MLB Pitch Quality - Final Paper

Ryan Patterson
Virginia Tech
Blacksburg, United States
ryanp18@vt.edu

Jared Joo
Virginia Tech
Blacksburg, United States
jaredjoo@vt.edu

Quinn Duggan
Virginia Tech
Blacksburg, United States
dqduggan@vt.edu

ABSTRACT

In Major League Baseball (MLB), data is important when evaluating athletes and team performance. Unfortunately, the current state of the art in baseball analytics is mostly unknown due to the inaccessibility of team internal models. As a result, the goal is to create a model for pitch quality assessment that is both accurate and useful. Using data from PyBaseball, we selected 10 key features and applied feature engineering and machine learning techniques. XgBoost was ultimately chosen as the best model for our purposes. The initial attempt to convert runs to the grading scale failed due to the discovery that specific pitch types reduced the number of runs scored more effectively. The solution was to create a unique grading scale for each pitch type: fastballs, breaking balls, and off-speed pitches. Overall, this project successfully created a model that quantifies pitch quality for past and future MLB seasons, with potential to open doors for more open-sourced pitch quality models.

1 INTRODUCTION

In Major League Baseball (MLB), there is a large emphasis on data because of its importance in evaluating performance in athletes and overall benchmarks. Currently, this data is tracked through the Hawkeye system, which is a collection of high frame rate cameras placed around the stadium to track the ball, the players, and any action that occurs during a play. Using the data captured by the Hawkeye system, we want to be able to create a model to predict the quality of any given pitch thrown in an MLB game.

Though this data is posted online after baseball games, the current state of the art is unknown because MLB teams are unwilling to share their internal models as giving other teams insight into their player evaluation process puts them at a disadvantage. Because of this, current models aren't widely available to the public. This presents a roadblock in the advancement of the sport. This roadblock is our motive: if we can create a model for pitch quality that turns out to be successful and useful, we plan to make the model open-source to help further future research into the topic. Our hope is that this will lead to an elevation in the level of the play in the sport.

Our main challenge will be identifying key variables to predict the quality of a given pitch, and in turn, we hope to produce a model that will accurately and fairly rate the quality of any pitch, past or present. By doing this, we can advance the public's understanding of the game of baseball, as well as what makes each pitcher good or bad.

2 RELATED WORK

As mentioned in the introduction, the current bleeding edge state of the art in pitch evaluation models isn't very clear. In this section, several models in the current literature are discussed, but they are not as in depth or state of the art as the ones used behind closed doors in different MLB teams. This is not likely to change, as individual MLB teams do not want to offer any competitive advantage by allowing their models to be used by opponent teams.

First, we will look at a model by Driveline, who lets the public interact with their model. Driveline is a private player development trainer who specializes in improving pitchers. Their model is called Stuff+, and the main pitch characteristics that they look at are pitch velocity, horizontal and vertical break, arm angle, and release extension [2]. Another publicly available measure of pitch quality comes from an article on Fangraphs, a highly respected baseball research website. In the article, they discuss their own quality of pitch metric, where they disclose some of the variables that they took into account. These variables include pitch velocity, vertical and horizontal movement, rise, lateness of break, and pitch movement [6]. We see some common variables between Driveline and Fangraphs in pitch velocity and movement, which leads us to believe that these are commonly accepted as valuable predictors. However, the Fangraphs article does not include much information of what the rise and lateness of break variables actually are, which limits our ability to use them unless we find a solid definition on them.

Next, a paper released by members of Simon Fraser University outlines how they attempted to define a quality pitch. They did it through predicting the probability of certain events occurring as a result of the pitch [5], which was a new approach from the other papers and articles. They also disclosed the variables that they included, which were pitch count, pitch location, pitch velocity, count for the at-bat, and pitch type [5]. While pitch velocity is consistent among all of the papers so far, pitch movement is not included in this paper. This can likely be attributed to this paper coming out at the advent of the advanced pitch tracking era, so they used pitch type as a proxy variable for pitch movement. The next article is the most recent, where the data scientist from Pitcher List released their own pitch quality metric called PLV. Like the paper from Simon Fraser, they took a probabilistic modeling approach, using multiple model combined together in a manually defined decision tree [3]. By doing this, they were able to determine the probability that certain events would occur, as well as how they impact the perceived quality of a pitch. Combining their approach with Driveline's approach will make for a good starting point.

A final related article by Jake Sauberman published in Towards Data Science attempts to measure how "deceptive" a pitch is, as opposed to just modeling pitch quality in general terms. The article defines deception as a combination of unpredictability, indistinguishability, and unexpectedness [4]. Unpredictability is defined as the batter not being able to guess what pitch type is coming given the pitch count [4]. For indistinguishability, it deals with pitch tunneling, which is having two pitches of different types appearing to be indistinguishable from each other for as long as possible

[4]. Unexpectedness is defined as the difference between the pitch movement that the batter expects to occur from the release point the pitcher has, and the actual pitch movement [4]. With all of these factors combined, we would be able to measure how deceptive any individual pitch is, which could boost or lower the quality of a pitch. The final article that we looked into was also focused on trying to identify additional variables to include in the model, which once again came from Fangraphs. In the article, the focus is on the metric Vertical Approach Angle (VAA), or the angle that the pitch makes with the ground as it crosses the plate [1]. By observing this angle, we can determine whether or not a pitcher throws a pitch flatter or steeper than other pitchers[1], which lends itself to be somewhat descriptive for deception, which could tie together very well with the previous article that we looked at.

The main limitations for existing models is that they do not share their methodology behind creating the model, as well as most models not being public at all. Without knowing the methodology that people used to create their model, there is no way for the public to vet the quality of the model on their own, which gives people two options. They can either blindly trust the pitch quality numbers they see, or they can decline to use them, neither of which are ideal options. By publishing the methodology that we use, the public will be able to see exactly how we arrived to the conclusions we did, and whether or not they agree with our methodology will allow for them to decide on their own whether or not to use our model.

3 PROPOSED APPROACH

3.1 Problem Definition

For our project, we have taken a regression approach for measuring the quality of a given pitch. For this, we regressed upon the delta_run_exp variable in the dataset, which measures the change in run value as the state within the game changes. The reason we have this variable as our target is because it is free from any potential bias that other measures could have had, as it is simply measuring the change in value as the state changes. With other potential targets, such as whiff percentage, there would be implicit bias within the target, as not all pitches are thrown with the intent of eliciting a swing and miss, which would inherently make these pitches seem lower quality, even if they are doing what they were designed to do. For our evaluation metric, we used mean squared error (MSE), as our target variable is very small, and we wanted the model to punish very large residuals.

MSE punishes large residuals more than RMSE or MAE do since the variable is centered around 0, which provides us with the most accurate model possible.

3.2 Data Pre-processing & Feature Engineering

With regards to data pre-processing, there was not much that we needed to ctually do. Since this data is the same data that MLB teams are using, they have a massive financial and competitive incentive to maintain the quality of the data. This means that there are very few incorrect or incomplete observations within our dataset. Since our dataset is currently 1.8 million pitches, we safely dropped any rows that were incomplete or incorrect without much fear of losing valuable information.

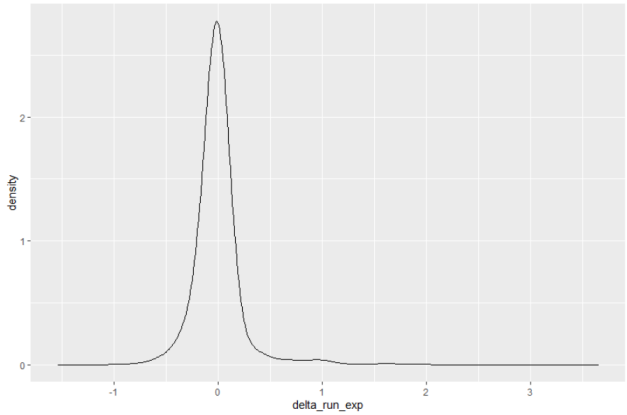

Figure 1: Target Variable Density Plot

With our target now defined, we then turned to the features that we have decided to include within our model. From the original data set, we included release speed, horizontal and vertical pitch movement, pitch spin rate, pitch spin axis, and release extension. We also engineered some features such as arm angle, velocity, and pitch movement differentials off of the primary fastball. For arm angle, we utilized the release position x and z variables within the dataset and basic trigonometry to capture an approximation of the angle of the arm at release. While the release position x variable does depend on the horizontal location that the pitcher stands on the mound, it is a good enough proxy for the horizontal position of the arm that it should not impact the overall angle too much. For measuring the difference of velocity and movement compared to the primary fastball, we looked at each outing that a pitcher has and identified which form of fastball was their primary one by seeing which one they threw the most. Once we identified the primary fastball, we found the average velocity and movement of that pitch during that outing. Once we obtained that information, we joined it back into the original dataset, and for each relevant column, we took the difference between the actual pitch and the average primary fastball metric. These variables were identified as key predictors within our literature review, which gave us confidence that the model accounted for most of the key variables that can predict pitch quality.

For our model of choice, we chose xgboost. We chose it because other public models have stated that it is the model type that they have decided to use, and because in our testing it outperformed the Random Forest model we tried. Ultimately, xboost is a solid model choice as it handles large datasets and non-linear variable interactions very well. These factors are very important for us, as we have a very large amount of observations, and there are no easy and immediate relationships within our selected variables. For our model training, we used a 75/25 train/test split, leaving our training set at around 1.35 million pitches and the testing set at 450 thousand pitches. This gave us a sufficient amount of data to properly train and evaluate the model. We also used a 5 fold cross validation while training the model, as well as a random grid search for hyper-parameter optimization. This allowed us to ensure that the hyper-parameters that we selected truly optimized the model,

but also constrained run time and allowed us to explore many options. Another benefit to xgboost is that it allowed us to see the relative feature importance of each included variable within the model, which showed us what the biggest drivers of pitch quality are.

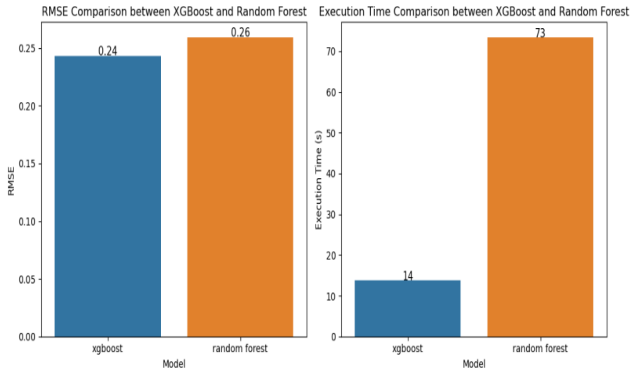

Figure 2: RMSE and Training Time Comparison for Random Forest and XGBoost Models

Looking at these results, we can see that while the xgboost model had a slight edge over the Random Forest model in RMSE, the big thing that sets the models apart was the execution time for training each model. While testing, we originally used the default number of decision trees for the Random Forest from the RandomForestRegressor SciKit library, however, the execution time took too long that it terminated our kernel. We attempted to improve the performance of the Random Forest model by implementing Principle Component Analysis (PCA's) which reduces the dimensionality in hope to improve performance, however, doing so continued terminating our kernel prematurely. In order to get the execution time for the random forest down to a reasonable time without having our kernel time out, we had to reduce the number of decision trees drastically to 5. While with xgboost we were able to train a considerable amount of trees each training, we were only able to train 10 trees in approximately 5 times as much time with Random Forest. Considering our intentions to tune the hyper-parameters of these models, this made it very hard for us to choose the Random Forest model, as the tuning process would have simply took too long.

3.3 Pipeline Diagram

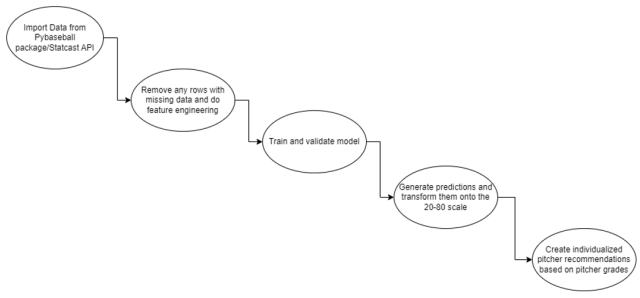

Figure 3: Pipeline Diagram

Our project pipeline is relatively simple, as there is not much that we need to do in order to collect and clean the data. The most time intensive steps of our pipeline came in the form of model creation, as well as visualizing the output of our model. Since the grades we produce for each pitcher are the main deliverable of our model, we have to take care to ensure that the visuals we produce for them are accurate and relevant.

4 EXPERIMENTAL EVALUATION

A key aspect for our project is that we wanted the variables included within the model to be something that pitchers can reliably control with consistency. This is why pitch location was ultimately not selected within our model, as the average MLB pitcher misses their intended location by approximately a foot. To us, this indicated that this was not something that a pitcher could easily control, so we decided to not use it. However, when we looked at pitch movement, we saw that pitchers were able to more consistently replicate their typical movement patterns.

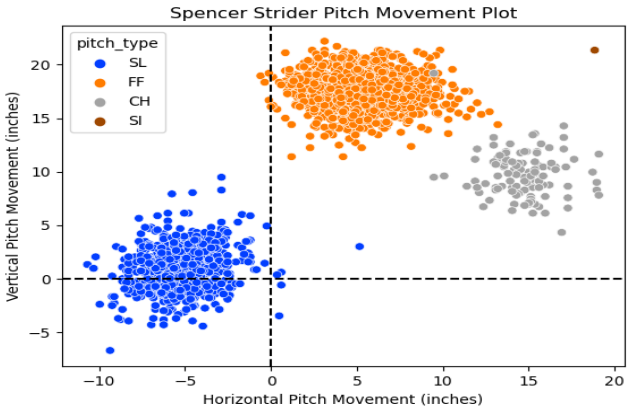

Figure 4: Spencer Strider Pitch Movement Plot

As we can see above, Spencer Strider is able to consistently separate his pitch types from each other, with there being no major outliers within the dataset. In a similar vein, pitch velocity and spin rate are consistent within each pitch type for a pitcher, as they are the main trainable features that pitchers have been working on in the past few years as data tracking has increased.

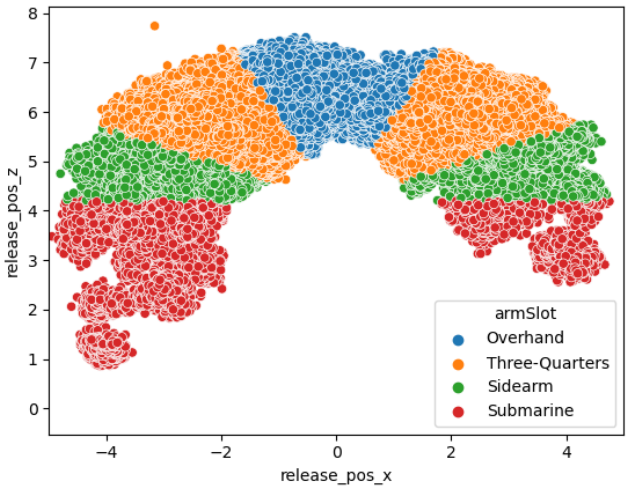

Figure 5: Arm Angle Plot

Looking at the arm angle feature that we calculated earlier, we can see that there is an even distribution of arm angles for both left and right handed pitchers. This allows us to have confidence that during the training of the model, there is no imbalance between any of the arm angles, which will ensure that the model does not accidentally overvalue some of the arm angles due to a lack of observations.

As of this paper, we have completed our cross validation and random grid search hyper-parameter tuning for our xgboost model. The following figure shows the feature importance from our previously untuned xgboost model.

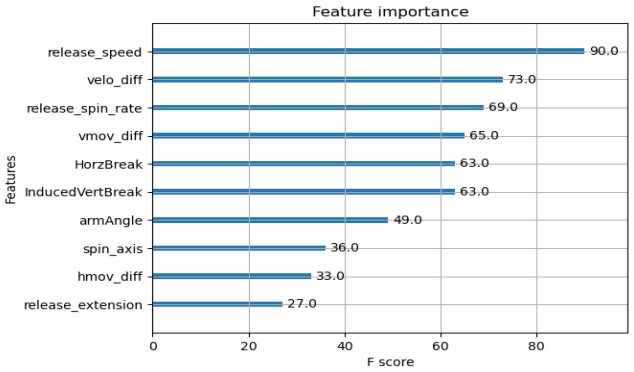

Figure 6: Untuned Xgboost Feature Importance Plot

The F score metric simply measures the number of times a certain variable is used as the break point within the underlying decision tree. While this is a very simple way of measuring the importance of each feature, it does tell us a lot about which features the model has decided that do a good job at segmenting the data. Above all, we see that pitch velocity is the biggest driver of pitch quality, which is not shocking, as the harder a pitch is thrown, the less time a batter has to react to the pitch. The rest of the features can be broken down into two different groups, with the next 5 features being all worth roughly the same amount, with the following 4 being worth a bit less. What this tells us is that the differential variables are generally valuable, but the key features that pitchers can control are velocity and vertical pitch movement. We were also able to calculate feature importance for the Random Forest model that we had tested, which revealed the following:

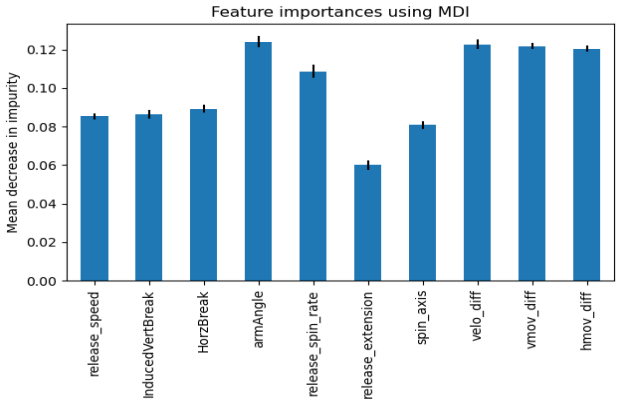

Figure 7: Untuned Random Forest Feature Importance Plot

While the Random Forest is not the model we chose, we can still gather valuable insights by learning which variables it decided were most important. As we can see, the RF model viewed the primary predictor of pitch quality to be arm angle, followed by the the differential variables that we calculated. This runs against what the xgboost model showed, which indicates to us that each model is capturing different variable interactions. They both end up at roughly equivalent RMSE values, which means that including more trees in the xgboost model would potentially be a good idea, as it would help capture more interactions in the data.

After tuning our xgboost model, we revisited the feature importance for it, to see if there were any noticeable changes between the relative importance of each feature.

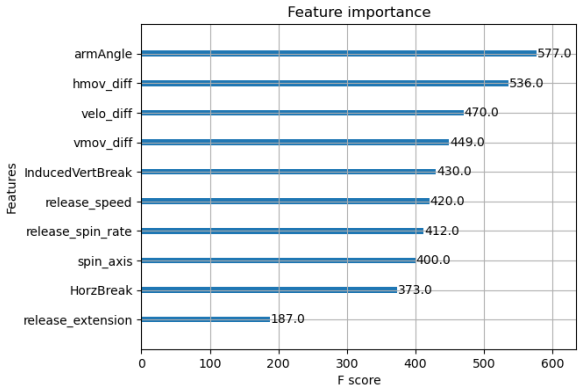

Figure 8: Tuned Xgboost Feature Importance Plot

As we can see, like the Random Forest feature importance plot, arm angle and the differential variables are the most important features that the model identified. With that said, pitch velocity and pitch movement are not lagging far behind, so the initial importance from the untuned xgboost model still holds some validity. Overall, this finalized model appears to cover both of the relationships that the original xgboost and Random Forest models initially uncovered, which is a byproduct of us training more trees in the model, as well as making each individual tree have more layers. The only real outlier that we see here is release extension, which while it is clearly the lowest performing feature among all of the models, we are going to keep it in the model as it was something identified throughout our literature review as an important feature. While it does not provide a massive amount of value to the model, it can provide valuable information in niche scenarios which is valuable to us. Another key thing to note is that while we did see some very slight gains in RMSE after tuning the model, it was not a significant difference. We believe this is because there is underlying variability within our target variable, which we have approached with our model.

In baseball, players are measured on the 20-80 scale, which is a scaled normal distribution, with 50 being the mean, and every 10 points in either direction represents a standard deviation away from the mean. We will be transforming our data onto the 20-80 scale, as it is a scale that people within baseball understand. However, we cannot just take the raw predictions and convert them, as different pitch groups have different predicted values, with breaking balls being the most effective pitches, and off speed pitches being the least effective. If we did a straight conversion, we would have every top grade be a breaking ball, and every bottom grade being an off speed. To circumvent this issue, we will be grading each pitch relative to the other pitches within its own group. For example, all sliders and curveballs will be graded relative to each other, and no other pitch type, as it would skew the distribution. This will be our ultimate evaluation tool for a given pitcher.

Using these pitch grades, we are able to create slices of the variables within the dataset and see how the grades change with the variables changing.

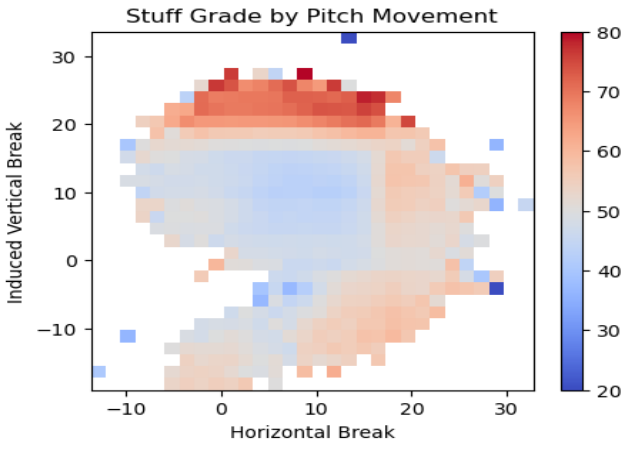

Figure 9: RHP Fastball Stuff Plot

In figure 9, we subset the data to just fastballs and sinkers thrown by right handed pitchers for the purposes of this plot. We then plot the horizontal and vertical break of each pitch, and create bins for each combination. We can then find the average grade of the pitch thrown within the bin, and it creates the color mapping that we see. The more red a bin is, the higher quality the model believes it is. Conversely, the more blue it is, the worse the model thinks it is. As we can see from the plot, pitches thrown with generally equal parts horizontal and vertical break have lower grades, while pitches thrown with a lot of vertical movement see the highest grades. Pitches with a lot of horizontal movement all grade out well as well, which indicates to us that having non-extreme pitch movement leads to lower quality. While this plot is all fastballs thrown by right handed pitchers, we could further filter down the dataset to be between two velocity ranges, where higher velocity fastballs will likely have less blue in the plot overall, as the velocity will likely give the pitchers a bit more leeway with the movement required for the pitch to be effective.

Our next iteration of our visualizations takes a specified pitcher and changes the underlying data being used for the plot to be more relevant to the specified pitcher, by looking at pitches that are thrown at a similar velocity to the pitches the pitcher throws.

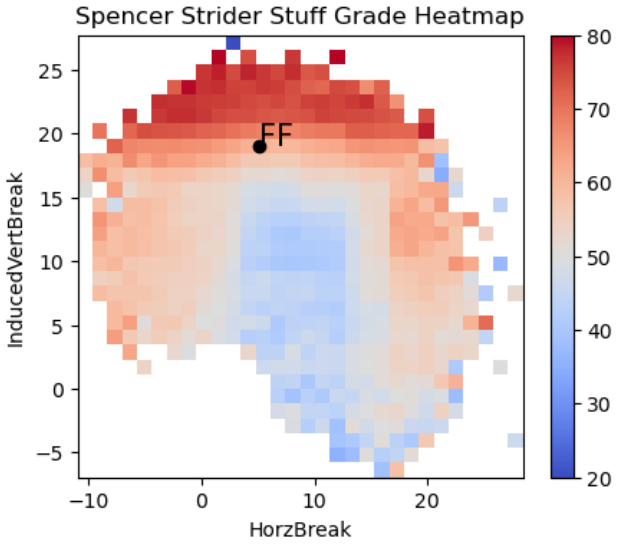

Figure 10: Spencer Strider Fastball Stuff Plot

Looking at figure 10, we are once again looking at fastballs thrown by right handed pitchers, but this time, we're specifically looking at fastballs thrown around the velocity that Spencer Strider throws his at. Since he throws his fastball at around 98-100 MPH, we see that there is a significantly increased amount of red on the plot, signifying better pitches. We also are able to plot the movement profile that Spencer Strider has himself, so you can easily see where he stacks up relative to the plot. As we can see, he is firmly within the 65-70 grade color of the plot, so we would not recommend that he changes anything about his fastball, as it is already an elite pitch.

Our final iteration of our visualizations is built upon our previous iteration, where we create a block of heatmaps for coaches and

players to look at. Our thinking with this was instead of having them run the individual plot multiple times, simplify the work that they have to do and present all of the relevant information to them at one time.

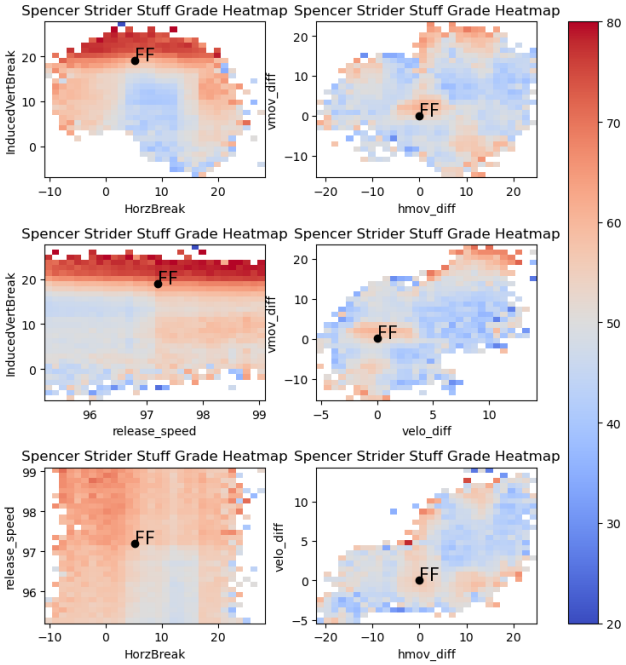

Figure 11: Spencer Strider Fastball Heatmaps

Once again, we are looking at Spencer Strider's fastball. The left column of plots is looking at raw velocity and pitch movement numbers, while the right column is looking at the differential variables that we defined. For fastballs, the primary plots to look at are the left column, as a pitcher with only one fastball will have 0 for all of their differential variables, which does not give any information to the coach or player. The differential variables are much more useful when it comes to the off speed and breaking ball plots for a given player. However, with this plot, a coach and player are able to instantly see how their pitches line up relative to the rest of MLB, which can minimize the amount of time they spend deciding what changes to potentially make, and maximizing the amount of time that they can spend actually changing the pitch. Ultimately, these plots allow for accessibility to the grades we give each pitcher, as well as providing direction and ideas for ways to improve their pitches.

5 FUTURE WORK

As of now, we have finalized our model with regards to hyperparameter tuning. From here on out, we will be focused on the new MLB season's data, as well as finalizing our visualizations. We will once again be using the Pybaseball package to scrap pitch-by-pitch data for our project, which will allow us to access 2023 data. Visualizations are also important in order to easily convey the data to the audience with the limited time we have.

For Milestone 2, we will have pitch quality scores determined for each pitcher on their 2020-2022 data and their 2023 data. From this, we should be able to determine what specific pitchers need to work on to improve their pitches. From this step onward, we will be tracking a list of chosen pitchers throughout the 2023 MLB season (which will have been running for 2-3 weeks at this point), determining if any changes in their pitching strategy align with what we prescribed for them.

For our final presentation on May 2, we will summarize our work by creating a website and poster. Our website will be static and built using React and deployed through Netlify. We aim for the website to be user-friendly and informative, showing our findings for both feature engineering and applications of machine learning. Our resulting dataframe including the player's features along with their stuff grade metric will be displayed as a table on the website and will be interactive, so that the user can look up an MLB player's name and quickly obtain their quantified grading metric. We also plan on showing how our predictions in the 2023 MLB seasons stacked up against what actually happened. By this point, the MLB season will have been running for a little more than 4 weeks. A sign of success for our model would be pitchers we tracked changing their pitches based on our recommendations. If successful, we will discuss how this model compares to other existing models in the industry, any challenges we faced during our semester-long project, and future works that can be associated with our project.

6 CONCLUSION

Due to MLB teams utilizing the same data as us, we were able to start working on our project very early on due to the data already being cleaned for us, as well as there being packages that make the data very easily accessible. After researching numerous types of models, we went with the xgboost model using a 75/25 train test split which is equivalent to 1.35 million training pitches and 450 thousand testing pitches. At this point, we have tuned the model's hyper-parameters, so our focus from here on out will be on the visuals that we create. Looking at figure 9, we noticed from our visualization that right handed pitcher's fastballs thrown with generally equal parts horizontal and vertical breaks produced lower grades while those with a lot of vertical movement produced higher grades, with fastballs being thrown with more horizontal movement also graded out as plus pitches. These plots are something that we will continue to use to explore the model's outputs and what makes each pitch grade out as good or bad, and will be our primary driver for evaluating the quality of a pitcher.

REFERENCES

[1] Alex Chamberlain. 2022. *A Visualized Primer on Vertical Approach Angle (VAA).* https://blogs.fangraphs.com/a-visualized-primer-on-vertical-approach-angle-vaa/

[2] Chris Langin. 2022. *Pitch Design: What Is Stuff+? Quantifying Pitches with Pitch Models.* https://www.drivelinebaseball.com/2021/12/what-is-stuff-quantifying-pitches-with-pitch-models/

[3] Nick Pollack. 2023. *What Is PLV? – An Introduction To Pitch Level Value And Its Applications.* https://www.pitcherlist.com/what-is-plv-an-introduction-to-pitch-level-value-and-its-applications/

[4] Jake Sauberman. 2020. *Quantifying Pitcher Deception.* https://towardsdatascience.com/quantifying-pitcher-deception-7fb2288661c8

[5] Philippa Swartz, Mike Grosskopf, Derek Bingham, and Tim Swartz. 2016. *The Quality of Pitches in Major League Baseball.* https://www.sfu.ca/~tswartz/papers/pitching.pdf

[6] Jason Wilson. 2017. *Measuring the Quality of a Pitch.* https://tht.fangraphs.com/measuring-the-quality-of-a-pitch