IST 652 Final Project

Kevin Harmer

May 9th, 2022

1 Introduction

In the world of American sports, March has always been reserved for one thing: College Basketball. Year after year, 358 collegiate basketball teams spend their seasons competing for an opportunity just to play in the final NCAA tournament. But the real magic starts as the tournament begins when the nation watches basketball superpowers get upset by small colleges making their first tournament appearance. The chaos of ”March Madness” is unlike playoffs of any major sport.

After beginning in November, teams spend their seasons trying to prove their skill in order to get seeded (or a good seed for some teams) in the March tournament. Due to the importance of seeding and the recent growth of data analytics in today's society, some have acquired documented data analyzing each team's statistics to get a better understanding or their performance, bracketology, and tournament potential. In this case, kaggle.com will be used to examine the 2013-2019 tournament seeding and tournament results (details of data set found below). The data set contains information on each Division I Men's Basketball team for the 2013-2019 seasons (before playoffs).

- TEAM: name of team being analyzed
- CONF: name of conference the team plays for
- G: Games played
- W: Games won
- ADJOE: Adjusted Offensive Efficiency (Estimate of Average Points Scored per 100 Posessions)
- ADJDE: Adjusted Defensive Efficiency (Estimate of Average Points Allowed per 100 Posessions)
- BARTHAG: Power Rating (Chance of Beating an average D1 team)
- EFG_O: Field Goal Percentage
- EFG_D: Field Goal Percentage Allowed
- TOR: Turnover Percentage
- TORD: Steal Rate
- ORB: Offensive Rebound Rate
- DRB: Defensive Rebound Rate
- FTR: Free Throw Rate
- FTRD: Free Throw Rate Allowed
- 2P_O: Two-Point Shooting Percentage

- 2P_D: Two-Point Shooting Percentage Allowed
- 3P_O: Three-Point Shooting Percentage
- 3P_D: Three-Point Shooting Percentage Allowed
- ADJ_T: Adjusted Tempo (Posessions per 40 minutes)
- WAB: Wins Above Bubble (Wins against top teams)
- SEED: Seed in the NCAA March Madness Tournament
- POSTSEASON: Tournament Result (round lost in)
- YEAR: Year which team was recorded

Data set URL: https://www.kaggle.com/andrewsundberg/college-basketball-dataset?select=cbb21.csv

Using this data, this paper will address two major questions. First, from the statistics listed above, how can one predict a team's seeding? Although there is much that goes into a college basketball team's analytics, these major stats are good starting points for a team's talent. Second, do these stats along with some respective seeding indicate tournament success? Again, there are many more analytical methods for addressing this question, but this paper will focus on the statistics provided by kaggle.

2 Data Processing

The imported data from kaggle was mostly clean. The data set has 2455 rows (one for each team on any year from 2013-2019) characterized by 24 different columns. There were no duplicates recorded in the data set. Other than "SEED" and "POSTSEASON", each column did not have any NA or NULL values to worry about. "SEED" and "POSTSEASON" had nan values for the teams that did not make the tournament (as expected) and are adjusted necessarily in the analysis section. All of the observations were justified; there were no visible anomalies. To better fit upcoming algorithms, "SEED" and "POSTSEASON" are adjusted to numeric scores, based on 0-16 (0 no tournament bid, 1 being 16 seed, 2 being 15 seed, etc.) for SEED and 0-7 based on total tournament games before losing (round of 68 counted as 0.5).

3 Statistical Analysis

In a similar coding project with R instead of python, the statistics of the numeric variables in this data set were analyzed to determine which variables are best predictors of seeding and postseason results. Identical data processing was conducted to get SEEDSCORE and TGBL for numeric analysis. The methodology will not be described in this paper, only that the resulting significant variables were as follows:

1. SEEDSCORE:
    - G
    - W
    - ADJOE
    - ADJDE
    - BARTHAG
    - EFG_O
    - TOR

- ORB
- FTR
- WAB

2. TGBL:

- G
- W
- ADJOE
- ADJDE
- BARTHAG
- DRB
- WAB
- SEEDSCORE (not directly, but still included in analysis)

The specifics on these results can be found in the accompanying R Notebook file. By no means is it necessary to review these results for the understanding of this paper or accompanying python code. Instead, the purpose of this section is to recognize the reduction of variables and provide a supplement which explains the statistical reasoning for the variable reduction.

4 Machine Learning Algorithms

4.1 Adjusting Data Based on Statistics

As concluded from the statistics results, some variables are not needed in predicting team seeding or tournament results. As a result, the seeding data set dropped the columns: YEAR, EFG_D, TOR, DRB, FTRD, 2P_O, 2P_D, 3P_O, 3P_D, ADJ_T, SEEDSCORE, POSTSEASON and TGBL. The SEED column is extracted into a different data set with all original nan values converted to "NTA" (for No Tournament Appearance). The result data set is dealt with in a similar fashion by dropping the following columns: YEAR, EFG_O, EFG_D, TOR, TORD, ORB, FTR, FTRD, 2P_O, 2P_D, 3P_O, 3P_D, ADJ_T, SEED, and TGBL. POSTSEASON is similarly extracted into its own data set with "NTA" replacing all the nans from the original data set.

Once the data is organized, it is randomly split into testing and training sets, resulting in 8 total sets. Specifically, there are 4 training data sets corresponding to the seeding predictors and SEED along with the tournament results predictors and POSTSEASON. The other 4 data sets are the testing data sets which are similar in structure to the training data sets. In this trial, 25% of the data was split into testing data and the remaining 75% of the data into the training set. From this data and the help of scikit-learn, several machine learning algorithms were generated. This, along with last year's analysis, was the primary goal of the accompanying python code.

4.2 Support Vector Machines

To tune the support vector machine seeding model, a grid search was used. After testing through 3 folds, the program found a cross validation score of 0.826 (with parameters: C: 0.2, degree: 1, gamma: 5, kernel: 'poly', probability: True) after fitting the training set. After running the model, the testing set produced a 0.850 accuracy score.

The support vector machine tournament result model was tuned identically. After testing through 3 folds, the program found a cross validation score of 0.574 (with parameters: C: 0.2, degree: 1, gamma: 1, kernel: 'linear', probability: True) after fitting the training set. After running the model, the testing set produced a 0.639 accuracy score.

4.3 Decision Tree Classifier

The decision tree classifier did not use a tuner for the model. Instead, the seeding model used a max depth of 16 (for the 17 possible seeding combinations) and a minimum samples of 20 (to compensate for training set total seed minimum). For the tournament result model, the max depth was given as 7 with no minimum number of samples). The seeding model achieved an accuracy of 0.845 and the tournament result model saw an accuracy of 0.496.

4.4 K-Nearest Neighbors

The tuning procedure for K-Nearest Neighbors was similar to the SVM method. After testing through 3 folds, the seeding model found the best parameters as a manhattan metric with 5 neighbors. The cross-validation accuracy for the method was 0.812. After compiling the model, the test set produced an accuracy of 0.834.

Tuning the resulting model was very similar; through 3 folds, the results model found the best parameters as a manhattan metric with 5 neighbors. The cross-validation accuracy, however, for the method was 0.504, which later led to an accuracy of 0.529 for the test set.

4.5 Model Results

Each model was relatively consistent with each other. The seeding predictions consistently saw accuracy between 0.80 and 0.85, which is a pretty good prediction considering a) the weaker r-squared results from regression techniques (seed in preliminary results) and b) the non-implementation of automatic bids (bids from teams who win their conference tournament, regardless of their season statistics). Although more data and trials may improve the accuracy, the given analysis is promising for tournament seeding predictions. The SVM model proved to be the strong prediction tool in the seeding analysis phase.

Tournament results, on the other hand, were not predicted very accurately by the machine learning algorithms. With SVM at 0.64 and the other two coming in around 0.5 accuracy, these initial models are not great tools to predict the tournament results. A data set predicting all instances to lose in the round of 64 would achieve an accuracy around 0.5, meaning these methods are only slightly better than guessing. For future classification techniques, the models should be adjusted so that a) the data is treated as non-linear and b) the results are categorized, meaning there is exactly one team as a champion, exactly one runner up, exactly two other final four picks, and that procedure down to the round of 68 (per year). These adjustments would likely resolve the issues which led to no meaningful output.

5 Empirical Model

Because the machine learning classification algorithms struggled with the non-linearity of the tournament results, two regression models were developed to better predict results. After observing the scatter plots between Total Games Before Loss and each of the result variables, it was found that there was some rapidly decaying nature for the results, which is theoretically in line of the asymmetric distribution of tournament results. This, however, was not the case with Defensive Rebounds, which is shown below.

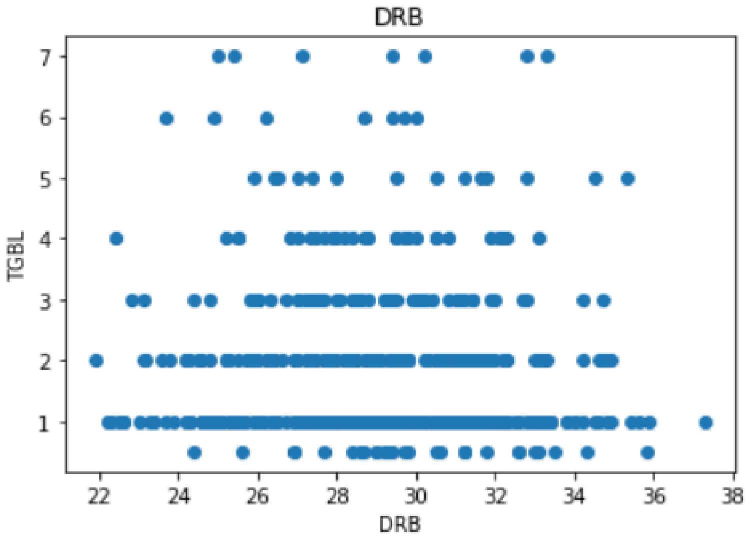

Figure 1: No strong relationship seen between DRB (Defensive Rebounds) and TGBL (Total Games Before Loss)

Due to the apparent lack of association between DRB and TGBL, DRB was excluded from analysis at this point. Otherwise, two mathematical models were designed to run regression analysis with, which are shown below.

$$TGBL_{inv} = \sum_{i=1}^{n_{var}} \frac{b_i}{x_i - x_{0i}} = \frac{b_1}{G - G_0} + \frac{b_2}{W - W_0} + \dots \tag{1}$$

$$TGBL_{exp} = \sum_{i=1}^{n_{var}} a_i \mathrm{e}^{-b_i \cdot (x_i - x_{0i})} = a_1 \mathrm{e}^{-b_1 \cdot (G - G_0)} + a_2 \mathrm{e}^{-b_2 \cdot (W - W_0)} + \dots \tag{2}$$

For the inverse equation, the b coefficients represent the amplitude of each term, while the var_0 represent the horizontal shift. The bounds for each parameter are expressed in the code to empirically fit the variable's distribution.

Similarly for the exponential equation, the a coefficients represent the amplitude of each term, the b coefficients represent the decay rate, and the var_0 represent the horizontal shift. The bounds, again, are set accordingly in the python code.

Using a curve fit function from scipy, regression models were constructed for the 2013-2019 data. The inverse function returned an r-squared value of 0.68 and the exponential function reached an r-squared of 0.77. These empirical functions, therefore, were much more successful in predicting tournament results than the previous classification algorithms. Although they are not perfect, they are successful in improving prediction methods for tournament results. Some of the relationships are depicted below in additional scatter plots.

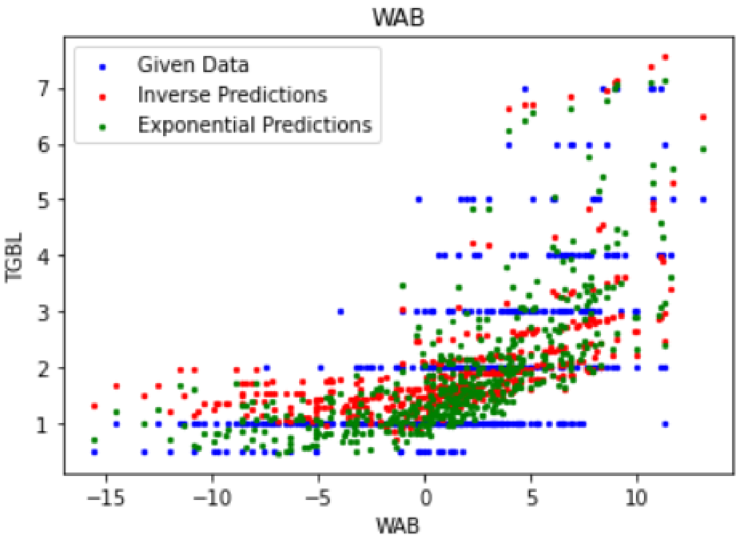

Figure 2: Total Games Before Losing based on Amount of Season Wins Above Bubble (with labeled inverse and exponential regression functions). Variability accounted for me other variables.

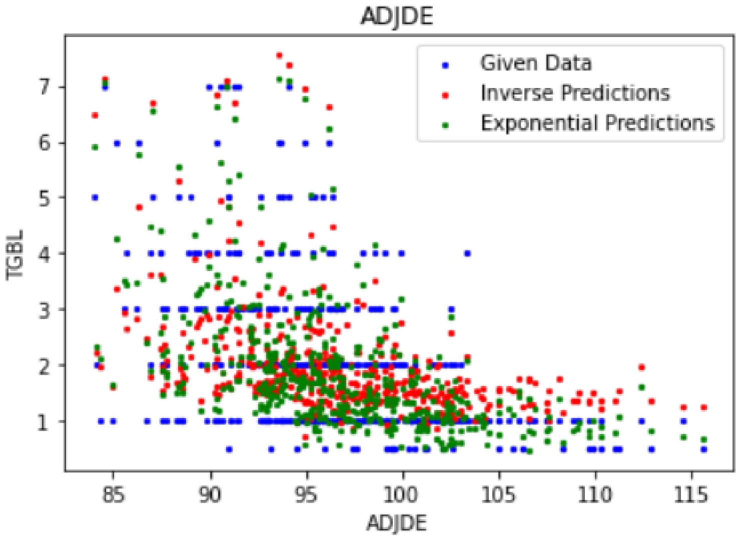

Figure 3: Total Games Before Losing based on Adjusted Defensive Efficiency (with labeled inverse and exponential regression functions). Variability accounted for me other variables.

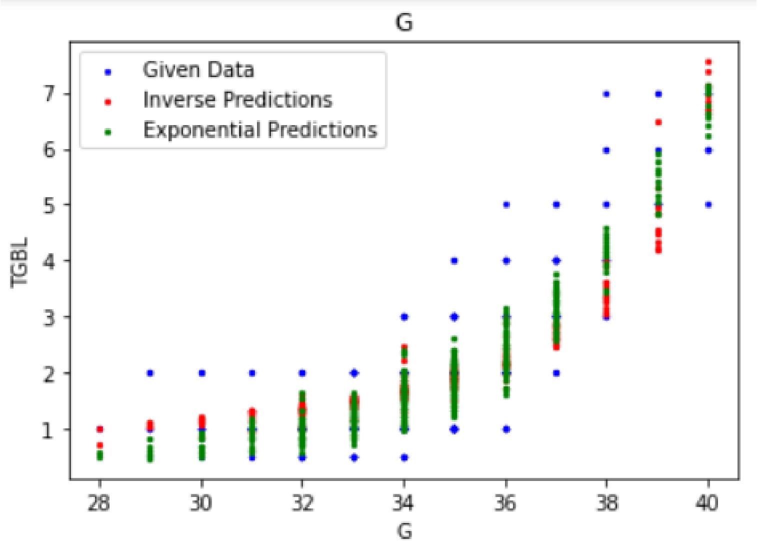

Figure 4: Total Games Before Losing based on Total Amount of Games (with labeled inverse and exponential regression functions). Variability accounted for me other variables.

6 Predicting the 2021 Tournament

Now that models have been constructed to predict tournament seeding and tournament results, they can be applied to future data. kaggle.com most recent version of college basketball data in this format was for the 2021 tournament, meaning these models on seeding and tournament results can be used for analysis on last year's tournament.

6.1 Tournament Seeding

Because the data set had given seeding, the same process was used to create a SEEDSCORE variable. The models from the previous steps produced the following results.

Model	Accuracy
SVM	0.810
Decision Tree	0.830
K-Nearest Neighbors	0.807

Table 1: Seeding Project Accuracy for the 2021 Season

Overall, the accuracy was similar to the model testing, meaning that these models are reliable enough to predict tournament seeding with up to 80% accuracy. To put this information into perspective, the seeds of each of the final four teams were extracted for 2021. The results are shown in the following table.

Team	Actual Seed	SVM Projected Seed	DTC Projected Seed	KNN Projected Seed
Baylor	1	2	1	1
Gonzaga	1	3	1	1
Houston	2	5	5	5
UCLA	11	10	10	10

Table 2: Tournament Seeding of 2021 Final Four Teams

6.2 Tournament Results

Different from the tournament seeding, the imported data did not have associated postseason results with it. As a result, analysis will be done solely with the regression models; the classification results will not be included due to the over predictions of losses in the round of 64. Baylor, the tournament champion, was projected by both models to lose in the Round of 32, which is certainly not good for tournament accuracy. The models, however, both projected Alabama (a 2 seed) as tournament champion. Alabama made it to the Elite 8, where they lost to UCLA (another good team). So although the models did do a relatively good job predicting tournament results, it failed to completely predict top teams.

7 Conclusion and Further Considerations

Overall, this data was sufficient in improving predictive measures when it came to tournament seeding and tournament results. Moving forward, the improvement of these models could be conducted in a few different ways.

First, association rule mining would be a great improvement for the tournament results. Specifically, there was only one 1-seed to ever lose in the first round of the NCAA tournament. Therefore, a rule selecting a 1 seed past the first round would be beneficial for the result. Incorporation of automatic bids in seeding could also be feasible with association rule mining.

Aside from association rule mining, dichotomous testing could be conducted. The probability of winning the championship or making the NCAA tournament, for instance, would be much easier to predict with the access binary predictive systems like logistic regression. Adapting the data in this fashion may improve the tournament result accuracy.

More data rarely hurts a model. Other factors like current coach, team experience rating, or even a match up predictor may benefit the models. Furthermore, the addition of previous tournaments may improve model training, thus making improving the accuracy in the long run. Even with the current data, natural language processing of the categorical data would be another variable for the models to predict; word vectors can be as valuable as other numeric data.