

Pre Election 2022

Noam Malki

23.8.22

Abstract

This is the time in Israel when the all political parties are getting ready for another elections. Like we didn't have enough in the last 3 years, just kidding.

One of the most important questions is **which party will "pass" the elections threshold?** After that the most important subject in the Israeli Election is **the amount of mandates that party got**. This is important in order to assemble a coalition.

Using explanatory data analysis and clustering, I explored the data of the last two elections (March 2020 and March 2021) in order to get some insight about the participating parties. Afterwards, I tried predict the vote percentage for each party in the last elections using PCA and linear regression. Finally I concluded with the number of mandates each party gets. In addition, I discussed some interesting discoveries and some limitations of this analysis in the end.

Background

Like many democracies, elections in Israel occur every 4 years. In recent 3 years, there is a political crisis in Israel. Meaning, there have been various unsuccessful attempts to assemble a steady government for the last 3 years.

The last attempt was in March 2021, when Naftali Bennet and Yair Lapid became the prime minister and and the "alternative prime minister" respectively. That attempt succeeded at first, but failed recently. In other words: more elections. This is great (not).

The question is what's next? Which party (and which not) will get elected this time? In this essay, I will try to answer the last question.

In order to answer this question, I will use data of the last two elections from the Central Elections Committee website. The data sets contain votes to each participating party in each city in Israel and more details like the numbers of voters in each city, number of invalid voters etc.

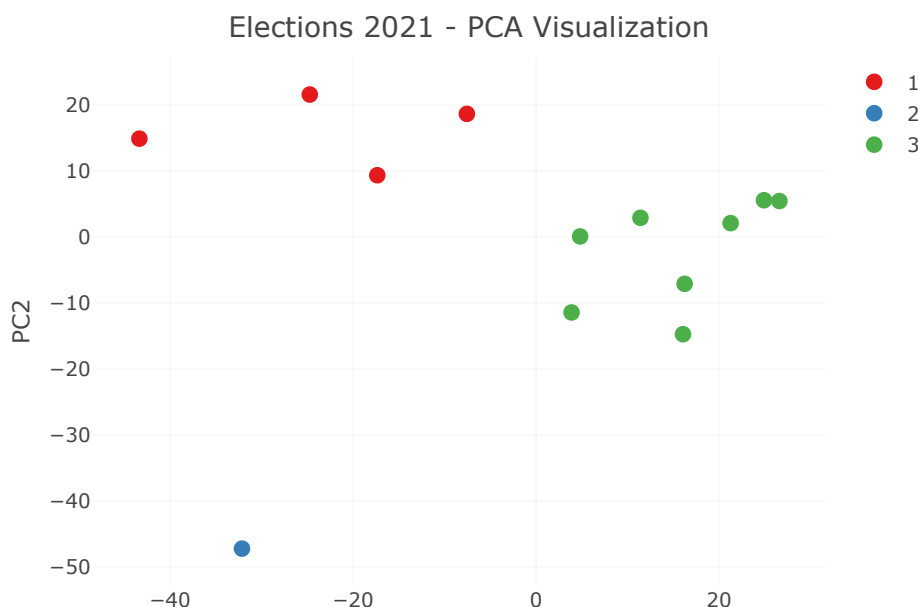
Links to the data will be provided in the reference list.

Exploratory Data Analysis

From the start, we have problem analyzing the data: there are too many variables in the data sets. At first, I focused on votes for "elected parties" in each city. Meaning I focused on the votes for the parties which "passed" the election threshold of 3.25%. I did it with the two data sets.

The remaining data sets contain 8-13 variables each. There are several ways to analyze these amounts of data. I used PCA in order to analyze the data. In order to do PCA on the parties, I made a transpose the data matrix and then applied PCA on it. In addition, I scaled the data for easy-to-analyze results.

The following graph shows the PCA visualization of elected parties in the 2021 elections. I also used the "K-Means" algorithm for clear distinction between the parties.



PC1

As you can see, there are 3 distinctive groups of parties:

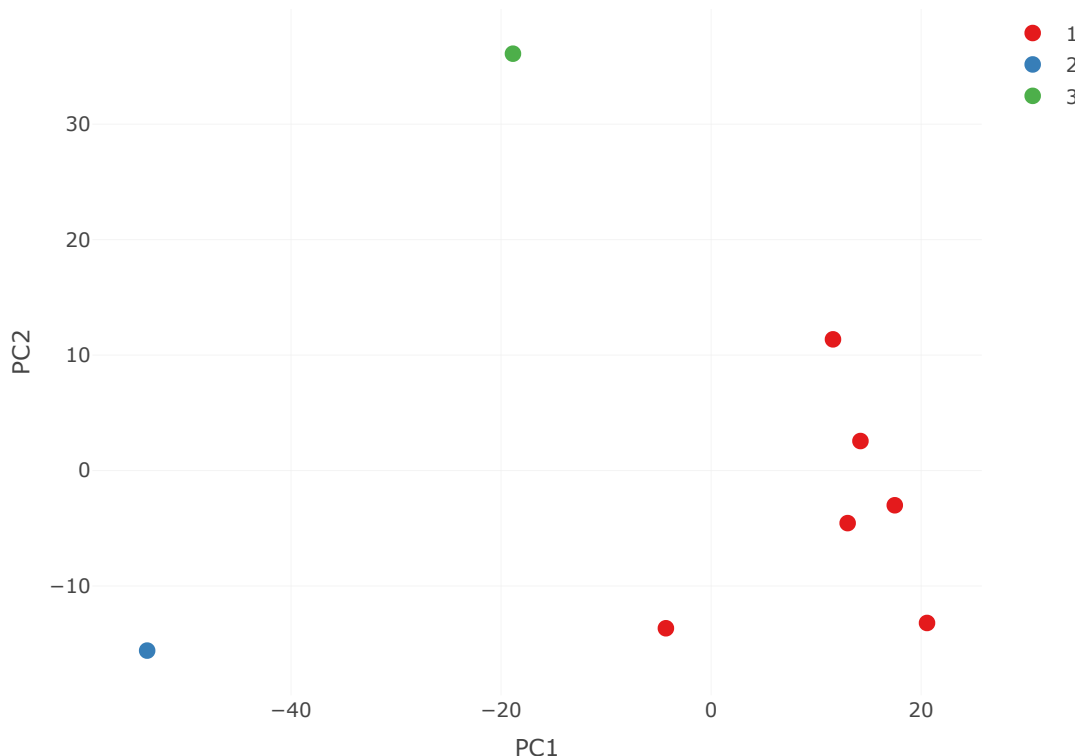
1. Yesh Atid (יש עתיד), Blue and White (כחול לבן), Labor (העבודה) and Meretz (מרצ).
2. Ra'am (רע"ם), Joint List (הרשימה המשותפת), New Hope (תקווה חדשה), Shas (ש"ס), Yamina (ימינה), Yisrael Beiteinu (ישראל ביתנו), Religious Zionist (הציונות הדתית) and UTJ (יהדות התורה).
3. Likud (הליכוד).

This distinction between the groups can tell us about the meaning of the Y-axis (PC2). With the distinction, we can assume that PC2 describes the degree of opposition to Benjamin Netanyahu among the parties' voters. The UTJ party might be an outlier, but it's good to remember that majority of their voters (mostly Ashkenazi ultra-Orthodox jews) doesn't really care about important political issues.

Looking on the X-axis (PC1) in the graph, we can assume that its meaning is how much the parties' voters are homogeneous.

The following graph shows the PCA visualization of elected parties in the 2020 elections. Once again, I used the "K-Means" algorithm for clear distinction between the parties.

Elections 2020 - PCA Visualization



As you can see, the Likud and "Blue and White" parties got their own separate group. In addition, the rest of the parties got the third group.

This distinction between the groups can tell us about the meaning of the X-axis (PC1). It seems that PC1 describes how idealistic the party are or the extent of ideological premise of the party. The Likud party seems like an outlier, but it's good to remember that the party didn't express "pure ideology" recently. The Likud party members have some opinions on political issues, but they "attack the opposition" most of the time. The Blue and White might seem like an outlier, but it's good to remember that party members didn't express their opinions on political issues clearly at that time.

Looking on the Y-axis (PC2) in the graph, we can assume that its meaning is the degree of support in Benjamin Netanyahu within the party.

What's next?

As stated before, Israel are getting ready for another elections. Meaning we once again going to vote and who knows what gonna happen. According to our previous experience, we might find ourselves in another loop of elections (one after another).

So, I want to answer the question: How much votes will each party get with regard to the others?

Methodology

In order to find an answer for the aforementioned question, I used data of the last two elections from the Central Elections Committee website as stated earlier. In addition, I define the vote rate for each party in election 2021 as the response variable.

Denote r_i as the vote rate of party i . r_i is define by this formula:

$$r_i = \frac{\text{all the valid votes party } i}{\text{total valid votes}}$$

However the previous problem still remains: there are too many variables in the data sets. Looking back, we saw that PCA is powerful tool for analyzing a data set with many variables, so we used it.

For prediction of r_i , I used linear regression model because they much easier to compare to one another in regard of other models.

Predictive Analysis

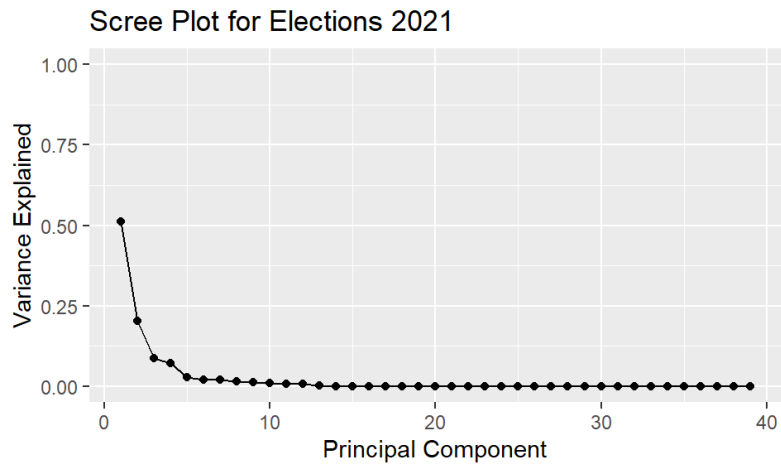
At First, we will build a model data using PCA. Afterwards we will some linear regression models. At last we will choose the best model(s) using the adjusted R^2 measurement.

PCA

Firstly I did PCA for all votes of all the parties in elections 2021. It's worth noting that I scaled the votes before that.

Secondly I calculate the explained variance of the aforementioned PCA. Note that there are 39 principal components, so we have to use only a subset of them.

The following graph is a scree plot which shows how much variance each principal adds to the total.



As you can see, after the tenth principal the line starts to converge. Meaning, at the most 10 principal components save the majority of the data.

The following table shows the cumulative explained variance for the 10 first components:

PC Number	Cumulative Variance
1	0.512
2	0.717
3	0.803
4	0.875
5	0.904
6	0.925
7	0.945
8	0.960
9	0.972
10	0.982

As you can see, the tenth component added only 1% for the cumulative explained variance. So, 9 PCs (with 97.2% explained variance at total) should be enough for us.

Linear Regression

As stated earlier, I will use linear regression model to predict the vote rate.

I will use the 2021 elections' PCA scores as variables. It's worth noting that this data has 39 observations (one for each party), so it should be enough for a linear regression (it's a rule of thumb).

Initial Model

For a start, I fitted the model using all the variables. The summary of the model is as follows:

```
##
## Call:
## lm(formula = y ~ ., data = model_data_2021)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.0088178 -0.0004049 -0.0003778 -0.0002388  0.0189137
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02564103  0.00066978  38.283 < 2e-16 ***
## PC1          -0.00166565  0.00002721 -61.220 < 2e-16 ***
## PC2          -0.00113220  0.00004309 -26.273 < 2e-16 ***
## PC3           0.00099260  0.00006606  15.026 3.21e-15 ***
## PC4          -0.00002788  0.00007280  -0.383  0.70452
## PC5          -0.00034646  0.00011356  -3.051  0.00484 **
## PC6          -0.00100859  0.00013478  -7.483 3.01e-08 ***
## PC7           0.00086923  0.00013781   6.307 6.88e-07 ***
## PC8          -0.00171594  0.00015980 -10.738 1.28e-11 ***
## PC9          -0.00090216  0.00017595  -5.127 1.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.004183 on 29 degrees of freedom
## Multiple R-squared:  0.9941, Adjusted R-squared:  0.9923
## F-statistic: 545.6 on 9 and 29 DF, p-value: < 2.2e-16
```

As you can see the Adjusted R^2 is 0.9923, which means that the model explains the variance very well (99.23%). On the other hand, PC4 didn't turned out significant.

The model is good, but maybe we can do better. We can remove the insignificant variable or try to find an interaction between two variables (There are more options that we will discuss later). Let's remove it from the model and see how it influences the model (it makes sense because there is no interpretation to variables):

```
##
## Call:
## lm(formula = y ~ PC1 + PC2 + PC3 + PC5 + PC6 + PC7 + PC8 + PC9,
##     data = model_data_2021)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.0088642 -0.0004879 -0.0004606 -0.0003912  0.0187936
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02564103  0.00066019  38.839 < 2e-16 ***
## PC1         -0.00166565  0.00002682 -62.110 < 2e-16 ***
## PC2         -0.00113220  0.00004248 -26.654 < 2e-16 ***
## PC3          0.00099260  0.00006511  15.245 1.14e-15 ***
## PC5         -0.00034646  0.00011193  -3.095  0.00424 **
## PC6         -0.00100859  0.00013285  -7.592 1.82e-08 ***
## PC7          0.00086923  0.00013584   6.399 4.60e-07 ***
## PC8         -0.00171594  0.00015751 -10.894 6.00e-12 ***
## PC9         -0.00090216  0.00017343  -5.202 1.32e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.004123 on 30 degrees of freedom
## Multiple R-squared:  0.9941, Adjusted R-squared:  0.9925
## F-statistic: 631.8 on 8 and 30 DF, p-value: < 2.2e-16
```

As you can see the adjusted R^2 has gone up to 0.9925, which is excellent! In addition, the removal of PC4 from the model didn't affect the other variables.

In comparison to the previous model, this sub model explains the response variable much better!

Let's evaluate this sub model's errors. I'll use MSE, RMSE and MAE measurements.

```
## [1] "The MSE of the model is 0.00001"
```

```
## [1] "The RMSE of the model is 0.00362"
```

```
## [1] "The MAE of the model is 0.0016"
```

All the aforementioned measurements are very low, which means the error of the model is very low!

In other words the initial sub model predicts r_i very well.

Advanced Models

Next, I fitted a linear regression model using the 2021 elections' PCA scores and the 2020 elections' PCA scores of the elected parties. I took the 8 PCA scores from the initial model and combine them with the the first 8 principal components of election 2020. The idea behind this combination is that the 2020 elections influenced the 2021 elections.

There are some parties which split prior the elections like "Blue and White" and "Yesh Atid", "New Hope" and "Likud" etc. This is can be problem because we can't merge the 2020 data with the 2021 data. In order to overcome it, I did the following:

1. Duplicated the rows of relevant parties.
2. Attributed the duplicates to the new parties in elections 2021.
3. Took the average of the old and new rows.

In the end, we got a new data set with 39 observations and 8 explanatory variables.

Firstly, I fitted a model with all the variables. The summary of the model is as follows:

```
##
## Call:
## lm(formula = y ~ ., data = total_model_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.034658 -0.000965 -0.000874 -0.000733  0.036769
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0358350  0.0017548  20.421  < 2e-16 ***
## x1          -0.0025418  0.0001011 -25.141  < 2e-16 ***
## x2          -0.0036510  0.0002867 -12.736  1.24e-13 ***
## x3           0.0024196  0.0004675   5.176  1.42e-05 ***
## x4          -0.0026022  0.0005162  -5.041  2.07e-05 ***
## x5          -0.0043134  0.0007005  -6.158  8.97e-07 ***
## x6          -0.0001831  0.0005701  -0.321  0.750279
## x7           0.0035080  0.0008414   4.169  0.000239 ***
## x8           0.0009154  0.0009067   1.010  0.320759
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01063 on 30 degrees of freedom
## Multiple R-squared:  0.9607, Adjusted R-squared:  0.9503
## F-statistic: 91.79 on 8 and 30 DF,  p-value: < 2.2e-16
```

As you can see the adjusted R^2 is 0.9503, which means that the model explains the variance very well (95.03%). On the other hand, PC6 (x_6) and PC8 (x_8) didn't turned out significant.

In order to improve it, we can remove the aforementioned variables or to remove outliers. Because it's small data set, removing outliers is not an option. So we will remove the aforementioned variables. This raises the question: Which one do we remove? Or should we remove both of them? I considered all the options as follows:

Sub model without PC8

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7, data = total_model_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.034572 -0.000992 -0.000815 -0.000729  0.035438
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03582802  0.00175532  20.411  < 2e-16 ***
## x1          -0.00252234  0.00009927 -25.408  < 2e-16 ***
## x2          -0.00365138  0.00028675 -12.733  7.47e-14 ***
## x3           0.00238435  0.00046631   5.113  1.55e-05 ***
## x4          -0.00252598  0.00051077  -4.945  2.51e-05 ***
## x5          -0.00409532  0.00066652  -6.144  8.14e-07 ***
## x6          -0.00012924  0.00056778  -0.228  0.821428
## x7           0.00337156  0.00083078   4.058  0.000311 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01064 on 31 degrees of freedom
## Multiple R-squared:  0.9594, Adjusted R-squared:  0.9502
## F-statistic: 104.7 on 7 and 31 DF,  p-value: < 2.2e-16
```

Sub model without PC6

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x7 + x8, data = total_model_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.034911 -0.001022 -0.000887 -0.000796  0.036990
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0358102  0.0017275  20.729 < 2e-16 ***
## x1          -0.0025410  0.0000996 -25.513 < 2e-16 ***
## x2          -0.0036628  0.0002801 -13.075 3.73e-14 ***
## x3           0.0023181  0.0003394   6.830 1.18e-07 ***
## x4          -0.0025979  0.0005085  -5.109 1.57e-05 ***
## x5          -0.0042705  0.0006776  -6.303 5.20e-07 ***
## x7           0.0034709  0.0008213   4.226 0.000194 ***
## x8           0.0008881  0.0008896   0.998 0.325813
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01048 on 31 degrees of freedom
## Multiple R-squared:  0.9606, Adjusted R-squared:  0.9517
## F-statistic: 108 on 7 and 31 DF, p-value: < 2.2e-16
```

Sub model without PC6 and PC8

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x7, data = total_model_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.034753 -0.000999 -0.000824 -0.000741  0.035623
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03581051  0.00172745  20.730 < 2e-16 ***
## x1          -0.00252216  0.00009779 -25.792 < 2e-16 ***
## x2          -0.00365980  0.00028011 -13.065 2.26e-14 ***
## x3           0.00231280  0.00033931   6.816 1.05e-07 ***
## x4          -0.00252455  0.00050311  -5.018 1.89e-05 ***
## x5          -0.00406938  0.00064691  -6.291 4.69e-07 ***
## x7           0.00334802  0.00081201   4.123 0.000247 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01048 on 32 degrees of freedom
## Multiple R-squared:  0.9593, Adjusted R-squared:  0.9517
## F-statistic: 125.9 on 6 and 32 DF, p-value: < 2.2e-16
```

Only the model without PC6 and the model without both of them has improved regarding their adjusted R^2 .

Let's evaluate this sub models' errors:

	Full model	Sub model without PC8	Sub model without PC6	Sub model without PC6 and PC8
MSE	0.00009	0.00009	0.00009	0.00009
RMSE	0.00933	0.00948	0.00934	0.00949
MAE	0.00433	0.00466	0.00438	0.00464

Despite the improvement in the adjusted R^2 , the second and the third sub models perform worse than the full model in regard of the error measurements. Although the MSE is equal for all models, the RMSE and MAE are higher for all sub models. Meaning the full model is preferable. The only sub model that I could consider is without PC6, because its RMSE and MAE are the closest to the

measurements of the full model. Because of that, I considered it and decided to not reject it.

Predictions

So, we have 3 validated models: the initial sub model, the advanced full model and its sub model without PC6. Now we want to predict r_i and compare the predictions between the models, but we have a problem: we don't have new data to predict it. We used the relevant data (2021 elections data) to build our model. If we use it to predict r_i , we might have an overfitting in our results. I could split the data for training set and testing set, but because we have datasets with low number of observations I abandoned that idea.

In order to overcome this problem, I used the "eligible votes" data. In each city in Israel, there are "eligible voters". These are the people who have the suffrage in the elections. Not all of them will vote (of course), but they have the right to vote. The "eligible votes" data is the votes party each party if every eligible voter would actually vote.

This is how I derived this data from the existing data set:

Denote v_{ij} as the valid votes in city i for party j , so $\sum_j v_{ij}$ will give us the total votes in city i . Denote p_{ij} as the the vote rate of party j in city i . We can derive p_{ij} as follows:

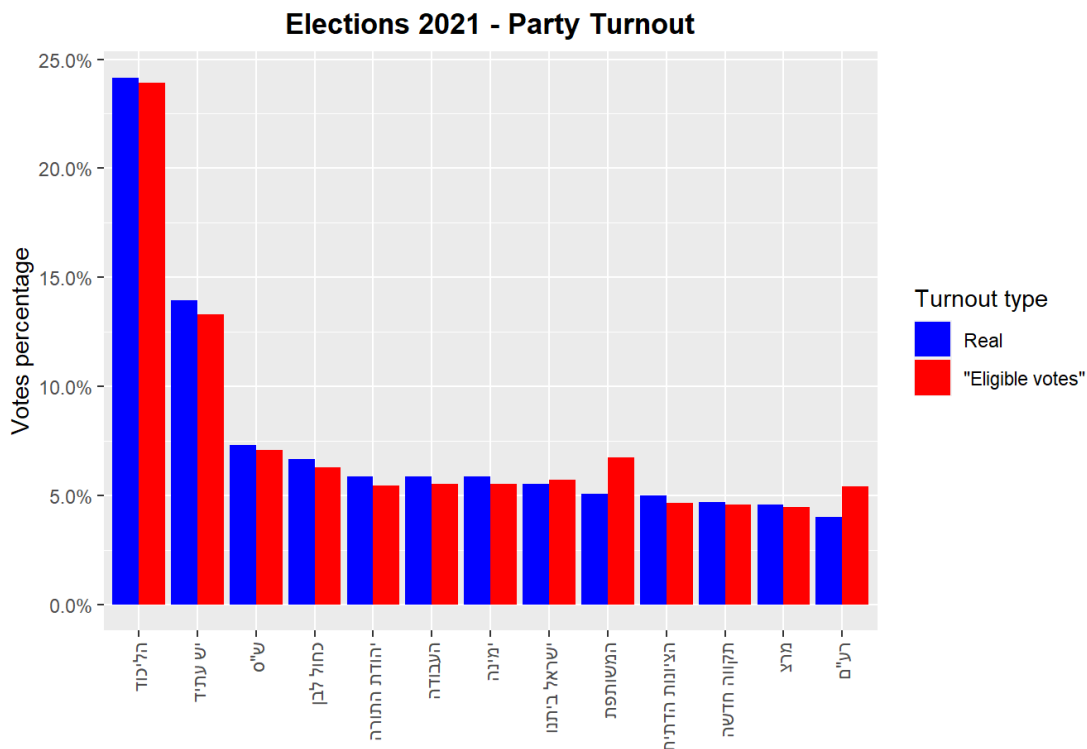
$$p_{ij} = \frac{v_{ij}}{\sum_j v_{ij}}$$

Denote b_i as the number of "eligible voters" in city i . In addition, denote u_{ij} as the valid votes in city i for party j if every eligible voter would actually vote. Therefore we can derive u_{ij} as follows:

$$u_{ij} = b_i \cdot p_{ij} = b_i \cdot \frac{v_{ij}}{\sum_j v_{ij}} \quad \forall i, j$$

After I made a new data set of votes according to aforementioned equations, I rounded the votes in order to convert them to integers. One question still remains: Is this new data set different and realistic in the same time?

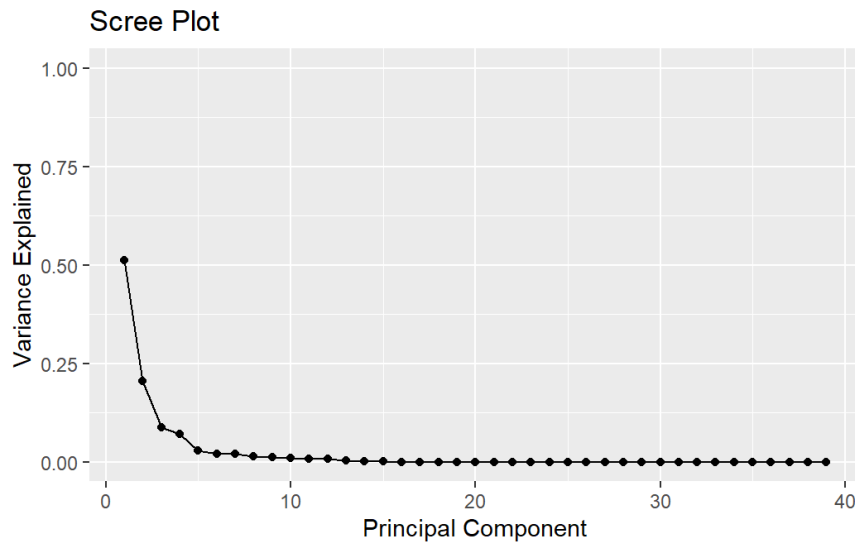
Let's compare between the party turnouts for each data set:



As you can see, there are some different between the turnouts: Yisrael Beiteinu, the Joint List and Ra'am would get more votes if all the eligible voter actually votes. There difference in in Yisrael Beiteinu's turnout somewhat surprising. We can assume that it's due to complacent in the elections. However, the differences in Ra'am and the Joint List are expected because of the low vote percents in the Arab sector.

So, the "eligible votes" data set is realistic and different in the same time. Therefore we will use it to predict r_i . Similar to the "real votes" data set, I did PCA to the data set.

The following scree plot describes the explained variance of the aforementioned PCA:



It seems that after the tenth principal the line starts to converge. Because all our models consists of 8 variables at most, it is important to check whether the first 8 components have a lot of explained variance. Also, because the initial sub model has the ninth PC as a variable, we should check this component's variance

The following table shows the cumulative explained variance for the 9 first components:

PC Number	Variance	Cumulative Variance
1	0.512	0.512
2	0.204	0.717
3	0.087	0.803
4	0.072	0.875
5	0.029	0.904
6	0.021	0.925
7	0.020	0.945
8	0.015	0.960
9	0.012	0.972

As you can see, we have 96% cumulative explained variance with 8 PCs at most (this is very good). Without PC4, we have 90% cumulative explained variance (still it's very good). Remind you that PC4 has been removed as variable from the initial model. The next thing to do is making the predictions of r_i .

After I made the prediction, I encountered a problem: some of the r_i turned out negative. Negative predictions aren't making sense (vote rate should be positive), so I transformed it by applying the absolute value function. If we denote \hat{y}_i and \hat{r}_i as the regression prediction and the estimated vote rate (respectively) for party i , we get:

$$\hat{r}_i = |\hat{y}_i|$$

I will check the error measurements for \hat{y}_i in order to determine whether the transformation was justified

The following tables contain evaluation the models performance using the aforementioned error measurements:

Initial sub model evaluation:

	Original Predictions	Transfromed Predictions
MSE	0.00003	0.00003
RMSE	0.00559	0.00559
MAE	0.00276	0.00276

Advanced model evaluation:

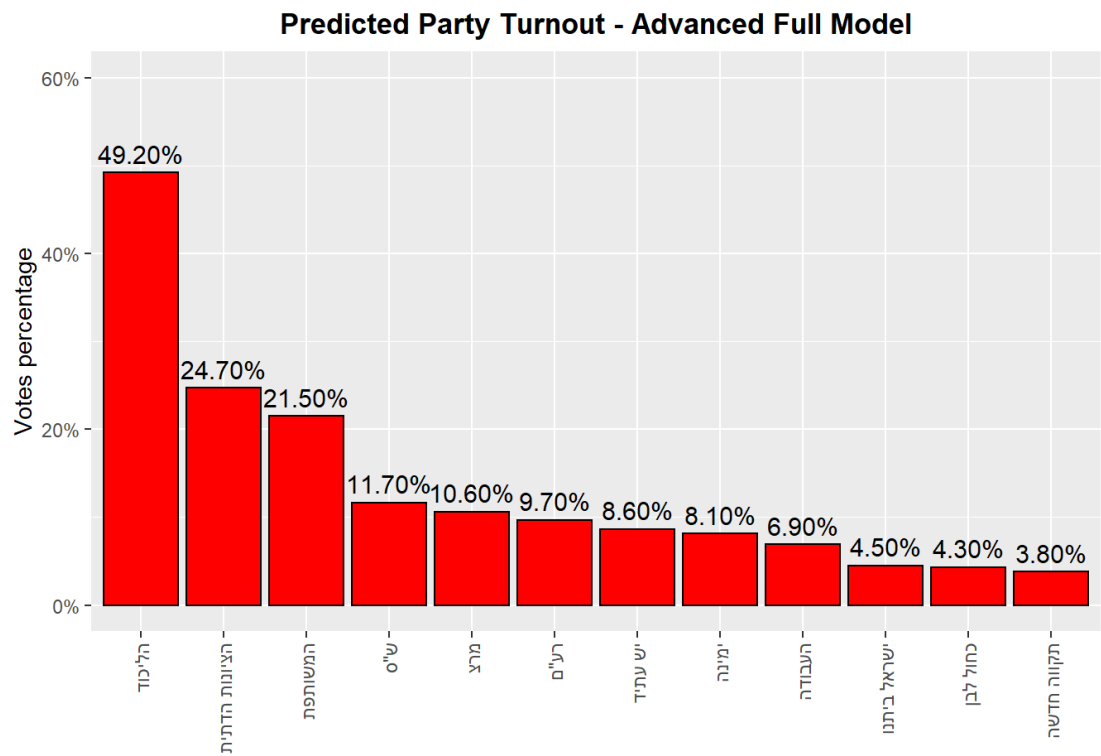
	Original Predictions	Transfomed Predictions
MSE	0.00363	0.00362
RMSE	0.06024	0.06021
MAE	0.03041	0.03020

Advanced sub model evaluation:

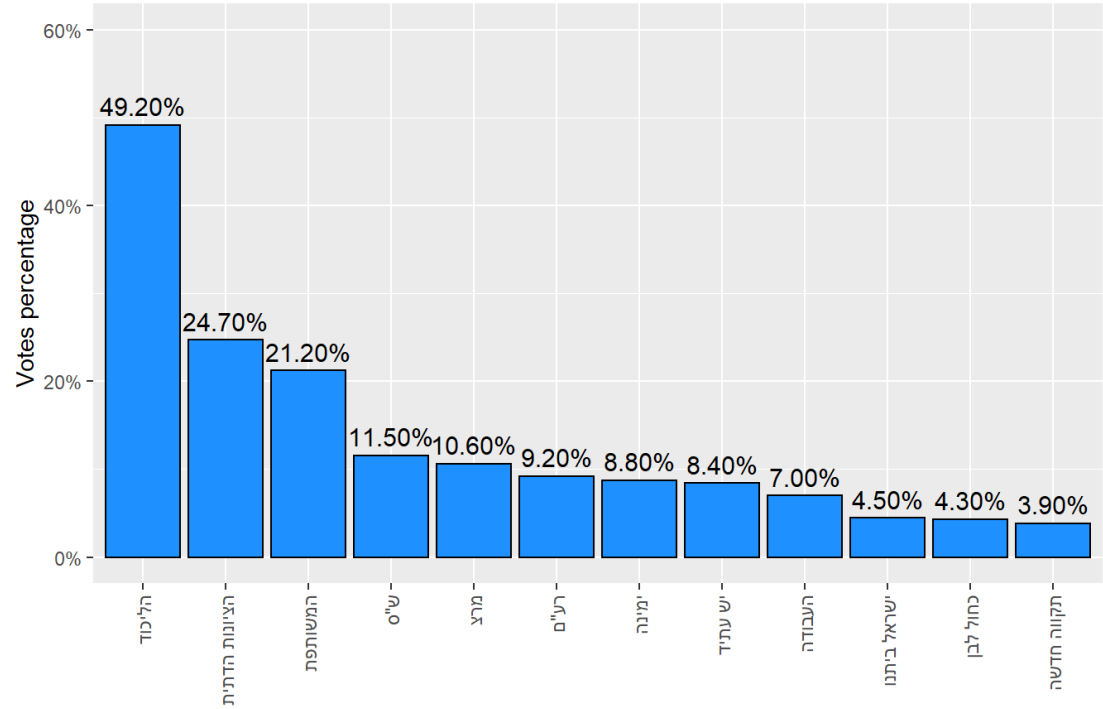
	Original Predictions	Transfomed Predictions
MSE	0.00361	0.00360
RMSE	0.06005	0.06002
MAE	0.03035	0.03017

As you can see, all the measurement are still low for all the models. That is great. Among the models, the initial sub models preformed the best. Also, all the measurements for the transformed predictions are lower or equal to the original. Therefore the transformation was justified. Because all the model preformed well, I chose to keep all of them.

Let's the predictions:



Predicted Party Turnout - Advanced Sub Model



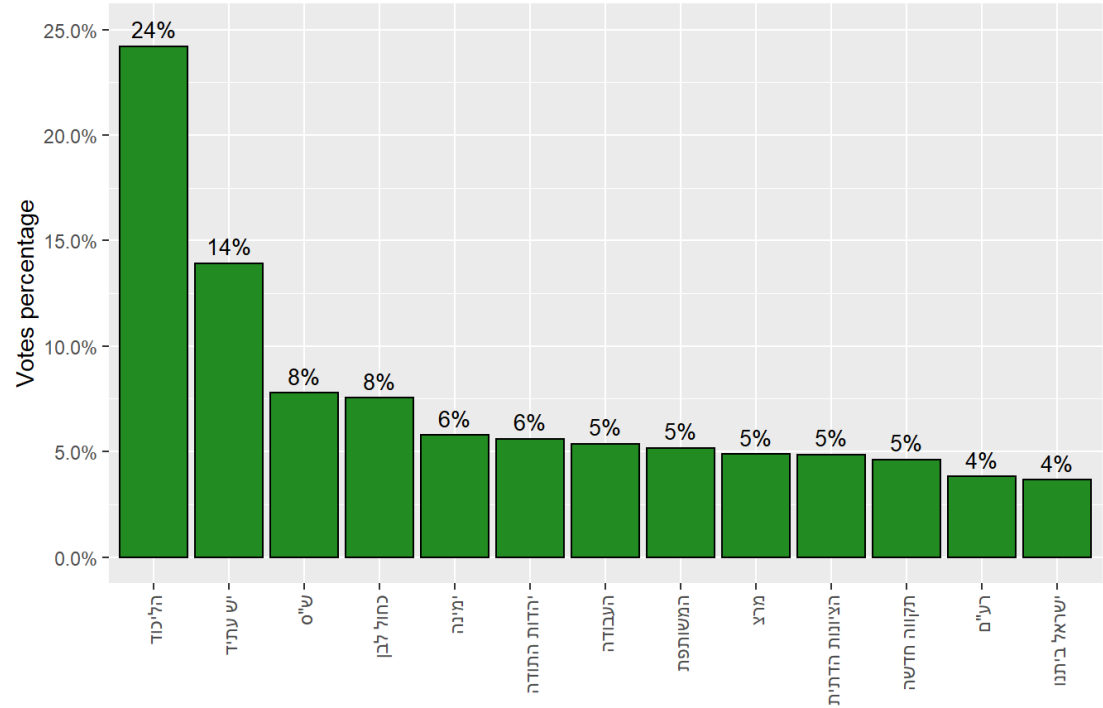
As you can see, the advanced models predicted that:

- The Likud party will get 49% of the votes.
- Religious Zionist party will get more political power.
- Meretz and Ra'am will continue to get political power.
- The UTJ party **will not “pass”** the elections threshold!

The latter is very surprising because the UTJ have a strong base of voters.

Other than that, among the models there is a slight competition between Yamina and Yesh Atid for higher vote percentage.

Predicted Party Turnout - Initial Sub Model



As you can see, the initial sub model predicted more realistic turnout than the other models.

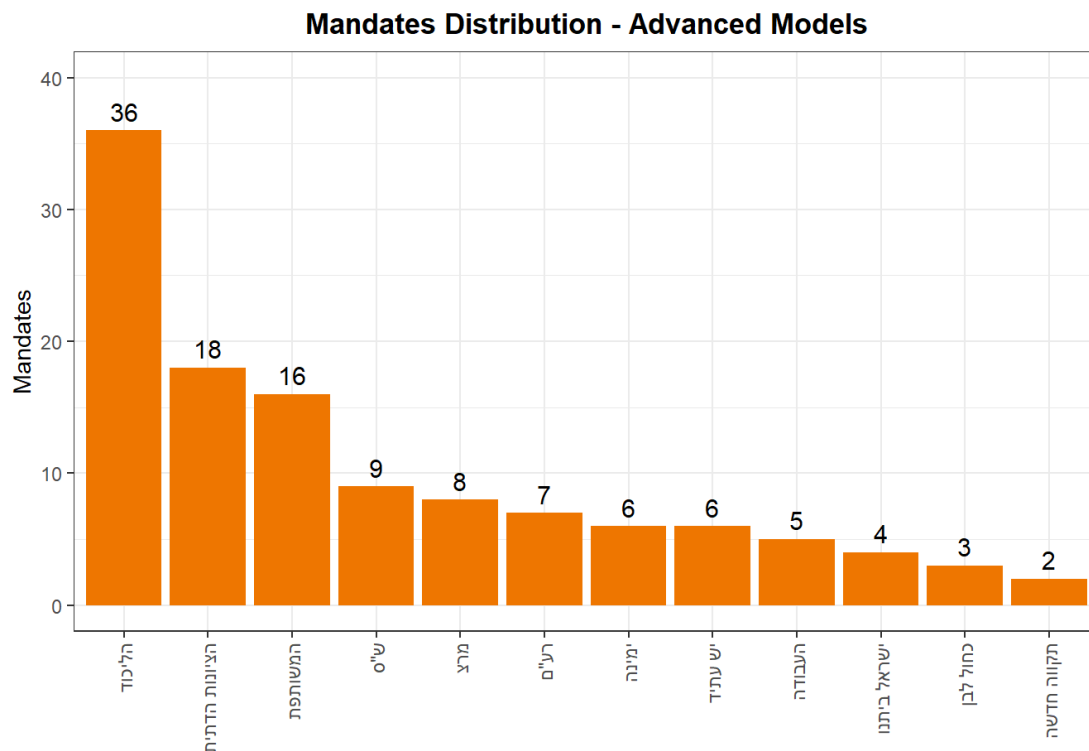
The only surprise here is the decrease of Yisrael Beiteinu's vote percentage.

Mandates distribution:

In this section, I will try to distribute mandates between the parties according to the aforementioned predictions. The distribution was carried out according to the Jefferson's method. Note that we need to use the total number of the "eligible voters", instead of the total number of the valid votes. That's because we used the "eligible voters" data to make the predictions.

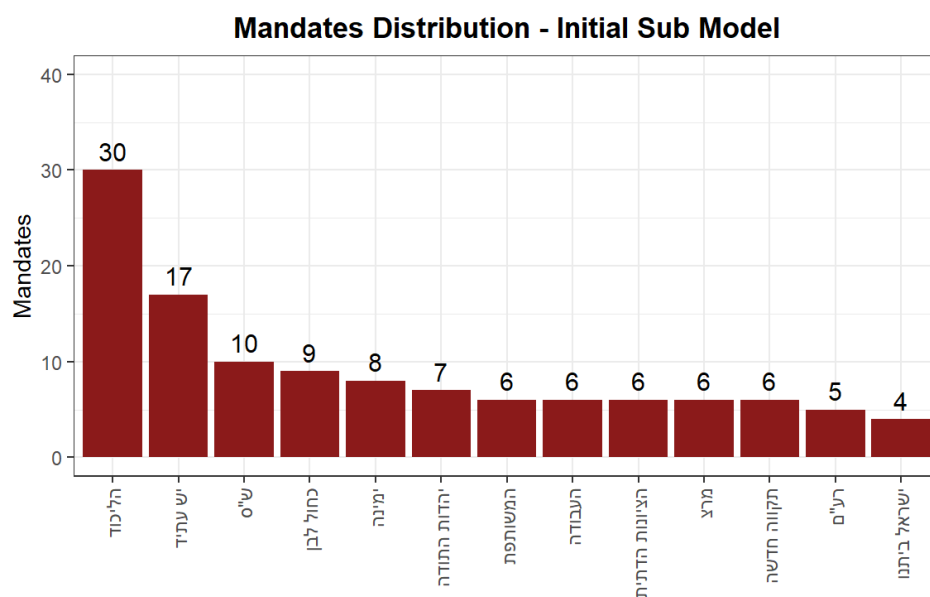
The advanced models' predictions gave the exact same distribution of mandates, so I will refer to only one of them.

The following graph show this distribution:



As you can see, the Likud party got 36 mandates while its opposition got 16 mandates at most. However, Ra'am and Meretz surprised me as they got stronger (politically) with their 7 and 8 mandates (respectively). In this scenario, the Likud can potentially assemble a coalition with Shas and the Religious Zionist party as they would have 63 mandates overall.

But this distribution looks like results of an election poll than a realistic distribution because of the number of mandates that "Blue and White" and "New Hope" got. In Israel, a party that passes the election threshold, gets at least 4 mandates. Therefore it's doesn't make any sense that "Blue and White" and "New Hope" 3 and 2 mandates respectively.



As you can see, this distribution is more realistic than the previous one as it reminds of the results of the 2021 elections. However, Yisrael Beiteinu surprised me as it got weaker (politically) with only 4 mandates. In this scenario, the Likud can't assemble a coalition with only its supporting parties as they would have only 53 mandates overall at most.

Conclusion

In this essay, I tried to predict the result of the upcoming elections using the previous elections data sets. In order to make the predictions, I used the PCA algorithm (for pre-processing the data) and linear regression. I built 6 linear models and chose 3 of them predict the vote rate of each party. Based on my predictions, I distributed mandates to parties using the Jefferson's method. Predictions of only one model (the initial sub model) produced a realistic mandates distribution.

Even though I got nice results, My analysis have some limitations:

Firstly, the model data was small (only 39 observations). This might have affected the predictions. Further researches should find a way to deal with low number of observations. Moreover, there was no option for split the data for training set and testing set because the low number of observations. I solved this problem indirectly. However, further researches should implement properly the "train-test split" in their analysis.

Secondly, maybe linear regression is not best way to predict the vote rate. Further researches should examine more regression models for this particular prediction problem.

Finally, the inclusion of the 2020 elections data in the analysis produced bad results. Further researches should examine whether it's worthwhile to include it the analysis. Alternatively, they should examine ways to integrate it properly.

Discussion

Although we got some results that doesn't make any sense, we can see 2 interesting phenomena from the predictions.

The first one is the fall of UTJ and the rise of the Religious Zionist parties. That's seem odd at first. However, the models confirmed the concerns of the UTJ leaders that their potential young voters will eventually vote to the Religious Zionist party. With the rising popularity of Itamar Ben Gvir, it's no surprise that we see strengthening of the Religious Zionist at other parties' expense.

The second phenomenon is the fall of Yisrael Beiteinu. As we know, Avigdor Liberman was unpopular minister of finance. In addition throughout the years, the party has been associated with criminal affairs such as corruption. Maybe the the model predicted the Yisrael Beiteinu's fall of grace.

All in all, we got interesting phenomena to analyze further. Let's hope that we will have a stable government soon.

References List

1. 2021 Israeli legislative election (in English), Wikipedia (https://en.wikipedia.org/wiki/2021_Israeli_legislative_election)
2. Israeli elections 2021 data, Central Elections Committee website (<https://votes24.bechirov.gov.il/>)
3. Israeli elections 2020 data, Central Elections Committee website (<https://votes23.bechirov.gov.il/>)
4. Jefferson's method, Wikipedia (https://he.wikipedia.org/wiki/%D7%A9%D7%99%D7%98%D7%AA_%D7%92%27%D7%A4%D7%A8%D7%A1%D7%95%D7%9F)
5. "Vote stealing" concerns of the UTJ regarding the Religious Zionist party, ICE news website (<https://www.ice.co.il/elections2022/news/article/878977>)