Spam Mail Detection

1. Overview

This project focuses on building a machine learning model to classify emails as **spam** or **ham**.

- **Spam** refers to unwanted, irrelevant, or deceptive emails often sent in bulk for advertising, phishing, or fraud.
- **Ham** refers to legitimate, non-spam emails that are relevant and safe for the recipient.

2. Objective

The primary goal of this project is to develop an automated system that can accurately distinguish between spam and ham emails. By leveraging machine learning techniques, the model aims to:

- Enhance **email security** by filtering out harmful or misleading messages.
- Reduce **user inconvenience** by minimizing unwanted emails in inboxes.
- Improve **email organization** by ensuring that only relevant emails reach users.

3. Methodology

Data Collection & Preprocessing

- The dataset is read from `"Spam Detection.csv"`, containing email messages and their corresponding labels (`ham` or `spam`).
- The labels are mapped to numerical values (`ham = 0`, `spam = 1`) for model training.
- The dataset is split into training (80%) and testing (20%) subsets using `train_test_split()`.

Feature Engineering

- Text data is tokenized using `CountVectorizer`, which converts emails into a **bag-of-words representation**.

Model Training & Evaluation

- **Algorithm Used**: Multinomial Naïve Bayes (`MultinomialNB`), which is effective for text classification problems.
- The model is trained using the transformed training data.
- Performance is evaluated using:
    - **Accuracy Score**
    - **Confusion Matrix**
    - **Classification Report**

Model Testing

- A few test emails (both spam and ham) are provided to check model predictions.

4. Results & Findings

- The model is trained and tested, achieving a reasonable accuracy.
- The evaluation metrics (precision, recall, F1-score) provide insights into the model's effectiveness.

5. Applications

- Can be integrated into email services to **filter spam emails automatically**.
- Could be further improved with **advanced NLP techniques** like TF-IDF, deep learning, or transformer models.

6. Future Enhancements

- Use **TF-IDF vectorization** instead of simple count-based tokenization.
- Explore **more advanced models** such as SVM or deep learning-based approaches.
- Implement **real-time email classification** with a deployment pipeline.