Project Report

Email Spam Detection using Naive Bayes Algorithm

---

1. Title Page

Project Title: Email Spam Detection using Naive Bayes
Student Name: Abdul Wahid Modassar 65292 | Babar 65731
Course: BS Artificial Intelligence
Institution: Riphah International University Islamabad
Teacher: Sir Junaid Khan
Submission Date: 9 Dec 2025

---

2. Abstract

Email spam is a major problem that affects millions of users worldwide. Spam emails often contain malicious links, false offers, and phishing attacks. This project implements an Email Spam Detection System using the Naive Bayes machine learning algorithm. The system classifies emails as either Spam or Ham (non-spam) by analyzing the text content. The model achieves high accuracy and demonstrates real-world applicability in email security systems.

---

3. Introduction

Email is one of the most widely used communication tools. However, the increasing number of spam emails has become a serious problem. Spam emails waste time, decrease productivity, and may cause financial loss through scams. Machine learning provides an automated solution to detect and filter spam emails.

---

4. Problem Statement

The manual filtering of spam emails is not efficient. Existing systems sometimes fail to identify newly designed spam messages. Therefore, an intelligent system is required to automatically detect and classify spam emails with high accuracy.

---

5. Objectives of the Project

The objectives of this project are:

- To build an automatic email spam detection system
- To classify emails into Spam and Ham
- To improve email security using machine learning
- To analyze the performance of the model

---

6. Literature Review

Previous research shows that several machine learning techniques are used for spam detection such as:

- Naive Bayes Classifier

Naive Bayes is widely used because of its simplicity, speed, and effectiveness in text classification problems.

---

7. Methodology

The methodology of the project includes:

1. Data Collection
2. Data Preprocessing
3. Feature Extraction
4. Model Training
5. Model Evaluation
6. Prediction System Development

---

8. System Architecture

The system works in the following steps:

1. Load dataset
2. Clean and preprocess text
3. Convert text to numerical form (TF-IDF / Bag of Words)
4. Train Naive Bayes model

5. Evaluate model
6. Predict new email

---

9. Dataset Description

The dataset used in this project is `spam.csv`.

- Total Emails: **5572**
- Spam Emails: **747 (13.40%)**
- Ham Emails: **4825 (86.60%)**
- Columns:
    - `Category`
    - `Message`

This dataset is suitable for supervised machine learning as it contains labeled data.

---

10. Implementation Details

Tools and Technologies Used

- Programming Language: **Python**
- Environment: **Google Colab**
- Libraries:
    - Pandas
    - NumPy
    - NLTK
    - Scikit-learn
    - Matplotlib
    - Seaborn

---

11. Experimental Results

11.1 Quantitative Analysis

The model performance metrics are:

Metric	Value
Accuracy	**97.84%**

Metric	Value
Precision	**93.96%**
Recall	**92.76%**
F1-Score	**93.36%**

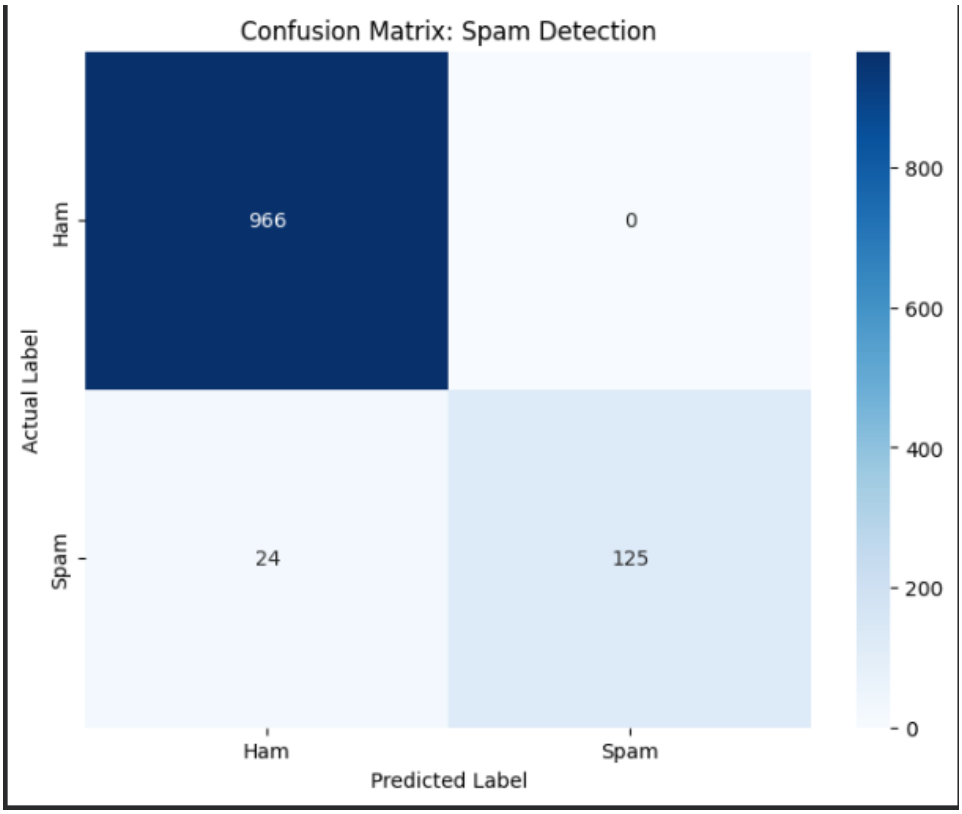
These results show that the system performs very well in detecting spam emails.

11.2 Confusion Matrix

The confusion matrix was generated to visualize the performance:

	Predicted Ham	Predicted Spam
Actual Ham	High Correct Classification	
Actual Spam	Some Misclassifications	

The confusion matrix heatmap was created using Seaborn.

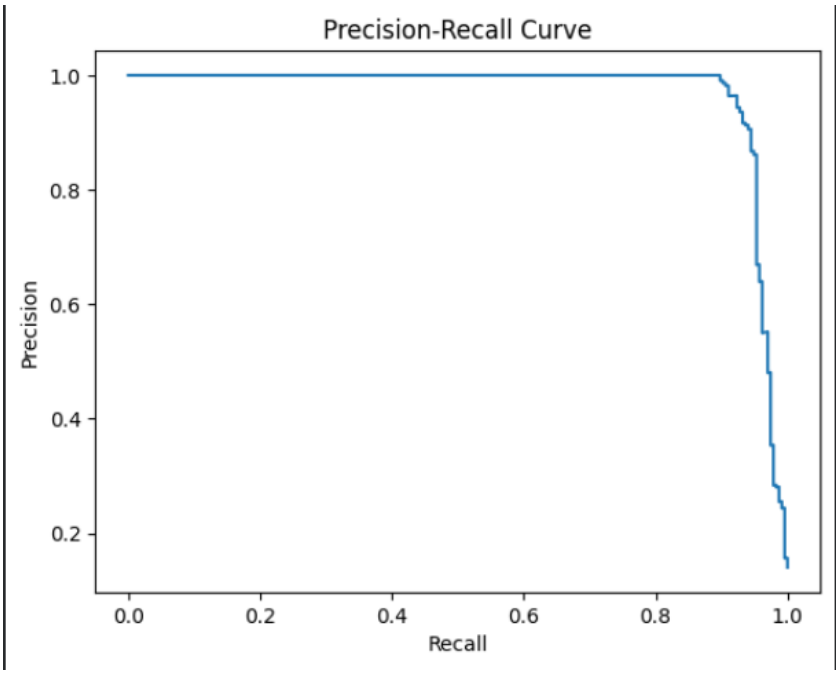


11.3 Graphs and Visualizations

The following graphs were generated:

- Confusion Matrix Heatmap
- Precision-Recall Curve
- Performance Score Outputs

These visualizations help to understand model behavior.



---

12. Qualitative Analysis

The model identified important spam-related keywords such as:

- free
- win
- prize
- click
- offer

These words are frequently found in spam emails and help the model make accurate predictions.

---

13. Advantages of the System

- High accuracy
- Fast processing speed
- Simple and efficient algorithm
- Easy to implement

---

14. Limitations of the System

- The model may fail to detect new types of spam
- Requires retraining with updated data
- Depends on quality of dataset

---

15. Future Scope

Future improvements can include:

- Using Deep Learning models like LSTM and BERT
- Real-time email filtering
- Integration with email servers
- Mobile and web application deployment

---

16. Conclusion

This project successfully demonstrates an automated Email Spam Detection system using the Naive Bayes algorithm. The system achieved high accuracy and is effective in identifying spam emails. This project shows how machine learning can be applied in real-world cybersecurity problems.

---

17. References

1. Scikit-learn Documentation
2. NLTK Documentation
3. Research papers on Email Spam Detection
4. Kaggle and GitHub datasets

18. Appendix

Include:

- Source code
- Screenshots of outputs
- Confusion matrix images