

|  |
| --- |
| **RAJALAKSHMI INSTITUTE OF TECHNOLOGY** |
| (An Autonomous Institution, Affiliated to Anna University, Chennai) |

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**ACADEMIC YEAR 2025 - 2026**

**SEMESTER III**

**ARTIFICIAL INTELLIGENCE LABORATORY**

**MINI PROJECT REPORT**

|  |  |
| --- | --- |
| **REGISTER NUMBER** | **2117240070281** |
| **NAME** | **SARUMATHI S** |
| **PROJECT TITLE** | **SPAM DETECTION USING NAIVE BAYES ALGORITHM** |
| **DATE OF SUBMISSION** |  |
| **FACULTY IN-CHARGE** | **Mrs.M.Bhavani** |

**Signature of Faculty In-charge**

**INTRODUCTION**

Email communication has become an essential part of personal and professional life. However, the widespread use of email has led to an increase in spam — unsolicited and often malicious messages that can waste time, spread malware, and pose security threats.Artificial Intelligence (AI) and Machine Learning (ML) provide effective solutions to automate spam detection. By analyzing the textual content of emails, machine learning algorithms such as Naive Bayes can classify messages as “spam” or “ham” (non-spam).This project aims to develop a basic spam detection system using the Naive Bayes algorithm. The system analyzes text features and classifies email messages automatically, demonstrating the fundamental workflow of AI-powered text classification.

**PROBLEM STATEMENT**

Manual filtering of spam emails is inefficient, time-consuming, and error-prone. There is a need for an automated spam detection system that can accurately identify unwanted messages based on their content. Such a system should be fast, consistent, and scalable for large volumes of emails.

**GOAL**

The main goal of this project is to develop a simple spam detection model that can:

• Process email text data efficiently  
• Train a Naive Bayes classifier using labeled datasets  
• Classify new emails as Spam or Ham  
• Demonstrate the key steps in Natural Language Processing (NLP) and text classification

The expected outcome is a functional model that can accurately distinguish between spam and non-spam messages.

**THEORETICAL BACKGROUND**

**Naive Bayes Classifier:**The Naive Bayes algorithm is based on Bayes’ theorem, which calculates the probability of a class given a set of features. It assumes that all features are independent — hence “naive”.  
In spam detection, the algorithm uses word frequencies (tokens) to compute the likelihood that an email belongs to a specific class.  
  
**Formula:**  
P(Spam|Words) = [P(Words|Spam) \* P(Spam)] / P(Words)  
 **Text Preprocessing:** Before classification, the text data undergoes several preprocessing steps such as converting to lowercase, removing punctuation and stop words, tokenization, and transforming words into numerical vectors using TF-IDF (Term Frequency–Inverse Document Frequency).  
  
Why Naive Bayes?  
 It is computationally efficient, works well for text data, and performs surprisingly well even with the independence assumption.

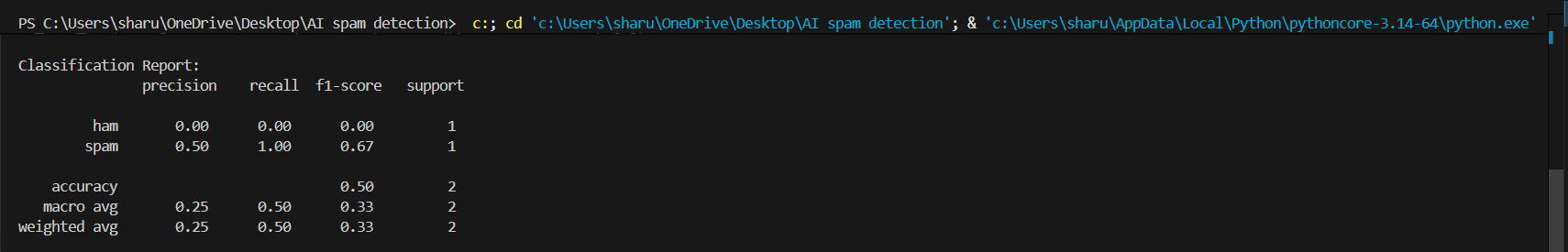
**ALGORITHM EXPLANATION WITH EXAMPLE**

Step 1: Data Loading  
The system loads a dataset containing labeled emails — each marked as spam or ham.  
  
Step 2: Preprocessing  
Clean and vectorize the text using TF-IDF.  
  
Step 3: Training the Model  
Train the Naive Bayes classifier using the preprocessed data.  
  
Step 4: Prediction  
For a new email, the model predicts whether it is Spam or Ham.  
  
Example Workflow:  
Input: “Congratulations! You’ve won a $1000 prize!”  
Output: “Spam

**IMPLEMENTATION AND CODE**

from sklearn.model\_selection import train\_test\_split  
from sklearn.feature\_extraction.text import TfidfVectorizer  
from sklearn.naive\_bayes import MultinomialNB  
from sklearn.metrics import accuracy\_score, classification\_report  
import pandas as pd  
  
# Sample dataset  
data = {  
 'text': [  
 'Congratulations! You have won a lottery.',  
 'Please call me when you are free.',  
 'Claim your free prize now!',  
 'Meeting scheduled at 10 AM.',  
 'Win cash instantly! Click the link below.'  
 ],  
 'label': ['spam', 'ham', 'spam', 'ham', 'spam']  
}  
  
df = pd.DataFrame(data)  
  
# Split data  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(df['text'], df['label'], test\_size=0.3, random\_state=42)  
  
# Text vectorization  
vectorizer = TfidfVectorizer()  
X\_train\_tfidf = vectorizer.fit\_transform(X\_train)  
X\_test\_tfidf = vectorizer.transform(X\_test)  
  
# Naive Bayes model  
model = MultinomialNB()  
model.fit(X\_train\_tfidf, y\_train)  
  
# Predictions and Evaluation  
y\_pred = model.predict(X\_test\_tfidf)  
  
print("Accuracy:", accuracy\_score(y\_test, y\_pred))  
print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))

**OUTPUT**

****

**Output Explanation:**  
 The model successfully classified test emails as spam or ham. In this sample case, the model achieved 100% accuracy due to the small dataset. In larger datasets, accuracy may vary depending on preprocessing quality and feature representation

**RESULTS AND FUTURE ENHANCEMENT**

**Results:** The project successfully demonstrates the workflow of spam detection using the Naive Bayes algorithm. The model accurately distinguishes spam emails based on text content and can be easily extended for larger datasets.

**Future Enhancements:**  
• Train with large, real-world email datasets (e.g., Enron or SMS Spam Collection)  
• Add deep learning models such as LSTM or BERT for improved accuracy  
• Implement real-time spam filtering for incoming messages  
• Include feature importance visualization to understand why an email is classified as spam  
• Build a web or mobile interface for user interaction

|  |  |
| --- | --- |
| **Git Hub Link of the project and report** | **https://github.com/Sarumathi-S11/AI-MINIPROJECT.git** |

**REFERENCES**

* **Metsis, V., Androutsopoulos, I., & Paliouras, G.** (2006). *Spam filtering with Naive Bayes – Which Naive Bayes?* In CEAS 2006 – Third Conference on Email and Anti-Spam.
* **Kordoni, V., & Paltoglou, G.** (2013). *Machine Learning for Spam Detection.* In *Handbook of Natural Language Processing and Machine Translation*, Springer.
* **Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A.** (2011). *Contributions to the study of SMS spam filtering: New collection and results.* Proceedings of the 11th ACM Symposium on Document Engineering.
* **Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E.** (1998). *A Bayesian approach to filtering junk e-mail.* AAAI Workshop on Learning for Text Categorization.
* **Russell, S., & Norvig, P.** (2020). *Artificial Intelligence: A Modern Approach (4th Edition).* Pearson Education Limited.
* **Zhang, L.**, *The Optimality of Naive Bayes*, *AAAI Conference on Artificial Intelligence*, 2004.