SMS CLASSIFICATION

Chetana N Patil – PES2UG20CS504

Mekala Sanjana – PES2UG20CS194

Steps:

1. Importing libraries

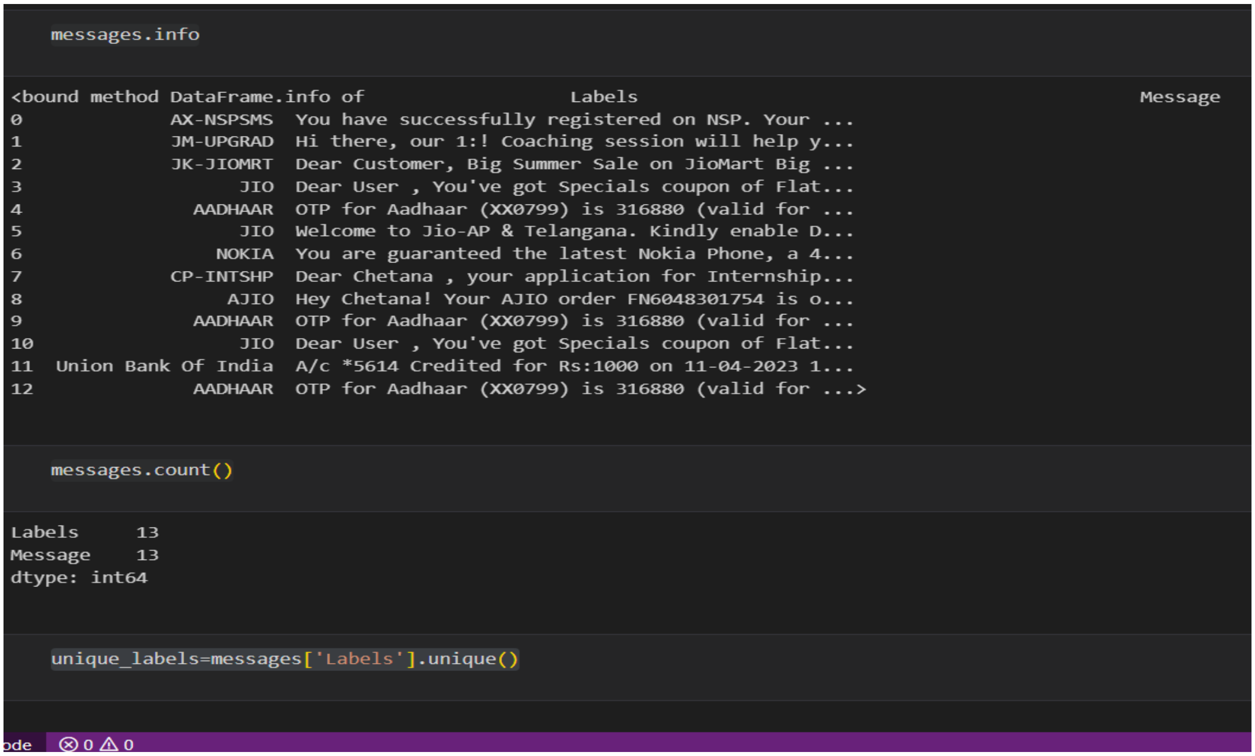
```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt
```

2. Collected data and stored in .csv file.

3. Loading dataset.

```
messages=pd.read_csv("SMS2.csv",encoding='latin-1')
```

4. Preprocessing dataset i.e, removal of null values.



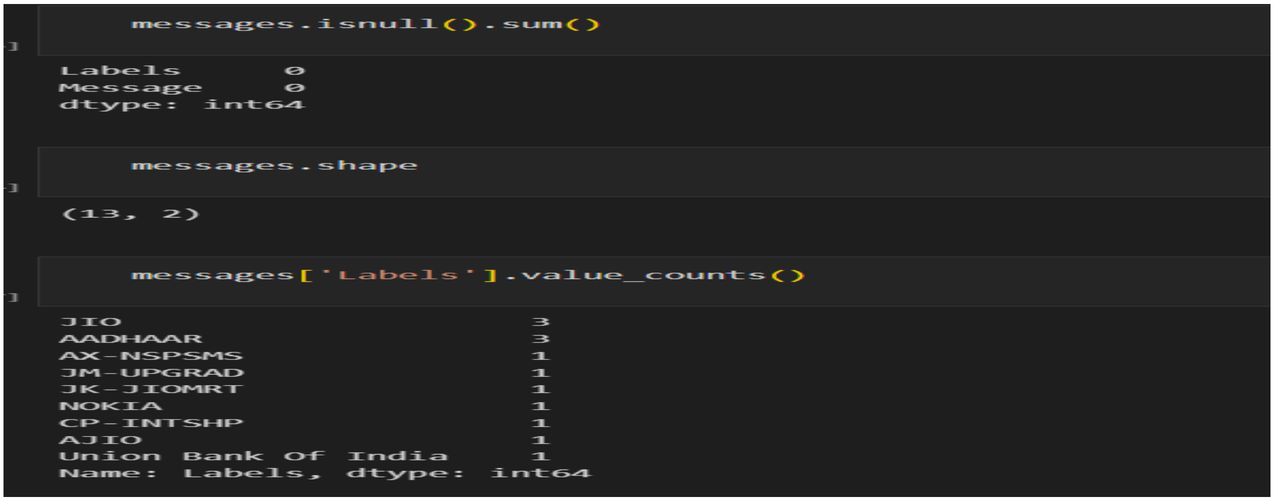
```
messages.info
```

```
<bound method DataFrame.info of                  Labels                                           Message
0            AX-NSPSMS  You have successfully registered on NSP. Your ...
1            JM-UPGRAD  Hi there, our 1:! Coaching session will help y...
2            JK-JIOMRT  Dear Customer, Big Summer Sale on JioMart Big ...
3                  JIO  Dear User , You've got Specials coupon of Flat...
4              AADHAAR  OTP for Aadhaar (XX0799) is 316880 (valid for ...
5                  JIO  Welcome to Jio-AP & Telangana. Kindly enable D...
6                NOKIA  You are guaranteed the latest Nokia Phone, a 4...
7            CP-INTSHP  Dear Chetana , your application for Internship...
8                 AJIO  Hey Chetana! Your AJIO order FN6048301754 is o...
9              AADHAAR  OTP for Aadhaar (XX0799) is 316880 (valid for ...
10                 JIO  Dear User , You've got Specials coupon of Flat...
11  Union Bank Of India  A/c *5614 Credited for Rs:1000 on 11-04-2023 1...
12             AADHAAR  OTP for Aadhaar (XX0799) is 316880 (valid for ...>
```

```
messages.count()
```

```
Labels     13
Message    13
dtype: int64
```

```
unique_labels=messages['Labels'].unique()
```



```
messages.isnull().sum()
```

```
Labels      0
Message     0
dtype: int64
```

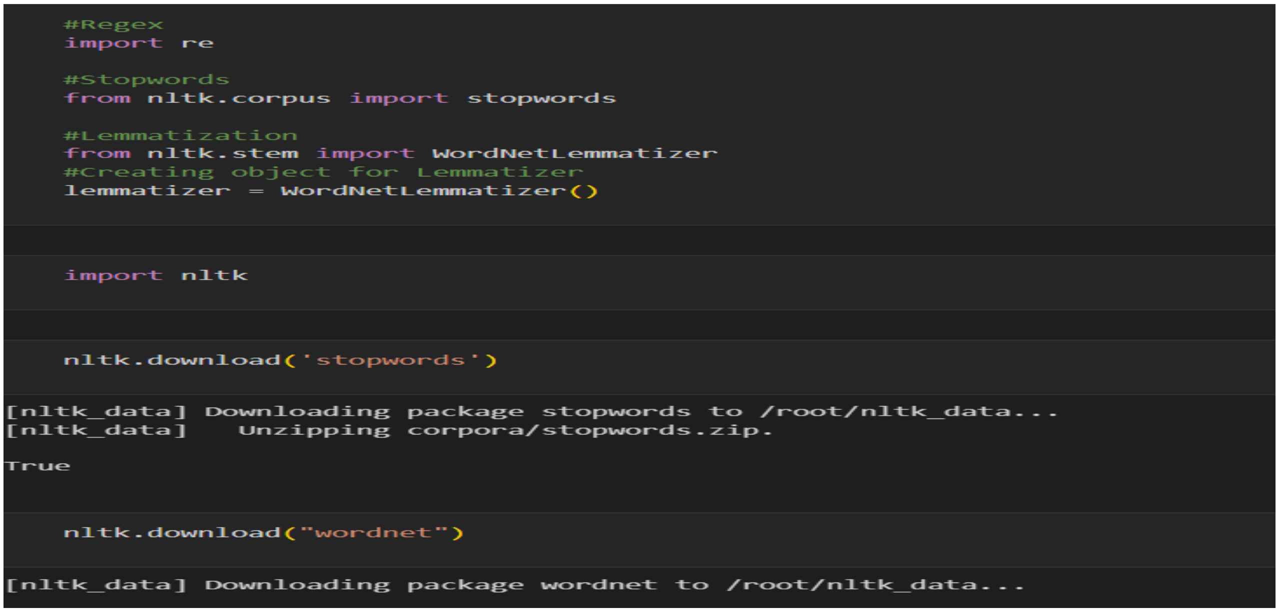
```
messages.shape
```

```
(13, 2)
```

```
messages['Labels'].value_counts()
```

```
JIO                     3
AADHAAR                 3
AX-NSPSMS               1
JM-UPGRAD               1
JK-JIOMRT               1
NOKIA                   1
CP-INTSHP               1
AJIO                    1
Union Bank Of India     1
Name: Labels, dtype: int64
```

5. Extracting stopwords and applied lemmatizers in sentence.



```
#Regex
import re

#Stopwords
from nltk.corpus import stopwords

#Lemmatization
from nltk.stem import WordNetLemmatizer
#Creating object for Lemmatizer
lemmatizer = WordNetLemmatizer()
```

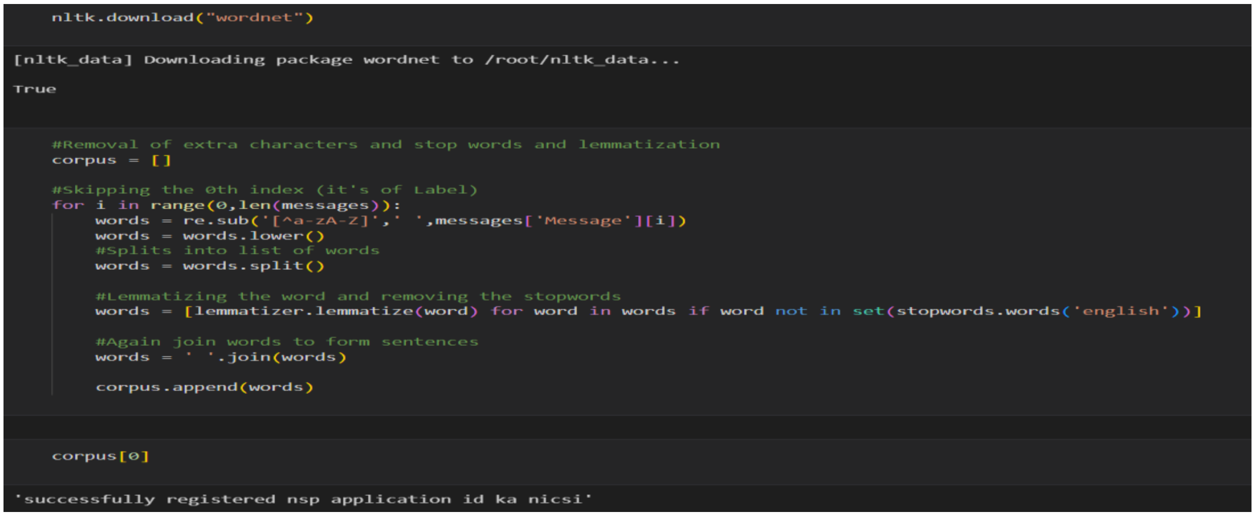
```
import nltk
```

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.

True
```

```
nltk.download("wordnet")
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
```



```
nltk.download("wordnet")
```

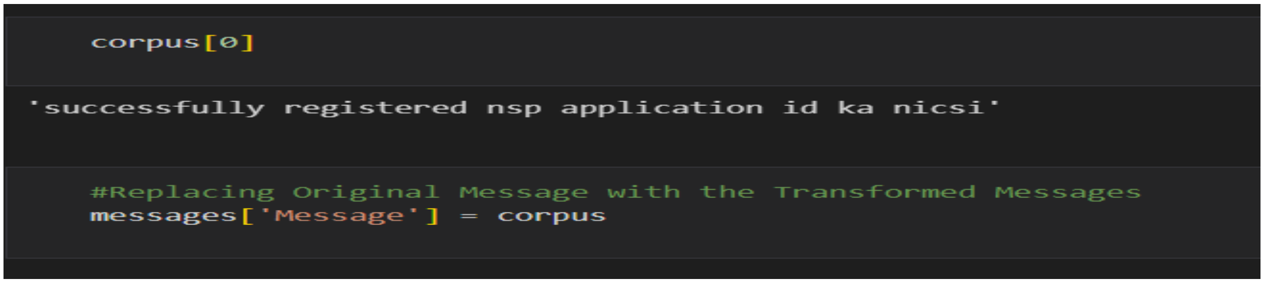
```
[nltk_data] Downloading package wordnet to /root/nltk_data...

True
```

```
#Removal of extra characters and stop words and lemmatization
corpus = []

#Skipping the 0th index (it's of Label)
for i in range(0,len(messages)):
    words = re.sub('[^a-zA-Z]',' ',messages['Message'][i])
    words = words.lower()
    #Splits into list of words
    words = words.split()

    #Lemmatizing the word and removing the stopwords
    words = [lemmatizer.lemmatize(word) for word in words if word not in set(stopwords.words('english'))]

    #Again join words to form sentences
    words = ' '.join(words)

    corpus.append(words)
```

```
corpus[0]
```

```
'successfully registered nsp application id ka nicsi'
```

6. Checking the corpus.

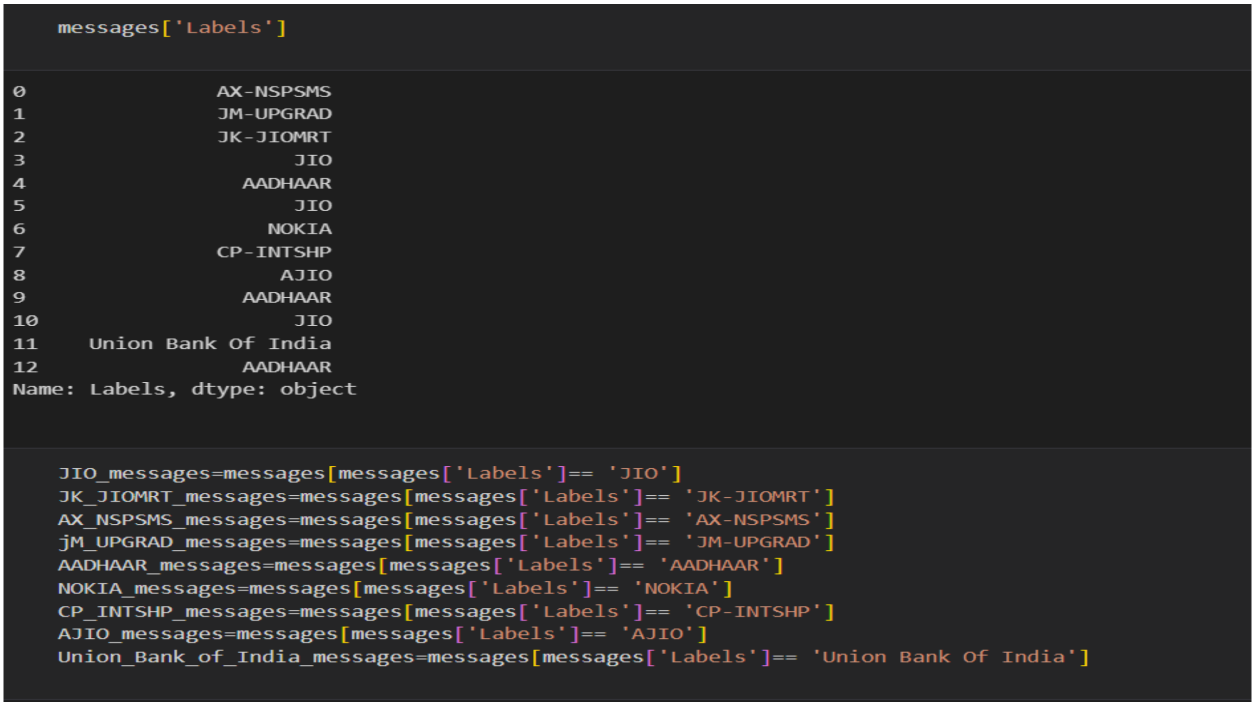


```
corpus[0]
```

```
'successfully registered nsp application id ka nicsi'
```

```
#Replacing Original Message with the Transformed Messages
messages['Message'] = corpus
```

7. Assigning labels

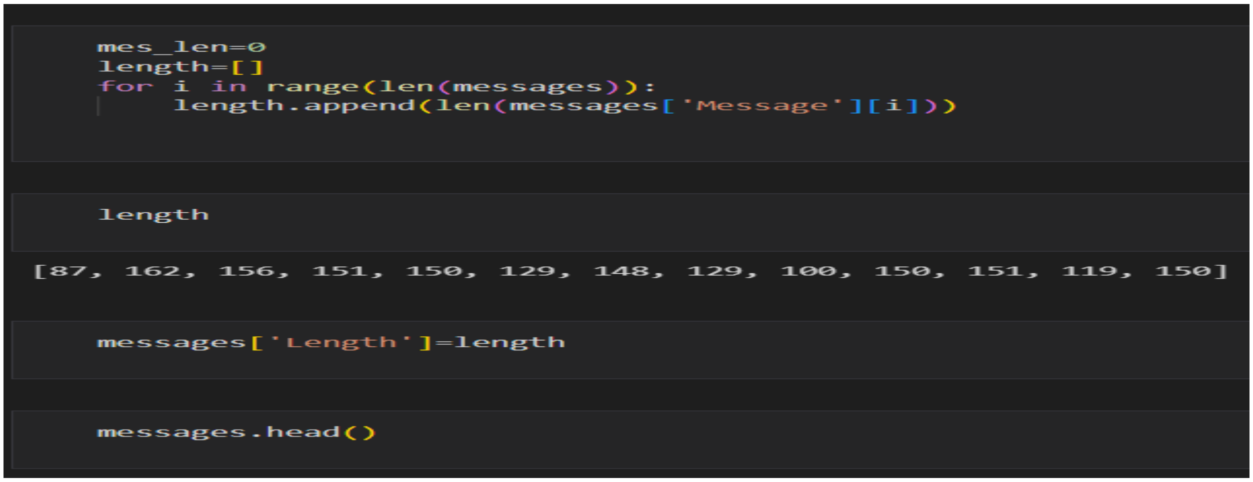


```
messages['Labels']
```

```
0             AX-NSPSMS
1             JM-UPGRAD
2             JK-JIOMRT
3                   JIO
4               AADHAAR
5                   JIO
6                 NOKIA
7             CP-INTSHP
8                  AJIO
9               AADHAAR
10                  JIO
11    Union Bank Of India
12              AADHAAR
Name: Labels, dtype: object
```

```
JIO_messages=messages[messages['Labels']== 'JIO']
JK_JIOMRT_messages=messages[messages['Labels']== 'JK-JIOMRT']
AX_NSPSMS_messages=messages[messages['Labels']== 'AX-NSPSMS']
jM_UPGRAD_messages=messages[messages['Labels']== 'JM-UPGRAD']
AADHAAR_messages=messages[messages['Labels']== 'AADHAAR']
NOKIA_messages=messages[messages['Labels']== 'NOKIA']
CP_INTSHP_messages=messages[messages['Labels']== 'CP-INTSHP']
AJIO_messages=messages[messages['Labels']== 'AJIO']
Union_Bank_of_India_messages=messages[messages['Labels']== 'Union Bank Of India']
```

8. Calculating length and punctuations in each text.



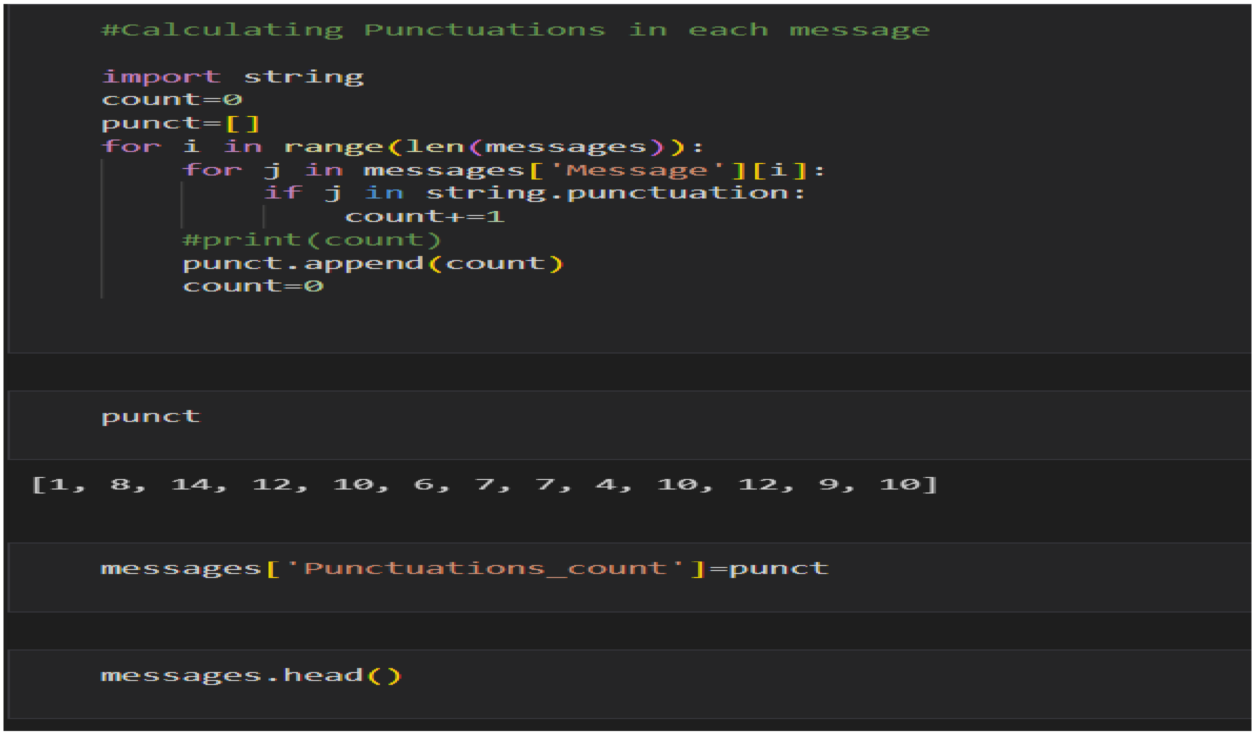
```
mes_len=0
length=[]
for i in range(len(messages)):
    length.append(len(messages['Message'][i]))
```

```
length
```

```
[87, 162, 156, 151, 150, 129, 148, 129, 100, 150, 151, 119, 150]
```

```
messages['Length']=length
```

```
messages.head()
```



```
#Calculating Punctuations in each message

import string
count=0
punct=[]
for i in range(len(messages)):
    for j in messages['Message'][i]:
        if j in string.punctuation:
            count+=1
    #print(count)
    punct.append(count)
    count=0
```

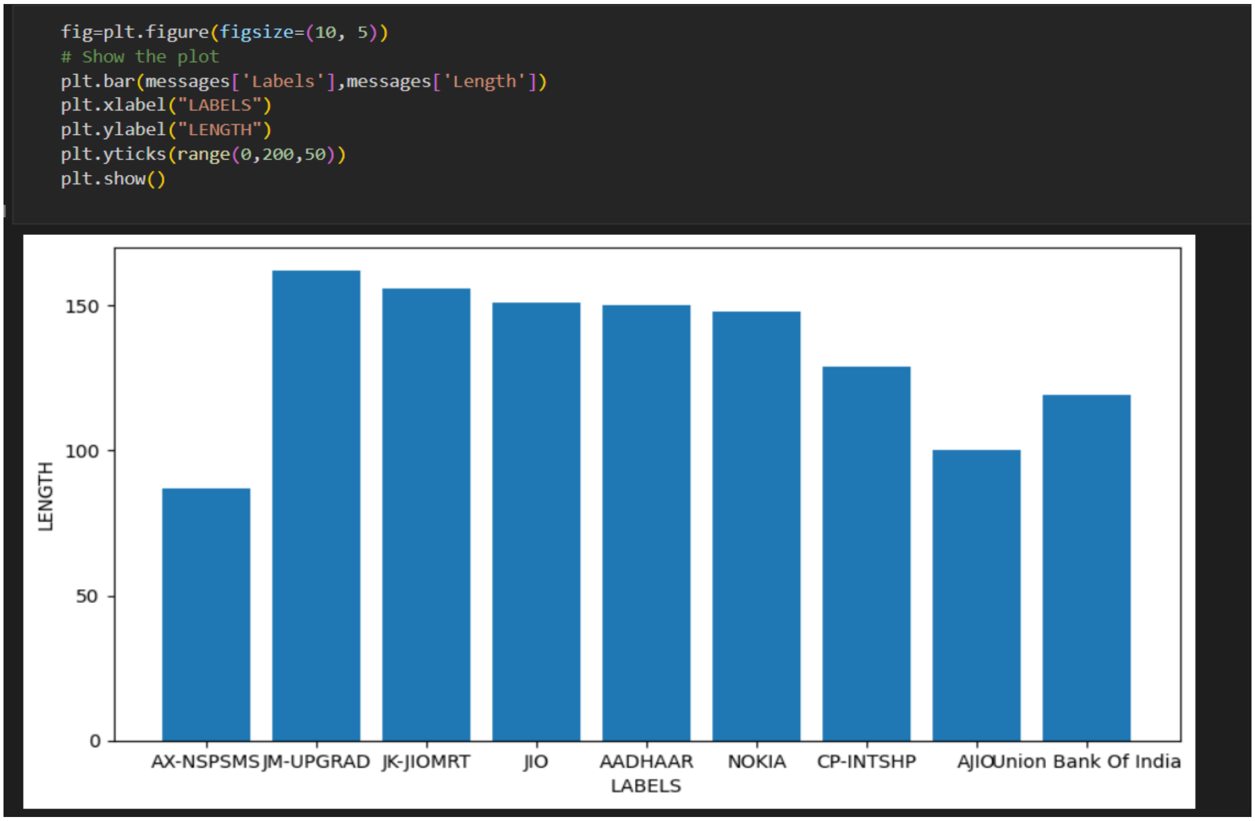
```
punct
```

```
[1, 8, 14, 12, 10, 6, 7, 7, 4, 10, 12, 9, 10]
```

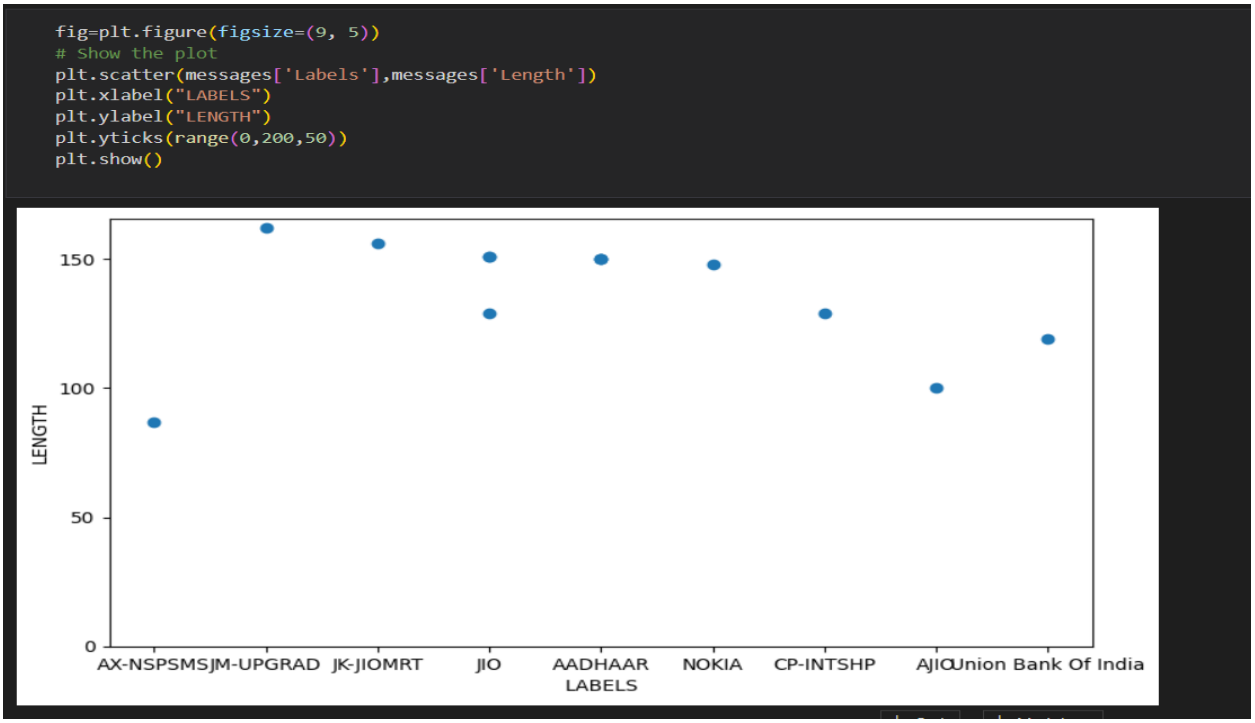
```
messages['Punctuations_count']=punct
```

```
messages.head()
```

9. We applied data visualization tool.

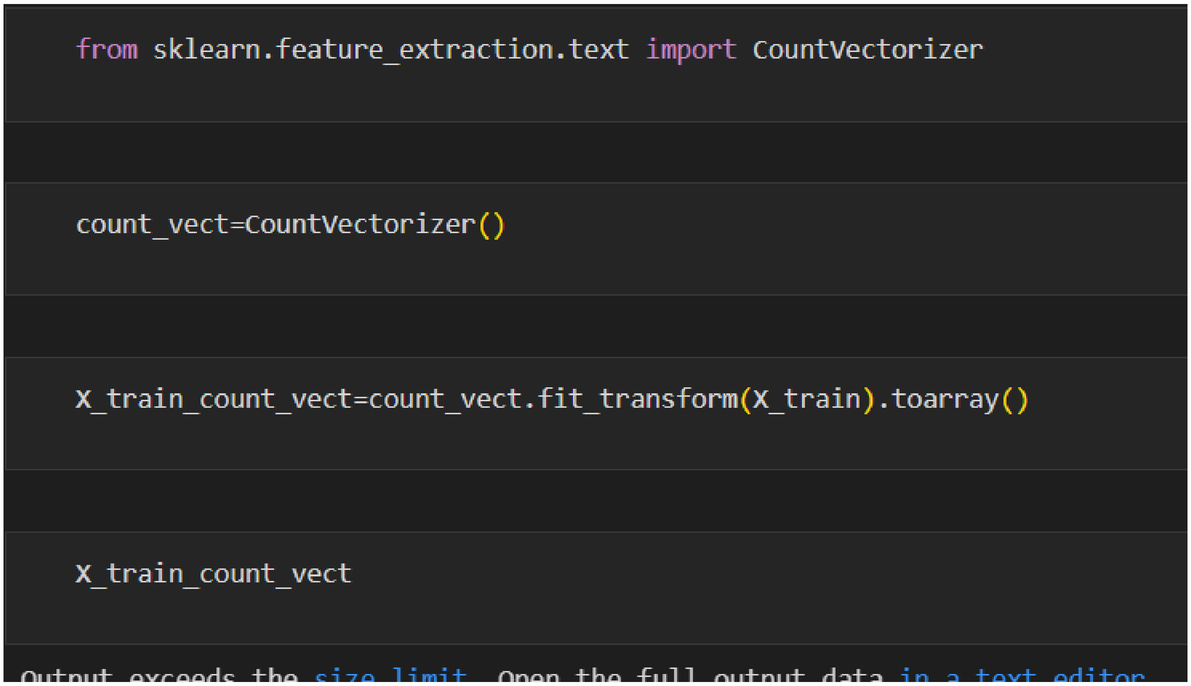


```
fig=plt.figure(figsize=(10, 5))
# Show the plot
plt.bar(messages['Labels'],messages['Length'])
plt.xlabel("LABELS")
plt.ylabel("LENGTH")
plt.yticks(range(0,200,50))
plt.show()
```



```
fig=plt.figure(figsize=(9, 5))
# Show the plot
plt.scatter(messages['Labels'],messages['Length'])
plt.xlabel("LABELS")
plt.ylabel("LENGTH")
plt.yticks(range(0,200,50))
plt.show()
```

10.Apply countvectorizer and tf-idf to calculate frequency of each term present in the document.



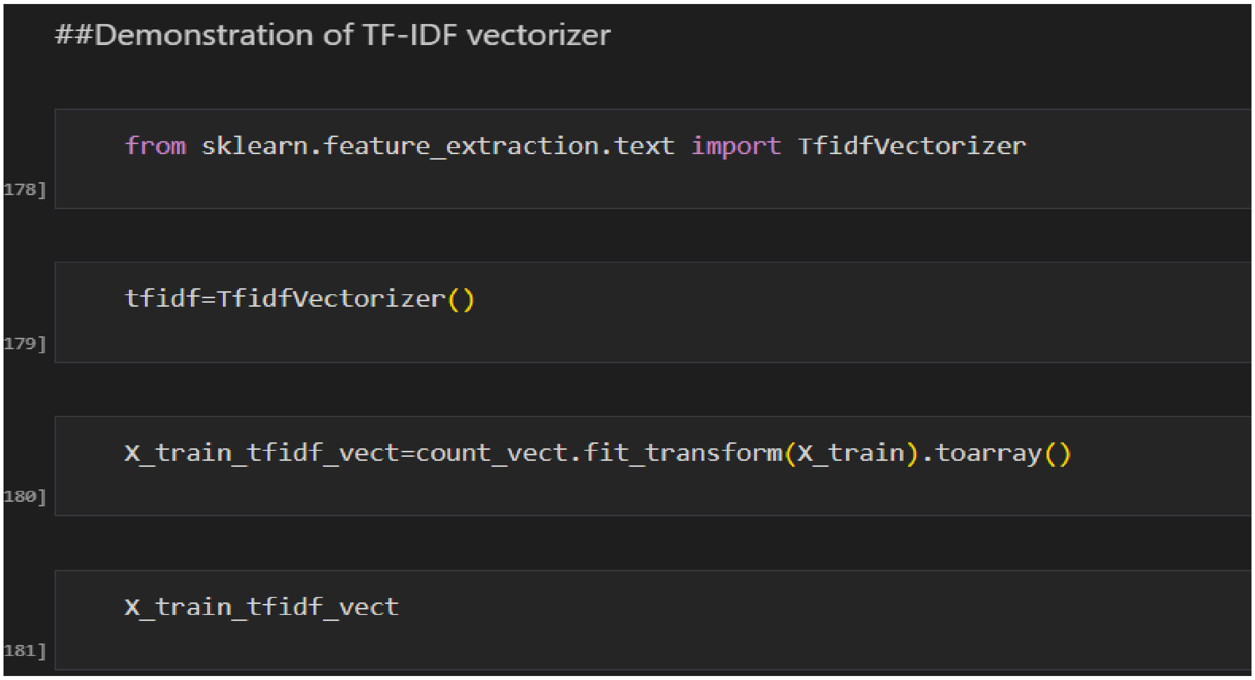
```
from sklearn.feature_extraction.text import CountVectorizer
```

```
count_vect=CountVectorizer()
```

```
X_train_count_vect=count_vect.fit_transform(X_train).toarray()
```

```
X_train_count_vect
```

```
Output exceeds the size limit. Open the full output data in a text editor
```



##Demonstration of TF-IDF vectorizer

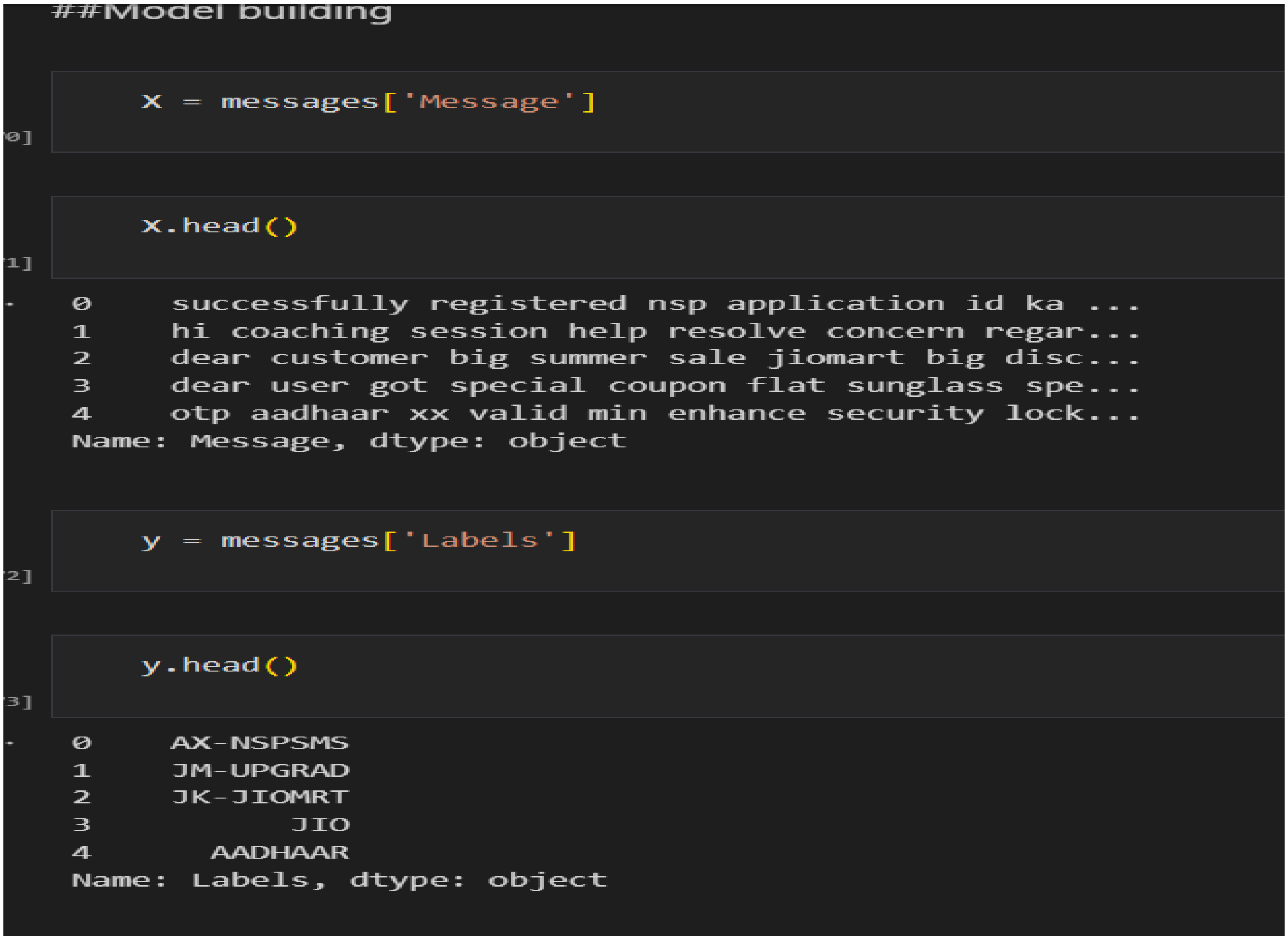
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
tfidf=TfidfVectorizer()
```

```
X_train_tfidf_vect=count_vect.fit_transform(X_train).toarray()
```

```
X_train_tfidf_vect
```

11.Building the model



##Model building

```
X = messages['Message']
```

```
X.head()
```

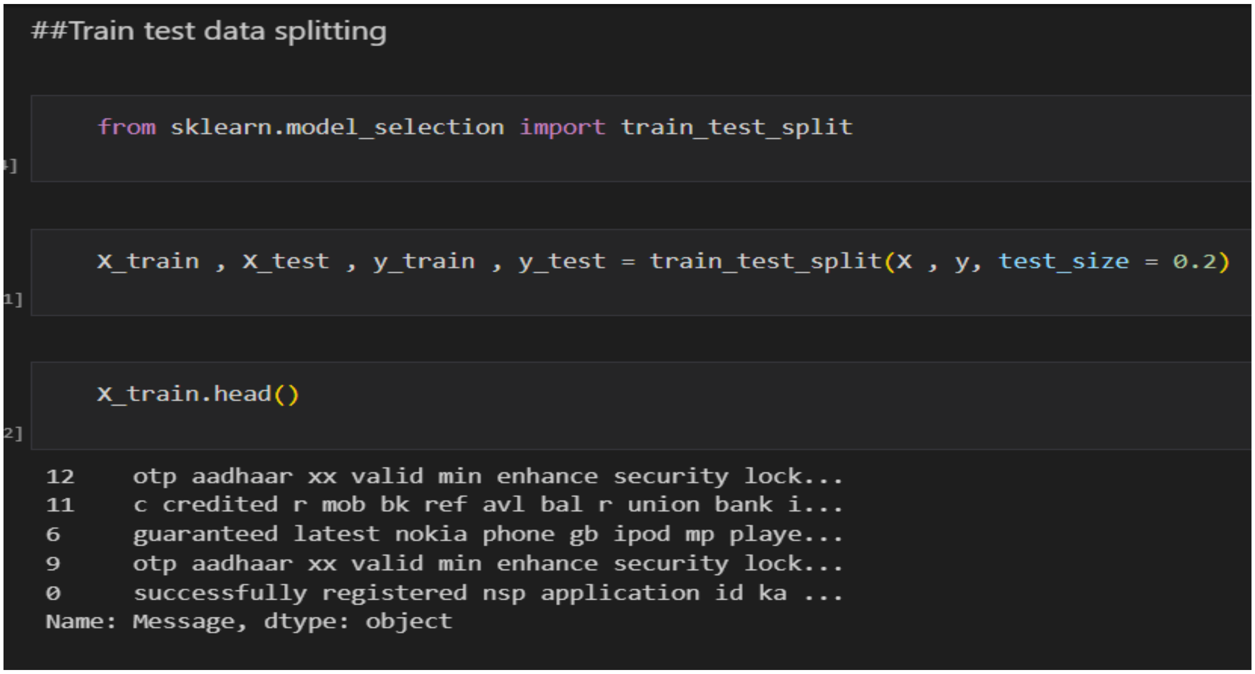
```
0    successfully registered nsp application id ka ...
1    hi coaching session help resolve concern regar...
2    dear customer big summer sale jiomart big disc...
3    dear user got special coupon flat sunglass spe...
4    otp aadhaar xx valid min enhance security lock...
Name: Message, dtype: object
```

```
y = messages['Labels']
```

```
y.head()
```

```
0    AX-NSPSMS
1    JM-UPGRAD
2    JK-JIOMRT
3          JIO
4      AADHAAR
Name: Labels, dtype: object
```

12.Splitting data into train and test data



##Train test data splitting

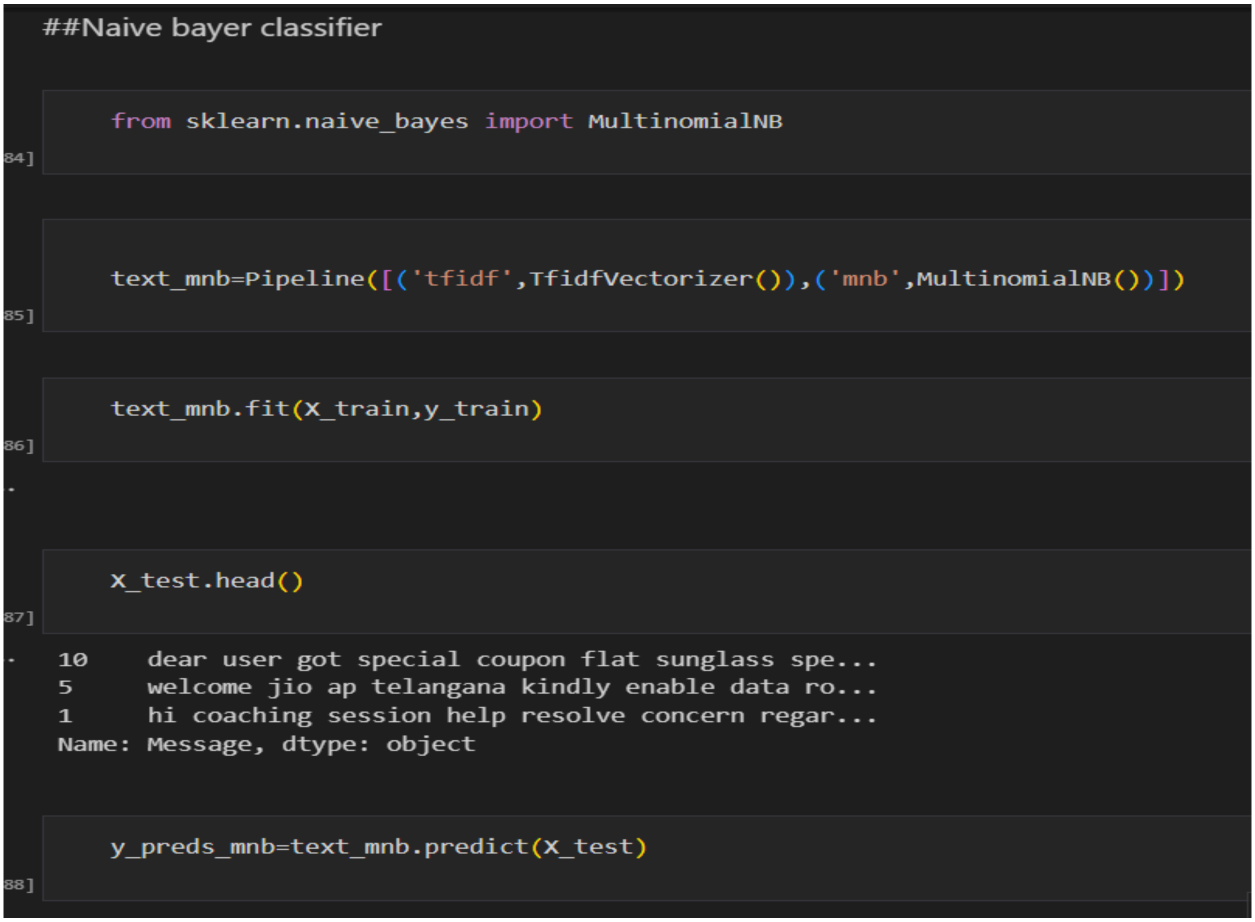
```
from sklearn.model_selection import train_test_split
```

```
X_train , X_test , y_train , y_test = train_test_split(X , y, test_size = 0.2)
```

```
X_train.head()
```

```
12    otp aadhaar xx valid min enhance security lock...
11    c credited r mob bk ref avl bal r union bank i...
6     guaranteed latest nokia phone gb ipod mp playe...
9     otp aadhaar xx valid min enhance security lock...
0     successfully registered nsp application id ka ...
Name: Message, dtype: object
```

13.We applied firstly Naive bayes algorithm it did not give accurate outputs.



##Naive bayer classifier

```
from sklearn.naive_bayes import MultinomialNB
```

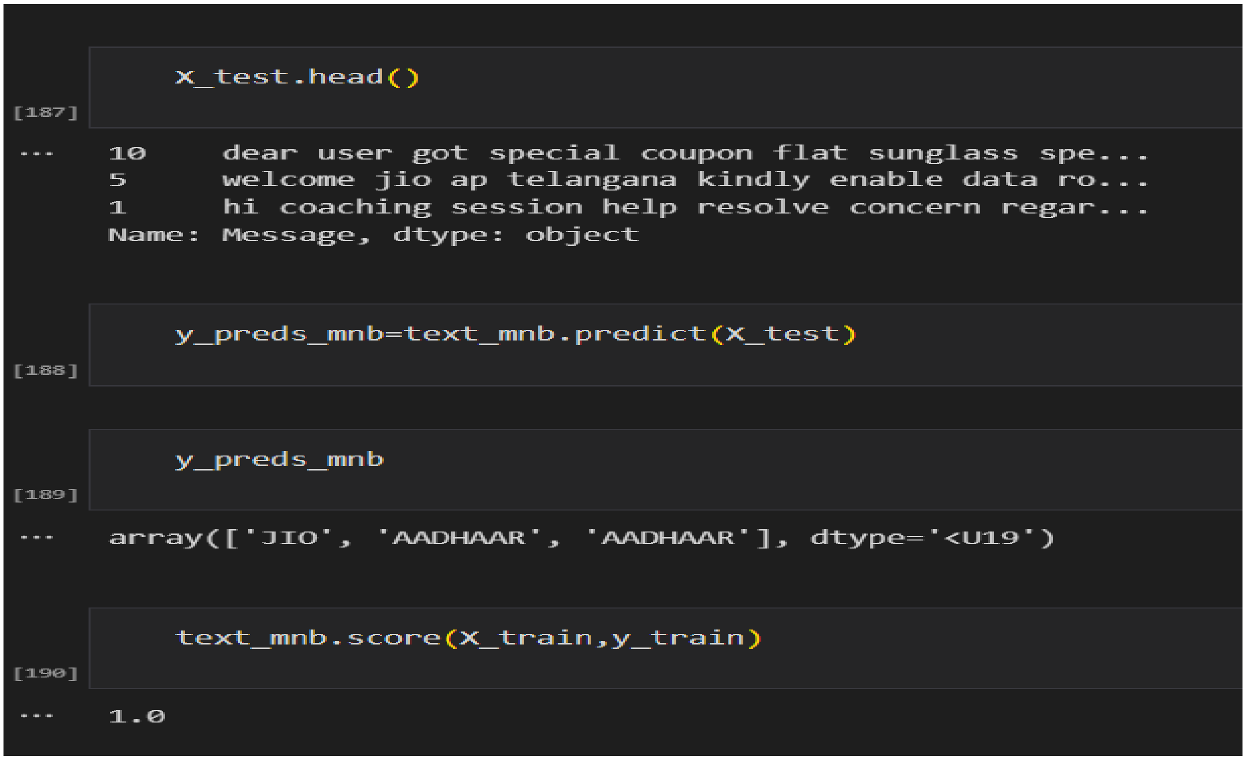
```
text_mnb=Pipeline([('tfidf',TfidfVectorizer()),('mnb',MultinomialNB())])
```

```
text_mnb.fit(X_train,y_train)
```

```
X_test.head()
```

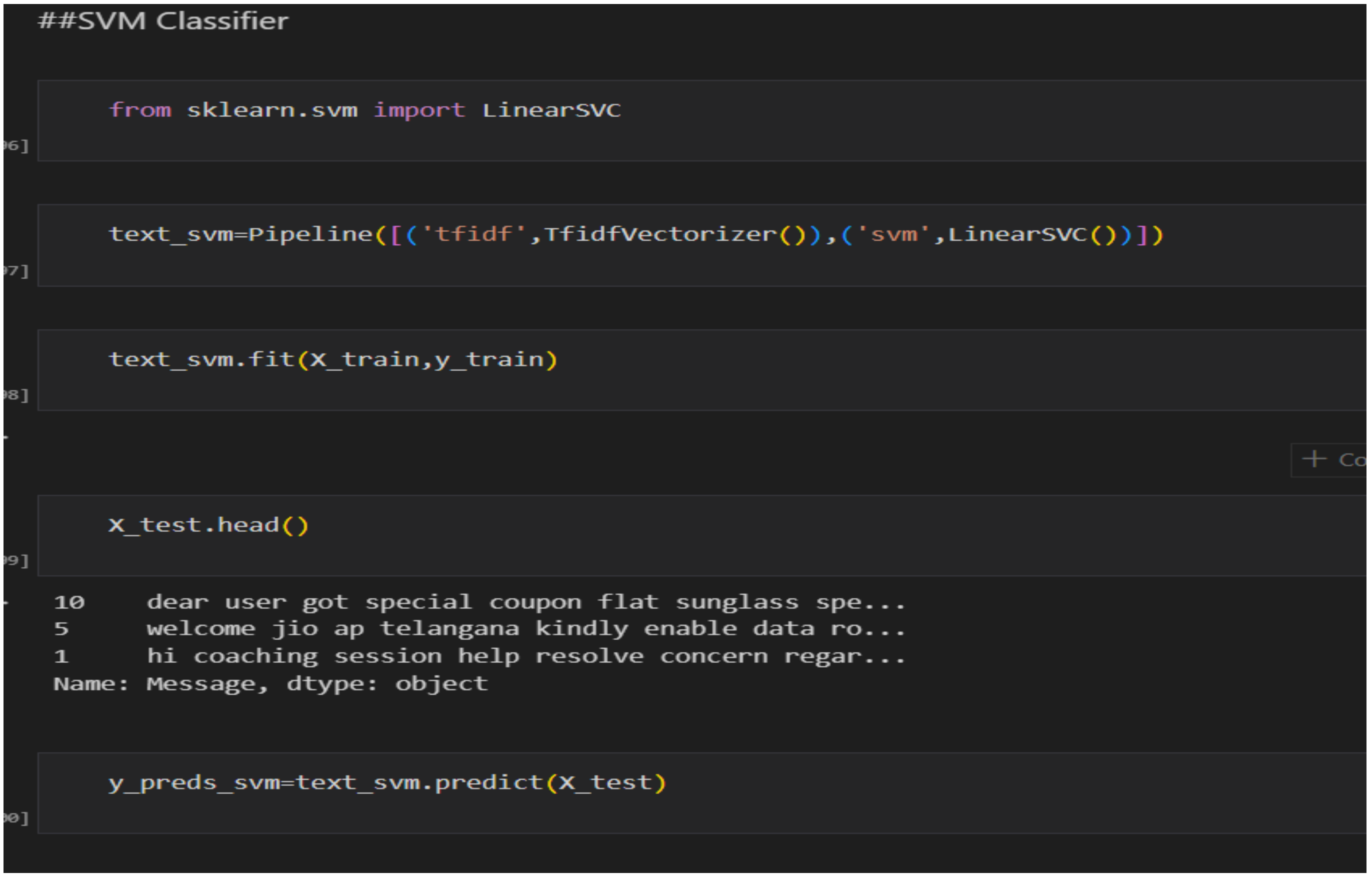
```
10    dear user got special coupon flat sunglass spe...
5     welcome jio ap telangana kindly enable data ro...
1     hi coaching session help resolve concern regar...
Name: Message, dtype: object
```

```
y_preds_mnb=text_mnb.predict(X_test)
```



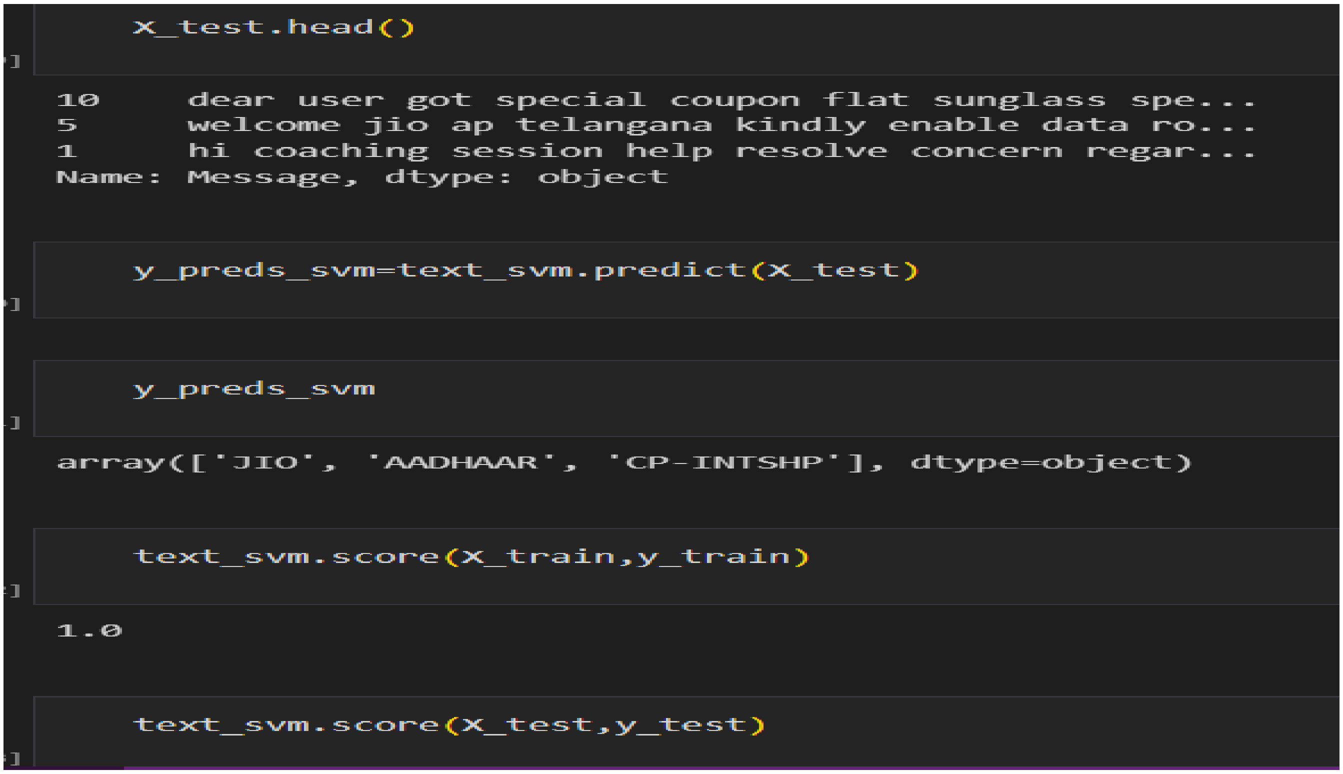
```
[187]     X_test.head()

...   10     dear user got special coupon flat sunglass spe...
      5      welcome jio ap telangana kindly enable data ro...
      1      hi coaching session help resolve concern regar...
      Name: Message, dtype: object

[188]     y_preds_mnb=text_mnb.predict(X_test)

[189]     y_preds_mnb

...   array(['JIO', 'AADHAAR', 'AADHAAR'], dtype='<U19')

[190]     text_mnb.score(X_train,y_train)

...   1.0
```

14.We used linearSVC algorithm and predicted output is correct.



##SVM Classifier

```
6]     from sklearn.svm import LinearSVC

7]     text_svm=Pipeline([('tfidf',TfidfVectorizer()),('svm',LinearSVC())])

8]     text_svm.fit(X_train,y_train)

9]     X_test.head()

   10     dear user got special coupon flat sunglass spe...
   5      welcome jio ap telangana kindly enable data ro...
   1      hi coaching session help resolve concern regar...
   Name: Message, dtype: object

0]     y_preds_svm=text_svm.predict(X_test)
```



```
X_test.head()
```

```
10     dear user got special coupon flat sunglass spe...
5      welcome jio ap telangana kindly enable data ro...
1      hi coaching session help resolve concern regar...
Name: Message, dtype: object
```

```
y_preds_svm=text_svm.predict(X_test)
```

```
y_preds_svm
```

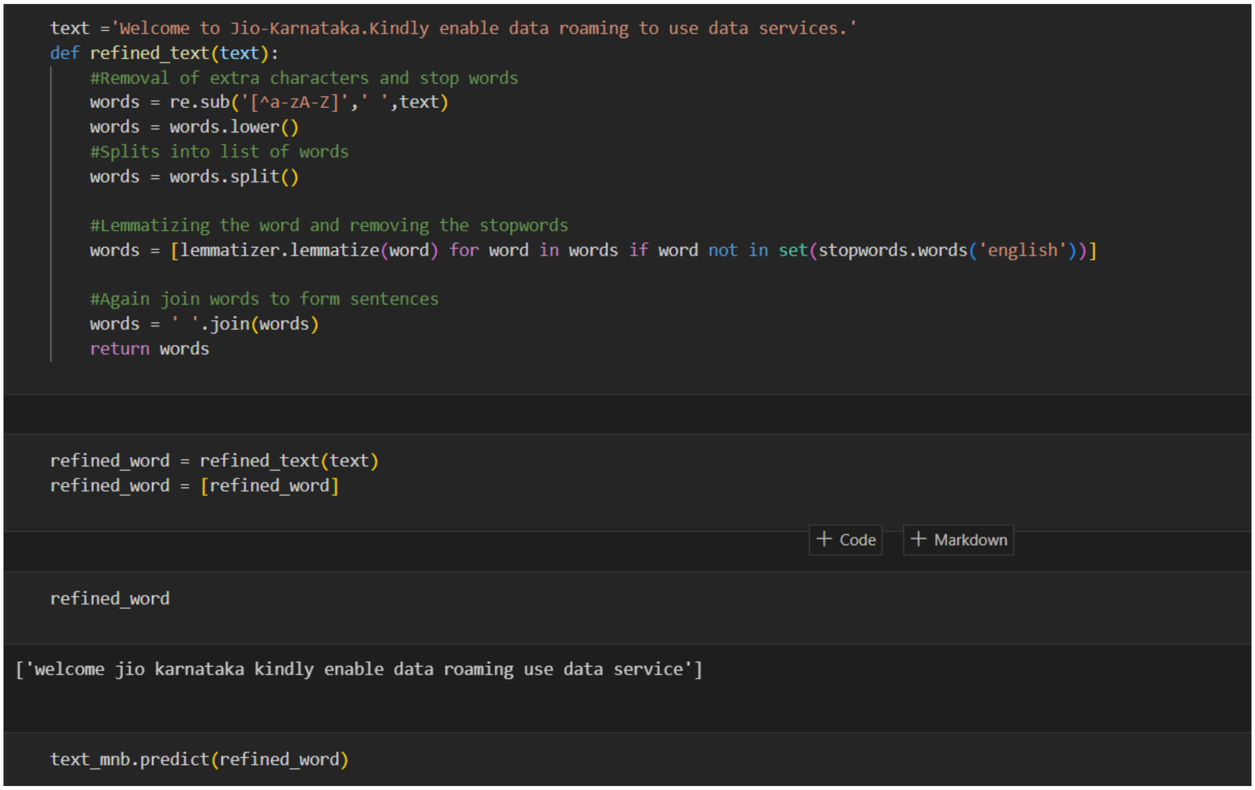
```
array(['JIO', 'AADHAAR', 'CP-INTSHP'], dtype=object)
```

```
text_svm.score(X_train,y_train)
```

```
1.0
```

```
text_svm.score(X_test,y_test)
```

15. Predicting the message label



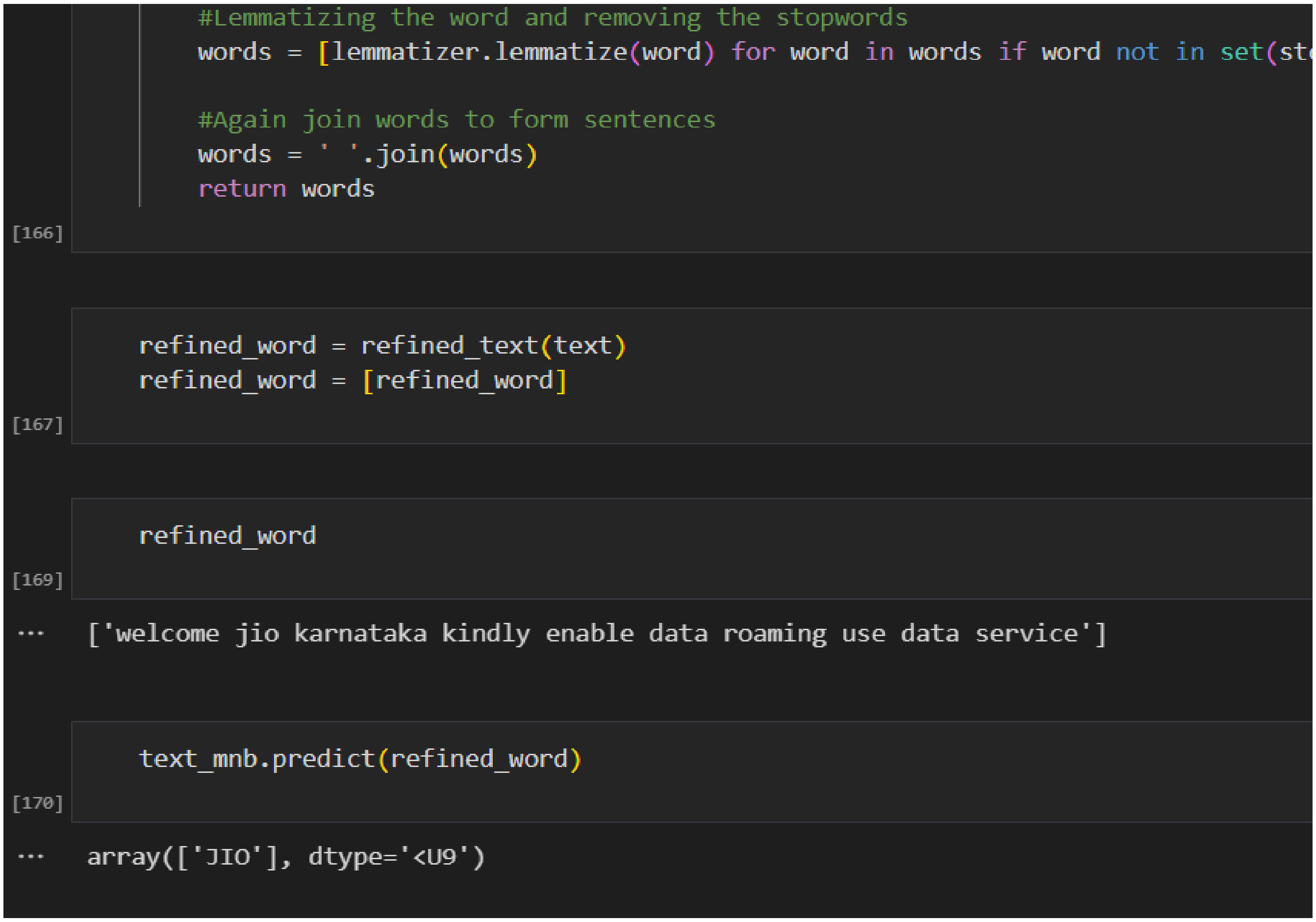
```
text ='Welcome to Jio-Karnataka.Kindly enable data roaming to use data services.'
def refined_text(text):
    #Removal of extra characters and stop words
    words = re.sub('[^a-zA-Z]',' ',text)
    words = words.lower()
    #Splits into list of words
    words = words.split()

    #Lemmatizing the word and removing the stopwords
    words = [lemmatizer.lemmatize(word) for word in words if word not in set(stopwords.words('english'))]

    #Again join words to form sentences
    words = ' '.join(words)
    return words
```

```
refined_word = refined_text(text)
refined_word = [refined_word]
```

```
refined_word
```

```
['welcome jio karnataka kindly enable data roaming use data service']
```

```
text_mnb.predict(refined_word)
```



```
        #Lemmatizing the word and removing the stopwords
        words = [lemmatizer.lemmatize(word) for word in words if word not in set(st

        #Again join words to form sentences
        words = ' '.join(words)
        return words
[166]
```

```
refined_word = refined_text(text)
refined_word = [refined_word]
[167]
```

```
refined_word
[169]
```

```
['welcome jio karnataka kindly enable data roaming use data service']
```

```
text_mnb.predict(refined_word)
[170]
```

```
array(['JIO'], dtype='<U9')
```