Spam Prediction using Machine Learning

Group Members:

Partha Sarathi Pal
Asansol Engineering College
Roll:-10800321120(ECE)

Swarnali Mukherjee
Asansol Engineering College
Roll:-10800321121(ECE)

Ritwik Bhattacharya
Asansol Engineering College
Roll:-10800321119(ECE)

Hrithik Raj
Asansol Engineering College
Roll:-10800321122(ECE)

Prasenjit Mondal
Asansol Engineering college
Roll:-10800320048(ECE)

Contents

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty, **Prof. Arnab Chakraborty** for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Partha Sarathi Pal

Swarnali Mukherjee

Ritwik Bhattacharya

Hrithik Raj

Prasenjit Mondal

Project Objective

An email spam filter is a necessity for every individual and organization operating emailing activities on a regular base. An average person roughly receives 100-120 emails a day, out of which an average of 80% of emails are spam. At its very root, keeping your communications flow smooth requires a reliable email spam filter. With the benefits of email spam filters, the security risk can be reduced since the user gets in hand the emails that have gone through various spam checks. Moreover, these email spam filters throw out malware, malicious, and virus-infested emails and protect user security.

Our methodology for solving the problems in the given project is described below:

- Load the required dataset.
- Study the dataset.
- Describe the dataset.
- VAnalyse the dataset.
- Find out if the dataset needs to be processed.
    - It will be determined on the basis of whether the dataset has null values or outliers or any such discrepancy that might affect the output of the models to be trained.
- Find out the principal attributes for training.
- Split the given dataset for training and testing purpose.
- Fit the previously split train data in the aforementioned 4 models.
- Calculate the accuracy of the 4 models and find out the classification reports.
- Plot the necessary graphs.
- Use each trained model to predict the outcomes of the given test dataset.
- Choose the best model among the 4 trained models bases on the accuracy and classification reports.

Project Scope

The broad scope of 'Spam Prediction using Machine Learning' project is given below:

- The given dataset has attributes based on which the whether a mail is spam or not will be predicted.
- It is a useful project as the Classifier models can be used to quickly identify if a mail is spam or not .
- Various banking institution can use these models and modify them according to their needs to use in their loan approval status. This will reduce the manual labour and time spent on determining whether to approve a loan application.
- Customers who intend to take a loan can use these trained models to check whether their loan application will be approved or not. The trained models would be required to be implemented in a platform or interface easily accessible as well as with an easy GUI.
- The dataset given to us is a shortened form of the original dataset from Kaggle. So, the results might have some mismatch with the real-world applications. But that can be avoided if the models are trained accordingly.

Data Description

Source of the data: Kaggle. The given dataset is a shortened version of the original dataset in Kaggle.

Data Description: The given train dataset has 5579 rows and 3 columns.

Sl.no	Category	Message
1	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
2	ham	Ok lar... Joking wif u oni...
3	Spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
4	Ham	U dun say so early hor... U c already then say...
5	ham	Nah I don't think he goes to usf, he lives around here though
6	Spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, å£1.50 to rcv
7	Ham	Even my brother is not like to speak with me. They treat me like aids patent.
8	ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune.

The following table shows the 5 number statistics of the given dataset:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term
Mean	4122.83000	1700.550000	134.600000	341.130000
Standard deviation	2258.89434	1947.668891	62.103856	61.022211
Minimum	1000.00000	0.000000	17.000000	60.000000
25%	2636.00000	0.000000	100.000000	360.000000
50%	3598.00000	1558.500000	120.000000	360.000000
75%	4710.00000	2394.500000	152.750000	360.000000
Maximum	12841.00000	10968.000000	349.000000	480.000000

Table 2: 5 number statistics of the given dataset

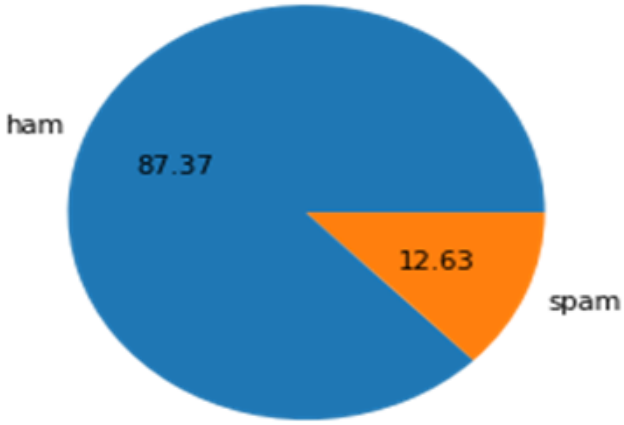
Now we will pre-process the data. The methodology followed is given below:

- Checking for null values.
    - If null values are present, we will fill them or drop the row containing the null value based on the dataset.
- Checking for outliers.
    - If outliers are present, they will either be removed or replaced by following a suitable method depending on the dataset.

Data Processing

As the given dataset is not a cleaned one, so we processed the data and applied changes accordingly.

In this dataset we have counted 87.37 ham mail and 12.63 spam mail.



```
import pandas as pd
import numpy as np
```

```
df=pd.read_csv('/content/mail_data.csv')
df.sample(5)
```

	Category	Message	Unnamed
2017	ham	Princess, is your kitty shaved or natural?	NaN
332	ham	Maybe i could get book out tomo then return it...	NaN
5115	spam	Get 3 Lions England tone, reply lionm 4 mono o...	NaN
5404	ham	Pls give her prometazine syrup. 5mls then &lt...	NaN
3769	ham	Where are you call me.	NaN

```
df.head()
```

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

```
df.shape #to get the dimension of dataset
```

```
(5578, 3)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5578 entries, 0 to 5577
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Category  5578 non-null   object
 1   Message   5572 non-null   object
 2   Unnamed   152 non-null    object
dtypes: object(3)
memory usage: 130.9+ KB
```

```
df.columns
```

```
Index(['Category', 'Message', 'Unnamed'], dtype='object')
```

```
df.sample(5)
```

	Category	Message	Unnamed
5012	spam	You have WON a guaranteed £1000 cash or a £200...	NaN
1782	ham	;-(oh well, c u later	NaN
4021	ham	University of southern california.	NaN
5507	ham	I want to be inside you every night...	NaN
4418	ham	says that he's quitting at least5times a day s...	NaN

```
df.isnull().sum()
```

```
Category       0
Message        6
Unnamed     5426
dtype: int64
```

```
df=df.dropna(subset=['Message'])
```

```
df.isnull().sum()
```

```
Category       0
Message        0
Unnamed     5420
dtype: int64
```

```
df.head()
```

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

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df.drop(columns=['Unnamed'],inplace=True)
```

```
df.sample(5)
```

	Category	Message
3171	ham	Mah b, I'll pick it up tomorrow
811	ham	S:)s.nervous <#> :)
2737	ham	Really? I crashed out cuddled on my sofa.
5104	ham	A Boy loved a gal. He propsd bt she didnt mind...
5234	ham	Ok cool. See ya then.

```
df.duplicated().sum()
```

```
418
```

```
df=df.drop_duplicates(keep='first')
```

```
df.duplicated().sum()
```

```
0
```

```
df.shape
```

```
(5154, 2)
```

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

```
df['Category']=Encoder.fit_transform(df['Category'])
```

```
df.head(10)
```

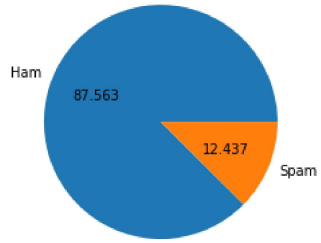
	Category	Message

Data Analysis

```
df['Category'].value_counts()
```

```
0    4513
1     641
Name: Category, dtype: int64
```

```
import matplotlib.pyplot as plt
plt.pie(df['Category'].value_counts(),labels=['Ham','Spam'],autopct="%0.3f")
plt.show()
```




```
df['Message'].value_counts()
```

```
Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
1
Promotion Number: 8714714 - UR awarded a City Break and could WIN a £200 Summer Shopping spree every WK. Txt STORE to 88039 .
SkilGme. TsCs087147403231Winawk!Age16 £1.50perWKsub    1
Senthil group company Apnt 5pm.
1
Why i come in between you people
1
It has issues right now. Ill fix for her by tomorrow.
1

..
BIG BROTHER ALERT! The computer has selected u for 10k cash or #150 voucher. Call 09064018838. NTT PO Box CRO1327 18+ BT Landline
Cost 150ppm mobiles vary                                  1
Buy Space Invaders 4 a chance 2 win orig Arcade Game console. Press 0 for Games Arcade (std WAP charge) See o2.co.uk/games 4 Terms
+ settings. No purchase                                   1
Call FREEPHONE 0800 542 0578 now!
1
Did u see what I posted on your Facebook?
1
Yes. Please leave at  &lt;#&gt; . So that at  &lt;#&gt;  we can leave
1
Name: Message, Length: 5154, dtype: int64
```

```
import nltk
nltk.download("stopwords")
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
df['num_char']=df['Message'].apply(len)
```

```
df.head()
```

	Category	Message	num_char
0	0	Go until jurong point, crazy.. Available only ...	111
1	0	Ok lar... Joking wif u oni...	29
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	0	U dun say so early hor... U c already then say...	49
4	0	Nah I don't think he goes to usf, he lives aro...	61

```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
True
```

```
df['num_words']=df['Message'].apply(lambda x:len(nltk.word_tokenize(x)))
```

```
df.head()
```

	Category	Message	num_char	num_words
0	0	Go until jurong point, crazy.. Available only ...	111	24
1	0	Ok lar... Joking wif u oni...	29	8
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37
3	0	U dun say so early hor... U c already then say...	49	13
4	0	Nah I don't think he goes to usf, he lives aro...	61	15

```
df['num_sentences']=df['Message'].apply(lambda x:len(nltk.word_tokenize(x)))
```

```
df[['num_char','num_words','num_sentences']].describe()
```

	num_char	num_words	num_sentences
count	5154.000000	5154.000000	5154.000000
mean	79.109430	18.558207	18.558207
std	58.398333	13.409557	13.409557
min	2.000000	1.000000	1.000000
25%	36.000000	9.000000	9.000000
50%	61.000000	15.000000	15.000000
75%	118.000000	26.000000	26.000000
max	910.000000	220.000000	220.000000

```
#ham messages describe function
df[df['Category']==0][['num_char','num_words','num_sentences']].describe()
```

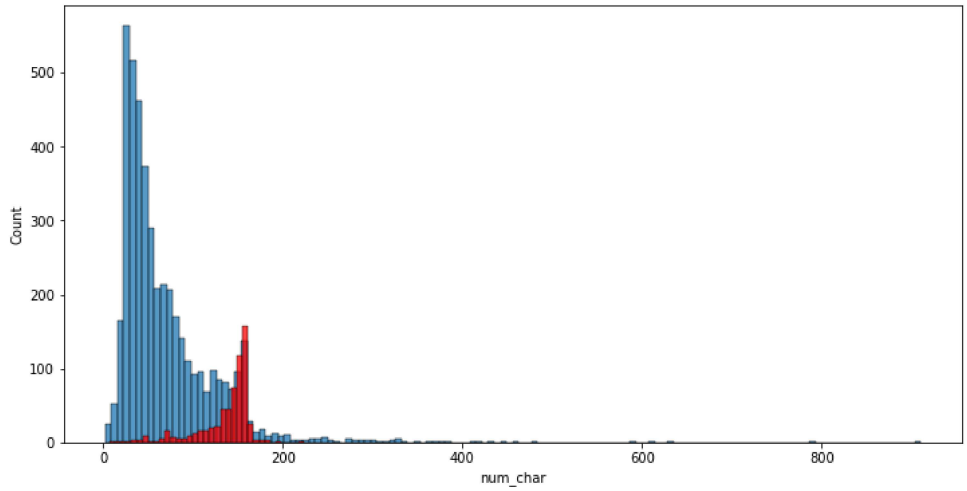
	num_char	num_words	num_sentences
count	4513.000000	4513.000000	4513.000000
mean	70.870153	17.264347	17.264347
std	56.725805	13.591948	13.591948
min	2.000000	1.000000	1.000000
25%	34.000000	8.000000	8.000000
50%	53.000000	13.000000	13.000000
75%	91.000000	22.000000	22.000000
max	910.000000	220.000000	220.000000

```
df[df['Category']==1][['num_char','num_words','num_sentences']].describe()
```

	num_char	num_words	num_sentences
count	641.000000	641.000000	641.000000
mean	137.118565	27.667707	27.667707
std	30.399707	7.103501	7.103501
min	7.000000	2.000000	2.000000
25%	130.000000	25.000000	25.000000
50%	148.000000	29.000000	29.000000
75%	157.000000	32.000000	32.000000
max	223.000000	46.000000	46.000000

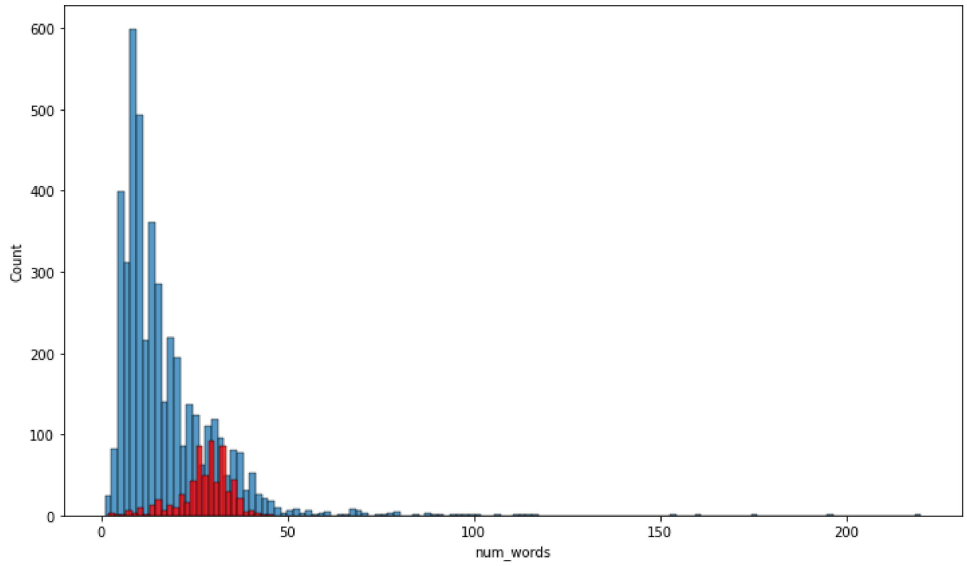
```
plt.figure(figsize=(12,6))
import seaborn as sns
sns.histplot(df[df['Category']==0]['num_char'])
sns.histplot(df[df['Category']==1]['num_char'],color='red')
```

```
<AxesSubplot:xlabel='num_char', ylabel='Count'>
```



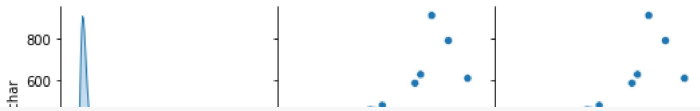

```
plt.figure(figsize=(12,7))
sns.histplot(df[df['Category']==0]['num_words'])
sns.histplot(df[df['Category']==1]['num_words'],color='red')
```

```
<AxesSubplot:xlabel='num_words', ylabel='Count'>
```



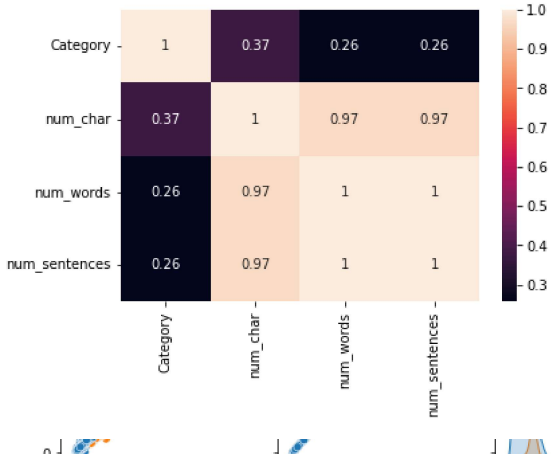

```
sns.pairplot(df,hue='Category')
```

```
<seaborn.axisgrid.PairGrid at 0x7fba67728820>
```




```
sns.heatmap(df.corr(),annot=True)
```

```
<AxesSubplot:>
```



DATA PROCESSING

```
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 loved the YT lectures on Machine Learning. How about you??")
```

```
'love yt lectur machin learn'
```

```
df['Message'][1]
```

```
'Ok lar... Joking wif u oni...'
```

```
df['Message'][0]
```

```
'Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got am
ore wat...'
```

```
transform_text("did You LIke my presentation on ML?")
```

```
'like present ml'
```

```
import string
string.punctuation
```

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```
from nltk.corpus import stopwords
stopwords.words("english")
```

```
['i',
 'me',
```

```
'my',
'myself',
'we',
'our',
'ours',
'ourselves',
'you',
"you're",
"you've",
"you'll",
"you'd",
'your',
'yours',
'yourself',
'yourselves',
'he',
'him',
'his',
'himself',
'she',
"she's",
'her',
'hers',
'herself',
'it',
"it's",
'its',
'itself',
'they',
'them',
'their',
'theirs',
'themselves',
'what',
'which',
'who',
'whom',
'this',
'that',
"that'll",
'these',
'those',
'am',
'is',
'are',
'was',
'were',
'be',
'been',
'being',
'have',
'has',
'had',
'having',
'do',
'does'.
```

```
from nltk.stem.porter import PorterStemmer
ps=PorterStemmer()
ps.stem("LOVing")
ps.stem("Dancing")
```

```
'danc'
```

```
df['transform_text']=df['Message'].apply(transform_text)
```

```
df.head()
```

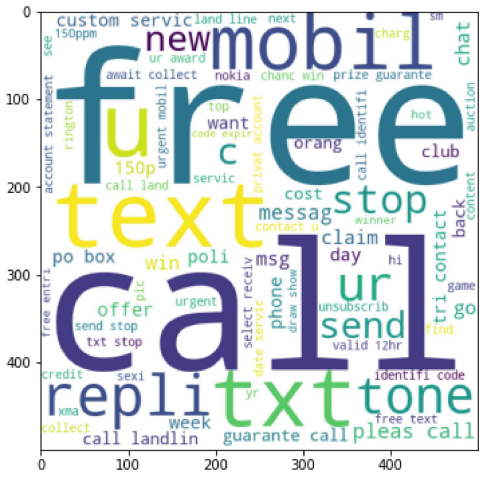
	Category	Message	num_char	num_words	num_sentences	transform_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	24	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	8	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	37	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	49	13	13	u dun say earli hor u c alreadi say

```
from wordcloud import WordCloud
wc=WordCloud(width=500,height=500,min_font_size=10,background_color='white')
```

```
spam_wc=wc.generate(df[df['Category']==1]['transform_text'].str.cat(sep=" "))
```

```
plt.figure(figsize=(15,6))
plt.imshow(spam_wc)
```

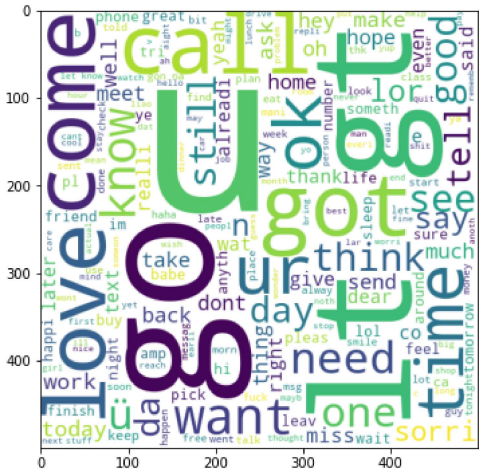
```
<matplotlib.image.AxesImage at 0x7fba5b647250>
```




```
ham_wc=wc.generate(df[df['Category']==0]['transform_text'].str.cat(sep=" "))
```

```
plt.figure(figsize=(12,6))
plt.imshow(ham_wc)
```

```
<matplotlib.image.AxesImage at 0x7fba5b674bb0>
```




```
df.head()
```

	Category	Message	num_char	num_words	num_sentences	transform_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	24	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	8	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	37	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	49	13	13	u dun say earli hor u c alreadi say

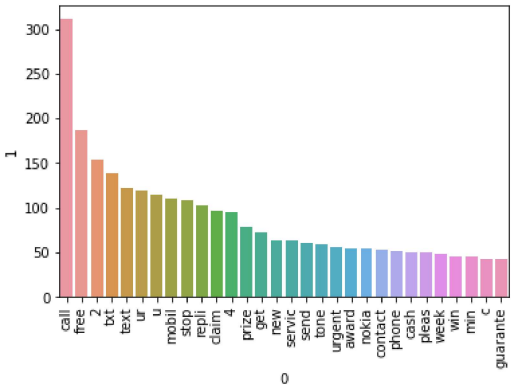
```
spam_corpus=[]
for msg in df[df['Category']==1]['transform_text'].tolist():
    for word in msg.split():
        spam_corpus.append(word)
```

```
len(spam_corpus)
```

```
9781
```

```
from collections import Counter
sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0],pd.DataFrame(Counter(spam_corpus).most_common(30))[1])
plt.xticks(rotation='vertical')
plt.show()
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following var
  warnings.warn(
```



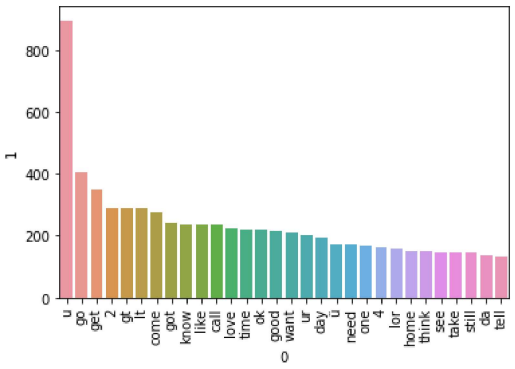

```
ham_corpus=[]
for msg in df[df['Category']==0]['transform_text'].tolist():
    for word in msg.split():
        ham_corpus.append(word)
```

```
len(ham_corpus)
```

```
35902
```

```
from collections import Counter
sns.barplot(pd.DataFrame(Counter(ham_corpus).most_common(30))[0],pd.DataFrame(Counter(ham_corpus).most_common(30))[1])
plt.xticks(rotation='vertical')
plt.show()
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following var
  warnings.warn(
```



Model Building

```
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
cv=CountVectorizer()
tfidf=TfidfVectorizer()
```

```
X=tfidf.fit_transform(df['transform_text']).toarray()
```

```
X.shape
```

```
(5154, 6779)
```

```
y=df['Category'].values
```

```
y
```

```
array([0, 0, 1, ..., 0, 1, 0])
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=2)
```

```
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

```
gnb=GaussianNB()
mnb=MultinomialNB()
bnb=BernoulliNB()
```

```
gnb.fit(X_train,y_train)
y_pred1=gnb.predict(X_test)
print(accuracy_score(y_test,y_pred1))
print(confusion_matrix(y_test,y_pred1))
print(precision_score(y_test,y_pred1))
```

```
0.8661493695441319
[[792 114]
 [ 24 101]]
0.4697674418604651
```

```
mnb.fit(X_train,y_train)
y_pred2=mnb.predict(X_test)
print(accuracy_score(y_test,y_pred2))
print(confusion_matrix(y_test,y_pred2))
print(precision_score(y_test,y_pred2))
```

```
0.9631425800193987
[[906   0]
 [ 38  87]]
1.0
```

```
bnb.fit(X_train,y_train)
y_pred3=bnb.predict(X_test)
print(accuracy_score(y_test,y_pred3))
print(confusion_matrix(y_test,y_pred3))
print(precision_score(y_test,y_pred3))
```

```
0.9709020368574199
[[900   6]
 [ 24 101]]
0.9439252336448598
```

```
#tfidf , mnb
```

```
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
```

```
svc = SVC(kernel='sigmoid', gamma=1.0)
knc = KNeighborsClassifier()
mnb = MultinomialNB()
dtc = DecisionTreeClassifier(max_depth=5)
lrc = LogisticRegression(solver='liblinear', penalty='l1')
rfc = RandomForestClassifier(n_estimators=50, random_state=2)
abc = AdaBoostClassifier(n_estimators=50, random_state=2)
bc = BaggingClassifier(n_estimators=50, random_state=2)
etc = ExtraTreesClassifier(n_estimators=50, random_state=2)
gbdt = GradientBoostingClassifier(n_estimators=50,random_state=2)
xgb = XGBClassifier(n_estimators=50,random_state=2)
```

```
clfs = {
    'SVC' : svc,
    'KN' : knc,
    'NB': mnb,
    'DT': dtc,
    'LR': lrc,
    'RF': rfc,
    'AdaBoost': abc,
    'BgC': bc,
    'ETC': etc,
```

```
    'GBDT':gbdt,
    'xgb':xgb
}
```

```
def train_classifier(clf,X_train,y_train,X_test,y_test):
    clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)
    accuracy = accuracy_score(y_test,y_pred)
    precision = precision_score(y_test,y_pred)

    return accuracy,precision
```

```
train_classifier(svc,X_train,y_train,X_test,y_test)
```

```
(0.9728419010669254, 0.970873786407767)
```

```
accuracy_scores = []
precision_scores = []

for name,clf in clfs.items():

    current_accuracy,current_precision = train_classifier(clf, X_train,y_train,X_test,y_test)

    print("For ",name)
    print("Accuracy - ",current_accuracy)
    print("Precision - ",current_precision)

    accuracy_scores.append(current_accuracy)
    precision_scores.append(current_precision)
```

```
For  SVC
Accuracy -  0.9728419010669254
Precision -  0.970873786407767
For  KN
Accuracy -  0.9078564500484966
Precision -  1.0
For  NB
Accuracy -  0.9631425800193987
Precision -  1.0
For  DT
Accuracy -  0.9456838021338506
Precision -  0.8
For  LR
Accuracy -  0.9544131910766246
Precision -  0.9333333333333333
For  RF
Accuracy -  0.9709020368574199
Precision -  0.9896907216494846
For  AdaBoost
Accuracy -  0.9631425800193987
Precision -  0.865546218487395
For  BgC
Accuracy -  0.9602327837051406
Precision -  0.8559322033898306
For  ETC
Accuracy -  0.976721629485936
Precision -  0.9902912621359223
For  GBDT
Accuracy -  0.9582929194956353
Precision -  0.9361702127659575
For  xgb
Accuracy -  0.944713870029098
Precision -  0.8863636363636364
```

```
performance_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy':accuracy_scores,'Precision':precision_scores}).sort_values('Precision',
```

```
performance_df
```

```python
# Import necessary libraries
from flask import Flask, render_template, request
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from os import environ

app = Flask(__name__)

# Read in data set
data = pd.read_csv('mail_data.csv')



#missing values
data.isnull().sum()

 #check for duplicate values
data.duplicated().sum()

data = data.drop_duplicates(keep='first')


# Preprocess the data
data['Message'] = data['Message'].str.lower() # Convert text to lowercase
data['Message'] = data['Message'].str.replace('[^\w\s]','') # Remove punctuation

# Convert text to numerical feature vectors
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(data['Message'])
y = data['Category']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Train a Naive Bayes classifier
clf = MultinomialNB()
clf.fit(X_train, y_train)

# Test the model
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Model accuracy:", accuracy)

@app.route('/')
def index():
    return render_template('index.html')


# Predict label of new SMS message
@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        new_sms = request.form['message']
    try:


        # Preprocess the input message
        new_sms = vectorizer.transform([new_sms])
        # Make a prediction
        prediction = clf.predict(new_sms)
    except:
            return ('error')


    # Print prediction
    if prediction[0] == 'spam':
        return render_template(
        'spam.html')
    else:
```

```
73          return render_template(
74          'not spam.html')
75
76  if __name__ == '__main__':
77          HOST = environ.get('SERVER_HOST', 'localhost')
78          try:
79              PORT = int(environ.get('SERVER_PORT', '5555'))
80          except ValueError:
81              PORT = 5555
82          app.run(HOST, PORT)
83
```

Future Scope of Improvement

- Various banking institution can use these models and modify them according to their needs to use in their loan approval status. This will reduce the manual labour and time spent on determining whether to approve a loan application.
- Customers who intend to take a loan can use these trained models to check whether their loan application will be approved or not. The trained models would be required to be implemented in a platform or interface easily accessible as well as with an easy GUI.
- We saw a high value of correlation of "Married" attribute with our target attribute. But the feature importance of "Married" attribute was significantly lower. With more data and further analysis, it might be possible to describe the reason of this mismatch.
- No loan application having value "Rural" in "Property_Area" attribute was approved. With further research and more data for analysis, a more decisive conclusion can be made.

Certificate

This is to certify that Mr.Partha Sarathi Pal of Asansol Engineering College, roll number: 10800321120, has successfully completed a project on *Spam Prediction using Machine Learning with Python* under the guidance of Prof. Arnab Chakraborty.

- Prof. Arnab Chakraborty

Certificate

This is to certify that Miss. Swarnali Mukherjee of Asansol Engineering College, roll number: 10800321121, has successfully completed a project on *Spam Prediction using Machine Learning with Python* under the guidance of Prof. Arnab Chakraborty.

- Prof. Arnab Chakraborty

Certificate

This is to certify that Mr. Ritwik Bhattacharya of Asansol Engineering College, roll number: 10800321119, has successfully completed a project on *Spam Prediction using Machine Learning with Python* under the guidance of Prof. Arnab Chakraborty.

- Prof. Arnab Chakraborty

Certificate

This is to certify that Mr. Hrithik Raj of Asansol Engineering College, roll number: 10800321122, has successfully completed a project on *Spam Prediction using Machine Learning with Python* under the guidance of Prof. Arnab Chakraborty.

- Prof. Arnab Chakraborty

Certificate

This is to certify that Mr. Prasenjit Mondal of Asansol Engineering College, roll number: 10800320048, has successfully completed a project on *Spam Prediction using Machine Learning with Python* under the guidance of Prof. Arnab Chakraborty.

- Prof. Arnab Chakraborty