NBA Dataset Performance Prediction

Jisu Chae(906287138), Mukhil Guna (806290565), Evan Lum (506231681), Elise Pham (106229897),

Ainsley Strang (905992057), Kenneth Chow (106283177)

Stats 101C - Lecture 2

Shirong Xu

13 December 2024

1. Introduction

This paper explores the prediction of the outcome of basketball games based on data for the 2023-2024 season. The dataset contains 2,460 entries and 24 columns, with game statistics like points, assists, and blocks of teams. Using methods taught in our STATS 101C class: Statistical Modeling and Data Mining, our goal is to get at least a 70% accuracy when predicting the outcome of a basketball game between two different teams using the data of both teams before a random given day. After deriving our features, we trained our model using Gradient Boosting, Support Vector Machines (SVM), Quadratic Discriminant Analysis(QDA). Linear Discriminant Analysis (LDA), Logistic Regression, and Random Forest. We found that Logistic Regression has the highest testing accuracy score of 0.7081.

1.1 Exploratory data analysis

Let's first breakdown our dataset and analyze the key statistics for the 2023-2024 season consisting of 82 games. Teams scored between 73 and 157 points, averaging 114 points per game, with a field goal percentage of 47.5% (ranging from 27.7% to 67.1%). Rebounds averaged 43.5 per team, ranging from 25 to 74. Our first visualization shows that PTS, AST, and FG% closely follows a normal distribution best, compared to all other variables. We centered the variables around 0 as +/- in order to indicate a team's net game performance.

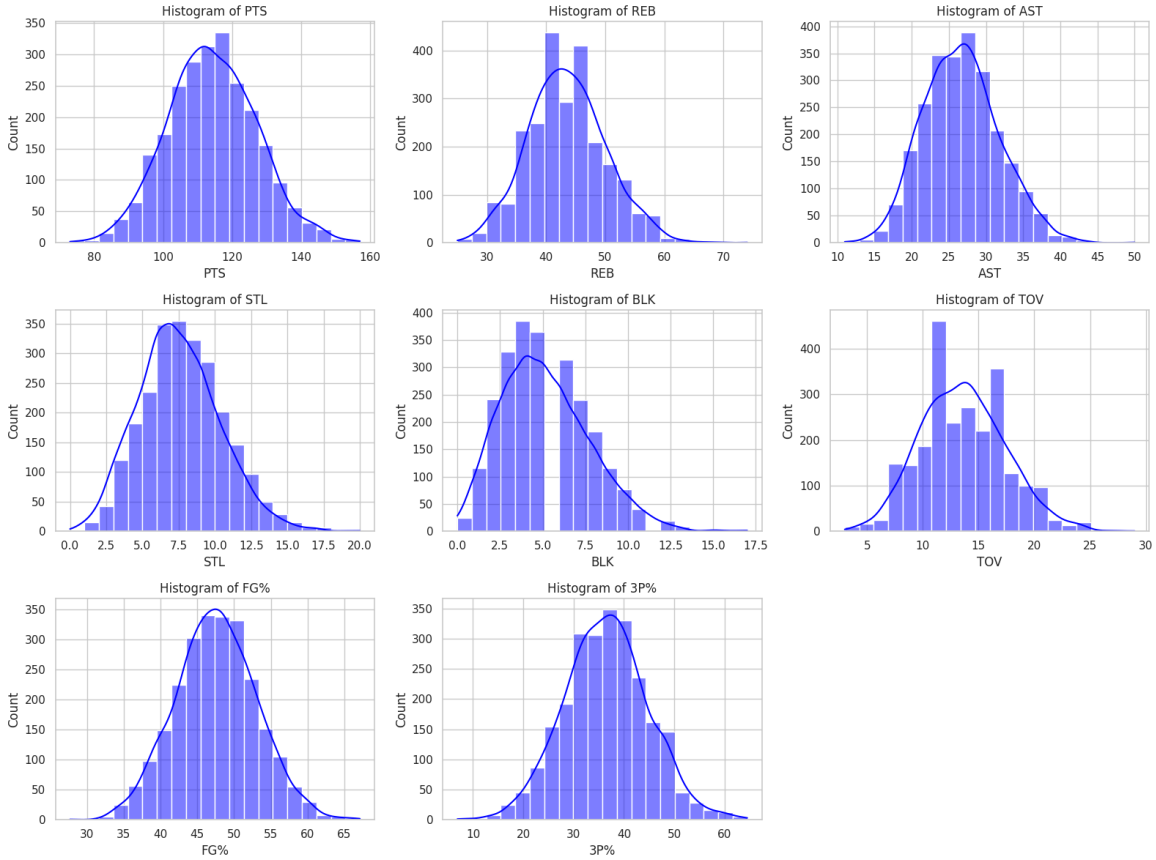

Figure 1.1 Normal Distribution Curves of Sample Variables

Our second visualization highlights the total number of points scored by each team. This metric is a strong predictor of NBA game outcomes, reflecting the overall scoring capability and offensive efficiency.

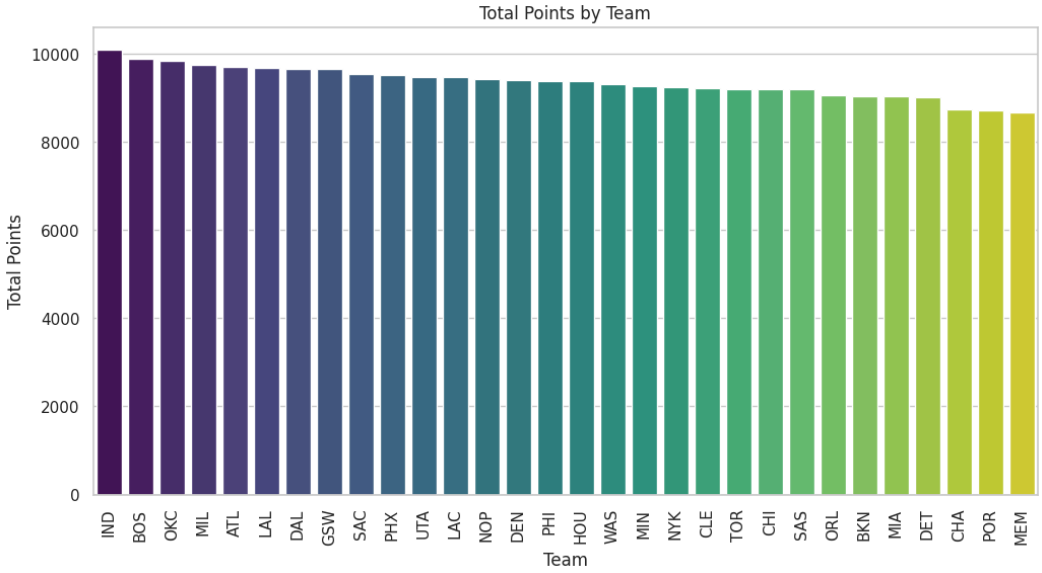

Figure 1.2 Total Points Scored by Each NBA Team During the 2023-2024 Season

Our third visualization graphs the distribution of points through a binary Win/Loss outcome. It is naturally expected that the more points a team scores, the more likely that they will win the game.

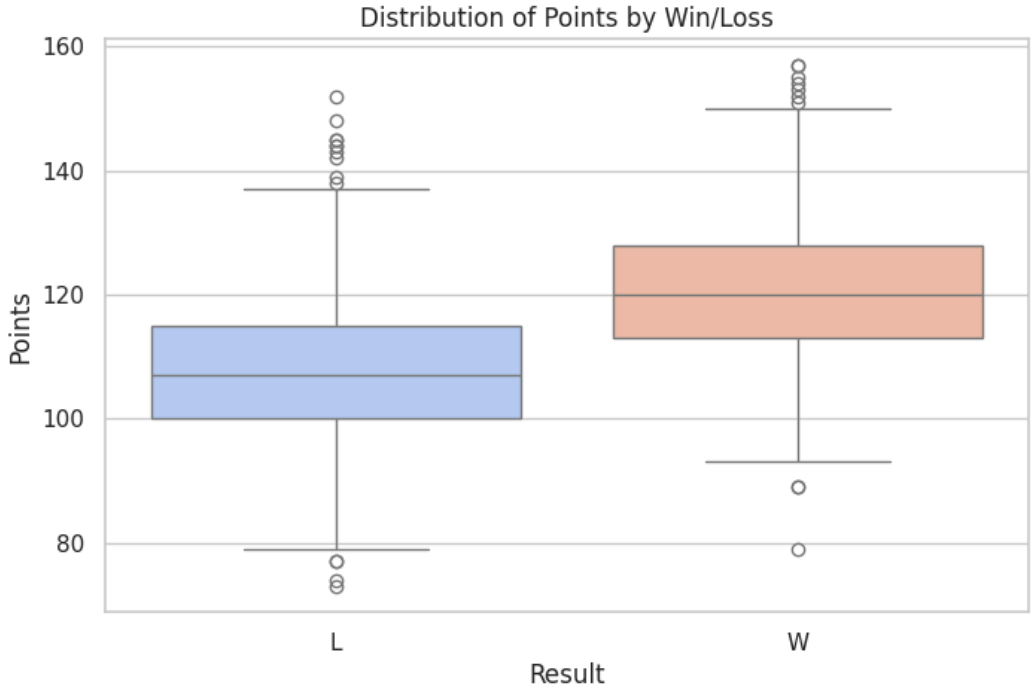

Figure 1.3 Total Distribution of Points by Win/Loss

Our final EDA visualization charts the average points scored over the season. The points vary over time, but it seems mostly consistent due to the games being independent from one another. However, it is noted that there was a sudden increase in points scored in early January 2024 and a sudden decrease from late February 2024 to early March 2024.

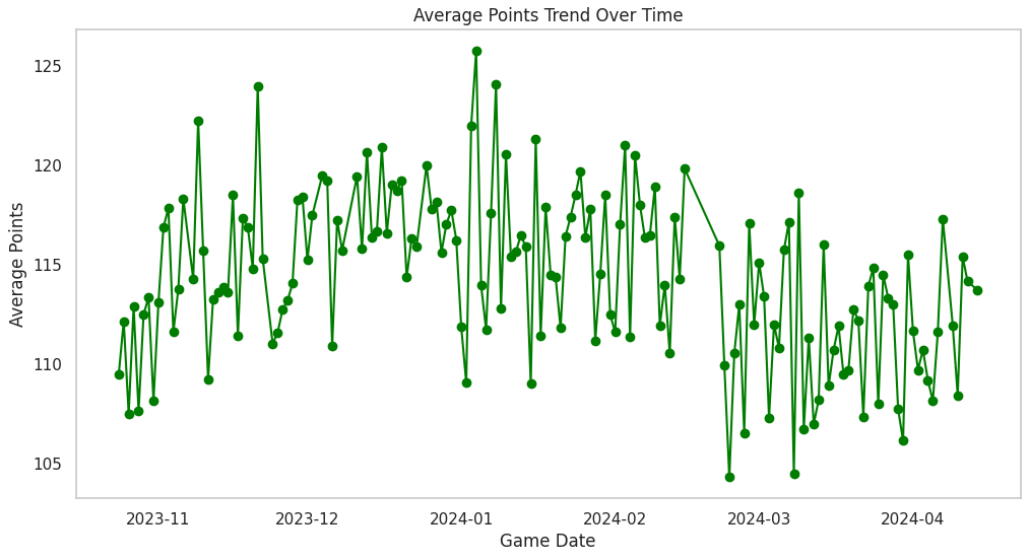

Figure 1.4 Trend of Average Points Over the Season

2. Data Pre-processing

We needed to process our raw data prior to training our models. Our preprocessing included extracting only certain columns for prediction of a team's success, developing weighted functions by identifying the variables and relationships relevant to our features, and denoting our observations as a binary class. In doing so, we were able to have an organized dataset that was ready to be modeled by our machine learning algorithms. We started with converting our target variable, “W/L”, into a numerical binary of 0s and 1s for easier manipulation.

2.1 Home Advantage

Firstly, we created a binary home and away feature by analyzing the game locations. Games marked with ‘vs’ were encoded as 1 which represented home. Games marked with ‘@’ were encoded as 0 and represented away. This separation was crucial and important to implement as it is historically documented that home court advantage plays a large role in determining the outcomes of basketball games.

2.2 Weighting Average

The idea behind the weighting function is that the games near the prediction date are more important as they provide a more accurate reflection of a team’s current capabilities. This function assigned higher weights to more recent games. We believe that a team’s recent performance better reflects a team’s current

form, taking into account factors like injuries, roster changes, coaching decisions, making for a stronger predictor for game outcomes. We also used α (alpha) as an exponential decay parameter. The weights are normalized to sum to 1 to ensure that there is proper scaling of the historical performance metrics. The function we used was: $w_i = \frac{exp(-\frac{\alpha * i}{5})}{\sum_{j=1}^{L} exp(-\frac{\alpha * i}{5})}$, with L representing the length of the data and i representing the index of the current element (from 1 to L).

2.3 Stability

Our team chose to judge stability as a feature to predict a team's future success. By considering the points statistic as a rolling variance over time, we could offset any teams that have a high variance in their points per game, leading to the team's struggle to win consistently. Variance is calculated with the following formula: $\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$ where x_i is the points for each game and $\bar{x}$ is the average points scored by the team. A team with a lower variance, means that their scoring is stable and consistent throughout their games, which plays a big factor in a team's chances of winning.

2.4 Feature Type Standardization & Data Scaling

We then converted all our numerical data into float type in order for consistency. This included points, field goal metrics, three point statistics, free throw data, and other metrics that determined performance. All of our features were standardized to have zero mean and unit variance using sklearn's StandardScaler. This allowed fair comparison between features of different scales.

2.5 Removing Extraneous Columns

Immediately, we removed columns regarding defensive plays such as "Blocks", "Assists", "Defensive Rebounds", and such. While these metrics can determine a game's success, we wanted to focus on offensive metrics. These variables were therefore impractical to our research. In addition, due to limited resources, our research focused on highlighting three main features rather than analyzing an exhaustive list of variables that contribute to the predictors.

3. Experimental Setup

After completing the feature engineering process, we decided to consider the subset of features that we would use when building our predictive models. First, we decided to check the correlations between our predictors.

As we can see below, variables that are closely related have the greatest correlation with one another. This includes three-point field goals made, number of free throws made, and offensive rebounds. This further proves the number of points scored and how you scored them plays a very important factor in determining a game outcome. Conversely, other features such as turnovers and personal fouls showed weaker

correlations, most likely because they have a more indirect or inconsistent impact on the final result. Therefore, we can rule these features out.

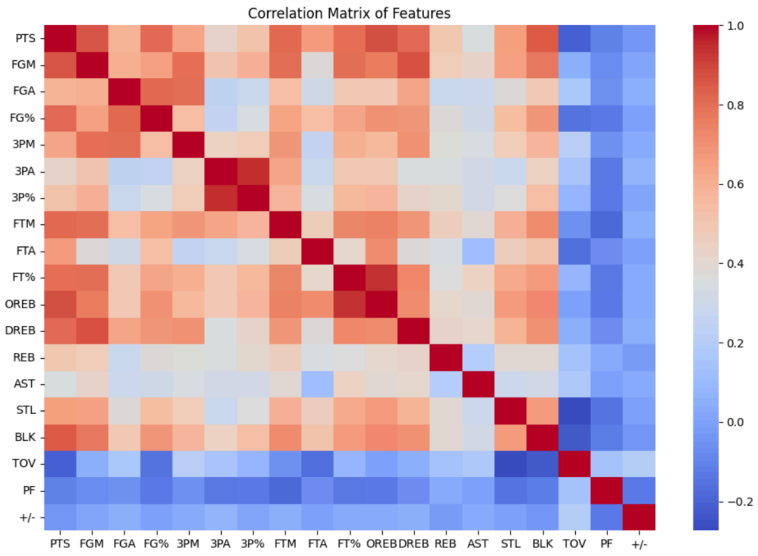

Figure 3.1 Features Correlation Heatmap

With great correlation, our next step was to implement principal component analysis (PCA) as an attempt to reduce the dimensionality of the feature space. We performed PCA on all features and found that the first 2-5 components explain the majority for the majority of the variance around 0.78 to 0.9.

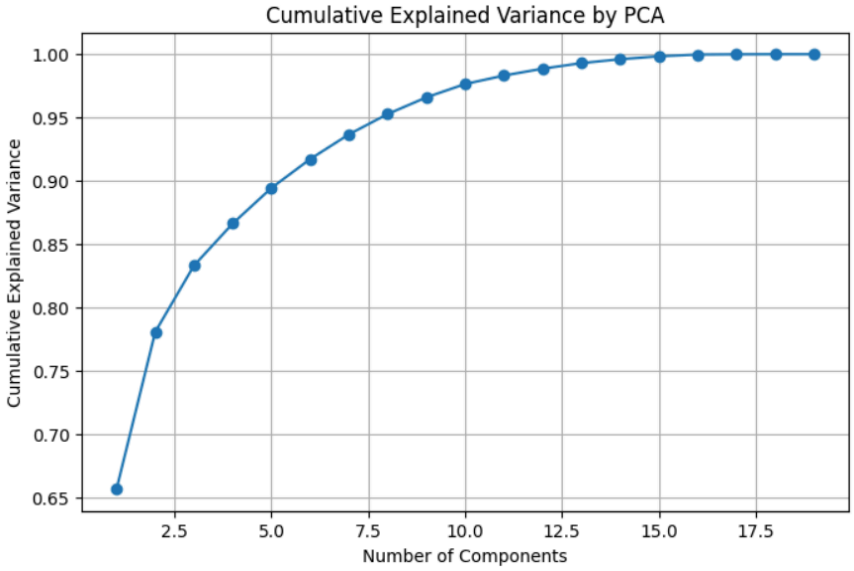

Figure 3.2 PCA Cumulative Explained Variance of Components

The elbow method suggests 10 parameters for modeling, but we chose to simplify it by using just 3 variables: PTS, Match up, and FT% while leaving W/L as our target variable. This balances model

simplicity with performance and makes the model easier to interpret and practical. Therefore, we decided not to combine any feature vectors and left it as is.

We then used 5-fold cross validation to determine the best predictors for each model. We chose to experiment with SVM, Gradient Boosting, Logistic Regression, LDA, and Random Forest. Below are examples of how the models were tested. It is worth noting that some models had fewer features to examine, potentially limiting their predictive performance. As a result, we decided to stick with our originally planned features.

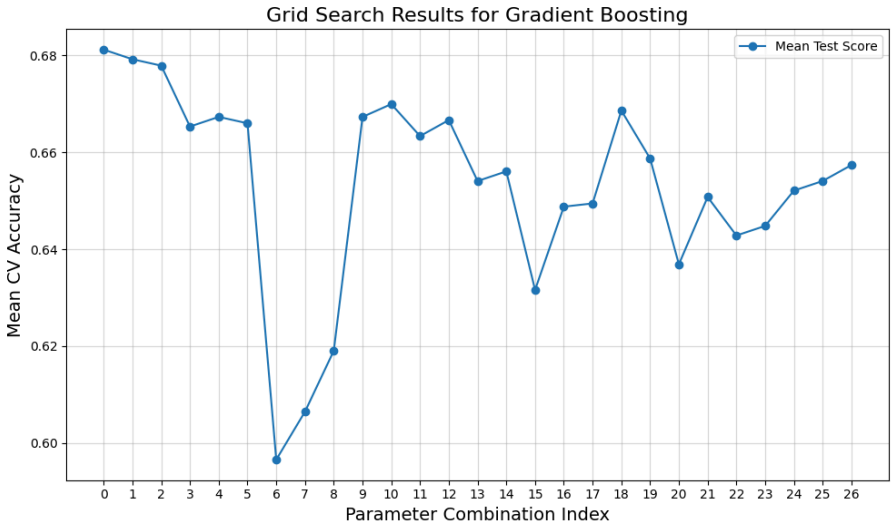

Figure 3.2a 5-Fold Cross Validation Gradient Boosting

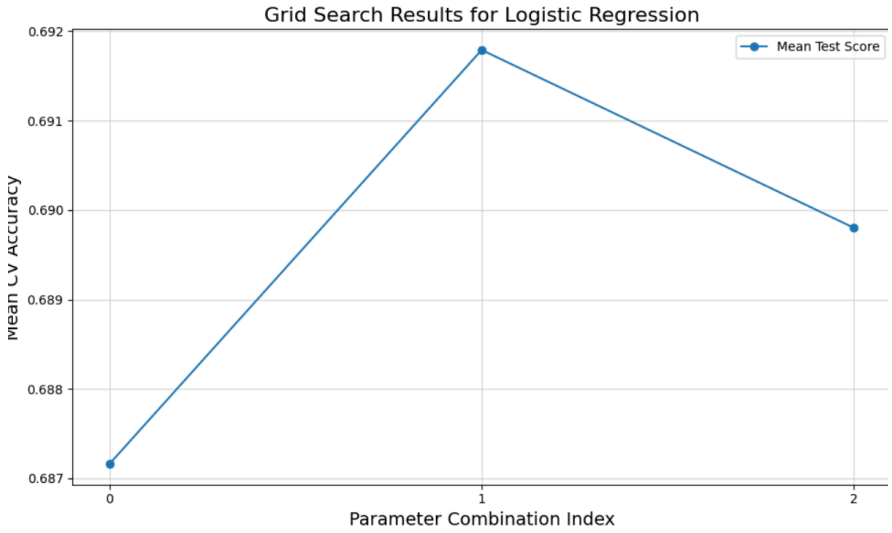

Figure 3.2b 5-Fold Cross Validation Logistic Regression

With our list of features finalized, our next step was to split our dataset into a training and testing subset. Because we wanted to use past data to predict future outcomes, we needed to skip the first 500 games to ensure that the models would have enough historical games to work with for all teams in the dataset. The remaining 2100+ data points would be split into 75% training and 25% testing, the first 500 games would be used for training. We used the same models as mentioned earlier to fit the training data and evaluate the performance on the testing set. The process of fitting each of the five models is detailed below.

1. **Logistic Regression** - The first model that we attempted was logistic regression. We first initialized the model, and applied L2 or penalty for ridge regression to shrink coefficients. By tuning the model using GridSearchCV, we found the best parameters for Logistic Regression.
2. **Logistic Discriminant Analysis (LDA)** - Another method we tried to use was LDA, which is used to model linear decision boundaries. No tuning was done and was trained on testing data.
3. **Random Forest -** These methods help reduce variance and overfitting by building multiple decision trees and choosing the class with the most votes. This concept was used because our data contains many features that can be separated into classes. This model helps counteract the multicollinearity from the features. Number of trees and tree depths was chosen, and tuned by GridSearchCV (sklearn library), where the best model was chosen from cross - validation.
4. **Gradient Boosting -** This method builds trees sequentially and corrects the errors before it. This idea was considered because the data consists of lots of complex interactions. This method is good for non-linear relationships, but still has issues with overfitting. The parameters that were tuned were the number of trees, tree depths, and learning rate.
5. **SVM -** This method creates a hyperplane between data to best maximize the distance to separate the points of each class. This works well with higher dimensional data but has high computational cost and overfits data. The method was tuned using GridSearchCV (sklearn library) for the penalty parameter, kernel type, and kernel coefficients.

4. Results and Analysis

After implementing and tuning five different models, we were able to evaluate their performance using 5-fold cross validation.

Model	Testing Accuracy	CV Mean Accuracy
Logistic Regression	0.7102	0.7081
LDA	0.7000	0.6998
Random Forest	0.6900	0.6878
Gradient Boosting	0.6900	0.6854
SVM	0.6900	0.6823

As reflected above, Logistic Regression turned out to be our best performing model with testing accuracy of 71.02% and a consistent cross-validation score of 70.81%. This consistent performance was apparent across different evaluation approaches. This is reasonable as logistic regression estimates the probability of an event occurring given a set of independent variables. This is especially clear as we chose to represent three features, one of which had a binary classification label. The model could be further improved if we had considered additional features. Logistic regression did a better job at generalizing and avoiding overfitting or underfitting to our data, compared to the other models.

Due to limited resources and time, our team was restricted to a few of the most popular modeling techniques. Employing other algorithms such as AdaBoost, autoencoders, or hidden Markov Models could have given vastly different or better results. In the future, we would like to improve on these or even increase the hyperparameters of our Random Forest model. It is also noted that our dataset was confined to only one basketball season, not nearly enough games to train the model accurately and effectively. We could have also tried ensemble methods and combined our top-performing models.

Our analysis demonstrates that machine learning can effectively predict NBA game outcomes. Although it is difficult to entirely predict every game's outcome with 100% accuracy due to the inherent variability that exists in games, our different models allowed for a clear understanding of which factors most strongly influence game outcomes.