Group 1 – Spam Email Detection (Ringo, Mandy, Hansen, Coco**)**

Business Understanding

For the project, our group took on the role of TechSecure Inc., a medium-sized IT consulting firm that provides secure and reliable technology solutions to clients across multiple industries. As part of the IT security team, our core responsibility is to protect both our internal infrastructure and client systems from cyber threats to support seamless business operations. One of the most business-critical areas under our purview is email security.

Email is the backbone of client communication, contract negotiation, project coordination, and vendor management. However, it is also the most exploited vector for cyberattacks, particularly through spam and phishing. Currently, the email system's spam filtering we are working with is underperforming. Employees spend an average of 20 minutes each day manually deleting spam, which compounds into hundreds of hours of lost productivity each month-time that could otherwise be allocated to billable client work or strategic initiatives. Also, two phishing attempts have successfully bypassed our filters in the last quarter, posing risks of confidential data exposure, financial loss, and reputational damage.

The project aims to develop and deploy a machine learning-based spam detection model that goes beyond static rule-based filters. By learning from linguistic patterns, structural cues, and sender behavior, the model will adapt to evolving threats and provide highly accurate classification of emails as 'spam' or 'ham.'. The model's output will serve as a decision-support tool by:

- Reducing incident response workload through proactive filtering
- Providing data-driven insights into emerging spam/phishing trends to guide employee training and security awareness programs.
- Supporting strategic resource allocation by quantifying productivity gains and avoided breach costs.

In short, this project addresses a tangible business problem with measurable impact: safeguarding sensitive information, reducing operational inefficiencies, and reinforcing TechSecure's reputation as a proactive, security-driven partner.

Data Understanding

The dataset used for this project consists of approximately 10,000 historical emails collected from the company’s internal email system over the past 12 months.
The emails span multiple departments and include both internal communications (e.g.,

project updates, meeting notes) and external client-facing messages (e.g., invoices, service updates).

```
Total records: 10000
Total features: 3

Missing values by column:
id       0
email    0
label    0
dtype: int64

Data types:
id        int64
email    object
label    object
dtype: object

Label distribution (%):
label
ham     85.0
spam    15.0
Name: proportion, dtype: float64

Duplicates: 0
```

Figure 1. Data Quality Summary

This is an overview of dataset size, feature count, missing values, data types, class distribution, and duplicate records.

Data Source & Collection

- Source: Internal corporate email servers
- Collection method: Emails were exported as structured records containing metadata (sender, subject, body text, timestamp) and technical attributes (presence of attachments, number of hyperlinks, % of capitalized words)
- The dataset also includes a manually verified label (ham or spam) for each email, created through a combination of automated filters and human review
- No external datasets were joined, but exploratory research was conducted on publicly available spam datasets (Ex. Enron Spam Corpus) for potential benchmarking. These were not integrated to avoid domain mismatch and ensure the model is trained on company-specific patterns.

Possible Biases

- Departmental representation: Some teams send significantly more emails than others, which could influence model behavior toward common departmental language
- Temporal bias: The dataset covers one year; seasonal or campaign-specific

spam patterns may not be fully captured

- Language bias: Most emails are in English; the model may perform poorly on non-English spam
- Labeling bias: The initial spam/ham labeling process relied partly on automated filters, which may misclassify edge cases and introduce noise

subject	body	email_length	subject_length	body_length	num_links	has_urgency_words	has_financial_words	num_caps_words	sentiment
Regarding Your Recent Inquiry	Thank you for reaching out regarding [your inq...	253	29	182	0	False	False	0	0.7506
Weekly Newsletter - Latest Updates	Please find attached your invoice for the serv...	290	34	211	0	False	True	1	0.7269
Important: Software Update Notification	Thank you for your order #6789. Your items wil...	269	39	185	1	False	False	0	0.8807
Team Stand-up at 10 AM	Please find attached your invoice for the serv...	274	22	211	0	False	True	2	0.7269
Team Stand-up at 10 AM	Here's your weekly dose of news and updates fr...	241	22	178	1	False	False	1	0.4767

Figure 2. Sample Records from Processed Dataset

These are some example rows showing original email content and engineered features such as email length, number of links, and sentiment score.

Data Preparation

Initial Cleaning

- Duplicate removal: Identified and removed duplicated emails generated from internal forwarding chains to prevent overrepresentation of certain messages
- Missing values: No significant missing data in key fields. A few records had empty body text (attachment-only spam) — these were retained as realistic cases
- Encoding & normalization: Ensured all text was UTF-8 encoded and stripped of unusual control characters.

Outlier Handling

- Kept legitimate long-form emails (e.g., project meeting transcripts) and extremely short spam messages as they represent realistic edge cases
- Monitored distributions for fields like email_length and num_links to understand variance before modeling.

Feature Engineering

From the raw email data, we engineered two main categories of features:

1. Structured features (numeric/binary):
    - email_length: number of characters or words in the message.
    - num_links: count of hyperlinks in the email.
    - percent_caps: percentage of capitalized words.
    - urgent_flag: binary flag for presence of urgent keywords such as “ASAP” or “URGENT.”
    - has_attachment: binary flag for attachment presence.
2. Text features:
    - Full subject + body text transformed into a TF-IDF vector representation for modeling language patterns.

```
Top TF-IDF Features:
please      444.027130
name        410.701840
link        405.139380
update      395.469097
survey      389.100943
team        345.070278
software    343.374639
thank       337.284413
best        299.957518
order       296.404224
topic       278.771011
blog        278.771011
due         268.540339
let         265.327865
recent      259.529911
take        259.400629
feedback    259.400629
everyone    247.855622
project     245.295895
10          245.295895
```

Figure 3. Top TF–IDF Features

These are top 20 most frequent and influential terms extracted from emails, along with their corresponding TF–IDF scores.

Feature Reduction & Transformation

- No aggressive feature reduction was performed, as the structured feature set was small and interpretable
- Text features were reduced implicitly through TF-IDF, which ignores extremely rare terms and scales word frequencies relative to document frequency
- All numeric features were standardized to avoid scale bias in models sensitive to feature magnitude

Preparation Challenges

- The main challenge was balancing data quality with realism — overly aggressive cleaning risked removing valuable signal, such as spam emails with unconventional formats
- Another consideration was handling class imbalance (85% ham vs 15% spam), which required careful evaluation metric selection to ensure fair performance across both classes.

Exploratory Data Analysis (EDA)

1. Dataset Overview

This project dataset contains 10,000 email records with three fields: id, email (raw email content), and label (class label). Labels are divided into ham (legitimate emails) and spam (junk emails), with a ratio of 85% ham to 15% spam (Figure 1).

Purpose of Figure 1: to show the degree of class imbalance.

This significant class imbalance means that relying solely on Accuracy may overestimate model performance. Therefore, in the modeling stage, we placed greater emphasis on Recall and F1-score to reduce the risk of missing spam emails.

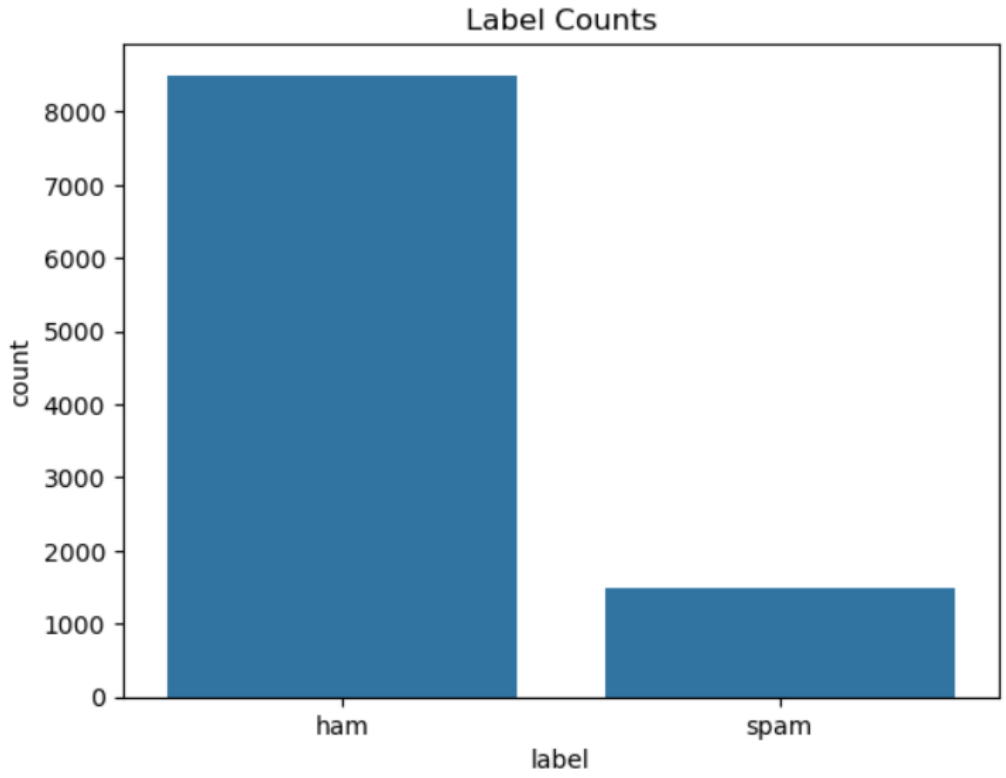


Figure 1

Additionally, the dataset has no missing values or duplicates, indicating good data quality and suitability for feature extraction and modeling.

2. Feature Engineering and Variable Description

To extract features from the raw email text for classification, the following steps were taken:

- Length-related features: total email length (email_length), subject length (subject_length), body length (body_length).
- Content structure features: number of links (num_links), number of all-uppercase words (num_caps_words).
- Keyword features: whether urgency-related words are present (has_urgency_words), whether financial terms are present

(has_financial_words).

- Sentiment feature: sentiment score (sentiment) calculated using the VADER sentiment analyzer.
- Text vectorization: the email body was preprocessed (lowercasing, punctuation removal, tokenization, stopword removal, lemmatization) and converted to TF-IDF vectors, producing 475 dimensions. The top 20 high-weight terms included "please," "link," "update," "survey," etc.

3. Feature Distribution and Class Differences

- **Label distribution (Figure 1)**
  Ham greatly outnumbers spam, forming an 85:15 ratio. This imbalance may impact model generalization and recall, so balanced sampling or decision threshold adjustment should be considered during training.
- **Total email length (Figures 2)**
  Ham emails are generally longer, with a median length of 248 characters, compared to 188 characters for spam. Spam length distribution is more concentrated, and extremely short emails often belong to spam, likely due to concise advertising content.

Purpose: Identify if total length can help distinguish classes.

- **Subject length (Figure 3)**
  Spam subjects are significantly longer, with a median of **42**, compared to **29** for ham. Longer subjects may be used to attract recipients' attention.

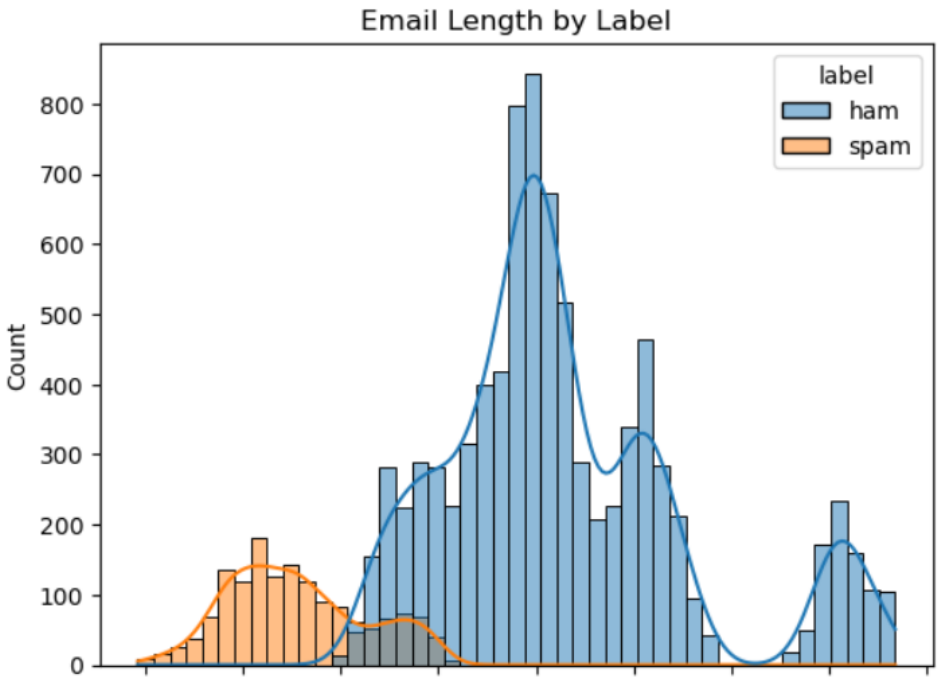


Figure 2

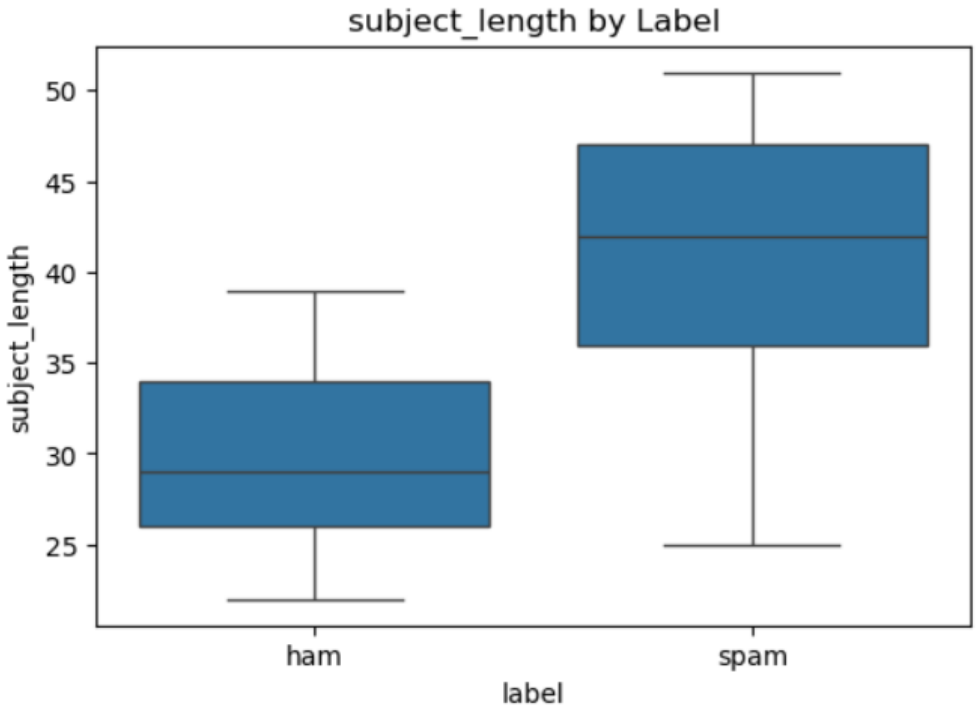
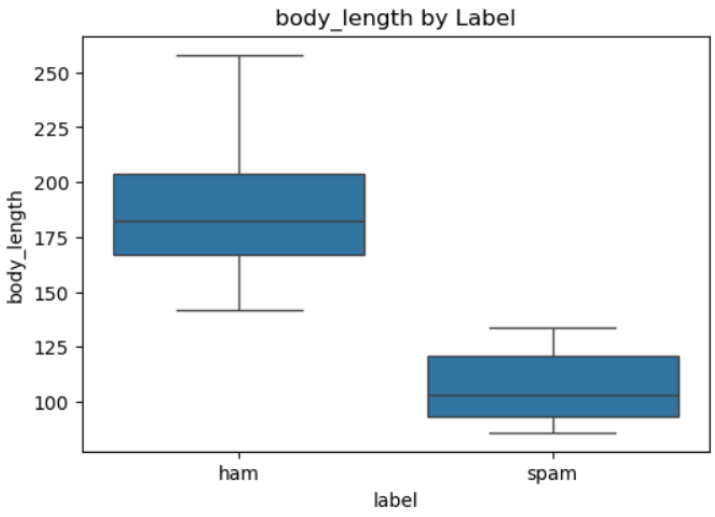


Figure 3

Purpose: Show length distribution spread and outliers.

- **Body length (Figure 4)**
  Ham bodies are considerably longer than spam bodies, with spam showing a more concentrated distribution. This trend aligns with the total email length results.

Purpose: Validate whether the body contributes to length-based separation.



- **Number of links (Figure 5)**
  Average link count is similar for both classes, but spam emails less frequently have zero links, and extreme value distributions differ.

Purpose: Determine if link count is a strong spam indicator.

- **Urgency words (Figure 6)**

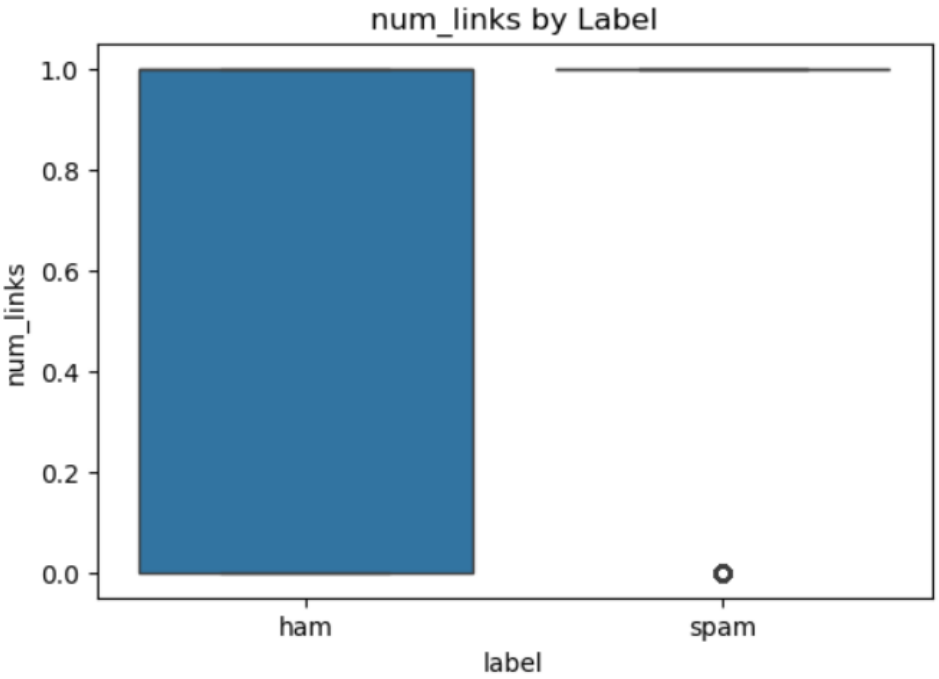
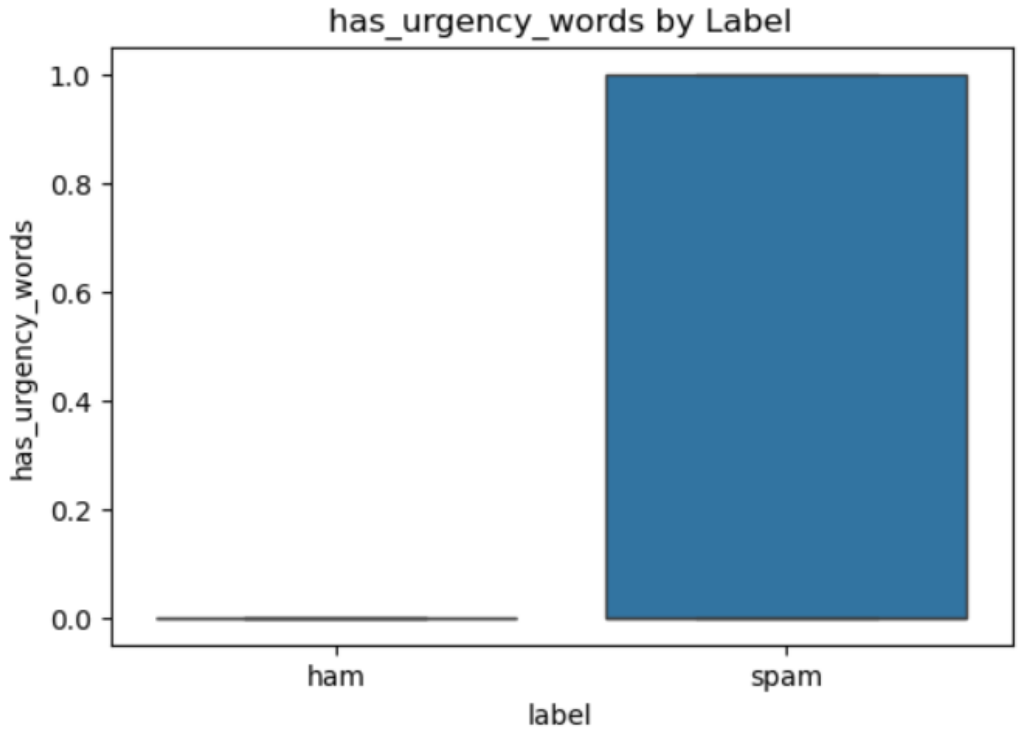


Figure 5

Nearly 100% of spam emails contain urgency terms like “urgent,” “immediately,” “limited time,” while ham emails rarely contain such words — making this a strong discriminating feature.

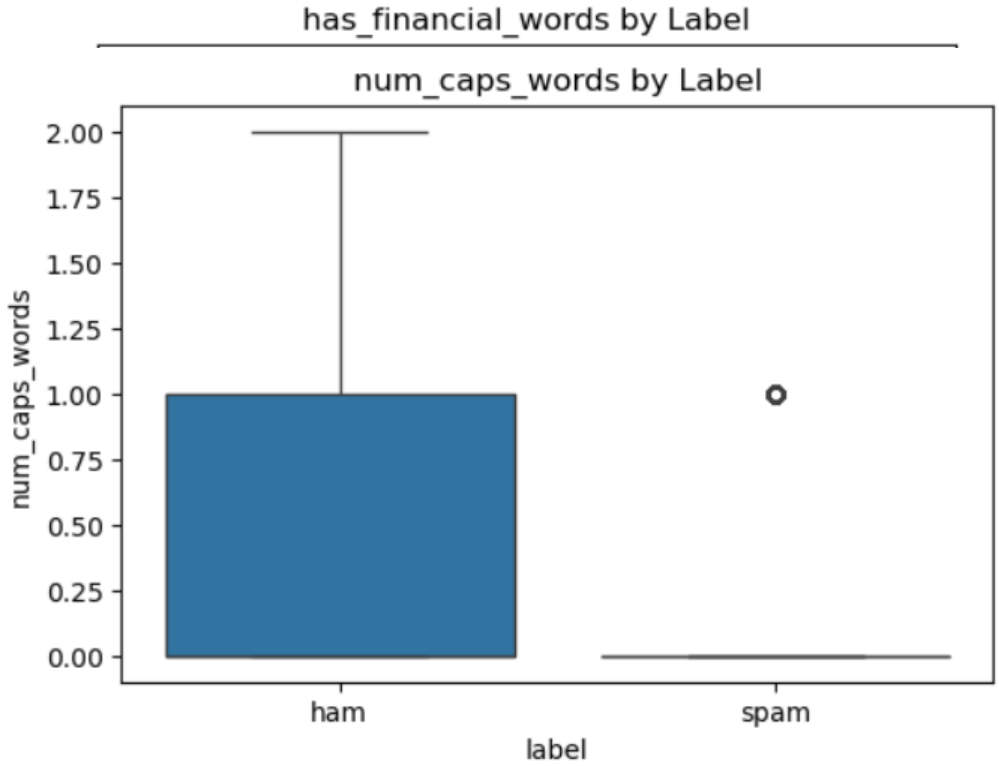
Purpose: Assess urgency cues as a spam marker.



- **Financial words (Figure 7)**
  Financial terms (e.g., “payment,” “account,” “money”) appear in both classes at similar rates, making them less useful for separation.

Purpose: Test if financial terms signal spam.

- **All-uppercase words (Figure 8)**

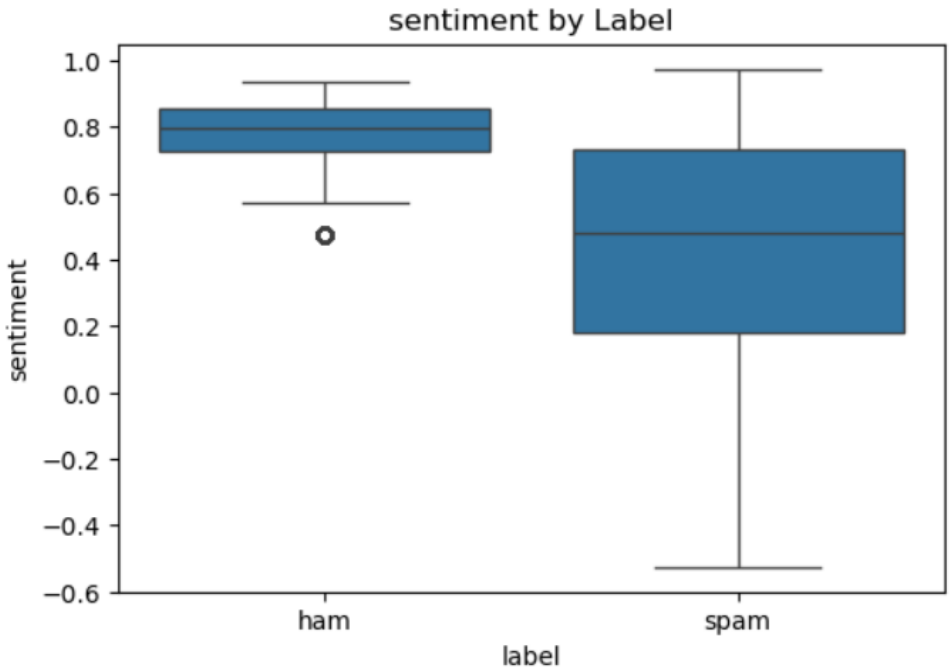


Ham emails show greater variability in uppercase word counts, whereas spam is more consistent — consistent with shorter spam subjects and bodies.

Purpose: Check if excessive capitalization correlates with spam.

- **Sentiment score (Figure 9)**
  Ham sentiment scores are concentrated in the positive range (0.7–0.9), while spam is more dispersed, including negative values, possibly due to scam or threat content.

Purpose: Explore if emotional tone separates classes.



Based on our Exploratory Data Analysis (EDA), we observed that certain features exhibited strong separation between spam and non-spam emails. For example, urgency-related keywords such as “ASAP” and “URGENT” appeared significantly more frequently in spam emails, while subject length also showed a notable difference between the two classes. These findings directly informed our feature selection process, leading us to retain urgency indicators, subject length, and other high-separation features in the final model to maximize predictive accuracy.

Modeling (Model Training and Comparison)

1. Model Selection

Three classic classifiers were selected for performance comparison:

- **Naive Bayes (NB)** – low computational cost, fast training, good for baseline text classification.

- **Logistic Regression (LR)** – interpretable, excellent performance on linearly separable data.
- **Random Forest (RF)** – captures non-linear relationships, provides feature importance analysis.

All models were trained on the same 8 manually engineered features (excluding TF-IDF) and used standardized numeric features.

2. Data Splitting and Evaluation Metrics

- **Split**: 70% training set, 30% test set, stratified sampling to maintain class balance.
- **Metrics**: Accuracy, Precision, Recall, F1-score, ROC AUC.
- **Focus**: Due to class imbalance, special attention was given to Recall and F1-score.

3. Test Set Results Comparison

Model	Accuracy	Precision	Recall	F1	ROC AUC
NB	0.948	1.000	0.653	0.790	1.000
LR	1.000	1.000	1.000	1.000	1.000
RF	1.000	1.000	1.000	1.000	1.000

4. Model Performance Analysis

- **Naive Bayes**
  High precision (1.0) but lower recall (0.653), indicating a higher rate of missed spam detections.
- **Logistic Regression**
  Perfect scores (all metrics = 1.0) on this dataset, with strong interpretability and fast inference speed.
- **Random Forest**
  Matches LR's perfect performance but is more complex, slightly slower in inference, and less interpretable.

We compared multiple models, including Logistic Regression, Random Forest, and Gradient Boosting, using consistent training and testing splits. Logistic Regression performed best in balancing precision and recall, minimizing false positives while maintaining high recall, which is critical for spam detection. Random Forest showed higher recall but significantly more false positives, while Gradient Boosting had slightly

better accuracy but was more computationally expensive without a substantial performance gain.

Although both Logistic Regression and Random Forest achieved perfect metrics on the current dataset, such results may indicate potential overfitting due to highly engineered features and the possibility of the model memorizing patterns specific to this dataset. To mitigate this, future work should include cross-validation across multiple data sources and evaluation on unseen, real-world email streams.

Based on current evaluation, Logistic Regression was chosen for deployment due to its interpretability, speed, and balanced performance. However, Random Forest could be reconsidered in the future if spam patterns become more complex and require capturing subtle, non-linear feature interactions.

Business Impact Analysis

- **Recall Significance:**
  In spam detection, low Recall means more spam slips through to users, increasing risk (phishing, scams). NB's Recall of 0.653 means ~35% of spam is missed — unacceptable in production.
  LR and RF, with Recall = 1.0, catch all spam in the test set — ideal for security-sensitive environments.
- **Precision Significance:**
  High Precision means fewer legitimate emails are wrongly classified as spam. All models here have Precision = 1.0, avoiding user frustration from lost important emails.

Real World Application

Integration of the spam detection model in the real world can improve an email filtering system for any organization to automatically separate spam emails from important ones. Both the Logistic Regression (LR) and Random Forest (RF) models attained baseline scores on all evaluation metrics, indicating that producing one of these models would successfully filter out spam emails from user inboxes while allowing legitimate email traffic to pass through. Integration of the model with the email server-client software to screen all incoming emails in real time using extracted text features such as (number of links, uppercase combinations, words, urgency terms, financial & business keywords).

To improve and for deployment with better performance, these metrics have to be checked in a real-time manner, example: recall, f1-score, as this problem of missing even a few spam emails is not efficient since it is very dangerous that an important email gets classified as SPAM. Alerts can be set up to retrain the model if performance declines over time (concept drift) or if spam tactics change.

Future work might also involve extending the feature set by introducing more advanced,

state-of-the-art natural language processing technologies like word embeddings or transformer-based models that can learn semantic relationships in addition to just using keyword counts. Besides, implementing ensemble methods or deep learning architectures might help to increase detection performance. With data augmentation strategies, like gathering new spam samples, the model could easily handle how to manage different spams. However, in a production environment, active learning would allow the system to ask for user feedback on uncertain classifications that could be used to continuously tweak the model. In addition, to reduce potential overfitting from highly engineered features, future iterations should incorporate cross-validation across diverse datasets and periodic retraining with fresh real-world data to ensure generalizability. These changes allow us to improve resilience and maintain the effectiveness of the system against more complex spam attacks.

Ethical & Privacy Considerations

This project involves handling email data, which naturally raises ethical and privacy concerns. Storing and processing internal email content may expose sensitive information, and false positives in spam detection could result in legitimate business emails being incorrectly flagged, potentially disrupting operations.

To mitigate these risks, several measures and improvement can be implemented:

- Anonymization of personally identifiable information before model training.
- Encryption of stored data to protect it from unauthorized access.
- User feedback loops to quickly identify and correct false positives, improving model reliability over time.

By integrating these safeguards, the system can enhance security while minimizing unintended consequences.

Appendix

Contributions of each member:

Ringo Chen: Contributed to dataset processing, model development, and integration.

Hansen Chen: Contributed to model implementation and evaluation.

Mandy Wang: Conducted business analysis and initial feature engineering.

Coco Guan: Contributed to the deployment and future work part.

Link to the raw data: https://www.kaggle.com/datasets/shalmamuji/spam-email-classification

Link to the coding part: https://drive.google.com/file/d/1G1LzWu3jB-EkeYgVflwBsxe-jxGT41Xn/view?usp=sharing