xSLG - Predicting Future Slugging in Baseball

Chase Diaz

Problem

The Major League Baseball regular season spans 162 games, with the typical starting player averaging around 550 at-bats. Over the course of a season, hitters experience both successful and challenging stretches – baseball contains a significant degree of randomness and luck. Even when hitters make perfect contact, a well-struck line drive off the barrel of the bat can still result in an out. Conversely, balls hit with weak contact, such as a swinging bunt, or jammed pop-fly landing behind 1st base, often leads to base hits. Yet, players will be statistically awarded for weak contact resulting in base hits and penalized for strong contact resulting in an out.

This is the issue with many statistics in baseball – they fail to accurately reflect a player's true ability. At any given point in the season, how can we estimate a player's "true" ability, removing luck and randomness? How can we assess which players have underperformed and overperformed thus far? Which players are likely to improve in the second half of the season? This project aims to address these questions through statistical modeling, with the goal of estimating a player's true ability in terms of slugging.

Overview

In baseball, slugging (SLG) is a key indicator of a hitter's productivity, representing the average number of bases earned per at-bat. SLG is calculated by summing the total number of bases a player has earned through hits and dividing that sum by the player's total number of at-bats. For example, consider a batter with 10 at-bats in a season, registering 3 hits - 2 singles and a triple. In this scenario, his SLG would be .500 since he recorded 5 total bases over his 10 at bats.

$$SLG = \frac{Total\ Bases}{Total\ At\ Bats}$$

The goal of this project is to estimate a player's "true" slugging, by developing an expected slugging (xSLG) model using batted ball data and then using the models results to predict 2nd half SLG.

Approach

My approach for making slugging (SLG) predictions for the 2nd half (2H) of the hitter-seasons consists of two parts:

1. **Develop an Expected Slugging (xSLG) metric using 1st-half (1H) batted ball data**
   We aim to estimate "true" 1H slugging based on the quality of each batted ball and characteristics of the pitch. Ensemble models, specifically XGBoost and Random Forest, have been selected for their ability to handle complex relationships within the data and to

provide a deeper understanding of the factors influencing slugging performance. The xSLG metric derived from these models serves to mitigate the impact of noise and randomness in a player's actual SLG. To ensure the reliability of our xSLG metric, we will assess the performance of each ensemble model against unseen testing data, and selecting the model with the lowest Root Mean Squared Error (RMSE) for implementation into the stacked model to predict 2H SLG. The xSLG metric is then calculated for each player-season, summarizing their batted ball statistics over the season.

2. **Incorporate the xSLG metric into a stacked model to predict 2nd-half (2H) SLG** Utilizing the insights and predictions from the chosen xSLG model, we will integrate xSLG with the most significant features for predicting SLG into a linear regression model to forecast 2H SLG.

Root Mean Squared Error (RMSE) was chosen as the primary accuracy metric for model evaluation. Using RMSE will penalize predicted values further from their true values by squaring the errors, and then taking the square root of the average of these squared errors. This process adds weight to larger errors, emphasizing the importance of minimizing error in evaluating model performance. The stacked model predictions will be benchmarked against a baseline assumption, comparing the predicted 2H SLG to the assumption that a player's performance will remain consistent with their 1H SLG. This enables us to gauge the effectiveness of our model relative to a baseline.

Additionally, we will assess the stacked model's performance in predicting whether a player's SLG will increase or decrease in the 2H of the season. These results will be compared to solely using the xSLG metric to predict 2H SLG, demonstrating the enhanced effectiveness of combining xSLG with significant features compared to relying on xSLG alone.

Data

The training data (first_half_batted_balls.csv) used for this project is batted ball data from the first half of a season. Each data point in the training data is the result of a ball hit in-play from a single at-bat containing 60,294 rows and 33 features. The testing data (second_half_slg.csv) contains the 2nd-half slugging for each hitter-season. The features are described below.

Column	Description	Type
month	An integer corresponding to the month in which the given play took place. 4 corresponds to April, 5 to May, and so on.	INT
season	An integer corresponding to the year in which the given play took place.	INT
game_id	A unique character identifier signifying a distinct game.	STR
inning	An integer corresponding to the inning in which the given play took place.	INT
top	An integer corresponding to whether the play took place in the top or bottom of the inning. If this column	INT

	is 1, then it took place in the top of the inning, if this column is 0, then it took place in the bottom.	
pa_of_inning	An integer corresponding to the PA number of the half-inning in which the given play took place.	INT
batter_id	A unique character identifier signifying a distinct batter.	STR
bat_side	A character signifying the handedness of the batter for a given plate appearance. "R" means the batter hit from the right side, "L" means the left.	STR
pitcher_id	A unique character identifying signifying a distinct pitcher.	STR
pitch_side	A character signifying the handedness of the pitcher for a given plate appearance. "R" means the pitcher user their right hand to throw the ball, "L" means they used their left.	STR
pre_balls	An integer reflecting the number of balls in the count before the pitch.	INT
pre_strikes	An integer reflecting the number of strikes in the count before the pitch.	INT
pre_outs	An integer reflecting the number of outs in the half-inning before the pitch.	INT
pitch_type	A character indicating what type of pitch was thrown by the pitcher, as automatically tagged by Trackman's pitch classification system.	STR
pitch_spin_rate	A float indicating the rate at which a pitch was spinning when thrown, given in rotations per minute.	FLT
pitch_release_speed	A float indicating the speed of the pitch as measured by Trackman, given in miles per hour.	FLT
pitch_tilt	A time value indicating the axis of rotation of a pitch in the x-z plane, given in hours and minutes. For more information on working with this value, see this resource here: https://www.drivelinebaseball.com/2019/09/mastering-the-axis-of-rotation-a-thorough-review-of-spin-axis-in-three-dimensions/	TIME
pitch_release_height	A float representing the vertical coordinate of the pitcher's throwing hand at the time the pitcher releases the pitch, given in feet, measured from the ground up.	FLT
pitch_release_side	A float representing the horizontal coordinate of the pitcher's throwing hand at the time the pitcher releases the pitch, given in feet, measured from the imaginary line running through second base and home plate.	FLT
pitch_release_extension	A float representing the distance between a pitcher's throwing hand and the pitching rubber at the time the pitcher releases the pitch, given in feet.	FLT

pitch_vert_break	A float representing how much a pitch deviated vertically from its straight line trajectory from the time of release to the time it crosses home plate, given in inches.	FLT
pitch_induced_vert_break	A float representing how much a pitch deviated vertically from its straight line trajectory from the time of release to the time it crosses home plate, removing the effects of gravity, given in inches.	FLT
pitch_horz_break	A float representing how much a pitch deviated horizontally from its straight line trajectory from the time of release to the time it crosses home plate, given in inches.	FLT
pitch_plate_height	A float representing the vertical coordinate of the location of the pitch as it crosses the front of home plate, given in feet, measured from the ground up.	FLT
pitch_plate_side	A float representing the horizontal coordinate of the location of the pitch as it crosses the front of home plate, in feet, measured from the imaginary line running through second base and home plate.	FLT
hit_exit_speed	A float representing the speed at which a ball left the bat of a hitter after contact, given in miles per hour.	FLT
hit_vert_exit_angle	A float representing the vertical angle at which a ball left the bat of a hitter after contact, given in degrees.	FLT
hit_horz_exit_angle	A float representing the horizontal angle at which a ball left the bat of a hitter after contact, given in degrees.	FLT
hit_spin_rate	A float representing the rate at which a batted ball was spinning after leaving the bat of a hitter after contact, given in rotations per minute.	FLT
hit_distance	A float representing the distance a batted ball traveled after leaving the bat of a hitter after contact, given in feet.	FLT
event_type	A character signifying the outcome of the plate appearance.	STR
slg	A float representing the SLG value of the outcome of a plate appearance, corresponding to the number of bases the batter recorded on a play.	FLT
second_half_wobacon	A float representing the average SLG of a hitter on batted balls, recorded for a particular season in the months of July, August, September, and October.	FLT

Data Exploration & Pre-Processing

The first step towards generating 2H SLG predictions began with data exploration, and understanding what pre-processing would be required for modeling. Looking through the data, there were several columns with missing values that would need to be imputed, **'hit_spin_rate'**, was the column with most missing values, missing a third of data. Feature distribution and summary statistics were analyzed before and after imputation to ensure the imputation process was implemented correctly. Pre-processing steps included removing batted ball events that will not be considered at-bats, such as sacrifice bunts and sacrifice fly balls. These events are not included in the equation for slugging percentage, and the data in these rows will not be used for modeling, so they were removed. The equation for SLG% = (Total SLG / Total # AB's). Missing values were imputed using KNN Imputer, leveraging the algorithm's ability to estimate missing values based on the similarity of instances across multiple features. This imputation process enhances the completeness of the dataset, ensuring the models will be trained on a more comprehensive and representative dataset. The data exploration and pre-processing steps described above are shown in the attached Python notebook **'01-EDA-&-PreProcessing.ipynb'**.

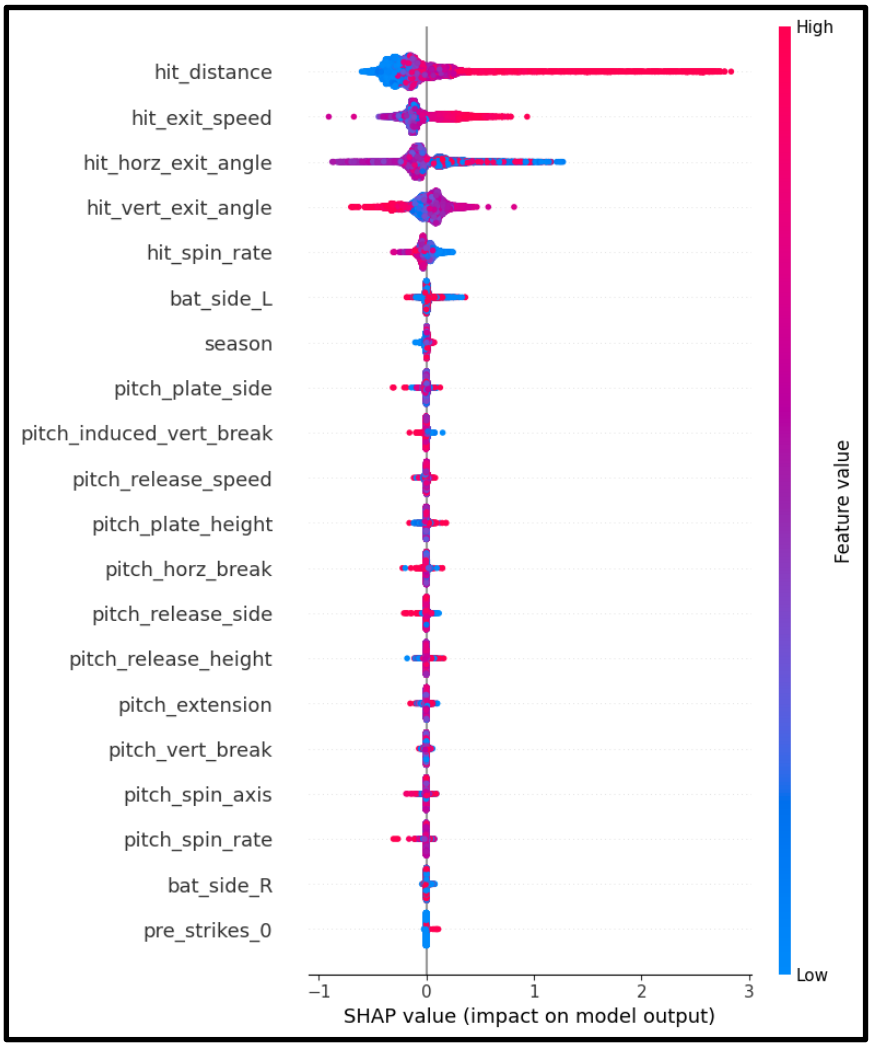
xSLG Modeling: XGBoost and Random Forest

The ensemble models chosen to estimate xSLG were Extreme Gradient Boosting (XGBoost) Regression and Random Forest Regression. The XGBoost model is a boosting model, improving predictive accuracy by iteratively training on the residuals (error) of previous models. Random Forest is a bagging model, building multiple independent models and aggregating their predictions to yield consistent outcomes. I felt these regression models were well-suited for this type of problem because while SLG is a discrete value (0, 1, 2, 3 or 4), we want to predict a continuous value that estimates the SLG quality of a batted ball. A continuous value of 1.5 conveys more information about the batted ball than a value of 1. Specifically, it tells us there is a high probability that the batted ball is a single, but there is a reasonable chance it could be a double, or potentially more bases based on the quality of the batted ball. This kind of projection allows us to filter out noise that may impact the actual SLG, and project a "true" SLG. Additionally, both models are examples of ensemble learners, combining the strengths of multiple weak learners to improve overall performance. After analyzing model performance with default parameters, the hyperparameters of both models were tuned using GridSearchCV to maximize their performance.

xSLG Model	R-Squared	RMSE
XGBoost	.647	.5927
Random Forest	.622	.6013

After training both models on the 1H data, and estimating the xSLG for each batter-season, we can interpret them to identify the most influential features and calculate their R-squared. R-squared is a measure that quantifies how much variance the models account for in predicting SLG. XGBoost returned an R-squared of .647, slightly larger than Random Forest at .622. This indicates 64.7% of the variability in SLG is explained by the independent variables in the model.

This is considered a moderately good fit and tells us the model captures a substantial portion of the variance in the data. However, it also implies that there is still ~35% of the variability that is not accounted for by the model, leaving room for improvement or the possibility that other factors not included in the model influence SLG. **Since the XGBoost xSLG model performed best against the unseen testing data, recording less error, and displaying higher predictive ability, we will use it to generate the xSLG metric incorporated into the stacked model.**



We can gain insights into the key features driving our model's predictions using Shapley values. Shapley values provide a way to allocate the contribution of each feature to the model's output. The Shapley values show us that **hit_distance** and **hit_exit_speed** are the most influential features for predicting SLG, while **hit_horz_exit_angle** and **hit_vert_exit_angle** are also important features. This makes sense, because the more frequently a player can hit the ball further and harder, the higher likelihood for extra base hits, which increases slugging. And the angle at which it comes off the bat also plays a part, balls hit to a players pull side generally will be hit harder, than balls hit to the opposite field. And balls hit into the air will yield more extra base hits than balls hit into the ground.

Stacked Model: xSLG + Significant Features ~ 2H SLG

To train the linear regression model for predicting 2H SLG, we will first run the XGBoost xSLG model on each batter-season to summarize each player's batted ball statistics over the first-half of the season. Then, since we have identified the most significant features for predicting SLG are hit distance, exit speed, and the vertical and horizontal launch angles, we will take the average of these features for each batter and use them as the other inputs into the linear regression model.

Results & Evaluation

Below is a table comparing the results from the stacked model, using the XGBoost xSLG alone, and the 1H SLG baseline against the actual 2H SLG test values. We see that the stacked model performed best in terms of RMSE and whether 2H SLG will increase or decrease, correctly predicting 67.9% of the 2H batter-seasons. Remarkably, using xSLG alone also performed well, with greater accuracy than the baseline assumption that each player will achieve the same SLG in the 2H as they did the 1H of the season. Ultimately, the stacked model resulted in 18.7% greater accuracy than baseline, which is remarkable and shows that the model holds value in predicting future SLG.

Model	RMSE	2H SLG Increase/Decrease
Stacked Model	.0982	67.9%
xSLG - XGBoost	.1117	59.6%
Baseline (1H SLG)	.1208	-

In conclusion, we found that hit_distance, hit_exit_speed and the launch angles were the most significant features in predicting slugging. This makes sense, a player that consistently hits the ball further and with a greater exit speed is making strong contact at a high rate, which translates to more extra-base hits. A use case for this model is assessing which player will make a greater impact for a team in an acquisition at the trade deadline. Since we achieved ~68% accuracy in predicting whether a player's SLG will increase or decrease in the 2H, we can use the model to assess which players may perform better in the 2H of the season.