Machine Learning : Practical Work

Abdelhak KHEMIRI

November 30, 2021

The goal here is to review in a practical way a good part of the different elements seen during the lessons concerning Machine Learning (ML). To do so, you will have to conduct a ML project from the start to the end, on a dataset named "ChronicDisease.csv" (available in the resources of this lesson).

The overall objective is to build an ML model to predict the chronic kidney disease. You will have to compute the following performance measures ***accuracy***, ***precision***, *and* ***recall***, and select the most adapted one for the task, justify your choice. Also note that as you will have to pre-process the data. The documentation of the DataSet is given in the "DataDocumentation.txt" file (available in the resources of this lesson).

The data are blood tests and other measures from patients with and without chronic kidney disease. There are 400 rows, one per patient; these are patients seen over a period of about two months at some point before July 2015, in a hospital in India.

1 Main Tasks

- Load the dataset "ChronicDisease.csv" in a dataframe named **data**
- Research the model that will be best for the dataset
  - Try with the 3 ML algorithm that we have seen in the course (SVM, KNN, Decision Tree)
  - For each ML algorithm "A_i" create a function named preprocessing_A_i() that take a dataframe, and where you will perform preprocessing and feature engineering tasks.
  - For each of the three algorithms, perform a hyperparameter optimization and evaluate their performance
  - Select the best performing one based on your selected performance measure.
- Using the Scikit-learn or TensorFlow/Keras libraries, explore and propose another ML algorithm different from the ones seen in the course.
- Once you have selected the best algorithm and, train the model and evaluate its performance
- Name your final model as "production_model". Suppose that it is going on production

The report of this practical work will be a Jupyter Notebook named as : **name_firstname.ipynb** (or **name1_firstname1-name2_firstname2.ipynb**, if you are doing this project in groups of two).

The report (Jupyter file) should be structured, have an easily readable code with explanatory comments. It also should contain markdown cells explaining and justifying the steps you're undertaking (To compile text in a Jupyter cell, you can change its code type to markdown (see top right of the notebook).