Predicting the Results of NBA Games

Tony Owens

December 2022

Abstract

The National Basketball Association (NBA) is one of, if not the largest professional basketball leagues in the world. With this comes a multi billion dollar sports betting industry, where oddsmakers seek to find the best predictive models to get the advantage over everyday fans who place bets on games. This paper seeks to determine the accuracy of a predictive model created from publicly available data using modern Machine Learning (ML) methods. This paper focuses on several ML models such as Neural Networks, Random Forest, Naive Bayes, and K Nearest Neighbors, and also more simple methods such as logistic regression and a linear classifier.

In an attempt to improve the accuracy, the results were also tested on the same dataset balanced on the outcome variable using undersampling. The results of this paper indicated that the more complex ML methods were not particularly effective when compared to the more simple methods, as the measured accuracy for the games remained fairly constant. Balancing the data also proved to be ineffective as the accuracies for the balanced dataset were slightly lower than that of the unbalanced. The neural network was the most accurate model performing right at 70% accuracy with the rest of the models hovering between the mid 68s and low 69s with the exception of the RF model which performed poorly at under 66% accuracy. The accuracy of the models were in line with that of the rest of the literature exhibiting respectable, but not incredible performance. The lack of difference between the accuracies of the models could point to issues in the dataset, or could represent an error that simply cannot be explained without more information than what was available in the dataset.

Table of Contents

Chapter 1 Introduction

The NBA season features 30 teams each of which play 82 games (each team will play 41 games at their home stadium) which comes to 1230 games in total. There is then a play-in tournament where the teams ranked 7-10 in each conference play to qualify for the playoffs while teams ranked 1-6 qualify automatically. This paper will only focus on games played during the regular season. As the playoffs represent a smaller sample size of both teams and games.

The objective of this paper is to predict the result of NBA regular season games using statistics of both the home team and the away team. These statistics will include each team's cumulative stats for the season such as wins and losses, as well as offensive and defensive stats taken on a per game basis such as points per game and points allowed per game. This data will be used to create a dataset to feed into the predictive models used.

This paper seeks to add to the current state of the literature regarding NBA prediction problems by collecting data over a longer period of time, and introducing features that were not seen in previous papers.

Chapter 2 Literature Review

There is a plethora of research in predicting the results of NBA games, and sports matches in general. This research all focuses on the same question, how accurate can one predict the results of matches using statistics related to the teams involved. The best performance on test data observed was over 74% (Loeffelholz, Bednar and Bauer, 2009). This accuracy was achieved using a neural network which seems to be one of, if not the most effective tools for making NBA related predictions. Other NBA related questions such as selecting the All-Star team (Ji and Li 2013) have also been approached with neural networks. The 2009 paper was done using data from the 2008-2009 season with statistics from espn.com. This paper does not make use of multiple seasons, and the only context provided for the results is that of the expert opinions from USA today for each game.

There was research that examined a multi-year timeframe (Aryan and Sharafat) of 2008-2013. This paper utilized a dataset made from web scraping similar to that which this paper will use. This paper implemented both linear and logistic regression as well as a support vector machine (SVM) and the models performed at around 66-68% accuracy.

Also of note was research that made use of the NBA API for its data (Perricone, Shaw and Święchowicz) and implemented a k nearest neighbors model (KNN), which was not common in the literature. This was along with two SVMs, a ridge regression model and logistic regression. All of these models performed at around 66-67% accuracy, further solidifying the range of acceptable accuracy values for this problem as ranging from the mid 60s to the lower to mid 70s.

Predicting the results of sports competitions in general is a large space within the machine learning field and as a result there is similar research in other sports that can provide further insight into the techniques to use. In a paper predicting the results of matches in the National Football League (Boulier and Stekler, 2003). This paper included the betting odds for the games as a reference and concluded that the betting odds were the best predictor of the winner. This was followed by a probit model based on rankings of the teams done by the New York Times. This paper also included a "naive prediction" as a benchmark which was the result that would have been obtained if the model always predicted the home team to win. The data for these predictions takes place from 1994-2000. It is clear that there is a large space for predicting NBA games and sports in general, and this paper seeks to add to that space by incorporating a mixture of elements from past papers, while including other methods as well as making changes to the approach of the research.

Chapter 3 Dataset

3.1 Choosing the time frame

The dataset consists of all NBA seasons from the year 2000 to 2021. The idea behind such a large timeframe was to ensure the models had enough data to accurately generalize to NBA games as a whole.

3.2 Scraping the data

The dataset was created via web scraping in python. First the schedule was created using basketballreference.com. This was a dataframe which included information about all of the games such as the score, each team involved and which team was playing at home. There was then a statistics data frame made based on data from basketball reference and basketball.realgm.com which was joined based on the team name. The statistics data frame was then joined on the schedule dataframe and this process was repeated for every season involved. Finally the data frames were concatenated to create the final data frame consisting of 22,335 rows and 58 columns.

3.3 Filtering features

There were some features that needed to be removed such as the amount of points each team scored, the names of the teams, the date and year of the game and the index column. The removal of these unnecessary features took the final number of columns to 49.

3.4 Splitting the data

For all of the subsequent analysis the dataset was split into a training set, validation set, and testing set. These splits were done using the rsample package to balance the proportion of the outcome variable in each set. The first split was into test and non-test dataset which represented 25% and 75% of the data respectively. The second split was into training and validation sets from the nontest set with splits of 60% and 40% respectively.

3.5 Features

(The prefixes “Home” and “Away” are added to distinguish between each team’s stats)

Parameter	Type	Explanation
Target (Outcome Variable)	Categorical	1 if Home Team wins, Away Otherwise
FGM	Double	The amount of shots (excluding free throws) that a team makes in a game
FGA	Double	The amount of shots (excluding free throws) that a team attempts in a game
3PM	Double	The amount of three point shots that a team makes in a game

3PA	Double	The amount of three point shots that a team attempts in a game
FTM	Double	The amount of free throws a team makes in a game
FTA	Double	The amount of free throws a team attempts in a game
ORB	Double	The amount of offensive rebounds (rebounding a shot by your own team) a team gets in a game
DRB	Double	The amount of defensive rebounds (rebounding a shot by the other team) a team gets in a game
APG	Double	The amount of assists a team gets in a game
SPG	Double	The amount of steals a team gets in a game
BPG	Double	The amount of shots a team blocks in a game
TOV	Double	The amount of turnovers a team

		gets in a game
PF	Double	The amount of personal fouls a team commits in a game
W	Double	The amount of wins a team has for the season
L	Double	The amount of losses a team has in a season
W/L %	Double	The percentage of a team's games that they have won
MOV/A	Double	Average Difference in the team's score minus their opponent's score adjusted for the record of their opponents
ORTG/A	Double	Estimated amount of points scored per 100 possessions adjusted for the record of their opponents
DRTG/A	Double	Estimated amount of points allowed per 100 possessions adjusted for the record of their opponents

b2b	Boolean	1 if the team in question played a game the day prior to the game in question

Chapter 4 Approach

4.1 Establishing a Baseline

In order to contextualize the accuracy of the models, an initial analysis was performed. In the dataset the home team won 59.6% of the games played. For a model to be of any value, it would have to outperform this baseline, otherwise it would be more effective to predict the home team to win in every scenario.

4.2 Testing with some simple models

After establishing a linear classifier, logit and probit models were implemented. This would show the level of accuracy attainable using the more simplistic models. For all of the models the cutoff value was determined by minimizing the distance to the point (0,1).

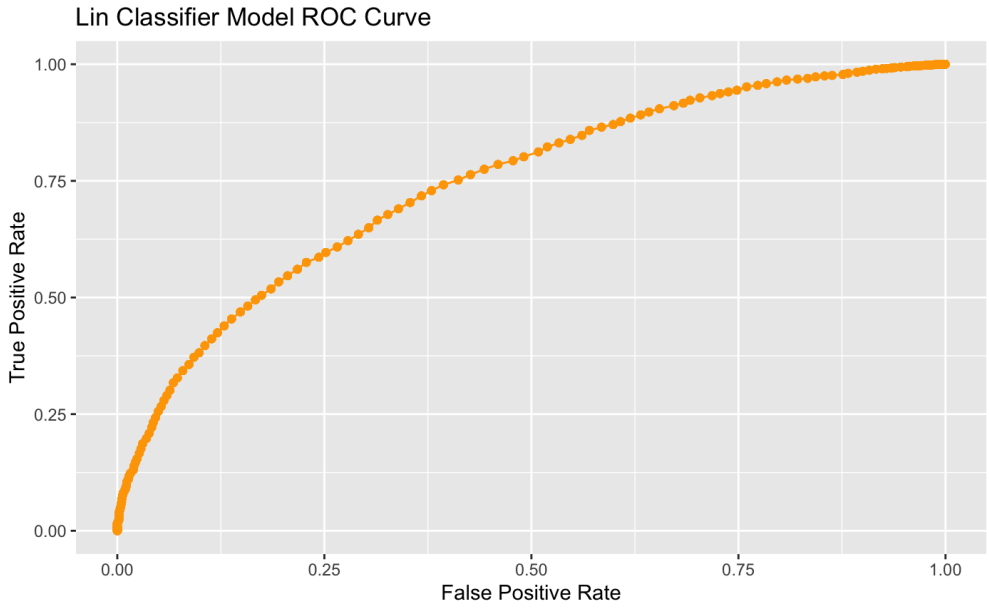

Figure 4.2.1: Tuning ROC curve for the Linear Classifier

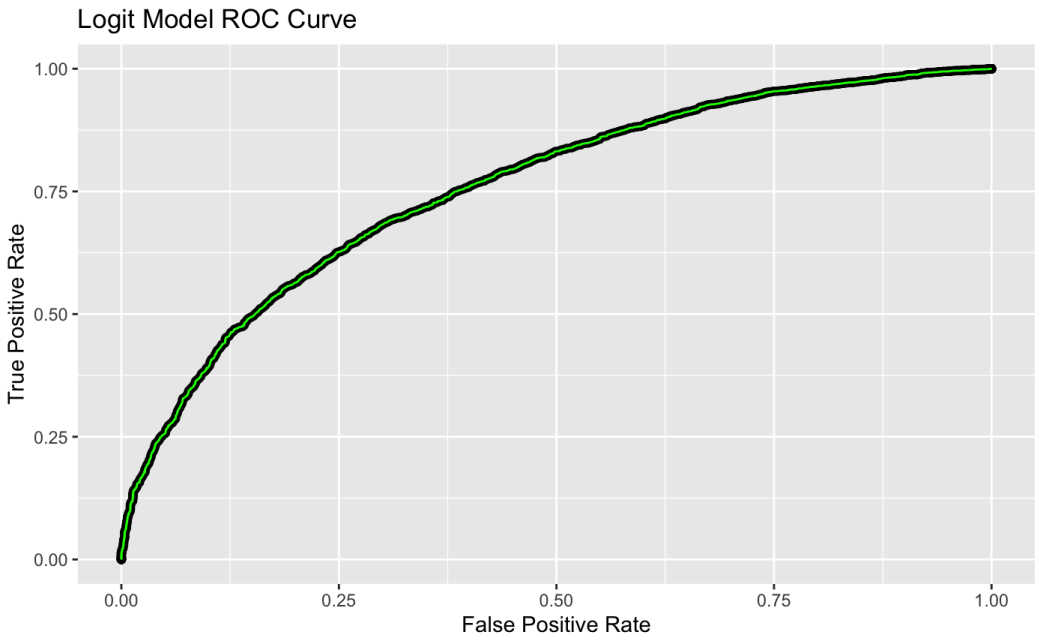

Figure 4.2.2: Tuning ROC curve for the Logit Model

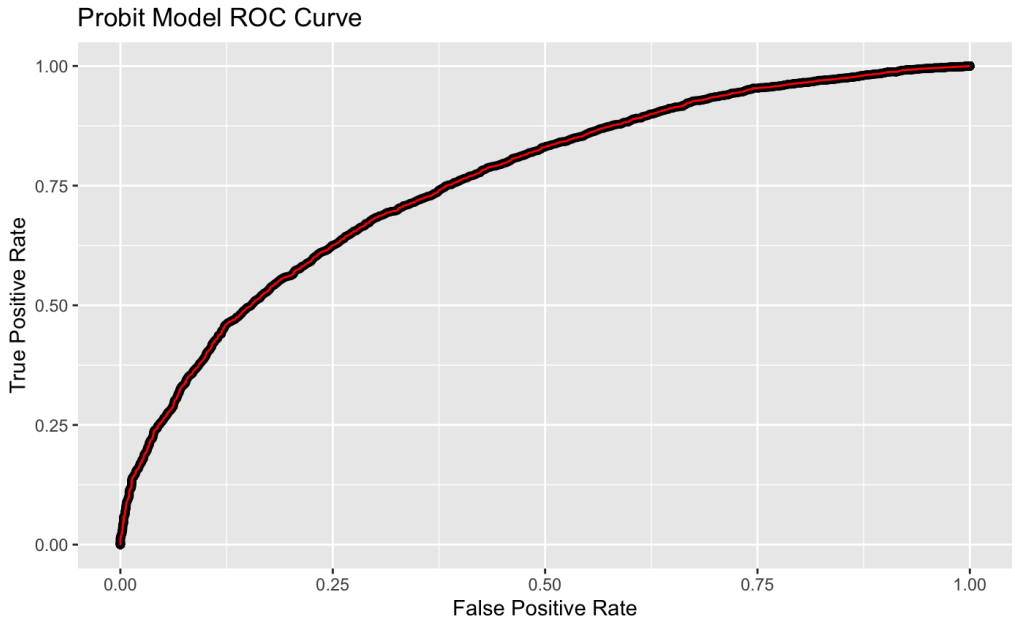

Figure 4.2.3: Tuning ROC curve for the Probit Model

These models all performed at a test accuracy of around 68-69% when tested with the optimal cutoff value determined by the ROC curve.

4.3 Testing with the more complex models

4.3.1 KNN

After the simple models, some of the more complex models were implemented, the first of which was the KNN. The KNN was tuned for the optimal K value using the validation set and then tested on the training set. It was determined that K = 241 gave the best accuracy from the plot below.

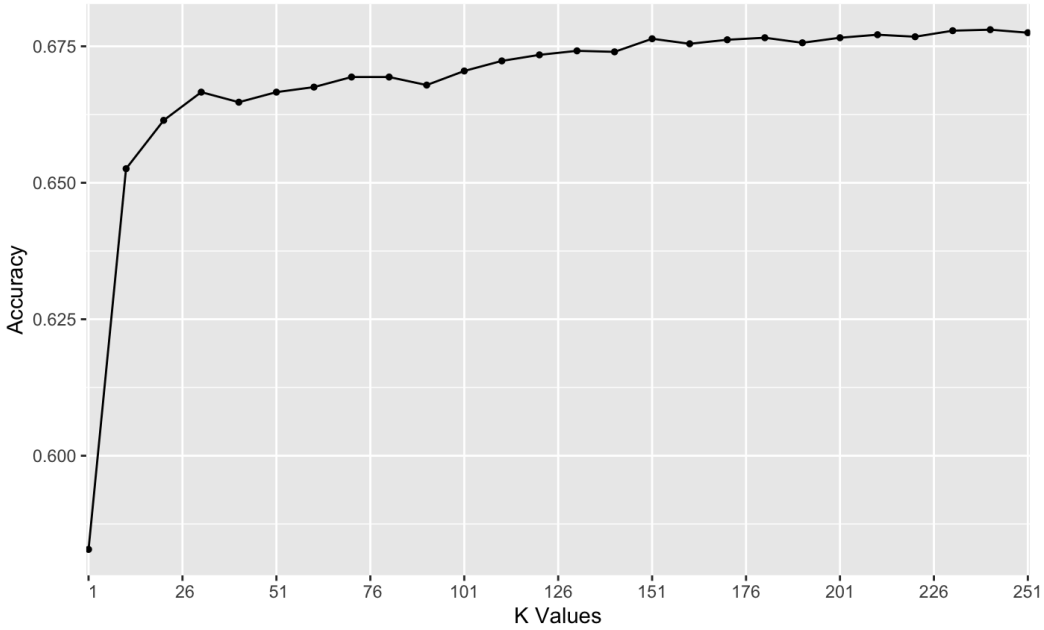

Figure 4.3.1.1: Tuning plot for the KNN (plots accuracy over K)

When tested with the optimal K value the accuracy was around 68%, similar to that of the linear models

4.3.2 Naive Bayes

For the naive bayes model, there was no tuning required, and the accuracy stayed around 69% when tested. This was in line with the linear classifier, and logit & probit models while slightly outperforming the KNN.

4.3.3 Random Forest

The random forest model required much more tuning with the validation set. This model was tuned for the number of trees as well as the mtry parameter, the number of variables to be randomly sampled. After tuning the optimal number was found to be 1.

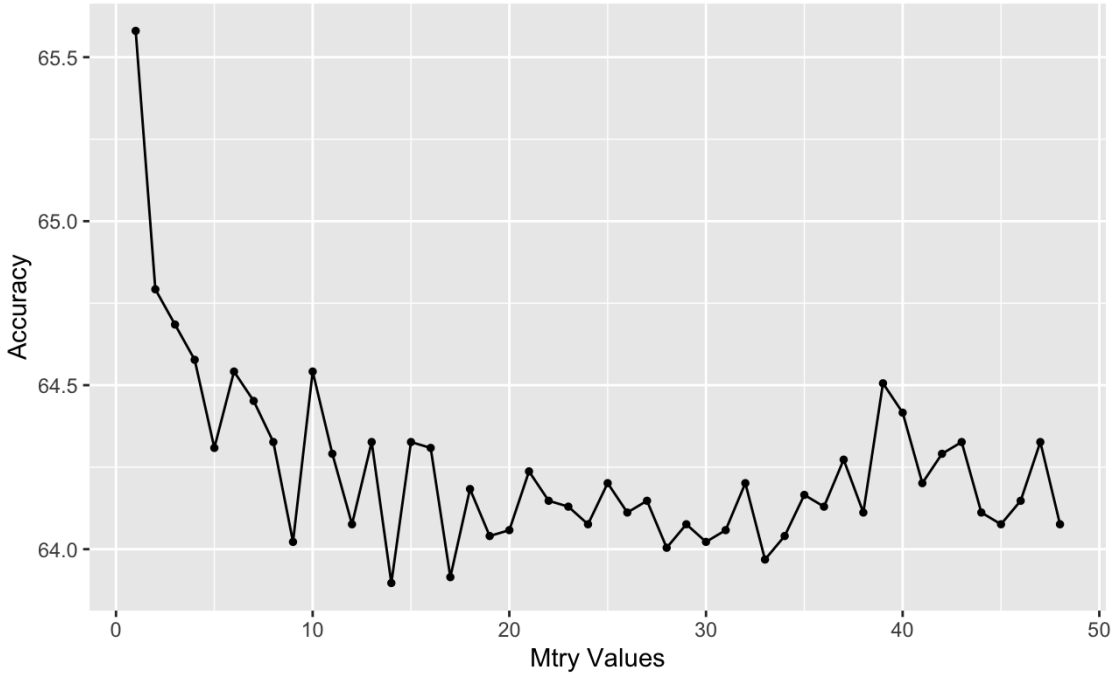

Figure 4.3.3.1: Plot for tuning mtry parameter for RF model

After the mtry was selected, the model was tuned for the number of trees.

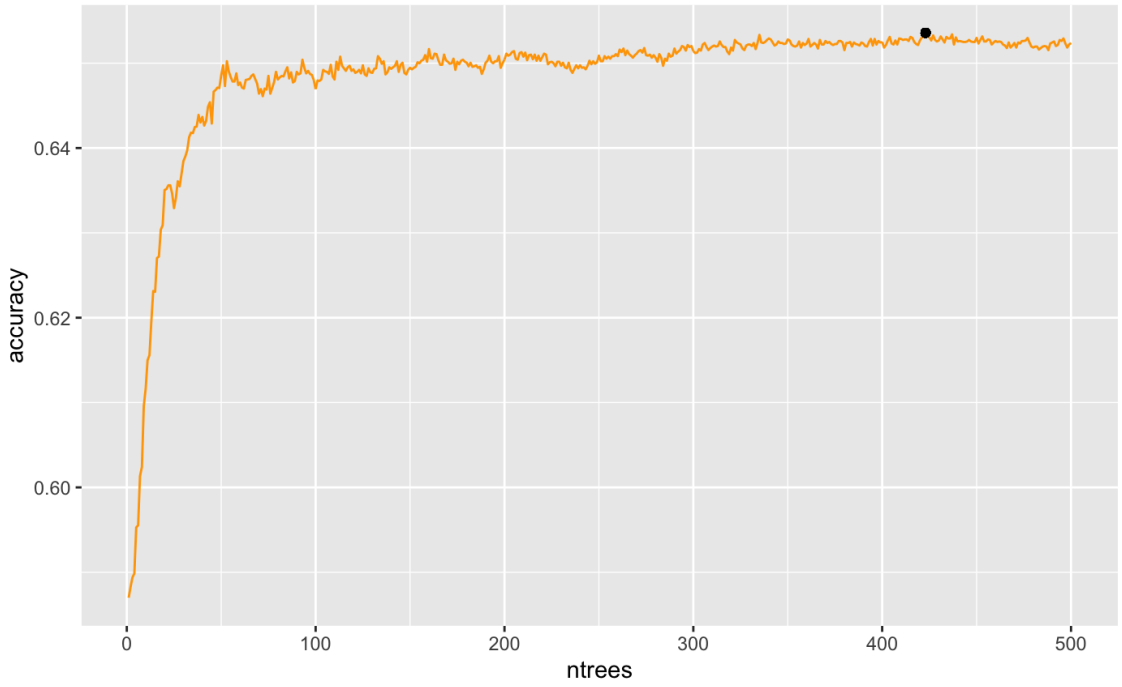

Figure 4.3.3.2: Accuracy of the RF model over the number of trees

The best number of trees was 405, and this combination of the best mtry and ntree hyperparameters yielded an accuracy of around 65-66%. This is an approximate accuracy since it varies slightly upon each iteration of the random forest model. It is important to note that this accuracy is still lower than that of the simple linear models.

4.3.3.1 Variable Importance

The random forest model can also provide insight into the importance of the variables. The figure below shows the importance of each variable scaled to set the most important feature(s) to 100. The most important feature was MOV/a for the home team, closely followed by Home W/L% and the MOV/a for the away team.

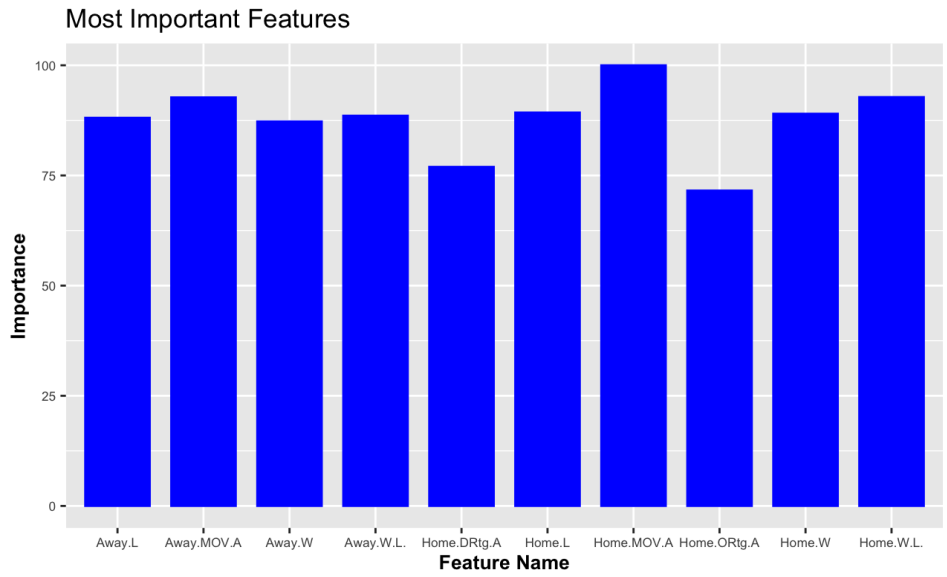

Figure 4.3.3.1.1: This figure shows the top 10 features from the Random Forest variable importance

The most important features are all pertaining to the "strength" of each team, and it is also worth noting that the home team makes up 6 out of the 10 most important stats, which is understandable since the home team wins the majority of the games in the NBA.

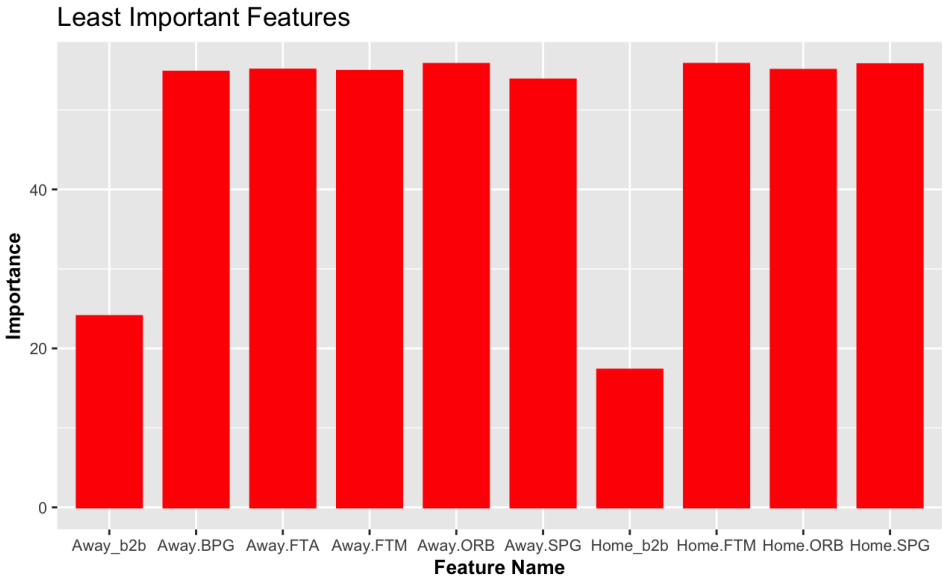

Figure 4.3.3.1.2: This figure shows the 10 least important features from the random forest analysis

The two least important features are clearly the indicators as to whether a game is a back to back. The rest of the features in the bottom 10 are also explainable as they do not mean much without additional context. You cannot explain whether a team plays good defense based on how many steals they get, and free throws made and free throws taken do not mean much without the other one to provide context as to the efficiency of a team in scoring from the freethrow line.

4.4 Neural Network

In addition to all of the models implemented in R there was also a neural network implemented using Pytorch. As mentioned earlier, in the literature the neural networks yielded the highest accuracy out of all of the models seen, so prior to testing the expectation was that this would be the same for the dataset used.

4.4.1 Architecture

The neural network consists of an input layer, three hidden layers and an output layer. The loss function is binary cross entropy loss, and the optimizer function is stochastic gradient descent. The Relu function is applied to layers one through four and the sigmoid function is applied to the output layer.

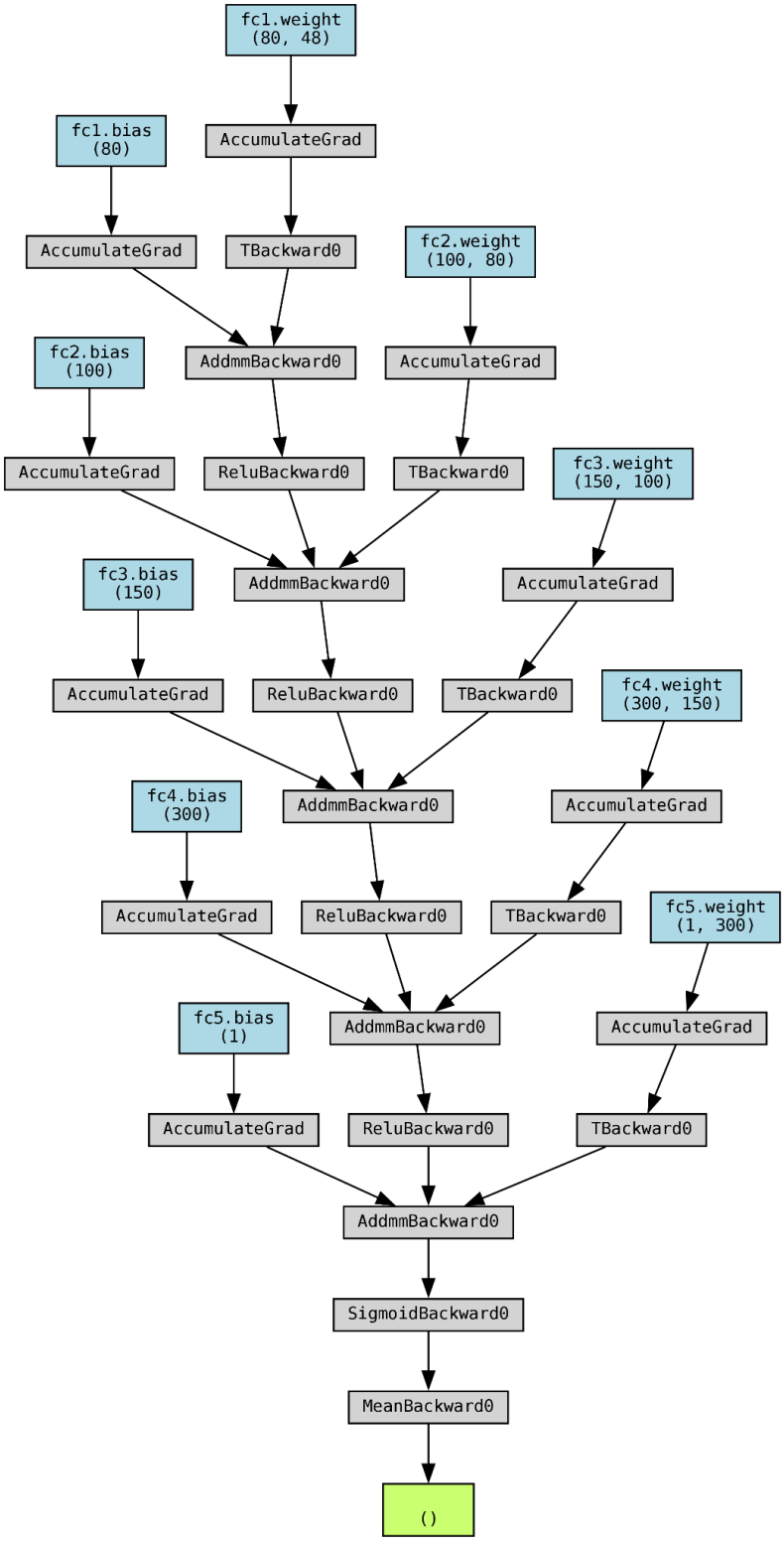

Figure 4.4.1.1: A visualization of the Neural Network architecture

4.4.2 Process

To test the model, the Neural Network took in both the training and testing datasets and used backpropagation to determine the proper weights for all of the layers. The network would then calculate the loss and the training and testing accuracy and repeat the process. This took place for 25,000 iterations.

4.4.3 Accuracy

The neural network did slightly outperform the other models as it nearly reached 70% accuracy on the test data in some of the iterations. After around 7,000 iterations, the test accuracy began to decrease due to overfitting on the training set.

Figure 4.4.3.1: Plot showing the change in training and testing accuracy over the iterations

Chapter 5 Balancing the Data

As mentioned earlier, just under 60% of the entries in the dataset feature the home team winning. As a result, the models were all much better at predicting the home team to win than predicting the away team. Each model typically boasted an accuracy of over 70% when predicting the home team to win but in the lower 60s when predicting the away team.

5.1 Balancing Methodology

The balancing was done undersampling with the ROSE package in R. The balanced dataset contained 18,179 rows and the home team won 9,168 times. The entire process was then repeated using the balanced dataset.

5.2 Impact of balancing

The balancing had no significant impact, as all of the balanced results were simply somewhat lower than the unbalanced results by a very small margin. This is understandable as the dataset was not terribly unbalanced in the first place (only a 60-40 split). The unbalanced results do perform better in terms of the split between home and away teams winning, but since the task is strictly about prediction this is not of use.

Model <chr>	True Positive Rate <dbl>	True Negative Rate <dbl>	Difference <dbl>
lin	0.68	0.69	-0.01
logit	0.81	0.52	0.29
probit	0.67	0.70	-0.03
bayes	0.70	0.66	0.04
rf	0.77	0.48	0.29
knn	0.84	0.44	0.40
nn	0.72	0.64	0.08

Figure 5.2.1: Home-Away splits for the unbalanced data

Model <chr>	True Positive Rate <dbl>	True Negative Rate <dbl>	Difference <dbl>
lin	0.66	0.70	0.04
logit	0.69	0.67	0.02
probit	0.68	0.68	0.00
bayes	0.68	0.69	0.01
rf	0.63	0.66	0.03
knn	0.71	0.60	0.11
nn	0.67	0.68	0.01

Figure 5.2.2: Home-Away splits for the balanced data

The balanced results do perform better in terms of the split between home and away teams winning, but this is not relevant when the task is entirely focused on the accuracy value.

Chapter 6 Results

The performance of the models did not differ greatly from the expectations. As mentioned earlier the unbalanced dataset consistently outperformed the balanced dataset, and the accuracy values hovered around a respectable 64-70% while not coming close to the highest seen value of 74%.

Model <chr>	Accuracy <dbl>
RF	65.44
Bayes	68.41
KNN	68.43
Linear Classifier	68.75
Probit	68.77
Logit	69.00
Neural Network	69.57

Figure 6.1: Accuracies for the of the unbalanced data

Model <chr>	Accuracy <dbl>
RF	63.92
KNN	67.85
Neural Network	67.90
Logit	67.92
Probit	67.97
Linear Classifier	68.07
Bayes	68.47

Figure 6.2: Accuracies for the balanced data

It is also of note that the accuracies of the linear models were extremely close to the other models, and oftentimes actually outperformed the linear models. The neural network was the best performing model, but the gap between the neural network and logit model represents a performance increase of under 1%.

Chapter 7 Future Improvements

There are several areas in which this work could be improved. The main issue is that the dataset fails to account for an individual team's performance in their home or away games. In the NBA there are teams that perform significantly better or worse in home or away games, and that is not captured in any way by the dataset. The addition of this information would likely further the data, especially in instances where a team that appears to be worse wins at home, since it is possible that the team plays well at home but cannot maintain this level of performance when traveling.

It is also possible that the dataset would benefit from normalized stats to adjust for the differences in play style year to year. Comparing the stats of a season from 2005 to 2015 using simply raw numbers likely is not sufficient for extracting the information that is required. Instead a better approach could be to use pace adjusted stats, which would have accounted for the change in speed of the game over time and could have resulted in a more apples to apples comparison between the different years.

Finally, it is possible that a more complex and intricate neural network architecture could have yielded better performance. Given the other results it is unlikely that there would have been massive gains, but an increase to 71-72% accuracy is well within the realm of possibilities.

Bibliography

Aryan, Omar, and Ali Reza Sharafat. “A Novel Approach to Predicting the Results of NBA Matches.” CS 229 Stanford. Accessed September 30, 2022. http://cs229.stanford.edu/proj2014/Omid%20Aryan,%20Ali%20Reza%20Sharafat,%20A%20Novel%20Approach%20to%20Predicting%20the%20Results%20of%20NBA%20Matches.pdf.

Boulier, Bryan L., and Herman O. Stekler. "Predicting the outcomes of National Football League games." *International Journal of forecasting* 19, no. 2 (2003): 257-270.

Loeffelholz, Bernard, Bednar, Earl and Bauer, Kenneth W. "Predicting NBA Games Using Neural Networks" *Journal of Quantitative Analysis in Sports* 5, no. 1 (2009). https://doi.org/10.2202/1559-0410.1156

Ji, Bigui, and Ji Li. "NBA All-Star lineup prediction based on neural networks." In *2013 International Conference on Information Science and Cloud Computing Companion*, pp. 864-869. IEEE, 2013.

Perricone, Jacob, Ian Shaw, and Weronika Święchowicz. “Predicting Results for Professional Basketball Using NBA API Data.” CS229 Machine Learning. Stanford

University. Accessed September 30, 2022. http://cs229.stanford.edu/proj2016/poster/PerriconeShawSwiechowicz-PredictingResultsforProfessionalBasketballUsingNBAAPIData-poster.pdf.