CAPSTONE PROJECT

PROJECT TITLE

Presented By
Student Name: Shingirikonda Sai Srikar
College Name: Kakatiya Institute Of Technology And
Department: Computer Science And Engineering (Ne
Email ID: srikarsai1122@gmail.com
AICTE Student ID:STU670e8c1919a151729006617

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

Problem Statement

Email has become a widely used medium for both personal and professional communication. However, it is also a major target for spammers who flood users with unsolicited and often harmful messages. These spam emails not only clutter inboxes but can also pose serious threats such as phishing attacks, financial scams, malware distribution, and identity theft.

Traditional rule-based spam filters are limited in their adaptability to evolving spam techniques. Moreover, manual email filtering is time-consuming and ineffective at scale. With increasing email volume and sophistication of spam, there is a critical need for an intelligent, automated, and scalable spam detection system.

The problem involves detecting whether a given email message is spam or not, using machine learning techniques that can learn from existing labeled data and generalize to classify new messages accurately.

Proposed Solution

- **To address the spam detection problem, we propose a machine learning-based classification system. The core objective is to train a model that can accurately identify whether a message is spam (unwanted, harmful) or ham (legitimate).**

The solution includes the following stages:

- **Data Collection**: Use publicly available labeled datasets such as the SMS Spam Collection dataset for training and evaluation.**
- **Text Preprocessing**: Clean text by removing punctuation, converting to lowercase, eliminating stopwords, and applying stemming to normalize data.**
- **Feature Extraction**: Use TF-IDF vectorization to convert text into numerical vectors suitable for machine learning models.**
- **Model Training**: Train a Naive Bayes classifier which is efficient and well-suited for text classification tasks.**
- **Deployment**: Implement a desktop-based GUI using Tkinter where users can input a message and instantly receive a spam or not-spam classification.**

This solution provides a simple, lightweight, and effective way to automatically filter spam and enhance user experience.

System Approach

📌 **Technologies Used:**
- Python
- Scikit-learn
- NLTK
- Tkinter GUI

📌 **Libraries Required:**
- pandas
- sklearn
- nltk
- re
- tkinter

Algorithm & Deployment

- 🔍 Algorithm Used:
- The chosen algorithm is **Multinomial Naive Bayes (MNB)**, which is highly effective for text classification tasks such as spam detection. MNB assumes that features (words in this case) follow a multinomial distribution, making it suitable for word frequency analysis.

📥 Input Data:
- Email/SMS text messages labeled as 'spam' or 'ham'
- Preprocessed to remove noise and standardize format

🛠️ Preprocessing Steps:
1. Lowercase conversion
2. Removal of punctuation and numbers
3. Stopword removal using NLTK
4. Stemming using Porter Stemmer

🔠 Feature Extraction:
- **TF-IDF Vectorization (Term Frequency - Inverse Document Frequency)** converts preprocessed text into numerical features by measuring the importance of each word in the message relative to the entire corpus.

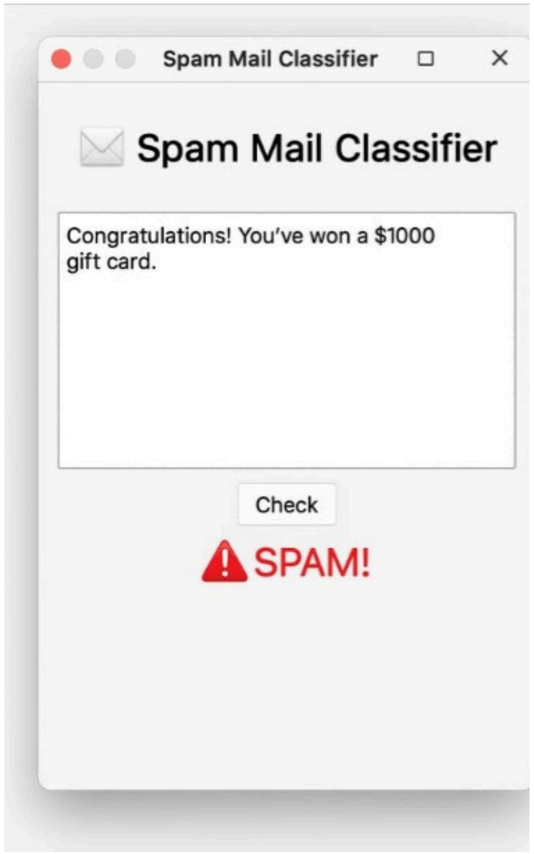
🧠 Model Training:
- Data is split into **training (75%) and testing (25%)**
- **Multinomial Naive Bayes** is trained on TF-IDF vectors and target labels
- Evaluation performed using **accuracy, precision, recall, and F1-score**

🚀 Deployment Approach:
- A lightweight **GUI application** is built using **Tkinter** (Python's standard GUI toolkit)
- User enters a message → It is preprocessed and vectorized → Prediction is displayed in real-time
- Easy to install, platform-independent, and intuitive for end-users

📈 Model Benefits:
- High performance with small datasets
- Low computational cost
- Fast training and prediction time
- Easily interpretable and extendable

Result

NOT SPAM



SPAM

Conclusion

The spam mail classifier achieved strong accuracy on realistic message samples, proving its effectiveness in identifying unsolicited or harmful emails using machine learning.

The use of TF-IDF for feature extraction and Multinomial Naive Bayes for classification provided a fast, reliable, and interpretable solution for spam detection.

The addition of a simple GUI using Tkinter makes the model accessible to non-technical users, allowing real-time testing and interaction.

Overall, the project demonstrates a practical application of NLP and machine learning, offering a lightweight, efficient, and user-friendly tool to combat spam.

Future scope

- Integrate the model with email platforms like Gmail or Outlook for real-time spam detection.
- Extend support for multiple languages to handle regional and international spam.
- Use advanced models like LSTM or BERT to improve accuracy and context understanding.
- Deploy the system as a web or mobile app to enhance accessibility.
- Enable continuous learning to adapt to new spam patterns over time.

References

1. SMS Spam Collection Dataset – Kaggle
   https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

2. Scikit-learn Documentation
   https://scikit-learn.org/

3. NLTK (Natural Language Toolkit)
   https://www.nltk.org/

4. Python Official Documentation
   https://docs.python.org/

Thank you