**Methodology:**

We begin by loading the dataset, which is divided into two parts: Fake News and Real News. Initially, we explored both datasets separately to understand their structure — including column names, number of rows, and more.

**Data Preprocessing:**

* Before merging, a new column Label was added:
  + 0 for Fake news
  + 1 for Real news
* Both datasets were merged and shuffled to remove any ordering bias.
* We analyzed the merged dataset using .info(), .shape, etc.
* The column names were converted to uppercase for consistency.
* Duplicate rows were checked and removed.
* Unimportant columns such as DATE and SUBJECT were dropped.

**Text Cleaning:**

We focused on the TEXT column:

* Removed leading/trailing spaces.
* Converted all text to lowercase.
* Removed punctuation and special characters.

**Tokenization and Lemmatization:**

* The cleaned text was tokenized using TF-IDF Vectorizer and stored in a new column tokens.
* Applied simple lemmatization to reduce words to their base forms (e.g., “cats” → “cat”).
* Then applied POS-based lemmatization for more accurate word normalization (e.g., “running” → “run” depending on its part of speech).

At this point, Data Preprocessing was complete.

**Exploratory Data Analysis (EDA)**

* Visualized the distribution of Fake vs Real News using a bar plot.
* Generated Word Clouds for both fake and real articles to identify most frequent terms.
* Used a histogram to analyze word counts in fake vs real news.
* Conducted sentiment analysis:
  + Fake news tends to show lower polarity compared to real news, indicating less neutral tone and more emotionally charged language.

**Feature Extraction**

* Combined lemmatized text into strings.
* Applied TF-IDF vectorization to convert text to numerical features.
* Added extra numerical features:
  + Length of text
  + Sentiment score

Final feature set: TF-IDF + text length + sentiment score

**Model Training**

* Feature matrix: X = [TF-IDF, length, sentiment]
* Labels: y = Label
* Performed train-test split
* Applied and evaluated two models:
  + Logistic Regression
  + Multinomial Naive Bayes

**Evaluation Metrics:**

**Used:**

* Accuracy Score
* F1 Score
* Precision
* Confusion Matrix
* Classification Report

**Design Choices**

* Logistic Regression: Chosen for binary classification (0 = Fake, 1 = Real), especially effective with linearly separable data.
* Multinomial Naive Bayes: Best suited for text classification when using Bag-of-Words or TF-IDF features.

**Evaluation Results:**

**Logistic Regression**

* **Training Accuracy: 0.9923**
* **Test Accuracy: 0.9866**

**Classification Report:**

precision recall f1-score support

Fake (0) 0.99 0.98 0.99 4652

Real (1) 0.98 0.99 0.99 4286

Accuracy: 0.99 (8938 samples)

**MultinomialNB**

* **Training Accuracy: 0.9486**
* **Test Accuracy: 0.9404**

**Classification Report:**

precision recall f1-score support

Fake (0) 0.95 0.93 0.94 4652

Real (1) 0.93 0.95 0.94 4286

Accuracy: 0.94 (8938 samples)

**Key Insights**

* Fake news tends to have more emotional or opinionated language, as revealed by higher polarity values from sentiment analysis.
* Real news is more balanced and neutral, indicating more objective reporting.

