University of California, Los Angeles

Fall 2024

Statistics 101C Final Project

Predicting Basketball Game Outcomes

Evan Garcia, Sreedeekshita Gorugantu Venkata, Gabriel Pham,
Mandy Vien, Nam Vien, Warren Wu

Professor Shirong Xu

Table of Contents

1. Introduction

This project focuses on analyzing NBA game data to predict basketball game outcomes based on historical statistics and engineered features. Utilizing a dataset containing 2,460 games with 24 variables, we aim to construct predictive models by exploring the relationships between game statistics and results, using methods discussed in foundational texts on statistical learning (James et al., 2013). Feature engineering is a key aspect, including considerations of home advantage, team stability, and weighted averages of recent performances to better reflect current team capabilities. Our methodology involves preprocessing data, creating informative features, and training various machine learning models, such as Random Forest and Gradient Boosting, to optimize predictive accuracy. By investigating these approaches, this study seeks to uncover insights into factors influencing game outcomes while improving prediction performance beyond established baselines.

2. Data Preprocessing

Before applying analysis methods to the data, we first pre-processed the raw data in the NBA dataset. The raw dataset contains 2,460 observations and 24 columns. These initial steps are crucial as they align with the best practices in predictive modeling, where data quality significantly influences the outcome (Hyndman & Athanasopoulos, 2021). This raw data, however, contains several shortcomings that needed to be fixed before proceeding with the analysis part of this project.

(1) Home Advantage Analysis

The **Home_Advantage** feature was created based on the **'Match Up'** column, while the **W/L** column was converted to numeric values (1 for wins and 0 for losses) to calculate win rates for home and guest teams.

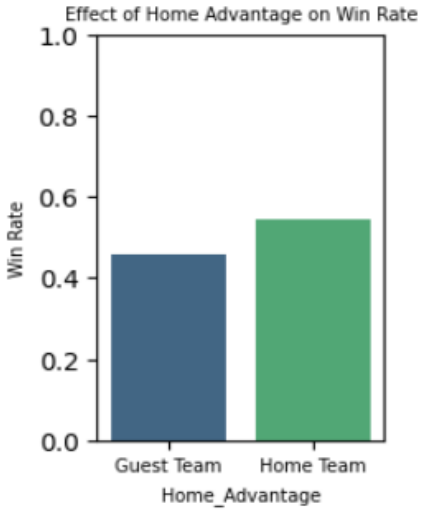

Figure 1. Win rates for home and guest teams, based on Home_Advantage and W/L features.

The purpose of this graph is to analyze the effect of home advantage on NBA game results and see if the home team tends to win more often than guest teams. The bar chart shows that the home team has a higher win rate than the guest team, indicating that the home advantage may impact the game

performance. Thus, home advantage can be considered as a feature in Win/Loss prediction, as well as its interaction with other factors such as weighted performance metrics.

(2) Feature correlation with W/L

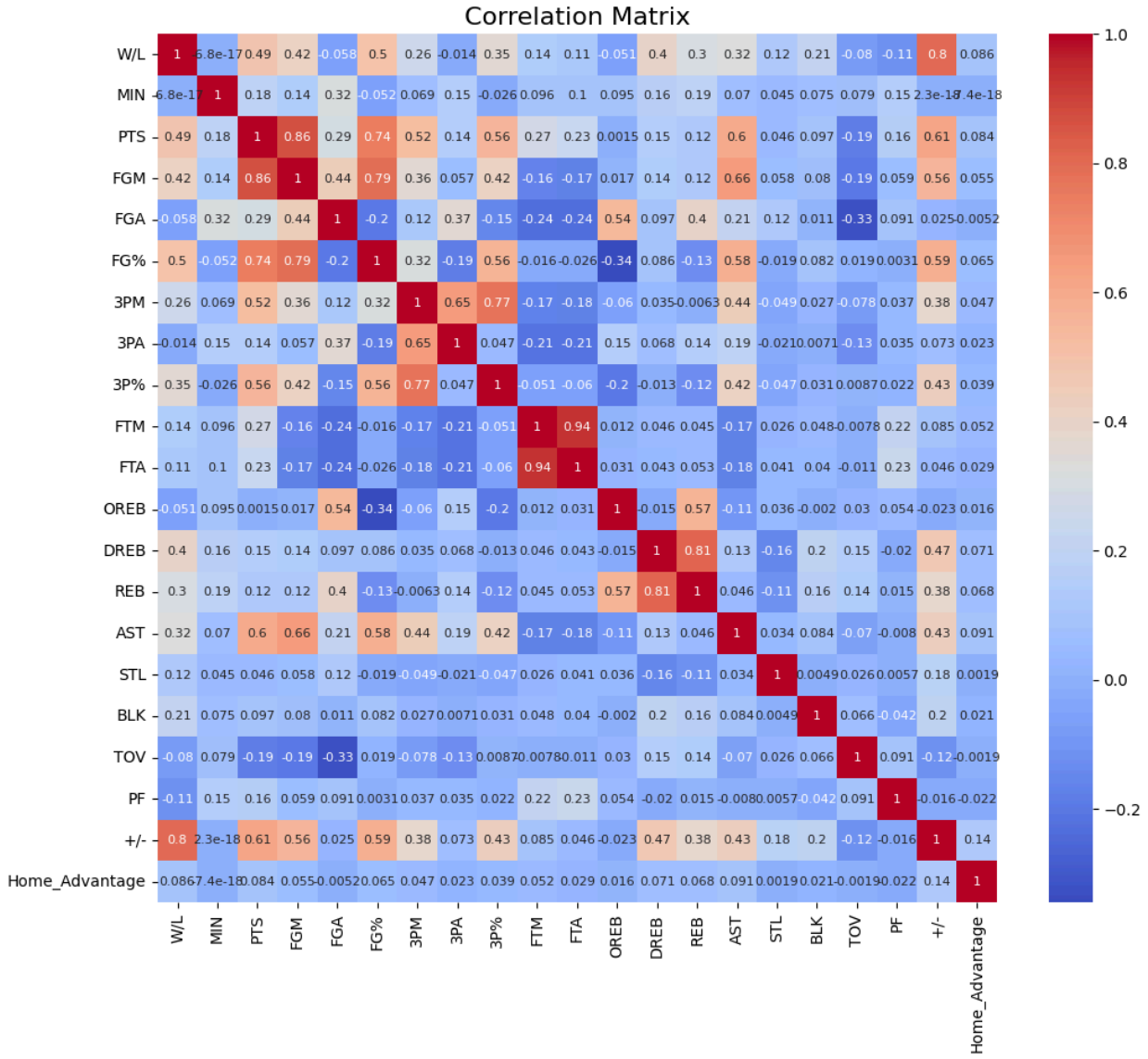

Figure 2. Correlation of features with W/L

The correlation analysis using the heatmap highlights key factors influencing game outcomes, with **Points Scored (PTS)**, **Field Goals Made (FGM)**, and **Field Goal Percentage (FG%)** showing strong positive correlations with **Win/Loss (W/L)**. These findings align with expectations, as efficient scoring directly contributes to winning games. Conversely, variables such as **Turnovers (TOV)** and **Personal Fouls (PF)** exhibit small negative correlations, indicating that mistakes and fouls slightly hinder performance.

(3) Top scoring teams based on win ratio and points scored

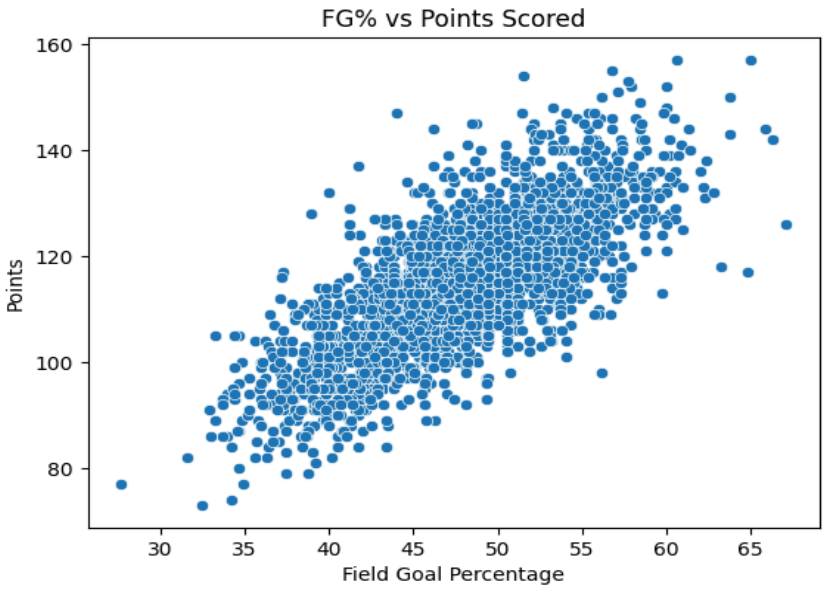

Figure 3. Field goals percentage vs. Points scored

We can observe that there is a clear positive correlation between FG% and points scored. As FG% increases, the number of points scored tends to increase. This makes sense because a higher FG% means more successful shots, contributing directly to more points.

We also notice that at higher FG% values (above 55%), the spread in points increases slightly. This could indicate that while FG% is a strong predictor of points scored, other factors like total field goals attempted and free throws also contribute to the variation in points.

2.1 New Features

We felt the existing variables in the raw data were not comprehensive enough to capture everything that happens within the course of a basketball season, as well as within a single basketball game. To better capture the variability here, we transformed the existing variables into new variables:

1. **Offensive Rebounds per Total Missed Shots (Rebound Efficiency)**: Measures a team's ability to recover the ball after missing a shot, reflecting their efficiency in offensive rebounding.
2. **Average steals across Games played previously**: Represents the average number of steals a team makes over the total games played before the game we are predicting, such as an entire season or the last X games.
3. **2 point field goal percentage (2P%)**: The efficiency of scoring 2-point field goals by excluding 3PA from the total field goal attempts and makes.
4. **FT%** contains non-numeric values such as (‘-’) were converted into numeric by treating them as 0, ensuring consistent and valid data for further analysis.
5. **Days since last game**: (Current date – date of most recent game), measures how much time the team has gone without playing a game

6. **Measurements of consistency**:
    a. Weighted variance in points scored in the last 10 games
    b. Weighted variance in field goals made in the last 10 games
    c. Where the weights vector was (1, … 10), with 10 corresponding to the most recent game, and 1 corresponding to the 10th most recent game
7. **Home field advantage**: a binary column assigned 1 if the team corresponding to the observation was the home team, 0 if they were the away team.

2.2 Features: Weighted and Unweighted

To predict game outcomes, we used only the matchup and date data, excluding other raw data. We applied an averaging function to each column (including our new features), considering all previous games for the team. Additionally, we used a weighted average for the last 10 games, with weights (1 to 10) giving more importance to recent games, as they are expected to have a greater influence on the outcome.

2.3 Ensuring a Single Prediction

In the raw dataset, each game has two observations: one for the home team and one for the away team. To avoid predicting both teams with a >50% chance of winning, we aggregated the rows for each game. We created new columns for the away team's stats, labeled "Guest_[stat]," and appended them, pivoting the dataset so each row corresponds to a unique game.

Similarly, before selecting features, we removed duplicate and redundant columns, which were:

- **'Match Up'** - the matchup info already exists in other columns
- All the columns from the raw dataset except for **'W/L_Home'** (since we are using that column as the response variable)
- **'MIN_Home_unweighted_avg'** and **'MIN_Guest_unweighted_avg'** - since each team plays the same amount of minutes every game, these columns are redundant
- **'days_since_last_Home_unweighted_avg'** and **'days_since_last_Guest_unweighted_avg'** - these columns don't tell us any information, and averaged out, each team has the same average amount of days in between games

3. Feature Selection

3.1 Dealing with Multicollinearity

The first round of feature selection involved pruning away the variables that were found to have a VIF value higher than 10, iterating through all of the given variables until they all met the desired criteria. The models were initially tested without a check for multicollinearity, which resulted in lower accuracies for models such as QDA/LDA. However after implementing the reduction their accuracies were much higher with only eight variables selected to be passed on to other feature selection techniques.

Selecting the required features are important for improving the predictive performance and interpretability of machine learning models. In this study, we employed **Lasso (Least Absolute**

Shrinkage and Selection Operator) and **Ridge regression** to select important features for predicting the outcome of basketball games played at home. We chose these methods due their ability to handle multicollinearity and perform feature regularization.

3.2 Employing LASSO and Ridge

(I) Lasso

We selected the features with non-zero coefficients after training the Lasso model as the most important predictors. The most significant features from Lasso included:

Table 1. Significant features from Lasso

Weighted Guest +/-	Weighted Home OREB	Weighted Home	Weighted Guest TOV
-0.118	-0.034	0.143	0.008

(II) Ridge

We analyzed coefficients from the Ridge model to gauge the importance of features that might not have been selected by Lasso due to its sparsity constraint. Ridge identified additional features with smaller yet notable contributions, such as:

Table 2. Significant features from Ridge

Weighted Guest TOV	Weighted Home DREB	FGA Guest (unweighted average)
0.026	0.025	-0.00018

The **Mean Square Error (MSE)** was used as the evaluation metric for these models, with the Ridge Test MSE reporting at 0.198 and Lasso Test MSE reporting at 0.192. Lasso had a slightly lower MSE, which confirms its effectiveness in both feature selection and prediction.

4. Model Selection

SVM: The Support Vector Machine (SVM) model with a Gaussian kernel was evaluated using 5-fold cross-validation. This technique splits the dataset into five subsets, with four used for training and the remaining set used for validation in each iteration. Performance metrics, including accuracy, precision, recall, and F1-score, were evaluated for both training and validation sets. The SVM model showed consistent performance across training and validation sets. The metrics were consistent, with minor differences between the training and validation sets.

Table 3. Support Vector Machine Mean Metrics over 5-fold Cross-Validation

Model Stage	Accuracy	Precision	Recall	F1-Score
Training	0.74	0.72	0.75	0.74
Validation	0.73	0.73	0.73	0.73

Random forest: To evaluate the feasibility of predicting game outcomes (*W/L_Home*) using historical game data, we implemented a Random Forest Classifier. This method is highly effective for such analyses, as detailed by Breiman (2001), who explains the classifier's ability to manage high-dimensional data. The dataset was first sorted chronologically based on game dates to ensure that only prior game data was used for training. To evaluate the performance of the Random Forest classifier, we employed 5-fold cross-validation across the dataset. The Random Forest classifier demonstrated a strong performance overall, with an average accuracy of 71%.

Table 4. Random Forest Mean Metrics over 5-fold Cross-Validation

Model Type	Accuracy	Precision	Recall	F1-Score
Validation	0.71	0.70	0.68	0.69

K-Nearest Neighbors: The K-Nearest neighbors model may be able to capture unique interactions among the variables reduced by multicollinearity & LASSO selection. Applying this technique with the nearest neighbors set to 35, upon a 5-fold cross-validated set consistent with the other models yielded an average accuracy similar to other models at approximately 70%.

Table 5. K-Nearest Neighbors Mean Metrics over 5-fold Cross-Validation

Model Type	Accuracy	Precision	Recall	F1-Score
Validation	0.70	0.71	0.69	0.70

Decision Tree: The Decision Tree Classifier is a supervised learning algorithm that splits data into subsets based on feature thresholds, creating a tree structure where each leaf node represents a prediction. This implementation evaluates the model using 5-fold cross-validation, ensuring robust performance assessment. Metrics like accuracy, precision, recall, and F1 score are used to evaluate the classifier. It provides a solid baseline with consistent and interpretable results. To improve performance, ensemble methods like Bagging or Gradient Boosting should be explored.

Table 6. Decision Tree Mean Metrics over 5-fold Cross-Validation

Model Type	Accuracy	Precision	Recall	F1-Score
Validation	0.71	0.70	0.69	0.69

Bagging: Bagging Classifier (Bootstrap Aggregation) works by training multiple base estimators on random subsets of the training data and aggregating their predictions to improve overall accuracy. This method reduces variance, enhancing the model’s stability and robustness, essentially working with high-dimensional data.

Table 7. Bagging Mean Metrics over 5-fold Cross-Validation

Model Type	Accuracy	Precision	Recall	F1-Score
Validation	0.71	0.70	0.68	0.69

Gradient Boosting Machine: This Classifier is one of the most competitive approaches for predictive modeling. It builds on the principles of sequential learning where each tree improves on its predecessor, a technique comprehensively described by Friedman (2001). It interprets the data through a sequential learning process where each tree corrects the errors of its predecessor, refining prediction over time. Using 5-fold cross-validation, the GBM model uncovers nuanced patterns, such as how team performance metrics interact to predict outcomes (Hometeam Win/Lose). Respectively, it demonstrates the ability to capture complex patterns and make reliable predictions.

Table 8. Gradient Boosting Mean metrics over 5-fold Cross-Validation

Model Type	Accuracy	Precision	Recall	F1-Score
Validation	0.71	0.70	0.70	0.70

LDA and QDA: The Linear Discriminant Analysis and Quadratic Discriminant Analysis models were evaluated using 5-fold cross-validation. This technique splits the dataset into five subsets, with four used for training and the remaining set used for validation in each iteration. Despite our predictor variables not meeting the assumption of multivariate normal distribution, the LDA and QDA models performed well.

The LDA model showed inconsistent performance across training and validation sets with the following mean metrics over 5-fold cross-validation, with each fold representing a subset of the data in ascending order (for example, the 0th fold represents the first 1/5 of the games). As the testing subsets went further into the season, the testing accuracy decreased. The 0th fold was substantially more accurate than the 4th fold. This result is strange because for the first several games of the season, a lot of the columns have values of 0 and have small sample sizes to compute the features on. This suggests that these variables are not actually informative, and so when their values are not 0 they are just adding noise to the prediction, decreasing the accuracy. It's also possible that teams are more consistent in the beginning of the season, so teams that have won more games will continue to win more games in the beginning of the season.

Table 9. LDA Mean metrics over 5-fold Cross-Validation

Model Type	Accuracy	Precision	Recall	F1-Score
Validation	0.74	0.73	0.70	0.72

Table 10. LDA Performance by Fold

Fold	0	1	2	3	4
Accuracy score	0.86	0.73	0.71	0.69	0.70

The QDA model showed inconsistent performance across training and validation sets with the following mean metrics, performing similar to LDA. Once again, however, the 0th fold testing performed the best, which suggests similar reasons to LDA for the discrepancy in accuracy between each fold.

Table 11. QDA Mean metrics over 5-fold Cross-Validation

Model Type	Accuracy	Precision	Recall	F1-Score
Validation	0.74	0.75	0.69	0.71

Table 12. QDA Performance by Fold

Fold	0	1	2	3	4
Accuracy score	0.84	0.74	0.73	0.70	0.68

Logistic Regression: This model is a supervised learning algorithm used for binary or multiclass classification tasks by modeling the probability of an outcome using a logistic function. In this implementation, the features are standardized using a StandardScaler to enhance model performance and ensure numerical stability. A 5-fold cross-validation approach is applied to evaluate the model, providing robust performance metrics like accuracy, precision, recall, and F1 score. This ensures the assessment is reliable and accounts for variability in the data. Logistic Regression offers a balance between simplicity, interpretability, and efficiency, making it a strong baseline model.

Table 13. Logistic Regression Mean Metrics over 5-fold Cross-Validation

Model Type	Accuracy	Precision	Recall	F1-Score
Validation	0.74	0.74	0.71	0.72

Principal Component Analysis (PCA): This is an unsupervised dimensionality reduction technique that transforms data into a smaller set of orthogonal components while retaining most of its variance. In this implementation, PCA is applied after standardizing the features to ensure equal scaling. By retaining 95% of the total variance, PCA reduces the dimensionality of the dataset, simplifying the model training process while minimizing information loss. A Random Forest Classifier is trained on the PCA-transformed data, and its performance is evaluated using 5-fold cross-validation. Metrics such as accuracy, precision, recall, and F1 score provide insight into the classifier's effectiveness.

Table 14. Principal Component Analysis Mean Metrics over 5-fold Cross-Validation

Model Type	Accuracy	Precision	Recall	F1-Score
Validation	0.50	0.47	0.42	0.44

5. Results and Analysis

The performance of machine learning models for predicting game outcomes varied significantly. **Support Vector Machine (SVM)** achieved accuracy at **73%** and precision at **73%**, though it showed signs of potential overfitting due to its reliance on non-linear boundaries. These results echo findings from comparative studies on the application of SVM in sports predictions, like the study by Sculley & Pasanek (2008), which also highlights the model's capabilities in accurately forecasting outcomes in sports settings. **Gradient Boosting Machine (GBM)** and **Logistic Regression** delivered competitive and balanced results, with all scores around **70-74%**, showcasing GBM's ability to capture nuanced relationships and Logistic Regression's interpretability as a strong baseline.

The **Random Forest Classifier** performed robustly with **71%** accuracy and F1 score of **69%**, effectively capturing complex patterns like home advantage. In contrast, models trained on **PCA-transformed data** underperformed, with accuracy dropping to **50%**, suggesting that critical information was lost during dimensionality reduction. Overall, SVM excelled, while Random Forest and GBM proved to be strong, balanced alternatives.

From the results, we learned that ensemble methods like Random Forest and Gradient Boosting emerged as the most effective approaches, demonstrating consistent and reliable performance. In contrast, PCA transformations showed limitations, highlighting the importance of retaining key features in predictive modeling. Further improvements may involve refining feature engineering, testing alternative weighting schemes, and exploring additional advanced models to enhance accuracy.

6. Conclusion

In this study, we analyzed NBA game data to predict outcomes using machine learning models and advanced feature engineering techniques. By incorporating factors such as home advantage, weighted game statistics, and team stability, we were able to construct predictive models with competitive accuracy. LDA and QDA emerged as the most effective models, achieving mean accuracies of 74%, highlighting the importance of generative classification methods in capturing complex relationships within the dataset. Conversely, dimensionality reduction through PCA showed limitations, emphasizing the need to retain key features for robust predictions. These results demonstrate the potential of machine learning in sports analytics while also pointing to areas for further improvement, such as refining weighting schemes and exploring alternative modeling techniques. Our work provides a framework for understanding the factors influencing game outcomes and offers valuable insights for future research in predictive modeling within the domain of sports analytics.

References

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: principles and practice*. OTexts. https://otexts.com/fpp3/

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer. https://doi.org/10.1080/24754269.2021.1980261

Sculley, D., & Pasanek, B. M. (2008). Predicting win-loss outcomes in MLB baseball, a comparative study using SVM and logistic regression. Presented at the National Conference on Artificial Intelligence.