EMAIL SPAM PREDICTION

Raja Moughrabi
MCIT Data Science Program
rajamoughrabi77@gmail.com

ABSTRACT

That is my abstract and it provides some information about the project. Email spam remains a persistent problem that affects productivity, security, and user experience. This project addresses the task of binary email spam classification using machine learning techniques. Experiments are conducted on the UCI Spambase dataset, which contains 4,601 instances described by 57 numerical features extracted from email content. Three models are evaluated: Gaussian Naive Bayes as a baseline, a fully connected neural network implemented in PyTorch, and a Random Forest classifier. Model performance is assessed using accuracy, precision, recall, F1-score, and confusion matrices. The results show that the Random Forest model achieves the best overall performance with an accuracy of 94.6% and an F1-score of 0.93, while the PyTorch model also demonstrates strong performance. These findings highlight the effectiveness of ensemble and neural network approaches over probabilistic baselines for spam detection.

1 INTRODUCTION

Electronic mail remains one of the most widely used communication tools for both personal and professional purposes. However, the widespread use of email has also led to a significant increase in unsolicited and malicious messages, commonly known as spam. Spam emails can contain advertisements, phishing attempts, or harmful content, leading to productivity loss and potential security risks for users and organizations.

Machine learning has become a standard approach for spam detection due to its ability to automatically learn patterns from historical data. Traditional rule-based filtering systems are often insufficient, as spam content continuously evolves to bypass fixed rules. In contrast, data-driven models can adapt to new spam characteristics by learning from labeled examples.

This project focuses on evaluating and comparing multiple machine learning approaches for the task of email spam classification. By analyzing both probabilistic and discriminative models, the goal is to understand how model complexity and feature interactions affect classification performance.

1.1 PROBLEM STATEMENT

The problem addressed in this work is a binary classification task. Given a set of numerical features extracted from an email message, the objective is to predict whether the email is spam or not spam. Formally, the task consists of learning a function that maps a feature vector to one of two class labels: spam (1) or non-spam (0).

The performance of the proposed models is evaluated using standard classification metrics, with particular attention to precision and recall. These metrics are critical in spam detection, as false positives may block legitimate emails, while false negatives allow spam messages to reach users.

2 BACKGROUND AND RELATED WORKS

Email spam detection has been widely studied in the machine learning literature, as it represents a practical and well-defined binary classification problem. Early approaches to spam filtering relied on manually crafted rules and keyword matching. While effective in limited scenarios, such systems lack adaptability and often fail when spam content evolves.

Probabilistic methods, particularly Naive Bayes classifiers, have been among the earliest and most popular machine learning techniques applied to spam detection. Naive Bayes models assume conditional independence between features given the class label, which allows for efficient training and inference. Despite this simplifying assumption, Naive Bayes classifiers have demonstrated strong baseline performance in many text and email classification tasks.

More recent work has explored the use of neural networks for spam classification. Fully connected neural networks can model nonlinear relationships between input features, enabling them to capture more complex patterns in the data. When combined with proper feature normalization and sufficient training data, neural networks often outperform simpler probabilistic models.

Ensemble learning methods, such as Random Forests, have also been successfully applied to spam detection problems. Random Forests combine the predictions of multiple decision trees to improve generalization and reduce overfitting. By aggregating diverse decision boundaries, ensemble methods often achieve higher robustness and accuracy compared to individual classifiers.

This project builds upon these established approaches by experimentally comparing a probabilistic baseline, a neural network model, and an ensemble-based method on a common dataset, providing insights into their relative strengths and weaknesses for email spam detection.

Earlier spam detection systems relied heavily on probabilistic approaches such as Naive Bayes due to their simplicity, interpretability, and low computational cost. These models were particularly effective when email content could be represented using frequency-based features. Despite their strong baseline performance, the conditional independence assumption often limits their ability to capture complex dependencies among features.

More recent research has demonstrated the effectiveness of neural network-based approaches for classification tasks involving structured and semi-structured data. Fully connected neural networks are capable of modeling non-linear feature interactions, which allows them to outperform simpler probabilistic models when sufficient training data is available.

Ensemble learning methods, including Random Forests, have gained popularity due to their robustness and strong generalization performance. By aggregating multiple decision trees trained on random subsets of data and features, Random Forests reduce variance and mitigate overfitting. Such properties make ensemble methods particularly suitable for tabular datasets such as Spambase, where feature relationships are complex but well-defined.

3 METHOD

This section describes the dataset used in this study, the preprocessing steps applied to the data, and the machine learning models evaluated for email spam classification.

3.1 DATASET

The experiments are conducted using the Spambase dataset from the UCI Machine Learning Repository Hopkins et al. (1999). The dataset consists of 4,601 email instances, each represented by 57 numerical features extracted from the content of email messages. These features capture characteristics such as word frequency, character frequency, and statistics related to capitalization. Each instance is labeled as either spam (1) or non-spam (0), making the task a binary classification problem.

The dataset is commonly used as a benchmark for evaluating spam detection algorithms due to its moderate size and well-defined feature set.

3.2 DATA PREPROCESSING

Before training the models, the dataset is divided into training and testing subsets using an 80/20 split. Stratified sampling is applied to preserve the original class distribution in both subsets.

For the neural network model, feature scaling is performed using standardization to ensure that all input features have zero mean and unit variance. This step is critical for stable and efficient neural

network training. The Gaussian Naive Bayes and Random Forest models are trained on the original feature values, as they are less sensitive to feature scaling.

3.3 GAUSSIAN NAIVE BAYES

Gaussian Naive Bayes is used as a baseline probabilistic model for spam classification. This model assumes that each feature follows a Gaussian distribution and that features are conditionally independent given the class label. Despite its simplifying assumptions, Gaussian Naive Bayes is computationally efficient and often provides strong baseline performance in text and email classification tasks.

3.4 FULLY CONNECTED NEURAL NETWORK - PYTORCH

A fully connected neural network is implemented using PyTorch to model nonlinear relationships between input features. The network consists of multiple dense layers with ReLU activation functions and a sigmoid output layer for binary classification. The model is trained using binary cross-entropy loss and the Adam optimization algorithm. Feature standardization is applied prior to training to improve convergence and performance.

3.5 RANDOM FOREST

Random Forest is an ensemble learning method that constructs multiple decision trees during training and aggregates their predictions through majority voting. By combining multiple trees trained on random subsets of features and samples, Random Forest reduces overfitting and improves generalization. This model is well-suited for tabular data and serves as a strong non-linear classifier for spam detection.

4 EXPERIMENTS AND METHODOLOGY

This section describes the experimental setup, evaluation metrics, and quantitative results obtained from the machine learning models evaluated in this study. All models are compared under identical conditions to ensure a fair and consistent evaluation.

4.1 EXPERIMENTAL SETUP

The dataset is divided into training and testing sets using an 80/20 stratified split to preserve the original class distribution. All models are trained on the same training data and evaluated on the same test set.

Gaussian Naive Bayes and Random Forest classifiers are implemented using the `scikit-learn` library, while the fully connected neural network is implemented using PyTorch. Model hyperparameters are selected based on commonly used default settings rather than extensive tuning, as the primary objective is comparative performance analysis.

4.2 EVALUATION METRICS

Model performance is evaluated using accuracy, precision, recall, and F1-score. Accuracy measures the overall correctness of predictions, while precision and recall provide insight into false positive and false negative behavior. The F1-score is used as a balanced metric that combines precision and recall, which is particularly important for spam detection tasks.

4.3 QUANTITATIVE RESULTS

Table 1 summarizes the performance of all evaluated models on the test dataset.

Table 1: Performance comparison of evaluated models

Model	Accuracy	Precision	Recall	F1-score
Gaussian Naive Bayes	0.834	0.718	0.953	0.819
Fully Connected Neural Network	0.937	0.937	0.901	0.919
Random Forest	0.946	0.951	0.909	0.930

4.4 Confusion Matrices

Figure 1 presents the confusion matrices for the evaluated models, providing a visual summary of correct and incorrect classifications for each approach.

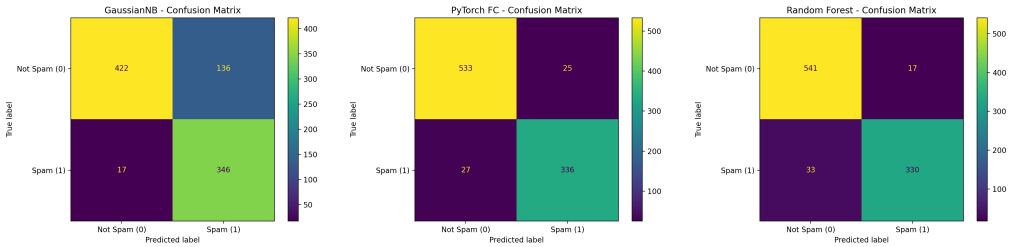

Figure 1: Confusion matrices for Gaussian Naive Bayes, Fully Connected Neural Network, and Random Forest.

4.5 ROC Curve

Figure 2 shows the Receiver Operating Characteristic (ROC) curve for the Random Forest classifier, illustrating the trade-off between true positive rate and false positive rate across different classification thresholds.

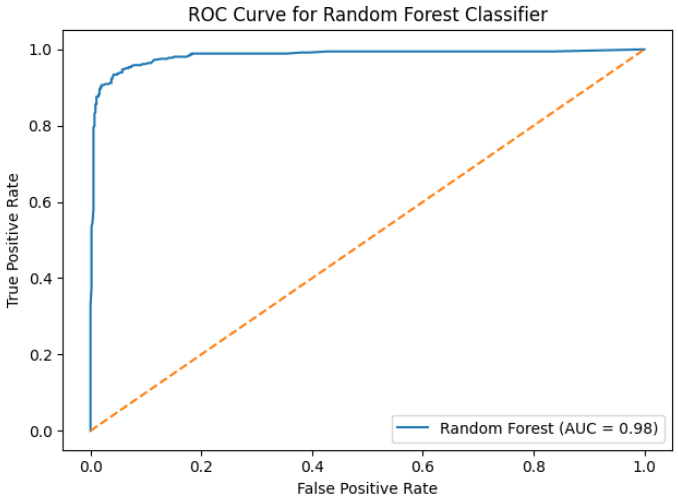

Figure 2: ROC curve for the Random Forest classifier.

5 Results Analysis

This section provides a detailed analysis of the performance differences among the evaluated models. While overall accuracy offers a general measure of correctness, spam detection requires careful consideration of precision and recall, as different error types carry different practical consequences. The observed behaviors are examined using confusion matrices and ROC analysis.

Gaussian Naive Bayes serves as a reasonable baseline model. Its relatively high recall indicates that it successfully identifies most spam emails. However, its lower precision reveals that a larger proportion of legitimate emails are incorrectly classified as spam. In real-world deployment, this behavior may negatively impact user experience by increasing false positives.

As shown in Figure 1 the neural network and Random Forest models reduce false positives compared to Gaussian Naive Bayes.

The fully connected neural network implemented in PyTorch demonstrates a substantial performance improvement over Gaussian Naive Bayes. By modeling non-linear relationships among the 57 input features, the neural network achieves higher accuracy and F1-score, resulting in a more balanced trade-off between precision and recall.

Random Forest achieves the strongest overall performance across all evaluation metrics. By aggregating multiple decision trees, the model reduces variance and improves generalization. The higher F1-score indicates that Random Forest provides the most balanced classification behavior, minimizing both false positives and false negatives.

As shown in Figure 2, further confirms the superior discriminative ability of the Random Forest classifier.

6 Error Analysis

In spam detection, different types of classification errors carry different practical consequences. False positives occur when legitimate emails are incorrectly classified as spam, potentially causing users to miss important information. False negatives occur when spam emails are classified as legitimate, allowing unwanted or potentially harmful messages to reach users.

The confusion matrices presented in Figure 1 illustrate these trade-offs across the evaluated models. Gaussian Naive Bayes produces a relatively high number of false positives, which is consistent with its lower precision. While this behavior may be acceptable in scenarios where missing spam is extremely costly, it is less suitable when user experience and trust are prioritized.

The fully connected neural network significantly reduces false positives while maintaining strong recall, indicating improved discrimination between spam and non-spam emails. Among all evaluated models, Random Forest achieves the most balanced error profile, minimizing both false positives and false negatives. This balanced behavior explains its superior F1-score and overall classification performance.

7 Limitations

This study has several limitations. First, the Spambase dataset consists of pre-extracted numerical features rather than raw email text. While this makes modeling efficient, it limits direct comparison with modern NLP-based spam filters that operate on raw text using TF-IDF, embeddings, or transformer architectures.

Second, the dataset represents a fixed distribution and may not reflect current spam strategies. In real-world deployment, spam content evolves over time, and models may require periodic retraining to remain effective.

Finally, model hyperparameters were selected using common default settings rather than extensive tuning. Although this is sufficient for comparative evaluation, further optimization could yield additional performance gains.

8 Conclusion and Future Work

This project evaluated multiple machine learning approaches for email spam classification using the Spambase dataset. Gaussian Naive Bayes served as a strong baseline due to its simplicity and high recall, while the fully connected neural network demonstrated the benefits of modeling non-

linear feature relationships. Among all evaluated models, Random Forest achieved the best overall performance, with the highest accuracy and F1-score.

The results highlight the importance of model selection in spam detection systems, where different error types carry different practical consequences. Ensemble-based methods, such as Random Forests, provide a robust and effective solution for tabular spam classification tasks.

8.1 Future Work

Future research could explore the use of raw email text combined with modern natural language processing techniques such as TF-IDF, word embeddings, or transformer-based models. Additional directions include cost-sensitive learning, cross-validation, and evaluation under dataset shift to better reflect real-world deployment scenarios

References

Mark Hopkins, Erik Reeber, George Forman, and Jaap Suermondt. Spambase data set. `https://archive.ics.uci.edu/ml/datasets/Spambase`, 1999. UCI Machine Learning Repository.