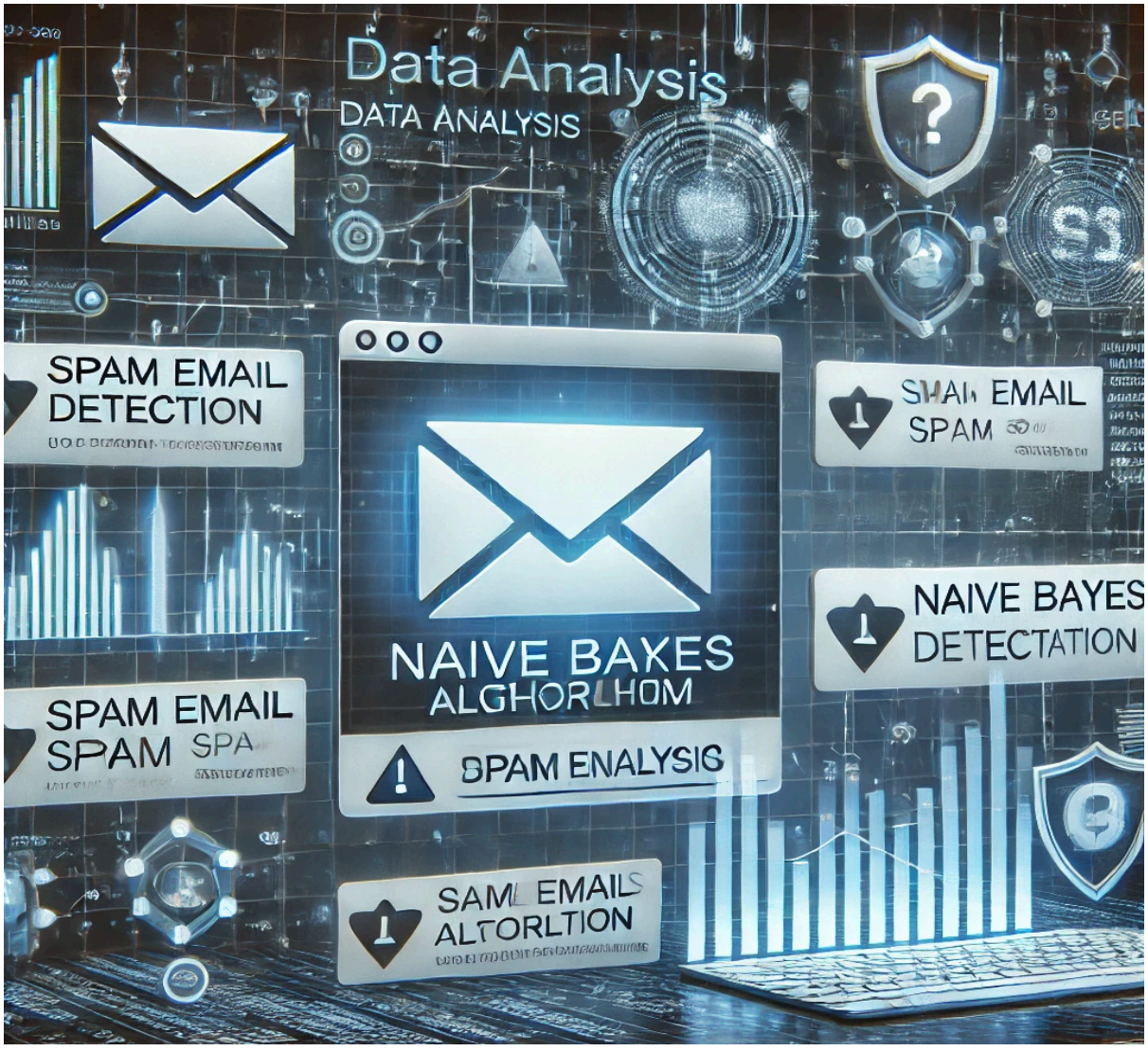
Spam Email Detection using Naïve Bayes

-Dinakaran.k

Overview:



This project focuses on developing an AI-powered **Spam Email Detection System** using **Naïve Bayes Classification**. The system analyzes email content to classify messages as **spam** or **not spam (ham)**, helping users filter out unwanted or malicious emails.

Key Features:

- **Machine Learning-Based Spam Detection:** Uses **Multinomial Naïve Bayes (MNB)** for text classification.
- **Dataset Utilized: Enron Email Dataset** from Kaggle.
- **Text Preprocessing:** Cleans email text by removing **punctuation, numbers, and extra spaces**.
- **TF-IDF Vectorization:** Converts email text into numerical form for better classification.
- **Model Evaluation:** Provides **accuracy, precision, and recall** metrics.
- **Real-Time Email Classification:** Allows users to input email details and predict whether it is spam.

Workflow & Implementation:

1. Data Collection & Preprocessing

- **Dataset Source: Enron Spam Email Dataset (Kaggle)**
- **Data Cleaning:**
    - Removed missing values.
    - Mapped `Category` column to binary labels (`1 = spam`, `0 = not spam`).
    - Applied **text normalization** (lowercasing, punctuation removal, whitespace trimming).

2. Splitting Data for Model Training

- **Training Set:** 80% of emails.
- **Testing Set:** 20% of emails.

3. Model Training

- **Algorithm Used: Multinomial Naïve Bayes (MNB)**
- **Pipeline:**
    - **CountVectorizer:** Converts email text into token frequency.
    - **TF-IDF Transformer:** Converts token frequencies into weighted numerical values.
    - **Naïve Bayes Classifier:** Classifies emails based on learned spam patterns.

4. Model Evaluation

- **Metrics Used:**
    - **Accuracy Score:** Measures overall correctness.
    - **Precision Score:** Measures how many predicted spam emails are actually spam.
    - **Recall Score:** Measures how many actual spam emails were correctly detected.

5. Real-Time Spam Detection

- **User Input:** Users enter email **subject** and **body**.
- **Preprocessing Applied:** Email content is cleaned before classification.

Prediction Output:

🚨 Spam
✅ Not Spam

Challenges and Solutions:

- **Challenge:** Handling dataset inconsistencies (e.g., missing email content).
    - **Solution:** Applied **data cleaning techniques** and removed empty rows.
- **Challenge:** Improving classification accuracy for borderline spam emails.
    - **Solution:** Used **TF-IDF weighting** to enhance keyword importance.
- **Challenge:** Handling large datasets efficiently.
    - **Solution:** Used **scikit-learn pipelines** to optimize text processing.

Progress and Next Steps:

Accomplishments:

- Successfully trained a **Spam Email Classifier** with **Naïve Bayes**.
- Achieved **high accuracy** and **real-time spam detection**.
- Integrated a **user-friendly email classification tool**.

Next Steps:

- Integrate **Deep Learning Models (LSTMs, Transformers)** for better accuracy.
- Implement a **real-time email filtering system** for incoming messages.
- Improve detection for **phishing emails** by incorporating domain analysis.

Conclusion:

The **Spam Email Detection using Naïve Bayes** provides an efficient solution for filtering spam emails. Using **ML-based text classification**, the model helps improve email security and user productivity by accurately identifying unwanted messages.