Who will be the 2019 NBA Season MVP?

Can we predict who will win?

Michael Oppong
Computer Science
York University
Toronto ON, Canada
mikestax@my.yorku.ca

ABSTRACT

The National Basketball Association (NBA) is a professional basketball league that consists of many teams across North America [1].

During the regular NBA season, each team plays eighty-two games in total, after that the top 16 teams face off to win the NBA championship in the playoffs. After all this, there will be a variety of awards given out to players for their work in the regular season. The most valuable player (MVP) award is one which is given out to the player for their hard work which is shown with their stats, and number of the games they help their team win.

The MVP award is one of the most craved awards in the NBA, next to a championship ring. Since the broadcasters and sports writers who choose the MVP, do not disclose the information about how they pick this person, we will come up with our own model to generate a player, or list of players who have potential to become MVP.

To predict the player who will win this award, or players who have potential to win, would take a great deal of skill and effort. This is what this paper will discuss; how can we predict, with a big data solution the MVP of the 2019-2020 NBA season.

1 Introduction

This project is about finding a way to predict who the MVP of the 2019-20 NBA season will be, and for future seasons. This project is meant to be self-sustainable meaning, that at any point during any NBA season to come in the future, as long as the application is run during that season, we will be able to predict a certain player or players who have potential to win the regular season MVP.

In this report, we will be looking at NBA players of the from 2009-10 season all the way to 2019-2020 season. We will be comparing the stats the MVP winners from the 2009-2010 season to the 2018-19, to get some form of consistency. We will then apply that same theory and analytics to players in the current 2019-20 NBA season to predict who will win the MVP award.

This model has great potential to be applied within the same field of basketball. We could predict the rookie of the year (ROY), and potentially, the championship team.

2 Data type and Analysis

The data which we used to predict the MVP for the current 2019-20 season was based on past data collected from (Basketball-Reference.com, 2019). The prediction was constructed on MVP winners from the 2009-10 season to the 2018-19 season. From these past seasons, data which was needed on each player who won the MVP award was their win-shares, team they were on, and the win/loss record for that team. The term win-shares is the estimated number of wins attributed by a player to their respective team. This in simple terms means out of all the games won by a team, how many of those wins were attributed by a player.

To put this in perspective, if player A played is on team B, and team B has 10 wins and 0 losses, and player A defended every single player on the opposing team, and scored every point, this would then give player A 10 win-shares. To understand the data used in this analysis, an example would be the 2018-19 MVP Giannis Antetokounmpo. The win-shares for Giannis was 14.4, and he played on the Milwaukee Bucks which had a win/loss ratio of 60/22. This means that out of the 60 wins his team had, he helped with 14.4 of them.

The previous paragraph talks about how we slowly found relationships between the players who won the MVP award through their win-shares, and teams win/loss ratio. This paragraph talks about how we will use that for prediction. For prediction, we get the same data (win-shares, team win/loss ratio), we compute using these stats to achieve a certain value for each player, and for each player within a value within a certain range, we stream data from sports websites through rss feeds, and tweets, then run sentiment analysis on the content coming in about each player, and the player with the highest value, or players, would be subject to winning the MVP at that specific instance the application is run. This computation is not run on all the players in the current season, it is only computed for players in the past years all-stars list, combined with players with high-win shares in the current NBA season. We do this because we found through multiple tests, that the MVP for the NBA season x, had always been in the past years all-stars list, except for 2 occurrences; Wilt Chamberlain 1959-60 and Wes Unseld 1968-69. To account for these types of mishaps, we get the players with the top 10 win- shares of the current season, just in case they were not all-stars the prior season, or they just happened to be superb during the current season.

3 Architecture

The current architecture of this system can be explained with four steps:

1. Gather the datasets of the prior NBA seasons; the list of all-stars from the prior season you want to predict for, league summary (showing the wins and losses of each team) for the current season, and the advanced player stats which has a column labeled win-shares. (These datasets can be retrieved from [2])
2. Sort the NBA player stats by win-shares and get the top 10 players and add them to the all-stars list. Compute the win-shares/team-wins, for each player to achieve a value. ***Equation 1***
3. Retrieve only those players with a value ranging from 0.2-0.35, and stream tweets from sports websites and tweets, while also running sentiment analysis on what is said about each player.
4. Plot the results from the sentiment values received for each player. Players with the highest sentiment value are assumed to have a higher percentage of winning MVP.

$$Pi_{WinShares}/Pi_{TeamWins}$$

Equation 1: How to calculate player value per player

We repeat steps 3 and 4 because are streaming tweets and headline articles, then running a sentiment analysis to determine whether the content the player is being mention in is either positive negative or neutral. Figure 2 is meant to be a resemblance of the lambda architecture. In Figure 2, we provide both the streaming layer, and the batch layer. The batch layer consists of updating the player stats and team standings, while the streaming layer is the content being streamed about players. The streaming layer is dependent on the batch layer, as it used information on which players were within the 0.2 to 0.35 range to stream.

The Batch layer only needs to be updated once a day, while the streaming layer is continuously (seconds). The batch layer needs to be updated a day, because we want to stream content on strictly those players within the range on 0.2 and 0.35, so though a player might be within that range today, tomorrow, they might not. This is crucial to the system as you would be predicting based on data from a specific date. Since the NBA is due to major changes, such as trades, injuries, and many more, we need to keep the bath layer updated once a day at least.

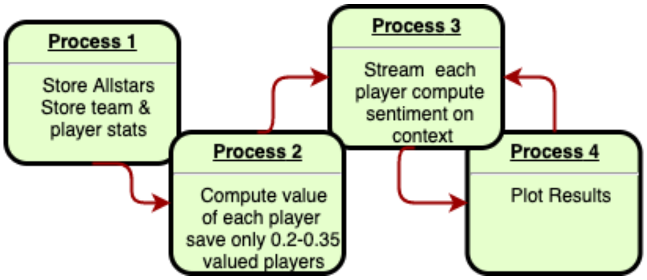


Figure 1: Describes the architecture of the system

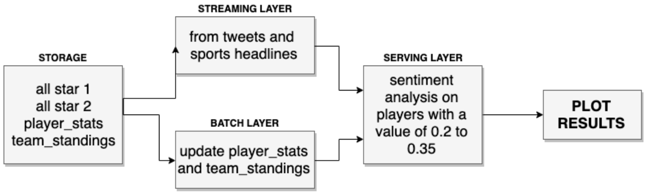


Figure 2: Describes how the architecture is similar to the lambda architecture with the streaming layer

4 Evaluation results

During the evaluation of the past MVP's, we found the win-shares divided by the total team wins of the past 10 NBA MVP's were all within the range of 0.2 and 0.35. As show in the *figure 2*. This graph shows the difference in the win-shares divided by total team wins for each of the past 10 MVP's compared to the players of the 2018-19 season. We can clearly see that this relationship does hold, and if we were to plot the same results with players from any particular

NBA season, we would get the same results, with the MVP's being at the top left, and the rest at the bottom.

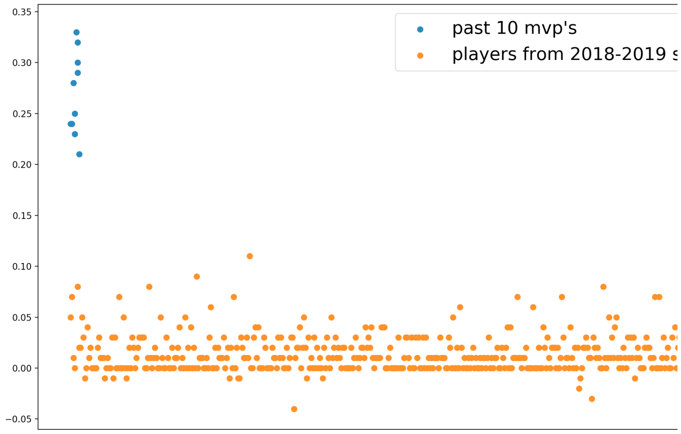
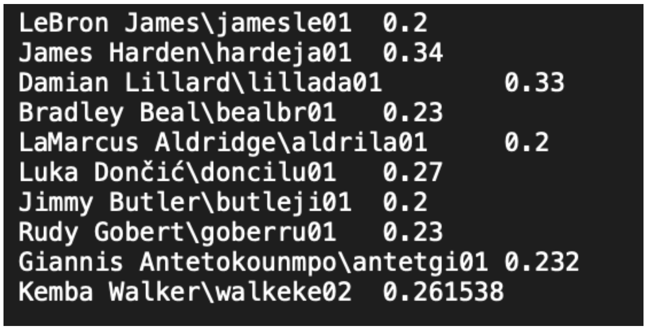


Figure 3: Shows difference in win-shares/total team wins for MVP's compared to regular players.

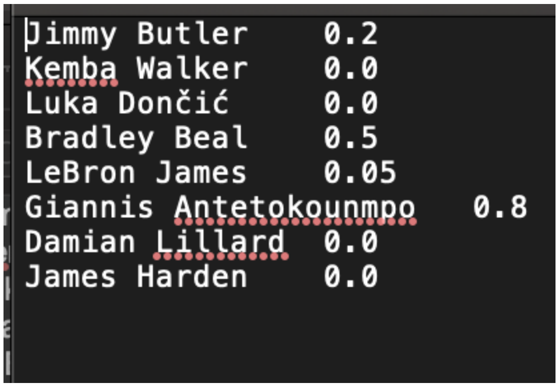
Noticing this trend, we run analysis on the all-stars, including the top 10 players with a high win-share, and arrived at a list of players within the 0.2 to 0.35 range. Figure 4 tells us who the players are



```
LeBron James\jamesle01  0.2
James Harden\hardeja01  0.34
Damian Lillard\lillada01        0.33
Bradley Beal\bealbr01   0.23
LaMarcus Aldridge\aldrila01     0.2
Luka Dončić\doncilu01   0.27
Jimmy Butler\butleji01  0.2
Rudy Gobert\goberru01   0.23
Giannis Antetokounmpo\antetgi01 0.232
Kemba Walker\walkeke02  0.261538
```

Figure 4: Shows the players who have the potential to win the MVP.

Figure 4 shows the players who are within the 0.2 to 0.35 range, separated by a '\' with their twitter handle names, along with the value received from the win shared divided by team wins. A stream of tweets was received on each player, to determine who was the highest valued through tweets. The conclusion can be seen in Figure 5. The player with the highest value, is Giannis Antetokounmpo. This means, based on my algorithm, Giannis, is the most favored to win the MVP award. Second place would be Bradley Beal, then Jimmy Butler, and the so-on as each player value decreases. Though no information was received for Rudy Gobert and LaMarcus Aldridge from this set of streams, it is perceived that they are unlikely to win due to the fact that they aren't being talked about within social media, though, they do have a higher change of wining compared to other players who are not within the 0.2 to 0.35 range,



```
Jimmy Butler     0.2
Kemba Walker     0.0
Luka Dončić      0.0
Bradley Beal     0.5
LeBron James     0.05
Giannis Antetokounmpo    0.8
Damian Lillard  0.0
James Harden     0.0
```

Figure 5: Shows difference in win-shares/total team wins for MVP's compared to regular players.

5 Conclusion

In conclusion, we were able to come to with a player or players who have potential to become MVP. Any average basketball enthusiast, a person who loves to watch the game, could have been able to come up with general names of players who are on the list we came up with. Though it doesn't seem as hard to come up with this list for the average person who watched basketball, for an individual who does not know anything about this would be very helpful. This would give them an overview of who the current top players are, and an idea of how well they help their team win.

For the future, the way the program is run, would be needed to be simplified, you run an initial program, which gives you a bar graph, of players contending for the MVP, and that is it. No requirement for the user to download files and then run the application with those files, just a simple run of the main py file, and the answer is given. In terms of the architecture, in the future, a more solidified version of the lambda architecture needs to be implemented. The batch jobs would update the player and team stats each day, while the streams of content for the players within the 0.2 to 0.35 range are being processed separately.

In brief this project taught me a lot about the lambda architecture, which is quite useful in various scenarios, where some data need to be updated more frequently than others.

REFERENCES

[1] NBA.com. “NBA Frequently Asked Questions.” *NBA.com*, NBA.com, 20 Mar. 2019, www.nba.com/news/faq.

[2] “Basketball Statistics and History.” *Basketball*, www.basketballreference.com/

Basketball-Reference.com. (2019). *Basketball Statistics and History | Basketball-Reference.com*. [online] Available at: https://www.basketball-reference.com/ [Accessed 2 Dec. 2019].

[3] McKinney, W., & others. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).

[4] John D. Hunter. **Matplotlib: A 2D Graphics Environment**, Computing in Science & Engineering, **9**, 90-95 (2007),

[5] Travis E. Oliphant. **A guide to NumPy**, USA: Trelgol Publishing, (2006).

[6] http://www.tweepy.org/,

[7] https://www.pythonforbeginners.com/