Anarchy? Nope. Just Probability
Predicting and Analyzing the 2023 NCAA Tournament

Spencer Kerch

April 25, 2023

Abstract

The yearly NCAA march madness tournament is one of the most difficult prediction problems, as a perfect bracket is effectively impossible. However, predicting the outcomes of each individual game is a much more surmountable challenge. In my project I use free data in-season and tournament data from Kaggle's March Machine Learning Mania challenge and from public advanced stats websites for college basketball in an attempt to assign a win probability to each possible match-up in the 2023 March Madness tournament, as well as probabilities of each team to advance for each round through a hierarchical model. I have built multiple predictive models and will compare and contrast the three and examine what is driving their differences.

1 Introduction

While predicting what teams can win in March is often a futile task, I believe my models are a success. I started to this project in hopes of using KenPom data as my main predictor. But found out through investigating the predictive power of his public data that I would require a subscription to access pre-tournament historical statistics. While I would love to supports KenPom I am still a college student and could not fit it into my budget. I eventually found a free alternative in Torvik Data from Bart Torvik. The data was very similar and allowed me easy access to in-depth data I could scrape. In this project I used Torvik data in conjunction with the data from the Kaggle competition to predict all possible game match-ups in the 2023 NCAA Tournament. I compare random forest and logistic regressions to find the optimal model for prediction, and use my results to build a hierarchical conditional probability model to assign each team a probability that they will advance to each given round and win the title. I end up using two separate models and compare them to assess which one was more successful.

2 Data

2.1 Data from Barttorvik.com

Most of the predictors I used came from Bart Torvik's website. They include

- Adjusted offensive(AdjO) and defensive (AdjD) efficiency, which is a team's points scored and allowed per 100 possesions, adjusted by opponent.

- Barthag, which is a variation of the Pythagorean Wins formula or

$$\frac{1}{1 + (pointsAllowed/pointsScored)^e}$$

  where e is a rational number, typically 2.

- Offensive (Efg) and defensive (EfgD) effective field goal percentage

- Turnover differential

- Offensive (Orb) and defensive (Drb) rebound percent

- Offensive (Ftr) and defensive (FtrD) free throw rate

- Offensive (2P) and defensive (2PD) two point shooting percent

- Offensive (3P) and defensive (3PD) three point shooting percent

- Adjusted Tempo (AdjT)

- Wins above bubble (WAB)

The data dates back to the 2007-2008 college basketball season and has no missing values - at least for tournament teams. Data from 2020 is left out due to the cancellation of the NCAA tournament that year. I scraped the code from his website in R. This caused a few problems as the rankings would get added on to the end of the values. For example, if a team led the nation with a AdjO of 120.02, they'd have a rank of 1 and the scraped value would be 120.021. for most values this only added at maximum 9 hundredths and was not a large issue. However if the previous value was an integer, 120 for example, it was a larger issue as it would increase all the too 1201. This issue was avoided by writing code that recognized if a value was wildly higher or lower than what the distribution should be, and lowered it. Finally, since each game has 2 of each predictor, one per team, I took the difference between each value for team 1 and team 2 to decrease the dimensions in a way that didn't change the meaning of different values. Team 1 and team 2 were chosen at random to eliminate the chances of the model picking up on a confounding variable - the data listed teams as the winning team and the losing teams, if I let the first team always be the winner that could cause some problems in my training model.

2.2 Data from Kaggle

Some more data came from the March Machine Learning Mania Kaggle competition. They include

- AP poll rankings (end of regular season)

- Tournament Slots - where each seed would play their games given should they advance
  - Example: There are only 8 available slots in the 2nd round, so the winner of the 1 vs 16 matchup at the top of the region would have the 1 slot in round 2, and the winner of the 2 vs 15 matchup at the bottom of the region would have the 8 slot in the second round.

- Seeds

- Historic regular season and tournament results by game

While some of the data from Kaggle I did use when tuning the models, it was most helpful in organizing the tournament teams and calculating their probability of advancing in each round.

3 Predictive Model

3.1 Training

I trained both random forest and logistic regression models and compared their RMSE and accuracy as I tuned them to determine the best model. The features began with all of the relevant ones listed in the previous section, and were eliminated based on importance or significance and their relevance to the model.

Leave one season out cross-validation was used in the training which results in predictions for each game in the training set. This means that for each season from 2008 to 2022 - minus 2020 - the model is trained on all but one season and then predicts the season that was left out.

After the tuning was complete, the final predictor variables were the differences in:

- AdjO

- AdjD

- Barthag

- Orb

- FtrO

- WAB

- Turnover Differential

Below are the calibration plots for the final models and a table comparing their accuracy and RMSE:
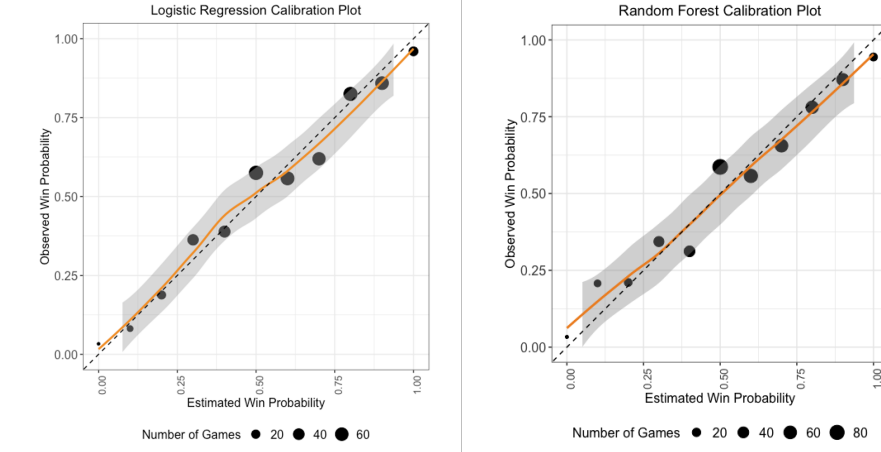

Figure 1: Training stage calibration plots
logistic regression is more precise

The calibration plots aren't perfect, but with a relatively small sample size (less than 1000 total games and no bucket larger than 80) it appears the models are running correctly. The model's error statistics also give us a clearer view of what model is performing better.

Model	MSE	Accuracy (.5 cutoff)
Logit	.1870	.7091
Random Forest	.1965	.6886

Here we can see the logistic regression is performing better and therefore we will be using the logistic regression in the training stage. This is somewhat of a surprise as random forests are more complex and typically perform better. However, many of the predictors are binary, meaning its either good or bad. For example, its is better to have a higher AdjO than your opponent and a lower AdjD, and having the opposite is worse. The random forest was likely unable to find the specific patterns in the data because the predictors didn't have many relationships that were too complex.

Finally, due to its high insignificance in the logistic regression and somewhat low importance in the random forest, adjusted tempo was left out. However I believe tempo is an important aspect of a teams identity and decided to run another model with all the same predictors plus AdjT. I will be comparing the two models to see if either (a) AdjT is not important but adds variation that allows the model to be more accurate due to the unpredictable nature of the tournament or (b) tempo is an important factor despite its low predictability in past seasons

3.2 Testing

For the testing stage, using data the 2023 regular season, the logistic regression assigned a win probability to every possible match-up - 2,278 - in the tournament and was assessed on the error and accuracy for the 67 games played. The error and accuracy of both the logistic regression (logit) and the logistic regression with added predictor AdjT (logit + AdjT) are below:

Model	MSE	Accuracy (.5 cutoff)
Logit	.2168	.6418
Logit + AdjT	.2165	.6267

Neither model performed too well, which isn't much of a surprise having watched the tournament this year as it was historically unpredictable. And while the Logit + AdjT model had a marginally lower MSE it's accurcy was a few percents lower (a difference of 1 game). In the next section we will analyze the models further and determine why, despite the inclusion of adjusted tempo, the logit model is better.

4 Model Comparison and Hiearchichal Model

4.1 Hierarchical Model

For predicting the championship probabilities of each team for each model, I calculated them with my own hierarchical model that multiplied the probability of a team advancing to a given round 1 thru 6 by the sum of the probability that they a given each team times the probability the given team also makes the given round. The model can be expressed in the formula below.

$$P_r = P_{r-1} \times \sum_{i=1}^{n} WP_i \times I_r$$

Where P_r is the probability of a team advancing to round r, P_{r-1} is the probability that team advanced to the previous round, n is the number of opponents a team could potentially face in round r, WP_i is the assigned win probability of beating team i, and I_r is the probability team i makes it to round r. Additionally, for $r = 1$, P_r is 1 for all 60 teams not in play-in games, and the probability of winning their play-in game for the 8 teams in a play-in game.

This guarantees the sum of the probabilities equal 64 for round 1, 32 for round 2, etc... up until the championship where it equals 1 as only 1 team can win the tournament.

Below are tables of advancing probability by round for both models

Figure 2: Every teams probability of advancing to each round: Sorted by championship probability