Mini Project – 2 B Based on ML

Spam Mail Prediction

2 A Based
on ML

(SEMESTER VI)

submitted in

partial fulfillment of requirement for the award of

degree of

Bachelor of Engineering

in

Information Technology

By

Name of Student:	Roll no.
1) Aditi Srivastava	72
2) Aishwarya Haresh Nimkar	42
3) Devayani Sudhakar Kurup	27
4) Alisha Irfan Khot	25

Guided by

Prof. Swati Powar

Department of Information Technology

Finolex Academy of Management and Technology, Ratnagiri

ACADEMIC YEAR 2022-23

Declaration

We, hereby declare that, the Mini Project titled **"Spam Mail Prediction"** submitted here in has been carried out by us in the Department of Information Technology at Finolex Academy of Management and Technology, Ratnagiri.

Name of the student:-

1. **Aditi Srivastava**
2. **Aishwarya Haresh Nimkar**
3. **Devayani Sudhakar Kurup**
4. **Alisha Irfan Khot**

Date:

Certificate

The Mini Project **"Spam Mail Prediction"** submitted by **"Aditi Srivastava, Aishwarya Nimkar,Devayani Kurup,Alisha Khot"** in the completion of TE Bachelor in Information Technology, has been carried out under my supervision at the Department of Information Technology at Finolex Academy of Management and Technology, Ratnagiri .The work is comprehensive, complete and fit for evaluation.

Prof. Swati Powar,

Assistant Professor,

Department of Information Technology, FAMT, Ratnagiri

INDEX

INTRODUCTION

The project title is "Spam Mail Prediction", the scope of the "Spam Mail Prediction" project seems to be to predict Spam and Ham mails. Users can automatically write evaluations or comments on e-commerce websites in the Web 2.0 era. Both customers and collaborations benefit greatly from user-generated content. On the one hand, reading these evaluations before purchasing a product or service can provide buyers with some knowledge about it. Business companies, on the other hand, might use these reviews to improve their goods and marketing methods. When it comes to purchase decisions, people are often influenced by reviews information, therefore favourable evaluations may bring a lot of money and recognition to businesses and individuals. This encourages the spread of false opinion spam (also known as false reviews) . However, spam filtering helps to reduce recipient overload to some extent, but with these adjustments, it is feasible to construct an email system that is more efficient and accurate. In addition, a system that gives user specific output has been sought. This ensures that everyone who utilizes the system has a positive experience. The objective is to create a machine learning model for predicting email spam or ham, which might eventually replace updatable classifier models by predicting outcomes in the form of the greatest accuracy by comparing supervised algorithms.

LITERATURE REVIEW

Author	Year	Title	Algorithm(s) Used	Dataset Size
N. Jindal and B. Liu	2007	Analyzing and detecting review spam	Decision Trees, Random Forests, Support Vector Machines, Artificial Neural Networks	768 mails
S. Xie, G. Wang, S. Lin and P. S. Yu	2012	Review spam detection via temporal pattern discovery	Gradient Boosting Decision Tree	7,364 mails
G. Wang, S. Xie, B. Liu and P. S. Yu	2012	Identify online store review spammers via social review graph	Logistic Regression, Support Vector Machines, Decision Trees, Neural Networks	19,564 reviews
J. K. Rout, S. Singh, S. K. Jena, and S. Bakshi	2017	Deceptive review detection using labeled and unlabeled data	K-Nearest Neighbors, Random Forests, Support Vector Machines, Naive Bayes	768 mails
N. Hussain, H. T. Mirza, I. Hussain, F. Iqbal, and I. Memon	2020	Spam review detection using the linguistic and spammer behavioral methods	Logistic Regression, Support Vector Machines, Decision Trees, Random Forests	2,321 reviews

SOFTWARE AND HARDWARE REQUIREMENTS

- **Software Requirements:**
  1. Required Python libraries (numpy, pandas, scikit-learn, etc.)
  2. Google Colab

- **Hardware Requirements:**
  1. Windows 10 operating system.
  2. Intel Core i7 processor
  3. 8GB RAM
  4. At least 100GB of free hard drive space.

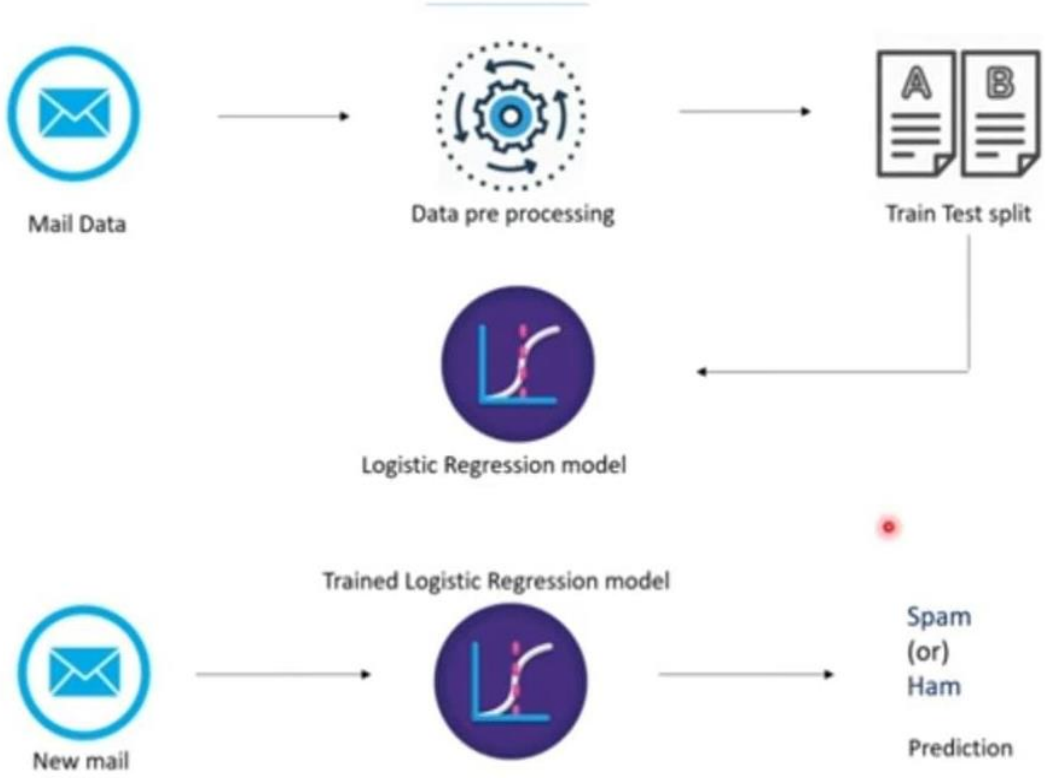
GANTT CHART

Task name	Start Date	End Date	Assigned to	Progress	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10
Finalizing requirements	14/01/2023	21/01/2023	Aditi, Aishwarya, Devayani, Alisha	100%										
Determine all roles and responsibilities and develop a schedule	22/01/2023	28/01/2023	Aditi, Aishwarya, Devayani, Alisha	100%										
Collecting resources and planning required technologies	29/01/2023	04/02/2023	Aditi, Aishwarya, Devayani, Alisha	100%										
Creating literature review	05/02/2023	18/02/2023	Aditi, Aishwarya, Devayani, Alisha	100%										
Logistics Regression Model Development	19/02/2023	04/03/2023	Aditi, Aishwarya, Devayani, Alisha	100%										
Model Evaluation and Comparison	19/03/2023	25/03/2023	Aditi, Aishwarya, Devayani, Alisha	100%										
Presentation Preparation	26/03/2023	01/04/2023	Aditi, Aishwarya, Devayani, Alisha	100%										
Documentation and Report Writing	02/04/2023	15/04/2023	Aditi, Aishwarya, Devayani, Alisha	100%										
Final review and submission	16/04/2023	20/04/2023	Aditi, Aishwarya, Devayani, Alisha	100%										

SYSTEM DESIGN

1. Spam Mail Prediction Using Logistic Regression Algorithm:

Work Flow



SOURCE CODE

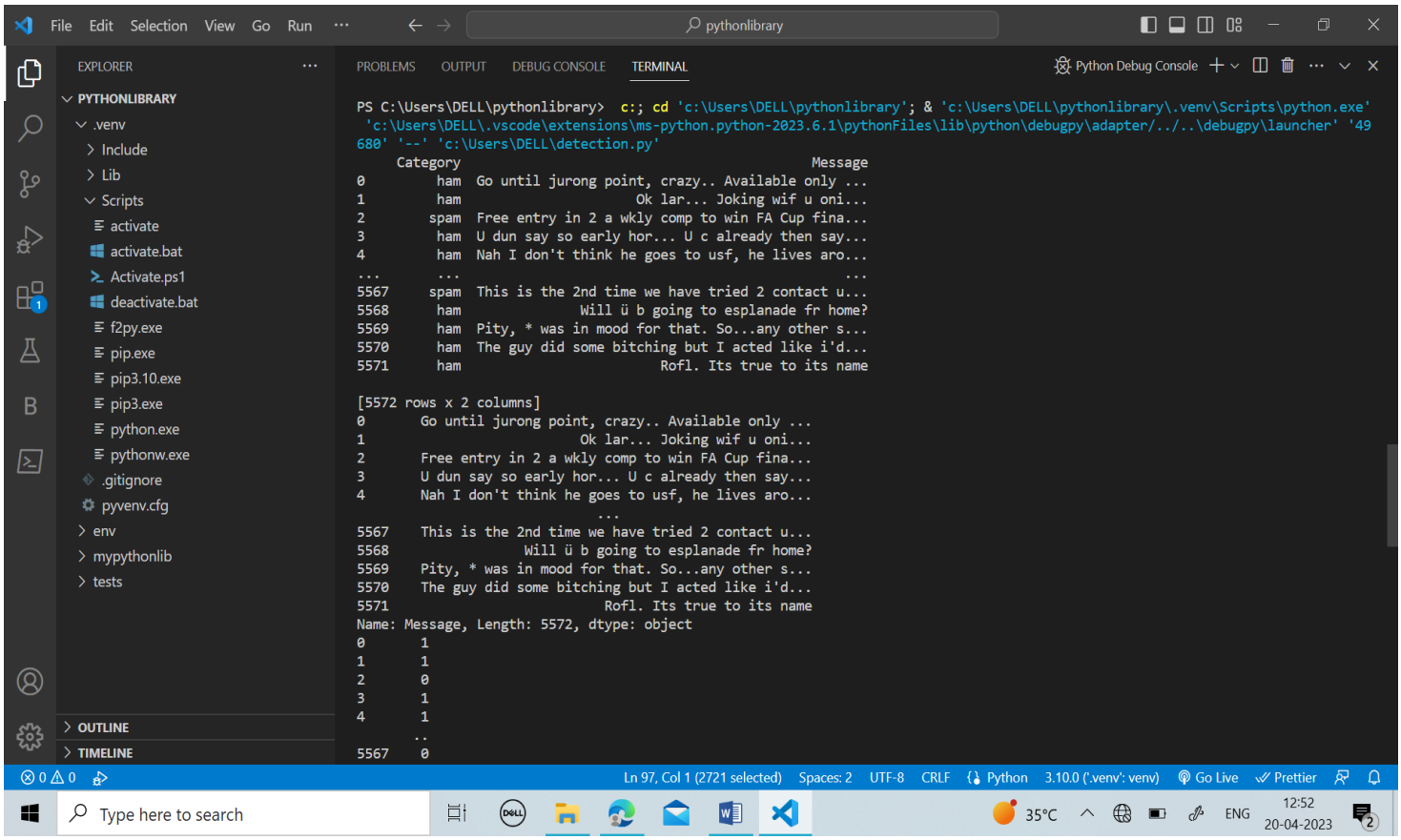
Spam Mail Prediction Using Naïve Bayes Algorithm:

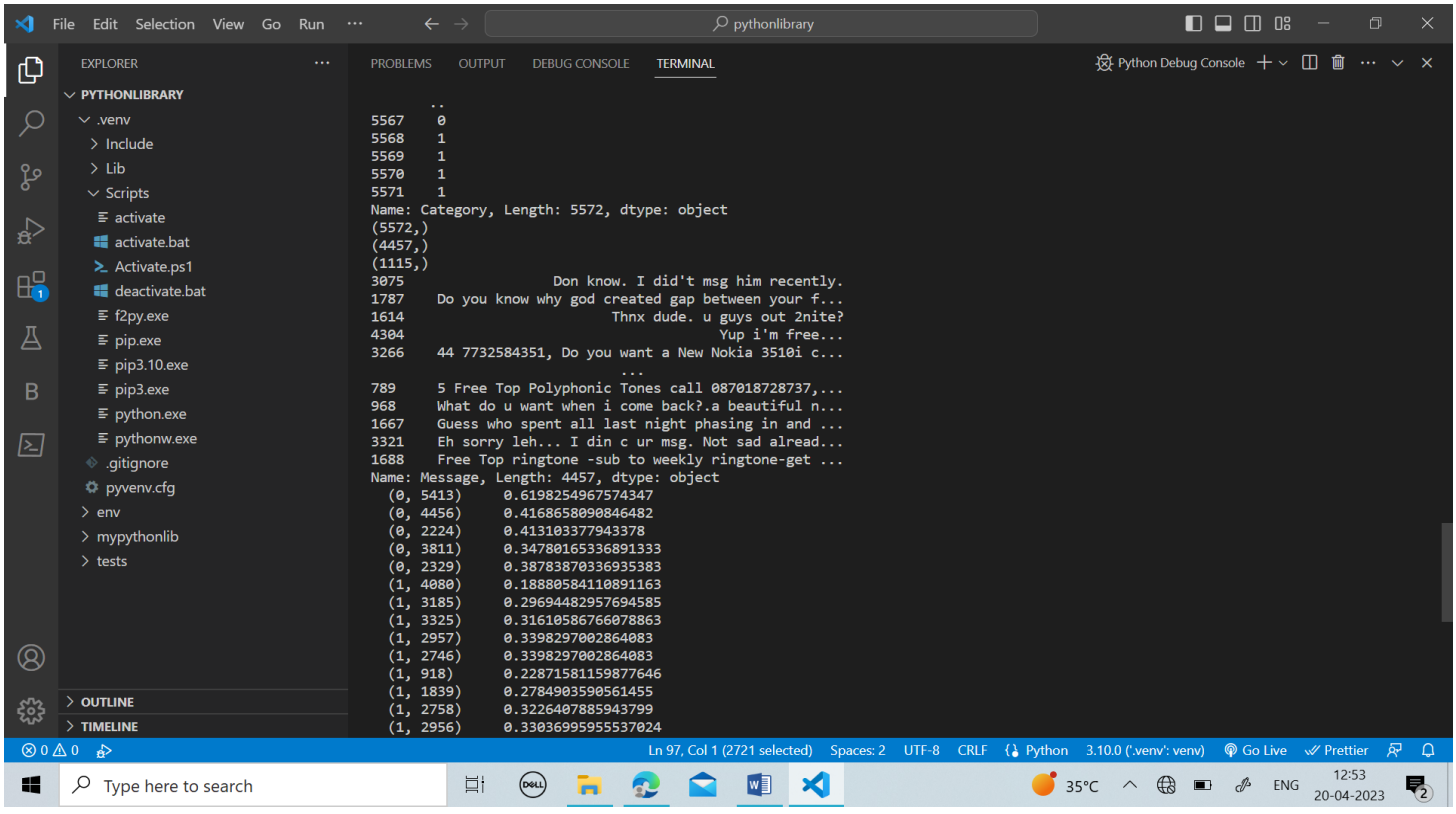
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# loading the data from csv file to a pandas Dataframe
raw_mail_data = pd.read_csv('C:/Users/DELL/Downloads/mail_data.csv')

print(raw_mail_data)

# replace the null values with a null string
mail_data = raw_mail_data.where((pd.notnull(raw_mail_data)),'')

# printing the first 5 rows of the dataframe
mail_data.head()

# checking the number of rows and columns in the dataframe
mail_data.shape

# label spam mail as 0;  ham mail as 1;

mail_data.loc[mail_data['Category'] == 'spam', 'Category',] = 0
mail_data.loc[mail_data['Category'] == 'ham', 'Category',] = 1

# separating the data as texts and label

X = mail_data['Message']

Y = mail_data['Category']

print(X)
print(Y)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=3)

print(X.shape)
print(X_train.shape)
print(X_test.shape)

# transform the text data to feature vectors that can be used as input to the Logistic
regression

feature_extraction = TfidfVectorizer(min_df = 1, stop_words='english', lowercase=True)
```

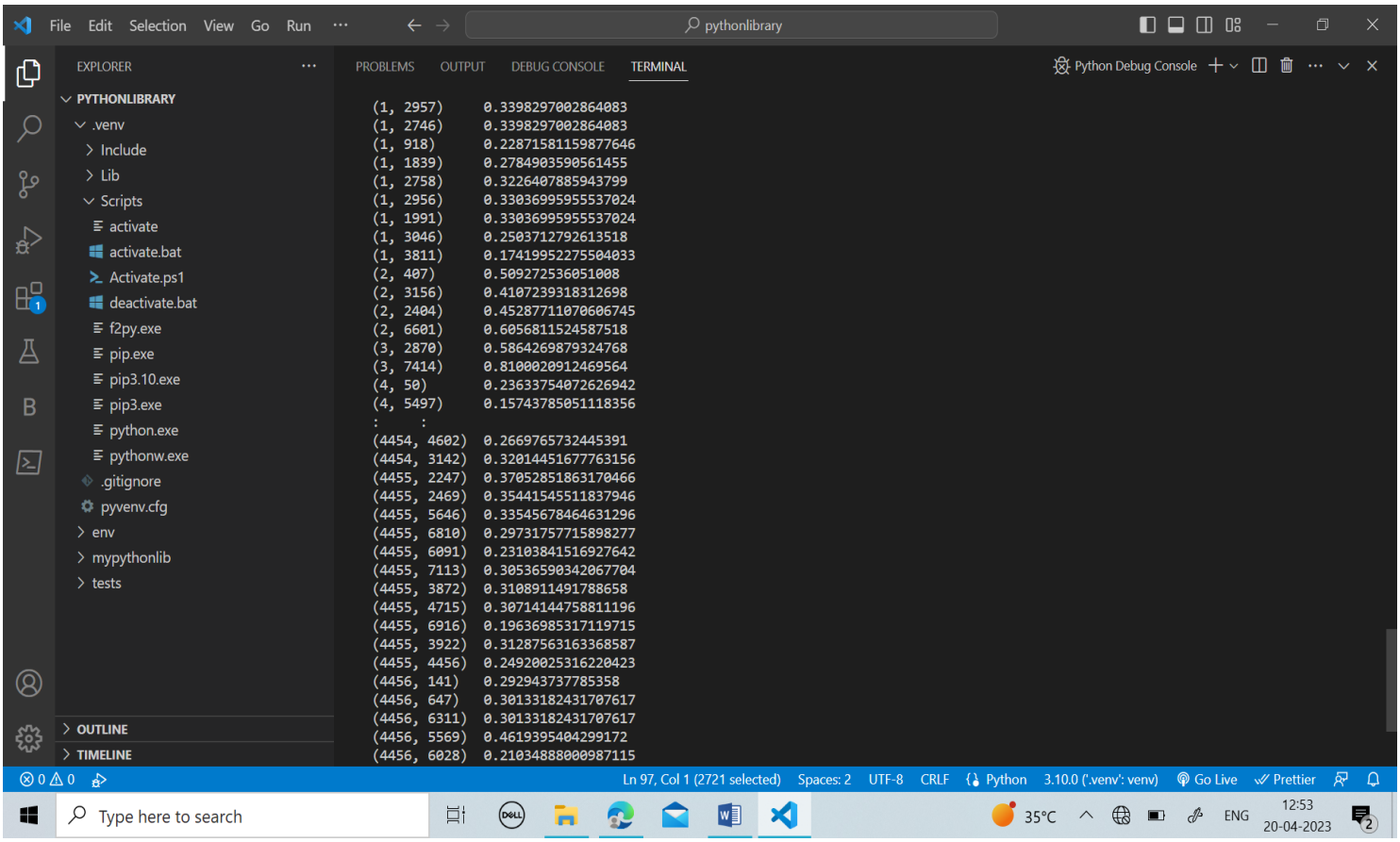
```
feature_extraction.fit(X_test)
X_train_features = feature_extraction.fit_transform(X_train)
X_test_features = feature_extraction.transform(X_test)

# convert Y_train and Y_test values as integers

Y_train = Y_train.astype('int')
Y_test = Y_test.astype('int')

print(X_train)
print(X_train_features)

model = LogisticRegression()

# training the Logistic Regression model with the training data
model.fit(X_train_features, Y_train)

# prediction on training data

prediction_on_training_data = model.predict(X_train_features)
accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)

print('Accuracy on training data : ', accuracy_on_training_data)

# prediction on test data

prediction_on_test_data = model.predict(X_test_features)
accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)

print('Accuracy on test data : ', accuracy_on_test_data)

input_mail = ["I've been searching for the right words to thank you for this breather. I
promise i wont take your help for granted and will fulfil my promise. You have been wonderful
and a blessing at all times"]

# convert text to feature vectors
input_data_features = feature_extraction.transform(input_mail)

# making prediction

prediction = model.predict(input_data_features)
print(prediction)

if (prediction[0]==1):
  print('Ham mail')

else:
  print('Spam mail')
```

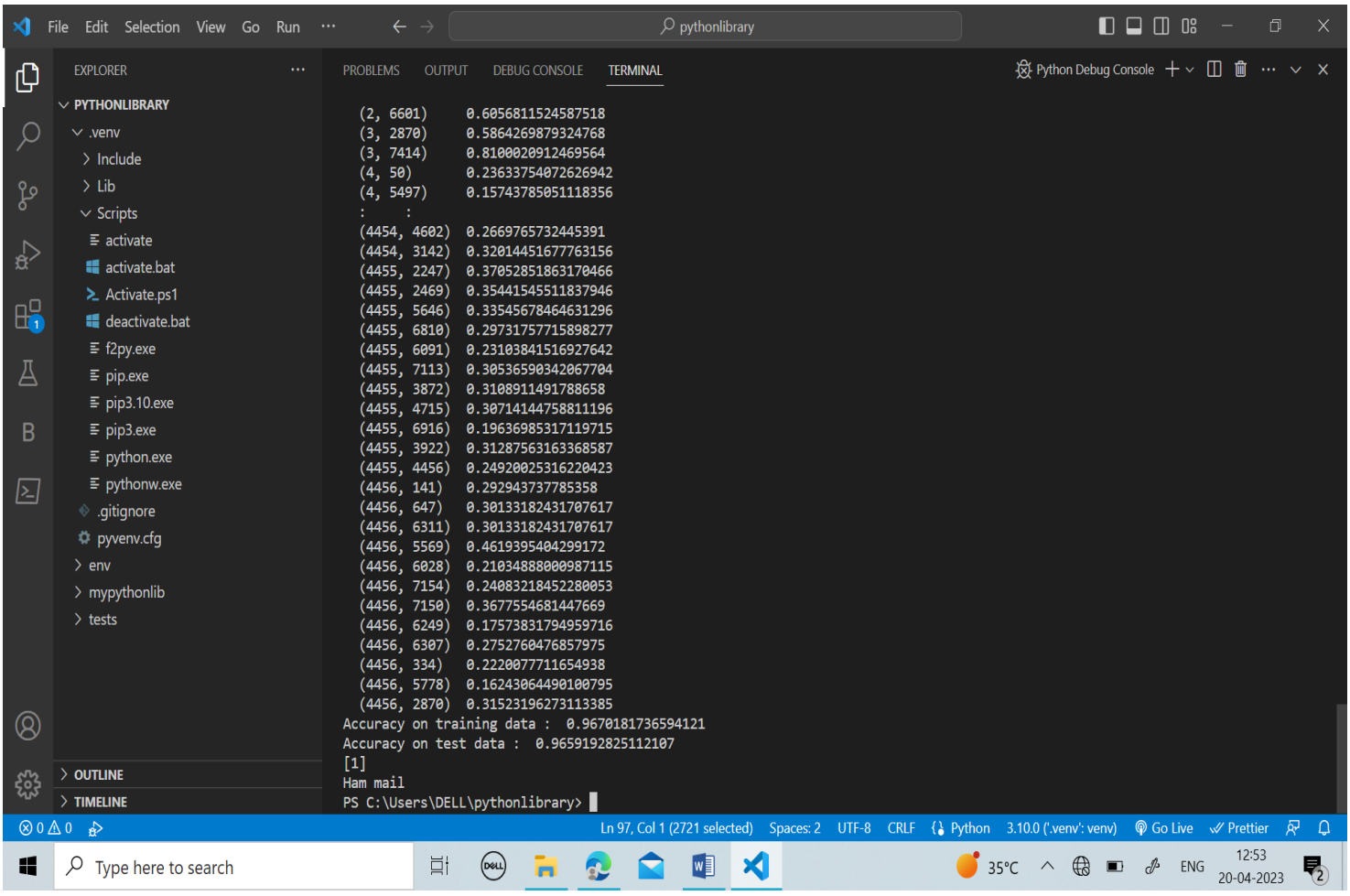
OUTPUT SCREENSHOTS

Spam Mail Prediction Using Naïve Bayes Algorithm:






File Edit Selection View Go Run
pythonlibrary
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console
PYTHONLIBRARY
.venv
Include
Lib
Scripts
activate
activate.bat
Activate.ps1
deactivate.bat
f2py.exe
pip.exe
pip3.10.exe
pip3.exe
python.exe
pythonw.exe
.gitignore
pyvenv.cfg
env
mypythonlib
tests
OUTLINE
TIMELINE
(1, 2957) 0.3398297002864083
(1, 2746) 0.3398297002864083
(1, 918) 0.22871581159877646
(1, 1839) 0.2784903590561455
(1, 2758) 0.3226407885943799
(1, 2956) 0.33036995955537024
(1, 1991) 0.33036995955537024
(1, 3046) 0.2503712792613518
(1, 3811) 0.17419952275504033
(2, 407) 0.509272536051008
(2, 3156) 0.4107239318312698
(2, 2404) 0.45287711070606745
(2, 6601) 0.6056811524587518
(3, 2870) 0.5864269879324768
(3, 7414) 0.8100020912469564
(4, 50) 0.23633754072626942
(4, 5497) 0.15743785051118356
: :
(4454, 4602) 0.2669765732445391
(4454, 3142) 0.32014451677763156
(4455, 2247) 0.37052851863170466
(4455, 2469) 0.35441545511837946
(4455, 5646) 0.33545678464631296
(4455, 6810) 0.29731757715898277
(4455, 6091) 0.23103841516927642
(4455, 7113) 0.30536590342067704
(4455, 3872) 0.3108911491788658
(4455, 4715) 0.30714144758811196
(4455, 6916) 0.19636985317119715
(4455, 3922) 0.31287563163368587
(4455, 4456) 0.24920025316220423
(4456, 141) 0.292943737785358
(4456, 647) 0.30133182431707617
(4456, 6311) 0.30133182431707617
(4456, 5569) 0.4619395404299172
(4456, 6028) 0.21034888000987115
Ln 97, Col 1 (2721 selected) Spaces: 2 UTF-8 CRLF Python 3.10.0 ('.venv': venv) Go Live Prettier
Type here to search
35°C
ENG
12:53
20-04-2023


File Edit Selection View Go Run
pythonlibrary
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python Debug Console
PYTHONLIBRARY
.venv
Include
Lib
Scripts
activate
activate.bat
Activate.ps1
deactivate.bat
f2py.exe
pip.exe
pip3.10.exe
pip3.exe
python.exe
pythonw.exe
.gitignore
pyvenv.cfg
env
mypythonlib
tests
OUTLINE
TIMELINE
(2, 6601) 0.6056811524587518
(3, 2870) 0.5864269879324768
(3, 7414) 0.8100020912469564
(4, 50) 0.23633754072626942
(4, 5497) 0.15743785051118356
: :
(4454, 4602) 0.2669765732445391
(4454, 3142) 0.32014451677763156
(4455, 2247) 0.37052851863170466
(4455, 2469) 0.35441545511837946
(4455, 5646) 0.33545678464631296
(4455, 6810) 0.29731757715898277
(4455, 6091) 0.23103841516927642
(4455, 7113) 0.30536590342067704
(4455, 3872) 0.3108911491788658
(4455, 4715) 0.30714144758811196
(4455, 6916) 0.19636985317119715
(4455, 3922) 0.31287563163368587
(4455, 4456) 0.24920025316220423
(4456, 141) 0.292943737785358
(4456, 647) 0.30133182431707617
(4456, 6311) 0.30133182431707617
(4456, 5569) 0.4619395404299172
(4456, 6028) 0.21034888000987115
(4456, 7154) 0.24083218452280053
(4456, 7150) 0.3677554681447669
(4456, 6249) 0.17573831794959716
(4456, 6307) 0.2752760476857975
(4456, 334) 0.2220077711654938
(4456, 5778) 0.16243064490100795
(4456, 2870) 0.31523196273113385
Accuracy on training data : 0.9670181736594121
Accuracy on test data : 0.9659192825112107
[1]
Ham mail
PS C:\Users\DELL\pythonlibrary>
Ln 97, Col 1 (2721 selected) Spaces: 2 UTF-8 CRLF Python 3.10.0 ('.venv': venv) Go Live Prettier
Type here to search
35°C
ENG
12:53
20-04-2023

CONCLUSION

In conclusion, in today's age of communication and technology, spam email is one of the most demanding and unpleasant concerns on the internet. For safeguarding message and e-mail transmission, spam detection is very necessary. The accurate detection of spam is a big challenge, and researchers have proposed a lot of detection approaches. These approaches, are incapable of proper and efficient detection of spam. To solve this problem, we suggested a spam detection model based on machine learning prediction models .When compared to other current methods, the proposed method attained a high accuracy of 96 percent. As a result, the suggested system is structured in such a way that it recognises unsolicited and undesired mails and blocks them, hence minimising spam messages, which would be beneficial to people.

REFERENCE

1. N. Jindal and B. Liu, "Analyzing and detecting review spam", Proc. IEEE Int. Conf. Data Mining, pp. 547- 552, Oct. 2007

2. S. Xie, G. Wang, S. Lin and P. S. Yu, "Review spam detection via temporal pattern discovery", Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 823-831, 2012

3. G. Wang, S. Xie, B. Liu and P. S. Yu, "Identify online store review spammers via social review graph", ACM Trans. Intell. Syst. Technol., vol. 3, no. 4, pp. 1-61, 2012

4. J. K. Rout, S. Singh, S. K. Jena, and S. Bakshi, “Deceptive review detection using labeled and unlabeled data,” Multimedia Tools Appl., vol. 76,pp. 3187–3211, 2017

5. N. Hussain, H. T. Mirza, I. Hussain, F. Iqbal, and I. Memon, “Spam review detection using the linguistic and spammer behavioral methods,” IEEE Access, vol. 8, pp. 53801–53816, 2020