Project SRS

> *Software Requirements Specification*

Project Title

Spam Detector using Machine Learning

Team Members

SRN	Name	Roll Number	Section	Role
PES1UG25CS092	Arjun D. Rao	26	C3	ML Model, ML-GUI Integration, GUI Implementation
PES1UG25CS726	Ayan Banerjee	36	C3	GUI Events Implementation
PES1UG25EC039	Anish Babu G A	57	C3	GUI Design
PES1UG25CS091	Arjun Anup	1	C3	GUI Design

Problem Statement

The advent of modern communication channels such as **SMS, email, and chat apps** have polluted the internet with **spam messages** that intend to mislead users with fake offers, phishing links, and deceitful promotions.
Manually identifying spam is time-consuming and unreliable, especially when attackers constantly evolve their message patterns.

There is a need for an **automated, intelligent system** capable of:

- Understanding message content
- Detecting linguistic patterns associated with spam
- Accurately classifying messages as **Spam** or **Not Spam**
- Providing instant, user-friendly results through a desktop application

This project solves that problem using a trained **Machine Learning pipeline** that learns from real-world SMS data and classifies messages, emails and any other text-based content.

Tech Stack Overview

- **Programming Language:** Python

- **Machine Learning Library:** scikit-learn
- **Feature Extraction:** TF-IDF Vectorizer
- **Classification Algorithm:** Logistic Regression
- **Dataset:** SMS Spam Collection Dataset by UC-Irvine
- **GUI Framework:** wxPython
- **Supporting Libraries:** pandas, numpy

Data Flow

1. SMS data is loaded and cleaned.
2. Text features are extracted using TF-IDF.
3. A Logistic Regression model is trained on the processed data.
4. User enters a message through the GUI.
5. The model predicts if the message is spam.
6. The GUI shows the result instantly.

Approach / Methodology / Data Structures Used

Methodology

1. **Data Loading**
    - The SMS Spam Collection dataset is loaded using `pandas.read_csv()`.
    - Labels are converted from text (`"spam"`, `"ham"`) into numerical form (`1`, `0`) for model training.
2. **Data Splitting**
    - The dataset is divided into **training (80%)** and **testing (20%)** sets using `train_test_split()` with stratification.
    - Stratification ensures that both sets maintain the original proportion of spam and ham messages.
3. **Feature Extraction (TF-IDF)**
    - Text messages are transformed into numerical vectors using **TF-IDF Vectorization**.
    - TF-IDF highlights important words by reducing the influence of commonly used ones.
    - Parameters used:
        - `ngram_range = (1, 2)` → considers single words and two-word combinations
        - `min_df = 2`, `max_df = 0.95` → removes very rare and overly common terms
4. **Model Selection & Training**
    - A **Logistic Regression** classifier is trained on the TF-IDF features.
    - It learns statistical patterns that distinguish spam messages from normal messages.
5. **Model Evaluation**

- Predictions are compared against true labels.
- Metrics such as **Accuracy, Precision, Recall, and F1-score** are printed to assess performance.

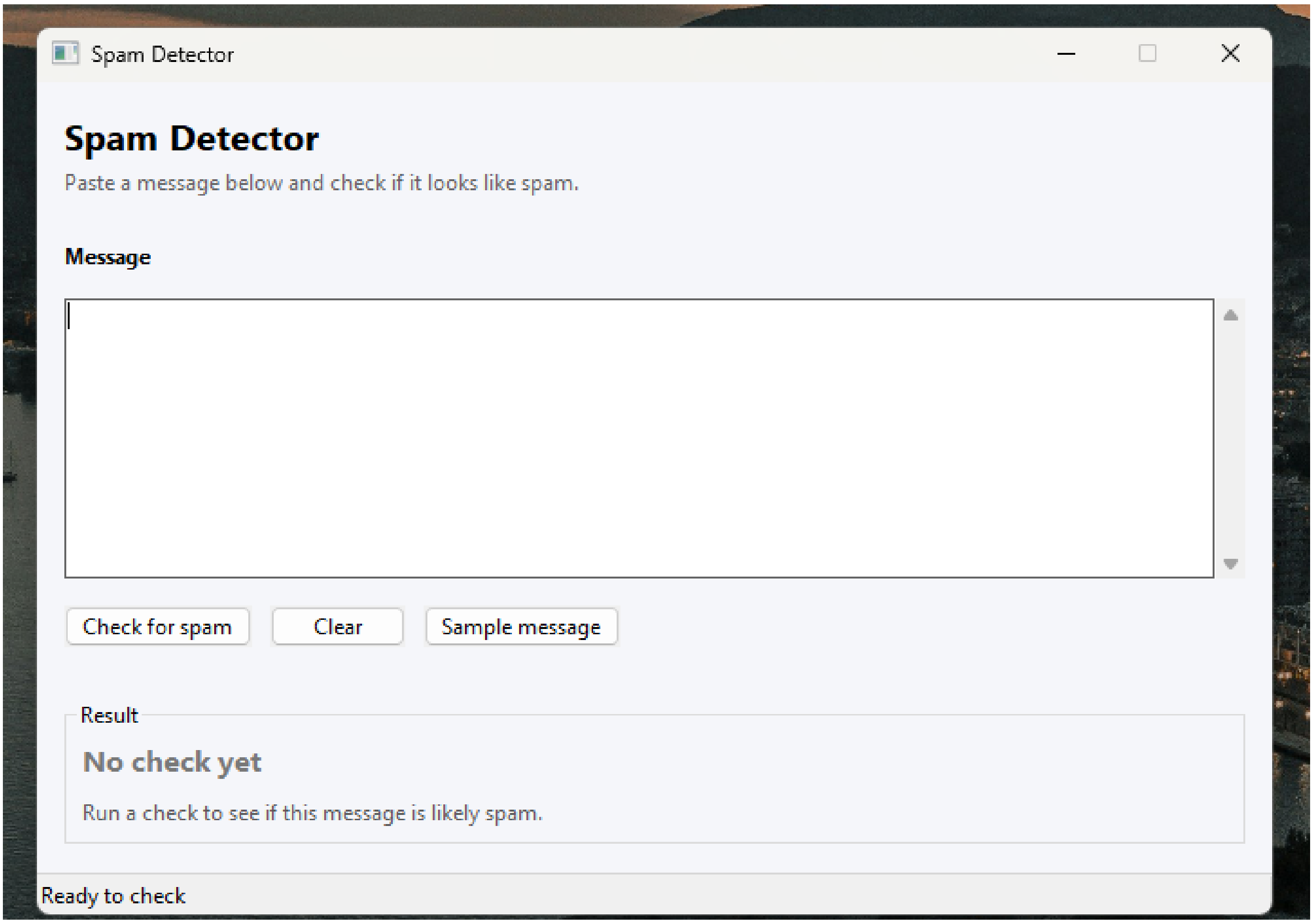
6. **GUI Integration**
    - A desktop user interface is built using **wxPython**.
    - The model's `predict()` function is called when the user clicks **Check for Spam**.
    - Based on prediction:
        - `1` → **SPAM**
        - `0` → **NOT SPAM**
    - The GUI displays results clearly with color-coded labels.

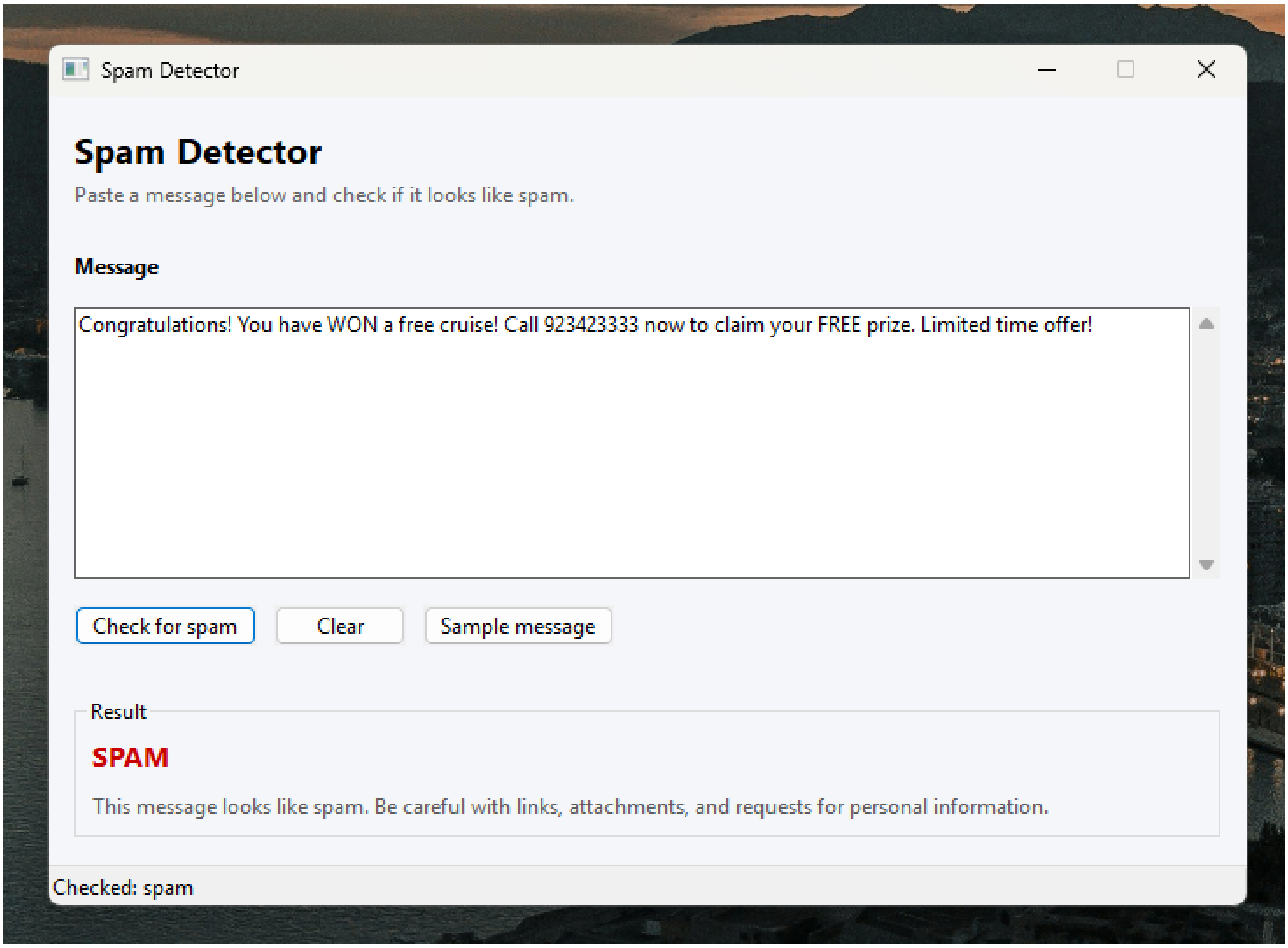
Data Structures Used

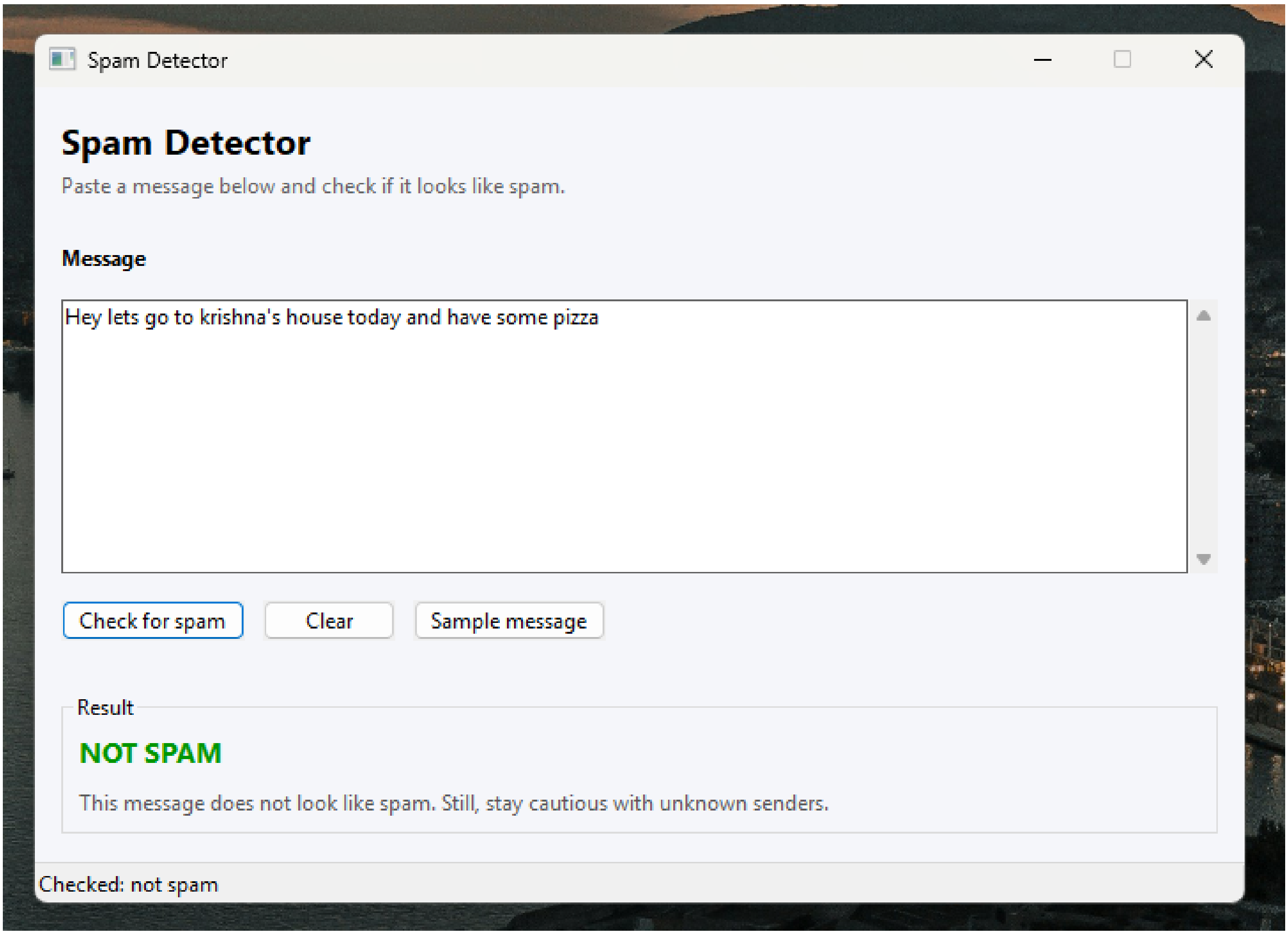
Component	Data Structure	Purpose
Dataset Storage	`pandas.DataFrame`	Holds SMS text and spam/ham labels
Labels	`numpy.ndarray`	Stores numerical values (0/1) for model training
Training Pipeline	`sklearn Pipeline`	Sequentially applies TF-IDF vectorization and Logistic Regression

---

Input & Output Screenshots


Spam Detector
Spam Detector
Paste a message below and check if it looks like spam.
Message
Check for spam
Clear
Sample message
Result
No check yet
Run a check to see if this message is likely spam.
Ready to check


Spam Detector
Spam Detector
Paste a message below and check if it looks like spam.
Message
Congratulations! You have WON a free cruise! Call 923423333 now to claim your FREE prize. Limited time offer!
Check for spam
Clear
Sample message
Result
SPAM
This message looks like spam. Be careful with links, attachments, and requests for personal information.
Checked: spam




```
(.venv) PS C:\Users\azgam\Documents\python-mini-project\spam-detector> py .\main.py
Accuracy: 0.9901
Precision: 1.0000
Recall: 0.9262
F1: 0.9617

Classification report:

              precision    recall  f1-score   support

           0     0.9887    1.0000    0.9943       966
           1     1.0000    0.9262    0.9617       149

    accuracy                         0.9901      1115
   macro avg     0.9944    0.9631    0.9780      1115
weighted avg     0.9902    0.9901    0.9900      1115
```

Challenges Faced

- Handling noisy, unstructured SMS text during preprocessing.
- Dataset imbalance made it harder for the model to detect spam accurately.
- Some legitimate messages resembled spam, causing misclassification.

Scope for Improvement

- **Expand the Dataset** The model is trained on a single SMS corpus. Adding more diverse datasets, multilingual spam data, and real-world user messages can help the model generalize to modern spam techniques.

- **Real-Time Learning System** Allow users to manually mark messages that were incorrectly classified. These labels could be stored and used later to retrain or fine-tune the existing model.

- **Improve GUI Experience** The current interface is functional but minimal. Enhancements could include:

  - A message history panel
  - Dark mode theme

- **Reduce False Positives** Some promotional or friendly messages may be incorrectly tagged as spam. Improving feature engineering and leveraging advanced models can reduce such misclassifications.

Summary

This approach uses a **machine learning pipeline** combining TF-IDF text representation and Logistic Regression to classify messages. The trained model is embedded into an interactive GUI so users can easily test messages and instantly identify potential spam.