Abstract

The goal of this project is to predict the salary of a given player based on their statistics for that year. I scraped data from baseball-reference.com and built a linear regression model with the information that I scraped. Once I refine the model, I have built visualizations of the interactions between variables as well as which model performed the best.

Design

The design for this project is that it is a tool that will be used by the owner of a baseball team to check the performance of his general manager. By being able to calculate a predicted value for salary, the owner will be able to see which players are overpaid and which are underpaid each season. This will allow them to make better decisions when negotiating contracts with their current players and will allow them to know how much to pay for future players.

Data

Each row of the dataset represents a year of an individual player. I am using every season played by all players on an active MLB roster. The dataset contains 4,163 rows and 62 features. All but two of the features are numeric and many features are collinear. A few examples of the features include games played, runs batted in, and wins above replacement.

Algorithms

Feature Engineering

1. Creating a column that represents if a player has played at multiple positions in the season by breaking down the position column
2. Creating one-hot-encoders for each postseason award by using a RegEx technique to identify each name in the award column
3. Adding polynomial features to see the interaction terms between the features

Models

The baseline for this project is a basic linear regression model. I've also tested a Lasso and Ridge model, and then I created polynomial features and used a lasso and ridge model on those as well. The polynomial features improved performance and combining that with LassoCV gave me the best results.

Model Evaluation and Selection

To start, I split the dataset into a 75/25 split where I would train the data on the 75% and use the remaining 25% for testing. I did not use the test data until the very end and decided which model had the best performance by doing 5-fold cross validation with each model.

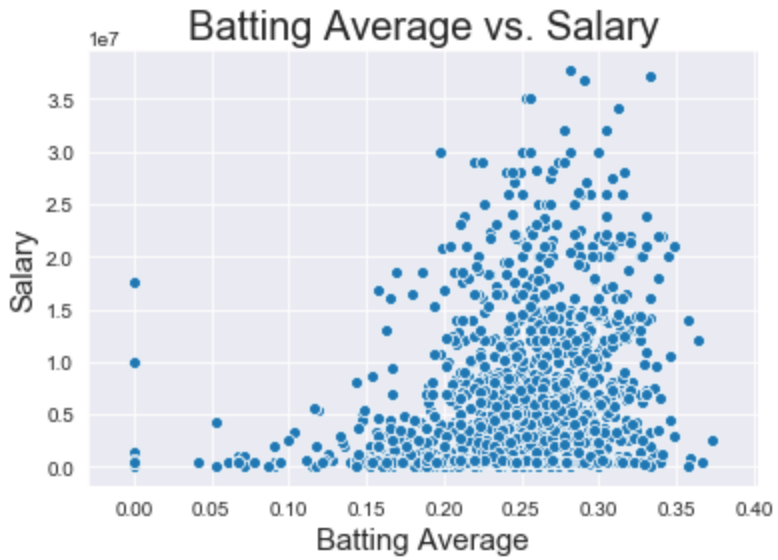
The metric that I used was R^2. This will show the improvement of using the model instead of guessing the mean. The R^2 on the test data of the best model was 0.56.

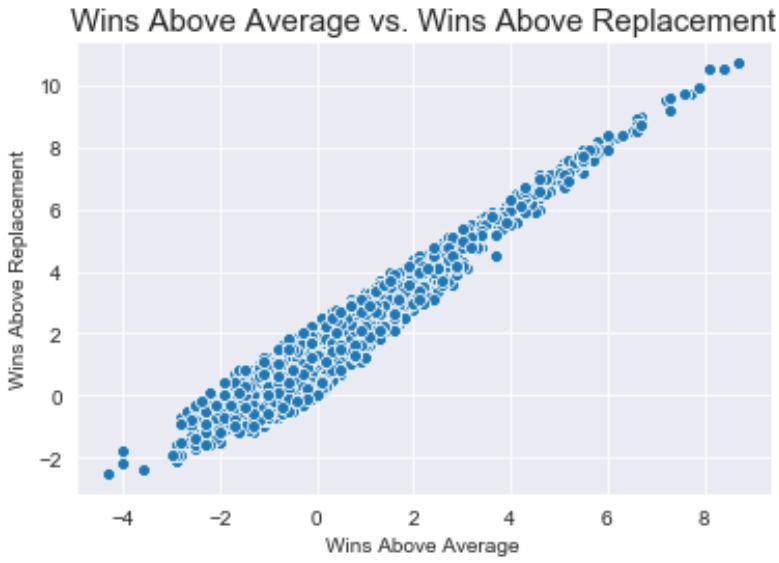
Tools

- Numpy and Pandas for data manipulation
- Scikit-learn for modeling
- Matplotlib and Seaborn for plotting
- BeautifulSoup for WebScraping

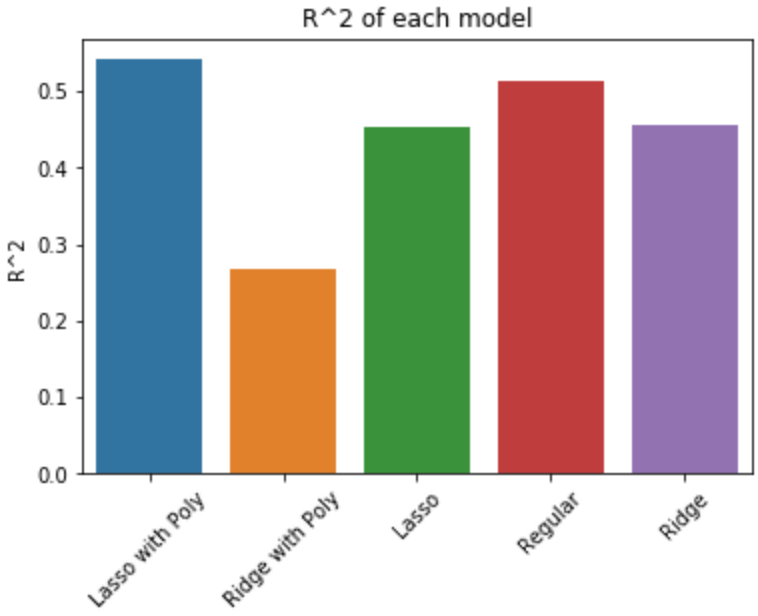
Communication

Here are the visualizations that I will be presenting.


Batting Average vs. Salary
1e7
Salary
Batting Average
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40


Wins Above Average vs. Wins Above Replacement
Wins Above Replacement
Wins Above Average
−2
0
2
4
6
8
10
−4
−2
0
2
4
6
8


Histogram of Salaries
Count
Salary
1e7
0
200
400
600
800
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5


R^2 of each model
R^2
0.0
0.1
0.2
0.3
0.4
0.5
Lasso with Poly
Ridge with Poly
Lasso
Regular
Ridge