12/19/2025

Spam Likely

A Supervised Machine Learning Spam Email Classifier

Project Final Report

Jackson Trader	jtrader@purdue.edu
Alex Bryant	bryan140@purdue.edu
Diego Lara	larad@purdue.edu
Mark Pack	packm@purdue.edu

Table of Contents

1. Abstract

Email spam continues to be a significant and common issue today, and it's only becoming increasingly difficult to distinguish from legitimate emails. Therefore, having a machine learning algorithm that can classify emails as spam or legitimate (ham) can improve work environments. Our goal of this project was to demonstrate and practice our skills for building a supervised machine learning system that could classify emails based on their textual content, or message. The system that we designed converts our message content to TF-IDF numerical feature vectors, so the computer can understand the text. We then trained three models—Logistic Regression, Multinomial Naïve Bayes, and LinearSVC—and compared their performances through a series of classification metrics provided by Scikit-learn. These metrics included accuracy, precision, recall, and an F1 score. The results showed that the LinearSVC model had the strongest overall performance. The baseline models—Logistic Regression and Multinomial Naïve Bayes—still performed well, but not as well as LinearSVC. Additional tests were also conducted to explore potential future improvements.

2. Introduction

Motivation

Email remains one of the most widely used forms of communication in the professional world. Email spam is a global issue that affects individuals and businesses in varying ways. As spam becomes increasingly more difficult to identify from legitimate emails, companies have been trying to upgrade their email spam classification algorithms. Because email spam contains text patterns, using supervised machine learning can be effective in identifying and classifying these emails if the dataset contains many meaningful patterns.

Problem Statement / Research Question

The main problem of this project is whether supervised machine learning models could accurately and effectively classify emails as spam or legitimate using only the message content. To understand this problem in more detail, we used two baseline models—Logistic Regression and Multinomial Naïve Bayes—to act as a baseline for our more advanced model, LinearSVC. Each model used the same feature representation, TF-IDF, and data splits. We also looked at how changing preprocessing steps, like lowercasing versus not lowercasing text, would impact the performance of our models. We examined how well the models could generalize by also testing them on an unseen dataset with different formatting.

Brief Outline of the Report

Our final report begins with introducing the problem with classifying emails as spam or legitimate using a supervised machine learning model. Furthermore, it explains our motivation towards choosing this task of classifying emails. Next, we provide background information on text classification, representing messages as feature vectors via TF-IDF, and the baseline algorithms we chose for this project. We then explain our methodology; this includes preprocessing the data, extracting features from the messages into numerical vectors, selecting our baseline and primary models, and finally evaluating our models on several experiments. The experimental section discusses the experiments we performed, including the primary comparison between LinearSVC and the baseline models, changing n-gram ranges, lowercasing versus not lowercasing text for the ablation study, and evaluating LinearSVC on unseen/paraphrased data. Next, we discuss our results of all of these experiments and the project as a whole, with a focus on the model's ability to generalize to unseen data. Finally, the report ends with a summary of our key takeaways and a discussion about improvements that could be made in the future.

3. Background

Theoretical Background

For our project, we used binary classification for classifying emails. An email was either spam or legitimate. This is a supervised learning task based solely on the message context of the email. Since machine learning models cannot process text, the messages of each email had to be converted into numerical feature vectors using TF-IDF. TF-IDF, or Term Frequency-Inverse Document Frequency, is used to represent text as numbers. It measures how important a word is in a message based on its frequency in the entire dataset. This helps give importance to rare words over common words. There are also other methods such as Bag-of-Words and Vector Space Models to represent text as numerical feature vectors, but for this project we stuck with TF-IDF.

In addition to using TF-IDF, we also looked at changing the hyperparameters of it. We tested n-gram ranges, which represent groups of consecutive words. This allows the model to capture more context when analyzing text rather than evaluating each word independently. We conducted only a small n-gram experiment, as our main focus was evaluating LinearSVC's overall performance and generalization.

To evaluate the performance of each model, we used Scikit-learn's classification metrics. Accuracy measures how often the model is correct overall. Precision measures how often the prediction of spam is actually spam. Recall measures how many actual spam emails the model found. The F1 score combines precision and recall into a single metric, so it is very useful for understanding the performance of a model if a dataset is imbalanced. In our case, our dataset was imbalanced, so we heavily relied on the F1 score to conclude that LinearSVC performed the best. Among our dataset, 87.57% of the emails were legitimate and 12.43% were spam. This is the dataset we used to train and evaluate our models. We also included an unseen dataset from Kaggle to evaluate furthermore.

Background for the Task / Baseline Algorithms

Email classification is a popular choice for machine learning projects and has been implemented in all sorts of ways. Because email messages can be represented as numerical feature vectors using TF-IDF and other similar methods, supervised machine learning algorithms can be used to solve this problem. In this project we focused on comparing multiple models, simple and complex, using the same conditions so we could better understand how different models respond to email spam classification.

Logistic Regression and Multinomial Naïve Bayes were chosen as our baseline models because they are simple and we learned about them already in our lectures. Logistic Regression is a linear classification model that estimates the probability of an email being either spam or legitimate based on its input features—our TF-IDF feature vectors. Multinomial Naïve Bayes is a probabilistic model that uses Bayes' theorem and word frequencies to estimate the likelihood that an email is either spam or legitimate. It assumes that features are independent, hence the term Naïve.

Additionally, we used a LinearSVC model, or more broadly a Linear Support Vector Machine. This is our advanced model that is well known to work on high-dimensional feature vectors. LinearSVC works by finding a decision boundary (hyperplane) that best separates binary classes with the highest margin possible. So, by comparing the performance of LinearSVC with our two baseline models, we were able to evaluate whether a more complex model would show improved performance when classifying spam emails.

4. Methodology / Approach

In this project, we followed the standard supervised machine learning approach used in similar projects. The process included loading and preprocessing the dataset, converting email messages into numerical feature vectors via TF-IDF, training multiple classification models, and evaluating their performance using Scikit-learn's metrics.

The dataset that we used contained a column labeled "spam" or "ham" (legitimate) and a message column for storing each email's message. Before training the models, the data was cleaned by removing missing and duplicate values to ensure it was ready for training. The dataset was split into training and testing sets using stratification to preserve the 80/20 percent distribution, since the dataset was heavily imbalanced.

Since machine learning models cannot process text, we used TF-IDF to convert each email message into numerical feature vectors. We chose TF-IDF over other vectorizers like Bag-of-Words and Vector Space Models because it weighs words by their importance and removes common words, like stopwords (ex. "the" and "is"). TF-IDF was used for each model to allow a fair comparison between all three models.

We trained three supervised machine learning models: Logistic Regression, Multinomial Naïve Bayes, and LinearSVC. We chose Logistic Regression and Multinomial Naïve Bayes as our baseline models because of their simplicity because we have previous knowledge about them from our previous lectures. We chose LinearSVC as our primary model because it is more advanced and is well known to work on high-dimensional feature vectors such as the ones returned by our TF-IDF vectorizer. All of our models were trained on the same training data and evaluated on the same test data for fairness.

We used Scikit-learn's classification metrics to evaluate our models. The metrics we used include accuracy, precision, recall, and an F1 score to evaluate each of our models' performance. Because of the dataset being imbalanced, we relied heavily on the F1 score on each model. We also conducted additional experiments such as an ablation study to see if lowercasing messages would increase performance, and we tested the models on an unseen dataset to evaluate how well they generalize. We also used LLMs to paraphrase a few messages to see how well the models generalize again.

5. Experiments

Experimental Setup

All of our experiments were conducted in a Jupyter Notebook using Python as the programming language, and Google Colab for our IDE. We also used common data science libraries including Pandas and NumPy for data wrangling and computing, Scikit-learn for model training and evaluation, and Matplotlib and WordCloud for visualization. All models were trained and evaluated using the same splits of the dataset and the same TF-IDF feature vectors for a fair comparison.

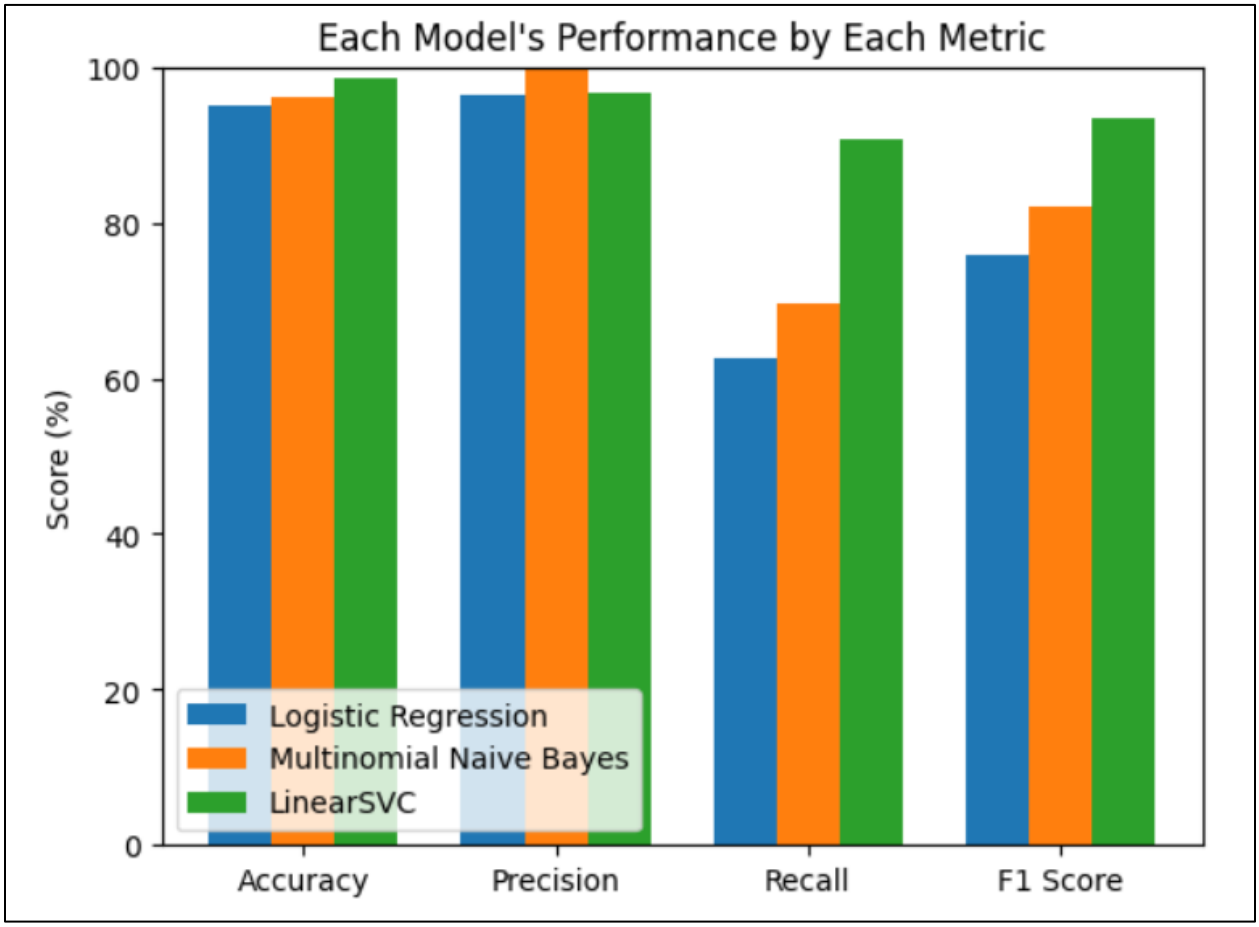
Logistic Regression and Multinomial Naïve Bayes were used as baseline models, while LinearSVC was evaluated against them due to its higher complexity. Each model's performance was evaluated on accuracy, precision, recall, and an F1 score.

Comparison with Baseline Methods (Primary Model)

The primary experiment we focused on was comparing the performance of the three classification models: Logistic Regression, Multinomial Naïve Bayes, and LinearSVC. All the models were trained using the same TF-IDF numerical feature vectors and evaluated on the same testing set. LinearSVC resulted in the highest performance across all the metrics, especially in recall and F1 score. The baseline models—Logistic Regression and Multinomial Naïve Bayes—performed well, but LinearSVC performed better on the testing data split. This further reinforced our idea that LinearSVC would be the better performer in general for email spam classification.

Results from Comparing with Baseline Methods in Table / Bar Chart Formats:

	Model	Accuracy	Precision	Recall	F1 Score
0	Logistic Regression	95.06	96.39	62.50	75.83
1	Multinomial Naive Bayes	96.22	100.00	69.53	82.03
2	LinearSVC	98.45	96.67	90.62	93.55



Experiment on N-Gram Range

This experiment was conducted using the LinearSVC model only, as it demonstrated the strongest overall performance in the primary model comparison, and it was the most complex model. In addition to our primary experiment, we also conducted a small experiment with TF-IDF's hyperparameters. We changed its ngram_range hyperparameter to see if it had an impact on the performance of the LinearSVC model. The results showed that LinearSVC performed best when the n-gram range was set to (1, 3); however, the differences in performance between the three were minimal. This suggests diminishing returns when increasing the n-gram range, and as a result, the default unigram range was sufficient for the experiments we conducted.

Results from N-Gram Range Experiment:

	n-grams	Accuracy	Precision	Recall	F1 Score
0	1-grams	98.55	98.29	89.84	93.88
1	2-grams	98.55	98.29	89.84	93.88
2	3-grams	98.64	98.31	90.62	94.31

Ablation Study

The ablation study was performed using only the LinearSVC model as well to keep the study simple. In this small ablation study, we analyzed the differences in performance when changing the preprocessing steps of the dataset. We looked at the effect of lowercasing messages versus keeping the default message's capitalization. All other factors were kept the same, allowing us to isolate the impact of this preprocessing change on performance. The results showed that lowercasing did not significantly degrade performance, and in some cases it did slightly improve performance for LinearSVC. This experiment helped us identify additional ways that could improve performance. Because we believed that "scammy words" like "FREE" or "CASH" were more present in spam emails, we also believed that we should not lowercase messages to preserve emphasis. From this result alone, we found that capitalization did not significantly impact the metrics.

Results from Ablation Study:

		Accuracy	Precision	Recall	F1 Score
0	Default	98.45	96.67	90.62	93.55
1	Lowercase	98.55	98.29	89.84	93.88

Experiment on Unseen & Paraphrased Data

Evaluation on unseen/paraphrased data was conducted using the LinearSVC model only, as it was the best overall model after our primary experiment. After we found out that LinearSVC was the best performer out of our three models, we wanted to put it to the test on unseen/paraphrased data to see how well it could generalize. Performance on the unseen dataset was lower than on the test split, but the model still performed relatively well despite the different formatting of the messages. The recall and F1 score of the model were severely impacted by the new unseen data, but accuracy and precision remained high. Paraphrased email examples were also generated from LLMs and tested to further see if LinearSVC could generalize well, but we have no metrics for this because of the small number of paraphrased emails we generated.

Results from Unseen Data:

```
LinearSVC on Unseen Data
Accuracy: 83.9 %
Precision: 100.0 %
Recall: 38.42 %
F1 Score: 55.51 %
```

Results from Paraphrased Data:

```
------------------------------------------------
Email: We attempted to reach you regarding your response about the free Nokia phone and camcorder. Call 08000930705 now to arrange delivery for tomorrow.
True Category: spam

Logistic Regression Prediction: legit
Multinomial Naive Bayes Prediction: legit
LinearSVC Prediction: spam

------------------------------------------------
Email: Congratulations! You've been selected to receive a one-year cinema pass for two people. Call 09061209465 today to claim your free offer before it expires.
True Category: spam

Logistic Regression Prediction: spam
Multinomial Naive Bayes Prediction: legit
LinearSVC Prediction: spam

------------------------------------------------
Email: Apologies, I'm currently in a meeting and will call you back later.
True Category: legit

Logistic Regression Prediction: legit
Multinomial Naive Bayes Prediction: legit
LinearSVC Prediction: legit

------------------------------------------------
Email: You'll need to grab yourself a one-dollar burger on the way home. I'm in too much pain to get up right now.
True Category: legit

Logistic Regression Prediction: legit
Multinomial Naive Bayes Prediction: legit
LinearSVC Prediction: legit
```

6. Discussion & Analysis

The results that we obtained after conducting our experiments show that supervised machine learning models could effectively classify emails as spam or legitimate using only message content. Out of all the models, LinearSVC performed the best overall, especially with the recall and F1 score metrics.

The baseline models, Logistic Regression and Multinomial Naïve Bayes, still performed well given the simplicity of their models and they served as a good comparison for LinearSVC. Both baseline models showed weaker performance when compared to LinearSVC because they were simpler models in the first place.

The ablation study helped us determine whether lowercasing messages was an appropriate preprocessing choice. The experiments involving lowercasing messages and adjusting the TF-IDF n-gram range showed diminishing returns, with only slight improvements at best. As a result, we stuck with the default unigram range for our primary experiment of the three models.

Our evaluations on the unseen/paraphrased data further showed that LinearSVC was the best general model for classifying emails as spam or legitimate, even when the text formats/phrasing were different. The performance on the unseen data was approximately half the performance on the test set. This outcome was expected and suggests that increasing the dataset with a wider variety of different phrases/formats could improve generalization.

Overall, these results show that preprocessing data, selecting models, and even the dataset itself, all have a significant impact on the performance when evaluating models. Therefore, more advanced models such as LinearSVC showed improved performance over the chosen baseline models.

7. Conclusion & Future Work

In this project, we looked at the use of supervised machine learning models for classifying emails as spam or legitimate using only the message as context itself. By comparing the baseline models—Logistic Regression and Multinomial Naïve Bayes—with the Linear SVC model, we found that while the baseline models were performant, LinearSVC performed better overall. LinearSVC achieved the highest accuracy, recall, and F1 score. This shows that it is the most reliable model for classifying emails in this scope of this project, when compared to the baseline models. We also conducted additional experiments such as the ablation study (lowercasing versus not lowercasing), n-grams range experiment, and evaluations on unseen/paraphrased data to further prove that email spam classification is a viable task for supervised machine learning models.

There were several areas we found that could be improved in the future. One idea of ours was to explore other preprocessing techniques, such as tuning additional TF-IDF hyperparameters. That could include changing the maximum number of features, stripping accents from letters (ex. é to e) or increasing the n-gram range further beyond just 3-grams. Another area for improvement is how capitalization is preserved. For example, fully capitalized words could be preserved while partially capitalized words would be lowercased. This pattern may help improve model performance.

Beyond message content and preprocessing, future work could incorporate additional email features. For example, we could include more columns for the subject and sender of an email. This could help the model better understand the context of each email. For senders of emails, professional domains often contain recognizable words, while scam domains tend to use random characters. Future work could also compare our baseline models with our ablation study, n-grams range experiment and the unseen/paraphrased data experiment.

Overall, this project shows that even simple supervised machine learning models can perform well on email spam classification. With additional features, preprocessing techniques, and hyperparameter tuning, future work would result in increased performance in spam detection.