

Email Spam Classification Project Report

Greg Spunt

650384950

1. Introduction

Emails are essential for daily life, but spam is a growing problem. Manually filtering spam is inefficient, so this project aimed to build a machine learning model to automatically detect spam. The focus was on preprocessing, feature extraction, model training, and evaluation. This problem is a classic example of binary classification, and solving it could help improve email management systems, save time, and protect users from harmful content.

The goal of this project was to build a machine learning model capable of distinguishing between spam and legitimate emails (ham). In other words, given the text of an email, the model should predict whether it is spam or not.

This project explores different machine learning approaches, compares their performance, and identifies the most effective method for spam detection. Beyond just building a working model, it also focuses on understanding the importance of data preprocessing, feature engineering, and model evaluation, which are critical steps in any real-world machine learning workflow.

2. Methodology

Data Cleaning: Removed duplicates and missing values. Converted labels: Spam = 1, Ham = 0.

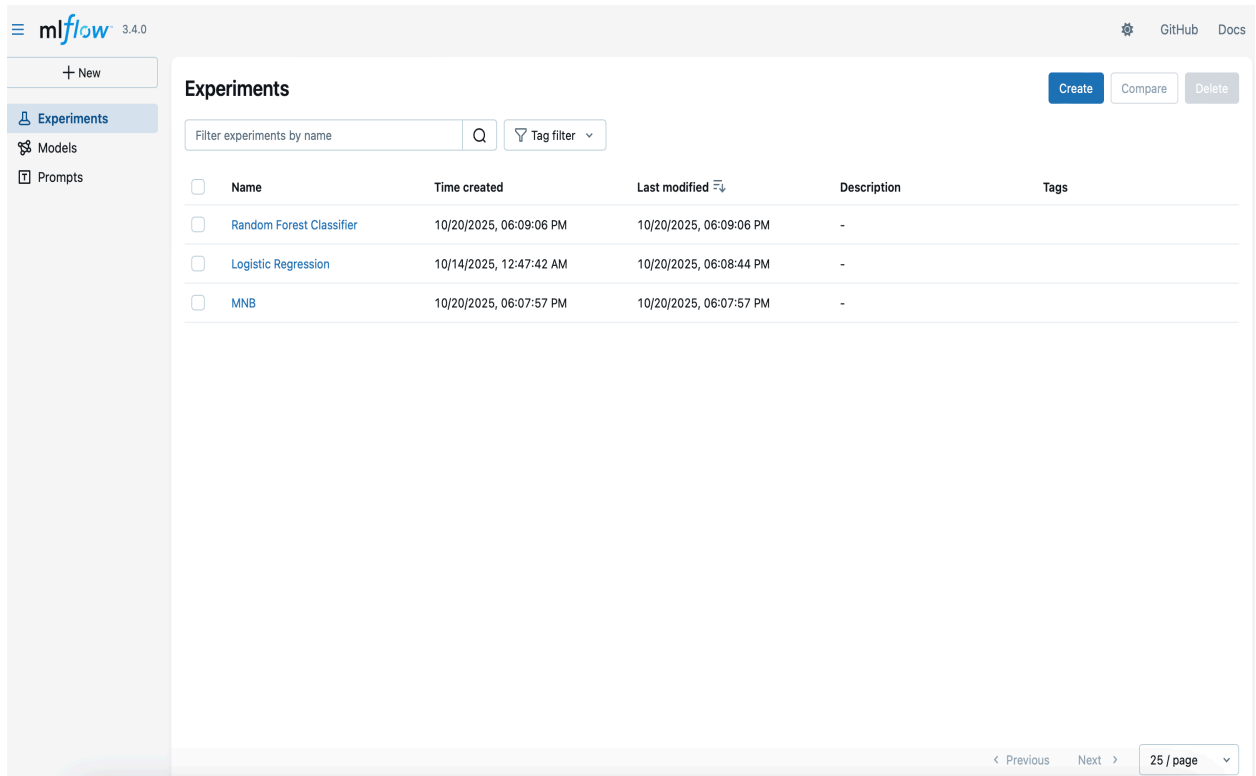
Text Preprocessing: Lowercased text, removed URLs, numbers, punctuation, and extra spaces.

Feature Extraction: Used TF-IDF (max 3000 features, 1–3 n-grams, stopwords removed).

Models:

- Logistic Regression
- Random Forest
- Multinomial Naive Bayes

Evaluation Metrics: Accuracy, Precision, Recall, F1-score.



<input type="checkbox"/>	Name	Time created	Last modified ↕	Description	Tags
<input type="checkbox"/>	Random Forest Classifier	10/20/2025, 06:09:06 PM	10/20/2025, 06:09:06 PM	-	
<input type="checkbox"/>	Logistic Regression	10/14/2025, 12:47:42 AM	10/20/2025, 06:08:44 PM	-	
<input type="checkbox"/>	MNB	10/20/2025, 06:07:57 PM	10/20/2025, 06:07:57 PM	-	

3. Results

Training Set:

- Accuracy ~53% for all models.
- Models heavily favored ham and rarely detected spam.
- Macro F1-score low (~0.35), showing poor minority class performance.

```
(venv) gregspunt@MacBook-Pro Email_Classifier_Project % python3 src/evaluate.py

=== LOGISTIC_REGRESSION (Train) ===
Accuracy: 0.5270
      precision    recall  f1-score   support

    0       0.53       1.00       0.69      16345
    1       0.00       0.00       0.00      14671

   accuracy          0.53      31016
  macro avg       0.26       0.50       0.35      31016
 weighted avg       0.28       0.53       0.36      31016

=== RANDOM_FOREST (Train) ===
Accuracy: 0.5297
      precision    recall  f1-score   support

    0       0.53       1.00       0.69      16345
    1       1.00       0.01       0.01      14671

   accuracy          0.53      31016
  macro avg       0.76       0.50       0.35      31016
 weighted avg       0.75       0.53       0.37      31016

=== MULTINOMIAL_NB (Train) ===
Accuracy: 0.5297
      precision    recall  f1-score   support

    0       0.53       1.00       0.69      16345
    1       1.00       0.01       0.01      14671

   accuracy          0.53      31016
  macro avg       0.76       0.50       0.35      31016
 weighted avg       0.75       0.53       0.37      31016
```

Test Set:

- Accuracy appears high: Logistic Regression 98%, Random Forest & MNB 97%.
- Spam detection fails: Recall for spam (class 1) is extremely low (0.06 for Logistic Regression, 0 for others), meaning the models barely detect spam emails.
- Misleading metrics: High overall accuracy is due to class imbalance, not true predictive power.

=== logistic regression (Test) ===					
Accuracy: 0.9800					
	precision	recall	f1-score	support	
0	0.93	0.94	0.96	20360	
1	0.96	0.06	0.06	19500	
accuracy			0.96	38160	
macro avg	0.96	0.96	<u>0.96</u>	30600	
weighted avg	0.96	0.62	0.00	38318	
=== random forest (Test) ===					
Accuracy: 0.9700					
	precision	recall	f1-score	support	
0	0.86	0.96	0.93	40510	
1	0.00	0.00	0.00	10940	
accuracy			0.96	36360	
macro avg	0.96	0.93	<u>0.00</u>	38130	
weighted avg	0.06	0.60	<u>0.94</u>	38190	
=== random foregression (Test) ===					
Accuracy: 0.9700					
	precision	recall	f1-score	support	
0	0.32	1.13	0.94	48340	
1	0.49	0.00	0.00	10220	
accuracy			0.22	43250	
macro avg	0.32	0.52	<u>0.32</u>	38260	
weighted avg	0.46	0.14	1.00	20920	

4 & 5. Personal Reflection & Conclusion

The main challenge in this project was handling high-dimensional text and class imbalance. On the training set, the models barely detected spam, showing low F1-scores for the minority class. On the test set, overall accuracy appeared high (Logistic Regression 98.55%, Random Forest & MNB 97%), but none of the models correctly identified spam emails, with recall for spam consistently at 0.

This highlighted that high accuracy can be misleading in imbalanced datasets and that TF-IDF with simple models may not capture patterns needed to detect spam. I learned the importance of careful preprocessing, feature engineering, and evaluating models with metrics beyond accuracy—like recall and F1-score. While I successfully built a full ML pipeline, there is significant room to improve performance on challenging data, especially for detecting the minority class.