March Madness through Machine Learning

Luke McMeans, Nithil Suresh, Bay Wiggins

Terrance Johnson

DS 3001

Abstract

In the following paper, we aim to predict how far each team will advance in the March Madness tournament using machine learning models trained on historical team performance data from 2008 to 2025, including advanced metrics from the KenPom and Barttorvik rating systems. We used Logistic Regression, Random Forest, and neural network models to treat each output as a classification problem based on tournament rounds. Logistic Regression served as a baseline and provided insight into feature importance and reached an accuracy of about ~51%. The Random Forest Model, paired with SMOTE and class balancing achieved ~47% overall accuracy. The neural network got a slightly higher overall accuracy of ~49% and showed stronger performance in correctly identifying first-round losses and occasional upsets. All models also shared similar accuracies across all rounds in the tournament, with only the Random Forest and Neural Network models getting at least one champion pick correct. Despite modest predictive power in the later rounds, our models captured meaningful trends, such as the influence of seeding and efficiency metrics. Ultimately, our findings signify that March Madness truly is as its title claims. Individual games, as well as a team's run in the tournament, are hard to predict. Being able to accurately predict the majority of the games in a given tournament is a tall task given the variability of upsets in post-season college basketball. Despite some interesting findings, we don't consider our models to be revolutionary in predicting the tournament.

Introduction

The month of March is known for being the first month of spring, St. Patrick's Day, and the National Collegiate Athletic Association (NCAA) Division 1 Men's basketball tournament, colloquially known as March Madness. This single elimination bracket-style tournament contains 64 competing teams (68 accounting for the play-in "First Four" games). The tournament is split into four regions: East, West, Midwest, and South. In each region, there are 16 teams and each team is assigned a seed representing how well that team performed in the regular season. Thus, the number one seeds in each region are some of the best teams while the number sixteen seeds in each region are amongst the worst of the teams that qualified. There are six total rounds, ranging from the Round of 64, Round of 32, Sweet 16, Elite 8, Final 4, and the Championship. A team must get through all of these rounds to claim the national championship title. Despite the seeding, the lower-seeded team can shock the world and defeat their better opponent. These 'upsets' give the tournament its title of 'March Madness.' In the tournament's long history, a 16-seed has beaten a 1-seed twice. Along with this, there have been a handful of teams that have made deep, improbable runs in the tournament: 2018 (11) Loyola Chicago, 2022 (15) Saint Peter's, 1985 (8) Villanova, and more.

In this paper, our primary question is figuring out how far each of the 64 teams in March Madness will advance in the tournament. Additional topics we would like to answer include being able to predict the tournament accurately, lower-seeded teams making unexpected runs (Cinderella Runs), and who is most likely to win the national championship.

Data

The dataset we chose to analyze is the KenPom Barttorvik set from Kaggle's March Madness Data. This dataset provides statistics for each team in the NCAA Division 1 Men's

Basketball Tournaments from 2008 to 2024. The tournament for 2025 was later added, with each value indicating the round set to 0. We have since manually input those values as the tournament has concluded. With these teams, their statistics detail their seeding, the furthest round they played, and a large selection of advanced data which can be found in the KenPom and Barttorvik ratings at the time for each respective tournament.

Methods

With this dataset, an observation would be a team in a given year's March Madness tournament. Each team has a set of ratings from the KenPom and Barttorvik sites, indicating their offensive and defensive efficiencies, tempo, statistical rankings, and more. Along with this, our data also contains the farthest round that each team went that year, which is what we will be predicting. To do so, we will use supervised learning. This dataset contains information from each tournament since 2008, detailing how far each team went in each tournament. We can use this to train our models. Since we're trying to predict how far a team goes, the predicted values are categorical (even though their values are numbers). Given this, we will be working with a classification approach. We only have a discrete amount of values for their 'round' value, and a strict number of teams that can have each value.

Given our desire to predict the tournament, we will approach this from multiple angles. We'll establish a baseline by doing logistic regression. This feels like the most simple approach and has good prevention of overfitting if correct regularization is applied. Along with this, logistic regression works better than most with smaller datasets. Though we have a lot of data to work with, we only have 17 tournaments overall, so this information could prove to be beneficial. It also can help determine which statistics (features) are most important in determining how far a team goes. We can also use a Random Forest Classifier to help with

non-linear relationships in our data. We are also able to determine the important features in determining predictions with this model. An important aspect of this model is that it can help the randomness of upsets. At the same time, this model is good at handling outliers, ensuring nothing is too crazy in our predictions. These two pieces of information, in tandem, ensure we can make reasonable predictions and upsets. We also find it beneficial to utilize a neural network. This type of model is very good at finding hidden trends and complex patterns in our data, which could be very helpful in our predictions given the variability in results year after year. This is another piece to help us find out what it takes to win in certain rounds, and even pull off some upsets. Hopefully, this can build off the Random Forest and create reasonable 'Cinderella' teams and underwhelming performances.

Our model works if we can predict reasonable results for each year's tournament given the ratings from the respective sites. For this to be accurate, the results must be considered reasonable. For example, a 16-seed has never made it past the second round, so we should be extremely suspicious if one makes a deep run in our predictions. As long as the general trends of the tournament, with appropriate winners and upsets, we should be in good shape. We can use a confusion matrix to see what rounds or years provided the most accurate predictions. This also provides a metric if the model overestimated or underestimated their predictions. Finally, as our classes are imbalanced, utilizing a metric such as an F-1 Score could provide more insight into performance beyond just accuracy.

In terms of potential weaknesses, we were worried that our predictions could be 'chalk,' meaning the higher seed will win significantly more than average (if not every game). Though their higher seed signifies they're the better team, this is called 'March Madness' for a reason. Sometimes, the worse team will make surprises against the better ones. We intended to pull

information from relevant Cinderella teams, align with finding how often subsets happen, to implement upsets where reasonable. Our Random Forest and Neural Network models could be beneficial in minimizing this, but you never know what can happen. Another potential flaw is that neural networks traditionally require large amounts of data to generalize well. Despite our previous concerns about the 'size' of this dataset, we still chose to continue with the approach just in case it turns out to be effective. Lastly, we may very well end up finding it practically impossible to figure this tournament out. The odds of predicting a bracket with perfect accuracy are one in 9.2 quintillion (assuming all games have 50/50 odds). With this, our goal is to be as accurate as possible.

Results - Logistic Regression

As previously mentioned the primary purpose of running our logistic regression was to create a baseline, where we can identify key features that help us figure out how far a team will advance in March Madness. The Logistic Regression model that was used was imported from SciKit. To run a classic logistic regression, we set the parameter *penalty = None*. Additionally, we set *max_iter = 1000* as a stopping condition to prevent the model from running forever. After we ran the regression, we achieved an accuracy level of about 50.917%. The final version gave us these results:

Round	**# Predicted**	**# Correct**	**Accuracy**
Lost in Round of 64	135	90	66.7%
Lost in Round of 32	46	15	32.7%
Lost in Sweet 16	19	4	21.1%
Lost in Elite 8	8	2	25.0%
Lost in Final Four	1	0	0%
National Runner-Up	7	0	0%

National Champion	2	0	0%

The logistic regression does fairly well in predicting the first few rounds of the tournament. However, during the later rounds starting in the Final Four the model's accuracy falls off completely, failing to achieve any accurate predictions. Here is where other models like a Random Forest Classifier and a Neural Network might be able to step in and make better predictions deeper into the tournament.

Next, we will look at some of the upsets that the Logistic Regression model was able to predict.

Year	Team	Seed	Predicted	Actual
2024	North Carolina	1	16	16
2023	Auburn	9	32	32
2023	Iowa	8	64	64
2023	Iowa St	6	64	64
2023	UCLA	2	16	16
2022	Boise St	8	64	64

As we can see from the table above, the model did not always pick the better ranked seed as it was able to pick up some impressive upsets. The prediction that the #1 North Carolina team would be knocked out of the tournament in the Sweet 16 was one of the best upsets predicted by the model.

Results - Random Forest Classifier

Our Random Forest Classifier (RFC) approach provided a traditional ensemble method for handling the prediction task. Unlike a method such as Neural Networks, the RFC depends

less on sequential learning and more on a majority vote across multiple decision trees. To address class imbalance (as most teams lose in the round of 64) we used a combination of **SMOTE** (Synthetic Minority Over-sampling Technique) and balanced class weights. This helped the classifier jump from ~2% accuracy on initial models to ~47% on the final version. This final version gave us these results:

Round	**# Predicted**	**# Correct**	**Accuracy**
Lost in Round of 64	109	75	68.8%
Lost in Round of 32	55	18	32.7
Lost in Sweet 16	27	6	22.2%
Lost in Elite 8	14	3	21.4%
Lost in Final Four	7	1	14.3%
National Runner-Up	3	0	0%
National Champion	3	1	33%

Other precision metrics were evaluated using precision, recall, and F1-scores per class. While accuracy was ~47% as mentioned above, it is a respectable result given the extreme imbalance and unpredictability of our dataset. The model showed improved recall and F1 in later rounds compared to the naive classifier used before our balancing techniques. The model maintained strong performance on the round of 64, and did not collapse entirely on later rounds. Using feature importance, we saw that seeding and efficiency metrics had the greatest influence on predictions, validating assumptions about what determines early success (i.e. good teams tend to have good results).

Results - Neural Network

For our Neural Network approach, our accuracies seemed to be very consistent. We transformed the data into a PyTorch dataset, emphasizing the seeding. With this, we're able to have higher seeds more likely to win games (which is more typical). Our procedure warranted the following results by round:

Round	**# Predicted**	**# Correct**	**Accuracy**
Lost in Round of 64	109	83	76.15%
Lost in Round of 32	55	15	27.27%
Lost in Sweet 16	27	4	14.81%
Lost in Elite 8	14	4	28.57%
Lost in Final Four	7	0	0%
National Runner-Up	3	0	0%
National Champion	3	1	33%

We see that this model has pretty decent accuracy in predicting teams that lose in the Round of 64. This should be expected, as half the teams lose in the first round, most of which are heavily predictable. For example, a 1-seed has only lost to a 16-seed twice ever, with 15 over 2 and 14 over 3 upsets being rare as well. The predictions get harder as we progress into the tournament. Again, this should be expected. In bracket pools, correct predictions in later rounds warrant more points. It's hard to get these correct, as the number of possible teams to fill that spot increases exponentially. We were fortunate to get a national champion correct, but we can see that this could've been luck as there were no correct predictions for teams that lost in the Final Four or National Championship. Overall, we received an accuracy of 49.08%, with that number fluctuating around 50% for each test of the code.

We also checked to see the accuracy of upsets. We were impressed to see there are a few that were predicted, especially for those of higher seeds losing early. However, the upset accuracy is only about 6% at best. This is not great by any means. At the same time, they're called upsets for a reason. They're unexpected, so all statistical reasoning can be disregarded when these happen. There seems to be true randomness with some of these games, too random to be accounted for by the model. Here are a handful of upsets that the model was able to predict during its trials.

2010 (1) Syracuse losing in the Sweet 16	2023 (13) Furman reaches the Round of 32
2025 (12) McNeese reaches the Round of 32	2024 (3) Baylor losing in the Round of 32
2021 (2) Iowa losing in the Round of 32	2013 (5) Oklahoma St loses in Round of 32
2013 (2) Georgetown loses in the Round of 64	2015 (2) Kansas loses in the Round of 32

Conclusions

As a bonus approach, we ran the 2025 dataset through rounds of filtration based on the previous winners. Every statistic had a highest and lowest value based on these champions, in which the only team that fits within all these intervals was 1-seed Florida. Conveniently, the Gators took home the championship with a win against 1-seed Houston, making this approach 'accurate' for the year. However, this was not a clear prediction model for every year. For 2024, 1-seed Connecticut (the correct champion) was the last team remaining in filtration but didn't meet every interval. Other years didn't have the same accuracy, either having no teams left or having the incorrect team remaining.

Below is an overall accuracy table for all three models, using all the previous data.

Round	**LR Accuracy**	**RF Accuracy**	**NN Accuracy**
Lost in Round of 64	66.7%	68.8%	76.15%

Lost in Round of 32	32.7%	32.7	27.27%
Lost in Sweet 16	21.1%	22.2%	14.81%
Lost in Elite 8	25.0%	21.4%	28.57%
Lost in Final Four	0%	14.3%	0%
National Runner-Up	0%	0%	0%
National Champion	0%	33%	33%
OVERALL	*50.92%*	*~47%*	*49.08%*

We can see that no approach truly stands out in overall accuracy, with every round having similar values as well. If we were to convert these round accuracies to a bracket pool score, Random Forest would be the best with 61.568 points, with Neural Network totaling 57.536 points, and Logistic Regression in last with 46.56. With the maximum possible points being 192, all of these would not boast well. With this, we conclude that it's hard to find patterns in the 'Madness.' The variability of this tournament overpowers the statistics of these teams. Every team wins for a different reason, so it's difficult to portray that statistically. We're sure we can get better accuracy with more time and a project of a higher caliber, but for now, we aren't able to find an accurate manner to predict the tournament. One limiting factor of our models was that we only looked at the statistics of the team we were trying to predict, not taking into account the statistics of the opposing teams. This could prove to be a crucial part of the picture as the model might be able to make better predictions by comparing the statistics of both teams.

Another key consideration for future work is the structure of the modeling task itself. Our current models treat each team's tournament outcome without taking matchups into consideration, which contradicts the actual format of March Madness. Each team's advancement is inherently conditional on who they match up with in each round. A different approach would model the bracket as a sequence of dependent events, using a prediction method that takes the

predicted outcomes of previous matches into account when predicting the next rounds. This would also allow for simulating entire tournaments using Monte Carlo simulations (a method that uses random sampling for predictions, used for problems where deterministic solutions are difficult or impossible), where randomness and matchup context can be used.

Additionally, while our models used statistical data from KenPom and Barttorvik, they ignored other factors such as injuries, coaching, player experience, or player rotations. These are all difficult to quantify but are often critical in real outcomes. Incorporating these factors could significantly improve context, and potentially the predictions. In conclusion, while our models offer some insights into March Madness outcomes, they fall short of providing truly actionable predictions. The inherent chaos and noise of the tournament defies data-driven forecasting. However, this unpredictability is what makes March Madness so enjoyable to watch. From nail biting conclusions to major upsets to Cinderella runs, a part of us is glad that no machine learning model (so far) has been able to harness the craze that is the NCAA March Madness tournament.

References

https://www.kaggle.com/datasets/nishaanamin/march-madness-data?resource=download&select=KenPom+Barttorvik.csv