SpringBoard Data Science Career Track
Capstone Project 2

Predicting the Outcome of a Basketball Game

Merle Glick

Table of Contents

I. Introduction of Problem

Every year, many people attempt to predict the outcome of the NCAA basketball tournament. The goal for this project is to use machine learning to predict the outcomes of the tournament games based on data from the current season games as well as data from past years. I will consider every possible matchup of teams, and predict the winning team based on the data. I will consider many different variables, such as field goal rate, free throw rate, turnovers, and rebound rates to determine what factors affect the probability of a team winning.

Many basketball enthusiasts and sports analysts are interested in this data, and how to predict the winners of the tournament. Knowing which factors are strongly correlated to winning would be very useful. Even a basketball fan who has very little understanding of data analysis could make better predictions if they knew which factors have the greatest impact on a team's likelihood of winning.

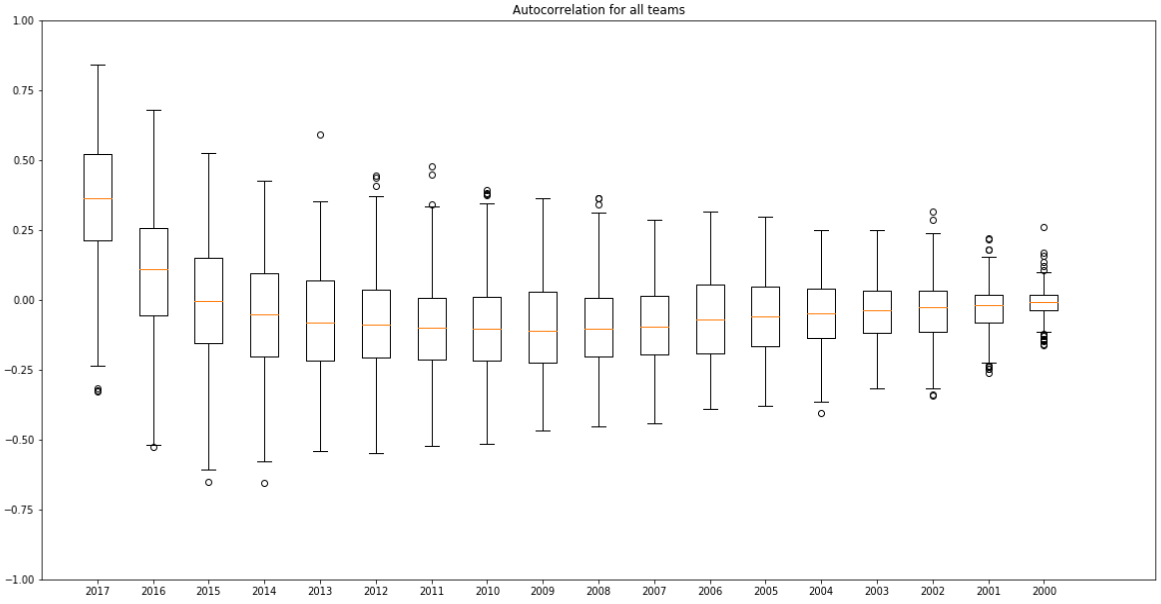
II. Data Acquisition and Cleaning

Most of the data was obtained from Kaggle, and is available in multiple csv files. This data contains team statistics going back as far as 1985. Some of the features that I would like to use are the field goal rates, free-throw rates, etc. The raw data does not contain these rates, but they can be easily calculated. For example, to find the field goal rate for a team, I can divide total field goals made by total field goals attempted. I calculated these rates and constructed a dataframe containing three columns: season, team, and field goal rate. I noticed that there were some null values in the field goal rate. I did some investigation, and discovered that these teams were not Division I teams in those years. I created other features as well, such as free throw rate, number of steals, number of blocks, etc.

I was able to find additional data from sports-reference.com. This website has the overall ratings for the teams, which was not available in the other datasets. The data from sports-reference.com was in html tables, so I had to do some web scraping to access the data in a format that I could use for analysis. I exported the data for each year in a csv file, extracted the columns that I needed, then merged it into a single dataframe. I had to do some cleaning to remove rows with null values, and also some rows that contained additional information that was not needed. I then calculated the correlation between the 2018 ratings and the ratings for each previous year, going back to 2008. There was still a fairly strong correlation between the 2018 ratings and the 2008 ratings, so I included the data for a few more years, going back to 2000. For the earlier years, especially prior to 2003, the correlation was not as strong.

I then looked at the autocorrelation of individual teams' rankings. I used the autocorrelation function from statsmodels to do this, and used matplotlib to construct boxplots displaying this data. First, I constructed a dataframe where each column was a year, starting at 2018 and going back to 2000, and each row was the overall rankings for a team. I used several for loops to access the data I needed, first calculating the autocorrelation values for each team,

then accessing all the values for 2018, all values for 2017, and so on. I constructed a boxplot for each year to get an overall picture of the autocorrelation. The plots are shown below. For the year 2018, all the values were equal to one, because it is the correlation of that year's ranking with itself, so I did not include the 2018 correlation values.

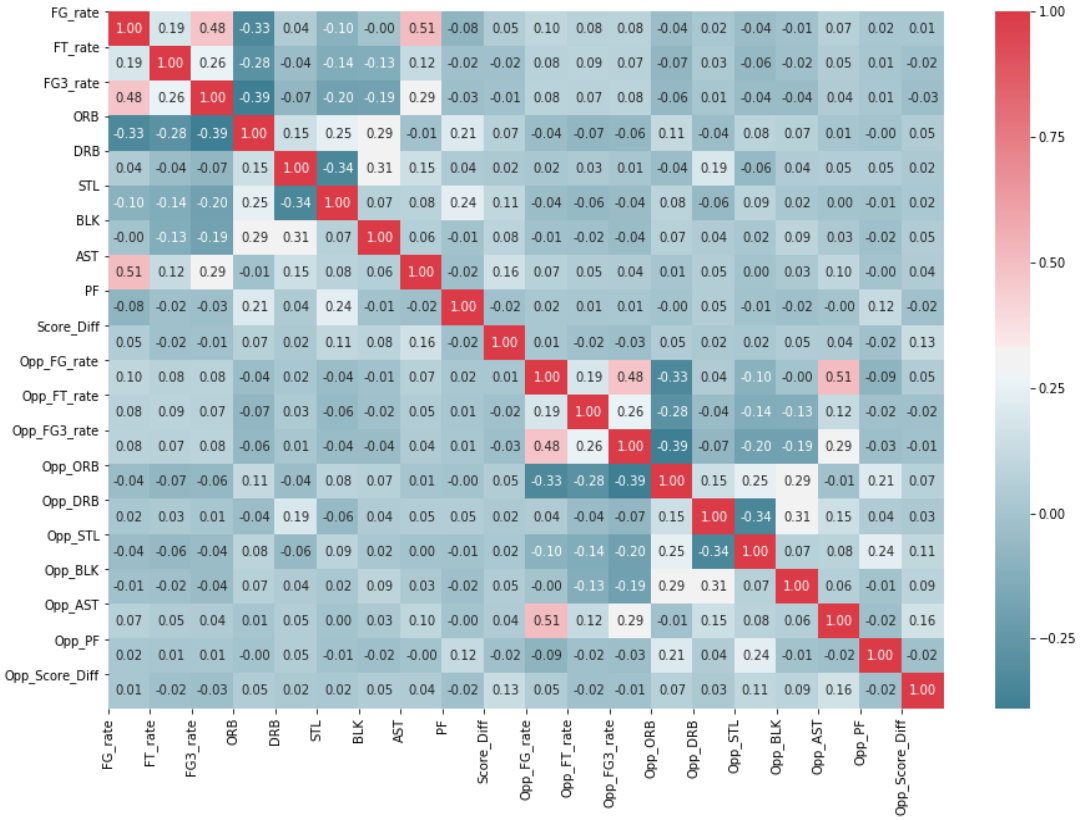


I constructed a dataframe with all the features that I had created, along with some of the variables from the original data. The created features included field goal rate, three-point rate, and free-throw rate, as well as average number of rebounds, steals, blocks, assists, and personal fouls. I also calculated the average difference in scores between the winner and the loser. All these values were calculated for the previous season. Other features included the season and the day number (since the beginning of the season). I then merged all this data, constructing a dataframe that contains a row for each game played by every team. I added a column of ones and zeros indicating whether the team won or lost. The goal was to build a model that would predict whether the team would win or lose, so this column was my target value.

The next step was to determine which of these features should be included in the model. I had 24 columns of features, including values for both teams for the previous season. More features might increase the accuracy of the model, but too many features can slow down the process, and can also decrease the power of the model. I did some exploratory analysis prior to fitting the model, then looked at the feature importance to determine whether some of the features could be dropped from the model without losing accuracy.

III. Exploratory Data Analysis

To conduct further analysis, I read in the file containing all the cleaned data, and selected all the quantitative features. I wanted to see whether any of these features were highly correlated with each other. The correlation heatmap, shown below, displays the correlations between all these features. More red color indicates a stronger correlation, and light blue indicates little or no correlation. This plot also displays the correlation coefficient for each pair of features.

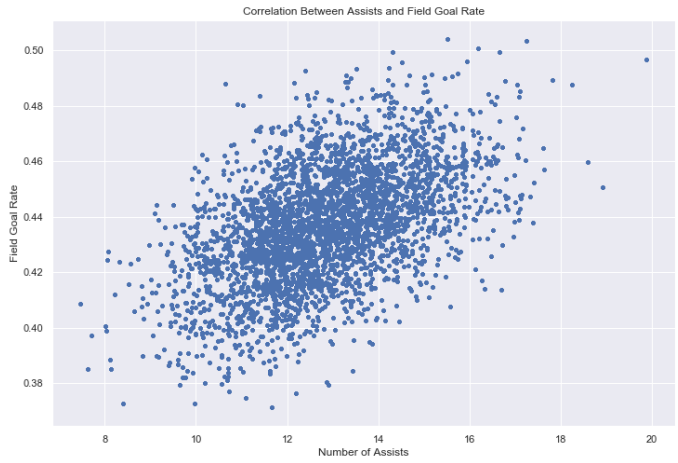


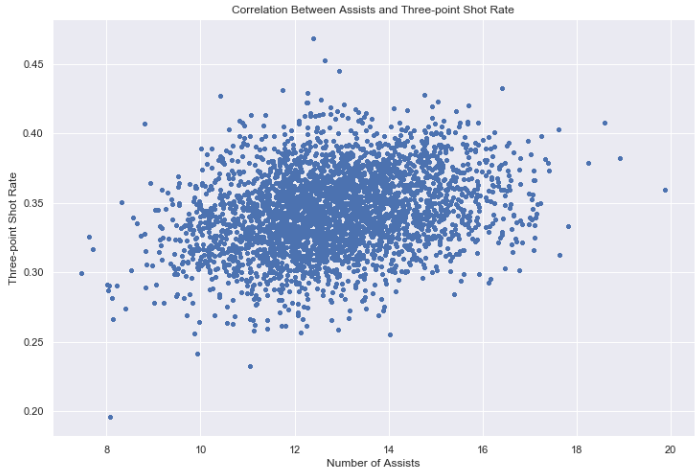
The plot indicates that most of the correlations are very weak. The only variables that appear to have a moderate positive correlation are field goal rate vs. three-point rate, and field goal rate vs. assists. We can see from the plot that these variables are correlated for both the winning team and the losing team.

After merging all the data and adding the features, I noticed that there were some null values in the dataframe. Upon further inspection, I noticed that there were several rows where the team ID was missing. I dropped these rows from the dataframe, since they would not be useful for prediction. There were also some rows where some of the feature columns had null values. For the quantitative variables, I filled these using the column mean. I now had a

dataframe with just over 100,000 rows and 25 columns. There were a few more features that I wanted to add, using external data that I had saved as csv files. First, I used data from the NCAA tournaments to add a column indicating whether or not each team had gone to the tournament in the previous season. Then I used the team rankings data (from sports-reference.com) to add a column with each team's ranking from the previous year. For these features, there were some missing values, and it did not make sense to impute the null values, so I decided to use a machine learning algorithm that can handle null values.

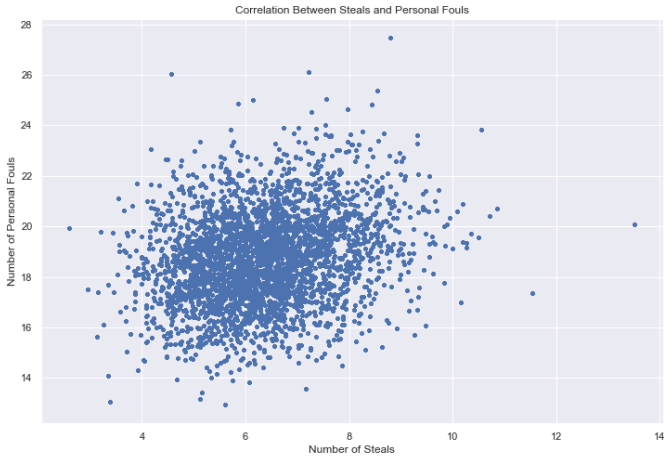
I wanted to look more at the correlations between some of the variables, so I constructed a few scatterplots. Two of these plots are shown below. There is a clear positive correlation between number of assists per game and the field goal rate. So in general, teams that have more assists per game tend to have a higher field goal rate. There is also a positive correlation between number of assists and three-point shot rate, but it is not as strong.



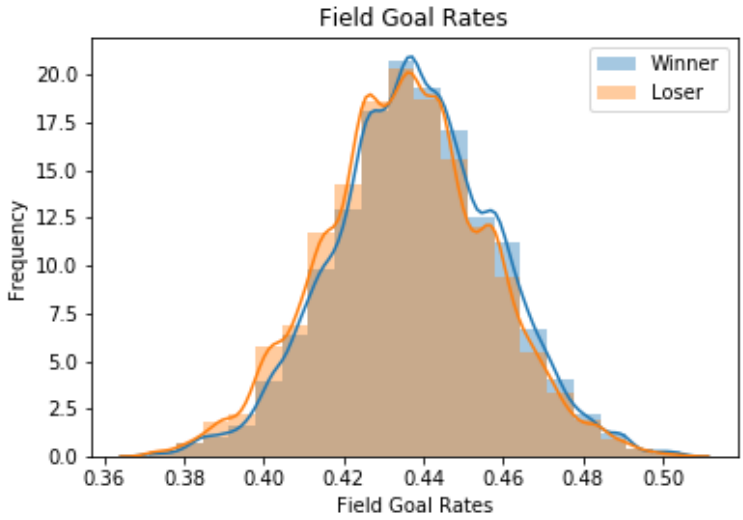


On the correlation heatmap, there also appeared to be a slight positive correlation between number of steals and number of personal fouls. The plot below shows this correlation. Although the previous plots did not surprise me, this correlation is not one that I would have

expected. However, it does make sense that there could be a correlation here. If a team is playing more aggressively, they will tend to steal the ball more, and they will also tend to have more personal fouls.



To determine which features might be good predictors of winning or losing, I constructed a few plots comparing the winner's data to the loser's data. The plot below shows the distribution of field goal rates for winning teams versus losing teams. We can see that the field goal rates are approximately normally distributed for both teams, but the winner's plot is shifted slightly to the right. This indicates that there is a difference between field goal rates for winning teams and field goal rates for losing teams, so this may be a good predictor to use in the model.



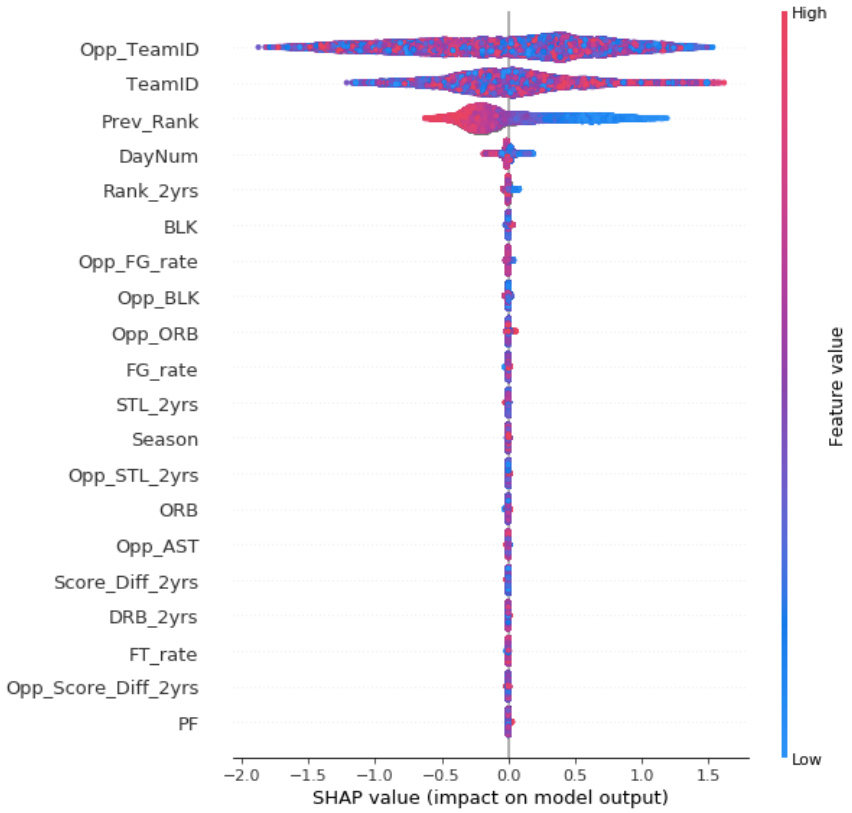
I then conducted hypothesis tests using all the features, to determine whether there is a difference between winners and losers. For most of the features, the p-value was very small, which indicates that there is a difference in that particular feature between the winning team and the losing team.

IV. Machine Learning

The next step was to fit a model to predict whether a team will win or lose. The dataset had some null values for the rankings, since some of this data was not available. Since it did not make sense to impute this missing data, I decided to use Lightgbm, a machine learning library that can handle null values. I used sklearn to split the data into training and test sets, and fit a binary classification model using the training data. I then used the model to predict values using the test set, and compared the predicted values to the observed values. The predicted values produced by the model are probabilities, so I converted them to binary values to compute the accuracy. The accuracy of this model was about 62%. I decided to add data from 2 years prior to try to improve the model. After adding these features, the accuracy improved to just over 63%.

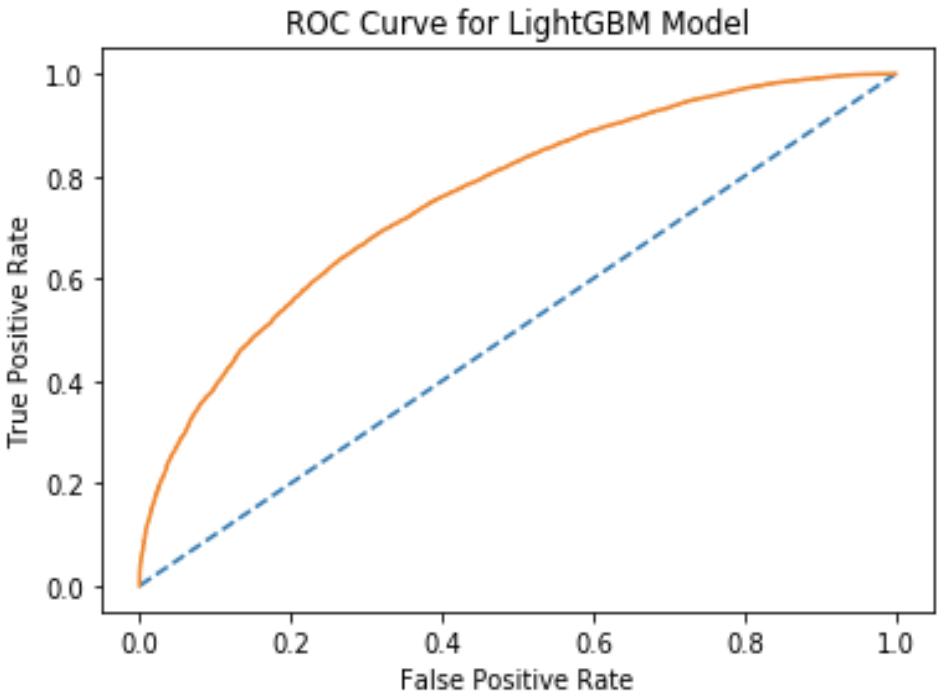
LightGBM has many hyperparameters that can be tuned to improve the model. I used a randomized search to test a range of values for each parameter, and chose the values that resulted in the best prediction. I also added the Season, team id, and the opposing team id to the model as categorical variables. I also adjusted the cutoff for converting the probabilities to binary values. Originally, I converted anything over 0.5 to a one, and anything below this to a zero. By testing a range of values, I found that increasing the cutoff value to 0.51 resulted in slightly better accuracy. After making these changes, the model predicted the outcomes with over 68% accuracy.

I used shap to create a plot (shown below) that shows the importance of the features in building the model.



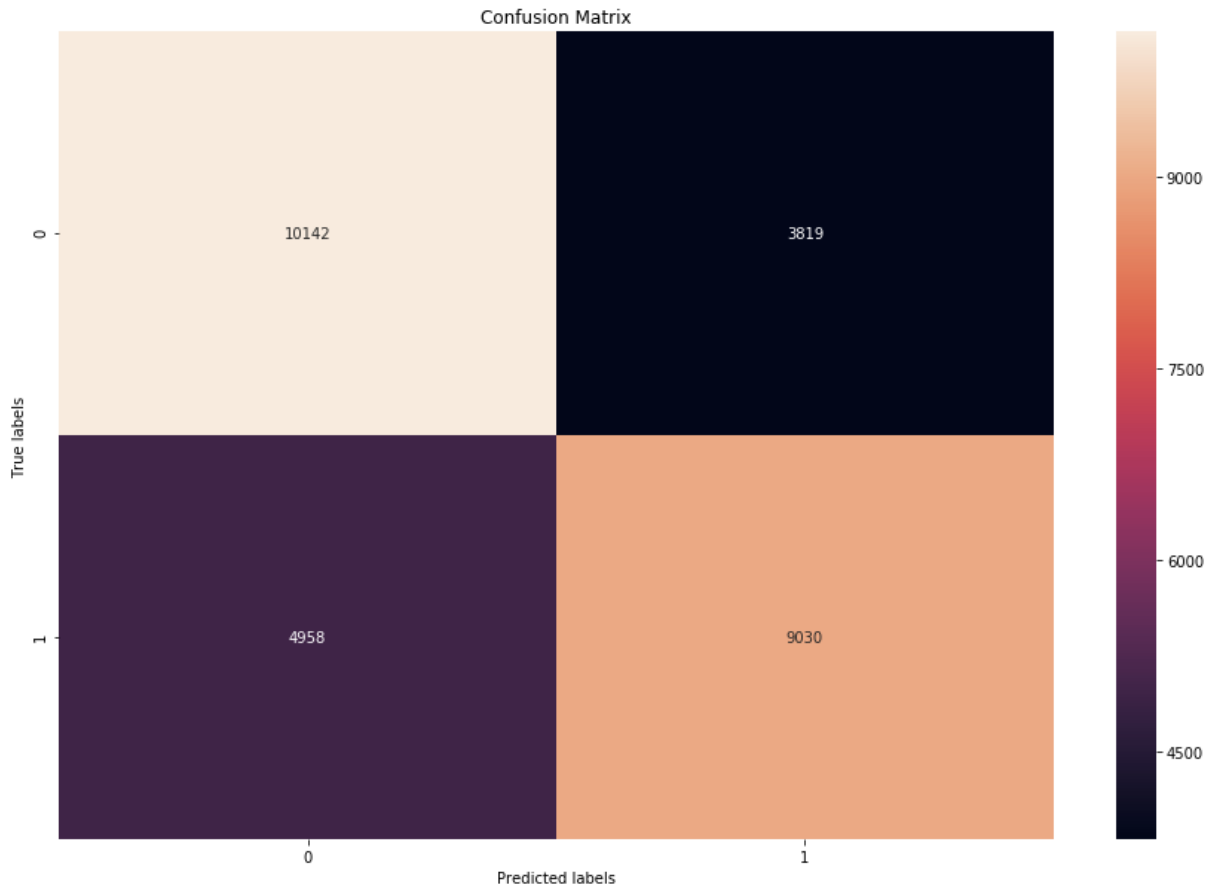
Based on this plot, I decided to eliminate some of the features. I dropped the 10 features with the lowest shap values, since they were having almost no influence on the model. So the features included in the final model were: team id, opposing team id, team ranking, day number (from beginning of season), number of blocks, field goal rate, offensive rebounds, and number of steals. For most of these features, I used the values from the 2 previous years. This final model resulted in an accuracy of 68.6%.

I also plotted the ROC curve, which shows how well this model works. It plots the false positive rate versus the true positive rate. The area under the curve (AUC) is a measure of how well the model predicts the outcomes. This area is over 75%, so the model performs 50% better than random guessing (the area under the blue line).




```
AUC:  0.7573180364960055
```

The confusion matrix (below) shows the number of values that were predicted correctly and those that were incorrectly predicted. This also gives us a measure of how well this model works. The values in the cells show how many ones and zeros were predicted correctly, and how many were predicted incorrectly.



V. Conclusion

In this project, we have discovered which features can be used to predict the outcome of a basketball game. By using data from the two previous years, we can predict whether a team will win or lose. By looking at the autocorrelation, and determining which features are correlated with each other, we have selected the best predictors and built a model that predicts the outcome of a game with over 68% accuracy. This information is useful for anyone wanting to predict the outcomes of basketball games, whether regular season games or tournament games. Future work for this project could include creating additional features, and possibly including individual player data, although this data may be more difficult to access. This model (or a similar model) could be used to predict wins for other sports as well, and possibly for other similar problems.