Predicting NBA Game Outcomes Using ML

CSC 422 Final Project Report

Samarth Jadia
CSC 422 - Undergraduate Student
NC State University
Morrisville, NC, USA
ssjadia@ncsu.edu

Ⅰ. Introduction and Background

Introduction

Predicting the outcomes of NBA games has always been a fascinating challenge that has captured the attention of sports enthusiasts, analysts, and sports betting markets. Accurate game outcome prediction models have the potential to revolutionize various aspects of the NBA ecosystem, including team strategy, player evaluation, fan engagement, and sports betting. However, the complexity of factors influencing game results, such as team dynamics, player performances, and contextual elements like home court advantage and team schedules, makes this a daunting task.

This project aims to tackle this challenge by developing a sophisticated machine learning model that leverages comprehensive historical data on team and player statistics spanning 10 NBA seasons from 2013-2014 to 2022-2023. By diving deep into the finer details of player/team performances and introducing new predictive features, this project aims to uncover hidden patterns and trends that can inform more accurate game outcome predictions. Moreover, it seeks to identify the most influential features in determining an NBA game's outcome, providing valuable insights into the key factors that contribute to a team's success or failure on the court.

Background & Relevant work

Traditional approaches to NBA game outcome prediction often rely on simple metrics such as team win-loss records or general box-score statistics, which fail to capture the complex interplay between team and player dynamics, as well as situational factors that significantly influence game outcomes .

Thabtah, Zhang, and Abdelhamid [3] employed machine learning methods including k-nearest neighbor, SVM, decision trees, linear regression, and Naive Bayes with a comprehensive set of team and player statistics. Their experiment resulted in a 67% accuracy baseline. Houde [2] found that existing models typically achieve 66-72% accuracy, correlated with a 32% upset rate, and identified key features impacting game outcomes. These findings allow for a realistic baseline to be set for this project.

Horvat [4] proposed a data-driven model using an extended team efficiency index, which consists of a team efficiency component based on individual player efficiency. Our model has adopted a similar approach that aims to harness individual player performance through the use of temporal player Impact Scores.

Zhao [5] proposed a model that considers the connections between NBA teams and the influence of opponents on game outcomes. Their model represents each team as a node in a graph, with edges connecting teams to their opponents and to their own past and future games. Similarly, our model incorporates opposing team information known beforehand, such as player/team recent performance and venue specific advantages (home or away).

Wang [6] investigated the use of various machine learning techniques, including Logistic Regression, Support Vector Machines, Deep Neural Networks, and Random Forest models, for predicting NBA game outcomes. The study found that Random Forest outperformed other models in predicting results, which supports our choice of using Random Forest as the core of our prediction model. Additionally, Wang identified field goal percentage as the most important

predictive feature. In our study, we aim to validate Wang's finding on the importance of field goal percentage as the most significant predictive feature. If our model yields different results, we will explore and identify the features that emerge as most significant for predicting NBA game outcomes.

These studies highlight the potential of machine learning in predicting NBA game outcomes and provide a foundation for our project, allowing us to set realistic benchmarks and refine our approach by incorporating novel features and techniques that capture the complex dynamics of the game.

Ⅱ. Methods

Novel Aspects

A key novelty of this project lies in the introduction of player-impact scores for the top 5 performers of each team (usually the starting 5), as well as a bench impact score. Unlike traditional prediction models that rely solely on team-level statistics, this approach aims to capture the contributions of individual players and the collective strength of the bench. The NBA uses the following formula to calculate Player Efficiency Rating (PER):

```
PER = [FG_make × 85.910 + Steals × 53.897 + 3P × 51.757 +
FT_make × 46.845 + Blocks × 39.190 + ORb × 39.190 + Assists ×
34.677 + DRb × 14.707 - Foul × 17.174 - FT_miss × 20.091 -
FG_miss × 39.190 - Turnover × 53.897] × (1 / Minutes_played)
```

Fig. 1 (Standard PER Formula)

Our model slightly tweaks this formula to emphasize offensive related statistics over defensive related stats by adjusting the weights. It also removes the aspect of standardizing the PER relative to the minutes played. This is done to emphasize a player's impact on a game rather than their efficiency. The modified formula expected to to represent Player Impact Scores (PIS) is as follows:

```
PER = [FG_make × 80.910 + Steals × 30.897 + 3P × 90.757 +
FT_make × 50.845 + Blocks × 39.190 + ORb × 45.190 + Assists ×
45.677 + DRb × 14.707 - Foul × 6.174 - FT_miss × 12.091 -
FG_miss × 30.190 - Turnover × 53.897]
```

Fig. 2 (Boosted Stats = Green; Buffed Stats = Red)

Another novel aspect of this project is the use of advanced feature selection techniques, specifically the ANOVA F-value feature selection and correlation analysis, to identify the most relevant and non-redundant features for predicting game outcomes. This approach helps to streamline the model while minimizing multicollinearity and improving interpretability by focusing on the most informative features.

Furthermore, another novel aspect of this work is the transformation and integration of three distinct time-series datasets – player stats, game information, and team stats – into a unified dataset suitable for machine learning tasks. Additionally, the project employs an innovative feature engineering approach that goes beyond the supplied features in the original datasets. It introduces novel statistics derived from domain expertise, such as player impact scores, momentum-related data, margins of victory, records, opponent records, and more. This domain-driven feature engineering allows the model to capture various aspects of the game that may not be immediately apparent from the raw data.

General Approach

The proposed approach involves leveraging a Random Forest classifier to predict NBA game outcomes. The ensemble learning combines multiple decision trees to make predictions. It is well-suited for handling a large quantity of features, capturing non-linear relationships, and providing feature importance rankings. A more detailed section of how the approach will be applied can be found in the *Experiment* section of this report.

Data Preprocessing: Merging the Game Info and Team Stats tables to form the primary dataset, and integrating advanced features derived from the Player Stats table.

Feature Engineering: Creating novel features such as player impact scores, bench impact score, and temporal features which capture individual player contributions and team performances in the most recent games.

Feature Selection: Employing ANOVA F-value feature selection and correlation analysis to identify the most relevant and non-redundant features for predicting game outcomes.

Model Training and Evaluation: Training the Random Forest classifier using a time-series cross-validation approach and evaluating its performance using appropriate metrics such as accuracy, precision, recall, and F1 score.

Hyperparameter Tuning: Tuning will be conducted on the Random Forest Model using Randomised Search Cross Validation. The parameters that will be considered are: number of estimators, max depth, minimum samples to split on, and minimum samples to split on for leaf nodes.

Rationale

Ⅰ. *Model*

The choice of a Random Forest classifier is based on its ability to handle a large number of input features, capture nonlinear relationships, and provide robust predictions. Random Forest utilizes an ensemble learning approach which allows the model to capture complex patterns and set aside different trees for subsets of features. This is particularly important in the context of NBA game prediction, where the relationship between various player and team statistics may not be linear.

Additionally, Random Forest is less prone to overfitting compared to individual decision trees,
making it a reliable choice for this prediction task. Other algorithms, such as ridge/logistic/linear regression were considered but not chosen due to its limitations in handling non-linear relationships and the potential for overfitting.

Ⅱ. *Key Attributes*

The introduction of player statistics within the team box scores is motivated by the need to capture the fine-grained details of player contributions and team depth. For example, a team with a very powerful starting 5 complemented by a weak bench may cause the model to decide otherwise. Traditional approaches that rely solely on team-level statistics may overlook the impact of individual player performances and the quality of the bench, which can significantly influence game outcomes.

The inclusion of temporal features derived from the team_stats table is based on the hypothesis that a team's recent form and historical performance *against their upcoming opponent* can provide valuable context for predicting their next game's outcome. These features, such as a team's aggregate statistics over their last 5 games, their season record against the next team they play, and their opponent's team statistics, seek to capture the team's short-term performance trends and head-to-head matchup dynamics. The choice of a 5-game window for aggregate statistics is influenced by the need for capturing recent performance while avoiding excessive noise from single-game variations.

Ⅲ. *Feature Selection*

The use of ANOVA F-value feature selection and correlation analysis is motivated by the need to identify the most informative features while reducing redundancy. Using this feature selection technique helps identify the features that have a significant relationship with the target variable (Fig. 3). In parallel, correlation analysis helps to identify and remove highly correlated features, which may lead to multicollinearity issues (Fig. 4). In recent years, the NBA has witnessed a significant shift towards a more perimeter-oriented and efficiency-focused style of play. As a result, combining my domain knowledge about the sport with the ANOVA and correlation based selection techniques allows the model to predict games based on the most relevant features.

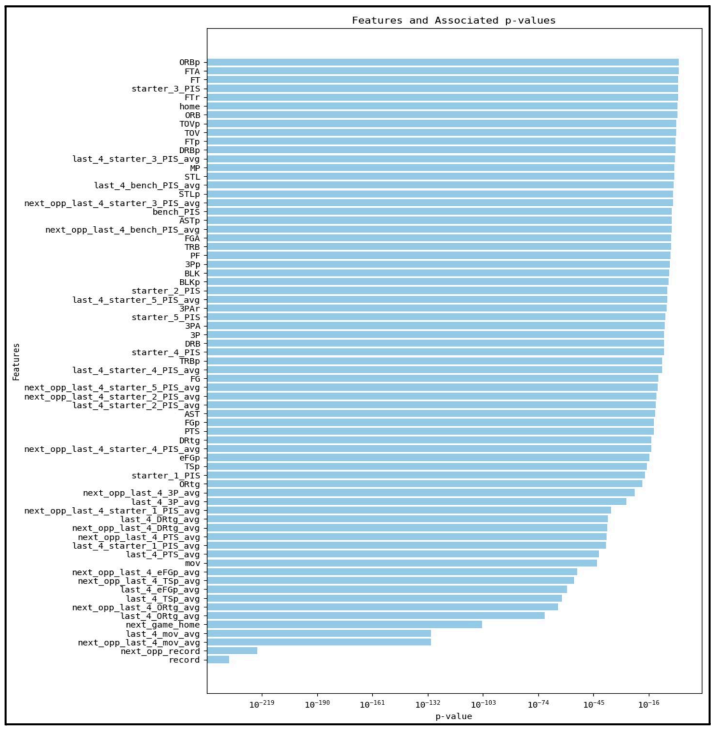

*Fig. 3: P-values for each team attribute (the lower the p-value, the more significant it is in predicting the target) (*Zoom in on PDF to view clearer)*

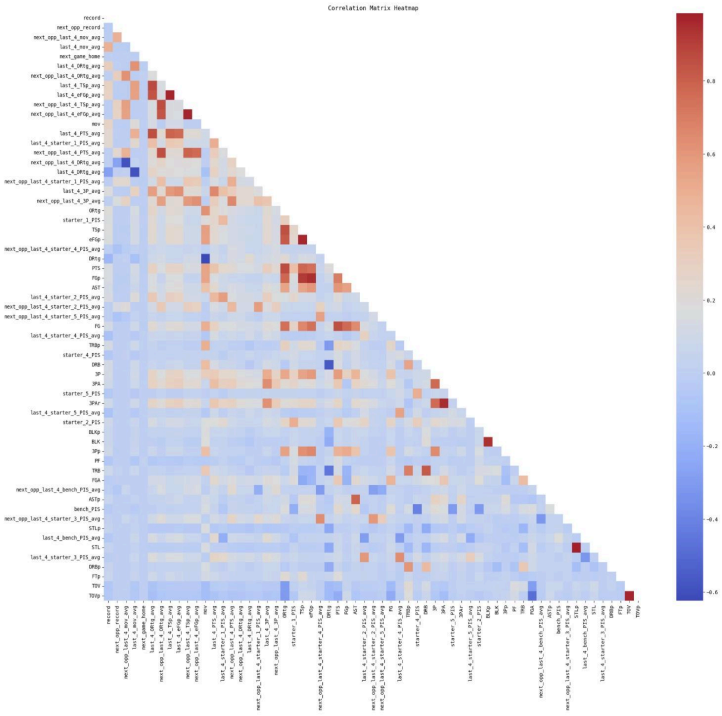

*Fig. 4: Correlation Heap Map of team based statistics (Darker Color = higher correlation) (*Zoom in on PDF to view clearer)*

IV. *Validation*

The time-series cross-validation approach is employed to ensure the model's ability to generalize to future unseen data. In other words, we don't want to use future NBA season data to predict past NBA season data. Traditional cross-validation techniques, such as k-fold cross-validation, assume that the data is independent and identically distributed, which is not the case for this time-series data. By using time-series cross-validation, where the data is split into training and testing sets based on chronological order, the model's performance can be assessed in a more applicable setting.

Ⅲ. Plan & Experiment

Dataset Description

The dataset used in this project is the "NBA Boxscore Dataset"(2023, Kaggle) [1] , which spans 10 NBA seasons from 2013-2014 through 2022-2023. The dataset consists of three distinct tables: Game Info, Team Stats, and Player Stats. The Game Info table and Team Stats table will be linked together by unique game IDs to form the primary dataset, while the Player Stats table will be used for advanced feature creation which will be integrated into the previously merged team dataset.

The Game Info table contains 12,000 rows and 8 columns, providing essential details about each game, such as the teams involved, date, season, scores, and the game outcome. The Team Stats table has 24,000 rows and 37 columns, aggregating player statistics to reflect team efficiencies per game. The Player Stats table is the most comprehensive, with 306,000 rows and 39 columns, offering deeper insights into individual player performances through box score statistics and efficiency metrics (Field Goal % vs. True Shooting %).

Hypotheses

The inclusion of 3-point related statistics as features in the Random Forest model is expected to significantly improve its predictive accuracy for NBA game outcomes. Given the increasing emphasis on the 3-point shot in the modern NBA, the impact of these features is hypothesized to become more pronounced with each progressive season. If this hypothesis does not hold true, a comprehensive investigation will be conducted to identify the most influential features driving the model's performance.

Experimental Design

To test the hypothesis, the following experiment will be conducted: (Supporting software and descriptions can be found in the python notebook accompanied by this paper)

Data Preprocessing: Perform these steps: (py notebook sections: "Importing Required Libraries" to "Adding the Target Variable")

- Merge the game_info and team_stats tables into a new dataframe, serving as the primary dataset.
- Reshape data so the result column represents whether the team has won or lost.
- Group data by team and shift the result column to create the target variable. The last game of each season won't have a next game, so replace appropriate null's in the target column with '2'.

Feature Engineering: Generate player impact scores (PIS), add temporal data about the current/opponent team regarding their last 4 game statistical averages, insert home or away attribute, calculate margin of victory, and calculate team record as a new attribute. These attributes will be inserted as a new attribute in

the primary data frame. Steps to perform: (py notebook sections: “Feature Engineering: Player Impact Score (PIS)” to “Feature Engineering: Add Next Game Stats”)

- *Player Impact Scores:* Execute the PIS formula (Fig. 2 on Page 2) on each row of the player stats dataframe. Identify starters and bench players for each team based on minutes played. Import starters' individual PIS and the bench's average PIS as attributes into the primary dataframe.
- *Margin of Victory:* Calculate the margin of victory (MOV) for each team based on home/away status and game scores.
- *Current Team Temporal Data:* Compute the team's last 4 games averages for relevant statistics, including offensive and defensive ratings, shooting percentages, points scored, and player impact scores. Use a rolling window to calculate averages, handling cases with fewer than 4 games.
- *Team Record:* Calculate the team's record by tallying wins and losses up to each game date.
- *Opposing Team Temporal Data:* Add opponent team statistics (ex. recent performance, home/away status, team record) to the primary dataframe. Sort the data frame by season, team, and date, then locate the opponent's row using the next game's ID. (To avoid data leakage, adjust the opponent's team record by adding or subtracting the result of the next game)

Feature Selection: Remove highly correlated and insignificant features using ANOVA and correlation analysis. Steps to perform: (py notebook section: “Feature Selection”)

- Remove irrelevant features (ex. game_id, season, date) and initialize an ANOVA object to compute p-values for each feature.
- Perform correlation analysis on the feature set.
- For feature pairs with high correlation (>0.7), remove the feature with the higher p-value (less significant) based on ANOVA results.

Model Training and Testing: Implement a custom function ('test_train') to handle time-series data splitting and model training/testing. The function trains the model on at least two previous seasons before predicting the next season, ensuring that future data is not used to predict past outcomes. With each iteration, the function will accumulate more seasons of data for training and predict the subsequent season. Steps to perform: (py notebook section: “Model Training and Testing”)

- Create a function that splits the data based on the time-series approach (described above). (Ensure function creates a dataframe with actual and predicted values for evaluation metrics.)
- Initialize a RandomForestClassifier and invoke the train_test function with the data frame, classifier, and the feature set

Hyperparameter Tuning: Focus on tuning the following hyperparameters: (py notebook section: “Model Training and Testing”)

- *Estimators:* Increasing the number of estimators can improve performance but also increases computational cost. Optimal range: [200, 225, 250, 275, 300].
- *Max Depth:* Higher values capture more complex relationships but may lead to overfitting. Optimal range: [8, 9, 10, 12, 14, 17].
- *Min Samples Split on Internal Node / Leaf:* Controls tree complexity and helps prevent overfitting. Internal Node Optimal range: [2, 5, 10]. Leaf Node Optimal range: [1, 2, 4].

(Optimal values for the Random Forest classifier were found to be: n_estimators = 275, min_samples_split = 2, min_samples_leaf = 4, and max_depth = 14)

Evaluating the Model: Create a confusion matrix and calculate precision, recall, and F1 assessing the model's performance. The model's accuracy should surpass the 57% baseline, which represents the home team's win percentage in the dataset. The model achieved an accuracy of 64.1%. Ensure that precision, recall, and F1 score are relatively similar, as false positives and false negatives are equally important in this application. (py notebook section: “Evaluating the Model”)

Hypothesis Evaluation: To evaluate the hypothesis, compare the performance of two models across multiple NBA seasons: one with 3-point statistics and another without. Steps to perform: (py notebook section: “Hypotheses Evaluation…”)

- Calculate per-season accuracy for the original feature set (including 3-point statistics).
- Calculate per-season accuracy for the feature set without 3-point statistics.

IV. **Results**

Results

After rigorous feature engineering, feature selection, and hyperparameter tuning, our model achieved a final accuracy of 64.1%, improving from 63.1% (before hyperparameter tuning).

The following confusion matrix and evaluation metrics below represent the final model's performance on the testing data consisting of seasons 2015-16 to 2022-23.

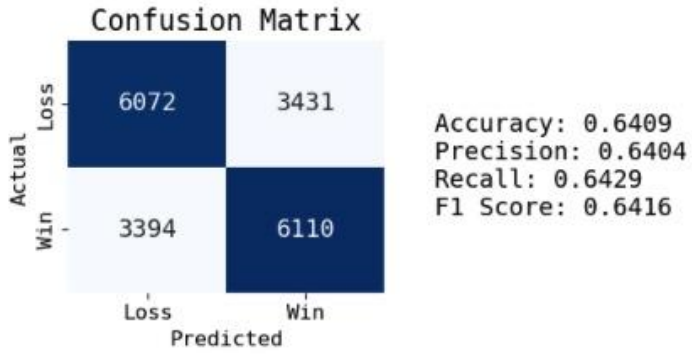

Fig. 5: Confusion Matrix of Final Model

The following table below highlights the final models accuracy per season of testing data with two differing feature sets to evaluate the hypothesis: 1) Final feature set containing 3-point related statistics 2) Final feature set excluding 3 point related statistics.

Season	Accuracy Score (with 3P)	Accuracy Score (without 3P)	Accuracy Difference
2015-2016	0.6561	0.6520	-0.0041
2016-2017	0.6313	0.6215	-0.0098
2017-2018	0.6630	0.6581	-0.0049
2018-2019	0.6484	0.6435	-0.0049
2019-2020	0.6398	0.6449	0.0052
2020-2021	0.6245	0.6181	-0.0065
2021-2022	0.6463	0.6431	-0.0033
2022-2023	0.6080	0.6096	0.0016

Fig. 5: Accuracies with and without 3 point related statistics (without 3P - with 3P)

The inclusion of 3-point related statistics slightly improved the model's accuracy, but not to the extent stated in the hypothesis (not significant at all). The accuracy difference between models with and without 3-point statistics did not increase over each season.

The top ten features of most significance are:

1. Team Record
2. Next Opponent’s Team Record
3. Next Opponent’s Last 4 Games Margin of Victory
4. Team Last 4 Games Margin of Victory
5. Playing Home or Away Next Game
6. Team Last 4 Games Offensive Rating
7. Next Opponent’s Last 4 Games Offensive Rating
8. Team Last 4 Game True Shooting Percentage
9. Team Last 4 Game Effective FG Percentage
10. Next Opponent’s Last 4 Games True Shooting Percentage

To compare the significance of these features in relation to each other (ex. How much more significant is Team Record than FG percentage?), refer to Figure 3 on Page 3. The smaller the blue bar, the smaller the p-value, and the larger the significance.

Discussion

The results did not fully support the initial hypothesis that the inclusion of 3-point related statistics would significantly improve the model's predictive accuracy, particularly in more recent seasons. The comparison of accuracies between models with and without 3-point statistics revealed only a slight improvement, which was not as substantial as hypothesized. Moreover, the accuracy difference between the two models did not exhibit an increasing trend over successive seasons.

The feature importance analysis (Fig. 3 on Page 3) provided valuable insights into the key factors driving the model's performance. Not so surprisingly, the current and opposing team's records emerged as the most significant attributes. This just indicates the better team usually wins. Additionally, a significant portion of the next most important features was related to temporal data about the previous games (margin of victory and home/away status being most influential). This captures the importance of a team’s momentum and how crucial it is when predicting their next game.

Additionally, the results align with some of the findings from prior work, such as the importance of incorporating player-level statistics. In particular, our findings support Wang's [6] observation that field goal percentage is a crucial predictor of NBA game outcomes. In our model, true shooting percentage, which is a more comprehensive measure of shooting

efficiency similar to field goal percentage, emerged as the most important traditional box score statistic (Fig. 3 on Page 3).

In regards to the confusion matrix and the evaluation metrics (Fig. 5), the similarity of these metrics suggests that the model's performance is consistent across different aspects of prediction. It indicates that the model is not biased towards either false positives (predicting a win when it's actually a loss) or false negatives (predicting a loss when it's actually a win). This balance is important because it shows that the model is equally effective in identifying both wins and losses.

To enhance the model's performance, future work could focus on incorporating more sophisticated player-level features, exploring advanced modeling techniques, and leveraging more comprehensive datasets.

Broader Impacts

The development of NBA game outcome prediction models has both positive and negative potential impacts on various stakeholders in the sports industry.

Positive Impacts (In progress)

- Enhanced fan engagement: Accurate game predictions can increase fan excitement and engagement, as they can make more informed decisions when participating in fantasy sports or sports betting.
- Player development: The features and insights derived from prediction models can be used to identify areas of improvement for individual players. By focusing on the factors that most significantly impact game outcomes, players can work on refining their skills and adapting their playing styles to better contribute to their team's success.

Negative Impacts (In progress)

- Unfair advantages in sports betting: The widespread use of NBA game outcome prediction models in sports betting can create an uneven playing field. If access to these models is limited to a select group of individuals or organizations, it could lead to unfair advantages and potentially manipulate betting markets.
- Over Reliance on models and loss of human intuition: As prediction models become more sophisticated and accurate, there is a risk that teams, coaches, and fans may become overly reliant on their outputs, neglecting the importance of human intuition, experience, and contextual understanding. While data-driven insights are valuable, they should not be the sole basis for decision-making.

V. Conclusions

Lessons Learned

Momentum in NBA Games: One of the most significant lessons I learned from this project was the importance of momentum in predicting NBA game outcomes. Prior to conducting this research, I believed that momentum was merely a narrative device used to add excitement to the game and had little actual impact on the results. However, the experimental results proved otherwise. This finding highlights the need to consider the short-term form and psychological factors that can influence a team's performance, in addition to their overall season statistics.

Shift Towards Offense-Oriented Basketball: Through the analysis of feature importance rankings (Figure 3 on Page 3), I noticed that the NBA has been transitioning towards a more offense-oriented style of play in recent years. The majority of the top-ranked features in terms of significance were related to offensive statistics, such as true shooting percentage, effective field goal percentage, and offensive rating. In contrast, defensive-related statistics generally had lower importance rankings. This observation aligns with the current trend in the NBA, where teams are prioritizing fast-paced, high-scoring offenses over traditional defensive strategies.

Real-World Application through Betting Simulation: One idea that I had for evaluating the model's performance in a real-world context was to simulate a betting strategy based on the model's predictions. By assigning different levels of bet amounts based on the model's confidence scores for each game, I could assess how profitable the model would be in a sports betting scenario. This approach would provide a more tangible measure of the model's effectiveness and help identify the optimal confidence thresholds for placing bets.

VI. Meeting Attendance

Project Completed Individually.

VII. References

[1] Diperna, L. (2023). *NBA Boxscore Dataset: Regular season player and team boxscore stats from the past 10 years* [Dataset; Kaggle].Kaggle. https://www.kaggle.com/datasets/lukedip/nba-boxscore-dataset/data

[2] Houde, M. (2021). Predicting the Outcome of NBA Games. *Bryant Digital Repository*. https://digitalcommons.bryant.edu/cgi/viewcontent.cgi?article=1000&context=honors_data_science

[3] Thabtah, F., Zhang, L., & Abdelhamid, N. (2019). NBA game result prediction using feature analysis and machine learning. *Annals of Data Science.* *https://link.springer.com/article/10.1007/s40745-018-00189-x*

[4] Tomislav Horvat, Josip Job, Robert Logozar, and Časlav Livada. 2023. A Data-Driven Machine Learning Algorithm for Predicting the Outcomes of NBA Games. Symmetry 15, 4, 798 (April 2023), 18 pages. https://doi.org/10.3390/sym15040798

[5] Kai Zhao, Chunjie Du, and Guangxin Tan. 2023. Enhancing Basketball Game Outcome Prediction through Fused Graph Convolutional Networks and Random Forest Algorithm. Entropy 25, 5, 765 (May 2023), 16 pages. https://doi.org/10.3390/e25050765

[6] Junwen Wang. 2023. Predictive Analysis of NBA Game Outcomes through Machine Learning. In The 6th International Conference on Machine Learning and Machine Intelligence (MLMI 2023), October 27–29, 2023, Chongqing, China. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3635638.3635646

VIII. Source Code

Github Link:
https://github.ncsu.edu/ssjadia/Predicting-NBA-Game-Outcomes-using-ML