# EXPERIMENT-5

## Problem Statement

Use any machine learning method to classify the email dataset

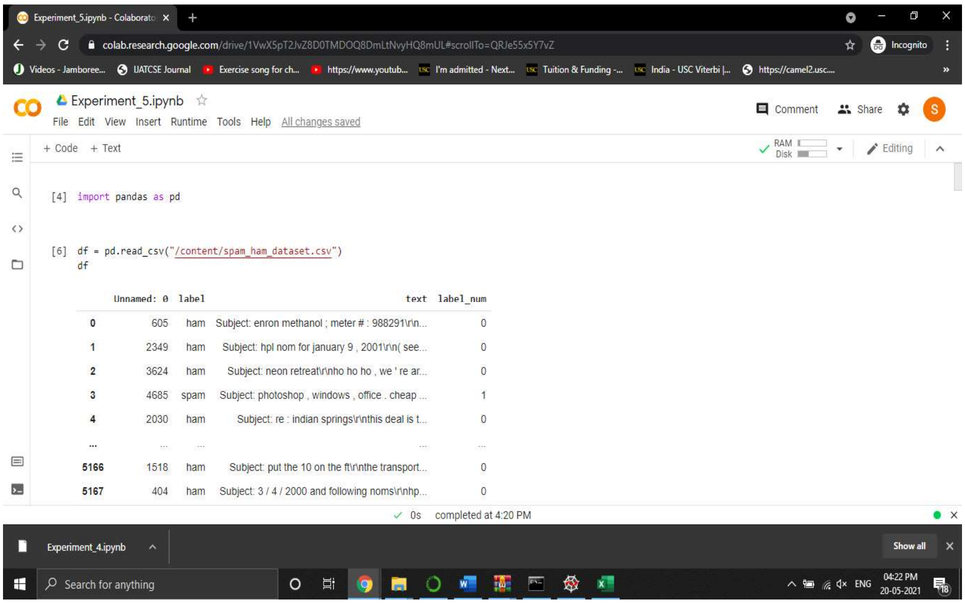
## Algorithm

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Naive Bayes classifier calculates the probability of an event in the following steps:
**Step 1:** Calculate the prior probability for given class labels
**Step 2:** Find Likelihood probability with each attribute for each class
**Step 3:** Put these values in Bayes Formula and calculate posterior probability.
**Step 4:** See which class has a higher probability, given the input belongs to the higher probability class.

## Program Snippet

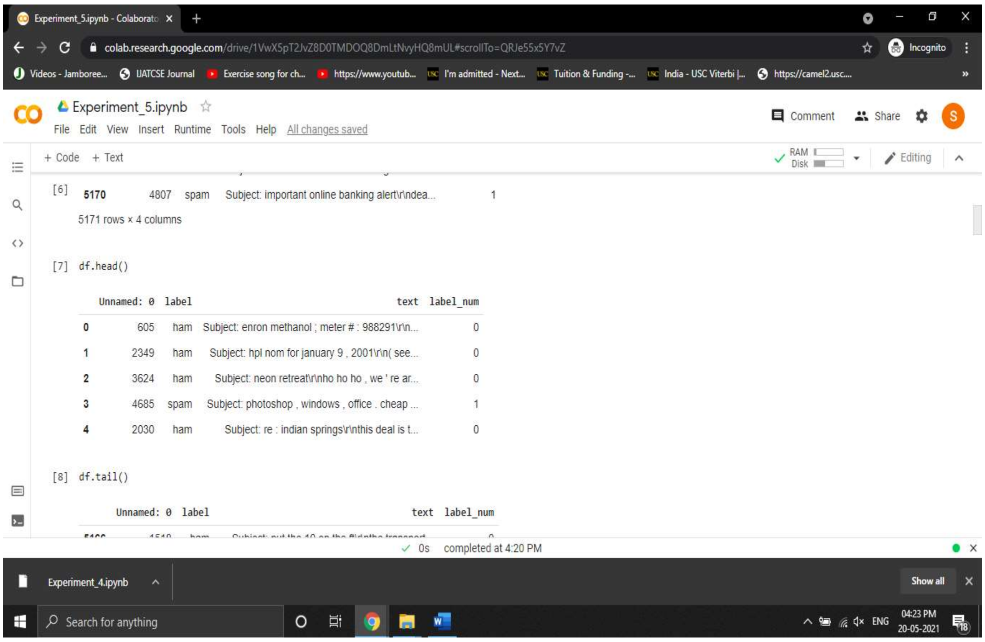
## Reading CSV



```
[4] import pandas as pd
```

```
[6] df = pd.read_csv("/content/spam_ham_dataset.csv")
    df
```

```
      Unnamed: 0  label                                             text  label_num
0            605    ham  Subject: enron methanol ; meter # : 988291\r\n...          0
1           2349    ham   Subject: hpl nom for january 9 , 2001\r\n( see...          0
2           3624    ham    Subject: neon retreat\r\nho ho ho , we ' re ar...          0
3           4685   spam   Subject: photoshop , windows , office . cheap ...          1
4           2030    ham      Subject: re : indian springs\r\nthis deal is t...          0
...          ...    ...                                              ...        ...
5166        1518    ham    Subject: put the 10 on the ft\r\nthe transport...          0
5167         404    ham  Subject: 3 / 4 / 2000 and following noms\r\nhp...          0
```



[6]

| | | | | |
|---|---|---|---|---|
| 5170 | 4807 | spam | Subject: important online banking alert\r\ndea... | 1 |

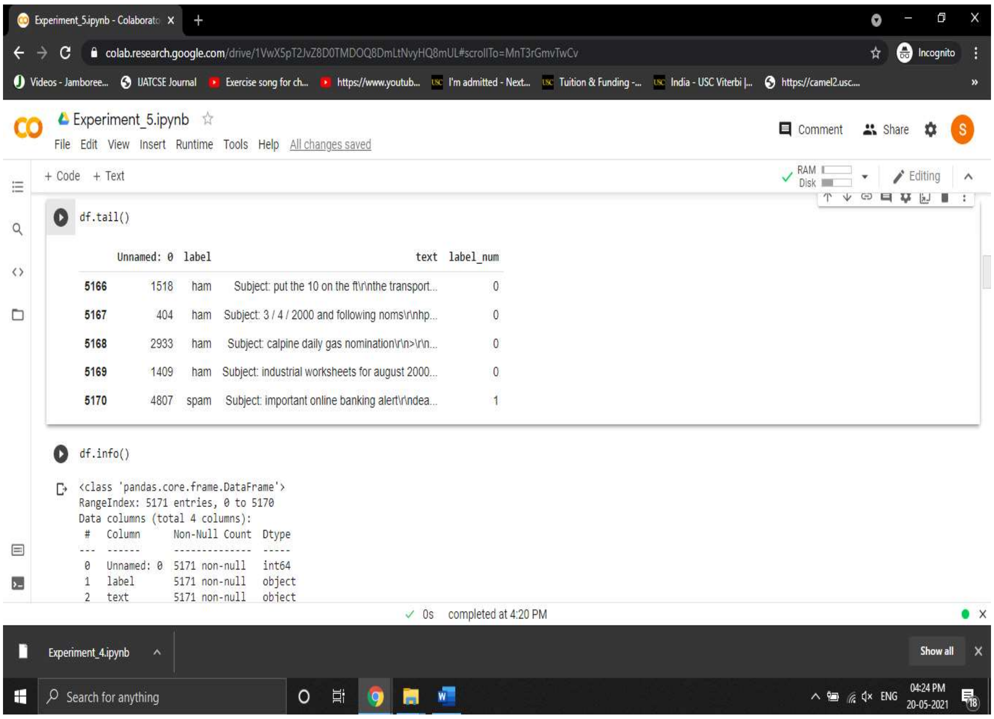
5171 rows × 4 columns

[7] `df.head()`

| | Unnamed: 0 | label | text | label_num |
|---|---|---|---|---|
| 0 | 605 | ham | Subject: enron methanol ; meter # : 988291\r\n... | 0 |
| 1 | 2349 | ham | Subject: hpl nom for january 9 , 2001\r\n( see... | 0 |
| 2 | 3624 | ham | Subject: neon retreat\r\nho ho ho , we ' re ar... | 0 |
| 3 | 4685 | spam | Subject: photoshop , windows , office . cheap ... | 1 |
| 4 | 2030 | ham | Subject: re : indian springs\r\nthis deal is t... | 0 |

[8] `df.tail()`

| | Unnamed: 0 | label | text | label_num |
|---|---|---|---|---|

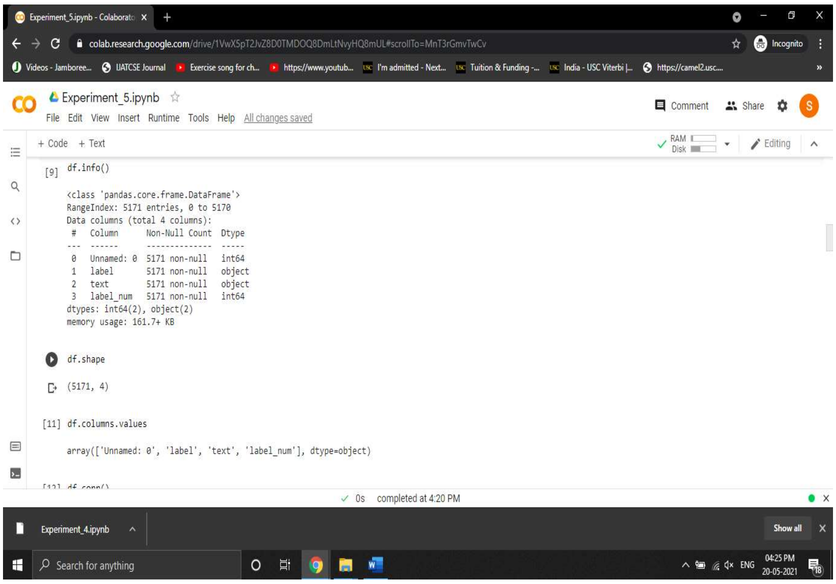


`df.tail()`

| | Unnamed: 0 | label | text | label_num |
|---|---|---|---|---|
| 5166 | 1518 | ham | Subject: put the 10 on the ft\r\nthe transport... | 0 |
| 5167 | 404 | ham | Subject: 3 / 4 / 2000 and following noms\r\nhp... | 0 |
| 5168 | 2933 | ham | Subject: calpine daily gas nomination\r\n>\r\n... | 0 |
| 5169 | 1409 | ham | Subject: industrial worksheets for august 2000... | 0 |
| 5170 | 4807 | spam | Subject: important online banking alert\r\ndea... | 1 |

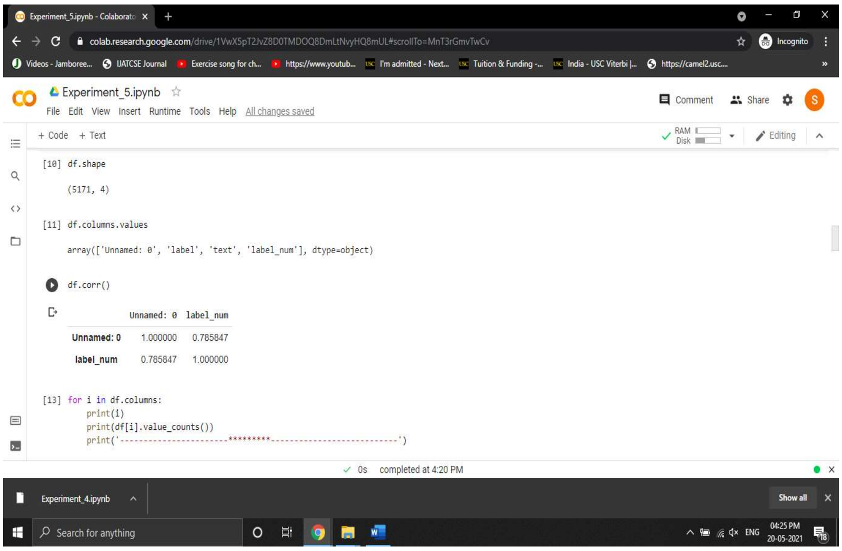
`df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5171 entries, 0 to 5170
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  5171 non-null   int64 
 1   label       5171 non-null   object
 2   text        5171 non-null   object
```

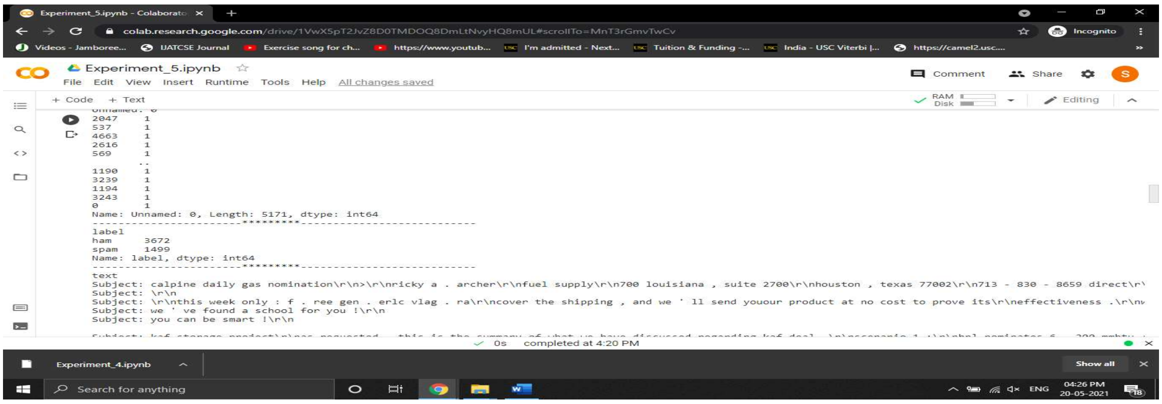


```
[9] df.info()

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 5171 entries, 0 to 5170
    Data columns (total 4 columns):
     #   Column      Non-Null Count  Dtype
    ---  ------      --------------  -----
     0   Unnamed: 0  5171 non-null   int64
     1   label       5171 non-null   object
     2   text        5171 non-null   object
     3   label_num   5171 non-null   int64
    dtypes: int64(2), object(2)
    memory usage: 161.7+ KB

    df.shape

    (5171, 4)

[11] df.columns.values

    array(['Unnamed: 0', 'label', 'text', 'label_num'], dtype=object)
```

# Estimating the Relation

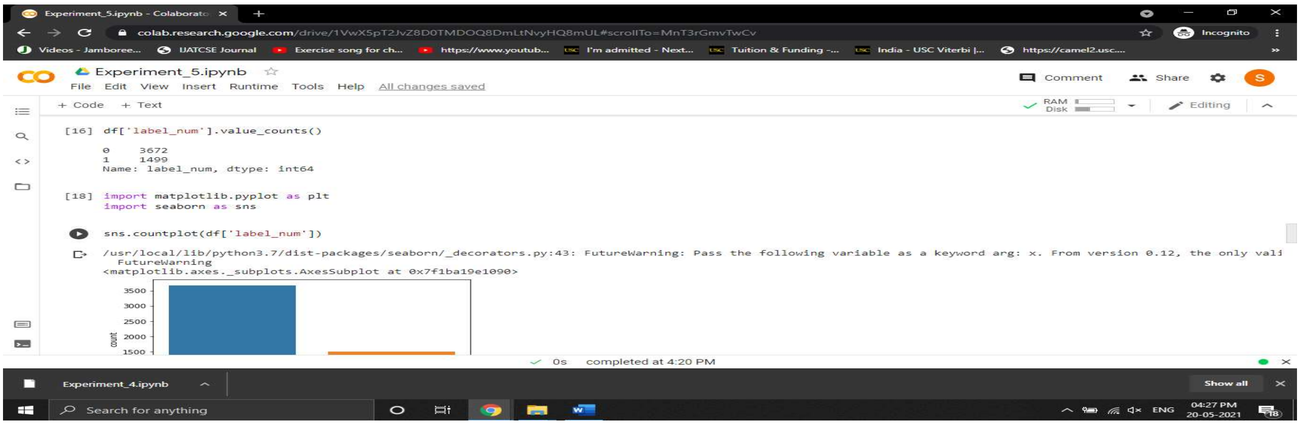


```
[10] df.shape

    (5171, 4)

[11] df.columns.values

    array(['Unnamed: 0', 'label', 'text', 'label_num'], dtype=object)

    df.corr()

                Unnamed: 0  label_num
    Unnamed: 0    1.000000   0.785847
     label_num    0.785847   1.000000

[13] for i in df.columns:
       print(i)
       print(df[i].value_counts())
       print('----------------------*********--------------------------')
```

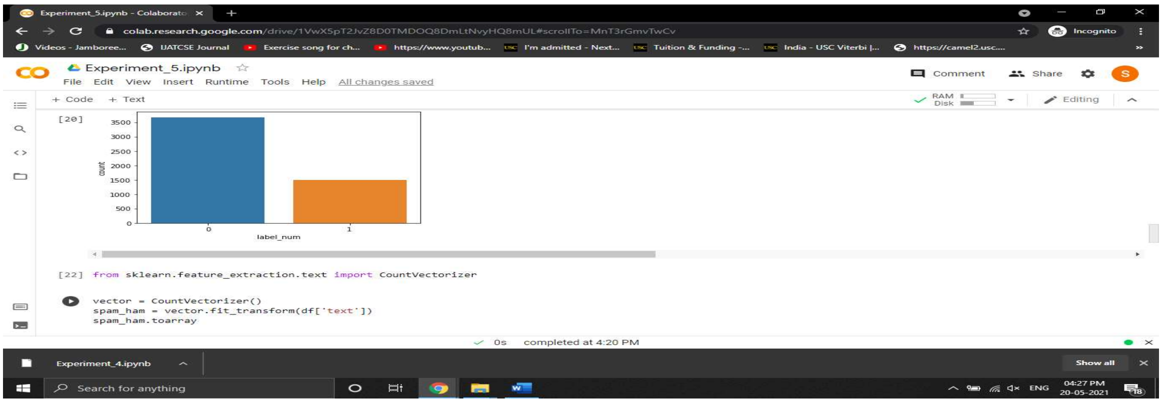


```
2047    1
537     1
4663    1
2616    1
569     1
        ..
1190    1
3239    1
1194    1
3243    1
0       1
Name: Unnamed: 0, Length: 5171, dtype: int64
----------------------*********--------------------------
label
ham     3672
spam    1499
Name: label, dtype: int64
----------------------*********--------------------------
text
Subject: calpine daily gas nomination\r\n>\r\nricky a . archer\r\nfuel supply\r\n700 louisiana , suite 2700\r\nhouston , texas 77002\r\n713 - 830 - 8659 direct\r\
Subject: \r\n
Subject: \r\nthis week only : f . ree gen . erlc vlag . ra\r\ncover the shipping , and we ' ll send youour product at no cost to prove its\r\neffectiveness .\r\n
Subject: we ' ve found a school for you !\r\n
Subject: you can be smart !\r\n
```

# Visualization

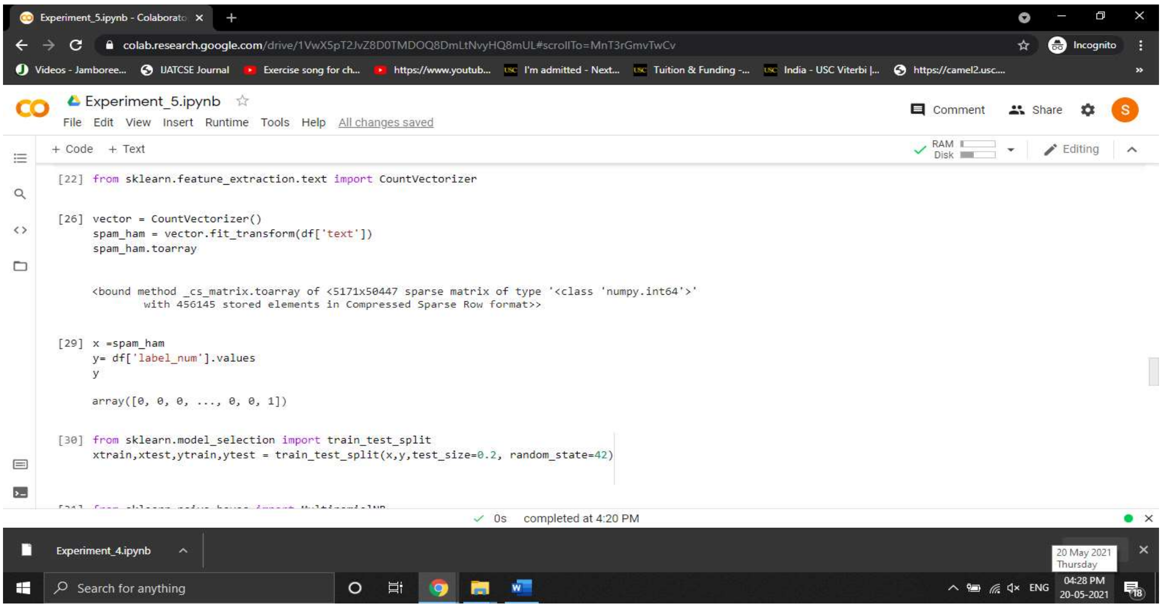


```
[16] df['label_num'].value_counts()

     0    3672
     1    1499
     Name: label_num, dtype: int64

[18] import matplotlib.pyplot as plt
     import seaborn as sns

     sns.countplot(df['label_num'])

     /usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only vali
       FutureWarning
     <matplotlib.axes._subplots.AxesSubplot at 0x7f1ba19e1090>
```

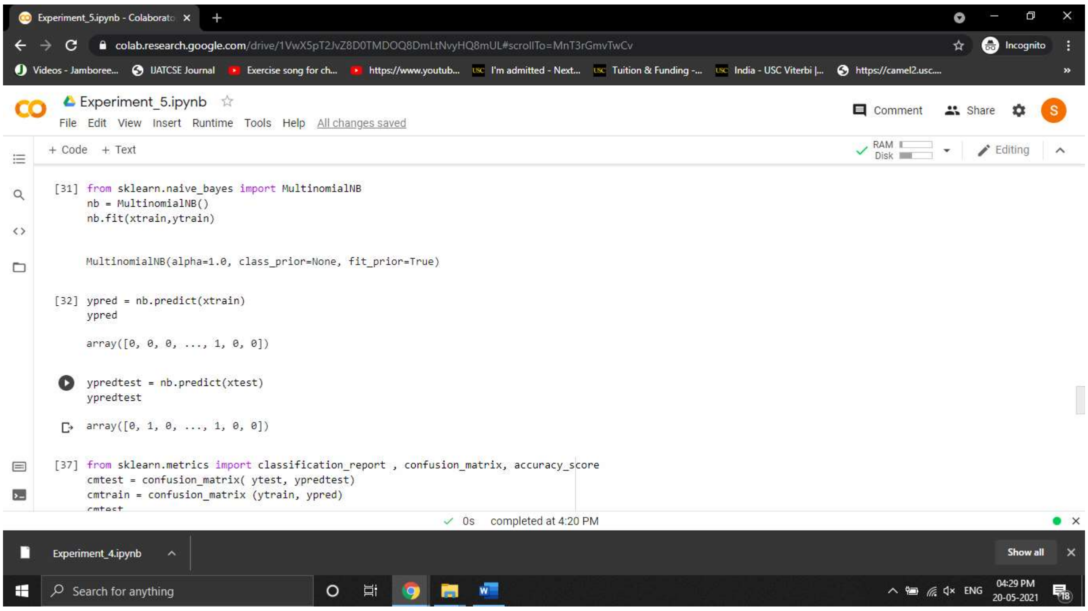


```
[20]

[22] from sklearn.feature_extraction.text import CountVectorizer

     vector = CountVectorizer()
     spam_ham = vector.fit_transform(df['text'])
     spam_ham.toarray
```

# Training and Testing the Dataset

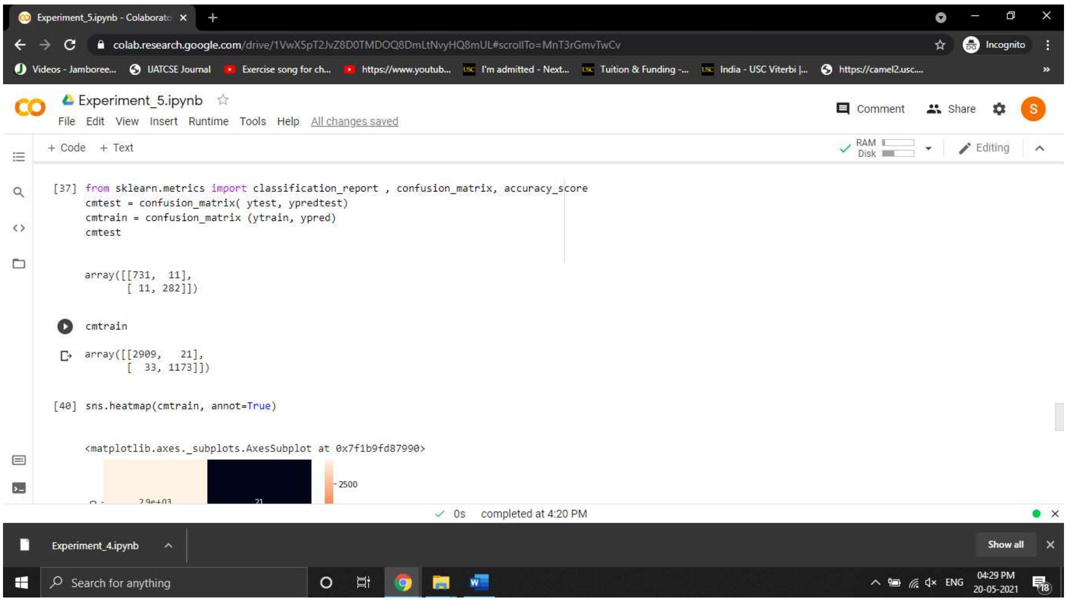


```
[22] from sklearn.feature_extraction.text import CountVectorizer

[26] vector = CountVectorizer()
     spam_ham = vector.fit_transform(df['text'])
     spam_ham.toarray


     <bound method _cs_matrix.toarray of <5171x50447 sparse matrix of type '<class 'numpy.int64'>'
             with 456145 stored elements in Compressed Sparse Row format>>


[29] x =spam_ham
     y= df['label_num'].values
     y

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

[30] from sklearn.model_selection import train_test_split
     xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.2, random_state=42)
```

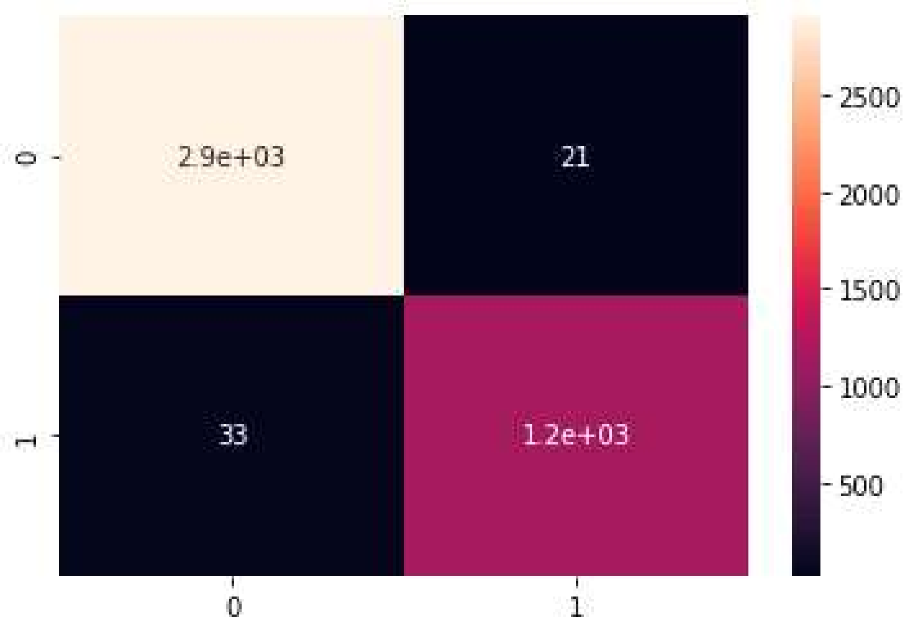

Experiment_5.ipynb - Colaborato
colab.research.google.com/drive/1VwX5pT2JvZ8D0TMDOQ8DmLtNvyHQ8mUL#scrollTo=MnT3rGmvTwCv
Incognito
Experiment_5.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
0s completed at 4:20 PM
Experiment_4.ipynb
Show all
Search for anything
ENG
04:29 PM
20-05-2021

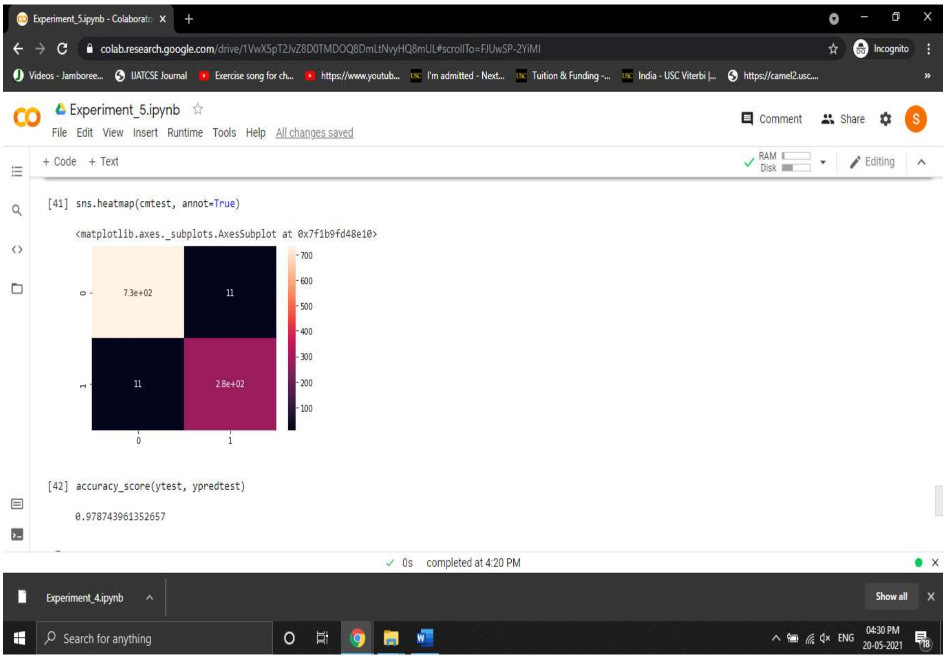
```
[31] from sklearn.naive_bayes import MultinomialNB
     nb = MultinomialNB()
     nb.fit(xtrain,ytrain)

     MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)

[32] ypred = nb.predict(xtrain)
     ypred

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

     ypredtest = nb.predict(xtest)
     ypredtest

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

[37] from sklearn.metrics import classification_report , confusion_matrix, accuracy_score
     cmtest = confusion_matrix( ytest, ypredtest)
     cmtrain = confusion_matrix (ytrain, ypred)
     cmtest
```

## Classification Report and Confusion Matrix


Experiment_5.ipynb - Colaborato
colab.research.google.com/drive/1VwX5pT2JvZ8D0TMDOQ8DmLtNvyHQ8mUL#scrollTo=MnT3rGmvTwCv
Incognito
Experiment_5.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
2.9e+03
21
2500
0s completed at 4:20 PM
Experiment_4.ipynb
Show all
Search for anything
ENG
04:29 PM
20-05-2021

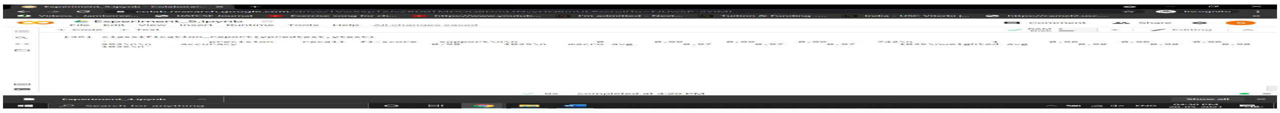
```
[37] from sklearn.metrics import classification_report , confusion_matrix, accuracy_score
     cmtest = confusion_matrix( ytest, ypredtest)
     cmtrain = confusion_matrix (ytrain, ypred)
     cmtest

     array([[731,  11],
            [ 11, 282]])

     cmtrain

     array([[2909,   21],
            [  33, 1173]])

[40] sns.heatmap(cmtrain, annot=True)

     <matplotlib.axes._subplots.AxesSubplot at 0x7f1b9fd87990>
```





```
[41] sns.heatmap(cmtest, annot=True)

     <matplotlib.axes._subplots.AxesSubplot at 0x7f1b9fd48e10>
```

```
[42] accuracy_score(ytest, ypredtest)

     0.978743961352657
```



```
[46] classification_report(ypredtest,ytest)
     [illegible]
```

# Github Link

https://github.com/chanpreet1999/ML-Assignment/blob/master/Exp5/Machine%20Learning%20Experiment%205.ipynb