PYTHON FOR DATA ANALYSIS

Study on « SkillCraft1 Master Table Dataset Data Set »

How is a player's rank affected?

This study is proposed by the following ESILV students:

CAZAUX Victor

COUTAU François

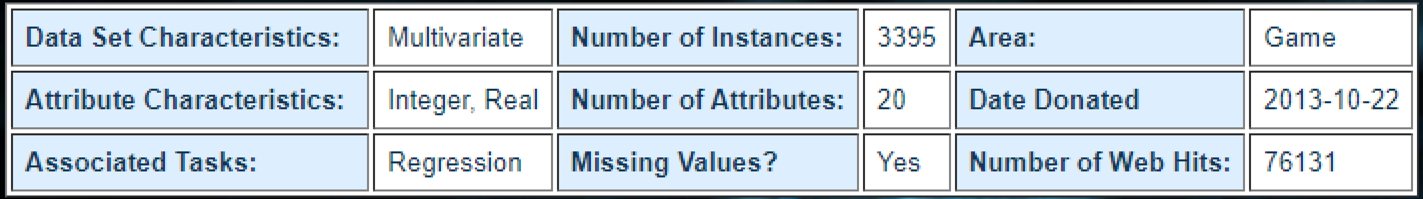
DAHMOUCHE Yanis

StarCraft 2 is a real-time strategy video game developed by Blizzard Entertainment.
Players are required to strategically and quickly manage
their Economy, Technology and Army in order to defeat their opponent. The basis of playing is,
on the one hand, to harvest resources and on the other hand using said resources to
purchase buildings, upgrades and units. (source Liquipedia)

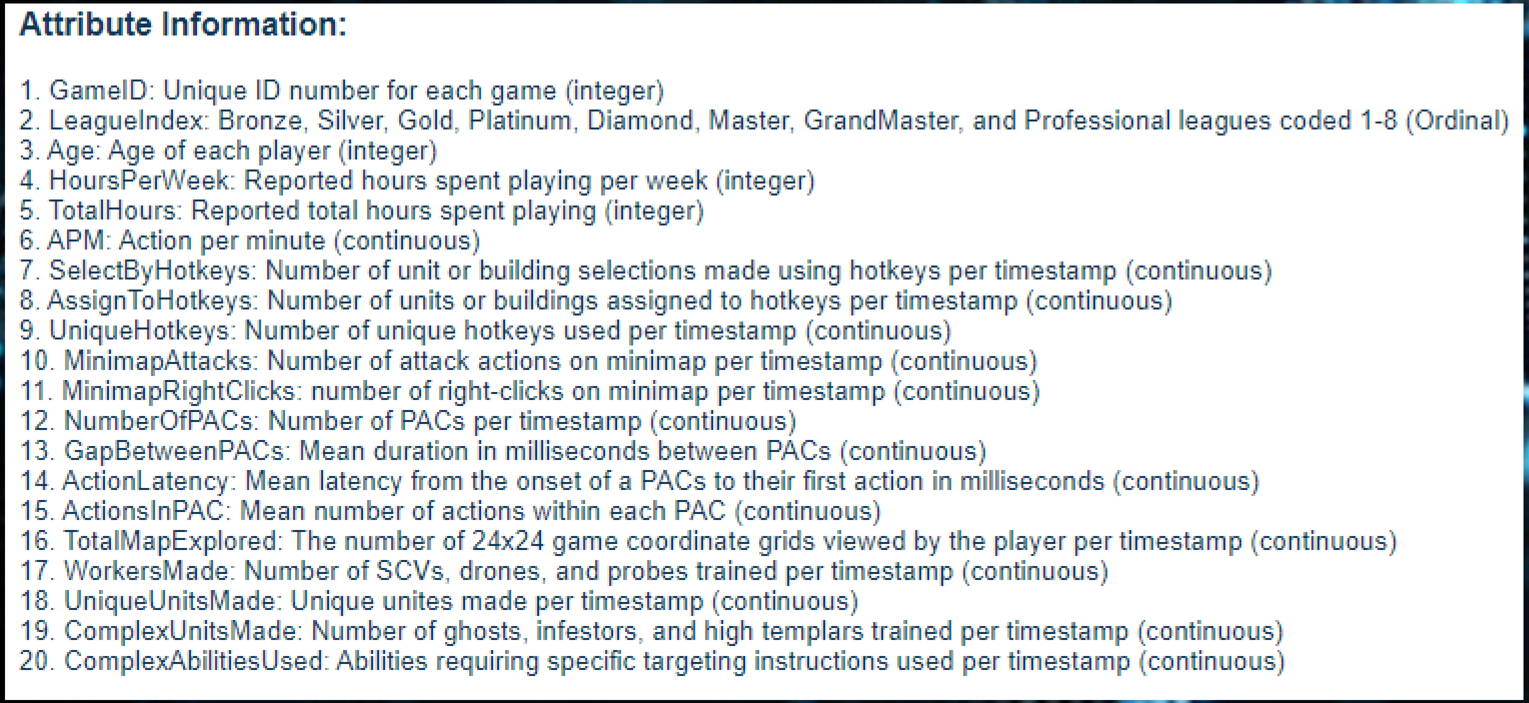
Our goal is to use this dataset to predict with the best possible accuracy the rank of the players.
We have to help us 3395 players and 20 columns of variables that inform us about their
gameplay, experience, and more.

We will first see if the data needs to be cleaned.
Then we will visualize the data to find the links they have with the rank of the players.
Then we will apply machine learning models to predict the rank of the players according to the
data we give them.

Here is the main information from our dataset:



Data Set Characteristics:	Multivariate	**Number of Instances:**	3395	**Area:**	Game
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	20	**Date Donated**	2013-10-22
Associated Tasks:	Regression	**Missing Values?**	Yes	**Number of Web Hits:**	76131



Attribute Information:

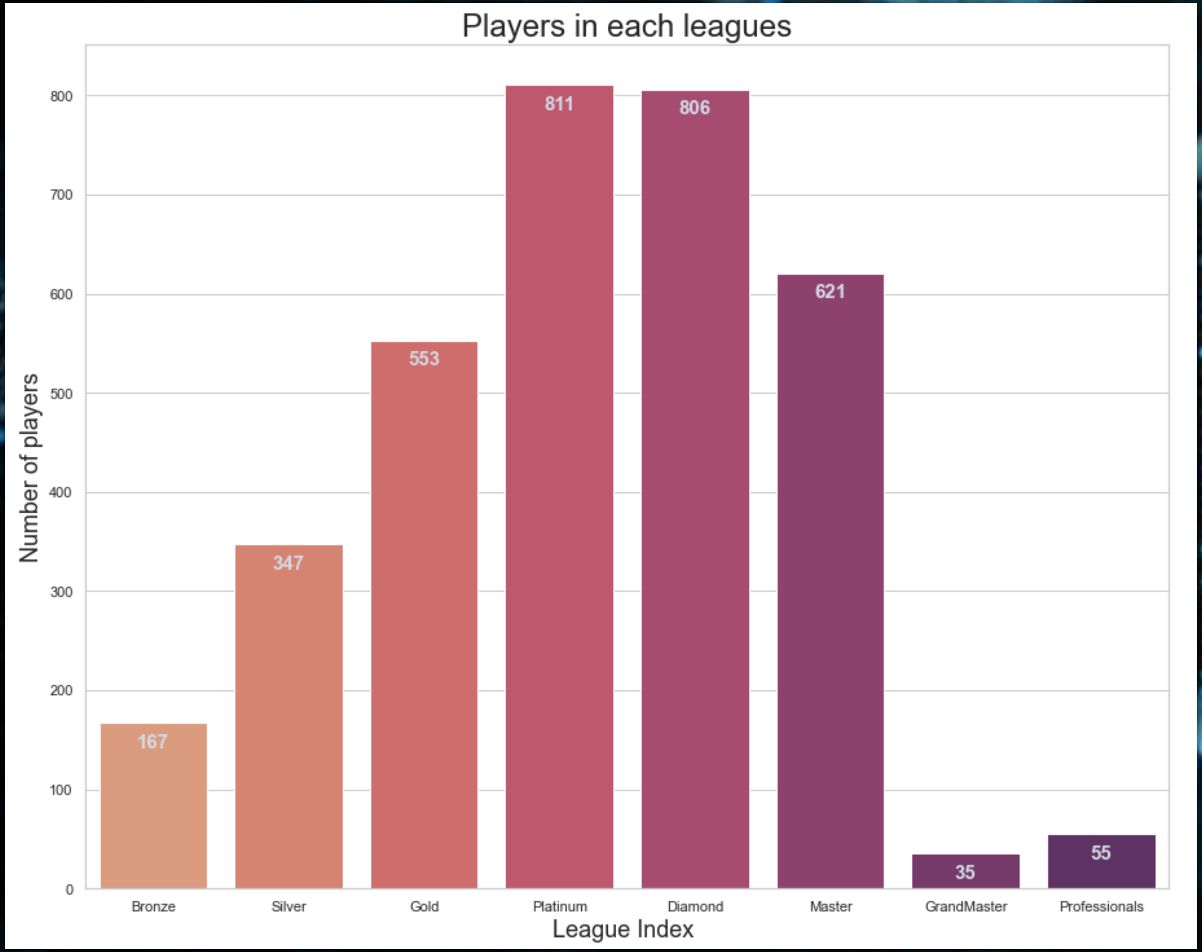
1. GameID: Unique ID number for each game (integer)
2. LeagueIndex: Bronze, Silver, Gold, Platinum, Diamond, Master, GrandMaster, and Professional leagues coded 1-8 (Ordinal)
3. Age: Age of each player (integer)
4. HoursPerWeek: Reported hours spent playing per week (integer)
5. TotalHours: Reported total hours spent playing (integer)
6. APM: Action per minute (continuous)
7. SelectByHotkeys: Number of unit or building selections made using hotkeys per timestamp (continuous)
8. AssignToHotkeys: Number of units or buildings assigned to hotkeys per timestamp (continuous)
9. UniqueHotkeys: Number of unique hotkeys used per timestamp (continuous)
10. MinimapAttacks: Number of attack actions on minimap per timestamp (continuous)
11. MinimapRightClicks: number of right-clicks on minimap per timestamp (continuous)
12. NumberOfPACs: Number of PACs per timestamp (continuous)
13. GapBetweenPACs: Mean duration in milliseconds between PACs (continuous)
14. ActionLatency: Mean latency from the onset of a PACs to their first action in milliseconds (continuous)
15. ActionsInPAC: Mean number of actions within each PAC (continuous)
16. TotalMapExplored: The number of 24x24 game coordinate grids viewed by the player per timestamp (continuous)
17. WorkersMade: Number of SCVs, drones, and probes trained per timestamp (continuous)
18. UniqueUnitsMade: Unique unites made per timestamp (continuous)
19. ComplexUnitsMade: Number of ghosts, infestors, and high templars trained per timestamp (continuous)
20. ComplexAbilitiesUsed: Abilities requiring specific targeting instructions used per timestamp (continuous)

All these columns are directly related to the players' playstyle. They will allow to predict their ranks.

Source : https://archive.ics.uci.edu/ml/datasets/SkillCraft1+Master+Table+Dataset

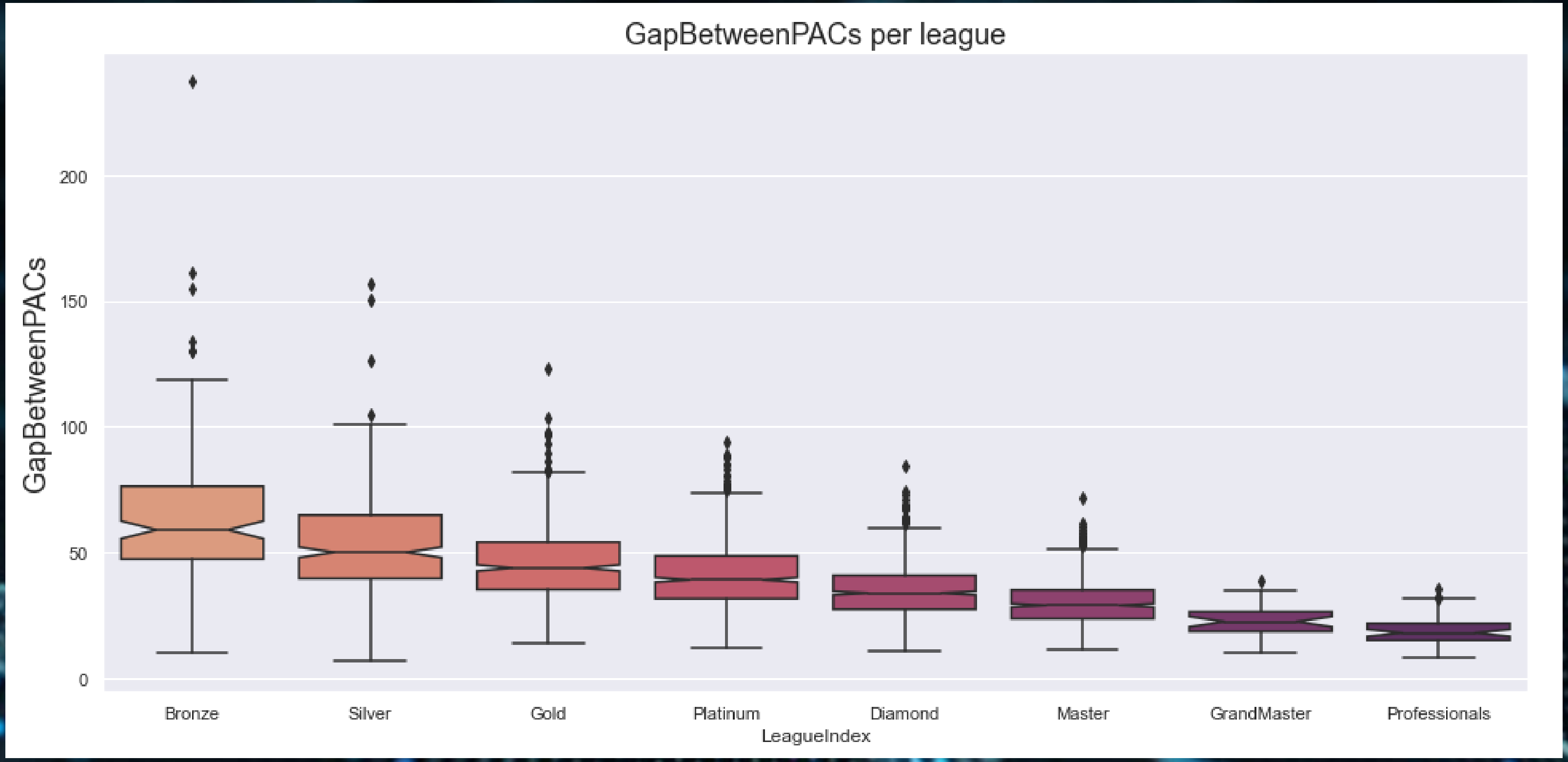
I – Data Cleaning + Exploration

There are about 50 lines with missing values. They were quickly identified and then assigned the column average to keep them. Then, we started to make plots to better understand our dataset.



The ranks go from Bronze, Silver, Gold, Platinum, Diamond, Master, GrandMaster, and Professional. This sample does not represent the reality, there are too many "Master“ (index 6) player, and not enough low level "Bronze" ,"Silver" and “Gold” (index 1-2-3) players.
This is due to the fact that the survey had to be filled out, so players who were more involved in their game were more likely to fill it out.

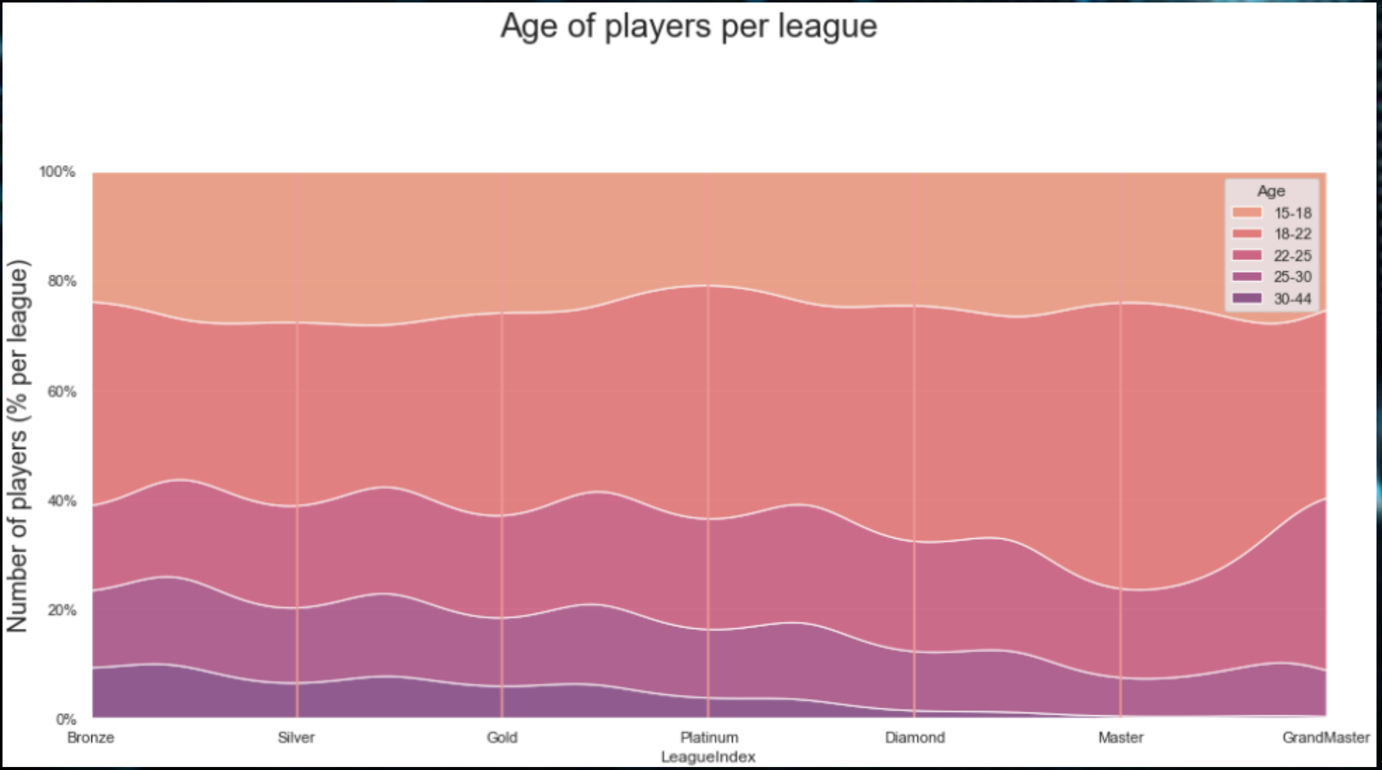
I – Data Cleaning + Exploration

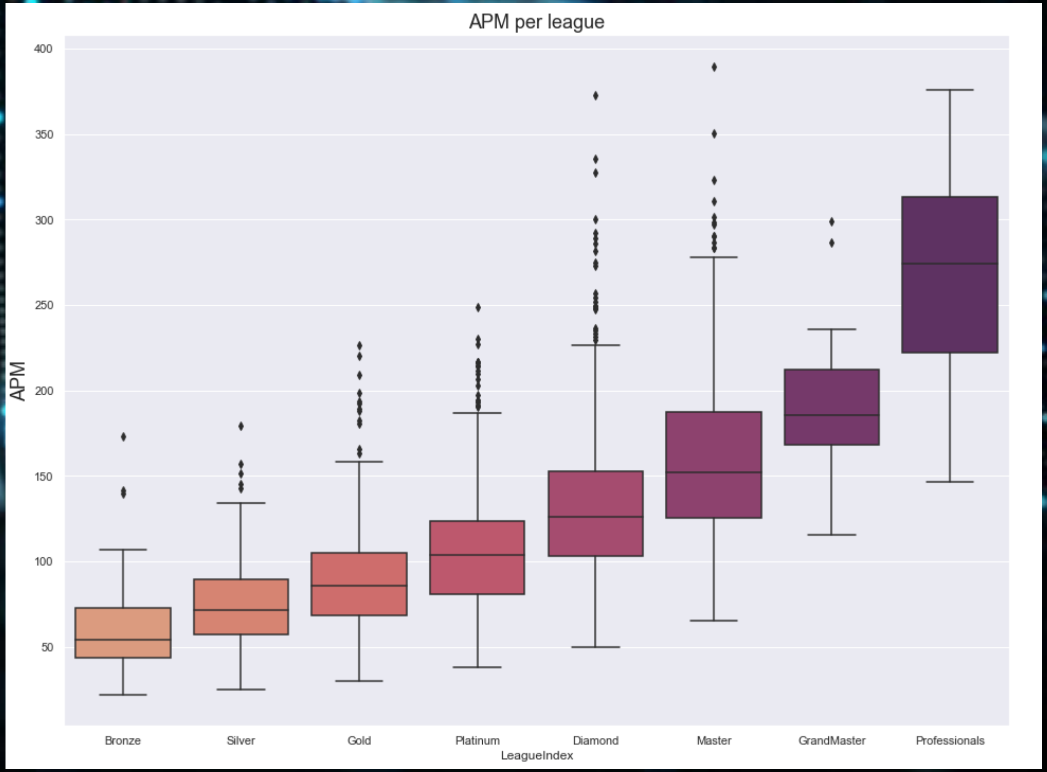


Here is an example of a very useful graph, but this one is negatively correlated.

The more time a player lets pass between his actions, the worse he is.
To be good at Starcraft2, you have to go very fast, constantly think about your next move, and not give your opponent any time to breathe.
The best players are those who optimize their time the most.

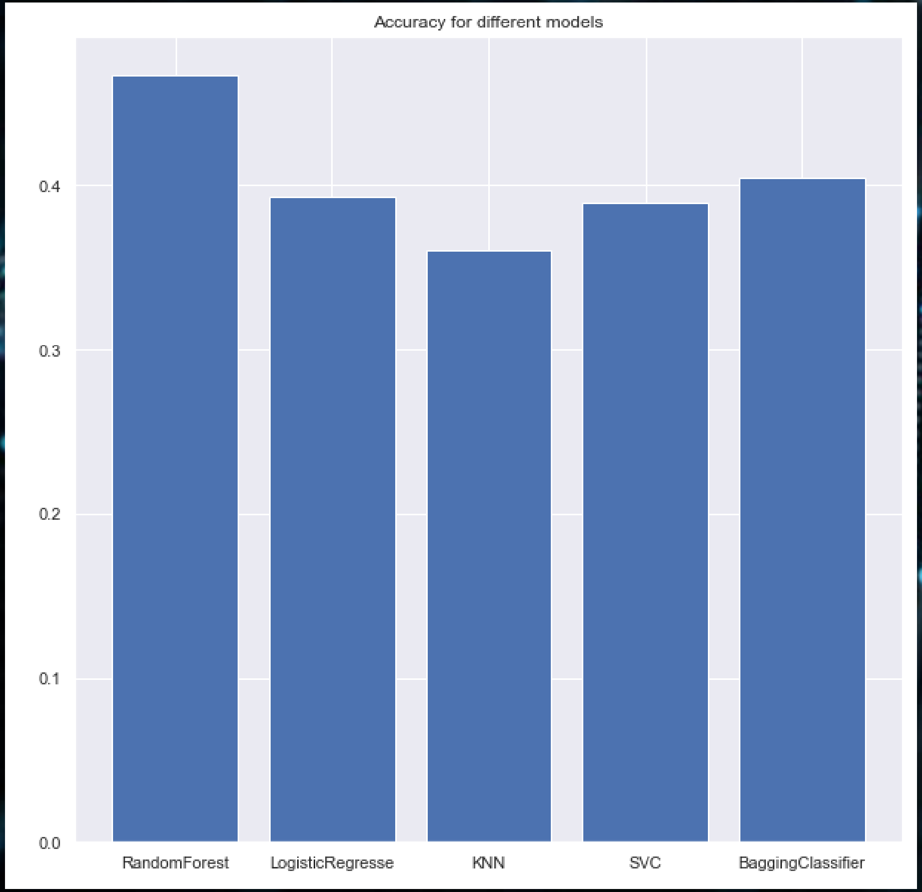
We did a lot of plots to explore all the data as best we could.
We used mostly the boxplots of the seaborn library but also the kdeplot,jointplot and pairplot.
All variables correlate with the rank of the players, so they are probably almost all important for the predictions.
We still identified the most correlated variables and those that are less correlated, in order to be able to adapt our predictions with various data sets.
On the graph on the right, we can see the APMs according to the player's rank, one of the most positively correlated variables in the dataset.





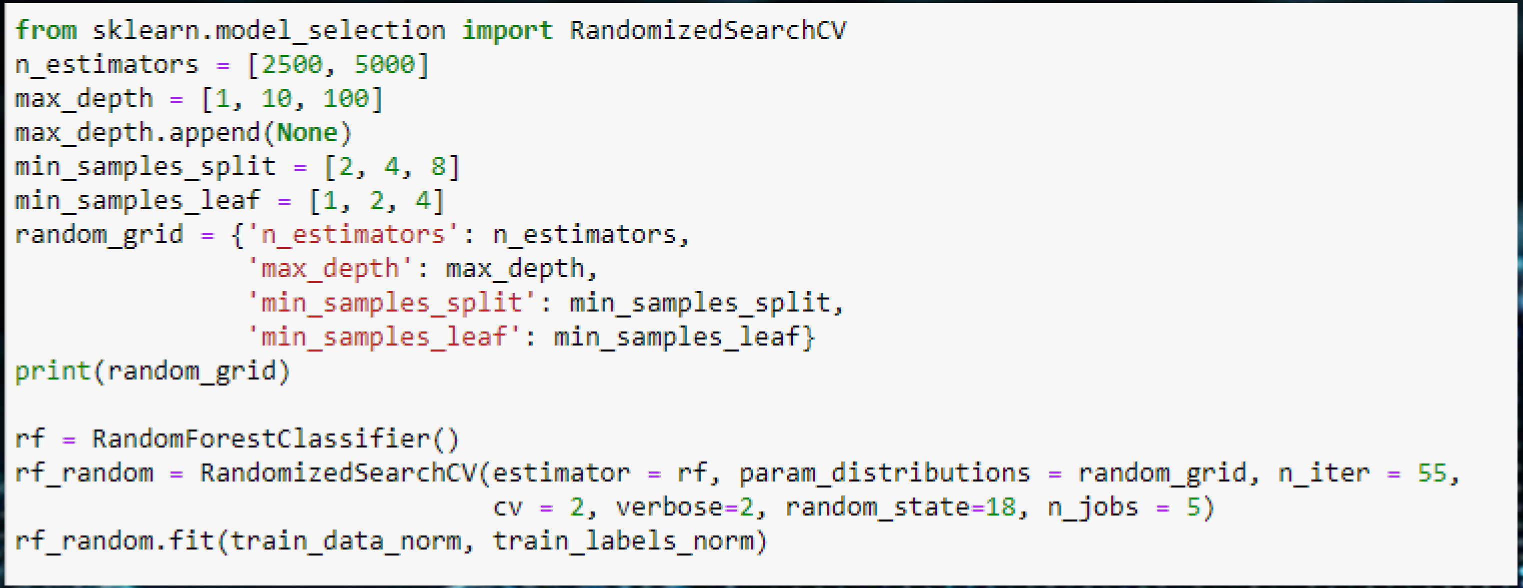
II – Models and predictions

First, we scaled the data.
We then tried the most known models for our problem:
Logistic Regression, KNN, SVC, Bagging classifier, random forest.
We tried all of them with tuning thanks to GridSearchCV function.
This were our accuracy :



Random Forest was the best one, so we decided to work more with it.

So we tried another function to change the hyperparameters of the random forest:

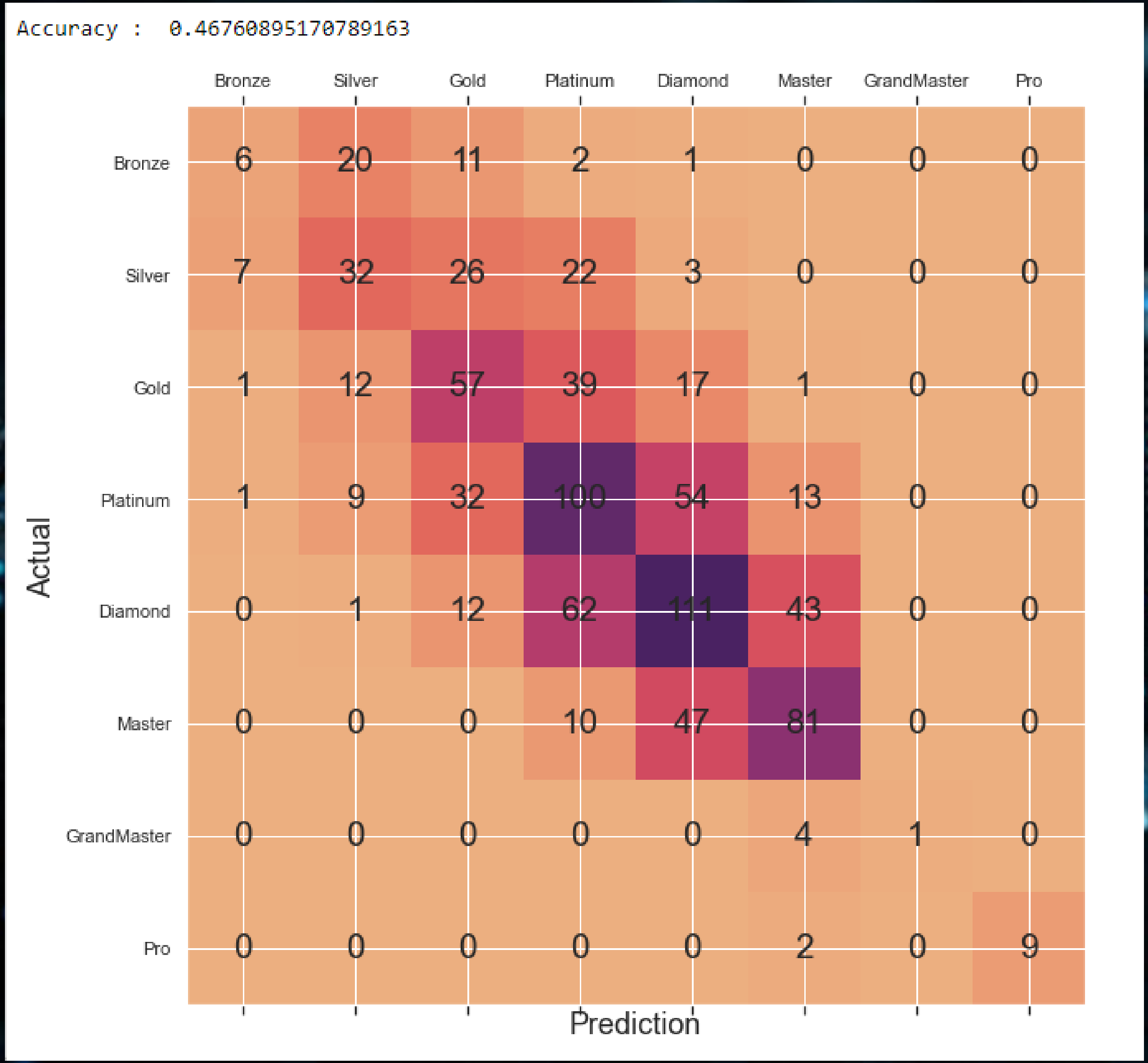


```python
from sklearn.model_selection import RandomizedSearchCV
n_estimators = [2500, 5000]
max_depth = [1, 10, 100]
max_depth.append(None)
min_samples_split = [2, 4, 8]
min_samples_leaf = [1, 2, 4]
random_grid = {'n_estimators': n_estimators,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf}
print(random_grid)

rf = RandomForestClassifier()
rf_random = RandomizedSearchCV(estimator = rf, param_distributions = random_grid, n_iter = 55,
                               cv = 2, verbose=2, random_state=18, n_jobs = 5)
rf_random.fit(train_data_norm, train_labels_norm)
```

Then, we make several tests by removing some columns to keep only the one which are the most correlated with the rank of the players. Without success. Our best result has all columns in incoming data.

Here is our best accuracy and its confusion matrix.



```
Accuracy :  0.46760895170789163
```

Actual \ Prediction	Bronze	Silver	Gold	Platinum	Diamond	Master	GrandMaster	Pro
Bronze	6	20	11	2	1	0	0	0
Silver	7	32	26	22	3	0	0	0
Gold	1	12	57	39	17	1	0	0
Platinum	1	9	32	100	54	13	0	0
Diamond	0	1	12	62	111	43	0	0
Master	0	0	0	10	47	81	0	0
GrandMaster	0	0	0	0	0	4	1	0
Pro	0	0	0	0	0	2	0	9

We therefore have an accuracy of 0.4676 which is kinda good compared to other models.
As you can see on the confusion matrix, it is very hard to determine the rank of a player if he is between bronze and diamond. Professional players are the easiest to predict, because their gameplay is so far above the rest.

We could have put the bronze-silver together and then the professionals and the grandmaster to have a better accuracy. Due to the lack of observations, predicting with better accuracy without changing the data further is complicated.

Conclusion

The study of this dataset allowed us to improve our skills in data visualization and machine learning. It allowed us to find the link between our target output, the different columns available, choose the best models, tune these models...

However, we are still disappointed with our accuracy, we would like to have more. The dataset lacks total observations (4000 players is too few) and maybe other variables would have allowed us to have a better result. But this game is very complex and a player's rank is not based on his APM or shortcuts used.

Finally, working as a team has motivated us to seek the best possible accuracy, we already plan to do machine learning competitions on Kaggle, to improve our skills.