Brian McCabe

Baseball Classification

Purpose

Classify Baseball Players Feilding Positions

- Batting and **Fielding** Statistics

Comparing Classification Algorithms

- K-Nearest Neighbors Classifier
- SGD Classifier
- Gaussian Naïve Bayes
- Decision Tree Classifier
- Random Forest Classifier

BRIAN MCCABE

Significance

What attributes are most important in predicting Player position?

Can organizations make a team around Players predicted positions?

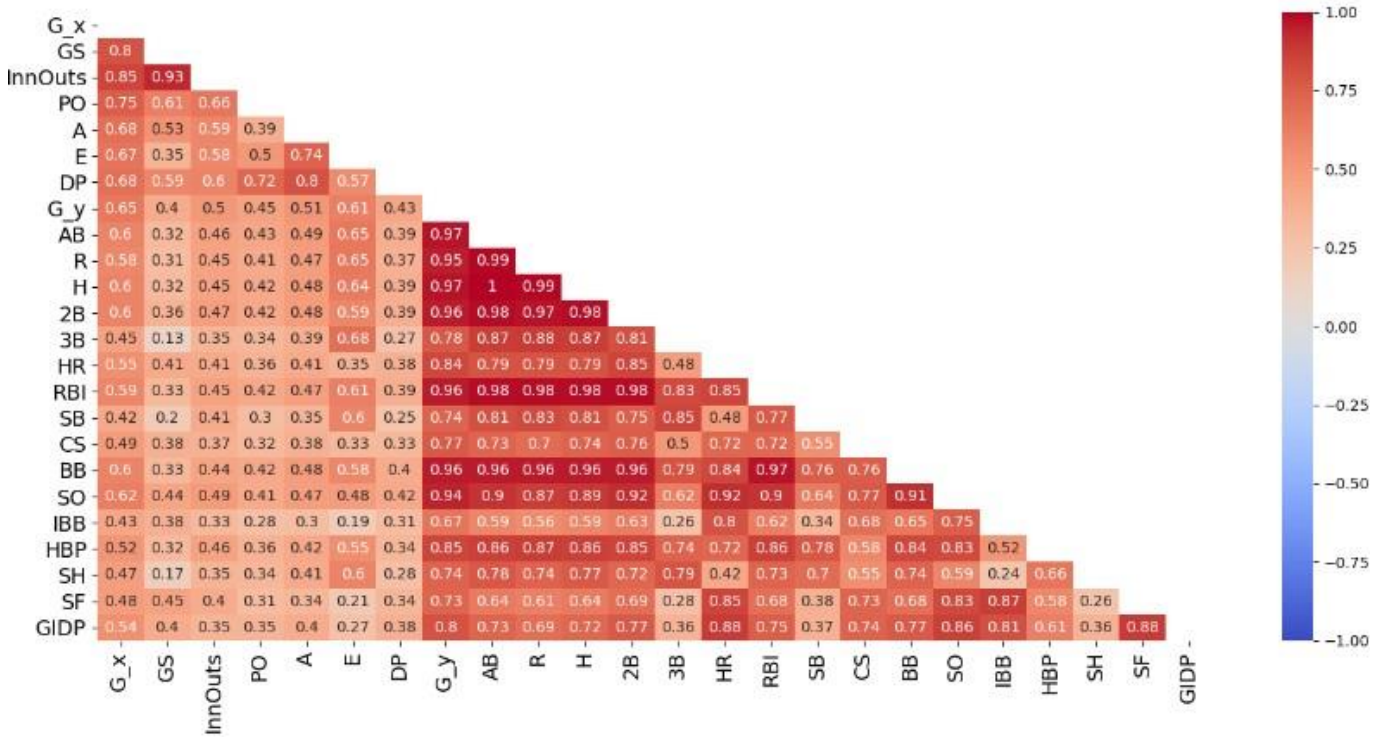
Which position is the hardest to predict why are they difficult to predict?

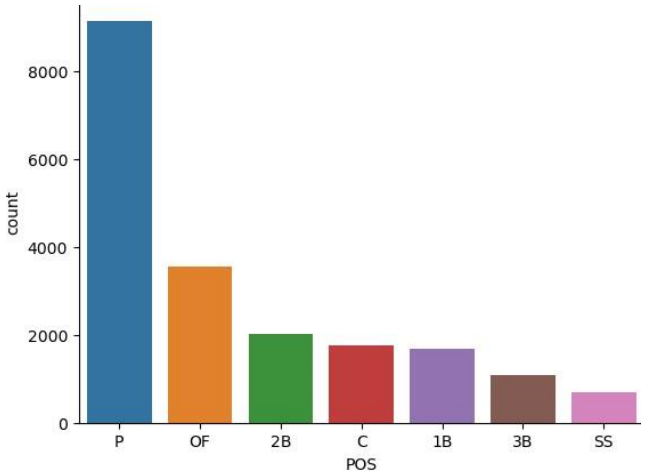
BRIAN MCCABE

Data Description

- 24 input variables
- Batting input variables
  - At Bats
  - Runs
  - Hits
  - Home Runs
  - Runs Batted In
  - Walks
  - Strike Outs
- Fielding input variables
  - Games Started
  - Put Outs
  - Assist
  - Errors
  - Inn Outs
  - Double Plays
- Out-Put variables: Player Position
  - Catcher (C)
  - Pitcher (P)
  - 1st Baseman (1B)
  - 2nd Baseman (2B)
  - 3rd Baseman (3B)
  - Short Stop (SS)
  - Out Fielder (OF)

BRIAN MCCABE

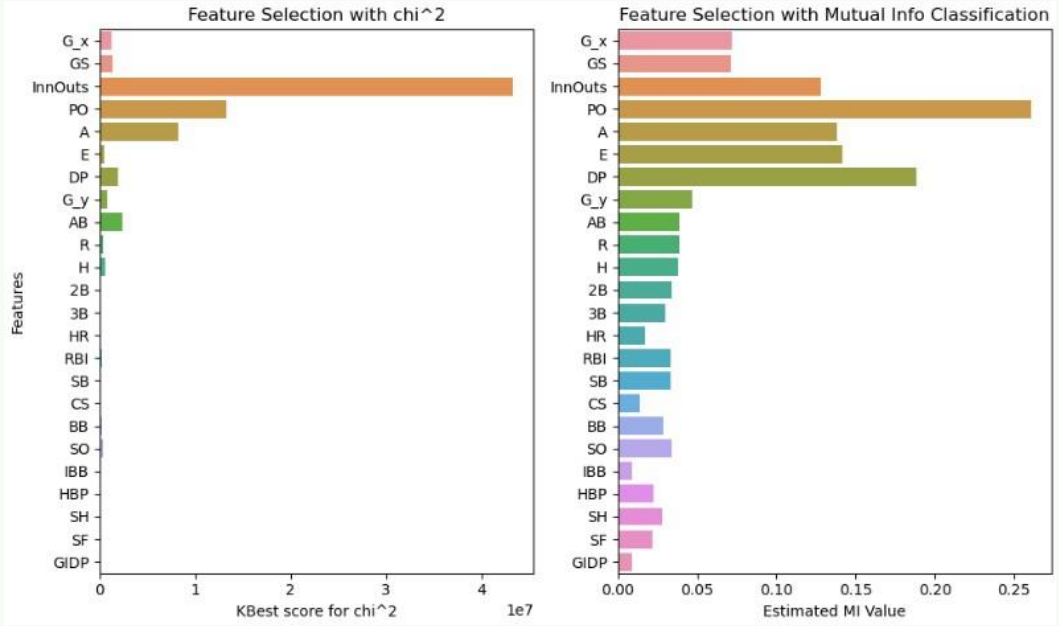


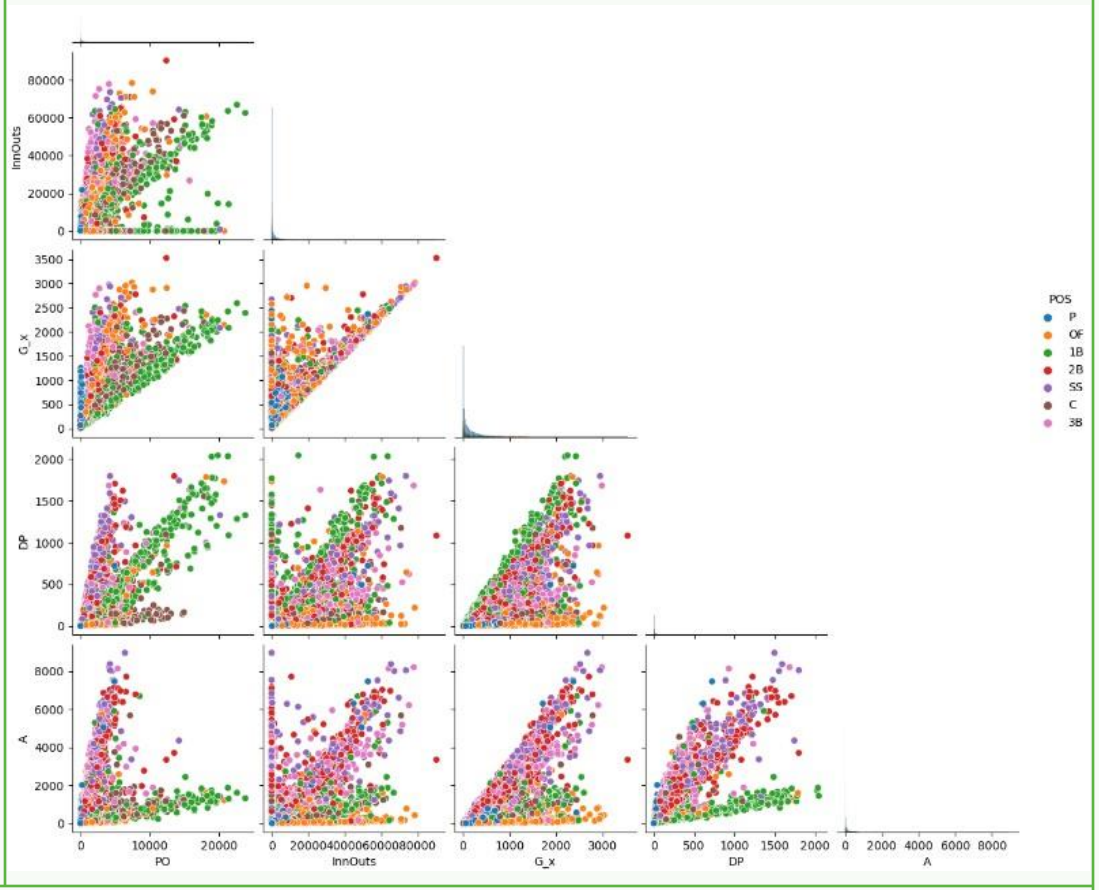


Exploratory

BRIAN MCCABE

Exploratory





Data Preparation

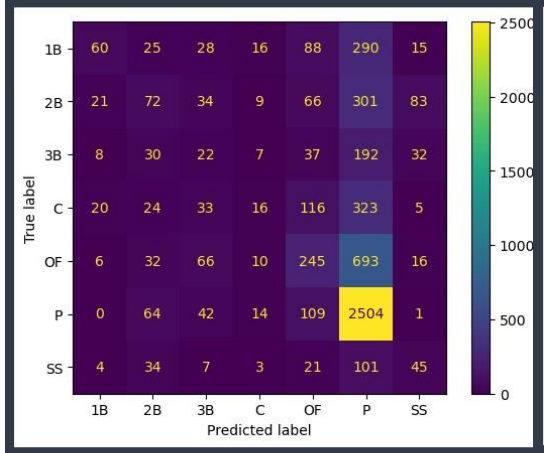
```
aggregation_functions_f = {'POS': 'first', 'G': 'sum','GS': 'sum', 'InnOuts':'sum',
                           'PO':'sum','A':'sum','E':'sum','DP':'sum'}
combined_f = fielding.groupby(fielding['playerID']).aggregate(aggregation_functions_f)
aggregation_functions_b = {'G': 'sum','AB': 'sum', 'R':'sum','H':'sum',
                           '2B':'sum','3B':'sum','HR':'sum','RBI':'sum','SB':'sum',
                           'CS':'sum','BB':'sum','SO':'sum','IBB':'sum','HBP':'sum',
                           'SH':'sum','SF':'sum','GIDP':'sum'}
combined_b = batting.groupby(fielding['playerID']).aggregate(aggregation_functions_b)
merged_df = pd.merge(combined_f,combined_b, on='playerID',how = 'outer')
merged_df = merged_df.fillna(0)
```

- Combined records that had same player ID
- Aggregate Function
- Merged batting and Fielding on player ID
- Filled NANs with Zeros
- 70/30 Train Test Split

```
y = merged_df['POS']
X = merged_df.drop('POS', axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
```

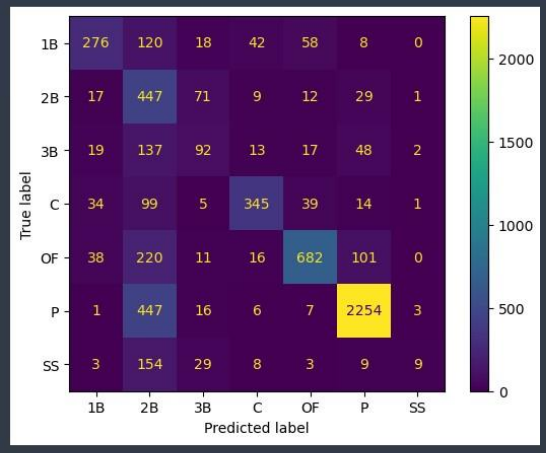
BRIAN MCCABE

model GaussianNB()



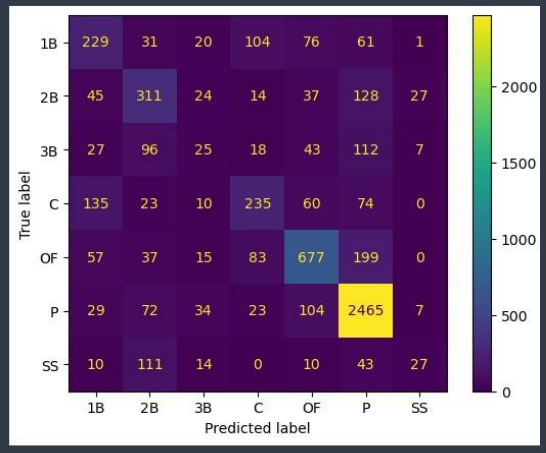
	precision	recall	f1-score	support
1B	0.11	0.50	0.19	119
2B	0.12	0.26	0.17	281
3B	0.07	0.09	0.08	232
C	0.03	0.21	0.05	75
OF	0.23	0.36	0.28	682
P	0.92	0.57	0.70	4404
SS	0.21	0.23	0.22	197
accuracy			0.49	5990
macro avg	0.24	0.32	0.24	5990
weighted avg	0.72	0.49	0.57	5990

model SGDClassifier()



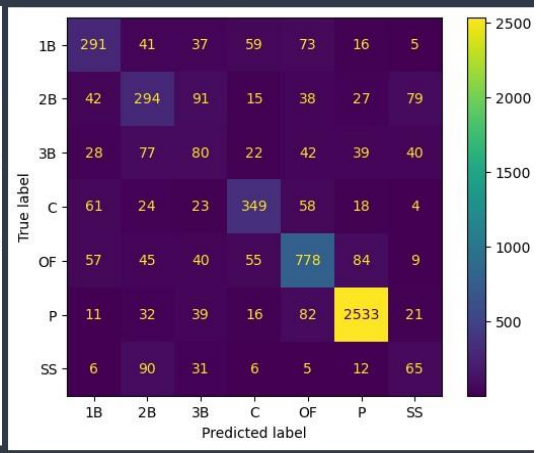
	precision	recall	f1-score	support
1B	0.53	0.71	0.61	388
2B	0.76	0.28	0.40	1624
3B	0.28	0.38	0.32	242
C	0.64	0.79	0.71	439
OF	0.64	0.83	0.72	818
P	0.82	0.92	0.87	2463
SS	0.04	0.56	0.08	16
accuracy			0.69	5990
macro avg	0.53	0.64	0.53	5990
weighted avg	0.73	0.69	0.67	5990

model KNeighborsClassifier(n_n



	precision	recall	f1-score	support
1B	0.44	0.43	0.43	532
2B	0.53	0.46	0.49	681
3B	0.08	0.18	0.11	142
C	0.44	0.49	0.46	477
OF	0.63	0.67	0.65	1007
P	0.90	0.80	0.85	3082
SS	0.13	0.39	0.19	69
accuracy			0.66	5990
macro avg	0.45	0.49	0.46	5990
weighted avg	0.71	0.66	0.68	5990

model DecisionTreeClassifier()

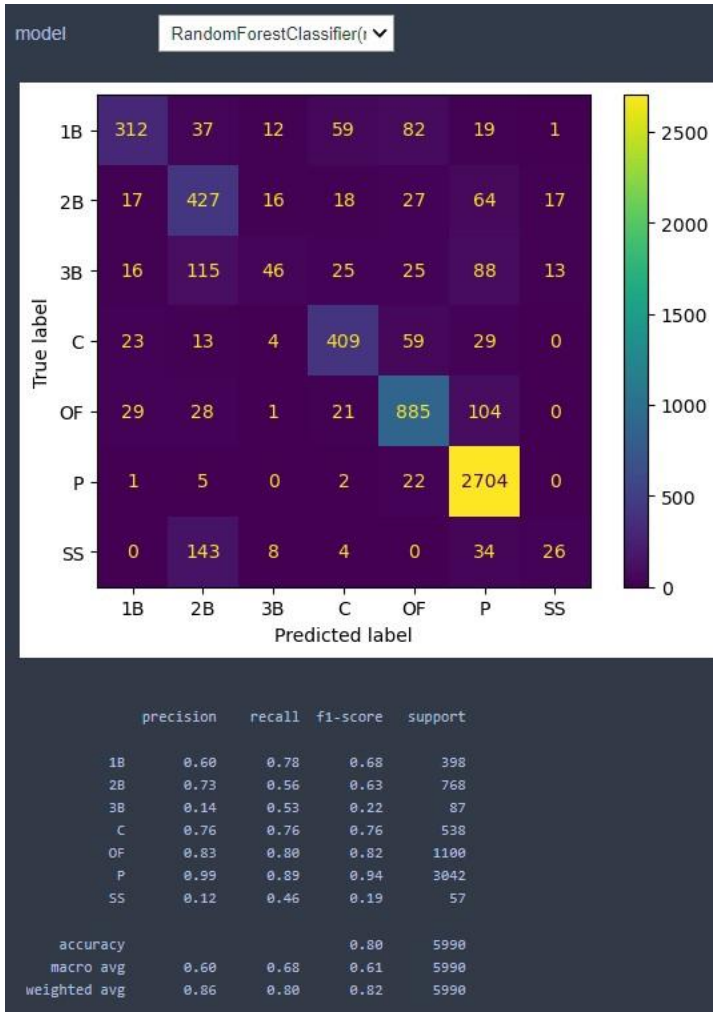


	precision	recall	f1-score	support
1B	0.56	0.59	0.57	496
2B	0.50	0.49	0.49	603
3B	0.24	0.23	0.24	341
C	0.65	0.67	0.66	522
OF	0.73	0.72	0.73	1076
P	0.93	0.93	0.93	2729
SS	0.30	0.29	0.30	223
accuracy			0.73	5990
macro avg	0.56	0.56	0.56	5990
weighted avg	0.73	0.73	0.73	5990

Model Analysis

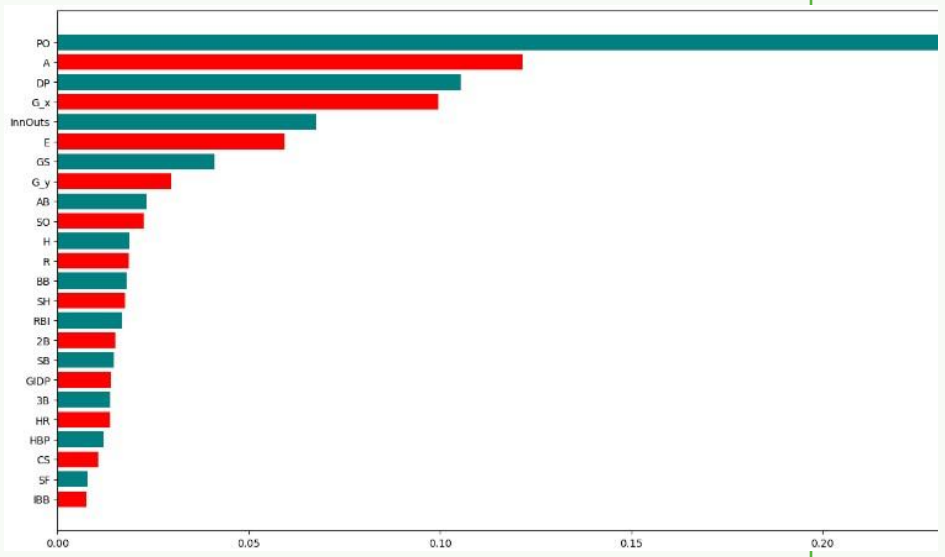
Model	Data	Accuracy	Precision	Recall	F1	Train Acc
GaussianNB()	features_df	0.494825	0.495	0.703	0.495	0.493202
SGDClassifier()	features_df	0.610518	0.611	0.781	0.611	0.610634
KNeighborsClassifier(n_neighbors=4)	features_df	0.662604	0.663	0.814	0.663	0.780163
DecisionTreeClassifier()	features_df	0.729048	0.729	0.854	0.729	0.999356
RandomForestClassifier()	features_df	0.796995	0.797	0.893	0.797	0.999356
RandomForestClassifier(max_features='log2', min_samples_split=5, n_estimators=450, n_jobs=-1, random_state=4)	features_df	0.802838	0.803	0.896	0.803	0.98626

Conclusion



	precision	recall	f1-score	support
1B	0.60	0.78	0.68	398
2B	0.73	0.56	0.63	768
3B	0.14	0.53	0.22	87
C	0.76	0.76	0.76	538
OF	0.83	0.80	0.82	1100
P	0.99	0.89	0.94	3042
SS	0.12	0.46	0.19	57
accuracy			0.80	5990
macro avg	0.60	0.68	0.61	5990
weighted avg	0.86	0.80	0.82	5990

- With 80% accuracy the Random Forest Classifier performed the best
- Feature Importance
  - Put outs
  - Assist
  - Double Plays
  - Games
  - Inn Outs



- Hardest Positions to predict
  - Short Stop
  - 3rd Baseman

Questions

Thank You

BRIAN MCCABE