**Email classification -Spam/Non-Spam**

**Prepared By,**

**Anjukutty Joseph**

Contents

[Objective 3](#_Toc13997)

[Data Set 3](#_Toc13998)

[Pre-Processing 3](#_Toc13999)

[Exploratory Data Analysis 4](#_Toc14000)

[Pre-Processing Continued 8](#_Toc14001)

[Model Creation 9](#_Toc14002)

[Model Selection 11](#_Toc14003)

[Model Evaluation 12](#_Toc14004)

[Appendix 13](#_Toc14005)

[References 13](#_Toc14006)

List of Figures

Figure 1: Count of spam and ham mails in training set .......................................................................... 5

Figure 2: Count of ham and spam mails in test set ................................................................................. 5

Figure 3: Top 20 words used in spam mails in training dataset .............................................................. 6

Figure 4: Top 20 words used in ham mails .............................................................................................. 7

Figure 5: Distribution of length of spam and ham mails ........................................................................ 8

Figure 6: Length of ham mails in training data ....................................................................................... 8

Figure 7: Length of spam mails in training data ...................................................................................... 8

Figure 8: Table Showing characteristics and performance of different Models .................................... 9

Figure 9: Model Comparison based on F1-Score .................................................................................. 11

Figure 10: Model comparison based on AUC ....................................................................................... 12

Figure 11:ROC Curve of selected model ............................................................................................... 13

# Objective

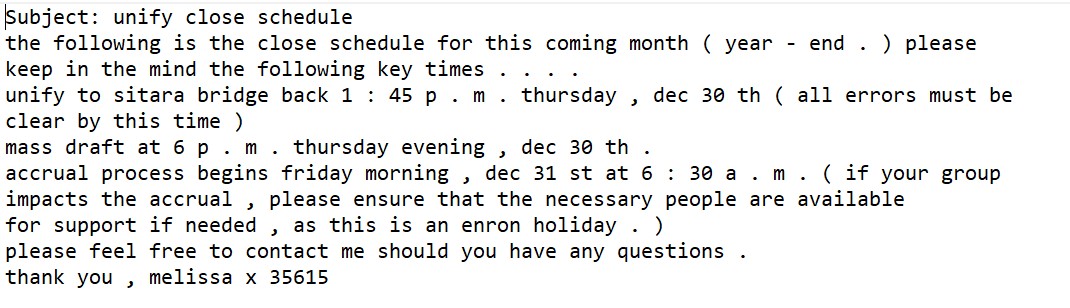
The objective of this project is to create a supervised classification pipeline model to classify an email as spam or non-spam using natural language processing techniques by making use of machine learning libraries in python such as scikit-learn, pandas etc**.**

## Data Set

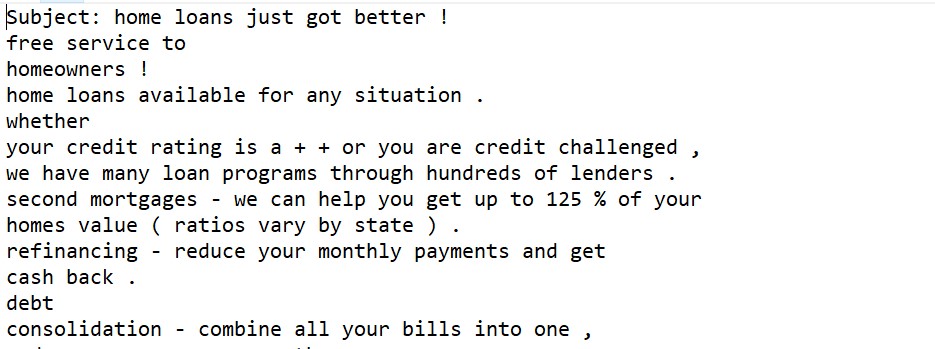
The dataset that we will use for this assignment is the Enron email dataset

[[http://www.aueb.gr/users/ion/data/enron-spam/]](http://www.aueb.gr/users/ion/data/enron-spam/) The dataset is a collection of public domain emails from the Enron corporation. For building a classification model I used pre-processed emails from enron1 folder which were already classified manually as spam and non-spam. There are 3612 emails in ham email folder and 1500 emails in spam folder. Below images shows a sample of how spam and non-spam emails look.

**Ham Email Sample**

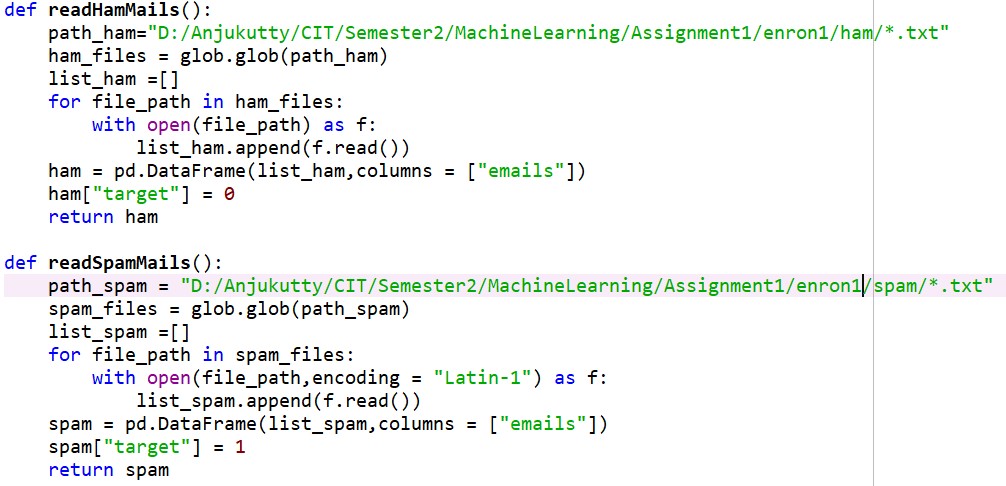


**Spam Email Sample**

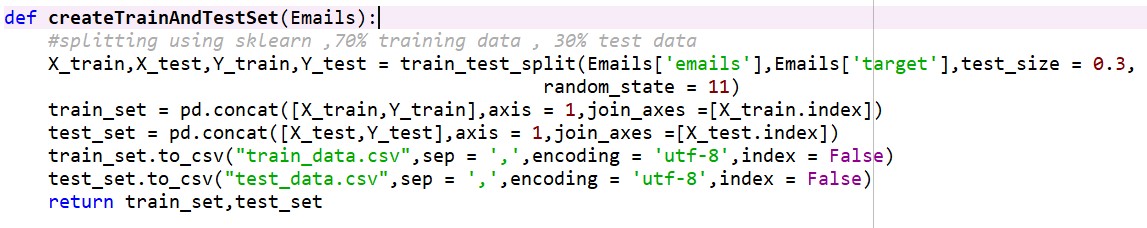


# Pre-Processing

As the first stage, the spam and non-spam(ham) emails were downloaded and extracted. The next task is to split the data for training and testing model. Using ***scikit-learn-train\_test\_split()*** function, emails are split in 70:30 ratio for model training and testing. Function ***readHamMails()*** read ham mails stored in local machine and create a python pandas dataframe. The dataframe has two columns ‘Email’ and ‘target’ where Email column store the email content and target column is assigned value 0 indicating the email is ham. Function ***readSpamMails()*** also does the same operations but target column has value 1 for indicating spam mails.

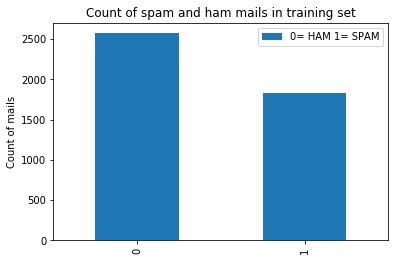


After forming ham and spam email data frame, ***createTrainAndTestSet()*** function shuffle spam and ham emails and split them in 70% train emails and 30 % test emails and contents in data frame are written to train\_data.csv and test\_data.csv to store in local machine.



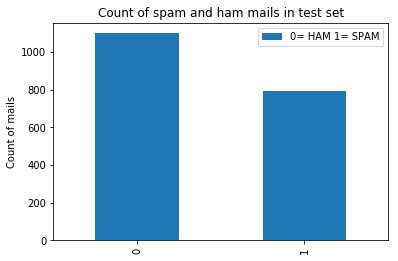
# Exploratory Data Analysis

Before proceeding with further pre-processing of data, to become accustomed with the data and gain insights into the kinds of features may be useful for classification following exploratory data analysis are being done. Three aspects of data set are explored here which are - count of ham and spam, find top 20 words used in mails and length of email messages. Functions *explDataAnalysis()* and *compareHamSpamMailLength() contains the code to produce these visualisations.*



### Figure 1: Count of spam and ham mails in training set

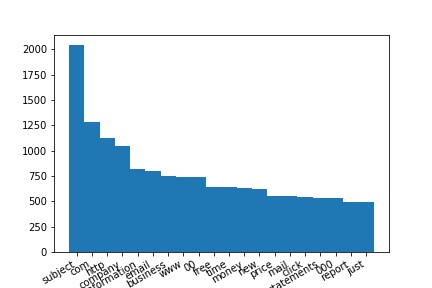
Figure1 above shows the count of spam and ham mails in training data. There are more ham mails than spam mails.



### Figure 2: Count of ham and spam mails in test set

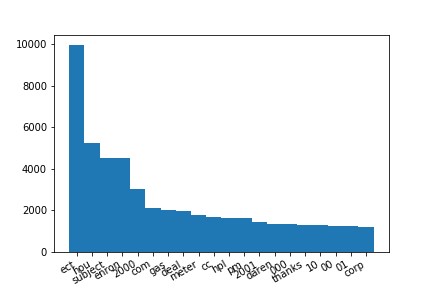
Figure2 shows the number of spam and ham mails in test set. Number of spam mails are less compared to ham mails.

To build a good classification model, understanding bag of words is critical. Below bar diagram shows the frequency of top 20 words used in spam and non-spam mails.



### Figure 3: Top 20 words used in spam mails in training dataset

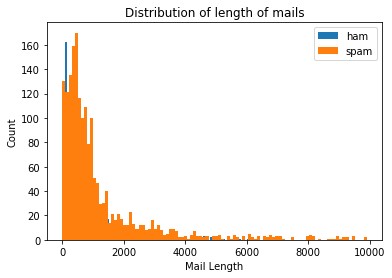
From the frequency plot in Figure 3 it can be seen that, words like ‘free’, ‘money’, ‘just’, ‘click’ are appearing more frequently. These words are commonly used in spam / promotional emails to attract people. Another noticeable thing here is the high frequency of word ‘subject’ and ‘com’. This can be due to the fact that, every email has word ‘subject’ by default as part of email structure.



### Figure 4:Top 20 words used in ham mails

Figure4 shows that frequency of top 20 words in ham mails. Word ‘ect’ appears top. Other noticeable words are ‘enron’, ‘com’, ‘hou’. There are numbers also coming in the top 20-word list. Number 2000 comes in top 5. It could be the year 2000 indicated in mails. Since both spam and ham mails have word ‘subject’ appearing common, I decided to remove this word from vocabulary during pre-processing. To create bag of words numbers are also removed during pre-processing the data

Next aspect investigated is the length of spam and ham emails. Box plot and histogram are made to compare the length of mails.



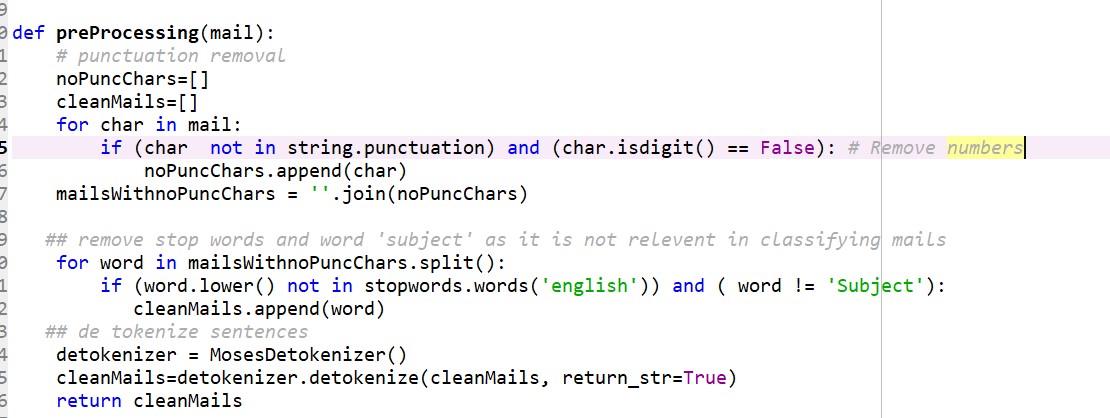
*Figure 5: Distribution of length of spam and ham mails*

It is clearly visible that, length of spam mails is significantly higher than ham mails. Below boxplots also conveys the same. However, there is one outlier in ham mails.

|  |  |
| --- | --- |
| *Figure 6: Length of ham mails in training data* | *Figure 7: Length of spam mails in training data* |

# Pre-Processing Continued

Since special characters and punctuations needs to be separated from the vocabulary I wrote below function ***preProcessing()*** function to eliminate those. This function also removes stop words like ‘is’, ‘a’ etc defined in python library as part of English language. Word ‘subject’ is also removed as it doesn’t feel to make a difference in classifying whether a mail in spam or non-spam. It also removes numbers from mails. This function iterates through each mail and take each character to check if it is punctuation or not. Only if it is not punctuation or number such characters are stored. These characters are again joined back to form words and detokenize to get back the original sentence after eliminating stop words and finally clean emails are returned.



# Model Creation

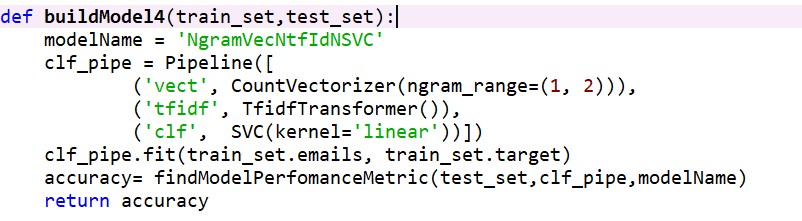
As the next step, 5 different models have been built using different features and algorithms with the help of Scikit package in Python. I tried different algorithms such as count vectorizer, multinomial Naïve Bayes, Tfid transformer, support vector machine to build models. I also decided to combine bag of words and length of emails to filter whether a mail is spam or not. Table shown in figure 7 indicates details of each model outlining the features, methods and accuracy in terms of F1Score and area under the curve.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Model-Name** | **Pipe Line-**  **Components** | **Function**  **Name In**  **Code** | **Featur es** | **F1Score** | **AUC** |
| 1 | CntVecNmNB | CountVectorizer, MultinomialNB | buildModel  1() | words | 0.57142  9 | 0.63888  9 |
| 2 | CntVecNtfidNmN  B | CountVectorizer,  TfidfTransformer,  MultinomialNB | buildModel  2() | words | 0.71428  6 | 0.77777  8 |
| 3 | CntVecNtfIdNSVC | CountVectorizer,  TfidfTransformer, SVC | buildModel  3() | words | 0.94736  8 | 0.91666  7 |
| 4 | NgramVecNtfIdN SVC | CountVectorizer,  TfidfTransformer, SVC | buildModel  4() | words | 0.94736  8 | 0.91666  7 |
| 5 | WordsNLength | CountVectorizer,  TfidfTransformer,  SVC,  FunctionTransform  er | buildModel  5() | Words & length of mail | 0.94736  8 | 0.91666  7 |

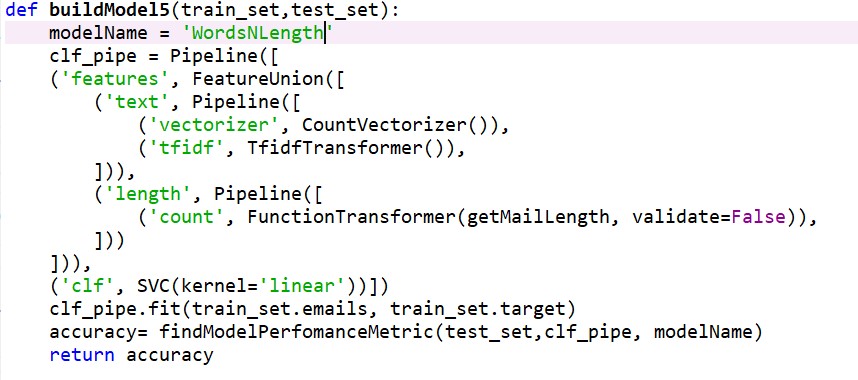
### Figure 8: Table Showing characteristics and performance of different Models

Python functions are wrote to build these models are given below. To keep it short, I am attaching only functions for model 4 & 5 in this report. Basically, all these functions accept two input arguments- train and test dataset and a classifier pipeline is built using ***sklearn.pipeline .*** The classifier is then trained on training set of emails and then used to predict and test accuracy of prediction on test dataset by calling function ***findModelPerfomanceMetric.***

Model4 classifier uses ngram range(1,2) , tfid transformer and SVM classifier.

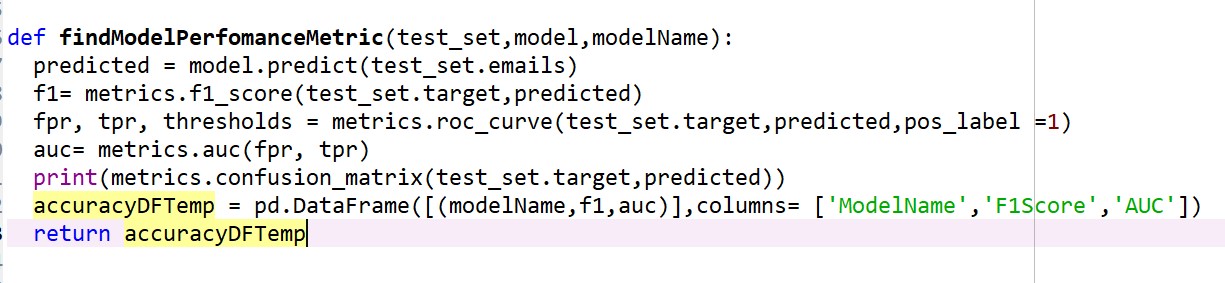


To build model5 classifier I used ***sklearn.FeatureUnion*** to add two levels of feature extraction. i.e. **Vocabulary as well as length of email is considered** for deciding whether a mail is spam or not.



In order to identify the accuracy of the trained models I used validation set approach. Thus***,*** function ***findModelPerfomanceMetric()*** predicts whether a mail in held out test set is spam or ham. The package used is ***sklearn.metrics***. I used 3 performance metric ***– F1Score, Confusion matrix and AUC.***

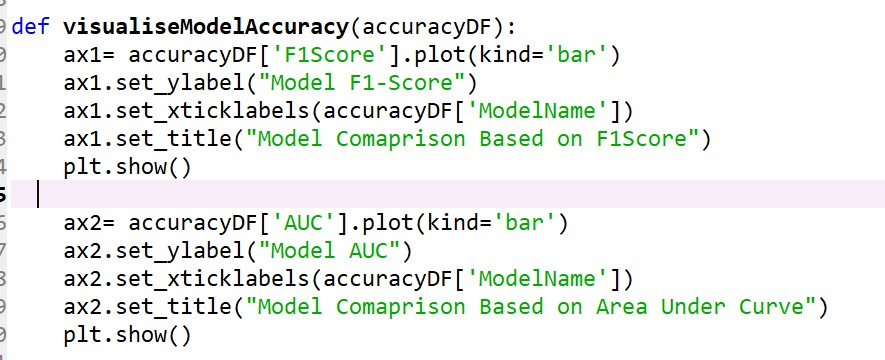
The accuracy measures are then added to a data frame for future comparison and to select the best model.

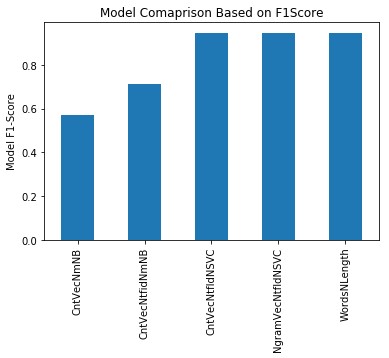


# Model Selection

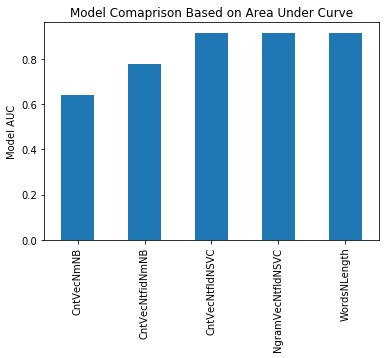
The target at this stage is to pick a best classifier that can give maximum accuracy for finding spam and ham mails in future production data set. To accomplish this, following bar charts are plotted which compare the F1- Score and AUC of 5 different models.

The function *visualiseModelAccuracy(*) takes a dataframe ‘accuracyDF’ which has stored the F1 score and AUC of 5 models and plot it in a bar chart.





***Figure 9: Model Comparison based on F1-Score***



### Figure 10: Model comparison based on AUC

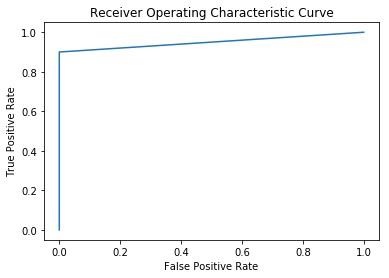
It can be seen that, model 3,4 and 5 perform good on test dataset in comparison with model1 and 2 which uses only count vectorizer and Naïve Bayes for classification. Applying TFID transformation on bag of words improves F1Score and AUC in model 2. It is interesting to see that, model 5 which combines vocabulary and length of email for classification is not giving any better AUC than other models which uses only vocabulary. It could be the case that information gain on length of mail is less compared to vocabulary.

Even though model3,4 and 5 shows comparable performance, ***I would like to choose model5*** as I believe length of email along with vocabulary can give better prediction in future emails. EDA results also is emphasizing that length of spam email is more than ham emails.

# Model Evaluation

Figure 11 below shows ROC curve for model 5. The model covers 91% of area under curve which is good. Table below shows the performance of chosen model on test set.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Model-Name** | **Pipe Line-**  **Components** | **Function**  **Name in**  **Code** | **Featur es** | **F1Score** | **AUC** |
| 5 | WordsNLength | CountVectorizer,  TfidfTransformer,  SVC,  FunctionTransform  er | buildModel  5() | Words & length of mail | 0.94736  8 | 0.91666  7 |



***Figure 11:ROC Curve of selected model***

# Appendix

I defined each functionality of the program as methods. All these functions are invoked from *main(*) method. Please run the script ***EmailClassification\_R00171244.py*** directly, it will output everything including graphs, model accuracy, ROC curve. Please replace the variable ‘*path\_ham’* and ‘*path\_spam’* in methods *readHamMails*(), *readSpamMails*() function with the ham and spam email path on your local machine. The program sometimes takes more time to run to give output as it includes pre-processing with tokenising and de-tokenizing and training multiple models.

# References

1. <https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html>
2. [https://medium.com/@cristhianboujon/how-to-list-the-most-common-words-from-textcorpus-using-scikit-learn-dad4d0cab41d](https://medium.com/@cristhianboujon/how-to-list-the-most-common-words-from-text-corpus-using-scikit-learn-dad4d0cab41d)
3. <https://stackoverflow.com/questions/34029865/how-to-plot-bar-chart-for-a-list-in-python>
4. [https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-andexamples-bf025f872908](https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908)
5. [www.scikit-learn.org](http://www.scikit-learn.org/)
6. [www.stackoverflow.com](http://www.stackoverflow.com/)