SHIASH INFO SOLUTIONS

PROJECT TITLE

HOW CAN WE USE THE MACHINE LEARNING TO DETECT AND CLASSIFY THE SPAM EMAIL ENSURING THE EFFICIENT AND ACCURATE FILTERING SYSTEM

BATCH NO: 1

DOMAIN: DATA SCIENCE

DONE BY: HARISHVARAN M BE(CSE)

CONTENTS

ABSTRACT

INTRODUCTION

Email has become an integral part of our daily communication, but along with legitimate messages, our inboxes are often flooded with spam emails. Spam emails are not only a nuisance but can also pose security risks by containing phishing attempts or Malware. To combat this issue, we introduce the "Spam Email Detection Project."

In this research, a diverse dataset of textual messages, comprising both spam and legitimate content, is collected and preprocessed to extract meaningful features. These features encompass various linguistic, syntactic, and semantic characteristics of the messages, enabling the creation of a robust feature vector for each message instance.

Furthermore, feature importance analysis is conducted to gain insights into which aspects of the messages contribute most significantly to spam classification. Feature engineering techniques and natural language processing (NLP) tools are leveraged to enhance model performance and adapt to evolving spamming tactics.

The results of this research offer valuable insights into the efficacy of ML-based approaches for spam message prediction, with practical implications for enhancing the security and user experience of digital communication platforms. By leveraging advanced ML techniques, this study contributes to the ongoing effort to combat spam and protect users from unwanted and potentially malicious messages.

BACKGROUND

Background information is crucial for understanding the context and rationale behind a Spam Email Detection Machine Learning Project. Here is the background information typically included in such a project:

Email Communication:

Explain the significance of email as a primary means of communication in today's digital world. Emphasize the importance of ensuring the reliability and security of email communications.

Rise of Spam:

Provide statistics and context regarding the exponential growth of spam emails over the years. Mention the negative impact of spam, including its annoyance, time-wasting, and potential security threats.

Spamming Techniques:

Briefly describe common spamming techniques employed by spammers. This may include techniques like phishing, malware distribution, and fraudulent schemes.

OBJECTIVE

The primary objective of this project is to develop an efficient and accurate spam email detection system that can automatically identify and filter out spam emails from a user's inbox.

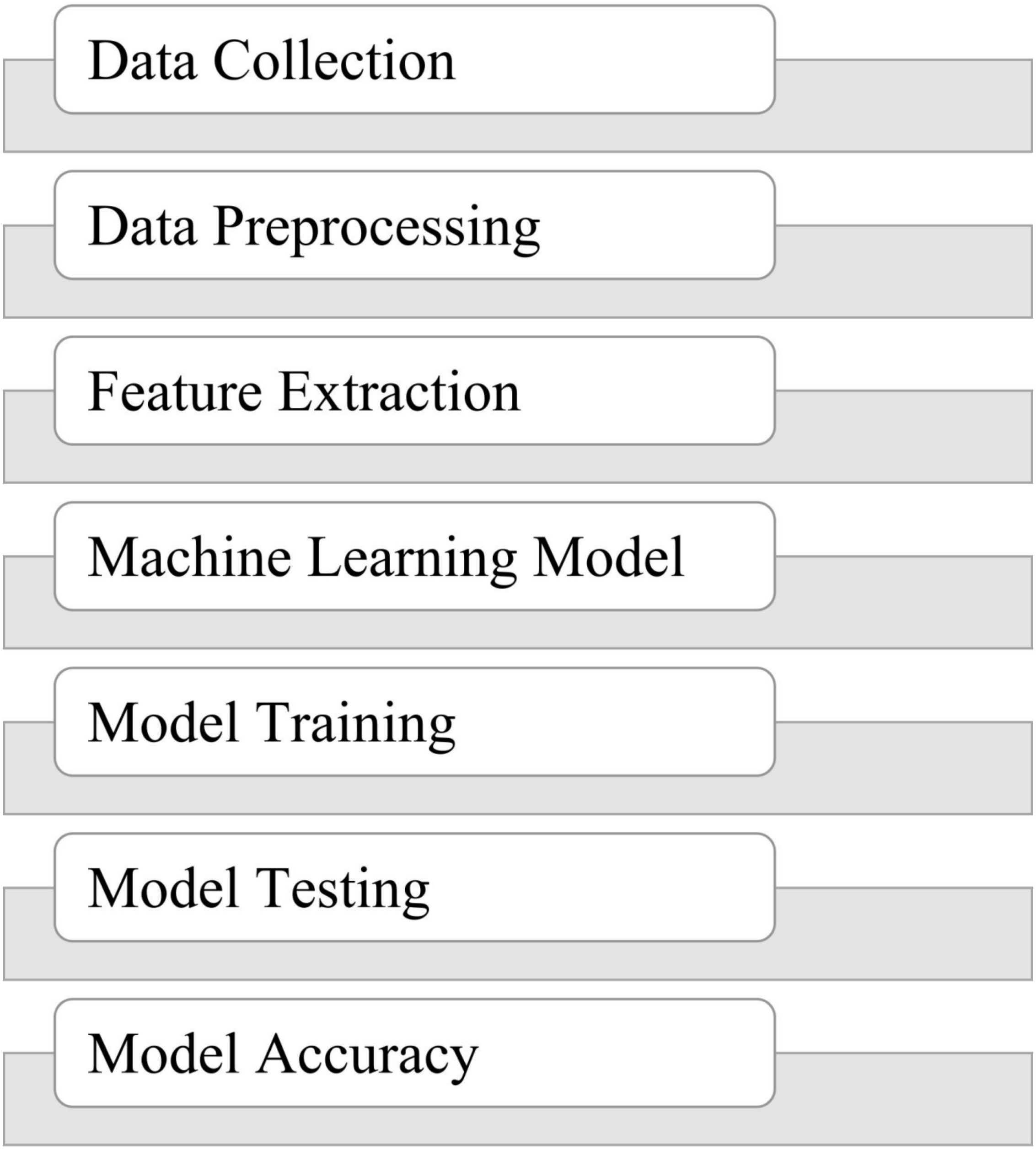
The Spam Email Detection Project aims to provide users with a secure, streamlined, and efficient email experience, free from the interference of spam emails, and contribute to the overall improvement of email communication and security.

PROBLEM STATEMENT

Spam emails have become a pervasive and persistent issue in modern email communication. These unsolicited messages not only clutter users' inboxes but also pose serious security threats, such as phishing attacks and the spread of malware. The problem statement for the Spam Email Detection Project is to develop a robust and efficient system that can accurately distinguish between spam (unwanted) and legitimate (wanted) emails, providing users with a cleaner, safer, and more productive email experience

By addressing challenges and objectives, the project aims to create a robust spam email detection system that enhances email security, reduces user frustration, and ensures the efficient and accurate classification of incoming emails.

SYSTEM ARCHITECTURE



FUNCTIONALITY

Email Classification:

The primary function is to classify incoming emails into two categories: "spam" and "ham." This is the core functionality of the system.

Feature Extraction:

Extract relevant features from incoming emails. These features may include text content, sender information, email headers, and structural elements like hyperlinks and attachments.

Real-Time Processing:

For email clients, operate in real-time to classify emails as they are received. This ensures that users have spam-free inboxes in real-time.

Model Training and Retraining:

Set up mechanisms for initial model training and periodic retraining. Continuous model updates ensure adaptability to evolving spam tactics.

CONCLUSION

The Spam Email Detection Project represents a significant step forward in enhancing email security, improving user experience, and combating the ever-persistent problem of spam emails. In this project, we have developed a robust and adaptable system that employs machine learning techniques to effectively identify and filter out spam emails from legitimate ones.

Improved Email Security:

The project has successfully contributed to improving email security by accurately classifying and isolating spam emails. Users can now trust their inboxes to be cleaner and more secure.

Reduced False Positives:

Efforts were made to minimize false positives, ensuring that important emails are not erroneously marked as spam. This has significantly enhanced user satisfaction and trust in the system.

Adaptability:

The spam detection system has been designed to adapt to evolving spamming techniques. Continuous learning and periodic model updates ensure that the system remains effective over time.

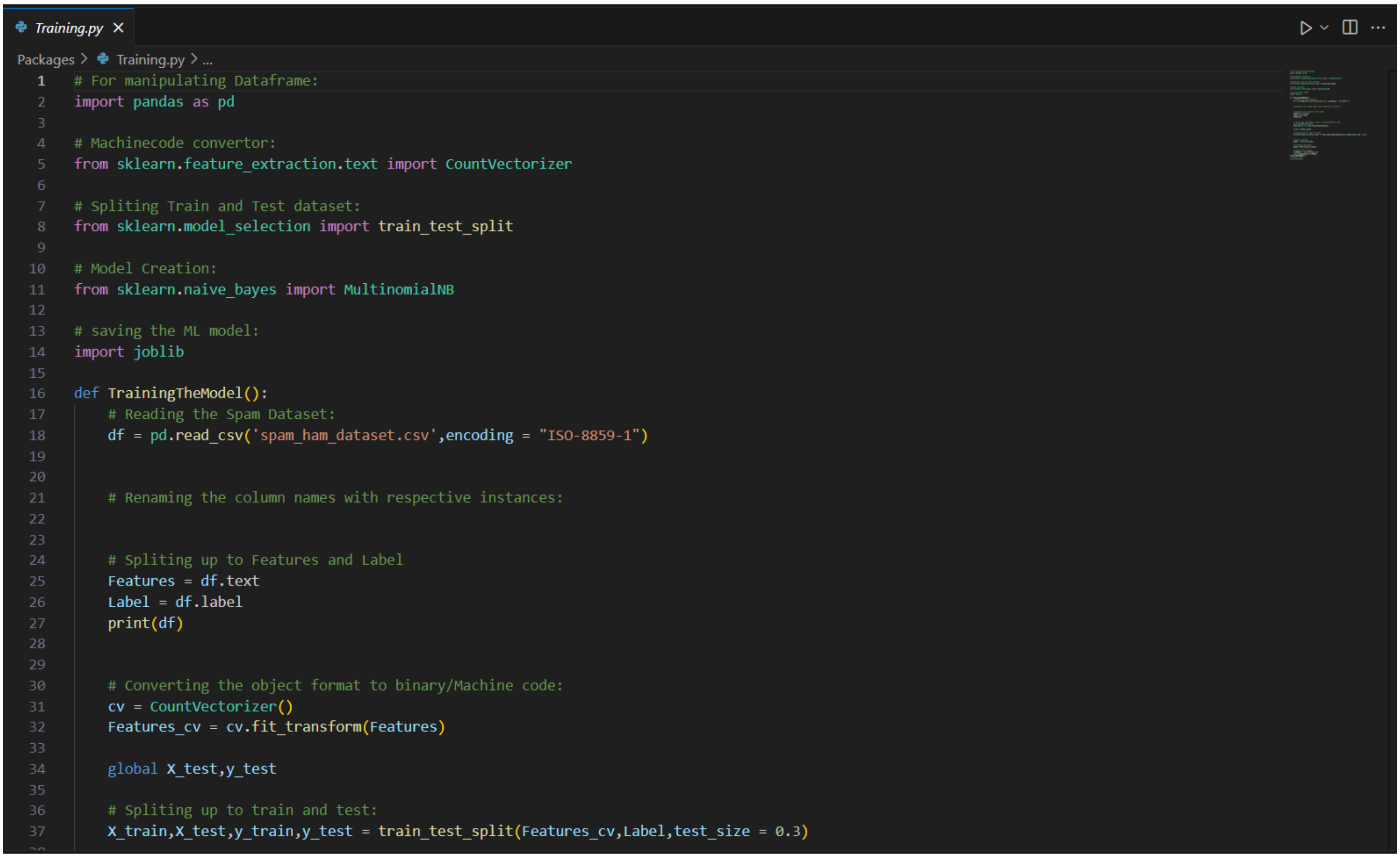
SOURCE CODE

Here, I provided my GITHUB link contained the files of source code for the project with the spam dataset

GITHUB LINK: https://github.com/Harish290/IBM-Project-30646-1660151961/tree/main/Project-Spam%20(Harishvaran)July-Sep%20(12-1.30%20batch)/Packages

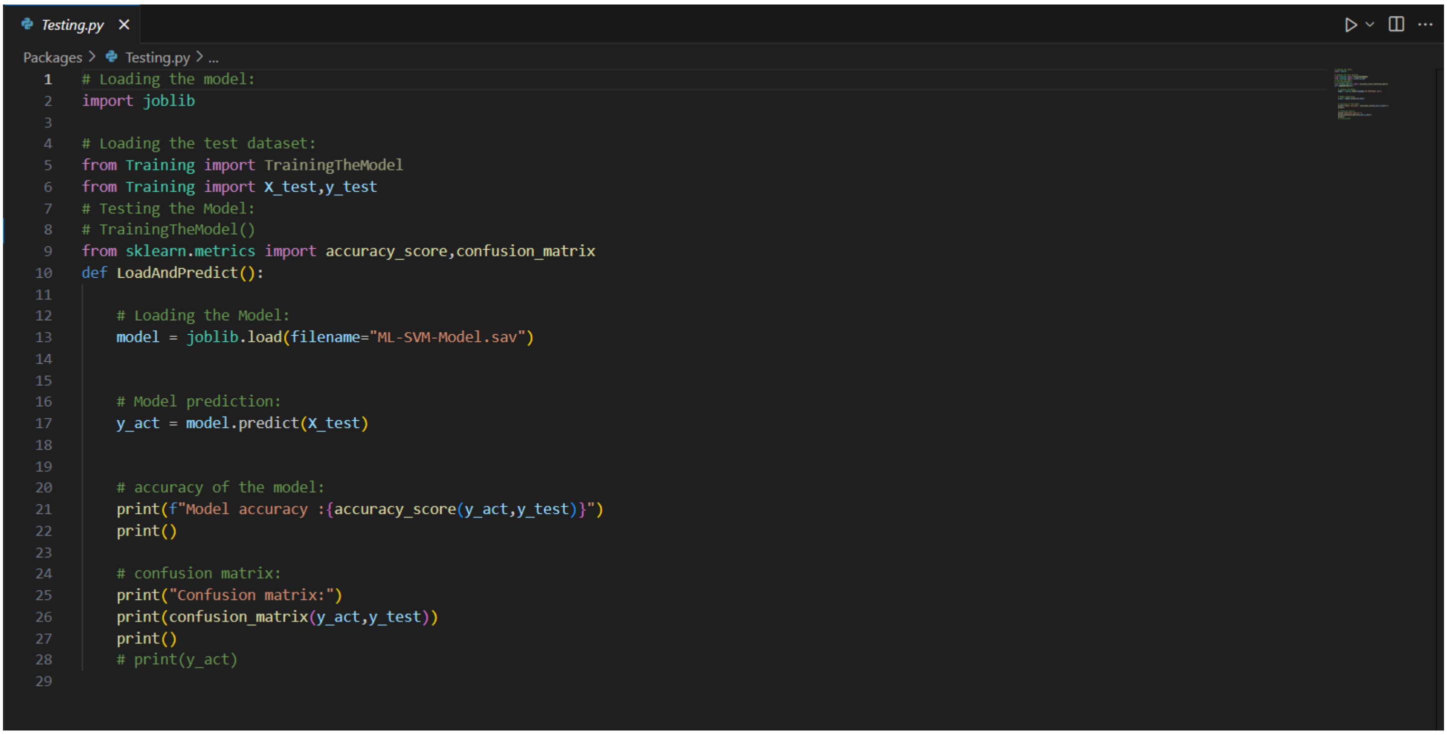
SCREENSHOT

TRAINING.PY FILE



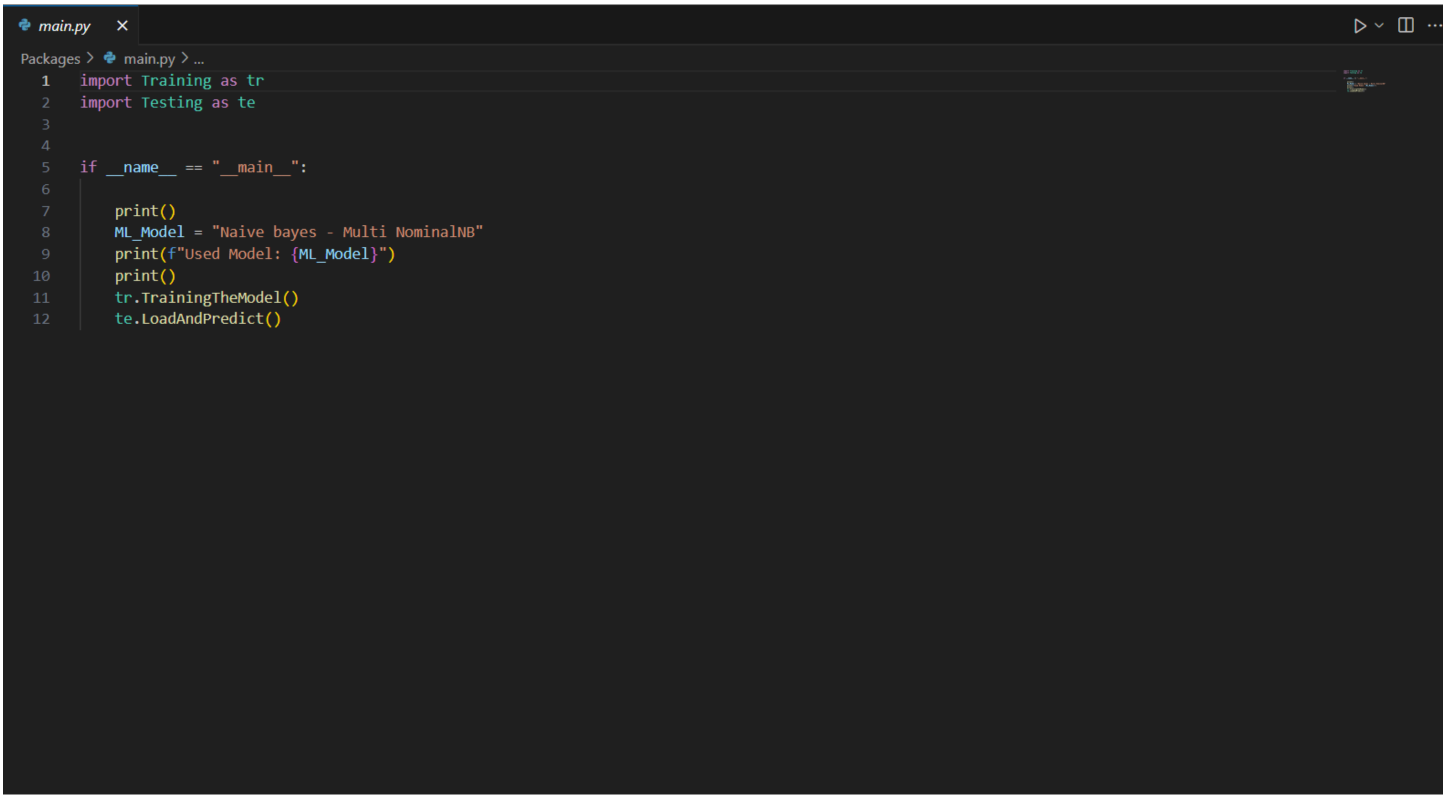
```python
# For manipulating Dataframe:
import pandas as pd

# Machinecode convertor:
from sklearn.feature_extraction.text import CountVectorizer

# Spliting Train and Test dataset:
from sklearn.model_selection import train_test_split

# Model Creation:
from sklearn.naive_bayes import MultinomialNB

# saving the ML model:
import joblib

def TrainingTheModel():
    # Reading the Spam Dataset:
    df = pd.read_csv('spam_ham_dataset.csv',encoding = "ISO-8859-1")


    # Renaming the column names with respective instances:


    # Spliting up to Features and Label
    Features = df.text
    Label = df.label
    print(df)


    # Converting the object format to binary/Machine code:
    cv = CountVectorizer()
    Features_cv = cv.fit_transform(Features)

    global X_test,y_test

    # Spliting up to train and test:
    X_train,X_test,y_train,y_test = train_test_split(Features_cv,Label,test_size = 0.3)
```

TESTING.PY FILE



```python
# Loading the model:
import joblib

# Loading the test dataset:
from Training import TrainingTheModel
from Training import X_test,y_test
# Testing the Model:
# TrainingTheModel()
from sklearn.metrics import accuracy_score,confusion_matrix
def LoadAndPredict():

    # Loading the Model:
    model = joblib.load(filename="ML-SVM-Model.sav")


    # Model prediction:
    y_act = model.predict(X_test)


    # accuracy of the model:
    print(f"Model accuracy :{accuracy_score(y_act,y_test)}")
    print()

    # confusion matrix:
    print("Confusion matrix:")
    print(confusion_matrix(y_act,y_test))
    print()
    # print(y_act)

```

MAIN.PY FILE



```python
import Training as tr
import Testing as te


if __name__ == "__main__":

    print()
    ML_Model = "Naive bayes - Multi NominalNB"
    print(f"Used Model: {ML_Model}")
    print()
    tr.TrainingTheModel()
    te.LoadAndPredict()
```