# Developer manual

Author: (Chen Zeng 152180)

## Data selection (at least 50 words)

I select all features except word frequency of George and area code 650 as these two features are the indicators of non-spam for a personalized filter as stated in the dataset documentation whereas we need a general purpose of spam filter. Other features in the dataset are well structured and were tested as good indicator of spam or non-spam. I also decide not to add new features because the dataset does not contain the data for the new features. For instance, if we add the features-word frequency of the word reward, we need to solve the problems like the dataset does not have the information of how many times reward occurs in those emails.

## Data Preprocessing (at least 50 words)

The attribute-word frequency of 1999 is replaced with word frequency of 2016 as the dataset was conducted in 1999 and we can use those data of 1999 for 2016 because the impacts of this kind of the number (current year) in filtering email does not change over time. The way to replace it is to change the column name of the dataset by using colnames () function. To remove the attributes of word frequency George and 650, I set the column of data frame as null by using emailspam$word\_freq\_george<-NULL. The other data in dataset are good without missing attributes values as stated in the dataset documentation so I did not process it.

## Data Transformation (at least 50 words)

I decide not to transform the data as the range of data is not broad. The reason is the nature of the attributes- word frequency count. The types of attribute in the dataset do not vary too much so the range of data is fine for classifying. I decide not to generalize the data also because the nature of the attributes- counting the word frequency which are the best described as numeral values and they do not need to putted into higher level concepts.

## Data Mining (at least 100 words)

In this process, Neural Network approach, k Nearest Neighbour approach, J48 Decision Tree, JRip Ripper, SMO SVM Classifier, and Part classifier are used for training. As the training dataset is large, use a low bias/high variance classifier such as kNN would be good since high bias classifier such as Naïve Bayes are not strong enough to offer accurate models. Utilizing SVM here is also good because it can work well even though the data isn’t linearly separable in the base feature space. It is particularly popular in text classification issues where high-dimensional spaces are the norm. I also select J48 decision tree model as it is simple to interpret and it can easily handle feature interactions and we do not need to concern about outlier or whether the data is linearly separable. Neural network is also used here as it is proficient to provide better classification by sing non linear boundaries. Moreover, neural network is simple to overcome overfitting by several regularized setting.

## Pattern Evaluation (at least 100 words with tables or graphs that represent each error rate or performance)

Following table outlines each classifier’s performances with its error rate and confusion matrix. As we can see from the table, J48 achieves the best accuracy which correctly classified 92.9146% instances. Following are the neural network, part, jrip approach. Their performance are similar, and neural network and part classifier achieved slightly better result than jrip- less than 0.02 percent. Although J48 obtain the best result among these classifier, it has no significant advantages than neural network and jrip. From the confusion matrix of SVM, we can observe that it correctly classified 2655 instances of non-spam email, which are the best result among the 6 classifiers yet the ability to classify spam email lower its overall performance. The worst result is obtained by knn approach, which is the only classifier whose correct instance are less than 90%. Through the analysis, j48 is defined as the best classifier in this dataset and will be used in the detect spam function.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | J48 | JRip | SVM |
| Correctly classified instances |  | 92.9146% | 92.5886% | 90.1326% |
| Error Rate |  | 7.0854% | 7.4114% | 9.8674% |
| Confusion Matrix |  | |  |  |  | | --- | --- | --- | | Non-spam | spam |  | | 2632 | 156 | Non-spam | | 170 | 1643 | spam | | |  |  |  | | --- | --- | --- | | Non-spam | spam |  | | 2650 | 138 | Non-spam | | 203 | 1610 | spam | | |  |  |  | | --- | --- | --- | | Non-spam | spam |  | | 2655 | 133 | Non-spam | | 321 | 1492 | spam | |
|  |  | Neural Network | KNN | PART |
| Correctly classified instances |  | 92.7589% | 81.3903% | 92.7842% |
| Error Rate |  | 7.2411% | 18.6097% | 7.2158% |
| Confusion Matrix |  | (30% of dataset)   |  |  |  | | --- | --- | --- | | Non-spam | spam |  | | 808 | 31 | Non-spam | | 69 | 473 | spam | | (30% of dataset)   |  |  |  | | --- | --- | --- | | Non-spam | spam |  | | 721 | 118 | Non-spam | | 139 | 403 | spam | | |  |  |  | | --- | --- | --- | | Non-spam | spam |  | | 2610 | 178 | Non-spam | | 154 | 1659 | spam | |