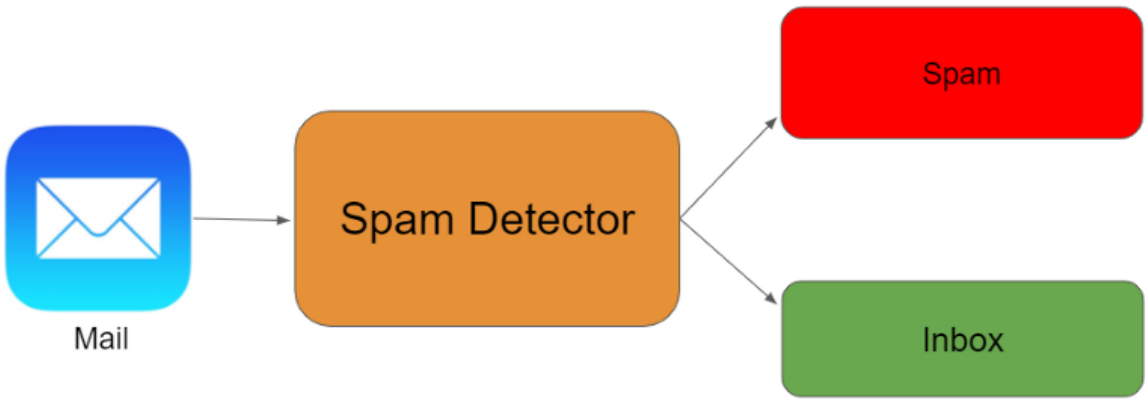
FLIP ROBO

EMAIL SPAM DETECTION



Submitted by:

RAMYASHREE R

ACKNOWLEDGMENT

Help to complete this project was greatly taken from classes and notes offered by data trained portal.
References were taken from previous evaluation projects done.
Some references were taken from google, Wikipedia, toward data science, Kaggle and GitHub

INTRODUCTION

- ## Business Problem Framing

In recent times, unwanted commercial bulk emails called spam has become a huge problem on the internet. The person sending the spam messages is referred to as the spammer. Such a person gathers email addresses from different websites, chatrooms, and viruses. Spam prevents the user from making full and good use of time, storage capacity and network bandwidth. The huge volume of spam mails flowing through the computer networks have

destructive effects on the memory space of email servers, communication bandwidth, CPU power and user time. The menace of spam email is on the increase on yearly basis and is responsible for over 77% of the whole global email traffic. Users who receive spam emails that they did not request find it very irritating. It is also resulted to untold financial loss to many users who have fallen victim of internet scams and other fraudulent practices of spammers who send emails pretending to be from reputable companies with the intention to persuade individuals to disclose sensitive personal information like passwords, Bank Verification Number (BVN) and credit card numbers.

- ## Conceptual Background of the Domain Problem

Now-a-days almost everyone will have an email id which would mean they would be using some or the other kind of email messaging client, each and every person in this list would have had experience with unwanted emails, in other words called as spam.
Detection of spam has become a social issue and it needs to be addressed diligently.

- ## Review of Literature

There was not much research done on this problem statement understanding, because lot of us are aware of this kind of problem and we can relate this with our day to day life. We do see many spams in our personal mailbox itself which has variety of content and evil ways to trick the receiver to take disadvantage of them because of which users will fall prey and lose things financially or emotionally.

- ## Motivation for the Problem Undertaken

This project has been undertaken by me because it was my a project assignment in my internship with FLIPROBO as an intern.

Analytical Problem Framing

- ## Mathematical/ Analytical Modeling of the Problem

Throughout the project multiple mathematical and analytical model have been used, first we have checked the ratio of spam and ham emails in our dataset.
Then we have used regular expressions to clean the data.
Used regular expressions to clean the message column which contained body of the email. We then used TfidfVectorizer , to transforms text to feature vectors that can be used as input to estimator.

- ## Data Sources and their formats

Dataset has been provided as part of the assignment.

	subject	message	label
0	job posting - apple-iss research center	content - length : 3386 apple-iss research cen...	0
1	NaN	lang classification grimes , joseph e . and ba...	0
2	query : letter frequencies for text identifica...	i am posting this inquiry for sergei atamas (...	0
3	risk	a colleague and i are researching the differin...	0
4	request book information	earlier this morning i was on the phone with a...	0

Dataset contains 3 columns, which has subject of the mail in first column, body of the email in second column and third column label indicates if the mail is spam or ham, this label column has integer datatype contains 0 and 1, 0 indicating the mail is ham and 1 indicates spam.

#	Column	Non-Null Count	Dtype
0	subject	2831 non-null	object
1	message	2893 non-null	object
2	label	2893 non-null	int64

- ## *Data Pre-processing Done*

We first removed the subject column as we considered it irrelevant because many columns were missing anyway and just by looking into the subject of the mail, we would not be able to make conclusion whether it's a spam mail.
Now we are left with 2 columns, label column was already an integer we don't have to take any action.
The only column that is left is the message column, since any of the machine learning models can only understand numbers, we had to convert set of words, numbers and characters present in the mail body to a format that can be understood by machine learning, hence we have used TfidfVectorizer to transform text to feature vectors that can be used as input to estimator.

- ## Data Inputs- Logic- Output Relationships

We have analysed the words that were present in the spam and ham mails, based on the words present and the data we already have which says if the mail is ham or spam, we are going to train the model to predict the same.

- ## State the set of assumptions (if any) related to the problem under consideration

None

- ## Hardware and Software Requirements and Tools Used

Jupyter notebook is the IDE, and the libraries that are used are joblib, sklearn, scipy,pandas, seaborn, matplotlib, numpy, TfidfVectorizer, NLTK, wordcloud

Model/s Development and Evaluation

- *Identification of possible problem-solving approaches (methods)*

  As the target column was bivariant data and the algorithm that we choose depends on this target variable, I have done classification analysis for this project.

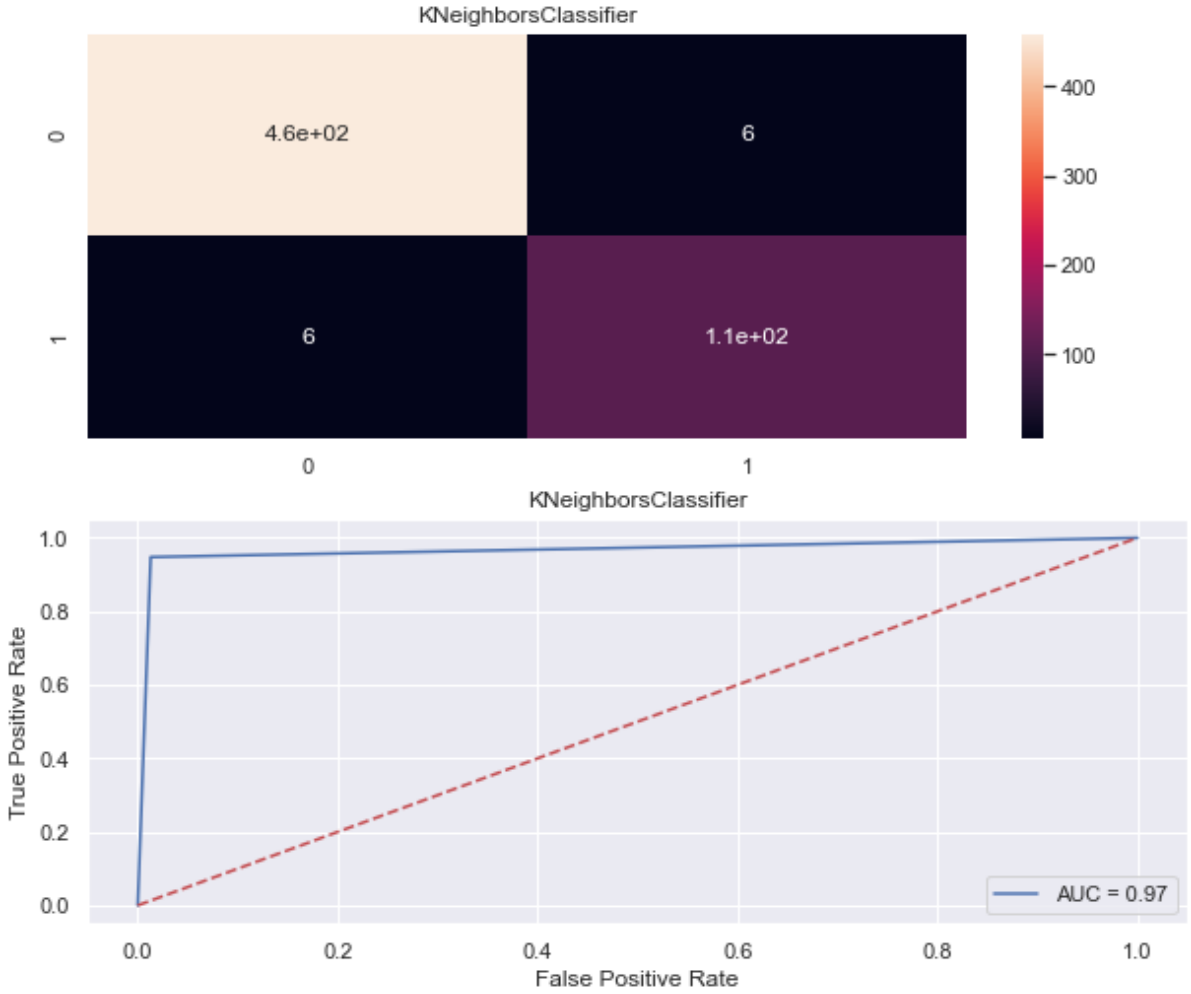
- *Testing of Identified Appproaches (Algorithms)*

  The algorithms tested on this data sets are as follows, KNeighborsClassifier,LogisticRegression, DecisionTreeClassifier, GaussianNB, RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier, ExtraTreesClassifier and SVC

- *Run and Evaluate selected models*

  All the algorithms were executed using the same code which was a loop executing models one after the other

```
*------------------------------* KNeighborsClassifier *----------------------
---------*


KNeighborsClassifier()


Accuracy_score =  0.9792746113989638


Cross_Val_Score =  0.9681971125164062


roc_auc_score =  0.9674475262368815


classification_report
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       464
           1       0.95      0.95      0.95       115

    accuracy                           0.98       579
   macro avg       0.97      0.97      0.97       579
weighted avg       0.98      0.98      0.98       579



[[458   6]
 [  6 109]]


AxesSubplot(0.125,0.808774;0.62x0.0712264)
```



------------------------------ SVC *------------------------------*

SVC()

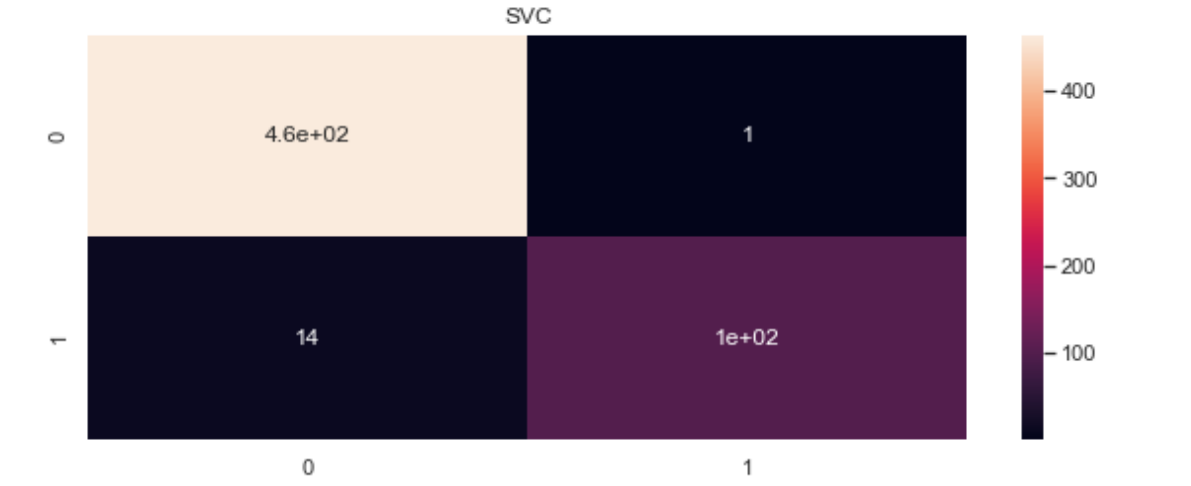
Accuracy_score = 0.9740932642487047

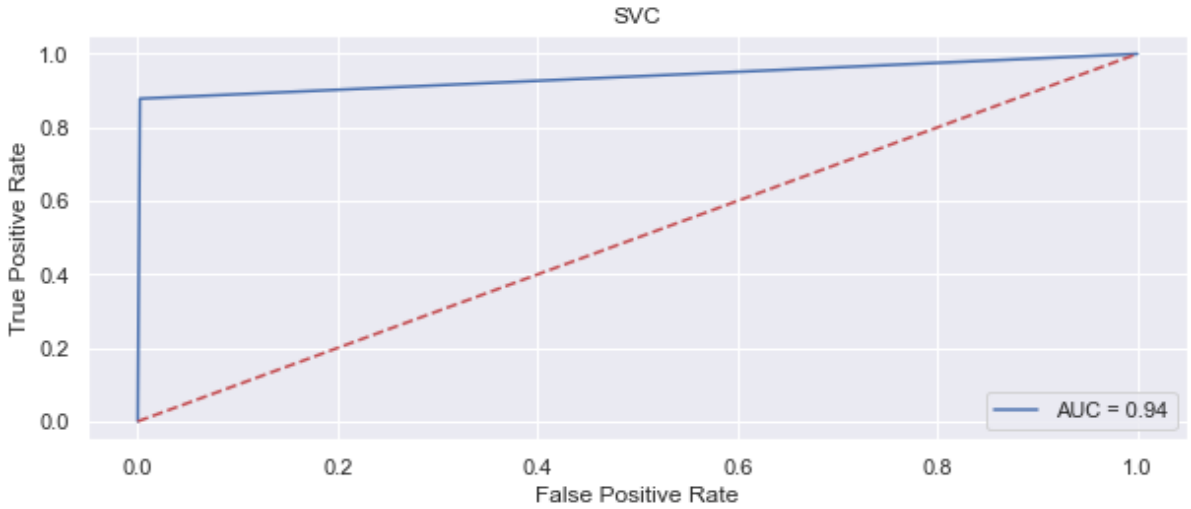
Cross_Val_Score = 0.9747679274549576

roc_auc_score = 0.9380528485757121

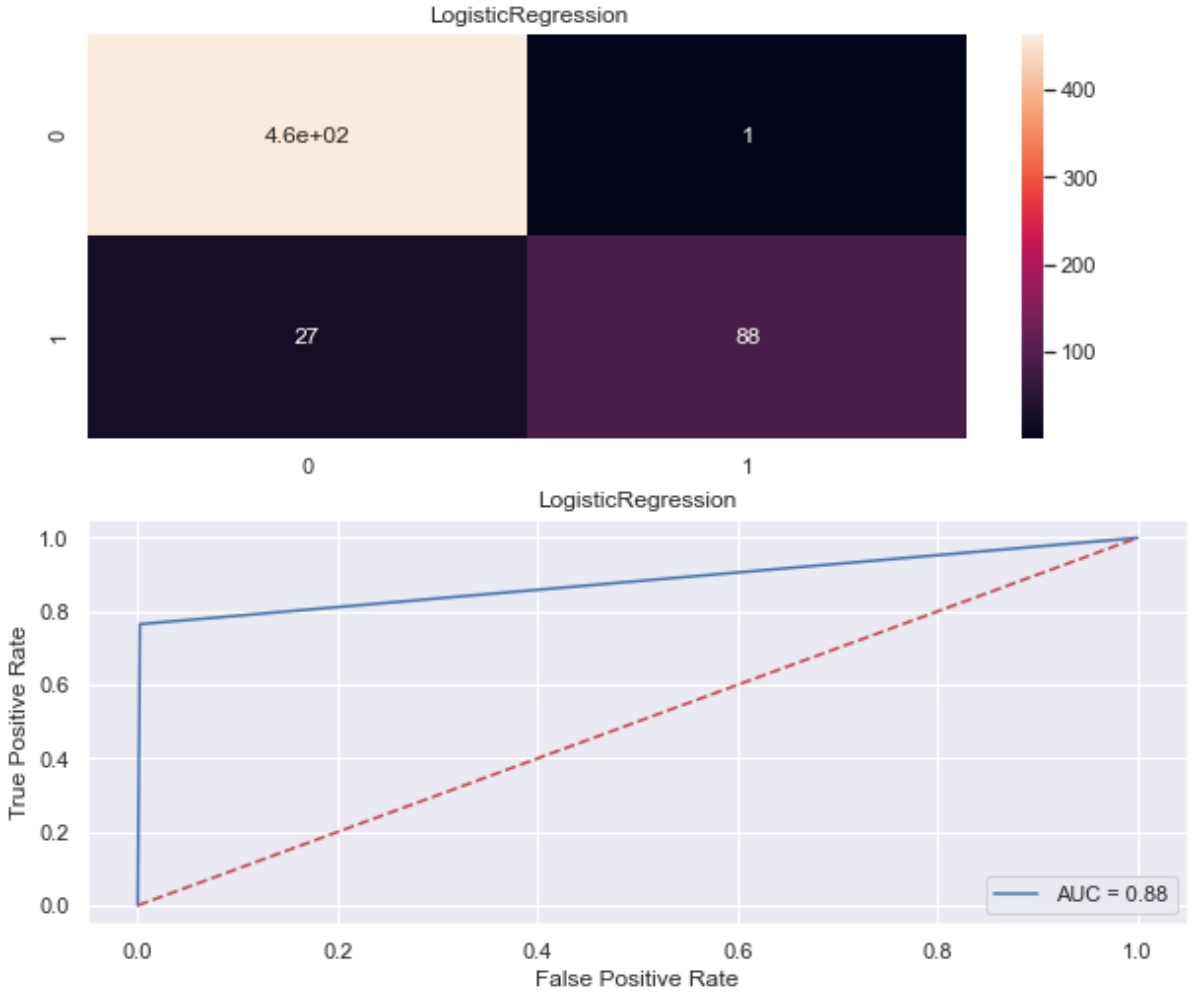
classification_report

```
                precision    recall  f1-score   support

           0       0.97      1.00      0.98       464
           1       0.99      0.88      0.93       115

    accuracy                           0.97       579
   macro avg       0.98      0.94      0.96       579
weighted avg       0.97      0.97      0.97       579


[[463   1]
 [ 14 101]]


AxesSubplot(0.125,0.808774;0.62x0.0712264)
```





```
*------------------------------* LogisticRegression *-----------------------
-------*


LogisticRegression()


Accuracy_score =  0.9516407599309153


Cross_Val_Score =  0.95333015153323


roc_auc_score =  0.8815311094452773




classification_report
               precision    recall  f1-score   support

           0       0.94      1.00      0.97       464
           1       0.99      0.77      0.86       115

    accuracy                           0.95       579
   macro avg       0.97      0.88      0.92       579
weighted avg       0.95      0.95      0.95       579



[[463   1]
 [ 27  88]]


AxesSubplot(0.125,0.808774;0.62x0.0712264)
```



------------------------------ DecisionTreeClassifier *--------------------
-----------*

DecisionTreeClassifier(random_state=56)

Accuracy_score = 0.9637305699481865

Cross_Val_Score = 0.9502195442071353

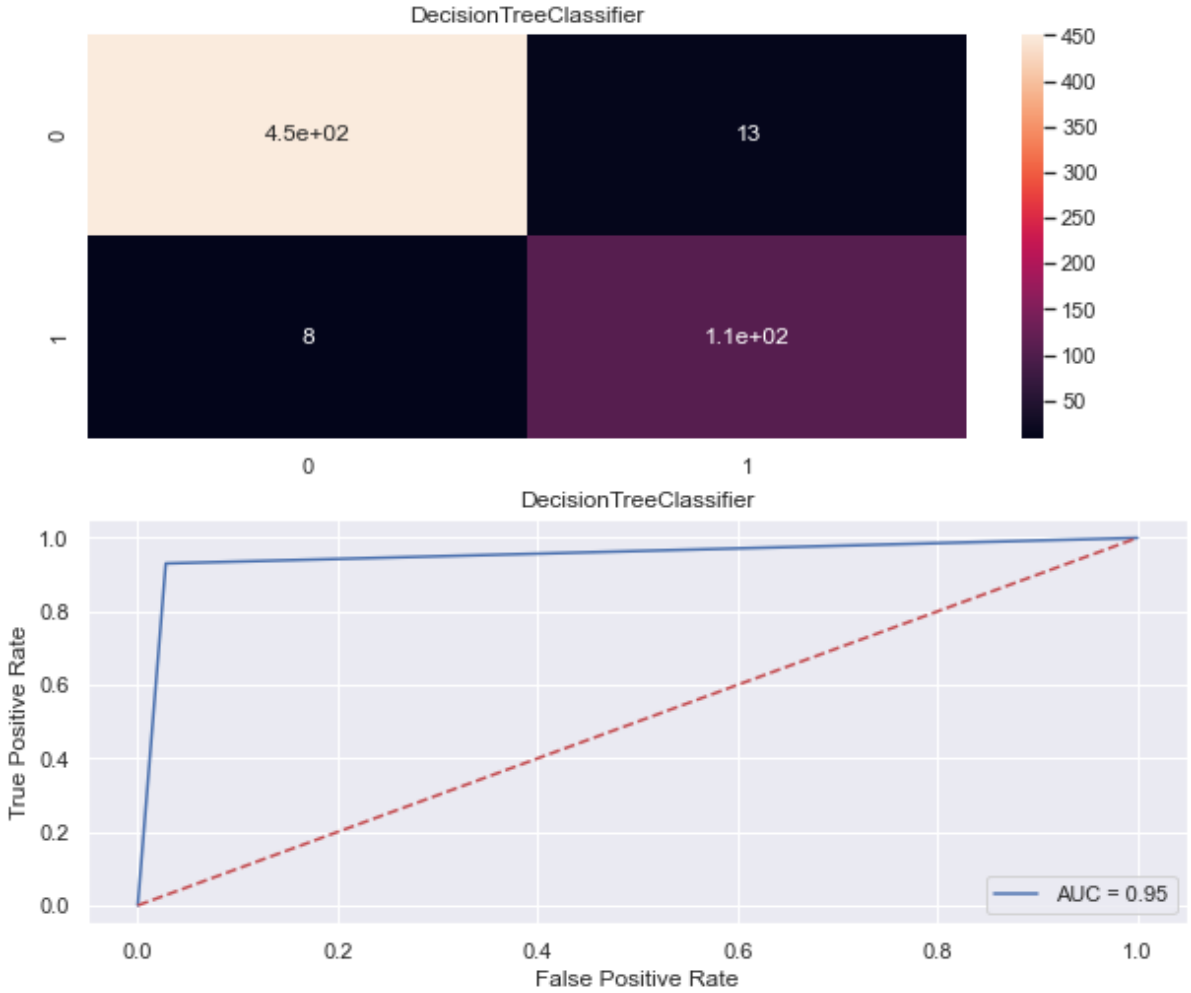
roc_auc_score = 0.9512087706146926

classification_report
              precision   recall f1-score   support

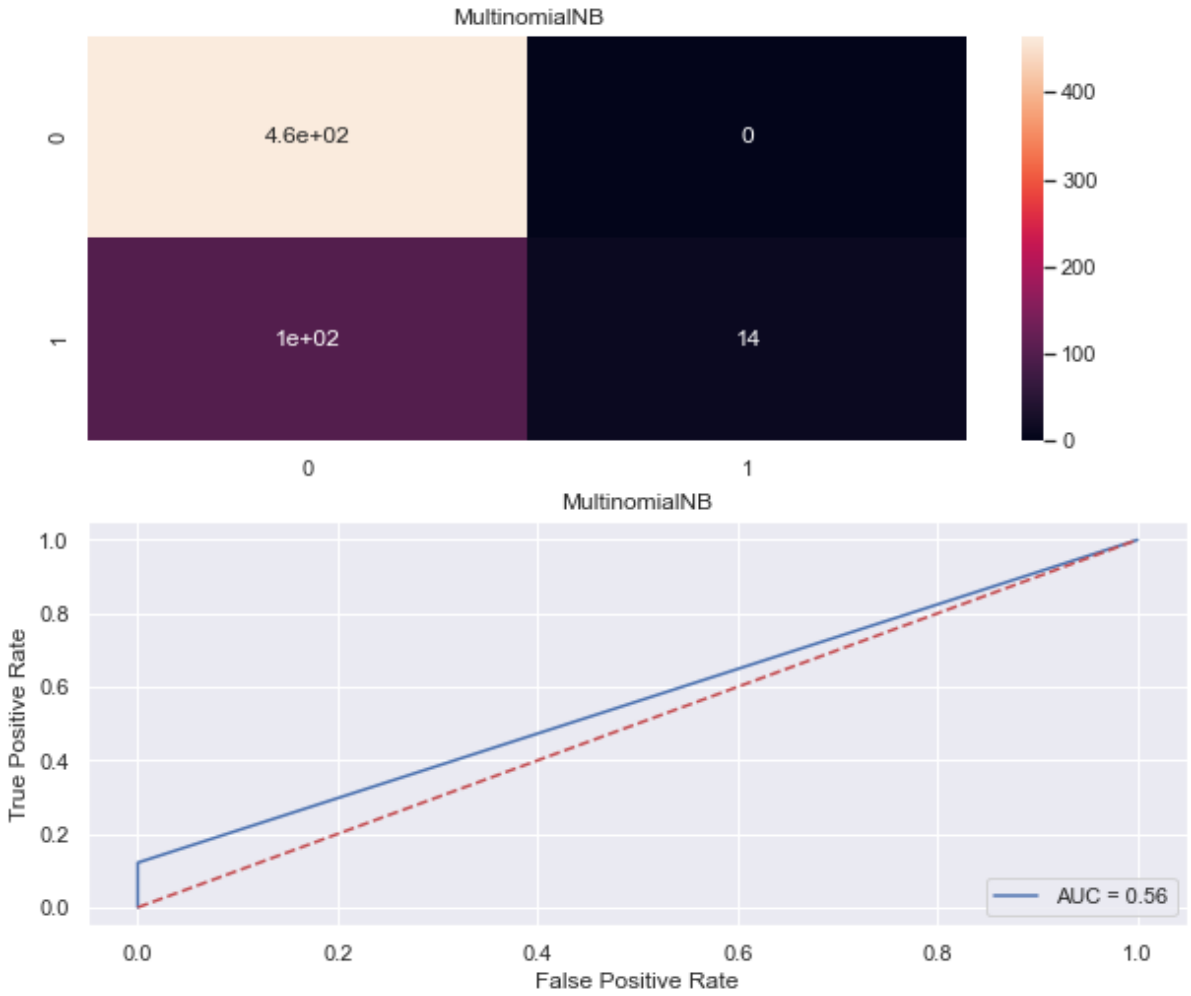
```
              0       0.98      0.97      0.98       464
              1       0.89      0.93      0.91       115

    accuracy                            0.96       579
   macro avg       0.94      0.95      0.94       579
weighted avg       0.96      0.96      0.96       579
```

```
[[451  13]
 [  8 107]]
```

```
AxesSubplot(0.125,0.808774;0.62x0.0712264)
```



```
*------------------------------* MultinomialNB *------------------------------
--*


MultinomialNB()


Accuracy_score =  0.8255613126079447


Cross_Val_Score =  0.8596575587638705


roc_auc_score =  0.5608695652173913




classification_report
               precision    recall  f1-score   support

           0       0.82      1.00      0.90       464
           1       1.00      0.12      0.22       115

    accuracy                           0.83       579
   macro avg       0.91      0.56      0.56       579
weighted avg       0.86      0.83      0.77       579



[[464   0]
 [101  14]]


AxesSubplot(0.125,0.808774;0.62x0.0712264)
```



------------------------------ RandomForestClassifier *------------------
-----------*

RandomForestClassifier(n_estimators=200, random_state=56)

Accuracy_score = 0.9758203799654577

Cross_Val_Score = 0.9723517480014318

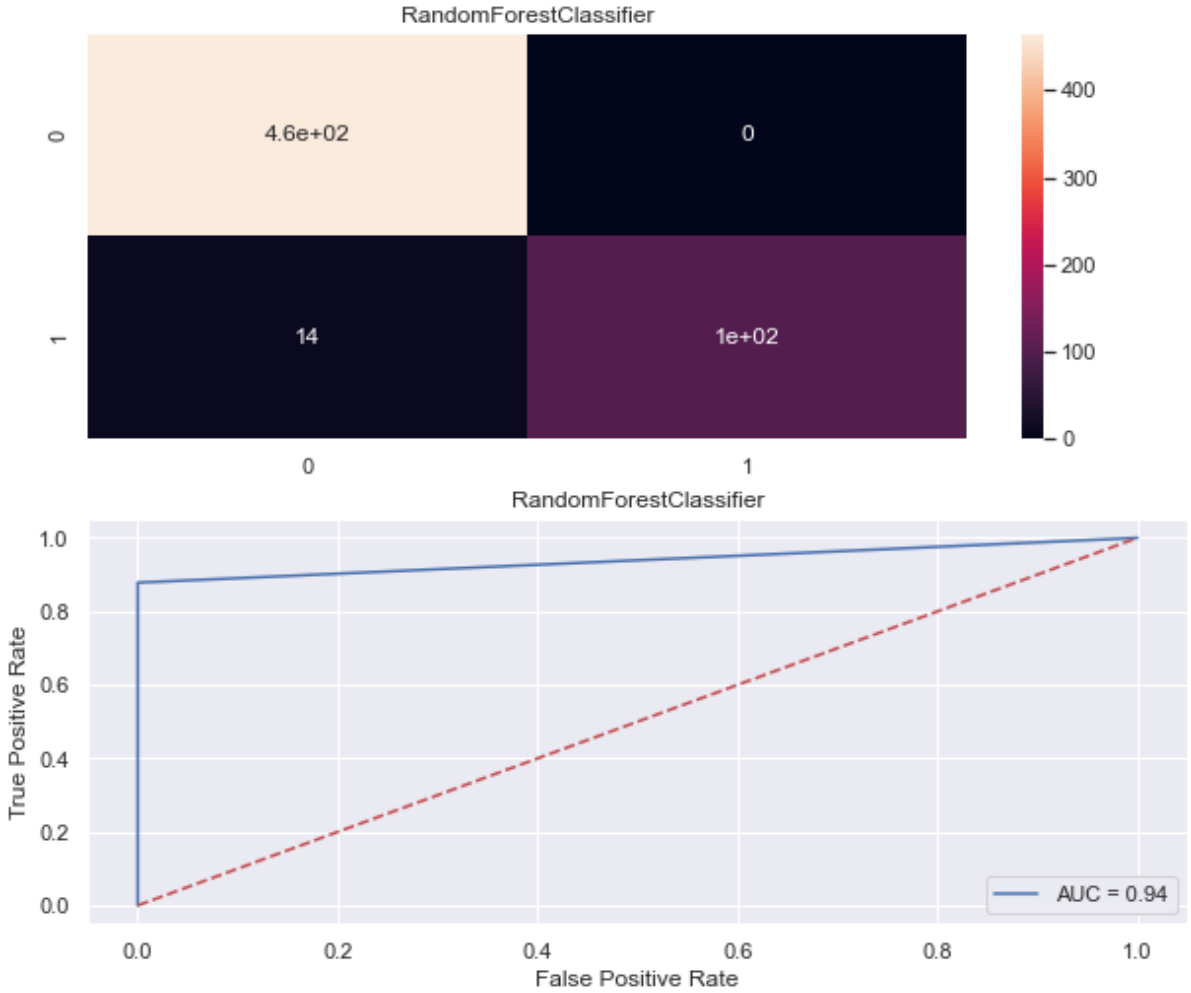
roc_auc_score = 0.9391304347826087

classification_report
              precision   recall f1-score   support

```
              0       0.97      1.00      0.99       464
              1       1.00      0.88      0.94       115

    accuracy                            0.98       579
   macro avg        0.99      0.94      0.96       579
weighted avg        0.98      0.98      0.98       579
```

```
[[464   0]
 [ 14 101]]
```

```
AxesSubplot(0.125,0.808774;0.62x0.0712264)
```



------------------------------ GradientBoostingClassifier *---------------
---------------*

GradientBoostingClassifier(n_estimators=200, random_state=56)

Accuracy_score = 0.9792746113989638

Cross_Val_Score = 0.9733874239350913

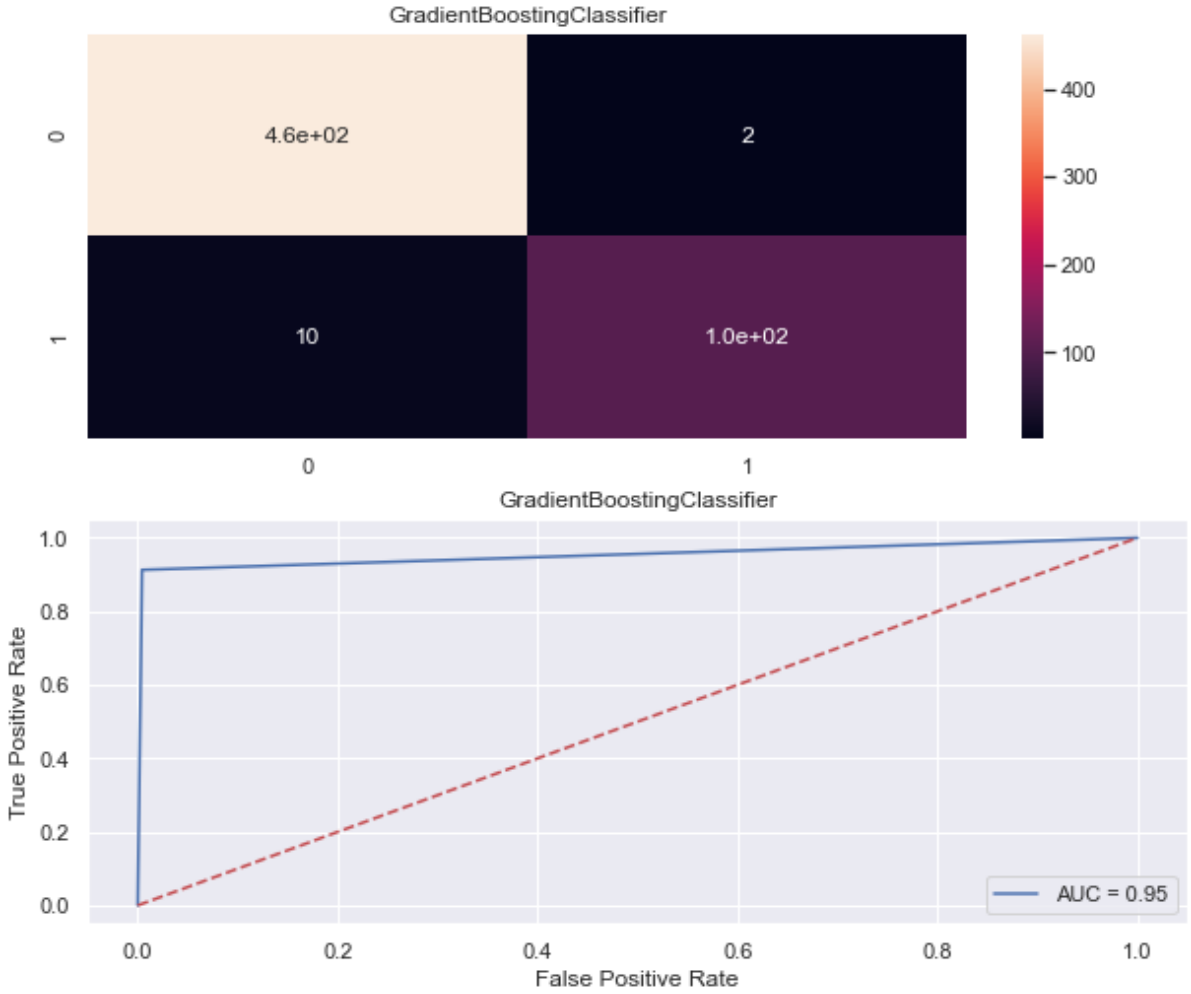
roc_auc_score = 0.9543665667166417

classification_report

	precision	recall	f1-score	support
0	0.98	1.00	0.99	464
1	0.98	0.91	0.95	115
accuracy			0.98	579
macro avg	0.98	0.95	0.97	579
weighted avg	0.98	0.98	0.98	579

```
[[462   2]
 [ 10 105]]
```

AxesSubplot(0.125,0.808774;0.62x0.0712264)



------------------------------ ExtraTreesClassifier *----------------------
---------*

ExtraTreesClassifier(random_state=56)

Accuracy_score = 0.9706390328151986

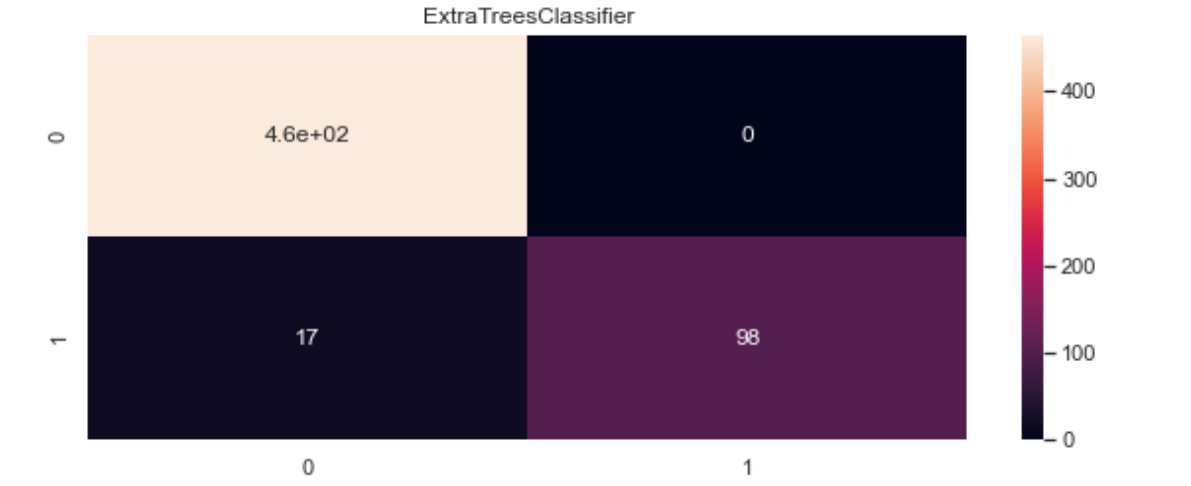
Cross_Val_Score = 0.9695788092113113

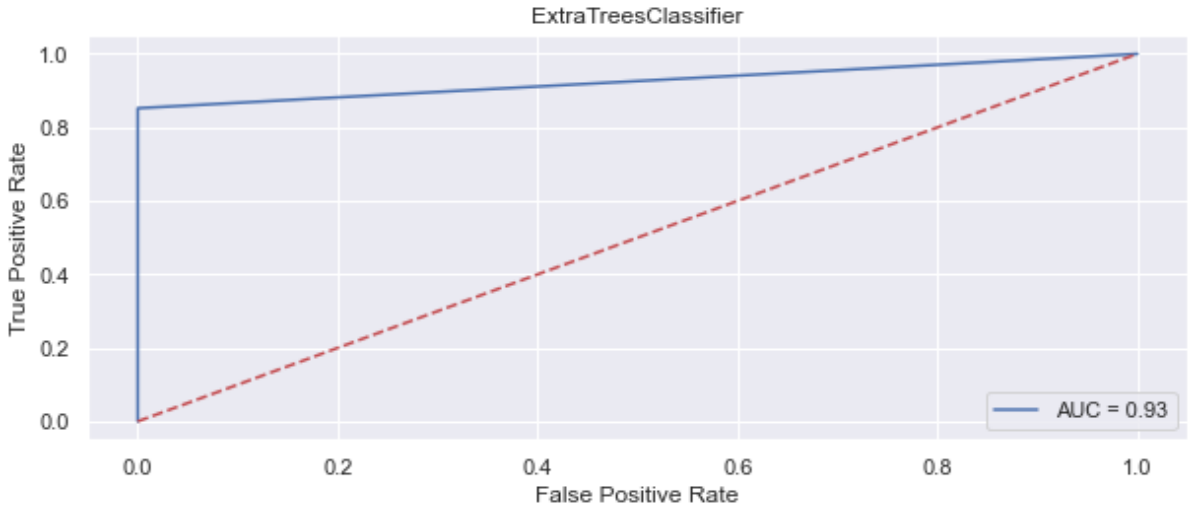
roc_auc_score = 0.9260869565217391

classification_report

```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98       464
           1       1.00      0.85      0.92       115

    accuracy                           0.97       579
   macro avg       0.98      0.93      0.95       579
weighted avg       0.97      0.97      0.97       579


[[464   0]
 [ 17  98]]
```

```
AxesSubplot(0.125,0.808774;0.62x0.0712264)
```

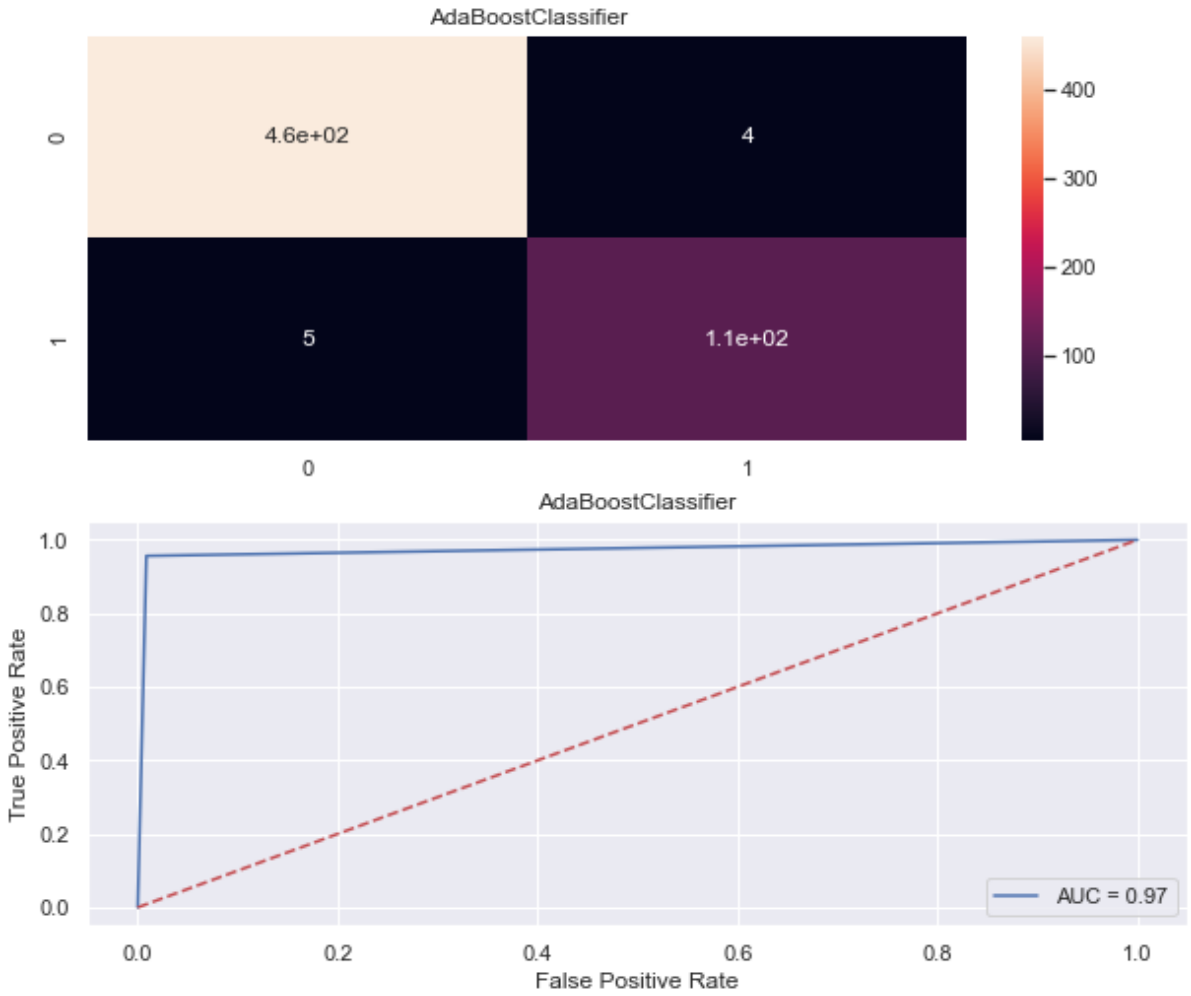




```
*------------------------------* AdaBoostClassifier *-----------------------
-------*


AdaBoostClassifier(random_state=56)


Accuracy_score =  0.9844559585492227


Cross_Val_Score =  0.9799522729984489


roc_auc_score =  0.9739505247376312
```

```
classification_report
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       464
           1       0.96      0.96      0.96       115

    accuracy                           0.98       579
   macro avg       0.98      0.97      0.98       579
weighted avg       0.98      0.98      0.98       579
```

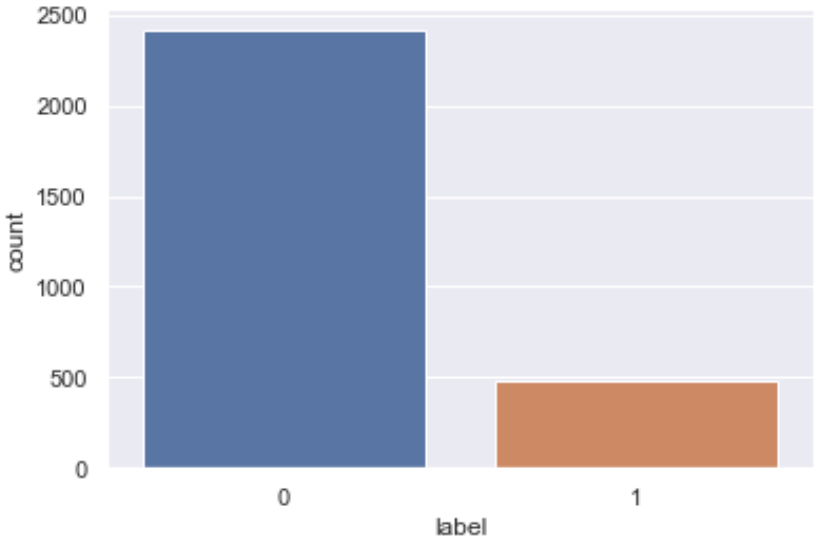
```
[[460   4]
 [  5 110]]


AxesSubplot(0.125,0.808774;0.62x0.0712264)
```



- *Key Metrics for success in solving problem under consideration*

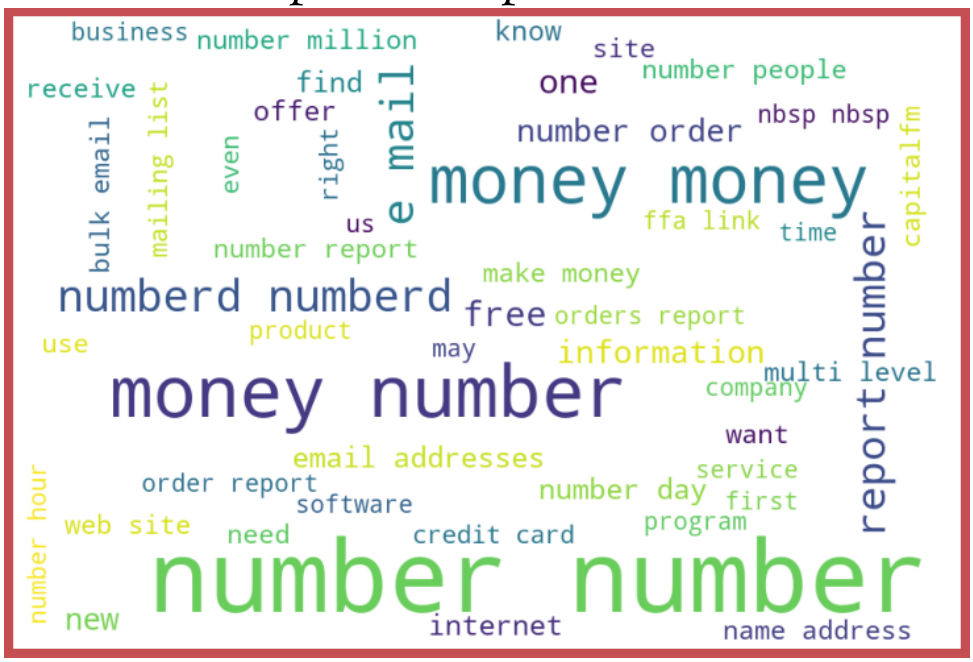
Key metric to finalise the model was confusion matrix and auc roc curve.

- *Visualizations*

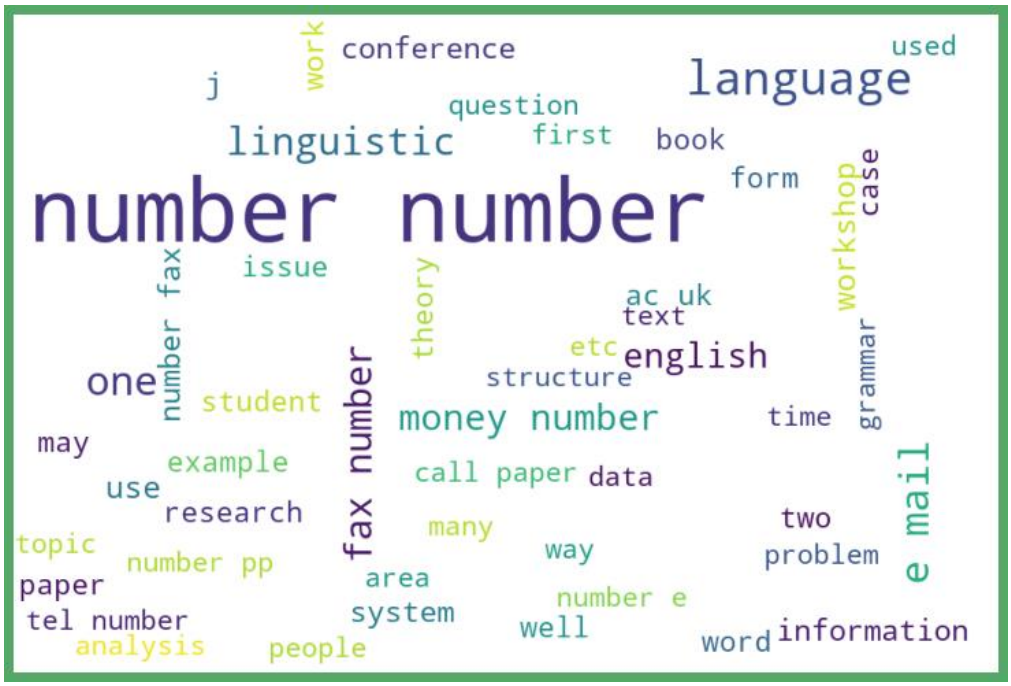


We can see that out of 2893 mails 481 mails are spam

Loud words present in spam mails.



Loud words present in ham mails.



- *Interpretation of the Results*

	Classification Model	Accuracy Score	Cross_val_score	Roc_auc_curve
0	KNeighborsClassifier	97.927461	96.819711	96.744753
1	SVC	97.409326	97.476793	93.805285
2	LogisticRegression	95.164076	95.333015	88.153111
3	DecisionTreeClassifier	96.373057	95.021954	95.120877
4	MultinomialNB	82.556131	85.965756	56.086957
5	RandomForestClassifier	97.582038	97.235175	93.913043
6	GradientBoostingClassifier	97.927461	97.338742	95.436657
7	ExtraTreesClassifier	97.063903	96.957881	92.608696
8	AdaBoostClassifier	98.445596	97.995227	97.395052