Phishing Email Detection

Names: Nicholas Conner, Devon Halford, David Garcia

Date: 5/1/25

Final Project Report

ITS 365 Machine Learning Foundations

Purdue University Northwest

Spring 2025, Hammond, IN

1. Introduction & Research Question (10 points)

In this project, we investigated how machine learning can be used to automatically detect phishing emails based on their text content. Phishing emails are a major cybersecurity threat, often used to steal credentials or spread malware. Traditional email filters struggle to keep up with new phishing techniques, making it critical to develop adaptive tools for email classification.

Our research question is:

Can a machine learning model accurately classify emails as phishing or legitimate using natural language processing (NLP) and text-based features?

This question is highly relevant in today's digital landscape, where email is still a primary attack vector. Accurate phishing detection has the potential to prevent data breaches and financial loss across industries.

2. Dataset (15 points)

We used a comprehensive phishing email dataset compiled from Kaggle to support phishing detection research. The dataset includes content and metadata from the following initial datasets:

Enron and Ling Datasets: Core phishing content including subject lines, body text, and spam/legit labels.

CEAS, Nazario, Nigerian Fraud, and SpamAssassin: Enhanced context, including sender and recipient information, timestamps, and spam classifications.

Final Dataset Overview

Total emails: Approximately 82,000

Spam emails (phishing): 42,891

Legitimate emails: 39,595

Labels: 1 = phishing, 0 = legitimate

The dataset merges multiple trusted sources to provide a diverse and realistic view of phishing activity across multiple formats and contexts.

Exploratory Data Analysis (EDA)

No missing values were found in the dataset columns used.

Label distribution is fairly balanced (~52% phishing, 48% legitimate), ideal for training

Spam/legit distribution was fairly balanced (52% spam, 48% legit), which is ideal for model training.

Visualizations:

A count plot visualized class distribution

Outliers included some unusually short phishing messages, which were retained as realistic attack variants

This robust and balanced dataset supports generalized model learning across phishing styles.

3. Methodology (15 points)

Preprocessing Steps

- Combined subject and body text into a single column: text_combined
- Removed special characters and punctuation using regex
- Converted all text to lowercase
- Removed English stopwords using NLTK
- Created a new column clean_text for model training**Feature Extraction**

Feature Extraction

We applied TF-IDF vectorization to convert the cleaned email content into numerical vectors. This approach emphasizes uncommon but meaningful words and down-weights frequently occurring generic terms.

Model Choice

We selected Logistic Regression for its simplicity, speed, and strong performance on binary classification tasks. It provides a reliable baseline while remaining interpretable.

Training & Testing

- We performed an 80/20 train-test split using train_test_split() with a fixed random seed for reproducibility
- The model was trained on the clean_text TF-IDF vectors and evaluated on a held-out test set of 16,500 emails
- Labels were not exposed during testing**Evaluation**

Evaluation: We assessed the model using:

Precision

Recall

F1-score

Confusion Matrix

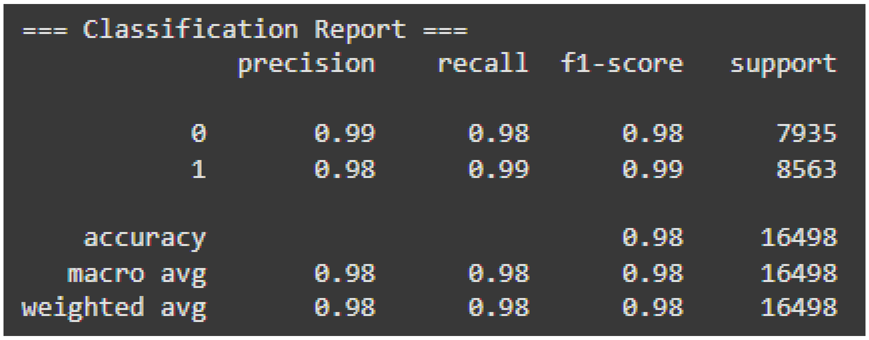
Metric visualizations (bar plots)

These metrics were chosen to evaluate both accuracy and the model's ability to detect phishing without excessive false positives.

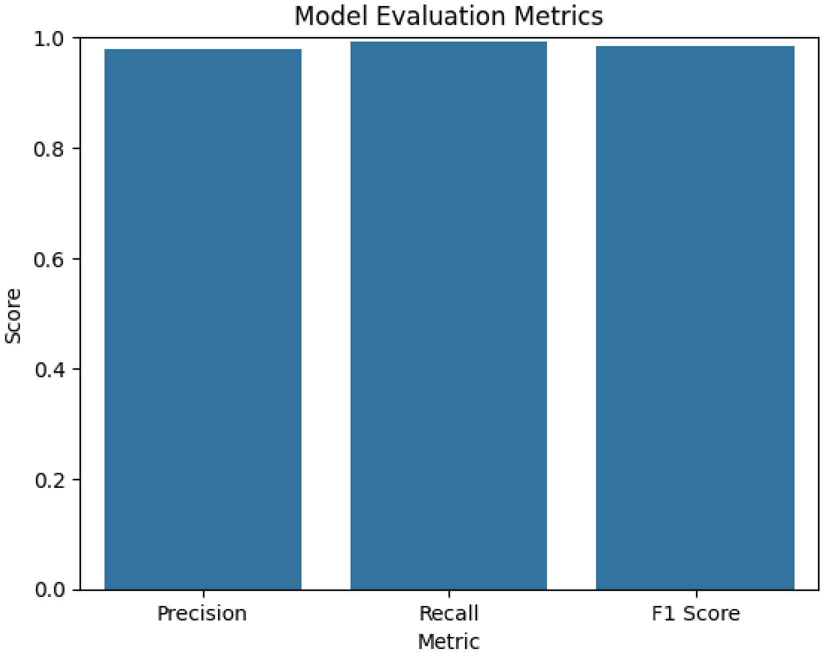
4. Results (20 points)

We evaluated our logistic regression model using a held-out test set of approximately 16,500 emails. The model achieved an overall accuracy of 98%, demonstrating strong performance in detecting phishing emails from text content alone.

For phishing emails (label 1), the model achieved a precision of 0.98, recall of 0.99, and F1 score of 0.99. This means it successfully flagged nearly all phishing messages while generating very few false alarms. Legitimate emails (label 0) showed similarly strong results, with a precision of 0.99, recall of 0.98, and F1 score of 0.98.

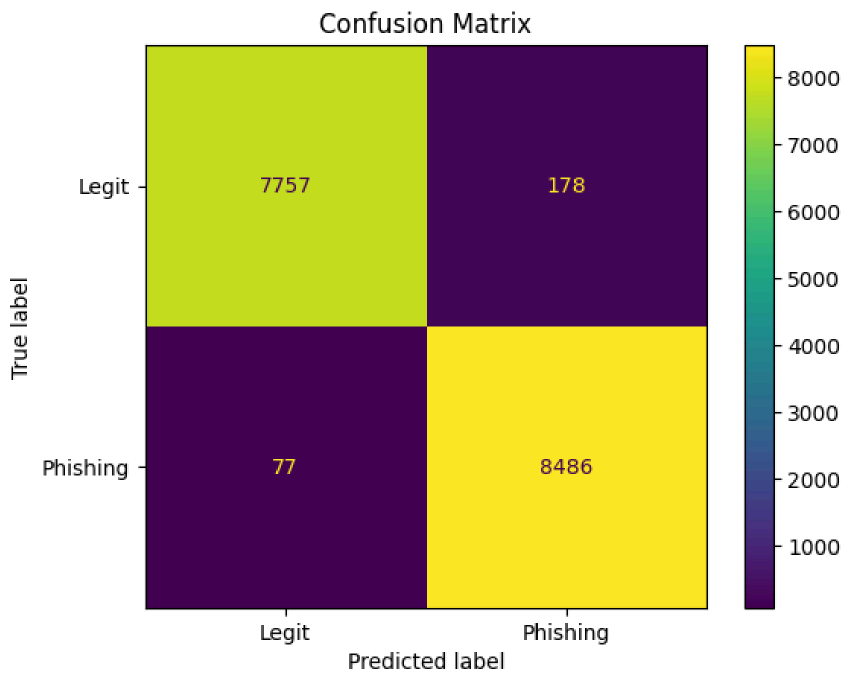


```
=== Classification Report ===
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      7935
           1       0.98      0.99      0.99      8563

    accuracy                           0.98     16498
   macro avg       0.98      0.98      0.98     16498
weighted avg       0.98      0.98      0.98     16498
```

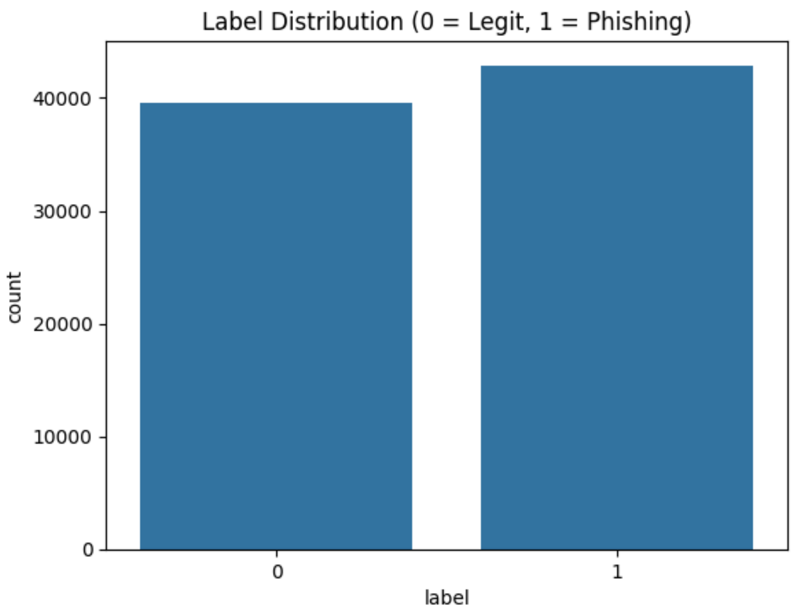


The confusion matrix further illustrated these outcomes. The model correctly identified 8,486 phishing emails and 7,757 legitimate emails. Only 178 legitimate emails were falsely flagged as phishing (false positives), and just 77 phishing emails were missed (false negatives)

This shows strong performance with very few misclassifications, especially high precision for phishing, which is essential to minimize disruption for users.



We also visualized the label distribution to confirm class balance in the dataset. Phishing emails made up about 52% of the total, while legitimate emails accounted for 48%. This balanced distribution supported fair training and evaluation of the classifier.



We omitted the email length histogram from the final analysis due to extreme outliers skewing the chart. These outliers did not affect model performance and were retained in training to preserve real-world diversity.

5. Discussion (20 points)

The results of our model demonstrate that phishing email detection using only textual content is both feasible and highly effective. Our logistic regression classifier achieved a 98% accuracy rate, with a near-perfect precision and recall for phishing emails. This means that when the model flags an email as phishing, it is almost always correct — a critical factor in minimizing false positives that could disrupt user experience.

One pattern we observed was that phishing emails tended to be shorter and more urgent in tone, while legitimate emails were more varied in length and often contained more detailed language. This reflects known phishing tactics that aim to provoke quick, impulsive actions from recipients.

However, the confusion matrix revealed that 77 phishing emails were misclassified as legitimate. These false negatives suggest that certain phishing messages were written to closely mimic legitimate communication, making them harder to detect based on text alone. This highlights an area for future improvement — potentially by using more complex models or incorporating additional features.

Currently, our model relies solely on email text. It does not analyze:

- Hyperlinks or embedded URLs
- Sender domain reputation
- Message timing or frequency
- Attachments or embedded media

Future work could include integrating these elements into the classification pipeline. We also plan to test more advanced models such as Random Forests, Support Vector Machines, or LSTM-based neural networks. Finally, we aim to explore deployment in real-time environments, such as a browser extension or email server integration, for live phishing detection.media

Future work could include integrating these elements or testing more complex models such as Random Forests, Support Vector Machines, or LSTM-based neural networks. We also plan to experiment with real-time classification in a deployed environment, like a browser extension or mail server integration.

6. Code Review (15 points)

Our code was written in Python using Google Colab, which allowed for easy collaboration, testing, and visualization. All results, tables, and plots presented in the report were generated directly from the code.

We organized our code into clear, well-commented sections:

- Data loading
- Exploratory data analysis
- Text preprocessing
- Feature extraction (TF-IDF)
- Model training and prediction
- Evaluation and visualization

Variable names were meaningful and consistent (clean_text, X_test, y_pred, etc.), and we avoided unnecessary or unused code. Comments were included in key sections such as the preprocessing function and visualization steps to clarify logic and reasoning. All visual outputs, including the confusion matrix and metric bar charts, were implemented using matplotlib and seaborn for readability.

The notebook is fully executable from top to bottom and generates the same results consistently.

7. Collaboration (5 points)

All group members collaborated closely throughout the project. We jointly selected the phishing email dataset from Kaggle, discussed the approach, and worked side-by-side in Google Colab to develop and test the code. Everyone contributed to preprocessing decisions, visualization design, and debugging the model. We also shared editing duties on the final PowerPoint slides and Word report, giving each other feedback and making revisions together. Although we occasionally divided tasks to save time, every major decision and deliverable was reviewed by all three members to ensure equal participation and understanding. The final product reflects a shared effort from start to finish.

8. Conclusion

Our project successfully demonstrated that phishing email detection using machine learning can be accurate, efficient, and explainable. By training a logistic regression model on a comprehensive dataset of over 80,000 labeled emails, we achieved strong results — including 98% accuracy and 0.99 precision for phishing detection.

The model correctly identified most phishing emails while avoiding false flags on legitimate messages. This balance makes the system practical for real-world application, such as in email filtering or security software.

Looking forward, we plan to:

- Integrate metadata features like sender address and embedded links
- Test advanced models like deep neural networks
- Deploy our model as a browser extension or part of a spam-filtering pipeline

Our work confirms that even basic machine learning models, when trained on diverse, high-quality data, can have significant impact in the fight against phishing.

9. References

Al-Subaiey, A., Al-Thani, M., Alam, N. A., Antora, K. F., Khandakar, A., & Zaman, S. A. U. (2024, May 19). Novel interpretable and robust web-based AI platform for phishing email detection. ArXiv. https://arxiv.org/abs/2405.11619

Kaggle. (n.d.). Phishing email dataset. https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset

Scikit-learn Developers. (2023). Scikit-learn: Machine learning in Python. https://scikit-learn.org

Loper, E., & Bird, S. (2002). NLTK: Natural language toolkit. https://www.nltk.org

Python Software Foundation. (2024). Python documentation. https://docs.python.org

Appendices (If Applicable)

- *Include supplementary materials such as additional figures, detailed tables, or extended code snippets.*