Spam Email Filtering using Machine Learning

Implementation with Multinomial Naive Bayes & Random Under-Sampling

Presented By:

Lakshya Sharma (AD24B1038)

Bandi Navadeep (AD24B1014)

Table of Contents

01. Introduction

Understanding the problem of email spam and security implications.

02. Literature Survey

Review of traditional vs. modern machine learning approaches.

03. Methodology

Data preprocessing, balancing, and Multinomial Naive Bayes model.

04. Results & Analysis

Performance metrics, confusion matrix, and deployment.

What is Spam Filtering?

- Automated classification of emails into "Ham" (legitimate) or "Spam".
- **Critical Security:** Mitigates risks like phishing, fraud, and malware distribution.
- **Efficiency:** Reduces server load, storage costs, and network congestion.
- **User Protection:** Prevents data theft and identity compromise.
- Utilizes **NLP techniques** to analyze text patterns effectively.

Literature Survey

Traditional Methods

Rule-based filters and keyword matching are rigid and easily bypassed by attackers.

Statistical Models

Naive Bayes serves as a robust baseline due to its effectiveness with text data.

Data Handling

Techniques like TF-IDF and resampling are vital for handling imbalanced datasets.

Methodology Pipeline

1

Preprocessing

Regex cleaning, lowercase conversion, and structure extraction.

2

Balancing

Random Under-Sampling (RUS) to achieve 1:1 Spam/Ham ratio.

3

Vectorization

TF-IDF extraction with top 4000 features and bigrams.

4

Modeling

Training Multinomial Naive Bayes (MNB) with alpha smoothing.

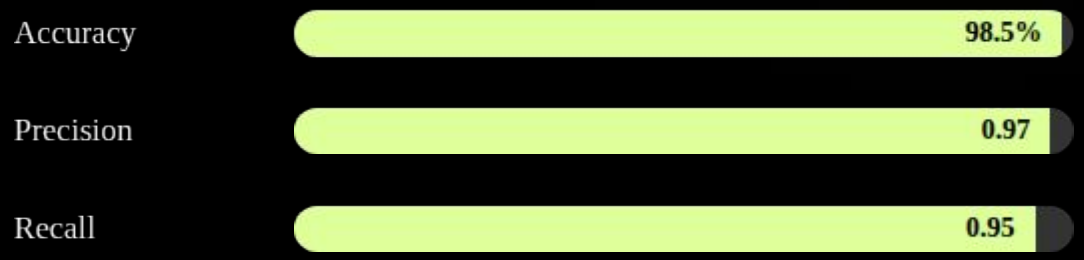
Model Results

Confusion Matrix

Classification counts on the test set:

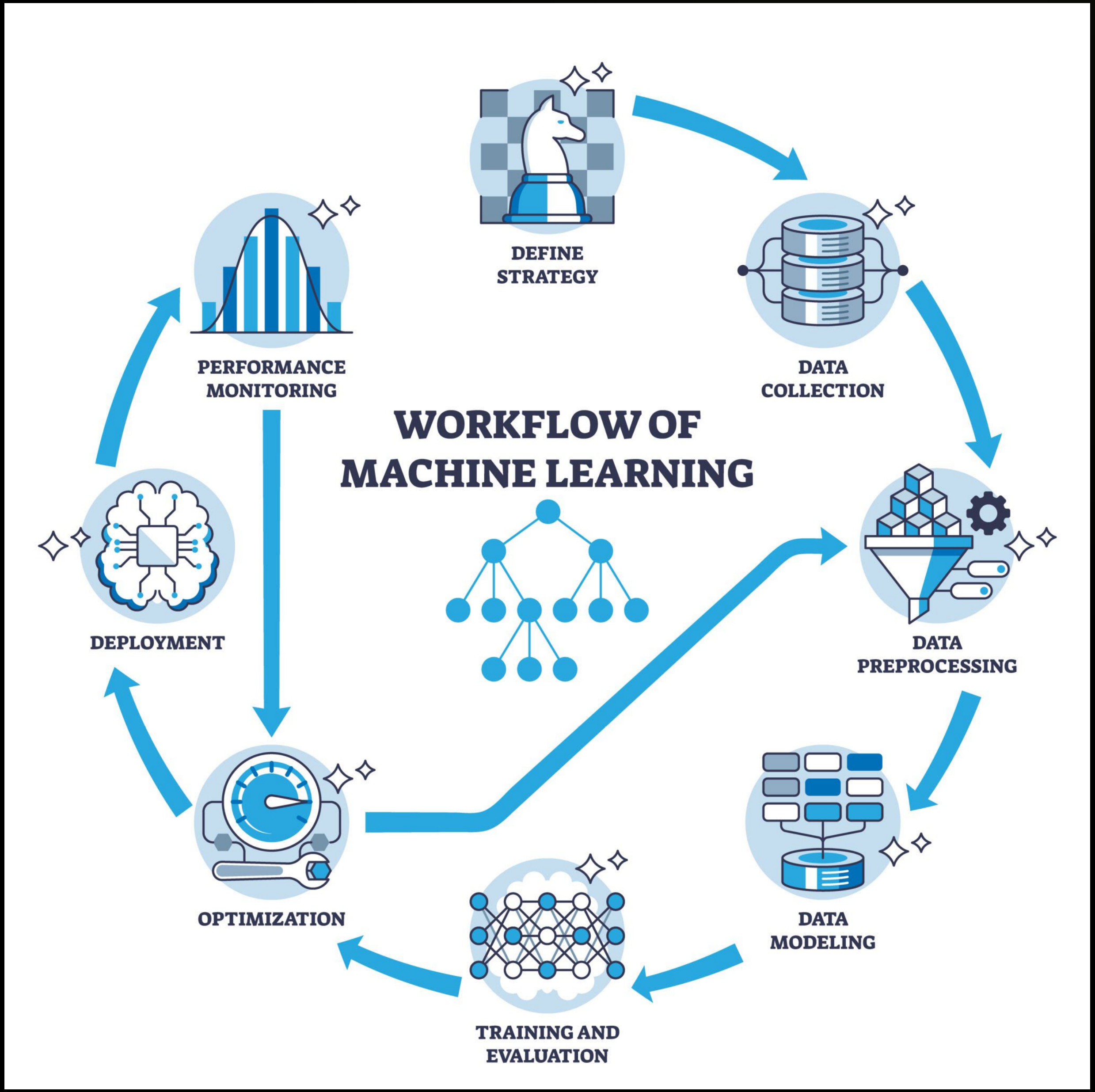
Actual \ Predicted	Ham (0)	Spam (1)
Ham (0)	True Negative	False Positive
Spam (1)	False Negative	True Positive

Key Metrics



Analysis & Deployment

- **Why MNB?** Highly efficient for high-dimensional text data compared to Decision Trees.
- **Impact of RUS:** Under-sampling prevented model bias toward the majority "Ham" class.
- **Low False Positives:** High precision prioritizes user trust by not flagging safe emails.
- **Streamlit Deployment:** Provides real-time interface with confidence scoring (Probabilities).



Conclusion

- Successfully implemented a robust spam filter using **Multinomial Naive Bayes**.
- Deployed a user-friendly web app using **Streamlit**.
- Achieved **98.5% accuracy** with minimal false positives.
- **Future Work:** Integrate Deep Learning (LSTM/BERT) and advanced visualization.