U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Data Mining Project

G44
Henrique Silva - up202105647 - 33.33%
Rita Leite - up202105309 - 33.33%
Tiago Azevedo - up202108699 - 33.33%

Business Goals

Theme

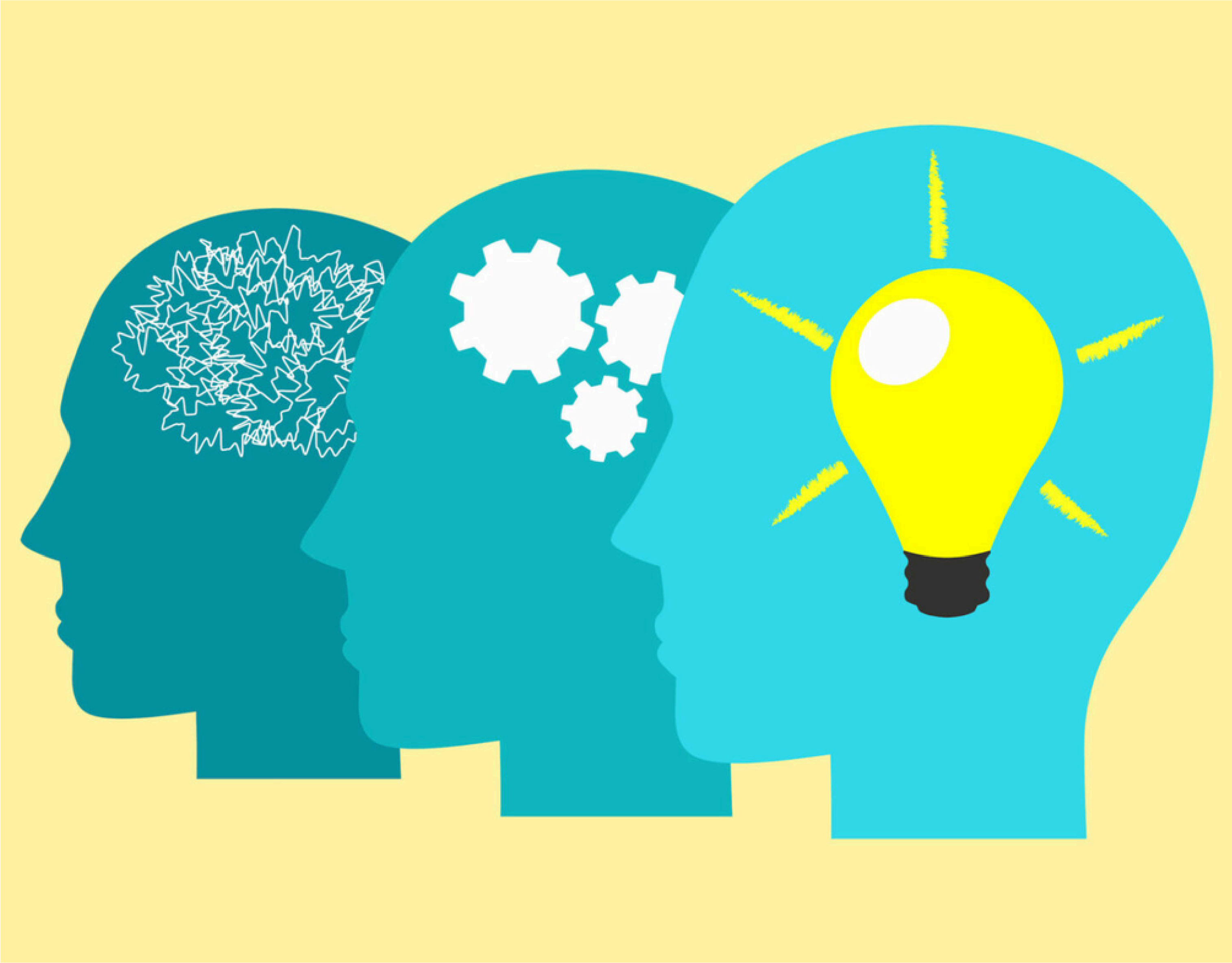
The theme is the WNBA.

Goal

Predict which teams will advance in the playoffs for a given year.

Context

- Each team is associated with one of two conferences: the Western Conference or the Eastern Conference.
- In the **first part** of the season, teams play against each other. In the **second part**, the top four teams from each conference advance to the playoffs.
- The playoffs consist of three rounds: the quarterfinals, semifinals, and finals, in that order. In each playoff round, the team that loses a game is eliminated and does not advance to the next stage.
- One of the datasets provided includes information from 10 consecutive years of the WNBA league, and the goal is to predict which teams will advance in the 11th year.

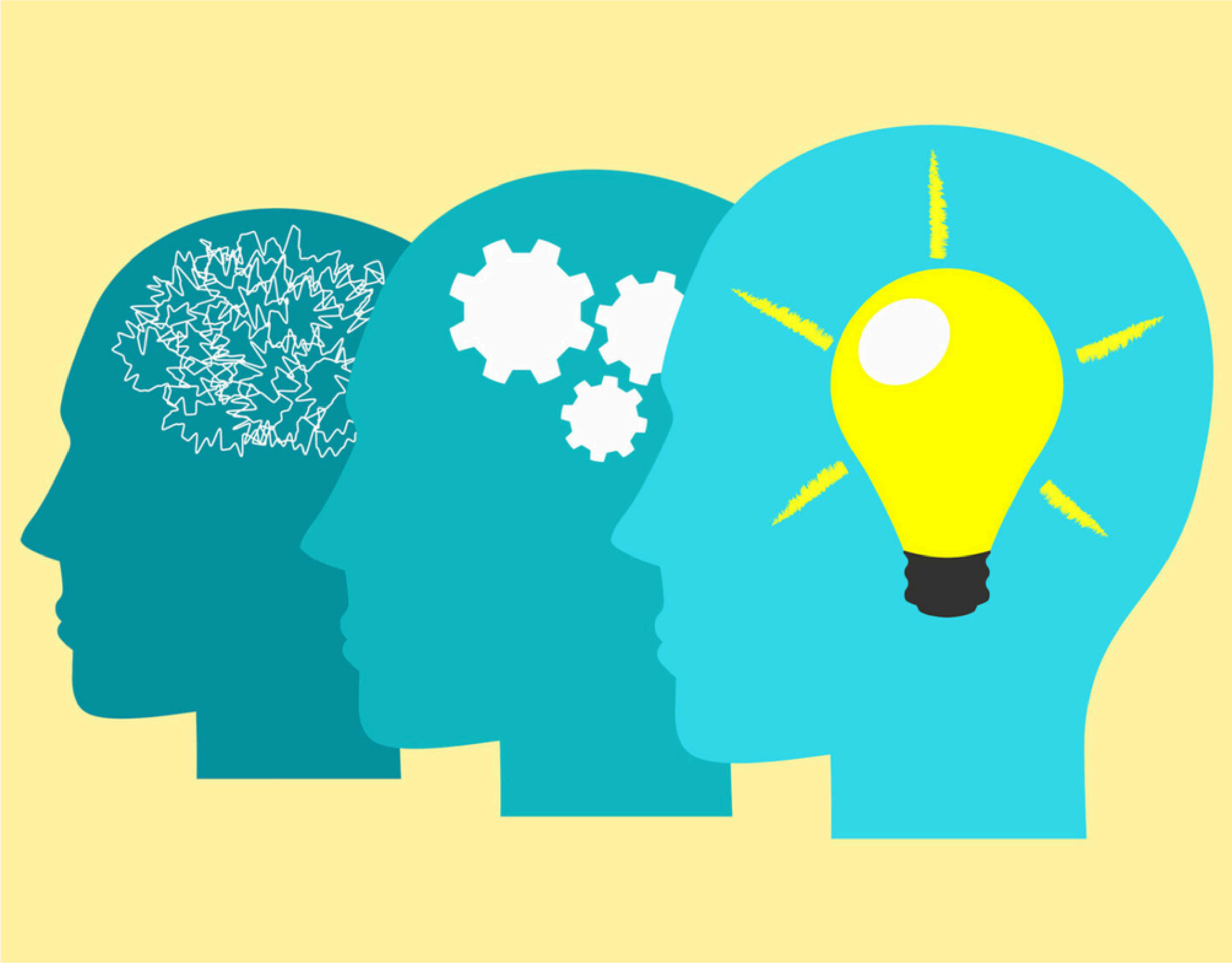


Data Mining Goals

Goal

Predict which teams will advance in the playoffs for a given year.

- It can be defined as a classification problem, with the target variable being the playoff ("yes" or "no"). The positive class is "yes" since it is the one we are most interested in.
- We define the model as successful if it achieves a **precision greater than 75%**. In other words, precision is calculated as the number of true positives divided by the total number of predicted positives. The goal is to only make mistakes in predicting 2 teams as playoff contenders, since 6 out of 8 equals 0.75.



Domain Description

- “**players**” - personal information of the players.
  - There are 893 players.
- “**players_team**” - statistics of each player in a given year.
  - There are 1876 entries of (player, team, year).
- “**coaches**” - team that each coach coached in a given year, as well as their statistics from that same season
  - There are 57 coaches.
  - There are 162 entries of (coach, team, year).
- “**teams**” - statistics of each team in a given year, as well as personal information such as name, conference, among others
  - There are 18 teams.
  - There are 142 entries of (team, year).
- “**series_post**” - information about the games that took place in the playoffs, for each year.
  - There are 70 playoff games.
- “**awards**” - information about who won a particular award in a particular year.
  - There are 12 awards.
  - There are 96 entries of (player / coach, award, year).

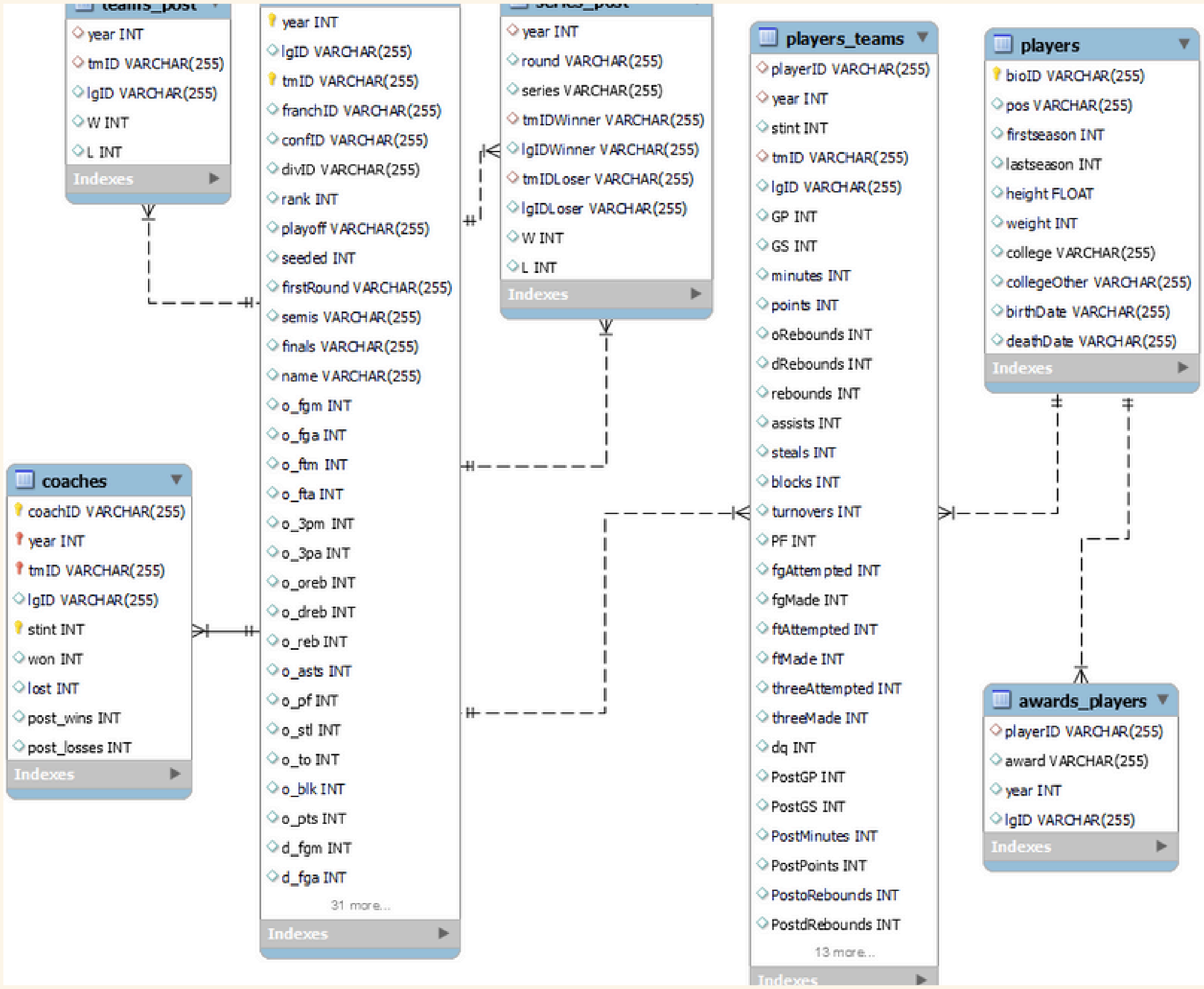


Figure 1 - Domain Model

Exploratory Data Analysis

Figure 2 - The teams that make it to the playoffs more often are also those with the highest average number of wins. The opposite is also true, the teams with the most losses and the fewest wins are also those that have made it to the playoffs the least number of times.

There seems to be a relationship.

Figure 3 - The variation of the number of games won by each team, by year, seems to not change that much.

There seems to be a pattern.

Figure 4 - Neither the teams with the lower reaking have higher assist numbers, nor the teams with the highest have lower attendance levels. It is important to notice that the higher the ranking, the worst are the team's results.

There doesn't seem to be a relationship.

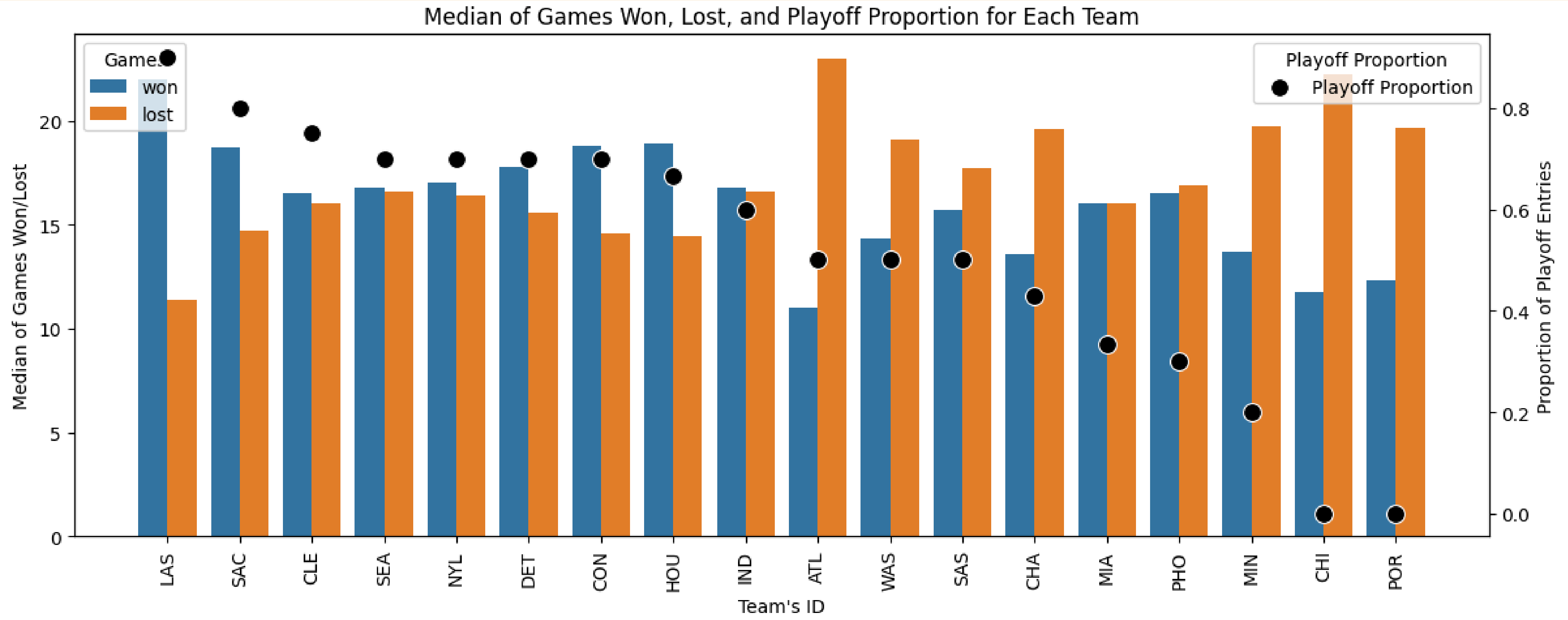


Figure 2 - Median of games won, lost, and playoff's proporcion for each team

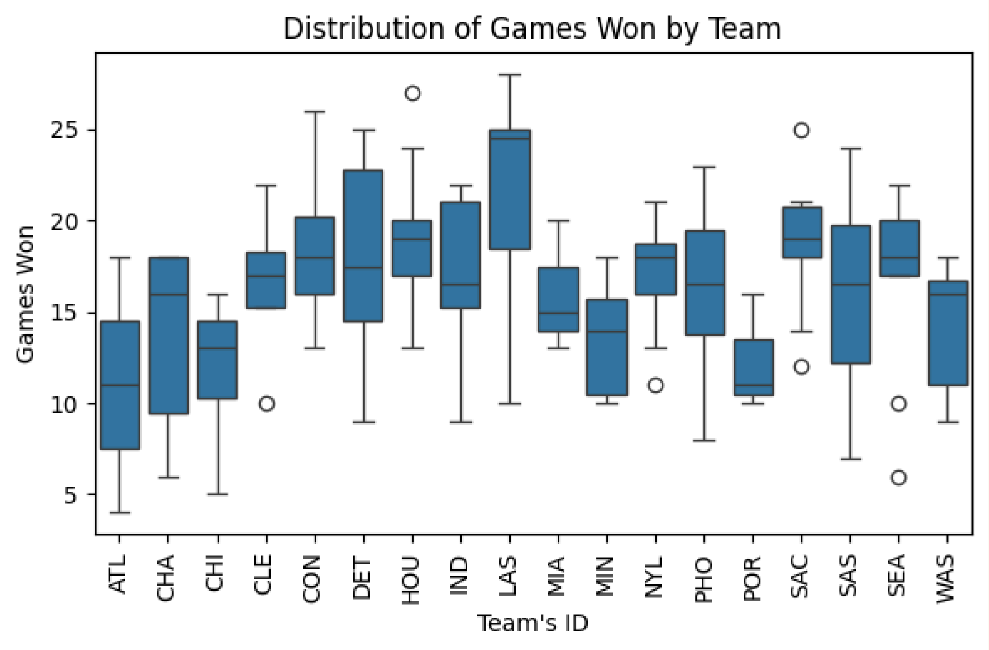


Figure 3 - Distribuition of games won by team

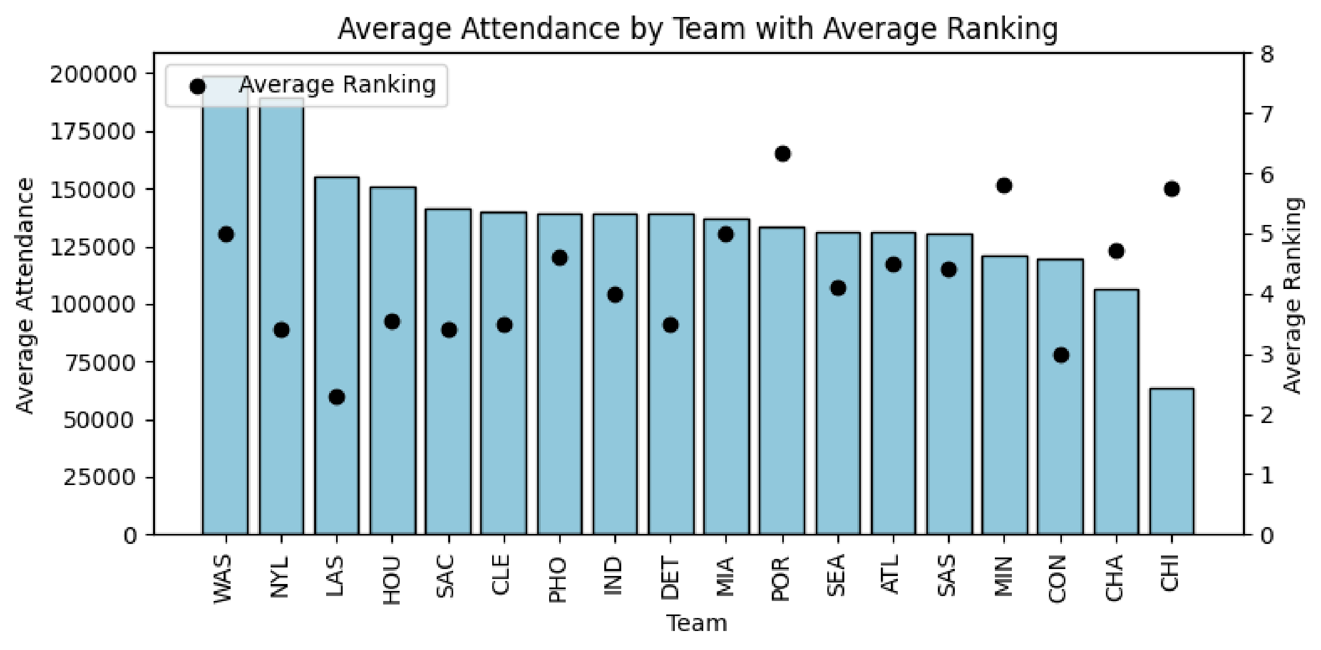


Figure 4 - Average attendance by team with average ranking

Exploratory Data Analysis

Figure 5 - The variation of the ranking of each team seems to be small in most of the cases. In most cases, the ranking variation appears to be progressive, with few cases in which the team's ranking varies by more than 3 values from one year to the next.

There appears to be a consistent pattern.

Figure 6 - The variation of the presence in the playoffs of each team seems to be small in most of the cases. The execeptions are WAS and NYL.

There appears to be a consistent pattern.

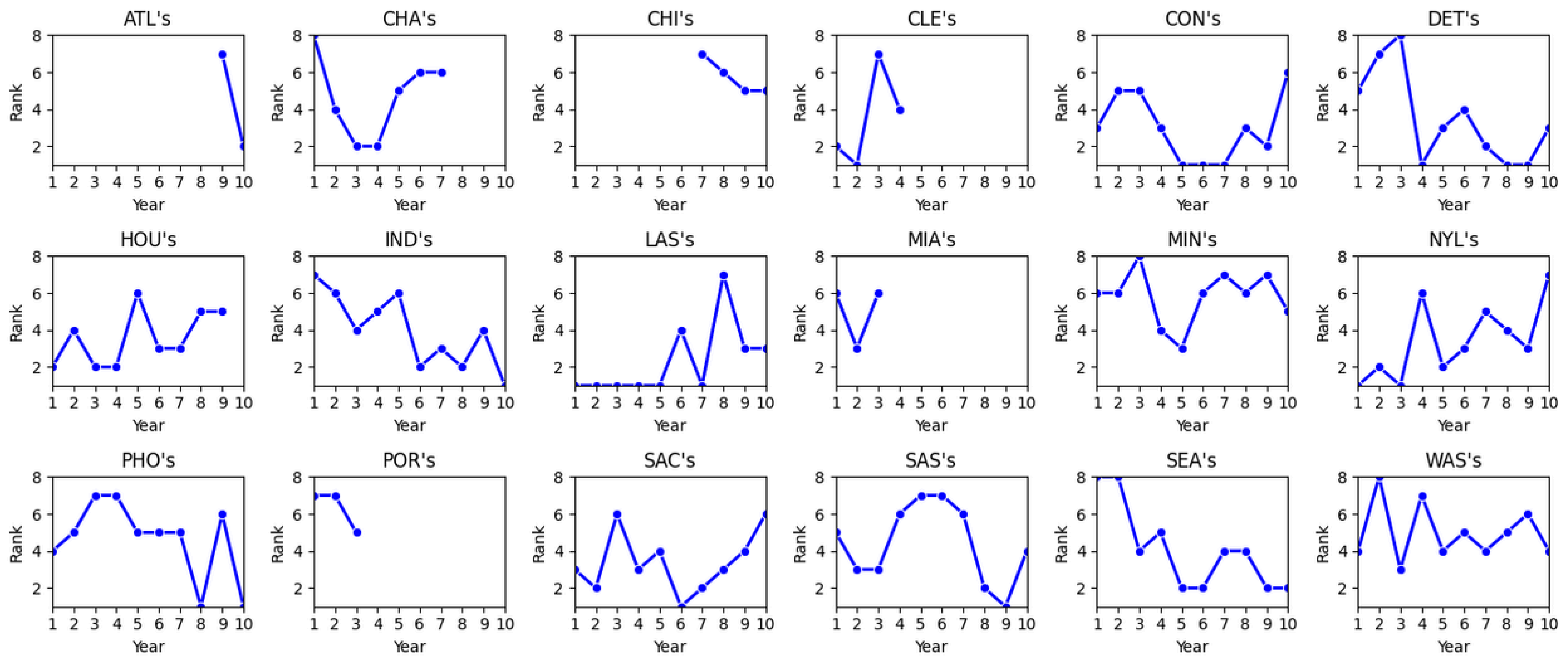


Figure 5 - Team’s ranking over the years

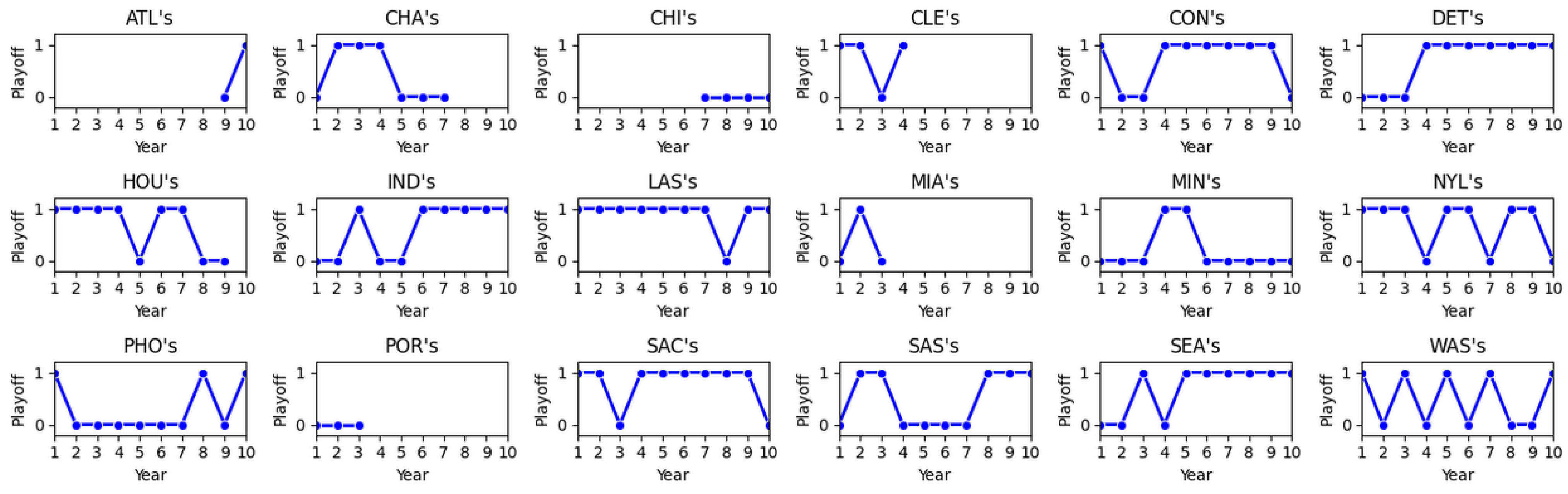


Figure 6 - Team’s presence in playoffs over the years

Exploratory Data Analysis

Figure 7 - The number of player awards a team has doesn't seem to have much of a correlation with its playoff presence. For example, the New York Liberty have a lot of playoff appearances but few awards.

There doesn't seem to be a relationship.

Figures 8 and 9 - All teams have more wins and fewer losses when playing at home. However, some seem to be more influenced than others.

For example, Atlanta doesn't show a big difference, but Seattle does.

There appears to be a consistent pattern.

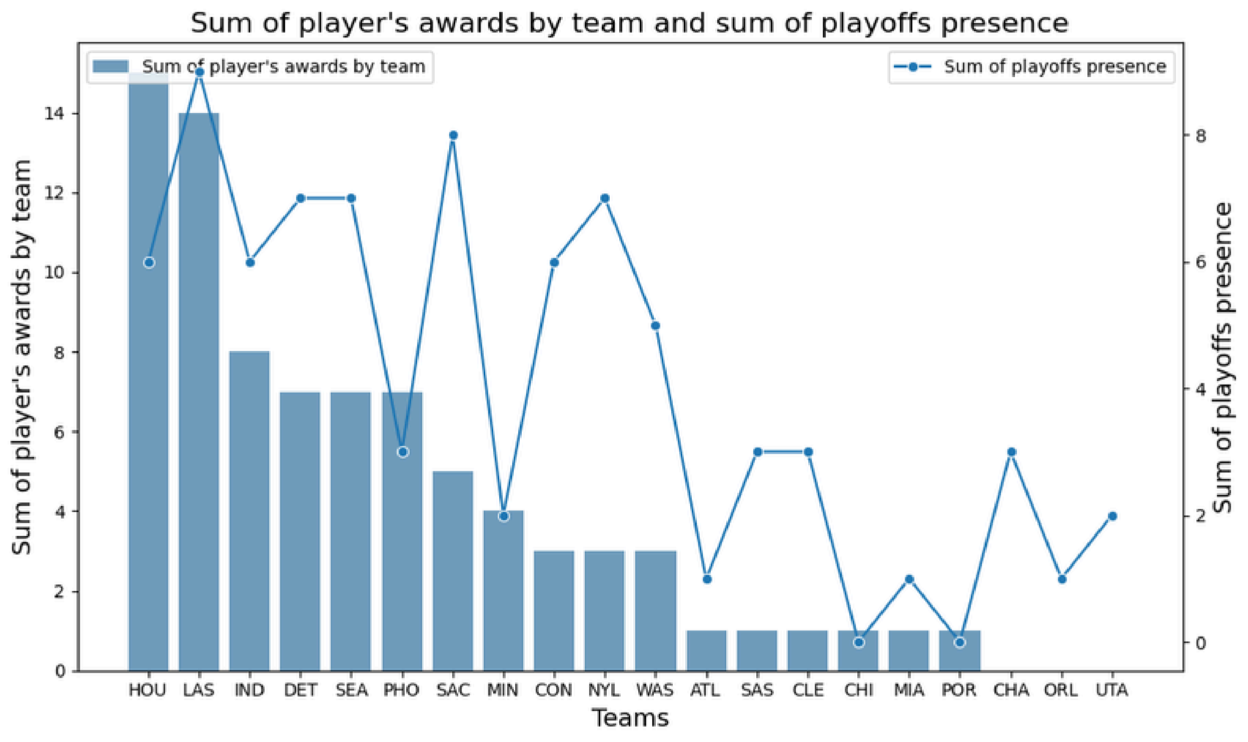


Figure 7 - Relation between teams numbers of playoff and numbers o awards

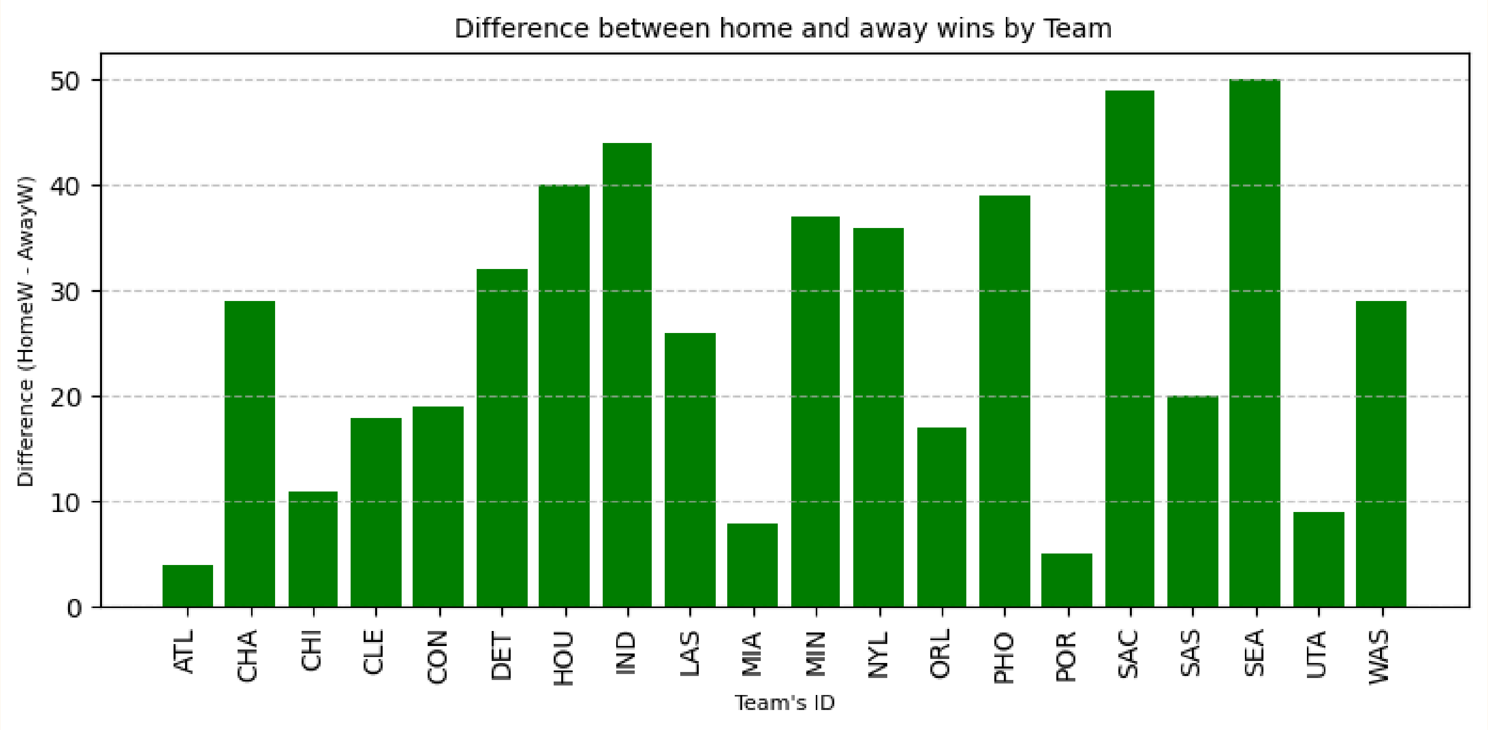


Figure 8 - Difference between home and away wins by Team

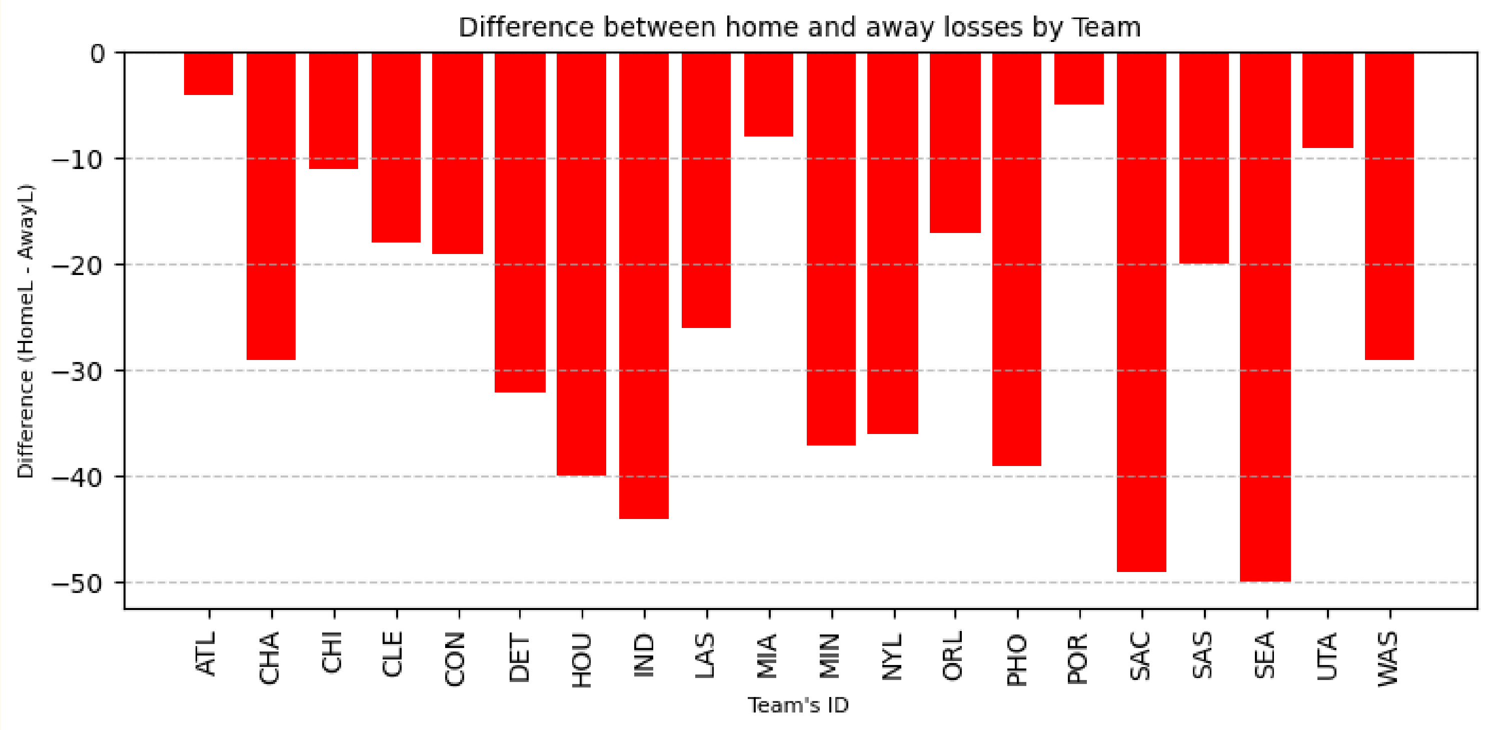


Figure 9 - Difference between home and away losses by Team

Exploratory Data Analysis

Figure 10 - Some features have a high correlation with other features. However, in the specific case of the "playoff" attribute, it does not seem to be very correlated with the other attributes.

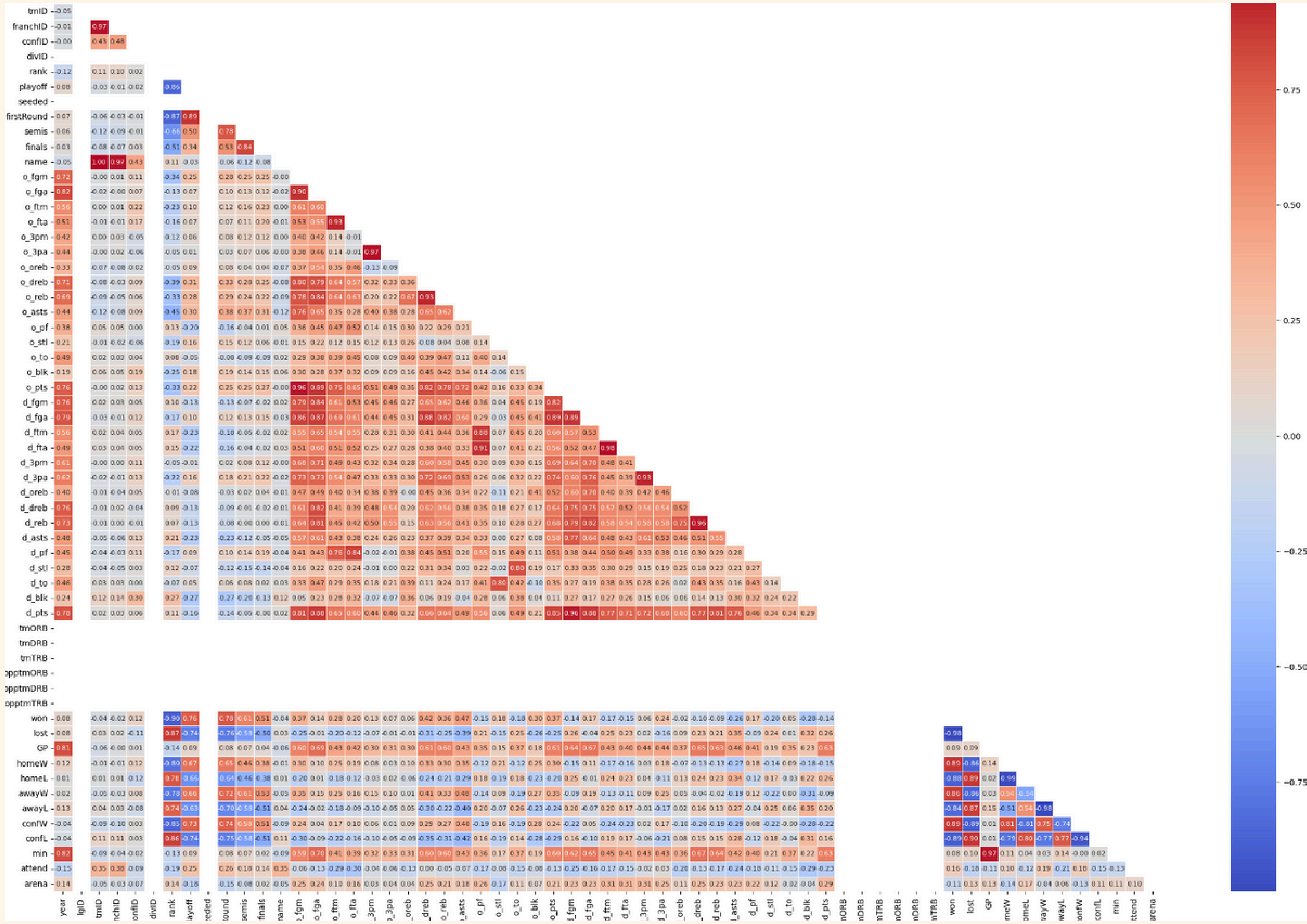


Figure 10 - Correlation between team features

Exploratory Data Analysis

Figure 11 - From teams_post it is possible to see that some teams have a high number of wins, as well as losses, which leads us to believe that they are the ones that go to the playoffs more often.

Figure 12 - To prove what was predicted from figure 11, in figure 12 it is possible to see in fact the teams that stand out in the number of appearances in teams_post, as is the case of LAS and SAC.

Figure 13 - As expected, the more minutes played, the more points are scored. It will therefore be interesting to evaluate the number of points scored per minute to assess the player's performance.

Figure 14 - You can see the number of goals scored by each player, and the team they were associated with when they scored. As you can see, all teams have players of different levels. However, teams like UTA and ATL seem to have fewer players scoring many goals, which suggests that they are less strong.

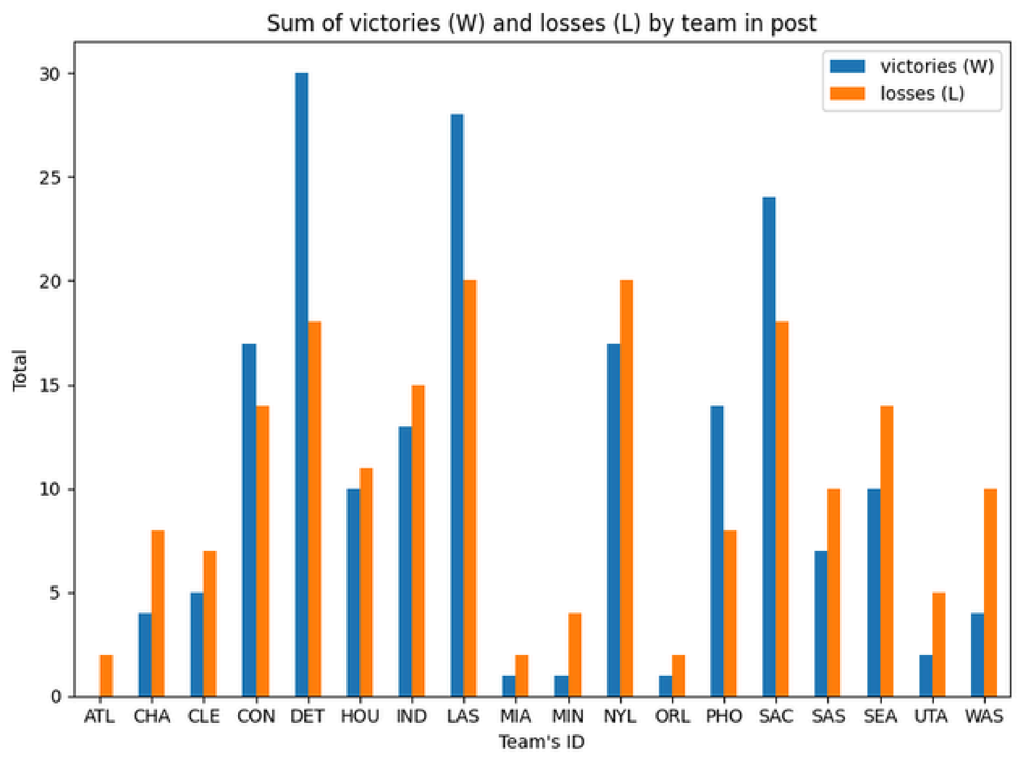


Figure 11 - Wins and losses in teams_postt

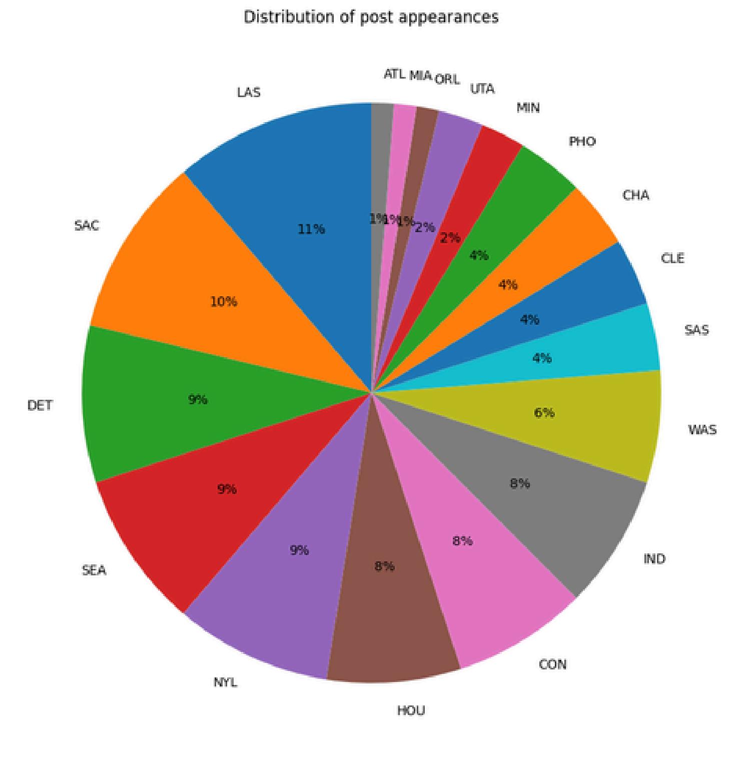


Figure 12 - Distribution of post appearances

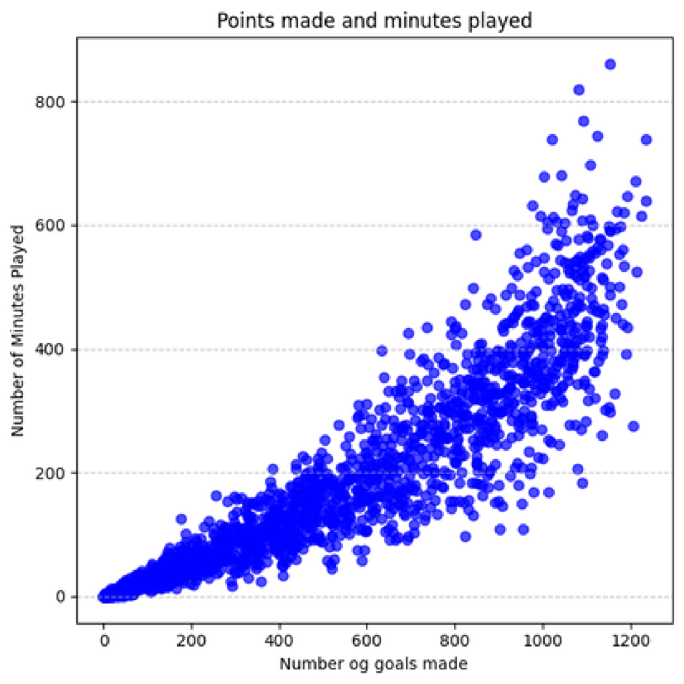


Figure 13 - Points made and minutes played

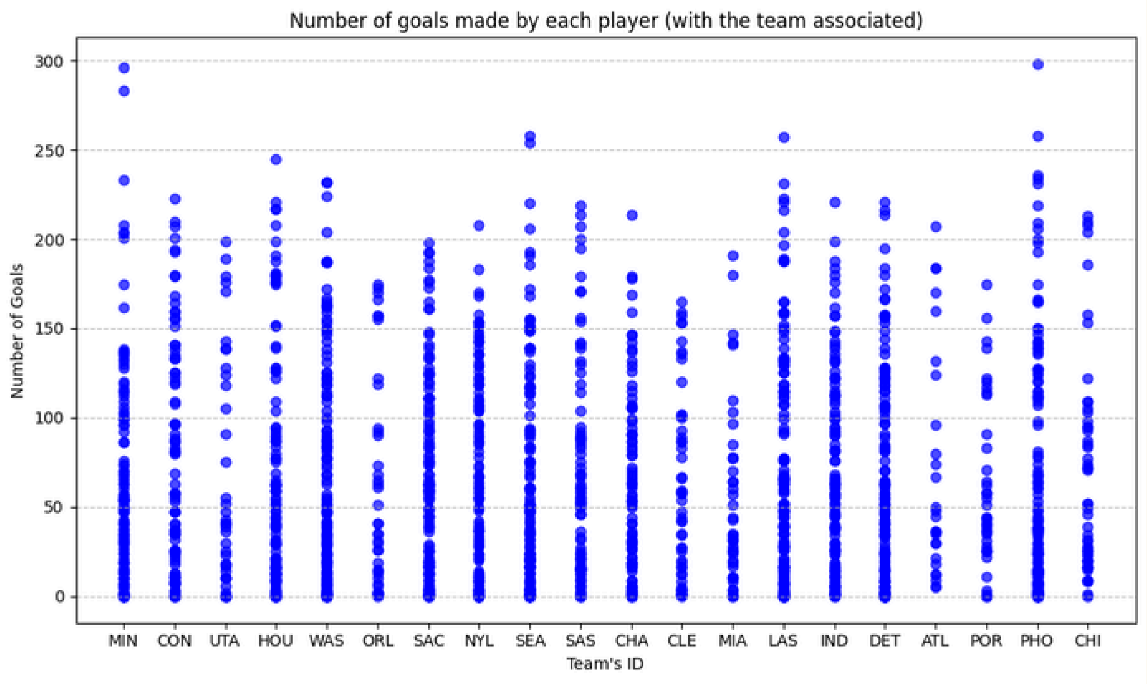


Figure 14 - Number of goals made by each player

Exploratory Data Analysis

Figure 15 - The height and weight of players always seem to be around the same values, but there are cases where a player has a height of 0 and/or a weight of 0. This is impossible to happen.

There seem to be inconsistencies.

Figure 16 - There are several players who only played for one year, with few playing for years. The number of times they played can be an indication of their experience, and therefore their quality.

Figure 17 - The distribution of player positions is not uniform. The "G" and "F" positions are much more represented in our dataset.

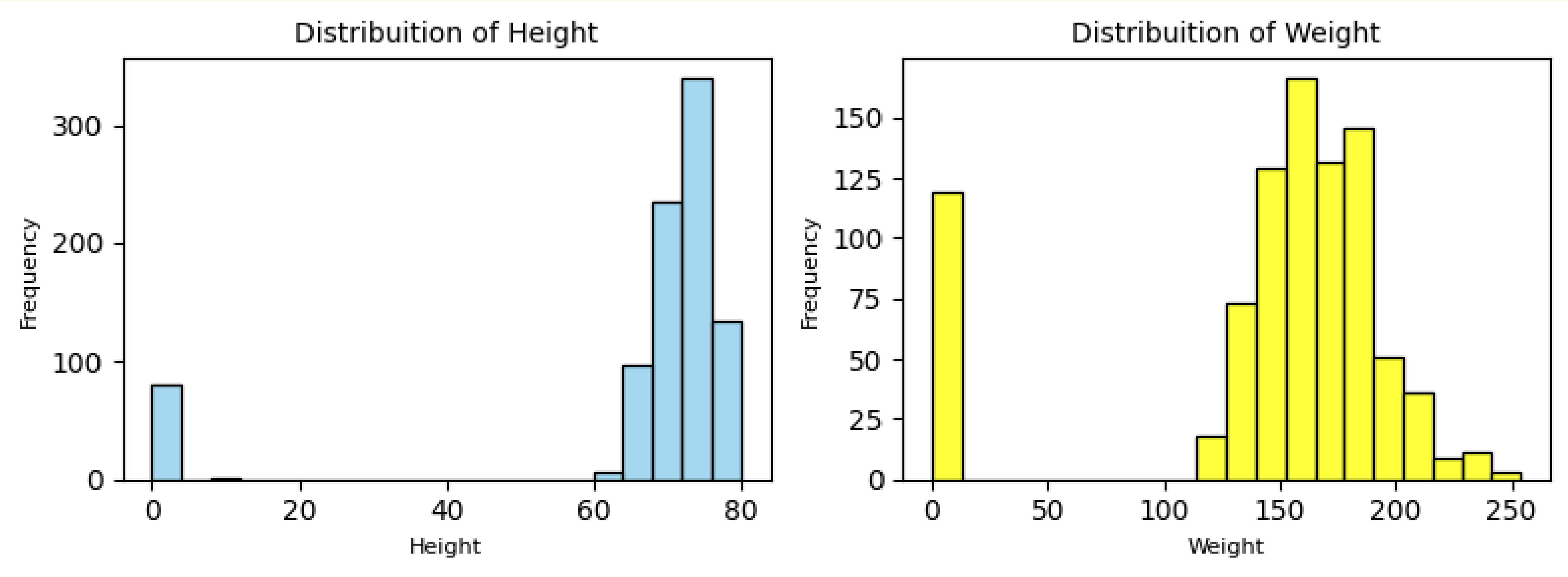


Figure 15 - Distribuition of the height and weight of the players

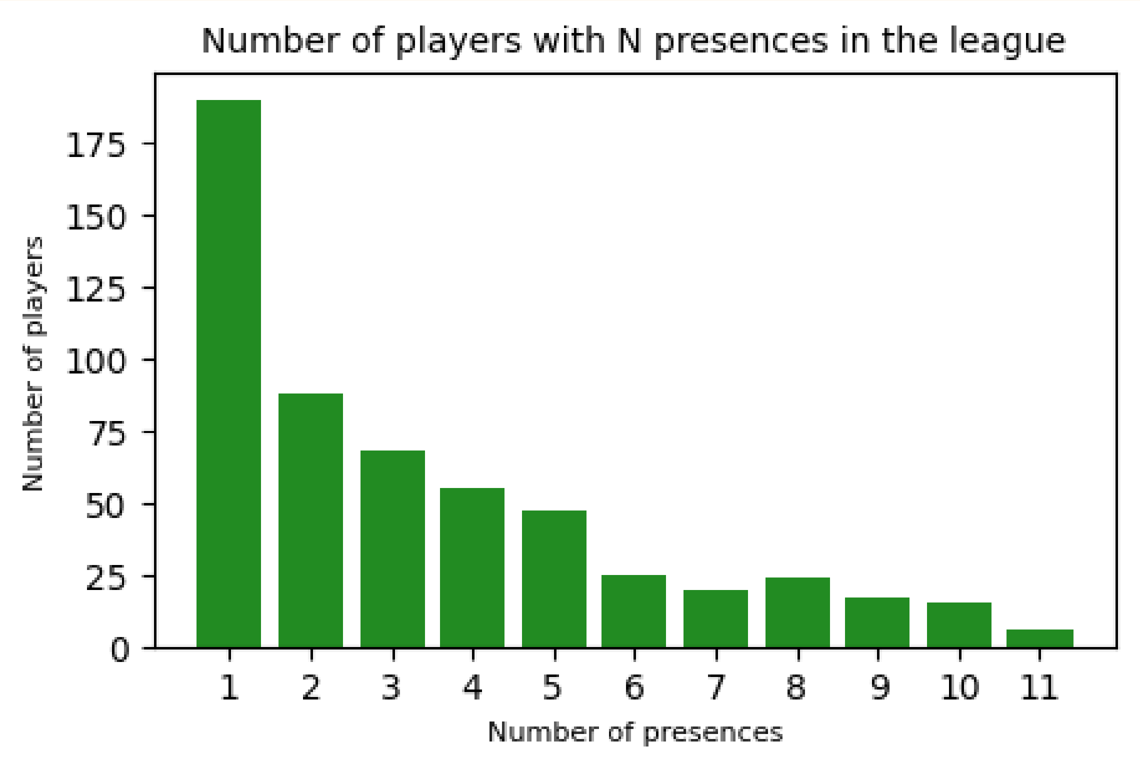


Figure 16 - Number of players with N presences in the league

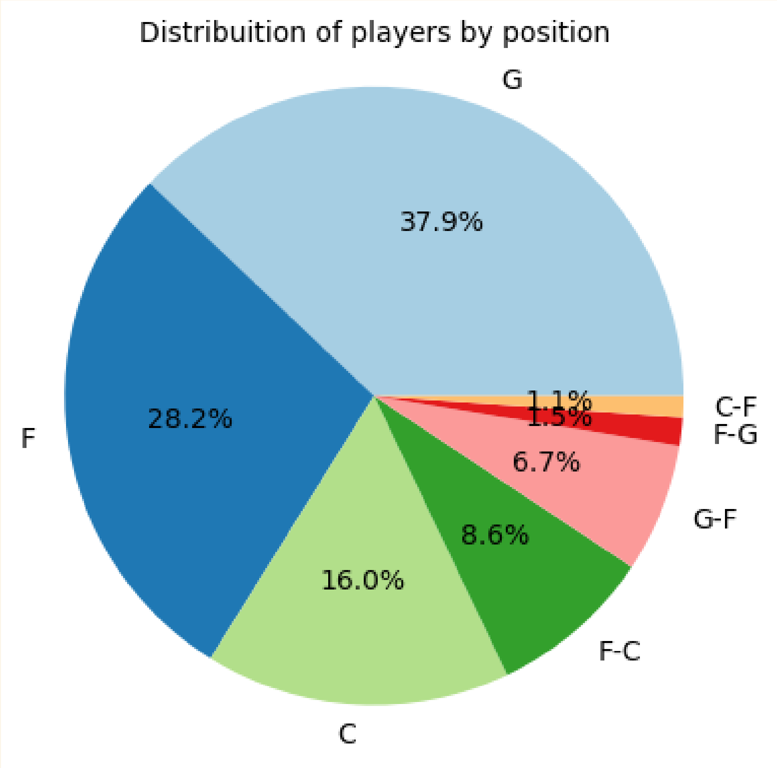


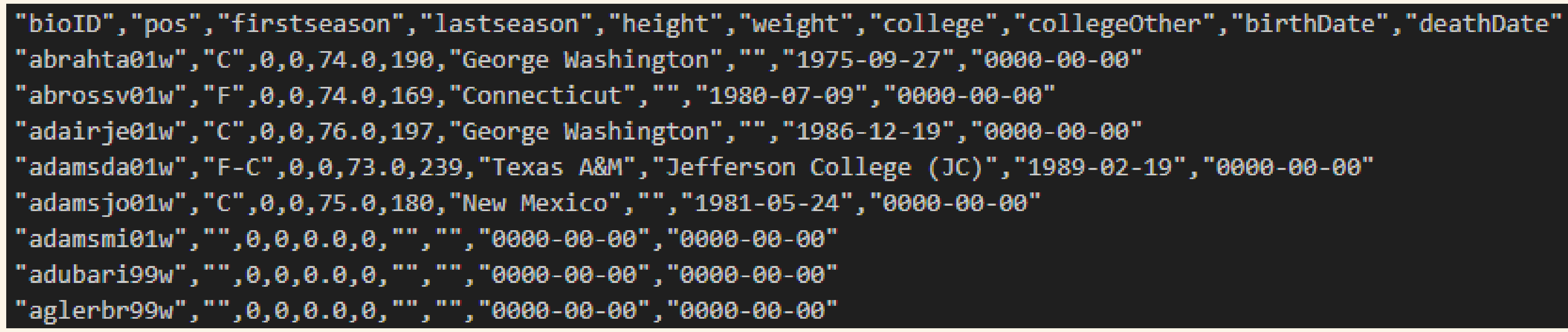
Figure 17 - Distribuition of players by position

Data Preparation

Assessment of Dimensions of Data Quality

Uniqueness

The data is unique, there are no duplicates.



```
"bioID","pos","firstseason","lastseason","height","weight","college","collegeOther","birthDate","deathDate"
"abrahta01w","C",0,0,74.0,190,"George Washington","","1975-09-27","0000-00-00"
"abrossv01w","F",0,0,74.0,169,"Connecticut","","1980-07-09","0000-00-00"
"adairje01w","C",0,0,76.0,197,"George Washington","","1986-12-19","0000-00-00"
"adamsda01w","F-C",0,0,73.0,239,"Texas A&M","Jefferson College (JC)","1989-02-19","0000-00-00"
"adamsjo01w","C",0,0,75.0,180,"New Mexico","","1981-05-24","0000-00-00"
"adamsmi01w","",0,0,0.0,0,"","","0000-00-00","0000-00-00"
"adubari99w","",0,0,0.0,0,"","","0000-00-00","0000-00-00"
"aglerbr99w","",0,0,0.0,0,"","","0000-00-00","0000-00-00"
```

Figure 18 - Inconsistencies in height, weight, date of birth and date of death values in the player dataset.

Accuracy

The accuracy is affected by the fact that some players have their height and/or weight set to 0, or dates of birth as 0000-00-00.

Completeness

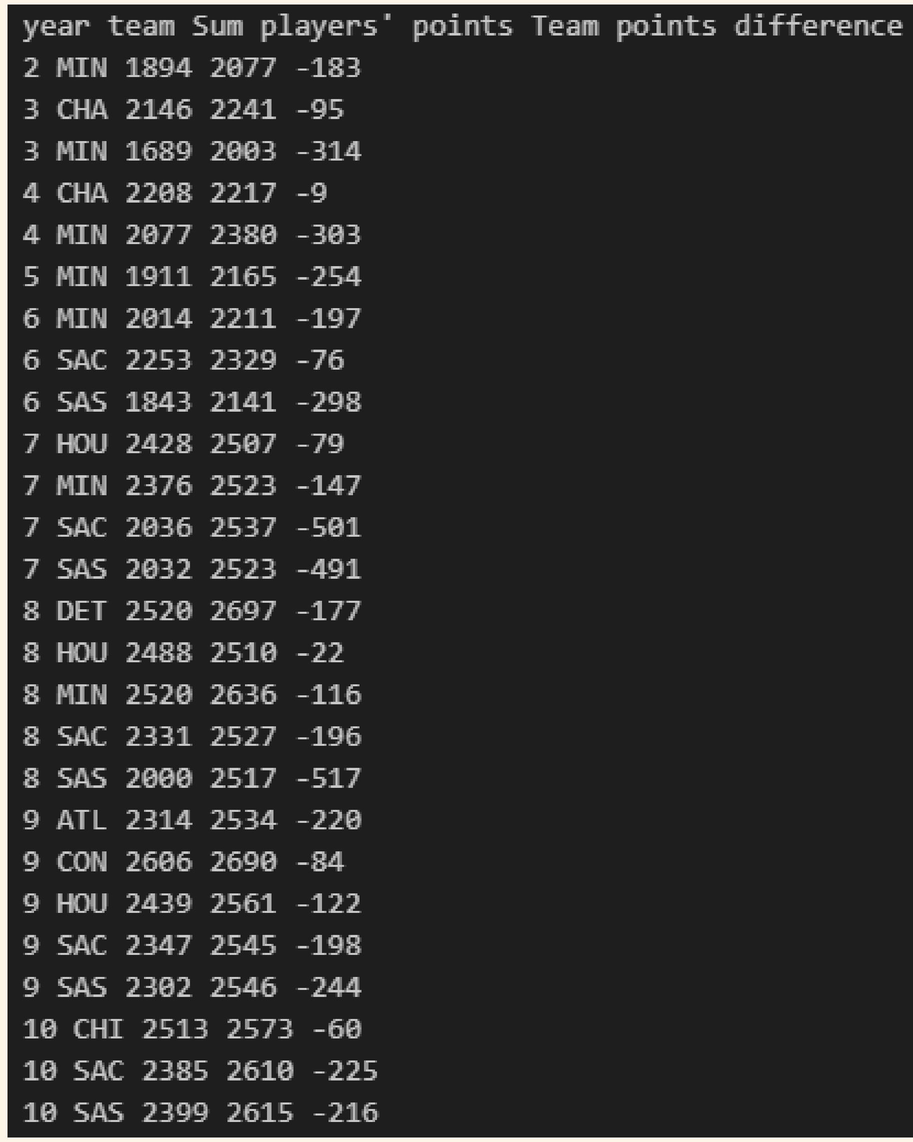
Across all the different data sets, there are multiple missing values, and even entire columns are left unfilled.

Validity

The validity of the data is affected by the fact that there are birth dates recorded as 0000-00-00, and heights and weights as 0.

Consistency

The data is not always consistent. When calculating the data relating to points, assists, field goals made, field goals attempted, etc., of each team in the year from the sum of the values of each player we notice that there are inconsistencies when compared with the values recorded in the teams (Figure 19).



```
year team Sum players' points Team points difference
2 MIN 1894 2077 -183
3 CHA 2146 2241 -95
3 MIN 1689 2003 -314
4 CHA 2208 2217 -9
4 MIN 2077 2380 -303
5 MIN 1911 2165 -254
6 MIN 2014 2211 -197
6 SAC 2253 2329 -76
6 SAS 1843 2141 -298
7 HOU 2428 2507 -79
7 MIN 2376 2523 -147
7 SAC 2036 2537 -501
7 SAS 2032 2523 -491
8 DET 2520 2697 -177
8 HOU 2488 2510 -22
8 MIN 2520 2636 -116
8 SAC 2331 2527 -196
8 SAS 2000 2517 -517
9 ATL 2314 2534 -220
9 CON 2606 2690 -84
9 HOU 2439 2561 -122
9 SAC 2347 2545 -198
9 SAS 2302 2546 -244
10 CHI 2513 2573 -60
10 SAC 2385 2610 -225
10 SAS 2399 2615 -216
```

Figure 19 - Difference between sum of players's points and team's points.s

Timeliness

Training and testing data are from sequential years, so timeliness is guaranteed.

Data Preparation

Data Cleaning

1	Replace values in columns “tmID”, from all the datasets, for the value in “franchID”, from the **teams**.
2	Remove columns with missing values or one unique value, except “weight” and “height” from the table **players**.
3	Remove lines where “minutes” is 0 from **players_teams**, as it means that the player did not play during the season.
4	Remove “award” from **awards_players**, because we did not consider it to be a relevant column.
5	Remove “stint” from **coaches** and **players_teams**.
6	Remove “college”, “collegeOther”, “birthDate” and “deathDate” from **players**, as we did not consider them to be relevant.
7	Remove “franchID”, “firstRound”, “semis”, “finals”, “attend”, “name” and “arena” from **teams**.
8	Fill missing values in “heigh” and “weight” from the table **players** using RandomForestRegressor. An outlier was found and its value was also replaced (before it had the value 9 in “height”).

Data Preparation

Data Integration

1	Join the table **players** with **players_teams**, by the attributes “bioID” (from **players**) and playerID **(**from **players_teams**) .
2	Add the column “awards” in the tables **players_teams** and **coaches**. This column is the sum of the awards won by a certain player or coache in the year. This information comes from the table **awards_players**.
3	Join the table **teams_post** with **teams**, by the attributes "year" and "tmID".
4	Join the table **series_post** with **teams**, by adding columns wonGamePost and lostGamePost, that are the sum of games won/lost in the playoffs.
5	Join “awards” column of **players_teams** and **coaches** with **teams**, by adding columns “awards_players“ and ”awards_coaches” that are the sum of awards won in the year.
6	Join **teams** with **players_teams**.

Data Preparation

Feature Engineering

1	Add "pie" and "per" to **players_teams**, which are the result obtained from the player impact estimate’s formula and from the player efficiency rating’s formula, respectively.
2	Add “avg_pie” and “avg_per" to **teams**, which are the average “pie” and “per”, respectively, of the 5 players with the most minutes on each team.
3	Add "offensive_efficiency", “defensive_efficiency”, "possession" and “opponent_possession” to **teams**.
4	Add "play percentage" to **teams**, which is the team's performance during the game, that is, without free throws.
5	Add "factors4" to **teams**, which is the combination of the percentages of effective field goal and offensive rebounding, with the turnover and free throw rates.
6	Add “per_o_fgm”, “per_o_ftm”, “per_o_3pm”, “per_d_fgm”, “per_d_ftm” and “per_d_3p” to **teams**, which represent the percentages of goals, shots and 3 points made divided by those attempted, respectively. On the other hand, the columns "o_fgm", "o_fga", "o_ftm", "o_fta", "o_3pm", "o_3pa", "d_fgm", "d_fga", "d_ftm", "d_fta", "d_3pm" and "d_3pa" were deleted.

Data Preparation

Data Transformation

We apply label encoding to binary features such as "playoff" and "confID", and also apply it to categorical features such as "tmID".
We also normalize the data with StandardScaler, as it is essential for using the SVM algorithm.

Imbalanced Data

We apply SMOTE to the training data in order to improve the model's performance, thus having more examples of the minority class.

Feature Selection

Feature selection was performed using the SelectKBest method with the ANOVA F-value test (f_classif) to identify the most relevant features. This approach evaluates the statistical relationship of each feature with the target variable (Y_train) and assigns an F-score, with higher scores indicating greater relevance. In our case, we chose to select only 20 features with the highest F-scores.

Training and Testing Dataset

The training dataset includes data from all years prior to year 10, while the test dataset is exclusively composed of data from year 10.

Experimental Setup

Algorithms Tested

- **Decision Tree Classifier -** Uses a tree structure, creating divisions in the data based on features to separate different classes.
- **Support Vector Machine Classifier -** Finds the hyperplane that separates the classes so that the distance between them is maximized.
- **Gaussian Naive Bayes -** Uses Bayes' Theorem to make predictions, assuming that features are independent of each other and follow a normal (Gaussian) distribution.
- **Random Forest Regressor -** Creates several decision trees, training each one on a random subset of the data and features, called bagging. The final prediction is the average of the predictions from all trees.
- **Linear Regression -** Finds the line that minimizes the sum of squared residuals, this is, differences between predicted and actual values.

Performance Metrics

- **Time** (s) - Duration it takes for an algorithm to train and make predictions.
- **Accuracy** - Evaluates how often the prediction model correctly predicts the outcome. It's the ratio of correct predictions (both true positives and true negatives) to the total number of predictions.
- **Precision** - Calculates the proportion of true positive predictions out of all the instances the model predicted as positive.
- **Recall** - Measures the proportion of actual positive instances that were correctly identified by the model.
- **F1** - Combines precision and recall into a single metric by calculating their harmonic mean.
- **AUC** - Refers to the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate (recall) against the false positive rate.
- Error

Analysis of Results - submission 1

- The results show that the SVM and LR are the models that stands out in terms of performance, outperforming the other models. Meanwhile, DT, GNB, and RFR exhibit similar performance, with lower accuracy and precision. Although DT and GNB are faster, SVM offers a better balance between performance and results, while RFR is the slowest and least efficient model.
- To produce Figure 20 the error was normalized to be between 0 and 1 (not between 0 and 12). The error was also inverted to make it easier to visualize, and is now called "success", because the higher the "success" the better the algorithm.
- **This initial results were obtained without applying SMOTE, parameter tuning, or feature selection.**

Model	Time (s)	Error	Accuracy	Precision
DT	0.016	5.00	0.43	0.50
SVM	0.017	2.99	0.71	0.75
GNB	0.012	3.85	0.43	0.50
RFR	0.191	4.22	0.43	0.50
LR	0.027	2.37	0.71	0.75

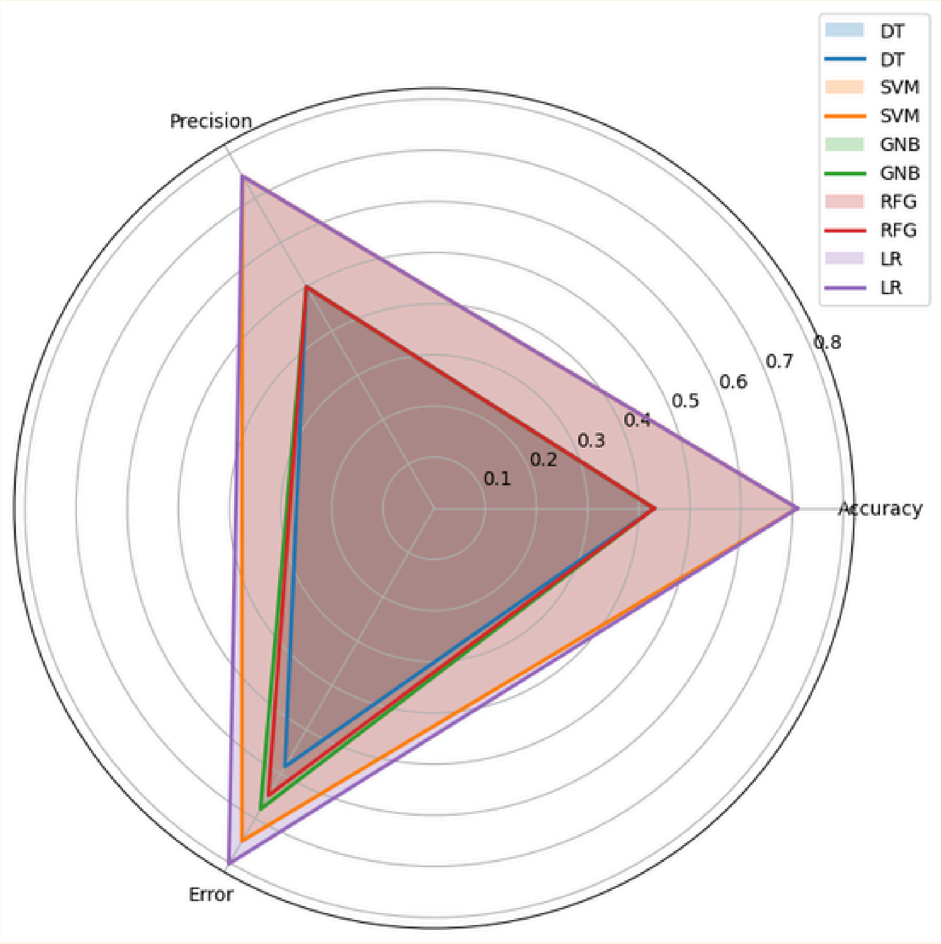


Figure 20 - Comparing the algorithms

Analysis of Results - submission 2

- The hyperparameters were optimized using **Optuna**, with the model's performance evaluated over the years 2 to 10, using an objective function that minimizes the negative average score across these years. After optimization, the best hyperparameters are used to train a final model, which is subsequently evaluated on year 11.
- This optimization brought clear improvements for some models, especially in terms of accuracy and error, but the impact on execution time varies depending on the model. The optimization also highlighted that for some models, such as LR, the available hyperparameters may have little impact on performance, in its case “fit_intercept” and “copy_X”.
- We stopped testing with the RFR model since with this optimization its execution time increased considerably.

Model	Time (s)	Error	Accuracy	Precision
DT	0.010 (before 0.016)	3.33 (before 5.00)	0.57 (before 0.43)	0.62 (before 0.50)
SVM	0.010 (before 0.017)	3.10 (before 2.99)	**0.71** (before 0.71)	**0.75** (before 0.75)
GNB	0.015 (before 0.012)	3.82 (before 3.85)	0.50 (before 0.43)	**0.50** (before 0.50)
LR	**0.027** (before 0.027)	**2.37** (before 2.37)	**0.71** (before 0.71)	**0.75** (before 0.75)

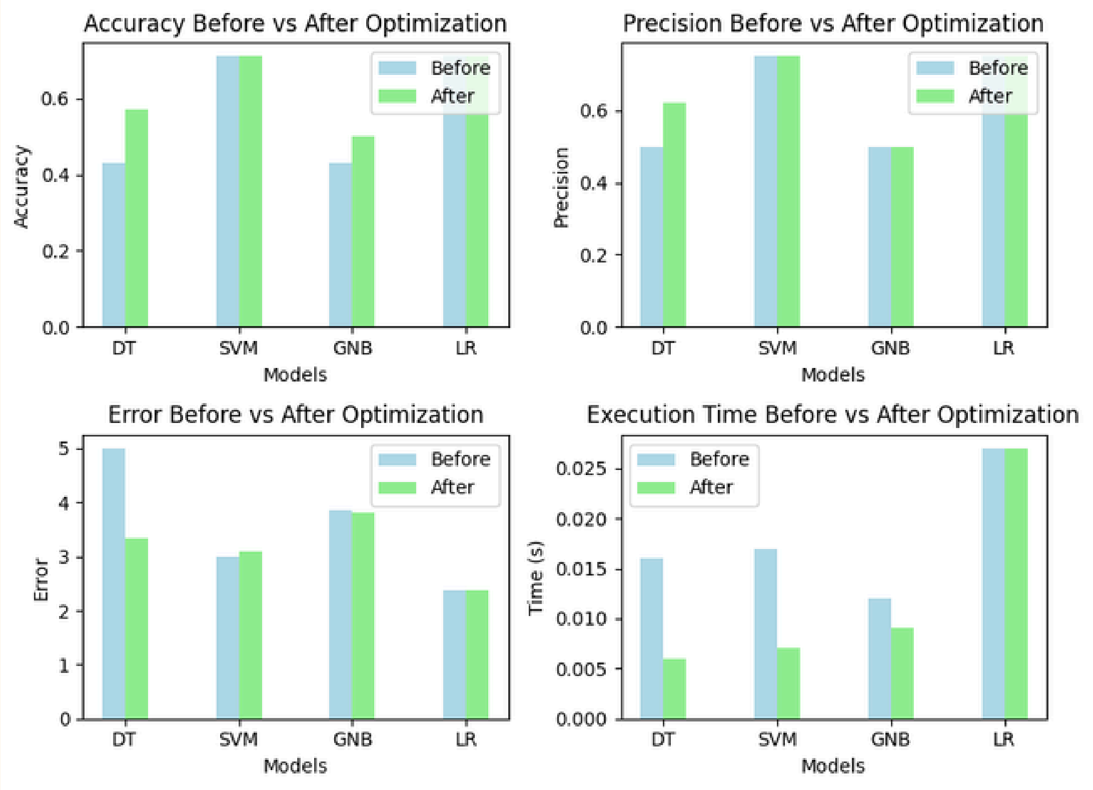


Figure 21 - Comparing the algorithms

Analysis of Results - submission 3

- We now applied **SMOTE** to the training data, so that the data is balanced
- SMOTE was effective in balancing the training data, which helped improve some metrics such as accuracy and precision, especially for models that benefit from balanced data, such as DT and LR.

Model	Time (s)	Error	Accuracy	Precision
DT	0.016 (before 0.010)	3.36 (before 3.33)	0.71 (before 0.57)	0.75 (before 0.62)
SVM	0.013 (before 0.010)	3.35 (before 3.10)	0.57 (before 0.71)	0.62 (before 0.75)
GNB	0.012 (before 0.015)	3.68 (before 3.82)	0.43 (before 0.50)	**0.50** (before 0.50)
LR	0.059 (before 0.027)	2.27 (before 2.37)	0.86 (before 0.71)	0.88 (before 0.75)

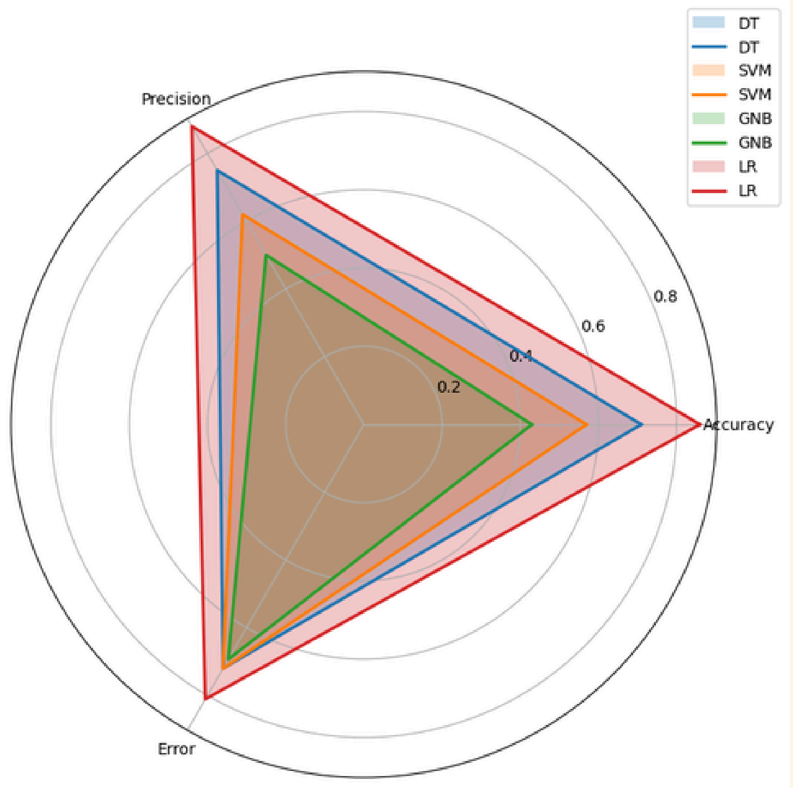


Figure 22 - Comparing the algorithms

Analysis of Results - submission 4

- We performed feature selection using **SelectKBest**. For this, we used the F-statistic from the **ANOVA** test (**f_classif**) as the metric, which measures how well each feature separates the classes of the target variable. In our case, we selected only the 20 best features. New features have also been introduced

- Feature selection brought improvements in runtime for all models, but the impact on performance metrics was mixed. **DT and GNB benefited the most**, while SVM and LR maintained more stable performance, although with some loss of accuracy in LR. This suggests that reducing the number of features may be beneficial for simpler models, but may not have the same positive effect on models such as LR, which can be more sensitive to information loss.

Model	Time (s)	Error	Accuracy	Precision
DT	0.004 (before 0.016)	2.93 (before 3.36)	**0.71** (before 0.71)	**0.75** (before 0.75)
SVM	0.007 (before 0.013)	3.15 (before 3.35)	**0.57** (before 0.57)	**0.62** (before 0.62)
GNB	0.008 (before 0.012)	3.08 (before 3.68)	0.57 (before 0.43)	0.62 (before 0.50)
LR	0.004 (before 0.059)	2.72 (before 2.37)	0.57 (before 0.86)	0.62 (before 0.88)

Analysis of Results

- The execution time benefited mainly from the selection of only 20 features.
- The DT model showed the most constant evolution, having practically always improved its results as more optimizations were made.
- On the other hand, the LR model, which had already obtained good results in the optimized version, ended up being harmed by the new changes. Although in its final version it continues to be the one that appears to produce results with less error, its precision and accuracy ended up decreasing considerably. This happened due to the reduction in the number of features with which the algorithms were trained, which caused this algorithm to be greatly harmed by the loss of information.

The models showed a positive evolution in terms of execution time and performance, with emphasis on the stability of the Decision Tree. The application of iterations proved to be beneficial for some models, especially with regard to time optimization, but the impact on accuracy and error was varied.

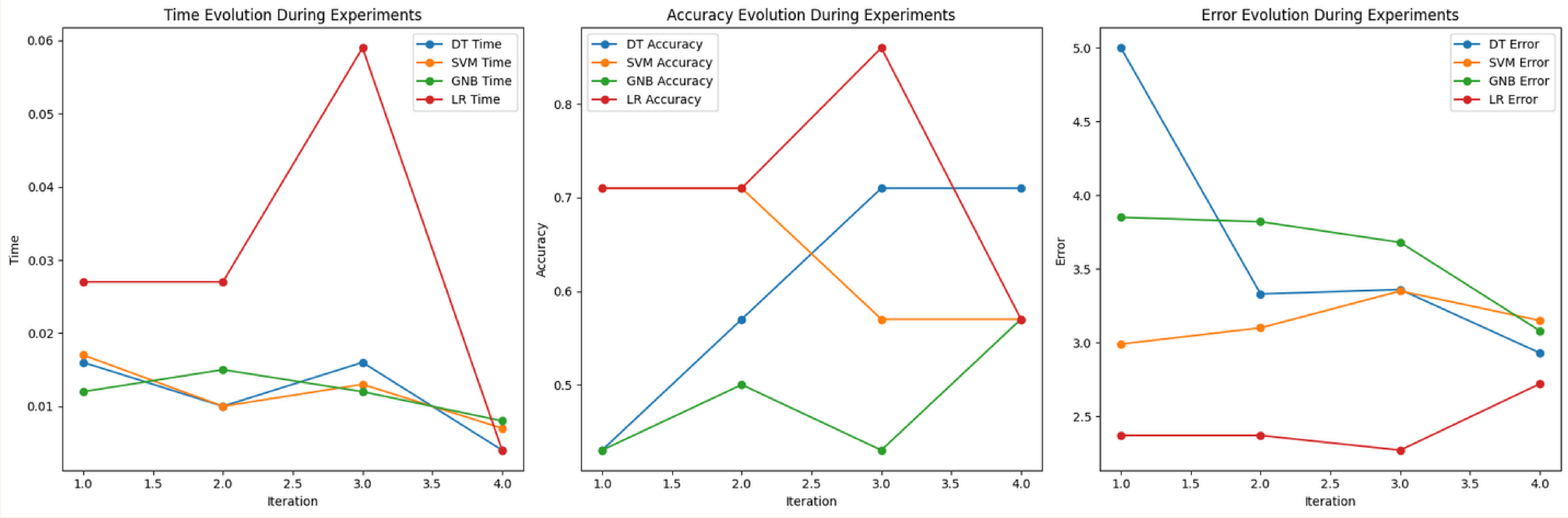


Figure 23 - Comparing the algorithms

Conclusions, Limitations and Future Work

- **With the Decision Tree model, we achieved an accuracy of 0.75, which we consider a successful outcome.**

- The dataset provided contains some **inconsistencies**. For example, the sum of the points of the team's players in a given year is different from the value recorded by the team in that same year.

- **One area for improvement is the feature selection process**. Currently, we limit the selection to the top 20 features based on statistical relevance, but a more dynamic approach could be implemented. Specifically, we could use methods that allow the algorithm to determine not only which features are most important but also the optimal number of features to include. This could be achieved through techniques such as Recursive Feature Elimination with Cross-Validation (RFECV).

Annexes

Data Preparation

Feature Engineering

New columns were created from existing ones. Here we present the formulas used to obtain them as well as their meaning.

Player Impact Estimate (PIE)

PIE is a metric for evaluating a player's overall contribution to the game. Since we don't have the statistical data for a single game, we have adapted the formula so that instead of being a player's contribution in a game, it is in relation to the season. Almost all statistical categories in the box score are involved in the PIE formula.

players_events = pts + fgm + ftm – fga – fta + Deff.reb + Off.reb/2 + ast + stl + blk/2 – pf – to
game_events = Game.pts + Game.fgm + Game.ftm – Game.fga – Game.fta + Game.Deff.reb + Game.Off.reb/2 + Game.ast + Game.stl + Game.blk/2 – Game.pf – Game.to
PIE = players_events / game_events

Possession

It counts as team possession every time a player from that team attempts a field goal, misses a shot and does not catch the offensive rebound, loses the ball and goes to the line for two or three shots and makes the last shot or does not catch the rebound of a last missed shot.

Possession = 0.96 * (fga + to + 0.44 * fta - Offensive.reb)

Pace

Pace factor is an estimate of the number of possessions per 40 minutes by a team.

Pace = =[240/(tmMin)]*(tmPossession+oppPossession)/2

Data Preparation

Feature Engineering

New columns were created from existing ones. Here we present the formulas used to obtain them as well as their meaning.

Offensive Efficiency

Offensive efficiency is the number of points a team scores per 100 possessions.

```
Offensive_Efficiency = 100 * (points_scored / possessions)
```

Defensive Efficiency

Defensive efficiency is the number of points a team allows per 100 possessions.

```
Offensive_Efficiency = 100 * (points_allowed / possessions)
```

Four Factors

Four Factors are the box score derived metrics that correlate most closely with winning basketball games. These factors also identify a team's strategic strengths and weaknesses. Four factors can be applied to both a team's offense and defense, hence it gives us eight factors.

```
effective_field_goal_percentage = (fgm + 0.5 * 3pm ) / fga
turnover_rate = to / ( fga + 0.44 * fta + to )
offensive_rebounding_percentage = Offensive.reb / ( Offensive.reb + Opponent.Defensive.reb )
free_throw_rate = ftm / fga
four_factor = 0.40 * effective_field_goal_percentage + 0.25 * turnover_rate + 0.20 * offensive_rebounding_percentage + 0.15 * free_throw_rate
```

Data Preparation

Feature Engineering

New columns were created from existing ones. Here we present the formulas used to obtain them as well as their meaning.

Play Percent

The metric that indicates the percentage of the time a team will score if not sent to the free throw line.

```
Play_Percent = fgm / ( fga - Offensive.reb + to )
```

Player Efficiency Rating

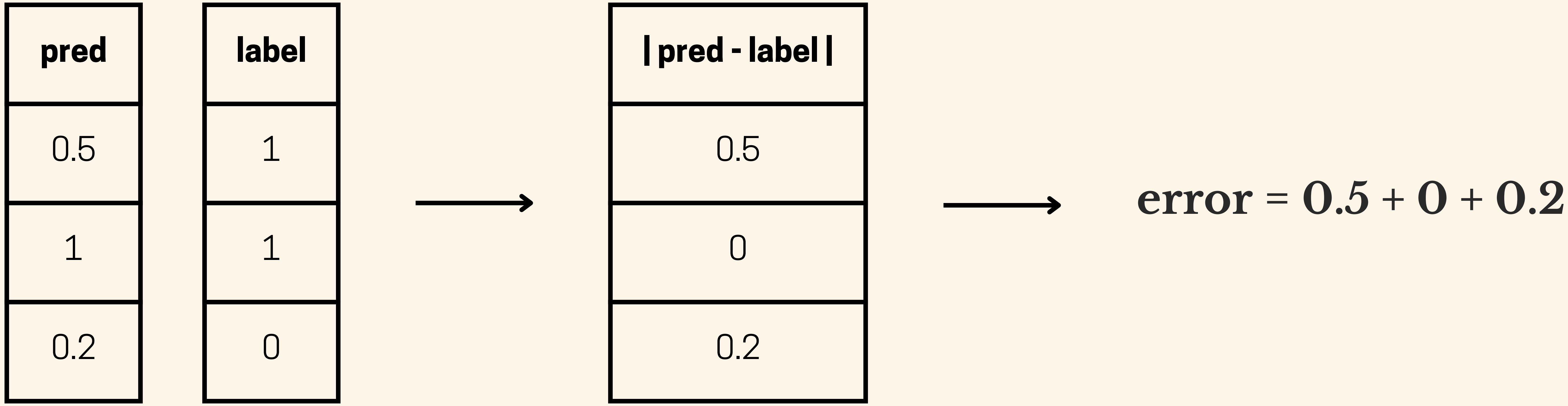
Attempts to collect all of a player’s contributions into one number.

```
uPER = (1 / MP) *
   [ 3P + (2/3) * AST + (2 - factor * (team_AST / team_FG)) * FG + (FT *0.5 * (1 + (1 - (team_AST / team_FG)) + (2/3) * (team_AST / team_FG)))
   - VOP * TOV - VOP * DRB% * (FGA - FG) - VOP * 0.44 * (0.44 + (0.56 * DRB%)) * (FTA - FT)
   + VOP * (1 - DRB%) * (TRB - ORB) + VOP * DRB% * ORB + VOP * STL + VOP * DRB% * BLK
   - PF * ((lg_FT / lg_PF) - 0.44 * (lg_FTA / lg_PF) * VOP) ]
where :
   - DRB% = (lg_TRB - lg_ORB) / lg_TRB
   - factor = (2 / 3) - (0.5 * (lg_AST / lg_FG)) / (2 * (lg_FG / lg_FT)) adjusts the contribution of assists relative to field goals, accounting for how much assists influence a player's efficiency. It balances the impact of a player's ability to create scoring opportunities (assists) versus their own scoring efficiency (field goals).
   - VOP     = lg_PTS / (lg_FGA - lg_ORB + lg_TOV + 0.44 * lg_FTA), which measures the value of a possession in league-wide terms. It's a normalization constant that accounts for offensive efficiency.
aPER = (lg_Pace / team_Pace) * uPER  - pace adjustment
PER = aPER * (15 / lg_aPER) - -  normalized to the league
```

Performance Metrics

Error = sum(|pred - label|)

- Model gives the probability of each team to pass to the playoffs;
- Calculate the module of the difference between the prediction and the labeled result (0 or 1);
- Sum all modules to retrieve error;



pred	label
0.5	1
1	1
0.2	0

<table>
<tr><th>| pred - label |</th></tr>
<tr><td>0.5</td></tr>
<tr><td>0</td></tr>
<tr><td>0.2</td></tr>
</table>

error = 0.5 + 0 + 0.2