Machine Learning Workflow

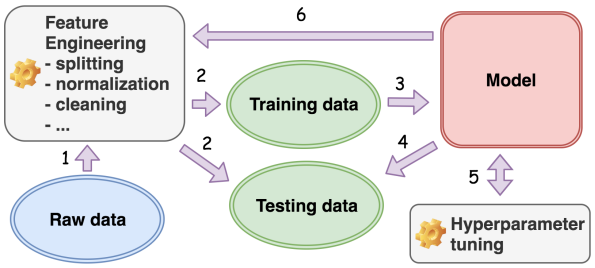
In the previous section, we clarify the notion of machine learning model. In this section, we discuss a typical `workflow` to construct a machine learning model.

First of all, one cannot talk about machine learning, without mentioning about the `data`. The relationship between the data and the machine learning model, is as critical as the fuel to the engine of rocket.

Data-Centric Workflow

> The workflow to build a machine learning model is centralized around the data.

It is not exaggerating to say that the data dictates how the machine learning model is built. In the following graph, we illustrate a typical workflow involved in a project of machine learning.



Starting from the data, we first determine which type of machine learning problems we would like to solve, *i.e.* ***supervised*** vs. ***unsupervised***. We say that the data is *labeled*, if one of the attributes in the data is the desired one, *i.e.* the target attribute. For instance, in the task of telling whether there is a cat on a photo, the target attribute of the data could be a boolean value [Yes|No]. If this target attribute exists, then we say the data is labeled, and it is a supervised learning problem.

For the supervised machine learning algorithms, we further determine the type of the generated model: ***classification*** or ***regression***, based on the expected output of the model, ***i.e.*** discrete value for classification model and continuous value for the regression model.

Once we determine on the type of model that we would like to build out of the data, we then go ahead to perform the ***feature engineering***, which is a group of activities that transform the data into the desired format. Here are a few examples:

- For almost all cases, we ***split*** the data into two groups: training and testing. The training dataset is used during the process to train a model, while the testing dataset is then used to test or validate whether the model we build is generic enough that can be applied to the unseen data.
- The raw dataset is often ***incomplete*** with missing values. Therefore, one might need to fill those missing values with various strategies such as filling with the average value.
- The dataset often contains categorical attributes, such as country, gender *etc.* And it is often the case that one needs to ***encode*** those categorical string values into numerical one, due to the constraints of algorithm. For example, the Linear Regression algorithm can only deal with vectors of real values as input.

> The process of feature engineering is ***not*** a ***one-off*** step. Often one needs to repeatedly come back to the feature engineering later in the workflow.

Once the data is ready, we then select one of the machine learning algorithms, and start to feed the algorithm with the prepared ***training data***. This is what we call the ***training process***.

Once we obtain our model at the end of the training process, we then test the model with the reserved ***testing data***. This is what we call the ***testing process***.

It is rarely the case that we are happy with our first trained model. One would then go back to the training process and tune some parameters that are exposed by the model that we selected. This is what we called the ***hyper-parameter tuning***. The reason that it is highlighted as 'hyper' is because the parameters that we tune are the outermost interface that we interact with the model, which would eventually have impacts on the underlying parameters of the model. For example, for the decision tree model, one of its hyper-parameters would be the maximum height of the tree. Once set manually before the training, it would limit the number of branches and leaves that a decision tree can grow at the end, which are the underlying parameters that a decision tree model consists of.

> As one can see, the steps involved in the machine learning workflow form a ***cycle*** with a focus on the data.