NBA Matchup Prediction Application

Alex Brockman, Andrew Haisfield, Anish Prasanna, Carlos Samaniego

November 8, 2019

1 Introduction

The National Basketball Association (NBA) is amongst the most difficult to predict sports in the world. For this reason, it continually generates massive revenue from a dedicated fan base. The aim of our final project is to predict the probability of a team winning its game at crucial junctions of the match up.

For example, let's say that on Tuesday November 19th, the Heat and the Bulls were to play each other. Through our model, we would have an initial probability of either team winning the game. For the sake of example, let's say the Heat were projected to win at 60 percent. After every successive quarter that probability would change to indicate the eventual outcome of the match.

After the first quarter finishes, depending one how the game is going so far, the probability changes to 55/45 in favor of the Heat, 50/50 after the second quarter finished, the 55/45 once the 3rd quarter finishes. Once the 3rd quarter finishes, the team who has the probability of winning will be the team projected to win that game.

This information could be practically used for numerous applications. Firstly, one use could be for TV broadcasters to use this information to make an educated decision as to which game to cover given several games are being televised simultaneously. This is of tremendous significance for the NBA, since optimizing the viewership of games is crucial in generating interest. If boring games are televised, at the expense of more interesting games, viewership may be impacted significantly. Therefore, it is of utmost importance for the NBA, and its broadcasters to understand this crucial factor, in order to maintain its business.

An additional application could be for gambling purposes. As games progress, often times Sportsbooks and Casinos will modify their betting lines to reflect the change in betting demands on the two teams. Therefore, this movement can be exploited by interested gamblers to gain value on the lines. To illustrate this idea, suppose at the inception of a match-up, the underdog has heavy betting interest, and so the Sportsbook designates them at a near even line, suggesting

that the match-up is a close one. If a gambler wanted to bet on the underdog, the start of the match-up would not be a time that would present any inherent value. However, if the gambler were to use this application, he would be able to identify moments in the game where significant value could stand to be gained. Optimizing value effectively increases the expected profit long-term, which could be a more interesting monetarily related use of our application.

2 Data Description, Proposed Methods/Approaches

Data Description: The data that has been fueling our results are all from scraping websites (No API's)

The links we scraped:

- `https://www.basketball-reference.com/leagues/NBA_2020_games-november.html`
- `https://www.basketball-reference.com/leagues/NBA_2020.html#all_team-stats-per_poss`
- `https://www.basketball-reference.com/leagues/NBA_2020.html#all_opponent-stats-per_poss`

In order to scrape our data we used BeautifulSoup to web scrape the page that contained the "Team Stats per 100 Possessions" table. The web page contained comments that needed to be removed in order to accurately parse the table data. We used the "find_all" method in the Beautiful Soup library to convert the comments into html. The last task was to find the necessary table in the html that we needed to use for our project. That table was then entered into a pandas DataFrame that we were able to feed into our model.

Once we have all of the data, it is now time to use it in order to put together a model that predicts the outcome of each NBA game for a given day. Initially we tried predicting the total points scored for each team, but what we quickly realized was that this model over-estimated performance for teams with a good offense but a bad defense, while under-estimating teams with bad offenses but good defenses. Therefore, we decided to update our model to predict margin of victory (Points Scored - Points Allowed), as this accounted for defensive-minded teams much better.

After settling in on what we want to predict, the next step was to determine which subset of statistics would most-accurately project the margin of victory for a given team. With that being the case, we designed our model to evaluate the R-Squared value for each possible combination of predictor-variables (starting from length=1 up through length=total number of variables. From there, we found the subset of variables for each evaluated-length that had the

highest R-Squared value and grouped them together. By looking at a plot of the R-Squared values for these selected subsets, we can look at the point of highest-curvature on the graph to determine the model we want to use. For example, in the graph at the end of the document, we would choose the subset with 5 features as our model.

Now that we have our optimal model, we can then predict the margin of victory for each of the 30 teams in the NBA. Then, the next step would be to pull down the NBA schedule for a given day, and based on what the matchups are for that day, we would predict that the team with the higher-projected margin of victory would win the matchup.

3 Preliminary Results

What is Working:

- We can successfully scrape the NBA schedule and Team Stats tables from basketball-reference.com
- We can select the best subset of variables that formulates the optimal model for predicting margin of victory
- Based on margin of victory, we can predict the result of any given NBA matchup fairly accurately
- We created two links within our website, localhost:5000/predictions and localhost:5000/schedule
- The schedule link successfully outputs data frame tables onto the site
- The predictions link successfully outputs our predictions into a simple html page

What is not Working:

- We need to automate the process of selecting which date to analyze so that we can make predictions for today's date. As of right now, we have to manually input which slate of games to look at.
- The model currently displays the projected win margin, but we would like to convert that into a confidence metric (percentage chance that the team wins)
- There are ways that we can experiment making the model more accurate. For example, it currently does not take into account whether the team is playing at home or on the road.
- We need to automate which games are displayed based on what day it is currently as well as the initial predictions for that day.

- We also need to make it all look nicer. We can do this by adding logos of the teams playing, and having an overall nice style.
- We also should figure out how to display the data frame table in a pretty way within our html page, so far we're having trouble integrating the dataframe table within the html page. It is currently being outputted in a hard coded way.

4 Next Steps

- Fixing up the style and design of the overall site, and working to integrate our info within it
- Possibly adding a database to make our info persistent in such a way where we can look at post probabilities of teams winning in the past.
- Design the program to automatically display the games/predictions for the current date
- Convert the displayed projected win margin into a win percentage chance metric
- Adjust model to give the home team a slight advantage