**Python - NBA Shooting Project**

# Objective

Based on a large NBA dataset describing all plays from 2020-2021 season, the main objective was to **predict if a shot will be made** or not based on numerous plays features.

# Scheme

Principal dataset is sent by a private third-party. After describing all features included in the dataset, a **cleaning phase** was needed to eliminate bad quality data and **new independent variables were created** to include them as inputs in ML algorithms. A brief **Exploratory Analysis is conducted** before comparing predictions and accuracies of two different ML supervised binary classification algorithms (**Random Forest, Gradient Boosting**). This projects is clearly a binary classification.

# Data Description

The dataset received contains information regarding all plays from 2020-2021 season (shots, fouls, rebounds, turnovers…). It consists in 541,348 samples (individuals) and 44 features (variables). The object of this research is to focus only on shooting plays (shot is made or not) and features having a potential impact on shooting results. Therefore, after filtering and eliminating bad quality data (missing values mainly), **the final dataset is reduced to 201,524 individuals**.

The **main features gather information on WHO the shooter is** (age, experience, position, points in the game before taking the shot, points in the season before the game), **WHEN the shooter shoots** (quarter, time of the shooting play, elapsed time since the beginning of the game) and **WHERE the shot is taken** (shot distance, shooting areas). The annex A presents features originally present in the dataset and features issued from features engineering (creation of new variables). In total, **the final dataset consists of 62 independent variables and one dependent variable (shot is made or not).**

It is to be noticed that datas regarding defensive plays (distance between defensive player and shooter, who is the direct defensive player, …) are not included which is unfortunate as they are main features to explain the difficulty of a shot taken.

# Exploratory Data Analysis

Graph 1 shows the distribution of shots taken by position. Clearly, we can assess that **players with positions PG and SG are those who shoot the most and by far** (plot bars) despite the fact that they are the **less efficient players** (less than 45% for both categories). In the meantime, it is to be underlined that Small Forwards (SF) are players who clearly shoot less than others with a modest efficiency (around 46%) which is a bit far from other players shooting inside the 3-points line such as Centers and Power Forward (55% of efficiency and 48% respectively). Clearly the position will have an impact in the ML algorithms predictions.

Chart

Description automatically generated

Fig 1 – Distribution of NBA shots by position

On the graph 2-a, we clearly notice that shooting areas have an impact on shots results. As expected, shot **under the circle is the most used and efficient weapon in Basket-Ball**. It is the only area where **efficiency is better than 50%.** Players from all positions have a general interest in shooting from this area as it is well shown by the graph 2-b. Indeed, players from the League made between 12,000 and 6,000 shots from there, depending on the players positions, which is a huge amount in comparison with the second most used area (middle area in the paint or Short Paint Shots on the graph) with a maximum of 4,500 shots taken by Point Guards from there.

In the meantime, 3pts-shots areas such as **3pts top right, top left and facing the circle with a distance less than 27 feet are also important areas for PG and SG** (both positions and all 3 areas combined for almost 24,000 shots taken). But it is also to notice that **Power Forward players, who, in the past, used to work and exist mainly in the paint tend to take shots from longer distance** **than Centers** and have a shooting chart almost equivalent to the one of Small Forward players.

Regarding **2pts shooting or 3pts corners areas outside of the paint, we can conclude that NBA players avoid these areas** considering the small amount of shots made from all positions.

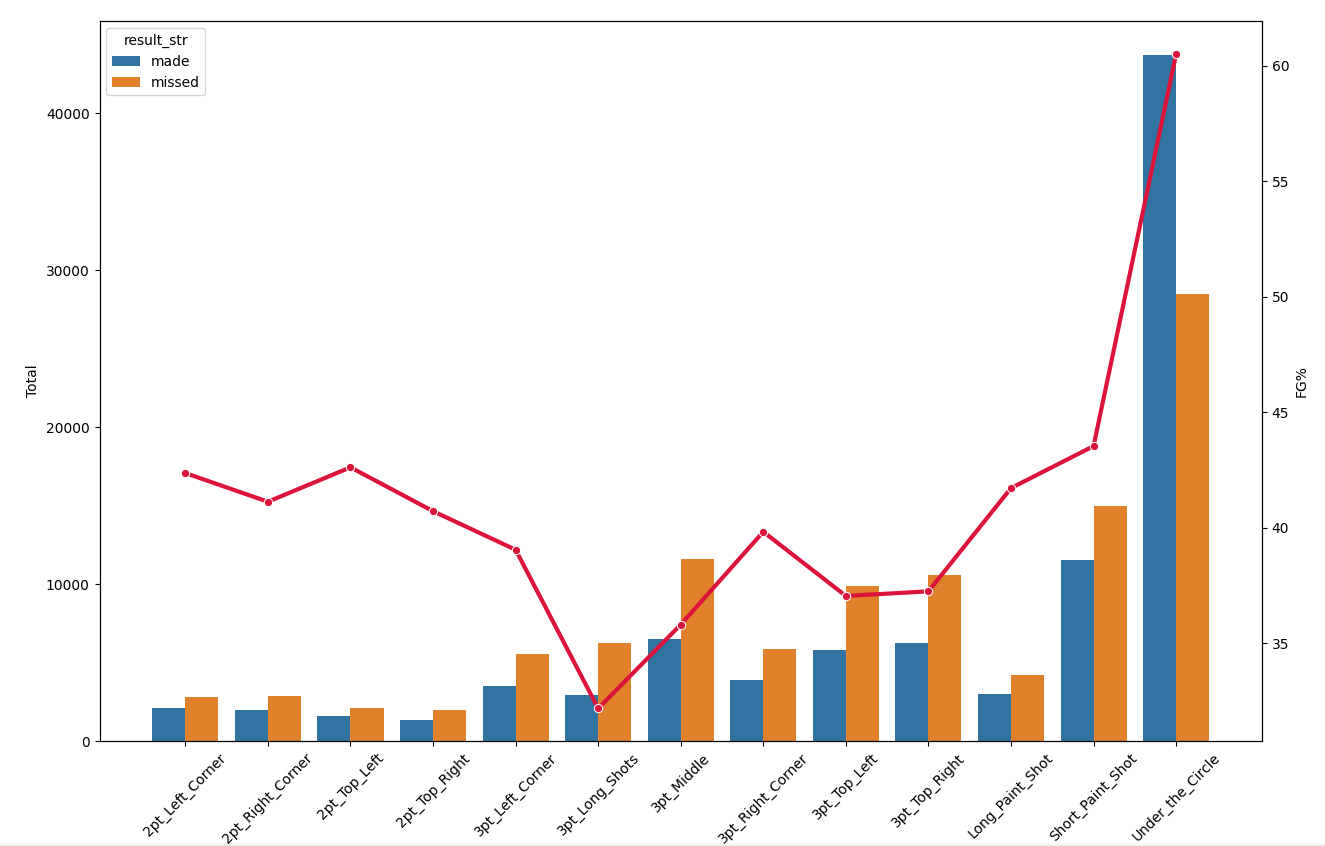


Fig 2-a – Distribution of total shots by shooting areas

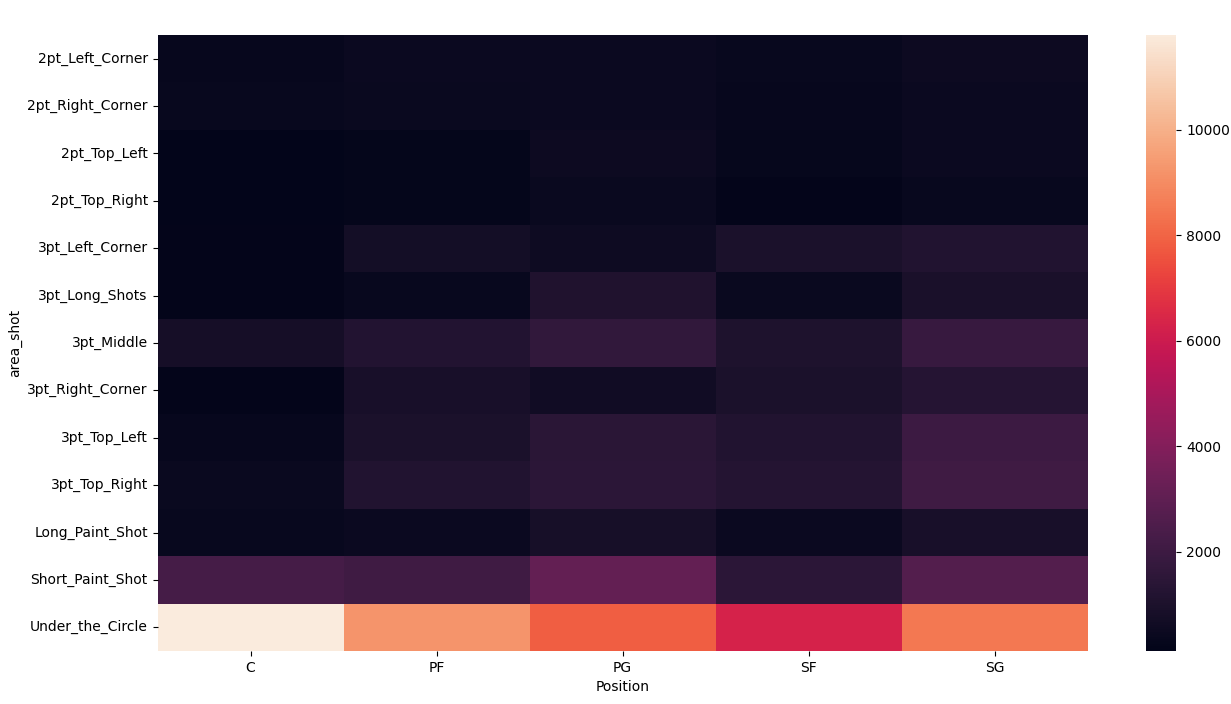


Fig 2-b – Distribution of total shots made by positions & shooting areas

It was also important to analyze if the time elapsed had an impact on the quality of the shots taken by players from the NBA League. The following graph 3 represents number of shots made and missed and the efficiency by **2 minutes intervals**. A similar pattern among quarters was found. **It seems that shooting efficiency increases during a quarter before reaching a peak and starts decreasing until reaching the lowest efficiency point during the last interval of the quarter**. The negative slope of the curve for each quarter might be explained by tiredness and a large number of impossible shots attempted few seconds before the end of the quarter. It is also interesting to note that **the last quarter (from interval [36-38] to [46-48]) is the quarter with the lowest efficiency percentage in weighted average (45.5%).**

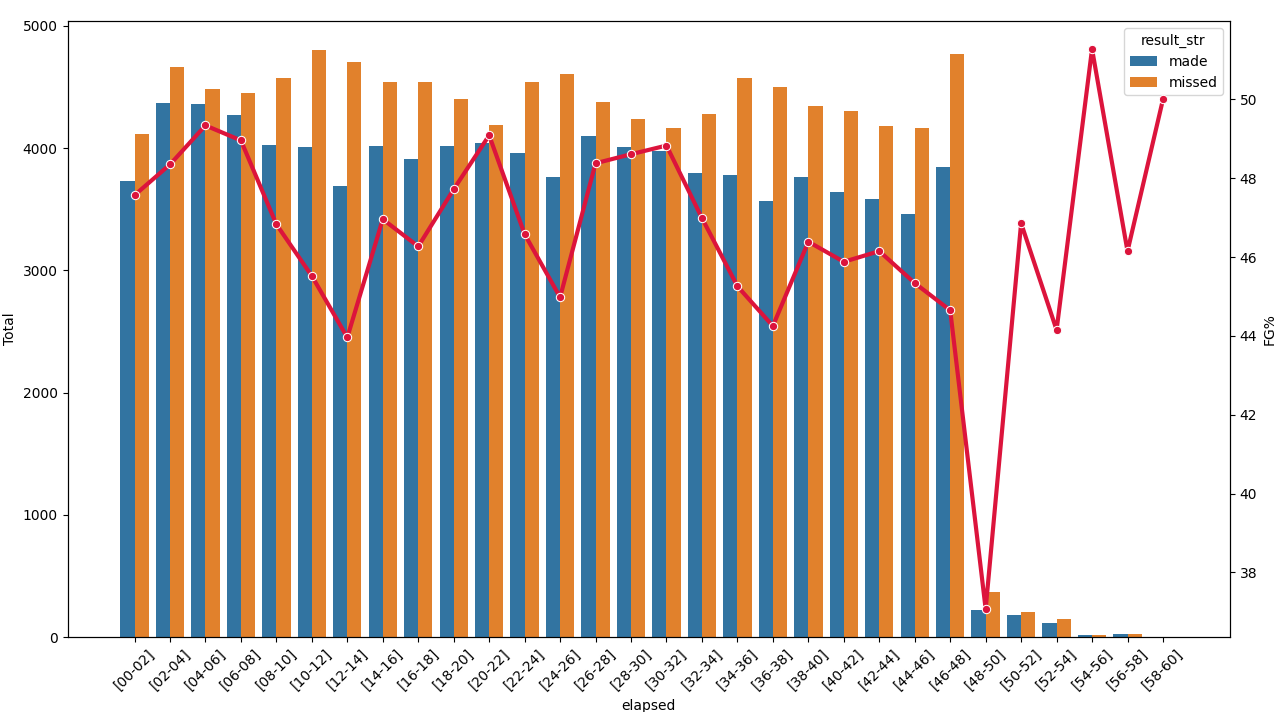


Fig 3 – Distribution of NBA shots by intervals of time

# Machine Learning Predictions

Only two algorithms were considered to predict the result of a shot (binary prediction). Considering the number of features who is really high (62 including dummy variables), a **random forest** was chosen to reduce the variance by training on different samples of the data (bagging method). This algorithm is compared with the **XG Boosting algorithm**. Unlike the random forest taking into account decisions trees (estimators) independently from others, XG Boosting is still based on decisions trees but each estimator will be trained by learning from the predictions errors from the previous one. Therefore, it was interesting to compare those two methods. For each algorithms, optimizations of hyperparameters were done but results were almost the same that those using hyperparameters set by default. Optimization processes are accessible within Python code.

To compare those algorithms, we use **confusion matrixes** classifying the actual and predicted values as ‘made’ or ‘missed’ shots. Results are shown in the below matrixes.

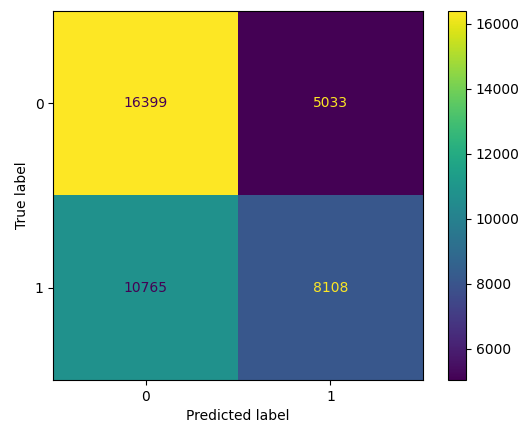


Fig 4-a – Confusion matrix of a Random Forest

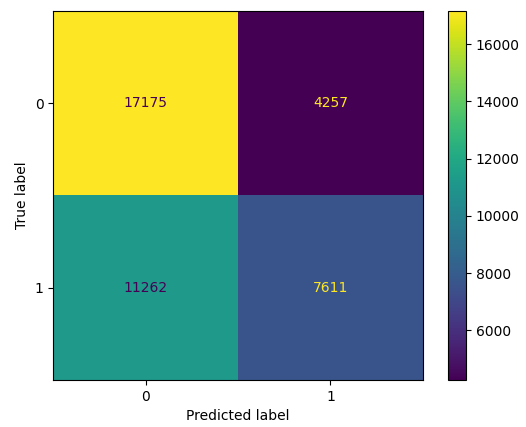


Fig 4-b – Confusion matrix of a XG Boost

|  |  |  |
| --- | --- | --- |
| **MISSED SHOTS** | Random Forest | XG Boost |
| Recall (TP / (TP + FN)) | 0.77 | 0.8 |
| Precision (TP / (TP + FP)) | 0.6 | 0.6 |
| F1 (2\*Recall\*Precision/(Recall+Precision) | 0.67 | 0.69 |
| **MADE SHOTS** | Random Forest | XG Boost |
| Recall (TP / (TP + FN)) | 0.43 | 0.4 |
| Precision (TP / (TP + FP)) | 0.62 | 0.64 |
| F1 (2\*Recall\*Precision/(Recall+Precision) | 0.51 | 0.5 |

Fig 5 – Comparisons of predictive scoring

# Conclusion

In addition to the above grid results, **the accuracy scores are 60.8% for random forest and 61.2% for XG Boost making them relatively equivalent regarding the results**. Missed and made shots are relatively well balanced in the outcome tested (y\_test in the code) and it seems that **models have a more important issue regarding the classification of truly made shots while it classifies missed shots with much more accuracy (see fig 5).**

Reducing the number of measures using a dimension reduction analysis (PCA) might be a good project to test the algorithms with less complex dataset. Moreover, some really important variables that could explain results of shots were not included in this project as they were missing within original dataset. But including features regarding the shooting defense (distance between shooter and defensive player, shooter is taller or smaller than defensive player, …) appear to be primordial to build a more stable and reliable algorithm…