Project Documentation

Project Title:

AI-Based Spam Message Classifier

Abstract:

The growth of digital communication has led to an increase in spam messages, including fraudulent SMS, promotional emails, and phishing attempts. Manual filtering is time-consuming and unreliable.

This project proposes an AI-based Spam Message Classifier using text preprocessing, TF-IDF vectorization, and the Naive Bayes classification algorithm. The model analyzes SMS/email content and predicts whether the message is Spam or Not Spam with high accuracy.

The system is lightweight, fast, and suitable for real-world applications such as email filtering, SMS classification, and security enhancement.

Introduction:

Spam messages cause inconvenience, waste time, and often lead to financial or security threats. Automated spam detection systems help organizations and individuals maintain a clean communication environment.Traditional rule-based methods fail due to continuously evolving spam patterns. Machine learning techniques allow the system to learn from historical spam data and classify messages based on patterns, word frequency, and linguistic features.This project demonstrates how simple machine learning models can solve real-world communication problems efficiently.

Problem Statement:

With the exponential increase in SMS and email communication, identifying fraudulent or promotional messages manually is difficult.
The absence of automated spam detection increases the risk of:

- Phishing attacks
- Fraud messages
- Misleading promotional content
- Unwanted advertising

There is a need for an AI-based system that automatically classifies messages into Spam or Not Spam.

Objectives:

The primary objectives of this project are:

1. To build an intelligent spam detection classifier using machine learning.
2. To preprocess raw SMS/email text into meaningful features.
3. To implement the Naive Bayes classifier for spam detection.

4. To evaluate system performance using metrics such as accuracy and precision.

5. To provide a simple interface to test messages in real time (optional).

Scope of the Project:

In-Scope

- Classifying SMS or small email text
- Binary classification: Spam / Not Spam
- Machine learning implementation
- Use of a standard dataset
- Text-vectorization and model evaluation

Out-of-Scope

- No deep learning or large email datasets
- No advanced NLP techniques (transformers, BERT)
- No multi-language support

System Requirements:

Software Requirements:

- Python 3.x
- Jupyter Notebook / VS Code / PyCharm
- Libraries:
  - Scikit-learn
  - Pandas
  - NumPy
  - NLTK
  - Regex

Hardware Requirements:

- Minimum 4GB RAM
- 1GB free storage
- Any OS (Windows/Linux/Mac)

Dataset:

The model uses the SMS Spam Collection Dataset, containing 5,572 SMS messages labeled as:

- ham → Not Spam
- spam → Unwanted message

Dataset source: UCI Machine Learning Repository (public dataset).

System Architecture:

1. Data Collection

Load SMS dataset with labeled spam/ham messages.

2. Data Preprocessing

Convert text to lowercase

Remove symbols/punctuation

Remove stopwords

Apply stemming/lemmatization

3. Feature Extraction

Convert text to numerical vectors using TF-IDF Vectorizer.

4. Model Training

Train a Multinomial Naive Bayes classifier

5. Prediction

System analyzes input text and outputs:

- Spam
- Not Spam

6. Evaluation

Metrics:

- Accuracy
- Precision
- Recall

- Confusion Matrix

Methodology:

1.Text Preprocessing

Steps include:

1. Lowercasing
   Example: “FREE Money!!!” → “free money”

2. Removing punctuation & numbers
   Regex is used.

3. Stopword removal
   Words like “the”, “is”, “an” are removed.

4. Tokenization
   Message → list of words.

5. Stemming/Lemmatization
   “running” → “run”

2. TF-IDF Vectorization

TF (Term Frequency)

Indicates how often a word appears in a document.

IDF (Inverse Document Frequency)

Gives importance to rare words.

TF-IDF = TF × IDF

Converts text to numerical feature vectors for ML models.

3.Naive Bayes Algorithm

The model uses Multinomial Naive Bayes, ideal for text classification.

Characteristics:

- Probabilistic model
- Fast and scalable
- Works well with word frequency data

Formula:

$$\mathbf{P(A \mid B)=P(B)P(B \mid A)P(A)}$$

Implementation Details:

Steps:

1. Load dataset
2. Preprocess all messages
3. Apply TF-IDF vectorization
4. Split dataset (80% training, 20% testing)
5. Train Naive Bayes model

6. Predict and evaluate

7. Test with user input

Evaluation Metrics:

1. Accuracy

Measures correctness of model.

2. Precision

How many predicted spam messages were actually spam?

3. Recall

How many real spam messages were detected?

4. F1 Score

Balance between precision & recall.

5. Confusion Matrix

Shows True/False results of model prediction.

Expected accuracy: **94–98%**

Results:

- Model identifies spam words like “win”, “free”, “reward”, etc.

- High accuracy and low false positives.

- Effective even with minimal preprocessing.

Applications:

- Email spam filtering
- SMS spam detection
- Fraud detection in chat apps
- Customer support filtering
- Social media automated moderation

Advantages:

- Simple and lightweight
- High accuracy
- Low computational requirement
- Works well on small datasets
- Easy to integrate in mobile and web apps

Limitations:

- Naive Bayes assumes word independence
- Cannot detect context or meaning deeply
- Not suitable for long emails or advanced phishing attacks
- Performance depends on dataset quality

Future Enhancements:

- Deep learning models (LSTM, BERT)
- Multi-language support
- Real-time spam detection mobile app
- Integration with email clients
- Better handling of phishing links

Conclusion:

The AI-Based Spam Message Classifier is a highly effective solution for filtering unwanted SMS and email messages.Using Naive Bayes and text preprocessing, the system achieves excellent accuracy with minimal computational cost.
This project demonstrates how AI and NLP can solve real-world communication problems efficiently and reliably.