Ball State Undergraduate Mathematics Exchange
http://www.bsu.edu/libraries/beneficencepress/mathexchange
Vol. 12, No. 1 (Fall 2018)
Pages 15 – 23

Predictive Model for the NCAA Men's Basketball Tournament

Cody Kocher and Tim Hoblin

Cody Kocher graduated in 2017 with a degree in Actuarial Science from Ball State University. He now works as an actuarial analyst for Nyhart in Indianapolis.

Tim Hoblin graduated in 2017 with a degree in Actuarial Science from Ball State University. He now works as an actuarial analyst for Allstate in Chicago.

1 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 ever been documented. Our goal was 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.

Sixty-eight 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 champion. 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.

2 What Others Have Done

Attempting to predict the NCAA Tournament is not a brand-new phenomenon. There are many algorithms that have been made public over the years. From well-known statisticians, to machine learning competitions on kaggle.com, many have tried their hand at using data to aid their bracket predictions. Silver's model [6] is based on a composite of 8 general equally-weighted team ratings (6 computer rankings and 2 human rankings) through 2017. Each computer rating is based on very similar statistics such as wins and losses, strength of schedule, margin of victory, and offensive and defensive efficiency. These statistics are all computed from performance throughout the season.

Another previous study on this topic was written by Ezekowitz [3] 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. 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.

Simple models often see more success in general predictive modeling applications, and the NCAA Tournament is no exception. In 2014 Kaggle.com, a site that hosts a variety of predictive modeling competitions, held a competition called "March Machine Learning Mania," a contest for predicting the respective NCAA Tournament. Many models were entered using a wide array of mathematical techniques.

3 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 stats.ncaa.org [5], 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 [7]. Strength of schedule from 2006 was collected from cbssports.com [1].

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 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 percentages to fall by almost two percent. We cannot directly compare values from before and after this rule change because 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 from cbssports.com 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 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 that some of our statis-

tics were highly correlated. For example, a team's win-loss record has a strong positive correlation to its average margin of victory because 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 because 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. In our model, principal component analysis was used to replace two correlated statistics by a linear combination of the two to produce one single statistic. 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.

4 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 traditional linear regression, which takes the form

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k.$$

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

A GLM is similar to traditional regression, but the GLM extends the modeling capabilities. On the left-hand side of the equation the output Y is replaced by a function $g(Y) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$. The function $g(\cdot)$ 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 linear combination of predictors X_i 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 because 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 allows us to express the response variable Y as a function of the independent variables [2]. 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 software R is particularly useful for the computation of generalized linear models. The built-in glm() function in R takes input data and computes the coefficients $\beta_0, \beta_1, \ldots, \beta_k$ using maximum likelihood estimation to make the model output best fit actual history. For a GLM, maximum likeli-

hood estimation is used to compute the coefficients instead of the minimization of the sum of squared errors technique used by traditional linear regression.

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 this data entry would be:

Win	Seed	PPG
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 two 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 data sets 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. This would calculate the probability that a team would win a game given a set of independent variables.

We made eight brackets using various GLMs. We made one bracket using all of the raw statistics that we had available and made another using all of the statistics after they had been normalized. We made a bracket using the GLM with only the five statistics that our decision trees analyses (described in the next section) had shown to be most impactful, and also made a bracket using the GLM with the statistics that we arbitrarily thought were most important. Another bracket was made using the GLM for the data using the principal component analysis statistics in place of highly correlated statistics. We also made a bracket using a different GLM for each round of the tournament because some statistics may have had more predictive power in some rounds than in others. We also included brackets made from using GLMs with only offensive statistics or only defensive statistics, even though our analysis had shown that these did not have great predictive power.

5 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 into correlations between predictive

variables and the response 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 response variable and predictive variables and splits on the variable that gives the most distinct result.

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. 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 algorithm rather than the information gain algorithm.

The Gini impurity works 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" [4]. The author of the aforementioned article also provides a good example of calculating this probability:

Example 1. *Suppose we have a set with 6 elements: red, 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.

One downside to decision trees is that they do not incorporate any randomness in their predictions. With the various decision tree buckets that we explored, we found that different outlooks gave different results. For example, while seed is often ranked very highly from year to year, the exact number where it splits differs from year to year. A way to incorporate these different measures is through ensembling an array of decision trees and averaging them to obtain what is formally called a *random forest*. An additional benefit of random forests is that 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.

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. After running the random forests, the output uses the removed subsets of data and tests them on the created model. This plot

measures the increase in mean squared error (MSE) of the model if a designated variable were removed from a model. This is essentially another predictor to what statistics hold the most predictive power and measures such power.

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 ranked statistics from our tests. 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 of each bracket. One used the rankings of all our historical games in each round to determine a single probability for a team that we then applied 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. 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 combinations of brackets gave a good spectrum from simplicity to complexity.

6 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 all those entered and also lists what percentile a bracket is in. For example, a bracket in the 60th 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 which 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 intuitions.

After the first 16 games of the tournament on the first day, our brackets were doing very well. Five 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. Twenty-two of our brackets were above the 50th 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 50th 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 50th 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 50th percentile.

Removing the 3 control brackets, 17 of our 21 predictive brackets finished above the 50th percentile. If the results of the tournament were totally random, there would be a 0.35% chance of this happening, which translates to a roughly 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 22nd.

7 Conclusion

Obviously, none of our models were perfect and from the start we never expected them to be. 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 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 it also has applications in many other fields. It is 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.

Bibliography

[1] CBS Sports. (2017). *2006 Strength of Schedule*. Retrieved from `http://www.cbssports.com/collegebasketball/rankings/sos`.

[2] Derrig, R., Frees, E., Meyers. G. (2014). *Predictive modeling applications in actuarial science.* New York: Cambridge University Press.

[3] Ezekowitz, J. (2011, May 18). 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/.

[4] Gorman, B. (2014, June 2). *Magic Behind Constructing a Decision Tree.* Retrieved from https://gormanalysis.com/magic-behind-constructing-a-decision-tree/.

[5] National Collegiate Athletic Association. (2017). *Archived Team Statistics.* Available from http://stats.ncaa.org/team/inst_team_list?sport_code=MBB&division=1.

[6] 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/.

[7] Team Rankings. (2017). *Team Statistics.* Available from https://www.teamrankings.com/ncb/team-stats/.