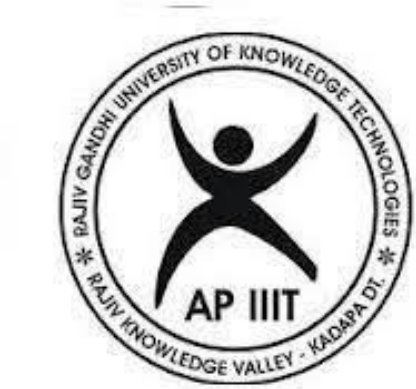
“EMAIL-SPAM CLASSIFICATION”

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING



RGUKT

Rajiv Gandhi University of Knowledge Technologies

R.K.VALLEY

Submitted by

C.MOUNIKA -R170422

Under the Esteemed guidance of

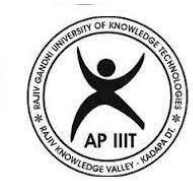
Mr. Satya Nandaram N
RGUKT RK Valley.

DECLARATION

We hereby declare that the report of the B.Tech Major Project Work entitled **"EMAIL-SPAM CLASSIFICATION"** which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for award of any degree.

C.MOUNIKA- R170422
Dept. Of Computer Science and Engineering

RAJIV GANDHIUNIVERSITY OF KNOWLEDGE TECHNOLOGIES



RGUKT

(A.P.Government Act 18 of 2008)
RGUKT, RK VALLEY
Department of Computer Science and Engineering

CERTIFICATE FOR PROJECT COMPLETION

This is certify that the project entitled **"EMAIL SPAM CLASSIFICATION"** submitted by **C.Mounika(R170422)** under our guidance and supervision for the partial fulfillment for the degree Bachelor of Technology in Computer Science and Engineering during the academic semester -2 2021-2022 at RGUKT, RK VALLEY.To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma.

Mr.N.Satya Nandaram ,
Project Internal Guide
Assistant Professor
RGUKT, RK Valley

Mr.P.Harinadha,
Head of the Department
HOD Of CSE
RGUKT, RK Valley

Abstract

EMAIL SPAM detection is a supervised machine learning problem. This means you must provide your machine learning model with a set of examples of spam and ham messages and let it find the relevant patterns that separate the two different categories. .Here we have tried a lot of classification algorithms to see which algorithms performs best. Algorithms like Gaussian naive-bayes, **MultinomialNB, BernoulliNB , DecisionTreeClassifier , RandomForestClassifier etc.** we have also deployed the project on streamlit website using pycharm application.In the website user can clearly see the input box where user enter a message and click on predict ,and the message will be classified as spam/ham based on its pattern.The bernouli NB has performed well with **accuracy score 0.9758220502901354 and precision score of 0.9829059829059829.**So ,we took the same for classification over the other algorithms

Index

INTRODUCTION

Now a days Email/Messages are the primary source of communication .This communication may vary from individual , business ,corporate to government.
With this rapid increase email communication ,there is an increase Spam emails also .
Spam email/message is unsolicited and unwanted junk email sent out in bulk to an indiscriminate recipient list to gain publicity,promoting products,or counterfeit messages. Email spam classification is extremely important for an organization or an individual .Not just filtering but keep garbage out of email inboxes.
In this project we have used various methods to perfectly classify mails as spam or ham. Ham is basically a safe/legit email or message .This was done in many steps from Data collection to Deployment.

Problem Statement

Spam emails are increasing day by day in millions.This is causing a huge financial loss,data loss and even the lifes of people so there is a strict need of spam detection and classification for every organization/individual to overcome this problem.
A tight competition between filtering method and spammers is going on per day, as spammers began to use tricky methods to overcome the spam filters like using random sender addresses or append random characters at the beginning or end of mails subject line.

Gathering Data

The Data set used in this project is downloaded from Kaggle website. Which is a free source of Data sets for machine learning ,Data science.
It is a reliable source , so we took data from kaggle.The step of gathering data is the foundation of the machine learning process.
Dataset :spam.csv
The dataset contains relevant data that can be used to differentiate between the 2 types of messages. Different parameters can be used to classify a message/e-mail as spam or ham.The test Data contains 653 number of spam emails and 4516 of ham messages. These are in human readable form only.

Libraries

We have imported few libraries which are needed for the whole proccess.

1.NUMPY:-

import numpy as np

Numpy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python. It also supports to add large, multidimensional arrays and matrices. So, in Python, we can import it as:

2.SEABORN:-

import seaborn as sns

Seaborn is a library for **making statistical graphics** in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data.

3.SCIKIT-LEARN:-

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

4.PANDAS:-

import pandas as pd

Pandas is **an open source library in Python**. It provides ready to use high-performance data structures and data analysis tools. Pandas module runs on top of NumPy and it is popularly used for data science and data analytics.

5.MATPLOTLIB:-

import matplotlib.pyplot as plt

Matplotlib is a python library **used to create 2D graphs and plots by using python scripts**. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc.

6.Nltk:-import nltk
nltk.download('punkt')
from nltk.stem.porter import PorterStemmer
from nltk.corpus import stopwords

The Natural Language Toolkit (NLTK) is **a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP)**. It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning

7.Pickle:-
import pickle

Pickle in Python is **primarily used in serializing and deserializing a Python object structure**. In other words, it's the process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network.

8.WordCloud
import WordCloud

A word cloud is **a collection, or cluster, of words depicted in different sizes**. The bigger and bolder the word appears, the more often it's mentioned within a given text and the more important it is.

9.String
import string
string.punctuation
Python String module contains some constants, utility function, and classes for *string* manipulation.

Data Pre-processing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.
When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data pre-processing task.

It includes the following steps:-

1.Data Cleaning:-

In the Data cleaning step we have cleaned the data ,which means we just kept the data that we need ,all he remaining data is dropped.And renamed the columns as target and text to understand better with a glance.

2.Removing Null values

In this step we have checked for the null values an found some null values which are unuseful columns so dropped the columns.And later found non-null values.

3.Label Encoding

As the data that we took is having target as a text format as Ham/Spam .It make it understandable to the machine we should encode them as 0 or 1.
By using label Encoder we labeled spam as 1 and Ham as 0.
The Library that we use here is :-

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df['target']=encoder.fit_transform(df['target'])
```

4.Duplicates

The Data set that we had may not have unique values ,it may consist duplicates which causes model to learn the same patterns again and again .That causes model not to perform well while predicting.So it’s better to remove them while pre-processing.
We remove duplicates using :-

```
df.duplicated().sum()
403
df=df.drop_duplicates(keep='first')
df.duplicated().sum()
0
```

2.Eda (Exploratory data analysis):-

We analyse the relationship between the target spam and ham using ploting and graphs.

df.head()

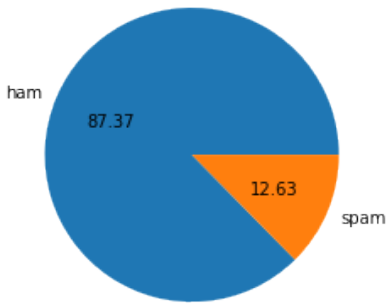
	target	text
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

df['target'].value_counts()

```
0    4516
1     653
Name: target, dtype: int64
```

Now let's see how they are distributed using pie plot from matplotlib

plt.pie(df['target'].value_counts(),labels=['ham','spam'],autopct="%0.2f")



spam :-12.63%
Ham:-87.37%

Findings:

The data is imbalenced.

Number of characters,Number of words and Number of sentences are appended to the data frame. :-

df['num_characters']=df['text'].apply(len)
df['num_words']=df['text'].apply(lambda x:len(nltk.word_tokenize(x)))
df['num_sentences']=df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))

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

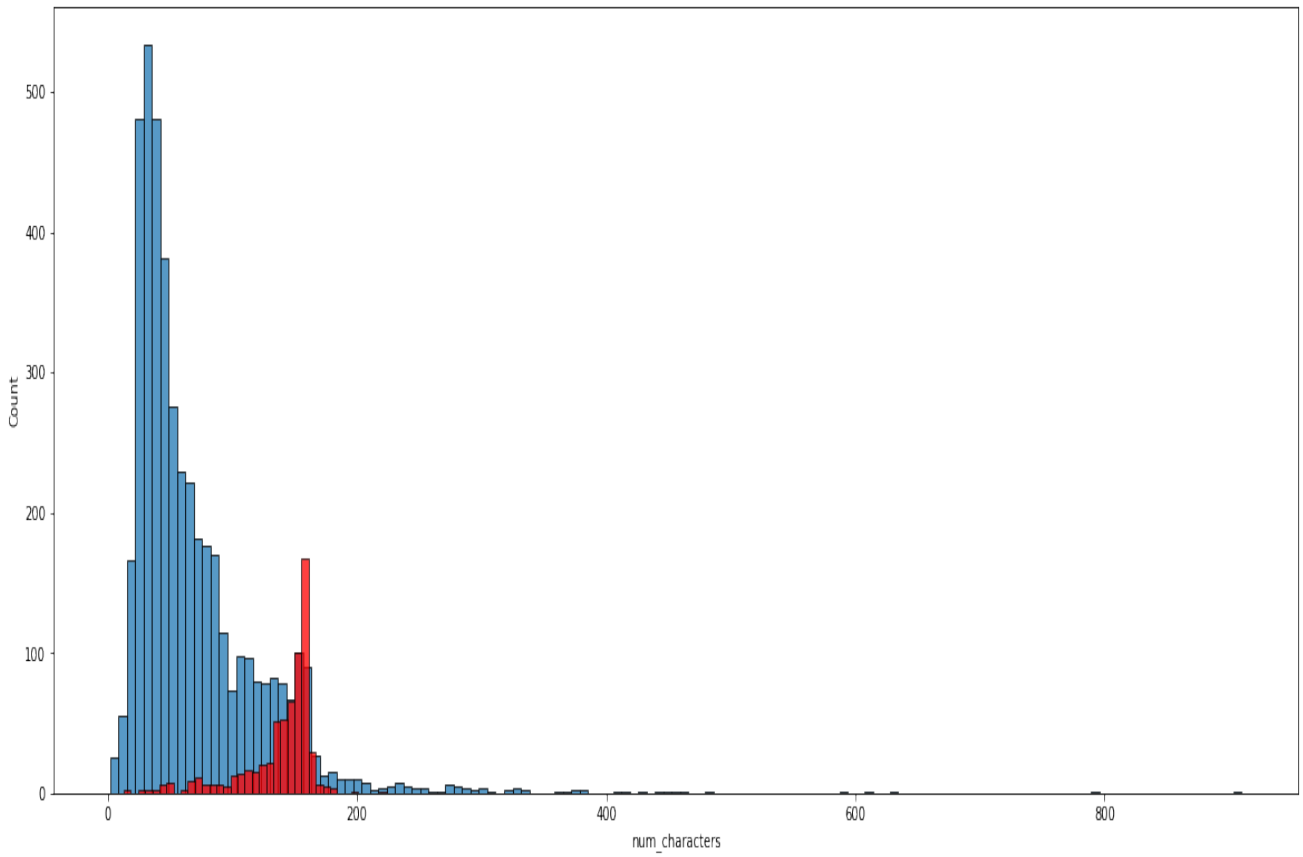
Findings:-

Number of characters and Words in spam mails are less compared to ham mails.

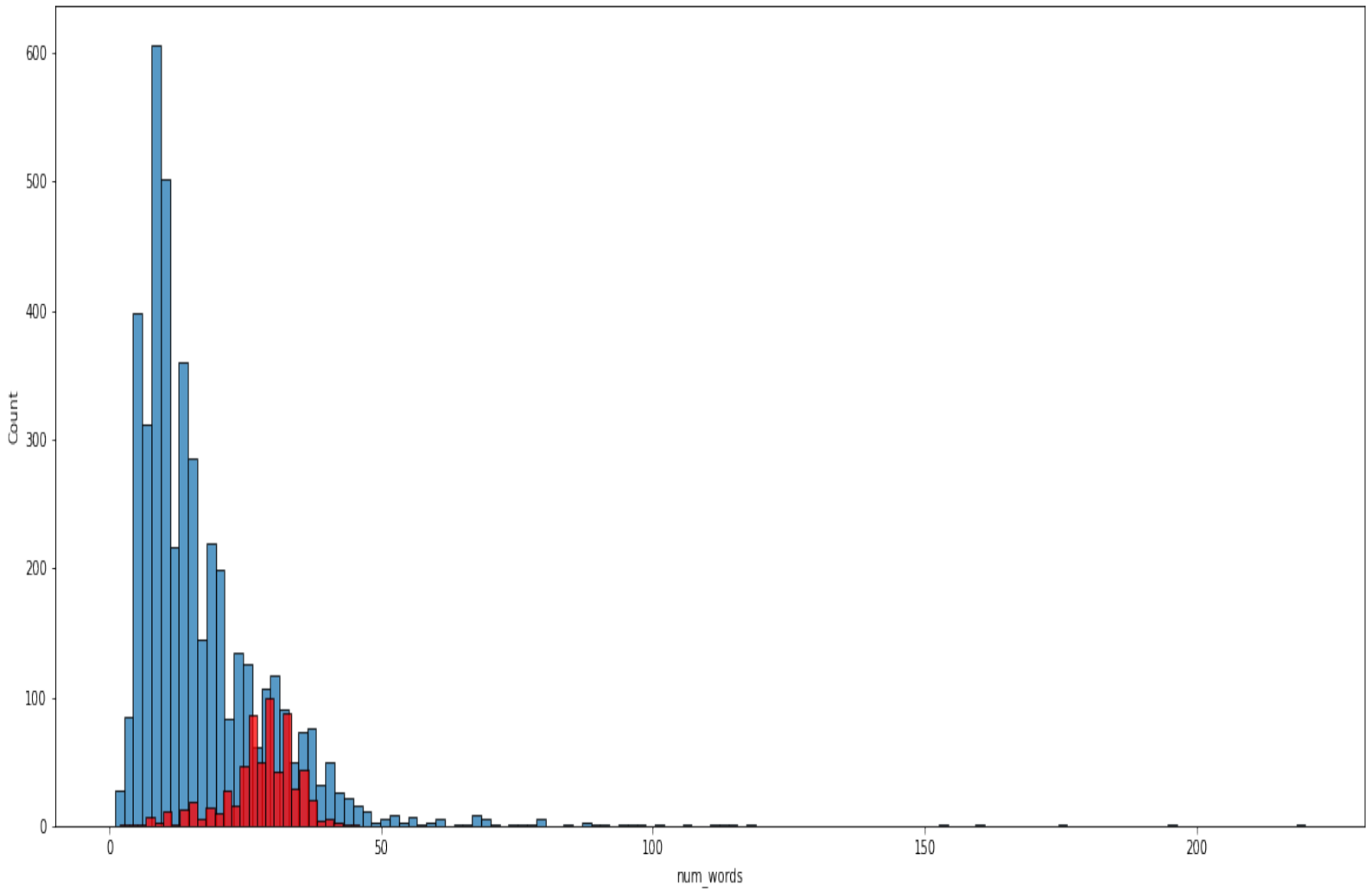
Graphs:-

Now lets see clearly how words ,characters and sentences in both spam and ham are distributed using seaborn graphs.

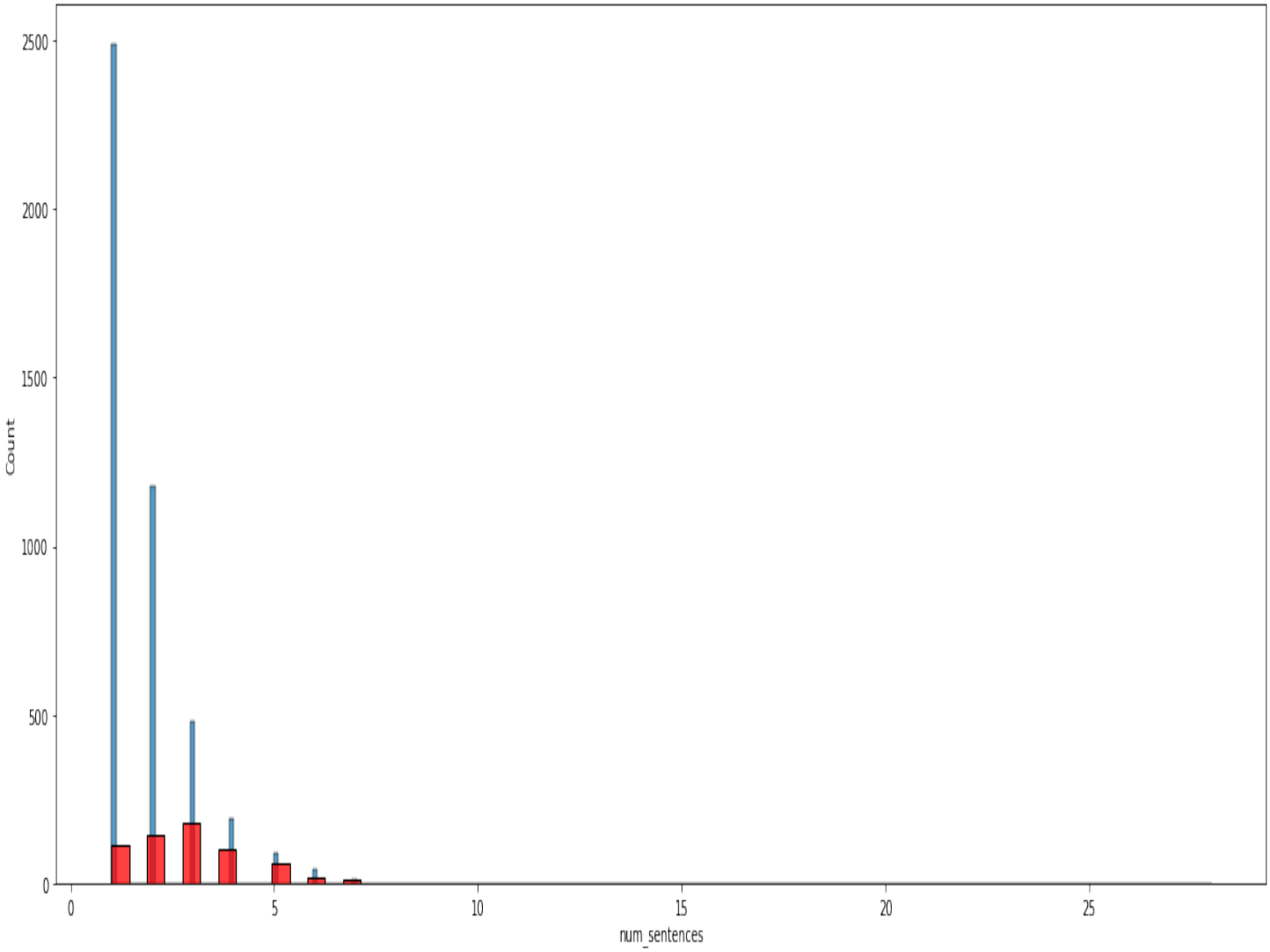
```
plt.figure(figsize=(20,8))
sns.histplot(df[df['target']==0]['num_characters'])
sns.histplot(df[df['target']==1]['num_characters'],color='red')
```



```
plt.figure(figsize=(20,8))
sns.histplot(df[df['target']==0]['num_words'])
sns.histplot(df[df['target']==1]['num_words'],color='red')
```



```
plt.figure(figsize=(20,8))
sns.histplot(df[df['target']==0]['num_sentences'])
sns.histplot(df[df['target']==1]['num_sentences'],color='red')
```

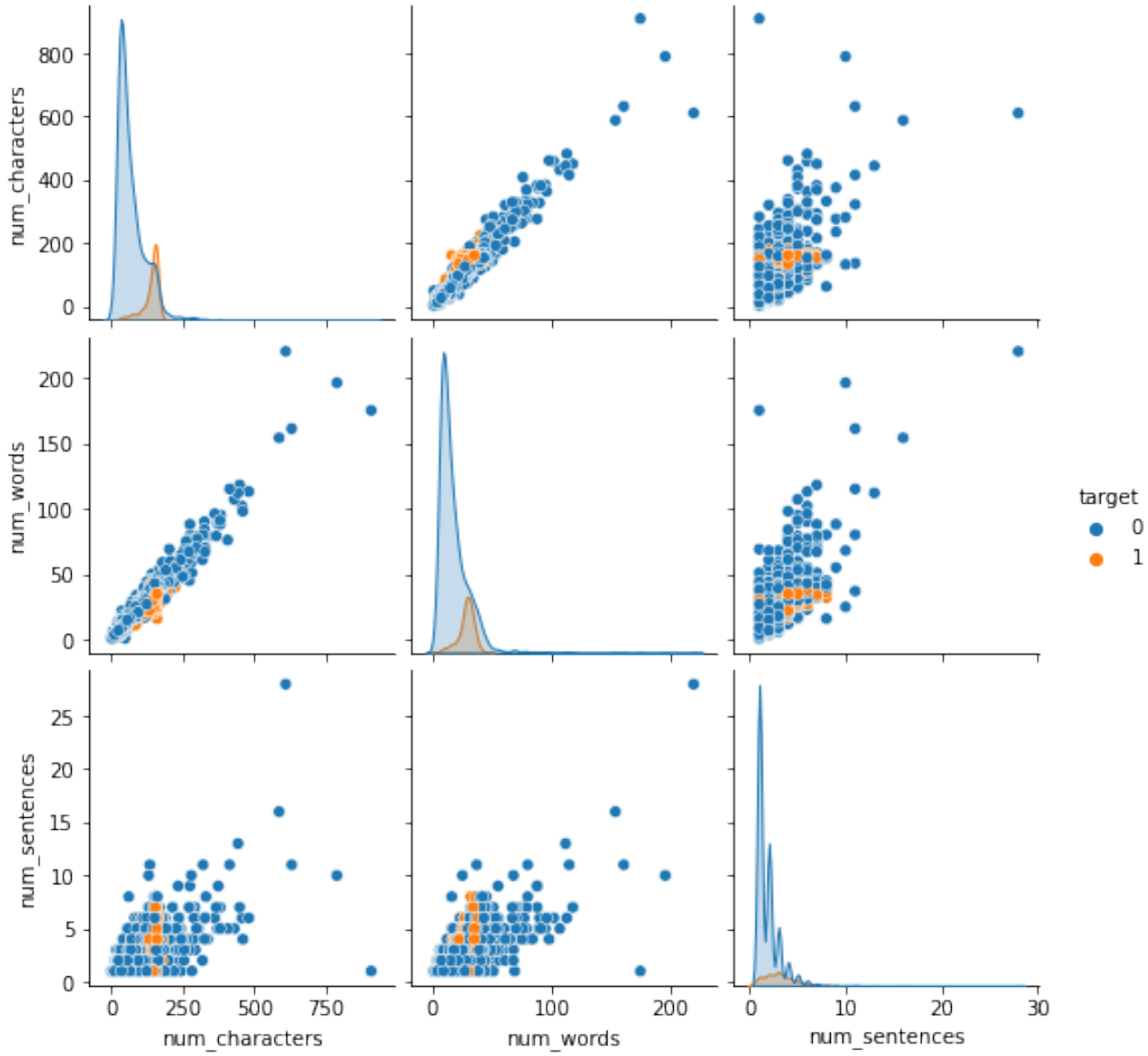


Findings:-

1.Spam mails consists of 189-200 characters occuring too often,where as ham 0-100 are occuring too often.

2.Spam mails contains 30-40 words too often where as ham mails 0-25 are occuring too often .

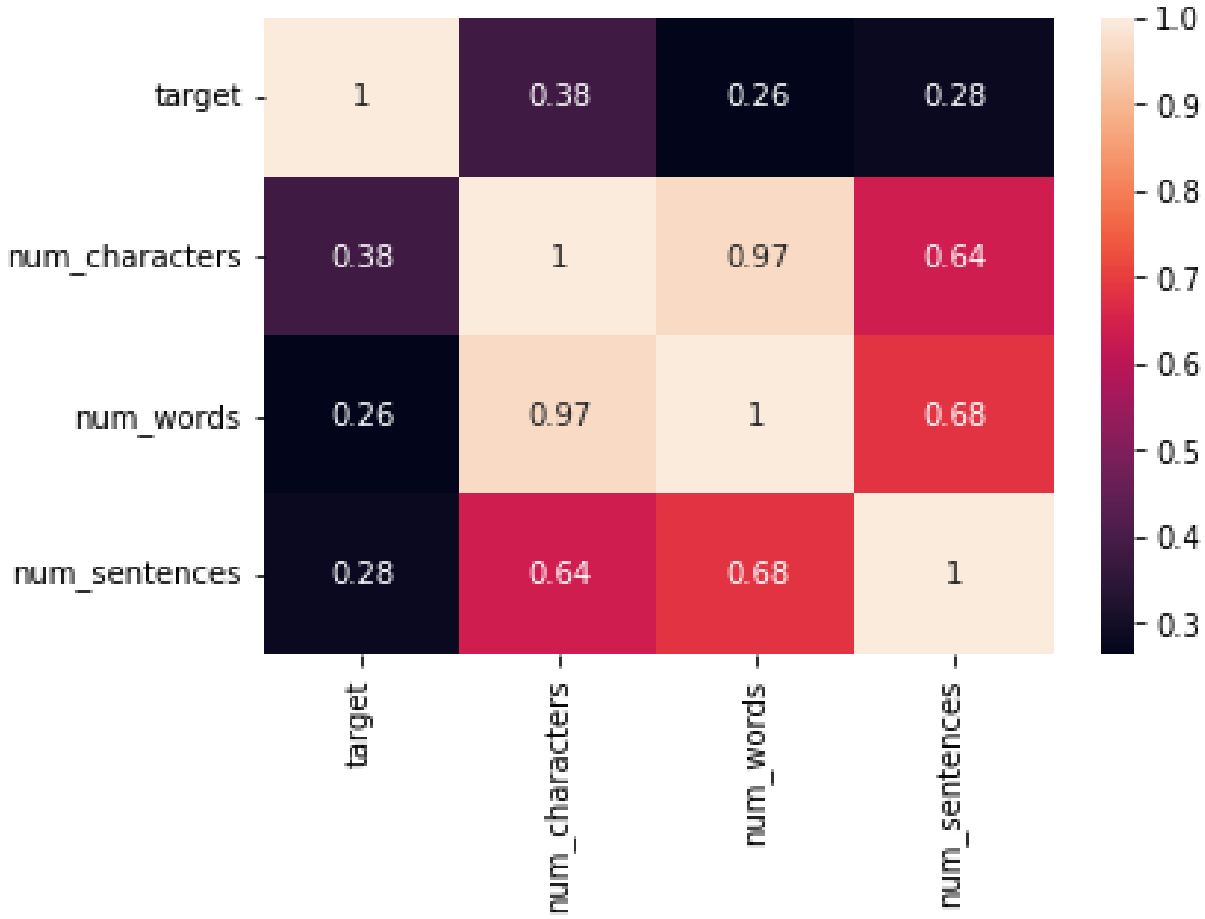
3.Number of sentences below 5 are occuring upto 100 times in spam mail and upto 2500 in ham mails.



Pairplot is describing relationship between spam and ham mail characteristics more clearly .

Heatmap:-

We use heatmap to find strength of correlation among the variables.



Text Pre-processing

Raw data may be uneven like case, and may have unwanted things like stopwards,punctuations etc. So in this step we remove all unnecessary stuff.

Convert to lower case/upper case:-

The data should be even so we make all the characters to lower case .

Tokenization:-

We make the data into tokens wise in this step.Using Tokenizer or can write own code to divide based on spaces/tabs.

Removing special characters:-

We remove all the characters except alphanumeric characters.Like ?+-*& etc.

Removing stop words and punctuation:-

We also remove punctuations residing in the code using string.punctuations.

Stemming:-

As the words in the text may have various forms like present tense,past,continuous tense so,disregarding the grammer we use ‘Bag of words’ to normalize text to its original format.Using portStemmer() method from nltk library.

Ex:-
loving
ps.stem(loving)
love#stemmed

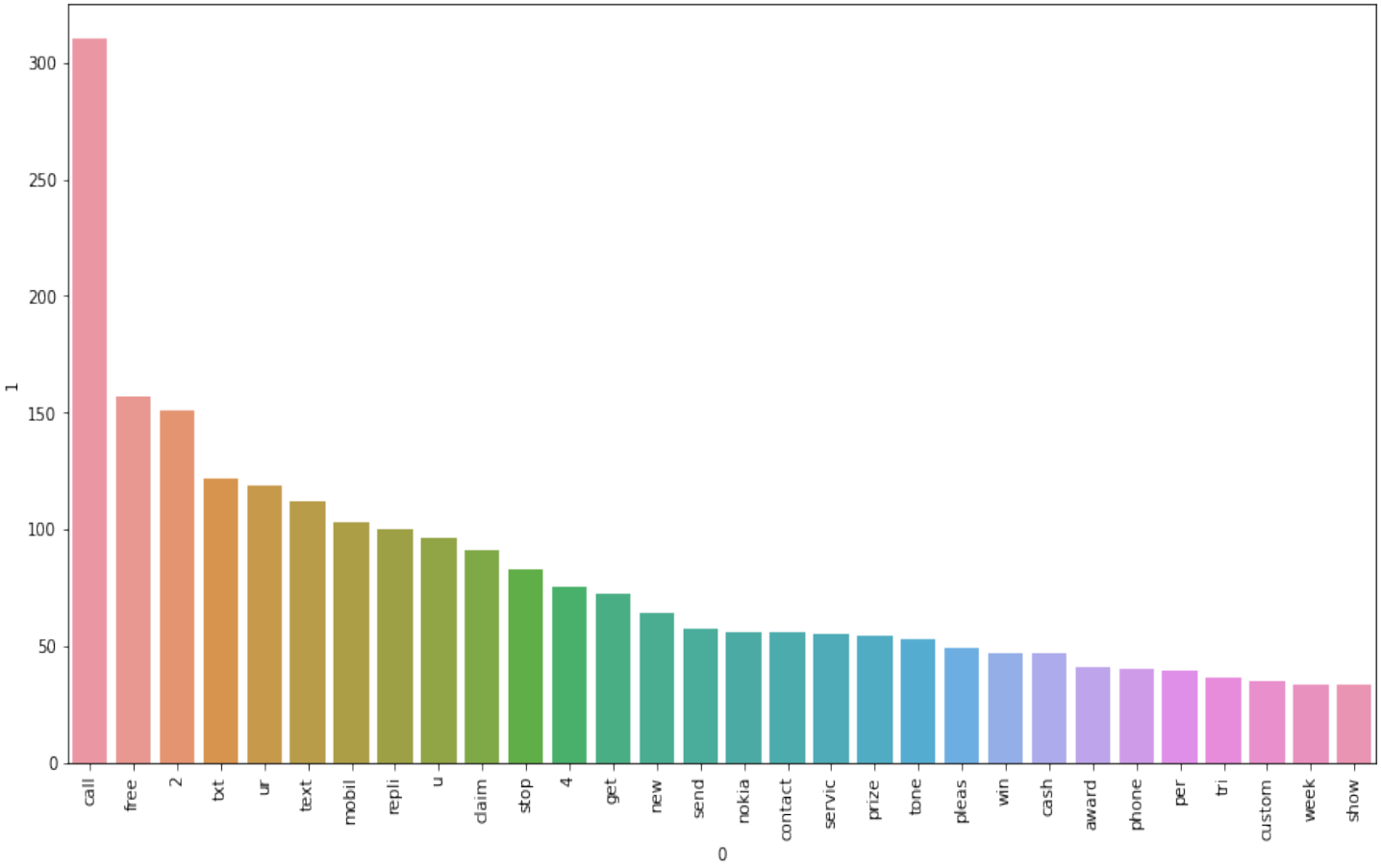
Word Cloud:-

It is a data visualization techniqe used for representing text data in which the size of each word indicates its frequency or importance.

```
spam_wc=wc.generate(df[df['target']==1]['transformed_text'].str.cat(sep=" "))
plt.figure(figsize=(12,10))
plt.imshow(spam_wc)
```

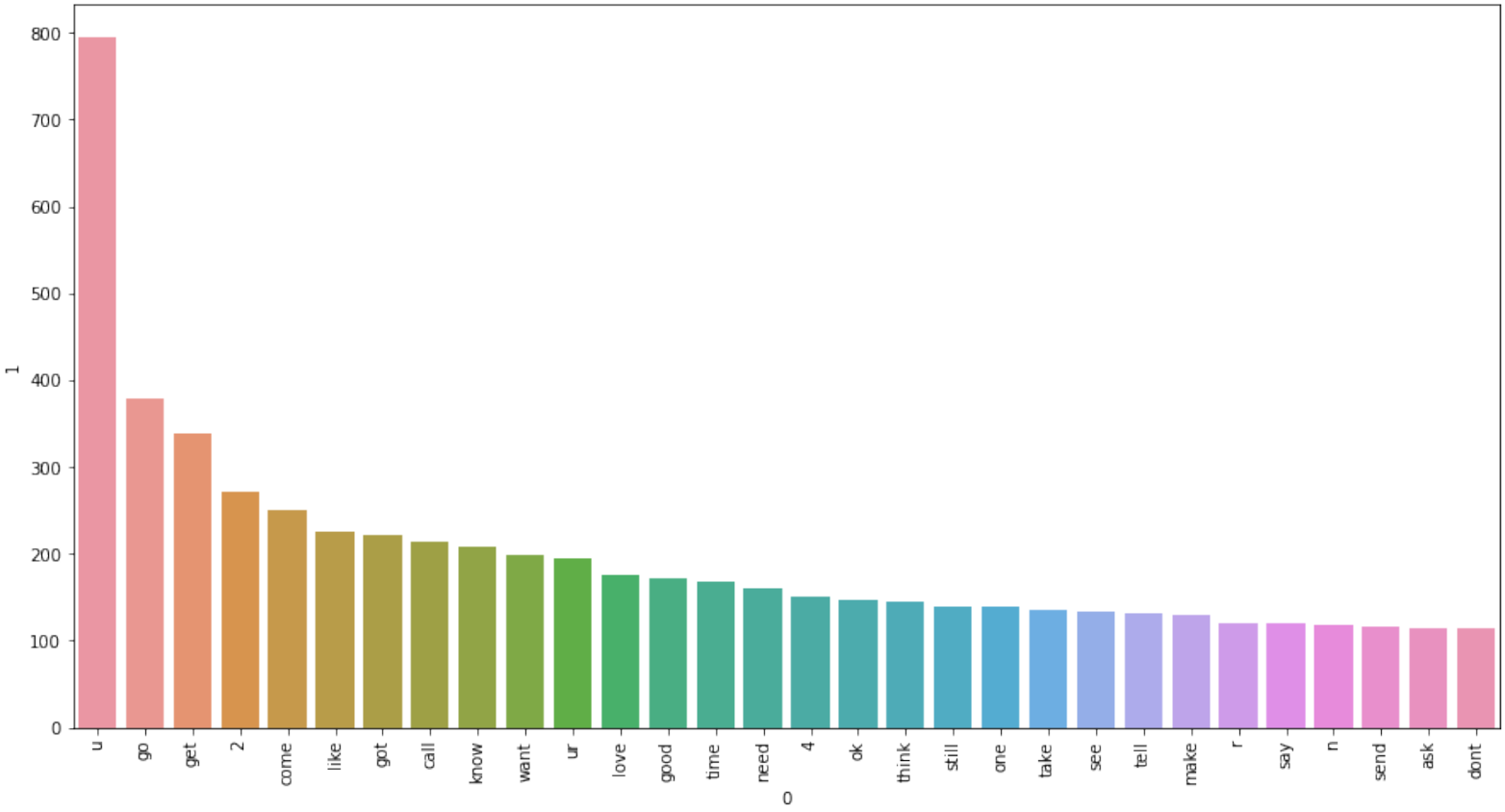
```
ham_wc=wc.generate(df[df['target']==0]['transformed_text'].str.cat(sep=" "))
plt.figure(figsize=(12,10))
plt.imshow(ham_wc)
```

Count of the frequently occuring words in spam messages are as follows this graph was printed using counter library.



Findings:- Most frequently occurred words are call,free,your,claim,mobile ..etc

Count of the frequently occuring words in ham messages are as follows this graph was printed using counter library.



Findings:- Most frequently occurred words are you,go,get,come,call,love ..etc

Model Building

Choose a model:-

Our main goal is to train the best performing model possible, using the pre-processed data.

Supervised Learning:

In Supervised learning, an Machine learning system is presented with data which is labelled, which means that each data tagged with the correct label.

The supervised learning is categorized into 2 other categories which are "**Classification**" and "**Regression**".

Classification:

Classification problem is when the target variable is **categorical** (i.e. the output could be classified into classes — it belongs to either Class A or B or something else).

These some most used classification algorithms.

- **K-Nearest Neighbor**
- **Naive Bayes**
- **Decision Trees/Random Forest**
- **Support Vector Machine**
- **Logistic Regression**

  Ours is a Classification problem with target 0 and 1. And the classification algorithm best suit is Naive bayes, bernouli naive bayes.

Training and testing the model:-

First we need to split data into train and test sets.

Training set: The training set is the material through which the computer learns how to process information. Machine learning uses algorithms to perform the training part. A set of data used for learning, that is to fit the parameters of the classifier.

Test set: A set of unseen data used only to assess the performance of a fully-specified classifier.

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)

Now we have to import the libraries needed first.

from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliN

from sklearn.metrics import accuracy_score,confusion_matrix,precision_score

Confusion Matrix:- It is a performance measurement for machine learning classification.

Precision:- Precision is how good the model is at predicting a specific category.

Accuracy:- Accuracy tells you how many times the ML model was correct overall

Evaluation:-

```
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))
```

Gaussian NB:-

```
0.7940038684719536
[[707 189]
 [ 24 114]]
0.3762376237623762
```

Findings:-

It has very less precession and accuracy.

MultinomialNB:-

```
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))
0.9632495164410058
[[895   1]
 [ 37 101]]
0.9901960784313726
```

Findings:-

It has good precession but accuracy is not upto the mark.

Bernouli NB:-

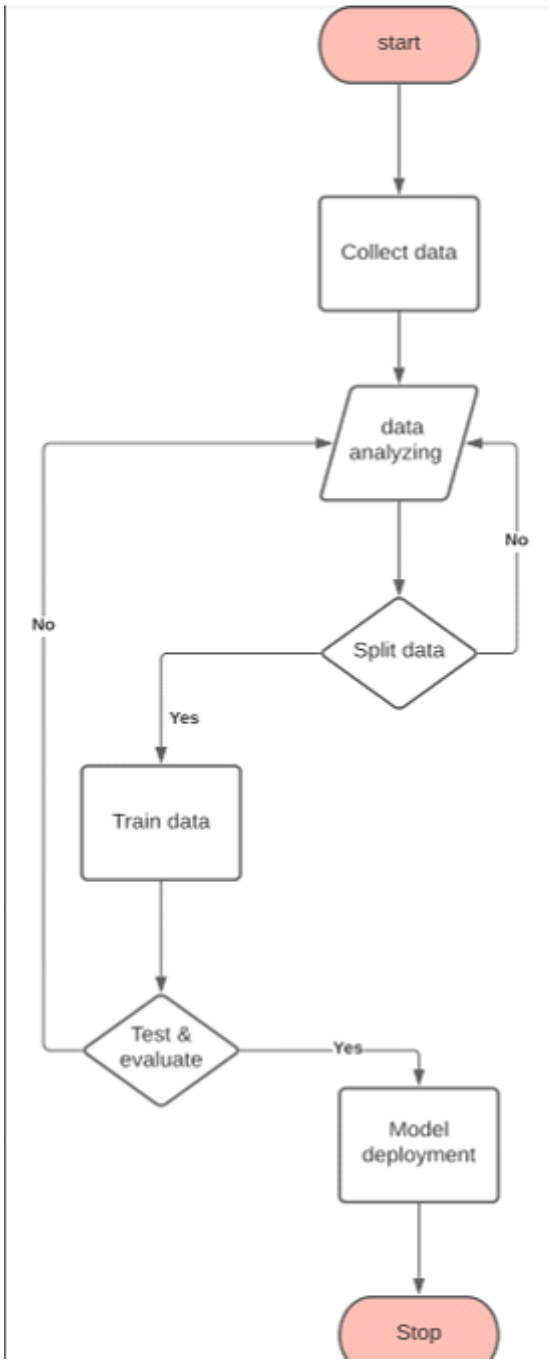
```
bnb.fit(X_train,y_train)
y_pred3=bnb.predict(X_test)
print(accuracy_score(y_test,y_pred3))
print(confusion_matrix(y_test,y_pred3))
print(precision_score(y_test,y_pred3))

0.9758220502901354
[[894   2]
 [ 23 115]]
0.9829059829059829

Findings:-
```

It has good precession and good accuracy.So we chose this algorithms.

Process Flow/Flow Chart



Improvements:-

We improve the model by using Lemmatization instead of stemming ,I used stemming because of it's speed. But lemmatization is context based and do accurately whereas as stemming is not context based but just remove suffixes.

And we may also check more Algorithms to perform better than Naive Bayes.

Deployment

Before deployment we have change the code in object form to byte stream.So we used pickle library to do this.

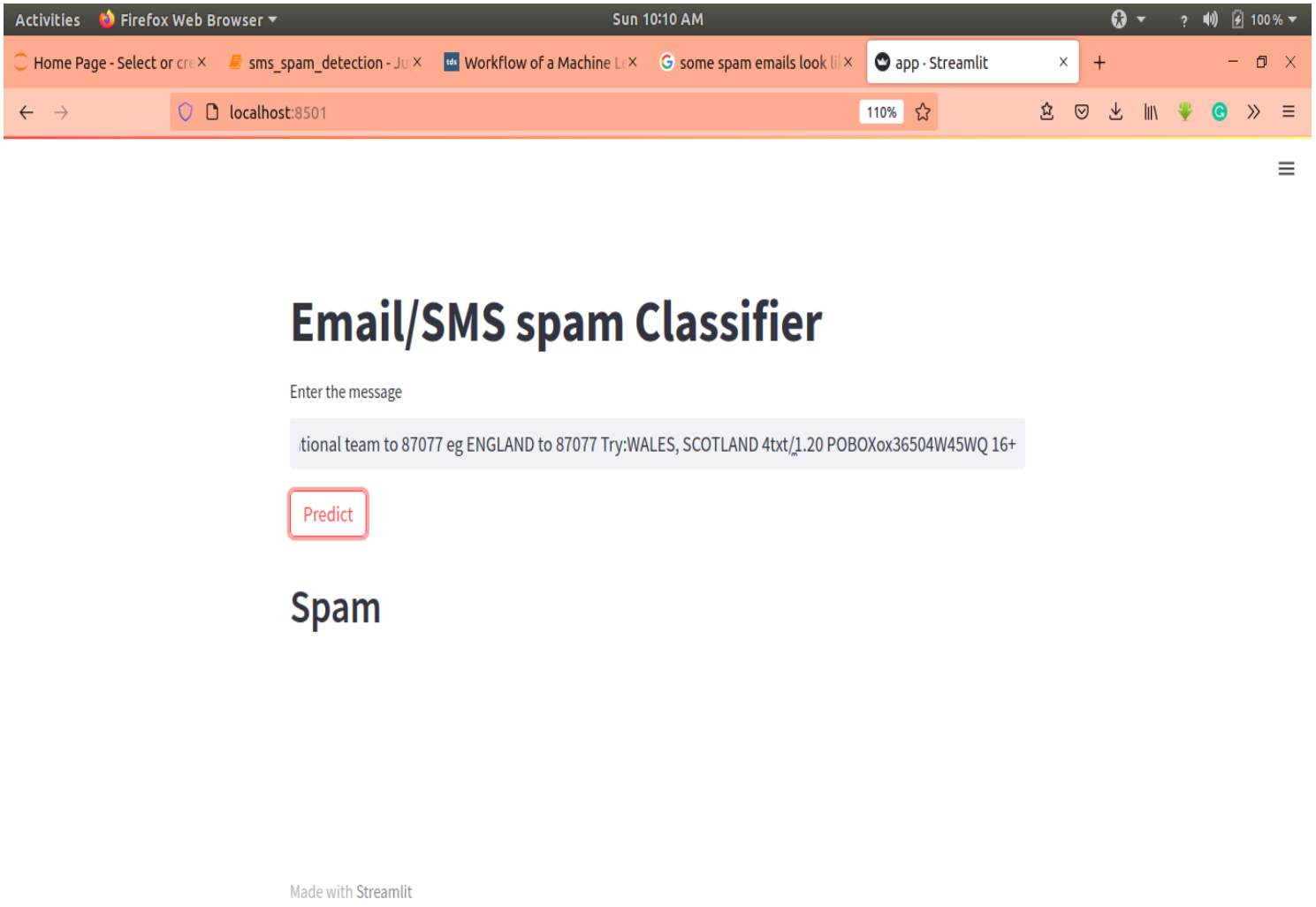
```
import pickle
pickle.dump(tfidf,open('vectorizer.pk1','wb'))
pickle.dump(mnb,open('model.pkl','wb'))
```

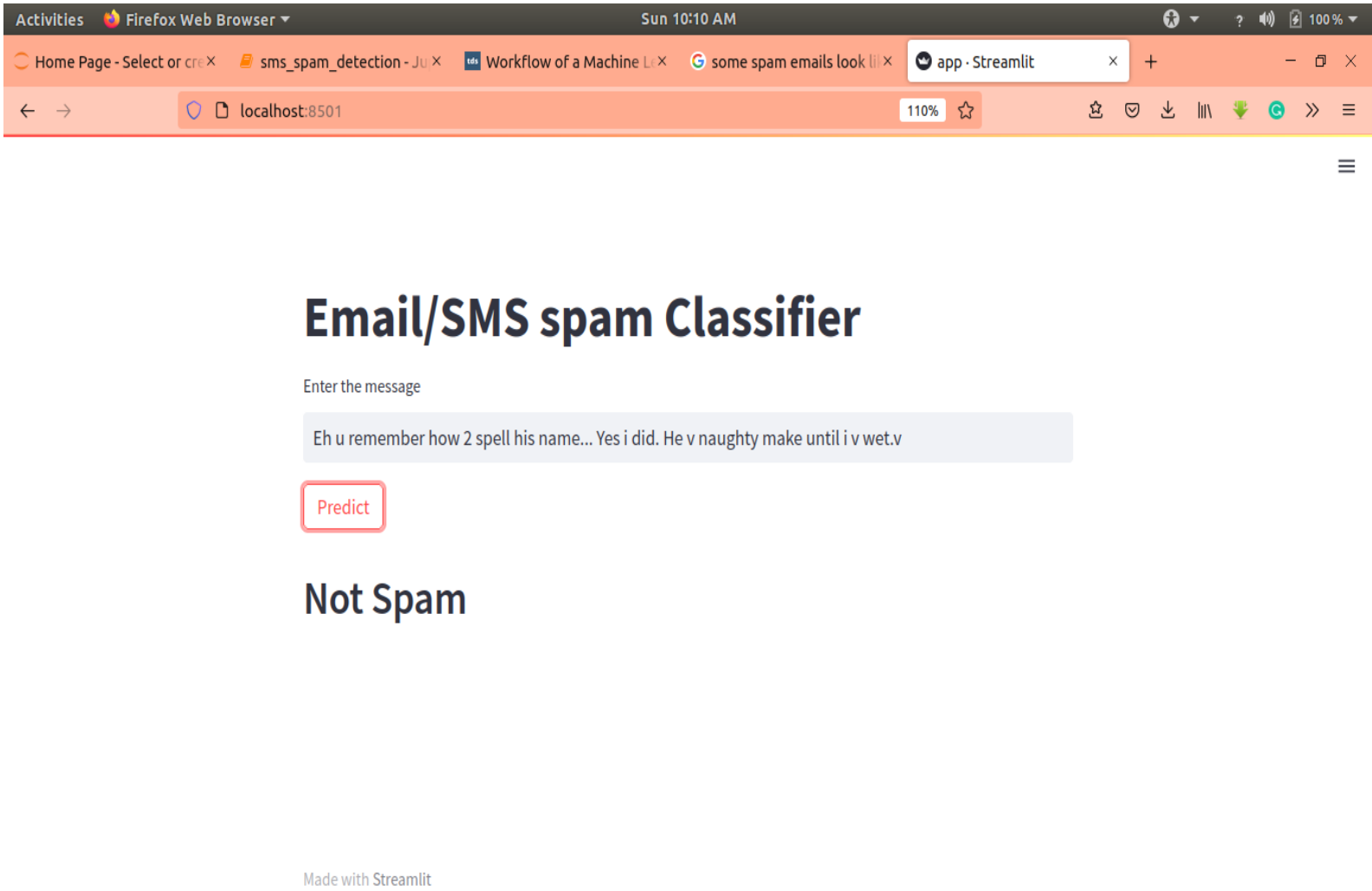
Now in the pycharm application we loaded all the necessary files and started deploying application.

```
st.title("Email/SMS spam Classifier")
input_sms = st.text_input("Enter the message")
if st.button('Predict'):
    # 1.Preprocess
    transformed_sms = transform_text(input_sms)
    # 2.vectorize
    vector_input = tfidf.transform([transformed_sms])
    # 3.predict
    result = model.predict(vector_input)[0]
    # 4.Display
    if result == 1:
        st.header("Spam")
    else:
        st.header("Not Spam")
```

Result:-

The spam and ham messages are now getting predicted by my model.




Activities
Firefox Web Browser
Sun 10:10 AM
100 %
Home Page - Select or cre
sms_spam_detection - Ju
Workflow of a Machine L
some spam emails look li
app · Streamlit
localhost:8501
110%
Email/SMS spam Classifier
Enter the message
Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.v
Predict
Not Spam
Made with Streamlit

Conclusion

Our model performed good with expected accuracy and precision.The messages are getting predicted as spam or ham .We can say Bernoulli Naive Bayes is a good classification algorithm,when compared to other available algorithms like SVM ,Decision Tree,Random forest ..etc.

References

[1]https://www.irjet.net/archives/V8/i1/IRJET-V8I1164.pdf
[2]https://monkeylearn.com/blog/data-preprocessing/
[3]https://www.researchgate.net/figure/Spam-detection-block-diagram_fig1_347267306
[4]https://www.tutorialspoint.com/machine_learning/machine_learning_implementing.htm
[5]https://www.codingninjas.com/codestudio/library/bernoulli-naive-bayes
[6] https://fdocuments.in/document/email-spam-detection-using-machine-learning.html?page=2
[7]https://www.getsafeonline.org/personal/articles/spam-and-scam-email/
[8]https://www.google.com/search?channel=fs&client=ubuntu&q=literature+survey+in+project
[9]S. K. Tuteja, “Classification Algorithms for Email Spam Filtering", 2016.
[10] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi, "Email Classification Research Trends: Review and Open Issues", 2017.