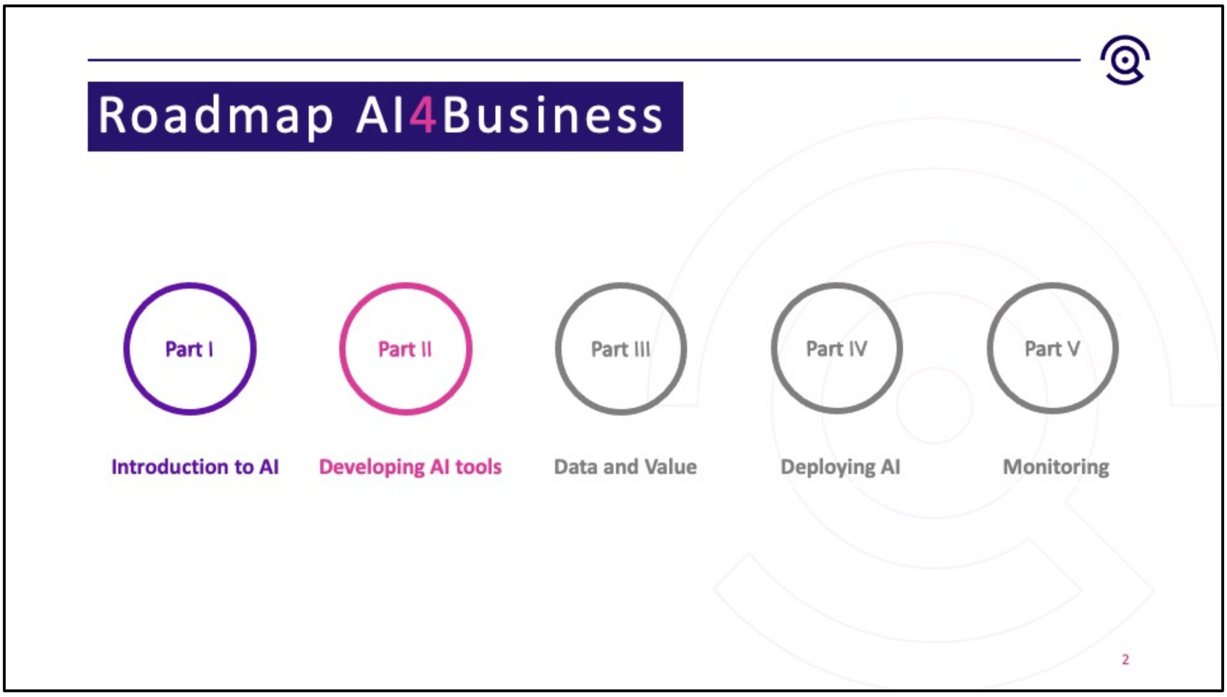
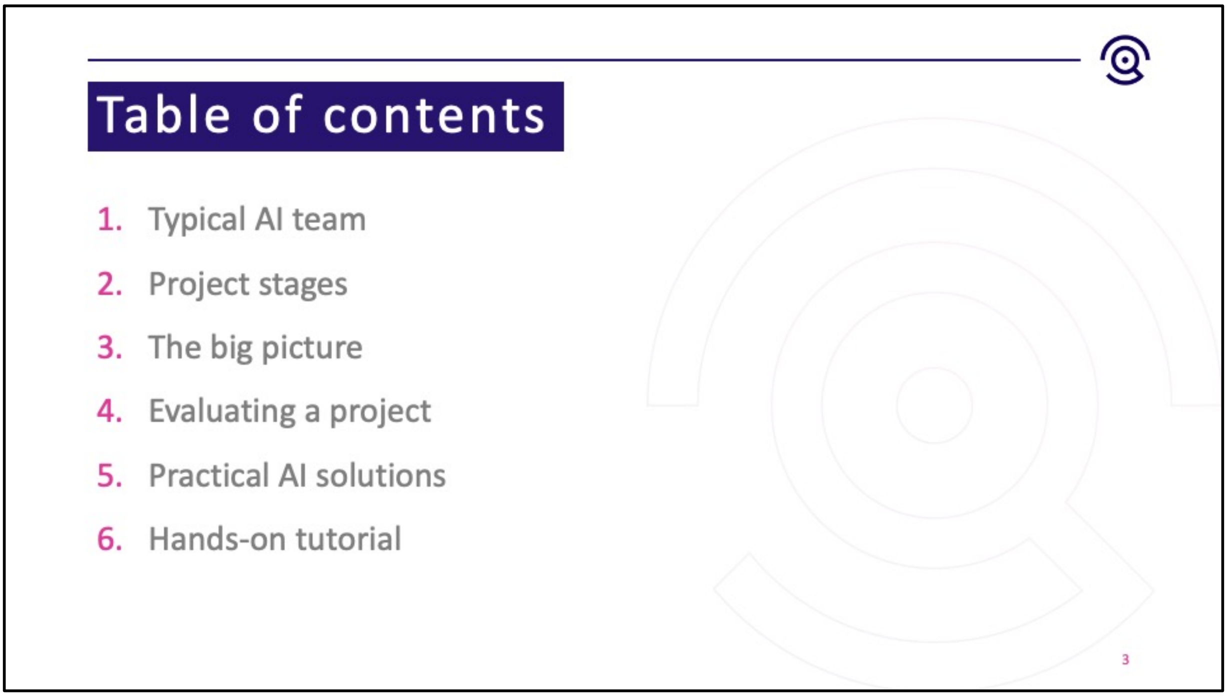
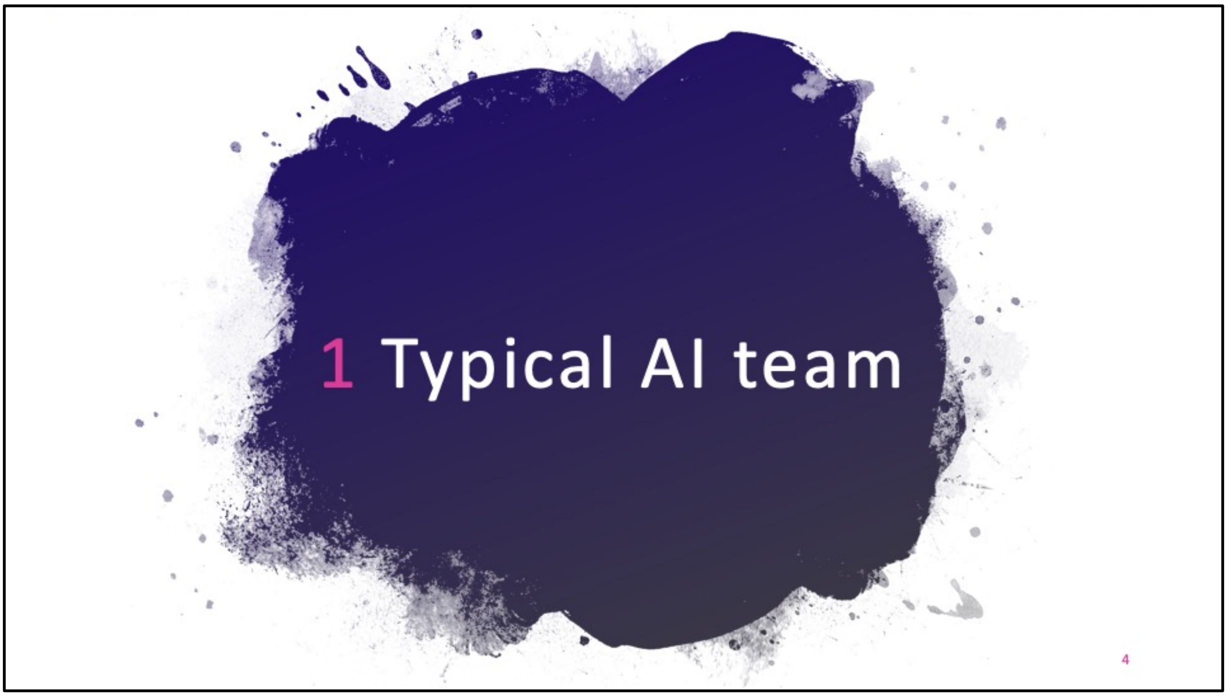
Welcome to the second module of the AI4Business course. In this module we give an overview of several important aspects in the development of AI tools. The first part of this module discusses how to successfully set up and structure AI projects. We highlight the typical type of people involved in an AI team and illustrate an AI workflow from start to end. The second part of this module is a hands-on demonstration of such a workflow where we develop AI models in a Jupyter notebook with Python. The focus is not on learning how to code, but simply to show how the "theoretical" concepts translate into practice.



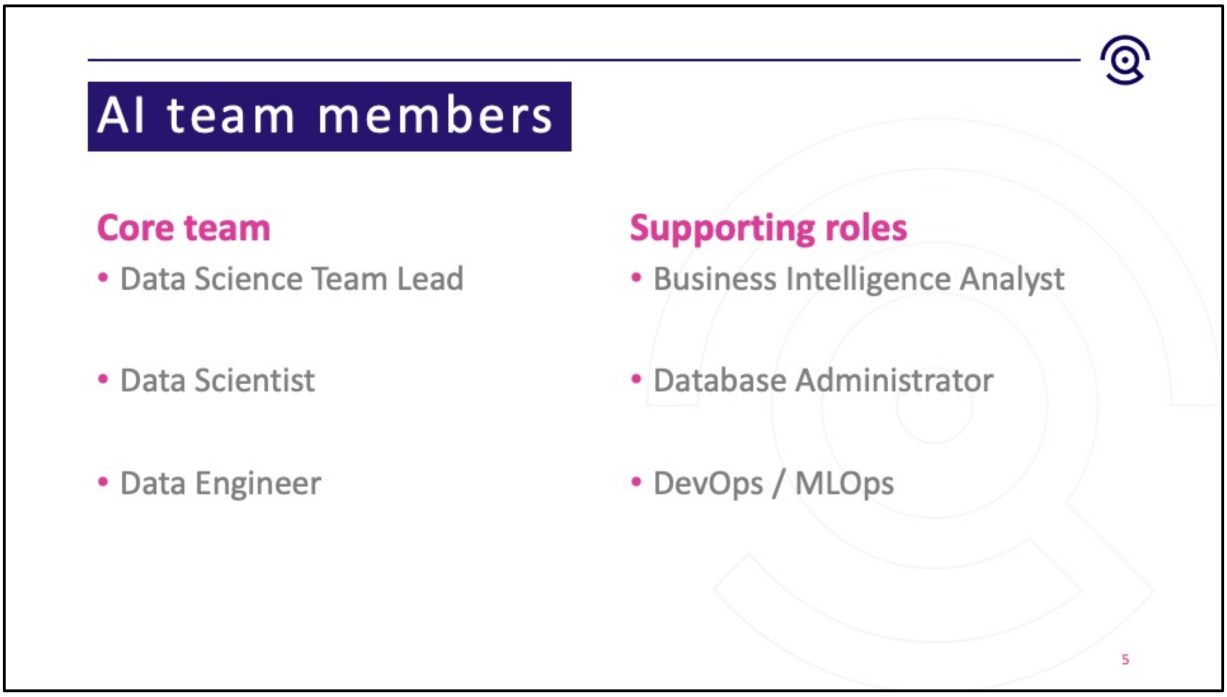
This module is the second out of five modules in the AI4Business course. The first module gave a general introduction to the most important AI concepts. We will use that knowledge today and learn how to develop AI tools from a high-level view. The next modules build further on what we learn today and tackle some aspects in more detail.



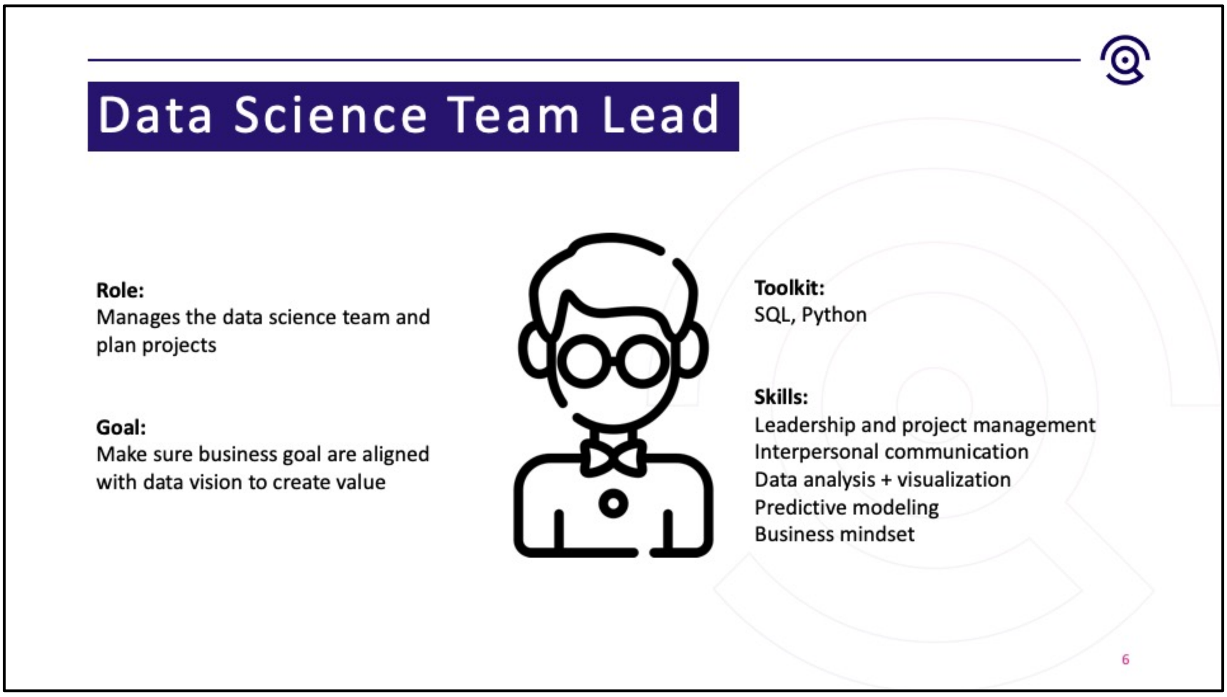
Developing AI tools is a journey. We start by discussing the typical people involved in an AI team. We then highlight the stages that an AI project flows through and illustrate this with a big picture example. We also focus on how to evaluate an AI project after completion via key performance indicators (KPIs). Finally, we discuss a framework to develop practical AI solutions developed by our own Prophecy Labs, which has been proven successful in the past. This module is wrapped up by a hands-on tutorial in Python where we bring all the theoretical material to practice.



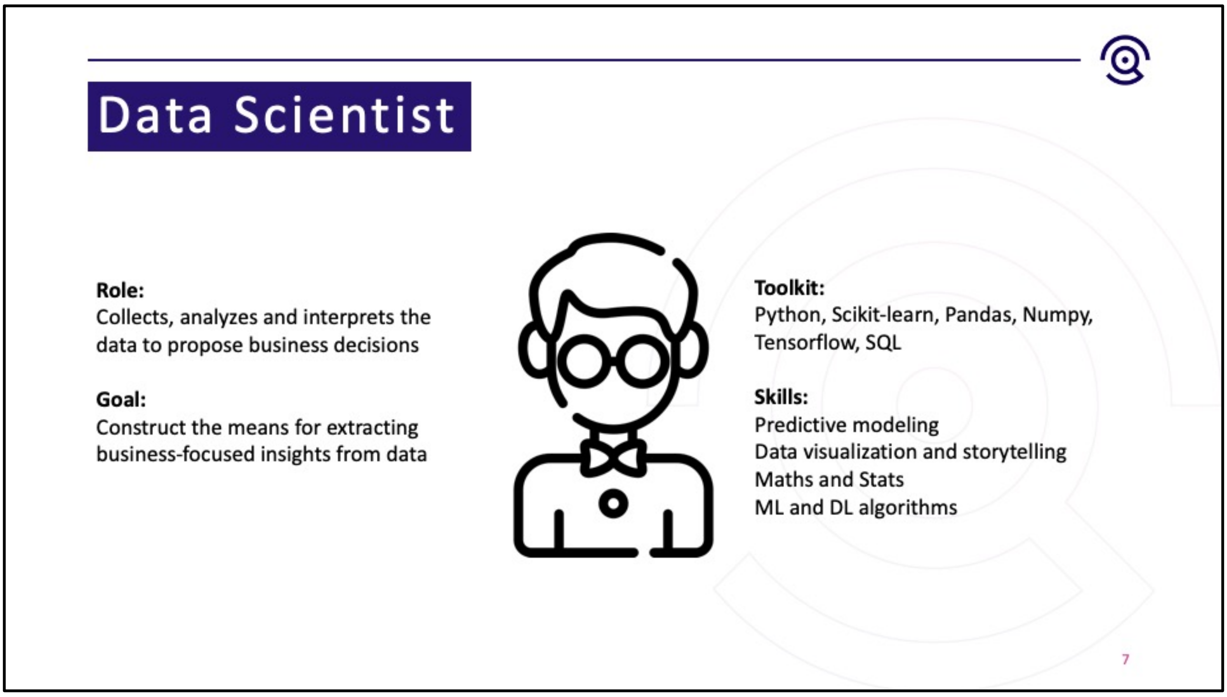
Let’s see which people are typically part of an AI team.



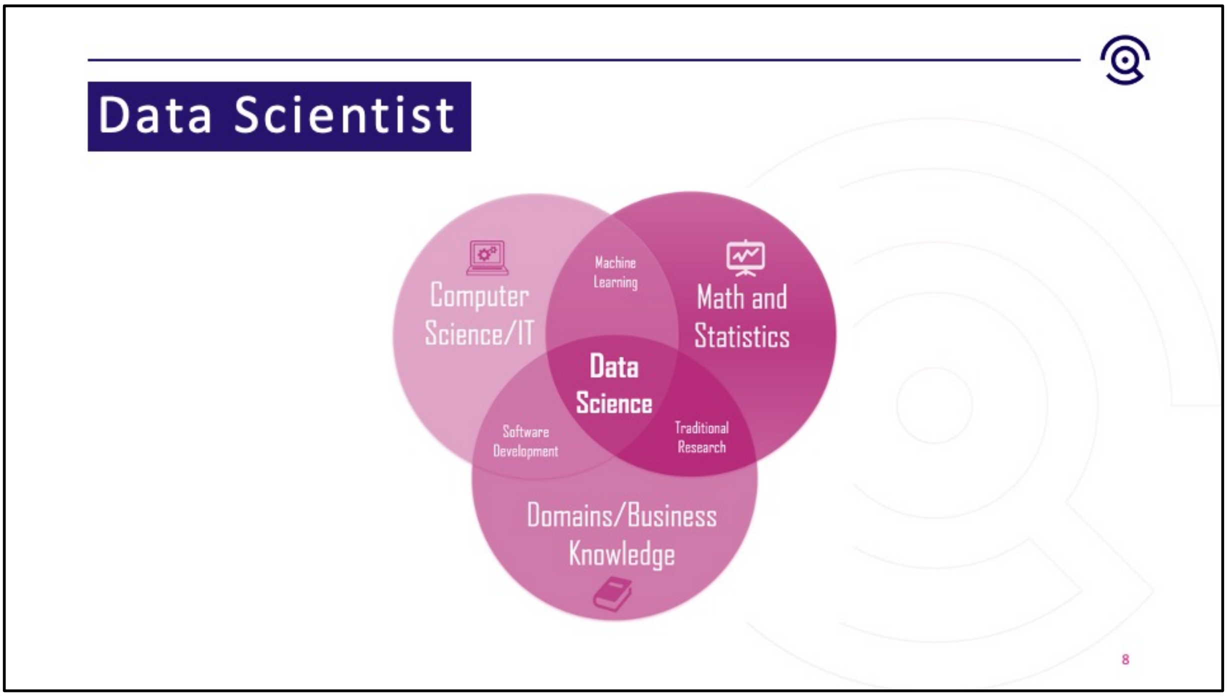
We can make a distinction between core team members and supporting roles. Both groups are very important for the project's success of course. A small AI team should at least have the following functions represented. The core team consists of a data science team lead, a data scientist and data engineer. In the supporting roles we have a business intelligence analyst, a database administrator and someone with DevOps/MLOps skills. Notice that a big team will contain multiple data scientists or engineers for example.



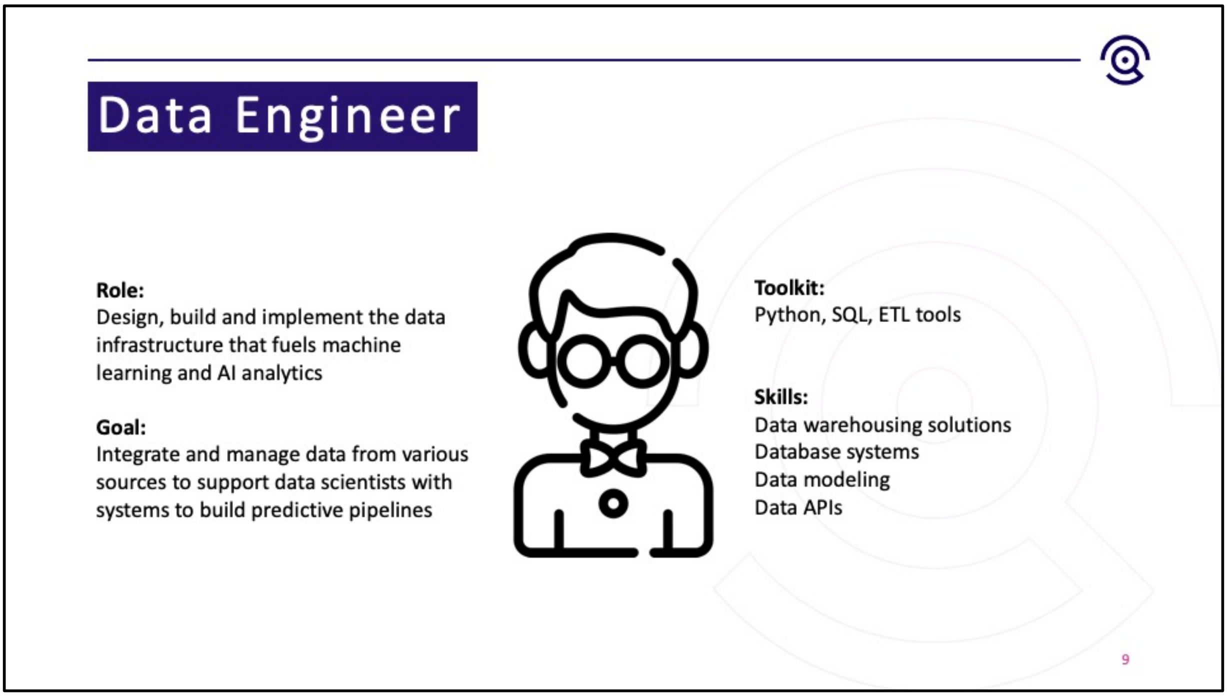
The data science team lead manages the data science team and plans the projects. It is this person's responsibility to make sure that the business goals are aligned with the data vision to create value. The team lead should be proficient in SQL and Python. Important skills are leadership and project management, interpersonal communication, data analysis/visualization, predictive modeling and a business mindset. This is basically an experienced person who is technically very solid and knows how to manage people and a business.



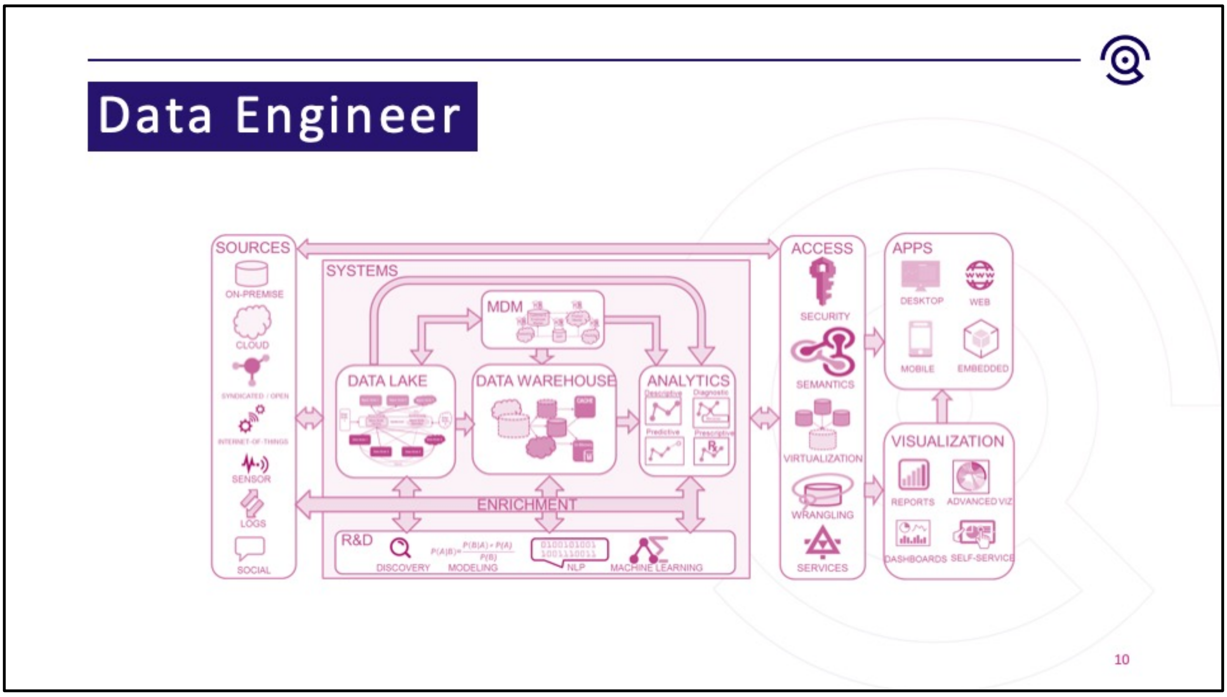
Data scientists typically focus on collecting, analyzing and interpreting the data to propose business decisions. The goal is to construct the means for extracting business-focused insights from data which bring value. The data scientist should be proficient in SQL and Python with several of it’s packages such as Scikit-Learn, Pandas, Numpy and Tensorflow. Important skills are predictive modelling, data visualisation and storytelling, mathematics and statistics and ML/DL algorithms. Data scientist know their way around data and modeling and use this to create actionable decisions that generate business value for a company.



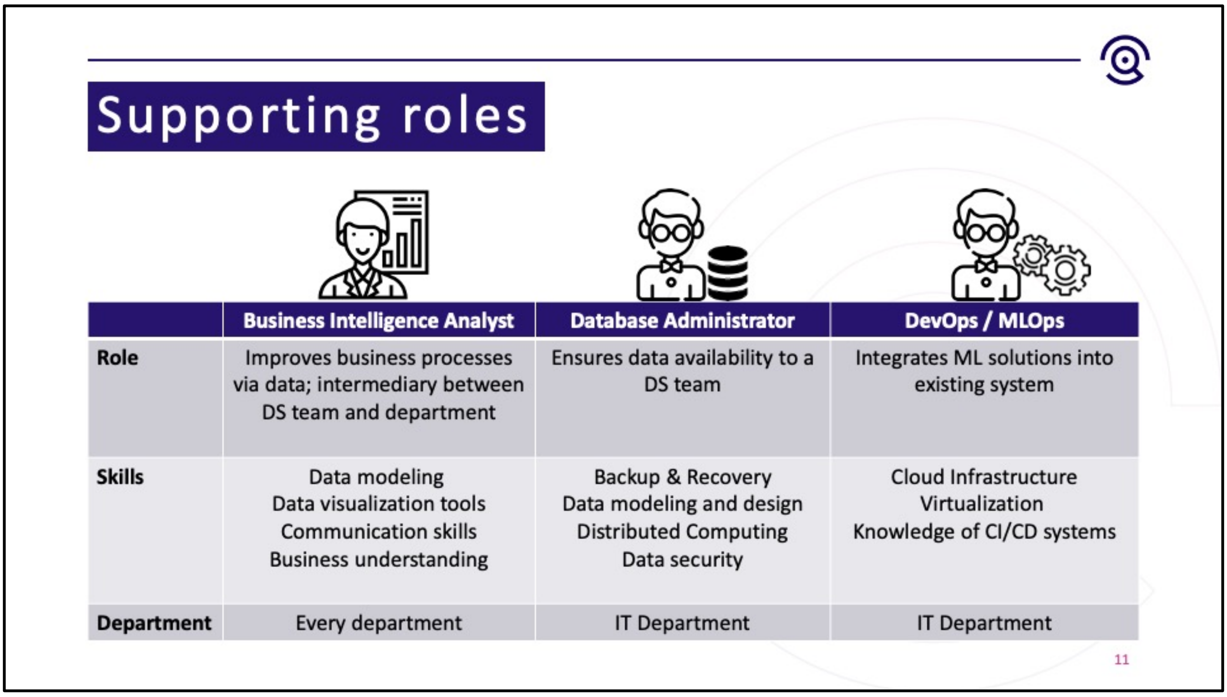
Data scientists are often described as the unicorns who are skilled in computer science, maths/stats and business knowledge. It is therefore someone with a solid technical understanding of IT systems, a good mathematical foundation to tackle problems and business insights on how to create value from all the available data.



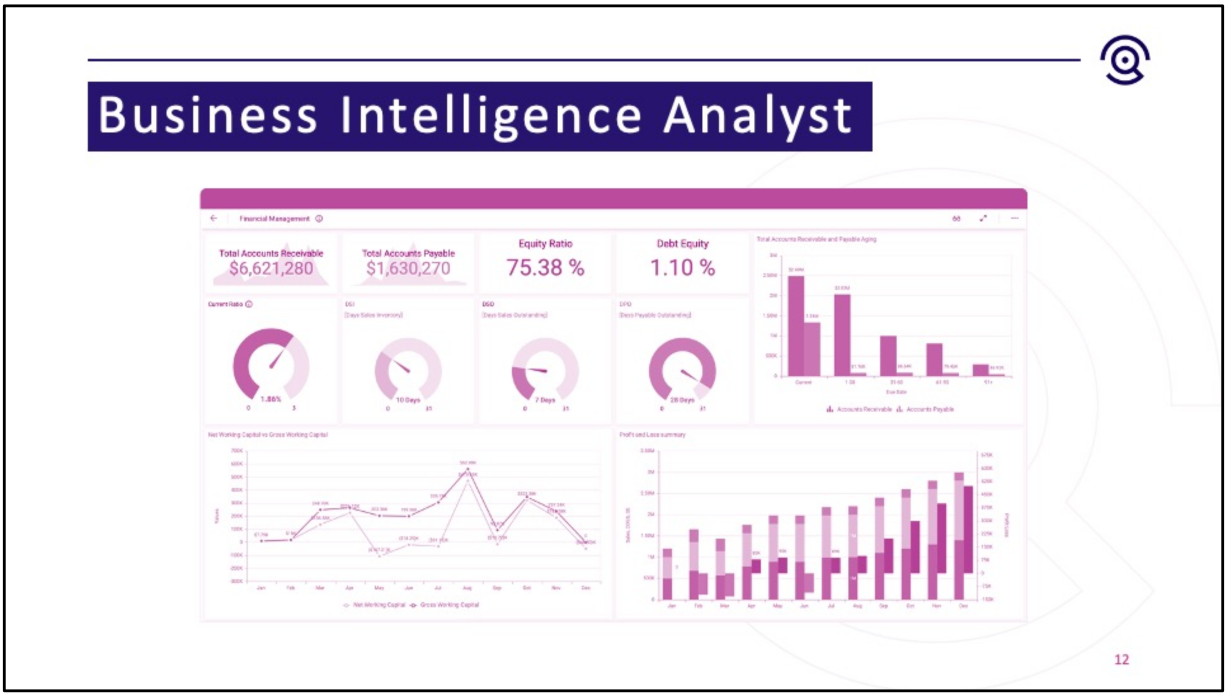
A data engineer designs, builds and implements the data infrastructure that fuels machine learning and AI analytics. The goal is to integrate and manage data from various sources to support data scientists with systems to build predictive pipelines. The data engineer should be proficient in Python, SQL and ETL (Extract, Transform and Load) tools. Important skills are data warehousing solutions, database systems, data modelling and APIs. The data engineer builds all the underlying infrastructure that makes data science possible and allows data scientists to work smoothly.



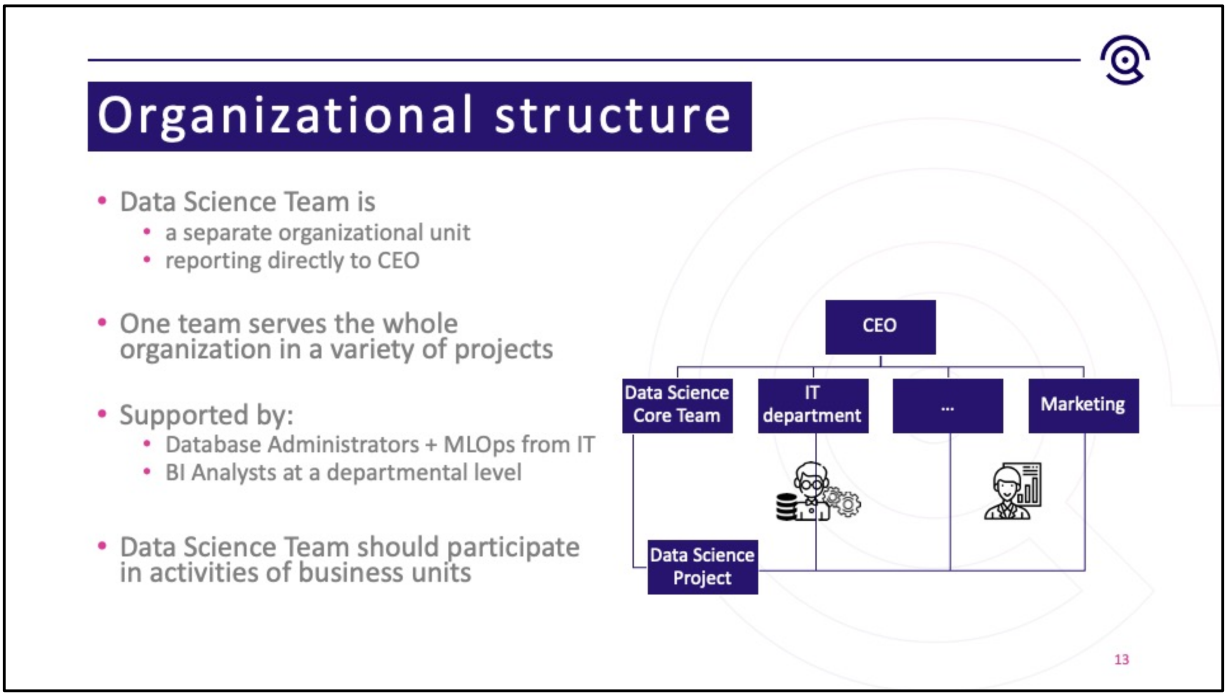
The job of a data engineer is very diverse and not to be underestimated. Data is extracted from various sources such as on-premise storage, the cloud or via sensors. This data is often transformed before it is loaded in the company’s data systems such as the data lake or warehouse. Data engineers also make sure that the data can be accessed in a smooth way via apps and some visualizations. Most of these things happen behind the screens but are extremely important.



The business intelligence (BI) analyst uses data to improve business processes and acts as intermediary between the data science team and the business department. A typical BI analyst is proficient in data modeling and visualization with strong communication skills and a solid business understanding. The database administrator ensures the availability of data to the data science team. Necessary skills are backup and recovery, data modeling and design, distributed computing and data security. A DevOps/MLOps engineer integrates machine learning solutions into the existing production system. This requires knowledge of the cloud infrastructure, virtualization tools and CI/CD (continuous integration/delivery/deployment) systems. Both the database administrator and MLOps engineer are usually located in the IT department, whereas the BI analyst can be found in any business department.

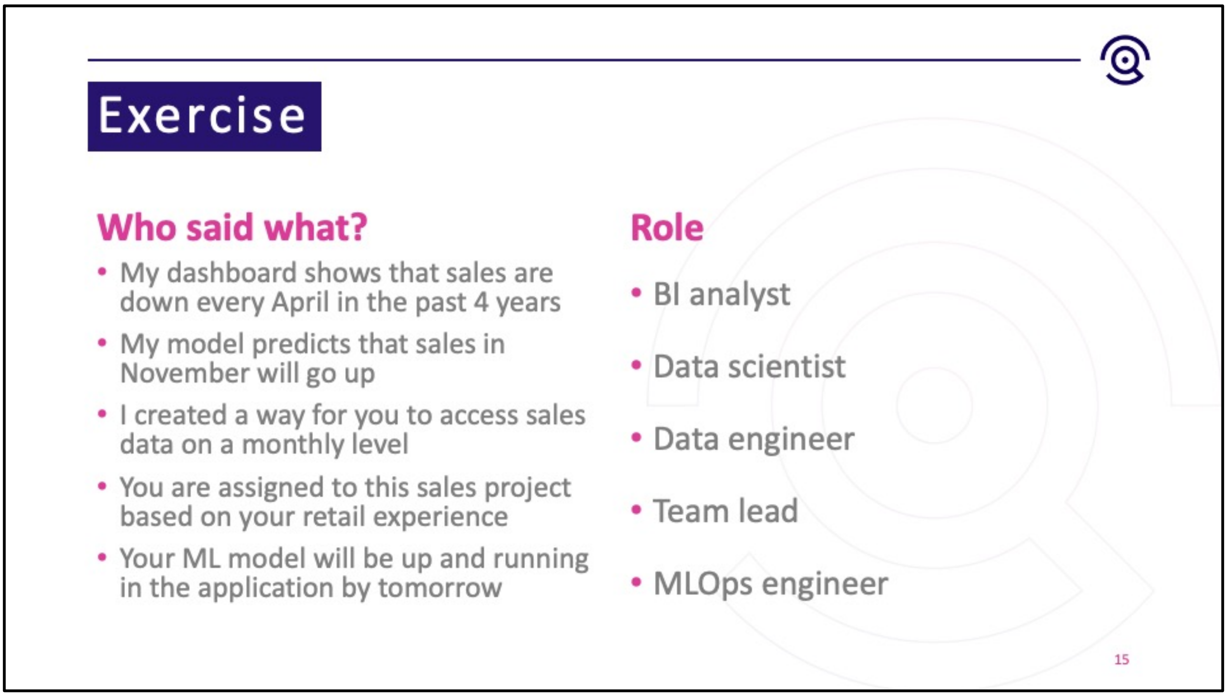


BI analysts typically develop dashboards to find patterns and trends in a company's historical data to derive business insights. Data scientist take this one step further by searching for predictors and significance behind those patterns. These two roles go hand in hand and are complementary, where the BI analyst's goal is descriptive and the data scientist's goal is more predictive. Both use their insights to propose business decisions that generate value.

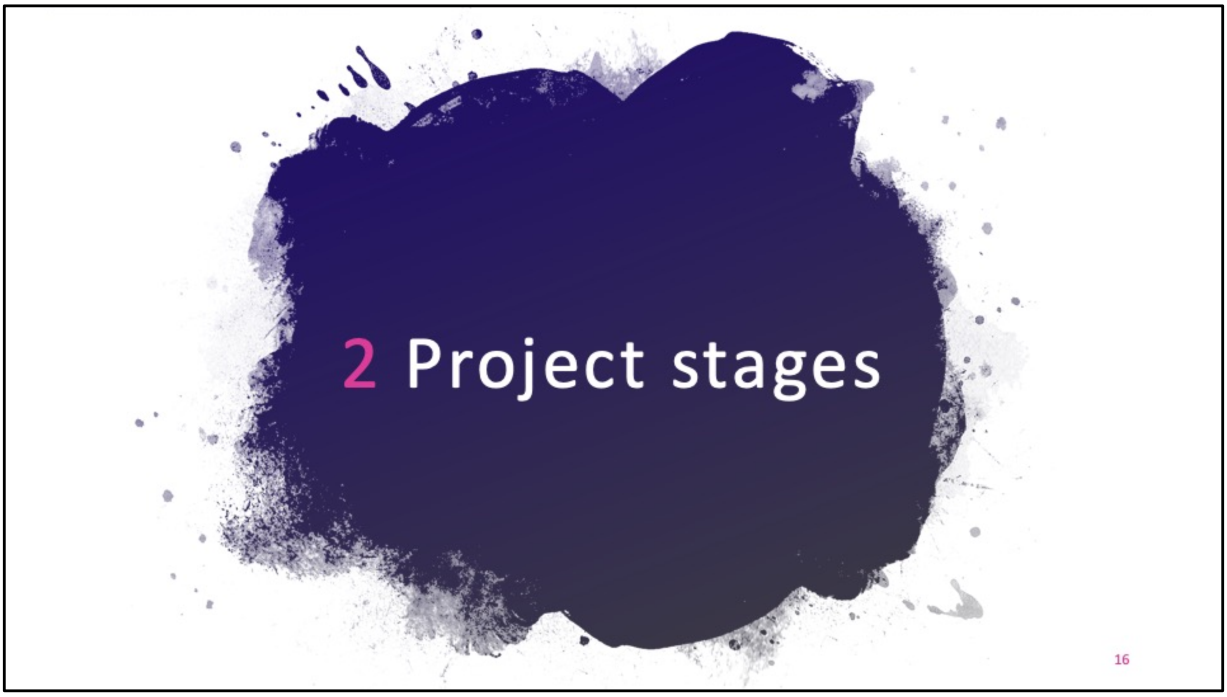


We show a simple organizational structure to situate the business units involved in a data science project. The data science team is best set up as a separate organizational unit which reports directly to the CEO (or CTO for example). This structure makes it easier for one team to serve the whole organization in a variety of projects. This is more efficient than having a distributed group of data scientists at separate business units. The data science team is supported by database administrators and MLOps engineers from the IT department and BI analysts from other departments. It is important for the data science team to participate in activities of the business units to get a better grasp of the typical problems that need to be solved and build good relations with these units.

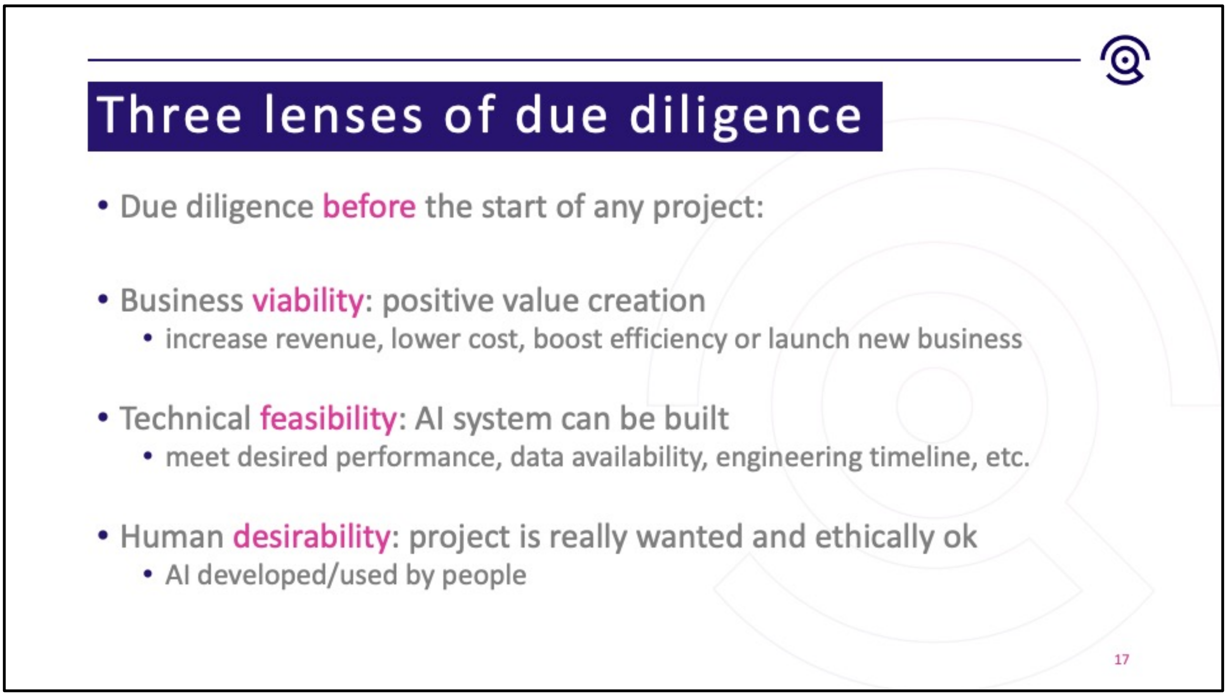
Let’s do a very silly exercise on “Who said what”. Which of the roles we just discussed would say the statements on the left? Take your time for this exercise by pausing the video. Once you resume playing, we will check the solutions together.



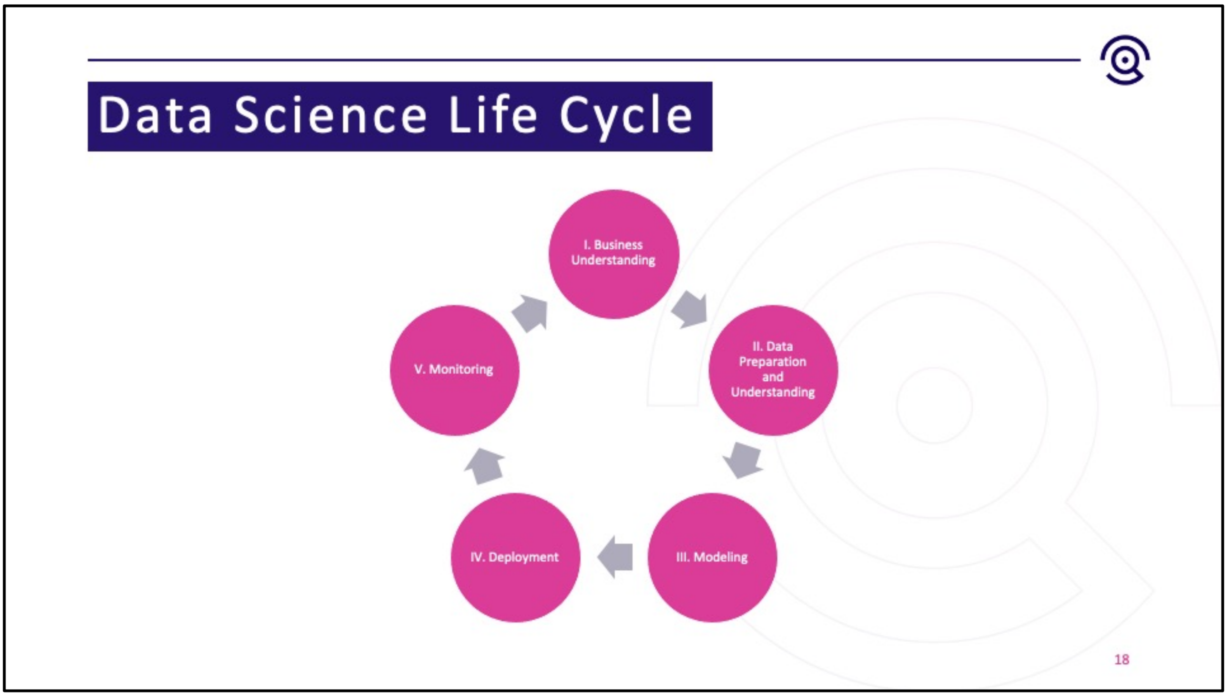
"My dashboard shows that sales are down every April in the past 4 years" is something a BI analyst would say. "My model predicts that sales in November will go up" is data scientist talk. A data engineer would say "I created a way for you to access sales data on a monthly level". The team lead would say "You are assigned to this sales project based on your retail experience" and finally, "Your ML model will be up and running in the application by tomorrow" is something an MLOps engineer would say.



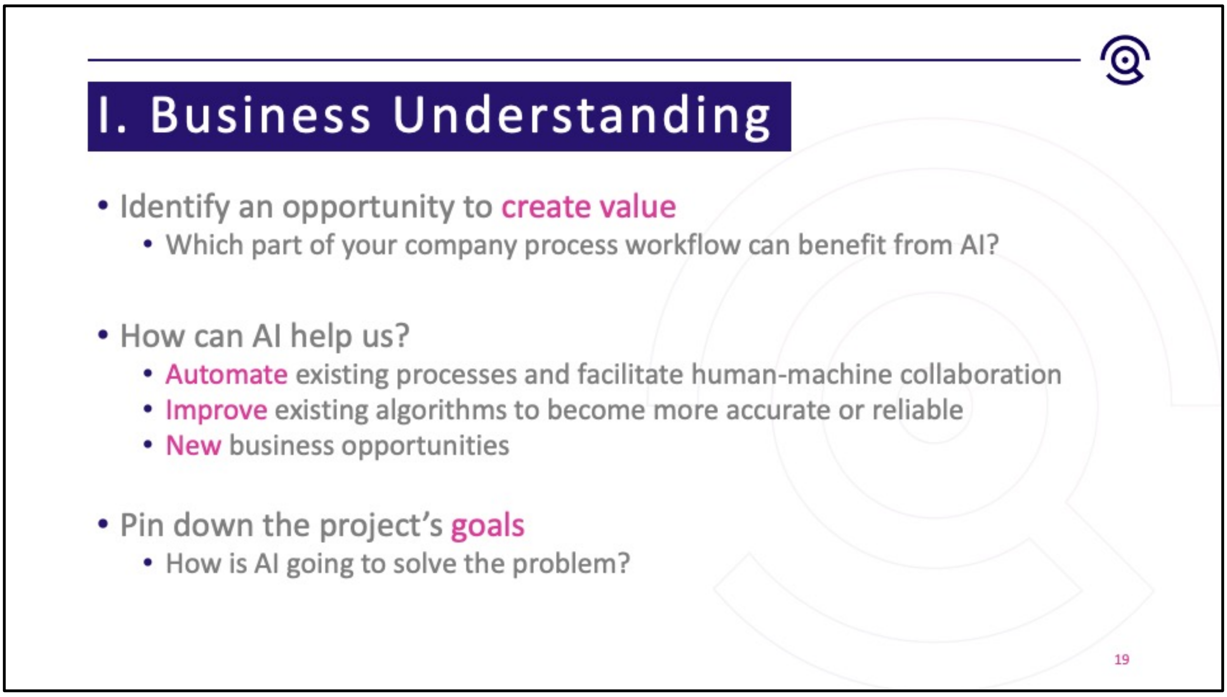
Let’s have a closer look at the different AI project stages.



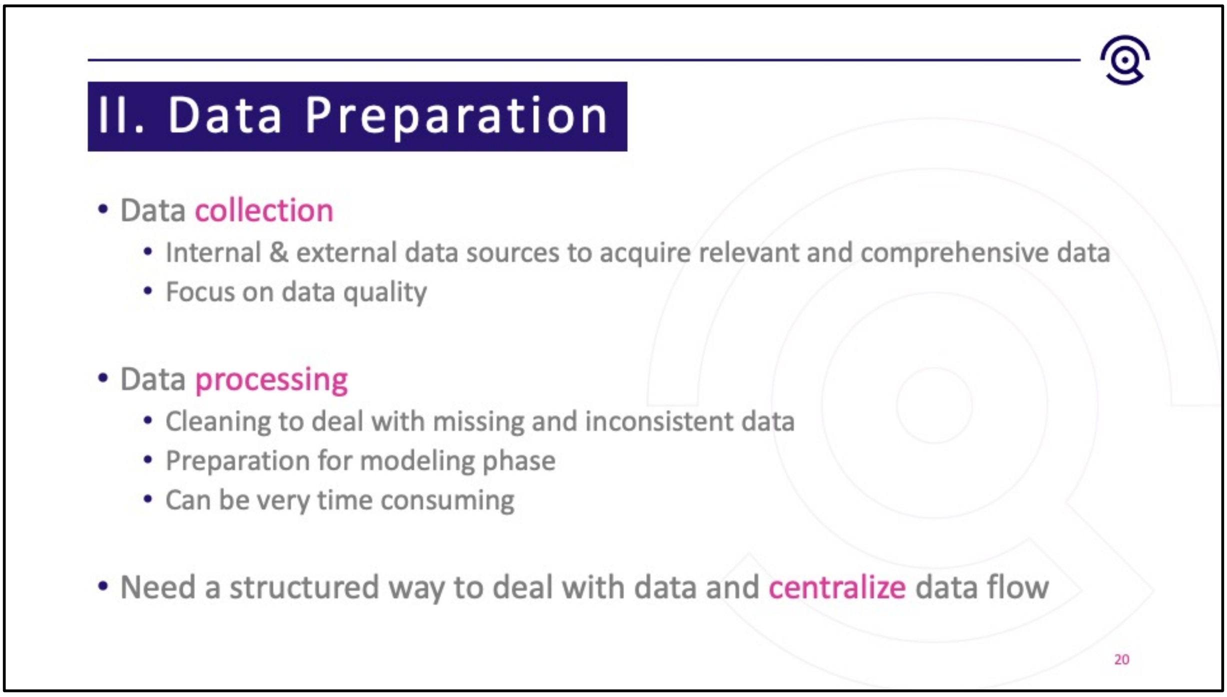
Before starting any project, it is important to assess the three lenses of due diligence. The first is business viability by checking how we are planning to create positive value from this project. This could be to increase revenue, lower costs, boost efficiency or launch new business products or services. The second is technical feasibility by making sure that the AI system can be built. This should consider the desired performance, data availability, engineering timeline and infrastructure needs. The third is human desirability to ensure that the project is really wanted and ethically approved. In the end AI is developed and used by people so it is extremely important to take the human factor into account. If all these three lenses are checked, then the project is ready for liftoff.



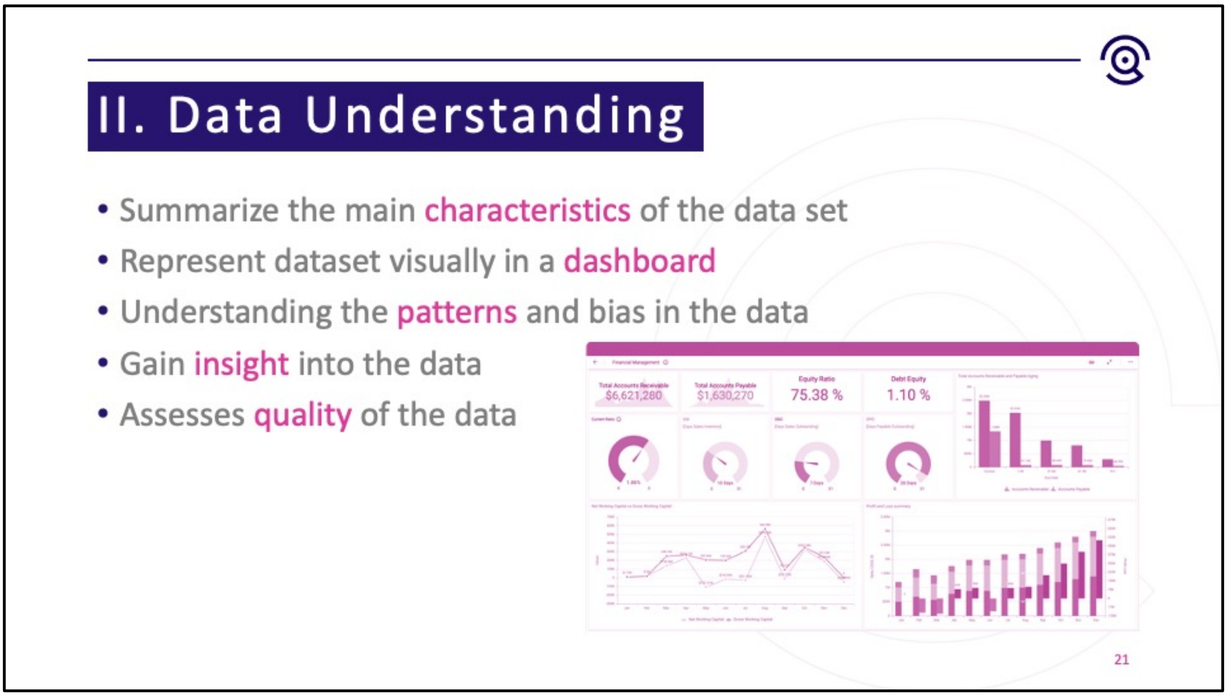
A data science project flows through a life cycle of five steps. The first step is to gain business understanding in order to define clear goals for value creation with AI. The second step is data preparation and understanding to make sure that we have the proper data to solve our business problem. The third step is modeling where we develop AI tools that address the business goals based on the available data. The fourth step is the deployment phase, where the AI solution is put in the production environment of a company and can actually start to generate business value. The fifth step is monitoring of the deployed AI solution to track that it keeps on working as expected and possibly also update/improve the model over time. These five steps are drawn as a unidirectional cycle, but feedback loops exist. After modeling it is for example possible that we decide to collect new data to improve the model if it does not achieve the desired level of performance. After deployment or monitoring we might decide to go back to the data or modeling phase to tweak the AI solution. Even though it is not necessarily a straight path from step one to five, all these steps are extremely important in any data science project. We will therefore discuss each of them in more detail.



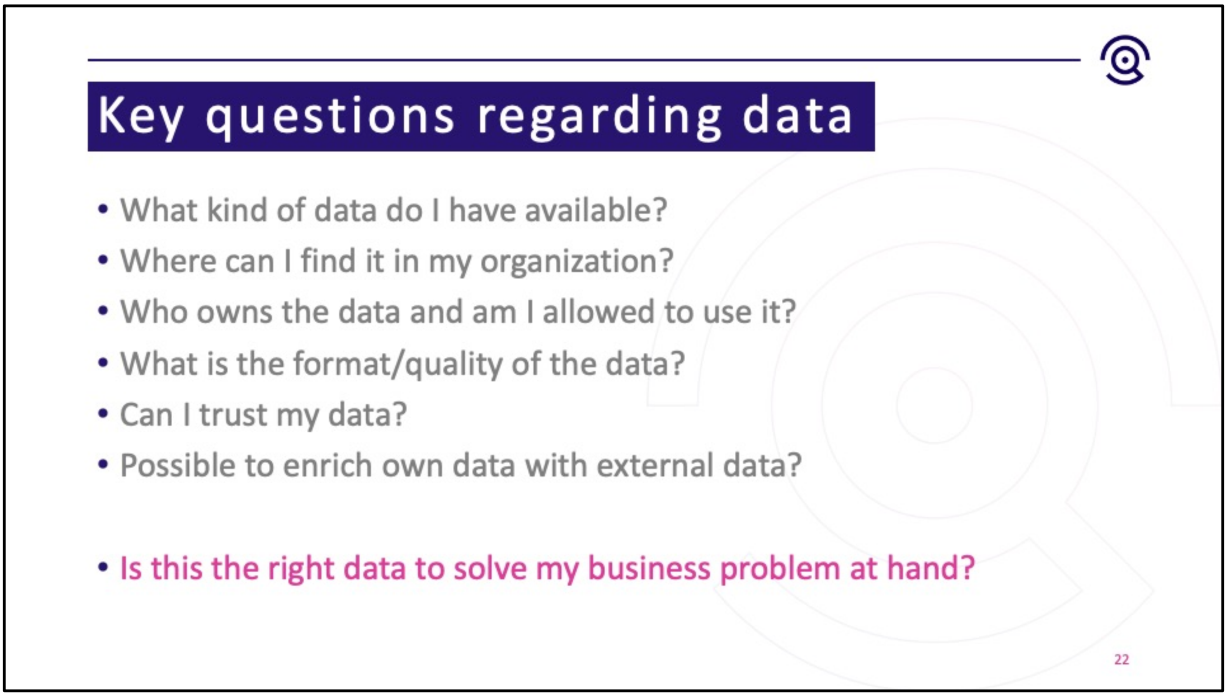
A proper business understanding is needed to identify opportunities for AI to create value in a company. Try to discover parts of the company process workflow that can benefit from AI. In module one we discussed some questions to ask when trying to identify valuable AI use cases. AI can help in several ways to improve processes. One can automate existing processes with AI and facilitate collaboration between humans and machines. One can improve existing algorithms that are already valuable, but can become more accurate or reliable. One can discover new business opportunities that impact core business processes and activities of a company. In this phase it is really important to pin down what the project’s goal is and how AI is going to solve the problem.



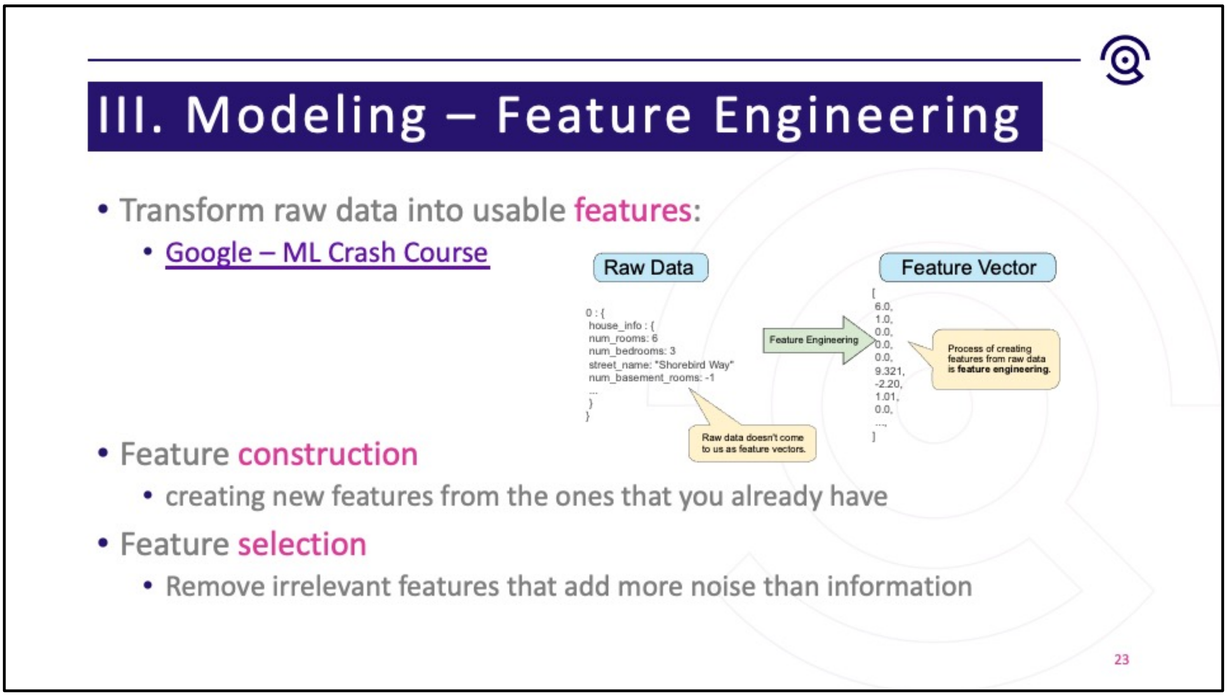
After pinpointing the business goals and expectations, it is time to get our hands dirty with data. The first step is data collection from internal and external data sources to acquire relevant and comprehensive data. Data quality is extremely important and often a problem in companies. Data is the new gold, so it is worth to make sure that this information is properly managed. After data collection follows data processing. Data cleaning is needed to deal with missing and inconsistent data, because data is messy. Never trust your data as it comes, always be very careful and perform a series of checks to make sure that everything is ok. Data processing also involves preparing the data for the modeling phase by putting it in the right format to use. All these steps can be very time consuming but are really important. This asks for a structured way to deal with data and centralize the data flow. This is exactly where the data engineer comes in to ensure a smooth process for the data scientist.



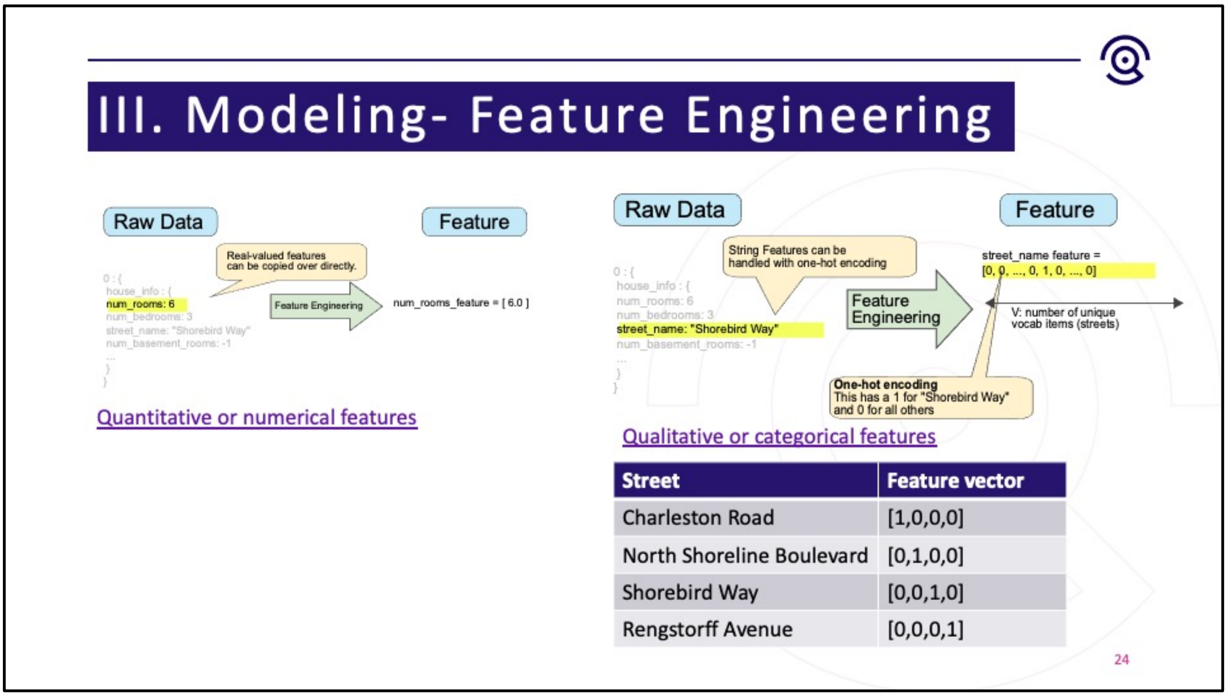
Once the data is prepared for use, it is important to gain an understanding of the data. It is always good to summarize the main characteristics of the data set. For example, which information is available and how it is distributed in the data. Representing the data visually in a dashboard can be very informative. This helps in understanding the patterns (and possible bias) in the data, to gain useful insights and to assess the quality of the data. In this step the data scientist and BI analyst can work together to get optimal results.



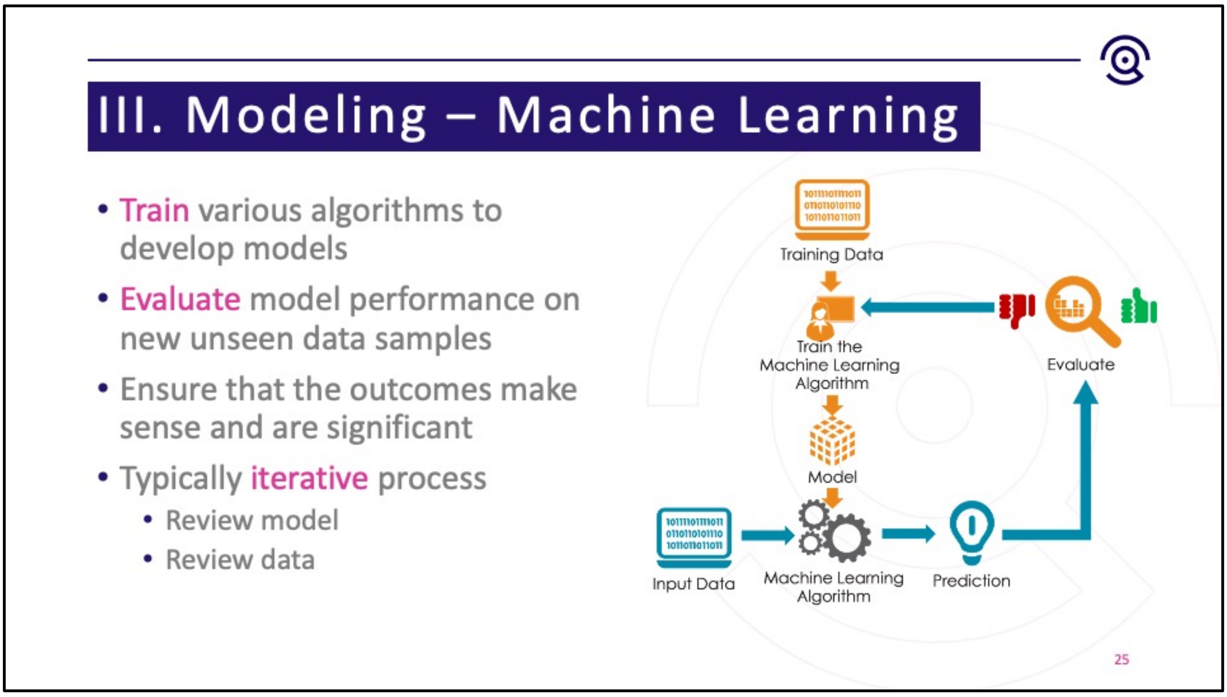
There are some key questions to ask about your data. What kind of data do I have available? Where can I find it in my organization? Who owns the data and am I allowed to use it? What is the format/quality of the data? Can I trust my data? Is it possible to enrich own data with external data? And possibly the most import question of all: Is this the right data to solve my business problem at hand? After step 2 you should be convinced that the answer to this last question is a resounding yes! If not, then it might be good to revisit one of the other questions and make sure that you collect the correct data to solve your problem.



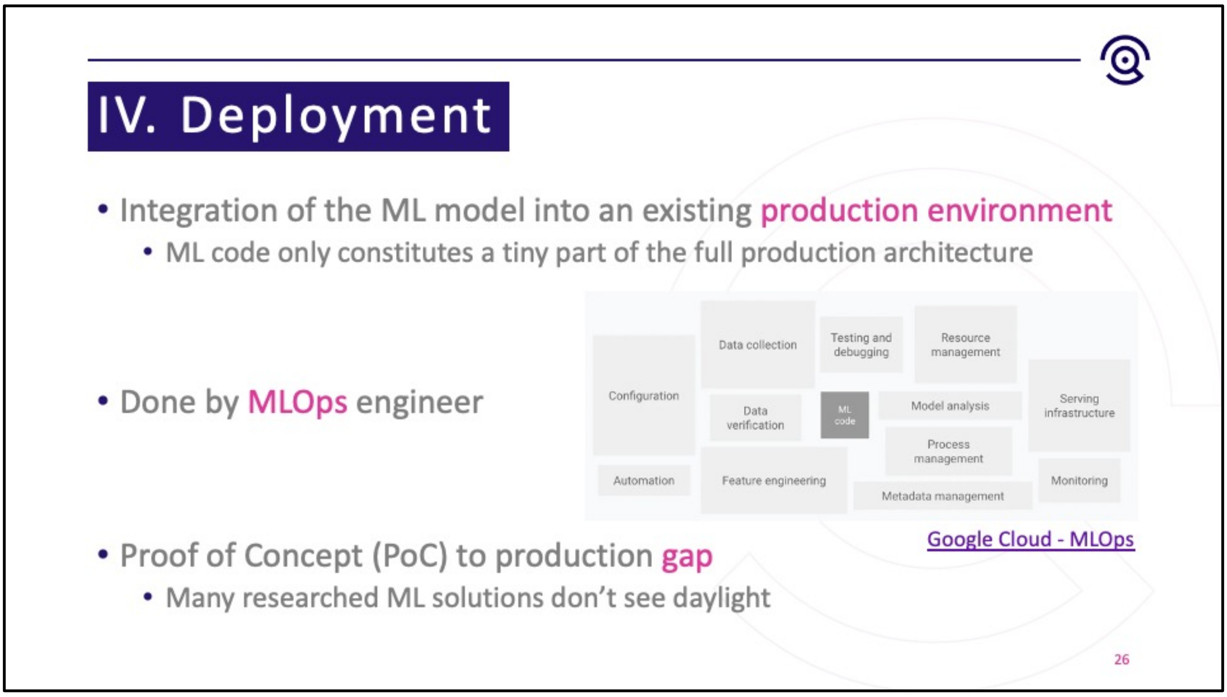
Once the data is prepared and understood, it is time for the fun to begin and start modeling. Feature engineering is the process to transform raw data into usable features. The left side of the graph shows raw data from an input source on housing information. The right side of the graph shows a feature vector as a set of numeric values that represent the same information. Most ML models need such numeric vectors to analyze and process the data, but the problem is that raw data does not come in this format. Feature engineering transforms the raw data information in such feature vectors that a ML model can actually use. Feature construction involves creating new features from the ones that you already have. Feature selection deals with removing irrelevant features that add more noise than information. The quality of the features often has a huge impact on the predictive performance of an ML model. Unfortunately this is usually a manual task, so be prepared to spend some time on proper feature engineering.



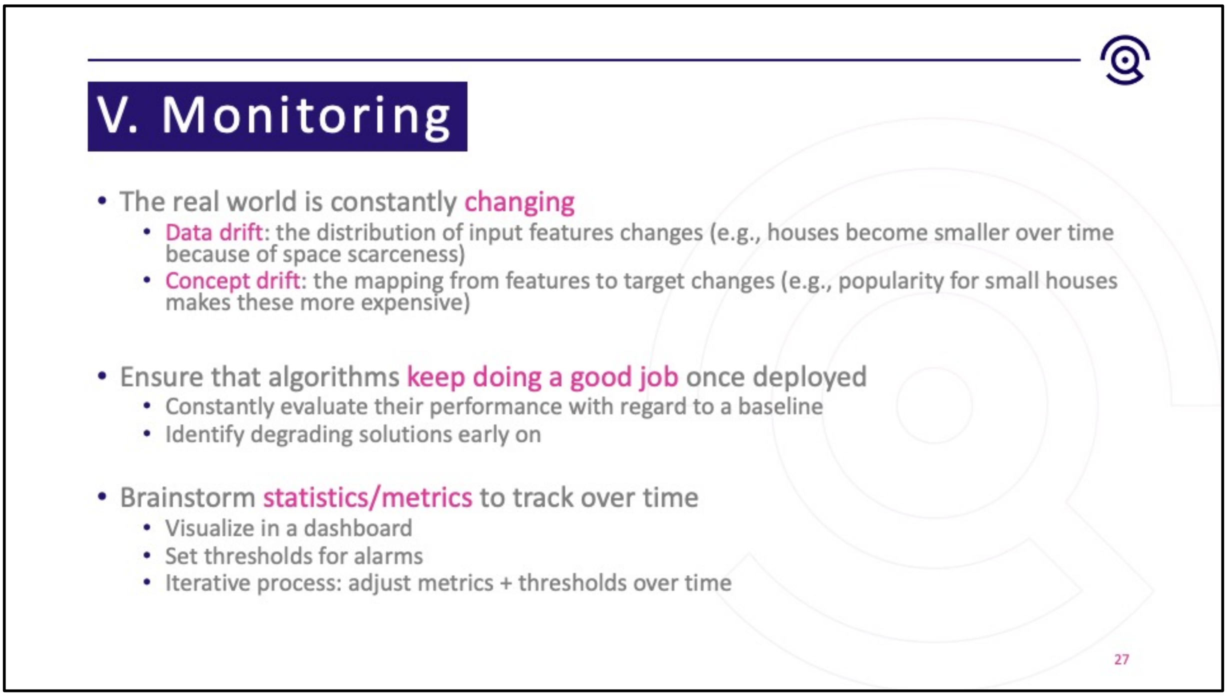
Feature engineering for quantitative (or numeric) values is usually trivial as we can simply copy its value. Engineering qualitative (or categorical) features with a limited number of possibilities is less trivial. Assume there are four possible streets: {'Charleston Road', 'North Shoreline Boulevard', 'Shorebird Way', 'Rengstorff Avenue’}. One could assign an index to each street, namely 1 to 4, in order to get numeric values. Easy right? The problem with this approach is that is assumes an ordering between the streets as 4 > 3 >2 > 1. This is of course not what one wants. A better approach is to use the so-called one-hot encoding approach. This creates a vector of length equal to the size of the number of categories, four in our example. Only one element of that vector takes the value of 1, which element depends on the street’s name, and the other values equal 0. 'Charleston Road’ is then represented as [1,0,0,0] for example, 'North Shoreline Boulevard’ is represented as [0,1,0,0] and so on. This way the ML model can use the categorical feature with a clear distinction in the different values.



Once the features are ready to be used it is time to do the actual modeling with machine learning. Training ML models involves experimenting with various algorithms to develop competing models. Each model is then evaluated on new unseen data samples in order to assess the generalization performance. This allows a data scientist to pick the best model for the specific use case. It is always important to make sure that the outcomes make sense and are significant, both from a technical and a business perspective. The modeling phase is typically an iterative process where we try some things, see if it works or not, and try again. Models are often reviewed and it might also happen that we go back to the data for review after seeing the model's results.

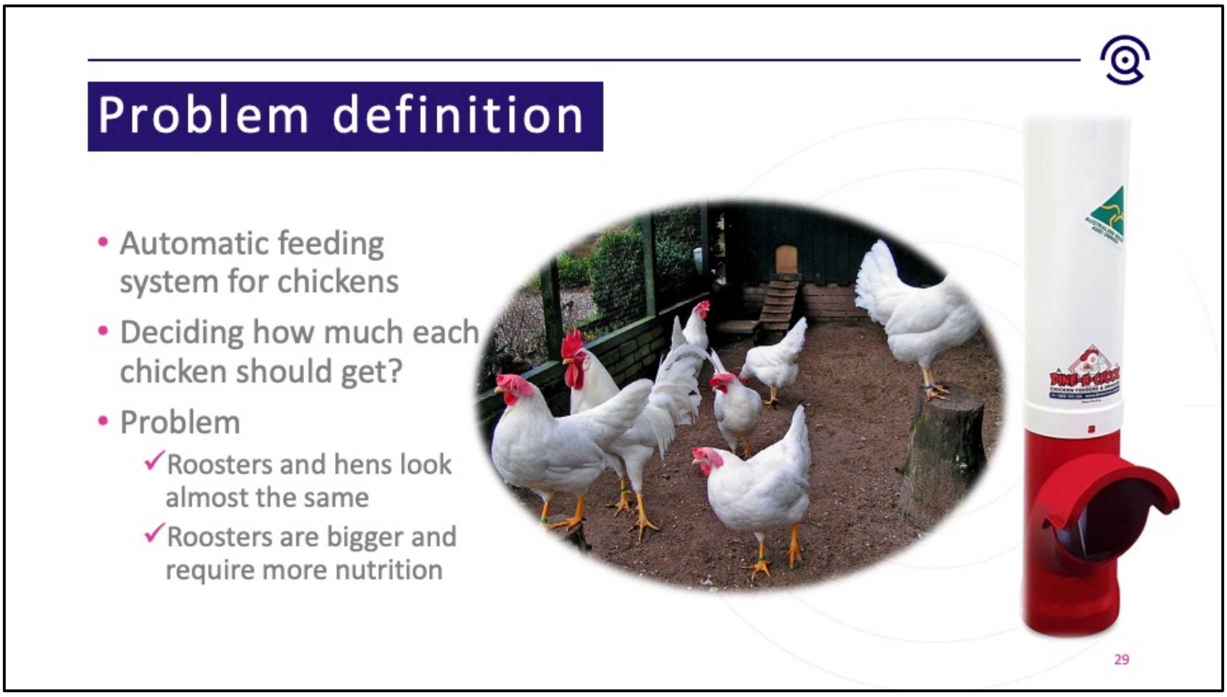


Once the ML model is developed in the data scientist's playground, it is time to deploy it into practice. So far the ML model is mainly a proof of concept (PoC) but it is not generating value yet. Deployment refers to the integration of the ML model into an existing production environment. This is often overlooked, but the ML code only constitutes a tiny part of the full production architecture. Other elements involve testing, serving infrastructure, resource management, automation and so on. An MLOps engineer ensures that the AI solution is deployed successfully in this environment and starts to generate value. The PoC to production gap refers to the fact that many researched ML solutions never see daylight and simply cease to exist, which is a big problem in many companies. To avoid wasted resources it is of course vital that all good solutions make it to production.

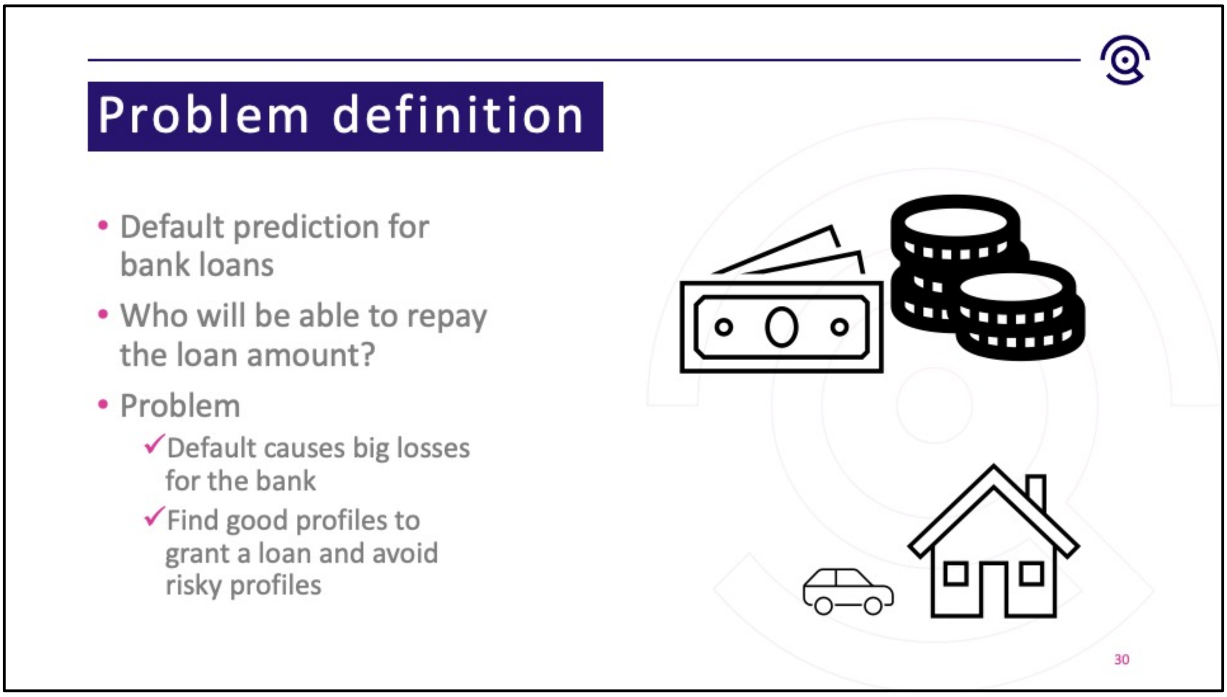


Once a model is deployed in production, it is not over yet. The real world is constantly changing and this can have an impact on your ML model. Data drift refers to the situation where the distribution of input features changes over time, for example when houses become smaller because of space scarceness. Concept drift implies that the mapping from features to target changes, for example when the popularity for small houses makes these more expensive over time. When you leave your model unchanged it will start performing worse and worse. It is therefore key to ensure hat algorithms keep doing a good job once they are deployed. One should constantly evaluate their performance with regard to a baseline in order to identify degrading solutions early on. To achieve this one can brainstorm statistics/metrics to track over time which summarize the performance of the model. These can be nicely visualized in a dashboard with some thresholds for alarms. Also this is an iterative process, where both metrics and thresholds can be adjusted over time. It seems like AI is never done, but with these measures in place you are well on track to keep generating value from the ML solutions in production.

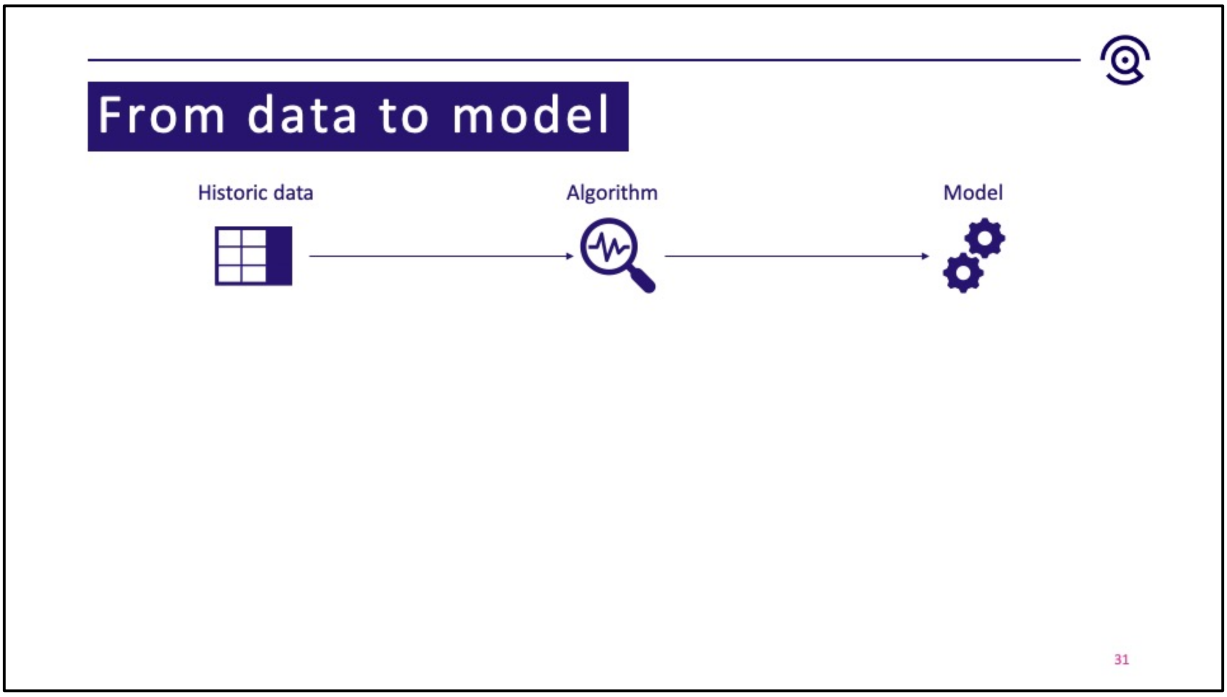
We have seen the five project stages of the data science life cycle. Let's now illustrate this process in more detail with some examples.



Our first problem deals with an automatic feeding system for chickens. This system needs to decide how much food each chicken should get. The problem is that roosters and hens look very similar, but roosters are bigger and therefore require more nutrition. We want to develop an AI solution that classifies roosters and hens based on the visual inspection of images.

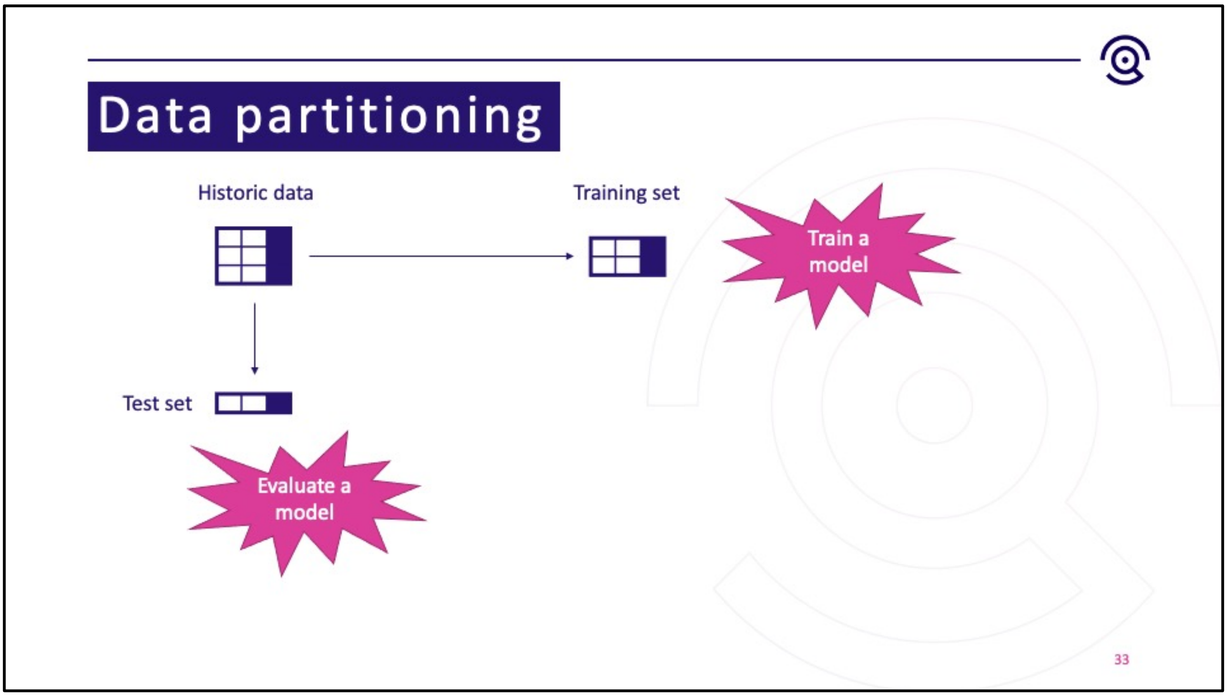


Our second problem deals with default predictions for bank loans. We need to estimate who will be able to repay the loan amount and who won't. The problem is that defaults cause big losses for the bank. We are therefore interested in developing an AI system that finds good customer profiles to grant a loan, while avoiding loans to risky profiles.

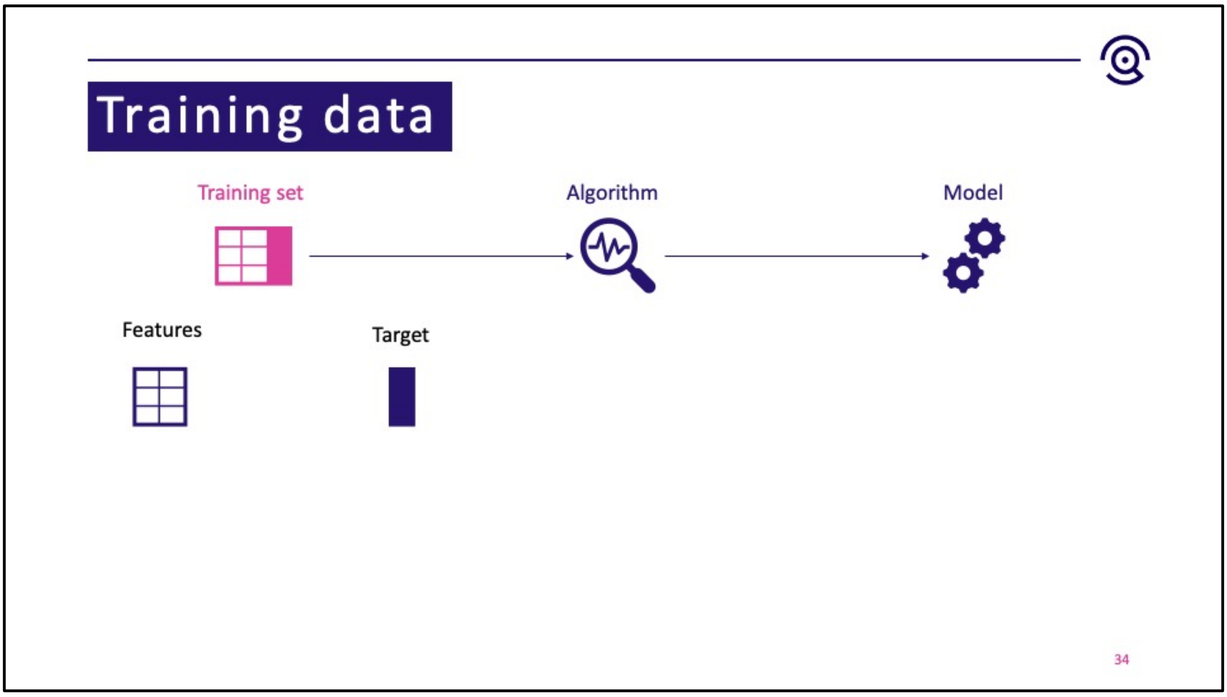


For each of our problems, the goal is to use relevant historical data to extract useful insights which are then captured in a ML model.

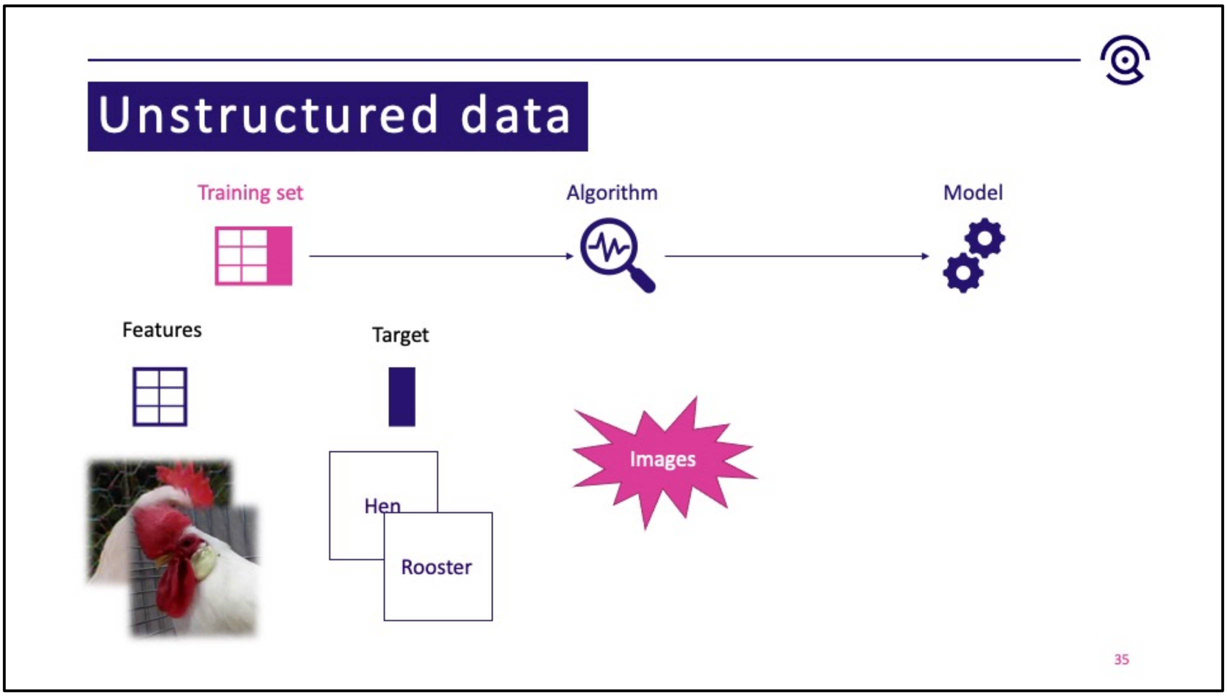
We start from the full dataset.



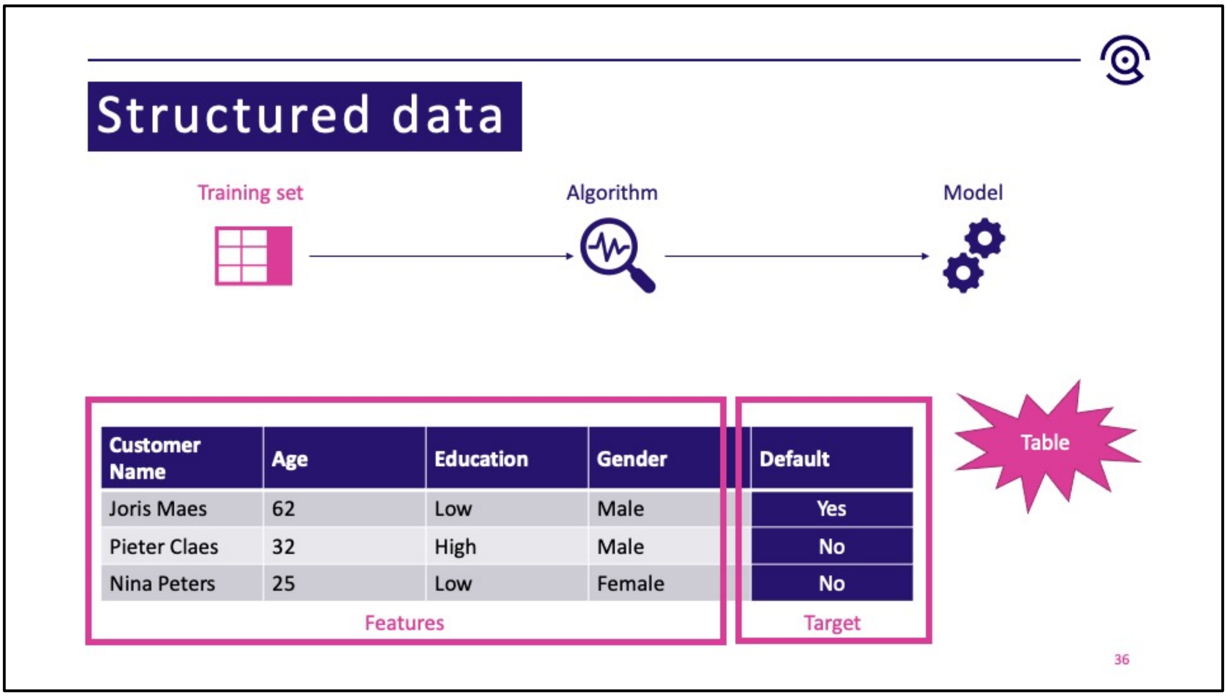
Before starting to model anything, we partition the full dataset in a train set and test set. The train set will be used to develop the ML model, while the test set is used to evaluate the model afterwards.



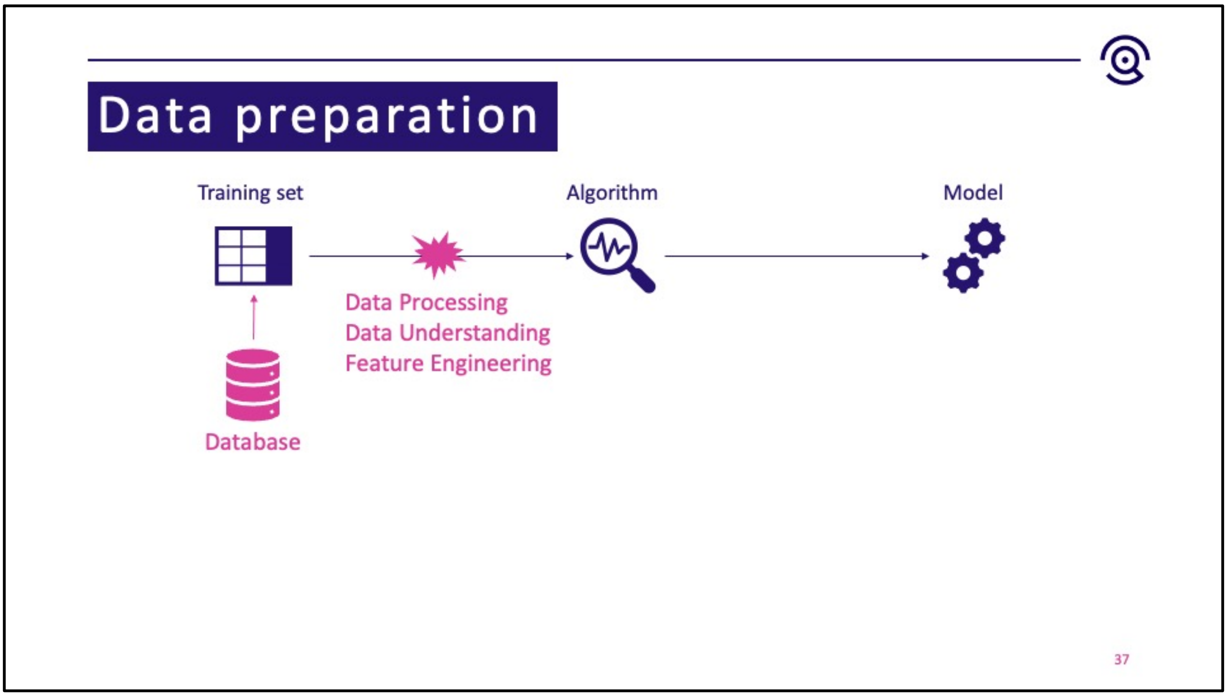
The train set is used to develop the ML model and contains both features and a target. This indicates that we are working with supervised problems. However, are they classification or regression problems?



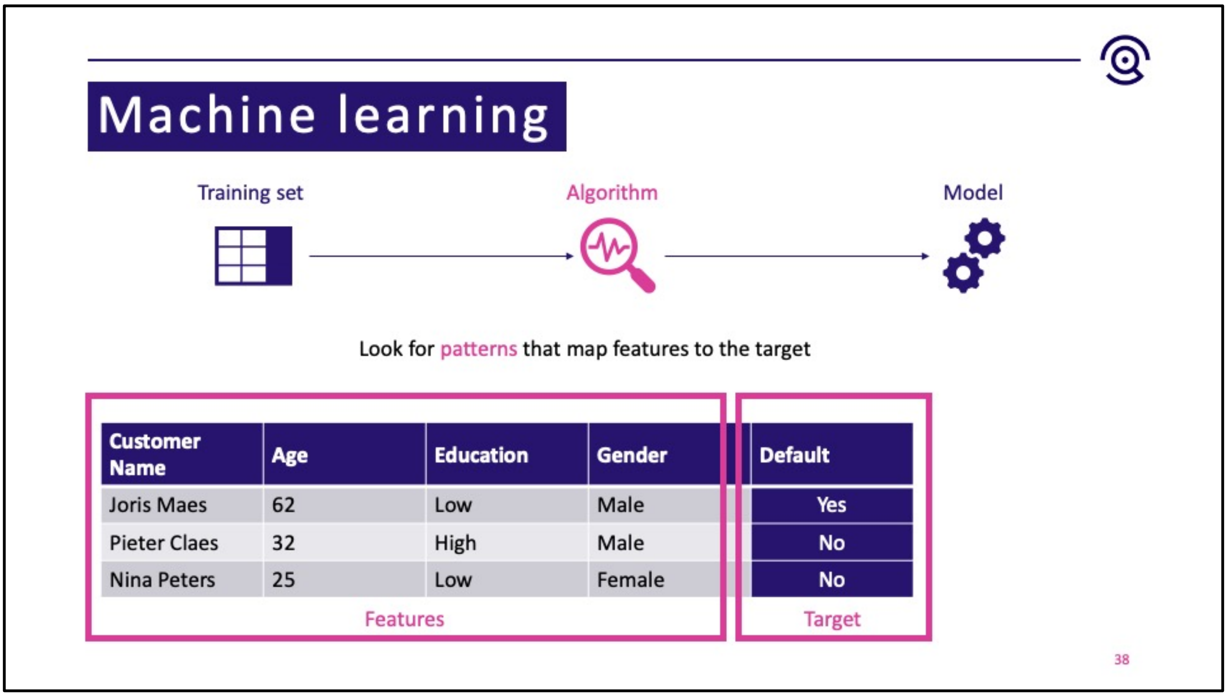
For the chicken example our features are images of the chickens while the target is an indicator for either “hen” or “rooster”. This is therefore a classification problem. Such image data is a typical example of unstructured data which can not be simply captured in a data table with rows and columns.



For the default example our features are customer characteristics while the target is an indicator for either “default” or “no default”. This is therefore also a classification problem. Such tabular data is a typical example of structured data which can be easily captured in a data table with rows and columns. The rows represent different observations/customers and the columns capture the default target and features such as the age, education level and gender.



The whole data preparation process typically involves to collect data from a database, followed by data processing, understanding and feature engineering. Notice that we only use the training data here to avoid information leakage of the test set in our modeling process.



In the modeling phase we look for patterns that map features to the target, so we want to explain the default indicator based on the available features.

Model predictions

Customer Name	Age	Education	Gender	Default
Joris Maes	62	Low	Male	Yes
Pieter Claes	32	High	Male	No
Nina Peters	25	Low	Female	No
Erica Maes	27	Low	Female	???

39

Once these relations are captured in a model we can make predictions for new unseen data examples.

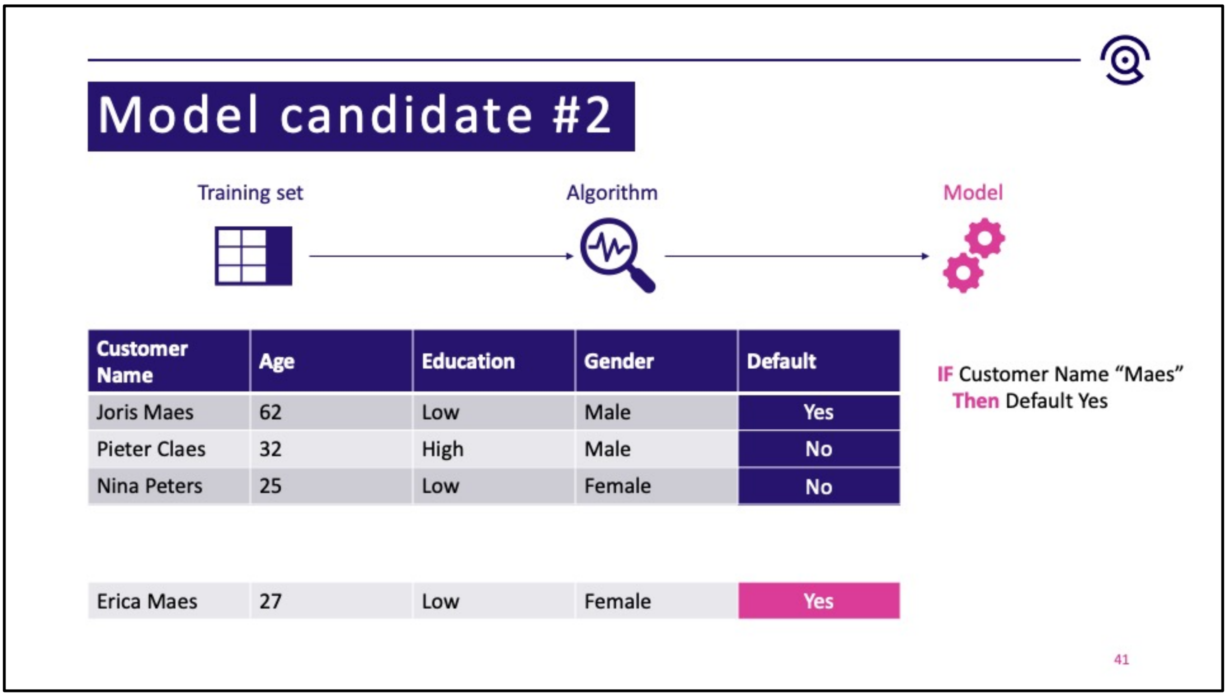
Model candidate #1

Customer Name	Age	Education	Gender	Default
Joris Maes	62	Low	Male	Yes
Pieter Claes	32	High	Male	No
Nina Peters	25	Low	Female	No
Erica Maes	27	Low	Female	No

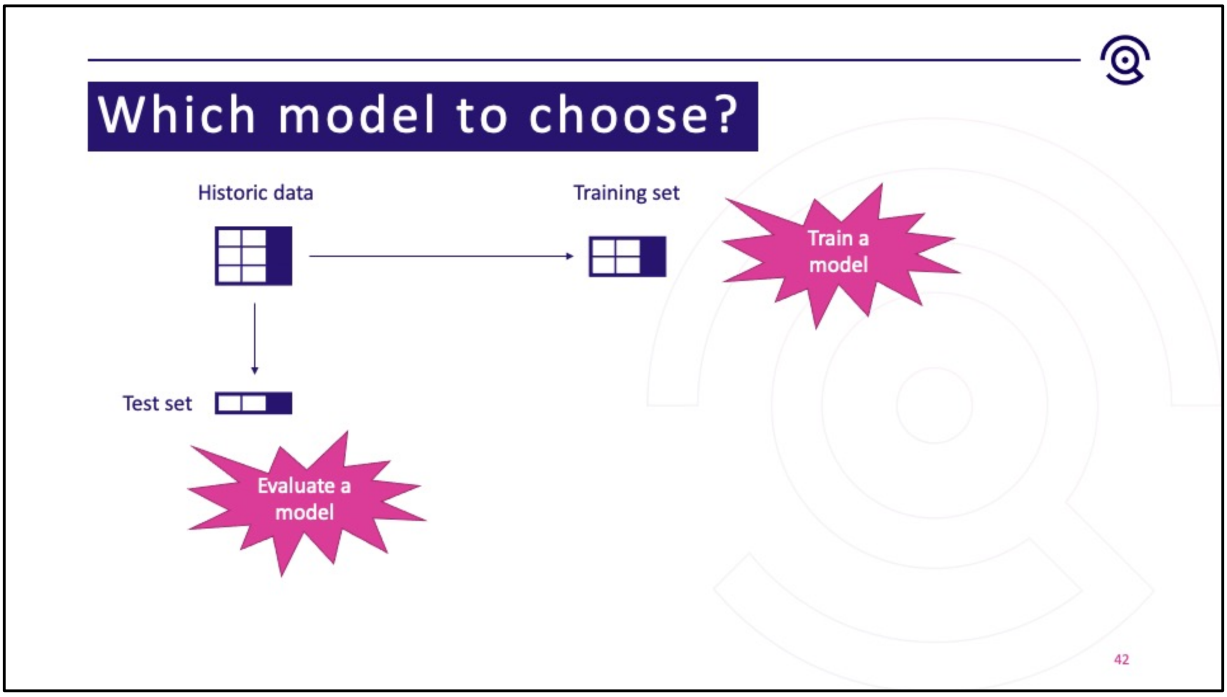
IF Customer Young **&** Female
Then Default No

40

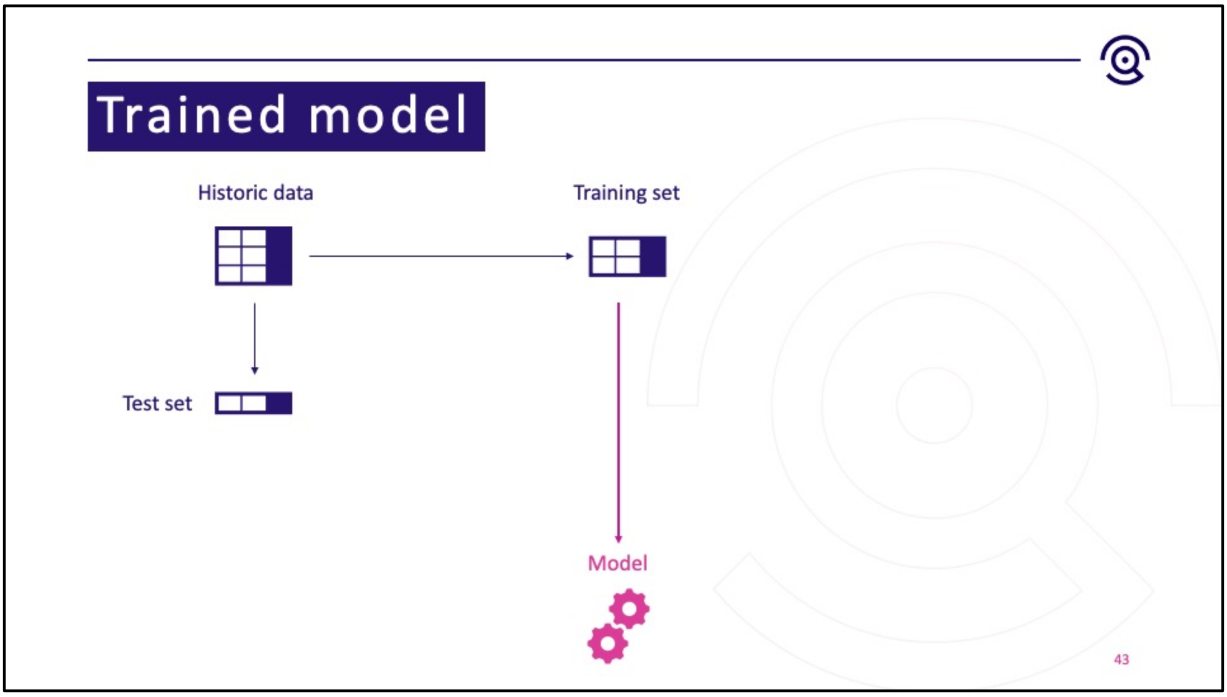
The first model candidate is very simple and predicts “no default” for all young and female customers. The new data example therefore seems to be a good profile to grant a loan.



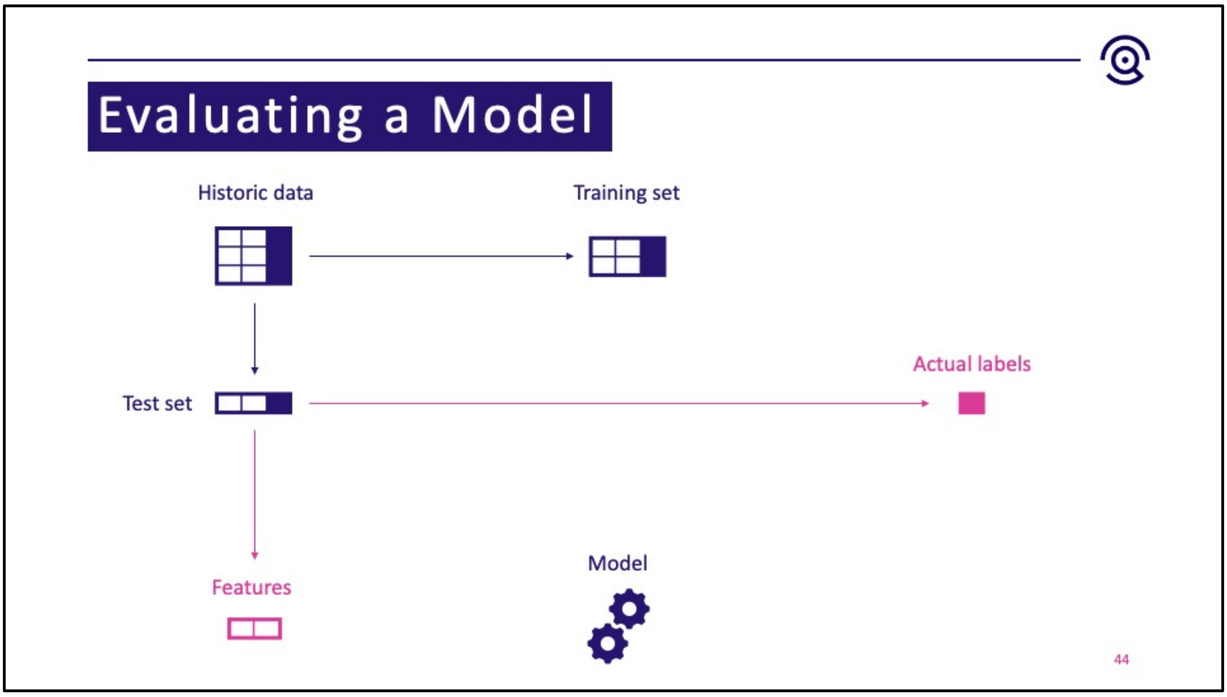
The second model candidate is equally simple and predicts "default" for all customers with the name "Maes". The new data example therefore seems to be a risky profile and will not receive a loan.



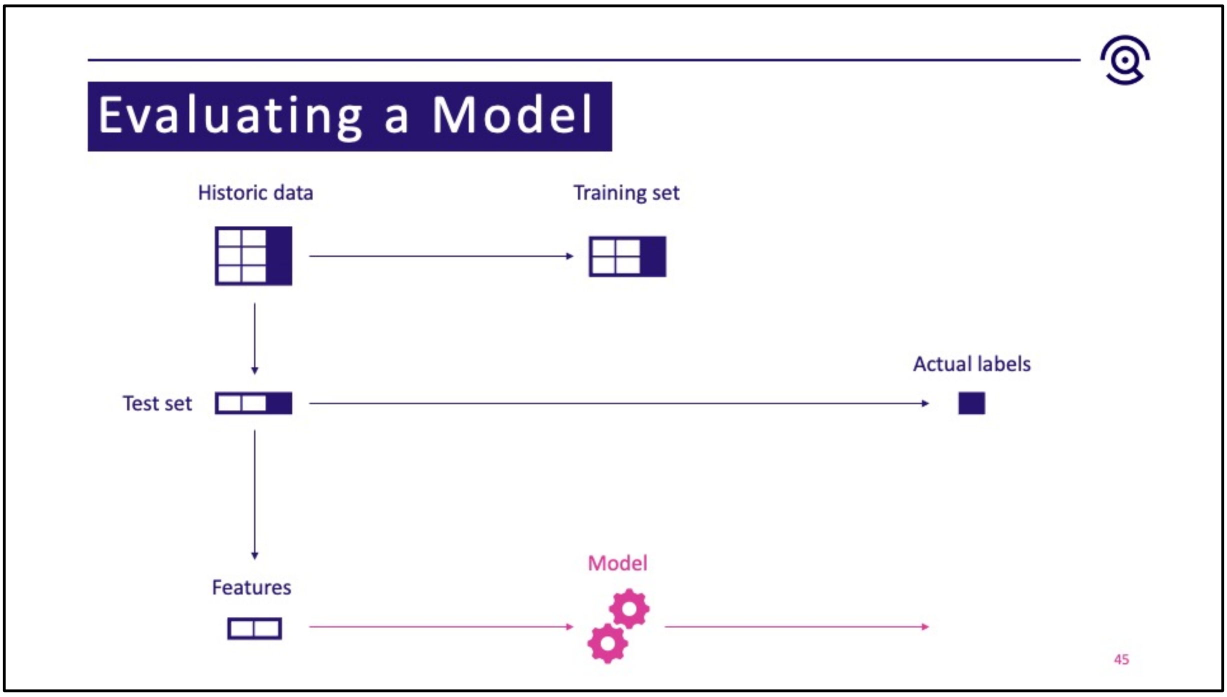
We now have two different models, which one should we choose? This is where the test data comes into play



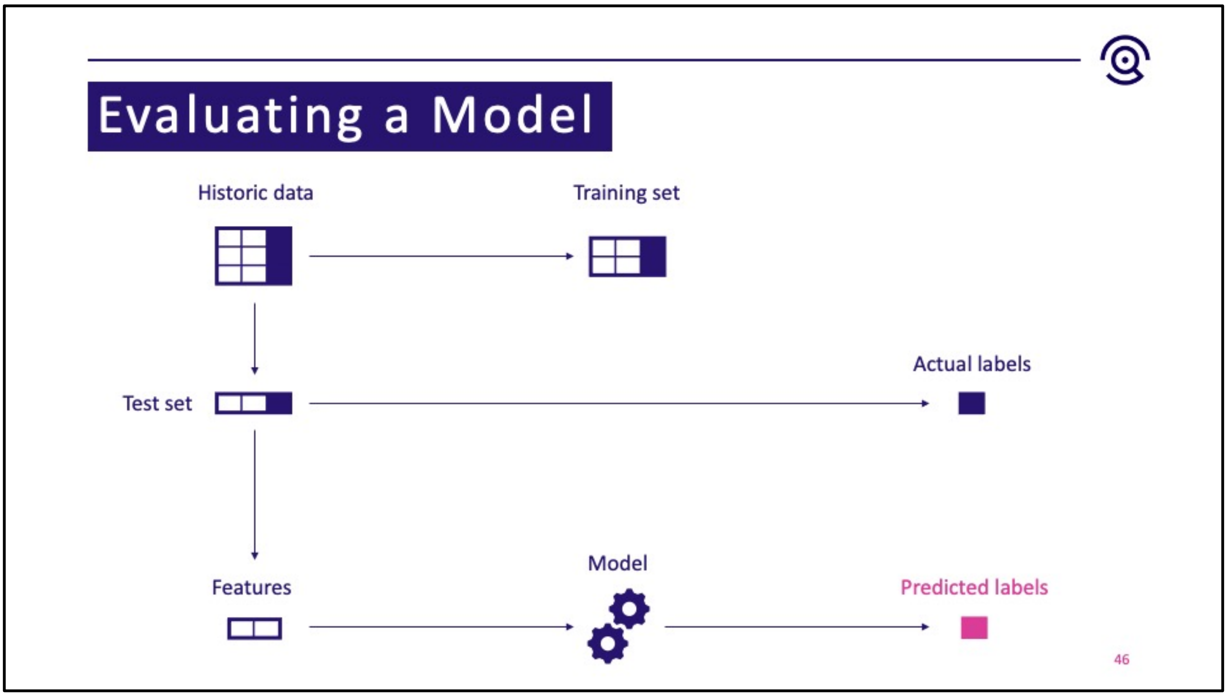
The models are developed on the train data, without looking at the test data.



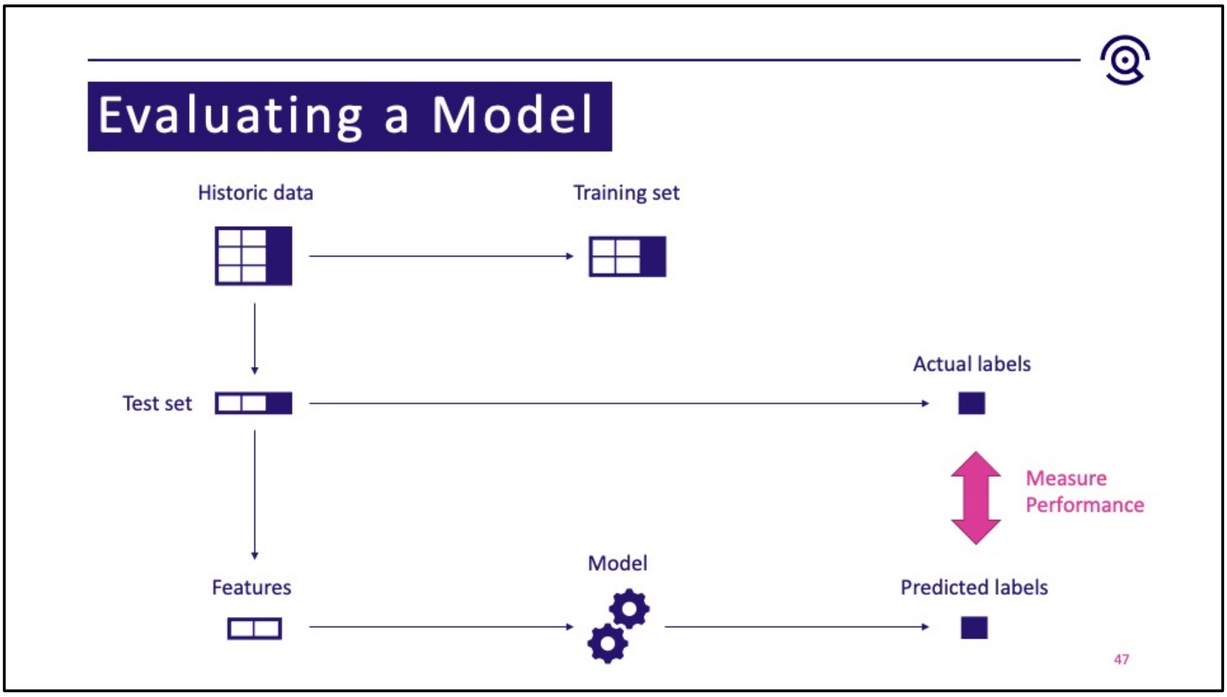
We now take the test data and split the features and targets.



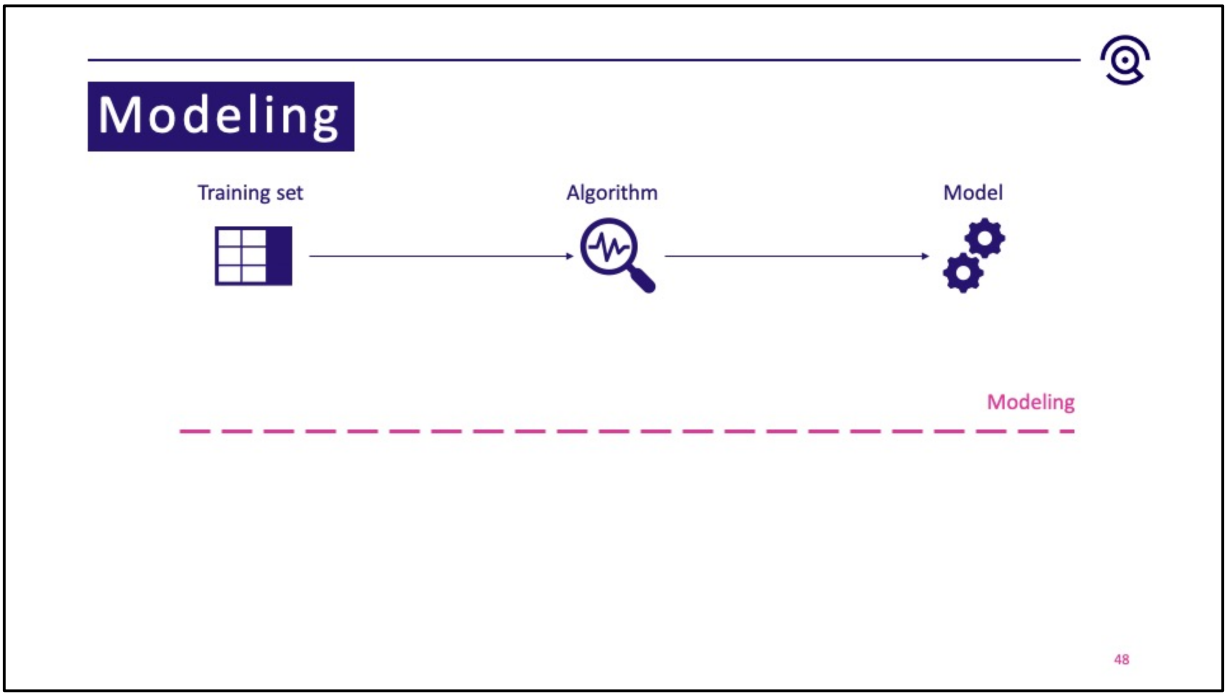
The features are put through the model…



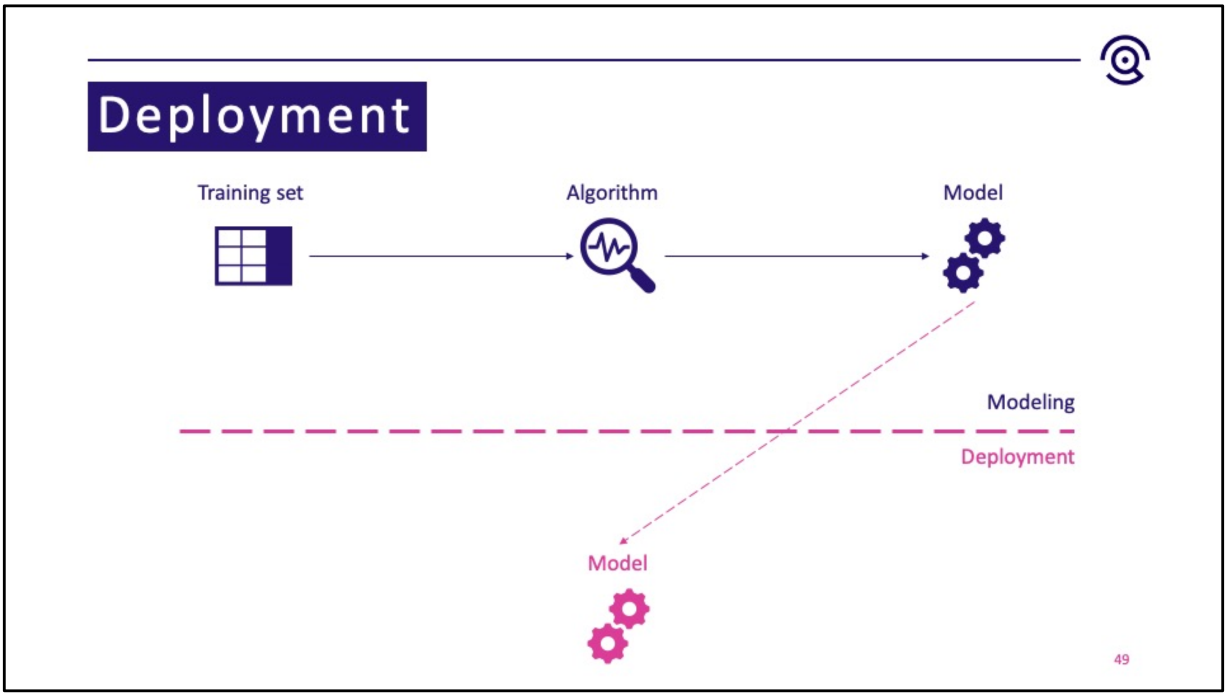
… to generate predicted labels.



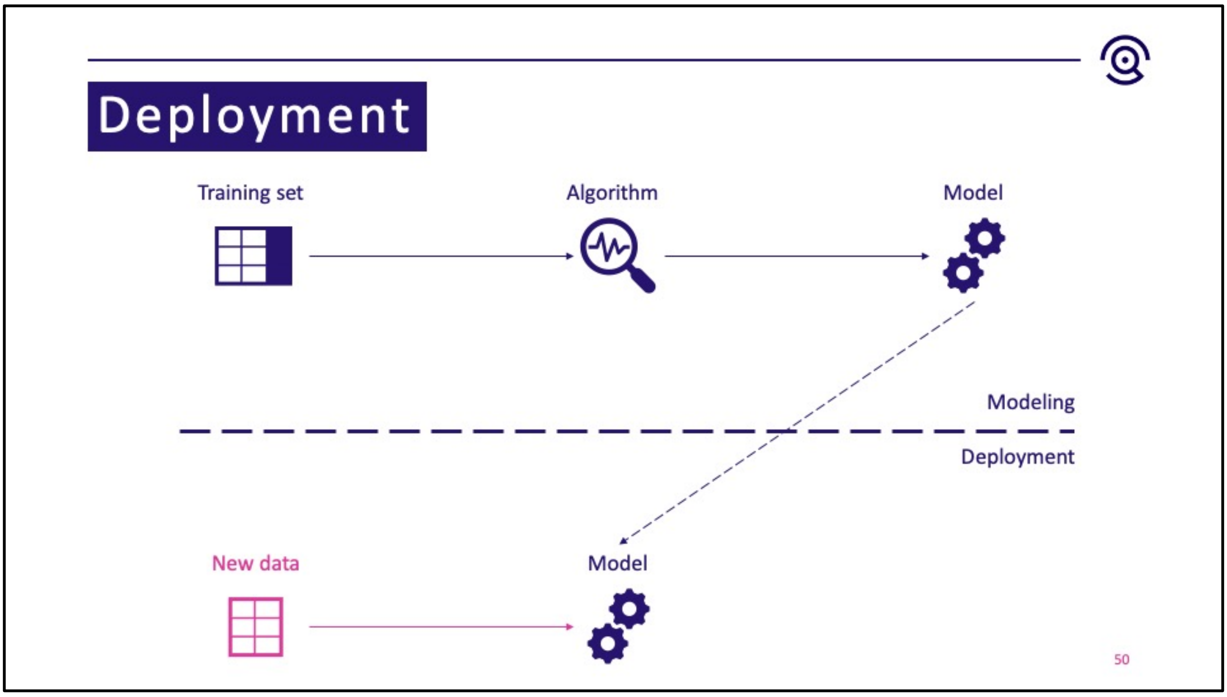
The predicted labels are then compared against the actual labels of the test data in order to measure performance of each model.



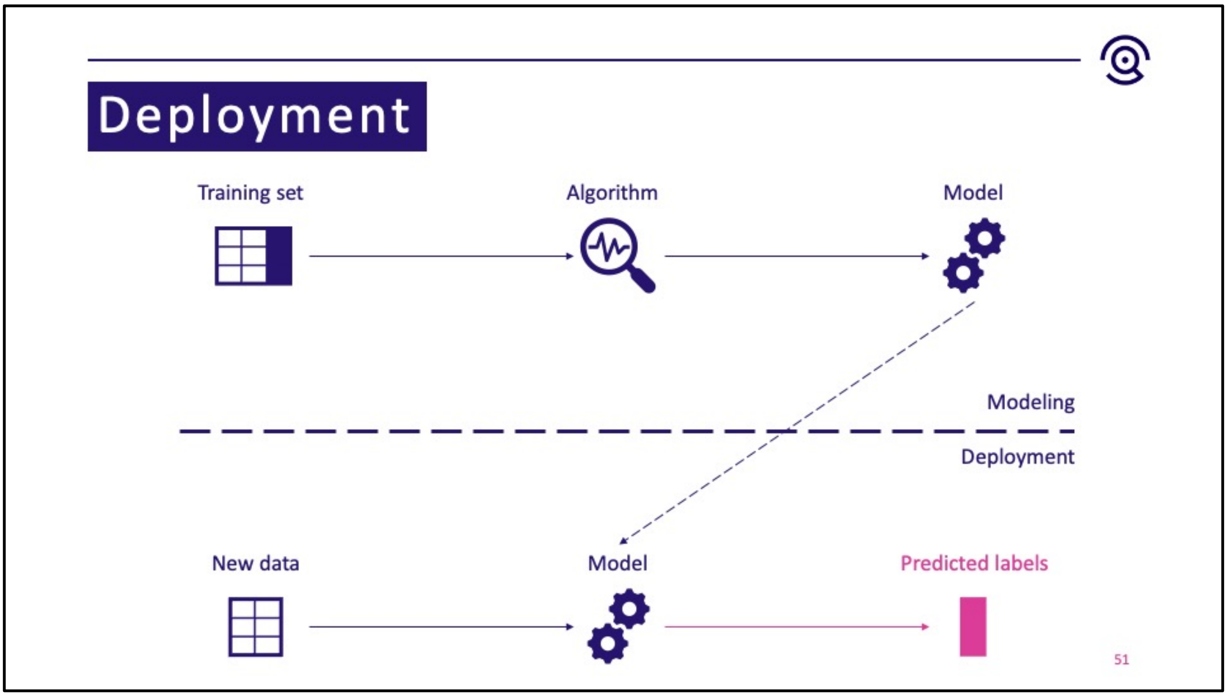
So far we focused on the modeling phase, namely step 3 in the data science life cycle.



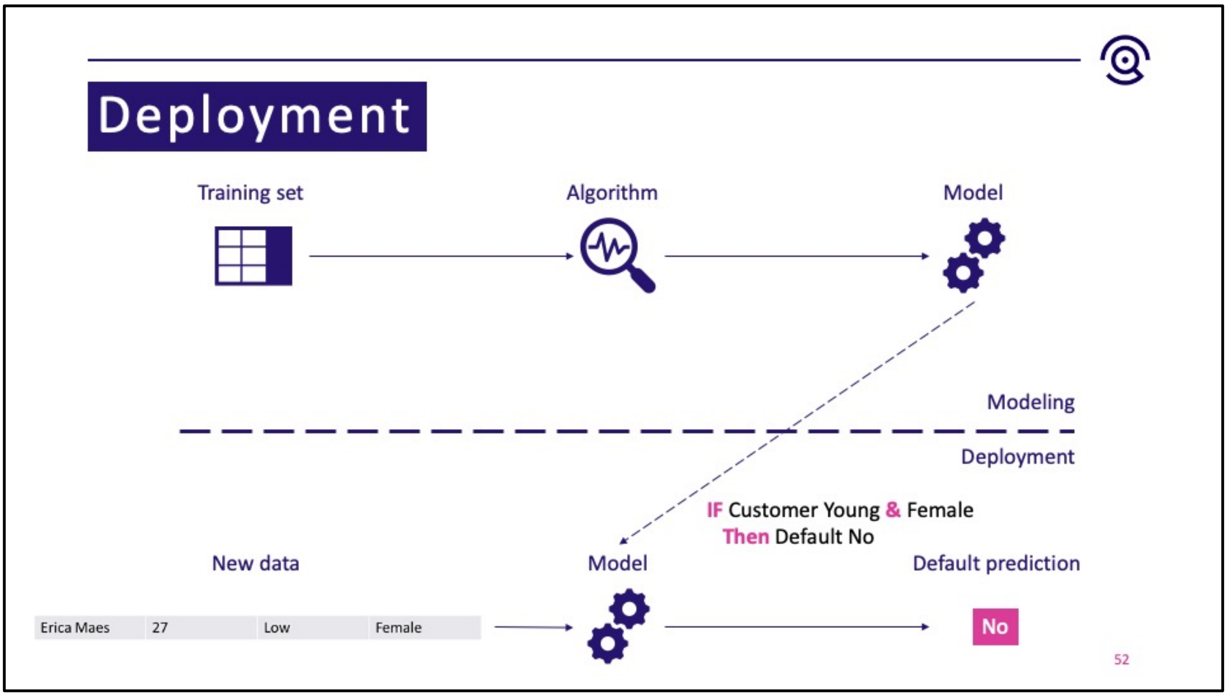
The goal is to take the ML model and deploy it in production.



We then feed in new data…

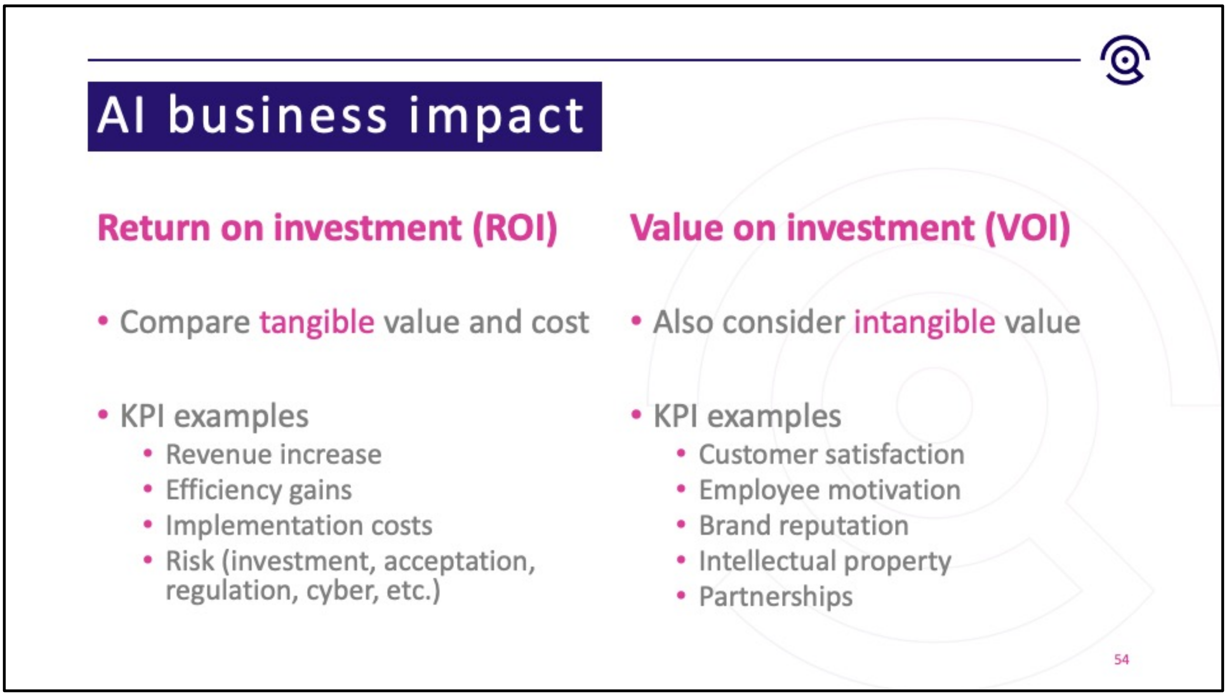


… to obtain predictions that are used to generate business value.

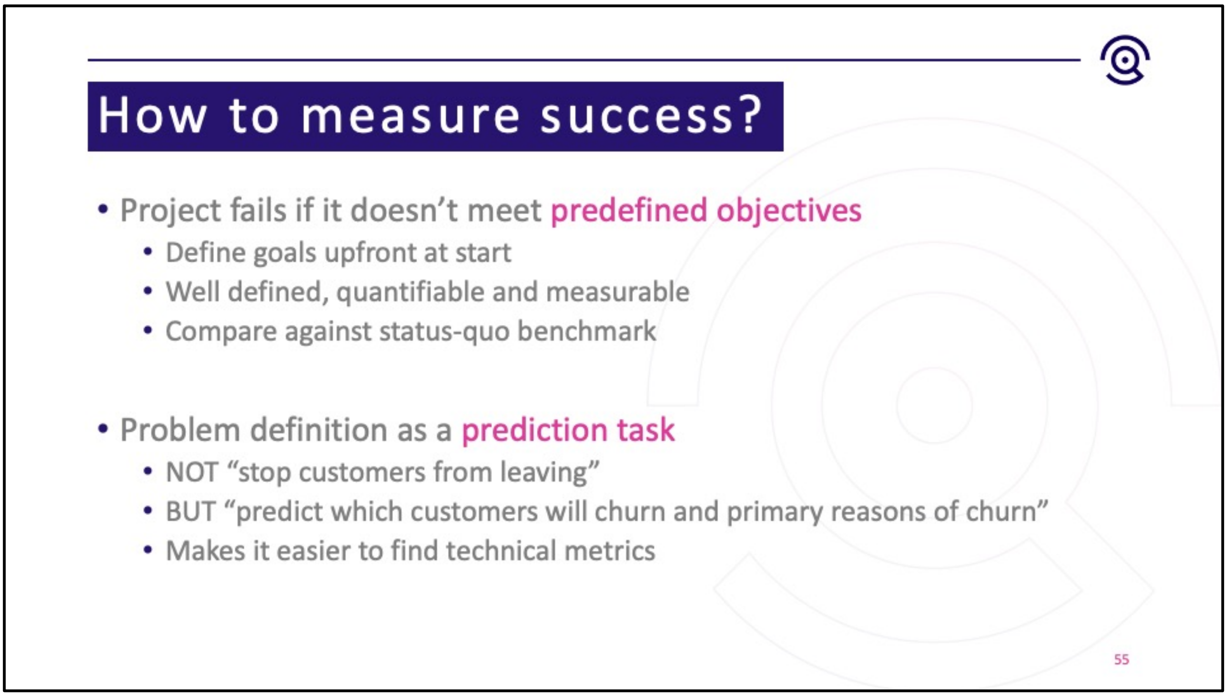


When a new customer comes to the bank, the teller can input the person's characteristics in the AI system. The model predicts no default so the customer is able to receive the loan and leaves the bank happy.

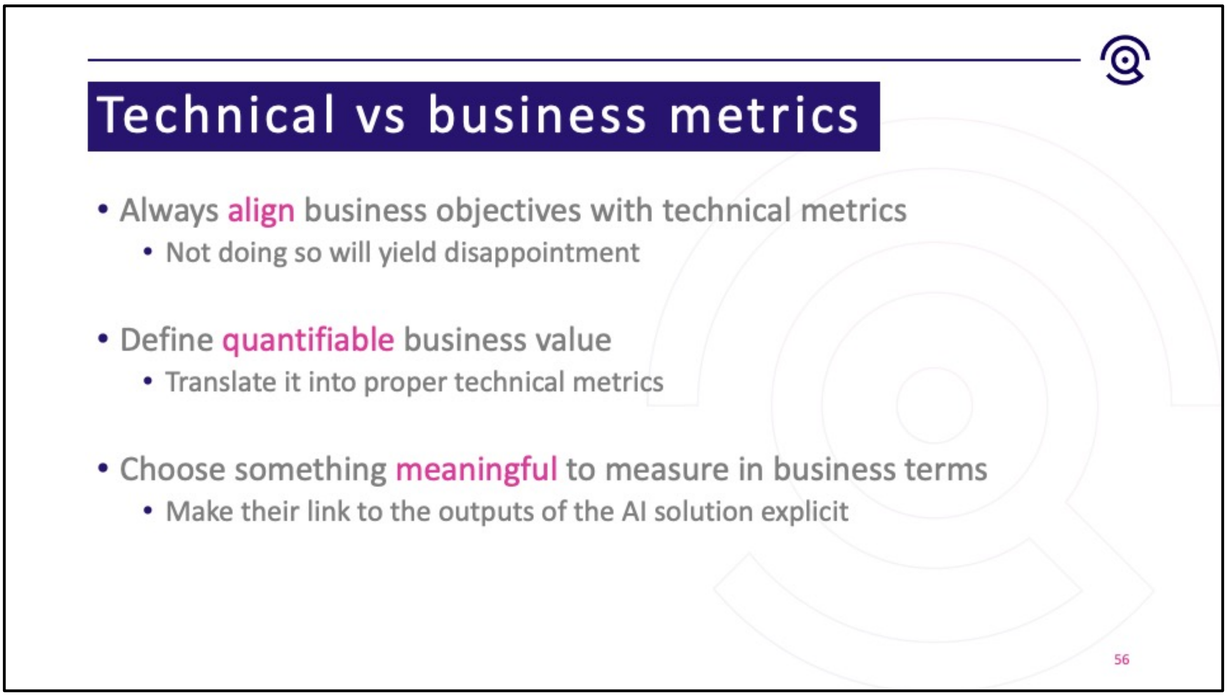
Once an AI project is finished it is important to evaluate the performance via key performance indicators (KPIs). Was the project a huge success or a total failure?



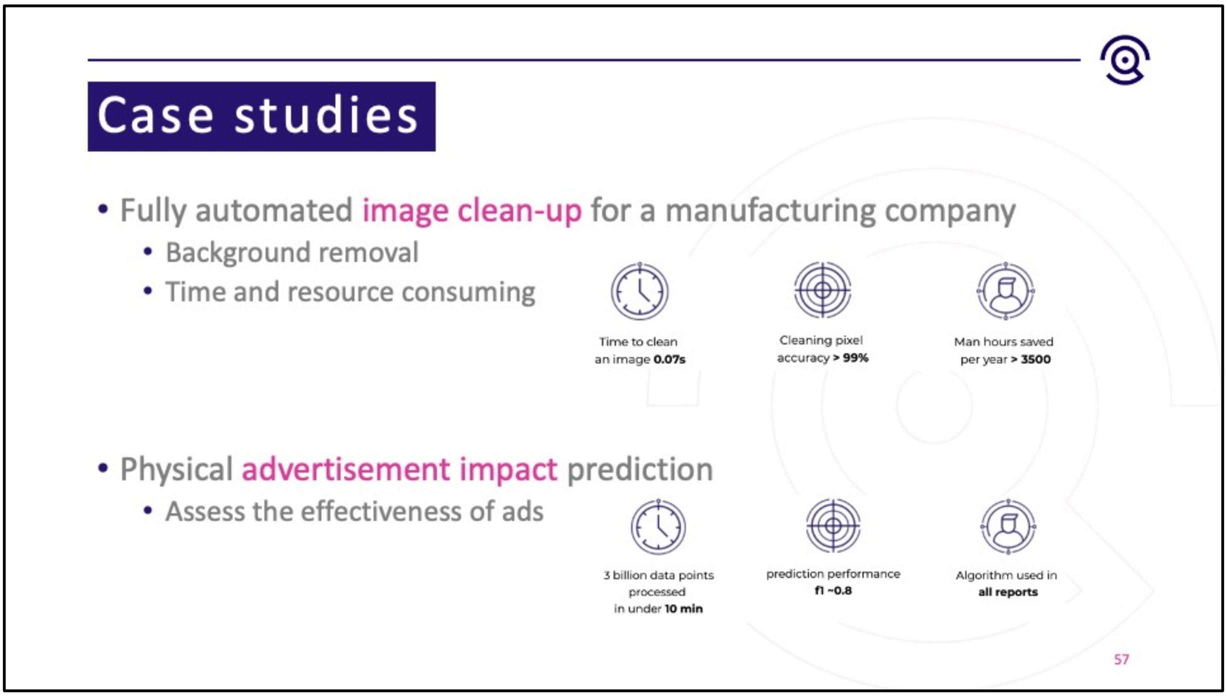
The business impact of AI solutions should be measured in two complementary ways. The first approach is the return on investment (ROI) which compares tangible value and costs. Typical KPI examples are revenue increase, efficiency gains, implementation costs and several types of risk (investment, acceptation, regulation, cyber, and so on). The second approach is the value on investment (VOI) which also considers intangible value created. Typical KPI examples are customer satisfaction, employee motivation, brand reputation, intellectual property and partnerships. Both ROI and VOI should be considered when evaluating an AI project.



How do we measure the success of a project? The project is considered as a failed endeavor when it doesn't meet the predefined objectives. It is important to define the goals upfront at the start of the project. Think carefully about this because an AI solution will only be as good as its objectives. Furthermore, these goals should be well defined, quantifiable and measurable such that they can easily be tracked and mean the same for every stakeholder. The real value can be discovered by comparing the performance against a status-quo benchmark. The project's problem definition should ideally be stated as a prediction task. So don't use "stop customers from leaving" as the goal but be more specific: "predict which customers will churn and the primary reasons of churn". Such specific goals make it easier to find technical metrics to evaluate the performance of the AI solution.



It is important to always align business objectives with technical metrics. Not doing so will yield disappointment in the long run. Start by defining quantifiable business value and then translate this into proper technical AI metrics. You can also choose something meaningful to measure in business terms and make the link to the outputs of the AI solution explicit. Let’s look at some examples.

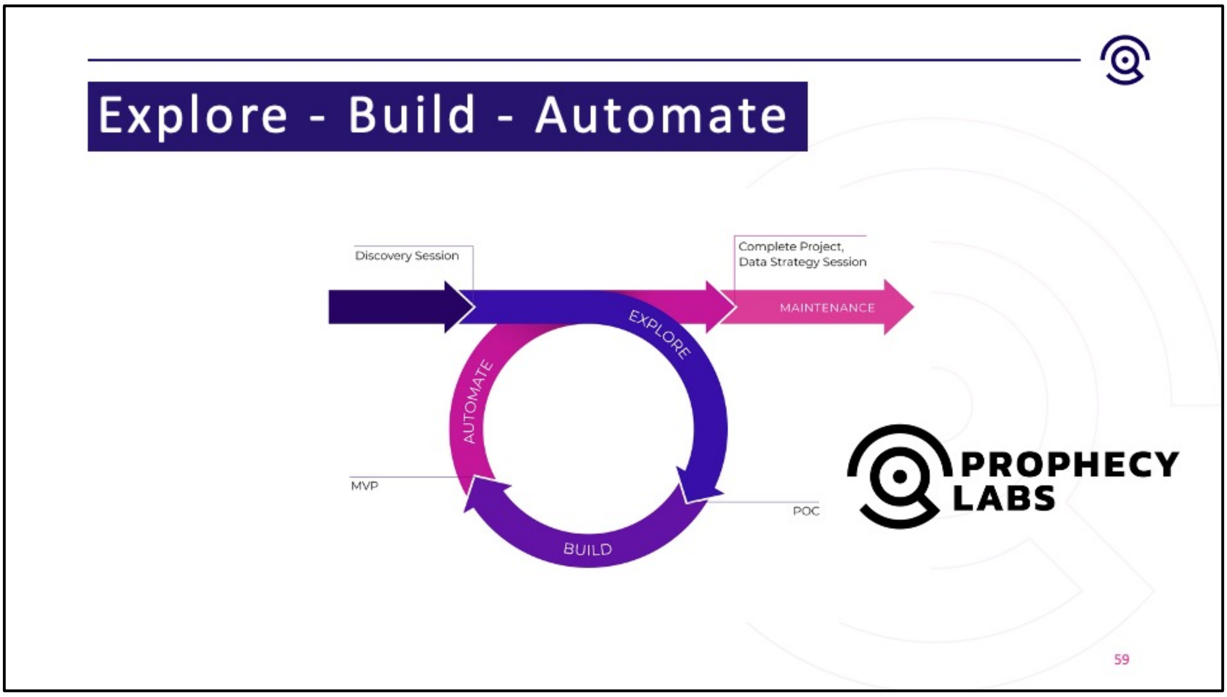


Any manufacturing company that produces tons of new products also produces tons of product images for its websites and catalogue. Most pictures are taken with a background which is normally removed manually using photoshop and other tools. This is very time and resource consuming. Our software reduced costs, radically streamlined and significantly sped up the image clean-up process by automating almost all steps required to fully process the images. The time to clean an image was only 0.07 seconds, the pixel cleaning accuracy exceeded 99% and over 3500 man hours were saved per year.

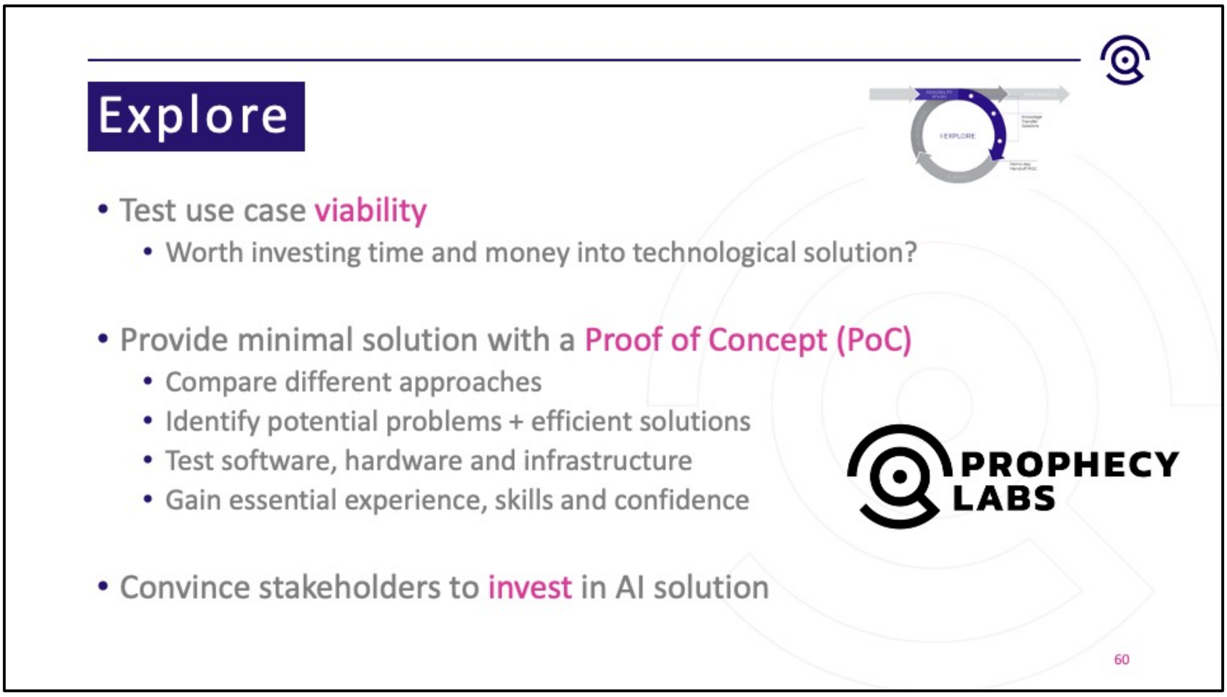
Assessing the effectiveness of physical ads for a specific target demographic is a pressing problem that can provide a multitude of actionable insights, ranging from the desired budget, the optimal poster placement or even the choice of advertising platform. Based on our algorithm that, among other things predicted whether users saw an advertisement, the client generated reports that were sold and used by most leading physical ads companies in Switzerland. We processed 3 billion data points under 10 minutes, the F1 prediction score was around 0.8 and our algorithms was used in all reports for ads.

Both examples show a combination of timing, technical performance and business impact KPIs.

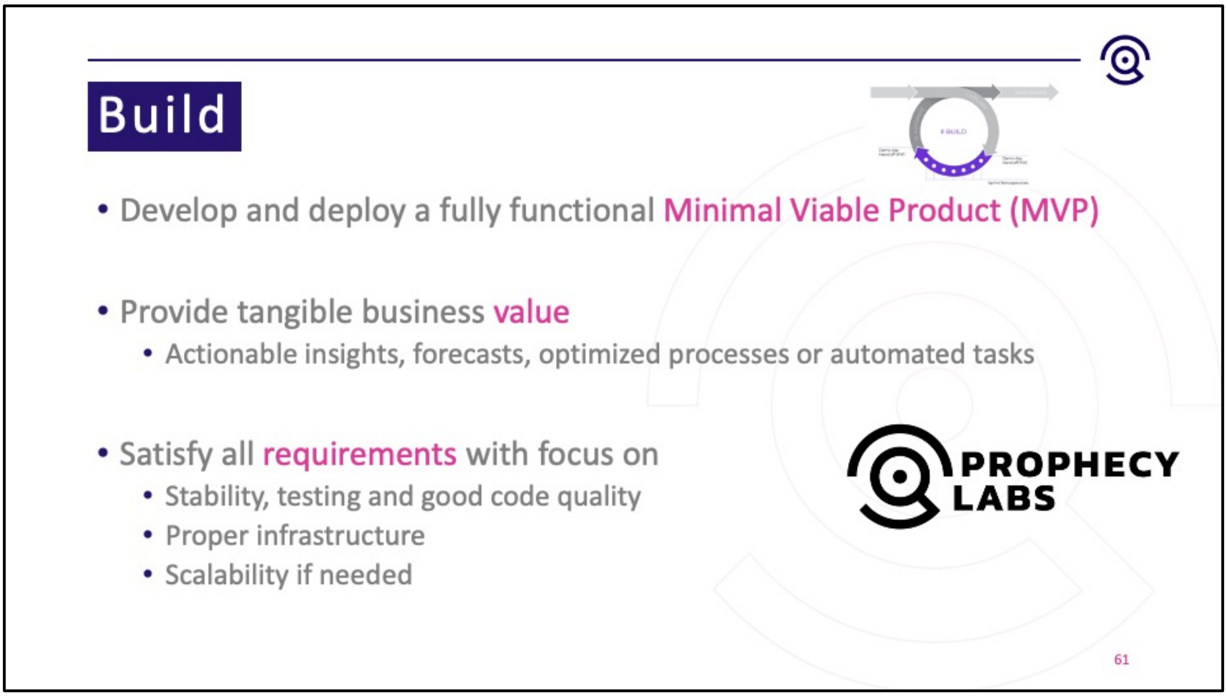
We will now introduce our own Prophecy Labs project strategy to develop AI solutions.



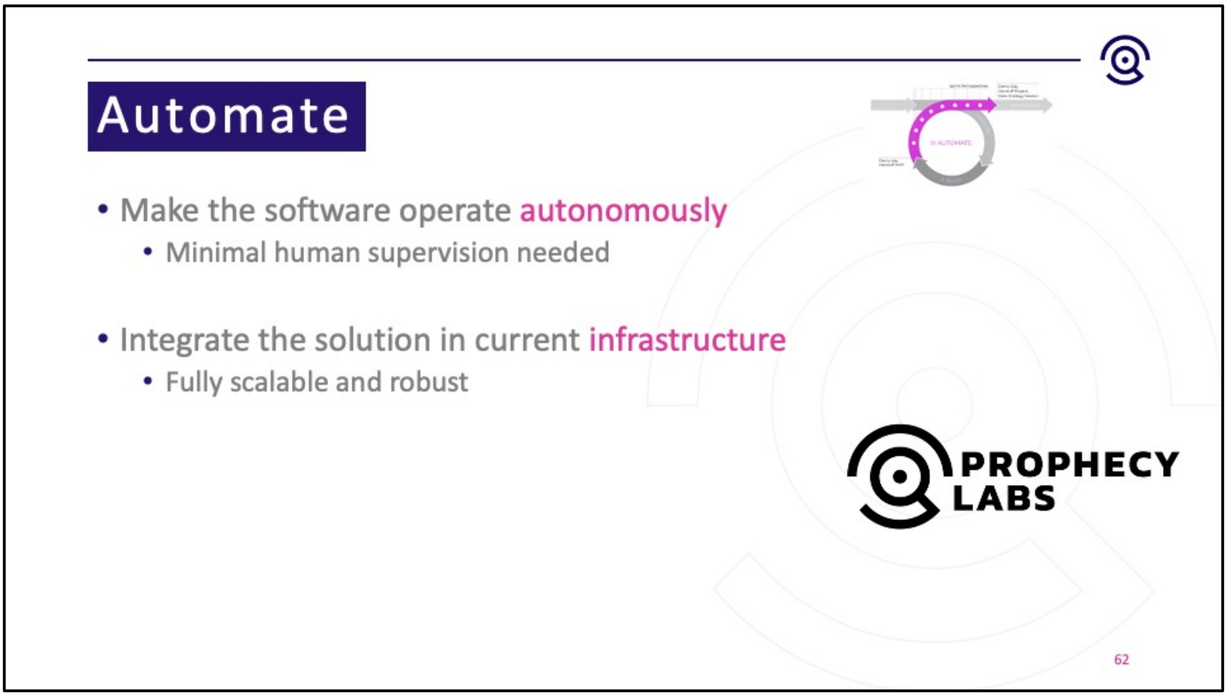
Our project strategy consists of three cornerstone phases: explore, build and automate. Each of the phases is separately scoped, billed and executed. This allows us to stay agile, provide a transparent approach and to deliver the projects on time while meeting or exceeding the expectations.



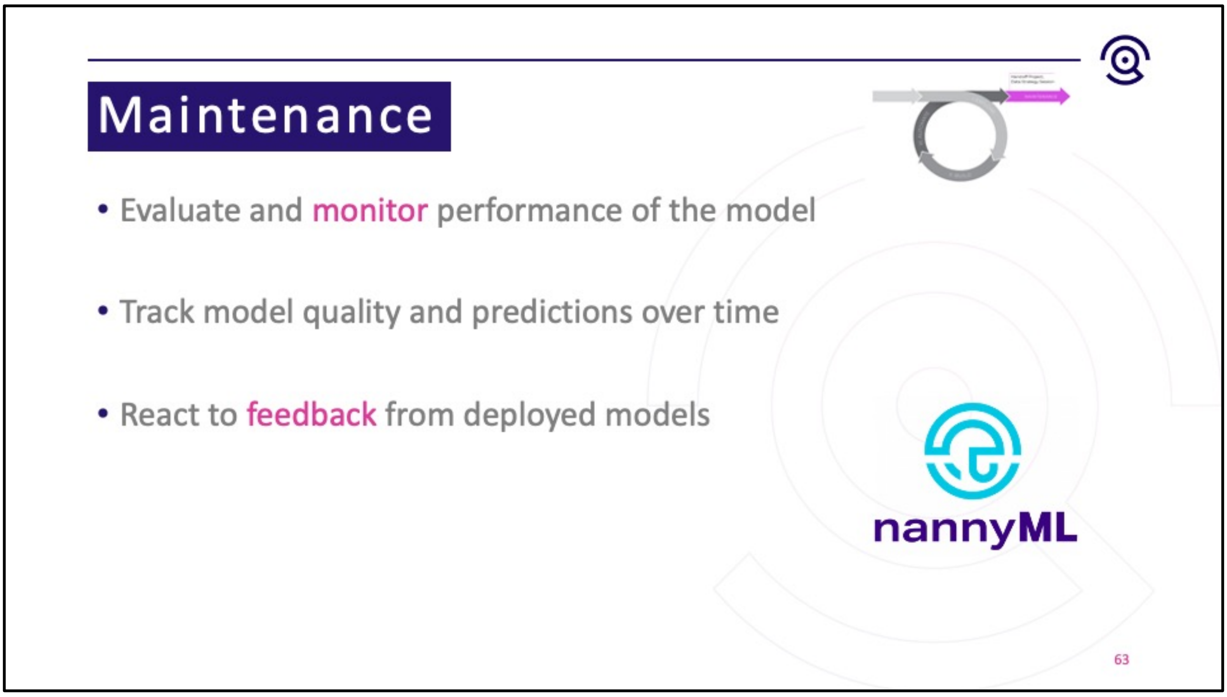
The goal of the explore phase is to test the use case viability by answering the question “Is it worth investing time and money into a technological solution?”. This allows us to offer quick advice without wasting your precious time or money. The explore phase provides a minimal solution to the problem with a proof of concept (PoC). We compare different approaches without spending too much time, identify potential problems and find efficient solutions. We also test software, hardware and infrastructure that is going to be used and make sure you gain essential experience, skills and confidence working with AI and ML. The PoC should convince the company’s business stakeholders to invest in the AI solution.



In the build phase we develop and deploy a fully functional minimal viable product (MVP). This solution provides tangible business value in the form of actionable insights, forecasts, optimized processes or automated tasks. The MVP satisfies all the project requirements with a focus on stability, testing, good code quality, proper infrastructure and scalability if needed.

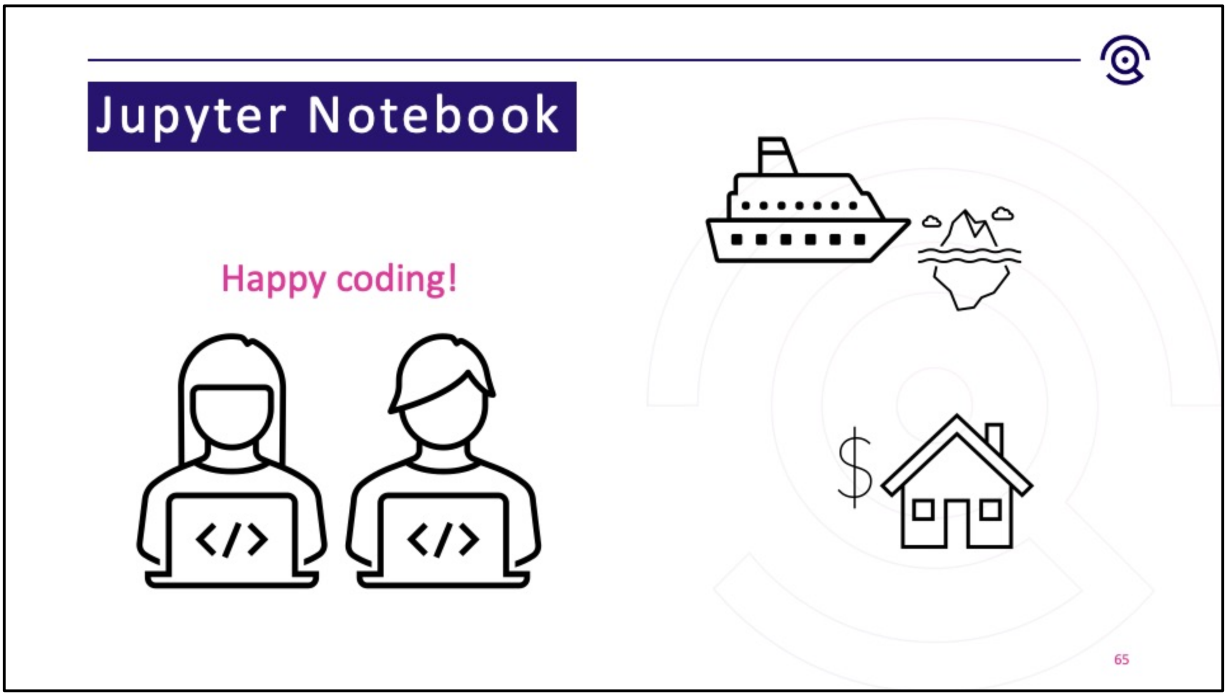


The automate phase makes the software operate autonomously with minimal human supervision needed. We integrate the solution in the current infrastructure such that is fully scalable and robust.



Maintenance is needed to evaluate and monitor the performance of the deployed model. The world is constantly changing, so it is important to track the model quality and predictions over time. This also allows to react to feedback from deployed models and improve them further.

The last part of this module it a hands-on Python tutorial with Jupyter notebooks where we go over parts of the data science lifecycle.



The videos of each tutorial can be found in the course material on module 2. There is a tutorial for a classification problem and one for a regression problem, with some hints on this slide already. Happy coding!

This is the end of the second module. I hope to see you again in the next module where we will, bye.