March Madness Men's Competition 2018

Matthew Bierman
The University of Texas at Dallas
matthew.bierman@utdallas.edu

I. Introduction and Problem Description

March Madness, an annual college basketball tournament hosted by the NCAA, is one of the most anticipated sporting events each year. Sixty-eight of the best teams in the nation face off in a single elimination tournament until only one team remains. The sixty-eight teams compete in sixty-seven games over the span of twenty days, from March 13 to April 2. Due to the unpredictable nature of sports, there is no machine learning algorithm or statistical analysis that can perfectly predict the outcome of every game, regardless of the sport. This unpredictability mixed with skill and pushing the limits of the human body draws hundreds of millions of people from around the world to watch sporting events like March Madness. From the first March Madness tournament in 1939, no one has ever correctly predicted every single game of that year's tournament. The odds of predicting a perfect bracket is 1 in 9.2 quintillion [1], so it is not likely that it will happen any time soon. However, these scary odds do not stop people from trying. People often host contests with coworkers, family members, or friends to see who can create the best bracket, which is the tournament bracket with the most correct predictions.

With the increase in public data availability and machine learning/statistical learning use cases over the past few decades, many people have applied machine learning to predicting the March Madness tournament and other sporting events. This year, Kaggle and Google Cloud teamed up to host a prediction competition for the March Madness Men's Competition and provided a plethora of data to aid in competitors' prediction efforts [2]. In addition to competing in the Kaggle competition, I decided to use my predictions to fill out a bracket and participate in the NCAA March Madness Bracket Challenge sponsored by Capital One [3], since the two competitions used different scoring metrics, as described in section V. For both metrics, I used Logistic Regression as my machine learning algorithm to compute the probability of the first team winning when given two teams. Initially, I thought about also using Naïve Bayes, but Naïve Bayes works better with smaller datasets and as a given dataset grows larger, Logistic Regression will perform better. Since I will be dealing with relatively large datasets in this project, I decided Logistic Regression would be a better fit in this case.

II. Related Work

Many statisticians and basketball columnists, typically around one week before the beginning of the tournament, will announce their predicted team rankings, which is an ordered list of which teams are most likely to win the tournament. Ken Pomeroy, author of 2018 Pomeroy Ratings [4], created a ranking which includes rankings for all 351 Division I NCAA teams, including features like strength of schedule and luck. FiveThirtyEight, an online blog that features statistical analysis on politics, science, health, as well as sports, created a bracket with probabilities for which team was most likely to win each game [5]. Due to the popularity of FiveThirtyEight and its high predictive accuracy in other sports, I decided to compare my results with the results of this major publication, as described in section VI. Furthermore, students at Katholieke Universiteit Leuven in Belgium claimed to have seen a "glass ceiling" effect on their NCAA Men's Basketball predictions, with their predication accuracy capping at 74% – 75% [6]. Also described in section VI, I decided to compare my results with this "glass ceiling" claim.

III. Dataset Description

With Google Cloud joining Kaggle in hosting this March Madness prediction competition, a large amount of data was available for use during the competition, fifty-two datasets in total. For this project, I used the following nine datasets: Conferences, NCAA Tourney Compact Results, NCAA Tourney Seeds, Regular Season Compact Results, Regular Season Detailed Results, Sample Submission Stage 2, Team Coaches, Team Conferences, and Teams. From these datasets, I created my own datasets called Full Team Data, containing all relevant data for each Division I team, which were extracted from the datasets provided by Kaggle. I created nine Full Team Data datasets in total, one for each year from 2010 to 2018. The 2010 to 2017 data was used for training and the 2018 data was used for creating the 2018 predictions. The features in the Full Team Data datasets are as follows:

- TeamID — Unique ID of team, since team names can be written multiple ways (e.g. Nevada and UNR are the same team)
- TeamName — Name of team
- ConfAbbrev — Abbreviation of the conference name the team belongs to
- ConfFullName — Full name of the conference the team belongs to
- CoachName — Name of coach
- TotalCoachYears — Number of years the coach has been an NCAA coach
- CurrCoachYears — Number of years the coach has coached his current team
- Seed — Ranking of team, 1 through 16 (20 for unseeded, as described in

section IV)

- Wins — Number of wins in regular season (Number of wins = number of OT wins + number of non-OT wins)
- WinsOT — Number of wins in overtime in regular season
- Losses — Number of losses in regular season (Number of losses = number of OT losses + number of non-OT losses)
- LossesOT — Number of losses in overtime in regular season
- HomeWins — Number of wins at home in regular season
- HomeLosses — Number of losses at home in regular season
- AwayWins — Number of wins at away in regular season
- AwayLosses — Number of losses at away in regular season
- NeutralWins — Number of wins at neutral in regular season
- NeutralLosses — Number of losses at neutral in regular season
- NeutralWinsPct — Percentage of neutral wins = neutral wins / overall wins
- NeutralLossesPct — Percentage of neutral losses = neutral losses / overall losses
- AvgScore — Average points scored per game
- AvgFGM — Average number of field goals made per game
- AvgFGA — Average number of field goals attempted per game
- AvgFGM2 — Average number of 2-point field goals made per game (AvgFGM2 = AvgFGM – AvgFGM3)
- AvgFGA2 — Average number of 2-point field goals attempted per game (AvgFGA2 = AvgFGA – AvgFGA3)
- AvgFGM3 — Average number of 3-point field goals made per game
- AvgFGA3 — Average number of 3-point field goals attempted per game
- AvgFTM — Average number of free throws made per game
- AvgFTA — Average number of free throws attempted per game
- AvgOR — Average number of offensive rebounds per game
- AvgDR — Average number of defensive rebounds per game
- AvgAst — Average number of assists per game
- AvgTO — Average number of turnovers per game
- AvgStl — Average number of steals per game
- AvgBlk — Average number of blocks per game
- AvgPF — Average number of personal fouls committed per game

IV. Pre-Processing Techniques

Some of the pre-processing for this project included extracting data from the datasets provided by Kaggle and combining it into one dataset (one per year) called Full Team Data, as described in section III. From the Seed dataset, I removed the conference prefix of the seed rank and added the tournament seed of that year to each team in Full Team Data. For unranked teams, which means that the team was not invited to the March Madness tournament, I set the seed to 20. While in most cases you can just replace NA values with 0, giving unranked teams a seed of 0 means the difference in seeds between an unranked team and a #1 seed team is one, while the difference in seeds between an unranked team and a #16 seed team is sixteen. This would lead to highly inaccurate results, so an unranked team must have a seed value lower than 16. From the Coaches dataset, I removed all coaches that were not coaching that year and reduced the dataset to get the total number of years each coach had coached for and the total number of years each coach had coached for their current team. From the Conferences and Conferences Full Names datasets, I matched each conference with its full name and added both to all teams in those respective conferences. From the Regular Season Detailed Results dataset, I reduced each game to count the number of wins, overtime wins, losses, overtime losses, home wins and losses, away wins and losses, and neutral wins and losses for each team. A "neutral" game is a game that is played at a location where neither team has home field advantage. Since every game of the March Madness tournament is at a neutral location, I used the neutral wins and losses values to compute the percentage of wins and losses in neutral games. I also computed the average number of field goals made and attempted, two-point field goals made and attempted, three-point field goals made and attempted, free throws made and attempted, offensive and defensive rebounds, assists, turnovers, steals, blocks, and personal fouls for each team over all games in the regular season from the Regular Season Detailed Results dataset. All of these features were merged into one dataset, Full Team Data, for every Division I team for every year from 2010 to 2018.

When creating the training and testing datasets for training the Logistic Regression model, each example requires two teams and the result that will be predicted must be a decimal value between 0 and 1. A value close to 0 means the second team is predicted to win and a value close to 1 means the first team is predicted to win. Since both teams must be in the same example, there will be every feature from the Full Team Data dataset for both teams. I combined each feature of both teams by taking the difference in values. For example, the difference in seed rankings, titled SeedDiff, is equal to the seed ranking of the first team minus the seed ranking of the second team. These "difference" features were used in the Logistic Regression model to predict the probability of which team would win in any given game.

V. Proposed Solution and Methods

As mentioned in section I, I chose to use a Logistic Regression model with accuracy and log loss as its metrics. I decided not to use Naïve Bayes for two reasons. First, Naïve Bayes works best with small datasets, and second, Naïve Bayes assumes all features are conditionally independent, which would not work well since features like field goals made and 2-point field goals made are strongly dependent, along with many others.

Logistic regression uses the logit function, as shown below, to predict an outcome between 0 and 1. Each X is a feature used in predicting the outcome [7].

$$Ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

This March Madness prediction project is unique compared to other machine learning and big data projects in that the training and testing datasets could not be divided up randomly. Instead, the training data would come from regular season results and the testing data would come from the March Madness postseason tournament. By doing this, the 2018 regular season data would be the input for the Logistic Regression model that would predict the March Madness 2018 tournament results.

With all machine learning models, it is important to test various parameters and features to find what works best for a given problem, since no single algorithm works best for every problem, as explained by the No Free Lunch theorem. For my March Madness prediction, I created six different Logistic Regression models, each of which were tested on every year from 2010 to 2017. The differences between the models are described as follows:

- **Model 1**: Using all features as predictors with the regularization parameter at its default value
- **Model 2**: Using all features with a larger regularization parameter value
- **Model 3**: Using all features with a smaller regularization parameter value
- **Model 4**: Using only features that were most significant from every year in Model 1 with the default regularization parameter value
- **Model 5**: Using only difference in seed ranking between the two teams with the default regularization parameter value
- **Model 6**: Using multiple features with interaction effects with the default regularization parameter value

In model 4, I selected six features that had strong significance values for models 1, 2, and 3 when using input data from every year from 2010 to 2017. Of the twenty-five features used in models 1, 2, and 3, the six that were selected include: seed ranking, total number of years the current coach has been an NCAA coach, number of wins, number of losses at a neutral location, average number of defensive rebounds, and average number of blocks. The interaction effects in model 6 include the interaction between seed ranking, number of wins, and number of wins at a neutral location, the interaction between seed ranking, number of losses, and number of wins at a neutral location, and the interaction between number of years the current coach has been coaching for that team, number of losses, and number of wins at a neutral location.

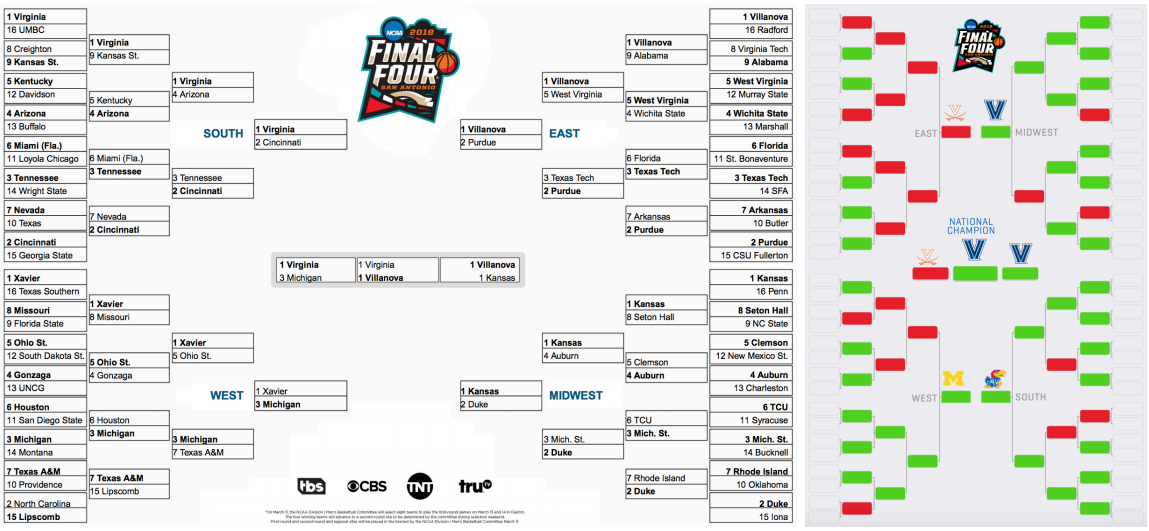
After testing models 1, 2 and 3 on data from every year from 2010 to 2017, I found that the regularization parameter had no significant impact on accuracy nor log loss, so I used the default regularization parameter value for the last three models. In section VI, I will discuss the results from each Logistic Regression model and explain the models I chose for predicting the 2018 March Madness tournament.

VI. Experimental Results and Analysis

Below are two graphs that display the accuracy and log loss of each Logistic Regression model for each year from 2010 to 2017. One interesting attribute from the graphs I noticed was that for log loss graph, each model followed the same general trend and they all seemed to converge towards the same value for 2017. However, with the accuracy graph, the models also followed the same general trend until 2017, where some models had a sharp increase (models 5 and 6) in accuracy while others had a sharp decrease (models 1 and 2).

Since models 5 and 6 had abnormally high accuracies in 2017 but roughly showed signs of increase from 2010, I decide to pick those models for my Kaggle and NCAA Bracket Challenge submissions. While log loss was also a factor in deciding which model to use, the log loss graph showed that each model had similar log loss values, which wasn't as drastic of a difference compared to the models when looking at accuracy.

After choosing models 5 and 6, I used that model on the 2018 regular season data to predict the 2018 March Madness tournament. For the Kaggle competition, I submitted my predictions from models 5 and 6. For the NCAA Bracket Challenge, I submitted my predictions from model 6 and FiveThirtyEight's predictions, so I could compare my results. The left image below shows my predictions (the predicted winner is in bold) and the right image shows the accuracy of my predictions (green means correct, red means incorrect) for the NCAA Bracket Challenge.



For the NCAA Bracket Challenge, I correctly predicted 41 out of 63 games, which is an accuracy of 65.08%. One key fact to point out is I correctly predicted the overall winning team of the tournament, Villanova, which had a 1.47% chance of occurring. These predictions placed my bracket in 15,914th place out of over 1.5 million brackets, which is in the top 1%. For comparison, FiveThirtyEight correctly predicted 37 out of 63 games, which is an accuracy of 58.73% and placed them in 126,564th place (top 6%). For the Kaggle prediction competition, I ended with a log loss of 0.597115 for model 5 and 0.612063 for model 6, which is in the top 30% and 50%, respectively.

While these accuracies and log losses weren't as impressive as the values in the training results, they did reinforce the idea of a "glass ceiling effect", where it becomes nearly impossible to maintain a consistent accuracy above 75% [6]. While an accuracy of 80%

in the training results seemed promising, the rest of the results showed that it won't stay that high for future years.

VII. Conclusion

This year's March Madness tournament was full of surprises, for instance, Loyola Chicago reaching the Final Four for the first time since 1963, UMBC beating #1 seed Virginia in only their second time participating in the tournament, Nevada reaching the Sweet Sixteen for the second time, and much more. Whether it is skill or simply luck, the nature of basketball, and all sports, makes it difficult to predict outcomes through data alone, as shown by the results in this paper. In the future, I plan to use different machine learning algorithms to compare the effectiveness of predicting games in college basketball and other sports.

VIII. References

1. Robert J. Szczerba. Bracketology 101: Picking A Perfect Bracket Is Actually Easier Than You Think https://www.forbes.com/sites/robertszczerba/2015/03/17/bracketology-101-picking-a-perfect-bracket-is-actually-easier-than-you-think/#5e8641162abd
2. Kaggle March Madness Prediction Competition https://www.kaggle.com/c/mens-machine-learning-competition-2018
3. NCAA Bracket Challenge https://bracketchallenge.ncaa.com/
4. Ken Pomeroy https://kenpom.com/
5. FiveThirtyEight's March Madness Predictions https://projects.fivethirtyeight.com/2018-march-madness-predictions/
6. Zifan Shi, Sruthi Moorthy, Albrecht Zimmermann. Predicting NCAAB match outcomes using ML techniques – some results and lessons learned http://www.ecmlpkdd2013.org/wp-content/uploads/2013/09/mlsa13_submission_12.pdf
7. Logistic Regression https://www.theanalysisfactor.com/what-is-logit-function/