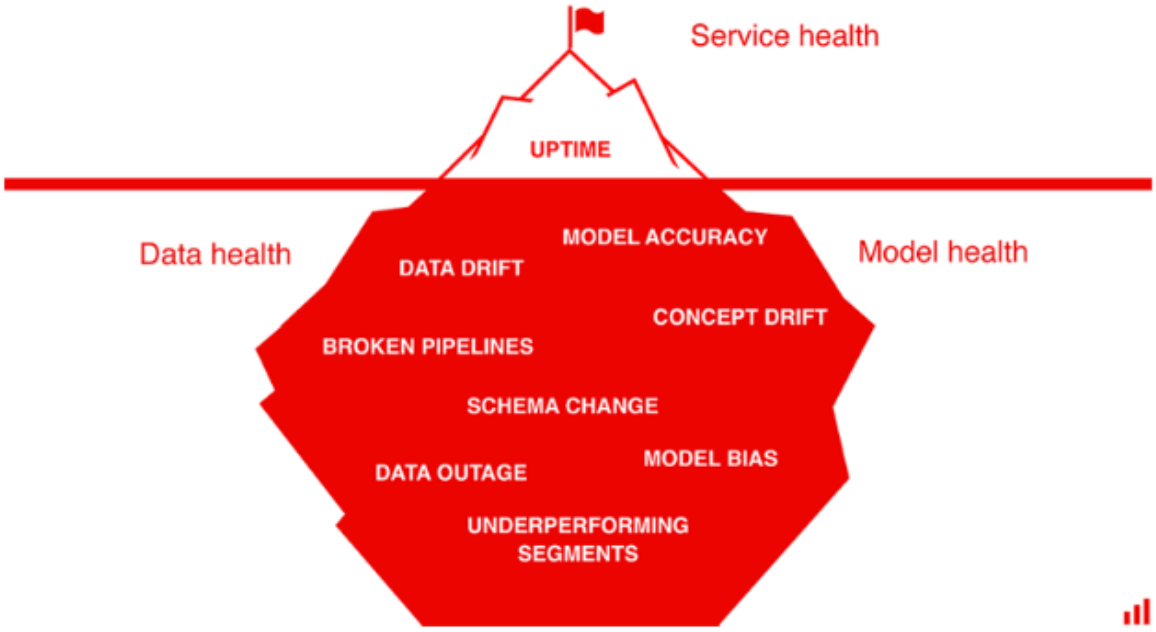
upGrad

System Design Process

After taking a look at the diagram below, you must have realised how difficult it is to keep an ML model in production. One of the major changes businesses saw was the change in customer behaviour during the COVID-19 pandemic. Customers' buying patterns and behaviour changed drastically. As the model that put into production was trained on historical data (pre-COVID), do you think it will function well with the data it is receiving now, which is entirely different?



Twitterbot, the example we took a look at earlier, failed in production because the historical data it was trained on did not anticipate people's racist comments. It did not have a disaster-recovery method and could not detect model bias.

Consider that the front-end team changed the user interface (UI) of the application to suit the audience due to which some of the variables/data that were being captured before are not being captured now. For example, the rating system was changed from a 5-star rating system to a thumbs-up rating system. In this case, this might result in schema changes and broken pipelines, which would affect the uptime of the model.

There could be multiple scenarios in which the model's uptime is reduced. To maintain the uptime, we must follow a proper ML system design process, which includes the MLOps best practices.

In this session, you will learn about the following key elements in detail:

- **ML model vs ML system**: Until now, you have developed your code in a Jupyter Notebook and have also tried to save the model parameters in joblib files. ML systems have many components other than ML model.
- **Design-thinking process**: You will learn about design-thinking process for a product life cycle.
- **ML systems design process**: You will understand how to inculcate the product life cycle in the ML system process and get to know how different stakeholders get involved at different stages.

ML Model Vs ML System

You learned that 'A system is a whole unit that consists of parts, each of which can affect its behaviour or its properties'.

The human body as an example of a system. The failure of any system, for example, the nervous system or the digestive system, can cause the entire body to fail. Similarly, if you take the example of a car as a system, if the brake fails, the entire system fails. It is also important to note that a system is not just a sum of the parts but a product of the interactions between them. If the wire between the brake and the tire fails and is unable to transmit the required information, even then the system will fail.

Therefore, you must remember that the parts of a system and the interactions between them make the system.

Similarly, an ML model is one part of an ML system. The ML code may be the 'brain' of an ML system, but it is still just one part of it. The entire system is much bigger as you can see in the diagram below from 'Hidden Technical Debt in Machine Learning Systems'.

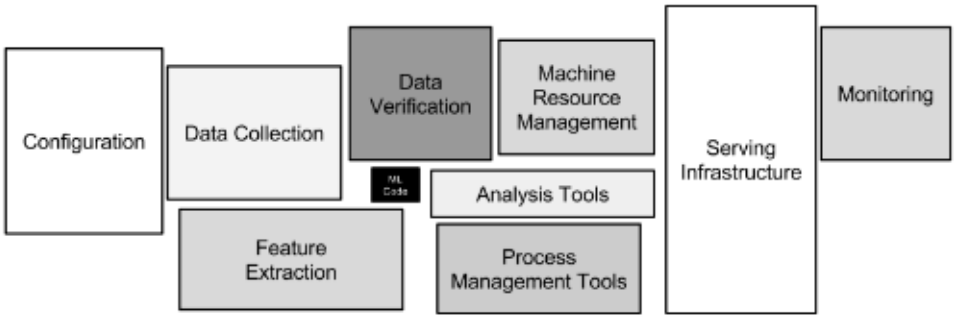
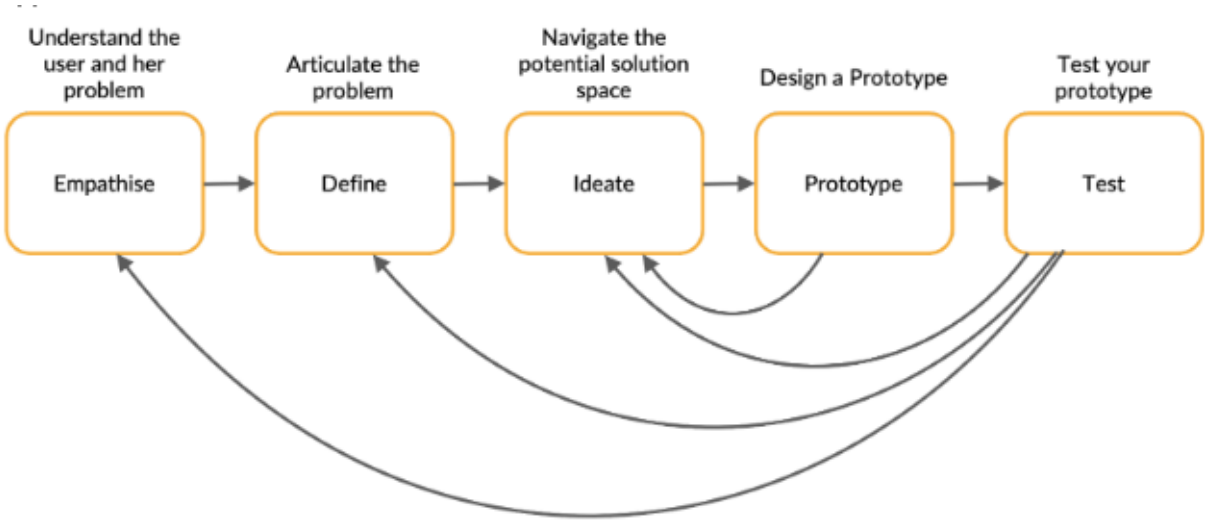


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

upGrad

Design-Thinking Process



Design thinking is a process that involves solving problems creatively. A design mindset can be applied to any life situation, and it aids in making decisions by helping one consider the bigger picture.

The five steps in the process of design thinking are as follows:

Empathise:

At this stage, what is required of you is that you put the user at the centre and understand all the problems that they face. Before you get started with this stage, you must also be clear about who your users are. The end users of the application may not be the users of your product. Let's say you are building a customer churn use case in telecom industry. In this case, your end users will be the internal sales and marketing team. Hence, the end users of a product should be identified properly.

Once you have identified the end user, you can then put yourself in their shoes and understand their problems. This stage is all about asking the right questions and gaining the relevant insights to formulate the core objective.

Define:
As you might have figured, users of a product might face multiple problems. At this stage, all the problems are navigated for the user, and the problem that will generate the maximum return on investment is chosen. We establish criteria for ranking and evaluating solutions/designs on the basis of feasibility.

Ideate:

In the 'define' stage, finalise the problem statements that you propose to solve. Ideation is about navigating the potential solution space for these problem statements. This is the brainstorming phase, where the team members who are involved in developing the product come together to ideate possible solutions.

Prototype:

At this stage, we build a set of potential solutions that we aim to test in the 'real' world. An ideal experiment or prototype will help you assess viability, usability, feasibility and value with the least amount of labour. Your prototype does not have to be the best product at one go. Rather, it should have the potential to measure the metrics. It should have the ability to adapt to changes.

Test:

The final stage is all about exposing your prototype to the environment it is intended to operate within. Based on the results of these tests, we measure and estimate the viability, usability, feasibility and value of the potential solutions/designs. You must remember that the end user is also a part of the design thinking process.

The design-thinking process has the following characteristics:

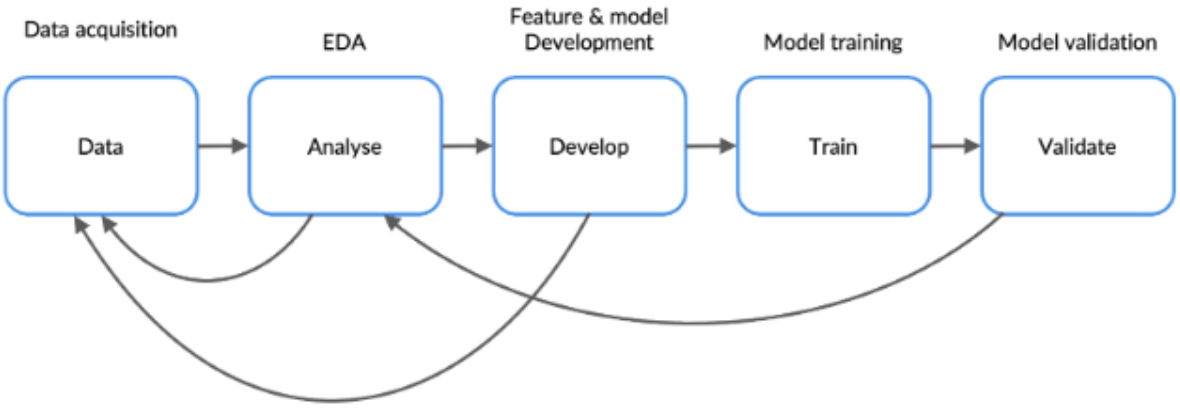
- **User-centric approach**: The user is kept at the centre of the product design process.
- **Non-linear process**: Suppose you already put a product or a new feature in production. Now, let's say you had a hypothesis that said that the users will like this product/feature. You did your market research; however, users did not adopt your product. Why do you think this happened? This happened because you based the prototype on the previous historic data.In some cases, you might have to go back and change your prototype or change the idea in ideation stage or you can entirely start making the product from scratch and start from empathise stage. In this changing world, customers' needs change so fast that it is always a non-linear process.
- **Rapid prototyping and testing**: As discussed previously, customer needs change fast. Hence, you must have a process whereby you can create rapid prototypes and test them efficiently to get user feedback.
- **Baseline approach**: A baseline is the simplest solution that you want to put in production. But this does not mean that you can create any random model. Certain

criteria must be adhered to. For instance, the performance of the baseline model should be better than the approach that is being followed in the present stage.

- **Solving the right problem**: Often, businesses end up solving the wrong problems, and doing this does not yield any results. Suppose you want faster horses so you can get from ‘a’ to ‘b’ faster. But is having fast horses the right problem to solve or we should solve the problem of reducing the time to get from a to b. This can be done using cars or vehicles as well. You must be careful if you are solving the root cause of the problem or the surface-level problem.

Many templates are available for the product design framework, but the main principles remain the same. While this template will initially be completed in sequential order, it will naturally involve non-linear engagement based on iterative feedback. We should follow this template for every major release of our products so the decision-making is transparent and documented.

ML Development And Operation Process



According to the diagram above:

Data acquisition:

You might have heard this popular phrase in data science: “Garbage in, garbage out”.

In this phase, you collect data sets from different sources. It is one of the most important steps, as the data that you collect will determine how well your model works.

Exploratory Data Analysis (EDA):

At this stage, you understand the different variables in your data, their distributions and interactions. This helps you in data cleaning as well as in feature engineering.

Feature and model development:

At this stage, you develop features from the raw data using proper pre-processing. After you have developed your features, you try out different models that are relevant to your use case.

Model training:

At this stage, you train your model by changing hyper parameters or the configurations.

Model validation:

You test your final model on the data that you have kept aside for testing and validation and decide if the model that you have trained is good enough to be moved to production. During this stage, data scientists are the ones working on the project.

You will understand what each block means. You saw these blocks when we discussed the MLOps life cycle:

Build: At this stage, you convert a Jupyter notebook into Python scripts. We also have to serialise the model. Serialising the model means saving the parameters of the final model as a pkl or joblib file. We did something similar on heart disease classification in the previous module.

Test: At this stage, we perform different types of testing. Unit testing is applied to functions in the Python script. We also have to check data and model tests. We test if the data that we are receiving in the production environment has the same data type and schema as that of the training data. In model tests, we check if the metrics we are achieving are the same as before.

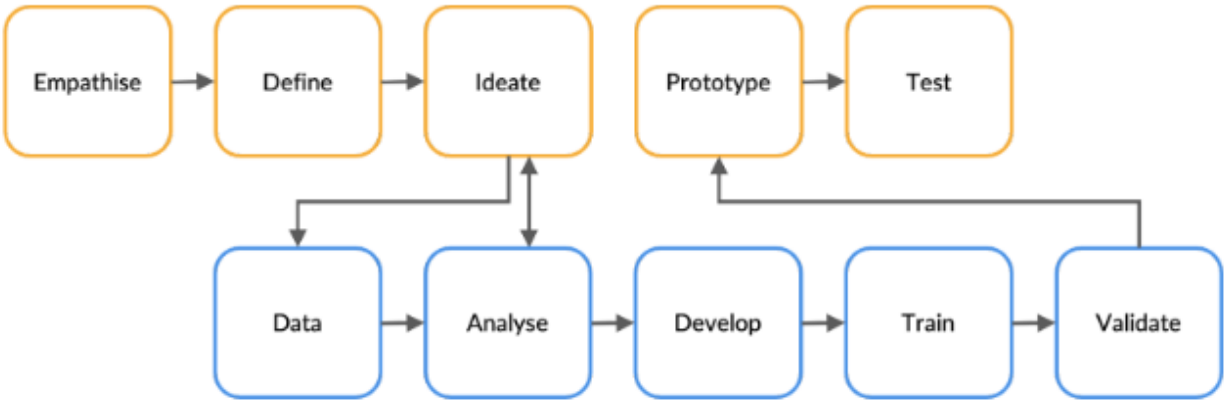
Release: At this stage, we version the code, model and data. We create different versions for different users. We also perform beta testing at this stage to decide which version works better for which users. For example, whenever Facebook launches a new feature, it is not released to every user but to a selected few users at the beginning. After careful evaluation of the feature, it is launched for all the users.

Deploy: At this stage, you deploy the model into a central server where all the users of that particular version can use it. This is similar to publishing a website. Initially, you work on your local system to make changes to the website to make it functional so users can use it. But then after you publish it by hosting it on a server, the website becomes available for everyone to access.

Monitor: After you have deployed a model, you must monitor it for different issues such as data drift and model drift. At this stage, the ML operations team and ML engineers are involved.

ML System Design Process

We have our product team and business team working on designing a product. If we combine only the model development cycle with the product designing cycle, we will get the flow depicted below.



Let's say that you, as a data scientist, have collaborated with the product design team to create an ML model for the recommendation engine of a food delivery application as discussed in the video.

After you have understood the user's problems with the food delivery application, you, along with the product team, decide on which problem to solve in the 'define' stage.

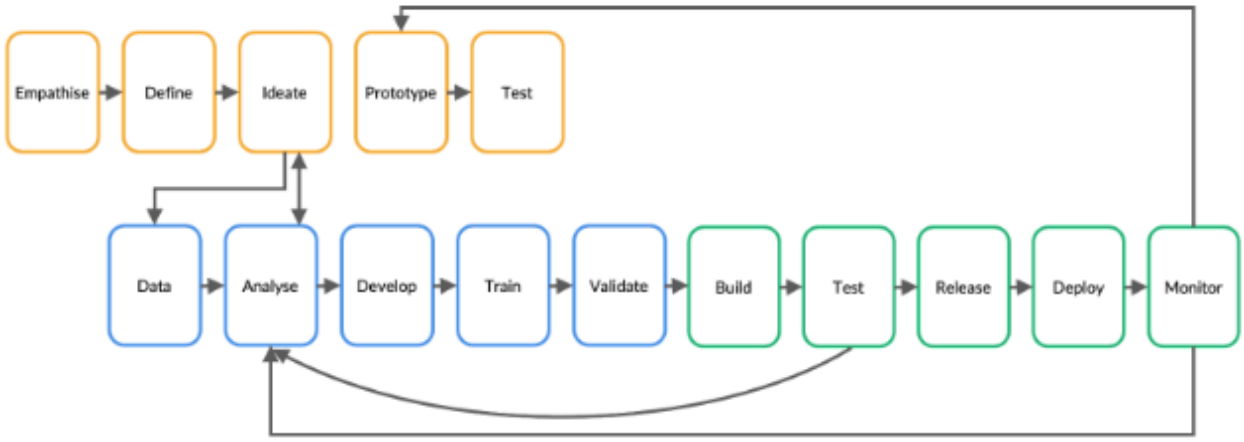
You then go into the ideate stage, where you evaluate different solutions to the problem. At this stage, you take a look at the data, analyse it and go back to the ideate stage if required. At the 'ideate' stage, different internal stakeholders are involved, such as the product, sales, and data science teams. The following discussions can happen at this stage:

- **Product team**: Create a model that recommends the best restaurants, which will maximise the clickthrough rates.
- **Sales team**: Create a model that recommends the best restaurants, which will maximise the sales.
- **Data Science team**: Create a model that has 99.99% accuracy, which will fulfil the criteria of the sales and product teams.

Now, at this stage, the team has decided to create a model that has the highest accuracy of the model. You, as part of the data science team, have started building the model, training it and validating it. The model that you create is then exposed through an API (Application Program Interface) and integrated into the prototype for user testing.

This seems like a straightforward and good way to deploy the model. However, you might run into some problems here.

The proper ML system design life cycle is given below.



Now, let's take the same scenario of the recommendation engine and try to see how we can apply the above system design process to that problem.

In our initial approach, we did not include the operations cycle in the design process. There could be a scenario in which the data science team comes up with a great model. The model also happens to be complicated.

When they want to put this complex model into production, the operations team has no resources to put that model in millions of servers as the cost is very high for GPUs required for deployment. Months of effort and resources would go to waste in this case. Hence, we must include the operations and machine learning engineering team in the initial discussions. In this case, it would be better if we first create a baseline version and then change it iteratively (as you saw in one of the discussions on MLOps best practices in the previous sessions).

Including the operations cycle in the design process will ensure that proper testing and monitoring are conducted before a model is put into production. If we happen to face a disaster-recovery situation, we can easily roll back to previous versions. This is because we have saved all the versions of the model, data and code at the release stage.

upGrad