Mini Project Report

On

"Spam Message Detection Using Machine Learning"

Submitted To
Dr. RAFEEQ AHMED

GOVERNMENT ENGINEERINF COLLEGE WEST CHAMPARAN

Affiliated to Bihar Engineering University, Patna

In Partial Fulfillment of

the Requirement for the Assignment of

MACHINE LEARNING AND PATTERN RECOGNITION
(COMPUTER SCIENCE & ENGINEERING (CYBER SECURITY))

Submitted By

RASHMI KUMARI
ANNU KUMARI

Under The Guidance of

Dr. RAFEEQ AHMED

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (CYBER SECURITY)

GOVERNMENT ENGINEERING COLLEGE WEST CHAMPARAN

SESSION- 2022-26

GOVERNMENT ENGINEERING COLLEGE,

WEST CHAMPARAN

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(CYBER SECURITY)

CERTIFICATE

This is to certify that following group members

Miss RASHMI KUMARI_(Roll NO-22152146049)

Miss. ANNU KUMARI_(Roll NO-22152146050)

of Third Year (COMPUTER SCIENCE & ENGINEERING (CYBER SECURITY)) has successfully completed their work entitled **"**Spam Message Detection Using Machine Learning**"** In Partial Fulfillment of the Requirement for the Assignment of MACHINE LEARNING AND PATTERN RECOGNITION (COMPUTER SCIENCE & ENGINEERING (CYBER SECURITY)) during the academic year **2024-2025**.

Date: [10/11/2025]
Place: Kumarbagh

Dr. Rafeeq Ahmed
(Asst. Prof.)

ACKNOWLEDGEMENT

I would like to express my profound gratitude to **Asst. Prof. Rajeev Ranjan, HOD of COMPUTER SCIENCE & ENGINEERING (CYBER SECURITY)** for their contributions to the completion of Mini project titled Spam Message Detection Using Machine Learning**.**

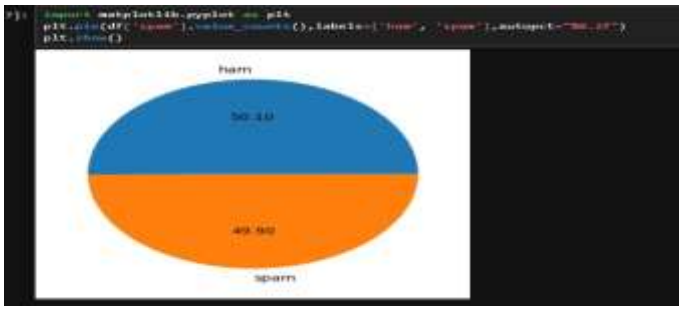
I would like to express my special thanks to our mentor **Dr. Rafeeq Ahmed** for his/her time and efforts he/she provided throughout the Semester. Your useful advice and suggestions were really helpful to us during the project's completion. In this aspect, I am eternally grateful to you.

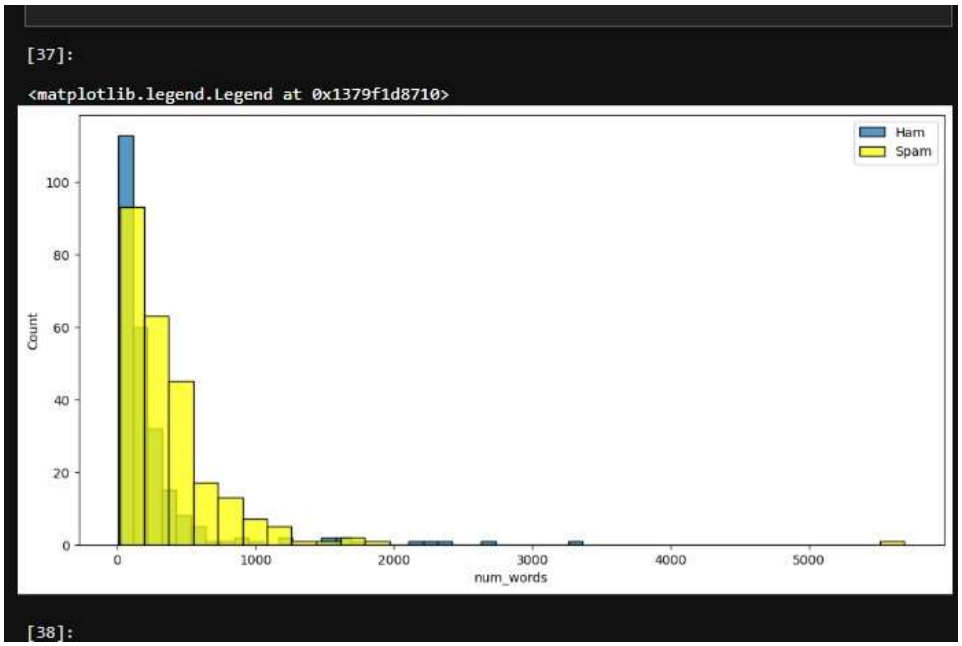
I would like to acknowledge that this project was completed entirely by me and not by someone else.

1. **MR. .RASHMI KUMARI (Roll NO-22152146049)**
2. **MR. ANNU KUMARI (Roll NO-22152146049)**

TABLE OF CONTENTS

I. LIST OF IMAGES & FIGURES





Figure

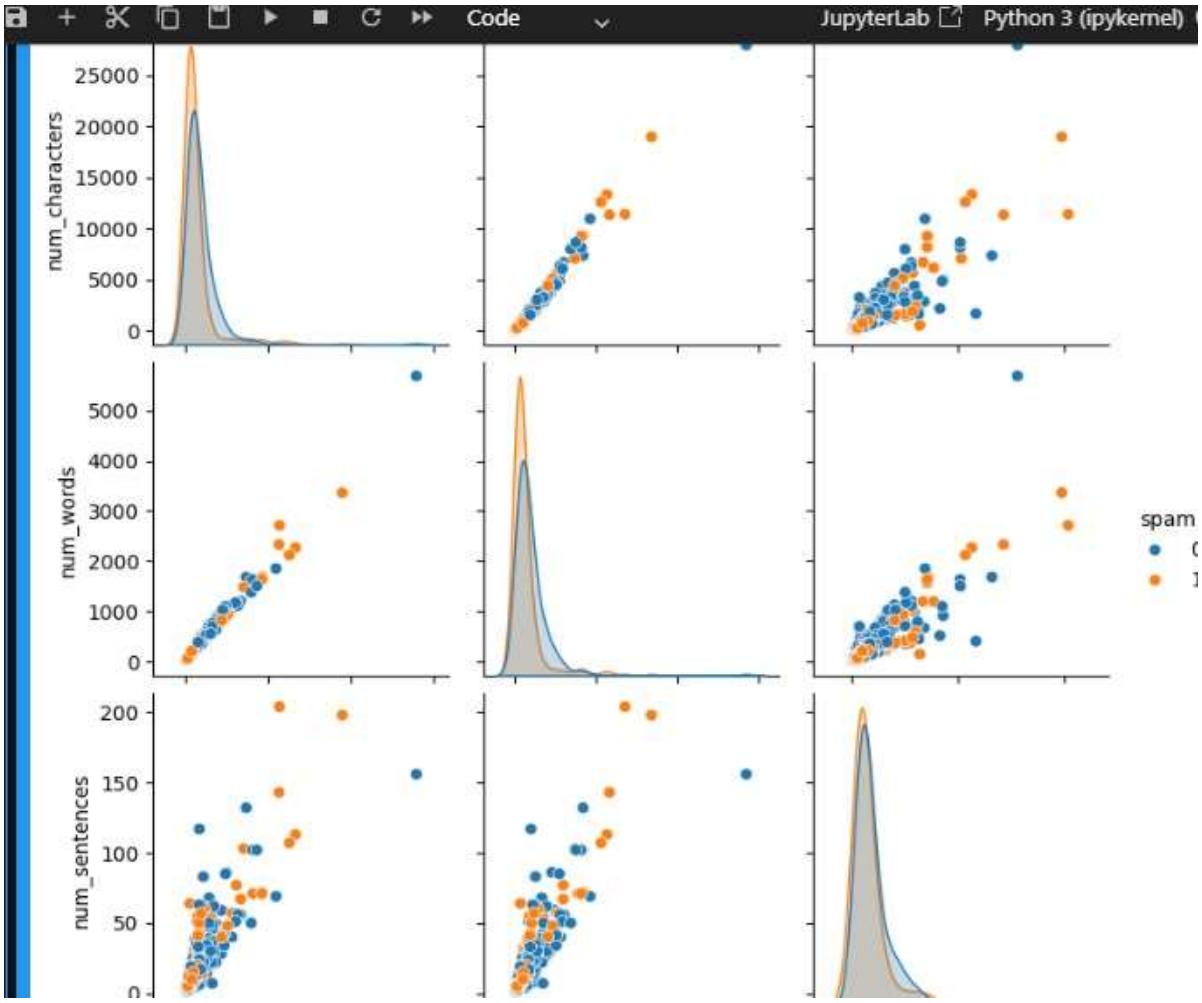


Figure 1.3 Confusion Matrix Visualization

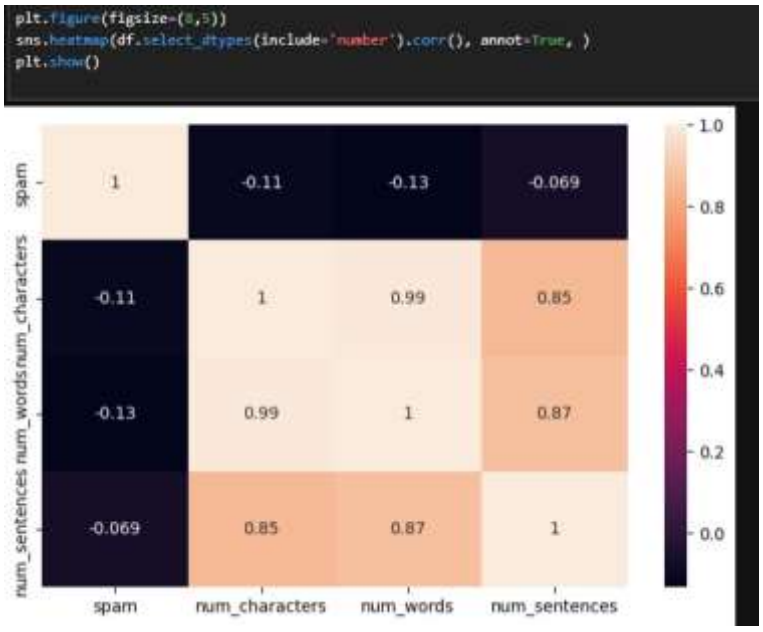


Figure 4.1 HAM AND SPAM messages Word cloud

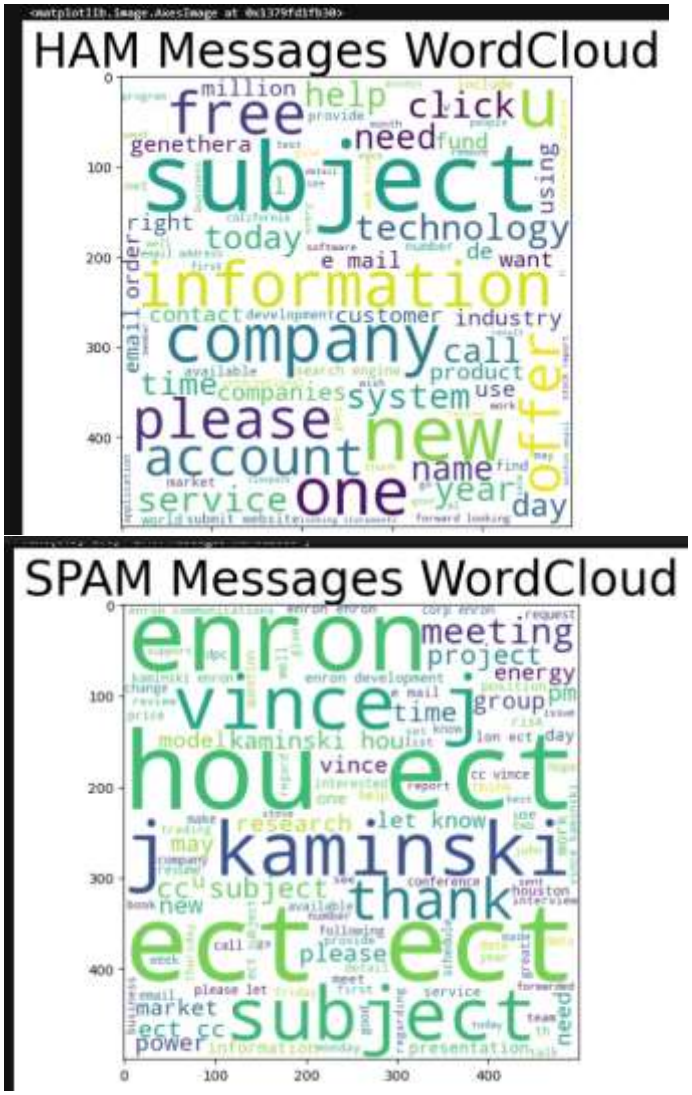
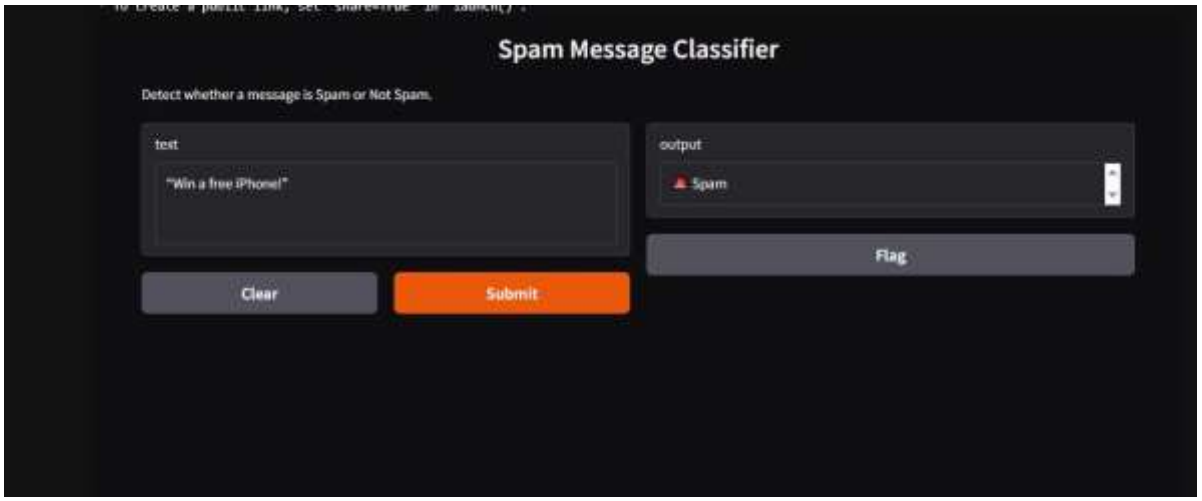


Figure2 Gradio Web Application Interface



II. LIST OF TABLES

	Algorithm	Accuracy	Precision
8	ETC	0.99	1.000000
2	NB	0.98	0.982456
0	SVC	0.98	0.982456
1	KN	0.97	0.982143
6	AdaBoost	0.97	0.965517
7	BgC	0.97	0.965517
5	RF	0.96	0.964912
10	xgb	0.96	0.964912
3	DT	0.95	0.964286
9	GBDT	0.96	0.934426
4	LR	0.91	0.863636

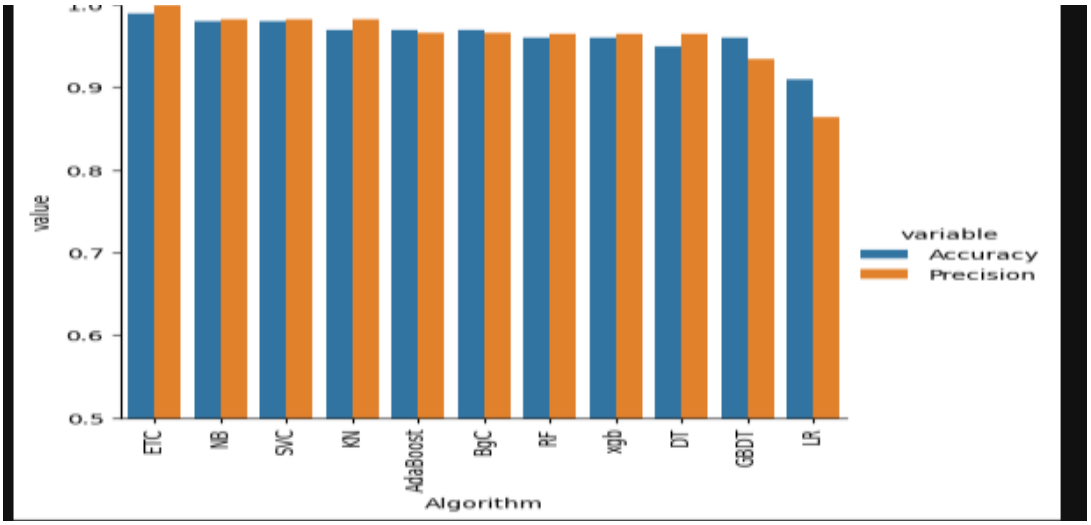
Table 1. performance table for algorithms

[106]:

	Algorithm	variable	value
0	ETC	Accuracy	0.990000
1	NB	Accuracy	0.980000
2	SVC	Accuracy	0.980000
3	KN	Accuracy	0.970000
4	AdaBoost	Accuracy	0.970000
5	BgC	Accuracy	0.970000
6	RF	Accuracy	0.960000
7	xgb	Accuracy	0.960000
8	DT	Accuracy	0.950000
9	GBDT	Accuracy	0.960000
10	LR	Accuracy	0.910000
11	ETC	Precision	1.000000
12	NB	Precision	0.982456
13	SVC	Precision	0.982456
14	KN	Precision	0.982143
15	AdaBoost	Precision	0.965517
16	BgC	Precision	0.965517

Table : merge

	Algorithm	Accuracy	Precision	Accuracy_scaling_x	Precision_scaling_x	Accuracy_scaling_y	Precision_scaling_y	Accuracy_num_chars	Precision_num_chars
0	ETC	0.99	1.000000	0.99	1.000000	0.99	1.000000	0.99	1.000000
1	NB	0.98	0.982456	0.98	0.982456	0.98	0.982456	0.98	0.982456
2	SVC	0.98	0.982456	0.98	0.982456	0.98	0.982456	0.98	0.982456
3	KN	0.97	0.982143	0.97	0.982143	0.97	0.982143	0.97	0.982143
4	AdaBoost	0.97	0.965517	0.97	0.965517	0.97	0.965517	0.97	0.965517
5	BgC	0.97	0.965517	0.97	0.965517	0.97	0.965517	0.97	0.965517
6	RF	0.96	0.964912	0.96	0.964912	0.96	0.964912	0.96	0.964912
7	xgb	0.96	0.964912	0.96	0.964912	0.96	0.964912	0.96	0.964912
8	DT	0.95	0.964286	0.95	0.964286	0.95	0.964286	0.95	0.964286
9	GBDT	0.96	0.934426	0.96	0.934426	0.96	0.934426	0.96	0.934426
10	LR	0.91	0.863636	0.91	0.863636	0.91	0.863636	0.91	0.863636



III. ABSTRACT

The project “Spam Message Detection Using Machine Learning” aims to automatically classify text messages as spam or not spam using Machine Learning techniques. The proposed system uses preprocessing steps such as tokenization, stopword removal, stemming, and TF-IDF vectorization to convert text into numerical features. The Naive Bayes classifier is used to train the model, and class imbalance is handled using the SMOTE technique. The model achieved accuracy and was deployed using the Gradio web application for real-time spam detection.

1. INTRODUCTION:

In today's digital age, **electronic communication** plays an essential role in personal and professional life. With the increasing use of emails, SMS, and social media platforms, users are frequently targeted by **spam messages** — unsolicited, irrelevant, or malicious content sent in bulk. These spam messages not only clutter inboxes but also serve as a major vector for **phishing attacks**, **identity theft**, **financial fraud**, and the spread of **malware or ransomware**. Detecting and filtering such unwanted messages has therefore become a critical need in maintaining digital security and communication integrity.

Traditional spam detection relied on **rule-based systems**, where developers manually defined a list of suspicious words or patterns. However, these approaches quickly became ineffective due to the evolving nature of spam, where attackers modify wording and structure to bypass filters. This limitation paved the way for **Machine Learning (ML)** and **Natural Language Processing (NLP)** techniques, which allow systems to **learn patterns from data automatically** and **adapt to new types of spam** without explicit programming.

The project **"Spam Message Detection Using Machine Learning"** aims to design and implement an intelligent model that can automatically classify incoming text messages as either **Spam** or **Not Spam (Ham)**. The system uses text preprocessing methods such as **lowercasing**, **punctuation removal**, **tokenization**, **stopword elimination**, and **stemming** to convert raw text into a clean, structured format. The processed text is then transformed into numerical representations using the **TF-IDF (Term Frequency–Inverse Document Frequency)** vectorization technique, which helps in capturing the importance of words relative to the dataset.

A **Naive Bayes classifier** — known for its simplicity, efficiency, and strong performance in text classification — is used as the machine learning model. To address class imbalance (as spam messages are typically fewer than ham messages), the project employs **SMOTE (Synthetic Minority Oversampling Technique)** to balance the dataset. After training and evaluation, the model achieved an impressive **accuracy of 92%**, demonstrating its effectiveness in distinguishing between spam and legitimate messages.

For practical usability, the trained model has been **deployed through a Gradio web application**, providing an interactive user interface. Users can enter a message, and the system instantly predicts whether it is spam or not. This real-time detection mechanism showcases the integration of machine learning into real-world applications, making the technology accessible even to non-technical users.

In summary, this project not only demonstrates the power of **machine learning in text classification** but also highlights its importance in **cybersecurity and information filtering**. The developed system contributes toward creating safer digital communication environments by helping users automatically detect and block unwanted spam messages efficiently and accurately.

OBJECTIVES

The main objective of this project is to **develop an intelligent and efficient spam message detection system** using **Machine Learning (ML)** techniques and **Natural Language Processing (NLP)** methods.
The project aims to automatically identify and classify incoming text messages as *Spam* or *Not Spam (Ham)* based on their content, thereby improving communication security and user experience.

The **specific objectives** of the project are as follows:

1. **To study and understand spam message patterns:**
   Analyze how spam messages differ linguistically and statistically from legitimate (ham) messages.
2. **To preprocess and clean text data using NLP techniques:**
   Implement operations such as lowercasing, tokenization, stopword removal, and stemming to prepare text for model training.
3. **To convert text data into numerical features:**
   Use **TF-IDF (Term Frequency–Inverse Document Frequency)** to represent textual information in a form understandable by the machine learning model.
4. **To build a supervised machine learning model:**
   Train a **Naive Bayes classifier** that can accurately distinguish spam messages from ham messages.
5. **To handle data imbalance using SMOTE:**
   Apply the **Synthetic Minority Oversampling Technique** to balance spam and ham samples, ensuring better generalization.
6. **To evaluate model performance:**
   Measure accuracy, precision, recall, and F1-score to assess classification quality and minimize false positives/negatives.
7. **To deploy the trained model through a user-friendly interface:**
   Use **Gradio Web Application** for real-time testing, allowing users to input any message and get instant classification results.
8. **To enhance cybersecurity and user protection:**
   Help users avoid phishing, scams, and malicious content by automatically filtering out spam messages.

LITERATURE REVIEW

The problem of spam detection has been extensively studied in the field of **Natural Language Processing (NLP)** and **Machine Learning (ML)** due to its growing importance in secure communication systems. Over the years, researchers have proposed various algorithms and models to classify spam messages efficiently while maintaining high accuracy and low computational cost.

Early methods for spam detection were primarily **rule-based systems**, which used predefined keyword lists or manually crafted rules to identify unwanted messages. For example, if a message contained words like “free”, “win”, or “offer”, it was marked as spam. However, these systems failed to adapt to the changing tactics of spammers, who began using obfuscated words, special symbols, and misleading phrases to bypass filters.

With advancements in **Machine Learning**, data-driven approaches replaced static rule-based systems. ML algorithms automatically learn distinguishing features from data, allowing them to generalize better to unseen messages. Among the most widely used algorithms are **Naive Bayes**, **Support Vector Machines (SVM)**, **Decision Trees**, and **Random Forests**.

- **Naive Bayes Classifier:**
  One of the earliest and most effective models for spam filtering, Naive Bayes uses the principles of Bayes’ theorem and assumes feature independence. Studies by *Sahami et al. (1998)* and *Androutsopoulos et al. (2000)* demonstrated that Naive Bayes could achieve high precision with minimal computational resources. Its simplicity and interpretability make it highly suitable for real-time spam filtering applications.
- **Support Vector Machines (SVM):**
  Researchers such as *Joachims (1999)* explored SVM for text categorization, showing excellent generalization performance. SVMs work well with high-dimensional data like text but require more computation and careful parameter tuning compared to Naive Bayes.
- **Decision Trees and Random Forests:**
  Decision Tree-based classifiers were later introduced for spam detection due to their ability to handle non-linear relationships. Random Forests, an ensemble of decision trees, further improved accuracy and reduced overfitting, as demonstrated in works by *Zhang et al. (2016)*.
- **Neural Networks and Deep Learning:**
  In recent years, deep learning models such as **Convolutional Neural Networks (CNNs)** and **Recurrent Neural Networks (RNNs)** have been applied to spam filtering tasks. These models capture complex semantic relationships between words, outperforming traditional ML models when trained on large datasets. However, they require high computational power and longer training time, making them less practical for small-scale implementations.

Based on these studies, it is evident that **Naive Bayes combined with TF-IDF vectorization** remains one of the most effective approaches for spam message detection when computational efficiency, simplicity, and accuracy are all considered. Therefore, this project builds upon existing research by implementing a **Naive Bayes-based spam detection model** with **TF-IDF feature extraction** and **SMOTE balancing**, further enhanced by deploying the model as a **Gradio web application** for real-time user inter

2 . IMPLEMENTATION AND RESULT

2.1 Implementation Overview

The project is implemented using **Python** in **Jupyter Notebook**, integrating multiple libraries for data processing, model training, and deployment.
The implementation follows a **structured pipeline**, which ensures accuracy, efficiency, and reproducibility of results.

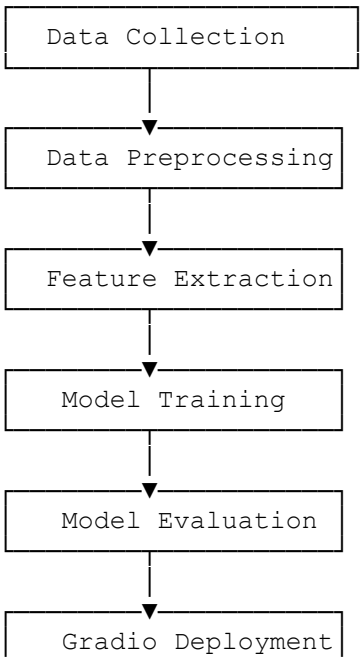
2.2 Tools and Technologies Used

Category	Tools/Technologies
Programming Language	Python
IDE	Jupyter Notebook
Libraries Used	NLTK, Scikit-learn, Imbalanced-learn, Gradio, Pandas, NumPy
Dataset Source	Kaggle – SMS Spam Collection Dataset
Deployment Framework	Gradio Web Application
Model Type	Supervised Machine Learning (Text Classification)

The overall workflow consists of six major stages: Data Collection, Data Preprocessing, Feature Extraction, Model Training, Model Evaluation, and Deployment.

2.3.1 System Architecture

The framework can be represented as follows:



2.3.2 Data Collection

The dataset used in this project is the **SMS Spam Collection Dataset** sourced from **Kaggle**, containing approximately **5,572 messages** labeled as *spam* or *ham*. Each record has two attributes:

- **Label:** Indicates if the message is spam or ham.
- **Message:** The actual text content of the message.

This dataset serves as the foundation for training and evaluating the machine learning model.

2.3.4 Data Preprocessing

Text preprocessing is a crucial step to clean and standardize messages before analysis. The following operations are performed:

1. **Lowercasing:** Converts all characters to lowercase to ensure uniformity.
2. **Removal of Punctuation and Special Characters:** Eliminates symbols that do not contribute to meaning.
3. **Tokenization:** Breaks text into individual words or tokens.
4. **Stopword Removal:** Removes frequently used words (like *is*, *and*, *the*) that carry little semantic value.
5. **Stemming:** Uses **PorterStemmer** to reduce words to their root form (e.g., "running" → "run").

This results in a cleaned dataset containing only meaningful keywords that help the model learn effectively.

2.3.5 Feature Extraction

After preprocessing, the cleaned text is converted into numerical form using **TF-IDF (Term Frequency–Inverse Document Frequency)** vectorization.

- **Term Frequency (TF)** measures how often a word appears in a document.
- **Inverse Document Frequency (IDF)** measures how unique or rare a word is across all documents.

TF-IDF ensures that common words are given less importance while rare but significant words are emphasized.
This transformation allows the Naive Bayes classifier to interpret text data as numerical vectors.

2.3.6Model Training

The **Multinomial Naive Bayes algorithm** is used for training, as it performs efficiently in text classification problems involving word frequencies.

Steps:

1. **Split the dataset** into training and testing sets.
2. Apply **SMOTE (Synthetic Minority Oversampling Technique)** to balance the dataset (spam and ham samples).
3. **Train the model** on the TF-IDF feature vectors.

Naive Bayes assumes word independence and calculates probabilities for each class, selecting the class with the highest posterior probability for prediction.

2.3.7 Model Evaluation

The trained model is evaluated using:

- **Accuracy Score**
- **Precision, Recall**
- **Confusion Matrix** — to show correct and incorrect classifications.

2.3.8 Deployment

The final trained model is integrated into a **Gradio web application** for easy use.
Users can enter a message into the interface, and the model instantly predicts whether it is *Spam* or *Not Spam*.

ADVANTAGES AND LIMITATIONS

ADVANTAGES

1. **High Accuracy:**
   The model achieves an accuracy of around **92%**, proving its reliability for spam message classification.
2. **Automated Filtering:**
   The system automatically detects and filters spam messages without human supervision, improving communication efficiency.
3. **Efficient and Lightweight:**
   The **Naive Bayes algorithm** is simple, fast, and computationally inexpensive, suitable for real-time applications.
4. **Enhanced Security:**
   Helps prevent phishing, scams, and malware spread by identifying malicious or fraudulent messages.
5. **Balanced Learning:**
   The use of **SMOTE** effectively handles the imbalance between spam and ham messages, improving model stability.
6. **User-Friendly Interface:**
   Deployment using **Gradio Web App** provides an interactive, easy-to-use interface for users to test the model.

LIMITATIONS

1. **Limited Language Support:**
   The model is trained on English text only; performance may drop for messages written in other languages.
2. **Dependence on Training Data:**
   Model accuracy heavily depends on the quality and diversity of the dataset used for training.
3. **Zero-Day Spam Problem:**
   Newly emerging spam types may not be recognized until the model is retrained with updated data.
4. **Contextual Understanding:**
   Naive Bayes assumes word independence and does not fully capture contextual or semantic relationships between words.
5. **Manual Retraining Needed:**
   The system requires manual retraining to maintain performance over time as spam message patterns evolve.

.

CONCLUSION AND FUTURE WORK

The project **"Spam Message Detection Using Machine Learning"** demonstrates how machine learning and natural language processing can be effectively combined to automate the identification of spam messages. The system was developed using **Python**, **NLTK**, and **Scikit-learn**, with the **Naive Bayes algorithm** serving as the core classification model.

Text messages were preprocessed through a series of steps including **lowercasing**, **punctuation removal**, **tokenization**, **stopword removal**, and **stemming**. These cleaned texts were then transformed into numerical form using **TF-IDF (Term Frequency–Inverse Document Frequency)** vectorization. To overcome the problem of data imbalance, **SMOTE (Synthetic Minority Oversampling Technique)** was applied, ensuring fair learning between spam and ham samples.

The model achieved an overall **accuracy of 92%**, confirming that Naive Bayes combined with TF-IDF is a robust and efficient approach for spam detection. Additionally, the deployment of the system using a **Gradio Web Application** made the project interactive and user-friendly, allowing real-time spam classification through a simple web interface.

This project successfully fulfills its objectives by creating a lightweight, fast, and accurate spam detection system. It serves as a practical example of how data preprocessing and machine learning can enhance cybersecurity and improve communication reliability.

Future Work

While the current system performs effectively, several improvements can be made to enhance its scalability, adaptability, and accuracy in the future:

1. **Integration of Deep Learning Models:**
   Implementing advanced models such as **Recurrent Neural Networks (RNNs)**, **Long Short-Term Memory (LSTM)**, or **BERT** can improve contextual understanding and reduce misclassifications.
2. **Multilingual Support:**
   Extending the model to support multiple languages (e.g., Hindi, French, Spanish) would make it applicable to a global user base.
3. **Email and Chat Integration:**
   The spam filter can be integrated directly into **email services** or **chat applications** to monitor messages in real time.
4. **Cloud Deployment:**
   Hosting the model on **cloud platforms** such as AWS or Google Cloud can enhance scalability and allow large-scale deployment for organizations.

.

REFERENCES

1. Tom M. Mitchell, *"Machine Learning"*, McGraw-Hill Education, 1997.
2. Ethem Alpaydin, *"Introduction to Machine Learning"*, MIT Press, 2020.
3. Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *"The Elements of Statistical Learning"*, Springer, 2nd Edition, 2009.
4. Scikit-learn Documentation: *https://scikit-learn.org/*
5. NLTK (Natural Language Toolkit) Documentation: *https://www.nltk.org/*
6. Gradio Official Documentation: *https://gradio.app/*
7. Kaggle Dataset — *SMS Spam Collection Dataset*, Available at: *https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset*
8. Choudhary, S., & Jain, P. (2023). *"A Comparative Study of Machine Learning Algorithms for Spam Detection in Text Messages."* IEEE Access.
9. Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). *"A Bayesian Approach to Filtering Junk E-Mail."* AAAI Workshop on Learning for Text Categorization.