STATS 101C Final Project - NBA 2023-2024 Dataset

Brandon Erickson [redacted]

2024-12-12

Contents

1 Introduction

This dataset captures the current game history of the NBA from the 2023 to 2024 season. It captures game statistics for every game planned in the season, such as points, assists, rebounds, field goals, blocks, turnovers, essentially every available piece of data from competitive games. I will be using this data to train a test different models to complete the task: predict the outcome of the game (Win or Loss) based on historical data. While completely being able to predict a game outcome is impossible, otherwise you'd be able to become a millionaire through betting, we can use the data gathered and our analytical minds to draw reasonable conclusions about NBA team outcomes.
We will be using five different methods to build and analyze our model, this way we can see the best conclusion to draw and decision to make for our task. The five methods are: Logistic Regression, Support Vector Machine (SVM), Gradient Boosting, Quadratic Discriminant Analysis (QDA), and K-Nearest Neighbors (KNN). These give a variety of different methods and perspectives to best handle our data. If two of our models are similar, we will check the Area Under Curve (AUC) and K-fold Cross Validation to check for which model is ultimately better. Lets begin with data preprocessing to ensure accurate model performance.

2 Data Preprocessing

My data preprocessing began with simply cleaning the data of any missing values, to which there was just one. I then moved on to removing outliers from the data completely, there is more than enough data to where these matches do not mean much. Perhaps there was an injury during these games, or a player was dealing with personal issues during the match, these cannot be assessed with our data, and these most likely let to the outliers. Furthermore, the nature of our data does not rely so much on what these outliers are telling us, bad games happen, lucky games happen, and these were likely our outliers that would interfere with our data. Lastly, I created extra variables to help with predictions, weighted recent games, and considered stability in separate columns. This led to a lot of variables, but a lot of these ended up being removed.

2.1 Creating the Win/Loss (W/L) Binary Target Variable

The main goal of our data analysis and model building is to predict a win or loss. Thus, our target variable is *W/L*, and in order to work with this variable, it was converted to binary, with 0 representing a loss for the home team, and 1 representing a win for the home team. Before proceeding, we should evaluate our *W/L* variable to ensure an equal distribution, and a semi-equal observation for each class.

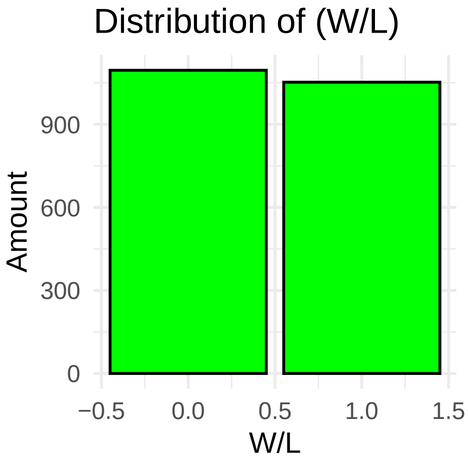

Figure 1: Balanced distribution of wins and losses

As we can see from the plot above, our binary target variable W/L is balanced well, both classes have equal observations. We will continue our data preprocessing smoothly now and eventually be able to build a model predicting this variable. We can train our model under the rightful assumption that wins and losses are balanced, remember: 0 is for a loss, 1 is for a win.

2.2 Creating HomeGame Variable

Playing a game at your home arena can be a huge advantage for the teams, the energy is on your side, and the fans are behind your back. However, if you are playing an away game, everything but your teammates are against you, and an unfamiliar court, and energetic fans rooting for you to feel. Home games can turn the tide of a matchup, and for this reason, I created another binary variable for if the team was at a home game, 1 if they were, 0 if they were not. This will be a useful predictor for our model as it provides additional, useful context to if the team had home court familiarity and support, thus leading to their victory, *Home_Game* is the name of our new variable, and will be used to build our future models.

2.3 Not Including Team, Match Up, or Game Date Variables

Although these variables are incredibly useful for distinction, they are mainly used to create other variables, and keep the data organized. However, they will not be included in our future models as you cannot build predictions off of these variables. The individual team data is important, and matchups will be addressed in a later variable, but these ultimately are not suited to be included in our model nor can they be converted to a different type of classification. Thus, these three variables will be used to build other predictors, but not used to actually predict our target variable.

2.4 Removing FGM and FGA

These two variables, *FGM* and *FGA* are rather redundant, as both can be expressed with *FG%*. For the purposes of our model, I will be removing these two variables to not only avoid overfitting but to also avoid using information that is already provided from another variable. FG% and FGM have a 0.81 correlation, which is extremely high, and while FG% AND FGA only have a -0.19, it is still acceptable to remove as there is still some correlation, and it would help avoid multicollinearality. Furthermore, FGA and FGM have a moderate correlation as well at 0.42. For this reason, we will be working solely with *FG%*.

Variables	**FGA**	**FGM**	**FG%**
FGA	1.00	0.42	-0.19
FGM	0.42	1.00	0.81
FG%	-0.19	0.81	1.00

Table 1: Correlation between FGA, FGM, and FG%

2.5 Removing 3PM and 3PA

For the exact same reasons above, I will be removing the **3PM** and **3PA** variables in exchange for just using **3P%**. The former two variables are both explained within 3P%. Although 3P% and 3PA have a lower correlation, 3PA has a high correlation with 3PM, and will be removed alongside it. Logically, 3P% is the variable that matters as it explains the overall accuracy, and thus it is the only variable out of these three we will be working with. The same reasoning is used for OREB and DREB variables, although they may provide niche insight, they are not worth overfitting our model with.

Variables	**3PA**	**3PM**	**3P%**
3PA	1.00	0.66	0.78
3PM	0.66	1.00	0.05
3P%	0.78	0.05	1.00

Table 2: Correlation between 3PA, 3PM, and 3P%

2.6 Creating Averages Based on Past Games (Recent Averages)

Using data for the past three games for each row, which inevitably led to excluding the first two games for the sake of our predictions, I captured the overall trend of how a team was playing during a certain point in the season. Rather than using complete long term data for predicting the next game, taking into account simply how a team has been performing on the last three games in a certain window helps to better predict and take into account circumstances the variables do not inherently do themselves, such as injuries, and this helps enhance stability. This was done for the PTS, FG%, 3P%, AST, and REB variables.

2.7 Adding Weights to More Recent Games (Weighted Averages)

Similar reasoning to the last, but slightly different, we will also consider weighted averages for the games as well, putting more emphasis on recent games. While the recent averages put weight equally on three-game windows, weights put unequal emphasis, making recent games more important while exponentially making less recent games less important. This also helps with stability, but more so for simply predicting the win-loss outcome. The weight function used is as follows:

$$w(t) = e^{-\lambda t}$$

where $\lambda = 0.1$, and t is the current number of games.

2.8 Final Feature Engineering & Variable Statuses

The below table consists of all of our original variables and their status after all of the data preprocessing. One important note is that I have not considered team matchup history, for one simple reason: it was deemed unnecessary after the weighting and averaging of recent games. I consider more recent performance to be a better predictor of a game's outcome rather than previous matchups. This was also not included to avoid overfitting. The below table highlights our data preprocessing and feature engineering steps.

Original Variable	Preprocessing Modifications	Final Result
Team	Utilized for variables, not model building	-
Match Up	Utilized for variables, not model building	-
Game Date	Utilized for variables, not model building	-
W/L	Converted to binary	W/L (1/0)
MIN	Discarded	-
PTS	Averaged and Weighed	RecPTS and WtPTS
FGM	Discarded	-
FGA	Discarded	-
FG%	Averaged and Weighed	RecFG and WtFG
3PM	Discarded	-
3PA	Discarded	-
3P%	Averaged and Weighed	Rec3P and Wt3P
FTM	Discarded	-
FTA	Discarded	-
FT%	Discarded	-
OREB	Discarded	-
DREB	Discarded	-
REB	Averaged and Weighed	RecREB and WtREB
AST	Averaged and Weighed	RecAST and WtAST
STL	Normalized	STL
BLK	Normalized	BLK
TOV	Normalized	TOV
PF	Discarded	-
+/-	Discarded	-
-	Created variable to represent home-court advantage	HomeGame

Table 3: Data Processing & Feature Modification Details

3 Experimental Setup and Model Decisions

Before we decide on the final model decision and experimental design, we must check for multicollinearality and ensure our variables are not overfitting for the sake of overfitting, it may be useful to remove certain variables that are explained by others. A correlation matrix was constructed but not included due to the amount of variables and interactions. The table below shows variables struggling with multicollinearality:

Variable 1	Variable 2	Correlation
WtPTS	WtFG	0.91
Wt3P	WtFG	0.87
WtPTS	Wt3P	0.84
RecPTS	RecFG	0.75

Table 4: Variables exhibiting high correlation

The above table highlights a few issues, mainly that our weighted field goal and recent field goal averages have high correlation with points and 3-pointers. We will be removing these variables in our model to avoid overfitting. Although weighted points and 3-pointers are highly correlated, we will keep both of these in our model as they both provide useful information despite being highly correlated, one focuses on overall performances, the other on efficiency. Once more, we will perform a 10-Fold Cross Validation test to see which variables should remain in our model.

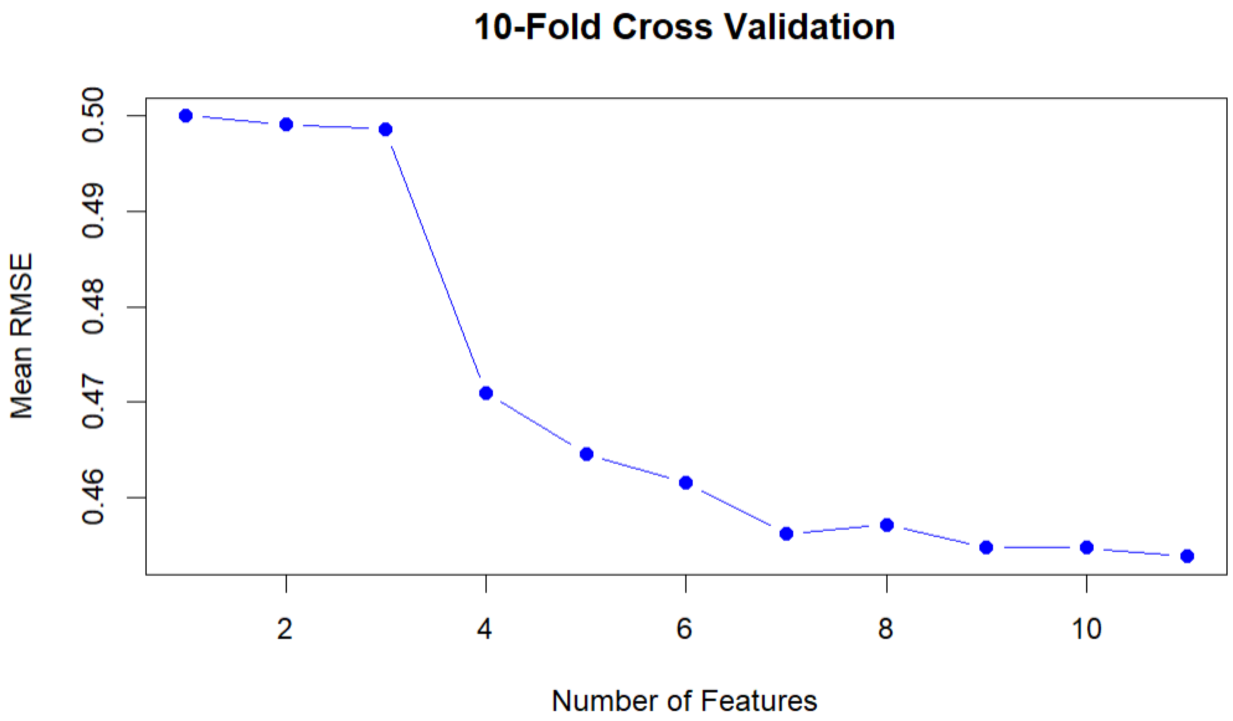

Figure 2: 10-Fold Cross Validation (via logistic regression) for Variable Selection

Based on the K-Fold Cross Validation plot, we will be using seven variables to maximize our model performance and minimize noise, overfitting, and over-complication. I understand this plot may seem confusing, but it was created with the general logistic regression method, following predictor sets and combinations to evaluate our variables. Each x-value is a set of predictors, with each x-value adding an additional predictor. Random forest may have been another way to validate this, but this method was chosen for its readability, interpretability, and direct observation of the mean RMSE. It highlighted where the model began to plateau, and although it reached its lowest point at nine variables, seven variables were selected to avoid overfitting/overcomplicating (which random forest struggles with) and the fact that our model RMSE went up with the eighth variable, which is not a good sign. Our seven variables are weighted 3-pointers, weighted assists, weighted rebounds, recent average points, recent average 3-pointers, recent average rebounds, and home-game advantage. Later model tests will determine if STL should be included in the model as well, or if home-game advantage should be excluded, but for now these seven variables will be the basis of our model to avoid overfitting.

Although we will be using both weighted and recent averages in our model, this helps account different factors in the NBA basketball season. Furthermore, removing redundant variables helps to improve the overall performance. The **Area Under Curve (AUC)** value is 0.72, meaning that this model is particularly good at distinguishing between wins and losses, and much better than just simply randomly guessing. With our model:

$$\boldsymbol{W/L \sim Wt3P + WtAST + WtREB + RecPTS + Rec3P + RecREB + HomeGame}$$

Finally, with our features and variables selected, we will now be conducting five different tests with a 70% training data, 30% testing data split. Our target variable is balanced with equal wins and losses and we will now proceed with the following five different tests:

1.) Logistic Regression: A simple, computationally easy test to see how our model performs under basic conditions.

2.) Quadratic Discriminant Analysis: A more complicated version of LDA, but helps identify nonlinear relationships, which our data is most likely full of. This is good for our different variances and it is flexible.

3.) Support Vector Machines (SVM): Handles complex relationships, and might help with the potential overfitting in our model.

4.) Gradient Boosting: Helps with numerical and binary data, and accounts for how structured our dataset is.

5.) K-Nearest Neighbors (KNN): Simple, easy way to test our data, but lacks complexity that our model might need.

The results and decisions will be decided in the next section.

4 Results, Analysis, and Decisions

The below table highlights what we have gathered from the five models, both their training and testing accuracy with a 70-30 training testing split.

Model	Training Accuracy	Testing Accuracy
Logistic Regression	67.27%	64.75%
Quadratic Discriminant Analysis (QDA)	66.53%	66.61%
Support Vector Machines (SVM)	69.13%	66.46%
Gradient Boosting	69.06%	65.68%
K-Nearest Neighbor (KNN)	73.59%	64.44%

Table 5: Model Results

Based on our information above, our best model suiting for predicting wins and losses is Quadratic Discriminant Analysis (QDA). Although our testing accuracy could be improved with the inclusion of other variables, or perhaps replacing the HomeGame variable with the STL variable, our model avoids overfitting and takes into account a unique factor rather than simply numerical values, so we will proceed with what we have. QDA was able to capture our complex model and captured the natural interactions between our multiple variables. The overall results for our QDA is shown below.

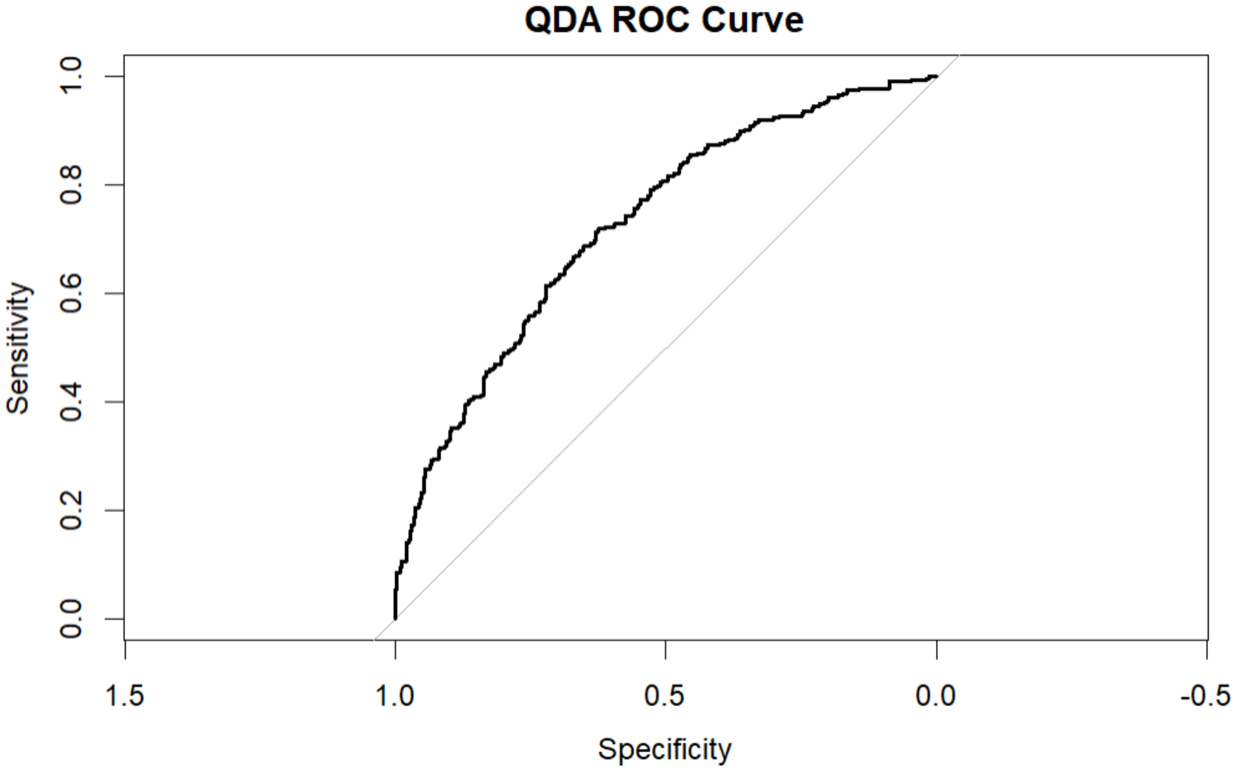

Figure 3: ROC Curve for QDA Model, AUC: 0.73

Alongside this information, we have a precision of 67% and a F_1 of 0.885. These are good results and highlight how our model is not just randomly guessing. There is a good balance between precision and recall and, although the model definitely has room for improvement, this is a satisfying result.

The model's AUC of 0.73 is not the most incredible result possible, 0.9 is usually the safest bet for a solid model, but 0.73 definitely captures that our model is sufficient for the most part. However, our precision of only 67% means it is only a bit better than 50%, which is just random guessing. This could use some fine

tuning in the future, perhaps by using the STL variables over the newly created HomeGame variable. With additional modifications, Gradient Boosting may be better for our model as it can capture the complexities of the dataset.A model testing accuracy of 66.61%, while it is far from bad, it is also far from terrible. This could most likely be improved in the future with better variable selection and elimination processes.
The decision of a QDA model was the most reliable, but others should be explored more in depth in the future, specifically SVM, as they are extremely close in testing accuracy, but also able to capture more complexity. The results are promising, predicting a win or loss is certainly possible, but there are modifications to be made.

5 Conclusions

The main objective of this project was to use historical data to predict wins and losses given the data from the NBA 2023-2024 season. This objective has been completed to moderate success, but there are some modifications that should be made in the future for better success and predictions. With a testing accuracy of 66.61%, an AUC of 0.73, and a F_1 of 0.885, our model holds up moderately well when it comes to making accurate predictions. However, a precision of 66% could use some work, as it only makes the right decision two-thirds of the time, which while better than random guessing, is still not incredible.

The Quadratic Discriminant Analysis model (QDA) was selected to be our final model, but it performed similarly to SVM and GBM. Further exploring either of these models with different variables or weights may be useful in the future to see what the better decision is. While I am proud of the results of the model, a good checkpoint would be to aim for around 70% accuracy, and while I did do this, improvement could be had with additional or different variables. Overall, QDA was the correct decision, with room for improvement in accuracy. The tradeoff for QDA was for accuracy for complexity, which may have been the right decision in terms of predicting, but the wrong decision in terms of interpretability compared to logistic regression.

One main issue could have been splitting the weights of numerical data and the influence of recent games. Perhaps doing just one or the other would have worked fine and left room for other variables to help with accuracy. However, it did overall help with more logical decisions, as more recent games compared to the current game play a huge role. One future modification to consider is team matchup history. This was ignored, as I considered individual game history to be more important, but it would be useful to look into.

Another potential issue is the HomeGame variable, as it overall was not significant in our model but still included as it provides an insight that normal numerical does not. This may have led to unnecessary overfitting or sacrificing of variables that could have performed better, but I consider the insight to be unique and invaluable. Variable selection could have been done better in general, a random forest selection could have been considered, but was left unused for a regression stepwise selection. Furthermore, our ROC Curve did not extend to the 0, 1 corners, indicating it could have been done a bit better.

Overall, our QDA model is sufficient, it is better than random guessing and uses insights properly from past data. However, in the future, the aforementioned adjustments should be made to see just how much our model prediction power can improve, or even other model predictions. Our model cannot predict the future, nor should any model be able to, but it has a good chance at predicting two out of three games, which is a powerful tool. The ability to predict a win or loss based on historical data was achieved, but there is still much work to be done in the future.