**Predicting Strikeout Percentage Among MLB Relief Pitchers Using Statcast Pitch Data**



*A batter for the Texas Rangers strikes out swinging — Thearon W. Henderson/Getty Images*

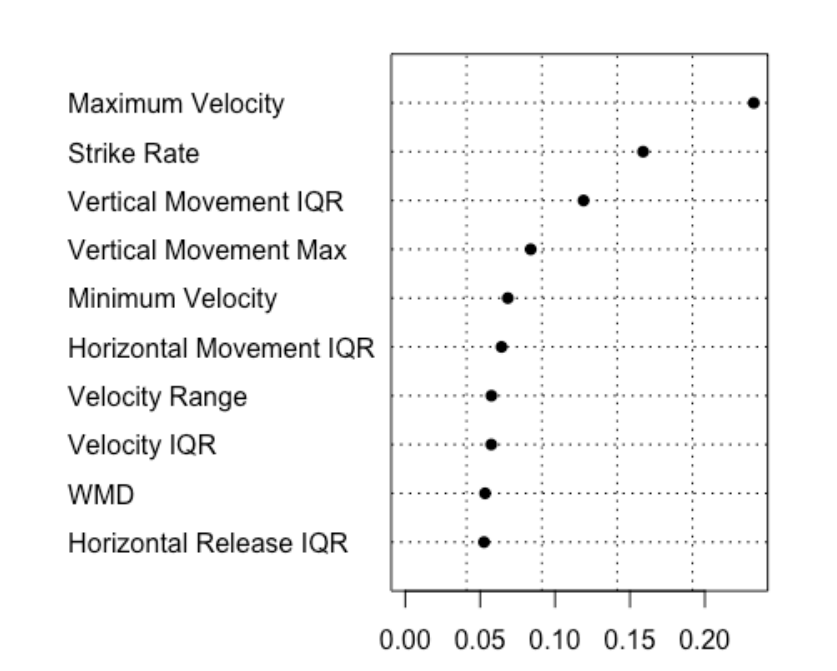
The strikeout is perhaps the most exciting feat that a pitcher in baseball accomplishes regularly. The swing and the miss. The perfect framing. Only three mistakes and you’re finished for the inning. There’s something about it that consistently grabs the attention of fans, analysts, and writers alike. Nolan Ryan’s seemingly unbreakable record of 5,714 strikeouts is still revered as the definition of complete and utter pitching dominance.

Yet, an issue with the strikeout is that it is greatly affected by playtime as a cumulative statistic. Simply getting more starts or appearances or having a long career will net you more strikeouts. Indeed, Ryan’s record can be mostly attributed to his 27 seasons playing baseball, the most for any player ever. In order to solve this problem, we have another statistic that levels the playing field: strikeout percentage.

Strikeout percentage is defined as the percentage of strikeouts that a pitcher earns for every batter they face in a season. As a ratio statistic, pitchers with any amount of games or plate appearances can have a value for strikeout percentage, which is especially useful if the pitcher was unable to play for the entire season due to injury.

In this article, I will be predicting strikeout percentages for relief pitchers. The statistic benefits relief pitchers in particular as their amount of innings pitched is much less consistent than with starting pitchers. They do not follow a five game rotation and may appear in less games due to their performance or role. The part of the game that the relief pitcher appears in has a large impact on this: closers will appear in every game for the away team, whereas firemen and setup relievers will have less as there are more of them, and home team closers may not play every day if no bottom of the ninth is played.

In order to make my predictions, I will focus on three different types of pitching statistics: Statcast pitch data, situational data such as leverage indexes and platoon advantage, and number of pitch types via pitch clustering. I will be using the XGBoost model in the R programming language for this. My research is primarily inspired by the paper [“Predicting Major League Baseball Strikeout Rates from Differences in Velocity and Movement Among Player Pitch Types”](https://global-uploads.webflow.com/5f1af76ed86d6771ad48324b/5f6d38971aa75c2f6af77911_Predicting-Major-League-Baseball-Strikeout-Rates-Update.pdf) written by Eric Martin for the 2019 MIT Sloan Sports Analytics Conference. He states in his conclusion that future analysis could be to utilize his methodology for relievers instead of starters. One key difference between this project and his research is that I am placing less focus on pitch clustering, and all of my analysis will be performed per pitcher as opposed to per pitch type. This is due to a number of constraints that I faced, such as with computational complexity and the time that I had to complete the project. For starting pitchers, Martin found that maximum velocity, strike percentage, and vertical movement were the most important predictors for strikeout percentage.



The feature importance from a Random Forest model that Martin uses in his analysis of predicting strikeout percentage on pitch data and clusters

The first step to building the prediction model is to organize all of the data. The primary source for this data is a Statcast database, which has data for every event in every MLB game, with specific information about the velocity, movement, location, spin rate, etc. related to every pitch. The creator of the baseballr library for R, Bill Petti, [has a guide](https://billpetti.github.io/2021-04-02-build-statcast-database-rstats-version-3.0/) on how to build this database. The code he provides scrapes BaseballSavant — the online repository of the Statcast data — for a given range of years and appends it to a PostgreSQL database.



A snapshot of the data from the Statcast database opened up in HeidiSQL, a popular free SQL IDE.

The database will be mostly between the years 2015 and 2021, which is the period that Statcast data has been available. The year 2020 is omitted as the season was greatly impact by the COVID-19 pandemic, with the shortened season and other externalities caused by the pandemic potentially impacting the statistics for the season. Looking back, this was not a necessary choice, but it does allow for the 2021 data to be easily split into the testing data in the model. For these six years, the PostgreSQL database I created contained data for a whopping 4.4 million pitches and took up around six gigabytes of storage on my computer. In order to access this data I had to learn some SQL, which I used to aggregate these pitches to be by pitcher by year. The aggregated stats I created included maximums, minimums, averages, ranges, and interquartile ranges for pitch velocity, vertical and horizontal movement, spin rate, and more. The rest of the work with the data was done in R.

My goal with this data is to filter it down to just be relief pitchers, who I define as pitchers with less than 1,000 pitches per season and less than 45 pitches per game. Pitchers with more than 1,000 pitches are usually starting pitchers. The latter figure accounts for starting pitchers who did not play the amount of games necessary to be able to throw that many pitches. However, one problem remains, which are position players pitching. If a team is losing by a large amount of runs and has ran out of relief pitchers, a position player such as a catcher will pitch instead. Because the appearance is so short, it would fall under the relief pitcher category. I dealt with this by using Sean Lahman’s baseball database of all MLB players to select and merge only players with the pitcher position listed.



Dodgers catcher Drew Butera strikes out Diamondbacks utility player Martín Prado in a rare position player pitching appearance

Merging dataframes was a key aspect of preparing the data for the model. Different sources have their own methods for identifying players (the main ones being MLBAM ID’s, Baseball-Reference ID’s, and FanGraphs ID’s), so I needed a dataset that contained all of these in order to link my data together. This can be found in the Chadwick Bureau that contains information for every individual in history to play baseball at least at the collegiate level. However, the baseballr function to access this data was not working as a result of them splitting up their data from one very large file to 16 smaller files. The API was not changed to reflect this, meaning that I had to combine each of these files and then filter them down to just MLB players. I was then able to merge my data.

While the pitch data from Statcast contained basically all of the information I needed for my analysis, it did not include some other necessary pitching statistics, such as strikeout percentage. I created a method to find this statistic per pitcher per year using information from the “events” column of the database. I used data for the outcomes of pitches to do the same for strike percentage, which is related to strikeout percentage. If you throw more strikes, you are probably going to get more strikeouts. I then created similar methods to calculate the data for games played and pitches thrown.

Other pitching statistics were brought in to account for the differences between starters and relievers. Oftentimes relief pitchers are brought into the game if the matchup against the batter is favorable for the pitcher, which is primarily determined by handedness. This is called the platoon advantage, and same-handedness is what favors the pitcher, such as a righty on righty matchup. I created a statistic called “First Batter Platoon Advantage” that is the percentage of the times that the first batter a relief pitcher faces in his appearance has the same handedness as that pitcher. The reasoning for this potentially being related to strikeout percentage is that if the platoon advantage is more often than not with the pitcher, then they should be able to strike out the batter more.

However, the notion of platoon advantage being a necessary factor in pitching is often challenged. It sometimes is derided as being a symptom of overmanaging, and opponents state that the actual stats of the batter should be what is taken most into account. For example, in Game 2 of the 2022 American League Wild Card Series matchup between the Seattle Mariners and the Toronto Blue Jays, the Mariners’ switch-hitting slugger Carlos Santana hit a crucial home run off of Blue Jays’ reliever Tim Mayza to start a comeback from a seven run deficit. The Blue Jays had secured their 8–1 lead in part due to their starter, RHP Kevin Gausman, preventing the Mariners from scoring off of his dominant splitter. However, Gausman was pulled after a poor 6th inning, where he loaded the bases. Yet, he managed to strike out the next batter and got the next to hit a pop-up. Despite him reaching 95 pitches, this would typically be a signal to the manager that the pitcher still has some gas left in him. The deciding factor for Blue Jays manager John Schneider was that the next batter, Santana, could bat left handed, which would give him the platoon advantage. In a perfect world, the lefty Mayza would be able to match Santana and record the final out of the inning, getting the Blue Jays out of a jam and surely securing a victory to force another game.

Batting right-handed, Carlos Santana hits a three-run home run off of LHP Tim Mayza in Game 2 of the 2022 ALWCS between the Seattle Mariners and the Toronto Blue Jays

The Blue Jays do not live in a perfect world. First, Mayza threw a wild pitch that scored one runner. He then gave up the home run to Santana, who was batting right handed, which was his stronger side for that season. Santana being a switch-hitter obviously placed Schneider in a tough situation: let a tired yet confident Gausman stay as he loses his platoon advantage, or bring in a fresh, but less skilled Mayza who also does not have the platoon advantage. Schneider obviously made the wrong choice as Mayza was shaky to start out and, most importantly, the platoon advantage was never a factor that should have gone into Gausman being taken out of the game. The Mariners would go on to win 10–9, the second largest comeback in playoff history, and the largest on the road. Blue Jays fans were outraged at Schneider’s decision to pull Gausman, with [one Redditor saying](https://www.reddit.com/r/Torontobluejays/comments/xz7tz4/postgame_thread_october_8_seattle_mariners/irksoub/):

*You’re Kevin Gausman. You load the bases in a must win post season game. You’re pissed. You’re hungry. You strike the next guy out. Crowd goes f\*\*\*ing nuts. . . . You’re going to get out of this. You’re ready. Then at the corner of your eye you see John approaching the mound. You say to yourself, “Not now John. I’m in the mother f\*\*\*ing zone”.*

*John: “You’re out. I’m playing the matchup. Good game”*

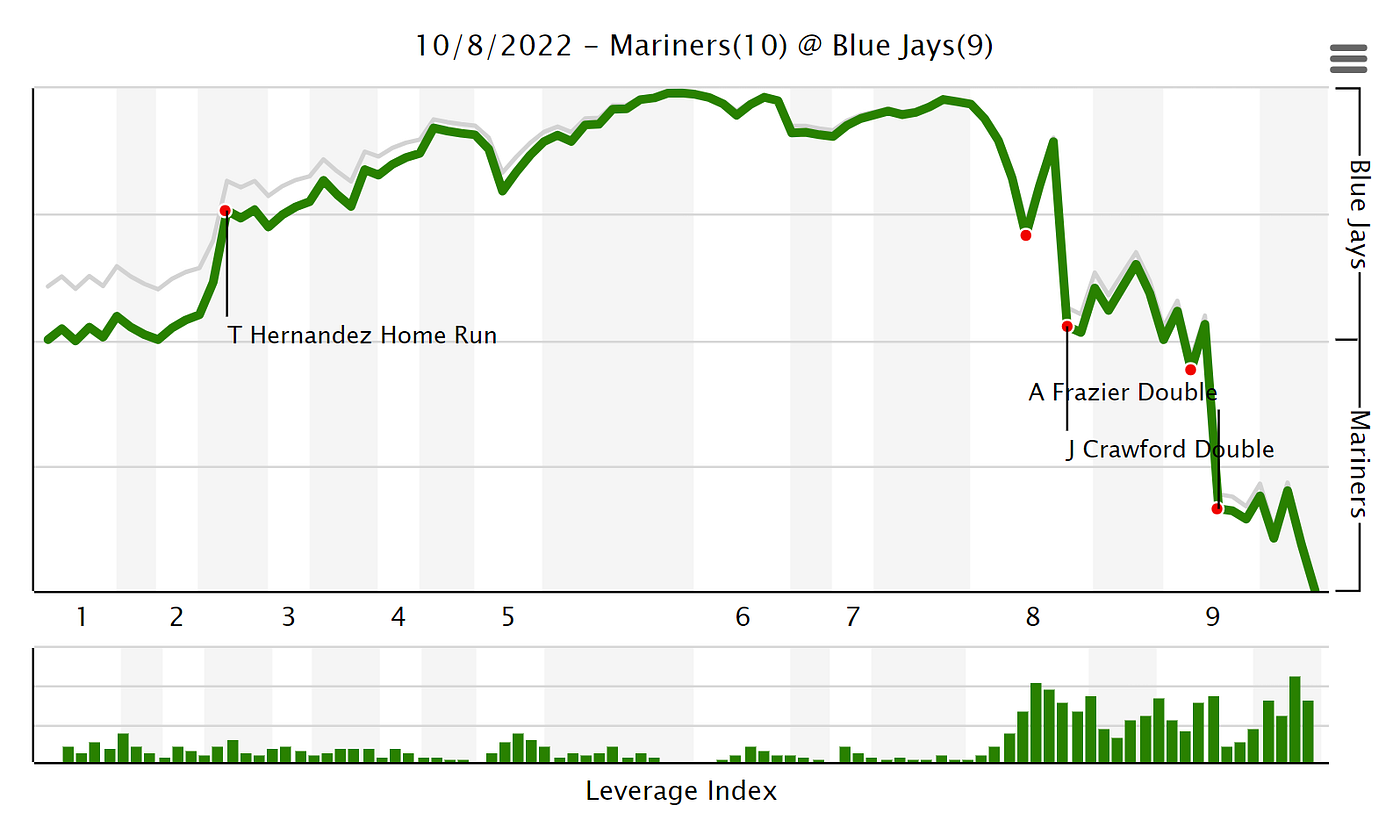
*John taps you on the ass as you walk away deflated.*

*Momentum shifts. And the rest is history.*

*F\*\*\*out of here with Matchups. Biggest bulls\*\*\* of new age baseball. Trust your f\*\*\*ing arms to get out of the f\*\*\*ing inning.*

The vast majority of other Blue Jays fans agreed with that sentiment. I seek to find if whether or not the platoon advantage is a good predictor for strikeout percentage, and whether or not that result matches this sentiment.

There is also a statistic that is more formal in its treatment of context-dependent aspects of the game, known as the leverage index. This is a stat based on win probability, and represents the swing in WPA (win probability added) that could occur as a result of a given event. I selected two out of the four leverage indices that make the most sense for this analysis. The first is the average game leverage (pLI), which is the leverage for every pitch in an appearance. This isn’t as useful for looking at when to bring in a reliever, but still is useful as a stat to measure how good or bad an appearance was. The gmLI for a pitcher’s appearance may be increased if they get themselves into a jam and decreased if they are able to get themselves out of it.



A FanGraphs chart of both the win probability and leverage index for the ALWCS game between the Mariners and the Blue Jays discussed prior

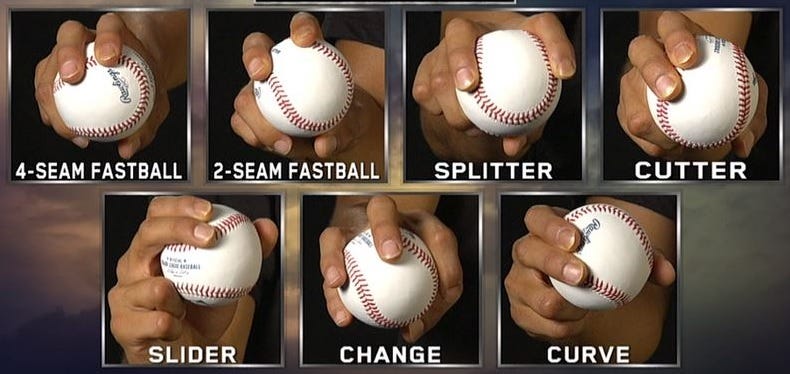
The more interesting stat is the game entrance leverage (gmLI), which is the leverage index for the beginning of each appearance the pitcher has averaged over the whole season. If this index is higher, the reliever entering the game will be facing more trouble and will be under pressure. The two other forms of leverage index, inLI and exLI, are more useful for starting pitchers as opposed to relief pitchers as they track the leverage only at the beginning and ends of innings. Inning leverage may sound appealing for relief pitchers, but it’s most useful for multi-inning appearances, which are rare among relief pitchers. Likewise, exit leverage for one reliever or the starter can be better measured by the game entrance leverage (gmLI) of the next reliever.

A notable example of these leverage indexes could be found on a May 15th, 2022 game between the Mariners and the New York Mets. Going into the 9th inning, the Mariners had a solid 8–5 lead, with starter Robbie Ray keeping it a close game for a six inning appearance until a pair of home runs from Mariners’ sluggers Julio Rodríguez and Cal Raleigh blew the game wide open. In the 8th, manager Scott Servais elected to have his #1 closer Paul Sewald pitch first, while giving the #2 closer Drew Steckenrider a shot at the 9th due to the strong lead the Mariners’ had. While Sewald pitched well, Steckenrider was not on his game that day (or in future days given that he was optioned to AAA only 11 days later), and gave up two earned runs off of four hits in only a third of an inning pitched. A Brandon Nimmo double alone added an increase to the Mets’ win probability of 34%, with Steckenrider’s appearance alone adding 49% to their chances. He was pulled immediately after giving up the second run, which left his pLI at only 1.83, due to him not having to deal with the consequences of his actions. Instead Servais went to a new pickup in the bullpen, the experienced reliever Diego Castillo.

Diego Castillo strikes out two in 0.2 innings against the New York Mets in a very high leverage situation. Skip to the 7 minute mark to see the resolution of the game

The gmLI for Castillo’s appearance was a whopping 6.03 as he entered the game. With two outs left in the bottom of the 9th in a one run game facing the top of the lineup of one of the best teams in baseball, the stakes were extremely high. He first faced the hard hitting Starling Marte, who he promptly struck out swinging with a nasty slider. At this point, his pLI had decreased marginally, but was still facing the meat of the Mets’ lineup. Next was Francisco Lindor, who had been in a bit of a slump leading up to the game but looked dangerous throughout it, having already hit a home run off of Ray. A switch hitter just starting an eventual 5.5 WAR season was not who Castillo wanted to see here, and he was intentionally walked.

Castillo’s pLI immediately jumped up as the bases were now loaded and anyone getting on base could tie the game or win it outright for the Mets. Castillo now faced perhaps their scariest player, Pete Alonso, who was just coming off of a 37 home run season and looked to one-up that in 2022 (which he eventually did, hitting 40). Going with only his slider, Castillo worked the count full while not necessarily having the best control. He went with his slider one more time and Alonso checked his swing just a little too far. Game over, Mariners win. Castillo’s pLI dropped to an 8.30 average while he struck out two batters in an extremely clutch situation, adding to the credence of the two statistics being related. A key feature of his appearance was his slider; pitch types are the final part of the data for this model.



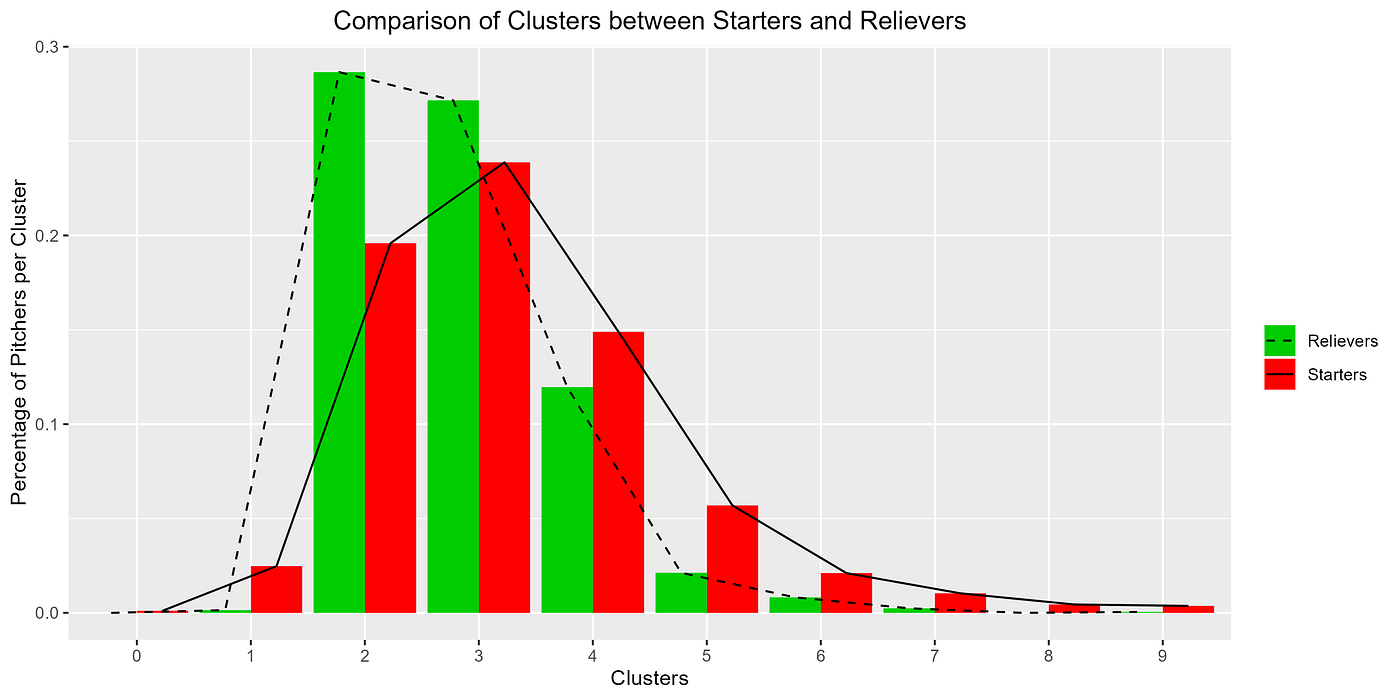
The grips necessary for different pitch types

My goal with pitch clustering was for it to be an extension of my project, utilizing techniques that Martin describes in his paper. Pitches are already clustered by type, such as fastballs, sliders, and others; however, two different types of fastballs for two different pitchers could look exactly the same to the batter despite the different ways the pitchers throw and grip the ball. The goal of pitch clustering is to establish pitch arsenals based on the physical attributes of the pitch itself rather than its name. We can identify clusters using Gaussian Mixture Models to identify probabilistic regions that pitches could inhabit. These use Mahalanobis distances (the distance between an item in a group and the Gaussian ellipse representing the distribution for that group) as well as covariance matrices and other complicated linear algebra to identify which pitch belongs to which cluster.



Different types of Gaussian ellipses (not from my analysis), with the concentric rings being different standard deviations. The Mahalanobis distance for a given point would be the distance between it and one of these rings

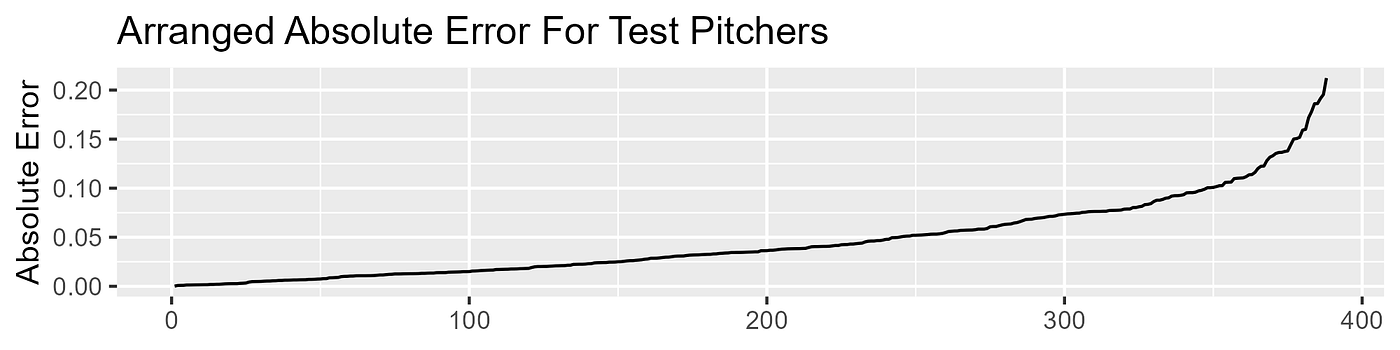
I used the MClust library in R to perform the clustering per pitcher per year. Due to computational as well as time constrains I was unable to do much with these clusters, such as removing outlier clusters, which can radically change the clustering. If a pitcher has four hundred pitches in a season that split into three clusters, but has another six pitches that don’t fit into these that create another two clusters, there would be five clusters instead of three. This is a drastic change in the context of at most nine clusters. With this in mind, we can still observe some conventional logic which holds that relief pitchers throw less pitches than starting pitchers, as they pitch less in general. My clustering showed that they on average pitch one less than starters.



Within my clusters, starting pitchers on average had one more pitch type than relief pitchers. Note the amount of starters with only one or two clusters; this is due to a rather lackluster implementation of the clustering

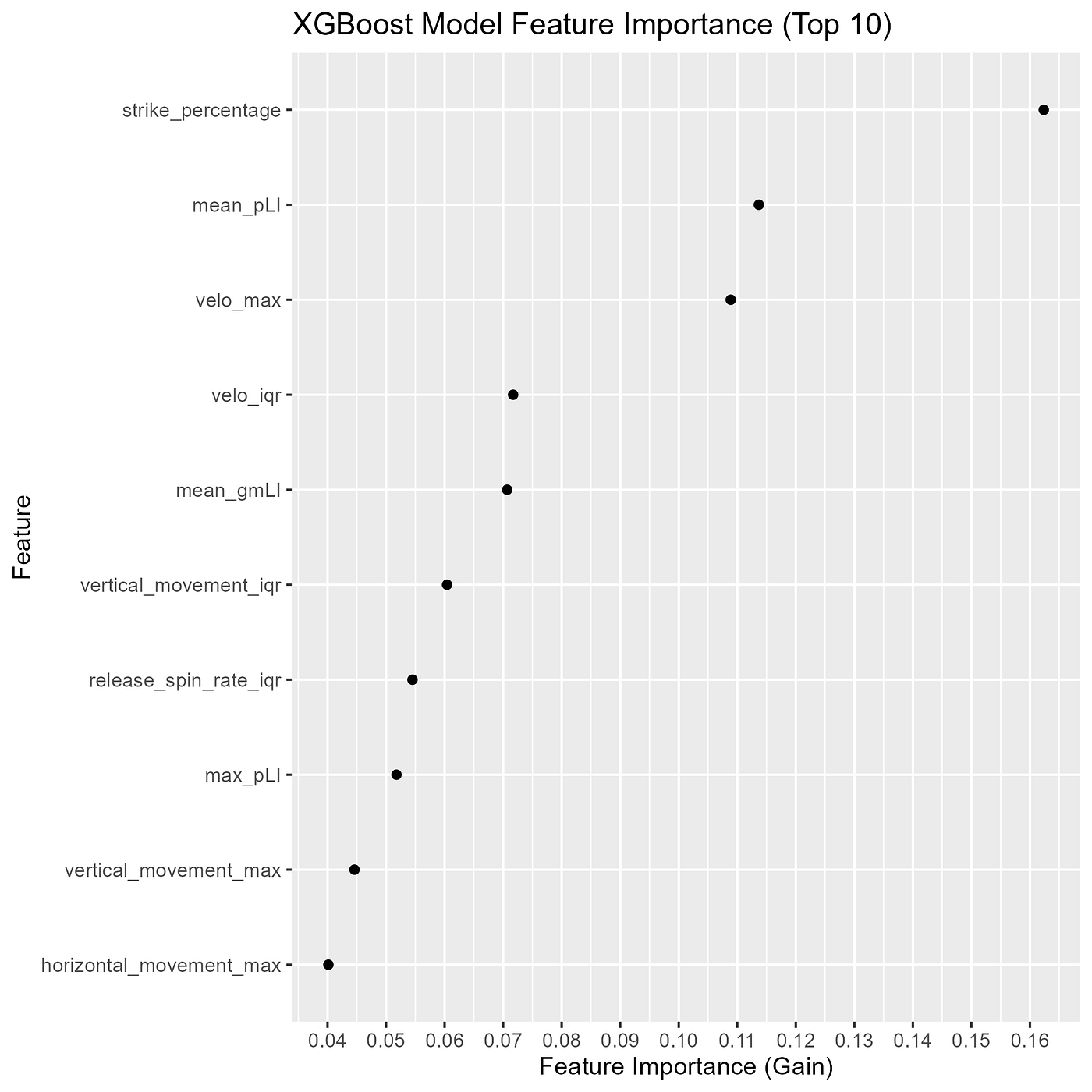
Utilizing all of the data and statistics created for it I was able to set up my model. I used the XGBoost gradient booster machine learning model to predict strikeout percentage, mostly due to its ease of use and power. I split my data into testing and training data, with 2021 being for testing and 2015–2019 being for training, with about a 20% train test split. I removed relief pitchers with three or fewer appearances in a season in order to improve the quality of my data, leaving me with 1,670 relief pitchers in the training data, and 388 pitchers in the testing data. These were then further split into features and labels, with the label being strikeout percentage.

Running my model with default parameters gave me a mean absolute error of approximately 5.9%. The mean absolute error (MAE) is the average distance between the test data value for strikeout percentage and the prediction line for it the model creates. My target was around 3%, which was the lowest MAE from Martin’s research. For strikeout percentage, an MAE of around 6% means that a prediction of 20% for a given pitcher may truly be 14% or 26%, which is a pretty wide swing in the context of the league averages for the stat. In order to lower the error I performed hyperparameter tuning using a grid search of the permutations of different parameters for the model.



The MAE for each pitcher in my test data, arranged from lowest error to highest. The x-axis here is each pitcher in the test set

I mainly worked with the hyperparameters ETA (the learning rate), the number of rounds the model would run, the maximum amount of depth each tree could go, and the percentage of features used to train each tree. I also looked through others that control how the model runs, such as alpha, gamma, and lambda, but none of these impacted my MAE by more than a marginal amount while increasing computational complexity. My model’s MAE settled around 4.6%, a solid improvement from just using the default parameters. I attribute this being higher than Martin’s value due to using a different dataset (relievers instead of starters), a different model (XGBoost instead of Random Forest), and a different method for my analysis (per pitcher instead of per pitcher’s pitch type). I am satisfied with this value as the vast majority of pitchers had a very low absolute erro, showing that my model was accurate.



The top ten most important features from my XGBoost model

With a solid model, I used feature importance to determine what factors were the best predictors of strikeout percentage. I found that strike percentage was the most important feature, which follows conventional logic that the more strikes you through, the easier it is to get strikeouts as there will be less walks. The average pLI was the second most important predictor. There are different interpretations of how leverage index could predict strikeout percentage. In low leverage situations, the pitcher may not need to focus too much on getting a strikeout and may decide to try out more of his breaking ball, or may find it easier to get strikeouts against worse opponents. On the other hand when the leverage is high, the pitcher may perform worse and won’t get many, or will feel “in the zone” and will have an elevated performance. Feature importance tells us less about what the outcome of a given stat will be, but rather what the best predictors of that stat are. Max velocity as well as its interquartile range are the next most important predictors, with maximum velocity being explained by faster pitches more easily being able to get past batter, thus causing strikeouts. The IQR may be explained by consistency, as a reliever who has great command over their pitch speeds may be better at getting strikeouts. Conversely, their consistency may cause them to be predictable, with batters knowing how to hit off them; pitch sequencing would tackle this question. Finally, the average gmLI comes in at fifth on the list, showing that the leverage at the start of the appearance is also quite important for the same reasons as discussed for pLI.

Notably, the first batter platoon advantage stat was not a particularly important feature. I am not going to immediately discount the platoon because of this, as there are very well measured differences in splits between handedness, but it may not be the best predictor for strikeout percentage. Furthermore, the number of pitch clusters (the only real application of clustering in this project) was also not significant, which can be explained by a lack of granularity in the possible values. Martin’s paper came to a similar conclusion about the number of clusters.

My main future goal for this project is to rework it to be per pitch cluster, as I this will let me apply my predictions to a more significant part of today’s game. I also will be adding the 2020 and 2022 years to my data, as there was never a really good reason for me not including the 2020 data and the 2022 data should now be available.

Overall, this project was a very positive experience for me. It served as the first time that I got hands on with real baseball data, and I look forward to using the techniques and strategies I learned from this process for future research. Furthermore, I now have a large dataset that I can utilize for other baseball research. The GitHub repository that contains all of my final work as well as the progression to that point can be found [here](https://github.com/LukeVanHouten/baseball_spa_drp).