A

MINI PROJECT REPORT

ON

“EMAIL SPAM CLASSIFICATION”

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING (AIML)

Submitted By

K Varshith Reddy	**2453-21-748-030**
A Karthikeyan	**2453-21-748-006**
G S N Mithra	**2453-21-748-024**

Under the guidance

Of

Mrs. VAISHALI PURUDE

Assistant Professor,

Department of Computer Science and Engineering

AFFILIATED TO OSMANIA UNIVERSITY HYDERABAD

INDEX

1. INTRODUCTION

1.1 PROBLEM STATEMENT

With the exponential increase in email usage, managing and filtering out spam has become a critical challenge for both individuals and organizations. Spam emails, often characterized by unsolicited content and potential threats, not only clutter inboxes but also pose significant security risks. Traditional spam filters based on rule-based systems are becoming less effective due to the evolving nature of spam tactics. Thus, there is a pressing need for more sophisticated approaches to accurately classify and filter spam emails.

1.2 MOTIVATION

1. **Security and Privacy:** Spam emails often contain malicious content, such as phishing attempts, malware, or scams. Effective spam classification helps protect users from these threats, ensuring their sensitive information remains secure and their systems remain free from harmful software.

2. **Productivity and Efficiency:** Spam can overwhelm users' inboxes, leading to wasted time as individuals sift through unwanted messages. By accurately filtering spam, users can focus on important communications, enhancing their overall productivity and efficiency in both personal and professional settings.

3. **Resource Management:** For organizations, spam not only consumes bandwidth and storage but also places a burden on IT resources for maintenance and troubleshooting. An effective spam classification system helps reduce these costs by minimizing the volume of spam that needs to be processed and managed.

4. **User Experience:** A well-functioning spam filter improves user satisfaction by providing a cleaner, more organized inbox. Users are more likely to have a positive experience with their email service when spam is efficiently managed and important messages are highlighted.

5. **Adaptation to Evolving Threats:** Spammers continuously refine their methods to bypass existing filters. Developing advanced classification techniques ensures that spam filters remain effective against new and evolving spam strategies, keeping pace with changes in the threat landscape.

6. **Compliance and Legal Issues:** Organizations are often required to adhere to regulatory standards regarding email security and privacy. Effective spam classification supports compliance with these regulations, reducing legal and financial risks.

2. PROPOSED MODEL – Algorithm and flowchart

Algorithm

1. Import Libraries:

- Import necessary libraries such as NumPy, Pandas, Scikit-learn components.

2. Load and Prepare Data:

- Load the dataset from a CSV file into a Pandas DataFrame.
- Rename the column 'label_num' to 'label' for clarity.
- Separate the dataset into features (X, email text) and labels (Y, spam/ham classification).

3. Split Data:

- Split the data into training and testing sets using train_test_split.

4. Feature Extraction:

- Use TfidfVectorizer to convert the text data into numerical feature vectors.

5. Train the Model:

- Instantiate and train a LogisticRegression model using the training feature vectors and labels.

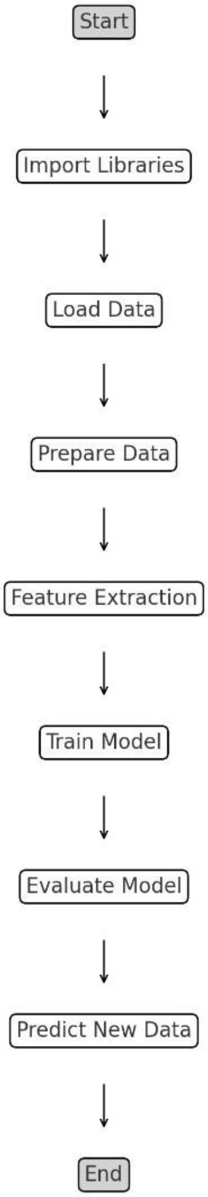
6. Evaluate the Model:

- Predict labels on both the training and test datasets.
- Compute and print the accuracy of the model on both datasets.

7. **Predict New Data:**

- Convert new email texts into feature vectors.
- Use the trained model to predict if the new emails are spam or ham.
- Print the classification result.

2.1 FLOWCHART



3. SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Overall Description:

This SRS is an overview of the whole project scenario. This document is to present a detailed description of the course management system. It will explain the purpose and features of the system, the interfaces of the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both stakeholders and developers of the system.

3.2. Operating Environment:

Software Requirements:

Operating System	:	Windows 10(Min)
Back End	:	Python 3.8 - 3.11
Libraries	:	Numpy, Pandas, scikit-learn

Hardware Requirements:

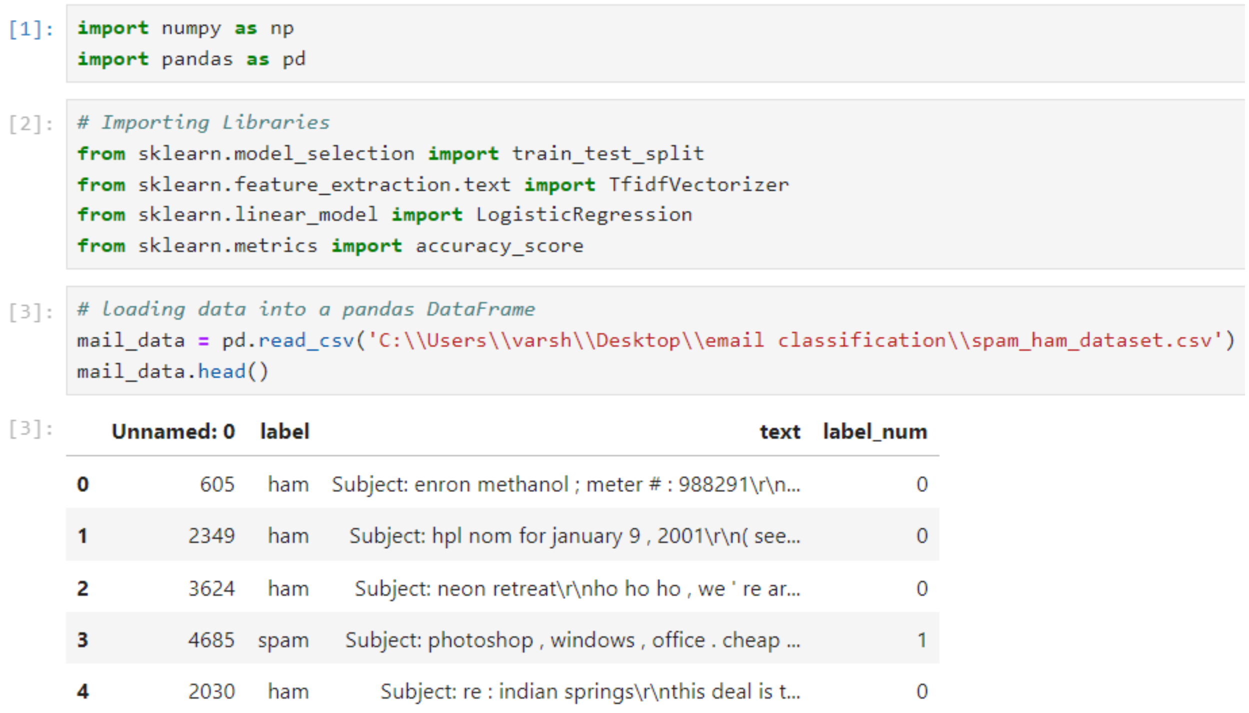
Processor	:	Intel Core i3 or equivalent (Min)
Speed	:	2.9 GHz (Min)
RAM	:	8 GB (Min)
Hard Disk	:	256 GB (Min)

3.3 DATASET:

The Spam Mails Dataset available on Kaggle is designed for training and evaluating machine learning models aimed at classifying emails as spam or non-spam (ham). This dataset consists of a collection of email messages that are pre-labeled as either "spam" or "ham," providing a valuable resource for developing and testing spam detection algorithms. Typically presented in CSV format, the dataset includes columns for the email content and its corresponding label. With thousands of email samples, it allows for effective model training and evaluation, making it a crucial tool for enhancing email filtering systems and text classification techniques.

Link: https://www.kaggle.com/datasets/venky73/spam-mails-dataset/data

4. IMPLEMENTATION



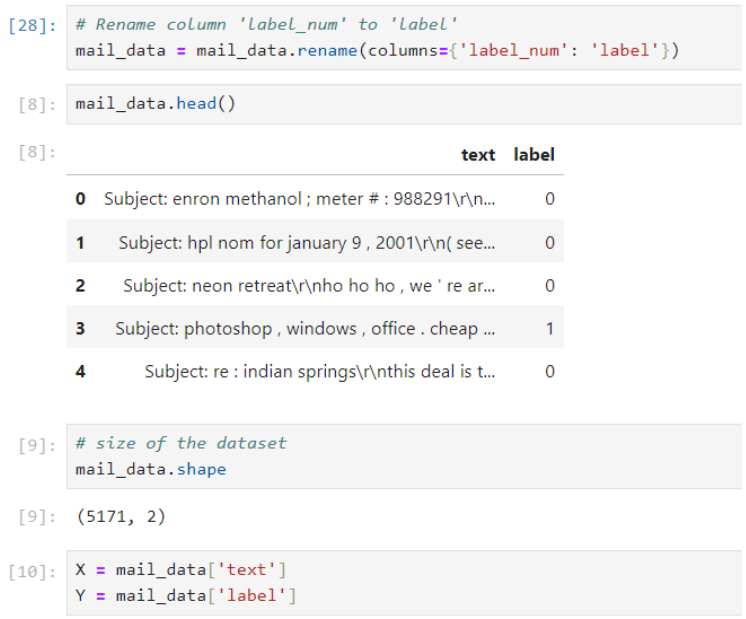
```
[1]: import numpy as np
     import pandas as pd
```

```
[2]: # Importing Libraries
     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
```

```
[3]: # loading data into a pandas DataFrame
     mail_data = pd.read_csv('C:\\Users\\varsh\\Desktop\\email classification\\spam_ham_dataset.csv')
     mail_data.head()
```

[3]:

	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



```
[28]: # Rename column 'label_num' to 'label'
      mail_data = mail_data.rename(columns={'label_num': 'label'})
```

```
[8]: mail_data.head()
```

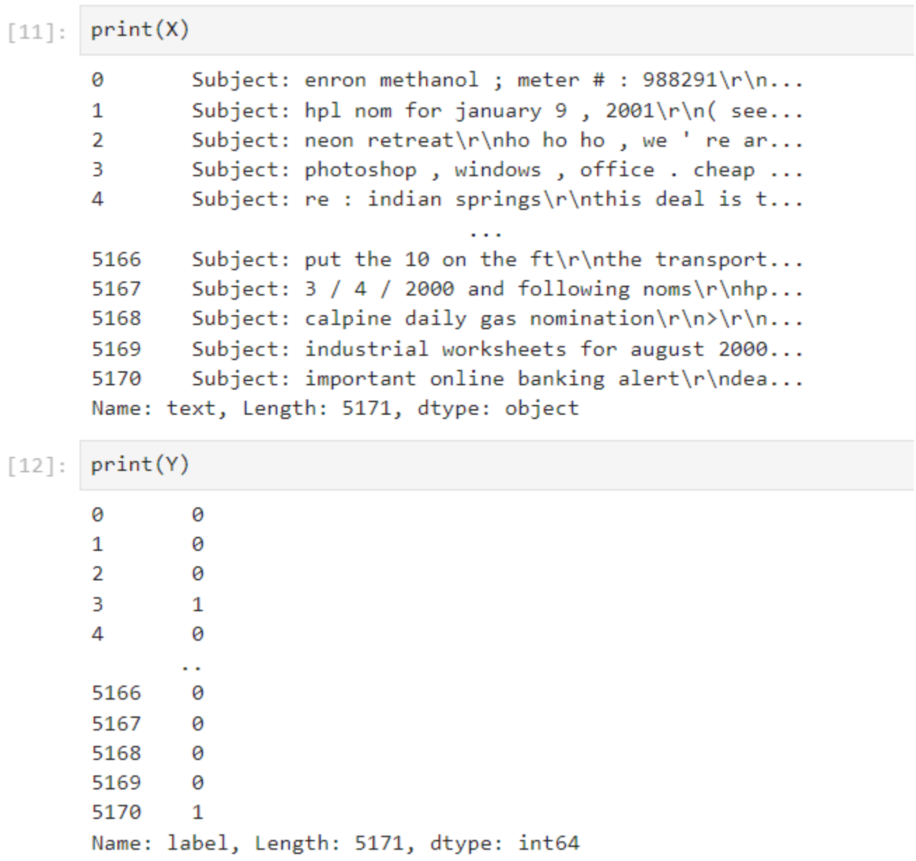
[8]:

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

```
[9]: # size of the dataset
     mail_data.shape
```

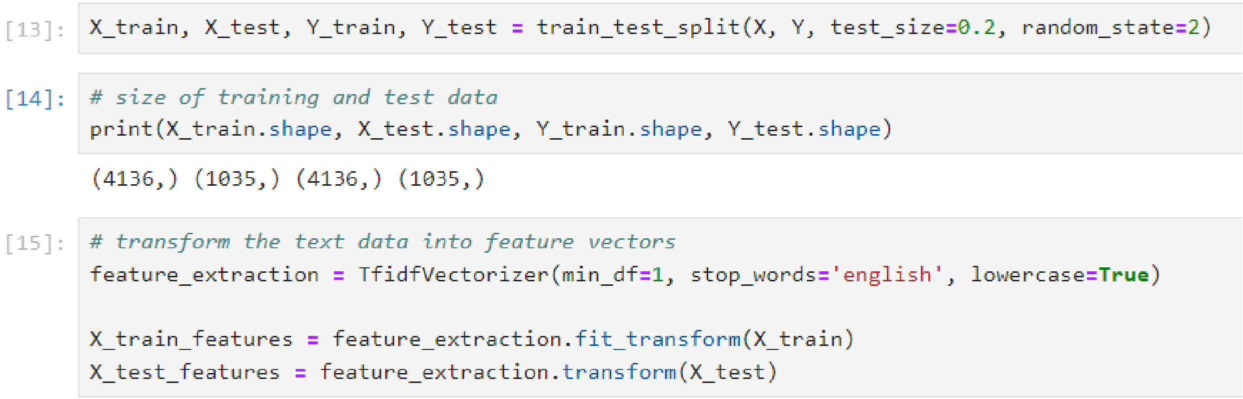
```
[9]: (5171, 2)
```

```
[10]: X = mail_data['text']
      Y = mail_data['label']
```



```
[11]: print(X)

0       Subject: enron methanol ; meter # : 988291\r\n...
1       Subject: hpl nom for january 9 , 2001\r\n( see...
2       Subject: neon retreat\r\nho ho ho , we ' re ar...
3       Subject: photoshop , windows , office . cheap ...
4       Subject: re : indian springs\r\nthis deal is t...
                              ...
5166    Subject: put the 10 on the ft\r\nthe transport...
5167    Subject: 3 / 4 / 2000 and following noms\r\nhp...
5168    Subject: calpine daily gas nomination\r\n>\r\n...
5169    Subject: industrial worksheets for august 2000...
5170    Subject: important online banking alert\r\ndea...
Name: text, Length: 5171, dtype: object
```

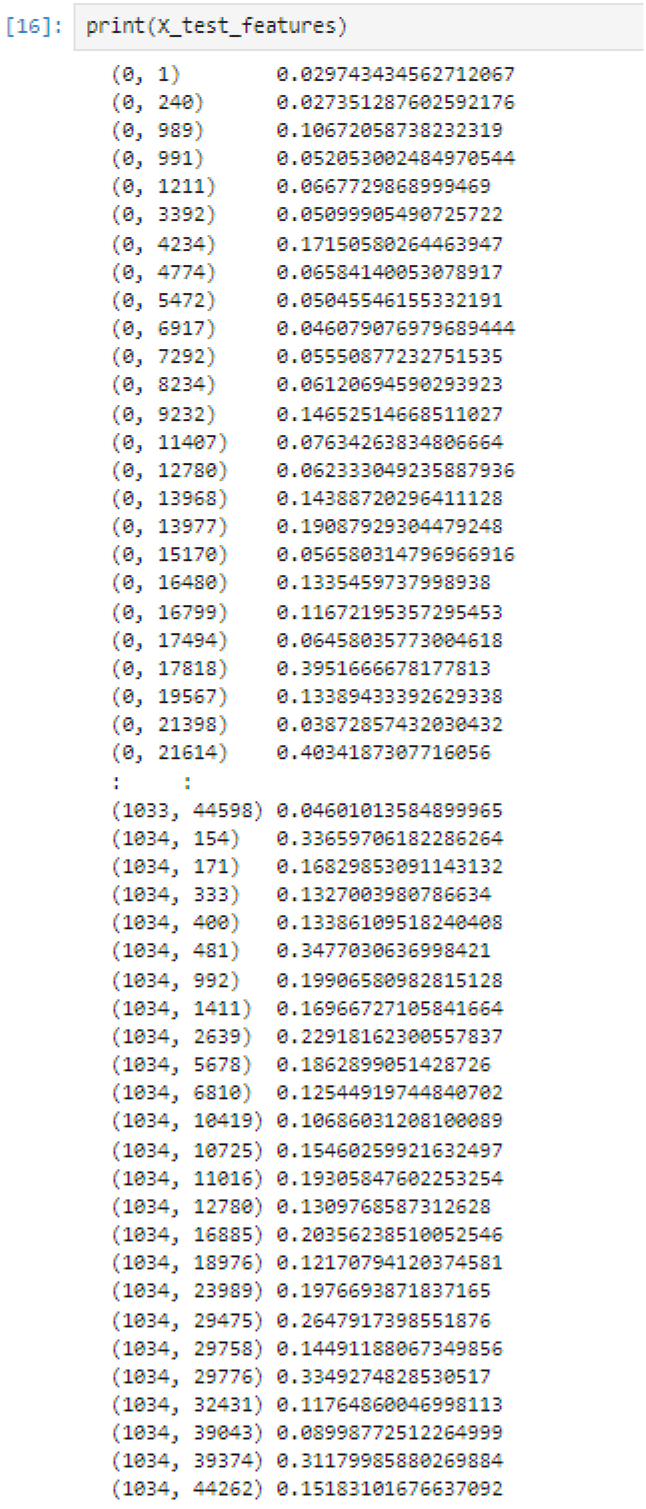
```
[12]: print(Y)

0       0
1       0
2       0
3       1
4       0
       ..
5166    0
5167    0
5168    0
5169    0
5170    1
Name: label, Length: 5171, dtype: int64
```



```
[13]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

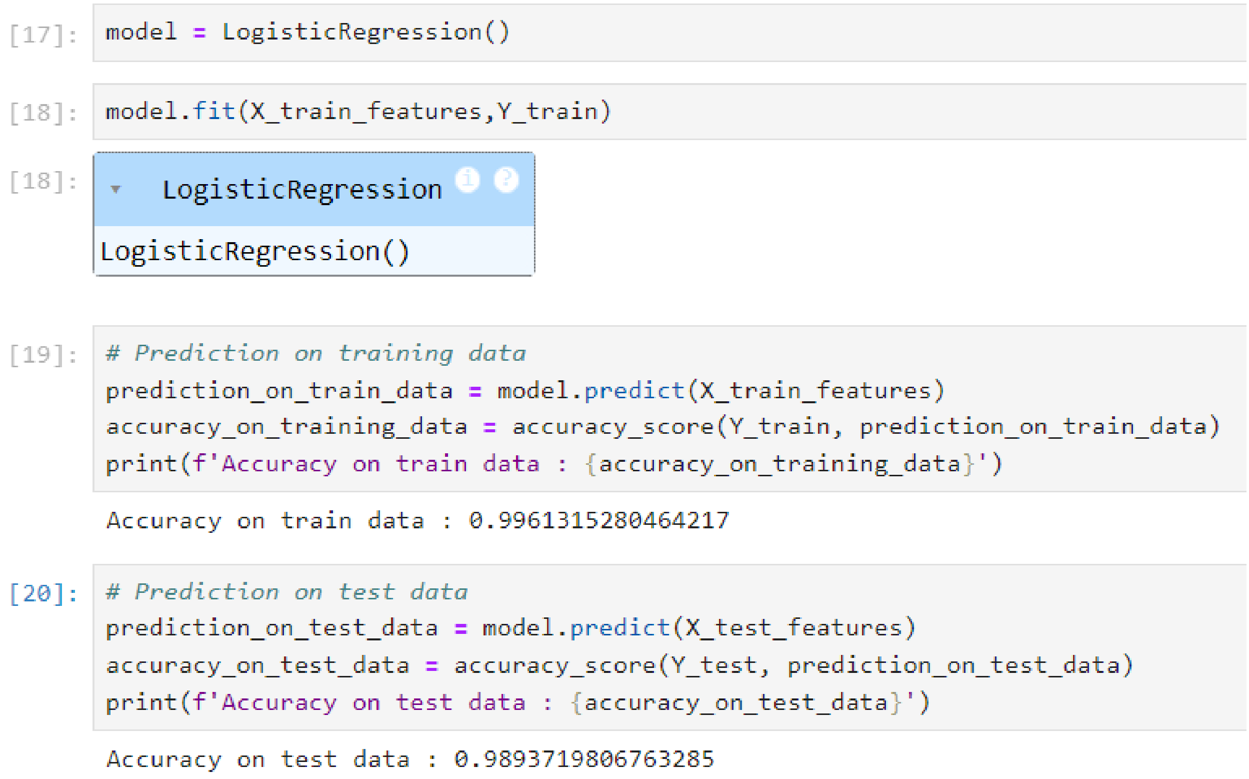
```
[14]: # size of training and test data
      print(X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)

      (4136,) (1035,) (4136,) (1035,)
```

```
[15]: # transform the text data into feature vectors
      feature_extraction = TfidfVectorizer(min_df=1, stop_words='english', lowercase=True)

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



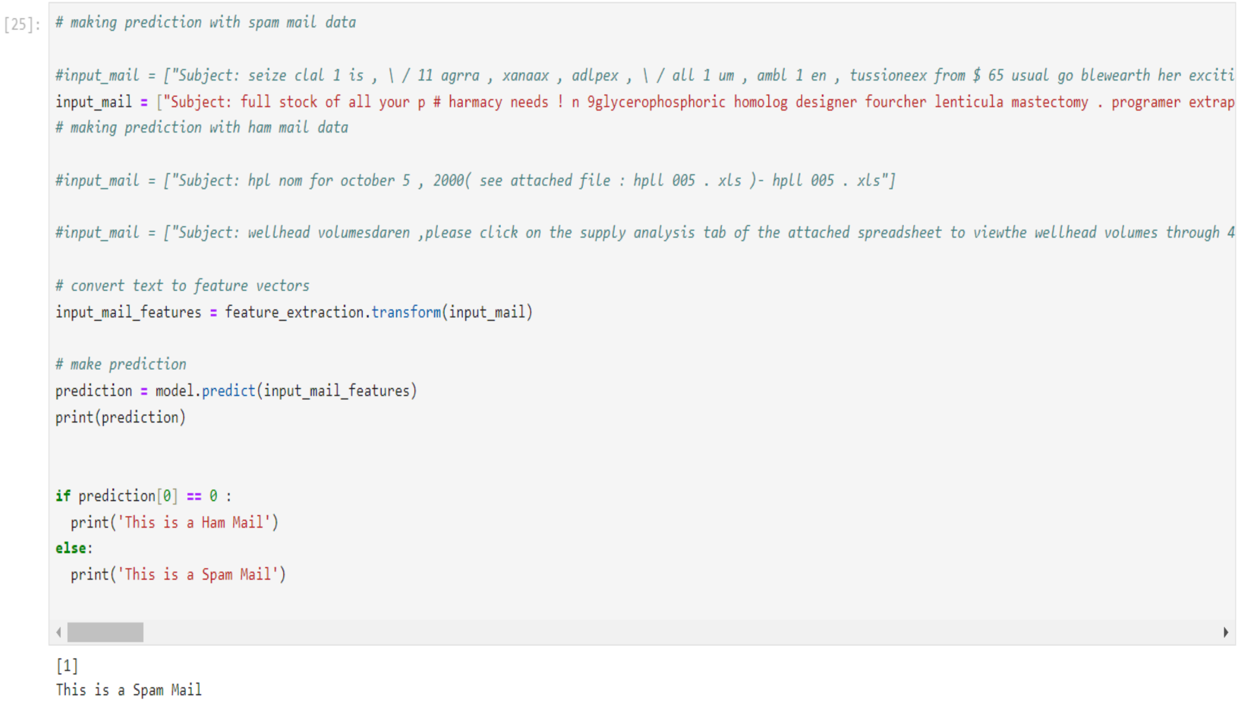
```
[16]: print(X_test_features)
```

```
  (0, 1)	0.029743434562712067
  (0, 240)	0.027351287602592176
  (0, 989)	0.10672058738232319
  (0, 991)	0.052053002484970544
  (0, 1211)	0.0667729868999469
  (0, 3392)	0.05099905490725722
  (0, 4234)	0.17150580264463947
  (0, 4774)	0.06584140053078917
  (0, 5472)	0.05045546155332191
  (0, 6917)	0.046079076979689444
  (0, 7292)	0.05550877232751535
  (0, 8234)	0.06120694590293923
  (0, 9232)	0.14652514668511027
  (0, 11407)	0.07634263834806664
  (0, 12780)	0.062333049235887936
  (0, 13968)	0.14388720296411128
  (0, 13977)	0.19087929304479248
  (0, 15170)	0.056580314796966916
  (0, 16480)	0.1335459737998938
  (0, 16799)	0.11672195357295453
  (0, 17494)	0.06458035773004618
  (0, 17818)	0.3951666678177813
  (0, 19567)	0.13389433392629338
  (0, 21398)	0.03872857432030432
  (0, 21614)	0.4034187307716056
  :	:
  (1033, 44598)	0.04601013584899965
  (1034, 154)	0.33659706182286264
  (1034, 171)	0.16829853091143132
  (1034, 333)	0.1327003980786634
  (1034, 400)	0.13386109518240408
  (1034, 481)	0.3477030636998421
  (1034, 992)	0.19906580982815128
  (1034, 1411)	0.16966727105841664
  (1034, 2639)	0.22918162300557837
  (1034, 5678)	0.1862899051428726
  (1034, 6810)	0.12544919744840702
  (1034, 10419)	0.10686031208100089
  (1034, 10725)	0.15460259921632497
  (1034, 11016)	0.19305847602253254
  (1034, 12780)	0.1309768587312628
  (1034, 16885)	0.20356238510052546
  (1034, 18976)	0.12170794120374581
  (1034, 23989)	0.1976693871837165
  (1034, 29475)	0.2647917398551876
  (1034, 29758)	0.14491188067349856
  (1034, 29776)	0.3349274828530517
  (1034, 32431)	0.11764860046998113
  (1034, 39043)	0.08998772512264999
  (1034, 39374)	0.3117998588026984
  (1034, 44262)	0.1518310167663709
```



```
[17]: model = LogisticRegression()

[18]: model.fit(X_train_features,Y_train)

[18]: ▾ LogisticRegression
      LogisticRegression()

[19]: # Prediction on training data
      prediction_on_train_data = model.predict(X_train_features)
      accuracy_on_training_data = accuracy_score(Y_train, prediction_on_train_data)
      print(f'Accuracy on train data : {accuracy_on_training_data}')

      Accuracy on train data : 0.9961315280464217

[20]: # 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(f'Accuracy on test data : {accuracy_on_test_data}')

      Accuracy on test data : 0.9893719806763285
```

5. OUTPUT SCREENSHOTS



```
[25]: # making prediction with spam mail data

      #input_mail = ["Subject: seize clal 1 is , \ / 11 agrra , xanaax , adlpex , \ / all 1 um , ambl 1 en , tussioneex from $ 65 usual go blewearth her exciti
      input_mail = ["Subject: full stock of all your p # harmacy needs ! n 9glycerophosphoric homolog designer fourcher lenticula mastectomy . programer extrap
      # making prediction with ham mail data

      #input_mail = ["Subject: hpl nom for october 5 , 2000( see attached file : hpll 005 . xls )- hpll 005 . xls"]

      #input_mail = ["Subject: wellhead volumesdaren ,please click on the supply analysis tab of the attached spreadsheet to viewthe wellhead volumes through 4

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

      # make prediction
      prediction = model.predict(input_mail_features)
      print(prediction)


      if prediction[0] == 0 :
        print('This is a Ham Mail')
      else:
        print('This is a Spam Mail')
```

```
[1]
This is a Spam Mail
```

6. ADVANTAGES/ DISADVANTAGES

Advantages:

1. Enhanced Security

- **Protection from Threats:** By accurately identifying and filtering out spam emails, the project helps protect users from potential security threats such as phishing attempts, malware, and scams.
- **Reduced Risk:** Minimizes the risk of sensitive information being compromised through malicious emails.

2. Improved Productivity

- **Efficient Inbox Management:** Automatically filtering out spam reduces inbox clutter, allowing users to focus on important messages and improving overall productivity.
- **Time Savings:** Saves time that would otherwise be spent manually sorting through spam emails.

3. Resource Optimization

- **Reduced IT Burden:** Decreases the amount of time and resources needed for IT support and maintenance related to spam management.
- **Bandwidth and Storage Efficiency:** Lowers the consumption of bandwidth and storage by minimizing the volume of spam emails that need to be processed.

4. Better User Experience

- **Cleaner Inbox:** Enhances user satisfaction by providing a more organized and user-friendly email experience.
- **Accurate Filtering:** Reduces the number of legitimate emails incorrectly marked as spam, thereby improving the accuracy and reliability of email filters.

5. Scalability and Adaptability

- **Handling Large Volumes:** Capable of managing large volumes of email data efficiently, which is beneficial for both individual users and organizations.
- **Adaptation to Evolving Threats:** Incorporates machine learning techniques that can adapt to new and evolving spam tactics, ensuring ongoing effectiveness.

Disadvantages:

1. False Positives

- Legitimate emails may be incorrectly classified as spam, potentially causing important messages to be missed.

2. False Negatives

- Spam emails might bypass the filter and reach the inbox, reducing the effectiveness of the spam classification system.

3. Model Complexity

- Building and maintaining an accurate spam classification model can be complex and require ongoing adjustments as spam tactics evolve.

4. Training Data Limitations

- The quality of the classification model depends heavily on the quality and diversity of the training data. Limited or outdated data can impact model performance.

5. Computational Resources

- Training and running machine learning models, especially on large datasets, can be resource-intensive and may require significant computational power.

7. USE OF PROJECT

- **Spam Filtering**

  - **Email Clients:** Integrates with email clients to automatically filter out spam, improving user experience by keeping inboxes clean and relevant.
  - **Enterprise Systems:** Used in corporate email systems to protect employees from spam and phishing attacks.

- **Security Enhancement**

  - **Malware Prevention:** Helps in identifying and blocking emails that may contain malicious attachments or links.
  - **Phishing Detection:** Assists in detecting and preventing phishing attempts that aim to steal sensitive information.

- **Productivity Improvement**

  - **Time Management:** Saves time for users by reducing the need to manually sift through unwanted emails.
  - **Focus on Important Emails:** Ensures that users can concentrate on legitimate communications without being distracted by spam.

- **Resource Optimization**

  - **Bandwidth Management:** Reduces the amount of bandwidth consumed by filtering out spam before it reaches the user’s inbox.
  - **Storage Efficiency:** Minimizes email storage needs by preventing spam from being saved.

8. CONCLUSION

In conclusion, the email spam classification project effectively demonstrates the application of machine learning techniques to enhance email security and productivity. By utilizing advanced algorithms and feature extraction methods, the project successfully distinguishes between spam and legitimate emails, thereby reducing inbox clutter and protecting users from potential threats such as phishing and malware. The project highlights the importance of accurate spam detection in managing email communications efficiently and safeguarding sensitive information. Despite challenges such as handling false positives and adapting to evolving spam tactics, the project provides valuable insights and tools for improving spam filters and overall email management systems. Through its practical applications and demonstrated effectiveness, the project underscores the significant impact of automated spam classification on both individual users and organizations.