Statistics 101C Final Project

Predictive Analysis of NBA Match Outcomes

Aashman Rastogi

Agastya Rao

Alison Hui

Ashley Chan

Carlotta Gherardi

Celine Zoe Tambrin

December 14, 2024

Department of Statistics, UCLA

Contents

1 Introduction

In this report, we analyze the NBA 2023-2024 Dataset, which provides game statistics for NBA teams. The dataset contains information on 2,460 games, capturing metrics such as points scored, field goals attempted, rebounds, assists, and other team performance indicators. The objective of this analysis is to predict the outcome of an NBA game—win or loss—using pre-game historical data.

To achieve this, we engineered features to capture team capabilities, recent performance trends, and home-court advantage. These features include but are not limited to, aggregated historical game statistics for individual teams and calculated differences between opposing teams, highlighting their relative strengths and weaknesses. We also implemented feature selection techniques to enhance predictive accuracy while maintaining model interpretability.

We trained and evaluated several machine learning models on this task, including Logistic Regression, Support Vector Machines, Random Forest, and Quadratic Discriminant Analysis. Initial experiments using a Random Forest classifier with unweighted historical averages produced a baseline testing accuracy of 67%. By incorporating weighted averages that emphasized recent games and adding the home-court advantage feature, we were able to improve prediction accuracy to 72%. Additional features, such as team stability and prior head-to-head outcomes, were explored to further enhance the model's performance.

This report details the steps taken for data preprocessing, feature engineering, and model evaluation. It also discusses the improvements achieved through feature enhancements and provides insights into the comparative performance of the models tested.

2 Data Preprocessing

The original, raw dataset has 2,460 rows and 24 columns. Each row represents a game played between two teams in the NBA 2023-2024 season. Each column is a metric related to the games. Several preprocessing steps were implemented to prepare the NBA dataset for analysis and modeling to clean and transform the data and ensure it was suitable for predictive modeling.

2.1 Handling Missing Data

To address missing values in the dataset, we made adjustments to a few key columns. The `FT%` column, which represents free throw percentages, contained entries marked as `'-'` to indicate missing values. These entries were replaced with a placeholder value of 1 to retain data completeness. While this placeholder may not accurately represent real-world

performance, it allowed the column to remain usable in feature engineering. Beyond the `FT%` column, other missing values across the dataset were filled with 0. This ensured the dataset remained complete and avoided potential errors during analysis.

2.2 Encoding Categorical Variables

To make the dataset compatible with the models we utilized, categorical variables were converted into numerical formats. The `W/L` column, which indicates whether a team won or lost a game, was encoded as 1 for a win and 0 for a loss. This transformation allowed the column to be used as the binary target variable for classification tasks. Additionally, a new binary column, `Home`, was created to represent home-court advantage. The `Home` variable was derived from the `Match Up` field, with a value of 1 assigned to games played at home and 0 to games played away. Additionally, the `Game Date` column was converted to a standard date-time format to support time-sensitive computations. This conversion allowed for the filtering and sorting of the dataset based on game dates, enabling aggregation of historical data.

2.3 Creating Unique Identifiers

We also introduced unique identifiers to maintain game-level consistency and support feature engineering. The opposing team for each game was extracted from the `Match Up` column by removing the current team's name and the home/away indicator. To support pairwise analysis, a combined identifier, `Team Pair`, was created by concatenating and sorting the names of the two teams in each game. This ensured that matchups were represented symmetrically, regardless of the home or guest team. A final identifier, `Game ID`, was constructed by appending the game date to the `Team Pair`, creating a unique label for each game.

2.4 Restructuring and Transforming the Dataset

The dataset originally consisted of one row per team per game. To focus on the relationship between opposing teams, we transformed the structure to represent differences in performance metrics between teams in each game. For each game, historical metrics such as points scored, rebounds, and assists were aggregated into averages based on each team's previous games. These averages were then subtracted to compute differences between the home and guest teams, highlighting their relative strengths and weaknesses.

To prioritize recent performance, we introduced weighted averages using an exponential decay function, where the unnormalized weights are defined as:

$$w_i = e^{-0.05 \cdot \frac{i}{5}}$$

These weights were then normalized so that they sum to 1. This method assigned higher weights to more recent games, capturing trends over time and emphasizing temporal relevance in the features.

Additionally, to ensure sufficient historical data for these calculations, the first 500 rows of the dataset were excluded from the analysis. This ensured that every game included in the modeling process had at least eight prior games of historical data for both teams.

The table below demonstrates the transformation process using the game between the Minnesota Timberwolves (MIN) and the Oklahoma City Thunder (OKC) on November 28, 2023. This example illustrates how raw game data is transformed into the features used for predicting game outcomes. For each metric, the "Raw Row" column displays the team's actual performance during the game, while the "Weighted Avg (Team)" and "Weighted Avg (Opponent)" columns represent the weighted averages of prior games for MIN and OKC, respectively. The "Difference (Stored)" column highlights the calculated difference between these weighted averages, which captures the relative strengths of the two teams and serves as an input feature for the machine learning models. For brevity, only the first five rows of metrics are displayed as an example.

Table 1: Weighted Average Calculation Example for MIN vs. OKC on 2023-11-28

Metric	**Raw Row**	**Weighted Avg (Team)**	**Weighted Avg (Opponent)**	**Difference (Stored)**
PTS	106	114.375	119.0	-4.625
FGM	33	42.0	44.142857	-2.142857
FGA	78	83.5	86.571429	-3.071429
FG%	42.3	50.35	51.085714	-0.735714
3PM	14	11.25	14.0	-2.75

2.5 Standardizing the Dataset

To standardize the dataset and ensure that features were on comparable scales, we applied standard scaling to ensure each feature had a mean of 0 and a standard deviation of 1, ensuring that features with larger numerical ranges, such as points scored or rebounds, did not disproportionately influence the models. Binary variables were excluded from this scaling process.

3 Baseline Model

3.1 Process

Our baseline model did not use a weighted average and did not incorporate any engineered features. The rows in the dataset used for the baseline model consisted of the the difference between the average past metrics of the home team (taken from the home team's past home games only) and the average past metrics of the guest team (taken from the guest team's past guest games only).

3.2 Results and Analysis

After preparing the dataset for analysis, we used 75% of the data for training and 25% for testing. Using all the raw features without doing any feature selection, we built our machine learning models. Here are the results:

Table 2: Training and testing performance using raw data without any feature selection

Model	Testing Accuracy	Training Accuracy
Logistic Regression	0.7020	0.6510
QDA	0.6265	0.6122
SVM	0.7	0.7163
Random Forest	0.6898	0.9367

However, using all the raw features without doing any feature selection presents itself with certain limitations, the main ones being overfitting and not detecting multicollinearity issues. In order to tackle these limitations, we applied LASSO for feature selection.

LASSO selected the following features: `FG%`, `3PM`, `DREB`, `AST`, `TOV`, `+/-`

Below are the results of applying the various machine learning models using only the features selected by LASSO:

Table 3: Training and testing performance using raw data without any feature selection

Model	Testing Accuracy	Training Accuracy
Logistic Regression	0.7041	0.6388
QDA	0.7082	0.6367
SVM	0.7122	0.6837
Random Forest	0.6898	0.8883

Compared to the model that used all the raw features, the testing accuracy is higher for logistic regression, quadratic discriminant analysis and support vector machine using this model with fewer features. This makes sense because removing irrelevant features allows

the model to focus on the most predictive ones, resulting in improved performance on testing data.

4 Feature Engineering

In our predictive analysis of NBA match outcomes, feature engineering played a pivotal role in improving the performance of machine learning models. Below, we detail the steps taken to create and refine features, as well as the rationale behind their inclusion.

- **Home-Court Advantage:**
  - We included the `Home` column as a binary feature, which indicates whether the basketball game is played at the home court. This was excluded in the baseline model. This addition improves the accuracy of the second model, as home-court advantage can significantly impact game outcomes: it is always expected that the home team will perform better compared to the away team due to the home-court advantage.
- **Weighted Averages of Performance Metrics:**
  - We incorporated the weighted averages of the performance metrics in the baseline model for both the home and guest teams, computed separately for each team's past games at home and away, respectively. Performance metrics for each game include scoring metrics like points scored (`PTS`), efficiency metrics like field goal percentage (`FG%`), and three-point percentage (`3P%`), rebounding metrics like rebounds (`REB`), assist metrics like assists (`AST`), possession metrics like turnovers (`TOV`), and specialty metrics like steals (`STL`). The weights are computed using exponential decay ($\lambda = 0.05$), making more recent games more important than older games. This gives a better representation of the team's current form.
- **Home vs. Guest Performance Differences:**
  - Unlike the baseline model, which uses the raw performance metrics without transformations, we now include the difference between the weighted averages of metrics for the home and guest teams. For each game, the model calculates the difference between (1) weighted average past metrics for the home team based on previous home games and (2) weighted average past metrics for the guest team based on previous away games. This provides a relative measure of performance differences between the competing teams.
- **Feature Selection Using LASSO:**
  - Both models use LASSO for feature selection. However, the input features to LASSO differ due to the engineered features (e.g., weighted averages, team

differences, and inclusion of the Home binary variable) introduced in the Revised model. For the Revised model, the features selected by LASSO are field goal percentage (`FG%`), defensive rebounds (`DREB`), turnovers (`TOV`), plus-minus (`+/-`), and home court (`Home`).

5 Results and Analysis from Revised Model

After training each of our 4 models for the hyperparameters identified, we evaluated each on the testing data, which made up 25% of the dataset after preprocessing. The accuracy of each model is given below.

Table 4: Training and testing performance across classical models

Model	Testing Accuracy	Training Accuracy
Logistic Regression	0.7245	0.6456
QDA	0.7184	0.6422
SVM	0.6898	0.6640
Random Forest	0.6633	0.8578

As reflected above, Logistic Regression was our best performing model with a testing accuracy of 0.7245 and a training accuracy of 0.6456. This result is not surprising, as Logistic Regression is well-suited for binary classification tasks. It calculates the probability of class membership based on a weighted sum of input features, making it effective when the underlying relationship between the features and the target is approximately linear.

The Revised Model was selected as the best model over the Baseline Model due to its superior performance in predictive accuracy and its incorporation of meaningful features that enhanced the model's understanding of the game context. While the Baseline Model relied on raw features and applied LASSO for feature selection, it failed to consider the inherent dynamics of the home advantage and the temporal significance of recent games.

The Revised Model addressed these shortcomings by introducing a weighted average of past metrics that prioritized recent games and adding a binary indicator for home advantage. These enhancements captured critical game context and improved the model's ability to predict match outcomes, as demonstrated by higher test accuracy across various classifiers.

While the Revised Model demonstrated significant improvements in predictive accuracy by leveraging a weighted average of past metrics and incorporating a binary indicator for home advantage, its performance plateaued when additional basic features were included. To further explore the potential for improvement, the below outlines additional engineered features that capture team dynamics, historical interactions, and contextual factors such as stability, momentum, and past game performance differences. These features aimed to refine the model's understanding of nuanced relationships that influence match outcomes.

6 Additional Models

To further enhance the predictive capabilities of our baseline model, we explored additional engineered features and evaluated their impact on various machine learning algorithms. This section details the supplementary features introduced, the methodology employed to incorporate them, and the subsequent results derived from these extended models.

6.1 Additional Engineered Features

Building upon the initial feature set, we engineered several new features aimed at capturing deeper insights into team performance and historical interactions. The rationale behind each feature is outlined below:

- **Past Head-to-Head History**:
  - **Win Difference (`Win_Diff`)**: Calculated as the difference in the total number of wins between the two teams from their previous matchups.
  - **Average Margin Difference (`Avg_Margin_Diff`)**: The difference in average win/loss margins from past games between the teams.
  - **Win-Loss Ratio Difference (`Win_Loss_Ratio_Diff`)**: The difference in win-loss ratios, categorized into `Class_Win_Loss_Ratio_Diff` based on thresholds (+1 for a difference greater than 0.3, −1 for less than -0.3, and 0 otherwise).
- **Team Stability**:
  - `Team_Stability` and `Opponent_Stability`: Measured by the standard deviation of points scored (`PTS`) in past games, reflecting the consistency of each team's performance.
  - `Stability_Diff`: The difference between the team's stability and the opponent's stability.
- **Momentum**:
  - `Team_Balance` and `Opponent_Balance`: Calculated as the difference between the number of wins and losses over the most recent 15 games, indicating each team's current form.
  - `Balance_Diff`: The difference between the team's balance and the opponent's balance.
- **Interaction Features**:
  - Although we explored features such as the ratio between average team offensive rebounds and average opponent defensive rebounds, and the difference in rest days before a game, these did not contribute positively to the model's performance and were excluded from the final feature set.

The above features were meticulously engineered to encapsulate both historical performance and recent trends, with the expectation that they would provide the models with a richer context for making accurate predictions.

6.2 Results

After integrating the additional engineered features, we proceeded to train and evaluate the models to assess the impact of these enhancements. The following steps were undertaken:

1. **Feature Selection with LASSO**: To mitigate potential overfitting and enhance model interpretability, we employed LASSO regression for feature selection. This method penalizes less significant features, effectively reducing the feature space to the most impactful variables.
2. **Model Training and Evaluation**: Utilizing the selected features, we trained four machine learning models—Logistic Regression, Quadratic Discriminant Analysis (QDA), Support Vector Machines (SVM), and Random Forest—on the training dataset and evaluated their performance on the testing set.

6.2.1 Feature Selection Outcomes

LASSO regression identified the following features as significant predictors: `DREB` (Defensive Rebounds), `TOV` (Turnovers), `+/-` (Point Differential), `Balance_Diff` (Balance Difference)

These features were deemed most relevant in distinguishing between winning and losing outcomes, while the other engineered features did not receive non-zero coefficients and were thus excluded from the final model training.

6.2.2 Model Performance Metrics

The performance of each model, using only the LASSO-selected features, is summarized below:

Table 5: Training and testing performance across classical models

Model	Testing Accuracy	Training Accuracy
Logistic Regression	0.6776	0.6571
QDA	0.6776	0.6571
SVM	0.6735	0.6578
Random Forest	0.6776	0.8415

6.2.3 Analysis of Results

The incorporation of additional engineered features did not lead to an improvement in prediction accuracy. In fact, the performance metrics indicated a slight decline compared to the baseline model, which achieved a testing accuracy of 72

- **Logistic Regression, QDA, and SVM**: These models exhibited similar test accuracies around 67.3% to 67.76%, which is below the baseline performance.
- **Random Forest**: While the test accuracy was comparable to the other models (67.76%), the training accuracy was notably higher (84.15%), suggesting potential overfitting. This overfitting indicates that the model may have captured noise in the training data, reducing its generalizability to unseen data.

6.2.4 Possible Reasons for Underperformance

Several factors may have contributed to the lackluster performance of the additional features:

- **Feature Redundancy**: Some engineered features may have overlapped in the information they provided, leading to redundancy without offering new predictive power.
- **Overfitting**: The inclusion of numerous features increases the complexity of the model, which can cause it to overfit the training data, as observed with the Random Forest model.
- **Irrelevant Features**: Not all engineered features were equally relevant. Features like `Class_Win_Loss_Ratio_Diff` and certain stability metrics did not significantly contribute to the prediction task.
- **Insufficient Feature Transformation**: Some features might benefit from further transformation or interaction terms to unlock their predictive potential.

In summary, while the additional engineered features did not enhance the predictive performance in this iteration, they provide a foundation for further experimentation and refinement in subsequent analyses.

7 Conclusion

In this report, we undertook a comprehensive analysis of the NBA 2023-2024 Dataset with the primary objective of predicting the outcomes of NBA games—specifically, determining whether a team would win or lose based on pre-game historical data. Our

approach encompassed meticulous data preprocessing, strategic feature engineering, and the evaluation of multiple machine learning models to achieve accurate and interpretable predictions.

7.1 Summary of Findings

7.1.1 Data Preprocessing and Feature Engineering

The initial phase involved cleaning and transforming the raw dataset, which comprised 2,460 games and 24 performance metrics. We addressed missing data by imputing appropriate values, encoded categorical variables to facilitate model compatibility, and created unique identifiers to maintain consistency across game records. A significant portion of our feature engineering efforts focused on capturing team capabilities, recent performance trends, and home-court advantage. Notably, we developed aggregated historical statistics and calculated differences between opposing teams to highlight their relative strengths and weaknesses.

7.1.2 Baseline Model Performance

Our baseline model, utilizing a Random Forest classifier with unweighted historical averages, achieved a testing accuracy of 67%. By incorporating weighted averages that emphasized more recent games and integrating the home-court advantage feature, we enhanced the prediction accuracy to 72%. This improvement underscored the importance of recent performance trends and the inherent advantage teams possess when playing on their home court.

7.1.3 Impact of Additional Engineered Features

In an effort to further bolster model performance, we introduced additional engineered features such as past head-to-head history, team stability, and momentum indicators. However, the incorporation of these features did not yield the anticipated enhancements. Models trained with the expanded feature set exhibited test accuracies ranging from 67.3% to 67.76%, which were marginally below the baseline performance. The Random Forest model, in particular, demonstrated signs of overfitting, with a training accuracy of 84.15% compared to its test accuracy.

7.2 Implications and Future Work

The findings from this analysis highlight the delicate balance between feature complexity and model performance. While initial feature engineering efforts yielded promising results,

the subsequent addition of more features without addressing redundancy and overfitting concerns led to diminished performance. Moving forward, the following strategies could be employed to enhance model accuracy and robustness:

- **Refined Feature Engineering**: Develop more nuanced features that capture unique aspects of team dynamics and performance without introducing redundancy. This may involve domain-specific insights or leveraging advanced statistical techniques.
- **Advanced Feature Selection**: Implement more sophisticated feature selection methods, such as Recursive Feature Elimination (RFE) or tree-based feature importance measures, to identify and retain the most impactful features while eliminating those that contribute little to no predictive power.
- **Regularization Techniques**: Apply stronger regularization methods to prevent overfitting, particularly in complex models like Random Forest. Techniques such as cross-validation and hyperparameter tuning can also be beneficial.
- **Model Ensemble Methods**: Explore ensemble strategies that combine multiple models to leverage their individual strengths and mitigate their weaknesses. Techniques such as stacking, boosting, or bagging could offer improved performance.
- **Incorporation of External Data**: Augment the dataset with additional relevant information, such as player statistics, injury reports, or team morale indicators, to provide a more comprehensive basis for predictions.
- **Cross-Validation and Robust Evaluation**: Employ more rigorous cross-validation strategies to ensure the stability and generalizability of model performance metrics. This can help in assessing how well the models perform across different subsets of data.

7.3 Final Thoughts

Predicting the outcomes of NBA games is inherently challenging due to the multitude of dynamic factors influencing each game. While our analysis achieved a respectable baseline accuracy, the journey towards more accurate and reliable predictions necessitates continuous refinement of data preprocessing techniques, feature engineering strategies, and model evaluation methodologies. By addressing the limitations identified in this study and exploring the proposed future directions, there is significant potential to enhance the predictive capabilities of machine learning models in the realm of sports analytics.