Justin Littman

DS4400 Final Project

Spring 2018

Predicting NCAA Tournament games is one of the most popular activities in American sports culture. Roughly 70 million brackets are filled out every year [1]. Wealthy executive Warren Buffett has offered to pay $1 Million per year for life to anyone who fills out a perfect bracket [2]. And yet, year after year, his promise goes unpaid. The probability of randomly selecting a perfect bracket are as low as 1 in 9.2 quintillion [3]. Most people do better than that, and use prior knowledge and other information to make better informed picks. And yet, it still doesn't get people much closer to that elusive perfect bracket. The next step, going beyond human intuition, is to develop a predictive model to help assist with making picks. There is a lot of data available on teams in the NCAA Tournament, and I believe there is a real opportunity to use that information to make better predictions. This year, out of 17 million brackets filled out on ESPN, the top bracket predicted only 44/63 (70%) of matchups correctly. This number does not seem notable at first glance, but it is pretty remarkable when you consider it. A big problem with the NCAA Tournament is that incorrect picks, especially early on, have compounding costs. If a team you pick to win multiple rounds loses in the first round, that results in multiple rounds that will now have incorrect picks. The key to a good bracket is balancing risks with safe selections to win.

I used sports-reference.com to acquire data on all NCAA Tournament teams over the last 20 seasons. I downloaded both standard and advanced statistics on the teams, and combined all the teams into a single file. I felt the data available through this site would be interesting, and it would capture many of the variables that could be relevant. The dataset had good quality, and there was only one variable (Pace) that was missing in over 50% of observations. I had three other variables that were missing values in

38% of observations. I felt that these descriptive variables could offer significant predictive power, and I wanted to find a way to include them in my model development. Rather than discarding the variables, I decided to impute their values. I used the MICE package in R, which uses a multiple imputation method to impute values. This method was a good choice because while many values were generated and added to the dataset, the distribution of each of the descriptive variables remained largely unchanged.

One challenge for this dataset was that the interpretation of data was not necessarily consistent from year to year. I had a dataset that spanned 20 years, and the game of basketball has seen many changes over that time span. For example, a team that averaged 70 points per game in 1997 might have been considered to be a high scoring team, but that number is strikingly average today. To account for this, I decided that I was going to normalize my data on a year by year basis. I converted each continuous variable against the mean and standard deviation of the given year, and by doing this I could mitigate differences in the data from year to year.

Of all the challenges I ran into during the development of this project, the most time consuming one required me to convert all of the team statistics into individual matchups, which had a binary outcome variable as a target. The class of interest was to predict whether the lower-seeded team would win their matchup. I had to do this conversion because I wanted to ultimately predict the outcome of NCAA Tournament matchups, and I needed to transform the relationship between two teams into a single matchup vector. After doing some research, I found that a viable approach would be to

simply subtract the statistics between the two teams. All of my descriptive variables were continuous, so this seemed like a reasonable option.

I struggled to find a data source that listed out each NCAA Tournament game in a matchup format that would be easy to parse. There did not seem to be much information available outside of the standard bracket format created each year. I decided that if I really wanted to work with this data, I was going to have to go one by one, and manually convert each matchup. This process took many hours, but at the end I had curated a viable dataset to use to predict individual matchups.

The two machine learning models I chose to develop for this task were logistic regression and a boosted decision tree. Logistic regression seemed like a reasonable choice for this type of binary classification, because it will simply output a probability of the lower-seeded team winning. A boosted decision tree also seemed to be a good choice because less than 30% of the target outcomes in my dataset represent the class of interest, and I wanted to focus on trying to improve the model sensitivity. Additionally, I thought it would be interesting to approach this problem with two different types of machine learning models.

The last thing I needed to do before training my model was to check for any descriptive variables that may be highly correlated with one another, and only keep variables in the dataset that have little correlation to others. I removed many variables during this process, as there were a significant number of highly correlated variables. Turnovers vs Turnover Percentage, Free Throws Attempted vs Free Throw Rate, and Points Per Game vs Pace were all examples of descriptive variables that were highly correlated, and it did not make sense to keep all of them. I ultimately removed 26

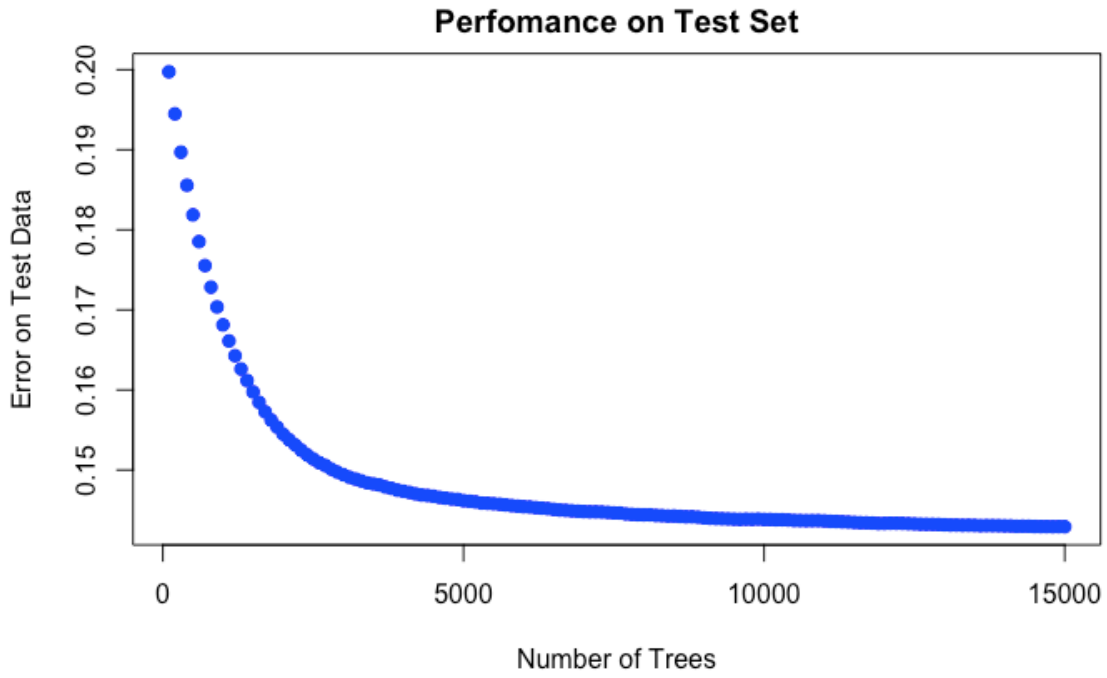
descriptive variables, and was left with a dataset of 20 descriptive variables to help me predict outcomes.

The logistic regression model was the first model I built. I used backward selection to discard insignificant variables, and ended up with a model that had 7 significant variables.

```
Call:
glm(formula = result ~ overallW + SRS + homeW + homeL + awayW +
    FG. + X3PA, data = train[, -21])

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.83049  -0.28625  -0.09817   0.32140   0.91075

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.494357   0.017793  27.783  < 2e-16 ***
overallW    -0.069199   0.004336 -15.958  < 2e-16 ***
SRS         -0.006067   0.002030  -2.989  0.00288 **
homeW        0.037952   0.005897   6.435 1.98e-10 ***
homeL       -0.024355   0.007907  -3.080  0.00213 **
awayW        0.055298   0.005749   9.619  < 2e-16 ***
FG.          0.032577   0.010061   3.238  0.00125 **
X3PA         0.018571   0.009272   2.003  0.04549 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.1362264)

    Null deviance: 189.59  on 925  degrees of freedom
Residual deviance: 125.06  on 918  degrees of freedom
AIC: 791.92

Number of Fisher Scoring iterations: 2
```

When evaluated against the test data, this model had an accuracy of 81%, and a sensitivity of 58%. I understand predicting NCAA Tournament matchups is a very difficult task. I was not sure what results to expect from my model, and I was fairly pleased with the accuracy. However, I was primarily interested in identifying lower-seeded victories, and the sensitivity had some room for improve.

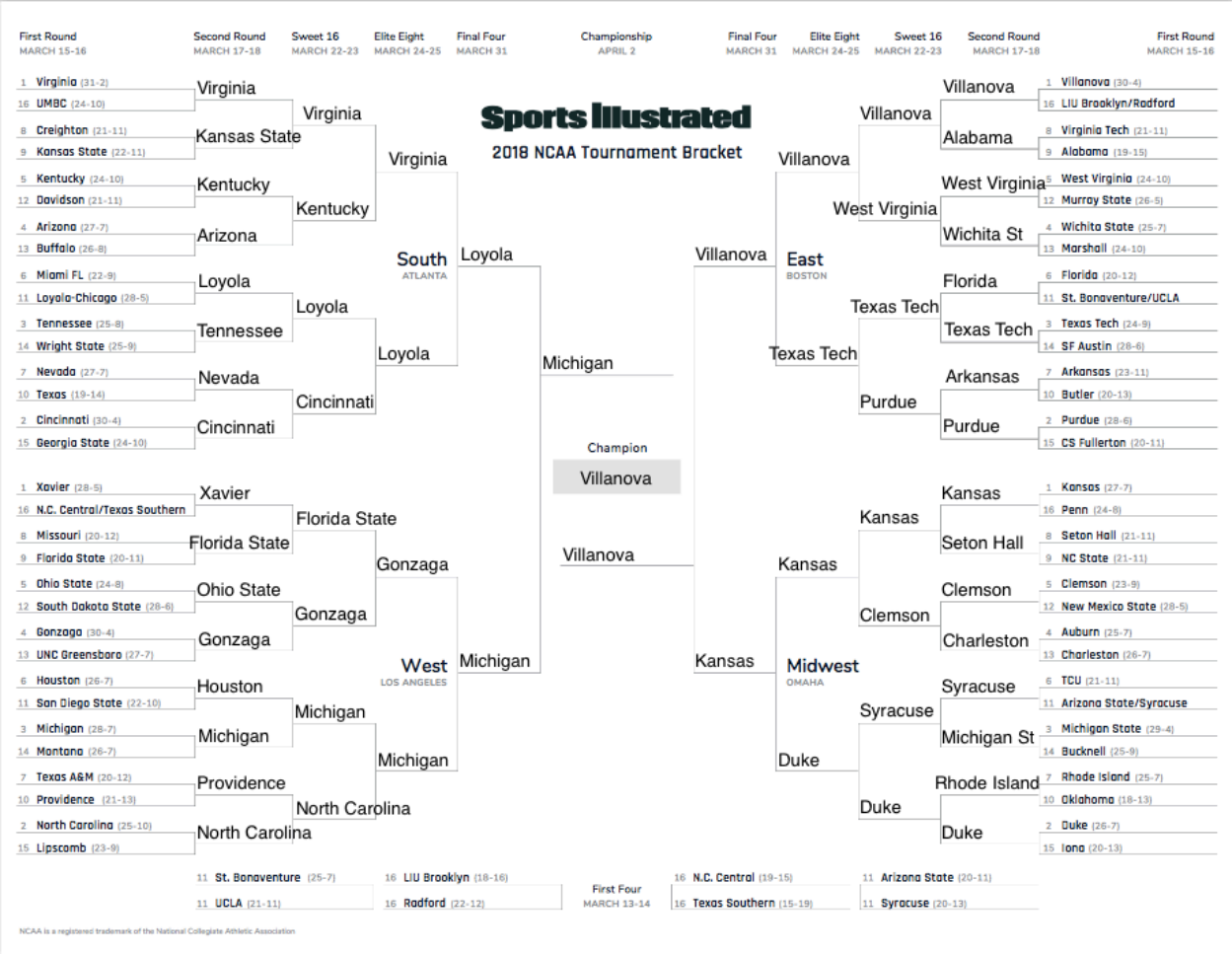
I built the boosted decision tree next, and I built it up to 15,000 trees because the error continued to decrease. Ultimately, the 14,000th tree iteration had the lowest error on the test set.



When compared with the test data, this model had a slightly better accuracy measure than the previous model, with an accuracy of 78%. More significantly, however, was that the sensitivity of this model was 69%. While the accuracy was slightly worse than the logistic regression model, this model performed notably better on the class of interest. I was certainly pleased with these results.

My initial goal was to finish this project in time for the NCAA Tournament, but I was unable to develop a model by that point. However, with a model trained on NCAA Tournament data from 1997 to 2017, I thought it might be interesting to deploy it and see how it performed on the 2018 NCAA Tournament. I retrieved data for all teams that qualified for the 2018 Tournament, and transformed it into the proper format. I then went ahead and retroactively filled out a bracket, and the results were astonishing. The model correctly predicted the exact final four, championship matchup, and eventual champion. Using ESPN's scoring system, this bracket would have been the top performer out of

over 17 million brackets that were filled out. Below is the bracket filled out with predictions from my predictive model.



These results were remarkable, and better than anything I could have possibly expected. After closer investigation, I started to realize that perhaps these predictions would not have been as accurate if I deployed the model before the tournament. The 2018 statistics I retrieved for each team were cumulative statistics for the entire season, including any games played in the NCAA Tournament. If a team over performed in the NCAA Tournament, it could have skewed the statistics for the season. Once again, I came to realize that predicting March Madness outcomes is still an incredibly difficult task. Even when you try and account for the challenges that I described earlier, there

are still many more challenges that end up arising. While my preliminary results were encouraging, even if they were flawed, I plan to continue working on this project to have a model I can deploy next year. I would like to see if it is possible to retrieve data on teams before they play in the NCAA Tournament. I might also try to predict margin of victory rather than a binary outcome, because not all victories are the same. I might also try to account for the role that luck plays in single matchups. I looked through some academic papers on accounting for luck in sporting events, but I may try to investigate in further detail.

References

[1]http://bleacherreport.com/articles/2697846-march-madness-2017-70-million-brackets-104-billion-in-bets-expected

[2]http://time.com/money/5195356/warren-buffett-march-madness-2018-bracket-challenge/

[3]https://www.google.com/search?q=odds+of+a+perfect+bracket&oq=odds+of+a+perfect+bracket&aqs=chrome.0.0l6.4420j0j7&sourceid=chrome&ie=UTF-8