Machine Learning Engineer Nanodegree

Andrew Doyle
August 17, 2017
Capstone Report

Definition

Section 1 - Overview

Domain Background

For my Capstone project I will be applying several different unsupervised learning algorithms to Major League Baseball data, specifically clustering algorithms, in order to properly classify each pitch into its type (i.e. Four-Seam Fastball, Curveball, etc.). There are several reasons I chose the domain of baseball for my final project. First and foremost, I am a big fan of baseball, particularly the analytical and quantitative side of the game. It was this interest that lead me to the field of machine learning in the first place, and even helped me learn some machine learning and statistical software (Marchi & Albert, 2013). Statistical analysis is in the DNA of the game of baseball, as game data has been collected for as long as the game has been around. Because of the vast amount of data, the field lends itself very well toward machine learning.

Secondly, with the advent of the Statcast system, recently collected data can offer insights into the game that were previous unavailable. Per the website Baseball Savant (https://baseballsavant.mlb.com/about), Statcast

> is a state-of-the-art tracking technology, capable of measuring previously unquantifiable aspects of the game. Set up in all 30 Major League ballparks, Statcast collects data using a series of high-resolution optical cameras along with radar equipment. The technology precisely tracks the location and movements of the ball and every player on the field, resulting in an unparalleled amount of information covering everything from the pitcher to the batter to baserunners and defensive players.

Finally, the datasets for MLB data are both large and free to use by anyone, making it very conducive toward this type of analysis. Machine learning has been applied to baseball data both publicly among the baseball community for pleasure and privately among professional teams with the goal of gaining a competitive advantage. An example of using this field of research on a Statcast dataset would be using a Random Forest classifier to predict whether a batted ball will be a hit or an out (http://www.hardballtimes.com/using-statcast-data-to-predict-hits/).

Datasets and Inputs

Statcast tracks many different features of each pitch thrown in a major league baseball game, with all data being free to use and publicly available via www.BaseballSavant.com. The impetus for the project came from watching the game - pitch recognition is a vital aspect of being a successful hitter. This recognition is based on some combination of the velocity of the pitch, the spin direction and rate, and horizontal and vertical movement (known as the break).

The specific data set I will be using will be every pitch thrown by Chris Sale, starting pitcher for the Boston Red Sox. Chris Sale has been one of the very best pitchers in baseball throughout his entire career, and this year has arguably been the best of his career and one of the best statistical years by a starting pitcher in Red Sox franchise history. He is on pace to set career highs in Wins Above Replacement, Strikeouts and Strikeouts per Nine Innings Pitched, and is already in the discussion for not only the American League Cy Young Award but the Most Valuable Player Award as well (http://www.fangraphs.com/blogs/chris-sale-for-mvp/).

Section 2 – Problem Statement

The problem I will be attempting to solve is "can unsupervised Machine Learning algorithms correctly classify pitches thrown by a Major League Pitcher into their proper pitch type?" Using Statcast data, consisting of perceived velocity, spin rate, horizontal movement, and vertical movement I will apply unsupervised clustering algorithms in my effort to accomplish this. Statcast tracks the above features for every pitch that is thrown in major league baseball, as well as the result of each pitch and numerous other metrics. The metric I will be attempting to identify is the resulting label of the pitch type.

Section 3 – Metrics

Because the label of each pitch is already known, I can use the pitch type as a way to evaluate the accuracy of each cluster created, as well as gain any insights into the pitch repertoire of this particular player.

Analysis

Section 4 – Data Exploration

The first thing to do is to download Chris Sale's dataset from Baseball Savant, which can be found here: (Chris Sale Data). I will save the file as 'sale_pitch_data.csv'. Once the file is downloaded it can be loaded into the iPython file the we can explore what we have.

Now that the dataset has been downloaded, we can begin by viewing the data as is, prior to any processing. Analyzing the data shows us that the dataset has 2626 samples with 78 features each. Not all 78 features will be relevant to this analysis however, and some fields aren't populated at all. For this particular analysis I will be using a subset of those 78 features, specifically 'pitch_type', the type of pitch, 'effective_speed', the perceived velocity of the pitch (in miles-per-hour), 'release_spin_rate', the rotational spin rate of the pitch, 'release_extension', the amount of extension the pitcher achieves on the pitch, 'pfx_x', the horizontal movement of the pitch, 'pfx_z', the vertical movement of the pitch. I will name the subset of the original data, which consists only of the above reference fields, as 'df_pitch'.

Now that I have the subset of data which I will use in the clustering algorithm, it needs to be processed before any further analysis can be done. The first step in this is to remove any null values in the original data, which exist throughout the dataset. Once those indices are removed, the data type of each

feature needs to be examined to ensure that the processing will work correctly. Examining the data type of each feature gives me the following:

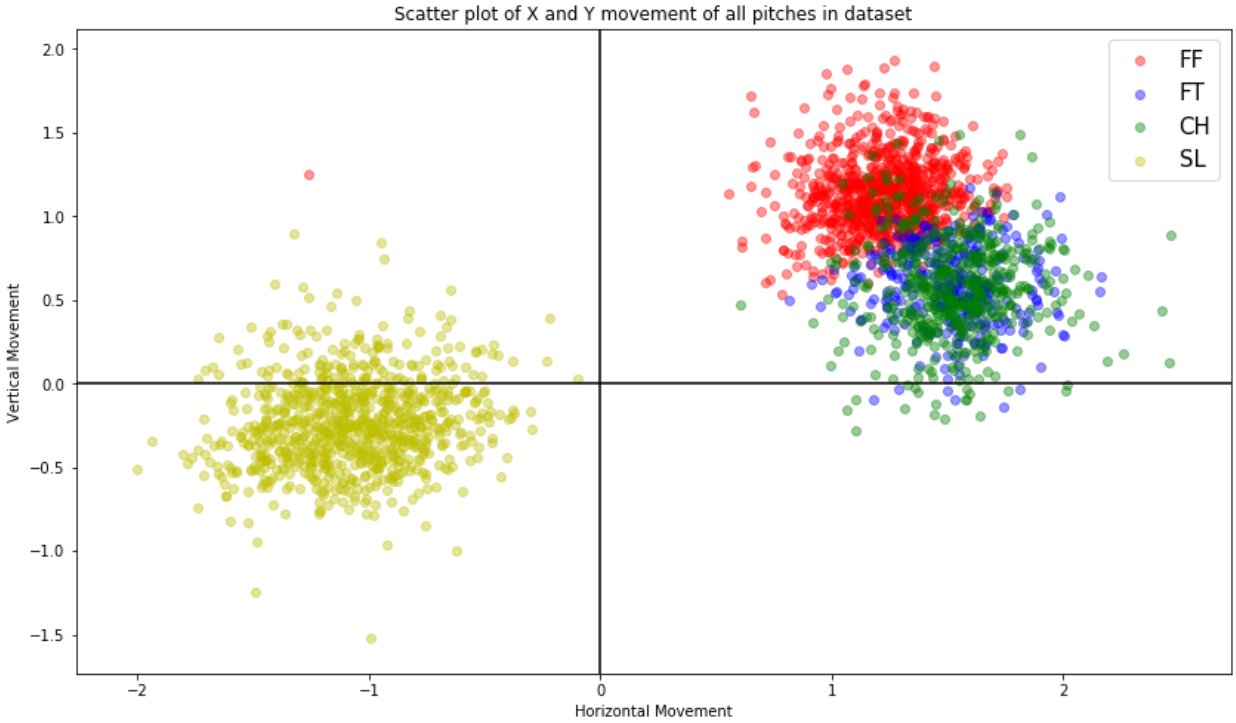
pitch_type	object
effective_speed	object
release_spin_rate	object
release_extension	float64
pfx_x	float64
pfx_z	float64
dtype: object	

A few of these features need to be converted to a different data type, in this case, I want to convert all numerical data to a float data type.

Once the null values have been removed and all numerical features are stored as floats, I will calculate 2 new features based on the existing data. Those 2 features are the 'break', which is the magnitude of the movement vector, and the 'angle', which is the angle the pitch moves at relative to the release point of the pitcher.

The 'break' can be calculated by using the Pythagorean theorem on the horizontal and vertical movement variables for each pitch. Similarly, the angle each pitch breaks at can be calculated by using trigonometric functions based on the same variables.

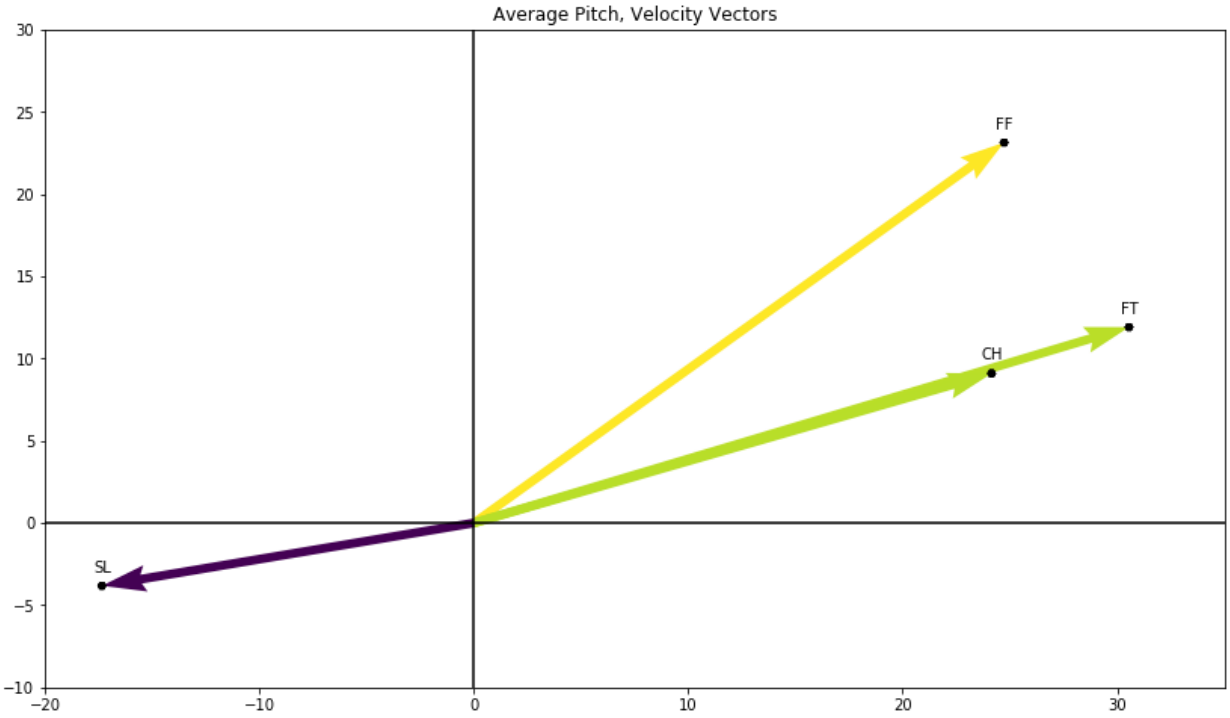
Calculation of the angle requires careful consideration however. Normally angle is calculated with respect to the positive portion of the x-axis. This becomes a problem however when a particular pitch type exhibits positive vertical movement in some instances and negative vertical movement in other instances. Because of this issue, I want to make sure that the axis which the angle is calculated relative to does not pass through a group of pitches, thereby skewing the calculation of the angle, which in turn would throw off any clustering algorithms performed in the data. To illustrate this point, I will plot the horizontal and vertical movement of each pitch in the dataset:



Using the positive portion of the x-axis as the starting point for calculating the angle would result in some pitches being classified with an extremely low angle (<15°) and others being classified with an extremely high angle (>345°), when these pitches are very similar in terms of their horizontal and vertical movement. For this reason I will use the negative portion of the y-axis as my line of demarcation from which angle will be calculated.

To illustrate how the angle is calculated, let's suppose Chris Sale threw a changeup that exhibited no vertical break, and positive horizontal break. This pitch would thus like on x-axis, and it's break angle would be calculated at 90°. Similarly if Chris Sale threw a slider that exhibited no vertical break, but had negative horizontal break, the break angle would be 270°.

Now that the dataset is complete, I want to explore each type of pitch a little more to get an idea of how they vary from one another. By looking at the mean values for the 4 types of pitches Chris Sale thrown, namely 4-seam fastballs ("FF"), 2-seam fastballs ("FT"), changeups ("CH") and sliders ("SL"), we can view the features for the average pitch of each type. In addition to calculating these mean values, I also will plot a vector for each average pitch so we can see how the mean values of each pitch differ from one another.



These vectors are merely to give an idea of what the average fastball, changeup and slider thrown by Chris Sale does. The length of the vector represents the velocity and the angle of the vector is the angle the pitch typically breaks at. There are a few things we can infer about each pitch based on the graph.

The first is that it looks like both types of fastballs and the changeup move "up", but this doesn't mean the ball is rising, in the traditional sense. The horizontal and vertical movement of each pitch recorded is done so relative to a pitch thrown with no spin whatsoever and which is only affected by gravity (http://www.fangraphs.com/library/pitch-type-abbreviations-classifications/). In this case, the 4-seam fastball, 2-seam fastball and changeup all have "positive" vertical movement in the sense that they do not exhibit the expected amount of drop due to gravity.

The second takeaway from this plot is the similarities and differences between each type of pitch. The 4-seam has the highest average velocity and tends to have positive vertical and horizontal break. The 2-seam fastball and changeup are very similar in terms of both break and velocity, however the 2-seam is typically thrown harder. The slider Chris Sale throws is clearly the most unique of the pitch-mix: it is both the pitch with the lowest average velocity and the only one with "negative" movement.

Now that the dataset has been properly formatted and holds all features we are going to use, I will select a set of sample points which we will look at in the future once the cluster analysis has been completed. I am primarily interested in how each pitch type is classified and whether the clustering algorithms can properly label each pitch type. I will thereby randomly select 4 indices, each one classified as a different pitch type.

Sample	pitch_type	effective_speed	release_spin_rate	release_extension	pfx_x	pfx_z	break	angle
472	FF	93.903	2318	5.726	1.0795	1.1848	1.602832	137.66259
837	FT	92.712	2299	6.013	1.9656	0.6799	2.079867	109.080476
1484	CH	85.242	2042	5.668	1.4825	0.2493	1.503315	99.545656
2431	SL	75.803	2349	5.083	-1.3468	-0.6067	1.477144	294.25038

Section 5 – Feature Relevance

Now that we have explored the pitch dataset a bit, we can start to look closer at the different features to see how relevant they are to one another. The first step in this process is to run a supervised regression algorithm on the dataset with a feature removed to view how well that feature can be predicted by the rest of the dataset. First, I will attempt to predict the angle of break using a regressed decision tree.

When the angle is attempted to be predicted, the dataset does so with a correlation coefficient of >.99, meaning that more than 99% of the variation in the angle can be determined by the rest of the dataset. The angle can be predicted with incredible accuracy – which should be expected, as the angle was calculated using the information we already had. So let's try to predict another feature which wasn't calculated, effective_speed. Running the same test but on the effective speed of each pitch produces a correlation coefficient of >.83, again a very high number, which means the dataset can predict the velocity of each pitch with a high degree of accuracy.

Continuing with our exploration of the features in the dataset, I will create a scatter plot of each feature paired with every other feature to show how the related to one another, as well as how each feature is distributed.

Section 6 - Algorithms and Techniques

As indicated in the introduction I will be performing 2 types of cluster analysis on the dataset, KMeans and Gaussian Mixture. Each has it's own pros and cons.

K-means clustering identifies a specified number of centroids that cluster the data points together. These centroids aim to minimize the 'inertia' of each cluster. This algorithm works well for very large sample sizes, however because it is iterative until a minimum is calculated it can take time to run. It scales well to extremely large samples.

GMM clustering is a clustering algorithm used when you are trying to estimate covariance in the data. It is a faster algorithm, however if the sample is too small covariance can be difficult to calculate.

The clustering method varies between the two however. K-means attempts to separate the data into groups of equal variance, whereas GMM will create clusters of covariance.

The goal of this project is to see how each algorithm describes the dataset and then compare to the actual pitch labeling. Because this is exploratory in nature, I will perform both types of clustering on the dataset and see what each says about the data.

Methodology

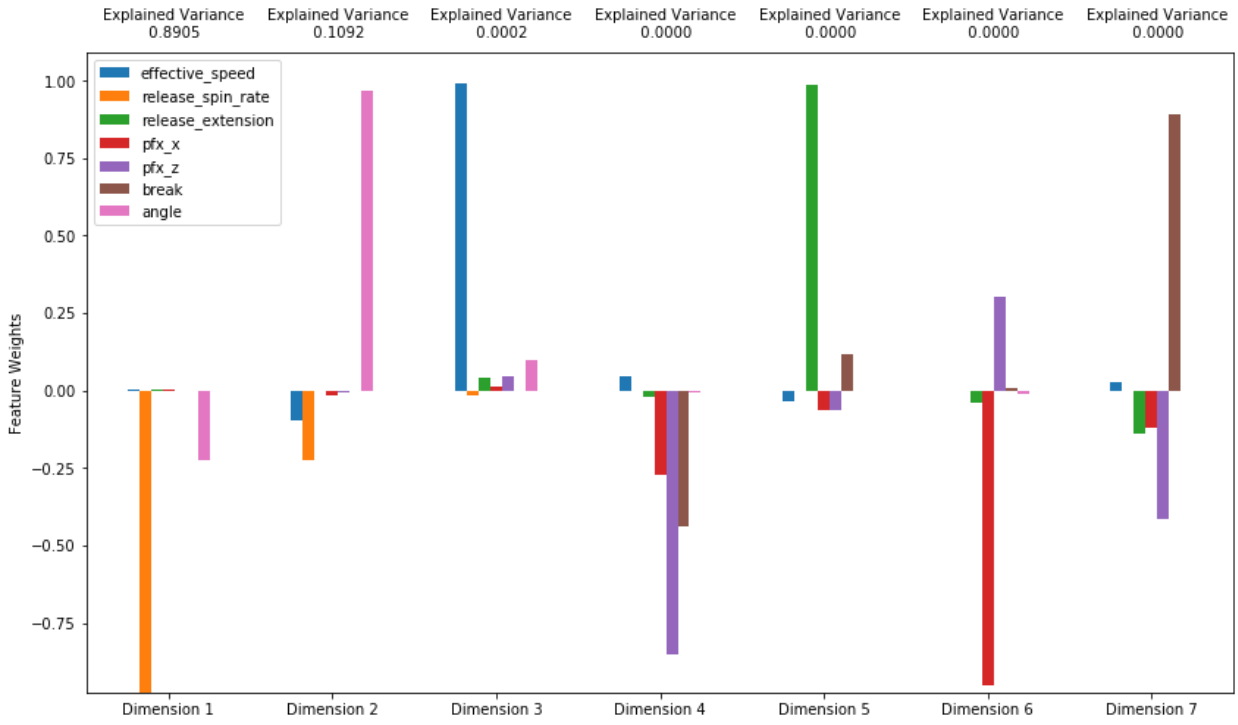
Section 7 – Data Preprocessing

Now that we have analyzed the data closely and viewed the features, we can start the process of performing the clustering. Prior to doing this the dataset needs to be preprocessed so it is in a format that the clustering algorithms can use.

The primary step in preprocessing is reducing the dimensionality of the dataset through principal component analysis. Principal component analysis, per scikit-learn, is

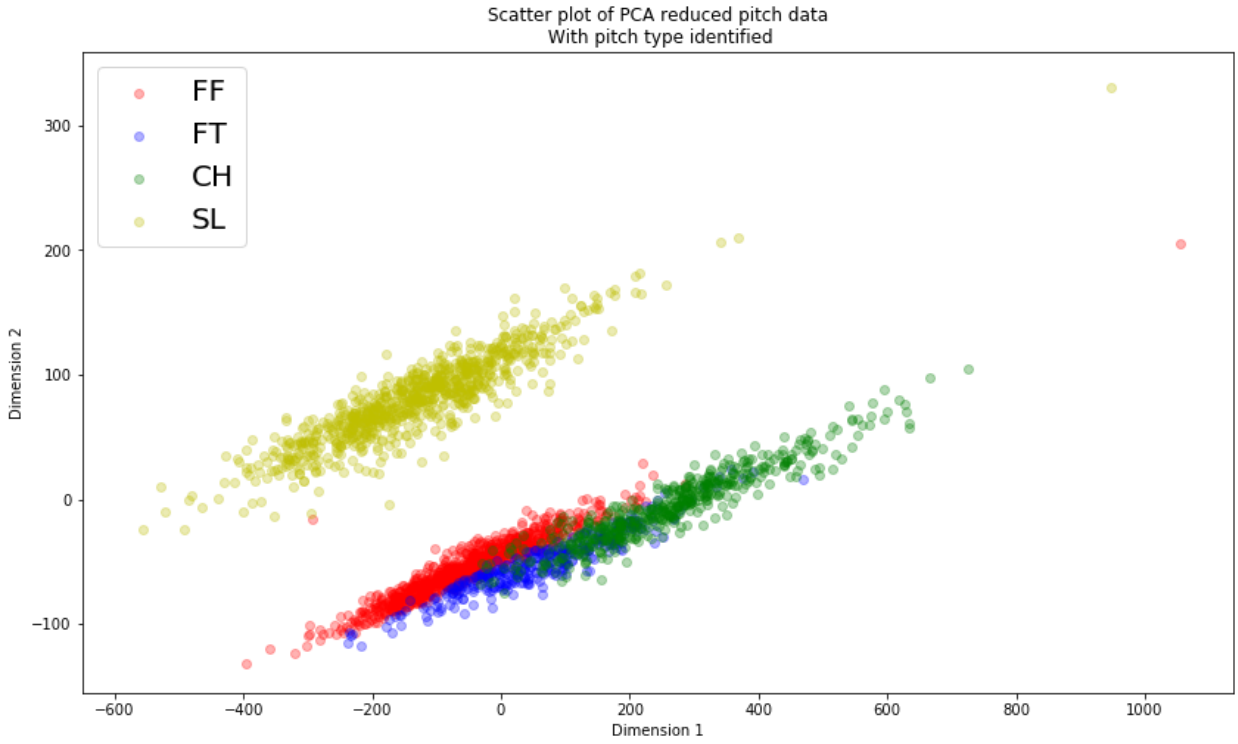
> used to decompose a multivariate dataset in a set of successive orthogonal components that explain a maximum amount of the variance. In scikit-learn, PCA is implemented as a transformer object that learns n components in its fit method, and can be used on new data to project it on these components (http://scikit-learn.org/stable/modules/decomposition.html#pca).

Once we reduce the dataset to 2 dimensions, we can continue to explore the data. The first thing to do is view how much variance is explained by each dimension calculated, which is shown below.



We can then calculate the total explained variance of the first two dimensions, which is the dataset we will perform the clustering on. That total variance is >.99, meaning nearly the entirety of the variance in the dataset is explained by the first 2 dimensions.

Now that PCA has been performed on the data, we can construct a scatter plot to see where each pitch lies relative to the first 2 dimensions.

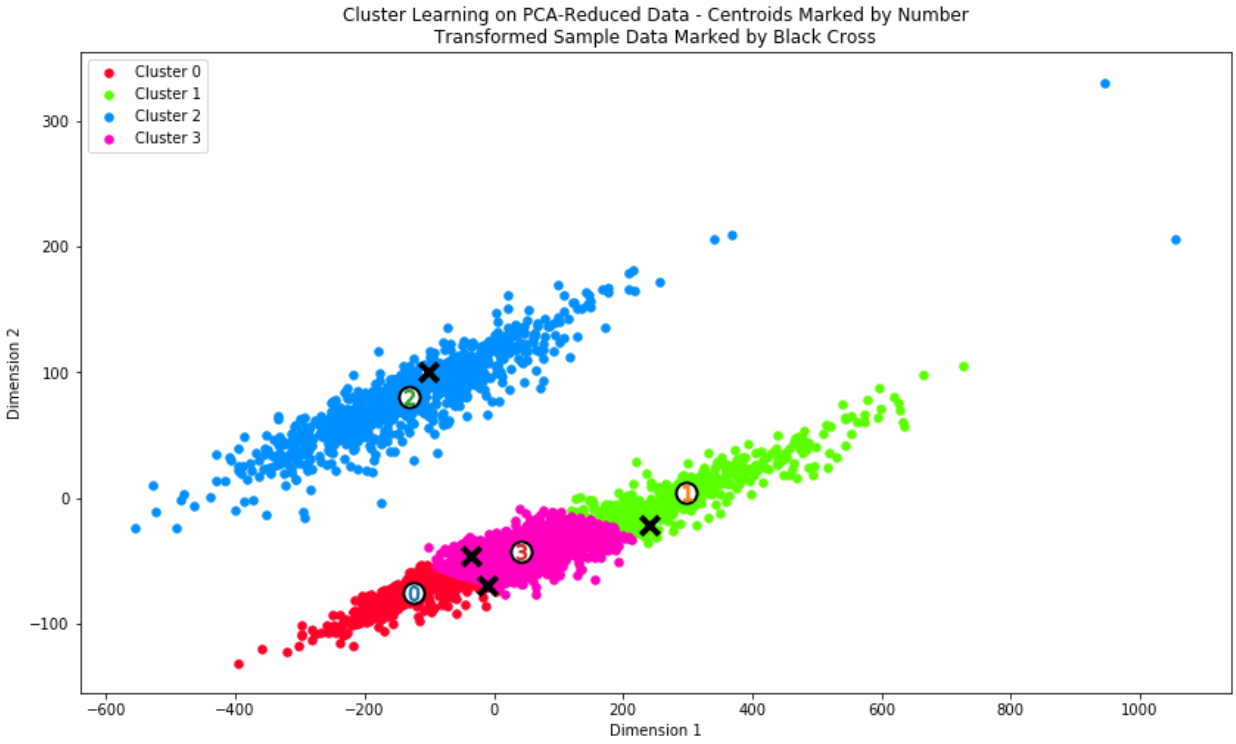


This illustrates how the 4 pitch types are clustered together when the features are reduced to 2 dimensions. We will refer to this plot once we perform our cluster analysis to see how well the algorithms label each pitch.

Section 8 – Implementation

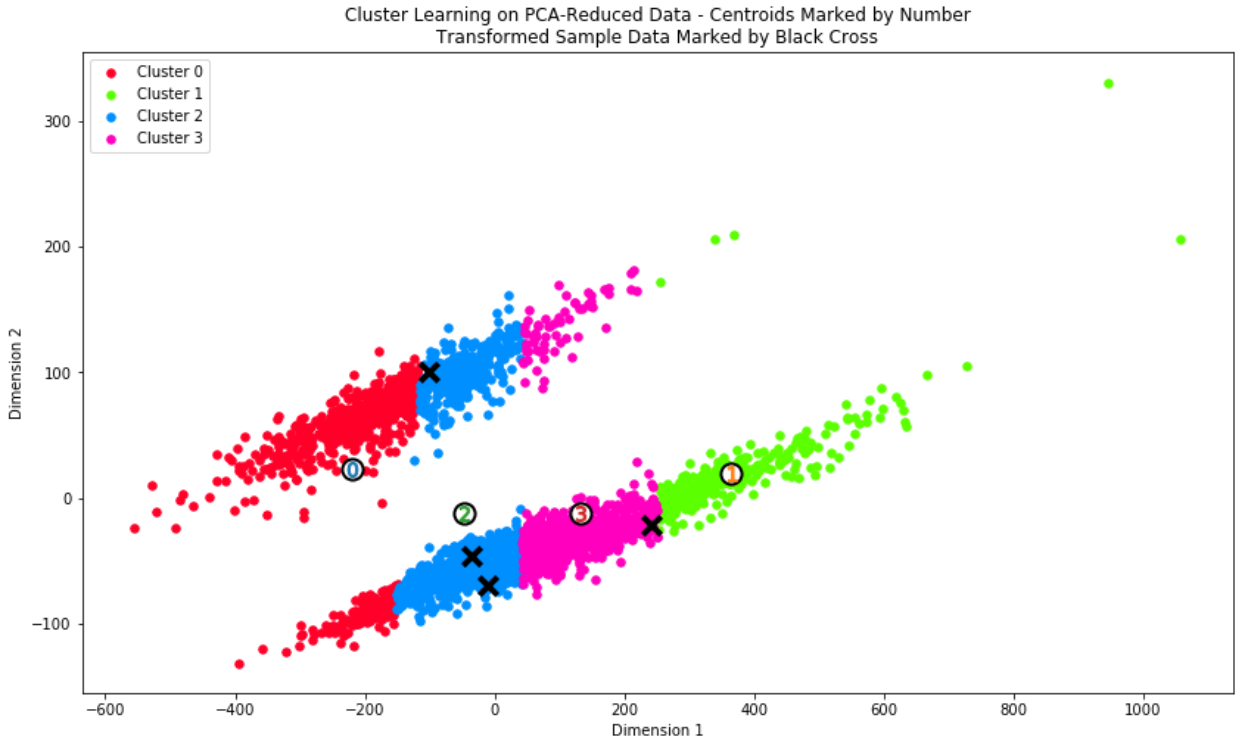
Now that the dataset has been preprocessed and examined, it is ready for the clustering algorithms. I will first run each clustering algorithm on the dataset with 4 clusters, as we already know there are 4 pitch types.

Running a Gaussian Mixture model on the reduced data produces the following plot:



Here the 4 clusters are identified along with their centers, as well as the 3 sample points identified earlier. This clustering algorithm does a decent job at labelling each pitch, with a few noted exceptions. Sliders and changeups are classified very well, however the algorithm does not differentiate between the 2 fastball types well.

Running a KMeans algorithm on the data produces the following plot:



As illustrated, KMeans does not perform as well as the GMM clustering did. The pitches are not differentiated properly, with sliders being classified into the same cluster as 4-seam fastballs, 2-seam fastballs and changeups.

For the next step in the analysis, I won't provide the algorithms with the number of clusters ahead of time. Instead, I'll will iterate the each model with a different number of clusters and calculate the Silhouette score that results. The Silhouette score, as defined by scikit-learn, is

> composed of two scores:
>
> a: The mean distance between a sample and all other points in the same class.
>
> b: The mean distance between a sample and all other points in the next nearest cluster.
>
> The Silhouette Coefficient s for a single sample is then given as:

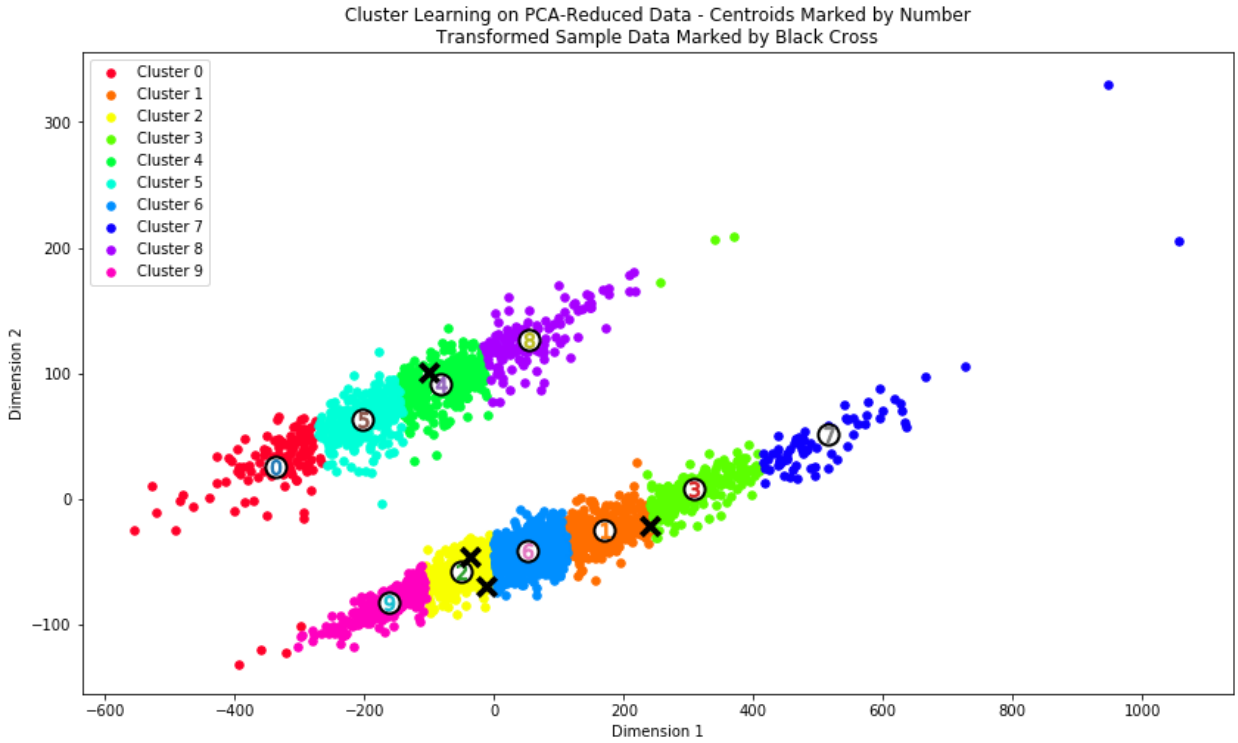
$$s = \frac{b - a}{max(a, b)}$$

This coefficient for a data point measures how similar it is to its assigned cluster from -1 (dissimilar) to 1 (similar). Calculating the mean silhouette coefficient provides for a simple scoring method of a given clustering. The closer the silhouette score is to 1, the more representative each cluster is of each data point.

First I will calculate the number of clusters with the maximum silhouette score for the Gaussian Mixture model, then I will do the same for the KMeans model. For the GMM, the optimal number of clusters calculated is 2, with a silhouette score of .449. With the number of clusters equal to 2, the GMM clustering produces the following plot:

While the silhouette score is better, this doesn't perform as well at labeling each pitch type as the model did when 4 clusters is used.

After calculating the optimal silhouette score for the KMeans algorithm, the results are 10 clusters, with a score of .508. That model produces the following plot:



Now this is interesting! According to the KMeans clustering algorithm, Chris Sale doesn't have a 4-pitch mix, he actually has a 10-pitch mix! Is it possible Chris Sale throws 10 different types of pitches?

Results

Section 9 – Model Evaluation and Validation

If we are to take the results of the optimized KMeans clusters as true, we can use the model with 10 clusters as representative of Sale's pitch repertoire. Let's explore that model a little further.

By looking at the centers of all 10 clusters, we can see what Chris Sale's true pitch mix looks like, as determined by KMeans clustering. After converting the reduced data back to the original features, a feature subset of the 10 centers look are the following:

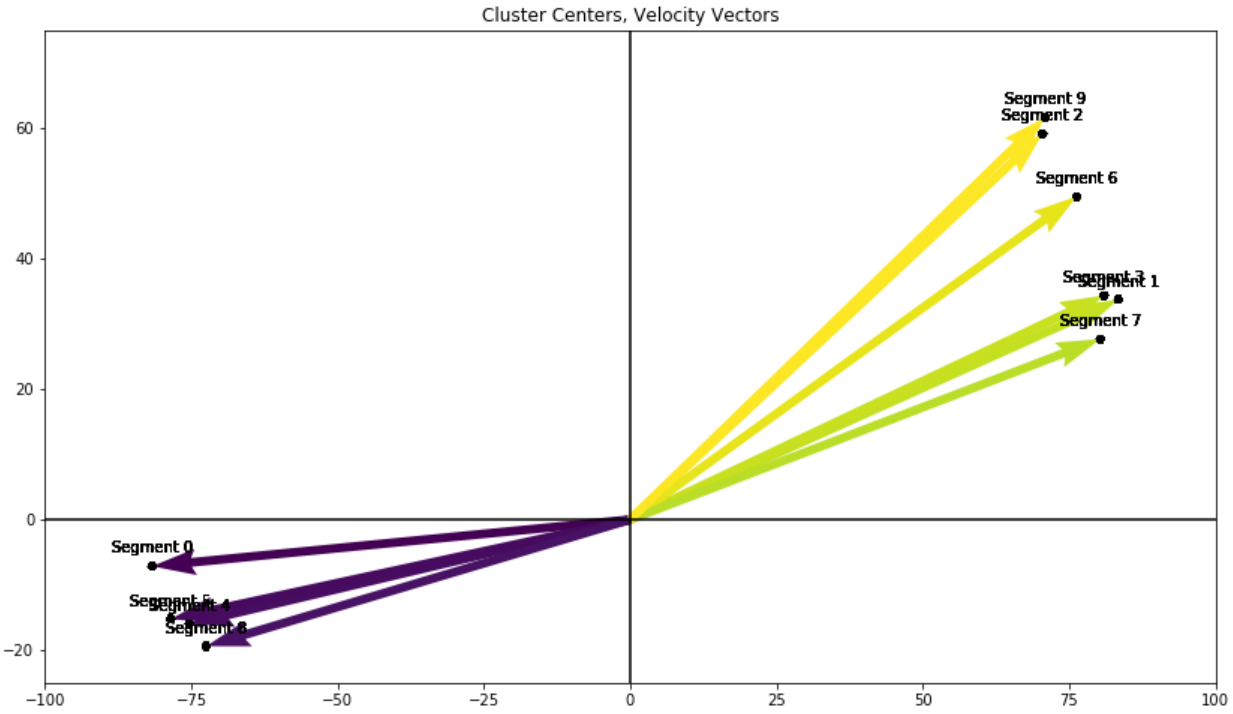
	effective_speed	release_spin_rate	release_extension	pfx_x	pfx_z	break	angle
Segment 7	85	1758	6	2	0	2	109
Segment 1	90	2111	6	1	1	2	112
Segment 3	88	1970	6	1	1	2	113
Segment 6	91	2231	6	1	1	2	123
Segment 2	92	2336	6	1	1	2	130
Segment 9	94	2449	6	1	1	2	131
Segment 0	82	2596	5	-1	0	1	275
Segment 5	80	2456	5	-1	0	1	281
Segment 4	77	2333	5	-1	0	1	282
Segment 8	75	2192	5	-1	0	1	285

For convenience, I've then sorted by angle from smallest to largest to illustrate how Chris Sale varies his pitches.

If we are to separate his pitch type into a fastball-changeup group (FF-FT-CH) and a breaking pitch group (SL), we can see that his pitches really operate on a scale as oppose to four distinct types of pitches. Segments 7, 1, 3, 6, 2 and 9 represent the scale for that first grouping of pitches, what I refer to as the fastball-changeup group. This can be interpreted as a scale going from a true changeup (85 mph at 109°) to a true 4-seam fastball (94 mph at 130°). The pitches between those two extremes are stops on that scale, with 2-seam fastballs, harder changeups and 4-seam fastballs making up that group.

The next group of cluster centers are the breaking ball variations Chris Sale throws. Again, this can be interpreted as a scale, with a harder, tighter slider representing Segment 0 (82 mph at 275°) and a softer slider with more break (75 mph at 285°) representing Segment 8.

To illustrate these 10 different pitches even further, let's plot the velocity vectors again, however instead of plotting the average velocity and break angle for each of Sale's 4 pitch types, I will plot the velocity and break angle of the 10 cluster centers.



This is Chris Sale's true pitch repertoire, as determined by KMeans. He throws one type of breaking pitch, a slider, but by varying the velocity and break of that pitch, can attack the hitter with 4 different variations of that pitch. The same is true for his fastball-changeup combination. He throws a 4-seam, 2-seam, and changeup, but by again varying the velocity and break, can turn that 3-pix mix into a 6-pitch mix.

Finally, let's go back to our samples we chose at the beginning of this process. I randomly chose 1 pitch of each type from the dataset, and we can look at the characteristics of those samples as well as what cluster they were assigned to.

	pitch_type	effective_speed	release_spin_rate	release_extension	pfx_x	pfx_z	break	angle	cluster
0	FF	93.903	2318	5.726	1.0795	1.1848	1.602832	137.66259	2
1	FT	92.712	2299	6.013	1.9656	0.6799	2.079867	109.080476	2
2	CH	85.242	2042	5.668	1.4825	0.2493	1.503315	99.545656	1
3	SL	75.803	2349	5.083	-1.3468	-0.6067	1.477144	294.250383	4

The first sample is a 4-seam fastball which was assigned to cluster 2. Cluster 2 is an interesting cluster, because it contains pitches labelled as 4-seam fastballs, 2-seam fastballs and some changeups, with a velocity of 92 mph and a break angle of 130°. Relative to the cluster center, this pitch was thrown harder and with a larger break angle.

The second sample is the 2-seam fastball, also assigned to cluster 2. With a velocity of just over 92 mph and a break angle of 109°, which is roughly as fast as the cluster center in cluster 2 however it breaks at a lower angle.

The third sample is the changeup, with a velocity of 85 mph and an angle of 99°, and it was assigned to cluster 1. The center of cluster 1 had a velocity of 90 mph and an angle of 112°, so this particular pitch was thrown with less velocity and had a larger break angle.

The final sample is the slider, which was thrown at 75 mph and had 294° of break. It was assigned to cluster 0, which had a center point with a velocity of 77 mph and a break angle of 282°, so this pitch was thrown slightly lower speed but at a slightly greater angle.

Section 9 – Justification

By the results of the KMeans clustering algorithm, Chris Sale may throw 10 different "types" of pitches, despite there being only 4 classifications in the Statcast data. It certainly seems that our clustering algorithms don't label each pitch accurately. But is there any way we can validate the labelling done by Statcast? Is it possible that the labelling isn't fully representative of the differences in the pitches Sale throws? To do this, we can look no further than a piece by sportswriter Brian MacPherson, from a story he did back in early July of this year:

> Sale didn't want to talk much about how he goes about his task after Saturday's game.
>
> "I'm just trying to go out there and win games," Sale said. "However that shakes out, it does."
>
> But others in the Red Sox clubhouse were happy to do it for him.
>
> What makes Sale exceptional is the way **he expands a three-pitch repertoire into an eight- or nine-pitch repertoire.**
>
> **"He'll add and subtract with everything, basically," Pomeranz said. "He has three really good pitches, and through changing speeds, he turns them into even more than that. It's all about getting them off-balance. They never know what to expect because everything is jumping all over the scale of speeds."**
>
> **Sale doesn't just throw a fastball. Sale throws several fastballs — a two-seam fastball he calls a batting-practice fastball for an early-count strike; a two-seam fastball in the low-90s to get weak contact; and a four-seam fastball he can ride up to 99 when he needs to, either up and away from a lefty or in on the hands of a righty.**
>
> **Sale doesn't just throw a changeup. Sale throws two changeups, and he has two different changeup grips.**
>
> **Sale doesn't even throw just a slider. Sale throws a get-me-over slider in the mid-70s; a back-door slider around 80 that elicits weak contact; and a wipeout slider around 80 with which he gets most of his swings and misses.**
>
> **"He's giving you different looks within all of the pitches themselves," Bannister said.**
>
> Just adding to and subtracting from a fastball is rare ability, to be able to pitch with the fastball at several different speeds within the same outing.

> A first-inning strikeout of Troy Tulowitzki saw Sale throw a two-seam fastball 92 mph and get a foul ball on it — and then come back with a four-seam fastball at 96 to which Tulowitzki had no chance to catch up.
>
> Bannister pitched for five seasons in the major leagues but has been around the game his entire life. He said he's seen only two other pitchers who can vary speeds on their fastballs to the degree Sale does — Detroit's Justin Verlander and Arizona's Zack Greinke.
>
> "They can comfortably pitch 92-95 and then reach the upper 90s at will," he said. "It's a special ability. Most people try to do that, but you just can't."
>
> "He can be throwing 93 — and then here comes 98," Pomeranz said. "That's what makes him even more special. He'll throw a slider at 76 and then he'll throw one at 82 and then he'll throw a changeup a little slower and then he'll throw a BP fastball — and if you start sitting on soft, all of a sudden, here comes 99."
>
> The featured pitch for Sale on Saturday was that slider. Sale threw nearly 50 sliders against the Blue Jays, ranging from 75-83 mph, and he elicited nine swings and misses. It wasn't until the fourth inning that a Toronto hitter put a Sale slider in play.
>
> One of his most impressive sequences came when he struck out Toronto second baseman Ryan Goins on three pitches, all sliders. He threw one at 79 for a called strike. He threw one at 75 that got a foul ball. He finished Goins off with a slider at 80 in the dirt.
>
> "It was running like this, all the way across," said Pomeranz, holding his hands about two feet apart.

Source: http://www.telegram.com/sports/20170701/red-sox-chris-sale-excels-at-turning-three-pitches-into-arsenal

If we are to believe Sale's teammates, due to the way he can vary the speed and break of each "type" of pitch he throws, there is enough variation that he may as well be throwing 10 different pitches!

Conclusion

Section 10 – Reflection

As initially stated at the outset of this project, the problem I attempted to solve is "can unsupervised Machine Learning algorithms correctly classify pitches thrown by a Major League Pitcher into their proper pitch type?" In a way, this problem was both unable and able to be solved. The clustering algorithms used were not able to correctly label each pitch the same way Statcast does. However through the clustering analysis, I was able to show that the pitches Sale throws have enough variation in them that there may be 10 different "types" of pitches in his arsenal, which in turn was justified by an article digging into the way he uses his 4 different pitch types. Depending on which labeling system you want to believe is "correct" – Statcast's 4 types or the 9+ types indicated by the article – the KMeans clustering model could be interpreted as either correctly or incorrectly clustering each pitch.

Section 11 – Improvement

In my attempt to look for ways to improve the efficacy of the KMeans algorithm, there is a single feature which I believe could explain variation in pitch type extremely well which is not currently tracked by Statcast. Statcast records the speed, spin rate and break of each pitch which are the features I used to created the clusters. However I would be interested to see the results of the algorithm if axis of rotation, or rotational direction, could be incorporated as a feature. My reasoning for this is because axis of rotation will differ for 2 pitches that may have similar speeds, spin rates and break. The addition of this feature could provide more distinct clusters and greater insight into Chris Sale's true pitch repertoire.