Predicting March Madness

A Player Based Approach

Darren Lund

Brigham Young University
darrenlund@byu.net

April 18, 2018

Abstract

The NCAA March Madness Tournament provides a unique opportunity for sports analytics. Every year, millions of fans try to predict who will win each and every game, with no one managing to achieve a perfect bracket. Due to the vast amount of data collected in sporting events, the task for achieving the perfect bracket is well-suited for prediction algorithms. While many different methods have already been implemented, this project outlines a player based approach to predict some of the more nuanced factors of basketball, and thus more accurately predict upsets and appropriately named cinderella teams that bust brackets year after year. Data for this project was scraped from https://kenpom.com.

Problem Statement

The art of predicting the outcome of a competition is at least as old as the ancient Roman Empire. Just as contests and competitions of skill have intrigued humankind, people have wanted to determine beforehand who the victor would be. One of the biggest instances of this in our day and age is the NCAA March Madness (Men's College Basketball) Tournament. A total of 68 teams coming together in a single elimination tournament to determine who will come out on top...and millions of fans predicting who will win each game.

The goal of the NCAA bracket competition, like so many others, is to score more points than others. Points are scored by correctly guessing which of two teams, paired against each other, will win in a single game at a neutral location. Games further along in the tournament are worth more than the initial games, with each successive round worth double the points of the previous round. For example, in the round of 64 (where there are 64 teams competing), games are worth 10 points. In the round of 32, games are worth 20 points, the sweet 16 games are worth 40, elite 8 are worth 80, final four worth 160, and the championship game worth a total of 320 points. The goal then, is to choose teams to maximize the bracket's total score by maximizing the weighted number of correctly chosen games. This project will seek to accomplish that by correctly predicting each game independent of where it is in the tournament, in the hopes that this will increase the total number of correctly predicted games. Thus, instead of just optimizing the bracket's score, this project seeks to also have the most number of winners correctly predicted. There are two principal ways that it will attempt to do this. The first is through some typical classification algorithms discussed in the methods section using the data organized in a fundamentally different way than this information is typically used. The

second is through a method developed specifically for this task by the author. It will do this by predicting the final score of each game individually, round by round, through estimating the number of points scored by each player individually, and thus determining the winner. Testing data was originally used on the 2017 tournament in order to pick preferred parameters, but due to recent finish of the 2018 tournament, analysis will be done on it's results in the 2018 tournament (with the 2017 data included in training).

Previous Studies

Many previous algorithms have been used to try and solve this problem. The most common is a pagerank algorithm, similar to Google's algorithm for search results. Other methods worth mentioning are the Logistic Regression Markov Chain (LRMC) model, and the Massey Ratings, developed by Kenneth Massey, an assistant professor of mathematics at Carson-Newman University in Tennessee. These methods primarily focus on ranking teams based on a few factors, namely when the game was played, where the game was played, and what the score was. This project seeks to do things a little differently. Namely, by using team statistics (and a lot of them) to predict the winners of particular games directly, instead of ranking the teams first. While less common, the hope is that by using team statistics, the algorithm will be able to better identify potential nuances that contribute to cinderella teams (ones that get much further than predicted, such as Loyola-Chicago in 2018) and giant slayers (low ranked teams that beat a highly ranked team, like UMBC beating Virginia in 2018). While many attempts have been made to better understand and predict these events, no algorithm has consistently identified them.

Data

All data was scraped from kenpom.com, a website managed by Ken Pomeroy. The scraping process involved using Selenium to crawl through the website to find player and team data for the 68 competing teams for the years 2013-2018, then parsing the html through BeautifulSoup to save the tables that contained the data. The html tables were then transformed into panda dataframes, which were cleaned, then saved as csv files. Since Ken keeps the data stored so well, there was no missing data to infer, merely transforming it into a usable format throughout. This process involved taking every player's seasonal statistics (such as points scored, offense rating, steals, personal fouls...) and averaging them into a single vector (in the same format as is stored for a single game). Then for each team, all their player's data was averaged together into a single vector of the same form, so that each team was represented as an average of all player's stats. These stats were then appended to the offensive and defensive team stats from the scouting report, creating one vector with 68 entries. Ken has collected college basketball data for years, is independent of all teams in the NCAA, and readily gave me permission to use his data for this project when emailed. As such, there is no reason for any of the data to be biased, one way or another. His data is used by many prominent basketball analysts, including Tim Chartier. Since the purpose of this project is to predict the results of the NCAA tournament (and thus get a perfect bracket) as opposed to predict who will end up participating in the tournament, data was gathered on the 68

teams per year that have participated in the tournament since 2013. This data includes individual player statistics for each player of these teams on a game-by-game basis, as well as a yearly team scouting report, which contains data for how the team works as a unit. The tournament itself was stored as a dictionary. The keys were the number of teams in a given round, and the values were the teams that made it to that round.

Methods

Regular Classification Methods

There were five different "out of the box" classifiers that were used to try to predict the games. These were the PageRank algorithm, Gaussian naive-bayes (GNB), logistic regression (LR), support vector machine (SVM) and random forest (RF) classifiers. Other classifiers, like XGBoost weren't considered; in xgboost's case, this was because the number of data points collected was deemed insufficient to gain much advantage on just using the RF classifier, especially considering the increase in training time. Due to the fact that the data is in $\mathbb{R}^{68}$, the SVM classifier was considered a good option, particularly for reducing the dimension. For all of these prediction algorithms, the data was reorganized into a new form. This was done by storing the each game in the training data as the difference between the "first" team and the "second" team. The result was then classified as a one (1) if the "first" team won, and a zero (0) if the "second" team won. This was done with the averaged player statistics (across all season games and all players), the scouting report, and a combination of the two. Thus each team was represented by a single vector, which then could be subtracted from any other team for any given match up. A general Gridsearch was used to determine the parameters that performed best for the LR, SVM, and RF classifiers, as well as each combination of data (strictly player, strictly team, or combination of both). Not surprisingly, the best results came from using the combined player and team data. The parameters that were used for the classifiers on the 2018 data were as follows:

Classifier	Parameters
LR	$C = 0.1$
SVM	ker=linear
RF	$max\_dep = 5$, $num\_est = 40$

While for the RF classifier these parameters were the best when tested on the 2017 data, the others were chosen after the 2018 data had been determined and tested. This is because one interesting finding of this project is that in different years, different parameters do better than others. This is discussed in the results and analysis section.

My Classification Method

My classifier (MYC) went about things in a rather unique way. It tried to select which players will contribute the most to the tournament, and then divide scoring opportunities among them.

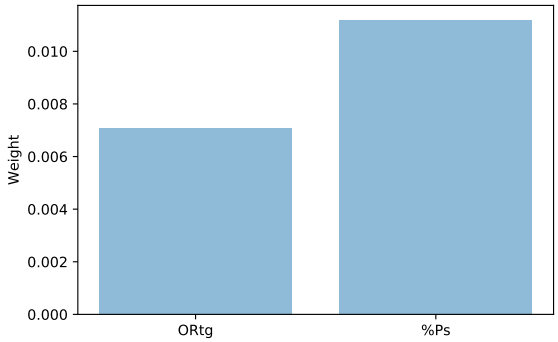
Player Selection

The first task is to classify players into one of two categories: significant contributors to the tournament, and non-significant contributors. This is because each team has more players than they usually will have play in the tournament, so we need only worry about players who actually will contribute to the tournament games. Whether or not someone was a significant contributor in any given game was determined by the percentage of possessions used during that

game. If you were part of 10% or less of total possessions, then you were not considered a significant contributor. Anything greater was considered a significant contribution to the game. The reason for 10% is because there are multiple possible reasons a player could have some participation, but not actually be a significant contributor. For instance, if a game is going so well that the coach has no fear of losing, that coach may choose to put in players who, up to that point, had contributed nothing to the game so that they can have the experience of playing in the tournament. Alternatively, it's possible that a player started playing in the game, but before they managed to do very much became injured, and had to leave the game. Since there are 5 players on the court at a time, the 10% threshold requires that, on average, you made some contribution to the game every other possession.

Obviously, some of the statistics gathered have little to no bearing on whether or not a player will play in the tournament. For instance, no coach is going to keep you out of the game because you had 0 blocks, especially if you score 20 points on average per game. To determine which statistics were important and which were not, SKLearn's Lasso method was used on the averaged season statistics. The Lasso method utilizes L1 logistical regression for feature selection. Essentially, this amounts to weighting each statistic by how important it is in determining whether or not they will be significant contributors in the tournament while trying to keep the total weights as small as possible. Nonzero weights correspond to features that were important for classification, and zero coefficients those that could be ignored. The only variables with nonzero coefficients were Offensive Rating (ORtg) and Percent of Possessions Used (%Ps), as shown in Figure 1. Utilizing these two variables alone achieved 87.89% accuracy [1] in determining which players would contribute to the tournament and which would not. If, for one reason or another, the algorithm predicted that fewer than five players would contribute, everyone was used instead, since you can't have less than five players for a game.

Figure 1: *Variable weights from Lasso Method (weights of 0 excluded)*



Scoring

Once the players have been determined, the next step is to predict how much each team will score. This is done by 1) approximating the number of scoring possessions each team will get; 2) removing the predicted number of scoring possessions lost due to turnovers; 3) using that number of scoring possessions and rebound probabilities to determine the number of scoring opportunities each team will receive. Those scoring opportunities are then divided among two point and three point attempts, with free-throw attempts calculated separately (due to the fact that free-throws can come from scoring attempts or non-shooting fouls if the team is in the bonus). These percentages were taken

[1] Data from 2013-2016 was used for training with 2017 data used for testing.

directly from the teams scouting report. Those shots are then distributed to the players most likely to shoot them, and the player's data is used to determine how many they will make. The distribution was determined by averaging the percentage of the team's two-point, three-point, and free throw shots that each player took over the season. Thus, if a certain player, on average, shot 20% of the teams three-point shots, then 20% of the number of three-point shots calculated were assigned to that player. Then say that that player had a 50% average for three-point attempts. Then 10% of the teams three-point shots are assigned as "made," contributing 3 points to the team's total score each. This was done for all combinations of shot type and player. After totaling the points from made baskets, the team with the most points is predicted to be the winner. Ties went to the team that was ranked with the higher seed in the tournament. During this process, all number of possessions, attempts, and made shots are rounded to integers to prevent illogical calculations (such as Nigel-Williams Goss of Gonzaga making 6.024 of his 7.593 three point shots).

Results

The methods, in general, performed alright. However, as stated earlier, the optimal parameters for 2018 turned out to be different from those for 2017. For instance, in the 2017 data, the optimal parameters were:

Classifier	Parameters
LR	$C = 1e-8$
SVM	ker=sigmoid
RF	$max\_dep = 5$, $num\_est = 40$

While the reasons for this disparity are not explored here, it raises the question of whether this year was anomalous, or if every year you can expect different parameters to perform better. The actual evaluation of this, however, is a bit beyond the current purposes of this project. As such, the following results are from using the optimum parameters on the 2018 dataset. The RF classifier, due to the random nature of the classifier performs differently every time it is run. Rarely, it does better than every other classifier. Also rarely, it does far worse. On average, it seems to perform about on par with the PageRank classifier. The LR classifier consistently scores the best (with the exception of the RF classifier on few select occasions) with 850 points. The SVM classifier came just above that, with 860 points. Third was the standard PageRank algorithm with 620 points, followed closely by the RF classifier with a total of 600 points. In the basement are the GNB (510 points) and MYC (360 points) classifiers. The cumulative scores, by round, are displayed in Figure 2. The random line shows the expected score from randomly guessing, as opposed to any one random bracket. For reference, the following

Figure 2: *Total Score Through Time : The cumulative score for each algorithm after every round has been finished.*

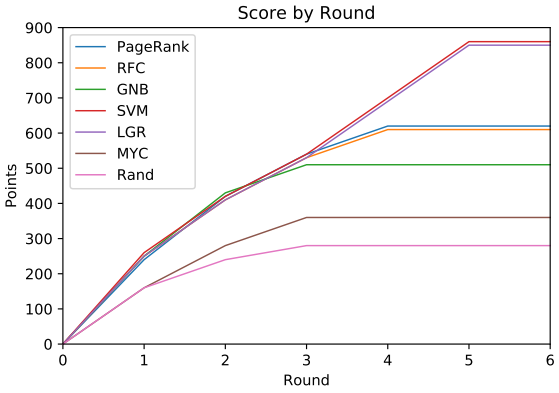


table contains which team each classifier predicted to win the entire tournament, and when that team was eliminated.

Class.	Champion	Round Eliminated
PgRnk	Duke	Elite 8
RFC	UNC	R 32
GNB	Duke	Elite 8
SVM	Virginia	R 64
LGR	Virginia	R 64
MYC	Duke	Elite 8

Analysis

Before beginning analysis on the classifiers in general, it is important to mention that this year was a statistical anomaly as far as the NCAA Tournament is concerned. To illustrate this, since 1985 (the year the tournament became what it is today), there have only been four years that a 12 seed team has not beaten a 5 seed team: 1988, 2000, 2007, and this year, 2018. Second, since 1985, there have only been four teams that were ranked as an 11 seed that made it to the final four: LSU in 1986, George Mason in 2006, VCU in 2011, and Loyola-Chicago in 2018. Finally, and also since 1985, there has only been one instance of a 16 seed team beating a 1 seed team: this year, UMCB upset Virginia. Not only was Virginia the national favorite to win the tournament (most people in the U.S. picked them to win the championship) but they were also highly ranked by every other ranking algorithm researched. All three of these individually anomalous things happened this year. This helps illustrate the difficulty of trying to correctly predict the NCAA tournament. There will always be an element of randomness that cannot be predicted, regardless of the techniques used.

Of the "out of the box" classifiers used, the ones that performed the best the most consistently are the LR classifier and the SVM classifier. When tested on the 2017 data, they achieved an accuracy of about 72.6%, which is about what other models have achieved. In the 2018 tournament, they finished with 850 and 860 points respectively, which would place them above the 80^{th} percentile in the nation. While the RF classifier can outperform them, it's success is varied and random, and thus much less reliable. This was easy enough to determine from the 2017 data testing, as on different runs, the parameters that gave the best results changed on practically every single iteration. Thus, the choice of picking the best parameters alone is a guess that can greatly effect the success of the RF model. In fact, it sometimes does just as poorly as the GNB classifier, which, of the regular classifiers, is by far the worst. This was actually a little surprising, since it tested almost as well as the LR classifier on the 2017 data, coming in at approximately 72.2%. This may be due to the fact that this year was rather anomalous, or it could just be that in 2017 it performed extraordinarily well. Unsurprisingly, the MYC classifier was by far the worst. While it did perform better than random, it still has a lot of adjusting that needs to happen before it can adequately compete with any of the other classifiers. Reversing the predictions of the MYC classifier also doesn't significantly improve the score. This is because it's just a little better than random, and as such, wouldn't have any better chances in later rounds by switching it's previous decisions. To illustrate this point, the expected value for a randomly guessed bracket is the sum of the number of games predicted, times the probability of guessing that game correctly, times the score for that game. However,

it's important that we exclude any round where it's expected score is smaller than the points that would be awarded for correctly guessing a single game. Since this happens from the elite eight onward, the expected value is computed by

$$\begin{aligned}\mathbb{E}[x] &= \sum_{i=1}^{3} 2^{6-i}\left(\frac{1}{2^i}\right)\left(2^{i-1}\right) * 10 \\ &= \sum_{i=1}^{3} 2^{5-i} * 10 \\ &= 280\end{aligned}$$

The MYC classifier got 360 points, which is 80 points higher. The others are much better, and while they don't perform as close to perfect as desired, still manage to get into the higher percentiles. The other metric used to evaluate the classifier's success was the number of games guessed correctly (GGC), as well as the number of games that the classifier couldn't get right, because of previous guesses. That is, how times did the classifier pick a team that had already been eliminated in previous rounds. These games are referred to as "Impossible Games" (or IGs) because they were impossible for the classifier to correctly predict. Likewise, the points from those games are referred to as "Impossible Points" (or IPs). Those a comparison of those values are displayed in Figure 3 and Figure 4.

The random values were calculated using the fact that $\left(1 - \frac{1}{2^{i-1}}\right)\%$ of games for each of round $i = 2, ..., 6$ should be impossible games. Again, it's important to note that the RF classifier varies, and can performs better than the LR classifier at times, as well as worse than the GNB classifier. In this metric, the LR model again consistently performs the best (with few exceptions from arbitrary runs of the RF classifier), followed by the PageRank classifier with the GNB classifier next, closely followed by the SVM classifier. So for this tournament, the GNB classifier correctly predicted more of the early games, but then the SVM classifier correctly guessed more of the heavier weighted games later on, resulting in a higher overall score (which is evidenced from Figure 2). All of them, though, are better than the expected value of randomly guessing.

Figure 3: *Games Guessed Correctly vs Impossible Games*

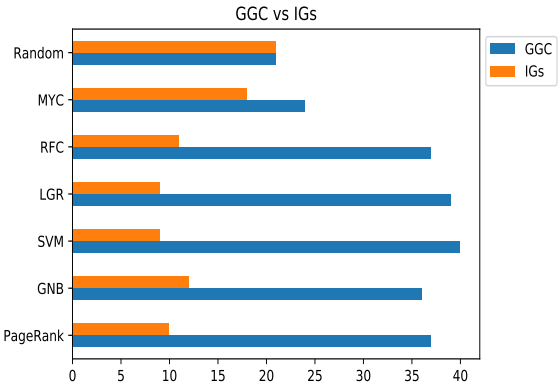
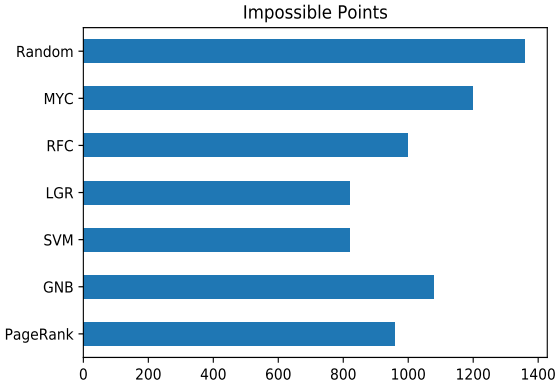


Figure 4: *Impossible Points for Each Classifier*



Conclusion

In conclusion, after organizing the data as described, standard classifiers such as logistic regression and support vector machines work just as well as traditional ranking methods for predicting March Madness. The scoring algorithm created is not much better than random chance, and currently not worth the time and energy to run it. While this is more than a little disappointing, it would makes sense; especially since there are other points of data that were not included in this model. Some of these variables that may play a significant role are distance the court is away from each teams respective homes, current winning streaks, schedule difficulty, rebounds, and others. The impact of these statistics should be considered in future iterations of the MYC classifier.

The main success of this project has been the ability to predict the tournament with about as much accuracy as a ranking system, without ranking the teams. Since the vast majority of algorithms to predict the tournament ranks the teams, and then determined the winners based solely on those rankings, this provides a new and exciting chance to approach the problem from a different angle.

This project has also opened up some new questions for further study. As stated before, one of these is how the best parameters change from year to year. Are they consistent, or arbitrarily random? Another good question is, what if you tried to classify teams based on how far they'll go in the tournament, as opposed to whether or not they'll win certain games? Would a method like that be beneficial, or would you end up with 20 teams that could be in the elite eight and end up with less information than you started? Is previous success in the tournament a determining factor, or merely noise in the system? Are the best parameters for each classifier something that can, and should, be predicted, or random enough that it's impossible to guess which ones will perform best? The search for better ways to predict march madness certainly isn't over. As a result, the quest for the perfect bracket will continue.