Andrew Fitzsimmons
11/17/2022
DSC478

Final Project - Main Report
Major League Baseball - All-Star Predictions

Team: Andrew Fitzsimmons - Graduate Student (Data Science)

Type of Project: Data Analysis

Table of Abbreviations:

Abbreviation	Definition
ASCII	American Standard Code for Information Interchange
PA	Plate Appearances
KNN	K Nearest Neighbors
MLB	Major League Baseball
BA	Batting Average (determined by dividing a player's hits by his total at-bats for a number between 0 and 1)
OBP	On Base Percentage (how frequently a batter reaches base per plate appearance)
SLG	Slugging Percentage (the total number of bases a player records per at-bat)
OPS	On Base Plus Slugging (adds on-base percentage and slugging percentage)

Project Goal
This project is a data analysis project that serves to analyze Major League Baseball offensive statistics by player. My goal was to build a machine learning algorithm to predict/classify whether a player will be selected to make the all-star team based on similarity to offensive statistics of players from previous years, and whether or not they were selected for the all-star team. I also will be performing qualitative cluster analysis using Kmeans clustering.

Design Decisions

- Only position players' offensive statistics will be reviewed for all-star prediction.
- Pitchers will be excluded, and defensive metrics will be excluded.

- Only the first half (opening day to the all-star break) of each season will be reviewed since this is the period within which all-star selection takes place.
- American League and National League players will be grouped together and not separated for the sake of this analysis.
- 100 Plate Appearances will gate eligibility for consideration for an all-star spot. This is based on histogram review of PA values of All-Stars.
- Since the dataset is biased towards non all-stars (12% of observations in the training data set are classified as all-stars), I did not use KNN to classify based solely on the majority class of 'k' nearest neighbors. Instead, I used each instance of an all-star in the neighborhood as a 'vote' towards that player's all-star selection. I then calculated a 'total distance' metric for each player that received votes by summing the distances of all all-stars in the neighborhood. Finally, players were ranked from most votes to least, and sorted within their '# of votes' groups from lowest to highest based on the 'total distance' metric.
- Data from seasons [2017, 2018, 2019, 2021] will be used as the training set.
- Data from the 2022 season will be used as the test set.

Data Import and Cleaning

I initially imported the data into R using the 'baseballr' package. The data was pulled using the 'daily_batter_bref' function, which pulls standard batting statistics for all players within a specified date range from baseballreference.com. I pulled player data ranging from the beginning of each season to the all star game from 2017-2022 (excluding 2020 since there was no all star game that year due to covid). This equated to about 3 months of stats per season. I cleaned and formatted the resulting data set so that it would be ready for exploratory analysis and modeling. The primary data cleaning/preprocessing activity for this project consisted of joining my imported player stats dataset with my manually collected dataset on which players made the all star team for each year within the dataset.

All-Star Target Data (1 = Selected for All-Star Team)

yearName	AS_Roster
2018 Wander Suero	0
2018 Austin Voth	0
2017 Bryce Harper	1
2017 Daniel Murphy	1
2017 Anthony Rendon	0
2017 Ryan Zimmerman	1

Standard Batting Statistics

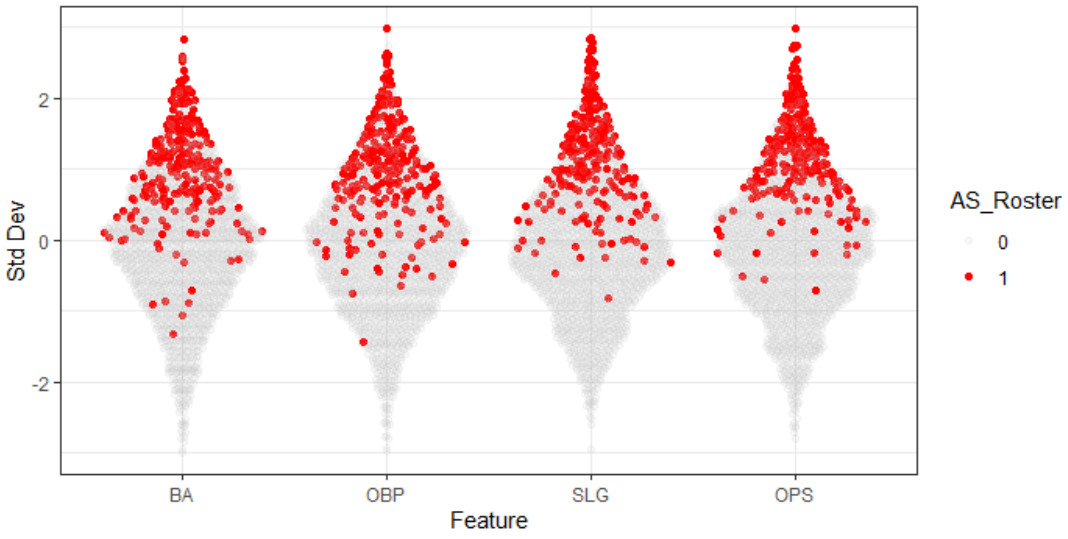
bbref_id	season	Name	Age	Level	Team	G	PA	AB	R	H
668731	2022	Christian Walker	31	Maj-NL	Arizona	90	370	313	44	64
669134	2022	Ketel Marte	28	Maj-NL	Arizona	82	336	292	45	79
668843	2022	Daulton Varsho	25	Maj-NL	Arizona	84	335	301	39	70
592273	2022	Geraldo Perdomo	22	Maj-NL	Arizona	83	289	250	31	50
621453	2022	David Peralta	34	Maj-NL	Arizona	80	286	259	26	61
660650	2022	Josh Rojas	28	Maj-NL	Arizona	61	253	221	34	60

List of Variables

```
data                                        1036 obs. of 30 variables
    $ bbref_id: chr [1:1036] "547989" "660670" "642715" "571431" ...
    $ season  : int [1:1036] 2021 2021 2021 2021 2021 2021 2021 2021 2021 2021 ...
    $ Name    : chr [1:1036] "Marcus Semien" "Whit Merrifield" "Freddie Freeman" "Vladimir Guerrero Jr." ...
    $ Age     : num [1:1036] 30 32 31 22 24 26 23 30 26 32 ...
    $ Level   : chr [1:1036] "Maj-AL" "Maj-AL" "Maj-NL" "Maj-AL" ...
    $ Team    : chr [1:1036] "Toronto" "Kansas City" "Atlanta" "Toronto" ...
    $ G       : num [1:1036] 154 155 153 153 150 151 151 149 152 145 ...
    $ PA      : num [1:1036] 687 686 672 664 662 658 656 655 655 655 ...
    $ AB      : num [1:1036] 616 635 581 577 605 612 607 585 591 578 ...
    $ R       : num [1:1036] 111 90 118 117 102 85 116 105 86 81 ...
    $ H       : num [1:1036] 166 172 176 181 160 162 181 146 163 155 ...
    $ X1B     : num [1:1036] 83 122 119 107 85 109 127 87 117 120 ...
    $ X2B     : num [1:1036] 39 40 25 28 38 41 27 20 37 24 ...
    $ X3B     : num [1:1036] 2 3 2 1 7 3 1 2 0 1 ...
    $ HR      : num [1:1036] 42 7 30 45 30 9 26 37 9 10 ...
    $ RBI     : num [1:1036] 97 64 81 105 104 53 99 92 52 55 ...
    $ BB      : num [1:1036] 65 38 81 82 47 34 40 53 55 68 ...
    $ IBB     : num [1:1036] 0 1 15 7 2 1 0 2 0 2 ...
    $ uBB     : num [1:1036] 65 37 66 75 45 33 40 51 55 66 ...
    $ SO      : num [1:1036] 138 99 105 106 124 94 129 158 107 93 ...
    $ HBP     : num [1:1036] 3 3 8 3 3 6 5 9 4 4 ...
    $ SH      : num [1:1036] 0 0 0 0 0 2 0 0 1 0 ...
    $ SF      : num [1:1036] 3 10 2 2 7 4 4 8 4 5 ...
    $ GDP     : num [1:1036] 9 12 11 19 4 4 9 12 10 15 ...
    $ SB      : num [1:1036] 13 40 8 4 19 27 25 1 3 4 ...
    $ CS      : num [1:1036] 1 4 3 1 4 5 1 0 6 2 ...
    $ BA      : num [1:1036] 0.27 0.271 0.303 0.314 0.265 0.265 0.298 0.25 0.276 0.268 ...
    $ OBP     : num [1:1036] 0.341 0.311 0.394 0.401 0.317 0.308 0.345 0.318 0.339 0.347 ...
    $ SLG     : num [1:1036] 0.544 0.376 0.508 0.6 0.499 0.386 0.475 0.48 0.384 0.365 ...
    $ OPS     : num [1:1036] 0.884 0.687 0.902 1 0.816 0.694 0.819 0.798 0.724 0.712 ...
    - attr(*, "baseballr_timestamp")= POSIXct[1:1], format: "2022-10-14 23:18:27"
    - attr(*, "baseballr_type")= chr "MLB Daily Batter data from baseball-reference.com"
```

Exploratory Visualization

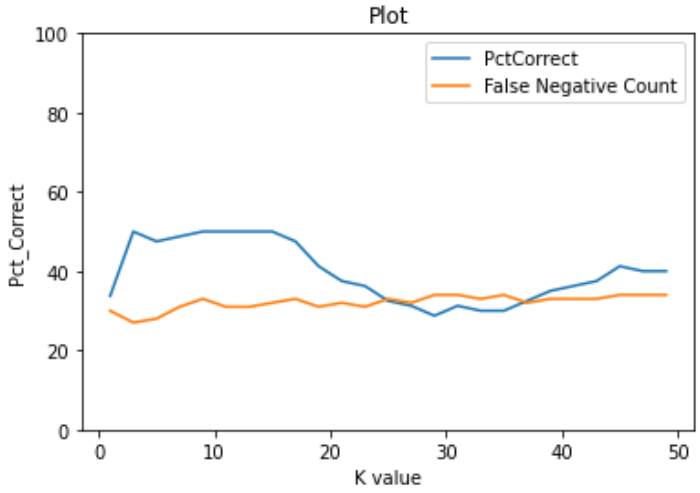
I performed exploratory visualization in R. I first checked histograms of my attributes on only the records of 'all-stars' in my dataset. I noticed that the majority of all-stars tend to be under 30, and that none had less than 100 Plate Appearances (100 PA was 4.68 std dev from average PA for all-stars). Having greater than 100 plate appearances became the initial filter that I used on my test and training datasets before sending them to KNN for classification. I then normalized feature values for each player based on z score within each season (i.e. performance relative to the rest of the league for a given season). Finally, I created beeswarm plots to visualize the distribution of z-scores of all attributes for all players. I coded all-star players as red on the plots to better identify any patterns in attributes that separate all-stars from the rest of the players in the league. The image below shows that in all 4 of the primary offensive metrics (BA, OBP, SLG, and OPS), All-Stars tend to cluster to the higher extreme of the distribution.



Classification and Clustering

I utilized a KNN algorithm for my classification. In order to select the best value of ‘k’, I generated counts of true positive, true negative, false positive, and false negative predictions at odd values of k between 1 and 50. I then plotted the false negative counts at each k with the total percent of correct all-star picks. I also tried different distance metrics (cosine vs euclidean). K value of 3, and euclidean distance metric were selected for the final model.

Percentage of Correct All-Star Predictions and False Negative Count by varying ‘K’ val



Since my dataset is biased towards non all-stars (12% of observations in the training data set are classified as all-stars), I did not use KNN to classify based solely on the majority class of ‘k’ nearest neighbors. Instead, I used each instance of an all-star in the neighborhood as a ‘vote’ towards that player’s all-star selection. I then calculated a ‘total distance’ metric for each player that received votes by summing the distances of all all-stars in the neighborhood. Finally, players were ranked from most votes to least, and sorted within their ‘# of votes’ groups from lowest to highest based on the ‘total distance’ metric. This resulted in a neatly sorted list that reads from top to bottom as the ‘most likely to be selected’ to ‘least likely to be selected’. I found that the prediction accuracy for #of votes is highest for individuals with 3 votes (maximum votes), and lowest for individuals with 1 vote (minimum number of votes to be eligible for selection).

Prediction Accuracy Based on Votes Recieved

```
Observed Accuracy Based on Votes Recieved:
       AS_Selection
Votes
1          0.441176
2          0.565217
3          1.000000
```

Sorted All-Star Voting Table

	Name	Votes	AS_Selection	Distance Score
97	2022 Garrett Cooper	3	1	0.6360958863051904
4	2022 Freddie Freeman	3	1	0.8575271499193013
13	2022 Pete Alonso	3	1	0.8812986297197546
32	2022 Xander Bogaerts	3	1	0.8953661815444751
29	2022 Rafael Devers	3	1	0.9452037248003773
98	2022 Mike Trout	3	1	1.094442095534846
20	2022 Aaron Judge	3	1	1.1747228821853903
167	2022 Bryce Harper	3	1	1.2073150765724685
18	2022 Paul Goldschmidt	3	1	1.2115767808945368
111	2022 Yordan Alvarez	3	1	1.3991805976288478
23	2022 Shohei Ohtani	2	1	0.3614264712073376
108	2022 Giancarlo Stanton	2	1	0.41107161465611003
102	2022 Luis Robert	2	0	0.4195858357455855
11	2022 Vladimir Guerrero Jr.	2	1	0.4364536940444583
15	2022 Dansby Swanson	2	1	0.46026905552221686
196	2022 Josh Naylor	2	0	0.4664091242245496
60	2022 Manny Machado	2	1	0.49539721075763987
36	2022 Nolan Arenado	2	1	0.5008522031509858
12	2022 JosAC Abreu	2	0	0.5166013085700447
9	2022 Trea Turner	2	1	0.5228436179379882
30	2022 Brandon Nimmo	2	0	0.5546022760578901
3	2022 Francisco Lindor	2	0	0.5579963210500911
14	2022 Josh Bell	2	0	0.5711902876250146
1	2022 Jake Cronenworth	2	1	0.5789970018758763

I also performed an exploratory Kmeans cluster analysis on the dataset to identify and analyze interesting groupings of players by centroid attributes by returning the top ‘n’ nearest players to each cluster centroid. I ran kmeans multiple times for different values of ‘k’. I ultimately selected k=4 since it seemed to give the most interesting segmentation of player groups. ‘Centroid 0’ appeared to represent players with very few plate appearances. These are either players who have been injured, or players who are just on the edge of making the Major league team and possibly spend some time in the minors with limited time in the majors. ‘Centroid 1’ and ‘Centroid 2’ both appeared to be geared towards players with a higher number of plate appearances, so typically better players. However, the power numbers appear to be higher in ‘Centroid 1’, with more home runs, RBIs, and higher slugging percentage. ‘Centroid 2’ on the other hand, had higher hits, and batting average. Finally, ‘Centroid 3’ represents average

players that get more playing time than the ‘Centroid 0' players, but less than the everyday players.

Kmeans Clustering Centroids

	PA	R	H	X1B	X2B	HR	RBI	BB	BA	OBP	SLG	OPS
0	0.101420	0.147785	0.155944	0.145504	0.165063	0.123077	0.157590	0.136251	0.498263	0.501032	0.397705	0.427164
1	0.700044	0.528127	0.553335	0.420927	0.468483	0.491181	0.574908	0.433684	0.511912	0.534656	0.515198	0.522835
2	0.682149	0.459767	0.600339	0.577079	0.481841	0.233328	0.406732	0.301490	0.580543	0.514303	0.402197	0.436579
3	0.346374	0.288125	0.331810	0.292063	0.298148	0.235436	0.301903	0.231312	0.527887	0.512072	0.449310	0.468811

I was able to generate a table of the top 5 players nearest to each centroid, and rejoin those players with their original batting data. Seeing the real player data represented by the centroids confirms that ‘Centroid 3’ appears to represent good players with average # of PA. ‘Centroid 2’ represents players with many PA and a high batting average and # of singles. ‘Centroid 1’ represents power hitters with HR totals ranging from [12:19] for players in the top 5 nearest to that centroid. Finally, ‘Centroid 0’ with a low number of plate appearances and as a result, a lower number of values in the “count” attributes (Hits, Runs, Singles, etc).

Kmeans Qualitative Analysis

	Name	Centroid	Distance	PA	R	H	X1B	X2B	HR	RBI	BB	BA	OBP	SLG	OPS
206	2022 Jace Peterson	3	0.003	241	33	54	32	12	8	30	22	0.252	0.325	0.439	0.764
205	2022 Christopher Morel	3	0.0044	241	36	57	34	10	9	26	23	0.266	0.338	0.477	0.814
256	2022 Aledmys DA-az	3	0.005	195	20	42	30	6	6	21	14	0.233	0.292	0.367	0.659
255	2022 Emmanuel Rivera	3	0.0053	195	22	42	25	8	6	21	11	0.231	0.282	0.407	0.689
167	2022 Bryce Harper	3	0.0058	275	49	77	40	21	15	48	26	0.318	0.386	0.599	0.985
119	2022 Gio Urshela	2	0.0021	308	33	74	49	16	8	36	21	0.262	0.308	0.410	0.718
12	2022 JosAC Abreu	2	0.0026	395	52	105	70	24	11	46	44	0.304	0.387	0.470	0.857
103	2022 J.T. Realmuto	2	0.0042	322	42	72	48	14	8	38	24	0.252	0.323	0.399	0.722
14	2022 Josh Bell	2	0.0044	394	49	106	69	21	13	50	42	0.311	0.390	0.504	0.895
68	2022 Brendan Rodgers	2	0.0046	348	41	83	52	20	9	47	24	0.260	0.313	0.420	0.733
23	2022 Shohei Ohtani	1	0.0028	382	51	86	50	15	19	56	44	0.258	0.348	0.487	0.835
22	2022 Rhys Hoskins	1	0.004	386	49	82	43	18	19	44	45	0.244	0.337	0.479	0.816
78	2022 Matt Chapman	1	0.0042	340	45	69	38	16	15	45	29	0.227	0.300	0.428	0.728
91	2022 Wilmer Flores	1	0.0042	334	47	74	44	17	13	51	32	0.253	0.335	0.445	0.781
133	2022 Luke Voit	1	0.0046	302	33	58	31	15	12	40	34	0.220	0.308	0.413	0.721
338	2022 Christian Arroyo	0	0.0046	127	14	26	18	4	4	13	6	0.224	0.278	0.362	0.640
317	2022 Orlando Arcia	0	0.0057	146	13	33	24	5	4	18	14	0.256	0.329	0.388	0.716
322	2022 Kris Bryant	0	0.0059	142	22	38	26	8	4	12	13	0.302	0.366	0.460	0.827
341	2022 Marwin Gonzalez	0	0.0059	124	12	26	16	7	3	10	9	0.234	0.301	0.378	0.679
323	2022 Brandon Lowe	0	0.0102	142	24	31	17	7	5	12	13	0.246	0.324	0.452	0.776

Conclusion

From the final sort of my KNN results (sorted by number of votes and total distance of voters), I selected the top 60 players as my all-star predictions. Each all-star team (American League and National League) has 25 offensive players each. I chose to select 60 total players for my prediction set because I feel my algorithm would be most useful not in directly picking all-stars, but rather to suggest a pool from which likely all-star candidates will be drawn. For this reason, my all-star selection pool was 10% greater than the total roster capacities.

When running the KNN algorithm on the test set of 2022 MLB players, with k=3, I was able to correctly predict 66% of the 2022 MLB All-Stars. I am happy with the performance of the KNN algorithm in this application, especially considering that All-Star targets only make up 12% of the overall training data population.

KNN Output - Selection Accuracy

64	2022 J.P. Crawford	1	0	0.22026296147877 65
120	2022 Kyle Farmer	1	0	0.22310078623669322
41	2022 Ian Happ	1	1	0.22407529417136507
122	2022 Andrew Vaughn	1	0	0.2252768511290169
150	2022 Taylor Ward	1	0	0.22605293617405567
87	2022 Starling Marte	1	1	0.22722447387176498
26	2022 Julio Rodriguez	1	1	0.23590644924515422
173	2022 Joc Pederson	1	1	0.23669102435470446
24	2022 C.J. Cron	1	1	0.23966647772451677
79	2022 Brandon Drury	1	0	0.24850309077548932
2	2022 Bo Bichette	1	0	0.25508246286315317
156	2022 Andres Gimenez	1	1	0.26257273541638054
101	2022 Jose Altuve	1	1	0.27389416505161235
182	2022 Brendan Donovan	1	0	0.2813590248871035
89	2022 Mookie Betts	1	1	0.2877411539559512
110	2022 Steven Kwan	1	0	0.2969947931680238

When Selecting 60 Players of a possible 50, selections were 66.0% correct.

My Kmeans cluster analysis returned some interesting groupings of players. The top 5 players nearest to each centroid seemed to be fairly well defined by their grouping. The kmeans clustering did a good job to separate ‘power hitters’ from ‘base hitters’. Centroids 1 and 2 appeared to be the most useful since they provided unique separation between ‘good’ players. Whereas Centroids 0 and 3 appeared to separate the more massive part of the data set which is average to below average players, and players with minimal Plate Appearances.

List of Appendices

Appendix 1 - All Star Prediction KNN and Cluster Analysis (.ipynb)
Appendix 2 - All Star Prediction KNN and Cluster Analysis (.html)
Appendix 3 - Data Cleaning (.R)
Appendix 4 - Exploratory Visualization and Preliminary Data Analysis (.R)
Appendix 5 - Dataset for KNN and Clustering (.csv)
Appendix 6 - All Star Rosters (.csv)

References

- Acquiring and Analyzing Baseball Data - baseballr
https://billpetti.github.io/baseballr/index.html

- Developing the baseballr package for R
https://tht.fangraphs.com/developing-the-baseballr-package-for-r/

- 2022 all-star Game Rosters
https://www.mlb.com/news/2022-all-star-game-rosters