CSC 272 April 22nd 2024

"Rounding the Data-Bases: Harnessing Baseball Quality of Contact Statistics for Predictive Analytics"

by Evan Hucke, Mikaela Schultz and Ainsley Yoshizumi

Introduction

Baseball has long been at the forefront of statistical analytics in the sports world. Sabermetrics, or the empirical analysis of baseball through statistics, used to predict the performance of players, giving teams a winning edge. With the help of sabermetrics, teams can forecast results by making predictions based on previous data and gain insights into player performance and game strategy.

Our analysis focuses on data obtained from Baseball Savant, a comprehensive repository of MLB statistics. Baseball Savant gives us access to a plethora of free baseball statistics that we can use by examining various metrics, including quality of contact, pitches & locations, and basic statistics. We seek to uncover patterns and trends that can inform predictive models and enhance our understanding of player performance. In this project, we aim to leverage baseball quality of contact statistics to explore predictive analytics. Specifically, we want to use techniques learned in our data mining course this semester to predict statistics such as total numbers of singles, doubles, triples, and homeruns for a given player. In our research. We found that quality of contact statistics can predict the number of singles and home runs hit by a player in a season more accurately than the number of doubles and triples hit.. Being able to predict these statistics will provide teams with statistical information that teams can use to leverage their game strategy based on the types of hits and their players' success in each hit category.

Dataset Description

There are several tremendous resources for acquiring data on the MLB. These include Baseball Reference, Retrosheet, and Fan Graphs. Our dataset is sourced from Baseball Savant, providing a rich array of hitting statistics for MLB players. The dataset includes key attributes such as flare-burner percentage, topped percentage, under percentage, weak barrel percentage, sweet spot percentage, barrel batted rate, and hard-hit percentage. Each of these attributes offers valuable insights into contact quality and play outcomes.

Dataset link:

https://baseballsavant.mlb.com/leaderboard/custom?year=2023&type=batter&filter=&min=q&

selections=pa%2Ck_percent%2Cbb_percent%2Cwoba%2Cxwoba%2Csweet_spot_percent%2Cbarrel_batted_rate%2Chard_hit_percent%2Cavg_best_speed%2Cavg_hyper_speed%2Cwhiff_percent%2Cswing_percent&chart=false&x=pa&y=pa&r=no&chartType=beeswarm&sort=xwoba&sortDir=desc

Type of Statistics in Our Dataset

Our dataset from Baseball Savant originally encompassed three main categories of statistics: Basic stats, Quality of Contact, and Pitch Location. At first we were interested in many different attributes in order to find patterns and predictability. We wanted to have a dataset with a lot of different attributes as we started this project with a broad approach. As we narrowed our research, we lessened the amount of attributes used. We decided to only use types of hits and quality of contact. These metrics offer a comprehensive, yet specific view of player performance and hitting outcomes on the baseball field.

Table 1

Statistic	**Explanation**
Name (ï»¿)	First and last name of the player
player_id	MLB organizes data by a given number for each player
year	Year of the MLB season the instance is from
hit	A ball hit into fair territory that results with the batter reaching base safely without error
single	A hit that results in the batter reaching first base
double	A hit that results in the batter reaching second base
triple	A hit that results in the batter reaching triple base
home_run	A hit that results in the batter reaching base and scoring

Quality of Contact

Quality of contact metrics delve deeper into the outcomes of batted balls, reflecting the effectiveness of how a hitter makes contact with the ball. Metrics such as sweet spot percentage, barrel batted rate, and hard-hit percentage quantify the quality and effectiveness of contact generated by hitters against a pitcher's pitches.

Table 2

Statistic	**Explanation**	**Type of Attribute**
sweet_spot_percent	How often a player produces a ball in the launch angle sweet spot zone ranging from 8-32 degrees	Numeric
barrel_batted_rate	How often a player produces a barreled ball with a launch angle between 26-30 degrees and mph>98	Numeric
solidcontact_percent	Rate which a player produces a batted ball with a LA=23.6 degrees	Numeric
flareburner_percent	How often a batted ball is classified as a flare or burner per MLB statcast.	Numeric
poorlyunder_percent	Type of batted ball, LA=45.15 degrees	Numeric
poorlytopped_percent	Type of batted ball, LA<=0 degrees	Numeric
poorlyweak_percent	Type of batted ball <= 59 mph	Numeric
hard_hit_percent	Percentage of batted balls hit	Numeric

	with an exit velocity >95mph	
oz_contact_percent	Percentage of out balls hit out of strike zone	Numeric
iz_contact_percent	Percentage of balls hit inside the strike zone	Numeric

Data Preparation

Our data preparation was fairly simple because we were dealing with all numeric attributes, no missing values, and the summary of each attribute in Weka demonstrated that we did not have any egregious outliers. Additionally, Baseball Savant did most of the work in organizing our data in an accessible way that allowed us to download as a .Csv with very little editing. We only had to select the statistics and years that we wanted in Baseball Savant and it would enabled us to download the data directly from the website. After running through multiple variations of datasets and experimenting with different attributes, we decided to combine the data from multiple years and remove statistics that weren't direct quality of contact statistics.

Since we have such a large amount of data, we decided to use a 80/20 split of our data where a randomized 80% is saved as the training data, and the remaining 20% is reserved as test data. We performed this split using Weka's randomize filter on the full dataset and then the remove percentage filter. We initially started with 30 different attributes including swing and miss percentages, walk percentage, and strikeout percentage. We later realized we wanted to focus more specifically on ball contact statistics for the purposes of our experiment. This information dealt more directly with the class attribute we wanted to predict so it did not make sense to keep unnecessary data in our CSV files we were uploading to Weka.

Data Analysis and Clustering Exploration

In our data analysis we used both clustering and numerical estimation for this project. Our main objective was to see the predictive ability of selected statistical attributes on how many of each hit a player gets in a season. However, another interesting experiment we wanted to explore is to see if we could cluster hitters based on their playing style. This would be only for the 2023 data set. Clustering is one of the two unsupervised learning strategies we learned about in class. This means that unlike classification and numeric estimation (classified), it does not use a class attribute, but looks at similarities between the instances.

Clustering is useful when there is a possibility for groupings in a dataset. Clustering algorithms use distance measures to see how similar instances are to each other, and then organize them into different groups, or clusters. We wanted to see if by running clustering algorithms, we would be able to make out distinct hitter types in the clusters. For example, if a certain cluster had many more home runs, that would be the sluggers, and a grouping of players with a high amount of total hits would be contact hitters.

We looked at K-means for clustering. K-means clustering organizes the instances in a way to minimize the sum of square distances between the instance values and their assigned cluster centroid. There are a couple of steps when running the k-means algorithm. First, the number (k) of clusters needs to be chosen. We chose to have three clusters. Then, for each cluster, a centroid is randomly assigned. As previously stated, the algorithm's goal is to make clusters with a minimized distance to each respective centroid. Then, we assign each instance to its closest centroid. Then, we recomputed the centroids of each cluster, find the new clusters, and repeat until stopping is necessary. Stopping is necessary when the centroids do not change or the points remain in each cluster. We also used the classes to clusters evaluation using player name as this would enable us to classify each player. Our algorithm stopped after 16 iterations and gave us the results on the next page.

Table 3

Table for Simple K-Means Clustering (From 2023)				
Attribute	Full Dataset	Cluster 0	Cluster 1	Cluster 2
Sweet Spot %	34.6%	32.51%	35.76%	35.57%
Barrel Batted Rate %	8.97%	7.93%	5.38%	12.71%
Solid Contact %	6.65%	6.34%	5.95%	7.46%
Flare/Burner %	24.53%	25.56%	25.61%	22.76%
Under %	25.20%	20.84%	28.88%	26.25%
Topped %	30.8%	35.62%	29.51%	27.47%
Weak %	3.56%	3.31%	4.39%	3.15%
Hard Hit %	41.65%	43.72%	33.59%	46.07%
Out of Zone Contact %	58.64%	58.44%	65.69%	53.31%
In Zone Contact %	83.27%	84.23%	87.45%	79.15%
Instances	134	45 (34%)	39 (29%)	50 (37%)

Looking at clusters, we can form hypotheses about how Weka groups players. Cluster 1 seems to make the most contact with the ball, but also makes the most weak contact. It also has the lowest hard hit percentage and barrel batted rate. Cluster 2 seems to be the heavy hitters. It has the highest barrel batted rate by far, as well as the highest hard hit and solid contact percentages. This group has the lowest percentage of weak contact. Cluster 0 seems to be the balanced group. Out of the ten attributes, cluster 0 had the middle value for seven of them. It also had the lowest sweet spot and under percentage, and the highest topped percentage. After looking at this data, we decided to use some of the basic statistics to see how these players performed.

Table 4

Type of Hit Stats for Sample of Clusters			
Attribute	Cluster 0	Cluster 1	Cluster 2
Hits	126.4	141	133
Singles	82	94	83.2
Doubles	24.8	30	29
Triples	2	4.6	3.4
Home Runs	17.6	12.4	29.4
Observations	5	5	5

When doing this short analysis, we decided to pick a few players from each cluster. This was done by choosing every fifth player from each cluster until we get to a five players for each cluster. This confirms our initial finds. Cluster 1 needs to get singles and makes a lot of contact. Cluster 2 contains the sluggers and hits the most home runs. Cluster 0 was in the middle in terms of hitting home runs, but was actually the worst for everything else. This might mean that even though the quality of contact statistics showed they tended to be in the middle, this may only impact home runs. However, we would have to run some numeric regressions to see the true impact of contact quality.

Results

After running linear regression models with class attributes as singles, doubles, triples, and home runs respectively, we found that there is a higher correlation between our selected attributes and singles and homeruns in both the testing and training data. In the test data, the correlation coefficient is 0.75 for singles, and 0.784 for homeruns. For doubles, it was 0.395 and for triples it was 0.268. Looking at the regressions we can see that the most predictive attribute for singles is the flare-burner percentage coefficient of 1.7, which indicates a high influence in the model.

For homeruns the most predictive attribute was solid contact percentage at 0.979. Given these models we can predict that players who had more hits in a season hit more singles and homeruns. Using this information, MLB teams can focus on player development and their quality of contact statistics depending on goals of their organization, whether that means more baserunners or more homeruns/RBI's.

```
Linear Regression Model

single =

     -0.8651 * launch_angle_avg +
      0.7455 * solidcontact_percent +
      1.7013 * flareburner_percent +
     -0.9807 * poorlyunder_percent +
     -0.3859 * hard_hit_percent +
      0.5283 * oz_contact_percent +
      0.6165 * iz_contact_percent +
     10.4437

Time taken to build model: 0.01 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.7454
Mean absolute error                     11.75
Root mean squared error                 14.3839
Relative absolute error                 69.3844 %
Root relative squared error             66.6664 %
Total Number of Instances              186
```

```
Linear Regression Model

home_run =

      0.5725 * launch_angle_avg +
     -0.5021 * sweet_spot_percent +
      2.3086 * barrel_batted_rate +
      0.9797 * solidcontact_percent +
     -0.2046 * poorlyunder_percent +
     -0.3208 * poorlyweak_percent +
     -0.2042 * hard_hit_percent +
      0.1183 * oz_contact_percent +
      0.2913 * iz_contact_percent +
    -10.3713

Time taken to build model: 0 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.7836
Mean absolute error                      4.226
Root mean squared error                  5.9337
Relative absolute error                 54.0245 %
Root relative squared error             62.6625 %
Total Number of Instances              186
```

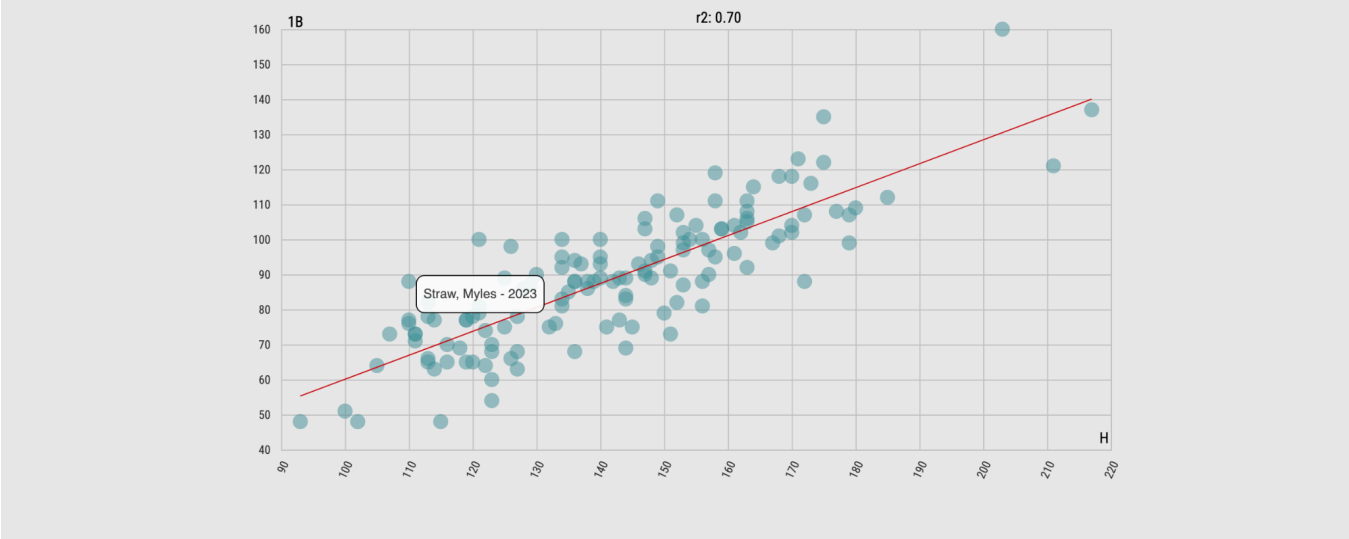


Figure 1: Hits vs. Singles

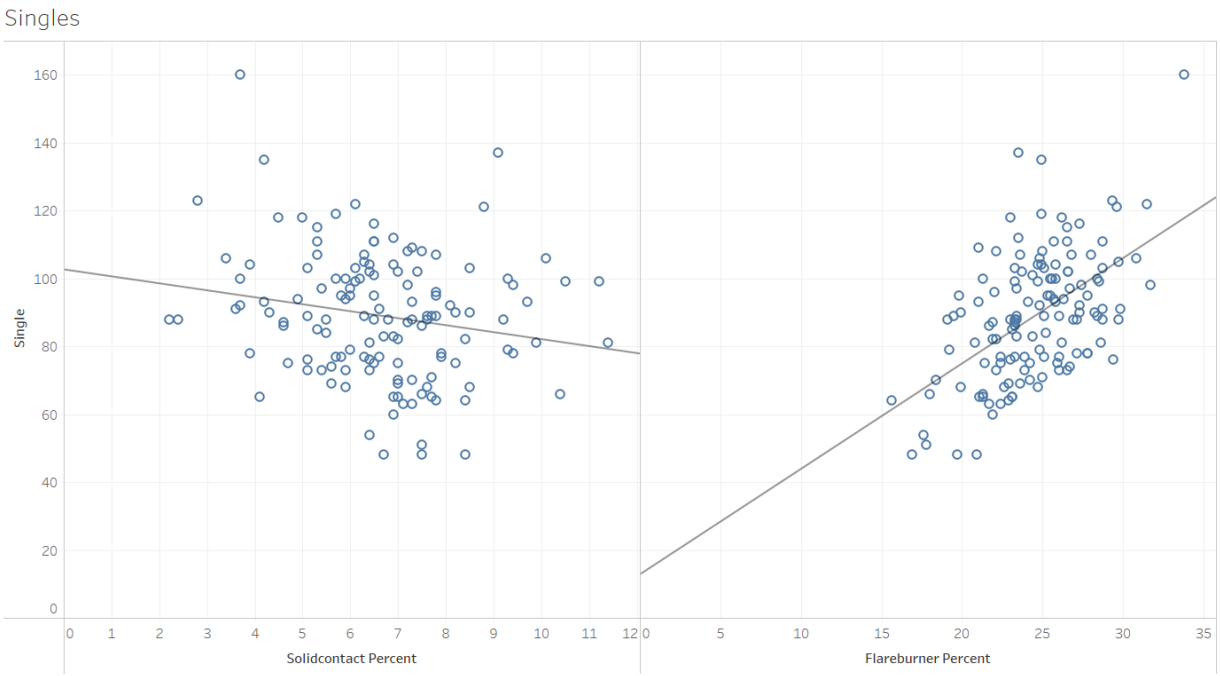


Figure 2: A higher barrel batted rate leads to somewhat less singles, while a higher flame/burner percent leads to more singles.

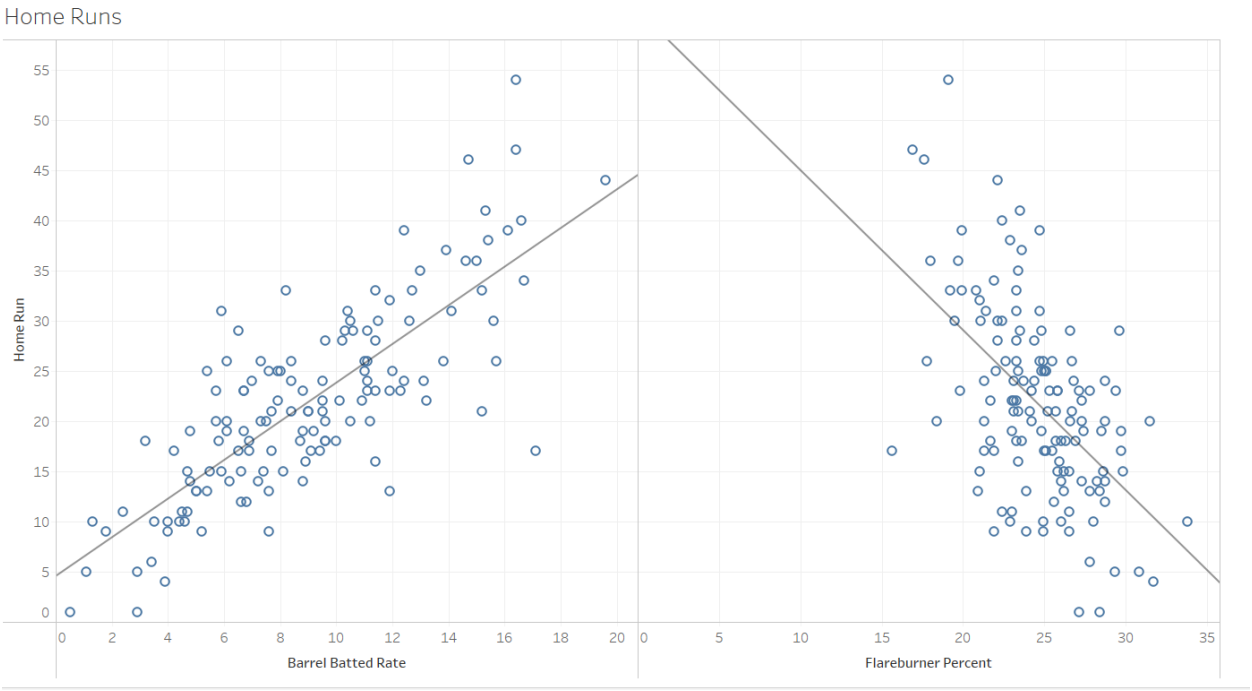


Figure 3: A higher solid contact percent leads to more home runs, while a higher flame/burner percent leads to less.

Conclusions

The main purpose of this project was to see if we could predict the type of hits given the data from Baseball Savant. By focusing specifically on quality of contact metrics sourced from Baseball Savant, we aimed to develop predictive models capable of forecasting player performance. In the preprocessing stage we reduced the overall number of attributes and created training and testing data for our new dataset. With these newly created datasets we ran both classification and numeric estimation algorithms on the numeric datasets.

One conclusion about baseball quality of contact statistics on the type of hit is that they provide valuable insights into the likelihood and effectiveness of different types of hits. By analyzing metrics such as sweet spot percentage, barrel batted rate, and hard-hit percentage, we can gain a deeper understanding of how well a player is making contact with the ball and the potential outcomes of their at-bats. For example, a higher sweet spot percentage indicates that a player consistently hits the ball in the optimal launch angle zone, which may correlate with a higher

likelihood of extra-base hits such as doubles and home runs. Similarly, a higher barrel batted rate suggests that a player makes solid contact with the ball, increasing the probability of hard-hit balls and consequently, more successful hits.

By examining these quality of contact statistics in relation to different types of hits, teams and analysts can identify players who excel at generating specific types of hits and tailor their strategies accordingly. This could involve adjusting batting lineups, defensive positioning, or even player development programs to capitalize on individual strengths and maximize overall team performance.

We discovered that baseball is a very tough sport to try to predict, mainly because there are a plethora of statistics that not only rely on each other, but also are influenced by statistics that we did not use in this experiment. For example, singles, doubles and triples are also influenced by how the defense plays the ball and not just the batter contact. After conducting these tests with the dataset we created we hypothesize that better results may be yielded by including more statistics in future models.

Appendix

Linear Regression Model on Training Data

```
Linear Regression Model

single =

      1.0415 * sweet_spot_percent +
     -1.6631 * solidcontact_percent +
      1.245  * flareburner_percent +
     -1.1752 * poorlyunder_percent +
      0.5265 * oz_contact_percent +
      0.7631 * iz_contact_percent +
    -32.3271

Time taken to build model: 0.08 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.5307
Mean absolute error                     13.1078
Root mean squared error                 16.3068
Relative absolute error                 92.753  %
Root relative squared error             92.0216 %
Total Number of Instances               47
```

```
Linear Regression Model

double =

     -0.4597 * launch_angle_avg +
      0.5589 * solidcontact_percent +
     -0.5603 * poorlytopped_percent +
     -0.773  * poorlyweak_percent +
      0.3448 * iz_contact_percent +
     22.5754

Time taken to build model: 0.01 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.3187
Mean absolute error                      5.2445
Root mean squared error                  6.889
Relative absolute error                 96.2856 %
Root relative squared error             94.836  %
Total Number of Instances               47
```

```
Linear Regression Model

triple =

     -0.2045 * poorlyunder_percent +
     -0.2287 * poorlytopped_percent +
     -0.1513 * hard_hit_percent +
     20.4955

Time taken to build model: 0.01 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.5087
Mean absolute error                      2.0494
Root mean squared error                  3.0331
Relative absolute error                 94.2272 %
Root relative squared error             95.5288 %
Total Number of Instances               47
```

```
Linear Regression Model

home_run =

      0.5578 * launch_angle_avg +
     -0.5757 * sweet_spot_percent +
      2.4874 * barrel_batted_rate +
     -0.2282 * poorlyunder_percent +
     -0.3167 * poorlytopped_percent +
      0.6808 * poorlyweak_percent +
     -0.1298 * hard_hit_percent +
      0.2274 * oz_contact_percent +
     19.8243

Time taken to build model: 0 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.849
Mean absolute error                      5.1701
Root mean squared error                  6.5664
Relative absolute error                 54.3128 %
Root relative squared error             50.6714 %
Total Number of Instances               47
```

Linear Regression Model on Test Data

```
Linear Regression Model

single =

     -0.8651 * launch_angle_avg +
      0.7455 * solidcontact_percent +
      1.7013 * flareburner_percent +
     -0.9807 * poorlyunder_percent +
     -0.3859 * hard_hit_percent +
      0.5283 * oz_contact_percent +
      0.6165 * iz_contact_percent +
     10.4437

Time taken to build model: 0.01 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.7454
Mean absolute error                     11.75
Root mean squared error                 14.3839
Relative absolute error                 69.3844 %
Root relative squared error             66.6664 %
Total Number of Instances              186
```

```
Linear Regression Model

double =

     -0.2466 * launch_angle_avg +
      0.3112 * sweet_spot_percent +
     -0.1794 * barrel_batted_rate +
     -0.32   * poorlytopped_percent +
     -0.3724 * poorlyweak_percent +
      0.3107 * hard_hit_percent +
      0.154  * oz_contact_percent +
      0.3047 * iz_contact_percent +
    -12.2662

Time taken to build model: 0.01 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.3949
Mean absolute error                      5.5939
Root mean squared error                  6.7934
Relative absolute error                 89.8689 %
Root relative squared error             90.8439 %
Total Number of Instances              186
```

```
Linear Regression Model

triple =

     -0.0953 * launch_angle_avg +
     -0.0797 * barrel_batted_rate +
     -0.0911 * flareburner_percent +
     -0.107  * poorlyunder_percent +
     -0.1203 * poorlytopped_percent +
     -0.1747 * poorlyweak_percent +
     -0.1006 * hard_hit_percent +
     17.7772

Time taken to build model: 0 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.2683
Mean absolute error                      1.8313
Root mean squared error                  2.3538
Relative absolute error                 96.2617 %
Root relative squared error             96.5446 %
Total Number of Instances              186
```

```
Linear Regression Model

home_run =

      0.5725 * launch_angle_avg +
     -0.5021 * sweet_spot_percent +
      2.3086 * barrel_batted_rate +
      0.9797 * solidcontact_percent +
     -0.2046 * poorlyunder_percent +
     -0.3208 * poorlyweak_percent +
     -0.2042 * hard_hit_percent +
      0.1183 * oz_contact_percent +
      0.2913 * iz_contact_percent +
    -10.3713

Time taken to build model: 0 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.7836
Mean absolute error                      4.226
Root mean squared error                  5.9337
Relative absolute error                 54.0245 %
Root relative squared error             62.6625 %
Total Number of Instances              186
```

Cluster Information

=== Run information ===

S
Instances: 134
Attributes: 11
sweet_spot_percent
barrel_batted_rate
solidcontact_percent
flareburner_percent
poorlyunder_percent
poorlytopped_percent
poorlyweak_percent
hard_hit_percent
oz_contact_percent
iz_contact_percent
Ignored:
ï»¿
Test mode: Classes to clusters evaluation on training data

=== Clustering model (full training set) ===

kMeans
======

Number of iterations: 16
Within cluster sum of squared errors: 30.620400713438507

Initial starting points (random):

Cluster 0: 30.3,8.4,7.3,24.1,23.9,33.7,2.5,46.5,56,86.6
Cluster 1: 35.4,5,6.3,28.4,23.6,33.6,2.6,37.6,61.8,91.7
Cluster 2: 36.9,11.1,6.6,28.7,19.5,32.4,1.6,47.3,47.3,82.1

Missing values globally replaced with mean/mode

Final cluster centroids:
Cluster#

Attribute	Full Data (134.0)	0 (45.0)	1 (39.0)	2 (50.0)
sweet_spot_percent	34.597	32.5067	35.7615	35.57
barrel_batted_rate	8.9724	7.9311	5.3821	12.71
solidcontact_percent	6.6455	6.3444	5.9487	7.46
flareburner_percent	24.5306	25.5622	25.6077	22.762
poorlyunder_percent	25.2015	20.8422	28.8821	26.254
poorlytopped_percent	30.803	35.62	29.5128	27.474
poorlyweak_percent	3.5649	3.3133	4.3923	3.146
hard_hit_percent	41.647	43.7178	33.5923	46.066
oz_contact_percent	58.6358	58.4378	65.6923	53.31
iz_contact_percent	83.2716	84.2289	87.4538	79.148

Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0 45 (34%)
1 39 (29%)
2 50 (37%)

Class attribute: ï»¿
Classes to Clusters:

0 1 2 <-- assigned to cluster
0 1 0 | Friedl Jr., TJ
0 1 0 | Candelario, Jeimer
1 0 0 | Springer III, George
0 1 0 | France, Ty
0 1 0 | India, Jonathan
1 0 0 | Nimmo, Brandon
1 0 0 | Lowe, Nathaniel
1 0 0 | Bell, Josh
1 0 0 | LeMahieu, DJ

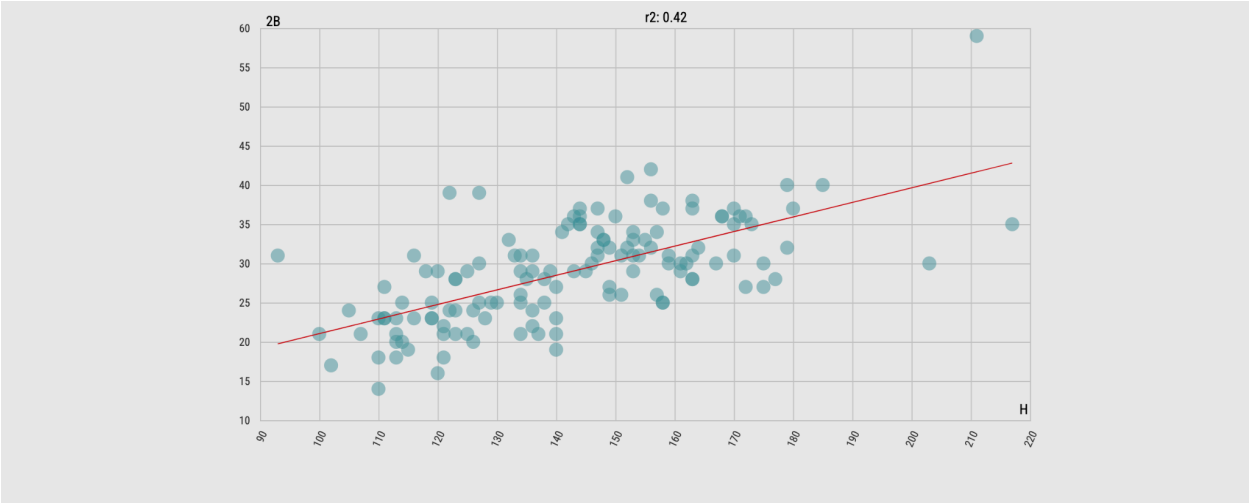
1 0 0 | Suzuki, Seiya
0 0 1 | Santander, Anthony
0 1 0 | Kwan, Steven
0 0 1 | Freeman, Freddie
1 0 0 | Taveras, Leody
1 0 0 | Soto, Juan
0 0 1 | Arozarena, Randy
1 0 0 | Hayes, Ke'Bryan
0 1 0 | Semien, Marcus
0 0 1 | Lindor, Francisco
0 0 1 | Rosario, Eddie
1 0 0 | Davis, J.D.
1 0 0 | Reynolds, Bryan
0 0 1 | Tucker, Kyle
0 0 1 | Suwinski, Jack
1 0 0 | Tovar, Ezequiel
0 1 0 | HernÃ¡ndez, Enrique
1 0 0 | DÃaz, Yandy
1 0 0 | PeÃ±a, Jeremy
1 0 0 | Thomas, Lane
1 0 0 | Bohm, Alec
1 0 0 | RodrÃguez, Julio
0 0 1 | Olson, Matt
0 0 1 | Grisham, Trent
1 0 0 | DÃaz, Elias
0 0 1 | Raleigh, Cal
0 1 0 | Turner, Justin
0 0 1 | Goldschmidt, Paul
0 1 0 | Hoerner, Nico
0 0 1 | Muncy, Max
0 1 0 | Profar, Jurickson
0 0 1 | Happ, Ian
0 0 1 | Realmuto, J.T.
0 1 0 | Stott, Bryson
0 0 1 | Witt Jr., Bobby
0 1 0 | Arraez, Luis
0 0 1 | Seager, Corey
1 0 0 | Edman, Tommy

0 0 1 | Riley, Austin
0 0 1 | Perez, Salvador
0 1 0 | Crawford, J.P.
0 0 1 | SuÃ¡rez, Eugenio
0 0 1 | Devers, Rafael
0 0 1 | Chapman, Matt
0 0 1 | Casas, Triston
0 1 0 | RamÃrez, JosÃ©
1 0 0 | Stephenson, Tyler
0 1 0 | McKinstry, Zach
0 1 0 | Verdugo, Alex
1 0 0 | Harris II, Michael
0 1 0 | Abrams, CJ
0 0 1 | Drury, Brandon
1 0 0 | Arcia, Orlando
0 1 0 | Varsho, Daulton
1 0 0 | Correa, Carlos
1 0 0 | Yoshida, Masataka
0 1 0 | Benintendi, Andrew
1 0 0 | Nootbaar, Lars
1 0 0 | Gurriel Jr., Lourdes
0 0 1 | Melendez Jr., MJ
0 0 1 | Walker, Christian
1 0 0 | Garcia, Maikel
0 0 1 | Betts, Mookie
0 0 1 | Robert Jr., Luis
0 1 0 | Paredes, Isaac
0 1 0 | Cronenworth, Jake
1 0 0 | Rosario, Amed
0 1 0 | Albies, Ozzie
1 0 0 | Anderson, Tim
0 0 1 | HernÃ¡ndez, Teoscar
0 0 1 | Torkelson, Spencer
0 1 0 | Canha, Mark
1 0 0 | Bogaerts, Xander
0 0 1 | McMahon, Ryan
0 0 1 | Adames, Willy
0 1 0 | Straw, Myles

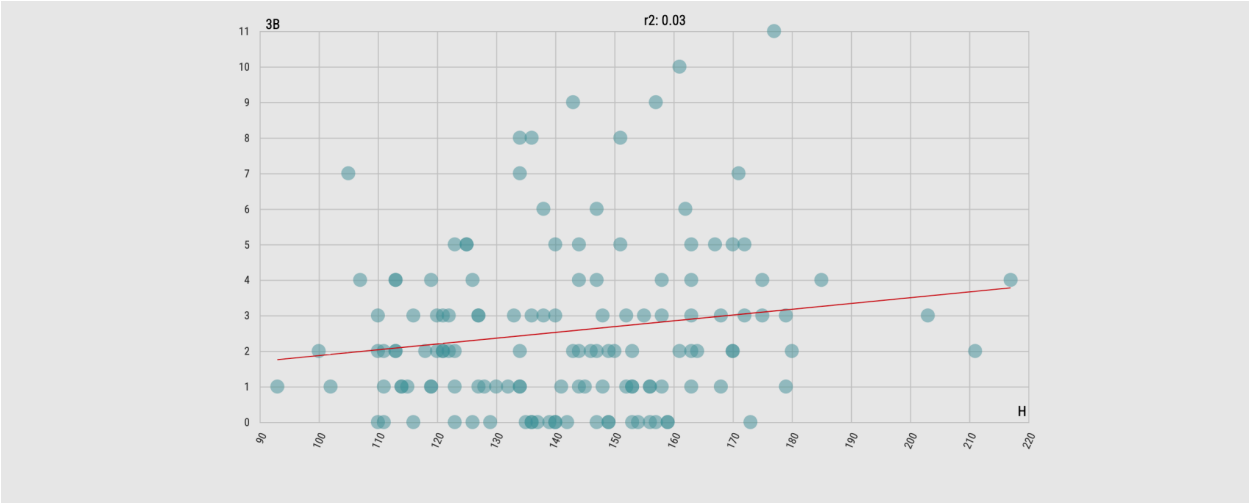
0 0 1 | Turner, Trea
0 0 1 | Volpe, Anthony
0 1 0 | Kim, Ha-Seong
0 1 0 | Wade Jr., LaMonte
0 0 1 | Schwarber, Kyle
0 0 1 | GarcÃa, Adolis
0 1 0 | Smith, Will
0 1 0 | McNeil, Jeff
0 1 0 | GimÃ©nez, AndrÃ©s
0 1 0 | Steer, Spencer
0 1 0 | Ruiz, Keibert
0 0 1 | Tatis Jr., Fernando
0 0 1 | Burger, Jake
0 0 1 | Swanson, Dansby
1 0 0 | Guerrero Jr., Vladimir
1 0 0 | Santana, Carlos
1 0 0 | Hays, Austin
0 1 0 | Estrada, Thairo
0 0 1 | Harper, Bryce
0 1 0 | Merrifield, Whit
0 1 0 | Smith, Dominic
0 0 1 | Ohtani, Shohei
0 1 0 | Rutschman, Adley
0 0 1 | Alonso, Pete
0 0 1 | Soler, Jorge
1 0 0 | Carroll, Corbin
0 1 0 | Torres, Gleyber
0 1 0 | Bregman, Alex
1 0 0 | BÃ¡ez, Javier
0 0 1 | De La Cruz, Bryan
1 0 0 | Vierling, Matt
1 0 0 | Abreu, JosÃ©
1 0 0 | Marte, Ketel
0 0 1 | Castellanos, Nick
0 0 1 | Rooker Jr., Brent
0 1 0 | Arenado, Nolan
1 0 0 | Yelich, Christian
0 0 1 | Ozuna, Marcell

1 0 0 | Contreras, William
1 0 0 | AcuÃ±a Jr., Ronald
0 0 1 | Outman, James
0 1 0 | Bellinger, Cody
0 0 1 | Machado, Manny
1 0 0 | Renfroe, Hunter
0 0 1 | Jung, Josh
1 0 0 | Bichette, Bo
1 0 0 | Vaughn, Andrew
1 0 0 | Meneses, Joey
0 0 1 | Henderson, Gunnar

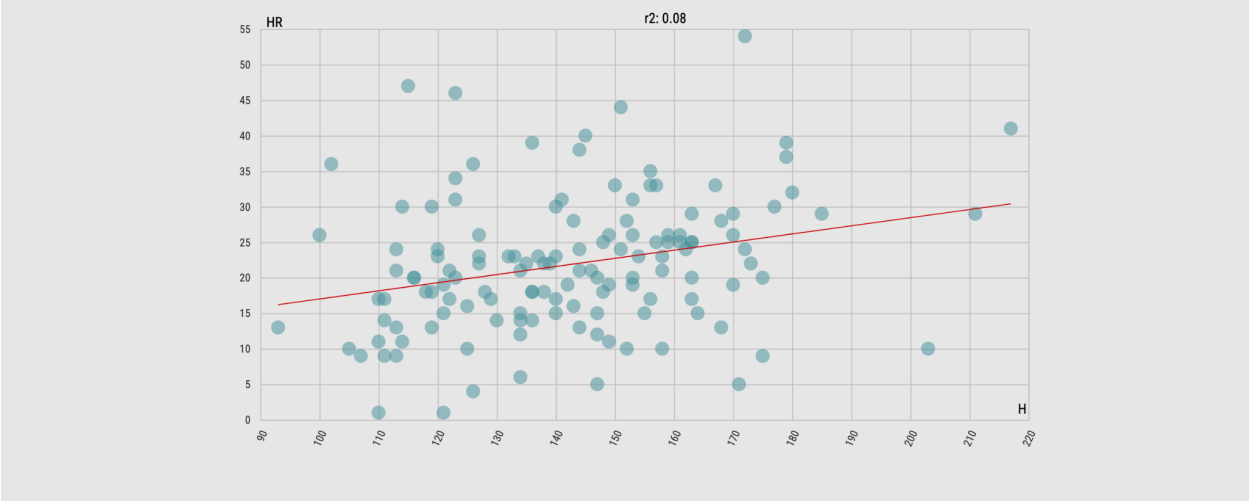
Linear Regression Visualized VIA BaseballSavant



Hits and Doubles



Hits and Triples



Hits and Home Runs

References

Baseball Savant:
https://baseballsavant.mlb.com/leaderboard/custom?year=2023&type=batter&filter=&min=q&selections=pa%2Ck_percent%2Cbb_percent%2Cwoba%2Cxwoba%2Csweet_spot_percent%2Cbarrel_batted_rate%2Chard_hit_percent%2Cavg_best_speed%2Cavg_hyper_speed%2Cwhiff_percent%2Cswing_percent&chart=false&x=pa&y=pa&r=no&chartType=beeswarm&sort=xwoba&sortDir=desc

Sports Analytics: How Different Sports Use Data Analytics:
https://www.datacamp.com/blog/sports-analytics-how-different-sports-use-data-analysis

Term Glossary:
https://www.mlb.com/glossary/statcast/sweet-spot

How to Find Raw Data:

https://sabr.org/sabermetrics/data

The Ultimate Guide to K-Means Clustering: Definition, Methods and Applications:

https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/