Pitch Prediction

Can we predict the next pitch?

Springboard Data Science Career Track

Mark Rojas

Concerns

- Major League Baseball (MLB) attendance dropped more than **6%** this year, continuing a steady decline.

"It's the lowest league-wide attendance since 2003 and the largest single-season drop in a decade."

- While some clubs saw a jump in attendance, **17** of the **30** franchises sold fewer tickets than they did last year.
- Using average ticket prices from Team Marketing Report, that comes to about **$93.7 million in lost ticket revenue in 2018**.

- money.com

But why?

Possible Reasons:

- High Ticket Prices
- Poor Weather Conditions
- Off-season Inactivity
- **In-game Action! (*or lack of*)**

So what can we do?

- Increased player salaries makes lowering ticket prices unsustainable.
- As of today, we are still unable to control the weather.
- Not every team can acquire the ‘big’ name free-agent during the off-season or **win the World Series like the Houston Astros did in 2017 who recorded a $23.7 million boost in sales in 2018.**

Let’s improve the in-game action!

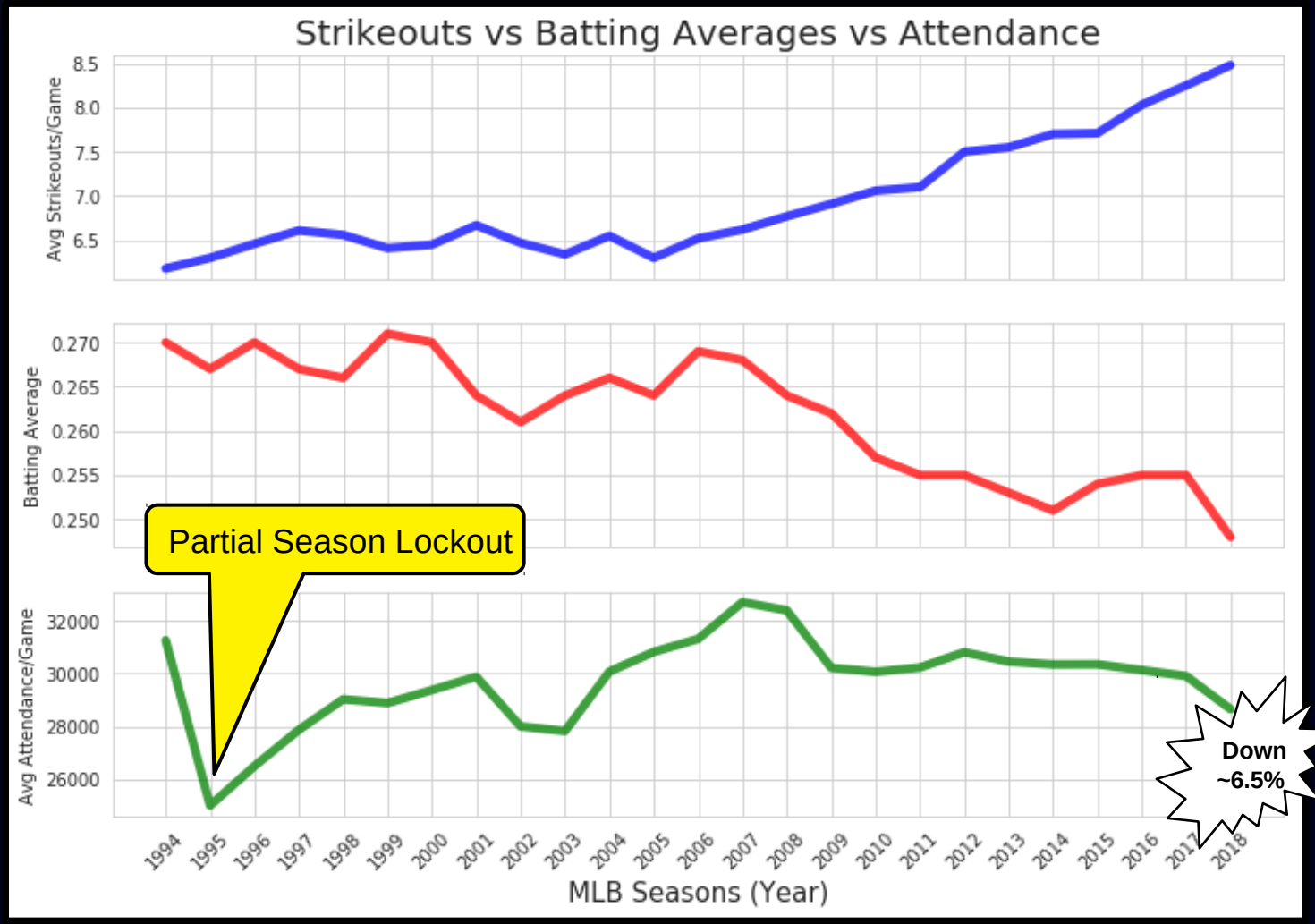
But how?

Impact on Attendance
Strikeouts | Batting Average

Average Strikeouts per Game

Batting Averages (Hits / At Bats)

Average Attendance per Game



Goal

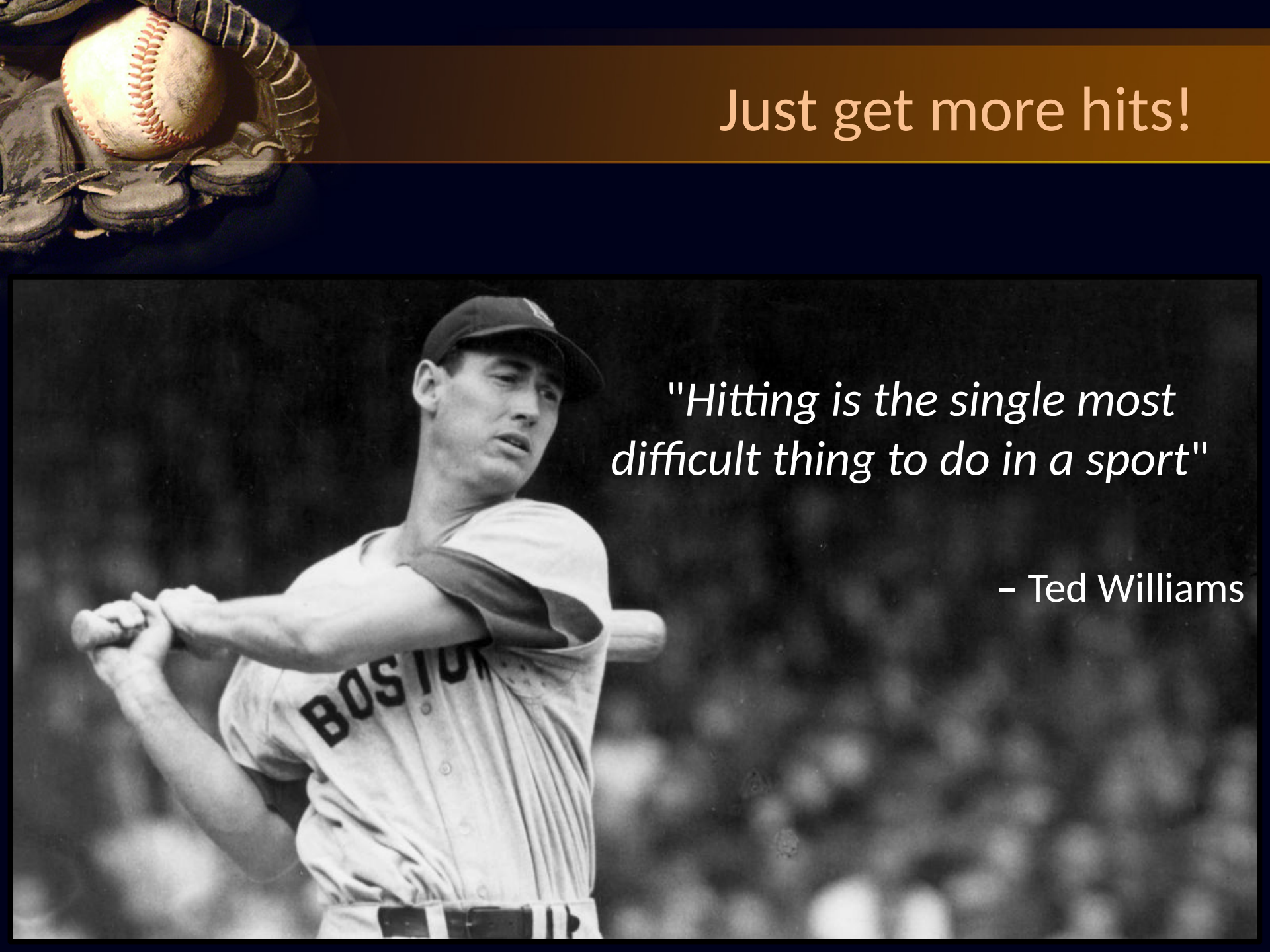
By increasing the mean Batting Average (BA) for hitters, we:

Deliver an action-packed game!

Putting butts in the seats!

- Increase number of base runners
- Increase chances for stolen bases
- Increase opportunities to pick-off runners
- Increase potential runs

Just get more hits!



Throw Hitters a Life Line

- Many pitchers these days are taking steps to avoid ‘tipping’ their pitches, so it’s harder to tell what pitch is coming.

- Teams found in violation of ‘sign stealing’ could be punished with forfeited draft picks and international spending money.

We propose the use of Machine Learning to predict the next-pitch and in turn improve a hitters overall Batting Average.

Pitchers of Interest (POI)

'Top' **20 pitchers of interest** selected from the 2016, 2017, and 2018 seasons.

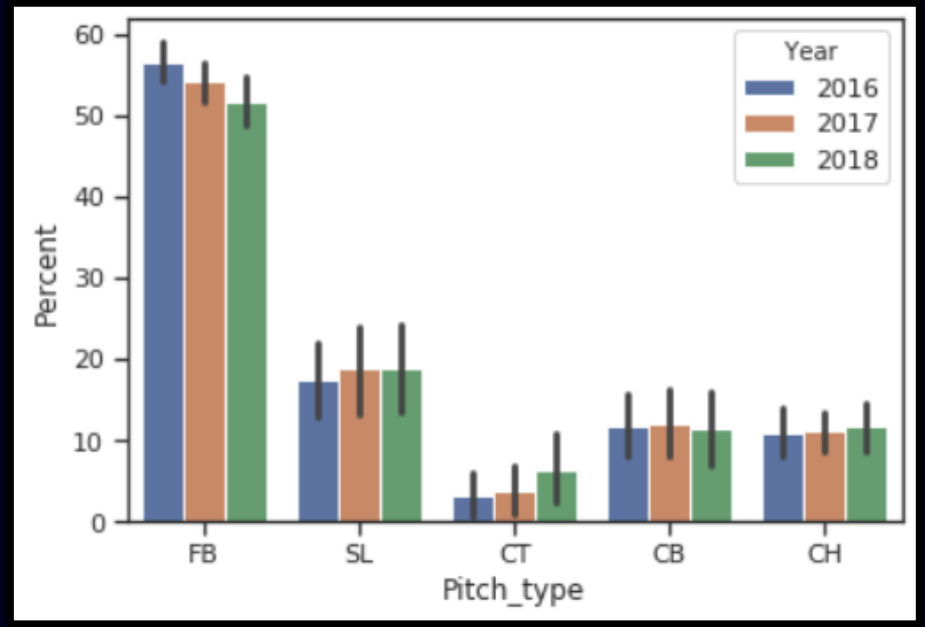
GS – Games started / season
IP – Innings pitched / season
Pitches - Pitches thrown / season
IP/GS – Innings pitched / game
Pitches/GS – Pitches thrown / game
Pitches/IP – Pitches thrown / inning

Note: The averages for these pitchers of interest are in-line with the rest of the starting pitchers in MLB.

Name	GS	IP	Pitches	IP/GS	Pitches/GS	Pitches/IP
Aaron Nola	26.67	167.33	2561.00	6.20	95.41	15.40
Carlos Carrasco	29.67	191.67	2762.67	6.44	92.90	14.43
Carlos Martinez	32.00	213.67	2715.67	6.67	85.26	12.88
Chris Archer	31.33	191.33	3109.00	6.08	98.83	16.25
Chris Sale	30.33	204.33	3130.00	6.70	102.69	15.36
Clayton Kershaw	24.33	161.67	2296.00	6.68	94.61	14.19
Corey Kluber	31.33	216.33	3107.67	6.91	99.29	14.36
Dallas Keuchel	27.67	176.67	2727.67	6.41	98.58	15.38
David Price	27.00	174.33	2522.00	6.46	90.55	14.03
Gerrit Cole	28.67	176.67	2834.33	6.11	98.13	16.06
Jacob deGrom	28.67	189.33	2880.33	6.58	100.30	15.25
Jake Arrieta	30.67	185.33	2880.00	6.04	93.89	15.54
Jose Quintana	32.00	197.67	3127.67	6.18	97.74	15.84
Justin Verlander	33.67	221.67	3544.33	6.58	105.30	16.00
Marcus Stroman	28.00	175.00	2655.33	6.16	94.17	15.33
Max Scherzer	32.67	219.00	3393.33	6.70	103.80	15.49
Michael Fulmer	25.00	158.00	2371.33	6.31	94.78	15.03
Stephen Strasburg	24.67	154.67	2434.00	6.26	98.70	15.78
Yu Darvish	18.67	113.00	1793.67	5.86	94.75	16.23
Zack Greinke	30.33	196.33	2963.00	6.46	97.60	15.12

Pitch Type Consistency

Percent of Pitch Types Thrown from 2016 - 2018



- We continue to see roughly the same ratios of Fastballs, Sliders, Cutters, Curveballs, and Change-ups thrown from year-to-year.
- From 2016-2018, the percent of Fastballs has decreased slightly while Cutters have increased.

The Dataset – Sportradar API

Past Events:

Pitch Type
Pitch Velocity
Pitch Location*
Pitch Outcome*
Pitch Sequences*
Hit Type*
Hit Zone*
Is_Bunt_Shown*
Is_Strike_Zone*

Current Situations:

Inning
Top / Bottom Inning*
Pitch Count
Hitter Count
Hitter Handedness*
At-Bat Count*
Number of Outs
Cumulative Pitch Type Sums
Cumulative Pitch Type %

*** categorical features**

167,427 observations | 41 features for all 20 POI's

Past Events – are those that describe the previous pitch or a result of the previous pitch.

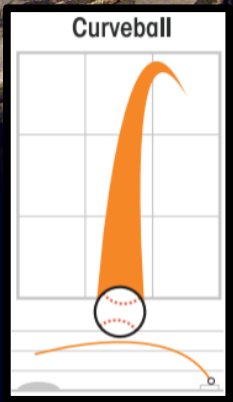
Current Situations – are those that describe the current in-game condition.

Our **Target (labeled pitch types)**, will be used to train our model and attempt to predict the **Next Pitch Type**.

Target Pitch Types:

```
FB = Fastball
SL = Slider
CT = Cutter
CB = Curveball
CH = Change-up
SP = Splitter
KN = Knuckleball
```

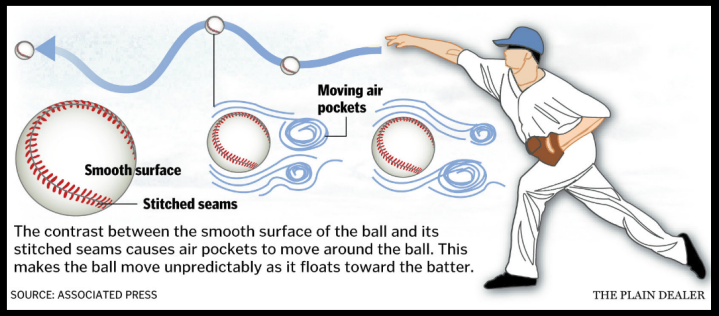
Common Pitch-Types



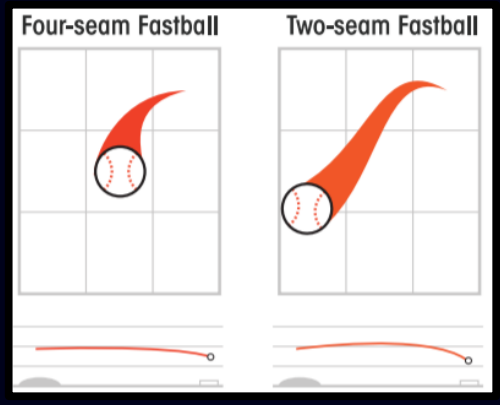
Breakers:

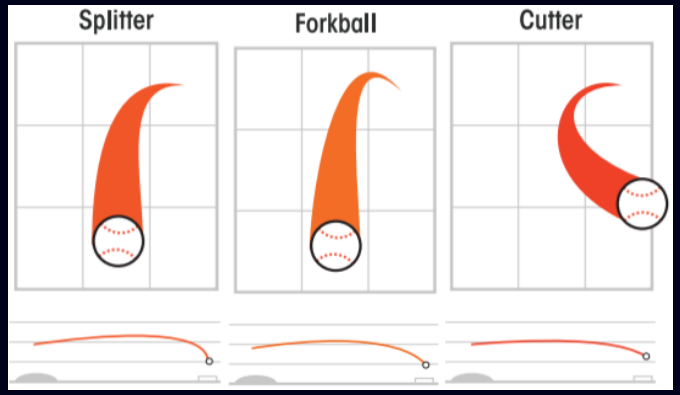
Curveball (70-80 mph)
Breaks from top to bottom

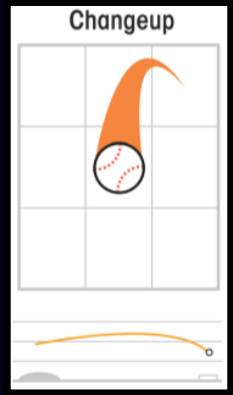
Slider (80-90 mph)
Breaks down and away from Right Handed Hitters
It is between a fastball and curve



Knuckleball
(60-70 mph)
Unpredictable pitch, movement will vary







Change-ups:

Change-up (70-85 mph)
Slower than a fastball, but thrown with the same arm motion

Fastballs:

Four-seam (85-100 mph) - Fastest, straightest pitch, little movement

Two-seam / Sinker (80-90 mph) - Moves downward

Splitter (80-90 mph) – Breaks down suddenly before reaching plate

Forkball (75-85 mph) – Like splitter but gradual downward movement

Cutter (85-95 mph) – Faster than slider, more movement than fastball

Pitchers Arsenal of Pitch Types

Name	FB%	CH%	CB%	CT%	SL%	SF%
Aaron_Nola	53.39	14.57	31.77	0	0.06	0
Carlos_Carrasco	48.83	17.2	20.06	0.03	13.67	0.01
Carlos_Martinez	52.63	16.99	8.8	5.28	16.04	0
Christopher_Archer	47.68	9.38	0.37	0	42.52	0
Christopher_Sale	50.8	18.26	0.14	0.01	30.71	0
Clayton_Kershaw	47.25	0.59	16.25	0.02	35.74	0
Corey_Kluber	46.12	5.63	22.38	16.41	9.36	0
Dallas_Keuchel	53.81	11.53	0.43	12.8	21.32	0
David_Price	51.77	19.33	5.93	22.83	0.1	0.02
Donald_Greinke	48.63	18.85	12.4	0.01	19.79	0.01
Gerrit_Cole	61.1	6.49	13.53	0.03	18.53	0
Jacob_Arrieta	61.33	7.67	12.2	0.11	18.49	0
Jacob_deGrom	55.78	12.52	9.44	0.01	22.14	0
Jose_Quintana	65.88	7.93	25.9	0	0.13	0.03
Justin_Verlander	58.61	4.8	15.48	0.38	20.56	0.01
Marcus_Stroman	56.82	5.08	6.05	11.96	19.99	0
Maxwell_Scherzer	51.5	13.63	8.05	4.07	22.61	0
Michael_Fulmer	59.26	16.34	0.16	0.03	24.03	0
Stephen_Strasburg	53.79	17.58	18.38	0.04	10.02	0
Yu_Darvish	55.29	1.2	5.72	14.61	22.02	1.02

Percent of pitch-types each pitcher throws as part of their arsenal (2016-2018)

Class Imbalance!

Naive Approach

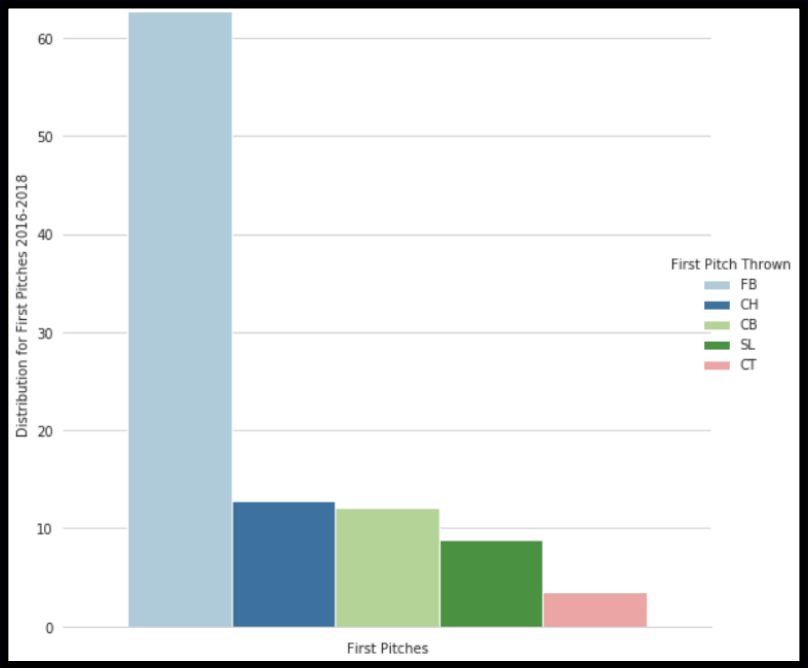
First, let's consider only the previous pitch when predicting the next pitch. There are two ways to apply the naive approach.

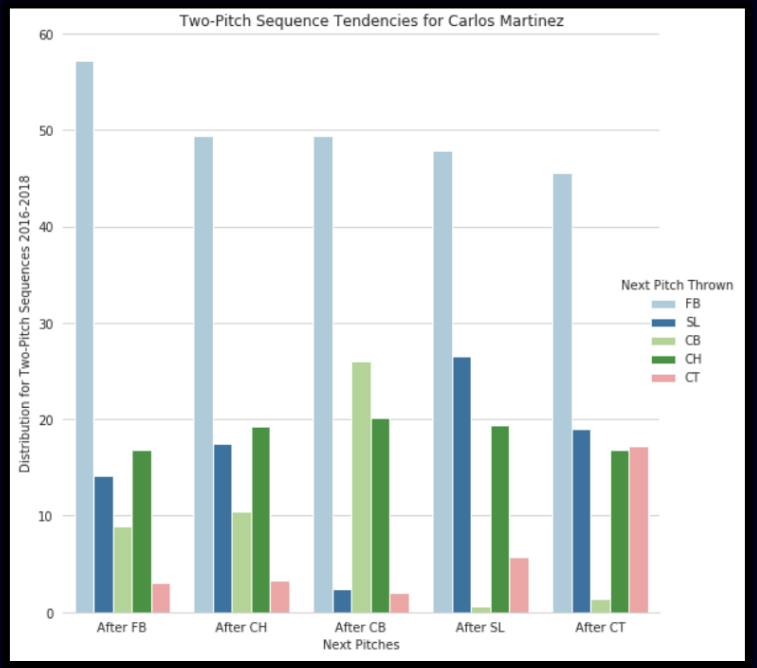
First pitches

Start of every inning, there is a first pitch, when there is not a previous pitch

Two-pitch sequences

Sequence of pitches consisting of previous pitch followed by the next pitch





Distribution of **First Pitch** and **Two-Pitch Sequences** for **Carlos Martinez** (2016-2018)

Naive Approach

Measure Performance Examples - **Accuracy**

In the case for Carlos Martinez, we see that we have **exactly 5 classes**, requiring **multi-class classification**.

Pitch Type	Percent Thrown
Fastball (FB)	0.538
Cutter (CT)	0.039
Curveball (CB)	0.095
Slider (SL)	0.154
Change-up (CH)	0.174

Pitch Types and Percentage Thrown for **Carlos Martinez** (2016-2018)

When considering **accuracy** as a performance metric, the majority class is the **Fastball (FB)** at **0.538**.

So, if we assign Fastball to all of the instances, our model would have a **majority class baseline** of **53.8%** on this training set and our target to beat when comparing other models.

Accuracy Not Ideal!!!

Naive Approach

Measure Performance Examples – **Multi-class Log Loss**

In addition to using accuracy to measure a models performance, we can also compute the **multi-class logarithmic loss (cross entropy)**.

When considering the **previous pitch** as a way to predict the **next pitch** for Carlos Martinez, we see that after he throws a Fastball, the next pitch can be either another **Fastball**, **Change-up**, **Slider**, **Curveball**, or **Cutter**.

Based on Carlos' **two-sequence** pitch data where the first pitch was a Fastball, we calculate the following mean values as predicted probabilities.

prev_2_pitches	count	first_pitch	next_pitch	mean
FB_FB	2178	FB	FB	57.195378
FB_CH	639	FB	CH	16.780462
FB_SL	538	FB	SL	14.128151
FB_CB	339	FB	CB	8.902311
FB_CT	114	FB	CT	2.993697

Computed Log Loss **61.8%**

Correlated Features

- Identified and removed features with a 95% or greater correlation.
- Removed a total of 21 unique features for one or more pitchers.

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

Feature Importance

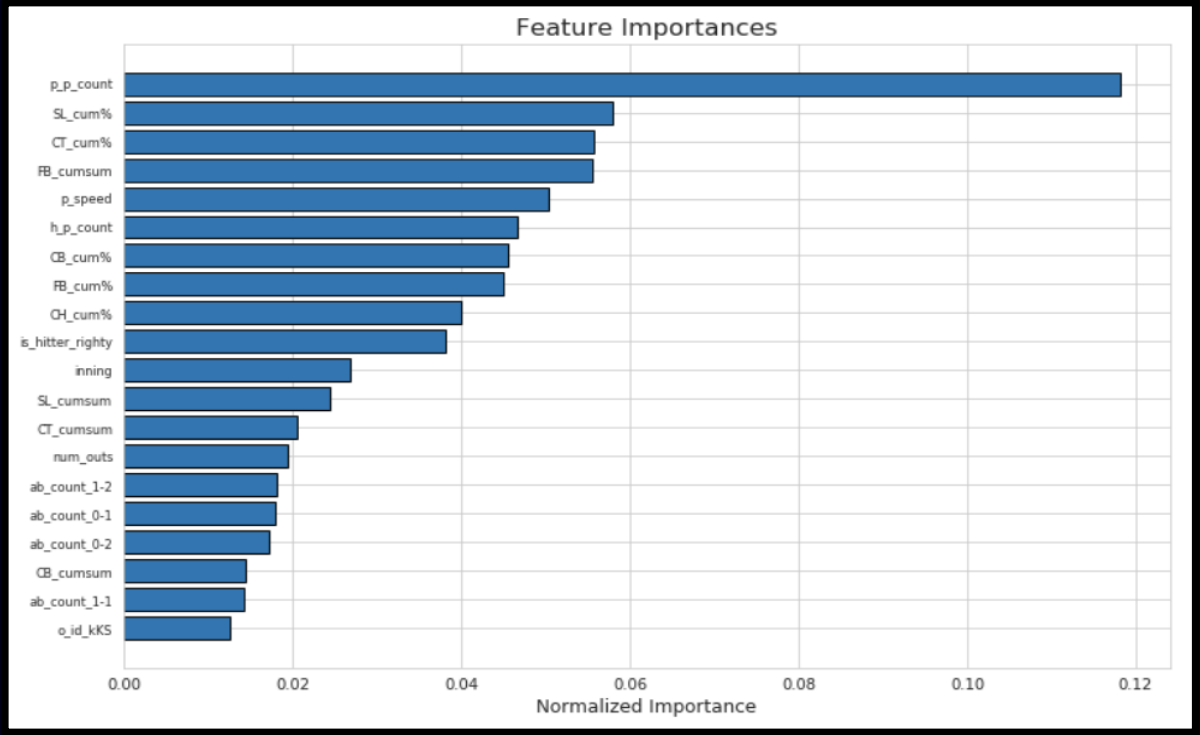
Estimated Feature Importance using Gradient Boosting. Features with zero-to-low importance were removed.

Top Feature for each Pitcher

- Number of pitches thrown in the game (**p_p_count**)

Other top Features

- Cumulative sums and %'s for varying pitch types
- Previous pitch speed
- Inning
- At-bat count (balls - strikes)



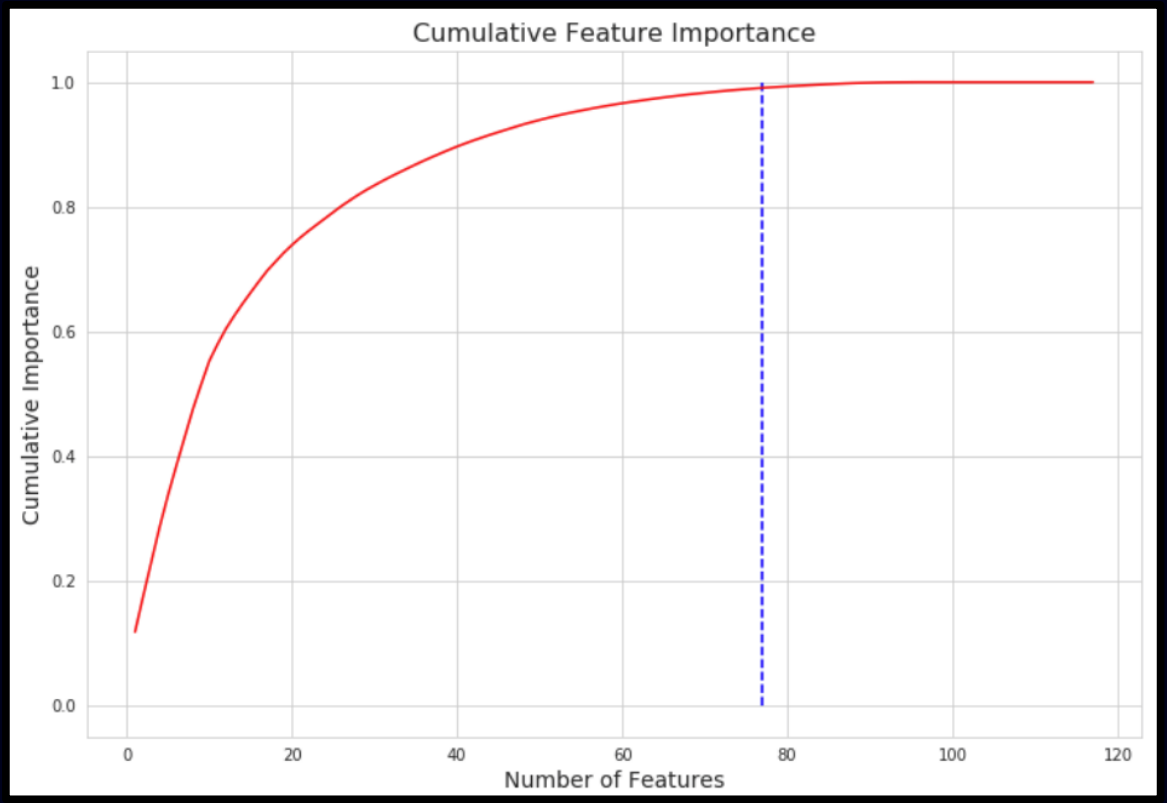
Top 20 Important Features for Marcus Stroman

Feature Importance

Top features contributing to **99%** of cumulative importance.

In this example

- **77** features required for 99% cumulative importance
- **41** zero-to-low importance features removed from data set



Cumulative Feature Importance Curve for Marcus Stroman

Predictive Model Selection

Now that we know more about the data, we can select a predictive model to assess the data in a way that helps make predictions about what might happen in the future.

- Labeled Data
- Multiple Classes
- Imbalanced Targets

Supervised Learning

Multi-class Classification

Commonly used Machine Learning Algorithms:

Logistic Regression can perform multinomial classification, applying a non-linear function (sigmoid).

Support Vector Machine maximizes the margin between the classes and the hyperplane using a loss function. However, training time can be higher than other models and less effective with noisier data.

Decision Trees are easily interpretable and non-parametric (distribution-free), managing outliers but susceptible to overfitting.

Predictive Model Selection

Ensemble Methods

Random Forest is an ensemble of randomized decision trees. Each decision tree gets a random sample of training data and a subset of features to base a decision on.

Gradient Tree Boosting essentially converts weak learners (i.e., decision trees) into strong learners. Not easily interpretable and sensitive to small changes in the set of features.

Ensemble Voting Classifier combines machine learning base estimators for classification via plurality voting to achieve improved generalization and robustness over a single estimator.

Ensemble Voting Approach

Decision: use an **ensemble approach**, combining multiple models with varying predictions in hopes of creating a stronger final prediction.

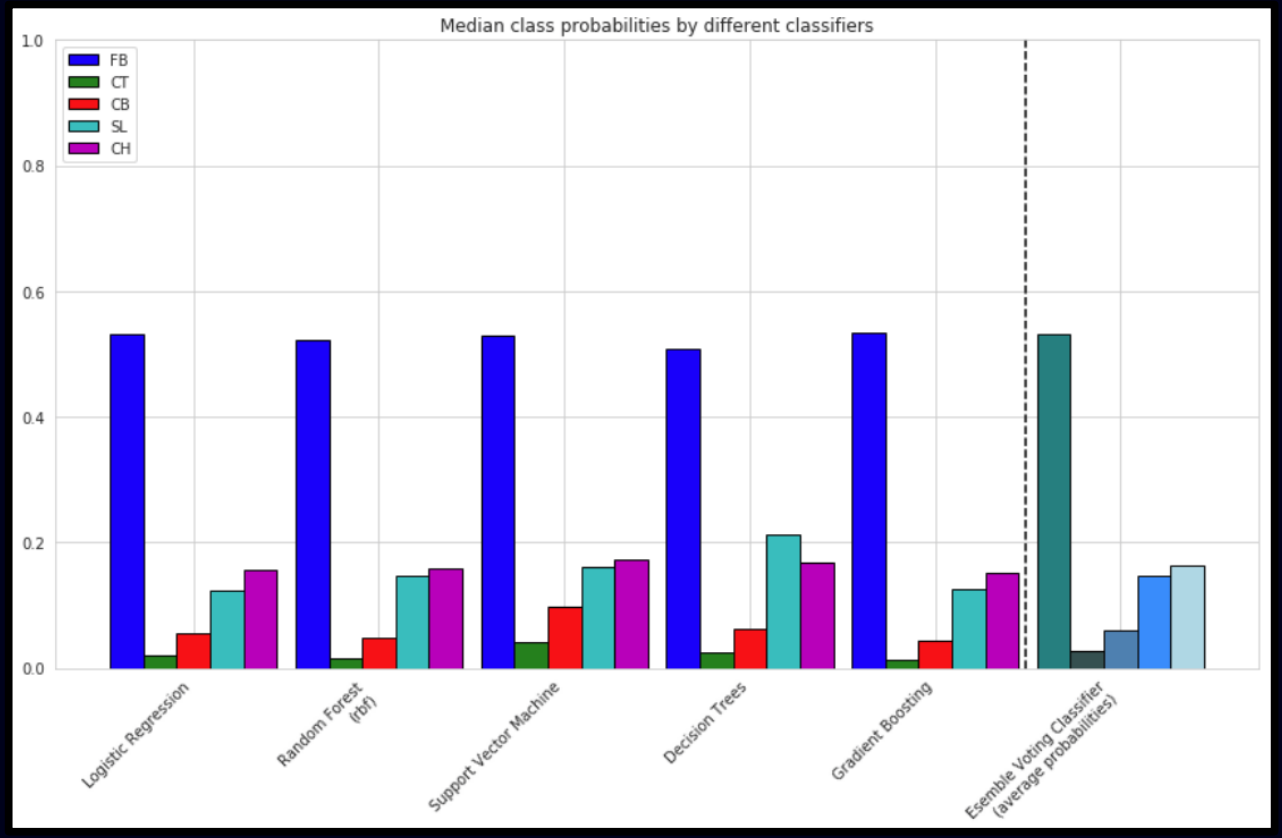
- Performed cross-validation to pre-evaluate ‘out-of-the-box’ models.
- Selected the following classifiers to use as estimators:
  1. Logistic Regression
  2. Support Vector Machine (with Radial Basis Function kernel)
  3. Decision Trees
  4. Random Forest
  5. Gradient Boosting
- Performed Grid SearchCV on each classifier for hyper-parameter tuning.
- Scaled and centered numerical features based on quantile ranges.
- **‘Best Estimators’** used for each classifier in ensemble voting classifier.

Comparing Class Probabilities

Individual Classifiers vs. Ensemble of Classifiers

Observations:

- Similar median values for Fastballs and Cutters
- SVM predicts slightly more Curveballs
- Decision Trees predict more Sliders than Change-ups



Median Class Probabilities for each Classifier for Carlos Martinez

Interpretation of Confusion Metrics

		Predicted class	
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Example: Predicting if pitch-type is a **Fastball** or **Not a Fastball**

True Positives (TP) – Correctly predicted positive pitch-type
e.g., **Predicted**: Fastball | **Actual**: Fastball

True Negatives (TN) – Correctly predicted negative pitch-type
e.g., **Predicted:** Not a Fastball | **Actual:** Not a Fastball

False Positives (FP) – Incorrectly predicted positive pitch-type
e.g., **Predicted:** Fastball | **Actual:** Not a Fastball

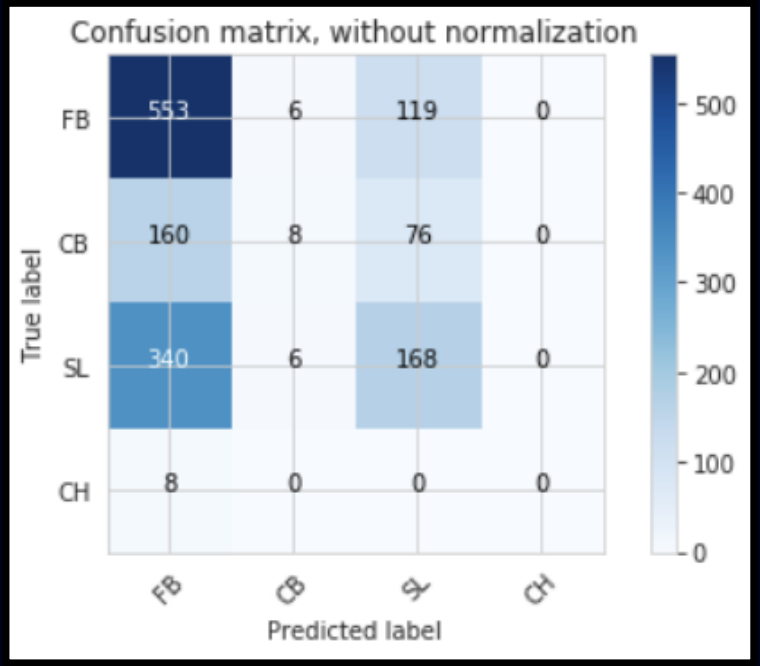
False Negatives (FN) – Incorrectly predicted negative pitch-type
e.g., **Predicted:** Not a Fastball | **Actual:** Fastball

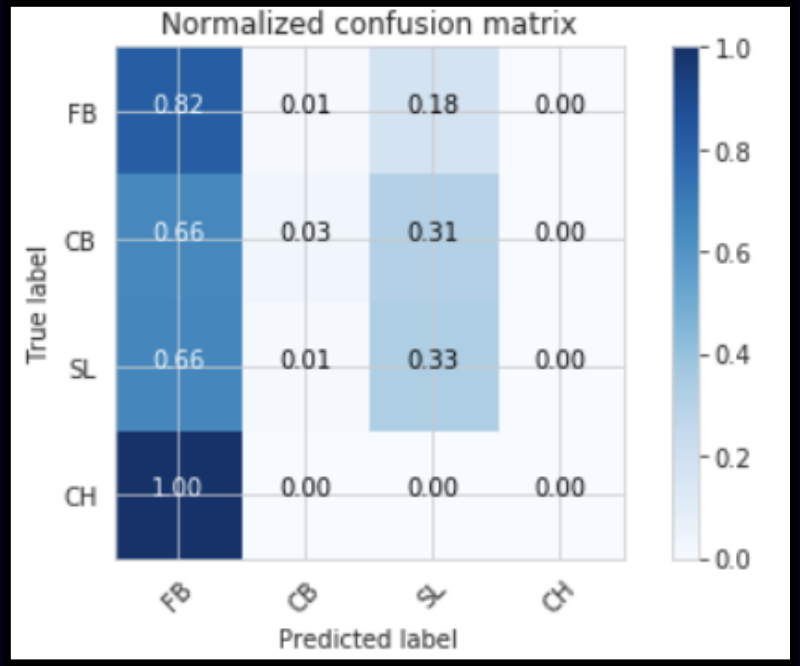
Multi-class Confusion Matrix

Confusion Matrix for Clayton Kershaw

- Most pitch-types were predicted to be Fastballs (FB)
- Second-most predicted were Sliders (SL)

As a result, the model is unable to effectively distinguish between FB's and SL's.





Interpretation of Performance Measures

	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Accuracy

Ratio of correctly predicted observation to total observations

(TP + TN) / (TP + FP + FN + TN)

Precision

Ratio of correctly predicted ‘positive’ observations to total predicted positive observations

TP / (TP + FP)

Recall

Ratio of correctly predicted ‘positive’ observations to all observations in actual ‘yes’ class

TP / (TP + FN)

F1 Score

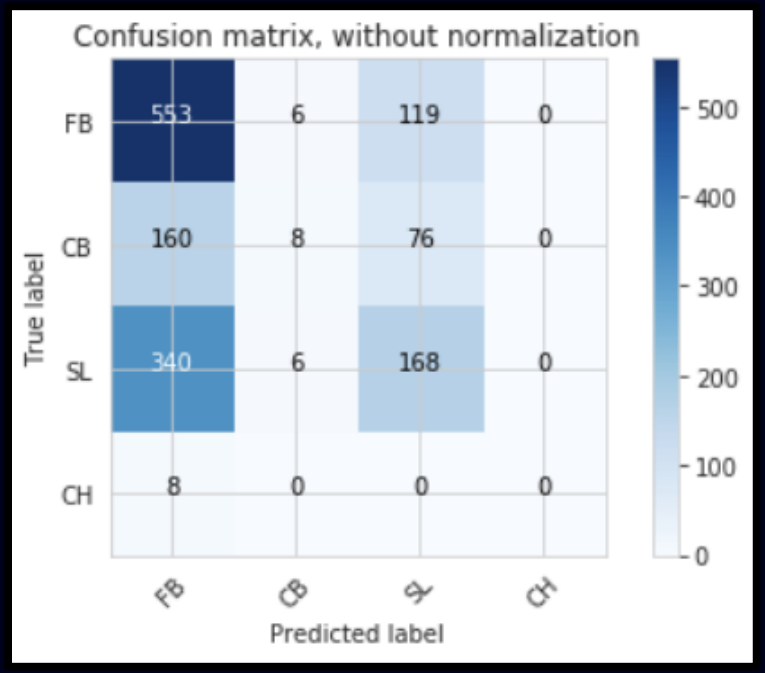
Weighted average of Precision and Recall, considers False Positives and False Negatives

2*(Recall * Precision) / (Recall + Precision)

Problem with Accuracy

Classification Report for Clayton Kershaw

	precision	recall	f1-score	support
1	0.52	0.82	0.64	678
4	0.40	0.03	0.06	244
5	0.46	0.33	0.38	514
7	0.00	0.00	0.00	8
micro avg	0.50	0.50	0.50	1444
macro avg	0.35	0.29	0.27	1444
weighted avg	0.48	0.50	0.45	1444



Accuracy = (TP+TN) / (TP+FP+FN+TN)

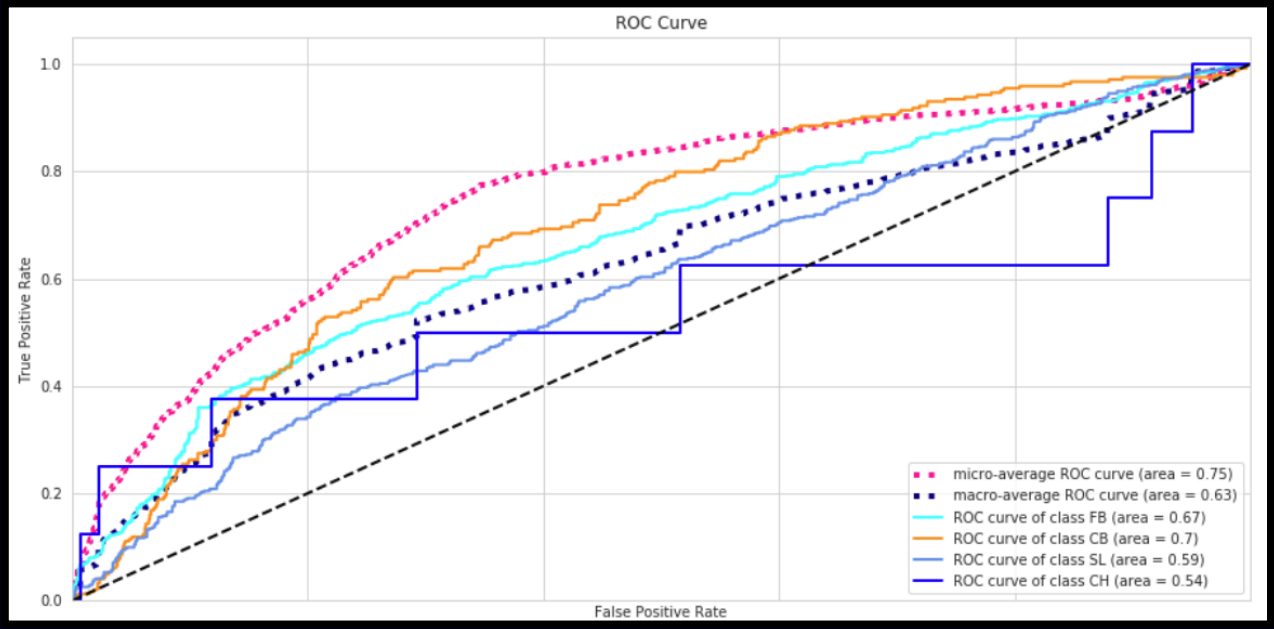
Pitch-type	TP	TN	FP	FN	Accuracy
Fastball (FB)	553	258	508	125	0.562
Curveball (CB)	8	1188	12	236	0.828
Slider (SL)	168	735	195	346	0.625
Change-up (CH)	0	1436	0	8	0.994

Average Accuracy 0.75

Receiver Operating Characteristic (ROC) Curve

- While most classification machine learning models can be validated by accuracy estimation techniques, this is not the case for this project.
- ROC curves are good for summarizing the trade-off between the True Positive Rate and False Positive Rate at different probability thresholds.

Indicates difficulty distinguishing between classes

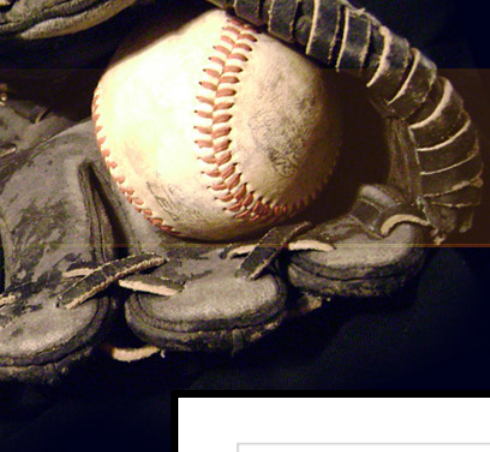


- **Not appropriate when working with imbalanced classes.**

Precision-Recall

Average Precision Score

- Precision-Recall is a useful measure of success of prediction when the classes are very imbalanced.
- High AUC represents High Recall (low False Negative rate) and High Precision (low False Positive rate).
- **`High Recall + Low Precision`** = Many results with many incorrect predictions
- **`Low Recall + High Precision`** = Few results but with many correct predictions



Precision-Recall

Per Class

- Breakdown of Precision-Recall Curves for each Class

Model Performance Comparison

Best Accuracy = 1

Cells in Green
Top Accuracy Score

Cells in Red
Worst Accuracy Score

Accuracy Scores
Predictive Models vs. Majority Class Baseline

Pitcher	Accuracy Baseline	LR (OVR) Accuracy	GB Accuracy	RF (OVR) Accuracy	Ensemble Accuracy	Pitch-types
Aaron Nola	0.528	0.529	0.531	**0.331**	**0.537**	3
Carlos Carrasco	0.486	0.486	**0.489**	**0.279**	**0.489**	4
Carlos Martinez	0.538	0.539	**0.540**	**0.497**	0.538	5
Chris Archer	0.479	0.522	0.568	**0.440**	**0.578**	3
Chris Sale	0.513	0.513	**0.518**	**0.309**	0.517	3
Clayton Kershaw	0.472	0.481	0.502	**0.302**	**0.508**	4
Corey Kluber	0.462	0.465	**0.496**	**0.214**	**0.496**	5
Dallas Keuchel	0.541	0.541	**0.554**	**0.360**	0.551	4
David Price	0.503	0.504	**0.512**	**0.497**	0.510	4
Gerrit Cole	**0.604**	**0.604**	**0.584**	**0.583**	**0.583**	4
Jacob deGrom	**0.555**	**0.555**	**0.548**	**0.543**	**0.546**	4
Jake Arrieta	0.617	0.617	**0.621**	**0.621**	**0.621**	4
Jose Quintana	0.660	0.661	**0.662**	**0.662**	**0.662**	3
Justin Verlander	0.589	0.590	**0.597**	**0.487**	0.595	4
Marcus Stroman	0.578	0.578	0.589	**0.503**	**0.590**	5
Max Scherzer	0.517	0.517	0.518	**0.338**	**0.520**	5
Michael Fulmer	0.594	0.594	**0.598**	**0.598**	0.597	3
Stephen Strasburg	0.539	**0.540**	**0.527**	**0.350**	**0.532**	4
Yu Darvish	0.544	0.545	**0.550**	**0.271**	0.547	6
Zack Greinke	0.486	0.486	**0.498**	**0.244**	0.490	4

- Gradient Boosting and Ensemble approach showed best results.
- Random Forests (ensemble of randomized decision trees) performed poorly.
- Accuracy results inconsistent.

Model Performance Comparison

Best Logloss = 0

Cells in Green
Worst Logloss Error

Cells in Red
Top Logloss Error

Multi-class Logarithmic Loss
Predictive Models vs. Mean Baseline

Pitcher	Logloss Baseline	LR (OVR) Logloss	GB Logloss	RF (OVR) Logloss	Ensemble Logloss	Pitch-types
Aaron Nola	0.694	0.576	0.564	0.574	0.562	3
Carlos Carrasco	0.609	0.509	0.487	0.500	0.488	4
Carlos Martinez	0.597	0.398	0.360	0.406	0.368	5
Chris Archer	0.730	0.549	0.512	0.526	0.514	3
Chris Sale	0.681	0.591	0.566	0.577	0.564	3
Clayton Kershaw	0.782	0.451	0.421	0.440	0.421	4
Corey Kluber	0.566	0.425	0.393	0.417	0.393	5
Dallas Keuchel	0.638	0.481	0.454	0.471	0.455	4
David Price	0.643	0.493	0.478	0.495	0.483	4
Gerrit Cole	0.681	0.450	0.452	0.460	0.454	4
Jacob deGrom	0.651	0.476	0.471	0.478	0.473	4
Jake Arrieta	0.686	0.442	0.420	0.436	0.427	4
Jose Quintana	0.802	0.497	0.477	0.491	0.482	3
Justin Verlander	0.688	0.441	0.426	0.439	0.427	4
Marcus Stroman	0.612	0.388	0.368	0.382	0.369	5
Max Scherzer	0.594	0.409	0.383	0.408	0.387	5
Michael Fulmer	0.718	0.555	0.532	0.547	0.534	3
Stephen Strasburg	0.636	0.486	0.479	0.490	0.481	4
Yu Darvish	0.618	0.329	0.324	0.324	0.324	6
Zack Greinke	0.611	0.508	0.478	0.499	0.482	4

- Imbalanced classes, Logloss a better performance measure than Accuracy.
- All predictive models outperformed the mean Logloss Baseline.
- For most pitchers, Gradient Boosting outperformed all other models.
- Logloss errors were best for Pitchers who throw >=5 different pitch-types.

Conclusion

- On average, more than 50% of all pitches thrown are Fastballs.
- Due to imbalanced classes, we should not use Accuracy and ROC curves to measure a models performance.
- When comparing Logarithmic Log Loss (cross-entropy) for predictive models against the mean baseline, it is possible to minimize errors by as much as 10% - 40%.
- Pitcher’s pitch count provides greatest importance to predicting the next pitch.
- Similarity in pitch speed and location between pitch-types from same pitch group make it difficult to distinguish.

Recommendations

- Compare in-game pitches to predicted pitches to monitor success rate.
- Select a subset of hitters to essentially perform a randomized controlled trial to estimate the effect size (ROI).
- Monitor before and after batting averages for selected hitters.
- **Increased attendance and a boost in revenue will be biggest measure of success.**

Future Work

- Increase data to include all starting and relief pitchers for more than just three seasons. ***(decrease variance)***
- Extend feature data to include weather conditions, pitch movement, fielders position, score differential, batting averages, and team record, just to name a few. ***(decrease bias)***
- Consider multi-label, multi-class classification problem by attempting to predict:

  Pitch-type + Pitch Group + Pitch Speed + Pitch Location

- Apply a combination of Supervised and Unsupervised Learning to address and include unlabeled pitch data.

Questions?

Mark Rojas

Cell: (832) 330-2870

Email: rojas.mm@gmail.com

Skype: markrojas

Linkedin: https://www.linkedin.com/in/mark-rojas/

Github: https://github.com/markrojas

Pitch Prediction Repository:

https://github.com/markrojas/Springboard_DSCT/tree/master/capstone_projects/capstone_I