocate for transparency and strong data protection measures within their professional community. Eric understood that by doing so, they could contribute to building a healthcare system where AI was used responsibly and ethically, and where patients' data privacy was never compromised.

10.5.2 Data privacy

In healthcare, where patient privacy and data integrity are paramount, **privacy and data governance are critical for building and maintaining trustworthy AI systems**. Adherence to these principles not only safeguards patient information, but also ensures that AI-driven healthcare solutions are reliable and ethical, ultimately benefiting both healthcare providers and patients.

Privacy in the field of healthcare comprises many different aspects:

- **Data minimization**: Only collect and use data that is necessary for the AI's intended purpose, reducing the risk of unauthorized access or misuse. In healthcare AI, a hospital's AI-driven diagnostic tool should only access and use the patient's medical history and relevant test results necessary for the diagnosis. Unrelated personal information, such as social security numbers, should not be collected.
- **Informed consent**: Before using a patient's data for AI-based research, healthcare institutions must obtain explicit consent from the individual, explaining how their data will be used.
- **Data security**: Healthcare AI systems should employ robust encryption and access controls to protect sensitive patient data from unauthorized access or cyberattacks.
- **Transparency**: Patients should be informed about the algorithms used in their medical treatment and understand how their data contributes to diagnostic or treatment recommendations.
- **Data retention policies**: Hospitals should establish policies for retaining patient data for a specified period after treatment, and data should be securely deleted when no longer needed.
- **Data portability and access**: Patients should have the ability to access their medical records and share them with other healthcare providers as needed.

10.5.3 Data governance in healthcare

Data governance is the framework and set of practices that ensure high data quality, manage data effectively, and maintain data security and compliance within an organization. It involves establishing policies, procedures, and roles for data management, including data collection, storage, access, and usage.

Data governance aims to maximize the value of data assets, reduce data-related risks, and maintain data integrity while aligning with the organization's strategic objectives and regulatory requirements. It covers various aspects, such as data ownership, data stewardship, data quality, data privacy, and data compliance, and it often involves collaboration between different departments to ensure that data is an asset and used responsibly in the organization's operations and decision-making processes.

- **Data quality**: Healthcare AI systems must ensure the accuracy of patient records, as even minor errors can have a significant impact on diagnosis and treatment decisions.
- **Data compliance**: Healthcare institutions must comply with regulatory standards like HIPAA (Health Insurance Portability and Accountability Act) in the United States, which sets strict guidelines for patient data privacy and security.

- **Ethical data use**: When developing AI algorithms for healthcare, it is essential to avoid bias, especially in decision-making processes that could lead to unfair treatment of patients from different demographic groups.
- **Data catalogue and inventory**: Hospitals should maintain an organized inventory of patient data sources, including electronic health records, medical imaging data, and research datasets.
- **Data stewardship**: A Chief Data Officer or responsible teams need to be assigned within healthcare organizations to oversee data governance, ensuring that data is handled responsibly.
- **Data auditing and accountability**: Regular audits of data usage and storage should be conducted, with clear accountability for breaches or violations.

10.5.4 Legal framework

You can access information on the current **data governance, data privacy laws and jurisdictions per country** at https://www.dataguidance.com. On this website, you can access a comprehensive, continuously updated interactive world map with the privacy laws per country.

In 2022, legislative bodies across 127 countries enacted a total of **37 laws that specifically incorporated the term "artificial intelligence" into their legal frameworks**. The United States took the lead by passing nine of these laws, with Spain closely following with five, and the Philippines with four. These legislative measures covered a range of issues and areas of concern related to AI. For instance, in the Philippines, one law addressed the need for education reform to adequately address the challenges brought about by emerging technologies, with a particular emphasis on AI. In Spain, a notable bill aimed at ensuring non-discrimination and accountability in the use of AI algorithms, with the intent to create a fair and transparent AI landscape. Additionally, in the United States, an act was enacted to establish an AI training program, which operates under the U.S. Office of Management and Budget. This program focuses on equipping individuals with the necessary skills and knowledge in the field of artificial intelligence.

Notably, these recent legislative actions are part of a broader trend. Since 2016, countries have collectively passed a total of 123 bills specifically related to artificial intelligence. This surge in AI-related legislation underscores the **increasing recognition of the importance of AI governance and regulation** as AI technologies become more integrated into various aspects of society. It reflects the global effort to establish legal frameworks that ensure responsible and ethical AI use and promote innovation while safeguarding against potential risks and challenges.

10.5.5 Privacy-preserving approaches

We are currently in an era of unprecedented data availability. As datasets expand in size and richness, machine learning models thrive. With each iterative refinement of these models, the accuracy of insights improves, outcomes become more reliable, and predictive capabilities grow increasingly precise.

However, the healthcare sector faces unique challenges in embracing the current data-driven revolution. **The lack of freely distributable health records has long been a barrier to innovation in healthcare**. Privacy restrictions, security laws, and stringent regulations, along with organizational guidelines safeguarding protected health information, limit access to extensive electronic hospital record databases for research purposes. Obtaining approval from local Ethical Committees becomes a cumbersome and time-consuming regulatory process, impeding research and complicating data sharing and collaboration. To strike a balance between safeguarding privacy and enabling research accessibility, **data anonymization** becomes essential. This process ensures that patient re-identification

is impossible, mitigating the risk of privacy breaches. Effective anonymization techniques can pave the way for the global release of data, democratizing access for all researchers and facilitating the use of real-world data as a foundational resource for healthcare studies.

In the quest to address the challenge of securely **sharing healthcare data while preserving privacy**, two distinct avenues have emerged:

- Strategies such as **federated machine learning, multi-party computation, and distributed learning** utilize advanced machine learning techniques to train models across a multitude of decentralized edge devices or servers that retain local data samples. Importantly, these approaches sidestep the need for direct data exchange, addressing some of the core concerns held by healthcare data centres. However, they do come with certain limitations and constraints.
- Alternatively, there are methods and tools designed to generate **synthetic health records** at scale, eliminating the risk of accidental data disclosure. This approach substantially lowers the existing barriers to entry for promising early-stage research endeavours. Furthermore, as the machine learning community's interest in synthetic data continues to grow, the synthetic datasets generated are becoming increasingly comprehensive, detailed, and lifelike over time.

Federated learning

Federated learning is a distributed machine learning (ML) approach that enables training models across decentralized devices or servers while keeping data localized and private, contrary to traditional ML, where the centralized storage of data is necessary.

Federated machine learning is gaining in popularity as data become increasingly distributed among various participants aiming to reach a common goal, making use of the collection of all participants' data. For example, in healthcare applications, the participants would consist of various hospitals or specialized clinics, each having its own collection of data collected from various patients. The goal is often a statistical learning task, such as, for example, predicting epilepsy from patients' EEG recordings. Traditional learning tasks consider that the data is readily available on a single device and can be used in its entirety. In a setting where we have multiple participants, this would require collecting the data of all participants in a central device, such as a server.

However, **data centralization tends to be avoided** in many applications for two main reasons:

- Transmitting the data to a central server is costly, especially for real-time applications where new data is continuously collected from all participants.
- Centralizing the data often leads to data privacy problems, especially in clinical settings where patients' data usually cannot be shared across hospitals.

Federated learning is a framework aiming to solve this problem by making the **participants indirectly collaborate in the learning task through a central server while avoiding sharing their data with the server or between each other.**

The key innovation of federated learning is that the training process takes place on individual devices, and only model updates, typically in the form of gradients, are shared and aggregated to improve the global model (see figure above). This approach is particularly useful when dealing with sensitive data, such as healthcare records or personal information, where data privacy is of utmost importance.

In the context of federated learning, the computation of gradients follows these steps:

1. **Initialization**: A global model is initialized centrally, typically at a server or coordinator and is transmitted to the clients (nodes).
2. **Local training**: Participating devices or nodes receive a copy of the global model and train it using their local data. Each device performs forward and backward passes on its data to calculate the gradients of the model's parameters with respect to the local loss function. After each local training round, the gradients are calculated based on the local loss and the local data. These gradients represent how the model's parameters should be adjusted to better fit the local data.
3. **Aggregation and transmission**: The calculated gradients from all participating devices are sent to the central server for aggregation. The central server computes the average or some other weighted combination of these gradients to create a new, updated global gradient.
4. **Model update**: The central server applies the aggregated gradient to the global model, adjusting its parameters to reflect the collective knowledge learned from all the participating devices.
5. **Iteration**: The process of local training, gradient calculation, aggregation, and model update is repeated for multiple rounds. Each round helps the model gradually improve its performance based on insights from diverse local datasets.

By using this process of **sharing gradients instead of raw data**, federated learning preserves data privacy. The central server never has access to individual data samples, ensuring that sensitive information remains on local devices. This is particularly advantageous in healthcare, where patient data confidentiality is critical. Federated learning is a powerful way to train accurate and robust models on data distributed across different locations while addressing privacy concerns. It strikes a balance between utilizing collective data insights and safeguarding the individual data privacy of participants.

Synthetic data

Advantages

- **Privacy**: In an ideal scenario, synthetic data closely mimics the characteristics of the original dataset, enabling analysis as if it were the authentic data. However, its significant advantage lies in its ability to address privacy concerns effectively. Synthetic data allows for the sharing and acceleration of research without jeopardizing sensitive information. When synthetic datasets are created, they **sever any direct ties to individual patients**, transforming into collections of observations that retain the statistical patterns of the original data. This transformation means that researchers from various institutions can freely exchange synthetic derivatives without encountering the same privacy and regulatory roadblocks associated with genuine patient data.
- **Time saver**: The time required to derive valuable insights from synthetic data can be significantly shorter due to **reduced regulatory oversight** compared to real data. A distinguishing feature of synthetic data is its scalability; it can be **generated in vast quantities**, facilitating more extensive experimentation and analysis. In contrast, collecting real-world data typically progresses linearly, with each additional data point consuming a similar amount of time as the previous one. The capacity for mass generation empowers researchers to explore a wider range of scenarios and hypotheses, ultimately enabling more robust and comprehensive studies.

Limitations

Despite the fact that the primary goal of synthetic data is to act as suitable replacement for real data, or comprise a realistic substitute for existing data, it has been shown that there are serious limitations

of synthetic data that must be considered. While synthetic data can be easy to create, cost-effective, and highly useful in some circumstances, there is still a heavy **reliance on human annotated and real-world data**. The only way to guarantee a model is generating accurate, realistic outputs is to test its performance on well-understood, human annotated validation data. Generating realistic synthetic data might have become easier over time, real-world human annotated data remains a necessary part of machine learning training data. Models for synthetic data generation seek for common trends in the original data, but may not cover outlier cases that the authentic data did. In some instances, this may not be a critical issue. However, in most system training scenarios, this will severely limit its capabilities and negatively impact the output accuracy. The quality of synthetic data is always dependent on the quality of the model that created it.

10.5.6 Quality and integrity of data

The quality and integrity of data play a pivotal role in the successful performance of AI systems. When we consider the implementation of AI, it is essential to recognize that the **data utilized as input is not always perfect**. In fact, it often carries with it a range of imperfections, including socially constructed biases, inaccuracies, errors, and mistakes. It is crucial to address these issues before utilizing the data for training AI models.

- One of the most prominent challenges that data quality poses, is the existence of **societal biases**. These biases can seep into data through various means, reflecting historical prejudices, stereotypes, or systemic inequalities (we will come back on types of bias in one of the following subsections). When AI systems are trained on such biased data, they can inadvertently perpetuate and even exacerbate these biases, resulting in unfair outcomes and discriminatory behaviour. Therefore, ensuring data quality involves not only addressing technical inaccuracies but also addressing the ethical implications of the data being used.
- The accuracy and reliability of data are paramount. **Inaccurate or unreliable data** can significantly undermine the performance of AI systems. The old adage "garbage in, garbage out" holds true in the context of AI. Flawed data can lead to incorrect decisions, flawed predictions, and unreliable insights, which can have real-world consequences. To mitigate these issues, data must undergo thorough scrutiny and validation.
- The integrity of data is equally crucial. It is not just the accuracy of data that matters, but also the trustworthiness of the source and the processes involved in data collection and handling. Malicious actors may attempt to **manipulate data** intentionally to subvert AI systems. This can be particularly concerning for self-learning AI systems, as malicious data input can change their behaviour in unexpected ways. To ensure data quality and integrity, the processes and data sets used in AI development must be rigorously tested and documented at every step, from the initial planning stages to training, testing, and deployment. This documentation not only facilitates transparency but also ensures accountability. Stakeholders can trace the data's journey and understand the decisions made at each stage.

FAIR principles

The **FAIR principles** of data emphasize several key aspects to ensure fairness and equity in data-related processes. These are international guidelines for research data management, aiming to facilitate the reuse of research data.

These principles are closely connected to data quality and integrity in the following ways:

- **Findable (F)**: Findability is closely related to data quality, as well-organized and documented data is easier to find and use. Ensuring that data is accurately labelled, structured, and properly

documented enhances its quality and usability. The quality of metadata and data descriptions is vital to making data findable.

- **Accessible (A)**: Data integrity is essential for accessibility. Data should be stored in a reliable and secure manner, protecting it from corruption or unauthorized access. Accessible data should also be complete and accurate, as missing or erroneous data would hinder its usability.
- **Interoperable (I)**: Interoperability is dependent on data quality and consistency. Ensuring that data is well-structured and adheres to standardized formats and metadata facilitates its integration with other datasets and systems. Data quality issues, such as inconsistencies or inaccuracies, can hinder interoperability. We will further elaborate on interoperability in the next sections.
- **Reusable (R)**: Data quality is integral to data reusability. High-quality data, which is accurate, well-documented, and free from biases, is more likely to be reusable for various research purposes. Maintaining data integrity, especially in long-term storage and preservation, is essential for continued data reusability.

Adhering to these principles not only promotes the discoverability and accessibility of research data but also ensures that the data is accurate, reliable, and free from biases, making it truly reusable for the scientific community and beyond.

10.6 Transparency

10.6.1 Zarah's quest for clarity: The importance of AI explainability

In this new subsection of the module, we will delve into the concept of **transparency**. Let's start with the following story...

One evening, as the family gathered for dinner, Zarah's thoughtful expression didn't go unnoticed. They had recently implemented an advanced AI model in her clinic to assist with patient diagnoses and treatment recommendations.

Noah, always curious, couldn't contain his curiosity any longer. "Mom, you've got that look. What's on your mind?"

Zarah sighed and replied, "Well, it's this AI system we've introduced at the clinic. It's supposed to help me make more accurate diagnoses and treatment decisions, but I just can't seem to trust it. I don't understand why it recommends certain treatments or why it makes specific decisions."

The family exchanged concerned glances, realizing that even a seasoned doctor like their mother could struggle with AI-powered healthcare tools when they lacked explainability.

Grandma Vivian, with her infinite wisdom, chimed in. "It's perfectly natural, my dear."

Aisha intervened "Trust in AI systems, especially in healthcare, is built upon understanding how they arrive at their conclusions. Without that transparency, it's hard to have confidence in their recommendations."

Eric, who had a knack for simplifying complex issues, added, "So, it's like following a GPS blindly without knowing the route it's taking you on. It might be right, but it's unsettling not to know why it's making certain choices."

Zarah nodded in agreement. "Exactly, dear. I need to know why the AI is recommending a particular treatment or diagnosis. I need to be able to explain it to my patients and ensure that the decisions align with my medical expertise."

The family discussed the importance of AI explainability further. They understood that in healthcare, the stakes were high, and decisions could have life-altering consequences. It was crucial for Zarah to have the ability to scrutinize and understand the AI's reasoning to ensure that patient well-being was always prioritized. As the family continued their discussion that evening, they recognized that the journey toward implementing AI in healthcare was a journey toward not only innovation but also understanding.

10.6.2 Explainability

Explainability is a critical aspect of AI systems, encompassing the need to comprehend both the technical workings of the AI algorithms and the human decisions influenced by these systems.

Technical explainability

Technical explainability demands that the **decision-making processes of AI systems are transparent and understandable** to human stakeholders. However, achieving explainability often involves navigating a delicate balance between enhancing a system's explainability, which may potentially reduce its accuracy, and prioritizing accuracy at the expense of explainability.

When AI systems significantly impact people's lives, it becomes imperative to demand a comprehensive explanation of the AI system's decision-making processes. This explanation should be readily accessible, timely, and tailored to the specific expertise of the stakeholders involved, whether they are laypersons, regulators, or researchers. The complexity of AI systems necessitates explanations that are comprehensible even to non-technical individuals, enabling them to grasp the underlying rationale behind the AI-driven decisions affecting their lives.

Business model transparency

Furthermore, business model transparency is essential in ensuring that stakeholders have access to comprehensive explana**tions regarding the extent to which AI systems influence and shape organizational decision-making processes**. These explanations should include insights into the design choices of the AI system and the underlying rationale for its deployment. Business model transparency not only fosters trust between stakeholders and the organization but also provides a deeper understanding of how AI systems function within the broader context of the organization's goals and strategies.

Organizations and developers must prioritize the development of AI systems that not only achieve high levels of accuracy, but also maintain a sufficient degree of explainability. Balancing these two factors requires careful consideration of the context in which the AI system operates and the impact it has on various stakeholders. By providing comprehensive and accessible explanations of AI processes, organizations can foster trust, enhance accountability, and ensure that AI systems align with ethical, legal, and regulatory standards. Ultimately, prioritizing explainability in AI systems serves to promote transparency, trust, and responsible AI development and deployment.

In the context of AI in healthcare, various aspects are **crucial for the acceptance and use of AI in clinical applications** and in daily routine:

- **Patient trust**: When an AI system is involved in diagnosing a potentially life-altering condition, patients and their families naturally have concerns. They want to trust the AI's recommendations, but trust is significantly enhanced when the AI system can explain why it arrived at a particular diagnosis or recommendation.
- **Medical professionals**: Explainability is essential for medical professionals who work alongside AI systems. Doctors need to understand why the AI made a particular diagnosis, especially

when the AI's recommendation contradicts their own assessment. This explanation can facilitate collaboration between AI and human healthcare providers.

- **Legal and ethical compliance**: In healthcare, there are strict legal and ethical standards that must be upheld. Having an AI system that can explain its diagnostic decisions is crucial for ensuring that these standards are met. It also allows for accountability in case of errors or disputes.
- **Research and continuous improvement**: An explainable AI system in healthcare can provide valuable insights for research and development. Understanding how the AI reached its conclusions can help researchers refine and improve the system over time.

Suppose an AI system is analysing a chest X-ray and detecting a potential lung abnormality. Instead of simply providing a diagnosis, an explainable AI system would not only highlight the regions of interest on the X-ray but also outline the specific features or patterns that led to its conclusion. It might point out the size, location, or shape of the anomaly and refer to specific medical literature or databases where similar cases were diagnosed. This explanation provides the medical professional and the patient with insights into **why the AI system reached a particular diagnosis**. It enables a more informed discussion and helps build trust in the AI's capabilities. If there are doubts or concerns, the medical professional can consider the explanation and conduct additional tests or consultations, leading to a more comprehensive and reliable diagnosis. In this scenario, explainability in AI-assisted diagnostics not only **enhances the diagnostic process**, but also ensures that healthcare practitioners, patients, and regulatory bodies can **trust and validate the recommendations**, ultimately leading to better patient care and outcomes.

According to our expert Rob Heyman, full explainability might not be necessary and only adequate explainability might be required. The level of necessary explainability must be decided based on the use case and the explainability needs of the stakeholders. For example, the activation of airbags during driving does not need to be fully explainable on the technical specifications, but you must focus on maximizing reliability.

10.6.3 Traceability

Traceability is a fundamental concept in AI and data science, particularly in the context of AI system decision-making in various industries, including healthcare, finance, and autonomous systems. This principle underscores the importance of documenting data sets, data processing processes, and AI algorithms to a high standard, enabling transparency, accountability, and the identification of errors.

Traceability is a cornerstone of responsible AI, offering a means to ensure the reliability and quality of AI systems.

Aspects of traceability

- One of the primary aspects of traceability is **data collection and labelling**. To maintain high standards of data traceability, it is essential to document the sources of data, data collection methodologies, and any human input or labelling involved. Proper documentation allows stakeholders to understand the data's origins and assess the data's quality, reducing the likelihood of biased or erroneous data impacting AI decisions.
- Moreover, **documenting** the **algorithms and processes** used in AI development is pivotal for ensuring traceability. Transparency in algorithmic decision-making is essential. Documenting the algorithms used, including their parameters and configurations, allows for a clear understanding of how AI arrives at its decisions. This documentation aids not only in explaining

the AI system's decisions but also in identifying any potential biases or errors within the algorithms.

- The importance of traceability extends beyond the technical realm to include **AI system decisions**. When an AI system makes a decision, it should be documented in a way that enables stakeholders to trace back to the rationale behind that decision. This means recording the factors considered, the data sources consulted, and the specific algorithmic procedures applied. Such documentation is essential for understanding and auditing the decision-making process.

Advantages of traceability

- One of the key advantages of traceability is its contribution to **error identification and prevention**. By thoroughly documenting data, algorithms, and decision-making processes, organizations can pinpoint the reasons behind erroneous AI decisions. This information is invaluable in terms of refining the AI system, making necessary adjustments, and ultimately preventing similar mistakes in the future. This iterative process of learning from errors enhances the reliability and safety of AI applications.
- In addition, traceability is closely linked to **auditability and explainability**. By documenting data, processes, and decisions, an organization can provide a transparent record that can be audited to ensure compliance with regulations and ethical standards. Moreover, when an explanation of an AI decision is required, a well-documented trace allows stakeholders to understand why a particular decision was made, promoting transparency and trust.

In conclusion, traceability is an indispensable aspect of responsible AI. It ensures that data, algorithms, and AI decisions are meticulously documented, promoting transparency, accountability, and reliability. Through traceability, organizations can identify and rectify errors, enhance their AI systems, and maintain a higher standard of quality in decision-making processes. By adhering to traceability principles, businesses and institutions can gain the trust of stakeholders and ensure that AI systems are aligned with ethical, regulatory, and operational standards.

10.6.4 Risk of openness

Openness in AI, often synonymous with the sharing of data, algorithms, and research findings, offers numerous advantages in advancing the field, driving innovation, and democratizing access to AI capabilities. However, it also brings about certain risks, particularly concerning intellectual property (IP) and the protection of proprietary information.

Compromise of intellectual property

One of the primary risks associated with openness in AI is the potential compromise of intellectual property. Intellectual property refers to a broad category of legal rights that protect creations of the mind, such as inventions, innovations, and creative works. In the context of AI, intellectual property may include patented algorithms, proprietary datasets, or unique AI models. When organizations or individuals openly share their AI research, there's a risk that their intellectual property may be exposed to others who could potentially **replicate, modify, or use it without permission**. This can have adverse effects on the incentives for innovation and investment in AI. If innovators fear that their proprietary AI technologies may not be adequately protected, they might be less motivated to engage in AI research and development.

Data breaches or misuse of patient information

Furthermore, the risk extends to data privacy and security. Healthcare, for example, deals with highly sensitive and personal patient data. When healthcare institutions open up their AI research, there's a

potential for data breaches or misuse of patient information, which can lead to serious legal and ethical consequences. Striking the right **balance between openness and data protection** is crucial. That is mainly the reason that AI applications in healthcare lack in reproducibility.

Solutions

To mitigate this risk, organizations in the healthcare industry, and AI more broadly, often adopt strategies that balance openness with IP protection. They may publish research findings while withholding sensitive details that could jeopardize their IP. Alternatively, they might engage in collaborations, where shared data and insights are safeguarded through contractual agreements. In the context of healthcare, especially when dealing with patient data, organizations should prioritize data protection and privacy. Compliance with regulations like the HIPAA in the United States is crucial and GDPR in EU. Sharing healthcare-related AI research openly while ensuring data anonymization and strict access controls can help mitigate the risk of data breaches and protect patient privacy.

In conclusion, while openness in AI fosters collaboration, innovation, and knowledge sharing, it also poses risks to intellectual property and data security. Striking a balance is essential. Organizations must carefully consider the potential consequences of open sharing, especially in sensitive areas like healthcare, and take measures to protect proprietary AI technologies and ensure data privacy and security. This balanced approach allows the AI field to benefit from open research while protecting valuable innovations and sensitive data.

10.7 Fairness and bias

10.7.1 Types of bias

In this new subsection, we will discuss **fairness and bias**.

Bias, both in general contexts and within AI systems, can significantly impact decision-making processes, perpetuate inequalities, and hinder progress. In healthcare, where decisions directly influence patient outcomes and well-being, understanding the different types of bias is crucial for delivering equitable and effective care.

Let's dive deeper into various types of bias, both general and specific to AI, and their implications in healthcare.

- Societal biases are ingrained prejudices and stereotypes existing within a society, often based on factors such as **race, gender, or socioeconomic status**. In healthcare, societal biases can lead to disparities in treatment and outcomes for patients from marginalized communities. For example, studies have shown that racial biases can influence the quality of care and the types of treatments offered to patients, leading to unequal healthcare access and poorer health outcomes for certain racial groups.
- Sample bias occurs when the data used for analysis is not representative of the entire population, leading to skewed results. In healthcare, sample bias can occur when research studies predominantly include participants from specific demographics, such as certain **age groups or geographic regions**. This can limit the generalizability of medical research findings and lead to treatments that are not universally effective.
- Algorithmic bias refers to biases that arise from the **design and implementation of AI algorithms**, leading to unfair outcomes for certain groups. In healthcare AI, algorithmic bias can manifest in various ways, such as biased diagnostic outcomes or treatment recommendations. For example, if an AI diagnostic tool has been trained on datasets that are

not diverse enough, it may fail to accurately diagnose certain conditions in specific demographic groups, leading to disparities in healthcare delivery.

- Confirmation bias occurs when decision-makers favour information that confirms their **preconceived beliefs or hypotheses**, while ignoring contradictory evidence. In healthcare, this bias can lead to diagnostic errors or the overlooking of critical symptoms, potentially impacting patient treatment and care. For instance, a physician influenced by confirmation bias might ignore certain symptoms that do not align with their initial diagnosis, leading to delayed or incorrect treatment plans.
- Cognitive biases are **inherent tendencies or patterns in human thinking** that can affect decision-making processes. In healthcare, cognitive biases can impact clinical judgment and treatment choices. An example is the availability bias, where clinicians rely on readily available information in their memory, potentially overlooking less common but critical diagnoses or treatment options.

In healthcare AI applications, biases can be inadvertently integrated into the algorithms, leading to **unequal or inaccurate outcomes for certain patient groups**. For instance, if an AI system is trained on historical patient data that predominantly represents one demographic group, it may not accurately capture the healthcare needs of other demographics, leading to biased treatment recommendations or diagnoses. This can exacerbate existing healthcare disparities and result in unequal access to appropriate care for marginalized communities.

Furthermore, when developing AI-driven predictive models for disease risk assessment or treatment planning, the absence of diverse and comprehensive datasets can lead to **skewed results**. If certain demographic groups are underrepresented in the training data, the AI model may not provide accurate risk assessments or treatment plans for these groups, perpetuating healthcare disparities and affecting patient outcomes.

Moreover, the integration of AI in decision support systems can inadvertently **amplify cognitive biases**. If AI systems are designed without accounting for cognitive biases that clinicians may possess, there is a risk of reinforcing these biases within the decision-making process, potentially leading to suboptimal patient care and outcomes.

Addressing these biases in healthcare AI requires a multifaceted approach. It involves ensuring diverse and representative datasets for training AI models, implementing robust validation processes to detect and mitigate biases, and promoting diversity and inclusivity in AI research and development teams. Additionally, healthcare institutions must prioritize ongoing education and training for clinicians and developers on recognizing and mitigating biases to foster a more equitable and effective healthcare system.

By acknowledging and actively addressing various types of bias in healthcare and AI, stakeholders can work towards building fairer, more accurate, and inclusive healthcare systems that prioritize the well-being and outcomes of all patient groups.

10.7.3 Accessibility

In the realm of AI development and deployment, **accessibility and universal design principles** play a pivotal role in ensuring **equitable access and use of AI products and services**. These principles are essential to upholding the rights and inclusion of all individuals, regardless of age, gender, abilities, or characteristics. The significance of accessibility in AI cannot be overstated, particularly in light of the diverse and varied user base that these systems serve. In business-to-consumer domains, creating AI systems that are user-centric and consider the diverse needs of individuals is paramount.

AI systems should not adhere to a one-size-fits-all approach. Instead, they should be designed with the **flexibility** to accommodate a wide range of user characteristics, abilities, and needs. This concept is encapsulated in the principles of universal design, which focus on addressing the needs of the broadest spectrum of users possible, irrespective of their individual traits.

Universal design and accessibility standards

Universal design principles advocate for creating products and services that can be used and understood by as many people as possible, without the need for special adaptations or design alterations. This approach recognizes that diversity is an inherent aspect of human existence and seeks to promote inclusivity by taking this diversity into account.

In the context of AI, following relevant accessibility standards and guidelines is instrumental in realizing universal design principles. These standards encompass a wide array of considerations, ranging from **user interface design** to ensuring compatibility with **assistive technologies**. For example, ensuring that AI interfaces are navigable by screen readers for users with visual impairments or making AI-powered content available in multiple languages to cater to diverse linguistic communities.

By embracing these principles, AI systems can provide equitable access and active participation for all individuals in computer-mediated human activities. Moreover, they promote compatibility with assistive technologies, which are essential tools for individuals with disabilities. For instance, **text-to-speech applications, screen readers, and voice recognition software** are assistive technologies that enable individuals with visual or mobility impairments to access and interact with AI systems.

Stakeholder participation

In the development of AI systems that prioritize trustworthiness and ethical considerations, engaging stakeholders is crucial.

Stakeholders are individuals, groups, or entities who may be directly or indirectly affected by the AI system throughout its life cycle. This includes not only end-users but also those involved in the design, development, deployment, and regulation of AI.

Soliciting **feedback** and involving stakeholders in the decision-making process is a valuable practice. Their insights can help in identifying potential ethical concerns, biases, or unintended consequences. Moreover, their perspectives can contribute to refining AI systems to better meet the needs and expectations of the various user communities they serve.

Stakeholder participation should not be confined to the **initial stages** of AI development; it should extend throughout the system's life cycle, including the **post-deployment phase**. Establishing long-term mechanisms for ongoing feedback, consultation, and participation ensures that AI systems remain aligned with the evolving needs of their users and the broader community.

For example, in the workplace, involving workers in the implementation of AI systems can help address concerns related to job displacement, workplace dynamics, and the ethical use of AI technologies. It fosters a collaborative and transparent approach to AI integration, promoting trust and fairness.

In conclusion, promoting **accessibility, universal design, and stakeholder participation** in AI systems is integral to creating technology that respects the rights and inclusion of all individuals. By adhering to these principles, AI developers and organizations can ensure that their systems are not only technically proficient but also ethically sound and user-centric. This approach contributes to a more equitable and inclusive digital landscape, where AI serves as a tool for the betterment of society as a whole.

10.8 Sustainability and societal impacts

10.8.1 Noah's vision: Embracing green AI for a sustainable future

In this new subsection, we will delve into **sustainability and societal impacts** of AI. Let's start with the following story...

One afternoon, the family members gathered in their living room to discuss the family's growing reliance on AI technology and its environmental impact. Noah had been doing some research and couldn't wait to share his findings.

"Hey, everyone," Noah began, "I've been thinking about our increasing use of AI, especially in healthcare. While it's incredibly helpful, it consumes a lot of electricity, and that's not good for the environment."

Eric, Zarah, Aisha, and Vivian listened attentively, knowing that Noah's concerns were rooted in his deep respect for nature and his desire to protect it. Eric asked, "What do you mean, Noah? How does AI consume so much power?"

Noah explained, "AI systems, especially those in healthcare, use vast amounts of computational power for tasks like analysing medical data and running complex algorithms. This requires energy from power plants, many of which rely on fossil fuels, which contribute to greenhouse gas emissions and climate change."

The family exchanged contemplative glances, realizing that their use of AI was not without consequences. Aisha, with her knack for finding solutions, suggested, "So, what can we do to make AI more eco-friendly, Noah?"

Noah's face brightened. "Well, we can look into using green AI. Green AI focuses on developing AI systems that are energy-efficient and have a lower carbon footprint. This way, we can still enjoy the benefits of AI without harming the environment."

Eric nodded approvingly. "That sounds like a responsible approach. We should also consider using renewable energy sources to power our AI systems, like solar or wind energy."

The family discussed ways they could reduce the environmental impact of their AI usage, from selecting energy-efficient AI systems to advocating for green AI initiatives within their community.

10.8.2 Green AI

Large-scale AI models have a substantial environmental impact, primarily stemming from multiple factors, including the extensive number of parameters within these models, the efficiency of the data centre power usage, and the overall grid efficiency. Among the AI models, **GPT-3** stands out as the most significant carbon emitter, given its sheer size and complexity. However, even the comparatively more efficient BLOOM model consumed a significant amount of power during its training process, totalling 433 megawatt-hours (MWh). To put this into perspective, the energy consumption of training BLOOM alone could sustain the average American household for an astonishing 41 years.

This raises important concerns regarding the environmental footprint of AI technologies, especially as the development and deployment of large AI models become increasingly prevalent. The energy-intensive nature of these models necessitates a focus on energy-efficient training methods, the utilization of renewable energy sources, and a concerted effort to mitigate the ecological consequences associated with their use. It highlights the need for sustainable practices and greater consideration of

the environmental implications as AI continues to advance and become an integral part of various industries.

How can we train the AI models more sustainably?

In an era characterized by an increasing reliance on AI technologies, there is a growing need to address their environmental impact. AI, especially large-scale machine learning models and data centres, is known for its substantial energy consumption and carbon emissions. To counter this, the concept of "**green AI**" has emerged as a significant stride towards more sustainable and environmentally-friendly AI development.

One of the primary approaches to achieving green AI is through **optimizing AI models for energy efficiency**. This entails designing algorithms that require fewer computational resources and minimizing unnecessary processing. Techniques like model pruning, quantization, and knowledge distillation help create leaner models that still perform effectively. Additionally, utilizing low-power hardware and specialized AI accelerators can further reduce energy consumption.

Green AI emphasizes the use of **renewable energy sources** to power data centres and AI infrastructure. Transitioning to clean energy sources, such as solar, wind, or hydropower, can significantly reduce the carbon footprint of AI operations. Tech giants like Google and Apple have made strides in this direction by committing to powering their data centres with 100% renewable energy.

Modern data centres and AI hardware are increasingly designed for efficiency. This includes the use of energy-efficient **cooling systems**, such as free cooling, that reduce the energy required to maintain the optimal operating temperature of servers. Furthermore, deploying advanced cooling techniques like liquid cooling can enhance energy efficiency. Strategically siting data centres in regions with temperate climates can leverage natural cooling to reduce the need for energy-intensive cooling systems. Innovative data centre designs, such as modular and containerized data centres, enable scalability and efficient use of resources.

Green AI is an area where **collaboration** is pivotal. Open source initiatives and partnerships among tech companies, research institutions, and governments can facilitate knowledge-sharing and the development of energy-efficient AI solutions. These collaborations often lead to best practices and technologies that promote sustainability.

Green AI isn't solely about energy efficiency during operation. It also includes considering the en**tire lifecycle** of AI technologies, from design and manufacturing to usage and disposal. Minimizing waste and adopting responsible recycling practices for outdated AI hardware are integral components of sustainability.

Sustainability in AI goes hand in hand with **ethical considerations**. Ensuring that AI models and algorithms do not perpetuate biases or contribute to environmental harm is a critical aspect of Green AI. Ethical AI practices, such as fairness and transparency, are integral to the development of responsible and sustainable AI solutions.

Green AI represents a holistic approach to reducing the environmental impact of artificial intelligence. It acknowledges the pivotal role AI plays in addressing global challenges and strives to make AI more environmentally sustainable. As the AI community continues to grow, it's essential to prioritize sustainability and develop AI systems that not only enhance our capabilities but also safeguard the planet. Through ongoing research, innovation, and commitment, Green AI aims to make artificial intelligence a driving force for positive change while minimizing its carbon footprint.

10.8.3 Societal cohesion and social impact

The pervasive integration of social AI systems into various aspects of our lives, spanning education, employment, healthcare, and entertainment, carries profound implications that extend well beyond mere technological convenience. As these systems become increasingly ubiquitous, we find ourselves on the cusp of a transformative journey, one that stands to reshape our fundamental notions of social agency and redefine our relationships and attachments.

While AI systems undoubtedly offer the potential to **enhance our social aptitudes**, they also hold the power to **contribute to their deterioration**, a duality that holds implications for our physical and mental well-being. Consequently, it becomes imperative to subject the effects of these systems to meticulous scrutiny and contemplation.

Social impact and the balancing act

AI's foray into the realm of social interaction ushers in a complex interplay. On one hand, these systems can act as facilitators, **bolstering our social skills** and aiding us in navigating intricate social dynamics. They hold the potential to provide support in contexts such as social education, assisting individuals in building effective communication and interpersonal abilities. On the flip side, the same systems can inadvertently contribute to the erosion of these skills. **Relying excessively on AI**-mediated communication may, for instance, impede our capacity for authentic, emotionally rich human interaction, posing a subtle but significant challenge to our social fabric. Moreover, the transformation in our social dynamics and relationships due to these technologies can bear repercussions on our physical and mental well-being. It becomes paramount to maintain a vigilant watch over these effects and ensure that they align with our collective vision for a harmonious and thriving society.

Society, democracy, and the collective impact

The transformative influence of AI transcends individual implications and seeps into the very foundations of society. Assessing the ramifications of AI's development, deployment, and use necessitates a shift in perspective—from a focus on the individual to a broader societal view. AI systems wield a formidable impact on our institutions, democracy, and society at large. In the realm of democracy, they infiltrate not only **political decision-making** but also electoral contexts. AI's involvement in **electoral processes** is an example of a nuanced and multifaceted challenge. While it can potentially optimize aspects of the electoral process, like the management of voter data or the analysis of campaign strategies, it simultaneously raises concerns related to voter privacy, electoral manipulation, and transparency. These systems can influence political discourse and voter behaviour, demanding a careful balance between technological advancement and the safeguarding of democratic values.

Crucial need for deliberation

In light of these multifarious implications, we are compelled to **engage in deliberate and comprehensive discussions and considerations**. The integration of AI systems into our social fabric and democratic processes is a multifaceted endeavour that necessitates a fine-tuned orchestration of technological advancement and societal preservation. We must be vigilant, for technology, though a remarkable ally, can become a double-edged sword when not harnessed with care.

To navigate the era of social AI effectively, a collaborative effort is called for. This includes scholars, technologists, policymakers, and the wider public coming together to grapple with the profound transformations that AI is ushering in. It involves ethical considerations, regulatory measures, and a keen awareness of how AI is permeating our lives, our societies, and our democracies.

In conclusion, the ascendancy of social AI systems in our lives heralds a paradigm shift that goes beyond mere technological convenience. It touches the very essence of our humanity, our social dynamics, and the functioning of our democratic institutions. While the potential for positive change is immense, the challenges are equally significant. It is only through a harmonious blend of technological advancement, ethical vigilance, and societal reflection that we can steer the course of AI toward a future that aligns with our collective aspirations for a better world.

10.9 Accountability and human oversight

10.9.1 Accountability unveiled: The human in the loop

In this last subsection of the module, we will discuss **accountability and human oversight**. Let's begin with the following story first...

Grandma Vivian was talking to a friend on the phone about how she got the wrong suggestion on the diabetes management app (see previous story). Fortunately, the crisis was averted, but questions remained about who would be held accountable for the potentially life-threatening recommendation.

The phone talk reinitiated the discussion within the family about the incident. Soon, it became clear that, while AI in healthcare has its merits, accountability is a complex issue. Eric raised a crucial point, "The AI system is a tool, but the ultimate responsibility for the patient's health lies with the healthcare provider and the patient themselves. We should use AI as a valuable resource, not as a replacement for human judgment."

Zarah added, "AI can provide valuable insights, but we need to maintain a human in the loop, especially in critical decisions. We should never blindly follow AI recommendations, and healthcare providers must exercise their judgment."

The family realized that the human in the loop was essential for maintaining accountability in AI in healthcare. Grandma Vivian's experience served as a stark reminder that blindly trusting technology could have severe consequences. As the family continued to support Vivian in managing her diabetes, they remained committed to using AI as a tool rather than a replacement for human judgment. They understood that accountability in AI healthcare was a shared responsibility, and the human touch was irreplaceable when it came to safeguarding patient well-being. It was a lesson they would carry with them as they navigated the evolving landscape of healthcare technology and its ethical complexities.

10.9.2 Governance mechanisms: HITL, HOTL, or HIC

In the ever-evolving landscape of AI, one of the fundamental tenets is the incorporation of human oversight to safeguard human autonomy and prevent adverse consequences. Human oversight serves as a critical counterbalance to the capabilities of AI systems, allowing us to shape the technology's development, operation, and its broader societal implications. This oversight can be realized through various governance mechanisms, such as **"human-in-the-loop" (HITL), "human-on-the-loop" (HOTL), and "human-in-command" (HIC)** approaches. These mechanisms provide the necessary checks and balances to ensure AI is harnessed for the greater good without sacrificing core human values.

Human-in-the-loop (HITL): Balancing intervention with feasibility and desirability

At its core, HITL emphasizes the capability for human intervention **in every decision cycle** of an AI system. While this approach ensures a high degree of human control, it may not always be feasible or desirable. HITL may be suitable in contexts where critical decision-making necessitates real-time human input, such as complex medical diagnoses or military applications. In such cases, human intervention can ensure ethical and accurate outcomes. However, in many scenarios, it is neither

practical nor efficient to have humans directly involved in every decision cycle, as it may hinder the speed and scale of AI-driven processes.

+ **High quality and accuracy**: HITL ensures that human expertise is directly involved in decision-making, resulting in high-quality and accurate outcomes.
+ **Customization**: It allows for customization and adaptation to specific situations and preferences.
+ **Continuous learning**: Human oversight enables ongoing learning and improvement of AI systems.
- **Time-consuming**: Real-time human intervention can slow down processes, making it less suitable for time-sensitive applications.
- **Resource-intensive**: HITL often requires significant human resources and associated costs.
- **Dependency**: Overreliance on human oversight can limit the full potential of AI systems.

Example: HITL in radiology involves AI systems assisting radiologists in interpreting medical images. The AI system highlights areas of interest, potential anomalies, or regions that merit closer examination. Radiologists maintain the final decision-making authority.

Human-on-the-loop (HOTL): Guiding the design and monitoring process

HOTL, on the other hand, strikes a balance by enabling human intervention **during the design cycle** of the AI system and the **ongoing monitoring** of its operation. This approach acknowledges that while real-time intervention may not be required, having human oversight during the system's development and continuous operation is pivotal. During the design phase, human experts can set ethical and operational guidelines, define decision boundaries, and ensure that the AI system aligns with human values. Subsequently, ongoing monitoring allows humans to detect anomalies, correct biases, and assess whether the AI system remains within acceptable ethical boundaries.

+ **Efficiency**: HOTL strikes a balance between AI automation and human intervention, optimizing efficiency.
+ **Timely intervention**: It allows for timely human intervention when necessary, reducing errors and risks.
+ **Continuous monitoring**: Ongoing monitoring ensures that AI systems remain within ethical and operational boundaries.
- **Delay**: While more efficient than HITL, HOTL still introduces a delay before human intervention.
- **Expertise required**: Skilled human experts are needed to monitor AI systems effectively.
- **Limited autonomy**: AI systems may remain partially dependent on human oversight, limiting their independence.

Human-in-command (HIC): Broader control and ethical decision-making

HIC extends human oversight to the macro level. It encompasses the ability to **oversee the overall activity** of the AI system, including its economic, societal, legal, and ethical impact. Moreover, HIC includes the **authority** to determine when and how the AI system is employed in specific situations. This encompasses the ability to decide against using the AI system when it may lead to adverse consequences. HIC can also establish varying levels of human discretion during system operation, ensuring that ethical considerations guide the system's actions. Additionally, HIC allows humans to override decisions made by the AI system when they conflict with human values or ethical norms. HIC is instrumental in ensuring that AI is wielded responsibly and ethically across different domains.

+ **Ultimate control**: HIC ensures that humans retain ultimate authority and decision-making power, particularly in critical situations.

+ **Ethical oversight**: It allows for ethical and value-based decisions, aligning AI actions with human values.
+ **Flexibility**: Healthcare providers have the flexibility to adapt AI recommendations to unique patient cases.
- **Possible errors**: Depending solely on human decisions might lead to occasional mistakes or biases.
- **Resource-intensive**: HIC can be labour-intensive and may extend decision-making time, especially in complex situations.
- **Resistance to AI**: Over-reliance on human command might hinder AI adoption and limit the benefits it offers in terms of efficiency and scalability.

The general pros and cons that we mentioned, provide an overview of the considerations associated with each methodology in various AI applications, not specifically to healthcare. The choice between HITL, HOTL, and HIC depends on the context, the balance required between human expertise and AI efficiency, and the potential risks associated with the application.

Public enforcer oversight: Aligning with mandates for ethical AI governance

To reinforce the role of human oversight, it is imperative that **public enforcers**, such as regulatory bodies and governmental agencies, have the capacity to exercise oversight in line with their mandates. Public enforcers play a crucial role in ensuring that AI systems adhere to legal and ethical norms, and they must have the authority and capability to intervene and enforce compliance.

The necessity for **oversight mechanisms** varies based on the application area and potential risks associated with an AI system. In high-stakes domains, such as healthcare, autonomous vehicles, and finance, greater oversight may be required to mitigate potential adverse effects. In these scenarios, a combination of HITL, HOTL, and HIC may be necessary to safeguard against unintended consequences. Conversely, in less critical areas, where AI supports routine tasks, oversight mechanisms can be more relaxed.

Oversight mechanisms should not exist in isolation but should be integral components of a broader strategy for AI safety and control. The less direct human oversight a particular AI system permits, the more rigorous testing and stricter governance measures are imperative. This synergy between oversight, testing, and governance creates a robust framework for the responsible development and deployment of AI.

In conclusion, human oversight in AI systems is a multi-faceted concept that ensures ethical and responsible AI. HITL, HOTL, and HIC approaches provide flexibility in the level of human intervention required, depending on the application and risk profile. Public enforcers further reinforce these mechanisms, ensuring that AI aligns with legal and ethical norms. It is through this collaborative effort, striking the right balance between oversight and autonomy, that we can harness the full potential of AI while preserving our core values and principles. The future of AI is one in which technology and humanity walk hand in hand, in harmony and responsibility.

10.9.3 Accountability and auditability

The advancement and integration of AI into our lives necessitate a robust framework of principles and requirements to ensure not only its functionality but also its ethical and responsible application. One of the foundational requirements in this framework is the principle of accountability. **Accountability** is closely aligned with the overarching concept of fairness and extends to mechanisms that oversee the entire lifecycle of AI systems, including their development, deployment, and usage.

Accountability within the context of AI demands that systems and their outcomes are subject to responsibility and oversight. This accountability encompasses both proactive measures before AI systems are put into operation and reactive responses to their impacts. The aim is to ensure that AI serves the broader good and adheres to the values and principles of fairness, transparency, and responsible innovation.

Auditability: Illuminating the black box of AI

Auditability is a crucial component of accountability, emphasizing the need to examine the inner workings of AI systems. This does not necessarily imply revealing proprietary business models or intellectual property, but it does require that the mechanisms for assessing algorithms, data, and design processes are in place. These assessments can be performed by both internal and external auditors, and the resulting evaluation reports contribute significantly to the trustworthiness of AI technology.

In cases where AI systems impact fundamental rights or involve safety-critical applications, the ability for **independent auditability** is essential. This means that external experts or regulators should be able to evaluate the AI system's functioning, decision-making processes, and potential biases. Such independent audits serve as a check and balance, ensuring that AI systems adhere to ethical and legal norms.

Accountability demands not only the ability to **report** actions or decisions contributing to specific AI system outcomes, but also the capacity to **respond** to the consequences of such outcomes. Identifying, assessing, documenting, and minimizing the potential negative impacts of AI systems is particularly crucial for those directly or indirectly affected. It necessitates robust mechanisms for reporting concerns, thereby protecting whistle-blowers, non-governmental organizations (NGOs), trade unions, or other entities raising legitimate issues about AI systems.

To mitigate negative impacts, the use of **impact assessments**, including methods like "red teaming" or Algorithmic Impact Assessments, proves instrumental. These assessments can be employed both before and during the development, deployment, and utilization of AI systems. Their application must be proportionate to the risks posed by the AI systems. Assessing potential harm, biases, or ethical dilemmas is a proactive way to identify and rectify issues early in the AI development process.

In the pursuit of accountability, tensions may arise among the various requirements. These tensions often result in inevitable **trade-offs**, where one ethical principle may conflict with another. Addressing these trade-offs necessitates a rational and methodological approach, acknowledging that they may be an inherent part of AI development.

To navigate these trade-offs, it is vital to identify the relevant interests and values at stake. When conflicts arise, these trade-offs should be explicitly recognized and evaluated in terms of their risk to ethical principles and fundamental rights. If ethically acceptable trade-offs cannot be identified, the development, deployment, and use of the AI system should not proceed in that form. Any decision regarding trade-offs must be well-reasoned, documented, and subject to accountability. Decision-makers are responsible for how the trade-off is made and should continually review its appropriateness.

When unjust adverse impacts occur, mechanisms for **redress** should be accessible to ensure that individuals or entities affected have avenues for adequate recourse. Knowing that redress is possible when things go wrong is essential for fostering trust in AI systems. Particular attention should be paid to vulnerable persons or groups who may be disproportionately affected by AI's adverse consequences.

In conclusion, accountability in AI forms a cornerstone of ethical and responsible development, deployment, and usage of AI systems. It encompasses auditability to unveil the inner workings of AI, mechanisms to minimize and report negative impacts, a systematic approach to addressing trade-offs, and accessible redress mechanisms for affected parties. By embedding accountability into the AI lifecycle, we not only enhance transparency but also build trust and ensure that AI aligns with ethical principles and respects the fundamental rights of individuals and society. This multifaceted approach is vital for creating a responsible AI ecosystem that benefits humanity while minimizing harm.

Accountability and auditability in AI healthcare

Example 1: Clinical decision support systems

- **Accountability**: In clinical decision support systems used in healthcare, accountability is essential. These systems provide recommendations to healthcare professionals, and it's crucial to track the decisions made based on these recommendations. The accountability mechanism involves the ability to trace back to the AI system's output and determine which recommendations were followed. This is particularly important in cases where AI recommendations may have an adverse impact on patient care.
- **Auditability**: Auditability in this context requires the ability to assess not only the AI algorithms but also the data used to train these systems. Independent auditors, which could be external experts or internal quality assurance teams, examine the decision-making processes within the AI system. They review how data is collected, processed, and used to generate recommendations, helping identify potential biases or inaccuracies. This is vital in healthcare to ensure that AI-based decisions align with medical best practices and ethical standards.

Example 2: Medical imaging and radiology

- **Accountability**: In the field of medical imaging and radiology, AI is increasingly used to assist radiologists in interpreting images such as X-rays, MRIs, and CT scans. Accountability in this context involves ensuring that the final diagnosis is attributed to the responsible parties - both the AI system and the radiologist. If there is a discrepancy or a misdiagnosis, it should be clear which entity (human or AI) was accountable for the outcome. Accountability mechanisms help in identifying the source of errors or inaccuracies.
- **Auditability**: Auditability in medical imaging AI entails the ability to scrutinize the functioning of the AI algorithm, especially in situations where the technology is used for critical diagnoses. External auditors, regulatory bodies, or medical associations can independently assess the AI system's performance. They examine how the algorithm processes images, identifies anomalies, and generates diagnostic suggestions. Additionally, they can review the data used for training the system to ensure that it represents diverse patient populations. Auditing helps ensure the technology's accuracy, safety, and compliance with medical standards.

These examples illustrate how accountability and auditability are vital components in AI healthcare applications. They help maintain transparency, traceability, and oversight in AI systems, ultimately ensuring that AI-driven decisions align with ethical, medical, and legal standards, while also minimizing potential negative impacts on patient care.

11 AI in action: Kidney transplant rejection classification

11.1 Welcome to Module 11

Welcome to Module 11. This is one of several "AI in action" modules, where we shift our focus from theoretical concepts to practical use cases, allowing clinical and technical experts to share their insights and experiences with AI applications in the healthcare field.

Why the "AI in action" modules matter

These modules about real-life use cases are significant as they bridge the gap between theory and real-world impact. By delving into the insights of experts who have developed or utilized AI applications in healthcare, you will gain a nuanced understanding of the transformative potential of AI in the field. Each use case offers a unique perspective on the practical challenges, risks and opportunities that AI brings to healthcare.

This module is about kidney transplant rejection classification. Discover how AI can be applied in the field of kidney transplantation through interview clips with an expert in this field.

Learning goals

- Examine case studies and real-world applications where AI has been successfully integrated into the clinical practice, and in this use case: kidney transplant rejection.
- Learn the importance of data preprocessing and harmonization, including the use of standardized protocols like the Banff Classification for kidney transplant biopsy analysis.
- Gain knowledge on various AI techniques and models suitable for healthcare data, particularly unsupervised and semi-supervised learning methods for discovering new phenotypes in kidney transplant rejection.
- Explore the process of model training, including splitting, validation, and hyperparameter tuning, with a focus on clustering techniques and their application in medical data analysis.
- Assess the potential impact and benefits of AI in healthcare, including the challenges of privacy, data quality, and the future role of AI in augmenting medical professionals' decision-making processes.

11.2 Context and clinical impact

Zarah felt hopeful as her patient, who had long awaited a kidney transplant, finally received the life-saving surgery. Hope soared, only to be shattered by the cruel reality of graft failure. It had especially been difficult to estimate the disease activity and injury severity of the kidney transplant.

As she consoled her patient and reflected on the unpredictable nature of transplant outcomes, Zarah felt the need for a more precise means of assessing kidney transplant failure. What if there is the possibility of creating an AI tool to provide quantitative insights into the disease process?

Although kidney transplantation is considered the first-choice therapy for kidney failure, the risk of graft failure remains a problem. The cause of **kidney transplant failure** is complex and multifactorial. Although some (acute) inflammatory lesions are observed mainly early after transplantation, others accumulate over time and reflect chronic injury processes, sometimes already ongoing in the donor before transplantation.

Currently, no **validated data-driven system** exists to realistically **characterize the chronic pathology** of kidney transplants that represents the dynamic disease process and spectrum of disease severity. In

this use case, we will explore the development of a tool describing the chronicity and severity of renal allograft disease, and integrate it with the evaluation of disease activity.

The evaluation of total chronicity provides information on kidney transplant pathology that **complements the estimation of disease activity** from acute lesion scores. Using a data-driven algorithm may provide a holistic and quantitative assessment of kidney transplant injury phenotypes and severity.

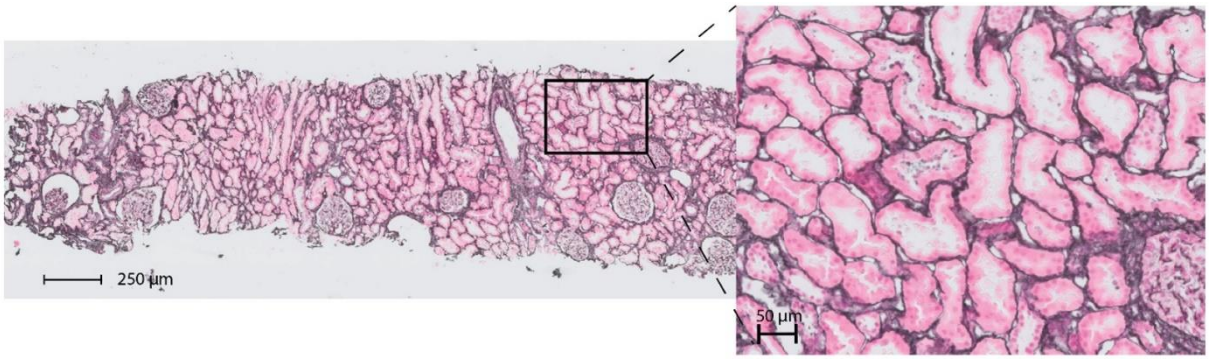
11.3 Data

Data preprocessing and harmonization

The data in this use case are highly standardized as they follow a strict protocol. This means that, everywhere in the world, pathologists of transplantation centres follow the exact same protocol to define and score histological lesions of kidney transplant. This protocol, known as the **Banff Classification**, has been developed by expert-consensus over more than 20 years.

It consists of two parts:

- First, a rigorous description of well-defined **histological lesions** and corresponding **semi-quantitative scores** is performed on the whole biopsy slide. The figure below is an example of a kidney transplant biopsy stained with haematoxylin and eosin stain (from Vaulet et al.):



- Then, a set of "**if-then-else**" rule statements combines those lesions to reach a final histological diagnosis (e.g., antibody mediated rejection). This process is somehow similar to decision trees where at each internal node a binary decision is made based on a single variable.

Of course, despite the standardization, the process still remains pathologist-dependent. Fortunately, in our case, all the biopsies were assessed and scored by the same pathologist, which virtually removes inter-observer variability. When biopsies are scored by different pathologists, there is unfortunately no easy way to adjust for a potential inter-observers' variability. One solution to mitigate this is consensus scoring, where the same slide is graded from different pathologists, and a consensus diagnosis is achieved with majority voting. This is obviously costlier and more time intensive.

Data quality, artifact removal, denoising

All the data are checked to ensure that they are in the **correct format**, and that there are **no outliers** which would suggest potential encoding errors.

Data normalization

Data normalization is **not applicable** in this use case. Every lesion is simply scaled to the unit interval to make them comparable for subsequent steps.

Ethical considerations regarding (re)use of data

The medical data used in this use case are retrospective data coming from UZ Leuven and approved by local ethic committees for reuse. Usually, these protocols include a clear statement about second uses of collected data. At the time of biopsy, the patient is informed, and his/her **consent** is taken. The patient is also informed on the potential reuse of its data.

The data are always **pseudo-anonymized** such that it is not possible to identify a patient without retrieving the identifiers. To map a biopsy to a patient name, one would need explicit access to the UZ Leuven electronic healthcare system dedicated to the medical practitioners, where every access is monitored and stored in logs.

11.4 AI in action

Selection of the appropriate machine learning model

Looking for potentially new phenotypes using data driven methods, amounts to group the data into different classes not known beforehand. This is exactly the goal of **unsupervised clustering** techniques.

In this use case, a general framework called **consensus clustering** (Strehl et al.) was used. Unlike traditional clustering which relies only on a single partition of the data, consensus clustering repeats the clustering process a large number of times (i.e., more than 500 times) using different samples of the original data, either with subsampling technique or bootstrap approach (Monti et al.) and different initial conditions (e.g., different random seed). It then combines those various partitions using a consensus function to get the final partition. Typical consensus functions include majority voting and k modes. Some more complex consensus functions rely on the construction of graphs. The consensus can also be reached by clustering the consensus matrix directly, i.e. the square matrix M where each entry (i, j) represents the level of agreement or similarity between data points i and j based on their clustering assignments across multiple clustering partitions. Any clustering algorithms that rely on a similarity matrix, such as spectral clustering, can be used. Overall, consensus clustering approaches are more robust to instability.

Consensus clustering is a general framework that can be used with many clustering algorithms. The core algorithm used was **k-means** (see Module 4), a well-known distance-based clustering algorithm. K-means create non-overlapping clusters by iteratively assigning data points to the nearest cluster centre and updating those centres based on the mean of the points assigned to each cluster until convergence criteria are met. The notion of nearest cluster depends heavily on a distance metric.

In this use case, we did not want to simply cluster the data. We wanted our clusters to be meaningful from a clinical perspective and not simply reflecting (un)-known histological patterns. To that end, a **semi-supervised clustering** approach was used, where additional proxy variables are used to guide the clustering process. Semi-supervised clustering acts therefore as a hybrid approach between fully supervised and fully unsupervised learning. The clinical outcome of interest we used as proxy variable was **graft failure**, so we developed a special weighting of the lesions scores based on their association with graft failure: lesions that were strongly associated with graft failure got a higher weight than lesions poorly associated with the clinical outcome. The weights were derived from individual Cox regression models, and we used a weighted Euclidean distance to implement this behaviour in our k-means algorithm.

Additionally, a weighted scheme was implemented such that the algorithm was not influenced by the overrepresentation of “protocol biopsies”, i.e. biopsies that are performed according to a pre-defined protocol and not due to a potential problem (“indication biopsies”). By doing so, the algorithm can “focus” more on inflamed biopsies instead of being driven by the large number of “normal” biopsies.

Training of the models (splitting, validation)

Given the low proportion of rejection cases compared to no rejection, the weighted Euclidian distance was derived on the whole training data to obtain confident association measures.

There was only one important parameter to train: k, the number of clusters. We used our derivation cohort (training dataset) to define k, based on the metrics described below. We used an external dataset from another transplantation centre to validate our findings. External validation remains the gold standard to validate this kind of models and assess it generalizability in external settings.

Hyperparameter tuning

In the field of clustering, the main parameter remains k, the number of clusters. Some clustering methods directly optimize this parameter internally, but most of the time it has to be defined by the user regarding the performance on a pre-defined metric. Of course, one should not blindly use automated approach to determine k.

The main metric used is called **proportion of ambiguous clustering (PAC)**, which was specifically developed for consensus clustering. To keep it simple, it summaries the proportion of pairs of datapoints that are systematically clustered together (or systematically clustered in separate clusters) across the whole set of different partitions. To get a stable clustering, you aim for the lowest PAC, i.e. a high proportion datapoints keep the same assignments irrespective of the initial parameters.

Additional secondary metrics were used to have a more comprehensive overview of the clusters performance:

- **Adjusted rand index (ARI)**: This metric computes the degree of similarity between two different partitions but also accounts for overlapping partitions due to chance. We used the existing Banff classification as baseline. An ARI of 1 means a perfect overlap between two partitions of the same data. An ARI of 0 means two completely independent partitions.
- **Log-rank test**: We used the log-rank test of the consecutive two best clusters to assess the relevance of the clusters to discriminate between two different survival trends. If two clusters have similar survival trajectories (log rank test non-significant), it is probably not useful to have two separate clusters.

Possibility of multimodality

Multimodality is possible using Multimodal clustering/ Multiview clustering. However, in this case we wanted the clusters to be solely based on the histology and not on additional clinical/labs variables. It is possible to integrate different sources of data at various steps of the clustering process (early integration, intermediate integration, late integration), similar to data integration in supervised learning.

11.5 Evaluating AI

Ground truth and evaluation of AI performance

Currently, the ground truth in the domain of (kidney) transplantation remains the pathologist diagnosis. However, in the field of clustering, the notion of ground truth is not always as clear as in supervised learning. The fact that clustering is mostly unsupervised, by definition, lim**its the use of external labels or ground truth**. We have to define other metrics to assess the performance of the algorithms.

Improvement labels by AI

The improvement of labels was indeed the goal of the project in this use case. Can data-driven methods lead to better classification, i.e. more meaningful labels than the current classification for kidney transplant biopsies?

Relevant metrics

Evaluating the result of clustering methods is not as straightforward as in a supervised setting, due to the absence of "ground truth". While some use dispersion or stability measure to assess the clusters validity, in the current use case, one clinically important metric was the **association of the clusters with graft failure**. In that regard, our work relied more on **traditional statistical metrics** (hazard ratios, C-index, restricted mean survival time, log rank test), rather than metrics encountered in machine learning projects such as (R)MSE, precision-recall, F1-score,…

What is the gain for the doctors?

One of the gains would be to have **more clinically useful phenotypes of rejection**, as the clusters are developed to be associated with graft failure. It can also serve as an **independent validation of the current classification system** of kidney transplant rejection: if a data-driven approach can recover the existing "human crafted" classification without any guidance, that would provide additional credibility to the current classification. In our work, we could only partly recover the existing Banff classification. The new phenotypes we found can also be used to define Banff-independent clinical endpoint for future research protocols.

Additionally, we also developed simple **activity and chronicity scoring and visualization** systems that allow to plot on intuitive both the phenotype trends and its severity (level of inflammation and chronic damages). We are now studying the clinical potential of these scores on the top of the existing Banff classification. These indices would act as continuous clinical outcomes and bring more nuance to the analyses as they reflect the continuum spectrum of the rejection process (unlike the current black/white classification system).

11.6 Challenges

Privacy of AI

The issue of privacy in AI is fundamentally different than it is elsewhere in computer science. AI is really just a bunch of models/methods used to perform specific tasks; it is agnostic to the concept of privacy. By definition, AI in healthcare works with highly sensitive data. The whole chain from collecting data in research to the final implementation of the model in clinical practice should take privacy of patients into account, for instance with technical safeguards.

In the current use case, we worked with **pseudo-anonymized data** to train the model. Once the model is trained, the prediction of it is based solely on the nearest cluster centroid. This means **no personal data is potentially stored in the model** (unlike complex neural network models for instance). The website that currently hosts the algorithm (rejectionclass.eu.pythonanywhere.com) complies with the GDPR and does not store any confidential data, nor does it require to provide identifiers to run the models, only abstract histological lesion scores.

Barriers to the patients with use of AI

In this specific use case, there is not any barrier that would directly impact the patients. The biopsy would still be taken and read by a human pathologist in the first place, following the protocol described earlier, so it does **not modify anything in the medical process**.

Quantity of data

Unlike other medical data modalities generated by (semi)-automated platforms (medical imaging, genomic data) which generate large amount of data on a daily basis, qualitative data on kidney transplant biopsies are rather **scarce**.

Kidney transplant biopsy is not a common procedure in general hospitals, and it requires a qualified pathologist to review the slides. In UZ Leuven, which is a reference centre in Belgium for kidney transplantation, around 700 biopsies are performed per year. In that regard, we are lucky to work with one of the largest curated datasets of kidney transplant biopsies in the world.

11.7 Future perspectives

Challenges in the future

The first challenge is practical and linked to data collection. How can we collect **sufficient data** to develop and validate new algorithms, especially regarding new innovative and costly technologies (scRNA for instance)? Given the scarcity of kidney transplant biopsies data, one solution could consist in securing **collaborations** with international teams to build large datasets.

Next, the future of medicine (not restricted to pathology) will rely more and more on **numerical assessment of objective data** (e.g. molecular biomarkers) rather than on the subjective judgment of clinicians. From a clinical perspective, it is convenient to work with discretized medical entities and put diseases into boxes. However, we now know that most of the medical conditions exist on continuous spectra, which renders such black and white thinking obsolete. This would require a paradigm shift from the clinicians from working with categorical entities to **continuous and probabilistic thinking**. This would involve re-defining the existing phenotypes using more data-driven approaches. From a clustering perspective, this could mean switching from hard clustering (where data points can only belong to a unique cluster) to **soft clustering** (where a datapoint can belong to several clusters at the same time reflected by its membership values)

Finally, one major challenge about AI is how to **educate people to use AI**. What can people expect from an AI model, how to evaluate its real performance, how (not) to trust its predictions. Building critical minds is a key step for an appropriate use of AI in the medical practice.

Can a model replace the doctors?

No, but AI will ultimately revolutionize the way medicine is practiced. As said earlier, in the (near) future, pathology will probably rely more heavily on objective and numerical assessment of the biopsies instead of simply relying on the pathologist review. We can imagine having additional data provided by the automated analyses of digitalized slides or of whole slide images with advanced molecular markers. We are not there yet, but there is no major technical obstacle to prevent it. Pathologists would then be **assisted by AI tools** to recognize and grade lesions, to generate automated reports and assist them in administrative tasks.

However, there will always be tricky cases and unforeseen situations that will make AI models struggle. Currently, AI models in healthcare remain highly task-specific, they do not see outside of their training field and do not have access to external information on the patients which renders complex decision making impossible for AI models only. In the future, daily interactions between AI tools and medical doctors are ineluctable, which is why it is highly important to educate medical doctors on using AI appropriately, to know its limitations and benefits, its biases, etc.

12 AI in action: Facial-based syndrome classification

12.1 Welcome to Module 12

Welcome to Module 12. This is one of several "AI in action" modules, where we shift our focus from theoretical concepts to practical use cases, allowing clinical and technical experts to share their insights and experiences with AI applications in the healthcare field.

Why the "AI in action" modules matter

These modules about real-life use cases are significant as they bridge the gap between theory and real-world impact. By delving into the insights of experts who have developed or utilized AI applications in healthcare, you will gain a nuanced understanding of the transformative potential of AI in the field. Each use case offers a unique perspective on the practical challenges, risks and opportunities that AI brings to healthcare.

This module is about facial-based syndrome classification. Discover how AI can be applied for aiding the diagnosis of genetic disorders using facial shape and appearance, through interview clips with an expert in this field.

Learning goals

- Experience the contribution and limitations of AI in real-world applications for the case study of facial-based syndrome classification.
- Clarify the significance of overseeing children's development and the function of clinical genetics in detecting genetic disorders.
- Relate the importance of 2D and 3D facial photography as an aspect of interest in the clinical workup leading to diagnosis.
- Relate syndrome diagnosis to a multi-class classification problem and discuss appropriate evaluation metrics.
- Describe broader implications and limitations in this use case.

12.2 Context and clinical impact

When children grow from infancy to toddlerhood, adolescence, and finally into adulthood, several developmental milestones are expected within specific timeframes. As a clinician, Zarah knows this very well, and she keeps a close eye on the development of the youngest in the family. Now at the age of two, little Jack is expected to have certain language and communication skills, motor skills, cognitive skills, social and emotional development, play and creativity and social interaction. However, during the latest visit to the paediatrician, some concerns were noted; little Jack is showing some delay in development. Every child is unique and might develop certain skills earlier or later than others, but one possible cause for this delay might reside in genetic factors. To exclude or confirm a genetic cause, little Jack is referred to the Department of Clinical Genetics at the hospital.

Clinical Genetics is a medical specialty that focuses on the diagnosis, management, and counselling of individuals and families who have or are at risk of having genetic disorders. Developmental delays reflect such an elevated risk, but other common reasons for an elevated risk of having a genetic disorder include intellectual disability, birth defects or congenital abnormalities, recurrent miscarriages or stillbirths, family history of genetic disorders, unexplained medical issues, etc. As a rough estimation, 6% of all birth defects have a (partial) genetic cause.

Note that our expert is researching the genetics of the human face and mentions that the face is an aspect of interest in the clinical workup leading to a diagnosis. This is not by coincidence, because the way our face develops, follows a biological plan encoded in our genes. Therefore, our face provides valuable clues about our genetics. When there are atypical differences in facial appearance due to genetics, it is called **facial dysmorphism**. In 30 to 40% of genetic disorders a form of facial dysmorphism is noted, going from obvious and easy to recognize atypical facial characteristics, to a combination of more subtle facial anomalies.

As an example, take a look at the following two famous individuals with a genetic disorder, one of which the facial dysmorphism is easy to recognise and the other not so much.

Do you know which genetic disorders affect these famous persons?

- Peter Dinklage has achondroplasia. **Achondroplasia**, which is a genetic disorder that affects bone growth, resulting in dwarfism. Peter Dinklage, the actor known for his role in "Game of Thrones" and other projects, has openly discussed having achondroplasia. He has become an advocate for diversity and representation in the entertainment industry and has been praised for his talent and contributions to acting.
- Abraham Lincoln, the 16th President of the United States, is believed to have had a condition called **Marfan syndrome**, which is a genetic disorder that can affect various parts of the body, including the skeletal and cardiovascular systems. While there is no definitive historical evidence confirming his diagnosis, some historical accounts and medical analyses have suggested that Lincoln might have exhibited features consistent with Marfan syndrome.

Let’s think together where AI can be useful in this use case. First, given the added values of AI in healthcare in Module 1, in which of seven **application domains in healthcare** (Diagnosis & Prognosis, Treatment & Therapy, Screening and Prevention, Evidence & Personalized Medicine, Organization & Logistics, Assistance & Interaction, Education & Training) do you think AI can make an impact?

12.3 Data

Facial appearance is the main data source in this use case. Traditionally, the face can be imaged using standard **2-dimensional (2D)** photography, which we all know from using smart-phones these days. However, in clinical settings we also have access to **3-dimensional (3D)** photography.

2D photographs are created through perspective projection, where light passes through a lens and onto an imaging plane. When digitally captured or digitized afterward, these photographs are recorded as a grid of square or rectangular pixels, each assigned a specific colour value. Typically, the colour of each pixel is defined by three values, which represent the colour in a standard system like RGB (Red Green Blue). Grayscale photographs, on the other hand, are depicted using a single intensity value per pixel, going from back, over grey to white.
3D photographs comprise 3D point coordinates sampling the facial surface densely. Usually this 'point cloud' is augmented with information about how the points are interconnected. Thus, a 3D photograph is usually an irregular polygon comprising triangular or quadrilateral faces that together define the surface. Such a representation is called a 'mesh'

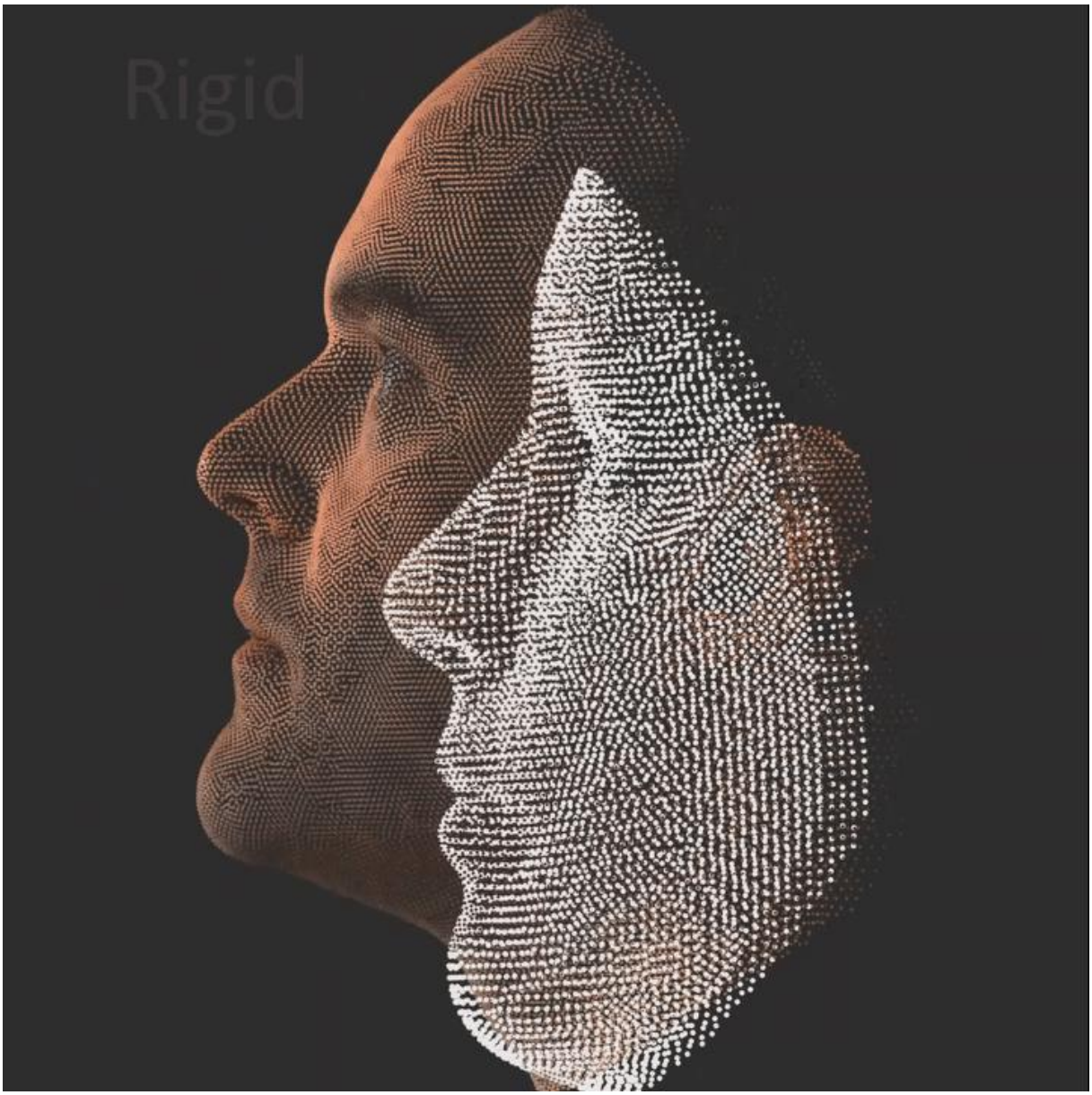
The primary advantage of 3D photography in comparison to its 2D counterpart lies in the preservation of facial shape information, which inherently exists as a 3-dimensional structure. Consequently, the data obtained from a single face is more comprehensive when utilizing 3-dimensional photography. However, the acquisition of 2-dimensional photographs is simpler due to the prevalence of affordable cameras embedded in smartphones. Consequently, commencing from either imaging dimensionality results in two distinct paths: **data-centric AI and model-centric AI.**

Counterintuitively, 3D images are less complex than 2D images. The 3D facial shape induces indirect variations (such as shadows) in a 2D photograph, while it is directly quantified from a 3D image. Although obtaining 2D photographs is easier, 3D images are less susceptible to factors like camera angles, focal depth, and lighting variations. Therefore, the full variability of 2D colour images is high, and the variation of projections of 3D structures onto the 2D plane is complicated. Therefore, AI models trained on 2D images aim to simulate the three-dimensional aspects of facial features based on learned patterns from the highly variable 2D representations. However, achieving this requires more data examples for the model to learn from and a more complex model architecture to handle it. Consequently, learning from 2D images typically follows a model-centric approach. To give you an idea, facial recognition networks trained by Facebook and Google, were trained on millions of facial images uploaded by users, like you and me, on their platform. In contrast, due to the higher data quality and easier normalization process for 3D images, learning from 3D images typically aligns with a data-centric approach.

Data-centric AI is capable of learning from less data in cases where it is difficult to do so due to the rarity of genetic disorders. Some genetic disorders are relatively common and affect a significant portion of the population (e.g., Down Syndrome affects 1 in every 700 births). On the other hand, there are numerous rare genetic disorders, often referred to as "orphan diseases", (e.g., Hutchinson-Gilford progeria syndrome, characterized by rapid ageing affecting 1 in 20 million individuals) that individually affect a very small number of people. These rare disorders can be challenging to study and diagnose due to the limited availability of patients for research and clinical trials. This is where data-centric AI approaches can be valuable. By better leveraging available data, even from a limited number of cases, AI can help identify patterns and correlations that might not be apparent through traditional methods. This can lead to improved diagnosis, treatment, and understanding of these rare genetic conditions.

As noted before, in data-centric AI, the emphasis is on using **high-quality data** that is well-organized and pre-processed, **eliminating spurious variations** in the data that are not related to or important for the task at hand. Besides the inherent advantage of 3D photography in preserving 3D facial shape, as compared to 2D photography where 3D facial shape is only indirectly coded, both imaging techniques can still improve the quality of the data through data normalization.

Proper **data normalization** removes spurious variations in the data that are not relevant to the task at hand. For instance, the pose of the face is irrelevant for the analysis of facial dysmorphism. In using 2D photography, one can work with standardized frontal and lateral portrait images. In using 3D photography, one can work with a fixed 3D facial template of 3D points consistently indicated across all 3D facial surface images. This is achieved by using techniques known as **3D rigid and non-rigid surface registration**, as illustrated in the animation below. A fixed template (white 3D point cloud) is first rigidly aligned with a target 3D facial image by shifting, rotating and rescaling it. Subsequently, the template is locally deformed step by step until it fits onto the shape of the target 3D facial image. When this is done for a large database of 3D facial images, all faces are now represented using the same amount of 3D points. Once this is done, all faces are also moved into the same pose.



12.4 AI in action

A broad spectrum of facial dysmorphism exists, ranging from **easily recognizable to difficult to discern**. Our expert, Michiel Vanneste, along with his colleagues, classifies genetic disorders into three distinct categories: A, B, and C.

- **Category A**: These are conditions that are clinically recognizable using facial features, meaning the features are known to be typical for the clinical diagnosis, and the condition is genetically homogeneous. I.e. the manifestation of facial features is consistent from one patient to another in the same group.
- **Category B**: These are conditions that are clinically recognizable using facial features, meaning the features are known to be typical for the clinical diagnosis. However, in contrast to Category A, the condition is genetically heterogeneous. I.e. the manifestation of facial features can vary from one patient to another in the same group.
- **Category C**: These are conditions that cannot be recognized from facial features; therefore, facial characteristics do not provide typical clues in clinical diagnosis.

Since the facial examples used in the exercise above are facial averages, they cannot illustrate the difference between categories A and B, but they do illustrate the difference for both categories A and B, with C.

The table below describes the database of 3D facial images that was used by Michiel Vanneste to train an AI model in support of diagnosis of genetic disorders.

- Examples of **category A** include Achondroplasia (Peter Dinklage), Williams, Down etc.
- Examples of **category B** include Noonan, Treacher Collins, Coffin Siris, etc.
- Examples of **category C** include Marfan (Abraham Lincoln), Russel Silver, Loeys-Dietz etc.

In total the dataset comprises **3285 images** of 51 different syndromes and one group of 138 controls.

TABLE I: Data demographics: Syndrome group name, sample size (N), mean and standard deviation of age (M±SD), the female/male ratio (F/M), and the category .

Name	Size	Age Range	Sex Ratio	Category	Name	Size	Age Range	Sex Ratio	Category
Williams	221	17.57 ± 13.9	0.46	A	BBS	87	26.33 ± 14.78	0.48	C
22q11_2 Del	180	10.74 ± 6.03	0.49	A	Neurofibromatosis	85	20.18 ± 18.01	0.54	C
Wolf Hirschhorn	155	11.03 ± 9.42	0.57	A	Loeys Dietz	84	25.38 ± 17.15	0.57	C
Smith Magenis	129	14.32 ± 9.09	0.55	A	Joubert	75	10.57 ± 8.58	0.48	C
Down	117	21.64 ± 11.14	0.49	A	Ectodermal Dysplasia	71	15.09 ± 15.32	0.28	C
Prader Willi	96	19.34 ± 13.24	0.51	A	Rett	70	13.32 ± 10.54	0.89	C
Fragile X	77	17.65 ± 12.56	0.3	A	Cardiofaciocutaneous	59	12.21 ± 8.55	0.53	C
Achondroplasia	70	22.62 ± 18.34	0.59	A	Klinefelter	57	22.91 ± 14.58	0	C
Rubinstein Taybi	63	13.54 ± 11.73	0.52	A	Mucopolysaccharidosis	57	21.51 ± 13.51	0.47	C
Costello	58	12.39 ± 9.24	0.66	A	Alstrom	52	21.28 ± 9.4	0.54	C
Cohen	33	18.27 ± 10.46	0.52	A	Fibrodysplasia Ossificans Progressiva	50	21.81 ± 12.37	0.56	C
Pitt Hopkins	29	8.53 ± 5.7	0.62	A	Fabry	48	32.37 ± 16.53	0.44	C
Pallister Killian	23	9.59 ± 7.19	0.26	A	Sotos	45	17.92 ± 12.22	0.49	C
Crouzon	22	10.22 ± 6.27	0.55	A	Russell Silver	44	10.18 ± 10.32	0.34	C
Smith Lemli Opitz	19	11.75 ± 7.05	0.32	A	Cockayne	41	12.15 ± 7.37	0.44	C
Apert	13	14.55 ± 10.73	0.62	A	Pseudoachondroplasia	35	28.06 ± 20.53	0.51	C
Coffin Lowry	12	13.76 ± 9.16	0.08	A	Osteogenesis Imperfecta	31	16.72 ± 14.61	0.68	C
Cornelia de Lange	183	12.1 ± 9.14	0.54	B	1p36 Del	29	8.82 ± 7.83	0.62	C
Noonan	155	14.06 ± 12.51	0.45	B	Trisomy 18	27	8.79 ± 8.71	0.85	C
Angelman	106	9.97 ± 7.57	0.47	B	Beckwith Wiedemann	26	9.68 ± 6.66	0.42	C
Stickler	45	22.31 ± 17.45	0.62	B	EED CLP	20	23.24 ± 17.71	0.65	C
Treacher Collins	39	18.48 ± 13.5	0.49	B	Vander Woude	16	10.16 ± 4.75	0.56	C
Kabuki	37	12.09 ± 6.62	0.65	B	Goltz	14	9.4 ± 5.03	0.86	C
Coffin Siris	16	12.08 ± 9.65	0.63	B	Rhizo Chondro Punct	13	7.57 ± 5.53	0.69	C
Marfan	153	26.34 ± 16.85	0.58	C	Zellweger Syndrome	11	7.33 ± 9.49	0.09	C
Turner	102	24.25 ± 19.03	0.98	C	Controls	100	30.94 ± 11.64	0.72	CONTROL

12.5 Evaluating AI

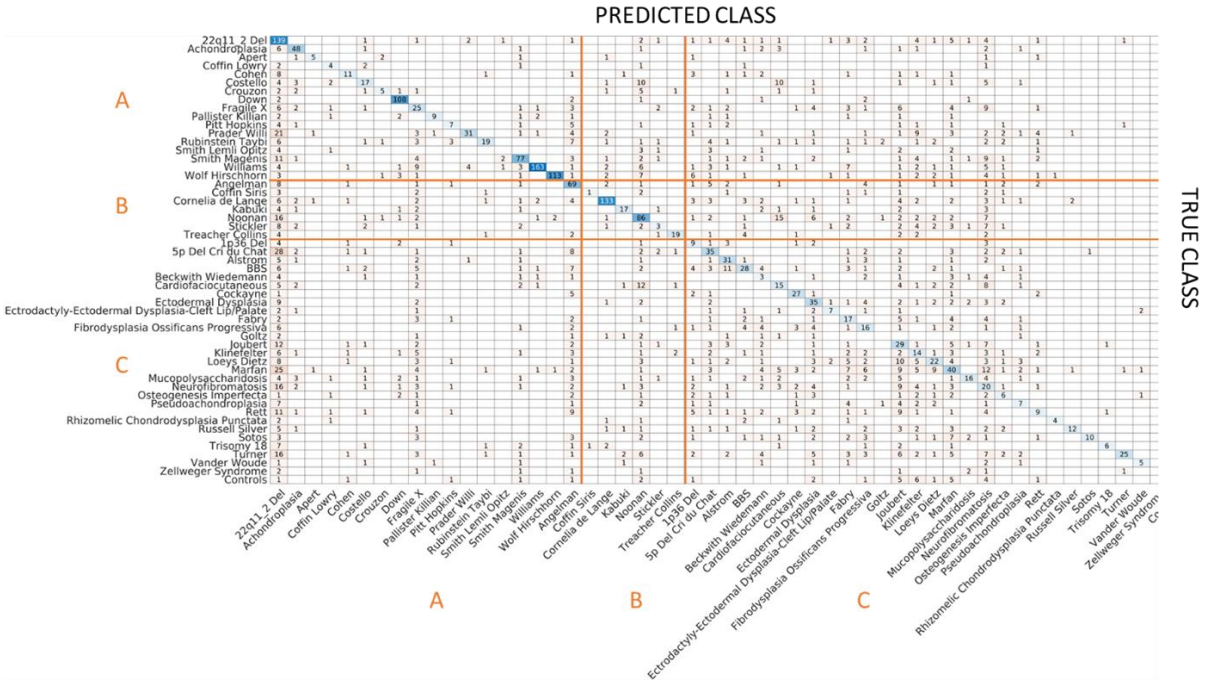
Diagnosis can be seen as a classification task, where the goal is to assign an individual to one of two classes: "has the disease" or "does not have the disease". This is the most common situation. However,

there are also cases where diagnosis can be more complex, such as when there are multiple possible diseases or when the disease is present in varying degrees.

In this use case, complexity arises from the fact that we are dealing with a **multi-class classification problem** where individuals can be classified into multiple possible genetic disorders. The ground truth comprises the clinically and molecularly confirmed diagnoses of patients in the database, serving as the labels for prediction and testing.

Our expert Michiel employed a 5-fold cross-validation scheme for evaluation. 5-fold cross-validation involves dividing the data into 5 non-overlapping subsets or folds. Each fold is then used once as a validation while the remaining folds form the training set, resulting in a total of 5 evaluation rounds. In a single evaluation round 657 data points are used in the test set.

For each evaluation round, we assess classification performance, beginning with a **confusion matrix**, as illustrated in the figure below. It resembles the matrix used in a two-class classification problem, but with a size of 52 by 52, instead of 2 by 2. This size is determined by the number of syndromes groups (51 different syndromes) plus one group for the general population.



In a **two-class classification** problem, each test case yields either a correct prediction (matching the positive class for positive examples or the negative class for negative examples) or an incorrect prediction (mismatching classes).

- **Correct predictions** appear on the diagonal of the confusion matrix (shaded blue colours).
- **Incorrect predictions** are found in the off-diagonal cells (shaded red colours).

In multi-class classification, a vector of classification scores is generated to rank syndromes from most to least likely. Strictly speaking, a prediction is considered correct only if the correct syndrome class ranks first in the sorted list of scores. This is known as a **top-1 correct prediction**. Otherwise, the prediction is deemed incorrect. This again leads to correct predictions appearing on the diagonal of the confusion matrix (shaded blue colours), while incorrect predictions are found in the off-diagonal cells (shaded red colours). Given the large confusion matrix above, it is visually clear that it is not easy to provide an overall statement of performance. For many syndromes we see a good amount of correctly

classified individuals on the diagonal, especially for the syndromes in category A and B, and less so for those in category C. However, for each syndrome irrespective the category we also observe some misclassifications. Therefore, it is clear that the system is not 100% perfect.

In **multi-class classification**, there is room to take a more lenient approach to performance evaluation. Given that facial analysis in syndrome diagnostics serves as an initial clue in the clinical work-up, its primary goal is to help clinicians narrow down the pool of possible syndromes for further investigation. In practical terms, clinicians may be interested in identifying the first 3 or even the first 10 most likely syndromes among the sorted classification scores. If the correct syndrome appears within the first X sorted classification scores, it is considered a **top-X correct prediction**.

Now for each of the syndromes it is of interest to report on multiple top-X performances, as done in the three figures below. Specifically, for categories A, B, and C, we calculated sensitivity using the formula:

sensitivity = true positives / (true positives + false negatives).

A true positive is registered when the correct syndrome label appears in the top-1 (most stringent), top-3 (moderately stringent), or top-10 (least stringent) of the predicted syndrome labels; otherwise, it is considered a false negative. Notably, the sensitivity increases as the definition of true positive becomes less strict.

12.6 Challenges

Michiel touched upon the following aspects of coexistence:

- Technical robustness and safety
- Transparency and explainability
- Fairness and bias

12.7 Future perspectives

In his future perspective, Michiel explicitly mentions an already available application that aids clinical diagnosis using facial features. This **application** is called Face2Gene, developed by FDNA. Face2Gene uses 2D facial images to generate a sorted list of likely syndromes when a patient's face is imaged. It is the first of its kind, and future developments are expected to further integrate facial-based diagnostics into clinical genetics. Its initial success is attributed to its accessibility to clinicians through smartphone integration. Looking ahead, as 3D imaging technology in smartphones continues to advance, these devices will also be capable of capturing and processing 3D images and depth information.

Michiel also anticipates the future expansion of AI in the clinical workflow of clinical genetics, progressing from facial imaging to variant prioritization and, ultimately, automated reporting.

Variant prioritization involves identifying and ranking genetic variants based on their potential significance or relevance to a patient's condition. I.e., different variants obtain a risk score. Machine learning models can be trained to prioritize variants based on factors like known disease associations, evolutionary conservation, and predicted functional impact. This process helps prioritize which variants should be further investigated or considered for potential clinical significance.

AI can also be used to **automate the generation of detailed reports** summarizing the findings from genetic testing. In other words, they can convert complex genetic data into understandable reports for clinicians and patients. These reports can include variant interpretations, potential clinical implications, and recommended actions.

13 AI in action: Quality support in colonoscopy

13.1 Welcome to Module 13

Welcome to Module 13. This is one of several "AI in action" modules, where we shift our focus from theoretical concepts to practical use cases, allowing clinical and technical experts to share their insights and experiences with AI applications in the healthcare field.

Why the "AI in action" modules matter

These modules about real-life use cases are significant as they bridge the gap between theory and real-world impact. By delving into the insights of experts who have developed or utilized AI applications in healthcare, you will gain a nuanced understanding of the transformative potential of AI in the field. Each use case offers a unique perspective on the practical challenges, risks and opportunities that AI brings to healthcare.

This module is about quality support in colonoscopy. Discover how AI can be applied in colonoscopy and the screening for colorectal polyps, through interview clips with clinical and technical experts in this field.

Learning goals

- Experience the contribution and limitations of AI in real-world applications for the case study of quality support in colonoscopy.
- Relate the use of colonoscopy as screening tool for colorectal cancer.
- Outline the data from a colonoscopy as input to an AI system and explain the use of recurrent neural networks with video as sequence data.
- Explain polyp detection and segmentation, and their dependency on high level and low level image features, respectively.
- Discuss how patient well-being can be prioritized during a clinical trial in follow-up to cross validation.
- Describe broader implications and limitations in this use case.

13.2 Context and clinical impact

Zarah is approaching her **50th birthday** next month, and she eagerly anticipates celebrating this significant milestone with her entire family. In Flanders, Belgium, reaching the age of 50 also prompts the tradition of sending a personal invitation letter. However, this letter is not a typical birthday congratulation; rather, it extends an invitation for Zarah to undergo an **iFOB-test**. Pardon me, what kind of test is that?

The **iFOB test** stands for **immunochemical faecal occult blood test**. Through this test, one can detect colorectal polyps or tumours at an early stage by identifying very small amounts of blood in the stool. In Flanders, a population-wide program is implemented, reaching out to men and women aged 50 to 74 to perform a stool test every two years. Individuals with an abnormal iFOB test result (indicating the presence of blood in the examined stool) are referred for a visual colon examination or colonoscopy. Colorectal cancer develops slowly. It takes an average of 10 years for a polyp (a protrusion in the wall of the colon) to become a malignant tumour. Using the iFOB test allows colorectal cancer to be detected early in many cases, often before it actually becomes cancer.

Zarah comprehends the significance of the test, essentially emphasizing its crucial role in one of the seven application domains in healthcare from Module 1, namely Screening & Prevention.

The iFOB test is relatively straightforward and cost-effective to undergo, so Zarah proceeds with it. She collects a sample, and sends it to a laboratory in a prepaid envelope. After 14 days, Zarah receives the test results, revealing an abnormality with more than a certain amount of blood in the stool. While this does not necessarily indicate cancer, just to be on the safe side, she is referred to a specialist for a visual examination of the colon, known as a **colonoscopy**.

Let's introduce our clinical expert for this use case: Pieter Sinonquel.

Pieter is one of the experts in Flanders performing colonoscopies. This is a medical procedure in which a flexible tube with a camera at its tip (**colonoscope**) is inserted into the rectum to examine the inner lining of the large intestine (colon) and rectum. This camera generates a video acquisition of the inner lining of the colon. It is commonly performed for the diagnosis and screening of colorectal conditions such as polyps, tumours and inflammation. During a colonoscopy, the physician can also perform biopsies and remove any abnormal growths detected. The procedure is crucial for **early detection and prevention of colorectal cancer** after a positive iFOB test.

The medical procedure of colonoscopy follows clear guidelines aimed at detecting polyps before they become malignant and lead to colorectal cancer. However, the procedure is not infallible, and there is a possibility that some **polyps may be overlooked**. Various factors contribute to this, including blind spots during the complete colon screening, inadequate colon preparation, subtle and very small polyps, the extensive area to cover and inspect, and the variable focus of the medical expert.

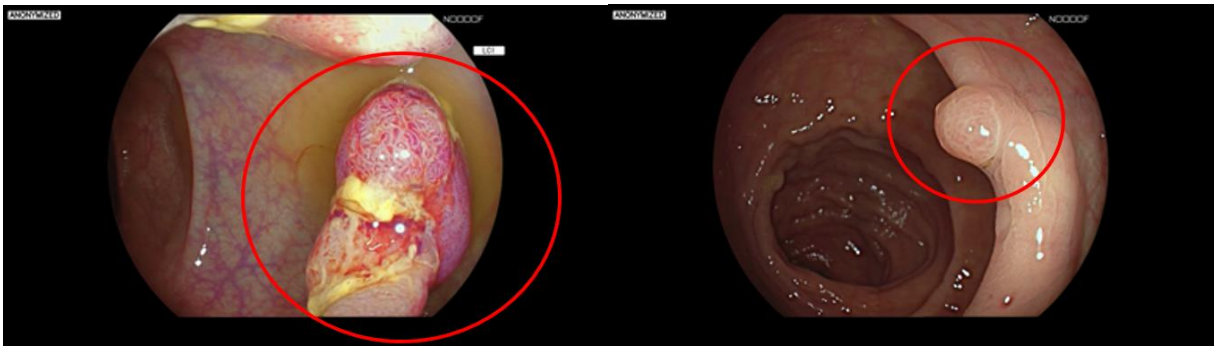
Considering these challenges, the question arises: **How can AI potentially assist in colonoscopy?** Let's consult our expert to gather some insights on this matter.

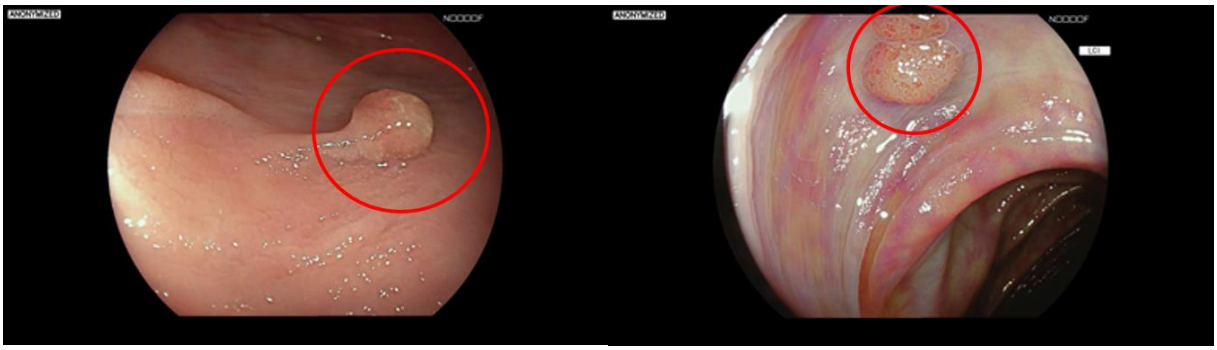
It is evident that our clinical expert supports the use of AI, anticipating positive impacts on colorectal cancer screening. Now, let's introduce our technical expert to share his insights on the potential role of AI in this context.

In summary, both experts share the goal of utilizing AI in colorectal cancer screening to enhance the detection of polyps throughout the colon during endoscopic examinations. They view AI as a valuable complement to existing clinical practices. In the upcoming sections, we will delve into the data involved, the functioning of the AI system implemented by Tom, and the methods for assessing its added value in clinical practice.

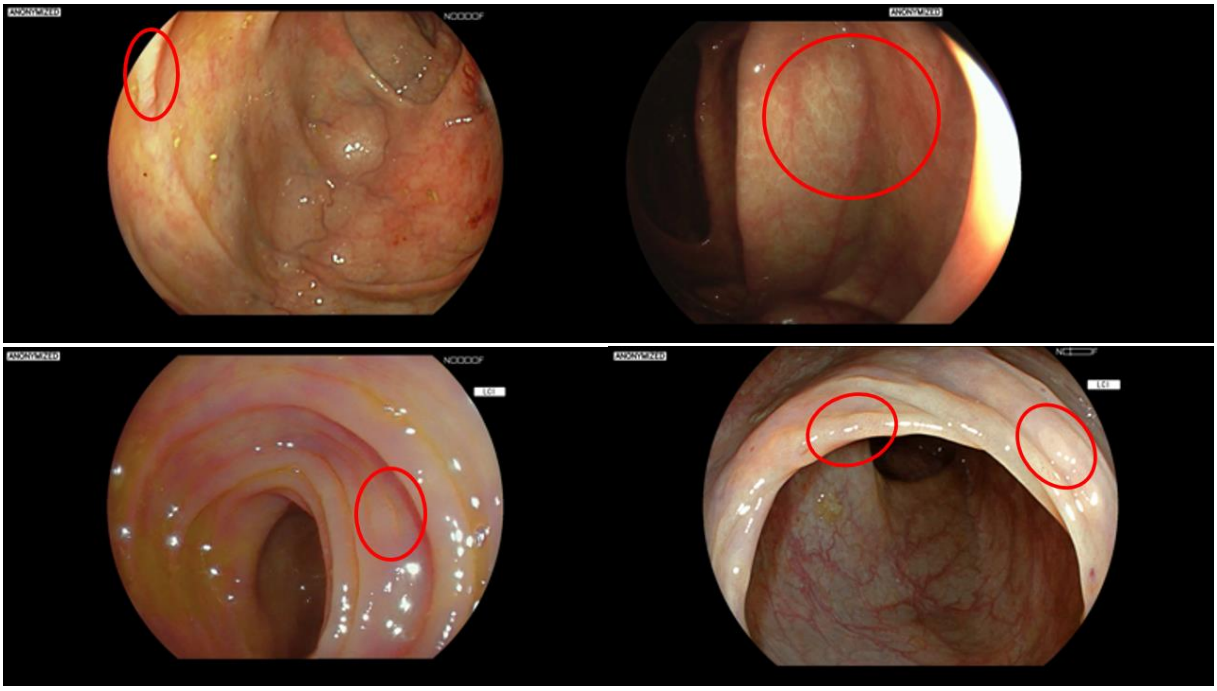
13.3 Data

Let’s first explore the challenge of identifying polyps from endoscopic recordings. Given the five examples below, can you see the polyps?





Let's be honest, the level of expertise required so far was limited. But what about the following five cases: can you identify the polyps again, knowing that there is at least one in each example?



These examples highlight the necessity for expertise in identifying all polyps. This naturally prompts the question of how AI could offer assistance. To delve into this, we need to understand how AI can be beneficial, and what **data** is essential for training and supporting an AI system.

In essence, the key data source for learning is derived from **video segments** containing **manually labelled** polyp information. This encompasses identifying the specific frames in the video where the polyp is visible, along with outlining or segmenting the polyp in those frames. As Pieter elucidated, this process is time-intensive, yet they successfully annotated over 3000 polyps in their available dataset. However, based on this initial annotated dataset, Tom manages to facilitate a **semi-automated annotation of more data**, such that together there is enough data available to train an AI system.

Video as sequence data: Sequence data refers to a type of data where the order of elements matters. It is characterized by a specific arrangement or sequence of individual elements. Each element in the sequence has a position or index, and the relationships between elements are defined by their order.

Video data is a prime example of sequence data. **In a video, each frame follows the previous one in a specific order, creating a sequence of images**. The frames are presented rapidly to give the illusion of motion. Each frame contributes to the overall understanding of the video, and the temporal order of frames is crucial for interpreting the content accurately.

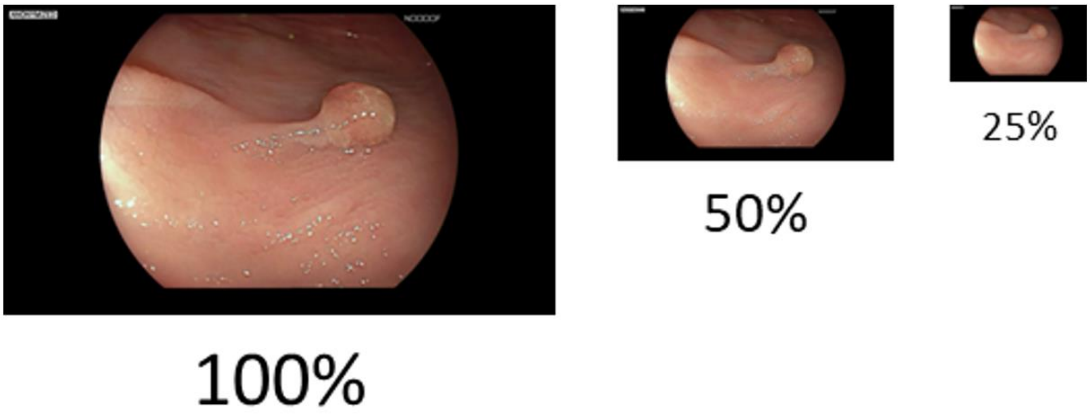
Data ethics approval

The gathering of data by Pieter and Tom has received approval from an ethics committee. An essential aspect in this approval is that every patient provided **informed consent**, granting permission for the utilization of recorded data for scientific purposes. In addition, all data has undergone **anonymization**, ensuring privacy, and only pertinent metadata, including demographic details and pathological evaluation results, are retained for the development and validation of the AI system.

Data preprocessing and harmonization

The endoscope recordings used to train the AI system, originate from various vendors and endoscope models, resulting in notable **variations** in video quality, resolution, frame rate, format, and more.

One requirement is that all the data has the same **fixed resolution**, which is a typical requirement when working with e.g., convolutional neural networks as the most powerful image processing tool today. In consideration that video data is memory intensive, this resolution cannot be too high given today's hardware limitations (GPUs with quite some operational memory, known as RAM, are best used in this use case).



In the figure above, you can see the effect of changing the resolution of one video frame or image. The original resolution is downscaled from 100% to 50% and 25%. Choosing the right resolution is a matter of trading off computational efficiency (speed and memory) versus level of detail.

Taking the level of detail and computational efficiency into account, Tom is resizing all input videos to a **resolution** of 384x288 pixels per video frame. Using some empirical experimentation, it turns out that this resolution is small enough for fast computation, but large enough to show polyps with enough detail for detection.

Aside from the resolution, altering the **quality of the recordings** from a range of different devices post-capture is challenging, and a true harmonization of the data is therefore not possible. Therefore, instead of further harmonizing the data, the model is to be complex enough to handle the variability seen in the data due to different vendors and endoscopes. As long as the **size and diversity of the training data** is reflective of the natural diversity encountered in clinical practice, a model-centric AI system will accommodate these quality differences effectively. Therefore, once trained, it should generalize well on new unseen data, from different vendors and endoscope models.

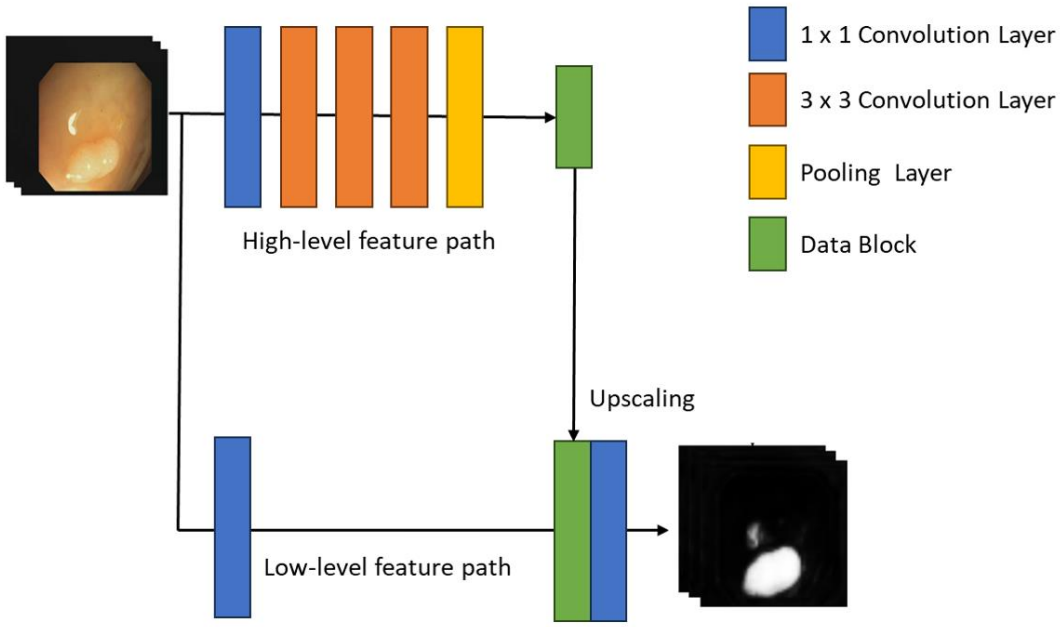
13.4 AI in action

Inside the AI system

Based on Tom's explanation, it is clear that video introduces **temporal context** to the complexity of detecting and delineating polyps. In simpler terms, if a polyp is visible in one video frame, it is likely also visible in the frames immediately preceding and following it. However, incorporating this temporal information into the system and ensuring robust learning of such dependencies is a non-trivial task. Therefore, Tom's AI system is constructed in two stages. The **initial stage** overlooks temporal dependencies, treating each video frame independently (like a static image) to address the polyp segmentation task. The **subsequent stage** leverages the ability to segment a polyp at the individual frame level, and then integrates temporal context to enhance segmentations consistently across various video frames. This integration ensures that the segmentation in one frame aligns with the segmentation in the adjacent frames. The assumption is that in doing so the final detection and segmentation rate becomes better.

Stage 1: polyp detection using segmentation

Visual object detection involves identifying specific objects in an image by utilizing high-level, abstract features to recognize object classes, while **segmentation** entails delineating object boundaries using low-level features like image edges and boundaries to distinguish the object as foreground from the background. Despite their close relation, these image-based tasks differ in the type of information required to accomplish their respective goals. In contemporary deep learning, the concept of multitasking, addressing multiple related problems simultaneously, is widely embraced as it demonstrates improved performance compared to attempting to solve these tasks independently. Hence, Tom suggests employing a network with dual feature extraction paths: the first prioritizing **high-level features** through a combination of multiple convolutional and pooling layers, akin to encoding tasks such as those in an auto-encoder; the second concentrating on **low-level features** with a solitary convolutional layer, as illustrated in the schematic overview of the network below:

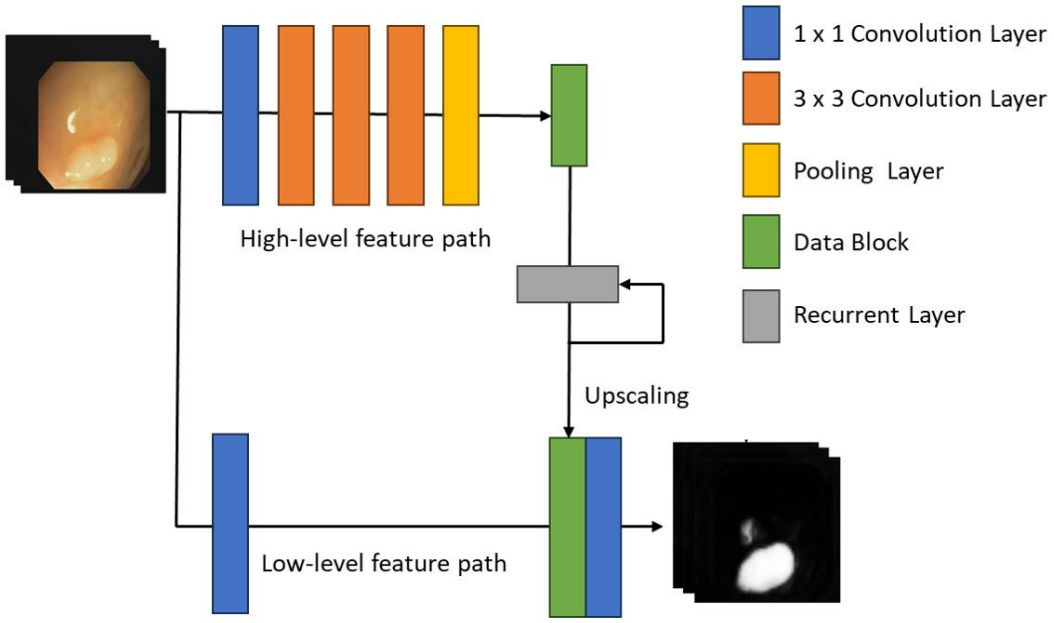


The network takes a video frame as input and generates a grayscale image as output, where each pixel's grey value represents the likelihood of that pixel being part of a polyp. Pixels inside the polyp area are

rendered white, while the background or non-polyp regions appear black, resulting in entirely black frames for video segments without any polyps. Such a network output, provides both polyp detection and segmentation feedback.

Stage 2: incorporating sequence dependencies

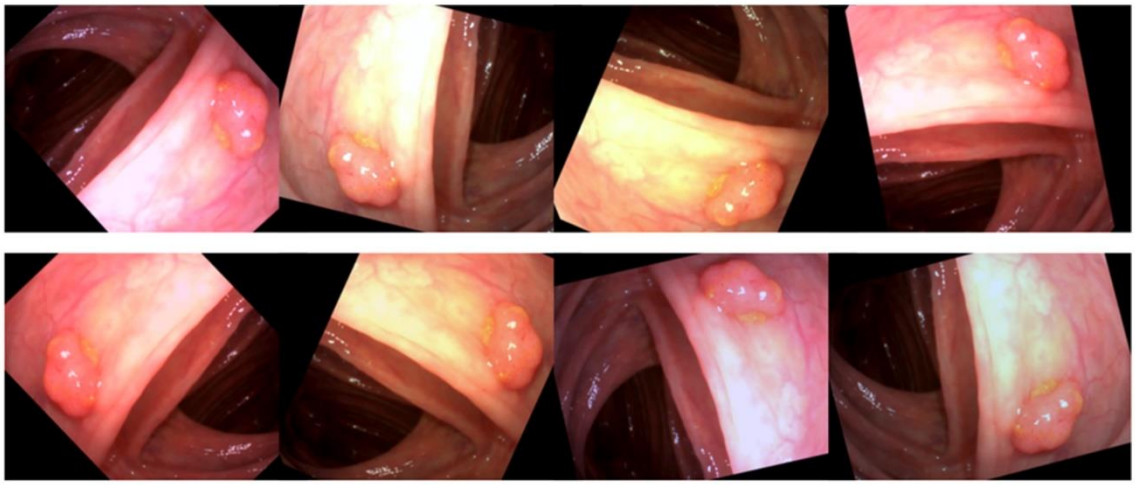
When accounting for temporal information and analysing multiple video frames, it is anticipated that high-level features, such as the presence of a polyp, change less rapidly over time. Conversely, low-level features, like the location of the polyp, exhibit significant changes between frames. Thus, the low-level feature extraction path remains unchanged, given the challenge of capturing rapidly changing information. To address temporal dependencies and maintain consistency across video frames, a **recurrent layer** is only introduced in the high-level feature extraction path. The recurrent layer operates like a memory and is therefore able to take the knowledge (e.g. there is a polyp in this video segment) from one frame as input to next few frames. This adaptation results in the following modification of the aforementioned network:



Training the AI system

When training CNNs, as opposed to more traditional 2D image analysis techniques, we want to **maximize the variability** in the images used for training. Changes in illumination, alignment, contrast, etc. are all beneficial to create a robust model that can handle unseen data when applied in the real world. In order to further enhance the variability, intensive **data augmentation** is applied: random horizontal and vertical flipping, rotation with a random angle in the range of 0 to 90 degrees, blurring with a gaussian kernel and random sigma in the range 0 to 0.1, brightness disturbance by multiplying all the pixel values in a single image with a random value in the range 0.9 to 1.1, and colour disturbance by multiplying all the channel values in a single image with a random value in the range 0.9 to 1.1. Most of these operations are standard image processing techniques that can be applied using programs like Photoshop.

To give you an idea of what this data augmentation looks like, have a look at the following eight different random augmentations, starting from the same input image:



The **network training** occurs in three sequential steps.

1. Initially, the network, devoid of temporal dependencies, undergoes pretraining on a vast and general dataset for object detection and segmentation in images—specifically, the PASCAL VOC 2012 challenge dataset.
2. Subsequently, through transfer learning, the pretrained network is finetuned on the available polyp dataset, gradually enabling more layers as trainable, starting from the final layer.
3. In the final step, the network is expanded with a recurrent layer, and training continues. Notably, training in this third step exclusively concentrates on the recurrent layer, leaving the other layers, previously trained and finetuned, unaltered.

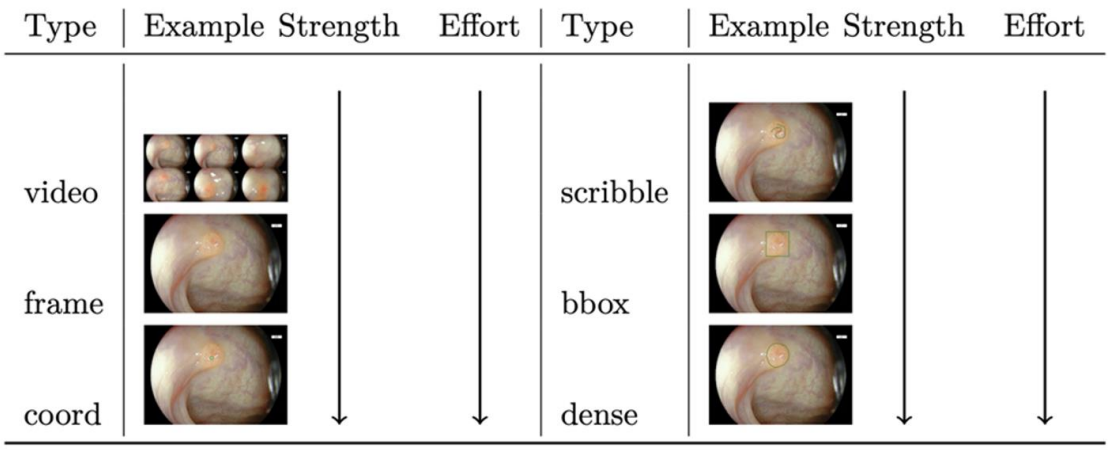
While the training process may seem straightforward, the stepwise approach is crucial, particularly due to the notorious difficulty in training recurrent networks. Recurrent networks often face challenges, such as steep learning curves or difficulty in converging to sensible solutions, making the incremental training steps a strategic necessity for successful implementation.

The **PASCAL Visual Object Classes (VOC)** 2012 dataset encompasses 20 object categories, including vehicles, household items, animals, and others: aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, TV/monitor, bird, cat, cow, dog, horse, sheep, and person. Each image in this dataset is annotated with pixel-level segmentation, bounding box, and object class information. Widely utilized as a benchmark for object detection, semantic segmentation, and classification tasks, the PASCAL VOC dataset is divided into three subsets: 1,464 images for training, 1,449 images for validation, and a private testing set.

During training, the **dice-based loss function**, commonly applied in image segmentation tasks, evaluates the disparity between predicted and ground truth segmentations. It quantifies dissimilarity by measuring twice the intersection of the regions divided by their sum. A dice score of 1 signifies perfect overlap and therefore a flawless segmentation result, while a score of 0 indicates no overlap and therefore a completely wrong segmentation result. The network aims to maximize the dice score, pushing it close to 1 for the training examples with provided ground truth segmentations. This supervised learning approach relies on labelled data, specifically ground truth segmentations, to update the network during training.

The **labelled data** needs to provide a notion of the presence and/or location of the polyps in recorded videos. Several types of labels could be used to achieve this goal: video/frame level labels, a coordinate pair for a point inside the polyp boundary, a scribble or set of points inside the polyp boundary, a

bounding box enclosing the polyp or a dense delineation of the polyp boundary. For each of these types, there is a trade-off between the strength of the label and the amount of effort it takes to acquire them. The strength of the label is determined by how much information it contains about the location of the polyp. For example, a dense delineation of the polyp boundary is a very strong label since the location of the polyp is fully and unequivocally defined. A video-level label, however, has an inherent amount of uncertainty since the polyp can be present in any number of frames and at a different location in each of those frames (but that information is not given). An example of each of these label types and their relative strength and uncertainty is shown here:



Generating **dense delineations** for polyps, although more time-consuming than bounding box (bbox) or scribble labels due to the manual annotation of numerous points, is preferred for its effectiveness in training the neural networks involved. In the initial stages of the project, collaboration between the clinical and technical experts (Pieter and Tom) respectively, focused on collecting dense delineations from hours of pre-recorded video footage. To optimize the strength-to-effort ratio, Tom introduced a **semi-automatic technique**. This method infers dense delineation masks for entire polyp videos based on just two to three manually delineated frames. This approach ensures robust labels while minimizing the annotation workload for expert clinicians.

The crucial **hyperparameter** to fine-tune for the recurrent layer is the number of timesteps used during training (i.e. how long should the recurrent layer memorize). This parameter dictates the scope of temporal dependencies the model can capture and significantly impacts training behaviour. Hyperparameter analyses were conducted on a validation set, constituting approximately 10% of the complete dataset. This validation set played a key role in determining the optimal timestep within a tested range (ranging from 1 to 5 frames) for effective model performance.

13.5 Evaluating AI

The choice for a clinical trial is not a random choice, but rather an important one. Tom emphasizes that while traditional machine learning evaluation protocols (e.g., cross-validation) using in-house recorded video data, can be employed to compare different versions or developments of the AI system, the fundamental question revolves around whether the system truly enhances the clinician's performance during colonoscopy. Answering this question necessitates a more extensive and time-consuming clinical trial. However, due to practical constraints during a clinical trial, it is often impractical to test numerous different versions of the system. Ideally, **only the top-performing system is chosen for testing in the clinical trial**. This underscores the importance of in-house evaluations, serving as a

preliminary step to identify the top-performing system before proceeding to the more resource-intensive clinical trials.

Thus, first focusing on **in-house evaluation** of different developments of the AI system. In the training phase, dense delineations of polyps served as labels to supervise the segmentation task based on a Dice-score, as explained in the previous section. However, during the evaluation phase, the emphasis shifts towards the detection rather than precise segmentation. The focus on detection aligns with the clinical goal of effectively identifying potential malignancies in the screening process. Consequently, the in-house performance analysis of the AI system prioritizes enhanced polyp detection. This involves measuring the number of polyps detected by the AI system and assessing false positive detections. Doing so facilitates the offline comparison of individual models and allows for the optimization of hyperparameters during the development phase. When this type of evaluation is finished, a clinical trial with the top-performing implementation can be organized.

A **clinical study** aims to replicate real-life scenarios for assessing the practical benefits of an AI system. However, a standard randomized clinical trial may not ensure a blind and unbiased assessment. To address this, Tom has introduced a novel study design where the **endoscopist performing the colonoscopy remains blinded to the AI output** unless they potentially missed a lesion. In such cases, the **AI output is revealed only to a second observer**, also experienced in polyp detection.

This design enhances objectivity by limiting the influence of AI output on the performing endoscopist's decisions, providing a more rigorous evaluation of the system's real-world impact. Moreover, it minimizes risks for the patient. Indeed, in a traditional randomized trial, patient examinations are either supported by the AI system or not, having two distinct groups of examinations. However, if the AI system identifies a polyp that the endoscopist misses, it is in the patient's best interest to have the polyp documented. In the updated study design, all patients benefit from both the expertise of the endoscopist, and the detections made by the AI system. This approach prioritizes patient well-being while evaluating the AI system's contribution in a realistic clinical setting, and will serve as a template to test other similar AI systems in the future.

Two clinical studies conducted across 10 European centres using Tom’s AI system have revealed that endoscopists, on average, detect 5% more polyps when assisted by the AI system. Notably, these studies have demonstrated that **less experienced endoscopists derive the greatest benefit from the AI tool**, with their detection rate for malignant colorectal polyps more than doubling. These findings underscore the potential of AI assistance in enhancing polyp detection rates, particularly for less experienced practitioners.

13.6 Challenges

Both our clinical and technical experts have outlined specific challenges that the proposed AI system must address before its deployment in standard clinical care.

The proposed AI system for polyp detection during colonoscopy faces typical challenges before it can be implemented in standard clinical care.

- **Clinical challenges** include building trust through interpretability (explainability), seamless integration into existing workflows, generalization across diverse patient populations, and rigorous clinical validation.
- **Technical challenges** involve ensuring data quality and quantity, robustness to variability, real-time processing capabilities, scalability, and compliance with regulatory standards (MDR: Medical Device Regulation).

Successful deployment requires collaborative efforts to address these challenges effectively.

13.7 Future perspectives

What is the technical breakthrough needed to have the system adopted in the future? Let's see what our technical expert thinks is currently still missing.

Tom's vision for the future is to make sure the AI system is providing a broader support, not just enhancing the detection of polyps, but also classifying them into benign and malignant. Or even, as also seen in other use cases (e.g., Facial-based syndrome classification), **automated reporting**. The idea behind automatic reporting during a colonoscopy is to assist in the real-time analysis and interpretation of the procedure. If implemented properly, this process comes with several potential benefits:

- **Efficiency**: Automatic reporting can significantly reduce the time and effort required for manual reporting, allowing clinicians to focus more on patient care.
- **Real-time insights**: Clinicians receive immediate insights into findings during the colonoscopy, facilitating prompt decision-making and intervention if necessary. This enables quicker communication of results to patients, contributing to a more efficient and patient-friendly experience.
- **Consistency**: Automation helps ensure consistent and standardized reporting across different procedures, reducing variability and improving the overall quality of documentation.
- **Workflow integration**: Integration of automatic reporting into the workflow streamlines the reporting process, making it more seamless and less prone to delays.

While automatic reporting brings advantages, it should complement, not replace, the endoscopist's expertise. The goal is to enhance efficiency and accuracy, improving patient outcomes and overall clinical care. However, the question arises: **can AI replace future experts in colonoscopy**?

It is safe to conclude that AI in colonoscopy will not be a replacement for expert and expertise, but instead it will support our daily clinic operations, elevating our performance to the highest and most effective therapeutic levels attainable. "It is basically a GPS, helping us to become more efficient in the task at hand".

14 AI in action: AI-assisted surgery

14.1 Welcome to Module 14

Welcome to Module 14. This is one of several "AI in action" modules, where we shift our focus from theoretical concepts to practical use cases, allowing clinical and technical experts to share their insights and experiences with AI applications in the healthcare field.

Why the "AI in action" modules matter

These modules about real-life use cases are significant as they bridge the gap between theory and real-world impact. By delving into the insights of experts who have developed or utilized AI applications in healthcare, you will gain a nuanced understanding of the transformative potential of AI in the field. Each use case offers a unique perspective on the practical challenges, risks and opportunities that AI brings to healthcare.

This module is about AI-assisted surgery. Discover how AI can be applied in surgery through interview clips with an expert in this field.

Learning goals

- Examine case studies and real-world applications where AI has been successfully integrated into the clinical practice and in this use case: AI-assisted surgery.
- Gain knowledge on preprocessing steps (e.g., normalization and augmentation) and architecture choices in image analysis tasks.
- Learn the steps involved in creating a labelled dataset, particularly for surgical images, and understand the importance of accurate annotations for training effective models.
- Understand what on-edge deployment entails, its advantages, especially in time-sensitive environments like surgery, and the challenges associated with implementing such solutions.
- Analyse the use of AI and computer vision technologies in real-world surgical environments through case studies, focusing on the improvements in surgical outcomes and the challenges faced.

14.2 Context and clinical impact

In the past ten years, **3D models** have been used in oncologic **surgery** to improve outcomes in renal surgery. However, there are three main reasons why integrating 3D models into the actual surgery has been difficult.

1. Firstly, **aligning the model with the patient's anatomy** during surgery is a major challenge, because organs can shift during the operation, and the patient's position may be different during computed tomography scans.
2. Secondly, **automatically registering the organs** in a moving surgical video has been a challenge for a long time.
3. Lastly, when overlaying the 3D model, it can block the **view of non-organic items** like sharp instruments, which can create a hazardous situation instead of assisting the surgery. Finding a solution to this problem of **occlusion** has been the subject of ongoing research and could greatly advance various surgical fields and applications.

To tackle the occlusion problem, Orsi Academy, a Belgium-based training & innovation centre in minimal invasive & robotic surgery, brings research to practice with a **real-time augmented reality (AR)**

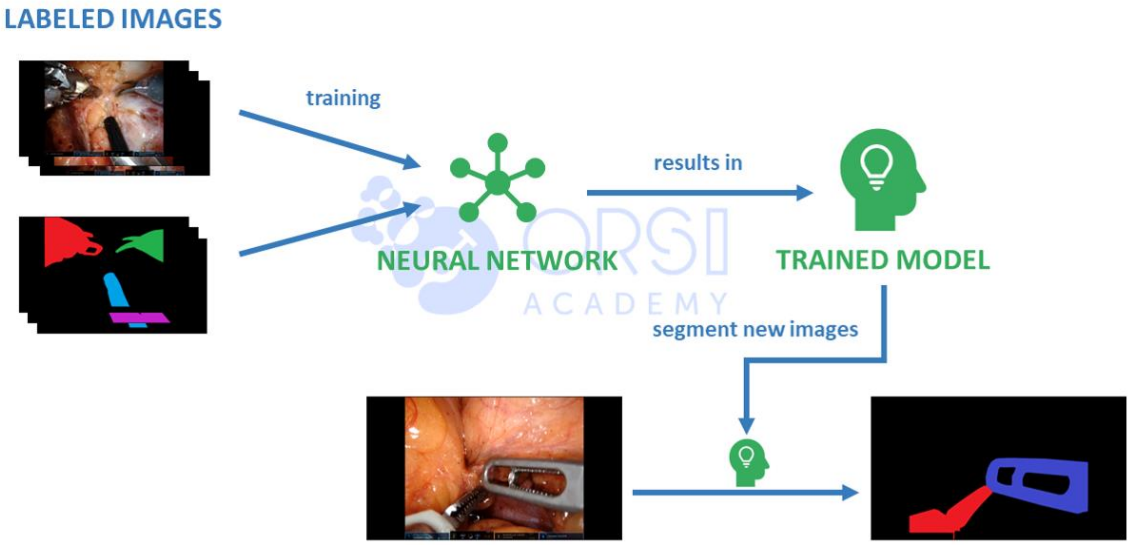
pipeline powered by NVIDIA Holoscan, a platform enabling medical device developers to accelerate development and deployment of AI applications at the edge.

14.3 Data

Videos and images of a surgery are the main data source in this use case. With deep learning, the AI application can be developed. **Deep learning** is like having a smart assistant that can help you extract or compile information of the immense amount of medical data available today. It is a computer-based approach that allows machines to learn from data and make predictions or decisions without being explicitly programmed. In the context of medicine, this means using computers to assist healthcare professionals in various ways.

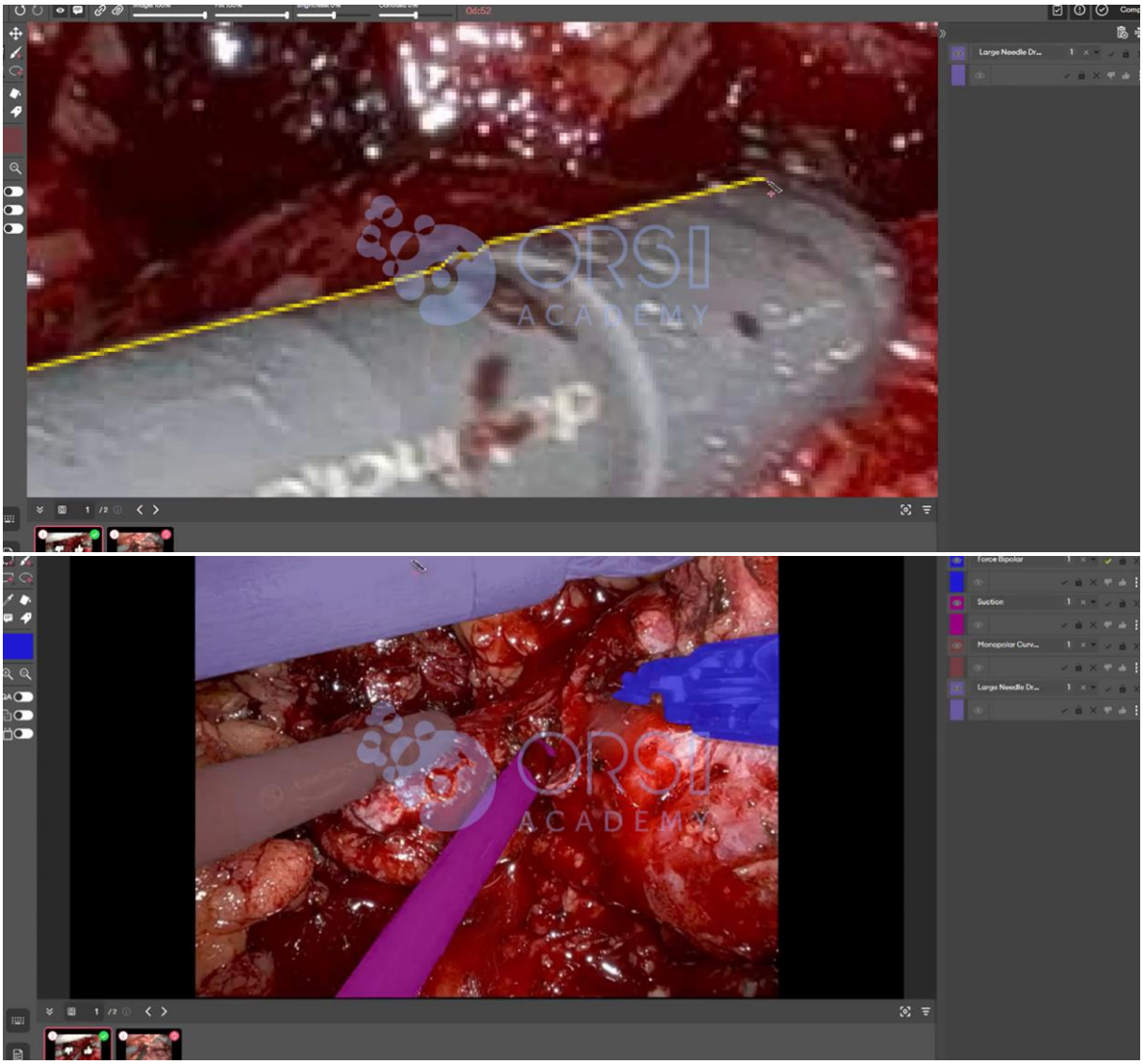
How to create a labelled dataset?

The main property of a deep learning model is that it learns by example. This implies that one needs to create a large dataset of **correct examples to learn from**. As is the case with human learning, the more ground covered in terms of different situations, the more suited the model's responses will be. **Data labelling** is a crucial step in the process of training deep learning models. It involves annotating or tagging the data you use to create this example dataset. The figure below shows a schematic overview of the different steps for creating and using a deep learning model, in this case a **segmentation model**, meaning the model will learn which pixels in an image belong to which type of structure. More specifically, the model will learn which pixels correspond to robotic surgery instruments.



To learn such a task, the format of examples in the training dataset should be pairs of original images and **masks**. All pixels in the masks should be black, or contain no information, except for the ones corresponding to the structures one wants to identify.

Data annotation can be facilitated using software platforms. The figure below shows an example of monopolar curved scissors being annotated manually. The edges are delineated, and afterwards the pixels inside the yellow area are labelled as being part of the "monopolar curved scissors" class. Secondly, the final annotation for this frame is displayed where all instruments are labelled.



How to preprocess data for training?

Resizing

Preprocessing image data is an essential step in preparing it for deep learning tasks like image classification, object detection, or segmentation. Typically, images are full HD or higher quality, meaning 1920x1080 pixels per image. When constructing a dataset with over tens of thousands of examples, this amounts to a lot of storage. When training a machine learning model, batches of data are often used instead of learning the examples one by one. This is done to firstly speed up the training process, and secondly because a model learns better by generalizing over multiple examples in parallel rather than sequentially. This implies that the more images can be processed together, the better. To **maximize hardware use**, images are typically resized to a power of two prior to training. Often used sizes are 512x512, 256x256, 128x128 and sometimes even 64x64 or 32x32. This is of course subject to the complexity of your task and what the requirement of detail is. A good place to start is the image quality or size that you need as a human to still extract the information.

Normalization

Deep learning models can be seen as very complex black box functions, which transform the input data into a desired output format. The black box function is shaped during training, but at its very core it is

a collection of multiplications and sums. The model will take the pixel values from the input images, which are between 0-255, and use these as a starting point for the next calculations. To **avoid these calculations becoming too large or complex**, the images are typically normalized before training with respect to the training data set. This is done by subtracting the mean dataset pixel value and dividing by the standard deviation dataset pixel for every image in the dataset.

Augmentation

To **increase the diversity of your dataset** and improve model generalization, you can apply data augmentation techniques. These include random rotations, flips, and brightness adjustments to generate variations of your existing images. This way one can create more examples than present in the current dataset. In the context of robotic surgery, an example is that sometimes the endoscope is covered with lens stains, obscuring part of the view. To emulate this, one could cover part of the image and the corresponding mask with blur or a black box to create a new training example.

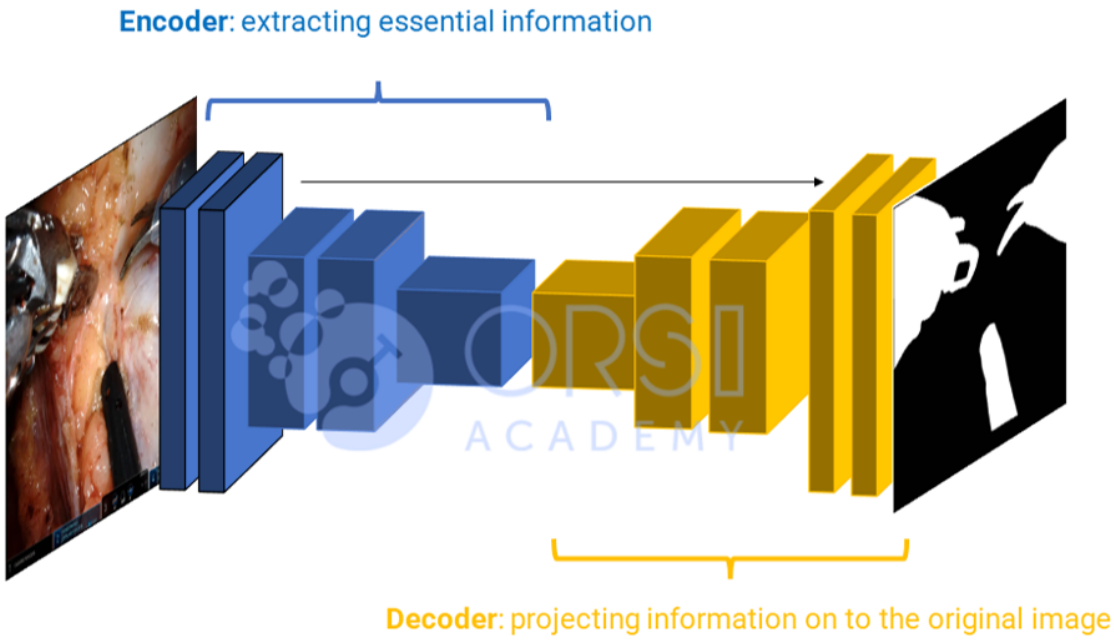
Splitting of the dataset

Eventually you want your deep learning model to be deployed in the real world. Unfortunately, it is impossible to capture the entirety of possible examples in a single dataset, even when there are tens of thousands of examples. When training a deep learning model, the dataset is divided into three parts: training, validation, and testing sets. A common split is 70-80% for training, 10-15% for validation, and 10-15% for testing. This allows you to **train, validate, and evaluate your model's performance on separate data**. The validation set is a recommended intermediary step because it allows you to change properties of the model to achieve a better score without overfitting on your test set, that should serve as a final assessment of how your model would behave inside a real world deployment.

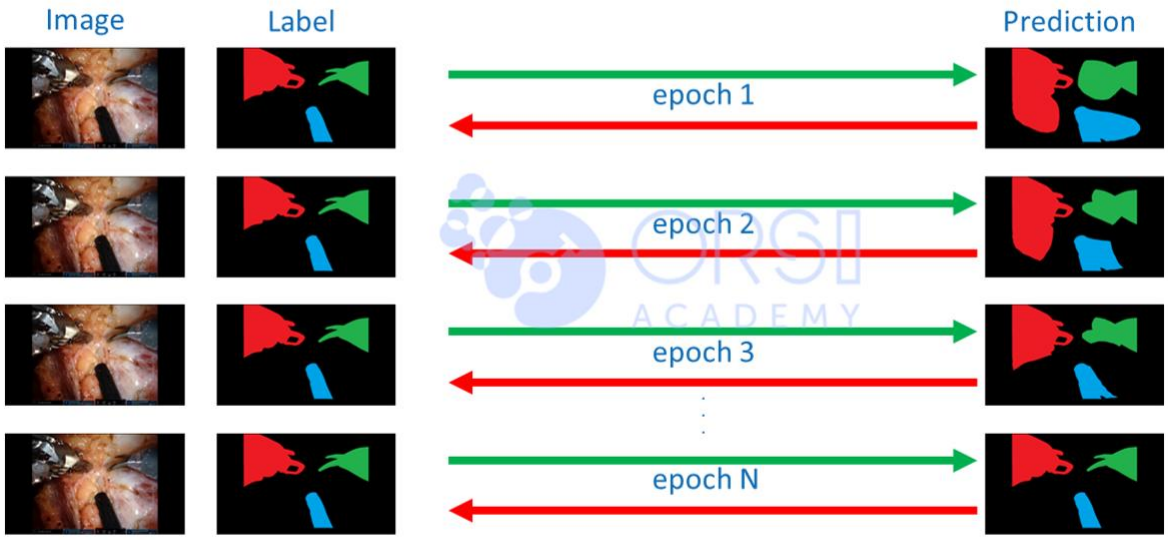
14.4 AI in action

How to train an AI model?

Choose an appropriate deep learning model architecture for your task. This could be a **convolutional neural network (CNN)** for image tasks, a recurrent neural network (RNN) for sequential data, or another architecture that suits your problem. In the case of the surgical instrument segmentation problem, the CNN type of networks are best suited. The figure below displays the general idea behind a CNN. From left to right the, information from the input images traverse the layers of the CNN, which can be roughly divided into an **encoder** and decoder part. Every layer of the CNN reduces the image size and transforms the information in the image by applying a filter convolution operation, hence the name. What essentially happens, is that in every layer, a window slides over the input image and only lets through the most relevant information. When the input arrives at the centre of the CNN, it has thus been condensed into the most relevant and compact representation possible. The **decoder** part does the reverse operation: it translates this compact representation back to the original dimensions, but this time transforming the format into something relevant to the problem at hand. In this case, a mask indicating the pixels corresponding to a surgical instrument.



A deep learning model is trained in a couple of iterations over the dataset, called **epochs**. Every epoch of the model is presented with input and output pairs, in this case an image of the surgical scene and the corresponding annotated instrument mask. The output produced by the model is compared to the corresponding label, and the mistakes are quantified. This is done by a suiting loss function, typically in the form of the amount of correctly predicted pixels. Feedback on the prediction error is fed back to the model, and the model parameters are updated in order to improve. The figure below illustrates how the predicted mask iteratively improves, and how the prediction resembles the label more over the different epochs.



What is on-edge deployment?

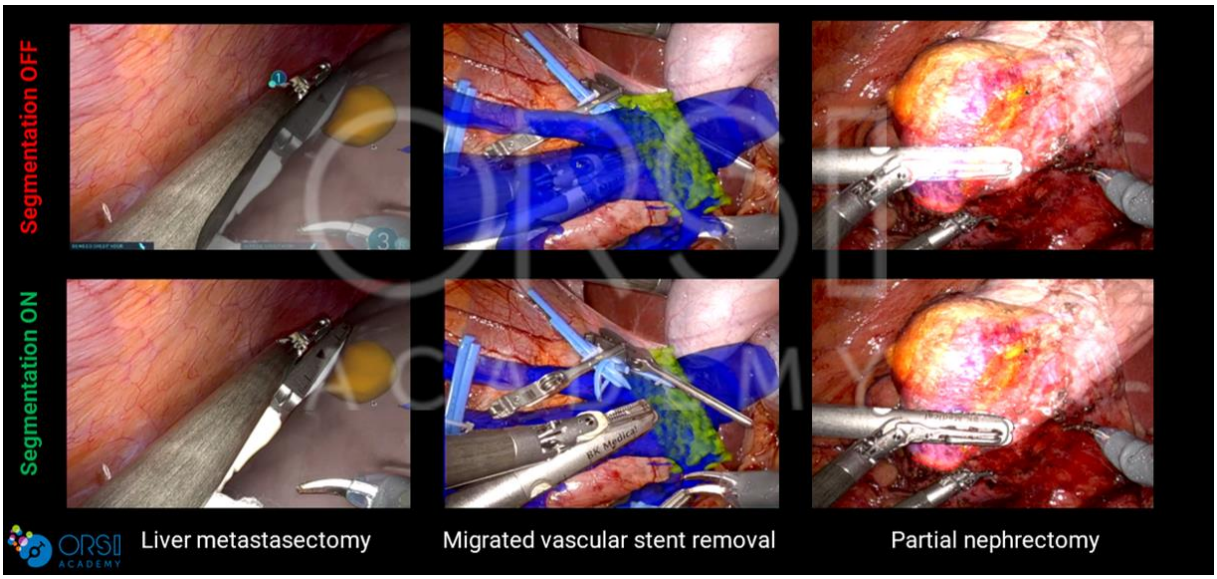
On-edge deployment, also known as edge computing or edge deployment, involves running an algorithm or software application directly on a device, rather than on a centralized server or in the cloud.

This approach offers several advantages, especially in scenarios where **real-time or low-latency** processing is crucial. The device is literally running at the edge, meaning right next to where the data is being created. The benefit of on-edge deployment in the context of medical applications is low-latency, where the surgeon needs to get immediate feedback on patient imagery. On-edge deployment does have some challenges, e.g. devices need to have enough processing power and memory to run the algorithms.

Orsi Holoscan Application

As we mentioned in the introduction, the integration of 3D models into surgery has been difficult. One particular problem is that the view of non-organic items can be blocked by the overlaying 3D model, which can create a hazardous situation. To tackle this problem, Orsi Academy brings research to practice with a **real-time augmented reality (AR)** pipeline. The solution uses a deep binary segmentation model trained to identify 37 classes of non-organic items including robotic and laparoscopic instruments, needles, wires, clips, vessel loops, bulldogs, gauzes and more. The non-organic segmentation mask is used to compose the AR scene enabling de-occlusion of the instruments.

The figure below shows the application of the pipeline with segmentation enabled/disabled during three in-human live surgery cases (liver metastasectomy, migrated vascular stent removal and partial nephrectomy). The images show the AR application being used during tumour demarcation and stent localization, while being validated by the ultrasound probe. The AR solution facilitates **targeted navigation** and in essence provides endoscopic ultrasound. Additionally, because the instruments are de-occluded, the segmentation provides a **sense of depth** during AR.



Endoscopic video is captured from the daVinci Xi Vision Cart using the SDI video output connectors at the back. After processing, the result is fed back into the surgical tower through DVI TilePro input, a connection that allows the surgeon to display third party information systems inside the surgical console. The output signal is split to a secondary monitor positioned at the bedside, where an assisting surgeon does the 3D model alignment to the surgical scene.

Zooming in on the second case, the application was successfully used to verify stent location during a nutcracker syndrome stent removal. **Nutcracker syndrome** is a rare vein compression disorder where the left renal vein is squeezed between the superior mesenteric artery and abdominal aorta, obstructing blood drainage and even risking blood to flow backwards, causing pain and blood in the urine. Blood flow is typically restored through endovascular stenting. Although effective, the stent had

migrated beyond the compression site over time, causing the symptoms to return, and in addition posing an incidental obstruction for blood flow.

What other information can be extracted using surgical instruments?

Next to binary segmentation, one can also train algorithms to **classify and track different instruments**. This allows for all kinds of metadata to be extracted from vision information, and draw parallels between different procedures. The figure below shows an example of automated tracking of surgical instruments where every tool is assigned an ID, a class, and a percentage indicating how certain the model is about the prediction.

Which type of tools were used during the procedure? How long was each tool used? Do procedures, where certain tools were used longer, have things in common? When plotted over a longer period, we can construct a **surgical fingerprint** specific for that procedure, and compare it with others.

14.5 Evaluating AI

Start of transcript. Skip to the end. The evaluation of the AI model can happen on two aspects.

First of all, you can assess the real time aspect of a model. Meaning, if I put a video in, how rapidly will it be able to tell me what is happening inside this video? For instance, suppose that a surgical error happens. You want to know as quickly as possible if that error is happening. If the system can only tell you one minute later that this error is happening, maybe a lot of damage has been done already. Even if you want to go one step further and you want to predict if errors will happen, you of course need to forecast, and the time limit actually becomes negative, and you want to predict, so you want to be before your surgeon. So that's one aspect to take into account, the real time aspect.

The other aspect is performance. Performance on how precise can we detect what we have been doing. And if you go into the most basic aspect of tracking surgical activities, you could think about one task is tracking instruments. So you're tracking how instruments move throughout the surgical image. And you delineate them very precisely. So you delineate them and you want to track if this is the image that it needs to track, am I really tracking my instrument like this or am I tracking my instrument with a large circle around it? So it's actually covering everything, but it's not very precise. So metrics that we use

for this, on this basic computer vision task is for instance, mean intersection over union, where you look at how is my intersection of my prediction versus what it should detect compared to the union of both of them together. So this is one aspect that is often used in computer vision. Other ways of looking at performance is, of course, sensitivity and specificity. So recall bias, the classical performances, area under the curve. Because if you want to detect an error in a time point, you want to know have I detected the error or did I miss it? And it depends on the application. Sometimes you do not want to miss a single error. You want to be very sensitive. Otherwise it doesn't really matter. And you want to be as precise as possible, but you don't want to have that overly sensitive system. In our case, we tend to look at it to be very sensitive so that we detect every error so that we can also say, oh actually this was not an error, so that we are sure that we do not miss an error. So we want to go for a very sensitive system that is not necessarily specific.

Is there any golden threshold for choosing sensitivity or specificity? In my opinion, for AI in healthcare this really depends on the clinician. This depends on the use case. This depends on what is useful. If a surgeon tells you, actually if I know it will happen, or if I know that this artery is located in this position, or if this error just happened, it doesn't really matter if I know it with 60% specificity or precision or recall or if I know it with 80%, I will still have a double check, because I cannot permit to make the error. So I think in healthcare or AI in healthcare, we should not always strive to get the super upper best performance and go for ways to achieve 98% rather than 97%. But we should think about how is it adding value to the use case. And I think that's the whole debate on performance. Can somewhat be a bit nuanced towards what clinicians really think and what they really want to do.

14.6 Challenges

As you can imagine, there are several challenges when applying AI in surgery.

The challenges to apply AI and surgery are at one point very evident. If you make a wrong decision during surgery, well, it's basically irreversible because you're doing the surgery at that point and it's not something easy to revise after, or get a colleague, or get help. So it's very imminent and it should be very precise. The systems that we make to improve the surgeon’s skills through overlays of anatomy or through detecting errors, they help to inform a surgeon, but they do not give a precise definition of okay, this is the next step to take during your surgery. if you want to use this, we do this very research based and in a lab setting. If you want to use this in a real life setting, it has huge ethical constraints or ethical implications as well as regulatory, because you're making a medical device which is of the highest risk because it's interventional. And if something goes wrong, it's very difficult. So this is a real showstopper, rightfully, for the use of AI in surgery.

The other point is that, at present, the state of the art in surgery is so basic or so premature that it's sometimes difficult, to convince surgeons about the use of artificial intelligence in surgery. Some of them think it can never be done, while others think that in ten years’ time, the robot will overtake their surgery, and they will not have a job anymore. I think the truth is somewhere in between. So informing clinicians or surgeons on what the future could be is very crucial. But creating awareness that you need to systematically, for instance, record cases. Or that you systematically need to think about the data infrastructure in your hospital is something where we face a lot of challenges. Even simply recording video cases in a high quality to be able to do this type of developments, is a major challenge. So it's as simple as that. Just pressing a record button is sometimes very difficult in an operating room.

And then there are the very difficult challenges. Like what does it mean when a system takes a decision and how will it influence your surgery at that point in time? Nobody really has the answer. Maybe we

will know it in a couple of years. But at present, this is really pure hypothesis. And is enduring discussions.

14.7 Future perspectives

The best-case scenario to enable artificial intelligence in surgery is that we can show that it impacts the outcomes of our patients. And so we are very based on the intraoperative part where we look at surgical actions, surgical video, surgical manipulations.

One type of research is looking at, we have 10,000 patients operated with this technique, and postoperative we see these values of haemoglobin, and these values of CRP or infection, these days of hospital stay. And then it's mainly one type of advanced statistics or forecasting where you're crunching numbers and you're looking for correlations. It's still machine learning, but it's not the intraoperative artificial intelligence that we are looking for. So if you want to show the value of intraoperative artificial intelligence, you also need to link that to outcomes, which means that suppose that the system can help you define which phase of the surgery you're in, or that you might make an error and you can prevent it, you want to link that to the outcome that actually you made the patient better. But it's very difficult if you prevent an error to show that this had an impact on outcome, because you're comparing it to surgeries where this error would not have happened. So to show that you can impact the surgery using artificial intelligence and link that to outcome is actually a chicken or the egg story. Because you're looking at ways to improve the surgery and you want to prove that compared to outcomes. But the gain you can have there is sometimes very small. So you need a lot, a lot, a lot a lot, ...of patients. What we would need to show a link to outcomes is very big collaborations. And we're building some of those at least in Flanders for now, but also on a European level, to make sure that we can aggregate all this data and all these videos and all these outcomes to be able to show real clinical value, because if we can show clinical value, this is what the essence is of evidence-based medicine. We show that it impacts outcomes and we show that this AI we feel now or the researchers feel that it can have a real impact, but we do not have the hard numbers yet to show that this can have an impact. So, I hope that this in the future will change. We're pushing very hard to make that happen. But collaboration is really key for us.

I think that artificial intelligence will not replace the surgeon in our case, but it will perfect the surgery. So it means that, this is a very classical quote that has been around for like ten years, that physicians using AI will replace physicians that are not using AI. And for surgery, this is also very true. If you know that, suppose you're in the middle of the night, and you're a young surgeon like myself, and you don't have any support, but you have an AI system in the back that has seen 20,000 cases, which is probably 19,000 cases more than you. If you're at some point doubtful, and you know that this system is 99% sure that what you have done is okay. It might help you improve in a critical time, critical situation where you have no direct access to your supervisor because he or she is not directly in the hospital, or was not physically available.

So I think we will have improvements and it will still take a lot of years before these systems get in place, because of regulatory issues, ethical issues and whatnot, but we should also be able, as physicians, to estimate where the system can help us and where not, and to understand where the system has flaws or where the system is helping us. And if this is not the case, or if you don't understand these risks, you cannot use this technology very faithfully. Because if you don't know that the system is actually always wrong at this type of surgery or at this type of phase in the surgery, why would you trust it to some extent? Or if you don't know what's behind it, or if, for instance, the system has been trained on a type of surgery from a certain population and you're always operating in another population and you don't know how a data set transfers to another data set, for instance, these are all

things that you need to get a feeling about. And if you as a clinician then get these questions or you get somebody in your operating room that gives you this new system, you should be able to interpret it to its value. So I think we will have real value of artificial intelligence. But the final decision will always, should always be taken by the clinician. But if you can improve your care by giving you a push in the back, why not use it? If you show that you can do better with AI than without AI, it's a no-brainer to use it. So, I think that's relevant for every type of AI in healthcare application. You will probably do a better job. If not, the AI should not be in the hospital.

15 AI in action: Hypothesis generation in psychiatry

15.1 Welcome to Module 15

Welcome to Module 15. This is the last of several "AI in action" modules, where we shift our focus from theoretical concepts to practical use cases, allowing clinical and technical experts to share their insights and experiences with AI applications in the healthcare field.

Why the "AI in action" modules matter

These modules about real-life use cases are significant as they bridge the gap between theory and real-world impact. By delving into the insights of experts who have developed or utilized AI applications in healthcare, you will gain a nuanced understanding of the transformative potential of AI in the field. Each use case offers a unique perspective on the practical challenges, risks and opportunities that AI brings to healthcare.

This module is about hypothesis generation in psychiatry. Discover how AI can be applied in psychiatry through interview clips with an expert in this field.

Learning goals

- Examine case studies and real-world applications where AI has been successfully integrated into the clinical practice and in this use case: hypothesis generation in psychiatry.
- Explore how AI can augment traditional psychiatric practices which currently rely on clinical interviews and self-report questionnaires.
- Learn and understand the basics of Gaussian Graphical Models (GGMS) and Bayesian Networks (BNs).
- Examine how GGMs and BNs are used to analyse psychiatric data, identify relationships among symptoms, and generate hypotheses for further research.
- Evaluate the benefits and limitations of using AI in psychiatry, particularly the challenges of data privacy, informed consent, and the potential for bias and fairness issues.

15.2 Context and clinical impact

As Zarah, Eric and Vivian sat around the kitchen table, their conversation lingered on Noah's struggles at school. The parents had just visited Noah's school for the parent-teacher conference, where the teacher told them that Noah has some difficulties with learning, related to low self-confidence.

"It's not easy, understanding and addressing mental health issues, particularly related to self-worth," Zarah sighed, her concern palpable. Grandma Vivian, a retired psychologist, offered her seasoned perspective on the matter. "Indeed, dear. Mental health is a complex puzzle. Psychiatrists continue to face challenges in accurately diagnosing and treating conditions that impact self-worth, such as depression and anxiety."

Eric interjected, "But couldn't AI lend a hand? Help psychiatrists sort through the complexities?"

Vivian's expression turned thoughtful. "AI has its strengths, but mental health is more than just data points and algorithms. Each person's experience is unique, and understanding their narrative is crucial for accurate diagnosis and treatment." She continued, "Psychiatrists face the challenge of interpreting these narratives, which are often nuanced and layered. While AI can aid in processing data, it may struggle to grasp the subtleties of human communication."

Zarah nodded. "AI algorithms may lack the ability to understand the nuances of patient communication and may rely too heavily on quantifiable data, potentially leading to misdiagnosis or overlooking important clinical information."

The use of AI in psychiatry has emerged out of a **need for more accurate, efficient, personalized, and scalable mental healthcare**. Current approaches to diagnosing psychiatric disorders largely rely on clinical interviews and physician/patient questionnaires. While these methods are the gold standard, they also have their limitations. For instance, they can be time-consuming, rely heavily on the patient's ability to accurately report symptoms, and are susceptible to variability in clinicians' judgment. AI has the potential to augment traditional psychiatric practices in several ways.

The intersection of AI and Psychiatry is not a novel concept. In fact, the first book to explore the potential of AI in Psychiatry was published as early as 1985. Authored by Hand, the book delved into the capabilities of algorithms in the realm of mental health, shedding light on how computational techniques could revolutionize psychiatric practices.

However, the question arises: why should the current decade be any different? **Why is there a renewed interest in the amalgamation of AI and Psychiatry now?** The answer lies in a confluence of factors that have emerged and evolved over time.

1. There has been a significant economic shift in the healthcare sector, with decreased financing in mental health. This has necessitated the search for innovative and **cost-effective solutions to provide mental health services**, and AI has emerged as a promising candidate.
2. The **prevalence of mental disorders** has seen an alarming increase in recent years, as in 2019, 1 in every 8 people is estimated to live with a mental disorder. This escalating crisis calls for robust and scalable solutions, and AI holds the potential to meet this demand.
3. Despite advancements in mental health awareness, **social stigma** associated with mental illness persists. This stigma often discourages individuals from seeking help, leading to delayed or no treatment. AI-powered solutions, such as chatbots and online therapy platforms, offer a level of anonymity that can help overcome this barrier.
4. The advent of new technologies and the increasing digitization of healthcare have made it possible to reach out to mental health professionals in ways that were not possible before. However, access to in-person mental health services remains a challenge for many, especially in remote or underserved areas. AI can bridge this gap by enabling **remote diagnosis and treatment**, making mental health care more accessible.

In the current use case we will focus on the human desire to feel worthy, which is an important constituent of human behaviour. A troubled self-esteem has been shown to contribute to several psychiatric disorders such as eating disorders (Pearl et al., 2014), substance abuse (James, 2011), and schizophrenia (Xu et al., 2013).

15.3 Data

The dataset of this use case comes from a study with 680 French-speaking students in Belgium (59% females) (Briganti et al, 2019). The students were asked to fill in the Conting**encies of Self-Worth Scale (CSWS)**, which is a psychometric tool proposed by Crocker et al. (2003), composed of 35 items meant to assess self-worth contingency in the following seven domains:

- **Family Support (FS)** measures the influence of perceived approval, support and love from family members on the feeling of self-worth (e.g., "Knowing that my family members love me makes me feel good about myself");

- **Competition (C)** evaluates how self-worth is influenced by feeling better than others (e.g., “Knowing that I am better than others on a task raises my self-esteem”);
- **Appearance (A)** quantifies how physical traits influence the way people evaluate themselves (e.g., “When I think I look attractive, I feel good about myself”);
- **God's Love (GL)** measures the association between religiosity and self-esteem (e.g., “My self-worth is based on God's love”);
- **Academic Competence (AC)** evaluates the impact of grades on self-esteem (e.g., “Doing well in school gives me a sense of self-respect”);
- **Virtue (V)** measures the connection between self-worth and the adherence to a moral code (e.g., “Doing something I know is wrong makes me lose my self-respect”);
- **Other's Approval (OA)** measures the influence of perceived approval from others on self-esteem (e.g., “I can't respect myself if others don't respect me”)

For each of the seven domains there are 5 related items in the questionnaire. The items are shuffled in the questionnaire. Item score ranges from 1 (strongly disagree) to 7 (strongly agree); some reverse-scored items are included. Hence, the values that each of the above domain variables can take is from 5 till 35 (having either the lowest score in all 5 related questions or the highest score).

Ethics

The analysis of these three data sets was approved by ethical committees (Comité d’Éthique hospitalier CHU Brugmann, Ref CE2020/39, CE2021/58, and Comité d’Éthique hospitalo-facultaire Erasme-ULB; Ref. P2017/379).

Data preprocessing, harmonization and normalization

The dataset in this use case comes from questionnaires using a predefined psychometric scale, hence, no missing values were present and no extensive harmonization or preprocessing was needed. Normalization through Min Max Scaling was performed:

$$\text{scaled value} = \frac{\text{value - min value}}{\text{max value - min value}}$$

15.4 AI in action

Discovering new methods for hypothesis generation is particularly vital in psychiatry, a field distinct from other medical areas due to the ongoing debates around the nature of mental illnesses and a pervasive lack of clarity about the underlying mechanisms of mental disorders. This challenge has not only obstructed progress in understanding psychiatric illnesses, but also in comprehending broader constructs like personality and emotions. Recently, the study of psychiatric entities as complex systems comprising interconnected components, has spurred significant knowledge growth, facilitated by the accessibility of tutorials and open software aimed at applied researchers. Despite this, such advancements are often diminished by a limited focus on theory construction.

The past decade has witnessed the adoption of **network theory** in the field of psychopathology, which views episodes of disorders as resulting from the interplay among various symptoms. Network theory is supported by a suite of statistical techniques known as **network analysis**, wherein networks representing symptom interactions comprise nodes (the symptoms themselves) and edges (the connections among symptoms). Unlike social networks, where nodes (people) and edges (like

friendships) are visible, psychopathology networks require statistical methods to reveal the unseen connections between symptoms.

In many areas of psychopathology, such as posttraumatic stress disorder, depression, and bipolar disorders, **Gaussian graphical models (GGMs)**, also known as partial correlation networks, are commonly utilized for estimating these network structures. These models are part of a broader class known as pairwise Markov random fields, which also include using models for binary data and mixed graphical models for handling mixed data types. However, a limitation of these models is their inability to ascertain causal directions since their edges are undirected, thus preventing any definitive conclusions about whether one symptom causes another.

More info on Gaussian graphical models (GGMs)

A **Gaussian graphical model (GGM)** is a way of representing how different variables are related to each other under the assumption that these relationships are normally (Gaussian) distributed. Imagine you have several factors—like temperature, humidity, and wind speed—that you suspect might be interrelated. A GGM visually maps out these relationships using a **network**, where each **node** represents a factor, and the lines between them (called **edges**) represent the strength and nature of their relationships. If there is no line between two factors, it suggests that, given other factors, these two are independent.

The strength of a GGM is that it helps you understand **which variables are directly related and how strongly**, which can be very useful for understanding complex systems where many variables interact.

The key computation in a GGM involves the inverse of the covariance matrix (precision matrix) of the data. Here's how it typically works:

- Covariance matrix: Compute the covariance matrix of these variables, which measures how changes in one variable are associated with changes in another.

- Precision matrix: Calculate the inverse of this covariance matrix. The precision matrix is crucial, because its entries directly indicate partial correlations (i.e., the correlation between two variables while controlling for others). A zero in the precision matrix indicates conditional independence between two variables.

- Graph construction: A graph is then constructed where nodes represent variables, and an edge between two nodes is included if the corresponding entry in the precision matrix is non-zero, suggesting a direct dependency.

To address these shortcomings, **Bayesian networks (BNs)** can be particularly effective. Grounded in principles of causal reasoning, BNs utilize a **directed acyclic graph (DAG)** and a joint probability distribution of the variables to define the conditional independencies and the strength of causal effects among variables. Unlike GGMs, the directed edges in Bayesian networks make them ideal for modelling and understanding **plausible causal relationships** in observational data, thereby enhancing insights that are derived from partial correlation networks. They require strong assumptions for causal inference, including the need for a comprehensive measurement of all variable causes and the absence of latent variables or selection bias.

More info on Bayesian networks (BNs)

Bayesian networks (BNs) are similar to GGM but with a direction and a clear notion of causality. They are models that represent variables and their conditional dependencies via a **directed acyclic graph (DAG)**. Each **node** in the graph is a variable, and a **directed edge** from one node to another indicates

that the first node has a direct influence on the second. The graph is "acyclic," meaning it doesn't loop back on itself, ensuring there's a directional flow from causes to effects.

These networks are "Bayesian", because they use Bayes' theorem to update the probability of the outcomes as new evidence is introduced. This makes Bayesian networks powerful tools for decision making and predictive analytics, as they can adapt to new data.

Bayesian networks estimation:

1. Structure learning:

Creating a BN starts with determining its structure—a directed acyclic graph (DAG) where nodes represent variables like symptoms, and directed edges show causal relationships. There are three main approaches for structure learning:

- **Expert knowledge**: In fields such as psychiatry, expert insights help define probable causal paths.

- **Constraint-based algorithms**: These use statistical tests to check for conditional independence among variables, employing algorithms like the PC algorithm to build the network. (This approach has been selected for the example in the current use case, the exact definition of the algorithm is out of the scope of this course)

- **Score-based algorithms**: These involve scoring multiple potential network structures to see how well they fit the data, often using search methods as those described in Module 3.

2. Parameter estimation:

After defining the structure, the next step is to estimate the strengths of connections, or the conditional probability distributions (CPDs), for each node. Nodes with parents have CPDs that quantify parental influences, while root nodes (those without parents) have prior probability distributions. Estimation methods include:

- Maximum likelihood estimation: Optimizes probabilities to best fit the observed data.

- Bayesian estimation: Updates prior parameter distributions with new data to form posteriors, ideal for handling limited data and integrating prior beliefs.

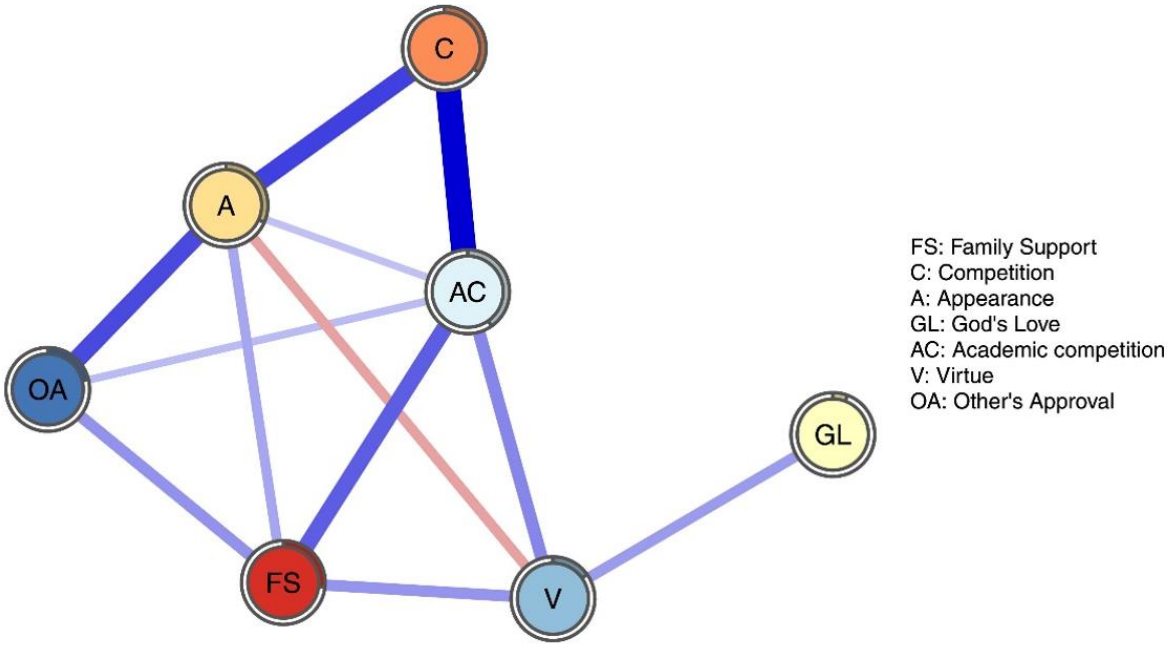
3. Utilization of Bayes' theorem:

Bayes' theorem is pivotal, especially in the inferential phase of BNs. It updates node probabilities based on new data, crucial for refining diagnostics or predictions in response to new symptoms observed in patients, thus enhancing decision-making in clinical settings.

Bayesian networks serve as a robust framework for predictive modelling, where each **node** represents a variable, and **directed edges** indicate conditional dependencies. In this network, pairs or groups of nodes can function as individual predictive models. For instance, if one node represents stress and another represents depressive symptoms, knowing the level of stress can help predict the likelihood of depressive symptoms. Each node is equipped with a **conditional probability distribution (CPD)** that quantifies the effects of its parent nodes, enabling precise predictions about its state based on known values of influencing factors. This setup allows for identifying which variables are most influential or predictable. Nodes with multiple influencing factors (many incoming edges) are typically more predictable, while those that influence many others (many outgoing edges) are key in forecasting multiple outcomes in the network. This analysis helps prioritize variables for effective monitoring and intervention in fields like psychiatry.

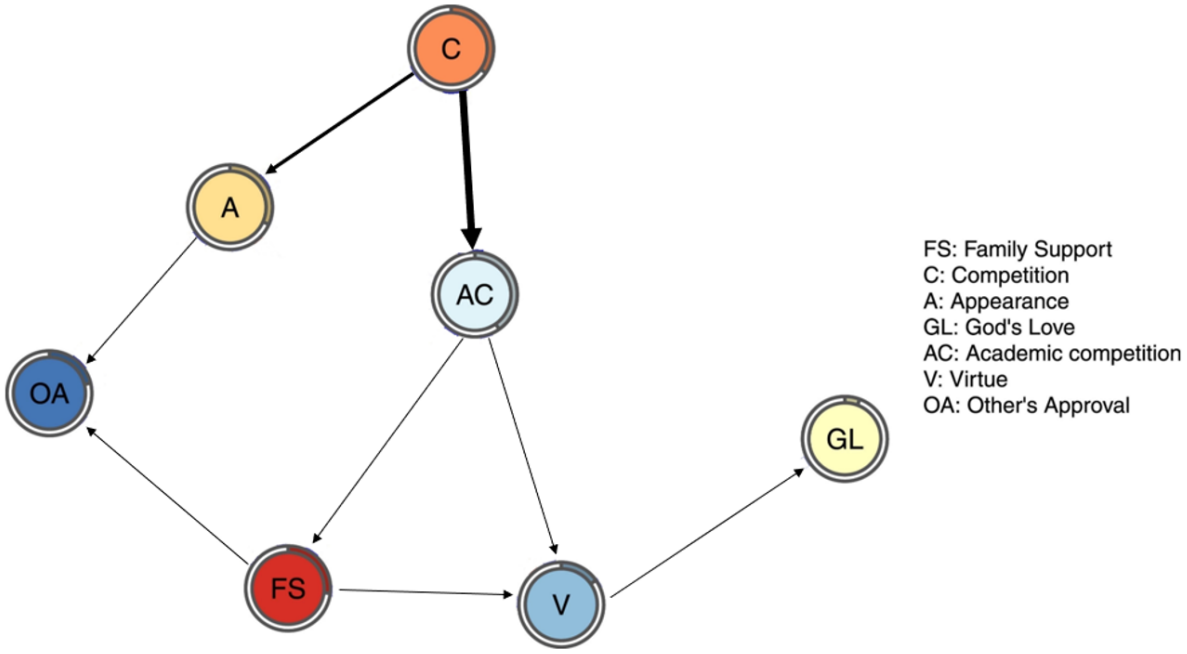
Gaussian Graphical Model (GGM) of self-worth

The figure below illustrates the estimated **seven-domain network of self-worth**. The network is composed of domains that connect with each other. Each domain is represented with a different colour. Competition and Academic Competence share the strongest connection in the network; Other's Approval also shares a strong edge with appearance. Competition and Appearance, as well as Academic Competence and Family Support, are also positively connected. Family Support is positively connected with most domains. God's Love is only connected to Virtue. Appearance and Virtue share a negative connection.



Bayesian Network (BN) of self-worth

The **Bayesian network learned from the self-worth data** is shown in the figure below. It has Competition as a root node that has both Appearance and Academic Competence as children. Academic Competence is a parent node of Family Support and Virtue. Other's Approval is a child node of Appearance and Family Support. God's Love is a child of Virtue. The Self-worth Bayesian Network also empirically supports the idea that Competition, grossly understood as feeling worth when self-comparing with other people, is one of the core sources of self-worth. Moreover, the three other main parent nodes for other nodes in the network, Appearance, Academic Competence, and Family Support, are all defined through interpersonal sources of self-worth, recognized as core parts of the constructs from developmental points of view.



To go a step-further, but also to "simplify" things at the same time, we will focus only on the variable **Other's Approval (OA)**. As we can see from our networks, it shares connections with Appearance (A), Academic Competence (AC), and Family Support (FS), and has high predictability. To predict the "OA" variable based on the "AC", "A", and "FS" variables, we can try different regression models and evaluate their performance to find the best one. We will do:

- **Data splitting**: We will split the data into training and testing sets to evaluate the performance of the models.
- **Model selection**: Linear Regression, Decision Tree Regressor, Random Forest Regressor, Support Vector Regressor.
- **Model evaluation**: We will evaluate the models using appropriate metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2), which are metrics relevant for the regression task of the estimation of a variable.

	Linear Regression	Decision Tree	Random Forest	Support Vector Regressor
MAE	4.383342	6.909926	4.886566	4.384703
MSE	29.404960	70.890165	38.292407	29.700208
R2	0.327381	-0.621565	0.124087	0.320628

As we have shown, the use of **GGNs and BNs can be exceptionally useful in identifying a vast array of independent and dependent variables**, as well as numerous potential clinical prediction models. For instance, within a network, each symptom can be linked to several local predictors, which themselves can be predicted by other variables easily identified on a graph (in the previous example: prediction of OA from A, AC and FS). This structured approach not only significantly enhances the quality of **clinical prediction models**, but also facilitates **hypothesis generation** in psychiatric research. By mapping out how symptoms and their predictors interact, researchers can formulate new hypotheses about underlying mechanisms and causal relationships, which are critical for advancing understanding and guiding innovations in clinical practice.

This capability of Bayesian Networks to systematically **reveal connections and predict outcomes** helps researchers to pose and test hypotheses, enhancing both the depth and breadth of investigations in

psychiatry. Such a methodological framework supports the continuous refinement and validation of existing models, paving the way for more accurate and effective therapeutic strategies.

15.5 Evaluating AI

Performance evaluation is, of course, of paramount importance when we deal with AI models, specifically in mental health. Now, I like to use white box kind of algorithms for artificial intelligence. This is why graphical models such as pairwise marking random fields or Gaussian graphical models can be interpreted as we see them from a graphical point of view. This makes it so that oftentimes performance can be interpreted in terms of precision and accuracy of the model to retrieve true effect of two given symptoms, or two getting signs of mental diseases, but also in the way that the algorithm helps us for further understanding of the disease. And so interpretation of performance can be both objective through measures that are collectively and objectively collected, such as bootstrapping and other methods of performance evaluation, but also more subjectively in terms of how much does the model help me understand what's going on in patients?

15.6 Challenges

Privacy concerns due to symptoms of mental disorders

People with mental disorders often share **deeply personal and sensitive information** during their treatments. AI systems, therefore, need to be designed with robust privacy safeguards to ensure the confidentiality of this data. There is a risk that data related to mental health could be used in ways that stigmatize or discriminate against individuals. For instance, it could potentially be **misused** by employers, insurance companies, etc. Thus, **ethical usage and secure storage** of data become paramount.

User experience problems due to mental disorders

Individuals with certain mental disorders might find it challenging to interact with AI systems due to increased cognitive load. The user interface should be designed to be intuitive and not to overwhelm the user. AI systems should be **accessible to people with various mental disorders**, including those who may have impaired cognitive function. This involves developing interfaces that are easy to use and understand, incorporating visual aids, simple language, etc.

Lack of clear view on the added value (Possibility of personalized models)

AI has the potential to leverage vast amounts of data to create highly personalized treatment plans. However, there is a lack of a clear understanding of how to effectively implement this in psychiatry, partly due to the complexity and variability of mental health disorders. The field needs to **develop a strong evidence base** demonstrating the effectiveness of AI-powered interventions. This includes conducting rigorous clinical trials to validate the efficacy of AI tools. To realize the full potential of AI in psychiatry, there needs to be close collaboration between AI experts and mental health professionals. This collaboration could foster the development of AI tools that are grounded in psychiatric theory and practice, thereby providing a clear view of the added value.

Quality and quantity of data in the wild

In psychiatry, the quality of data is highly dependent on several critical factors, making it a complex and often challenging field for data analysis.

- One primary factor is the **quality of Electronic Health Records (EHRs)**, which can vary significantly depending on the standards upheld by individual doctors and hospitals. These records are crucial as they contain detailed patient information and treatment histories.

- Another important aspect is the **classification systems and terminologies** used, such as ICD-11, DSM-5, and SNOMED-CT. These systems provide a framework for diagnosing and coding disorders, but differences and updates in these classifications can lead to inconsistencies in how data is recorded and interpreted.
- Additionally, the incorporation of **patient-reported data** plays a vital role in the quality of psychiatric data. This type of data includes subjective reports from patients regarding their symptoms, treatment responses, and overall well-being, which are essential for a comprehensive understanding of their health status.
- Furthermore, data in psychiatry is extremely **heterogeneous**, adding another layer of complexity. This heterogeneity is evident in the variety of diseases treated, which can differ greatly among specialized centres and influence the types of data collected. The modes of treatment, whether outpatient or inpatient, also vary and significantly impact the data, as do the treatment modalities employed, ranging from psychotherapy and psychopharmacology to more intensive interventions like Electroconvulsive Therapy (ECT).
- The variability extends to the **demographic and geographical diversity** of patients, which introduces additional variations in data due to cultural, social, and economic factors that influence health outcomes and treatment approaches. As a result, psychiatrists and researchers must navigate a multifaceted landscape of data, where synthesizing information from disparate sources becomes a critical challenge.
- Moreover, the **dynamic nature of psychiatric conditions**, where symptoms and their intensities can fluctuate over time, further complicates data collection and analysis. This variability necessitates longitudinal studies and continuous data collection to capture the full spectrum of the disorder and its progression or remission over time.

Consequently, the robustness of data analysis and the resulting conclusions in psychiatry depend significantly on addressing these complexities effectively.

15.7 Future perspectives

The integration of AI in psychiatry promises significant advancements in diagnosing and treating mental health disorders, but it also brings several challenges that need to be carefully managed to ensure ethical and effective implementation in the future.

- **Informed consent**: The deployment of AI in psychiatric care intensifies the need for clear informed consent processes. Patients must be thoroughly informed about how their data will be used by AI systems, the implications of these technologies, and the potential risks and benefits. The complexity of AI systems makes this particularly challenging, as it is difficult to explain AI processes in understandable terms, which can hinder patients' ability to make well-informed decisions about their care.
- **Bias and fairness**: AI systems inherently carry the risk of perpetuating existing biases present in the training data. In psychiatry, this could translate into diagnostic and treatment biases, where certain demographics might receive less effective or inappropriate care based on skewed AI insights. This can exacerbate existing inequalities in mental health care. Addressing these biases requires a continuous effort in developing unbiased data collection methods, diversifying training datasets, and implementing algorithmic fairness assessments.
- **Regulatory and ethical challenges**: As AI systems become more autonomous in making clinical decisions, they pose unique regulatory and ethical challenges. Defining the liability for decisions made by AI, especially when they result in adverse outcomes, is complex. Regulatory

frameworks are still evolving and need to keep pace with the rapid advancements in AI to ensure that these systems are safe, effective, and operate within ethical boundaries.

- **Integration into clinical practice**: Integrating AI tools into existing healthcare systems involves significant logistical and practical challenges. This includes training healthcare professionals to use these tools effectively, modifying existing workflows to accommodate AI technologies, and ensuring that these tools genuinely enhance, rather than complicate, patient care. Additionally, the potential for AI to dehumanize aspects of care—reducing patient interactions to data points—must be carefully managed to preserve the empathetic core of psychiatric practice.
- **Long-term impact on professional skills**: As AI systems take on more diagnostic and therapeutic tasks, there is a potential impact on the skills of psychiatric professionals. Maintaining and developing the professional skills of human practitioners is crucial, particularly those skills related to interpersonal patient interactions and complex decision-making that AI cannot replicate.

In summary, while AI offers transformative potentials in psychiatry, addressing these multifaceted challenges is essential for harnessing its benefits responsibly and effectively. The future of AI in psychiatry will depend not only on technological advancements but also on how well these ethical, privacy, and integration issues are addressed.