Mori Levinzon 308328467
Nadav Rubinstein 208686659

Mandatory Part – Loading and Preparing the Data:

First, as requested, we loaded the sample file we received at the beginning of the course again, and applied the preliminary manipulations to prepare the information as we did in the previous exercises, these manipulations include:

1. Divide the total sample set (on the Stratified Shuffle Split) into three sets:

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

2. Removing values that are outliers, like negative values.
3. Completion of missing values according to common methods: feature correlation, mean and majority.
4. Selecting the Right Feature Set as Selected in Exercise # 3.
5. Normalize to categorical values and Z-scale to nominal values.
6. Export the information to 3X2 CSV files (3 before and 3 after the change).

As part of adapting the new test set in this exercise, we performed the exact same manipulations we performed on the training set, on the unlabeled test set, in order for our classifier to deal with the test set as we did throughout the semester.

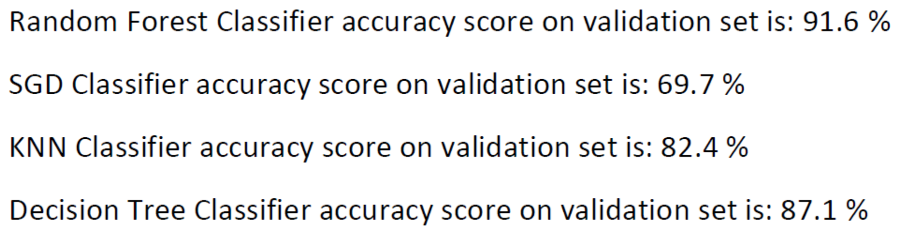
Then, we approached the prediction task when we were required to predict:

- Who is the winning party according to the test set.
- What is the votes split according to the test set.
- For each voter from the test set, predict the vote.
- Predict a stable and homogeneous coalition that includes at least 51% of all votes.

Mandatory Part – Voting Predictions:

In order to make the predictions above on the unlabeled test set we must build a high generalizability classifier that will result in high accuracy because we cannot evaluate the classifier performance on the unlabeled test set.

In HW3 we did similar tasks and then we got the following results:



```
Random Forest Classifier accuracy score on validation set is: 91.6 %

SGD Classifier accuracy score on validation set is: 69.7 %

KNN Classifier accuracy score on validation set is: 82.4 %

Decision Tree Classifier accuracy score on validation set is: 87.1 %
```

Because we wanted to achieve a classifier with 90-95% accuracy, we came to the conclusion that we needed to improve our classification ability.

In order to do to put together a classifier with good generalizability, we decided to form a committee that includes the three most prominent classifiers we saw during the course, and this committee will include three classifiers and will consist of Random Forest, Multi-Level Perception, SVM.

We chose these three categories because they stood out for their inclusion ability and good performance as we saw in the course.

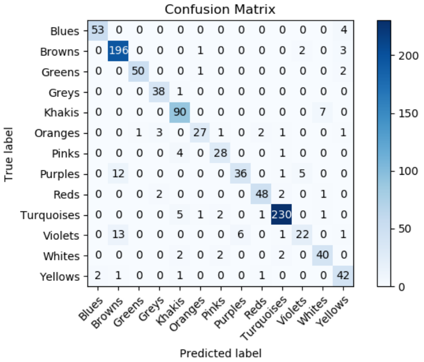
Because in previous assignments we only used Random Forest we had to adjust the parameters of the other two and in addition we decided to try to find better parameters for the Random Forest as well.

Finding the best parameters for each classifier was performed as follows: For each classifier we performed Random Search on a selection of parameter sets when each set of parameters was evaluated by K-fold cross validation.

The motivation to do this is to maximize the inclusiveness of each of the classifiers on the committee and thus create a stronger overall committee than any of the classifiers themselves.

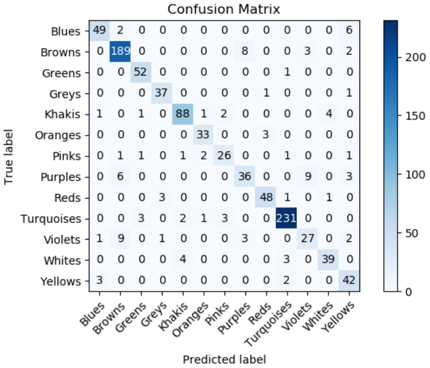
The performance examination of each of the classifiers with hyperparameters found:

Random Forest - yielded 90.0% accuracy on the validation set.



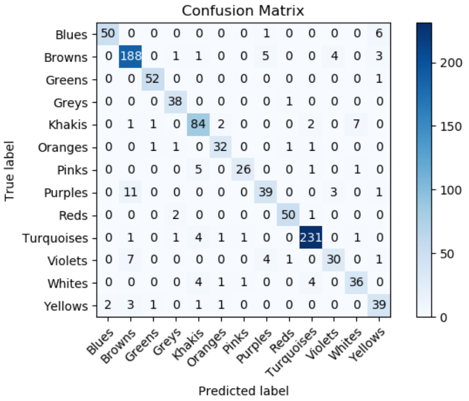
As we can see, the purples and violets tend to be classified as Browns.

MLP - yielded 89.7% accuracy on the validation set.



As you can see, the violets Party tends to be classified as the Brown Party.

SVM - yielded 89.5% accuracy on the validation set.



As you can see, the Purples and Violets parties tend to be classified as the Brown Party and tends to classify the Khakis Party as the Whites Party.

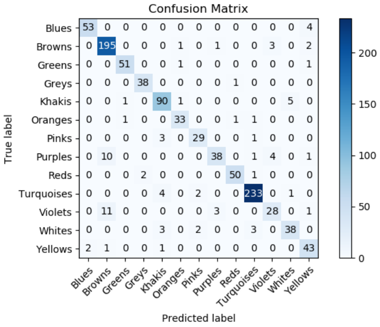
After finding the hyperparameters for each of the classifiers, we set the parameters for each classifier, wrapped all of them into a one classifier, which, when training, trains all classifiers within it and, when predicting, will perform the prediction by applying some manipulation to the predictions produced by each classifier.

Now we have to figure out how to tune the classifier to make predictions, we thought of two ways:

1. When there is majority, the prediction would be decided by the majority of the committee. when all the predictions of all classifiers differs from one another, a master classifier will take the decision.
2. Predict the probability for each prediction, find the probability for each prediction and finally decide on the prediction with the greatest probability.

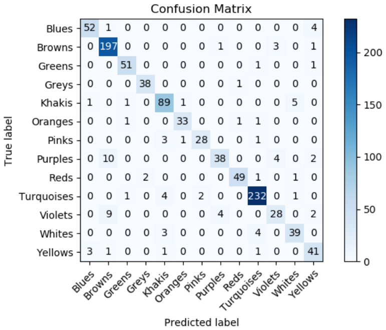
Choosing the way to predict:

First we tried the 1 way option with the classifier that decides when there is no majority is the Random Forest, the accuracy on the validation set was 91.9%.



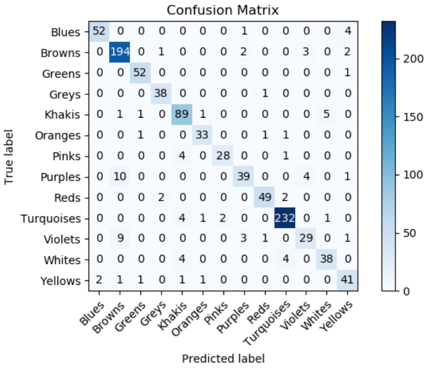
	Blues	Browns	Greens	Greys	Khakis	Oranges	Pinks	Purples	Reds	Turquoises	Violets	Whites	Yellows
Blues	53	0	0	0	0	0	0	0	0	0	0	0	4
Browns	0	195	0	0	0	1	0	1	0	0	3	0	2
Greens	0	0	51	0	0	1	0	0	0	0	0	0	1
Greys	0	0	0	38	0	0	0	0	1	0	0	0	0
Khakis	0	0	1	0	90	1	0	0	0	0	0	5	0
Oranges	0	0	1	0	0	33	0	0	1	1	0	0	0
Pinks	0	0	0	0	3	0	29	0	0	1	0	0	0
Purples	0	10	0	0	0	0	0	38	0	1	4	0	1
Reds	0	0	0	2	0	0	0	0	50	1	0	0	0
Turquoises	0	0	0	0	4	0	2	0	0	233	0	1	0
Violets	0	11	0	0	0	0	0	3	0	0	28	0	1
Whites	0	0	0	0	3	0	2	0	0	3	0	38	0
Yellows	2	1	0	0	1	0	0	0	0	0	0	0	43

When the classifier that decides when there is no majority is the MLP, the accuracy of the committee on the validation set was 91.5%.

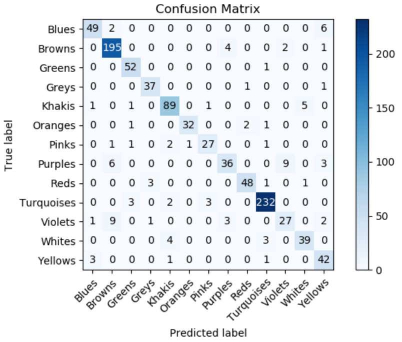


	Blues	Browns	Greens	Greys	Khakis	Oranges	Pinks	Purples	Reds	Turquoises	Violets	Whites	Yellows
Blues	52	1	0	0	0	0	0	0	0	0	0	0	4
Browns	0	197	0	0	0	0	0	1	0	0	3	0	1
Greens	0	0	51	0	0	0	0	0	0	1	0	0	1
Greys	0	0	0	38	0	0	0	0	1	0	0	0	0
Khakis	1	0	1	0	89	1	0	0	0	0	0	5	0
Oranges	0	0	1	0	0	33	0	0	1	1	0	0	0
Pinks	0	0	0	0	3	1	28	0	0	1	0	0	0
Purples	0	10	0	0	0	0	0	38	0	0	4	0	2
Reds	0	0	0	2	0	0	0	0	49	1	0	1	0
Turquoises	0	0	1	0	4	0	2	0	0	232	0	1	0
Violets	0	9	0	0	0	0	0	4	0	0	28	0	2
Whites	0	0	0	0	3	0	0	0	0	4	0	39	0
Yellows	3	1	0	0	1	0	0	0	0	1	0	0	41

When the classifier that decides when there is no majority is the SVM, the accuracy of the committee on the validation set was 91.4%.



<u>Second way</u>: Making the prediction by weighting the probabilities produced 90.5% accuracy results on the validation set.



Conclusions and choice of way to make a prediction:

- We can see that the mispredictions we saw before for each of the classifiers were reduced.
- All roads have increased the accuracy of all classifiers.
- The inclusion capacity of the committee we assembled is better than any of the independent classifiers.

Finally, we decided to choose option 1 and the classifier that decides the prediction when there is not majority is the Random Forest. Although this method has achieved as high accuracy as the others the classifier itself has a high percentage of accuracy compared to the MLP and SVM and because we do not want to suffer from overfitting or underfitting and that’s why we chose this method.

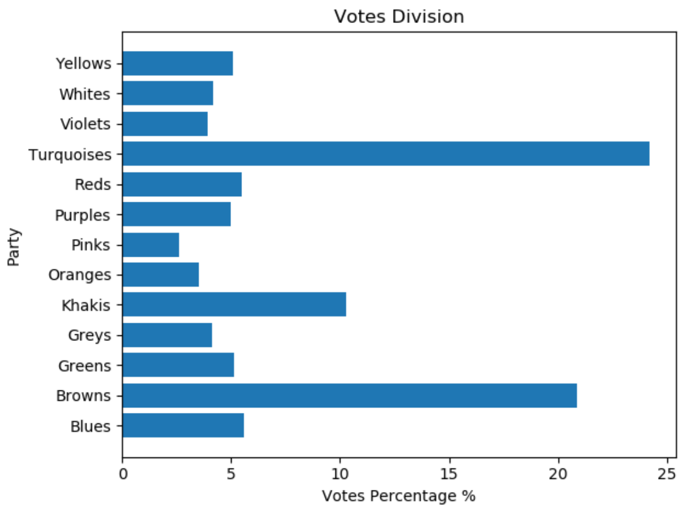
Comparing classifier results on labeled test set from the training set to the unlabeled test set:

Labeled test set:

For a labeld test set that the overall classifier did not trained at all and did not rely on at all, we were able to reach an accuracy ratio of 92.12%, which is higher than the accuracy percentage of all the classifiers independently.

In order to predict the winning party by this set we predicted all the votes and the winning party that has the most votes, according to test set (from the total training set) the winning party is the turquoises.

Distribution of votes according to this test set:



Color	Vote Percentage
Blues	5.60%
Browns	20.88%
Greens	5.12%
Greys	4.12%
Khakis	10.28%
Oranges	3.52%
Pinks	2.64%
Purples	%4.96
Reds	5.48%
Turquoises	24.2%
Violets	3.92%
Whites	4.2%
Yellows	5.08%

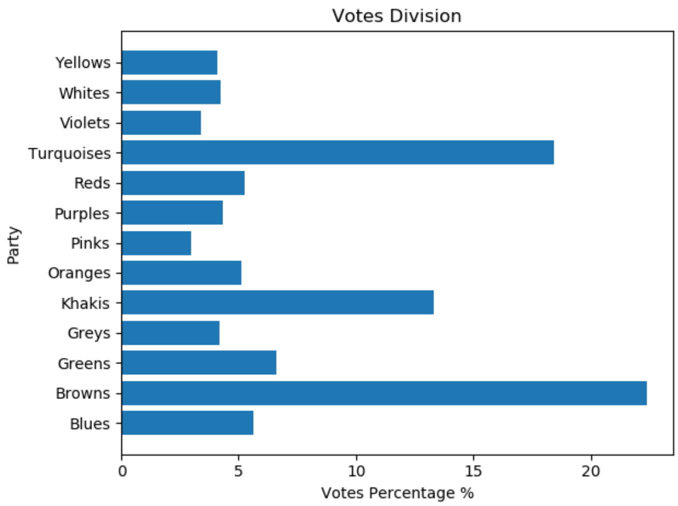
Unlabeled test set (these predictions are the predictions for submission):

Now that we have an Unlabeled test set, we have re-trained the classifier on all the sample set from the beginning of the course (including the test set we put aside at the beginning of the course) because the test set now does not depend on the training set.

We will note that as we train the classifier across the entire training set, our classifier has now changed, its inclusion ability has changed and to some extent we have lost the performance evaluation we did before.

Admittedly, the motivation to do this is assuming that as our classifier trains more examples then the inclusion capacity increases and thus we can reach a higher accuracy, in addition, when we examined the classifier earlier, we trained it on 6,500 samples and on the test set which has 2,500 samples compared to this unlabeled test set, it contains 10,000 samples, so we would like our classifier to practice on more samples to maintain performance.

In order to predict the winning party by this set, we predicted all the votes. The party with the most votes is the Browns.



Color	Votes	Vote Percentage
Blues	994	5.64%
Browns	**2102**	**22.39%**
Greens	834	6.62%
Greys	576	4.17%
Khakis	93	13.32%
Oranges	734	5.13%
Pinks	853	2.98%
Purples	2113	4.33%
Reds	427	5.27%
Turquoises	335	18.44%
Violets	17	3.38%
Whites	452	4.24%
Yellows	473	4.09%

As we can see from the prediction for the unlabeled test set, the party with the most votes is has changed to the browns. Because, we saw a clear tendency of our classifier to confuse the votes of the brown party with the votes of other parties and so despite the large deviation in favor of the browns, we assume that these results are reliable.

We exported these results to a CSV file called “test_predictions.csv” as required in the exercise.

Building Steady Coalition Using Clustering Model:

To build a coalition with the unlabeled test set, we decided to change approach from HW4 and examine new steps to form the coalition in order to be more compatible with a stable coalition definition.

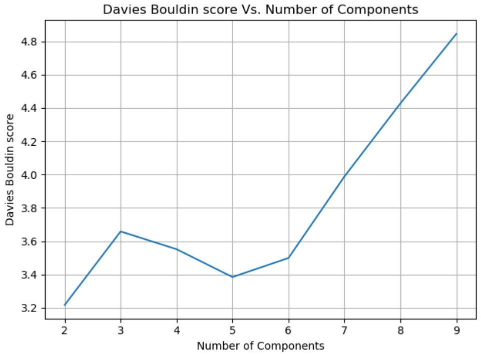
Despite the approach change, we still chose to use the Gaussian Mixture algorithm which is a Clustering model because it yielded good results in a previous assignment. The motivation to use such a model is the ability to identify similarities between the party voters by grouping them into Clusters.

First we had to understand with how many Clusters we had to train our model, this is actually a hyperparameter of the model we had to find. To find the parameter, we trained the algorithm with several different Clusters size and tested its performance. We have trained each model using the whole training set from the beginning of the course, because now it is meaningless for the classification of the unsupervised learning, but it is more meaningful for the amount of examples that the model has to deal with and the quality of the clustering.

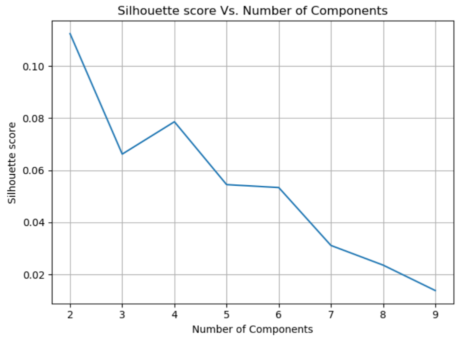
We evaluated the model using the following metrics:

Internal metrics:

Davies Bouldin score - examines the average similarity between the two most similar Clusters, when the similarity is the distance within the Cluster and the distance between them. We would prefer low values.

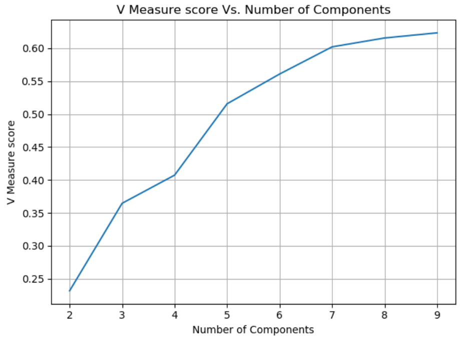


Silhouette score — Tests for each sample how well it fits into the Cluster it belongs to. Its values are between -1 and 1 and we would prefer values closer to 1.



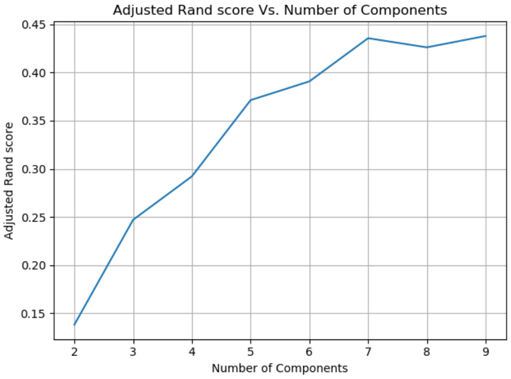
External metrics:

V Measure score - Harmonic average between homogeneity and completeness measures. When the homogeneity measure measures how much each cluster is loyal to a single label. A measure of completeness measures how much each labeling is loyal to a single cluster. The index yields values between 0 and 1 when we would prefer values close to 1.



Stability and ratio metrics:

Adjusted Rand score - Measures the degree of agreement between two runs of the algorithm. Values between -1 and 1 when we would prefer values close to 1.



We decided to build our model with 5 Clusters, because for this number of Clusters we achieved good results for all the metrics.

The coalition was built as follows:

After selecting the Clusters number (5), we retrained the Gaussian Mixture with the untagged test set.

In order to understand the similarities between the parties for each party, we analyzed its split according to the Clusters where the labeling of each sample from the set is actually the prediction of the committee we have built.

Cluster	Blues	Browns	Greens	Greys	Khakis	Oranges	Pinks	Purples	Reds	Turquoises	Violets	Whites	Yellows
0	0	0	641	2	1316	2	102	0	5	1803	0	416	1
1	2	243	21	356	16	511	11	99	466	33	69	8	215
2	551	23	0	0	0	0	0	15	0	3	17	0	177
3	0	1114	0	0	0	0	0	196	0	5	147	0	3
4	11	859	0	59	0	185	0	123	56	0	105	0	13

Party/Cluster	0	1	2	3	4
Blues	0	0.35%	97.70%	0	1.95%
Browns	0	10.85%	1.02%	49.75%	38.36%
Greens	96.82%	3.17%	0	0	0
Greys	0.48%	85.3%	0	0	14.14%
Khakis	98.80%	1.2%	0	0	0
Oranges	0.39%	99.61%	0	0	0
Pinks	34.23%	3.69%	0	0	62.08%
Purples	0	22.86%	3.46%	45.27%	28.4%
Reds	0.95%	88.42%	0	0	10.62%
Turquoises	97.78%	1.79%	0.16%	0.27%	0
Violets	0	20.41%	5.03%	43.49%	31.07%
Whites	98.11%	1.88%	0	0	0
Yellows	0.24%	52.57%	43.27%	0.73%	3.17%

```
cluster 0 distribution: OrderedDict([('Greens', 641), ('Greys', 2), ('Khakis', 1316), ('Oranges', 2), ('Pinks', 102), ('Reds', 5), ('Turquoises', 1803), ('Whites', 416), ('Yellows', 1)])

cluster 1 distribution: OrderedDict([('Blues', 2), ('Browns', 243), ('Greens', 21), ('Greys', 356), ('Khakis', 16), ('Oranges', 511), ('Pinks', 11), ('Purples', 99), ('Reds', 466), ('Turquoises', 33), ('Violets', 69), ('Whites', 8), ('Yellows', 215)])
cluster 2 distribution: OrderedDict([('Blues', 551), ('Browns', 23), ('Purples', 15),
('Turquoises', 3), ('Violets', 17), ('Yellows', 177)])

cluster 3 distribution: OrderedDict([('Browns', 1114), ('Purples', 196), ('Turquoises', 5),
('Violets', 147), ('Yellows', 3)])

cluster 4 distribution: OrderedDict([('Blues', 11), ('Browns', 859), ('Greys', 59), ('Pinks', 185),
('Purples', 123), ('Reds', 56), ('Violets', 105), ('Yellows', 13)])

Greys dist: [(0, 0.004796163069544364), (1, 0.8537170263788969), (4,
0.14148681055155876)]

Khakis dist: [(0, 0.987987987987988), (1, 0.012012012012012012)]

Browns dist: [(1, 0.10853059401518535), (2, 0.010272443054935243), (3,
0.4975435462259937), (4, 0.38365341670388564)]

Turquoises dist: [(0, 0.9777657266811279), (1, 0.017895878524945777), (2,
0.0016268980477223427), (3, 0.0027114967462039045)]

Oranges dist: [(0, 0.003898635477582846), (1, 0.9961013645224172)]

Greens dist: [(0, 0.9682779456193353), (1, 0.03172205438066465)]

Reds dist: [(0, 0.009487666034155597), (1, 0.8842504743833017), (4,
0.1062618595825427)]

Yellows dist: [(0, 0.0024449877750611247), (1, 0.5256723716381418), (2,
0.43276283618581907), (3, 0.007334963325183374), (4, 0.03178484107579462)]

Whites dist: [(0, 0.9811320754716981), (1, 0.018867924528301886)]

Purples dist: [(1, 0.22863741339491916), (2, 0.03464203233256351), (3,
0.45265588914549654), (4, 0.2840646651270208)]

Pinks dist: [(0, 0.3422818791946309), (1, 0.03691275167785235), (4,
0.6208053691275168)]

Blues dist: [(1, 0.0035460992907801418), (2, 0.9769503546099291), (4,
0.019503546099290780)]

Violets dist: [(1, 0.20414201183431951), (2, 0.05029585798816568), (3,
0.4349112426035503), (4, 0.3106508875739645)]
```

We can notice that every party except the brown, pink, purple, violets and Yellow were associated with a single Cluster with at least 85% of the party votes, since this is not a division into 2 Clusters we concluded that the chosen classifier does a very good job.

After looking at the distribution of one of the parties, we grouped them into blocks with similar distribution among the clusters. The motivation for which is to identify similarities between the parties and to form a coalition that is built from parties that has a similarity between the voters. (These blocks are not clusters that the algorithm created but derived from them).

The estimated percent of votes for each of the blocks:

Block 0: Blues, Khakis, Violets, Turquoises, Yellows -> 44.87% of the votes

Block 1: Reds, Oranges, Greys-> 14.57% of the votes

Block 2: Browns, Whites, Purples-> 30.96% of the votes

Block 3: Pink-> 2.98% of the votes

Block 4: Greens-> 6.62% of the votes

We will note that none of the blocks can build a coalition on their own, so we will need to find a link between them to form a coalition.

We will notice that a smaller coalition will have more homogeneity and more stability (as in politics in reality) and so in order to maintain the homogeneity of the coalition, we will try to assemble one with the help of Block 0, which contains the largest number of votes.

We would like to find the block closest to it, for this we have calculated for each block its "center of mass" and the distance from each of the other blocks:

	Block 0	Block 1	Block 2	Block 3	Block 4
Block 0	-	1.13	1.85	1.42	1.24
Block 1	1.13	-	2.28	0.89	1.90
Block 2	1.85	2.28	-	2.49	2.27
Block 3	1.42	0.89	2.49	-	2.03
Block 4	1.24	1.90	2.27	2.03	-

According to this table, block 0 is closest to block 1 and then to block 4.

Since Block 1 has more parties than Block 3 and the difference in proximity is not significant, Block 4 is chosen because it contains fewer parties and upholds better the definition of a stable coalition.

Finally, the coalition we formed is made up of the parties: Blues, Khakis, Violets, Turquoises, Yellows and Greens. The coalition includes 51.49% of the votes.