A Why Machine Learning

After the previous chapters, one should be able to tell in general `what` Machine Learning (ML) algorithms are, and should have a brief idea on `how` to apply ML in a project.

Now, in this chapter, it would be the right moment to reflect a bit more on the question: *`why` do we need ML algorithms ?*

First of all, let's acknowledge that at this moment (2018) we do need the ML algorithms in many aspects of our lives. Noticeably, it is omnipresent in the Internet services (*e.g.* social networking, search engine *etc.*) that we are indulging daily. In fact, as revealed in a recent paper from Facebook (https://research.fb.com/publications/applied-machine-learning-at-facebook-a-datacenter-infrastructure-perspective/), ML algorithms become so important that Facebook started to redesign their datacenters from hardware to software, to better cater to the requirements of applying ML algorithms.

> "*At Facebook, machine learning provides key capabilities in driving nearly all aspects of user experiences... Machine learning is applied pervasively across nearly all services.*"

Here are a few examples of how ML is applied in Facebook:

- Ranking of stories in the News Feed is done via ML.
- When, where and who to display Ads is determined by ML.
- The various search engines (*e.g.* photos, videos, people) are each powered by ML.

One could easily identify many other scenarios where ML is applied, in the services (*e.g.* Google search engine, Amazon e-commerce platform) that we are using nowadays. The ubiquitous presence of ML algorithms becomes a norm in the modern life, which justifies its raison d'etre at least for the moment and the near future to come.

Why ML ?

> ML algorithms exist, because they can solve problems that non-ML algorithms are not able to, and because they offer advantages that non-ML algorithms do not have.

One of the most important characteristics that tells a ML algorithm apart from non-ML ones, is that it decouples the model from the data so that a ML algorithm can adapt to different business scenarios or the same business case but with different contexts. For instance, a classification algorithm can be applied to tell if there is a face shown on a photo. It can also be applied to predict if users are going to click on an Ads. In the case of face detection, the same classification algorithm can be used to train a model that can tell whether or not there is a face presented on a photo, as well as training another model that tell precisely who is presented on the photo.

Through the separation of model and data, ML algorithms can solve many problems in a more *flexible*, *generic* and *autonomous* manner, *i.e.* much like a human, the ML algorithms seem to be able to `learn` from the environment (*i.e.* the data) and adjust its behaviors (*i.e.* the model) accordingly in order to solve a specific problem. Without explicitly coding the rules (*i.e.* the model) in the ML algorithms, we construct a sort of meta-algorithm that is able to *learn* the rules/patterns from the data, in a supervised or even unsupervised manner.