Predicting Swing Probability in College Baseball

Macklin Yarris, mmy5167@psu.edu, Jiso Awe, moa5484@psu.edu

***Abstract*—The evolution of data tracking in baseball has grown to allow players, coaches, and data scientists to better understand the game. Through our research, we have been able to generate various models focused on predicting the likelihood a batter will swing at a given pitch. Through aggregating scrimmage data collected by Penn State's version 3 of TrackMan, we were able to test out the accuracy of different machine learning models. During our research we created multiple models that utilized several machine learning classification techniques to examine the likelihood a given pitch is a swing and misses. When we emphasized certain features of the pitch, we found that the model with the most success in determining swing probability was the XGBoost model we developed. Through our research and work on this project we were able to build a stronger understanding between how a pitch's location, speed, and movement characteristics relate to the swing outcome of the pitch.**

I. Introduction

The purpose of our data science project is to use data of the Penn State Baseball team gathered with Trackman technology to build predictive models. Baseball is the sport where analytics are the most prominent; as other sports are newly learning how to implement data science practices, baseball managers and team executives have been using analytics for years to maximize team performance. This practice is known as sabermetrics, and due to how long it has been implemented, there are many machine learning models that already exist that do a good job of providing quality insight into baseball performance. As Penn State students, we hope to develop a model using knowledge we already have about baseball statistics to build predictive models that can benefit the Penn State baseball team. More specifically, we hope to build a model that can predict the probability that a collegiate batter will swing on any given pitch.

There are many different things to consider as we approach this classification task. First of all, we need to consider that publications related to our research are different for many reasons. Much of published research that aims to build predictions for baseball outcomes is done using MLB data. Sabermetrics in the MLB is very advanced, and so many useful insights have already been found. Limiting our research to NCAA baseball, we recognize that their is less data at our disposal. Although this means that it can be harder to find related work that can help us advance ours studies, it also means that there is a lot of room for improvement in the use of sabermetrics in college baseball. With our research, we have the chance to develop an accurate model that could improve the use of analytics in college baseball, giving players and staff of different NCAA baseball programs valuable tools for improvement.

We hope that our project will be able to benefit the Penn State Baseball team's players and coaches in their goal of maximizing performance in their competitions. A potential goal we have for this project is to build multiple models that can accurately predict different components of a baseball game; rather than focusing solely on predicting the probability a pitch will be swung on, we may want to additionally focusing on models that can make predictions applicable to the various pitches that can possibly be seen from opponents. The main goal of this project was to use Trackman data as effectively as we possibly can. We want to gain valuable insight that will improve Penn State Baseball's performance and ensure that they are using the vast amount of data that they have in order to see positive results.

Our research can significantly contribute to the field of sabermetrics in baseball, particularly at the collegiate level. There is not a vast amount of publications addressing how to improve the implementation of sabermetrics in college baseball. Sabermetrics has become increasingly popular in Major League Baseball, but remains underdeveloped and unexplored for college baseball teams. Also, our research is unique as it works towards building a highly accurate model that can predict whether or not a certain pitch will be swung at. This is an exploratory topic not commonly addressed, but if made correctly can help advance college baseball significantly. There are endless possibilities with a model that addresses this specific part of baseball performance. Our findings can help improve the scouting capabilities of baseball coaches leading to an improvement in their preparation for competition. It also has the ability to enhance the analysis of specific batter vs. pitcher match ups, giving teams that use this model a competitive advantage in how they prepare for opponents.

The final results of our model show that it is fairly accurate and can efficiently make predictions on whether or not a pitch will be swung at. Our best performing model had an Area under the ROC curve measurement of 0.86, classifying its performance as highly accurate. Our research efforts, if they are continued, can contribute to the use of sabermetrics in college baseball. If we expand on the research we have already done, we could potentially train this model with real time, in-game data, making it useful during actual competition rather than limiting its use to scouting efforts. We could also train the model with data on individual pitchers and batters, improving the knowledge college baseball players and coaches have on pitching and batting performance, highlighting what pitch and hit strategies are good, bad, or ineffective. We hope that our research can contribute to the college baseball community by improving the capabilities of their sabermetric efforts.

II. Related Works

Given how prominent sabermetrics is in the sport of baseball, we understand that many researchers have attempted to tackle similar problems before. Reviewing other published works of people using machine learning techniques to solve baseball related problems helps give us insight on what kind of methodology has produced successful results in the past, and also gives us a good idea on how to improve on already existing work.

The first paper we looked at was entitled "Machine Learning Applications in Baseball: A Systematic Literature Review" written by Kaan Koseler and Matthew Stephan [1]. In this publication, Koseler and Stephan dive into machine learning's growing involvement with baseball and its many applications in the sport. They were able to focus on two research questions:

- What are all the different ways ML has been applied to baseball?
- What is the distribution of these applications across the ML problem classes?

Koseler and Stephan conducted this research with these two guiding questions, and some of their findings can apply to our research as well. Koseler and Stephan found that one of the most common types of ML application in baseball is Binary Classification. In this paper, the authors look at how two researchers used Linear Support Vector Machines to classify pitches with MLB data from 2008 and 2009, attempting to predict whether a pitcher's next pitch will be a strike or not. The results showed that their model was significantly better than a Naives Bayes classifier, predicting the result of the next pitch correctly around 70 percent of the time.

Koseler and Stephan go through many ways in which machine learning techniques are applied in baseball predictions. The results of their findings were that they found a total of 5 articles exploring Binary Classification, 8 articles for Multiclass classification, and 19 articles for Regression problems dealing with MLB data. Our research is on a Binary Classification problem, which there seems to be a lack of compared to other problems according to the findings of Koseler and Stephan. Reviewing this work reinforced our idea that building a Binary Classifier for batter swinging would serve as a useful application of machine learning techniques in baseball.

Another publication that we reviewed was entitled "Applying Machine Learning Techniques to Baseball Pitch Prediction." This was written by Michael Hamilton, Phuong Hoang, Lori Layne, Joseph Murray, David Padgett, Corey Stafford and Hien Tran. This team of researchers looked at MLB data with a goal of using machine learning classification methods to classify pitches by different types, ultimately limiting their features to only information known before a given pitch is thrown.

The findings of this publication are particularly interesting because they were able to determine that many features that are presumably important when predicting pitch type had no significant effect on their machine learning model. For example, the often discussed handedness matchup between pitchers and batters surprisingly had no effect on this model's pitch predictions. This publication does a great job of showing how important a process feature selection is; after reviewing the results of this paper, we feel like we have a better understanding of how to determine the importance of different variables when it comes to predicting swings.

A paper written by two former University of California Berkeley students, Dibya Ghosh and Maxwell J. Weinstein, focuses on finding a connection between what the batter sees when a pitch is thrown and their decision of whether or not to swing at the pitch. They scraped about six million pitches from the 2008-2014 Major League Baseball seasons from the pitch database PitchFX. PitchFX generates data through the use of high resolution cameras which capture numerous pitches for each pitch and from these pictures they are able to determine the physical trajectory parameters of the pitch. The pair approached this data by focusing on the connection through the use of physics. Baseball physicist Alan Nathan suggested a nine parameter model that would approximate the location of the ball at the end of the pitch within 0.5 inches. PitchFX provides data on initial velocity, acceleration, and position and through the use of singular value decomposition they were able to determine the final location of the pitch. Through this they were able to create models of pitch movements and predict whether or not the pitch would be swung at, which is what our model focuses on predicting.

In addition to these publications, we believe that our research can provide new, insightful findings into how machine learning techniques can be used to improve the use of sabermetrics. These publications have produced lots of significant results, but we believe our work is different for a few reasons. Most notably, most of the research on how machine learning can be used to make predictions in baseball is done with MLB data. We have made the scope of our research College Baseball, focusing on data gathered from games between Penn State Baseball and their opponents. In general, the strike zone for college baseball is very different from the strike zone in the other exhibition levels. Traditionally, in NCAA baseball the strike zone is shorter and wider, which affects how players interact with pitches. Due to the fact that umpires are humans, they are a lot more prone to error and biases when calling pitches. This makes pitch location data different from data found from related research.

Also, with our project we are trying to predict swing probabilities based on pitch types and metrics. This idea has a very explicit actionable purpose. Most baseball related projects have been focused solely on pitching, but our project takes both pitching and hitting into consideration for the model.

With a combination of pitching and swinging data, we hope to build a model that can predict whether or not a pitch will be swung on, helping players and coaches of college baseball improve their performance and analytical review of their data.

III. Data

Our dataset was gathered with TrackMan technology during the Penn State Baseball Team's Fall 2021 scrimmages. TrackMan records lots of data on each individual pitch, including scientific and precise measurements of a pitch's location, speed, spin velocity, break angles, etc. It also records the outcome of each pitch, specifically if it was swung at and missed, classified as a foul ball, or if it was put into play. The entire dataset consists of over 3,200 pitches from 33 different college baseball pitchers. A preview of this data can be seen in Figure 1.

	PitchNo	RelSpeed	SpinRate	InducedVertBreak	HorzBreak	PlateLocSide	PlateLocHeight	ZoneSpeed
0	1	91.31813	2115.387638	17.54001	14.25583	2.20333	2.98657	83.24518
1	2	92.25342	1671.583699	12.53052	10.74146	0.41052	3.44076	84.87349
2	3	81.35562	2236.681310	-2.12973	-7.99288	-0.34575	2.44097	75.66338
3	4	81.62514	2569.976060	3.08008	-10.24032	-1.91768	0.87508	76.32373
4	5	93.31665	1918.602426	13.62296	12.53012	-0.10321	1.32272	86.28469

Fig. 1. A quick look at the landscape of our dataset, in csv format viewed in a Python environment.

IV. Methodology

To better understand our data we examined the dataframe and the metrics that are measured. We first looked on TrackMan's website for intuitive explanations as to what all the variables mean. Penn State utilizes the v3 edition of the TrackMan system; when a user ends the game, the data is condensed into a comma separated value file and uploaded onto the TrackMan server. The file produced can contain hundreds of columns of data, about each pitch and any action that resulted from that pitch. Due to the type of information that TrackMan monitors, there are numerous variables that have NA values present. For example, if a pitch wasn't swung at, there will be no data present about the exit velocity or bearing. Due to this, we were aware that not all of the variables would be significant in understanding what type of pitches would be swung at. After researching what the numerous variables mean and how thet are measured, we were able to find subset quantitative variables that we could evaluate. We had some initial theories on variables that would be most useful, including spin rate, pitch location in relation to home plate, horizontal and induced vertical break, and release side.

Some of the work we tried to incorporate in the preprocessing of our project was examining the data through visualizations. The three visualizations we focused on were the umpire report, the heatmap, and the correlation matrix. The first visualization is the umpire report, this visualization plotted out the locations of all the strikes and balls from a game. The purpose of creating this visual was to point out how prone to error umpires can be, especially compared to what the strike zone is. You can see from the visualization that the umpire was highly inconsistent on what pitches were called balls and which ones were called strikes. First we were able to draw a visual representation of the strike zone to understand the relative location of the pitches. In baseball all of the balls inside of the strike zone should be considered strikes and all the pitches outside of the zone should be balls. This visualization assisted us in emphasizing how unreliable umpires can be.

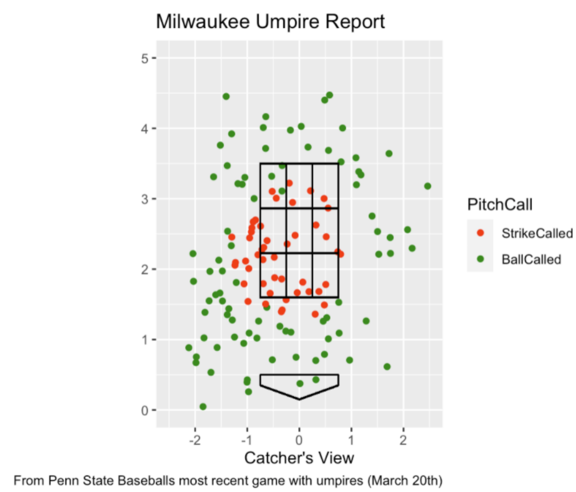

Fig. 2. An umpire report showing pitch location and whether or not each pitch was classified as a strike or ball by the umpire.

As we can see from Figure 2, umpires in games are often making mistakes due to the impact human error has on effectively identifying pitch locations in real time. Although our location data is scientifically precise, it still does not account for the fact that the batter and umpire may be seeing the same pitch but making different assessments on whether or not it is a strike. This will affect how important pitch location data is in the accuracy of our model's predictions. To us, this means that there is room for improvement on how we decide to analyze location data in this dataset. During the feature engineering process, it might be useful to work on creating new variables that can account for the actual location of the pitch compared to whether or not it was called a strike or a ball.

The next visualization we made was the swing and miss heatmap seen in Figure 3. The purpose behind creating the heatmap was to see a general area of where the swinging strikes had occurred from the dataset. First we were able to draw a visual representation of the strike zone to understand the relative location of the pitches.This also helps us to get a better understanding of whether the pitch would have been called a ball or strike if it wasn't swung at.

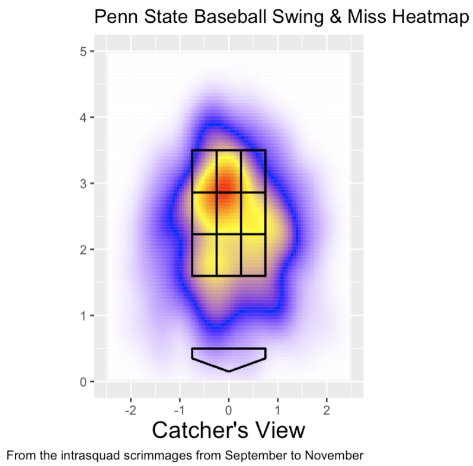

Fig. 3. Heat Map showing areas of the strike zone with high and low swing frequencies.

Finally, we created a correlation matrix (Figure 4) during the preprocessing of the data to find any trends between the different metrics. Using background knowledge we were able to identify some statistics that we knew we needed to include in the project, such as horizontal break, induced vertical break, and spin axis. Through using various packages in R, we were able to find some levels of correlation between all of our selected variables, which aided us in creating the model. This step was crucial in providing insight as to what features we thought would be important indicators of whether or not a pitch would be swung at.

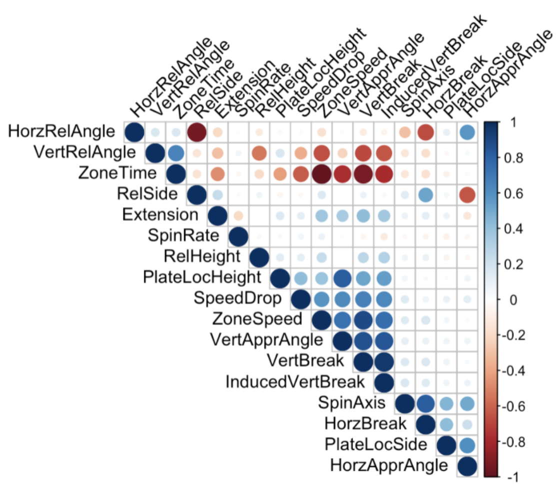

Fig. 4. Correlation Matrix visualizing how different pitch measurements related to one another.

From our correlation matrix we can see how different features relate to each other. Correlation values closer to 1 reflect positively correlated attributes, while negative values depict a negative relationship between variables in the dataset. With this correlation matrix, we can see that variables relating to a pitch's location are heavily correlated with variables that describe a pitch's breaking angles. As we already know how important pitch movement is in determining whether or not a batter will swing at a pitch, this matrix showed us which specific variables will be significantly related to one another, providing insight on what we should consider including as predictors in our model.

Experimenting with the data and creating different visuals was a crucial step in figuring out how we were going to approach the Binary Classification task of predicting whether or not a pitch was swung at. It is important to understand that human error plays a significant role in whether or not a pitch is called a ball or a strike; while an umpire might see a strike, a batter might see a ball and feel no inclination to attempt a swing. The location data gathered by TrackMan technology includes no human error, so it is important to factor in the difference between real time analysis and technological data gathering and how both affect the predictions in this research. It was also important to quantify the importance of some of these features in the dataset. Using correlation as our main metric to determine feature importance, we were able to assume how important some of our variables would be in predicting swings. This provided us with a good start on how to approach classification, but we did not limit ourselves to only one metric to determine feature importance.

V. Experiments

We were able to access data for all of the Fall 2021 games for the Penn State College Baseball Team. This data was collected with Trackman technology and placed into a CSV file with each recorded row representing a pitch thrown. There is lots of valuable information taken on each pitch. There are descriptive variables describing both the batter and the pitcher such as what side they bat from and what hand they throw with. There are also detailed recordings about what type of pitch each recording is; the different pitch types are big indicators on whether or not a batter will swing, and the location and speed of these pitch types are detailed with precise recordings. We first cleaned the dataset and built our target variable. The variable "PitchCall" is recorded with 8 possible values: "BallCalled", "StrikeCalled", "StrikeSwinging", "InPlay", "BallinDirt", "FoulBall", "HitByPitch", and "BallIntentional." From this information, we created a variable called "swungAt" which we initially assigned a value of 0, subsequently assigning every row with values "StrikeSwinging", "InPlay" and "FoulBall" a value of 1. This resulted in a binary target variable where 0 represented no swing while 1 indicated that the batter had swung.

Next we test what features we figured would be important in predicting swing data for these pitches, hoping to ultimately build a predictive model. From our correlation matrix, we were able to take a closer look at some of the important variables and how they related to one another. We started by initially selecting ten important explanatory variables:

Pitch Speed, Spin Rate, Vertical Break, Horizontal Break, Pitcher Handedness, Batter Side, Pitch Type, Pitch Height, Pitch Side Location, and Zone Speed. With these features, we decided to experiment with feature engineering to build new variables. We ended up building features that represented whether or not a pitch was faster or slower than the average pitch speed, as well as whether or not a pitch's location was higher or lower than the average pitch location. Developing these features quantified how much a pitch deviated from an average looking pitch that these batters would see, giving us good variables to represent how ordinary a pitch was. To complete feature selection, we dropped all "irrelevant" features, which we defined as features with a correlation of less than 0.025 to the target variable.

We wanted to test multiple classification algorithms so we could compare the accuracy measurements for different algorithms, ultimately helping us select the best approach. We split the data into 80 percent training and 20 percent testing. We used four different algorithms to fit our data: Logistic Regression, Support Vector Machines, Random Forest, and Naive Bayes. The data of Table 1 shows the comparison of our results.

Model	Accuracy	Precision	Recall
Logistic Regression	0.606	0.152	0.446
Support Vector Machines	0.596	0.123	0.4
Random Forest	0.743	0.658	0.664
Naive Bayes	0.667	0.761	0.544

TABLE I
ACCURACY, PRECISION, AND RECALL MEASUREMENTS FOR DIFFERENT ALGORITHMS.

Our best model in this collection was Random Forest, which recorded an accuracy score of 0.743. This is not ideal, as it means that our model is making incorrect predictions on about 25 percent of the pitch outcomes. Before proceeding with an attempt to try improving our model's accuracy, we wanted to gain insight on our model's performance and what features were important in making predictions These can be seen in Figure 5.

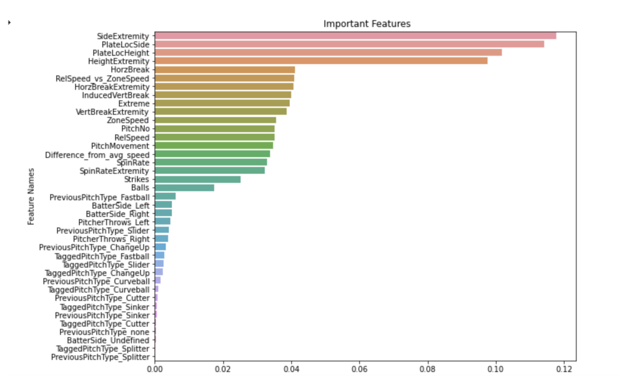

Fig. 5. Feature Importance from our Random Forest model.

The results from these experiments show us that accuracy is most likely not a sufficient metric for performance. The low precision and recall for the four models used for making predictions leads us to believe that the dataset may be imbalanced. This caused us to compute a confusion matrix of the Random Forest model's predictions to help us better understand our results, seen in Figure 6:

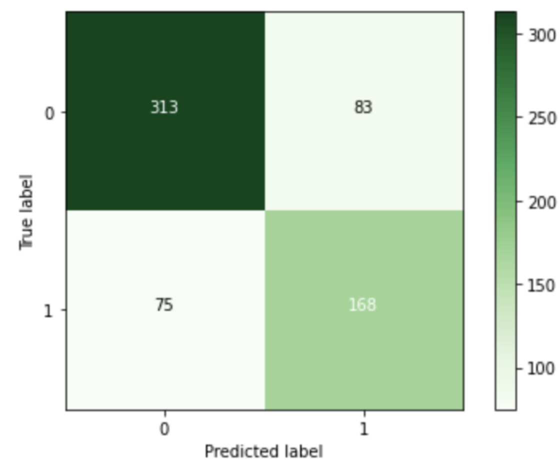

Fig. 6. Confusion Matrix showing the results of predictions made by our Random Forest classifier.

The results of this confusion matrix show us the true and false prediction, and we can see that it performs well with 478 true positive and 335 true negative predictions. However, we can see that the dataset is slightly imbalanced, meaning we need to focus on another metric to evaluate the predictive accuracy of our model. This led to us introducing XGBoost, a machine learning model that implements a Gradient Boosting machine learning technique to combine multiple machine learning algorithms to build a strong and accurate predictor. The Gradient Boosting technique creates a combination of weaker models, typically in the form of decision trees, to build a predictive model. A mapping of a Gradient Boosting technique can be reflected by Figure 6:

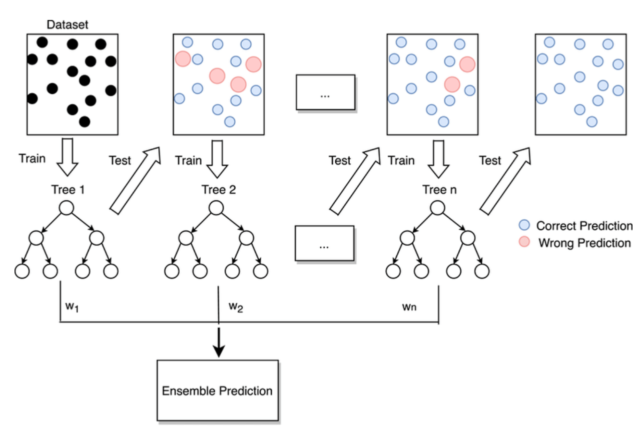

Fig. 7. A general mapping of the Gradient Boosting machine learning method.

Before moving forward by making predictions with the XGBoost algorithm, we decided to attempt feature engineering once again, building quality predictors for our dataset. The selection of pitches is a highly strategic process for pitchers and their coaches, and it is determined with many considerations. This means that as batters face off against pitchers, the pitch type and previous pitch type both play a significant role in a batter's decision making. Considering this strategic aspect of at-bats in baseball, we developed a feature that records what type of pitch the previous pitch was.

It was also important for us to create new features taking into account how extreme a pitch was based on its speed, movement and placement. We created features calculating how extreme a pitch was compared to an average pitch in the dataset, combining factors such as spin rate, zone speed and location and computing how different they were from typical pitches.

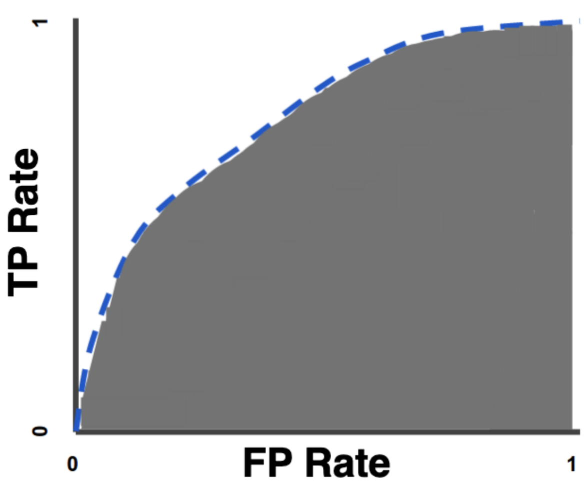

Fig. 8. General ROC curve and the area under the curve (AUC).

Due to the imbalance in our dataset, we decided to evaluate the performance of our model using Area under the ROC curve (AUC). An ROC curve plots the true positive rate against the false positive rate of model predictions. An illustration of this can be seen in Figure 8. Using AUC to evaluate the performance of our model makes sense due to how unreliable accuracy is when the data a model is trained on is imbalanced. Using the features in Figure 9, we created an XGBoost model that had an AUC score of 0.858 on the testing data, meaning XGBoost was much more accurate than the other algorithms we used, even significantly better than Random Forest, which was previously our best model.

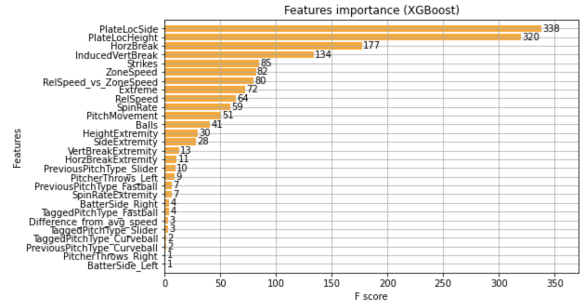

Fig. 9. Features included in our first XGBoost model and their importance.

Figure 9 also depicts how important some of the features were in our XGBoost model. From these results, we learned that our feature engineering process was both productive and unnecessary. Features we created accounting for how extreme pitches were based on their location and movement were significant in helping our model predict whether or not the pitch was swung at. However, it was surprising to discover that the previous pitch type had such insignificant importance in our model's performance. Testing for feature performance helped us determine what features to include in our revised model, which can be seen in Figure 10.

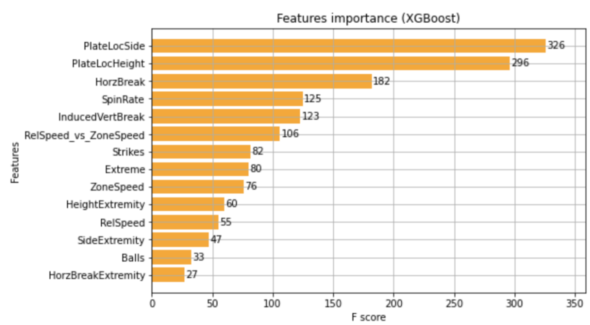

Fig. 10. Features included in our best performing XGBoost model.

Making our model simpler by removing unimportant features and only including features with a feature importance score over 20 helped us increase the AUC score on the testing data to 0.86, making this XGBoost model our most accurate predictor yet.

VI. Discussion

After lots of data preprocessing and feature manipulation, our most accurate predictions came from fitting our model with XGBoost. We were able to build a model that predicted whether or not a pitch would be swung with 0.86 AUC, making our model highly accurate.

We learned that many of the features we created actually did provide the model with quality predictors for whether or not a pitch would be swung at. When using XGBoost, we took a deeper look at the feature importance and quantified how each feature used in the model was contributing towards predicting whether or not the pitch was swung at. As we can see in these results, some of the most important features for predicting whether or not a pitch was swung at when using XGBoost were our engineered features: Extreme, a calculation of how rare a pitch in the dataset was based on its residual from the average pitch speed and location, carried significant predictive weight. We also helped our model's predictive accuracy comparing the relative speed and zone speed as this variable shows a pitch's change in speed as it travels from the pitcher's mound to the plate. As we look forward into how to improve our model, we know that we have a general idea on what features we want to continue looking at and what features might be useless in our goals to improve the accuracy of our predictions.

It was also surprising to see that some features have so little impact in predicting whether or not a pitch was swung at. For example, in all of our models, it seems that the type of pitch is not significantly helping the model accurately predict whether or not the batter will swing. This is a shocking realization as different pitch types often play a significant role in how effectively a batter can make contact with the ball. We also had no success with our feature accounting for what the previous pitch type was. This was not expected, as the pitch progression throughout a hitter's at bat is commonly analyzed and seems to have real life importance when it comes to a batter's decision making.

Although our dataset was imbalanced, we think that using the AUC measurement to analyze our XGBoost model's performance was a productive alternative. However, this is worth considering when interpreting how practical this model is, as ideally, the accuracy measurement would be ideal for determining how good our model is at making swing predictions.

As we continue to improve the model's accuracy and capabilities, we would like to see it applied to sabermetrics in college baseball in few different ways. Ideally, we would like to give this model the ability to train on in-game live data so that batting coaches could make strategic decisions on how their player's should approach their at bats. Pitching coaches could also use our findings to determine what pitches should be thrown throughout a baseball game to maximize a pitcher's chances at a successful outing.

With our best model being classified as highly accurate according to its AUC score of 0.86, we think that this model is highly capable and can be relied on by college baseball teams for making informative predictions and building advanced scouting reports. This measurement also shows that this model can be built on and improved. If we continue or research, we can make this model highly practical.

VII. Conclusion

Moving forward, we think it is important to continue on with the process of finding the right combination of features in order to improve our model's prediction accuracy. Currently making predictions with a highly accurate 0.86 AUC score, we are not satisfied with the results. We had a lot of success when it came to feature engineering, as our intuition led us to creating new variables with data manipulation of other features in the dataset. Given how important some of our own engineered features ended up being for prediction purposes, we think that there is a lot of room for improvement.

We also think that we can do a better job at including some features that we may have overlooked. With the ultimate goal of this project being to serve the Penn State baseball team with a useful tool that they can use to improve their in-game analysis, we will need to figure out a way to make this model work in real time. We would like to get to a point where we can use data to predict swinging outcomes ahead of future competitions. An accurate model can be great for Penn State Baseball scouting reports; we can train the model on specific opponents to predict what pitches they are most likely inclined to swing at. We can also attempt to build reports simulating how well we think a given batter or pitcher will perform based on the data collected throughout the baseball seasons. A model that is autonomous with newly updated data can prove to be very useful to coaches and players looking to prepare for each game ahead of them. In order to do this, we will need to keep improving how we manage our data and how it is organized.

As we stand right now, we feel like this model has not reached its full potential for usefulness and accuracy. As we continue to make improvements on how our data is organized as well as how we decide what techniques and methods to experiment with, we can improve the accuracy and practicality of this model greatly.

References

[1] K. Koseler and M. Stephan, "Machine learning applications in baseball: A systematic literature review," in Applied Artifical Intelligence, 2018, pp. 745–763. DOI: 10.1080/08839514.2018.1442991.

[2] Ghosh, D., Weinstein, M. J. (n.d.). Swing or Take: Decision Making under Uncertainty Based on Visual Information. UC Berkeley.

[3] Google. (n.d.). Figure 4. Tp vs. Fp rate at different classification thresholds. Classification: ROC Curve and AUC. photograph. Retrieved from https://developers.google.com/machine-learning/crash-course/images/ROCCurve.svg.

[4] Hamilton M., Hoang P., Layne L., Murray J., Padget D., Stafford C. and Tran H.. Applying Machine Learning Techniques to Baseball Pitch Prediction. DOI: 10.5220/0004763905200527 In Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods (ICPRAM-2014), pages 520-527 ISBN: 978-989-758-018-5

[5] Zhang, T., Lin, W., Vogelmann, A. M., Zhang, M., Xie, S., Qin, Y., Golaz, J. C. (2021). Improving convection trigger functions in deep convective parameterization schemes using machine learning. Journal of Advances in Modeling Earth Systems, 13(5). https://doi.org/10.1029/2020ms002365