ASSIGNMENT ON DEVELOPMENT OF SPAM FILTER USING SUPPORT VECTOR MACHINE

INTRODUCTION

This project aims to analyse a data set for the classification of email as spam or not a spam and develop a model for the classification using the Support Vector Machine Algorithm. The algorithm uses the features available in the data set for the training of the model. The model is then tested upon the test dataset evaluated using different evaluation metrics.

This report gives a brief description about the data available with us, the processing done on the data, model developed and the accuracy of the model with the actual data.

DATA DESCRIPTION:

Dataset comes along with a documentation file which gives us idea about the dataset. The dataset is available in .txt format. In this data set there are 58 columns. 57 of these columns are the features and the last column is the target variable. Total rows available in the dataset is 4601.

When looked on the features closely, it is observed that there are 3 categories of feature set. First 48 columns are the frequency count of different words in the mail. The next 6 columns are the frequency count of the few particular characters and the last 3 columns of feature set are the lengths of the Capital letters in the mail.

As mentioned in the data documentation, False positive are highly undesirable and the misclassification error is expected to be within 7%.

The last column consisting of the target variable, classifies the mails as spam or not spam. The label 1 represents SPAM and 0 as NOT SPAM.

DATA PRE-PROCESSING:

The dataset available for the problem is in a separate .txt file and the column headers of the features are in a separate .txt file. So, first step was to merge the files and get a complete dataset with the column headers of the features. The target variable was given manually as “SPAM/NO SPAM”.

The X and Y arrays were created from the complete dataset for developing the SVM algorithm model. The complete dataset was then split into Train set and Test set with a ratio of 70%-30% respectively.

Normalisation: When we analysed the data, it was observed that the last 3 columns, giving data about the length of the capital letters are very large with respect to other values which lie in the range of 0-2. This can create an issue when developing the model. Hence it is important that the values of the complete dataset are brought into the same scale. For this reason, the data normalisation is done on the train and test set. For the normalisation, Standard normalisation is done and the parameters are obtained from the train dataset and applied on both the train and test dataset.

EXPLORATORY DATA ANALYSIS

The data is analysed for the mails that are marked as Spam. The mean of the columns is calculated and sorted in descending order. A very interesting data is revealed among the word and character frequencies among the mails labelled as Spam.

- Apart from the common words used for constructing a sentence like you, your, we, the word **"FREE"** and the character **"!"** is used for more than a **50%** of the content of the mails marked as spam. We can guess, this is done for attracting the attention of the users receiving the mails.
- The words **"BUSINESS"** has approximately **28%** and the words **"MONEY", "INTERNET"** and **"CREDIT"** is almost **21%** of the mail content, indicating that the majority of the spam mails try to lure the customers by something related to free money or free credit on the internet.
- Among the mails labelled not spam, the majority of the word frequencies are **"GEORGE", "HP", "ADDRESS", "MEETING"**, indicating that these were majorly personal mails addressed to some person "George" in the company HP(Hewlett-Packard).
- We can also see a large difference in the mean value of number of capital letters used in the mail content between the spam a not spam. In the spam mails, the number of capital letters used are much more than the non-spam mails, and this can be inferred as an action intended to grab attention of the receiver of the mail.

capital_run_length_total	161.470947
capital_run_length_longest	18.214491
capital_run_length_average	2.377301
word_freq_you	1.270341
word_freq_george	1.265265
word_freq_hp	0.895473
word_freq_will	0.536324
word_freq_your	0.438702
word_freq_hpl	0.431994
word_freq_re	0.415760
word_freq_edu	0.287184
word_freq_address	0.244466
word_freq_meeting	0.216808
word_freq_all	0.200581

NON-SPAM mails

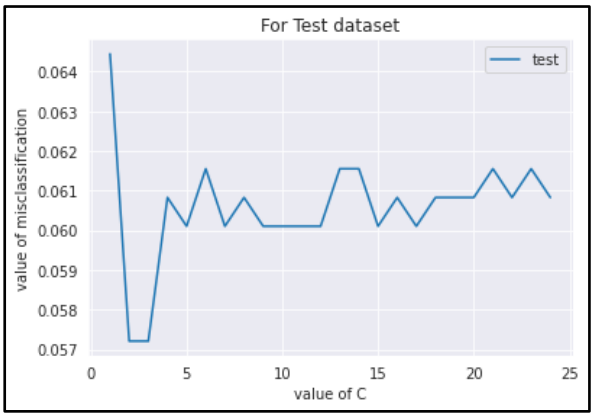
capital_run_length_total	470.619415
capital_run_length_longest	104.393271
capital_run_length_average	9.519165
word_freq_you	2.264539
word_freq_your	1.380370
SPAM/NO_SPAM	1.000000
word_freq_will	0.549972
word_freq_free	0.518362
word_freq_our	0.513955
char_freq_!	0.513713
word_freq_all	0.403795
word_freq_mail	0.350507
word_freq_email	0.319228
word_freq_business	0.287507
word_freq_remove	0.275405
word_freq_000	0.247055
word_freq_font	0.238036
word_freq_money	0.212879
word_freq_internet	0.208141
word_freq_credit	0.205521

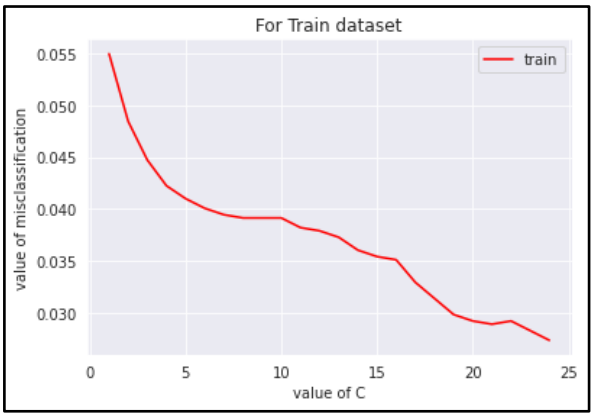
SPAM mails

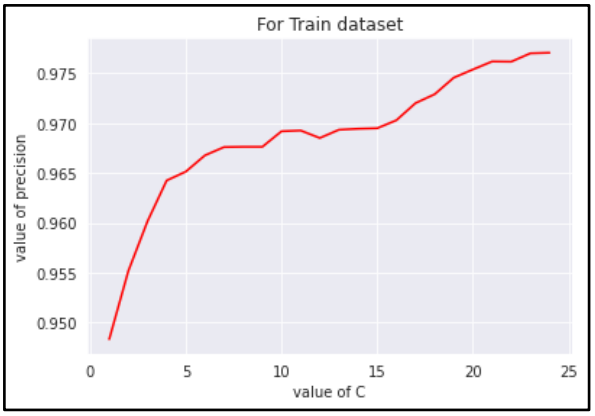
MODEL DEVELOPMENT AND EVALUATION

The Machine Learning Model with Support vector Algorithm is developed in Python. Multiple models were developed, for different value of regularisation parameters. For different values of C, the regularisation parameter, the precision and misclassification error were calculated by testing the model on the test dataset and recorded.

Finally for different values of C vs the Precision and Misclassification error were plotted and the following graphs were obtained for test data and train dataset.







From the above plot of the regularisation parameter vs the precision and misclassification error, we can observe:

- For the train set, the precision is continuously increasing and misclassification error is decreasing as the value of C increases.
- But for the test data set, we can observe that we get the best value of the precision and misclassification for some particular value of C, after which the results deteriorate. This is explained by overfitting. As the value of C increases, the model becomes less and less generalised and becomes more and more particular for the train dataset, as a result the bias decreases but the variance increases.
- Best value obtained are: **for C = 2, misclassification = 0.057, for C = 3, Precision = 0.953.** As mentioned in the documentation, False positives are most undesirable, hence we prioritise the best precision value and select the regularisation parameter value as C=3. For C=3, misclassification rate obtained is also 5.7%, which is less than 7%, hence it is acceptable.

CONCLUSION

Therefore, the best model obtained is for regularisation parameter value of 3, and it gives a precision of 95.3% and F1 score of .93 and accuracy of 94%.