The Open Source Hit By Pitch Database: A Typology

Using unsupervised learning to uncover underlying types of hit-by-pitch events

By Jon Godin

Background / Data Science Problem

Baseball Prospectus: The Open Source Hit By Pitch Database

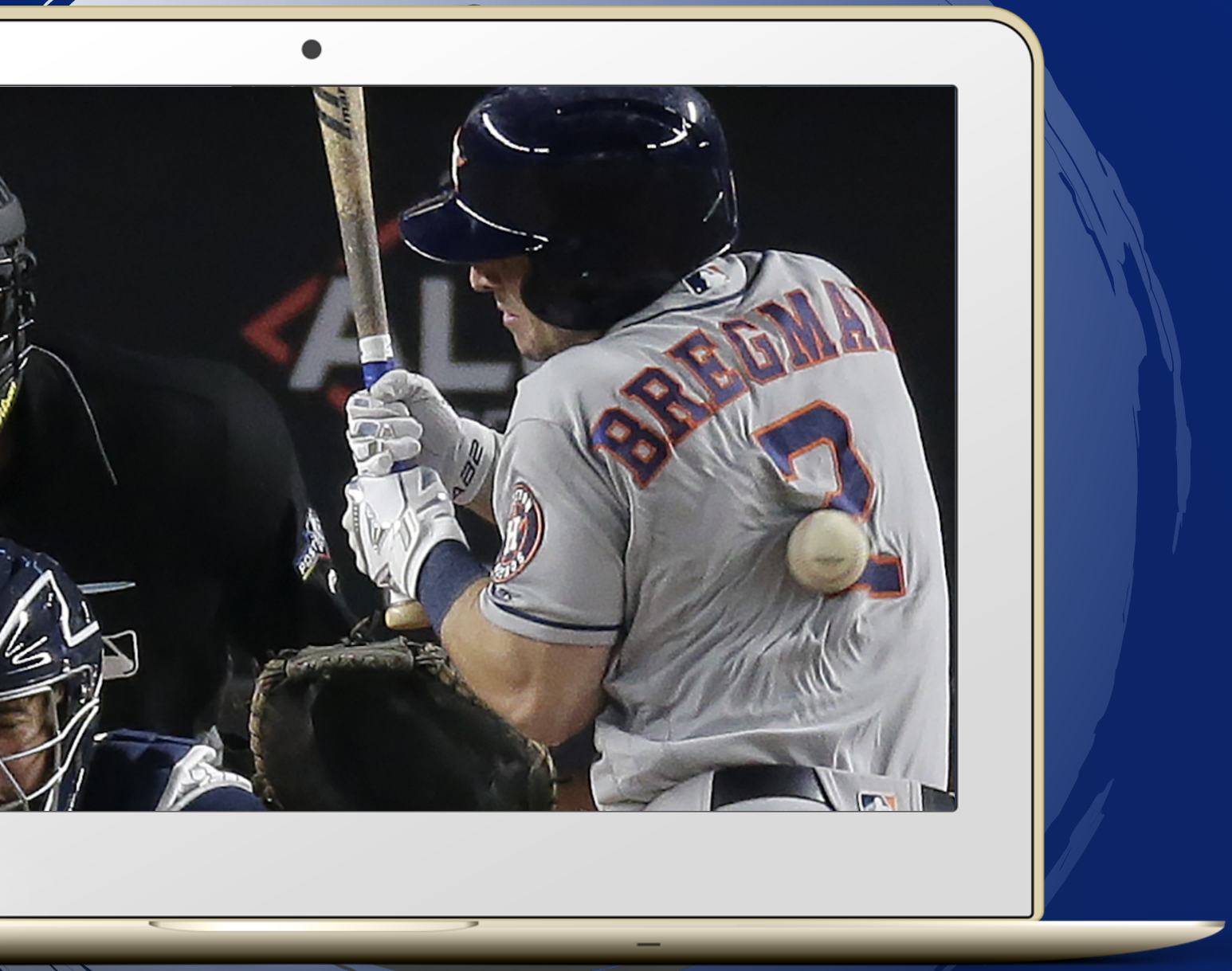
- http://www.baseballprospectus.com/news/article/58297/veteran-presence-the-open-source-hit-by-pitch-database/
- Data include every hit-by-pitch occurrence in MLB since 1969
- Over 1.8 million cells of data
- Opened for research in April 2020
- Article author Rob Mains asked for help to “See what you can make of it”

Stathead.com’s Baseball Event Finder has additional data not in the BP database

- https://stathead.com/tiny/si7Zm
- The additional statistics collected by Stathead include:
  - # of HBPs already in game
  - Pitch count at time of HBP
  - Whether an RBI occurred as a result
  - Win Probability Added (WPA)
  - Base Out Runs Added (RE24)
  - Leverage Index (LI)

Data Science Problem

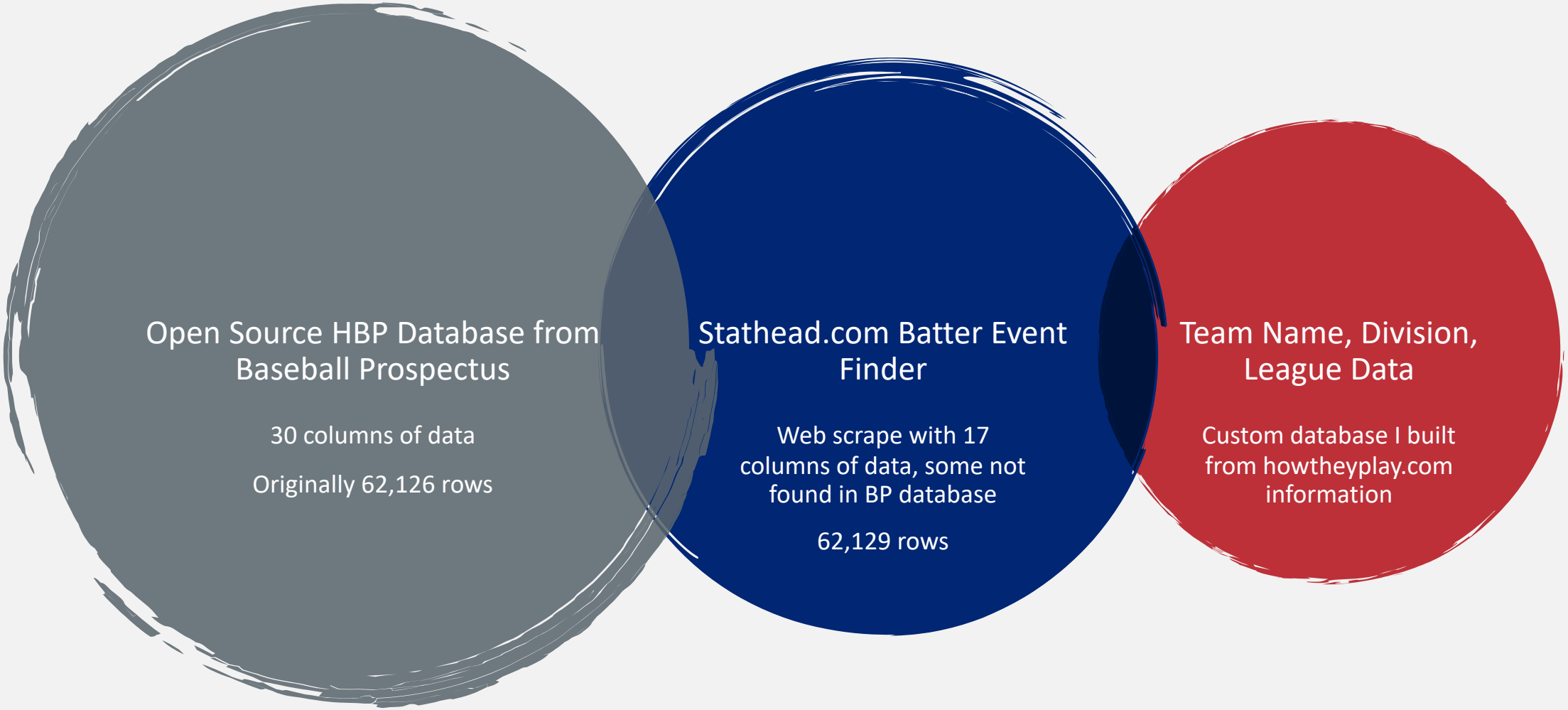
- Problem: Can the 62k+ HBP events be reduced into a typology where all events can be classified as a particular type of HBP?
- Approach:
  - Combine BP’s database with a webscrape of Stathead’s data
  - Create additional features via feature engineering
  - Use unsupervised learning to segment or type the data
  - Then used supervised learning to create a classification algorithm based on the typology



THE DATA

Three sources of data needed to be merged

See Appendix for details



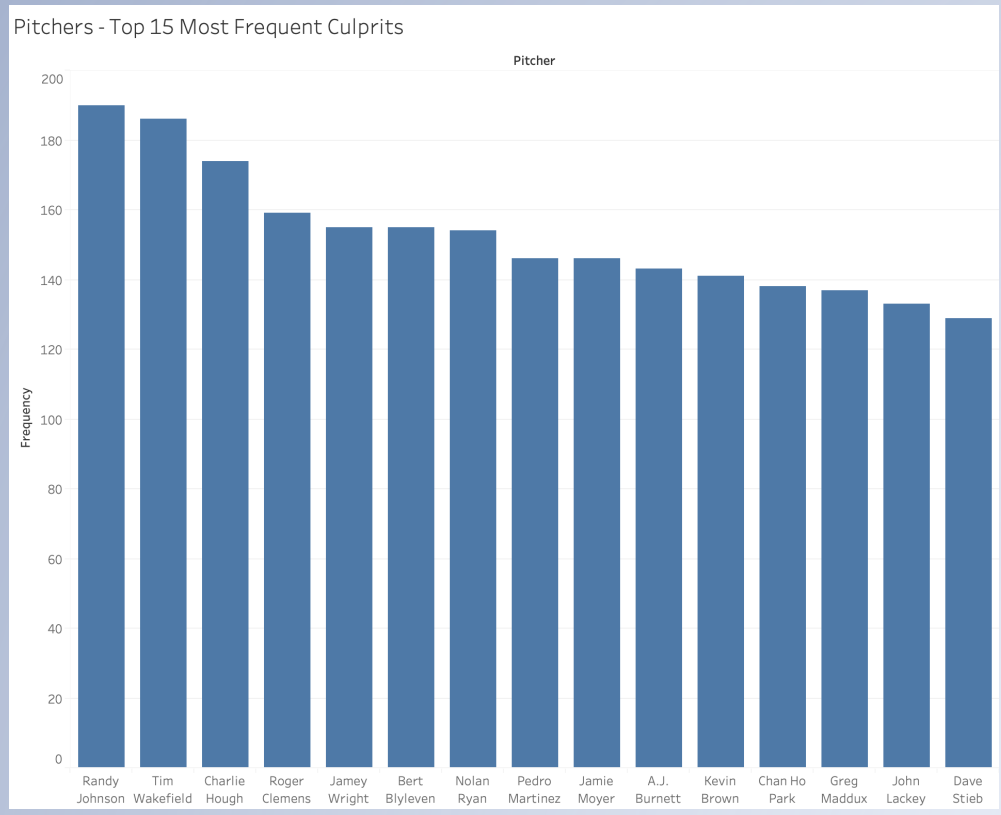
It's all baseball statistics, shouldn't that be easy to merge?

Also, the data is from reputable sabermetric sites, shouldn't the data be clean?

- **The lesson: NEVER TRUST YOUR DATA**
- Things you would expect to be simple and consistent were not so:
  - Different conventions used for player names
  - Different three letter acronyms used for team names
  - Statistics for the same HBP event not matching
  - Even the number of events occurring between 1969 to 2019 didn't match across the two sources, as you may have noticed
  - Missing data
  - Inconsistent coding
- Most of these could be resolved, but it required a lot of time and additional research, combing through historical box scores on baseball-reference.com
- Once the data was cleaned-merged-cleaned again, I constructed over 15 additional measures using feature engineering
- Details of the cleaning & engineering are in the Appendix

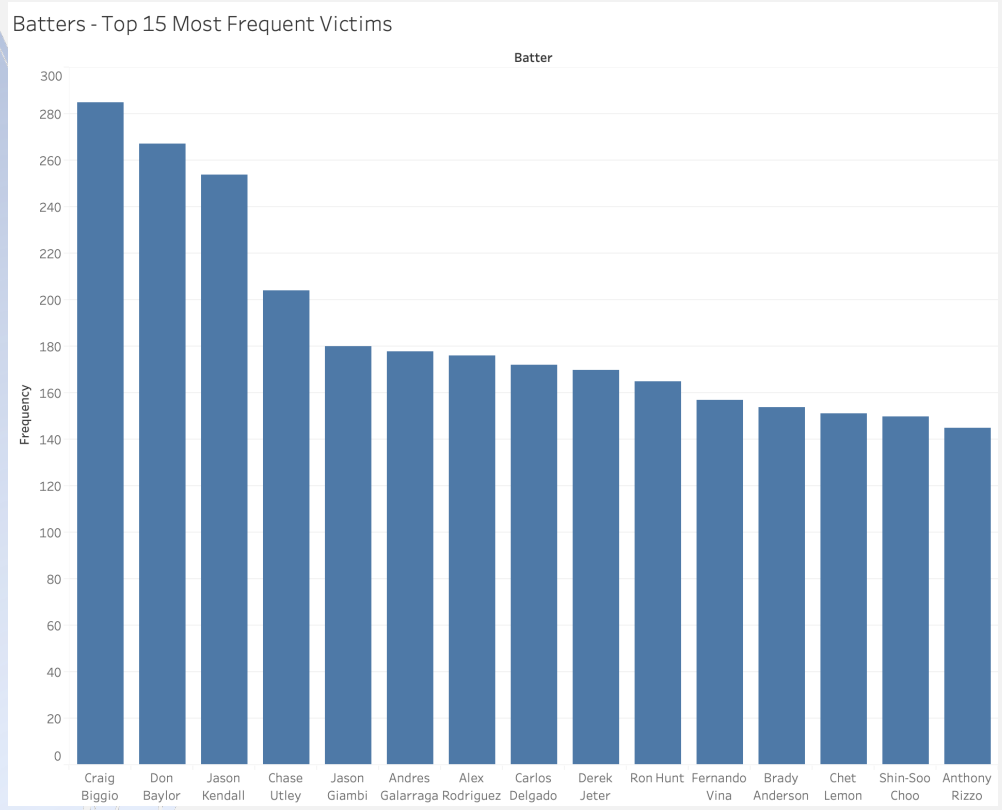
EXPLORATORY DATA ANALYSIS

Worst Culprits / Victims of Hit By Pitches – Overall

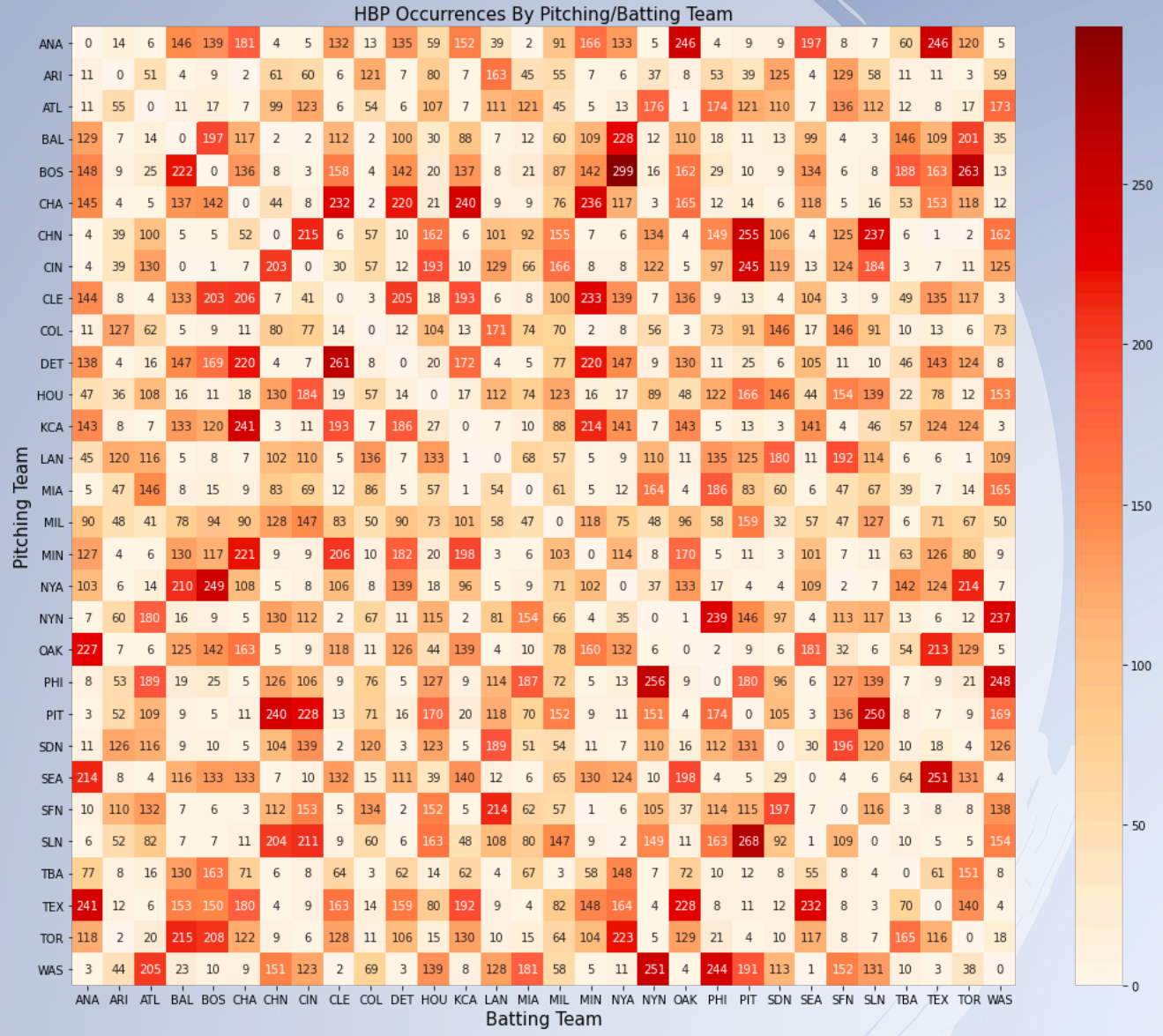


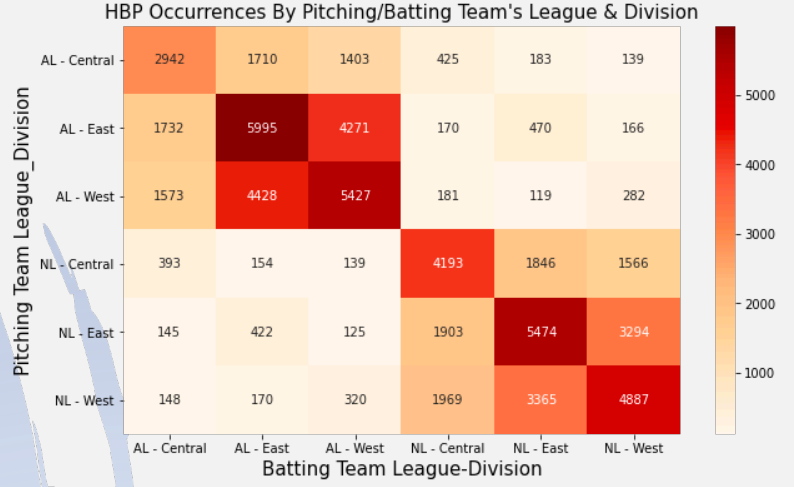
Top 15 Pitcher --> Batter Combinations

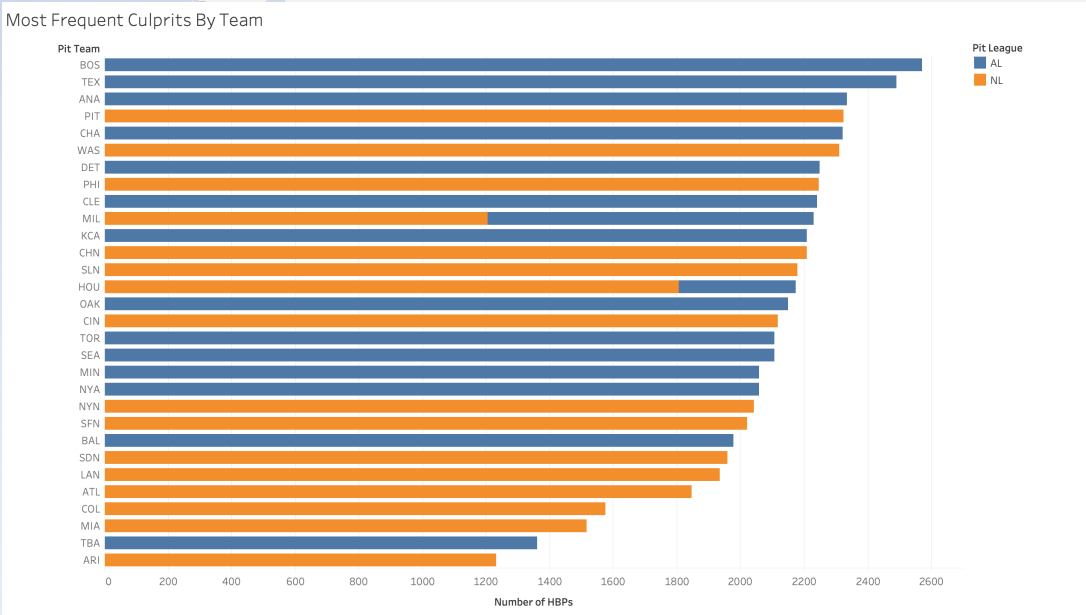
David Wells --> Jason Giambi	8
Tim Hudson --> Chase Utley	7
Tanner Roark --> Derek Dietrich	7
Pedro Astacio --> Craig Biggio	7
Charlie Hough --> Don Baylor	7
Tim Wakefield --> Shannon Stewart	6
Mark Buehrle --> Travis Hafner	6
Jeff Weaver --> A.J. Pierzynski	6
Jamey Wright --> Jason Kendall	6
Alfredo Simon --> Starling Marte	6
Chuck Finley --> Brady Anderson	5
Chris Carpenter --> Melvin Mora	5
Charlie Morton --> Jon Jay	5
Casey Fossum --> Reed Johnson	5
Bert Blyleven --> Chet Lemon	5



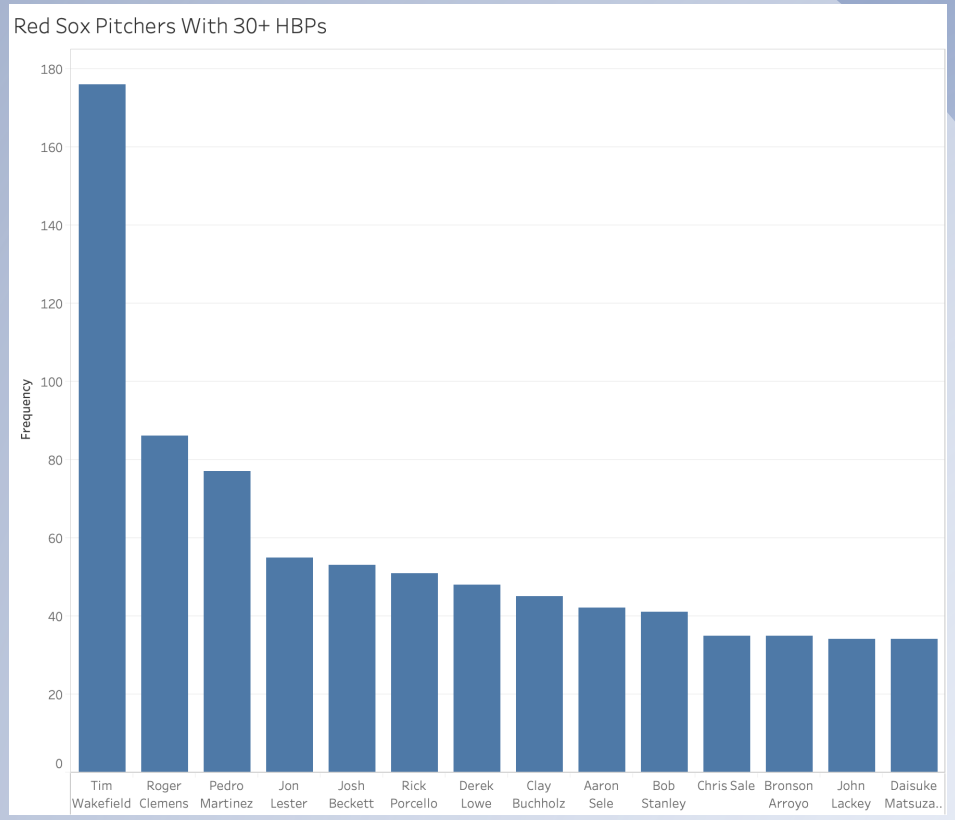
Most Frequent Team Combos? Red Sox Pitchers Hitting Yankee Batters!

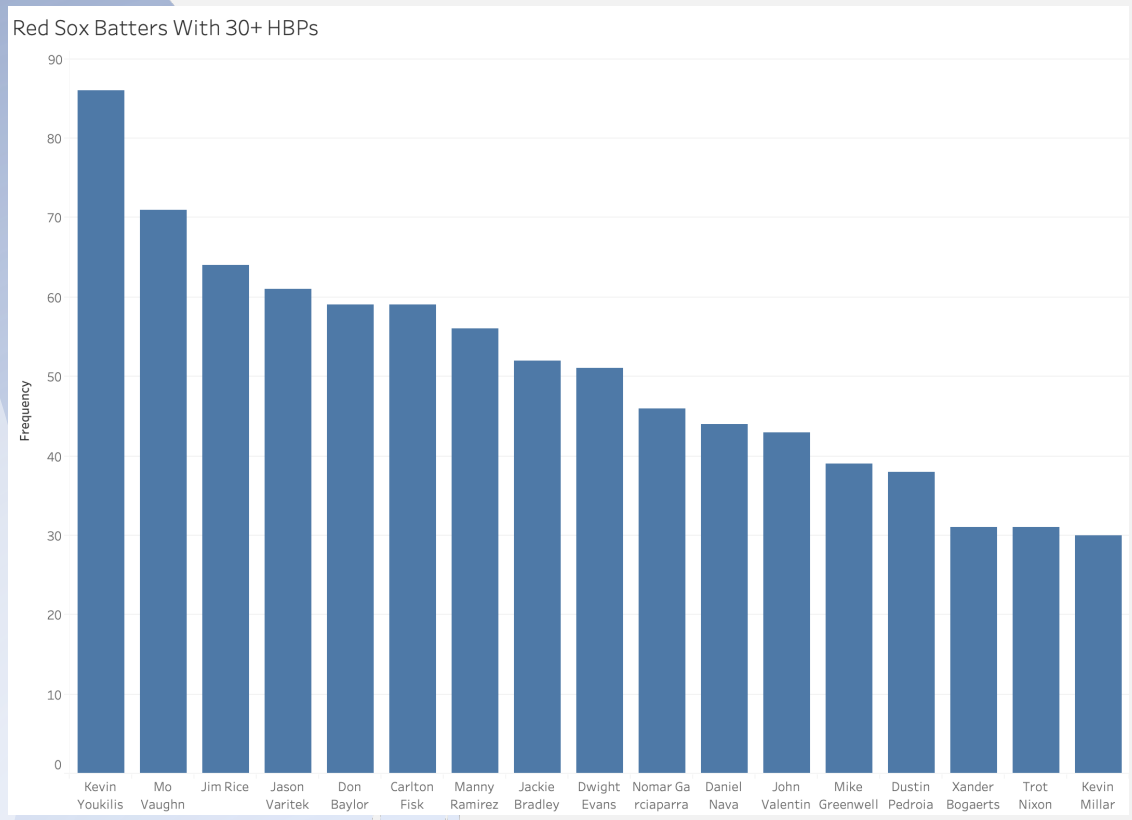


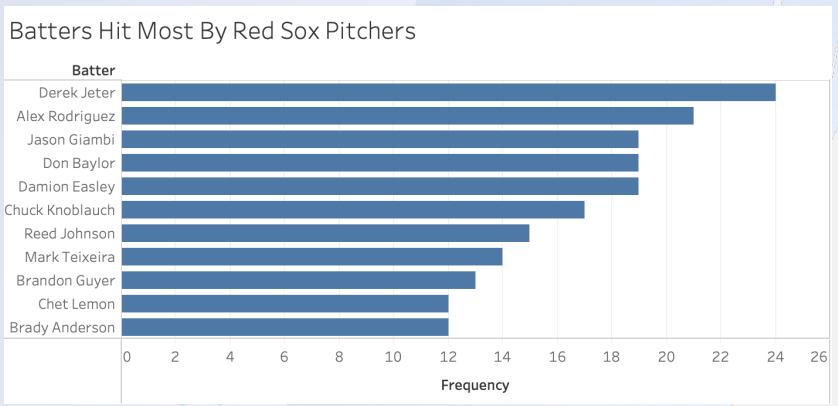




About those Red Sox…







Lineup spots, Positions, and Handedness

Batter Position by Lineup Spot For Epoch All

	Position											
Lineup Spot	P	C	1B	2B	3B	SS	LF	CF	RF	DH	PH_PR	Grand Total
1	2	67	99	1,818	277	880	1,309	2,280	733	183	83	7,731
2		230	322	1,977	759	1,258	725	1,015	739	185	90	7,300
3		230	1,555	556	858	375	1,166	871	1,275	587	39	7,512
4		423	2,067	197	1,027	133	1,167	390	1,106	1,034	38	7,582
5	3	825	1,303	295	1,081	193	982	516	1,047	732	74	7,051
6		1,078	912	462	1,223	288	924	580	1,006	427	108	7,008
7	4	1,659	545	682	968	590	609	538	676	266	164	6,701
8	8	2,248	195	759	552	1,197	288	492	302	82	258	6,381
9	726	708	61	611	217	857	164	329	161	41	988	4,863
Grand Total	743	7,468	7,059	7,357	6,962	5,771	7,334	7,011	7,045	3,537	1,842	62,129

Throws/Bats Handedness

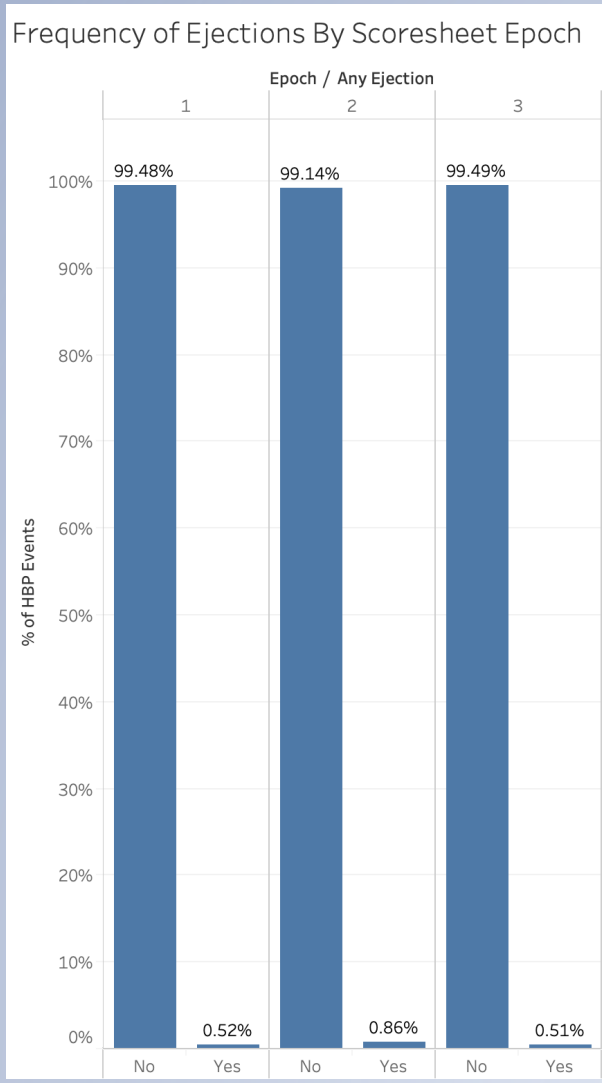
	Throws		
Bats	L	R	Grand Total
L	10.34%	25.24%	35.59%
R	15.31%	49.11%	64.41%
Grand Total	25.65%	74.35%	100.00%

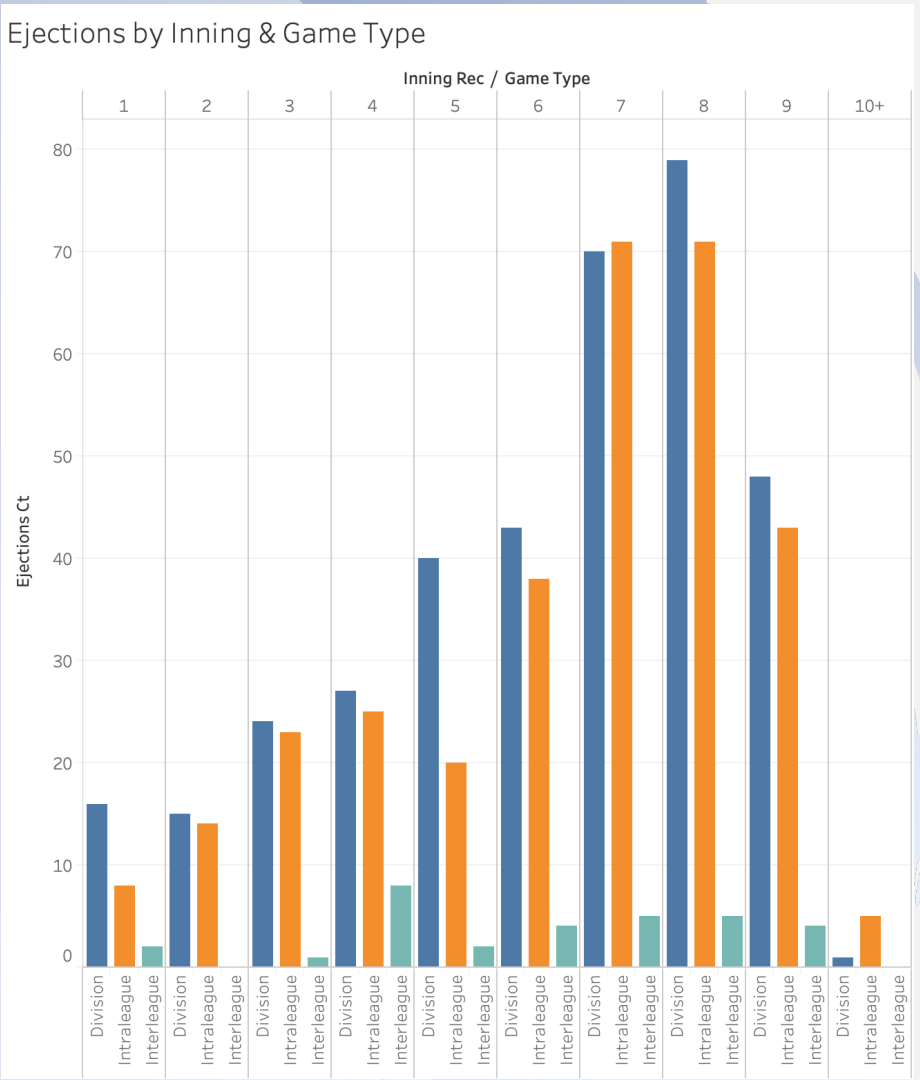
Leadoff hitters get hit the most frequently overall, with centerfielders batting leadoff having the highest frequency of HBPs overall.

However, by position alone, catchers are the most frequently hit by pitches, and among primary position players, shortstops are least hit.

In terms of handedness, there are only slightly higher incidences of Lefty-Lefty and Righty-Righty HBPs than expected (9% and 48%, respectively)

I only came here for the ejections, but they're quite rare (0.67% of HBPs)





Pitcher vs Batter Ejections

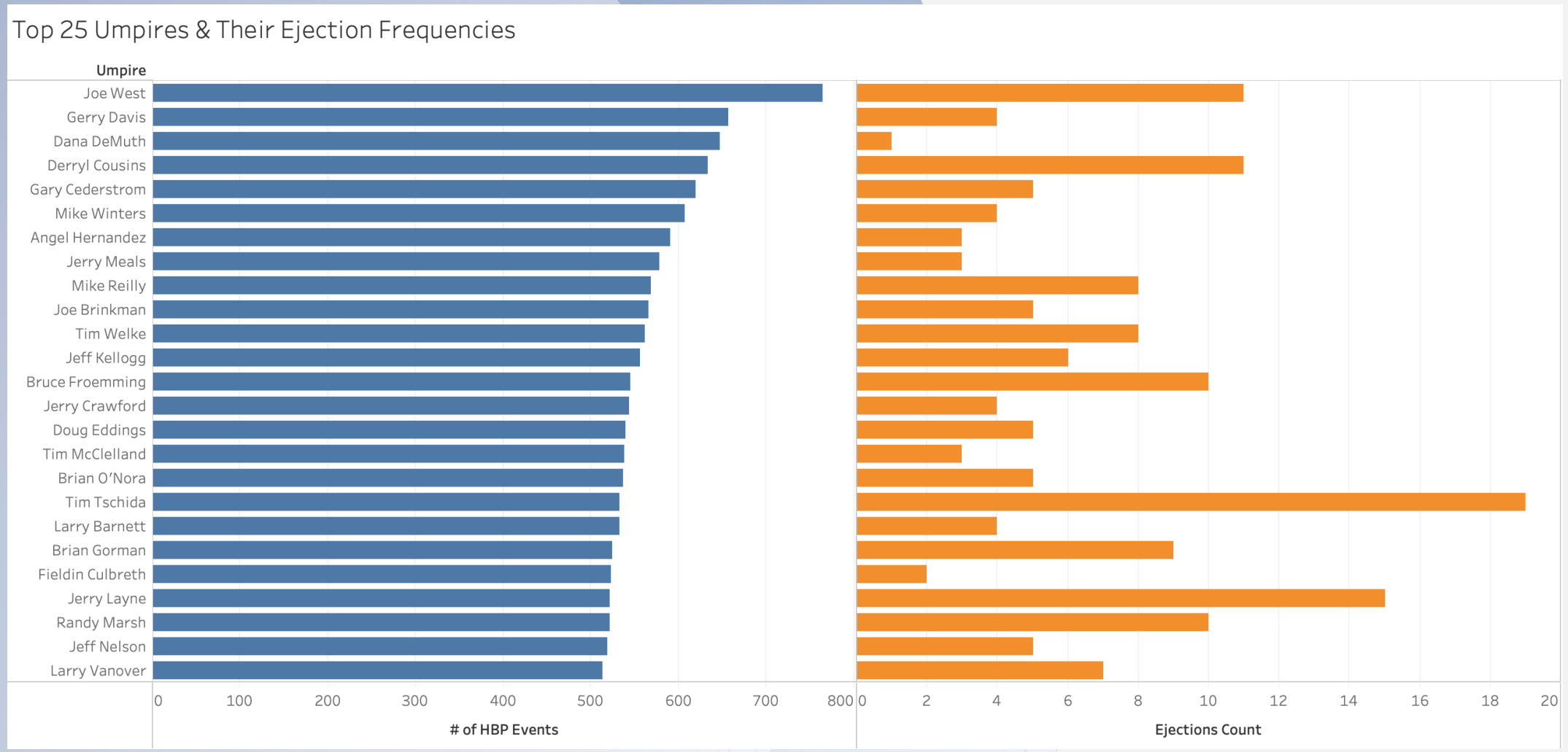
	Pit Ejected		
Bat Ejected	No	Yes	Grand Total
No	99.49%	0.34%	99.83%
Yes	0.08%	0.09%	0.17%
Grand Total	99.57%	0.43%	100.00%

Ejections tend to be more frequent in the 7th-9th innings for Division and Intraleague games

FAVORITE DESCRIPTION

There had been threatening pitches in the game one day earlier; the first pitch in this inning hit Dwight Gooden on the knee; Gooden, who had already hit two Phillies earlier in the game, took off his helmet and charged the mound; he was tackled and repeatedly punched by Phillies catcher Darren Daulton; a 20-minute melee erupted; 2B umpire Joe West threw Phillies pitcher Dennis Cook to the ground and HP umpire Randy Marsh considered asking for help from the police; Phillies ejections: Pat Combs, Dennis Cook, Darren Daulton, coach Mike Ryan; Mets ejections: Dwight Gooden, Darryl Strawberry, Tim Teufel; all ejections by Marsh

Some umpires were more likely to throw out players than others



Visualizations only relevant for Epoch 3 (PITCHf/x Era)

HBPs from Fastballs occur more frequently in hitters' counts; In late-and-dangerous situations, pitchers tend to turn up the heat

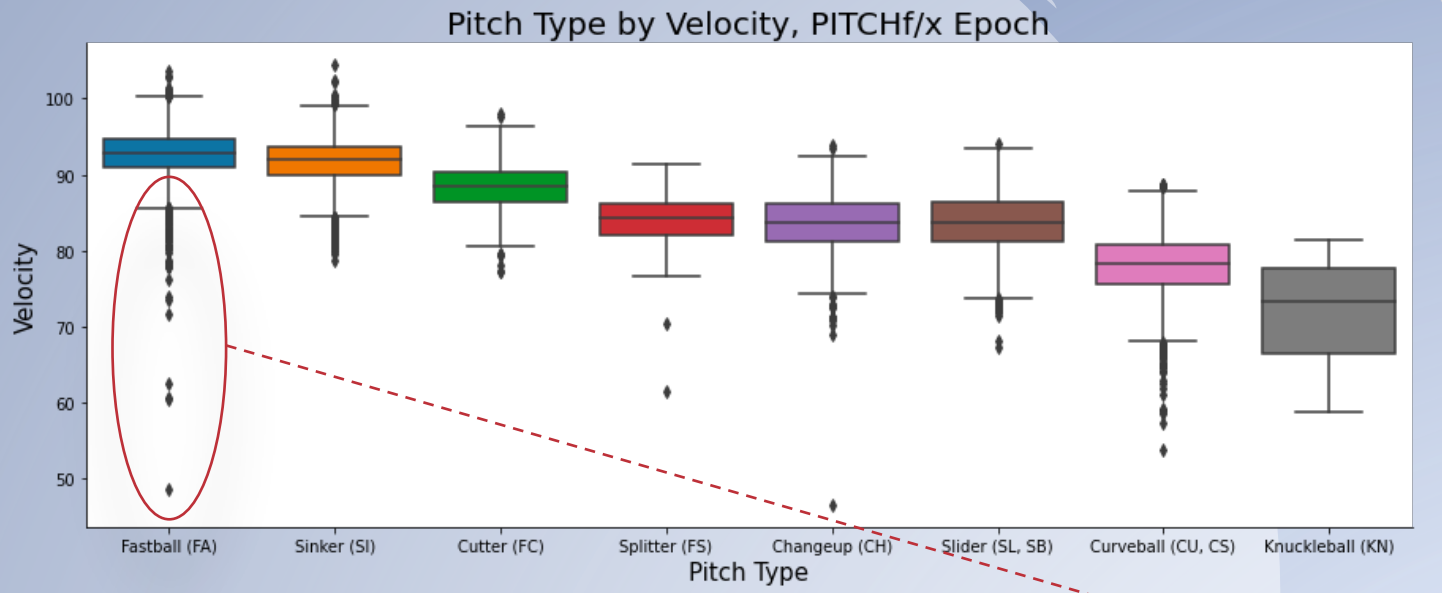
Pitch Type by Count - Epoch 3

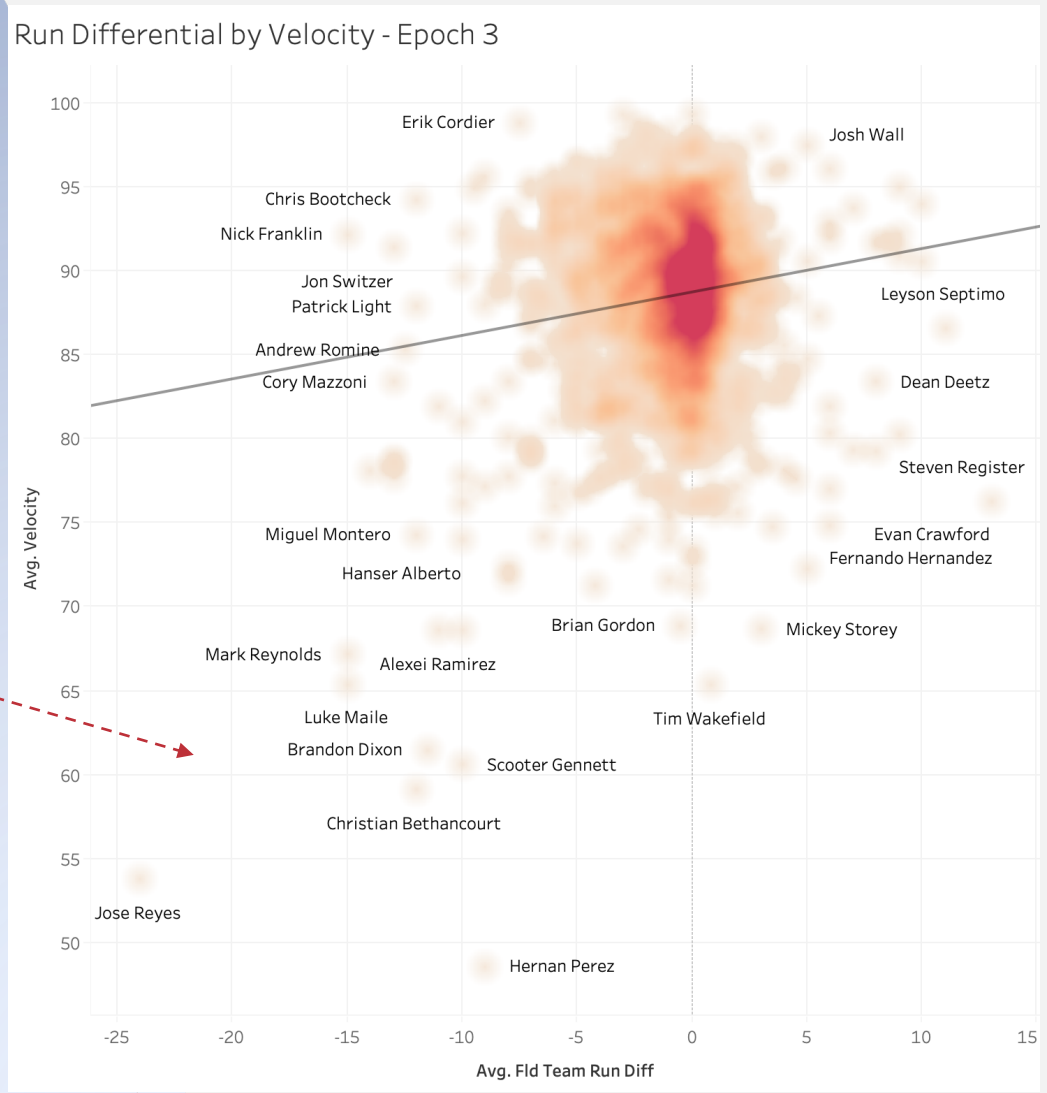
	Pitch Type								
The Count	Fastball (FA)	Sinker (SI)	Slider (SL, SB)	Curveball (CU, CS)	Cutter (FC)	Changeup (CH)	Splitter (FS)	Knuckleball (KN)	Grand Total
0-0	35.23%	29.45%	11.15%	14.40%	5.65%	2.37%	1.01%	0.72%	100.00%
0-1	37.71%	30.16%	10.80%	9.28%	7.16%	3.30%	0.79%	0.79%	100.00%
0-2	40.34%	19.03%	18.07%	13.02%	5.15%	2.53%	1.29%	0.57%	100.00%
1-0	34.83%	32.54%	10.71%	8.67%	6.54%	4.67%	0.93%	1.10%	100.00%
1-1	34.65%	30.80%	11.98%	9.58%	6.11%	5.34%	1.15%	0.38%	100.00%
1-2	35.15%	19.52%	18.80%	13.02%	6.92%	4.31%	1.96%	0.33%	100.00%
2-0	43.28%	35.08%	5.90%	3.93%	4.59%	5.57%	1.31%	0.33%	100.00%
2-1	34.49%	31.74%	10.00%	7.83%	6.81%	7.54%	1.16%	0.43%	100.00%
2-2	33.97%	22.84%	15.92%	12.35%	6.47%	5.79%	2.22%	0.45%	100.00%
3-0	72.58%	20.97%	3.23%		1.61%			1.61%	100.00%
3-1	40.00%	35.38%	9.74%	7.69%	4.10%	1.03%	1.03%	1.03%	100.00%
3-2	33.38%	24.58%	15.39%	11.77%	4.92%	8.02%	1.29%	0.65%	100.00%
Grand Total	36.12%	26.50%	13.63%	11.53%	6.20%	4.11%	1.32%	0.60%	100.00%

Velocity by Inning and Runners on Base - Epoch 3

	Inning Rec										
Runners On	1	2	3	4	5	6	7	8	9	10+	Grand Total
empty	88.93	88.55	88.05	87.71	87.63	87.90	88.83	88.69	89.75	89.05	88.42
1st	88.87	88.52	88.37	88.35	88.75	88.80	89.12	89.46	90.09	89.47	88.91
2nd	89.10	88.21	88.55	88.44	88.73	87.61	88.58	89.52	89.75	87.87	88.71
3rd	88.88	89.00	89.50	88.75	89.04	90.29	90.28	89.49	91.01	90.23	89.50
1st_2nd	88.09	87.42	88.25	88.69	88.42	89.46	89.36	89.71	90.03	89.17	88.87
1st_3rd	88.61	88.41	89.28	88.87	88.56	89.37	89.64	91.10	88.17	91.38	89.23
2nd_3rd	89.34	88.82	88.07	88.29	89.84	87.41	88.48	90.10	91.74	93.54	89.15
full	88.69	89.46	87.59	87.32	88.56	88.52	88.02	91.14	90.12	89.75	88.87

Note that not all Fastballs are necessarily all that fast…





These outliers are most likely position players who are asked to pitch in the late innings of blowout games – they throw fastballs since they probably don't have a real pitch repertoire, but they don't necessarily have comparable Big League heat either

In Epoch 3, PIT has hit the most batters, but BOS→TBA is most frequent

Pitching/Batting Team HBP Totals Epoch 3

		Bat Lg Div / Bat Team																														
		AL - East					AL - Central					AL - West					NL - East					NL - Central						NL - West					Grand
Pit Lg Div	Pit Team	BAL	BOS	NYA	TBA	TOR	CHA	CLE	DET	KCA	MIN	ANA	HOU	OAK	SEA	TEX	ATL	MIA	NYN	PHI	WAS	CHN	CIN	HOU	MIL	PIT	SLN	ARI	COL	LAN	SDN	SFN	Total
AL - East	BAL		67	91	91	88	43	37	22	19	31	31	24	28	21	28	7	4	3	6	26	2	2		3	10	1	3	2	6	11	2	709
	BOS	67		108	114	88	41	44	28	30	28	34	18	34	21	40	10	5	5	17	3	8	1	2	4	3	2	4	2	6	6	2	775
	NYA	72	89		87	75	18	30	30	32	21	28	15	28	34	41	3	4	19	8	3	4	6	2	5	3	4	4	4	4	2		675
	TBA	59	73	64		61	32	22	26	21	28	38	11	31	25	25	3	18	2	7	4	2	5	2	2	5	3	6	2	2	5	3	587
	TOR	102	72	92	100		41	41	26	30	22	23	14	34	29	29	13	8	1	8	1	7	5	1	7	3	5	2	3	7	9	6	741
AL - Central	CHA	25	41	30	32	27		90	80	79	81	27	14	18	21	38	4	8	3	10	8	25	6	1	2	9	4	3	1	9	2	4	702
	CLE	21	45	36	28	30	76		70	67	76	25	12	23	31	28	4	4	4	4	3	6	27		2	4	3	4	1	2	2	3	641
	DET	28	37	33	25	19	80	87		55	62	32	13	31	31	33	7	3	7	4	3	1	3	3	1	17	4	1	4	3	4	8	639
	KCA	38	30	35	35	33	87	83	53		59	32	13	23	28	24	6	6	4	3	1	2	6	6	3	3	26	2	3	2	2	2	650
	MIN	18	20	26	38	10	98	71	71	70		27	17	32	22	30	3	4	6	4	7	4	3		20	5	4	2	10	2		4	628
AL - West	ANA	18	29	24	28	15	36	36	27	32	23		56	76	82	105	3	2	3	2	4	3	3		9	8	6	4	7	22	3	3	669
	HOU	13	9	13	18	11	15	12	12	12	9	45		47	41	61	2	2	3	6	1	3	4		5	3	4	7	4		2	3	367
	OAK	19	26	29	29	25	40	26	17	17	20	79	41		70	76	4	8	5	2	2	4	6		2	3	4	1	3		3	14	575
	SEA	25	28	29	35	18	36	33	25	33	17	67	38	62		108	4	6	6	4	2	7	3		3	5	5	4	7	5	16		631
	TEX	37	24	45	32	31	41	52	30	30	26	97	59	81	86		3	2	2	6	1	3	4	3	5	7	3	8	6	3	5	1	733
NL - East	ATL	4	6	5	6	7	6	6	2	7	4	10	3		6	6		66	70	67	68	29	27	5	28	37	41	33	21	18	25	23	636
	MIA	3	4	4	21	5	6	4	3		3	4	4	4	4	6	81		85	85	73	38	30	9	32	29	36	26	30	24	19	19	691
	NYN	6	2	18	8	4	3	1	7	2	2	6	4	1	3	5	63	72		88	75	25	25	10	29	32	36	31	26	21	28	23	656
	PHI	4	15	6	4	13	2	5	3	4	1	5	1	6	4	8	64	87	95		74	33	24	16	35	38	35	24	29	31	26	25	717
	WAS	15	8	4	4	4	6	2	2	2	2	3	3	1		1	71	88	69	80		38	31	12	26	28	31	26	20	34	22	19	652
NL - Central	CHN	4	4	2	5		31	4	6	2	2	2	5	3	3	1	32	33	26	31	33		86	22	96	84	99	18	22	31	31	27	745
	CIN		1	8	1	9	2	14	5	5	2	2	4	2	8	3	34	26	33	26	37	103		25	102	101	78	16	24	33	24	20	748
	HOU	1	2	3	4				1							5	7	15	8	10	16	14	18		40	16	18	10	13	6	14	20	241
	MIL		2	3	5	5	1	2	6	3	18	5	2	5	1	2	22	22	24	26	33	81	85	18		69	67	20	18	29	13	23	610
	PIT	8	3	4	5	6	7	8	10	8	4	2		2	1	2	37	24	27	38	37	104	98	18	93		97	27	18	32	30	32	782
	SLN	7	5	1	6	3	3	4	2	20	4	2	4	8	1	4	15	32	33	28	24	83	79	24	78	89		30	13	28	25	18	673
NL - West	ARI	3	8	2	5	2	2	1		4	6	6	17	2	1	8	29	20	24	24	35	42	32	7	33	19	24		70	93	67	70	656
	COL	3	4	4	7	3	8	7	5	6	2	4	6	2	6	8	22	26	19	28	18	35	29	10	34	36	29	58		78	72	52	621
	LAN	5	5	7	5	1	5	5	4		2	24	7	1	3	4	14	32	29	33	18	27	22	10	26	32	33	61	60		57	61	593
	SDN	2	6	7	9	4	4	1	1	3	7	4	2	6	14	14	20	16	26	25	24	32	19	9	24	44	35	70	47	75		50	600
	SFN	1	3	3	2	7	3	5	1	3	1	6		22	1	2	13	27	33	28	30	30	40	17	23	27	20	67	59	85	67		626
Grand Total		608	668	736	789	604	773	733	575	596	563	670	407	613	598	745	600	670	674	708	664	795	729	232	772	769	757	572	529	691	592	537	###

The Chicago Cubs have been hit the most in this era with 795

Tampa Bay leads all AL teams with 789

MACHINE LEARNING MODELS

UNSUPERVISED AND SUPERVISED APPROACHES TO BETTER UNDERSTAND THE HBP DATA

Before starting to model

Several considerations were necessary

Binarize Categorical Data

Most ML models can’t deal with strings or categories very well

Split Data by Epoch

Systematic differences in scorekeeping and the inability to impute missing values warrant separate models by era

Mixed Data Types

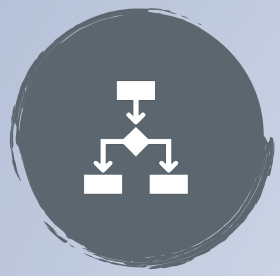
Most unsupervised learning approaches do not handle mixed data types well, so modeling options were limited

Modeling included several steps

Some worked quite well, others didn't produce very useful results

Unsupervised Learning

(to attempt to uncover any underlying classes in the data)



HDBSCAN modeling

New-ish hierarchical version of DBSCAN model

Used custom distance metric called HEOM to deal with mixed data types among X variables

K-prototypes modeling

Alternative version of k-means model using a different distance metric that better handles mixed data types

PCA (visualization)

Since there is very high dimensionality in the data, I tried using PCA to visualize segment differences

Final X list had 114 potential segmentation variables – see Appendix

Supervised Learning

(to predict classes from unsupervised learning)

LDA classification

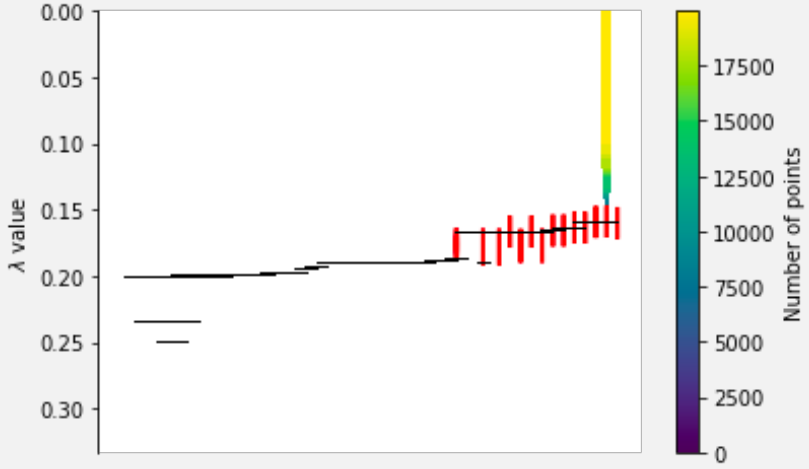
Linear Discriminant Analysis can be used for dimension reduction like PCA, but it also works as a decent classifier frequently used in business for its simplicity

Xgboost classification

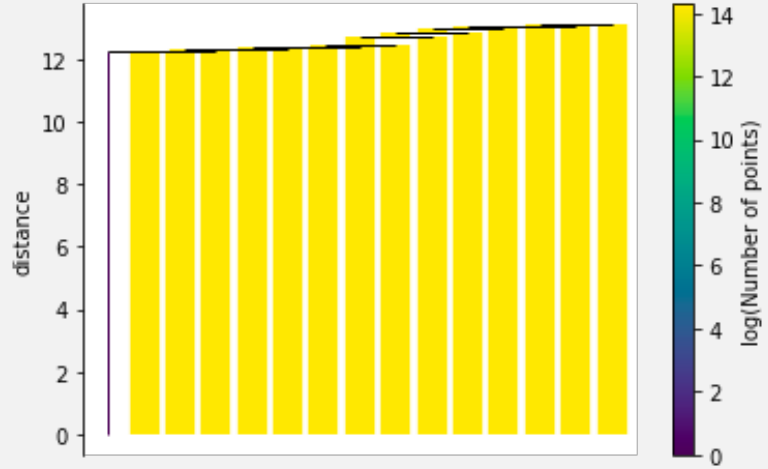
Very powerful and fast implementation of a gradient boosting classifier in this case; great benefit is its ability to handle missing data, along with usual high performance

HDBSCAN + HEOM Results

Condensed Tree Plot

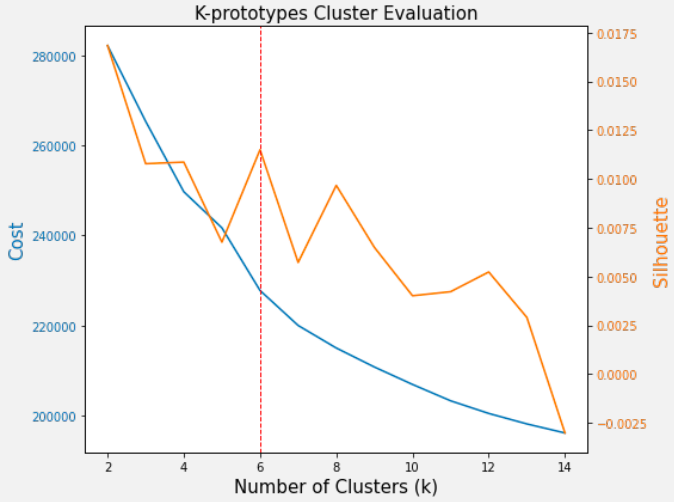


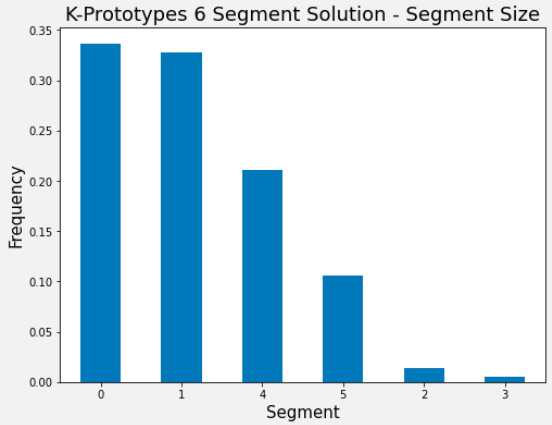
Single Linkage Tree

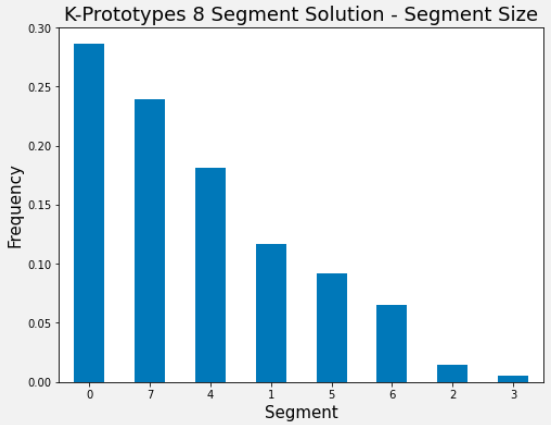


- Unfortunately, this was a case where the model had trouble finding a strong enough signal amidst the noise
- Initial model found 13 clusters in the data (including Noise, which isn't a true cluster)
- However, 88.6% of cases were labeled as being Noise, one cluster had 10.8% of cases, and the remaining clusters uncovered were therefore quite miniscule
- The charts to the left show this, with one initial giant yellow bar representing the noise and not much pulled out beyond that
- Additional model tunings were attempted (including switching from the eom cluster selection method to the leaf method, and adjusting the min cluster size and min samples hyperparameters), but an acceptable solution was not found
- Best had 8 groupings including noise (94%), 3 clusters > 1% and remaning clusters >= 0.5%

K-prototypes Version 1 Results

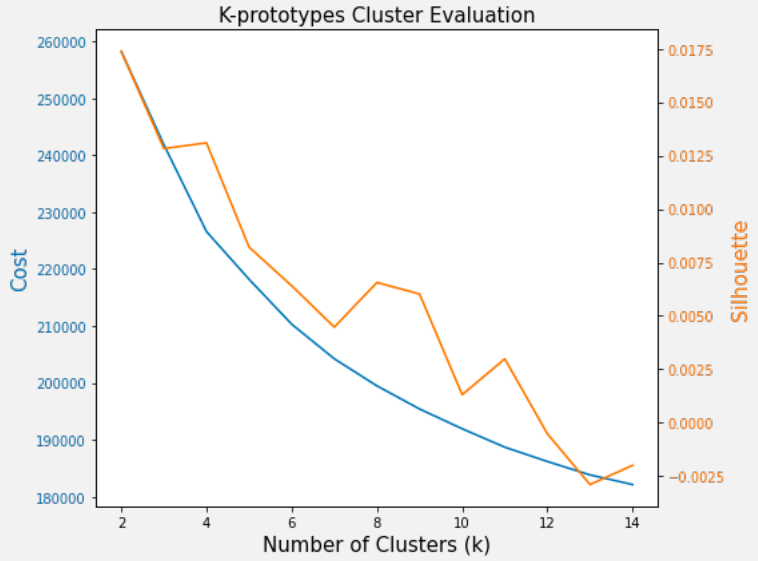


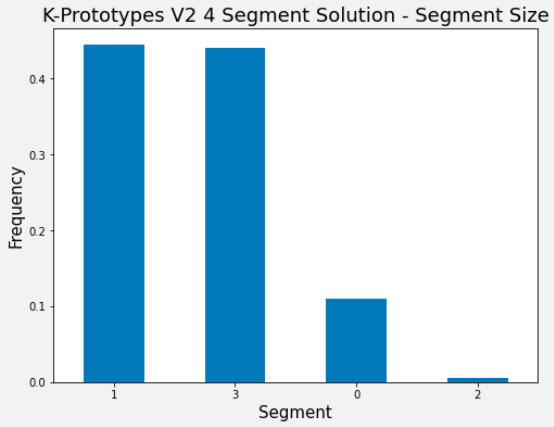


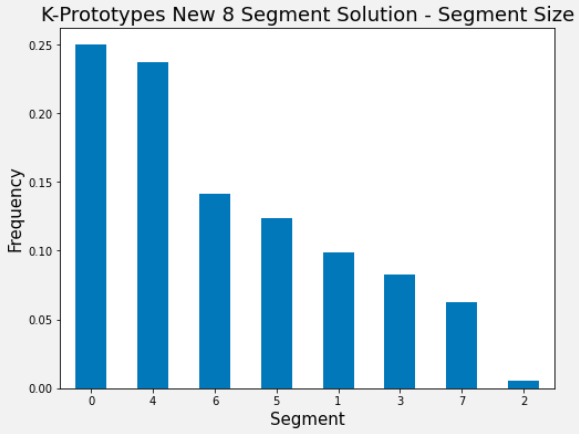


- A series of solutions were run, ranging from 2 to 14 clusters – resulting diagnostic plot is shown to the left
- An elbow in the Cost line can be observed at 6 clusters, which also has a corresponding relatively high Silhouette score (though all of these are fairly low in an absolute sense – another sign of potentially noisy data)
- There is also an interesting spike in the Silhouette score for the 8-cluster solution, so that was also profiled
- The cluster sizes for the 6-cluster solution were not particularly desirable, as they are dominated by two fairly large segments that together capture almost two-thirds of the Epoch 3 HBP events
- In contrast, the 8-segment solution evinces a much more balanced solution
- Curiously, both solutions group ALL HBP events ending in ejections into a tiny segment (since there are so few of them to begin with) - #3 at left in both solutions
- In addition, all cases with missing velocities reset to 0 were clustered together, which means they weren't effectively clustered at all - #2 at left in both solutions
- So, not great, but promising

K-prototypes Version 2 Results – Exclude 287 cases with no velocities

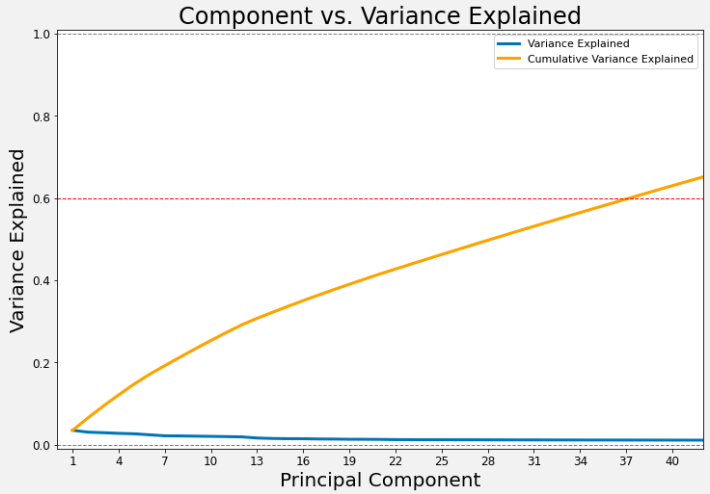


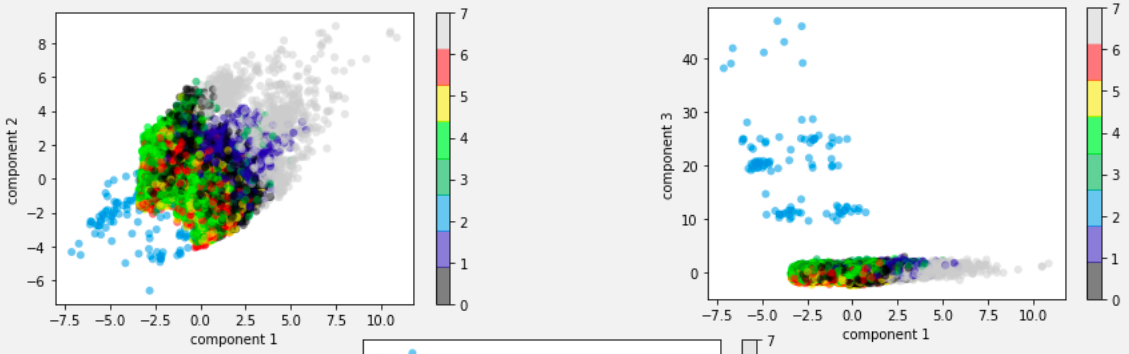


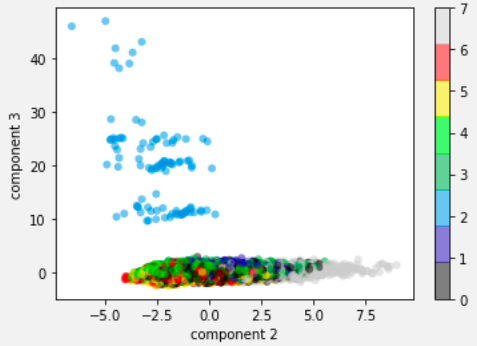


- Similar to the case with the Version 1 models, the observable elbows in the Cost plot occur at 4, 6, and 8 clusters
- In this case, though, the silhouette scores for the 8-cluster solution are higher than those for the 6-cluster solution
- The 4-cluster solution is too strongly dominated by two clusters, comprising 88% of the data collectively
- Once again, the 8-cluster solution has a relatively nice balance of cluster sizes, with no cluster > 25% of the data

PCA was run on both the full and dropped-missing Epoch 3 data





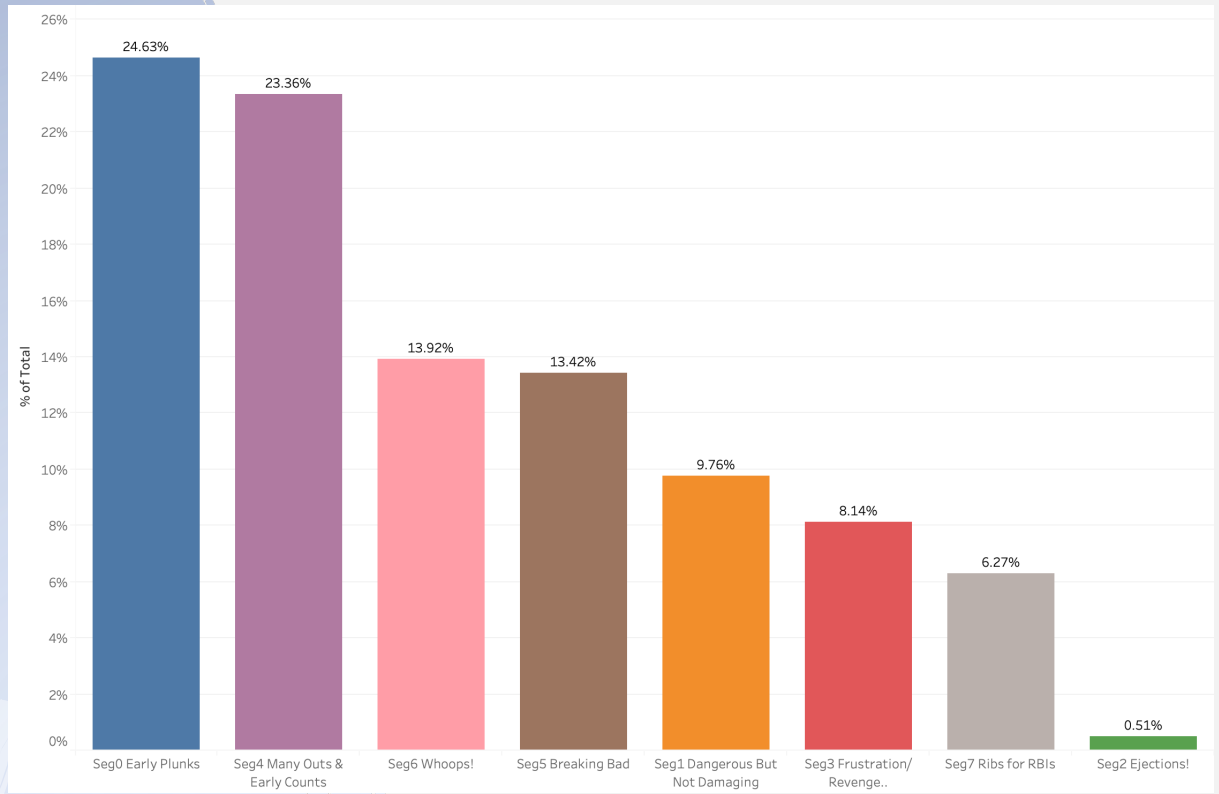


- Results were similar in both cases, so only the results from the analysis excluding the missing Epoch 3 data is shown here
- With 114 dimensions in the data, the hope was that Principal Components Analysis could be used to reduce the dimensionality for better plotting to visualize the cluster "clouds" on key components
- Unfortunately, each component in the PCA explained very little variance in the data, with 37 components needed to even achieve 60% of cumulative explained variance in the data
- Further evidence of this can be seen in the three PCA plots, using the V2 8-cluster solution, plotted PC1 vs PC2, PC1 vs PC3, and PC2 vs PC3
- The PC1 vs PC2 plot shows that components 1 and 2 both induce the most differentiation between cluster 2 (blue) and cluster 7 (light grey), with most remaining clusters in a bit of a jumbled mess
- Running through all combinations of the 37 key principal components to try to distinguish between the clusters was therefore deemed not worth the effort
- Traditional cluster profiles will be examined to see if any meaningful story exists for this solution

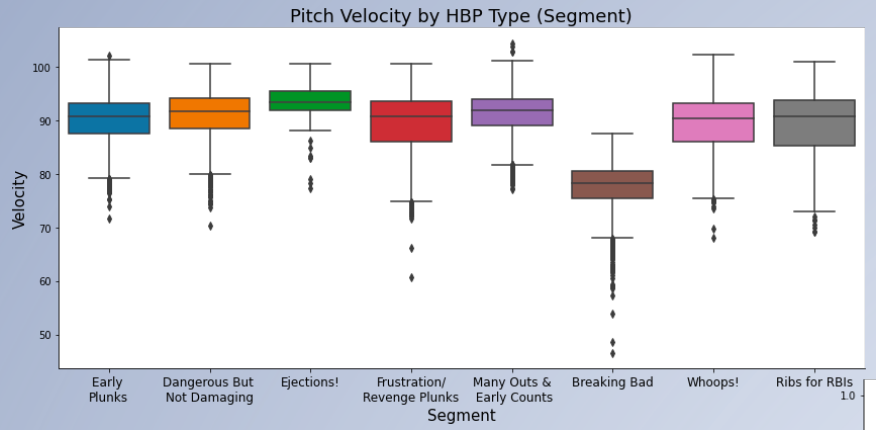
K-prototypes V2 8-cluster solution: High-level overview

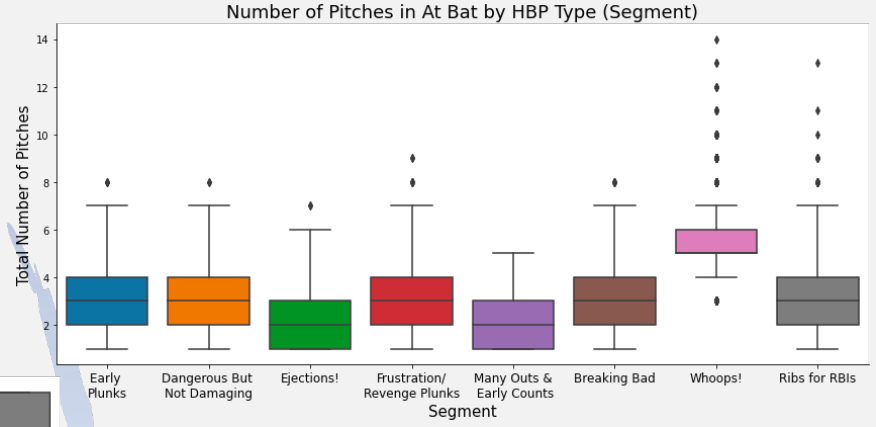
Cluster 0	Cluster 1	Cluster 2	Cluster 3
WILD THINGS/ EARLY PLUNKS High base out runs added, high win prob added, average velocity, no baserunners, pit team home, early innings top of lineup, few outs, no risp, low game stat, not a curve	DANGEROUS BUT NOT DAMAGING Higher leverage, high risp, late and close, more outs than average, pitching team ahead, pitching team wins, higher velocity, higher win prob added, low base out runs added, pit. team is away, low number of pitches	EJECTIONS! High game stat (more hbps in game), no runners on , fastball, no strikes in the count, high velocity, pitching team down by many runs, late innings, low leverage, low win prob added, low number of pitches	FRUSTRATION / REVENGE PLUNK High game stat, pitching team behind, down by many runs, late innings, pitcher/batter more likely to be removed, low leverage, low win prob added

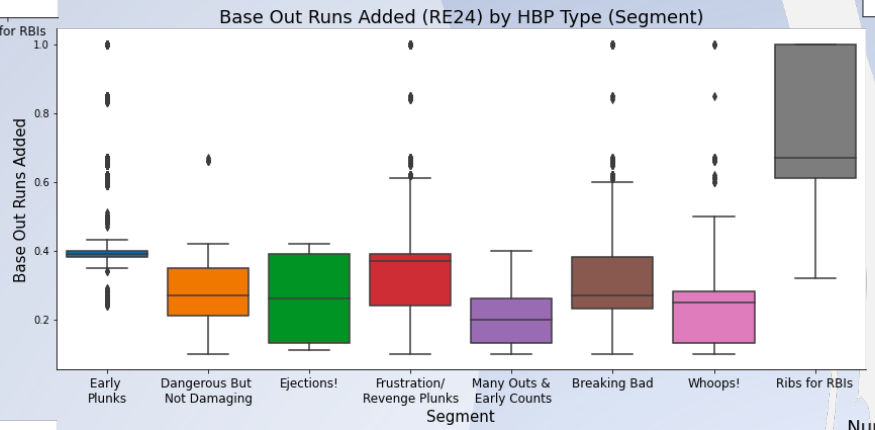
Cluster 4	Cluster 5	Cluster 6	Cluster 7
MANY OUTS & EARLY COUNTS High number of outs, high velocity, pitching team home, fastball or sinker, early count, early innings, low base out runs added, low win prob added, low leverage, low number of pitches, not a curveball, low game stat	BREAKING BAD Low velocity, Curve, Knuckler, Slider, no runners on, lefty pitcher, intraleague game, low leverage, low game stat, low win prob added, low number of pitches	WHOOPS! High pitch count, two strikes, higher than average outs, division game, no runners on, low base out runs added, low win prob added, low leverage, low game stat, pitching team is away	RIBS FOR RBIs Highest win prob added, highest base out runs added, highest leverage, bases full, rbi on play, runners in scoring position, pitching team ahead, late and close, higher than average game stat, few outs

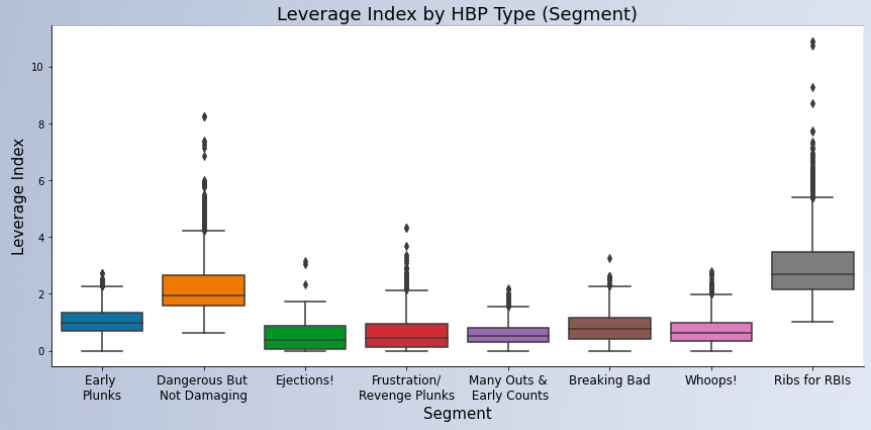


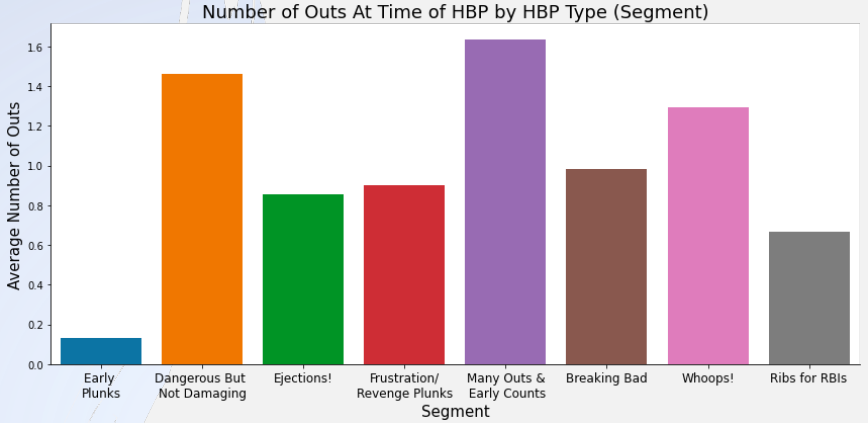
Despite the struggles of some models, clear differences do emerge



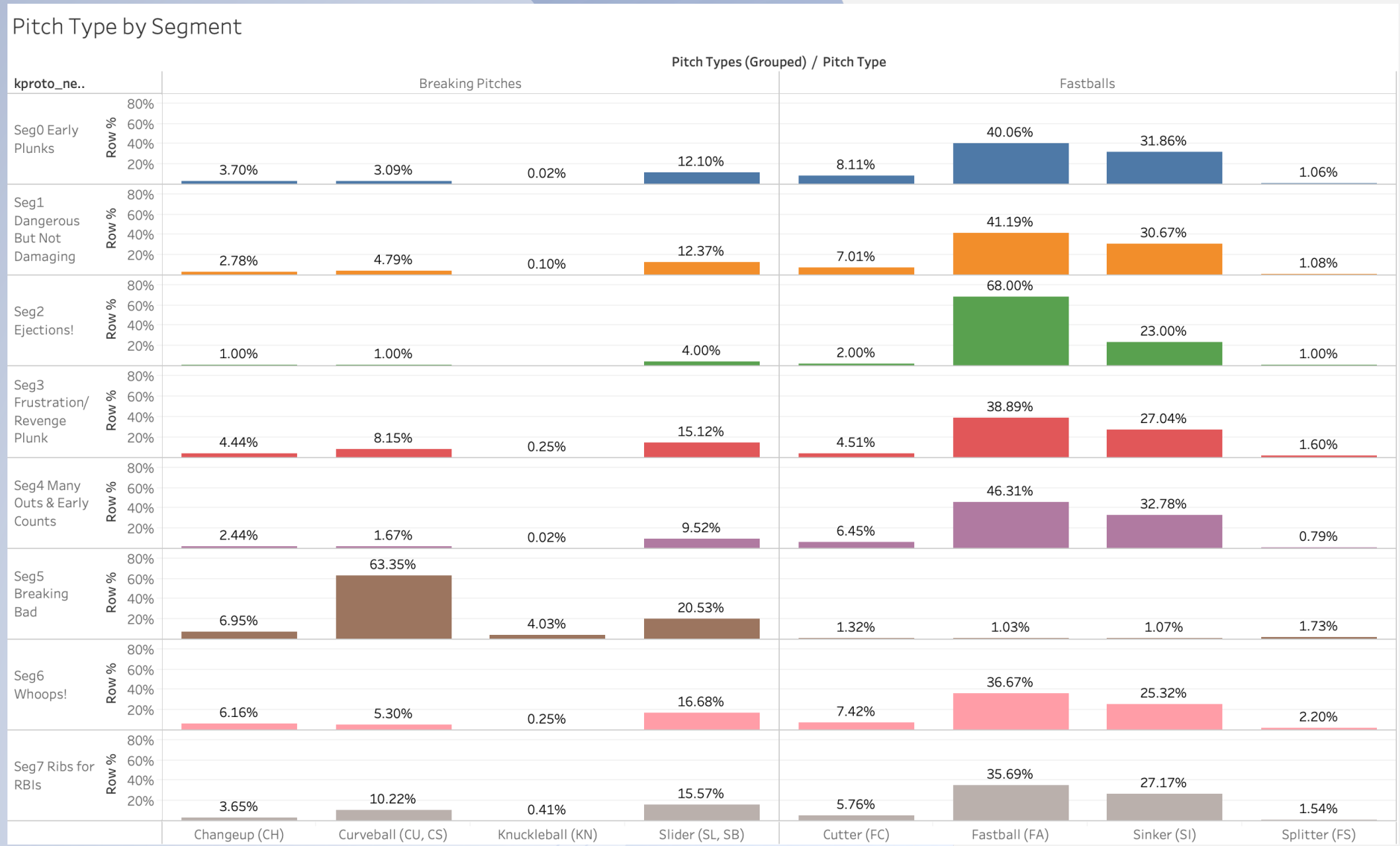








There were almost no fastballs of any type for the Breaking Bad cluster

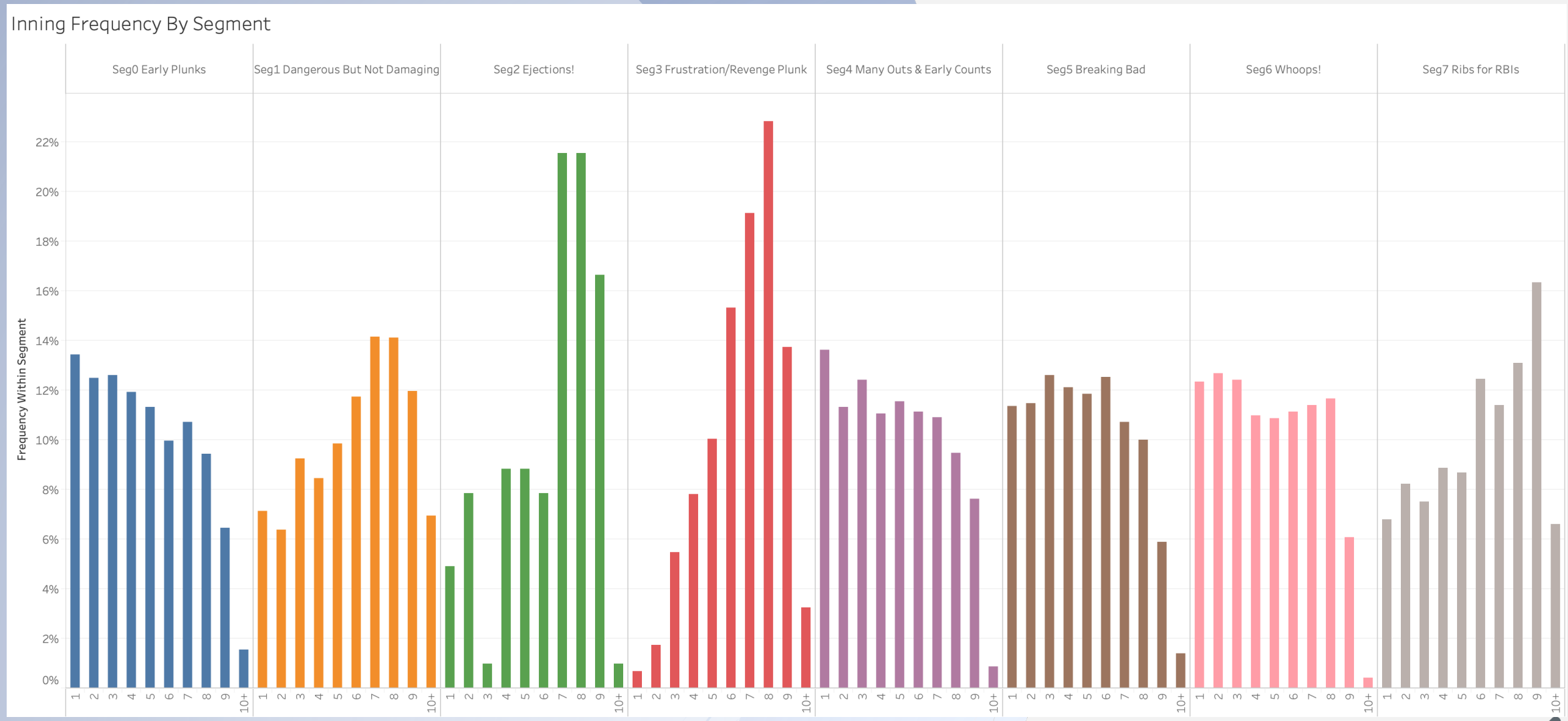


Ejection-worthy HBPs tend to occur early in counts (0-0, 0-1, 1-0), whereas Whoops! HBPs are almost exclusively 2-strike pitches

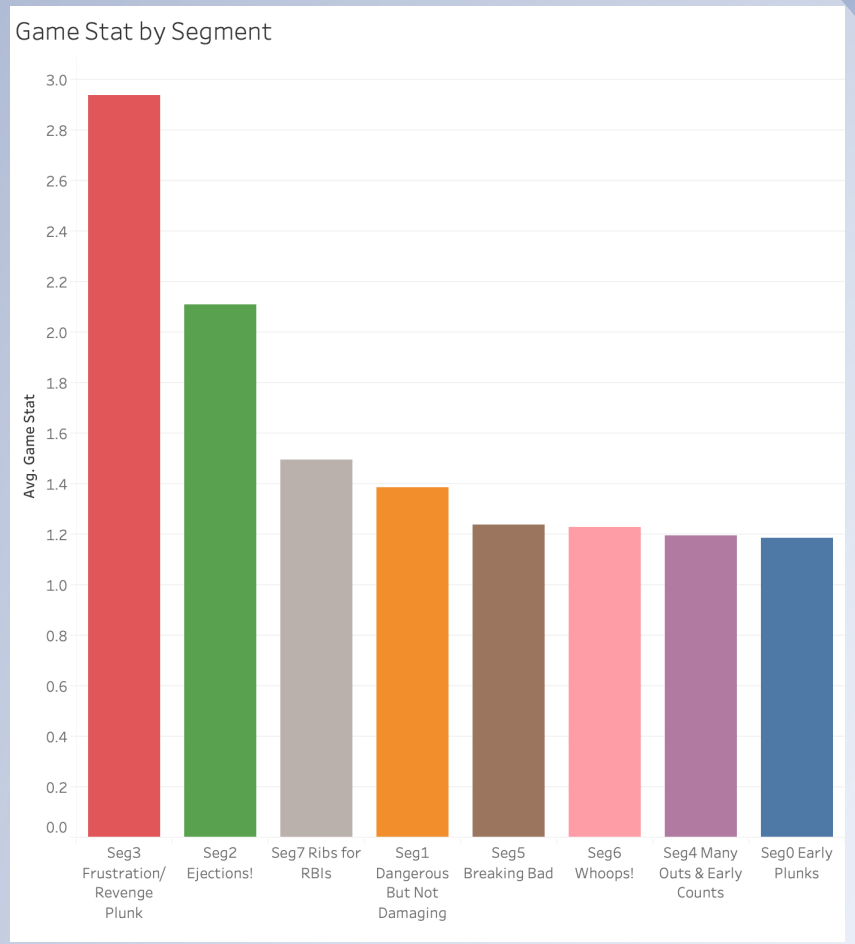
The Count by Segment

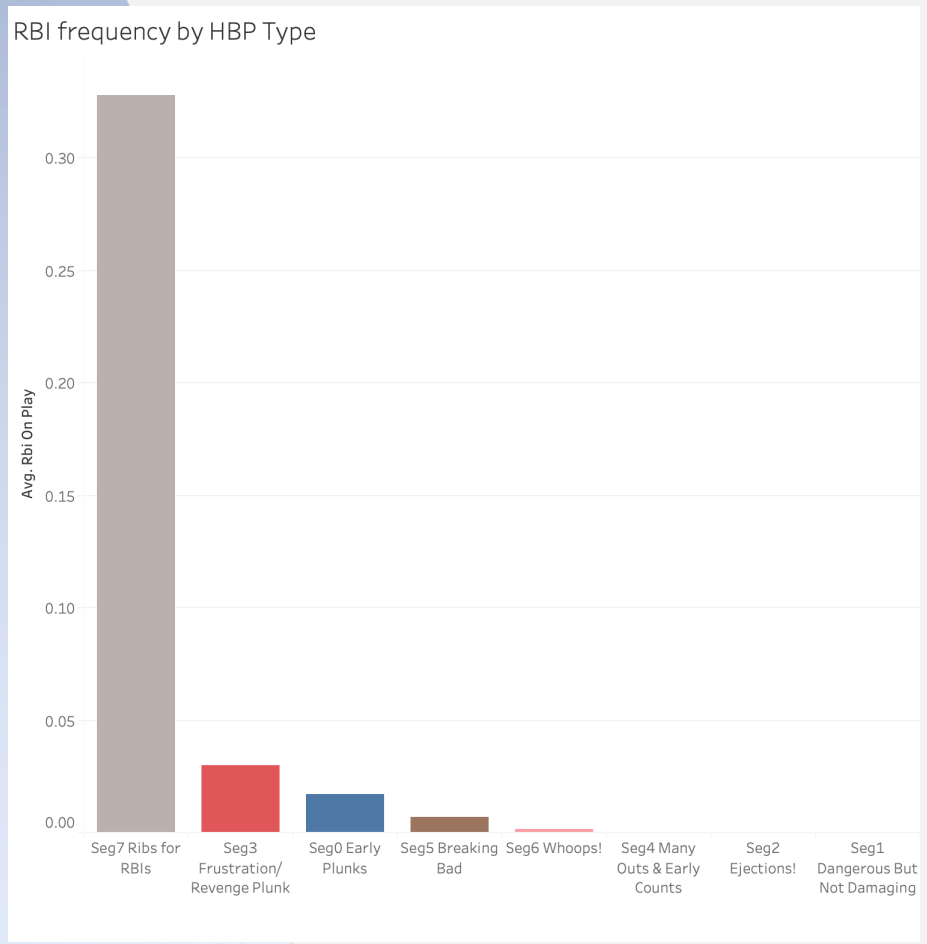
The Count	Seg0 Early Plunks	Seg1 Dangerous But Not Damaging	Seg2 Ejections!	Seg3 Frustration/ Revenge Plunk	Seg4 Many Outs & Early Counts	Seg5 Breaking Bad	Seg6 Whoops!	Seg7 Ribs for RBIs
0-0	18.93%	23.31%	39.22%	20.31%	26.54%	23.92%		23.06%
0-1	17.63%	17.61%	11.76%	17.17%	23.35%	16.12%		14.21%
0-2	12.32%	11.19%	1.96%	10.71%	11.94%	11.72%	3.99%	11.25%
1-0	6.16%	7.44%	13.73%	7.51%	8.06%	5.56%		7.02%
1-1	12.04%	10.93%	6.86%	11.75%	14.34%	11.46%	1.04%	8.38%
1-2	16.88%	13.09%	8.82%	13.48%	9.33%	15.67%	26.52%	15.80%
2-0	1.50%	1.54%	4.90%	2.71%	2.25%	0.97%	0.25%	1.44%
2-1	3.48%	4.67%	0.98%	4.18%	3.17%	3.40%	3.85%	2.39%
2-2	8.68%	7.24%	6.86%	8.00%	0.51%	8.28%	41.02%	12.05%
3-0	0.22%	0.31%	3.92%	0.49%	0.36%	0.15%	0.36%	0.32%
3-1	0.81%	1.18%		1.23%	0.13%	0.78%	2.99%	0.64%
3-2	1.34%	1.49%	0.98%	2.46%		1.98%	19.97%	3.43%

Certain HBP types occur more frequently late than others



Frustration Plunks and Ejections! tend to occur in games with multiple HBPs; Ribs for RBIs also shows why it gets its name



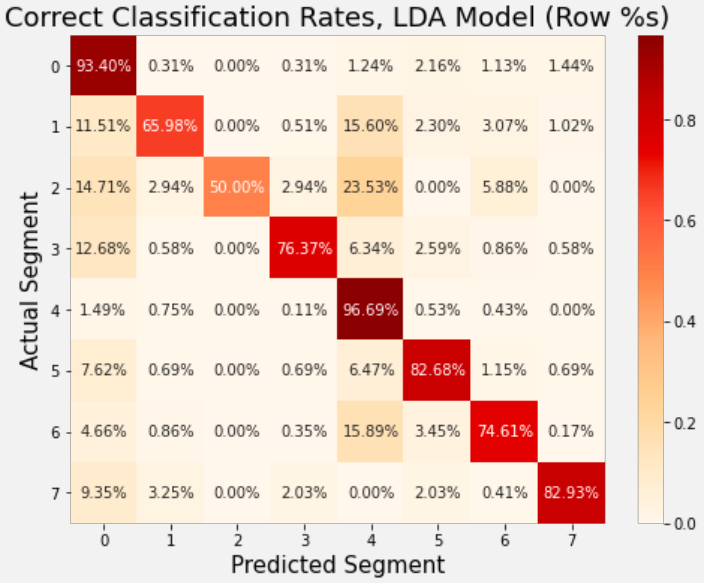


So we have a typology, now can we build a supervised classifier to predict new or historical events into one of the types uncovered?

Supervised Learning Attempt # 1: Linear Discriminant Analysis

- Linear Discriminant Analysis is another dimension reduction technique that is also useful for building classification algorithms; unlike PCA, it is a supervised approach as it requires a categorical dependent variable
- LDA tends to be popular in marketing research circles because it's a relatively straightforward algorithm that is easy for data collection vendors to implement for on-the-fly segment typing
- LDA searches for a line (or really a set of lines) that will maximally differentiate between labels/classes while minimizing variance within each class at the same time, then projects the data onto this new axis (or new axes)
- Similar to PCA, the first discriminant function explains the most variance in the data; there is one fewer discriminant function than the number of classes in the data
- In this case, we are attempting to predict the labels from the V2 K-prototypes 8-cluster solution that we like best; leave-one-out rules were used for categorical predictors

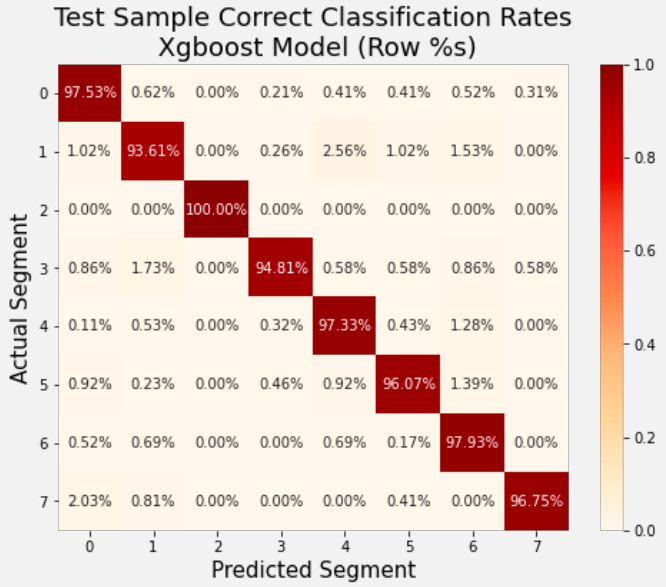
- On the whole, the model performs relatively well:
  - Training data accuracy: 85.39%
  - Test data accuracy: 84.99%
  - 10-fold cross-validation accuracy: 84.80%
- Not much overfitting to speak of, and with 8 segments to predict, the accuracy is fairly high (6.8x random)
- Unfortunately, the model only correctly classifies cluster 2 (EJECTIONS!) 50% of the time, when this should be the easiest segment to predict



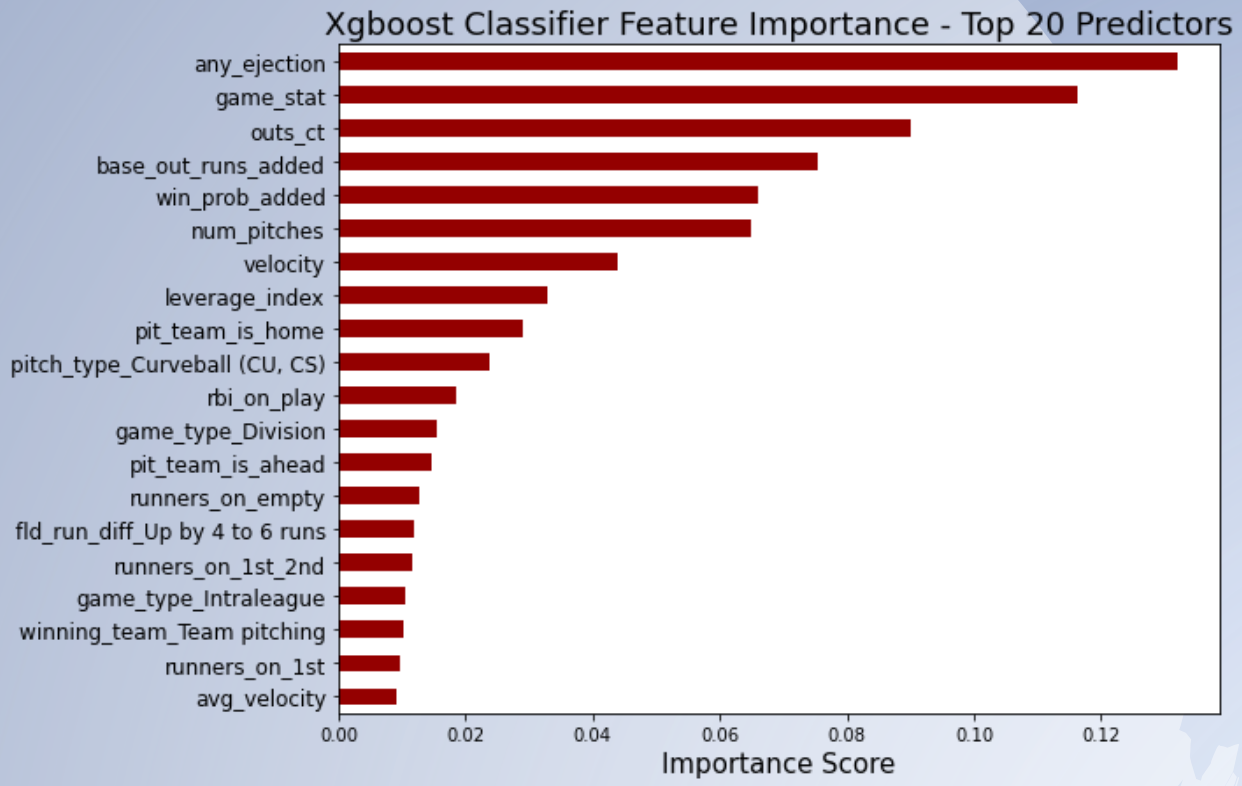
Supervised Learning Attempt # 2: Xgboost Classifier

- Xgboost is a family of gradient boosting machines, offering a supercharged version that outperforms standard gbms implemented in scikit-learn
- In addition to improved speed and increased accuracy, Xgboost can also handle missing data, which cannot be said for its scikit-learn counterpart
- It's no wonder Xgboost has been used in many winning Kaggle competition entries
- There are many hyperparameters that can be tuned using cross-validation
- Like any good gbm, the model benefits from a relatively slow learning rate, and a relatively large number of estimators
- My final model used 500 estimators, a max depth of 3, a learning rate of 0.1, and a subsample rate of 0.75

- Unsurprisingly (to me anyway), the resulting model's performance was outstanding:
  - Training data accuracy: 100%
  - Test data accuracy: 96.72%
  - 10-fold cross-validation accuracy: 96.52%
- While there is some overfitting which I could spend more time trying to reduce, this degree of accuracy is quite acceptable for our purposes
- Unlike LDA, all clusters are accurately predicted



Xgboost also provides feature importance scores



- There are several scoring options available; the scores to the left are calculated using the “gain” approach – essentially the relative gain in accuracy achieved by the presence of the feature
- The top predictor in the model is whether there is any ejection as a result of the HBP, which makes sense since cluster 2 is deterministically defined this way
- Other top predictors include how many total hit-by-pitches occurred at this point of the game, the number of outs at the time of the HBP, the RE24 and WPA statistics, the number of pitches thrown, the velocity and the leverage index – mostly continuous measures
- However, the second set of 10 predictors includes a number of categorical features such as which team is home, whether the pitch was a curveball, the game being a Division or Intraleague game, and so on

Finally, the Xgboost model was used to predict the missing 287 cases

- Since Xgboost can include missing data, the 287 cases from Epoch 3 that were missing velocity data could be classified using this algorithm, with Numpy NaN values in place
- FORTUNATELY, the Xgboost classifier works on this data…
- … but UNFORTUNATELY, the model appears to still be a bit puzzled to find good surrogates for missing velocities to produce truly nuanced segment assignments
- 86.7% of these cases get classified into Cluster 5 (BREAKING BAD), which is the HBP type that tends to have the lowest velocity of pitches thrown, with lots of breaking pitches
- The remaining clusters assigned include:
  - Cluster 7 (RIBS FOR RBIs) – 7.0%
  - Cluster 1 (DANGEROUS BUT NOT DAMAGING) – 2.8%
  - Cluster 3 (FRUSTRATION/REVENGE PLUNK) – 1.7%
  - Cluster 6 (WHOOPS!) – 1.0%
  - Cluster 2 (EJECTIONS!) – 0.7%
  - None of these cases were classified into Clusters 0 (EARLY PLUNKS) or 4 (MANY OUTS & EARLY COUNTS)
- So, the jury is still out for whether this solution will work sufficiently for typing earlier, non-PITCHf/x epochs of the HBP data; it should work great moving forward for any 2020 or later HBP events, though

NEXT STEPS

BESIDES GETTING SOME SLEEP…

A lot of work is left to be done, including:

- "Typing" the data from Epochs 1 & 2
  - This can be accomplished in potentially two different ways:
    1. Use the existing Xgboost classification algorithm to classify Epochs 1 & 2 using the typology built from Epoch 3 data
       a. As explained earlier, Xgboost can handle missing data, but the resulting segment sizes may be distorted, especially for Epoch 1 which is missing the most data
    2. Continue using a pattern sub-model method, where a unique typology is created for each Epoch individually, using unsupervised learning
       a. Resulting segments/types can be compared across eras to see whether any overlaps exist
- Another approach could include continuing to work with the Epoch 3 data, but refining the k-prototypes algorithm to skew towards relying more on the categorical/binary data for uncovering HBP types so that historical classifications might work a little better
  - The 287 cases with missing data from Epoch 3 might actually fit better with Epoch 2 data, which shares the same pattern of missingness, so that could be another consideration
- Continue to refine Xgboost model to reduce remaining overfitting and/or produce a more compact model
- Build a Flask app to score new (or historical) data into one of the types from this report or from further analysis above
- Run a Network analysis to try to understand the implied social network of hit-by-pitch events – how many steps does it take to get from Chris Sale to Nolan Ryan via hit batters (like six degrees of Kevin Bacon)
- Lots and lots of additional visualizations could also be created – there's so much data to play with here!
- Write blog posts to share progress/findings

Appendix

Data & Modeling Details

Baseball Prospectus Open Source HBP Database

Description of data

BP's collection of HBP data includes 62,129 hit-by-pitch events since the start of the 1969 season

The database now includes 30 columns of information after Rob Mains and his team added some additional data at my suggestion (items in red in table)

1,863,870 data points are in the full file

The first version of the database also only had 62,126 rows, as three cases from 1973 were inadvertently missed; I filled these in with information from Stathead's data in concert with cross-referencing on baseball-reference.com

FIELD NAME	DESCRIPTION	VALUE DESCRIPTION/RANGE	N Missing
DATE	Date when HBP event occurred	April 7, 1969 – September 29, 2019	0
WIN_TEAM	Team that eventually wins the game	Any of the 30 MLB franchises, 3-letter code used	0
PIT_ID	Baseball Prospectus Pitcher ID	Numeric code	0
PITCHER	Pitcher's name	4,301 unique pitchers	0
PIT_TEAM	Pitcher's team	Any of 30 MLB franchises, 3-letter code used	0
THROWS	Which arm pitcher throws with (for the at bat)	Left or Right	0
BAT_ID	Baseball Prospectus Batter ID	Numeric code	0
BATTER	Batter's name	3,969 unique batters	0
BAT_TEAM	Batter's team	Any of 30 MLB franchises, 3-letter code used	0
BATS	Stance of batter (for the at bat)	Left or Right	0
POSITION	Batter's field position played at time of at bat	Numeric position code 1 to 11 (10=DH, 11=PH,PR)	0
LINEUP_SPOT	The batting order position of the batter	1 thru 9	0
BAT_TEAM_SCORE	Number of runs already scored by batting team at time of HBP	0 to 25 runs	0
FLD_TEAM_SCORE	Number of runs already scored by pitching team at time of HBP	0 to 21 runs	0
BASES_STAT	Number of baserunners on base (and on which base) at time of at-bat	12 values, from ___ to 123 (empty to full)	0
HALF	Half inning at time of HBP	Top or Bottom	0
INNING	Inning at time of HBP	1 to 22	0
OUTS_CT	Number of outs at time of HBP	0, 1, or 2	0
BALLS	Number of called balls at time of HBP – only available from 1988 on	0, 1, 2, or 3	14,104
STRIKES	Number of called strikes at time of HBP – only available from 1988 on	0, 1, or 2	14,104
PITCH_TYPE	Type of pitch thrown that hit batter – only available from 2008 on	10 pitch types	42,447
VELOCITY	Velocity of pitch that hit batter – only available from 2008 on	46.6 mph to 104.4 mph	42,447
PIT_REMOVED	Whether the pitcher was removed from the game following HBP	Y or N	0
BAT_REMOVED	Whether the batter was removed from the game following HBP	Y or N	0
PIT_EJECTED	Whether the pitcher was ejected from the game following HBP	Y or N	0
BAT_EJECTED	Whether the batter was ejected from the game following HBP	Y or N	0
UMPIRE	Name of umpire during at bat when HBP occurred	355 unique umpires	1,124
EJECTIONS_CT	Total number of ejections in game as a result of HBP (may include other players than just batter or pitcher)	0 to 9	0
EJECTIONS_PL	Player names/IDs for all players ejected due to the HBP	Numeric codes	61,713
EJECTION_DES	Text description of what led to ejections	Text	61,713

Stathead.com Baseball Event Finder

Description of data

Stathead.com uses data based on play-by-play accounts accumulated by Retrosheet.com

You can set search criteria for a large number of different baseball events, like HRs, Triples, Sacrifice Bunts, or Hit By Pitches

I included the years 1969 to 2019 in my scrape to match BP's dataset

17 columns of data were included in the scrape

Fields not included in the BP database are in **bold**

FIELD NAME	DESCRIPTION	VALUE DESCRIPTION/RANGE	N Missing
EVENT_ID	Event # for the range of data selected (Orig. Yr#)	1 to 62,129	0
GAME_STAT	Number of HBP events occurring in game previous to and including this at bat (Orig. Game#)	1 to 7	0
DATE	Date of HBP event	April 7, 1969 – September 29, 2019	0
BAT_ID	Baseball Reference ID# for batter	Alphanumeric code	0
BATTER	Batter's name	3,969 unique batters	0
BAT_TEAM	Batter's team	Any of 30 MLB franchises, 3-letter code used	0
PIT_ID	Baseball Reference ID# for pitcher	Alphanumeric code	0
PITCHER	Pitcher's name	4,301 unique pitchers	0
PIT_TEAM	Pitcher's team	Any of 30 MLB franchises, 3-letter code used	0
HALF_INNING	Inning (including half) when HBP occurred	t1 to b22 (t=Top, b=Bottom)	0
ON_BASE	Number of baserunners on base (and on which base) at time of at bat	12 values, from ___ to 123 (empty to full)	0
OUTS	Number of outs at time of HBP	0, 1, or 2	0
PIT_COUNT	Number of pitches thrown by pitcher in this at bat	Includes both total number of pitches plus the count itself (3-2)	14,429
RBI_ON_PLAY	Whether a run was driven in by batter due to the HBP	0 (No) or 1 (Yes)	0
WIN_PROB_ADDED	Given average teams, the change in win probability caused by this batter during the game	-1 to 1	0
BASE_OUT_RUNS_ADDED	Given the bases occupied/number of outs situation, how many runs did the batter add in the resulting play. Normalized to per 24 out basis (Orig. RE24)	0 is average, above 0 is better than average	0
LEVERAGE_INDEX	The pressure the pitcher or batter saw this at bat.	1 is average, below 1 is low, above 1 is high	0

I also created a database of my own containing franchise information

https://howtheyplay.com/team-sports/major-league-baseball-expansion-and-franchise-relocation

Based on information gleaned from howtheyplay.com, I compiled a database containing information about the team/franchise name, league, and division for each team for each year since 1969.

Several franchises have relocated and/or changed names over the years, and the division alignments have changed a lot as well, going from a two-divisions-per-league structure (East & West) to the current three (East, Central & West). Two teams have switched leagues during that time as well.

DATA CLEANING & FEATURE ENGINEERING DETAILS

Data Cleaning: Prior to merge

BP Data

- Create/combine half and inning variables to match half_inning from Stathead.com (SH hereafter)
- Combine the following teams, which relocated or changed names at one point to become current team:
  - Seattle Pilots (SE1) → Milwaukee Brewers (MIL) in 1970
  - Washington Senators (WS2) → Texas Rangers (TEX) in 1972
  - California Angels (CAL) → Anaheim Angels (ANA) in 1997
  - Montreal Expos (MON) → Washington Nationals (WAS) in 2004
  - Florida Marlins (FLO) → Miami Marlins (MIA) in 2011
- Pitcher and Batter names were not consistently recorded across each database, so prior to merging these needed to be made consistent. For example:
  - Vangorder → Van Gorder
  - Decinces → DeCinces
  - Mccarty → McCarty
- Correct runners on base inconsistencies (two instances where SH data was correct and BP data was not)

SH Data

- Make format of DATE variable consistent with BP format
  - Created doubleheader variable prior to doing so (SH uses (2) after the date to indicate the second game of the day)
- Combine franchises and adjust abbreviations to match
  - KCR → KCA CHC → CHN STL → SLN NYY → NYA
  - NYM → NYN SFG → SFN SDP → SDN LAD → LAN
  - SEP → MIL WSA → TEX CAL → ANA MON → WAS
  - WSN → WAS FLA → MIA TBD → TBA TBR → TBA
  - LAA → ANA CHW → CHA
- Adjust last names of players, including removal of suffixes
- Fix game discrepancies
  - 11 HBP events where the game was played in one team's ballpark, but the other team was designated "home" team because of rescheduling, stadium conflicts, etc. BP data was correct, SH data was not. Confirmed on b-r.com.
- # of outs and on-base status matched after 3 cases were added back into BP database

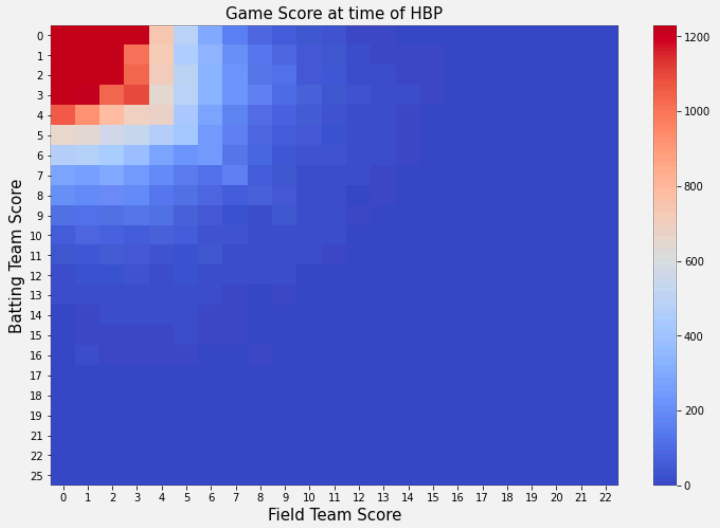
Data Cleaning: Post merge

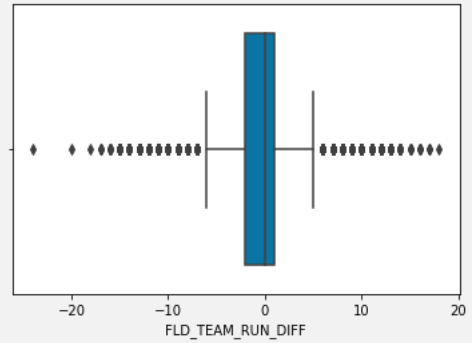
- Removed duplicated columns
- Checked missing data
  - 1124 umpire names are missing for some reason; umpire will not be used for any modeling, so these remain
  - Balls & Strikes were not recorded in the box scores until the 1988 season, so these are missing for any 1969-1987 games
  - Velocity & Pitch Type were not recorded until PITCHf/x came into being in 2008, so these are missing for any 1969-2007 data
    - There are also 287 instances of missing velocity data post-2008 for some reason
  - Because these missing data instances are systematic and due solely to scoresheet practices in a given era, it will be impossible to analyze all 62,129 cases together; data cannot be imputed without gross error, so analysis will need to be done by era
  - Instead, I created an Epoch variable to separate each of the three scoresheet epochs
    - 1 = Old School Scoresheets (no balls & strikes, no velocities or pitch types) – 1969-1987 – **14,104 HBP events**
    - 2 = Project Scoresheet Era (balls & strikes begin to be recorded, but not velocities or pitch types) – 1988-2007 – **28,056 HBP events**
    - 3 = PITCHf/x Era (all balls & strikes are recorded, along with velocities and pitch types) – 2008-2019 – **19,969 HBP events**
- Fixed the Pat Verditte Problem
  - Pat Verditte is the only pitcher in the database to be designated as a “switch pitcher”, meaning he can pitch with either his left or right arm
  - For the batters in the database, any switch hitters were not called out as such, they were recorded based on the side of the plate they were batting from (i.e., right-handed or left-handed batting stance)
  - Research showed that Verditte almost always pitched with the same handedness as the batter was batting, so the “S” designation in the database was adjusted to R or L for each of Verditte’s HBP events

A number of features were created or modified from the original data

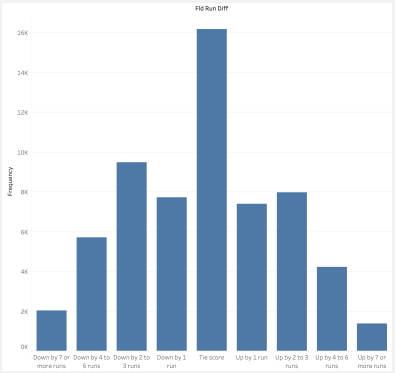
- Combined low-frequency pitch types
  - Slow Curves (CS) were designated as Curves (CU)
  - Screwballs (SB) were designated as Sliders (SL)
- Extra-inning games are infrequent, so any inning beyond the 10th was designated with a 10
- Created a “PIT_BAT” variable to combine who the pitcher and batters were for each event for easier EDA
- Recoded winning team variable so it was relative to the pitcher (and not based on team name alone)
  - In earlier eras, tie games actually happened, so also created a Tie Game indicator
- Indicator showing whether pitching team was home team was added
- SH data originally had the number of pitches and the count in a single field [“11 (3-2)”], so the number of pitches were split into its own variable

- Computed a run differential variable relative to the fielding team
- Most games don’t have large run differentials, as you can see here



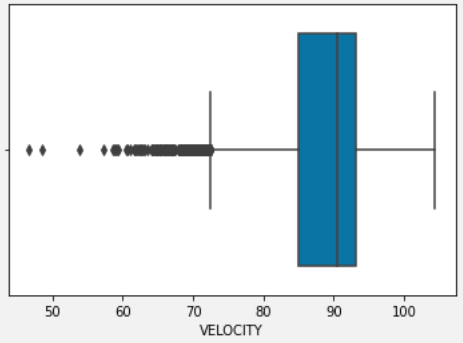


- So, I also created a categorized version which runs from “Down by 7+ runs” to “Up by 7+ runs”

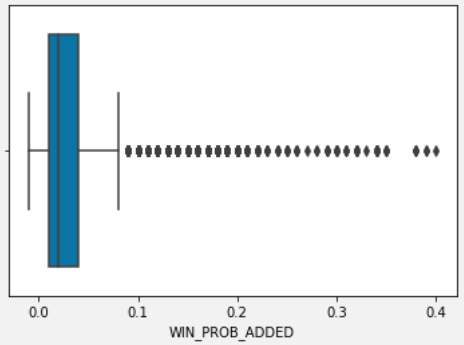


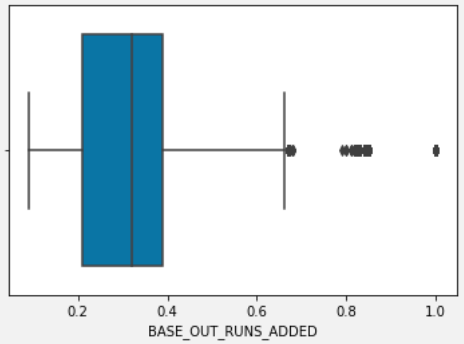
Feature engineering, continued

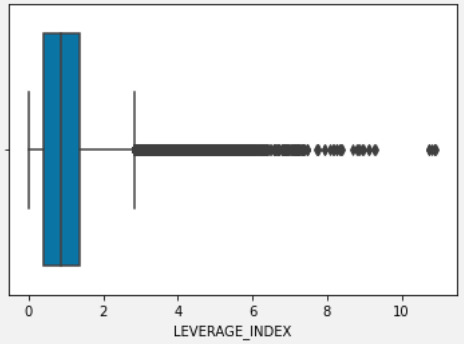
- Using the information gleaned from howtheyplay.com, new variables were created based on league and division information
  - PIT_LEAGUE / BAT_LEAGUE
  - PIT_DIVISION / BAT_DIVISION
  - PIT_LEAGUE_DIVISION / BAT_LEAGUE_DIVISION
- Based on these new variables, I created a new GAME_TYPE variable to indicate the nature of each game:
  - “Division” – games between teams in the same division
  - “Intraleague” – games between teams in the same league, but not the same division
  - “Interleague” – games between teams in different leagues, regardless of division
- Ejection & removal variables were converted from Y/N to 1/0 variables
  - An “any ejection” indicator was created, equaling 1 if the ejections count was > 0, otherwise equaling 0
- Batter (field) positions were slightly adjusted
  - PH (=11) and PR (=0) fields were combined into PH_PR
  - Values were simultaneously converted into their string equivalents for easier interpretation (e.g., 1=P, 2=C, …, 10=DH, 11=PH_PR)
- Created a POSITION_GROUP variable which grouped batter (field) positions into four categories:
  - Infielders
  - Outfielders
  - Pitchers & Catchers
  - Designated Hitters, Pinch Hitters, and Pinch Runners
- Grouped velocity two ways:
  - Low/Avg/High binaries
  - Categorical ranges:
    - Under 70 mph
    - 70-79 mph
    - 80-89 mph
    - 90-95 mph
    - 96+ mph



Features not further engineered







- Because Win Probability Added (WPA), Base Out Runs Added (RE24), and Leverage Index (LI) are relatively well-known in the sabermetrics community, I decided not to further recode or bucket these variables
- WPA and LI have tight distributions with narrow ranges for the most part, though a large number of statistical outlier cases are evident in the box-and-whisker plots to the left

MODELING DETAILS

List of potential clustering variables - continuous in red

1. month_3
2. month_4
3. month_5
4. month_6
5. month_7
6. month_8
7. month_9
8. month_10
9. doubleheader
10. pit_team_is_home
11. game_type_Division
12. game_type_Interleague
13. game_type_Intraleague
14. game_stat
15. throws_L
16. bats_L
17. position_1B
18. position_2B
19. position_3B
20. position_C
21. position_CF
22. position_DH
23. position_LF
24. position_P
25. position_PH_PR
26. position_RF
27. position_SS
28. position_group_DH_PH_PR
29. position_group_Infielder
30. position_group_Outfielder
31. position_group_Pitcher_Catcher
32. lineup_spot_1
33. lineup_spot_2
34. lineup_spot_3
35. lineup_spot_4
36. lineup_spot_5
37. lineup_spot_6
38. lineup_spot_7
39. lineup_spot_8
40. lineup_spot_9
41. fld_run_diff_Down by 1 run'
42. fld_run_diff_Down by 2 to 3 runs
43. fld_run_diff_Down by 4 to 6 runs
44. fld_run_diff_Down by 7 or more runs
45. fld_run_diff_Tie score
46. fld_run_diff_Up by 1 run
47. fld_run_diff_Up by 2 to 3 runs
48. fld_run_diff_Up by 4 to 6 runs
49. fld_run_diff_Up by 7 or more runs
50. pit_team_is_ahead
51. pit_team_is_behind
52. runners_on_1st
53. runners_on_1st_2nd
54. runners_on_1st_3rd
55. runners_on_2nd
56. runners_on_2nd_3rd
57. runners_on_3rd
58. runners_on_empty
59. runners_on_full
60. risp
61. half_Top
62. inning_rec_1
63. inning_rec_2
64. inning_rec_3
65. inning_rec_4
66. inning_rec_5
67. inning_rec_6
68. inning_rec_7
69. inning_rec_8
70. inning_rec_9
71. inning_rec_10
72. outs_ct
73. num_pitches
74. the_count_0-0
75. the_count_0-1
76. the_count_0-2
77. the_count_1-0
78. the_count_1-1
79. the_count_1-2
80. the_count_2-0
81. the_count_2-1
82. the_count_2-2
83. the_count_3-0
84. the_count_3-1
85. the_count_3-2
86. pitch_type_Changeup (CH)
87. pitch_type_Curveball (CU, CS)
88. pitch_type_Cutter (FC)
89. pitch_type_Fastball (FA)
90. pitch_type_Knuckleball (KN)
91. pitch_type_Sinker (SI)
92. pitch_type_Slider (SL, SB)
93. pitch_type_Splitter (FS)
94. Velocity
95. low_velocity
96. avg_velocity
97. high_velocity
98. velocity_range_70-79 mph
99. velocity_range_80-89 mph
100. velocity_range_90-95 mph
101. velocity_range_96+ mph
102. velocity_range_Under 70 mph
103. velocity_range_Unknown
104. rbi_on_play
105. win_prob_added
106. base_out_runs_added
107. leverage_index
108. winning_team_Team pitching
109. pit_removed
110. bat_removed
111. pit_ejected
112. bat_ejected
113. any_ejection
114. ejections_ct

Unsupervised Learning Attempt #1: HDBSCAN with HEOM Distances

- HDBSCAN was developed by a group of academics in order to produce better-behaving DBSCAN models – see YouTube for a great introductory video
- HDBSCAN is implemented to work with any distance metric in scikit-learn, but can also use a custom distance metric
- Since we have mixed data, standard distance metrics aren't an option
- Enter Heterogeneous Euclidean-Overlap Metric (HEOM), included in a Python package called distython, created by a graduate student named Kacper Kubara
  - HEOM distances are computed as follows:
    - If categorical – returns 0 if the attribute is of the same class, 1 otherwise
    - If numerical, computes the distance using the normalized Euclidean distance metric (so no additional scaling is required)
    - If missing, returns 1
    - Final result is the square root of the sum of every distance squared
- As with DBSCAN, HDBSCAN does not attempt to classify every case whether it is warranted or not
  - Any cases not surrounded by enough density in multidimensional space are considered noise by the model, and classified as -1
- HDBSCAN extends DBSCAN by converting it into a hierarchical clustering algorithm, then using a technique to extract a flat clustering based on the stability of the clusters - see https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html for more detail
- Upshot – this is a really interesting approach, but if you have noisy data, it may end up classifying most observations *as* noise
  - In that case, the model is probably doing its job well, but it makes it hard to draw inferences from the data as a result

Unsupervised Learning Attempt #2: K-Prototypes Version 1

- K-prototypes is a variation on K-means clustering, where instead of using pure numeric (often Euclidean) distances, a hybrid distance metric is used that can handle mixtures of continuous and categorical attributes
- The dissimilarity measure between an observation and the "prototype" (fancy name for centroid of a cluster) is based on a weighted combination of:
  - Euclidean distances between continuous attributes
  - The number of non-matching categorical attributes
- The weight (λ) can be tuned to give more power to either the numeric or categorical features, or treat them equally with a value of 0.5
- Data needs to be scaled before running the model
- Like K-means, K-prototypes will label each and every case into the cluster based on the nearest prototype it finds in the data (can be good or bad)
- Also like K-means, K-prototypes depends on the data scientist telling the learner how many prototypes to find, and in the real world (unlike with simulated data) there is no known truth here
- So, a common approach is to use different randomized starting points for a range of different numbers of clusters
- A pseudo grid search can be constructed to accomplish this task
- The resulting solutions can be compared based on the model cost (think reproducibility) and silhouette score metrics
  - A typical approach is to look for "elbows" in the cost plot – where an elbow exists is considered to be a reasonable solution, which hopefully corresponds with a higher silhouette score
- For Version 1 of K-prototypes, the 287 missing velocities in Epoch 3 were set to 0 so they could be included

THANK YOU