EMAIL SPAM CLASSIFIER PROJECT

Submitted by:

MIENGANDHA SINHA

ACKNOWLEDGEMENT

I would like to express my gratitude to the Company to give this project to me. In making of this project I hereby used to take help from the references which is given by the company as sample documentation and details related to project and professionals and SME guided me a lot in the project and the other previous projects helped me and guided me in completion of the project.

INTRODUCTION

- Email spam classifier

This project is all about to classify the email whether it is spam or not. So, in the project the SMS collection is a set of SMS tagged messages that have been collected for spam research. It contains messages in English of 5,574 messages, tagged according being ham which means legitimate or spam. The total corpus of 5728 documents. The target feature consists of two classes ham and spam, the column name is spam. The classes are labelled for each document in the dataset and represent our target feature with a binary string-type alphabet of ham and spam and these are further mapped to integer 0(ham) and 1(spam).

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

```
df['label'] = encoder.fit_transform(df['label'])
```

```
df.head()
```

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

- ## Conceptual Background of the Domain Problem

The project aims to classification email into two categories and it is titled by Email Spam classification
Is implemented by using some methodology like Data preparation, Modelling and Evaluation steps. It is implemented using Python class object based style. The spam detection is done using machine learning algorithms classifier like Naïve Bayes, ANN(artificial neural networks), and SVM(support vector machines).

- ## Review of Project

A subset of 3,375 SMS randomly chosen ham messages of the NUS SMS Corpus (NSC), which is a dataset of about 10,000 legitimate messages collected for research at the Department of computer science at National University of Singapore. These messages were collected from volunteers who were made aware that their contributions were going to be made publicly available. In order to deal with spam emails, it need to build a robust real-time email spam classifier that can efficiently and correctly flag the incoming mail spam, if it is a spam message or looks like a spam message. The latter will further help to build an Anti-Spam Filter. There is a great scope in building email spam classifiers, as the private companies run their own email servers and them to be more secure because of the confidential data, in such cases email spam classifiers solutions can be provided to such companies.

- ## Motivation for the Problem Undertaken

Here, the datasets have the total of 5169 entries with 2rows, no null values, EDA has to be performed to see whether it gain or loss in the variable and its compare to the price among every aspects, to build machine learning models, to determine the optimal values of Hyper parameters and the selection of the best model, by predicting of the value can help to the clients and for the further change in the market from the new data.

ANALYTICAL PROBLEM FRAMING

- Mathematical/Analytical Modelling of the Problem

  In this project, mathematical/analytical modelling are used. Checking the null values found that having no null values in the datasets, the type of data frame is in pandas, data frame info tells that object(5 variables), by using the data visualization they are:

  - the data frame shape and its info:

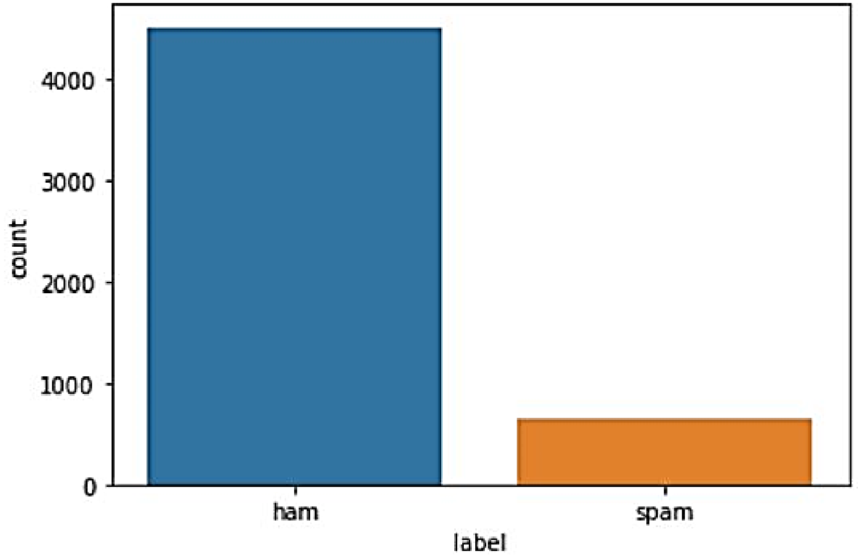
```
1  df.shape
```

```
(5572, 5)
```

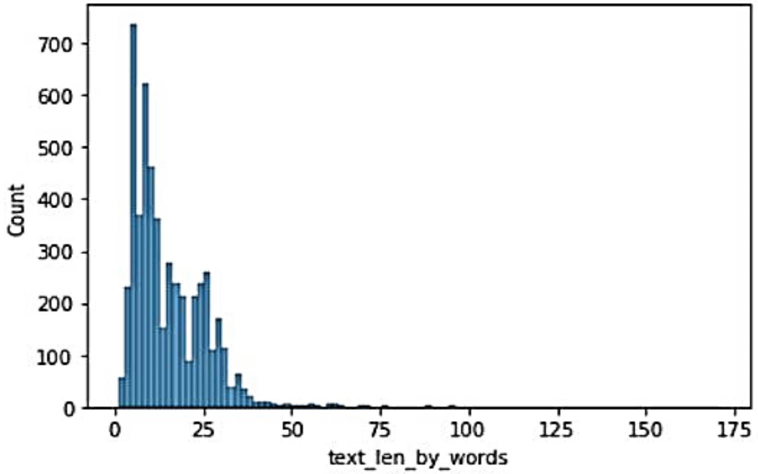
```
1  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   V1           5572 non-null   object
 1   V2           5572 non-null   object
 2   Unnamed: 2   50 non-null     object
 3   Unnamed: 3   12 non-null     object
 4   Unnamed: 4   6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

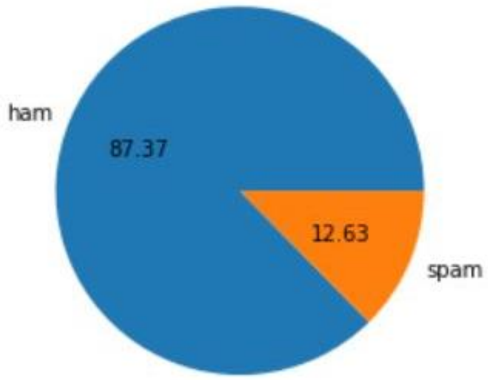
1. It is about the label of email whether it is spam or non-spam (ham):
The blue bar depicts ham emails which is on the highest point in comparison to the orange bar which depicts spam emails in the dataset.

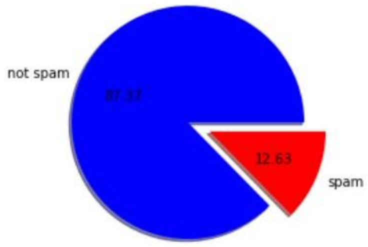


2. It is about the histogram of text length of words

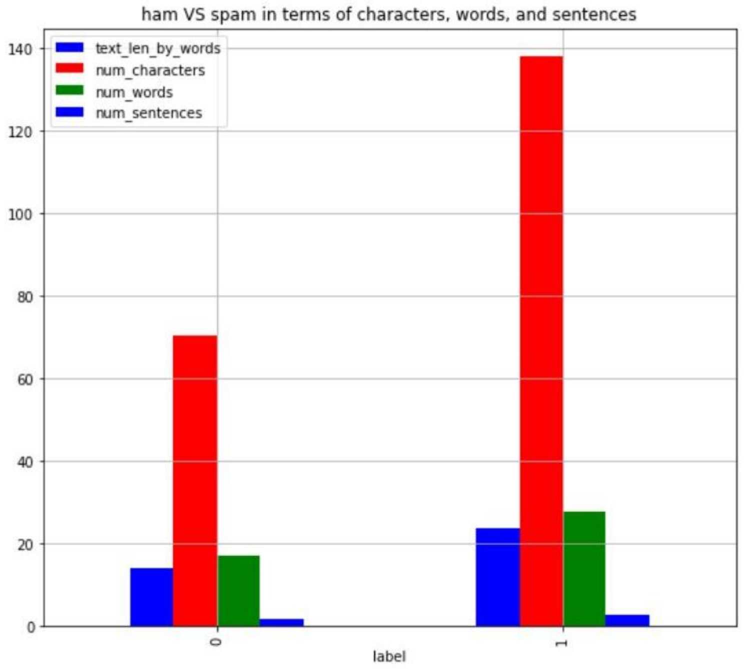


3. It is about the pie chart of label which is divided into two slices. Both slice represents the count or percentage of ham 87.37 and spam 12.63

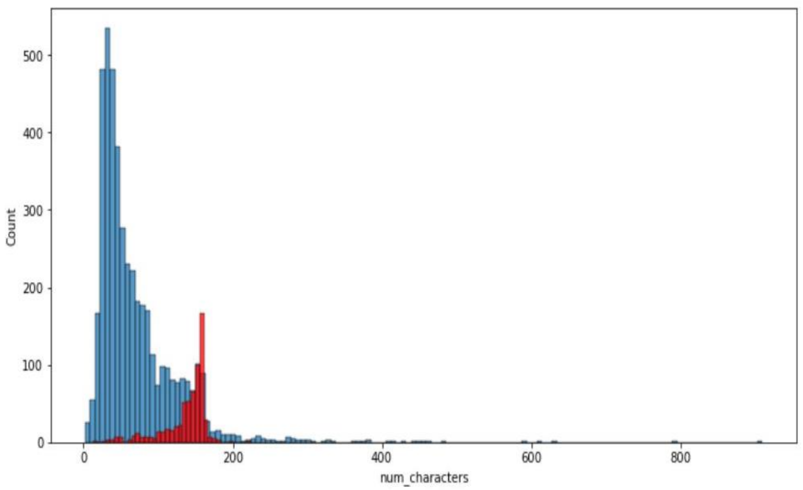




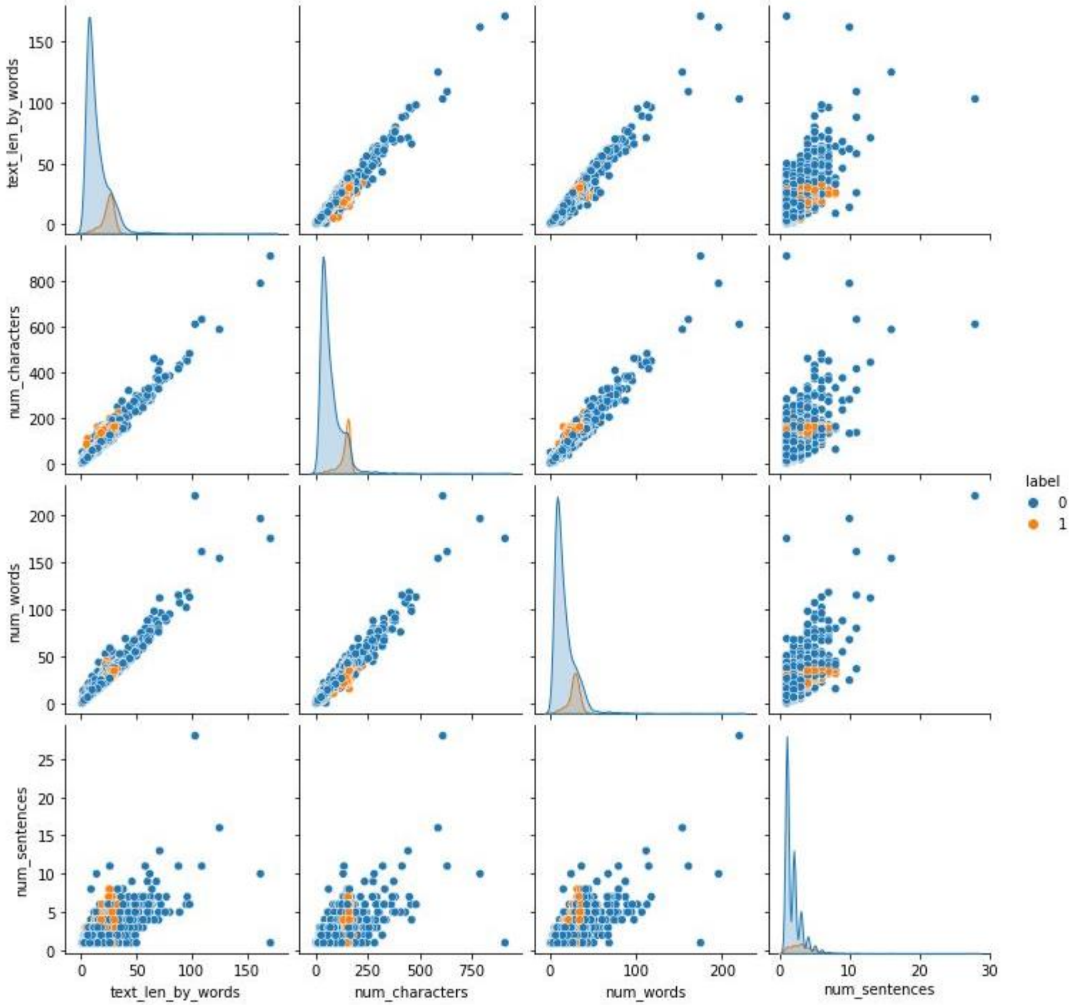
4. It is about the terms of characters, words and sentences used in the emails



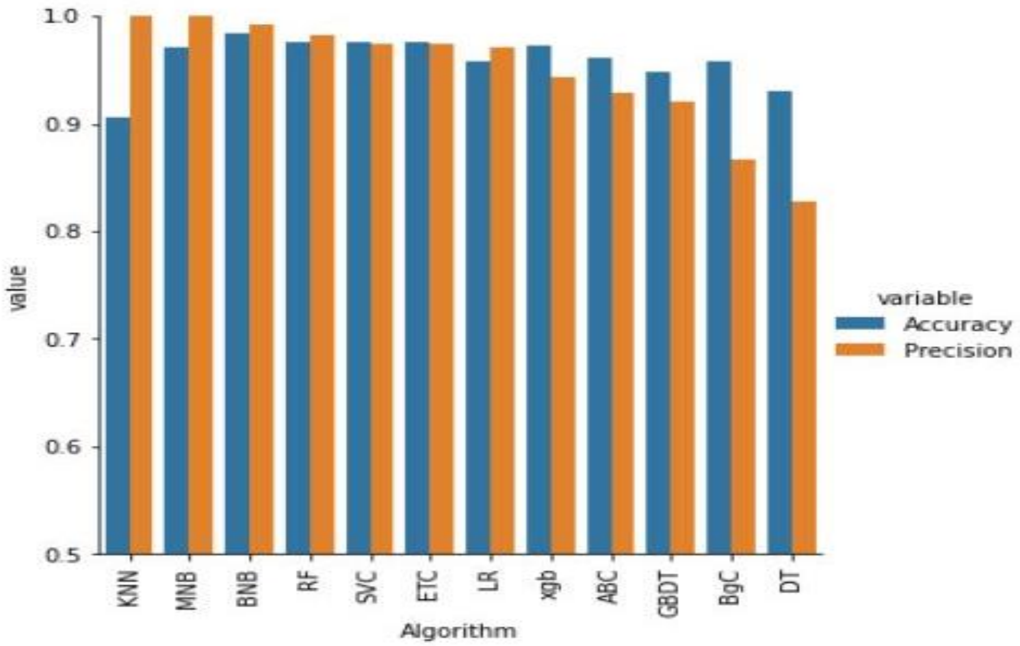
5. It is about how many number of characters used in emails



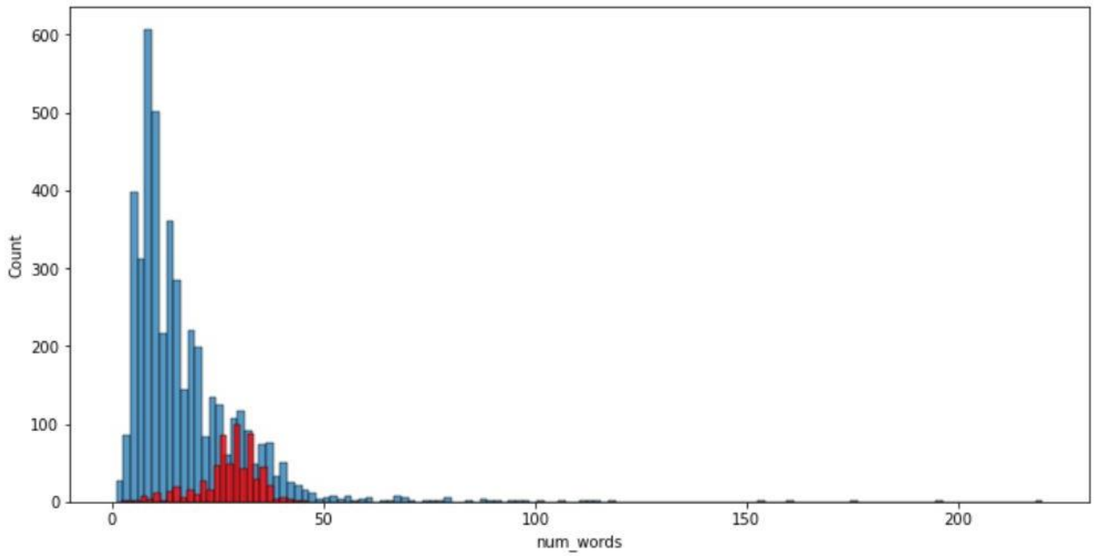
6. This is pair plot of data



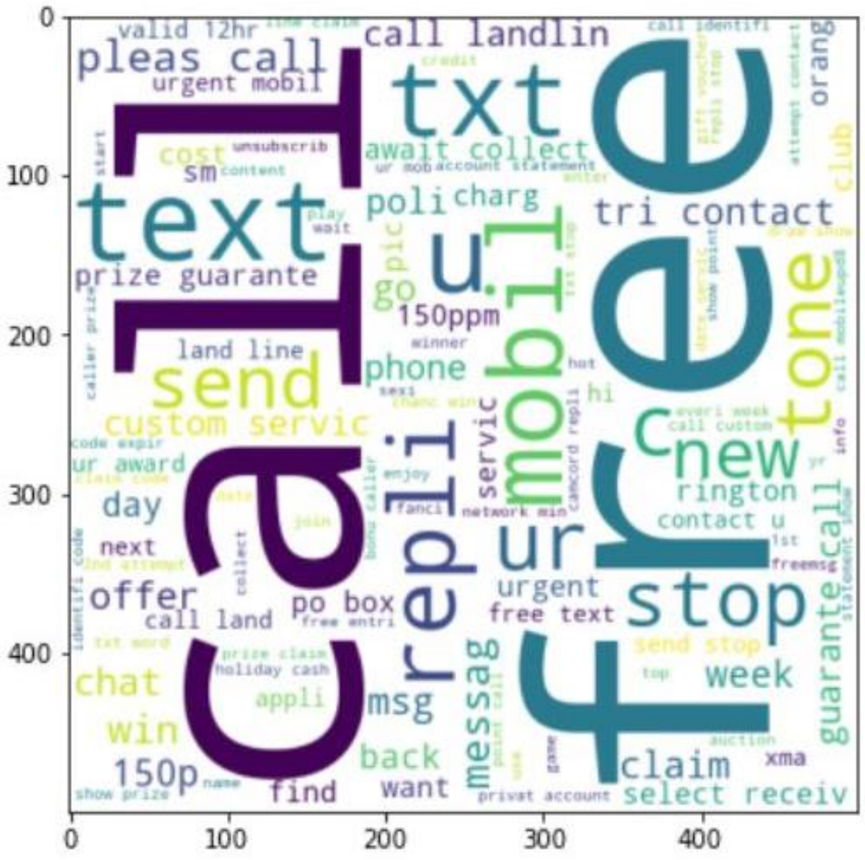
7. This is showing the algorithm of the variable carrying all the model building classifier accuracy and precision



8. It is about the histogram of distribution of number of words are in emails



9. word cloud for spam emails



10. word cloud for ham emails

- # Data Sources and their formats

The data sources and their formats are from .csv file.

```
df=pd.read_csv('C:/Users/user/Downloads/Spam Project/spam.csv',encoding='latin1')
df.sample(5)
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
1553	ham	Ok how you dear. Did you call chechi	NaN	NaN	NaN
5014	ham	I think the other two still need to get cash b...	NaN	NaN	NaN
3813	ham	Can. Dunno wat to get 4 her...	NaN	NaN	NaN
5509	ham	Lol they were mad at first but then they woke ...	NaN	NaN	NaN
3625	ham	Yeah right! I'll bring my tape measure fri!	NaN	NaN	NaN

- # Data Preprocessing Done

The steps followed for the cleaning of the data is Label Encoder after then importing preprocessing there transform the target columns into features then lastly it has to set for the data frame.

There are some methods while doing data preprocessing:

```python
from nltk.corpus import stopwords
import string
from nltk.stem.porter import PorterStemmer
ps=PorterStemmer()
```

- lower case
- tokenization
- removing special characters
- removing stopwords and punctuation
- stemming

```python
def transform_text(text):
    text = text.lower()
    text = nltk.word_tokenize(text)

    y = []
    for i in text:
        if i.isalnum():
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        if i not in stopwords.words('english') and i not in string.punctuation:
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        y.append(ps.stem(i))


    return " ".join(y)

```

```
transform_text("I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.")
```

'gon na home soon want talk stuff anymor tonight k cri enough today'

```
df['text'][10]
```

"I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today."

```
ps.stem('loving')
```

'love'

```
df['transformed_text'] = df['text'].apply(transform_text)
```

```
df.head()
```

	label	text	text_len_by_words	num_characters	num_words	num_sentences	transformed_text
0	0	Go until jurong point, crazy.. Available only ...	20	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	6	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	28	155	37	2	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	11	49	13	1	u dun say earli hor u c alreadi say
4	0	Nah I don't think he goes to usf, he lives aro...	13	61	15	1	nah think goe usf live around though

- # Data Inputs- Logic- Output Relationships

The relationships between inputs and outputs can be studied extracting weights of the trained model. Regression is that relationships between them can be blocky or highly structured based on the training data. It requires the data scientist to train the algorithm with both labeled inputs and desired outputs.

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

Presumptions are by using regression label encoding, classifier, selection of the best models, confusion matrix that it means the relationship between the dependent and independent variables look fairly linear. Thus, our linearity assumption is satisfied.

- **Hardware and Software Requirements and Tools Used**

By importing many libraries are

1.IMPORT LIBRARIES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, plot_confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier

import warnings
warnings.filterwarnings('ignore')
```

MODEL/s DEVELOPMENT AND EVALUATION

- Identification of possible problem-solving approaches(methods)

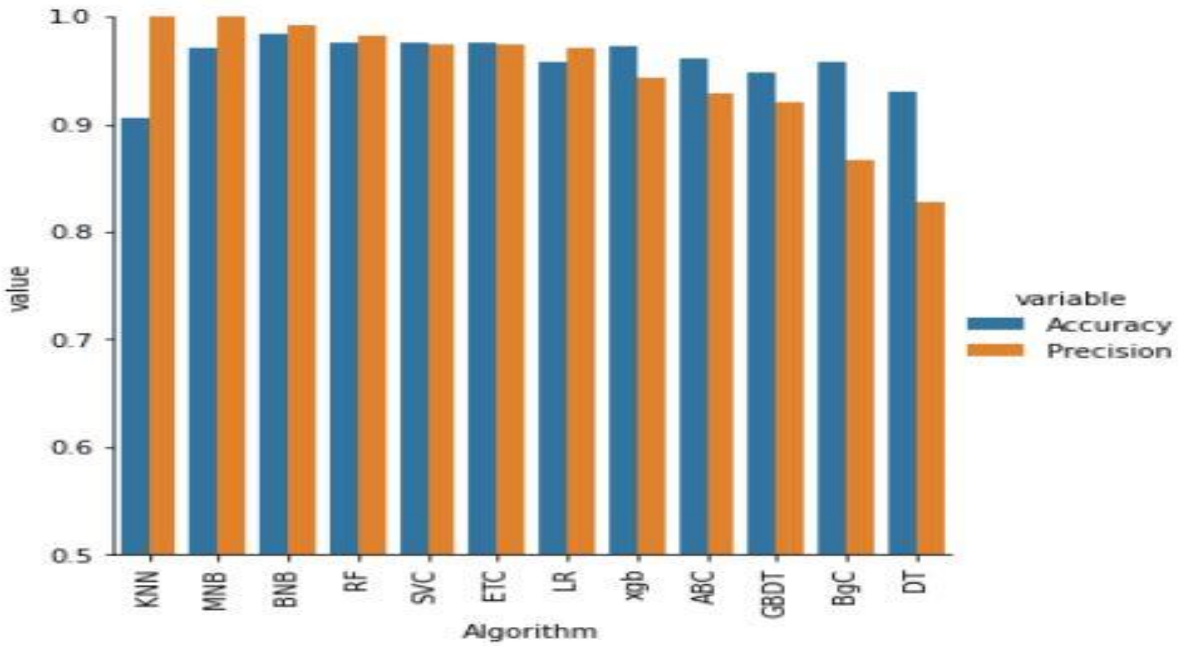
  The collection and interpretation of data in order to uncover patterns and trends. It is a component of data analytics. Statistical analysis can be used in situations like gathering research interpretations, statistical modelling or designing surveys and studies. The approaches/methods of identification are descriptive and inferential statistics which are describes as the properties of sample and population data, and inferential statistics which uses those properties to test hypotheses and draw efficient conclusions in terms of outputs.

- Testing of Identified Approaches(Algorithms)

  There is no outliers.

- Run and Evaluate selected models

```
sns.catplot(x='Algorithm', y='value', hue='variable', data=performance_data1,kind='bar',height=5)
plt.ylim(0.5,1.0)
plt.xticks(rotation='vertical')
plt.show()
```



```
1  temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_max_ft_3000' :accuracy_scores,'Precision_max_ft_3000' : precision_scores}).sort_values(
```

```
1  temp_df = pd.DataFrame({'Algorithm' : clfs.keys(), 'Accuracy_scaling' : accuracy_scores,'Precision_scaling' :precision_scores}).sort_values('Precision_sca
```

```
1  new_df=performance_data.merge(temp_df,on='Algorithm')
```

```
1  new_df_scaled=new_df.merge(temp_df,on='Algorithm')
```

```
1  temp_df=pd.DataFrame({'Algorithm' :clfs.keys(),'Accuracy_num_chars' : accuracy_scores,'Precision_num_chars' : precision_scores}).sort_values('Precis
```

```
1  new_df_scaled.merge(temp_df,on='Algorithm')
```

	Algorithm	Accuracy	Precision	Accuracy_scaling_x	Precision_scaling_x	Accuracy_scaling_y	Precision_scaling_y	Accuracy_
0	KNN	0.905222	1.000000	0.905222	1.000000	0.905222	1.000000	
1	MNB	0.970986	1.000000	0.970986	1.000000	0.970986	1.000000	
2	BNB	0.983559	0.991870	0.983559	0.991870	0.983559	0.991870	
3	RF	0.974855	0.982759	0.974855	0.982759	0.974855	0.982759	
4	SVC	0.975822	0.974790	0.975822	0.974790	0.975822	0.974790	
5	ETC	0.974855	0.974576	0.974855	0.974576	0.974855	0.974576	
6	LR	0.958414	0.970297	0.958414	0.970297	0.958414	0.970297	
7	xgb	0.971954	0.943089	0.971954	0.943089	0.971954	0.943089	
8	ABC	0.960348	0.929204	0.960348	0.929204	0.960348	0.929204	
9	GBDT	0.947776	0.920000	0.947776	0.920000	0.947776	0.920000	
10	BgC	0.957447	0.867188	0.957447	0.867188	0.957447	0.867188	
11	DT	0.929400	0.828283	0.929400	0.828283	0.929400	0.828283	

MODEL BUILDING

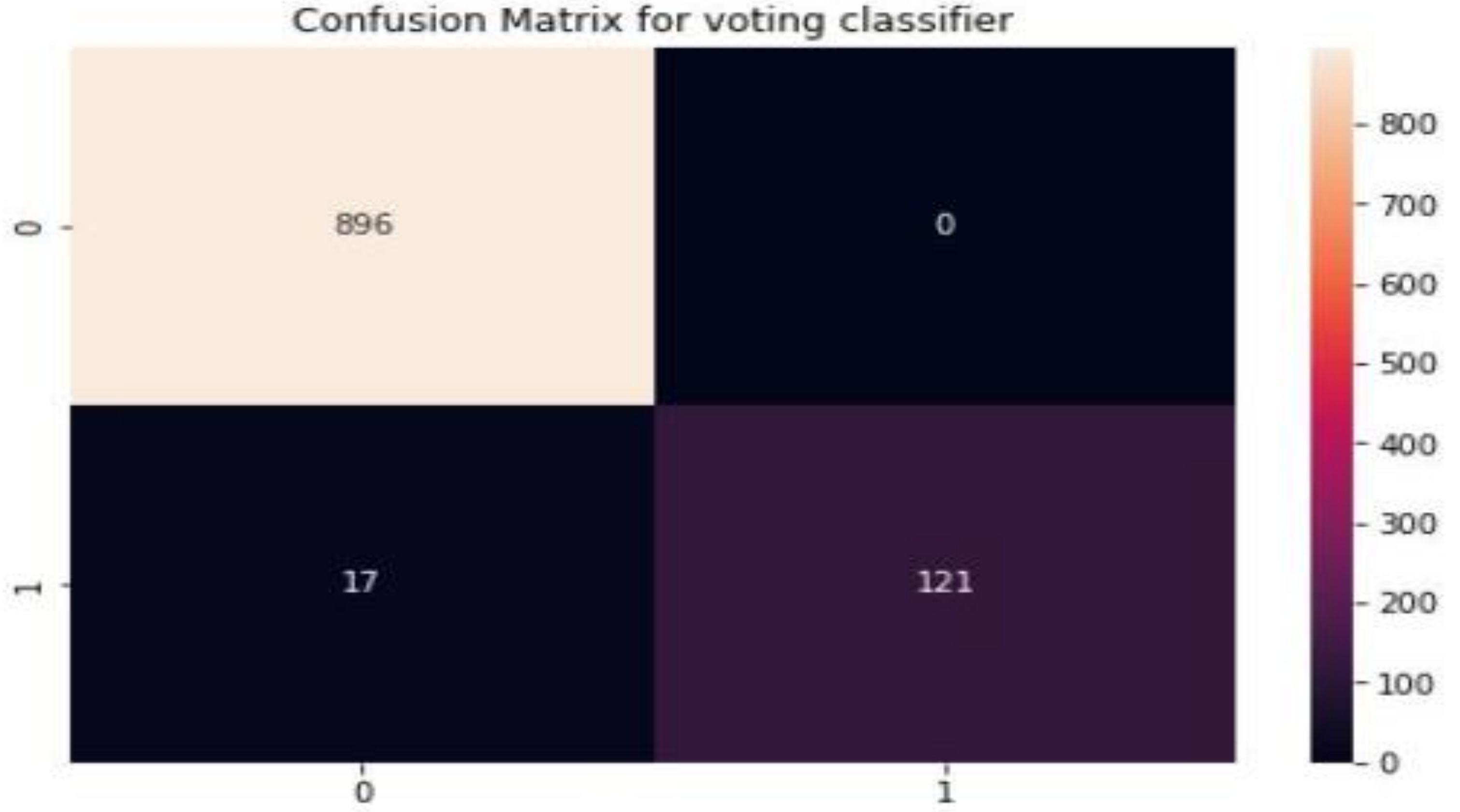
Modelling :

- Using word embedding technique CountVectorizer.

	Algorithm	variable	value
0	KNN	Accuracy	0.905222
1	MNB	Accuracy	0.970986
2	BNB	Accuracy	0.983559
3	RF	Accuracy	0.974855
4	SVC	Accuracy	0.975822
5	ETC	Accuracy	0.974855
6	LR	Accuracy	0.958414
7	xgb	Accuracy	0.971954
8	ABC	Accuracy	0.960348
9	GBDT	Accuracy	0.947776
10	BgC	Accuracy	0.957447
11	DT	Accuracy	0.929400
12	KNN	Precision	1.000000
13	MNB	Precision	1.000000
14	BNB	Precision	0.991870
15	RF	Precision	0.982759
16	SVC	Precision	0.974790
17	ETC	Precision	0.974576
18	LR	Precision	0.970297
19	xgb	Precision	0.943089
20	ABC	Precision	0.929204
21	GBDT	Precision	0.920000
22	BgC	Precision	0.867188
23	DT	Precision	0.828283

- Models used: Email spam classification done using traditional machine learning techniques comprise Naive Bayes and SVM (support vector machines), due to not having sufficient hardware resources, takes less time to train. Also, not opting for neural algorithms due to less data and computing resources.

Accuracy 0.9835589941972921
Precision 1.0



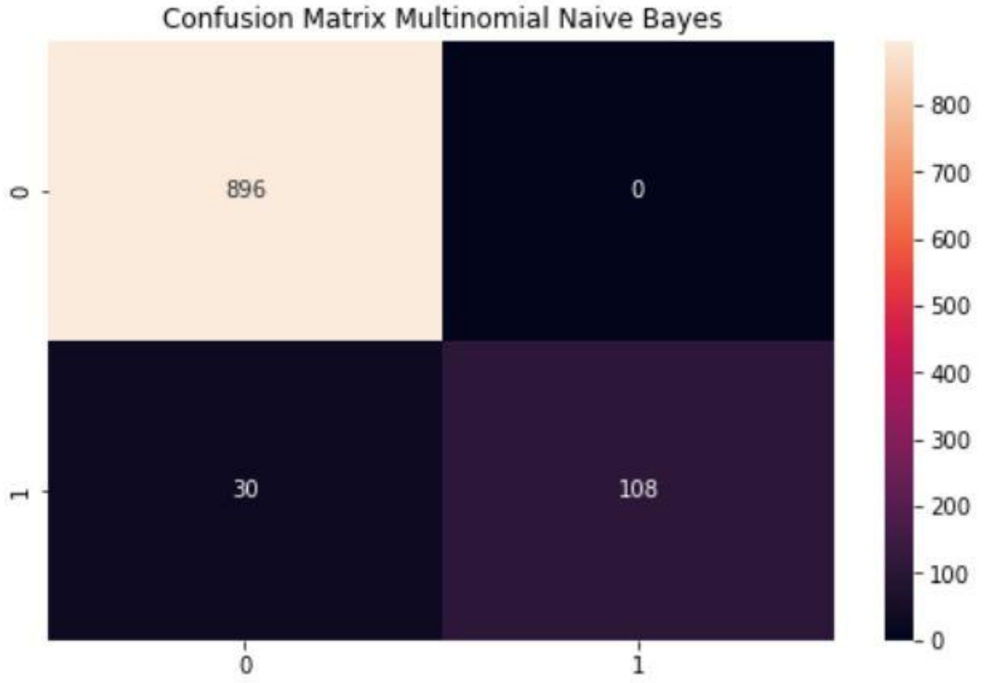
- # Visualizations

```python
mnb.fit(X_train,y_train)
y_pred2 = mnb.predict(X_test)
print("Accuracy Score: ",accuracy_score(y_test,y_pred2))
print(f"Confusion Matrix: \n {confusion_matrix(y_test,y_pred2)}\n")
cm=confusion_matrix(y_test,y_pred2)
print("Precision Score: ",precision_score(y_test,y_pred2))
```

```
Accuracy Score:  0.9709864603481625
Confusion Matrix:
 [[896   0]
 [ 30 108]]

Precision Score:  1.0
```

```python
plt.figure(figsize=(8,5))
plt.title("Confusion Matrix Multinomial Naive Bayes")
sns.heatmap(cm,annot=True,fmt='g')
plt.show()
```



• Interpretation of the Results

The results were interpreted from the visualizations, preprocessing and modelling:

1. Comparing both Naïve Bayes and SVM, I found that Naïve Bayes has 1% improvement over the SVM model when the result compared to test data set.

2. Most of the transactions and business is taking through e-mails.

3.Nowadays, email becomes a powerful tool for communication as it saves a lot of time and cost. But, due to social networks and advertisers, most of the emails contain unwanted information called spam.

4. Even though lot of algorithms has been developed for email spam classification, still none of the algorithms produces 100% accuracy in classifying spam emails.

5. In this project spam dataset is analysed using data mining tool to explore the efficient classifier for email spam classification.

6. Initially, feature construction and feature selection is done to extract the relevant features.

7. Then various classification algorithms are applied over this dataset and cross validation is done for each of these classifiers.

8. Finally , best classifiers for email spam is identified based on the error rate, precision and accuracy.

CONCLUSION

We are able to classify the emails as spam or non-spam.
With high number of emails lots if people using the system it will be difficult to handle all possible mails as our project deals with only limited amount of corpus.

THANK YOU