March Madness Full Report

By: Juan-David Dominguez & Michael Gofron
juandominguez2017@u.northwestern.edu &
michaelgofron2017@u.northwestern.edu

EECS 349 Machine Learning, Northwestern University

Motivation

We were motivated to find a way to generate a perfect bracket or a better bracket than average based on the idea that it is a mathematically interesting task and economically advantageous. It is mathematically interesting because the predicted odds of getting a perfect bracket are 1 in 1,610,543,269. These odds are so slim because one would have to guess the correct outcome of 63 consecutive games, where an early mistake guarantees the rest of the bracket is incorrect. Additionally, it is economically advantageous because we can make money by betting with our friends and others on who has a better bracket, and there is usually money offered for a perfect bracket by multiple companies.

Tools

We are using the classifiers from the python Scikit library. This library includes our current main classifier Random Forest. We took in many features of a team that was compiled throughout the season games*.

Attributes & Data Manipulation

These team statistics include a team's performance and their opponents' collective efforts. For example, team statistics would include field goal points made, field goal points attempted and field goal percentage, as well as opponent field goals made, opponent field goal points and opponent field goal percentage, among many others. We also manipulated each feature (except for percentages) to be per game to prevent skewing from the number of games a team could play in the whole season, since some teams played more games than other teams. The manipulation of features to per-game was critical since it ensured that the teams that played more games (aka teams that made it further in the tournament) were not biased as significantly in the whole tournament. We then took the differential between a team's proper statistic and its opponent statistic. In all, we took in 120 features for a team into our decision tree as can be seen in *Appendix A*. For training, a game would be the instance where we would pit team A's statistics vs. team B's statistics and then whether team A won was the outcome.

Previous Classifier

We originally experimented with using a simple decision tree as a classifier before using a Random Forest. We had chosen to use decision trees for multiple reasons. From a human-perspective, the way in decision trees print in "if-then" routes where the early splits show which attributes give more information gain allows for us to gain a better intuitive understanding for the learning problem in general and helps us prepare for using other classifiers in the future. Furthermore, the problem tends towards decision tree. The discrete, binary output space works well with decision trees. The instances are clearly labeled making it supervised learning and the instances are represented by attribute-value pairs, which makes this learning problem well fitted for a decision tree. Lastly, decision trees are very robust to noise and this is very helpful since we are using a large data set. Unfortunately, a decision tree is much more prone to not capture the highest probability of which team is likely to win because it is a single instance. As a result we ultimately chose to use the Random Forest classifier.

Chosen Classifier

We chose Random Forest to use as our classifier for several reasons. Random Forest is an ensemble learning method for classification which creates a multitude of decision tress at training time and outputs the class that is the mode of the classes or mean of the individual tree. The selection of random subsets that occurs in Random Forest means that it trains on the data many times producing many decision trees and ultimately creating the most “likely” decision tree. As a result, random forests are much more likely to capture the best function that maps the data and avoid over fitting the data. Additionally, in *Appendix B*. we see that Random Forest generally performs much better than decision trees.

Results

The results are very interesting and further work needs to be done. The more exceptional result is that differential statistics are the most important features and, in particular Point differential is the most important feature. This seems to be in line with professional basketball analysts’ opinions. In addition in *Appendix B*, we see that our random forest generally outperforms decision trees for training on the playoff data testing on the playoff data, and training on the regular season data and testing on the playoff data. Our current accuracy is on ranges from 63% to 73% and is on average over all season and playoffs 67%. These results confirm that in general our random forests have much less variation than decision trees meaning that they are more likely capturing the true function rather than over fitting. The accuracy is better for tournaments that have been deemed to have fewer “upsets” such as can be seen in the 2015 tournament. In *Appendix D*, the bracket is one that our algorithm ultimately produced based on the teams that it predicted would win.

Suggestions for Future Work

We would like to try using more algorithms to see the validity of other algorithms and their impact. We also want to stack algorithms on top of one other in order to produce a more effective general algorithm for the future. Additionally, there should be more work done in separating the dataset between the playoffs and the regular season for the team statistics to ensure statistics are not skewed based on the number of games a team has played in a season. We would like to make the bracket making process more automatic.

Appendix A

ATeam = Away Team
HTeam = Home Team
DTeam = Differential statistics between home and away teams

['ATeam [0] team_fgm, 'ATeam [1] team_fga, 'ATeam [2] team_fgpct',
'ATeam [3] team_three_fgm, 'ATeam [4] team_three_fga,
'ATeam [5] team_three_fgpct', 'ATeam [6] team_ft',
'ATeam [7] team_fta', 'ATeam [8] team_ftpct', 'ATeam [9] team_pts',
'ATeam [10] team_ptsavg', 'ATeam [11] team_offreb',
'ATeam [12] team_defreb', 'ATeam [13] team_totreb',
'ATeam [14] team_rebavg', 'ATeam [15] team_ast', 'ATeam [16] team_to',
'ATeam [17] team_stl', 'ATeam [18] team_blk', 'ATeam [19] team_fouls',
'ATeam [20] opp_team_fgm', 'ATeam [21] opp_team_fga',
'ATeam [22] opp_team_fgpct', 'ATeam [23] opp_team_three_fgm',
'ATeam [24] opp_team_three_fga', 'ATeam [25] opp_team_three_fgpct',
'ATeam [26] opp_team_ft', 'ATeam [27] opp_team_fta',
'ATeam [28] opp_team_ftpct', 'ATeam [29] opp_team_pts',
'ATeam [30] opp_team_ptsavg', 'ATeam [31] opp_team_offreb',
'ATeam [32] opp_team_defreb', 'ATeam [33] opp_team_totreb',
'ATeam [34] opp_team_rebavg', 'ATeam [35] opp_team_ast',
'ATeam [36] opp_team_to', 'ATeam [37] opp_team_stl',
'ATeam [38] opp_team_blk', 'ATeam [39] opp_team_fouls',
'HTeam [40] team_fgm', 'HTeam [41] team_fga', 'HTeam [42] team_fgpct',
'HTeam [43] team_three_fgm', 'HTeam [44] team_three_fga',
'HTeam [45] team_three_fgpct', 'HTeam [46] team_ft',
'HTeam [47] team_fta', 'HTeam [48] team_ftpct', 'HTeam [49] team_pts',
'HTeam [50] team_ptsavg', 'HTeam [51] team_offreb',
'HTeam [52] team_defreb', 'HTeam [53] team_totreb',
'HTeam [54] team_rebavg', 'HTeam [55] team_ast', 'HTeam [56] team_to',
'HTeam [57] team_stl', 'HTeam [58] team_blk', 'HTeam [59] team_fouls',
'HTeam [60] opp_team_fgm', 'HTeam [61] opp_team_fga',
'HTeam [62] opp_team_fgpct', 'HTeam [63] opp_team_three_fgm',
'HTeam [64] opp_team_three_fga', 'HTeam [65] opp_team_three_fgpct',
'HTeam [66] opp_team_ft', 'HTeam [67] opp_team_fta',
'HTeam [68] opp_team_ftpct', 'HTeam [69] opp_team_pts',
'HTeam [70] opp_team_ptsavg', 'HTeam [71] opp_team_offreb',
'HTeam [72] opp_team_defreb', 'HTeam [73] opp_team_totreb',

Appendix A

'HTeam [74] opp_team_rebavg', 'HTeam [75] opp_team_ast',
'HTeam [76] opp_team_to', 'HTeam [77] opp_team_stl',
'HTeam [78] opp_team_blk', 'HTeam [79] opp_team_fouls']
'DTeam [80] team_fgm', 'DTeam [81] team_fga', 'DTeam [82] team_fgpct',
'DTeam [83] team_three_fgm', 'DTeam [84] team_three_fga',
'DTeam [85] team_three_fgpct', 'DTeam [86] team_ft',
'DTeam [87] team_fta', 'DTeam [88] team_ftpct', 'DTeam [89] team_pts',
'DTeam [90] team_ptsavg', 'DTeam [91] team_offreb',
'DTeam [92] team_defreb', 'DTeam [93] team_totreb',
'DTeam [94] team_rebavg', 'DTeam [95] team_ast', 'DTeam [96] team_to',
'DTeam [97] team_stl', 'DTeam [98] team_blk', 'DTeam [99] team_fouls',
'DTeam [100] opp_team_fgm', 'DTeam [101] opp_team_fga',
'DTeam [102] opp_team_fgpct', 'DTeam [103] opp_team_three_fgm',
'DTeam [104] opp_team_three_fga', 'DTeam [105] opp_team_three_fgpct',
'DTeam [106] opp_team_ft', 'DTeam [107] opp_team_fta',
'DTeam [108] opp_team_ftpct', 'DTeam [109] opp_team_pts',
'DTeam [110] opp_team_ptsavg', 'DTeam [111] opp_team_offreb',
'DTeam [112] opp_team_defreb', 'DTeam [113] opp_team_totreb',
'DTeam [114] opp_team_rebavg', 'DTeam [115] opp_team_ast',
'DTeam [116] opp_team_to', 'DTeam [117] opp_team_stl',
'DTeam [118] opp_team_blk', 'DTeam [119] opp_team_fouls']

Appendix B

Figure 1. Random Forest vs. Decision Tree training on previous playoffs and testing on a playoff.

```
Checking playoffs vs Playoff
Excluding year 2010
Performance of Decision Tree:  0.52380952381
Performance of Random Forest:  0.650793650794
Excluding year 2011
Performance of Decision Tree:  0.587301587302
Performance of Random Forest:  0.634920634921
Excluding year 2012
Performance of Decision Tree:  0.714285714286
Performance of Random Forest:  0.650793650794
Excluding year 2013
Performance of Decision Tree:  0.666666666667
Performance of Random Forest:  0.634920634921
Excluding year 2014
Performance of Decision Tree:  0.571428571429
Performance of Random Forest:  0.650793650794
Excluding year 2015
Performance of Decision Tree:  0.555555555556
Performance of Random Forest:  0.746031746032
>>>
```

Figure 2. Random Forest vs. Decision Tree training on previous regular season and testing on each seasons playoffs.

```
Doing Reg Season 2010 vs Playoff2010
Performance of Decision Tree:  0.619047619048
Performance of Random Forest:  0.650793650794
Doing Reg Season 2011 vs Playoff2011
Performance of Decision Tree:  0.571428571429
Performance of Random Forest:  0.634920634921
Doing Reg Season 2012 vs Playoff2012
Performance of Decision Tree:  0.650793650794
Performance of Random Forest:  0.68253968254
Doing Reg Season 2013 vs Playoff2013
Performance of Decision Tree:  0.634920634921
Performance of Random Forest:  0.650793650794
Doing Reg Season 2014 vs Playoff2014
Performance of Decision Tree:  0.587301587302
Performance of Random Forest:  0.730158730159
Doing Reg Season 2015 vs Playoff2015
Performance of Decision Tree:  0.650793650794
Performance of Random Forest:  0.730158730159
Doing all Reg Seanson vs All Playsoffs
Performance of Decision Tree:  0.611111111111
Performance of Random Forest:  0.669312169312
>>>
```

Appendix C

Table 1. The table below shows the percentage of games that we correctly guessed at the Midterm Report using a decision tree vs. using the random forest

Year (s)	Testing Data	Training Data	Percent Correct at Midterm Report	Percent Correct at Final Report
2010	Playoff Data	Regular Season	0.603174603175	0.650793650794
2011	Playoff Data	Regular Season	0.68253968254	0.634920634921
2012	Playoff Data	Regular Season	0.619047619048	0.682539682540
2013	Playoff Data	Regular Season	0.634920634921	0.650793650794
2014	Playoff Data	Regular Season	0.619047619048	0.730158730159
2015	Playoff Data	Regular Season	0.777777777778	0.730158730159
2010 - 2015	Playoff Data	Regular Season	0.571428571429	0.669312169312

Appendix D

Figure 3. Example of outputted Bracket