March Madness Report - Lab 3

Madison Greenough

2023-03-10

Introduction and Background

On Thursday March 16th, the biggest event in NCAA college basketball, March Madness, began. The end of season tournament hosts 64 teams playing each other in a one-loss elimination format. There are 6 rounds, starting with the Round of 64, and ending with the championship game. The bracket challenge has become a popular way for fans to get engaged in the tournament. This challenge consists of fans filling out brackets to predict how the tournament will unfold. There are tens of millions of brackets filled out every year. There are 63 games to be predicted, meaning the odds of creating a perfect bracket is 1 in 9.2 quintillion. Warren Buffett challenged fans and offered to pay $1 billion if someone could create a perfect bracket. This poses the perfect challenge for statistical modelling. Using unsupervised machine learning, can a model generate a better bracket than the biggest sports fans? If it performs well, it could become an important tool to compete for prize money and predict sporting outcomes in similar tournaments and leagues.

Methodology

There are many models out there that attempt to do the same thing. Most of these models generate a predicted outcome for each head-to-head match up. Given the time constraints and the desire to do something different than the norm, I decided to generate expected finishing ranks for each team, and then transfer this over to the bracket. I acknowledge that this may not be the best model, but I thought this would be a unique and interesting way of doing things. Also, given that there are 4 regions, technically, for example, the best four teams may not make it to the final four, so I thought that this approach would be a good way to test the even allocation of the regions. Note that the regions are made uniquely and don't actually correspond to the geographical region where the school is. They attempt to evenly allocate the rankings when assigning teams to each region.

After scraping, gathering, cleaning, and merging data, I had a list of 64 teams and 57 different prediction variables for both the 21/22 season and the current 22/23 season. I used the 21/22 data to train the model, and the current 22/23 data to test the model to generate predictions based on in-season performance and March Madness tournament results from the previous year. I analyzed the data and, after generating many sub-par models, I ended up with a linear model obtained through step-wise selection and a random forest model generated by the ranger package.

With the training data, the linear model performed surprisingly well. With an adjusted R-squared of 90% and very significant p-values, this model looked like it represented the tournament performance well. The plots also appeared to fit a normal distribution.

The random forest model did not appear to be as good as the linear model with the test data. The model gave a poor OOB R-squared of 0.30 and a large MSE of 92.

Results

After performing predictions on the 2023 data using these two models, I filled out a bracket given each model's predictions. I approached this by starting from the bottom of the model's predicted rankings, and chose the worst teams to lose, and then moved up the rankings from there. The input tournament rankings from the test model (21/22 data) were from 1 to 33, where 1 is the champion is 33 is a team that lost in the first round and did not make it to the top 32. Note that there are 32 teams that will have a ranking of 33, as there are 32 teams that lose in the first round.

As the tournament has just began, it is too early to determine the results, but it was interesting to note that the expected ranking for the linear model ranged from 0.6 to 36, whereas the random forest model gave a range of 12 to 30, which is much narrower. When filling out the brackets, there were conflicts, as this model doesn't account for which teams play each other and must force a team to move on, even if it they are both predicted to lose in a given round. In the first round, there were 7 of these conflicts with the linear model and 1 in the random forest model. I was surprised to see that the random forest model typically selected the top seed, whereas the linear model chose several big upsets early on.

Conclusion

Over the course of the tournament, it will be very interesting to see which model performs better, and if either of the are able to outperform the majority of fans. In the future, I would hope to put restrictions on teams so that it forces the model to choose a winner given the bracket layout so that there aren't as many conflicts.