**SMS Spam Detection — Project**

* **Project Title :**SMS Spam Detection using Machine Learning
* **Name:** S.Harinie
* **Institution:** SKP Engineering College
* **Department**:B.tech Information technology
* **Date:** 19.10.2025
* **GitHub link:** [**https://github.com/HARINIE-1107/Harinie-07.git**](https://github.com/HARINIE-1107/Harinie-07.git)

**2. Table of Contents**

**Section No. Title Page No.**

1 Title Page 1

2 Table of Contents 2

3 Project Overview 3

4. Objectives & Problem Statement. 4

4.1. Problem statement. 4

4.2. Objectives. 5

5. Proposed Solution. 6

6 Features. 7

6.1. Functional Features. 7

6.2. Non Functional Features. 8

7. Technologies & Tools. 9

8 System Architecture. 10

9. Implementation Steps. 11

10. Output & Screenshots 13

11. Advantages. 15

12. Future Enhancements. 16

13. Conclusion. 17

**3. Project Overview**

The rise of mobile communication has led to an exponential increase in unsolicited text messages,commonly known as SMS spam.

These messages are not only annoying but can

also pose risks

such as phishing attacks, financial scams, and privacy breaches. Detecting such messages using

machine learning ensures safe and secure communication for users. This project focuses on

building a system that classifies SMS messages into “spam” or “ham” using supervised learning.

The scope of this project is limited to English-language SMS messages. The project includes data

preprocessing, model training, and a simple web interface to classify new messages interactively**.**

**4. Objectives & Problem Statement**

**4.1 Problem Statement**

Given a text message (SMS), the objective is to automatically classify it as either spam or ham. The main challenge involves handling short, informal text containing abbreviations, emojis, and typos.

**4.2 Objectives**

* Develop a machine learning model for SMS spam detection.
* Implement preprocessing techniques such as tokenization and stopword removal.
* Compare different ML algorithms and choose the best performer.
* Build a Streamlit-based web interface for real-time SMS classification.
* Evaluate the system using precision, recall, and F1-score metrics.

**5. Proposed Solution**

The project adopts a supervised learning approach using a labeled dataset of spam and ham

messages. Text data is preprocessed and transformed into numerical features using TF-IDF

vectorization. Models such as Multinomial Naive Bayes and Logistic Regression are trained and

evaluated. A Streamlit web interface allows users to input SMS messages and view classification result instantly

**6. Features**

**Functional Features:**

* Accepts SMS text input and outputs “spam” or “ham”.
* Displays prediction confidence score.
* Batch processing for multiple SMS inputs.
* Interactive Streamlit web interface.

**Non-Functional Features:**

* Fast and accurate predictions.
* Lightweight and easy to use.
* Can be extended for multilingual detection.

**7. Technologies & Tools**

* Language: Python
* Libraries: pandas, numpy, scikit-learn, nltk, joblib
* Web Interface: Streamlit
* Development Environment: Jupyter Notebook / VS Code
* Version Control: GitHub

**8. System Architecture**

The system consists of modules for input, preprocessing, feature extraction, model prediction, and

web interface. Data flows from user input → text preprocessing → vectorization → model prediction

→ output display on Streamlit.

**9. Implementation Steps**

1. Data Collection (SMS Spam Collection Dataset).

2. Text cleaning: lowercase, punctuation removal, stopword filtering.

3. Feature extraction using TF-IDF Vectorizer.

4. Model training using Naive Bayes and Logistic Regression.

5. Evaluation with accuracy, precision, recall, and F1-score.

6. Saving trained model using joblib.

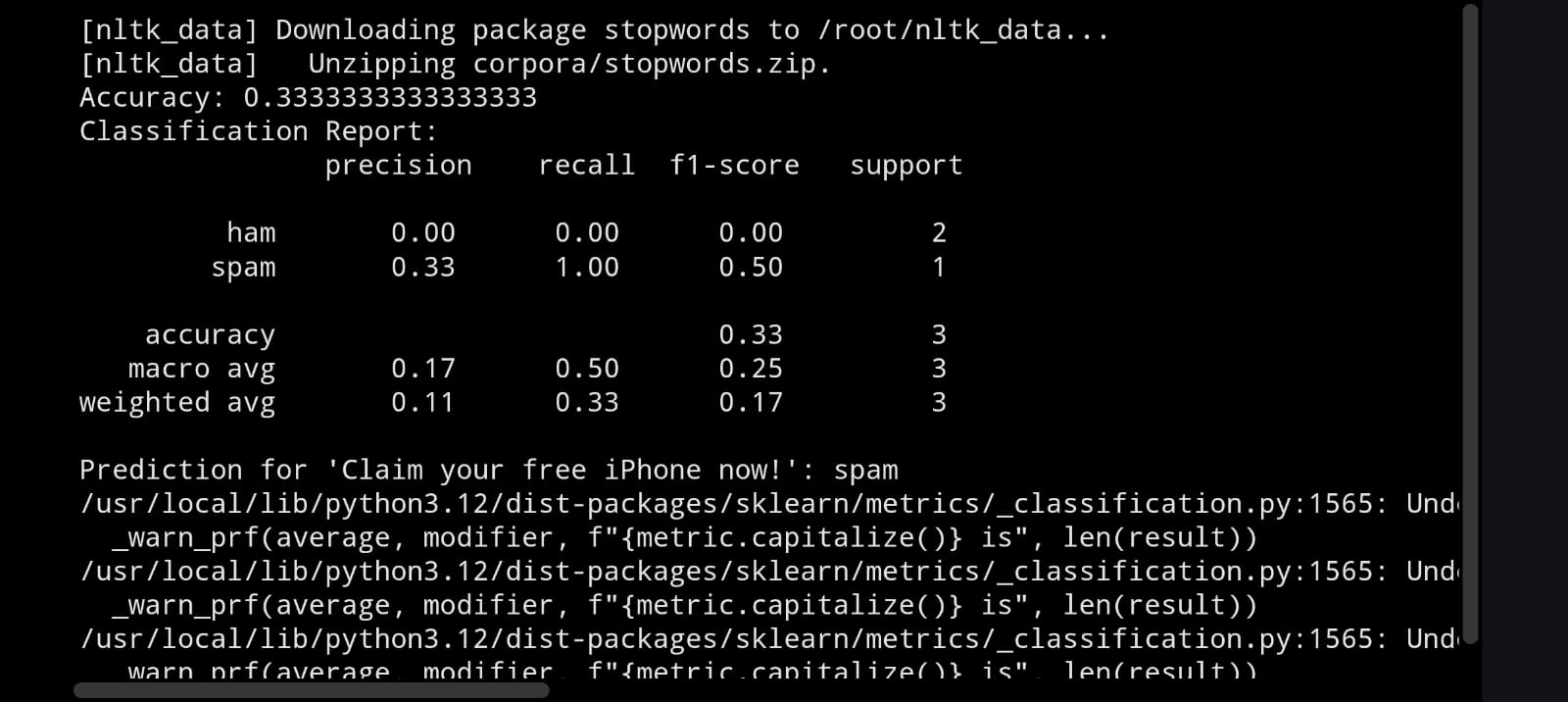
7. Building Streamlit interface for predictions.

8. Testing with unseen messages.

**10. Output / Screenshots**

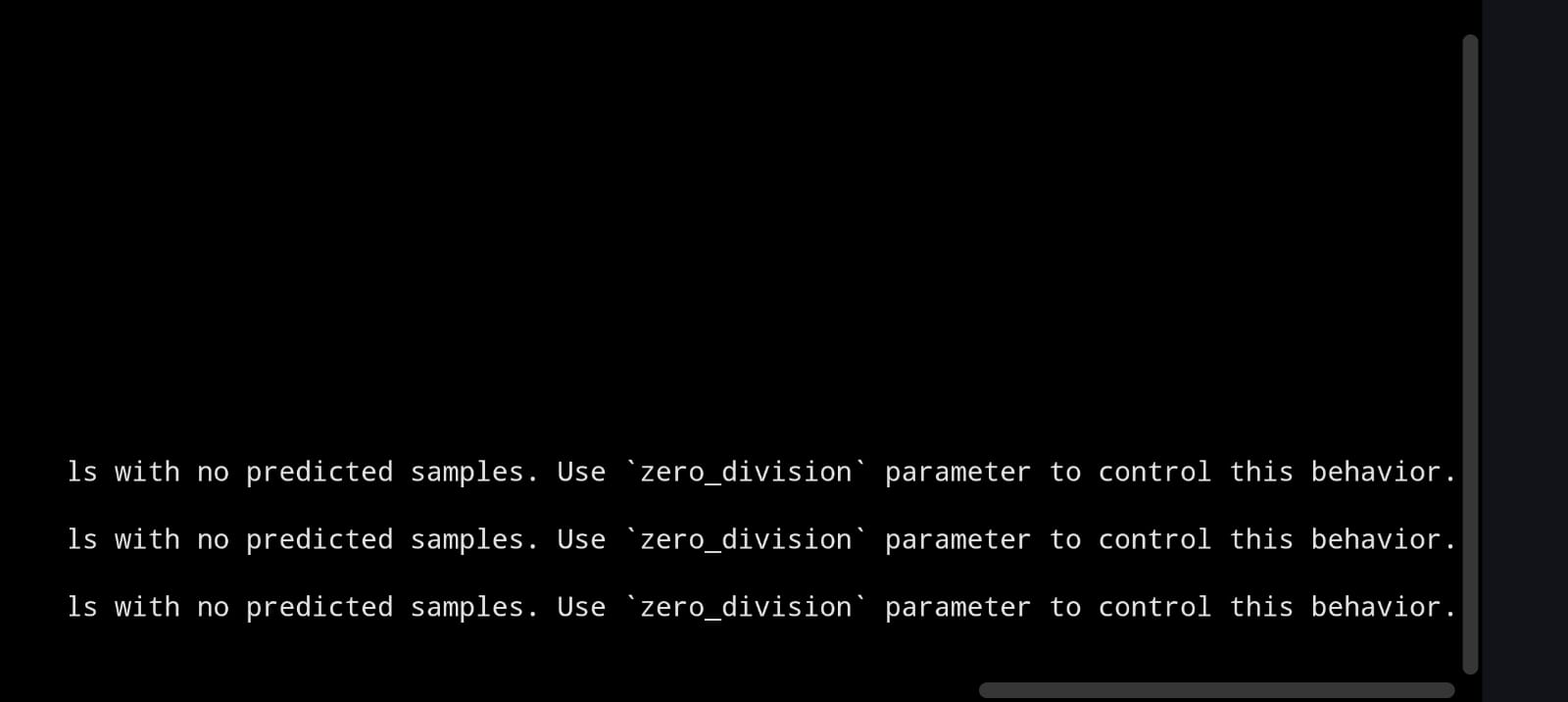
Sample Input: "Congratulations! You have won a free ticket to Maldives. Reply WIN to claim now!"

Predicted Output: Spam (Confidence: 98%)



Sample Input: "Hey, are we meeting for lunch today?"

Predicted Output: Ham (Confidence: 97%)



**11. Advantages**

* Automates spam message detection efficiently.
* High precision and recall with lightweight architecture.
* Easy to use with the Streamlit interface.
* Supports quick retraining and deployment.

**12. Future Enhancements**

* Use deep learning models like LSTM or BERT.
* Extend to handle multiple languages.
* Implement real-time API for integration with messaging apps.
* Enhance UI/UX for mobile use.

**13. Conclusion**

The SMS Spam Detection project successfully demonstrates how machine learning can be applied

to improve digital communication safety. Using TF-IDF features and supervised algorithms, spam

messages can be accurately detected. The Streamlit web app provides an easy interface for

real-time testing. Future improvements can integrate deep learning and multilingual support