1. Introduction

Spam detection is an important task in text classification that helps filter out unwanted or fraudulent messages. In this assignment, we developed a machine learning model to classify SMS messages as either spam or ham (not spam) using the SMSSpamCollection dataset.

2. Dataset Description

The dataset used is the SMSSpamCollection from the UCI Machine Learning Repository, which contains 5,574 short messages that are labeled as spam or ham.

- Spam (1): Unwanted promotional or fraudulent messages.
- Ham (0): Legitimate user messages.

All messages were preprocessed to eliminate noise and converted into numerical features for training the model.

3. Data Preprocessing

To ensure optimal model performance, the following preprocessing steps were applied:

- Text Cleaning:
  - Converted text to lowercase.
  - Removed punctuation and numbers.
  - Trimmed unnecessary spaces.
- Feature Extraction:
  - I used TF-IDF vectorization to transform the text into numerical representations.
- Dataset Splitting:
  - 80% training, 20% testing using `train_test_split()` from Scikit-learn.

4. Exploratory Data Analysis (EDA)

Several EDA techniques were used to understand the dataset:

- Spam vs. Ham Distribution: Visualized using a count plot.
- Message Length Distribution: Spam messages were generally shorter than ham messages.
- Common words found in spam messages include "free," "win," "text," and "claim."

5. Model Implementation

I implemented two classification models:

1. Naïve Bayes (MultinomialNB)
2. Logistic Regression (with Hyperparameter Tuning using GridSearchCV)

Each model was trained on the processed dataset and evaluated on test data.

6. Model Evaluation

Performance was evaluated using Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.

Naïve Bayes Performance

- Accuracy: 96.86%
- Precision: 100%
- Recall: 76.51%
- F1-Score: 86.69%

Logistic Regression Performance

- Accuracy: 96.77%
- Precision: 99.13%
- Recall: 76.51%
- F1-Score: 86.36%

Confusion Matrix Analysis:

- Naïve Bayes achieved perfect precision (1.0), indicating no false positives were predicted; however, it had lower recall.
- Logistic Regression had slightly lower precision but performed well overall.

7. Model Comparison and Insights

Model	Pros	Cons
Naïve Bayes	Perfect Precision (100%) ✅	Lower Recall (76.51%) ❌
Logistic Regression	Balanced Precision & Accuracy ✅	Lower Recall (76.51%) ❌

Best Choice?

- If avoiding false positives (FP) is crucial, consider using Naïve Bayes.
- If a balance between Precision and Recall is needed, then Logistic Regression is appropriate.
- Future improvements may include additional feature engineering, ensemble models, or deep learning.

8. Challenges Faced & Solutions

- Achieving a Balance Between Precision and Recall: Implemented various classifiers and analyzed their performance.
- Optimized Text Preprocessing: Employed TF-IDF rather than simple count vectorization. Addressed Logistic Regression Convergence Warning: Increased max_iter to 500 to resolve the issue.

9. Conclusion

Using the Naïve Bayes and Logistic Regression models, spam detection was effectively done in this project. Each model had unique characteristics and both did well. To further improve classification accuracy, future developments might incorporate deeper learning methods or more feature engineering