**DATA:**

This dataset was easy to work with in terms of dropping na/imputing because very few columns had a large amount of missing data (with the exception of coalition\_total, left\_rightx, and left\_righty). Regardless, I ran basic summary statistics on each variable in the dataset.

The first step I took towards understanding the dataset was to check for any NA values; the only columns with a significant amount of missing data were coalition\_total, left\_rightx, and left\_righty (caretaker also had a -9999 value). Following this step, I ran summary statistics for each variable and the following table displays the results (the table also compares the summary statistics to those after dropna is implemented):

A graph with numbers and text

Description automatically generated with medium confidenceA graph with text on it

Description automatically generated

The standard deviation seems to drop significantly for some variables after dropping na, but I’ll go into how I deal with missing data later. My next step was to look for multicollinearity among the variables because, through theory and simply looking at the data, it is clear to see that many of the IV’s are simply ratios of one another. To test this, I created a VIF table and a correlation matrix to view which variables may be strongly correlated (I know that a correlation matrix does not capture correlation between groups of variables, but I still found it helpful to run for a basic view of correlation within the dataset). This is a sample of my correlation matrix: A table of numbers with text

Description automatically generated

The high correlation between Banzhaf, Shapley, splus, and miw\_proportion most stands out I took note of these relationships as I went on to feature selection.

**FEATURE SELECTION/PCA:**

As I mentioned earlier, it is extremely important to be able to capture all of the relationships between each variable because there are so many in this particular dataset. For instance, base and seats are identical, and the voting-power scores like Banzhaf have very high correlation. Regardless, my first step was to drop any unnecessary variables and my process is the following:

'base', - identical to seats

'party' – identifying info

'election\_year', - identifying info

'country', - ideally, there shouldn’t be bias towards a particular country because the voting system should operate such that it does not matter where the country is from – otherwise the system of establishing coalitions is biased. In this model, voting bias related to a country is not being studied. (thus all country variables are dropped).

'cabinet\_name' – identifying info

'caretaker' – after compared with DV, has little to no predictive power and is 0 for nearly all rows.

'cabinet\_id' – identifying info

'party\_id' - identifying info

'prime\_minister' – does not have any relationship with # of seats picked.

'cabinet\_party' – shouldn’t be biased towards particular party, could be collinear with seats of a party because if a party has a lot of seats, it’s typically part of a coalition (excluded because it’s a weaker predictor than # of seats and also varies vary little)

'left\_righty' – too many NA

'left\_rightx' – similar to left\_righty

'cabinet\_seats' – entirely collinear with the DV, needs to be dropped otherwise perfect prediction

'party\_name' - identifying info

'party\_name\_english' - identifying info

'country\_id' - identifying info (again, maybe useful for predicting bias, but not for the scope of this project)

'election\_id' - identifying info

'election\_date' - identifying info

'start\_date' - identifying info

'post\_election' –very little variance

'mingov'

'bicameral',

The two above are simply cabinet descriptors not related to number of seats.

'largest\_parl'

'largest\_cab'

'lag\_largest\_parl'

'lag\_largest\_cab'

All of the above are binary characterization of cabinet and party and aren’t necessarily needed and shouldn’t be used because of possibly collinearity with DV.

'A','B','B\_star','C','D','E','country\_dummy1','country\_dummy2','country\_dummy3','country\_dummy4','country\_dummy5','country\_dummy6','country\_dummy7','country\_dummy8','country\_dummy9','country\_dummy10','country\_dummy11','country\_dummy12','country\_dummy13'

All of these are binary variables that I initially choose to drop, but I will reimplement them after running a regression to see if they have any predictive power.

'coalition\_total' – too many NA

After dropping IV’s, I decided to drop na on the few remaining rows that still had nan values with the following results (essentially identical statistics before and after dropping):

A graph with brown squares

Description automatically generatedA graph with brown squares

Description automatically generated with medium confidence

**As for explanations of my 2 latent variables:**

pca1 = ['shapley', 'splus', 'banzhaf', 'miw\_proportion']

It clear that Banzhaf, shapley, and splus are very colinear because they are all measures of voting power of the party and appear to be describing the same thing. miw\_proportion is included because it represents a measure of voting power of the party, so I decided to include it. Miw\_new wasn’t included because it seems not to be correlated with the other miw variable (initially I included it but did not get very good results through the principal components and I got a much better proportion of variance explained after taking it out).

A graph of a graph

Description automatically generated with medium confidence

pca2 = ['seats','seats\_share','seats\_proportion']

This latent space was created because all of these variables are highly correlated, and all simply related to the number of seats a party has in a parliament. Seats\_total was not included in the pca2 group because it is a number for the whole parliament rather than the particular party. Additionally, seats\_total was dropped from all IV’s because it is part of the seats\_proportion ratio.

A graph with blue rectangular bars

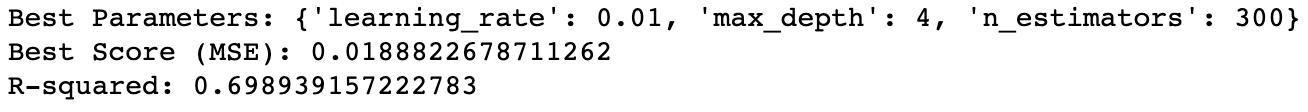
Description automatically generated

I also tried a pca of ‘W’,'A', 'B','B\_star','C','D','E' because they are related to the party coalition structure, but I got poor results and did not include it. Thus, I only kept W because it may have predictive power.

**LASSO/DECTREE/XGBOOST:**

(all methods use k-fold and loop over hyperparameters)

At this point, I have a list of around 11 variables (644 columns) and I want to first run xgboost to give a upper limit of what my performance should look like:



Following, I ran a lasso and decision tree model:

A screenshot of a computer code

Description automatically generated  
A graph with blue dots

Description automatically generatedA screenshot of a computer

Description automatically generatedA diagram of a company

Description automatically generated

(After including each of the binary variables I previously excluded, none were viewed as important by either model)

**EVALUATION OF MODEL:**

The performance of both models is sound, and the variables picked between the decision tree and lasso are more or less the same which is a good indicator that I did not mess up somewhere along the way in coding each model (also, I changed the hyperparameters a bit to check if my models were fragile or not).

Lasso picked: 'latent1', 'latent2', 'sq\_cabinet'

Dec Tree split on:

Feature Importance

8 latent1 0.777272

9 latent2 0.186486

7 W 0.022786

0 sq\_cabinet 0.013456

**Latent1:**

I presume latent1 is picked because it was designed to represent the voting power/influence of a particular party. Thus, if a party has more influence, it is more likely to have seats- there should ideally be a positive correlation, and this checks out in the lasso. This also explains the concept of pivotality as even smaller party can display a strong voting power.

**Latent2:**

Latent2 is a latent variable of a percentage/proportion of seats a party has obtained in the parliament and, intuitively, if a party has many seats in parliament- it will also have a lot of seats in cabinet. However, this is a weak correlation because a lot of parties can have a large number of seats but do not form coalitions well because of political orientation or other alliance-based reasons (i.e., a very left-wing party cannot easily form a coalition with a right wing). Therefore, it is likely chosen with less importance than latent1 because of the intricacies of the way voting is established in which the calculated scores found in latent1 better explain the DV.

**Sq\_cabinet:**

Sq\_cabinet shows the party was in the cabinet previously, and this is likely picked as a decent predictor because, if a coalition was previously successful, it is more likely that this party will participate in another successful coalition and gain seats.

**W:**

This represents the threshold for forming a coalition and if the threshold is high/low then many/few parties have to participate in the coalition. So, this variable indicates the distribution in the seats of the cabinet and can weakly predict the DV.

Thus, it makes sense that latent1 is always picked first in importance because it ideally shows a direct relationship of the voting power of a party with the cabinet proportion. This justifies why such indices, like Banzhaf and Shapely, to be attempt to predict cabinet seats have been introduced in the first place. Additionally, Shapley introduced the idea of cooperative game theory, and this can be directly applied to coalition formation and can help explain why the index of voting power is a better predictor than simply the number of seats a party has in parliament.

As for whether I trust the lasso or the decision tree; I’d rather trust the results of the decision tree because of its ability to capture nonlinear relationships in the data which may be found in the indices in latent1 which are non-linear. They are non-linear because voting power is not necessarily proportional to the number of seats a party. For example, the large party can have a lower power index if they are unable to form a coalition by themselves whereas a middle-sized party could be critical to form the coalition. Also, the decision tree includes some variables that the lasso leaves out, like W, which make theoretical sense to be included in the prediction model (explained earlier). Thus, I feel that a decision tree is more equipped to handle this type of data, at least in a theoretical sense.