**These are the Steps You Need to Take to Create a Good Data Science Product**

**From problem to production**

**A few years ago I built a machine learning application for a company. It had predictions, explanations of the predictions, a dashboard that combined many data sources and much more. Then the tool went live. And…it was hardly used. What went wrong? I had no idea: I had weekly contact with the business, the tool was tightly integrated with the existing system and I listened carefully to the wishes of the users.**

In hindsight I think I should have done many things differently. The tool was pretty complex and not intuitive. And I think we waited too long before we went live and should have had more business people involved. This brings me to an important question: *What is the right way to apply machine learning to solve business problems?* This article guides you through the essential steps. Questions you need an answer to in the beginning, data issues, tips for modelling and operationalising your model. I hope this prevents you from making the same mistakes as me!



Oops. Photo by [Sarah Kilian](https://unsplash.com/@rojekilian?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

Here is an overview of the steps I will explain in this post:

* Step 1. Get to the core of the problem
* Step 2. Understand and get to know the data
* Step 3. Data processing and feature engineering
* Step 4. Model the data
* Step 5. Operationalise the model
* Step 6. Improve and update

It isn’t always the case that you can start at step 1 and finish at step 6. Sometimes you need to iterate. During step 4, 5 or 6 you can discover ways to improve your model, for example after performing error analysis. You can return to a previous step, like creating new features (step 3) or gathering more data (step 2).

**Step 1. Get to the core of the problem**

The first step is probably the most important one. If you really want to build a good product, you should get to the core of the problem. You should dive into the material, talk to stakeholders, ask the right questions and think about technical requirements. This can take some time, but eventually you save time because the scope of the problem becomes smaller. You know up front where impediments might show up.

You can handle this step systematically. To make it easy, I have divided the step into six sub parts. These parts are value and purpose, possible solutions, people, technical aspects, process and legislation. Let’s walk through them.

**Value & purpose**

What is the goal of the product? What problem does it solve? Sometimes the actual question isn’t the true question behind the problem. To get to the true question, try to understand the business motivations and test your assumptions. How is success measured? What is the benefit for the end users? It might help to dive into the current process (if exists). This can give you a baseline performance and helps understanding the context.

Talking about performance, this is the time to establish a performance metric. When possible, use a single metric, because this makes it much easier to rank models based on performance. Try to find a simple, observable metric that is easy to explain to less technical people and covers the goal of the problem.

Another interesting question is: Are there others who can benefit from the product? This can help convincing people and makes the product even more interesting.

**Possible solutions**

After defining the value and purpose of the product, you can start thinking about possible solutions. You can do some research: Read literature dealing with similar problems or organise a brainstorm session with the team.

This part isn’t meant to completely solve the problem, but it will give directions during the actual solving phase. And maybe you discover that machine learning isn’t necessary, but that a rule-based approach might work also.

**People**

People involved are users, stakeholders, sponsors and the development team. Does the development team cover all aspects needed? Do you have enough technical expertise to complete the project successfully? Where should you go if impediments show up?

Discuss with the users how they can test the product. Involve them, the sooner the better. In early phases it’s easier to change and adjust: If you receive feedback early, you can implement it right away. Make sure you talk to everyone involved on a regular basis, which brings us to the process.

**Process**

How is the process managed? It’s a best practice to update the users and stakeholders regularly. When you work according to Agile principles, for example with scrum, it’s easy to fix the standard meetings (stand up, review, retrospective) on certain time slots. It is possible that the process is not fixed, for example because you work for a small company. Provide an update to those involved at least every other week. Try to quickly deliver a first version of the product so that your end users can test the product and provide feedback.

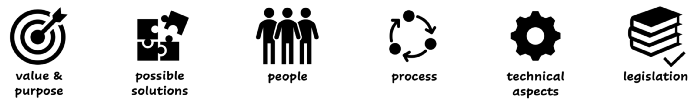
**Technical aspects**

Time to talk about data! Where is the data coming from? Is it accessible and available for the development team? When will it be updated? Besides data, think about other technical aspects, like the deployment, architecture, infrastructure, maintenance and tools that will be used.

If the solution will be integrated with other systems, don’t have a planning that is too optimistic. It’s easier to build a stand alone product, with the risk that it will be used less. Latency and throughput are also things to consider.

**Legislation**

A little less interesting, but no less important. Are there legal or ethical concerns you should take into account? Think about regulatory issues and how security will be arranged. You might also want to establish the impact of wrong predictions. How can you prevent that people will be harmed because of the predictions of your model?



The six parts of ‘getting to the core of the problem’. Image by author.

Asking the right questions that make the scope of the problem smaller will save you time later! If you don’t have an (in depth) answer to all of the above questions, it’s not an issue. Problems differ in scope and complexity. The easiest way to complete this step is to fill out a machine learning use case canvas, in consultation with the people involved. There are many use case canvases available online, you can try to find one that fits your needs or create one for yourself, based on the parts described above.

**Step 2. Understand and get to know the data**

The next step is all about the data. Data sources, understanding the data and data exploration.

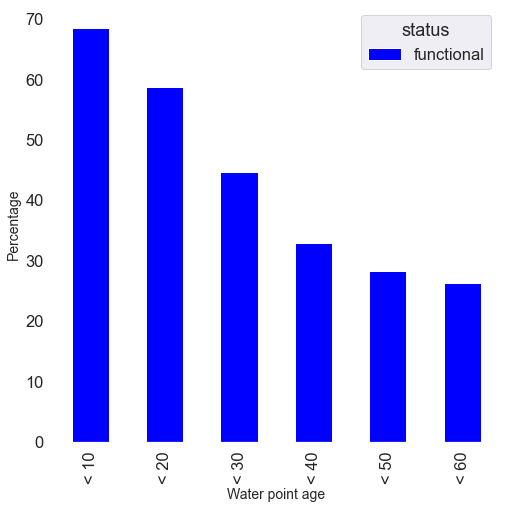
**Data sources**

The data sources you use are important, because the better the quality of the data, the better the model will perform. And do you have enough data? Or is there a need to acquire more via web scraping, data augmentation or maybe buying data?

Sometimes there is no data schema or data description available. If that’s the case, there has to be someone you can ask questions to. It’s hard (or impossible) to understand the meaning of tables and columns without any explanation or description.

**Exploratory data analysis**

Now it’s time to get your hands dirty and start with exploratory data analysis. Create summary statistics and plot distributions, histograms, bar and count plots. Try to find the first relationships between variables and the target, to discover features with predictive value, for example with a correlation matrix. Features without variance or with many null values can be flagged to remove in the next step.



An easy bar plot with a clear relationship between age (feature) and functionality (target). Image by author.

**Step 3. Data processing and feature engineering**

The insights from the exploratory data analysis are input for the next step: data processing and feature engineering.

**Data processing**

During the data processing step, you drop irrelevant data, clean missing values, remove duplicate rows and [detect](https://towardsdatascience.com/are-you-using-feature-distributions-to-detect-outliers-48e2ae3309) and [take care of outliers](https://towardsdatascience.com/dont-throw-away-your-outliers-c37e1ab0ce19). Errors in the data also need to be addressed.

**Feature engineering**

Then you can start creating features. It depends on the case what features will work best. Some basic suggestions for quantitative variables are transformations or binning. If the dataset has a high number of dimensions, dimensionality reduction like UMAP or PCA can be effective. For categorical variables you can try one hot encoding or hashing.

This step is a bit more complicated when you work with unstructured data. For textual data, you need to take care of stemming, lemmatisation, filtering, bag-of-words, n-grams and word embeddings. With images, you need to take care of noise, color scales, enhance the image or detect shapes.



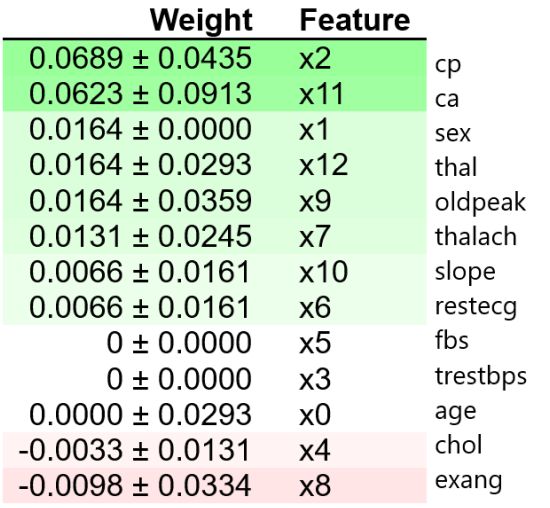
Image processing can be time consuming. Image by author.

**Step 4. Model the data**

Try different models on the processed data. The type of model depends on multiple factors, like the training and prediction speed, volume and type of data and the type of features. Some projects require an explainable model, while with others performance is more important. If you want to use a model that is hard to explain but where explainability is important, [here](https://towardsdatascience.com/model-agnostic-methods-for-interpreting-any-machine-learning-model-4f10787ef504) are some methods you can use.

During the model evaluation phase, you can use a train, validation and test split and/or cross-validation. Tune hyperparameters and compare different models. Detect the importance of different features and check (if necessary with the business) if these features make sense. Regularise models to avoid overfitting and make sure you handle data imbalance. Train the final model on the complete data set.

Share your results and the performance with the team and stakeholders. From this step, you can decide to continue and operationalise the model, or you can go back to the data processing step to extract new features.

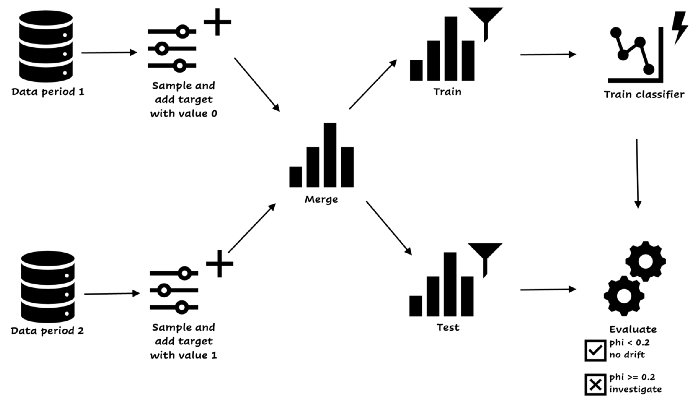


Permutation feature importance, one of the ways to interpret a machine learning model. Image by author.

**Step 5. Operationalise the model**

The deployment process is called MLOps. You can use different tools here, like [MLflow](https://mlflow.org), [Airflow](https://airflow.apache.org) or a cloud based solution. Decide if you can make predictions in batch or if it’s necessary to predict in real time. This determines if you need to focus on high throughput or latency, respectively. A hybrid approach is also possible.

When performance degrades, there should be a process that automatically retrains the model with new data. And be aware of data drift, if the model is really important and it’s interesting to know how the data changes, a [data drift process](https://medium.com/bigdatarepublic/detecting-data-drift-with-machine-learning-adb177544312) could be a good addition.



A way to detect covariate drift with machine learning. Click to enlarge. Image by author.

**Step 6. Improve and update**

It would be nice if you could say: ‘My model is live, let’s start with something new!’ Unfortunately, in real life, that’s not how things work most of the time. You should keep track of the model and the business objectives to make sure your model keeps performing the way it should. You can perform error analysis to analyse wrong predictions.



Error analysis. Photo by [Cookie the Pom](https://unsplash.com/@cookiethepom?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

**Summary**

Creating a good data science product can be tough. You have to deal with many things besides modelling, like users and stakeholders, data, deployment and maintenance. This article helps you by explaining best practices during the phases of the machine learning lifecycle.

First, get to the core of the problem. No need to solve it yet, but make sure you get all the information you need to convince yourself you *can* solve it and that the case is worth it. This step consists of gathering the right people, establishing the product goal and the measures of success, baseline performance, and an overview of technical aspects, like data sources and deployment.

When you truly understand the problem, you can dive into the data. Start with an exploratory data analysis, followed by data processing and feature engineering. Then, it’s time to model the data. It’s possible that you go back and forth between data sources, feature engineering and modelling. For example when the performance of the model isn’t good enough.

If the results of the model are satisfying, you can deploy your model. Keep track of the performance, and make sure you have a retraining process in place. If necessary, keep improving the model, for example with error analysis.