Stats 101C Final Project
Predicting NBA Wins

STATS 101C | Prof. Shirong

Tessa Gervase (406229527), Cade Miller (006284059), Rory Freck (306145948), Max Chalekson (505696407), Vikas Sundar (205926235), Marc Walden (905979937)

Table of Contents

[1] Introduction

In this paper, we discuss our analysis of the NBA 2023-2024 Dataset, which is an extensive collection of game statistics that tries to predict the game outcome, (as a Win or Loss). The dataset has a total of 2,460 entries and 24 columns, and contains important features like points scored, assists, turnovers, rebounds, etc. To make sure we followed the project guidelines, we constructed features only based on historical game data that was available before each game. Some notable engineered features to keep in mind include point differentials, turnovers, cumulative weighted averages of prior game statistics, and binary indicators for home advantage.

To proceed with our task at hand, we used a wide range of machine learning models, like Logistic Regression, Random Forest, and Quadratic Discriminant Analysis (QDA), in addition to feature selection techniques. The use of recursive feature engineering allowed us to explore new dimensions, including differential defensive and shooting efficiencies, mapping gameplay dynamics with a higher granularity.

Our most optimal model was our Logistic Regression one, which achieved a consistent testing accuracy between **70.2%** and **71.4%**, highlighting its strength under different feature engineering scenarios. Other models had their own strengths and weaknesses, such as Decision Tree classifiers provided interpretable predictions, while QDA showed limitations in handling the complex interactions in the dataset, achieving a lower performance. These results demonstrate the important role of feature engineering, especially when weighing recent games more heavily and using home advantage as a predictor.

Our report summarizes the data preprocessing, feature construction, and experimental design processes, following the guideline of using only pre-game data. We also review the strengths and weaknesses of each model, find ways to further optimize it, and suggest refinements to enhance predictive performance.

[2]Data Preprocessing

Before constructing any models, our data needed to be processed. This included creating binary variables via one-hot encoding, feature engineering viable predictors, and handling sparse columns in our dataset.

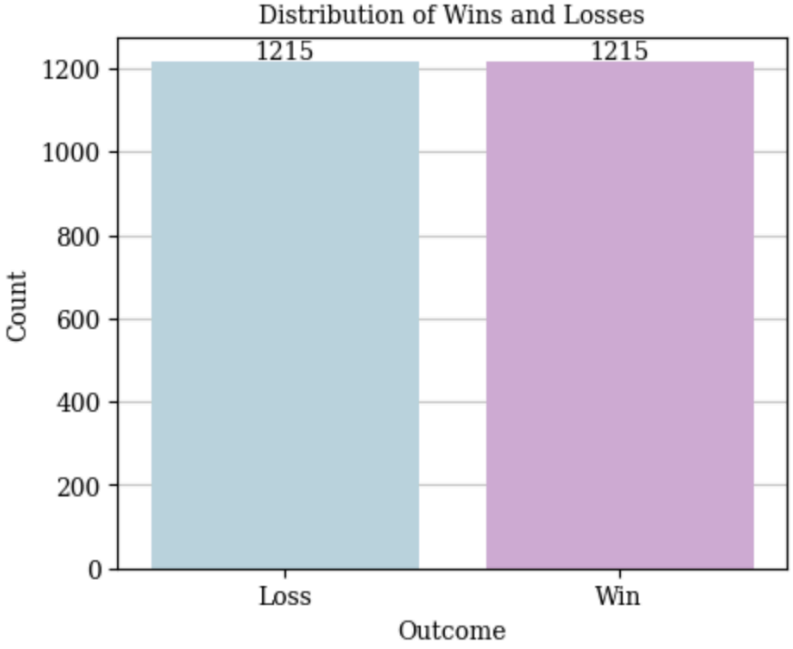
Table 5: Descriptive Statistics of Selected Features

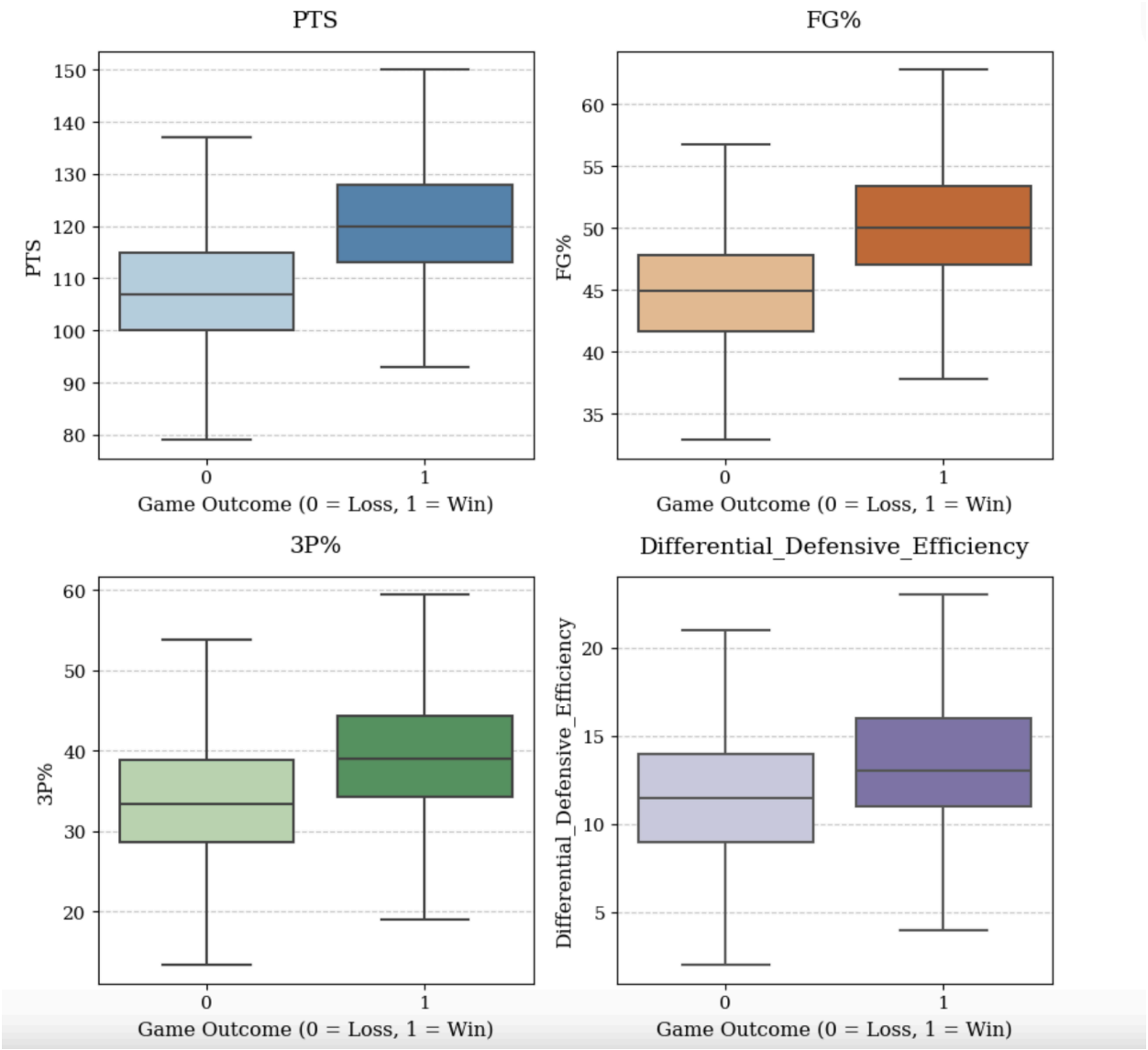
	W/L	MIN	PTS	FGM	FGA	FG%	3PM	3PA	3P%	FTM
Count	2460.000	2460.000	2460.000	2460.000	2460.000	2460.000	2460.000	2460.000	2460.000	2460.000
Mean	0.500	241.362	114.211	42.170	88.903	47.521	12.837	35.104	36.494	17.034
Std	0.500	6.351	12.846	5.343	7.013	5.498	3.837	6.542	8.341	5.890
Min	0.000	240.000	73.000	26.000	67.000	27.700	2.000	12.000	6.900	0.000
25%	0.000	240.000	105.000	38.000	84.000	43.800	10.000	30.000	31.000	13.000
50%	0.500	240.000	114.000	42.000	89.000	47.500	13.000	35.000	36.550	17.000
75%	1.000	240.000	123.000	46.000	93.000	51.200	15.000	39.000	41.700	21.000
Max	1.000	290.000	157.000	65.000	119.000	67.100	27.000	63.000	64.500	44.000

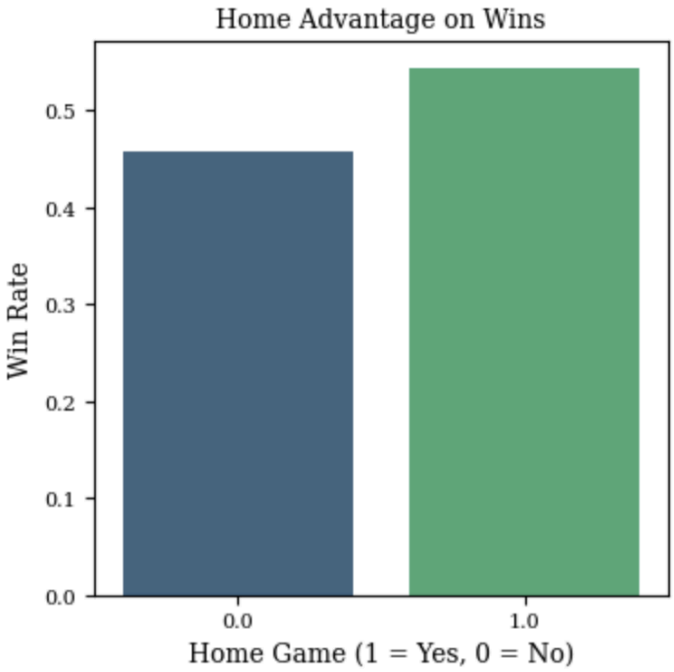
This is a summary table of some of our variables prior to any feature engineering. Based on this, we will process the data in order to improve the quality of our variables. Additionally, we will explore potential transformations or interactions between them to improve model performance. After these preprocessing steps, the data will be ready for feature selection and model training, ensuring that we can build a more robust predictive model.

2.1) Creating the Binary Variable (Win/Loss)

The foundation of the entire project rests on our model's ability to accurately predict if a team is going to win or lose an individual game based on the statistics of that same game. To do this, we must first encode our dependent, or "predicted" variable, "W/L". Naturally, we chose to one-hot encode wins with a 1 and losses with a 0 for ease of interpretability.


Distribution of Wins and Losses
1215
1215
1200
1000
800
600
400
200
0
Count
Loss
Win
Outcome


PTS
PTS
150
140
130
120
110
100
90
80
0
1
Game Outcome (0 = Loss, 1 = Win)
FG%
FG%
60
55
50
45
40
35
0
1
Game Outcome (0 = Loss, 1 = Win)
3P%
3P%
60
50
40
30
20
0
1
Game Outcome (0 = Loss, 1 = Win)
Differential_Defensive_Efficiency
Differential_Defensive_Efficiency
20
15
10
5
0
1
Game Outcome (0 = Loss, 1 = Win)



Another consideration when dealing with classification algorithms is class balance (or imbalance) within our dependent variable. Fortunately, by construction, our dataset has an equal number of wins and losses. This allows us to disregard the possibility of misleading accuracy scores when comparing, choosing, and improving our models.

2.2) Creating Differential Dataframe

To prepare the data for our model to predict future game outcomes from past game data, we created a differential data frame from the dataset provided. First, we converted the 'Game Date' column to panda's datetime format for easy sorting. We then sort the dataset to be in chronological order. Using this newly sorted information, for each scheduled game, we then calculate the difference in the average of the two team's stats prior to that game date. The result was a feature-rich dataset that summarizes past performance and conditions leading into each game, giving us the foundation we need to build predictive models to forecast future game outcomes.

To produce our data frame, we used the following exponential decay formula:

$$Weight = e^{-\lambda(Days\ Before)}$$

This has a decay factor, λ. A higher λ gives more weight to recent games while older games have less influence. We wanted to ensure this is the case when creating our data frame because in the NBA, we can assume a game from last week would be more indicative of the team's standing than a game result from last year. This could result from player morale, injured/benched players, newly developed skills, etc. We tested multiple λ's and found $\lambda = 0.05$ to produce the most statistically significant results.

2.3 Feature Engineering

The motivation behind feature engineering was to improve model accuracy by capturing a more holistic and better-weighted set of predictor variables. We based the features we engineered on their practicality. The most statistically significant ones were "Home Advantage", "Defensive Efficiency", and "Shooting Efficiency". We will now discuss the motivation behind the construction of these features.

"Home Advantage" is exactly what it sounds like, a home court advantage. This was the first feature we engineered because teams tend to perform better in familiar environments, benefiting from the support of a home crowd, reduced travel fatigue, and familiarity with the court's nuances. The most recent meta-analyses show that home teams win around 60% of games, making it significant in predicting game outcomes.

"Defensive Efficiency", which we defined as the sum of steals and blocks, is a strong predictor of NBA wins as it reflects a team's ability to disrupt the opponent's offensive plays. Teams with high defensive efficiency can limit scoring opportunities and create fast-break chances, directly impacting game outcomes. After all, everyone knows defense wins championships

"Shooting Efficiency", which we calculated as the ratio of made field goals, three-pointers, and free throws to total attempts, measures a team's scoring effectiveness per opportunity. Teams with higher shooting efficiency are more likely to capitalize on possessions, leading to more points scored and therefore a greater expectation of winning games. This is the most holistic quantification of offensive ability.

After all feature engineering, we ended with the following variables in our models:

Table 4: Table of All Features

Index	Original Column	Preprocessing Action	Final Column
0	Home	Engineered as binary (1 for home games, 0 for away games)	home
1	Differential_Defensive_Efficiency	Calculated based on basketball-specific metrics	Differential_Defensive_Efficiency
2	Differential_Shooting_Efficiency	Calculated based on basketball-specific metrics	Differential_Shooting_Efficiency
3	PTS	Retained directly from dataset	PTS
4	FG%	Retained directly from dataset	FG%
5	3PM	Retained directly from dataset	3PM
6	3PA	Retained directly from dataset	3PA
7	REB	Retained directly from dataset	REB
8	AST	Retained directly from dataset	AST
9	STL	Retained directly from dataset	STL
10	BLK	Retained directly from dataset	BLK
11	TOV	Retained directly from dataset	TOV
12	FT%	Retained directly from dataset	FT%

2.4 Filling Missing Values

When working with the NBA dataset for the project, to predict the wins, we need to account for the missing values in the dataset. This was done to ensure the integrity of the modeling and analyses done for this project. Through the different modeling methods that we've attempted: decision tree, logistic regression, random forest, and quadratic discriminant analysis (QDA) - numerical missing values in the feature matrix were replaced with 0.

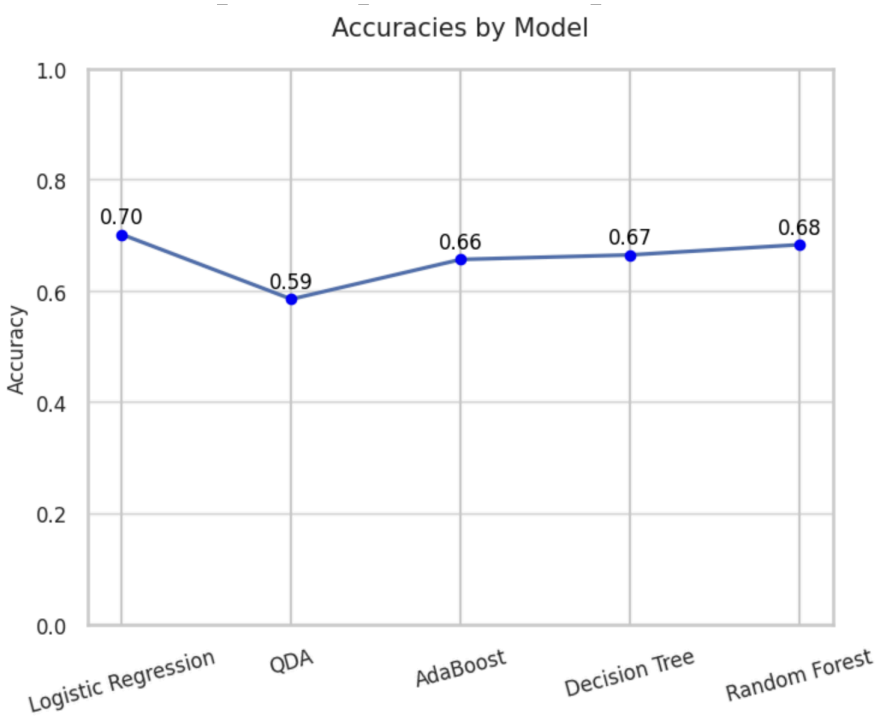
When looking at the FT% column, it contained potential non-numeric entries, which were converted to `NaN` and filled with 0. This approach allowed us to avoid data loss and maintain consistency within our calculations, especially during the model training, and ensure the accuracy of each of the models.

3 Experimental Setup

3.1 Comparing Model Performance

To assess the performance of each model, we compared their mean accuracies. We accomplished this by running 5-fold on each model and calculating their respective mean accuracies. First, we calculated the accuracy of each model using the validation set of each fold. Accuracy is determined by dividing the number of correct predictions by the total number of predictions. Then, we averaged the accuracies

across all folds to obtain the mean accuracy for the model. This procedure was repeated for all models twice, first using 5-fold cross-validation and then 10-fold cross-validation. The results are as follows:



3.2 Model Selection

Table 1: Confusion Matrix - Logistic Regression

True Label	Predicted: Loss	Predicted: Win
Loss	173	73
Win	73	171

Table 3: Classification Report

Class	Precision	Recall	F1-Score	Support
Class 0	0.70	0.70	0.70	246
Class 1	0.70	0.70	0.70	244
Accuracy		0.70		490
Macro Avg	0.70	0.70	0.70	490
Weighted Avg	0.70	0.70	0.70	490

We chose logistic regression as our model. Using mean accuracy as the performance metric, logistic regression outperformed the other models on unseen data. Additionally, logistic regression allows for meaningful interpretations of the results

through the use of odds. By exponentiating the coefficients of the logistic function, we can interpret them similarly to how we interpret coefficients in linear regression. This makes it easier to understand the relationship between features and the target variable, providing actionable insights for team performance evaluation and strategy development.

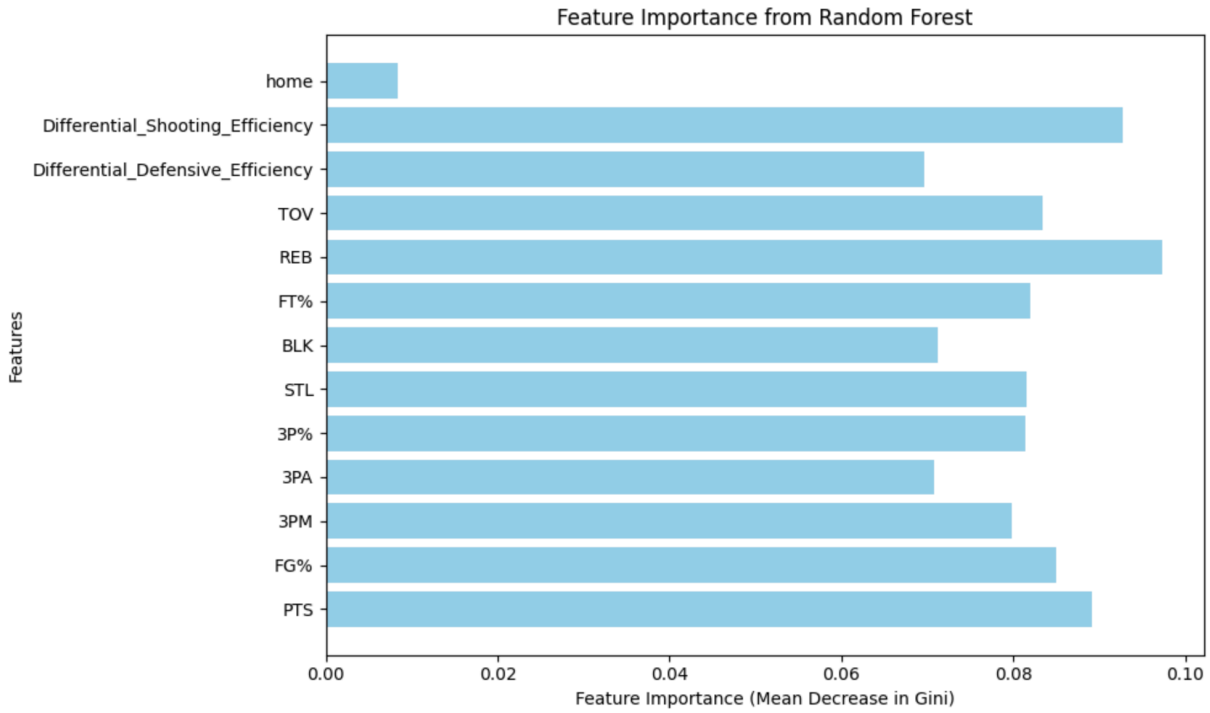
4 Results and Analysis

After selecting the logistic regression model, we wanted to ensure that all preexisting or engineered predictors significantly contributed to the model's success. To do so, we ran tests to check the significance of our variables and finalized our model.

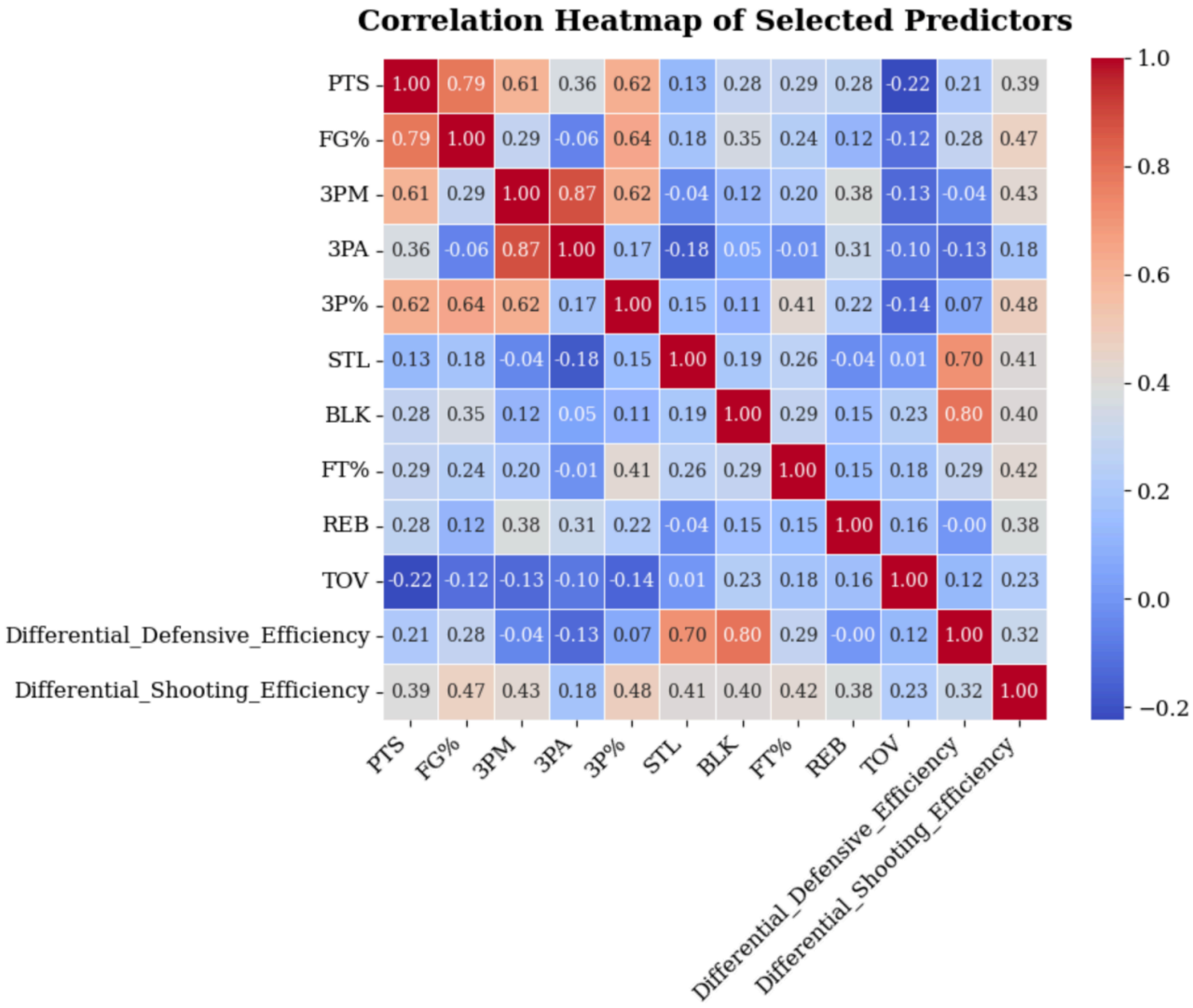
4.1 Model Enhancement

Once we selected logistic regression as the best model in ***3.2***, we wanted to examine the variables used in the model. This is not only to ensure no variables are negatively impacting our model but also to check if there are any variables we may need to add. We chose to use a Random Forest model to determine the significance of our features since it provides importance scores for each feature based on how well it contributes to reducing impurity in splits. As we learned in class, this is using the mean decrease in Gini as a test for feature selection.

First, we split the columns from our original logistic regression model using a 25/75 split. Using the training data, we trained the Random Forest model for feature selection using the built-in functions from scikit-learn. Then, we fit the model and applied it to discover feature importance yielding the following plot:



This showed us that Home_Advantage is contributing to the success of our model significantly less than the rest of our predictors. Removing this predictor did not decrease the accuracy of our model, so for our final product, we left it out.



4.2 Final Model Results

After evaluating several models and refining our approach, we reached our final selection based on a combination of model performance and interpretability. The final model results are derived from the cross-validation performance of the models we explored, including Logistic Regression, Decision Tree, Random Forest, AdaBoost, and Quadratic Discriminant Analysis (QDA).

As shown in Table 1, Logistic Regression consistently performed well across both 5-fold and 10-fold cross-validation, with a mean accuracy hovering around 83%. Moreover, it also outperformed the other models in terms of other metrics, including a Recall and F-1 score of around 83 - 84%, showing its ability to effectively predict true positives and true negatives. This demonstrates a high level of consistency and

generalizability, reinforcing its selection as the final model. Its performance is particularly strong due to its ability to balance the various features and prevent overfitting. Additionally, we tried incorporating both L1 and L2 penalty norms to enhance the model's ability to generalize to new data. After testing various values for the penalty lambda, we observed only minor improvements in accuracy. Ultimately, we chose to use the standard logistic regression model making it easier to interpret.

Quadratic Discriminant Analysis (QDA) was the second-best performer. While slightly behind Logistic Regression, QDA outperformed models like Decision Tree and Random Forest. Its ability to capture nonlinear relationships and handle multivariate data made it well-suited for predicting NBA game outcomes. However, due to its more complex assumptions and lower interpretability, it was not selected as the final model but remains a strong alternative.

The Random Forest model was another strong contender. It performed slightly worse than Logistic Regression, with a mean accuracy of 80.89% in 5-fold cross-validation and 80.65% in 10-fold cross-validation. Despite being slightly less accurate, Random Forest's ability to capture complex relationships between features made it an appealing choice. The model's ensemble nature, which involves aggregating predictions from multiple decision trees, contributes to its robustness.

AdaBoost performed similarly to Random Forest, with mean accuracies of 80.61% for 5-fold and 80.98% for 10-fold cross-validation. AdaBoost's ability to improve model performance through a weighted combination of weak learners allowed it to slightly outperform Random Forest in some cases, but it still did not surpass Logistic Regression in terms of overall accuracy. Finally, while Decision Trees provide an easy-to-understand structure, they are prone to overfitting, especially when dealing with complex datasets like the NBA game dataset.

Based on these results, Logistic Regression emerged as the final model. It achieved the highest accuracy with stable performance across cross-validation folds and offered the interpretability needed for actionable insights into NBA game predictions. The Random Forest and AdaBoost models were considered strong alternatives, especially for capturing non-linear interactions, but they were not as interpretable as Logistic Regression. Therefore, Logistic Regression was chosen for deployment, ensuring both high accuracy and the ability to derive meaningful insights for future strategic recommendations in NBA game predictions.

5 Conclusion

In conclusion, our analysis and modeling of the NBA 2023-2024 dataset provided valuable insights into predicting game outcomes. Through comprehensive preprocessing, feature engineering, and model testing, we identified key predictors like shooting efficiency and defensive efficiency, which played a crucial role in determining game outcomes. Among the various models tested, Logistic Regression emerged as the most reliable, delivering consistent performance with an accuracy of approximately 83%. Although other models, such as Random Forest and AdaBoost, showed competitive performance, Logistic Regression stood out for its ability to generalize well to unseen data by still being a fairly interpretable model. Moving forward, further refinements and additional feature engineering could enhance the model's predictive power, but our current approach provides a robust framework for predicting NBA wins.

Plots I want to build:

- Distribution of wins and losses (bar graph)
- Random forest (horizontal bar graph)
- Correlation plot for predictors used in model (heat map)
- Confusion matrix (in latex)
- Plot of odds with confidence intervals
- Coefficient plot (in latex)