Team Data Boys
Project Report

CS 105 - Project Report

CS 105
March 11, 2020

Summary

After starting our project and reading in our data set, we were not sure on exactly what we wanted to try to predict using ML. Once we read in the data and thought about it, we decided to go with predicting a team's season-long number of homeruns, since this is probably one of the most exciting stats surrounding the sport of baseball. After reading in our data, along with homeruns, we had 28 other features and over 3000 rows spanning years 1980 - 2019.

Phase 1 and 2 - scraping data and cleaning

The beginning of our phase 1 started with grabbing our data from a very good baseball data website using HTML parsing with BeautifulSoup. Example Code:

```

#scrape data into BS object and append into large dataframe
res0 = requests.get("https://www.baseball-reference.com/leagues/MLB/2019-standard-batting.shtml")
soup0 = BeautifulSoup(res0.content,'lxml')
table = soup0.find_all('table')[0]
df1 = pd.read_html(str(table))[0]
df = df.append(df1, ignore_index=True)
```

Website: https://www.baseball-reference.com/leagues/MLB/2019-standard-batting.shtml

After scraping all required data into a DF, the next step was cleaning.

We checked for missing data and found many missing rows due to placeholders from the website. We then located all and deleted them.

```
[4] df.loc[df["Tm"] == "Tm"]
```

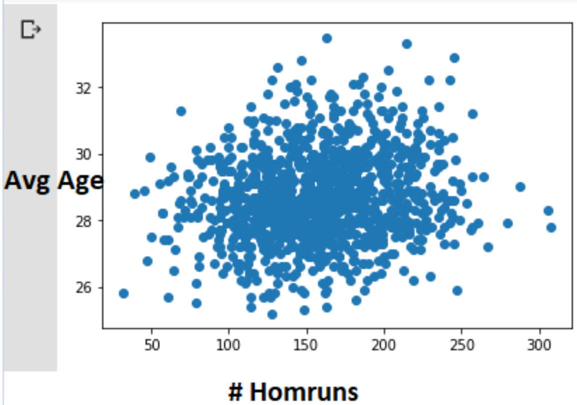
	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB
31	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB
64	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB
97	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB
130	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB
163	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1140	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB
1169	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB
1198	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB
1227	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB
1256	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB

40 rows × 29 columns

Once this was complete was double-checked for missing data, proceeded to find no more missing data.

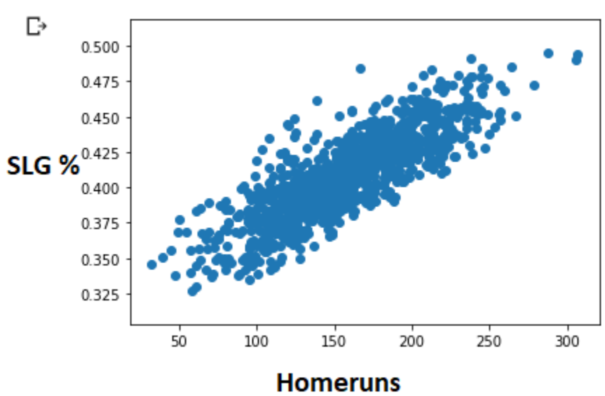
The next cleaning step was to remove any irrelevant features.
We found the the average age was not correlated with Homeruns.

```
data = df[['HR', 'BatAge']]
data = data.astype(float)
x = data['HR']
y = data['BatAge']
plt.scatter(x, y)
plt.show()
```



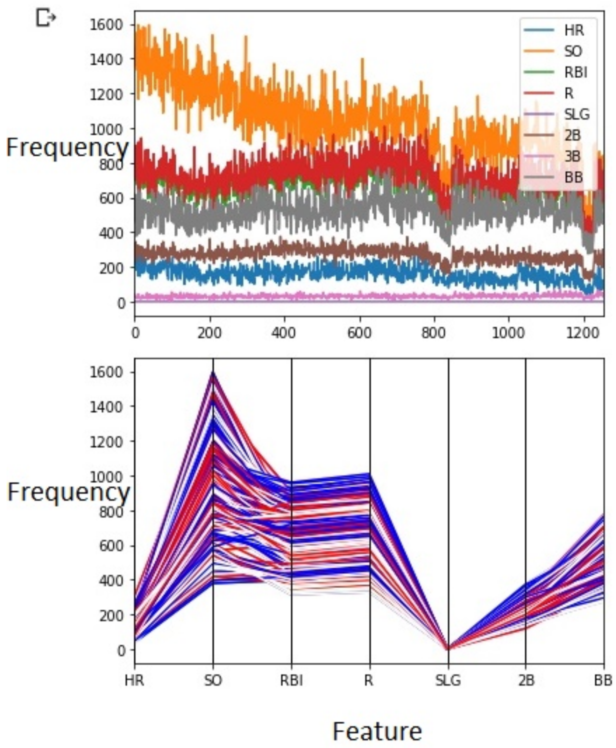
Since there was no correlation we removed this feature.

We then tested and found features that had a strong correlation with homeruns for training purposes

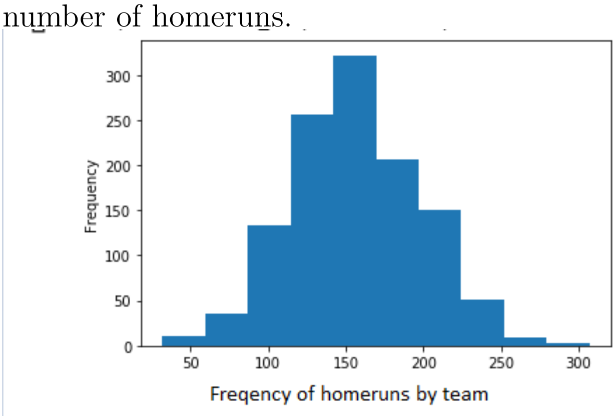


In summary, the features selected to keep for the best training of our data were [homeruns, strikeouts, RBIs, runs, slugging percentage, doubles, triples, and walks]

The following graphs show the frequency distribution of our training data.



We plotted a histogram to show us the distribution of homeruns throughout the entire dataframe so we could get an idea of what ranged to expect when we try to predict the number of homeruns.



from this we knew that most teams never hit over 300 Hrs and we could likely expect our range to be in the 100-200 range based off past data.

Phase 3

Before training/using some Machine Learning models, the questions we would like to know are:

1) How accurately can a ML model predict a real score with the data we have scraped thus far? Potentially trivial question, but I am curious to see how accurate this will be.

2) If we train our data with seemingly less relevant features, how far off will the prediciton be from our initial selected features?

We used 2 different models to predict our results after training the data.
To test our prediction, we selected one teams season data. In this particular season, The Milwaukee Brewers hit 127 homeruns.

```
[ ] df.iloc[1001:1002].HR
```

```
1067    127
Name: HR, dtype: object
```

```
[ ] df.iloc[1001:1002]
```

	Tm	#Bat	BatAge	R/G	G	PA	AB	R	H	2B	...	SLG	OPS	OPS+	TB	GDP	HBP	SH	SF	IBB	LOB
1067	MIL	39	29.2	4.14	161	6126	5461	667	1393	255	...	.385	.707	90	2105	122	27	53	53	26	1143

1 rows × 29 columns

1. Linear regression: Our linreg model was the most accurate prediction with 131 homeruns.

```
xdata = df.iloc[1:1000]
xtest = df.iloc[1001:1002]

X_train = xdata[['2B','RBI','3B']]
X_test = xtest[['2B','RBI','3B']]
y_train = xdata['HR']

model = LinearRegression()
model.fit(X=X_train, y=y_train)
model.predict(X=X_test)
```

```
array([131.52468648])
```

2. Cross Validation: While still very accurate, it was slightly worse then linreg by guessing 144.

```
[ ] from sklearn.model_selection import cross_val_predict
    from sklearn import datasets, linear_model

    train_data = df.iloc[1:1010]

    X = train_data[['2B','RBI','3B']]
    y = train_data['HR']
    lasso = linear_model.Lasso()
    y_pred = cross_val_predict(lasso, X, y, cv=3)
    y_pred[1000:1001]
```

```
array([144.85705651])
```