SMS Spam Detection Using Machine Learning

Name: Sanjay R

Institution: Tagore Engineering College

Department: B.Tech Artificial Intelligence and Data Science

Date: 20-10-2025

Table of Contents

[1. Project Overview 3](#_Toc211957306)

[Background / Motivation 3](#_Toc211957307)

[Scope 3](#_Toc211957308)

[High-Level Description 3](#_Toc211957309)

[Dataset / Data Source Summary 3](#_Toc211957310)

[2. Objectives & Problem Statement 4](#_Toc211957311)

[Problem Statement 4](#_Toc211957312)

[Objectives 4](#_Toc211957313)

[3. Proposed Solution 5](#_Toc211957314)

[Approach / Methodology 5](#_Toc211957315)

[Pipeline Overview 5](#_Toc211957316)

[Justification of Choices 5](#_Toc211957317)

[4. Features 6](#_Toc211957318)

[Functional Features 6](#_Toc211957319)

[Non-Functional Features 6](#_Toc211957320)

[5. Technologies & Tools 7](#_Toc211957321)

[6. System Architecture 8](#_Toc211957322)

[Architecture Diagram 8](#_Toc211957323)

[Component Descriptions 8](#_Toc211957324)

[Data Flow 9](#_Toc211957325)

[7. Implementation Steps 9](#_Toc211957326)

[8. Output / Screenshots 10](#_Toc211957327)

[9. Advantages 11](#_Toc211957328)

[10. Future Enhancements 11](#_Toc211957329)

[11. Conclusion 11](#_Toc211957330)

# 1. Project Overview

## Background / Motivation

Spam messages have become a major issue for users worldwide, often leading to fraud, phishing, and unwanted advertisements. Automated spam detection ensures safer communication, improved user experience, and protection against data theft and scams.

## Scope

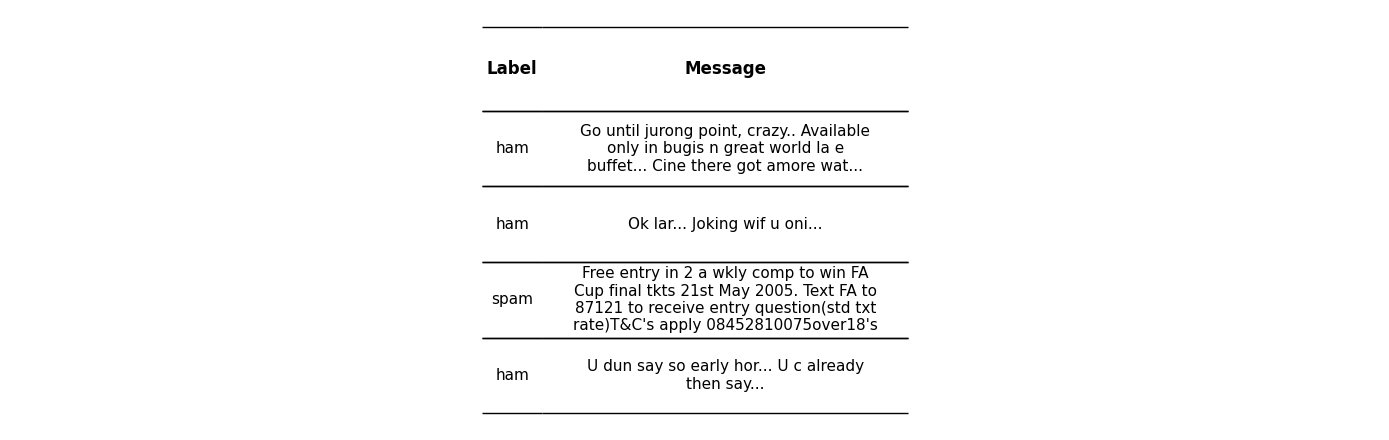
This project focuses exclusively on SMS messages, classifying them into two categories — Spam or Ham (Not Spam). It deals with noisy, short, and unstructured text, including slang, typos, and abbreviations.

## High-Level Description

A machine learning model is developed to classify messages using text-processing techniques and supervised learning algorithms. The project also includes a simple interface for real-time predictions.

## Dataset / Data Source Summary

The dataset used is the UCI SMS Spam Collection Dataset, containing around 5,000+ labeled SMS messages categorized as spam or ham.



# 2. Objectives & Problem Statement

## Problem Statement

Given an SMS message, the system should classify it as spam or ham accurately. Challenges include short text, noisy data, slang, and class imbalance between spam and ham messages.

## Objectives

* Build a predictive model with high accuracy, precision, and recall.
* Implement effective text preprocessing and feature extraction.
* Compare multiple machine learning algorithms.
* Develop a simple interface for real-time message classification.
* Evaluate model performance using various metrics.
* Document results and provide insights for future improvements.

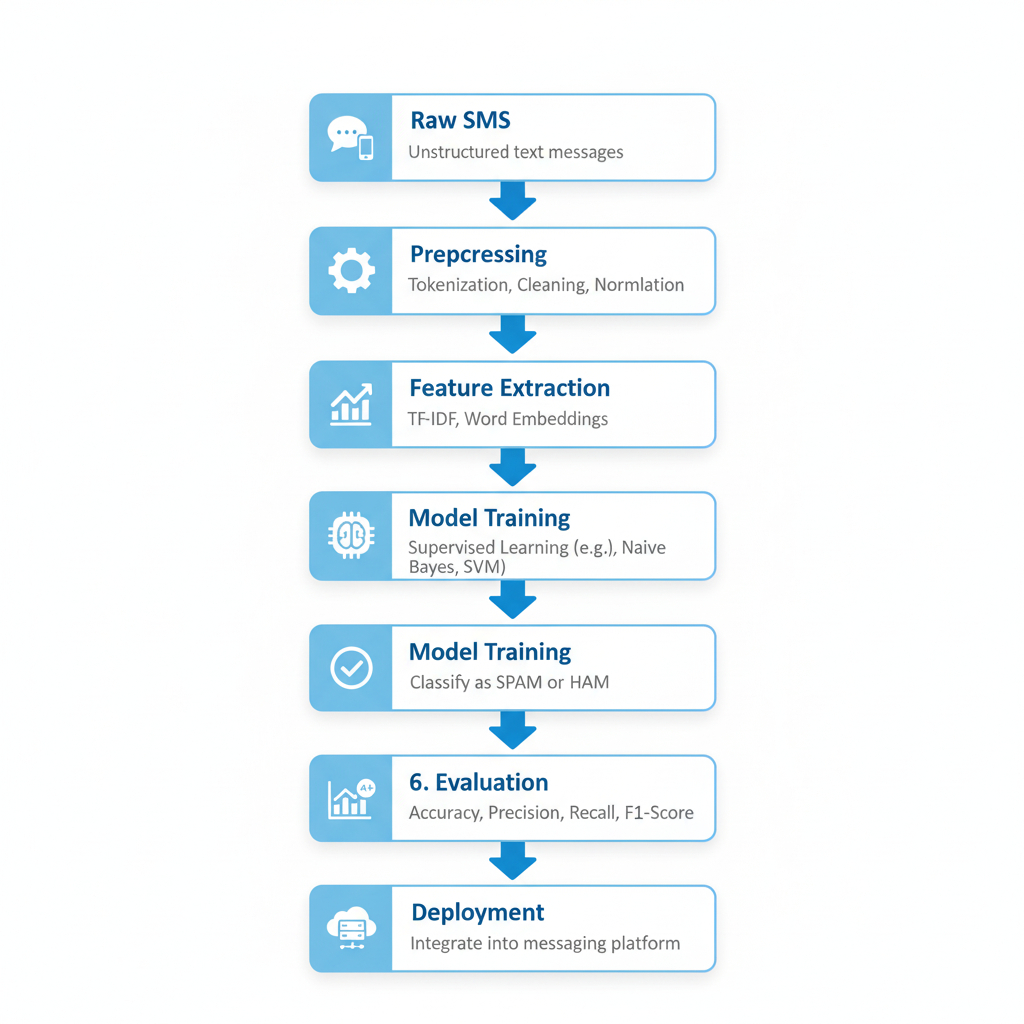
# 3. Proposed Solution

## Approach / Methodology

The project applies supervised learning techniques using text-based features. Models such as Naive Bayes, Logistic Regression, Random Forest, and SVM are trained and compared.

## Pipeline Overview

Raw SMS → Preprocessing → Feature Extraction → Model Training → Prediction → Evaluation → Deployment



## Justification of Choices

TF-IDF Vectorizer used for effective text representation. Naive Bayes chosen as baseline due to efficiency on text data. SVM / Logistic Regression tested for performance comparison. Streamlit used for building a simple, interactive web interface.

# 4. Features

## Functional Features

* Accepts raw SMS text input and predicts 'Spam' or 'Ham'.
* Provides prediction confidence score.
* Supports batch classification for multiple messages.
* Displays key influencing words (interpretability).
* Web interface using Streamlit.

## Non-Functional Features

* Fast and accurate predictions.
* Lightweight and scalable.
* Easy to use and maintain.
* Supports future model updates.

# 5. Technologies & Tools

Language: Python

Data Handling: pandas, numpy

Visualization: matplotlib, seaborn

NLP / Text Processing: nltk, re, spaCy

Feature Extraction: scikit-learn (CountVectorizer, TfidfVectorizer)

Machine Learning: scikit-learn (Naive Bayes, Logistic Regression, Random Forest, SVM)

Model Saving: pickle, joblib

UI / Deployment: Streamlit

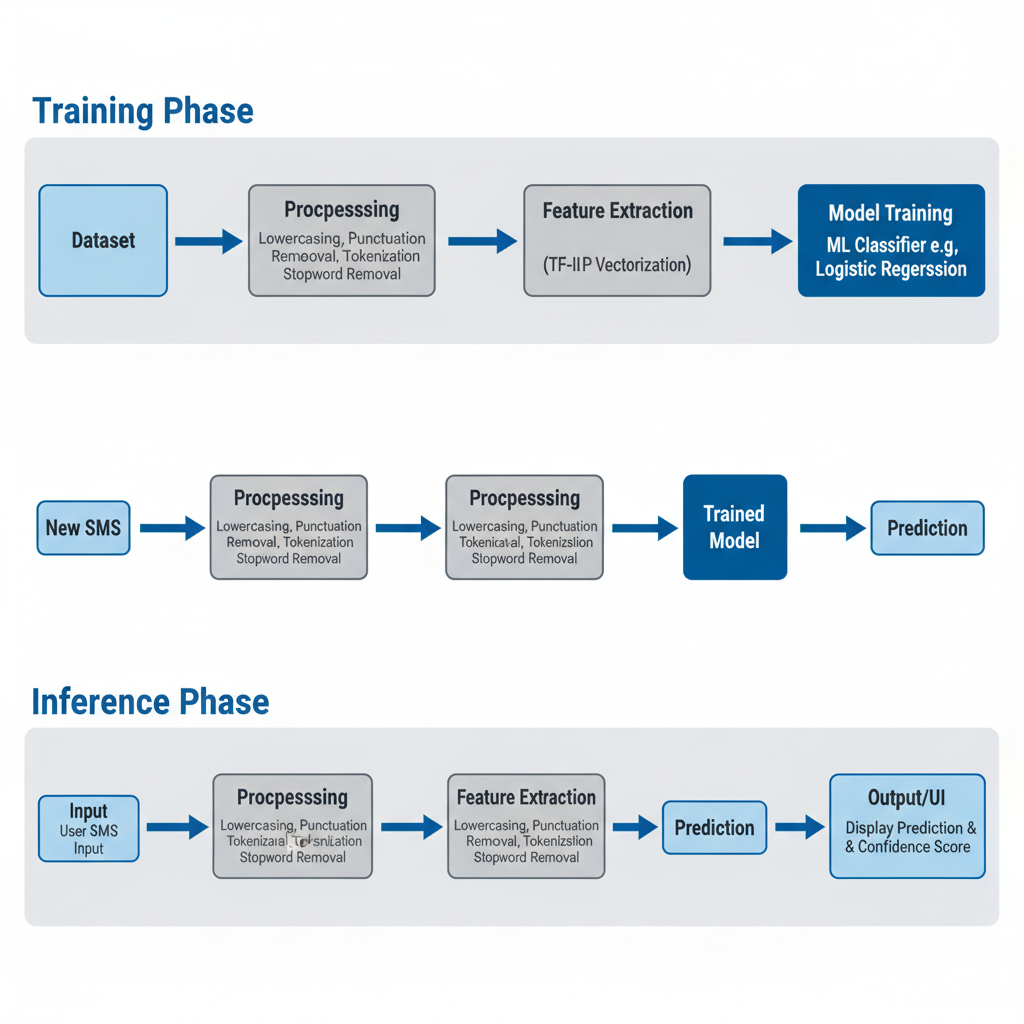
Development Environment: Jupyter Notebook, VS Code

Version Control: Git / GitHub

# 6. System Architecture

## Architecture Diagram

Input → Preprocessing → Feature Extraction → Model → Prediction → Output/UI



## Component Descriptions

Input Module: Accepts user SMS input.  
Preprocessing Module: Cleans text (lowercasing, punctuation removal, tokenization, stopword removal).  
Feature Extraction: Converts text to numerical form using TF-IDF.  
Model: Trained ML classifier (e.g., Logistic Regression).  
Output/UI: Displays prediction and confidence score.

## Data Flow

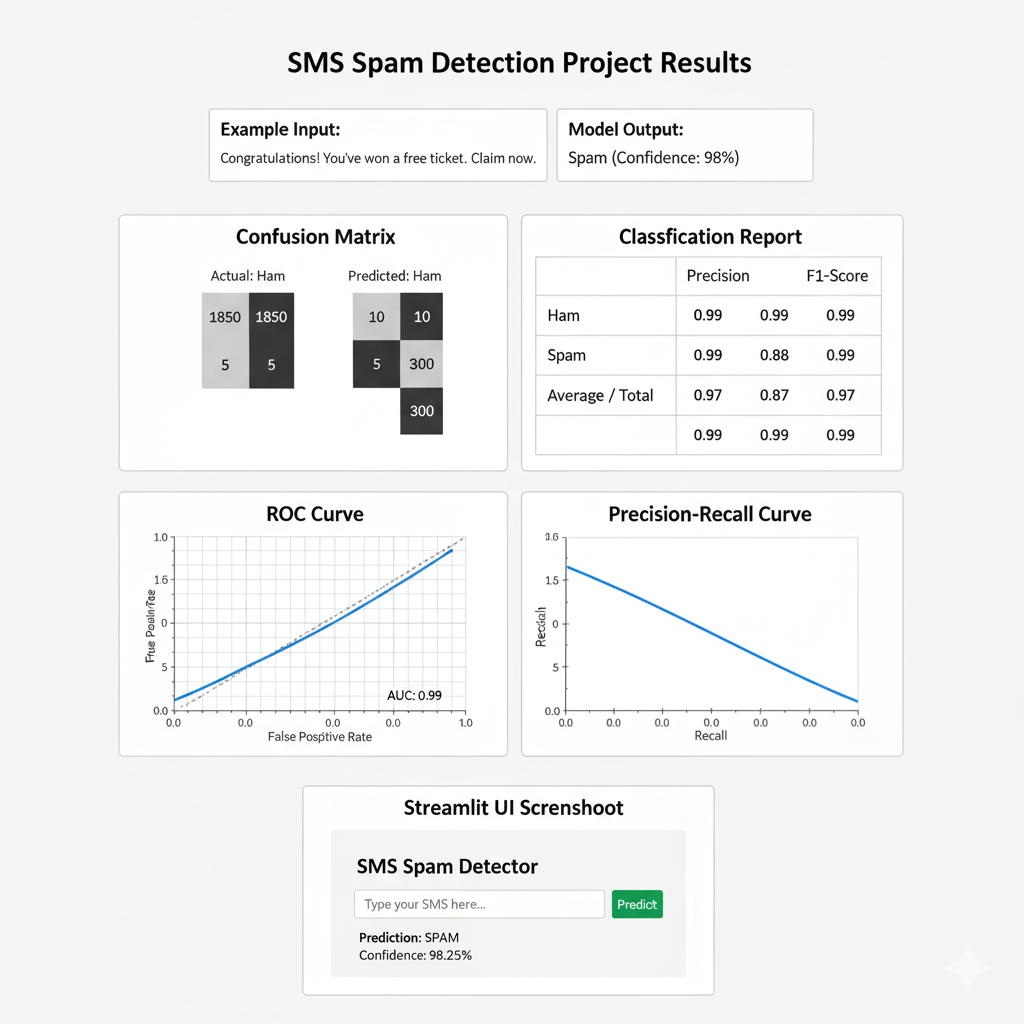
Training Phase: Dataset → Preprocessing → Feature Extraction → Model Training  
Inference Phase: New SMS → Preprocessing → Feature Extraction → Trained Model → Prediction

# 7. Implementation Steps

1. Data Acquisition: Load UCI SMS Spam dataset.
2. Exploratory Data Analysis: Understand data distribution and class imbalance.
3. Text Preprocessing: Lowercasing, punctuation removal, tokenization, stopword removal, stemming/lemmatization.
4. Feature Extraction: Apply CountVectorizer and TF-IDF.
5. Train-Test Split: 80:20 ratio for model evaluation.
6. Model Training: Train baseline (Naive Bayes) and compare with Logistic Regression, Random Forest, SVM.
7. Hyperparameter Tuning: Use GridSearchCV for optimization.
8. Model Evaluation: Measure accuracy, precision, recall, F1-score, and confusion matrix.
9. Error Analysis: Analyze misclassified messages.
10. Save Model & Vectorizer: Using pickle/joblib.
11. Build Streamlit App: For real-time message classification.
12. Testing: On unseen data samples.
13. Documentation: Prepare report and visuals.

# 8. Output / Screenshots

Example Input: 'Congratulations! You’ve won a free ticket. Claim now.'  
Model Output: Spam (Confidence: 98%)



# 9. Advantages

* Automates detection of spam messages efficiently.
* High precision and recall scores achieved.
* Lightweight and scalable system.
* Easily retrainable for new data.
* User-friendly interface with quick predictions.
* Interpretability through word frequency visualization.

# 10. Future Enhancements

* Integration of deep learning models like BERT or LSTM.
* Multilingual spam detection.
* Cloud-based API deployment for large-scale use.
* Real-time detection on incoming SMS streams.
* Enhanced visualization and mobile app version.
* Handling extreme class imbalance with SMOTE or ensemble techniques.
* Categorize spam into phishing, advertisements, etc.

# 11. Conclusion

The SMS Spam Detection system successfully classifies text messages as spam or ham using supervised machine learning models. The project achieved promising accuracy with TF-IDF and Naive Bayes / Logistic Regression approaches. It demonstrated effective preprocessing, feature extraction, and model evaluation pipelines. Although challenges like slang and data imbalance persist, this project provides a strong foundation for advanced text classification systems. Future work includes deep learning models, multilingual datasets, and real-time deployment for broader usability.