Project Luther Write-Up

Problem Statement: To predict percentage of the vote received a Hall of Famer receives on the first year they are on the Hall of Fame ballot. A player needs to receive 75% of the vote to be inducted and can appear for up to 10 years on the ballot. Writers can vote for up to 10 players on a ballot

Data Source: Baseball-reference.com has well organized tabular data. I put together a python file with functions for the downloading and cleaning. Definitely need to do something similar for rest of notebook.

Exploratory Analysis

The percentage vote received is positively skewed. Most players that appear on the ballot receive no or few votes. Based on the appearance of my seaborn pair plots, I'm going to transform my %vote with the inverse of the logistic sigmoid (see below) and run some regressions with the transformed vote% and regular vote%. I am incrementing each vote by a little bit as I have 100+ records which do not have any votes and wanted to keep that data, rather than throwing it out with the transform.

```
#665 records total when including with 0 votes
#546 records when not including 0 votes
#Create a new column with %vote incremented by constant

increment_to_vote = 0.2
hof_df['%vote_shift'] = hof_df['%vote'] + increment_to_vote
hof_df['%vote_shift_transformed'] = - np.log((100.2-hof_df['%vote_shift']) / hof_df['%vote_shift'])
```

Next, I ran a correlation matrix on all the variables. There are very high correlations (0.5 - 0.9). This makes sense as some variables directly flow into the computation of others (ie lots of home runs contribute to a high slugging percentage).

What I do like for a reduced combination of variables based on the correlations is the correlation between home runs and batting average is quite low, and those are historically the two "headline stats" for a hitter. Combine that with the stolen base category in a simple regression, would be interesting to see if that can capture a fair amount of explanatory power.

Simple OLS

I did four simple OLS regressions: The full data set on both the transformed and untransformed vote% and just the batting average, home runs and stolen bases on those same two targets.

With the full dataset, can see one hallmark of multi-collinearity as there is a large negative statistically significant coefficient on slugging percentage, which makes no sense that a higher slugging percentage would lower the vote received.

On all regressions, just less so when using the transformed target, can see straight line down in residuals due to skew and then drift up in residuals at the higher vote percentage targets. Hope a polynomial form may mitigate latter drift up in residuals.

Cross-Validation

Try linear regression, ridge and both second and third degree polynomial with full feature set on the transformed vote percentage

```
Simple regression scores:  [0.59355153597351151, 0.52738455539431861, 0.39115106256252929, 0.31680170936078234, 0.588
34514136122729, 0.50117726969779541, 0.28430717370833847, 0.40564457359591799, 0.54711367277431289, 0.250262366329785
49]
Ridge scores:  [0.5926771504871321, 0.5296156339470387, 0.39042342505393285, 0.31811289248363694, 0.5866349324561201
9, 0.50165274061328136, 0.28275932856036923, 0.40540532867777146, 0.54958587509165224, 0.25435134800347881]

Poly2 scores:  [0.75569718870238645, 0.63064456853736695, 0.56274976752980477, -0.10395670525713552, -4.7306727486515
268, 0.61723906234528525, -0.35738599265584331, 0.51223319943685564, 0.42327433413235749, 0.28985956353190701]

Poly3 scores:  [-0.28513992977635083, -3634730.6712668194, -50455967.568011463, -0.047463261205429985, 0.411213658543
65077, 0.24291375850636157, -349428.2647246886, 0.093715860257472361, -871.98699463288233, -1.9922441836965827]

Simple mean cv r^2: 0.441 +- 0.121
Ridge mean cv r^2: 0.441 +- 0.121
Poly2 mean cv r^2: -0.140 +- 1.566
Poly3 mean cv r^2: -5444100.007 +- 15042459.182
```

Second degree polynomial had good performance on some of the folds. Try reduced feature set to see if helps with overfitting.

```
Simple regression scores:  [0.31029745944803211, 0.33708434074824511, 0.26887998061259866, 0.42318503480519271, 0.508
03657755668352, 0.36268309352513861, 0.31613874169706557, 0.24911142204637182, 0.4563810578098641, 0.3273259142006390
6]

Ridge scores:  [0.31142791018998406, 0.33716098540478945, 0.26989249094000012, 0.42274230563375503, 0.507825044917679
1, 0.36224954965405476, 0.31602575482000528, 0.25113645753338032, 0.45555717939059265, 0.32715767388933814]

Poly2 scores:  [0.39375820577213583, 0.4923728913400171, 0.36443182036711058, 0.48142862568657724, 0.6284496999970097
3, 0.33533073223877385, 0.35232519567296977, 0.39925462553103458, 0.56513856781893379, 0.44449130210949794]

Poly2 only interaction  scores:  [-0.28513992977635083, -3634730.6712668194, -50455967.568011463, -0.0474632612054299
85, 0.41121365854365077, 0.24291375850636157, -349428.2647246886, 0.093715860257472361, -871.98699463288233, -1.99224
41836965827]

Simple mean cv r^2: 0.356 +- 0.079
Ridge mean cv r^2: 0.356 +- 0.078
Poly2 mean cv r^2: 0.446 +- 0.091
Poly2 only interaction mean cv r^2: 0.384 +- 0.091
```

Create holdout set (can see that probably should have done this earlier) and tune regularization parameter for both polynomial (added regularization) and linear model. So workflow will be as follows:

Cross - Validation with 10-folds on 80% to get the best regularization value on alpha for both Linear and Poly2.

Reshuffle the 80%, then perform cross-validation with 10-folds and see which model has the best the performance at its chosen alpha value from previous step.

Select best model, and retrain on the full 80%.

Next test on the held out 20% to see performance

I went over results from this in presentation, only notable is transforming the target back into RMSE into original terms to get an interpretable average error of 19 percentage points.

Attempt with Reduced Dataset

I repeated this workflow with a reduced dataset. I dropped players who received less than 5% of the vote. Those players get dropped from the ballot in any event. Data is still skewed, but less so. Dataset was too sparse to get meaningful results

Lasso Path using LARS algorithm

I subset the data for pre/post the year 2000. I wanted to see if the advanced statistic WAR was chosen at a different time by the LARS process when using LASSO. WAR was chosen first in both datasets.