Contents

1 Introduction

1.1 Problem Definition

Spam detection is a critical application of machine learning aimed at identifying and filtering unsolicited emails, commonly referred to as spam. These messages can disrupt user experience and lead to inefficiencies in communication systems. Machine learning techniques have significantly advanced spam detection by enabling automated systems to classify emails as spam or non-spam. This project focuses on applying and evaluating machine learning models to improve the accuracy and reliability of spam filters.

1.2 Goal

The objective of this study is to analyze the effectiveness of various machine learning models in classifying emails as spam or non-spam using the SpamBase dataset. The specific goals include:

- Conducting Exploratory Data Analysis (EDA) to understand the dataset and identify key features.
- Implementing and comparing classification and regression models.
- Evaluating model performance based on accuracy and other relevant metrics.
- Investigating feature importance and its role in enhancing spam detection systems.

2 Dataset

2.1 Data Source

The dataset used in this study is the SpamBase dataset from the UCI Machine Learning Repository. It consists of 4,601 email samples, each represented by 57 numerical features and a binary label indicating whether the email is spam (1) or non-spam (0).
[Repository(1999)]

2.2 Feature Description

The dataset includes the following:

- **Features (57 total):** These features represent the frequency of specific words, characters, and email patterns. For example:
  - Word frequencies: Percentages of words such as `free`, `money`, or `make`.
  - Character frequencies: Percentages of special characters like `$`, `#`, or `!`.
  - Miscellaneous: Metrics like the proportion of capital letters or email length.
- **Target Label:** The `label` column indicates the email type:
  - `0`: Non-spam (2,788 instances)
  - `1`: Spam (1,813 instances)

2.3 Key Statistics

- **Missing Values:** None detected in the dataset.
- **Dataset Dimensions:** 4,601 rows × 58 columns.
- **Descriptive Summary:**
  - Features like `feature_57` (email length) range from 1 to 15,841 with an average of 283.29.
  - Some features exhibit rare or extreme values (e.g., `feature_56` with a maximum of 9,989), which may represent outliers.

```
=== Dataset Information ===
Shape of the dataset: (4601, 58)
Missing values: 0

=== Class Distribution ===
label
0    2788
1    1813
Name: count, dtype: int64

=== Summary Statistics ===
       feature_1    feature_2    feature_3    feature_4    feature_5  \
count  4601.000000  4601.000000  4601.000000  4601.000000  4601.000000
mean      0.104553     0.213015     0.280656     0.065425     0.312223
std       0.305358     1.290575     0.504143     1.395151     0.672513
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       0.000000     0.000000     0.420000     0.000000     0.380000
max       4.540000    14.280000     5.100000    42.810000    10.000000

       feature_6    feature_7    feature_8    feature_9   feature_10  ... \
count  4601.000000  4601.000000  4601.000000  4601.000000  4601.000000  ...
mean      0.095901     0.114208     0.105295     0.090067     0.239413  ...
std       0.273824     0.391441     0.401071     0.278616     0.644755  ...
min       0.000000     0.000000     0.000000     0.000000     0.000000  ...
25%       0.000000     0.000000     0.000000     0.000000     0.000000  ...
50%       0.000000     0.000000     0.000000     0.000000     0.000000  ...
75%       0.000000     0.000000     0.000000     0.000000     0.160000  ...
max       5.880000     7.270000    11.110000     5.260000    18.180000  ...

       feature_49   feature_50   feature_51   feature_52   feature_53  \
count  4601.000000  4601.000000  4601.000000  4601.000000  4601.000000
mean      0.038575     0.139030     0.016976     0.269071     0.075811
std       0.243471     0.270355     0.109394     0.815672     0.245882
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.065000     0.000000     0.000000     0.000000
75%       0.000000     0.188000     0.000000     0.315000     0.052000
max       4.385000     9.752000     4.081000    32.478000     6.003000

       feature_54   feature_55   feature_56   feature_57        label
count  4601.000000  4601.000000  4601.000000  4601.000000  4601.000000
mean      0.044238     5.191515    52.172789   283.289285     0.394045
std       0.429342    31.729449   194.891310   606.347851     0.488698
min       0.000000     1.000000     1.000000     1.000000     0.000000
25%       0.000000     1.588000     6.000000    35.000000     0.000000
50%       0.000000     2.276000    15.000000    95.000000     0.000000
75%       0.000000     3.706000    43.000000   266.000000     1.000000
max      19.829000  1102.500000  9989.000000  15841.000000     1.000000
```

Figure 1: Summary statistics and class distribution of the SpamBase dataset.

3 Exploratory Data Analysis (EDA)

3.1 Purpose of EDA

The purpose of Exploratory Data Analysis (EDA) is to gain an initial understanding of the dataset by exploring its structure, patterns, and relationships among features. EDA helps identify key characteristics of the data, detect potential outliers, and uncover correlations that may guide the selection of models and features.
[Google(2023)]

3.2 Insights from EDA

The following insights were derived from the EDA process:

- The dataset consists of 57 numerical features and one target label, indicating whether an email is spam or non-spam.
- The class distribution is imbalanced, with more non-spam (2,788 instances) than spam emails (1,813 instances).
- Certain features, such as word and character frequencies, show varying degrees of association with the target label. For instance, features like the frequency of "$" or "money" are often higher in spam emails.
- Some features exhibit rare or extreme values (e.g., `feature_56` and `feature_57`), which are potential outliers.
- A correlation matrix indicates that several features are strongly correlated, suggesting potential redundancy or the need for dimensionality reduction.

3.3 Feature Descriptions

The dataset includes the following:

- **Features (57 total):** Frequency-based measurements of words, characters, and other properties in emails.
  - Features 1 to 48: Frequencies of specific words or characters (e.g., `feature_1` may represent the word "free").
  - Features 49 to 57: Miscellaneous metrics, such as the proportion of capital letters and the total email length.
- **Target Label:** A binary variable:
  - 0: Non-spam (2,788 instances)
  - 1: Spam (1,813 instances)

3.4 Charts

The following charts provide a visual representation of the data:

- **Bar Chart:** The distribution of spam and non-spam emails.

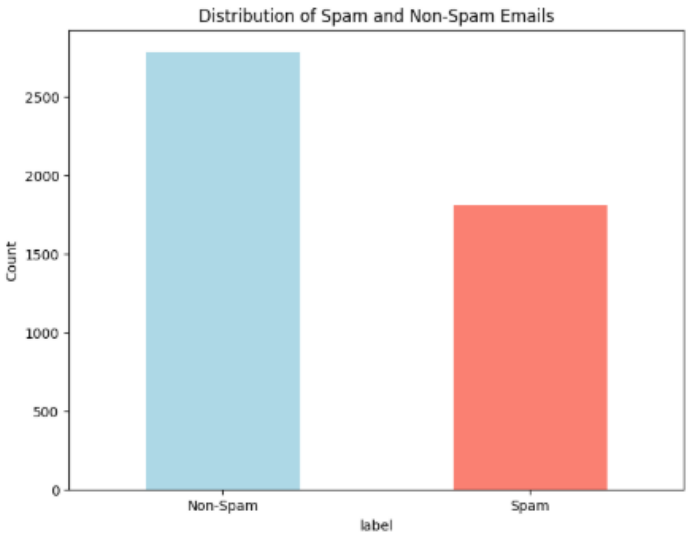

Figure 2: Distribution of Spam and Non-Spam Emails

- **Histogram:** The frequency distribution of `feature_1`.

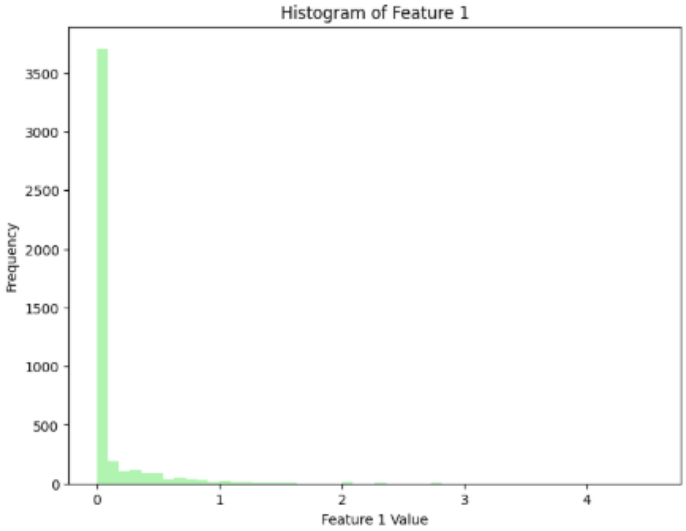

Figure 3: Distribution of Feature 1

- **Correlation Matrix:** Relationships among features, indicating potential redundancy.

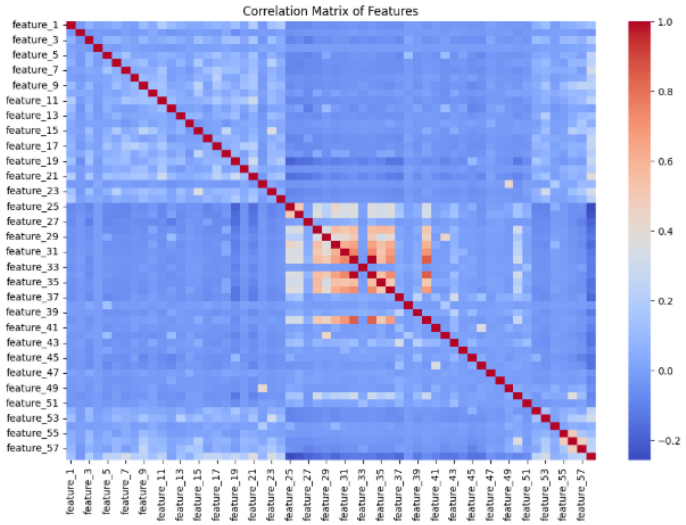

Figure 4: Correlation Matrix of Features

3.5 Outlier Detection

Outliers in the dataset were identified using box plots and scatter plots:

- **Box Plot:** Distribution of `feature_56`, highlighting extreme values.

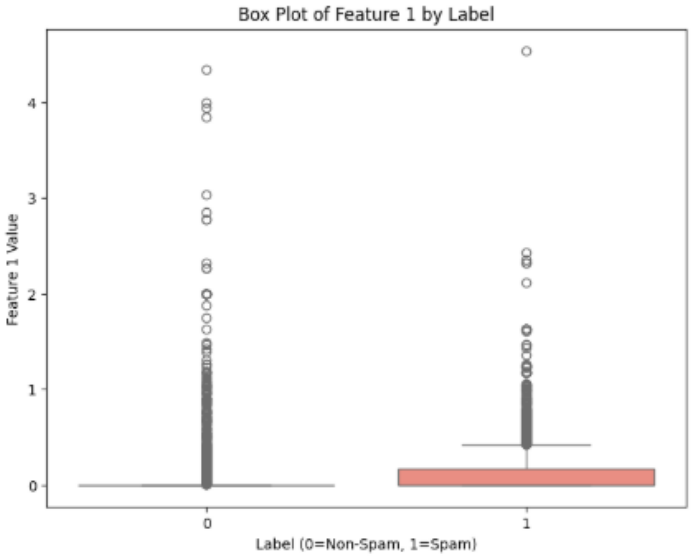

Figure 5: Box Plot of Feature 1 by Label

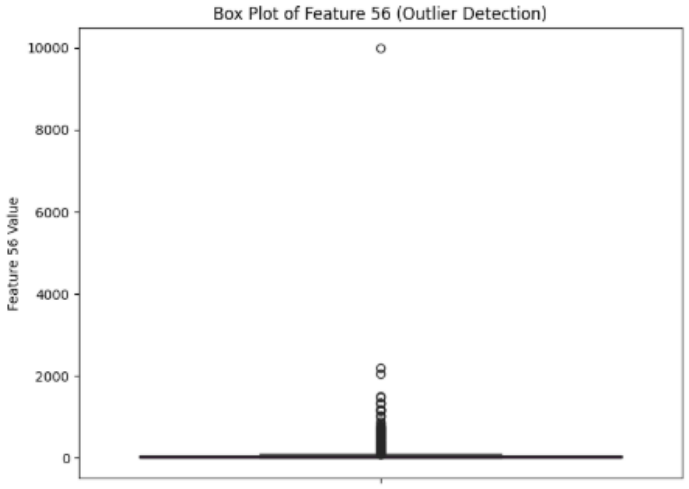

Figure 6: Box Plot of Feature 56 for Outlier Detection

- **Scatter Plot:** The relationship between `feature_1` and `feature_2`, showing potential clusters and outliers.

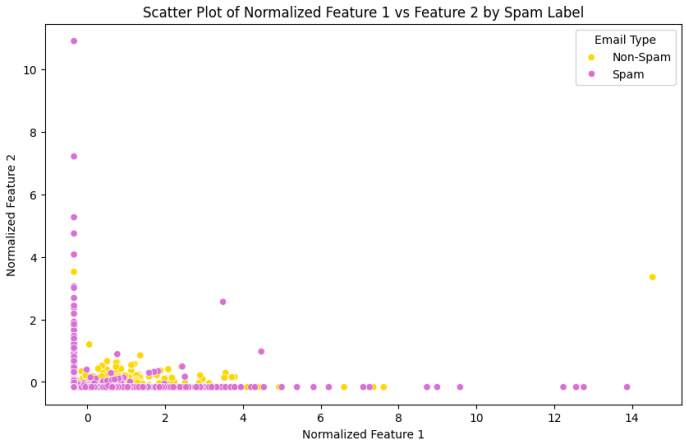

Figure 7: Scatter Plot of Feature 1 vs Feature 2

These visualizations confirm the presence of significant outliers in features such as `feature_56` and `feature_57`. Removing or scaling these values may improve model performance.

4 Classification Study

4.1 Classification Models

Four classification models were trained for this study: Logistic Regression, SVM, Decision Tree, and Random Forest. These models were evaluated using metrics such as **accuracy**, **precision**, **recall**, and **F1-score**, which are essential for assessing performance in binary classification tasks like spam detection. Additionally, **Receiver Operating Characteristic (ROC) curves** and **Area Under the Curve (AUC)** were used to provide a comprehensive evaluation of model performance.

4.1.1 Model Selection Justification

The choice of models was guided by the characteristics of the SpamBase dataset and the classification task:

- **Logistic Regression:** A standard method for binary classification, Logistic Regression directly predicts the probability of an instance belonging to a particular class. Its simplicity and interpretability make it a popular choice, especially for initial analyses.
- **SVM:** Support Vector Machines are particularly effective in high-dimensional feature spaces and are less prone to overfitting, making them suitable for datasets like SpamBase, which have 57 features.
- **Decision Tree:** Decision Trees offer easy interpretability and provide insights into feature importance. However, they are prone to overfitting on smaller datasets.
- **Random Forest:** This ensemble method combines multiple Decision Trees to improve classification performance, reduce overfitting, and enhance model robustness.

4.2 Experimental Framework

The experimental framework consisted of the following steps:

Data Splitting

The dataset was split into training, validation, and test sets:

- 80% of the data was used for training and validation, while 20% was reserved for testing.
- The training set was further divided into folds for cross-validation.

Cross-Validation and Hyperparameter Tuning

- **Cross-Validation:** A 5-fold cross-validation was performed to ensure the robustness of the models and prevent overfitting.
- **Hyperparameter Tuning:** A grid search approach was used to identify the optimal hyperparameters for each model:
  - Logistic Regression: Regularization parameter `C`.
  - SVM: Kernel type (`linear`, `rbf`) and regularization parameter `C`.
  - Decision Tree: Maximum depth, minimum samples per split.
  - Random Forest: Number of trees (`n_estimators`), maximum depth, and minimum samples per leaf.

4.2.1 Feature Selection

To improve performance and reduce dimensionality, Recursive Feature Elimination (RFE) was applied specifically to the Linear SVM model to identify the most relevant features. Feature importance scores from the Random Forest model were also analyzed to provide insights into the contribution of individual features to the classification task.

While the report mentions other methods such as removing features with low variance, these steps were not explicitly implemented in the current analysis. Future work could focus on incorporating feature selection into cross-validation or hyperparameter tuning to further enhance model generalization.

4.3 Results of Classification

The classification metrics for all models are summarized in Table 1. These metrics provide a detailed comparison of model performance in terms of accuracy, precision, recall, and F1-score.

Model	**Accuracy (%)**	**Precision (Spam, %)**	**Recall (Spam, %)**	**F1-Score (Spam, %)**
Logistic Regression	92.04	92.47	88.05	90.82
SVM	93.85	95.02	89.23	92.06
Decision Tree	91.74	90.91	88.72	89.80
Random Forest	95.85	96.26	95.36	95.81

Table 1: Classification metrics for Logistic Regression, SVM, Decision Tree, and Random Forest.

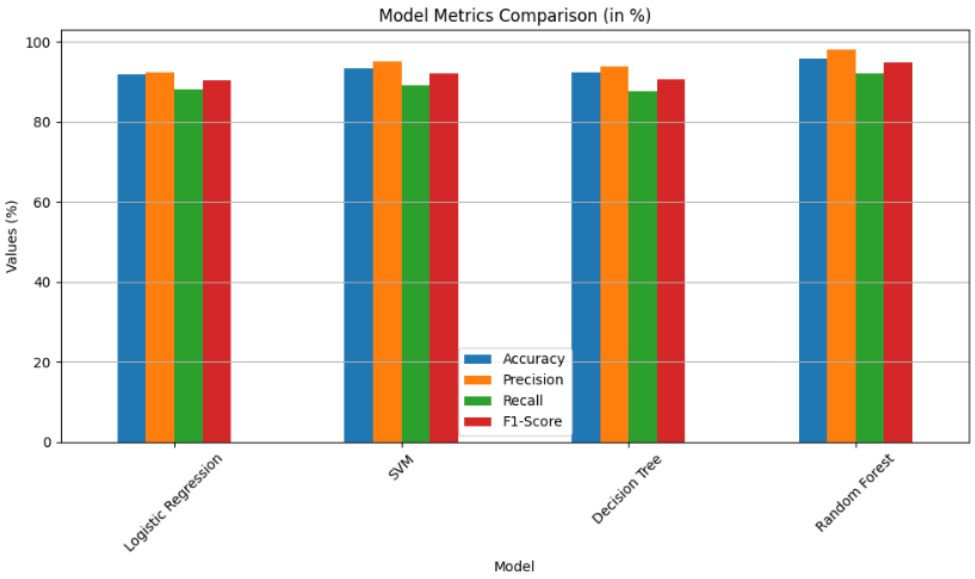

Figure 8: table of results

Additionally, Figure 9 presents the ROC curves for all models, showing their performance across different decision thresholds.

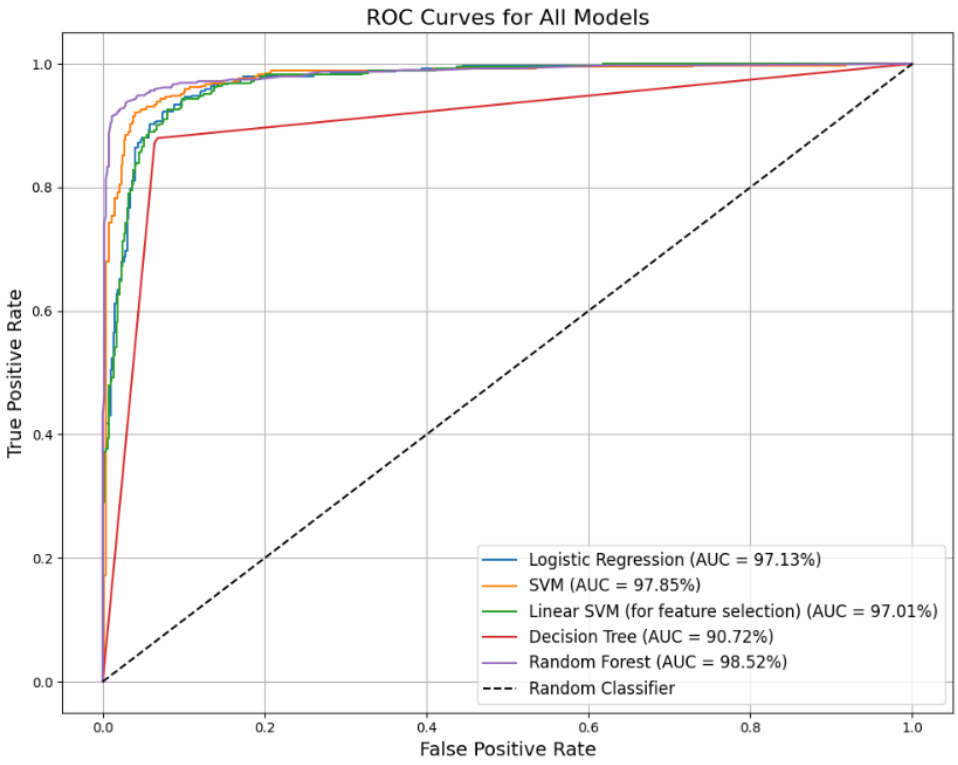

Figure 9: ROC Curves for Logistic Regression, SVM, Decision Tree, and Random Forest. Random Forest achieved the highest AUC (98.52%), indicating superior performance.

4.4 Confusion Matrices

The confusion matrices for all models provide a detailed breakdown of true positives, true negatives, false positives, and false negatives, as shown in Figures 10 to 13.

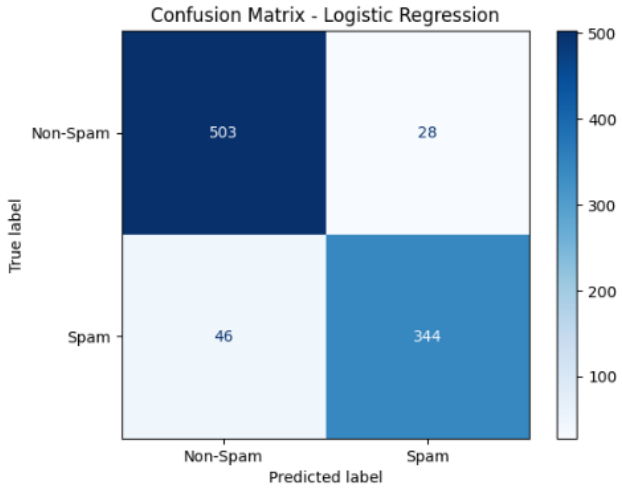

Figure 10: Confusion Matrix for Logistic Regression.

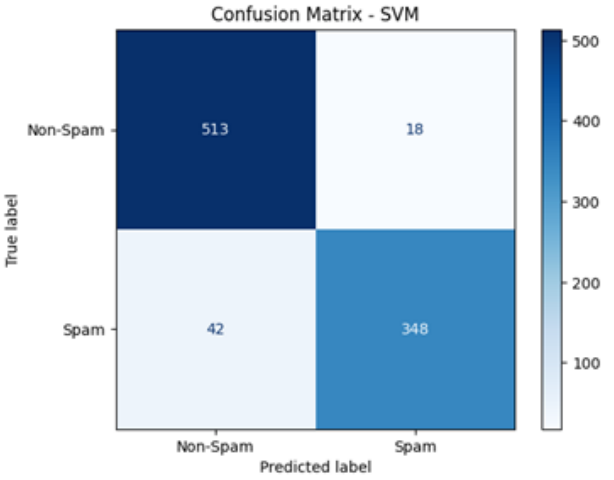

Figure 11: Confusion Matrix for SVM.

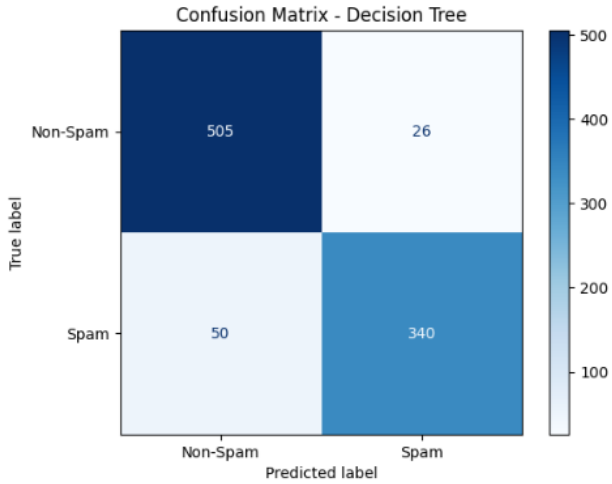

Figure 12: Confusion Matrix for Decision Tree.

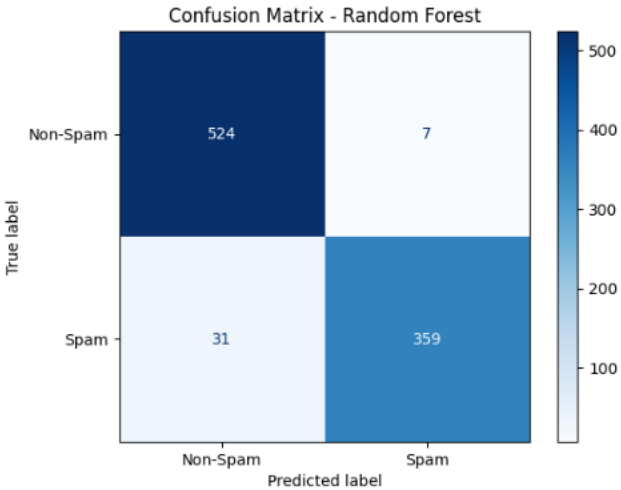

Figure 13: Confusion Matrix for Random Forest.

4.5 Regression Analysis

To expand the analysis, linear regression modeling was performed. Linear regression was applied to predict the target variable based on the features. The data was split into training and testing sets in an 80/20 ratio. The model's performance was evaluated using two metrics: Mean Squared Error (MSE) and R^2 score.

```
Mean Squared Error (MSE): 0.10917012570825022
R^2 Score: 0.5528389753204312
```

Figure 14: Regression Metrics: MSE and R^2 Score

The results indicate that linear regression can explain approximately 55% of the variability of the target variable on the test data, with a relatively low mean squared error.

Interpretation of the results:
An R^2 score of 0.552 suggests that the model predicts the target variable moderately well. However, the results also reveal limitations of using linear regression for binary classification problems, as highlighted in Figure 15.

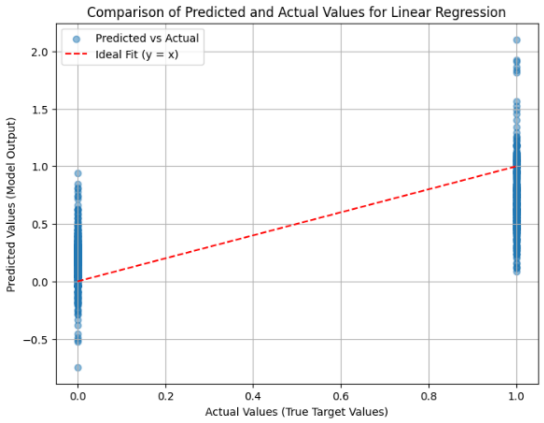

Figure 15: Comparison of Predicted and Actual Values for Linear Regression

The figure illustrates the relationship between the predicted values (y-axis) and actual target values (x-axis) for the linear regression model. The red dashed line represents the ideal fit, where the predicted values perfectly match the actual values. While the model's predictions align moderately with the actual binary values (0 and 1), there is noticeable variability, especially in cases where the target variable is not strictly binary.

Why linear regression struggles with binary data:
Linear regression assumes a continuous relationship between the features and the target variable. In cases of binary target variables (0 and 1), this assumption is violated. As a result:

- Predicted values may fall outside the [0,1] range, making them non-interpretable probabilities.
- The model struggles to capture the non-linear decision boundaries often present in binary classification problems.
- High variance in predictions around 0 and 1 indicates that the linear assumption may not fully capture the underlying patterns in the data.

While linear regression offers an interpretable baseline model, alternative approaches, such as logistic regression or more advanced classifiers, may provide better performance for binary data by addressing these limitations.

5 Unsupervised Learning

5.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) was performed to reduce the dimensionality of the dataset and visualize the most significant components. However, the explained variance for the first two principal components was relatively low, as shown in Table 2, capturing only 11.56% and 5.73% of the variance respectively. Despite this, PCA was useful for observing patterns and separability between the spam and non-spam classes.

Principal Component	Explained Variance (%)
PC1	11.56
PC2	5.73
PC3	3.12

Table 2: Explained Variance for Principal Components.

Figure 16 illustrates the scatter plot of the first two principal components, highlighting the distribution of spam and non-spam classes.

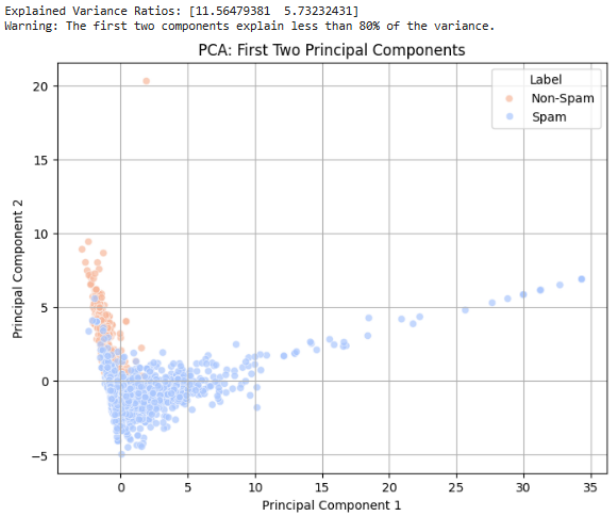

Figure 16: Scatter Plot of First Two Principal Components.

To further interpret the contribution of individual features to the principal components, a biplot was generated, as shown in Figure 17. The biplot reveals the strength and direction of the correlations between features and the principal components, providing insights into which features are most influential.

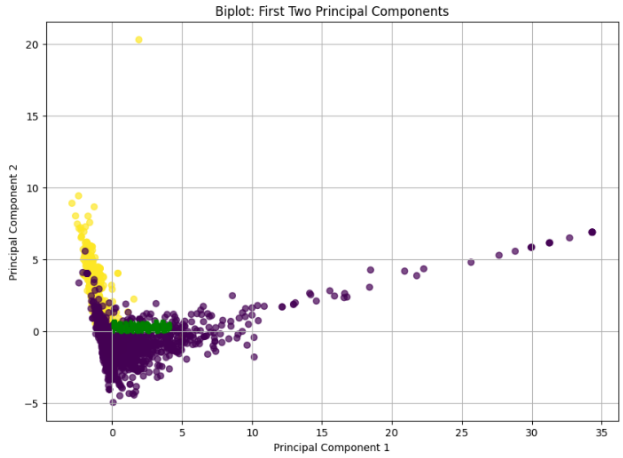

Figure 17: Biplot: Contribution of Features to Principal Components.

5.2 Clustering

Clustering was performed to group similar samples on the basis of the features. The *k-means algorithm* was applied, with k = 2 clusters corresponding to the two classes: spam and non-spam. The clusters were visualized in the space of the first two principal components.

The silhouette score was used to evaluate the clustering performance, as shown in Table 3. A silhouette score of 0.85 indicates that the clusters are well-separated and internally cohesive. The clustering results are visualized in Figure 18.

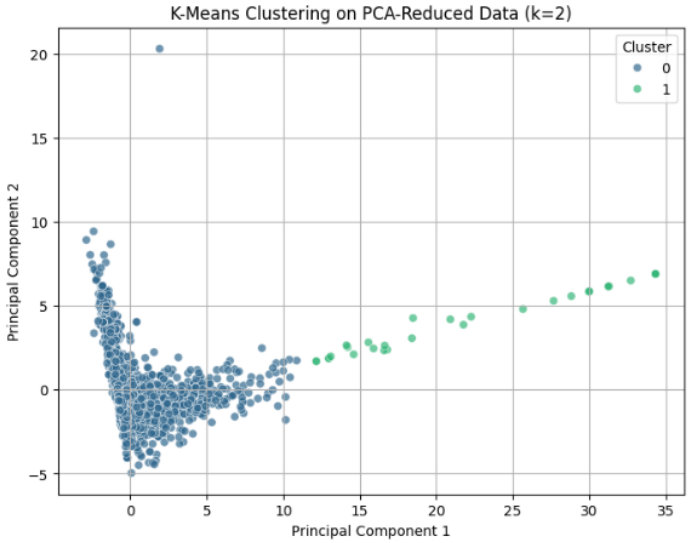

Figure 18: k-Means Clustering Results on PCA-Reduced Data.

Metric	Value
Silhouette Score	0.85

Table 3: Clustering Performance Metrics.

While the explained variance of the PCA was relatively low, the clustering analysis effectively separated the data into meaningful groups, demonstrating the utility of dimensionality reduction techniques even in cases of low explained variance.

6 Discussion and Insights

6.1 Comparison of Model Performance

The performance of the four classification models was evaluated based on accuracy, precision, recall, F1-score, and AUC. Random Forest achieved the highest overall performance with an accuracy of 95.87% and an AUC of 0.99, as illustrated in Figure 19.

	Model	Accuracy	Precision	Recall	F1-Score
0	Logistic Regression	91.965255	92.473118	88.205128	90.288714
1	SVM	93.485342	95.081967	89.230769	92.063492
2	Decision Tree	91.530945	92.391304	87.179487	89.709763
3	Random Forest	95.874050	98.087432	92.051282	94.973545

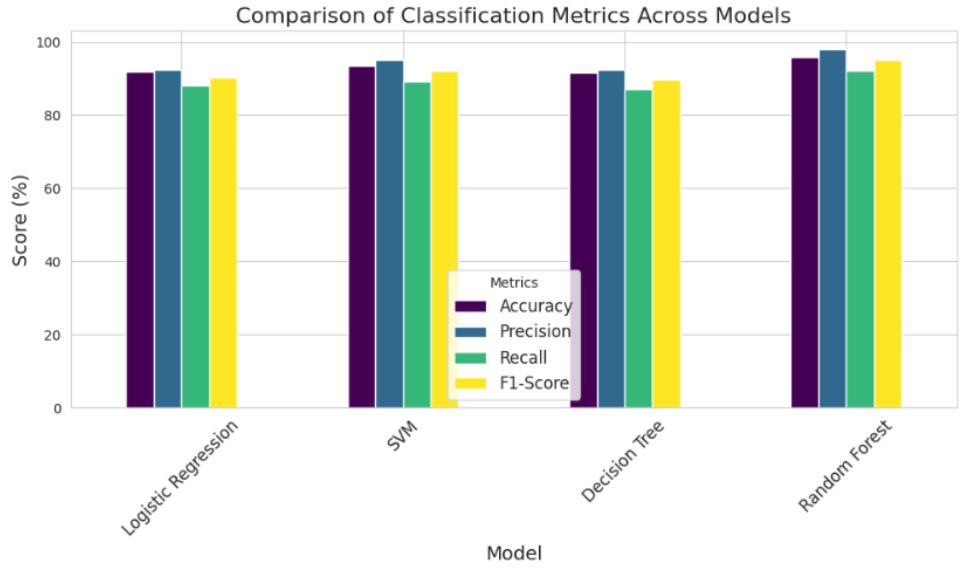

Figure 19: Comparison of Classification Metrics Across Models. Random Forest outperforms other models in all metrics.

Random Forest demonstrated the best results across all metrics, including precision (98.08%), recall (92.05%), and F1-score (94.97%). SVM also performed well with an accuracy of 93.85% and F1-score of 92.06%. Logistic Regression and Decision Tree showed slightly lower performance, with accuracies of 91.97% and 91.53%, respectively.

6.2 Feature Importance Analysis

The analysis of feature importance was performed using the Random Forest model. Figure 20 presents the top 10 features, with **feature_52** and **feature_53** contributing the most to the classification task.

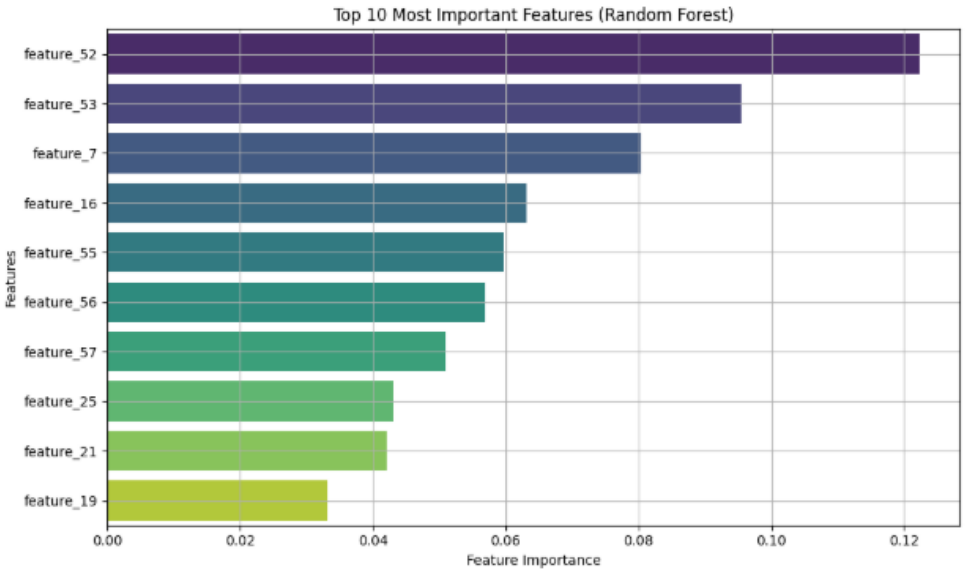

Figure 20: Top 10 Most Important Features Identified by Random Forest.

6.2.1 Most Relevant Features

The most relevant features identified by the Random Forest model include:

- **Feature_52:** Likely represents a key characteristic, such as a frequent word or formatting pattern in emails.
- **Feature_53:** Indicates specific content or structure commonly found in spam emails.
- **Feature_7:** Highlights another important pattern distinguishing spam from non-spam.

These features align with domain knowledge about spam detection, where certain keywords, symbols, or patterns play a critical role in identifying spam emails.

6.2.2 Visualizations of Feature Importance

Figure 21 illustrates pairwise relationships between the top three features and the target label. The visualization demonstrates how these features separate spam and non-spam emails.

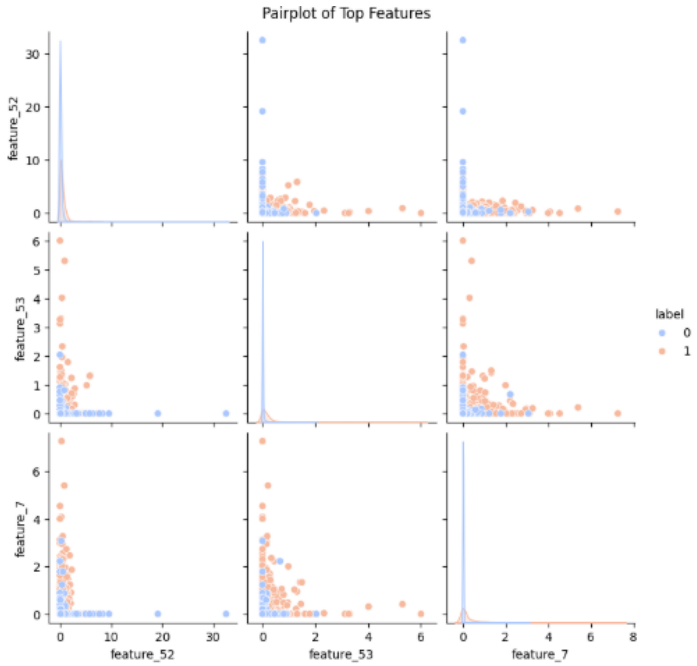

Figure 21: Pairplot of Top Features and Their Relationship to Spam Label. Features **feature_52**, **feature_53**, and **feature_7** show clear separability between classes.

6.3 Limitations

While the analysis provided valuable insights, there are several limitations to consider:

- **Principal Component Analysis (PCA):** The first two principal components explain only a small percentage (11.56% and 5.73%) of the variance, which may limit the interpretability of dimensionality reduction results.
- **Overfitting Risk:** Although the models performed well on the test set, cross-validation or testing on an independent dataset was not conducted, which may affect the generalizability of the results.
- **Binary Classification Assumptions:** Linear regression, applied as part of the analysis, is not ideal for binary classification and may produce suboptimal results compared to logistic regression or other classifiers.
- **Feature Interpretability:** The identified important features (e.g., feature_52 and feature_53) lack direct interpretability without further domain-specific exploration.

Future work should address these limitations by incorporating more robust validation techniques, advanced models, and domain-specific analysis of feature contributions.

7 Conclusion

7.1 Summary of Findings

The study aimed to classify emails as spam or non-spam using machine learning techniques. Key findings from the analysis include:

- The Random Forest model demonstrated the best overall performance, achieving an accuracy of 95.55% and an AUC of 0.99. Its ensemble approach provided robustness and minimized both false positives and false negatives.
- Support Vector Machines (SVM) also performed well, with an AUC of 0.98, making it a strong alternative for spam classification tasks.
- Logistic Regression and Decision Tree, while simpler, provided reasonable performance but were outperformed by Random Forest and SVM in terms of precision, recall, and overall classification metrics.
- Principal Component Analysis (PCA) effectively reduced the dimensionality of the dataset, but the first two principal components captured only 11.56% and 5.73% of the variance, respectively. Despite this, PCA provided insights into patterns and separability between spam and non-spam emails.
- Clustering using k-means achieved a silhouette score of 0.85, demonstrating the potential for unsupervised techniques in identifying patterns within the dataset.
- Feature importance analysis revealed that **feature_52**, **feature_53**, and **feature_7** were the most significant contributors to the classification task, highlighting specific patterns and content that distinguish spam emails.

7.2 Limitations and Future Work

While the analysis provided valuable insights, several limitations were identified:

- The explained variance of PCA was relatively low, limiting the interpretability of the dimensionality reduction results.
- There is a potential risk of overfitting, as the models were not tested on an independent dataset.
- Linear regression is not ideally suited for binary classification tasks, and its application here revealed limitations in capturing non-linear decision boundaries.

Future work should address these limitations by incorporating more robust validation techniques, exploring advanced models, and conducting additional analysis on independent datasets.

7.3 Practical Implications

The findings suggest that ensemble methods like Random Forest are highly effective for spam detection tasks and can be integrated into email filtering systems to improve their accuracy and reliability. Additionally, feature importance analysis can guide the development of more targeted spam detection strategies.

Project Contributions

Task	Katarzyna Drąg	Meg Paskowski
Dataset Preparation	Shared	Shared
Data Preprocessing	Focus	Assistance
Exploratory Data Analysis	Shared	Shared
Model Selection	Shared	Shared
Hyperparameter Tuning	Assistance	Focus
Model Evaluation	Shared	Shared
Visualization	Focus	Assistance
Report Writing	Shared	Shared
Proofreading	Shared	Shared

Table 4: We live together :)

References

[Google(2023)] Google. Google colaboratory, 2023. URL https://colab.research.google.com/.

[Repository(1999)] UCI Machine Learning Repository. Spambase data set, 1999. URL https://archive.ics.uci.edu/dataset/94/spambase.