BOSTON UNIVERSITY

Classification of Sentiments of Somerville Residents

Team Members:

1) Sumit Dhaundiyal (U26474764)

2) Suraaj Shrestha (U05652372)

Data Mining goal

- Our goal is to predict the sentiment of the residents of Somerville based on the inputs given by the residents during the survey. We will classify if the residents are happy or unhappy based on 28 inputs out of 54 from the survey.
- We want to find the most relevant columns with respect to our class attribute so that we can predict the class attribute more efficiently using the reduced dataset which contains only the most important columns. Hence, reducing the runtime of the algorithm.
- We are also trying to find the most efficient algorithm (KNN, Decision Tree, Random Forest, Naïve Bayes, Neural Networks) to predict the class attribute for our dataset using cross fold validation and testing various scenarios based on the reduced dataset that we are generating using attribute selection methods (Pearson Correlation, Chi-Squared, Recursive Feature Elimination, L1-based feature selection, Tree-based feature selection).

Dataset Description

- A happiness survey is sent out by the City of Somerville every two years to a random sample of Somerville residents. Residents are asked to score their own happiness, well-being, and satisfaction with City services as part of the survey.
- The survey responses from 2011 to 2019 are included in this consolidated dataset. The Somerville Happiness Survey Responses Dataset contains rating given by residences based on happiness, how satisfied they are with Somerville as a place to live, neighborhood, how proud they are as Somerville residents, availability of information about comity services, coat of housing, public schools, local police and etc

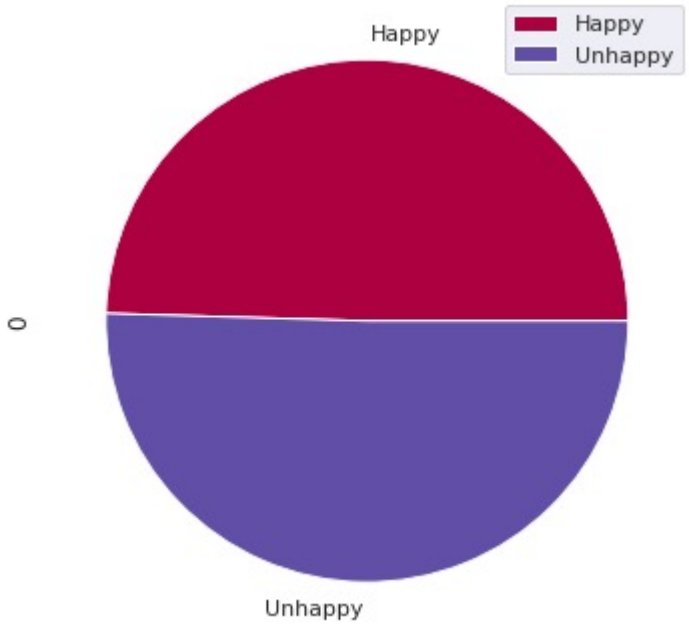
Pre-processing

- There were 8886 rows in total, some of which contained all null values, therefore we filtered data on the basis of following steps:
- For classification purpose we are considering the resident with 5 and above rating in the column 'How.happy.do.you.feel.right.now' as happy resident and unhappy otherwise. Hence, we added a new column named ‘Class’ which contains 0, if rating given is more than 4 and 1 otherwise
- Since we are using Random Forest algorithm (which requires a lot of data to train inorder to get good results), we checked the count of target variable, i.e., number of responses (Happy and Unhappy) since the count of ‘Unhappy’ responses are very low as compared to ‘Happy’ responses. We dropped rows that has at least 28 out of 29 column values empty.

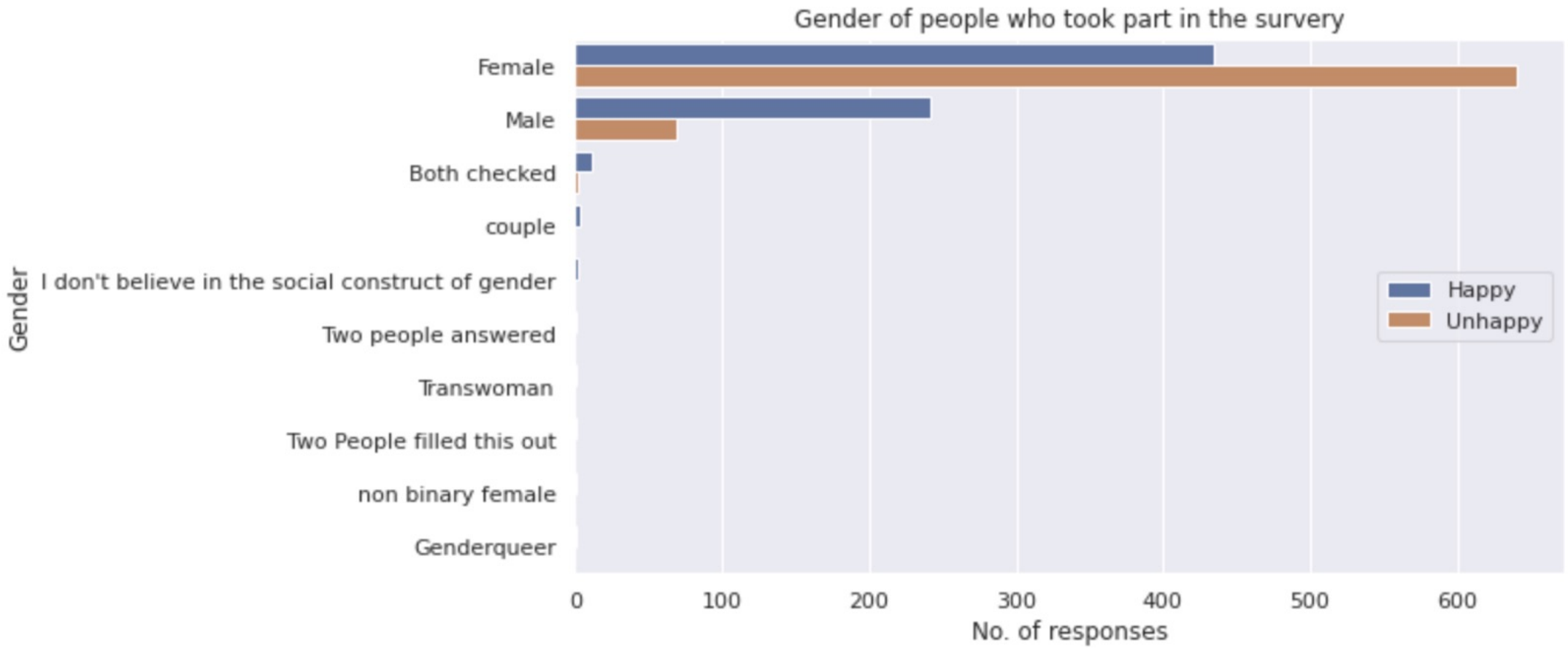
Class distribution of Responses

```
frames = [df_all_Happy,df_all_unhappy]
Somerville_happiness = pd.concat(frames)
Somerville_happiness['Class'].value_counts()
```

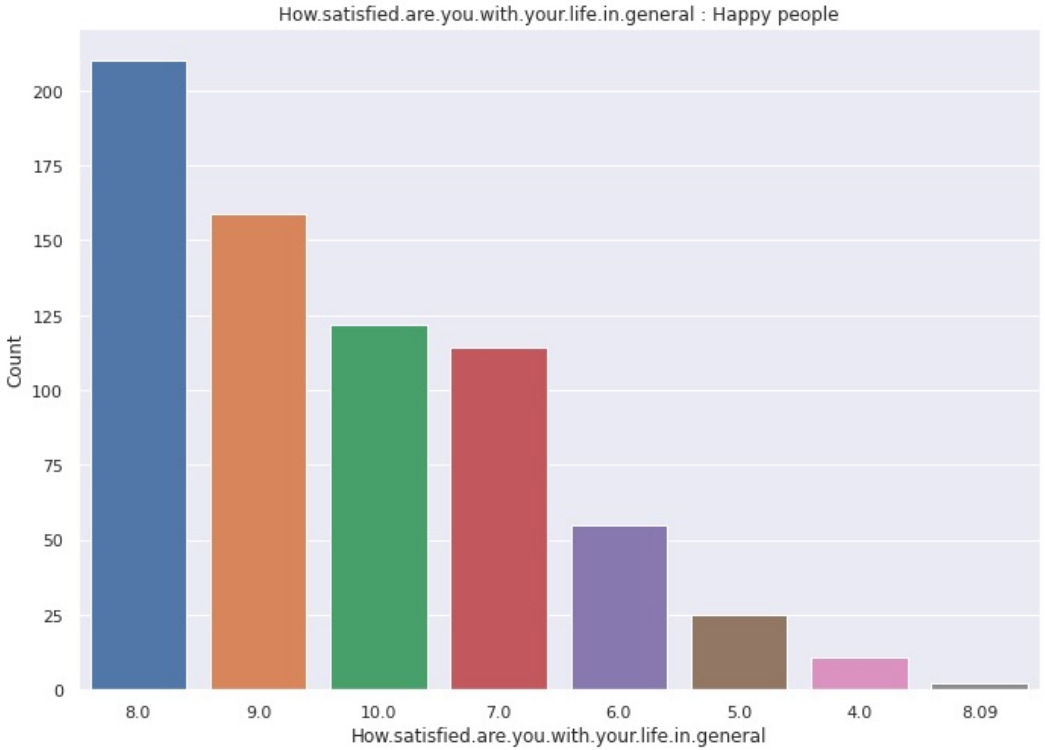
```
1    711
0    698
Name: Class, dtype: int64
```

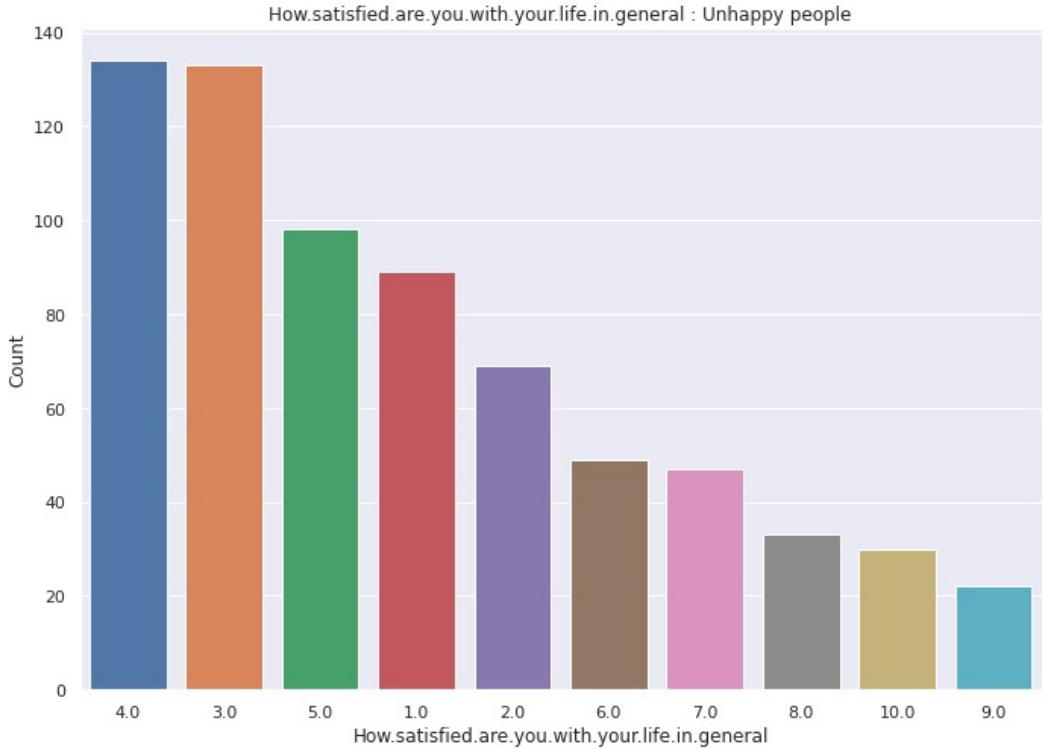


Response given by people of various genders

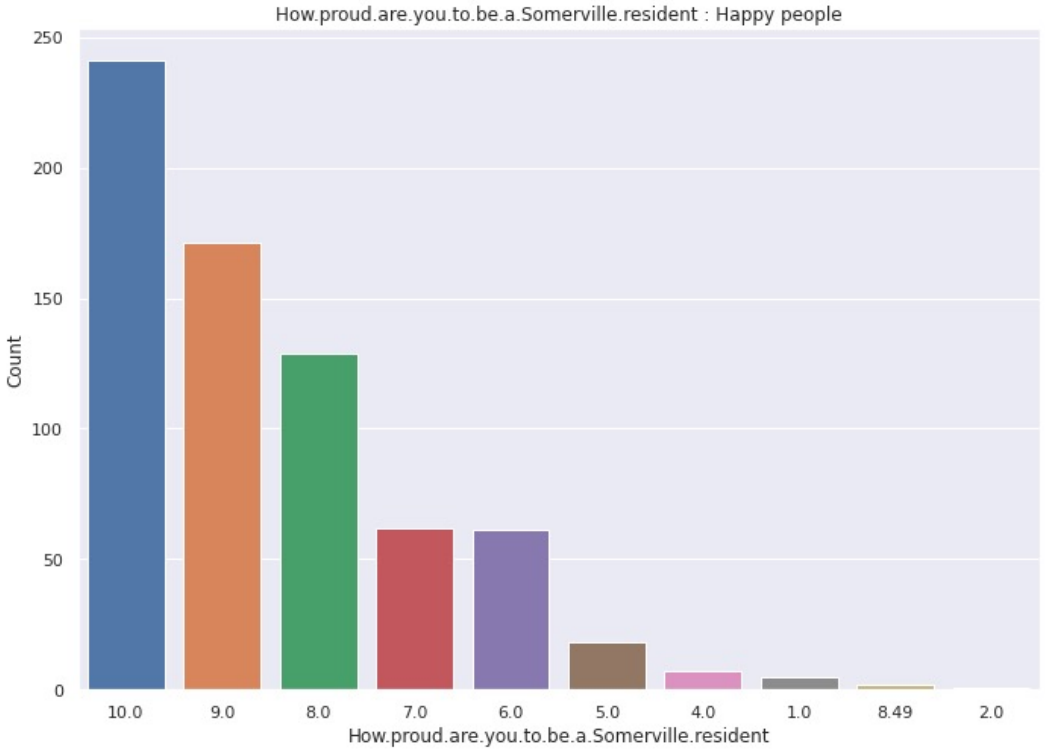


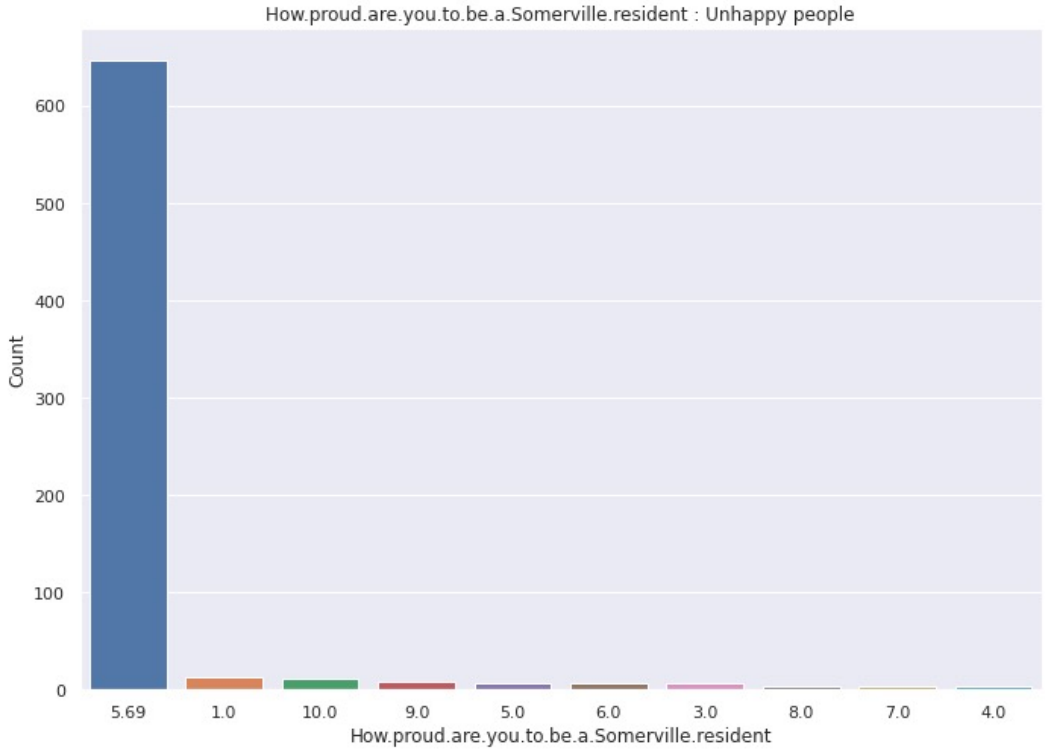
EDA


How.satisfied.are.you.with.your.life.in.general : Happy people
Count
0
25
50
75
100
125
150
175
200
8.0
9.0
10.0
7.0
6.0
5.0
4.0
8.09
How.satisfied.are.you.with.your.life.in.general


How.satisfied.are.you.with.your.life.in.general : Unhappy people
Count
0
20
40
60
80
100
120
140
4.0
3.0
5.0
1.0
2.0
6.0
7.0
8.0
10.0
9.0
How.satisfied.are.you.with.your.life.in.general

EDA Continues





Classification Algorithm Used

1. K Nearest Neighbors

- K-Nearest Neighbors is a machine learning technique and algorithm that can be used for both regression and classification tasks. It classifies the data based on the labels of its neighbors.
- K-Nearest Neighbors examines the labels of a chosen number of data points surrounding a target data point, in order to make a prediction about the class that the data point falls into. K-Nearest Neighbors (KNN) is a conceptually simple yet very powerful algorithm, and for those reasons, it's one of the most popular machine learning algorithms.

2. Decision Tree

- Decision Tree Analysis is a general, predictive modelling tool with applications spanning several different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on various conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
- The decision tree classifier creates the classification model by building a decision tree. Each node in the tree specifies a test on an attribute, each branch descending from that node corresponds to one of the possible values for that attribute

3. Random Forest

- Random Forest is a robust machine learning algorithm that can be used for a variety of tasks including regression and classification. It is an ensemble method, meaning that a random forest model is made up of a large number of small decision trees, called estimators, which each produce their own predictions. The random forest model combines the predictions of the estimators to produce a more accurate prediction.

4. Naïve Bayes

- A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
- Naive Bayes is the most straightforward and fast classification algorithm, which is suitable for a large chunk of data. Naive Bayes classifier is successfully used in various applications such as spam filtering, text classification, sentiment analysis, and recommender systems. It uses Bayes theorem of probability for prediction of unknown class

5. Neural Network - MLP Classifier :

- The multilayer perceptron (MLP) is a feedforward artificial neural network model that maps input data sets to a set of appropriate outputs. An MLP consists of multiple layers and each layer is fully connected to the following one. The nodes of the layers are neurons with nonlinear activation functions, except for the nodes of the input layer. Between the input and the output layer there may be one or more nonlinear hidden layers.

Attribute selection methods being used

1. Pearson Correlation : We calculated the Pearson's correlation between the Class variable and predictors. We keep the top 10 features which are strongly correlated with the class variables.

2. Chi-Squared: The chi-square metric between the class variable and the predictors is calculated in this procedure, and only the variable with the highest chi-squared values is chosen.

3. Recursive Feature Elimination:

- RFE is a feature selection method that fits a model and removes the weak feature (or features) until the provided number of features is reached. The model's coef_ or feature importance_ attributes rank features, and RFE seeks to minimize dependencies and collinearity in the model by recursively eliminating a small number of features per loop.
- We have used DecisionTreeClassifier()as Estimator instance

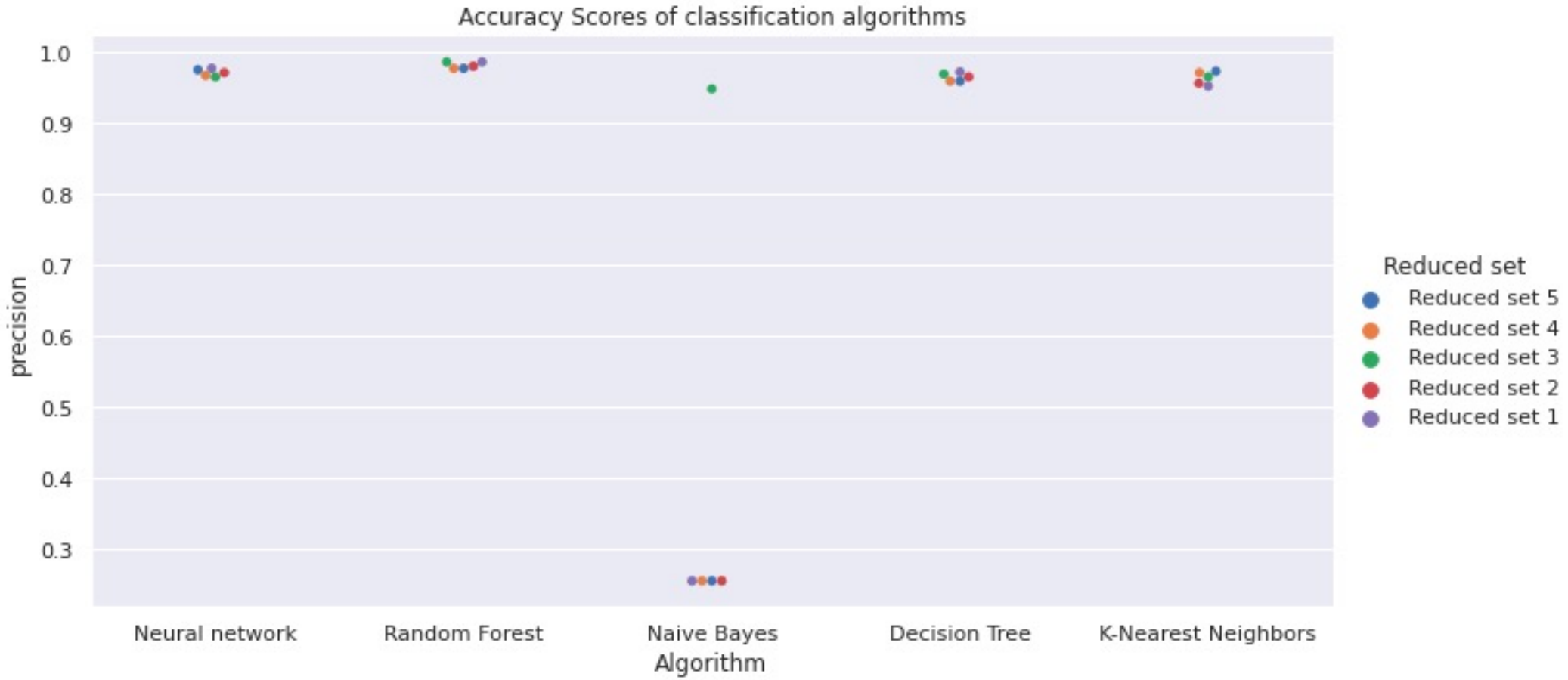
4. L1-based feature selection:

- Many of the estimated coefficients in linear models penalized with the L1 norm are zero, resulting in sparse solutions. They can be used in conjunction with SelectFromModel to choose non-zero coefficients when the purpose is to reduce the dimensionality of the data for use with another classifier. The Lasso for regression and LogisticRegression and LinearSVC for classification are examples of sparse estimators effective for this purpose.

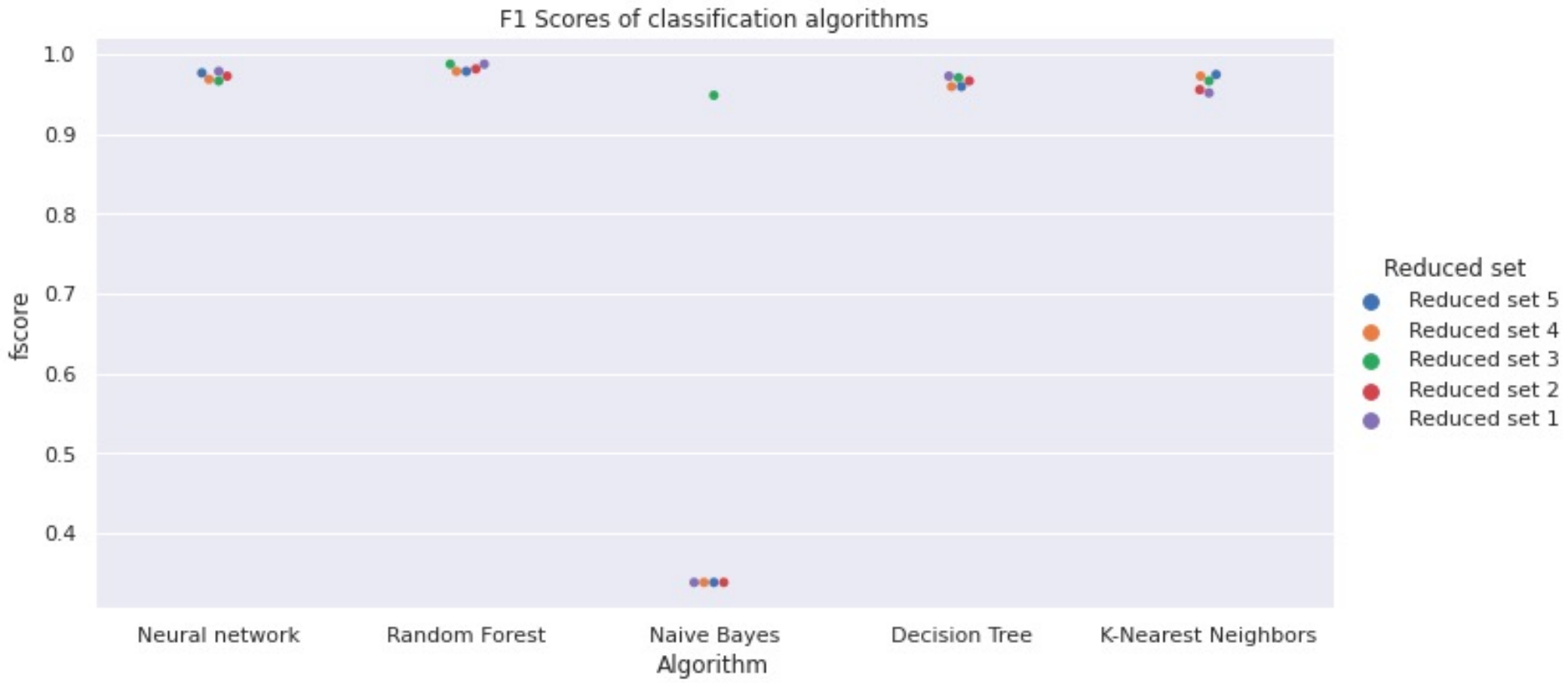
5. Tree-based feature selection:

- Impurity-based feature importances can be computed using tree-based estimators (see the sklearn.tree module and forest of trees in the sklearn.ensemble module), which can then be used to reject irrelevant features (when coupled with the SelectFromModel meta-transformer). We are using ExtraTreesClassifier model for the selection.

Results(Accuracy)



Results (F1 Score)



Results (Recall)

Conclusion

- The attribute selection method helped boost the performance of all the models.
- The attribute selection method - Pearson Correlation is only choosing the attributes that are strongly correlated with class attribute hence it is boosting the performance of all the models except Naive Bayes which is unexpected. It is same for the chi square method as well. An attribute selection method - Recursive Feature Elimination is selecting attributes based on DecisionTreeClassifier, hence it was to be expected that it will boost the performance of tree based algorithms. Which it is doing. The attribute selection method - Lasso: SelectFromModel is boosting the performance of all models but it is less for tree-based methods as compared to others. The last selection method is Tree-based: SelectFromModel, we were expecting a boost in random forest and decision tree but it is not the case which is unexpected.

Thank You