Name: Salman Anwaar

Roll No I18-1410

Machine Learning for Data Science

Introduction

Now a day's email is necessary for register on any website, but there is a chance that these details can be misused for promotion and fake messages. You could skip through the spam messages but there is good chance that you could skip through the important messages from authentic senders as well. In this project we have to apply algorithms that best classify ham and spam email step by step from simple algorithm to complex algorithm and then compare their performance.

Data Set Detail

This dataset includes the text of emails messages along with a label in the form of filename indicating whether the message is spam or not. Junk messages are labeled spam, while legitimate messages are labeled ham. There were total of 2412 non spam emails and 481 spam emails. This is shown in figure below.

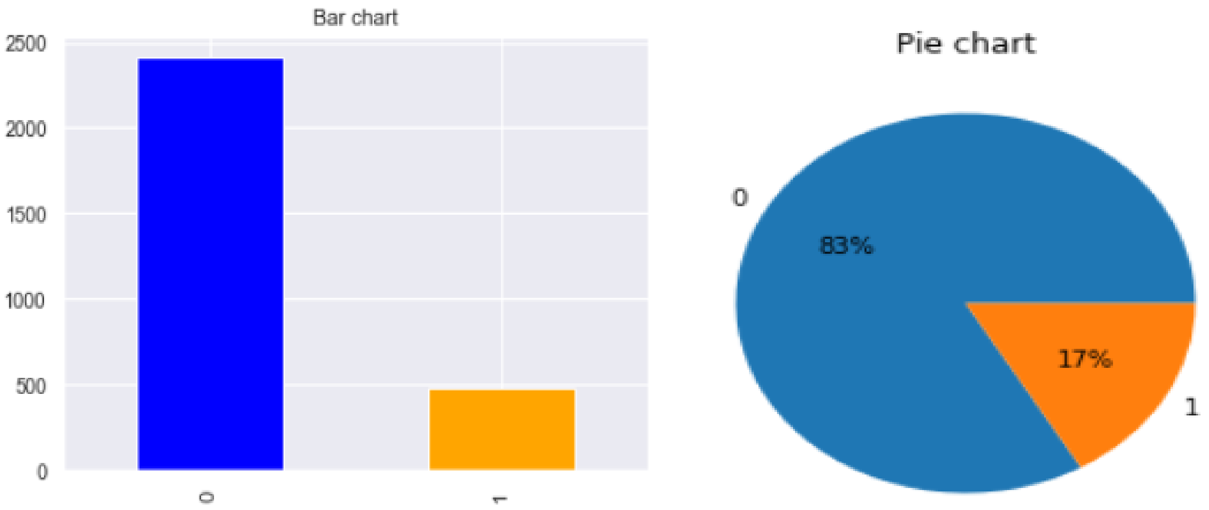


Figure 1 : Showing Class Distribution of Data

Word clouds are graphical representations of word frequency that give greater prominence to words that appear more frequently in a source text. We have combined all the ham email text in one variable and all the spam emails text in other variable and then applied word cloud, left picture words reflect non spam words, whereas right word cloud represent spam words.

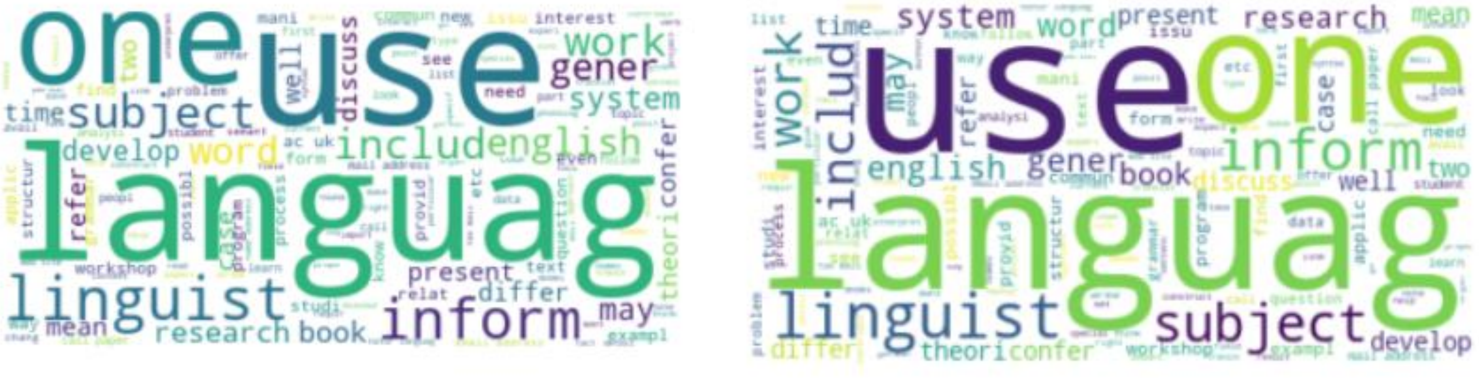


Figure 2 : Showing Word Cloud in Spam and Ham Emails. Left one is Ham and Right one is Spam

I used email dataset of samples by splitting the two sets explained below.

Training set

I used 80% of original data for training our model. For training, it is ensured that our data is fully mixed with equal composition of spam and non-spam emails. But we don’t have equal spam and non-spam emails.

Testing Set

Rest of the data 20% is used for testing data from our dataset, we cannot divide this data into validation set as our data set size is too small.

Error Analysis Techniques

Listed are the error analysis techniques which I have used for model evaluation. These are defined as below:

Precision:

In binary classification **precision** is refer to the positive predicted value from the given output of the trained model. This refers to the following formula:

$$Prsicion = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive(FP)}$$

Recall:

In binary classification **Recall** is refer to the sensitivity of fraction of relevant instances among the retrieved instances from the given output of trained model. This refers to the following formula:

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negitive(FN)}$$

F1 Score:

F1 Score In binary classification '**F1Score**' is refer to the comparison between above explained two terms which are precision and recall this refer to the following formula:

$$F1\ Score = \frac{2\ (Precision * Recall)}{Precision + Recall}$$

Accuracy Score:

Accuracy Score in binary classification **is** refer to overall, how often is the classifier correct?

$$Accuracy\ Score = \frac{True\ Positive\ (TP) +\ True\ Negative\ (TN)}{True\ Positive\ (TP) +\ False\ Positve\ (FP) + False\ Negative\ (FN) +\ True\ Negative\ (TN)}$$

BASE LINE MODEL:

Count Vectorizer -> TF-IDF -> ML Algorithms (Logistic, Svm and Decision Tree)

Count Vectorizer: Convert a collection of text documents to a matrix of word counts

TF –IDF: TF*IDF is an information retrieval technique that weighs a term's frequency (TF) and its inverse document frequency (IDF). Each word or term has its respective TF and IDF score. The product of the TF and IDF scores of a term is called the TF*IDF weight of that term.

ML Algorithms: I have used three algorithms for classification. (i) Logistic Regression (ii) SVM Linear Kernel (iii) Decision Tree

Variation in Preprocessing:

These are the variation which I have applied on dataset.

Iteration 1: Applied ML pipeline on unprocessed Data.

Iteration 2: Applied ML pipeline after removing special characters and digits on emails.

Iteration 3: Applied ML pipeline after removing special character, digits, stop words and applied stemming.

Iteration 4: Applied ML pipeline on preprocessed data by oversampling less percentage class.

Iteration 5: Applied ML pipeline on preprocessed data by under sampling high percentage class.

Results:

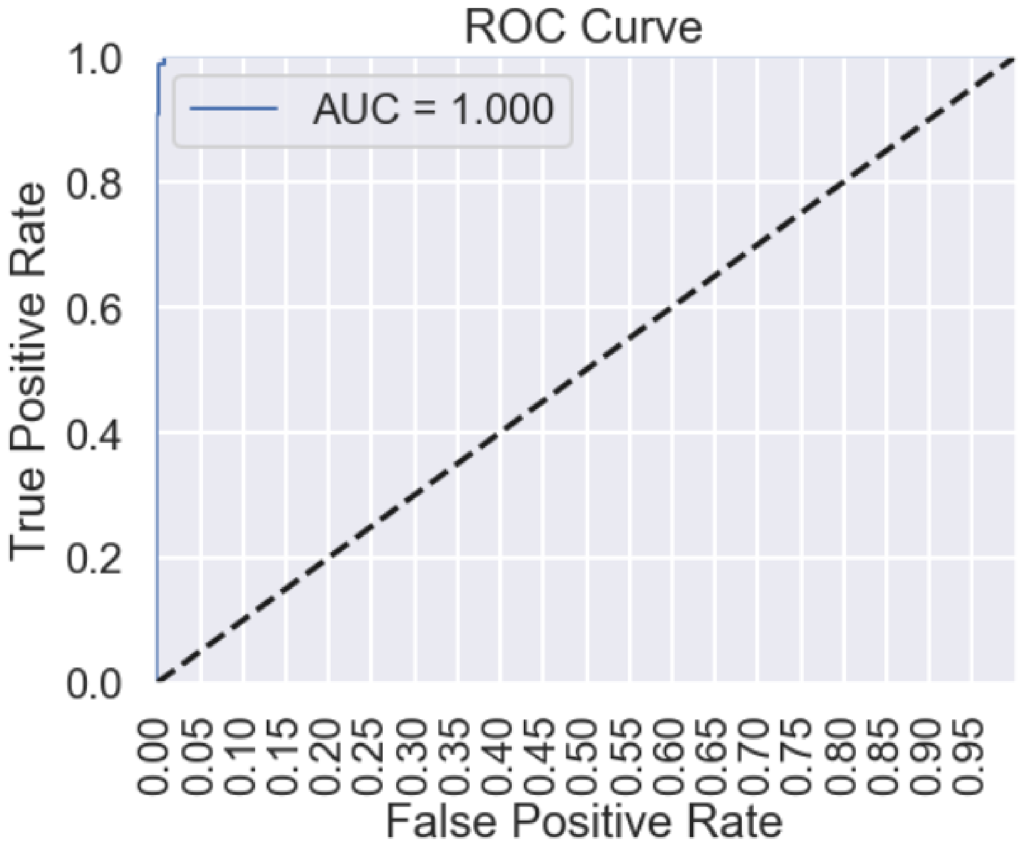
Models	Accuracy	Precision	Recall	F1 Score	AUC_ROC Score
On Un Processed Data					
Logistic Regression	0.9827288428324698	1.0	0.8837209302325582	0.9382716049382717	0.998726355016746
SVM Algorithm	0.9896373056994818	0.9761904761904762	0.9534883720930233	0.9647058823529412	0.9996226237086654
Decision Tree	0.9706390328151986	0.8709677419354839	0.9418604651162791	0.9050279329608939	0.9539600924571914
After Removing Special Characters And Digits					
Logistic Regression	0.9827288428324698	1.0	0.8837209302325582	0.9382716049382717	0.9987027689985377
SVM Algorithm	0.9896373056994818	0.9761904761904762	0.9534883720930233	0.9647058823529412	0.9996226237086654
Decision Tree	0.9740932642487047	0.9176470588235294	0.9069767441860465	0.9122807017543859	0.9463889806122929
After Lower Casing, Stemming Words and Removing Special Characters , Digits and Stop Words					
Logistic Regression	0.9671848013816926	0.9855072463768116	0.7906976744186046	0.8774193548387097	0.999481107599415
SVM Algorithm	0.9896373056994818	0.9878048780487805	0.9418604651162791	0.9642857142857143	0.9997405537997076
Decision Tree	0.9775474956822107	0.9397590361445783	0.9069767441860465	0.923076923076923	0.9484173781782158
Up Sampling Data For Dealing With Class Imbalance Problem					
Logistic Regression	0.9930915371329879	0.9880952380952381	0.9651162790697675	0.9764705882352942	0.999764139817916
SVM Algorithm	0.9896373056994818	0.9878048780487805	0.9418604651162791	0.9642857142857143	0.999764139817916
Decision Tree	0.9723661485319517	0.926829268292683	0.8837209302325582	0.9047619047619047	0.9357752724185103
Down Sampling Data For Dealing With Class Imbalance Problem					

Logistic Regression	0.9930915371329879	0.9880952380952381	0.9651162790697675	0.9764705882352942	0.999764139817916
SVM Algorithm	0.9896373056994818	0.9878048780487805	0.9418604651162791	0.9642857142857143	0.9997641398179159
Decision Tree	0.9689119170984456	0.8953488372093024	0.8953488372093024	0.8953488372093024	0.9385466295579981

Best Result Confusion Matrix:

	Predicted Not Spam	*Predicted Spam*
Actual Not Spam	492 (True Positives)	1 (False Positives)
Actual Spam	5 (False Negatives)	81 (True Negatives)

Best Result ROC_AUC Curve



:

Conclusion:

It is not always that changing in features improves accuracy, we can try more algorithms like Boost, Random Forest and Neural Net that can outperform SVM results. More over class imbalance was the problem, we synthetically make more data for more accuracy.