Trust the pRocess: NBA Award Predictions

STAT 385 SP2019 - Team Hurst

Ajay Dugar (dugar3)
Eric Qian (ewqian2)
Joshua Immanuel (joshuai2)
Bhanuchandra Kappala (kappala2)

May 9th, 2019

Abstract

This project will involve looking at 30 years of individual NBA player statistics. We will develop a number of new advanced analytics and statistics to determine player efficient and value. Using these numbers, we will rank players and determine who should win prominent NBA awards. We will visualize these results using bar graphs and sentiment analysis.

Contents

1 Introduction

1.1 Problem Statement

While fans and media pundits try to guess the winners of the major NBA awards (Most Valuable Player, Defensive Player of the Year, Rookie of the Year, and Sixth Man of the Year), the issue is that there's no real criteria or concrete way of predicting the winners of these awards. In this project, we seek to use historical data to predict these winners.

1.2 Relevance

Valuing player assets are important. Being able to accurately determine player value is one of the most fundamental skills in basketball front offices. Being able to do so accurately can be the difference between winning and losing. Many different subjects (statistitcs, differential equations, analytics) can be applied to these problems. With the success of Daryl Morey and Sam Hinkie, we see that these approaches have been successful in the NBA. Many interesting and unsolved issues in the NBA require statistical analysis, which is why we chose this project.

1.3 Description of the data

The data is a series of dataframes of regular and advanced statistics spanning from 1989 and 2019.

- Cite papers or reputable sources that back up this claim. (You may want to find - material using Google Scholar.) http://www.basketballanalyticsbook.com/ https://www.nbastuffer.com/analytics-101/ https://squared2020.com/
- Where did the problem or topic come from?

Our shared passion of basketball and the NBA led us to this topic. This is a topic that some of us have talked about and previous academic curiousity have led us to this topic. We wanted to know if we could use a variety of variables to accurately predict NBA Award winners.

- What is your idea for addressing the problem or topic?

First, find and clean NBA player data. Then create a series of advanced metrics to measure player performances. Train a model that utilizes these metrics to correlate them with team success. Finally, calculate total offensive value, total defensive value, and total overall value to determine player awards. Finally, test its predictive power on a series of previous award winners and visualize our results.

- How does your idea match with the course's focus on statistical programming?

We will be dealing with large datasets, validating data, and building statistical models. All of these skills form the foundations of statistical programming and we will apply what we have learned in our project.

-What is our data? Where did it come from? How will it be useful in answering your problem?

Our data will come from the official NBA website. It will be 30 years of individual player data. The long time period and amount of data for our data points will allow us for a greater degree of accuracy and precision in the evaluation of our model.

https://stats.nba.com/

2 Related Work

Address the following questions:

- What other ideas have been attempted?

While not specifically related to this project, the application of sabermetrics has become a point of interest for many baseball teams. Likewise, we have began to see this same transition in basketball analytics. The long term view of teams that have embraced analytics has been very good: the Philadelphia 76ers and Houston Rockets are both top seeded teams that have a significant chance at being the Championship team. While we haven't seen rigorous academic analysis of predicting NBA awards, there have been some implementations done by various individuals. There are also award trackers that just use raw data - specifically VORP (Value over Replacement Player) and PER (Player Efficiency Rating) and the player with the highest value is predicted to win. While these ways of predicting awards works somewhat well, they tend to have problems undervaluing certain players (the inventor of PER notes that it is not an end all metric - it rewards inefficient shooting and doesn't sufficiently reward effective defense) so our idea will be unique.

- Why is your team's idea original compared to prior work?

A lot of the statistics in use with modern publications are based on the research of one individual: Daryl Morey. We will derive our own variables and determine how impactful they are in determining wins using a regression model. By doing so, we will establish other variables and metrics that may be better suited for different situations and will allow for better comparison of players across time periods and eras.

3 Methods

The **methods** section should discuss how you plan to solve your problem. The overall details of the project including any preliminary work. In particular, the implementation details behind the approach should be explained at length here. The more details you can provide, the better feedback your group can receive. As a result, the section serves as a roadmap of what features are going to be developed and any external dependencies that are required. **The majority of your code should be *suppressed* from the displaying in this section**. Please refer to code and figures placed in the appendix. The latter can be referenced using:

`Figure \\ref{fig:code-chunk-name-here}.`

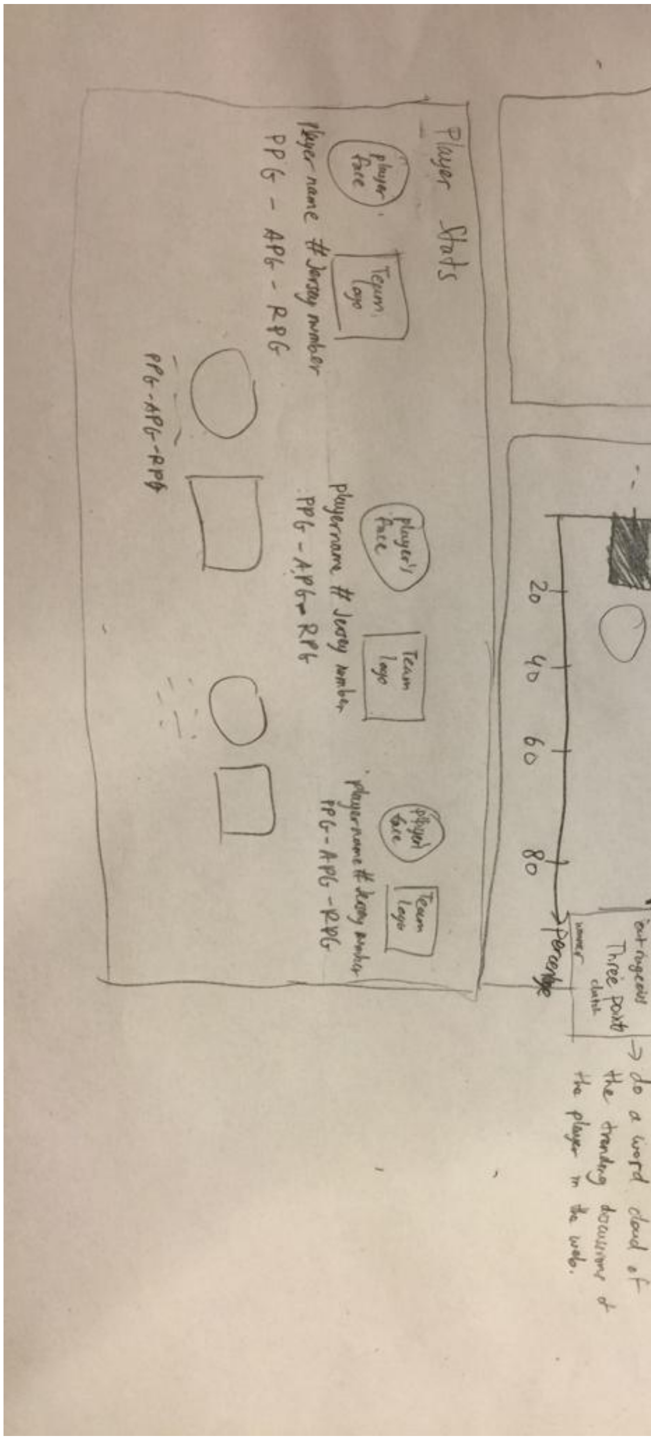
For example, the figure of the data science workflow is accessible via Figure **??**.

To satisfy this section, provide detailed responses for the following:

- What packages will you use in your implementation?
  - Data transformation: tidyr, dplyr, ballr
  - Data visualization: ggplot2, plotly, gganimate
  - Interactive Interface: shiny
  - Regression Analysis: leaps, glmnet, caret
- What code will the group need to write for the project?
  1. Code for importing the data
  2. Code for cleaning the pre-collected data and randomizing it
  3. Code to create the linear regression in obtaining the weights for each respective variables
  4. Code to create the toggle bar for the user interface input
  5. Code to create the interactive bar graph where it also displays the summary of the players
- Provide low-fidelity prototypes (e.g. *sketches* on paper) in the **Appendix** of:
  - Visualisations
    - What kinds of graphs will you use?

    We will use a horizontal bar graph where the y-axis contains the player's names and the x-axis contains the predicted percentage scores. We will put a picture of the athlete on the right side of their respective bars. The user will be able to interact with the graph by toggling on the face of the athlete and a word cloud will appear. The word cloud will consist of the google trending discussions on the athlete.
    - Label axes, provide a title, and mention any interactivity.
  - Interface

- * All projects need a Shiny Application.



- * Sketch how a user will work with the shiny application.

- What have you done or learned so far for the project?

We have learned that we need to gather the statistical player data and filter them so that the we have relevant data. We need to gather the variables and do regression to determine weights of each variables. We need to gather data on player discussion trends to project it in the graph. We also need to create a user interface where the we get the user input data and graph the predicted results as well as the player statistics.

We are primarily wanting to ensure that your project has met the criterion of the data science pipeline. In essence, we want to see evidence that your project has:

- Reading data into *R* or accessing data via an API.
- Data transformations (e.g. Tidying (`tidyr`), Summarizing (`dplyr`), et cetera.)
- Data visualization (e.g. `ggplot2`, `plotly`, `gganimate`)
- R functions either in external packages or included in a new *R* package
- Interactive Interface (e.g. `shiny`)
- Reproducibility

4 Feasibility

The **Feasibility** section is meant to act as a way to reflect upon the proposal. Generally speaking, there will be three weeks of heavy development time afforded to the group. Building a detailed ecosystem or heavily scripting in a different language will likely not lead your team to success. Hence, please provide a project management overview of *who* on your team will be doing *what* and *when* by answering:

- Is this project able to be completed before the end of the semester?

It should be - while the data set is large, outside of the resource requirement, it should not be overly difficult to implement this specific idea. We will need to familiarize ourself with the specific data used here - lots of unique compiled stats are used in basketball as opposed to the typical counting stats.

- What steps must occur to complete the project before the end of the semester?

Accumulate the data in a timely manner - we must find what data we are going to use and apply our model to it. We plan to have 30 years worth of data so data cleaning/valdiation is an important and time consuming portion. We must also create specific derived variables that will more effectively illustrate the value of each player to their team for all the specific criteria. We must then model each of the awards - they will all have unique weights for each of the statistics.

- What is the work plan to accomplish the necessary tasks before the end of the semester?
  - Specify who is doing what and when.
  - Consider making a Gantt chart to highlight each stage of the project.
    Ajay - obtain data (primarily from BBallRef) and create derived variables Bhanu - build model and generate output (train model for each individual award) Joshua and Eric - build visualization components of the Shiny app Ajay and Bhanu - add sentiment analysis portion to visualizations (make a word cloud)

5 Conclusion

If you can accurately predict NBA awards, there is significant money to be made betting on winners of these awards. Additionally, by looking at this topic, NBA teams can get a better idea of what a winning player looks like, and can construct teams accordingly.

This project will involve looking at 30 years of individual NBA player statistics. We will develop a number of new advanced analytics and statistics to determine player efficient and value. Using these numbers, we will rank players and determine who should win prominent NBA awards. We will visualize these results using bar graphs and sentiment analysis.

6 Appendix

The **Appendix** section contains figures, sample data, and other miscellaneous entries. Generally, this sketch seeks to contain all of your *planning* information.

- Provide the sketches of visualisations and the shiny application.
- Provide an overview on the desired functions.
  - What is a function's input? Output? How are functions related to each other.
  - For example, `read_data("hospital_data.csv")` must be called before `tidy_hospital()`, et cetera.
- Provide a sample of the data set you intend to use (~10 observations).

```
##               PLAYER TEAM GP  W  L    W.    WQ  MIN  PTS  FGM  FGA  FG.
## 1       James Harden  HOU 81 54 27 0.667 0.506 36.4 29.1  8.3 18.9 44.0
## 2  Russell Westbrook  OKC 81 46 35 0.568 0.409 34.6 31.6 10.2 24.0 42.5
## 3      DeMar DeRozan  TOR 74 47 27 0.635 0.468 35.4 27.3  9.7 20.9 46.7
## 4          John Wall  WAS 78 48 30 0.615 0.467 36.4 23.1  8.3 18.4 45.1
## 5      Isaiah Thomas  BOS 76 51 25 0.671 0.473 33.8 28.9  9.0 19.4 46.3
## 6       Jimmy Butler  CHI 76 40 36 0.526 0.406 37.0 23.9  7.5 16.5 45.5
## 7     Damian Lillard  POR 75 38 37 0.507 0.379 35.9 27.0  8.8 19.8 44.4
## 8       LeBron James  CLE 74 51 23 0.689 0.543 37.8 26.4  9.9 18.2 54.8
## 9         Kyle Lowry  TOR 60 36 24 0.600 0.468 37.4 22.4  7.1 15.3 46.4
## 10    Andrew Wiggins  MIN 82 31 51 0.378 0.293 37.2 23.6  8.6 19.1 45.2
## 11  DeMarcus Cousins  NOP 72 30 42 0.417 0.297 34.2 27.0  9.0 19.9 45.2
## 12      Devin Booker  PHX 78 24 54 0.308 0.224 35.0 22.1  7.8 18.3 42.3
## 13      Kyrie Irving  CLE 72 47 25 0.653 0.477 35.1 25.2  9.3 19.7 47.3
## 14     Kawhi Leonard  SAS 74 54 20 0.730 0.508 33.4 25.5  8.6 17.7 48.5
## 15     Stephen Curry  GSW 79 65 14 0.823 0.573 33.4 25.3  8.5 18.3 46.8
## 16     Blake Griffin  LAC 61 40 21 0.656 0.464 34.0 21.6  7.9 15.9 49.3
## 17    Gordon Hayward  UTA 73 46 27 0.630 0.453 34.5 21.9  7.5 15.8 47.1
## 18      Eric Bledsoe  PHX 66 22 44 0.333 0.229 33.0 21.1  6.8 15.7 43.4
## 19      Goran Dragic  MIA 73 40 33 0.548 0.385 33.7 20.3  7.3 15.4 47.5
##    X3PM X3PA X3P. FTM  FTA  FT. OREB DREB  REB  AST TOV STL BLK  PF
## 1   3.2  9.3 34.7 9.2 10.9 84.7  1.2  7.0  8.1 11.2 5.7 1.5 0.5 2.7
## 2   2.5  7.2 34.3 8.8 10.4 84.5  1.7  9.0 10.7 10.4 5.4 1.6 0.4 2.3
## 3   0.4  1.7 26.6 7.4  8.7 84.2  0.9  4.3  5.2  3.9 2.4 1.1 0.2 1.8
## 4   1.1  3.5 32.7 5.4  6.8 80.1  0.8  3.4  4.2 10.7 4.1 2.0 0.6 1.9
## 5   3.2  8.5 37.9 7.8  8.5 90.9  0.6  2.1  2.7  5.9 2.8 0.9 0.2 2.2
## 6   1.2  3.3 36.7 7.7  8.9 86.5  1.7  4.5  6.2  5.5 2.1 1.9 0.4 1.5
## 7   2.9  7.7 37.0 6.5  7.3 89.5  0.6  4.3  4.9  5.9 2.6 0.9 0.3 2.0
## 8   1.7  4.6 36.3 4.8  7.2 67.4  1.3  7.3  8.6  8.7 4.1 1.2 0.6 1.8
## 9   3.2  7.8 41.2 5.0  6.1 81.9  0.8  4.0  4.8  7.0 2.9 1.5 0.3 2.8
## 10  1.3  3.5 35.6 5.0  6.6 76.0  1.2  2.8  4.0  2.3 2.3 1.0 0.4 2.2
## 11  1.8  5.0 36.1 7.2  9.3 77.2  2.1  8.9 11.0  4.6 3.7 1.4 1.3 3.9
## 12  1.9  5.2 36.3 4.7  5.7 83.2  0.6  2.6  3.2  3.4 3.1 0.9 0.3 3.1
## 13  2.5  6.1 40.1 4.1  4.6 90.5  0.7  2.5  3.2  5.8 2.5 1.2 0.3 2.2
## 14  2.0  5.2 38.0 6.3  7.2 88.0  1.1  4.7  5.8  3.5 2.1 1.8 0.7 1.6
## 15  4.1 10.0 41.1 4.1  4.6 89.8  0.8  3.7  4.5  6.6 3.0 1.8 0.2 2.3
## 16  0.6  1.9 33.6 5.2  6.9 76.0  1.8  6.3  8.1  4.9 2.3 0.9 0.4 2.6
## 17  2.0  5.1 39.8 5.0  5.9 84.4  0.7  4.7  5.4  3.5 1.9 1.0 0.3 1.6
## 18  1.6  4.7 33.5 5.9  6.9 84.7  0.8  4.1  4.8  6.3 3.4 1.4 0.5 2.5
## 19  1.6  4.0 40.5 4.1  5.2 79.0  0.8  3.0  3.8  5.8 2.9 1.2 0.2 2.7
```

If you used previous code chunks within the document, this information can be dynamically retrieved and embedded.

```r
# Sets default chunk options
knitr::opts_chunk$set(
  # Figures/Images will be centered
  fig.align = "center",
  # Code will not be displayed unless `echo = TRUE` is set for a chunk
  echo = FALSE,
  # Messages are suppressed
  message = FALSE,
  # Warnings are suppressed
  warning = FALSE
)
# All packages needed should be loaded in this chunk
pkg_list = c('knitr', 'kableExtra', 'magrittr', 'bookdown')

# Determine what packages are NOT installed already.
to_install_pkgs = pkg_list[!(pkg_list %in% installed.packages()[,"Package"])]

# Install the missing packages
if(length(to_install_pkgs)) {
  install.packages(to_install_pkgs, repos = "https://cloud.r-project.org")
}

# Load all packages
sapply(pkg_list, require, character.only = TRUE)
example = read.csv("example.csv")
print(example)
kable(
  head(mtcars, 20),
  format = "latex",
  caption = "This is an example of a table in the Appendix. Notice that it is way too big, and has way 
  booktabs = TRUE
) %>%
  kable_styling(latex_options = c("striped", "scale_down"))
kable(
  head(mtcars, 20),
  format = "latex",
  caption = "This is another example of a ridiculous table. Notice that it is automatically numbered.",
  booktabs = TRUE
) %>%
  kable_styling(latex_options = c("striped", "scale_down"))
```

6.1 Formatting Notes

6.1.1 `R` Code and `rmarkdown`

An important part of the report is communicating results in a well-formatted manner. This template document should help a lot with that task. Some thoughts on using `R` and `rmarkdown`:

- Chunks are set to not echo by default in this document.
- Consider naming your chunks. This will be necessary for referencing chunks that create tables or figures.
- One chunk per table or figure!
- Tables should be created using `knitr::kable()`.
- Consider using `kableExtra()` for better presentation of tables. (Examples in this document.)

- Caption all figures and tables. (Examples in this document.)
- Use the `img/` sub-directory for any external images.
- Use the `data/` sub-directory for any external data.

6.1.2 LaTeX

While you will not directly work with LaTeX, you may wish to have some details on working with TeX can be found in this guide by UIUC Mathematics Professor A.J. Hildebrand.

With `rmarkdown`, LaTeX can be used inline, like this, $a^2 + b^2 = c^2$, or using display mode,

$$\mathbb{E}_{X,Y}\left[(Y - f(X))^2\right] = \mathbb{E}_X \mathbb{E}_{Y \mid X}\left[(Y - f(X))^2 \mid X = x\right]$$

You **are** required to use BibTeX for references. With BibTeX, we could reference the `rmarkdown` paper (Allaire et al. 2015) or the tidy data paper. (Wickham and others 2014) Some details can be found in the `bookdown` book. Also, hint, Google Scholar makes obtaining BibTeX reference extremely easy. For more details, see the next section…

7 References

The **References** section acts as a bibliography for all papers referenced in the **Introduction**, **Related Works**, and **Method** sections. The references should be formated in Chicago author-date format, which is the default for RMarkdown.

- Provide a list (5+) of papers or items you have read to write this proposal.
- Please list all *R* packages or software referenced.

To acquire software citation information, *R* has a built-in command that creates a BibTex and in-line text citation. To generate the citation of an installed *R* package, type:

```r
# In R
citation(package="pkg_name")
```

For example, to cite `dplyr`, one would generate the BibTex entry from:

```r
citation(package="dplyr")
```

```
@Manual{dplyr:2018,
    title = {dplyr: A Grammar of Data Manipulation},
    author = {Hadley Wickham and Romain François and Lionel Henry and Kirill Müller},
    year = {2018},
    note = {R package version 0.7.7},
    url = {https://CRAN.R-project.org/package=dplyr},
}
```

Note, we added a “name” to the autogenerated citation of `dplyr:2018`. Using this name, we can reference the work within the paper via (Wickham et al. 2018) or Wickham et al. (2018).

Allaire, JJ, Joe Cheng, Yihui Xie, Jonathan McPherson, Winston Chang, Jeff Allen, Hadley Wickham, Aron Atkins, and Rob Hyndman. 2015. "Rmarkdown: Dynamic Documents for R." *R Package Version 0.5*.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2018. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and others. 2014. "Tidy Data." *Journal of Statistical Software* 59 (10). Foundation for Open Access Statistics: 1–23.