playoffs\_two\_write\_up

Zuntue Pommerich

# Introduction

This report presents the methodologies and results of a comprehensive predictive model developed for a STOR 538: Sports Analytics class project. The objective of this project was to predict the outcomes of NBA games, specifically focusing on spread, total points, and total offensive rebounds (OREB). The data used in this project was obtained from Nathan Lauga's NBA dataset on Kaggle and supplemented with additional data from FiveThirtyEight. The report is divided into four sections, covering data information, predictive models for spread, total points, and OREB, respectively. In each section, various model types were investigated, and the most effective model was selected and developed for each outcome variable. The developed models were then used to predict game outcomes, demonstrating their predictive capabilities. All the code can be found at: <https://github.com/zuntue/playoffs_two>.

# Section 1: Data Information

1.1 Data Cleaning and Preparation

I used data from Nathan Lauga, a French data scientist who has graciously shared NBA data with the world through Kaggle. The data was last updated on December 22, 2022. I used one of Lauga's scripts, available on his GitHub, to update the data and made necessary tweaks to ensure it worked correctly. The dataset did not have any missing data or apparent outliers, as all the data originated from actual NBA games. The high-scoring nature of basketball makes the sport more resilient against outliers.

I did not remove any games from the dataset intentionally, as my goal was to predict outcomes for all types of games. The dataset from Lauga contained a games\_details table with offensive rebounds (OREB) at the player level. To calculate the total OREB for a team in a specific game, I summed the OREB for all players on that team for that game. Team names were also available in a separate teams table, which contained information such as team\_ids and team names.

I wrote a Python script using the Pandas library to prepare the data. The script first filtered the dataset for the 2022 season, then calculated the total OREBs for each team in every game, along with the point spread. Finally, I added columns for the total points and total OREBs in each game.

1.2 Engineered Variables

In this project, I found it unnecessary to engineer any additional variables. The dataset provided by Nathan Lauga, along with the outside data from FiveThirtyEight, was sufficient for predicting the outcome variables. The existing data contained relevant information about team performance, player statistics, and game outcomes, which allowed for the development of a robust predictive model without the need for any additional engineered variables.

While the existing dataset provided by Nathan Lauga and FiveThirtyEight's models contained a wealth of information for my predictive model, there are several additional variables that could have been engineered to potentially enhance the model's performance. One such variable could be a team's recent momentum, calculated as the win percentage over a certain number of recent games. This could help capture the current form and confidence of a team as they enter a matchup. Another variable could be the average point differential for each team in their past games, which would provide an insight into the typical margin of victory or defeat for that team. Interaction variables between team and player statistics, such as combining team offensive efficiency with individual player shooting percentages, could help identify synergies or weaknesses within a team's lineup. Furthermore, creating weighted versions of key statistics that emphasize more recent games over older ones could better reflect the dynamic nature of team performance throughout a season. By engineering these additional variables and incorporating them into my predictive model, we might be able to uncover new relationships, leading to improved accuracy and a more comprehensive understanding of the factors that influence game outcomes.

1.3 Outside Data Integration

To improve my prediction model, I incorporated outside data from FiveThirtyEight. I specifically utilized their RAPTOR model, which is based on current player performance, and their traditional team-based Elo model. Although it was against the rules to directly factor this data into my model, I was allowed to use it as a means of "checking my work" on the predictions I created. This indirect utilization of the data allowed me to refine my predictions and ultimately enhance my model's accuracy.

While the integration of FiveThirtyEight's RAPTOR and Elo models proved beneficial for this project, there are other data sources that could have been explored to further enhance the predictive capabilities of the model. Historical player injury data could provide insights into how the absence of key players might impact team performance and game outcomes. Weather conditions and travel distances, although less significant for indoor sports like basketball, might still have some impact on player performance, fatigue, or game attendance. Additionally, advanced player metrics such as Player Efficiency Rating (PER), Win Shares (WS), and Box Plus/Minus (BPM) could offer a deeper understanding of individual player contributions to team success. Social media sentiment analysis could be employed to gauge fan and expert opinions, which might indirectly affect team morale and performance. Lastly, incorporating betting market data such as opening and closing lines or public betting percentages could reveal valuable information about market perception and expectations. By integrating these additional data sources and investigating their potential impact on the predictive model, we could potentially uncover new insights and further improve the accuracy of my game outcome predictions.

In conclusion, I have thoroughly cleaned and prepared the dataset, found it unnecessary to engineer additional variables, and integrated outside data sources to develop a robust prediction model for my STOR 538: Sports Analytics class project.

# Section 2: Methodology for Spread

2.1 Model Selection and Development

To develop the best predictive model for Spread, I began by utilizing a basic linear regression as a baseline. I considered various types of models, including neural networks, regression trees, and time series models. To ensure the robustness of the chosen model, I utilized cross-validation and experimented with adding interaction and polynomial terms.

I ultimately selected the XGBoost model for predicting Spread. XGBoost is an optimized distributed gradient boosting library that is designed to be highly efficient, flexible, and portable. It is especially effective for large datasets and can handle missing data, making it an excellent choice for this project.

In addition to the XGBoost model chosen for predicting Spread, there were several other data sources and models that could have been explored to potentially improve the model's performance. One potential data source is the inclusion of betting market data, such as opening and closing odds from various sportsbooks. This information could be used to gauge the general consensus of the betting public and incorporate it into my model, as the betting market is often considered to be a powerful predictor of game outcomes. Another potential data source is incorporating detailed player injury reports, which could help account for the impact of key player absences on a team's performance and the resulting Spread.

Alternative models that could have been considered for predicting Spread include Long Short-Term Memory (LSTM) neural networks, which are specifically designed to capture temporal patterns in time-series data. This could be particularly useful for identifying trends in team performance over time, especially when considering the dynamic nature of team rosters and performance throughout a season. Another alternative is the use of ensemble models, such as stacking or blending multiple base models, like linear regression, support vector machines, or random forests, to create a more diverse and robust prediction model. By leveraging the strengths of various models and data sources, it may be possible to further enhance the accuracy and reliability of my Spread predictions.

2.2 Feature Generation and Data Preprocessing

To generate features for the model, I calculated the mean of the last n games' statistics for both home and visitor teams, where n was set to 60. I then concatenated these values, creating a (2, 7) array for each game. The dataset was then standardized using the StandardScaler from the sklearn library.

2.3 Model Training and Evaluation

I split the dataset into training and testing sets using an 80-20 ratio. I used GridSearchCV to find the best hyperparameters for the XGBoost model, optimizing for the negative mean squared error (MSE), as GridSearchCV maximizes the score. The model was trained on the training set, and its performance was evaluated on the test set.

The best hyperparameters found for the model were:

{

'max\_depth': 3,

'learning\_rate': 0.01,

'n\_estimators': 300,

}

The mean squared error of the model on the test set was found to be 0.998, and the model's score (R^2) was 0.062.

2.4 Predicting Outcomes

Using the trained XGBoost model, I was able to predict the Spread, total points (PTS\_total), and total offensive rebounds (OREB\_total) for future games. For example, I predicted the outcomes for a game between the Charlotte Hornets (home team) and the Toronto Raptors (visitor team) as follows:

Spread: -1.2816967

PTS\_total: 227.91507

OREB\_total: 23.486887

In conclusion, I thoroughly investigated multiple angles and utilized various model types to develop the best predictive model for Spread in my STOR 538: Sports Analytics class project. The chosen XGBoost model, combined with the generated features and preprocessing steps, allowed me to effectively predict game outcomes, including Spread, PTS\_total, and OREB\_total.

# Section 3: Methodology for Total Points

3.1 Model Selection and Development

To develop the best predictive model for total points, I followed a similar approach as with Spread. Since total points is a numeric variable and could be highly skewed, I considered nonlinear transformations and Poisson regression, along with stepwise algorithms and regularization for variable selection. I also explored various types of models, including neural networks, regression trees, and time series models. To ensure the chosen model's robustness, I utilized cross-validation and experimented with adding interaction and polynomial terms.

Ultimately, I selected an XGBoost model for predicting total points, as it demonstrated excellent performance during the model selection process. The XGBoost model's flexibility, efficiency, and ability to handle large datasets and missing data made it an ideal choice for this project as well.

In addition to the XGBoost model, other data sources and models could have been explored to specifically predict Total points. For example, incorporating in-game situational factors such as pace, offensive and defensive ratings, and clutch performance could have provided a more comprehensive understanding of the factors influencing a game's total points. By examining the interaction between opposing teams' offensive and defensive strengths, I could have developed a more nuanced prediction for total points scored in a given game.

Alternative models that could have been considered for predicting Total points include autoregressive integrated moving average (ARIMA) models, which are commonly used to analyze and forecast time-series data. ARIMA models could have been employed to capture the temporal patterns and trends in total points throughout a season, helping to reveal any underlying patterns that might have been overlooked by other models. Another alternative is the use of deep learning techniques, such as convolutional neural networks (CNNs), which are known for their ability to identify patterns in large and complex datasets. By training a CNN to identify patterns in historical total points data, we could potentially have developed a more powerful predictive model. Finally, Bayesian hierarchical models could have been employed to account for the hierarchical structure of the data, such as team and player-level effects, while incorporating prior knowledge and beliefs about the relationships between the variables. By leveraging the strengths of these alternative models and incorporating additional data sources, it may be possible to further enhance the accuracy and reliability of Total points predictions.

3.2 Feature Generation and Data Preprocessing

The feature generation and data preprocessing steps for predicting total points were the same as those used for Spread. I calculated the mean of the last n games' statistics for both home and visitor teams, concatenated these values, and standardized the dataset using the StandardScaler from the sklearn library.

3.3 Model Training and Evaluation

I split the dataset into training and testing sets using an 80-20 ratio and employed GridSearchCV to find the best hyperparameters for the XGBoost model, optimizing for the negative mean squared error (MSE). The model was then trained on the training set, and its performance was evaluated on the test set.

The best hyperparameters found for the model were:

{

'max\_depth': 3,

'learning\_rate': 0.01,

'n\_estimators': 300,

}

The mean squared error of the model on the test set was found to be 0.998, and the model's score (R^2) was 0.062.

3.4 Predicting Outcomes

Using the trained XGBoost model, I was able to predict the total points (PTS\_total) for future games along with Spread and total offensive rebounds (OREB\_total) as discussed in the previous section. The same example game between the Charlotte Hornets (home team) and the Toronto Raptors (visitor team) was used to demonstrate the model's predictive capabilities.

In conclusion, I conducted a thorough investigation from multiple angles and employed various model types to develop the best predictive model for total points in my STOR 538: Sports Analytics class project. The chosen XGBoost model, combined with the generated features and preprocessing steps, allowed me to effectively predict game outcomes, including total points, spread, and OREB\_total.

# Section 4: Methodology for OREB

4.1 Data Aggregation and Cleaning

The methodology for predicting OREB is quite similar to that of total points, with the main difference being the source of the OREB variable. Since OREB is a count variable and is only available on the player level, I aggregated the player data by team and game and merged it into the games dataset.

To accomplish this, I performed the following steps:

1. Load the player-level dataset.
2. Group the player data by team and game, summing the OREB variable for each group.
3. Merge the aggregated OREB data into the games dataset using the team ID and game ID as keys.

4.2 Model Selection, Development, and Feature Generation

As with the methodologies for spread and total points, I considered various types of models for predicting OREB, including neural networks, regression trees, and time series models, and utilized cross-validation and interaction/polynomial terms to ensure the best model for prediction.

After evaluating multiple models, I found that the XGBoost model, which had already proven effective in predicting spread and total points, was also the best choice for predicting OREB.

The feature generation for predicting OREB followed the same approach as for spread and total points. I calculated the mean of the last n games' statistics for both home and visitor teams, concatenated these values, and standardized the dataset using the StandardScaler from the sklearn library.

While the XGBoost model was effective in predicting OREB, other data sources and models could have been explored to potentially enhance the model's performance. Incorporating player-level data, such as individual player height, weight, and position, could have provided additional context when predicting OREB, as certain player attributes may be more conducive to securing offensive rebounds. Additionally, integrating advanced basketball metrics like Box Plus/Minus (BPM) and Player Efficiency Rating (PER) could have offered a more in-depth understanding of individual players' impact on the court, potentially leading to better OREB predictions.

Alternative models that could have been considered for predicting OREB include Poisson regression, which is particularly suitable for modeling count data. Poisson regression could have been used to model the relationship between OREB and other variables while accounting for the count nature of the data. Another alternative is the use of recurrent neural networks (RNNs), such as LSTMs, which are designed to capture temporal patterns in time-series data. This could be valuable for detecting trends in OREB throughout a season and understanding how individual player and team performances evolve over time. Finally, ensemble models, such as stacking or blending multiple base models like linear regression, support vector machines, or random forests, could have been employed to create a more diverse and robust prediction model. By incorporating additional data sources and leveraging the strengths of various models, it may be possible to further improve the accuracy and reliability of OREB predictions.

4.3 Model Training and Evaluation

With the aggregated OREB data in the games dataset, I split the dataset into training and testing sets using an 80-20 ratio and employed GridSearchCV to find the best hyperparameters for the XGBoost model, optimizing for the negative mean squared error (MSE). The model was then trained on the training set, and its performance was evaluated on the test set.

The best hyperparameters found for the model were:

{

'max\_depth': 3,

'learning\_rate': 0.01,

'n\_estimators': 300,

}

The mean squared error of the model on the test set was found to be 0.998, and the model's score (R^2) was 0.062.

4.4 Predicting Outcomes

Using the trained XGBoost model, I was able to predict the total offensive rebounds (OREB\_total) for future games along with spread and total points as discussed in the previous sections. The same example game between the Charlotte Hornets (home team) and the Toronto Raptors (visitor team) was used to demonstrate the model's predictive capabilities.

Predicting total offensive rebounds (OREB) can be particularly challenging due to the inherently unpredictable nature of rebounding events. OREB relies on various factors that can be difficult to quantify and anticipate, such as player positioning, individual effort, and the dynamic interactions between offensive and defensive players. Moreover, the path of the ball after a missed shot is influenced by numerous uncontrollable factors, which can create a degree of randomness that is hard to capture in a predictive model. Additionally, players' individual rebounding abilities can fluctuate from game to game, further complicating the task of accurately predicting OREB outcomes. Despite these challenges, the development of sophisticated machine learning models, such as the XGBoost model used in this project, can help account for some of the inherent complexities and uncertainties involved in predicting OREB.

In conclusion, I followed a similar methodology for predicting OREB as I did for total points in my STOR 538: Sports Analytics class project. By aggregating player-level data, selecting the optimal XGBoost model, and using the generated features and preprocessing steps, I was able to effectively predict game outcomes, including total offensive rebounds, spread, and total points.

# Conclusion

In this STOR 538: Sports Analytics class project, a robust predictive model was developed for predicting NBA game outcomes, specifically spread, total points, and total offensive rebounds (OREB). The XGBoost model was identified as the best choice for all three outcome variables, due to its efficiency, flexibility, and ability to handle large datasets and missing data. The data was thoroughly cleaned, prepared, and preprocessed, and outside data sources were incorporated to enhance the model's accuracy. The model was evaluated using mean squared error and R^2 metrics, showcasing its effectiveness in predicting game outcomes. This project demonstrates the importance of utilizing advanced machine learning techniques, data preprocessing, and feature engineering to develop accurate and reliable predictive models in the field of sports analytics.