**NLP Exercise: Spam Classification**

**Objective**

The goal of this exercise is to classify messages as either **spam** or **ham** using Natural Language Processing techniques. You will preprocess the text data, apply feature extraction methods, train a machine learning model, and evaluate its performance.

**Dataset link:-** [**Spam classification dataset link**](https://drive.google.com/file/d/1X8awBL7BvjblDLBWObCiW2xmiapLSXPC/view?usp=drive_link)

**Explanation of the Columns**

* **Class**: The label indicating whether the message is **ham** (not spam) or **spam**.
* **Message**: The actual content of the message.

**Steps to Follow**

**1. Data Loading**

1. Load the sample dataset into a Pandas DataFrame.
2. Inspect the first few rows to understand the structure of the data.
3. Check for any missing or null values and handle them appropriately.

**2. Text Preprocessing**

Preprocess the text data to make it suitable for machine learning:

1. **Lowercasing**: Convert all text to lowercase to ensure uniformity.
2. **Remove Punctuations and Special Characters**: Use regular expressions to clean the text.
3. **Stop Word Removal**: Eliminate common words (like "and", "the") that do not contribute much to the analysis.
4. **Tokenization**: Split the text into individual words or tokens.
5. **Lemmatization**: Reduce words to their base form to unify different word forms (e.g., "running" -> "run").

**3. Feature Extraction**

Convert the preprocessed text into a numerical format suitable for machine learning:

1. **Bag of Words (BoW)**: Create a matrix where each row represents a message and each column represents a word. The value in each cell is the count or frequency of the word in the respective message.
2. (Optional) **TF-IDF**: Apply Term Frequency-Inverse Document Frequency to highlight the importance of unique words in each message.

**4. Model Training**

1. Split the dataset into training and testing sets (e.g., 80% training, 20% testing).
2. Train a machine learning model:
   * algorithms: Logistic Regression, Naive Bayes, or Random Forest.
3. Fit the model to the training data using the extracted features.

**5. Prediction**

1. Use the trained model to predict whether new messages are **ham** or **spam**.
2. Store the predictions for evaluation.

**6. Evaluation**  
**Confusion Matrix**: To understand how well the model correctly identifies spam and ham messages.