Group 4:

Name: Babar Ali

Sap Id:65731

Name: Wahid Modassar

Sap Id:65292

Submitted to: Sir Muhammad Junaid Khan

Project Report: Email Spam Detection System

1. Problem Definition

In the digital age, Email communication is widely used for personal and business purposes. However, this popularity has led to an increase in unsolicited spam messages, which can be annoying, fraudulent, or malicious (e.g., phishing attacks). Manual filtering of these messages is inefficient and impractical. Therefore, there is a critical need for an automated system capable of distinguishing between legitimate messages ("ham") and unsolicited messages ("spam") with high accuracy.

2. Dataset Description

The project utilizes the **Email Spam Collection** dataset (`spam.csv`).

- **Source:** Publicly available research dataset (often hosted on UCI Machine Learning Repository or Kaggle).
- **Size:** 5,572 email messages.
- **Structure:** The dataset consists of two columns:
    - **Category:** The label indicating the class of the message (`ham` or `spam`).
    - **Message:** The raw text content of the email.
- **Distribution:** The dataset is imbalanced, with a majority of messages being legitimate ('ham') and a smaller portion being 'spam'.

3. Objectives

The primary objectives of this project are:

1. **Text Preprocessing:** To implement a robust pipeline for cleaning and standardizing raw text data.
2. **Feature Engineering:** To convert textual data into numerical vectors suitable for machine learning algorithms.
3. **Model Development:** To train a Classification algorithm (Multinomial Naive Bayes) to predict message categories.
4. **Performance Optimization:** To achieve a classification accuracy of greater than 95%, with a specific focus on maximizing **Precision** to ensure legitimate messages are not

incorrectly flagged as spam.

4. Methodology

The project follows a standard Machine Learning pipeline:

A. Data Loading

The dataset is loaded directly from a raw GitHub URL to ensure reproducibility and ease of access without manual file uploads.

B. Data Preprocessing

Raw text is noisy and requires cleaning. The following steps are applied:

- **Lowercasing:** Converting all text to lowercase to ensure consistency.
- **Cleaning:** Removing URLs, HTML tags, special characters, and punctuation using Regular Expressions (Regex).
- **Tokenization:** Breaking sentences into individual words (tokens).
- **Stopword Removal:** Removing common words (e.g., "the", "is", "and") that add little semantic meaning.

C. Feature Extraction (Vectorization)

The cleaned text is converted into numerical data using **TF-IDF (Term Frequency-Inverse Document Frequency)**. This technique highlights words that are important to a specific message but rare across the entire dataset, which is highly effective for spam detection.

D. Model Training

- **Algorithm:** Multinomial Naive Bayes (MultinomialNB).
- **Rationale:** Naive Bayes is a probabilistic classifier that is computationally efficient and historically performs exceptionally well on text classification tasks with high-dimensional features.
- **Training Split:** The data is split into 80% training data and 20% testing data.

5. Expected Results

Based on the methodology and dataset properties, the expected outcomes are:

- **High Accuracy:** The model is expected to achieve an overall accuracy between **96% and 98%**.

- **Effective Filtering:** The system will successfully identify common spam keywords (e.g., "free", "winner", "cash", "urgent").
- **Deployment Readiness:** The final output includes a serialized model (`.pkl` file) capable of predicting the class of new, unseen messages in real-time.

GitHub Link: https://github.com/babar-a11y/AI-BS_AI-3-1-