Statistics 101C Final Project

Predicting NBA Game Outcomes: Feature Engineering and Model Optimization Using Historical Game Statistics

Ivy Le, Priscilla Dixon, Kuan-Yi Chen, Bryan Mui, Justin Tong, Will Bodeau

Table of Contents

1. Introduction

This dataset provides detailed metrics on NBA team performance, including statistics such as points scored, offensive and defensive rebounds, assists, and shooting efficiency. These various metrics allow us to gauge teams' performance. Our goal is to leverage these metrics and create additional features to develop a predictive model of game outcomes.

In order to properly predict the outcome of game results, we must effectively construct additional features that let us see trends in team performance and the consistency of teams in competition. We will be adding features such as home court advantage and win streaks, which may have positive indications for a team's outcome. For example, home court advantage may greatly increase the likelihood of a team winning while a low amount of defensive rebounds may hint at a team's lackluster defense which can lead to a loss. These are critical factors based on a team's historical data that provide us with more information in order to construct predictive models.

We employ an ensemble of models, including logistic regression, k-nearest neighbor (KNN), random forest, linear and quadratic discriminant analysis, and XGBoost. These models allow us to improve upon the baseline accuracies of 67-70% established in prior analyses due to their ability to capture different patterns and relationships within the data and diverse predictive capabilities. By incorporating various feature engineering techniques and evaluating the impact of different statistics on game results, we aim to uncover key predictors of game results and provide actionable insights for predictive analytics in the NBA.

2. Data Preprocessing

2.1 Explore key features, summarize statistics, visualize distribution

The dataset comprises 2,460 game summaries with various statistics including team performance metrics (e.g., points scored, rebounds, assists). Each record represents one team's performance in a single game, such that each game is represented in two rows. There are some missing values in the data, marked with a "-". Whenever a numeric column contained missing data, it was replaced with the average of that column. The data consists of 30 teams, each with 82 games, and 1230 individual games in the season.

Redundant features were removed, such as FG%, 3P%, and FT%, as they represent ratios of already existing features. Including these correlated features in the model would result in unnecessary complexity, overfitting, and reduced interpretability. Similarly, the rebounds were removed, as it was simply the sum of defensive rebounds and offensive rebounds.

Metric	Mean	Standard deviation	Min	Max
Points	114.2	12.8 %	73	157
Fields goals percentage	47.5	5.5	27.7	67.1
Rebound	45.8	8.2	29	65
Rest days	2.1	0.99	0	9

Table 1. Key Summary Statistics

Teams typically score between 100–130 points per game, but outliers suggest occasional high-scoring games above 150 points. Rebounds and rest days show moderate variability, possibly influencing performance trends.

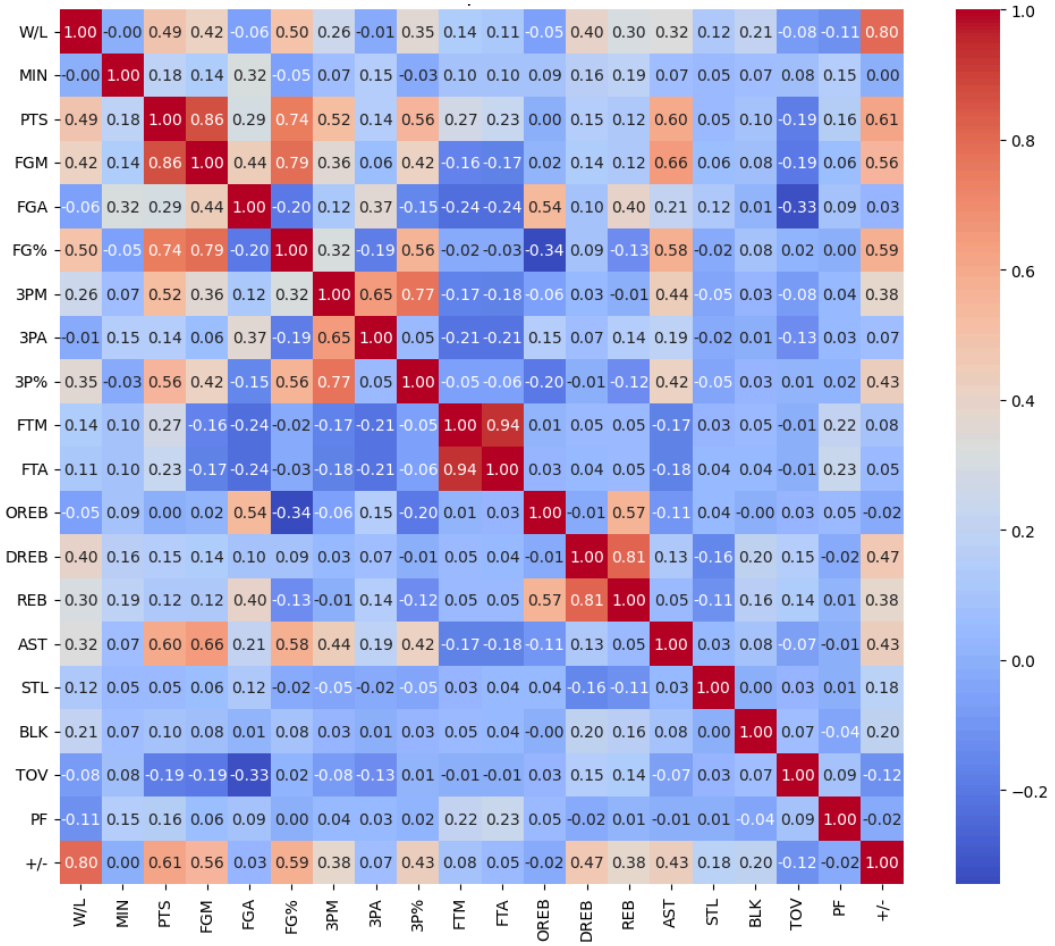


Figure 1. Correlation Heatmap of Numeric Variables

The correlation heatmap highlights relationships between all of the pre-existing features, with strong positive correlations (e.g., points and field goal percentage) and weaker relationships for other features. The point differential feature shows a strong 80% correlation to the win/loss of a team. This is explained by the fact that the +/- feature will be positive when a team wins and negative when a team loses, since it represents how many points a team was up or down by at the end of the game. In a preliminary logistic regression model, this resulted in a 100% accuracy rate. Information from the target variable was directly explained by this relationship. Intuitively, the direct relationship between a positive point differential and that team winning was giving our model inflated accuracy metrics. Other variables that were sources of data leakage were Field Goals Made and Points, which were highly correlated with each other, since the majority of points made in a basketball game are field goals. Similarly to the point differential feature, the model learned that the team with the most points won that game, directly revealing the outcome of the game.

2.2 Feature Engineering

In order to gain further insights into the trends that can help predict future games, we created 6 additional features:

1. **Home Advantage:** Home Advantage was a feature that was created, as a team playing on their home court has roughly a 55% chance of winning that game.
2. **Win Streak:** This is defined as the number of games a team has won in a row leading up to the current game. This considers homestreaks while home and while away.
3. **Days Since Last Game:** The next feature that was created was Days Since Last Game, which is designed to account for rest periods and travel periods that teams may experience. Teams may perform better if they have additional rest days.
4. **Instability:** Although we removed the number of points from the model, a statistic derived from points called instability is defined as the standard deviation of points for that team. This removed the issue resulting in a data leakage, but still allowed us to include information regarding points in the model.
5. **Weights on More Recent Games:** Weights were assigned to more recent games in order to account for changes in a teams strategy, roster, or overall condition. A team's most recent set of games is a more viable predictor of future games compared to a game that happened much earlier in the season.
6. **Season Segment:** Season Segment is the sixth added feature, and is designed to account for different strategies and plays, lineups, injuries, and pushing for playoffs. One-hot encoding was used to turn this into 3 binary columns, each column representing true or false based on whether it was early, mid, or late season.

3. Model Selection

Ensemble methods combine several base models to make a robust and generalizable prediction, as each base learner can capture different patterns in the data. Regularization was added for each method in the ensemble, as well as feature scaling for SVR and KNN. These adjustments aim to reduce redundancy and improve generalization. The ensemble aggregates predictions based on a majority vote that uses weighted probabilities, such that the prediction with the highest average probability is chosen. An ensemble method is more stable than one single individual model as it reduces the effects of an individual model's weakness.

A diverse number of learning approaches were included in this model. Logistic regression, LDA, and QDA serve to model linear relationships. Non-linear models, which include Random Forest, XGBoost, SVM, and KNN, capture complex and non-linear relationships. Models like Logistic regression have high-bias, so low-bias models like random forest and XGboost compensate for this. Conversely, low-variance models (LDA, QDA, logistic regression) stabilize predictions from models like KNN and SVM, which are prone to overfitting. A mix of models with different decision boundaries and distributional assumptions ultimately makes the ensemble more robust and accurate.

3.1 Random Forest. Random Forest is a machine learning model that allows us to gain more insight in classification and regression tasks. The way the model works is by forming multiple decision trees and combining them to increase model accuracy and reduce overfitting. Some key benefits of the random forest model include providing clear information on feature importance and introducing randomness.

3.2 Logistic Regression. Logistic Regression is a discriminative and parametric model used for binary classification. The goal of logistic regression to estimate the conditional probability

$P(Y = 1 \mid X = x)$. Logistic regression assumes a linear relationship between the features and the log-odds of the target variable, which is then transformed into a probability distribution using the logistic function. We chose Logistic Regression for its simplicity and interpretability, as it allows us to clearly understand how different game statistics (for example, points scored or rebounds) influence the likelihood of a win.

3.3 KNN. K-nearest neighbor (KNN) is a non-parametric, proximity-based algorithm that predicts the class of a sample based on the majority vote of its k-nearest neighbors. For our dataset, KNN is well-suited because it leverages historical game statistics and differences in performance metrics to identify patterns in team matchups. To optimize the model, we normalized the feature set using **StandardScaler**, ensuring that all variables contributed equally to the Euclidean distance calculations. The number of neighbors k was selected as 45, based on the square root rule ($k = \sqrt{1968}$), rounded to the nearest odd integer to avoid ties. The value of k was chosen to balance bias and variance, minimizing overfitting on training data while capturing

meaningful local patterns. This approach enabled the KNN model to effectively predict game outcomes, achieving robust performance when tested on unseen data.

3.4 LDA/QDA. Linear discriminant analysis (LDA) is a classification technique that finds a linear combination of features that best separate the different classes apart. It does so by projecting the data onto a lower dimensional space. Quadratic discriminant analysis (QDA) shares a similar goal but is able to handle non-linear decision boundaries. QDA is more flexible because of this and the fact that it does not assume a common covariance matrix.

3.5 XGBoost. XGBoost uses a gradient boosting method in which decision trees are iteratively grown based on the information of the preceding tree. The final outcome is a weighted sum of each weak classifier.

4. Results and Analysis

After training our models with the selected hyperparameters, we evaluated them on the testing data, which represents 30% of the dataset after preprocessing. The table below presents the training and testing accuracy for each model.

Model	Training Accuracy	Testing accuracy
Random Forest	1.00	0.7764
Logistic Regression	0.8318	0.7886
LDA	0.8262	0.7927
QDA	0.7241	0.6890
KNN	0.8105	0.7053
SVM	0.8877	0.8069
XGBoost	1.000	0.8028
Ensemble	0.9451	0.8089

Table 2. Training and testing accuracies across classification models

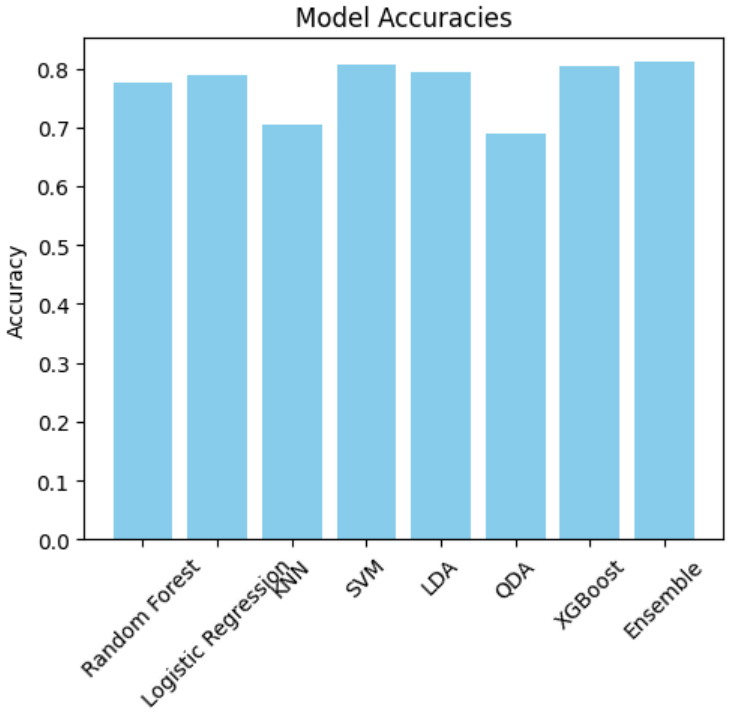
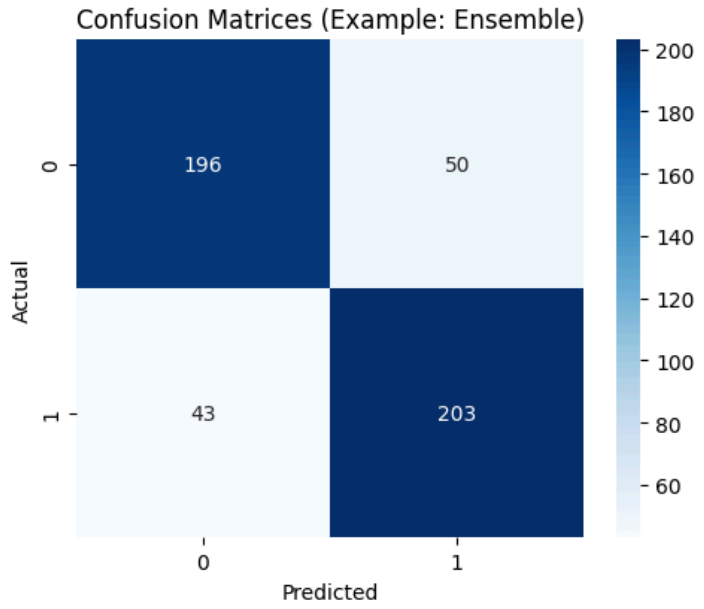


Figure 2 compares the accuracy of various models, with the Ensemble and XGBoost models achieving the highest performance, while KNN and Logistic Regression showed moderate accuracy, and SVM and QDA performed relatively lower.

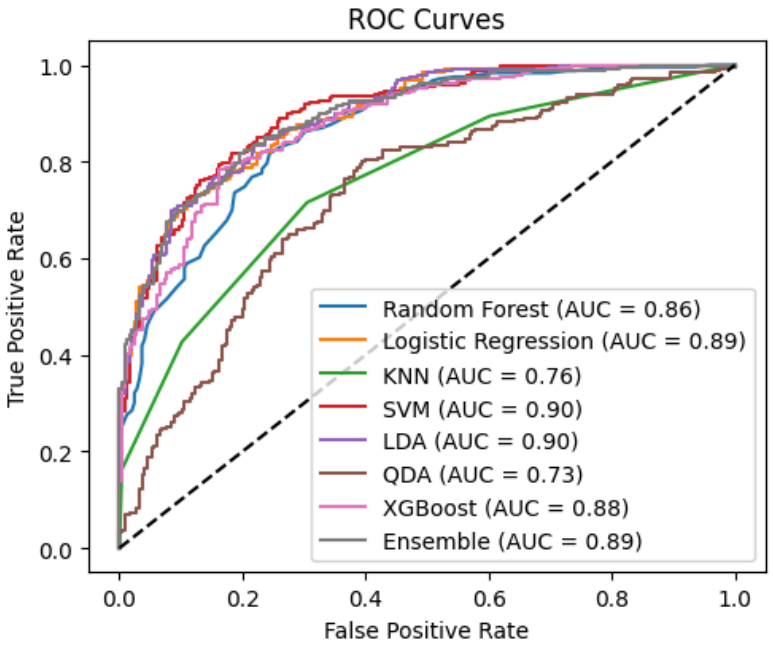
Figure 2. Bar plot of model testing accuracies



The confusion matrix shows True Negative (loss) and True Positive (win) in the upper left and bottom right corners respectively. The upper right and bottom left are False Positive (predicted win when the game was a loss) and False Negative (predicted loss when the game was a win) predictions. Overall, the model predicts roughly equal proportions of TN and TP, and FP and FN.

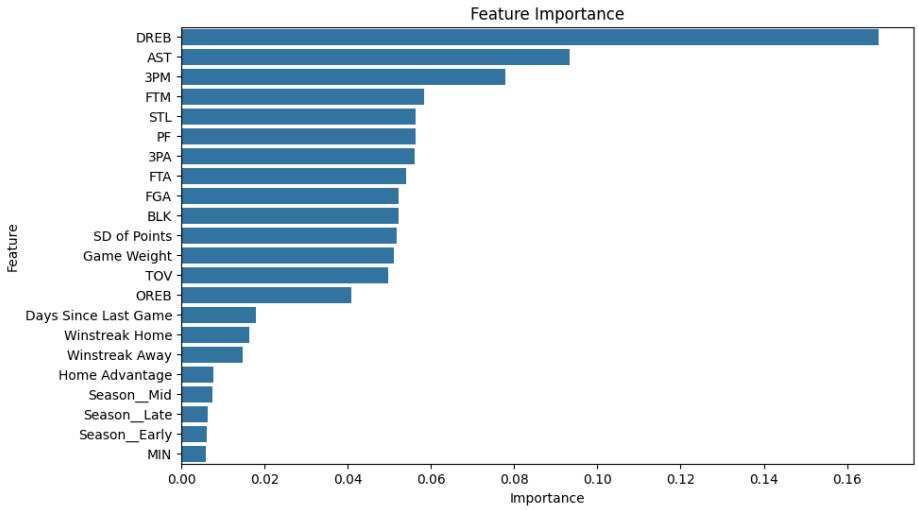
Figure 3. Confusion Matrix of Ensemble Method

Figure 4. ROC Curve of Classification Methods



This graph measures the performance of various classification models based on their receiver operating characteristic (ROC) curves. The area under the curve (AUC) values indicate model effectiveness, with SVM and LDA achieving the highest AUC of 0.90, followed closely by Logistic Regression and Ensemble methods at 0.89. Models like KNN and QDA exhibit relatively lower AUC values, suggesting reduced predictive performance. The results highlight the significance of model selection in accurately predicting NBA game outcomes.

Figure 5. Feature Importance (%)



The bar chart of feature importance ranks the impact of each feature to the accuracy of the ensemble model. Defensive rebounds, assists, and 3 point field goals contribute most to the model.

5. Conclusion

This project highlights the power of feature engineering and model optimization in predicting NBA game outcomes based on historical game statistics. By exploring the dataset of 2,460 entries, we identified key predictors such as home court advantage, win streaks, and rest days, which significantly influence game results. Through careful data preprocessing, redundant features were removed to reduce complexity and improve model interpretability. We implemented multiple classification models, including Logistic Regression, KNN, Random Forest, SVM, and Ensemble methods, achieving strong predictive performance. Among these, XGBoost, SVM, and Ensemble models demonstrated the highest accuracy and AUC scores, reflecting their robustness in handling game outcome predictions.

In order to maximize a team's chance of winning, they should make lots of defensive rebounds and attempt lots of 3 point field goals. Similarly, their willingness to work together and make assists will greatly improve their performance. Our findings emphasize the importance of incorporating situational factors and historical trends in sports analytics. The insights gained from this analysis can inform team strategies, improve forecasting accuracy, and serve as a foundation for further research into predictive analytics in sports. Future work could explore additional features, like player-level statistics or in-game events, to refine predictions further and enhance model performance.