Senior Thesis in Mathematics

Random Forests and Beyond

Author:
Zachary Senator

Advisor:
Dr. Jo Hardin

Submitted to Pomona College in Partial Fulfillment
of the Degree of Bachelor of Arts

April 29, 2020

Abstract

This paper seeks to expand on previous literature in using Random Forests to predict sports games. Random forests were first introduced by Leo Breiman in 2001. [2]I use a Random Forest model, to simulate the 2017-2018 NBA playoffs. The variables included in this model are taken from other research papers in this field. I seek two combine the most powerful predictors in other papers' models, which include bookmaker odds and win percentage, in order to create a powerful model. I use the probabilities given from the random forest and use Bernoulli trials to simulate the 2017-2018 NBA playoffs. Overall the Random Forest did a good job of predicting winners in series which went to seven games, however it struggled with some of the series in which there was a clear favorite.

Contents

Chapter 1

Introduction

1.1 Goals of Paper

The goal of this thesis is to correctly simulate and predict the winners of NBA playoff games. Using a complex data set and advanced machine learning techniques, I hope to correctly classify the winners and losers in the 2017-2018 NBA Playoffs.

1.2 Literature Review

The framework of my thesis is largely based off of two other studies. The first uses a hybrid random forest model to predict outcome of countries during the 2018 world cup. This paper helped me with the idea of using Random Forest probabilities to conduct simulations.[6] The second paper uses a Random Forest along with other machine learning models to predict NBA playoff games. [8] Similar to this paper, I include many of the same variables in my feature space, and predict the same target variable–the winner of the NBA Playoffs. Both of these papers are discussed in further detail later in this paper. Other papers on Random Forests discuss the win probability of football teams given the current situation of the game. The results are novel, because coaches decisions will be effected by an advanced machine learning technique.[9] As a sports fan and mathematician, I am motivated to create my own model using machine learning techniques that correctly predicts winners of sports games.

Chapter 2

Trees

2.1 Regression Trees

In order to better understand a regression tree we will begin with an example. Consider a model that attempts to predict a baseball players salary based on the number of years he has been in the league and the number of hits he recorded the previous season.

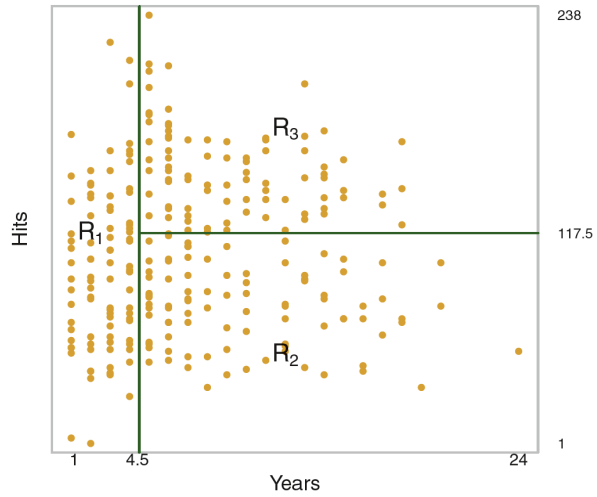


Figure 2.1: A simple model of a Players salary based on years played and hits. [7]

Figure 2.1 shows a regression tree fit to the baseball data set containing

years in the major leagues, and number of hits, amongst other variables.[7] A split is represented by the black lines on the graph. Splits are referred to as internal nodes. In this regression tree, the splits occur at years ≥ 4.5 and at hits $= 117.5$. R_1, R_2, and R_3 represent terminal nodes. Consider a player in the data set referred to in Figure 2.1, Player i. If Player i falls into R_1, the mean of all salaries in R_1 will be the predicted salary value for player i. The same holds if a player falls into region R_2 or R_3. For example, the predicted response for a player with less than 4.5 years of experience will be $\$165,174$. Further, given that a player is less experienced, the number of hits in his previous year has no effect on his predicted salary. On the other hand, among players who are more experienced, the number of hits a player had in the previous year does play a factor in salary. This is shown by a split at 117.5 hits. [7] This also make sense given the structure of rookie baseball contracts because players are signed for $3-5$ years before being resigned.

2.2 Building a regression Tree

There are two steps for building a regression tree which are outlined as follows:

1. Divide the predictor space for feasible values of $X_i, X_1, X_2, ..., X_p$ into j distinct non overlapping regions $R_1, R_2, ..., R_j$. [7]

2. For every observation that falls into the region R_j, we make the same prediction (As illustrated in the example above). [7]

To further illustrate the second point, let's consider another example. Consider 2 regions R_1 and R_2 where the mean response of the training data is as follows:

- $\hat{Y}_{R1} = 100$
- $\hat{Y}_{R2} = 200$

If player $i \in R_1$, then the model will predict 100 for player i. If player $j \in R_2$, then the model will predict 200 for player j. [7]

Let's take a look at step 1, which is the most complex and important part of a decision tree. In order to construct the boxes or regions seen in the example above, regression trees follow a decision rule. The tree finds

boxes $R_1, ..., R_J$, which minimize the residual sum of squares (RSS). The RSS $= \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{Rj})^2$, where $\hat{y}_{Rj}$ is the mean response for the training observations within the j^{th} box. Essentially, the tree considers possible values for splits, and at each feasible value of X_i the model minimizes the RSS, and it chooses to split at the value where the RSS is minimized. [7] Moreover in the example above, consider the first split the tree attempts to make. It cycles over all year values, for example lets say it starts at 1.5 years, it calculates the RSS, looks to see if it's minimized, and then keeps doing this for all possible values until the RSS is minimized. In this case it is minimized at 4.5 years.

2.3 Another Decision Tree Example

To better understand decision trees, consider a rudimentary example, taken from an article written by Will Koehrsen.[11]. Consider the following flow chart in Figure 2.2. Figure 2.2 represents a flow chart, and a decision tree from a human perspective. Our goal is to predict the maximum temperature in Seattle tomorrow. Today is December 27^{th}. We start by forming an initial reasonable range given our domain knowledge, which for this problem might be between 30-70 degrees. Next, we need to ask a series of questions in order to narrow the range in the hope of arriving at a final estimate. The first question we might ask is: what season are we in? The answer to this question is Winter, and thus we narrow our range to between $30-50$ degrees. Next we may be interested the historical average maximum temperature tomorrow, which we see is 46 degrees. Therefore, we refine our range to be between $40-50$ degrees. Lastly, we ask what the maximum temperature was today, and see that it was 43 degrees, and thus we make a final prediction of 44 degrees. To sum up this thought process, we start with an initial guess based on our knowledge and refine our estimate as we gain more information. Eventually we stop gathering data and make a decision, in this case is the max temperature in Seattle tomorrow. Our natural approach to the problem is called a question and answer flow chart. This is not quite a decision tree, because as humans we take some shortcuts that make sense to us, but are not intuitive to a machine. [11] There is one main difference between our illustrated decision process and a real decision tree. We have neglected to list the alternative branches. In other words, the decision considers all possible alternatives to our questions. For example, if the season had been

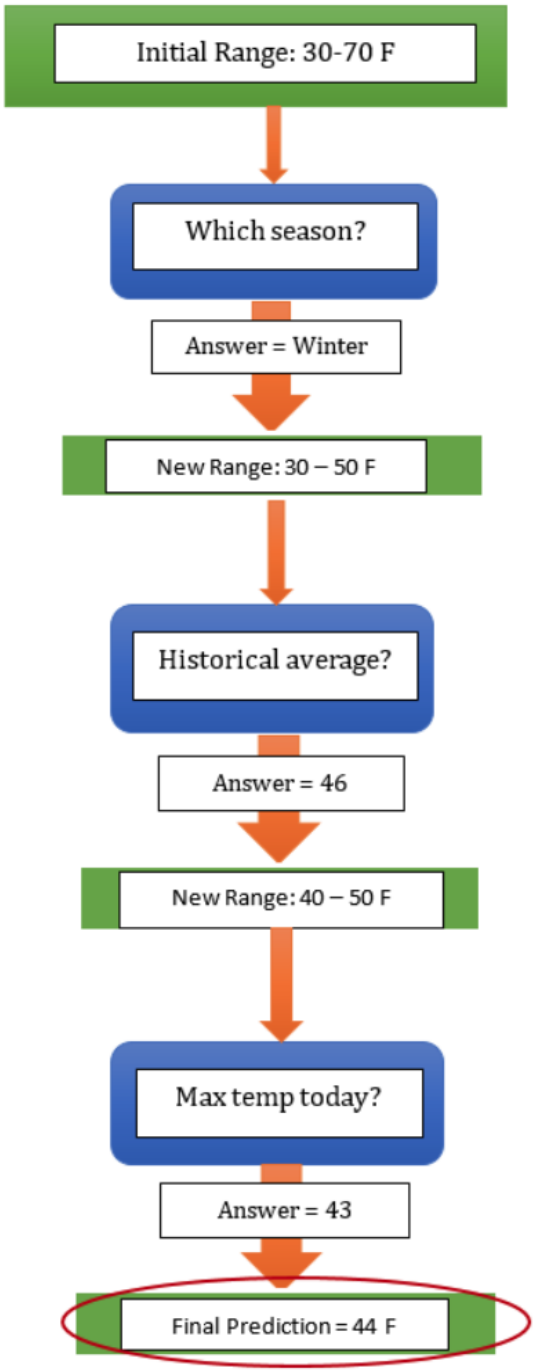


Figure 2.2: Flow chart example, predicting the maximum temperature in Seattle tomorrow. [11]

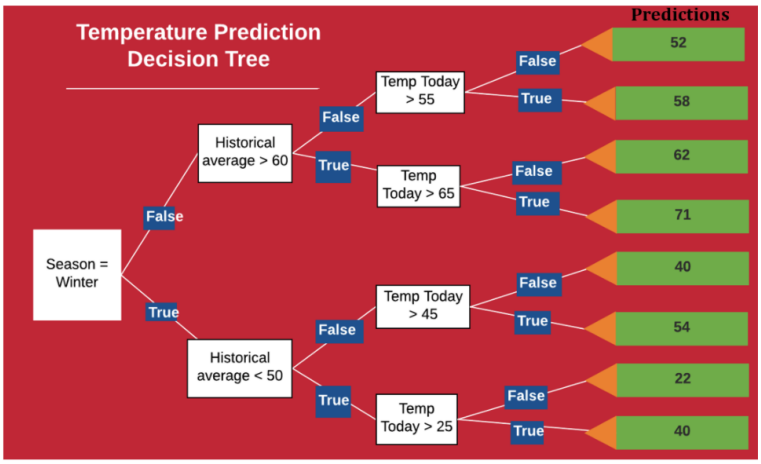


Figure 2.3: A Decision Tree
[11]

summer instead of winter, our range of predictions would have shifted higher. A decision tree considers all possible values and at each value attempts to minimize the residual sum of squares in order to make a decision, where as in our case, we relied on intuition to make our guesses. Further, we phrased our questions such that they could take on any number of answers, and a decision tree will consider all feasible values. Consider figure 2.3, which is an example of a real decision tree. This decision tree might look differently from our own flow chart, but it follows the same intuition, start at the node on the left, and progress through the tree answering the questions along the way. Furthermore, in machine learning, we give the model any historical data that is relevant to the problem domain (feature space), and the value we want it to learn to predict (target space). The model then learns any relationships between the data and values we want it to predict. Next when we ask the decision tree to make a prediction for tomorrow, we must give it the same data used during training, the feature variables, and it gives us an estimate based on the structure it has learned. [11]

2.3.1 A Brief Intro to Random Forests

Looking back at the original decision tree in Figure 2.2, the prediction of 44 degrees is probably wrong. If you went out and collected data on the weather tomorrow and constructed your own decision tree, yours would most likely be

wrong too. Both of our decision trees will predict a value that is most likely higher or lower than the true value of the weather tomorrow. In technical terms, this means that the predictions will have high variance, because they will be widely spread around the right answer. But now, consider hundreds of thousands of individuals going out, collecting their own data, and constructing their own decision trees. If we average all the estimates from the different decision trees, we will obtain a prediction with low variance. This is the fundamental idea of Random Forests. Continuing with our example, the flow chart of a human decision can be thought of as a singular decision tree, and the hundreds of thousands of other human flow charts can be thought of as other decision trees. When we take the average off all these decision trees, we have created a random forest.

Chapter 3

Building up to Random Forests

3.1 Bootstrapping and Bagging

In random forests and other machine learning algorithms, the goal is always to produce the most accurate model on the test data. This is challenging to do when constructing only one model, and thus using tools like boosting and bagging, we employ strategies to create many models. We then take the average which will hopefully yield a more precise result than just one tree. In order to understand bootstrapping, consider a descriptive statistic such as a mean. Let's assume we have gone out and collected many observations, the bootstrapping process is as follows:

1. Create many re-samples of the data set with replacement.

2. Calculate the mean of each re-sample.

Bagging, is a general-purpose method for reducing the variance of a statistical learning method such as a Random Forest. [7]. Using bootstrapping we can create many samples of the training data, apply our statistical learning method to each one, and then average their predictions to obtain a final result. Consider a set of independent observations $x_1, ..., x_n$. Also note the mean of these observations is $\bar{X}$, and the individual variance of these terms is σ^2. Note the variance of the mean is $\frac{\sigma^2}{n}$. In other words, averaging a set of observations reduces variance. [7].

3.2 Leave one out Cross Validation

Cross Validation is a statistical technique used to prevent over fitting, discussed in section 3.5. Cross validation is a clean and simple way to test the model on the training data."Leave one out Cross Validation involves splitting the training data set into two sections." [7] The first section is a single observation, for example (x_1, y_1) and the second section is the rest of the data set, for example $(x_2, y_2), ..., (x_n, y_n)$. The statistical learning method is fit on the $n-1$ observations, and the model is tested on the observation that was left out. [7] Because, the observation (x_1, y_1) was left out of the training, one can calculate a mean squared error(MSE_1), $(y_1 - \hat{y}_1)^2$ for this observation. This procedure is repeated by selecting another observation. For example (x_2, y_2) is chosen as the section to leave out of the training data, and the statistical learning method is trained on the $n-1$ observations, $(x_1, y_1), (x_3, y_3), ..., (x_n, y_n)$. And again, compute the $MSE_2 = (y_2 - \hat{y}_2)^2$. This approach is then repeated n times, and produces n squared errors. [7] "The leave one out cross validation estimate for the MSE is the average of these n test error estimates: $\frac{1}{n}\sum_{i=1}^{n} MSE_i$." [7]

3.3 K-fold Cross Validation

Another method of cross validation is called K-Fold cross validation, and is described as follows. In this method, one randomly divides the set of observations into k groups, or $folds$ (hence the name k-fold), all of approximately equal size. Similar to leave one out cross validation, one of these folds serves the purpose of the validation set, and the random forest is trained on the $k-1$ folds. The MSE is measured on the fold that is left out, and the procedure is repeated k times in the same way, using a different fold each time. The resulting MSE, $MSE_1, MSE_2, ..., MSE_k$ is computed by average these vales: $\frac{1}{k}\sum_{i=1}^{k} MSE_i$. [7]

3.4 Out of Bag Error Estimation

Another alternative to cross validation, is called Out of Bag Error Estimation. The process is as follows:

1. Trees are repeatedly fit to Bootstrapped sets of the observations.

2. On average, each bagged tree makes use of $\frac{2}{3}$ of the observations.

3. The remaining $\frac{1}{3}$ of the observations not used to fit a given bagged tree are referred to as Out of Bag observations.

4. We can predict the response for the i^{th} observation using each of the trees in which that observation was OOB.

5. This yields $\frac{B}{3}$ observations for the i^{ith} observation.

6. To obtain a single prediction for the i^{th} observation, average the predicted responses, in the case of regression, or take a majority vote (classification). [7]

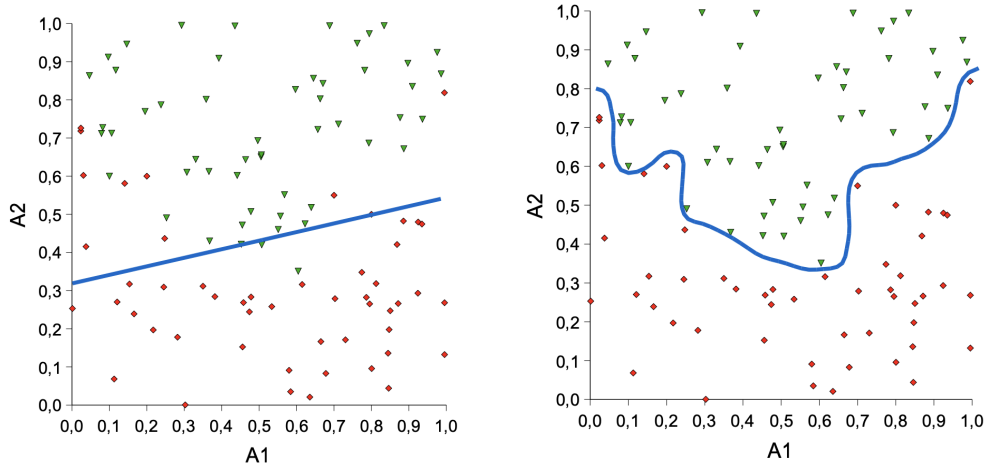
3.5 Overfitting

As mentioned earlier, statisticians cross validate to avoid the problem of over fitting. In machine learning, a researcher attempts to predict an outcome by training training data. In the case of decision trees, it is important that the model isn't fit perfectly to the training data. This is because the ultimate goal is to feed it data it has never seen, and obtain an accurate prediction. Therefore, it is important to use methods such as cross-validation, bootstrapping, bagging, and random forests, to ensure that models are not over fit to the training data. As explained in the next section, a random forest isn't allowed to even consider all variables when making a split which injects randomness into the model. Before getting into the random forest model, it is important to discuss the bias variance trade off.

3.6 Bias Variance Trade off

In order to best understand the Bias Variance Trade off and over fitting, lets consider an example motivated by the paper written by Pierre Geurts. [4] Now consider trying to find a line that splits the red diamonds and green triangles. Figure 3.1 displays two models presented in Dr. Geurts Thesis'. [4].

Both of these methods, (we will refer to the first one as $A1$ and the second one as $A2$, are splits of the data points on the graph but have their pros and cons. The first attempt $A1$, doesn't do a great job of classifying the data,



(a) First method of split the data - A1 (b) Second method of splitting the data - A2

Figure 3.1: Two different methods of splitting the scatter plot of points. [4]

because it is linear. From simply observing the data, a straight line will never perfectly classify this data. The resulting predictions produced by the classification will therefore be incorrect and biased. [4] The second model seems unbiased because of the perfect fit to the training data is "too good", because it "learned too much information from the training data". [4] If we were to test this second complex model on a new sample, it would very likely produce wildly different predictions, because it is over fit to the training data. $A2$ is referred to as a model with high variance, because there is a lot of variability in trying to predict the i^{th} value in using a bunch of different training data and models. [4]

3.7 Random Forest Model

Similar to Bagging, we build a number of decision trees on bootstrapped training samples.[2] But, when building decision trees, each time a split in a decision tree is considered, a random sample of m predictors are chosen as candidates from the full set of p predictors. The split is allowed to use only one of those m predictors. A fresh sample of m predictors is taken at each split. Typically we choose $m \leq \sqrt{p}$. Further, the number of predictors considered at each split is approximately equal to the square root of the to-

tal number of predictors. In other words, at each split, the random forest algorithm isn't even allowed to consider the majority of the available predictors. Suppose there is one very strong predictor along with other moderately strong predictors. In the collection of bagged trees, most will use this strong predictor in the top split. All of the bagged trees will look pretty similar, and taking the average of highly correlated values does not yield a substantial reduction in variance. Thus, Random Forests solve this problem by forcing each split to consider only a subset of the predictors. We will refer to this process as de-correlating trees. [7]

3.8 Variable Importance Measures

Another interesting aspect of Random Forests, are the variable importance measures. Recall the Residual Sum of Squares, or RSS, that is minimized at each split of the tree. Therefore we can look at how much the RSS is decreased due to splits over a given variable. A larger decrease indicates that the the predictor is important because the predictor reduces the RSS. A small decrease indicates that a given predictor did not help the tree learn very much.

3.9 Choosing the Correct Number of Predictors

Consider the example proposed in the textbook titled,"An introduction to statistical Learning." [7]. In this example, they applied a random forest to a high dimensional biological data set consisting of, "expression measurements of $4,718$ genes measured on tissues samples from 349 patients.[7] There are over $20,000$ genes in humans, and the goal is to predict the cancer types with the 500 genes that have the largest variance. Genes were classified on a scale of $1-15$ different levels, where one level is normal and the other 14 represent different types of cancer.[7] They proceeded to randomly divide the observations into a training and test set, and then applied random forests for 3 different values of m, the number of splitting variables. They found that $m = \sqrt{p}$ gave the smallest test error rate, which performed better than bagging. The results are displayed in the figure below. Also note the optimal number of trees is about 400. [7]

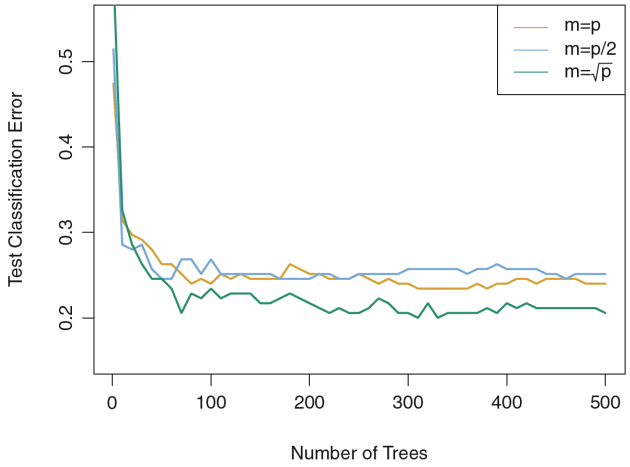


Figure 3.2: This figure displays the test error as a function of the number of trees. A single classification tree has an error rate of 45.75, which is clearly inferior to bagging and random forests. This figure shows that using a random forest, and thus using $m = \sqrt{p}$ as a decision rule, yields the smallest error rate. [7]

Chapter 4

Applied Model

4.1 Motivation

The motivation for my applied model came from a variety of papers, but for the purpose of this Thesis I will focus on two.

4.2 A Hybird Random Forest

The first paper, written by Groll et al., is called, "A Hyrbid Random Forest to Predict Soccer Matches in International Tournaments." [6]. In this paper, the authors attempt to predict and simulate the 2018 world cup. They do this by using data from past tournaments including past world cups, all the way back to the 2000 world cup. The authors propose "A hybrid modeling approach for the scores of international soccer matches which combines random forests with Poisson ranking methods." [6] Their Random Forest includes many different types of variables, all in the hope of estimating a more powerful random forest to predicting the winner of the 2018 world cup. Their paper expands on previous papers in the field, and they use a Bivariate Poisson distribution to estimate ranks for each of the teams in the world cup. Combining this new ranking variable they generate, with a plethora of other variables, they successfully generate a model which outperforms all other models in the field. This new hyrid random forest includes three different types of variables in the feature space.

1. Economic Factors

- GDP Per Capita: "To account for the general increase in the GDP during 2002-2014." [6]
- Population: The population of an individual country divided by the size of the global population.

2. Sportive Factors

   - ODDSET Probability: They gather bookmaker odds, and convert them into probabilities. (This is one of the main ideas I use in my paper).
   - FIFA Rank: The current rank of a team, by the FIFA Ranking system.
   - ELO Rating: Based on the Elo rating system, another way to rank soccer teams, and is said to, "Aims at reflecting the current strength of a soccer team relative to its competitors." [6]

3. Home Advantage

   - Host: A dummy indicating if the team is the host country
   - Continent: A dummy indicating if a team is from the same continent as the host country.
   - Confederation: Confederation of the soccer team.

4. Factors Describing the team's structure

   - Maximum number of teammates: This variable aims to measure a teams chemistry by the number of teammates playing together on the same club team.
   - Average age
   - Number of Champions League Players: The number of players that have competed in Europe's most elite soccer league.
   - Number of players abroad: The number of players that play on a club team not from the respective players host country.

5. Factors describing the teams coach

   - Age

- Duration of tenure

I take some variables and ideas from this paper to include in my own model. Most notably, and one variable the authors note is a very important predictor, is ODDSET Probability. They take the money line for each game from a German bookmaker and convert it into a probability. A further discussion of money lines will be included in a description of my variables. I also take the idea of a ranking variable, which in my model is represented by a teams odds of winning a match and their win loss record and percentage. The big idea I use from this paper is the simulation aspect. The authors take the probabilities outputted by the random forest, and use these probabilities to simulate the 2018 world cup. I attempt to do this with the $2017-2018$ NBA Playoffs.

4.3 Predicting National Basketball Association Winners

The second paper I examine, written by Lin et al., called, "Predicting National Basketball Association Winners" is a computer science thesis attempts to predict NBA playoff games using various models, including random forests. [8] They collect data on NBA box score information (note box score statistics will be explained in depth when discussing my model). Box scores provide summary statistics for sports games, and in basketball games, describe basic statistics such as points, assists, rebounds, steals, etc. This paper also includes a win/loss record statistic which describes the total number of wins and losses from teams that are playing each other. One of the most important ideas in this paper, which I include in my model, is the idea of aggregating each box score statistic. At each point in the season, the box score statistic is the sum of all of previous games. This paper is more relevant to my model, because it predicts NBA Basketball games, which is exactly what I attempt to do. Therefore I use many of the ideas in this paper, except for a difference in modeling techniques. For the feature space of their random forest, they subtract one teams box score statistics from the other, where as I include both the home team and away team's box score statistics.

4.4 Data Set and Variables

The data set I use for my analysis was downloaded from Kaggle. I was fortunate enough to pull a data set already containing both box score information and a separate data set containing betting information.[1] Both data sets contained game identification numbers, and thus combining the two data sets was simple. Therefore my final data set contained NBA box score information and betting data going all the way back to the 1950's. A description of my variables is as follows:

1. Game Date: The date of the game
2. Is Home: A dummy indicating if a team is home or not
3. Win Loss: A variable indicating the outcome of the game, (this is what I use as my target variable)
4. Win: Total Wins
5. Loss: Total Losses
6. Win Percentage: Wins divided by total games
7. Field Goals Made: Total shots made, either two point or three point shots
8. Field Goals Attempted: Total shots attempted, either two point or three point shots
9. Field Goal Percentage: Ratio of total shots made divided by total shots attempted
10. Free Throws Made: Total shots made from the free throw line, either from a foul or technical foul
11. Free Throws Attempted: Total shots attempted from the free throw line, resulting from a foul or technical foul
12. Free Throw Percentage: Free throws made divided by free throws attempted
13. Three Point Field Goals Made: Total three point field goals made

14. Three Point Field Goals Attempted: Total three point field goals attempted

15. Three Point Field Goal Percentage: Total three point field goals made divided by total three point field goals attempted

16. Offensive Rebounds: Players get offensive rebounds from gaining control of the ball off of a missed field goal or free throw attempt on the offensive side of the court

17. Defensive Rebounds: Players get defensive rebounds from gaining control of the ball off of a missed field goal or free throw attempt on the defensive side of the court

18. Total Rebounds: A sum of Defensive of Offensive Rebounds

19. Assists: A pass leading to a field goal

20. Steal: Forcing a change of possession

21. Blocks: A field goal attempted that is swatted away before it can get to the basket

22. Turnovers: A team loses possession of the ball, before a field goal attempt

23. Personal Foul: A physical foul committed on the court

24. Spread: The point spread or handicap, is the number of points a team is predicted to win or lose by. For example in the final game of the NBA Finals the Warriors were 4 point favorites, meaning their spread was -4 and the Cavaliers Spread was $+4$. This means that if a bettor bet the warriors -4, the warriors would have to win by at least 5 points in order to win the bet. On the other hand, if a bettor took the Cavaliers $+4$, then the Cavaliers can't lose by more than 3 points in order for the bettor to win their bet.

25. Money Line: Simply put, a money line is betting on a team to win a game. Consider the following example where the Yankees are playing the Blue Jays in baseball: Yankees -165 vs Blue Jays $+140$. This means the Yankees are the favorite to win the game. Also note that

in order to win \$1 a bettor must place \$1.65. On the other hand, if a bettor thinks the Blue Jays are going to win, they can place \$1 to win \$1.40. Money lines can be converted to probabilities through the following formula in which I include in the feature space of my random forest. There are two cases, either a negative money line, in the example above, the Yankees, and a positive money line, in the example above the Blue Jays. For the case of the minus money line, the formula is: $\frac{(negativemoneyline)}{((negativemoneyline))+100)}$, and the positive money line is $\frac{100}{(positivemoneyline+100)}$. [10]

Variable Descriptions for box score statistics come from the website "Sports Rec". [5] Variable Descriptions for the Money Line and Spread statistics comes from the website "SIA Insights". [3]

4.5 Random Forest Model

My model attempts to combine both aspects of the two papers above, and predict the winners of the 2017-2018 NBA Playoffs, using the results from the regular season. Further, my training data is the 2017-2018 Regular Season, and my test data is the 2017-2018 Playoffs.

4.5.1 Modeling Techniques

First, note that the box score variables are cumulative sums. For example, if the Lakers are playing the Clippers in the their third game of the season, the points variable for the Lakers will equal the sum of their points in the first two games of the season. This is important because when I ask the random forest to predict the winner, if the forest knew the points scored by both teams, it would easily predict the winner of that game. If I were trying to simulate the playoffs right now, I wouldn't know any of the box score information, because the game wouldn't have occurred. Secondly, from previous literature on modeling NBA basketball games there were two good ways to proceed. Either, I could create a feature vector where the variables represent the difference between the box score statistics of the two teams, or I could include both team's box score statistics in the model. For example, the model would contain away team points and home team points. I tried both methods in this case, and the latter method performed better, and therefore I proceeded with the latter method.

4.5.2 The Random Forest

As mentioned above, my training data is every game from the 2017-2018 regular season, and my test is every game from the 2017-2018 NBA Playoffs. Note that each row of the data set, represents one game, containing the box score statistics of both teams, the respective probabilities of each team to win the game converted from the teams money line, and their spreads. Each row also contains a dummy variable indicating if a team is home or away. My training data contains $1,039$ games, and the test data contains 82 games. I trained my model using 100 trees, and the results are represented in Figure 4.1. The right half of the bracket represents the actual Playoff bracket and what occurred. The grey column represents the first round match ups, the second column shows the second round match ups, the third column shows the conference finals, the fourth column shows the finals match up, and the fifth column shows the winner. As shown in the bracket, the Golden State Warriors played the Cleveland Cavaliers in the 2018 NBA Championship, and the Warriors won the series 4 games to 0. In each series I predicted every single game and reported the result of each game. Also note that if a team is highlighted in green, the random forest predicted more than half of the games within a series correctly. If a team is highlighted in yellow, the random forest predicted half of the games correctly. If a team is highlighted in red, the random forest predicted more than half of the games incorrectly. The left hand side of the figure shows the number of games predicted correctly and incorrectly within each series. Note that in the first round of the playoffs the random forest did a pretty good job in predicting these games - especially in predicting the Boston Milwaukee series which went to seven games, (note NBA playoff games are best of seven, who ever wins 4 games first advances). The random forest did a stellar job of predicting challenging series, in which there wasn't a clear favorite. For example consider the Eastern Conference finals, Boston Celtics vs the Cleveland Cavaliers. In this match up, a young Boston Celtics team faced off against a Lebron James lead Cavaliers team. The series was hard fought, and the Cavaliers ended up winning in 7 games. My model successfully predicted the outcome of the first 6 games, but in the seventh game in which both teams had almost an equal chance of winning, as predicted by Vegas Bookmakers, my model predicted the winner incorrectly. This game came down to the last few minutes, in which Lebron James carried his team to the finals. Further, this game was nearly impossible to predict, and many described as a coin flip. My model, along with NBA experts

had trouble predicting this game, and sometimes when super-stars such as Lebron James have impeccable games, like he did in game 7 of the Eastern Conference finals, it makes forecasting very challenging. Similar to the series before, when the Cavaliers swept the Toronto Raptors, my model along with experts struggled to predict the outcome of this series. Many thought the Raptors would win this series, but again, it is impossible to control for and include in the model, what some refer to as the "Lebron James" affect. Some people called him Super Human in this series, and that is nearly impossible to control for. The Raptors were the number one seed in the East and the Cavaliers were the number 4 seed. Because my model learned from data in the regular season, it struggled to predict this match up, because no signs in the regular season pointed to the Cavaliers playing this well in the series. In future work, it would be smart to include variables for player accolades in order to attempt to control for a team having super stars. Furthermore, my model surprisingly struggled to correctly predict the outcomes of the Golden State Warriors Series'. This was very surprising, especially in the second round of the playoffs where the Golden State Warriors had a significant edge over the New Orleans Pelicans. My model predicted the Pelicans to win 3 games which they actually lost. This was confusing given the Warriors were favorites in every game. Further, when the Warriors matched up in the Western Conference Finals against the Houston Rockets, which many called the best series in the playoffs, my model correctly predicted 6 out of the 7 games even though the teams were very evenly matched up. Again, this was perplexing given the series before the Warriors were heavy favorites, and it performed so poorly. The problem came up again in the finals, when the random forest only correctly predicted the first two games, but struggled with the final two. Although the Warriors weren't heavy favorites, they were clearly the superior team, and my model struggled. In the end, the Random Forest did a good job predicting some of the more complicated series, but really struggled with some of the series in which there was a clear favorite.

4.5.3 Discussion of Results

Another interesting feature of Random Forests are the Variable Importance Graphs. Because decision trees use the Residual Sum of squares as a decision rule, one can look at how much the Residual Sum of Squares is decreased at each split, and which variable is responsible for the largest decrease. Therefore we can see the most important variables in predicting the winner of

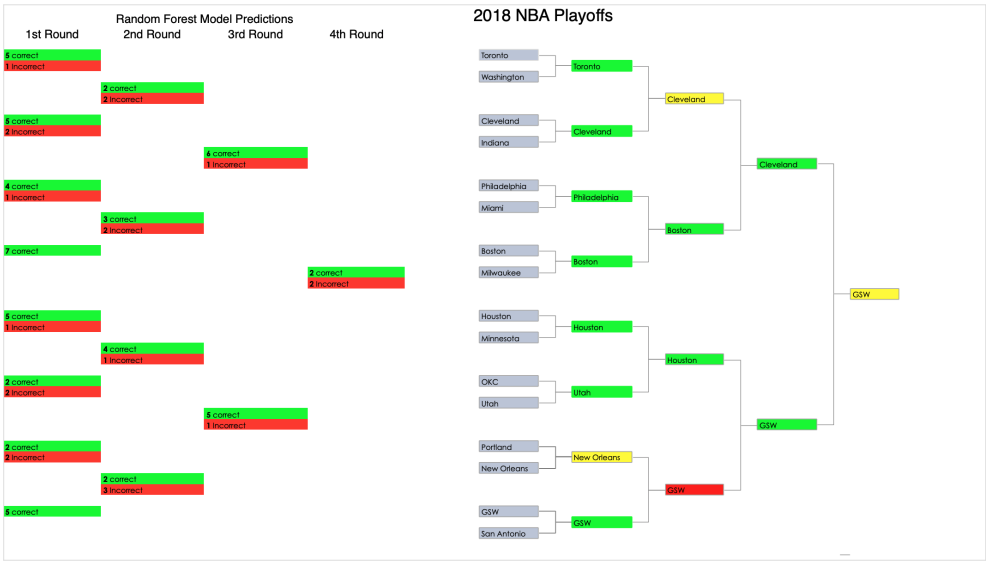


Figure 4.1: 2018 Playoff Bracket

basketball games, shown in Figure 2. The top 4 variables, win percentage of the home team, probability of winning of the home team, probability of winning for the away team, and win percentage of the away team, all make sense as the most important predictors. In the other two papers I discussed earlier in the chapter, ODDSET, or bookmaker odds, was one of the most important variables in determining a winner in the first paper, [6], and win percentage was the most important in determining a winner in the second paper. [8]. Therefore I hypothesized combining the two would create a superior ranking method. This also goes to show how spot on bookmakers are in predicting the outcome of matches. Aside from the most important ranking variables, the most important box score variables are field goal percentage, and three point field goal percentage. This makes sense given that the amount of points a team scores is highly correlated with their field goal percentage, but doesn't help to explain why the model struggled so heavily with Golden State given they always were favored by bookmakers and had a high three point and field goal percentage.

4.5.4 Shortcomings

Overall the model did a good job of predicting the 2017-2018 NBA Play-

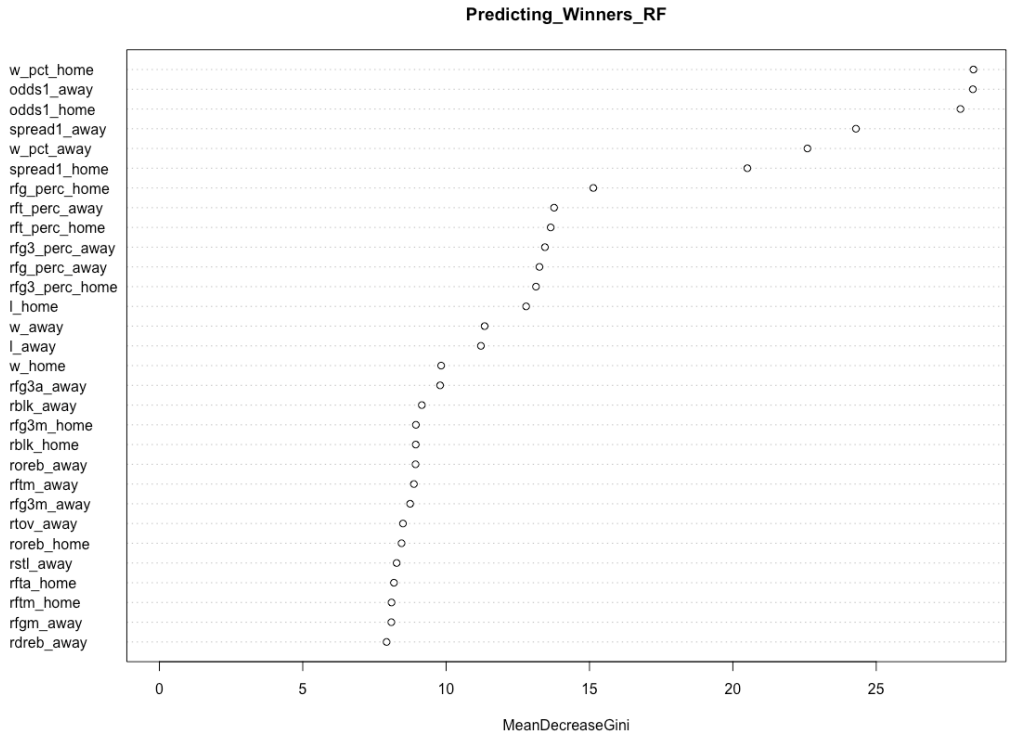


Figure 4.2: Variable Importance Graph

offs, but like any model, the results are not perfect. In this section I will discuss some ways to improve the model in future work. The most important addition would be including a ranking variable, which would track a teams position in their respective conference at each point in the season. Secondly, including a measure for accolades of individual players on a team would help improve the model in the playoffs, when teams with superstars are facing teams without superstars. For example, in the Toronto vs Cleveland series, Cleveland unexpectedly swept the number one seeded Toronto Raptors. If the model knew that Lebron James had 3 championships under his belt, where as all players on the Toronto Raptors combined had 0, maybe the model would have given a different prediction. Moreover, I think it would help to include descriptive statistics about the coaches, their win loss records, win percentages and tenure in the league. Also including more descriptive statistics of the players on each team would improve the accuracy of the model. For example number of all star appearances, number of MVP awards, number of first team all-NBA selections, to name a few.

Chapter 5

Simulations

In the last chapter of my thesis, I attempt to simulate the 2017-2018 NBA Playoffs using the probabilities outputted by the random forests.

5.1 Motivation

In the paper, "A Hybrid Random Forest to Predict Soccer Matches," the authors use the probabilities outputted by the hybrid random forest along with betting odds probabilities in the attempt to simulate each round of the world cup.[6] I attempt to do the same, but only using the probabilities outputted by the random forest.

5.2 Theory

Due to the way Random Forests are constructed, one can look at the probability of an outcome, by dividing the number of trees which make a certain prediction by the total number of trees. The resulting prediction can be thought of as a probability. For example, in the case of my model, if 80% of the decision trees predict a team to win, the probability of that team winning given by the random forest is 80%. I take this probability for the home team, and then use Bernoulli trials to simulate each series. Bernoulli trials are a fancy way of saying coin flips, but each Bernoulli trial, or flip of the coin, is weighted by the win probably given by the random forest. I chose to use Bernoulli trials, because like coin flips, there is a lot of luck involved in winning a basketball game, especially a close one. For example, even if

Figure 5.1: NBA Playoff Simulations

a team has an 80% chance of winning a game, there is no guarantee they will win, because of factors outside their control that might happen during a game, bad calls by the ref, injuries, or even the other team playing better than expected. Bernoulli trials introduce the same variability into the simulations in which I believe occurs in a basketball game.

5.3 Bernoulli Trials

As mentioned above, I take the probabilities outputted by the random forest, and use Bernoulli Trials to simulate each game, within a series. Note that a playoff series in the NBA playoffs is a best of seven series, meaning who ever wins 4 games first advances to the next round. The results of the simulations are reported in Figure 5.1. Figure 5.1 shows the teams that advanced in the playoffs, but the color coding shows the predictions of my simulations. If a team is colored in green, this means I simulated the series correctly. If

a team is colored in red, this means I simulated the series incorrectly. For example, the simulation incorrectly predicted that the Pelicans would beat the Golden State Warriors. It also incorrectly predicted that the Rockets would beat the Golden State Warriors, and surprisingly, in this singular simulation, the Cleveland Cavaliers beat the Golden State Warriors in the NBA Finals in seven games. Given the results of the random forest, it makes sense the simulations predicted a different winner than what actually happened. Although I may have simulated many of the series correctly, there are many cases where the number of games it took for a team to win, was more or less than what actually happened in the playoffs. For example, in the series between the Cleveland Cavaliers and Toronto Raptors, where the Cavaliers swept the Toronto Raptors and won in 4 games, in my simulation the Cleveland Cavaliers won in 6 games. Also note that when simulating games that didn't occur, I had to be creative in which money lines and spreads to include in the feature space, because those games didn't actually occur, and thus had no associated odds. Coming up with the appropriate money line and spreads took some knowledge of the NBA Playoff structure. The NBA playoffs follows a format as follows:

1. The higher ranked will host the first two games.
2. The lower ranked team will host the second two games.
3. If the series isn't decided in 4 games, the 5^{th} game will be hosted by the higher ranked seed.
4. If the series isn't decided in 5 games, the 6^{th} game will be hosted by the lower ranked team.
5. If the series isn't decided in 6 games, the higher ranked team will host the 7^{th} and final game.

I used this format to create a rule. If the 5^{th} game didn't actually happen, I knew which team would be the home team if it did happen, and I used the money line and spreads associated with the first game of the series. If the 6^{th} game didn't actually happen I used the money line and spreads from the 3^{rd} game of the series because I knew the lower ranked team would be the home team in this case. Lastly, if the 7^{th} game of the series didn't actually occur, I used the money line and spreads from the first game of the series when the higher ranked team was home, because I knew the higher ranked team

would have been at home in this game if the game had actually occurred. Although the money lines and spreads are not exact representations of what they would have been if the games had occurred, they would be pretty close. Lastly, the general simplified version of my simulations are as follows:

1. Flip a weighted coin, using the probability from the Random Forest.
2. If the flip resulted in a 1 the home team won, otherwise the away team won.
3. Keep doing this until one team wins seven games, and therefore a winner of the entire series is decided.

5.4 Discussion of Results and Shortcomings

5.4.1 Results

The main discussion of results will focus on the amount of games that occurred in reality in contrast with the amount of games that occurred in my simulations. For example in the first round, where the Boston and Milwaukee game went into seven games, my simulation did as well, and successfully simulated the winner. Although within the series, I simulated Boston losing the second game in which they won, and winning the fourth game in which they lost. Ultimately, I reached the same conclusion, and simulated correctly the winner of the series. Similarly, in the second round of the NBA playoffs, when the Toronto Raptors played the Cleveland Cavaliers, I simulated six games, in which the Cleveland Cavaliers won, where as in reality it only took them four games. The simulations incorrectly predicted the Golden State Warriors losing three out of the four series in which they won. For example, in my simulation, they lost the series $4 - 0$ to the New Orleans Pelicans, a preposterous result, given the fact they were by far the dominant team. This takes me into the next topic of shortcomings.

5.4.2 Shortcomings

The simulations were largely successful, given the probabilities from the random forest. I did not expect my simulations to predict Golden state to win the series in which the random forest predicted they would lose, because

the probabilities inputted in the Bernoulli trials were so low. Therefore the theory behind the simulations is sound, but with a few tweaks, they could be vastly improved. For example, I only conducted one simulation, but in many other papers, including the the Hybrid Random Forest [6], simulations were conducted thousands of times. Due to the variability in simulating with Bernoulli Trials, if we simulate something thousands of times, the results will eventually approach the true value of the probability given by the Random Forest. Furthermore, the simulations are only as good as the probabilities given by from the Random Forest. If I were to conduct simulations again, I would first conduct the simulations thousands of times, and ideally would have a better decision rule or algorithm for making spreads and money lines for games that didn't occur. The biggest problem in my simulations is that I am not able to simulate games that didn't occur, because I don't have a way to create one of the most important feature space variables, the money line. If could have created my own odds for these games, I would not only be able to simulate the series, but also would be able to take the winners of those series and match up them up against different teams even if the match up hadn't occurred. Therefore I would be able to simulate the results of the entire NBA playoffs instead of just the individual series. Due to the fact that I do not know how to create odds for a match up that didn't occur, I believe my simulations are limited and could be improved with this knowledge.

Chapter 6

Conclusion

This paper explored the fundamentals of decision trees, and techniques used to build random forests. Random forests are an advanced machine learning technique first introduced by Leo Breiman in 2001 [2]. The basic idea is to aggregate many uncorrelated weaker learners or decision trees in order to create a low variance statistical learning model. This paper attempted to predict and simulate the 2017-2018 NBA playoffs using the win probabilities given by the random forest and trained on the 2017-2018 regular season games. The random forest did a good job of predicting winners in most cases, but struggled with "easier" games. For example it struggled identifying Golden State as one of the strongest teams in the playoffs. Further, the simulations were theoretically sound, but without running them thousands of times, and being able to generate bookmaker odds for games that didn't happen, accurate results were tough to obtain. In conclusion, with a few tweaks to the random forest and simulations, I am confident one can correctly classify a majority of basketball games in the NBA playoffs.

Bibliography

[1] Kaggle nba historical stats and betting data. https://www.kaggle.com/ehallmar/nba-historical-stats-and-betting-data. Accessed: 2020-02-26.

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] F. Doyle. Money line bets. https://news.sportsinteraction.com/guide/moneyline-betting-explained, 2020.

[4] P. Geurts. *Contributions to decision tree induction: bias/variance tradeoff and time series classification*. PhD thesis, University of Liège Belgium, 2002.

[5] J. Gordon. How to read a basketball box score. https://www.sportsrec.com/read-basketball-box-score-2062845.html, 2007.

[6] A. Groll, C. Ley, G. Schauberger, and H. Van Eetvelde. A hybrid random forest to predict soccer matches in international tournaments. *Journal of Quantitative Analysis in Sports*, 15(4):271–287, 2019.

[7] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[8] J. Lin, L. Short, and V. Sundaresan. Predicting national basketball association winners. *CS 229 FINAL PROJECT*, 2014.

[9] D. Lock and D. Nettleton. Using random forests to estimate win probability before each play of an nfl game. *Journal of Quantitative Analysis in Sports*, 10(2):197–205, 2014.

[10] K. Miller. Converting sports betting odds. https://www.gamblingsites.org/sports-betting/odds-converter/, 2020.

[11] K. Will. Random forest in python, Dec. .