IST 707 Final Project

Gabe Herz, Dan Beim, Kabir Thakur, Shane Halpin, & Pranjal Singh

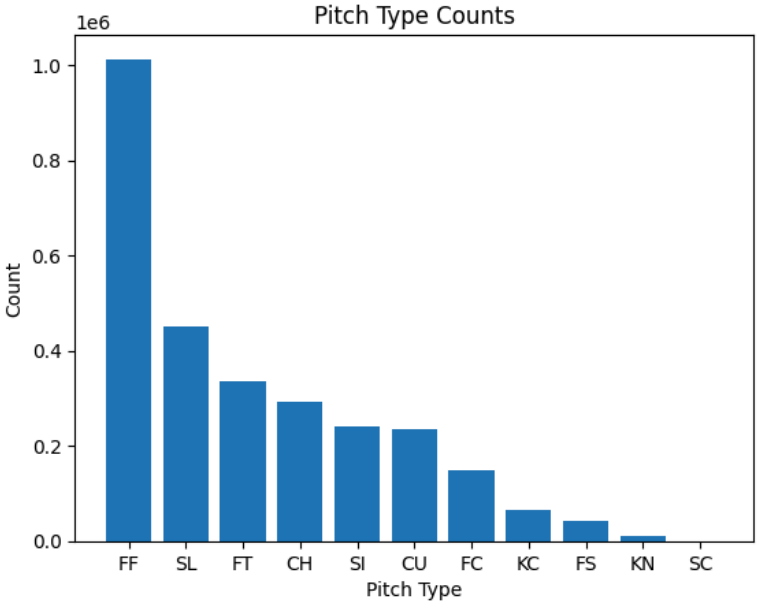
Overview:

In Major League Baseball, front office executives are tasked with identifying and acquiring talent that will help win games. Once signed, on-field personnel such as instructors and coaches work with the talent to maximize their abilities. With the rise of analytics throughout the game, there is more data now than ever that helps indicate whether a player may have success and what they should do to maintain success. For pitchers, the ability to throw a variety of pitches at different speeds and locations is imperative. For batters, the ability to recognize these pitches and the sequence in which they may arrive is key to having success.

Using MLB pitch-by-pitch data from 2015-2018, our group attempted to solve multiple issues that MLB front offices look at daily. Through association-rules mining we were able to provide actionable insights on both sides of the ball, helping pitchers identify what pitches are most effective in certain scenarios, and helping hitters prepare for pitches they may see in certain scenarios. Using k-Means clustering, we were able to expand our findings to a different area, helping teams prepare for matchups against unfamiliar pitchers, based on previously faced pitchers with similar pitch physics.

Data Summary:

Our dataset was taken from Kaggle under the name "MLB Pitch-Data 2015-2018." The dataset contained eight files, of which we used three. The pitches.csv file contained every single pitch thrown between the 2015 and 2018 seasons, adding up to around 2 million rows.

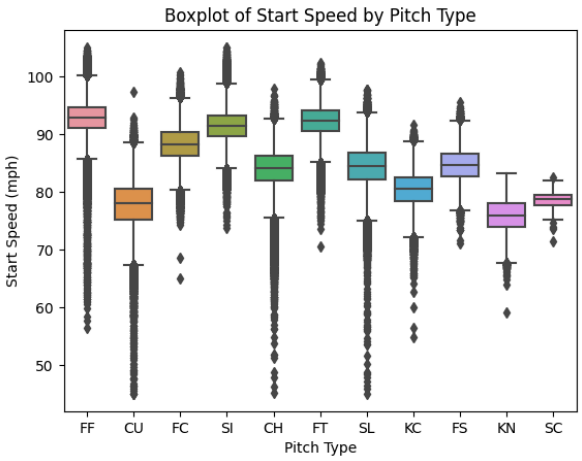


Number of Pitches Thrown by Pitch Type

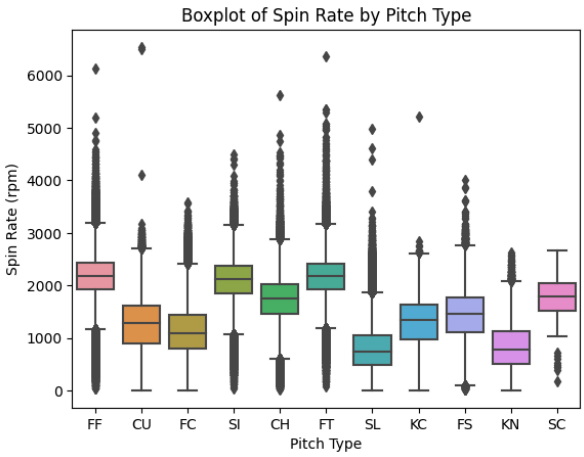
The table above, shows the breakdown of these two million pitches broken down by pitch type. The eleven pitches in the dataset are represented by two-letter abbreviations. These pitches are

four-seam fastball (FF), slider (SL), two-seam fastball(FT), changeup (CH), sinker (SI), curveball (CU), cut-fastball, or cutter, (FC), knuckle-curve (KC), split-fingered fastball (FS), knuckleball (KN), and screwball (SC).

The features of the dataset include physical data of the pitch, such as start speed, end speed, vertical break, horizontal break, spin rate, spin direction, and break length. Some basic differences in speeds and spin rates of these pitch-types are shown below in tables 2 and 3.



Velocity by Pitch Type



Spin Rate by Pitch Type

The dataset also includes game specific data such as the count, number of outs, and dummy variables indicating if there were runners on base. The atbats.csv contains the outcome of each at-bat, as well as the id of each pitcher and hitter. For association rules mining, we merged atbats with pitches so we could see which pitch metrics led to certain outcomes. For some of our clustering analysis, we merged the atbats file with the player_names.csv file, so we could see which actual pitchers performed better or worse against actual batters.

Association Rules Mining:

We used association rules mining to answer two main questions of interest. Firstly, we used pitch type association rules for game situations to discover what pitches are most commonly thrown in various game situations. The purpose of this analysis is to learn how pitch selection is associated with game situations, such as runners on base, outs, and ball/strike count, in order to achieve an edge from a hitting perspective. We also use outcome association rules for pitch type in order to discover what physical properties of various pitch types are associated with favorable outcomes (strikeouts for pitchers, base hits for batters). This analysis was focused on the two most common pitch types (fastballs and sliders). In this case, these association rules can help pitchers better understand how game outcomes can vary depending on the physical nature of how they throw these pitches.

The first steps of this process consisted of pre-processing the data in a way that was suitable for the apriori association rules algorithm, specifically for the first set of arules (for game situation). This entailed selecting only the relevant variables for game situations, and converting them into factor variables. This included creating a new variable called "count", which labeled each pitch based on the amount of balls and strikes prior to the pitch being thrown.

After preprocessing, the apriori algorithm was run, setting the right hand side equal to four seam fastball. Ideally, the results will show what in game situations are most associated with pitch type equal to four seam fastball. Selecting the ideal support and confidence minimums was executed mostly via a guess and check strategy. However, it is important to note the size of the data set and the base rates of the selected right hand side when selecting minimum support and confidence. Support is the percentage of observations in the dataset where the left hand and right hand sides are occurring at the same time. Ideally, there needs to be enough of these cases in order to make a valid analysis from the association rule. In this situation, we are working with a dataset consisting of millions of observations, so a support value even as low as 0.0015 still correlates with thousands of observations. Thus, the minimum support was set to 0.001, as smaller support values are still valid with this dataset. As for confidence, this value tells us the percentage of the time the right hand side occurs, when the conditions of the left hand side are met. Thus, as long as the value of the confidence for each association rule is significantly greater than the base rate of the right hand side in the dataset, then the rule is valid. In this case, four seams fastballs are observed roughly 35% of the time, so as long as the confidence of the association rule is significantly greater than this base rate, then the rule is valid. Thus the minimum confidence parameter was set to 0.6. The results of the algorithm generated 37 rules that meet these minimum parameters with lift greater than 1.0. The first 10 of these rules by confidence are presented below.

	lhs <chr>		rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>	count <int>
[1]	{Outs=0, On_1b=0, On_2b=0, On_3b=0, Count=Three-Zero}	=>	{Pitch_Type=FF}	0.001568323	0.6713318	0.002336136	1.881571	4461
[2]	{Outs=0, On_1b=0, On_2b=0, Count=Three-Zero}	=>	{Pitch_Type=FF}	0.001588010	0.6702775	0.002369183	1.878616	4517
[3]	{Outs=0, On_2b=0, On_3b=0, Count=Three-Zero}	=>	{Pitch_Type=FF}	0.001928675	0.6679654	0.002887387	1.872136	5486
[4]	{Outs=0, On_1b=0, On_3b=0, Count=Three-Zero}	=>	{Pitch_Type=FF}	0.001689260	0.6673611	0.002531254	1.870442	4805
[5]	{Outs=0, On_1b=0, Count=Three-Zero}	=>	{Pitch_Type=FF}	0.001734964	0.6656326	0.002606488	1.865597	4935
[6]	{Outs=0, On_2b=0, Count=Three-Zero}	=>	{Pitch_Type=FF}	0.001973675	0.6654813	0.002965786	1.865173	5614
[7]	{Outs=0, On_3b=0, Count=Three-Zero}	=>	{Pitch_Type=FF}	0.002135394	0.6638977	0.003216450	1.860735	6074
[8]	{Outs=0, Count=Three-Zero}	=>	{Pitch_Type=FF}	0.002229613	0.6618660	0.003368677	1.855041	6342
[9]	{Outs=1, On_1b=0, On_2b=0, On_3b=0, Count=Three-Zero}	=>	{Pitch_Type=FF}	0.001159806	0.6583516	0.001761682	1.845191	3299
[10]	{On_1b=0, On_2b=0, On_3b=0, Count=Three-Zero}	=>	{Pitch_Type=FF}	0.003695982	0.6575557	0.005620790	1.842960	10513

1-10 of 37 rows Previous 1 2 3 4 Next

Unfortunately, not much can be said about specific game scenarios where pitchers select the four seam fastball more frequently than its base rate. This is likely because the data are for ALL major league pitchers combined. It is likely that different pitchers tend to vary their approach in these various game scenarios to an approach that suits their individual strengths. It is likely that we would find better rules if looking at individual pitchers, one at a time. Nonetheless, there is still one strong association between four seam fastball usage and game scenarios for all pitchers. As we can see in the table above, every one of these rules contains the component "Count = Three Zero" (Three balls, zero strikes) in the left hand side, and each contains a confidence level

from 0.65-0.68. This can allow us to determine that no matter the game situation for runners on base and outs, if the count consists of a high ball/low strike scenario such as “Three Zero”, the pitcher is much more likely to throw a four seam fastball compared to its base rate.

Next, we ran the same apriori algorithm for association rules, however this time we set the right hand side equal to pitch type equals “Slider”, an offspeed breaking pitch that sweeps from one side of the plate to the other. It is important to note that the base rate at which sliders are thrown is roughly 16% of all pitches, so the minimum confidence was set to 0.2. The top 10 rules from these results by confidence with lift greater than 1.0 are shown in the table below.

	lhs <chr>		rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>	count <int>
[1]	{On_1b=1, On_2b=1, On_3b=0, Count=Zero-Two}	=>	{Pitch_Type=SL}	0.001065588	0.2401172	0.004437780	1.515829	3031
[2]	{On_1b=1, On_2b=1, Count=Zero-Two}	=>	{Pitch_Type=SL}	0.001451604	0.2395984	0.006058486	1.512554	4129
[3]	{Outs=2, On_2b=1, Count=Zero-Two}	=>	{Pitch_Type=SL}	0.001223439	0.2355649	0.005193641	1.487090	3480
[4]	{Outs=2, On_2b=1, Count=One-Two}	=>	{Pitch_Type=SL}	0.001868558	0.2342029	0.007978372	1.478492	5315
[5]	{Outs=2, On_2b=1, On_3b=0, Count=One-Two}	=>	{Pitch_Type=SL}	0.001411526	0.2333760	0.006048291	1.473272	4015
[6]	{On_1b=0, On_2b=1, Count=One-Two}	=>	{Pitch_Type=SL}	0.002084417	0.2331407	0.008940600	1.471786	5929
[7]	{On_2b=1, Count=Zero-Two}	=>	{Pitch_Type=SL}	0.002707035	0.2327972	0.011628299	1.469618	7700
[8]	{On_2b=1, On_3b=0, Count=Zero-Two}	=>	{Pitch_Type=SL}	0.002079144	0.2325325	0.008941303	1.467947	5914
[9]	{On_3b=1, Count=One-Two}	=>	{Pitch_Type=SL}	0.001999339	0.2316969	0.008629115	1.462672	5687
[10]	{On_1b=0, On_2b=1, On_3b=0, Count=One-Two}	=>	{Pitch_Type=SL}	0.001658674	0.2309914	0.007180675	1.458219	4718

1-10 of 126 rows Previous 1 2 3 4 5 6 ... 13 Next

These results show similar associations between game scenario and pitch type however in the opposite manner compared to when the right hand side was set to four seam fastball. Every one of these rules contains “Count = Zero-Two” or “Count=One-Two” as a component in its left hand side, meaning that a slider is more frequently associated with low ball/high strike scenarios. It is also notable that each of these rules also contains runners on either first or second base, meaning that there is an even greater association of sliders with low ball/high strike counts when there are runners on base.

The next step is to generate various association rules algorithms in order to discover what physical properties of individual pitch types lead to specific in-game outcomes, above their base rates. We do this by loading a new dataset that only looks at the pitches that generated an in-game result (in play outs, in play hits, strikeout, etc…). The two pitch types we are assessing are again fastballs and sliders and the two outcomes we are examining are the two most efficient in game outcomes in baseball: strikeouts for pitchers and general base hits for batters (single, double, triple, home run). We then preprocess the data in a specific way to suit the apriori algorithm. It is important to convert all of the numeric variable features that explain the physical properties of the pitches into categorical factor variables. Once this is done we have to create a variable called “general result” that groups all results that constitute a base hit into “Hit”. The data is now ready for association rules.

We begin by filtering for only the results that were generated by a fastball and set the right hand side equal to “result = strikeout” for the first set of rules and “general result = hit” for the second set of rules. The outputs will ideally show us the association between the physical properties of fastballs and the given outcome. It is important to note the base rate of strikeouts for the fastball

result data as roughly 23% and the base rate of general base hits as roughly 29%. The results of both sets of rules are shown in the tables below.

	lhs		rhs	support	confidence	coverage	lift	count
	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
[1]	{start_speed=Fast, px=In Left, pz=Out High}	=>	{R=Strikeout}	0.01309731	0.5021930	0.02608024	2.778522	1374
[2]	{px=In Left, pz=Out High, spin_rate=High}	=>	{R=Strikeout}	0.01093349	0.4658814	0.02346841	2.577618	1147
[3]	{px=In Left, pz=Out High, spin_dir=High}	=>	{R=Strikeout}	0.01308778	0.4652660	0.02812968	2.574213	1373
[4]	{start_speed=Fast, pz=Out High, spin_rate=High}	=>	{R=Strikeout}	0.01145777	0.4508627	0.02541298	2.494523	1202
[5]	{px=In Left, pz=Out High}	=>	{R=Strikeout}	0.02168587	0.4483642	0.04836665	2.480700	2275
[6]	{start_speed=Fast, pz=Out High}	=>	{R=Strikeout}	0.02232454	0.4360454	0.05119773	2.412542	2342
[7]	{start_speed=Fast, pz=Out High, spin_dir=High}	=>	{R=Strikeout}	0.01375504	0.4325540	0.03179959	2.393225	1443
[8]	{pz=Out High, spin_rate=High, spin_dir=High}	=>	{R=Strikeout}	0.01105741	0.4100389	0.02696674	2.268654	1160
[9]	{px=In Right, pz=Out High}	=>	{R=Strikeout}	0.01373598	0.4097242	0.03352493	2.266913	1441
[10]	{pz=Out High, spin_rate=High}	=>	{R=Strikeout}	0.01840678	0.3963465	0.04644113	2.192897	1931

	lhs		rhs	support	confidence	coverage	lift	count
	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<int>
[1]	{start_speed=Medium, px=In Right, pz=In High, spin_rate=Moderate}	=>	{GR=Hit}	0.01380392	0.3668390	0.03762937	1.2456681	1135
[2]	{start_speed=Medium, pz=In High, spin_rate=Moderate, spin_dir=High}	=>	{GR=Hit}	0.01509310	0.3620187	0.04169150	1.2292996	1241
[3]	{start_speed=Medium, px=In Right, spin_rate=Moderate, spin_dir=High}	=>	{GR=Hit}	0.01010666	0.3575731	0.02826460	1.2142041	831
[4]	{px=In Right, pz=In High, spin_rate=Moderate, spin_dir=High}	=>	{GR=Hit}	0.01405933	0.3550369	0.03959963	1.2055916	1156
[5]	{start_speed=Medium, px=In Right, pz=In Low, spin_dir=High}	=>	{GR=Hit}	0.01014315	0.3544411	0.02861730	1.2035688	834
[6]	{start_speed=Medium, pz=In High, spin_rate=Moderate}	=>	{GR=Hit}	0.02983350	0.3523919	0.08466001	1.1966102	2453
[7]	{start_speed=Medium, px=In Right, pz=In Low}	=>	{GR=Hit}	0.02039575	0.3495935	0.05834134	1.1871077	1677
[8]	{start_speed=Medium, px=In Right, spin_rate=Moderate}	=>	{GR=Hit}	0.02040792	0.3494377	0.05840215	1.1865788	1678
[9]	{px=In Right, pz=In High, spin_rate=Moderate}	=>	{GR=Hit}	0.02645245	0.3484460	0.07591550	1.1832112	2175
[10]	{px=In Right, pz=In Low, spin_rate=Moderate}	=>	{GR=Hit}	0.01197962	0.3481796	0.03440643	1.1823065	985

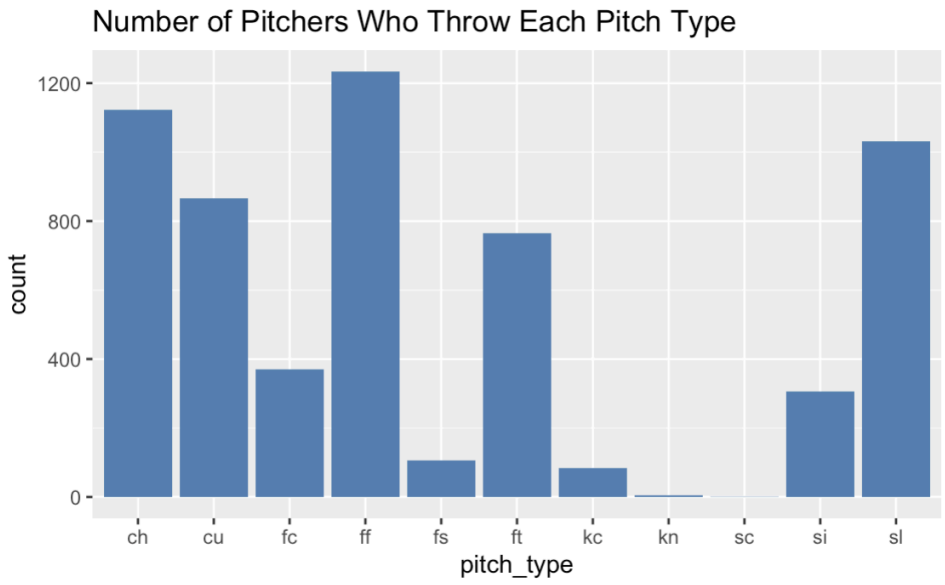
The results show us a few interesting associations between the physical properties of a fastball and the result as either a strikeout or hit. For fastballs that result in strikeouts, there is a strong association when the fastball is thrown high and outside the zone (pz = Out High), as every rule on the list contains this component. This vertical pitch location is also typically associated with a strikeout when the horizontal pitch location (px) is inside the strike zone and to the left side of the plate (pz = In left). These left hand side components also typically include “start speed = fast” or “spin_rate = high” allowing us to determine that there is even greater association between fastballs and strikeouts for balls thrown above the strike zone when both the spin rate and velocity are high. As for fastballs that are associated with base hits, these are typically found when the vertical location of the pitch is inside the zone (pz = In Left or In Right present in every rule). Many of these rules also include start_speed equals medium or slow. It is important to note that these rules are only for fastball data against left handed hitters. We also generated rules for right handed hitters which found similar results (see code file).

Finally, we also create association rules for sliders and outcome by filtering the outcome data for only sliders that generated in-game results and replicating the same process as with the fastball data. It is important to note the base rate of strikeouts for the slider result data as roughly 39% and roughly 22% for general hits. (To save space and time see code for results). The main takeaway from these rules are sliders that are left high and inside the strike zone are typically associated with base hits while sliders thrown low and outside the strike zone are typically associated with strikeouts.

k-Means Clustering:

For our clustering algorithm, we first attempted to look at which pitchers throw the same pitch type similarly. This would be helpful for MLB teams looking to compare a pitcher who they have faced very little of to someone who they have faced in the past, helping prepare better for

their upcoming matchups. It's also a useful tool to compare the relative strengths of the pitchers and the pitches they throw. The first step was to take all the pitches thrown over the sample and group them by pitch type and pitcher.

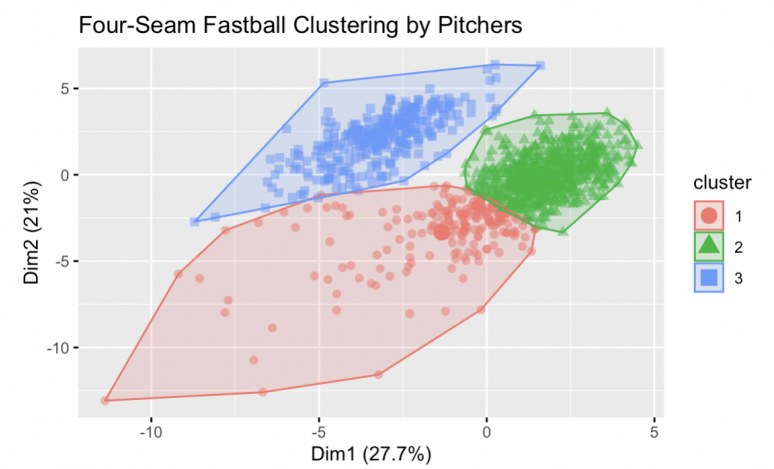


Number of PItchers who throw each pitch

The table above shows the counts of pitchers who throw each pitch. By grouping pitches by the amount of pitchers who throw each one, we can see which pitches we can make clusters based on. Out of the 11 pitch types, we ran clusters on nine of them, omitting knuckleball and screwball, as the number of pitchers who threw these pitches were four and one, respectively. This was not enough to generate clusters.

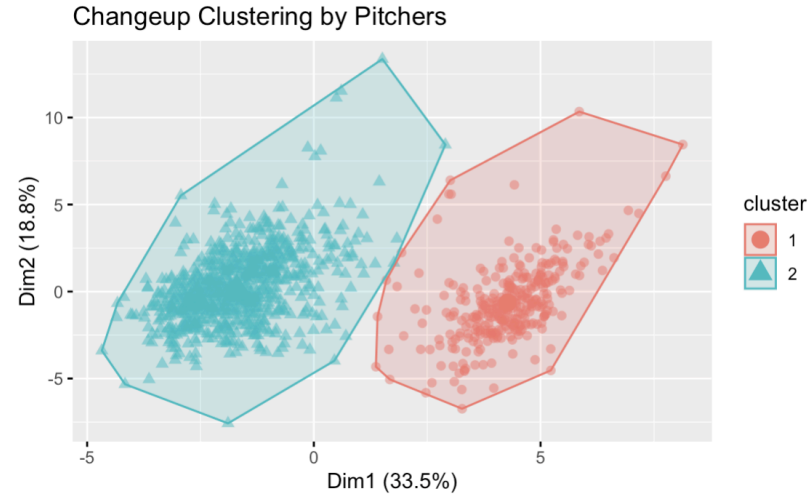
Next, the dataset was broken down into nine separate data frames, one for each pitch. For each of these nine dataframes, the same process was used. The data was scaled, a silhouette test was conducted, and then a k-means algorithm with the appropriate number of clusters was run and a cluster plot was displayed. Out of the nine clusters, adequate findings were found in seven of them. We could not get non-overlapping plots for curveballs and cutters.

For the purpose of the report, we can examine the clusters for the three most commonly thrown pitches in the sample, four-seam fastball, changeup, and slider.



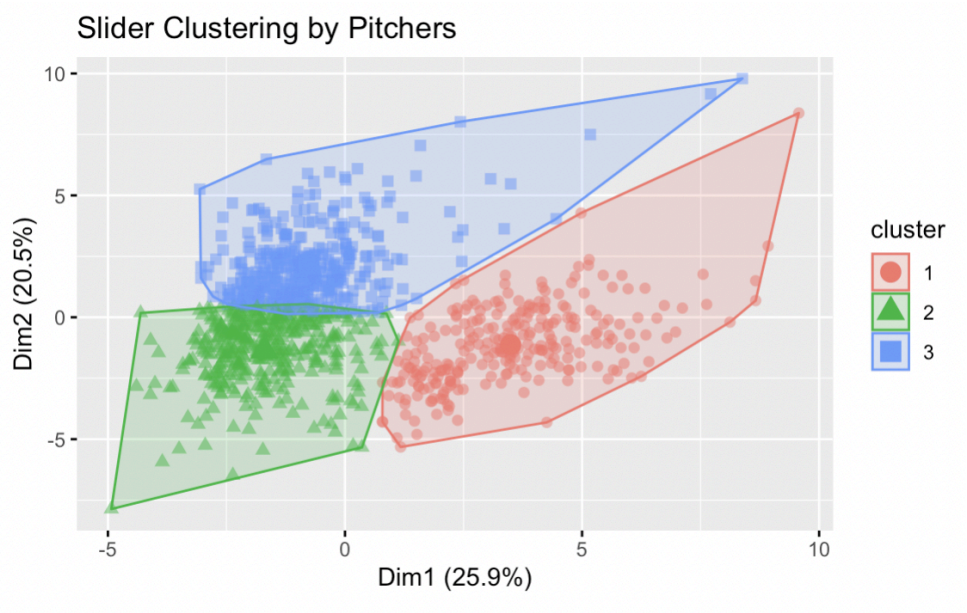
Cluster Plot for Four-Seam Fastballs

Based on the centroids of the clusters, we can identify which clusters are associated with certain characteristics. For the plot above, which shows the clusters for four-seam fastballs, cluster 1 was associated with less spin and less velocity. Cluster 2 contained the fastest pitches, with the most spin, and the most vertical movement. Cluster 3 contained middle-of-the-pack velocity and spin, but the most horizontal movement. Based on this, pitchers who throw the strongest fastballs would find themselves in the second cluster, while the worst would be bucketed in the first.



Cluster Plot for Changeups

The table above shows the cluster plots for changeups. The silhouette test returned an optimal value of two clusters for this pitch type. Ideally, a changeup is thrown at the same arm angle as a fastball, but it should be slower and have good vertical movement. The slower the changeup is, ideally, the better. Cluster 1 is associated with slower changeups, as well as higher vertical and horizontal break. Cluster 2 was associated with faster changeups with less movement, indicating that they come in too flat and similar to a bad fastball. If an opposing pitcher is classified in cluster 1, his changeup should be considered very effective. If in cluster 2, that pitcher's changeup could be an area to exploit and wait on, offensively.



Slider Clustering

The plot above shows the cluster plot for sliders. Ideally, a slider maximizes horizontal movement. Velocity is important in the absence of movement. Cluster 1 is associated with the least velocity and movement. Cluster 2 is associated with the fastest, highest spin sliders, with the most vertical movement. Cluster 3 is associated with sliders with the most horizontal movement. The results of this analysis are interesting. While we can confidently say that pitchers grouped in cluster 1 have exploitable sliders, the pitch physics present in both clusters 2 and 3 could result in effective pitches. Cluster 2 sliders are thrown hard, and have sharp downward movement which can be difficult to hit. Cluster 3 sliders are thrown with more finesse but break a lot horizontally, creating a sweeping effect across the strike zone. All else equal, cluster 3 sliders are likely the strongest, however the gap between them and cluster 2 is not too large.

Additional Clustering:

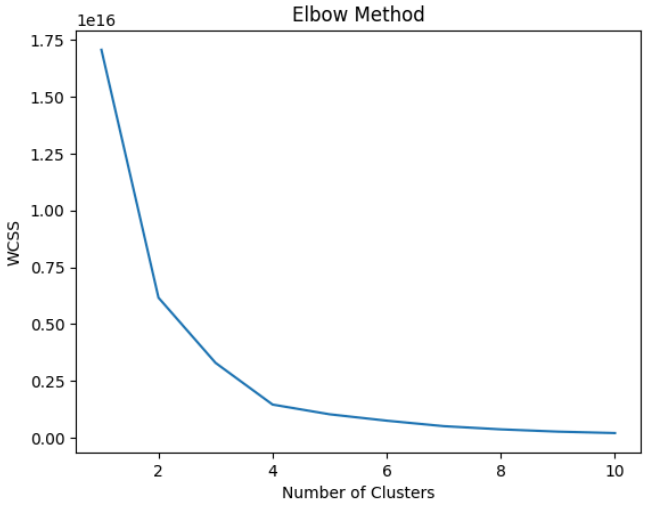
We performed k-means analysis on baseball pitches to identify clusters of pitchers with similar pitching characteristics. The primary goal is to recommend pitchers who are likely to have a positive outcome when playing against a particular batter, where a positive outcome is in favor of the pitcher.

The process involves the following steps:

Data cleaning and preprocessing: The 'pitches.csv' and 'atbats.csv' datasets are read, merged, and cleaned to remove unwanted pitch types and NaN values. The 'pitch_type' and 'type' columns are transformed into numeric values using the 'transformation' function.

Feature scaling: The selected features are scaled using StandardScaler from scikit-learn.

Clustering: K-means clustering is performed on the preprocessed data. The optimal number of clusters is determined as k=4 using the elbow method, which helps to minimize the within-cluster sum of squares (WCSS). Each pitcher is then assigned to a cluster based on their pitching characteristics.



Outcome calculation: A custom function, 'event_to_outcome', is created to categorize events as positive or negative outcomes in favor of the pitcher. A Decision Tree Classifier is trained on the data to predict the outcome (positive or negative) of each batter-pitcher pair.

Pitcher recommendation: A custom function, 'recommend_pitchers_with_confidence', is created to recommend pitchers with the highest probability of a positive outcome against a given batter. The function returns a list of recommended pitcher names.

Recommended pitchers to play against batter 518792

	id	first_name	last_name
24	276351	Jason	Grilli
68	150274	Joe	Nathan
70	112526	Bartolo	Colon
77	234194	Buddy	Carlyle
80	276514	Kevin	Gregg
147	115629	LaTroy	Hawkins
154	276542	Joaquin	Benoit
158	285064	Ryan	Vogelsong
243	279571	Matt	Belisle
250	285079	R.A.	Dickey
280	150359	A.J.	Burnett
282	218596	Tim	Hudson
287	282332	CC	Sabathia
322	150302	Jason	Marquis
329	279824	Mark	Buehrle
351	329092	Randy	Choate
484	136600	Bruce	Chen
681	150116	Randy	Wolf
728	217096	Barry	Zito
967	276520	Bronson	Arroyo

Overall, this code aims to identify the best-suited pitchers to play against specific batters based on their pitching styles and past performances, using k-means clustering and classification techniques.

The accuracy of this model may be limited due to the insufficient number of instances for each batter-pitcher pair in the dataset. The performance of the model is heavily dependent on the quality and quantity of the data available. As more data on batter-pitcher pairs become available, the accuracy and reliability of the model's predictions are expected to improve.

Conclusion:

Using association-rules mining and k-Means clustering algorithms, our group was able to provide actionable insights into problems MLB teams and front offices face on a daily basis. Through our arules model, we identified what pitches are most effective in certain scenarios and what pitches

are most common in certain scenarios. This helps a team's pitchers throw optimal pitches depending on the scenario, and a team's hitters to prepare for pitches they will likely see in certain scenarios. Using k-Means clustering, we were able to expand our findings to a different area, helping teams prepare for matchups against unfamiliar pitchers, based on previously faced pitchers with similar pitch physics. Our final k-Means model was not too useful given the data we had, however, with years more of data the results could prove to be very insightful, helping teams make optimal matchup decisions.

References:

https://www.kaggle.com/datasets/pschale/mlb-pitch-data-20152018?select=atbats.csv