Machine Learning Final Writeup

Looking at Data

Upon first looking at the data, a couple of things struck out to me. First, there's a big outlier of -9999 for GRENG's caretaker value, which I edited to be 0. Second, there is a lot of missingness in the coalition_total and left_righty columns. Removing all missing data would have resulted in too few data points and imputing would have resulted in it potentially overfitting, so I dropped those columns.

After this, there were still a couple missing values, so I did KNN imputation using the code below.

```
k_neighbors = 2

# Create KNN imputer
knn_imputer = KNNImputer(n_neighbors=k_neighbors)

# Perform imputation
df_knn = knn_imputer.fit_transform(df)

# Convert the NumPy array back to a DataFrame
df = pd.DataFrame(df_knn, columns=df.columns)
```

Then, I found outliers in the dependent variable using the 1.5 IQR method. The code found 66 outliers out of a total of 657 data points, which seemed to be too many to be considered outliers. Understanding that the proportion of cabinet ministries in each party was heavily zero-inflated, I decided against removed the "Outliers".

To understand and visualize the relationships between the independent variables, I looked at a correlation matrix, covariance heatmap, plots of each independent variable against cabinet_proportion, and VIF table.

Baseline Models

Before removing any variables, I ran a lasso with k-fold cross validation and decision tree to get a baseline model for what I'd get without touching the data. The results of

lasso are shown below:

Optimal alpha: 0.0001747528400007683
Equation: 0.1294 + -0.0082 * seats + -0.0008 * sq_cabinet + 0.0047 * sq_pm + -0.0001 * base + -0.0134 * miw_new + 0.0204 * banzhaf + 0.0000 * shapley + 0.0072 * splus + -0.0009 * cabinet_id + -0.0006 * party_id + 0.0006 * caretaker + 0.0307 * cabinet_party + 0.0027 * prime_minister + 0.0004 * left_rightx + 0.1710 * cabinet_seats + -0.0217 * total_cabinet_size + 0.0116 * party_count + -0.0051 * cab_count + 0.0000 * country_id + 0.0002 * election_id + 0.0169 * seats_share + -0.0014 * post_election + 0.0000 * enpp + 0.0018 * mingov + 0.0023 * bicameral + -0.0062 * largest_parl + 0.0155 * largest_cab + -0.0050 * lag_largest_parl + -0.0003 * lag_largest_cab + 0.0050 * seats_total + -0.0097 * miw_proportion + 0.0000 * cabinet_proportion + 0.0031 * seats_proportion + 0.0000 * W + -0.0000 * A + 0.0018 * B + 0.0003 * B_star + -0.0001 * C + 0.0000 * D + 0.0041 * E + 0.0000 * country_dummy1 + -0.0000 * country_dummy2 + -0.0001 * country_dummy3 + -0.0012 * country_dummy4 + 0.0000 * country_dummy5 + 0.0064 * country_dummy6 + 0.0032 * country_dummy7 + 0.0000 * country_dummy8 + -0.0000 * country_dummy9 + -0.0000 * country_dummy10 + 0.0025 * country_dummy11 + -0.0007 * country_dummy12
Mean R-squared: 0.9589803297480562
Training MSE: 0.0016402386011704875
Training R-squared: 0.9679995390941044
Test MSE: 0.0013678397061940599
Test R-squared: 0.9733761336749791

Despite the lasso model having a good performance with a test r-squared of .97, this is obviously not interpretable because of the number of variables involved in the equation. However, it did give me a sense for which variables are most and least important so I could do feature selection later. The variables with the highest absolute coefficients, like banzhaf, cabinet_party, and cabinet_seats were more important while all the country dummy variables were unimportant, as their coefficients were essentially 0, even with such a small alpha value.

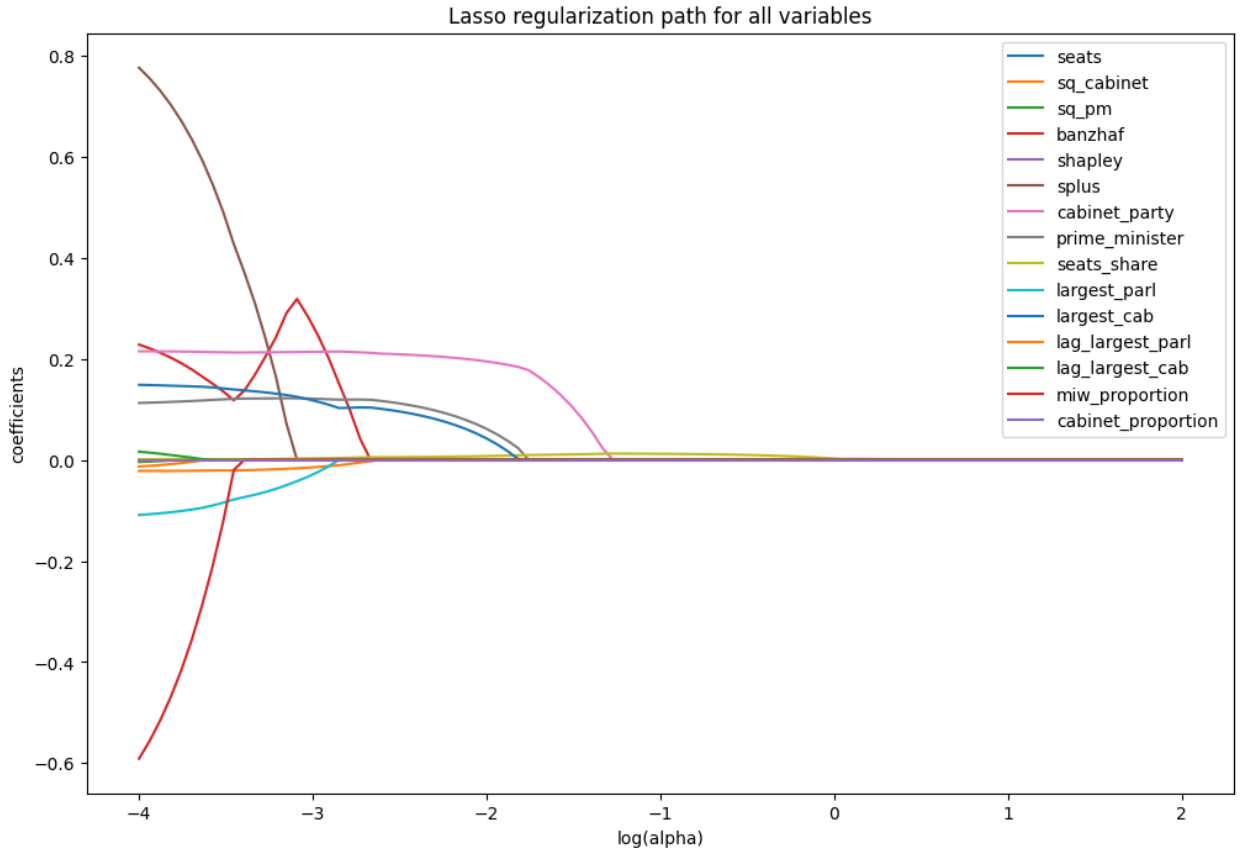
Understanding Relationships between IV's

From the correlation matrix and plots, I saw that all of the country's dummy variables didn't have any correlation with the dependent variable, so I deleted them. Many other variables, like A, B, B_star, C, D, E, and others had little to no correlation to the dependent variable, so I deleted them knowing they would not help with the regression.

A lot of these uncorrelated variables make sense to remove because the proportion of cabinet ministries by each party should be most dependent on the measures of pivotality. I also deleted all of the variables that had nonnumerical values, like party_name and party_name_english. Since we are not working with time-series data, I removed the election_date and start_date columns. cabinet_proportion is calculated by finding cabinet_seats/total_cabinet_size, which means those two independent variables directly calculated the dependent variable, so I deleted them as well. The seats and base variable are the exact same data, so I deleted base.

Lasso with new IV's

That left seventeen independent variables in the dataframe. Now that they were narrowed down, I wanted to see if these variables could all be modeled by a decision tree or lasso, or if I needed to do PCA. The results of the lasso with cross validation are shown below.



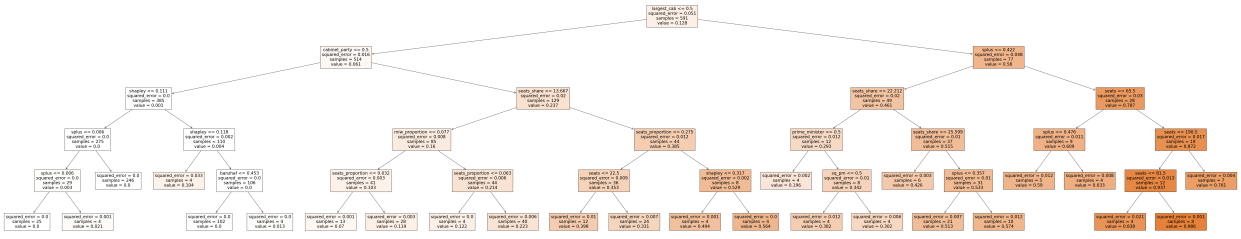
Optimal alpha: 0.0006428073117284319

Mean R-squared: 0.8363700320720285
Training MSE: 0.006554991078619206
Training R-squared: 0.8721144987075906
Test MSE: 0.006716199733410161
Test R-squared: 0.8692747380378466
Equation: 0.1294 + 0.0065 * seats + -0.0094 * sq_cabinet + -0.0004 * sq_pm + 0.0251 * banzhaf + 0.0000 * shapley + 0.1226 * splus + 0.1022 * cabinet_party + 0.0373 * prime_minister + 0.0094 * seats_share + -0.0355 * largest_parl + 0.0491 * largest_cab + -0.0032 * lag_largest_parl + 0.0038 * lag_largest_cab + -0.0653 * miw_proportion

The lasso regularization path shows that seats_share and cabinet_party are the most important variables in the regression, which is unsurprising given their high correlation with the dependent variable, and it makes sense because the parties with the greatest share of seats and the parties in the winning coalition should have a greater number of votes, hence greater cabinet_proportion. Shapley had one of the highest correlations with the dependent variable, so it was surprising that shapley seemed to be the least important variable, as its coefficient went to 0 with an alpha value basically at 0. With an alpha value so low, the model is basically not penalizing complexity at all, meaning the model is not overfit. Although this model seems strong, it is still uninterpretable because of the sheer number of variables in it.

Decision Tree with New IV's

```
Best Parameters: {'ccp_alpha': 0.0, 'criterion': 'squared_error', 'max_depth': 5, 'min_samples_leaf': 4, 'min_samples_split': 2}
Best Mean Squared Error: 0.005190597652527759
Mean Squared Error on Training Data: 0.002216740147211226
Mean Squared Error on Test Data: 0.0020742193440828993
R-squared: 0.9639758560461477
```



At first glance, the performance of the decision tree seems to be incredibly good, with an r-squared of .96. The variables with the highest correlation are chosen as the first few splits in the decision tree, which is unsurprising because these splits result in the

highest impurity. Although the model seems pretty good, the complexity is still very high, which makes it hard to interpret.

PCA 1

I still wanted to narrow it down to at most five variables before running regressions to prevent overfitting and improve interpretability. Accordingly, I decided that PCA was the best option to reduce the dimensionality. To do this, I needed to group together the variables with the highest multicollinearity. I looked at a combination of the VIF table and the covariance heatmap, and it showed that banzhaf, shapley, splus, miw_proportion and seats_proportion, seats_share were heavily multicollinear, as shown in the VIF table below.

	Variable	VIF
0	seats	inf
1	sq_cabinet	2.701790
2	sq_pm	4.334056
3	base	inf
4	banzhaf	323.066500
5	shapley	579.770730
6	splus	235.525198
7	cabinet_party	4.549894
8	prime_minister	6.130834
9	cabinet_seats	7.764797
10	seats_share	630.517077
11	largest_parl	5.620232
12	largest_cab	7.976134
13	lag_largest_parl	4.763625
14	lag_largest_cab	5.137753
15	miw_proportion	52.699542
16	seats_proportion	674.354995

This makes intuitive sense, as these variables are measures of pivotality, which is a strong measure of the proportion of cabinet seats each party holds. If a party has great pivotality means it has a lot of power with each vote, and is more likely to swing the

overall result of the election. For this reason, parties with greater pivotality should have a greater proportion of cabinet ministries.

Putting these six variables into a PCA results in one latent variable explaining 95% of the variance, which was a good amount to definitely reduce the six variables into one latent space.

```
Variance explained by each latent variable in PCA:  [0.95160188]
   Explained Variance  Cumulative Explained Variance
0            5.718315                     5.718315
          PC1
0   -1.491552
1    5.893208
2   -0.929852
3    0.238362
4    4.480637
..        ...
652 -2.118068
653  4.314443
654  2.352023
655 -1.317691
656 -2.078836
```

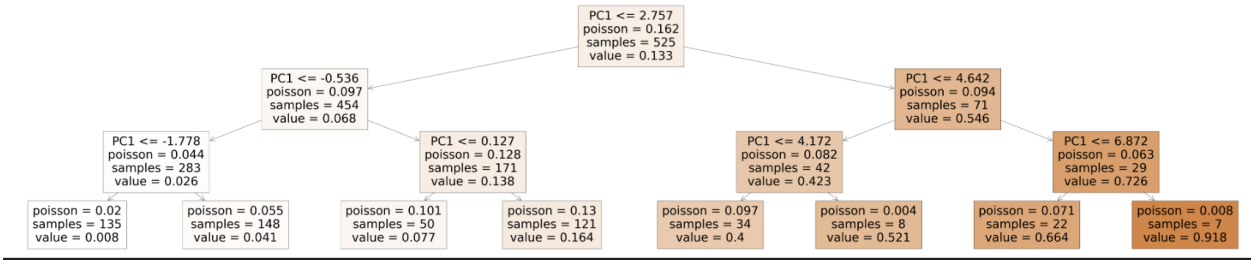
Testing Models with PC1

Since this PCA variable seemed to hold the variables with the highest correlation with the dependent variable, I tried running a linear regression, polynomial regression and decision tree on it, just to test if only using these variables would produce a good model. The results of the linear and polynomial regression are shown below.

```
Linear Regression (Train) - Training MSE: 0.0212, R-squared: 0.5863
Linear Regression (Test) - Training MSE: 0.0242, R-squared: 0.5289
Polynomial Regression (Train) - Training MSE: 0.0202, R-squared: 0.6056
Polynomial Regression (Test) - Training MSE: 0.0237, R-squared: 0.5379
Linear Equation
0.1345 + 0.0740 * PC1
Polynomial Equation
0.1116 + 0.0000 * PC1^1 + 0.0653 * PC1^2
```

Although these models are much more interpretable than the baseline model with all the variables thrown in, they perform much worse with an r-squared of around .52. Now knowing that the model could perform better, I wanted to add more variables to the regression.

```
Best Parameters: {'ccp_alpha': 0.0, 'criterion': 'poisson', 'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2}
Best Mean Squared Error: 0.021112223139222838
Mean Squared Error on Training Data: 0.0183330737406026
Mean Squared Error on Test Data: 0.021917449132793827
R-squared: 0.5733950160574136
```



The decision tree doesn't do much better, with an r-squared of only .57. Also, the decision tree didn't really make sense because there was only one variable, and it was only splitting that variable by value at each level.

PCA 2

Knowing that I could not fit a good model with just PC1, I needed to include the other correlated variables into the model. The other variables outside of these six variables were only mildly multicollinear, given by the covariance heatmap, so it would have to take more latent variables to represent all eleven of the variables. The results of performing PCA on the rest of the variables are as follows:

```
['partyPC1', 'partyPC2', 'partyPC3', 'partyPC4']
Loadings:
                 partyPC1  partyPC2  partyPC3  partyPC4
cabinet_party     0.246100  0.400871 -0.632311  0.004147
prime_minister    0.366185  0.371837  0.091315 -0.205842
seats             0.254513  0.155243  0.206786  0.929264
sq_cabinet        0.248083 -0.343729 -0.668838  0.144657
sq_pm             0.364468 -0.375090  0.048814 -0.017707
largest_parl      0.379036  0.182324  0.214194 -0.135191
largest_cab       0.378472  0.358648  0.094971 -0.192092
lag_largest_parl  0.358976 -0.332520  0.191640 -0.090686
lag_largest_cab   0.361126 -0.381651  0.087934 -0.097167
Variance explained by each latent variable in PCA:  [0.56802595 0.12840423 0.10463016 0.07712949]
   Explained Variance  Cumulative Explained Variance
0            5.120027                      5.120027
1            1.157400                      6.277426
2            0.943107                      7.220533
3            0.695224                      7.915757
     partyPC1  partyPC2  partyPC3  partyPC4
0   -1.288041 -0.049447  0.574707 -0.140985
1    6.410413 -0.481952  0.033187 -1.844815
2   -1.348372 -0.086246  0.525690 -0.361261
3   -1.342887 -0.082901  0.530146 -0.341236
4    6.388475 -0.495333  0.015362 -1.924916
..        ...       ...       ...       ...
...
655 -1.293525 -0.052792  0.570251
656 -1.326433 -0.072865  0.543514

[657 rows x 3 columns]
```

The fourth variable in this PCA explained much less than a variable of variance, shown by its eigenvalue and the fact that none of the variables had a strong correlation with PC4, shown from the loadings table. Accordingly, I only used the top three PCA variables.

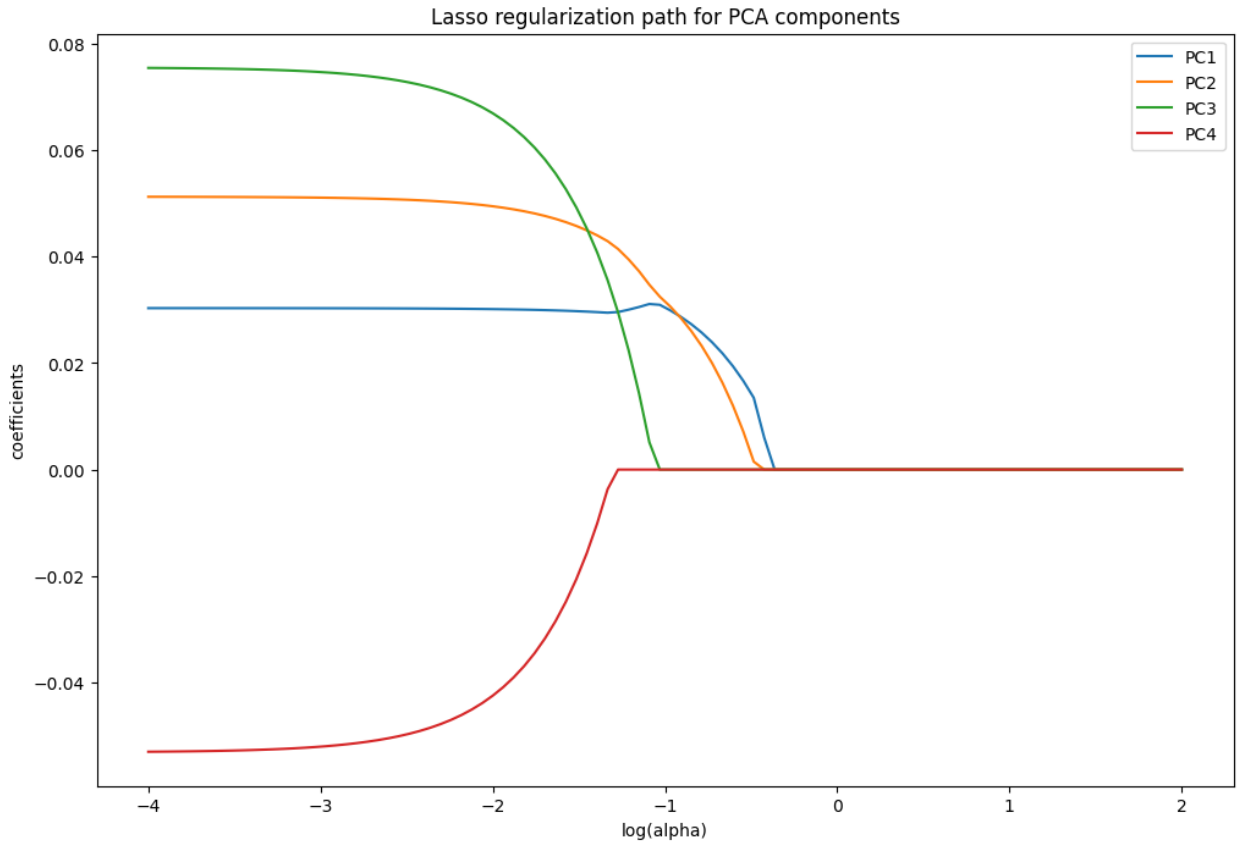
Linear Regression with all PCA Variables

I added these three PCA variables into a dataframe with the other PCA variable that summarized the pivotality variables. Now that there was more data to work with in addition to the significantly correlated pivotality PCA variable, I expected the regressions to have strong predictive power, especially because all the variables in the second PCA had good correlation with the dependent variable. I first ran a linear regression model, then a lasso with cross validation, then a decision tree to see if that would improve on the linear regression model. The results of the linear regression are below:

```
Linear Regression (Train) - Training MSE: 0.0091, R-squared: 0.8233
Linear Regression (Test) - Training MSE: 0.0089, R-squared: 0.8276
Linear Equation
0.1322 + 0.0312 * PC1 + 0.0520 * partyPC1 + 0.0737 * partyPC2 + -0.0535 * partyPC3
```

PC1 is the latent variable I got from running the first PCA, and partyPC1, partyPC2, and partyPC3 are the latent variables from the second PCA. This model seemed to perform well, with a high r-squared without overfitting or underfitting, as the test MSE and rsquared are very similar to the train MSE and r-squared. Now, I wanted to compare this to the lasso.

Lasso with PCA Variables



Seen in the lasso regularization path, which shows how the coefficients of each variable move with respect to alpha, PC1 is the most important of the latent variables because it requires the greatest penalty term in order for the coefficient to converge to 0. This is expected: PC1 explained the most variance and contained the variables that had the

highest correlation with the dependent variable. The results of the lasso with cross validation are shown below.

```
Optimal Alpha: 0.0004
Mean Squared Error (Lasso + LOOCV): 0.0093
R-Squared: 0.8195
Equation: y = 0.1293 + 0.0303 * PC1 + 0.0512 * partyPC1 + 0.0752 * partyPC2 + -0.0527 * partyPC3
```

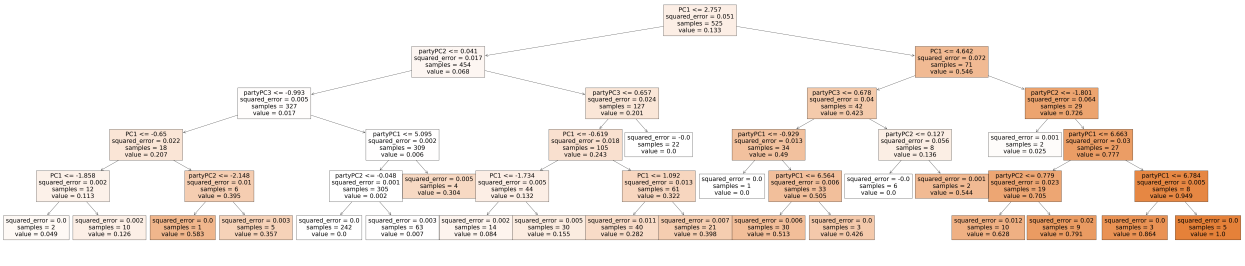
As seen, the alpha value that was found to be optimal is tiny, which doesn't change much about the model. For this reason, it is very similar to the linear regression model. Given that the optimal alpha is so low, this shows that the model is not overfitting because if it was, the penalty term would be higher to try to reduce complexity.

Both the linear regression and lasso models are relatively interpretable. PC1 represents pivotality, and the greater pivotality a party has, the greater cabinet proportion, thus why there is a positive correlation between the two in the equation. partyPC3 has a strong negative correlation with the variables cabinet_party and sq_cabinet, both of which have a positive correlation with the dependent variable; if the party has held power in the previous election(sq_cabinet) or is in the winning coalition(cabinet_party), that party should have more power and thus more seats in the cabinet. Thus, it makes sense why partyPC3 has a negative correlation with the dependent variable.

Decision Tree with PCA Variables

Lastly, I did a decision tree model with grid search on the full PCA data frame.

```
Best Parameters: {'ccp_alpha': 0.0, 'criterion': 'squared_error', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 5}
Best Mean Squared Error: 0.009133580411584203
Mean Squared Error on Training Data: 0.003727727146926512
Mean Squared Error on Test Data: 0.007406385362786333
R-squared: 0.8558408467326364
```



Although the model looks complex, it is not too overfit, shown by the comparison between the training MSE and test MSE. The higher levels of the decision tree are split

by the most important variables which explain most variance. For example, the first split is PC1, which makes sense because based on the other models and the variables seen in PC1, it seems to be the most correlated with the dependent variable. As a result, this first split results in high impurity. Despite its good performance, the model isn't very interpretable because of its complexity and a new reader cannot understand the relationships between the independent variables and the dependent variables from this decision tree, unlike the linear regression and lasso models.

Conclusion

In searching for the best-performing, most interpretable model, it was important to understand the relationships between each of the variables and the dependent variable in order to understand which variables needed to be included and excluded in the model. It was proven that the measures of pivotality(Banzhaf, Shapley, splus, miw_proportion, seats_share, and seats_proportion) were most influential in the model. They also measured very similar things, which is why PC1 summarized all six variables. I acknowledge that it probably would have been better to exclude the variables that directly calculated the others, like how seats_share and splus because they were very similar to seats_proportion and Shapley, respectively.

Despite losing interpretability, including the other PCA variables improved the model significantly. The first PCA variable didn't have a high enough performance while the lasso and decision tree models with all variables mixed in was uninterpretable, making the models with the four PCA variables a good balance between the two.

The most interpretable model that performed well was the linear regression with the PCA variables, as described in the "Linear Regression with All PCA variables" section.

cabinet_proportion = 0.1322 + 0.0312 * PC1 + 0.0520 * partyPC1 + 0.0737 * partyPC2 + -0.0535 * partyPC3