Suhas Buravalla and Carter Rogers

Professor Lo

DATS6501

10 May 2023

Data Science Capstone – Fantasy Baseball Projections

Overview

As part of the fulfillment of the M.S., Data Science completion criteria, we did our capstone on *Fantasy Baseball Projections*. In this report, we will provide background on our topic, why we selected this topic, and our problem statement. We will then give an overview of the data set, features and targets, and our approach and methodology to the modeling process. To conclude, we will share results of the modeling process and lessons learned.

Background

Major League Baseball is one of the most popular sports leagues in the United States. Within the last 20 years or so, the concept of "fantasy" sports has become increasingly popular; in fantasy sports, the performance of a real-life player dictates how your virtual fantasy team performs. Fantasy baseball is a common game for sports fans to participate in, and the most popular ruleset for fantasy baseball is the Rotisserie 5x5 Standard format. In this method of scoring, there are 5 stats for hitters, and 5 for pitchers. They are: Home Runs (HR), Runs scored (R), Runs Batted In (RBI), Stolen Bases (SB), and Batting Average (AVG) for hitters, and Wins (W), Saves (SV), Earned Run Average (ERA), Strikeouts (SO or Ks), and Walks + Hits per

Inning Pitched (WHIP). These are very simple, surface-level stats that make it easy to track scoring for a player's on-field performance.

Baseball is one of the most data-driven, analytically advanced sports played today because of its complex strategy and unique playing style. On a basic level, baseball is a simple sport where an athlete hits, runs, and throws the ball. However, advancements made in the last few decades have placed data at the forefront of Major League Baseball. "Moneyball", first a book written by Michael Lewis and later adapted into a movie, made sports analytics a more mainstream concept that revolutionized the way baseball is played and studied. However, many of the revelations discovered by the "Moneyball" Oakland Athletics' baseball operations department, such as valuing a player's ability to get on base over their batting average, are out-of-date and have become common thinking among the sports' decision-makers. The level of analysis runs much deeper now, to the extent that teams are employing entire departments filled with data scientists and data engineers to help drive their decision-making processes. In today's Major League Baseball environment, teams are using pitch-tracking data from Statcast machines as well as biometric data to capture player movements. Most of this data is not publicly available, but it is widespread inside the industry and is being used to gain a competitive advantage over other teams.

The 10 fantasy scoring stats used in the standard Rotisserie 5x5 format are very basic, and only provide a high-level view of a particular player's performance. This works very well in the context of fantasy sports, as it makes the game accessible to even a casual baseball fan, but it only scratches the surface in terms of how data is being used in baseball today. The 30 Major League teams, as well as experts outside of the sport, are constantly developing new ways to measure and predict player performance. For the real-life sport of baseball, being able to predict

how a player will perform can give a team a significant competitive advantage in trades and player acquisitions. In fantasy baseball, predicting future player performance is critical to winning your fantasy baseball league, which is especially useful for those who play fantasy baseball for money (for context, the National Fantasy Baseball Championship will hand out over $200,000 in prize money upon completion of the 2023 season). For a highly competitive fantasy baseball league, simply using surface-level stats to predict how a player will perform is not nearly enough. Industry experts have developed their own projection systems, such as Dan Szymborski's ZiPS, Derek Cardy's The BAT X, Ariel Cohen's ATC, and several others. Without being able to peek under the hood, it is difficult to know the methodologies used to develop these projection systems, and all have their own strengths and weaknesses. For example, The BAT X is notorious for regressing player performance towards the mean, which works well across the general population but will often miss on players on opposite ends of the spectrum.

Problem Statement

After surveying the current Major League Baseball and fantasy baseball landscape, our problem statement becomes clear: our goal is to develop a predictive model using real-world baseball data to project future fantasy baseball results.

Prior Research

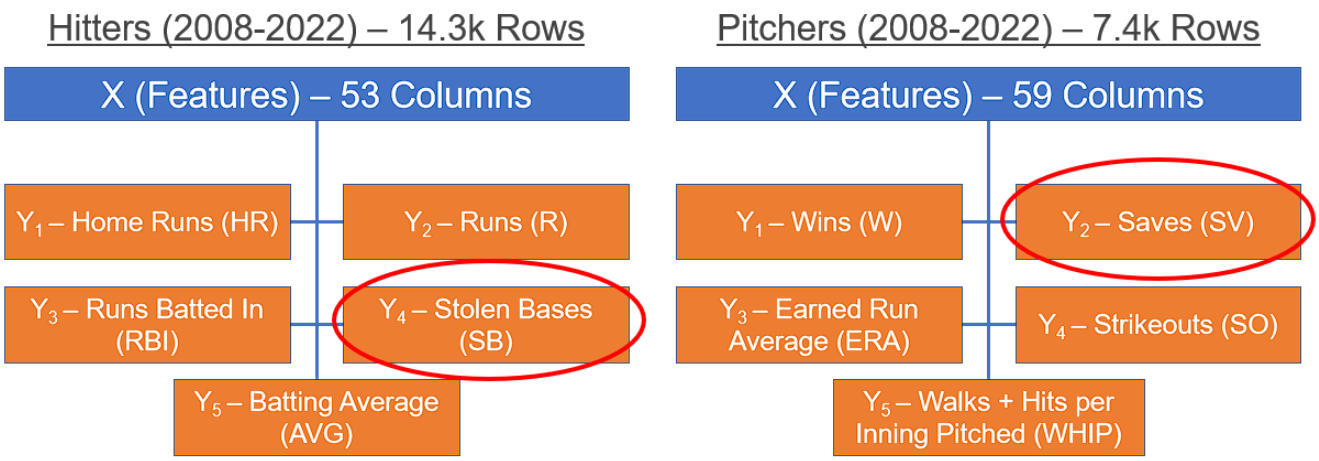
There are relevant pieces of research available that provide some insight to which metrics may be useful in predicting player performance. Eno Sarris of The Athletic has found that Barrel%, the percentage of plate appearances resulting in a batted ball with optimal exit velocity and launch angle, is more predictive of future power than any other metric. This is valuable for our purposes because at least 3 of the 5 fantasy baseball stats for hitters (Home Runs, Runs, and

RBIs) are related to a hitter's ability to hit for power. Alex Fast of PitcherList.com developed a new metric called CSW%, or Called Strikes + Whiffs Percentage, that was found to stabilize in smaller samples and is highly correlated with strikeouts. This is important for our project for two reasons: first, we value metrics that are meaningful in small samples because part of being able to predict a player's performance is making accurate predictions even when the sample size (how many times a hitter bats or a pitcher pitches) is relatively low. Second, one of the fantasy baseball stats for pitchers is Strikeouts, so we are interested in metrics that may hold predictive power for Strikeouts. Finally, Ben Clemens of Fangraphs.com found that Exit Velocity (how hard a ball is hit) and Contact Rate (how often a hitter makes contact with the ball) are the 'stickiest' metrics from year-to-year. Contact Rate is highly correlated with Batting Average, so we expect a hitter's batting average to be relatively stable from year-to-year.

Our Data Set

Fangraphs.com is an extremely valuable resource for Major League Baseball data and serves as a historical repository for basic and advanced baseball metrics dating back to at least 1901. However, the further back in time we increase our scope, the less reliable the data records become. We will limit the scope of this project to players who played in Major League Baseball from 2008-2022, as per-pitch data started to become available in 2008 with the introduction of new pitch-tracking technology. In 2015, Major League Baseball installed Statcast machines at all 30 ballparks that can track advanced metrics on each individual pitch, such as how hard the ball was hit and the angle at which it was launched off the bat. We pulled a wide range of features, both basic and advanced metrics, from Fangraphs.com for both hitters and pitchers from 2008-2022 to comprise the data set used for this project. We then merged consecutive seasons for each player to obtain the features and targets. All columns from the second year outside of the targets

were dropped, as they are not needed for the scope of this project. We imputed the missing Statcast metrics for the years in which they were unavailable (2008-2014, before the introduction of Statcast).



The features for both the Hitters and Pitchers consist of basic stats, such as the prior year's fantasy stats, games played, age, and team, as well as more advanced metrics such as batted ball quality and swing decisions. For Pitchers only, we have features that describe how often a pitcher throws a specific pitch (for example, 60% fastballs, 30% curveballs, and 10% changeups). The 5 targets for Hitters are: Home Runs, Runs, RBIs, Stolen Bases, and Batting Average, corresponding with the 5 Rotisserie 5x5 Standard fantasy stats. For Pitchers, the 5 targets are: Wins, Saves, ERA, Strikeouts, and WHIP. Our goal is to develop a model using data science techniques to predict the 5 targets for both Hitters and Pitchers (10 targets in total).

Modeling

Our approach to modeling will need to result in building a multi-target regression model to successfully predict the 10 targets with some degree of accuracy. The first step is to identify a baseline model with which we can compare the performance of our model. The baseline model

we will use for this project is the naïve model, which simply predicts that a player's future performance will be the same as their previous year's performance. The baseline naïve model performs the worst with AVG (Hitters), ERA, and WHIP (pitchers). These targets are rate stats that tend to fluctuate from year-to-year based on factors that can often be out of a player's control. For example, if a team's defense is poor, their pitchers' ERA will suffer due to fewer balls in play being converted into outs. We anticipate that our model will struggle to predict these targets accurately. Likewise, our model may struggle to predict Stolen Bases (Hitters) and Saves (Pitchers) as these targets are largely unrelated to the other targets. For example, hitting a home run will also grant you at least one RBI and one Run, while also improving your Batting Average. Stolen Bases are dependent on team context, as some teams prioritize stealing bases to score runs, while others ignore it as a viable strategy. Stolen Bases are also not strictly a function of how fast a player is, as even some fast players choose not to steal for a variety of reasons. Saves are awarded to a relief pitcher for coming into the 9th inning and preserving a close lead. Relief pitcher performance in general is very difficult to predict as their performance tends to fluctuate year-over-year because they often struggle to consistently throw strikes. Relievers are almost always former starting pitchers who had to convert to a relief role because of their ineffectiveness; it is easier to perform well over short stretches (1-2 innings at a time) than to handle a starting pitcher's typical workload of 5-6

Target*	Naïve Model MSE
HR	0.516
R	0.419
RBI	0.434
SB	0.521
AVG	0.832

Target*	Naïve Model MSE
W	0.841
SV	0.717
ERA	1.816
SO	0.695
WHIP	1.762

*normalized

innings per appearance. To make matters worse, the closer, or the designated reliever to get opportunities for Saves, may not be the best relief pitcher on their own team.

There will be limitations to our modeling process based on the availability of data and factors that are beyond the scope of this project. The biggest limitation to our modeling approach is that we cannot account for injuries. Professional athletes are always at a high risk of injury, but it's especially true for pitchers due to the unnatural and stressful method used to throw a baseball. The extreme stress on the arm, specifically the elbow and shoulder, causes pitchers to get injured at an extremely high rate. Our model will not be able to take into account a pitcher's risk of injury because it is very difficult to quantify. The number of innings a pitcher threw is one of the features present in the data set, which serves as a rough estimate for how injured they were the prior year. We also cannot take into account real-life transactions such as trades and free agent signings. These do have an impact on a player's performance, but there is no way to predict them in a way that a model can interpret due to the subjective nature of the analysis required to evaluate the impact on performance. Finally, Major League Baseball is constantly tweaking rules and regulations around the game in ways that affect player performance but are difficult to account for in modeling. For example, in 2019 the league used balls that flew further in the air which led to a league-wide spike in Home Runs. A model trained on data from 2018 would not be able to account for this change.

To determine our approach to modeling, we need to first examine the attributes of our specific problem. Both of our data sets are very wide, with 53 and 59 features for Hitters and Pitchers respectively, but only ~14,300 and ~7,400 observations. Because of this, we expect Pitchers to be more difficult to model as the data set is nearly half the size of the Hitters data set. We prefer models with native support for multi-target regression output, inherent feature

selection (due to our wide data sets), and better performance on data sets with fewer observations. The reason we are choosing to build 2 multi-target regression models as opposed to 10 separate models for each target is because our targets (outside of Stolen Bases and Saves) are all related to each other. Modeling each target separately would add a layer of complexity to the project that may be overwhelming to deal with.

Our model selection process leaves us with several options to choose from. One option would be a Linear Regression model, but we quickly ruled this out due to the feature selection process needed for an effective LR model. Because LR models are prone to overfitting if trained on too many features, it is not a good choice of model for our problem as we simply have too many features present in the data. K-Nearest Neighbors Regression is another option, but the KNN algorithm scales poorly with increased dimensionality, so we ruled this out as well. We ultimately decided to use both a Decision Tree Regression (DTR) model and a Multi-Layer Perceptron (MLP) neural network on both data sets, for a total of 4 models. Both DTR and MLP support multi-target regression and have inherent feature selection; the DTR algorithm only uses a subset of the features it's trained on, while the MLP adjusts the weights and biases of each feature during the training process. The only downside is that MLP networks, like many neural networks, perform best with more observations to train from.

The architecture we used for both MLP models consisted of an input layer, two dense layers (128 and 64 neurons each), and 5 separate output layers all connected to the second dense layer. We chose to connect the output layers to the same dense layer because of how the targets are all related to each other. The relatively low number of neurons for the two dense layers was chosen because of the size constraints of our data sets; we didn't want to use an overly complex model that would struggle due to a lack of available training data. We chose to use the Adam

optimizer during training, with a learning rate of .00001, and trained both models for 1,000 epochs. The number of features used for training the MLP models differs from the number of features used for the DTR models because we converted the 'Team' variable (the only categorical variable present in the data set) to a dummy variable, giving us 31 extra features for the MLP models. The DTR models were both trained using a maximum feature space of the square root of all available features and 10-fold cross validation.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 84)]	0	[]
dense (Dense)	(None, 128)	10880	['input_1[0][0]']
dense_1 (Dense)	(None, 64)	8256	['dense[0][0]']
HR_y_output (Dense)	(None, 1)	65	['dense_1[0][0]']
R_y_output (Dense)	(None, 1)	65	['dense_1[0][0]']
RBI_y_output (Dense)	(None, 1)	65	['dense_1[0][0]']
SB_y_output (Dense)	(None, 1)	65	['dense_1[0][0]']
AVG_y_output (Dense)	(None, 1)	65	['dense_1[0][0]']

Total params: 19,461
Trainable params: 19,461
Non-trainable params: 0

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 96)]	0	[]
dense (Dense)	(None, 128)	12416	['input_1[0][0]']
dense_1 (Dense)	(None, 64)	8256	['dense[0][0]']
W_y_output (Dense)	(None, 1)	65	['dense_1[0][0]']
SV_y_output (Dense)	(None, 1)	65	['dense_1[0][0]']
ERA_y_output (Dense)	(None, 1)	65	['dense_1[0][0]']
SO_y_output (Dense)	(None, 1)	65	['dense_1[0][0]']
WHIP_y_output (Dense)	(None, 1)	65	['dense_1[0][0]']

Total params: 20,997
Trainable params: 20,997
Non-trainable params: 0

Results

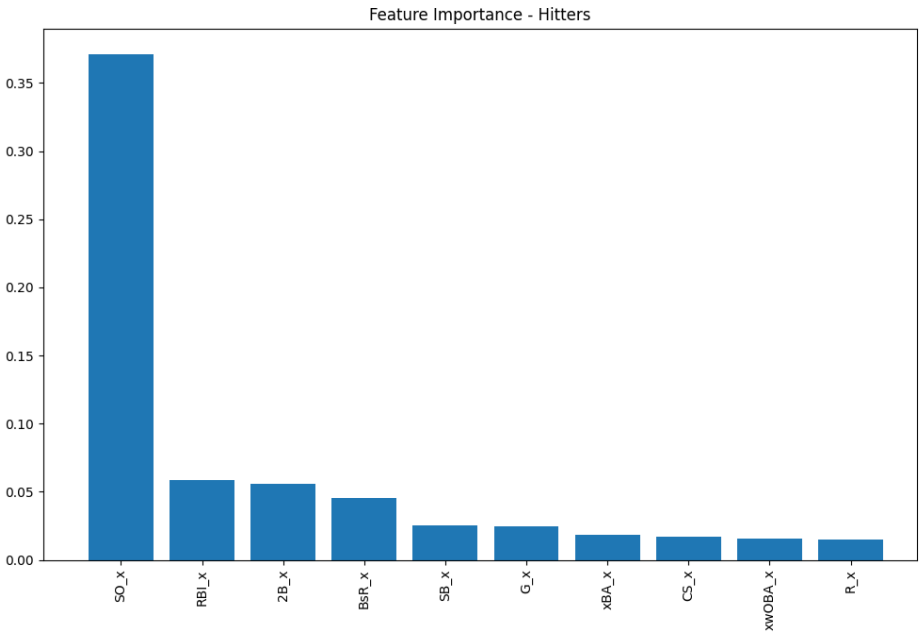
The MLP models for both Pitchers and Hitters outperformed the baseline naïve models, while the DTR models performed very poorly and were unable to outperform the baseline, using Mean Squared Error (MSE) to evaluate model performance. As expected, the MLP model for Hitters performed much better than the MLP model for Pitchers. We suspect this is because there were more observations for the Hitters MLP model to train from, as well as the fact that Hitters'

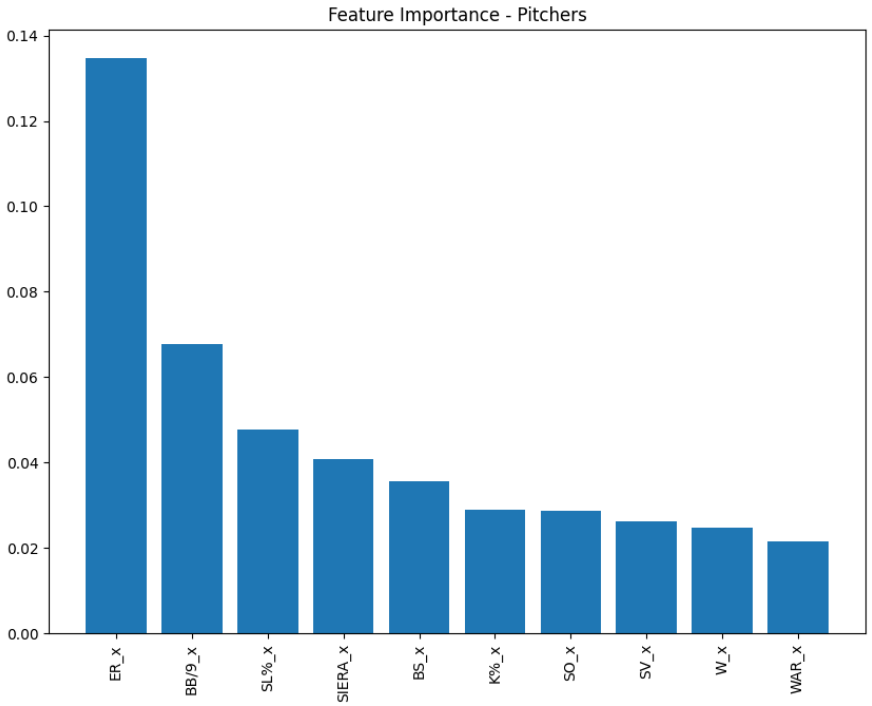
performance is generally easier to predict than Pitchers' performance, which is reflected in the baseline MSEs for both. As expected, the MLP for Hitters performed the worst on the Stolen Bases target. The MLP for Pitchers performed the worst on ERA and WHIP, the two rate stats that are subject to the most year-over-year fluctuations. The DTR models performed much worse than expected; in examining the feature importance plots, it appears that the DTR model for hitters chose to prioritize counting stats over metrics that measure the quality of a player's performance. The top three features in terms of feature importance for the DTR Hitters model were Strikeouts, RBIs, and Doubles, all stats that increase as a Hitter's games played and plate appearances increase. However, these stats have very little predictive power in how well a player will perform, especially in the case of Strikeouts which are the worst outcome for a hitter. The model trained itself to predict that players with lots of Strikeouts will also accumulate Home Runs, Runs, and RBIs. This may be true for some players that are able to post good fantasy stats despite striking out a lot, but will lead to some incredibly misguided predictions with high error values. Similarly, the highest feature in terms of feature importance for the Pitchers DTR model was Earned Runs (not Earned Run Average, the rate stat). Pitchers will give up more runs the more they pitch, but they should not be rewarded for giving up more runs. Pitchers will

Model Performance - Mean Squared Error (MSE)			
Target	Baseline - Naïve	Decision Tree Regressor	Multi-Layer Perceptron
HR_y	0.516	0.902	0.424
R_y	0.419	0.664	0.339
RBI_y	0.434	0.710	0.352
SB_y	0.521	0.699	0.507
AVG_y	0.832	1.030	0.463

Model Performance - Mean Squared Error (MSE)			
Target	Baseline - Naïve	Decision Tree Regressor	Multi-Layer Perceptron
W_y	0.841	1.184	0.592
SV_y	0.717	1.133	0.589
ERA_y	1.816	2.122	1.007
SO_y	0.695	1.198	0.510
WHIP_y	1.762	2.286	0.890

accumulate more Strikeouts and Wins simply by pitching more often, but their ERA and WHIP will suffer the more runs they allow. As a result, the DTR model for Pitchers performed poorly by training itself to inaccurately determine that Pitchers who gave up lots of Earned Runs will have strong fantasy stats, leading to high error values.





Conclusion

We were able to successfully develop a model that projects a player's fantasy baseball performance based on their metrics from the prior year more accurately than a baseline naïve model. The models performed the best on Runs (Hitters) and Strikeouts (Pitchers), while performing the worst on Stolen Bases (Hitters) and ERA (Pitchers). In general, we were able to predict Hitters' performance more accurately than Pitchers' performance. There are many reasons for why this is the case; first, hitters tend to get injured less frequently than pitchers. Hitting is far less stressful on the human body. Second, team context is more important for pitchers, in that the team they play for has a more noticeable impact on their performance. Pitchers on bad teams are awarded fewer Wins, and their rate stats (ERA and WHIP) tend to be worse as bad teams often struggle on defense. Finally, we chose to lump all pitchers together in the same data set, but in real life there are two primary types of pitchers: starting pitchers, and relief pitchers. Starting pitchers will never get Saves but will have many more Strikeouts provided they pitch a full season.

This project only scratches the surface of what can be achieved using baseball as a platform for data science. The possibilities of what can be done using predictive modeling are almost limitless provided one has access to per-pitch data. Unfortunately, this data is not available to the public, but it can be used in interesting and insightful ways. Currently, there is a Kaggle competition to predict whether a pitch will result in a swing based on the pitch's velocity, spin rate, and a variety of other factors. New metrics are being developed every year to both describe past performance and predict performance in the future. America's pastime has endless possibilities when it comes to the realm of data science.

References

Sarris, Eno. “Sarris: 10 Bold Predictions for the 2023 MLB Season.” The Athletic, 24 Mar. 2023, https://theathletic.com/4312841/2023/03/24/sarris-ten-bold-predictions-mlb-2023/.

Fast, Alex. “CSW Rate: Intro to a New Fantasy Baseball Metric.” Pitcher List, 5 Dec. 2022, https://www.pitcherlist.com/csw-rate-an-intro-to-an-important-new-metric/.

Clemens, Ben. “You Can't Fake Exit Velocity.” FanGraphs Baseball, 6 Feb. 2023, https://blogs.fangraphs.com/you-cant-fake-exit-velocity/.

Link to GitHub repo: https://github.com/suhasburavalla/Fantasy-Baseball-Projections