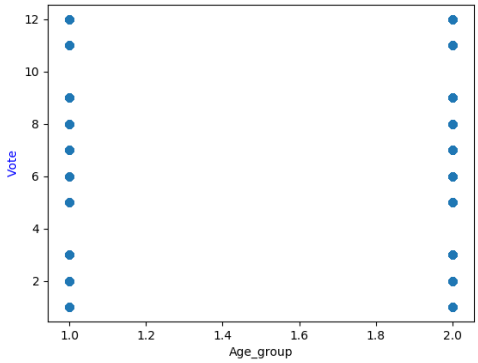
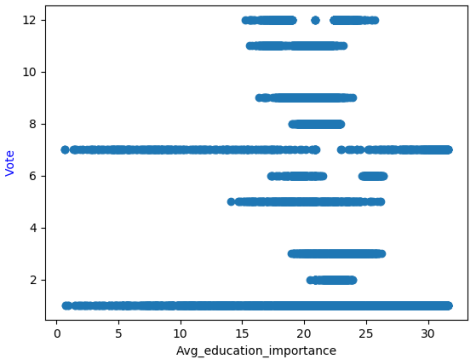
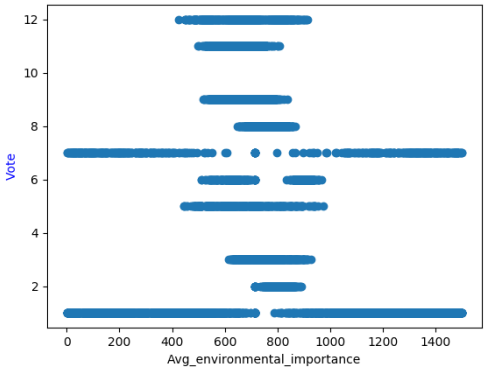
Itay Israelov

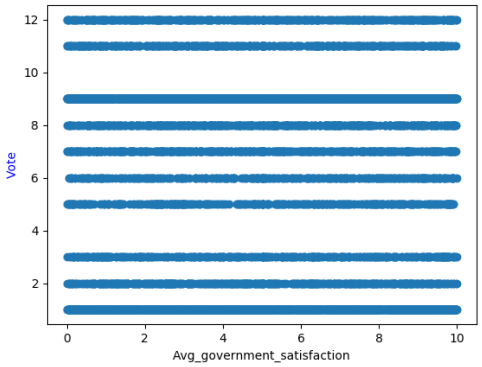
Report hw2

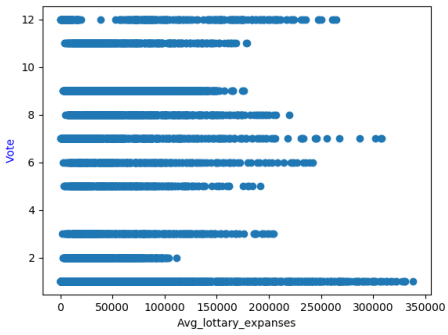
- ***Relation or lack of relation between the features and the labels features:***

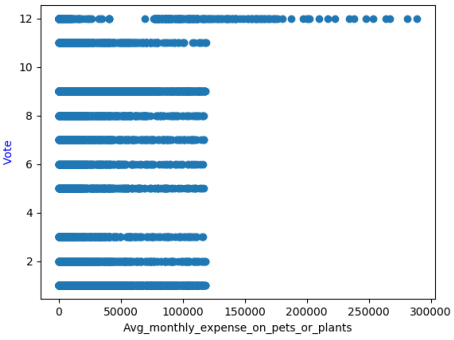


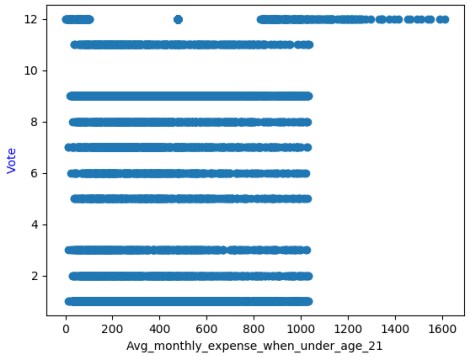


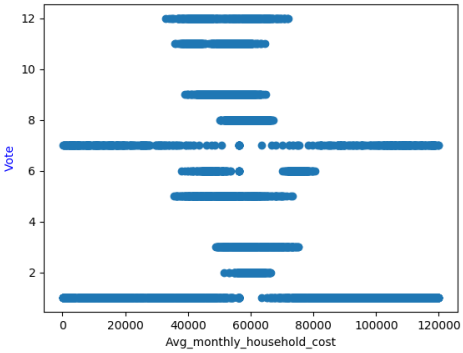




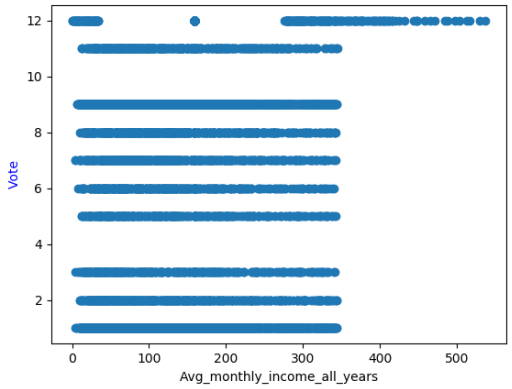


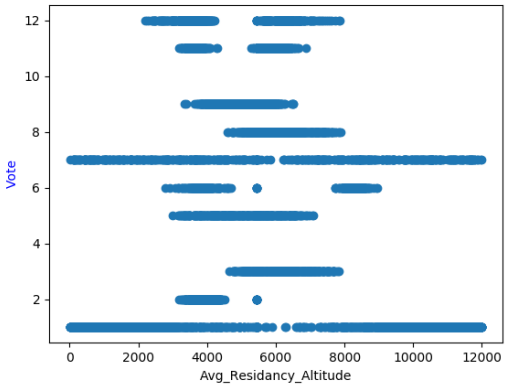


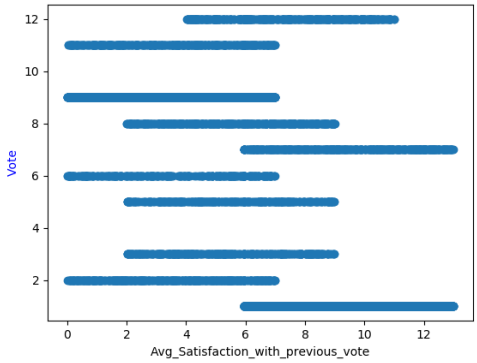


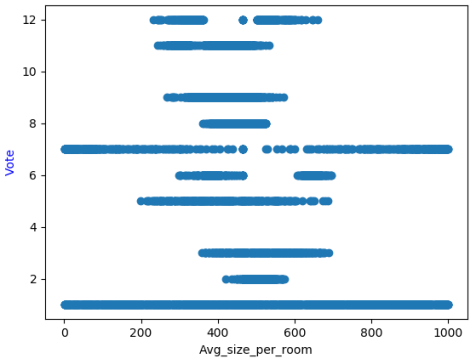


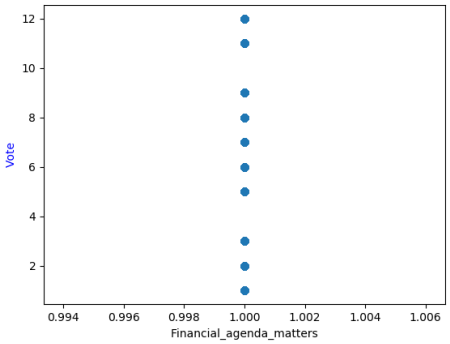
Itay Israelov

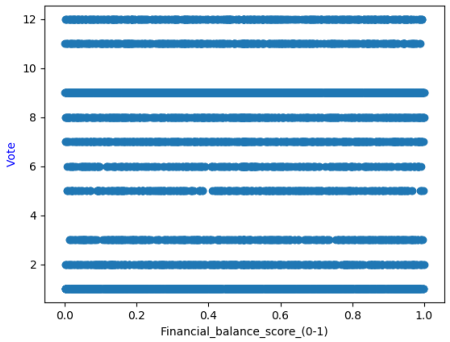


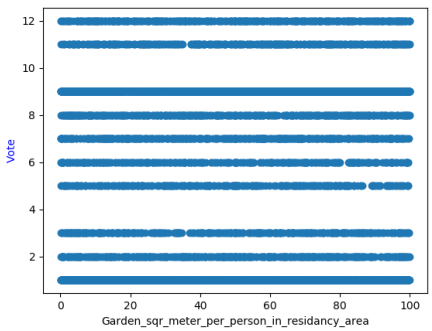


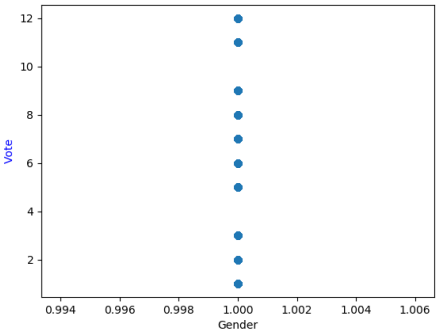




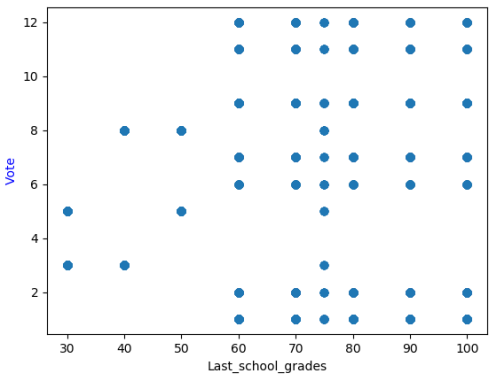


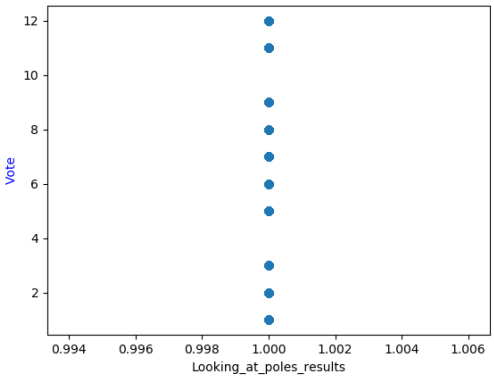


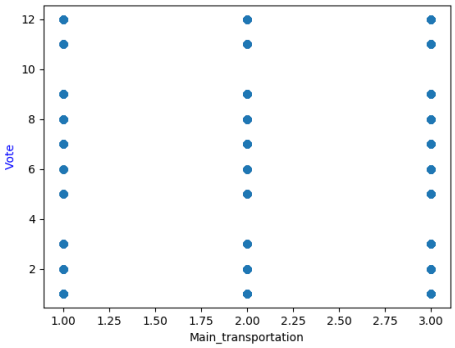


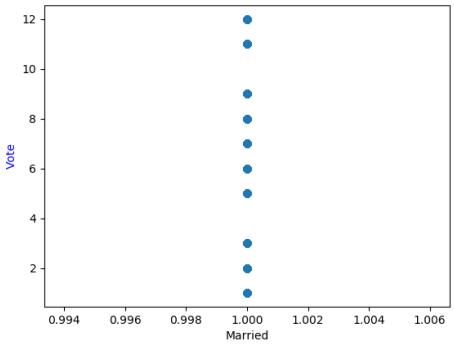


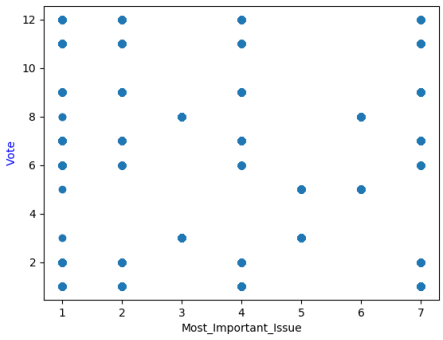
Itay Israelov

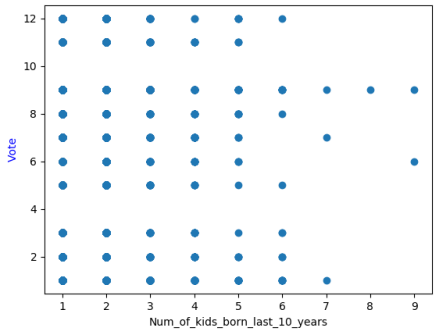


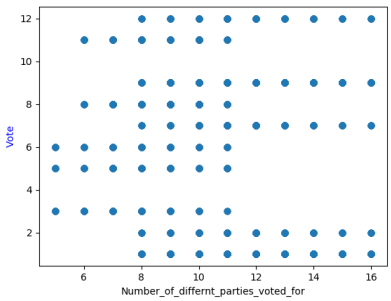


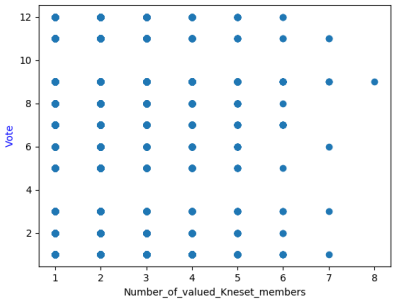




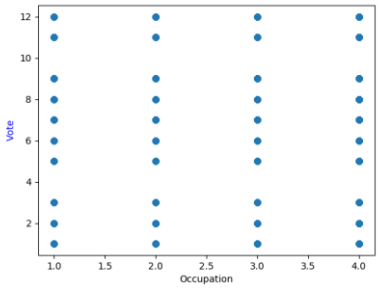


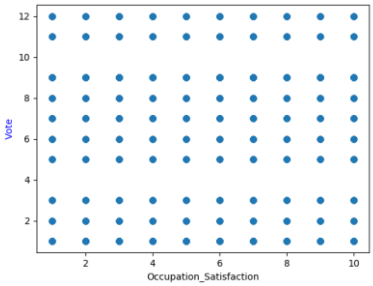


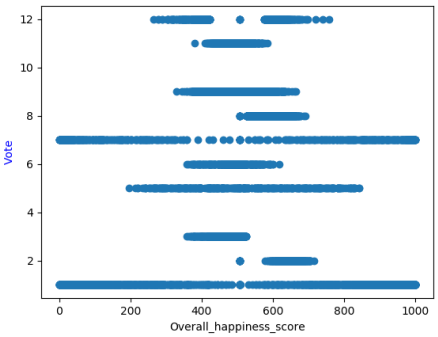


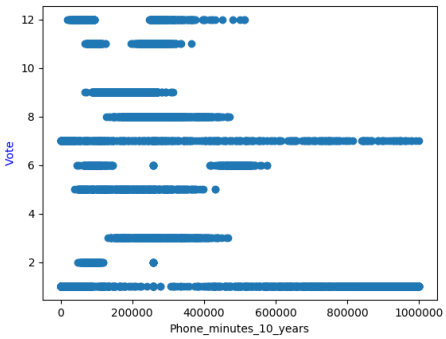


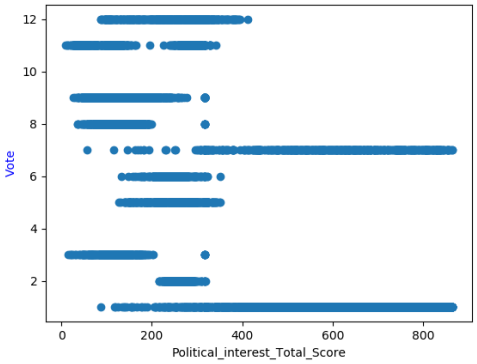
Itay Israelov

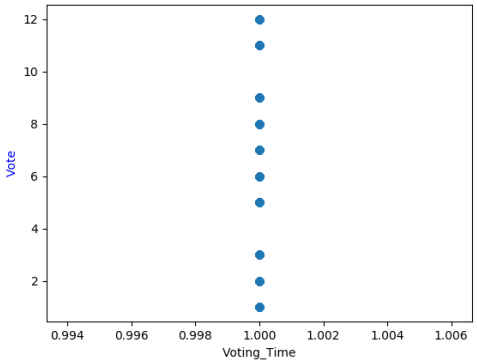


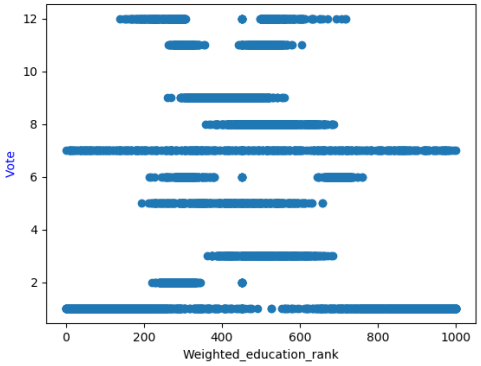


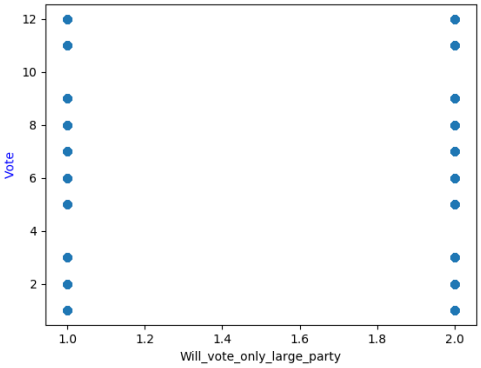




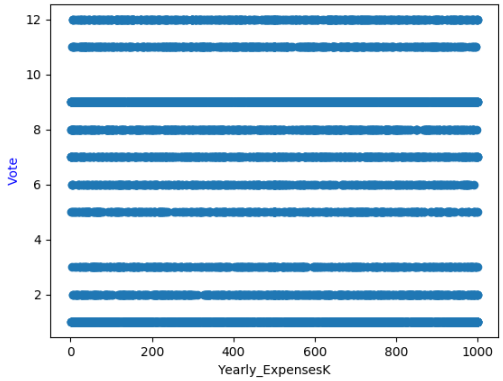


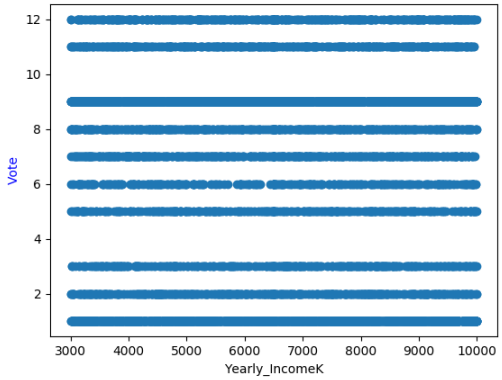


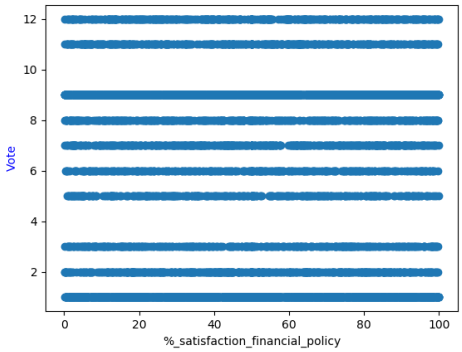


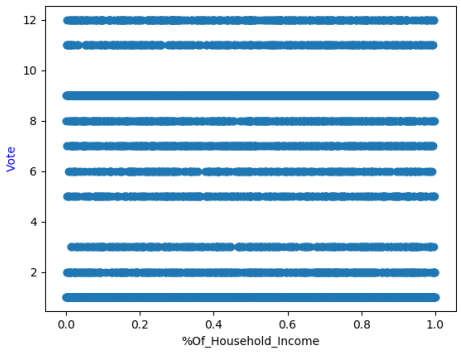


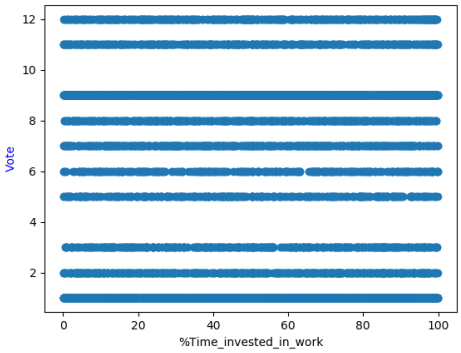
Itay Israelov











'%_satisfaction_financial_policy' – no correlation

'%Of_Household_Income' - no correlation

'%Time_invested_in_work' - no correlation

'Age_group' - no correlation

'Avg_education_importance' - people who care about their education vote to Browns and Purples.

'Avg_environmental_importance' - same for the environment

'Avg_government_satisfaction' - no correlation

'AVG_lottary_expanses' – people who spent a lot of money on the lottery is a vote for Browns

Itay Israelov

'Avg_monthly_expense_on_pets_or_plants' - people who spent a lot of money on pets or plants is a vote for Yellows

'Avg_monthly_expense_when_under_age_21' – some is before

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

'Avg_monthly_income_all_years' – if the income greater from 350 then vote for Yellows, otherwise no correlate.

'Avg_Residancy_Altitude' – if the number of residences greater from 10000 or less then 2000, then vote for Browns or Purples.

'Avg_Satisfaction_with_previous_vote' – if satisfaction greater then 8 vote for Browns, Purples, or Yellows.

'Avg_size_per_room' - if the size of the room greater from 700 or less then 200, then vote for Browns or Purples.

'Financial_agenda_matters' - no correlation

'Financial_balance_score_(0-1)' - no correlation

'Garden_sqr_meter_per_person_in_residancy_area' - no correlation

'Gender' - no correlation

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

'Looking_at_poles_results' - no correlation

'Main_transportation' - no correlation

'Married' - no correlation

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

'Num_of_kids_born_last_10_years' - if someone has greater then 7 kids is a vote for Turquoises or Pinks.

'Number_of_differnt_parties_voted_for' – if the number of different parties voted is less then 6, then is a vote for Greys, Oranges, or Pinks.

'Number_of_valued_Kneset_members' – if greater then 7, then vote for Turquoises.

'Occupation' - no correlation

'Occupation_Satisfaction' - no correlation

'Overall_happiness_score' - if the score is greater from 800 or less then 200, vote for Browns or Purples.

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

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

'Voting_Time' - no correlation

Itay Israelov

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

'Will_vote_only_large_party' - no correlation

'Yearly_ExpensesK' - no correlation

'Yearly_IncomeK' - no correlation

Summary:

18 - no correlation

19 – yes correlation

no_correlation = [

'%_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' ,

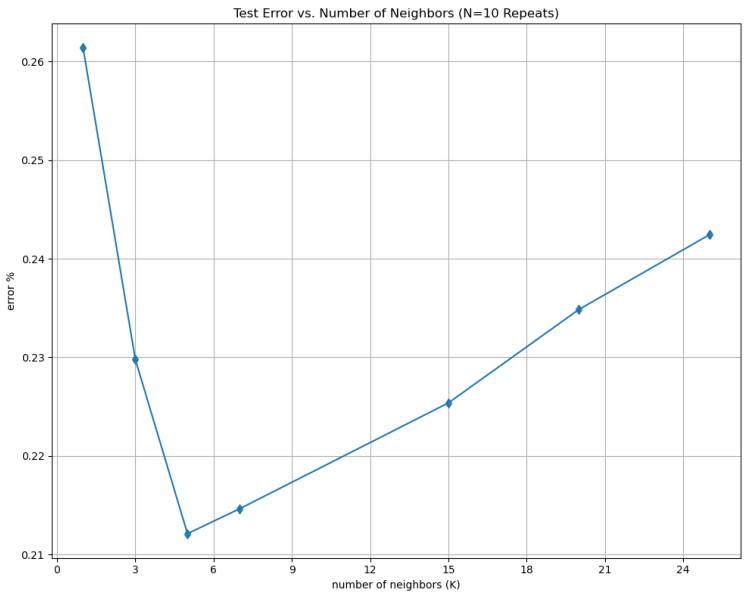
'Yearly_ExpensesK' ,

'Yearly_IncomeK']

Itay Israelov

```
yes_correlation = [
'Avg_education_importance' ,
 'Avg_environmental_importance',
 'AVG_lottary_expanses' ,
 'Avg_monthly_expense_on_pets_or_plants' ,
 'Avg_monthly_expense_when_under_age_21' ,
 'Avg_monthly_household_cost' ,
 'Avg_monthly_income_all_years' ,
 'Avg_Residancy_Altitude' ,
 'Avg_Satisfaction_with_previous_vote' ,
 'Avg_size_per_room' ,
 'Last_school_grades' ,
 'Num_of_kids_born_last_10_years' ,
 'Number_of_differnt_parties_voted_for' ,
 'Number_of_valued_Kneset_members' ,
 'Overall_happiness_score' ,
 'Phone_minutes_10_years' ,
 'Political_interest_Total_Score' ,
 'Weighted_education_rank',
'Most_Important_Issue' ]
```

- ***Relief algorithm:***
  Their strengths are that they are not dependent on heuristics, they run in low-order polynomial time, and they are noise-tolerant and robust to feature interactions, as well as being applicable for binary or continuous data; however, it does not discriminate between redundant features, and low numbers of training instances fool the algorithm.



['Vote' 'Avg_monthly_expense_when_under_age_21' 'Avg_lottary_expanses'
'Avg_environmental_importance' 'Financial_balance_score_(0-1)'
'Avg_Residancy_Altitude' 'Yearly_ExpensesK' '%Time_invested_in_work'
'Avg_Satisfaction_with_previous_vote' 'Avg_government_satisfaction'
'Last_school_grades' 'Number_of_differnt_parties_voted_for'
'Political_interest_Total_Score' 'Overall_happiness_score']

'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.

- ***SBS algorithm:***

Pros:

Conservative, can choose how much features u want

Cons:

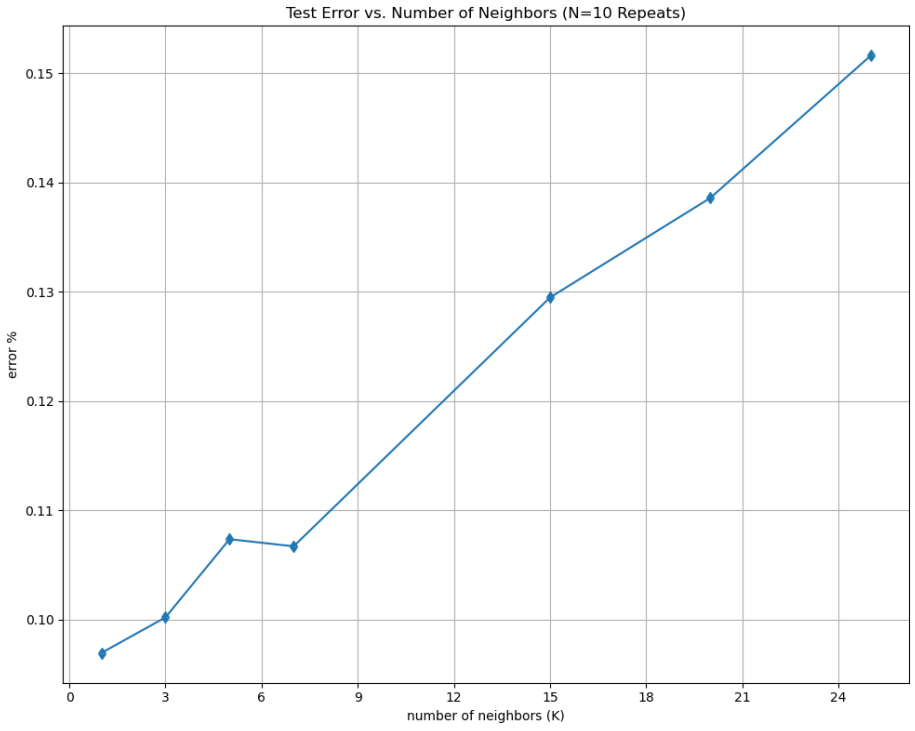
May take a lot of time if have a lot of features or the classifier is heavy

KNN:

```
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:

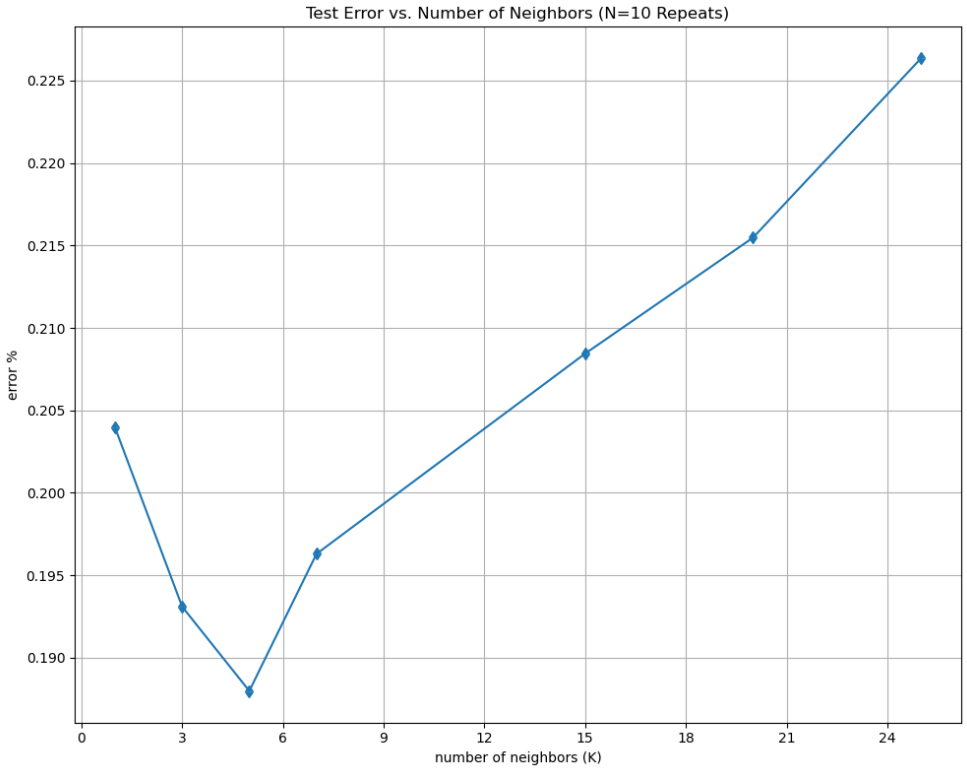
['Vote' 'Avg_environmental_importance' 'Yearly_IncomeK'
 'Avg_Residancy_Altitude' 'Avg_Satisfaction_with_previous_vote'
 'Last_school_grades' 'Number_of_differnt_parties_voted_for'
 'Political_interest_Total_Score' 'Overall_happiness_score']
give us 90% accuracy.



Itay Israelov

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')
```




```
['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.

Itay Israelov

Summey for features selection:

(SBS from KNN) + (SBS from GradientBoost) + (Relief) – (no_correlation) =

final =
[
'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'
'Last_school_grades'
'Avg_lottary_expanses'
]