U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

WNBA Playoffs Machine Learning Predictions

M.EIC- Machine Learning

Dinis Sousa (up202006303) - IF: 33%
João Matos (up202006280) - IF: 33%
João Pinheiro (up202008133) - IF: 33%

Table of contents

01

Domain description

Domain Description

Aimed at forecasting the **playoff access** results for teams in the Women's National Basketball Association (WNBA), we were provided with **10 years of information** regarding team statistics, player performance metrics and various contextual factors.

The WNBA playoff access is determined based on the regular season standings. At the end of the regular season, the **top 8 teams** from the Eastern and Western Conferences combined qualify for the playoffs and compete for the WNBA championship. Teams are **ranked by their win-loss records**, with tiebreakers applied as per specific league rules.

02

Data analysis

Data analysis - main takeaways

The exploratory data analysis was executed using **exploratory tools** of the *pandas* library (checking for nulls, printing unique values, etc.) as well as by plotting different **graphs** (box plots, histograms, correlation maps, etc.). The main conclusions are:

- There are columns with **no variability** like leagueID and some statistics;
- There are **dead players**;
- There are 338 players that have **not played in the seasons** provided;
- There is the need to do null value uniformization, as there are some columns with **empty strings**, others with **default 0** values and other **values that represent null**;
- Some columns have **binary** (confederationID, playoff) or **ternary** (finals, semis, firstRound) values;
- The **number of games played** by each team **differs**, so they are not directly comparable. Win percentage should be used;
- There are players with **no position** and **no college assigned**;

Data analysis - main takeaways

- In terms of win percentage, it seems like a competitive league, with **more than half** of the teams having a **win percentage of 50% or more**, taking advantage of the worst teams. There is also just one team below 40% of wins;
- There are a lot of **highly correlated variables**;
- Player attributes **do not follow Gaussian Distributions** ;
- The "Post*" attributes are the **most correlated** with the "Playoff" variable;
- There are teams that are **no longer playing**;
- A team that **wins a game in the playoffs**, will most probably **qualify for them again next year**.

03

Predictive Data Mining

Problem Definition

Objective

Develop a **machine learning model** that accurately **predicts and forecasts** the teams likely to secure **playoff access** in the WNBA based on team statistics, player performances, and other **data from previous years**.

Specifics

We want to be able to correctly predict if a team qualifies for the playoffs, which is indicated in the ***playoff* column** of the **table *Teams***. For that we have to use the information present in that table and **combine** it with features from **other tables**. The final dataset must be properly cleaned, filtered and prepared with relevant attributes. Data from recent years may have a bigger weight.

Data preparation

Null uniformization

There were several columns with **null values**, that were **not identified as such**. Dates with 00-00-00, integer values with 0 and string values with “” (empty string) were all transformed in None.

Remove players that have not played in the years given

Players that had not played in any game in the given years of the dataset were **removed** from the data.

Transform binary attributes

Several different **attributes were binary**, but were **represented by strings**. We **transformed** them into **actual binary attributes**.

Normalization and Standardization

Normalized attributes that followed a Normal distribution. **Standardize attributes** that didn’t, linearly from 0 to 1.

Data preparation

Feature engineering

Created new attributes for both the **Teams**: **Win rates**, **Total** season **stats** (regular season + playoffs), **Average** per game **stats**; and **Players**: **Durability Ratio**, **Point Ratio** and **Position specific metrics**.

Remove most correlated attributes

Removed the attributes that were the **most correlated** with each other. We used a threshold of 0.95 as the maximum correlation between two attributes.

Merge information from tables

We **merged data** from the players table with the teams table, to get more insights for the data modelling.

Experimental setup

Choosing a prepared dataset

We ended up settling on a dataset with **linearly normalized** non gaussian attributes, **standardized** gaussian attributes that uses **merged data of the average** of some player attributes.

Using different classifiers

We experimented with different classifiers: Random Forest, KNN, Gradient Boosting, XGBoost, SVM and Logistic Regression.

Forcing the best 8 to pass

To guarantee exactly 8 teams pass, the teams with the 8 highest probabilities (4 for each confederation) will pass.

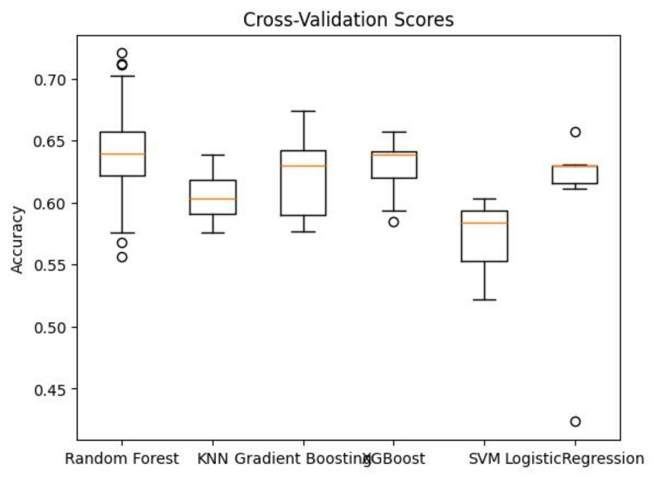
Rolling window

For each team in a year, we take data from ***N*** previous years to do a **weighted mean.** This can be either, getting the **average of the stats of all players in a team for each previous year** or calculating the **averages for each of the players in the team in the test year using data from previous years**.

The rolling window dataset is used to train a model (without the test year). Finally we test the accuracy of the model against the testing year.

Results

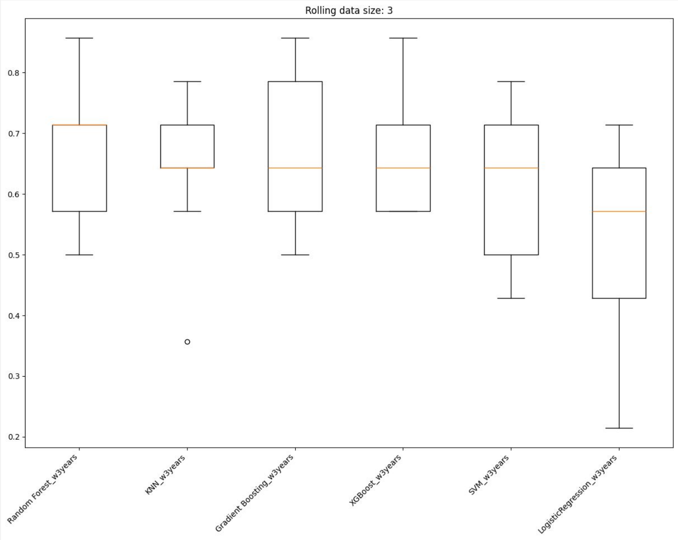
- We first evaluate the models using the rolling window that uses the team data from the past years.
- Below we have box plots of running grid search with cross-validation .
- The model that achieves the best accuracy result and the best average result is the Random Forest Classifier.



Cross validation scores: Using a rolling window that looks at the teams results in past years for different models

Results

- Then we used a rolling window that takes into account not how the team performed in the past years, but how their players performed in the past years. This method was more effective and provided better results
- Below we have box plots of the accuracy scores of the rolling window using only 3 years, having tested multiple rolling window sizes before.
- We did consider creating an *ensemble*, but as the random forest model usually had the best results we feared it would only diminish the final accuracy score.

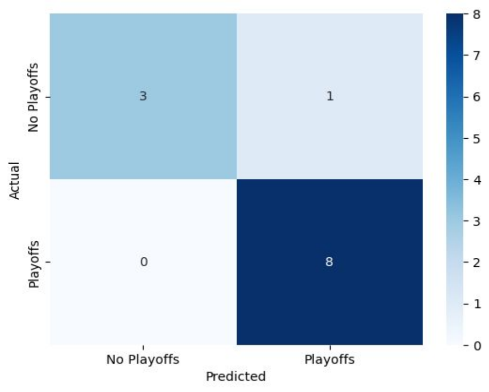


Rolling window scores: Using a rolling window that uses the 3 previous years, with the data from the players

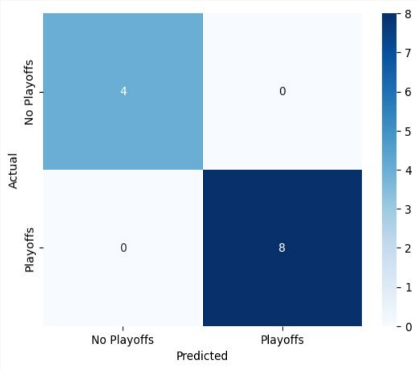
Results

- We then selected the best performing model, which was random forest, and trained it with the competition data, using the rolling that uses players past year data
- We were able to achieve 100% accuracy, when comparing to the real data, with our model trained with the past 10 years data.

83% accuracy



Without force qualify



With force qualify 8 teams

100% accuracy

Conclusions

Current state

In depth understanding and preparation of the data, with a deep understanding of the data and our objective. On top of that, we tested different models to produce predictions.

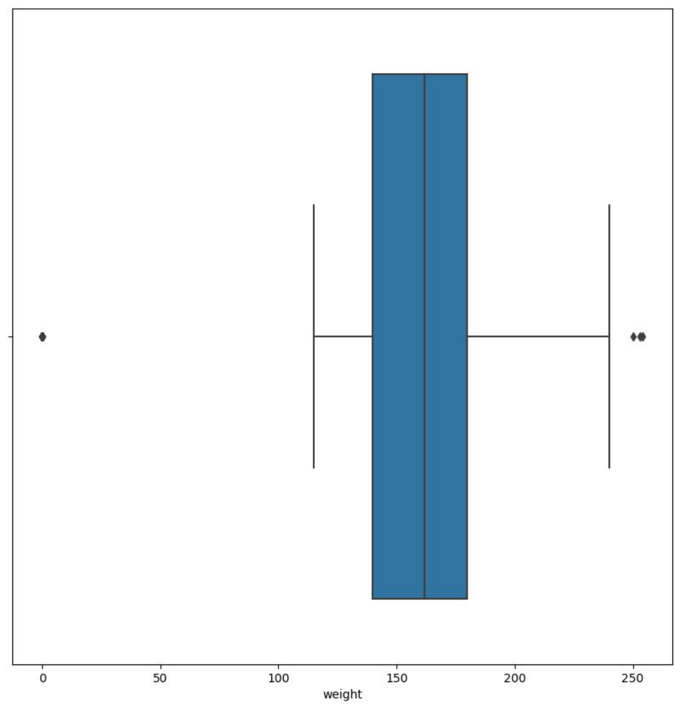
Current results

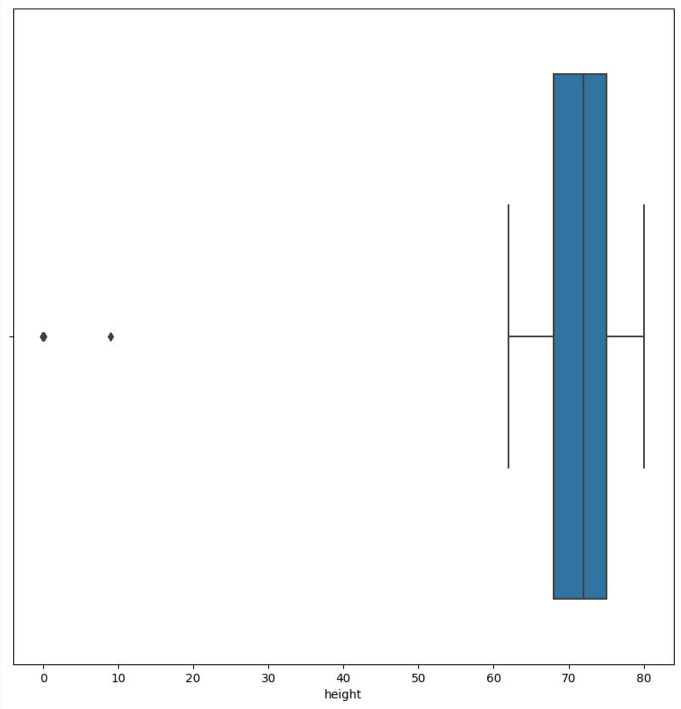
We are pleased with the current results. The accuracy scores seem to have stabilized and we feel our rolling window is robust, reliable and credible.

Future Work

- Explore the integration of ensemble methods to combine predictions from multiple models for improved accuracy and stability.
- Investigate the potential benefits of combining different types of models (e.g., stacking or bagging).
- Explore methods to provide meaningful explanations for the model's predictions. We used Random Forest and so we can analyze the trees.

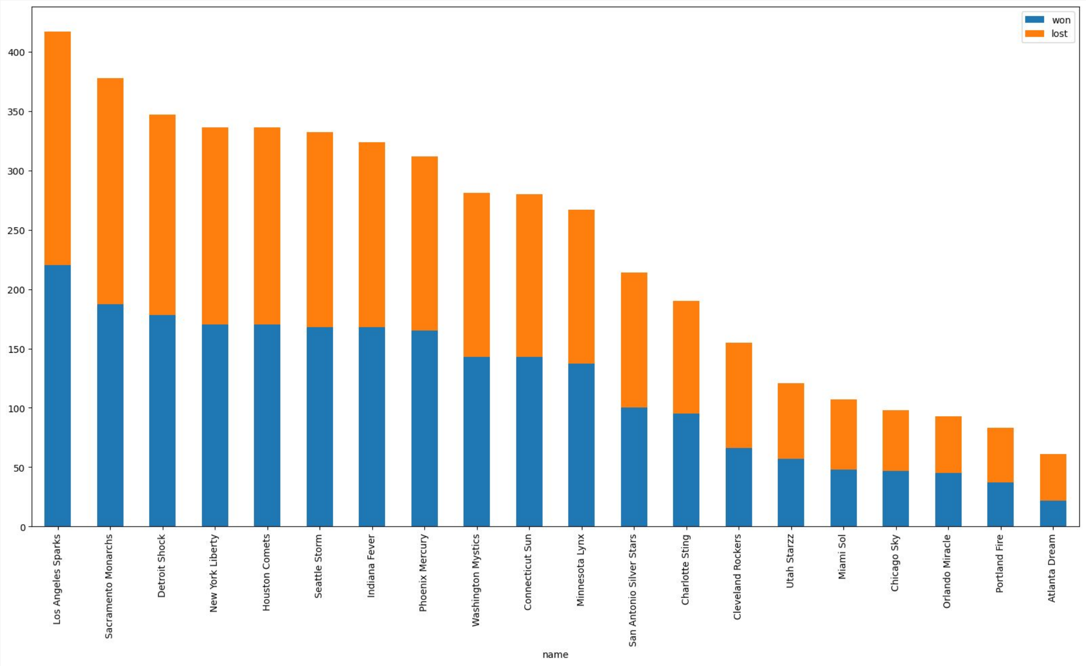
Annexes





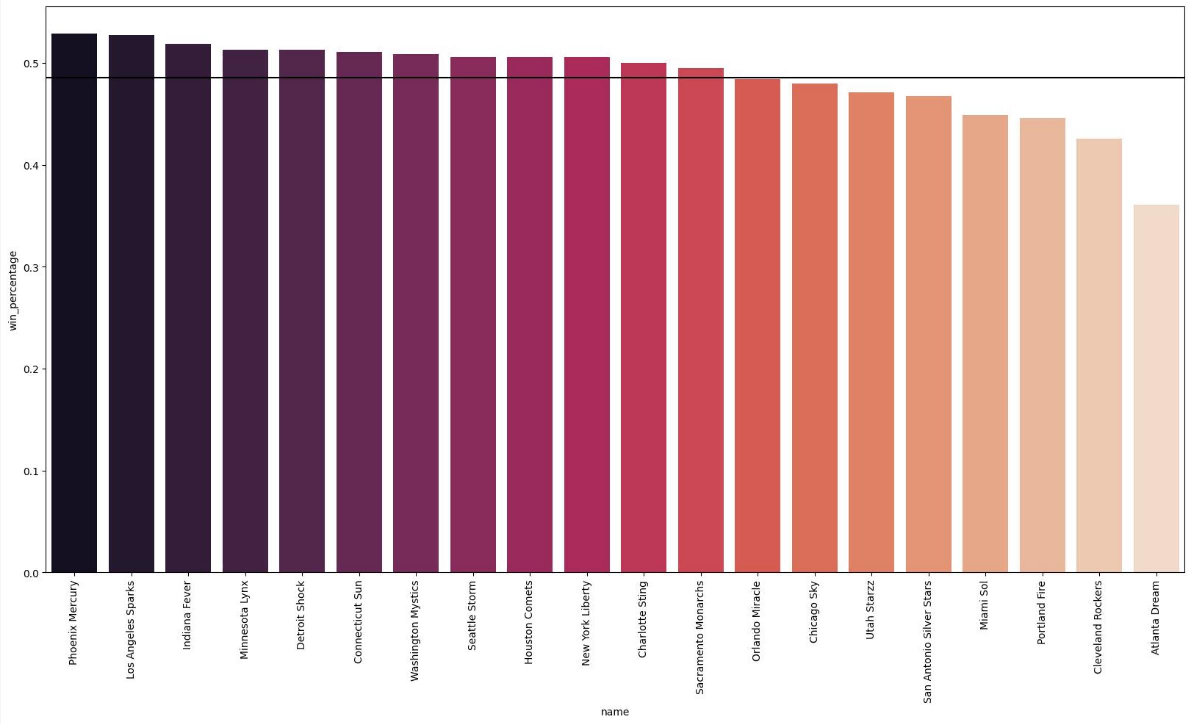
Distribution of players weights and heights

Annexes



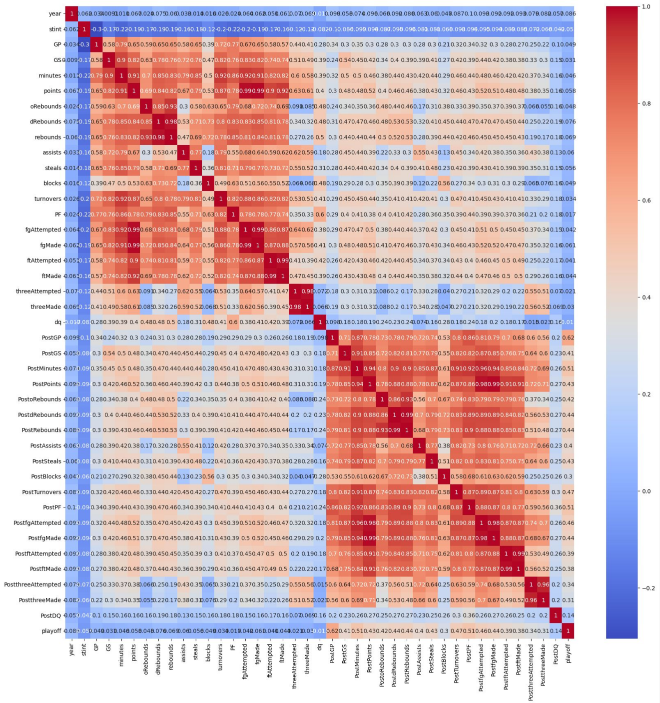
Total wins and defeats per team

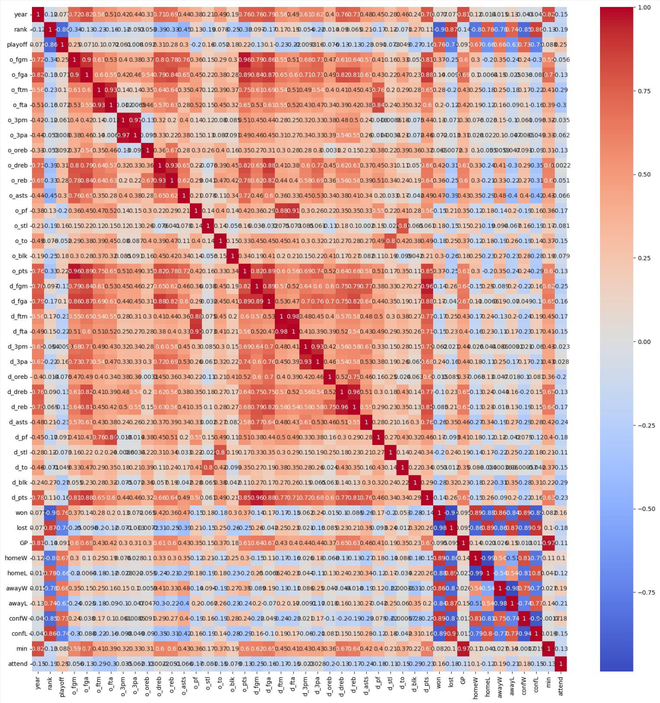
Annexes



Win rate per team

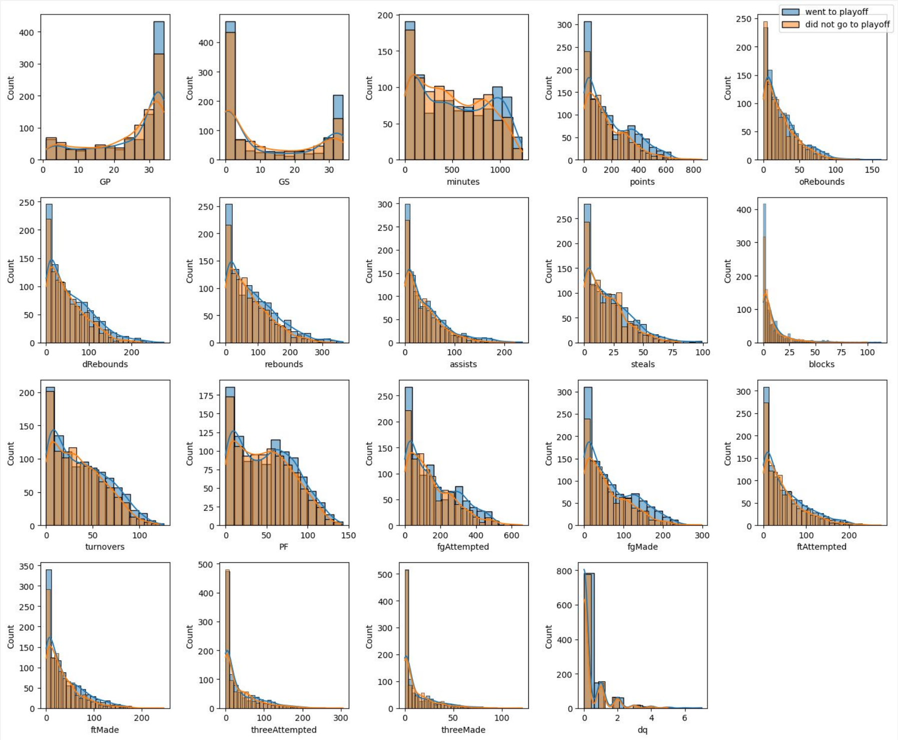
Annexes





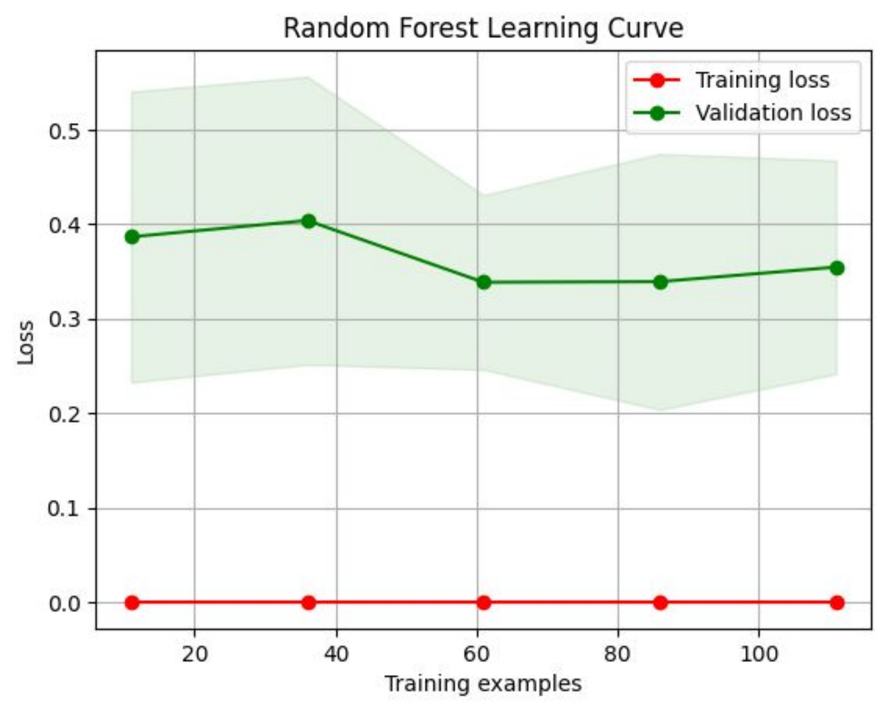
Correlation matrix for the players and teams table

Annexes

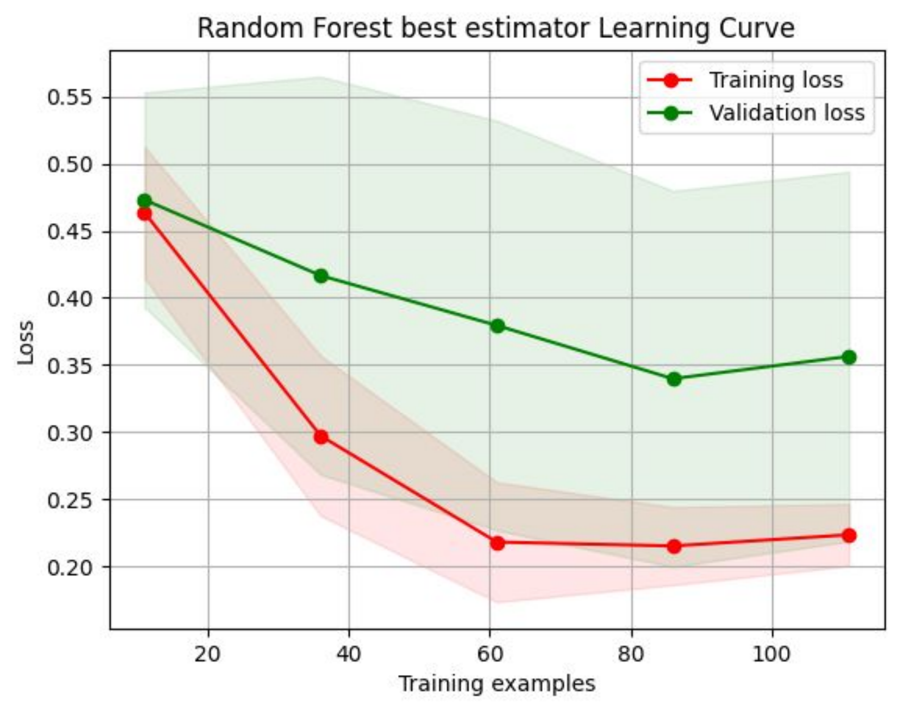


Attribute distribution: Went to playoff vs Did not went to playoff

Annexes

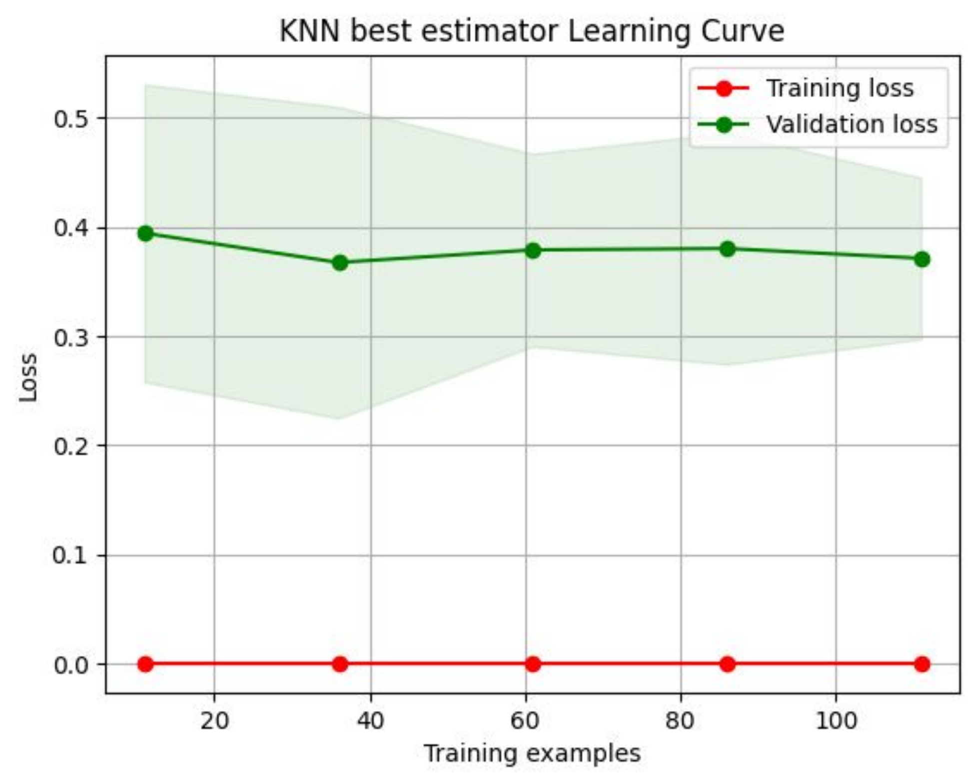


Random Forest Learning Curve
Training loss
Validation loss
Loss
0.5
0.4
0.3
0.2
0.1
0.0
20
40
60
80
100
Training examples



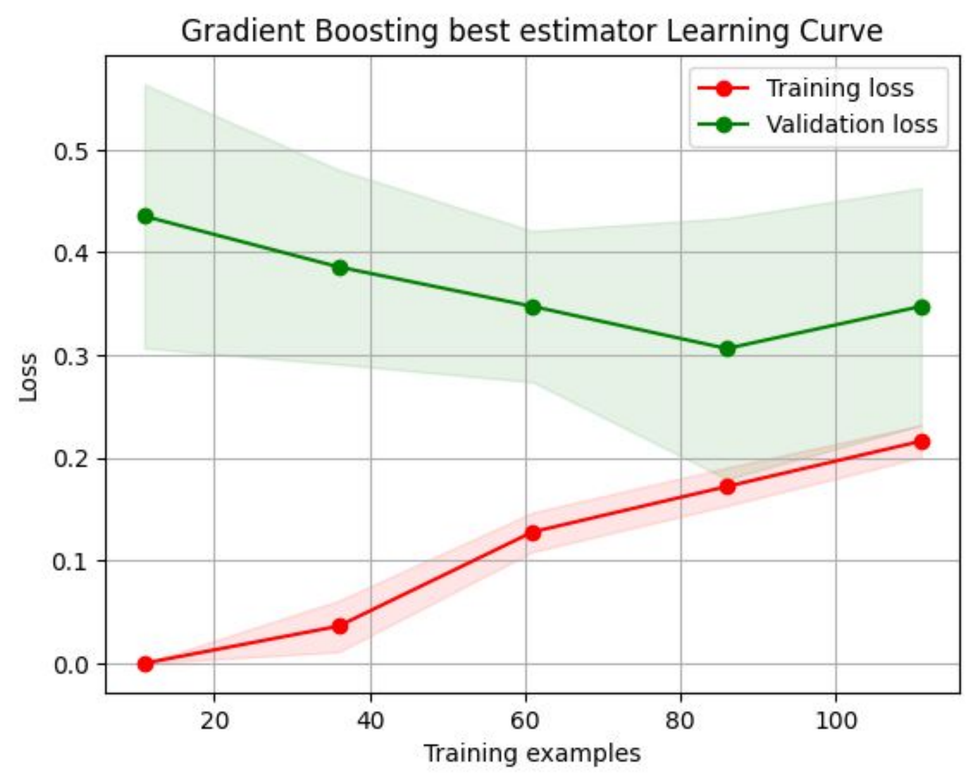
Random Forest best estimator Learning Curve
Training loss
Validation loss
Loss
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
20
40
60
80
100
Training examples

Annexes

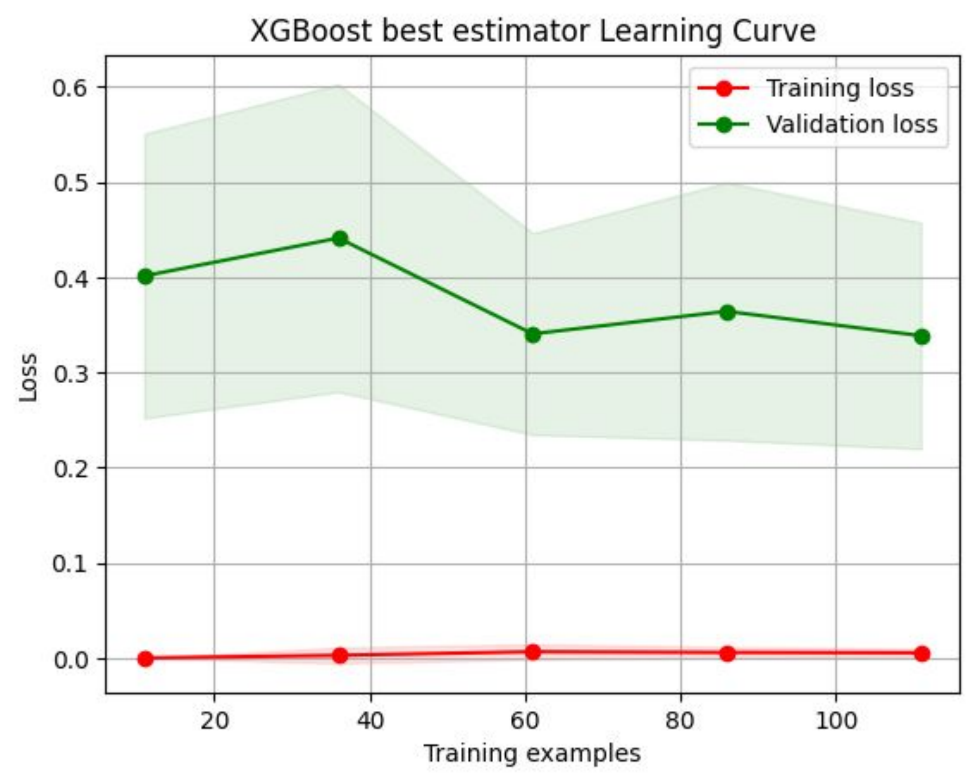


KNN best estimator Learning Curve
Training loss
Validation loss
Loss
0.0
0.1
0.2
0.3
0.4
0.5
20
40
60
80
100
Training examples

Annexes

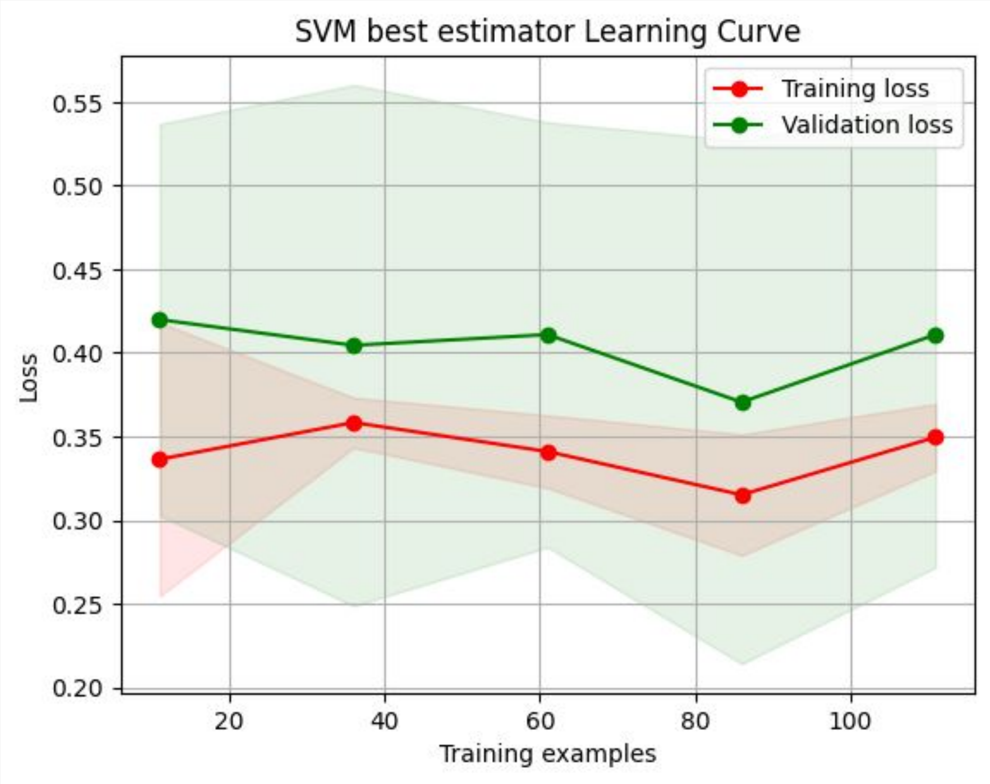

Gradient Boosting best estimator Learning Curve
Training loss
Validation loss
0.5
0.4
0.3
0.2
0.1
0.0
Loss
20
40
60
80
100
Training examples

Annexes



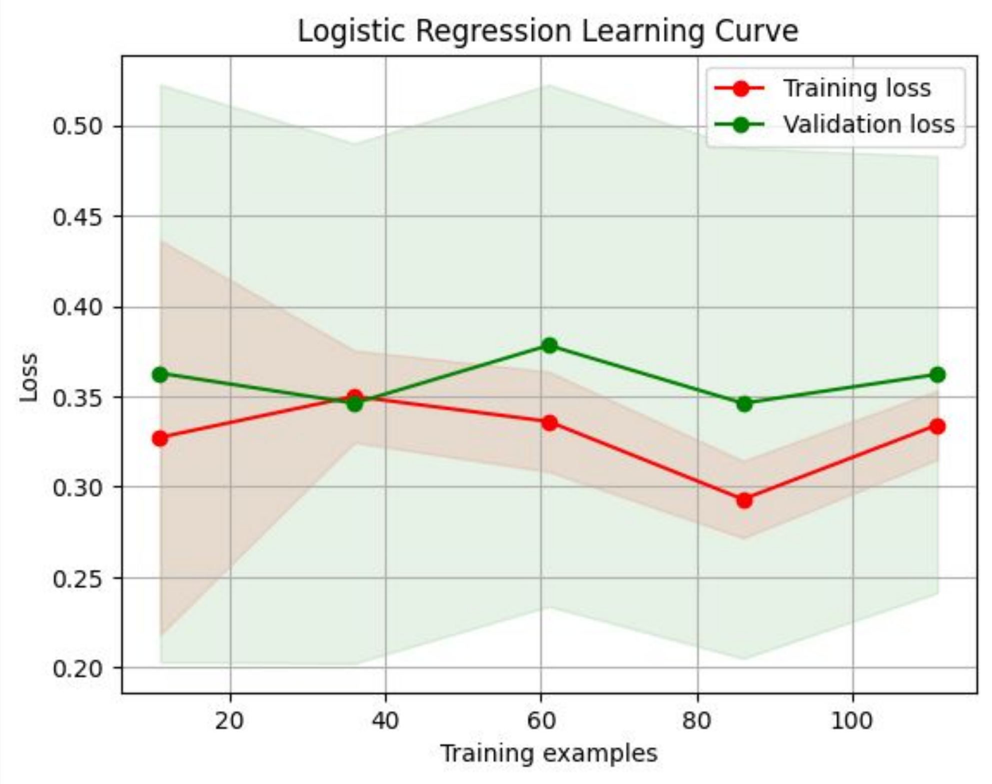
XGBoost best estimator Learning Curve
Training loss
Validation loss
Loss
0.6
0.5
0.4
0.3
0.2
0.1
0.0
20
40
60
80
100
Training examples

Annexes

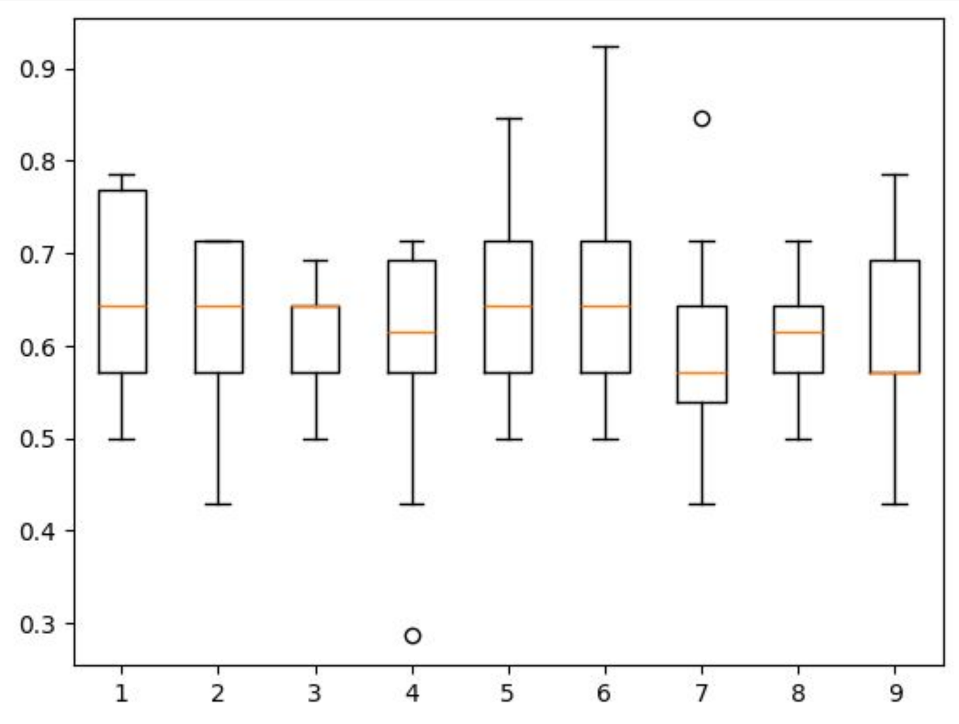


SVM best estimator Learning Curve
Training loss
Validation loss
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
Loss
20
40
60
80
100
Training examples

Annexes

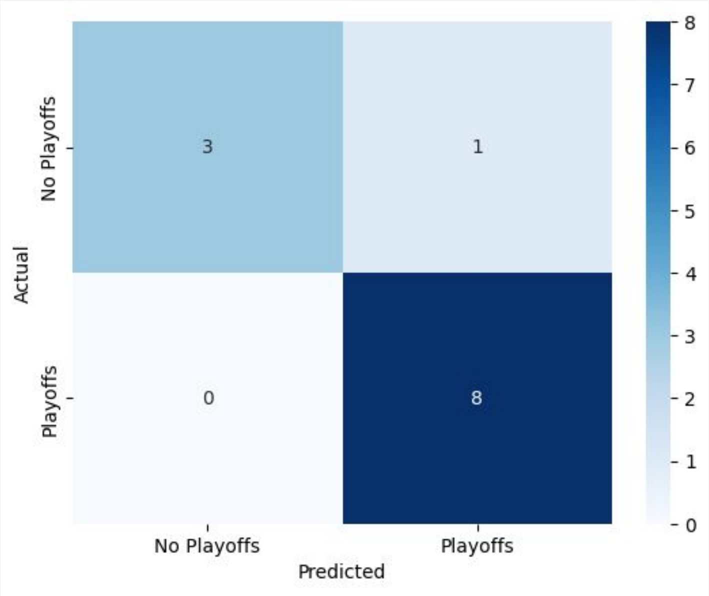

Logistic Regression Learning Curve
Training loss
Validation loss
0.50
0.45
0.40
0.35
0.30
0.25
0.20
Loss
20
40
60
80
100
Training examples

Annexes

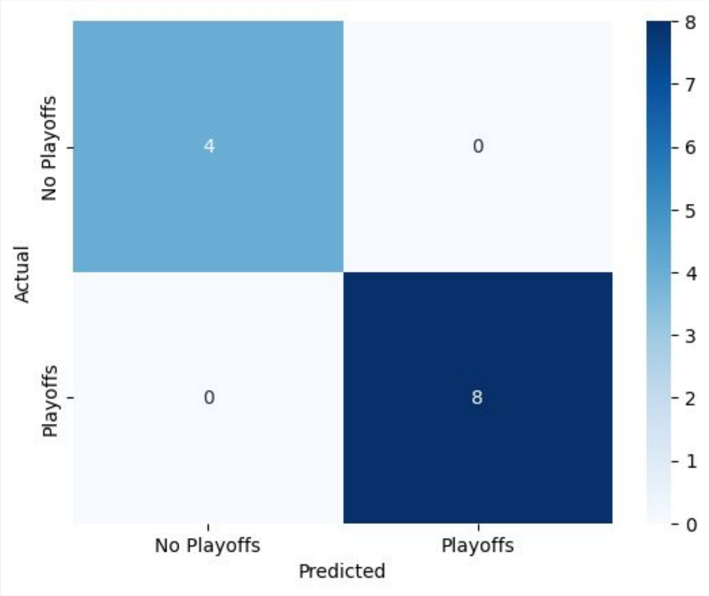


Comparing different rolling window sizes accuracy

Annexes



Without force qualify



With force qualify 8 teams

Comparing our prediction results with the year 11 real results

Annexes

All the code used for this assignment is in: https://github.com/jogp10/WnbaPlayoffsPredicting