Predicting the Outcome of the NBA Championship Finals Using MapReduce and Machine Learning

Henry Bogardus

Turner Strayhorn

Big Data 2016

Vanderbilt University

Dr. Fabbri

Abstract

The overall goal of this project was to attempt to predict the outcome of the 2015-2016 season NBA championship based on a variety of team statistics, game logs, and player data that we scraped from the websites Basketball-Reference.com, stats.nba.com, and ESPN.com. We used Yelp's MRJob to transform our data as described later in the paper and the python scikit-learn library to make our predictions based on the transformed data.

Introduction

Each year, the NBA Finals produces one winner and, each year, there is an incredible amount of data collected by the NBA and other private parties that is put online, available for our viewing, collecting, and processing. Throughout the class, we have learned and made use of data transforming assets like map-reduce and machine learning libraries like sci-kit learn. There are, obviously, unpredictable reasons that a team may not win or be predicted to win, then lose such as random player injury (i.e. Derrick Rose in 2011). However, we want to know if there is 1) an obvious way, and 2) some less obvious ways to predict the outcome of the NBA championships using transformations on this data we have collected and machine learning with sci-kit learn.

Data Collection

1. NBA Stats

To collect our team season statistics and game logs, we scraped from the NBA stats website. By using Chrome's developer tools and going to the Network/XHR tab, the NBA site displayed a couple different linked JSON files that made up the statistics displayed on the web page. By opening those JSON files in the browser and copying the URL, we were able to scrape the data by using the Python Requests module. Each JSON for each season was read into its own Pandas DataFrame, which was then stored in a list. When all data was collected the DataFrames were all concatenated and stored in the CSV files. Depending on the amount of the information present in each JSON, some additions needed to be made in terms of adding a year or team ID to each table as it was pulled into the CSV. To create the champs.csv file we created a dictionary of which team ID won a championship in each year and then ran the rows of all teams' season statistics through it and wrote to the file a 1 next to a team if they were the champions during a specific year or 0 if they weren't.

2. Basketball-Reference

We collected team rosters and player data from Basketball-Reference. This website was much trickier to scrape than NBA Stats and has stated that they prefer not to be scraped and they have been known to ban IP addresses if they think scraping is occurring from the address.

For this site, we chose to use selenium browser automation instead of Python Requests module that we used with NBA Stats for this reason exactly. The website did, luckily, provide an option for exporting player data as a csv which allowed us to automate all of the downloads. The URLs were all mostly uniform and described as the following:

```
http://www.basketball-reference.com/teams/<team-abbreviation>/<year>.html
```

So we were able to make a python list of each team abbreviation and open the url in Google Chrome with selenium web browser. To export the player information, we used a JavaScript call that is made when the export button was pressed. Initially, we put a timer on the script to make sure it didn't seem to suspicious. However, we quickly realized how slow it was to collect the data. There are 30 NBA teams and 2 CSVs downloaded per team, the roster and the player totals for the season. The site is slower and would take about 20-30 seconds to load each csv leading to about 30 minutes of data collection per season with the timers. Even after we removed the timer, the collection was still quite slow and would take about 20 minutes per season. We decided the past 10 years of data was more than plenty for our collection. The rosters were all compiled into one big csv and the season and team abbreviation was appended.

Data Transformations

1. Game Logs

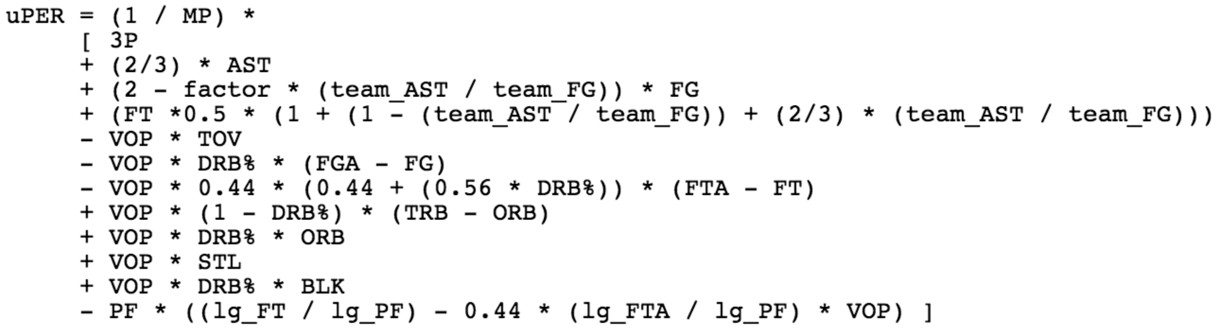
Most data transformations were done using yelp's MRJob module for python. We first did transformations on game-logs to extract, per season, for how many games did the winning team rely on free throws to win. The mapper step mapped the log data by the GAME_ID. The combiner step then read in that data, determined the winning team, and calculated the number of points the team got from free throws. The combiner step also calculated the difference in the number of points the winning team won by and sees if the number of free throws is greater than that value. Finally, the reduce step sums the number of games won because of free throws. We discussed and decided not to take into account the losing teams' free throws when coming up with this metric because we determined it was irrelevant for what we wanted to determine.

2. Player Stats

For player logs, we wanted to see if average player weight, height, and number of colleges had any correlation with NBA championships. We wrote a map-reduce job for each of these metrics that read in the player rosters for each team and each season and reduced them with the key as the season and the team name. Finally, we decided to use the player efficiency rating PER[11]. We totaled the PERs of each team's players and added that to our list. The equation is

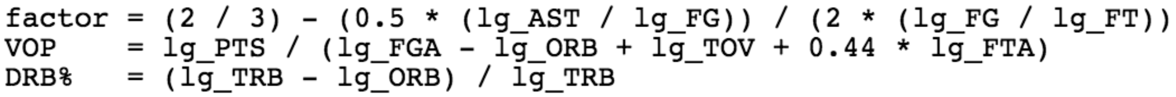
[1] http://www.basketball-reference.com/about/per.html

described as the following.

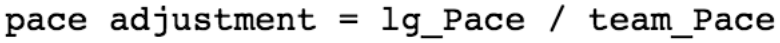


```
uPER = (1 / MP) *
     [ 3P
     + (2/3) * AST
     + (2 - factor * (team_AST / team_FG)) * FG
     + (FT *0.5 * (1 + (1 - (team_AST / team_FG)) + (2/3) * (team_AST / team_FG)))
     - VOP * TOV
     - VOP * DRB% * (FGA - FG)
     - VOP * 0.44 * (0.44 + (0.56 * DRB%)) * (FTA - FT)
     + VOP * (1 - DRB%) * (TRB - ORB)
     + VOP * DRB% * ORB
     + VOP * STL
     + VOP * DRB% * BLK
     - PF * ((lg_FT / lg_PF) - 0.44 * (lg_FTA / lg_PF) * VOP) ]
```

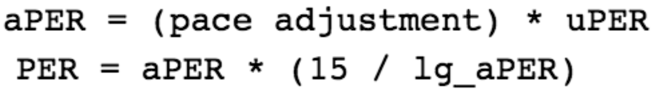
where



```
factor = (2 / 3) - (0.5 * (lg_AST / lg_FG)) / (2 * (lg_FG / lg_FT))
VOP    = lg_PTS / (lg_FGA - lg_ORB + lg_TOV + 0.44 * lg_FTA)
DRB%   = (lg_TRB - lg_ORB) / lg_TRB
```



```
pace adjustment = lg_Pace / team_Pace
```



```
aPER = (pace adjustment) * uPER
 PER = aPER * (15 / lg_aPER)
```

For each team, every player’s PER is weighted by average minutes per game they play, and then is averaged over he total minutes per game the team plays. Finally, the result is subtracted by 15 to give an average team a PER of zero. This metric, as ridiculous as it looks, does a good job "summing up the positive accomplishments of a player, subtracts the negative accomplishments of a player, and returns a per-minute rating of a player's performance.” This metric is aimed at doing a better job at factoring in a team’s ‘Star Power,’ which is widely known to become more important as the playoffs progress and it becomes tougher to score on stingier defenses. Using the code found in the PERcalculator.py file, we were able to calculate every player’s PER for each year. One caveat is that the script tends to very slightly inflate the metric if it is above ~10 and slightly deflate it if it is below 10 (for example the formula produces a PER of 35 for Stephen Curry while his actual PER for the 2015-16 season was only 31.5).

Machine Learning

To solve our high level problem of which team exactly is the most likely to win the 2016 NBA championship, we ran 1000 simulations of our machine learning algorithm to determine the percent chance of every team to win for the 2016 season. Starting off, we based a lot of our functionality off of Dr. Fabbri's machine learning example file on GitHub and then tailored it to our needs and the differences in how the data was stored. The classifier that produced the set of results that was the most easily understood was the SGD linear model. We achieved giving each team a percent chance of winning by running a thousand simulations, keeping a running counter of how many times each team got a simulated win, then dropping teams not in the playoffs that year and normalizing to 100% shared between the sixteen playoff teams. We then can see which teams have the highest percentage chances of winning. To train the classifier we had a number of features (over 30) to choose from to find the right combination to most accurately pick a champion. There were a number of complications involved with keeping the NBA data consistent over the past 20 years, as there have been a number of changes to team names, team cities, and the number of teams in the league. These problems showed up when using Sci-Kit Learn's Predefined Split and defining exactly which rows to train on and which to test on. To fix this, I added a quick couple lines before the main method that standardized the process of which lines were for testing versus which lines were for testing based on the year for which we were looking for results. In addition to using the SGD classifier, we also use a Gaussian Naïve-Bayes classifier and a Random Forest Calculator to test the Area Under the Curve score.

Outcomes

To predict the outcome of the 2015-16 NBA Finals, we used the model described above with a few different sets of features. The base set of features includes win percentage, field goal, three-point, and free throw percentages, team plus-minus, rebound percentage, assist-to-turnover percentage, assists-per-field goal percentage, steals, blocks, net rating, and pace. When the model is run using these features, the conference finalists are correctly picked 75% of the time, the conference champions are picked 63% of the time, and the champion is correctly predicted 57% of the time. A better statistic is that the actual champion originates from the predicted top four team 84% of the time. When our team PER statistic was factored into the algorithm, the conference finalists were correct 71%, the conference champions were correct 50%, and the champion was correct 52% of the time. However, champion originated from the predicted top four 89% of the time (exceptions were early 2000s Lakers and the 2011 Mavericks the year they beat the Heat). Finally, when factoring in the free throw metric, conference finalists were correct 71%, finalists were correct 45%, and champions were correct 37% of the time, demonstrating that this metric may not be the best when trying to determine the NBA champion for a given season. When running the model on the current season using our baseline statistic, the results are given in the table below:

```
Atlanta Hawks: 0.7                        Miami Heat: 0.1
Boston Celtics: 0.5                       Milwaukee Bucks: 0.0
Brooklyn Nets: 0.0                        Minnesota Timberwolves: 0.0
Charlotte Hornets: 0.6                    New Orleans Pelicans: 0.0
Chicago Bulls: 0.0                        New York Knicks: 0.0
Cleveland Cavaliers: 10.6                 Oklahoma City Thunder: 12.9
Dallas Mavericks: 0.0                     Orlando Magic: 0.0
Denver Nuggets: 0.0                       Philadelphia 76ers: 0.0
Detroit Pistons: 0.0                      Phoenix Suns: 0.0
Golden State Warriors: 34.1               Portland Trail Blazers: 0.0
Houston Rockets: 0.0                      Sacramento Kings: 0.0
Indiana Pacers: 0.0                       San Antonio Spurs: 35.5
Los Angeles Clippers: 2.6                 Toronto Raptors: 2.5
Los Angeles Lakers: 0.0                   Utah Jazz: 0.0
Memphis Grizzlies: 0.0                    Washington Wizards: 0.0
```

Here we can see that the Warriors and Spurs, unquestionably the best two in the NBA right now each with about a 35% chance each of winning. The next most likely team is the Oklahoma City Thunder with 12.9%. The fourth most likely team and the only team from the Eastern Conference, the Cleveland Cavaliers, have a 10.6% chance of winning. When factoring in each team's average player efficiency, we result in the table below:

```
Atlanta Hawks: 0.3                        Miami Heat: 0.1
Boston Celtics: 0.1                       Milwaukee Bucks: 0.0
Brooklyn Nets: 0.0                        Minnesota Timberwolves: 0.0
Charlotte Hornets: 0.3                    New Orleans Pelicans: 0.0
Chicago Bulls: 0.0                        New York Knicks: 0.0
Cleveland Cavaliers: 8.1                  Oklahoma City Thunder: 12.6
Dallas Mavericks: 0.0                     Orlando Magic: 0.0
Denver Nuggets: 0.0                       Philadelphia 76ers: 0.0
Detroit Pistons: 0.0                      Phoenix Suns: 0.0
Golden State Warriors: 36.7               Portland Trail Blazers: 0.0
Houston Rockets: 0.0                      Sacramento Kings: 0.0
Indiana Pacers: 0.0                       San Antonio Spurs: 35.5
Los Angeles Clippers: 2.7                 Toronto Raptors: 3.6
Los Angeles Lakers: 0.0                   Utah Jazz: 0.0
Memphis Grizzlies: 0.0                    Washington Wizards: 0.0
```

Here we have very similar results, with the exception being that the Warriors now have a slightly higher chance of winning it all. However, the top four teams' percentages remain the same at relatively static percentages (on an interesting note, when the team PER is the only metric evaluated, the Raptors become the most likely team to win, followed closely by the Warriors). Finally, inclusion of the free throw metric results in the following:

```
Atlanta Hawks: 2.0                        Miami Heat: 0.4
Boston Celtics: 1.6                       Milwaukee Bucks: 0.0
Brooklyn Nets: 0.0                        Minnesota Timberwolves: 0.0
Charlotte Hornets: 1.5                    New Orleans Pelicans: 0.0
Chicago Bulls: 0.0                        New York Knicks: 0.0
Cleveland Cavaliers: 12.3                 Oklahoma City Thunder: 10.3
Dallas Mavericks: 0.7                     Orlando Magic: 0.0
Denver Nuggets: 0.0                       Philadelphia 76ers: 0.0
Detroit Pistons: 0.6                      Phoenix Suns: 0.0
Golden State Warriors: 28.5               Portland Trail Blazers: 0.1
Houston Rockets: 0.1                      Sacramento Kings: 0.0
Indiana Pacers: 0.9                       San Antonio Spurs: 35.2
Los Angeles Clippers: 3.3                 Toronto Raptors: 2.7
Los Angeles Lakers: 0.0                   Utah Jazz: 0.0
Memphis Grizzlies: 0.0                    Washington Wizards: 0.0
```

Conclusion

Using the three outcomes from the previous sets of metrics, our official prediction (as unbiased onlookers) for the champions of the 2015-16 NBA Finals is the San Antonio Spurs (this pick becomes self-prophesizing with the recent injury to Steph Curry). This is with the team statistic, taken from the past 15 years, the sums of player efficiency ratings for each team and, even the admittedly less accurate free throw metric calculated from the game logs. The Spurs chance of winning outweighs most other teams, save the Warriors, who trail fairly closely behind in some metrics, and farther behind in others.