

*Predictive Model for the NCAA Men's Basketball Tournament*

**An Honors Thesis (HONR 499)**

**By**

Cody Kocher

&

Tim Hoblin

**Thesis Advisor**

Prof. Gary Dean

**Ball State University**

**Muncie, IN**

*April 2017*

**Expected Date of Graduation**

*May 2017*

SPColl  
Undergrad  
Thesis  
LD  
2489  
.Z4  
2017  
.K63

## **Abstract**

Since 1940 the National Collegiate Athletic Association (NCAA) has held an annual competition pitting the best college basketball teams against each other. This single-elimination tournament has grown to include sixty-eight teams vying to be crowned national champion.

The teams are not the only ones competing for glory during the tournament. From its humble beginnings, the tournament that is aptly nicknamed March Madness has grown to include tens of millions of people betting billions of dollars on who they think will win each of the sixty-three games that make up the tournament. For decades, people ranging from die-hard sports fanatics to people who have never watched a game of basketball have attempted the difficult task of predicting the outcome of this tournament.

Many have debated whether there is a true statistical method for predicting outcomes in this tournament, so we put that to the test. We attempted to predict the 2017 NCAA College Basketball Tournament by applying generalized linear models and random forests, predictive modeling tools widely used in statistics. Based on our various predictive models, we submitted 21 brackets to ESPN's Tournament Challenge and tracked their success against the 18.8 million other entries submitted by the general population. We analyzed our findings based on the overall rankings of our entries on ESPN to determine if our predictive models held a statistical advantage over the population.

## **Acknowledgments**

We would like to thank our advisor Professor Gary Dean for the knowledge and support that he has given us while working on this project as well as during our entire Ball State career.

We would also like to thank Shea Parkes for taking time out his busy schedule to share his expertise in the field of predictive modeling and how to best approach our project.

We would lastly like to thank our families and friends for their support and interest throughout this project.

## **Table of Contents**

Abstract	2
Acknowledgments	3
Process Analysis Statement	5
Introduction	8
What Others Have Done	9
Data	13
Generalized Linear Models	16
Decision Trees and Random Forests	20
Results	27
Conclusion	32
Appendix	34
Works Cited	39

## **Process Analysis Statement**

This project was both challenging and rewarding because predictive modeling is one of the greatest challenges in mathematics. We can analyze the past and use that knowledge to try to predict future events, but this process is almost never completely successful. There is a special challenge in predicting the NCAA Men's Basketball Tournament because this tournament is consistently one of the most unpredictable sporting events of the year. This gave us a pair of unique challenges, to expand our knowledge and skill in predictive modeling, and to know that at least some of our predictions were going to be wrong, no matter how well we constructed our models.

Our research took many forms throughout the project. Because there are many methods of predictive analytics, we had to start with a wide scope of possible ideas and narrow them down into what we perceived to be the most applicable methods to our project. This involved familiarizing ourselves with many different methods and determining pros and cons to each method. Our research often guided us as certain methods led to other modeling techniques and mathematical concepts. These additional paths were not always fruitful, so it was important for us to recognize which paths to follow.

Working to predict the 2017 NCAA College Basketball Tournament using predictive modeling also gave us an opportunity to explore this field in regards to an event with readily obtainable data. This project also has another advantage: even though we knew that none of our predictions would be perfect, there is no real consequence to being wrong. This notoriously unpredictable tournament still offered a significant task because of the many directions the project could take. Exploring new mathematical, statistical programs, and having to apply in-depth analysis of our findings throughout the process certainly proved to be a worthy challenge.

Computer programs such as Microsoft Excel and the statistical package R proved to be a particular challenge of this project. We entered this project with very little combined experience with R, but the vast majority of our final models were generated using R. This shows the tremendous growth in knowledge we achieved through this project, which was very satisfying personally. Although we had more previous experience with Excel, it was still rewarding to apply our knowledge to create purpose-built spreadsheets for running our projections.

While utilizing these statistical software, we learned that efficiency and innovation are key. Creating data sets that worked in both Excel and R was not always easy. With the vast amount of data and analysis that was sorted through, it was important to manipulate our data in a variety of ways while still making sure the format was compatible between both programs. This taught us to be more innovative with our data in Excel so it could be more easily transferrable to R. Because we were less familiar with R, some of our data sets would not transfer perfectly. This taught us the value of careful attention and patience while working in these programs. One small error or formatting decision in Excel could make our code in R completely worthless, and it was not always easy to see why. This could be frustrating at times, so being patient and paying careful attention to detail helped us to problem solve more efficiently.

While this is certainly not the first time mathematicians have tried to apply predictive modeling to the NCAA Tournament, we believe this project provides a new outlook to the problem as well as represents significant personal growth. Predictive modeling is a very powerful technique that can be applied in many ways. Our project represents an opportunity for us to begin to explore these techniques in a low-risk environment as well as the opportunity to improve on these models for tournaments to come.

Looking back, there were many things that we could have done differently or more efficiently, but this was still a very enjoyable project. We made a significant application of our mathematical knowledge while also working on something that was of great personal interest to us. We knew when we started that whatever we created was going to be wrong, and to a degree it was, but this was still a genuinely fun project and a memorable experience.

## Introduction

The NCAA Division I Men's Basketball Tournament goes by many names. March Madness, The Big Dance, and simply The Tournament are just some of the names you might hear come March. Every year, 68 of the best college basketball teams in the country are selected for the tournament and play through a single elimination bracket until one team is crowned the national champion. The results of this tournament are notoriously unpredictable, with small schools frequently upsetting larger schools with more talented teams. Every year millions of people across the world fill out brackets attempting to predict the results of the tournament, but no perfect bracket has even been documented. Researchers have estimated the probability of picking a perfect bracket between 1 in 4,294,967,296 and 1 in 9,223,372,036,854,775,808, depending on the calculation (forbes.com). The goal of our honors thesis is not necessarily to predict the perfect bracket but to see if there is a mathematical way to more accurately predict the tournament and apply it to the 2017 NCAA Men's Basketball Tournament.

68 teams are invited every year to the NCAA Men's Basketball Tournament. Four play-in games eliminate four teams, leaving the 64 that make up the final bracket. The Round of 64 consists of 32 games, which eliminates half of the 64 teams. The second round reduces these 32 teams down to 16, referred to as the Sweet 16. The Sweet 16 is then cut down to the Elite 8, who play each other for one of the spots in the Final Four. These four teams compete for the two spots in the national championship game, which decides the team that will be named national champions. The traditional method for scoring tournament brackets is to award 1 point for each Round of 64 game correctly predicted, 2 points for each Round of 32 game, 4 points for every Sweet 16 game, 8 points for each Elite 8 game, 16 points for each of the Final Four games, and 32 points for correctly predicting the national champion. Almost every major bracket pool uses

these scoring rules, with the only exception being ESPN.com, which simply multiplies each of these values by 10. This creates a maximum possible bracket score of 192 (or 1920 on ESPN.com).

The ultimate goal of this project was to develop a system to predict the highest-scoring bracket possible. We did this by employing common techniques in predictive modeling today: generalized linear models, decision trees, and random forests. Using these methods, we developed an assortment of models to predict the outcome of the 2017 NCAA Tournament. We then entered these brackets on ESPN.com to track our results against the general population.

## **What Others Have Done**

Attempting to predict the NCAA Tournament is not a brand-new phenomenon. There are many algorithms out there that have been made public over the years. From well-known statisticians, such as fivethirtyeight's Nate Silver, to machine learning competitions on kaggle.com, many have tried their lot at using data to aid their bracket predictions.

Many approaches have been taken to make predictions, some very complex but others quite simple. Furthermore, a wide array of mathematical techniques involving various combinations of team statistics, and even the creation of new statistics, have been successfully utilized for these predictions. Composites of various human and computer rankings, logistic regressions, and application of complex probability concepts are just a few examples of methods utilized over the past few years. Exploring some of these models gave us important insights to consider when creating our own models.

Nate Silver is a statistician who gained respect applying statistics to baseball and later gained fame by predicting 49 of the 50 states in the 2008 presidential election. He has also rather successfully tried his hand in predicting the NCAA Tournament. Silver's model has been based on a composite of 7 general, equally weighted team ratings (5 computer rankings and 2 human rankings) up until this year. For the 2017 Tournament, a sixth computer ranking was introduced into the model. Each computer rating is generally based around very similar base statistics such as wins and losses, strength of schedule, margin of victory, and offensive and defensive efficiency. These statistics are all based on performance throughout the season.

The difference in these ratings is relatively small, usually only varying in terms of how specific statistics are calculated and weighted. Despite the marginal differences, Silver notes, "even small differences can compound over the course of a tournament that requires six or seven games to win" ([fivethirtyeight.com](http://fivethirtyeight.com)). Combining these computer ratings smooths out any biases or impurities in a specific model.

Some additional statistics these ratings consider are not relevant to season performance, but a team's status going into the tournament. For example, some rankings consider how far a team must travel for a tournament game. The tournament matchups do not take into consideration how far a team must travel, so a highly rated team could potentially travel across the country to play a lower rated team playing less than 50 miles from home. This has been shown to have a larger effect than previously thought in predicting games. The other commonly used rating adjustment accounts for injuries in a team's starting line-up. If a team loses a pivotal player to injury for the tournament, this can have a significant effect on the team's success. On the contrary, if a team has recently regained a key contributor, their rating could be boosted.

In addition to the 5 computer ratings, Silver's system also includes the tournament selection committee's seeding of the 68 teams as well as preseason polls from the Associated Press. The preseason polls shed light on how a team was expected to do throughout the season based on player and coaching experience and talent. This can explain why experienced teams who may have underachieved during the season still make deep tournament runs and vice versa. The NCAA selection committee is the committee that ultimately determines what teams make the tournament and seeds them to determine matchups. This seeding gives a general insight to a team's overall performance throughout the year per experts of the sport.

The additional computer ranking included in the 2017 Tournament is fivethirtyeight's own model called an Elo rating. The Elo system, originally created by Arpad Elo to rate chess players ([fivethirtyeight.com](http://fivethirtyeight.com)), is a simple model based solely on the final score of a game, home-court advantage, and each game's location. The Elo rating has been applied to many sports prior to college basketball and has made a nice addition to Nate's composite model. Nate Silver combines an array of complex ratings into a simple probability model.

Simple models often see more success in general predictive modeling applications, and the NCAA Tournament is no exception. Kaggle.com, a site that hosts a variety of predictive modeling competitions, held a competition called "March Machine Learning Mania" in 2014; a contest for predicting the respective NCAA Tournament. The *Journal of Quantitative Analysis in Sports (JQAS)* featured five innovative modeling techniques that performed well in the competition. These models employed a vast array of strategies. For example, one model incorporated a mixture of multiple logistic regressions, gradient boosted decision trees, and neural networks based on similar statistics used in fivethirtyeight's model. It also included a

method for decontaminating certain variables due to incorporation of previous tournament data in its predictions.

Another model went a different direction creating what the author calls “Nearest-Neighbor Matchup Effects.” While employing a simple linear model to rank teams, the model also made adjustments for over and under performance in team matchups based on outcomes against teams with similar strengths. This accounts for upsets and other surprising results seen throughout the season that are not likely to be repeated.

Despite many other innovative and advanced models, the winners of the 2014 competition used a rather simple model. Michael Lopez and Gregory Matthews created a weighted average of two logistic regressions. This plays to the point that simple models can do the job just as well or even better than more complex models.

Another previous study on this topic was written by John Ezekowitz and published in the *Harvard Sports Analysis Collective*. Ezekowitz conducted his analysis with the assumption that games in the NCAA Tournament are fundamentally different from those during the regular season. To test this, he used a variety of publicly available statistics that quantify a team’s regular season, and also developed a few of his own statistics to measure factors that he felt were important in the tournament. In particular, he developed statistics to quantify a team’s confidence and tournament experience. To measure confidence, he quantified the value of a team’s regular season wins over other tournament teams, under the theory that teams who had beaten a lot of tournament teams during the regular season would be more confident in their ability to do so during the tournament. To evaluate prior tournament experience, he looked at the number of players a team had returning from the previous year and combined this with their success in the tournament the previous year. The theory behind this was that teams with a large number of

players with a history of tournament success would be well suited to be successful in the tournament during the coming year. Using this model, Ezekowitz was able to predict the tournament more effectively than many of the other computer ranking systems that did not use his confidence and experience metrics.

H.O. Stekler and Andrew Klein of George Washington University looked at a variety of publicly available rankings. To do this, Stekler and Klein collected and combined between 30 and 45 sets of rankings published for each year between 2003 and 2010. This combination of rankings was then compared to the rankings implied by the tournament selection committee who assigns each tournament team a seed between 1 and 16, with 1 seeds being the best. Stekler and Klein then tested to see if the combination of outside rankings was a better predictor of tournament games than the seedings given by the selection committee. Their study showed that these rankings usually produced similar results; however, the combined rankings performed slightly better overall than the seedings. The study also showed that these methods of prediction were much more effective during the first three rounds of the tournament, and were not significantly better than chance during the later rounds of the tournament.

## **Data**

For our analysis, we collected data from eleven years of NCAA Men's Basketball Tournaments, dating from 2006 to 2016. Each of the statistics collected falls into one of four categories: general information, offense, defense, and ball control. The majority of the statistics collected come from the archived national statistics on NCAA.com, with a few exceptions. Turnover margin from 2006-2008, strength of schedule from 2007-2016, free throws attempted

per game for all years, and opponent free throws attempted per game for all years were collected from teamrankings.com. Strength of schedule from 2006 was collected from cbssports.com.

The statistics in the general information category include: NCAA Tournament seed, season win-loss record, average margin of victory (or defeat), and strength of schedule.

The statistics in the offensive category include: points scored per game, assists per game, field goal percentage, three-point field goals made per game, three-point field goal percentage, free throws attempted per game, and free throw percentage.

The statistics in the defensive category include: points allowed per game, blocks per game, steals per game, field goal percentage defense, and opponent free throws attempted per game.

The statistics in the ball control category include: rebound margin per game and turnover margin per game.

Once the statistics were collected, we worked to deal with any of the issues and inconsistencies that arose in the data. These inconsistencies were either caused by rule changes or by changes in the way a statistic was calculated. For example, starting in the 2009 season, the three-point line in college basketball was moved back one foot, causing three-point percentage to fall by almost two percent. We cannot directly compare values from before and after this rule change, since it would have been easier for a team to have a three-point percentage of 40%, for example, under the old rules than under the new rules. Another impactful rule change took place at the beginning of the 2016 season, when the shot clock was reduced from 35 seconds to 30 seconds, which increased the pace of play and increased average scoring by around 5 points per game. An example of inconsistent formulas being used to calculate a statistic can be seen in the strength of schedule data. Teamrankings.com used a different formula to compute strength of

schedule between 2012 and 2016 than it did from 2007 to 2011, and the 2006 data comes from cbssports.com, which used another different formula. This makes it impossible for us to directly compare the strength of schedule from one year to another.

To deal with this issue, we normalized the data for each year by subtracting the average value of that statistic for a given year and dividing by the sample standard deviation for that year. This process creates a normal distribution centered at 0, with values above 0 representing an above average value for a given statistic and values below 0 representing a below average statistical value. This allows us to compare statistics across years, since we can look at how above or below average a team was for a certain statistic, rather than just looking at a single statistic.

Another data issue that we needed to deal with was some of our statistics were very highly correlated. For example, a team's win-loss record has a strong positive correlation to its average margin of victory, since teams that win a lot of games tend to also have a high average margin of victory. Seed and strength of schedule are also highly correlated, since teams who play more difficult schedules are usually rewarded with better seeds. To deal with these correlations, we used a process called principal component analysis. This process takes two highly correlated statistics, and uses a linear combination of the two to produce one single statistic that can be used in place of the two highly correlated statistics. We used this process to produce two sets of principal component statistics, one combining seed and strength of schedule, and another combining a team's win percentage with its average margin of victory. The formulas used to calculate these statistics can be seen in the Appendix.

## Generalized Linear Models

One method of predictive modeling that we used was a generalized linear model. A generalized linear model (GLM) is a modified version of a traditional linear regression, which takes the form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$

The response variable  $Y$  is expressed as a linear combination of the independent variables  $X_i$ , where  $i$  can be any number of independent variables. A traditional linear regression solves for the constants  $\beta_0, \dots, \beta_k$  such that the sum of the squared errors between the actual values of  $Y$  and the predicted values of  $Y$  is minimized.

A generalized linear model is similar to a traditional regression, but plugs the output  $Y$  into another function  $g(Y)$ , which is called the link function. The link function can take a variety of forms, depending on what the application of the GLM will be. The inverse of this link function is then used to turn the output  $Y$  into a predicted value. A log link function takes the form  $\ln(Y) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$ . The inverse of this function expresses the response variable  $Y$  as a function of the independent variables, taking the form  $Y = e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k}$ . This allows us to model situations where values can only be positive, since the exponential function can only produce positive results.

The link function that we chose was the logit function, whose inverse takes the form

$$Y = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k}}$$

which again allows us to express the response variable  $Y$  as a function of the independent variables. The main advantage of this function is that it produces an output between 0 and 1, which allows us to model the probability that a team will win a given game.

The statistical package R is particularly useful for the computation of generalized linear models. The built-in `glm()` function in R takes input data and evaluates the coefficients  $\beta_0, \dots, \beta_k$  using maximum likelihood estimation to make the model output best fit actual history. An example of the R code for computing GLMs and the resulting coefficient outputs can be seen in the Appendix.

Since our data was not originally in a form that could be used in the GLM, some manipulation was required to properly arrange the data. We started by taking the historical results from each of the eleven tournaments for which we had data, and finding the winner of each given game. We also calculated the difference between each of our collected statistics for each matchup and recorded these for use as independent variables in the GLM. For example, one matchup in 2016 was Kansas, a number 1 seed scoring 81.6 points per game, against Austin Peay, a number 16 seed scoring 76 points per game. Since Kansas won this game 105-79, the beginning of the data entry for this entry would be:

<u>Win</u>	<u>Seed</u>	<u>PPG</u>
1	-15	5.6

This type of calculation was done for every statistic that we collected for every game in each of the past eleven years. Win was treated as the response variable that our model was to predict, and the other statistics, such as Seed and PPG, were used as the independent, predictive variables. This was done both for the raw data that we collected and for the normalized data created to remove any inconsistencies arising between separate years of data.

This produced 2 data sets of 722 entries each, with each entry corresponding to one of the 722 tournament games played between 2006 and 2016. We were able to use these datasets along with the `glm()` function in R to compute the coefficients to be used to calculate the  $Y$  to be

plugged into the logit function, which would calculate the probability that a team would win a game given a set of independent variables.

We created many different GLMs using different combinations of the independent variables we had available. We created models using every statistic that we had available, and also created models using subsets of the data. For example, we created a model whose only inputs were the statistics that corresponded to a team's offense, and we also created a model using only defensive statistics. We also created models using only statistics that showed a degree of statistical significance in the model using every statistic. Models were also created that used only seed, wins, losses, average margin of victory, and strength of schedules, which our study of decision trees had shown to be the most powerful predictive variables. We also created models that replaced some of the most highly correlated variables with the principal component values that had been created to fix some of the multicollinearity problems that affected other models.

Once we had created these models, we had to decide which models produced the highest scoring brackets. These would then be the models that we would use to predict the 2017 tournament. To evaluate these methods, we created a workbook in Microsoft Excel that filled out a bracket based on the coefficients from the GLM. For a given game, Excel calculates the differences between the statistics for each of the two teams, multiplies them by the corresponding GLM coefficients, adds them together, and plugs this value into the logit function to produce the probability that each team will win that game. The team with the highest probability of victory is declared the winner, and advanced into the next round of the tournament. This process is used to project each game of the tournament, round by round, all the way through the national championship game for each year. These projected results are then compared to the actual results of the tournament, and points are awarded using the same scoring

method used by ESPN.com. This gives us a way to compare the projections of one GLM against another, with the most effective GLMs producing the highest scoring brackets.

While this method is a good way to compare one model to another, the models we were using have biases inherently built into them. Since we were using models containing data from 2006-2016 to project the tournaments from 2006-2016, our models already contain results from whichever year is being predicted, producing biased projections that result in higher scores. To remove this bias, we recomputed the GLMs by removing three years of data at random and then tested the new GLM on the three years that had been removed. Since the new GLM no longer contained data from the year it was projecting, this gave us a more accurate measure of the predictive power of our model. As expected, this method produced lower scoring brackets on average, since we had removed the upward bias that had originally been built in. This allowed us to more accurately choose the best model. However, for our prediction of the 2017 tournament, we used the models computed using all years of data, since we would have a larger dataset but none of the built in bias, since we did not yet know the 2017 results and could not possibly build them into our model.

We also developed a model that utilized a different GLM for each round of the tournament. This comes from the assumption that different characteristics about a team may have a different impact in each round. For example, seed may be very important in the early rounds of the tournament, but as the tournament progresses and nearly every team is highly seeded, seed may become a less powerful predictor. To model this, we ran the GLM and got a different set of coefficients for each of the Round of 64, Round of 32, Sweet 16, and for games from the Elite 8 on. We had to group all games from the Elite 8 on together to get a sample size large enough for the GLM to produce logical results. These results matched our original intuition, and showed

that as the tournament progresses seed becomes less important, while defense and ball control become more important.

We decided to make eight brackets using various GLMs. We made one bracket using all of the raw stats that we had available, and made another using all of the stats after they had been normalized. We made a bracket using the GLM with only the five stats that our decision trees analysis had shown to be most impactful, and also made a bracket using the GLM with the stats that we arbitrarily thought were most important. Another bracket was made using the GLM for the data using the principal component analysis stats in place of highly correlated statistics. We also made a bracket using the model that used a different GLM for each round of the tournament. We also included brackets made from using GLMs with only offensive stats or only defensive stats, even though our testing had shown that these did not have great predictive power.

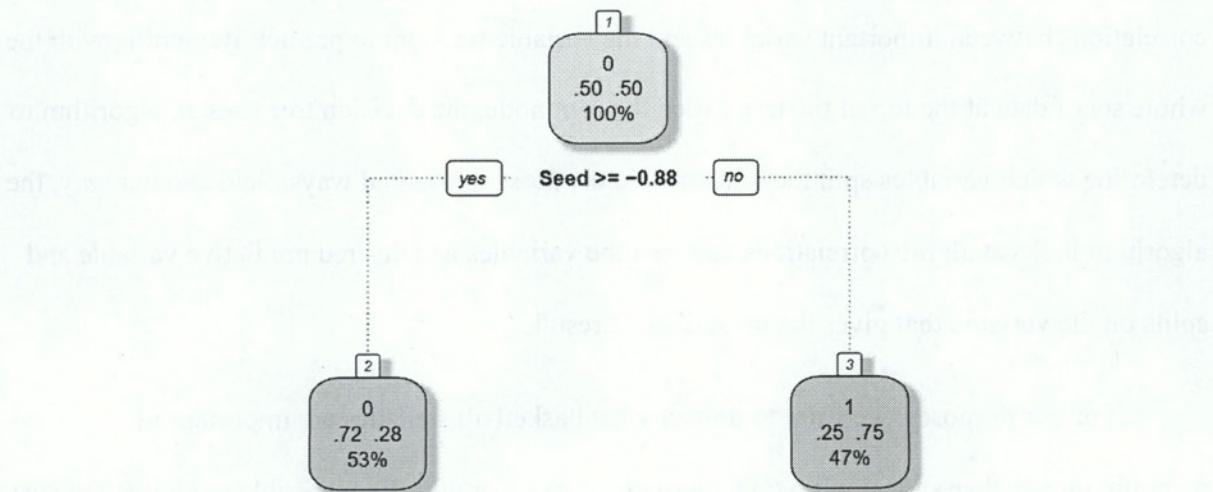
## **Decision Trees and Random Forests**

Decision trees are a popular form of machine learning that can be used in a plethora of ways. At a high level, they work by taking a set of data and determining multiple binary classification and/or regression subsets. These subsets give quick and accurate insight to correlations between important variables and the variable we want to predict. Beginning with the whole set of data at the top of the tree called the root node, the decision tree uses an algorithm to determine which variables split the outcomes in the most substantial ways. Said another way, the algorithm looks at all the correlations between the variables and desired predictive variable and splits on the variable that gives the most distinct result.

For our purposes, we want to predict what basketball statistics are important in determining whether a team will win a tournament game or not. We were able to look at the past

11 years of college basketball regular season statistics of tournament teams and compare them to how they performed in the respective tournament. The population has often championed certain statistics such as seed and points per game as important metrics for making their annual picks. How powerful and accurate are these predictors? Are there other, less obvious statistics flying under the radar that have an unexpected predictive power? Decision trees can help sort that out for us.

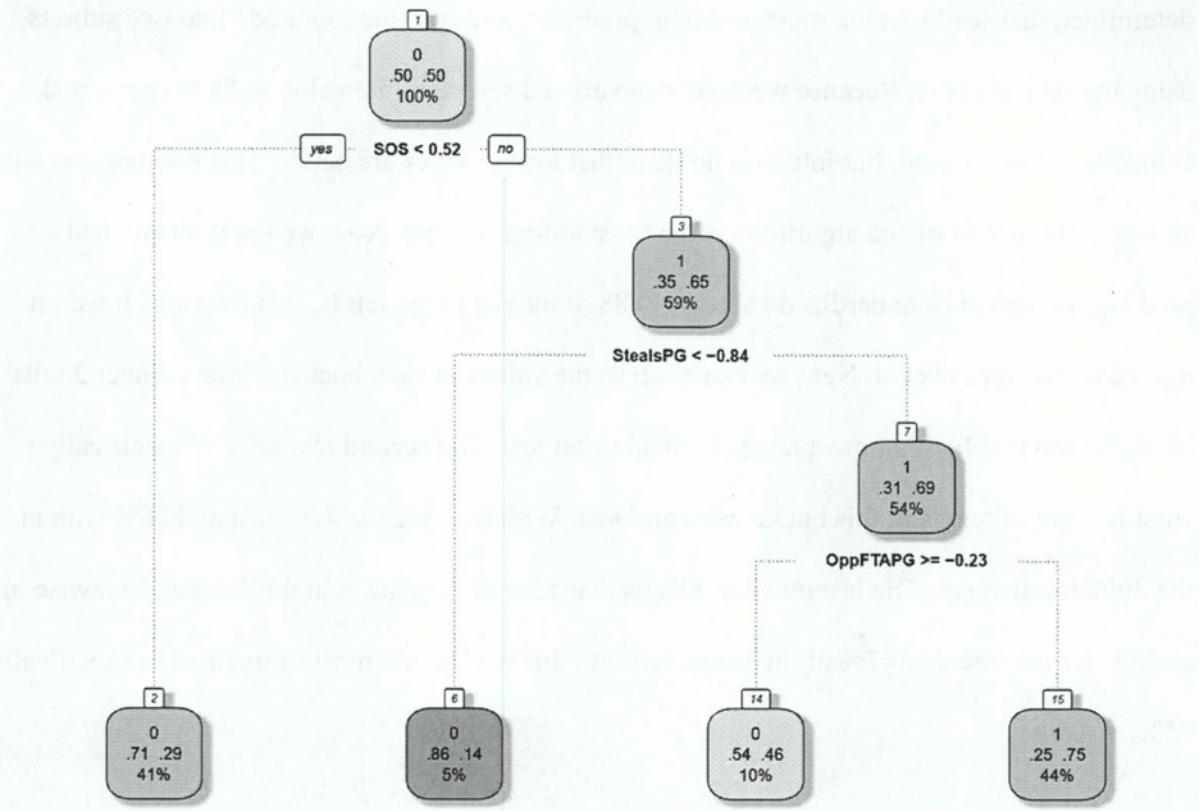
The ‘rpart’ package in R helps us make custom decision trees with any multivariate data set. Below is a simple example of a decision tree calculated from the 2009 tournament. Again, we start with the root node at the top which contains all the 2009 tournament games and results along with each team’s statistics for that tournament. We are looking to predict wins, so the algorithm uses that in our root node, denoting a loss as a ‘0’ and a win as a ‘1.’ What this root node tells us is first whether it’s a win or loss (it starts with 0), then it tells us the percent of teams that have losses (50%) and the percent that have wins (also 50%). This makes sense that wins and losses are evenly split in the root node because every game must have a winner and a loser. The last number tells us the percentage of the data that is in that bucket, with all the data (100%) being in the root node. The algorithm then compared all the relevant statistics,



determined that seed was the most powerful predictor, and split the root node into two subsets using the seed statistic. Because we used standardized statistics, the value -0.88 does not make complete sense for seed, but intuition holds in that lower values are better. This number was also strategically picked by the algorithm as the most telling number. Now we see if a team had a seed higher than the standardized value of -0.88, it moved to the left bucket. If it was lower, it moved to the right bucket. Next we can observe the values in each bucket. Node number 2 tells us by the top number that most teams in this bucket lost. The second row tells us specifically what percent of teams in this bucket won and lost. With that, we see 72% lost and 28% won in the 2009 tournament. The last number tells us that 53% of the data is in this bucket. Likewise in node 3, we can see that 47% of the teams fell into this bucket and most of them won; specifically 75%.

Further analysis and manipulation of different settings in the decision tree code can result in more complex trees like the one on the following page. This tree, taken from the 2016 tournament data, gives more in depth analysis of the relevant statistics and can give further predictive insight to these statistics.

There are two algorithms that we can utilize in the ‘rpart’ decision tree code depending on the data we are using and what we are trying to accomplish. Information gain or entropy is often used for classified variables and exploratory analysis while Gini impurity is better suited for continuous data and minimizing misclassification ([garysieling.com](http://garysieling.com)). Some say the two algorithms do not produce significantly different results, but since we are dealing almost exclusively with continuous data, we used the Gini impurity method.



How Gini impurity works is by measuring the disorder of a set of elements. This measurement “is calculated as the probability of mislabeling an element assuming that the element is randomly labeled according to the distribution of all the classes in the set” (*Magic Behind Constructing a Decision Tree*). The author of the aforementioned article also provides a good example of calculating this probability:

*Example: Suppose we have a set with 6 elements: {fred, red, blue, blue, blue, blue}. (The classes of this set are red and blue). We select an element at random. Then we randomly label it according to the distribution of classes in the set. This would be equivalent to labeling the selected element by rolling a 6 sided die with 4 blue sides and 2 red sides. The probability that we misclassify the element is equal to the probability that we select a red element times the probability that we label it blue plus the probability that we select a blue element times the probability that we label it red. This is  $2/6 * 4/6 + 4/6 * 2/6 = 16/36$ .*

An original Gini impurity is calculated for the root node. The larger the number of variables, the closer to 1 the beginning measure is. The goal of this algorithm is to minimize the average Gini impurity at each level. This is how the decision tree decides what variable to first split on, by choosing the split that minimizes the average Gini impurity the most.

The algorithm does this with each successive bucket until it reaches either a minimum bucket size or a maximum depth set by the user. It is important to not grow trees too deep, also known as overfitting, as being too specific can reduce the overall predictive power. For example, making dozens of splits on one tree will often result in ending buckets, or terminal nodes, with only 1 team. While this is very descriptive for a team that has that exact set of statistics, it is hard to apply this to other teams in the future. It is too unlikely that a future team will exactly mirror a past team's statistics. This is the issue with making models too specific. Often times simpler is better, but there is no hard and fast rule to knowing how deep to fit the data for a decision tree. Discretion should be used based on the size of the data, the number of variables, and the desired results.

Our approach to decision trees was to find the general trends in terms of the top statistics for each year, each round, and over the whole data combined. With the volatility of each tournament results, we wondered if different statistics held different weights in different seasons. We first started by including every statistic and seeing what the best split was for each year, round, and overall. After the top statistic was recorded, we eliminated it from the decision tree test to see what the algorithm valued second best. We repeated this until we found the top 10 statistics for each category. These results can be found in the appendix. Encouragingly, these results were mostly consistent with those found using the GLM.

One downside to decision trees is they do not incorporate any randomness in their predictions. With the various decision tree buckets we have been exploring, we find that different outlooks give different results. For example, while seed is often ranked very highly, the exact number where it splits differs from year to year. More specifically, the 2009 and 2011 tournaments both valued seed as the most predictive statistic, but 2011 set that split at 0.78, more than one standard deviation higher than 2009 split of -0.88.

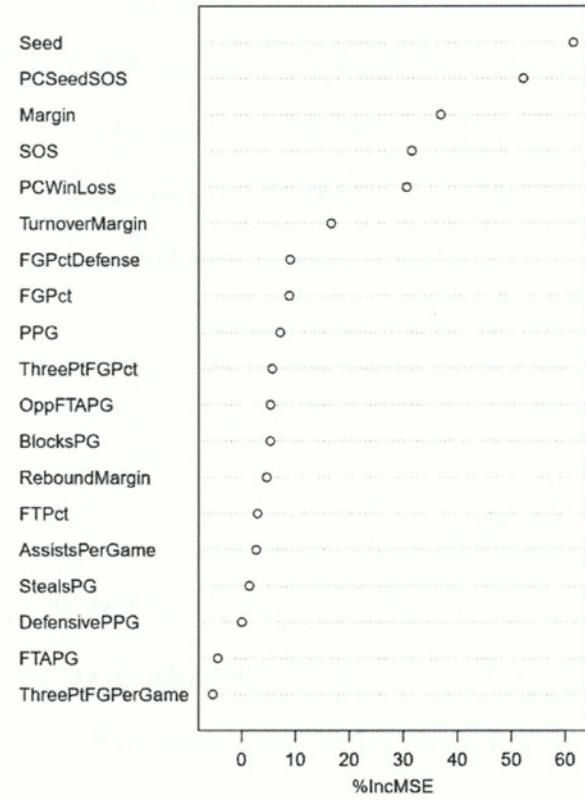
A way to incorporate these different measures is through ensembling an array of decision trees and averaging them. This ensemble creates what is called a random forest. An additional benefit of random forests is they can utilize the specificity of overfitting decision trees while negating the consequences. By default, random forests grow trees as far as possible and average them together. Because the algorithm for decision trees results in the same overfitted tree every time, random forests introduce a source of randomness. This is done by using various subsets of both the rows in the data as well as the variables. For our purposes, this helps us better incorporate the chances of an upset. While a 15 upsetting a 2 seed in the tournament is rare, it has happened a handful of times. Despite this, a model may never predict a 15 seed to beat a 2 seed in any decision tree based on statistics alone. This randomness introduces that chance.

The ‘randomforest’ package in R allows us to run such random forests and produce probability metrics for each individual team indicating their chance to win a particular game. The ‘randomforest’ package even allows us to determine how many trees we want to run. We ran 2,000 trees based on past data in different combinations and applied them to the 2017 data to give each team its probability to win each game.

After running the random forests, the output uses the removed subsets of data and tests them on the created model. One of these plots is found in figure 1 below. This plot measures the

increase in mean squared error (MSE) of the model if a designated variable was removed from a model. Like decision threes, this is another indicator to what statistics hold the most predictive power and even measures such power.

*Figure 1: This plot indicates the percentage that the mean squared error of the historical picks increases if the subsequent statistic is removed for the Round of 64. The higher the value, the more influential the statistic. The large drop off after the 5th statistic indicates that the statistics following only marginally improve the predictability, and even adding up many of those small errors does not noticeably increase predictability.*



Once these tests were completed, we set up various brackets generated by random forests. These brackets were determined using different combinations of statistics. Like the GLM, we created brackets incorporating all statistics tested as well as ones only using the general information statistics. This gave us general brackets and brackets generated by the consistently high ranking influence of the general information statistics. We also made brackets determined by the top 4 and top 6 statistics ranked by the ‘rpart’ decision trees and the top 4 and top 10 ranked statistics based on the MSE plots from the random forests. Using varying amounts of statistics can help us determine if there is any benefit to adding additional inputs.

Analyzing that not every round of the tournament values each statistic equally, we created two brackets for each statistical combination. One used the rankings of all our historical games in each round to determine a single probability for a team. We then applied this single probability to all rounds in 2017. In this case, each team had the same probability of winning a game, no matter what round they were potentially playing in. For reference, we will simply call them General Probability (GP) brackets. The other analyzed how different statistics ranked in each round and generated different probabilities for each team based on the round they would potentially be playing in. These will be referenced as Round By Round (RBR) brackets. These combinations of brackets gave a good spectrum from simplicity to complexity.

## **Results**

To track the success of our brackets, we created a group on ESPN.com. This scored the brackets for us, and also allowed us to compare the success of our brackets against the total of 18.8 million brackets that were entered on ESPN.com. ESPN provides statistics on where a bracket ranks out of the 18.8 million entered, and also lists what percentile a bracket is in. For example, a bracket in the 60<sup>th</sup> percentile has a higher score than 60% of the brackets entered on ESPN.

In total, our group was comprised of 24 brackets: 8 GLM brackets, 12 random forest brackets, 1 bracket from another computer-based prediction, and 3 control brackets. For one control, we picked the tournament based only on seed, which is the simplest method of filling out a bracket and also provided a benchmark to measure our predictive brackets against. The other two control brackets were the personal brackets that we filled out individually based on our own intuition.

After the first 16 games of the tournament on the first day, our brackets were doing very well. 5 of our 24 brackets predicted every game correctly, representing 20.8% of our group. For comparison, after the first day, only 0.8% of all brackets on ESPN were still perfect. We had a total of 20 out of 24 brackets miss two or fewer games, which was 83.3% of our group, compared with 26.3% of all brackets. 22 of our brackets were above the 50<sup>th</sup> percentile after day one. The average score of all brackets in our group ranked #7 out of the roughly 58,000 eligible groups that had been created on ESPN.com.

By the end of the first round, we did not have any perfect brackets remaining. However, 17 of our 24 brackets predicted at least 27 of the first 32 games correctly, which represented 70.8% of our group. Only 5.1% of all brackets on ESPN were this successful, showing that our predictive models were creating some value. At the end of the first round, 22 of our 24 brackets were above the 50<sup>th</sup> percentile, and our group ranked #10 out of all groups on ESPN.

The remaining rounds of the tournament were not as successful for our group. In the second round, Villanova was upset by Wisconsin, knocking out the team that 16 of our brackets had predicted to win the national championship. Despite this, after the second round 18 of our 24 brackets were still above the 50<sup>th</sup> percentile. However, our group fell to #3202 out of all groups on ESPN.

The biggest failure of our predictions was that none of our models predicted that eventual champion North Carolina would win the tournament. Our group's ranking suffered, as we fell behind many of the groups that included brackets picking North Carolina to win the tournament. At the end of the tournament, our group ranked as #26,000 out all eligible groups on ESPN.com. However, 18 of our 24 brackets were still above the 50<sup>th</sup> percentile.

Removing the 3 control brackets, 17 of our 21 predictive brackets finished above the 50<sup>th</sup> percentile. If the results of the tournament were totally random, there would be a 0.35% chance of this happening, which is roughly a 1 in 277 chance. This shows that our predictive models were effective and did add significant value to the process of filling out a bracket. The models were certainly better than our personal methods, as Tim's bracket finished tied for last in our group and Cody's bracket finished 22<sup>nd</sup>.

The results of the GLM brackets were very encouraging. Of the eight GLM brackets we entered in our group, seven finished above the 50<sup>th</sup> percentile. GLM brackets also made up four of the top five brackets in our group. The single highest scoring bracket in our group was also a GLM bracket. A copy of this bracket can be seen in the Appendix. The Appendix also contains a table showing the testing results for each of the GLM models and a description of each of the GLM formulas.

The most successful GLM bracket was the bracket created with a model using normalized data and only seed, win-loss record, average margin of victory, and strength of schedule. We were not surprised that this bracket performed well, since our analysis showed that these statistics were some of the most powerful predictors of tournament success. This bracket ended with a final score of 1170, which placed it in the 93<sup>rd</sup> percentile on ESPN.com.

One of the most surprising results from the GLM came from the brackets created using all available statistics. The bracket created using all available statistics and raw data finished 2<sup>nd</sup> in our group with a score of 1070, while the bracket created using all statistics and normalized data finished tied for 13<sup>th</sup> with a score of 730. We had expected the score for these two models to be very similar since they were using essentially the same inputs, but there was a large difference

between the two scores. This variance can probably be attributed to chance, since the models had scored very similarly in testing and only had a few different predictions for the 2017 tournament.

Another interesting result was the performance of the bracket using principal component statistics. We had utilized these statistics to try to eliminate some of the correlation between variables, and hoped that this would improve the model's predictive power. However, the bracket created with this model only finished 11<sup>th</sup> with a score of 770, which placed it right in the middle of the pack. This shows that even though principal component statistics may reduce the correlation between variables, they did not improve the efficiency of our models.

We were also not surprised by the two GLM brackets that had the lowest scores. These were the two brackets created using only offensive and defensive statistics, respectively. These methods had performed poorly in testing, but we felt that it was worth adding them to our group to see how they performed. The offense-only bracket finished 19<sup>th</sup> with a score of 660, while the defense-only bracket finished tied for 16<sup>th</sup> with a score of 670.

With 9 of the 12 brackets generated by random forests finishing in the top 50% of ESPN's rankings, random forests also proved to have predictive power. This set of brackets did particularly well in the first round. 5 of our 6 top brackets after the Round of 64 were created using random forests, with two brackets predicting the first 21 games correctly and 30 of the first 32 games. The latter two were RBR brackets (recall description on page 27) created using the top 4 and 6 statistics from the rpart decision tree rankings. Of the three other random forest brackets in the top 6, two of them also used the RBR method. On average, the RBR brackets picked 1.5 more games correctly than their GP counterparts.

Despite the strong start, the random forest brackets did not perform as well after the Round of 32. One of the potential downfalls to this set of brackets that may have contributed to

this was its tendency to pick the overall number 1 seed, Villanova, to win the championship. When Villanova fell in the Round of 32, this severely damaged this set's potential. Specifically, 2 of the top 6 brackets after the Round of 64 were also 2 of the 3 that did not finish the tournament in the top 50%. Combined, these particular brackets correctly predicted a mere 1 team of possible 8 to reach the Final Four. This round values correct picks at 80 points each opposed to the 10 points for each correct first round pick. This proves the importance of the later rounds. While picking so many of the first games correctly is a feat, correct picks in the later rounds are far more valuable.

The GP brackets on average had better overall scores than the RBR brackets. This is largely due to their success in predicting teams to reach the Elite 8. While these brackets only had an average of one correct pick more than their by round counterparts, these picks were worth 40 points apiece. This further proves the importance of predicting games correctly in the later rounds. However, it is much more difficult to predict the winners of later rounds, since there is no guarantee that a team will not be eliminated earlier in the tournament, such as Villanova was this year.

As for the specific models used, applying the top ranked statistics with the RBR method according to the MSE of the random forest placed 2nd and 3rd (both 73.2 percentile) among random forest brackets. Applying the GP method with MSE rankings proved to be significantly less successful; finishing in the middle of the pack. What's more, using the top 4 statistics in this measure versus the top 10 provided the exact same RBR and GP predictions. This shows adding additional statistics provided no further advantage or disadvantage.

Despite the findings above, another model that worked comparatively well was when all statistics were applied. Applying both RBR and GP methods, the models using all statistics

finished 4th (72.2 percentile) and 5th (61.7 percentile) respectively among random forest brackets. While these brackets were not among the top performers in the 1st round, they placed in the top 3 for the following 3 rounds. The scoring emphasis on the later games certainly helped these brackets hold firm through the later rounds.

As for the the model pairs that did not perform as well, the top 4 statistics according to the rpart decision tree rankings as well as using only the general information statistics did not prove fruitful. Three of these brackets finished below the 50th percentile, and the 4th easily could have. This bracket, calculated using only general information statistics, predicted only one Elite 8 team, Gonzaga, correctly. The fact that Gonzaga made it all the way to the national championship game aided this bracket's final standing. Had Gonzaga not made the championship, it too would have fallen below the 50th percentile. While these brackets performed well in the first round, correctly picking many of the upsets, they continued to pick the same upsets in the later rounds resulting in some very high seeds making deep runs. When these higher seeds bowed out early, these bracket placements severely plummeted.

## **Conclusion**

Obviously, none of our models were perfect. From the start, we never expected them to be perfect. This is one of the realities of predictive modeling; no matter how well constructed a model is, it will never be able to perfectly predict the future. However, it is clear that our methods were effective and added a significant amount of value to the process of filling out a bracket. From this standpoint, we view our project as a huge success. It also gave us a great opportunity to grow our skill and experience with predictive modeling. In the grander scheme of things, predictive modeling can help people make predictions not only for fun events such as

March Madness but also has applications in many other fields. It is truly proving to be an integral aspect in minimizing losses, forecasting disasters, and many other things. Overall, predictive modeling helps us understand how to best assist others in more efficient ways. The beauty of the NCAA tournament is that a new season starts next year, and we will refine and improve our models, so that we can try again, and be wrong again, next year.

## Appendix

### Principal Component Statistics Formulas

How to get Principal Component Values for Seed and SOS

1. Start with Raw Seed Data and Normalized SOS Data
2. Subtract 8.725784 from Seed and Divide by 4.6734983
3. Divide SOS by .9924682
4. ADD: (AdjSeed\*-.5218674+AdjSOS\*.5218674)
5. Multiply by .99931

How to get Principal Component Values for WinPct and Margin

1. Start with Normalized WinPct and Margin Values
2. Divide WinPct by .9924682
3. Divide Margin by .9924682
4. ADD: (AdjWinPct\*.5251833+AdjMargin\*.5251833)
5. Multiply by 1.00690143

### Sample GLM R code

```
# Set workspace
setwd("/Users/codykocher/Documents")

# Load excel package
library("xlsx")

# Load Data into R
DifferenceData = read.xlsx("GLMDATA.xlsx", sheetIndex = 3)

# Run GLM on all stats
mylogit <- glm(Win ~ Seed + Wins + Losses + PPG + Defensive.PPG +
  Margin + Rebound.Margin + Assists.Per.Game + Blocks.PG + Steals.PG +
  Turnover.Margin + FG.. + FG...Defense + X3.FG.Per.Game + X3.FG.. +
  FT.. + FTA.PG + Opp.FTA.PG + SOS, data = DifferenceData, family =
  "binomial")

# View GLM coefficients
summary(mylogit)
```

## Sample GLM coefficient output

Seed	-0.01283
Wins	0.3974
Losses	0.07022
PPG	-0.79645
Defensive PPG	0.71747
Margin	0.62926
Rebound Margin	0.27889
Assists Per Game	-0.24508
Blocks PG	0.05182
Steals PG	0.04011
Turnover Margin	0.29092
FG %	0.24139
FG % Defense	-0.02151
3 FG Per Game	0.15007
3 FG %	-0.11109
FT %	0.03268
FTA PG	-0.16168
Opp FTA PG	-0.06647
SOS	0.65594

## Sample Decision Tree Code

```
1 # Load the rpart libraries into R workspace
2 library(rpart)
3 library(rattle)
4 library(rpart.plot)
5 library(RColorBrewer)
6
7 #Sets the working directory to the desired location that the decision tree plot will be saved to
8 setwd("~/Users/Tim/Documents/2017 Spring/Thesis/Decision Trees/")
9
10 #For reference, here are the relevant statistical tree inputs for the decision tree:
11
12 # WinorLoss, Seed, PPG, DefensivePPG, Margin, WinPct, PCSeedSOS, PCWinpctMargin,
13 # ReboundMargin, AssistsPerGame, BlocksPG, StealsPG, TurnoverMargin,
14 # FGPct, FGpctDefense, ThreePtFGPerGame, ThreePtFGPct, FTPct, FTAPG,
15 # OppFTAPG, SOS, PCWinLoss, TournamentWins, TournamentLosses, PlayInWins, R64Wins,
16 # R32Wins, R16Wins, R8Wins, R4Wins, R2Wins
17
18 #Creates a decision tree splitting win or loss on the following statistics for the 2016 Tournament
19 fit <- rpart(WinorLoss ~ Seed + PPG + DefensivePPG + Margin + WinPct + PCSeedSOS + PCWinpctMargin +
20   ReboundMargin + AssistsPerGame + BlocksPG + StealsPG + TurnoverMargin +
21   FGPct + FGpctDefense + ThreePtFGPerGame + ThreePtFGPct + FTPct + FTAPG +
22   OppFTAPG + SOS + PCWinLoss,
23   data=tournament2016,
24   method="class")
25
26 #Plots the decision tree
27 fancyRpartPlot(fit)
```

## Sample Random Forest Code

```

1 #Load the randomForest library into R workspace
2 library(randomForest)
3
4 #Set working directory to desired destination for random forest prediction results
5 setwd("/Users/Tim/Documents/2017 Spring/Thesis/Decision Trees/Random Forests/2017 Predictions")
6
7 #Sets randomization seed for random forest algorithm
8 set.seed(444)
9
10 #Creates a random forest based on the desired data using the desired statistics and number of trees
11 fit <- randomForest(WinorLoss ~ Seed + SOS + PCWinpctMargin + OppFTAPG,
12                       data=allyears,
13                       importance=TRUE,
14                       ntree=2000)
15
16 #Fits the random forest
17 varImpPlot(fit)
18
19 #Determines predictive values for each team based on 2017 season data
20 Prediction <- predict(fit, tournament2017)
21
22 #Sets up data frame for writing predictions to Excel file
23 submit <- data.frame(TeamCode = tournament2017$TeamCode, Seed = tournament2017$Seed,
24                       winProbability = Prediction)
25
26 #writes predictions to Excel file
27 write.csv(submit, file = "allYears.csv", row.names = FALSE)

```

## Decision Tree Top Statistic Rankings

Rank	2006	2007	2008	2009	2010	2011
1	Seed	Seed	Seed	Seed	PCWinpctMargin	Seed
2	PCSeedSOS	PCSeedSOS	Margin	PCSeedSOS	Seed	OppFTAPG
3	Margin	PCWinpctMargin	PCSeedSOS	SOS	Margin	Margin
4	PCWinpctMargin	Margin	SOS	PCWinpctMargin	PCSeedSOS	PCSeedSOS
5	SOS	SOS	PCWinpctMargin	Margin	SOS	SOS
6	FGPct	BlocksPG	FGPctDefense	ReboundMargin	ReboundMargin	PPG
7	BlocksPG	FGPctDefense	TurnoverMargin	FTAPG	FGPctDefense	WinPct
8	OppFTAPG	DefensivePPG	ReboundMargin	AssistsPerGame	FTPct	PCWinpctMargin
9	ReboundMargin	FGPct	DefensivePPG	PPG	PPG	TurnoverMargin
10	PPG	ThreePtFGPct	BlocksPG	WinPct	WinPct	FGPct

Rank	2012	2013	2014	2015	2016	All Years
1	PCSeedSOS	PCSeedSOS	PCSeedSOS	Seed	SOS	PCSeedSOS
2	SOS	SOS	SOS	PCSeedSOS	PCSeedSOS	Seed
3	FGPctDefense	Seed	Seed	SOS	OppFTAPG	SOS
4	Seed	Margin	Margin	Margin	Seed	PCWinpctMargin
5	Margin	PPG	PCWinpctMargin	PCWinpctMargin	FTPct	Margin
6	BlocksPG	WinPct	DefensivePPG	FGPct	FTAPG	OppFTAPG
7	DefensivePPG	PCWinpctMargin	ReboundMargin	PPG	PCWinpctMargin	ReboundMargin
8	OppFTAPG	FGPct	FGPct	WinPct	FGPct	FGPctDefense
9	PCWinpctMargin	TurnoverMargin	OppFTAPG	ThreePtFGPerGame	StealsPG	FGPct
10	ReboundMargin	StealsPG	ThreePtFGPct	FGPctDefense	Margin	DefensivePPG

Rank	Round of 64	Round of 32	Round of 16	Round of 8	Round of 4	Round of 2
1	PCSeedSOS	PCSeedSOS	Seed	DefensivePPG	Wins	Margin
2	Seed	Seed	PCSeedSOS	PPG	PPG	ThreePtFGPct
3	SOS	SOS	Wins	StealsPG	FTPct	PCWinLoss
4	PCWinLoss	Wins	SOS	FGPct	Margin	FGPctDefense
5	Margin	PCWinLoss	Margin	FTPct	PCWinLoss	StealsPG
6	Wins	Margin	PCWinLoss	Margin	Losses	FTPct
7	Losses	Losses	Losses	SOS	BlocksPG	BlocksPG
8	OppFTAPG	FGPct	DefensivePPG	ThreePtFGPct	AssistsPerGame	FGPct
9	PPG	FGPctDefense	BlocksPG	PCSeedSOS	PCSeedSOS	SOS
10	TurnoverMargin	PPG	ReboundMargin	Seed	FGPct	PCSeedSOS

Normalized Some 6 Bracket



## GLM Projection Results

38

	Score by Year														Average	Rank			
	All-All	All-Some 1	All-Some 2	All-Some 3	All-Some 4	All-Some 5	All-Some 6	All-Some 7	All-PC	All-All	All-Some 1	All-Some 2	All-Some 3	All-Some 4	All-Some 5	All-Some 6	All-Some 7		
2016	1010	680	750	390	600	810	820	1000	1030	1020	680	750	380	320	810	820	750	742.4	8
2015	1140	1150	1050	750	720	1060	1100	1140	1160	1140	1150	1050	770	720	1100	1100	950	1014.7	4
2014	670	610	680	600	480	690	690	670	670	670	590	670	590	500	690	650	670	634.7	10
2013	840	900	830	490	720	810	810	830	840	820	900	820	520	710	810	810	870	784.1	7
2012	1290	940	1290	1090	850	1240	1320	1290	1250	1280	1260	1290	1080	890	1250	1320	1320	1191.2	1
2011	510	530	490	470	500	500	510	500	480	440	530	490	470	520	480	510	500	495.9	11
2010	810	1100	940	1050	980	890	830	1160	1140	1140	1080	920	1070	950	890	830	1160	996.5	5
2009	710	1160	810	530	910	960	1270	700	880	740	1200	790	550	950	810	1250	710	878.2	6
2008	1050	1450	900	930	1150	970	1450	1050	1170	1130	1440	900	1690	1370	970	1450	1070	1184.7	2
2007	1240	960	1050	1240	740	1260	1100	1250	1340	1240	1140	1180	800	850	980	1030	1110	1088.8	3
2006	900	660	640	750	650	700	710	860	660	620	660	620	750	710	720	710	620	702.4	9
Average	924.5	921.8	857.3	753.6	754.5	899.1	964.5	950.0	965.5	930.9	966.4	861.8	788.2	771.8	864.5	952.7	884.5		
Rank	7	8	13	17	16	9	3	5	2	6	1	12	14	15	11	4	10		

**Key**

Red background indicates models using raw data

**First Word**

All Model using all years of available data (2006-2016)

**Second Word**

All	Model using all available statistics
Some 1	Model using only statistics that we felt were important
Some 2	Model using only statistics that were statistically significant in All-All
Some 3	Model using only offensive statistics
Some 4	Model using only defensive statistics
Some 5	Model using only Seed, Wins, Losses, and SOS
Some 6	Model using only Seed, Wins, Losses, Margin, and SOS
Some 7	Model using only statistics whose coefficients were greater than 0.1 in All-All
PC	Model using all statistics with Principal Component statistics replacing Seed, SOS, Wins, Losses, Margin

## **Works Cited**

- B. (2014, June 2). Magic Behind Constructing a Decision Tree. Retrieved from  
<https://gormanalysis.com/magic-behind-constructing-a-decision-tree/>
- B. (2014, August 31). Decision Trees in R using rpart. Retrieved from  
<https://gormanalysis.com/decision-trees-in-r-using-rpart/>
- Boice, J. (2016, October 20). How Our 2015-16 NBA Predictions Work. Retrieved from  
<https://fivethirtyeight.com/features/how-our-2015-16-nba-predictions-work/>
- CBS Sports. (2006). *2006 Strength of Schedule*. Retrieved from  
<http://www.cbssports.com/collegebasketball/rankings/sos>.
- Derrig, R., Frees, E., Meyers, G. (2014). *Predictive modeling applications in actuarial science*. New York: Cambridge University Press.
- Ezekowitz, J. (2011). Quantifying intangibles: a new way to predict the NCAA tournament.  
*Harvard Sports Analytics Collective*. Retrieved from  
<https://harvardsportsanalysis.wordpress.com/2011/05/18/quantifying-intangibles-a-network-analysis-prediction-model-for-the-ncaa-tournament/>.
- Geiling, N. (2014, March 20). When Did Filling Out A March Madness Bracket Become Popular? Retrieved from <http://www.smithsonianmag.com/history/when-did-filling-out-march-madness-bracket-become-popular-180950162/>
- Hart, J. (2014, March 23). One perfect NCAA bracket remains in Yahoo Tourney Pick'em. Retrieved from [http://sports.yahoo.com/blogs/ncaab-the-dagger/one-perfect-ncaa-bracket-remains--with-a-catch-053219703.html;\\_ylt=AwrBT8oug.9YCM4Ape1XNyoA;\\_ylu=X3oDMTEyamNna3QxBGNvbG8DYmYxBHBvcwMyBHZ0aWQDQjM2MTFfMQRzZWMDc3I-](http://sports.yahoo.com/blogs/ncaab-the-dagger/one-perfect-ncaa-bracket-remains--with-a-catch-053219703.html;_ylt=AwrBT8oug.9YCM4Ape1XNyoA;_ylu=X3oDMTEyamNna3QxBGNvbG8DYmYxBHBvcwMyBHZ0aWQDQjM2MTFfMQRzZWMDc3I-)

- Klein, A. and Stekler, H.O. (2011). Predicting the outcomes of NCAA championship basketball games. *Research Program on Forecasting*. Retrieved from <https://www2.gwu.edu/~forcpgm/2011-003.pdf>.
- McCord, B. (2017, March 27). Ranking The Final Four Teams. Retrieved from <http://www.sportsgrid.com/real-sports/ncaa-basketball/ranking-the-final-four-teams/>
- National Collegiate Athletic Association. (2017). *Archived Team Statistics*. Available from [http://stats.ncaa.org/team/inst\\_team\\_list?sport\\_code=MBB&division=1](http://stats.ncaa.org/team/inst_team_list?sport_code=MBB&division=1).
- Phillips, K. (2014, March 17). Registration For Warren Buffett's \$1 Billion Basketball Challenge Opens Today. Retrieved from <https://www.forbes.com/sites/kellyphilipserb/2014/03/03/registration-for-warren-buffetts-1-billion-basketball-challenge-opens-today/#451c85c271bd>
- Ruyle, M. (2015, April 01). JQAS Highlights: Prediction Methods for the NCAA Men's Basketball Tournament. Retrieved from <http://magazine.amstat.org/blog/2015/03/01/jqas-highlights-prediction-methods-for-the-ncaa-mens-basketball-tournament/>
- Sieling, G. (2014, March 02). Decision Trees: "Gini" vs. "Entropy" criteria. Retrieved from <https://www.garysieling.com/blog/sklearn-gini-vs-entropy-criteria>
- Silver, N. (2014, March 17). Building a Bracket Is Hard This Year, But We'll Help You Play the Odds. Retrieved from <https://fivethirtyeight.com/features/nate-silvers-ncaa-basketball-predictions/>
- SportsCenter. (2017, March 27). @SportsCenter. Retrieved from <https://twitter.com/SportsCenter/status/846153308989468674>
- Team Rankings. (2017). *Team Statistics*. Available from <https://www.teamrankings.com/ncb/team-stats/>.