Ben Kaufmann

12/13/2022

Stat 419

# Final Report: Comparing Classification Techniques on Baseball Statistics

## Abstract

For this project, a dataset of over two-thousand retired professional baseball players with their hitting and defensive statistics was created. Several different statistical methods (SVM, k-Nearest Neighbors, and perceptron) were then used with the objective of finding which method yields the most accurate results when tested with unclassified data. To do this, the dataset was input, along with their classification of whether they are in the Hall of Fame to train each model. The variables chosen, different baseball statistics, needed to be frequently adjusted to account for external factors that may have skewed the final test results. After each model had been adequately trained, each model was tested with unclassified data and compared their classifications with the data's actual classifications. The accuracy of the results of the different models were then compared with one another, and those results were analyzed.

## Keywords

SVM (Support Vector Machine): A supervised learning model that uses hyperplanes to separate and then classify data.

KNN (K-Nearest-Neighbors): A supervised learning algorithm that classifies data by grouping similar data points together.

Perceptron: A supervised learning algorithm that binarily classifies data by adjusting variable weights over many iterations to find the impact of each variable on the classification variable.

PA (Plate Appearances): Cumulative statistic for the number of times a batter completes a turn batting.

dWAR (Defensive Wins Above Replacement): Cumulative statistic that estimates the defensive value of a player.

R (Runs): Cumulative statistic for the number of runs scored by a player.

H (Hits): Cumulative statistic for the number of hits scored by a player.

1B: Cumulative statistic for the number of singles scored by a player.

2B: Cumulative statistic for the number of doubles scored by a player.

3B: Cumulative statistic for the number of triples scored by a player.

HR: Cumulative statistic for the number of home runs scored by a player.

RBI (Runs Batted In): Cumulative statistic for the number of runs batted in by a player.

SB (Stolen Base): Cumulative statistic for the number of bases stolen scored by a player.

BB (Bases on Balls): Cumulative statistic for the number of walks scored by a player.

BA (Batting Average): Non-cumulative statistic calculated by a player's hits divided by their total at-bats.

OBP (On Base Percentage): Non-cumulative statistic that measures how often a player reaches base.

SLG (Slugging Percentage): Non-cumulative statistic that represents the total number of bases a player records per at-bat.

OPS (On-base Plus Slugging): Non-cumulative statistic calculated as the sum of a player's on-base percentage and slugging percentage.

OPS+: Non-cumulative statistic that takes a player's on-base plus slugging percentage and normalizes the number across the entire league.

HOF (Hall of Fame): Binary classification variable which denotes whether a player is in the baseball Hall of Fame or not.

# Introduction

Baseball is a game of statistics. Unlike most other sports, almost every instance in a game of baseball can be described using discrete events involving individuals. This makes the game easily measurable and favorable for high-level statistical analysis.

The statistical analysis used was a series of binary classifiers that were trained and then tested using data entries with hidden classifications. The classifiers would attempt to predict the classification of each entry, and the results were analyzed. The goal of this project was to discover which of the classifiers is best suited for this work and to analyze why this is the case.

# Dataset

The dataset was created using the Stathead Baseball advanced search feature. This feature allowed for querying from the Stathead database which houses an exorbitant number of different statistics for every player in professional baseball history. The desired variables were then included and tuples which could create noise and affect the findings of the project were pruned out. The complete dataset is submitted alongside this report.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **Player** | **PA** | **dWAR** | **R** | **H** | **1B** | **2B** | **3B** | **HR** | **RBI** | **SB** | **BB** | **BA** | **OBP** | **SLG** | **OPS** | **OPS+** | **HOF** |
| 2 | Josh Gibson | 2511 | 1.6 | 610 | 806 | 448 | 135 | 58 | 165 | 730 | 40 | 331 | 0.374 | 0.458 | 0.72 | 1.178 | 214 | y |
| 3 | Babe Ruth | 10626 | -2.3 | 2174 | 2873 | 1517 | 506 | 136 | 714 | 2214 | 123 | 2062 | 0.342 | 0.474 | 0.69 | 1.164 | 206 | y |
| 4 | Ted Williams | 9792 | -13.3 | 1798 | 2654 | 1537 | 525 | 71 | 521 | 1839 | 24 | 2021 | 0.344 | 0.482 | 0.634 | 1.116 | 191 | y |
| 5 | Oscar Charleston | 3962 | -1.2 | 862 | 1218 | 750 | 242 | 80 | 146 | 861 | 209 | 484 | 0.364 | 0.448 | 0.614 | 1.062 | 184 | y |
| 6 | Buck Leonard | 2556 | -1.4 | 538 | 728 | 448 | 138 | 47 | 95 | 545 | 34 | 396 | 0.345 | 0.452 | 0.59 | 1.042 | 180 | y |
| 7 | Lou Gehrig | 9665 | -9 | 1888 | 2721 | 1531 | 534 | 163 | 493 | 1995 | 102 | 1508 | 0.34 | 0.447 | 0.632 | 1.08 | 179 | y |
| 8 | Turkey Stearnes | 4291 | 0.2 | 910 | 1319 | 793 | 228 | 112 | 186 | 1001 | 129 | 416 | 0.349 | 0.417 | 0.616 | 1.034 | 177 | y |
| 9 | Rogers Hornsby | 9481 | 13.9 | 1579 | 2930 | 1919 | 541 | 169 | 301 | 1584 | 135 | 1038 | 0.358 | 0.434 | 0.577 | 1.01 | 175 | y |
| 10 | Mule Suttles | 3649 | -3.1 | 717 | 1089 | 624 | 214 | 72 | 179 | 877 | 84 | 370 | 0.34 | 0.41 | 0.619 | 1.029 | 172 | y |

*A snippet of the dataset. Note the HOF classification variable was changed from y/n to 0/1 to fit the classification model's syntax.*

## Variable Selection

When selecting the variables that would be included in the dataset, many things had to be considered. Statistics that best encapsulated the value of the batter and are statistically relevant to the Hall of Fame classification were prioritized for inclusion. Fortunately, standard baseball statistics do an adequate job of fulfilling this task. BA, OBP, SLG, OPS, R, SB, RBI, H, 1B, 2B, 3B, HR, BB, and PA are all commonly used to determine the value of hitters and were subsequently included in the dataset. Other common statistics such as SO (Strikeouts) and CS (Caught Stealing) were removed from the data set since they appeared as null values in the Stathead database for many of the older players since these were not always tracked. CS, for instance, only started being tracked in 1951 (Edes, 2013).

## Cumulative vs. Non-Cumulative Variables

Cumulative statistics such as R, SB, RBI, H, 1B, 2B, 3B, HR, BB, and PA represent the production a player has over their career. Meanwhile, Non-Cumulative statistics such as BA, OBP, SLG, and OPS represent the efficiency a player has over their career. In sports, a player must have both high overall production and efficiency to be considered for the Hall of Fame (Stueve, 2017). For this reason, it was decided to include a healthy mix of both Cumulative and Non-Cumulative statistics to represent both qualities in the dataset.

## Comparing Players of Different Eras

Professional baseball has changed a great deal over the 100+ years it's been played. Comparing players from different eras can be tricky since there have been varying levels of difficulty and league tendencies over time. OPS is a statistic that encompasses a hitter's value into a single number. This is helpful for comparing players who played in similar time periods but does not account for different time periods. OPS+, however, takes a player's OPS and compares it with the performance of other players who played at the same time (MLB). For example, Pete Browning and Aaron Judge have extremely different OPS values, but the same OPS+ values since Browning played at a time when there was less offensive production. OPS+ is included in the dataset to make the comparison of players who played in different eras fairer.

| Player | From | To | OPS | OPS+ |
|---|---|---|---|---|
| Pete Browning | 1882 | 1894 | .869 | 163 |
| Aaron Judge | 2016 | 2022 | .977 | 163 |

## Defensive Statistics

In baseball, defense is difficult to evaluate. There are many statistics used to represent defensive value, but each one has its own shortcomings, and none are ideal (Malinowski, 2020). Baseball Reference has a complicated formula to compute a defensive statistic called dWAR (Baseball Reference). dWAR compares the defensive efficiency of a player to the league average and outputs a single number. A dWAR of 0 indicates a perfectly average defender. It was decided to include this statistic in the data set since it has high praise from baseball statisticians. It was necessary to include dWAR since quality of defense is heavily judged when a player is considered for the Hall of Fame.

## Data Pruning

This data set required considerable pruning to minimize noise that could affect the results of the statistical methods. The first pruning done was eliminating players with too small of a sample size. This was done by pruning out any player who had under 2,500 plate appearances. This threshold was used because the vast majority of Hall of Fame players were over this mark. The Hall of Fame players that were not over

this mark were almost always inducted for their off-the-field achievements, which of course has no statistical representation.

In the old days of professional baseball (before the 1950's), many pitchers would play other positions and hit much more frequently than they do in the modern game. As a result, the dataset was filled with pitchers who racked up over 2,500 plate appearances but were generally inducted in the Hall of Fame for their pitching prowess rather than their hitting. To combat this, any player who spent over 40% of their games as a pitcher rather than a position player was pruned out. This threshold was used because it kept "hitters who pitched" like Babe Ruth in and excluded "pitchers who hit" that would skew the data set.

### Class Imbalance

Class Imbalance became a moderate concern during the creation of the dataset. According to machine learning expert Jason Brownlee, class imbalance becomes a problem when the minority class is out represented by more than 1:100 (Brownlee, 2020). Since the number of Hall of Fame players is incredibly small compared to the number of players not in the Hall of Fame, this became a potential issue. To steer clear of this potential problem, the number of players not in the Hall of Fame was limited in the dataset. The decision was made to use 2,000 of these players so the class imbalance ratio would be about 1:10. When choosing which of these players to include/exclude, the decision was made to use the top 2,000 OPS+ players. OPS+ is a general hitting statistic that is best for representing a hitter's value in a single number. As a result, the lowest OPS+ hitter not in the Hall of Fame included in the dataset ended up being around the same as the lowest OPS+ Hall of Fame player.

## Classifier Modeling

To create the three classifiers, the Scikit-learn Python library, which is one of the most popular libraries for implementing machine learning algorithms such as KNN, SVM and Perceptron, was utilized.

### Data Importation

The dataset was imported into a Python file using the Pandas library where it was then adjusted in preparation for importation into the classifiers. The Player Name variable was dropped since it should not have an impact on the classifiers, and the classification variable was separated into its own data frame for later use. The dataset was then split into training and testing data using the "train_test_split" function. According to machine learning expert Ajitesh Kumar, an 80:20 ratio of training to testing data is common and effective for machine learning algorithms, so this ratio was used accordingly (Kumar, 2021). This split created four variables: The training features, testing features, training classifications, and testing classifications.

| | PA | dWAR | R | H | 1B | 2B | 3B | HR | RBI | SB | BB | BA | OBP | SLG | OPS | OPS+ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2511 | 1.6 | 610 | 806 | 448 | 135 | 58 | 165 | 730 | 40 | 331 | 0.374 | 0.458 | 0.720 | 1.178 | 214 |
| 1 | 10626 | -2.3 | 2174 | 2873 | 1517 | 506 | 136 | 714 | 2214 | 123 | 2062 | 0.342 | 0.474 | 0.690 | 1.164 | 206 |
| 2 | 9792 | -13.3 | 1798 | 2654 | 1537 | 525 | 71 | 521 | 1839 | 24 | 2021 | 0.344 | 0.482 | 0.634 | 1.116 | 191 |
| 3 | 3962 | -1.2 | 862 | 1218 | 750 | 242 | 80 | 146 | 861 | 209 | 484 | 0.364 | 0.448 | 0.614 | 1.062 | 184 |
| 4 | 2556 | -1.4 | 538 | 728 | 448 | 138 | 47 | 95 | 545 | 34 | 396 | 0.345 | 0.452 | 0.590 | 1.042 | 180 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2171 | 2773 | 4.6 | 300 | 671 | 520 | 103 | 12 | 36 | 261 | 28 | 124 | 0.261 | 0.297 | 0.352 | 0.650 | 71 |
| 2172 | 4438 | 10.3 | 557 | 916 | 753 | 113 | 27 | 23 | 279 | 343 | 478 | 0.237 | 0.321 | 0.299 | 0.620 | 71 |
| 2173 | 3640 | 3.9 | 326 | 788 | 697 | 75 | 14 | 2 | 297 | 19 | 402 | 0.249 | 0.335 | 0.283 | 0.619 | 71 |
| 2174 | 3354 | 0.6 | 294 | 781 | 626 | 113 | 23 | 19 | 348 | 17 | 219 | 0.253 | 0.306 | 0.324 | 0.630 | 71 |
| 2175 | 2719 | -0.6 | 244 | 609 | 498 | 68 | 28 | 15 | 137 | 18 | 208 | 0.249 | 0.308 | 0.318 | 0.626 | 71 |

[2176 rows x 16 columns]

*Pandas DataFrame of the imported dataset.*

## Hyperparameter Adjustments

Hyperparameters are used to control the learning process of machine learning models. To optimize the classifiers, the best hyperparameter values needed to be found. For KNN classifiers, the most influential hyperparameter is the number of nearest neighbors (k). To find this optimal value, the accuracy results of KNN classifiers using different k values on the same sample of the data set were compared with one-another. After numerous iterations for consistency, a k-value of 7 was deemed optimal since it yielded the most accurate results without a heavy impact on the runtime.

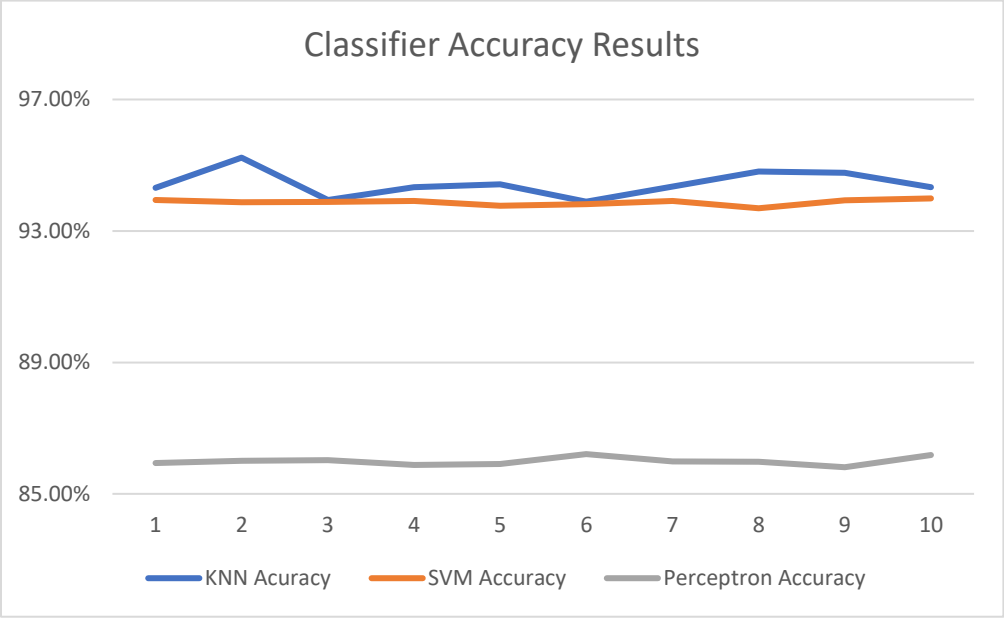
```
1 Clusters accuracy score: 0.944954128440367
2 Clusters accuracy score: 0.9380733944954128
3 Clusters accuracy score: 0.944954128440367
4 Clusters accuracy score: 0.9334862385321101
5 Clusters accuracy score: 0.9380733944954128
6 Clusters accuracy score: 0.9426605504587156
7 Clusters accuracy score: 0.9472477064220184
8 Clusters accuracy score: 0.944954128440367
9 Clusters accuracy score: 0.9472477064220184
10 Clusters accuracy score: 0.944954128440367
```

*The accuracy scores of different k-value classifiers.*

A similar process was used to adjust the learning rate for the Perceptron classifier. However, this hyperparameter had less of an effect on the accuracy results, so the default value (1) was used.

## Testing the Classifiers

Each of the classifiers was fitted with a sample of the training data and then was tasked to predict a sample of the testing data. The results of the predictions were then compared with the actual classifications of the testing data. This process was performed 100 times and the final accuracy result of the iteration was noted.




*Accuracy results of each of the classifiers across 10 iterations.*

# Conclusion

After analyzing the performance of each of the classifiers, KNN proved to be the best suited for this work based on how accurate it could classify unclassified data. SVM was a close second in accuracy results, but clearly was always a step behind KNN. Lastly, perceptron was a distant third which opens the question of why this may be the case?

## Understanding the Perceptron Classifier's Low-Accuracy Results

When trained and tested with the same data, Perceptron frequently performed worse than SVM (Kharel, 2020). A possible explanation for this disparity is the SVM algorithm using a technique called the kernel trick (Zvornicanin, 2022). The main idea of the kernel trick is that whenever the classes are not able to be separated in the current dimension, another dimension is added where the classes may be separable. When this technique is utilized, the data is transformed into a higher dimension which leads to higher accuracy results when the data is not easily separable. Like KNN, this leaves SVM susceptible to the curse of dimensionality, while perceptron does not face this. However, because this study includes a relatively non-complex data set, KNN and SVM did not face any challenges with over-complexity.

## False Classifications Analysis

After collecting accuracy results, an analysis was performed to discover why none of the classifiers could achieve perfect accuracy scores.

### Subjective Classification

The Hall of Fame voting process is subjective. According to the Baseball Writes' Association of America, "Voting shall be based upon the player's record, playing ability, integrity, sportsmanship, and contributions to the team(s) on which the player played" (BBWAA, 2022). All of these are subjective qualities which cannot be statistically represented in a data set. Because of this shortcoming, the classifiers frequently mislabeled many players who were on the cusp of making or missing the Hall of Fame.

### Players with Off-the-field Issues

Each of the classifiers predicted Barry Bonds to be a Hall of Famer 100% of the time. Bonds was never voted into the Hall of Fame due to steroid use, and he is an example of many players who would be Hall of Famers if not for off-the-field issues. Ideally, these players would have been pruned out of the data set before training and testing began, but it would have been unrealistic to research every player's history with breaking league conduct. Instead, these players became a limitation in the study and simply unfiltered noise in the data.

# References


Baseball Reference. *Position player war calculations and details*. Baseball. (n.d.). Retrieved November 20, 2022, from https://www.baseball-reference.com/about/war_explained_position.shtml

Baseball Writers' Association of America. (2022, December 5). Hall of Fame Election Requirements. BBWAA. Retrieved December 14, 2022, from https://bbwaa.com/hof-elec-req/#:~:text=Method%20of%20Election&text=An%20elector%20will%20vote%20for,National%20Baseball%20Hall%20of%20Fame.

Brownlee, J. (2020, January 14). *A gentle introduction to imbalanced classification*. MachineLearningMastery.com. Retrieved November 20, 2022, from https://machinelearningmastery.com/what-is-imbalanced-classification/

Edes, G. (2013, October 2). *Jacoby Ellsbury: Sultan of swipe*. ESPN. Retrieved November 20, 2022, from https://www.espn.com/boston/mlb/story/_/id/9757214/jacoby-ellsbury-quick-learner-craft-stealing-bases

Kharel, S. (2020, May 13). Perceptron vs SVM: A quick comparison. Medium. Retrieved December 14, 2022, from https://medium.com/@subashkharel/perceptron-vs-svm-a-quick-comparison-6b5d6b5d64f

Kumar, A. (2021, June 13). Machine learning - Training, Validation & Test Data Set. Data Analytics. Retrieved December 14, 2022, from https://vitalflux.com/machine-learning-training-validation-test-data-set/

Malinowski, I. (2020, April 15). *What is the best statistic for measuring defense?* DRaysBay. Retrieved November 20, 2022, from https://www.draysbay.com/2020/4/15/21219514/mlb-defense-best-statistic-baseball-prospectus

MLB. On-base plus slugging plus (OPS+): Glossary. MLB.com. (n.d.). Retrieved November 20, 2022, from https://www.mlb.com/glossary/advanced-stats/on-base-plus-slugging-plus

Nyuytiymbiy, K. (2022, March 28). Parameters and hyperparameters in machine learning and Deep Learning. Medium. Retrieved December 14, 2022, from https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac

Stueve, W. (2017, September 30). *What are the trademarks of a NFL hall of fame player?* Bleacher Report. Retrieved November 20, 2022, from https://bleacherreport.com/articles/1284262-what-are-the-trademarks-of-a-nfl-hall-of-fame-player#:~:text=There%20isn't%20just%20one,by%20position%20and%20total%20value.

Zvornicanin, E. (2022, November 8). Difference between a SVM and a Perceptron. Baeldung on Computer Science. Retrieved December 14, 2022, from https://www.baeldung.com/cs/svm-vs-perceptron