***236756 - Introduction to Machine Learning – HW5***

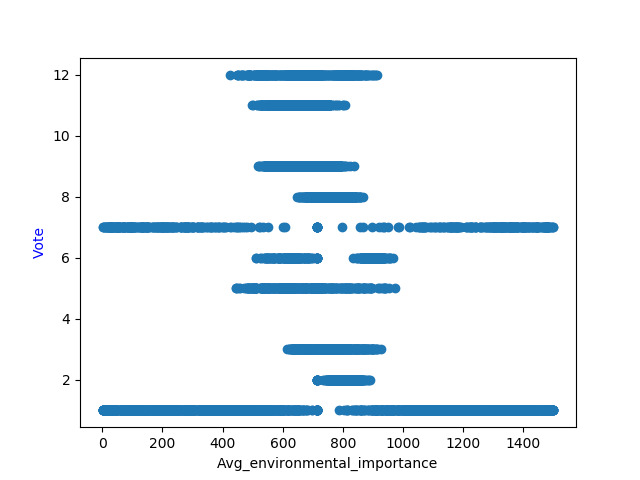
***Report***

***Data preparation:***

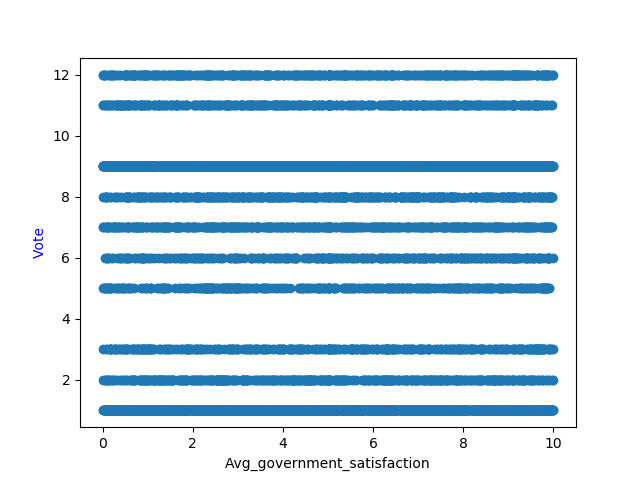
First, we converted every nominal feature to numbers, so we can work with numerical data.  
We found out there are 3 types of features:  
The nominals, which are strings. The integer features, which contain integers. And the float features, which contain values with decimals.

We decided to fill the missing values with the following way:  
For the nominals, the value will be the most present value in the column.  
For the integers, the value will use the mean, rounded up to the closest integer.  
And finally for the floats, we’ll use the mean.

This way, we fill with values of the same type for each feature.  
  
Now, about the imputations:  
We decided to use the z score for removing the outliers, as it seems to be a popular way to do so, with a threshold of 3.

Then came the normalization step:  
The non-nominal features should be normalized, so we divided them in two groups: the uniform features and the normal features.  
To do so, for each non-nominal feature we plotted a graph showing the distribution of the feature according to the votes. By looking at it, we think we could tell which feature belonged to which group. For instance:

On this graph, we can see that the distribution is not uniform, however, on this graph:



We can tell that each vote gets the same distribution, so we can conclude that the first one is a normal feature and the second is a uniform feature.

The uniform features we scaled to the range (-1, 1) and the normal ones were scaled using StandardScaler().

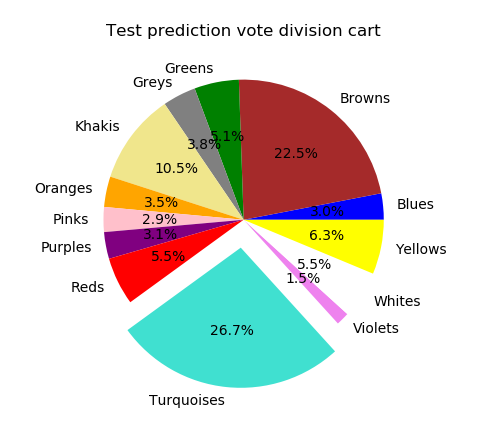
***Winner prediction:***

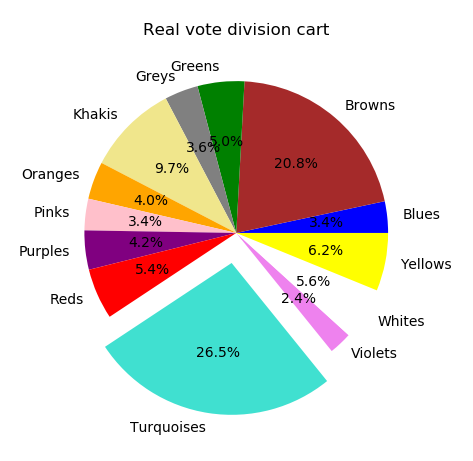
After many trials, we identified that *the RandomForestClassifier(n\_jobs=-1, random\_state=2, criterion='entropy')* gave the best results overall (performance-wise and accuracy-wise), so we decided to use it for the next steps.

The party that wins the elections is: ***’Turquoises’***.

***Division of votes***

Thanks to the data provided by our model, we were able to print the following distributions:





We can see that both are relatively close, which is a good sign.

Indeed, we get an accuracy of ***92.5%***

***Predictions on the new data:***

We predicted using our previously mentioned model, the output file named ***new\_test\_voting\_predictions.csv***

***Steady Coalition:***

Using clustering, we managed to form a steady coalition.

We are going to use the clustering model we studied in class, K-Means.  
By using cross validation, we approximated the best hyper parameters for the model.

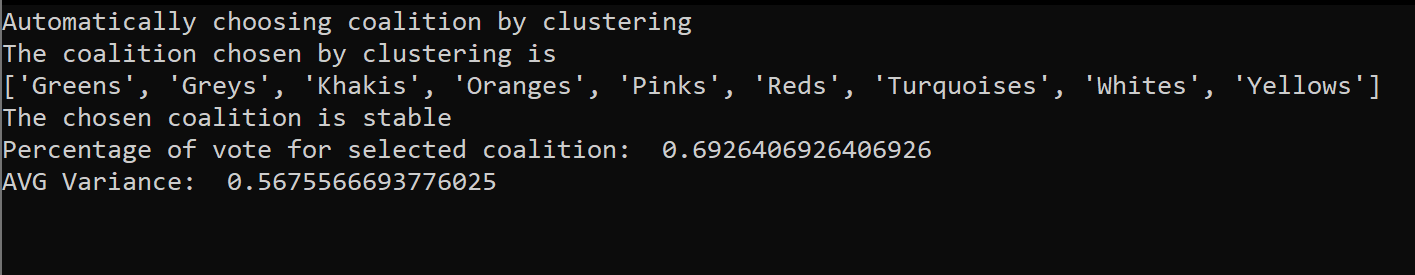
Then, using clustering we will group the similar voters and get their parties, in order to form a homogenous coalition.

We set a certain threshold value from which we decide that a voter belongs to a specific group, for example 45% for k=3 clusters.

We used the following process to form a stable coalition:

1. Training the K-Means models with the train set with different k values and threshold.
2. Checking the accuracy on the validation set and tuning hyper parameters (k and threshold) accordingly.
3. Checking the accuracy on the test set.
4. Then, form a coalition by getting parties belonging to the biggest cluster.
5. To get a coalition which is different than the opposition, we check that a given party is not overrepresented in the other clusters.

Finally, we get these results:



We can see from these results that we can get stable coalition for k=3 and threshold=0.45

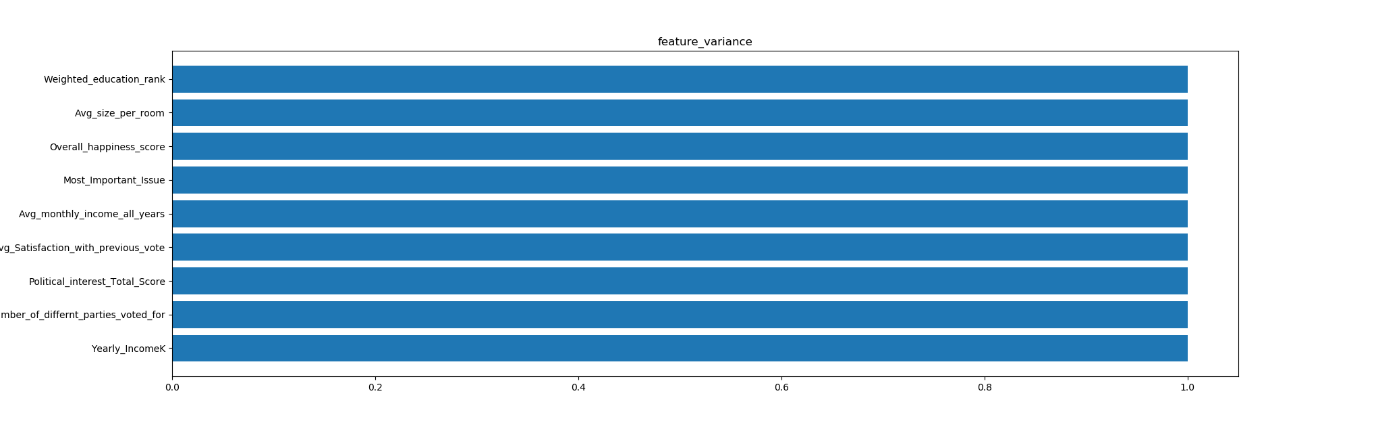
The final coalition consists of 9 parties, which are:



We get 69.26% of the votes in the test set with this coalition.

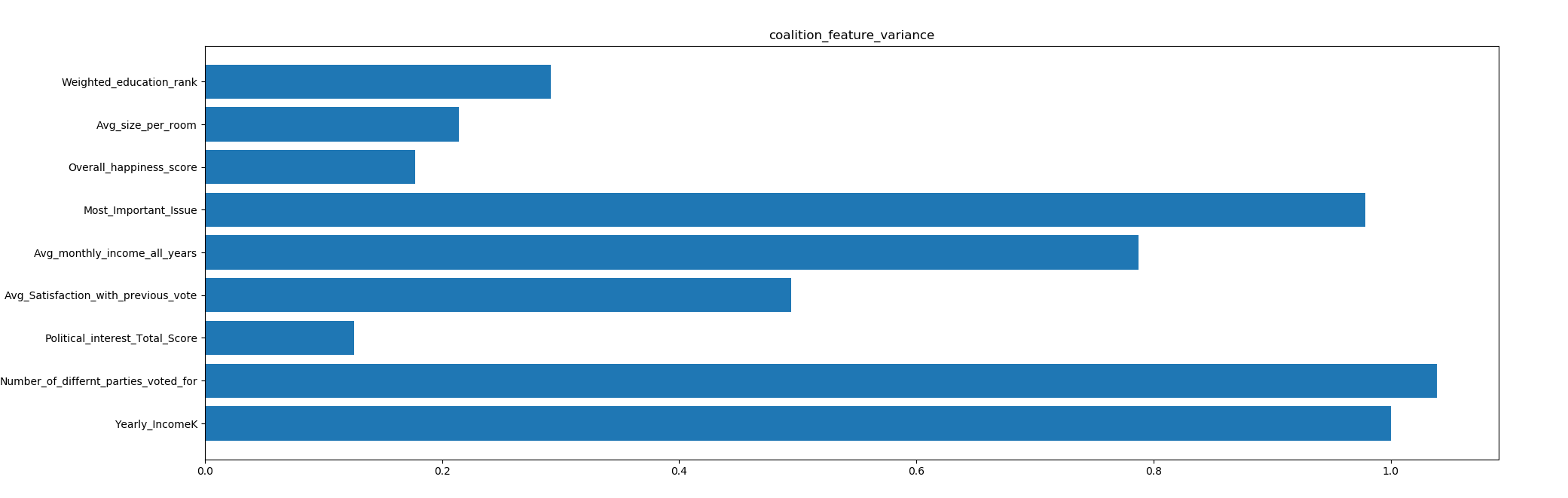
Homogeneous: The homogeneity of the coalition is determined by threshold. A bigger threshold gives a better the homogeneity. After lots of experiments, for different k and different threshold, we determined that we get stable coalition for k=3 and threshold=0.45, and if we increase the threshold more than k=0.45, the results are not good enough.

Before choosing coalition we got variance like this:



(Every feature has a variance of 1)

And after choosing coalition we get:



We can see from the graphs that the variance is smaller, hence the coalition is more homogeneous. This gives us an average variance of 0.5676