Data Mining case study: WNBA

Faculty of Engineering of University of Porto

M.EIC

Machine Learning

Group 52
António Rego — 202108666
Pedro Lima — 202108806
Pedro Januário — 202108768

Brian Babineau/NBAE via Getty Images

Domain description

Our work focuses on a dataset consisting of information about Women’s NBA teams, their players, coaches, awards and performance metrics throughout ten seasons.

One of the main aspects of the domain relates to whether a given team qualified to a playoff in the end of the first part of a given season.

Teams that qualify for the playoff then compete against each other in the road for the trophy.

Exploratory data analysis

For this purpose we utilized a dataset that has the following tables:

- **teams** - provides some general information about each team's statistics in each year, such as won matches, points scored, etc.
- **players** - has some general information about each player, such as height, weight, etc.
- **players_teams** - details statistics similar to the teams table, but regarding and individual to each player.
- **awards_players** - this table states the performance awards won in each year and by which players or coaches.
- **coaches** - this table has each team's coach in a given year and their stats, such as games participated in and of those which were wins or losses.
- **series_post** - describes the series' results.
- **teams_post** - describes the results of each team at the post-season.

Most of the work will be done in the **teams** table since it contains the feature for which we are going to train, the column **"playoff"**, which defines if that team made it to the playoffs that year.

Of course, all other tables will be utilized to reduce the prediction error without losing any possibly relevant features.

Exploratory data analysis

Past some basic sanity checks, such as removing NaN values and getting rid of duplicate entries, which weren't present in the dataset, some preparations were made in order to reduce the future training time and simplify the massive dataset.

The dataset was also pruned for outliers, but there weren't many, especially considering how most of what was considered "an outlier" was bad or great performances.

The first step was to do feature extraction.

Exploratory data analysis

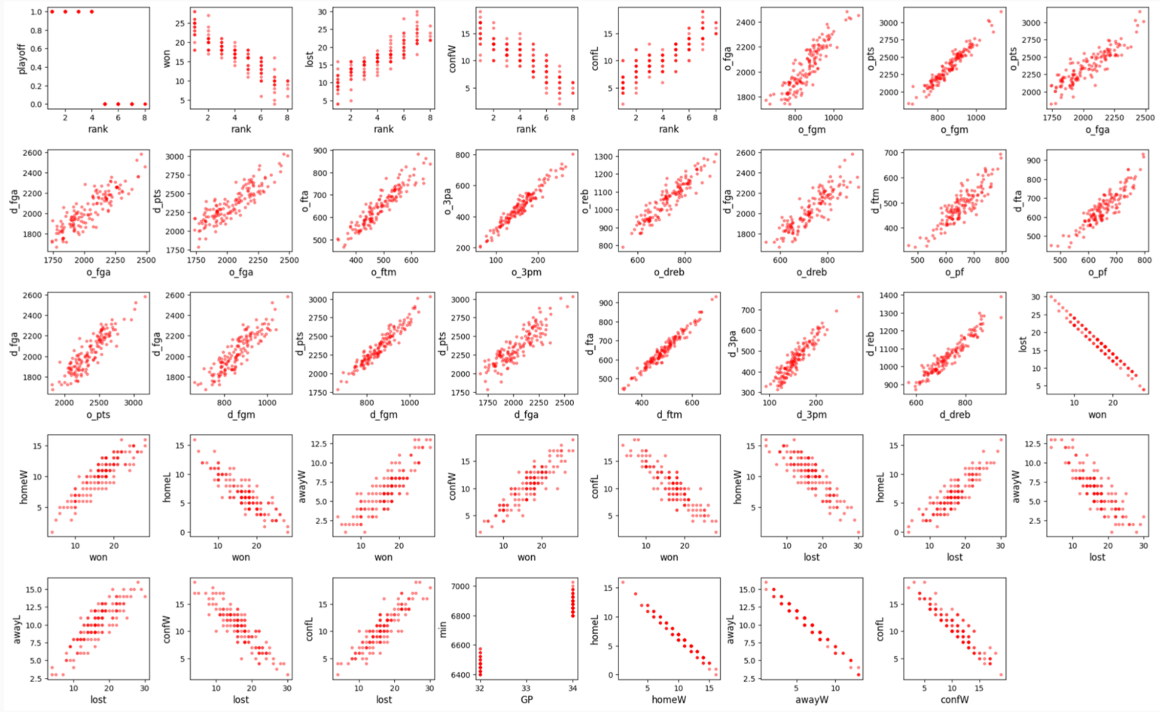
For that purpose, we checked the correlation between features in order to disambiguate the dataset.

Most tables did not have many highly correlated features due to a reduced amount of columns, but the **teams** table had plenty.

Most of these high correlations were in between features like **“field goals attempted”** and **“free throws made”**. For these cases, a new feature **“free throw rate**” was created, which is just the result of the **“made”** feature divided by the **“attempted”** feature. Then, we got rid of the **“made”** feature as it was already reflected in the percent rate. We did not, however, get rid of the **“attempted”** feature as to not lose any possibly relevant information, since without this value, the percent doesn’t mean much. This logic was also applied to columns that followed the logic of **“won”**, **“lost”**, and **“played”**, the **“won”** having been replaced for the rate and **“lost”** having been done away with, because it would be the inverse of the resulting percent.

Another common case was such of **“o_oreb”** or **“o_dreb”** and **“o_reb”**, the last being the sum of the first two. So, we decided to get rid of all the columns which were sums of others.

A very similar logic to here is applied to the player tables, since the statistics are similar, but at a lower level.



Predictive Data Mining problem

Problem definition

The problem addressed by the team focuses on achieving ways to, based on information about a given team's performance in a given year, predicting whether that team qualified for the playoff in the following season.

The aim is to obtain a probability for the qualification to happen.

The predictions are evaluated according to the sum of the errors of the predicted probabilities.

Data preparation

The general idea is that a team's statistics in one year will be used to predict the next year's results, thus, some changes had to be made. For this purpose we define three types of data:

- **Yearly** - most data is of this type, for example, the points a team scored in one year is only relevant for that year.
- **Continuous (Instantaneous)** - this data type grows over the years, for example, the awards a player has earned in a year should be the sum of all awards the player has earned in that year and the years before.
- **Continuous (Predictive)** - this data type is exactly the same as the instantaneous version, however it is how it is associated with the year which changes, this type associates with a year the known data in that year, but utilizes some information from the following year.

The **awards_players** table was treated as a **continuous (instantaneous)** type, so in one year any player or coach is associated with all the awards that person has earned up to that year.

The **coaches** table would have to be associated with the **teams** table, but it would not be as simple as just that. The statistics of a **coach** are treated as a **continuous (predictive)** type. In a year **X** we associate a team with their coach in year **X+1** but with the coach's statistics in year **X**. This because we predict year **X+1** and we want to use information from year **X** that is still relevant in year **X+1**, and having another coach associated which would not come to play in the following year would be information which is not valid or useful since, of course, that coach did not play for that team the following year.

When there are multiple coaches in one year for a team, a weighted average based on the amount of games played by each coach that year is used.

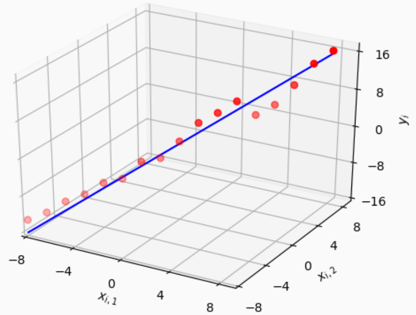
Data preparation

The idea was to create a rating for each player:

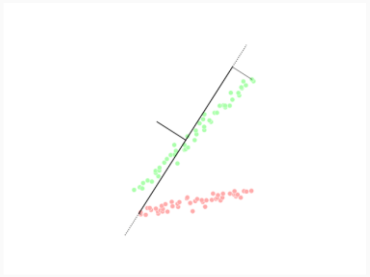
- Player Positions - Forward, Guard and Center (each player can have more than 1 position)
- For each **position** created a **Linear Regression** Model, to calculate the coefficients of each feature, based on wins of their team in that year.
- A player's rating corresponds to the average of values that results from linear regression model, for all previous years
- For each **team**, **year** and **position**, only included the features of the **top 3** players by rating and the **mean** rating of all players by position.

Even after all of this the table has 268 columns, which is to much, so we used PCA to reduce them.

In the end we will have 5 columns resulting from PCA, the rating of the top 3 players of each position and the mean rating of each position. The number of columns resulting from PCA has changed from submission to submission, and 5 was what gave us the best results.



Linear Regression by Qasim Chaudhari via Wireless Pi



PCA Projection Illustration by Amélia O. F. da S. via WikiMedia Commons

Experimental setup

We used GridSearchCV (from scikit-learn) to test several classification models. Each tested model was instructed to produce its classification as the probability of each team qualifying/not qualifying for the playoff in a given season.

We first defined a function that scores a set of predictions according to the intended evaluation criteria (sum of the errors).

Then, for each model, we defined a set of possible values for the models' hyper-parameters, in order to find out the best possible combinations (the best estimator) for each one of them.

Before executing GridSearchCV, we ran Sequential Feature Selection to find out a better feature subset.

Each execution (i.e., a moment where GridSearchCV tries to find the best estimator for a given model) presents the best parameter combination, as well as the total error and, given the number of seasons comprised in the given test set, the average error per season.

The train set was defined to include years 1 to 8, and the test set to include year 9.

The pair (estimator, feature selector) to be used against the query set were defined via the behaviour on the process described above.

Results (Approach behind best score on competition)

- Decision Tree Classifier
  - Best average error per season on train set: 0.91
  - Best average error per season on test set: 3.0
- Random Forest Classifier
  - Best average error per season on train set: 1.14
  - Best average error per season on test set: 6.09
- Histogram-based Gradient Boosting Classifier
  - Best average error per season on train set: 0.81
  - Best average error per season on test set: 7.15
- K-Nearest Neighbors Classifier
  - Best average error per season on train set: 1.05
  - Best average error per season on test set: 5.6

Results (Approach behind best score on competition)

- Neural Network Classifier
  - Best average error per season on train set: 0.91
  - Best average error per season on test set: 5.99

Thus, the estimator that “qualified” to be run against the query set was the Decision Tree Classifier.

Limitations — Future work — Conclusion

Limitations

It was necessary to simplify a lot features, mainly in players, which may lead to more inaccurate results. Also, the models return a probability of a team going to the playoffs, with no certainty.

Future Work

In the future, the training dataset should be expanded, and the evaluation metrics for player ratings should be further developed in order to obtain more diverse and accurate evaluations of player skill, one of the most integral parts of calculating a team's quality. A good example would be 2K's NBA 2K player ratings; getting to utilize their formulas and methods would certainly improve the model greatly as they are generally considered good estimations.

The dataset should also be split into two, one for the east conference and one for the west conference, since, before the playoffs, these teams only play against other teams in their conference, and not splitting them could mess with the model predictions.

Conclusion

This project highlights the power of machine learning in predicting WNBA playoff teams, by analyzing historical data and key metrics. Some models were developed to provide valuable insights into future playoff contenders.

Thank you

Faculty of Engineering of University of Porto

M.EIC

Machine Learning

Group 52

António Rego — 202108666

Pedro Lima — 202108806

Pedro Januário — 202108768

Individual Factor

All members contributed equally to the development of the work.

Annexes

“Diary” of approaches for each daily submission

The solution presented before corresponds to the best submission, which was in day 3.

Day 1: The estimator and feature selector to be used are defined via their behaviour against a training set with all the years with know label, so that there is no explicit test set, and we query the estimator right away. Models are chosen based on their performance in cross validated training test. There is no set used only for testing. The PCA from players returns 20 features.

Day 2: The Model Exploration notebook began to define the estimator, the parameters and execute feature selector, using year 9 as test, choosing the best model based on the error of year 9. In this day, we changed PCA from players to return 40 features.

Day 3: Changed PCA from players to return 5 features.

Day 4: Came back to the approach of day 1 but keeping the PCA returning 5 features. Here we didn't use the Decision Tree as it only returns binary results, and it never yields exactly 8 teams going to the playoffs.

Day 5: We did not deliver any submissions on that day.

Data preparation — Resulting table

The dataset that yielded the best results for training contained the following attributes:

- **year**: Year to which the data relates (corresponds to the year before the prediction).
- **tmID**: Team ID.
- **franchID**: Franchise ID.
- **confID**: Conference to which the team belongs; 0 for West Conference, 1 for East Conference.
- **rank**: Rank indicating where the team placed in the championship.
- **firstRound**: 1 if the team won the first round of playoffs, 0 otherwise.
- **semis**: 1 if the team won the semifinals, 0 otherwise.
- **finals**: 1 if the team won the finals, 0 otherwise.
- **o_fga**: Offensive field goal attempts by the team.
- **o_fta**: Offensive free throw attempts by the team.
- **o_3pa**: Offensive three-point attempts by the team.
- **o_oreb**: Offensive rebounds by the team.
- **o_dreb**: Defensive rebounds by the team.
- **o_asts**: Assists made by the team.
- **o_pf**: Personal fouls committed by the team.
- **o_stl**: Steals made by the team.

Data preparation — Resulting table (Cont.)

- **o_to**: Turnovers committed by the team.
- **o_blk**: Blocks made by the team.
- **d_fga**: Defensive field goal attempts by opponents.
- **d_fta**: Defensive free throw attempts by opponents.
- **d_3pa**: Defensive three-point attempts by opponents.
- **d_oreb**: Offensive rebounds by opponents.
- **d_dreb**: Defensive rebounds by opponents.
- **d_asts**: Assists made by opponents.
- **d_pf**: Personal fouls committed by opponents.
- **d_stl**: Steals made by opponents.
- **d_to**: Turnovers committed by opponents.
- **d_blk**: Blocks made by opponents.
- **tmORB**: Team offensive rebounds.
- **tmDRB**: Team defensive rebounds.
- **tmTRB**: Team total rebounds.
- **opptmORB**: Opponent team offensive rebounds.

Data preparation — Resulting table (Cont.)

- **opptmDRB**: Opponent team defensive rebounds.
- **opptmTRB**: Opponent team total rebounds.
- **won**: Rate of games won by the team.
- **confW**: Rate of conference wins by the team.
- **min**: Total minutes played by the team.
- **attend**: Attendance at games.
- **label**: Target variable indicating if the team made it to the playoffs (1 = Yes, 0 = No).
- **o_3prate**: Rate of baskets made from offensive three-point attempts.
- **d_3prate**: Rate of baskets made from defensive three-point attempts.
- **o_ftrate**: Rate of baskets made from offensive free throw attempts.
- **d_ftrate**: Rate of baskets made from defensive free throw attempts.
- **o_fgrate**: Rate of baskets made from offensive field goal attempts.
- **d_fgrate**: Rate of baskets made from defensive field goal attempts.
- **coach_won_rate**: Rate of games won by teams coached by this coach.
- **coach_total_games**: Total number of games coached by the coach.

Data preparation — Resulting table (Cont.)

- **cumulative_coach_awards**: Total number of awards the coach has won up to that year.
- **coach_games_played**: Number of games the coach was involved in during the year.
- **won_percentage_year_1**: Win percentage for the team in year 1 prior to the current year.
- **won_percentage_year_2**: Win percentage for the team in year 2 prior to the current year.
- **won_percentage_year_3**: Win percentage for the team in year 3 prior to the current year.
- **won_percentage_year_4**: Win percentage for the team in year 4 prior to the current year.
- **won_percentage_year_5**: Win percentage for the team in year 5 prior to the current year.
- **avg_win_last_3_years**: Average win percentage over the last 3 years.
- **AbstractPlayersFeature1**: Abstracted player feature resulting from PCA.
- **AbstractPlayersFeature2**: Abstracted player feature resulting from PCA.
- **AbstractPlayersFeature3**: Abstracted player feature resulting from PCA.
- **AbstractPlayersFeature4**: Abstracted player feature resulting from PCA.
- **AbstractPlayersFeature5**: Abstracted player feature resulting from PCA.
- **rating_max_C**: Rating of the best Center in the team.
- **rating_max_F**: Rating of the best Forward in the team.
- **rating_max_G**: Rating of the best Guard in the team.

Data preparation — Resulting table (Cont.)

- **rating_second_max_C**: Rating of the second-best Center in the team.
- **rating_second_max_F**: Rating of the second-best Forward in the team.
- **rating_second_max_G**: Rating of the second-best Guard in the team.
- **rating_third_max_C**: Rating of the third-best Center in the team.
- **rating_third_max_F**: Rating of the third-best Forward in the team.
- **rating_third_max_G**: Rating of the third-best Guard in the team.
- **mean_C**: Mean rating of all Centers in the team.
- **mean_F**: Mean rating of all Forwards in the team.
- **mean_G**: Mean rating of all Guards in the team.
- **no_new_players**: Number of new players added to the team for the year

Model exploration — Evaluation function

```
def get_error(estimator, X, labels):
 predictions = get_predictions(estimator, X)

 err = 0

 for i in range(len(predictions)):
   err += abs(predictions[i] - labels[i])

 return err

def scorer(estimator, X, y):
   return -get_error(estimator, X, y)      # negating so that greater errors mean
actually less score
```