Email Spam Filtering Project 📨

In this project, I developed an effective **Email Spam Filtering** system using natural language processing (NLP) techniques and machine learning algorithms. The primary objective was to classify email messages as either spam or ham (non-spam) based on their textual content.

Step 1: Import Libraries 📚

I started by importing essential libraries such as:

- **Pandas** and **NumPy** for data manipulation.
- **Matplotlib** and **Seaborn** for data visualization.
- **Sklearn** for machine learning tools including **TfidfVectorizer** and **MultinomialNB**.
- **NLTK** for natural language processing tasks such as tokenization and stemming.

These libraries were crucial for efficient data handling, preprocessing, and model building.

Step 2: Load and Inspect the Dataset 📂

I loaded the dataset using **Pandas** from a CSV file containing 5572 email messages, which included labels for spam and ham. Upon inspecting the dataset, I ensured the integrity of the data by checking for missing values and exploring the data types.

Step 3: Rename Columns and Basic Data Exploration 🔍

After loading the data, I renamed the columns for clarity and conducted basic exploratory data analysis (EDA). I assessed the distribution of classes and visualized the class distribution using bar plots and pie charts. This step highlighted that the dataset contained 4825 ham messages and 747 spam messages.

Step 4: Text Preprocessing ✍️

To prepare the text data for analysis, I developed a text preprocessing function that included:

- **Tokenization**: Breaking text into individual words.
- **Lowercasing**: Standardizing the text to lowercase.
- **Stopword Removal**: Filtering out common words that do not contribute meaning.
- **Stemming**: Reducing words to their base form using **Porter Stemmer**.

This preprocessing was vital for reducing noise in the data and improving the model's performance.

Step 5: Feature Extraction ⚙️

Using the **TF-IDF Vectorizer**, I transformed the cleaned text data into numerical features that represent the importance of words in each message. This resulted in a feature matrix with 7377 unique tokens, ready for machine learning.

Step 6: Splitting the Dataset 🔄

I split the dataset into training and testing sets, with 70% allocated for training and 30% for testing. This stratified approach ensured the model was evaluated fairly on unseen data.

Step 7: Model Training 🧠

I trained a **Multinomial Naive Bayes** model, which is well-suited for text classification tasks. The model was trained on the preprocessed feature set, capturing the relationship between the email text and its corresponding labels.

Step 8: Model Evaluation 📈

After training, I made predictions on the test set and evaluated the model's performance using the following metrics:

- **Confusion Matrix**: Visualised to understand the model's predictions against actual labels.
- **Classification Report**: Provided insights into precision, recall, and F1-score. The model achieved an **accuracy of 96.59%**, indicating high reliability in classifying emails.

Key Results:

- True Positives (Spam correctly predicted): 163
- True Negatives (Ham correctly predicted): 1452
- False Positives (Ham incorrectly predicted as Spam): 1
- False Negatives (Spam incorrectly predicted as Ham): 56

These results demonstrated the model's effectiveness, with a high precision of 0.99 for spam classification and a recall of 0.74.

Additional EDA and Visualization 🌟

To gain further insights, I created word clouds for both spam and ham messages. This visual representation helped identify common words in spam emails, providing a deeper understanding of spam characteristics.

Skills Acquired

Through this project, I enhanced my skills in:

- Data Cleaning, Data Visualisation, Text Preprocessing, Feature Engineering, Model Training, Model Evaluation

---

Hashtags

#DataScience #NLP #MachineLearning #SpamFiltering #EmailClassification #NaturalLanguageProcessing #Python #Pandas #Sklearn #DataAnalysis #DataVisualization #DeepLearning #ArtificialIntelligence #TFIDF #MultinomialNB #WordCloud #EDA #AI #BigData #TextMining