Predicting March Madness Outcomes

Lane Cazier

May 2021

1 Introduction

The NCAA Men's Basketball tournament, or March Madness, is one of the most popular and most watched sporting events in the U.S. There is a great deal of betting and speculation around it, and one of the biggest activities centered around March Madness is making brackets attempting to predict the outcome of every game. This paper aims to discuss Machine Learning Modeling that can help look for patterns in the data, in order to make better predictions about a team's performance in the tournament.

There are approximately 9*10E18 different possible tournament outcomes for March Madness,

$$2^{32} * 2^{16} * 2^{8} * 2^{4} * 2^{2} * 2 = 9.2234 * 10^{18}$$

and so obviously trying to reach perfection is nigh-impossible, but by utilizing careful analysis of data regarding the teams participating, it is possible to increase the accuracy of predictions. Because March Madness is such a large sporting event gathering so much interest, many models have already been made, giving a large pool of past experience to draw from. To avoid duplicating strategies already tried, a different method will be employed to hopefully find new information in the available data, and this method will be discussed later in this paper.

2 Literature Review

As mentioned in the introduction, many machine learning methods and regression models have previously been constructed to attempt to analyze the March Madness tournament. These past models can be used to help better inform new models, and so the models and past research done that can be used will be explored here.

To predict the outcome of individual games, past game statistics are often used, but past research has found that it is sometimes difficult to know which statistics are most causally related to winning (Gumm et al). Other models have used a myriad of factors, such as average field goals made per game, average three point shots made per game, average free throw attempts, average rebounds, average steals, average blocks, and since sports statistics are collected and kept consistently, there are an incredible number of possible variables that can be used in a model(Shen et al). However, some of the most effective models have actually been fairly simple and straightforward, such as the model developed by math Professors Gregory Matthews and Michael Lopez (Ellenberg). Using logit regression, they utilized only two variables in their model: the Las Vegas point spreads, and the team efficiency ratings developed by Ken Pomeroy (Ellenberg).Their model won a previous competition hosted by Kaggle, which had many people develop models to predict win probabilities of March Madness games. Models here will use some of the same data, and attempt to keep this simple approach, and not attempt to over complicate the model. The efficiency ratings developed by Ken Pomeroy and used in the previously mentioned model are also very useful and intended to be predictive on their own (kenpom.com).

Another attempt at analyzing March Madness proved particularly useful and enlightening. As has been observed, ”there is significant evidence that tournament models don’t reflect regular seasons models” (Gumm et al). One previous model attempted to analyze some of the factors that differentiated March Madness games from regular season games, so as to determine if some teams gain an advantage from variables often excluded from models of the tournament. John Ezekowitz of Harvard Sports Analysis developed what he called a ”Survival Analysis Model”, which looks at how several factors including strength of schedule, previous March Madness experience and consistency affect the teams staying power in the tournament (Ezekowitz). These variables and several others were regressed in a Cox Proportional Hazards model, on the Y variable of how many wins a team was able to get in the tournament. This model proved to be highly successful, and some variables from it will also be included in the models developed here.

3 Data

Data for this project was gathered from two sources, Kaggle and kenpom.com. Data was gathered about a number of variables for every season played by a NCAA division team, for a total 5790 observations, but this data is whittled down quite a bit. Only data regarding teams which made the March Madness tournament are relevant for this project, so despite the dataset containing observations for all teams, observations of teams that did not make the tournament in a season have been dropped. The data is for the years 2005-2019. The final dataset has 987 observations, as observations for some teams in the timeframe were dropped due to difficulties joining the Kaggle and Kenpom data entries for them. The teams dropped are Texas State and Missouri State. There are seven variables used for the model. These entries are MAR, as they are missing for reasons uncorrelated to the data itself. The number of March Madness games won by the team in the given season is the independent variable, starting with the round of 32. Other variables are the team's regular season win percentage, the team's offensive efficiency, the team's defensive efficiency, the number of tournament participating teams beaten in the regular season, the team's strength of schedule, and the number of tournament games the team won in the last two years.

The regular season win percentage should be a useful indicator of how well a team generally plays. The offensive and defensive efficiency of a team are ratings based on how many points the team scores and allows per 100 possessions, and serve the same purpose as the regular season win percentage; they measure how well the team generally plays. The number of other tournament qualifying teams beaten serves as an indicator of a teams success in higher pressure, tougher games, which should transfer well to March Madness. The team's strength of schedule and March Madness wins over the previous two seasons serve the same purpose.

All the data was normalized using the min-max method to enable it to be used for a K-Nearest Neighbors model.

4 Methods

Instead of the normal predictive method of generating win probabilities for individual tournament games, the following models turn March Madness into a classification problem, and use supervised machine learning to solve this problem. Any given team is capable of winning anywhere from zero to six games in the tournament, and the models here seek to classify each team according to how many games they are most likely to win.

Out of the data gathered, the sample was split into seventy-five percent training data, and twenty-five percent testing data. Two models were then generated using this data, one model using K-Nearest Neighbors, and a Naive Bayes Model. The number K in the K nearest neighbors was three, which is a somewhat smaller number, but using larger values of K proved ineffective due to the smaller dataset. For any team they are given data for, the model will classify based on the number of games the model thinks it is most likely to win.

The models were made in Python, using the scikit-learn package. In order to import the data into Python, the xlrd package was used to read data stored in an excel file. The splitting of the data was achieved without any package at all. The dataset was given a randomized order in excel, using the =randbetween function and sorting the data by this number. Once this was done, one in every four observations was moved into a different 2-D array in the python script itself, splitting the data randomly into the testing and training numbers.

5 Findings

Here are some metrics evaluating the accuracy for the K-Nearest Neighbors model:

For Classification	0	1	2	3	4	5
Recall	.765	.302	.158	.063	0	0
Precision	.579	.284	.375	.1	0	0
F-Score	.66	.292	.222	.077	0	0

Accuracy: .465

Note: No teams that won six games were randomly selected into the testing data. Given these teams represent 1/64 of the population, this is not shocking.

Overall, this classification predictor has some issues. It tends to predict lower numbers of wins for teams. Because one half of teams get eliminated every round, most teams end up with a lower number of wins. It makes sense then, that a K-nearest neighbors method with a smaller sample size would more frequently predict lower numbers of wins, as most of the population is comprised of low scoring teams that may end up as the closest neighbors of new data points. A larger dataset could help alleviate this issue. Considering that there are only fourteen six-win teams in the data, it is unsurprising that using nearest neighbors, these teams tend to be outweighed and outnumbered. If there were more of these teams, it would help the model group similar teams in the data, and predict similar outcomes. The largest number of wins the model predicted for teams was three total wins, which would bring a team into the round of eight. Considering that only four out of sixty-four teams do better than this, this predictive tendency towards low numbers of wins also makes sense, as that is the most likely outcome for most teams. The model also did better with recall than precision due to its tendency to assign teams a low number of wins. It frequently classified a team as a zero win team accurately when the team did get zero wins, but also when a team did better.

From one perspective, an accuracy of .465 represents an encouraging number. A program that randomly assigned a classification to each team would be right only 1/7 of the time. On the other hand, a program that classified each team as a zero would be right fifty percent of the time, of half. Clearly this model has some room for improvement.

Here are some metrics to help evaluate the Naive Bayes model:

For Classification	0	1	2	3	4	5
Recall	1	.016	0	0	0	0
Precision	.471	1	0	0	0	0
F-Score	.64	.031	0	0	0	0
Accuracy: .465						

Again note there are no six win teams in the testing data.

On first glance, the Naive Bayes model boasts a slightly higher accuracy score than the K-Nearest Neighbors model, but it is generally a worse model. It obtains its better accuracy by being far more likely to class a team as a zero win team. It does not predict any team to win more than one game in the tournament. It does boast a pristine recall of one for zero win teams, but this is again due to the fact it rarely classifies a team as anything else. This model was very disappointing, and not much can be salvaged here.

6 Conclusion

Generally, the models were somewhat disappointing. However, it is important to remember that attempting to predict March Madness results is very complicated, and accuracy is difficult. Improvements to the models are certainly possible, however. Some future work could include expanding the amount of data and observations gathered, and incorporating a new variable, such as the seed of a team, would probably help the models tremendously. Another thing I intend to try in the future are making several different models using the available data, such as a random forest model. While imperfect, I do think that the K-Nearest Neighbors model showed potential, and could be useful in its current form to show where a team with given data generally falls in an average year. I think it can be used to show if a team is more or less likely than its seeding suggests to lose early or to go further than expected. The models can be useful, and hopefully will be better with future work.

7 References

Ellenberg, Jordan. “The Math of March Madness.” The New York Times, The New York Times, 20 Mar. 2015, www.nytimes.com/2015/03/22/opinion/sunday/making-march-madness-easy.html.

J. Gumm, A. Barrett and G. Hu, ”A machine learning strategy for predicting march madness winners,” 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015, pp. 1-6, doi: 10.1109/SNPD.2015.7176206.

Pomeroy, Ken. “Ratings Explanation: The Kenpom.com Blog.” The Kenpomcom Blog, Kenpom.com, 29 Nov. 2016, kenpom.com/blog/ratings-explanation/.

Shen, Gang, et al. ”Predicting Results of March Madness Using Three Different Methods.” Journal of Sports Research 3.1 (2016): 10-17.

“Survival of the Fittest: A New Model for NCAA Tournament Prediction.” The Harvard Sports Analysis Collective, Harvard Sports Analystics, 12 Jan. 2013, harvardsportsanalysis.wordpress.com/2012/03/14/survival-of-the-fittest-a-new-model-for-ncaa-tournament-prediction/.