Problem Set 11

Beth Felkner

April 22, 2024

1 Introduction

I've been a huge baseball fan since I was 11 years old and watching nearly every Texas Rangers game is one of my most favorite past times, so naturally I knew I wanted to do something related to baseball for this project. As big of a Rangers fan as I am, however, I am perhaps an even bigger fan of all the statistics and metrics associated with baseball. Detailed statistics are kept for almost every baseball outcome imaginable, from Batting Average (BA or AVG) to walks to Earned Run Average (ERA) and everything in between. In addition to these "baseline" statistics, today's MLB statscape also includes a plethora of complex constructed metrics such as Wins Above Replacement (WAR) and On-Base Plus Slugging Plus (OPS+) which are designed to capture the overall value of a player to their team. And though it's not one of the metrics that will be included in my algorithms, I would be remiss to write a paper about baseball without mentioning the glorious and officially recognized by the MLB metric of the TOOTBLAN, which stands for Thrown Out On The Bases Like a Nincompoop. All this to say, the game of baseball includes a LOT of statistics.

For this paper, I wanted to focus on a particular baseball outcome that is often not well understood: election to the Baseball Hall of Fame (HoF). Players are considered

eligible for the HoF and automatically added to the ballot if they played in the MLB for at least 10 full seasons and have been retired for at least five full seasons. To be elected, they must receive a vote from 75% of ballot voters. If a player is not elected in their first year, they are eligible to remain on the ballot for up to 9 additional years so long as they received 5% of votes in the preceding year. After 10 years they will be removed from the ballot and considered no longer eligible.

Obviously, players are elected based on some combination of their performance across variables baseball metrics. However, the exact "formula" that distinguishes between HoF-level legends and those who just miss the mark is not clear. For this project, I used machine learning tree models to predict the HoF status of a player based on their performance across a number of hitting metrics.

2 Literature Review

Below are some sources I've found so far that I think could be helpful and may use in my literature review. But this part is definitely a work in progress and I need to dive deeper into these sources so that I can actually write the literature review and also look for other sources.

- Koseler and Stephan [2018]
- Ishii [2016]
- Hoang [2015]
- Valero [2016]

3 Data

I am using Hall of Fame Voting data from baseball-reference.com for the years 2015-2014. The datasets are published for each year on baseball-reference.com and include all players who became ballot eligible for that ballot year (meaning all players who played for at least 10 seasons and had been retired for five years as of that ballot year.) As a note, I had some confusion while I was compiling my data about if the baseball-reference.com datasets included all eligible players in each year or just a subset. I started with the baseball-reference data, but then I confused myself into thinking it was only a subset, so I moved towards data from a separate database of all retires from each year. But then but after I begin compiling that data and filtering it to only 10-year players, I realized that it was exactly the same data as the pre-compiled datasets from baseball-reference.com, which did in fact included all eligible players for each year. So I went to back to my originally datasets from baseball-reference.com and that is what I used.

I imported the datasets into R through direct web-scraping (I did not use an API). After importing each of the 10 individual year datasets, the first thing I did was manually add a HoF Yes/No dummy variable for each player, because this was conveyed on baseball-reference.com with a gold highlight which did not transfer over into the web-scraped data. This was easy because the HoF players were at the top of each dataset so I just had to see how many rows were highlighted in each baseball-reference dataset and then assign "Yes" to that many rows in each R data set and "No" to all other rows (I used an ifelse statement to so this.) I initially made this as a character variable, but later converted it to a 2-level factor when I realized that my tree model required a factor variable prediction output.

Next, I vertically bound my 10 individual datasets into one combined dataset using rbind. Then I began to do some cleaning and filtering. First, I changed some of the variable names because there were duplicates for things like G Games) and H (Hits)

on the batter and pitcher side, so I just renamed them as `Batter_G`, `Pitcher_G`and so forth. Then I removed some addendums to the player names, "X-" which indicates their last year on the ballot and "HOF" which indicates they were inducted in a later year. I also converted all of the statistic variables to be numeric variables, rather than the character type they defaulted to when I imported the data.

After that, my next step was that I needed to remove duplicates (players who appeared on the ballot multiple years) because it would be the same set of career statistics for each appearance. I used a slice(n) function to keep only the latest chronological appearance of each player name, which preserved the induction-year voting statistics for players who had appeared more than once (their career baseball statistics would be the same in each appearance). Lastly, I filtered my dataset to only include position players and exclude pitchers. I did this because I chose to focus on offensive hitting metrics only in my prediction model, and thus I wanted all my datapoints to be people who frquently hit and were good at it, not pitchers hit much less frequently than position players and usually significantly worse than league average.

(Note that I may also create a separate dataset of pitchers only and run separate models with that data, but I haven't decided yet because the sample size would be so small I'm not sure if it would be useful. But if I did decide to, creating that dataset would be easy).

After all this filtering, I am left with a dataset of only 123 observations. I know that this is very small for a machine learning dataset and that my models are likely weakened by such a small dataset, but there is unfortunately not a good option to expand it. If I go farther back, my data becomes less accurately predictive because baseball and all its metrics have changed so much in the modern era. If I went back further to the 2010 ballot for example, I may have players who were playing in the early 1990s which was a very different game than the current era I'm trying to predict.

(Note to Dr. Ransom on my draft: I know we already talked about this once but I am very open to suggestions if you have any of how to expand my sample. But as I discussed above I was confused about my original datasets and they already included all HoF eligible players so there are no more players from the same years I can add unless I went to less than 10-year players).

Summary statistics and explanation of variables ... (will add later)

4 Methods

I plan to use a tree model algorithm and classification mode. Depending on how this goes I may also run a KNN or some other kind of algorithm, but the tree model is what I have been working on thus far. I used 3-fold cross validation, and my equation for which I am seeking to make out-of-sample predictions is:

$$P(HoF = Yes|X) = \frac{1}{1 + e^{-(z)}} \tag{1}$$

where z is equal to:

$$\begin{aligned} z = \beta_0 + \beta_1 \text{Years} + \beta_2 \text{WAR} + \beta_3 \text{WAR7} + \beta_4 \text{JAWS} + \beta_5 \text{Jpos} \\ + \beta_6 \text{G} + \beta_7 \text{HR} + \beta_8 \text{RBI} + \beta_9 \text{SB} + \beta_{10} \text{BB} + \beta_{11} \text{BA} \\ + \beta_{12} \text{OBP} + \beta_{13} \text{SLG} + \beta_{14} \text{OPS} \end{aligned} \tag{2}$$

5 Findings

So this is where I am really not sure, and I plan to come to office hours this upcoming week to talk about my results and how to interpret them because I am confused. I ran my tree model with the above equation and the same parameters we used in Problem Set 10, and I got a mean accuracy of 0.867. But then out of curiosity I also ran a tree

model with a formula of only (Hof Years) as a test. I expected that this would be significantly less accurate, but to my surprise the mean accuracy was almost exactly the same. So I think I must either be missing something in my coding or misunderstanding something in my interpretation.

6 Conclusion

Because I don't fully understand my results yet, I can't really say a lot in terms of conclusion right now. I should be able to update this easily after I come to office hours. One conclusion I can comfortably make right now is that my results would be stronger with a larger sample size; I just don't know to achieve that in this particular case.

In terms of future research, I think there are so many ways in which machine learning (and also traditional econometrics) can be applied to the world of baseball. Baseball has always been the most statistics-heavy sport, and we have more power to draw insights from those statistics now than in the past. I think a new-age "Money Ball" tactic, in which a team uses ML algorithms to predict outcomes for a bunch of different player combinations and predict which hypothetical teams would be most successful at different price points, would be especially intersting. It makes me wonder how much of that is already going on internally at various MLB teams. I hope the field of MLB internal analytics continues to expand in the future because working in statistics or analytics for an MLB team would be pretty much my dream job.

References

Phuong Hoang. A dynamic feature selection based lda approach to baseball pitch prediction. *Trends and Applications in Knowledge Discovery and Data Mining*, pages 125–137, 2015.

Tatsuya Ishii. Using machine learning algorithms to identify undervalued baseball players. *Stanford Machine Learning*, 2016.

Kaan Koseler and Matthew Stephan. Machine learning applications in baseball: A systematic literature review. *Applied Artificial Intelligence*, 31:745–763, 2018.

Cesar Soto Valero. redicting win-loss outcomes in mlb regular season games – a comparative study using data mining methods. *International Journal of Computer Science in Sport*, 15, 2016.