Strategic Swinging Model: Building A Model That Optimizes Swinging Decisions in Baseball

Henry Zhan[1] and Alex Lyford[#]

[1]Middlebury College, USA
[#]Advisor

ABSTRACT

The most important battle in baseball is the battle between the pitcher and the batter, as hitting the baseball hard and far will drastically change the outcome of a game. In this research, we are attempting to build *the* strategic swinging model that can help Major League Baseball (MLB) hitters decide whether they should swing at a pitch before it has been thrown. We used random forest classifiers to output a probabilistic prediction of pitch type and pitch location, and estimated how well the hitter would like the pitch based on his past batting data. We evaluated the model by calculating how much better it performed compared to the scenario in which the batter did the opposite. The model outperformed any random swinging strategy. However, the decisions of most above average hitters are better than the model's decisions.

Introduction

Baseball has always been a world of statistics. It is an individualistic sport, in which the center of the battle is the battle between two players: the pitcher and the batter. It is also a relatively static sport, compared to other sports such as football, which involves 22 players moving on the field at the same time. Therefore, using statistics to measure the performance of players and teams has become a very popular choice. The average baseball fan is usually familiar with statistics such as Batting Average and On-base Plus Slugging (OPS), which are both measurements of a player's batting ability. For statisticians and professional baseball analysts, more advanced statistics are often considered when they try to model the game of baseball.

Modern baseball statistical analytics has its roots in the 1970s, when the term "sabermetrics" was coined by Bill James (Lewis, 2003). "Sabermetrics" means the application of statistical analysis to baseball. Along with the birth of sabermetrics came numerous advanced statistics measuring either pitcher or batter performance. For example, On-base Plus Slugging Plus (OPS+), is a modern statistic that normalizes a player's OPS against the entire league, taking into account external factors such as ballpark dimensions and altitude (Major League Baseball). In addition, Wins Above Replacement (WAR) is a statistic created by sabermetricians that directly measures how many additional wins that can be attributed to this player, compared to a regular MLB "bench player" (FanGraphs). These advanced statistics might be less interpretable, as their method of calculation often seems like a black box to the average baseball fan. Nevertheless, they are very crucial pieces of information for baseball analysts to have a deeper understanding of the game.

With the advancement of technology in the 21st century, more baseball data have become available, allowing baseball scholars to have a more nuanced understanding of the intricacies of a baseball game. The PITCHf/x system, developed by the MLB, tracks the velocity and trajectories of Major League Baseball pitches (Dimeo, 2007). Statcast takes it a step further, by also tracking batted ball trajectories, fielder movements, and base-runner movements (Berg, 2015).

In this research, with the help of Statcast pitch-level data, we aim at building a quantitative system, the strategic swinging model, that can inform the batters on whether they should swing at a pitch or not before the pitch has

been thrown. Major League pitchers often throw pitches that are over 90 miles per hour, and according to neuroscientist Jason Sherwin, batters only have 0.05 to 0.1 seconds to decide whether they should swing (Cascio 2020). Therefore, it would be beneficial if there were a system that can help hitters make that decision pre-pitch. Our model is based upon two separate machine learning models that predict the type and location of the pitch. After that, we also incorporate the batter's historical data and records to help them decide whether they should swing or not.

Literature Review

A substantial body of research has been dedicated to pitch type prediction in baseball, including the research done by Tran and Sidle (2018), in which they utilized multi-class classification models, such as Latent Dirichlet Allocation, Support Vector Machines, and classification trees to predict the type of the next pitch. Their models significantly outperformed simpler approaches like linear regression (Tran and Sidle, 2018). Similarly, Bock developed both a Multinomial Logistic Regression model and a Support Vector Machine model for pitch type prediction. He extended his research further by exploring how a pitcher's predictability could influence their Earned Run Average (ERA) (Bock, 2015).

Hoang (2015) introduced a novel approach by proposing a dynamic feature selection procedure for pitch type prediction, improving upon previous methods that relied on static feature selection algorithms. Additionally, he employed parameter optimization techniques to tackle the issue of class imbalance, leading to superior performance (Hoang, 2015). Ishii (2021) took a different route by developing a pitch prediction algorithm that focused on the pitcher's physical cues, identifying specific "tips" that might reveal the pitch. Yoshihara and Takahashi (2020) applied a probabilistic topic model to analyze pitch sequences, treating each sequence as a topic and each pitch as a word within that topic. Probabilistic topic models, such as Latent Dirichlet Allocation (LDA), are generative models that assume documents are mixtures of topics, where each topic is characterized by a distribution over words(Pritchard, 2000).

Ramos (2017) shifted the focus to predicting the zone of the next pitch. He began by clustering pitches into five distinct zones using the Gaussian Mixture Model and then developed a multinomial logistic regression model to predict the specific zone where the next pitch would likely land (Ramos, 2017). Lee (2022) used deep neural networks to make predictions on pitch type and pitch location at the same time, resulting in 34 different classes).

Despite the extensive research on pitch type and zone prediction, there has been relatively little focus on utilizing these predictions to enhance batters' hitting strategies. Our research builds on these existing studies, aiming to bridge this gap by taking the results of pitch type predictions and applying them to help batters refine their swinging strategies. Our primary research question is: Can we develop a strategic swinging model that leverages historical data to enhance batters' decision-making? Additionally, to what extent, if any, would this new model outperform individual baseball players?

Data and Methods

Table 1. Glossary of terms in dataset

Term	Definition
wOBA	Weighted On-Base Average, a statistic measuring how likely that a batted ball will lead to a base hit
exp_woba1	Expected wOBA calculated from pitch type prediction

exp_woba2	Expected wOBA calculated from pitch location prediction
bip_pct1	Expected contact probability calculated from pitch type prediction
bip_pct2	Expected contact probability calculated from pitch location prediction
strike_prob	Predicted probability that the next pitch is a strike
cdrv_288	A pitch-level statistics that measures the change in the expected amount of runs that will be scored in this inning as a result of a particular pitch. It is calculated by considering all 288 states in a baseball game and taking the difference in the average number of runs scored between two states.

All data used in this research is Major League Baseball Statcast data scraped from Baseball Savant. The datasets from 2019 to 2021 are used for training, and the dataset from 2022 is used for testing. These datasets can be found in the supplementary materials. These datasets include every pitch from the respective seasons, and for each pitch, available information includes balls, strikes, outs, baserunners, velocity of pitch, location of pitch, pitch type, etc. A brief description of each term used in our analysis can be found in Table 1. We excluded Spring Training games, because these games are of a less competitive nature than other games.

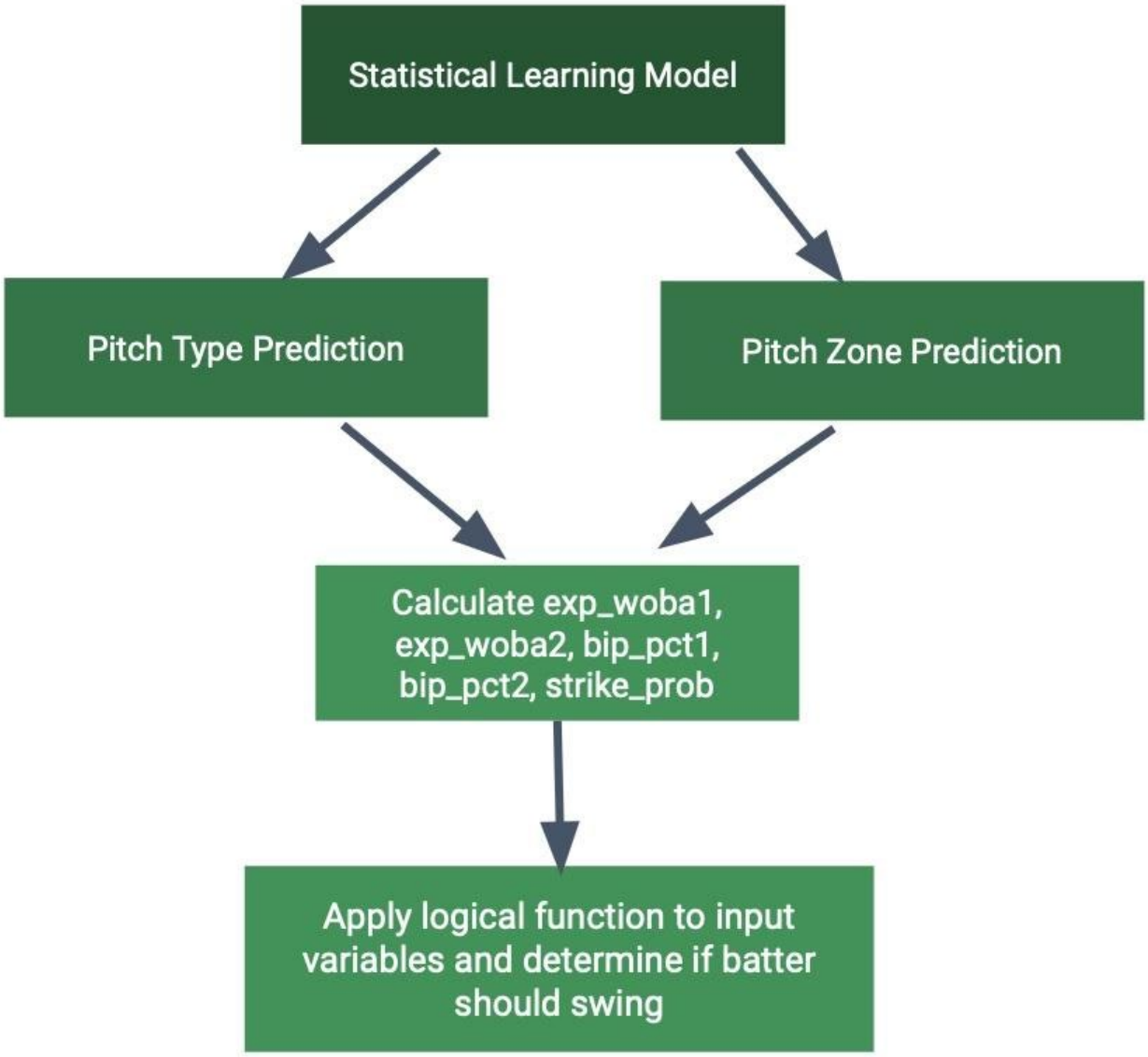

Figure 1. Flow chart of model

Figure 1 is a demonstration of the flow chart of the model. The first part of our research involves training two statistical learning models that can give a probabilistic prediction of the type and location of the next pitch. For this part of our work, we built upon the previous works of Sidle and Tran by using some of the features in their research as the input features used in our model (Tran and Sidle, 2018). Table 2 gives a list of all the features used in our model. Then, we trained two Random Forest Classifiers that will output a probabilistic prediction of pitch type and pitch location respectively.

Table 2. List of features used in random forest models

Feature	Explanation	Type of Variable
inning	The inning of the at-bat	Categorical
pitch_number	The pitch number for the pitcher	Numerical, Discrete
is_bottom	Whether it is the bottom half of the inning	Binary
pa_number	The plate appearance number for the game	Numerical, Discrete
score_diff	Batting score - Fielding score	Numerical, Discrete
is_lhb	Whether the batter is a left-handed batter	Binary
strikes, balls	Balls and Strikes	Categorical
b1, b2, b3	Whether there are runners on first, second, or third base	Binary
previous_speed	The speed of the previous pitch	Numerical, Continuous
previous_break	The break height of the previous pitch	Numerical, Continuous
Previous 5 pitch tendency	Zone and type tendency of the previous 5 pitches	Numerical, Continuous
Previous 10 pitch tendency	Zone and type tendency of the previous 10 pitches	Numerical, Continuous
Previous 20 pitch tendency	Zone and type tendency of the previous 20 pitches	Numerical, Continuous
Previous 5 pitch strike tendency	Zone and type tendency of the strikes among previous 5 pitches	Numerical, Continuous
Previous 10 pitch strike tendency	Zone and type tendency of the strikes among previous 10 pitches	Numerical, Continuous
Previous 20 pitch strike tendency	Zone and type tendency of the strikes among previous 20 pitches	Numerical, Continuous

Random Forest Classifier is a popular ensemble learning method. It is based upon the idea of decision trees. Decision trees are used commonly for classification tasks in machine learning. These trees split data into branches based on feature values. At each node, a feature is selected that best separates the data into distinct classes, and that leads to a final decision at the leaf nodes where classification is made based on the majority class of the data points in that branch (Quinlan 1986).

In random forest, the algorithm constructs multiple decision trees when training by selecting a random subset of the original training set as the training data each time, and eventually outputs the class selected by the most number of trees (Ho, 1995). The algorithm is less vulnerable to overfitting than simple decision trees (Hastie, Tibshirani, & Friedman, 2009).

Table 2 shows the features used in our random forest implementation and a brief description of each. When classifying pitch types, we clustered all different pitch types thrown by Major League pitchers into seven types: Fast-ball(FF), Cutter(FC), Sinker(SI), Slider(SL), Changeup(CH), Curve(CU), and Knuckleball(KN). As for pitch location classification, we kept the way that MLB divides the strike zone, with numbers 1 to 9 indicating different spots that are strikes and numbers 11 to 14 locating different spots out of the strike zone. In addition, we excluded pitchers who threw less than 100 pitches from the 2019 season to the 2021 season.

Once we have the probabilistic output from these two random forest classifiers, we turn our attention to the data from the 2022 season. Figure 2 demonstrates the evaluation method for our model. For each pitch, we compute five different statistics: exp_woba1, exp_woba2, bip_pct1, bip_pct2, and strike_prob. The first four statistics are calculated by taking a weighted average of the statistics of each player from the 2019 to 2021 seasons. Similarly, batters who have seen less than 100 pitches from the 2019 season to the 2021 season are excluded.

Finally, we can make the decision to swing or not based on the condition of different variables. If exp_woba1>0.33, exp_woba2>0.32, bip_pct1>0.37, bip_pct2>0.32, and strike_prob>0.45, the model will instruct the batter to swing. Alternatively, if there are already two strikes and the probability of a strike is high enough—in this case being higher than 0.5—the model will also instruct the batter to swing.

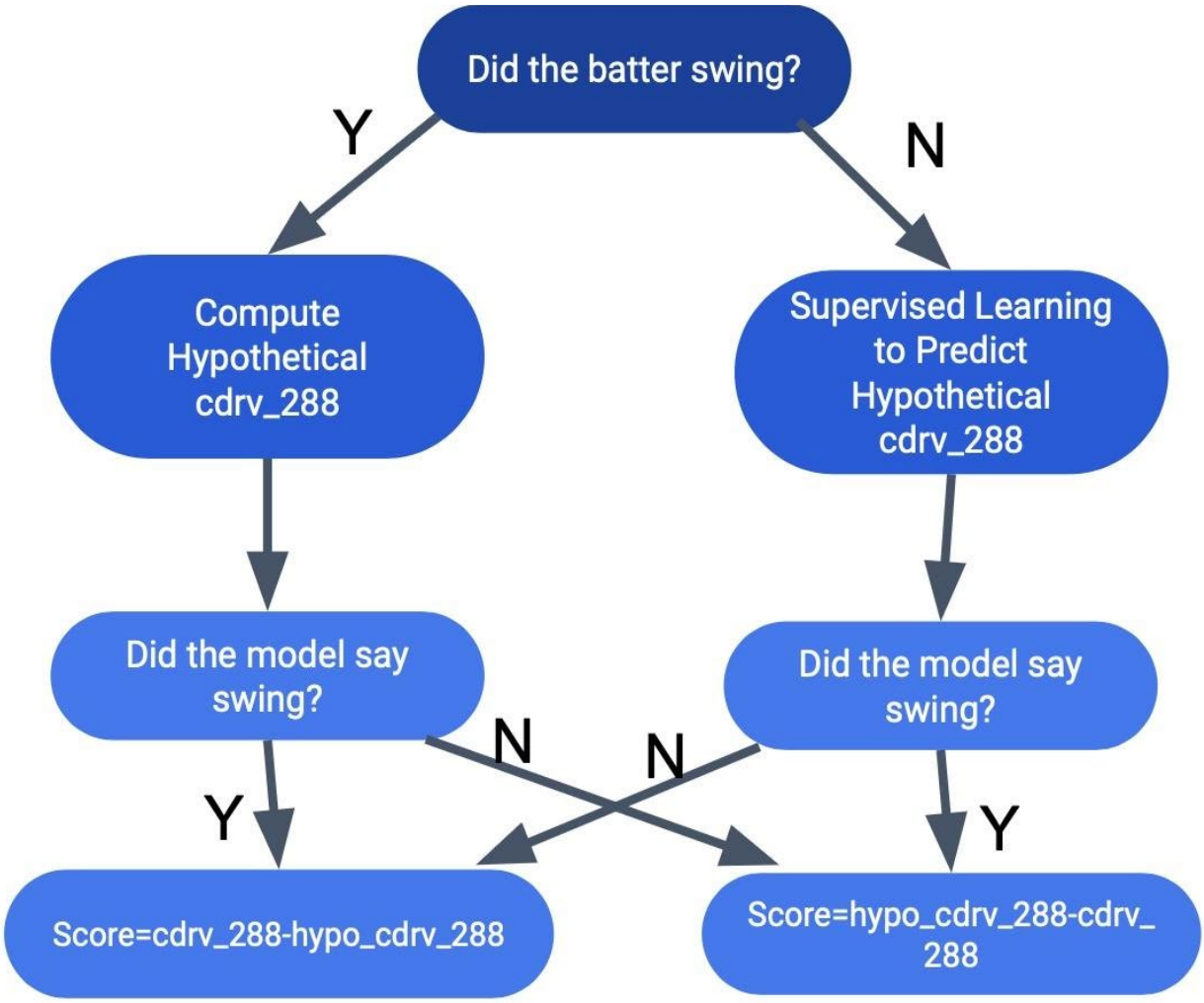

Figure 2. Flow chart of evaluation method

In order to evaluate our model, we first divided every pitch in the 2022 season into two groups: those where the batter swung and those where the batter did not. We consider the statistic cdrv_288 when evaluating the model. In both categories, we calculated a hypothetical cdrv_288 value, which is the cdrv_288 value for the hypothetical scenario in which the batter did the opposite of what he chose to do. For example, if the batter swung at the pitch, then we need the cdrv_288 score for the case where he did not. This can be directly computed, as we can directly infer what would happen if the batter did not swing. However, for the case where the batter did not swing, we trained a random forest regressor to predict the hypothetical cdrv_288 value for the hypothetical case where he did.

Finally, we evaluated how well the model did. If the model's instructions align with the batter's decision, we subtracted the hypothetical value against the real value, as in this case, the more positive the difference, the better our model did. If the model's instructions did not align with the batter's decision, we do the opposite. In this case, the more negative the difference, the better our model performed. All this is completed at a pitch-by-pitch level before we calculate the overall average score of the model.

Results

The 'average score' presented here in Table 3 is calculated by taking the average of the individual scores (as discussed in the previous section) across all pitches from the 2022 season.

Table 3. Average score of the model

Average Score When Batter Swung	Average Score When Batter Did Not Swing	Overall Average Score
-0.0005	0.025	0.0127

Overall, the model has a positive average score of 0.0127. However, once we reduce the results into two groups, the average score of the model is significantly higher for the cases where the batter did not swing than for the cases where he did. The former score is at 0.0249, while the latter one is a slight negative number at -0.005. As we will later see, on average the players outperform the model. Since the model is conservative by nature—it only instructs players to swing about 21% of the time—the model and the batter are more likely to align for the cases where the batter did not swing. Therefore, the model's score will be closer to the batter's score when the batter did not swing than when the batter swung. Therefore, the score for the former case is higher than the score for the latter case.

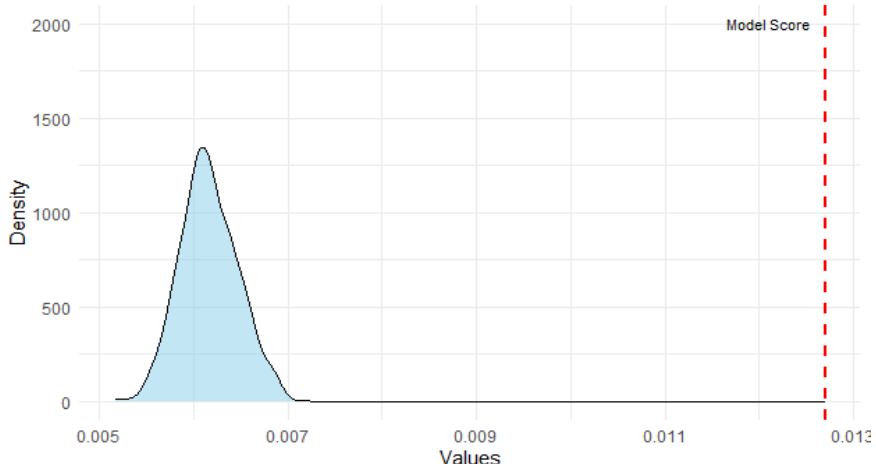

Figure 3. Density plot of the average score of 1,000 random swinging strategies

Figure 3 shows the distribution of the average score of 1,000 random swinging strategies. These strategies are randomly resampled from the strategic swinging model. The proportion of times in which the model determines that the batter should swing is kept the same throughout the resampling process. The average score of these random swinging strategies ranges from around 0.005 to around 0.007, which are all significantly lower than the average score of the strategic swinging model, which is around 0.0127. Therefore, the strategic swinging model significantly outperforms these random strategies.

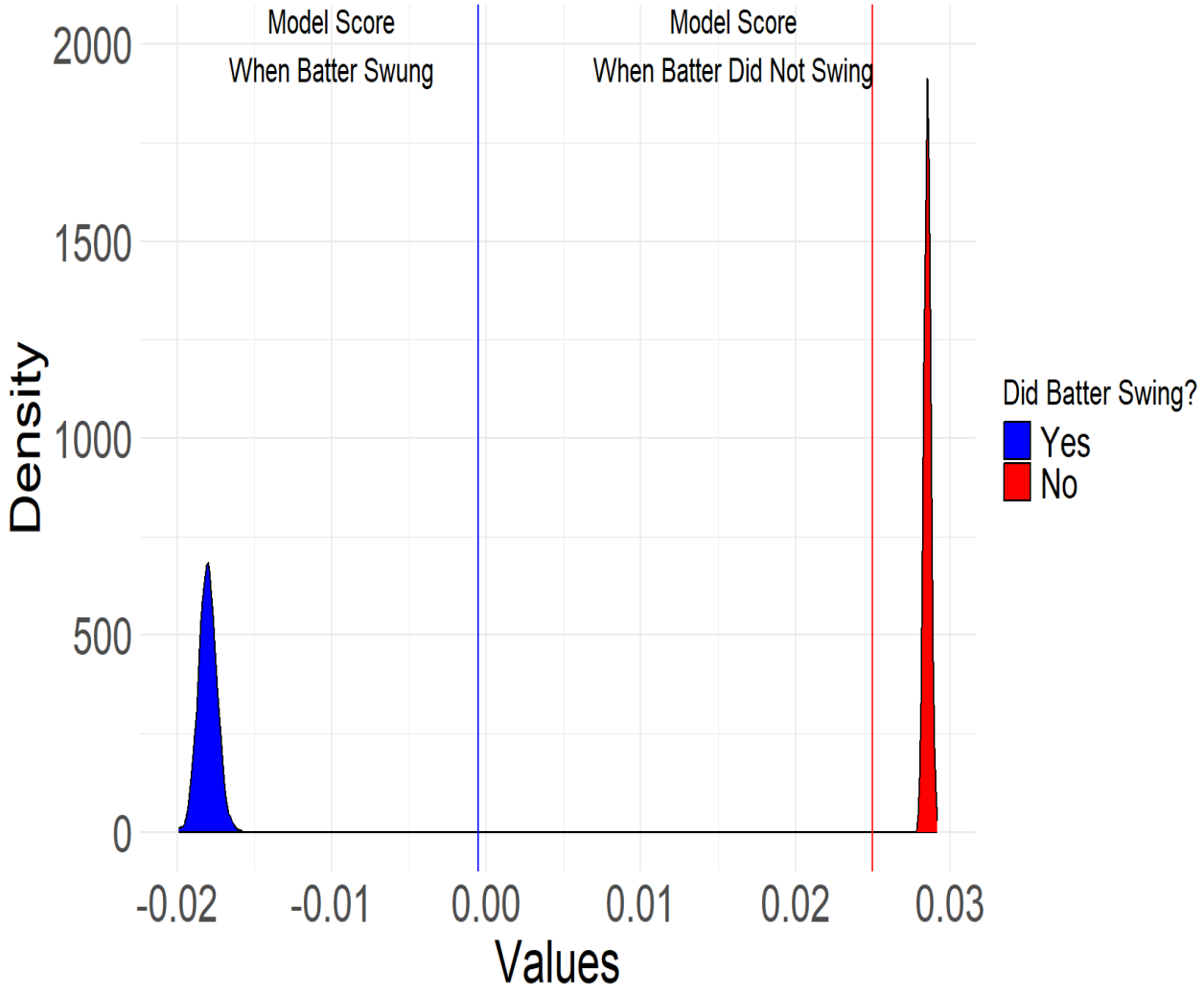

Figure 4. Density plot of random swinging strategies broken down into two groups

Figure 4 further breaks down the average scores of random strategies in Figure A into two groups: those where the batter swung at the pitch and those where the batter did not swing at the pitch. In the former case, the average scores of random strategies range from -0.02 to -0.016. The strategic swinging model, with an average of around -0.005, substantially outperforms these random strategies under this scenario. In the latter case, the average scores of random strategies are all around 0.028. It is interesting to note that these strategies outperform the strategic swinging model in this case, as the model only has a score around 0.025. When combining the two cases together, the model outperforms random strategies, because its advantage in the case where the batter swung more than offsets its disadvantage in the cases where the batter did not swing.

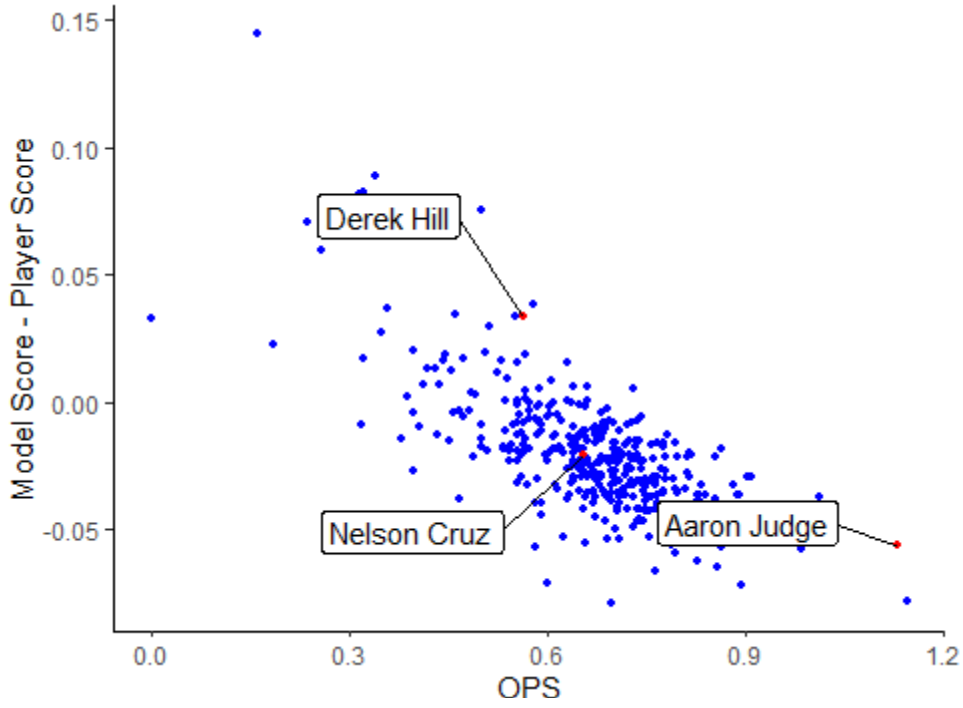

Figure 5. Quantitative advantage of strategic swinging model (y-axis) versus On-base Plus Slugging (x-axis)

Figure 5 shows how well the strategic swinging model performed for each player compared to the player's own decision to swing or not. These data represent all players from the 2022 season—excluding rookies—with at least 10 at bats. Rookies were excluded from this analysis because the strategic swinging model uses a player's past performance to predict the hypothetical result of swinging at a pitch that they did not swing at. The y-axis represents the difference between the model's score and the player's score—positive numbers indicate the model outperformed the player's decisions. The x-axis represents On-base Plus Slugging (OPS), a measure of hitter success. As a player's OPS increases, the model is more likely to underperform relative to the player's own decision making. Prolific hitters—such as Aaron Judge—tended to outperform the strategic swinging model, whereas players with lower OPS—such as Derek Hill—did significantly worse than the model. An average baseball hitter, such as Nelson Cruz, will tend to sit somewhere in the middle.

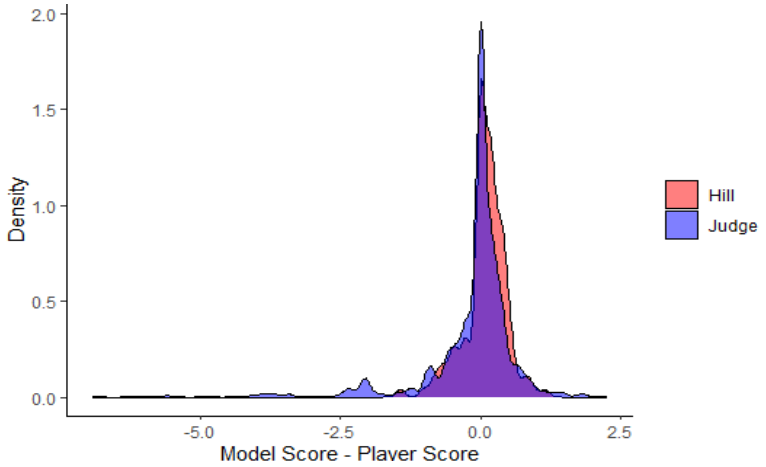

Figure 6. Distribution of model vs player score differences for Aaron Judge and Derek Hill

Figure 6 shows the approximate distribution of pitch-level differences between the strategic swinging model's score and the swinging decisions of two different players—Aaron Judge and Derek Hill. Decisions where the model and players agreed about whether to swing or not to swing have a score difference of zero and are not included in this graph. These scores of zero constitute 59% of all pitches. For Aaron Judge, the median of this distribution is nearly zero—approximately half of the values are positive, and half are negative. The distribution is heavily left skewed, however, and the mean is -0.14, indicating that Aaron Judge's swinging decisions have a 0.14 higher score than the strategic swinging model on average. This negative average is driven by a small number of pitches where the strategic swinging model determines that Aaron Judge should not have swung. In these cases, however, he chooses to swing, and the result is very positive (e.g., a home run). As for Derek Hill, the shape of the distribution is similar except that it is a bit more to the right. The mean of the distribution for Derek Hill is around 0.08, indicating that the strategic model on average has a 0.08 higher score than Derek Hill's swinging decisions.

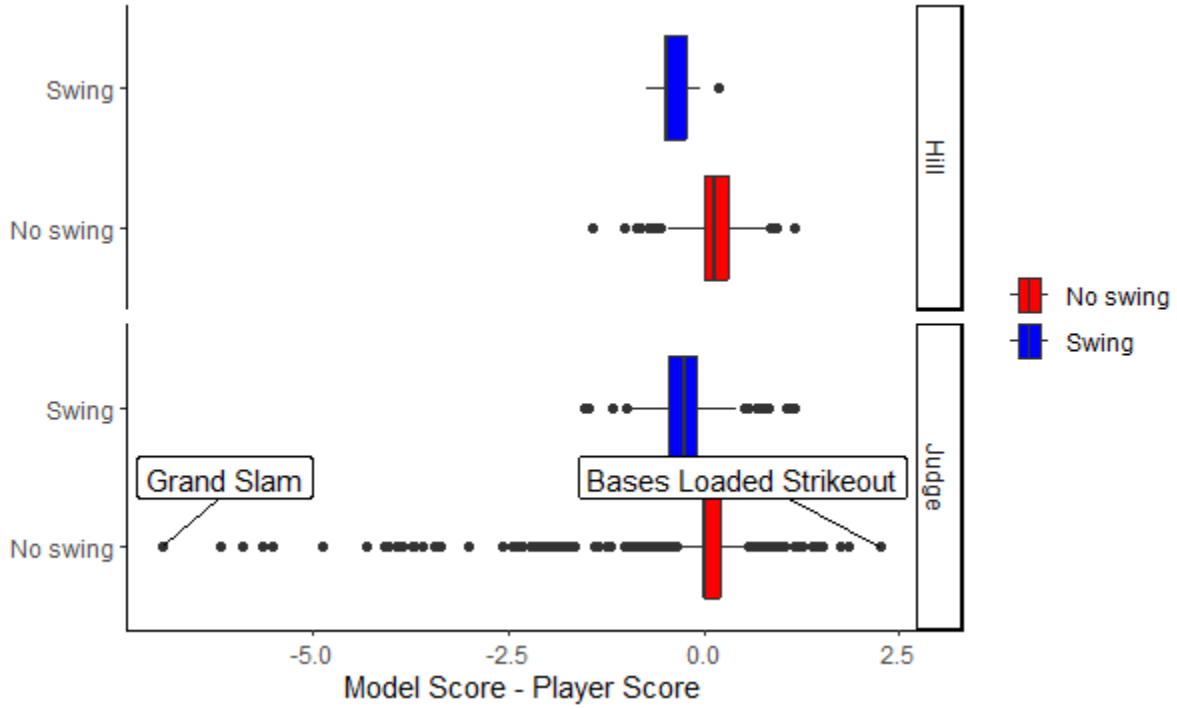

Figure 7. Broken down boxplot of model vs player score differences for Aaron Judge and Derek Hill

Figure 7 further refines the distribution shown in Figure 6 by categorizing the data into two distinct groups: instances where the strategic swinging model suggests that the players should have swung—yet they did not—represented by the boxplots at the top of each graph, and instances where the model advises against swinging—yet they did swing—represented by the boxplot at the bottom of each graph. As for Aaron Judge, the graph indicates that in the first scenario, where the model recommends swinging but the players refrain, Judge's decision generally outperforms the model's recommendation, as evidenced by the center of the boxplot being clearly positioned to the left of 0. In contrast, in the second scenario, where the model advises against swinging but Judge swings anyway, the model's recommendation tends to slightly outperform Judge's decision. However, the numerous extreme points show that the score differences in this second category exhibit significantly higher volatility. For instance, in one game, Aaron Judge came up to bat with the bases loaded. Despite the model's advice not to swing, Judge did and hit a grand slam. This

results in a score difference of -6.92. Conversely, in another game with the bases loaded and a full count, Judge ignored the model's recommendation to refrain from swinging. He struck out and left the runners stranded, leading to a score difference of +2.45.

Therefore, if we were to count the amount of times the model outperformed Judge vs the amount of times Judge outperformed the model at a pitch level, the amount appears to be roughly equal. However, according to the way we evaluate the model, the model will suffer from a huge penalty whenever Judge acts against the model and does well. That explains why the overall difference is a negative number. The boxplot for Derek Hill tells a similar story except that the distribution for when he did not swing is more to the right, resulting in an overall positive average.

Discussion

Overall, our strategic swinging model does perform better on average than almost all random swinging decisions. However, the model does not necessarily outperform good hitters. The swinging decisions of more prolific baseball players with a higher OPS were often better than the model's decisions, at least according to the way we evaluate the decisions. In addition, for cases in which the model suggested the batter not swing but they did, most of the time the model is making the better decision. Nevertheless, when the model is making the wrong decision in this scenario, it usually takes a huge penalty—such as when the player gets a multi-RBI hit that drastically changes the game. Therefore, the average score of the model ends up being worse than the average score of batters' strategies. The model is by nature more conservative than MLB hitters, as it only instructs players to swing around 21% of the time compared to the roughly 50% swing rate of MLB players. It is designed to identify those pitches that are very 'hittable' and then give hitters the instruction to swing.

The efficacy of our strategic swinging model is limited by three main factors—limited data in particular scenarios, the difficulty in predicting and quantifying the result of hypothetical cases where the model predicts a batter should swing when they do not, and quantifying the success or failure of the result of a single pitch. Since our model requires previous seasons' hitting data for the given player in order to make an informed decision about whether or not they should swing, the model does not work for rookies or players without previous hitting data. Although we did not try to make predictions for rookies, their data could theoretically be imputed using league averages in the given scenario in order for our model to make a swing suggestion.

Evaluating the cases where the model predicts that a batter should swing—but the batter does not—is challenging. Our current approach uses a player's past performance for the given scenario—including pitch type, pitch location, and count. It's likely that tricky-to-measure in-game metrics like weather, fatigue, and momentum could change the 'typical' result of such plays, ultimately changing the evaluation of the efficacy of our model. Major league hitters are all very experienced baseball players, and they have developed their own ways of making swinging decisions throughout their career. In addition, as players on the field, they have access to additional information that is not captured by our data, including specific cues and body movements from pitchers. This information constitutes a significant part of their decision making.

Developing a 'fair' metric to assess the result of a single pitch was challenging. Our approach relied on calculating (or estimating in cases where this value was incalculable) changes in cdrv_288. This variable is constructed using league averages for the change in expected runs scored before the pitch is thrown and after the result of the pitch. Several factors—including the specific player at bat, the combination of players playing defense, or the current state and the stakes of the game—certainly influence the change in expected runs scored before and after the pitch. Future work might consider using a different, more nuanced method for evaluating the efficacy of the model, such as assessing the result of the entire at bat instead of using pitch-by-pitch changes in expected runs scored. Also, the hyperparameters for the decision making process could be tuned. If given more time, we would run a grid search to look for the best hyperparameters that can maximize the model score.

References

1. Berg, T. (2015, May 6). MLB's Statcast is already changing the way we watch baseball. *USA Today*. https://ftw.usatoday.com/2015/05/mlb-statcast-stats-data-launch-angle-route-efficiency
2. Bock, J. (2015). Pitch sequence complexity and long-term pitcher performance. *Sports,3,* 40-55.
3. Cascio, J. (2020, October 23). The science behind hitting a 95 mile-per-hour fastball, explained by experts. *FOX 13 News*. *https://www.fox13news.com/news/fast-pitching-world-series#*
4. Dimeo, N. (2007, August 15). PITCHf/x: The new technology that will change baseball analysis forever. Slate. https://slate.com/culture/2007/08/pitch-f-x-the-new-technology-that-will-change-baseball-analysis-forever.html
5. FanGraphs. (2012, March 16). Wins above replacement (WAR). FanGraphs. https://library.fangraphs.com/war/
6. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd ed.). Springer.
7. Ho, T. K. (1995). Random decision forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition* (pp. 278–282).
8. Hoang, P. (2015). Supervised learning in baseball pitch prediction and hepatitis C diagnosis(Doctoral dissertation). North Carolina State University
9. Ishii, B. (2021). Using pitch tipping for baseball pitch prediction (Doctoral dissertation). California Polytechnic State University
10. Lee, J. S. (2022). Prediction of pitch type and location in baseball using ensemble model of deep neural networks. *Journal of Sports Analytics, 8*(2), 115–126. https://doi.org/10.3233/JSA-200559
11. Lewis, M. M. (2003). *Moneyball: The art of winning an unfair game*. W. W. Norton.
12. Major League Baseball. (n.d.). On-base plus slugging plus (OPS+). Major League Baseball. https://www.mlb.com/glossary/advanced-stats/on-base-plus-slugging-plus
13. Pritchard, J. K., Stephens, M., & Donnelly, P. (2000). Inference of population structure using multilocus genotype data. *Genetics, 155*(2), 945–959. https://doi.org/10.1093/genetics/155.2.945
14. Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106. https://doi.org/10.1007/BF00116251
15. Ramos, D. (2017). Creating a hitter's approach: analyzing at-bat data [Master's thesis, Texas State University].
16. Tran, H., & Sidle, G. (2018). Using multi-class classification methods to predict baseball pitch types. *Journal of Sports Analytics, 4*(2), 85-93.
17. Yoshihara, K., & Takahashi, K. (2020). Pitch sequences in baseball: Analysis using a probabilistic topic model.