The program is designed to classify emails as either \*spam\* (unwanted or malicious emails) or \*ham\* (legitimate emails). It uses a \*Naive Bayes classifier\*, a popular machine learning algorithm for text classification, to predict whether a given email is spam or not. The application provides a user-friendly interface where users can input an email message, and the system will classify it in real-time.

**Key Features:**

**1. Dataset Loading:**

The program loads a dataset (spam.csv) containing labeled email messages (spam or ham).

The dataset is preprocessed to map labels (ham and spam) to binary values (0 for ham and 1 for spam).

**2. Feature Extraction:**

The text data (email messages) is converted into numerical features using \*CountVectorizer\*, which transforms the text into a matrix of token counts.

**3. Model Training:**

The dataset is split into training and testing sets.

A \*Multinomial Naive Bayes\* model is trained on the training data to classify emails.

**4. Model Evaluation:**

The model's performance is evaluated using \*accuracy\* and a \*classification report\*, which includes precision, recall, and F1-score.

**5. User Interface:**

The application is built using \*Streamlit\*, a Python library for creating web apps.

Users can input an email message in a text area, and the system will classify it as spam or ham.

The model's performance metrics (accuracy and classification report) are displayed for transparency.

---

**How It Works:**

**1. User Input:**

The user enters an email message in the text area provided in the sidebar.

When the user clicks the \*"Check"\* button, the input message is processed.

**2. Prediction:**

The input message is transformed into numerical features using the same CountVectorizer used during training.

The trained Naive Bayes model predicts whether the message is spam or ham.

**3. Output:**

The result is displayed on the screen, indicating whether the email is classified as \*spam\* or \*ham\*.

If no input is provided, an error message prompts the user to enter an email message.

**4. Model Performance:**

The application displays the model's \*accuracy\* and a detailed \*classification report\* to give users insight into how well the model performs.

**Technologies Used**

\*Python\*: The core programming language used for the application.

\*Pandas\*: For loading and preprocessing the dataset.

\*Scikit-learn\*: For feature extraction (CountVectorizer), model training (MultinomialNB), and evaluation (accuracy\_score, classification\_report).

-\*Streamlit\*: For building the interactive web interface.

**How to Run the Program:**

**1. Install Dependencies:**

- Ensure you have Python installed.

- Install the required libraries:

bash

pip install pandas scikit-learn streamlit

**2. Download the Dataset:**

Place the spam.csv file in the same directory as the script.

**3. Run the Application:**

Save the script as spam\_detection\_app.py.

Run the Streamlit app:

bash

streamlit run spam\_detection\_app.py

**4. Access the Application:**

Open the provided URL in your web browser to interact with the app.