# Out “Hitting” the Odds

**Jack Arbuckle, Lauren Beaudreau, and Ben Scartz**

## Introduction

This project looks to analyze the probability of an MLB player getting a hit during a game. This idea stems from common hitter prop bets that have become very popular in online sports betting. Each day the odds of a person getting hit are released which usually fall between -400 and +200. Although most starters have a favored chance of getting a hit, sometimes this bet is parlayed together with a couple of players to create + odds. People look to capitalize on something that looks very enticing. All they need is one hit in about 5 to 6 at bats for the whole game. But as with most betting, the success rate usually falls with the sportsbook. Some of this betting failure comes from the nature of the sport. A better would think that a person hitting 0.300 would get a hit 3/10 at bats, so getting 1 hit per day seems very favorable. But unfortunately, baseball does not work like that. Baseball is a sport of average. Players hit 0.300 over the season, not 0.300 daily. Instead of going 1.5/5 each day, they might go ⅗ the first day and then 0/5 the next day. These drastic changes have to do with hitting streaks, pitching matchups, game situations etc. All these different factors make the hitter prop bets more of a tossup then they appear. Without reliable information it is hard to predict when these bunch hits or some hits or no hits will occur. Even the most productive hitters can go 0/5 any given day. The goal for this project is to create a model that takes hitter and pitcher metrics and situations into account to better predict the probability of that player getting hit for that certain day. The probability generated by the model for one or multiple players combined can then be used to make an educated bet that is more in the favor of the bettor than the sportsbook. The goal is to gain meaning behind the odds in order to bet smart, not bet on blind faith. The output of the model will be the probability of the player receiving a hit on that certain day. The input will rely on different metrics such as expected Batting average and percent of hard hit balls against certain types of pitches.

## Related work

There is some previous research on this topic, especially since hitting prop bets and games like “Beat the Streak” [1]. Some of the data is a very basic interpretation of using batting average and other “old school” odds to build not predictions, but analysis. For these more basic insights there are many websites that have information on hitting probability and different line graphs for their average over time. [2] While this is a good analysis for other things like long term success and predictions, it fails to provide real usable value for betting on daily odds. On the other hand, there is some more advanced research on predicting hits for MLB players. Numerous github pages had basic to advanced machine learning models that looked to analyze player hits on a daily and weekly basis. An example of one of the more advanced methods was a group [3] that used batter metrics, pitching metrics, stadium data, and weather data in order to try to predict the chances of a player getting a hit. They were inspired by the “Beat the Streak” game and wanted to see how long they could predict different players to get a hit on consecutive days. They also attempted with individual players on a given day but had mixed results due to limited machine learning models.  At heart their end goal was to predict the odds of consecutively getting hits correct while this project's focus is on daily prop betting, not consecutive consistency.

## Data Description

The dataset that will be used to build our model is the 2023 Statcast data. This dataset shows every pitch from each game throughout the season. Each row represents an individual pitch with 99 variables attached to that pitch. Pitch type, game date, pitcher ID, batter ID, and the event that occurred from each pitch are some of the variables of most interest to us. Statcast_2023 has 729,774 observations.

From the original dataset, statistics will be calculated, forming new tables grouped by batters, pitchers, and dates. First, the batter_stats table displays the statistics described in detail above, namely xBA, 90+%, and ZO-swings, for a given date. Pitcher_stats displays xBA, 90+%, and pitch usage by date. Finally, bullpen_data displays xBA for all relievers by team.

In creating a complete data set for testing, all statistics to date will be joined to the corresponding game results: did the hitter get a hit on the particular date, either against the starting pitcher or bullpen, whose statistics are described in the row.

## Methods

This project focuses on building an xgboost model to predict whether a player will get a hit on a certain day. It will use various metrics as explanatory variables for the response variable of hit or no hit. The first metric used was the expected batting average metric. Expected Batting Average is a statcast metric that measures the probability that a batted ball will become a hit. Every batted ball is assigned an xBA on comparable hit balls in terms of exit velocity, launch angle, and hit location. In addition to xBA metric, the percentage a player hits balls with an exit velocity of 90 mph or above on certain types of pitches was chosen as an explanatory variable, as this is a better indication of solid batting and overall batting performance over time. By looking at exit velocities of 90 mph or above, the model can examine the data on a pitch by pitch level rather than just overall plate appearances. The last main metric of interest is Batter Z-O Swings, with Z-Swings being pitches inside of the strike zone that the batter swings at while O-Swings are pitches outside of the strike zone that the batter swings at. The Z-O Swing metric is calculated by subtracting the O Swings from the Z swings, therefore a higher value is better. The Z-O Swing Percentage is a better indicator of plate discipline. The predictive model also includes pitcher data such as xBA against, and types of pitches thrown. These two results will then be combined and put into a predictive model.

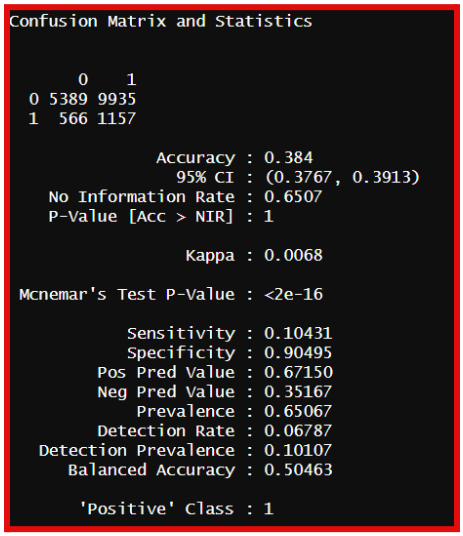
To build the predictive model that can be used for betting, every MLB player will be listed and a percentage of hits based on each pitch type that exists. For there to be a percentage listed, the batter must have 10 swings per pitch type and 25 overall batting appearances. This is to eliminate the players with a small sample size that may skew the data. Since the project goal is to be able to predict how many hits an MLB player will get during a given game, it is necessary to include all three of these metrics in the predictive model. In simple terms, the model will predict how many hits batter X will get when facing pitcher Y who tends to throw certain types of pitches.

The xgboost model chosen for predicting MLB player hits offers several strengths that make it a suitable approach. Its flexibility allows for capturing complex interactions between hitter and pitcher metrics, enabling the model to effectively assess the likelihood of a player getting a hit. Additionally, the model's ability to determine feature importance provides valuable insights into the key metrics influencing hit probabilities, guiding decision-making and strategy

development. Its scalability ensures efficient processing of large datasets, such as the extensive 2023 Statcast data used in this project. However, there are limitations to consider. The model's effectiveness depends heavily on the quality of input data and assumptions of stationarity may not always hold true in dynamic sports environments such as that of baseball. Additionally, the model's performance may be impacted by sample size variations among players. This is why we set a minimum plate appearance threshold but variability still remains. Despite these limitations, the xgboost model represents a powerful tool for informing betting strategies and decision-making in sports betting contexts.

## Results

Once the data was cleaned and put into the proper table, the xgboost model was run. The way this model's accuracy was measured was different due to our choice to try and evaluate our model based on betting methods. Instead of looking at overall accuracy we just look at only players we wanted to bet on. Therefore we set the cutoff for our confusion matrix and mode at 0.80. This meant that only players that had a probability of 80% or higher were chosen to be bet on. This is a more real world applicable situation because when a person bets they do not bet on 30+ players but rather around 3-5 players. Therefore to better try to analyze what players a person would actually bet on using the results of the model the cutoff was set very high. Below is the confusion matrix that analyzes the accuracy of the model. Due to our cutoff of 0.80 the overall accuracy of the model appears to be very low at only 38%. But for the analysis of this model this was irrelevant. The key metric that was focused on was the positive predictive value which was 67%.



```
Confusion Matrix and Statistics


     0    1
 0 5389 9935
 1  566 1157

               Accuracy : 0.384
                 95% CI : (0.3767, 0.3913)
    No Information Rate : 0.6507
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0068

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.10431
            Specificity : 0.90495
         Pos Pred Value : 0.67150
         Neg Pred Value : 0.35167
             Prevalence : 0.65067
         Detection Rate : 0.06787
   Detection Prevalence : 0.10107
      Balanced Accuracy : 0.50463

       'Positive' Class : 1
```

This is the accuracy of the players that were predicted at 0.80 or above that actually did get a hit. These are the players a person would be putting money on so this accuracy is what was important to boost. Therefore also the matrix is not trying to show that 9,935 got a hit but were predicted not to get a hit, but that 9,9935 had a hit probability of less than 0.80 so it would be unwise to bet on these players. With this problem being with a betting focus, we could limit the sample we chose to bet on. The model did not need to focus on all players.

Using this model with the 0.80 cutoff, the goal was to determine how these findings could be used to make accurate bets. To do this the results were plugged into a created function to pull out real world results over a season. The results that it showed were the overall accuracy of the bets, the number of bets made during the season, and the overall odds that would need to be averaged in order to turn a profit.

```
"Bet hit percentage: 80.3%"
"You would beat average odds of -408"
"Placing 61 bets over 173 days (average of 0.4 per day)."
```

This shows that making about 1 bet every other day (Betting on players that only have a 0.80 or higher probability) would need to average the odds of -408 or better in order to turn a profit. This seems very reasonable as most odds are more favorable than -408 even for superstar players. As mentioned, hitting odds usually vary from -300 - +200. Also betting 61 times per season is on the smaller side of average betting that a lot of individuals do.
This does produce a challenge as sometimes there are days when no player has a hit prob of 0.80 or above. This creates boredom for a bettor when trying to do the best bet. This is one of the disadvantages of the model. It is relatively accurate but also relatively cautious.

Finally the project looked at parlay accuracy. As mentioned earlier as well, betting on numerous players at once to get a hit (a parlay) is a common betting practice. Similarly, the model results were plugged into a created function. This function showed the overall accuracy and the overall odds that would be needed to average in order to turn a profit.

```
"Parlay hit percentage: 27.7%"
"You would beat average odds of +260"
```

With a cutoff of a 3 part parlay the above output shows that in order to return a profit the would need to average the odds of +260 or better. Although with an accuracy of only 27.7%. This shows the difficulty and sportsbook advantage of parlays.

## Actions

The logical progression of this project will be to prepare a daily program that can present modeling results based on each day's events. The BaseballR package includes functions to scrape the day's starting lineups (get_batting_orders()) and starting pitchers (get_probables_mlb()) for all MLB games on a particular day. Also, daily odds can be collected using the free API, The-Odds-API. While odds could not have been collected over the span of the entire train and test data sets, they can be collected daily for future application. The odds, lineups, and data described above can all be combined to produce a single table which would provide guidance on the most-highly-recommended bets for each day.

This would allow performance to be tracked on a more practical basis, using the odds to track dollar wins and losses for all recommended bets. This will allow for a better understanding of the success of the model, and it will guide future improvements to the data set.

## Conclusion and Future Work

This project aims to leverage statistical modeling, particularly using an xgboost model, to predict the probability of MLB players getting hits during games. By incorporating various metrics such as expected batting average, exit velocities, and plate discipline indicators, the model seeks to provide more accurate predictions for daily hitter prop bets, thus giving bettors an edge over sportsbooks.

The results of the xgboost model were evaluated based on a cutoff probability of 0.80, focusing on players with a high probability of getting hits. While the overall accuracy of the model may seem low at 38%, the positive predictive value, which measures the accuracy of predicted hits for chosen players, stands at 67%. This metric is crucial for betting purposes since it reflects the accuracy of predictions for players that bettors would actually wager on. The

implementation of these results would involve strategic betting based on the model's predictions. By selectively choosing players with high probabilities of getting hits, bettors can increase their chances of profitability. Visualizations such as confusion matrices and odds calculations provide insights into the potential profitability of using the model for betting purposes.

However, the report also acknowledges certain limitations and challenges, such as occasional days with no players meeting the 0.80 probability cutoff. Additionally, the analysis of parlay accuracy highlights the difficulty and sportsbook advantage associated with this type of betting strategy.

In conclusion, this project demonstrates the feasibility of using statistical modeling to predict MLB player hits for betting purposes. Further steps could involve refining the model by incorporating additional metrics or exploring alternative machine learning algorithms. Additionally, ongoing adaptation of the model's performance in real-world betting scenarios would be essential for optimizing profitability and ensuring its practicality for bettors.

## Contribution:

Jack Arbuckle: 33%, Introduction, Related Work, Results
Lauren Beaudreau: 33% Methods, Conclusions, and Future Work
Ben Scartz: 33% Data Description, Actions

**Bibliography**

**[1]** **https://www.mlb.com/apps/beat-the-streak/official-rules**
**[2]** **https://blogs.fangraphs.com/the-odds-of-hitting-for-the-cycle/**
**[3]** **https://eglouberman.github.io/MLB-hit-predictor/.**