Mori Levinzon 308328467
Nadav Rubinstein 208686659

Mandatory Part – Loading and Preparing the Data:

First, we loaded the sample file again and applied the preliminary data manipulations to prepare the information as we did in the previous exercises. These manipulations include:

1. Split the total sample set (using the stratified shuffle split) into 3 sets:

Data Set	Percentage from Original Data Set
Train set	65%
Validating set	10%
Test set	25%

2. Extract outlier values, for example negative values.
3. Choosing the right set of features as selected in exercise #3.
4. Normalize the categorical values and Z-scale the nominal values.
5. Export the information to 3*2 csv files : before and after the change.

After that, we approached the prediction task when in the mandatory part we were required to predict:

- A stable coalition comprising 51% of all votes.
- Identify dominant features for each party and manipulate in order to strengthen the coalition and build an alternative coalition.

Mandatory Part – Building Steady Coalition:

In order to build a stable coalition as defined in the exercise, we are required to find a group of parties with similarities of their voters. Since in this problem most of the voters (after the feature selection) are features of continuous numeric values, we can measure the distance between two samples using the numeric values of the sample's properties. In order to group the parties into a homogeneous coalition, we used the following two methods:

1. Clustering model – When we use this type of model we actually "get"
   The ability to identify similarities and differences between voters through the ability to refer to voters who are in the same cluster as having similar characteristics while voters who are not in the same cluster as unrelated.
   In this way, we will try to create a voting group that belongs to a particular cluster that is relatively homogeneous and that will form the basis of the coalition.
2. Generative Model - As a feature of this model after training the model we can get the properties of a particular party probability, for example, in the case of Gaussian Naïve Base we can extract the variance and expectation for a particular party distribution (we can note that for our problem these are vectors at the size of 9, as the number of features) . As a result, we can measure similarities between two parties by comparing the characteristics of the probability functions for each of them.

These are, in fact, the main ways of action that we have developed and with the help of each we have been able to build a coalition by definition as explained in more detail below.

Building Steady Coalition Using Clustering Model:

To build a stable coalition by using the Clustering Model, we trained two models we knew from the class: KMeans and Gaussian Mixture.

First, for each of the models, we will need to select the best coalition hyperparameters and for that we used K Fold Cross Validation, although these are models belong to unsupervised learning, we have to develop ourselves a metric on how to choose better hyperparameters for the problem we are dealing with.

The metric we developed is a model preference that tries to concentrate each party in a particular cluster rather than spreading it across several clusters in a manner indistinguishable to which cluster it belongs, meaning we prefer models that do not scatter each party among several clusters but create clusters that are distinctly distinguishable by which party Belongs to every cluster. For example, if a particular model tries to divide the information into 2 clusters and we get that for each party 51% of its votes are in a particular cluster and the rest in the other then the model does not divide the votes in a way that allows us to build a sufficiently homogeneous coalition. In contrast, there are parties in which at least 70% of the party votes belong to a particular cluster and the rest to the other, so we prefer this model more because it creates more clusters grouped by parties.

So the way we did K Fold Cross Validation is:

1. Model training on each training part.
2. Prediction on the test portion.
3. For each of the clusters created, we calculated how many parties belong to this cluster (cluster size). A party considered to belong to the cluster if 60% of its total votes belong to that cluster.
4. Calculate the sum of the sizes of all clusters from all parts.
5. Preference for models with larger clusters.

Using this method, we tested 3 possible cluster numbers: 2,3,4 for KMeans and Gaussian Mixture.

The selected models are:

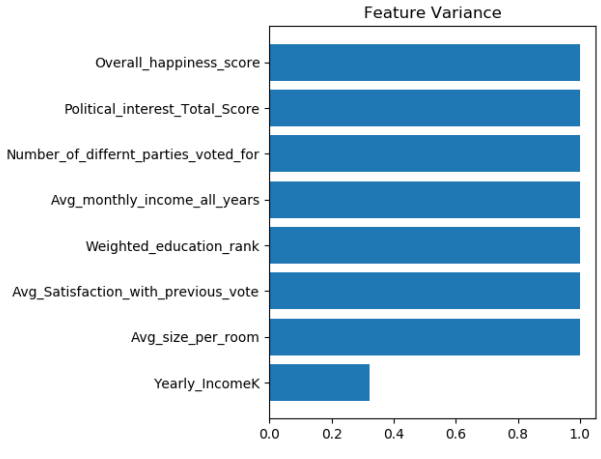
- KMeans with 2 clusters.
- Gaussian Mixture with 2 clusters.

After choosing these two models, we approached the forming of the coalition. The steps for forming the coalition are:

1. Training each of the models on each training set.
2. Assemble the expected coalition using the validation set. Each cluster that the model created can form the basis for a coalition, we define that **a party belongs to a particular cluster if and only if at least 80% of all it votes belong to that cluster** and in addition the total number of parties belonging to the cluster exceeds 51% of all votes. In this way we have built a coalition with at least most votes, a relatively homogeneous coalition because for parties that belong to the cluster, the model has chosen to put most of their voters in the same cluster and they all have similarities between their characteristics, as opposed to parties that do not belong to the cluster, since there is no significant majority of their voters in the cluster, that why they will be in opposition.
3. Filtering identical coalitions and selecting the most homogeneous coalition, a coalition where the variance between voter characteristics of the entire coalition is the smallest.
4. Examine the model performance by attempting to train a coalition using the test set.

Results:

We note that before a coalition is formed, it can be seen that this is the variance between the voters characteristics on the training set:

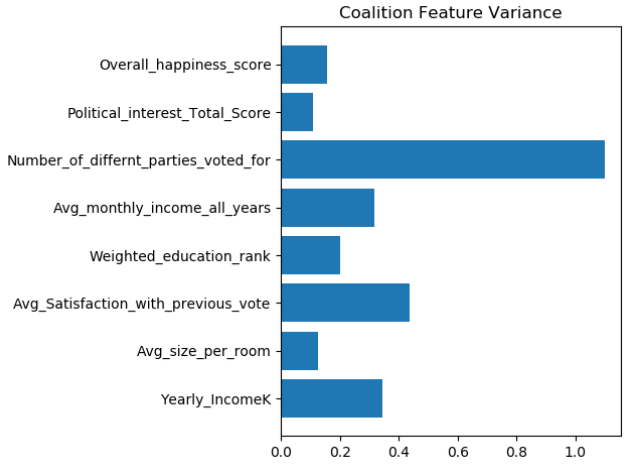


(the features are scaled)

The final coalition selected was built by the Gaussian Mixture model using the validation set . The coalition includes the following parties and comprises 59.7% of all votes.

Greens, Greys, Khakis, Oranges, Pinks, Reds , Turquoises, Whites.

After forming a coalition, we can see that there is a decrease in the variance of most voter characteristics belonging to the coalition on the training set, which indicates that this voter group is more homogeneous than the overall group:

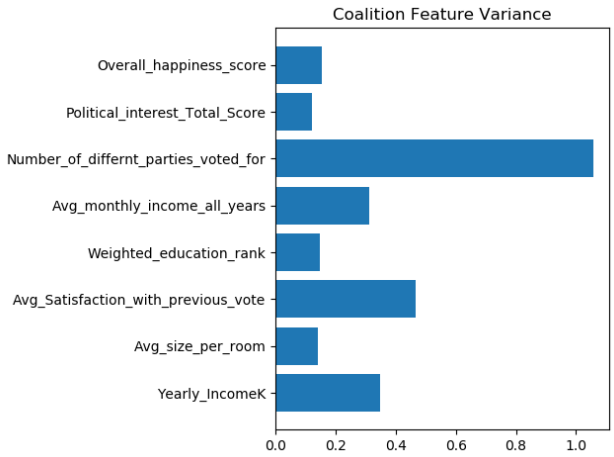


To select the model performance, we used the test set as when it comes to predict the rating of each voter we used the classifier we chose in the previous exercise with 92.64% accuracy and is:
RandomForestClassifier(random_state=0, criterion='gini', n_samples_split=3, min_samples_leaf=1, n_estimators=500)

The selected final coalition was built by the Gaussian Mixture model using the test set. The coalition includes the following parties and comprises 60.1% of all votes.

Greens, Greys, Khakis, Oranges, Pinks, Reds , Turquoises, Whites.

After forming a coalition, we can see that there is a decrease in the variance of most voter characteristics of the coalition on the training set and almost equal to the coalition built with the validation set.



The model we built was able to build a coalition by definition and even predict the coalition in advance with the help of the validation training set and thus is "Inclusive" well.

Building Steady Coalition Using Generative Model:

In order to build a stable coalition by using the Generative Model, we trained two models we know from the class: QDA and Gaussian Naïve Base.

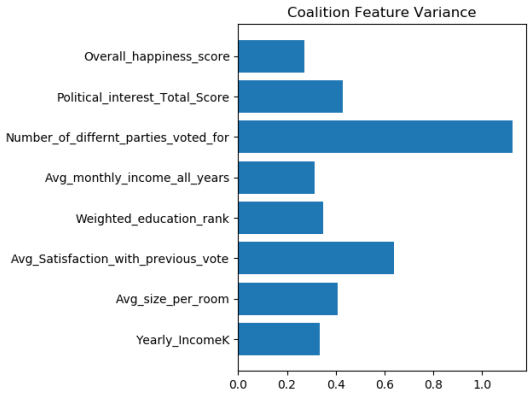
First, for each of the models, we need to select the best coalition hyperparameters and for that we used K Fold Cross Validation when the measure we wanted to maximize is the precision measure.

After choosing hyperparameters for each of these models, we approached the coalition assembly. The steps for forming the coalition are:

1. Calculate the expectancy vector of each party by using one Vs. all and training each of the models on each training set.
2. Assemble The Expected Coalition Using the Validation Set, inspired by the political system in Israel Where the coalition build rest on a particular party. We have tried to form a coalition by imposing the coalition on a specific party that forms the basis of the coalition. Another party to the coalition when its span vector is closest to the borrower's span vector trying to form the coalition. The coalition build process stops when a majority of at least 51% of all votes is obtained.
3. Filtering identical coalitions and selecting the most homogeneous coalition, a coalition where the variance between the voting characteristics of the entire coalition is the smallest.
4. Examine the model performance by attempting to build a coalition using the test set.

Results:

As shown earlier, this is the variance between the characteristics of the different voters on the training set:

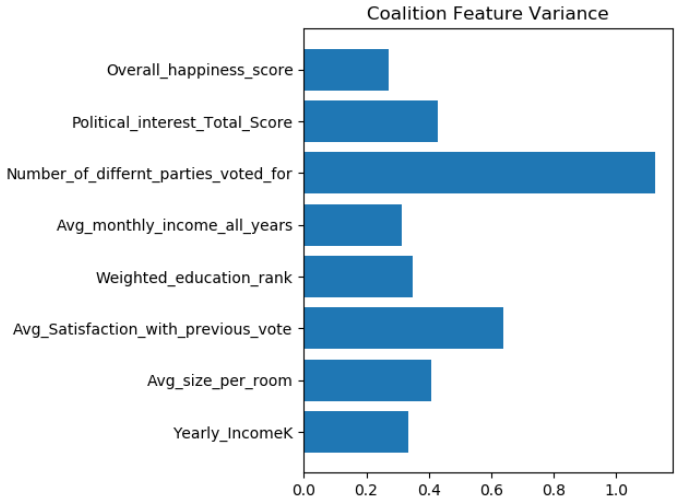


(The features after scaling)

The final coalition chosen was built by **both models** (the two models built the same coalition) with the help of the **validation set** including the following parties and comprising 58.7% of all votes.

Greys, Khakis, Oranges, Pinks, Reds, Turquoises, Violets, Whites.

After forming a coalition, one can see that there is a decrease in the variance of most voter characteristics of the coalition on the training set, indicating that this voter group is more homogeneous than the overall group:



To select the model performance, we used the test set and when we needed to predict the classification of each voter we used the classifier mentioned above.

Since the two models formed the same coalition, we used the Gaussian Naïve Base to build the final coalition using the test set, and the coalition that was formed was exactly the same coalition formed before.

In summary, this model also allowed us to build a coalition different from the coalition built by the previous model and with a nearly identical percentage of votes. When it comes the homogeneity of the 2 coalitions we built, the one that was built by the Clustering model was more homogeneous and therefore we would prefer it.

That is why the coalition we propose is:

Greens, Greys, Khakis, Oranges, Pinks, Reds, Turquoises, Whites.

Identifying Each Party Leading Features:

In the previous exercises we approached this problem by presenting each feature to the possible votes, in this exercise we will approach the problem differently by taking into account what we did in previous exercises.

In order to identify for each party the features that define it we used 'Embedded feature selection' which is an integral part of decision tree.

For each of the parties we have trained our chosen classifier:

```
RandomForestClassifier(random_state=0, criterion='gini', n_samples_split=3, min_samples_leaf=1, n_estimators=500)
```

The classifier was trained using one Vs. all method and we removed lead features using the feature 'feature importance's' that exists in this classifier.

(The Random forest classifier selects before every split in decision trees a set (sometimes a real sub-set and sometimes the whole set) of features from the total feature set. The feature importance feature represents the probability of being split according to the feature and hence the feature by which more nodes are split is considered as the feature of higher importance)

The important features for each party order by importance:

Party	Leading Features
Blues	1. Avg_size_per_room 2. Number_of_differnt_parties_voted_for 3. Yearly_IncomeK 4. Political_interest_Total_Score 5. Avg_Satisfaction_with_previous_vote 6. Most_Important_Issue 7. Overall_happiness_score 8. Avg_monthly_income_all_years 9. Weighted_education_rank
Browns	1. Number_of_differnt_parties_voted_for 2. Political_interest_Total_Score 3. Avg_Satisfaction_with_previous_vote 4. Yearly_IncomeK 5. Most_Important_Issue 6. Avg_size_per_room 7. Overall_happiness_score 8. Avg_monthly_income_all_years 9. Weighted_education_rank
Greens	1. Political_interest_Total_Score 2. Most_Important_Issue 3. Avg_Satisfaction_with_previous_vote 4. Number_of_differnt_parties_voted_for 5. Yearly_IncomeK 6. Avg_size_per_room 7. Overall_happiness_score

	8. Avg_monthly_income_all_years 9. Weighted_education_rank
Greys	1. Weighted_education_rank 2. Number_of_differnt_parties_voted_for 3. Avg_monthly_income_all_years 4. Most_Important_Issue 5. Avg_Satisfaction_with_previous_vote 6. Political_interest_Total_Score 7. Yearly_IncomeK 8. Overall_happiness_score 9. Avg_size_per_room
Khakis	1. Avg_monthly_income_all_years 2. Most_Important_Issue 3. Political_interest_Total_Score 4. Avg_Satisfaction_with_previous_vote 5. Number_of_differnt_parties_voted_for 6. Yearly_IncomeK 7. Weighted_education_rank 8. Avg_size_per_room 9. Overall_happiness_score
Oranges	1. Weighted_education_rank 2. Number_of_differnt_parties_voted_for 3. Political_interest_Total_Score 4. Avg_monthly_income_all_years 5. Avg_Satisfaction_with_previous_vote 6. Most_Important_Issue 7. Yearly_IncomeK 8. Overall_happiness_score 9. Avg_size_per_room
Pinks	1. Political_interest_Total_Score 2. Number_of_differnt_parties_voted_for 3. Avg_Satisfaction_with_previous_vote 4. Most_Important_Issue 5. Avg_monthly_income_all_years 6. Yearly_IncomeK 7. Avg_size_per_room 8. Overall_happiness_score 9. Weighted_education_rank
Purples	1. Avg_Satisfaction_with_previous_vote 2. Political_interest_Total_Score 3. Number_of_differnt_parties_voted_for 4. Most_Important_Issue 5. Yearly_IncomeK 6. Avg_size_per_room 7. Overall_happiness_score 8. Avg_monthly_income_all_years 9. Weighted_education_rank
Reds	1. Most_Important_Issue 2. Weighted_education_rank 3. Number_of_differnt_parties_voted_for 4. Political_interest_Total_Score 5. Avg_monthly_income_all_years 6. Avg_Satisfaction_with_previous_vote 7. Yearly_IncomeK 8. Overall_happiness_score 9. Avg_size_per_room
Turquoises	1. Number_of_differnt_parties_voted_for 2. Political_interest_Total_Score 3. Avg_monthly_income_all_years 4. Yearly_IncomeK 5. Avg_Satisfaction_with_previous_vote

	6. Most_Important_Issue 7. Weighted_education_rank 8. Overall_happiness_score 9. Avg_size_per_room
Violets	1. Avg_Satisfaction_with_previous_vote 2. Most_Important_Issue 3. Number_of_differnt_parties_voted_for 4. Yearly_IncomeK 5. Political_interest_Total_Score 6. Avg_size_per_room 7. Overall_happiness_score 8. Avg_monthly_income_all_years 9. Weighted_education_rank
Whites	1. Avg_Satisfaction_with_previous_vote 2. Most_Important_Issue 3. Political_interest_Total_Score 4. Number_of_differnt_parties_voted_for 5. Avg_monthly_income_all_years 6. Yearly_IncomeK 7. Weighted_education_rank 8. Overall_happiness_score 9. Avg_size_per_room
Yellows	1. Avg_size_per_room 2. Avg_Satisfaction_with_previous_vote 3. Number_of_differnt_parties_voted_for 4. Political_interest_Total_Score 5. Yearly_IncomeK 6. Most_Important_Issue 7. Overall_happiness_score 8. Avg_monthly_income_all_years 9. Weighted_education_rank

Before manipulating the features set and trying to change the existing coalition, we tried to figure out which features separated it from the opposition by the cluster centers:

Feature	Coalition Cluster	Opposition Cluster
Avg_size_per_room	-0.10708	0.158979
Avg_Satisfaction_with_previous_vote	-0.602	0.893795
Number_of_differnt_parties_voted_for	-0.30284	0.449627
Political_interest_Total_Score	-0.65106	0.966641
Yearly_IncomeK	0.001642	0.006928
Most_Important_Issue	3.125129	2.713303
Overall_happiness_score	0.010432	-0.01549
Avg_monthly_income_all_years	-0.14508	0.2154
Weighted_education_rank	-0.09109	0.135246

Diagnoses:

- These are the features that make a significant difference between the coalition and the opposition.

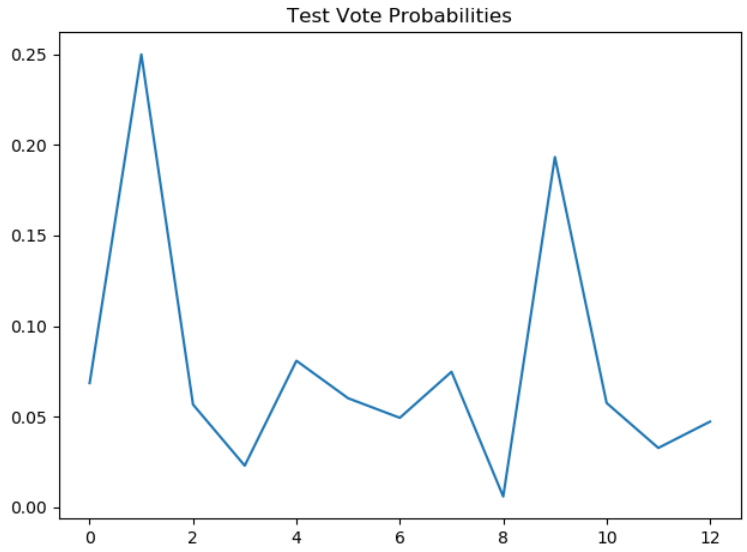
- We have noticed that the feature that does not lead in most coalition parties is **Avg_size_per_room** and so in order to build an alternative coalition using this method we would like to create homogeneity for this feature in the opposition parties.
- Leading feature in the coalition parties and opposition parties is **Number_of_differnt_parties_voted_for.**

<u>Changing the winning party and building an alternative coalition:</u>

In order to change the victorious party we want to manipulate the features: **Avg_monthly_income_all_years**, **Political_interest_Total_Score** which are dominant in the second largest brown party.

In addition, we want changes to form new clusters that will form the basis for an alternative coalition. From the previous exercise we recognized that the top two parties are turquoises and browns, so since browns is not in our coalition we want to strengthen them by manipulating the values(increasing Number_of_differnt_parties_voted_for and Avg_size_per_room values).

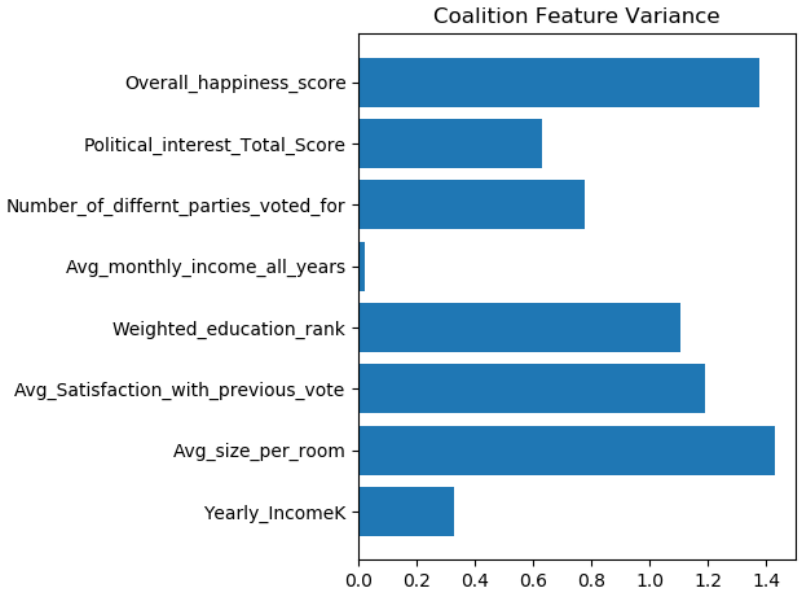
We were able to change the winning party to Browns (1) :



And after our manipulation, the following alternative coalition was received with 61.8% of the vote:

Browns, Pinks, Purples, Turquoises, Whites

And the following graph:



We can see that the homogeneity of this coalition is smaller (variance has increased in most features) than the previous one.

Strengthening the current coalition:

In order to strengthen the coalition, we thought about adding a party outside the coalition to it. To do this, we chose manipulations that would add the khaki party to the coalition, since the traits of its dominance are relatively close to the largest party in the coalition, the Turqouises. These features are Avg_monthly_income_all_years Political_interest_Total_Score, Avg_Satisfaction_with_previous_vote which also creates a distinct cluster separation.

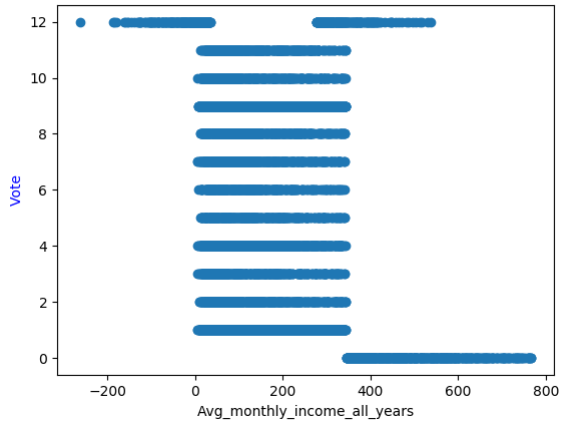
The manipulation we did was to bring these features closer to the cluster center (approximating the values for the cluster expected value) that made up the coalition and in this way prevent these features from separating the clusters and thus letting the other features separating the coalition from the opposition to form the separation of the clusters.

Following our manipulation, the following strengthened coalition was obtained with 62.32% of the vote:

Green, Greys, Khakis, Pinks, Purples, Reds, Turquoises, Violets , Whites.

We can notice that the parties that are left out are: Blues (0), Browns (1) and Yellows(12).

We will explain this using the feature that is important to them which is Avg_monthly_income_all_years which as we saw in Exercise II creates a distinction between Blues (0), Yellows (12) and the other parties and it is the one of dominant features to separate the clusters:



(Browns did not enter the coalition because they were too scattered between the .clusters)