***236756 - Introduction to Machine Learning – HW2***

***Report***

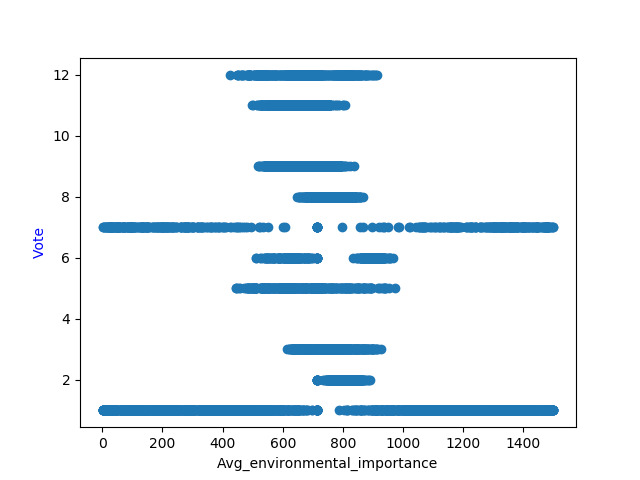
***Mandatory Assignment Part***

Here we will present our process.

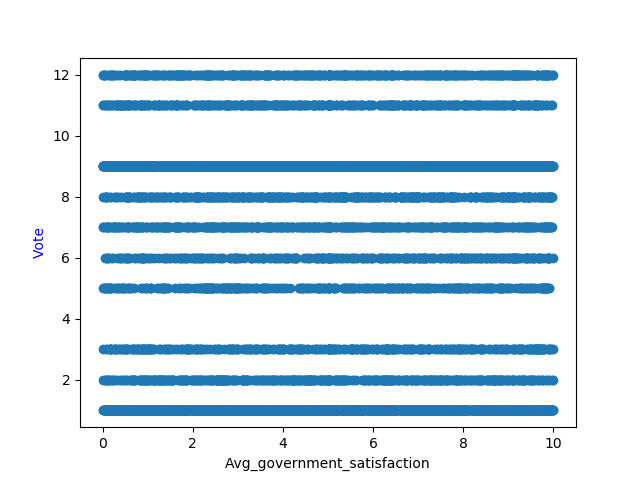
First, we converted every nominal feature to numbers, so we can work with numerical data.  
We found out there are 3 types of features:  
The nominals, which are strings. The integer features, which contain integers. And the float features, which contain values with decimals.

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:  


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.

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 89.36531%

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

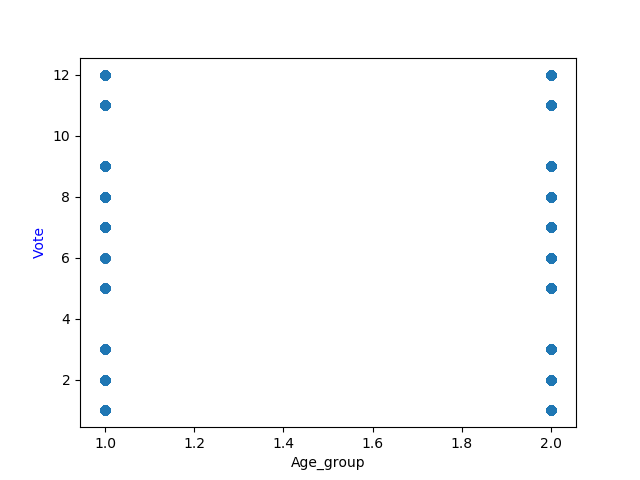
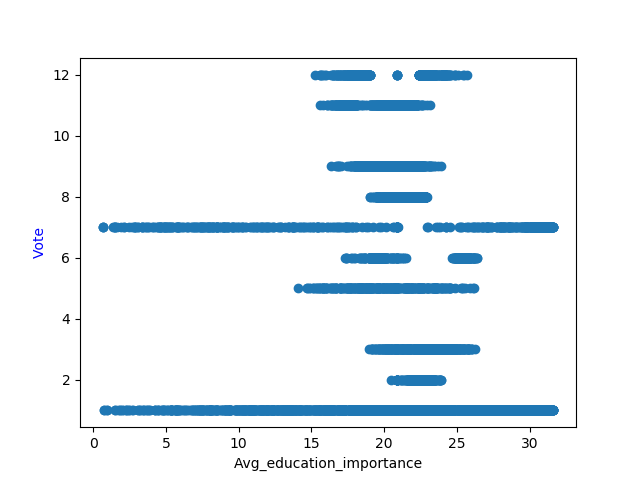
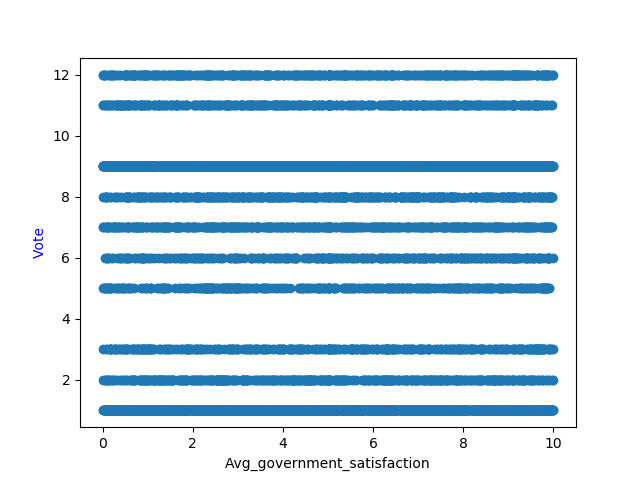
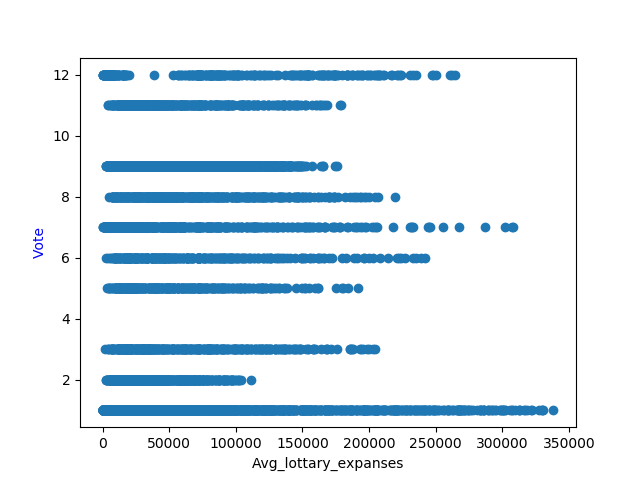
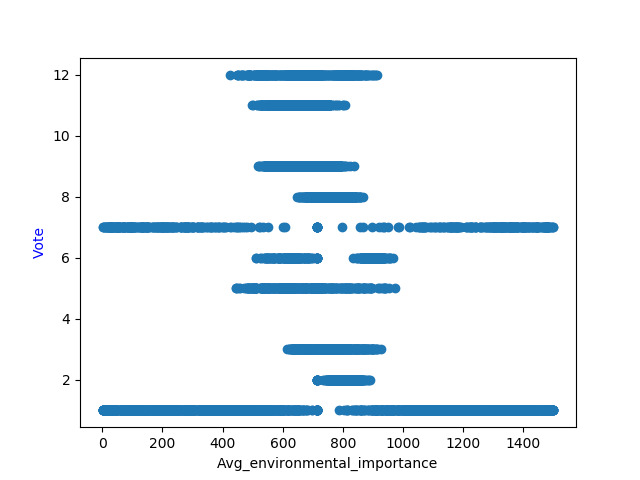
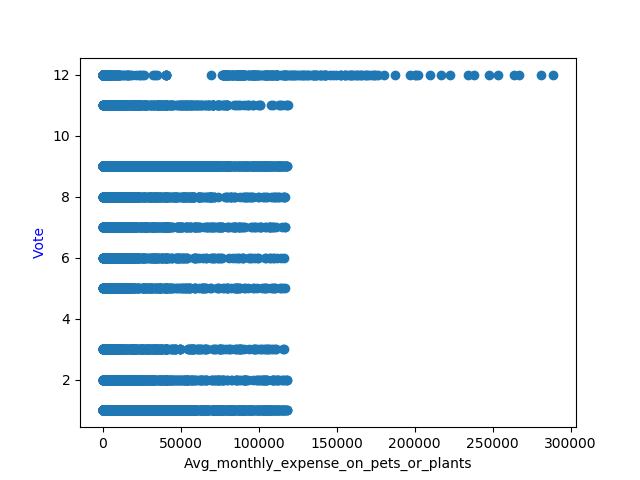
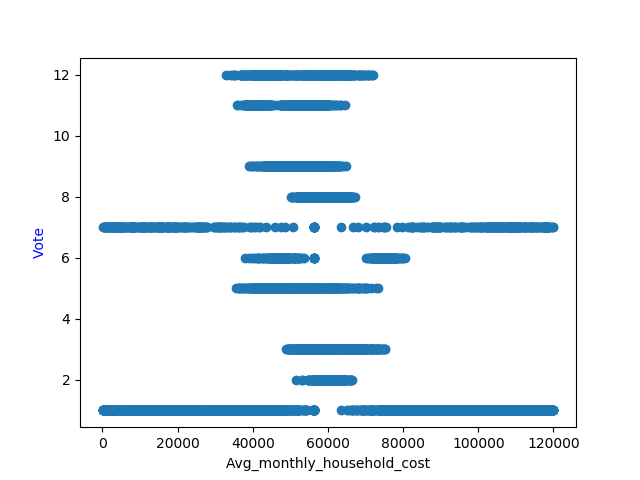
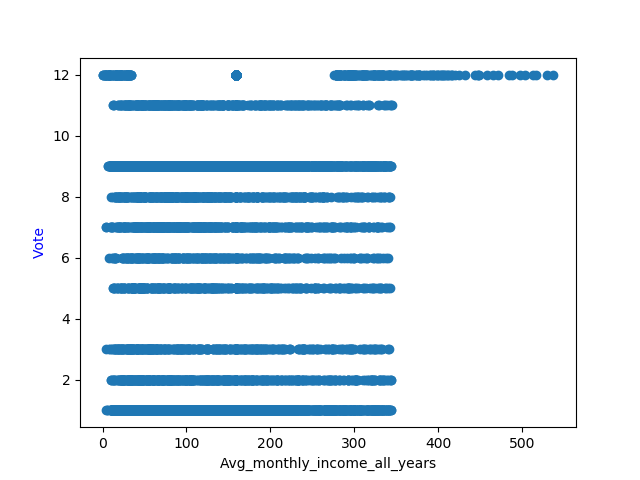
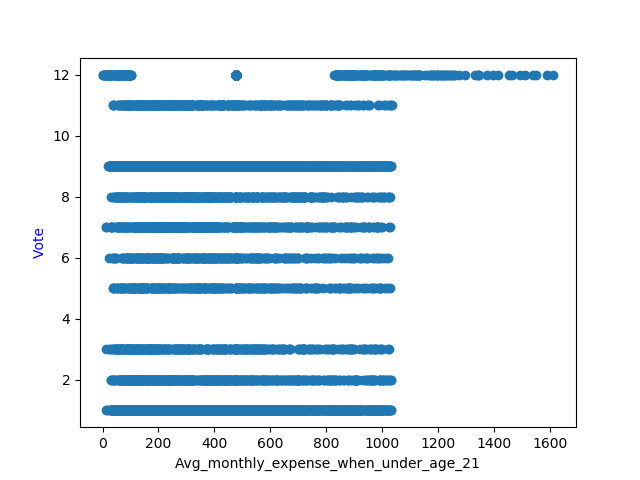
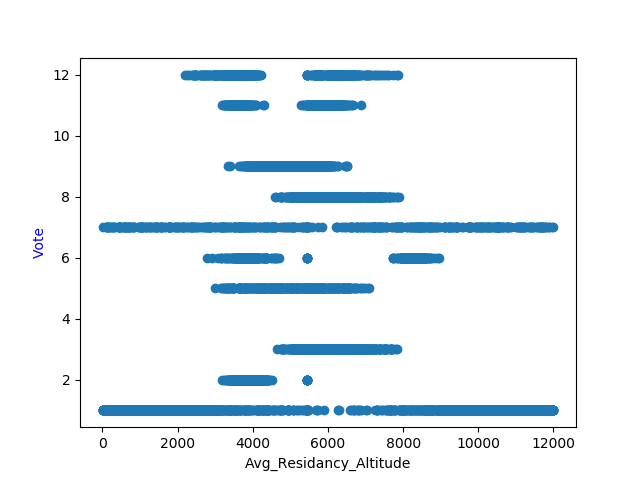
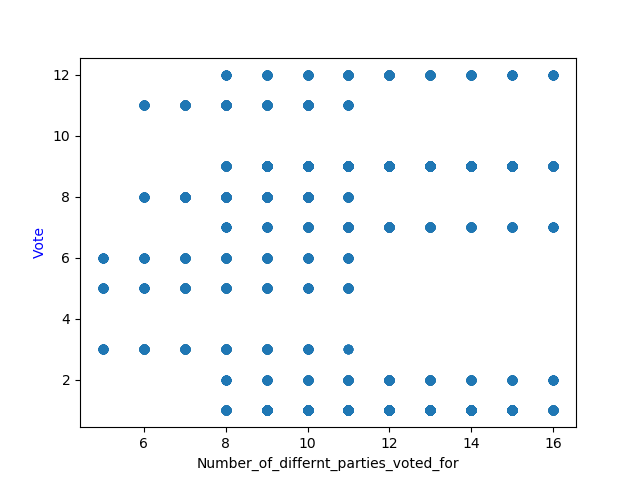
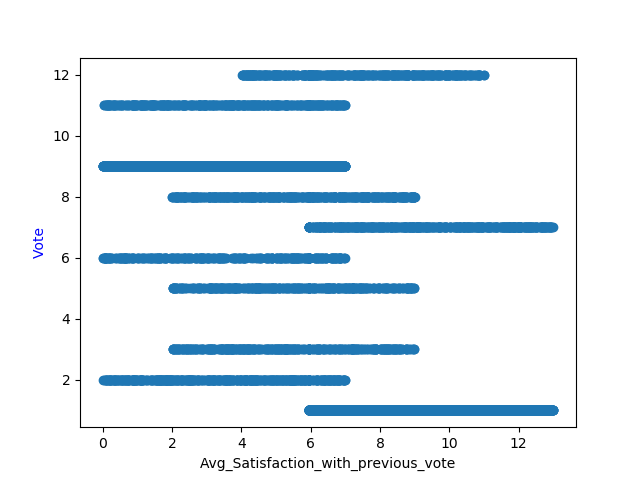
Political\_interest\_Total\_Score

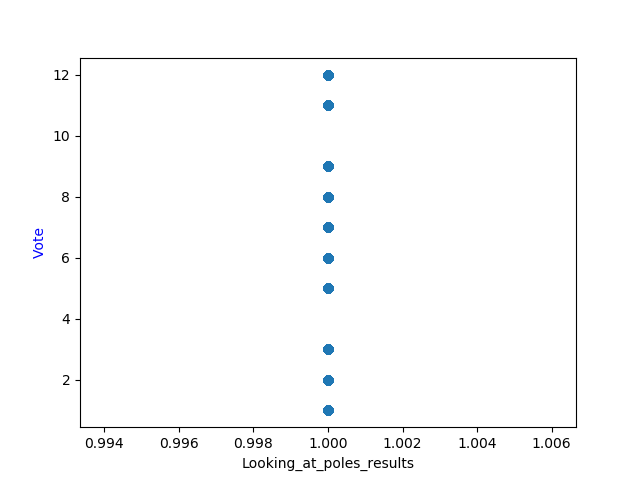
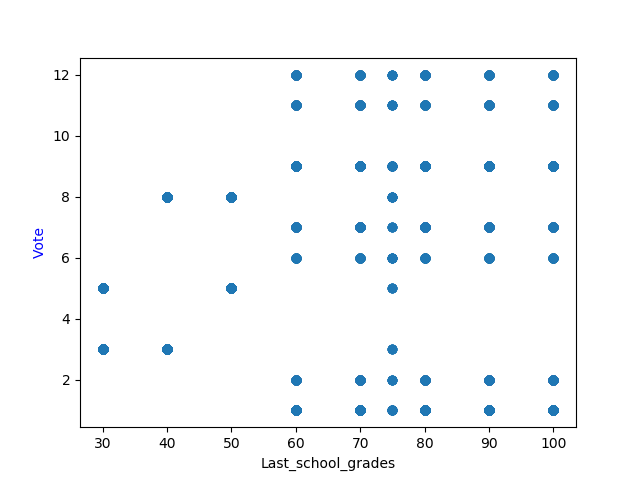
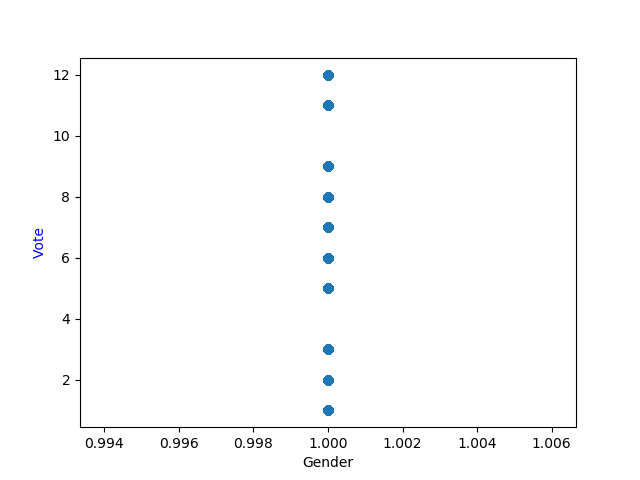
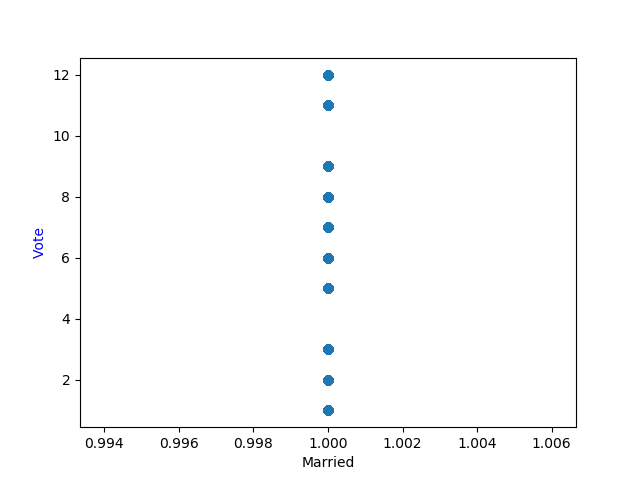
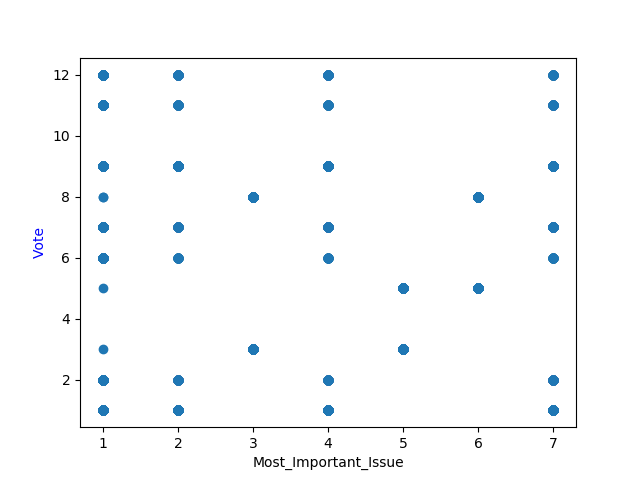
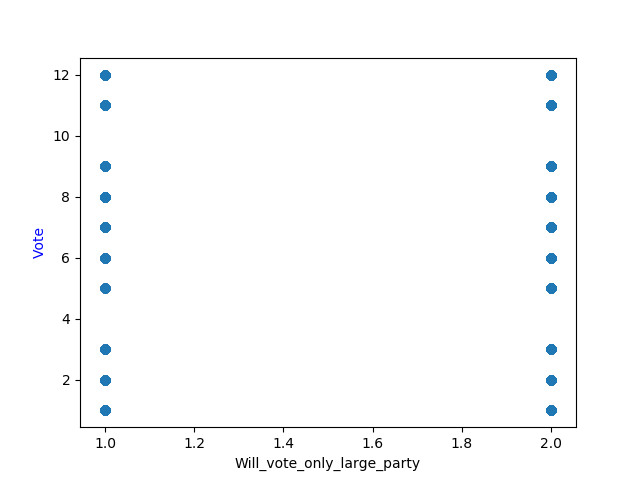
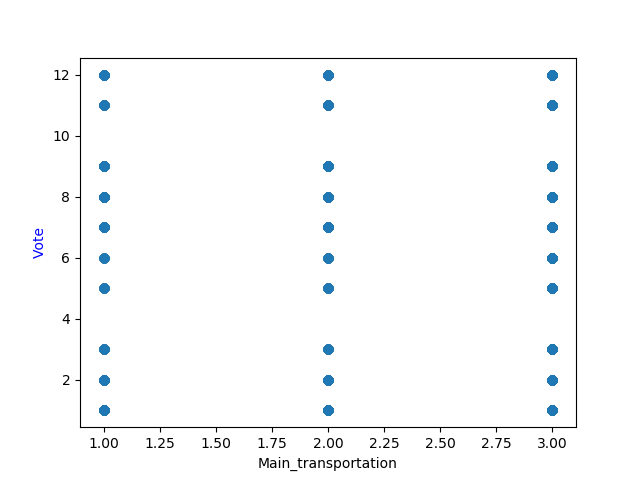
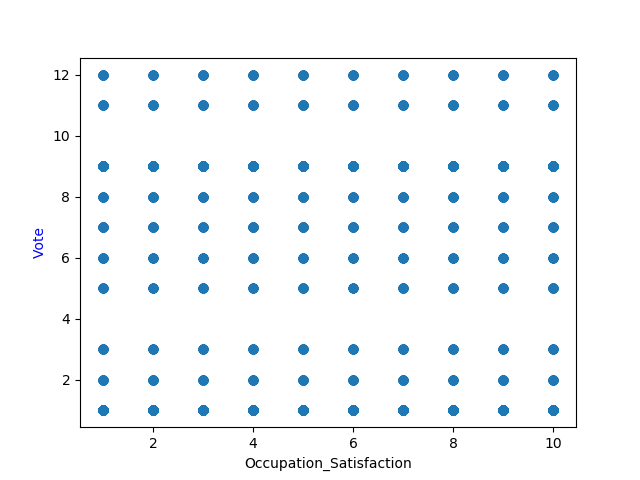
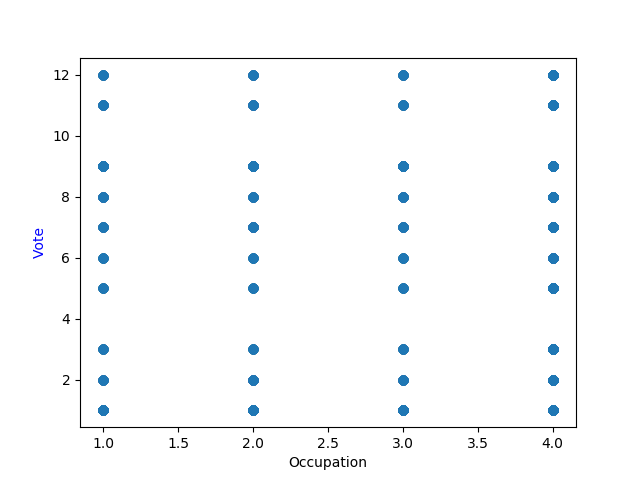
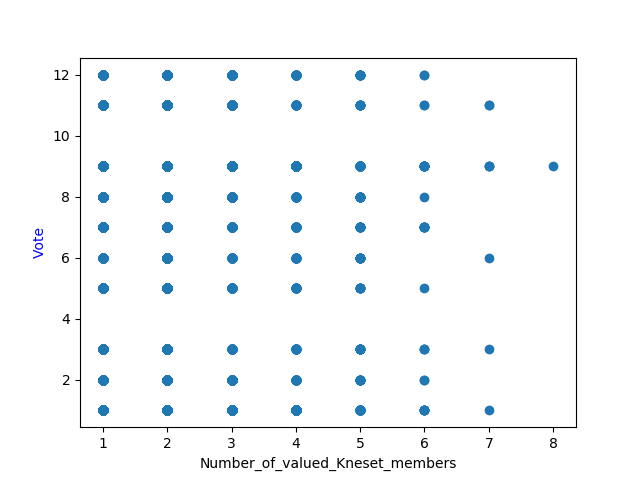
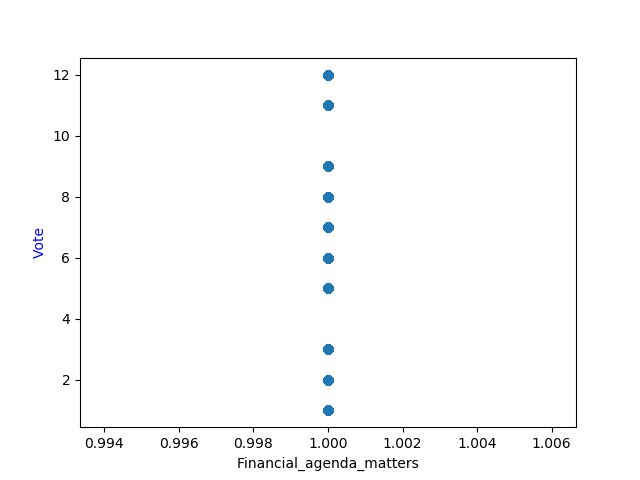
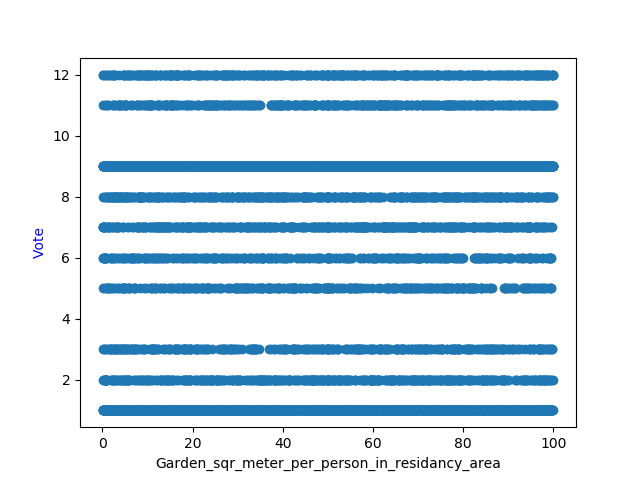
Overall\_happiness\_score

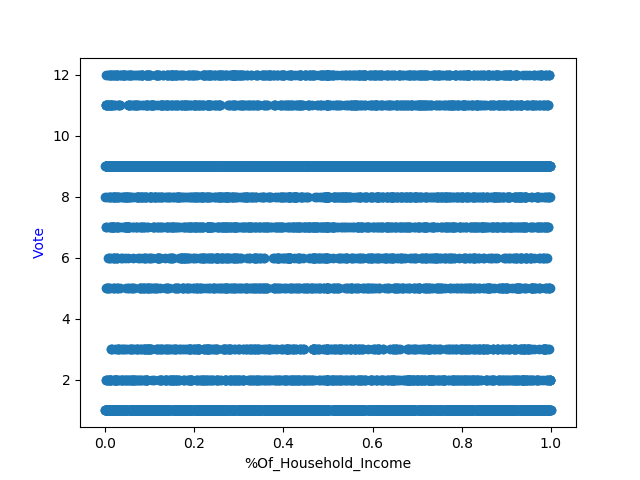
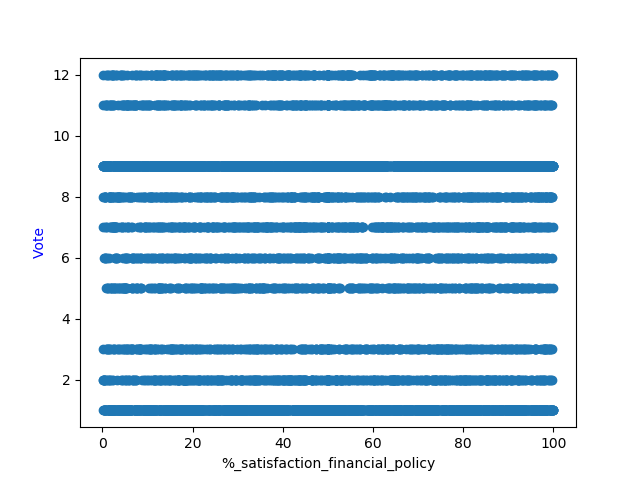
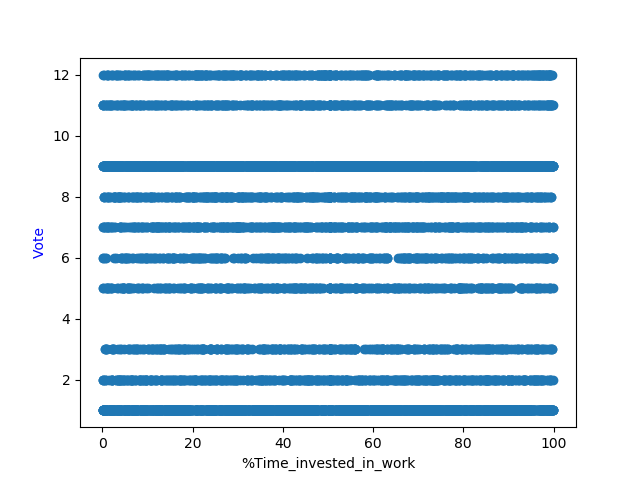
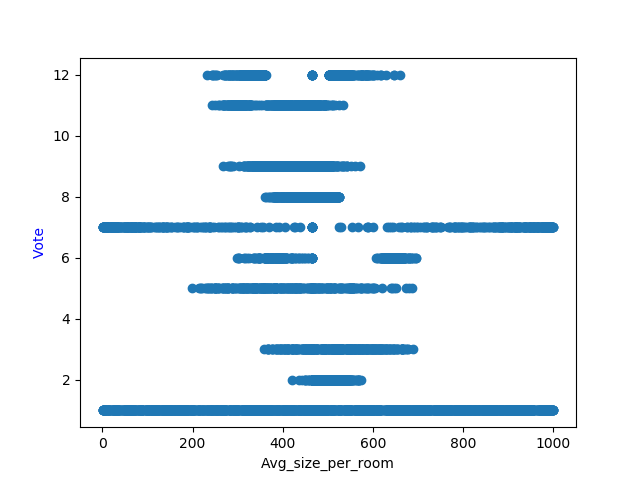
For a final accuracy of 89.36531% using the KNN classifier (with a parameter of 5 neighbors).

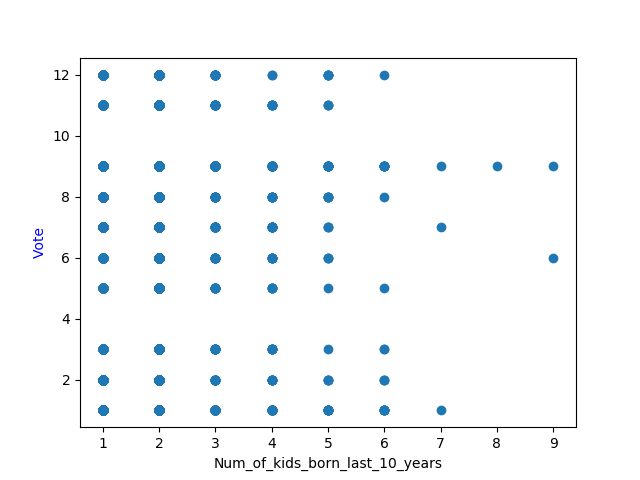
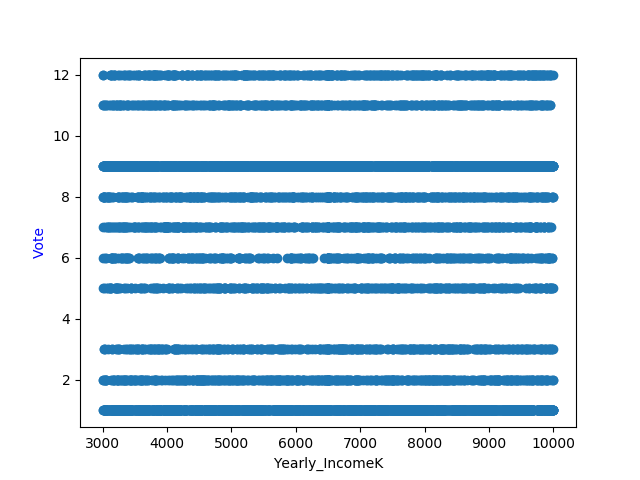
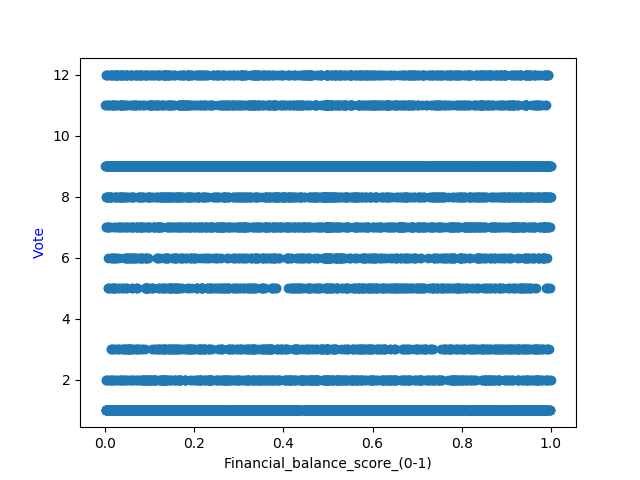
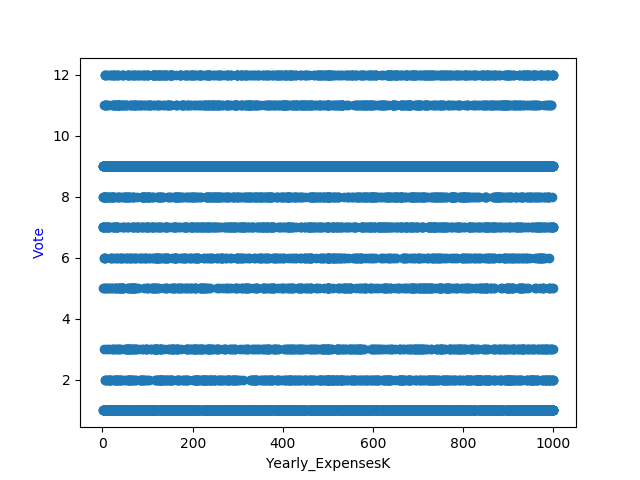
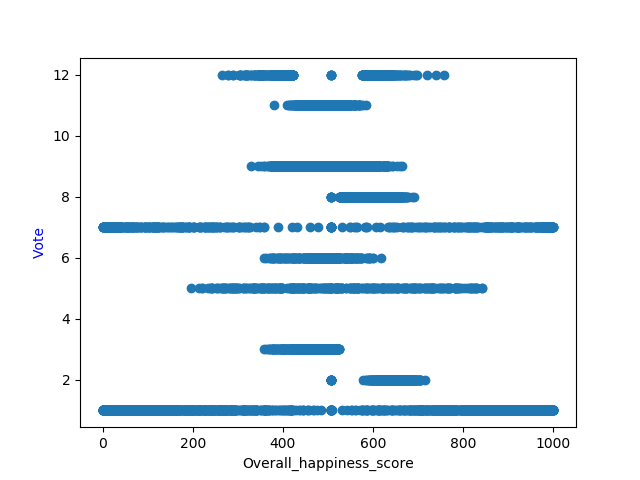
***Non-Mandatory (Bonus) Assignments***

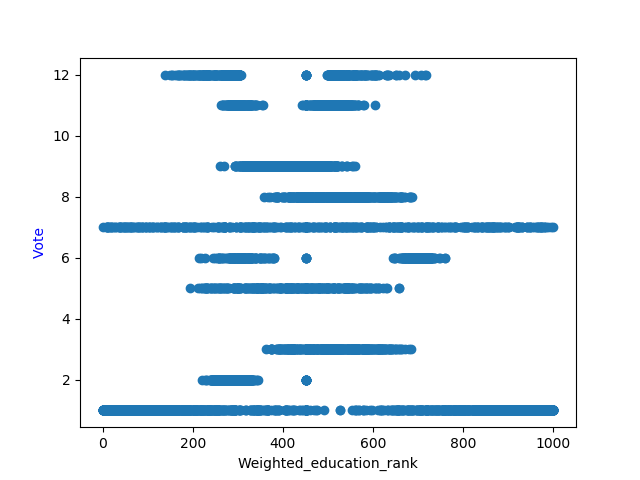
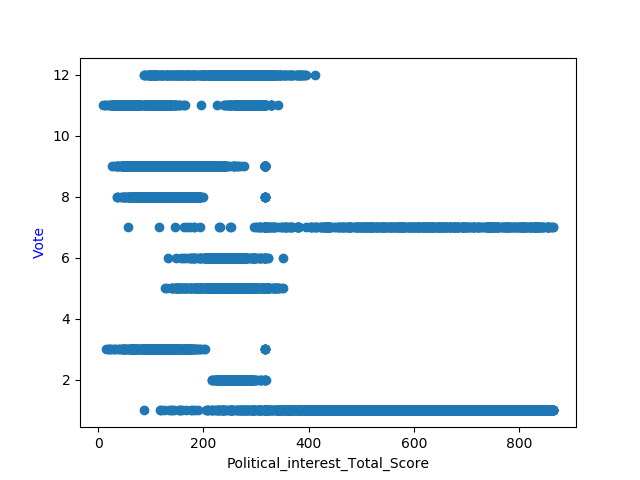
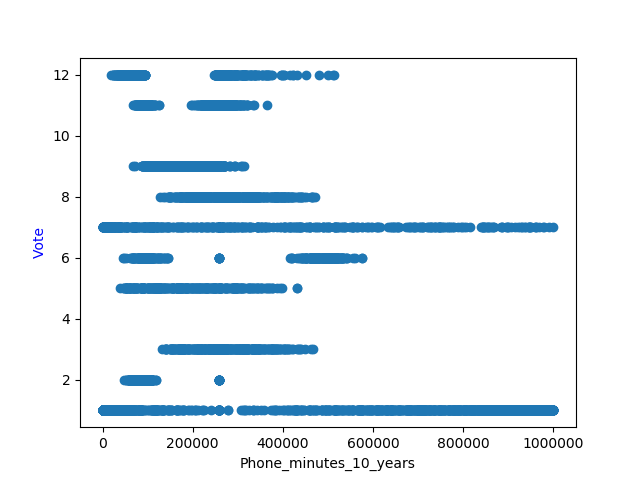
***A.***

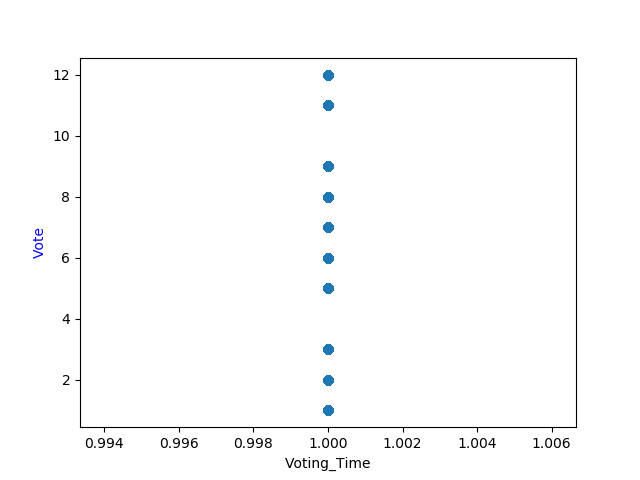
* ***Relation or lack of relation between the features and the labels features:***











'%\_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

'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

'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

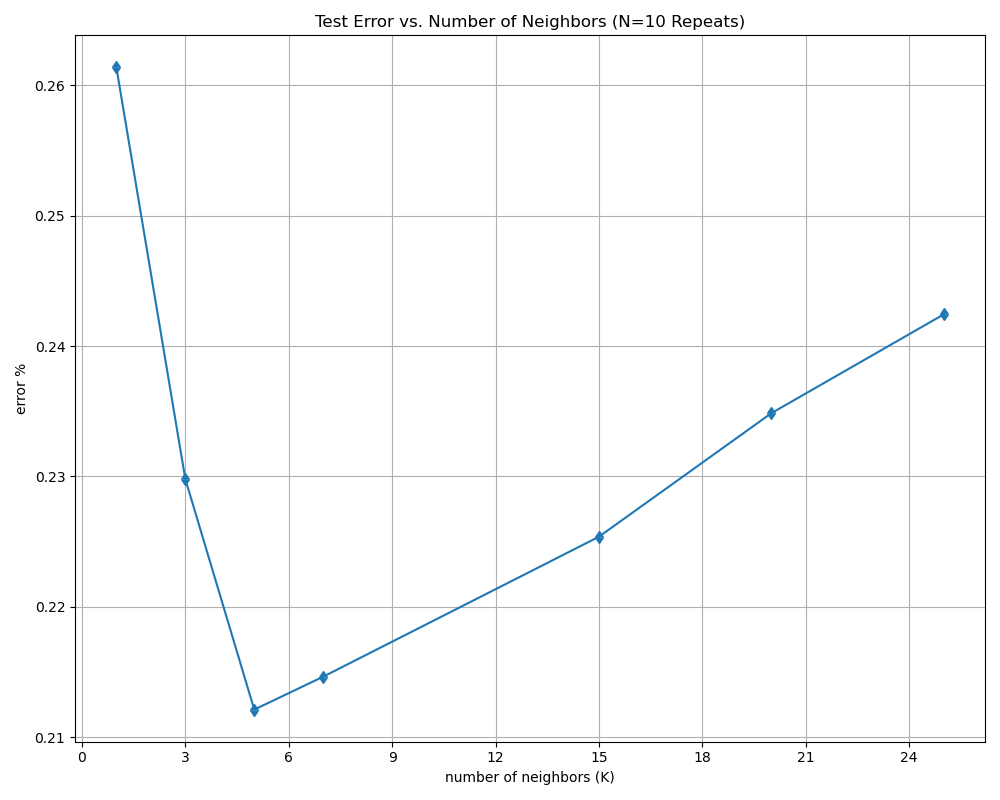
***B.***

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

***Triplets Mandatory Assignment (Bonus for Pairs)***

***A.***

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



['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:**

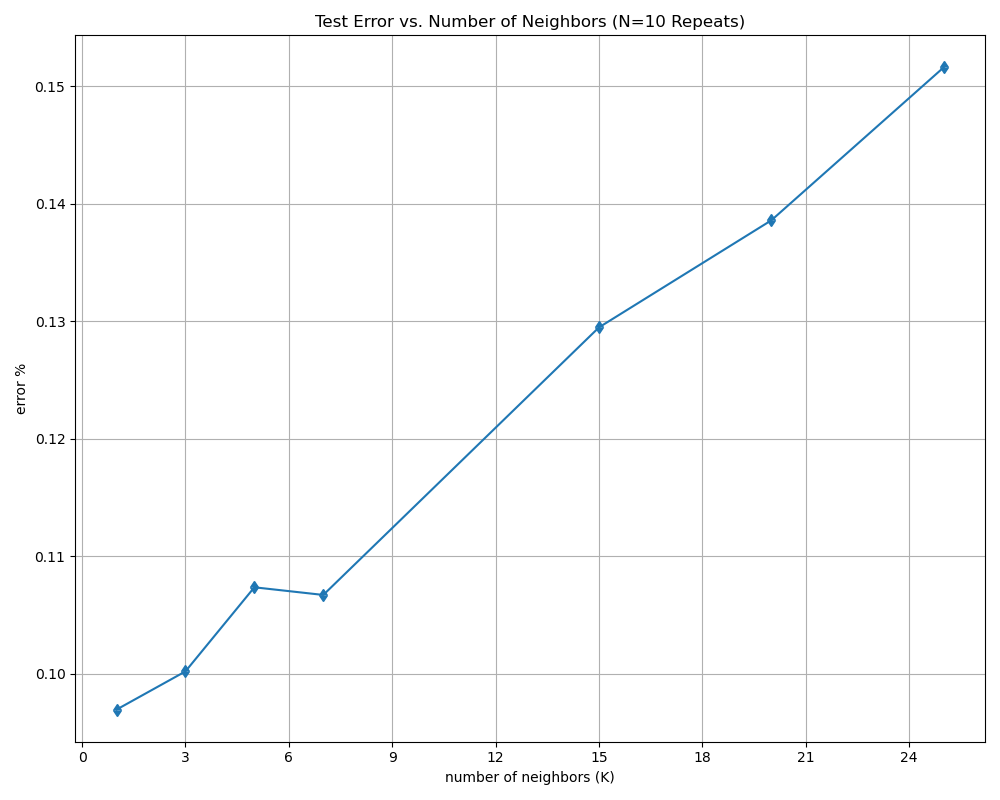
['Vote' 'Avg\_environmental\_importance' 'Yearly\_IncomeK'

'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.



**GradientBoost**

clf\_gradient **=** GradientBoostingClassifier**(**n\_estimators**=**100**,** random\_state**=**0**)**

sbs **=** SFS**(**clf\_gradient**,**

k\_features**=**8**,**

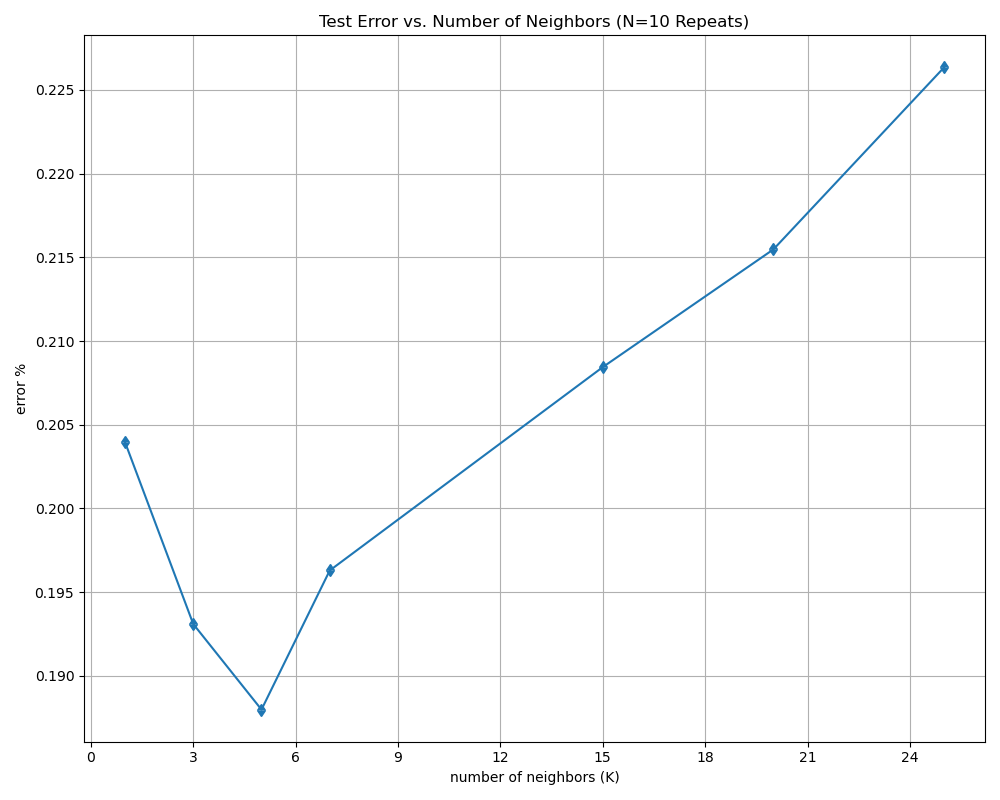
forward**=False,** # if forward = True then SFS otherwise SBS

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.