ISyE 6740 Project Final Report
Optimizing MLB Lineups by Maximizing Overall Team Pitch Arsenal Range

Cale Williams
gtID 903749108

Fall 2023

1 Problem Statement

In baseball, every pitcher has different strengths & weaknesses. Team general managers have to take these attributes into account when building a roster and managers have to take them into account when deciding who should play in a given scenario. Having a roster with a wide range of strengths may be an advantage for managers.

This analysis quantified pitcher arsenal range and built rosters with maximum flexibility and skillsets.

2 Data Source

The data used in this analysis is from the 2023 Major League Baseball (MLB) regular season and comes via their Statcast[1] tracking technology. The pitch-level data was scraped using the *pybaseball*[2] Python package. Other pitching-related data needed was retrieved manually from Baseball Savant[3]. Player salaries were retrieved manually from Spotrac[4].

There were 808 players[5] who logged at least 10 plate appearances[6] as a pitcher, with a total of 716,876 pitches. To reduce small sample sizes,

[1] mlb.com/glossary/statcast
[2] pybaseball GitHub
[3] baseballsavant.mlb.com
[4] Spotrac query
[5] Baseball Savant query
[6] A plate appearance refers to a batter's turn at the plate.

pitchers with fewer than the median of 685 pitches were removed. The histogram of pitch counts by pitcher is shown in Figure 1.

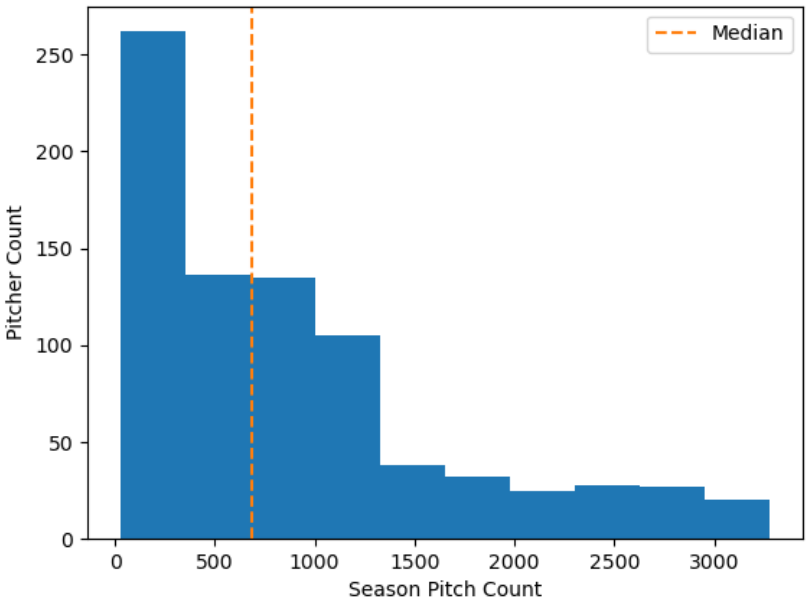

Figure 1: Pitch Count Histogram

This sampling method inherently biases high-volume pitchers due to ability and/or team strategy. If a pitcher is playing well, he may get outs with fewer pitches, but also stay in the game longer. If a pitcher is good, but does not have stamina, he may be removed from the game sooner. Further work could include a games played criteria to obtain a more representative pool of available pitchers. The resulting dataset included 405 unique pitchers comprising of 602,200 pitches.

3 Methodology

3.1 Pitch Classification

3.1.1 Data Preparation

Pitches were split by the pitcher's dominant hand and analyzed separately in this section, i.e., right-handed pitchers (RHP) are only compared to other

RHPs and left-handed pitchers (LHP) are only compared to other LHPs. The intent of this is to differentiate two pitchers who have pitches that are given identical names but may behave differently or vice versa. For example, the Kansas City Royals' Jordan Lyles (RHP) throws a 4-Seam fastball with an average velocity of 91 miles per hour and an average horizontal break of -6 inches. The San Diego Padres' RHP, Yu Darvish, has a splitter in his arsenal with an average velocity of 89 mph and horizontal break of -7 inches[7]. They are given different names, but according to these two metrics (albeit non-comprehensive), they are quite similar. All else equal, these pitches may be unique within each of the respective pitcher's arsenals, but to a batter, the pitches might be virtually identical. Therefore, to maximize uniqueness within a team, these pitches should be grouped together and a pitch with different characteristics should be identified and pursued.

3.1.2 Feature Selection & Dimension Reduction

The pitch features used were *release speed*, *release spin rate*, and *release spin axis*. See the Statcast documentation[8] for variable explanations, but this set of features should adequately capture the state of the ball at the time of release. Pitches missing these values were removed from the dataset. Less than 2% of each right and left-handed sets were missing. Further work could employ feature selection techniques.

First, the features were scaled and then to reduce dimensionality and visualize clustering results, principal component analysis (PCA) was performed on each dataset. These steps were done with the *scikit-learn* Python package using the *preprocessing.StandardScaler* and *decomposition.PCA* functions. The first two principal components were extracted and used for the remainder of this analysis.

3.1.3 Clustering

To classify the pitches, k-means clustering was employed using *sklearn.cluster.KMeans*. The elbow plot of within-cluster sum of squares (WCSS) vs. various values of k tested is shown in Figure 2.

At $k = 7$, WCSS appears to level off and adding clusters has diminishing returns. Therefore, $k = 7$ was chosen. The resulting pitch classifications and model decision boundaries are shown in Figures 3 and 4.

[7]Lyles & Darvish Baseball Savant query
[8]Statcast Search CSV Documentation

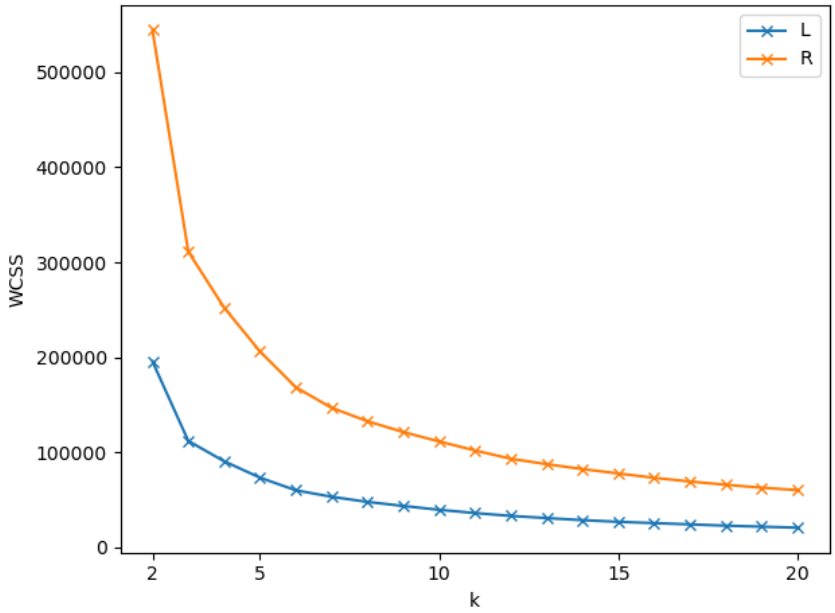

Figure 2: k-Means Elbow Plot

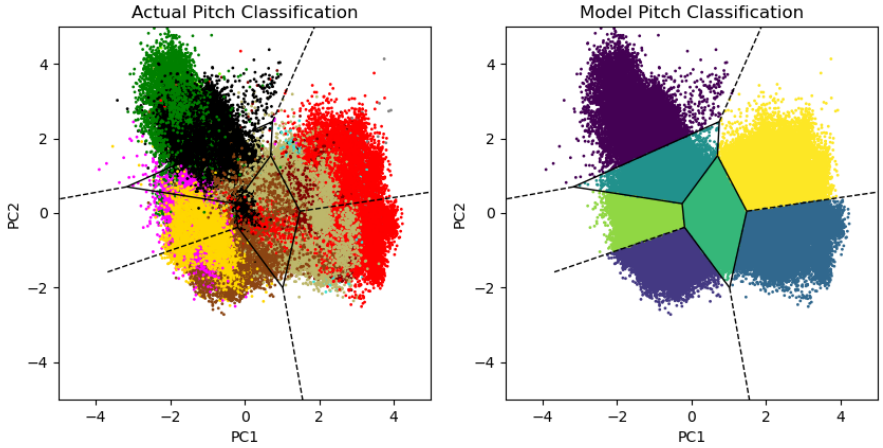

Figure 3: Pitch Clusters, RHP

For reference, the actual pitch classifications colors refer to the labels provided by Baseball Savant (i.e. fastball, slider, curveball, etc.). Although the decision boundaries do not line up exactly with the labelled data, this

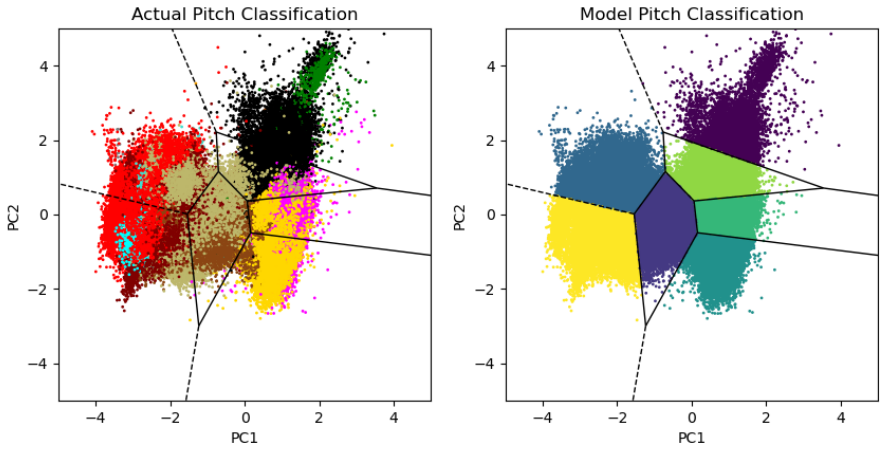

Figure 4: Pitch Clusters, LHP

was expected due to selecting a k less than Baseball Savant's number of pitch classes and performing PCA and clustering with data from all players rather than on a player-basis. It is also interesting that the plots mirror each other. This implies that the first principal component is a combination weighed more heavily by handedness, i.e., either *release spin rate* or *release spin axis*. There are also some visible clusters that were not differentiated in the model. This is due to limiting the k value as well as the lower densities in these regions. Future work could investigate using a Gaussian SVM model to classify pitches instead.

3.2 Pitcher Classification

Once pitches were classified, the three most frequent pitch classes were used to classify pitchers. With this "7 pitches choose 3" approach, there were 35 possible pitcher classifications. This was appropriate for some pitchers who had three pitches they often threw, but may not have been the best choice for pitchers with a different arsenal. Further work could improve this by taking the necessary top n classifications that make up some proportion of a pitcher's overall arsenal. A few examples of pitchers are shown below. For clarity, I will refer to the most frequent pitch class in a pitcher's classification as the *primary* class, the second most frequent as the *secondary* class, and the third most frequent as the *tertiary* class. For example, Nathan Eovaldi had a primary class of F, a secondary class of G, and a tertiary class of A,

as shown below.

Pitcher	Hand	n	Pitch Class Proportions [%] A	B	C	D	E	F	G	Class
N. Eovaldi	R	2,214	14.4	9.8	0.0	14.0	7.8	39.0	15.0	FGA
C. Kershaw	L	2,014	0.7	52.3	17.5	19.0	8.5	1.4	0.6	BDC
M. Stroman	R	2,123	2.2	16.9	27.2	6.0	6.7	40.0	1.1	FCB

As shown, Nathan Eovaldi was placed in the **FGA** category, but he threw a large number of pitches in the **D** cluster. His classification accounted for 68.4% of all of his pitches. However, for Clayton Kershaw and Marcus Stroman, three pitches was sufficient, accounting for 88.8% and 84.1% of their pitches, respectively.

3.3 Lineup Optimization

Finally, a pitcher lineup was generated via an optimization program using the *CVXPY* Python package. I did not attempt to quantify "how good" a pitcher is or what pitcher/pitch classification is best or leads to winning baseball games. Instead, I used the metric Expected Weighted On-base Average (xwOBA)[9] to quantify the quality of a pitcher. A lower xwOBA is better, hence the minimization problem. The optimization problem was given by:

$$\text{available pitchers, } n = 405 \quad \text{roster size, } r = 13 \quad \text{budget, } B = \$33,725,787$$

$$\text{xwOBA, } p = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} \quad \text{cost, } c = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad \text{handedness, } h = \begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix} \quad h = \begin{cases} 0 & \text{if left-handed} \\ 1 & \text{if right-handed} \end{cases}$$

$$\text{decision, } x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad x = \begin{cases} 0 & \text{if player is not selected for roster} \\ 1 & \text{if player is selected for roster} \end{cases}$$

$$\text{pitcher class index, } j = \{1, 2, 3\} \quad \text{pitch class, } k = \{A, B, C, D, E, F, G\}$$

[9] xwOBA

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} p_i x_i \\
\text{s.t.} \quad & x_i \in \{0,1\} \quad \forall i = 1, \ldots, n && (1) \\
& \sum_{i=1}^{n} x_i = r && (2) \\
& \sum_{i=1}^{n} c_i x_i \leq B && (3) \\
& \sum_{i=1}^{n} h_i x_i \leq 7 && (4) \\
& \sum_{i=1}^{n} h_i x_i \geq 6 && (5) \\
& \sum_{i=1}^{n} a_{j,k} x_i \leq 2 \quad \forall j, k && (6)
\end{aligned}
$$

The roster size, r, was determined by MLB roster limits[10]. The budget used was simply the median team pitching payroll in the 2023 season[11]. The x vector is the decision vector. Constraint 1 forces x to be a binary decision for each player. Constraint 2 forces exactly $r = 13$ players to be chosen. Constraint 3 forces the total salaries of selected players to be under budget. Constraints 4 and 5 force selection of 6-7 right-handed players. Of course, the remaining players will be left-handed. Constraint 6 is essentially 21 additional constraints. It forces the roster to have a maximum of 2 players with identical pitch classes in the primary, secondary, and tertiary classes. In other words, the roster cannot have pitchers with classifications of BAE, CAF, DAG, because of the identical secondary classes. However, a roster can have pitchers with classifications of ABC, BAC, and CBA. Although the first example may lead to a more diverse roster, it is restricted from being used. This is an area that can be improved upon in further work.

4 Evaluation & Final Results

The optimal lineup was found as:

[10] MLB 26-man Roster

[11] 2023 pitching payrolls

Pitcher	Hand	Class	xwOBA	Salary [$]
Hoby Milner	L	CFE	0.267	1,025,000
Tommy Kahnle	R	AFB	0.263	5,750,000
Alex Young	L	ADC	0.311	1,150,000
Justin Topa	R	FCG	0.268	720,000
Tim Mayza	L	EGC	0.285	2,100,000
Devin Williams	R	EBF	0.254	3,350,000
Felix Bautista	R	BAE	0.224	731,800
Tyler Rogers	R	GDA	0.258	1,675,000
Jeff Hoffman	R	CBD	0.247	1,600,000
Tanner Scott	L	DGB	0.246	2,825,000
Pete Fairbanks	R	BCG	0.252	3,666,666
Tarik Skubal	L	DEA	0.236	743,700
Tom Cosgrove	L	GED	0.254	1,200,000

All constraints were satisfied. There were 13 players, evenly split by handedness. The total salary was $26,537,166, well under budget. The average xwOBA was 0.259. A potential issue was the lack of starting pitchers. Many of these are relief pitchers, meaning they usually come in later in games and pitch fewer innings. This is probably due to starting pitchers having higher salaries. Future work could include splitting players into starting and relief groups and then adding constraints to ensure a more equitable split.

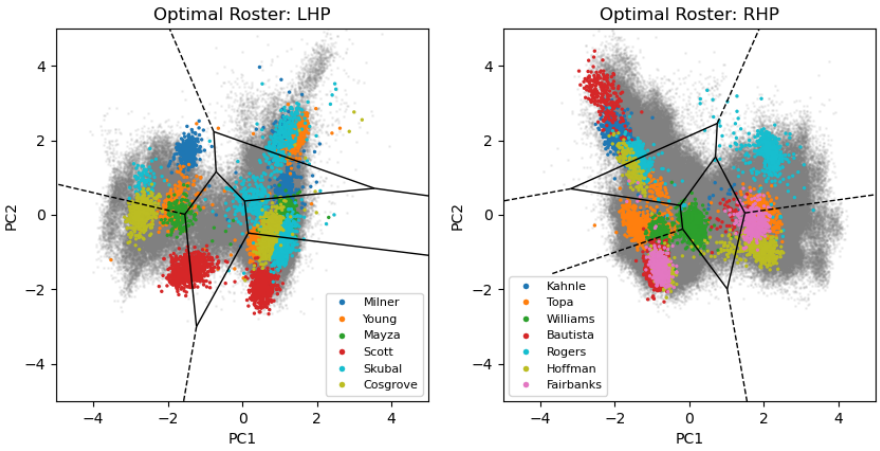

Figure 5: Optimal Roster

The pitches thrown by the rostered pitchers is shown in Figure 5. It is evident the roster was spread out amongst the principal component range.

A few pitcher's pitch ranges are spread across multiple clusters, which highlights the simple pitcher classification. Future work could include a variance measure in the optimization program constraints. Additionally, it should be noted that clusters in which there are more/fewer selected players may imply correlation between xwOBA and cluster. For instance, cluster **F** only appears four times in the optimal lineup, hinting that the attributes that describe this cluster are inversely related to xwOBA.

Overall, this project successfully employed a few machine learning methods to reduce dimensionality, classify datapoints, and finally build an optimal set subject to a defined objective function and constraints. Further improvements have been documented and, if implemented, could yield better results.