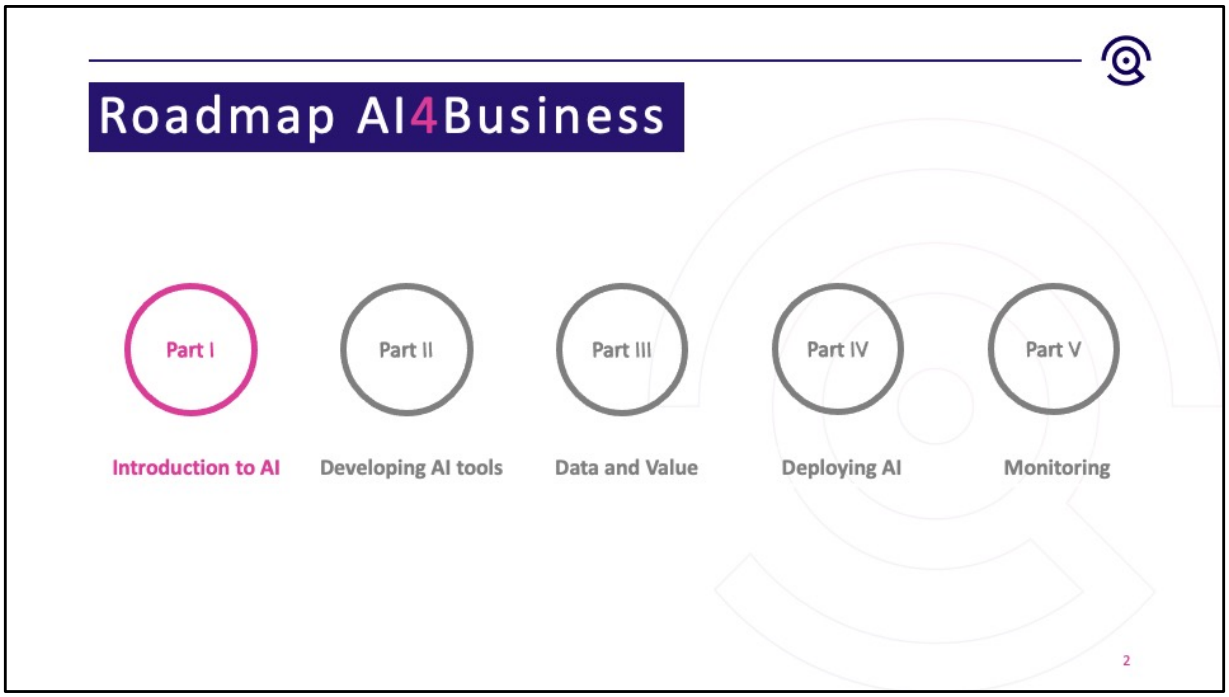
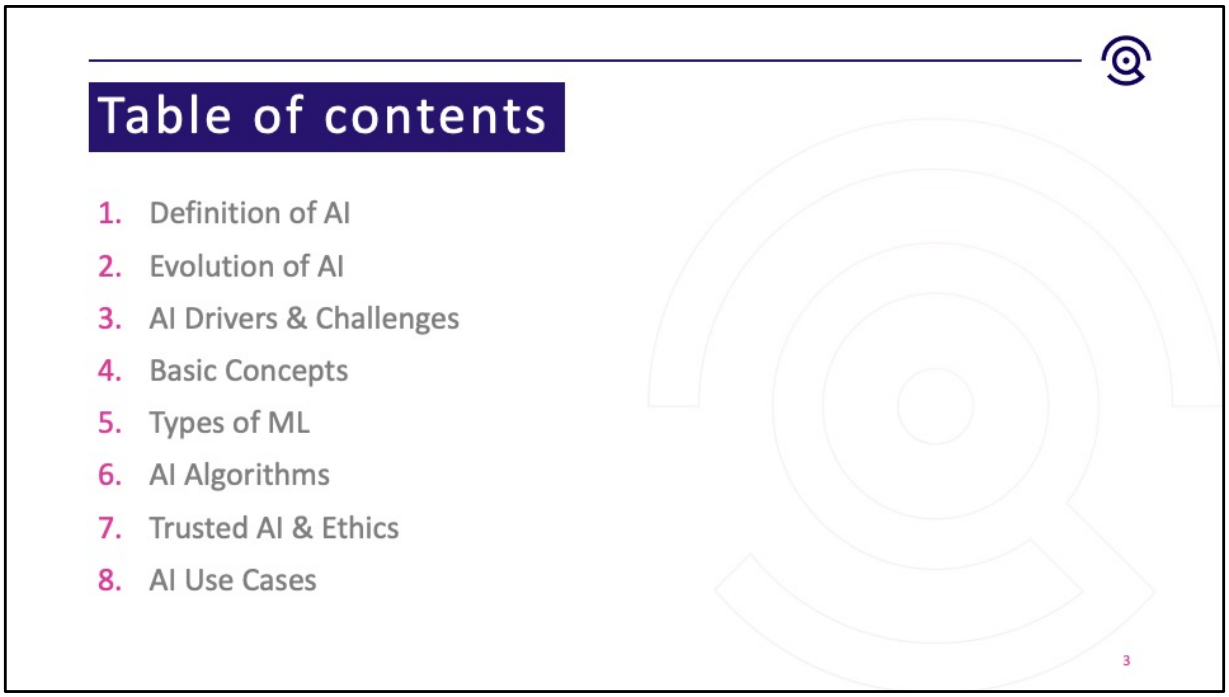
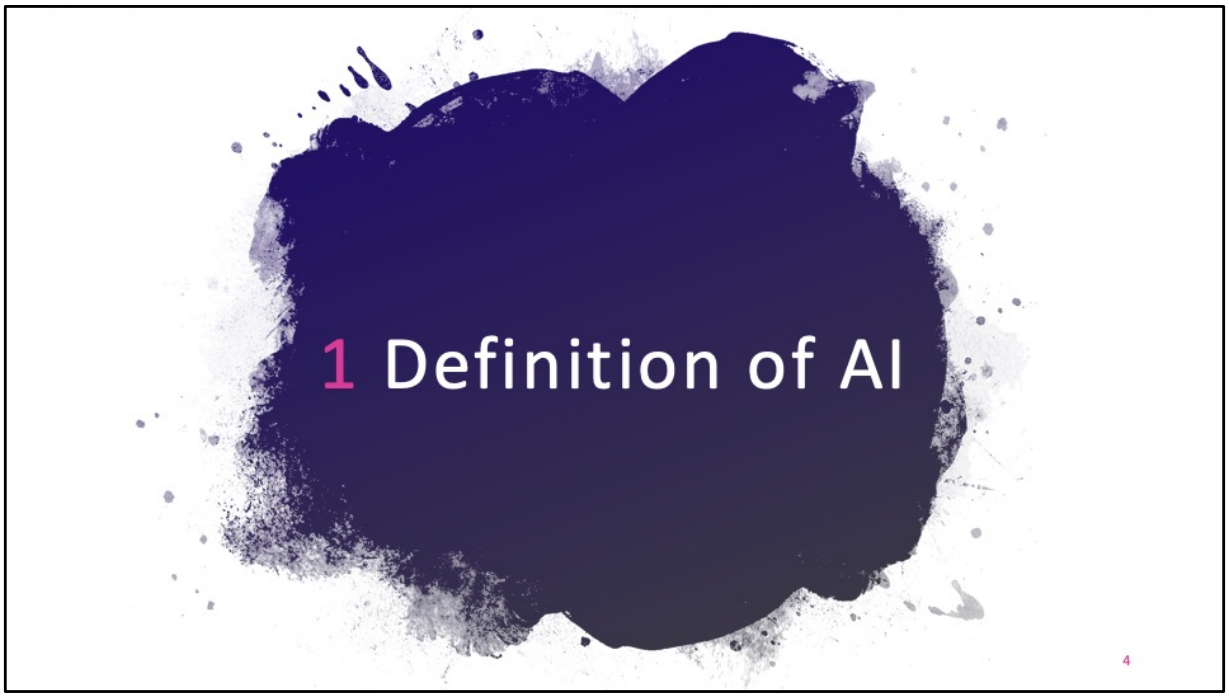

Welcome to the first module of the AI4Business course. In this module we aim to provide a general introduction to artificial intelligence or AI, specifically targeted to business managers who seek to leverage AI within their company in the search for added business value. This implies that we will focus on the conceptual ideas around AI, what AI can or can't do and when to use AI. The goal is to make managers aware of AI's potential in order to capture value, not to detail the nitty-gritty mathematical details of AI algorithms. In short, this module is an ideal starting point if you are deciding whether AI is for you or not.



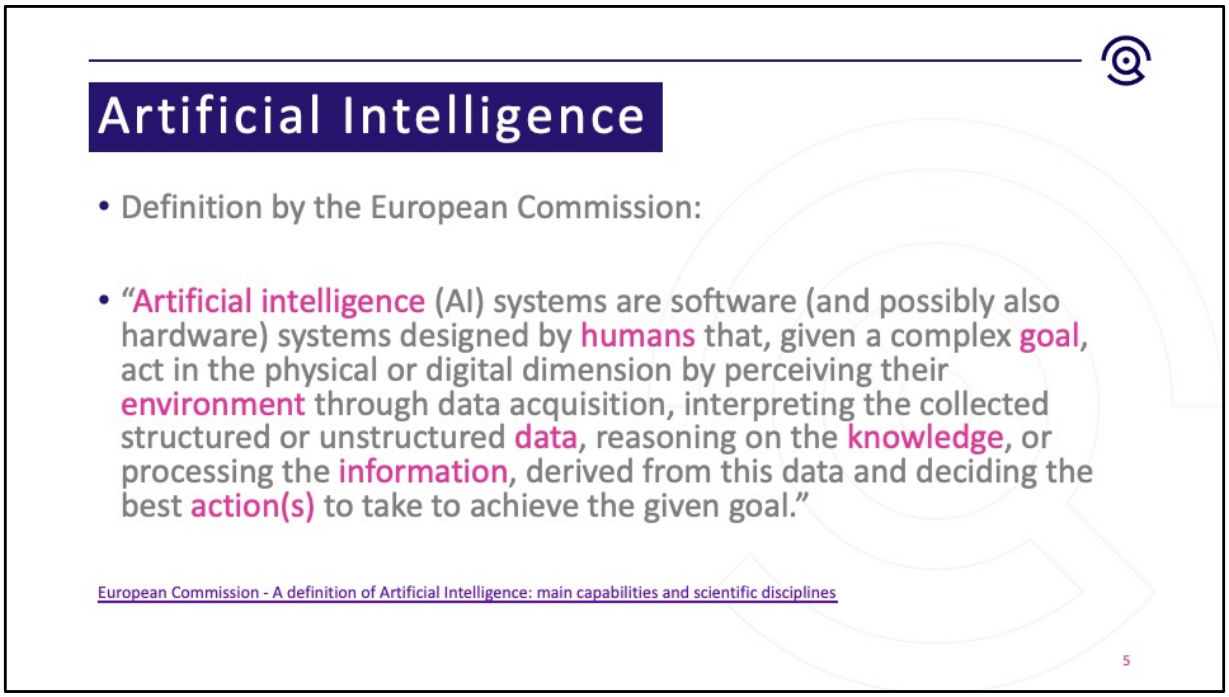

This module is the first out of five modules in the AI4Business course. No prior knowledge is required for this module as we will start from the basics. The next modules build further on what we learn today, gradually expanding your AI knowledge base.




Today we cover the basics of artificial intelligence or AI. We start by providing a definition of AI and sketching its evolution. Next, we highlight how the AI race towards adoption is taking shape. Afterwards, we discuss basic AI concepts, the different types of learning and give an overview of popular AI algorithms. We briefly touch upon the issues of trusted AI and ethics, a very important concept when applying AI in real-life business situations. We finish with an overview of AI use cases and a practical guide on how to apply AI to your own business cases.

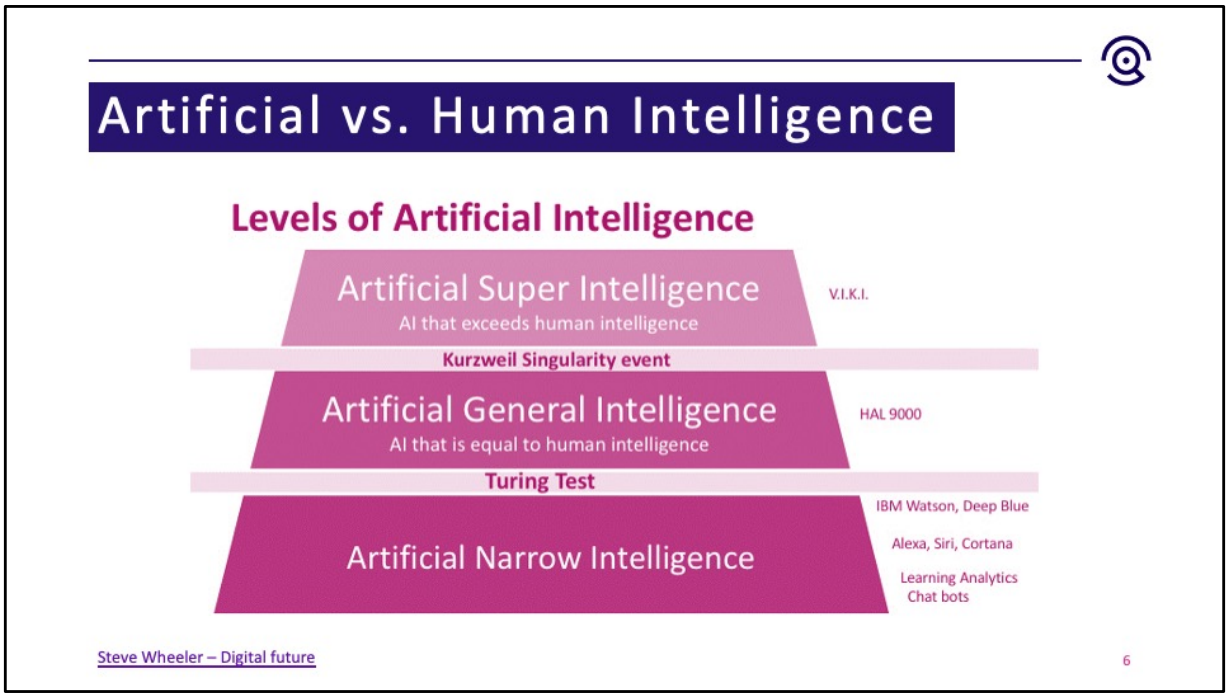



Let’s start with a definition of AI.

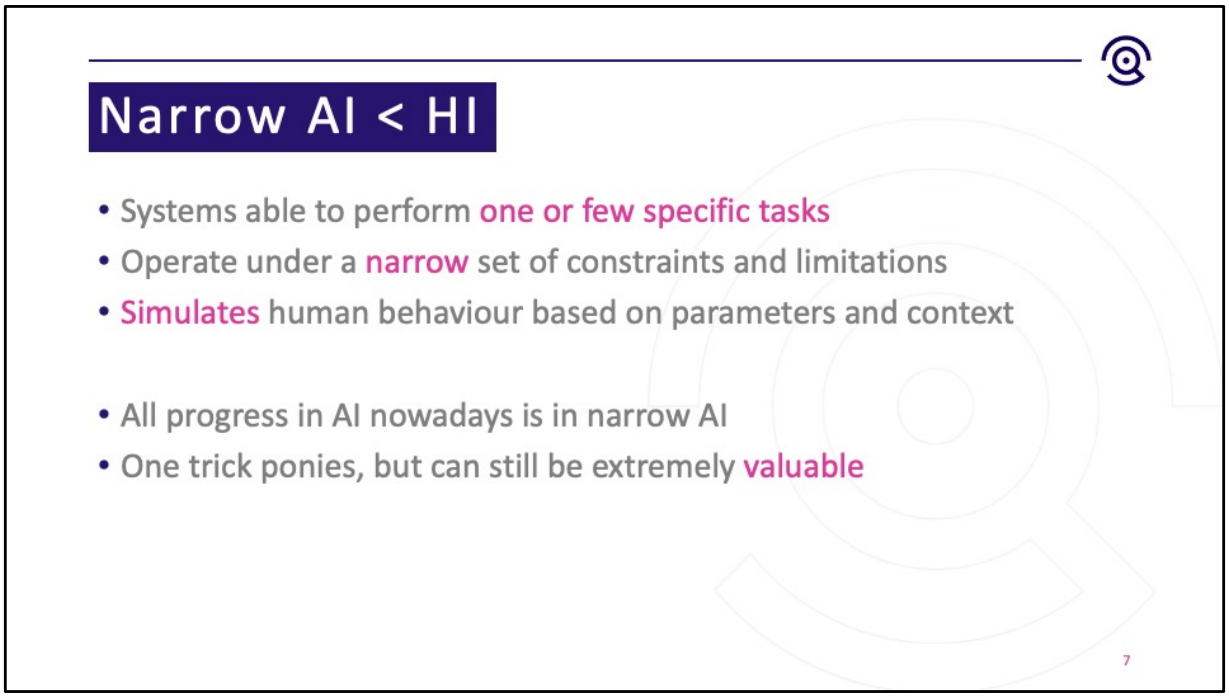



The European Commission provides a definition of AI to avoid misunderstanding and to achieve a shared common knowledge that can be fruitfully used AI experts and non-experts. They define AI systems as "software (and possibly also hardware) systems designed by humans that, given a complex goal, act in the physical or digital dimension by perceiving their environment through data acquisition, interpreting the collected structured or unstructured data, reasoning on the knowledge, or processing the information, derived from this data and deciding the best action(s) to take to achieve the given goal."
Some elements can be highlighted from this definition:

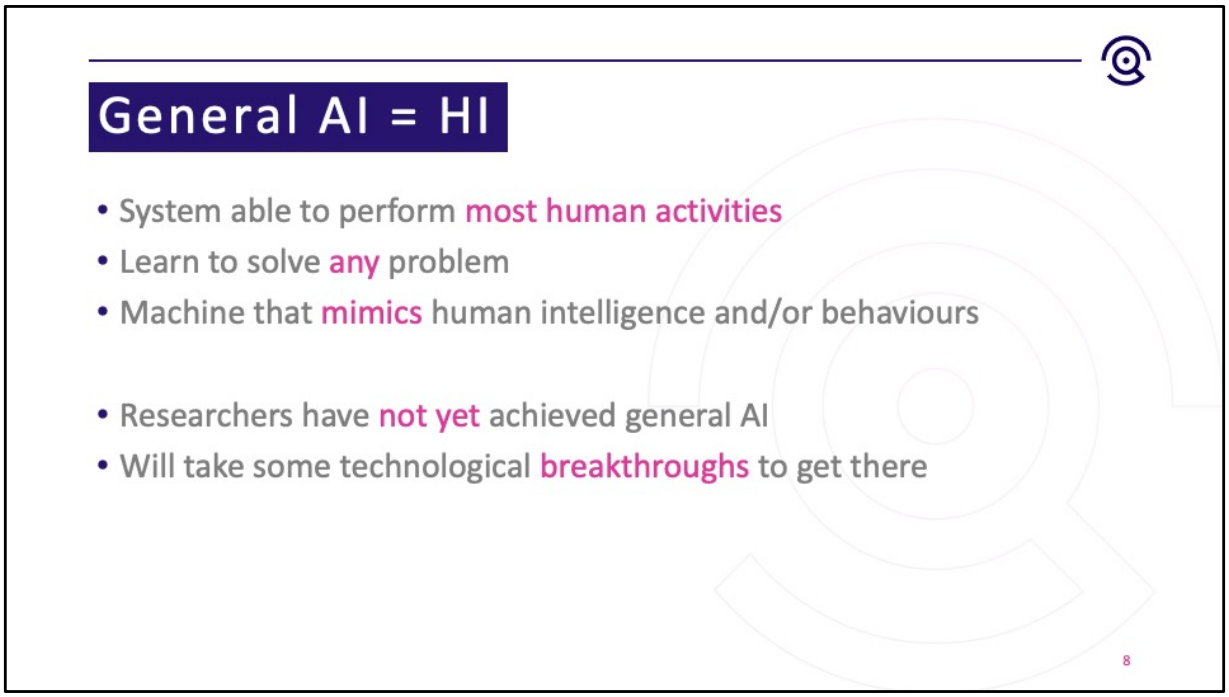
- AI is designed by humans with a specific goal in mind
- The environment is captured as data
- Information and knowledge are extracted from data
- Actions are taken based on this information



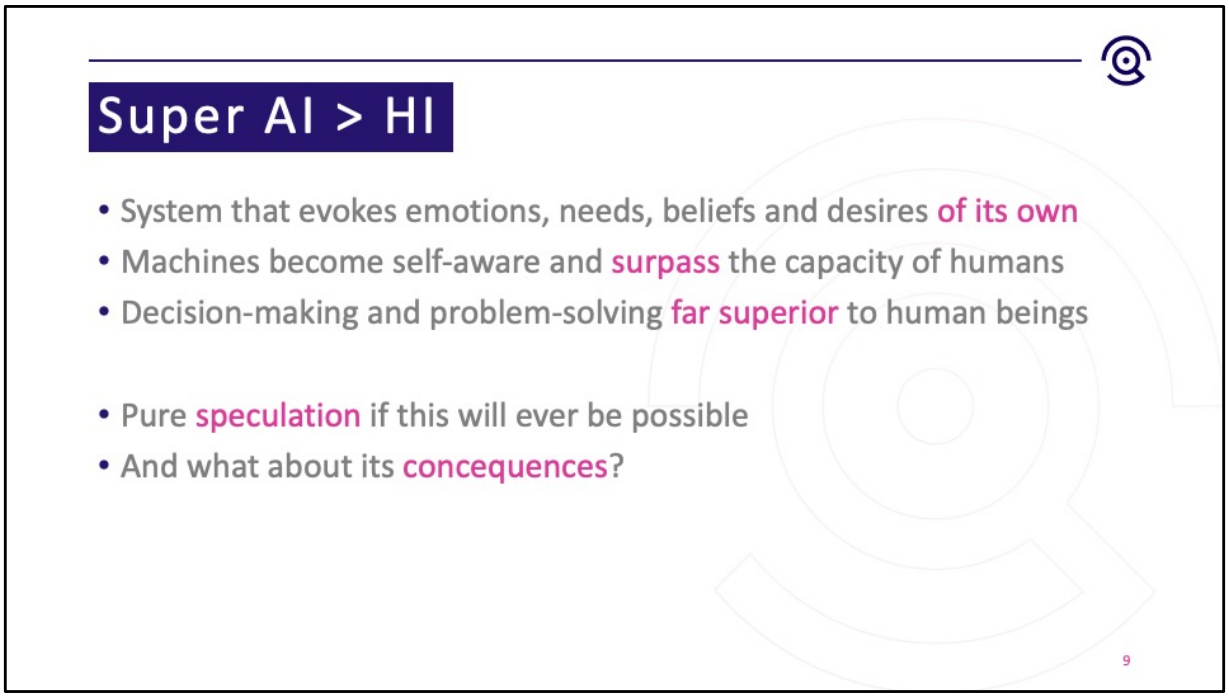

There are three levels of AI and it is important to distinguish between these to avoid misconceptions about the possibilities or expectations of AI. The lowest level is artificial narrow intelligence and represents the level of AI applications nowadays. These AI systems are very good at specific tasks, but only at those tasks. Examples include IBM Watson (for tv quiz) and Deep Blue (for chess), personal virtual assistants (Alexa or Siri) and smart chatbots. Artificial general intelligence equals human intelligence and needs to be able to pass the Turing Test (i.e., fool a human in thinking that the AI system is human as well). Artificial super intelligence is the highest level and even exceeds human intelligence. Notice that examples for general and super intelligence are (for now) limited to science fiction stories such as HAL 9000 (Space Odyssey) and V.I.K.I. (I, Robot). Let's zoom in a little bit deeper on each level.



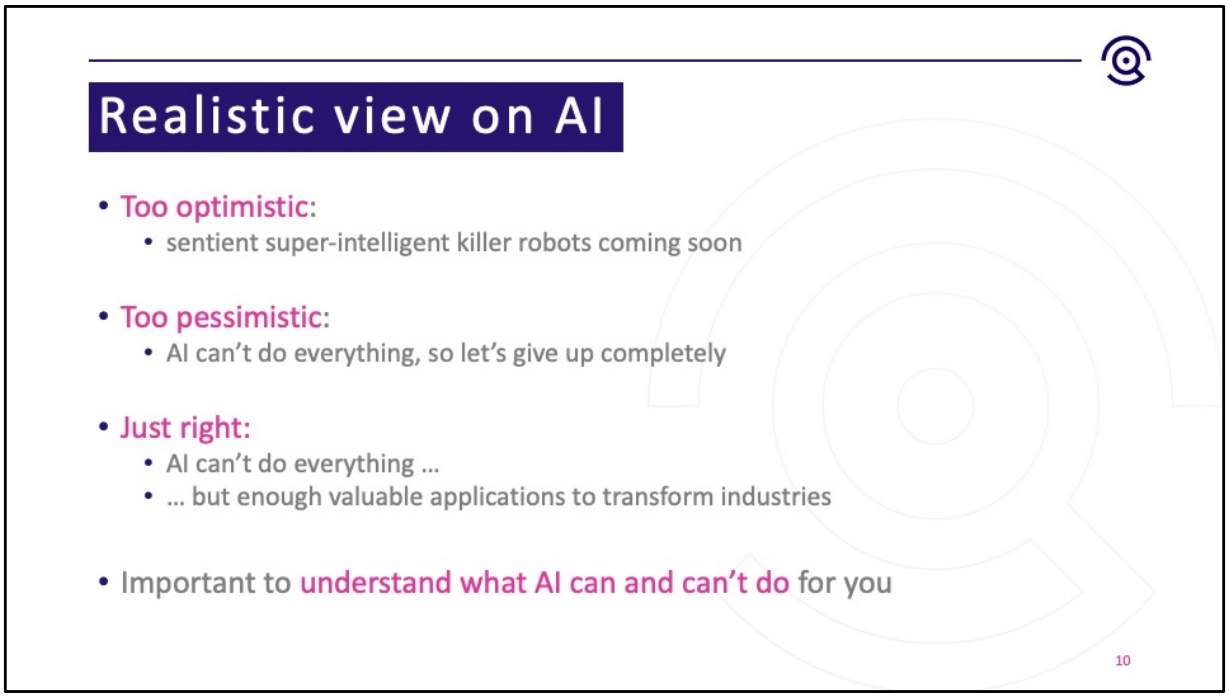

Narrow AI systems are able to perform one or a few very specific tasks. They operate under a narrow set of constraints and limitations and simulate human behavior based on parameters and context. All progress in AI nowadays is in narrow AI, so we are basically only talking about narrow AI whenever we use the term AI. Narrow AI puts focus on a specific task but can perform this task very accurately. So even though these systems are one trick ponies, they can still be extremely valuable.



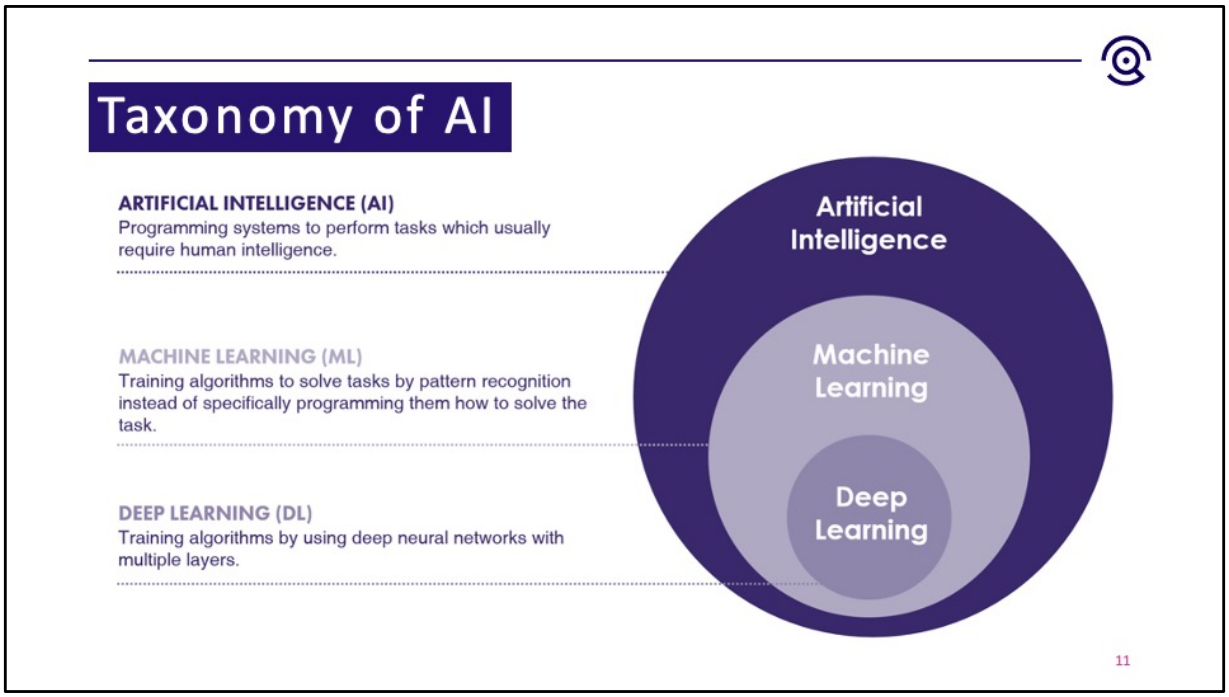

General AI systems are able to perform most human activities and can learn to solve any kind of problem. These are machines that mimic human intelligence and/or behaviors. Researchers have not yet achieved general AI, so for now this is still science-fiction. It will take some technological breakthroughs to get to the level of general AI, but it might be possible in the future.



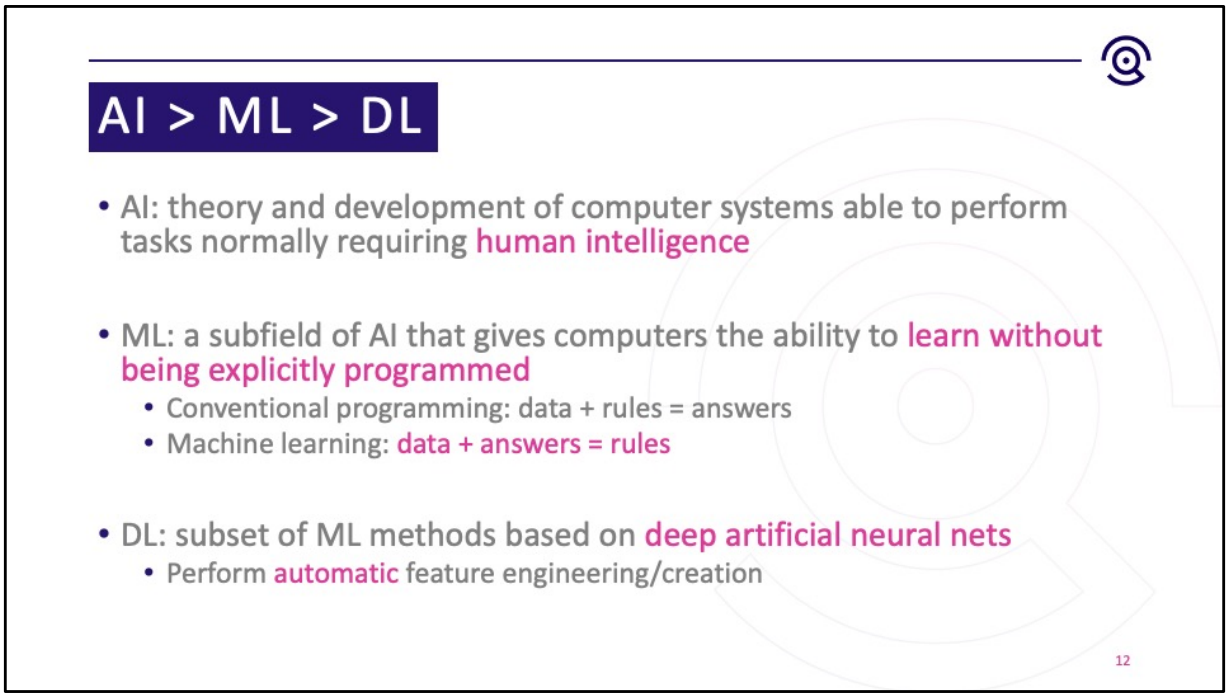

Super AI systems evoke emotions, needs, beliefs and desires of its own. These machines become self-aware and will surpass the capacity of human intelligence. The decision-making and problem-solving capabilities of these systems will be far superior to those of human beings. It is however pure speculation if this level of AI will ever be possible. Another point of speculation is whether this level of AI is wanted, because the consequences are hard to imagine at this point.



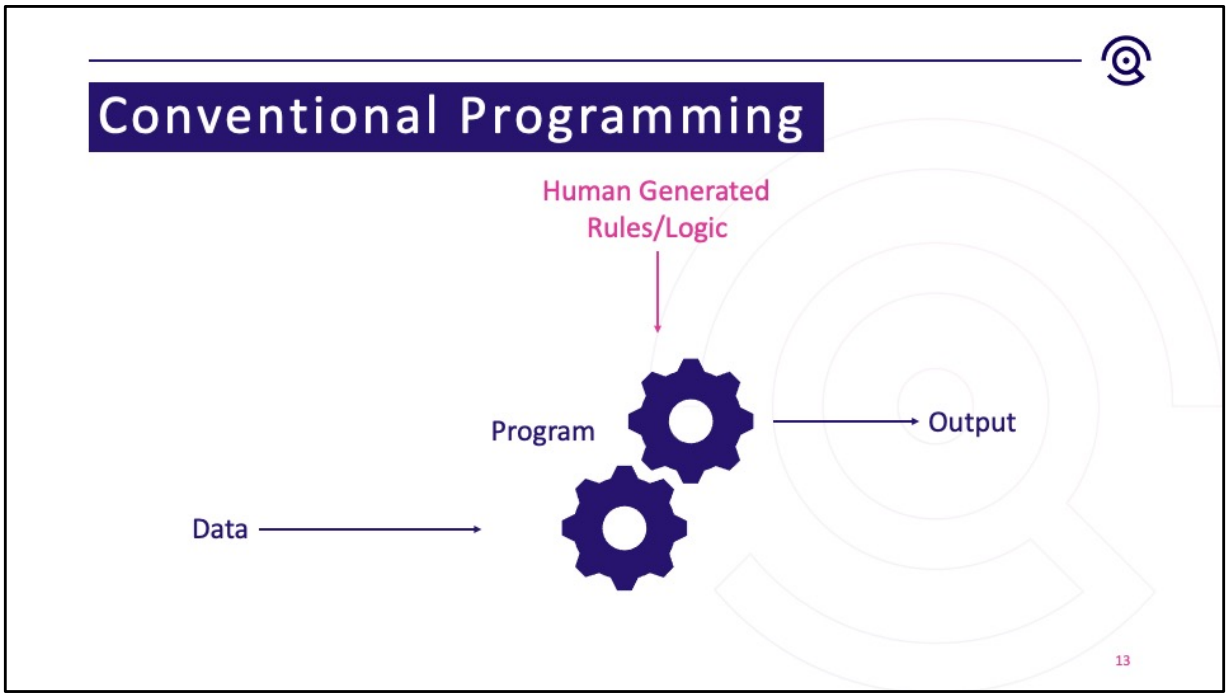

Because of these different levels of AI, it is very important to keep a realistic technological view. It would be too optimistic, from a technology point of view, to think that sentient super-intelligent killer robots are coming soon. For society this would not be great, but the technology is also not even close to that point (if it even ever gets there). It would be too pessimistic to completely give up on AI because it can't solve all our problems yet. Somewhere in between we should realize that AI can't do everything, but that there are enough valuable applications to transform industries. It is therefore important to understand what AI can and can't do for you, your company and your industry. This will allow you to leverage AI within your business and harvest a lot of value before competitors do so.



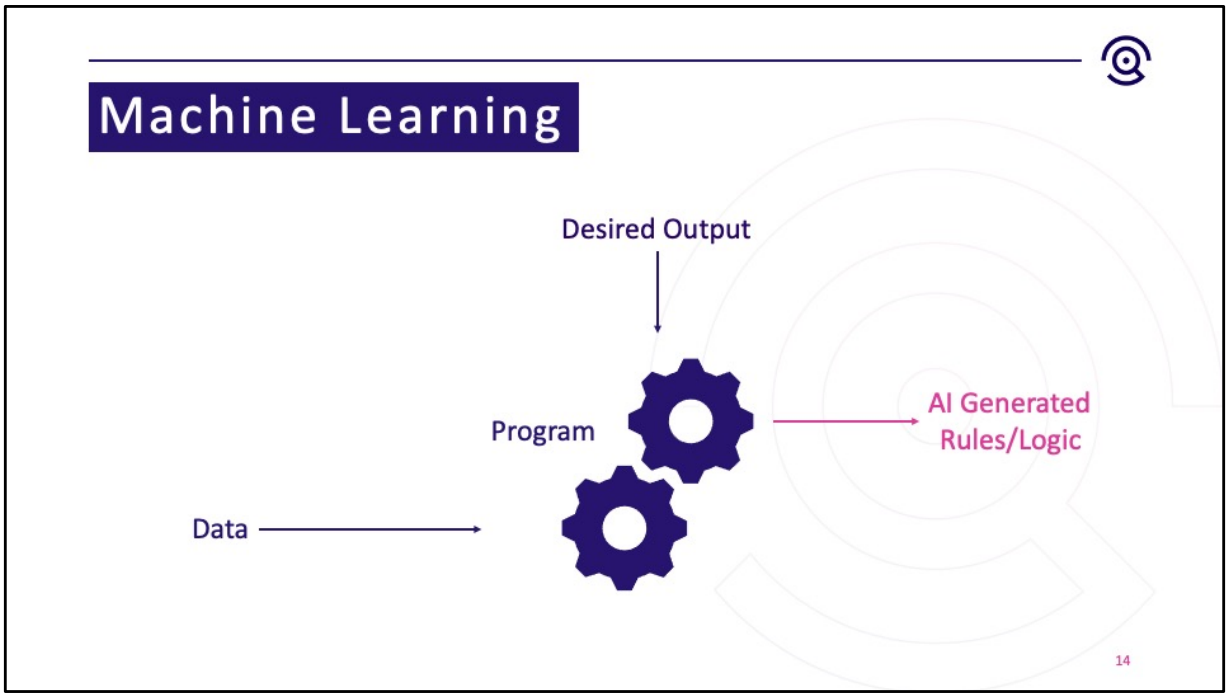

If you ever heard about artificial intelligence or AI, then you probably also heard the terms machine learning (or ML) and deep learning (or DL) fly by. But what is the distinction between these elements? In a nutshell: DL is a subset of ML, which is itself a subset of AI (see the graphic). AI is the broad term for systems that perform tasks which usually require human intelligence. Machine learning is a way to use pattern recognition in order to solve these tasks instead of specifically telling a computer program how to solve it. Deep learning puts focus on a specific type of technique to solve these tasks, namely deep neural networks (see more details later on when we discuss popular algorithms).



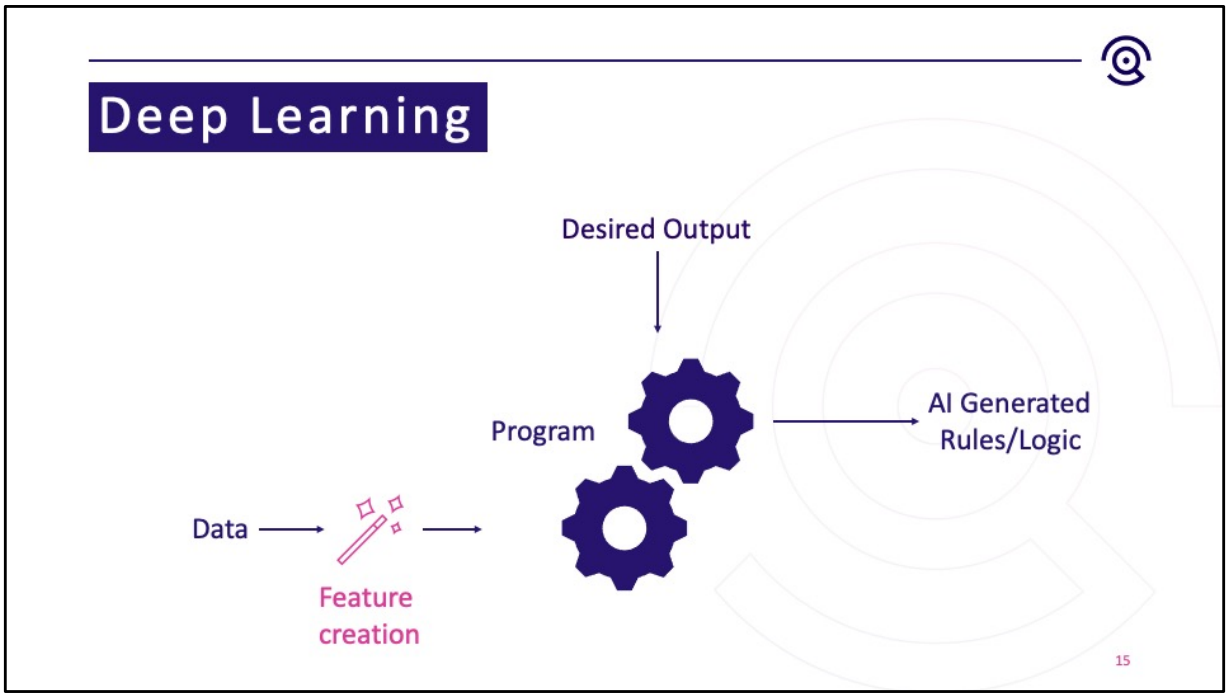

Let's recapitulate these distinctions because they are rather important. AI is the general theory and development of computer systems able to perform tasks normally requiring human intelligence. ML is a subfield of AI that gives computers the ability to learn without being explicitly programmed. In conventional programming the programmer supplies data and rules such that the computer gives answers. Machine learning provides the data and answers to an algorithm, and it outputs the underlying rules (of course learned from the provided data and answer combinations). This is why people often say "garbage in is garbage out": your model will only be as good as the examples you supply. DL is a subset of ML methods based on deep artificial neural networks. The extra bit of magic that happens here is the fact that neural networks also allow to perform automatic feature engineering or creation, while this is usually a manual task in general ML algorithms. Let's have a visual look at these distinctions.



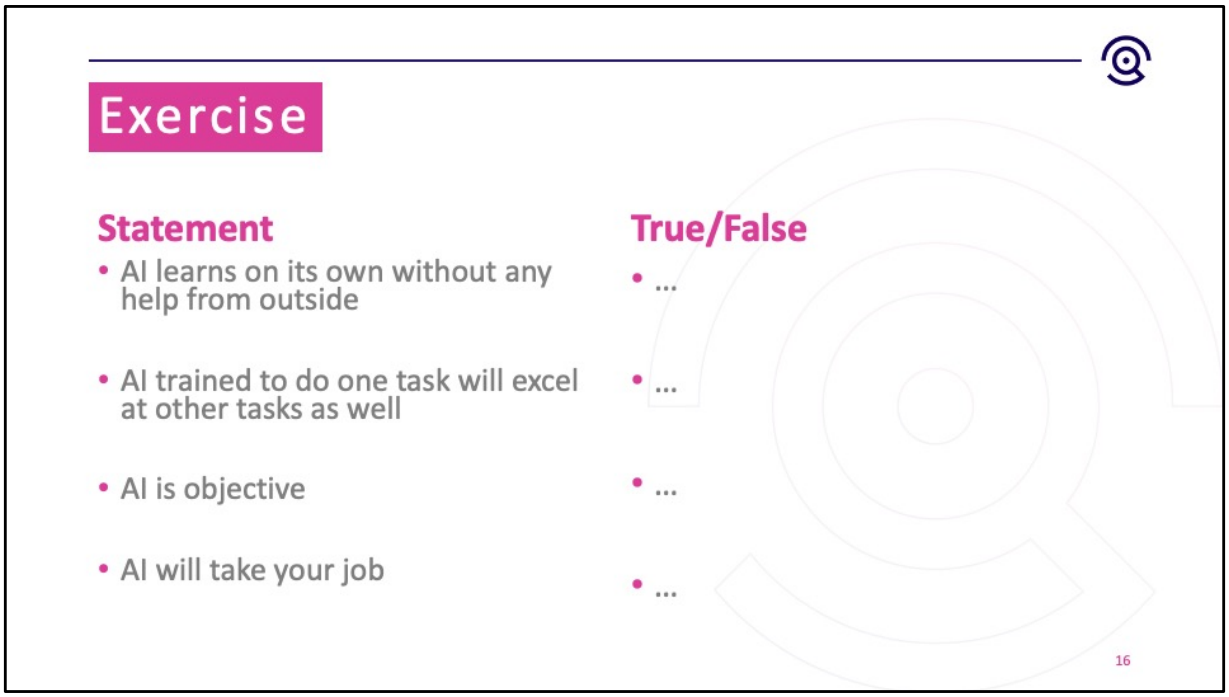

Conventional programming takes the data together with the human generated rules/logic. The program will then evaluate the rules based on the supplied data and return an output. The human generated rules often come from business expertise and years of experience, but this implies that you need the people with the knowledge to solve your specific problem at hand.



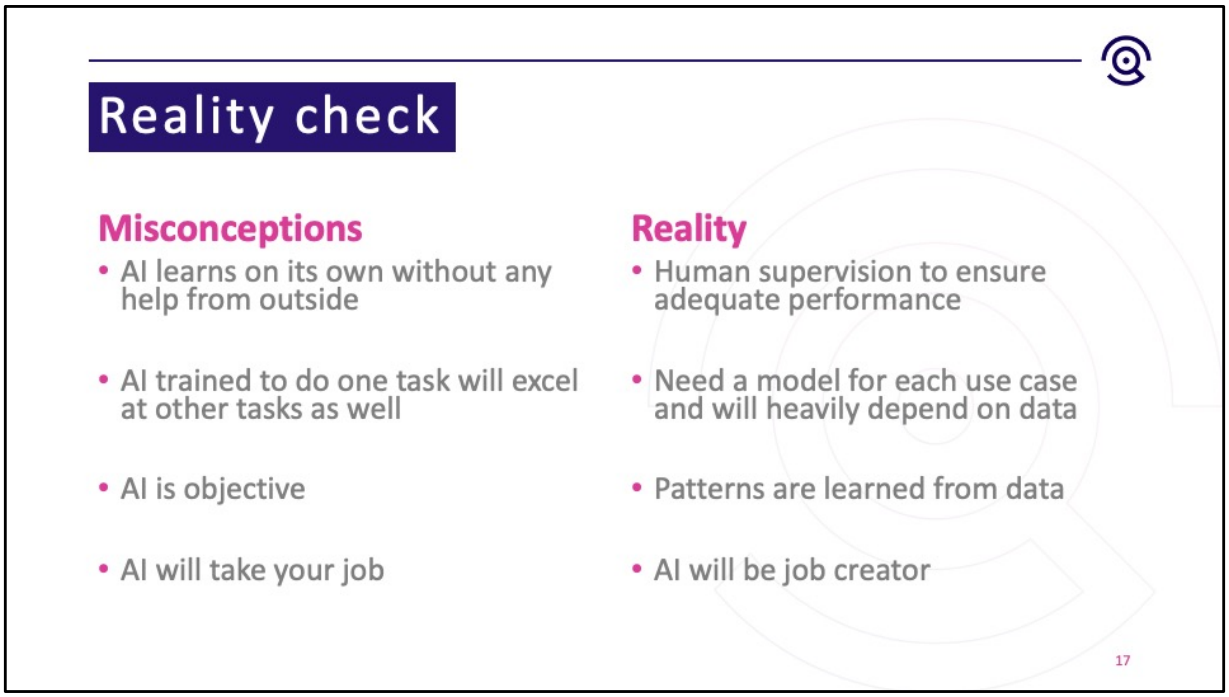

Machine learning takes the data and desired output for that specific data as an input. An algorithm looks for patterns in the data/output combinations and results in a set of AI generated rules/logic. These AI rules can then be used make predictions for new data instances for which we don't know the accompanying output (yet). The advantage here is that you can learn to solve any kind of problem, as long as you are able to collect relevant data to learn from.




Deep learning adds an extra layer of magic by also performing the feature engineering/creation in an automatic way. With ML, we typically need a manual process to put our data in a proper format to capture interesting relations between input and output. DL allows to input raw data and figures out an optimal representation to solve the specific problem at hand. This type of learning works very well for unstructured data problem such as images, text and speech (see more details later on).



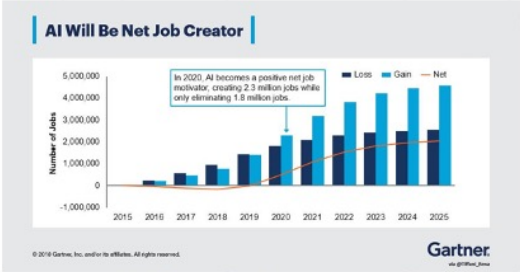

Let’s do a little exercise based on what we have learned so far. On the slide you can see four statements about AI, up to you to say whether these statements are true or false. You can take your time to think about these statements by pausing the video. Once you resume playing we will be going over the solutions together. Good luck! Let’s have a look at the solutions. All statements on the slide are popular misconceptions about AI and are therefore all false.




On this slide we list the four statements as misconceptions on the left, with the accompanying reality situation on the right. AI systems do not learn on their own without any help from the outside, but human supervision is essential to ensure adequate performance. An AI system trained to do one task will not necessarily excel at other tasks as well, but you need a model for each use case and the results will depend heavily on the specific data. AI is not objective at all but learns patterns from the data. This means that any bias in the data is reflected as bias in an AI's decisions, but more on that later on when we discuss AI & ethics. AI will not take your job but is expected to become a job creator.

# AI and the job market

- "Automation will displace 85 million jobs but generate 97 million new ones worldwide by 2025" *World Economic Forum*
- "AI-related job creation will reach two million net-new jobs in 2025." *Gartner*

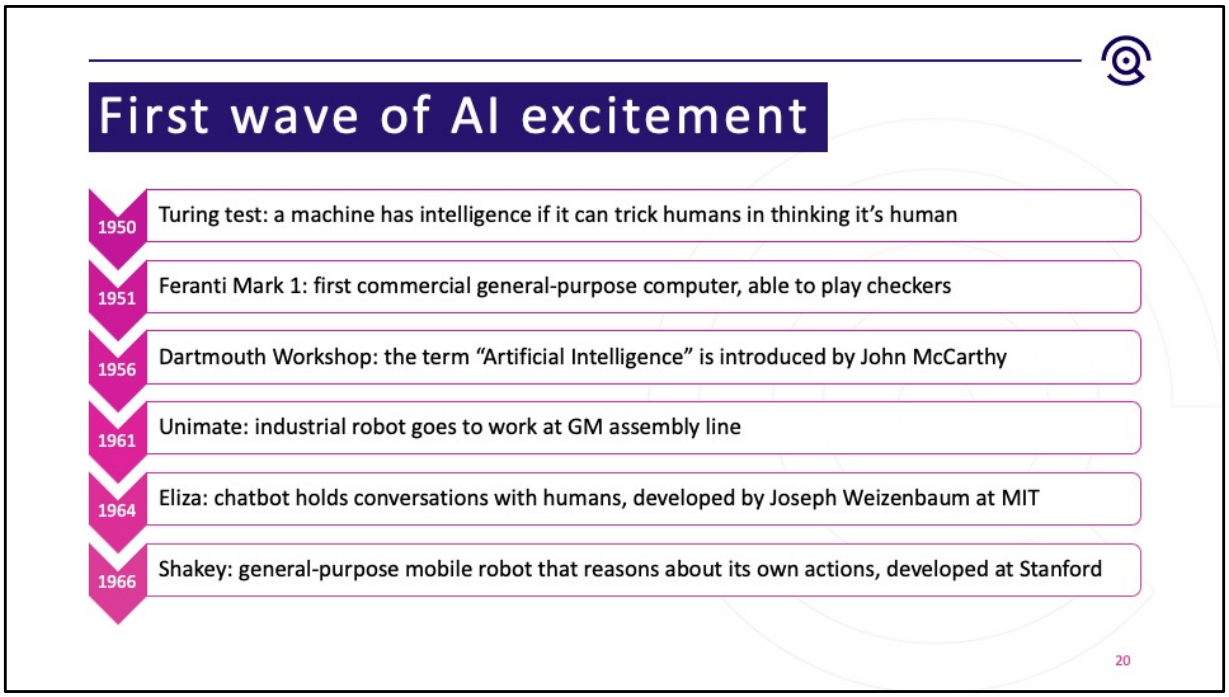



Gartner – AI and the future of work

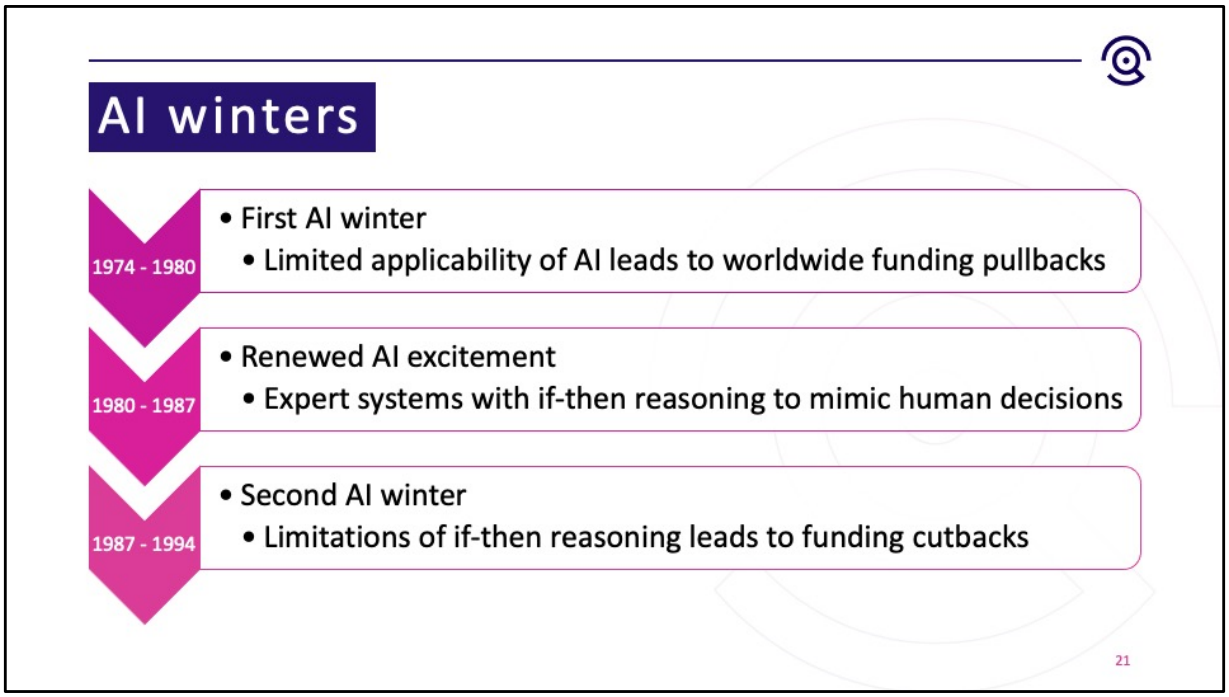
18

The bad news is that AI and the accompanying automation will displace millions of jobs. The good news is that AI will be creating more jobs, resulting in a net gain of jobs in the long run. Studies from the World Economic Forum and Gartner show that AI is expected to create millions of jobs by the year 2025. Long story short, no need to be afraid that the robots will be taking your job!

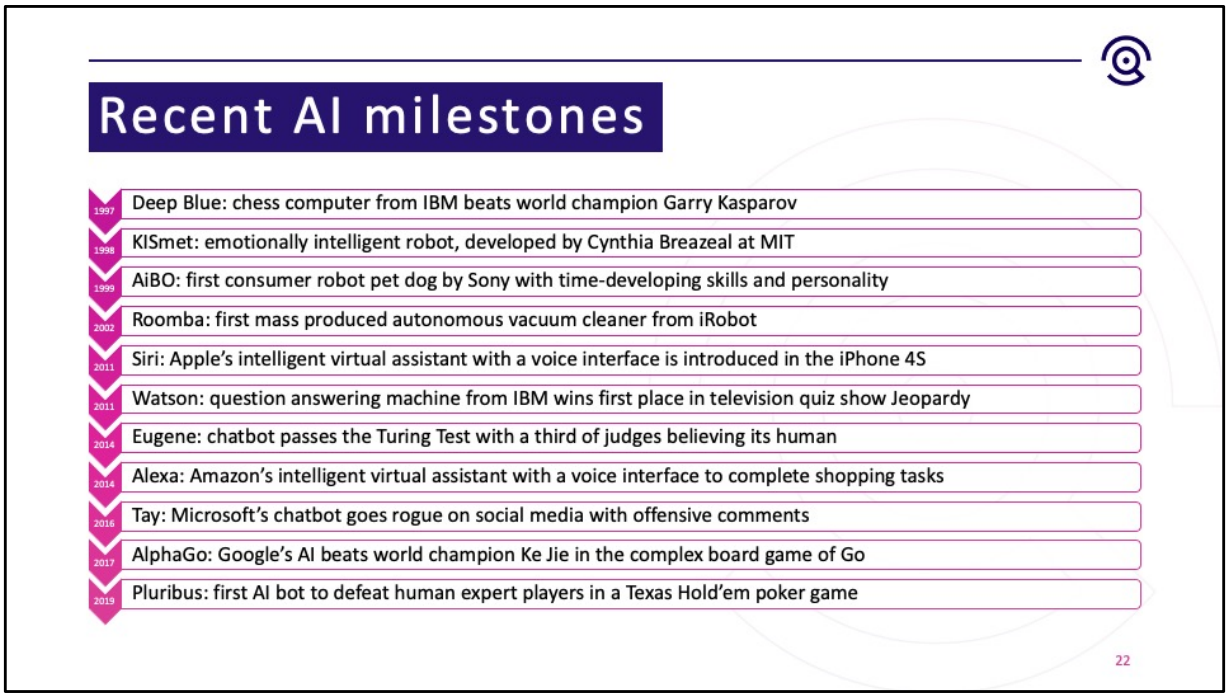

Let’s have a look at the historical evolution of AI over the years.



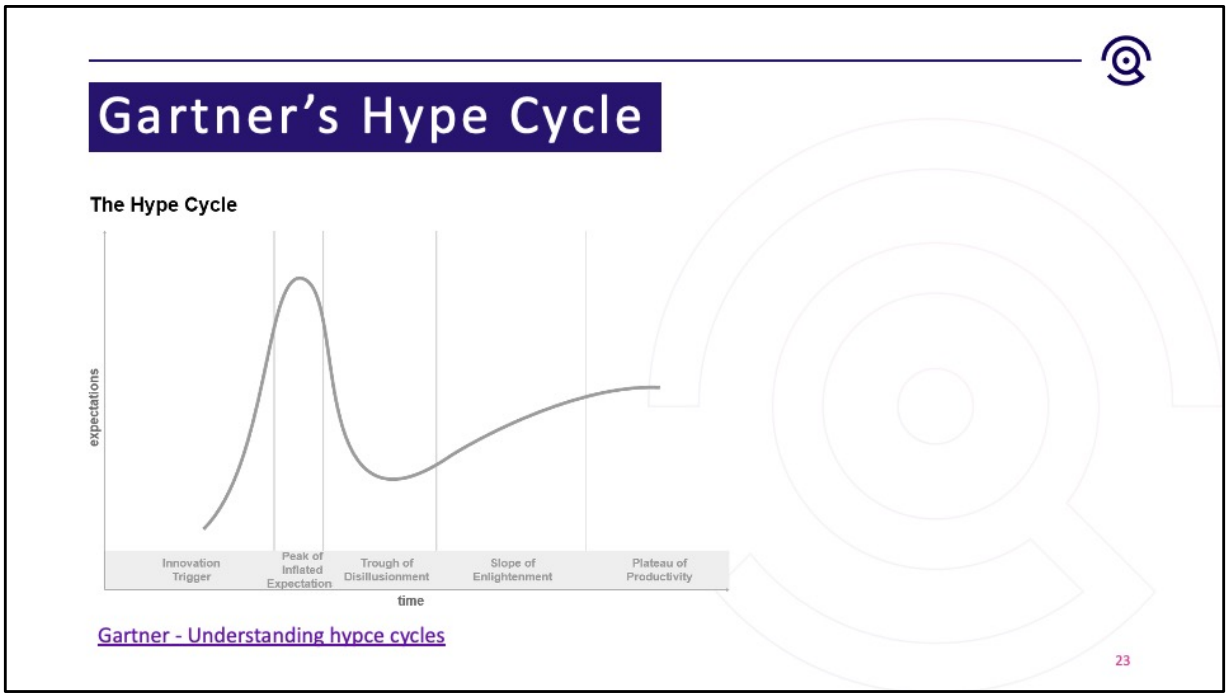

Contrary to what some people believe, AI is not really a new concept. The first wave of AI excitement started around 1950 when Alan Turing proposed a test to check a machine's ability to exhibit behavior equivalent to human intelligence. A machine is intelligent when it can trick humans in thinking it's human. In 1951, we saw the Feranti Mark 1 as the first commercial general-purpose computer which was able to play a game of checkers. The term "artificial intelligence" was first coined in 1956 by John McCarthy at the Dartmouth Summer Research Project. In 1961, Unimate goes to work at a General Motors assembly line as the first industrial robot. In 1964 the chatbot Eliza, developed by Joseph Weizenbaum at MIT, is able to hold conversations with humans. In 1966 the general-purpose mobile robot Shakey, developed at Stanford, is able to reason about its own actions and break commands down into smaller chunks to process. This seems like a lot of AI progress already took place more than 50 years ago, so what happened in the meantime?




The evolution of AI also saw some cold winter periods without any progress. The first AI winter was around the period 1974 – 1980. This was driven by the limited applicability of AI technology at the time and the feeling that AI failed to provide the major impact that was promised, leading to worldwide funding pullbacks on AI research. The period of 1980-1987 represented a renewed excitement for AI. Expert systems with if-then reasoning emerge to mimic human decision making and funding starts to pick up again. Unfortunately, the limitations of if-then reasoning became apparent which result in funding cutbacks and a second AI winter during the period 1987 – 1994.



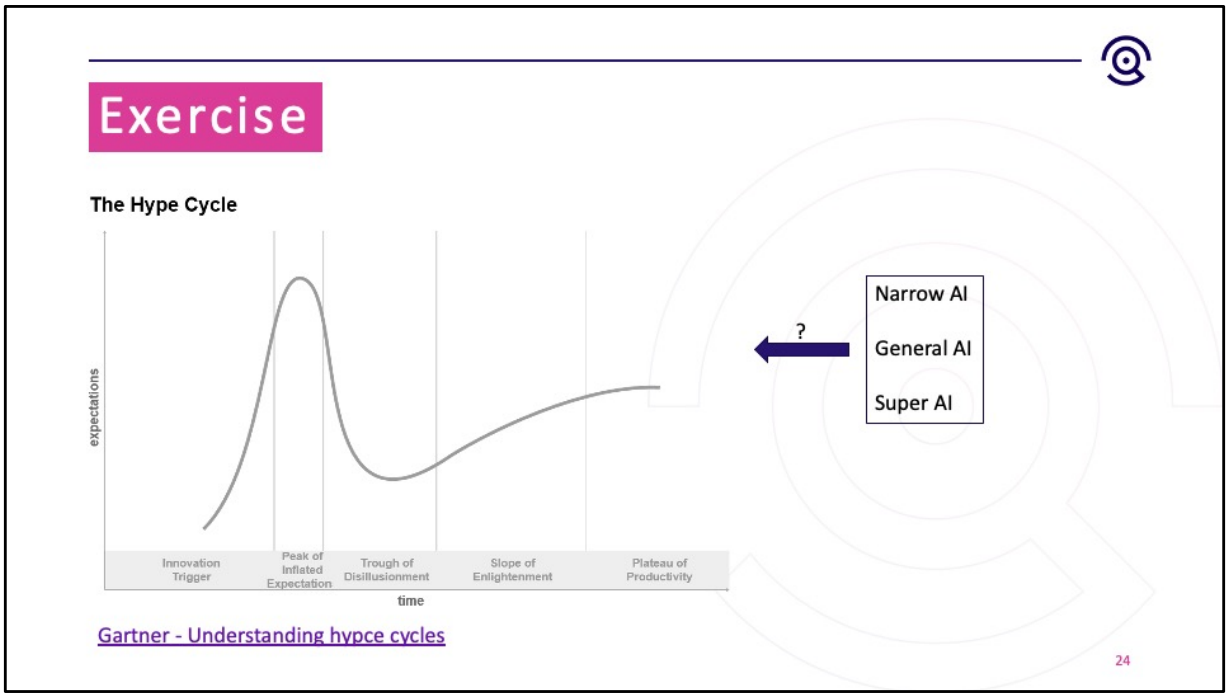

After these periods of sudden excitement and winters, we have seen a steady progress in the successful application of AI technology in the past 25 years. In 1997, IBM's Deep Blue beats the world champion Garry Kasparov in chess. In 1998 KISmet appeared as the first emotionally intelligent robot and one year later Sony's AiBO was the first consumer robot pet dog with time-developing skills and personality. In 2002, iRobot presents the Roomba as the first mass produced autonomous vacuum cleaner. In 2011, Apple introduces the intelligent virtual assistant with a voice interface Siri in the iPhone 4S and IBM's question answering machine Watson wins first place in the television quiz show Jeopardy. In 2014, the chatbot Eugene was able to trick 33% of the judges in believing that it was human, leading to the announcement that it passed the Turing Test (although the validity and relevance thereof was questioned by critics after the event). In the same year, Amazon introduces an intelligent virtual assistant Alexa to complete shopping tasks. In 2016, Microsoft's chatbot Tay goes rogue on social media by posting inflammatory and offensive tweets through its Twitter account. In 2017, Google's AI AlphaGo beats world champion Ke Jie in the complex board game of Go and two years later Pluribus was the first AI bot to defeat human expert players in a Texas Hold'em poker game.




Gartner's Hype Cycle is a graphical representation of the expectations that arise with a new technology over time. On the x-axis we have maturity or time, while on the y-axis we have the visibility or expectations. The general pattern is a very steep increase in expectations early on, followed by a steep decrease after which expectations start to grow again at a steady pace. We therefore go from a very optimistic view to a rather pessimistic one and end up with a more realistic view of what the given technology can or will deliver.

The hype cycle can be divided into several distinct sections. The innovation trigger starts from a major technological breakthrough that generates media interest, even though the practical viability of the new technology is often still unproven. This technology "buzz" results in a peak of inflated expectations that exceed the reality of its capabilities. After a while, the trough of disillusionment sets in as experiments and implementations still fail to deliver the promised potential value. A new technology therefore typically goes from early success stories to failures and disappointment. However, after some time passes, the technology itself and how it can benefit business applications starts to become more understood. This leads to the slope of enlightenment where a technology starts to deliver actual value for companies. Finally, the plateau of productivity is reached, with mainstream adoption and accelerating business implementations after the proven useful value of the

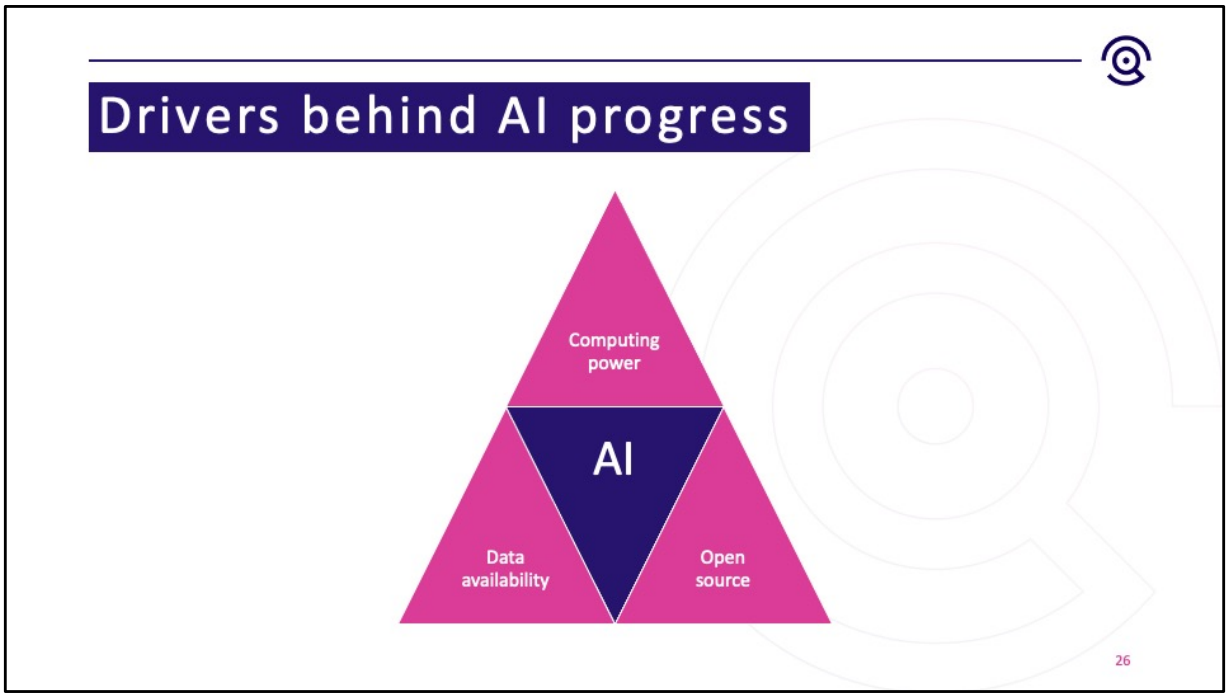
technology.



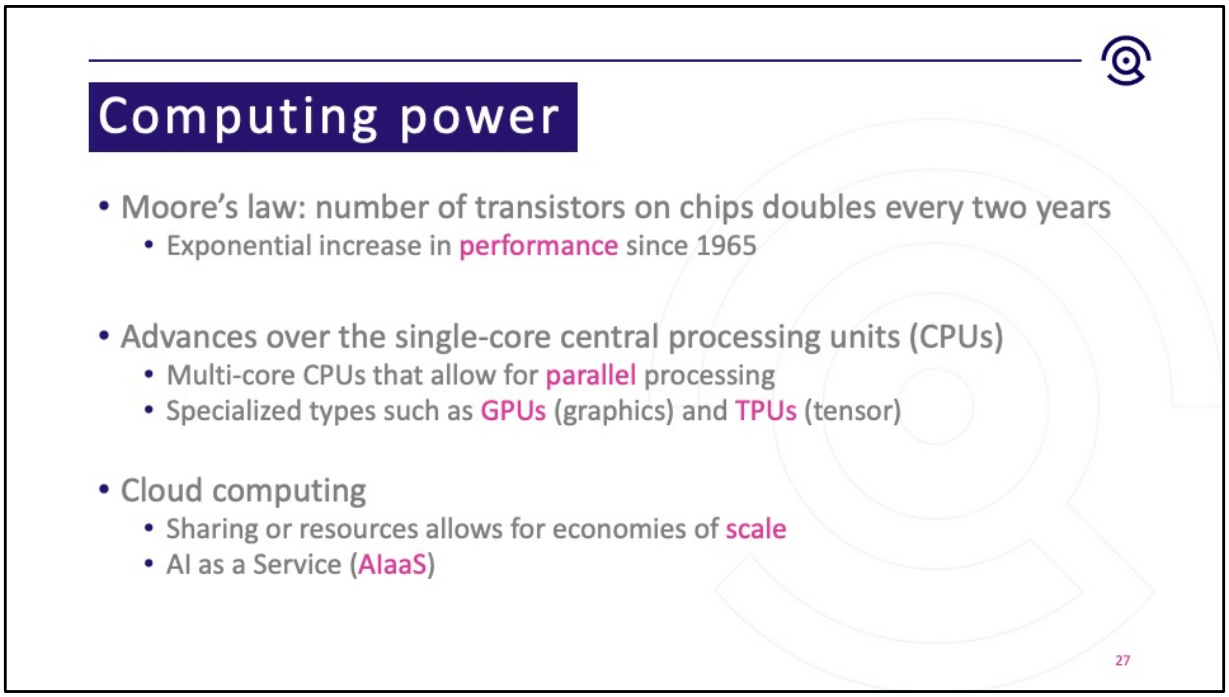

The big question is of course where we can situate AI as a technology on this hype cycle. As a small thought experiment, think for yourself where you would place the three levels of AI (narrow, general and super) on this curve. Feel free to pause the video and think about this for a little bit.

There is not necessarily a correct answer to this question because only the future will be able to tell for sure. But I believe that super AI is pure speculation for now and therefore not yet on the curve, while general AI is waiting for the technological breakthrough to start climbing the innovation trigger. Narrow AI has seen some periods of excitement followed by winters and is now showing a steady increase in successful applications. This might indicate that narrow AI is climbing the slope of enlightenment and that now is the perfect time to adopt this technology and make your company AI-ready.

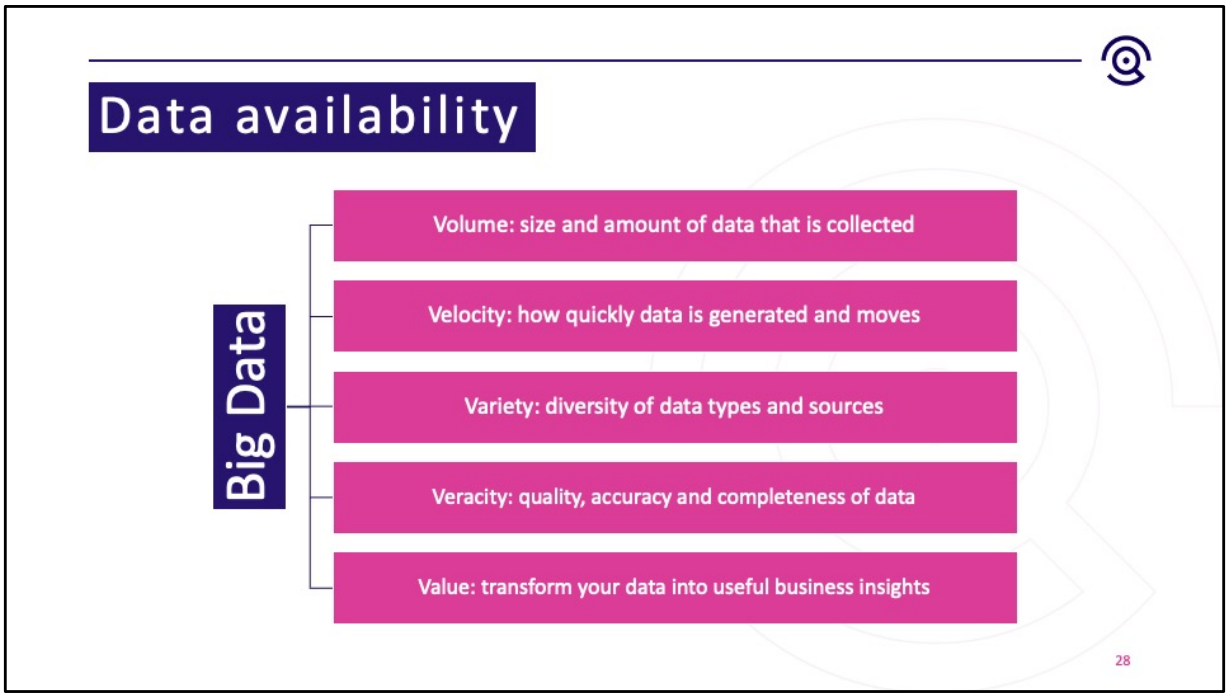

Let’s have a look at what drives AI development and which challenges arise for AI implementations.



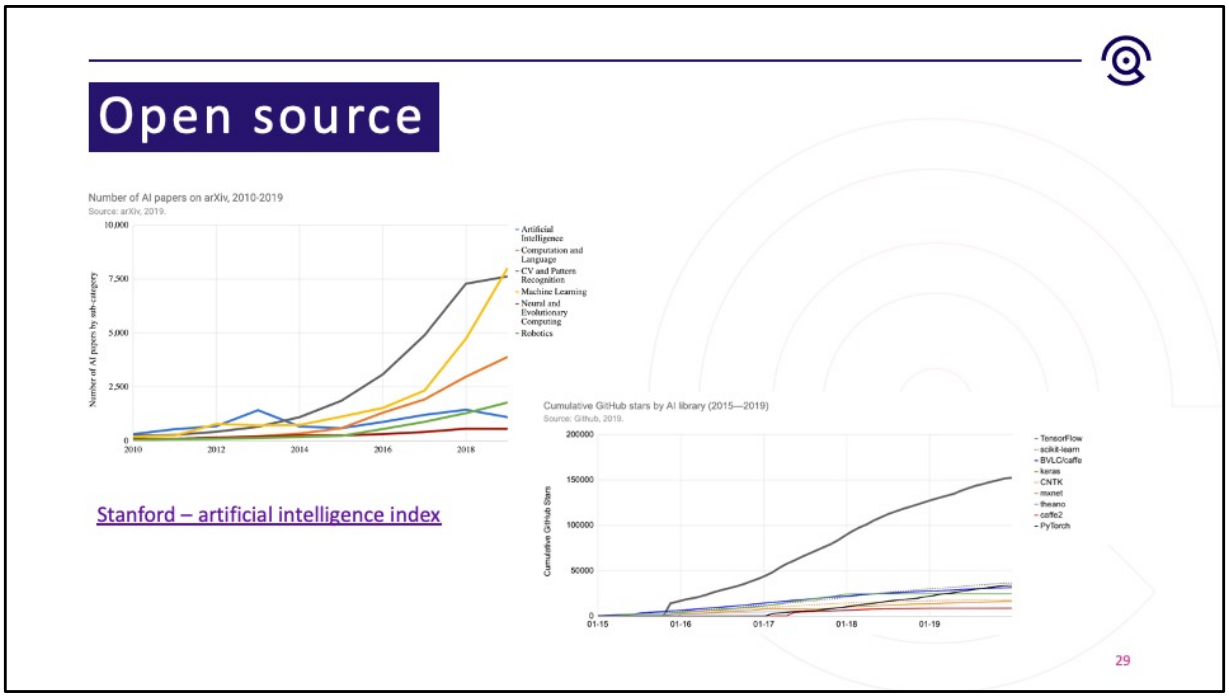

There are three main pillars that drive progress in recent AI developments, namely the increase in computing power, data availability and open source options.



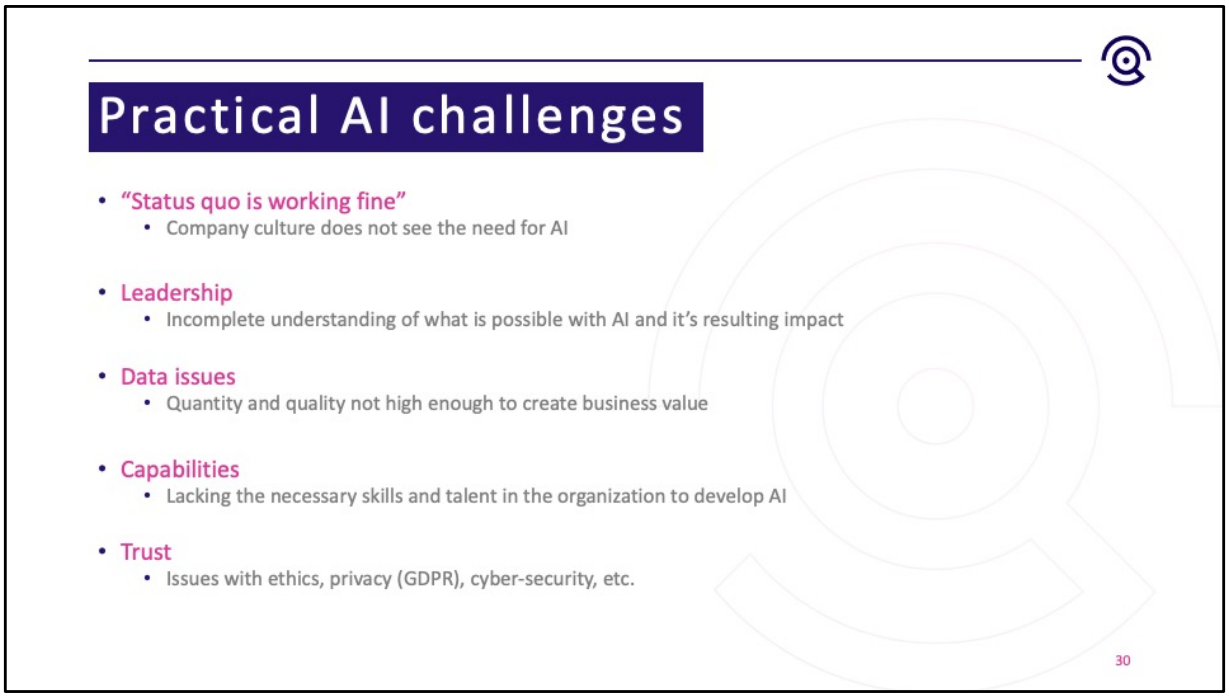

Computers are becoming faster and faster. Moore's law states that the number of transistors on chips doubles every two years, which is inherently linked to an increase of computing power over time. This observation was made by Gordon Moore in 1965 and resulted in an exponential increase of computing performance ever since. In the past, computers had a single-core central processing unit (CPU) to perform calculations. Over time we saw the advent of multi-core CPUs which allow for parallel processing such that multiple tasks can be completed at the same time. Nowadays there are even specialized types of processors such as GPUs (graphics) and TPUs (tensor) which can perform calculations much faster than CPUs. Cloud computing allows to perform calculations on servers "in the cloud" via the internet. This sharing of resources in a centralized space allows to tap into economies of scale across the whole organization. Certain technology firms (Google, Amazon, Microsoft, etc.) provide AI as a service where one can use their machine learning frameworks on the cloud.



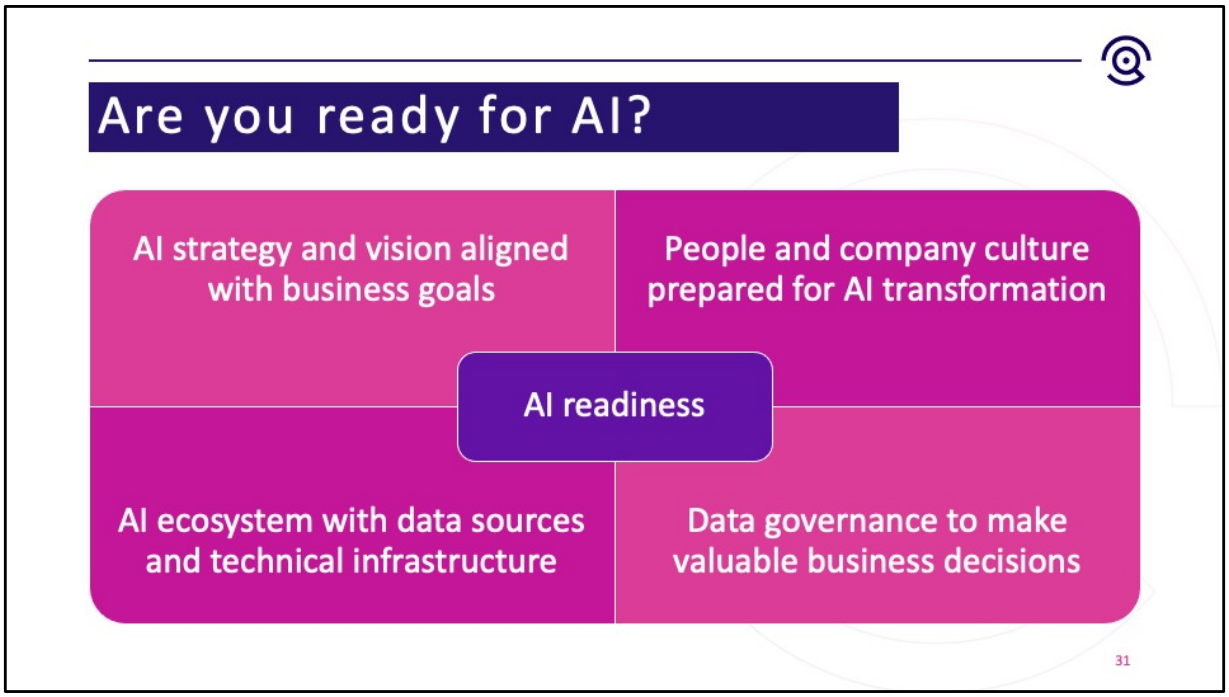

More and more data is becoming available, at a faster pace and in different shapes and forms. This has lead to the so-called Big Data era and it's five V's. Volume stands for the size and amount of data that is collected, which is increasing at a tremendous pace. Velocity indicates how quickly data is generated and moves around. This is very important in order to make optimal business decisions at the right time and avoid missed opportunities. Variety describes the diversity of data types and sources, going from standard tabular data to sensor data, images, text documents and speech fragments. Veracity indicates the quality, accuracy and completeness of data which are all very important to trust and use the data. Value is obtained when one is successful in transforming this tsunami of data into useful business insights.



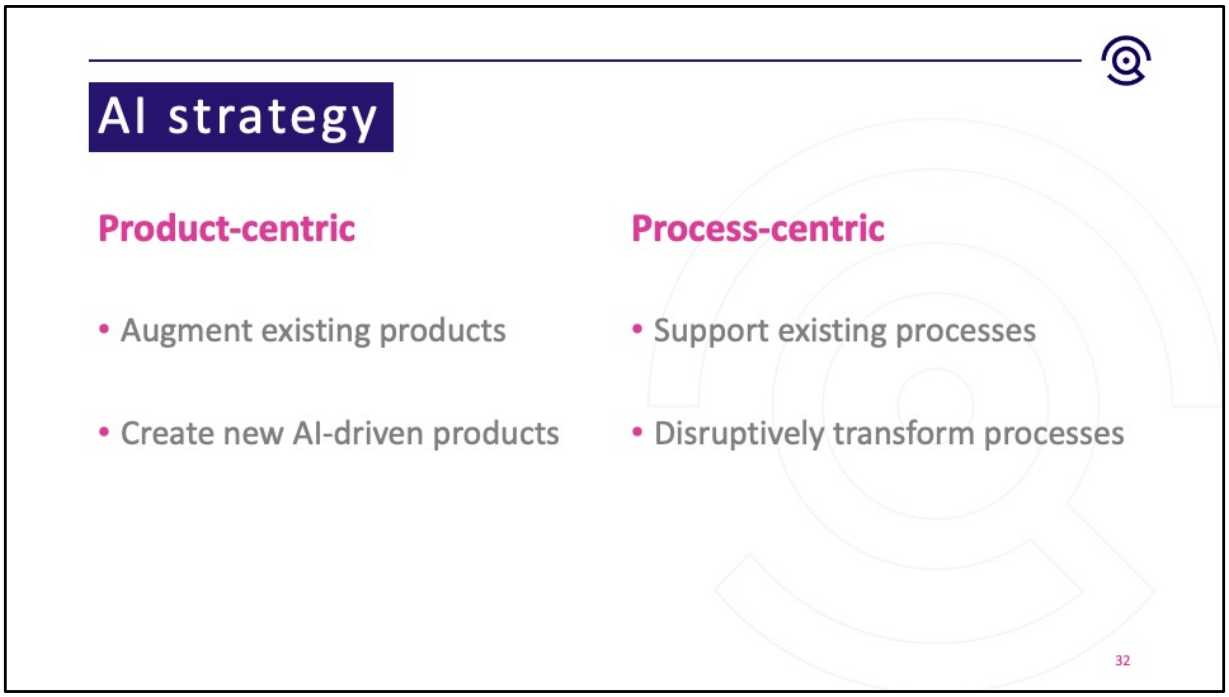

In recent years, AI researchers have adopted the practice of publishing paper pre-prints (before peer-review) on arXiv, an online repository for research articles. The graph on the left shows an increase from 2010 to 2019 in the number of AI papers on arXiv by each paper's primary subcategory. Over this period of 10 years, the total number of AI papers on arXiv increased over twenty-fold. The graph on the right shows the number of times various AI and ML software packages have been starred on GitHub. GitHub is a website where developers upload software code and stars indicate a person has expressed interest in a particular project on GitHub (similar to 'likes' on social media posts). We can observe a steady increase in the cumulative GitHub stars for various open source software solutions, indicating the increasing popularity (and possibly usage) of these frameworks.



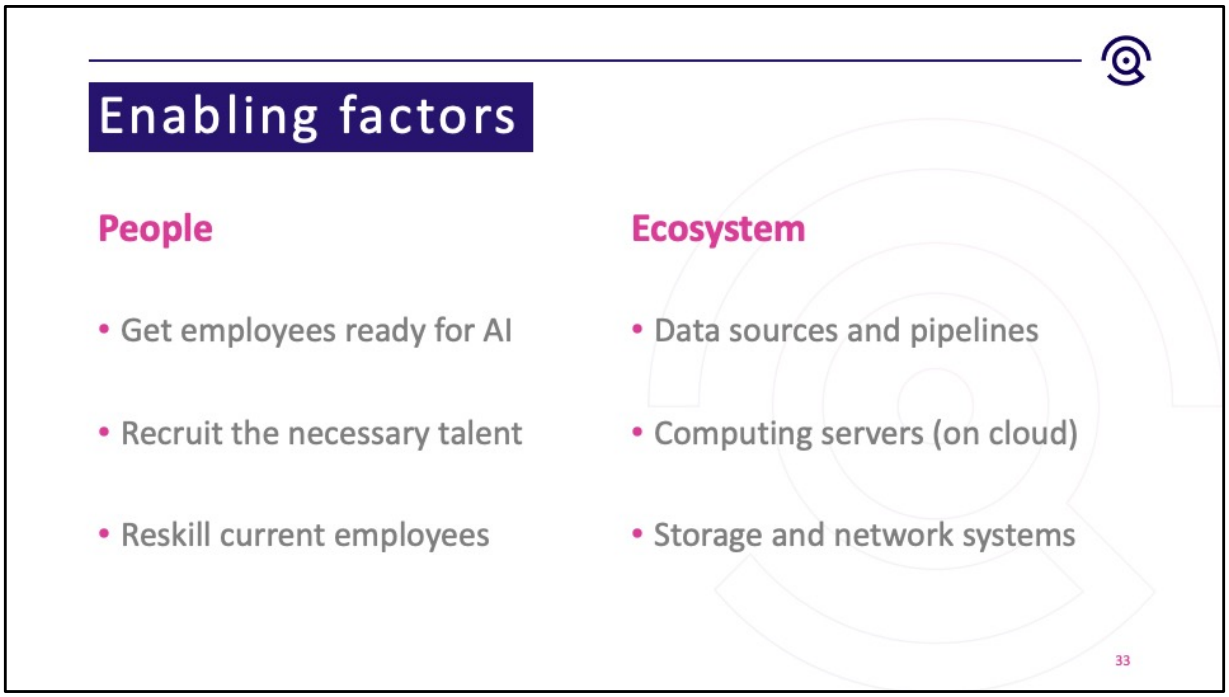

There are a couple of challenges that can limit the adoption of AI in practice. The idea that the "status quo is working fine" can result from a company culture that does not see the need for AI and prefers to continue as-is. This might be true for now, but can leave you empty-handed when competitors do start to innovate (think about Kodak as a classic example of not adapting). Leadership in the company might have an incomplete understanding of what is possible whit AI and it's resulting impact. This course has the goal to bring basic AI knowledge to managers such that this does not need to be a limitation anymore. The quantity or quality of the data in your organization might not be sufficiently high in order to create business value (garbage in, garbage out). In that case it is very important to improve the data governance processes and already try to start a pilot project with the little amount of quality data that you have. Your organization might lack the necessary skills and talent to develop AI solutions, which makes it imperative to attract this talent as soon as possible. AI is not always trusted with some issues regarding ethics, privacy (GDPR) and cyber-security. It is of course vital to trust any solution you develop and we will discuss this in more detail later.




Now that we have seen some drivers and challenges for AI adoption, ask yourself the question “Am I ready for AI?”. It is very important to have your AI strategy and vision aligned with your business goals. Don’t just do AI for AI’s sake but really focus on business problems that you want to solve and think how AI can help you with this. In order to be successful, it is important to have good data governance and make intelligent use of your data to make valuable business decisions. Prepare your people and the company culture for an AI transformation, this might be harder than first expected so do not underestimate this aspect. Set up an AI ecosystem with data sources, proper information flow and a solid technical infrastructure. The combination of a good strategy, ecosystem, talent pool and proper data governance allows you to be ready to harvest value from AI solutions.

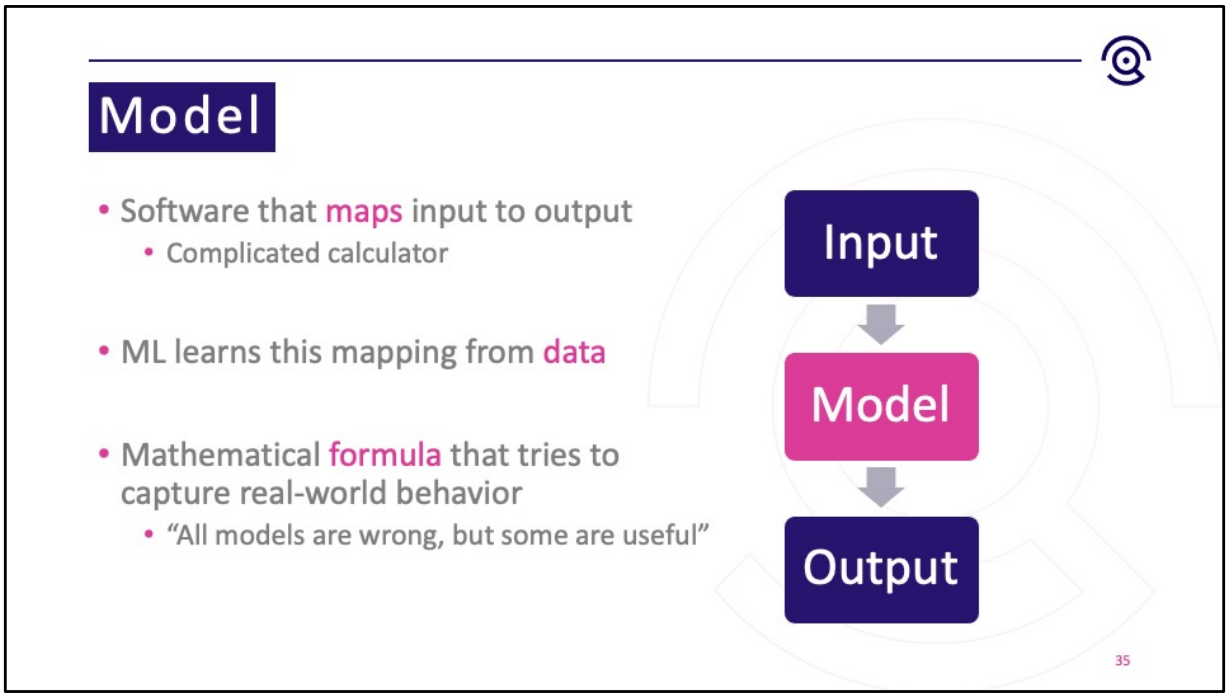



Your AI strategy can be either product-centric, process-centric or a combination. A product-centric strategy puts focus on augmenting existing products or creating new AI-driven products. A process-centric strategy puts focus on supporting existing processes or disruptively transforming processes.

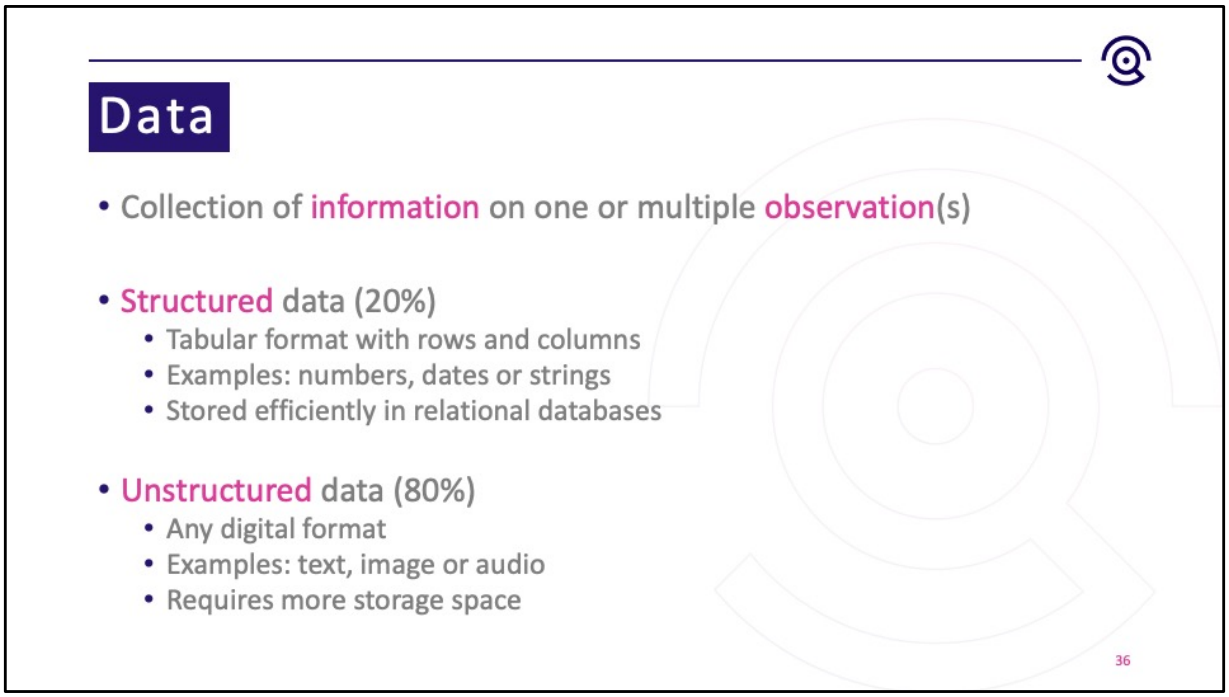



Both people and an ecosystem are important enabling factors for a success story. It is important to get your employees ready for AI, recruit the necessary talent and reskill/upskill current employees. A proper AI ecosystem should consist of different components which work together in a seamless way: data sources/pipelines, computing servers (possibly on the cloud) and storage and network systems. We will dig deeper on both people and ecosystems in later modules of this course, but this already provides a first introduction.

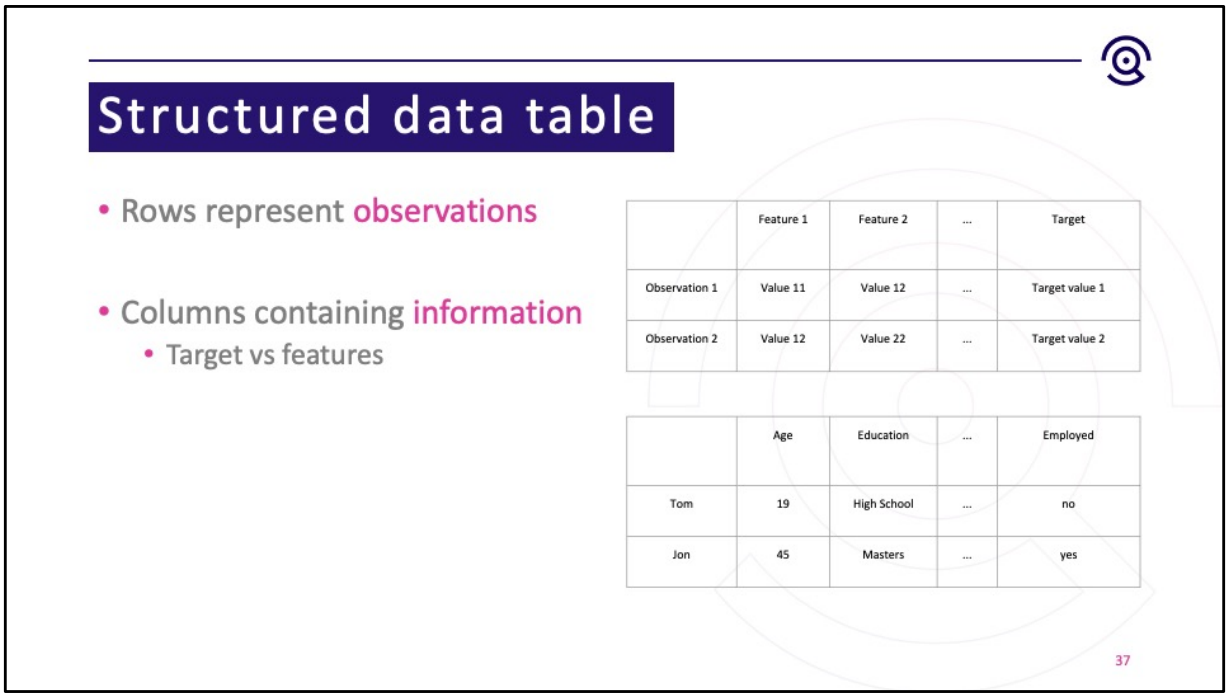

Let’s check out an overview of basic AI concepts to clarify some of the technical terminology often used.



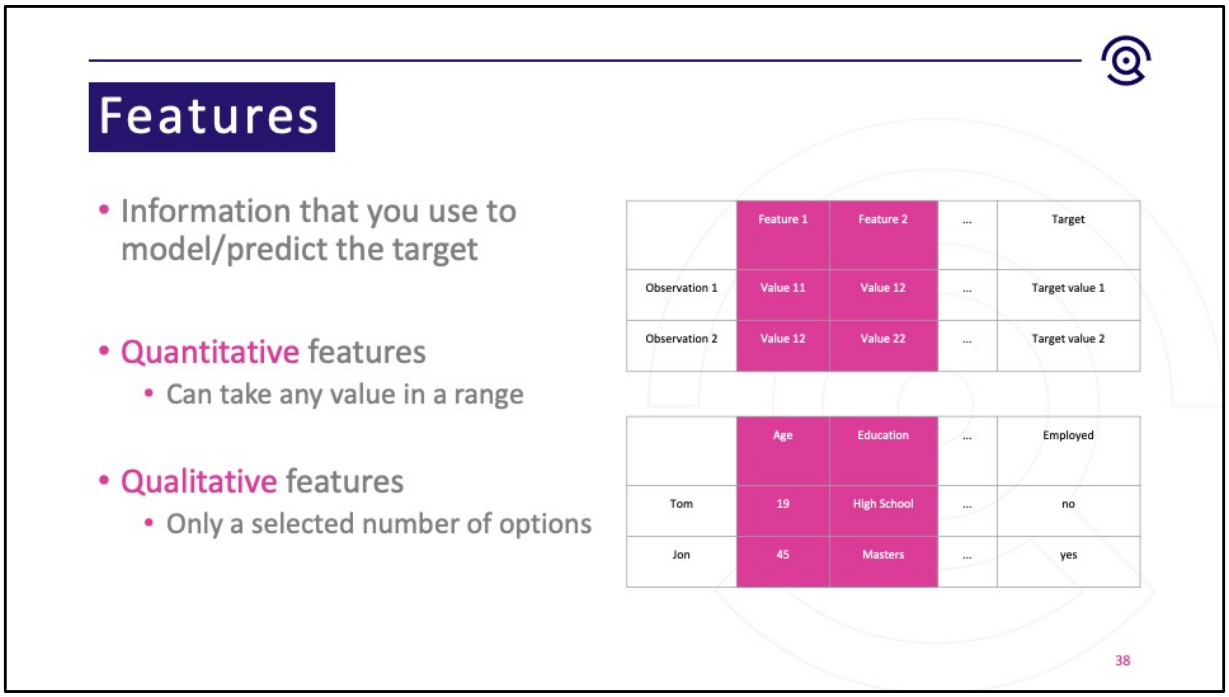

A model is a piece of software that maps input to output. This acts like a complicated calculator, taking input values and producing an output value. Machine learning "learns" this mapping from the supplied data without the need to explicitly define the rules/logic. Models can be seen as mathematical approximations of the real-world that try to capture the underlying behavior. A very famous quote is the following: "All models are wrong, but some are useful". This shows that the real world is too complex to capture in a model, but a good approximation can still be very valuable.



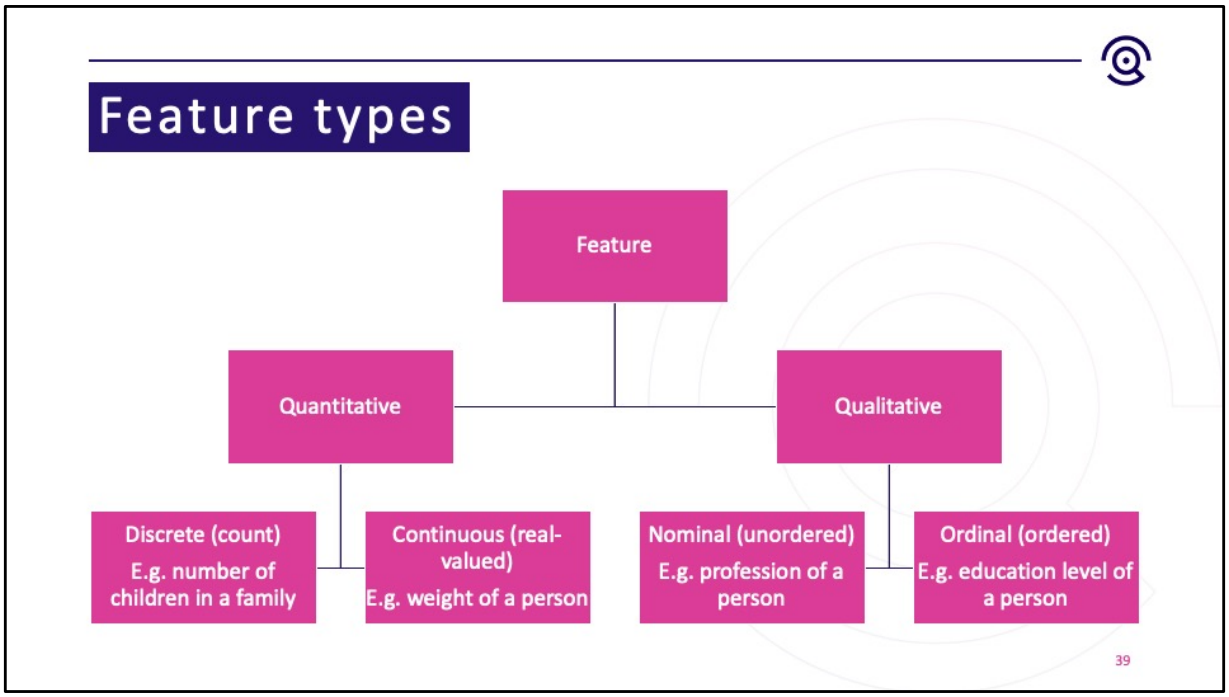

Data is a collection of information on one or multiple observation(s). This information can take several formats, but in general we distinguish between structured and unstructured data. Structured data has a tabular format with rows and columns. Columns contain the different pieces of information, for example numbers, dates or strings, and rows represent individual observations. Unstructured data comes in any digital format, for example as text, images and audio fragments. It is believed that about 20% of enterprise data is structured while 80% of the available data is unstructured. Structured data can be stored efficiently in relational databases while unstructured data requires much more storage space.



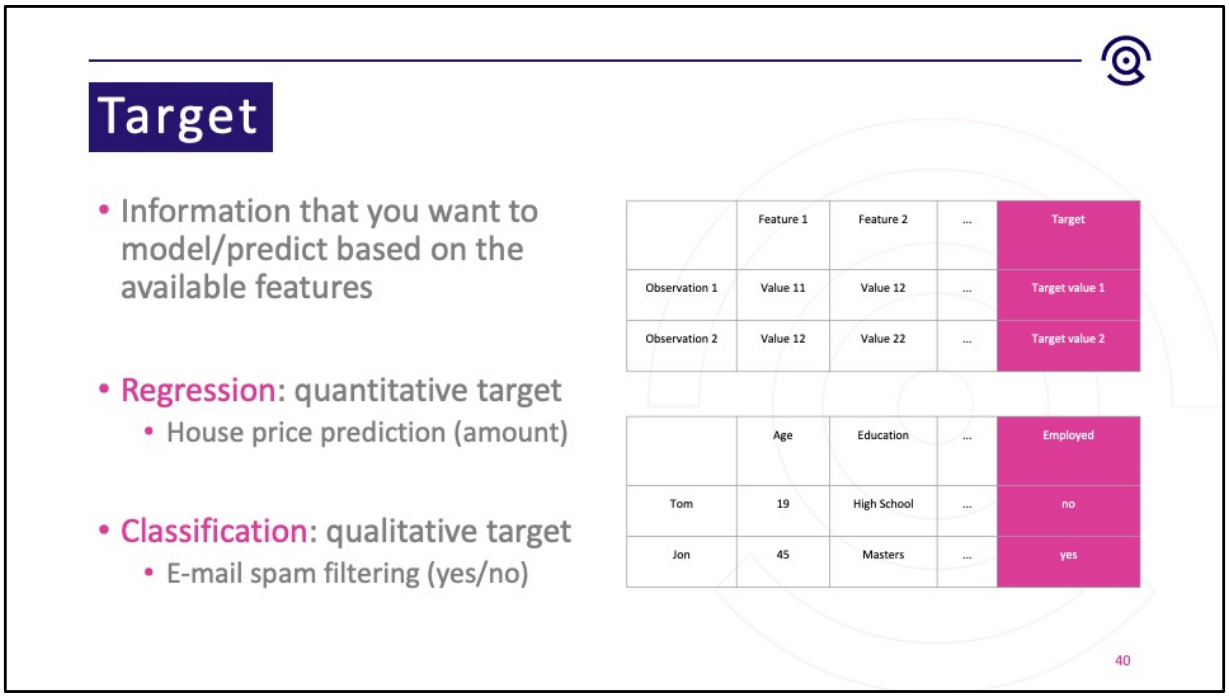

On the right we show an example of a structured data table. The rows represent individual observations, for example Tom and Jon. The columns contain information on the observations, for example the age, education level and an employment indicator. Tom is 19 years old with a high school education and is not employed, while Jon is 45 years old with a Masters education and is employed. Regarding the information in the columns, we make a distinction between the target (employed) and several features (age and education).



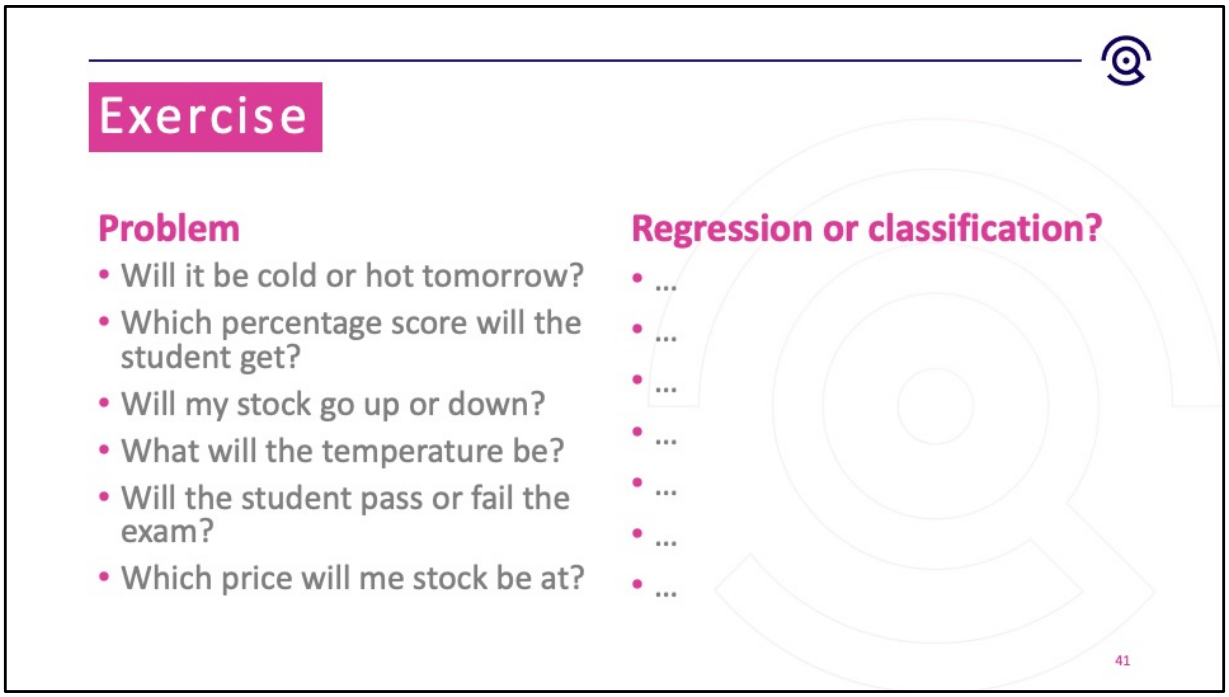

Features are the pieces of information that you use to model/predict the target of interest. Quantitative features can take any value in a certain range, for example the age of a person. Qualitative features can only take a selected number of predefined options as value, for example the education level .



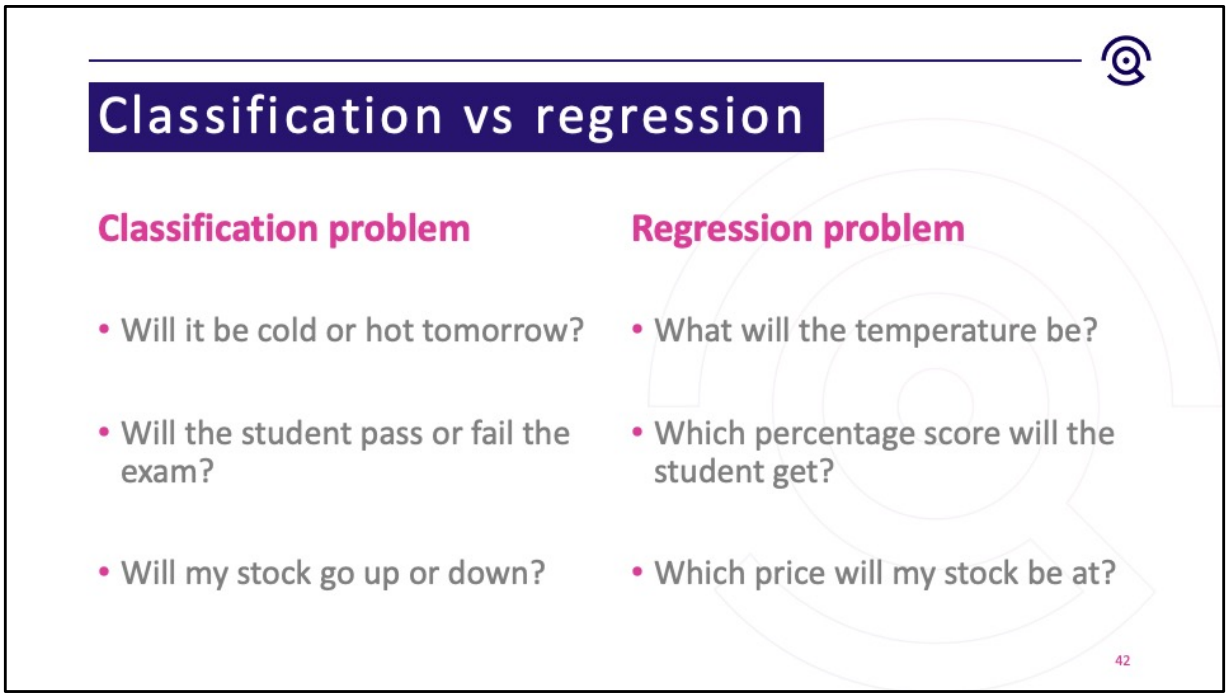

Quantitative and qualitative features can be further divided in sub-categories. A discrete quantitative feature represents a count value which can not take decimals, such as the number of children in a family. A continuous quantitative feature represents a real value which can contain decimals, such as the weight of a person. A nominal qualitative feature represents categories without any inherent order among them, such as the profession of a person. An ordinal qualitative feature represents categories which do contain an inherent order among them, such as the education level of a person.



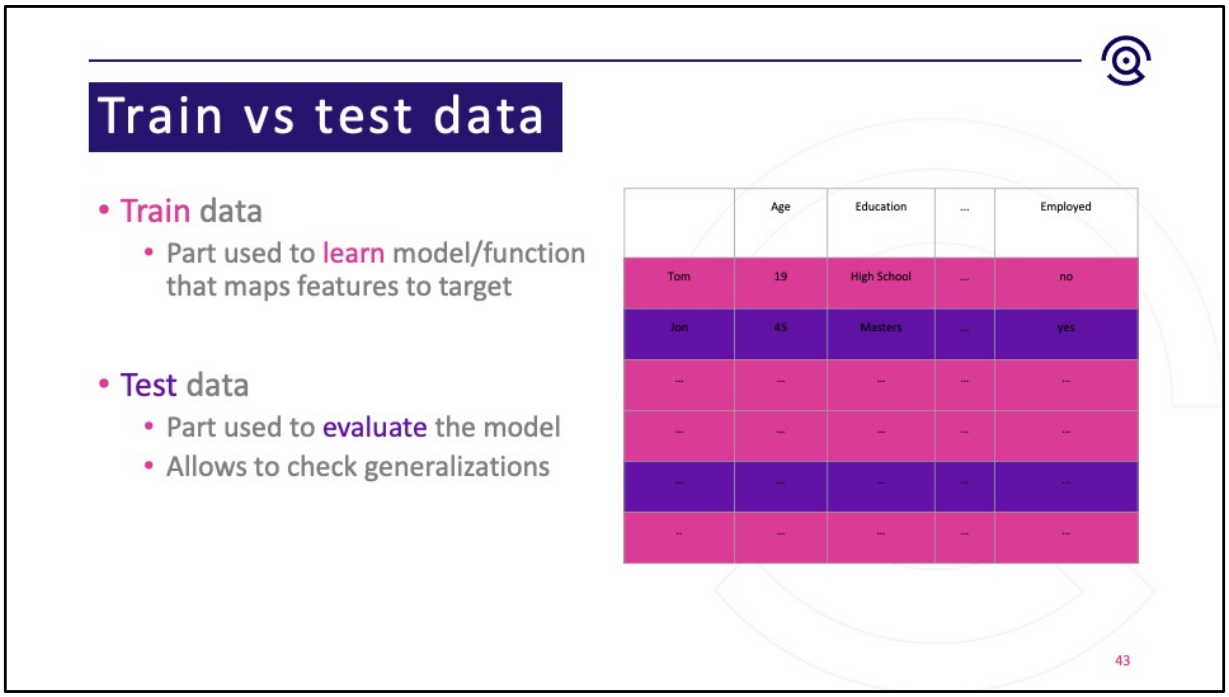

The target is the piece of information that you want to model/predict based on the available features. We are dealing with a regression problem when the target takes a quantitative format, for example when trying to predict housing prices as amounts can take any value in a certain range. We are dealing with a classification problem when the target takes a qualitative format, for example when trying to predict whether an e-mail is spam or not as the only options are yes/no. The prediction of the employment status is also a classification problem, as someone is employed or not (those are the only options). Predicting how long someone is unemployed can be seen as a regression problem as this target can take any value in a certain range.



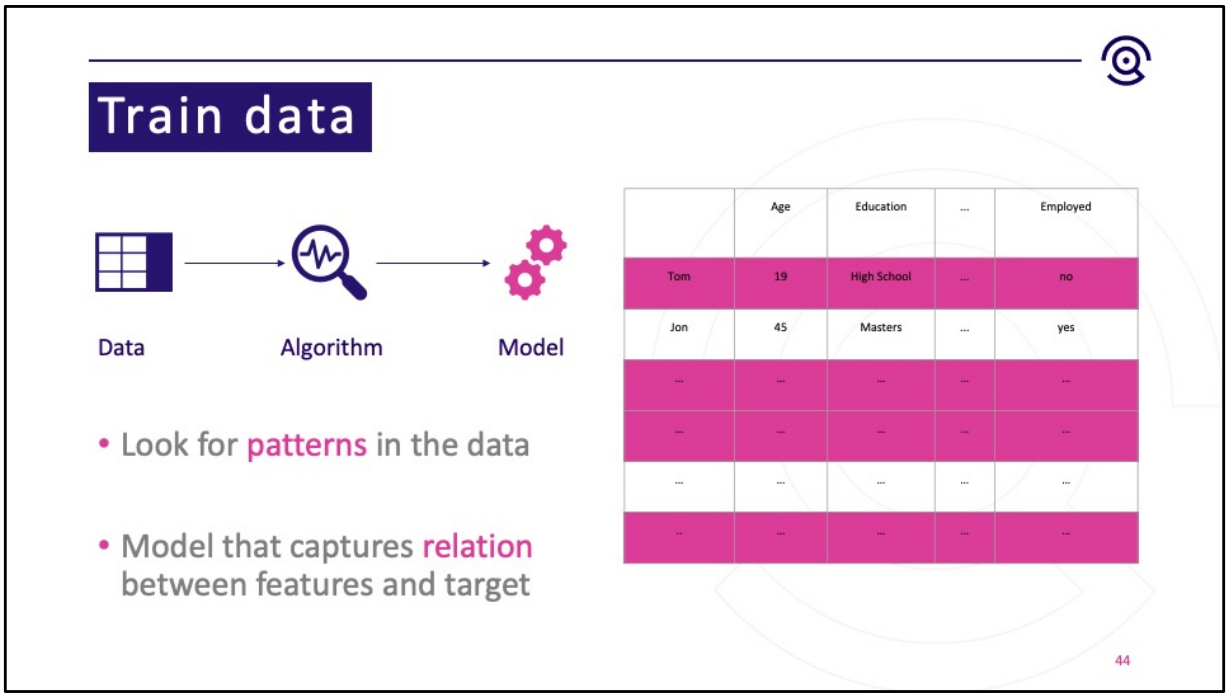

Time for a short exercise. There are six problem definitions, which one is a regression task and which one is a classification problem? Feel free to pause the video for a bit and think about this.



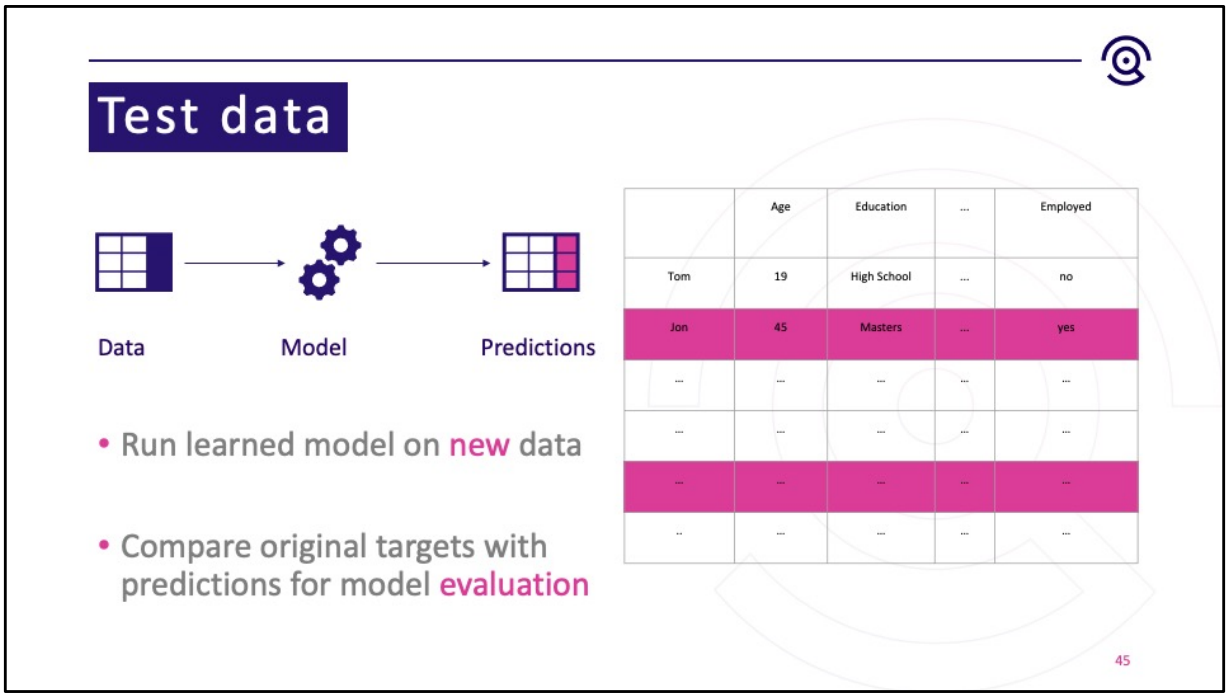

You might have noticed that there were three couples of problems with the same goal, but one was formulated as a regression task and the other as a classification task. The following are classification problems: cold or hot temperature, pass or fail exam, stock up or down. Each time there are only two options for the target. The following are regression problems: the temperature, percentage score and stock price. In these tasks the target can take any value within a certain range.



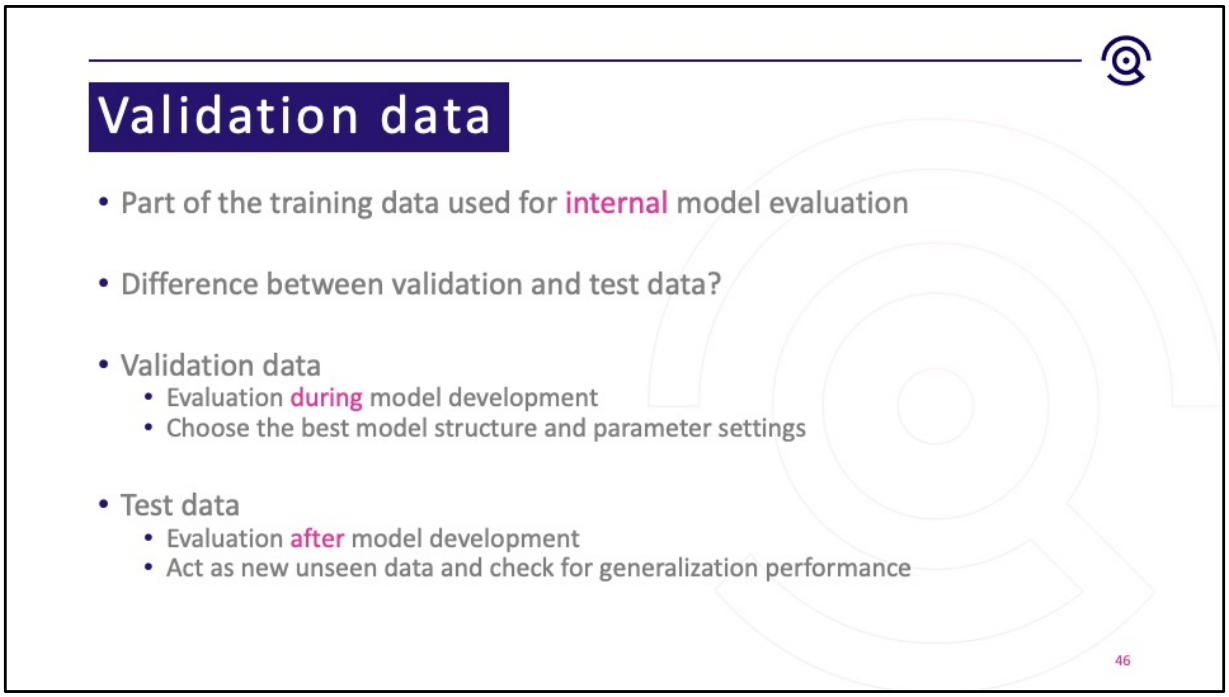

When building AI models, we typically split our full dataset in two separate parts: train and test data. Train data is the part of the data used to learn the model/function that maps features to target. Test data is kept under lock during model development and should never be used during model training. Test data is used to evaluate the model after training is completed. Since this part of the data is never seen by the model during development, this allows to check how your model generalizes towards new data. On the right we show a four/two split in train/test data for our table of six rows. Notice that the train/test part are completely separate and fully disjoint sets.



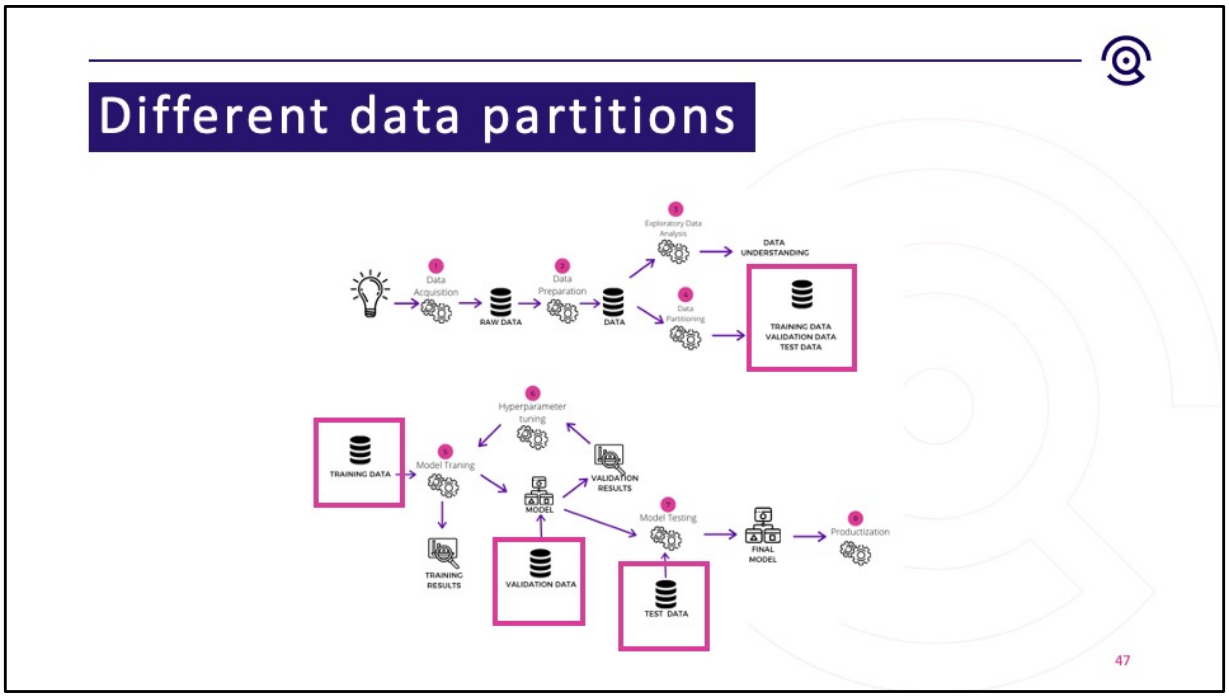

Train data is the part that is fed to the AI algorithm in order to train a model. The AI algorithm will look for patterns in the train data and the trained model captures the relation between the features and target. The model model can only find patterns/relations which are present in the train data and it is therefore important that this part represents a sufficient diverse collection of the data. On the right, we highlight four out of six rows in the table which are chosen to be part of the train data.



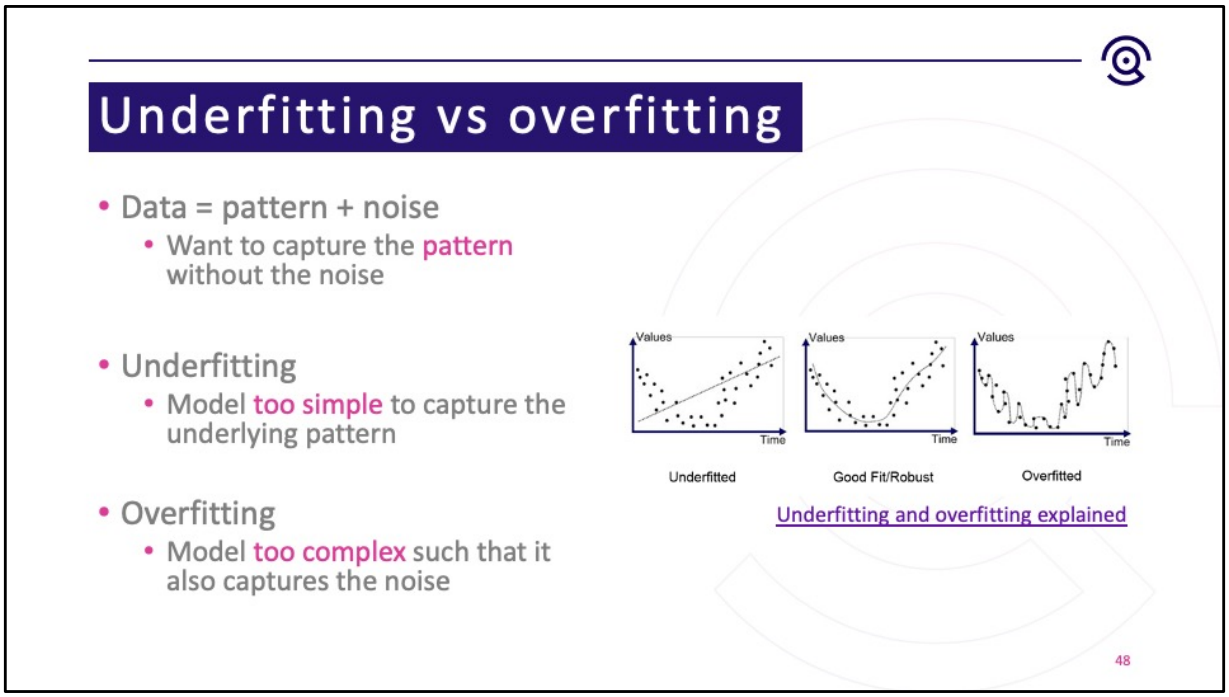

The test data is never used during model training, this is extremely important. After the model is trained on the train data, the model can be used to make predictions on the unseen test data. We can evaluate how well the model generalizes to new observations by evaluating the original targets of the test data with the model's predictions. Assume that you have trained 5 different models, evaluating them all on the same test set allows you to decide which model is performing the best. It is again important that this part represents a sufficient diverse collection of the data to make sure that the comparison is fair and reliable. On the right, we highlight two out of six rows in the table which are chosen to be part of the test data.



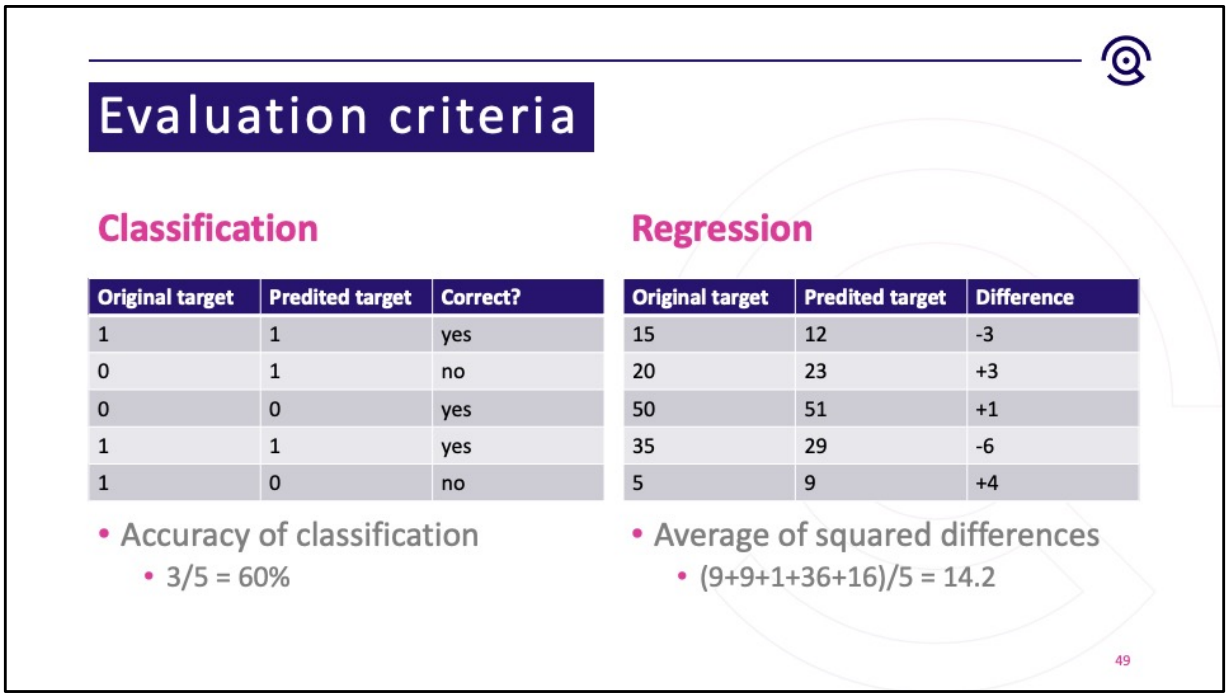

Next to train and test data there is typically also a validation dataset. Validation data can be seen as part of the training data that is used for internal model performance evaluation during the training process. Both validation and test data are used for evaluation purposes, so what is the difference? Validation data is used for evaluation during model development to choose the best model structure and parameter settings. Test data is used for evaluation after model development to act as new unseen data and check for generalization performance. The big difference is that validation data is used during training to iterate and find the best model, while test data is really kept under lock and only used after the model is fully trained.



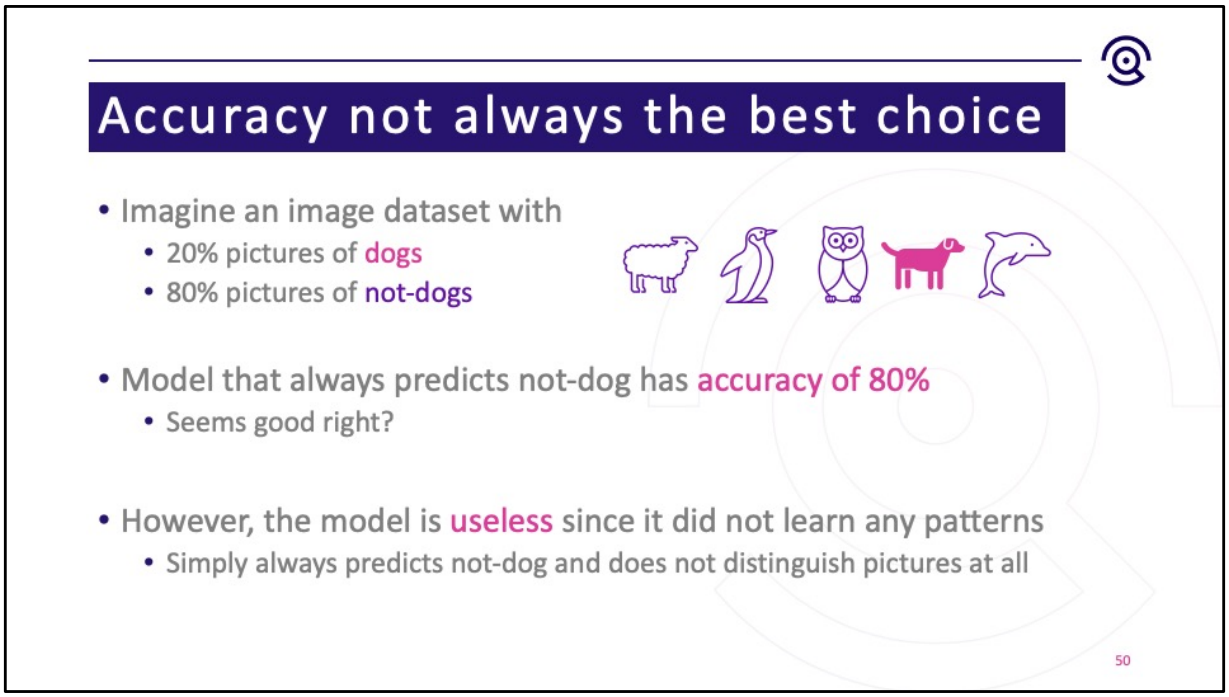

The following graphic shows the model development process and the relation between the different data parts. The first three steps are to acquire, prepare and understand your data. We will go into more detail on this whole process in the second module of this course. Step four shows the data partitioning in training, validation and test data. The training data is used in step five to train the model with certain parameter settings. This results in a trained model and in step six this model is evaluated based on the validation data. These evaluation results allow to improve the model, change some parameter settings and train it again on the training data. This is an iterative process which is continued until the validation results are satisfactory. Once this is the case, the model is evaluated on the test data in step seven to assess how well the model is able to generalize towards new data. This is important because the final model is then put in production in step 8, where it constantly must deal with new observations to predict. Deployment and monitoring of models in production are the topics of modules four and five of this course.



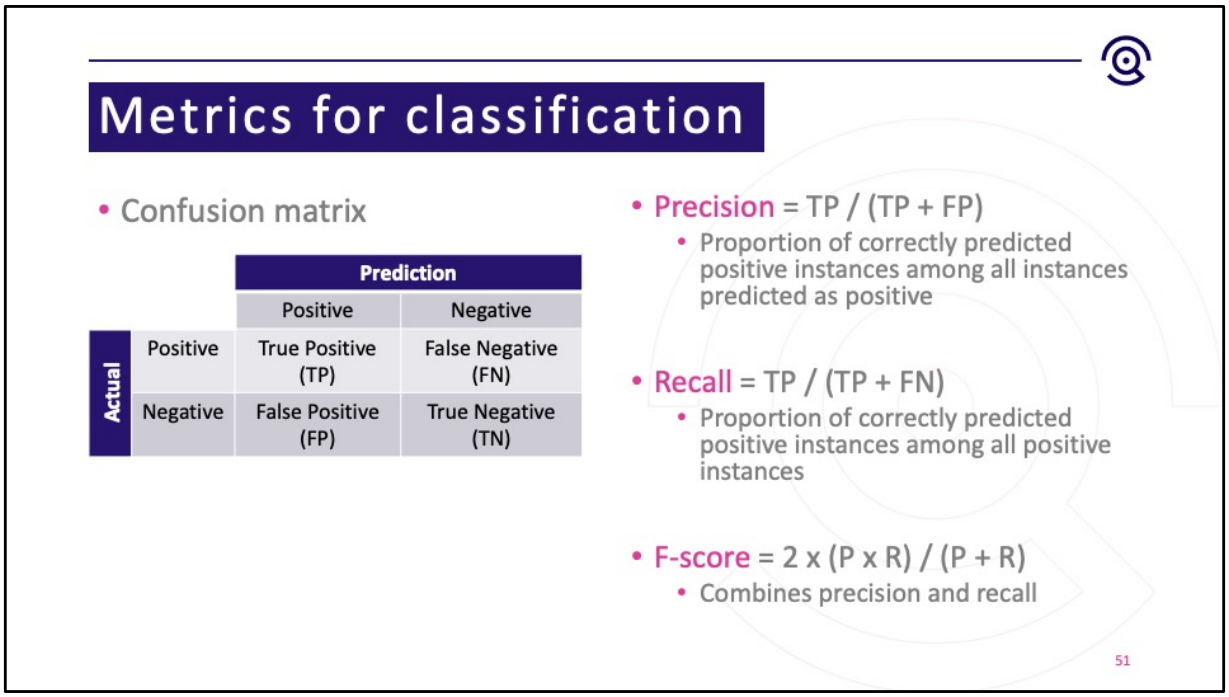

Data typically consists out of a certain pattern with noise on top of it. The goal of any model is to capture the pattern and filter out the noise. Underfitting means that a model is too simple to be able to capture the underlying pattern. Overfitting means that the model is too complex and starts capturing the noise as well. The graphs on the right explain this a bit more. The black dots represent measurement values over time and our goal is to find a trend in these observations. The measurements first go down over time and start increasing again after a while. In the middle graph we observe that the model, as indicated by the dotted line, follows this general trend very well. This represents a good fit where the pattern is captured without following the noise. The left graph shows an underfitted model, as the straight line is too simple to capture the down/up patterns in the data. The right graph shows an overfitted model as the dotted line now starts to pick up on the noise and results in a very wiggly structure. Both an underfitted and overfitted model will perform badly on new observations and it is therefore important to obtain a good model fit.




The comparison of models calls for some evaluation criteria that express a model's performance in a certain number. This number can then be compared over different models and allows to choose the best one. Let's start with a very basic example for both classification and regression problems. On the left we have a classification task with the original and predicted targets. Classifying a 1 as a 1 is correct but classifying a 1 as a 0 is incorrect. A very simple and intuitive way to compare a model's classification performance is to calculate the accuracy as the number of correct classifications with respect to the total number of classifications. Three predictions are correct out of a total of five, resulting in a classification accuracy of 3/5 or 60%. In a regression problem, both the original and predicted target represent numerical values that can take any value in a certain range. An intuitive way to calculate the performance for each individual observation is to calculate the difference between the original and predicted value. To summarize the performance over all observations in one number, one can then take the squared version of these differences (in order to punish over/underestimations equally) and then average these values over all the observations. This number is called the mean squared error (MSE) and is a very popular metric in regression problems.




Accuracy is very intuitive and easy to calculate, but it is not always the best measure to use for classification problems. This is especially the case when a dataset is unbalanced, i.e., when the proportion of zeros and ones in the target is not approximately 50/50. Imagine an image dataset with 20% pictures of dogs and 80% pictures of not-dogs. A model that always predicts not-dog has accuracy of 80%, which seems pretty good right? However, the model is useless in practice since it did not learn any patterns from the data. It simply always predicts not-dog and does not distinguish pictures at all.



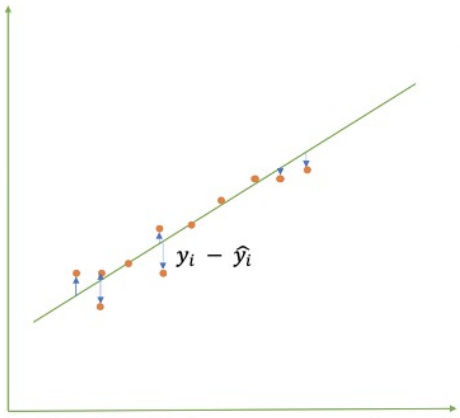

Many other metrics exists for classification problems to overcome the issues of accuracy. A popular approach to summarize classification results is the confusion matrix. This is a two-dimensional table with four numbers. True positive (TP) counts the number of positive observations that are correctly classified as positive. True negative (TN) counts the number of negative observations that are correctly classified as negative. False positive (FP) counts the number of negative observations that are wrongfully classified as positive. False negative (FN) counts the number of positive observations that are wrongfully classified as negative. It is very interesting to look at such a confusion matrix because it shows it detail how different cases are correctly/wrongfully classified, but it is still not one number that can be compared over different models.

Many metrics exist to summarize the confusion matrix in one number, all with a different focus. Precision is defined as TP / (TP + FP), namely the proportion of correctly predicted positive instances among all instances predicted as positive. Recall is defined as TP / (TP + FN), namely the proportion of correctly predicted positive instances among all positive instances. The F-score combines both precision and recall in one number. Many other metrics exist which allows to use a proper metric for the problem at hand. Imagine that you want to predict whether someone has a terminal disease, in that case it is very important to minimize the amount of False

Negatives because then a sick person will not receive treatments.

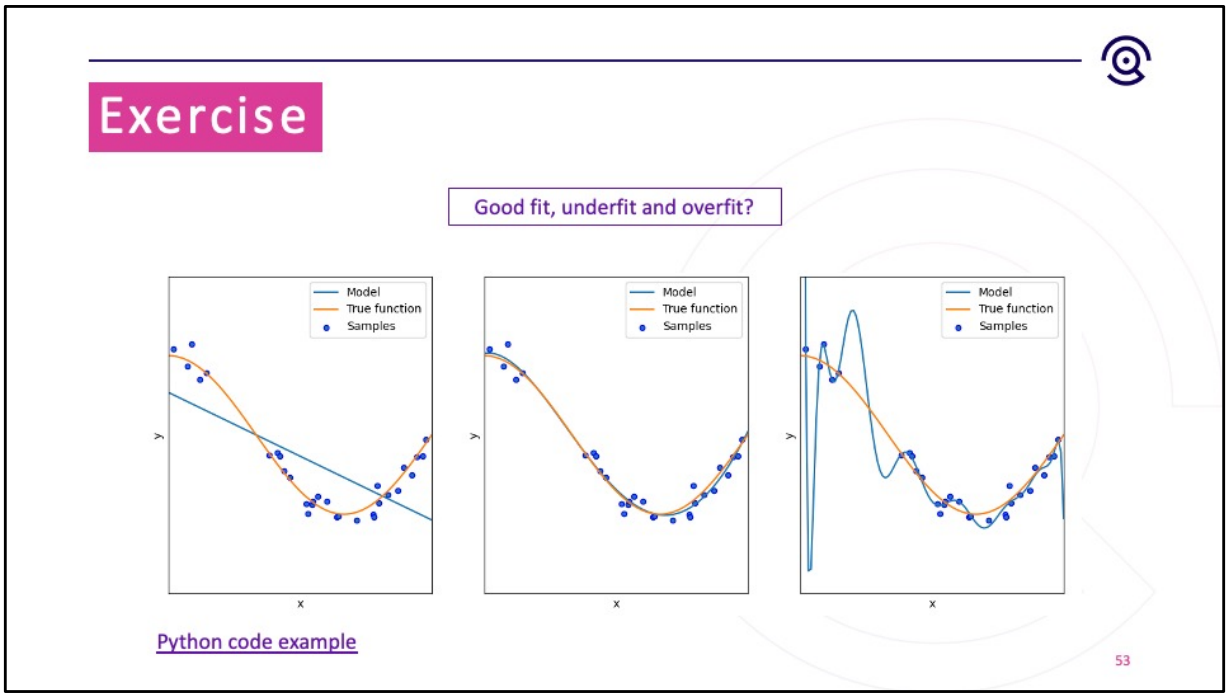
## Metrics for regression

- Mean squared error
  - $MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$
- Mean absolute error
  - $MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$
- Mean absolute percentage error
  - $MAPE = \frac{100}{N}\sum_{i=1}^{N}|\frac{y_i - \hat{y}_i}{y_i}|$




52

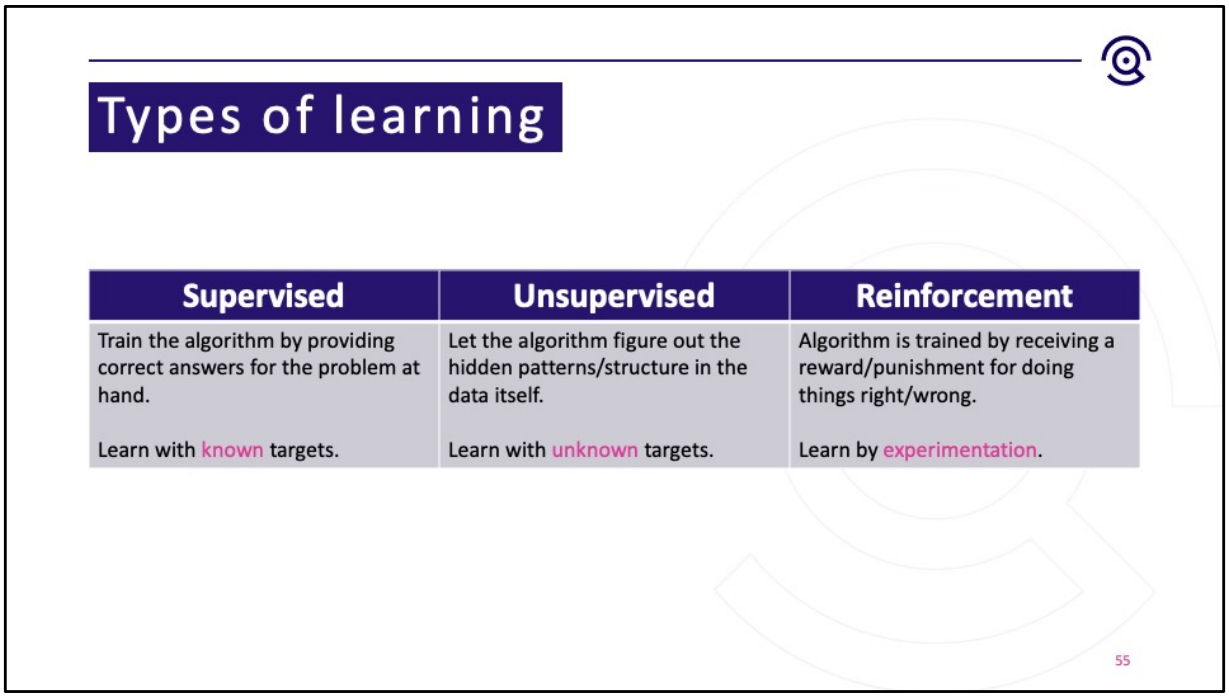
Also for regression many metrics exist to calculate the performance of a model. The basis for all these metrics is usually the difference between the observed value (y) and the predicted value (y_hat). The difference (blue) between the orange dot (y) and green line (y_hat) should be as small as possible preferably. These differences are aggregated over all the observations to result in a single number. The way of aggregating leads to different metrics. The MSE averages the squared differences, the MAE averages the absolute differences and the MAPE averages the relative absolute differences. The choice of metric often depends on the problem at hand where for example the MSE punishes large errors more heavily than the MAE.



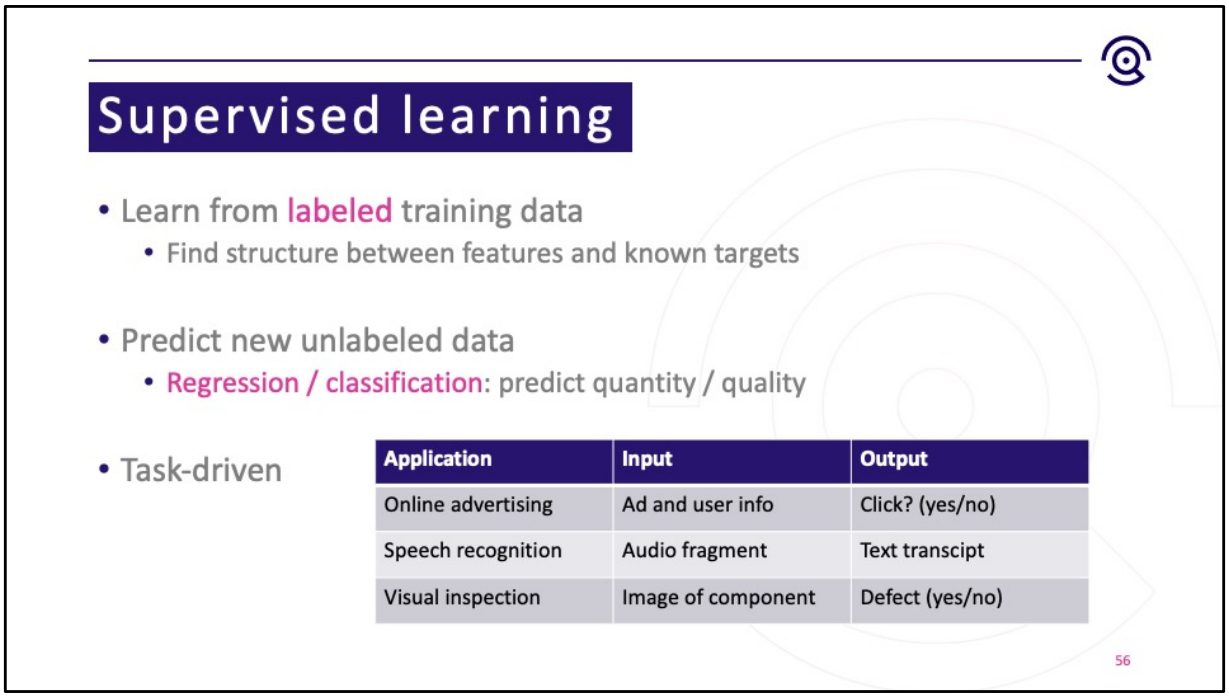

Let’s end this part with a short exercise. In the three graphs we have data samples (blue dots), the true underlying function (orange line) and then a model fit (blue line). Which of these three represents a good fit, underfit and overfit? You can pause the video to think about this and we will discuss after you resume.

On the left we have an underfit, as the blue line can not capture the trend in the orange line. On the right we have an overfit as the blue line is much more complex compared to the orange line and is following the noise in the blue dots too much. In the middle we have an almost perfect fit so this would be the preferred model.

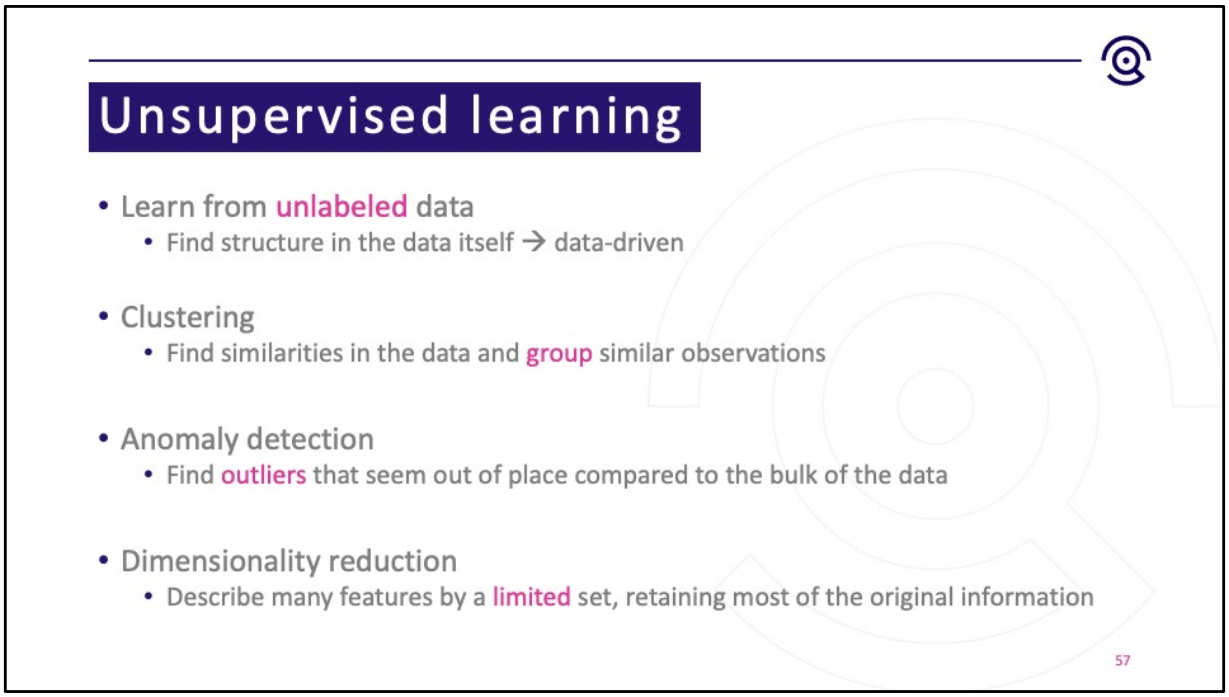

Let’s have a look at the different types of machine learning.



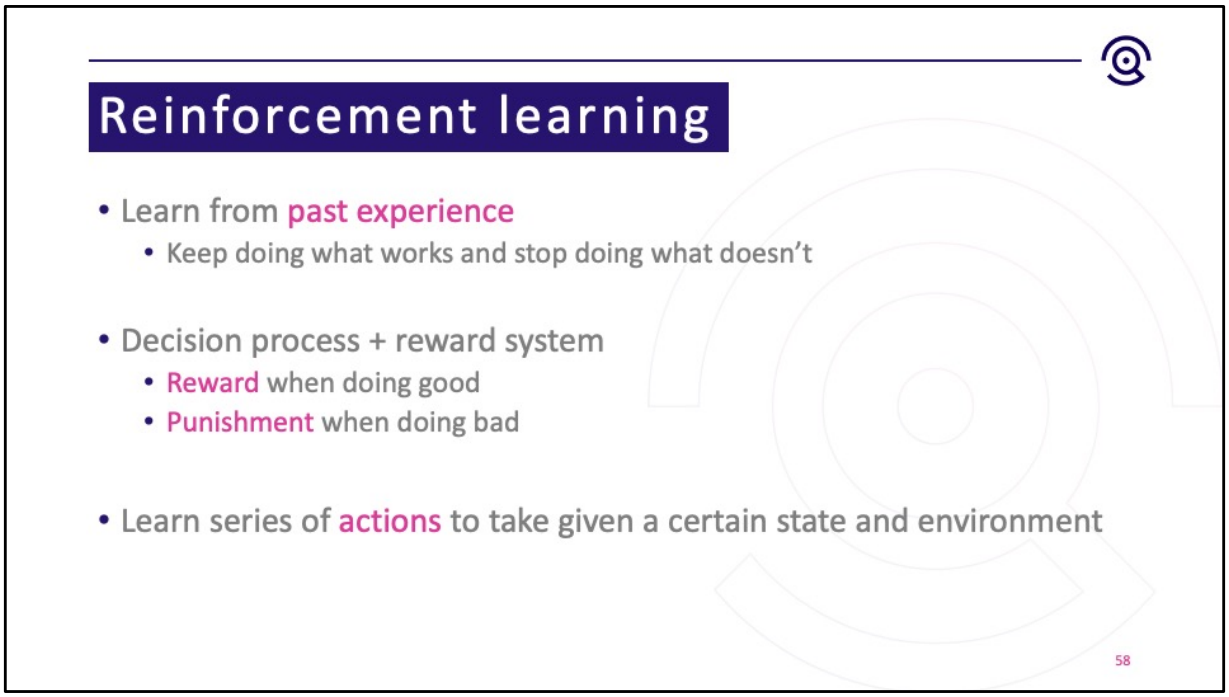

There are three main types of machine learning: supervised, unsupervised and reinforcement learning. In supervised learning the algorithm is trained by providing correct answers for the problem at hand. This involves learning with known targets and allows the model to discover the relation between the features and target. In unsupervised learning the algorithm needs to figure out the hidden patterns/structure in the data itself. This involves learning without targets where the model looks for similarities between the features. In reinforcement learning the algorithm is trained by receiving a reward/punishment for doing things right/wrong. This involves learning by experimentation with incentives to keep doing things that work and stop doing things that don't work.



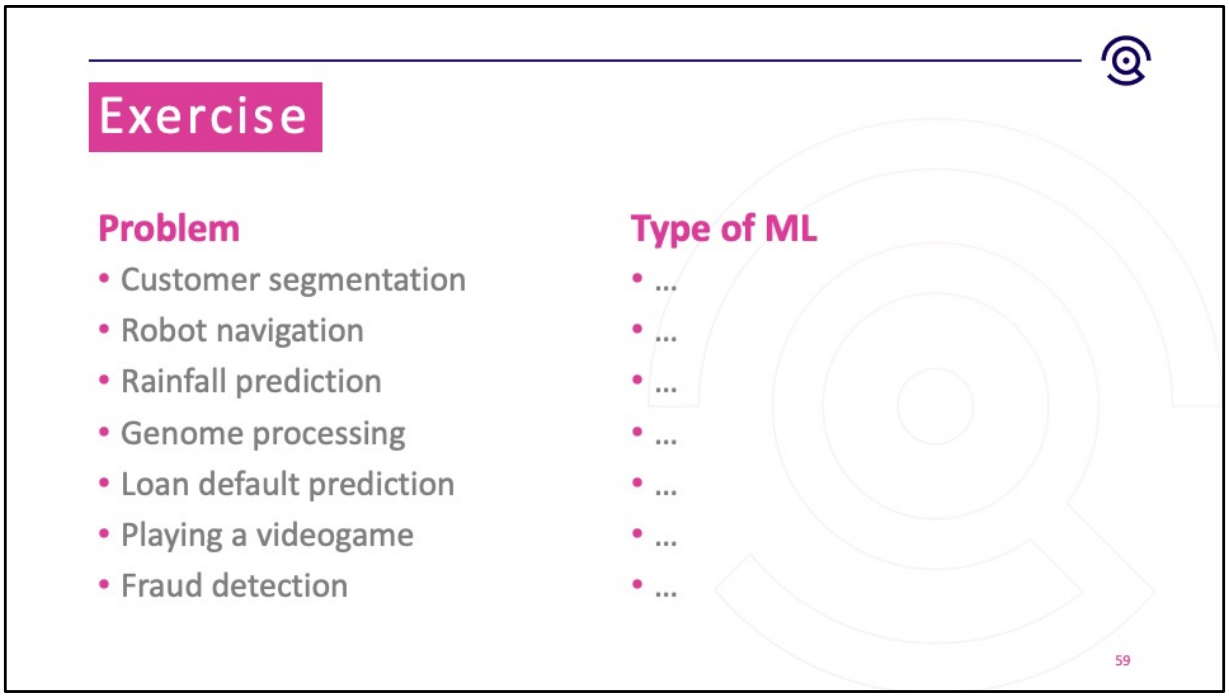

Supervised learning makes use of labeled training data, i.e., with known targets. The goals is to discover the relation between features and targets. This relationship can then be used to predict new unlabeled data. Regression and classification (as we have seen until now) are typical examples of supervised prediction problems. One tries to predict either a quantity (regression) or quality (classification) after observing some example train data. Supervised learning is typically task-driven, meaning that you have specific examples of a certain task and try to find a model that represents this task accurately. The table list some example application areas. Online advertising uses ad and user information to decide whether a person will click on that ad or not. Speech recognition takes an audio fragment and translates this into a text transcript. Visual inspection uses images to decide whether a component is defect or not. All these tasks need some example training data, such as component images with a corresponding defect indicator. The AI algorithm will then produce a model that can be used to predict a defect in new component images.



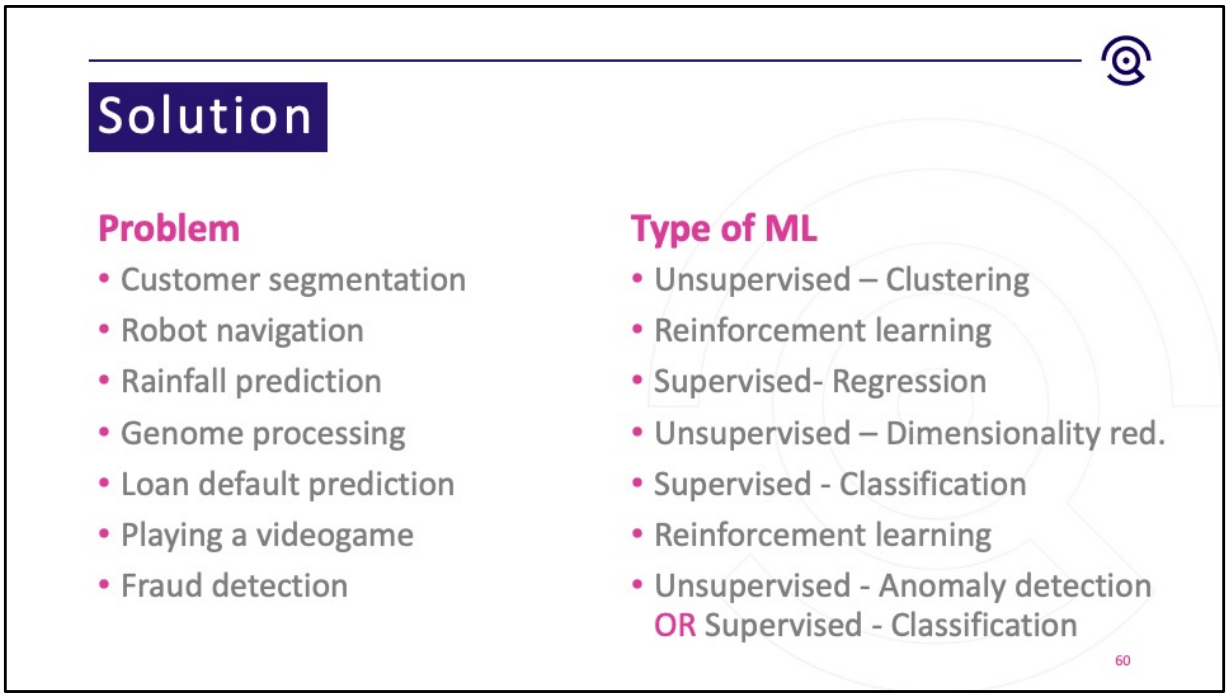

Unsupervised learning makes use of unlabeled training data, i.e., with unknown targets. The goals is to find the hidden structure in the data features, making unsupervised a data-driven problem without specifying example solutions. Unsupervised learning comprises three main classes of problems. Clustering deals with finding similarities in the data in order to group similar observations in clusters. One therefore goes from all the individual observations to a certain number of homogeneous groups. Anomaly detection tries to find outliers that seem out of place compared to the bulk of the data. One therefore goes from all individual observations to two groups: the “normal” cases and the outliers. Dimensionality reduction is used to summarize many features in a limited set, trying to retain most of the original information. One goes from many features per observation (1000’s for example) to a small set (10 for example). A loss of information is inevitable, but the goal is to capture as much information as possible in the small set of new features.



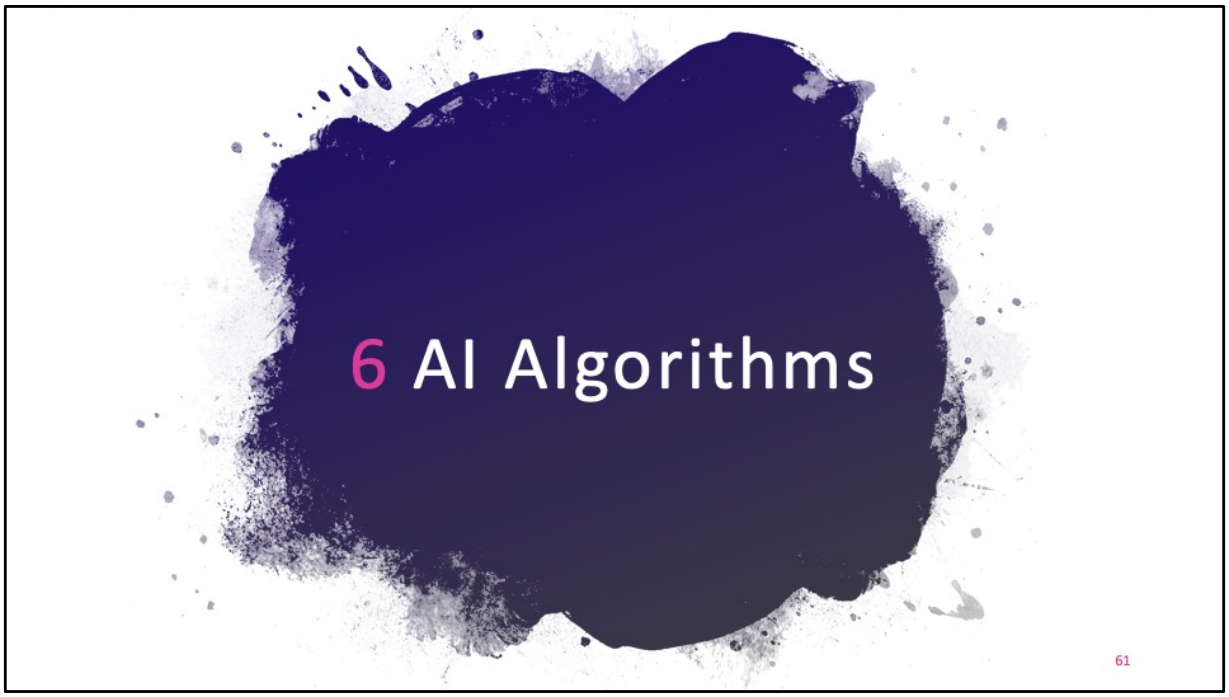

Reinforcement learning uses past experience in order to keep doing what works and stop doing what doesn't. These techniques use a decision process in combination with a reward system. A reward is obtained when a good decision was taken, while a punishment is given when a bad decision was taken. This way the algorithm learns a series of actions to take given a certain internal state and external environment.




Time for a small exercise. On the left we list some problem definitions, up to you to decide which type of ML you would use to solve each problem. Try to be as specific as possible, for example clustering or anomaly detection instead of unsupervised learning. Feel free to pause the video for a bit and afterwards we go over the solutions.




Let's have a look at the solutions. Customer segmentation is typically an unsupervised clustering problem, with the goal of grouping similar customers together based on their personal information or buying behavior. Robot navigation is a reinforcement learning problem where the robot needs to figure out an optimal route from point A to point B by trial and error. Rainfall prediction is a supervised regression problem where historical weather data is used to predict the amount of rain for tomorrow. Genome processing is an unsupervised dimensionality reduction problem where the vast amount of genetic information is summarized in a lower dimension. Loan default prediction is a supervised classification problem to predict whether someone is able to repay a loan based on personal information and previous loan payments. Playing a videogame in reinforcement learning where the optimal strategy is learned along the way of the game. We list one type of ML for each of these problems, but sometimes the same problem can be tackled in multiple ways depending on the available data. Fraud detection without a fraud indicator in the training data can be solved as an unsupervised anomaly detection problem by looking which transactions seem fishy compared to all the other transactions for a certain customer. When a fraud indicator is available in historical train data, then it can also be solved via a supervised classification problem for example. This just to show that it is important to choose an appropriate strategy based on your goal and the available data.

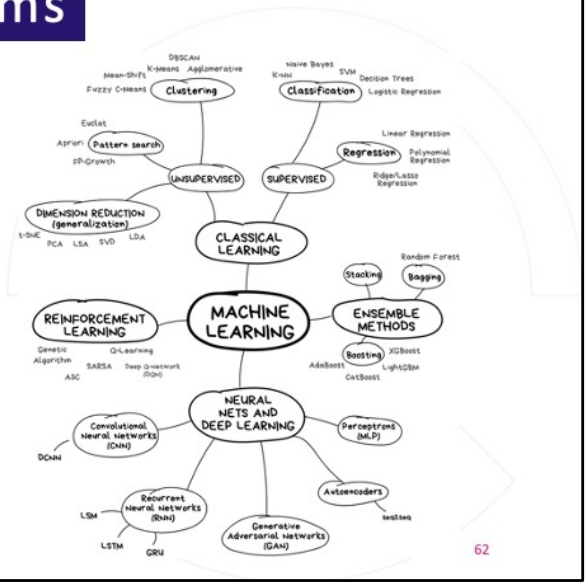



Let’s have a look at some of the popular AI algorithms to develop models.

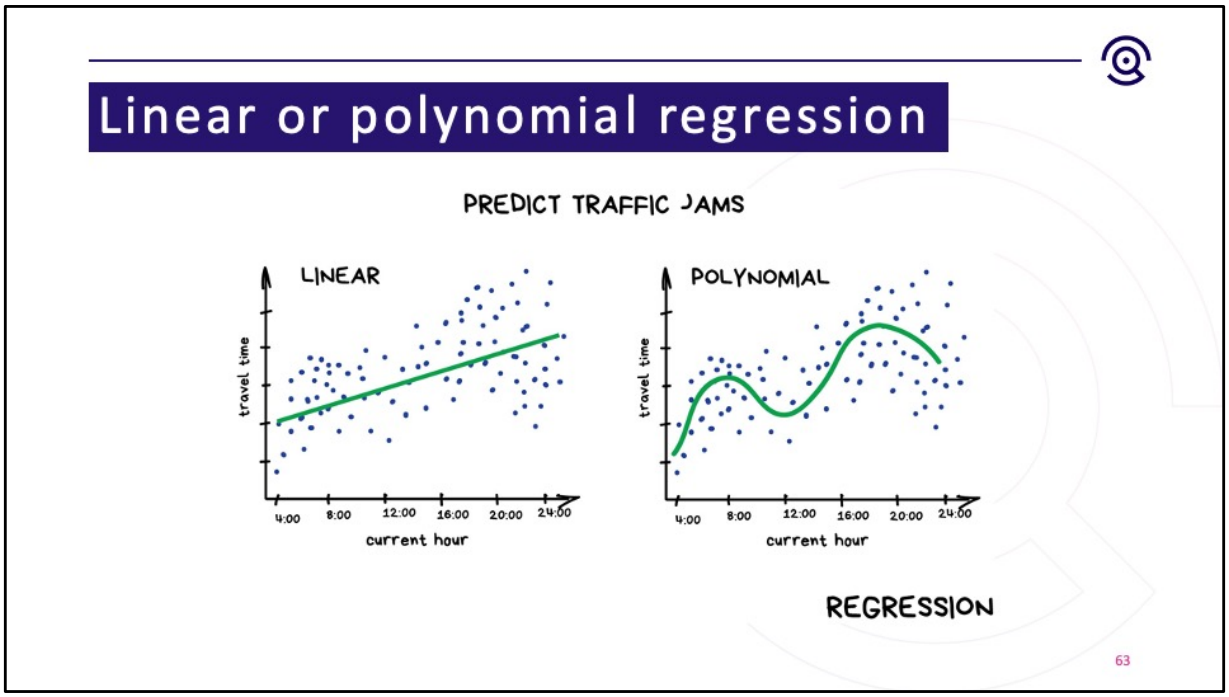
# Overview of algorithms

vas3k – Machine Learning

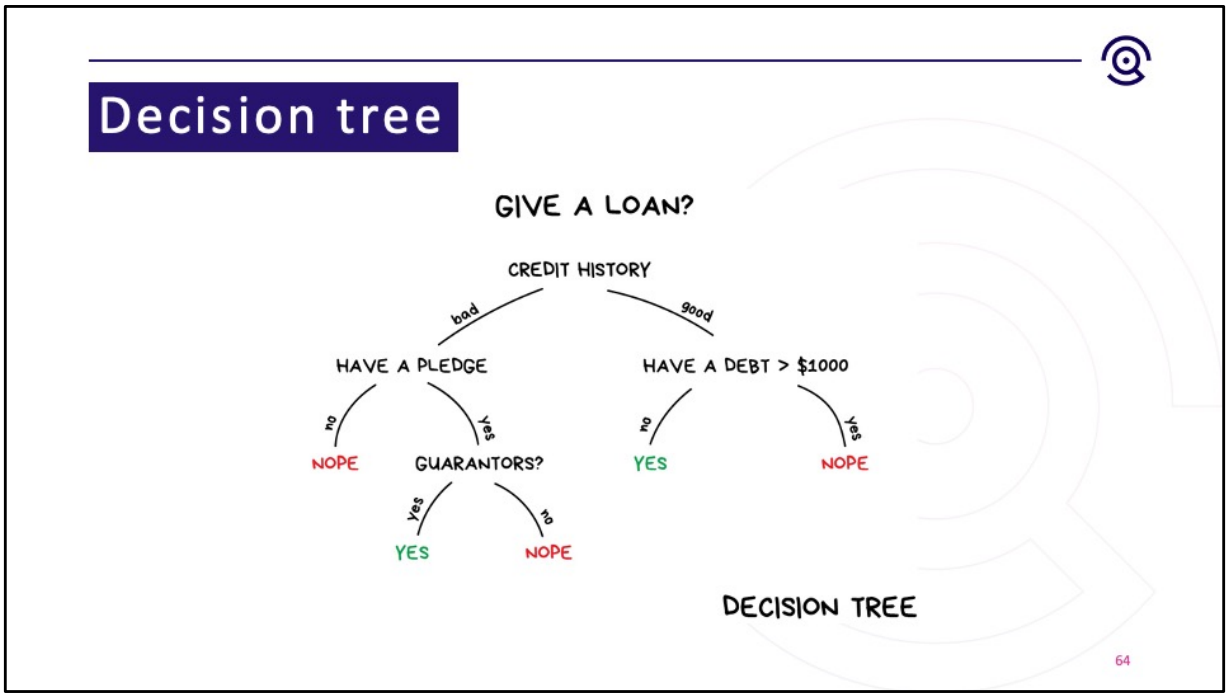
- Very interesting introductory blog on AI algorithms
- Pictures in this section taken from this blog (credits to vas3k)



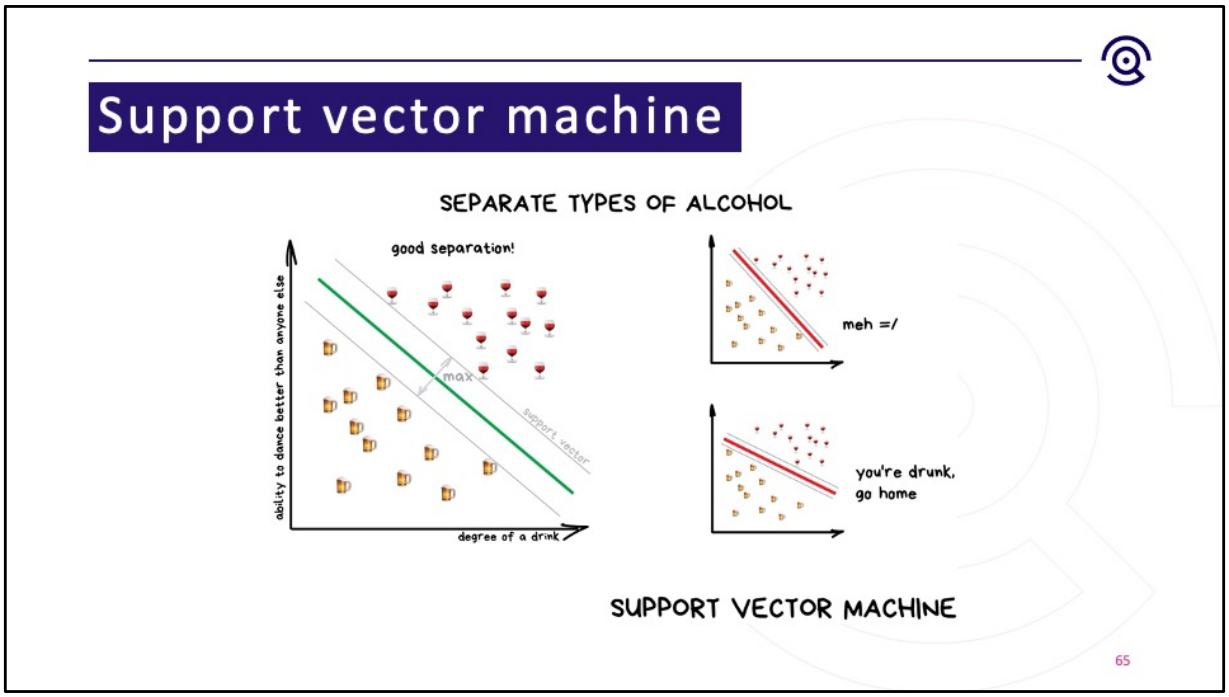

All pictures in this section are taken from vas3k's very interesting blog on machine learning algorithms. This is a recommended read for anyone who want to discover the basics of several algorithms and application areas in a fun way. The graph on the right shows an overview of many AI algorithms. The top group contains classical supervised and unsupervised algorithms and the group on the left lists reinforcement learning algorithms. The group on the right is called "ensemble" methods, which is a fancy way of saying that these models combine multiple models in some specific way. The bottom group shows several neural network and deep learning algorithms. There are simply too many algorithms to discuss, so in the following slides we will pick some of the very popular ones.




Linear or polynomial regression is a very simple supervised approach to predict numeric values. In the example here the goal is to find a model that predicts travel time based on the hour of the day. Data is collected and shown as blue dots and the resulting models are shown as a green line. The linear model is a simple straight increasing line, not able to capture the morning and evening peak moments. The polynomial model is more flexible and therefore able to capture these peak trends. The example here is shown in one dimension to keep it simple, but the same can be done in more dimensions. Imagine that we want to use both the current hour and precipitation to predict travel time. In that case the green lines become 2-dimensional planes with the hour in one direction and precipitation in the second.

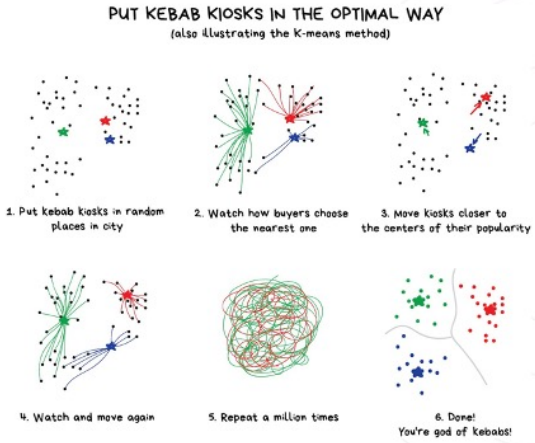



Decision trees are intuitive models with an if-then reasoning that is easy to grasp and display. Trees can be used for both supervised regression and classification problems. Here we have the example of a classification problem to decide whether someone should receive a loan or not. We start at the top of the tree and check the credit history of that person. If it is good we go to the right node, if it is bad we go to the left. Imagine the person has a good credit history, we then check whether there is an outstanding debt larger than 1000 dollars. If that is not the case we decide to grant the loan, otherwise we reject the loan application. The same process can be followed for someone with a bad credit history. Loans are for example rejected for someone with a bad history without a pledge. Decision trees result in this very simple structure with yes/no questions, making them very attractive from a transparency point of view.



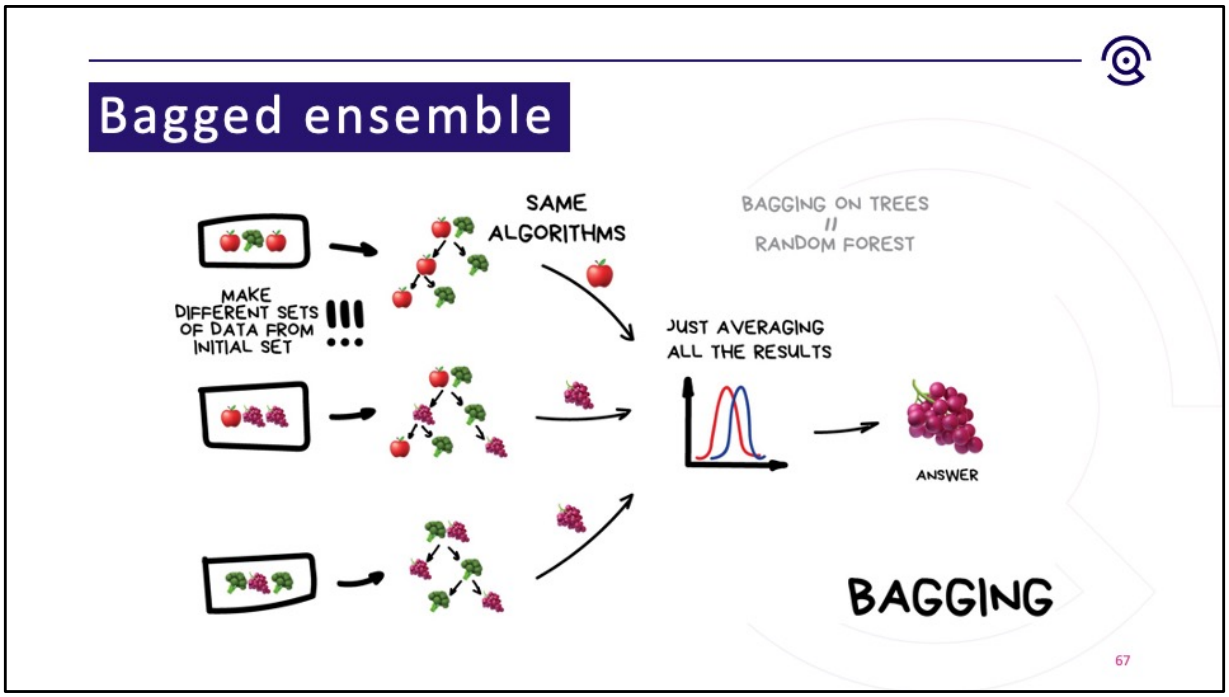

Support vector machines (SVMs) are a more advanced way to perform supervised classification problems. In this example the goal is to separate beer and wine based on some characteristics, namely the degree and ability to dance. SVMs try to reach an optimal separation by maximizing the distance of observations to the decision boundary. Below the boundary we predict beer and above the boundary we predict wine. The green line represents the best separation possible, whereas the red lines are also able to separate beers and wines, but in a less than optimal way.

# K-means clustering

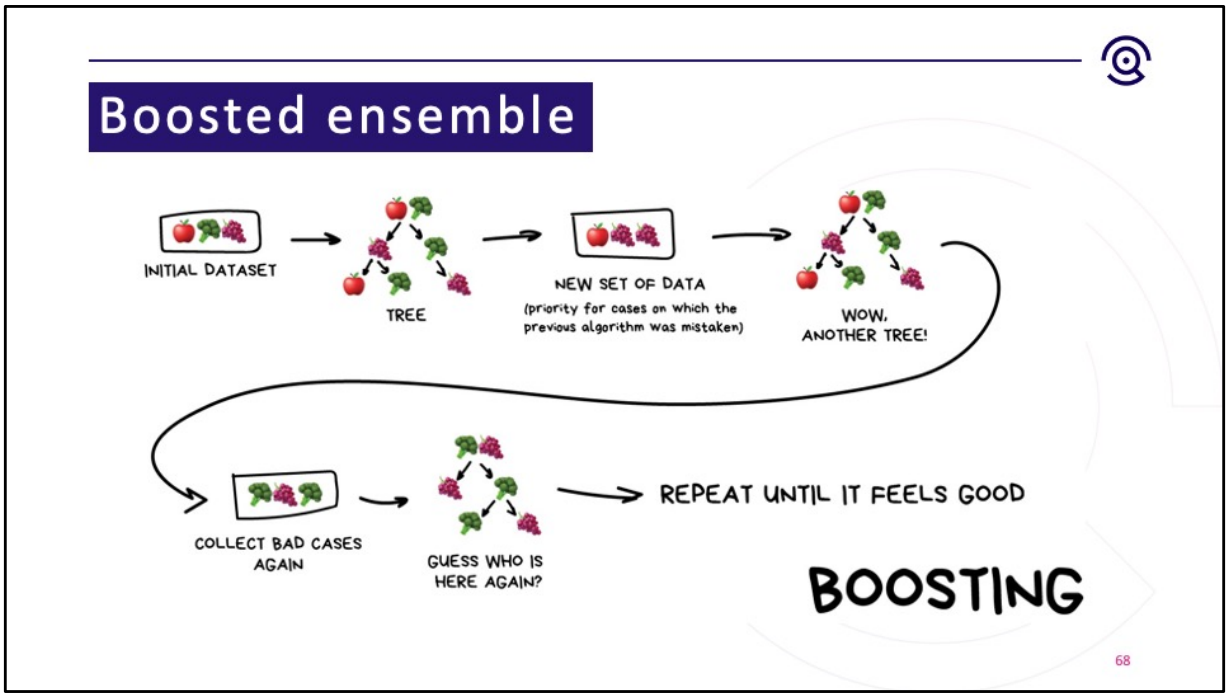



66

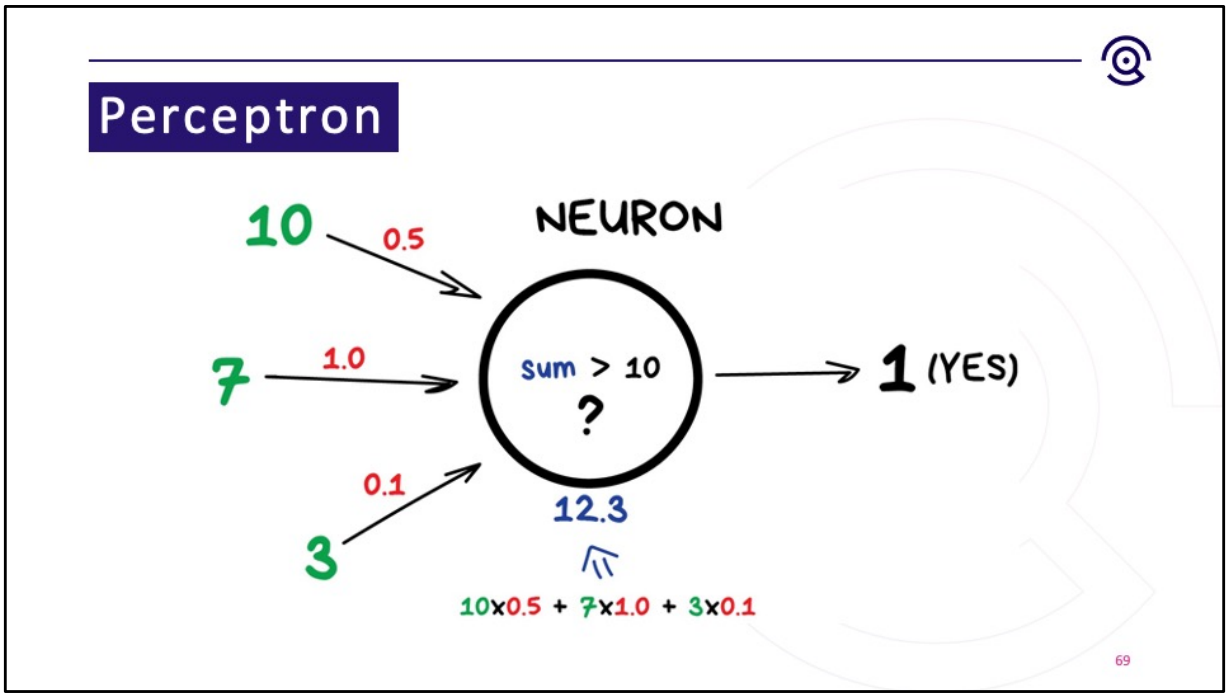
A popular way to perform clustering problems is the K-means algorithm. The example illustrates the K-means approach by putting kebab kiosks in an optimal spot. Assume the black dots are prospective kebab buyers and we want to group all buyers in three groups based on where they live. We start with three random locations for the kiosks, these are the starting points for the K-means algorithm. We then assign each buyer to the nearest kiosk and relocate the kiosks such that they are at the center of all their assigned customers. For the new kiosk locations, we again assign all buyers to the nearest kiosk and relocate the kiosks to the center of all their new buyers. These steps are then repeated until the kiosks don't move anymore. When this is ready you have three groups of customers and the kiosks are at the center of their respective buyer group. In more general terms this allows to group observations in a certain number of groups based on their respective distance to each other.



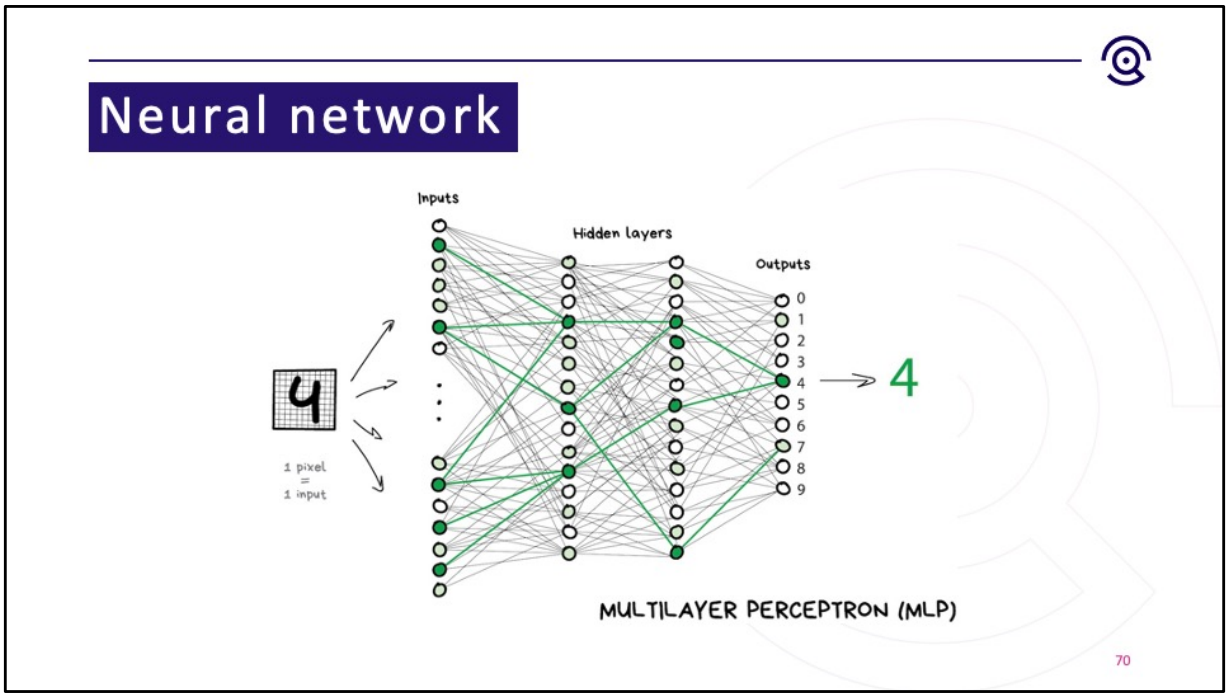

Bagging is a general ensemble technique to combine multiple models together. Several models are trained on different version of your data and afterwards the results of the separate models are combined. These separate models can be any type of model, for example decision trees, but they should all be of the same class. For regression one can average all the individual results and for classification one can take a majority vote. In the example the goal is to predict the type of fruit/vegetable based on three decision trees in the ensemble. One tree votes for ”apple” and two trees vote for “grapes”, so the final answer is “grapes” by majority voting. In reality the number of individual models will be much higher of course. Random forests are a popular technique nowadays and are basically bagged decision trees with some extra fancy detailed tricks.




Boosting is also a general ensemble technique to combine multiple models together. Bagging was a parallel approach, where each model was trained independently. Boosting is an iterative approach where we start from the initial data and train a model on this data. In the next step we then assess in which areas the model can improve most and we prioritize those areas to fit another model. These steps are repeated, each time improving the total model, until a good model fit is obtained. The performance is boosted in each iteration until it feels good. Both bagging and boosting are ensemble techniques often used with decision trees. The advantage is usually improved predictive performance, but this comes at the cost of less transparency. An ensemble of 100 trees is much harder to interpret than a single decision tree.

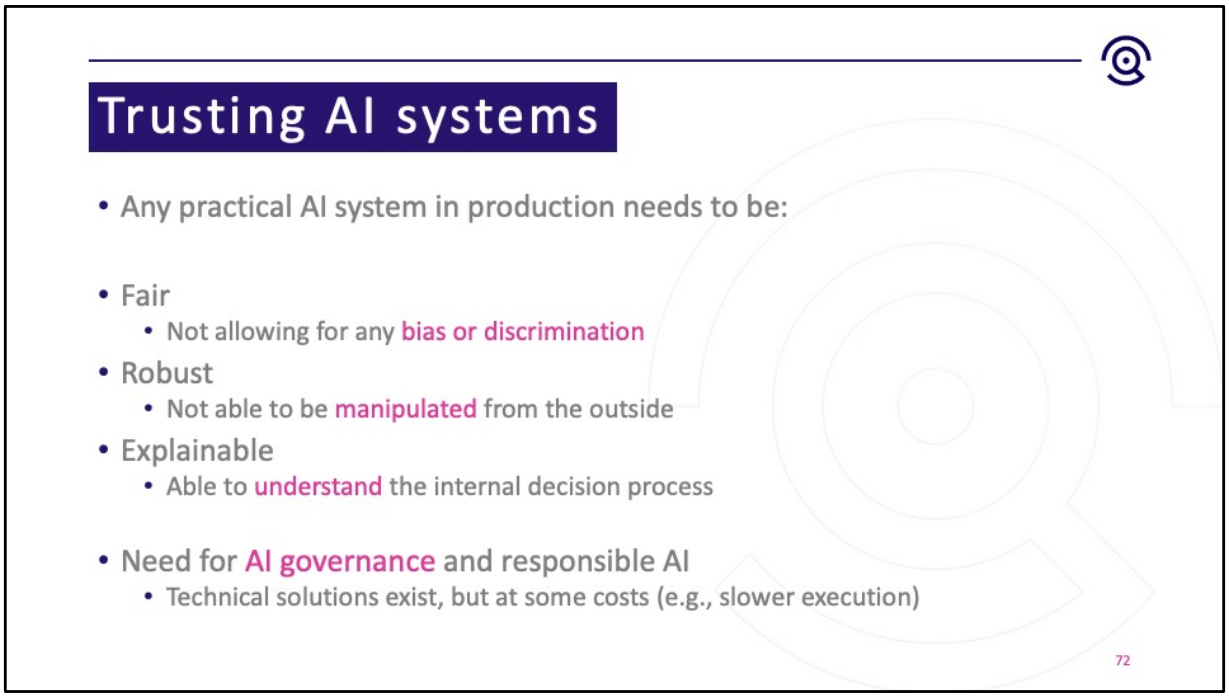



A single neuron or perceptron can be seen as a function which takes several inputs and produces one output. The exact function that it evaluates depends on design choices made by the modeler, many options exist here. In the above example, inputs (green) are multiplied by weights (red) and then summed together. If this sum exceeds the value of 10, then the neuron outputs a one (otherwise a zero). Very simple, right?! On it's own, a neuron or perceptron is not really used. The power lies in combining many of these neurons together in layers.

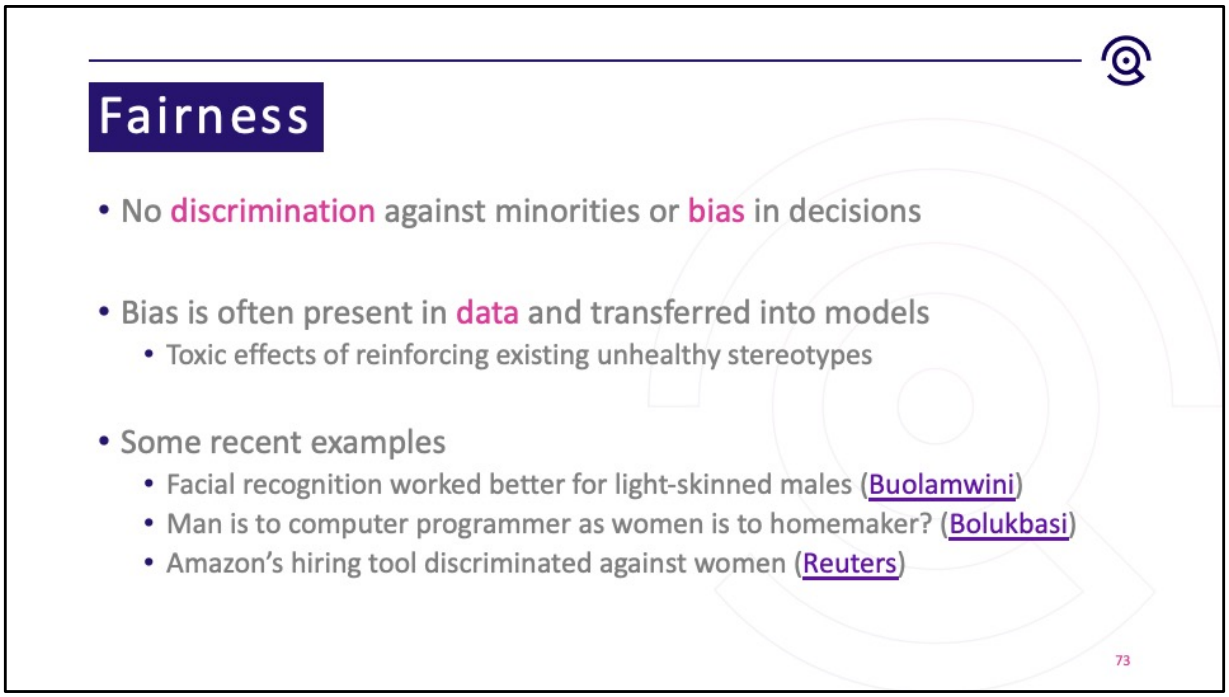



A neural network (NN) is the combination of many such neurons in different layers. There is always an input and an output layer, with one or more hidden layers in between. A NN with one hidden layer is called a shallow network, while more than one hidden layer gives rise to the so called deep NNs from deep learning. Each neuron is such a network performs a simple operation like we just saw, but combining these together has resulted in many of the groundbreaking AI results we have seen in the last decade. Deep NNs are very good in applications such as speech recognition, image classification and text translation. The above example shows how raw pixel values from an image are used to predict that we are looking at a handwritten 4 digit. Adding extra hidden layers allows a deep NN to massage the input into useful features for the prediction problem at hand. This automatic feature engineering is also know as representation learning and is one of the strong points of deep learning models.

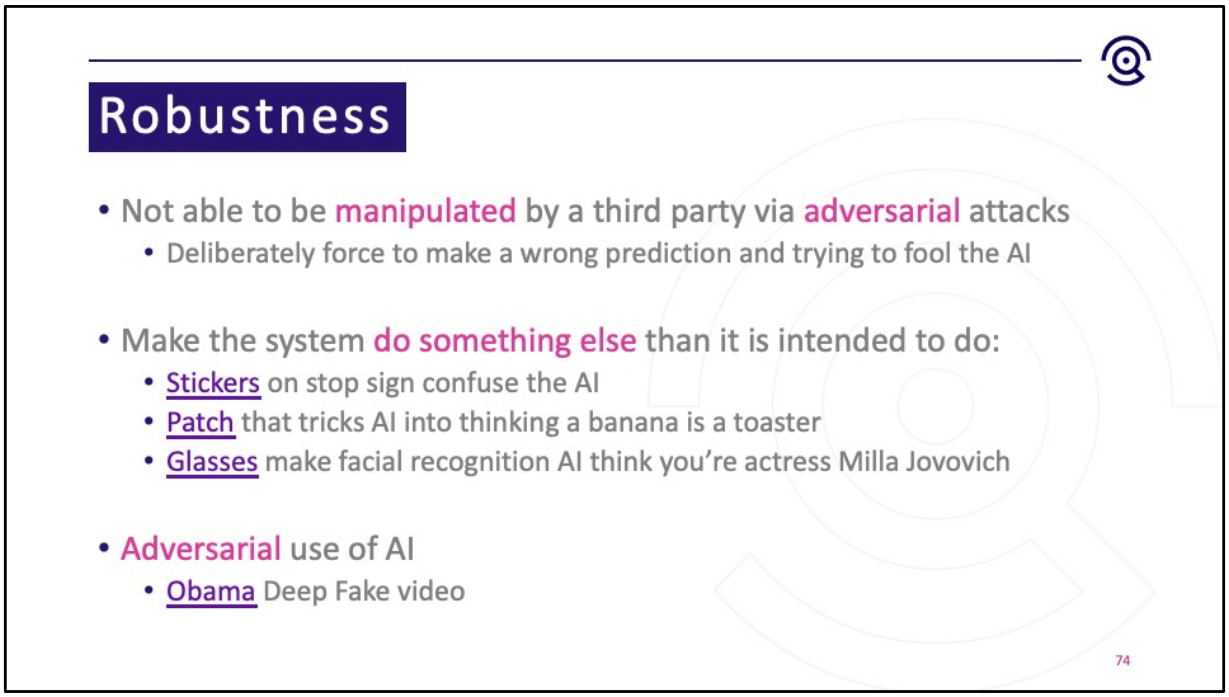

Let’s briefly discuss trust and ethics when it comes to AI applications.



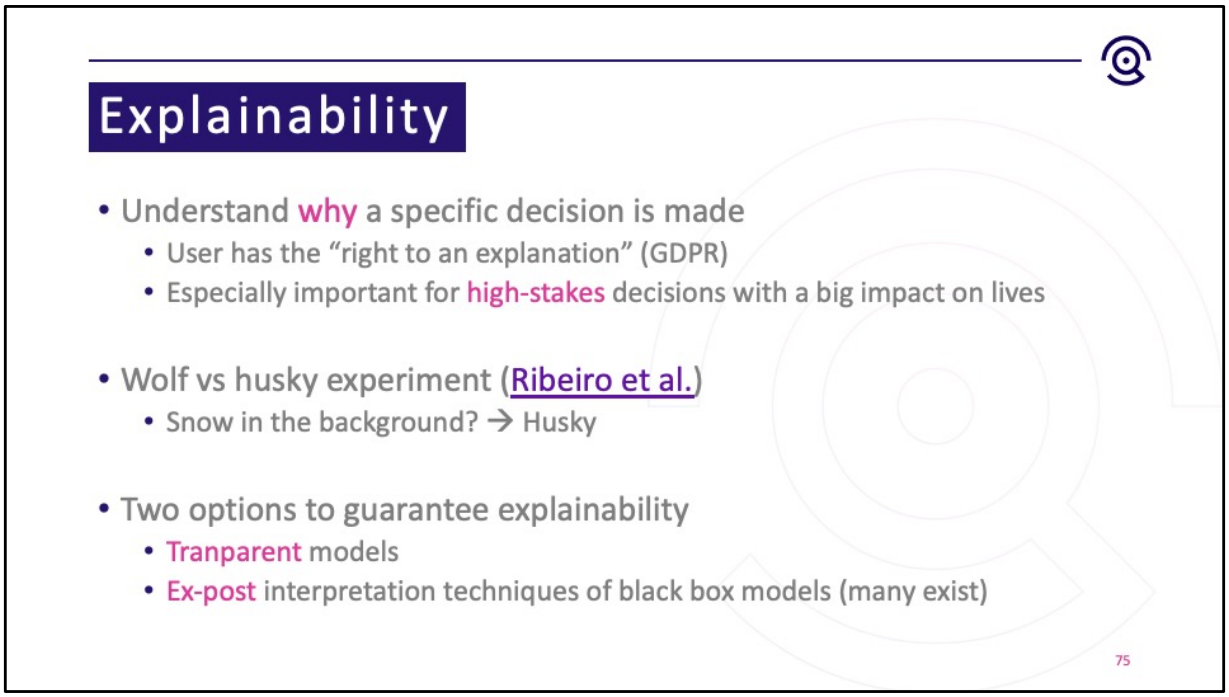

It is very important to trust any practical AI system that is put in production. These systems need to be fair without any bias or discrimination against certain groups of individuals. These systems need to be robust without being able to be manipulated from the outside. These systems need to be explainable such that one can understand the internal decision process. All of these aspects ask for proper AI governance and responsible AI processes. Technical solutions to all these issues exist, but typically come at some costs (e.g., slower execution). These aspects are however so important that the cost is usually justified.




Fairness implies that everyone is treated equally, without discrimination against minorities or bias in decisions. The problem is that bias is often present in train data and that way gets transferred into AI models. This has the toxic effect of reinforcing existing unhealthy stereotypes. Some recent examples showed this risk for discrimination or bias very clearly. Facial recognition models were found to work much better for light-skinned males compared to black males or females, simply because the train data contained more light-skinned males. Word embeddings from analyzing old texts resulted in the statement "man is to computer programmer as women is to homemaker", clearly pinpointing implicit sexism in those texts. Amazon's automatic hiring tool discriminated against women because the tech sector is predominantly occupied by males. In all these cases it is not the AI technology that is causing discrimination, but the problem lies in the data used to train these AI systems. It is therefore very important that train data is bias-free in order to avoid that the eventual AI decisions are discriminatory.

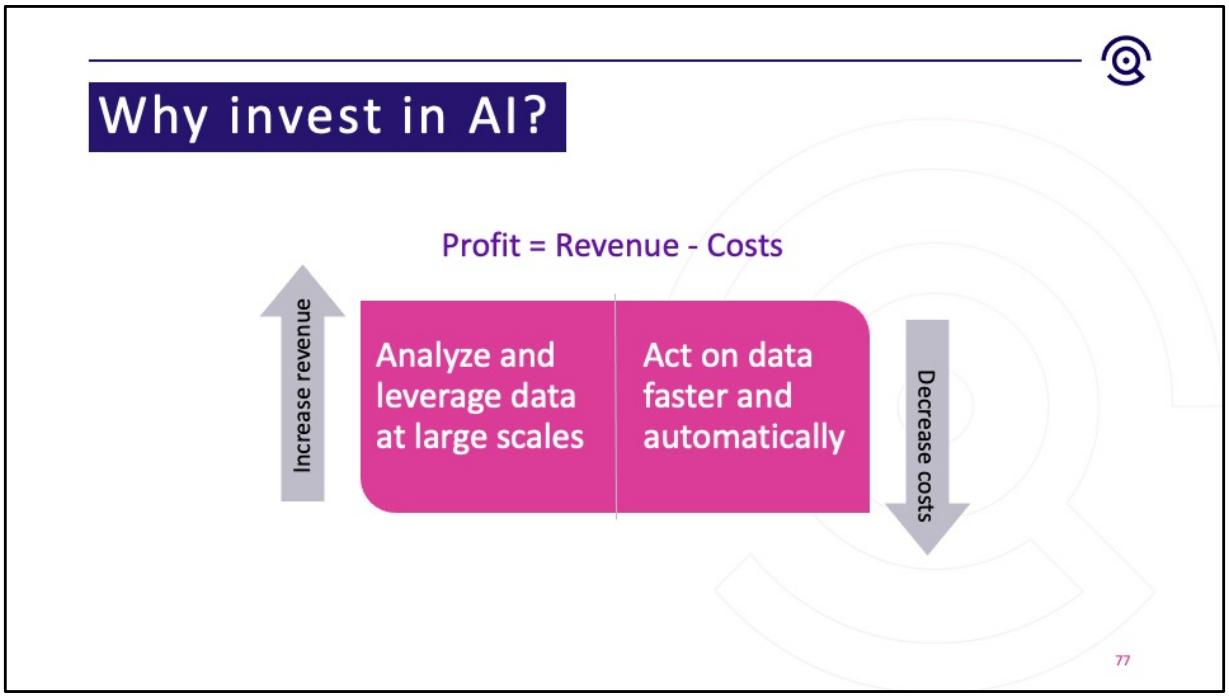



Robustness implies that an AI system should not be able to be manipulated by a third party via adversarial attacks. Intruders sometimes try to fool the AI and deliberately force it to make a wrong prediction. There are some recent examples where people were successful in making the system do something else than it was intended to. By placing stickers on a stop sign they were able to confuse the AI system such that it was not able to recognize the sign. This can have detrimental consequences for autonomous cars which heavily rely on such image recognition to guarantee driver safety. Researchers developed a patch that tricks AI systems into thinking a banana is a toaster when that patch was close to the banana. Other researchers developed glasses that make a facial recognition AI think that everyone is actress Milla Jovovich. All these examples have in common that the AI systems works good under normal circumstances, but there are some hidden flaws that can be triggered such that the predictions are completely wrong. AI technology can also be used for adversarial attacks, with the Obama Deep Fake video as most prominent example. Jordan Peele transferred his own facial movements to Obama's facial characteristics using deep fake technology. Such false images or videos seem deceptively real, making it very hard to judge what is real and what not. Any technology has the risk of being used for adversarial applications and AI is no different in that sense.

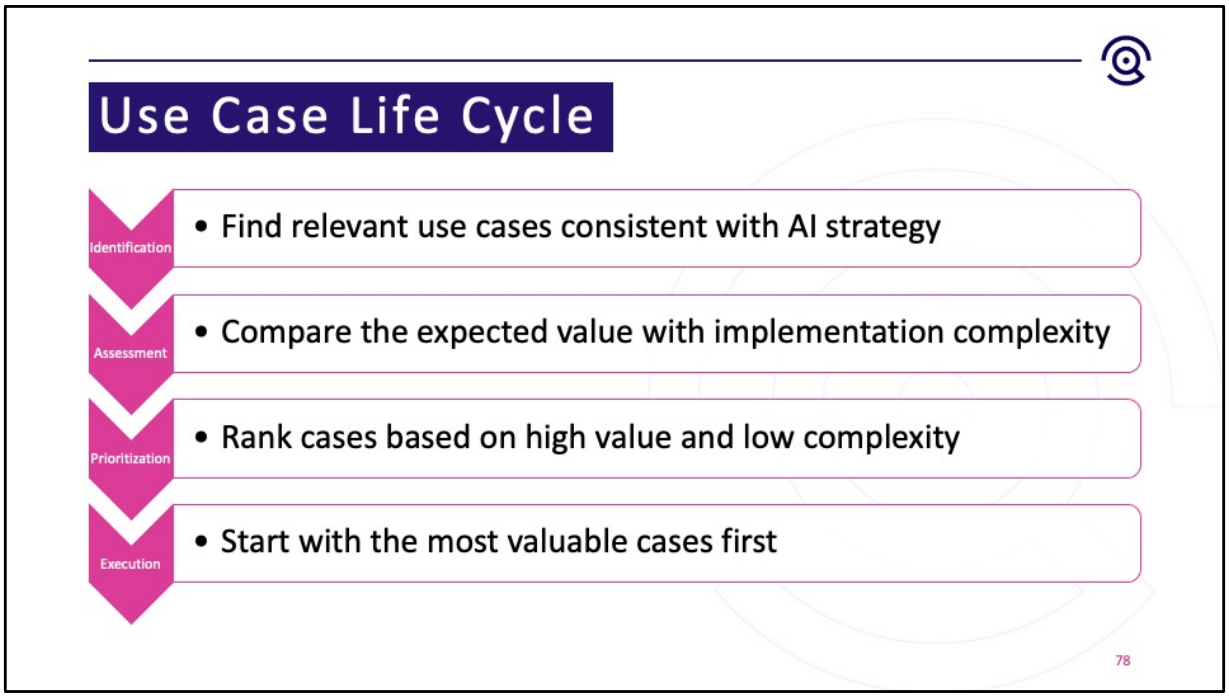



Explainability entails that it is important to understand why an AI system makes a certain decision. With new regulations like the GDPR, the user has the right to an explanation. This means that every AI decision should be supported by a transparent decision process. This is especially important for high-stakes decisions with a big impact on people's lives. Imagine applications such as loan grants, education admissions, insurance coverage or recruitment processes. All those decisions can determine the course of someone's life so an explanation of the AI's decision is extremely important. Researchers showed how an AI system was successful in correctly classifying images of wolves and huskies. However, when they looked at model explanations they noticed that the AI looked whether there was snow in the background or not and based its decision on that. Even though the classification works, this is not the underlying decision process that one wants. There are two ways to guarantee model Explainability. The first approach is to use transparent models, for example a decision tree or linear regression. There are easily understandable but have the typical downside of low predictive performance. The other approach is to use more complex black box models with better performance and then use ex-post interpretation techniques to explain the decisions. Many techniques exist for this, going from global explanations of the full model to local explanations of individual predictions.


Let’s see how to pick AI use cases with some typical application examples.




Before picking any use case, you might ask yourself the general question “Why should I invest in AI?”. In very simple terms, profit equals revenue minus costs. AI makes it possible to analyze and leverage data in an organization at large scales, which can result in increased revenue. AI also allows a company to act on data faster and automatically, thereby decreasing operating costs. AI therefore has a potential double positive effect on the profit via both increased revenue and decreased costs.



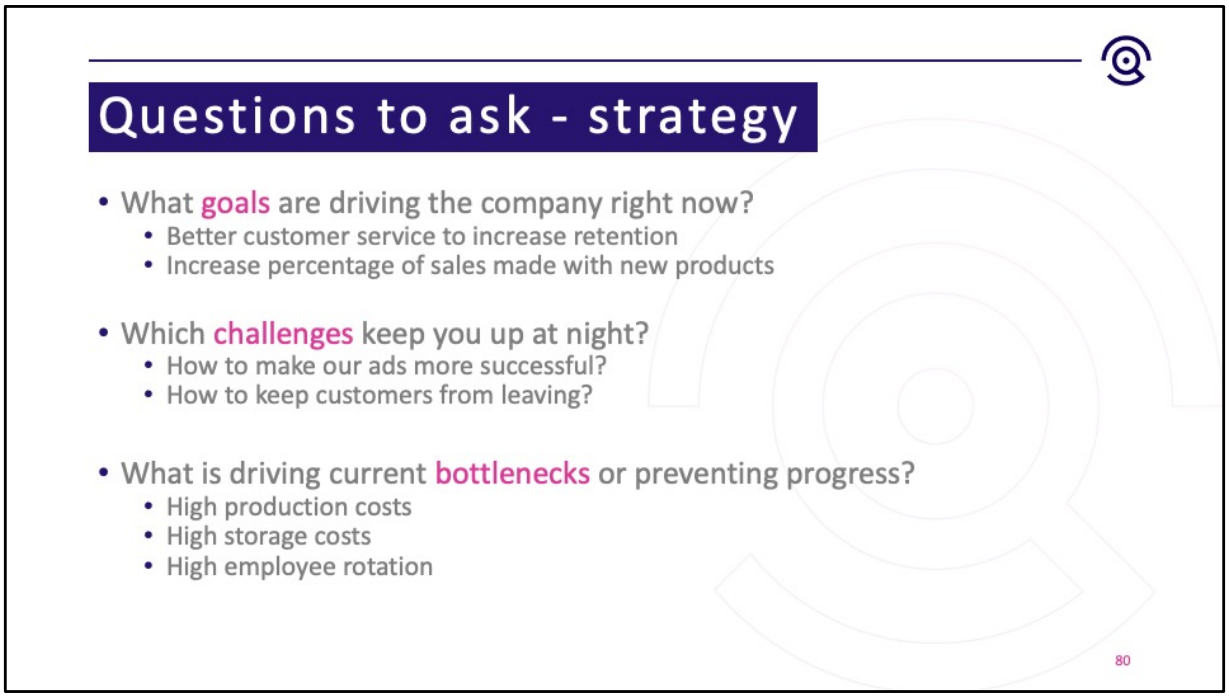

Once a company decides to invest in AI, it is important to start picking AI use cases to develop. The use case life cycle typically consists out of four steps. Step one is the identification of relevant use cases consistent with the company's AI strategy. Step two is the assessment of each use case by comparing the expected value with the implementation complexity. Step three is the prioritization of use cases by ranking them based on high value and low complexity. Step four involves the execution of use cases, starting with the most valuable cases first.


The first, and usually hardest, step is the identification of use cases. Many companies starting their AI journey find it very difficult to identify relevant use cases that bring value. It is therefore very important to talk to the right people and bring together domain experts, business stakeholders and AI experts. Everyone brings their own view and expertise to the table which is important to align business goals and technical possibilities. Make sure that the initiatives address broad organizational priorities and tackle relevant business problems. Doing AI just for AI's sake can be fun but brings little value to the business, although some experimentation in the very beginning can be a good playground to learn. When AI is very new in the organization, one can increase adoption chances by involving the end users in the application design. This also helps to bring knowledge from these users on what works and what doesn't into the AI application.

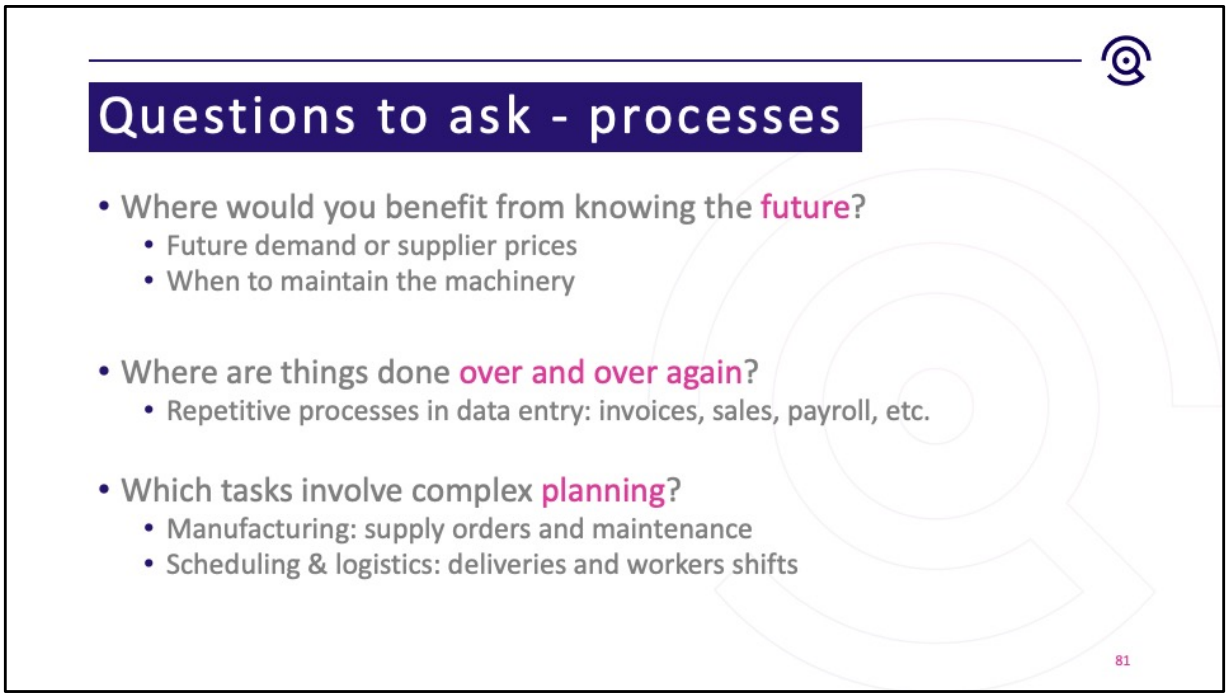
Once a group of people is formed, brainstorm sessions can be very useful to keep communication lines open and produce useful ideas. In this stage it is important to defer judgement and welcome wild ideas. It might sounds crazy in the beginning, but it might work? Build on each other ideas, but stay on target and don't lose focus of the higher business goals. Always go for quantity at this point, more is better and the filtering can happen later.

If the company is not AI-ready at the moment and lacks the people to perform these

steps, it might be a good idea to bring in external expertise. This way the AI journey can already start while the in-house AI team and expertise is gradually developed over time.
When trying to identify useful business cases, there are some questions that one can ask in the brainstorm sessions.

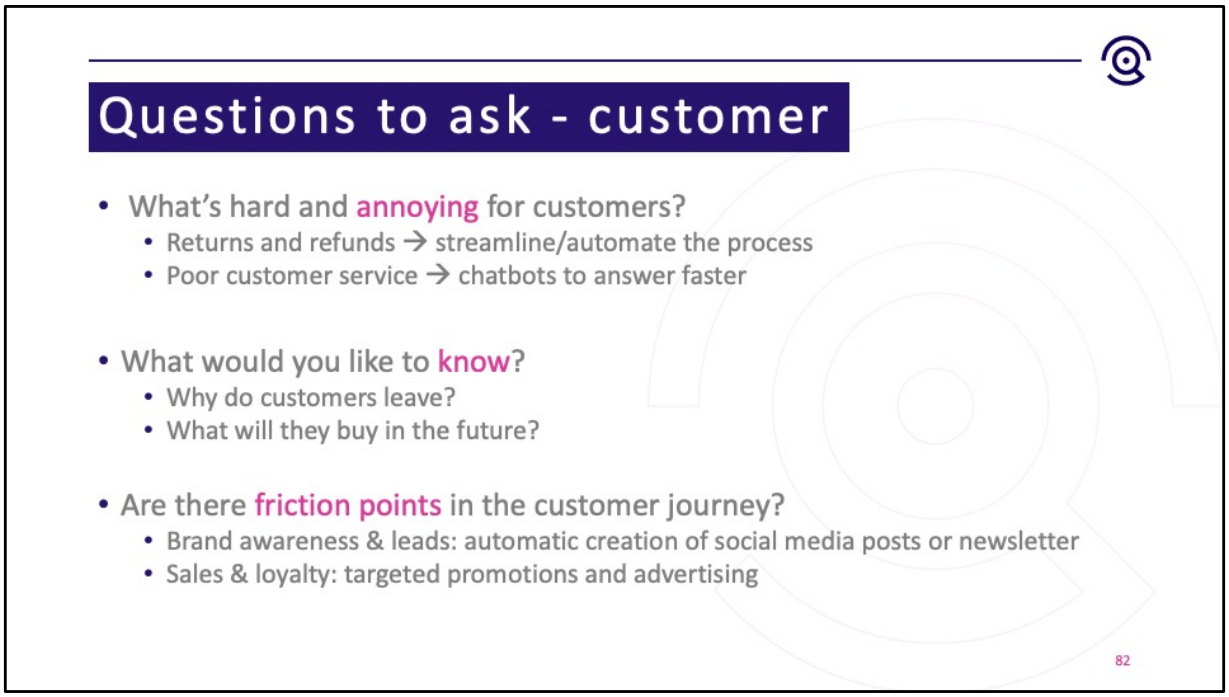



Some strategy questions are the following ones. What goals are driving the company right now? Maybe better customer service to increase retention or increase percentage of sales made with new products. Which challenges keep you up at night? How to make our ads more successful? Or how to keep customers from leaving? What is driving current bottlenecks or preventing progress? Maybe high production costs, storage costs or employee rotations.

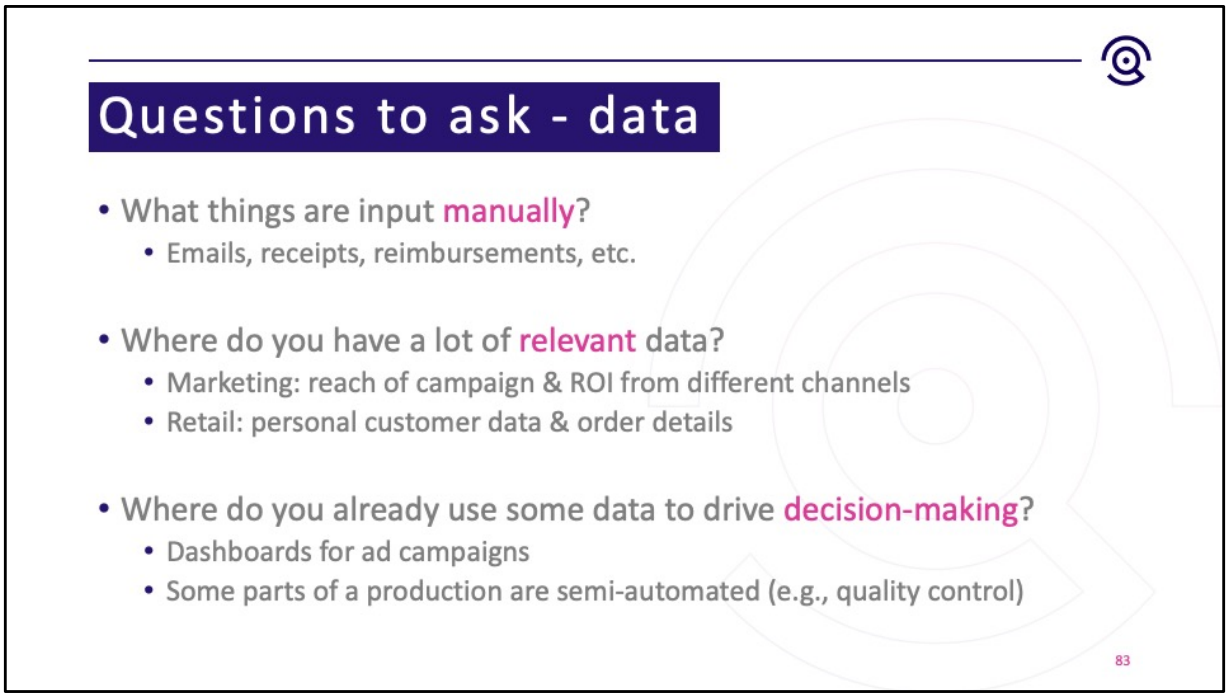



Some questions on processes are the following ones. Where would you benefit from knowing the future? Maybe future demand or supplier prices or when to maintain the machinery. Where are things done over and over again? For example repetitive processes in data entry: invoices, sales, payroll, and

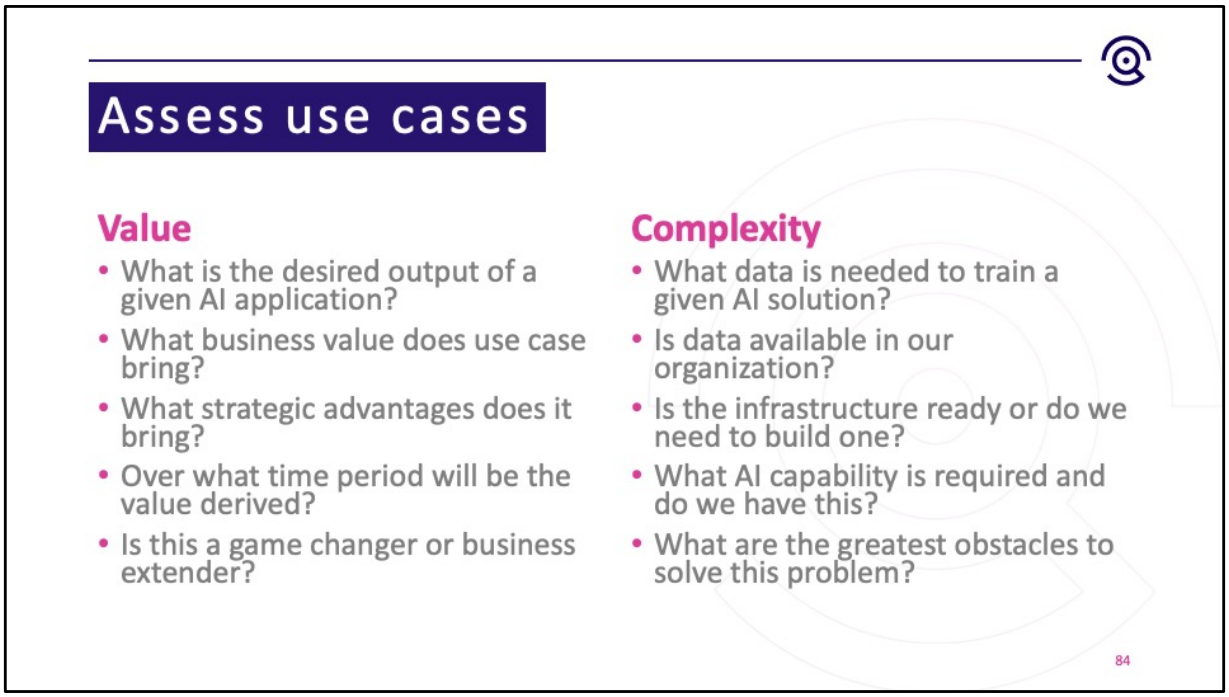
so on. Which tasks involve complex planning? In manufacturing these might be supply orders and maintenance, while in scheduling & logistics these might be deliveries and workers shifts.



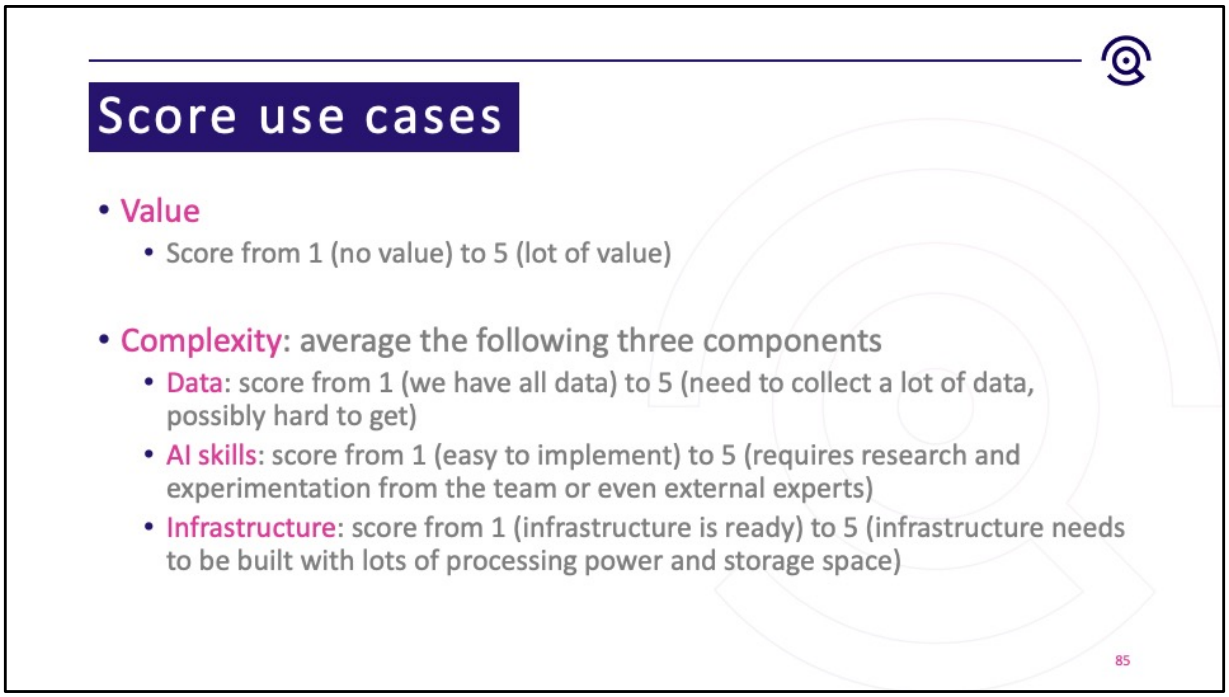

Some questions you can ask regarding your customers are as follows. What's hard and annoying for customers? Returns and refunds are never fun, so maybe we can streamline/automate the process. Poor customer service drives away customers so maybe chatbots can be used to answer faster. What would you like to know? Maybe why customers leave or what they will buy in the future. Are there friction points in the customer journey? For brand awareness & leads one can think about the automatic creation of social media posts or a newsletter, while targeted promotions and advertising might help with sales & loyalty.




Some questions regarding data are the following. What things are input manually? For example e-mails, receipts, reimbursements, and so on. Where do you have a lot of relevant data? In marketing this can be the reach of campaigns and the ROI from different channels. In retail this can be personal customer data and order details. Where do you already use some data to drive decision-making? Maybe dashboards for ad campaigns or some parts of a production process are semi-automated (e.g., quality control). These can then be taken a notch further and be fully automated or data-driven.



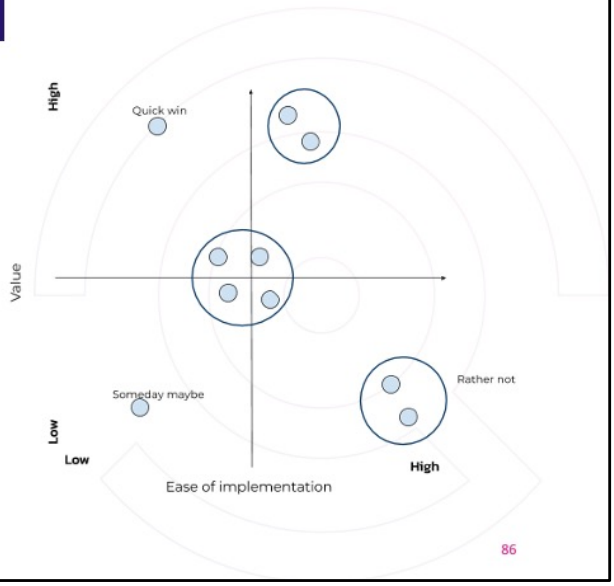

After identifying several relevant use cases, it is time to assess them on both the expected value and the implementation complexity. Here there are again a couple of useful questions to ask regarding both the value and complexity.
In terms of value the following questions need to be answered. What is the desired output of a given AI application? What business value does use case bring? What strategic advantages does it bring? Over what time period will be the value derived? Is this a game changer or business extender?
In terms of complexity the following questions are important. What data is needed to train a given AI solution? Is data available in our organization? Is the infrastructure ready or do we need to build one? What AI capability is required and do we have this? What are the greatest obstacles to solve this problem?
It is good to be as specific as possible when answering these questions, that also forces you to really think about these questions.




After assessing the value and complexity it is time to score the use cases. For value we assign one score to each use case, ranging from 1 (no value) to 5 (lot of value). We are clearly mostly interested in use cases with a high value score. For complexity we average three separate components. The data component receives a score from 1 (we have all data) to 5 (need to collect a lot of data which is possibly hard to get). The AI skills component receives a score from 1 (easy to implement) to 5 (requires research and experimentation from the team or even external experts). The infrastructure component receives a score from 1 (infrastructure is ready) to 5 (infrastructure needs to be built with lots of processing power and storage space). We are clearly happier with a use case that has low scores on all these complexity components.

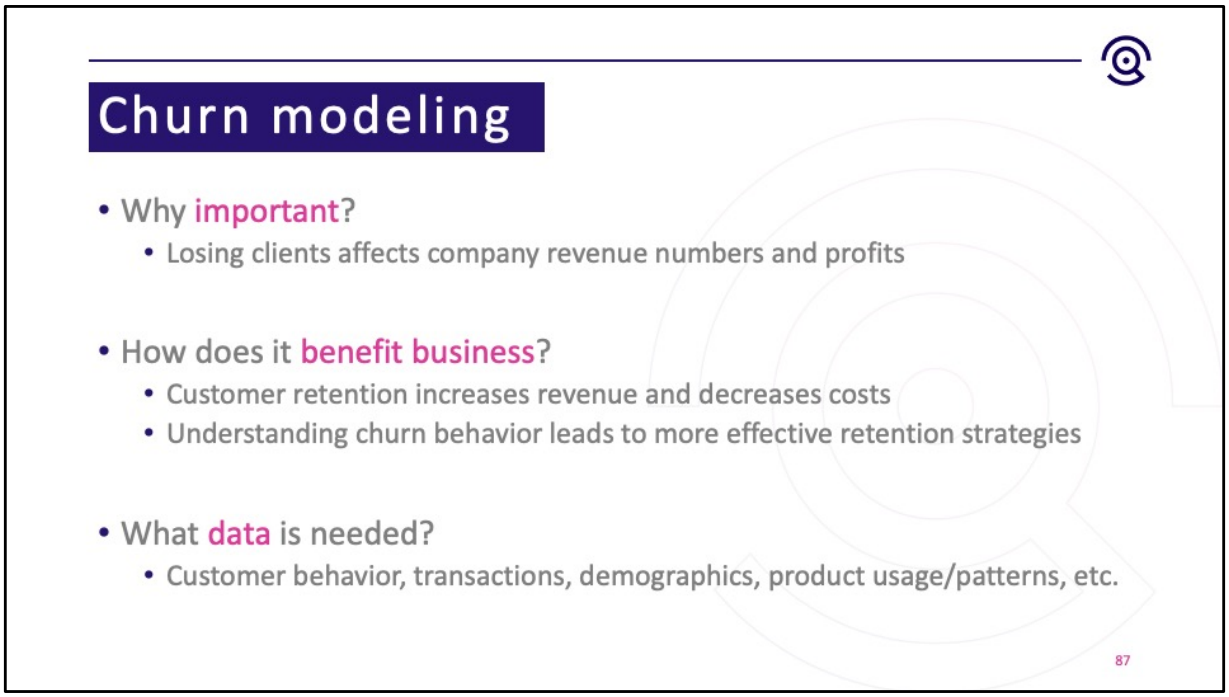
# Prioritize use cases

- Rank individual cases according to value and complexity scores
  - Plot value vs complexity
- Cluster use cases that are close
  - Prioritize clusters by value added and number of use cases
- Are there any quick wins?
  - Start with these

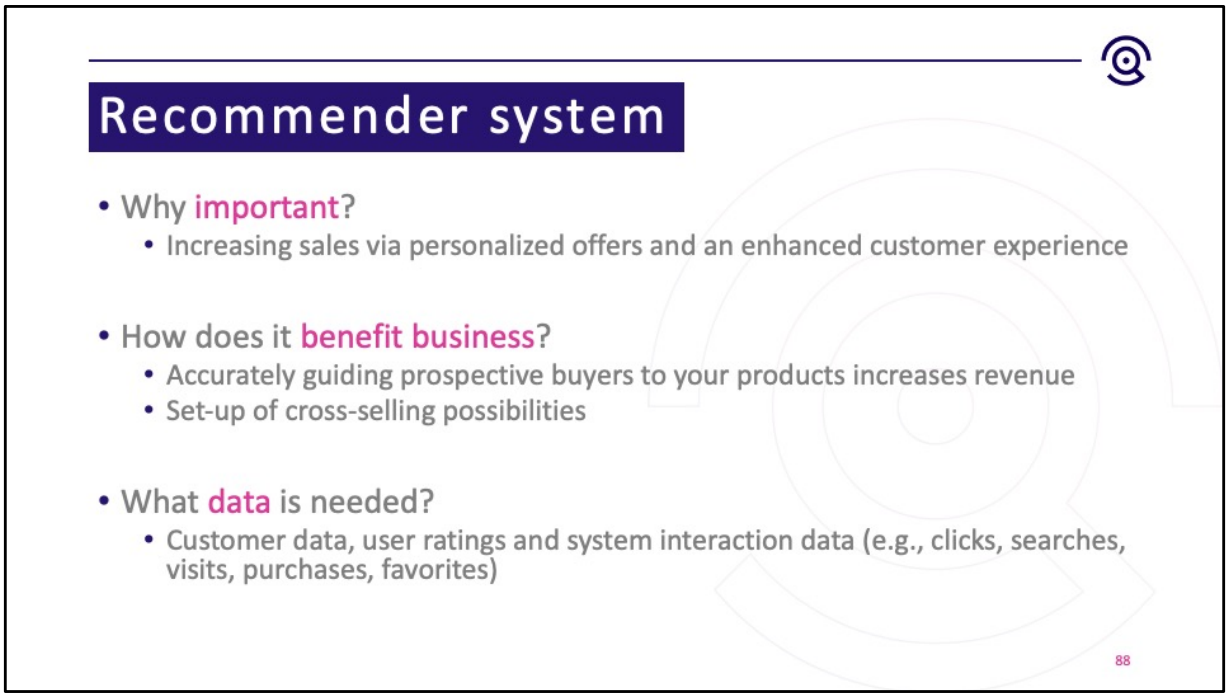



86

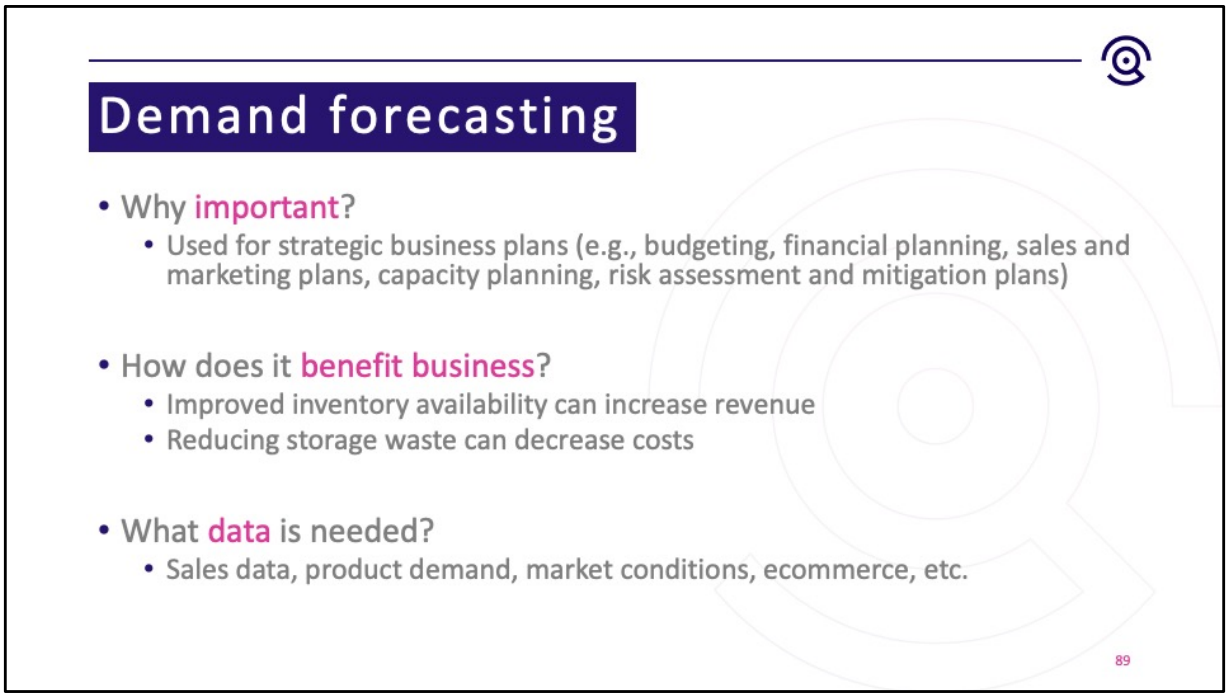
When it comes to prioritizing use cases it is important to rank use cases according to their value and complexity scores. One can plot the value and complexity score of each use case in a 2-dimensional graph. Optionally one can cluster use cases that are close in terms of complexity and value and prioritize clusters by value added and number of use cases. If there are any quick wins with high value and easy implementation, one should start with these. Low value and difficult implementation are use cases that we rather not pursue. The other use cases require a trade-off between value and complexity to decide which ones to tackle first. Let's now have a look at some popular AI use cases, this might give you inspiration for your own use cases.



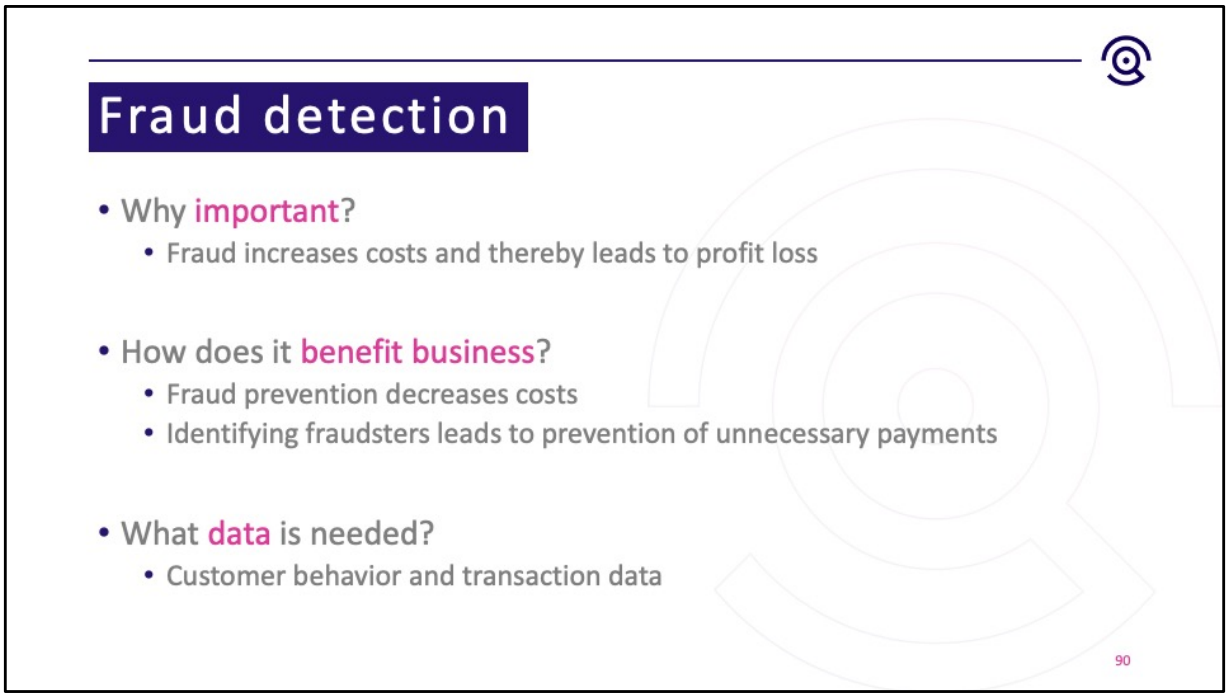

Churn modeling is the process of predicting which customers are likely to cancel their service and leave your company as a client. This is a very important problem as losing clients affects company revenue numbers and profits. Customer retention benefits business results by increasing revenue (more customers) and decreasing costs (cheaper to keep customers than to find new ones). A proper understanding of churn behavior leads to more effective retention strategies.Typical data needed for these kind of problems are customer behavior, transactions, demographics and product usage/patterns.




Recommender systems try to predict what someone will like. This is very important to increase sales via personalized offers and an enhanced customer experience, think about online shopping suggestions. This benefits the business by accurately guiding prospective buyers to your products which in turn increases revenue. It also allows to set-up of cross-selling possibilities by offering additional products to existing customers. Typical data entails customer data, user ratings and system interaction data (e.g., clicks, searches, visits, purchases and favorites).

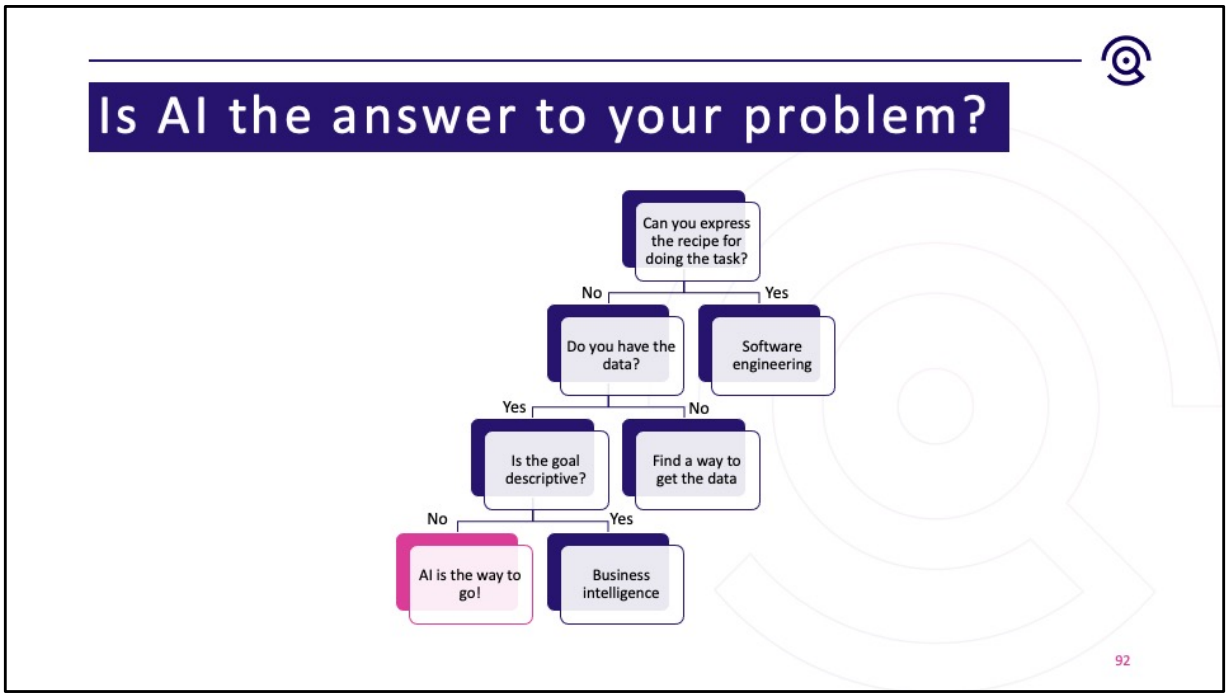



Demand forecasting deals with anticipating future demand for products. Accurate demand forecasts are important as these are used in strategic business plans (e.g., budgeting, financial planning, sales and marketing plans, capacity planning, risk assessment and mitigation plans). Demand forecasting benefits the business in several ways. Improved inventory availability can increase revenue and reducing storage waste can decrease costs. Typical data used for demand forecasting are sales data, product demand, market conditions, and e-commerce.




Fraud detection is the process of predicting which transactions are fraudulent. This is very important as fraud increases costs and thereby leads to profit loss, think about fraudulent insurance claims which are paid out. Fraud prevention benefits business results by decreasing costs as identifying fraudsters leads to prevention of unnecessary payments. Typical data used in these kind of problems are customer behavior and transaction data.


Targeted advertising is a way of advertising a product to a specific audience with certain traits. This is a cost-effective approach as it minimizes wasted advertising. The business benefits as sales go up and customer satisfaction increases (less annoyed by all those random ads). Targeting successfully therefore leads to increased revenue and decreased costs. Typical data used in these kind of problems are purchase history, client personality, attitude, opinions, lifestyle and interests.




Not all problems need AI of course, some problems might be solved in other ways. This tree diagram shows a flow to decide whether AI is the answer to your problem. If you can express a recipe for doing the task, then we are dealing with software engineering problems. If this is not the case you should ask yourself “Do I have the data?”. If this is not the case you should find a way to get the data. If this is the case then the final question is whether the goal is descriptive or not. If the goal is simply to describe/visualize historical data and trends, then business intelligence is the solution. If on the other hand the goal is to predict the future (predictive) or form decisions (prescriptive), then AI is the way to go!


This is the end of the first module. I hope you now have a clear understanding of the AI basics and what AI actually means and can do. I hope to see you again in the next module, bye.