# Predicting March Madness Using Machine Learning Techniques

Allen Zeng
Northwestern University
AllenZeng2014@u.northwestern.edu

Rich Chang
Northwestern University
RichardChang2014@u.northwestern.edu

## 1. Overview

March Madness is a NCAA Division I Basketball tournament that was founded in 1939 and has been played annually since. The tournament traditionally runs from the middle of March to the end of April and features the top 68 teams from around the country.

Filling out a March Madness bracket has been somewhat of an American tradition, with a history of high offers for anyone who can fill out a perfect bracket. In 2012, Fox Sports offered $1 million to anyone who could do it, and for this upcoming season, Warren Buffett offered $1 billion. Sadly, there has been no documented case of a perfect bracket in the history of the tournament.

## 2. Project Goal

Our goal for this project is to accurately predict a March Madness bracket for the upcoming 2014 tournament. We will use a number of machine learning classifier models and attempt to predict the winning teams by identifying the characteristics that determine a team's success in the tournament.

To construct our bracket, we will ask the following questions in our experiments:

- Which are the most important statistics about a team that determine its likelihood to win a game?
- Which classifiers will give us a model with stronger performance - decision tree, logistic regression, or another classifier?
- How accurately can we predict the outcome of a game given each team's season statistics? Is it as least better than chance (50%)?

## 3. Data

Working with sports data is both a blessing and a curse - there are a countless number of websites available that provide data but it is not always provided in a structured format that fits the user's specific needs. We surveyed a number of available sports data sites but we ultimately opted for Statsheet because the data was provided in a relatively structured format that made it straightforward to scrape.

Our dataset consists of information from Statsheet's NCAAB (http://statsheet.com/mcb) for the 2012-2013 season. The set contains all 351 NCAAB D1 teams, their conference schedule games (a total of 3207 games), and their season long statistics (54 for each team). The team statistics are a combination of binary, continuous and discrete variables that we are using as a feature set to represent each individual team. A sample of the types of features that we collected includes, but is not limited to: winning pct, possessions, floor pct, steals, turnovers, blocks, and fouls.

## 4. Approach

In order to conduct our experiments, we needed to create instance representing the games that occurred from the data that we collected for the 2012-2013 season.

### 4.1 Creating Training Instances

To create an instance, we looked at our game data, which was formatted in the following format:

- home team, away team, outcome for the home game
-

Using this information, we queried our data to find the season statistics for each team, giving us a set of numeric values describing each team. Next in order to model the actual game, we took the difference of

the stats between the respective teams (home team stats - away team stats) and appended the outcome of the game to it. This gave us a numerical representation of a game that we could use as a training example to feed into a classifier model.

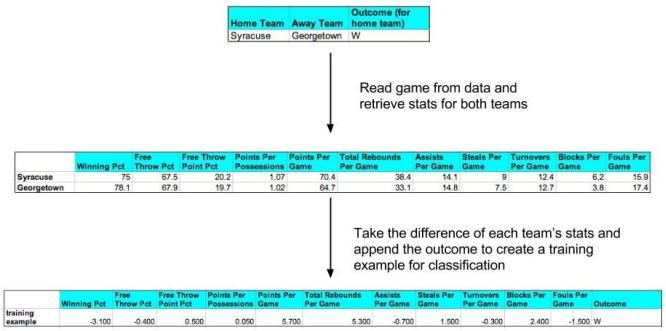


*Figure 1.* Outline of how we constructed our training examples (not all of our features are included in this diagram).

Our rationale behind modeling a game in this format was because it was a reasonable (and simple) way to represent a matchup between two teams - it is the difference in team skill (which we are representing as their season statistics) that ultimately decides which team wins in a basketball game. Intuitively, modeling a game in this way makes sense - a team that has a consistently higher average points per game is more likely to defeat a team that has a much lower average points per game.

### 4.2 Features

In Appendix A, we've included the full list of features. In this list, one can see that there are many repetitive statistics. For example, one can calculate the 3-pt percentage if the amount of 3-pt shots taken and 3-pt shots made are known. In this list of features, all 3 statistics are known, so we removed the 3-pt shots made. This occurs for many other statistics for field goal shooting, free throw shootings, and more.

In addition, we did try to "prune" our model by removing features and seeing if it increased the accuracy. However, we could not find a single feature that actually increased the accuracy. Even something as minor as disqualifications per season, when removed, decreased the accuracy of our model by 2%. Therefore, we kept all the non-repetitive features in our model.

### 4.3 Classifiers

The classifiers that we considered included:

- ZeroR: To give us a baseline to compare other classifiers, we used ZeroR to predict the outcome of the game.
- J48: We chose to use to use j48 because it was an intuitive way to interpret the features and understand the calculated relationships that result from a decision tree.
- Logistic Regression: Another useful classifier model that is effective when working with numeric attributes and understanding the relationships between them.
- Other classifier models implemented in Weka

### 4.4 Testing and Evaluation

To test the accuracy of our models, we used the games from the 2012-2013 March Madness tournament and generated test instances following the same method as with the training examples.

Once we had our training examples and test data, we conducted our experiment as follows:

1. Establish a baseline accuracy using the ZeroR classifier model in Weka
2. Explore other classifier models in Weka and evaluate their performance
    a. we looked at both the accuracies on cross validation (10-fold) and the test set
3. Further refined our features (eliminate redundant attributes if applicable) and classifier model selection

## 5. Results

After trying different models, the Naive Bayes classifier had the best percentage on 2013 March Madness tournament, predicting 71.6% of the games correctly. Naive Bayes is an optimal choice for our classifier not only because it had the best percentage, but because all of our features are numeric, the nature of how Naive Bayes is calculated synchronizes with our data. Naive Bayes relies on the probabilities calculated through the mean and standard deviation of a feature.

| Classifier | Cross validation | On Test Set |
|---|---|---|
| J48 | 71.6% | 55.22% |
| ZeroR | 59% | 56.7% |
| DecisionTable | 71.5% | 62.6% |
| BayesLogisticRegression | 71.8% | 64.2% |
| ADABoost | 72.4% | 64.2% |
| SimpleLogistic | 73.4% | 65.7% |
| NaiveBayes | 70% | 71.6% |

Table 1. The different classifiers we tested and their results.

The problem with decision trees is that with numeric values, they tend to overfit our data by splitting multiple times on numeric values. Therefore, we saw a very high success rate for our J48 tree (72%), but a rate that was worse than our ZeroR rate on the actual 2013 March Madness tournament (55%).

Once we chose the Naive Bayes classifier, we attempted to use vary the number of features on the classifier, in hopes a simpler model would be better. Unfortunately, after trying to get rid of many different features, we either were left with the same accuracy or worse. This is logical though, because all of the statistics we recorded are major statistics. Even getting rid of number of Disqualifications a team gets, what we thought was a rare and minor feature in our data, dropped our success rate to 70.15%.


*Figure 2.* How our predicted outcomes performed against the actual 2013 tournament.

## 6. Conclusion

Although 71.6% isn't a bad percentage for classifiers, it probably won't be good enough to predict 100% accuracy for a March Madness bracket. After filling out the 2013 bracket, we can see some flaws in our predictor. For instance, we predicted Creighton to beat Duke. This is due to the fact that Creighton had very high statistics because Creighton's conference is easier than Duke's. However, in the same bracket we see that we correctly picked Wichita (#9) to upset Pittsburgh (#8) and Iowa (#10) to upset Notre Dame (#7). Most dramatically, we correctly picked FGCU (#15) to upset Georgetown (#2). Therefore, we can see some fruit in our classifier.

## 7. Ideas for improvement

For future work, we should include some type of measure for how difficult a conference is to weight statistics accordingly. Additionally, if we could include player statistics also, this would probably create better accuracy, as player matchups in basketball are very important.

# Appendix A

Full list of features

1. Total Games
2. Winning Pct
3. Possessions
4. Possessions Per 40 minutes
5. Floor Pct
6. Efficiency
7. Field Goal Attempts
8. Field Goal Pct
9. Free Throw Attempts
10. Free Throw Pct
11. 3-pt Field Goal Attempts
12. 3-pt Field Goal Pct
13. Effective Field Goal Pct
14. True Shooting Pct
15. Free Throw Rate
16. Field Goal Point Pct
17. Free Throw Point Pct
18. 3-pt Field Goal Point Pct
19. Points Per Possessions
20. Points
21. Points Per Game
22. Rebound Pct
23. Total Rebounds
24. Total Rebounds Per Game
25. Offensive Reb Pct
26. Offensive Rebounds
27. Offensive Rebounds Per Game
28. Defensive Reb Pct
29. Defensive Rebounds
30. Defensive Rebounds Per Game
31. Team Rebounds
32. Team Rebounds Per Game
33. Assist Pct
34. Assists
35. Assists Per Game
36. Assist to Turnover
37. Steal Pct
38. Steals
39. Steals Per Game
40. Turnover Pct
41. Turnovers
42. Turnovers Per Game
43. Block Pct
44. Blocks
45. Blocks Per Game
46. Fouls
47. Fouls Per Game
48. Disqualifications
49. Outcome