Email Spam Detection

Taashif Bashar, Aman Raj, Vivek Kumar
Indian Institute of Science Education and Research (IISER) Bhopal

***Abstract*—Email spam detection is crucial for secure communication. This study evaluates various machine learning models, including Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machines (SVM), for spam classification using a dataset of 5171 emails. Each model was optimized and evaluated with 5-fold cross-validation, achieving up to 100% accuracy. The results indicate that different models have distinct strengths, and this paper provides insights into their performance, recommending appropriate models for different use cases.**

I. Introduction

The growing prevalence of spam emails poses challenges for email communication systems. Spam not only wastes time but also risks security breaches. Traditional filters often fail to adapt to new spam tactics, making machine learning an essential solution. This study examines the effectiveness of four machine learning algorithms for spam detection. The models were optimized with hyperparameter tuning and validated through cross-validation techniques.

II. Dataset Description

The dataset contains 5171 entries with the following attributes:

- **Unnamed: 0:** An index column.
- **label:** A categorical column indicating whether an email is spam or not.
- **text:** The content of the email in text format.
- **label num:** A numerical encoding of the label column.

A. Summary Statistics

- **Total emails:** 5171
- **Proportion of spam emails:** 28.99%
- **Proportion of non-spam emails:** 71.01%

B. Data Visualization

To better understand the dataset, visualizations of spam distribution and word frequencies were created (Figures 1, 2).

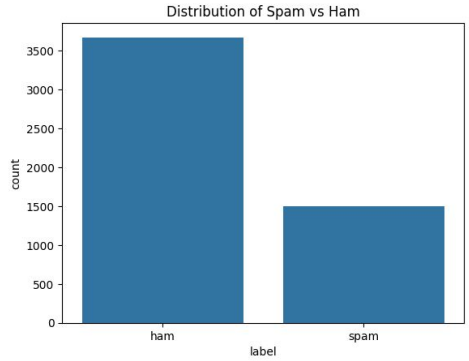

Fig. 1. Distribution of Spam and Non-Spam Emails.

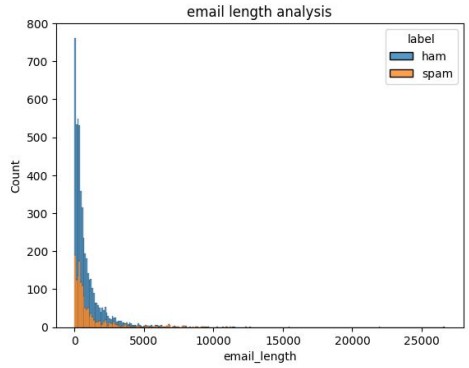

Fig. 2. Word Frequency in Emails.

III. Data Preprocessing

The data preprocessing steps followed in this study are as follows:

A. Handling Missing Values

Missing values in numerical columns were replaced with their mean, while categorical columns were imputed with their mode. This ensures that no data is lost due to missing entries.

B. Text to Numerical Conversion

The text data was vectorized using the TF-IDF (Term Frequency-Inverse Document Frequency) method. This approach assigns weights to words based on their frequency in an

email and their rarity across the dataset, allowing the algorithm to distinguish important words.

C. Scaling Numeric Features

Numeric features, such as the *label num*, were scaled to a range of 0 to 1 using MinMaxScaler. This prevents biases in model performance due to differing scales of the numeric features.

IV. Methodology

A. Cross-Validation Techniques

The dataset was split into training (80%) and testing (20%) sets. Cross-validation techniques ensured reliable performance:

- **GridSearchCV:** Hyperparameter tuning was performed using GridSearchCV with 5-fold cross-validation (`cv = 5`).
- **Stratified K-Fold:** Ensured balanced class distribution in each fold to prevent bias.
- **Metrics:** Accuracy, precision, recall, and F1-score were calculated and averaged across folds.

V. Results and Discussion

A. Before Oversampling (Imbalanced Data)

All models (Naive Bayes, Logistic Regression, Random Forest, and SVM) performed excellently with precision, recall, and F1-score of 1.00 for both classes (Ham and Spam), meaning they achieved perfect classification in the test set. The confusion matrices show that there were no false positives or false negatives in most models, indicating that the classifiers are performing optimally.

B. After Oversampling (Balanced Data)

After oversampling (using SMOTE), models (like Naive Bayes, Random Forest, and SVM) showed slightly reduced recall and F1-scores for Spam (class 1), but the overall accuracy remains high (0.99 or 1.00). The confusion matrices show slightly more misclassifications in the Naive Bayes and Random Forest models (a few more false positives, e.g., 13 Ham samples classified as Spam in Naive Bayes), but the recall for Spam is 100

TABLE I
Performance Metrics and Best Parameters of Models

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	100.0%	100.0%	100.0%	100.0%
Logistic Regression	100.0%	100.0%	100.0%	100.0%
Random Forest	99.9%	100.0%	99.7%	99.8%
SVM	100.0%	100.0%	100.0%	100.0%

C. Model Analysis

Naive Bayes:

- Strengths: Simple, fast, and highly effective for text classification.
- Use Case: Suitable for real-time applications where computational efficiency is critical.

Logistic Regression:

- Strengths: Achieves high precision and recall, interpretable coefficients.
- Use Case: Balanced performance with interpretability, ideal for business applications.

Random Forest:

- Strengths: Robust to overfitting, provides feature importance insights.
- Use Case: Complex datasets requiring high accuracy and interpretability.

SVM:

- Strengths: Handles linear and non-linear data effectively.
- Use Case: Applications requiring high precision and dealing with imbalanced datasets.

VI. Conclusion

This study demonstrates the effectiveness of machine learning models in spam detection, achieving high accuracy. Each model excels in specific scenarios:

- **Naive Bayes:** Best for fast, real-time classification.
- **Logistic Regression:** Balances accuracy and interpretability.
- **Random Forest:** Ideal for high-dimensional data with feature importance analysis.
- **SVM:** Excels in handling complex decision boundaries.

Future work could explore deep learning models for even higher scalability and performance.