

WNBA Playoff Prediction ML- 23/24

Inês Sá Pereira Estêvão Gaspar (up202007210@edu.fe.up.pt)
Individual Factor: 1.0

Lourenço Alexandre Correia Gonçalves (up202004816@edu.fe.up.pt)
Individual Factor: 1.0

Pedro Pereira Ferreira (up202004986@edu.fe.up.pt)
Individual Factor: 1.0



Introduction

- We were given a dataset of the statistics of the WNBA's teams for over 10 years;
- **Our goal: predict the teams that would be qualified for playoff the next year;**
- Methodology used: CRISP-DM;
- Several tasks made relative:
 - Business understanding;
 - Data analysis;
 - Classification.



1st Iteration

Business Understanding

- Basketball tournaments are usually split in two parts. First, all teams play each other aiming to achieve the greatest number of wins possible. Then, at the end of the first part of the season, a predetermined number of teams which were able to win the most games are qualified to the playoff season, where they play series of knock-out matches for the trophy.

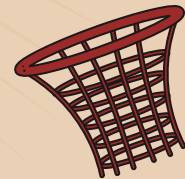
There were 16 teams in the league. For the playoffs, the four teams with the best record in each conference were seeded one to four.

Conference Semi-Finals Best-of-3			Conference Finals Best-of-3			WNBA Finals Best-of-3		
E1 Cleveland	1							
E4 Charlotte	2		E4 Charlotte	2		E4 Charlotte	0	
E2 New York	2		E2 New York	1		W1 Los Angeles	2	
E3 Miami	1							
W1 Los Angeles	2		W1 Los Angeles	2				
W4 Houston	0		W2 Sacramento	1				
W2 Sacramento	2							
W3 Utah	0							

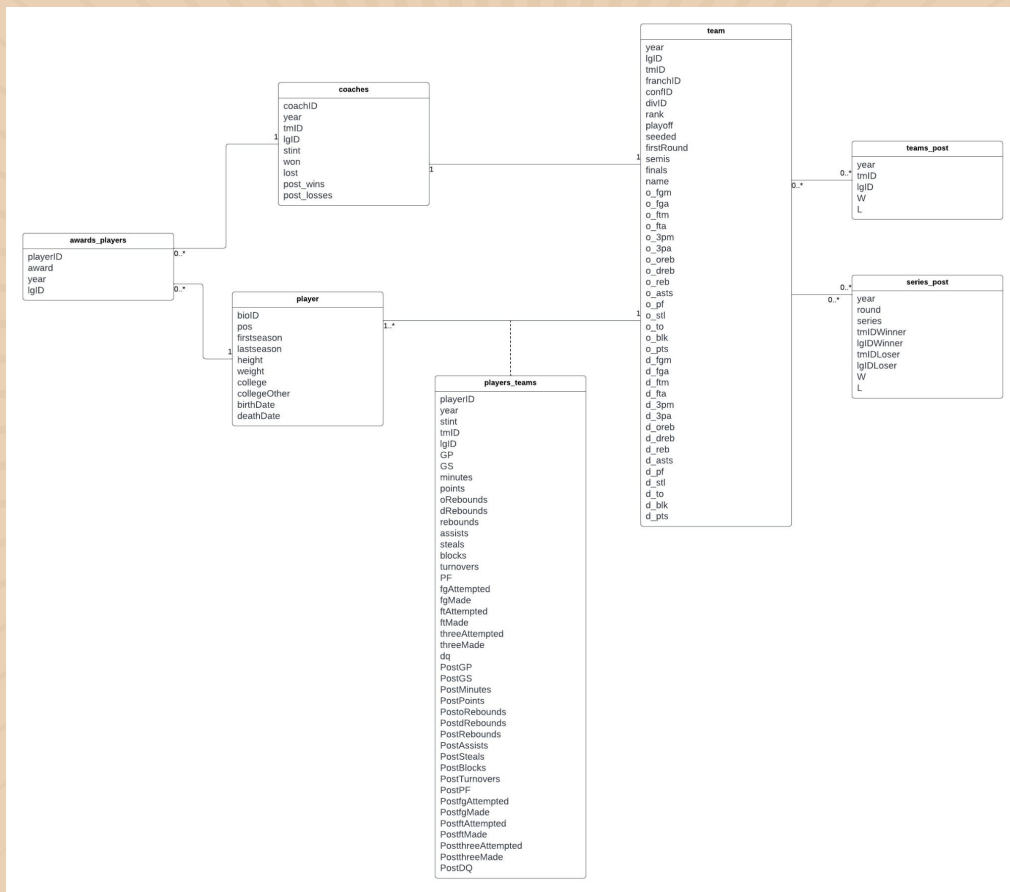
- Business goal:** Use this data to predict which teams will qualify for the playoffs in the next season, using the information given.

Datasets

- **awards_players:** Player and coach awards;
- **coaches:** Coach statistics;
- **players:** Player biological information;
- **players_teams:** Player statistics;
- **series_post:** Post-season history;
- **teams:** Team statistics;
- **teams_post:** Team post-season statistics.



Relational Model



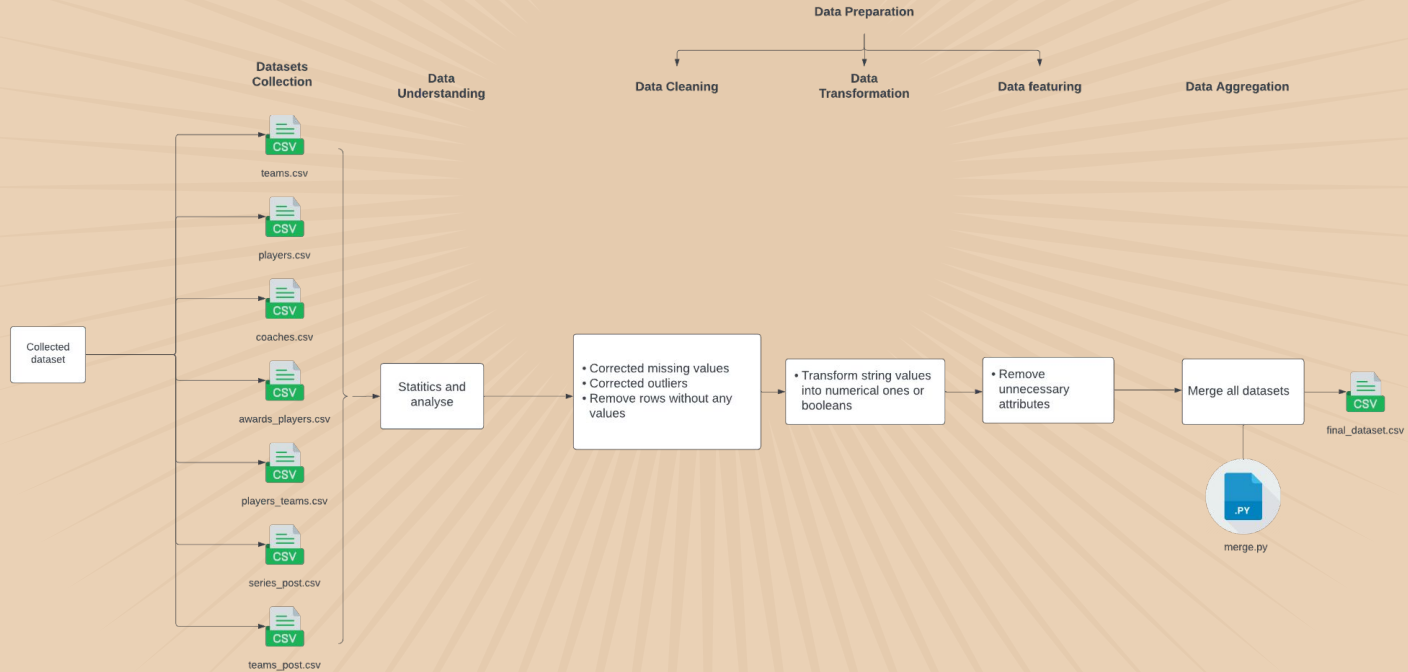
Project Management

- Tools used for:
 - Development



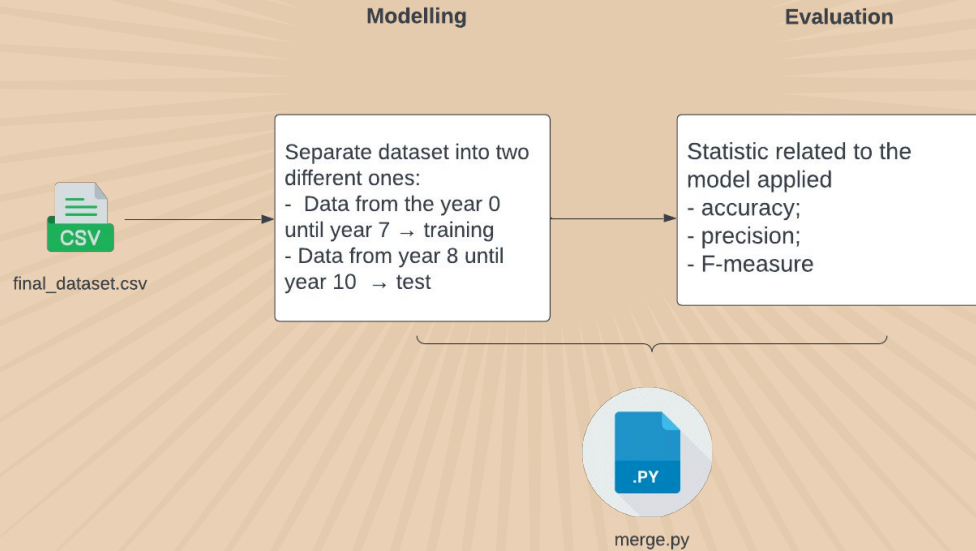
Pipeline

- Data treatment:



Pipeline (cont.)

- Modelling and evaluation:

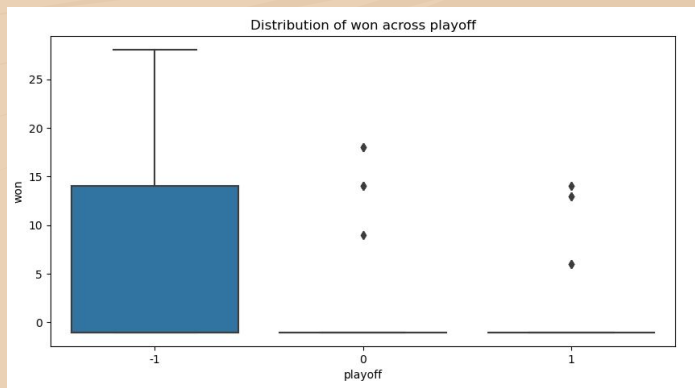


Pipeline- results

- Accuracy, precision e F-measure of the model:

```
The accuracy of the model is: 0.7418576598311218.  
The precision of the model is: 0.7908745247148289.  
The f-measure of the model is: 0.790874524714829.
```

- However, the data is not good enough, due to the merge:

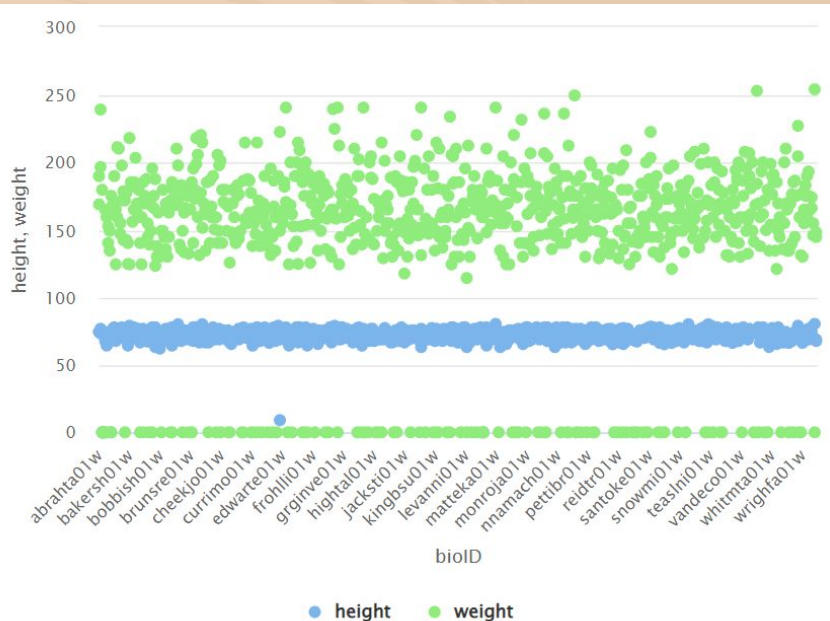


- Which leads to our second iteration.

2nd Iteration

Data understanding

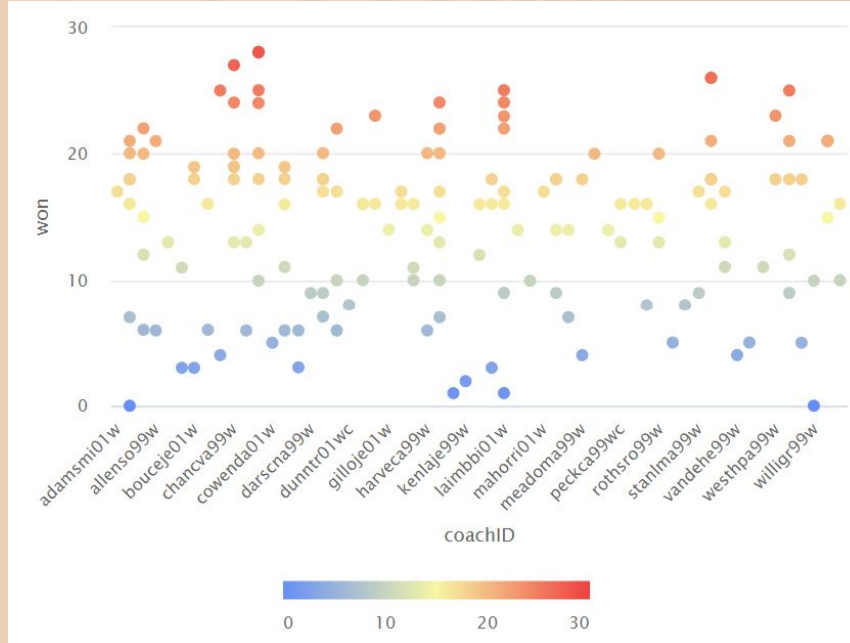
Players



By analyzing this dataset, we were able to detect quite a few missing values and outliers, especially for height and weight. However, in general terms they all have values between 50 and 100 for height and between 120 and 250 for weight. From the analysis of the attributes, we also realized that some would not be relevant to the prediction we were going to make.

Data understanding (cont.)

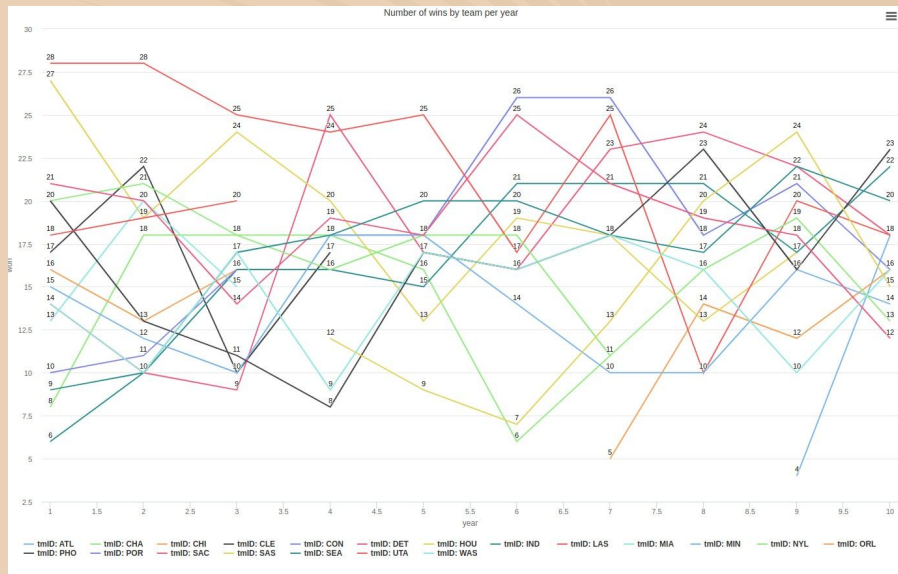
Coaches



In this dataset, all the data was organized and normalized, so there was no need to deal with outliers and missing values. By analyzing this dataset, we noticed that some of the attributes didn't seem relevant to the predictions we wanted to make. The remaining attributes bring value to the predictions in the sense that they are based on each coach's number of wins and losses, and are therefore directly related to the likelihood of their team qualifying or not for the playoffs.

Data understanding (cont.)

Teams

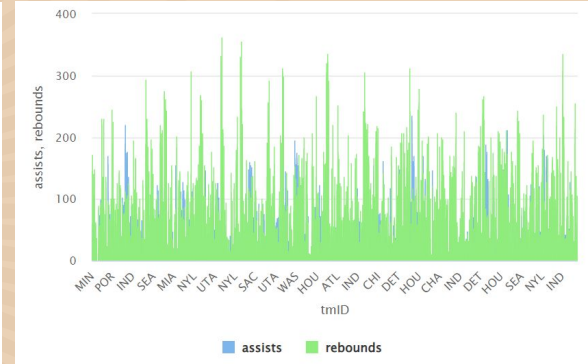
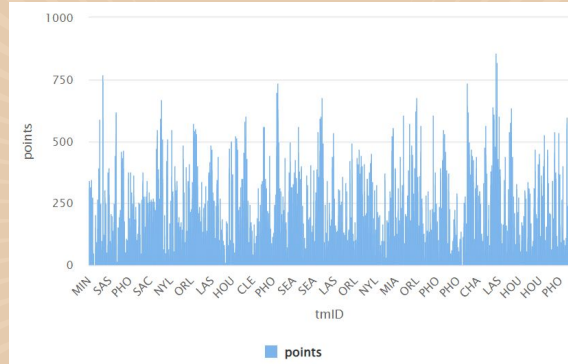
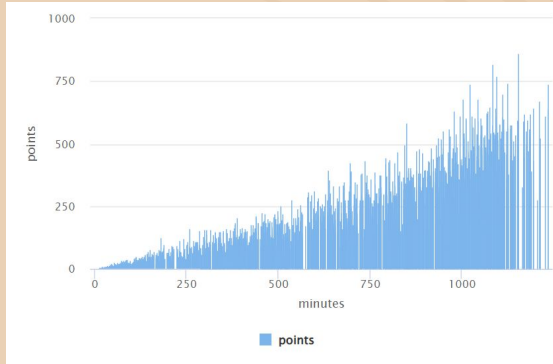


The most relevant attributes in the team's dataset are the number of wins and losses and all the games' statistics, such as points made, steals, blocks, etc. during the season. Also, since they all had the same league and division's ID, the same value for seed for all the teams, and the conference and the franchise the teams have is not relevant, we decided to eliminate those attributes, since they don't give us any relevant information.

Data understanding (cont.)



Players_teams

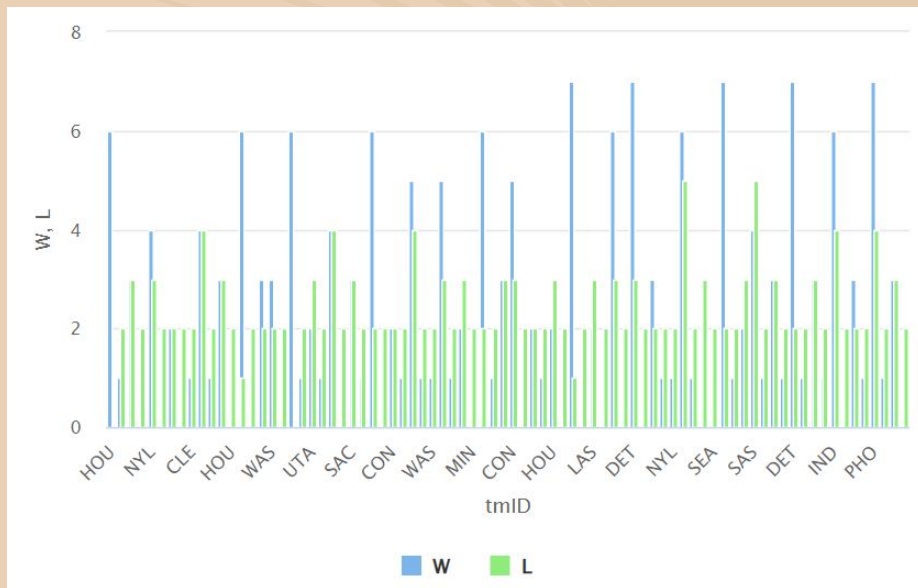


By analyzing this dataset we were able to ascertain that there are many variables related to game statistics that could be combined into just one at a later stage. We were also able to verify that the teams are balanced, as there is no team that has a much higher number of points than the others. We also see that some attributes are apparently related to points, since the more minutes you play, the more rebounds and assists you have, the more points the team has. We also noticed that from a certain year onwards, 34 games were played instead of 32, so the sum of the number of wins and losses wasn't always consistent.



Data understanding (cont.)

Teams_post



When we analyzed this dataset, we noticed that all the information was complete, so we didn't need to process the data at a later stage. Some of the attributes are not relevant, since as with other datasets the information is the same for all the rows. The most relevant attributes are the team's number of wins and losses.

Data Preparation

- **Data cleaning;**
- **Data transformation;**
- **Feature selection;**
- **Feature engineering.**



Data cleaning

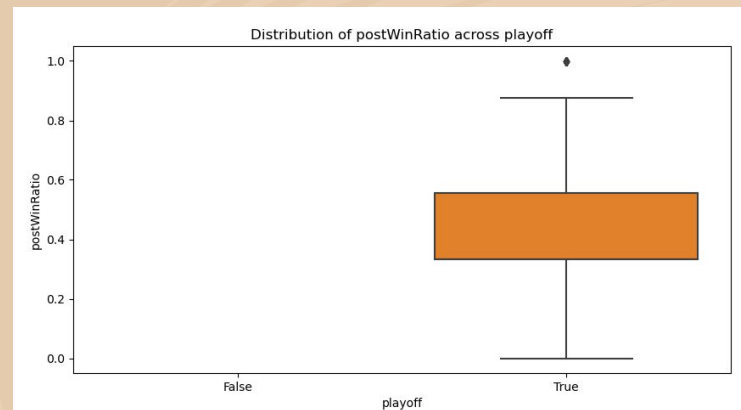
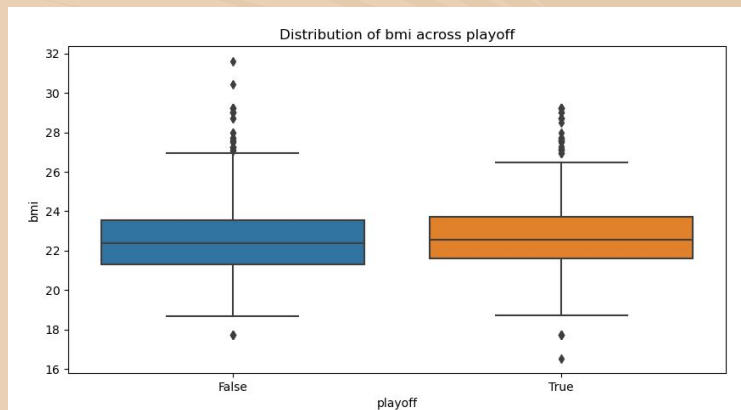
- So we divided up the datasets among us. The following had to be dealt with:
 - eliminate duplicates;
 - checking for undefined values and/or outliers;
 - eliminating rows that were not used or related to any other table, as they had all the values missing.
- Tables affected:
 - **players** (birthDate and deathDate, weights and heights);
 - **teams** (lgID and confID, GP/min., GP/won/lost/ homeW and awayW, homeL and awayL), similar to teams_post;
 - **coaches** (lgID, stint, post_wins).

Data transformation

- Due to a lot of information, we decided to simplify some attributes, in order to minimize the number of variables needed.
- **players:**
 - height and weight → bmi;
 - offensive and defensive statistics → player_score;
- **teams:**
 - offensive and defensive statistics → teams_score;
 - playoffs, firstRound, semis and finals are now booleans (instead of 0/1);
- **awards_players:**
 - name of the award of each players → number of awards won by each player per year;

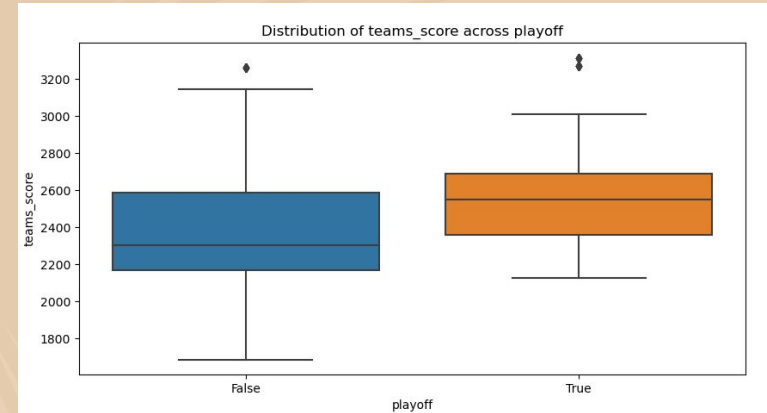
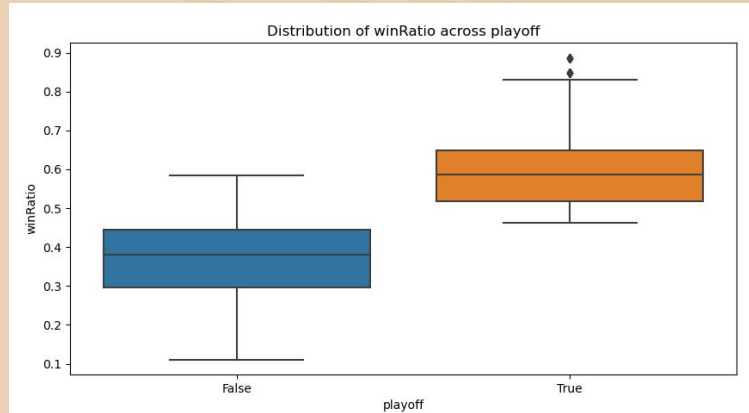
Result (data transformation + cleaning): 100 attributes → 22 attributes.

Feature selection (binaries-numericals)- boxplots



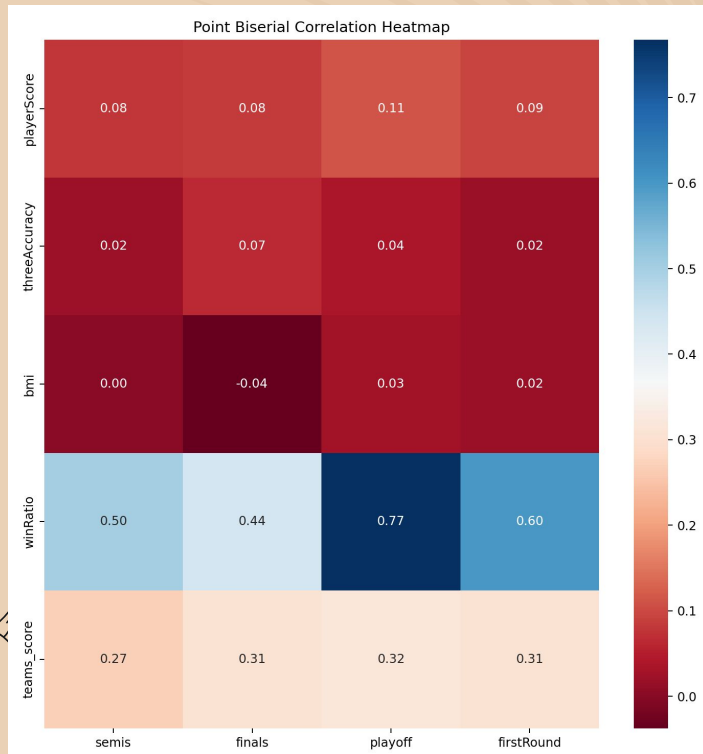
- Bmi doesn't affect whether the team makes it to the playoff or not and there are a lot of the outliers in this field.
- We had to treat the post-season-related fields with care, since they can cause data leakage!

Feature selection (binaries-numericals)- boxplots



- The winRatio and teams_score with a similar behavior: the greatest the value, the more likely is to that team be qualified in the playoff.

Feature selection (numericals-binaries heatmap)



- As we concluded before, the winRatio is highly correlated with the binaries fields;
- On the other hand, the numerical values seem not to have any significant relation with the binaries fields, including the target value.

Feature selection (numericals-numericals)

Attribut...	awards	playerS...	postPla...	dq	PostDQ	threeA...	postThr...	bmi	teams_...	winRatio	postWi...
awards	1	0.288	0.176	0.243	0.187	0.059	0.022	-0.110	0.084	0.137	0.066
playerSc...	0.288	1	0.270	0.220	0.101	0.233	0.015	0.016	0.134	0.095	0.046
postPlay...	0.176	0.270	1	0.114	0.063	0.052	0.369	-0.076	0.076	0.023	0.102
dq	0.243	0.220	0.114	1	0.155	-0.027	0.032	0.074	0.014	0.011	0.018
PostDQ	0.187	0.101	0.063	0.155	1	0.005	0.076	0.027	0.069	0.128	0.008
threeAcc...	0.059	0.233	0.052	-0.027	0.005	1	0.074	-0.069	0.045	0.048	0.045
postThre...	0.022	0.015	0.369	0.032	0.076	0.074	1	-0.012	0.022	-0.043	0.004
bmi	-0.110	0.016	-0.076	0.074	0.027	-0.069	-0.012	1	0.023	0.046	0.019
teams_s...	0.084	0.134	0.076	0.014	0.069	0.045	0.022	0.023	1	0.429	0.325
winRatio	0.137	0.095	0.023	0.011	0.128	0.048	-0.043	0.046	0.429	1	0.604
postWin...	0.066	0.046	0.102	0.018	0.008	0.045	0.004	0.019	0.325	0.604	1

- By analyzing the correlation between numerical attributes, we found that the ones we selected to remain in the dataset are no longer highly correlated with each other;
- Therefore, most of the numerical attributes we have provide information that represents new and different information from the rest.

Feature selection (ordinals)

Attributes	rank ↑
winRatio	-0.902
postWinRatio	-0.593
teams_score	-0.427
PostDQ	-0.141
awards	-0.131
playerScore	-0.083
postPlayerScore	-0.061
bmi	-0.044
threeAccuracy	-0.038
postThreeAccuracy	-0.007
dq	0.004
rank	1

- WinRatio, postWinRatio and teams_score are inversely correlated: the lower (better) the rank, the greater the ratio is, as expected;
- All other values almost don't have any correlation with the ordinal one (rank);
- This might indicate that the rank field might not have that relevance as it could have, since it doesn't relate with very variables.

Feature selection (Conclusions)

- **Binaries:** We decided to keep them all;
- **Categoricals:** Due to the low correlation between other relevant variables, we decided to discard the pos and awards attributes;
- **Numericals:** Since the numerical attributes are not highly correlated between themselves, we will remove those that are not very significant to the target variable;
- **Ordinals:** The only ordinal attribute 'rank' is highly correlated to some numerical attributes, so we will decided to remove it.

Feature engineering

- **Player/team score formula:**
$$(P + 0.4FGM - 0.7FGA - 0.4(FTA - FTM) + 0.7ORB + 0.3DRB + S + 0.7A + 0.7B - 0.4F - T) / \text{MINUTES}$$
- **Three-pointer accuracy:**
$$\text{threeMade} / (\text{threeMade} + \text{threeAttempted})$$
- **Body mass index:**
$$(\text{weight} / \text{height}^2) * 703$$



New evaluations

- In this iteration we decided to explore not only the Decision Tree, but also 2 new models: the Random Forest and the K-NN;
- We made several experiments with them, varying their hyperparameters and measure their performance, using the accuracy, performance and F1-measure for each year;
- After that, we made the average statistics for each model.

New evaluations (Average statistics)

- Decision Tree (default values):

```
The overall accuracy of the model is: 0.92228734018418  
The overall precision of the model is: 0.8905109489051095  
The overall F-measure of the model is: 0.9061206416773379
```

- K-NN (n_neighbors = 5, weights = uniform, algorithm = auto)

```
The overall accuracy of the model is: 0.6178980361001167  
The overall precision of the model is: 0.5826771653543307  
The overall F-measure of the model is: 0.5997709693098044
```

- Random Forest (default values):

```
The overall accuracy of the model is: 0.9613336075865349  
The overall precision of the model is: 0.8905109489051095  
The overall F-measure of the model is: 0.9245679936852951
```

Conclusions (2nd iteration)

- Decision Tree and Random Forest with the best results, KNN with worse performance than the first Decision Tree used;
- DT and Random Forest have the best results since they have the best characteristics for the work (feature importance);
- Not the case for the KNN, which can indicate this model didn't adapted to the dataset that well, or the data wasn't being used correctly for this model.



Final iteration

Improvements

- Split the data by each conference, since 4 from each conference go to the playoffs;
- Group data by player and by team would also be a wise decision, since it reduces the loss of information;
- We moved the data of a year to the following, so that we can predict the current year, using the past-related data. This way, we prevented the model from having data leakage and has more data to predict a year, as well.

Strategies


- We decided to predict with the Decision Tree Model, using the sliding window technique of 2 years.
- Getting player stats from the previous year, to take into account changes in teams.
- Predict the probability of a team getting to the playoff, and choose the ones with the highest probability.
- As a result of the previous experiments, the Decision Tree Model will use the 'gini' criterion and have 3 layers as maximum depth

Training and Testing

- Training data: `data[year in range(year-window_size, year)]`;
- Testing data: `data[year]`;
- This have made in this way, and in alternative to the use of cross validation, so that we avoided to use mixed data and have further years in the training data, which would lead to a data leakage and make the model make incorrect predictions.





Experiments



After applying the strategies mentioned above, we tested them with a few different models to see if our results improved. Among them, we highlight the following:

- Decision Tree;
- Random Forest;
- Logistic Regression.
- K-NN

Of these experiments, the model that gave the best results was the Random Forest with the default parameters. The worst results were obtained with logistic regression.



Models

We made various models using the following classifiers:

- Decision Tree;
- Random Forest;
- Logistic Regression;
- K-NN;

We experimented with different parameters, but we decided to keep the default ones given to scikit-learn for predicting the playoffs.

Performance Measures

- Like we did in the previous iteration, we decided to measure the statistics, as well as the average measures.
- We've also measure the model's performance in terms of time. This could have more varying results if we used the Machine Learning model, since the used ones were pretty faster.
- Random Forest performance:
 - Eastern Conference:




```
Year: 10, Top 4 Teams: ['Connecticut Sun ', 'Indiana Fever ', 'New York Liberty ', 'Detroit Shock ']  
The accuracy for year 10 is: 0.7886178861788617  
The precision for year 10 is: 0.7425742574257426  
The F-measure for year 10 is: 0.7425742574257425  
The time of the execution is: 2.7430057525634766 milliseconds.  
Year: 10, Top 4 Teams: ['Washington Mystics ', 'Detroit Shock', 'Indiana Fever ', 'Connecticut Sun ']
```

- Western Conference and average performance (both conferences):

```
Year: 10, Top 4 Teams: ['Seattle Storm ', 'Los Angeles Sparks ', 'Phoenix Mercury ', 'Minnesota Lynx ']  
The accuracy for year 10 is: 0.8928571428571429  
The precision for year 10 is: 1.0  
The F-measure for year 10 is: 1.0  
The time of the execution is: 4.808187484741211 milliseconds.  
Year: 10, Top 4 Teams: ['Seattle Storm ', 'Los Angeles Sparks ', 'Phoenix Mercury ', 'Minnesota Lynx ']  
The average accuracy is: 0.9012291183662415  
The average precision is: 0.9436112619608544  
The average recall is: 0.9436112619608544  
The average F1-score is: 0.9436112619608544  
The average time is: 4.7839615080091695 milliseconds.
```



Final conclusions

- 
- In this project, we have learnt a lot of different key concepts and phases of the Machine Learning area, such as the data preparation, data processing, the evaluation of the models, etc.;
 - In each phase, we have understood what are the fundamental steps to be taken, which have lead to a good model conception;
 - In spite of the fact that we have encountered some difficulties, especially in the data understanding and the data cleaning, as well as in the model selection due to the fact we have struggled in the merge of the datasets phase and to understand the fields too. However, we overcome the difficulties and have gained more knowledge in this learning process;
 - In the improvements made and as expected, the model had a good performance, even though the slight differences in both conference models' performances.
- 
- 

References

- [1]- WNBA Playoffs:
https://en.wikipedia.org/wiki/WNBA_playoffs, Wikipedia (consulted on 22/9/2023);
- Player Impact Estimate:
<https://www.nba.com/stats/help/glossary>, WNBA (consulted on 27/09/2023);
- Decision Tree Classifier:
<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>, Scikit Learn (consulted on 10/10/2023);
- K-NN Classifier:
<https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>, Analytics Vidya (consulted on 2/11/2023);
- Random Forest Classifier:
<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>, Scikit Learn (consulted on 2/11/2023).