Ria Vellanki

Introduction to Problem

Spam emails, which range from unwanted ads to harmful phishing attempts, are a major challenge for both users and email providers. They threaten privacy and cybersecurity, making effective spam filtering crucial given the high volume of emails sent daily. This project focuses on creating models to classify emails as spam or non-spam using a dataset of email content and labels. By analyzing word frequencies, the project can answer key questions like: How accurately can word frequency alone classify an email as spam? Which words are the strongest indicators of spam or non-spam? The project is used to find the best threshold for spam classification to balance precision and recall. This offers insights into text-based filtering, keyword importance in spam detection, and strategies for setting thresholds for optimal model performance.

Introduction to Data

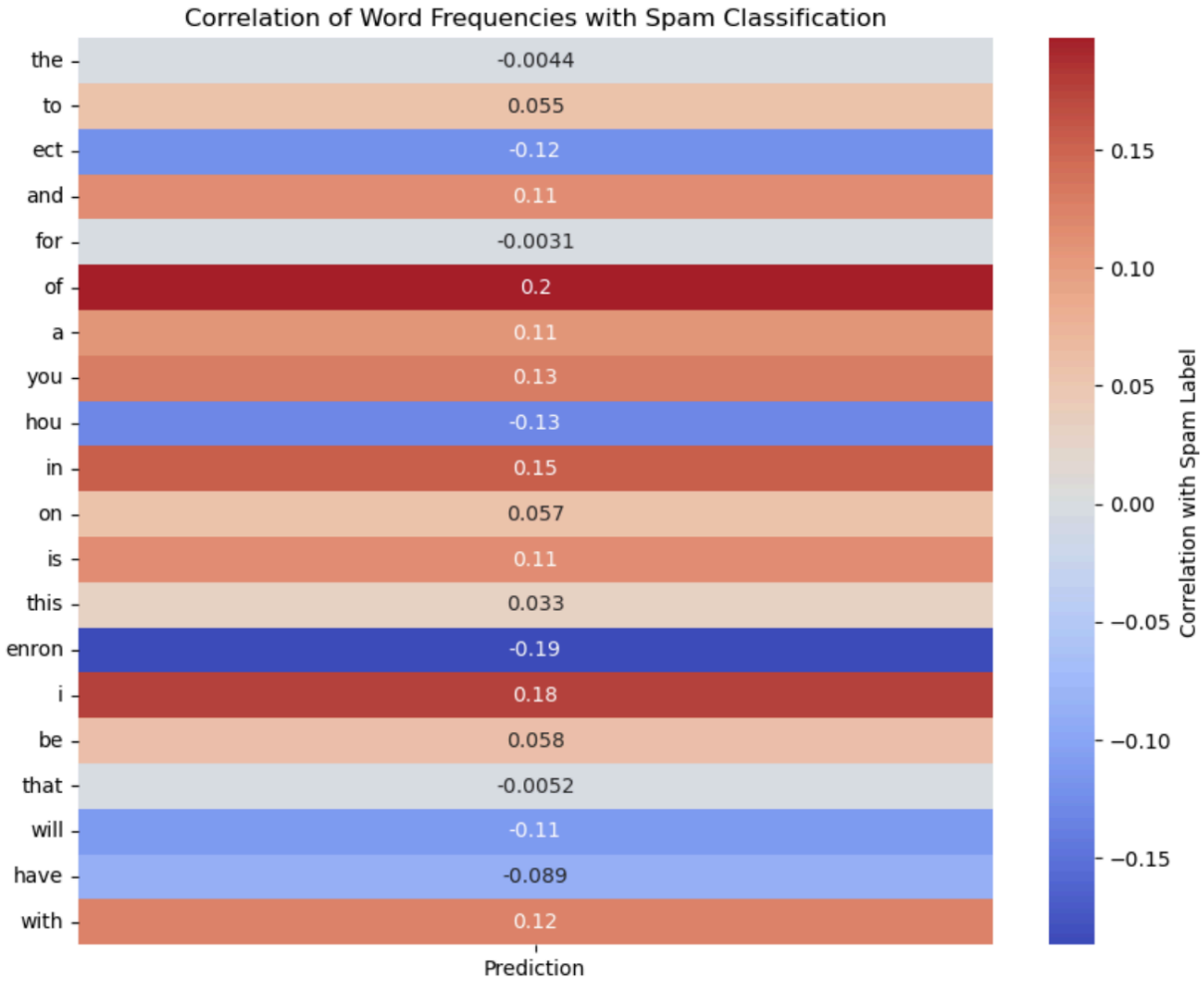
The dataset used for this project was an Email Spam Classification Dataset found on Kaggle at https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv. This dataset was designed for training models to classify emails as spam or non-spam based on word frequencies. It has 5,172 email records and 3002 columns, including an "Email No." column as an identifier as well as columns representing the frequency of common words like "the," "to," "and," and "you." These columns store integer values that indicate how often each word appears in each email. The "Prediction" column contains the binary label (spam or non-spam) for each email, allowing models to learn associations between word frequency patterns and spam classifications.

Pre-processing

I first created a DataFrame with the dataset and used “df.info()” to get an overview of its structure, identifying column data types and any initial issues. I checked for duplicate rows by running “num_duplicates = df.duplicated().sum()”, which confirmed there were no duplicates across the columns. I then looked for any missing values using “null_counts = df.isnull().sum()”, which showed that there were no null values in any columns. Since the dataset originally had 3,002 columns, I reduced its dimensionality to improve processing efficiency. I kept the first 21 columns, Email No. and other word frequency columns, and also the last column, “Prediction,” which indicates whether each email is spam or non-spam. I was able to put the first 21 columns and the last one together using “df = pd.concat([df.iloc[:, :21], df.iloc[:, -1]], axis=1)”,which resulted in a dataset with 22 columns for model training and analysis.

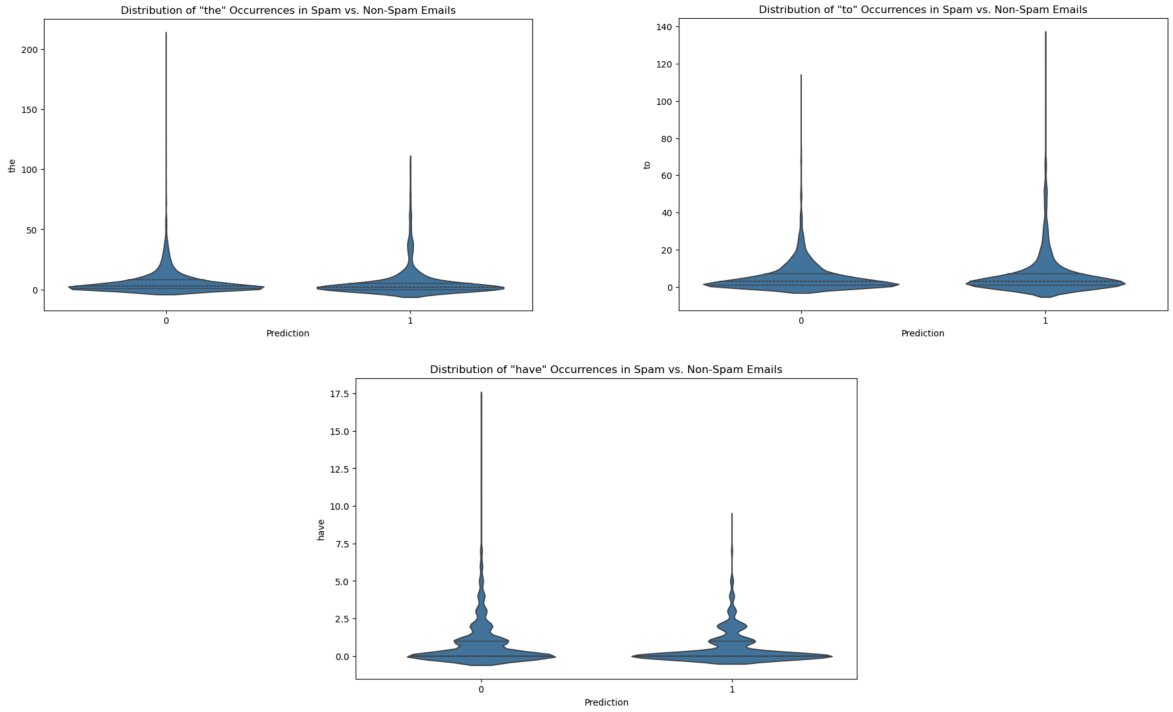
Data Understanding/Visualization

Ria Vellanki



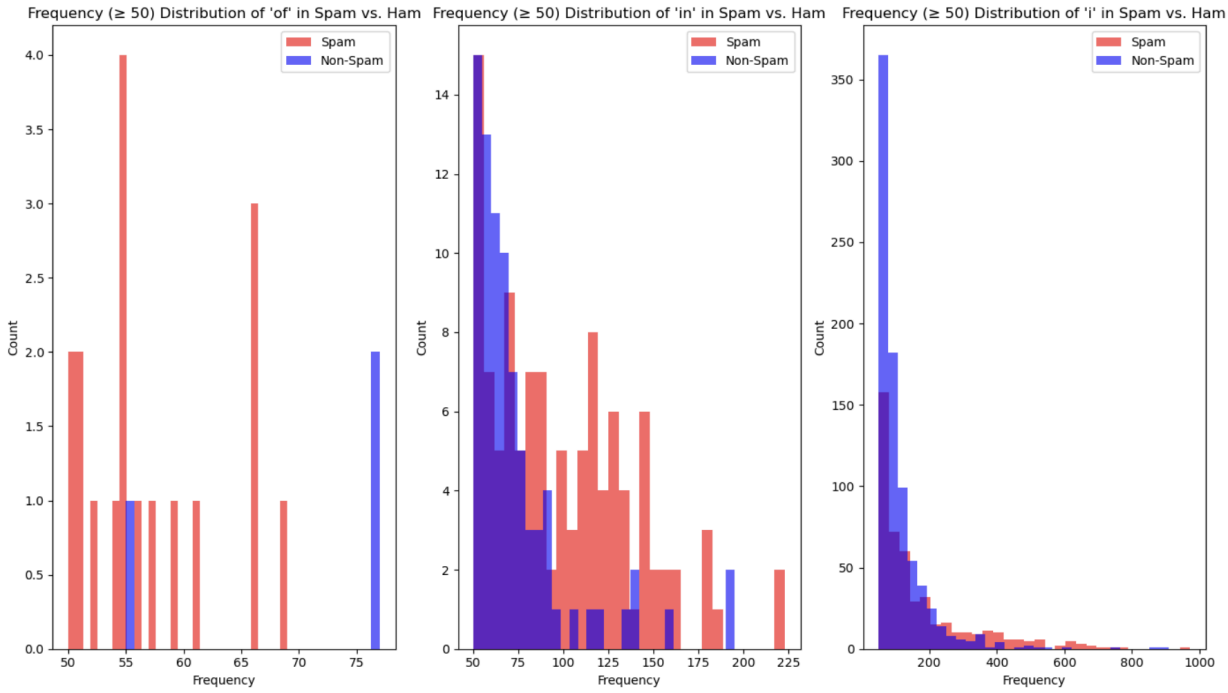
The heatmap visualizes the correlation between word frequencies in emails and their likelihood of being classified as spam, based on the "Prediction" column. Positive correlations indicate words more commonly found in spam emails, while negative correlations suggest words associated with non-spam emails. Words like "you" and "in" show modest positive correlations, implying a weaker association with spam but “i” and “of” have high correlations implying a stronger association with spam. The heatmap highlights strong indicators of spam (reds) versus non-spam (blues), giving a quick visual of word-level significance. This visual helped identify keywords for spam detection.

Ria Vellanki



The violin plots show the distribution of word occurrences in spam versus non-spam emails which emphasizes patterns in word usage. Common words like "to" are frequent in both categories, but spam emails often show longer tails, reflecting excessive repetition, while words like "have" are more often found in smaller quantities in spam. The plots also show significant skewness, with most emails having low word frequencies but occasional extreme values in spam. This visual confirms that excessive occurrences of certain words and distinct usage patterns are indicative of spam, requiring further analysis.

Ria Vellanki



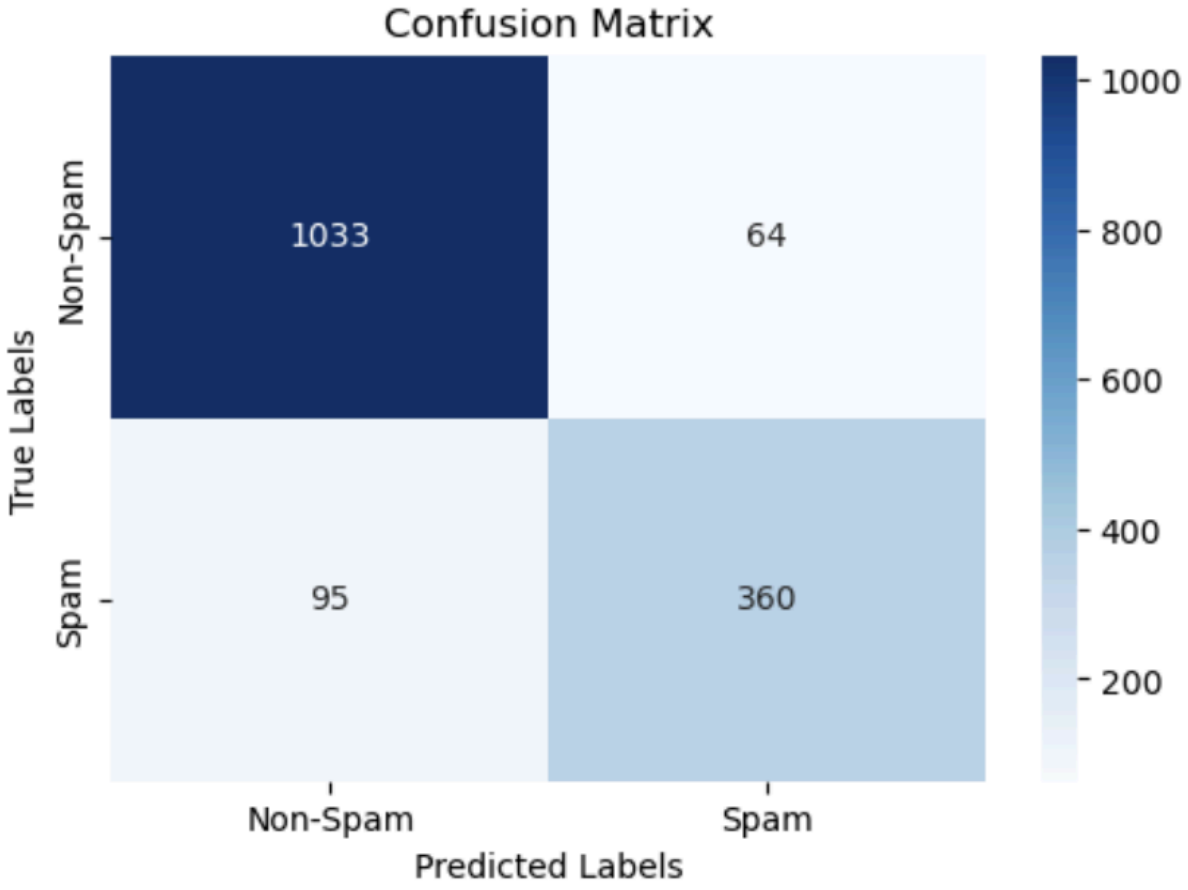
The visualization highlights the distribution of word frequencies in spam versus non-spam emails, focusing on frequencies of at least 50. Words like "of" are found more frequently in spam at higher frequencies, while "in" shows substantial distributions in both spam and not spam but has a lean toward spam. "i" is mostly found in non-spam emails at high frequencies, showing its strong correlation with non-spam emails. Words like "of" and "in" are strong spam predictors, while "i" suggests non-spam.

Model 1

I chose Random Forest Classifier because it is a good choice for classifying emails as spam or non-spam because it can handle complex data and find important patterns. It builds many decision trees on random parts of the data, which helps reduce overfitting and makes the model more accurate. Random Forest can also identify which words are most important for detecting spam. While it can be slow and might struggle with imbalanced data, these issues can be fixed using techniques like adjusting class weights or oversampling. Overall, it's a strong starting model for spam detection.

Ria Vellanki

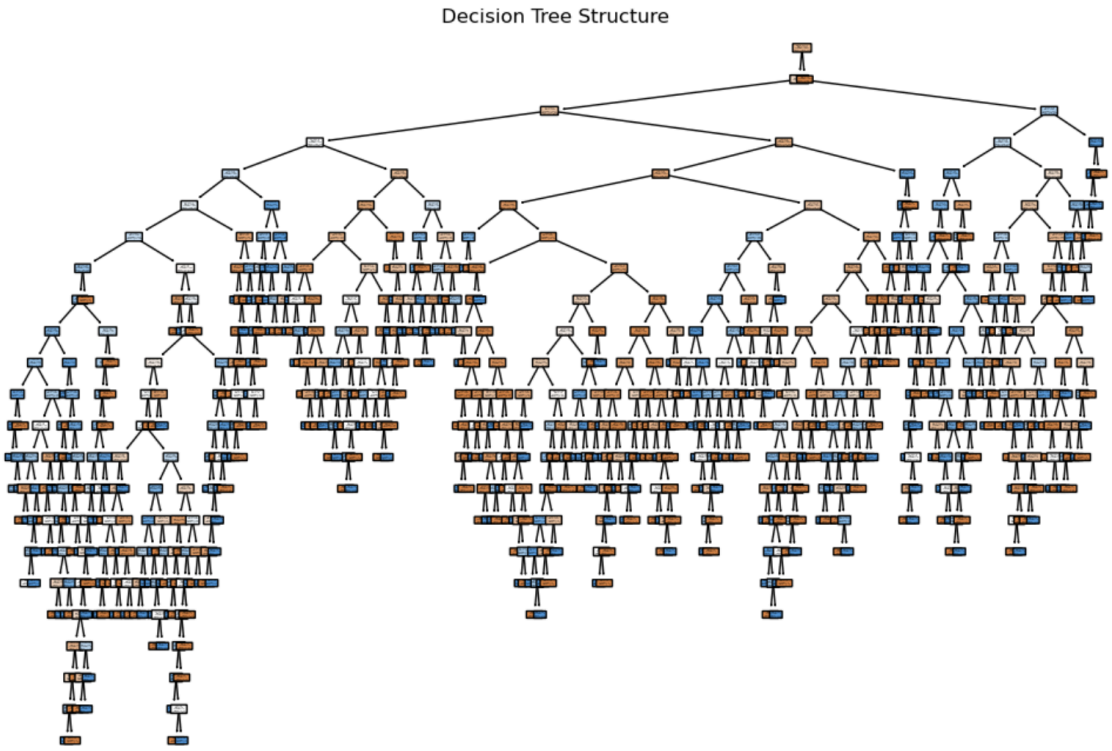
Evaluation 1



The confusion matrix provides a view of the model's classification performance by showing the number of correct and incorrect predictions for each class. The matrix shows that the model correctly classifies 1,033 non-spam emails and 360 spam emails. However, it misclassified 64 non-spam emails as spam and 95 spam emails as non-spam. This suggests that the model performs better at identifying non-spam emails than spam, which aligns with the precision, recall, and F1-scores. Using a confusion matrix as an evaluation metric is valuable because it highlights the distribution of errors across classes, giving a more well rounded view than accuracy alone. This information can help with improving the model through methods such as using a different threshold for classification or handling class imbalance to improve the detection of spam emails.

Ria Vellanki

Model 2



A Decision Tree is a machine learning model used for both classification and regression. It splits data into smaller parts based on feature values which creates a tree shaped structure. Each decision point in the tree is based on a feature and the leaves represent the final result, they are either a class label or a value. I chose it because Decision Trees are easy to understand and work with both numerical and categorical data. They can also capture complex relationships between features and the target.

Ria Vellanki

Evaluation 2

```
Accuracy: 0.8524484536082474

Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.89      0.89      1097
           1       0.74      0.77      0.75       455

    accuracy                           0.85      1552
   macro avg       0.82      0.83      0.82      1552
weighted avg       0.85      0.85      0.85      1552
```

The classification report shows an overall accuracy of 85.24% for the model's performance. The precision, recall, and F1-score metrics are calculated for two classes: 0 and 1. Class 0 has strong performance with a precision of 0.90, recall of 0.89, and an F1-score of 0.89, supported by 1,097 instances. Class 1 has a lower performance, with a precision of 0.74, recall of 0.77, and an F1-score of 0.75, supported by 455 instances. The macro average scores across both classes are 0.82 for precision, recall, and F1-score, highlighting a slight imbalance in class performance. The weighted average correlates closely with the overall accuracy, showing balanced model behavior influenced by the higher prevalence of Class 0.

Storytelling

This project uses data-driven models to solve the problem of spam email classification. Through data analysis, we found patterns in word usage that help tell spam from non-spam emails. Words like "of" and "you" were linked to spam, while "i" was more common in non-spam. Visualizations like heatmaps and violin plots showed how often certain words appeared, revealing that spam emails often repeat certain words.

In the modeling phase, Random Forests performed well with an accuracy of 89.8%, while Decision Trees were simpler and easier to understand, with an accuracy of 85.24%. The results showed that word frequency is a good way to tell spam from non-spam and helped identify key words for classification. However, it also highlighted the need to balance precision and recall to avoid misclassifying important emails as spam or letting spam slip through.

Impact

Ria Vellanki

This project has a big impact on society, as effective spam filtering helps improve productivity and protects users from phishing, malware, and fraud. Using machine learning to improve email filters contributes to better cybersecurity and a smoother online experience. However, there are ethical concerns. For example, models based on word frequency could accidentally filter out important emails because of how certain words are used in context. Additionally, spammers might adapt by using language similar to non-spam emails, requiring regular updates to the model. It's important to balance user convenience, privacy, and the risks of over-filtering, and to keep evaluating spam detection systems.