Email Spam Classifier

Author: Nithishwar T
Date: September 29, 2025
GitHub Repository: https://github.com/nithishwar17/Email-Spam-Classifier

1.Abstract:

In the era of digital communication, spam emails remain a significant problem, cluttering inboxes with irrelevant and potentially malicious content. This project presents a **Machine Learning–based Email Spam Classifier** that automatically categorizes emails as spam or ham (legitimate). Using Natural Language Processing (NLP) techniques and supervised learning algorithms, the system learns patterns in spam emails and provides accurate predictions for new messages. Experiments show that the developed models achieve high accuracy and precision, making this solution efficient and scalable for real-time email classification.

2. Proposed System:

The proposed system classifies emails into **Spam** or **Ham** categories. The workflow is as follows:

1. **Dataset Collection:** Emails are gathered from publicly available datasets like the UCI SMS Spam Collection or Kaggle datasets.
2. **Preprocessing:** Removal of stopwords, punctuation, and unwanted characters; tokenization; stemming/lemmatization.
3. **Feature Extraction:** Use of TF-IDF (Term Frequency–Inverse Document Frequency) or Count Vectorizer to convert text into numerical feature vectors.
4. **Model Training:** Training supervised ML algorithms such as Naive Bayes, Logistic Regression, and Support Vector Machine (SVM).
5. **Evaluation:** Models are validated using metrics like accuracy, precision, recall, F1-score, and confusion matrix.
6. **Prediction:** The best-performing model is deployed for real-time spam classification.

3. Technologies Used:

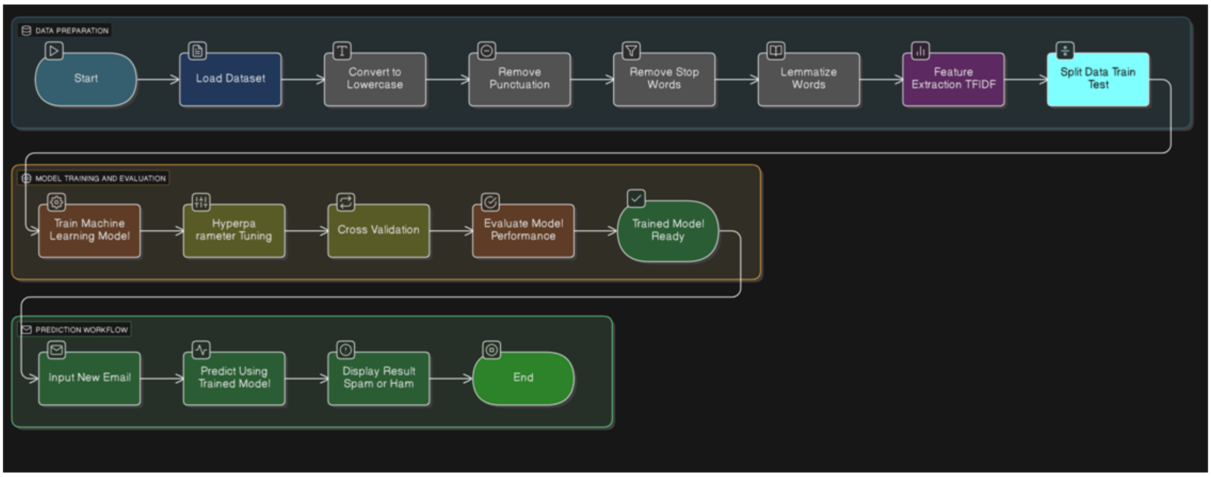
- **Language:** Python
- **Libraries:**
  scikit-learn → model training and evaluation
  pandas, NumPy → data handling
  NLTK / spaCy → text preprocessing
  Matplotlib / Seaborn → visualization
- **Algorithms:** Naive Bayes, Logistic Regression, Support Vector Machine (SVM)

- **Datasets:**
  UCI SMS Spam Collection Dataset
  Kaggle Email Spam Datasets

4. Implementation:

- **Step 1: Data Loading** → Import datasets into Python environment using pandas.
- **Step 2: Preprocessing** → Convert text to lowercase, remove special characters, stopword removal, tokenization, stemming/lemmatization.
- **Step 3: Feature Engineering** → Apply TF-IDF vectorization for meaningful feature extraction.
- **Step 4: Model Training** → Train classifiers (Naive Bayes, Logistic Regression, SVM) on training data.
- **Step 5: Model Evaluation** → Test models using unseen data, calculate accuracy, precision, recall, and F1-score.
- **Step 6: Visualization** → Plot confusion matrices and important spam keywords for interpretability.

5. Flow Diagram:



6. Results & Discussion:

- Models achieved **high accuracy and low false positive rates**.
- Naive Bayes showed strong performance due to its suitability for text classification.
- SVM and Logistic Regression provided competitive results, especially with large feature sets.
- Visualization of spam-indicative keywords helped in interpreting the model.

- Each model was evaluated on the unseen test data. The performance was measured using accuracy, precision, and recall. The **Support Vector Machine (SVM)** model emerged as the top performer.

Model	Accuracy	Spam Precision	Spam Recall
Logistic Regression	95.07%	0.96	0.66
Naive Bayes	97.49%	1.00	0.81
Support Vector Machine (SVM)	**97.58%**	**0.98**	**0.83**

7. Conclusion & Future Work

This project successfully developed a spam email classifier using **NLP + Machine Learning** techniques. The classifier demonstrated high accuracy and can be integrated into real-world email systems for spam detection.

Future Enhancements:

- Extend to detect phishing and scam emails.
- Deploy as a browser extension or email server plugin.
- Explore Deep Learning models such as LSTMs and Transformers for even higher accuracy.

8. **References:**

1. UCI SMS Spam Collection Dataset: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset
2. SpamAssassin Public Corpus
3. Research papers on NLP-based spam detection
4. Documentation of scikit-learn, NLTK, spaCy