Using Predictive Models to Assess Future Value of MLB Players

Connor Morehouse

April 2021

Abstract

There are metrics that are frequently used to determine how valuable a player is for the team in specific areas of the game of baseball. As it stands currently, many of these statistics are all observed samples. Now that we have access to more information afforded to us by ball tracking technologies, we can use the intrinsic characteristics of each batted ball event to build a model of expected values for these metrics that we can then compare to the observed values to the model to determine the natural variance from the mean, or how 'lucky' a player was. We will use the model in an attempt to predict the future value of the player based on the amount of variance of their predicted values from their observed values from their previous seasons.

1 Introduction

In Major League Baseball every year teams are faced with decisions about what players to acquire and what players to let go. The decision greatly depends on the value the team things the player will bring to the team in the coming years, but prediction of a player's value has historically been very difficult.

From the old and simple counting of hits to the new and relatively complex metrics like weighted on base average, there have been many performance metrics that have been used to in an attempt to measure a player's value. We will be looking at the yearly prediction of batting averages, the proportion of hits to total attempts, or at bats. We choose batting average because it has been used almost since the beginning of baseball and is a staple metric when assessing players from past eras while still being pertinent today.

2 The Data

The data we used was sourced from the Statcast database. This is a database that has a record of every pitch thrown in an MLB game since 2015 through 2020. There are 89 variables recorded per pitch and there have been about

4.9 million pitches thrown in that span. This creates a very large file so we immediately wanted to reduce the size of the data set. We removed over half of the columns that weren't pertinent to our project, like whether the game was post season or regular season or what runners were on base, and we also removed all pitches that didn't result in the end of and at bat (81% of total rows). We used three variables for the model: launch angle, exit velocity, and hit direction. We also used other variables for organizational purposes like player name, game year, and player id. Hit direction is not a variable recorded by Statcast therefore we calculated it using the x and y coordinates that are recorded by Statcast. We also removed players with less than 200 at bats in the particular season we were looking at.

3 The Model

3.1 Explaining GAM

We used a General Additive Model (GAM). The General part of the name means that we remove the assumption that most traditional statistical models use about the normal distribution of errors in the data. The Additive part of the name means that the model sums the effect from each of the predictors to generate a single output. Another feature of GAMs that we took advantage of we their ability to find non-linear relationships between predictors and response variables by using smooth functions as opposed to the standard linear regression lines. Another feature of that GAMs provide us with is regression, a method to prevent skew introduced by outliers.

3.2 Alternative Models

The two other types of models that we considered using were standard linear regression and logistic regression. Linear regression calculates a linear trend line that best fits the data and outputs a numeric value. It is not a general model so there is an assumption that the distribution of errors is normal.

A logistic model is a more specific version of a linear regression model. It generates the same trend line that fits the data then uses a logistic function to transform the trend line and output a new trend line that produces a probability for binary classification. This was better suited for our needs, as we are predicting the probability of each batted ball event being a hit.

3.3 Why we choose GAM

Ultimately we choose to use a GAM for a combination of reasons. Firstly, we can use it for the binary classification, so it suits our needs of classification as a hit or an out. Because our data has hard limits to the possible values, for example a ball can't be hit less than zero miles per hour, this caused the distribution of errors to be non-normal, and thus we needed a general model. Our predictors all had non-linear relationships with our output variable, so the

Year	**Accuracy**
2015	69.6%
2016	69.6%
2017	72.9%
2018	74.5%
2019	76.4%
2020	67.1%

Table 1: Model accuracy per year

GAM's ability to use smooth functions combined with regularization allowed for increased model accuracy. In summation, a GAM can do everything that the other models could do, but also had the added benefits of smooth functions and regularization for increased model accuracy.

3.4 Creating the Model

Before we created the model we could simplify some of the outcomes. For a strikeout, the probability of a hit will always be zero because the ball is not put in play. Due to reliability issues at extreme launch angles for the Statcast tracking system, popups and ground balls had unreliable launch angle metrics [4]. The hit probability of a popup is very close to zero, so we assumed all popups to outs. The is a very real chance that a ground ball will result in a hit, therefore we created a separate model that used only exit velocity and hit direction, as opposed to exit velocity, hit direction, and launch angle.

For each player we created a standard model and ground ball model every year. For example, we would use a players 2015 data to train a model, then use that model to predict their 2016 batting average. We choose to have a model for each player because each player has unique batted ball characteristics that we could more accurately account for, and we had enough data for each player to allow for this. We created a new model each year to try and account for changes in a player's batted ball metrics. These changes could be due to things the batter did like new training or a new approach, or they could be caused by changes in the way pitchers approaches the batter. With retraining each year, these differences will be accounted for in no more than one year.

3.5 Checking the Model

To check the accuracy of the model and asses the viability of the model, we decided to define an accurate prediction to be within .020 of the observed value. The accuracy values for each year are shown in table 1. The values are significantly above 50% without being unreasonably high (90%+), this was a

4 Difference

Our predictions will rarely if ever be exactly the same as the observed values. This is in part due to the random variance of that data that is natural when collecting data in the real world. We are confident that our model accounts for most of the variation that intrinsic to the data, therefore the difference we see between our predicted values and the observed values is mostly due to random variance, or 'luck.' Because of this the fluctuations of the data away from the predicted values will unsustainable over a large number of trials and we expect to see the observed values regress back to the predicted model in the next year. This will allow us to make informed decisions about the offensive performance of a player in the next year.

5 Example analysis

5.1 Positive Example

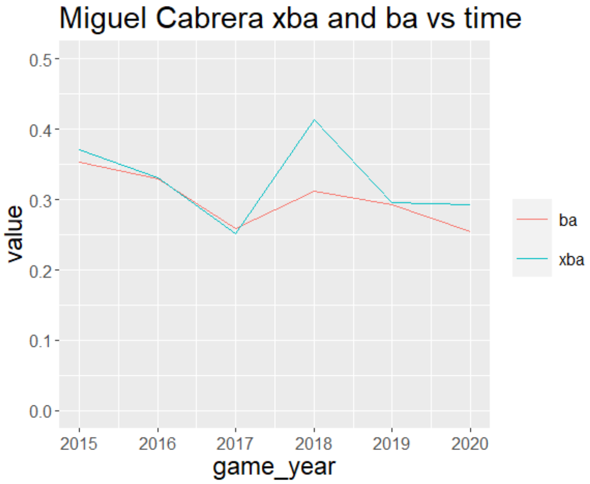

Figure 1: Miguel Cabrera batting average and predicted batting average over time

We see in Figure 1 the expected batting average for Miguel Cabrera is higher than what was observed in 2020. Like we stated earlier, because the difference between the two values is assumed to be due to luck and because luck is unsustainable over a large number of trials we expect the observed value to regress back to the predicted value. Therefore, we expect Miguel Cabrera's batting average to increase in the 2021 season. This is valuable to the team because it allows the team to make an informed decision regarding the future offensive value of the player. For example, if the player's contract was up in 2020, they could offer the player a contract based on their observed batting average while expecting the player to more valuable in the next season, hence they saved money and got more than they payed for.

5.2 Negative Example

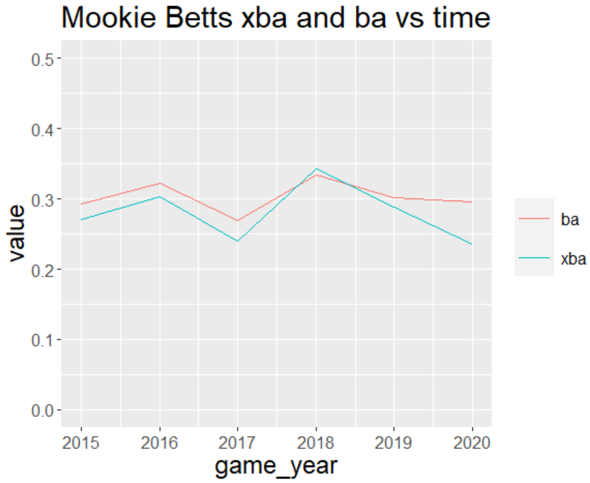

Figure 2: Mookie Betts batting average and predicted batting average over time

This is a similar example. We see in figure 2, Mookie Bett's expected batting average is lower than his observed batting average for 2020, so we expect to see his batting average decrease in 2021. This could be valuable to the team in the case of a contract extension because the team could propose an offer based on the predicted value, again to try and pay less for a player and save money. This might be a more difficult strategy to use considering the accuracy of the model especially in comparison to a concrete observed value.

5.3 Good Case for Prediction

As a result of the predictors we used and the different characteristics each player leverages to be successful, our model works much better for some player archetypes than others. In figure 3 we look at Joey Gallo, who is one of the best examples of a home run hitter. The keys to being a successful home run hitter is to hit the ball high and hard, two variables we account for. Therefore, we see that our model performs very well on for Joey Gallo having a maximum difference of only .013 for any season.

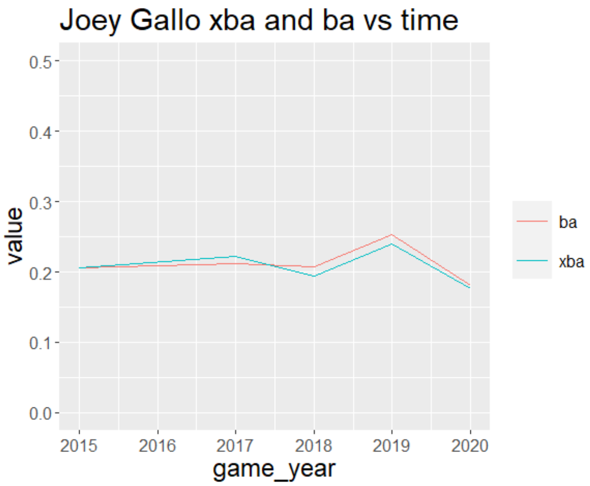

Figure 3: batting average and predicted batting average over time

5.4 Poor Case for Prediction

In contrast to the Joey Gallo example, figure 4 shows Billy Hamilton's observed and predicted batting averages. Billy Hamilton is one of the MLB's fastest players, and he uses his speed to create hits out of batted balls that would otherwise be outs for many other players. Because Billy Hamilton relies on his speed, something we didn't account for in the model, and has below average values for the predictors we used in the model, we consistently under predict his batting average almost every year.

6 Discussion

6.1 2017 Ball Change

In 2017 there was suspicion of a change in the ball used in major league baseball [2]. The suspicion was that the ball was made harder and the seams lowered which lead to an increase in exit velocities and flight distance. The theory is this was done to increase total offense in the league. The effect we observed on our model is consistently inflated predictions in the year 2017, because the 2017 model was trained on 2016 data that couldn't account for the change in the ball. This was corrected for in the following year and we didn't observe any similarly consistent errors.

6.2 Future Work

There are variables that intuitively could have significant impact on the probability of a hit that we weren't able to account for in our model. Some examples

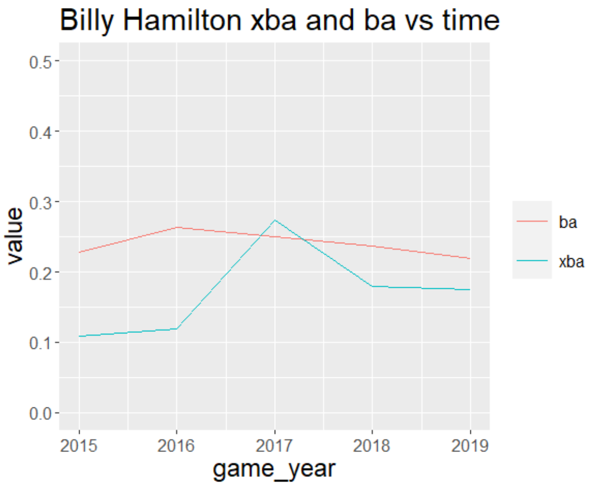

Figure 4: Billy Hamilton batting average and predicted batting average over time

are player speed and infield alignment (when the defense changes from standard positioning to one biased towards where the hitter is likely to hit the ball, in hopes of preventing a hit). Because we didn't have the data available to us, this could be reserved for future works.

Another potential for future work would be to model another statistic that was indicative of offensive value, like weighted on base average (wOBA) or weighted runs created (wrc). These would require a non-binary model because they take into account the quality of the hit as opposed to our model that only determined if the event resulted in a hit or not.

References

[1] Triple-a hrs surge after switch to big league ball.

[2] Ben Lindbergh. The juiced ball is back.

[3] Jason Loeppky Sarah R. Bailey and Tim B. Swartz. The prediction of batting averages in major league baseball.

[4] TangoTiger. Statcast lab: No nulls in batted balls launch parameters.

A Data Source

Our data was sourced from the Statcast data base available at `https://baseballsavant.mlb.com/statcast_search`

B R Code

See included files