Itay Israelov
Ilan Coronel

236756 - Introduction to Machine Learning – HW2

Report

Mandatory Assignment Part

Here we will present our process.

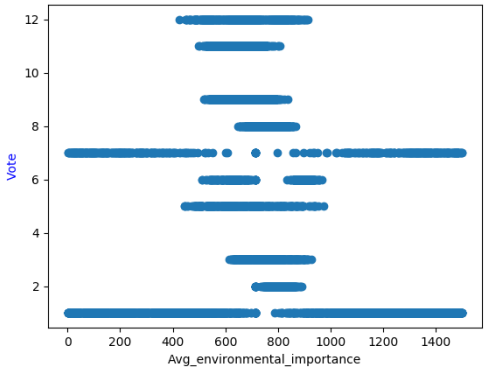
First, we converted every nominal feature to numbers, so we can work with numerical data.
We found out there are 3 types of features:
The nominals, which are strings. The integer features, which contain integers. And the float features, which contain values with decimals.

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.

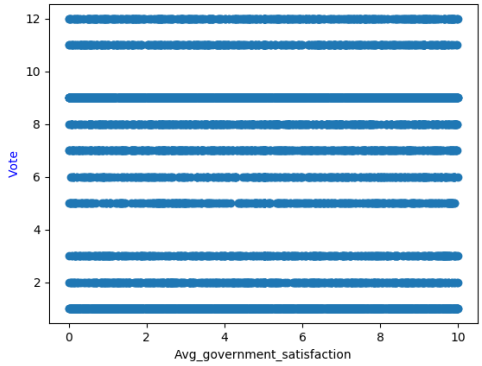
Now, about the imputations:
We decided to use the z score for removing the outliers, as it seems to be a popular way to do so, with a threshold of 3.3.
Also, we noticed that there are 2 parties in the ElectionData which do not exist, so we decided to remove each line containing a non-existent party, since we estimated that non-labelled data can't be correctly exploited (except for missing values, which is already done).

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:



Itay Israelov
Ilan Coronel

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()`.

Next is the feature selection process:

For the filter method, we first used a correlation filter method as seen in class. To do so we computed the correlation matrix and for each pair of highly correlated features, we removed one of them. However, this led to worse results than not using this filter at all, so we decided to not use it. Instead, we used a variance filter with a threshold of 0.05. This removes the features that have a very low variance, therefore not useful.

Now for the wrapper method, we've tested many methods and tried to pick the best.
To test them, we used to KNN classifier to get a score based on the data obtained by filtering with those methods.

We first used the regression model, however for some reasons it gave really poor results (around 68% of accuracy).

Then we used the Relief algorithm implemented in the *Non-Mandatory (Bonus) Assignment*, which gave better results. Indeed, it got us an accuracy of 86.18997%.
However, we decided to try another filter on top of that.
Therefore, we also tested the SFS algorithm implemented in the *Triplets Mandatory Assignment (Bonus for Pairs),* which led to even better results.
Indeed, we managed to get an accuracy of 88.2561%.
We can note that SFS gives better results and is quicker when it is used on top of Relief.

Itay Israelov
Ilan Coronel

After that, instead of using one on top of another, we tried to use both separately, and then filter our features using the union of the two results.
This led to an accuracy of 89.7109%

So eventually, we decided to keep the following features:

Avg_environmental_importance

Avg_size_per_room

Avg_Residancy_Altitude

Avg_education_importance

Avg_Satisfaction_with_previous_vote

Avg_monthly_household_cost

Phone_minutes_10_years

Weighted_education_rank

Last_school_grades

Most_Important_Issue

Number_of_differnt_parties_voted_for

Political_interest_Total_Score

Overall_happiness_score

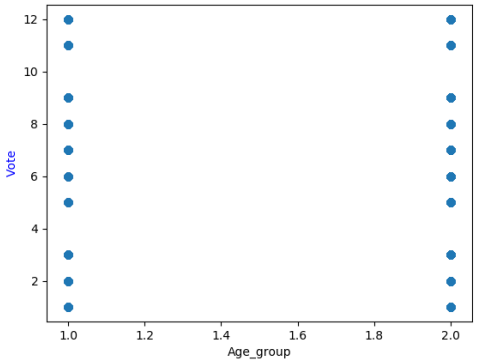
For a final accuracy of 89.7109% using the KNN classifier (with a parameter of 5 neighbors).

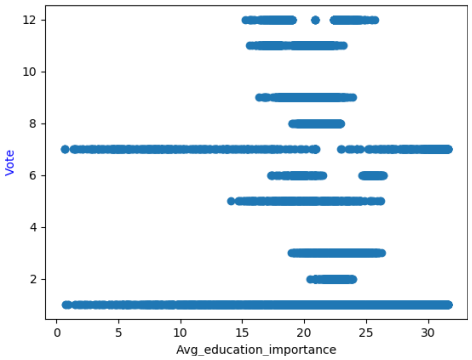
Itay Israelov
Ilan Coronel

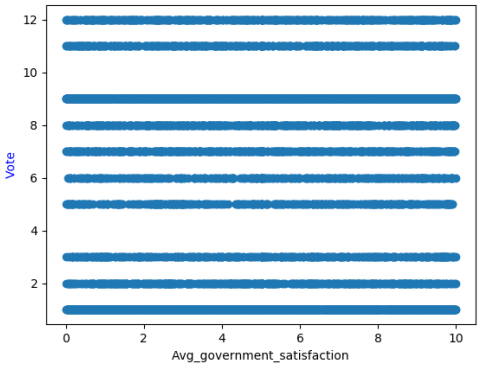
Non-Mandatory Assignments

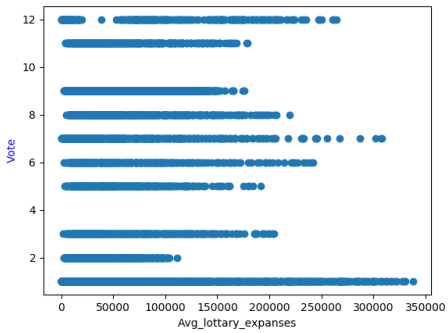
A.

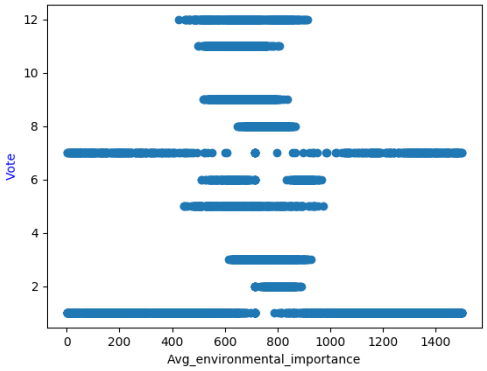
- ***Relation or lack of relation between the features and the labels features:***

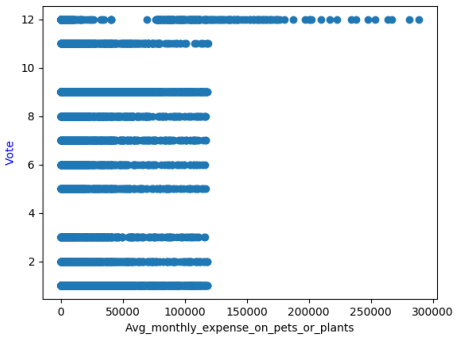


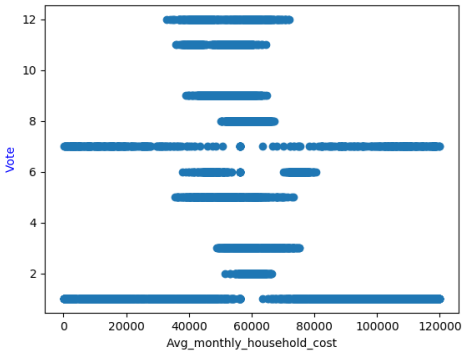


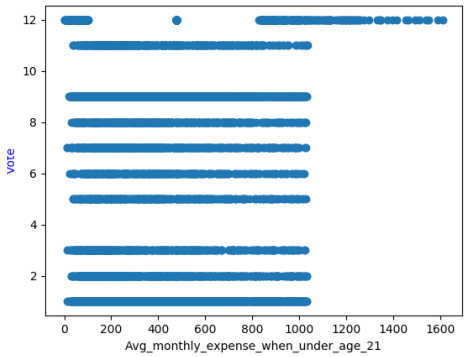


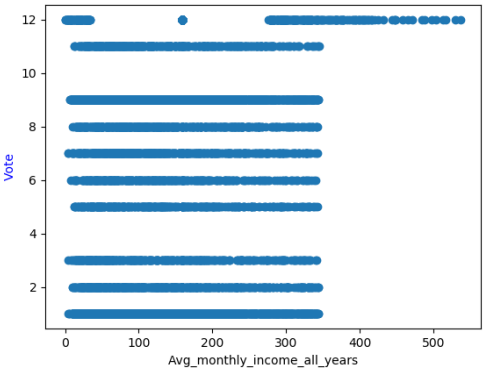


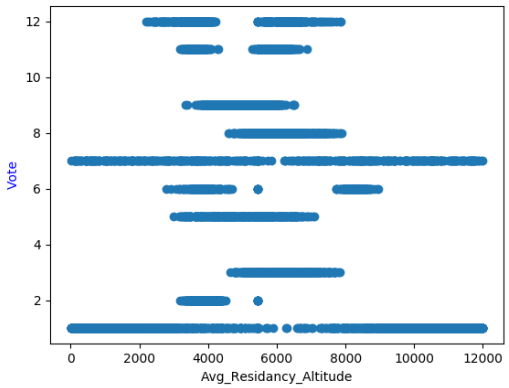


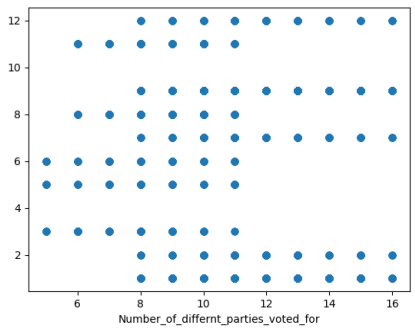


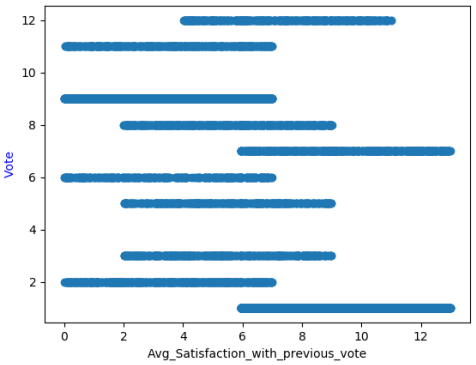




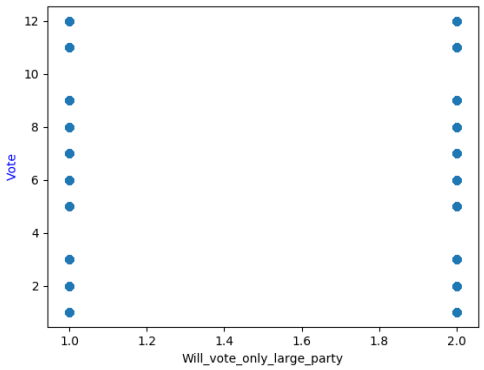


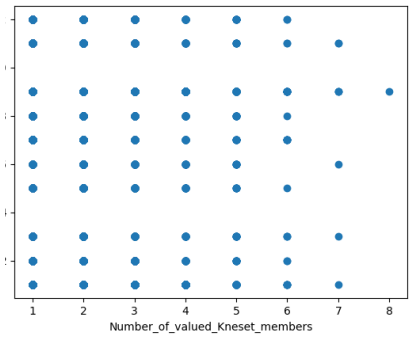


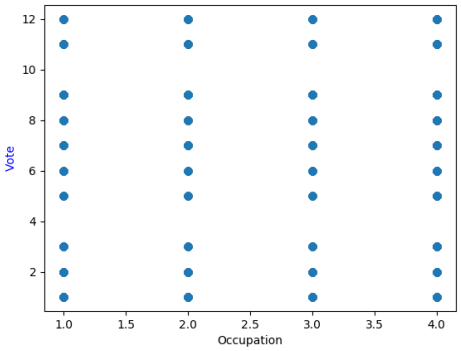


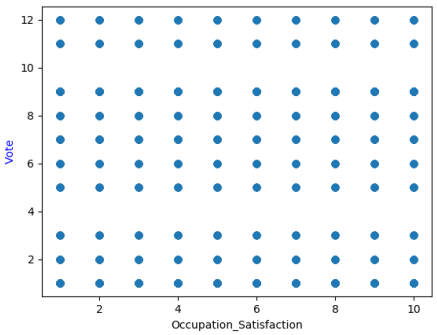


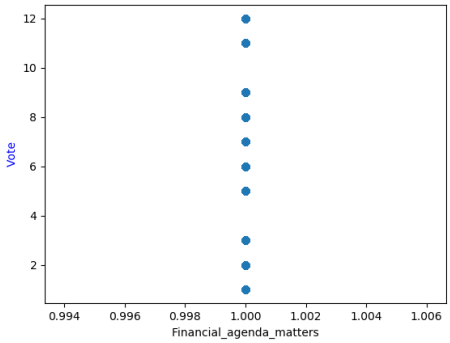
Itay Israelov
Ilan Coronel

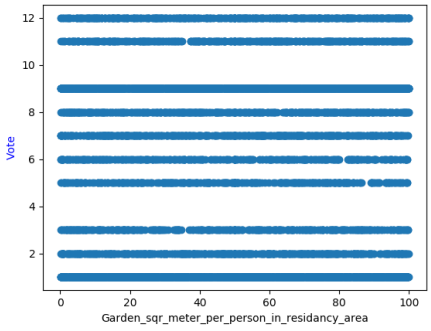


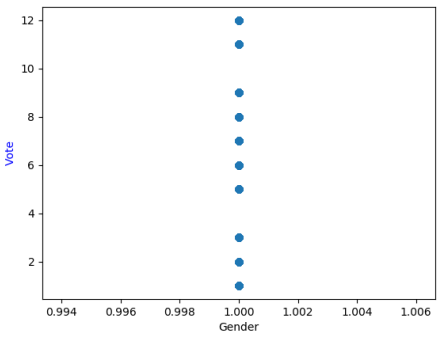


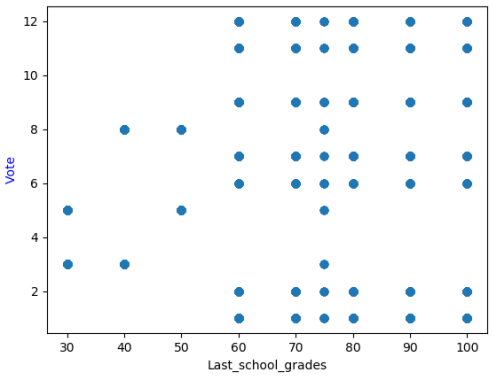


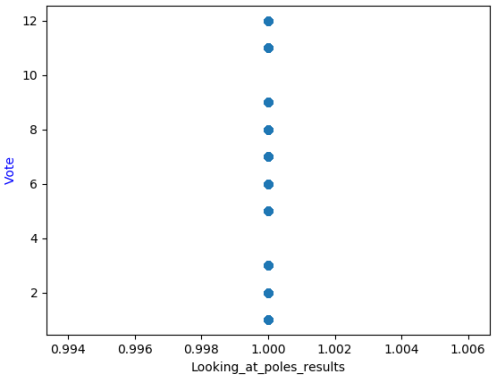


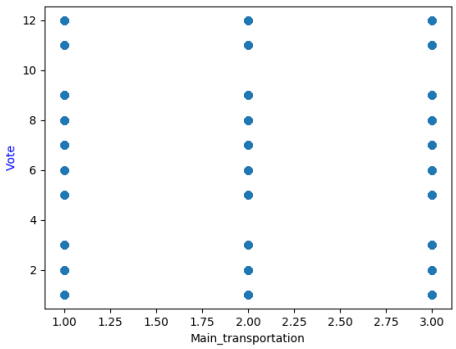


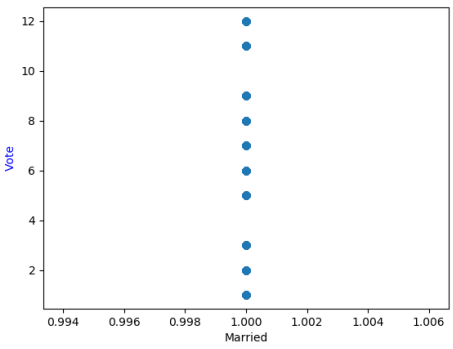


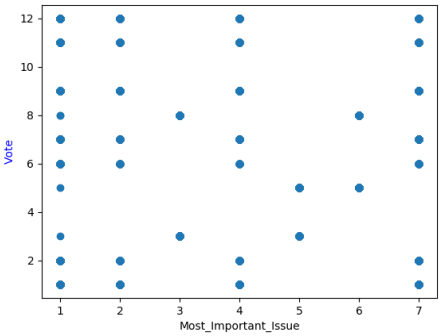




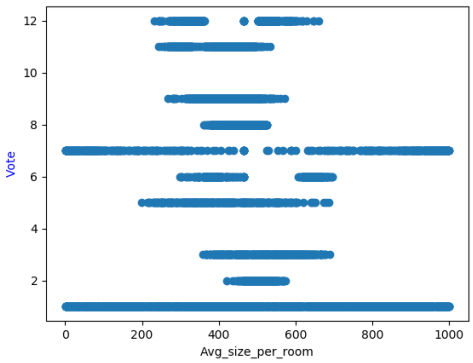


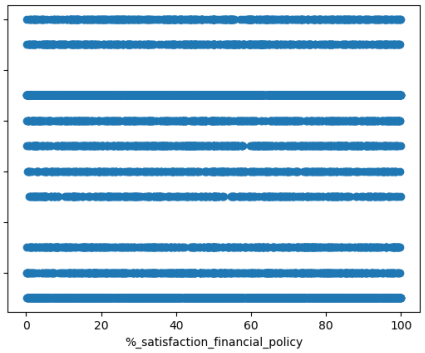


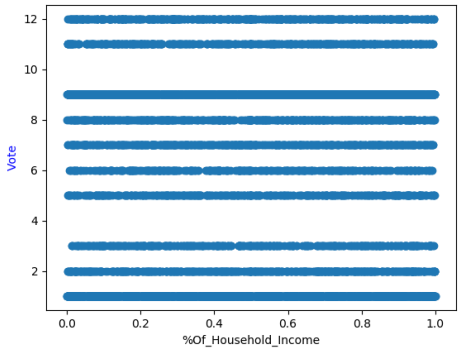


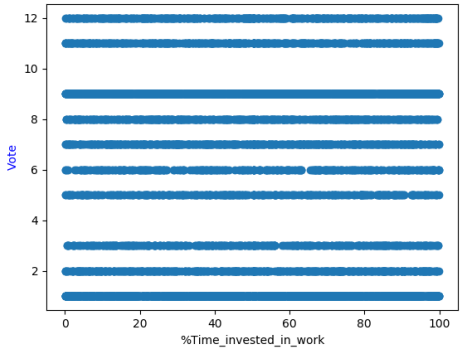


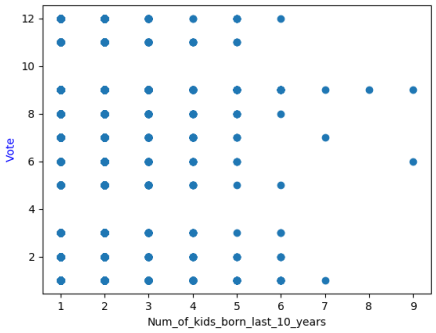
Itay Israelov
Ilan Coronel

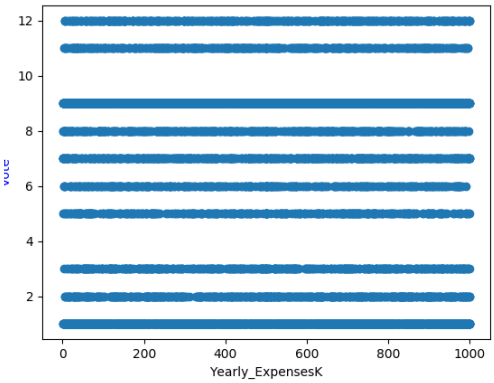


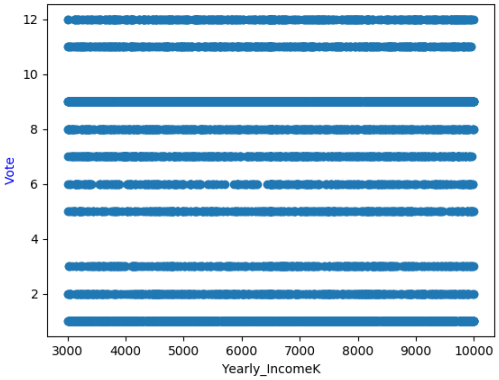


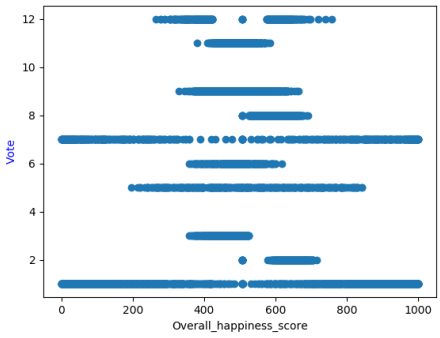


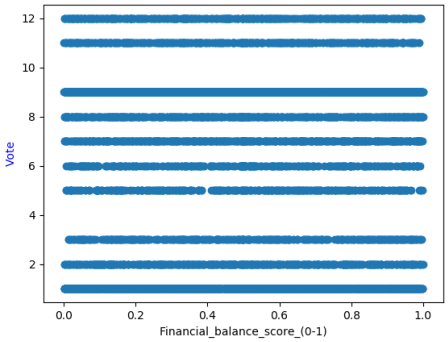


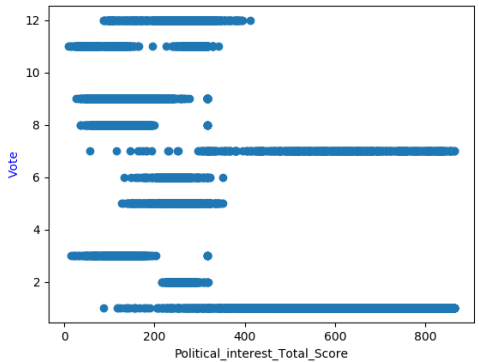


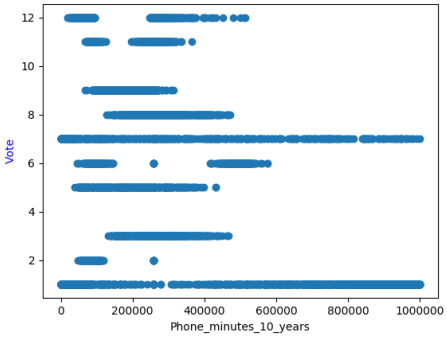


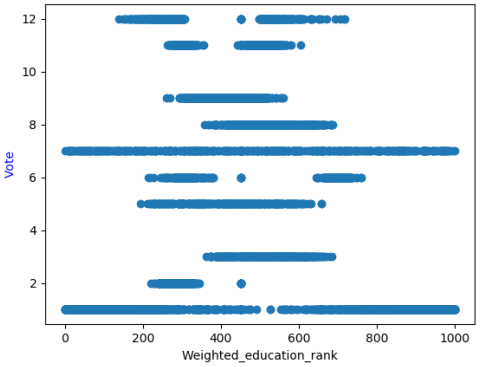




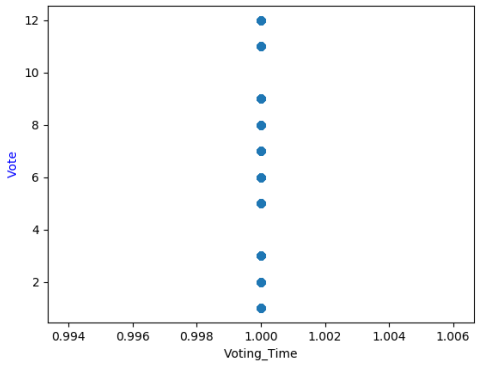








Itay Israelov
Ilan Coronel



We did not find any correlation for the following features:

'%_satisfaction_financial_policy' '%Of_Household_Income' '%Time_invested_in_work' 'Age_group' 'Avg_government_satisfaction' 'Financial_agenda_matters' 'Financial_balance_score_(0-1)' 'Garden_sqr_meter_per_person_in_residancy_area' 'Gender' 'Looking_at_poles_results' 'Main_transportation' 'Married' 'Occupation' 'Occupation_Satisfaction' 'Voting_Time' 'Will_vote_only_large_party' 'Yearly_ExpensesK' 'Yearly_IncomeK'

For the following features, we may have found some correlations:

'Avg_education_importance' - People who care about their education vote for Browns and Purples.

'Avg_environmental_importance' - Same for the environment

'AVG_lottary_expanses' – People who spend a lot of money on the lottery will vote for Browns

'Avg_monthly_expense_on_pets_or_plants' - People who spend a lot of money on pets or plants will vote for Yellows

'Avg_monthly_expense_when_under_age_21' – People who spend a lot of money when under age 21 will vote for Yellows

'Avg_monthly_household_cost' – People who care about household will vote for Browns and Purples.

'Avg_monthly_income_all_years' – If the income greater than 350 then they will vote for Yellows, otherwise there's no correlation.

'Avg_Residancy_Altitude' – If the number of residences greater than 10000 or less than 2000, then they will vote for Browns or Purples.

'Avg_Satisfaction_with_previous_vote' – If the satisfaction is greater than 8, then they will vote for Browns, Purples, or Yellows.

'Avg_size_per_room' - If the size of the room greater than 700 or less than 200, then they will vote for Browns or Purples.

'Last_school_grades' – people who get grades less than 55 will vote for Greys, Oranges or Reds.

Itay Israelov
Ilan Coronel

'Most_Important_Issue' – have a little correlation, if the important is number 3, 5 or 6 they vote for Greys, Oranges or, Reds.

'Num_of_kids_born_last_10_years' - If people with more than 7 kids will vote for Turquoises or Pinks.

'Number_of_differnt_parties_voted_for' – If the number of different parties voted is less than 6, then they will vote for Greys, Oranges, or Pinks.

'Number_of_valued_Kneset_members' – If greater than 7, then vote for Turquoises.

'Overall_happiness_score' - If the score is greater than 800 or less than 200, vote for Browns or Purples.

'Phone_minutes_10_years' - if the score is greater than 600000, then vote for Browns or Purples.

'Political_interest_Total_Score' - if the score is greater than 420, then vote for Browns or Purples.

'Weighted_education_rank' - if the rank is greater than 800 or less than 180, vote for Browns or Purples.

B.

The algorithm has been implemented in the file *featureSelection.py*

Relief is clearly the fastest algorithm among the others. However, it may not give the best results in terms of accuracy.

Itay Israelov
Ilan Coronel

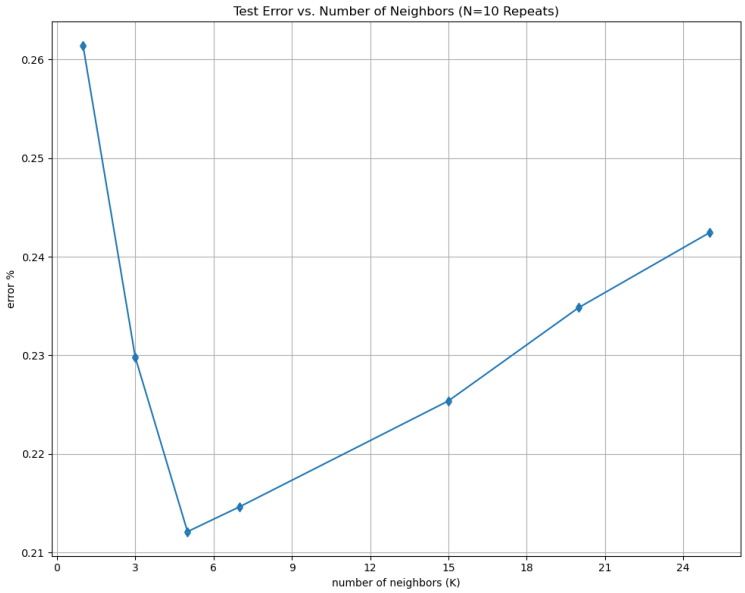
Triplets Mandatory Assignments

A.

The algorithm has been implemented in the file *featureSelection.py*

SFS is noticeably slower than Relief, especially with a big number of features. However, it seems that we get better results with him, compared to the others.

Itay Israelov
Ilan Coronel



'Avg_lottary_expanses', 'Financial_balance_score_(0-1)', 'Yearly_ExpensesK', '%Time_invested_in_work', 'Avg_government_satisfaction': appers in Relief but not apper in SBS.

- ***<u>SBS algorithm:</u>***

<u>Pros:</u>

Conservative, can choose how much features u want

<u>Cons:</u>

May take a lot of time if have a lot of features or the classifier is heavy

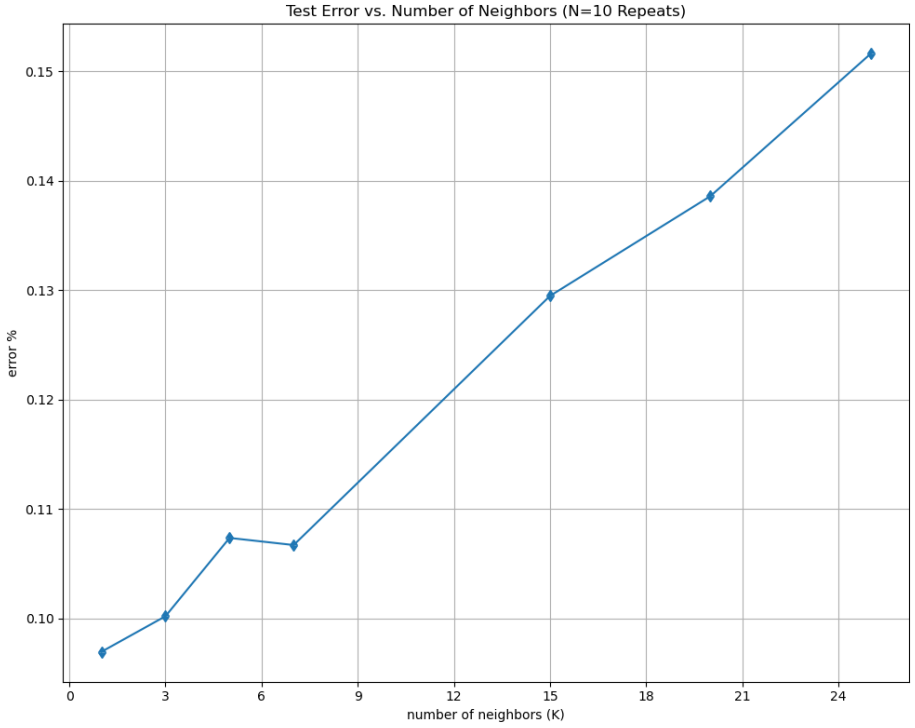
<u>KNN:</u>

Itay Israelov
Ilan Coronel

```
knn = KNeighborsClassifier(n_neighbors=3)
sbs = SFS(knn,
          k_features=8,
          forward=False,  # if forward = True then SFS otherwise SBS
          floating=False,
          verbose=2,
          n_jobs=-1,
          scoring='accuracy')
```

features:

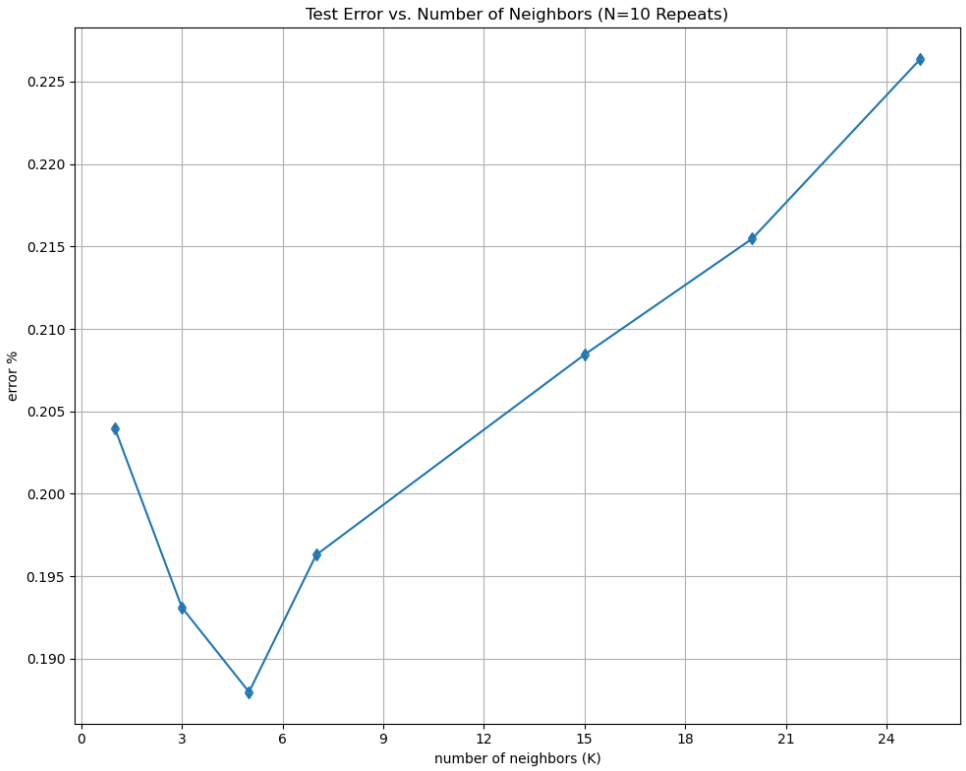
['Vote' 'Avg_environmental_importance' 'Yearly_IncomeK'
'Avg_Residancy_Altitude' 'Avg_Satisfaction_with_previous_vote'
'Last_school_grades' 'Number_of_differnt_parties_voted_for'
'Political_interest_Total_Score' 'Overall_happiness_score']
give us 90% accuracy.



GradientBoost

```
clf_gradient = GradientBoostingClassifier(n_estimators=100, random_state=0)
sbs = SFS(clf_gradient,
          k_features=8,
          forward=False,  # if forward = True then SFS otherwise SBS
          floating=False,
          verbose=2,
          n_jobs=-1,
          scoring='accuracy')
```

Itay Israelov
Ilan Coronel




```
['Vote' 'Avg_monthly_expense_when_under_age_21'
 'Avg_environmental_importance' 'Avg_Residancy_Altitude'
 'Avg_Satisfaction_with_previous_vote' 'Most_Important_Issue'
 'Number_of_differnt_parties_voted_for' 'Political_interest_Total_Score'
 'Overall_happiness_score']
```

'Avg_monthly_expense_when_under_age_21', 'Most_Important_Issue': appers in GradientBoost but not appers in KNN.

'Yearly_IncomeK', 'Last_school_grades': appers in KNN
but not appers in GradientBoost.