Christopher Tseng and Yibo Wang
CS 378
Final Project Report

Predictive Analysis of NCAA March Madness Results

Motivation

The goal of this project is to predict the winners of matchups between college men's basketball teams competing in the 2018 National Collegiate Athletic Association (NCAA)'s "March Madness" tournament. Since the early 1900s, it's been a lasting tradition between friends and family to see who can create the most accurate March Madness brackets. Figuring out the best way to accurately predict the bracket for that year has attracted interest because of the difficulty of correctly predicting the outcome of 64 teams playing a total of 63 games. Creating new methods for predicting sport game outcomes plays an important role in sports analytics and data science. There is even controversy due to the rise of platforms where one can bet money on who can pick the best team and score the most "points". In this field, there is historical precedent for using prediction models built from historical data and current team and player statistics to determine the optimal players and teams. We aim to contribute to this lasting basketball tradition by coming up with our own model to predict a tournament bracket.

Contributions:

- Yibo Wang: python code to pre-process basketball data from Kaggle, linear regression analysis and figures
- Chris Tseng: binary classification with neural net, decision tree, k nearest neighbor, analysis of upset teams using SVM, GamePredictor program

Related work

Historically, prediction models for basketball game outcomes have been important because they can factor many variables that go into the final outcome of any basketball game. The human memory has its limits in that it can only factor a certain limited number of variables at a time, so creating a prediction model is useful because it can handle so many correlations, trends, and assumptions.

In basketball, it has been determined that the future performance of a team will depend on many factors, including past performance, momentum in terms of win streaks, team standings, and external environmental circumstances, such as travel schedule, home-court advantage, off-season days, etc. Overall, these prediction models attempt to predict whether a team is weak or strong. Once this is determined, it gives a rough estimate of what teams are likely to appear on the top of the bracket and which are on the lower end.

There are a number of popular models used in predicting basketball game outcomes: K-means clustering, where players are clustered together based on a unique metric of player "strength", and from there should be able to see which teams should perform better as a result. Linear regression, where a team's rank for the current season is predicted based off of its rank from past seasons, or least angle regression (LARS) for data with many more features. Finally step-wise regression using the Akaike Information Criterion (AIC) has also been used to fit different regression models for the purpose of determining which features/statistics of a team can best explain a given outcome, such as the results of each game or its standing at the end of the season.

Approach

1. Decide which variables have the highest predictive power in determining which team is most likely to win a matchup in a given year. Given the wide variety of data provided, deducing these variables was accomplished using a number of machine learning techniques, including decision trees, linear regression, and clustering.
2. Use python and scikit-learn to plot linear regression graphs that show which factors have the highest correlation in determining which variables play the highest role in correctly predicting a tournament match-up, especially in upsets where a lower seeded team defeats a higher seeded team.
3. Create a consensus of those variables with different weights in order to construct a model that, when given two teams in a specific year, will be able to predict which team is most likely to win if they play against each other in the NCAA tournament.
4. Try to predict a NCAA March Madness tournament for a given year in which the prediction model factors in the variables most important to determining a team's chance to win.

Evaluation

We'll be using datasets from Kaggle's *"Google Cloud & NCAA ML Competition 2018-Men's"* machine learning competition. These datasets encompass a wide range of historical statistics every year since 1984, including the outcomes from past NCAA March Madness tournaments, team seedings, team points scored per game, team standings during regular season, region where a team is from, and more. The model will be evaluated based on accuracy, which is how many correct outcomes it can predict for a certain year's NCAA March Madness tournament after being trained using the given historical data. Specifically, better performing methods should predict the probability of two teams playing each other in the tournament for that year. The team that wins and moves on to the next stage, so in a sense, aim towards creating a "perfect bracket".

Due to the complexity of discerning how all this data interacts to produce relevant results, there is the lack of additional data not included in the given dataset that play a role in influencing outcomes. This problem continues to be the subject of keen interest by researchers recently. We understand that it is difficult for our team to be able to deliver a robust model capable of correctly predicting the outcome of a matchup between two teams in a given year. So it is even more unlikely to come close to figuring out the "perfect bracket". In this manner, our model can be evaluated in how accurately it can predict the winner of a game.

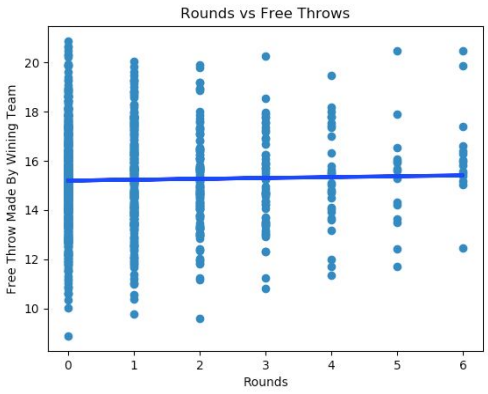
Pre-Processing

We'll be using datasets from Kaggle's *"Google Cloud & NCAA ML Competition 2018-Men's"* machine learning competition. These datasets have a range of statistics including the outcomes from past NCAA March Madness tournaments every year since 1984, team seedings, team points scored per game, team standings during regular season, and team. Our project focused on teams from 2003 onwards as that is when detailed game statistics for each NCAA Division team from its regular season and that season's March Madness tournament were collected. We also only included teams that made it into the Round of 64 and the full March Madness tournament. Data preprocessing was accomplished using Python, with a variety of statistics were calculated for each team that participated in March Madness since 2003. These statistics include the total number of home wins, away wins, neutral wins, as well as average regular season stats, including three pointers made and attempted, blocks, steals, rebounds, turnovers, point differential, etc.

Results

We performed preliminary analysis on the kaggle basketball dataset using linear regression. The linear regression line shows how strong the correlation is between a given season statistic - team wins, offensive rebounds, or point differential - and whether that statistic helps the basketball team advance in the March Madness tournament.

Example 1:

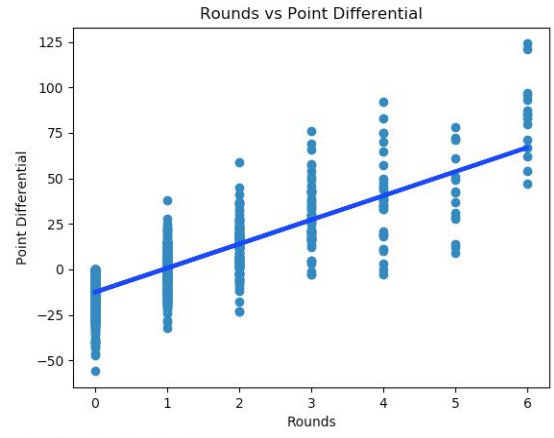


There are six rounds that can be won in total in March Madness which is why the x-axis ranges from 0 to 6. Thus a team plotted on the 6th x-axis point won the entire March Madness tournament for that season. In general the more rounds a team wins, the more successful its performance is in the March Madness tournament for that year.

In **Example 1**, the data plotted the number of rounds that different teams advanced in the tournament with the number of free throws made by the winning team. The slope of the linear regression line is 0.035. This indicates that there is almost no correlation between a basketball team advancing in the March Madness tournament based on the number of free throws it made.

Example 2:

In **Example 2**, the data plotted the number of rounds that different teams advanced in the tournament based on their point differential.



There is strong correlation since the slope of the linear regression line is 13.24. When tested with linear regression, the point differential had the strongest correlation out of all the basketball statistics tested - team wins, free throws made, offensive rebounds, etc.

The regular season point differential is the difference between the team's total number of points scored and opposing teams' total number of points scored. Wins in close games may not reflect a team's actual skill. Team that start the season with a high team standing may slow down towards the end and vice versa.

Categorical Classifiers

Initially, we labeled teams based on whether or not they reached the Final Four stage of the March Madness tournament. The classifiers were trained using regular season statistics for each team. An added note is that we excluded March Madness tournament game statistics for each team as there is a known correlation between features like seeding and likelihood of the likelihood of reaching the Final Four. In order to utilize Scikit-Learn implementation of classifier with default parameters: a neural network in the form of a multi-layer perceptron (MLPClassifier), a decision tree implemented using the CART algorithm (DecisionTreeClassifier), and K Nearest Neighbor (KNeighborsClassifier). Default parameters specified here:

Neural network: http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
Decision tree: http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
K Nearest Neighbor: http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

These classifiers were trained on aforementioned regular season statistics from 2003 to 2016, then tested on statistics from 2017. After training, here's the classifier accuracy after testing using the 2017 data. The neural network returned an accuracy of 94%, the decision tree returned an accuracy of 93%, and K Nearest Neighbor returned an accuracy of 96%. On first impression the accuracies appear high, but it became evident that this was due to the low number of "positive" classes, or number of teams that make it to the Final Four, which would only be 4 out of the 68 teams that participate in March Madness each season. Interestingly enough, it seems that all the classifiers were able to learn the correct proportion of teams that actually make it to the Final Four in general, which is around 4%. We tried expanding the number of elements in the positive class by including all Sweet Sixteen teams, or the 16 top teams instead of just 4 top teams. The accuracies returned from analysis of that data with those classes was 76%, 72%, 72% for the respective aforementioned classifiers. Upon closer inspection, though, as 25% of teams make it to the Sweet Sixteen (16/64=.25), the classifiers are actually predicting slightly above or in some cases worse than random. This points to a large issue at hand that would warrant further investigation.

Support Vector Machine

In order to determine which factors were the most important in determining if teams that were able to upset teams with a higher seed, I applied the Scikit Learn implementation of a support vector machine (SVM) to see if it would be better at classifying upset teams versus the team they upset given certain regular season statistics. Default parameters are specified here: http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
The classes were whether or not the team is an upset team or not. The factors are a single average regular season statistic. As can be seen, regular season point differential as a single factor leads to the overall highest accuracy out of all the different statistics.

Basketball Regular Season Statistic	SVM Accuracy
Team Win	0.5471698113207547
Team Loss	0.5849056603773585
Away Win	0.4716981132075472
Neutral Wins	0.5849056603773585
Field Goals Made	0.5849056603773585
Field Goals Attempted	0.4716981132075472
Three Pointers Attempted	0.5094339622641509
Assists	0.6226415094339622
Regular Season Point Differential	0.6792452830188679

Conclusion

The point differential shows how much comparative advantage one team has over another team when the teams face off. **Based on the data, knowing the point differential of the teams is the single biggest statistic in determining whether what team advances in that year's tournament.** As a result, we wrote a Python script called GamePredictor.py that takes three command line arguments: 2 team names and an year, and it prints out which team would win in a head to head game in that year. This game predictor solely uses regular point differential to determine which team would win. Teams that had a higher point differential for their regular season than the opposing team are predicted to be the winner.

Future Direction

Significant potential improvements can be made to the parameter tuning of the classification methods utilized in this project. We initially used the default parameters for all classifiers from Scikit-Learn, but due to time constraints were unable to perform enough experimentation with the different possible parameters to determine how changing those parameter values could lead to better classifier performance with the NCAA basketball data we have, and as such would be a necessary step for us to get better results. Furthermore, we also need to figure out how to properly attribute weights to the factors that have the highest explanatory power for predicting which team is more likely to win in a head to head matchup in the March Madness tournament. As of now, we're using regular season point differential as the main factor for determining which team would win versus another in a given year, which is how the GamePredictor program works, but there are definitely many other factors that would need to be taken into consideration for high performing model. Such work will help us achieve our final goal of fully predicting an entire March Madness bracket for a season and seeing how well it compares to the actual tournament results for that season, in line with the stated goal of the Kaggle competition.