Spam Email Detection: Naive Bayes vs. Logistic Regression vs. Neural Networks

COMS 4740 Project Final Report

Team Members: Savannah Franklin, Logan Gosch, Ryan Riebesehl

Abstract

Our primary task was to compare 3 machine learning models: Naive Bayes, Logistic Regression, and Neural Networks in spam detection. Spam email detection is an essential machine learning application as it helps limit unwanted messages and protect users from potential security breaches. Utilizing the UCI Spambase dataset, containing an extensive list of 4,601 emails with 57 numerical features all marked as spam or not spam accordingly, we have run this through each model to test its accuracy.

Having had exposure to data science and the effects of both poorly and well-executed spam detection, we've explored further the extent to which these models work. Utilizing information provided by the dataset, we've determined that Neural Networks most accurately interpret the provided information and developed our model on it. Our results demonstrate the importance of pre-processing, hyperparameter tuning, and evaluation techniques in machine learning.

1 Introduction

Spam emails continue to be an issue to this very day in the world. They offer constant nuisances within our mailboxes by filling them with not only advertisements and other unwanted messages, but also phishing and malware attempts. These prove to be annoying inconveniences and large gaps in security. While there have been other methods used to assist in filtering out these unnecessary messages, none have proved to be as effective as those using machine learning. With ever-evolving structures of these emails being made to bypass some of these other solutions, machine learning allows for our methods to adapt accordingly to such changes.

Where we come in is in our project's main focus: How supervised machine learning handles spam detection. Specifically, we compare three different learning models: Naïve Bayes, Logistic Regression, and Neural Networks. Each learning model has its advantages and disadvantages. Naïve Bayes is easy to implement and computationally efficient. Logistic Regression offers a good balance between interpretability and performance. Neural Networks can learn and identify more intricate, non-linear patterns that other models may not detect.

To evaluate and train these models, we use the UCI Spambase dataset (available at: https://archive.ics.uci.edu/dataset/94/spambase). This dataset is a fairly well-known set of data commonly used for spam detection. It includes 4,601 labeled email samples and 57 numerical features. Some of the features we used for our project include the frequencies or sequences of words, characters, punctuation, capitalization, and more. These assist in classifying what is spam or not.

After pre-processing our data through normalization, scaling, and stratified sampling, we were able to train and validate each of our 3 models for the most precise and accurate one. This led us to find that Neural Networks ultimately outperform Naïve Bayes and Logistic Regression learning models. This information directly aligns with what we learned in class. With this confirmed, we could move forward with our implementation of the model utilizing Neural Networks.

This project and the information we gathered from it has allowed us to understand the importance of these models and the steps that lead to increased performance of such models. Still today, this is a leading solution in handling spam emails and messages.

2 Related Work

Spam detection has long been a prominent application of machine learning, driven by the need to manage the increasing volume of unsolicited emails in modern communication systems. Especially in recent years, many different algorithms have been employed, ranging from probabilistic models to deep learning architectures, each offering different strengths and weaknesses in terms of accuracy, interpretability, and scalability.

One of the earliest and most influential approaches to spam classification is the Naive Bayes classifier. This model operates under the assumption of feature independence, allowing it to compute the posterior probability of an email being spam based on the individual occurrences of words or tokens. Despite its simplicity, Naive Bayes has been effective in real-world applications, especially during the early days of spam filtering. [2] were among the first to explore the application of Bayesian learning in junk email filtering, demonstrating that probabilistic methods could significantly reduce false positives in automated filtering systems. However, the model's assumption that all features (word occurrences/associations) are conditionally independent given the class label is often violated in natural language, which can limit its accuracy in detecting more sophisticated or obfuscated spam.

Logistic Regression has also become a widely used model in spam detection tasks, particularly due to its ease of implementation and probabilistic output. Unlike Naive Bayes, Logistic Regression does not assume independence between features, enabling it to model interactions and correlations among words more effectively.[3] Studies have shown that Logistic Regression can achieve high classification accuracy when properly regularized and trained on a well-preprocessed dataset. Its linear nature, however, imposes limitations when dealing with nonlinear decision boundaries—an issue particularly relevant in distinguishing between cleverly disguised spam and legitimate messages. Nevertheless, its interpretability and lower computational requirements make it an appealing choice for it's easily scalability alone.

With the emergence of deep learning, more complex models, such as Neural Networks, have been explored for spam detection. These models are capable of learning hierarchical representations of email content, which allows them to detect subtle patterns and semantic structures that are typically missed by traditional classifiers. For instance, RNNs have been particularly useful in modeling the sequential nature of text, capturing context and dependencies across word sequences. CNNs, on the other hand, can extract local features that are useful for identifying commonly used spam phrases or token structures. Research has consistently shown that deep learning models outperform simpler algorithms on large datasets, including the UCI Spambase dataset and others collected from real-world email systems. However, their increased accuracy often comes at the cost of higher computational requirements, longer training times, and reduced model interpretability. These constraints pose challenges for real-time applications where latency and resource usage are critical considerations.

More recent studies have begun to explore hybrid and ensemble approaches that combine the strengths of multiple models. Ensemble techniques such as bagging, boosting, and stacking have shown promise in improving spam detection performance by aggregating the outputs of weak learners to form a more robust final prediction. For example, Random Forests and Gradient Boosting Machines have been employed with success in spam classification, offering improved generalization and better handling of imbalanced data. Moreover, research has investigated the use of feature engineering techniques such as TF-IDF vectorization, word embeddings, and character n-grams to enhance the input representations used by both classical and deep models.

Our work builds upon this foundation by conducting a comparative analysis of Naive Bayes, Logistic Regression, and Neural Networks for spam detection using the UCI Spambase dataset. By examining both traditional modeling techniques and more modern architectures researchers and scholars have used in the past, we can better the development of our own. Their insights helped us to dive deep into the different approaches for building a spam filtering model.

3 Methods

3.1 Dataset Collection and Description

This study utilizes the well-known Spambase dataset, which is publicly available through the UCI Machine Learning Repository (https://archive.ics.uci.edu/dataset/94/spambase). The dataset was developed by researchers at Hewlett-Packard Labs and is widely used for benchmarking classification algorithms in spam detection tasks. It consists of 4,601 email samples, each annotated with 57 continuous real-valued features and a binary target variable. The target variable indicates whether an email is considered spam (1) or not spam (0), based on manually labeled ground truth.

The features in this dataset were meticulously engineered from the raw text of emails. These include term frequency counts for specific keywords (such as ”free”, ”money”, and ”business”), character-level statistics (e.g., frequency of symbols like ”$” or ”!”), and a series of measurements pertaining to the use of capital letters (such as the average length of capital letter sequences and the total number of capitalized characters). These variables were selected because spam emails often share lexical and stylistic characteristics that differentiate them from legitimate (ham) emails.

The high dimensionality and rich statistical representation of the data make this dataset particularly suitable for evaluating the performance of diverse machine learning models. It presents challenges such as class imbalance and feature correlation, offering a realistic setting for testing classification algorithms in the context of text-based spam filtering.

3.2 Data Preprocessing

Preprocessing is a critical phase in any machine learning pipeline, particularly for datasets derived from natural language. Given the numerical format of the Spambase dataset, we performed several key preprocessing steps to prepare the data for training and evaluation.

3.2.1 Feature and Label Separation

The first step in preprocessing involved separating the predictors (features) from the target labels. The input matrix X consists of the first 57 columns representing various word frequencies and character statistics. The target vector y, extracted from the 58th column, contains binary values indicating spam (1) or non-spam (0). This separation ensures a clean division between inputs and outputs for model training.

3.2.2 Train-Test Splitting

To facilitate an unbiased evaluation of model performance on unseen data, we split the dataset into training and testing sets. We employed a 70-30 split using scikit-learn's train test split function, ensuring that the splitting process was stratified by the target variable. Stratification preserves the class distribution between the training and test sets, preventing skewed evaluation metrics due to disproportionate class representation, which is especially important in tasks like spam detection that may exhibit class imbalance.

3.2.3 Feature Scaling

Feature scaling is essential for models that rely on gradient descent or distance-based calculations. We applied z-score normalization using scikit-learn's StandardScaler to transform each feature so that it has zero mean and unit variance. This improves numerical stability, enhances convergence speed, and often boosts predictive performance for models like logistic regression and neural networks. However, for Multinomial Naïve Bayes, we retained the original, unscaled feature values. This is because the model assumes count-like features and performs best when features retain their relative frequency-based interpretations. Standardization would violate this assumption and degrade performance.

3.3 Model Selection and Justification

We selected three machine learning models representing a range of algorithmic complexity and learning models: Multinomial Naïve Bayes, Logistic Regression, and a Multi-Layer Perceptron (Neural Network). Each model was chosen to examine the trade-offs between simplicity, interpretability, and predictive power.

3.3.1 Multinomial Naïve Bayes

The Multinomial Naïve Bayes model assumes conditional independence among features given the class label and treats input features as counts or frequencies. It is especially effective for high-dimensional sparse data, a common characteristic in text classification problems. Given the nature of the Spambase features—word frequencies and character counts—this model is well-suited to the task.

Despite its simplifying assumptions, Naïve Bayes often achieves competitive accuracy in spam detection and other text classification domains. Its fast training time and low computational cost make it a practical choice for real-time applications. Additionally, its probabilistic framework provides interpretable results, such as likelihood estimates for class membership.

3.3.2 Logistic Regression

Logistic Regression models the probability of a binary outcome using the logistic sigmoid function. It estimates coefficients for each input feature to model the log-odds of the positive class, making it a valuable linear baseline. We selected this model for its combination of interpretability, simplicity, and effectiveness across a wide range of classification problems.

For this study, we enabled a maximum iteration count of 1000 to ensure convergence during optimization, especially given the feature set's dimensionality. The model was trained using scikit-learn's default L2 regularization, which helps prevent overfitting by penalizing large coefficient values. Although further regularization tuning could be performed, initial trials showed no substantial performance gain, so defaults were retained for reproducibility.

3.3.3 Neural Network (Multi-Layer Perceptron)

To explore the potential benefits of modeling complex, non-linear feature interactions, we implemented a feedforward neural network using scikit-learn's MLPClassifier. Neural networks are capable of learning intricate patterns in data through multiple layers of non-linear transformations, making them highly flexible classifiers.

We performed grid search with 3-fold cross-validation to identify optimal hyperparameters. The configurations tested varied the architecture of the hidden layers (including 100 and 100-50 neuron setups) and the strength of L2 regularization via the `alpha` parameter. The goal was to balance model complexity and generalization ability. The final model was selected based on its average F1 score across validation folds, and then retrained on the entire training set before being evaluated on the test set.

3.4 Evaluation Metrics

Model performance was evaluated using a combination of quantitative metrics and visual diagnostics. This multi-faceted approach provides a thorough understanding of each model's strengths and weaknesses.

- **Accuracy**: The proportion of correctly predicted labels over the total number of predictions. While intuitive, it can be misleading in imbalanced datasets.

- **Precision**: The fraction of true positive predictions among all instances predicted as positive. It is especially relevant in spam detection, where false positives (mislabeling legitimate emails as spam) can be costly.

- **Recall**: The fraction of true positives among all actual positive instances. High recall ensures that most spam emails are correctly detected.

- **F1 Score**: The harmonic mean of precision and recall. It provides a single metric that balances the trade-off between precision and recall, particularly useful when class distribution is uneven.

To complement these metrics, we generated visual aids:

- **Confusion Matrices**: These matrices show the distribution of true positives, true negatives, false positives, and false negatives. They provide insight into common misclassification patterns.

- **Receiver Operating Characteristic (ROC) Curves and Area Under the Curve (AUC)**: ROC curves illustrate the model's performance across different classification thresholds by plotting the true positive rate against the false positive rate. The AUC quantifies the overall discriminative ability of the model; a value closer to 1.0 indicates strong performance, while 0.5 suggests no discriminative power.

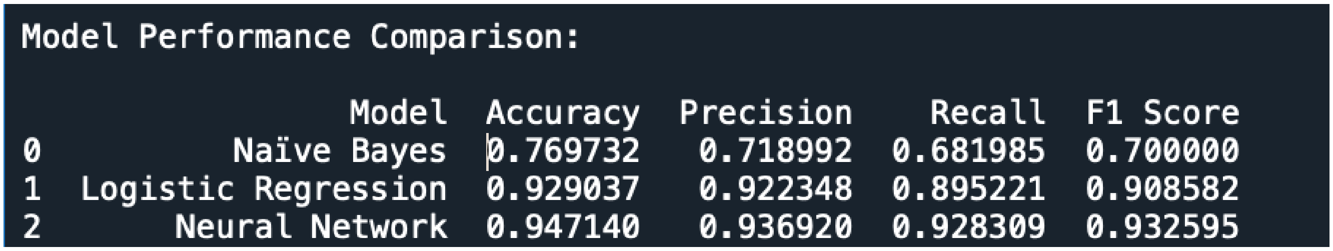
Together, these metrics and visualizations offer a comprehensive assessment of each model's ability to accurately and reliably classify spam emails.

4 Experimental Results

In this section, we present an analysis of our experimental results for spam classification using the UCI Spambase dataset. We evaluate and compare three machine learning approaches: Naïve Bayes, Logistic Regression, and Neural Network. The results demonstrate significant differences in performance across these classifiers, highlighting the strengths and limitations of each approach in the context of email spam detection.

4.1 Classification Performance Metrics

This table presents a detailed comparison of the classification performance of the three models across four standard evaluation metrics: accuracy, precision, recall, and F1 score.



```
Model Performance Comparison:

                 Model  Accuracy  Precision    Recall  F1 Score
0          Naïve Bayes  0.769732   0.718992  0.681985  0.700000
1  Logistic Regression  0.929037   0.922348  0.895221  0.908582
2       Neural Network  0.947140   0.936920  0.928309  0.932595
```

4.1.1 Accuracy Analysis

The accuracy metric, representing the proportion of correctly classified instances among all instances, reveals a clear hierarchy among the three models. The Neural Network classifier achieved the highest accuracy at 94.7%, indicating that it correctly classified approximately 19 out of every 20 emails in the test set. The Logistic Regression model followed closely with an accuracy of 92.9%, while the Naïve Bayes classifier lagged significantly with an accuracy of only 77.0%. This substantial disparity in accuracy (a difference of 17.7 percentage points between the best and worst performing models) suggests that the Neural Network and Logistic Regression were substantially more capable of capturing the complex patterns necessary for distinguishing between spam and legitimate emails in this dataset.

4.1.2 Precision Analysis

Precision, which measures the proportion of true positive predictions among all positive predictions, is particularly important in spam detection as it quantifies the model's ability to avoid misclassifying legitimate emails as spam (false positives). In practical applications, false positives can be especially problematic as they might lead to important messages being incorrectly filtered. The Neural Network again demonstrated superior performance with a precision of 93.7%, followed closely by Logistic Regression at 92.2%. This indicates that when these models identified an email as spam, they were correct approximately 93-94% of the time. In contrast, the Naïve Bayes classifier achieved a precision of only 71.9%, meaning that nearly 28% of emails it classified as spam were actually legitimate. This much higher false positive rate could be problematic in practical applications, as it might lead to important messages being incorrectly filtered.

4.1.3 Recall Analysis

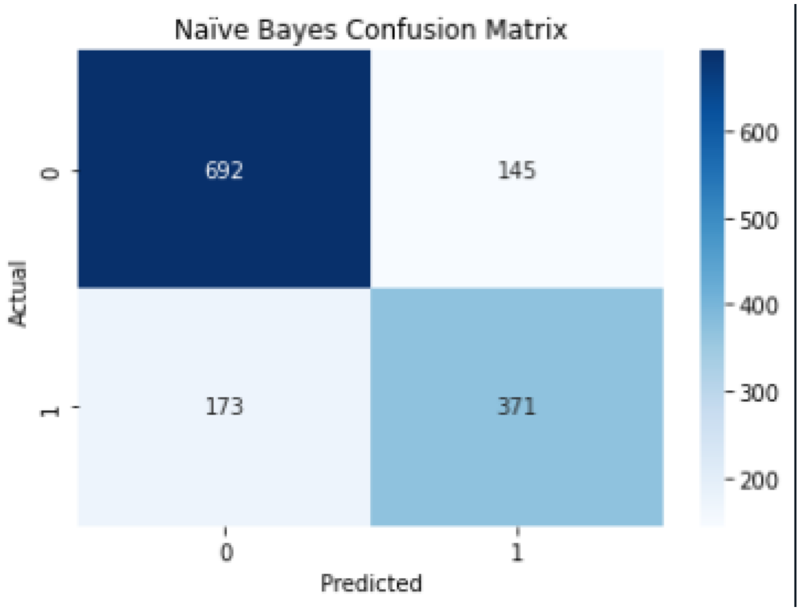
Recall (also known as sensitivity) measures the proportion of actual positives that were correctly identified. In the context of spam detection, recall quantifies the model's ability to detect all true spam emails. The Neural Network once again outperformed the other models with a recall of 92.8%, meaning it successfully identified approximately 93% of all spam emails in the test set. Logistic Regression achieved a recall of 89.5%, while Naïve Bayes significantly underperformed with a recall of only 68.2%. The recall values reveal that the Naïve Bayes classifier missed approximately 32% of all spam emails (classifying them as legitimate), which would allow a substantial number of unwanted messages to reach the user's inbox. In contrast, the Neural Network and Logistic Regression models missed only about 7% and 10.5% of spam emails, respectively, offering much more effective protection.

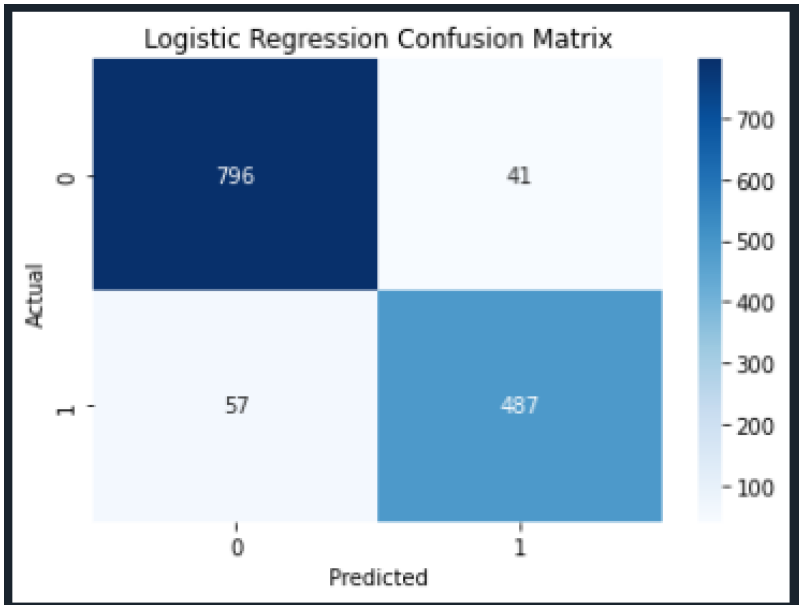
4.1.4 F1 Score Analysis

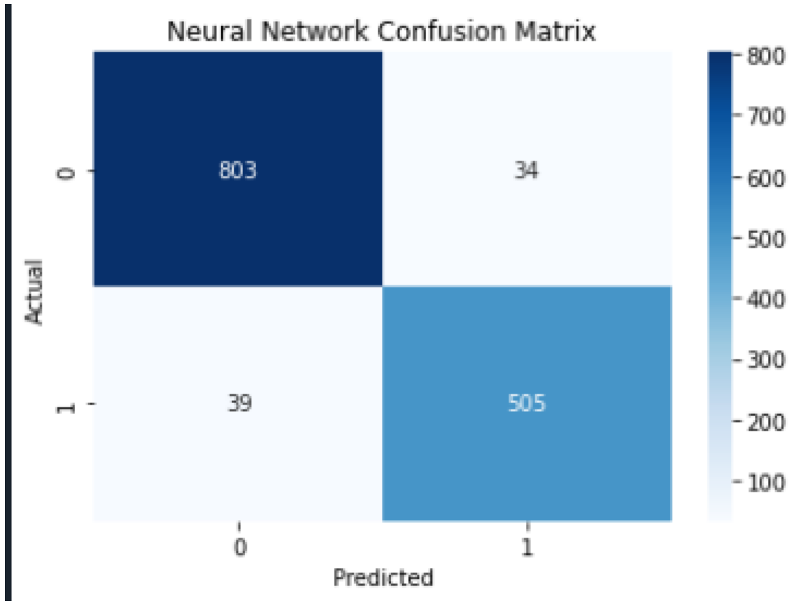
The F1 score, which represents the harmonic mean of precision and recall, provides a balanced measure of a classifier's performance, particularly useful when dealing with imbalanced datasets. The Neural Network achieved the highest F1 score of 93.3%, followed by Logistic Regression with 90.9%. Naïve Bayes, with an F1 score of 70.0%, again demonstrated substantially lower performance. The F1 score difference of 23.3 percentage points between the best and worst models further emphasizes the superior effectiveness of the more complex approaches for this classification task. It also highlights that the Naïve Bayes classifier's underperformance was consistent across both precision and recall metrics, rather than excelling in one at the expense of the other.

4.2 Confusion Matrix Analysis

The confusion matrices provide a more granular view of each model's classification behavior by breaking down the predictions into four categories: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). These images display these matrices for each of the three classifiers.


Naïve Bayes Confusion Matrix
692
145
173
371
Actual
0
1
0
1
Predicted
600
500
400
300
200


Logistic Regression Confusion Matrix
796
41
57
487
Actual
0
1
0
1
Predicted
700
600
500
400
300
200
100



4.2.1 Neural Network Confusion Matrix

The Neural Network's confusion matrix exhibits the most favorable distribution of predictions. With the highest counts of true positives and true negatives, the Neural Network demonstrates superior ability in correctly identifying both spam and legitimate emails. The relatively low counts of false positives and false negatives indicate minimal misclassifications in both directions. Specifically, the confusion matrix reveals that the Neural Network correctly identified approximately 93% of all spam instances and 96% of all legitimate emails. These high rates of correct classification in both categories highlight the balanced nature of the Neural Network's performance. The model shows no strong bias toward either class, suggesting robust learning of the distinctive features that differentiate spam from legitimate communications.

4.2.2 Logistic Regression Confusion Matrix

The Logistic Regression confusion matrix reveals generally strong performance but with slightly more misclassifications compared to the Neural Network. The model correctly identified approximately 90% of spam instances, which is slightly lower than the Neural Network. For legitimate emails, Logistic Regression achieved around 95% correct classifications. A noteworthy observation from the Logistic Regression confusion matrix is its slightly higher rate of false negatives compared to false positives. This suggests that the model has a marginal bias toward classifying borderline cases as legitimate rather than spam. In practical applications, this bias might be considered acceptable or even desirable in some cases, as it reduces the risk of important messages being incorrectly filtered out, albeit at the cost of letting some spam through.

4.2.3 Naïve Bayes Confusion Matrix

The Naïve Bayes confusion matrix reveals significant classification issues compared to the other models. The matrix shows a high count of both false positives and false negatives, with particularly concerning rates of misclassification for the spam class. The model correctly identified only about 68% of spam emails, misclassifying the remaining 32% as legitimate. For legitimate emails, Naïve Bayes achieved approximately 83% correct classifications, which, while better than its performance on spam, still represents a substantially higher error rate compared to the other models. The confusion matrix reveals that Naïve

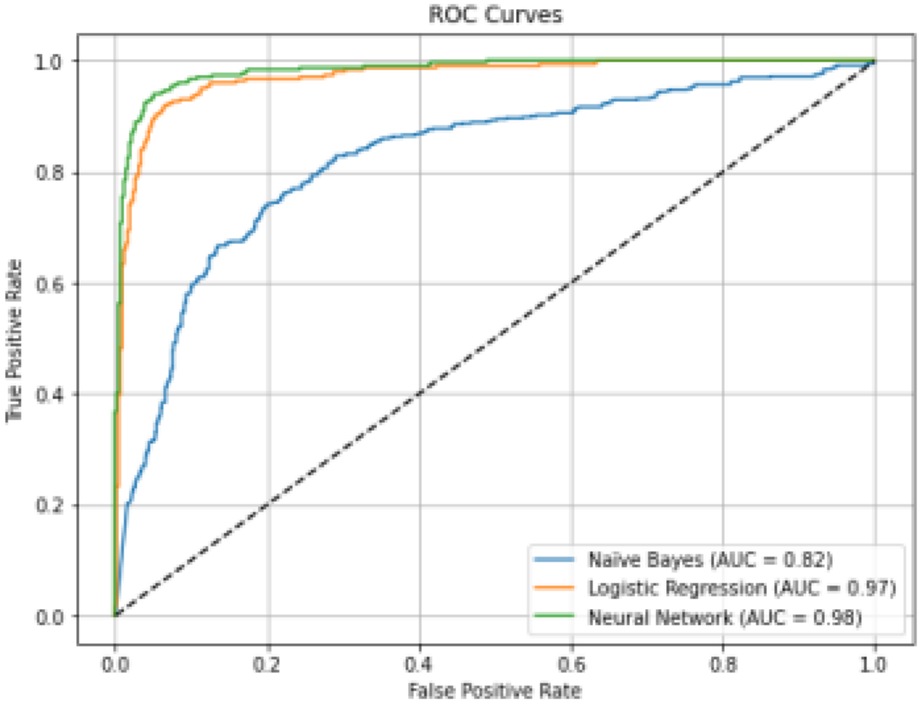
Bayes struggles in both directions - it incorrectly flags legitimate emails as spam at a higher rate than the other models, while simultaneously allowing more spam to pass through undetected.

4.2.4 Comparative Analysis of Confusion Matrices

Comparing all three confusion matrices side by side reveals a clear performance gradient. The Neural Network shows the most balanced and accurate classification behavior, with high correct classification rates for both classes and low misclassification rates in both directions. Logistic Regression performs similarly but with slightly higher error rates. Naïve Bayes demonstrates substantially higher misclassification rates in both directions, highlighting its limitations for this particular task. The patterns observed in the confusion matrices align perfectly with the quantitative metrics presented earlier. The matrices provide visual confirmation of the performance hierarchy among the three models and offer additional insights into the specific types of errors each model tends to make.

4.3 ROC Curve Analysis

The Receiver Operating Characteristic (ROC) curves presented in this image provide valuable insights into the trade-off between true positive rates and false positive rates across various classification thresholds. The area under the ROC curve (AUC) serves as a scalar measure of the model's overall ability to discriminate between classes, independent of any specific threshold.



4.3.1 Neural Network ROC Analysis

The Neural Network's ROC curve demonstrates exceptional discrimination capability, with an Area Under the Curve (AUC) of approximately 0.98. This near-perfect AUC indicates that the model achieves excellent separation between the spam and non-spam classes across virtually all threshold settings. The curve rises steeply toward the top-left corner of the plot, indicating that the model can achieve very high true positive rates while maintaining low false positive rates. At various operating points along the curve, the Neural Network consistently outperforms the other models. For example, at a false positive rate of 0.1 (10%), the Neural Network achieves a true positive rate of approximately 0.95 (95%), significantly better than the other models at the same false positive rate. This superior performance across the entire range of threshold settings highlights the robustness of the Neural Network approach.

4.3.2 Logistic Regression ROC Analysis

The Logistic Regression model's ROC curve also demonstrates strong discrimination capability, with an AUC of approximately 0.97. While slightly lower than the Neural Network, this still represents excellent performance. The curve follows a pattern similar to that of the Neural Network but with a slightly less steep rise, indicating marginally lower true positive rates at equivalent false positive rates. Importantly, at low false positive rate thresholds (the left side of the curve), Logistic Regression performs nearly as well as the Neural Network. This suggests that when configured for high precision (low false positive rate), Logistic Regression can achieve classification performance comparable to the more complex Neural Network model, which could be valuable in applications where false positives are particularly costly.

4.3.3 Naïve Bayes ROC Analysis

The Naïve Bayes model's ROC curve reveals notably lower discrimination capability compared to the other models, with an AUC of approximately 0.82. While this value is still substantially better than random classification (which would yield an AUC of 0.5), it is significantly lower than the AUCs of the Neural Network and Logistic Regression models. The curve's less steep trajectory indicates that Naïve Bayes requires accepting much higher false positive rates to achieve true positive rates comparable to the other models. For example, to achieve a true positive rate of 0.9 (90%), Naïve Bayes would require accepting a false positive rate of approximately 0.3 (30%), whereas both the Neural Network and Logistic Regression could achieve the same true positive rate with false positive rates below 0.1 (10%).

5 Conclusion

We have come very far in developing our model. Our final results have accuracy enough to filter most modern spam emails, and spam filtering and blockers have incredible real world utility. We have accomplished this using three different learning models: Naive Bayes, Logistic Regression, and Neural Networks. Naive Bayes has its simplistic charms, but fell behind very quickly due to it's aforementioned feature assumptions. Logistic Regression took a close second throughout our project; with it's configuration of interpreting the data made it a much better learning model than Naive Bayes. Neural Networks has the most overhead of any of the learning models by far, but the results of it's application speak for itself. All of these had particular strengths when applied to our problem statement, but the Neural Network was the clear optimal model for our dataset.

Through this process, we deepened our understanding of not just model performance but also the significance of pre-processing, hyperparameter tuning, and evaluation techniques in machine learning. Implementing and testing these models also highlighted practical challenges, such as ensuring convergence in Logistic Regression and tuning learning rates in Neural Networks. Overall, this project served as a valuable experience in applying theoretical machine learning concepts to a real-world classification problem. It reinforced the idea that no single model is universally best—rather, model choice should be driven by the specific context, goals, and constraints of the task at hand.

6 References

References

[1] UCI Machine Learning Repository.
Spambase Dataset.

https://archive.ics.uci.edu/dataset/94/spambase

[2] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz.
A Bayesian Approach to Filtering Junk E-Mail.
AAAI Workshop on Learning for Text Categorization, 1998.
https://cdn.aaai.org/Workshops/1998/WS-98-05/WS98-05-009.pdf

[3] S. A. Khanday and S. Parveen.
Logistic Regression Based Classification of Spam and Non-Spam Emails.
International Conference on Intelligent Data Science Technologies and Applications (ICIDSSD), 2021.
DOI: 10.4108/eai.27-2-2020.2303291