Project Title:

Building a Smarter AI-Powered Spam Classifier

Phase 3:Development Part 1

Overview:

In this phase, we're setting up the groundwork for our AI-Powered Spam Classifier. We start by getting the dataset, understanding what's in it, and then getting it in the right shape. This means cleaning it up by removing duplicates, handling missing data, and making sure the text data is ready for analysis. We also split the dataset for future model development and testing. This step is essential to make sure our spam classifier will work effectively.

Data Loading:-

Data loading is the process of copying and loading data from a source file, folder, or application to a database or similar application. It can also involve converting data from one format to another.

Data loading can involve:

- ✓ Copying data from a source
- ✓ Loading the data into a data storage or processing utility
- ✓ Converting data from one format to another

Extracting data from various sources:

- ✓ Transforming the data to fit the requirements of the target system
- ✓ Loading the data into the target system

Data Preprocessing:-

Data preprocessing is the process of converting raw data into a format that is understandable and usable. It is a crucial step in any Data Science project to carry out an efficient and accurate analysis.

Data preprocessing involves:

- ✓ Cleaning, transforming, and integrating data
- ✓ Checking for missing values, noisy data, and other inconsistencies
- ✓ Improving the quality of the data
- ✓ Making the data more suitable for the specific data mining task

Exploratory Data Analysis(EDA):-

Exploratory data analysis (EDA) is a technique that analyzes and investigates a dataset to summarize its main characteristics. EDA is an important step in any data analysis or data science project.

EDA helps you:

- ✓ Discover patterns
- ✓ Spot anomalies
- ✓ Test hypotheses
- ✓ Check assumptions
- ✓ Identify relationships between variables
- ✓ Understand the data in depth
- ✓ Learn the different data characteristics
- ✓ Form hypotheses based on your understanding of the dataset.

Data Splitting:-

Data splitting is the process of dividing data into two or more subsets. It's an important part of data science, especially for creating models based on data.

Data splitting is typically used to:

- ✓ Train a model
- ✓ Evaluate or test data
- ✓ Detect underfitting and overfitting

There are several ways to split data, including:

- ✓ Two splits: One for training and one for testing
- ✓ Three splits: One for training, one for testing, and one for validation

Coding:

```
import pandas as pd

import matplotlib.pyplot as plt

import nltk


# Download the NLTK stopwords and 'punkt' resources

nltk.download('stopwords')

nltk.download('punkt')


# 1. Data Collection - Download the Kaggle dataset and save it to your local environment.

# Assuming the dataset is saved as 'spam1' in the working directory.


# 2. Loading the Dataset

data = pd.read_csv('C:\\Users\\BYAMUNA1\\Downloads\\archive\\spam.csv', encoding='latin-1')


# 3. Exploratory Data Analysis (EDA)

# Distribution of spam vs. non-spam messages

spam_counts = data['v1'].value_counts()

plt.bar(spam_counts.index, spam_counts.values)

plt.xlabel('Label')

plt.ylabel('Count')

plt.title('Distribution of Spam vs. Non-Spam Messages')

plt.show()
```

```
#Message Length Analysis
data['message_length'] = data['v2'].apply(lambda x: len(x))
data.groupby('v1')['message_length'].plot(kind='hist', alpha=0.5, legend=True)
plt.xlabel('Message Length')
plt.title('Message Length Distribution by Label')
plt.show()

# 4. Data Preprocessing
# Remove duplicates and lowercase the text
data = data.drop_duplicates(keep='first')
data.loc[:, 'v2'] = data['v2'].str.lower()

# 5. Basic Data Processing Methods
# Display the first few rows of the dataset
print("\nBasic Data Processing Methods:\n")
print("First 5 rows of the dataset:\n")
print(data.head())

# Display basic information about the dataset
print("\nDataset Information:\n")
print(data.info())

# Display summary statistics of the dataset
print("\nSummary Statistics:\n")
print(data.describe())
```

 # 6. Data Splitting - Split the dataset into training and testing sets

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(data['v2'], data['v1'], test_size=0.2, random_state=42)
```

 # Print the shapes of the training and testing sets

```
print("\nShapes of the training and testing sets\n")

print("\nTraining set shape:", X_train.shape, y_train.shape)

print("\nTesting set shape:", X_test.shape, y_test.shape)
```

 # 7. Save Preprocessed Data - Save the preprocessed data for future use.

```
data.to_csv('preprocessed_sms_data.csv', index=False)
```

Program:-

spam1'.py - C:/Users/BYAMUNA1/Desktop/spam1'.py (3.12.0)

File Edit Format Run Options Window Help

```
import pandas as pd
import matplotlib.pyplot as plt
import nltk

# Download the NLTK stopwords and 'punkt' resources
nltk.download('stopwords')
nltk.download('punkt')

# 1. Data Collection - Download the Kaggle dataset and save it to your local environment.
# Assuming the dataset is saved as 'spam1' in the working directory.

# 2. Loading the Dataset
data = pd.read_csv('C:\\Users\\BYAMUNA1\\Downloads\\archive\\spam.csv', encoding='latin-1')

# 3. Exploratory Data Analysis (EDA)
# Distribution of spam vs. non-spam messages
spam_counts = data['v1'].value_counts()
plt.bar(spam_counts.index, spam_counts.values)
plt.xlabel('Label')
plt.ylabel('Count')
plt.title('Distribution of Spam vs. Non-Spam Messages')
plt.show()

#Message Length Analysis
data['message_length'] = data['v2'].apply(lambda x: len(x))
data.groupby('v1')['message_length'].plot(kind='hist', alpha=0.5, legend=True)
plt.xlabel('Message Length')
plt.title('Message Length Distribution by Label')
plt.show()

# 4. Data Preprocessing
# Remove duplicates and lowercase the text
data = data.drop_duplicates(keep='first')
data.loc[:, 'v2'] = data['v2'].str.lower()

# 5. Basic Data Processing Methods
# Display the first few rows of the dataset
print("\nBasic Data Processing Methods:\n")
print("First 5 rows of the dataset:\n")
print(data.head())
```

Ln: 47 Col: 32

Search 10:41 17-10-2023

spam1'.py - C:/Users/BYAMUNA1/Desktop/spam1'.py (3.12.0)

File Edit Format Run Options Window Help

```
#Message Length Analysis
data['message_length'] = data['v2'].apply(lambda x: len(x))
data.groupby('v1')['message_length'].plot(kind='hist', alpha=0.5, legend=True)
plt.xlabel('Message Length')
plt.title('Message Length Distribution by Label')
plt.show()

# 4. Data Preprocessing
# Remove duplicates and lowercase the text
data = data.drop_duplicates(keep='first')
data.loc[:, 'v2'] = data['v2'].str.lower()

# 5. Basic Data Processing Methods
# Display the first few rows of the dataset
print("\nBasic Data Processing Methods:\n")
print("First 5 rows of the dataset:\n")
print(data.head())

# Display basic information about the dataset
print("\nDataset Information:\n")
print(data.info())

# Display summary statistics of the dataset
print("\nSummary Statistics:\n")
print(data.describe())

# 6. Data Splitting - Split the dataset into training and testing sets for Phase 4.
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(data['v2'], data['v1'], test_size=0.2, random_state=42)

# Print the shapes of the training and testing sets
print("\nShapes of the training and testing sets\n")
print("\nTraining set shape:", X_train.shape, y_train.shape)
print("\nTesting set shape:", X_test.shape, y_test.shape)

# 7. Save Preprocessed Data - Save the preprocessed data for future use.
data.to_csv('preprocessed_sms_data.csv', index=False)
```

Ln: 61 Col: 53

Search 10:42 17-10-2023

Program Explanation:-

Here's a step-by-step explanation of what the program does:

1. **Import necessary libraries:** The code imports the `pandas`, `matplotlib`, and `nltk` libraries.

2. **Download NLTK resources:** The NLTK stopwords and 'punkt' resources are downloaded using the `nltk.download` function. These resources are essential for text processing.

3. **Data Collection:** The program assumes that you've already downloaded a Kaggle dataset (presumably related to spam classification) and saved it as 'spam.csv' in your working directory.

4. **Loading the Dataset:** It reads the dataset from the specified file path using `pd.read_csv` and encodes it using 'latin-1'.

5. **Exploratory Data Analysis (EDA):** This section of the code contains two main parts:

   - ✓ Distribution of spam vs. non-spam messages: It calculates and plots the distribution of spam and non-spam messages using a bar plot.
   - ✓ Message Length Analysis: It calculates and plots the distribution of message lengths for both spam and non-spam messages using histograms.

6. **Data Preprocessing:** In this step, the program performs data preprocessing tasks:

   - ✓ Remove duplicates: It removes duplicate rows from the dataset.
   - ✓ Lowercase text: It converts the text in the 'v2' column to lowercase.

7. **Basic Data Processing Methods:** This section displays the first five rows of the preprocessed dataset using `data.head()` and provides basic information about the dataset using `data.info()`. It also displays summary statistics of the dataset using `data.describe()`.

8. **Data Splitting:** The program splits the dataset into training and testing sets using `train_test_split` from the `sklearn.model_selection` library. The training set contains 80% of the data, and the testing set contains 20%. The shapes of the training and testing sets are printed to confirm the split.

This script mainly focuses on data preprocessing and initial data analysis steps, preparing the data for further machine learning tasks like text classification (e.g., spam detection) using natural language processing (NLP) techniques.

Output:

Basic Data Processing Methods:

First 5 rows of the dataset:

```
    v1  ... Unnamed: 4
0   ham  ...       NaN
1   ham  ...       NaN
2  spam  ...       NaN
3   ham  ...       NaN
4   ham  ...       NaN

[5 rows x 5 columns]
```

Dataset Information:

```
<class 'pandas.core.frame.DataFrame'>
Index: 5169 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   v1          5169 non-null   object
 1   v2          5169 non-null   object
 2   Unnamed: 2  43 non-null     object
 3   Unnamed: 3  10 non-null     object
 4   Unnamed: 4  5 non-null      object
dtypes: object(5)
memory usage: 242.3+ KB
None
```

Summary Statistics:

	v1	...	Unnamed: 4
count	5169	...	5
unique	2	...	5
top	ham	...	just Keep-in-touch\" gdeve.."
freq	4516	...	1

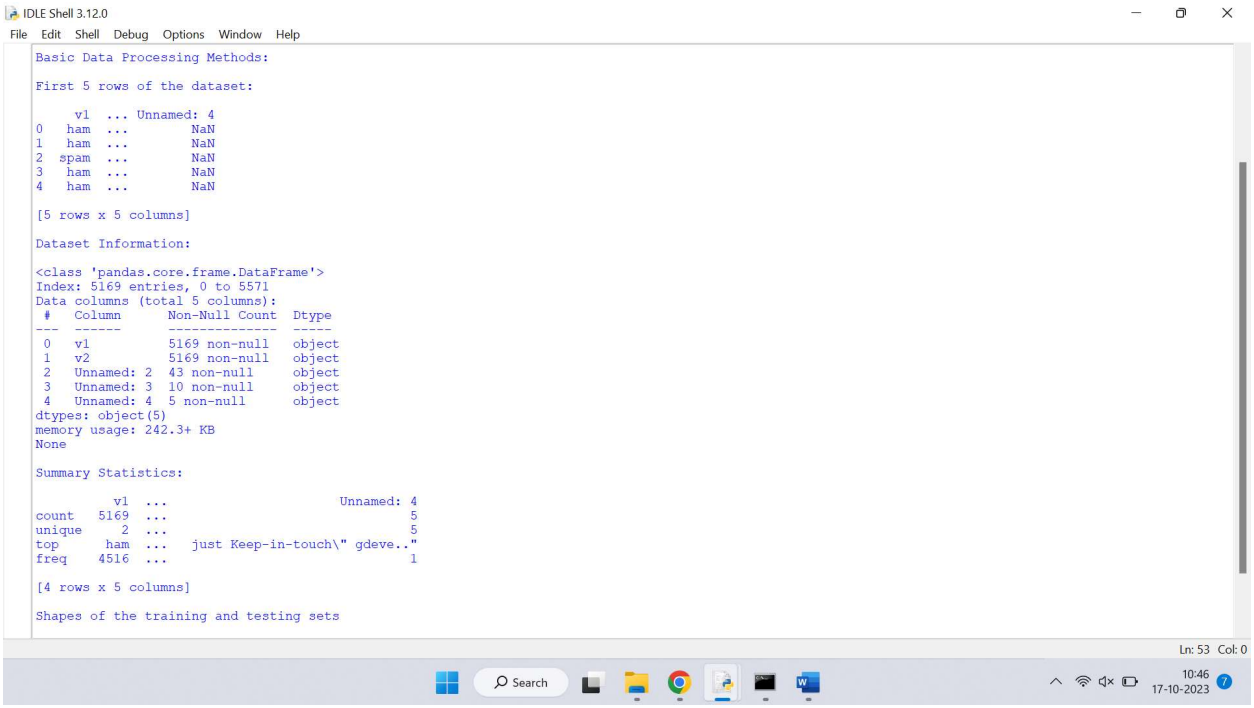
[4 rows x 5 columns]

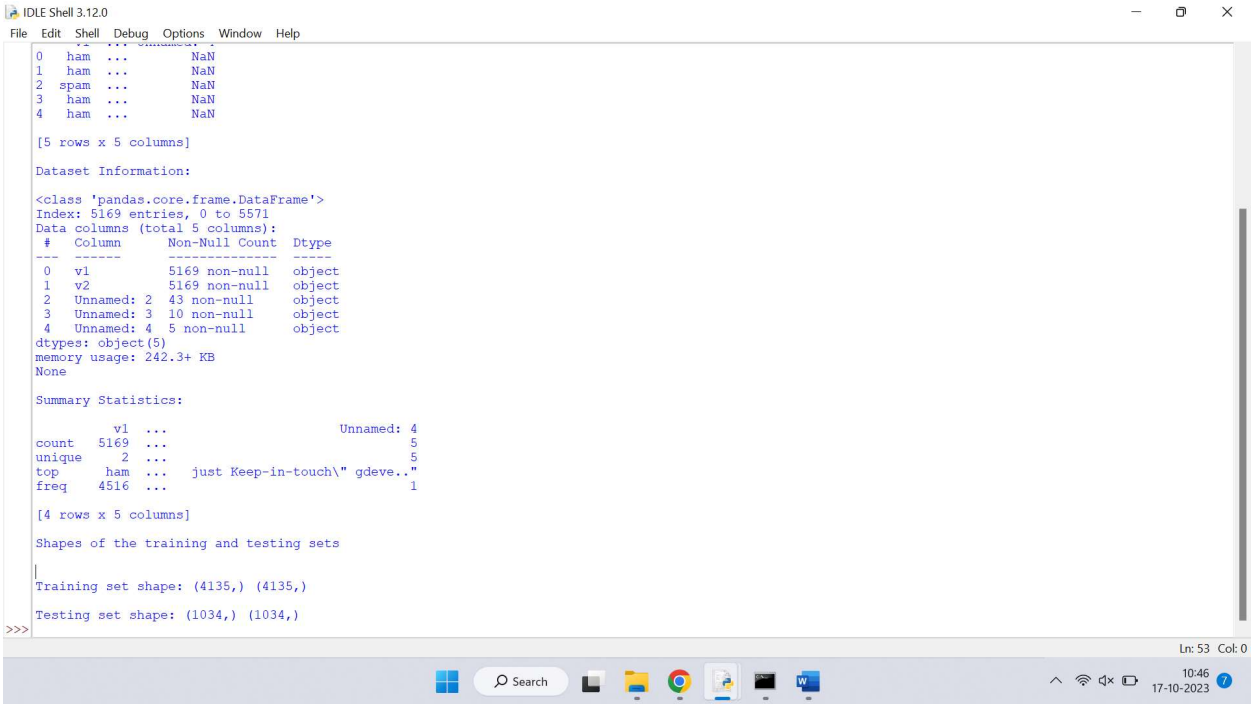
Shapes of the training and testing sets

Training set shape: (4135,) (4135,)

Testing set shape: (1034,) (1034,)

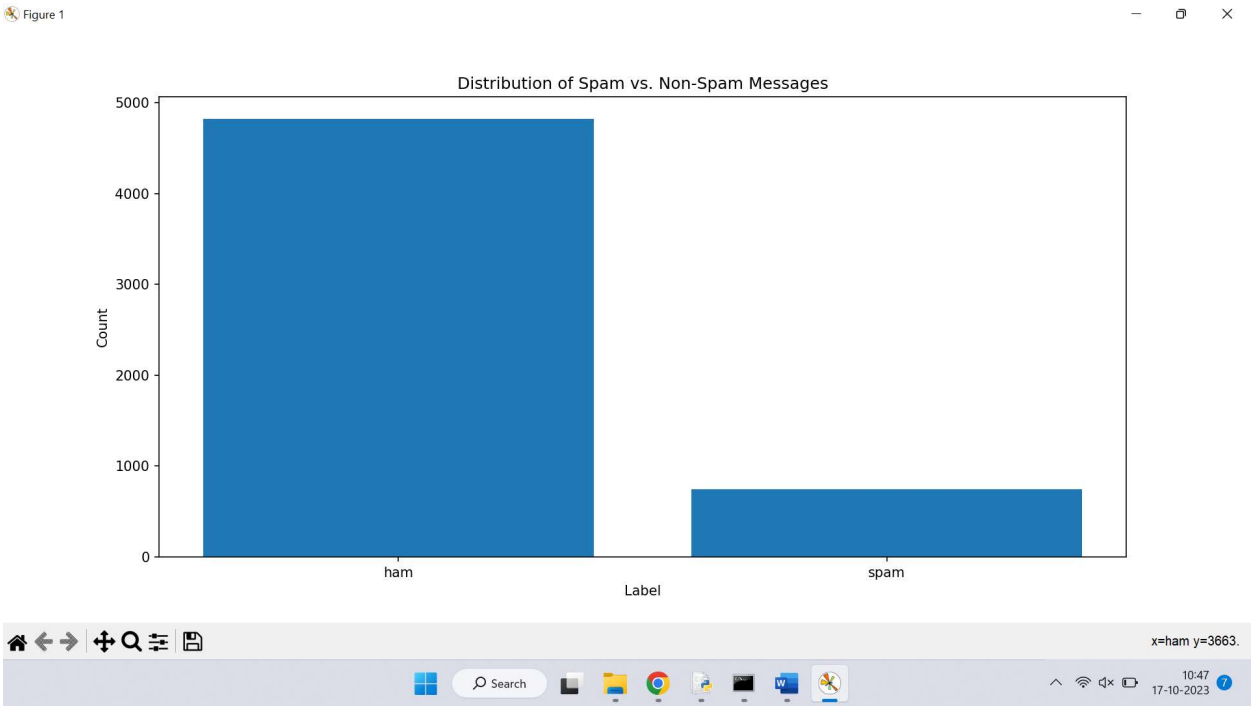
Output:-


IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help
Basic Data Processing Methods:
First 5 rows of the dataset:
v1 ... Unnamed: 4
0 ham ... NaN
1 ham ... NaN
2 spam ... NaN
3 ham ... NaN
4 ham ... NaN
[5 rows x 5 columns]
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
Index: 5169 entries, 0 to 5571
Data columns (total 5 columns):
Column Non-Null Count Dtype
0 v1 5169 non-null object
1 v2 5169 non-null object
2 Unnamed: 2 43 non-null object
3 Unnamed: 3 10 non-null object
4 Unnamed: 4 5 non-null object
dtypes: object(5)
memory usage: 242.3+ KB
None
Summary Statistics:
v1 ... Unnamed: 4
count 5169 ... 5
unique 2 ... 5
top ham ... just Keep-in-touch\" gdeve.."
freq 4516 ... 1
[4 rows x 5 columns]
Shapes of the training and testing sets
Ln: 53 Col: 0
Search
10:46
17-10-2023


IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help
0 ham ... NaN
1 ham ... NaN
2 spam ... NaN
3 ham ... NaN
4 ham ... NaN
[5 rows x 5 columns]
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
Index: 5169 entries, 0 to 5571
Data columns (total 5 columns):
Column Non-Null Count Dtype
0 v1 5169 non-null object
1 v2 5169 non-null object
2 Unnamed: 2 43 non-null object
3 Unnamed: 3 10 non-null object
4 Unnamed: 4 5 non-null object
dtypes: object(5)
memory usage: 242.3+ KB
None
Summary Statistics:
v1 ... Unnamed: 4
count 5169 ... 5
unique 2 ... 5
top ham ... just Keep-in-touch\" gdeve.."
freq 4516 ... 1
[4 rows x 5 columns]
Shapes of the training and testing sets
Training set shape: (4135,) (4135,)
Testing set shape: (1034,) (1034,)
>>>
Ln: 53 Col: 0
Search
10:46
17-10-2023

Exploratory Data Analysis (EDA):-

Distribution of spam vs. non-spam messages



Message length analysis:-