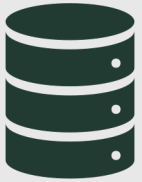
PYTHON FOR DATA ANALYSIS
FINAL PROJECT

DE WATRIGANT Albert
CROCHET Tom

Contents

I. Understanding the data

The game Starcraft2
The skillcraft1 dataset
The target
The features
Assumptions and preliminary thoughts



II. Preprocessing and data vizualisation

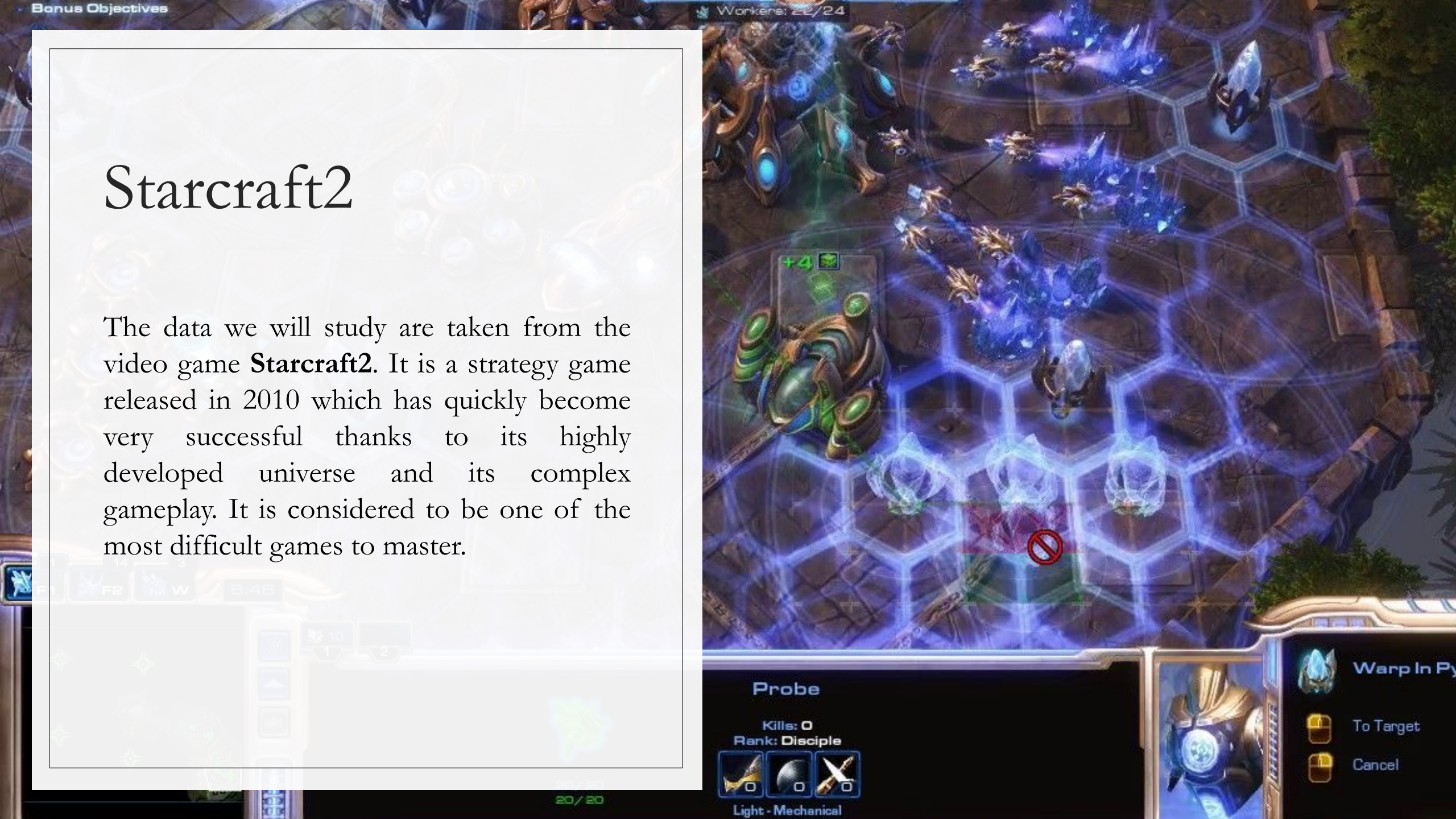
Data to be deleted
Vizualisation with Seaborn
Vizualisation with Pyplot
Vizualisation with Bokeh
Remove the least relevant features
Data scaling and PCA

III. Model predictions

With basic target
With modified target
API with Flask

Starcraft2

The data we will study are taken from the video game **Starcraft2**. It is a strategy game released in 2010 which has quickly become very successful thanks to its highly developed universe and its complex gameplay. It is considered to be one of the most difficult games to master.



The skillcraft1 Master Table dataset

It gathers information about randomly drawn Starcraft2 players.

3395 rows

19 features

1 target

The target : LeagueIndex

Our goal is to create a Machine Learning model that can determine the level of a player (LeagueIndex) based on his in-game activity and various statistics.

Originally, the league index is an integer between 1 and 8. Each number is associated with a league:

- 1 : Bronze
- 2 : Silver
- 3 : Gold
- 4 : Platinium
- 5 : Diamond
- 6 : Master
- 7 : Grand Master
- 8 : Professional

The features

General player information (4) :

- Id of the player : *GameID*
- His *Age*
- His playing time : *HoursPerWeek, TotalHours*

Information on his in-game activity (15) :

- Fluidity : *APM (Action per minute)*
- Use of hotkeys (raccourcis) : *SelectByHotkeys, AssignToHotkeys, UniqueHotkeys*
- Use of minimap : *MinimapAttacks, MinimapRightClicks*
- Reactivity : *NumberOfPACs, GapBetweenPACs, ActionsInPAC, ActionLatency*
- Strategy : *TotalMapExplored, WorkersMade, UniqueUnitsMade, ComplexUnitsMade, ComplexAbilitiesUsed*

Assumptions and preliminary thoughts

We believe that basic information such as **playing time** have a good impact on the player's level. Logically, the more time he spends on the game, the more likely he is to have a good level.

The statistics of his **in-game activity** will be even more crucial in the determination of the rank. We have access to indicators that will allow us to know if he has a good fluidity with the interface (action per minute) and a relevant use of the minimap and hotkeys.

We can also estimate his reactivity through the PACs variables. PAC means **Perception Action Cycles**. This is the time it takes for the player to perceive and process information.

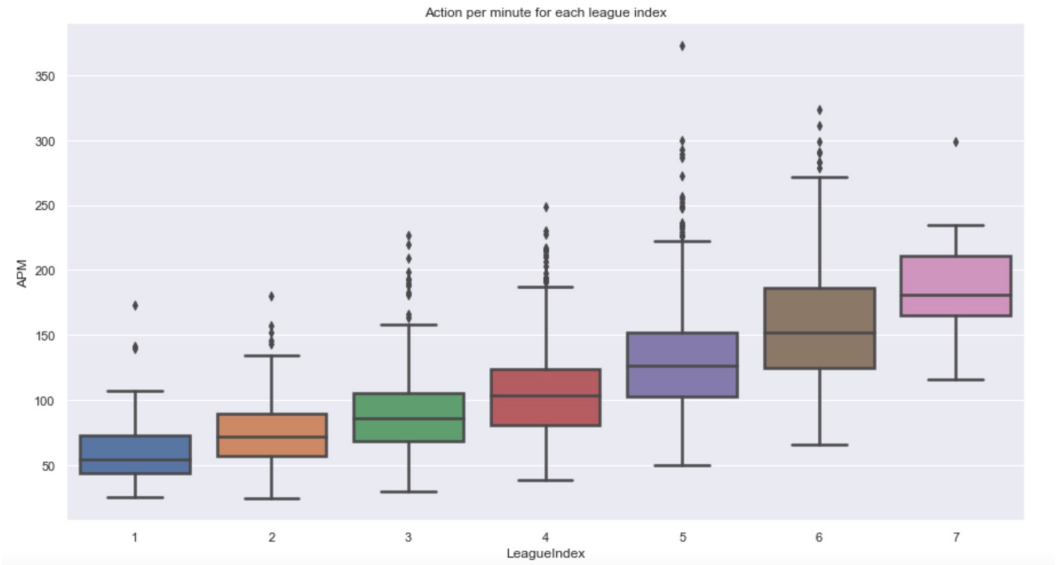
Data to be deleted

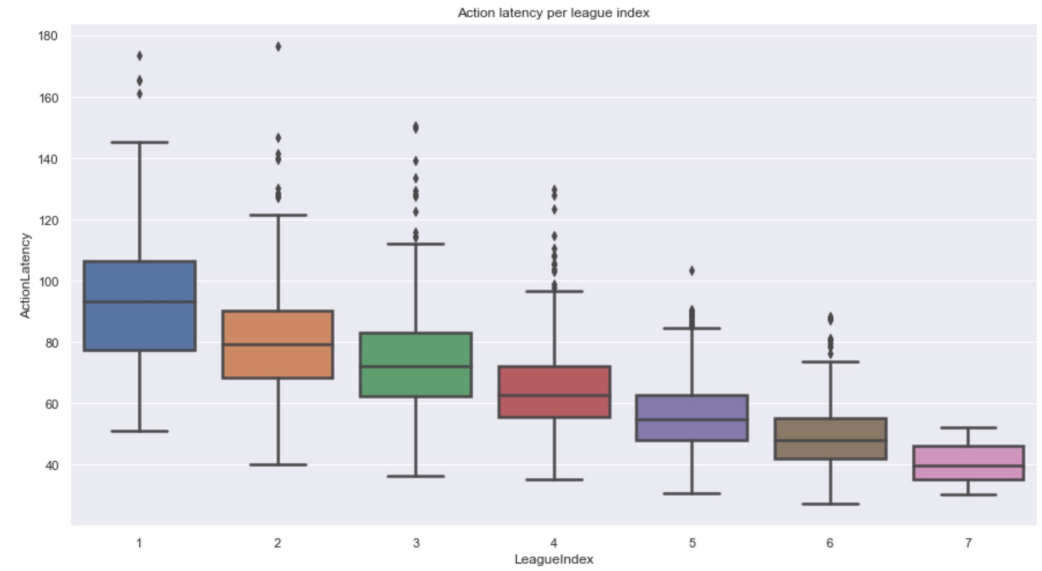
Removal of the league index 8

- We saw that a lot of data was missing for professional players. We will not be able to process so much missing data, we have chosen to delete them.

Outlier detection and removal

- We have spotted the presence of many outliers. To remove them efficiently, we have chosen to import two models allowing to identify them: IsolationForest and LocalOutlierFactor. We then deleted each player that was designated as an outlier by the two models.



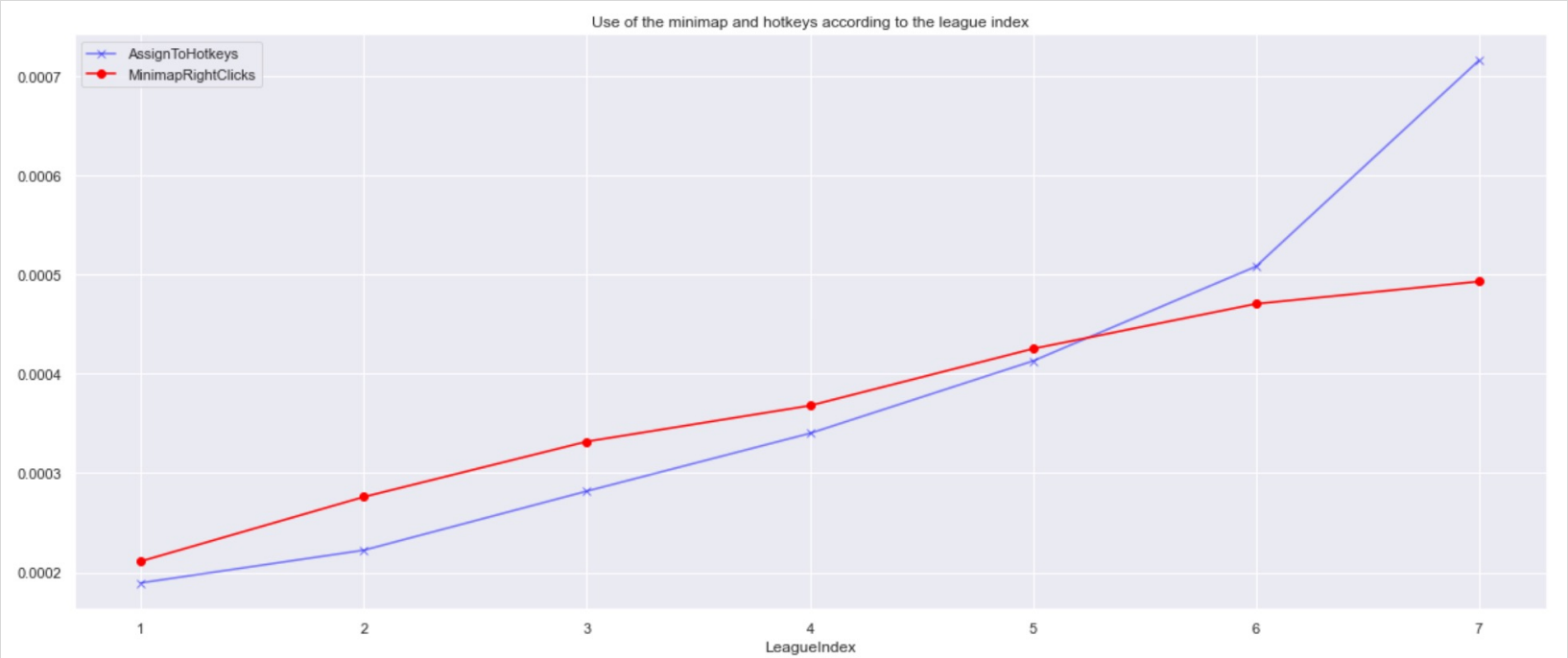


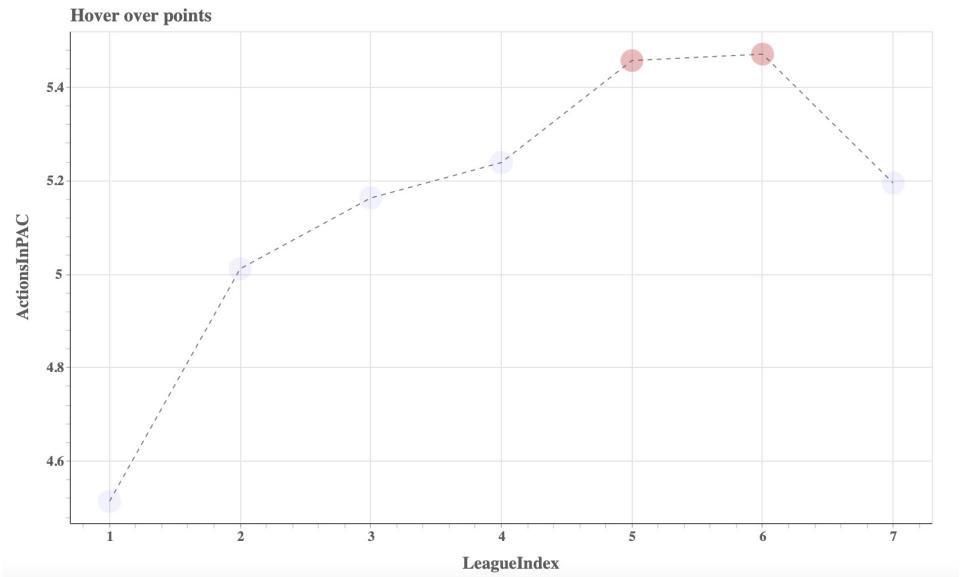
VISUALIZATION WITH SEABORN

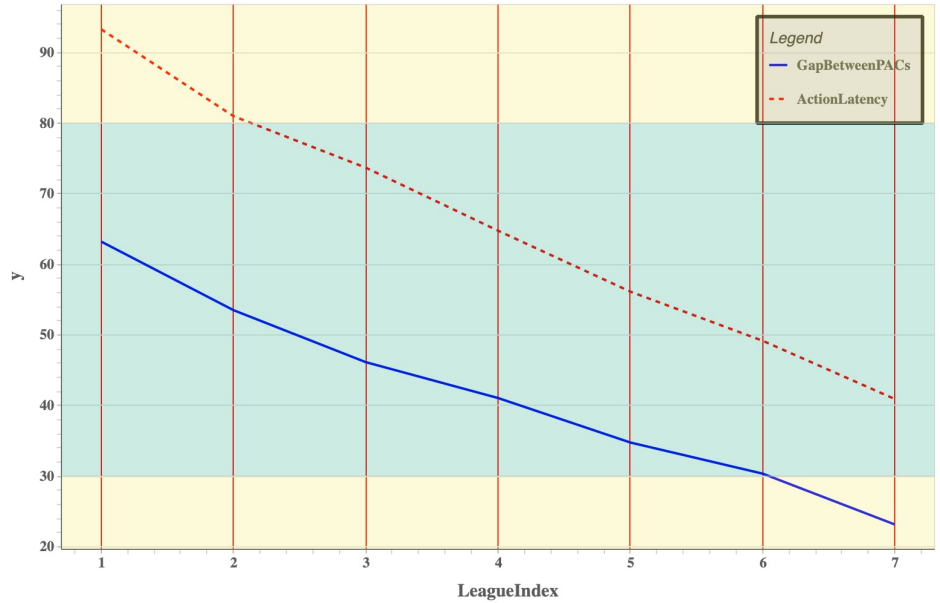
We have made many graphs using Seaborn. We liked it because it allows us to make very nice plots in a few lines and it is easy to use. We could observe the high correlation of some variables with the league index. We can see for example here that the higher the player's rank, the less latency he has in his actions and the more he can perform each minute. Its use also allowed us to observe the strong presence of outliers beforehand.

Visualization with matplotlib.pyplot

We know that the Seaborn library is based on matplotlib and that mastering it is necessary to make more accurate and customized graphics. Still in an effort to understand the data and their correlations with our target, we learned to use matplotlib to plot relevant graphs.

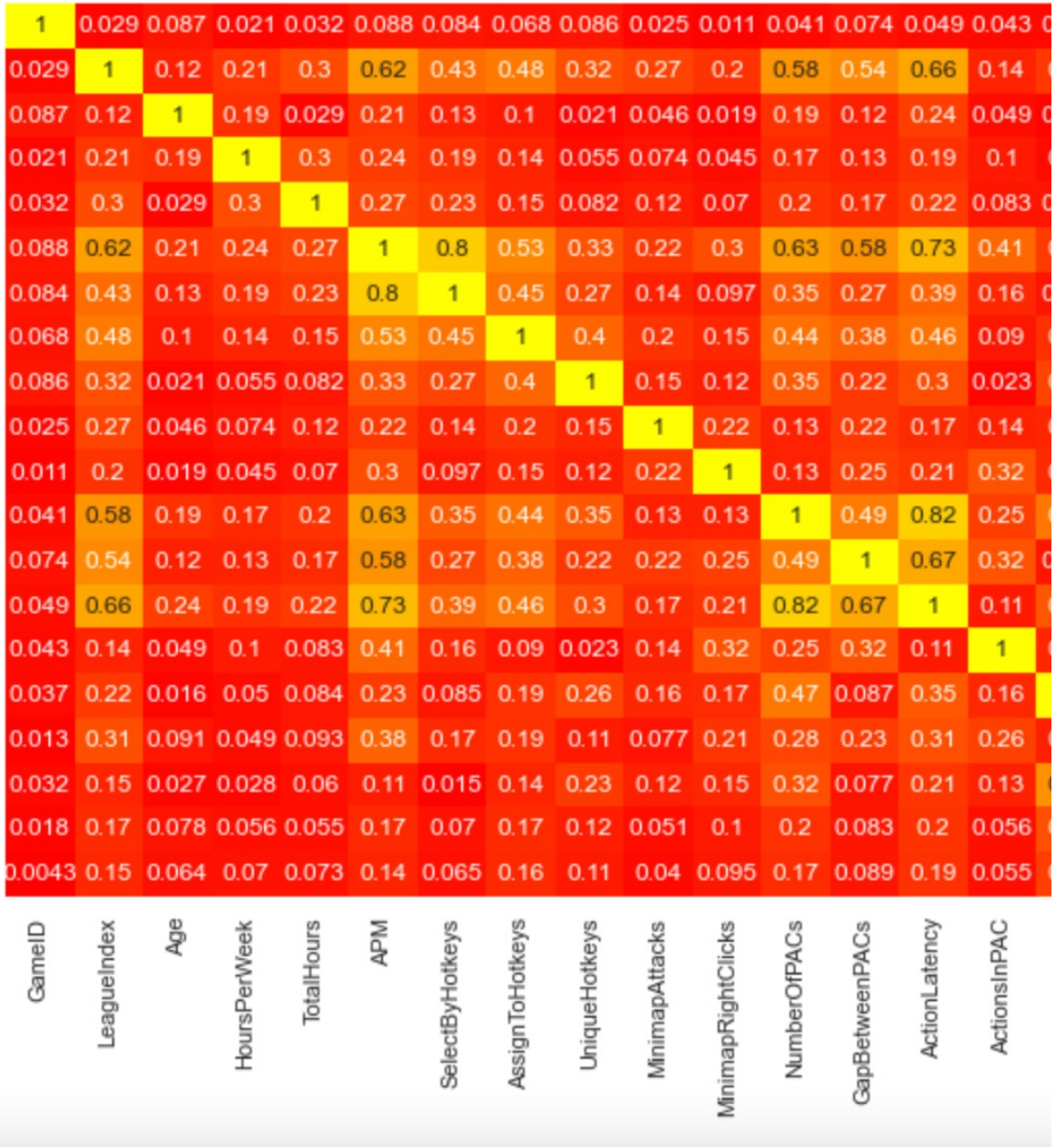






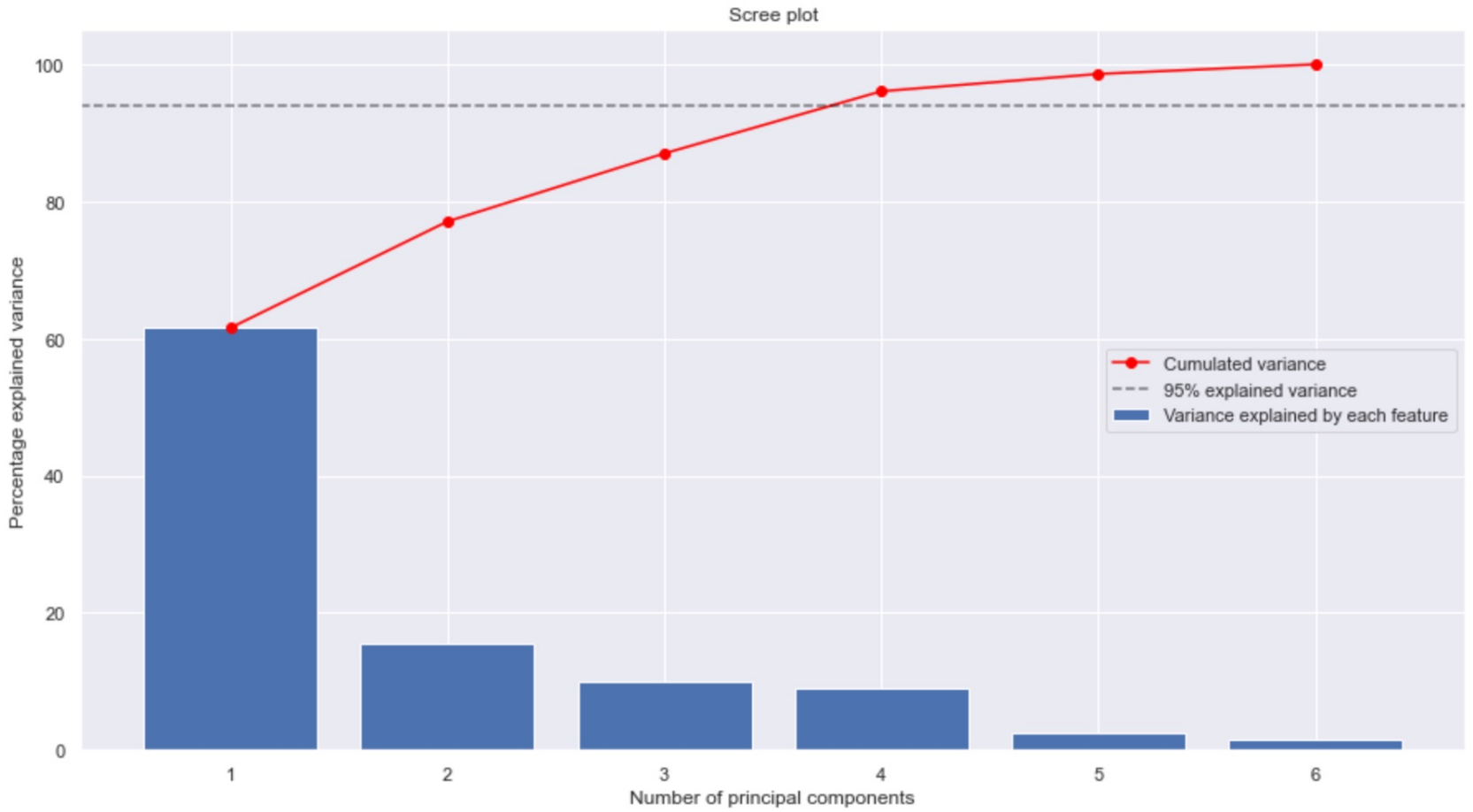
Visualization with Bokeh

Bokeh is a Python data visualization library that provides high performance interactive graphs and plots. It is a tool that we really appreciated. We could see how accessible it was to make powerful interactive graphics. However, for our case, the use of interactive plots is not necessary, that's why the graphs that we could make are not really the most relevant. We will keep this library in mind for our future projects and are happy to have learned how to use it.



Remove the least relevant features

- After studying the correlations of each feature, and after having done performance tests on our models (which we will see later), we decided to keep only the most significant features. You can see attached a correlation matrix in the form of a heatmap. We selected APM, SelectByHotkeys, AssignToHotkeys, NumberOfPACs, GapBetweenPACs and ActionLatency.

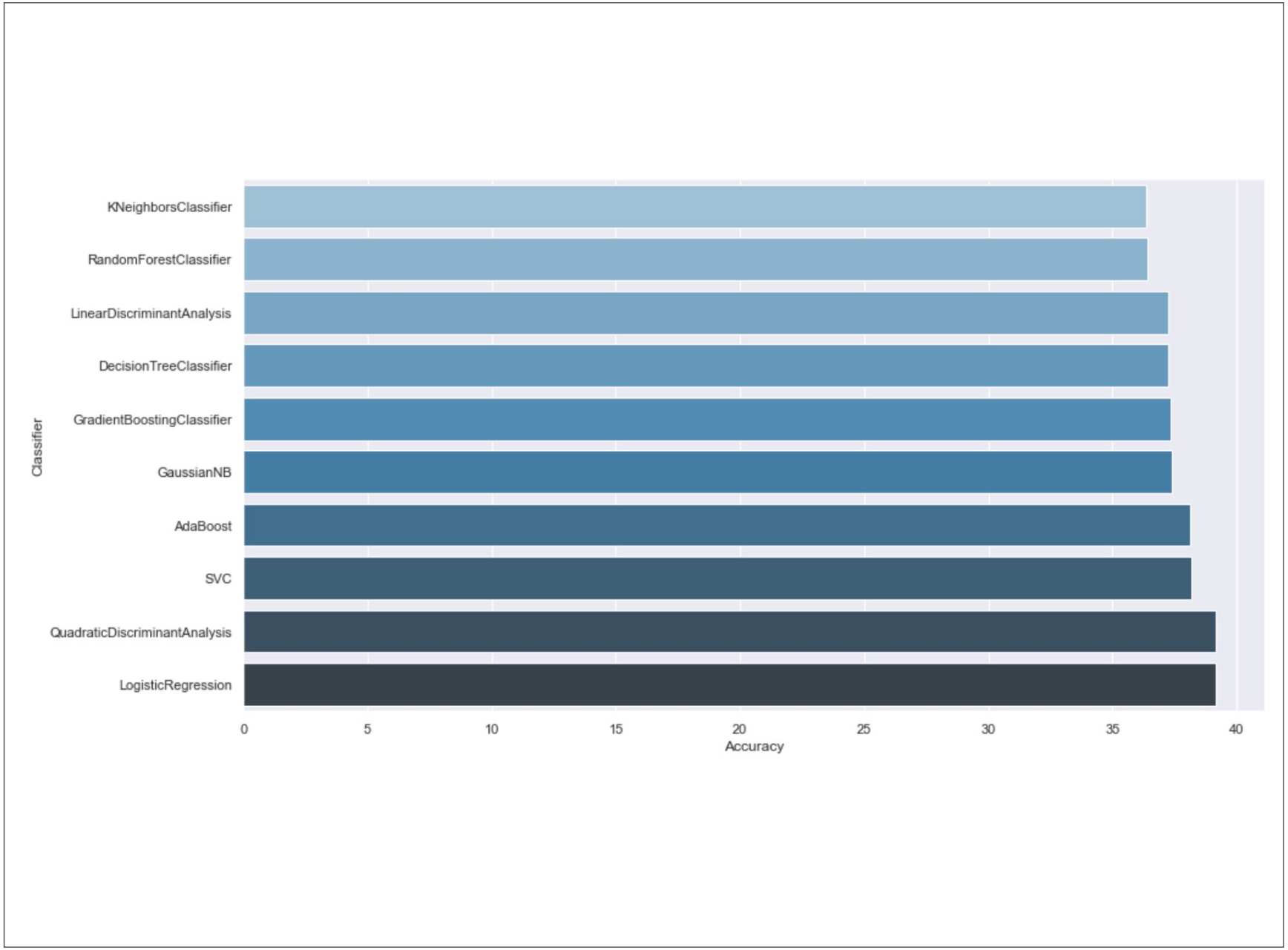


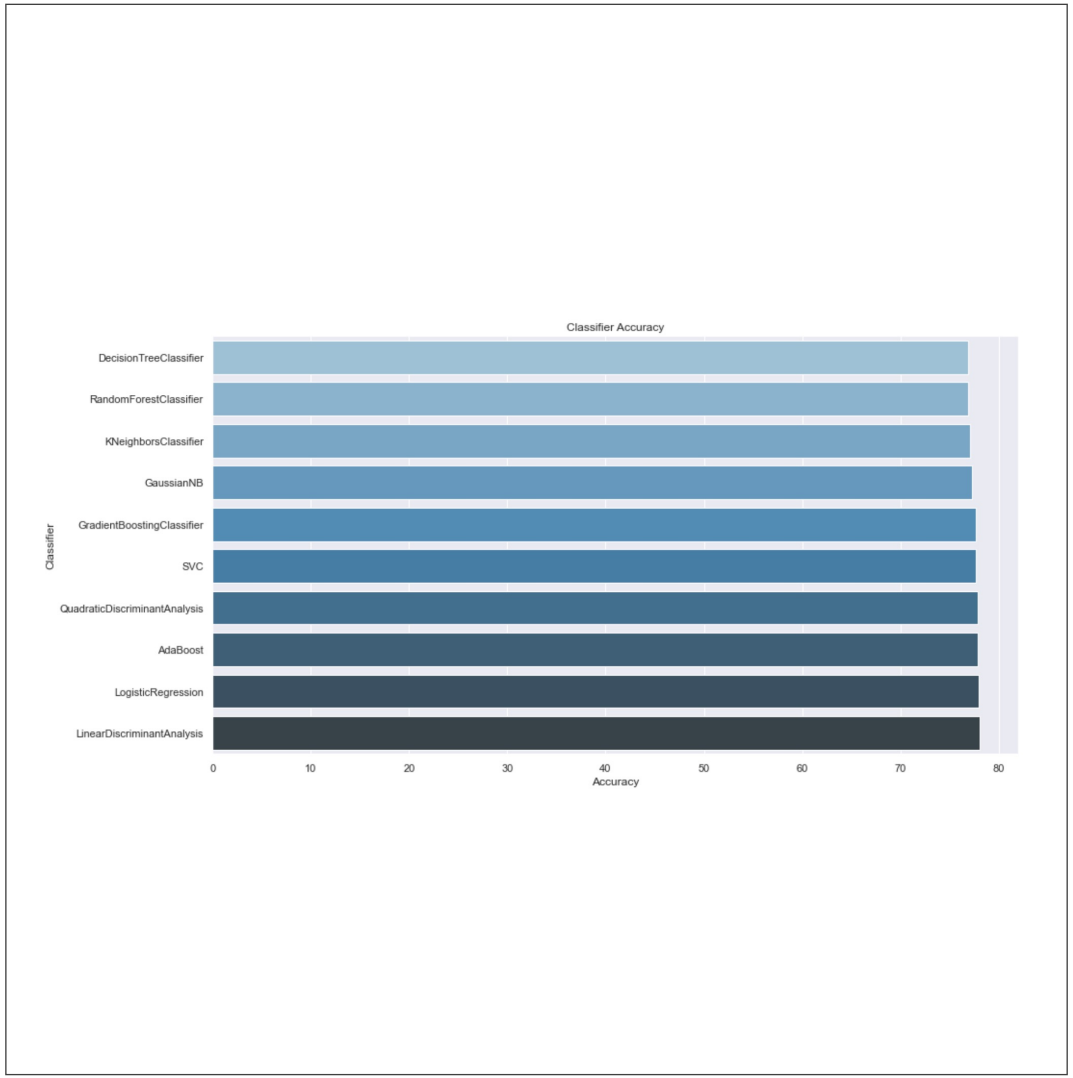
Data scaling and PCA

After that we decided to scale our data. For this, we tried several scalers like the MinMax or the Normalizer, but the one we finally chose is the standard one. Standardizing the features is important when we compare measurements that have different units. Variables that are measured at different scales do not contribute equally to the analysis and might end up creating a bais. We also tried to perform a PCA. The principal component analysis consists in transforming variables into new variables that are decorrelated from each other and carrying more information. Here you can see the scree plot of our PCA.

Model predictions with basic League Index

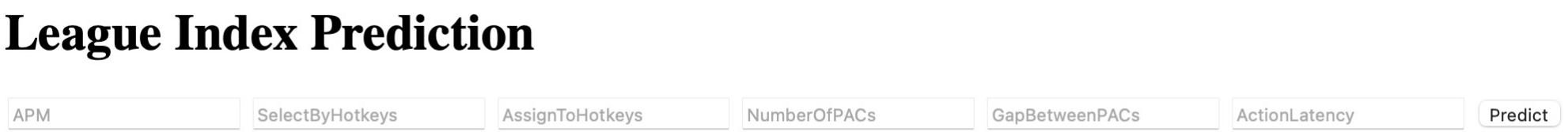
After all this, we set about the prediction by trying many models. We used the GridSearchCV method seen in class to optimize our hyperparameters. The results obtained are not very satisfying because the model with the best accuracy is the Logistic Regression with a score of less than 40%.





Modification of the target

Finally, we have chosen to modify our target to have a binary target with only two classes. Class 0 would be composed of the worst players by combining the Bronze, Silver, Gold and Platinum leagues, and class 1 would be composed of the best players by combining those ranked as Diamond, Master and Grand Master. This makes it much easier to learn the model and get better results. This does not accomplish our original goal of determining the league index, but at least we can roughly predict a player's level with a decent accuracy of almost 80%.



API WITH FLASK

Finally, we have made an API with Flask that allows us to insert a list of values for the features we have kept and it returns a prediction.