Making a Statistical Model for Predicting NBA Game Outcomes

Alex Clare
Maya Hirsch
Henry Johnson
Lior Levy Meruk
Mary McSweeney
Rishabh Sood

December 7, 2024

Abstract

This model attempts to predict the outcome (win or loss) for specific NBA games after April 1st, 2024. Our dataset started with 23 variables and 26 variables were added to the dataset that could predict game outcomes. We ran lasso regression to find significant predictors. Once completed, we performed five-fold cross-validation to train and test our model (games before April 1st). Our training data consisted of all games played before April 1st and our testing data consisted of games played after April 1st.

Table of Contents

1 Introduction

Predicting NBA game outcomes is a complex task that involves analyzing statistical data, historical performance, and various team dynamics. The dataset provided for this project includes a wide array of game statistics such as points, rebounds, assists, field goals made, and 3-point shooting metrics, alongside meta-information like home/away game indicators and cumulative team performance metrics.

This project aims to leverage significant predictors from the dataset to predict the outcomes of NBA games. Engineered features such as win percentages, weighted averages, and opponent-specific statistics were included to enhance the predictive model. Key contextual factors, such as home-court advantage and team stability, were also considered essential elements in the analysis.

The predictive model was constructed using a Random Forest classifier, which is particularly well-suited for capturing non-linear relationships and interactions among predictors. Feature selection was performed using Ridge and Lasso regression to identify significant predictors, ensuring the model focused on the most impactful variables. To evaluate model performance, K-fold cross-validation with $K = 10$ was employed to improve accuracy and minimize model variability.

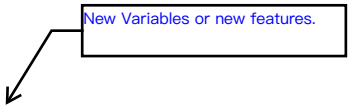
Trained on data before April 2024, the model successfully predicted the outcomes of games scheduled for the remainder of the season, offering valuable insights into the critical factors influencing NBA game outcomes.

2 Data Processing

For this project, we transformed the dataset to improve its usability for model building. New columns were constructed to ensure that each row contained only information from previous games. Numeric columns such as PTS (points), FGM (field goals made), and BLK (blocks) were recalculated into various statistical measures, including cumulative averages, weighted averages, total values, differences between opposing teams, and variances. These transformations helped capture more meaningful insights from the data, ensuring it was well-suited for the subsequent stages of analysis and model development.

2.1 Data Formatting and Initial Setup

In the initial stages of data preprocessing, several foundational transformations were applied to ensure the dataset was ready for analysis. The FT percent column, originally stored as a string, was converted into a numeric format to enable statistical computations. The Game Date column was standardized into a proper date format, allowing for time-based analyses and filtering. To better represent game-specific details, new columns were introduced. A Home Team column was created to indicate the team playing on their home court, while an Away Team column identified the visiting team. Additionally, an Opponent column was added to explicitly capture the opposing team for each game. These adjustments ensured the data was structured consistently and effectively for further feature engineering and modeling.



2.2 Creating a New Data Frame

To begin the preprocessing steps for model construction, a new data frame was created by excluding the first 100 rows of the original dataset. These rows were used to generate important features such as cumulative averages and differences, ensuring that each row represented the state of a team before the current game. This new data frame served as the foundation for the subsequent feature engineering process, providing a structure that could accommodate both team-specific and opponent-specific statistics.

Historical Team Performance Features needs to be proofed Several key features were created to capture the historical performance of each team. First, the total wins column was added to track the number of wins each team had accumulated before the current game. This was achieved by looping through the data and counting the number of victories for each team in the previous games. Similarly, the total games column was added to track the number of games played by each team up until that point. These columns were used to calculate additional features like win percentage, which represents the ratio of wins to total games played, offering an insight into a team's overall performance before each game.

Column Name	Description
`win_percentage`	Proportion of wins to total games played before the current game.
`home_adv`	Indicates if the team is playing at home (1) or away (0).
`<stat>_diff`	Difference between the team's and opponent's average statistics for `<stat>`.
`win_pct_vs_opponent`	Win percentage of the team against the specific opponent.
`win_pct_diff`	Difference between team and opponent.
`win_streak`	Longest winning streak of the team up to the current game.
`recent_win_pct`	Win percentage of the team in their last 3-5 games.
`weighted_avg_<stat>`	Weighted average of a specific statistic (`<stat>`), with weights inversely proportional to the days since each game.
`Win`	Indicates whether the team won (1) or lost (0) the current game.

Table 1: Summary of New Columns Added

2.3 Historical Team Performance Features

Several key features were created to capture the historical performance of each team. First, the total wins column was added to track the number of wins each team had accumulated before the current game. This was achieved by looping through the data and counting the number of victories for each team in the previous games. Similarly, the total games column was added to track the number of games played by each team up until that point. These columns were used to

calculate additional features like win percentage, which represents the ratio of wins to total games played, offering an insight into a team's overall performance before each game.

2.4 Opponent-Specific and Contextual Features

To capture the dynamics of each game in relation to the opponent, several features were engineered. The opponent average columns were created to represent the average statistics for the opponent up until the current game. This allowed the model to consider not only a team's performance but also how their opponent performed historically. Additionally, the win percent vs opponent column was calculated to reflect the win percentage of a team specifically against their upcoming opponent, which could offer a more nuanced prediction of the game outcome based on historical head-to-head performance.

This is good.

2.5 Advanced Statistical Features

In addition to basic performance metrics, more advanced features were created to capture variance and trends in performance. The total plus/minus column sums the plus/minus values from previous games, reflecting the overall performance impact of a team. Variance in-game statistics were also computed for each team, providing insight into how consistent a team's performance had been over the season. Additionally, recent win percent was calculated to measure a team's performance in their last few games, offering a glimpse into their current form. Finally, weighted averages of past performances were computed, with more recent games carrying greater importance, to give the model more recent context for predictions.

I would suggest removing those section titles, but just bold text to describe feature engineering step.

2.6 Outcome and Momentum Features

To help the model predict game outcomes, several features related to momentum and team performance were created. A win streak column was added to track the longest consecutive wins for each team, providing a measure of team momentum going into a game. The home adv column was also created to indicate whether a team was playing at home (1) or away (0), as home-court advantage can play a significant role in outcomes. These features, combined with all the previous performance data, helped provide a complete picture of each team's situation before the game.

3 Experiment Setup

After feature engineering, we looked into picking a subset of features to employ in building our predictive models. We began by assessing the correlations between our predictors, finding that most variables had low correlation values (0.1 and lower). Below is a correlation matrix demonstrating the predictions most strongly connected with. The predictors don't have a strong correlation with the outcome variable. There were no major indicators of multicollinearity between variables other than for +/- diff and win pct diff columns.

We utilized Lasso Regression to improve our model. Lasso regression shrinks coefficients and selects relevant features in linear models using L1 regularization to reduce overfitting. It improves

What is missing is the size of training and testing.

model performance and interpretability by eliminating irrelevant variables. According to Lasso, the features that should be kept include home adv, fga diff, 3pm diff, ft% diff, oreb diff, dreb diff, ast diff, tov diff, pf diff, +/- diff, win pct diff.

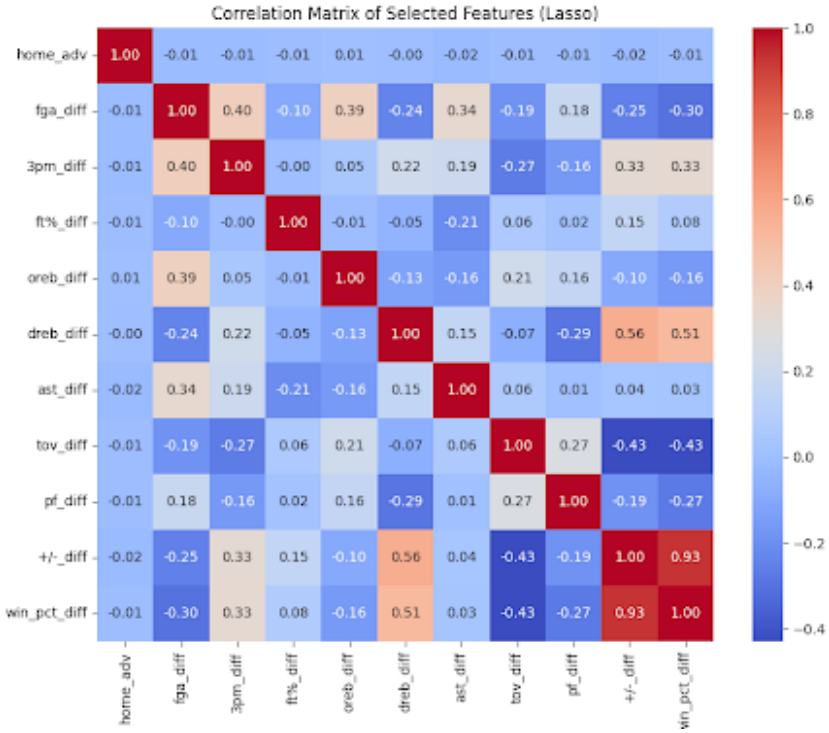


Figure 1: Due to the high correlation, we got rid of +/- diff and win pct diff columns. This corroborates our VIF analysis.

4 Results and Analysis

Logistic regression is a widely used statistical method for binary classification tasks. Logistic regression is a discriminative model that estimates the conditional probability of a sample belonging to class 1, P(Y = 1—X). In comparison to linear regression, which predicts continuous outcomes, logistic regression employs the logistic function to model the connection between input data and the likelihood of the target class. After running Logistic Regression, the cross-validation scores were [0.65254237 0.66949153 0.6440678 0.5720339 0.67372881]. The mean accuracy was 64.2%. Logistic Regression was the classification method with the highest mean accuracy.

Support Vector Machine (SVM) is a supervised machine learning algorithm primarily used for classification and regression tasks. It is particularly effective in high-dimensional spaces and for problems with a clear margin of separation. We used SVM to train and test our model with a Gaussian kernel. SVM achieved the second-highest accuracy rate out of all of our models, with the following rates: [0.59745763 0.66949153 0.60381356 0.58474576 0.65254237]. With a mean

Use table to summarize the testing accuracy and better average them.

accuracy of 62.7%, SVM was able to predict the game results pretty well, only second to Logistic Regression.

Random Forest is a supervised machine learning algorithm used for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to create a robust, accurate, and versatile model. With Random Forest, we achieved our third-highest accuracy score. Random Forest Cross-Validation Scores: [0.58898305 0.65042373 0.58262712 0.55508475 0.65254237]. With a mean accuracy of 60.5%, Random Forest is an average predictor of NBA games accuracy.

K-Nearest Neighbors (KNN) is simple and intuitive for classification and regression. It finds the K nearest data points (neighbors) to an input instance. The approach assigns the most common class label among these K neighbors, implementing a majority vote. KNN is a nonparametric model, which memorizes the training dataset and predicts during query time. The simplicity and effectiveness of KNN made it an appealing choice for us. After running KNN cross-validation, our 5 scores were [0.58898305 0.59745763 0.57627119 0.59745763 0.58898305]. With a mean accuracy of 58.9%, KNN was the least accurate method.

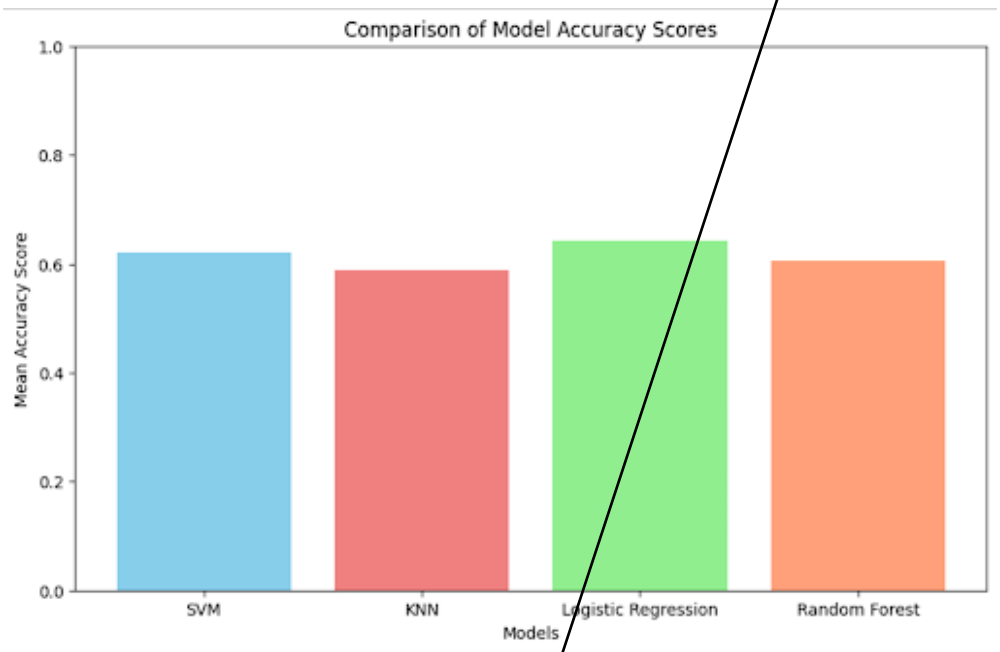


Figure 2: Mean accuracy scores of all classification methods

I think more experiments could be added, such as different values of K on the result and different depth in random forests, which will enhance the quality of this report.