Building a Smarter AI-Powered Spam Classifier

Phase-3 Document Submission

Project Overview

Project Name: Building a Smarter AI-Powered Spam Classifier Project

Project Phase: Phase 3 – Development Part 1

Phase Overview

In Phase 3 of the project, the data loading and preprocessing phase, our main objectives are to make the dataset ready for machine learning. We load and clean the data, convert text into numbers, and encode labels for spam and non-spam messages. These steps are vital for creating an effective spam classifier.

Code Explanation

1.Data Loading

- Data loading is a pivotal first step in developing a spam classifier. It involves sourcing the dataset containing examples of both spam and non-spam messages. This dataset is typically structured in a tabular format, with one column dedicated to the message text and another for the labels indicating whether a message is spam (unsolicited) or non-spam (ham). Upon loading the data, it is essential to conduct a preliminary examination to understand its quality, structure, and any potential issues. Once loaded into a Pandas DataFrame or a similar data structure, the data can be further cleaned, explored, and preprocessed to prepare it for model training. Ensuring the data's integrity, performing data splits, and maintaining data quality throughout the development process are integral aspects of data loading in the creation of an effective spam classifier.

Program:

```
#Import Necessary Libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')

# importing Stopwords
import nltk
from nltk.corpus import stopwords
import string

# models
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier

# train test split
from sklearn.model_selection import train_test_split

# Pipeline
from sklearn.pipeline import Pipeline

# score
from sklearn.metrics import
confusion_matrix,classification_report,ConfusionMatrixDisplay
from sklearn.metrics import accuracy_score



# Load the Dataset from Kaggle

df=pd.read_csv('spam_dataset.csv')
```

2. Data Exploration

- To understand the structure of our dataset, we display a sample of the data, including the first few rows.

Program:

```
# Explore the dataset
print("Dataset Sample:")
print(data.head())
```

3.Exploratory Data Analysis (EDA)

- Perform EDA to understand your data better. This includes checking for values, exploring the data's statistics, and visualizing it to identify patterns.

Program:

```
# Explore Data Analysis
Print(df.describe())
# Explore statistics
print(df.describe())
# Visualize the data (e.g., histograms, scatter plots, count plots etc.)
```

4.Feature Engineering

- Depending on your dataset, you may need to create new features or transform existing ones. This can involve one-hot encoding categorical variables, handling date/time data, or scaling numerical features.

Program:

```
# Example: One-hot encoding for categorical variables
df = pd.get_dummies(df, columns=['label'], prefix=['label'])
```

5. Split the Data

- Split your dataset into training and testing sets. This helps you evaluate your model's performance later.

```
X= df['text'] # Features
y=df['label'] # Target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
random_state=42)
```

6.Feature Scaling

- Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 and std=1) is a common choice.

Program:

```
pred_mnb = pipe_mnb.predict(X_test)
pred_rf = pipe_rf.predict(X_test)
```

Conclusion

In Phase 3: Development Part 1, we have successfully loaded and pre-processed the dataset., a spam classifier is pivotal in online communication, aiming to sift through spam messages while enhancing user experience and safeguarding against scams and phishing. It reduces data overload, maintains privacy, and ensures compliance with legal requirements. Through data preprocessing, model training, and evaluation, this tool provides a more secure and efficient digital communication environment.