Building a smarter AI powered spam classifier

Final project

Introduction:

- Building a smarter AI-powered spam classifier is a crucial endeavor in the age of information overload. As the volume of digital communication continues to surge, the need for efficient and accurate spam filters becomes increasingly vital. In this pursuit, we aim to develop an advanced AI system that can discern between legitimate and unsolicited content with unprecedented precision. By harnessing the power of cutting-edge machine learning algorithms and data-driven insights, we are dedicated to creating a solution that not only reduces the nuisance of spam but also adapts and evolves to stay ahead of the ever-changing tactics employed by spammers.

Abstract

- Now a days, email are used in almost every field, from business to education. Emails have two subcategories, i.e., ham and spam. Email spam, also called junk emails or unwanted emails, is a type of email that can be used to harm any user by wasting his/her time, computing resources, and stealing valuable information. The ratio of spam emails is increasing rapidly day by day. Spam detection and filtration are significant and enormous problems for email. Among all,these techniques are developed for detecting and preventing spam, filtering email is one of the most essential and prominent approaches.

Existing model of the system

- Spam refers to the term, which is related to undesired content with low-quality information, Spam referred to the major drawback of mobile business. When comes to spam detection in the campus network they did the analysis using Incremental Learning. For Collecting Spam detection on web pages. Moreover Sending out a Spam message was also analyzed. Data Collection was done privately by a limited company. From the data Collection. There also anti-spam filter system was evolved. Many parallel and distributed computing system has also processed this spam system. Machine learning algorithm provides accurate result. Text Mining analysis done separates ham and spam separately.

Innovations

- Behavioral Analysis: Instead of solely relying on keyword or pattern matching, AI could analyze user behavior to detect anomalies. For instance, sudden bursts of activity or unusual login locations could signal spam. Deep Learning for Image and Audio Spam: Extend AI spam detection to multimedia content. AI models could analyze images and audio within messages to identify spam content, such as inappropriate images or voice phishing attempts. AI could differentiate between genuine conversations and spam by assessing the relevance and coherence of the content. User Feedback Loops: Implement a feedback mechanism where users can report false positives/negatives. AI could learn from this feedback to continuously improve spam detection algorithms.

Loading and preprocessing

- Text Preprocessing:Text Tokenization: Split the text into individual words or tokens. Lowercasing: Convert all text to lowercase to ensure consistency.Removing Punctuation: Eliminate punctuation marks that don't carry significant meaning.Stopword Removal: Exclude common words (e.g., "and," "the," "in") that are unlikely to help classify spam. Stemming or Lemmatization: Reduce words to their base or root form to handle variations (e.g., "running" to "run").Word Embeddings: Use pre-trained word vectors like Word2Vec or GloVe to capture semantic meaning.Split the Dataset into training, validation, and test sets to evaluate your model's performance.

- Certainly!selecting a machine learning algorithm, training the model, andevaluating its performance are crucial steps in building a machine learning model.

- 1. *Selecting a Machine Learning Algorithm*:
- - Start by understanding your problem and data. Is it a classification,
- regression, or clustering problem?
- - Consider the nature of your data: Is it structured or unstructured? How many features do you have?
- - Choose algorithms based on the problem type (e.g., decision trees for classification, linear regression for regression).

Experiment with different algorithms and fine-tune hyperparameters to see which one works best.

2. *Training the Model*:

- Preprocess the data: Handle missing values, scale or normalize features, encode categorical variables.

- Split your data into training and testing sets to assess model generalization.

- Train the model using the training data, feeding features and their corresponding labels.

- Monitor the training process, adjusting hyperparameters if necessary.

3. *Evaluating Performance*:

- Use evaluation metrics specific to your problem (e.g., accuracy, precision, recall, F1-score for classification; RMSE, MAE for regression).

- Assess the model's performance on the testing dataset to ensure it generalizes well to new, unseen data.

- Consider using cross-validation to get a more robust estimate of your model's performance.

- Analyze any overfitting or underfitting issues and make necessary adjustments.

GIVEN DATASETS:

ham

Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got a...

ham

Ok lar... Joking wif u oni...

spam

Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entr...

ham

U dun say so early hor... U c already then say...

ham

Nah I don't think he goes to usf, he lives around here though...

ham

Nah I don't think he goes to usf, he lives around here though

spam

FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it s...

ham

Even my brother is not like to speak with me. They treat me like aids patent.

ham

As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertu...

PROGRAM:

```
# 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.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
```

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

# 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.loc[: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

```
(5572, 5)
(60, 2)
```

	OUTPUT	TEXT
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

CONCLUSION:

building a powered smarter AI spam classifier is a valuable endeavor with significant benefits. Leveraging advanced machine learning techniques, natural language processing, and a robust dataset, such a system can effectively and efficiently identify and filter out spam messages, improving user experience, reducing security risks, and enhancing communication platforms. It is essential to continually train and update the AI model to adapt to evolving spam tactics and to prioritize user privacy and data protection throughout its development and deployment.