Prediction of Batting Averages in the MLB Through Machine Learning

Joel Miller
Electrical and Computer Engineering
jgmiller@andrew.cmu.edu

Christopher Ng
Electrical and Computer Engineering
cyn@andrew.cmu.edu

Mark Prettyman
Electrical and Computer Engineering
mprettyman@cmu.edu

1 Problem Statement

In Major League Baseball, batting average (BA), which is calculated as a ratio of a players' hits to total at-bats (AB), is the primary means of evaluating a players batting performance. While this statistic is trivial to calculate and compare against other players, it is far more difficult to predict from year to year for a given player. This research aims to generate a model using machine learning to predict a players' BA for the length of an entire regular season, which consists of 162 games, with a high degree of accuracy. This will be accomplished through determination of what features most affect a player's batting average, what model will provide the best prediction level, and what hyperparameters will yield the most accurate results. The outputs of our model may prove valuable to major league organizations looking to gain a better understanding of potential performance of their or other teams' players.

2 Data

This study uses a data set of players' individual season summary statistics. Lahman's Baseball Database, which is a freely available source that aggregates a players batting statistics over a season is parsed and stored into a MySQL database that is used to gather data. While the database record statistics from 1871 onwards, only the statistics from 1990 to the current year are used. This is because of the many rule changes that baseball has undergone since 1871 and the reliability of the statistics before modern era is questionable. Choosing 1990 includes all of the major rule changes in the MLB while still providing ample data to train the model on.

2.1 Features

For features, the study uses the current year, a player's previous years batting average, their current age, their walk rate (walks per AB, shown BB), their strikeout rate (strikeouts per AB, shown SO), and whether they played in the NL or the AL. The number of at bats that a player had in the previous years is also taken into account in order for the model to adjust for the validity of the previous year's statistics. For example, a high previous batting average but with low at-bats should be less weighted the same batting average with more at-bats.

In the real world, the model would have access to all of these features, so the prediction scenario is realistic. Age is chosen as a feature because many players go through athletic declines in their career. The model can be trained to adjust for aging players as their batting averages are likely to decline as they near retirement age.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Previous year's walk rate is chosen as a feature because pitchers sometimes walk players that they believe are better, in an attempt to be risk adverse and not let them hit, making walks a possible indicator feature as a predictor for better batting averages. The previous year's strikeout rate is chosen as a feature because the more a player strikes out, the less balls he puts in play, which would decrease his batting average. Finally, NL and AL are chosen as a feature due to the slightly different rules in the two leagues. The rule that we were most interested in is the American League allows for designated hitters to hit in place of the pitcher whereas the National League does not. This may affect the overall batting average in the league, leading to a potentially higher mean batting average in a particular league.

2.2 Example Data

Table 1. Example statistics for a few players. AL and NL stand for American League and National League, respectively. Note that AB corresponds to at bats, BB corresponds to a ratio of walks to AB, and SO corresponds to a ratio of strikeouts to AB. All of these variables were determined to play an important role in overall BA. Additionally, Prev BA represents the players' BA from the previous season and BA is the players' average for the season listed in column 2.

Player ID	Season	AL	NL	Age	AB	BB	SO	Prev BA	BA
zimmery01	2018	0	1	35	288	.104	.190	.303	.264
yelicch01	2018	0	1	26	574	.118	.235	.282	.326
bradlja02	2017	1	0	27	482	.099	.257	.267	.245
odorro01	2016	1	0	22	605	.031	.223	.261	.271
wilsoda01	1996	1	0	27	491	.065	.179	.278	.285
catalfr01	2005	1	0	31	419	.088	.126	.293	.301

3 Method

3.1 Preprocessing

After inserting the data into an SQL table, the features for each player's season are extracted and the data is preprocessed. The preprocessing stage includes combining a player's stats from multiple "stints". A stint is a player's time with a particular team during a season. For example, a player who stayed on the same team for the whole season will simply have 1 stint, while a player who was traded two times during a season will have 3 stints, one for each team he played on that season. Since we are looking at the season as a whole, we sum these stats under each stint into a single stint. Since we are looking to predict based the beginning of a season (the model does not know that player will be traded during the season), we will use the first stint to determine the league they are in.

To continue with the preprocessing stage, there are many outliers that will throw off the model due to having very few data points (at-bats) for this player. For example, several players have 1 hit in just 1 at bat, which would yield a batting average of 1.000. For such a small number of at-bats, the results are mainly dictated by luck and cannot be predicted accurately using the features given due to the inherent randomness of the game. In other words, it is not useful to ask the model to predict for a season where the player had only a few at-bats because these results will mainly be dictated by luck. Therefore, we will eliminate any training examples where a player had less than 200 at-bats in a season. Importantly, we will still include a player's previous year's statistics regardless of how many at-bats he had, but we will include the number of at-bats in the previous year to allow the model to compensate for statistics generated by a smaller number of at-bats. This will allow the model to determine how valid the previous year's data for a player is and how much it should rely on these predictive features.

3.2 Training

After the data has been preprocessed and the features have been properly extracted, machine learning can be applied to predict new results. The data is collected from the database and shuffled. Then, the data is split into a training set and a validation set with the training set composed of a random 80% of the data and the validation set composed of the remaining 20% of the shuffled data. Then, using the SciKit Learn library, a simple least-squares linear regression model is fit on the training

data. The model then makes predictions on the validation set, which is compared with the ground truth, given by the player's actual batting average for that year. The difference in the prediction and the actual result is computed and a percent error is returned. We then repeat this 100 times- training a new model on a new training set containing different examples from the reshuffled data set and testing the predictions on a new validation set.

There was a consideration to use Empirical Bayes and Bayes Methodologies as found in a study that tried to predict a players performance in the second half of a season based on data from the first half (Brown). Based on the availability of data and scalability of a year to year machine learning model, this research takes a slightly different approach, hoping to provide insights for off-season strategy and predictions.

3.3 Final Training

In the final project, a gradient boosting algorithm can be used, which produces a prediction model in the form of decision trees. This is a similar method to that of a study to predict MLB game outcomes (Pharr). Hyperparameters are found using the hyperopt library. Two to three different gradient boosting models are trained and the predictions are combined to determine our predicted batting average. More features can be used as well including a 3-year average BA and position, which will likely improve the performance of the linear regression model already in place. Next, using weights on each example based on the number of at-bats in the prior year may lead to more accurate results than including it as a feature directly. This will allow examples that have very few at-bats in the previous year's statistics to be disregarded and help reduce the overall error in prediction. Finally, including pitchFx data for each player may allow more insight into a player's real skill. This dataset includes more detailed analysis like exit velocity of a hit ball and attack angle. This will allow any model to more accurately represent a player's skill in determining the overall the prediction of their BA.

4 Results

4.1 Preliminary Linear Regression Results

As a baseline, linear regression was used to predict a player's batting average for a given year. On 100 iterations, the prediction on the validation data set had an average error of 8.83% with a standard deviation of 0.156%. For comparison, in the paper "Forecasting Batting Averages in MLB" (Bailey, 17), the combined model that used linear regression to combine their logistic model with their PECOTA predictions achieved 2.07% average error. This implies the model succeeded in predicting player batting averages to a moderate degree compared with other similar models. The results were mediocre given the limited set of features used and the lack of optimization implemented in this baseline. The standard deviation implies the model was fairly reliable in predicting the batting average based on these features. In fact, the average error was never worse than 9.5%, which shows solid consistency.

Analysis of the trained model shows that the most important feature was the previous year's batting average, which is unsurprising. The next largest weight was the previous year's strikeout percentage with an inverse relation to the predicted batting average. This is intuitive as a player striking out more often means they hit the ball less often, which implies they will likely have a lower batting average. Finally, the age also negatively affected the output, which also makes sense because the older a player gets (after a certain point), the more he declines in skill. This also leads to a decline in batting average. For the final version of this study, it is more likely that we will see more of a curve on this feature, as it is expected for a batter to get better through the first half of their career.

5 Extending This Research

There are many ways in which the above approach and methodology can be extended to incorporate more factors of human performance. For example, MLB.com has more extensive data about each at bat performance. In other research predicting batting averages, this data has been used and was able to refine predictions (Bailey). This data potentially could be used to discover a players' at bat tendencies and use these to predict future batting average more accurately. Further extension of this

research could be using predicted batting averages to try to determine other statistics about the game, such as wins through a season and to use machine learning to refine wins above replacement for batters. Another interesting application of this research would be to apply it to softball, where the game is fundamentally different in the way it is played in that there are less home runs and has a higher emphasis on fielding.

References

Bailey, Sarah. "Forecasting Batting Averages in MLB." *Simon Fraser University,* 2017.

Brown, Lawrence D. "In-Season Prediction of Batting Averages: A Field Test of Empirical Bayes and Bayes Methodologies." *The Annals of Applied Statistics*, vol. 2, no. 1, 2008, pp. 113–152. *JSTOR*, www.jstor.org/stable/30244180.

Lahman, Sean. "Download Lahman's Baseball Database." *SeanLahman.com*, 31 Mar. 2019.

Pharr, Roger D. "Predicting MLB Game Outcomes with Machine Learning." *Medium*, Towards Data Science, 3 Aug. 2019.