A

PROJECT REPORT

ON

SMS SPAM COLLECTOR

Submitted towards partial fulfilment of the requirement
for the awards of the degree of

"BACHELOR OF COMPUTER APPLICATION"

Guided by
Mr.Vikas Kumar

Submitted by

Aryaman

Roll no – 231602010022

Department Of Computer Application

INDRAPRASTHA INSTITUTE OF MANAGEMENT & TECHNOLOGY UMHI KOTA SAHARANPUR

(Affiliated to Maa Shakubhari University Puwarka Saharanpur)

DECLARATION

This is to certify that the synopsis entitled “SMS Spam Filtering ” is an authentic work carried out for the partial fulfilment of the Bachelors Degree of Computer Application under the guidance of **Mr.Vikas Kumar** The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

Name – Aryaman

Roll No – 231602010022

ACKNOWLEDGEMENT

This project report on” **SMS SPAM FILTERING”** is the result of idea and suggestions to me by **MR. VIKAS KUMAR**

We have received unfailing encouragement and inspiration of **MR. VIKAS KUMAR** whose exceptional knowledge and unparalleled behavior is full of ardent inspiration in it. However, we can never adequate thank all those who have their assistance, guidance, cooperation criticism contributed to the improvement of this report. We are ebullient in expressing my intense in debtless heartiest gratitude to all of them.

Since performance feedback is essential for effective communication, mistakes and creative feedback of the report may be unhesitatingly communicated to me, who will be as far as possible duly acknowledged and most welcome.

In this report, whatever is beneficial comes from almighty, and whatever is faulty is mine.

NAME: ARYAMAN

ROLLNO:231602010022

Date 01/11/2025

Certificate

This certify that student **Mr. ARYAMAN (231602010022)** of **BCA Vth Semester , Indraprastha Institute Of Management Umhi Kota , Saharanpur ,** have collected information regarding their project entitled "**SMS SPAM FILTERING**" during period 01/09/2025 to 01/11/2025 under my guidance.

During this period his conduct was good. We wish him all the best for future endeavors.

Index

INTRODUCTION

In today's digital era, Short Message Service (SMS) has become one of the most widely used methods of communication. People receive dozens of SMS daily including bank notifications, OTPs, promotional offers, personal communications, and service alerts. However, along with useful SMS, users are also flooded with spam messages, which include advertisements, fake lottery claims, credit card scams, phishing links, job frauds and other malicious content. These spam messages are not only irritating but can also lead to financial loss, privacy leakage and cyber frauds

Traditional mobile SMS inboxes do not have smart mechanisms to differentiate between spam and genuine messages. As a result, users often get confused and sometimes become victims of cybercrime. Therefore, there is a strong need for an automated SMS Spam Detection System which can intelligently identify whether a received SMS is "Spam" or "Ham" (Not Spam).

This project "SMS Spam Detection System" is developed using Machine Learning and Natural Language Processing techniques. The system learns from past spam and non-spam messages and builds an intelligent model that can automatically classify a new SMS based on pattern recognition. The system uses TF-IDF text vectorization and the Multinomial Naive Bayes algorithm, which is highly efficient for text classification problems. A simple and attractive GUI is provided using Python Tkinter so that users can easily use the system and check messages without technical knowledge.

Thus, this project is a practical implementation of Machine Learning concepts and demonstrates how technology can help solve real-world problems and improve digital safety.

OBJECTIVE

The objective of this project is to design and develop an efficient, accurate and user-friendly SMS Spam Detection System using Machine Learning techniques.
The detailed objectives are:

To develop an intelligent system that can automatically identify whether an SMS is spam or not.

To analyze real-world SMS dataset and extract meaningful features using NLP techniques.

To apply TF-IDF vectorization to convert text data into numerical form suitable for Machine Learning.

To train and implement Multinomial Naive Bayes classifier for spam detection.

To build a Graphical User Interface (GUI) using Python Tkinter so that normal users can easily interact with the system.

To reduce the chances of fraud, scams and cybercrime by filtering spam messages.

To demonstrate the practical application of Machine Learning concepts in solving a real-life problem.

To make a system that works fast, gives accurate results and displays probability/confidence level for better understanding.

LIMITATION OF EXISTING SYSTEM

Before the development of smart spam detection systems, most mobile inboxes used very basic filtering mechanisms. These traditional systems mainly depended on:

- Keyword-based filtering
- Manually created rules
- Blacklist of numbers

However, these existing systems have several limitations:

1. They cannot learn new spam patterns automatically.
2. Spammers easily bypass keyword-based filters by using different spelling styles like “Fr33”, “W1n”, etc.
3. High chances of false detection where genuine messages are marked as spam or spam marked as normal.
4. Rules need to be manually updated regularly which is time-consuming.
5. No intelligent understanding of sentence meaning and context.
6. Limited accuracy and unreliable performance.
7. Do not provide probability or confidence level of prediction.

Therefore, there is a strong need for an automated, intelligent and learning-based system which can understand patterns from real data and detect spam messages more effectively. This leads to the development of Machine Learning based SMS Spam Detection System.

ADVANTAGES OF PROPOSED SYSTEM

The proposed SMS Spam Detection System provides several advantages over existing traditional systems:

1. Higher Accuracy: Machine Learning model is trained on thousands of real messages which helps it identify spam more accurately.
2. Intelligent Learning: Unlike rule-based systems, this model learns from data and understands patterns in spam.
3. Fast Processing: The system analyzes and predicts results within seconds.
4. Probability Based Result: It displays the confidence percentage of prediction.
5. User Friendly Interface: GUI designed using Tkinter makes it easy to use even for non-technical users.
6. Reduces Fraud Risk: Helps users identify suspicious messages and protects from scams.
7. Automatic Processing: No manual checking required.
8. Scalable: New data can be added to improve accuracy.
9. Lightweight Application: Does not require heavy hardware.

Thus, the proposed system is more powerful, reliable and intelligent than existing spam filtering techniques.

SOFTWARE AND HARDWARE REQUIRED

Hardware Requirements

- Processor: Intel / AMD Processor
- RAM: Minimum 4GB (8GB recommended)
- Hard Disk: Minimum 500MB free space
- Keyboard and Mouse
- Monitor / Laptop Display

Software Requirements

- Operating System: Windows 10 / 11
- Programming Language: Python
- IDE: VS Code / PyCharm / IDLE
- Python Libraries:
    - Pandas
    - NLTK
    - Scikit-Learn
    - Tkinter
    - Threading
    - Regex

These requirements are sufficient to develop, execute and run this project smoothly.

FRONTEND AND BACKEND USED

Frontend

- Python Tkinter is used as frontend.
- Provides graphical user interface.
- Allows user to type SMS easily.
- Displays output in attractive and readable format.
- Buttons and labels used for better interaction.

Backend

- Python programming language.
- Machine Learning logic implemented in backend.
- Dataset loading and preprocessing done.
- TF-IDF Vectorization converts text to numeric data.
- Multinomial Naive Bayes algorithm performs spam detection.
- Threading used for background model training.

Thus, frontend provides easy user interaction while backend performs all intelligent machine learning operations.

MODULES

This project is divided into the following modules:

1. Dataset Module

   □ Loads dataset

   □ Reads SMS and labels

2. Preprocessing Module

   □ Convert text to lowercase

   □ Remove punctuation

   □ Remove stopwords

   □ Prepare clean text

3. Feature Extraction Module

   □ Apply TF-IDF vectorization

   □ Convert text to numeric representation

4. Training Module

   □ Train Multinomial Naive Bayes model

   □ Learn pattern from dataset

5. Prediction Module

   □ Takes input SMS

   □ Predicts spam or ham

   □ Shows probability

6. GUI Module

   □ Takes input from user

   □ Displays result

DFD (DATA FLOW DIAGRAM)

Level 0 DFD

User → Inputs SMS

↓

System Processes SMS

↓

Spam / Ham Result

Level 1 DFD

User

↓

Enters SMS

↓

Preprocessing Unit

↓

Feature Extraction

↓

Machine Learning Model

↓

Result Output + Probability

ER DIAGRAM / SYSTEM WORKFLOW

Since this project is not database-based, ER Diagram is replaced by System Workflow Diagram.

Workflow:

Dataset

↓

Preprocessing

↓

TF-IDF Vectorization

↓

Model Training

↓

GUI Input

↓

Prediction

↓

Output

This shows how the model is built and how prediction is performed.

DATABASE TABLE (CONVERTED INTO DATASET DESCRIPTION)

Dataset Name: SMS Spam Collection Dataset

File Format: .txt / .csv

Total Messages: Around 5500+

Spam Messages: Around 700+

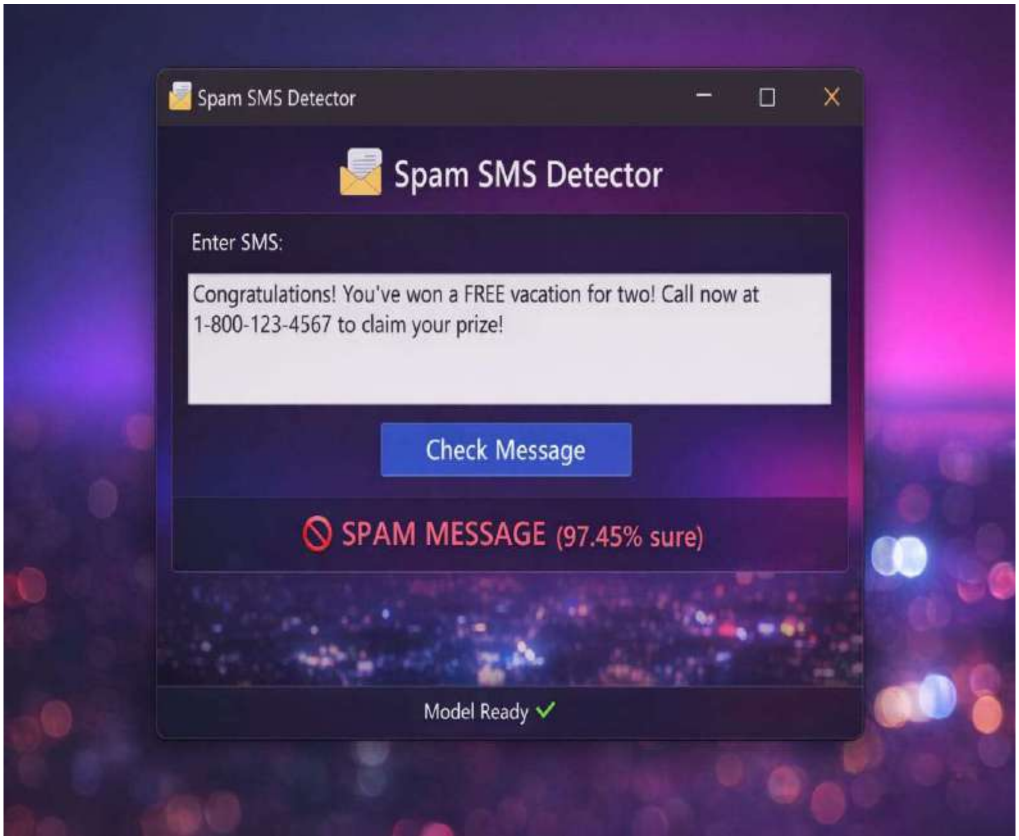
Ham Messages: Around 4700+

Columns:

1. Label – Spam or Ham
2. Message – SMS Text

Dataset contains real-world SMS and is highly reliable.

OUTPUT SCREENS



Enter SMS:

Congratulations! You've won a FREE vacation for two! Call now at 1-800-123-4567 to claim your prize!

🚫 SPAM MESSAGE (97.45% sure)

Model Ready ✓

CODING

```
import pandas as pd
import nltk
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
import tkinter as tk
from tkinter import messagebox
import threading
import re

nltk.download("stopwords", quiet=True)

# ------------ Preprocessing ------------
stop_words = set(stopwords.words("english"))

def clean_text(msg):
    msg = msg.lower()
    msg = re.sub(r"[^a-z0-9\s]", "", msg)
    words = msg.split()
    words = [w for w in words if w not in stop_words]
    return " ".join(words)

# ------------ Load & Train Model ------------
def train_model():
    global vectorizer, model

    try:
        df = pd.read_csv("SMSSpamCollection", sep="\t",
             names=["label", "message"])
    except Exception:
        root.after(0, lambda: status_var.set("Dataset not found ❌"))
        return

    df["label"] = df["label"].map({"ham": 0, "spam": 1})
    df["cleaned"] = df["message"].apply(clean_text)

    vectorizer = TfidfVectorizer()
    X = vectorizer.fit_transform(df["cleaned"])
    y = df["label"]
    model = MultinomialNB()
    model.fit(X, y)

    # Thread-safe UI update
    root.after(0, lambda: status_var.set("Model Ready ✔"))
    root.after(0, lambda: btn.config(state="normal"))


# ------------ Predict Function ------------
def predict_sms():
    if model is None:
        messagebox.showinfo("Please Wait", "Model is still training...")
        return

    msg = text_box.get("1.0", "end").strip()
    if msg == "":
        messagebox.showwarning("Empty", "Please enter a message.")
        return

    cleaned = clean_text(msg)
    vect = vectorizer.transform([cleaned])
    pred = model.predict(vect)[0]
    prob = model.predict_proba(vect)[0][1]   # Spam probability

    if pred == 1:
        result_label.config(
            text=f"🚫 SPAM MESSAGE  ({prob*100:.2f}% sure)",
            fg="#ff5c5c"
        )
    else:
        result_label.config(
            text=f"✔ NOT SPAM  ({(1-prob)*100:.2f}% sure)",
            fg="#4dff91"
        )


# ------------ GUI ------------
root = tk.Tk()
root.title("Spam SMS Detector - Stylish GUI")
root.geometry("650x460")
root.resizable(False, False)
root.configure(bg="#1e1e2f")

title = tk.Label(root, text="📩 Spam SMS Detector",
                 font=("Segoe UI", 22, "bold"), bg="#1e1e2f", fg="white")
title.pack(pady=15)

frame = tk.Frame(root, bg="#2c2c3e", bd=0)
frame.pack(padx=20, pady=10)

label = tk.Label(frame, text="Enter SMS:", font=("Segoe UI", 12),
                 bg="#2c2c3e", fg="#c6c6d1")
label.pack(anchor="w", padx=10, pady=5)

text_box = tk.Text(frame, width=65, height=6, font=("Segoe UI", 11),
                   bg="#f5f5f9", fg="#222", bd=0, padx=10, pady=10)
text_box.pack(padx=10, pady=5)

btn = tk.Button(root, text="Check Message", command=predict_sms,
                font=("Segoe UI", 12, "bold"), width=20,
                bg="#4e6cff", fg="white", activebackground="#364fc7",
                relief="flat", state="disabled")
btn.pack(pady=10)

result_label = tk.Label(root, text="", font=("Segoe UI", 18, "bold"),
                        bg="#1e1e2f", fg="#4dff91")
result_label.pack(pady=10)

status_var = tk.StringVar()
status_var.set("Training model... Please wait ⏳ ")

status_bar = tk.Label(root, textvariable=status_var, font=("Segoe UI",
              10),
                      bg="#1e1e2f", fg="#bbbbbb")
status_bar.pack(side="bottom", pady=10)

# Train model in background
model = None
threading.Thread(target=train_model, daemon=True).start()

              root.mainloop()
```

CONCLUSION

The SMS Spam Detection System developed in this project successfully demonstrates the practical application of Machine Learning and Natural Language Processing techniques to solve a real-world problem. With the rapid increase in mobile communication, spam messages have become a serious issue, causing inconvenience, privacy risks, and financial fraud. This project effectively addresses this problem by providing an intelligent system capable of automatically identifying spam messages with high accuracy and reliability.

The system uses a real-world SMS dataset and applies proper data preprocessing techniques such as text cleaning, stopword removal, and normalization. Feature extraction is performed using TF-IDF vectorization, which helps in identifying the importance of words within messages. The Multinomial Naive Bayes algorithm is used for classification due to its efficiency and suitability for text-based data. The trained model produces fast and accurate predictions and also provides a probability score, which increases transparency and user trust in the system's decision.

A major strength of this project is its user-friendly graphical interface developed using Python Tkinter. The GUI allows users to easily enter any SMS text and instantly receive the classification result without requiring any technical knowledge. The lightweight nature of the application ensures smooth performance even on systems with limited hardware resources. Background model training and thread-safe execution further improve usability and stability.

Overall, this project proves that Machine Learning-based approaches are far more effective than traditional rule-based spam filtering systems. The SMS Spam Detection System not only improves user experience but also contributes to digital safety by helping users avoid fraudulent and malicious messages.

FUTURE SCOPE

The SMS Spam Detection System developed in this project works efficiently and provides accurate results for detecting spam messages. However, there is still significant scope for improvement and expansion in the future. With advancements in Machine Learning and communication technologies, this system can be further enhanced to become more powerful and practical for real-world use.

One important future enhancement is the inclusion of Hindi and Hinglish language support. In India, many spam messages are received in regional and mixed languages. By training the system on multilingual datasets, the accuracy and usability of the system can be greatly improved.

Another future scope is the development of an Android mobile application. Integrating this system into a mobile app would allow automatic scanning of incoming SMS messages and real-time spam alerts, making the system more user-friendly and effective.

The system can also be expanded into a web-based platform, enabling users to check SMS content online and allowing easy integration with other messaging services. Additionally, the use of advanced Machine Learning or Deep Learning models such as LSTM or BERT can further improve accuracy by understanding message context more effectively.

In the future, this system can be deployed at the telecom network level to filter spam messages before they reach users. Overall, the project has strong future potential and can be enhanced into a complete spam filtering solution with further research and development.

BIBLIOGRAPHY

Multilingual Support

Future versions of the system can support Hindi and Hinglish languages. This will help in detecting spam messages written in regional or mixed languages, which are commonly used in India.

Android Mobile Application

The system can be converted into an Android application that automatically scans incoming SMS messages and alerts users in real time about spam messages.

Web-Based Spam Detection System

A web application can be developed so users can check SMS content online and organizations can integrate spam detection into their messaging services.

Use of Advanced Machine Learning Models

More advanced algorithms such as LSTM, RNN, or BERT can be implemented to improve accuracy by understanding the context and meaning of messages more effectively.

Real-Time SMS Monitoring

The system can be enhanced to monitor SMS messages in real time and classify them automatically without manual input.

Continuous Learning Mechanism

The model can be updated regularly with new data so that it can adapt to changing spam patterns and remain effective over time.

Telecom-Level Integration

In the future, the system can be deployed at the telecom network level to filter spam messages before they reach users.

Cloud-Based Deployment

The model can be hosted on cloud servers to handle large volumes of SMS data and provide scalable spam detection services.

Improved User Interface
The GUI can be further enhanced with better design, themes, and additional features for improved user experience.

Security and Privacy Enhancements
Additional security measures can be implemented to ensure that user data and SMS content remain safe and private.