PROJECT NAME:-

EMAIL SPAM DETECTION

SUBMITTED BY:

SHIVAM SHARMA

INTRODUCTION

Problem Statement :

Spam Detector is used to detect unwanted, malicious and virus infected texts and helps to separate them from the non spam texts. It uses a binary type of classification containing the labels such as '**ham'** (nonspam) and **spam**. Application of this can be seen in Google Mail(GMAIL) where it segregates the spam emails in order to prevent them from getting into the user's inbox.

The files contain one message per line. Each line is composed by two columns: v1 contains the label (ham or spam) and v2 contains the raw text.

A collection of 5573 rows SMS spam messages was manually extracted from the Grumbletext Web site. This is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the very spam message received. The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involved carefully scanning hundreds of web pages.

Review of Literature

The topic is about to build a Model which can predict the LABELS(‘ham’ or ‘spam’) of any email by analysing the message inside it.

Motivation for the Problem Undertaken

As the frauds increasing day by day due to increment in the technology rapidly, So that is very necessary to avoid the frauds that’s y it is very necessary to avoid spams from the messages or from emails so that we can prevent someone from the frauds.

Data Sources and their format

- Grumbletext Web site
- NUS SMS Corpus (NSC)
- The dataset import from the excel sheet

```
data = pd.read_excel("C:\\Users\\LENOVO\\Desktop\\spam.xlsx")
data
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

Data Preprocessing and EDA

The steps followed for the cleaning and EDA of the data:-

1) First we analyse the data by simply data.info() and data.describe() methods so that we can check about the datatypes and checked themean , median and deviation.
2) Then we drop some unwanted columns by drop method.

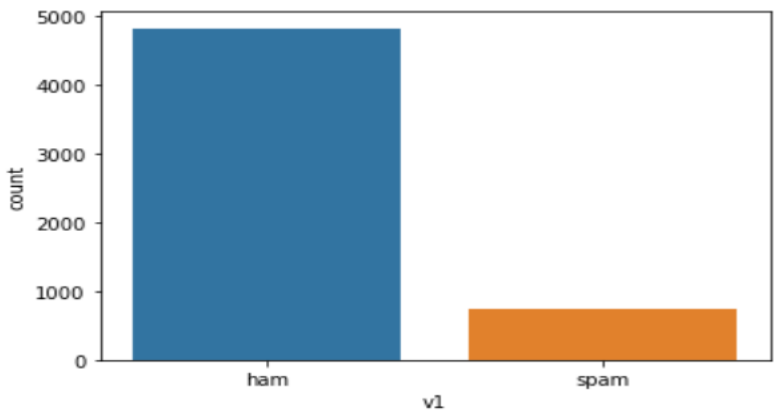
```
data = data.drop(['Unnamed: 2','Unnamed: 3','Unnamed: 4'],axis=1)
data.head()
```

	v1	v2
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

3) Then we checked the counts of spam and hams in the datset by the help of the countplot.

```
: sns.countplot(x='v1', data= data)
  data['v1'].value_counts()
```

```
: ham     4825
  spam     747
  Name: v1, dtype: int64
```



4) Then after we made a new Column of Length which contains the totallength of the message.

```
# Lets make the count of the length of the messages

data['length'] = data['v2'].str.len()
data.head(7)
```

5) After that we rename the name of the columns as we give our output to Label and to feature as message.

```
# Lets change the values of v1 so that we can assign ham to 0 and spam to 1

data.rename(columns={'v1':'label','v2':'message'},inplace = True)
data.head()
```

6) And also assign the value 0(ham) , 1(spam) from the column of label .

```
# label coding 0 And 1

data['label'].replace({'ham':0,'spam':1},inplace=True)
```

7) Then after we Convert all messages in lower case, we replaced email addresses with 'email', URLs with 'webaddress', money symbols with 'moneysymb' (£ can by typed with ALT key + 156), 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumber', and numbers with 'numbr'…

```
]: # Replace email addresses with 'email'

data['message'] = data['message'].str.replace(r'^.+@[^\.].*\.[a-z]{2,}$','emailaddress')
```

```
]: # Replace URLs with 'webaddress'

data['message'] = data['message'].str.replace(r'^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\S*)?$','webaddress')
```

```
]: # Replace money symbols with 'moneysymb'

data['message'] = data['message'].str.replace(r'£|\$', 'dollers')
```

```
]: # Replace 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumber'

data['message'] = data['message'].str.replace(r'^\(?[\d]{3}\)?[\s-]?[\d]{3}[\s-]?[\d]{4}$','phonenumber')
```

```
]: # Replace numbers with 'numbr'
data['message'] = data['message'].str.replace(r'\d+(\.\d+)?', 'numbr')
```

8) Then after we apply lambda function to remove the Stopwords and punctuations from the comment and then we used WordNetLemmatizer so that we can make Lemmas and words can bereduced into their meaningful roots..

```
: data['message'] = data['message'].str.replace(r'[^\w\d\s]', ' ')

: # REPLACE WHITESPACES BETWEEN TERMS WITH A SINGLE SPACE

  data['message'] = data['message'].str.replace(r'\s+', ' ')

: # Remove leading and trailing whitespaces

  data['message'] = data['message'].str.replace(r'^\s+|\s+?$','')

: data.head()
```

```
# Remove Stopwords

data['message'] = data['message'].apply(lambda x: ' '.join(term for term in x.split() if term not in stop_words))
```

```
lem=WordNetLemmatizer()

data['message'] = data['message'].apply(lambda x: ' '.join(lem.lemmatize(t) for t in x.split()))
```

9) Then after lemmatizing the comment we made a new column(name as Clear_lenght) so that we can check the original nad clear length separately by doing these cleanings(like stopwords, punctuations andmany more as written above)..

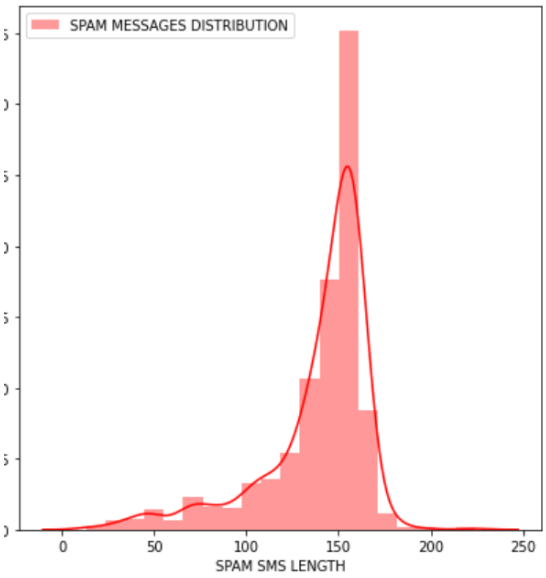
```
data['clean_length'] = data['message'].str.len()

data.head(5)
```

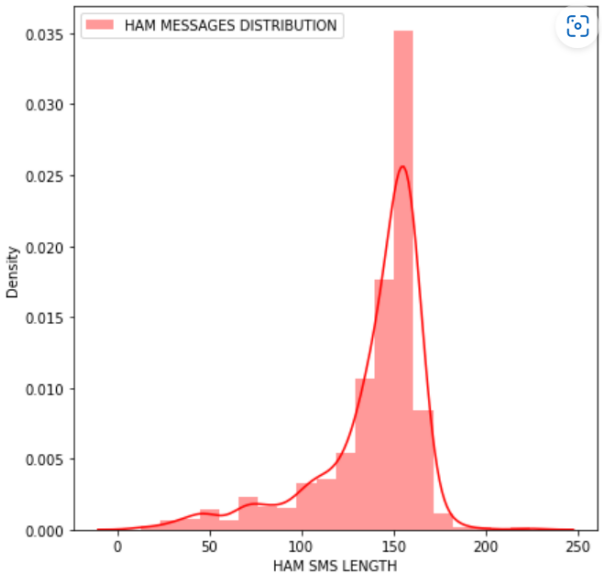
	label	message	length	clean_length
0	0	go jurong point crazy available bugis n great ...	111.0	82
1	0	ok lar joking wif oni	29.0	21
2	1	free entry numbr wkly comp win fa cup final tk...	155.0	139
3	0	dun say early hor c already say	49.0	31
4	0	nah think go usf life around though	61.0	35

10) By the help of matplot and seaborn we plot some graph before and after cleaning the distribution of the messages .

BEFORE CLEANING:-

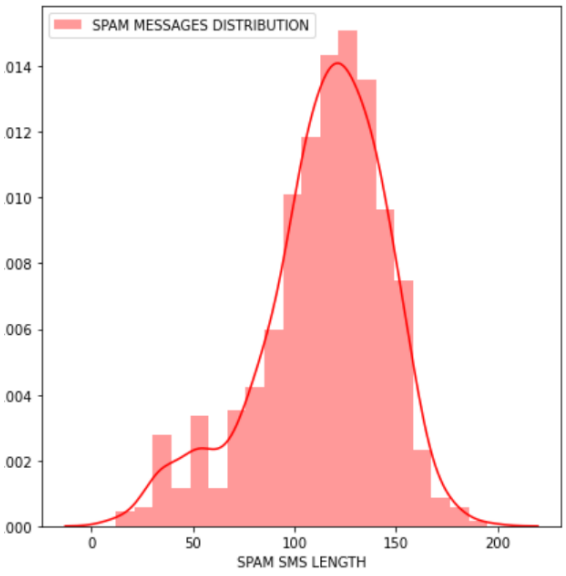
```
ion matplotlib.pyplot.show(close=None, block=None)>
```

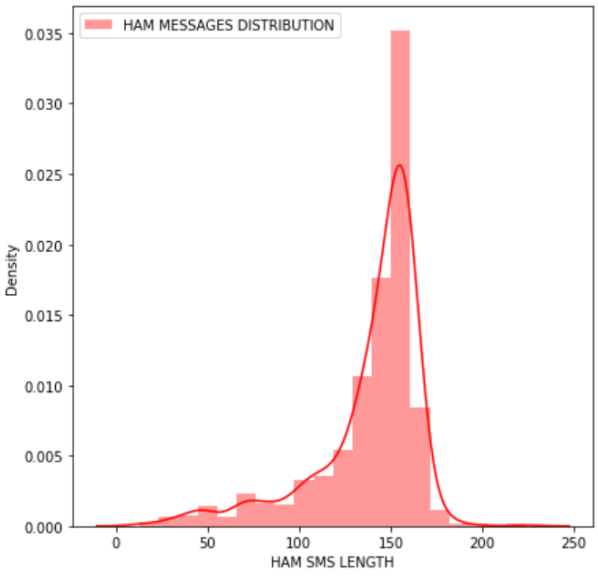




AFTER CLEANING:

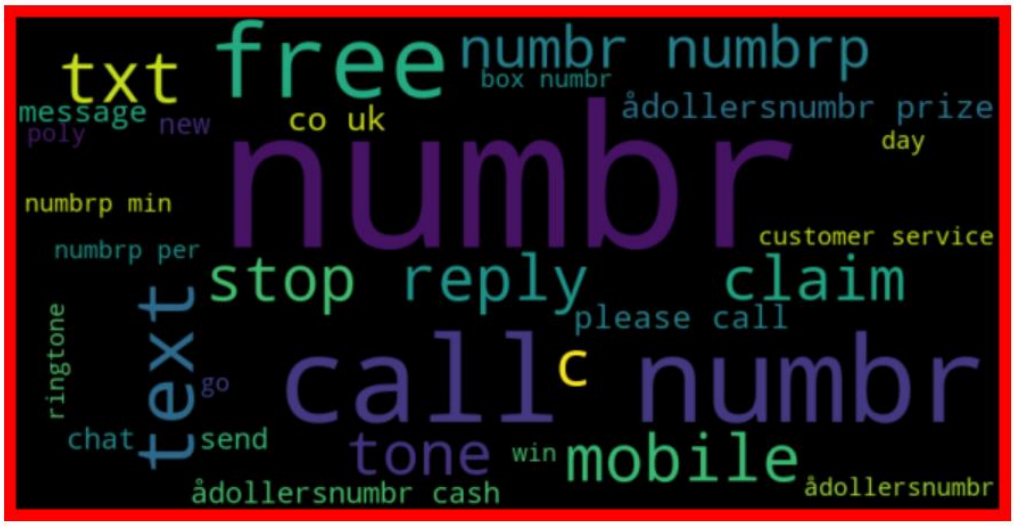
```
ction matplotlib.pyplot.show(close=None, block=None)>
```



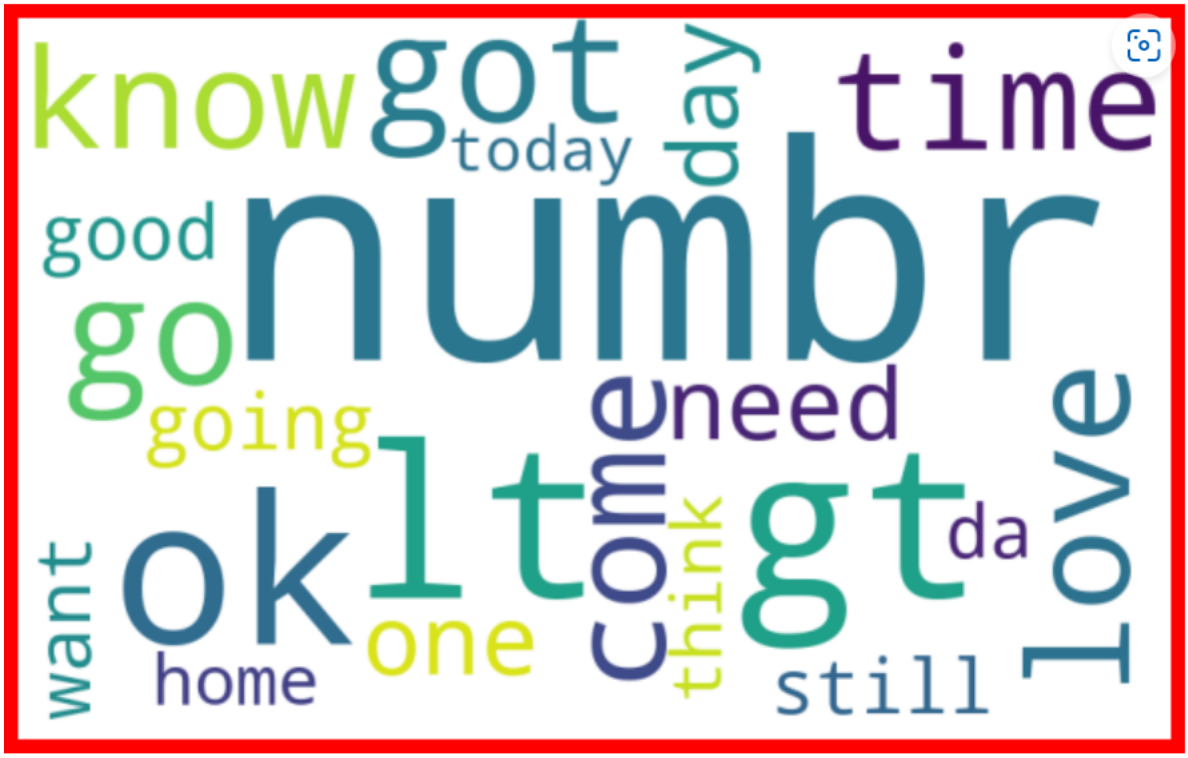


11) Then we used wordcloud for some words that are Loud so that our modle predict on the basis of these words.

FOR SPAM:



FOR HAMS:



12) Then after this we are ready to convert the messages into the Vector form (as machine can not understand object/strings) with thehelp of TF-IDF Vectorizer…

13) Now we are ready for the model building..

Algorithm used in this project:-

For building machine learning models there are several models present inside the Sklearn module.

Sklearn provides two types of models i.e. regression and classification. Our dataset's target variable is to predict whether fraud is reported or not. So for this kind of problem we use classification models.

But before the model fitting we have to seprate the predictor and target variable, then we pass this variable to the train_test_split method to create the training set and testing setfor the model training and prediction.

We can build as many models as we want to compare the accuracy given by these models and to select the best model among them.

I have selected 5 models:

1. MultinomialNB
2. Logistic Regression
3. Decision Tree Classifier
4. Random forest Classifier
5. Gradient boosting Classifier

BEST Algo. From these And why?

Random Forest classifier is the best algo. From all of these algorithms which is used in this data to predict because the difference between the cross val score and the accuracy score is minimum for the Random Forest algo. and it also gives BETTER ACCURACY(approx. 99.7%)that's why we USED THISAlgo.

- Then we hypertune this algo with the help of Grid searchCV .
- **Save the model for later predictions**

 # Now my model is ready to predict

CONCLUSIONS

```
Training accuracy is 0.9997435897435898
Test accuracy is 0.9838516746411483
[[1464    0]
 [  27  181]]
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1464
           1       1.00      0.87      0.93       208

    accuracy                           0.98      1672
   macro avg       0.99      0.94      0.96      1672
weighted avg       0.98      0.98      0.98      1672
```

So these are the classification report results of the Random algo.

Hence my model is ready to predict….