HW3 - Data Analysis

Camille Hascoët

December 13, 2023

1 Task 2

For this task, I use two correlations matrices. The first one is to determine whether a feature is correlated or not to the target, and the second one is to determine if two features are correlated between each others. I can then set thresholds to keep correlated features to the target and then remove some that are correlated between each others. The threshold were chosen by testing multiple times the program with different ones. It is important to say that I preprocess the features using a $MinMaxScaler$ and $PolynomialFeatures$ preprocessor. I tried with and without, and it was clearly better with, so I stuck with those.

2 Task 3

Here, I chose 3 models : $SupportVectorMachine$, $LogisticRegression$ and $RandomForest$. I chose their parameter using a $GridSearch$ with 5 cross validation. The choice were made based on simple criteria :

-RF : It is often the method in which I had the best results in my past programs, so I wanted to use it there.

-SVM : As I am using a polynomial feature preprocessor, I thought using this type of classifier would be a nice idea as it can handle high dimensional spaces and non-linear data.

-LR : It is a pretty simple classifier, really easy to understand and implement, and I wanted to know what kind of results it would get here.

3 Task 4

For determining the best number of clusters, I used the $Kmeans$ unsupervised method and I compared with two metrics : the $wcss$ score (or the $inertia$) and the $silhouette$ score. You can see the results on the graph below.

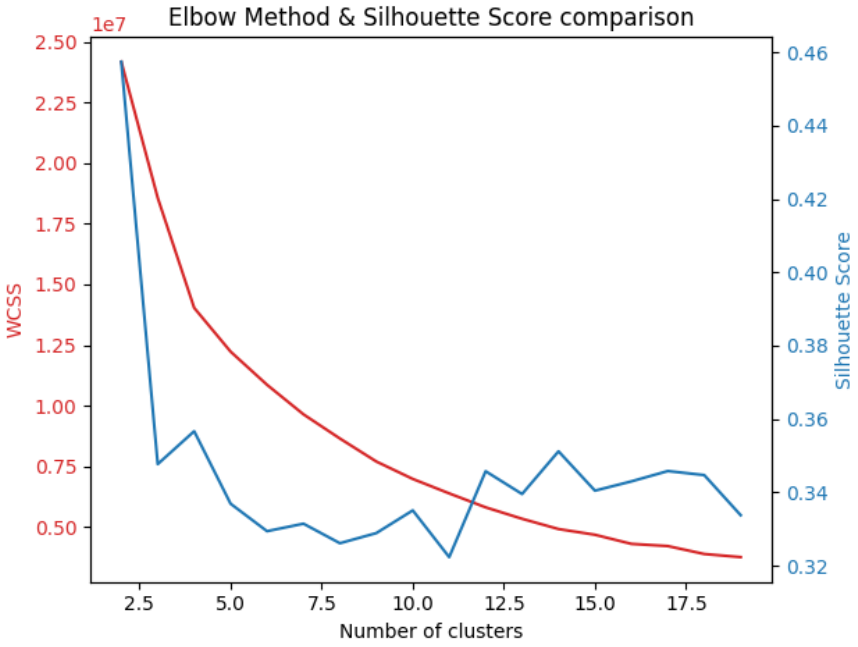

Figure 1: CWSS and Silhouette functions of n clusters

We can see the elbow for the *wcss* values at $n\_clusters = 4$ and the best value for the *silhouette* score at 4 too, so I'd say the best number of clusters would be 4 for this dataset.

4 Task 6 - Results

You will see below the confusion matrices for the three different models :

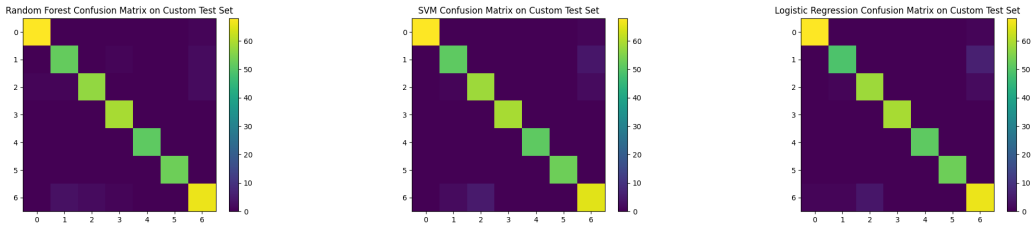

Figure 2: Comparison of the three different models' confusion matrices

Finally, you will find below the macro-averaged precision and recall values in the demanded precision-recall space.

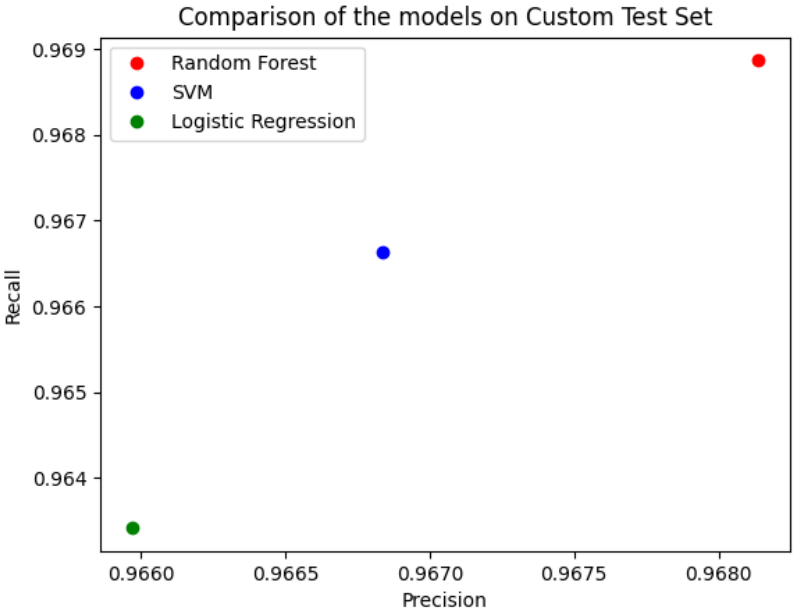

Figure 3: Precision and Recall for all three models

At the end, the results were pretty much as I expected : the random forest is the best, as I often find in my previous programs. Then the SVM which I thought was a good idea for this homework and then the LR that I found surprisingly good here !