Predicting NCAA Basketball Tournament Success with Machine Learning Models

University of North Carolina at Chapel Hill

Akash Krishna, Austin Hale, Caleb Kang, and Daniel Chang

May 1, 2020

Abstract

In this paper, we attempt to predict the success of teams participating in the 2015-2019 NCAA Men's Basketball Tournaments using Machine Learning algorithms and find which statistical features give the highest likelihood for games won. By training our data on multiple regression models, we find the best model to be Random Forest Regression. We also find that Power Rating (BARTHAG) has the largest impact on how far a team makes it in the tournament. Using our test results, we predict Dayton as the winner of the 2020 NCAA Tournament that had been canceled due to COVID-19.

Keywords: Machine Learning, Random Forest, Linear Regression, Lasso Regression, Ridge Regression, Cross Validation.

1 Introduction

Each year, millions of people complete March Madness brackets in an attempt to create the first-ever perfect bracket. The odds of filling out a perfect bracket are a staggering 1 in 9.2 quintillion. It is approximately 27 times more likely to be hit by lightning three times in a year than to achieve a perfect bracket. Additionally, due to the COVID-19 outbreak, the 2020 NCAA Division I Men's Basketball Tournament was canceled this year for the safety of its players.

Our motivations to develop models on NCAA data involve our goal to find a way to maximize our chance at creating perfect bracket and to have a rough estimate as to who would have performed well in the 2020 NCAA tournament had it not been cancelled. Through our exploration we hope to answer the following questions: Can we use Machine Learning models to maximize the likelihood of creating the perfect bracket? Given the season stats for all NCAA teams in a year, which model will be an accurate indicator of predicting how far a team goes in the tournament? What features are the strongest indicators of team success in the NCAA tournament? In this paper, we examine four different regression models that each use 19 predicting variables to discover which NCAA team has the highest likelihood to win the most tournament games.

2 Pre-Processing the Data

We were given NCAA Men's Basketball data from years 2015-2019. The data provided included features such as: `TEAM`, `CONF` (Conference), `G` (Games Played), `W` (Games Won), `ADJOE` (Adjusted Offensive Efficiency), `ADJDE` (Adjusted Defensive Efficiency), `BARTHAG` (Power Rating), `EFG_O` (Effective Field Goal Percentage Shot), `EFG_D` (Effective Field Goal Percentage Allowed), `TOR` (Turnover Rate), `TORD` (Steal Rate), `ORB` (Offensive Rebound Percentage), `DRB` (Defensive Rebound Percentage), `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), `WAB` (Wins Above Bubble), `POSTSEASON`, `SEED`.

To run the data through linear models, we needed to make sure that only numeric features could be included in the dataset so we removed values like `TEAM` and `CONF`. We initially tried to include `CONF` as a feature and convert that column into many dummy columns where each new column would represent one of the NCAA Division I conferences, but including the dummy columns significantly reduced the accuracy of the model. To avoid any problems with multicollinearity, we even removed one of the dummy columns, but the accuracy of the resulting models was still significantly lower than the ones without the inclusion of the dummy columns; so, we didn't include the dummy columns. We also removed the `SEED` column, as it was highly correlated with `BARTHAG`.

The variable we are trying to predict is how many wins a team will get in the NCAA tournament. So when fitting the model we had to change the dependent variable, `POSTSEASON`, to indicate how many wins each team had in the tournament. Initially, the POSTSEASON feature had the values: R64, R32, S16, E8, F4, 2ND, CHAMPIONS, and we converted those respectively to 0, 1, 2, 3, 4, 5, 6, to indicate that, for example, a team that reached S16 won 2 games.

The last step before running the models, we centered and standardized all our data.

3 Models

3.1 Linear Regression

We run the linear regression model provided in the sci-kit learn library. The model was optimized on finding the best feature coefficient values that maximized the negative mean squared error. We fitted the data using the data from year 2015-2018 and tested the accuracy on the 2019 year data. we found the mean squared error value as 38% and the coefficient of determination (R^2) as 61%.

3.2 Ridge Regression

Ridge regression adds a ridge penalty to linear regression to reduce the standard errors. This technique is especially useful for analyzing data that suffer from multicollinearity, in which independent variables are correlated. We utilized scikit-learn's cross-validated grid-search estimator on six different α values: 1e-3, 1e-2, 1, 5, 10, 20. We determined that the best α value is 1e-3. With this α value, we found the mean squared error value as 38% and the coefficient of determination (R^2) as 61%.

3.3 Lasso Regression

Lasso regression is very similar to Ridge regression except the regularization term is in absolute value. This model sets high values of coefficients to zero if they are not relevant, unlike Ridge regression. We utilized scikit-learn's cross-validated grid-search estimator on six different α values: 1e-3, 1e-2, 1, 5, 10, 20. We determined that the best α value is 1e-3. With this α value, we found the mean squared error value as 38% and the coefficient of determination (R^2) as 62%. We noticed Lasso regression performed slightly better than Ridge regression as a result of lowering the high coefficients.

3.4 Random Forest Regression

The last model we used is Random Forest Regression, also provided by the ski-kit learn library. We trained on tournament teams in 2015-2018 and tested on 2019 data. For training, we had a random state number of 5. In addition, we used 20 as the number of minimum samples to split a node in the tree. These parameters were decided with the help from Josh Starmer from StatQuest. After fitting our data, we were able to get a prediction for the number of wins for each team in the 2019 tournament. This resulted in a MSE of 31% and an R^2 of 68%.

4 Summary

4.1 Best Models and Accuracy

Model	Hyperparameters	Mean Square Error	R^2
Linear Regression	N/A	0.38	61%
Ridge Regression	1e-3	0.38	61%
Lasso Regression	1e-3	0.38	62%
Random Forest Regression	100 Trees	0.31	68%

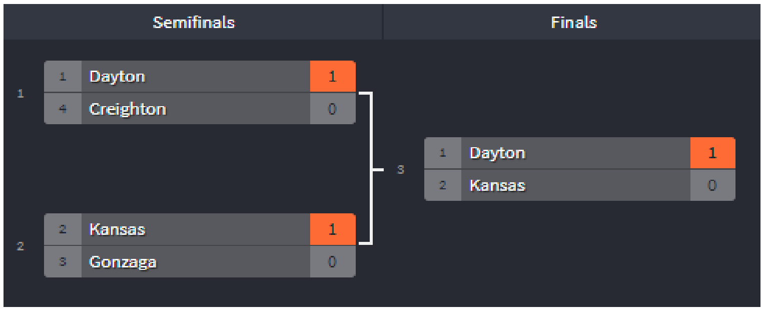
4.2 Feature Importance

Feature	Absolute Value of Feature Weight
BARTHAG (Power Rating)	0.633
Wins Above Bubble	0.078
Offensive Rebounds	0.043
Two-Point Shooting Percentage	0.035
Three-Point Shooting Percentage	0.023

The five features with the most impact on the prediction were Power Rating, Wins Above Bubble, Offensive Rebounds, Two-Point Shooting Percentage, Three-Point Shooting Percentage.

4.3 2020 Prediction

The Random Forest regression model turned out to be the model with the highest accuracy for us. Using this model we inputted the data from the NCAA regular season to develop the predictions for how far each team would go in the NCAA tournament had it not been cancelled. The model predicted that Dayton would have won the tournament with teams: Dayton, Kansas, Gonzaga, and Creighton rounding out the Final 4.



4.4 Github Code

For this project, we used Python and Jupyter Notebook to collaborate on the code. The link to the Github repository can be found using the following link: `https://github.com/COMP562-Machine-Learning-March-Madness/Final-Project`.