Mori Levinzon 308328467

Nadav Rubinstein 208686659

Dry – Data Preparation:

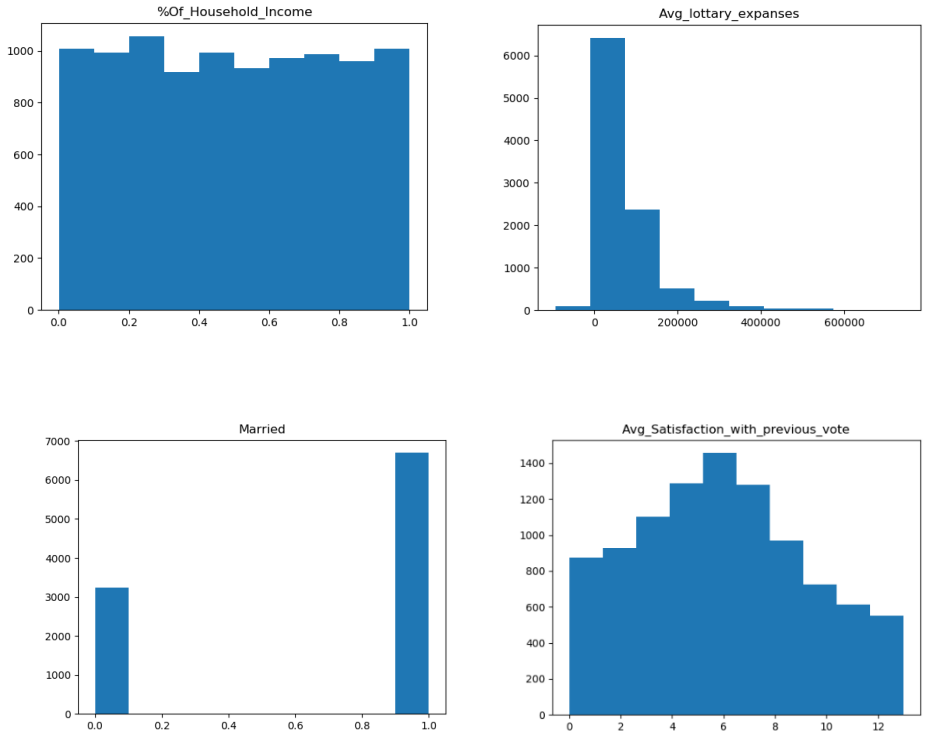
In this part we'll describe the actions that we made on the data given to us in order to prepare it for future leaning algorithm.

First we'll describe the process that made us understand better the data were working on:

1. **After we loaded the data we separated the features into two types: numerical features and nominal features**

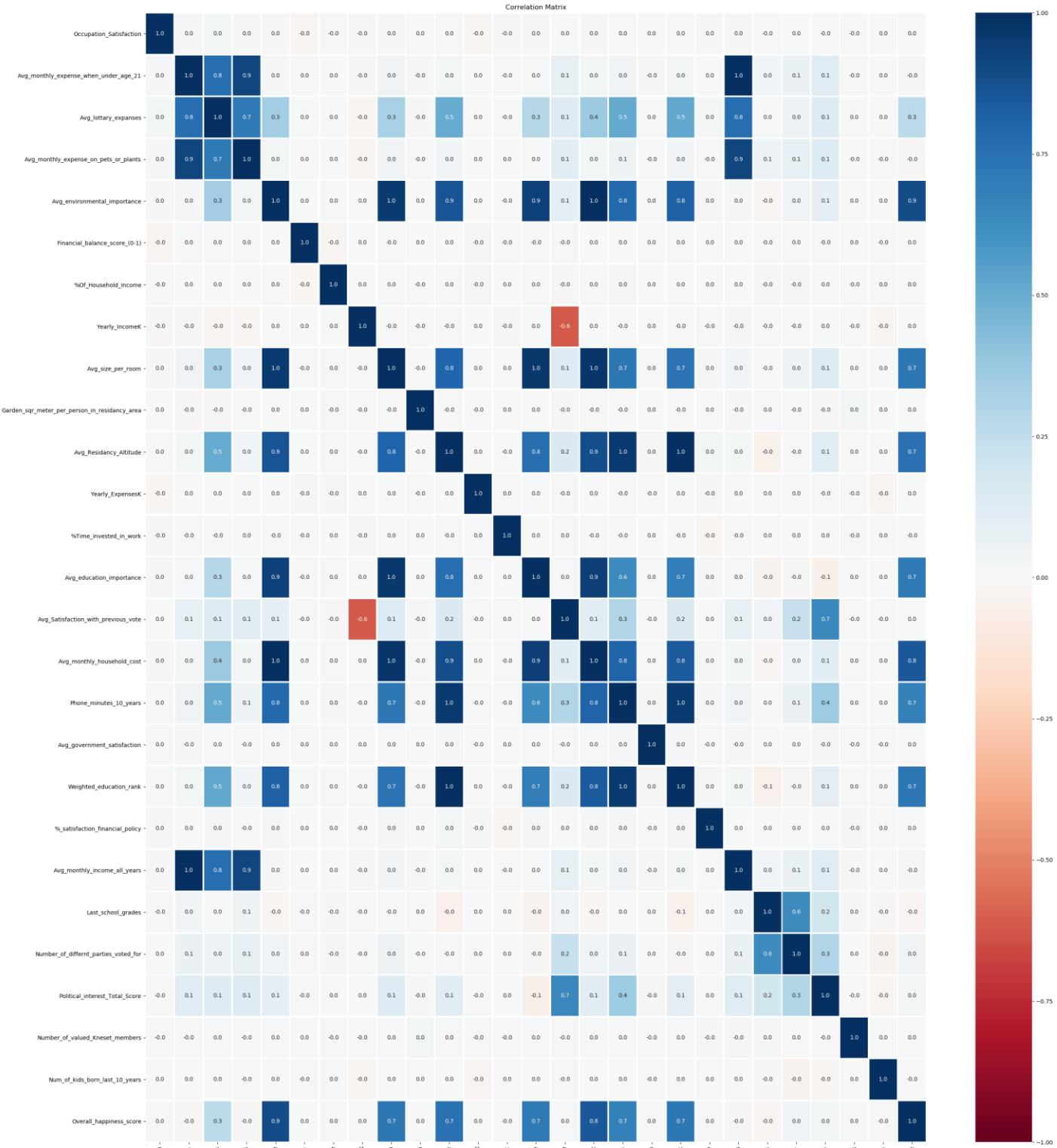
Numerical Features	**Nominal Features**
Occupation_Satisfaction	Vote
Avg_monthly_expense_when_under_age_21	Age_group
Avg_lottary_expanses	Looking_at_poles_results
Avg_monthly_expense_on_pets_or_plants	Married
Avg_environmental_importance	Gender
Financial_balance_score_(0-1)	Voting_Time
%Of_Household_Income	Will_vote_only_large_party
Yearly_IncomeK	Most_Important_Issue
Avg_size_per_room	Main_transportation
Garden_sqr_meter_per_person_in_residancy_area	Occupation
Avg_Residancy_Altitude	Financial_agenda_matters
Yearly_ExpensesK	
%Time_invested_in_work	
Avg_education_importance	
Avg_Satisfaction_with_previous_vote	
Avg_monthly_household_cost	
Phone_minutes_10_years	
Avg_government_satisfaction	
Weighted_education_rank	
%_satisfaction_financial_policy	
Avg_monthly_income_all_years	
Last_school_grades	
Number_of_differnt_parties_voted_for	
Political_interest_Total_Score	
Number_of_valued_Kneset_members	
Num_of_kids_born_last_10_years	
Overall_happiness_score	

2. In order to handle the data with more accuracy, we looked at the distribution of each of the features by making histograms of the values of each feature:



We can learn from the graph how the features are distibuted and which of the features are distibuted uniformly or normally.

3. We checked to see if there is a linear correlation between the features in case we need to fill missing data. We used a heatmap (supplied by the seaborn library) and generated a plot that showed for each pair of features their correlation:
Blue – positive correlation
Red – negative correlation
As the Absolute value of the value becomes greater so does the color gets darker.


Correlation Matrix
Occupation_Satisfaction
Avg_monthly_expense_when_under_age_21
Avg_lottary_expanses
Avg_monthly_expense_on_pets_or_plants
Avg_environmental_importance
Financial_balance_score_(0-1)
%Of_Household_Income
YearlyIncomeK
Avg_size_per_room
Garden_sqr_meter_per_person_in_residancy_area
Avg_Residancy_Altitude
YearlyExpensesK
%Time_invested_in_work
Avg_education_importance
Avg_Satisfaction_with_previous_vote
Avg_monthly_household_cost
Phone_minutes_10_years
Avg_government_satisfaction
Weighted_education_rank
%_satisfaction_financial_policy
Avg_monthly_income_all_years
Last_school_grades
Number_of_differnt_parties_voted_for
Political_interest_Total_Score
Number_of_valued_Kneset_members
Num_of_kids_born_last_10_years
Overall_happiness_score
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

4. From the names of the features we understood that their values must be non-negative.

The algorithm of the data preparation:

1. Load ElectionsData.CSV
2. Identify nominal features and map their values into integers
3. Split the data file into 3 separate files: train, validation and test for further changes.
4. Remove all negative values and change them to Nan
5. For every feature the distributed normally do standardization according the Z threshold and switch every value above 4.5 with Nan (needed because of the great gap between some values and the distributed values of some features).
6. Complete the missing values:
   - 6.1. For the features that had a high correlation (above 0.9) with other features we calculated the missing values according to effective linear coefficient between the two features.
   - 6.2. We used the closest fit algorithm for each set of data and filled missing values by the closest values.
   - 6.3. For each set we executed expectation maximization algorithm
   - 6.4. For the lasting remaining nominal values we completed the missing values to be the common value
   - 6.5. For the lasting remaining numeric values we set their values to be the average value of the feature.
7. Scaling:
   - 7.1. For features that distributed uniformly we scale them to the range [-1,1]
   - 7.2. For features that distributed normally we standardized according to Z
   - 7.3. Nominal features remained the same.
8. Feature selection:
   - 8.1. Filter method: for the numeric features we removed the features that their variance was small than the threshold 0.2 since those features didn't contributed more information therefore can be removed. The features that were removed were:
     - Avg_education_importance
     - Avg_environmental_importance
     - Avg_Residancy_Altitude
   - 8.2. Wrapper method: By using SGD-Classifier and selectKBest from the sklearn library we were able to found the group of features with the utmost

mutual information and with the size of 16 features which benefit that information.

The chosen features are:

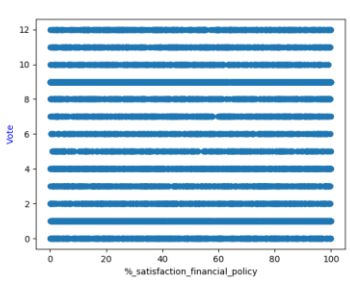
- Avg_monthly_expense_when_under_age_21
- Avg_lottary_expanses
- Avg_Satisfaction_with_previous_vote
- Avg_monthly_expense_on_pets_or_plants
- Avg_monthly_household_cost
- Phone_minutes_10_years
- Avg_size_per_room
- Weighted_education_rank
- Avg_monthly_income_all_years
- Last_school_grades
- Number_of_differnt_parties_voted_for
- Political_interest_Total_Score
- Overall_happiness_score
- Married
- Most_Important_Issue

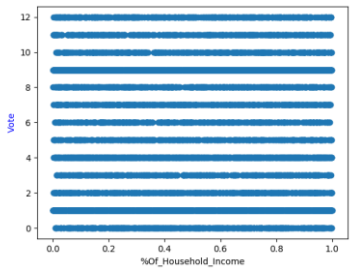
9. Saved the changed data sets (3) to a new csv files with their selected feature and fixed values.

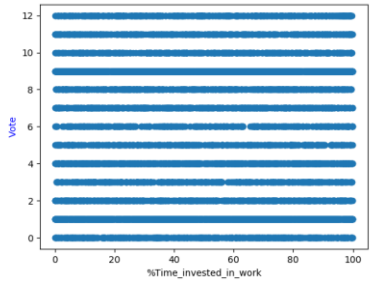
First Bonus – Relief Algorithm:

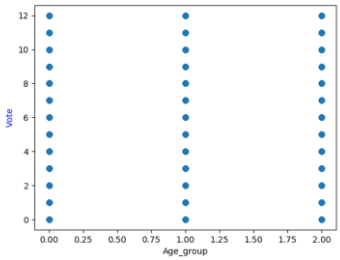
1. To identify the role of the attributes with respect to the Vote label, first we mapped the values of the vote label into numeric values:
   'Blues': 0,
   'Browns': 1,
   'Greens': 2,
   'Greys': 3,
   'Khakis': 4,
   'Oranges': 5,
   'Pinks': 6,
   'Purples': 7,
   'Reds': 8,
   'Turquoises': 9,
   'Violets': 10,
   'Whites': 11,
   'Yellows': 12

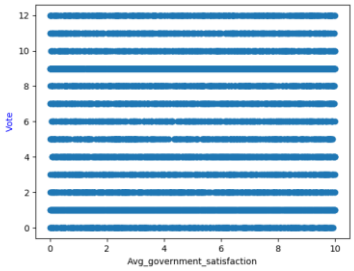
   Features we couldn't Conclude a connection between them to the vote label are those features that their values distributed uniformly over the vote label:

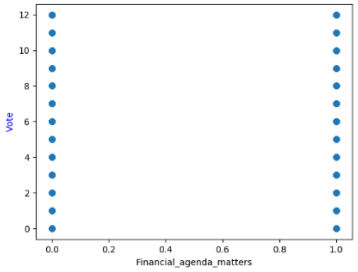


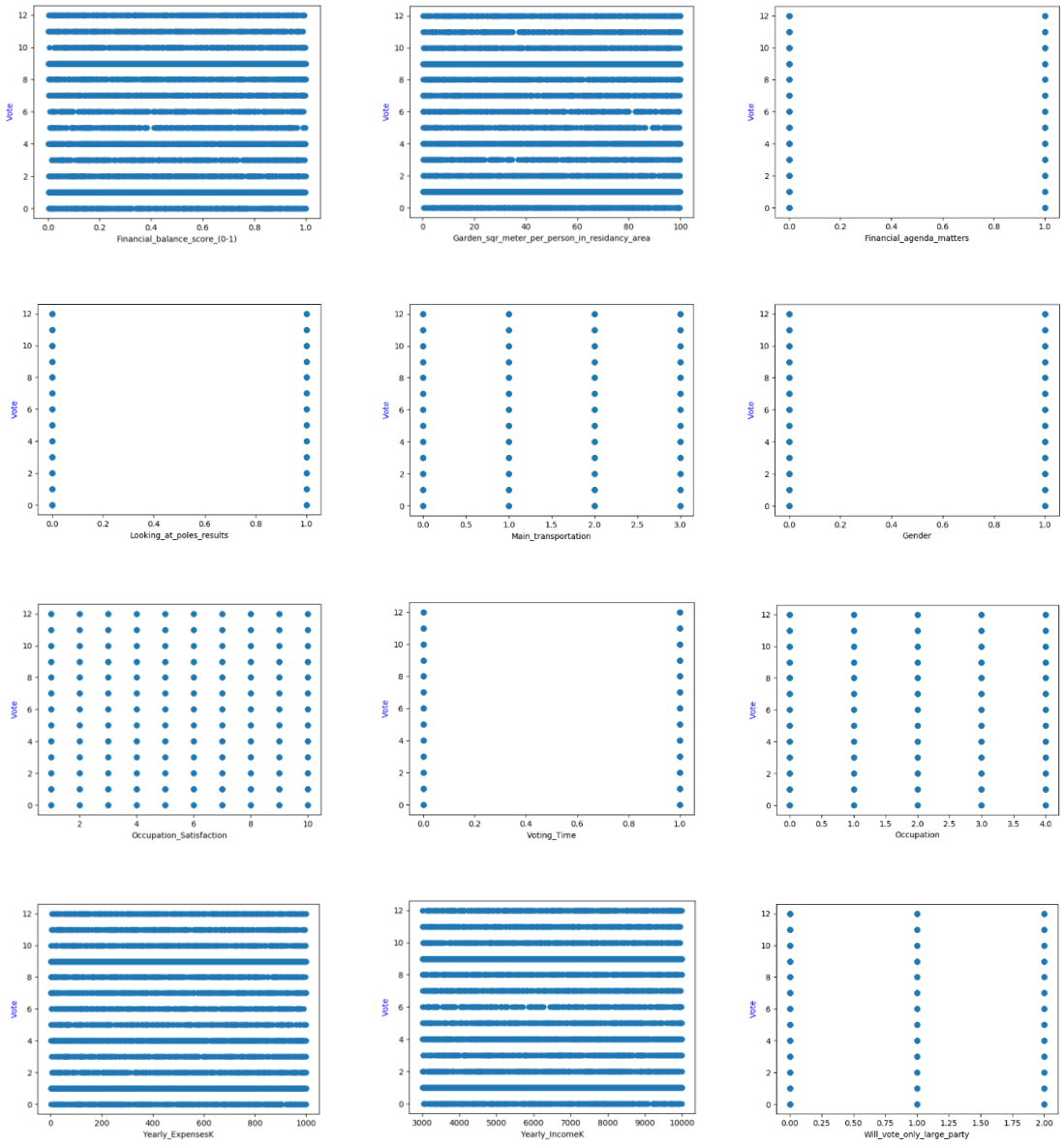






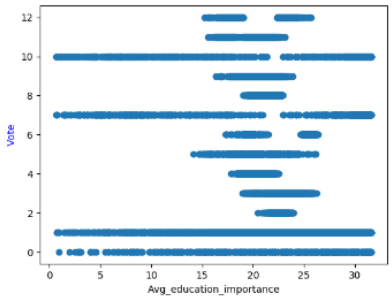




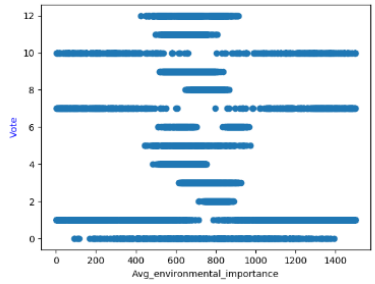


For these features we found a connection to the vote label:

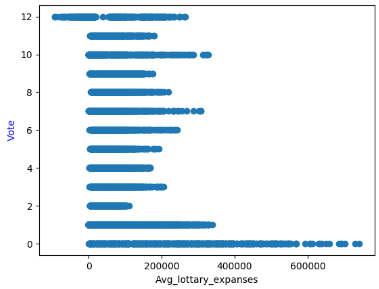
- Avg_education_importance: the Blue, Brown,Purpule and Violets are common among the entire range while the other colors are common only among the avg values



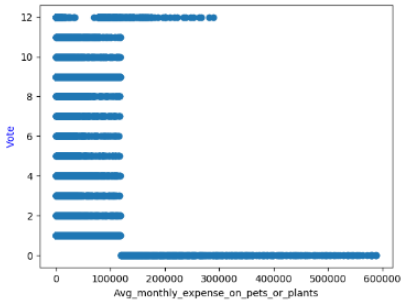
- Avg_environmental_importance: the Blue, Brown,Purpule and Violet are common among the entire range while the other colors are common only among the avg values



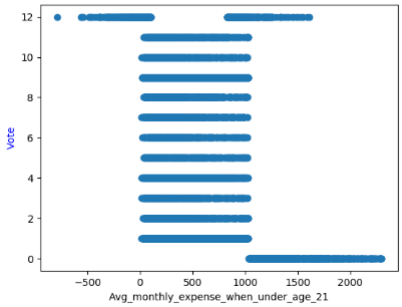
- Avg_lottary_expanses: people with high expanses tend to vote blue while people with low expanse tend to vote yellow.



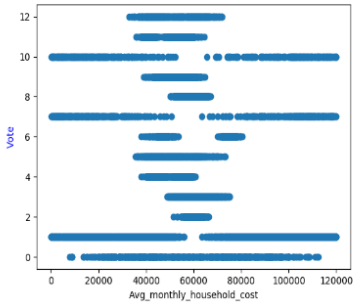
- Avg_monthly_expense_on_pets_or_plants: people with high expanses tend to vote blue while people with low expanse tend to vote yellow.



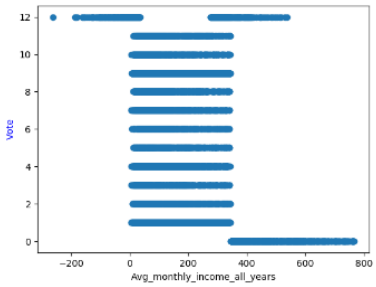
- Avg_monthly_expense_when_under_age_21: people with high expanses tend to vote blue while people with low expanse tend to vote yellow.



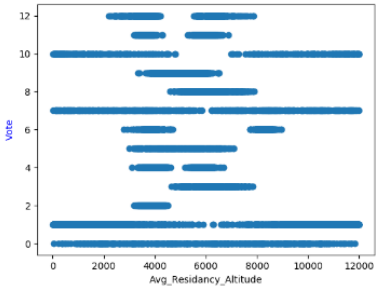
- Avg_mothly_household_cost: the Blue, Brown,Purpule and Violet are common among the entire range while the other colors are common only among the avg values



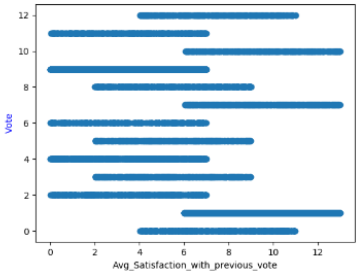
- Avg_monthly_income_all_years: people with high expanses tend to vote blue while people with low expanse tend to vote yellow.



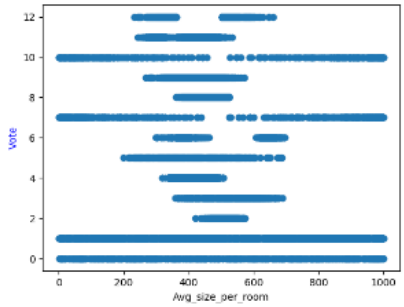
- Avg_Residancy_Altitude: the Blue, Brown,Purpule and Violet are common among the entire range while the other colors are common only among the avg values.



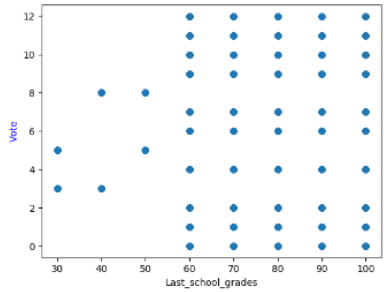
- Avg_Satisfation_with_previus_vote: for each group of satisfaction levels there is a different types of colors they choose.



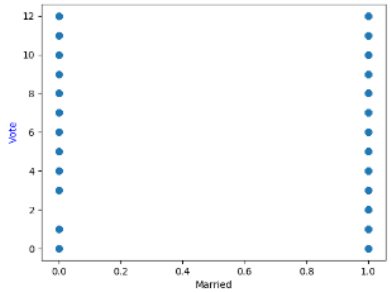
- Avg_Size_Per_Room: the Blue, Brown,Purpule and Violet are common among the entire range while the other colors are common only among the avg values.



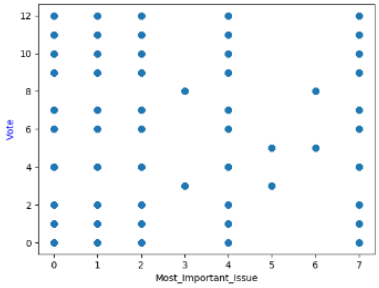
- Last_School_grades: people with low school grades tend to vote to grey, orange and red.



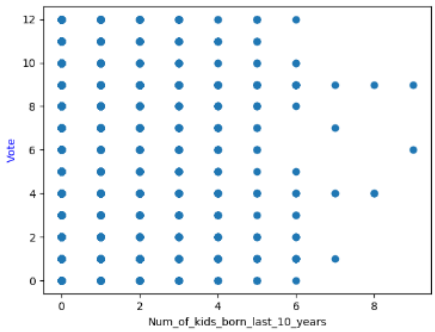
- Married: single people tend to not chose green.



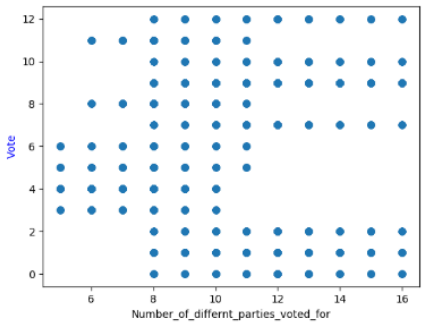
- Most_Important_Issue: the red, orange and grey are common among specific values.



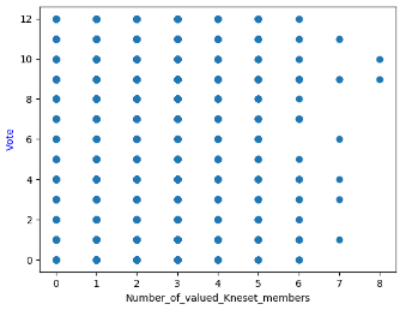
- Num_of_kids_born_last_10_years: families with high number of small children tend to chose brown, khaki, pink' purple and turquoises.



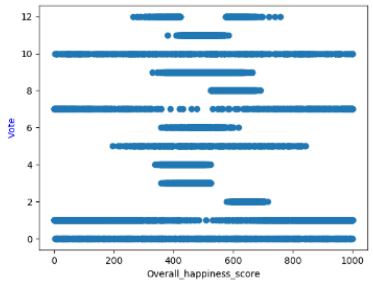
- Number_of_diferrent_parties_voted_for: mixed trend among high and low values.



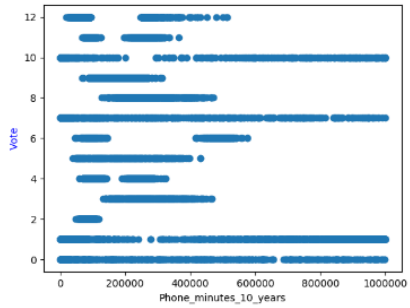
- Number_of_valued_kenset_members: the high values tend to choose specific colors such as: Turuoise pink and Khaki.



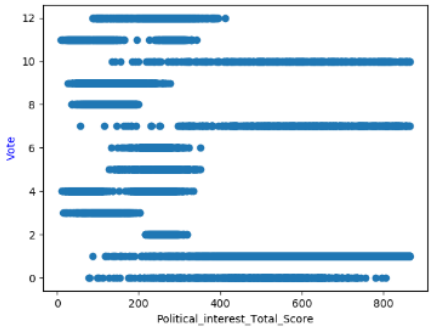
- Overall_hapiness_score: the Blue, Brown,Purpule and Violet are common among the entire range while the other colors are common only among the avg values.



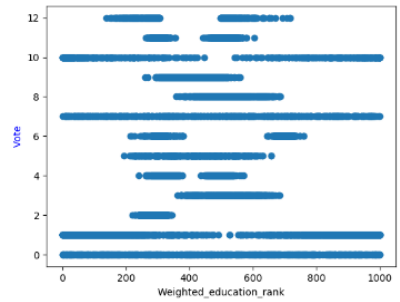
- Phone_minutes_10_years: the Blue, Brown,Purpule and Violet are common among the entire range while the other colors are common only among the lower values.



- Political_interest_total_score: the Blue, Brown,Purpule and Violet are common among the entire range while the other colors are common only among the lower values.



- Weighted_education_rank: the Blue, Brown,Purpule and Violet are common among the entire range while the other colors are common only among the avg values.



2. The relief algorythm is located at relief.py. The function gets the following parameters:
    - x_train: train data frame
    - y_train: train labels data frame
    - nominal_features_list: list of the nominal features
    - numerical_features_list: list of the numerical features
    - N , num_of_iterations: number of iterations
    - τ , threshold: threshold

    After n iterations the function relief return the selected features.

For τ=1 and N=1000 We got these selected features:

- 'Age_group'
- 'Gender'
- '%Of_Household_Income'
- 'Avg_Residancy_Altitude'
- '%Time_invested_in_work'
- 'Will_vote_only_large_party'
- 'Number_of_valued_Kneset_members'
- 'Occupation'
- 'Financial_agenda_matters'
- 'Overall_happiness_score'

Feature selection capabilities of Relief vs the other methods that we have used

Pros:

- Feature selection Can handle large amount of samples
- Features selection can be controlled by setting a threshold
- Using statistical analysis only
- Since we don't need to search after the best sub set of the feature and nor do we scale the set of the chosen features, the algorithm run at polynomial time.
- Easy to implement
- not affected from feature dependency

Cons:

- Not always clear what is the best threshold for optimal results
- Affected from calculations that based on samples distance, for example- every feature is given an equal weight when the distance is calculated (nominal feature that translated to integer value can change the distance although the distance between their values is insignificant)
- Need many iterations in order to get clear results.

Second Bonus – SFS Algorithm:

1. The algorithm was implemented in the file sfs.py in the function sfs_algo. It's parameters:
   - x_train: train data frame
   - y_train: train labels data frame
   - clf: classifier to examine
   - subset_size: user required subset size not mandatory

The algorithm is implemented using scoring of a Kcross fold validation (k=3) and returning the selected features. The selected geatures set is determined by the subset_size if some of the features improve the classification.

We examined the algorithm with two different classifiers:

- SGDClassifier:

SVM Classifier accuracy score before SFS is: 0.7687218468059891
SVM Classifier selected features are: ['Avg_Satisfaction_with_previous_vote', 'Yearly_IncomeK', 'Number_of_differnt_parties_voted_for', 'Last_school_grades', 'Phone_minutes_10_years', 'Avg_size_per_room', 'Avg_monthly_income_all_years', 'Political_interest_Total_Score', 'Avg_monthly_household_cost', 'Married', 'Looking_at_poles_results', 'Overall_happiness_score', 'Weighted_education_rank', 'Avg_education_importance', 'Financial_balance_score_(0-1)']
SVM Classifier score after SFS is: 0.7988705150609722

- KNN:

K Neighbors Classifier score before SFS is: 0.6159987327526332
K Neighbors Classifier selected features are: ['Weighted_education_rank', 'Avg_size_per_room', 'Overall_happiness_score', 'Last_school_grades', 'Number_of_differnt_parties_voted_for', 'Avg_education_importance', 'Phone_minutes_10_years', 'Avg_monthly_household_cost', 'Political_interest_Total_Score']
K Neighbors Classifier score after SFS is: 0.8465052560831677

feature selection capabilities of SFS vs the other methods that we have used:

Pros:

- Can deal with large amount of samples
- Can control the size of the selected features
- the selected features are determined by measures that can be changed by the user
- using classification algorithm we can create a learning curve that help us learn about the problem
- Easy to implement

Cons:

- We can't undo and reselect a feature even if it's completely redundant.
- Since the features are selected by the accuracy to the training set, there is a chance that there will be an over-fitting of the selected features.
- Algorithm run time depends on the number of features so it's preferable to run it on a small number of features.
- The algorithm chose to include a feature only if that feature improve the prediction accuracy and therefore doesn't take into account feature dependencies.
- The selecting features methodology is incremental thus it may not check other possibilities of other sub-sets of features that may have even higher accuracy level.