Is Your Pitcher Predictable?

You could say pitching is one of the most important aspects of Baseball. Pitchers are involved in every defensive play and based on ERA their performance makes a big difference in the flow of a game. But how can you tell how effective a pitcher is? This study will take a look at each pitcher in Major League Baseball and attempt to predict whether or not they are predictable.

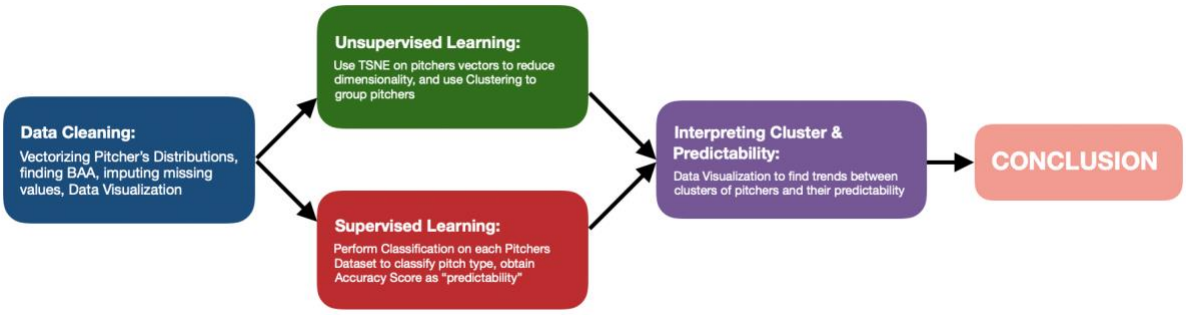
Using Pitch data from the 2015-2018 season we will predict the next pitch for each pitcher and depending on how well we are able to do so, we will determine how effective each pitcher is.

Table of Contents

Goals

One of the biggest goals of this project was to classify if a pitcher is predictable or not and come up with clear answers to interpret that "predictability" aspect of each pitcher. This was really difficult because a first I did not have much to go on in terms of predictability and what makes a pitcher predictable or not. Let's take a look at the pipeline to see the steps I took to come up with this metric.

Pipeline



The data preprocessing dealt a lot with imputing missing values, finding stats for each pitcher, and lots of visualization for understanding. After that I took a Supervised Learning route and found accuracy scores for each pitcher, and an unsupervised learning route that found relationships between pitchers based on the distribution of pitches that they continually throw. In the end I combined these two findings for a very interesting look at the predictability of each pitch.

Due to a lot of unknown at the beginning of this project, the pipeline can look somewhat intimidating. Here is the general overview about what I did and why:

The main goal that I set out was to find a relationship between a pitcher's performance and how predictable their pitches are. Using Supervised Learning I used a multi-classification model to classify the pitch type on each given pitch using, the count, runners on base, the batter, and several other features for each individual pitcher. This model provided an accuracy score for each pitcher that threw over 2,000 pitches which we can interpret as their "predictability". I also used dimensionality reduction techniques to perform unsupervised learning Clustering on each pitcher and their new "features" that can be interpreted as a combination of the percentages of each pitch type that they threw. By clustering each pitcher using Kmeans with 3 clusters, we were able to find impressive results between their individual "predictability" and their distribution of pitches that they throw.

Analysis

This Project involved a lot of EDA. This dataset was provided on Kaggle and contained Major League Pitching data from 2015-2018. The data was contained into separate csv's:

CSV NAME	CONTENTS	NUMBER OF ROWS	NUMBER OF COLUMNS

NAME	AB	CH	CU	EP	FA	FC	FF	FO	FS	FT	IN	KC	KN	SC	SI	SL
JUSTIN VERLANDER	0.0	0.05	0.15	0.0	0.0	0.002	0.59	0.0	0.0	0.001	0.0	0.0	0.0	0.0	0.0	0.2

ATBATS.CSV	count, outcome, batter name, pitcher name, inning, etc.	740,389	11
PLAYER_NAMES.CSV	first name, last name	2,218	3
GAMES.CSV	team, location, weather, score, etc.	9,718	17
PITCHES.CSV	speed, location, live game stats, etc.	2,867,154	40
EJECTIONS.CSV	name, time, game, etc.	761	10

Modeling

The interpretation of these pitcher's predictability came from two different types of modeling, Supervised Learning, and Unsupervised learning.

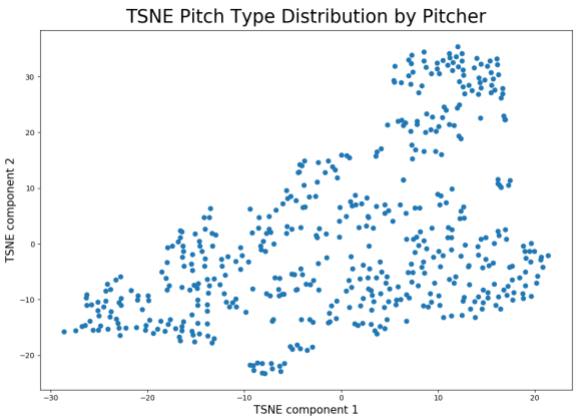
Supervised Learning:

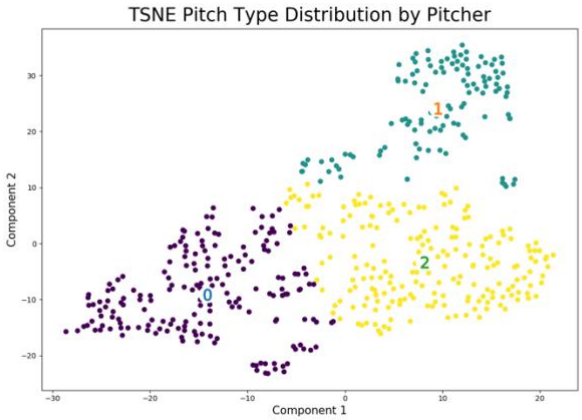
The first step was to run an out of the box Decision Tree Classifier on each pitcher and all of their pitches to predict the pitch type in each scenario. The Decision Tree Classification model was chosen because it performed the best in the shortest amount of time.

Unsupervised Learning:

After lots of eda and cleaning I was able to get a vector for each pitcher that describes the distribution of pitches that they throw. For example here is what Justin Verlander's vector would look like.

The next step was to use a dimensionality reduction technique called t-distributed Stochastic Neighbor Embedding. This technique took all of our pitchers and reduced the 16 different

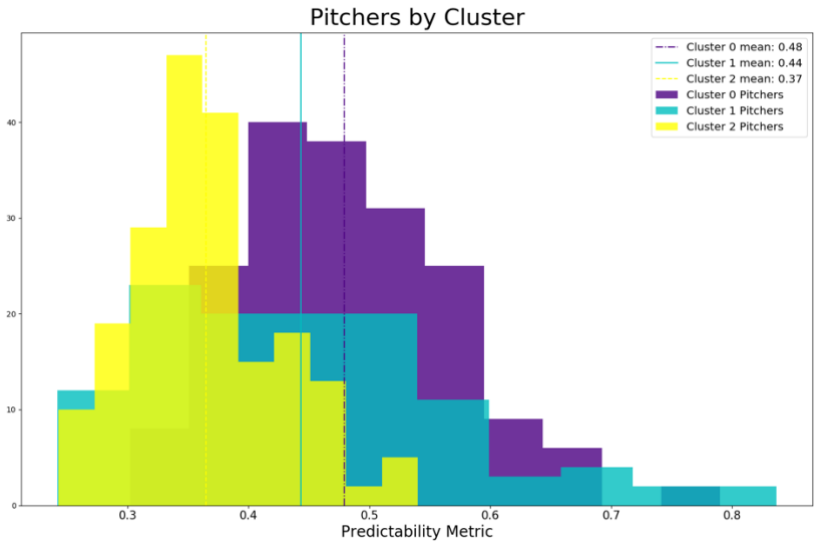




pitches into 2-dimensional data. Here is what the TSNE of our 500 pitchers looks like:

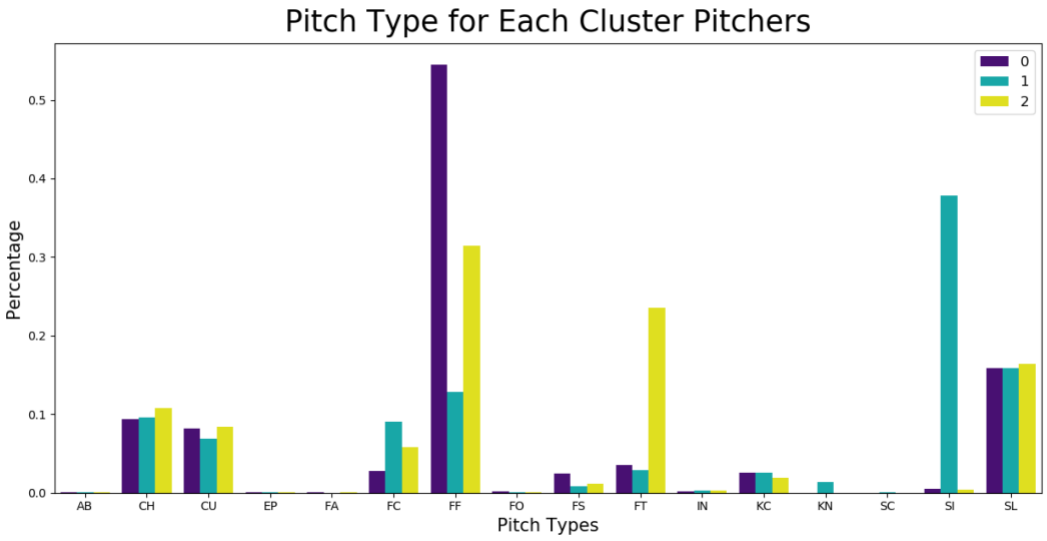
On the right I used Kmeans Clustering with 3 clusters to try to separate some of these pitchers.

But what do these clusters really mean? Let's take a look.



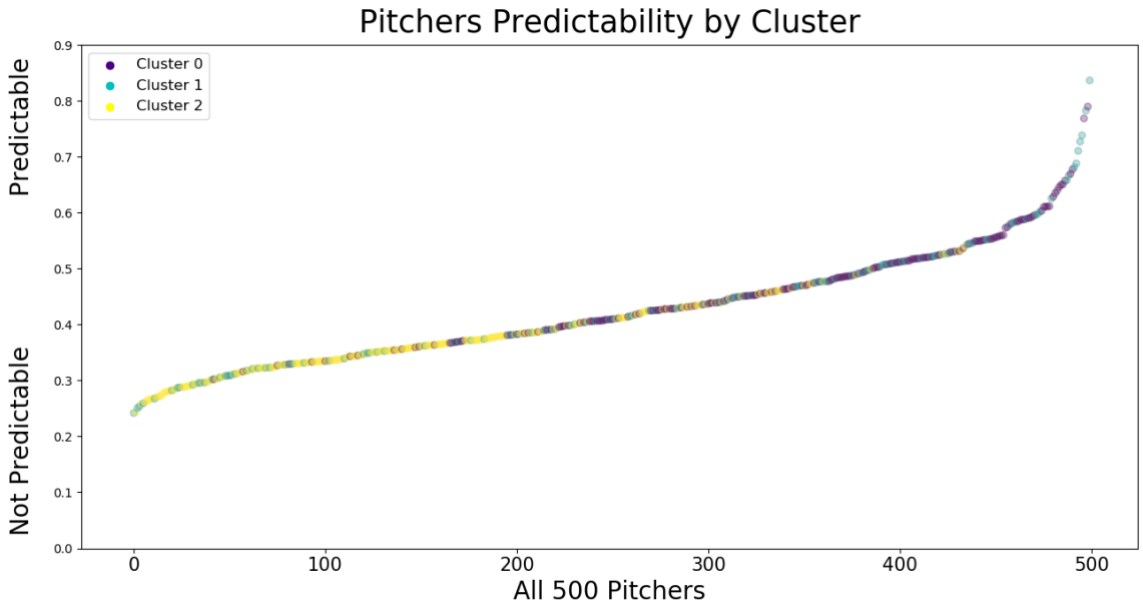
This plot contains histograms of all of the pitchers within each cluster. As we can see they all have different "predictability" means. Cluster 0 has the highest mean with a score of 0.48, and then Cluster 1 with a mean of 0.44 and Cluster 2 with a mean of 0.37.

The next thing I checked was to see what different clusters represented.



From this plot we can see all of the pitches along the x axis, and each bar corresponding to the percentage that that pitch comes up in each cluster. This is extremely interesting because now we can visualize what types of pitchers are in each cluster. For example, Cluster 0 contains our Four-Seam Fastball Pitchers, and Cluster 1 contains our sinker pitchers.

How does this all relate to the predictability?



If we look at each of our pitchers predictability and their cluster, we can see that pitchers in Cluster 2 and Cluster 1 are much less predictable than pitchers in Cluster 0.

Conclusions

1. Create a Model for each pitcher and obtain an accuracy score as their "predictability" measurement
    - If a pitcher is very predictable in the types and timing of pitches that they throw then they will have a high score.
    - If a pitcher is not predictable, then the model had a more difficult time determining which pitch they would throw in each situation, and they will have a lower score.
2. Use TSNE to reduce Pitchers Pitch Distributions
3. Perform Unsupervised Learning Kmeans to cluster pitchers pitch distributions

Findings

1. We can now interpret the predictability of a pitcher based on the type of pitches that they throw.
2. Does being less predictable mean, you are a better pitcher? Not necessarily. Being "predictable" in this study means how easily can our models predict the next pitch type that will be thrown by a certain pitcher. This has nothing to do with the outcome of the pitch. Being less predictable means that it is more difficult for our models to determine what the pitcher will throw next, not that the pitch they throw next will be good.

Although we can make statements about each pitcher specifically and how predictable they are. For example, given a scenario facing a certain pitcher that has a tendency to pitch so many Fast balls, so many Curve balls, and so many Knuckle balls, we can determine how likely they are to pitch a Fast ball next based on how predictable they are.

A pitcher that throws every pitch at the same probability, will be considered less predictable.

With this information we can educate batters with tendencies for each pitcher. By first categorizing them based on their predictability, and then observing their pitching tendencies.

There are 3 different types of pitchers that we have found.

- Pitchers that throw a lot of Four-seam Fast balls.
- Pitchers that throw a lot of Sinkers.
- Pitchers that throw a wide distribution of pitches.

Based on the type of category a pitcher falls in, we can educate batters about that pitcher's tendencies and what they should expect.