Predicting 2015 Fantasy Baseball Statistics

Dan Loman

March 31, 2015

1 Introduction

Lately I've been falling behind with my baseball knowledge, so when my annual fantasy draft began to approach I realized I was more unprepared than usual to draft my best team. As a competitive fantasy sports player I still wanted to gain an edge, so I turned to using analytics. My goal for this project was to use machine learning on historical baseball statistics to predict player fantasy statistics for the 2015 season, and use those predictions to identify the best fantasy value players.

In this paper I'll walk you through my methods of data acquisition, data preprocessing, machine learning and analysis for forecasting the 2015 fantasy baseball season.

2 Data Acquisition

The most time consuming part of this project was the data acquisition. For the data acquisition I scraped data from baseball-reference.com using Python and BeautifulSoup4 to end up with statistics for every player in their database. This process was a two step approach:

First, I had to find the urls of every player on baseball reference. Baseball reference sorts players alphabetically, with each letter having it's own page, and with a link to each player whose last name begins with that letter on that page. I extracted the urls by looping through every letter, finding where the player urls were located and doing some regular expressions to extract each one.

Next came the hard part - using these urls to extract player statistics. What made this process so difficult was the sheer amount of time it took to scrape and search through over 18,000 urls. By my estimation, it took roughly 9 hours to complete the process, so with taking my typical bumps and bruises it took almost all weekend just to extract the data I need. A few things that threw me off:

Data inconsistency: I quickly found that there were a few minor inconsistencies within some of the baseball reference tables. While this isn't an issue for folks browsing the site, it threw a few errors for me scraping the data. I got past this by overdosing on try/except statements, even in places I probably didn't need them but wanted to be safe.

Memory overloading: The biggest issue I faced with data acquisition was the amount of data I was dealing with. I began naively assigning each statistic to a list, but quickly found out that after scraping a few thousand players I ran out of memory pressure, and it wouldn't work. Using a dictionary helped, but it still wouldn't be sufficient for storing data on 18,000 players. What I ended up doing was only looping through 800 urls at a time, storing the data in a dictionary, saving to a json and then resetting the dictionary. I ended up with a few dozen json files for both batting and pitching stats but it worked and didn't overload my memory space. In the data preprocessing phase I combined these json files into 2 neat batting and pitching dataframes, which I then stored to a mysql database.

3 Data Preprocessing/Statistics

Once I had the data scraped, I had to preprocess it to use for machine learning. As mentioned above, the first thing I did was convert my json data into batting and pitching Pandas dataframes. These dataframes contain every player on baseball reference, with selected stats for each year (with a few exceptions).

My ultimate goal for this project is to predict fantasy baseball statistics for standard ESPN leagues, making my target variables the following:

Batting: Batting average, home runs, RBIs, runs, stolen bases
Pitching: Wins, ERA, WHIP, strikeouts, saves.

Ultimately I divided each of these stats (except for rate stats like batting average) by the number of games they were compiled in, so I was working with per game statistics. Therefore, games played and innings pitched also became crucial target variables because I would need to predict how many games the player would play in to get the total counting stats.

I also collected these statistics, plus player age, to supplement the primary stats:

Batting: At bats, hits, doubles, walks, and strikeouts
Pitching: Walks/9, home runs/9, hits/9, and K/9.

For each observation in my training and test data I used the above statistics lagged 1, 2 and 3 years to project stats for the current year. If the player hasn't been in the league long enough, I simply assigned it as NaN.

My training data consists of every historical baseball season since 1960, with thresholds of over 50 at bats for batters and over 10 innings pitched for pitchers. To extract my test data I looked at every player's season in 2014 and bumped up the year, player's age, and lagged stats. Since I use 2014 players for my 2015 test, a nontrivial problem with my test data is that it includes players who've retired and doesn't include rookies.

4 Machine Learning

To restate the purpose of the project, my goal was to predict 10 fantasy statistics for 2015. This included 5 variables for both batting and pitching, along with each player's games played and innings pitched for pitchers. Therefore I ran 12 different machine learning models, 1 for each target statistic.

I tested my machine learning models on my training data, using a 90/10 train/test split and 10 fold cross-validation. I used mean absolute error (MAE) to evaluate the performance of my models.

I started machine learning on my data using a random forest classifier, because random forests have typically been my strongest performing machine learning algorithm in my practicum and in class. Though they're meant to be used for classification, I found during my work at Flyr that they can work quite well for regression as well. However, when I ran my first random forest algorithm I found the results to be very poor, so I made some adjustments.

First, I adjusted my features for each target. My initial goal was to use all or almost all of my features for each target variable, but I found that adding too many stats cluttered my algorithm, and that fewer predictor variables lowered my MAE. Eventually through some trial and error I settled on simply using the player's age and particular lagged stats to predict each stat (example: features to predict a player's home runs are age, home runs lag 1, home runs lag 2, home runs lag 3). I'm not positive that this is the optimal solution, but I got good results and it is easy to interpret.

Next, I switched my model from a random forest to a linear regression. To my surprise, my results improved dramatically:

MAEs for batting stats	**Runs**	**Bat Avg**	**HR**	**RBIs**	**SB**	**Games**
Random Forest	21.27	.040	6.42	22.00	3.82	35.75
Linear Regression	14.92	.027	4.59	15.80	3.02	24.60

Since I am using lagged data to train and test my regressions, I also had to split my data into 3 separate groups: 2nd year players, 3rd year players and veterans. I didn't include rookies in my model because they wouldn't yet appear on baseball-reference.com. This is because 2nd and 3rd year players haven't yet compiled data for all of their lagged stats, so their models had to be trained differently. Unsurprisingly, my model's results suffered for less experienced players:

MAEs for batting stats	**Runs**	**Bat Avg**	**HR**	**RBIs**	**SB**	**Games**
Veterans	14.92	.027	4.59	15.80	3.02	24.60
3rd year	16.53	.030	5.51	16.01	3.63	26.74
2nd year	18.16	.033	6.73	17.82	3.66	30.39

5 Analysis

At this point I was extracted predictions for each player's statistic for 2015, now I had to make sense of it. The first problem in analyzing these projected stats was dealing with ERA and WHIP. These stats are tricky because you want to minimize them while maximizing innings pitched. I ended up using the following metric for scoring ERA and WHIP:

$$ERA\_score = (pred\_ERA_{max} - pred\_ERA) * pred\_innings \tag{1}$$

$$WHIP\_score = (pred\_WHIP_{max} - pred\_WHIP) * pred\_innings \tag{2}$$

To rate players as the sum of their parts I created a rating system similar to ESPN's player rater. I assumed normality of each statistic (admittedly a potentially dangerous assumption) and found the z score of each stat for each player to find each player's statistic scores. Finally, I took the sum of all statistic scores to get a composite rating for each player.

6 Results

Comparing my results to ESPN's fantasy projections, my results seem reasonable. Though there are some surprises, we generally have the same players at or near the top of our projections. We share 5 of the top 6 batters and 9 of the top 10 pitchers, with some variation in the order.

I won't get in too far into my results here because I wrote about it in more detail at:

https://greenandgoldanalytics.wordpress.com/2015/03/31/fantasy-baseball-2015-preview/

7 Issues

Unfortunately, I noticed that several players didn't make it through my data acquisition phase and to the final result. Given that two of those players are Jose Abreu and Yasiel Puig, two of ESPN's top 20 batters, I was somewhat disappointed.

Comparing my results to ESPN, I noticed a few stark differences. First, my rating system gives a large penalty to older players, since I use age as a predictor for every target statistic. Second, my model strongly favors the top base stealers, which indicates that it is poor at predicting an average amount of stolen bases. And finally, my results are very pessimistic overall, especially when projecting games played. Of my final batting predictions, no one projects to play over 150 games or hit above .292. This may indicate that I should try eliminating poorer players from my training data. However, while these pessimistic projections presents problems for accurately predicting stats, it works for my purposes because every player is affected so my overall rankings should stay the same.

Obviously, the final major concern is context. Since I restricted myself to using only what I could get from baseball reference, my test data is extremely naive and doesn't account for numerous external factors. If a player switch teams, has a bigger role, or is already injured to start the year, my model won't take that information into account. That's why even though I have this model, it's crucial I'm still on top of current baseball knowledge so that I can correctly interpret it.

8 Future considerations

As with my previous project where I forecasted the 2015 NCAA Tournament, my fantasy baseball project is one which I would like to turn into a multi year endeavor. Due to time constraints and a busy grad school courseload there were a few things I wasn't able to accomplish with this project but would like to try for future fantasy baseball seasons. In addition to addressing the problems described above, here are a few things I may consider if I revisit this project in the future:

More extensive feature engineering: I was able to achieve good results using a linear regression and fairly simple regressors, however I didn't conduct an exhaustive feature selection. I believe there may be better combinations of training features for each stat, and using something like an iterative leave one out approach for each target statistic might help me find the optimal combinations.

More detailed linear regression: I used a simple linear regression model to make my predictions. In the future it would be worth experimenting on polynomial regression or even other regression algorithms to improve results.

Positional value: In this study I separated players only in two categories: batters and pitchers. As many fantasy players know, there is more value in players who play a scarce position, or who play multiple positions. In the future I'd like to consider positional value as part of my player ratings. This would also allow me to combine batters and pitchers into one comprehensive player ranking.

9 Conclusion

I took on this project not to create a comprehensive and accurate list of player rankings, but to identify a few players who may be under or over drafted. In that sense I believe I succeeded. However, my prediction methods could use some further examination and I need to look more closely as to why players were being left out.

I'm glad I got the opportunity to apply what I've been learning in school to fantasy sports, and I was happy to see that I was able to achieve some decent results. There remains much to test and improve on for this project, but I believe I have a solid foundation in place for next spring.