BUILDING A SMARTER AI-POWERED SPAM CLASSIFIER

NAME:RAHULDASS S

COLLEGE CODE:4222

EXAM NO:422221104028

Abstract:- Spam Classification using Artificial Intelligence - For business purposes, email is the most widely utilized mode of official communication. Despite the availability of other forms of communication, email usage continues to rise. In today's world, automated email management is critical since the volume of emails grows by the day. More than 55 percent of all emails have been recognized as spam. This demonstrates that spammers waste email users' time and resources while producing no meaningful results. Spammers employ sophisticated and inventive strategies to carry out their criminal actions via spam emails. As a result, it is critical to comprehend the many spam email classification tactics and mechanisms. The main focus of this paper is on spam classification using machine learning algorithms properties used in various Machine Learning approaches.

Introduction: -

For the majority of internet users, email has become the most often utilized formal communication channel. In recent years, there has been a surge in email usage, which has exacerbated the problems presented by spam emails. Spam, often known as junk email, is the act of sending unsolicited mass messages to a large number of people. 'Ham' refers to emails that are meaningful but of a different type. Every day, the average email user receives roughly 40-50 emails. Spammers earn roughly 3.5 million dollars per year from spam, resulting in financial damages on both a personal and institutional level. As a result, consumers devote a large amount of their working time to these emails. Spam is said to account for more than half of all email server traffic, sending out a vast volume of undesired and uninvited bulk emails.

The majority of people in today's society own a mobile phone, and they all frequently get communications (SMS/email) on their phones. But the key point is that some of the messages you get may be spam, with very few being genuine or important interactions. You may be tricked into providing your personal information, such as your password, account number, or Social Security number, by scammers that send out phony text messages. They may be able to access your bank, email, and other accounts if they obtain this

information. To filter out these messages, a spam filtering system is used that marks a message spam on the basis of its contents or sender.

Steps to implement Spam Classification using OpenAI

Now there are two approaches that we will be covering in this article:

1. Using Embeddings API developed by OpenAI

Step 1: Install all the necessary salaries

Python:-

```
!pip install -q openai
```

Step 2: Import all the required libraries

Python:-

```
# necessary libraries
import openai
import pandas as pd
import numpy as np
# libraries to develop and evaluate a machine learning model
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
from sklearn.metrics import confusion_matrix
```

Step 3: Assign your API key to the OpenAI environment

Python:-

```
# replace "YOUR API KEY" with your generated API key
openai.api_key = "YOUR API KEY"
```

Step 4: Read the CSV file and clean the dataset

Our dataset has 3 unnamed columns with NULL values,

Note: Open AI's public API does not process more than 60 requests per minute. so we will drop them and we are taking only 60 records here only.

Python:-

```
# while loading the csv, we ignore any encoding errors and skip any bad line
df = pd.read_csv('spam.csv', encoding_errors='ignore', on_bad_lines='skip')
print(df.shape)
# we have 3 columns with NULL values, to remove that we use the below line
df = df.dropna(axis=1)
# we are taking only the first 60 rows for developing the model
df = df.iloc[:60]
# rename the columns v1 and v2 to Output and Text respectively
df.rename(columns = {'v1':'OUTPUT', 'v2': 'TEXT'}, inplace = True)
print(df.shape)
df.head()
```

Output:

	OUTPUT	TEXT	embedding
0	ham	Go until jurong point, crazy.. Available only ...	[-0.011956056579947472, -0.026185495778918266,...
1	ham	Ok lar... Joking wif u oni...	[-0.0024703105445951223, -0.0312176700681448, ...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	[-0.008984447456896305, 0.0006775223882868886,...
3	ham	U dun say so early hor... U c already then say...	[0.010833987966179848, -0.011291580274701118, ...
4	ham	Nah I don't think he goes to usf, he lives aro...	[0.012792329303920269, -1.7723063137964346e-05...

Step 6: Custom Label the classes of the output variable to 1 and 0, where 1 means "spam" and 0 means "not spam".

Python:-

```
class_dict = {'spam': 1, 'ham': 0}
df['class_embeddings'] = df.OUTPUT.map(class_dict)
df.head()
```

Output:

	OUTPUT	TEXT	embedding	class_embeddings
0	ham	Go until jurong point, crazy.. Available only ...	[-0.011956056579947472, -0.026185495778918266,...	0
1	ham	Ok lar... Joking wif u oni...	[-0.0024703105445951223, -0.0312176700681448, ...	0
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	[-0.008984447456896305, 0.0006775223882868886,...	1
3	ham	U dun say so early hor... U c already then say...	[0.010833987966179848, -0.011291580274701118, ...	0
4	ham	Nah I don't think he goes to usf, he lives aro...	[0.012792329303920269, -1.7723063137964346e-05...	0

Step 7: Develop a Classification model.

We will be splitting the dataset into a training set and validation dataset using train_test_split and training a Random Forest Classification model.

Python:-

```
# split data into train and test

X = np.array(df.embedding)
y = np.array(df.class_embeddings)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# train random forest classifier
clf = RandomForestClassifier(n_estimators=100)
clf.fit(X_train.tolist(), y_train)
preds = clf.predict(X_test.tolist())

# generate a classification report involving f1-score, recall, precision and
accuracy
report = classification_report(y_test, preds)
print(report)
```

Output:

```
precision    recall  f1-score   support

           0       0.82      1.00      0.90         9
           1       1.00      0.33      0.50         3
    accuracy                           0.83        12
   macro avg       0.91      0.67      0.70        12
weighted avg       0.86      0.83      0.80        12
```

Step 8: Calculate the accuracy of the model

Python:-

```
print("accuracy: ", np.round(accuracy_score(y_test, preds)*100,2), "%")
```

Output:

```
accuracy:  83.33 %
```

Step 9: Print the confusion matrix for our classification model

Python:-

```
    w confusion_matrix(y_test, preds)
```

Output:

```
array([[9, 0],
       [2, 1]])
```

- **Conclusion: –**

Following a thorough examination of the chosen study, Several study findings and observations have been identified as a result of our studies. These were previously discussed in detail.

portions that are well-explained In this section, we'll talk about concentrating more on the major findings and conclusions of the research Supervised machine learning has a high acceptance rate. Throughout the review, the approach can be noticed. This strategy is effective. is employed primarily because it produces more accurate findings. With less fluctuation, this strategy has a high level of consistency. Aside from that, we've discovered that certain algorithms work better than others. When compared to other techniques, such as Nave Based and SVM, there is a strong demand for them. Machine Learning Algorithms that aren't as well-known. The employed multi-algorithm. n order to achieve a better result, systems are increasingly commonly used. rather than a single algorithm