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

Project Midterm Report

Team Members: Ryan Riebesehl, Savannah Franklin

Abstract

Our primary task is to compare 3 machine learning models: Naive Bayes, Logistic Regression, and Neural Networks in spam detection. Utilizing the UCI Spambase dataset containing an extensive list of emails marked as spam or not spam accordingly, we can run this through each model to test their accuracy. Having had exposure to data science and the effects of both poorly and well-executed spam detection, we wish to explore further the extent to which these models work.

Currently, we have a basic Python program that tests the accuracy of the 3 models. Utilizing the data provided by the dataset, we can determine which model most accurately interprets the provided information. Our results currently show that the Logistic Regression model performs the best overall, while Naive Bayes tends to perform worse than our other models. Moving forward, we need to further explore and test for the most optimal model.

1 Introduction

Email spam has been an issue since the beginning of online communication. These unwanted messages tend to clog up people's inboxes, which can waste time and pose security risks like phishing scams. While traditional spam filters use fixed rules to block spam, they often struggle to keep up with the ever-changing tricks used by spam emails.

In this project, we are testing three popular machine-learning models to see which one handles spam detection the best. Naive Bayes is very fast and is great for text analysis. Logistic Regression is simple and easy to understand conceptually. Neural Networks are more powerful at spotting complex patterns but need more computing power to execute.

Our analysis is based on the UCI Spambase dataset
(available at: https://archive.ics.uci.edu/dataset/94/spambase), a well-established benchmark in spam detection research. The dataset has 4,601 emails, each represented by 57 numerical features, including word and character frequencies, as well as other different email attributes. We hypothesize that Neural Networks will achieve the highest accuracy due to their ability to model complex relationships within the data. However, we also consider other factors, such as computational efficiency and interpretability, which are critical for real-world applications.

2 Related Work

Spam detection has been extensively studied in the field of machine learning, with numerous approaches proposed over the years. One of the earliest techniques is the Naive Bayes classifier, which assumes feature independence and calculates the probability of an email being spam based on word occurrences. While this method has demonstrated strong performance in text classification tasks, its simplifying assumptions often lead to reduced accuracy when dealing with complex patterns in spam emails.

Logistic Regression has also been a popular choice for spam classification due to its probabilistic interpretation. Studies have shown that with proper hyperparameter tuning, Logistic Regression can achieve high classification accuracy. Unlike Naive Bayes, Logistic Regression does not assume feature independence, allowing it to capture more relationships between words in emails. However, it remains limited by its linear decision boundary, which may not effectively separate spam from non-spam emails in all cases.

With the rise of deep learning, Neural Networks have emerged as a powerful tool for spam detection. Research has demonstrated that deep learning models can extract complex patterns from text data, often outperforming traditional machine learning algorithms. Neural networks have been particularly successful in processing email text sequences even though they require significant computational resources and extensive tuning. Some studies suggest that while Neural Networks offer great accuracy, their deployment in real-time spam filtering systems may be challenging due to latency issues.

Several comparative studies have evaluated these machine learning models in spam detection, highlighting the trade-offs between accuracy, interpretability, and computational efficiency. For instance, a study by Sahami et al. (1998) explored Bayesian approaches to junk email filtering. Demonstrating the effectiveness of probabilistic models. More recent research has investigated ensemble learning techniques which combine multiple classifiers to achieve higher accuracy. Our project builds on these findings by directly comparing Naive Bayes, Logistic Regression, and Neural Networks using the UCI Spambase dataset with a focus on balancing performance with good accuracy.

3 Methods

3.1 Dataset Collection and Preprocessing:

*We use the UCI Spambase dataset, which consists of 4,601 emails labeled as spam or not spam.
*The dataset includes 57 features representing word frequencies, character frequencies, and other email attributes.
*We split the dataset into 70% training (3,220 emails), 20% testing (920 emails), and 10% validation (460 emails).
*Data preprocessing includes normalization, handling missing values, and feature selection.
*Scaling is applied to ensure consistency across models, particularly for Logistic Regression and Neural Networks, which tend to be sensitive.
*We use stratified sampling to ensure an even distribution of spam and non-spam emails across training, testing, and validation sets.

3.2 Machine Learning Models:

Naive Bayes:
*A probability-based classifier that assumes independence between features.
*We use the Multinomial Naive Bayes implementation from the Scikit-learn library.
*Strengths: Fast and very effective for text classification.
*Weaknesses: Assumes feature independence, which may not always hold during the analysis.

Logistic Regression:
*A linear model that predicts the probability of an email being spam.
*We implement Logistic Regression using Scikit-learn's LogisticRegression() function.

*Strengths: Pretty simple and performs well on structured data.
*Weaknesses: May struggle with complex, non-linear relationships in data.

Neural Networks:
*A deep learning model capable of capturing very complex patterns.
*We implement a Neural Network using the PyTorch library.
*Strengths: Can learn non-linear relationships and adapt to more complex patterns.
*Weaknesses: Computationally expensive and requires extensive tuning.

3.3 Evaluation Metrics:

*Accuracy: Percentage of correctly classified emails.
*Precision: Ratio of correctly predicted spam emails to total predicted spam emails.
*Recall: Ratio of correctly predicted spam emails to total actual spam emails.
*F1-score: Mean of precision and recall to balance the false positives and false negatives.

4 Preliminary Results

Table with Initial Results:

```
IPython 8.12.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/ryanriebesehl/.spyder-py3/PerliminartyResults.py',
wdir='/Users/ryanriebesehl/.spyder-py3')
/Users/ryanriebesehl/anaconda3/lib/python3.11/site-packages/sklearn/linear_model
_logistic.py:460: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressio
  n_iter_i = _check_optimize_result(
                Model  Accuracy  Precision    Recall  F1 Score
0         Naïve Bayes  0.782042   0.762357  0.694974  0.727108
1  Logistic Regression  0.929761   0.937956  0.890815  0.913778
2       Neural Network  0.918175   0.881579  0.928943  0.904641
```

Logistic Regression Performs Best Overall:
*Highest Accuracy (92.98%): Most emails are classified correctly.
*Best Precision (93.80%): It minimizes false positives, meaning fewer non-spam emails are mistakenly marked as spam.
*High Recall (89.08%): It successfully detects most spam emails while keeping the false negatives low.
*Strong F1 Score (91.38%): Good balance between precision and recall.

Neural Networks Perform Slightly Worse than Logistic Regression:
*Accuracy (91.82%) is slightly lower than logistic regression.
*Recall (92.89%) is the highest, meaning it detects more actual spam than the other models.
*Precision (88.16%) is lower than logistic regression, suggesting it classifies some non-spam emails as spam.

*Possible reason for underperforming: The model might be overfitting and needs more hyperparameter tuning.

Naïve Bayes Underperforms Compared to the Other Models:
*Lowest accuracy (78.20%) meaning it makes more mistakes than the other models.
*Lower precision (76.24%) suggests it is more prone to marking legitimate emails as spam.
*Lowest recall (69.50%) means it misses many actual spam emails.
*F1 Score (72.71%) indicates that it struggles to maintain a good balance.
*Possible reason for underperforming: Naïve Bayes assumes feature independence, which is unrealistic in spam detection (e.g., words like "free" and "win" often appear together).

5 Future plans

.

*Mess more with tuning hyperparameters for each model to optimize the performances.
*Experiment more with additional feature selection techniques.
*Test different Neural Networks.
*Explore a variety of ensemble learning methods.

6 References

*UCI Machine Learning Repository: Spambase Dataset. (https://archive.ics.uci.edu/dataset/94/spambase).
*Sahami, M., et al. (1998). A Bayesian Approach to Filtering Junk E-Mail.
(https://cdn.aaai.org/Workshops/1998/WS-98-05/WS98-05-009.pdf)