NCAA March Madness Prediction with Machine Learning

Yvon Kadeoua

DSC 540: Advanced Machine Learning

Abstract

March Madness is the NCAA Men's basketball tournament played each spring in the United States. With billions of dollars at stake to reward the correct forecast, the tournament became an absorbing challenge for the fans who wants to predict the tournament outcomes. For this reason, many fans are turning to Machine Learning and statistics to forecast the tournament bracket. A tournament bracket is a tree diagram that represents the series of games played during a single elimination championship. In the study discussed in this paper, I attempt to discover if we can predict the tournament Bracket with Machine Learning. I used Random Forest, Gradient Boosting, Support Vector Machine and Neural Network to forecast March Madness Bracket and identify the features that have the most impact on each game outcome. Each method was evaluated based on their accuracy scores and bracket scores. Gradient Boosting returned the highest accuracy scores with a score of 0.96 for the Acc score and 0.99 for the AUC score. SVM constructed the highest bracket score, earning 250 points for the 2017 tournament.

Introduction

March Madness was created in 1939 by the National Association of Basketball Coaches, and it occurs every March in the United States of America. The tournament consists of six rounds of single elimination between 68 men's college basketball teams judged to be the best in their region (East, West, Midwest, South regions). Each team is ranked(seeded) within their region from 1 to 16. Before the first round begins, the eight teams with the lowest seed(ranks) are matchup to determine the last 4 teams in the field of 64. Then the 64 teams are matched in the first round to determine the second round (round of 32). The 32 teams are matchup to determine the sweet 16 (round of 16), then the elite 8, and finally the final 4. The bracket is structure so that the team with the highest seed in a region plays against the lowest seed, the second highest plays the second lowest, and so on until all participating teams are matched.

Every year, before the first round, millions of fans submit their predicted brackets to sporting networks such as ESPN or yahoo network for a high reward. In 2019 ESPN estimated that more than 17.2 million brackets were submitted to ESPN, and Warren Buffett offered 1 million dollars for life to whoever could predict a perfect sweet 16. No one has ever predicted a perfect bracket.

To encourage fans to forecast March Madness outcomes by using computational and mathematical sciences such as Statistics or Machine Learning, Kaggle.com has launched the NCAA March Madness Data Science competition. Several historical data about college games and teams are available on Kaggle.com as a result of the continued collaboration between Google Cloud and the NCAA. Kaggle competition is divided in 2 stages. In the first stage, competitors use historical data to build and test their models. In the second stage, competitors forecast the championship outcomes. However, couple questions are being raised such as can we use NCAA historical data and Machine Learning to predict March Madness outcomes. Based on probabilities can we predict what team will move forward in the tournament? How accurate can these statistical results be compared to the bracket?

The purpose of my research is to demonstrate that we can use machine learning to predict the NCAA March Madness bracket, and that feature selection and feature engineering are critical steps in forecasting a good bracket. To conduct my analysis, I compared the performance of 4 ensemble methods in forecasting each game outcome and selecting the features that have the most importance on the championship outcomes. Ensemble methods are machine learning algorithm that use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. My first method of choice is Random Forest because of its ability to use categorical data, its capability to determine feature importance, and the winner of the 2016 Kaggle's competition used Random Forest. Gradient Boosting and SVM were also chosen because of their capabilities to determine feature importance and their abilities to handle classification problems. Neural Network was also chosen because its capability to handle classification problems, and its track record in healthcare analytics.

Related work

Although there has been some successful previous work that were able to predict the tournament bracket scores with high accuracy, some models have suffered from poor feature engineering or poor combination of data. However, there is no agreement on which model or features will return the best bracket scores.

Zimmermann, Shi and Moorthy [1] calculated teams adjusted efficiencies along with the "Four Factors" (Effective field goal percentage, Turnover percentage, Offensive Rebound percentage and Free throw rate). The "Four Factors" were identified by Dean Oliver (Author of Basketball on paper) as influential factors in a team success. Zimmermann, Shi and Moorthy trained multiple classifiers such as Artificial Neural Network model, Naïve Bayes and Random Forest. They found out that the simplest classifier (Naïve Bayes in this analysis) performed remarkably well when compared to more complex models. They claim that the difference in performance came down to the used of attributes and how they were calculated, and they ultimately came down to the conclusion that feature selection would be more important than modeling in predicting the March Madness' bracket. They also note an important point that teams that are nationally attractive can choose many of their opponents themselves, and often have little consistency in the composition of teams from one season to another since stars players will quickly move on to professional sports. Therefore, unbalanced results and unrealistic match statistics are not uncommon.

Levandoski and Lobo [2] trained a Random Forest model along with Adaptive Boosting, K-Nearest Neighbors, Naïve Bayes, Neural Network, logistic regression and Support Vector based on rolling averages of selected statistics up to, but not including, the current game, that is recomputed after each game. They limited their rolling average of features to only include the 15 most recent games. Their selected statistics were team-level total statistics for each game (total field goals attempted, offensive rebounds, etc.). By calculating the features' rolling averages, they could track each team's performance before and after each game. This approach gave their model a satisfactory predictive power that led their Random Forest models to earned 900 points in bracket score for the 2017 March Madness Bracket. This result was above the average score of human participants, 715.4 (Levandoski and Lobo, 1). They arrived at the conclusion that machine learning models demonstrate greater overall accuracy in predicting NCAA tournament outcomes than the average human.

Shen, Gao, Wen and Magel [3] attempted to predict the 2015 and 2016 March Madness outcomes. They implemented the difference between the two opponents' teams' statistics such as the difference between free throws attempted, defensive rebounds, assists and turnovers, then they trained a Bayesian Logistic linear model with probability self-consistency, an SVM model, and a Random Forest model. They used a single and a double scoring system to assess their model accuracy. In the single scoring system, a model can earn up to 63 points for predicting the first round, and in the double scoring system the score double per rounds for each matchup they predict accurately so a total of 192 points. For the 2015 championship their SVM model performed the best in the double scoring system earning a score of 116/192 = 60.4% followed by Random Forest at 57.3% points. For the 2016 championship, their Bayesian model performed the best in the double scoring system earning a score of 52.08%. Based on their scoring system they concluded that ensemble methods would deliver a better performance than Bayes model using probability self-consistency in predicting March Madness Bracket.

Gumm, Barrett and Gongzhu [4] analyzed the correlation between variables and selected the variables that have the highest correlation with the target variable (Wins variable, 0 for losers, 1 for winners). They claim that tournament seedings are as good predictors as other variables. This claim was previously rejected by Levandoski and Lobo [2] that claim that tournament seedings would create bias towards teams (Levandoski and Lobo, 5). They developed an aggregate-based model to calculate the chance of a team winning over another. Their model was ranked in the top 15th percentile of Kaggle.com March Madness Competition.

Stoudt, Santana, Baumer [5] aimed to determine the most effective model and the most relevant data to predict the championship probabilities. They note that what maybe considered a favorite machine learning technique may have poor performance if trained with irrelevant data. They trained a logistic regression model, SVM, Naïve Bayes, k-nearest Neighbor, Decision Trees, Random Forest, Artificial Neural Network with raw scores from the past 5 seasons, the "4 factors", and teams and conferences rankings. Their model was ranked in the top 10 of the Kaggle competition for a portion of the competition and finished in the top 25 in the first stage. However, their model did not perform so well in the last stage. They arrived at the conclusion that their model performance could have been hindered by overfitting, overly complex model, or an overlooked trend in the data.

Based on the successful model developed by Levandoski and Lobo [2], I decided to use the rolling averages in my model, and I would also add the "four factors" in my model. I also fallowed the suggestion of Gumm, Barrett and Gongzhu [4] to use tournament seeding in my list of features.

Data Source

The data used in this project is from Kaggle.com; it contains records on regular seasons and NCAA tournaments from 1984-85 season to 2018-19 season. The files used in this project are Team.csv, Season.csv, RegularSeasonDetailsResults.csv, RegularSeasonCompactResults.csv, NCAATourneyCompactResults.csv, NCAATourneySeeds.csv, SampleSubmissionStage1.csv.

Team.csv describes the different college teams present in the dataset, it contains information on team names, teams ID, the first season the team was division 1 and the last season the team was division 1. Season.csv describes the different seasons included in the classical data, it has records on the year in which the tournament was played, the day the season started and the date it ended. The days have been labeled starting from day 0; the file also has records on the 4 regions from which each team is from, East, West, South and Midwest regions. RegularSeasonDetailsResults.csv provides team level statistics for each game for the two opponent teams, the statistics contains in this file are the field goal made, field goals attempts, three pointers made, three pointers attempt, free throws made, free throws attempts, offensive rebounds, defensive rebounds, assists, turnovers committed, steals, blocks, personal fouls committed. RegularSeasonCompactResults.csv describes the game by game results for many seasons of the historical data, it identifies the winning and losing teams ID, the winning and losing scores, the location of the winning team like if the team was a home team or a visiting team, the number of overtime periods in a game starting from 0 to higher. NCAATourneyCompactResults.csv describes the game by game NCAA tournament results for all seasons of factual data. NCAATourneySeeds.csv identifies the seeds(rank) for all teams in each NCAA tournament for all historical data, each team is seeded from 1-16 in their region; SampleSubmissionStage1.csv list every possible matchup between tournament teams for the past 5 years(seasons 2014,2017,2016,2017,2018), it identifies opponent teams ID and the season they played, it contains the predicted winning percentage for the team with the lower ID.

Data Processing

Team.csv, RegularSeasonDetailsResults.csv, RegularSeasonCompactResults.csv and NCAATourneySeeds.csv were merged into a single dataset by TeamID, TeamName, Season and Seeds. Two sets of data for game winners and losers were created along with their game's statistics. A binary variable (Result) has been created based on each game outcome, for winning team Result = 1, and for losing teams Result = 0. The variables contained in the winners and losers' dataset were renamed to match each other, so that they could be concatenated into a single dataset. Rolling averages were calculated for field goal made, field goals attempted, three pointers made, three pointers attempted, free throws made, free throws attempted, offensive rebounds, defensive rebounds, assists, turnovers committed, steals, blocks, personal fouls committed and game scores. Each rolling average has been calculated with a rolling window size of 2 so that the previous 2 games' performance averages are calculated before the third game. In this manner I can keep track of the team average performance before each game. I used the set of calculation proposed by Zimmermann, Shi and Moorthy [1] to calculate the “Four Factors”, and therefore the “Four Factors” were calculated as follow:

Effective Field goal percentage = (field goal made + (0.5*three pointers made))/free throws attempted.

Turnovers percentage = turnovers/ (0.96*(field goal attempted - offensive rebounds – turnovers + (0.475*free throws attempted))).

Offensive rebounds percentage = offensive rebounds / (offensive rebounds + opponents defensive rebounds).

Free throws rate = free throws attempted / field goals attempted.

As a result, a dataset of 23 variables and 33639 rows were created. The variables were the calculated rolling averages, the “Four Factors”, the TeamName, TeamID, the seeds, the season and the Result of the game (binary variable, 0 for losers and 1 for winners). However, only the rolling averages, the “Four Factors” and the seed were fed to the model.

Methodology and Results

Ensemble methods are capable of handling classification problems, provide the likelihood that an event occurred and identify features that have the most impact on the target variable. Among the machine learning ensemble methods, Random Forest, Gradient Boosting and SVM are used to to predict each game outcome (winner or loser), and the likelihood of a team to win or loose a game. Random Forest grows multiple trees and classify labels based on the vote of all trees. Among all features, a subset of features is randomly selected for each node, and the best split on the subset of features is chosen to split node. On the other hand, gradient boosting uses votes of each weak classifier, which is learned at every iteration, to generate a strong classifier. Gradient boosting uses regression tree models as weak classifiers and generates a strong model based on the notion of gradients in a way that a loss function is minimized. In SVM features are plotted in a p-dimensional Euclidian space where SVM finds the hyperplanes that separate the classes of interest based on the difference in features characteristics. Ensemble methods were compared to neural network which is a set of algorithms, modeled after the human brain, that are used to recognized patterns.

Each model was trained on the classification problem of deciding on a winner and a loser for each matchup and the likelihood for each game outcome. Gradient Boosting and Random Forest were trained with 100 trees, SVM was trained with the linear kernel, and Neural network was trained with the “adam” solver. At first each model was implemented without features selection, then feature selection was implemented to identify the features that have most impact on each game outcome and to improve predictions scores. 10 folds cross validation evaluated the performance of the learning models generated using Random Forest, Gradient Boosting, SVM and Neural Network. Each model performance was also evaluated based on their bracket scores. The bracket scoring system was implemented in the way that each model would receive 10 points for each matchup that occurred in the first 64, then the points doubled for each additional round.

Before Feature selection

Gradient Boosting, Neural Network, Random Forest and SVM classifiers returned a predictive accuracy of 0.94,0.96,0.93,0.96 respectively and an AUC score of 0.98, 0.99, 0.98, 0.99 respectively. For the bracket scoring system Gradient Boosting, Neural Network, Random Forest and SVM classifiers scores were 200, 190, 200, 190 respectively. Gradient Boosting accurately predicted 16 games in the first 64, and 2 games in the second round. For example, in the first 64 Gradient Boosting accurately predicted Gonzaga vs South Dakota with 73.22% probability for Gonzaga to win, then it accurately predicted Gonzaga to play against Northwestern in the second round. Neural Network accurately predicted 17 games in the first 64, and 1 game in the second round. For example, Neural network predicted Purdue vs Vermont with 54.27% for Purdue to win. Random Forest accurately predicted 16 games in the first round, and 2 games in the second round. For example, Random Forest accurately predicted Northwestern vs Vanderbilt with 50% probability for Northwestern to win in the first round, and predicted Gonzaga vs Northwestern in the second round. SVM accurately predicted 17 games in the first round and 1 game in the second round. For example, SVM accurately predicted New Mexico vs Baylor with 84% probability for New Mexico to win.

After Feature Selection

Gradient Boosting feature selection method selected Game score, Field Goal attempted, Defensive Rebounds, and Personal fouls moving averages as the most import features in predicting games outcomes. Random forest selected Score moving average, Effective Field goal percentage, Field Goal Attempted, Defensive Rebounds, Assist and Personal Fouls moving averages as the most important features. SVM selected Score moving average, Effective Field goal percentage, Field Goal Attempted, Defensive Rebounds, moving averages, Offensive rebounds percentage, Offensive Rebound moving average, Defensive Rebound, turnovers committed, steals moving averages as the most important features. Gradient Boosting, Neural Network, Random Forest and SVM returned an Acc score of 0.92, 0.92, 0.92 and 0.94 respectively and an AUC score of 0.97, 0.97, 0.96 and 0.99 respectively. Gradient Boosting, Neural Network, Random Forest and SVM scored 200 points, 190 points, 200 points and 250 points respectively in the bracket scoring system. Gradient Boosting accurately predicted 16 games in the first round, and 2 games in the second round. Neural Network predicted 17 games in the first round, and 1 game in the second round. Random Forest predicted 16 games in the first round and 2 games in the second round. SVM predicted 17 games in the first round, 2 games in the second round and 1 game in the 3rd round. SVM accurate prediction is illustrated bellow along with their probabilities highlighted in blue.

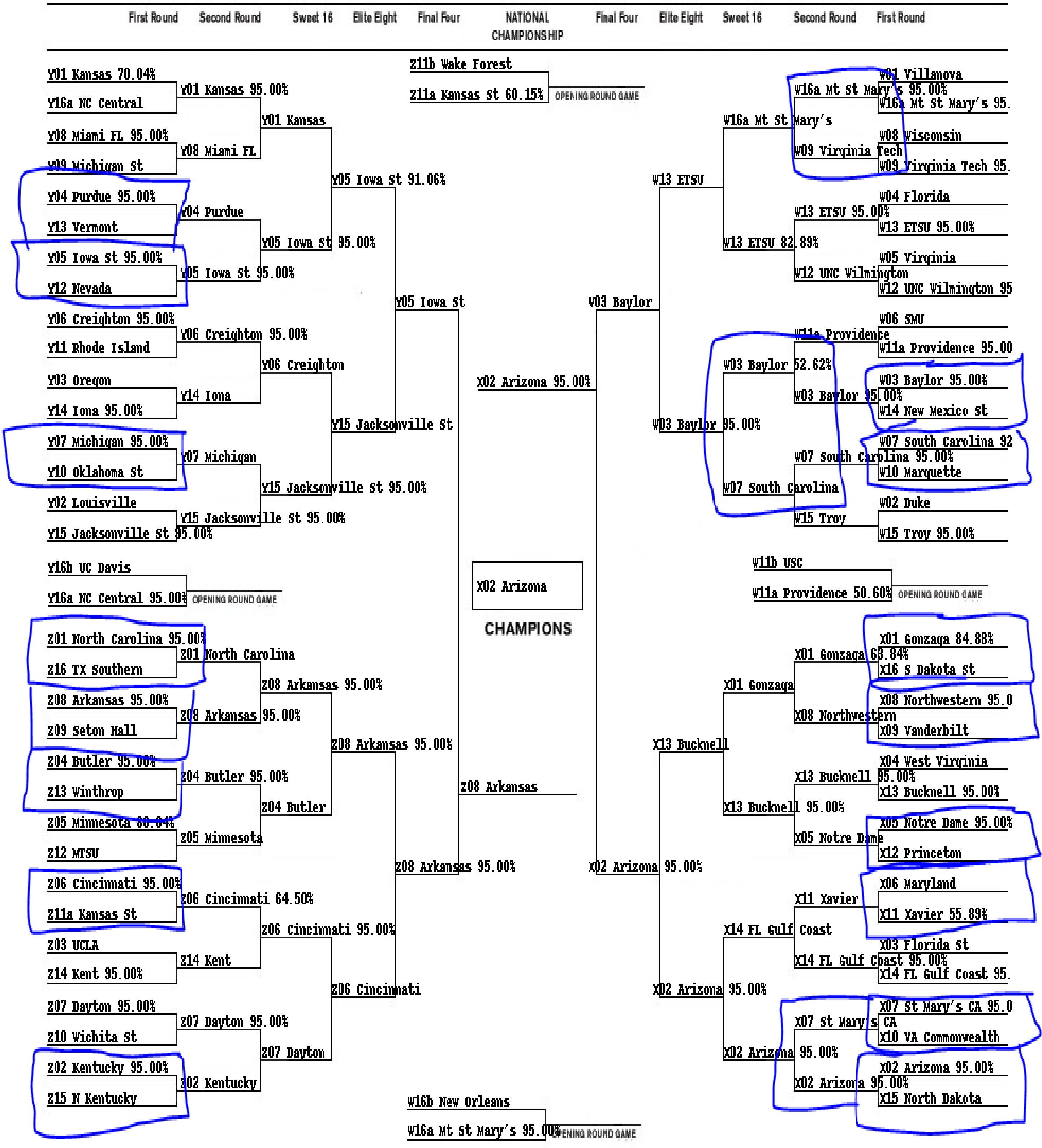

Table 1: SVM Bracket

Discussion and Conclusion

Although my model bracket score fell below the human average (715.4) stated by Levandoski and Lobo [2], most models scored an average of 16 games in the first 64. This is a confirmation that we can use Machine Learning to predict March Madness bracket. We can also use probabilities to predict the game outcome as SVM bracket shows that St Mary's CA would defeat VA commonwealth with 95% probability, and St Mary did defeat VA commonwealth. Although I used the moving averages suggested by Levandoski and Lobo [2] and the "4 factors", my model did not perform as well as theirs. One reason of this poor performance could be how I calculated my features. I previously used Raw scores to predict the game outcome and Neural Network scored higher than SVM with a score of 270 in the bracket scoring system. As noted by Zimmermann, Shi and Moorthy [1] how we calculate our variables is more important than modeling in predicting the championship bracket. It is also important to note that Levandoski and Lobo [2] used game location as a feature in their model. This implies that additional features can be used in my model. Kaggle.com has provided other variables such as team coaches, team conferences etc., then additional variables could use in the model. Hence, feature selection would be an important step in determining which variable has the most impact on the game outcome. As a result of my analysis, I came up with the conclusion that feature engineering and feature selection would be an important step in predicting the tournament outcomes.

Future Work

I would like to explore variables that have the highest correlation with winning and consider variables such as game location, team coaches or the number of stars players in a team. My assumption is that if a team performed well outside of its city, then the fact that the team plays as a visiting team during March Madness would have a positive effect on their performance. On the other hand, if a team performed poorly outside of its city, then playing as a visiting team during March Madness may have a negative effect on their performance. In addition, having a coach that have track records of winning championships would also influence the team performance. I also believe that the number of stars players in a team has a great impact on the team performance because if the number of stars players within a team A outnumber the number of stars players in team B, then team A has higher chances of winning against team B. I would also like to calculate the team adjusted performances which is the team performance compared to the national average. In this manner I could rank teams as having a performance above or below the national average, and therefore a team with adjusted performance above national average would have higher chances to win against a team with lower performance.

References

[1] Z. Shi, S. Moorthy, A. Zimmermann, “Predicting NCAAB match outcomes using ML techniques – some results and lessons learned”, October 2017.

[2] Andrew Lavandoski and Jonathan Lobo, “Predicting the Men’s Basketball Tournament with Machine Learning”, CS 2750: Machine Learning, April 2017.

[3] Gang Shen, Di Gao, Qian Wen, Rhonda Magel, “Predicting Results of March Madness using three different methods”, Department of Statistics North Dakota State University, Fargo, ND, 21 May 2016

[4] Jordan Gumm, Andrew Barrett, Gongzhu Hu, “A Machine Learning Strategy for Predicting March Madness Winners”, Department of Computer Science, Central Michigan University, June 2015

[5] Sara Stout, Loren Santana, Ben Baumer, “In Pursuit of Perfection: An Ensemble Method for Predicting March Madness Match-Up Probabilities”, Smith College, University of Utah