Predicitive NBA Over/Under Models and Analysis

Ethan Lewis

3/30/2021

Contents

List of Tables

Abstract

Sports betting is a billion dollar industry with a growing presence and future across the United States. In the report below, we detail the construction of two predictive models, an elastic net regression model and a random forest, aimed at estimating the total score of any given NBA game. Total score is an inherently useful piece of information in over/under sports betting (Will the total score of a game be higher or lower than the provided line?): an accurate model will ideally lead to long term betting success. The aforementioned models were constructed from a comprehensive (22 unique variables) NBA box score data frame which recorded information on 44,284 games spanning from 2012 to 2018. We have detailed the model building process, our findings, as well as the results of real world applications in the report below.

1.1 Data and Motivation

The following models and analysis have been trained and based on an NBA Enhanced Box Score data frame. The original data frame contains information on 78 variables (39 variables for both the home and away team) from the years 2012 to 2018 which amounts to 44,284 individual games.

A box score is a basketball game's stat sheet. A standard box score contains fairly self explanatory information (even to the non-basketball fan) such as Assists, Steals, 3-Pointer Percentage, Free Throw Percentage, etc. With 39 variables; however, the data frame we are working with is a bit more involved and contains a few relatively obscure game stats such as...

- Total Rebound Percentage (TREB%): "a measurement of the percentage of available offensive and defensive rebounds a team secures" (realgm.com)
- True Shooting Percentage (TS%): "a shooting percentage that factors in the value of three-point field goals and free throws in addition to conventional two-point field goals" (nba.com)
- Effective Field Goal Percentage (EFG%): "measures field goal percentage adjusting for made 3-point field goals being 1.5 times more valuable than made 2-point field" (nba.com)
- Floor Impact Counter per 40 minutes (FIC40): "a formula to encompass all aspects of the box score into a single statistic on a per-40 minute basis. The intent of the statistic is similar to other efficiency stats, but assists, shot creation and offensive rebounding are given greater importance" (realgm.com)
- Efficiency Differential (EDiff): "The difference between a team's Offensive Rating [points scored per 100 possessions] and Defensive Rating [points allowed per 100 possessions]" (realgm.com)
- Assist to Turnover Ration (AST/TO) and Steal to Turnover Ratio (STL/TO): "total number of assists or steals divided by total number of turnovers" (realgm.com)

A comprehensive list of all variables, their definitions, as well as links to their sources can be found in the Appendix on the final page of this report

The goal of the following report is to accurately predict the sum of the final score of any given NBA game. This is being done with the practical application of Over/Under sports betting (betting whether or not the total score of a game will exceed a given amount) in mind.

2.1 Data Cleaning

Prior to model building and analysis, the data frame needs to been cleaned and transformed. Firstly, we must create our response variable, "Total", by summing "teamPTS" and "opptPts" and subsequently removing them from the data frame.

Additionally, we removed several variables that report similar information to other variables in our data frame. Variables such as 2-Pointers, 3-Pointers, and Free Throws Attempted and Made were removed in

favor of their percentage values: “2P%”, “3P%”, and “FT%”. Similarly, all rebound related stats (“ORB”, “DRB”, “TRB”, “ORB%”, and “DRB%”) were removed in favor of Total Rebound Percentage, “TREB%”.

Other variables such as Assists, Turnovers, Steals, and Blocks remained in the data frame along with their respective percentage versions since they do not share a Success/Failure relationship similar to previously discussed variables.

Finally, Offensive Rating (“ORTG”) and Defensive Rating (“DRTG”) were removed in favor of “EDiff” which we previously mentioned is a function of the aforementioned variables. Assist Rate (“AR”) was also removed as it is a fairly obscure stat thus making it difficult to apply when making predictions.

Interestingly, in an advanced basketball stat sheet the percentage variables are on two different scales; some are expressed in terms of 100 while others are expressed in terms of 1. We would like to express all of our values in terms of 1 meaning we have divided the variables “TREB%”, “ASST%”, “TO%”, “STL%”, and “BLK%” by 100 to attain a consistent scale throughout the data frame.

3.1 Introduction

We will begin by training an elastic net regression model. Several of our variables are correlated/ calculated using functions containing other variables present in the data frame so we need to avoid using least squares linear regression (LSLR). Not to mention, performing LSLR and determining appropriate individual variable relationships and transformations would be extremely time consuming and likely far less accurate. With this in mind we are left to choose between ridge, lasso, and elastic net regression.

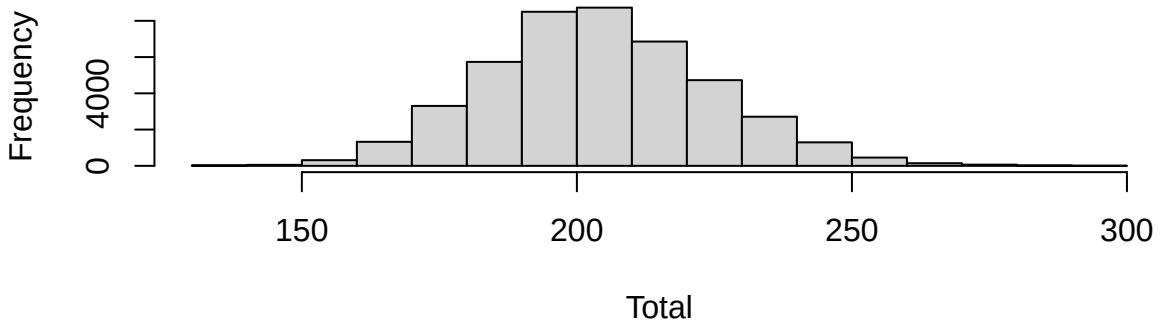
Even after cleaning, our data frame contains a relatively large number of variables and, as simply a shrinkage technique, ridge regression would not eliminate any in the model training process. Although ridge regression handles correlated variables well, we would like the opportunity to reduce the number of variables in our final model and, as a result, we will consider the selection techniques of lasso and elastic net regression.

Although we previously mentioned we are looking for the opportunity to reduce the number of variables involved in our model, lasso regression tends to take the selection process to the extreme and calculate the ‘slimmest’ possible model. Instead, we are going to pursue elastic net regression which essentially results in a healthy balance between ridge and lasso regression: a reduced model still with a respectable number of predictors.

3.2 EDA

Prior to training a model, let’s explore our response variable, “Total”, using Figures 3.2.1, 3.2.2, and 3.2.3 below.

Figure 3.2.1



Before exploring potential outliers within "Total" let's take a look at the variables distribution. This histogram in Figure 3.2.1 suggests "Total" has a unimodal, symmetric distribution centered around 210 points. We see some hints of potential outliers towards the tails of the plot.

Figure 3.2.2

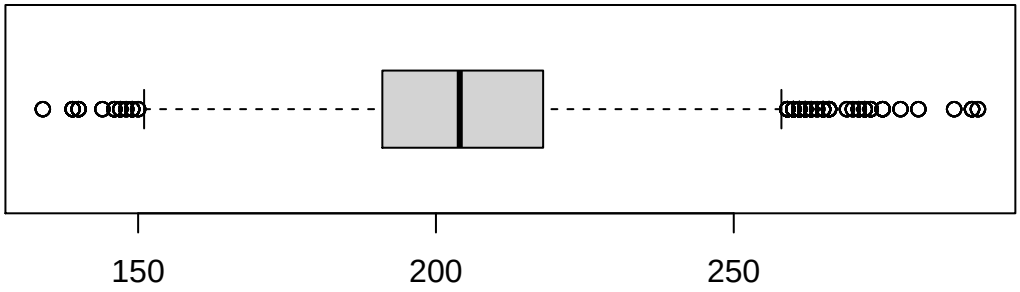
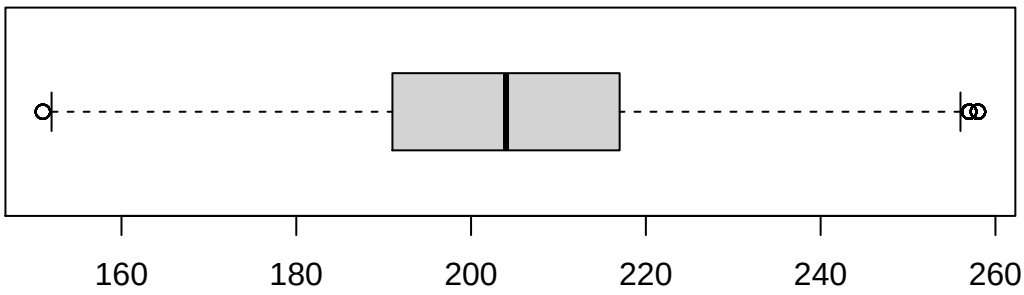


Figure 3.2.3



Similar to the previous histogram, Figure 3.2.2 suggests our response variable has a symmetric distribution. Additionally, the box plot indicates several outliers within "Total" lying beyond both the upper fence of 258 points and the lower fence of 151 points. Running an analysis of our data frame using these parameters, we know Figure 3.2.2 has indicated 390 games as outliers.

We are going to remove these points from the data frame for 2 distinct reasons. Firstly, 390 games compared to our sample of 44,284 games is minuscule and, secondly, the circumstances surrounding these games are likely fairly unique. While it is difficult to speak on the extremely low scoring games, the high scoring outliers are likely overtime games: a difficult circumstance to predict and one that almost always hits the over in over/under betting.

After removing the aforementioned outliers, the box plot of "Total" takes the shape of Figure 3.2.3. While there are still outliers present, this is simply a natural byproduct of refitting the box plot with new data. There are always going to be outliers in a data frame this size so we will stop the removal process here in order to avoid an endless cycle of outlier removal.

3.3 Elastic Net Test (100-Fold)

Finally, we are ready to train our model using elastic net regression. As we previously mentioned, elastic net is somewhat of a hybrid of ridge and lasso regression. Similar to LSLR, these techniques aim to minimize the model's residual sum of squares (RSS) and, ultimately, it's root mean squared error (RMSE). Unlike LSLR; however, in minimizing these values these techniques must also factor in a shrinkage penalty, a function of λ.

Ridge regression tests several models of the same size with varying λ values at an α of 0 while lasso regression tests several models of varying size with varying λ values at an α of 1. Each of these processes choose the model that results in the smallest RSS, MSE, and RMSE values.

As we know, elastic net regression is a mixture between the aforementioned model shrinkage and selection techniques. This means the elastic net training process tests several models of varying size with varying λ values but at a wide range of α values between 0 and 1.

Having trained our model using 100-fold elastic net regression (meaning our model was tested using 100-fold cross validation), we chose a model with an α value of 0.1, a λ value of 0.02105179, an R^2 value of 0.9884466, and an RMSE value of 2.096559. The model's coefficients (and included variables) can be found below in Table 3.3.1.

Table 1: Table 3.3.1

	Coefficients
(Intercept)	96.6801597
teamDayOff	0.0061121
teamAST	1.6811321
teamTO	2.9700758
teamSTL	0.2621638
teamBLK	1.0269315
teamPF	0.1729747
teamFG.	-70.0599099
team2P.	-2.2570595
team3P.	0.1553995
teamFT.	2.3729951
teamTREB.	0.0000000
teamASST.	-65.8391756
teamTS.	81.8990911
teamEFG.	1.6431357
teamTO.	-379.6781425
teamSTL.	-47.7602981
teamBLK.	-108.8213339
teamPPS	30.3089354
teamFIC40	0.1819997
teamEDiff	0.0000000
teamAST.TO	-1.2801182
teamSTL.TO	0.0144552
opptDayOff	0.0060849
opptAST	1.6858063
opptTO	2.9318962
opptSTL	0.2635145
opptBLK	1.0192799
opptPF	0.1736280
opptFG.	-70.6849693
oppt2P.	-2.3439401
oppt3P.	0.0886656

	Coefficients
opptFT.	2.3424623
opptTREB.	0.0000000
opptASST.	-66.0944715
opptTS.	81.5034658
opptEFG.	1.8138560
opptTO.	-375.1593826
opptSTL.	-48.3068438
opptBLK.	-108.3785053
opptPPS	30.3443353
opptFIC40	0.1846248
opptEDiff	0.0000000
opptAST.TO	-1.2836056
opptSTL.TO	0.0146726

3.4 Results

Having trained our model using elastic net regression, we can now asses it's predictive accuracy by revisiting a few previously mentioned metrics.

First, let's take a look at our model's R^2 value of 0.9884466. A value this close to 1 indicates our model has an extremely strong correlation with the response variable: it is doing an extremely good job of modeling a relationship with "Total". In fact, this value indicates 98.84466% of the variation seen in "Total" can be explained by our model.

Secondly, and more importantly, let's evaluate our RMSE value of 2.096559. Essentially, this value indicates that when testing our model using 100-fold cross validation, it's estimated prediction of total score was off by 2.096559 points on average. This is an extremely accurate prediction given our typical "Total" values tend to range from the mid-100s to the mid-200s.

With all of this in mind, our elastic net regression model did an excellent job in helping us attain our overall goal of total score prediction as it provided us with a strong correlation and high predictive accuracy.

Now, let's truly put our model to the test against a slate of NBA games. In Table 3.4.1 below, you will find a list of the 10 NBA games that occurred on May 7, 2021 along with their respective over/under line, predicted total score, prediction based bet decision, true total score, and prediction outcome.

Table 2: Table 3.4.1

Game	Over Under Line	Prediction	Bet Decision	Final	Prediction Result
Pelicans @ 76ers	224	211.44	Under	216 (Under)	Correct
Celtics @ Bulls	226	211.43	Under	220 (Under)	Correct
Magic @ Hornets	214	199.87	Under	234 (Over)	Incorrect
Timberwolves @ Heat	228	201.38	Under	233 (Over)	Incorrect
Rockets @ Bucks	233	211.17	Under	274 (Over)	Incorrect
Cavaliers @ Mavericks	217	199.67	Under	200 (Under)	Correct
Nuggets @ Jazz	215	216.14	Over	247 (Over)	Correct
Knicks @ Suns	217	205.56	Under	233 (Over)	Incorrect
Lakers @ Trailblazers	223	211.14	Under	207 (Under)	Correct
Spurs @ Kings	224	209.22	Under	217 (Under)	Correct

Over/Under Lines provided by Caesars Casino & Sportsbook (https://www.caesarscasino.com/sports/)

At a glance, in an extremely small sample of size 10 our elastic net regression model was able to produce a 60%

success rate. Interestingly, on several occasions our model was able to outperform the official Over/Under Lines (smaller magnitude residual). While we were perhaps hoping for a higher success rate, our model was still able to "beat the odds" within a small sample.

Predictions were made using season average values for the relevant variables. Perhaps accuracy would be improved if form based values (i.e. past 5 games averages) were used instead. Unfortunately, these values are much more difficult and time consuming to attain.

4.1 Introduction

In addition to an elastic net regression model, we will also be constructing a random forest aimed at predicting the total score of any given NBA game. Although we had the choice of constructing a bagged regression forest working with a numeric variable such as "Total", we elected to pursue a random forest in order to cut down on time since it tests significantly fewer variables at each split during the bootstrapping process.

The purpose of the random forest is more or less identical to that of the elastic net regression model. Both are geared towards prediction; we would like to see if one is more predictively accurate than the other, both in it's metrics as well as it's practical applications.

4.2 EDA

The exploratory data analysis for our random forest is identical to the process performed in Section 3.2. Revisit Figures 3.2.1, 3.2.2, and 3.2.3 for a reminder on the properties of our response variable, "Total".

4.3 Method

Now we are ready to train our random forest using 100 bootstrap samples. This means our forest is comprised of 100 trees each built on different variations of our original data frame. Within each of these trees, a random selection of 7 ($\sim\sqrt{44}$) of our 44 explanatory variables were considered at each split and the variable and splitting value that maximized the reduction in the tree's RSS value was chosen. This process continued throughout each tree until a stopping rule was reached or the RSS value stopped decreasing. When making predictions using a forest, a prediction is obtained within each tree and then averaged.

With this process in mind, training our random forest resulted in an RMSE value of 2.469506 and an R^2 value of 0.9935.

4.4 Results

The results of our random forest are fairly similar to those of our elastic net regression model. Our random forest has a slightly higher R^2 value of 0.9935 indicating an extremely strong correlation with our response variable and that 99.35% of the variation in "Total" can be explain by our model. The random forest also has a comparable RMSE value of 2.469506 indicating that when tested, our model's estimated prediction of total score was off by 2.469506 points on average.

We believe our RMSE value could be improved had we used a larger bootstrap sample (i.e. 1000 instead of 100). Unfortunately, performing a sample of this size was extremely computationally expensive and resulted in several computer crashes. Regardless, our model still appears to be very accurate.

All of this information combined indicates, once again, our model did an excellent job in helping us attain our overall goal of total score prediction as it provided us with a strong correlation and high predictive accuracy.

Similar to our elastic net regression model, let's put our random forest to the test agains the slate of May 7, 2021 NBA games using Table 4.4.1 below.

Table 3: Table 4.4.1

Game	Over Under Line	Prediction	Bet Decision	Final	Prediction Result
Pelicans @ 76ers	224	224.18	No Bet (Same)	216 (Under)	Correct
Celtics @ Bulls	226	223.87	Under	220 (Under)	Correct
Magic @ Hornets	214	210.69	Under	234 (Over)	Incorrect
Timberwolves @ Heat	228	216.82	Under	233 (Over)	Incorrect
Rockets @ Bucks	233	223.34	Under	274 (Over)	Incorrect
Cavaliers @ Mavericks	217	211.94	Under	200 (Under)	Correct
Nuggets @ Jazz	215	227.28	Over	247 (Over)	Correct
Knicks @ Suns	217	223.56	Over	233 (Over)	Correct
Lakers @ Trailblazers	223	220.98	Under	207 (Under)	Correct
Spurs @ Kings	224	218.29	Under	217 (Under)	Correct

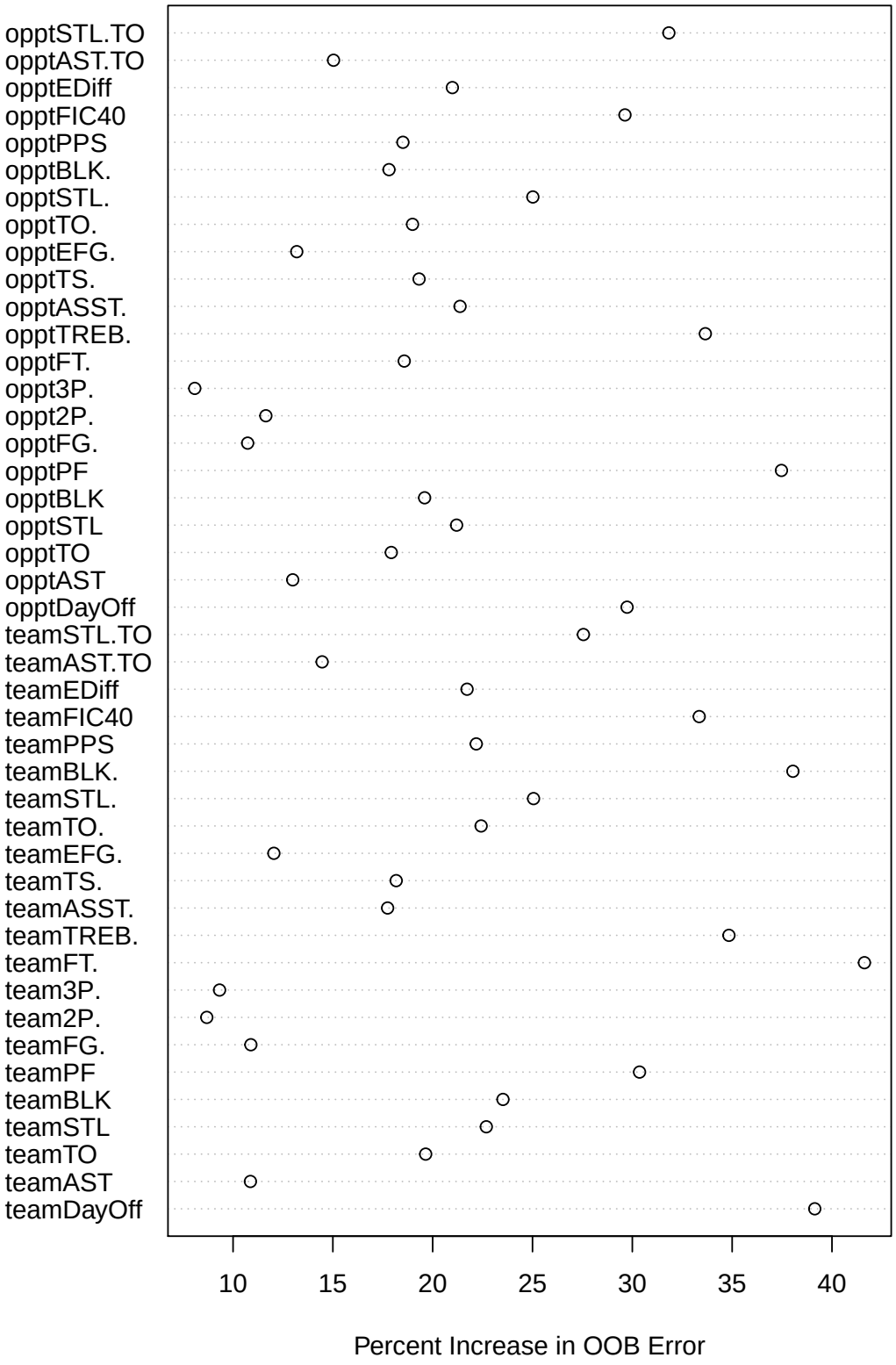
Over/Under Lines provided by Caesars Casino & Sportsbook (https://www.caesarscasino.com/sports/)

At a glance, in an extremely small sample of size 10 our elastic net regression model was able to produce a 70% success rate. More often than not, our model was also able to outperform the official Over/Under Lines (smaller magnitude residual). Despite a slightly larger RMSE value, our random forest was able to outperform our elastic net regression model by 10%.

Predictions were made using season average values for the relevant variables. Perhaps accuracy would be improved if form based values (i.e. past 5 games averages) were used instead. Unfortunately, these values are much more difficult and time consuming to attain.

Finally, a unique feature of our random forest allows us to visualize which variables were most important in predicting "Total": this can be done using Figure 4.4.1 on the next page. While individual variable importance can easily be explored by studying the plot, a more subtle trend emerges by taking a deeper look. A disproportionate number of home team variables were determined to be more important than away team variables; perhaps this is a visualization of home team advantage although that is difficult to attribute to total score which requires output from both the home and away teams.

Figure 4.4.1


opptSTL.TO
opptAST.TO
opptEDiff
opptFIC40
opptPPS
opptBLK.
opptSTL.
opptTO.
opptEFG.
opptTS.
opptASST.
opptTREB.
opptFT.
oppt3P.
oppt2P.
opptFG.
opptPF
opptBLK
opptSTL
opptTO
opptAST
opptDayOff
teamSTL.TO
teamAST.TO
teamEDiff
teamFIC40
teamPPS
teamBLK.
teamSTL.
teamTO.
teamEFG.
teamTS.
teamASST.
teamTREB.
teamFT.
team3P.
team2P.
teamFG.
teamPF
teamBLK
teamSTL
teamTO
teamAST
teamDayOff
10
15
20
25
30
35
40
Percent Increase in OOB Error

Conclusion

Both our elastic net regression model and random forest were able to meet our overall goal of accurately predicting the total score of any given NBA game. The models met these expectations both in terms of their RMSE values as well as their real world applications against a recent slate of NBA games.

Unfortunately; however, only one model can be used when attempting to predict "Total". With all findings in the above report considered, we recommend using the random forest for predicting "Total". Although the forest had a larger RMSE value there is room for potential improvement in this metric by increasing the bootstrap sample size with a more powerful computer. When applying the model to our sample of 10 NBA games it seemed to produce larger values more in line with the official over/under lines when compared to our elastic net regression model in addition to blatantly realizing more success.

Ultimately, we recommend the random forest based on it's potential for improvement. While only operating on 100 bootstrap samples, the forest was able to produce extremely accurate results: the model can undoubtedly be improved and fine tuned for further success.

Data Frame Works Cited

NBA Enhanced Box Score and Standings (2012-2018). Version 27. Retrieved February 25, 2021 from https://www.kaggle.com/pablote/nba-enhanced-stats?select=2012-18_officialBoxScore.csv

Appendix

- Days Off (DayOff): Number of days since last game
- Assist (AST): “The number of assists – passes that lead directly to a made basket – by a team” (nba.com)
- Turn Over (TO): “A turnover occurs when the team on offense loses the ball to the defense” (nba.com)
- Steal (STL): " Number of times a defensive team takes the ball from a player on offense, causing a turnover" (nba.com)
- Block (BLK): “A block occurs when an offensive team attempts a shot, and the defense tips the ball, blocking their chance to score” (nba.com)
- Personal Foul (PF): “The number of personal fouls a team committed” (nba.com)
- Field Goal Percentage (FG%): “The percentage of field goal attempts that a team makes” (nba.com)
- 2 Point Field Goal Percentage (2P%): “The percentage of 2 point field goal attempts of a specified criteria that a team makes” (nba.com)
- 3 Point Field Goal Percentage (3P%): “The percentage of 3 point field goal attempts that a team makes” (nba.com)
- Free Throw Percentage (FT%): " The percentage of free throw attempts that a team has made" (nba.com)
- Total Rebound Percentage (TREB%): “A measurement of the percentage of available offensive and defensive rebounds a team secures” (realgm.com)
- Assist Percentage (AST%): percent of offensive possessions ending in an assisted score
- Trues Shooting Percentage (TS%:): “a shooting percentage that factors in the value of three-point field goals and free throws in addition to conventional two-point field goals” (nba.com)
- Effective Field Goal Percentage (EFG%): “measures field goal percentage adjusting for made 3-point field goals being 1.5 times more valuable than made 2-point field” (nba.com)
- Turn Over Percentage (TO%): percent of offensive possessions ending in a turn over
- Steal Percentage (STL%): percent of defensive possessions ending in a steal
- Block Percentage (BLK%): percent of defensive possessions resulting in a block
- Points per Shot (PPS): “points scored per field goal attempt” (realgm.com)
- Floor Impact Counter per 40 minutes (FIC40): “a formula to encompass all aspects of the box score into a single statistic on a per-40 minute basis. The intent of the statistic is similar to other efficiency stats, but assists, shot creation and offensive rebounding are given greater importance” (realgm.com)
- Efficiency Differential (EDiff): “The difference between a team’s Offensive Rating [points scored per 100 possessions] and Defensive Rating [points allowed per 100 possessions]” (realgm.com)
- Assist to Turnover Ration (AST/TO): “total number of assists divided by total number of turnovers” (realgm.com)
- Steal to Turnover Ratio (STL/TO): “total number of steals divided by total number of turnovers” (realgm.com)