Project 1: Predicting MLB Hit Outcomes

A Kaggle Inspired Project

By
Eddie Dew

Sep 13, 2024

Introduction

Major League Baseball (MLB) has been at the forefront of incorporating analytics into the game to help teams strategize and gain an advantage over their opponents. It wasn't always like this however. Teams used to focus on simple statistics such as batting averages to indicate the success of a hitter, or earned run average and wins for a pitcher. While these stats encompass some context of a player's ability to perform in the game, they are also dependent on extraneous factors outside of the player's control. For instance, pitchers are responsible for runners on base, but what if a fielder commits an error or an outfielder makes a bad read on the ball which causes a runner to score? Those shouldn't be reflective of the pitcher if you want to judge their individual performance. People started to realize this and in the late 1990s to early 2000s, the Oakland Athletics decided to focus on stats more reflective of a player's individual performance. This was a major success and the Athletics made the playoffs as a result despite not having big name players. Today, every MLB team has incorporated analytics into their ballclub. One evolution that's been really popular today is statcast, MLB's tracking system that uses high speed cameras to track stats such as pitch velocity, break, exit velocity, etc. Wanting the fans to get involved, most of the data tracked is public for users to analyze and potentially revolutionize the way we interpret the game. That brings us to today, where statcast recently provided swing data to the public and a baseball analyst, Nick Wan, decided to host a competition where people are incentivized to make a model to predict the likelihood of a batted ball outcome using this new data.

The data he provided is linked in this Kaggle Dataset. It consists of a train, test, and example submission csv files. Figure 1 below shows the shape of the train and test datasets.

	Rows	Columns
Train	50,000	41
Test	14,220	39

Figure 1: Shapes of the Datasets

The idea with this is to train and test your model on the train dataset then make predictions on the test dataset. You submit your predictions from the test data and those predictions are scored based on Area Under the Curve, to gauge how well your predictions lean toward a correct prediction.

Exploratory Data Analysis

We now have our data loaded, but what are we trying to predict? I mentioned above that we are trying to predict hit outcomes but didn't define what they were. The hit outcomes we are trying to predict are outs, singles, doubles, triples, and home runs. Figure 2 shows the distribution of each outcome.

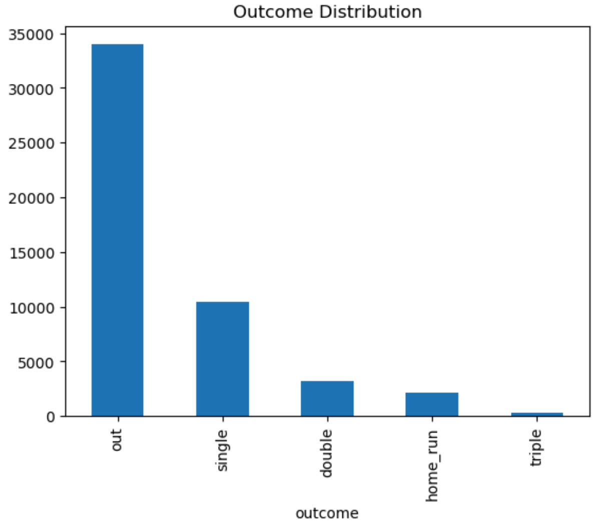


Figure 2: Hit Outcome Distribution from the dataset

This is an imbalanced distribution with outs taking up 68% of the data set. I also checked for NAs, and this dataset had none. Next, we need to find what features to use to make predictions. Since this project wasn't model based, I decided to have a little fun and throw as many features in as possible and try to make as many as I could think of (this was also the point of the Kaggle competition). Since there were a lot of features (28) I will highlight a few that required some searching to derive.

I first looked at the new swing data that was incorporated to see what they had in relation to the hit outcome. Figure 3 shows the relationship between swing length and the bat speed.

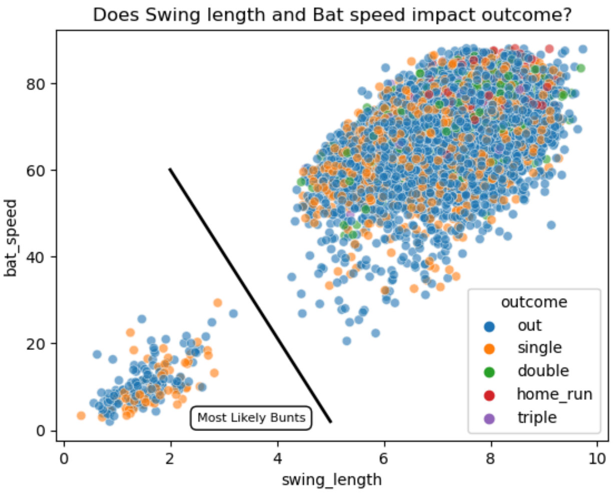


Figure 3: Bat Speed-Swing Length impact on Hit Outcomes

There appears to be 2 clusters in this scatterplot. I suspect that these would be bunts given a bunt in baseball is not going to have a long swing length nor bat speed. I decided to try to see if categorizing a bunt would help the model but it unfortunately was not predictive, so the feature was discarded.

There were a few features I derived using the swing data to see if they were good predictors. One was the Vertical Approach Angle (VAA). VAA is the angle a pitch takes as it reaches the plate. It was derived from a Fangraphs article written by Alex Chamberlain who interviewed a physicist to help him compile the metric using statcast data. I used this computation as an intermediary to try to estimate the swing plane of the hitter. My thought was if I could figure out how level the hitter's swing was, it could help determine if the hit was a line drive, ground ball, or pop fly which would help the model with predicting the outcome. Due to limited time and lack of public sources on this statistic, I tried to logically come up with a formula to derive a swing plane. I decided to break it down into two parts, the horizontal and vertical plane and take the magnitude between the two. It looks like the following formula:

$$horizontal\ component\ =\ swing\ length\ *\ cos(spray\ angle)$$

$$vertical\ component\ =\ launch\ angle\ -\ VAA$$

$$swing\ plane\ =\ \sqrt{(horizontal\ component)^2\ +\ (vertical\ component)^2}$$

It's not the best method for estimating this metric, especially since the distribution is heavily around 50 degrees which is not realistic since the swing plane is going to have more variation. A better way would've been taking the angle of the bat before and after the swing, but I unfortunately don't have that data.

A couple of other features I derived were exit velocity and launch angle since I knew these features were really good context indicators of a batted ball result. I derived exit velocity using a public formula from Beyond the Boxscore which looks something like this:

$$exit\ velocity\ =\ (bat\ effect\ *\ effective\ speed\ of\ pitch)\ +\ (1\ +\ bat\ effect)\ *\ bat\ speed$$

* Note: Bat effect was estimated as 0.2 for wood bats

I also artificially estimated launch angle by taking the tangent of the effective pitch speed divided by exit velocity. These are not the most accurate measures but should still do the job of helping the model make predictions. If we take a look at these values on a scatter plot in Figure 4, we see that they are mostly balls hit in the air (launch angle > 35). I am a little skeptical that the majority of the data were fly balls but I don't have a way of confirming or denying it. One positive is that the home runs were around the 25-45 degree range with exit velocities ranging from 80-120, which is realistic.

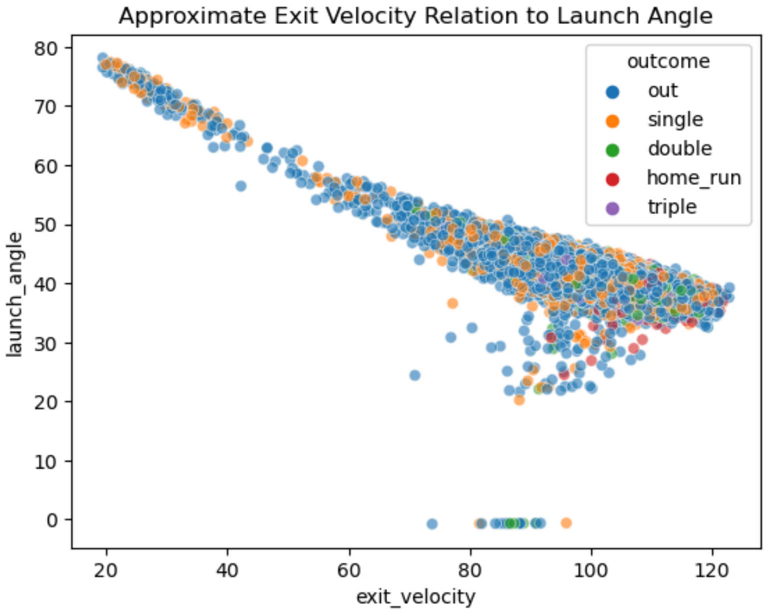
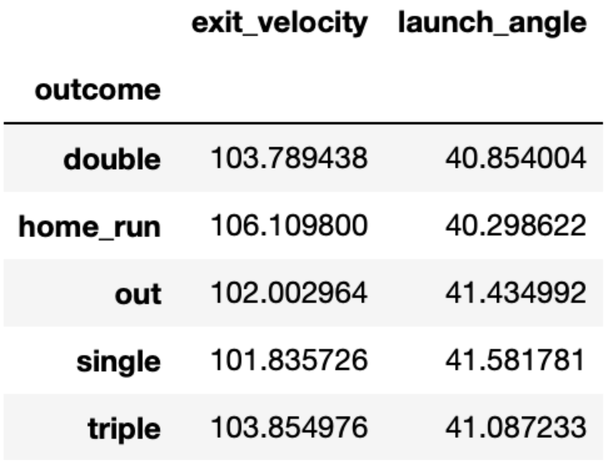


Figure 4: Exit Velocity and Launch Angle impact on Hit Outcomes

Figure 5 shows the median values for exit velocity and launch angle.



	exit_velocity	launch_angle
outcome		
double	103.789438	40.854004
home_run	106.109800	40.298622
out	102.002964	41.434992
single	101.835726	41.581781
triple	103.854976	41.087233

Figure 5: Median Exit Velocity and Launch Angle estimates

A positive takeaway from this are home runs were typically hit the hardest with a lower launch angle compared to the other outcomes, very realistic.

Model

The model I decided to use was eXtreme Gradient Boosting (XGBoost) because it excels at making classifications using decision trees, which are fine-tuned through its parameterization. The objective was tuned for multiple classification. After the parameters were tuned, I ran it through 10 folds of the data so the model would see different groupings of the dataset. As a result, the model performed with a best 0.732 Area Under the Curve (AUC). Awesome! After a rush of confidence, I turned in the predictions dataset into Kaggle and scored a 0.688 AUC, yikes. That put me outside of the top 20 on the leaderboard. It also means my model overfit to the data being fed to it, which was a result of desperation for a higher score and not so much predictiveness for the model.

Impact

While my model was unsuccessful at predicting hit outcomes compared to other models, there are some takeaways that can be helpful for someone who would like to improve this model. I found that hits that were pulled based on the batter handedness were pretty indicative of the outcome. The direction (spray angle) of the ball and swing plane were also helpful features. Making sure metrics like exit velocity and launch angle are correct will undoubtedly improve the model, so adding in those features into the dataset instead of deriving them would help. Maybe adding in other features such as how fast the runner is could help improve the model. Another thing worth trying is using another model. Over the course of this project, I learned that Catboost may be better at predicting categorical variables compared to XGBoost.

This was a fun project, but I wouldn’t classify it as a reliable one when applying to real world situations. I was able to gain more insight on different models and tuning methodology but

I want to emphasize that your model is only as good as the features you put into it. This project is a good example of that.

Works Cited:

Chamberlain, Alex. “A Visualized Primer on Vertical Approach Angle (VAA).” *FanGraphs Baseball*, 26 Dec. 2022, blogs.fangraphs.com/a-visualized-primer-on-vertical-approach-angle-vaa/.

Cole, Bryan. “What Does a 100mph Swing Speed Mean?” *Beyond the Box Score*, Beyond the Box Score, 11 Feb. 2015, www.beyondtheboxscore.com/2015/2/11/8010803/zepp-swing-sensor-perfect-game-100-mph-swing.

Wan, Nick. *Predict Hits with NEW MLB Swing Data*. Kaggle, 2024, https://kaggle.com/competitions/nwds-batted-balls.