📧 Email Spam Detection using Machine Learning

A lightweight, web-based spam detection system that classifies email content as **Spam** or **Not Spam** using a trained ML model. Built with Python, Flask, and Scikit-learn, this project is beginner-friendly and suitable for showcasing ML + web integration.

□ Problem Statement

Spam emails reduce productivity and can lead to security threats. The goal is to build a machine learning model that can **automatically classify email messages** as spam or not spam (ham).

🎯 Objective

- Build a spam email classifier using **Naive Bayes**
- Preprocess and vectorize text data using **TF-IDF**
- Evaluate the model for **high accuracy (90%+)**
- Create a user-friendly **web interface using Flask**
- Present results with a clean and modern UI (Dark Mode)

🛠️ Tech Stack

Component	**Tool/Library**
Language	Python 3.x
ML Model	Multinomial Naive Bayes
Libraries	pandas, scikit-learn, nltk, Flask
Frontend	HTML, CSS (Dark UI with SVG icon)
Deployment	Run locally with Flask server

📁 Project Structure

```
EmailSpamDetection/
│
├── app.py               # Flask web app
├── model_trainer.py     # ML training script
├── model.pkl            # Trained ML model
├── vectorizer.pkl       # TF-IDF vectorizer
├── index.html           # Dark mode UI (no templates folder)
├── spam.csv             # Dataset (from Kaggle)
├── requirements.txt     # Python dependencies
└── README.md            # Project documentation
```

📂 Dataset

Dataset used: [SMS Spam Collection Dataset](https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset)

- Format: CSV
- Columns: `label` (spam/ham), `text` (email content)

📊 Model Pipeline

1. Load and preprocess data
2. Lowercase, remove stopwords (NLTK)
3. Convert to TF-IDF vectors
4. Train **Multinomial Naive Bayes**
5. Evaluate using accuracy, precision, recall
6. Save model and vectorizer using `pickle`

Install dependencies

```
pip install -r requirements.txt
```

Or manually:

```
pip install pandas scikit-learn nltk flask
```

Train the model (creates model.pkl, vectorizer.pkl)

```
python model_trainer.py
```

Run the web app

```
python app.py
```