Predicting Major League Baseball Pitch Type

# Final Report by Louie Ligon

**Problem Statement**

As Hank Aaron once said, “Guessing what the pitcher is going to throw is 80 percent of being a successful hitter. The other 20 percent is just execution.” The core event that occurs in the sport of baseball is a pitch of the baseball from pitcher to batter. The pitcher is trying to throw a pitch that does not result in the batter getting on base while the batter aims to accomplish exactly that. All methods of getting on base require a pitch. Understanding the nature of pitches can benefit both the pitcher and batter. From the batter’s view, predicting pitch type will increase the likelihood of success with knowledge of pitch speed and location.

**Introduction**

Major League Baseball (MLB) is possibly the most sophisticated sports organization in the world when it comes to data analysis for the sport. While descriptive statistics have always been a source of performance measurement for players in the MLB, modern day technology introduced an “arms race” of data analysis to the sport. As of 2015, all 30 MLB teams have their stadiums equipped with Statcast infrastructure. Statcast is a camera based, highly accurate tool that measures player movements and athletic abilities.

Statcast has greatly increased the amount of information available for evaluating player performance in the MLB. The scope of this project is to focus on pitch-level data to better predict what type of pitch a batter should expect. When armed with a confident expectation of a specific pitch, this greatly increases the batter’s chances of getting on base compared to uncertainty in the impending pitch type. Pitched ball speeds can vary greatly and accurately predicting the pitch type associated with speed will benefit the batter and offense.

Statcast data is available to the public and made easily accessible with the Python package pybaseball. With the help of MLB statcast and pybaseball, the information required to predict pitches should be available in the source data. As a batter in the MLB, the problem of uncertainty in future pitch types is one that has traditionally been solved with intuition or player intelligence. Now, we aim to arm the batter with a statistically supported expectation of pitch type.

**Data Wrangling**

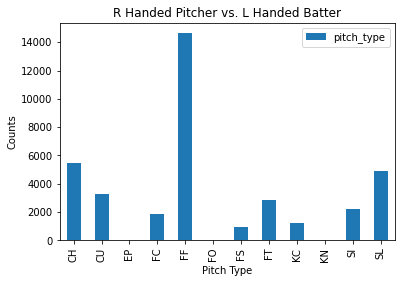
I started by loading just over 100k observations of pitch-level data with 90 features using the Python package Pybaseball. The data is Statcast data from July in the 2019 season. Many of the data fields were from an old, deprecated version of Statcast previously used by Major League Baseball. Some values for pitch type were missing. I addressed both by removing records with a missing pitch type value and old features that were no longer used.

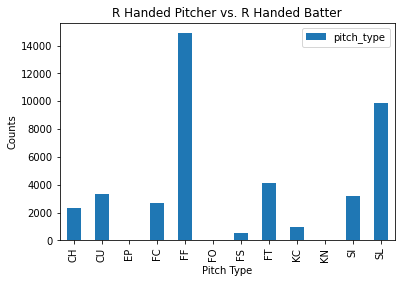
After reviewing some of the features in greater detail, combined with a review of the documentation explaining each feature, I decided to drop almost 30 features. This left me with 62 features and over 100k pitches. Moving into the next step, I needed to determine the methodology for prediction. Specifically, I needed to decide if I should arrange the data to reference prior pitches, in a respective at-bat, so that predictions consider more than treating each pitch as an independent event.

**Exploratory Data Analysis & Initial Findings**

EDA began with 61 feature variables and the target variable of 'pitch\_type'. Given the high dimensionality of the data, I was focused on exploring features to potentially drop. Many features are post-pitch data and could not be used to provide meaningful predictions. There was also a need to transform the data to discover pitch sequences within a single at-bat. There was an excessive number of feature variables in our dataset based on the Statcast data dictionary. After reviewing features in detail, another 38 were dropped leaving 23 features and 1 target variable.

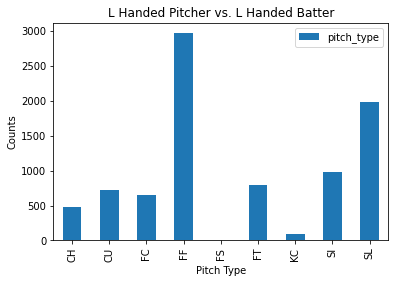
One key finding was related to the position of the batter in relation to the throwing hand of the pitcher.

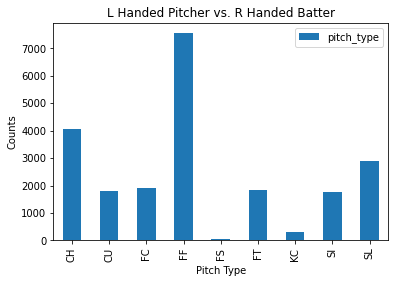




Right-handed (RH) pitchers are throwing fastballs (FF) the most to both sides of the plate, but there's a clear difference between the next most frequent pitch types. A RH batter receives a lot more sliders (SL) than a LH batter while a LH batter will see a changeup (CH) more than any other pitch except fastballs.

What about left-handed pitchers?





Not surprisingly, the reciprocal is occurring with LH pitchers. This is due to the nature of how pitches move. A slider breaks down and away from a batter of the same orientation which is one of the more desired pitch locations to make a batter miss the ball. Changeups (CH) are again most frequent after fastballs (FF) when the batter is opposite hand of the pitcher.

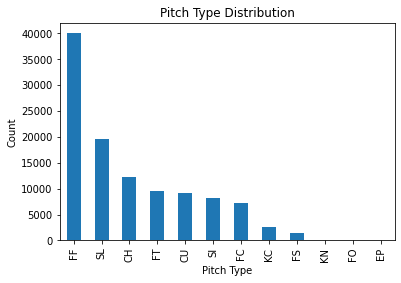
The plot below is a 2D representation of the catcher's view looking back at the pitcher's mound. Each plotted point represents the release point of a given pitch from the pitcher's hand. Remember that a right-handed pitcher is facing the catcher, so their release points will appear on the left while the blue points are released from a left-handed pitcher.

Chart, map, scatter chart

Description automatically generated

Notice the small clusters on each side. These represent the release points of unconventional pitchers known as sidearm pitchers.

Naturally, I wanted to get a sense of pitch type distribution.



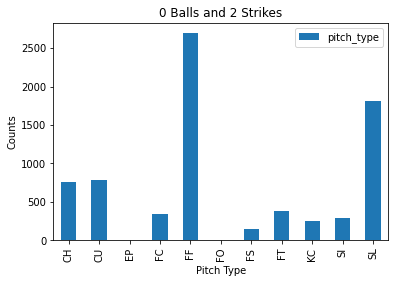
FF is the most frequent pitch type with 36.3% of all pitches. SL and CH are the only other types with more than 10% at 17.8% and 11.2% respectively. FT and FF are both fastballs and combine for 45% of all pitches. This will be useful in testing our model as we can compare it to simply guessing fastball for every pitch.

A couple significant features expected to affect the pitch type is the count of balls and strikes. Recall that for any given pitch, there will never be more than 3 balls or 2 strikes. A 4th ball results in a base on balls and the batter goes to 1st base. A 3rd strike results in a strike out and will retire the batter and add 1 to the count of outs. We can look at 2 extreme examples:

Chart

Description automatically generated with low confidence

All pitches where the count is 3 balls and 0 strikes (above) and all pitches where the count is 0 balls and 2 strikes (below).



I needed to transform the data so I could get a sense of the sequences of pitch types a batter will see. I discovered a very similar relative distribution of pitch type for any of the pitches numbered 1 through 10. Based on this information, in addition to the high count of unique combinations, I moved forward with treating each pitch type as independent from the previous pitch type. This assumption made it more practical for working with the data for modeling.

I started the EDA process by loading our previously wrangled data containing over 60 features and 110k observations. Exploration of all features resulted in 38 of the features being dropped. I also dropped a few hundred observations that contained at least 1 null value in 4 of the remaining features. The dropped features were almost entirely due to the nature of our prediction interest. Rather than simply demonstrating predictive power, I wanted to predict pitch types with the information available in between pitches and with enough time to hypothetically communicate these predictions to a batter. This means that any information gained from the pitch of interest cannot be used. For example, we can't use the speed, spin, or position of a pitch to predict the pitch with our intent.

During exploration of which arm a pitcher throws with compared to the position of the batter, we discovered meaningful differences in the pitch type distributions for same hand vs. opposite hand combinations. Specifically, when a right-handed pitcher throws to a right-handed batter, we see a higher count of sliders and lower count of changeups. The same is true for left-handed pitchers throwing to left-handed batters. We also found the count of balls, strikes, and outs to have minimal effect on the pitch type distributions except for extreme cases such as 3 balls, 0 strikes and 0 balls, 2 strikes. Finally, I transformed the data to allow for discovery of pitch sequences within at-bats. The very high count of unique combinations, combined with consistent pitch type distributions for any given pitch number in sequence, allowed me comfort in moving forward with pitches remaining as independent events.

**Modeling & Recommendations**

After having wrangled, explored, and processed Statcast data, I finally got to explore some machine learning models in attempt to predict baseball pitch types. I decided to consider 3 different datasets for modeling:

* no\_pitchers - All pitches without pitcher reference
* pitchers - All pitches with reference to top 5 pitchers based on pitch count
* first\_pitch - Only the first pitch in each at-bat without pitcher reference

I wanted to explore modeling various scenarios with the data above. I began by setting a baseline from predicting all pitches as the most common pitch type, FF. Our modeling efforts included scaling the data, creating a train/test split, and evaluating performance metrics. I also completed tuning of some hyperparameters relevant to each model. After creating a baseline prediction using sklearn's DummyClassifier, I modeled pitch type outcomes with a Decision Tree and Random Forest.

I began the modeling efforts with the ideal of accurately predicting MLB pitch types. At a minimum, I hoped to determine features of importance when attempting to predict pitch type. Any insights gained could help better understand what pitch might be coming next during a major league at-bat. Through this process, I have found some interesting information that guides the steps ahead in better prediction of pitch types.

In the beginning, I used a DummyClassifier to get a sense of how poorly a model would perform if it predicted FF for every first pitch thrown at a batter. I saw improvement in model performance if we used a default Decision Tree for the first pitch data. I visualized this tree to find the model valued pitch hand and batter stance in addition to the progression of the at-bat and game. As I introduced more pitches and another feature for which pitcher was throwing, the models improved even more.

Finally, after tuning hyperparameters for the Decision Tree, I went on to do the same for a Random Forest pipeline. Our Random Forest found the features for pitch\_number and at\_bat number to be of greatest importance. In the end, the model performance of the forest did not meaningfully outperform the best tree. I did gain the insight to look more into the count of balls and strikes and how that might affect pitch type. The models here have not given us a secret weapon in knowing what to expect when a pitch is being thrown, but they have provided meaningful insights into what to consider when looking to predict pitch type.

**Future Work**

Through exploration and modeling of Statcast data, I have discovered meaningful insights into predicting baseball pitch types. I considered 3 arrangements of data in my efforts and found the most predictive set to include pitcher reference with all pitches. There are most likely additional arrangements that could help improve our models and better our predictions of Major League Baseball pitch type.

After exploring distributions of pitch type, for each count of pitch, it was determined that we could proceed with pitches as independent events. I believe it would be valuable in future efforts to confirm this by arranging pitch data for each at-bat. There could be patterns that become evident when considering each of the previous pitches in each at-bat. This would require data manipulation beyond the scope of this project.

One method of simplification for prediction would be to combine specific pitch types into 3 categories. Rather than trying to predict 1 of 12 outcomes, we could aim to predict 1 of 3. The categories of interest would be fastballs, breaking balls, and off-speed pitches. These categories would address the two advantages of knowing an impending pitch type: timing and location. Another way to gain insight on location would be to add predictions to pitch zone location. There are 9 standard locations within a strike zone and pitches typically change position based on the pitch type.

Finally, I would want to focus the prediction efforts on specific teams and individual pitchers. Each team, and their corresponding pitchers, will have a subset of the pitches we explored in this project. Most pitchers have 3-5 common pitches they throw and those vary by pitcher. The last area of interest is to look at combined data from specific batter-pitcher matchups. In addition to the benefit of considering a single pitcher’s pitch types, we can also see how he might throw targeted pitches depending on the batter.