Pitch Prediction

Predicting the Next Pitch-Type Thrown by Major League Baseball (MLB) Starting Pitchers Using Machine Learning

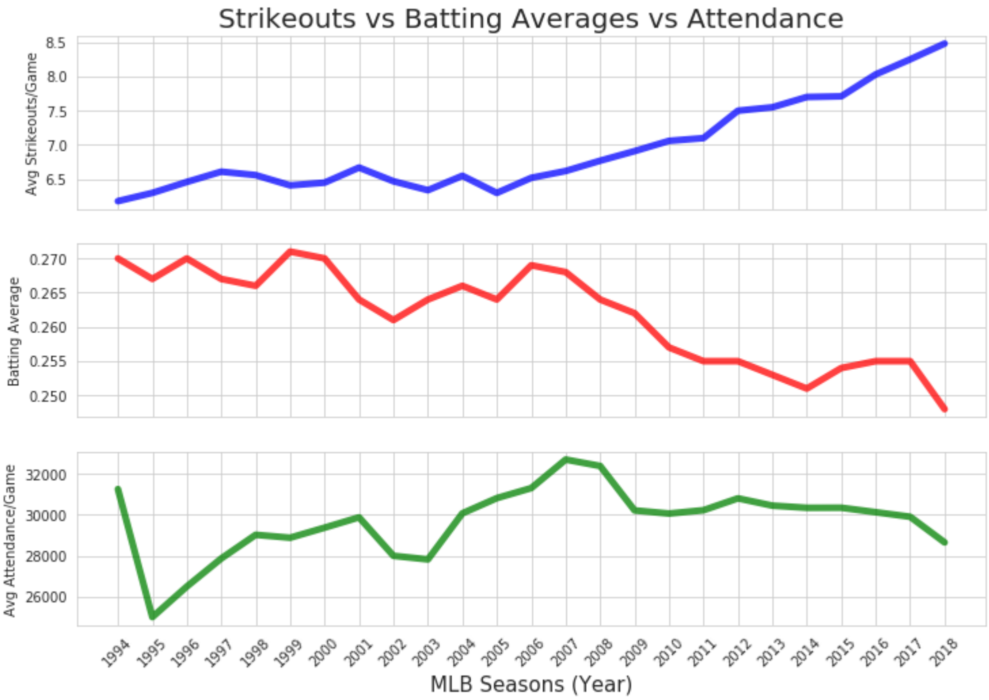
Capstone I Project: Final Report
Springboard Data Science Career Track

Mark Rojas

January 2019

Background

According to Ted Williams, one of the greatest hitters of all time, "*Hitting is the single most difficult thing to do in a sport*". Having less than 400 milliseconds to hit a fastball traveling 95 mph, it should be humanly impossible to hit a baseball. Even the best hitters today have batting averages less than .350 in a season. Since 1994, the average number of hits per game have decreased while the number of strikeouts per game has increased [1]. In addition, aside from the MLB strike in 1995 which caused the attendance to drop significantly then rise again over the next 3 seasons, the average attendance per game is gradually dropping. This can impact nearby restaurants and bars that depend on fans who frequent their establishments during the season.



Advancements in training methods, preparation, and use of technology appear to have benefited pitchers more so than the hitters. By facing a pitcher multiple times in a game and throughout their career, hitters can utilize historical data and past experiences to guesstimate the next pitch. However, this does not take into account the current game-time variables that impact the pitchers decision making, and as the great Yogi Berra once stated, "*You can't think and hit at the same time*".

1. https://www.baseball-reference.com/leagues/MLB/bat.shtml

Problem

Because of this, to assist their hitters, teams have resorted to stealing signs between the catcher and pitcher as well as observing the pitcher for telltale signs. Like in poker, telltale signs can present the coming pitch, giving the hitter the upper hand. For example, in the 2017 Major League Baseball (MLB) World Series between the Houston Astros and Los Angeles Dodgers, the Dodgers starting pitcher for game 3 and the decisive game 7 was Yu Darvish. In both games, Darvish was quickly removed in the 2nd inning after giving up 4 and 5 runs, respectively. Following the series in which the Astros won the best of seven, it was later revealed in a **Sports Illustrated article** [2] that Darvish was "tipping" his pitches, meaning he was unknowingly telegraphing what pitch he was going to throw next.

The problem is that not every pitcher "tips" their pitches and stealing signs is extremely frowned upon and usually leads to retaliation through hit batters. If we simply go by the chart above, specifically from 2001 and on, we may assume that batting averages have a greater impact on attendance than strikeouts. Therefore, we can claim that there is a need to improve batting averages to increase profits for the organization and establishments that benefit from fans attending the games.

[2] https://www.si.com/mlb/2017/12/11/dodgers-yu-darvish-tipped-pitches-world-series-astros

Client

MLB coaching staff and players.
Note: It is understood that should it ever be possible to accurately predict what a pitcher is going to throw so that it benefits the hitter, it is likely that the pitcher will also utilize this information to adjust their pitch selections.

Goal

The goal of this project is to 1) identify features / events of a game that have the greatest impact in determining the type of pitch a pitcher decides to throw and 2) see if it is possible to construct a model that can predict the next pitch-type with > 70% accuracy.

Data

Data Overview

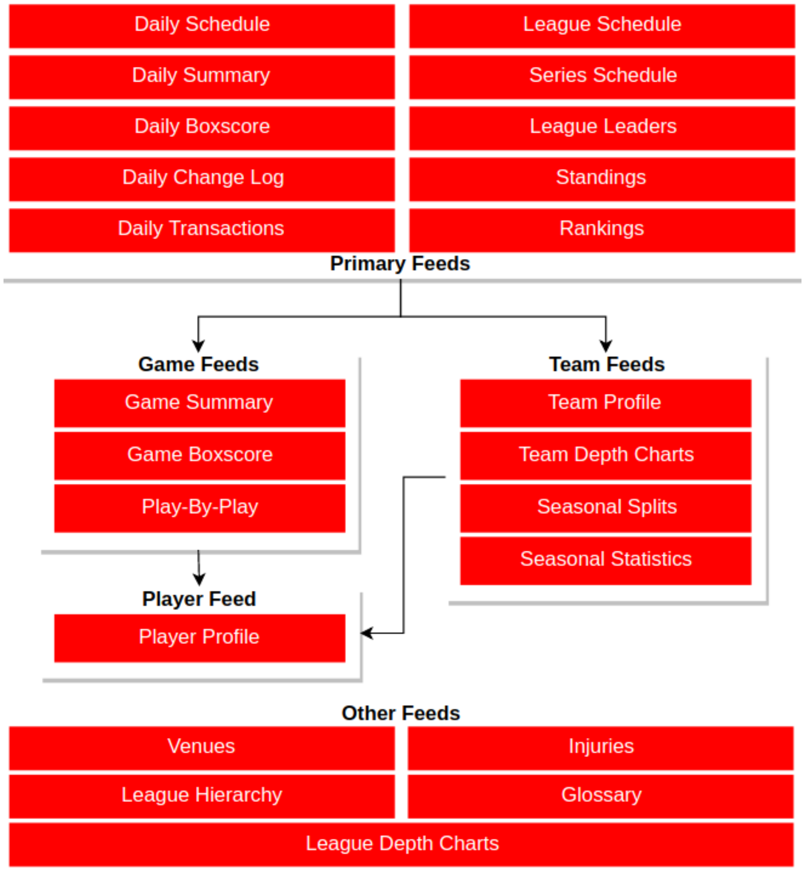
In the 2006 postseason (playoffs), Major League Baseball (MLB) first began using a camera-based system to track the trajectory, speed, spin, break, and location of a pitched ball, called **PITCH*f/x***. This information makes it possible to determine the "pitch type" (e.g., Fastball, Cutter, Change-up). Early models for classifying pitch types, however, were not very accurate for pitches similar in speed and break and the data from early years includes many misclassified pitch types. Since 2015, a more advanced system (**Statcast**) is being used which integrates doppler radar with high definition video to track all aspects of a game, including pitches, hits, and players. For the 2017 season, **Trackman**, a component of **Statcast**, officially replaced the previous **PITCH*f/x*** system.

Data Collection

The **Statcast** / **Trackman** pitch data was collected using an Application Programming Interface (**API**) provided by **Sportradar.com**. Access to the MLB feeds through standard **HTTP Requests** using **Python 3.6** with a **version 6.5 API Key** includes data for leagues, conferences, teams, games, and players in **JSON** format.

<u>Sportradar.com Feeds used for project in JSON format:</u>

1. **[Primary Feed] - League Schedule**: Date, time, and location for every game.
2. **[Team Feed] - Team Profile**: Top-level team information including all players currently on the 25 man roster, 40 man roster, or expected to join the team.
3. **[Game Feed] - Play-By-Play**: Detailed real-time information on every pitch and game event.
4. **[Other Feed] - Glossary**: Full text descriptions for pitch ids, player status ids, outcome ids, position ids, and game status ids.



The **PITCH*f/x*** system was used from 2006 – 2014 before changing to **Statcast** in 2015. Due to this, I did not consider pitch data prior to 2015 for this project. Instead, I collected data from the 2016-2018 seasons. Also, because the **API** is fee-based, I limited the number of starting pitchers of interest to **twenty** (**20**) who were considered “top” pitchers during the **2016**, **2017**, or **2018** seasons.
The list of pitchers of interest are:

```
pitchers =  ['aaron_nola', 'carlos_carrasco', 'carlos_martinez',
             'chris_archer', 'chris_sale', 'clayton_kershaw', 'corey_kluber',
             'dallas_keuchel', 'david_price', 'gerrit_cole', 'jacob_degrom',
             'jake_arrieta', 'jose_quintana', 'marcus_stroman',
             'justin_verlander', 'max_scherzer', 'michael_fulmer',
             'stephen_strasburg', 'yu_darvish',  'zack_greinke']
```

I ensured that approximately half of the pitchers in this list play for the **National League** while the other half play for the **American League** with some switching teams during the off-season or getting traded mid-season. It is important to note that National League pitchers are required to bat while American League pitchers are not. The American League uses designated hitters to hit in-place of the pitcher. This may have an impact on pitch selection during the game.

Data Cleaning

Cleaning the play-by-play data collected from Sportradar.com was the most challenging and time consuming process, yet it was also the most informative and educational experience gained from this project. The play-by-play JSON files consisted of detailed real-time information of every pitch and game event, however, data redundancy and unnecessary information such as jersey numbers, lineups, and warmups had to be removed.

Converting JSON files to pandas DataFrame(s) was complicated by deeply nested *lists of dictionaries* and *dictionaries of dictionaries*. Using Python, I created functions to iterate through the list of pitchers of interest for each year (2016-2018) and normalize JSON files, convert to a DataFrame, de-nest nested values, drop and rename columns, handle null values and merge cleaned data into a single CSV file for each pitcher.

For cases where **null values for pitch speeds and pitch types** were identified, null values for pitch speeds, were replaced with the mean() pitch speed for that game. To address pitch types, intentional walks were labeled as ‘IB’ for intentional ball. For other cases, the pitch type was filled in with ‘UN’ for unknown. If the majority or all of the pitch speeds or types were null, data for the entire game were removed from the dataset.

Data Formatting

Once the collection and cleaning of the data was complete, it was time to perform data wrangling. In other words, transform and map data from the original "raw" data form into another format with the intent of making it more appropriate and valuable for downstream analysis, aka Machine Learning. To achieve this, it was first important to identify variables/features that significantly impact a pitcher's pitch selection. These variables can be separated into two groups:

Past Events

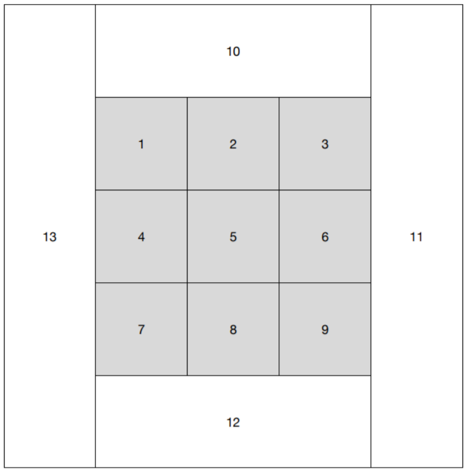
Previous pitch type
Previous pitch velocity
Previous pitch location and movement (as pitch crosses home plate)
Previous pitch outcome (e.g., hit, strike, ball, strikeout, homerun)

Current Game Situation(s)

Current Inning (total of 9 innings in standard game, excluding extra innings)
Pitch count (number of pitches pitcher has thrown to batter, inning, and game)
Hitter count (number of balls and strikes during at-bat)
Score
Runners on base
Weather conditions

While it is important to consider all variables that may impact the pitch type selection, not all data is relevant and due to time constraints, I selected only a handful of variables to utilize in this project. Should time allow and if an additional feature proves statistically significant to analysis, I will expand the number of variables considered in the machine learning approach(es). As of now, the following **Past Events** variables will be considered:

1. Pitch Types (10 different pitch types: 7 are actual pitches, 3 are not)
2. Pitch Velocity (Integer value)
3. Pitch Outcomes (82 different possible outcomes, but will only consider top 15)
4. Pitch Location (13 zones as the pitch crosses home plate)
   a. The strike zone is represented by zones 1-9



Pitch Location Zones

Pitch movement will be excluded so as to not over-complicate the project by adding additional dimensions to the equation. For **Current Game Situations**, the following variables will be considered in analysis:

5. Inning
6. Pitch Count
7. Hitter Count
8. Score
9. Runners on base

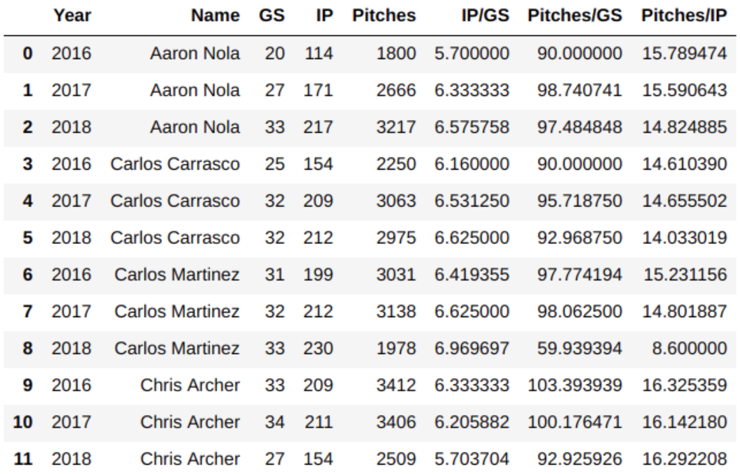
Weather conditions, which can play an important role in pitch selection, will be excluded as well. This is because this information is only available for 2018 and would require more time than currently available to trace past weather conditions during date and time game was played.

For pitch types, types which are significantly similar in speed and movement were grouped together. For example, Sinkers are grouped with Fastballs, Screwballs are grouped with Curveballs, and Forkballs are grouped with Splitters. Overall, we end up with the following list of pitch types:

FB = Fastball	CB = Curveball	CH = Changeup	SP = Splitter	IB = Intentional Ball
CT = Cutter	SL = Slider	UN = Unknown	PI = Pitchout	KN = Knuckleball

By aggregating the data for each pitcher per year, I was able to compute the number of games started (GS), innings pitched (IP), pitches thrown (Pitches), and their ratios as seen in the plot below:

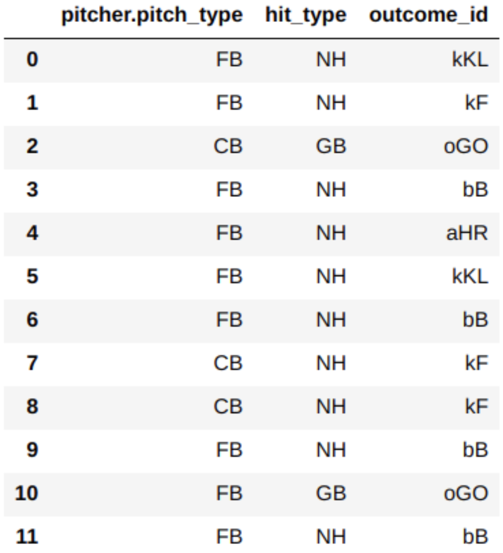
Pitching Stats / Year



	Year	Name	GS	IP	Pitches	IP/GS	Pitches/GS	Pitches/IP
0	2016	Aaron Nola	20	114	1800	5.700000	90.000000	15.789474
1	2017	Aaron Nola	27	171	2666	6.333333	98.740741	15.590643
2	2018	Aaron Nola	33	217	3217	6.575758	97.484848	14.824885
3	2016	Carlos Carrasco	25	154	2250	6.160000	90.000000	14.610390
4	2017	Carlos Carrasco	32	209	3063	6.531250	95.718750	14.655502
5	2018	Carlos Carrasco	32	212	2975	6.625000	92.968750	14.033019
6	2016	Carlos Martinez	31	199	3031	6.419355	97.774194	15.231156
7	2017	Carlos Martinez	32	212	3138	6.625000	98.062500	14.801887
8	2018	Carlos Martinez	33	230	1978	6.969697	59.939394	8.600000
9	2016	Chris Archer	33	209	3412	6.333333	103.393939	16.325359
10	2017	Chris Archer	34	211	3406	6.205882	100.176471	16.142180
11	2018	Chris Archer	27	154	2509	5.703704	92.925926	16.292208

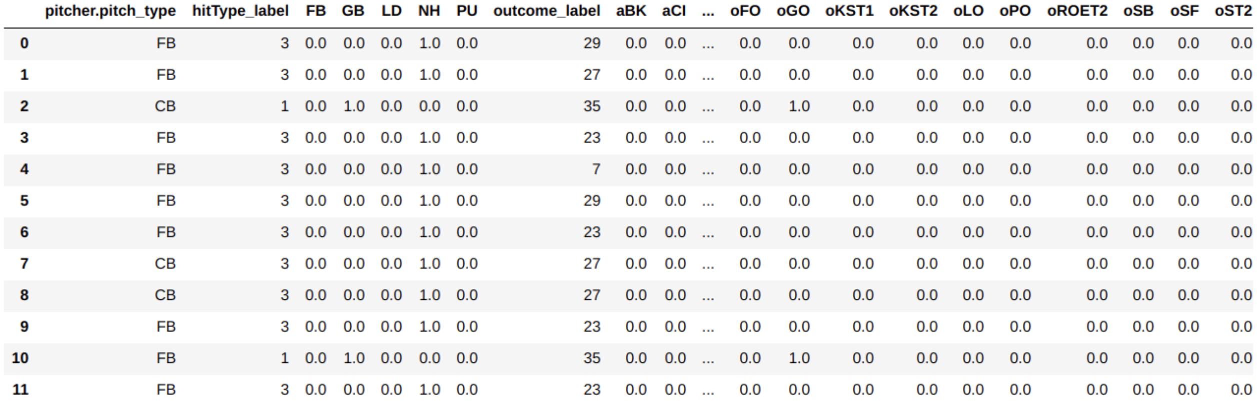
A majority of the pitch data is ‘non-ordinal’ / ‘categorical’, such as the Hit Type and Outcome ID’s, so it was important that I utilize **one-hot encoding** to format the data so that it could be used in downstream analysis. Below is an example of format conversion:

BEFORE ENCODING:



	pitcher.pitch_type	hit_type	outcome_id
0	FB	NH	kKL
1	FB	NH	kF
2	CB	GB	oGO
3	FB	NH	bB
4	FB	NH	aHR
5	FB	NH	kKL
6	FB	NH	bB
7	CB	NH	kF
8	CB	NH	kF
9	FB	NH	bB
10	FB	GB	oGO
11	FB	NH	bB

AFTER ENCODING:



	pitcher.pitch_type	hitType_label	FB	GB	LD	NH	PU	outcome_label	aBK	aCI	...	oFO	oGO	oKST1	oKST2	oLO	oPO	oROET2	oSB	oSF	oST2
0	FB	3	0.0	0.0	0.0	1.0	0.0	29	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	FB	3	0.0	0.0	0.0	1.0	0.0	27	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	CB	1	0.0	1.0	0.0	0.0	0.0	35	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	FB	3	0.0	0.0	0.0	1.0	0.0	23	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	FB	3	0.0	0.0	0.0	1.0	0.0	7	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	FB	3	0.0	0.0	0.0	1.0	0.0	29	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	FB	3	0.0	0.0	0.0	1.0	0.0	23	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	CB	3	0.0	0.0	0.0	1.0	0.0	27	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	CB	3	0.0	0.0	0.0	1.0	0.0	27	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	FB	3	0.0	0.0	0.0	1.0	0.0	23	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	FB	1	0.0	1.0	0.0	0.0	0.0	35	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	FB	3	0.0	0.0	0.0	1.0	0.0	23	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

One concern is that by using **one-hot encoding** to convert Outcome ID’s, we essentially added an additional 81 columns (**features**) to consider for each row (**observation**). Because only 1 outcome can occur per pitch, each observation will only have have one outcome with a **1:True value** in the **feature vector**. It may be necessary to exclude outcome’s which are not significant to predicting pitch types or are rare occurrences.

Exploratory Data Analysis

Overview

Using the aggregated data, we observe from the “**Pitching Stats / Year**” table above in Data Formatting section that when healthy,

- **starting pitchers average about 30 starts per year and pitch in approximately 200 innings per year**.

Not often, a starting pitcher completes a game (pitches 9 innings) and most make it only to the 6th or 7th inning before getting pulled for a relief pitcher. We can also expect,

- **starting pitchers to pitch around 100 pitches a game with an average of 15 pitches per inning**.

To ensure that the 20 selected pitcher's of interest (POI) are a good representation of all pitchers in the MLB, I opted to collect pre-aggregated statistics from **https://www.baseball-reference.com/** for all starting pitchers for the past 9 regular seasons (2010-2018), and compare them to the selected 20 POI's. The objective was to show that my 20 selected POI's and other MLB Starters average approximately the same number of **Innings pitched per game**, **Pitches thrown per game**, and **Pitches thrown per inning**. Results shown in **Figure 1**, indicate that the,

- **number of Innings Pitched and Pitches Thrown by my 20 Selected Pitchers of Interest are comparable to those of all starting MLB pitchers for 2016, 2017, and 2018.**

Also, the line plot indicates that the 20 POI's tend to throw more pitches per game, however, this could also be due to them also pitching in slightly more innings per game on average.

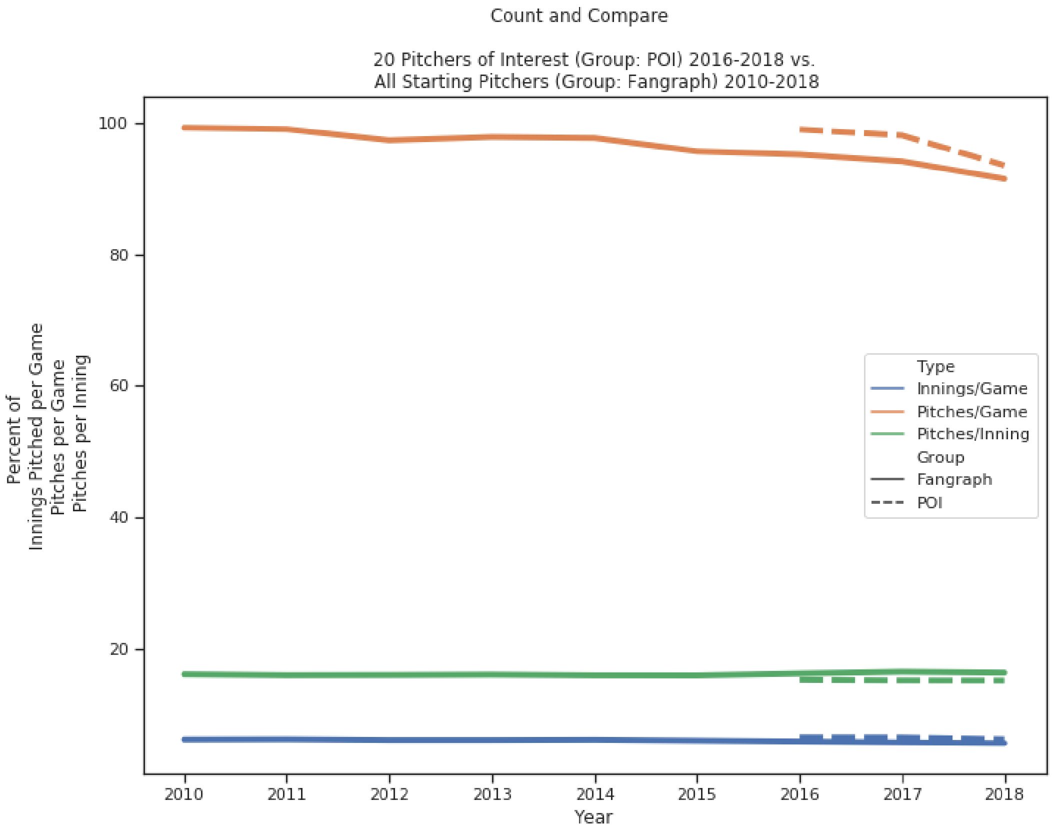


Figure 1: *20 MLB Starting Pitcher's of Interest Statistics vs. All MLB Starting Pitchers*

To identify possible trends in the data, I first looked at the percent of pitch types thrown per year. I also compared the 20 POI's against the other MLB Starting Pitchers to see if the percent of pitch types were similar. Results shown in Figure 2.

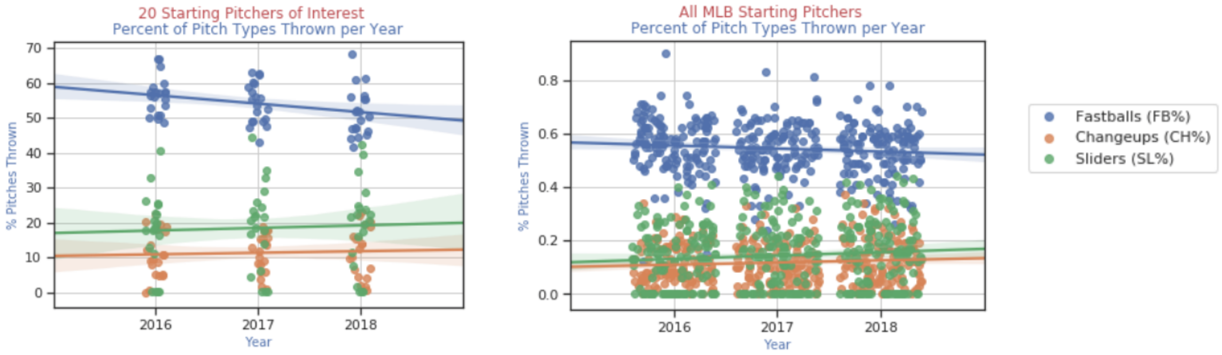


Figure 2: *Percent of Pitch Types Thrown per Year*

What we observe for both groups is the percentage of **Fastballs (FB) appear to be decreasing over the past 3 years (2016-2018)** and the **percentage of Sliders (SL) and Changeups (CH), however, appear to be increasing** over the same time span. It is possible that this trend is due to changes/improvements in the pitch classification system. Sliders are breaking pitches that are thrown with less velocity than Fastballs but faster than Curveballs. It is possible that Fastballs and/or Curveballs are now being more accurately classified as Sliders, thus the increase.

Finally, I wanted to also look at the percent frequency of pitch types per inning. Because Fastballs with a higher velocity are more effective at striking out a hitter it makes sense that a pitcher may decrease the number of times they throw a Fastball as the game progresses due to arm fatigue. Figure 3, confirms,

- **the percent of Fastballs (FB) thrown decreases as the game progresses.**
- **not expected was that the decrease happens rapidly from innings 1-3 and leveling off around 50% from innings 4-6.**

There is a little jump at inning 9 which actually makes sense considering if the starting pitcher makes it to this point, they are attempting to complete the game and are probably leading the game in runs, thus a good reason for the pitcher to give it all they got to complete the game.

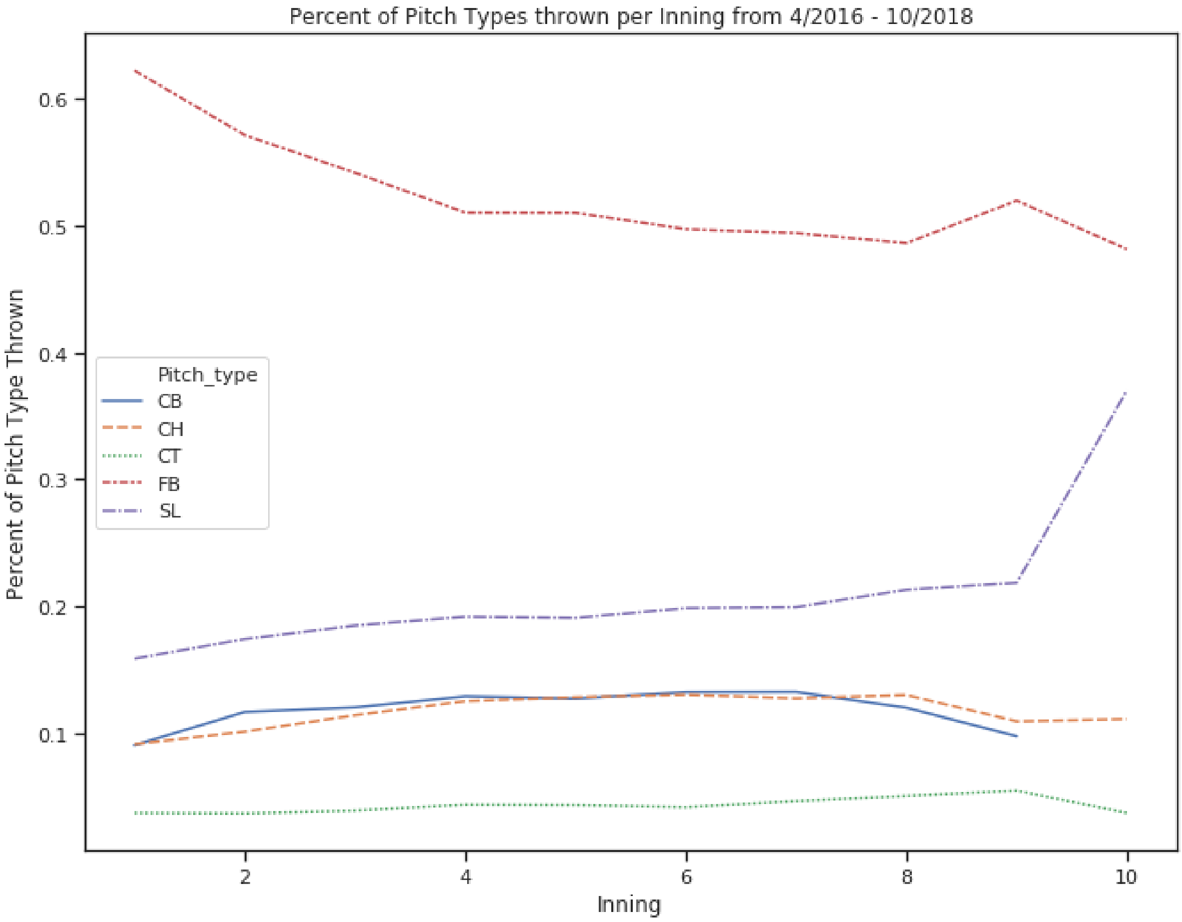


Figure 3: *Percent of Pitch Types Thrown per Inning from 2016 - 2018*

Multiclass Classification Predictive Modeling

Overview

When determining which machine learning algorithm to choose for predictive analysis, it was important to consider such factors:

- Size, quality, and nature of the data
- Computational time and resources
- Question(s) we want to answer with the data

The ‘cleaned’ data for 20 pitchers of interest consisted of 167,427 rows and 41 columns. The number of columns or features increased to 62-77 columns when categorical data were encoded. A majority of the numerical data were discrete values. Only the cumulative percent of pitch types was continuous. The remaining data types includes categorial (h_zone, h_type, o_id) and boolean (is_strike_zone, is_hitter_righty, is_top, is_bunt_shown). The quality of the ‘cleaned’ data was fairly good. There were no longer any null values and data formats were converted accordingly.

Computationally, I used Ubuntu 18.04 on an Intel Core i5-7400 CPU @ 3GHz x 4 with 8 GB of memory. Also, because the fairly ‘good’ quality data was not large for the computational resources I had available, I did not need to limit my model to something like a Naive Bayes which is computationally great for larger data sets.

The question I want to answer with this data was if it was possible to predict (with some probability) which **pitch type** a pitcher would throw next, given previous pitch and current game information? Ideally, I wanted a model that was fast and and accurate but also easily interpretable.

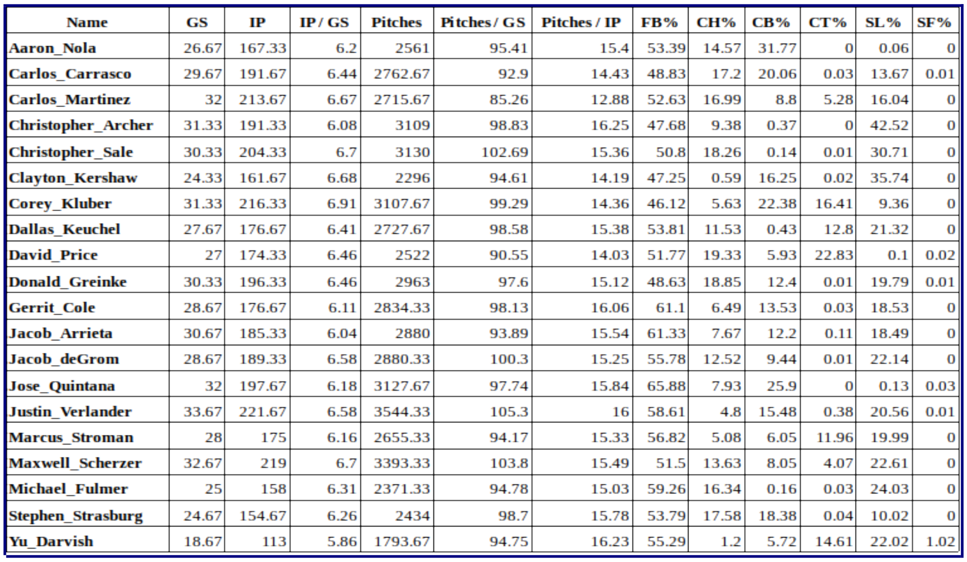
Train / Test Split

The target data consists of the following six (6) pitch-types:

(FB) Fastball (CH) Change-up (CB) Curveball
(CT) Cutter (SP) Split-finger (SL) Slider

The type of pitch thrown will vary from pitcher to pitcher. To predict the “next” pitch, multi-class classification was required. As seen in **Table 1** below, the target data is imbalanced. For these 20 pitchers of interest and most MLB starting pitchers, on average, 46% - 66% of all pitches thrown are Fastballs. This is between 8X to 5,000X greater than at least one of the other pitch types (CH, CB, SL, CT, SP) they throw as part of their arsenal.

Attempts to use **Synthetic Minority Over-sampling Technique (SMOTE)** and **stratify** the training data to establish a balanced data set did not improve predictive accuracy. A copy of the target data for each pitcher were also binarized for plotting Precision-Recall curves for each class rather than using a single ‘micro’-average score.



Name	GS	IP	IP / GS	Pitches	Pitches / GS	Pitches / IP	FB%	CH%	CB%	CT%	SL%	SF%
Aaron_Nola	26.67	167.33	6.2	2561	95.41	15.4	53.39	14.57	31.77	0	0.06	0
Carlos_Carrasco	29.67	191.67	6.44	2762.67	92.9	14.43	48.83	17.2	20.06	0.03	13.67	0.01
Carlos_Martinez	32	213.67	6.67	2715.67	85.26	12.88	52.63	16.99	8.8	5.28	16.04	0
Christopher_Archer	31.33	191.33	6.08	3109	98.83	16.25	47.68	9.38	0.37	0	42.52	0
Christopher_Sale	30.33	204.33	6.7	3130	102.69	15.36	50.8	18.26	0.14	0.01	30.71	0
Clayton_Kershaw	24.33	161.67	6.68	2296	94.61	14.19	47.25	0.59	16.25	0.02	35.74	0
Corey_Kluber	31.33	216.33	6.91	3107.67	99.29	14.36	46.12	5.63	22.38	16.41	9.36	0
Dallas_Keuchel	27.67	176.67	6.41	2727.67	98.58	15.38	53.81	11.53	0.43	12.8	21.32	0
David_Price	27	174.33	6.46	2522	90.55	14.03	51.77	19.33	5.93	22.83	0.1	0.02
Donald_Greinke	30.33	196.33	6.46	2963	97.6	15.12	48.63	18.85	12.4	0.01	19.79	0.01
Gerrit_Cole	28.67	176.67	6.11	2834.33	98.13	16.06	61.1	6.49	13.53	0.03	18.53	0
Jacob_Arrieta	30.67	185.33	6.04	2880	93.89	15.54	61.33	7.67	12.2	0.11	18.49	0
Jacob_deGrom	28.67	189.33	6.58	2880.33	100.3	15.25	55.78	12.52	9.44	0.01	22.14	0
Jose_Quintana	32	197.67	6.18	3127.67	97.74	15.84	65.88	7.93	25.9	0	0.13	0.03
Justin_Verlander	33.67	221.67	6.58	3544.33	105.3	16	58.61	4.8	15.48	0.38	20.56	0.01
Marcus_Stroman	28	175	6.16	2655.33	94.17	15.33	56.82	5.08	6.05	11.96	19.99	0
Maxwell_Scherzer	32.67	219	6.7	3393.33	103.8	15.49	51.5	13.63	8.05	4.07	22.61	0
Michael_Fulmer	25	158	6.31	2371.33	94.78	15.03	59.26	16.34	0.16	0.03	24.03	0
Stephen_Strasburg	24.67	154.67	6.26	2434	98.7	15.78	53.79	17.58	18.38	0.04	10.02	0
Yu_Darvish	18.67	113	5.86	1793.67	94.75	16.23	55.29	1.2	5.72	14.61	22.02	1.02

Table 1: List of Starting Pitchers

For each starting pitcher, table includes the average number of Games Started (GS), Innings Pitched (IP), Pitches Thrown (Pitches) over the last three years (2016-2018). Also shown, average percent of pitch types (Fastball: FB, Changeup: CH, Curveball: CB, Cutter: CT, Slider: SL, and Split-finger: SP) thrown.

Not shown, average percent of Non-pitches, Intentional Balls, and Unknown pitches.

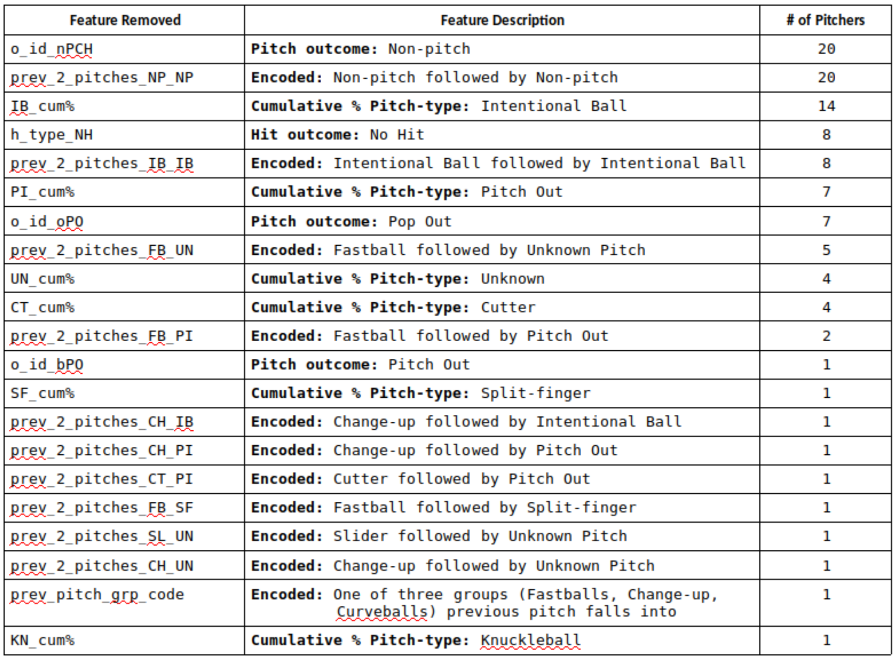
Feature Engineering

Single Unique Features

A feature with only one unique value cannot be useful for machine learning because this feature has zero variance. Such single unique features were removed. For example, the feature, 'is_pitcher_righty' will not change. If a pitcher is right-handed, the feature will always be 1.

Correlated Features

Identified and removed features (columns) that had a positive or negative correlation of 95% or greater. A total of 21 unique features (shown in Table 2) were removed from 1 or more pitchers data set due to high correlation.



Feature Removed	Feature Description	# of Pitchers
o_id_nPCH	**Pitch outcome:** Non-pitch	20
prev_2_pitches_NP_NP	**Encoded:** Non-pitch followed by Non-pitch	20
IB_cum%	**Cumulative % Pitch-type:** Intentional Ball	14
h_type_NH	**Hit outcome:** No Hit	8
prev_2_pitches_IB_IB	**Encoded:** Intentional Ball followed by Intentional Ball	8
PI_cum%	**Cumulative % Pitch-type:** Pitch Out	7
o_id_oPO	**Pitch outcome:** Pop Out	7
prev_2_pitches_FB_UN	**Encoded:** Fastball followed by Unknown Pitch	5
UN_cum%	**Cumulative % Pitch-type:** Unknown	4
CT_cum%	**Cumulative % Pitch-type:** Cutter	4
prev_2_pitches_FB_PI	**Encoded:** Fastball followed by Pitch Out	2
o_id_bPO	**Pitch outcome:** Pitch Out	1
SF_cum%	**Cumulative % Pitch-type:** Split-finger	1
prev_2_pitches_CH_IB	**Encoded:** Change-up followed by Intentional Ball	1
prev_2_pitches_CH_PI	**Encoded:** Change-up followed by Pitch Out	1
prev_2_pitches_CT_PI	**Encoded:** Cutter followed by Pitch Out	1
prev_2_pitches_FB_SF	**Encoded:** Fastball followed by Split-finger	1
prev_2_pitches_SL_UN	**Encoded:** Slider followed by Unknown Pitch	1
prev_2_pitches_CH_UN	**Encoded:** Change-up followed by Unknown Pitch	1
prev_pitch_grp_code	**Encoded:** One of three groups (Fastballs, Change-up, Curveballs) previous pitch falls into	1
KN_cum%	**Cumulative % Pitch-type:** Knuckleball	1

Table 2: List of Correlated Features Removed

Feature Importance

Features with zero-to-low importance for each pitcher were removed. Feature importance was determined using XGBoost Classifier. To avoid overfitting, multiple evaluation metrics (**merror** = multi-class error rate, **mlogloss** = multi-class negative log-likelihood) were averaged over 10 iterations to identify when early stopping should occur. Figure 4 shows results for both metrics for the first iteration with a 57.41% accuracy for pitcher, Chris Archer.

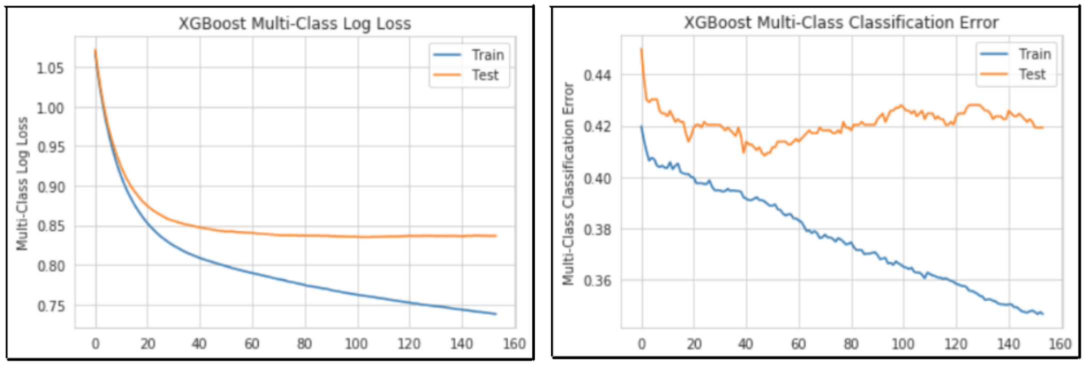


Figure 4: First Iteration Evaluation Metrics Comparison for Chris Archer

<u>Left</u>: Multi-class Log Loss Evaluation using XGBoost Classifier. <u>Right</u>: Multi-class Error Rate (#wrong cases /# all cases) Evaluation using XGBoost Classifier. In this case, early stopping can be applied around the 45-50 epoch.

The most important feature for every pitcher was the Pitcher's Pitch Count (p_p_count) which is the number of pitches the pitcher has thrown in a game. The top 20 features (Figure 5) are illustrated for Chris Archer.

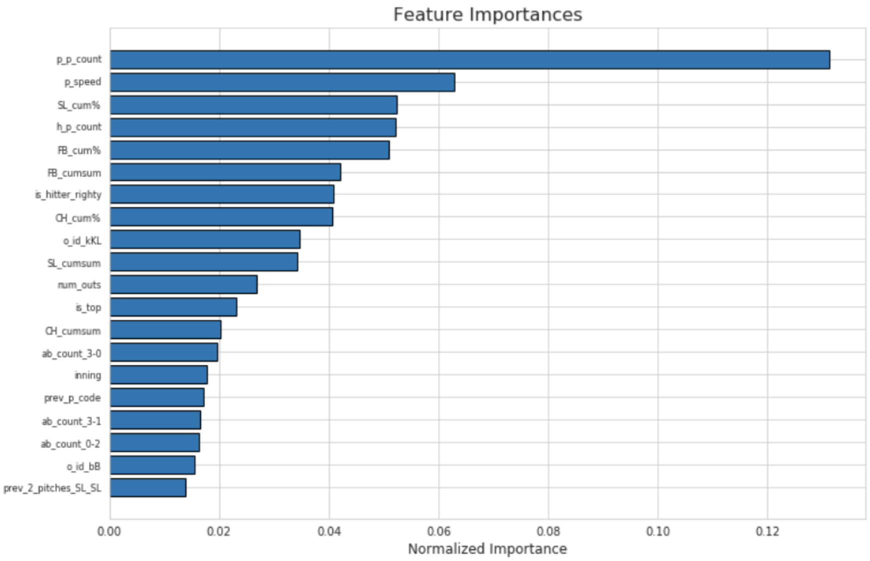


Figure 5: Top 20 Important Features for Chris Archer

With features importance sorted, only the features that contributed to 99% of cumulative importance were kept. For Chris Archer, only 67 of the 100+ features were required for 99% cumulative importance as shown in Figure 6.

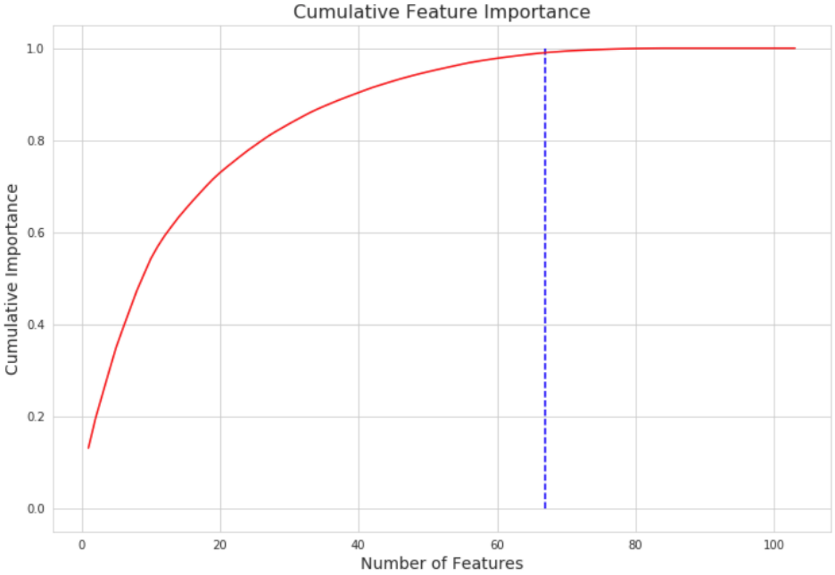


Figure 6: Cumulative Feature Importance Curve for Chris Archer

Compare Models

Naive Approach

The naive approach to predicting the next pitch-type would be to only consider the previous pitch. There are two ways to apply the naive approach.

1. **Consider the distributions for “first” pitches**

   At the beginning of every inning, there is a “first” pitch – when there is not a previous pitch
2. **Consider the distributions for two-pitch sequences**

   Previous Pitch → Next Pitch

When we consider the distribution of “first” pitches for Carlos Martinez, shown in Figure 7, we see that he throws a Fastball more than 60% of the time. The next probable pitch could be either a Change-up or Curveball at 12%.

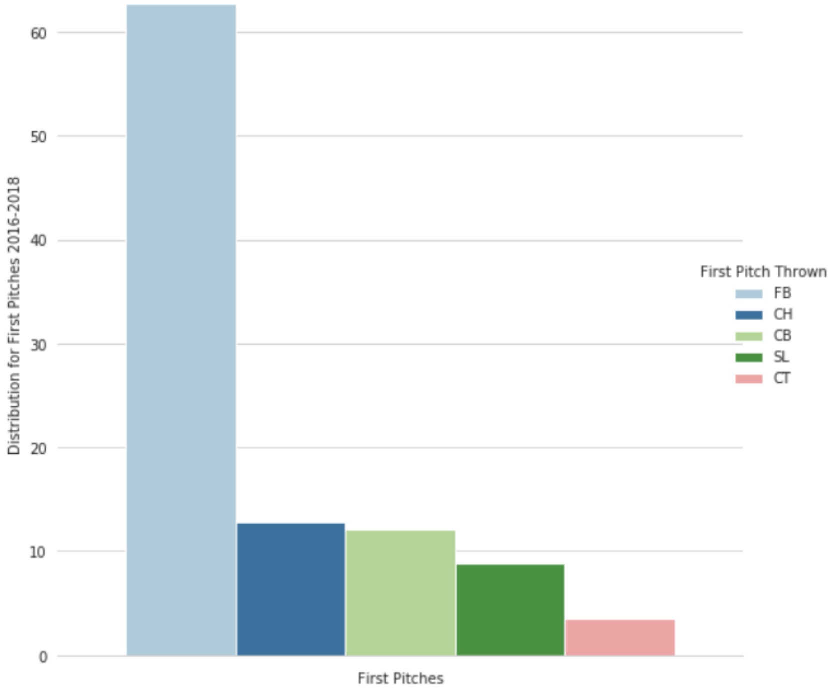


Figure 7: Distribution of First Pitches for Carlos Martinez (2016-2018)

However, when we consider the distribution for two-pitch sequences for Carlos Martinez, shown in Figure 8, we now see that after he throws a Fastball, more than 30% of the time, he follows it up with another Fastball. It is also interesting to see that when Carlos throws a Slider or Cutter, he almost never follows it up with a Curveball.

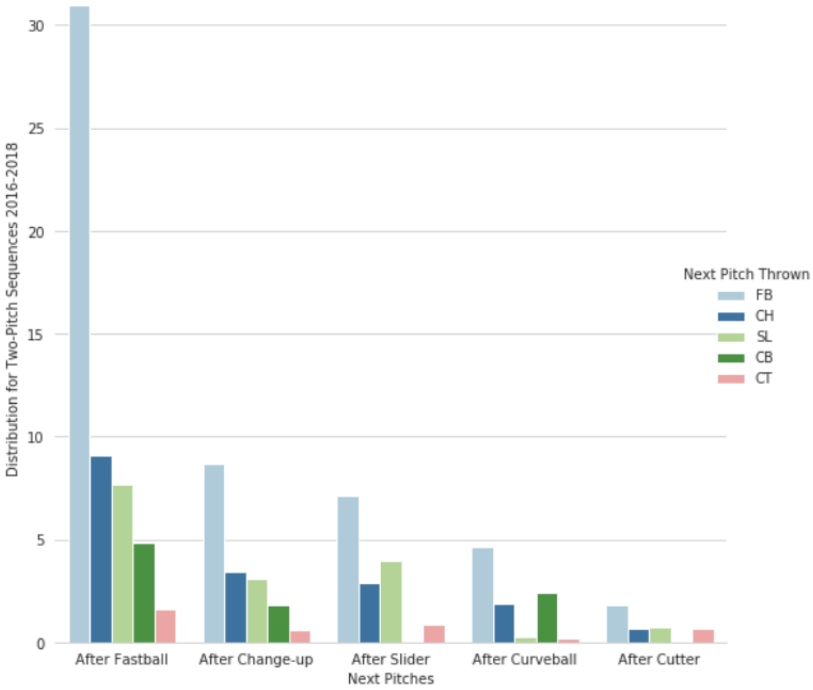


Figure 8: Distribution for Two-Pitch Sequences for Carlos Martinez (2016-2018)

As a performance metric, if we were to consider the previous pitch only scenario as a binary classification, we can assume a pitch is either a Fastball (1) or it is not (0). By using Logloss to measure the error if the,

- actual value of **y = {1 = yes, 0 = no}**
- prediction probability that value = 1 is **0.6**

then, for one observation where **True = 0** and **Predicted = 1**, we can calculate the **Logloss** as **log(0.6) ~ 0.916**, which is nearly 1 when we are attempting to reach an error of 0. **Our goal would be to implement a model that can generate a logloss error less than the naive approach.**

Test ‘Out-of-box’ Models

To determine the “best” model to use for predictive analysis, I first looked at eight “out-of-box” models by implementing a pipeline to normalize feature vectors into a representation more suitable for the downstream estimators and run 5-fold cross-validation. For Jake Arrieta, when considering the average test score, we can see which of the estimators performed the best:

```
Player 12:  Jake Arrieta
---------------------------------
Logistics Regression : 60.97
SVC - linear : 61.45
SVC - rbf : 61.45
K-neighbors : 57.58
Decision Trees : 45.71
Random Forest : 57.16
Gaussian NB : 11.3
Gradient Boosting : 60.02
Adaboost : 60.64
XG Boost : 61.18
-----------------------------------
```

Based on the average accuracy test scores and goal of project, I decided to try **Logistic Regression**, **Support Vector Machine** using **Radial Basis Function**, **Random Forest Classifier**, **Gradient Boosting**, and **Decision Trees**. I chose to keep Decision Trees to include as a “weak” classifier in an **Ensemble Voting Classifier**.

Hyper-parameter Tuning

For each of the six (6) estimators selected, I performed an exhaustive search over specified parameter values specific for each estimator using GridSearchCV. The “best_estimator” for each estimator was saved to .pkl file.

Compare ‘Tuned’ Models

Using an Ensemble Voting Classifier with all six (6) of the “best” estimators, I was able to compare the average class probabilities computed by each estimator against the combined voting classifier. For pitcher, Marcus Stroman who uses five (5) pitches as part of his arsenal, we can see slight differences between the classifiers in Figure 9. While we see that the proportion of class probabilities is similar across all classifiers, this is not always the case. For some pitchers, we do see where some pitch-types have a higher probability in one classifier than in others.

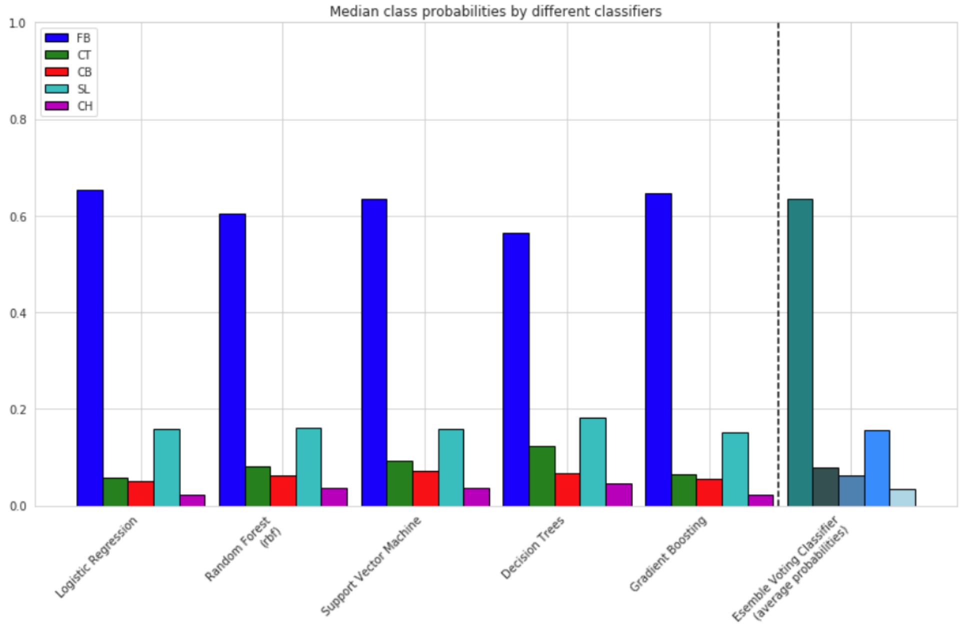


Figure 9: Median class probabilities for each classifier for Marcus Stroman

Performance Measurements

To determine if a predictive model is considered "good", we need to first define the purpose of the model and what is considered success criterion.

- For this project, the purpose of the classification model is to predict the next pitch-type thrown by a starting Major League Baseball pitcher.
- The model can be considered "**successful**" if it is able to predict the next pitch-type with greater accuracy/probability and less error than the naive approach.
- **Error was measured using Multi-class logloss / Cross-entropy loss.**

For each pitcher, by computing the accuracy of a classifier and penalising false classifications, we calculate the following Logloss error values below:

Pitcher	Multi-class Logloss Error
Aaron Nola	0.977
Carlos Carrasco	1.211
Carlos Martinez	1.257
Chris Archer	0.876
Chris Sale	0.979
Clayton Kershaw	1.005
Corey Kluber	1.289
Dallas Keuchel	1.129
David Price	1.19
Gerrit Cole	1.1
Jacob deGrom	1.149
Jake Arrieta	1.039
Jose Quintana	0.814
Justin Verlander	1.039
Marcus Stroman	1.173
Max Scherzer	1.257
Michael Fulmer	0.928
Stephen Strasburg	1.181
Yu Darvish	1.187
Zack Greinke	1.203

While most classification machine learning models can be validated by accuracy estimation techniques, this is not the case for this project. Because the data is imbalanced, we cannot use accuracy as best indicator. Also, while ROC curves summarize the trade-off between the True Positive Rate and False Positive Rate at different probability thresholds, they are more appropriate for when the observations are balanced. Because this data is imbalanced, Precision-Recall curves are better suited as they summarize the trade-off between the True Positive Rate and the Positive Predictive Value at different probability thresholds.

Confusion Matrix

To describe the performance of the voting classifier model on the pitcher data set where the true classes (pitch-types) are known, a confusion matrix was used. For Clayton Kershaw, we see in Figure 10 that most of the pitch-types predicted are Fastballs (FB) with Sliders (SL) being the second most predicted pitch-type. As a result, the model is unable to effectively distinguish between FB's, SL's, or CB's.

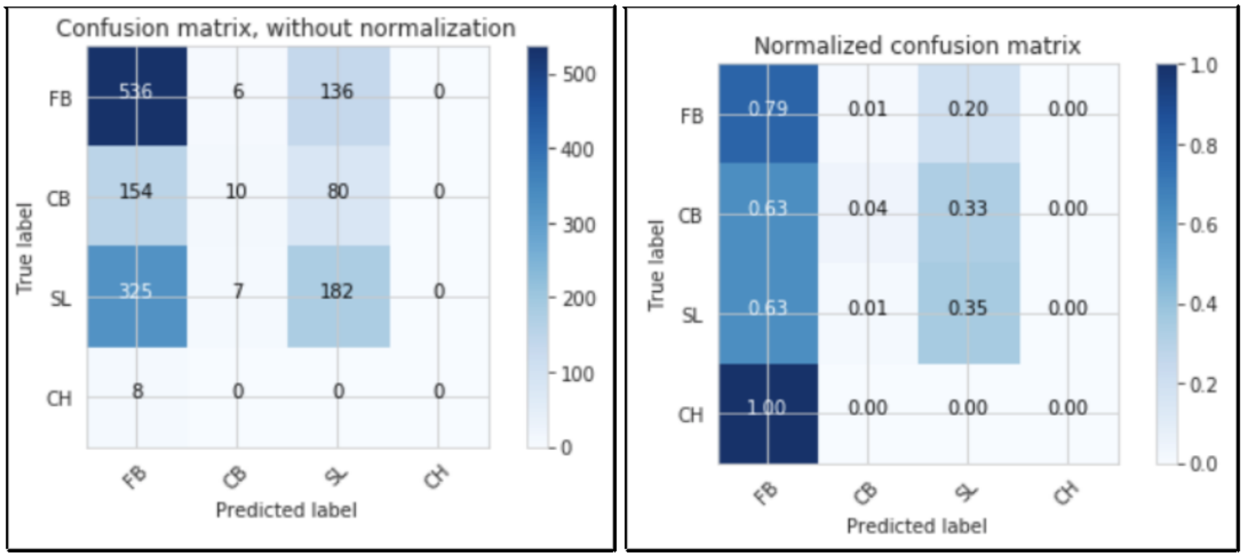


Figure 10: Confusion Matrix for Clayton Kershaw

Left: Raw predicted vs true values per class. Right: Normalized predicted vs true values per class.

ROC Curve

While ROC curves are not appropriate for this data set due to being imbalanced, using the One vs. All classifier model, we see in Figure 11, for Carlos Martinez the ROC curves for each class (pitch-type), including the micro- and macro- averages. Because the data is imbalanced, micro-average ROC curve should be considered.

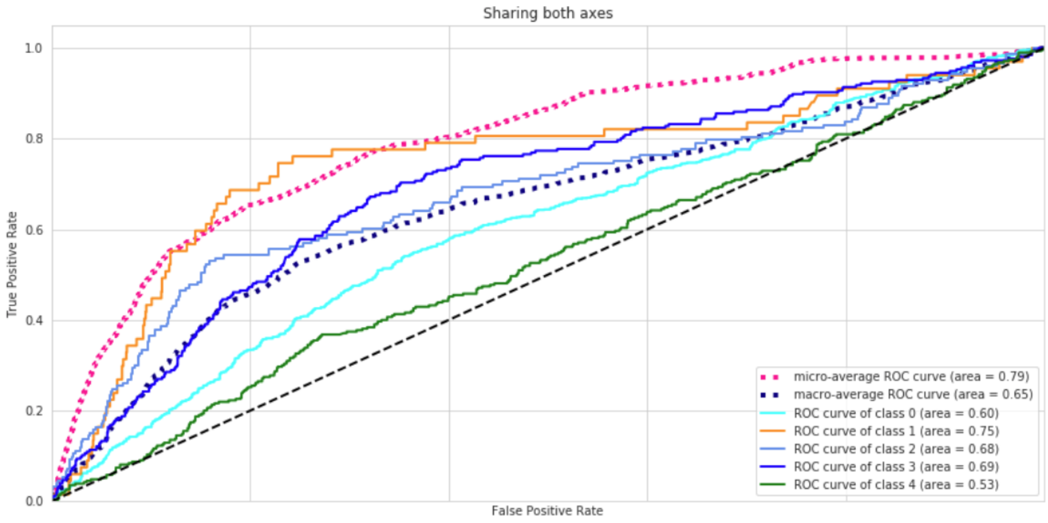


Figure 11: ROC Curve for Carlos Martinez

Pitch-type – class associations are as follows:
Fastball (FB) = 0, Cutter (CT) = 1, Curveball (CB) =2, Slider (SL) = 3, and Change-up (CH) = 4

Precision-Recall

In Figure 12, the Precision-Recall curve for Jose Quintana shows the micro-average – quantifying score on all classes jointly. We see that the micro-average is 0.66.

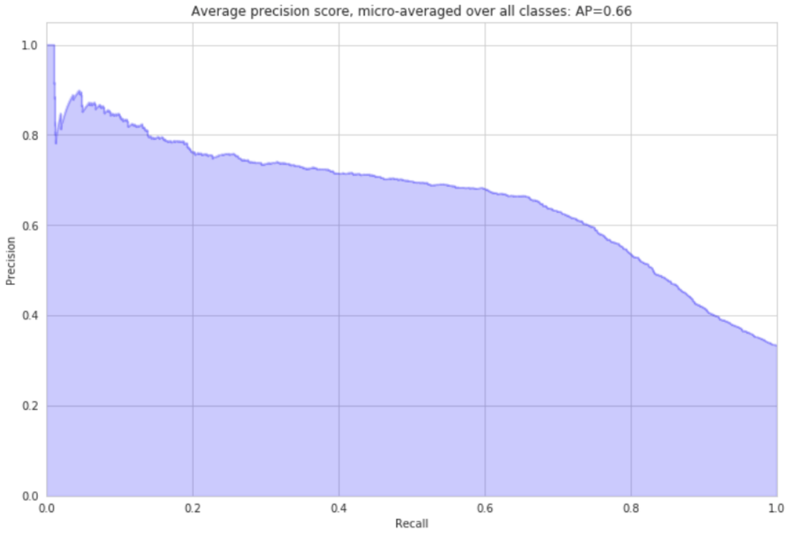


Figure 12: "Micro-average" Precision-Recall Curve for Jose Quintana

In Figure 13, we see the Precision-Recall curve and average precision score for each class (pitch-type) for Jose Quintana. The F1 score was also computed to be 0.54.

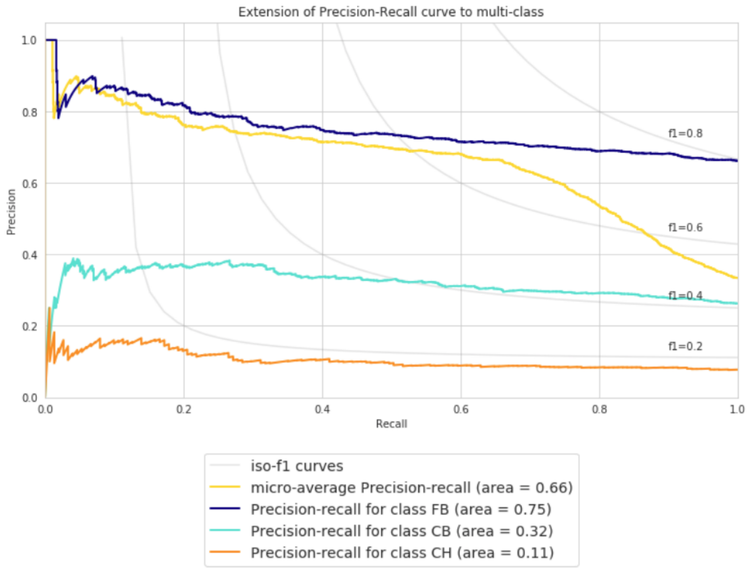


Figure 13: Precision-Recall Curve for each Class for Jose Quintana

Conclusion

Summary

Identifying and fitting the different models to the pitch data set for each pitcher was an extremely interesting and learning experience. Based on the sample size of 20 pitchers over 3 years, there are a couple of observations we can take away.

1. It does appear that we can predict the next pitch with a slightly higher probability than the naive approach for Fastballs, but much more difficult for other pitch-types.
2. For some pitchers, we are able to predict the next-pitch more accurately than for others.

   This may be due to the number of classes (pitch-types) -- Pitchers with more pitches in their arsenal make it more difficult to predict non-Fastballs

It is also important to note that only supervised learning approaches were used in this project and that unsupervised and reinforcement learning should be considered. Some features such a score differential, hitters statistics, weather conditions, and additional season should be included in further analysis to contribute to training process.