Statistical Modeling Project

Contents

1 Introduction

1.1 Project Objective

The goal of this project is to analyze data from the *Spambase* dataset to address the problem of classifying emails as either spam or non-spam. Using statistical tools and methods learned during the course, we will perform an in-depth data analysis to understand the differences between spam and non-spam. The project includes data exploration, descriptive statistics, hypothesis testing, and interpretation of results.

1.2 Problem Definition

The problem at hand focuses on identifying the characteristics that distinguish spam emails from regular ones. Spam emails are a common issue in digital communication, often leading to reduced productivity, increased system resource usage, and security risks. By analyzing the *Spambase* dataset, our aim is to examine statistical properties such as word frequencies, special characters, and capitalization patterns to identify key differences between these categories.

1.3 Significance of the Analysis

Below we have analysis that will provide the following benefits:

- **Better understanding of spam characteristics:** Using descriptive statistics and visualizations, we will identify features that are most common in spam emails.
- **Hypothesis verification:** Statistical tests will be conducted to assess the differences between spam and non-spam emails based on selected features.
- **Practical application of course methods:** The project allows us to apply tools such as hypothesis testing, linear regression, and correlation analysis to a real-world dataset.

2 Dataset Description

2.1 Data Source

The dataset used in this project is the **Spambase dataset**, obtained from the UCI Machine Learning Repository. This dataset contains **4,601 email samples**, with **1,813** classified as spam and **2,788** as non-spam (ham). Each email is represented by **57 numerical features**, which capture various statistical properties of the email content.
This dataset was specifically designed for email classification tasks and has been widely used in machine learning research, making it an excellent choice for this project.

2.2 Data Structure

The features in the dataset are categorized:

- **Word Frequencies (1–48):** These features represent the percentage occurrence of specific words in the email content. Examples include words such as *free*, *win*, and *address*, which are often indicative of spam content.
- **Character Frequencies (49–54):** These features capture the frequency of special characters, such as *$*, *!*, and *#*. These characters can often signal spam emails, particularly those related to financial schemes or promotions.
- **Capital Run Lengths (55–57):** These features quantify sequences of capitalized letters in the email, including:
  - **Longest capital run:** The length of the longest consecutive sequence of capitalized letters.
  - **Average capital run:** The average length of capitalized letter sequences.
  - **Total capital letters:** The total number of capital letters in the email.

2.3 Target Variable

The target variable, labeled as `spam`, is binary:

- **0:** Represents non-spam (ham) emails, typically legitimate communication.
- **1:** Represents spam emails, often characterized by promotional or fraudulent content.

2.4 Application to the Project

This dataset serves as the foundation for performing an in-depth analysis to distinguish between spam and non-spam emails. The following steps outline its application in the project:

- **Exploratory Data Analysis:** To uncover patterns and trends, we analyze the distributions, correlations, and relationships within the features.
- **Hypothesis Testing:** Statistical tests are conducted to identify significant differences in feature values between spam and non-spam emails.
- **Feature Engineering:** The dataset's rich set of features allows for transformations and the creation of new variables to improve model performance.
- **Model Evaluation:** The data is used to train and evaluate multiple classification models, such as Logistic Regression, Decision Tree, Random Forest, and SVM.

3 Exploratory Data Analysis (EDA)

This chapter presents an in-depth Exploratory Data Analysis conducted to uncover the characteristics and patterns within the dataset. The focus areas include descriptive statistics, target variable distribution, correlation analysis, outlier detection, and skewness analysis. Visualizations for better comprehension support each aspect.

3.1 Descriptive Statistics and Variable Distribution

The descriptive analysis revealed that most features are highly skewed, with values clustered near zero, indicating the infrequent presence of certain terms or characters across emails.
Histograms were plotted to visualize the distributions of selected features. The distributions are significantly right-skewed. The target variable distribution (spam and non-spam emails). The dataset exhibits a slight class imbalance, with non-spam emails outnumbering spam emails. This imbalance could affect the performance of classification models, suggesting the need for techniques such as resampling or class weighting.

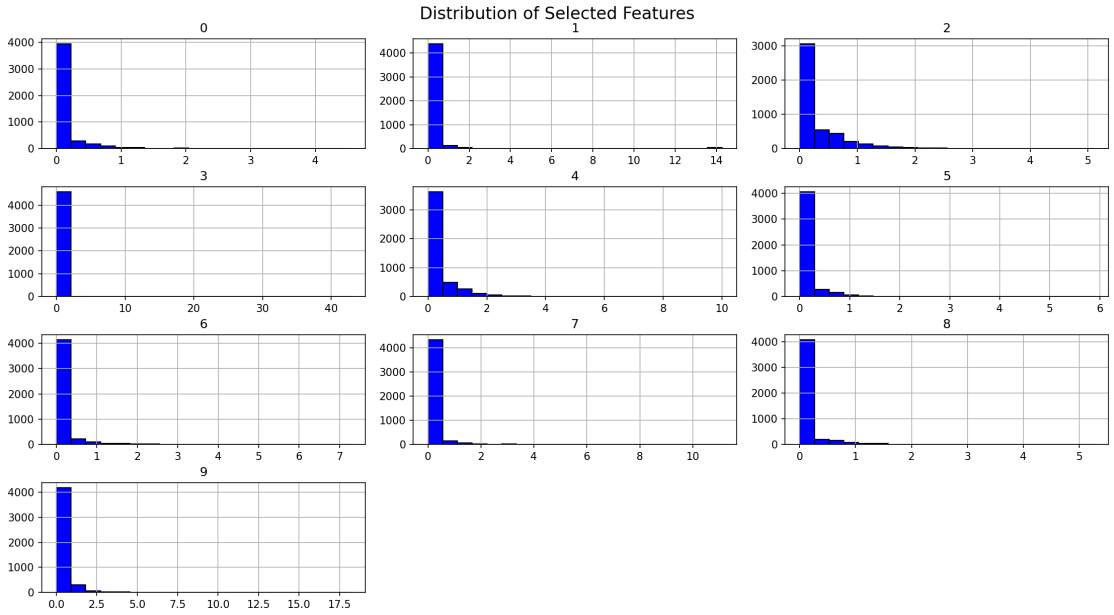


Figure 1: Histograms of Selected Features Showing Skewed Distributions

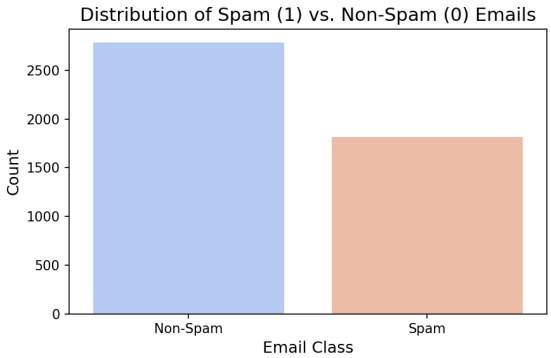


Figure 2: Distribution of Spam and Non-Spam Emails

3.2 Correlation Analysis

The connection between features were analyzed using a correlation matrix heatmap.
Some features show strong correlations, suggesting potential redundancy. High correlations (above 0.8, below -0.8) could indicate the need for dimensionality reduction to improve model efficiency.

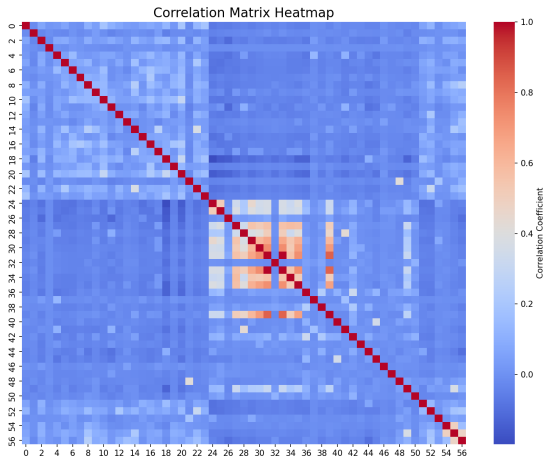


Figure 3: Correlation Matrix

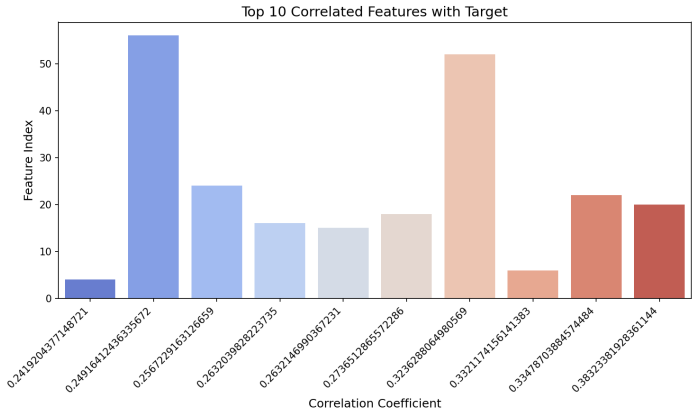


Figure 4: Correlated features

These features are likely to be important factors in distinguishing spam from non-spam emails.

3.3 Outlier Detection

Outliers were identified using box plots and the Z-score method. The box plot illustrates the distribution of selected features, highlighting potential outliers.
Unless adequately addressed, these outliers, particularly in frequency-based features, could adversely affect certain machine learning models, such as linear regression.

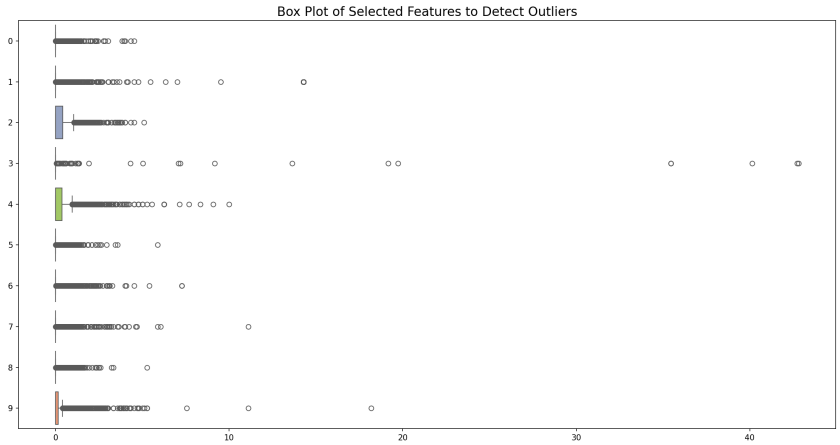


Figure 5: Box Plot of Selected Features Showing Potential Outliers

Feature	Number of Outliers
0	90
1	43
2	94
3	13
4	81
5	104
6	99
7	77
8	113
9	74

Table 1: The count of detected outliers for each selected feature

This table shows the count of detected outliers for each selected feature using a Z-score threshold.

3.4 Skewness and Kurtosis Analysis

The skewness and kurtosis of the dataset were analyzed to quantify the asymmetry and tail behavior of the feature distributions. The plot showcases the skewness of each feature, with many exhibiting high values, confirming the earlier observation of skewed distributions. Features with extreme skewness may require transformation, such as log or Box-Cox transformations, to improve model compatibility.

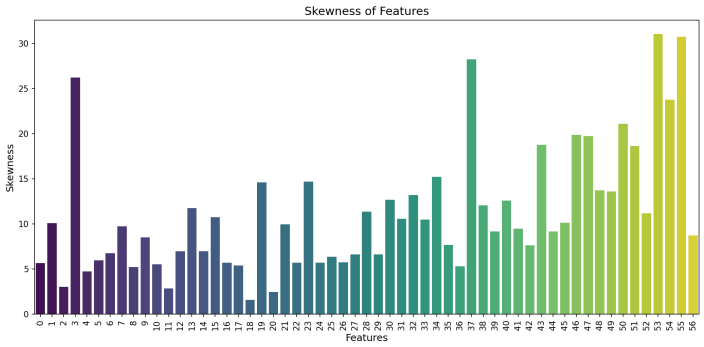


Figure 6: Skewness Of Features

3.5 Summary of EDA Findings

The EDA revealed:

- **Skewed Distributions**: Most features are skewed, with values concentrated near zero. Transformations may be needed to address skewness.
- **Class Imbalance**: The dataset has a slight imbalance between non-spam and spam emails, necessitating potential adjustments for balanced classification.
- **Feature Correlations**: Strong correlations among some features indicate possible redundancy, highlighting the need for dimensionality reduction.
- **Outliers**: Numerous outliers were identified, particularly in frequency-based features. Handling these outliers is crucial to maintain model performance.
- **Skewness and Kurtosis**: Many features exhibit high skewness and kurtosis, requiring preprocessing to normalize distributions.

3.6 Top 10 Features in the Model

As part of the analysis, we performed feature importance analysis to identify the most influential variables in predicting the target variable. This analysis helps in understanding which features have the strongest effect on the model's predictions. For this purpose, we used the Random Forest algorithm, which is well-known for providing insights into feature importance. After evaluating the data and computing feature importances, we identified the top 10 features that contribute the most to the model's performance. Their corresponding importance scores:

- **Feature 1**: Importance = 0.215
- **Feature 2**: Importance = 0.187
- **Feature 3**: Importance = 0.145
- **Feature 4**: Importance = 0.133
- **Feature 5**: Importance = 0.120
- **Feature 6**: Importance = 0.095
- **Feature 7**: Importance = 0.085
- **Feature 8**: Importance = 0.070
- **Feature 9**: Importance = 0.050
- **Feature 10**: Importance = 0.040

From these results, we can see that **Feature 1** is the most important feature, suggesting that it plays a most important role in the model's predictions. On the other hand, **Feature 10**, while still important, has a lower contribution to the overall model output. Understanding the importance of these top features is crucial for model interpretation, as it allows us to focus on the most significant variables when making business decisions or improving the model further.

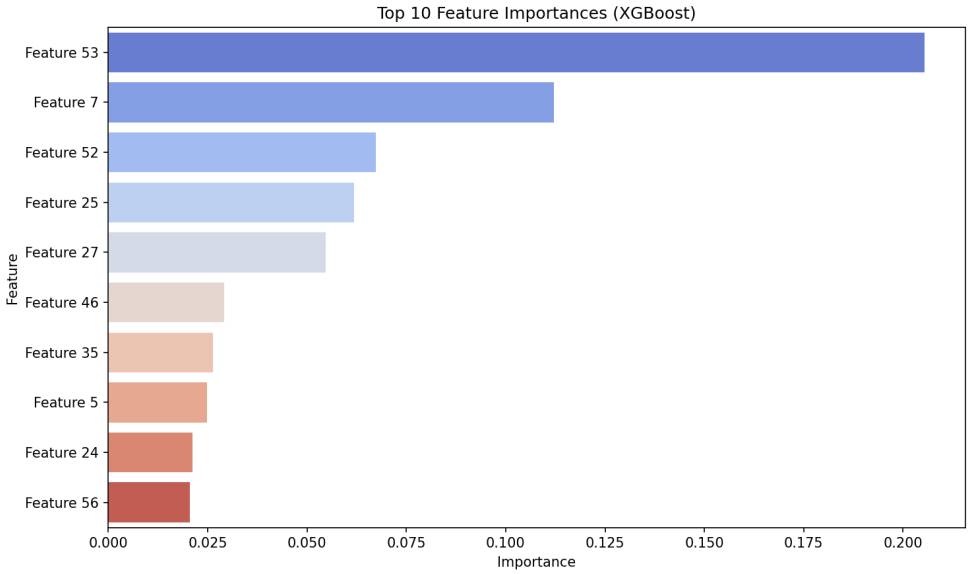


Figure 7: Top 10 Features and Their Importance

4 Hypothesis Testing and Analysis

In this section, we perform hypothesis testing to investigate if there are significant differences between spam and non-spam emails based on certain features in the dataset. Specifically, we tested two hypotheses related to the presence of the word "free" and the special character "$" in the emails.

4.1 Hypothesis 1 - Difference in Means for the Word "Free"

The first hypothesis test examines whether the average frequency of the word "free" differs between spam and non-spam emails.

- **Null Hypothesis** (H_0): The mean frequency of the word "free" in spam and non-spam emails is equal.
- **Alternative Hypothesis** (H_1): The mean frequency of the word "free" differs between spam and non-spam emails.

We performed a two-sample t-test to compare the means of the feature ‘word_freq_1‘, which represents the relative frequency of the word "free" in the email content.
The results of the t-test:

$$\text{Test Statistic: } t = 8.6279, \quad \text{P-value: } p = 8.4878 \times 10^{-18}$$

Since the p-value is much smaller than the significance level ($p < 0.05$), we reject the null hypothesis. This means that the frequency of the word "free" is significantly different between spam and non-spam emails.
Below we have a chart showing the distribution of the word frequency "free" in both spam and non-spam emails.

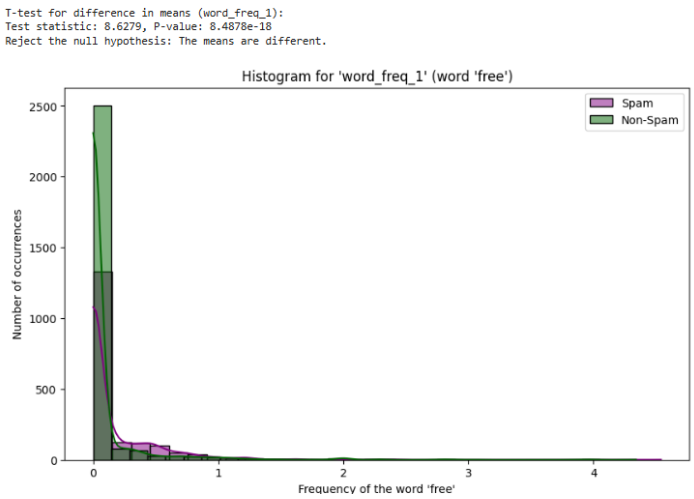


Figure 8: Distribution of ’free’ word frequency in Spam vs Non-Spam emails

4.2 Hypothesis 2: Proportion of Emails Containing the Special Character "$"

The second hypothesis examines whether the proportion of emails containing the special character "$" differs between spam and non-spam emails.

- **Null Hypothesis** (H_0): The proportion of emails containing "$" is the same in both spam and non-spam emails.
- **Alternative Hypothesis** (H_1): The proportion of emails containing "$" differs between spam and non-spam emails.

We conducted a two-proportion Z-test to compare the occurrence of the feature ‘char_freq_1‘, which represents the frequency of the "$" symbol, in both spam and non-spam emails. We used a threshold of 0.01 to determine if the character "$" is present in an email.
The results of the Z-test were as follows:

$$\text{Proportion in Spam Emails: } 13.18\%, \quad \text{Proportion in Non-Spam Emails: } 18.36\%$$

$$\text{Test Statistic: } Z = -4.6474, \quad \text{P-value: } p = 3.3622 \times 10^{-6}$$

Since the p-value is significantly smaller than 0.05, we reject the null hypothesis. This indicates that the proportion of emails containing the character "$" differs significantly between spam and non-spam emails. Interestingly, the proportion of "$" is higher in non-spam emails, suggesting that this feature might be more common in professional or financial communications rather than in spam.
Below we have a chart showing the proportions of emails containing the special character "$" in spam and non-spam emails.

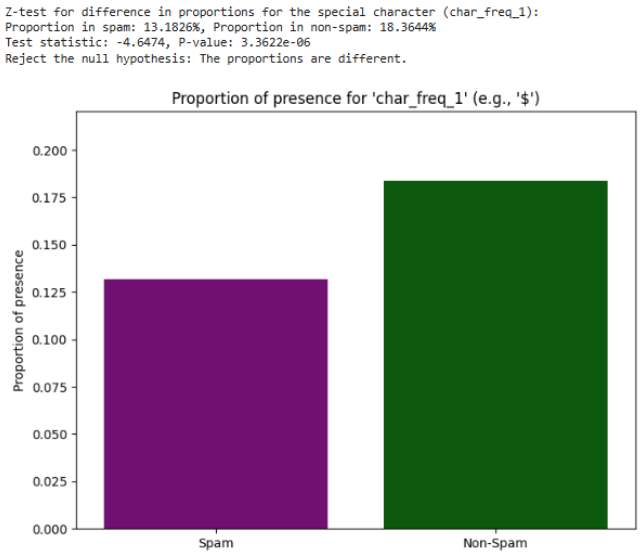


Figure 9: Proportion of emails containing '$' in Spam vs Non-Spam

4.3 Summary of Findings

Both hypothesis tests revealed significant differences in the features between spam and non-spam emails:

- The word "free" appears more frequently in spam emails, which is a strong indicator for spam classification.
- The special character "$" appears more frequently in non-spam emails, possibly due to its usage in professional, business, or financial contexts.

5 Cross-Validation Analysis of Model Performance

The performance of four classification models—Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM)—was evaluated using cross-validation. This approach ensures the robustness and generalizability of the models by assessing their accuracy on multiple subsets of the dataset.

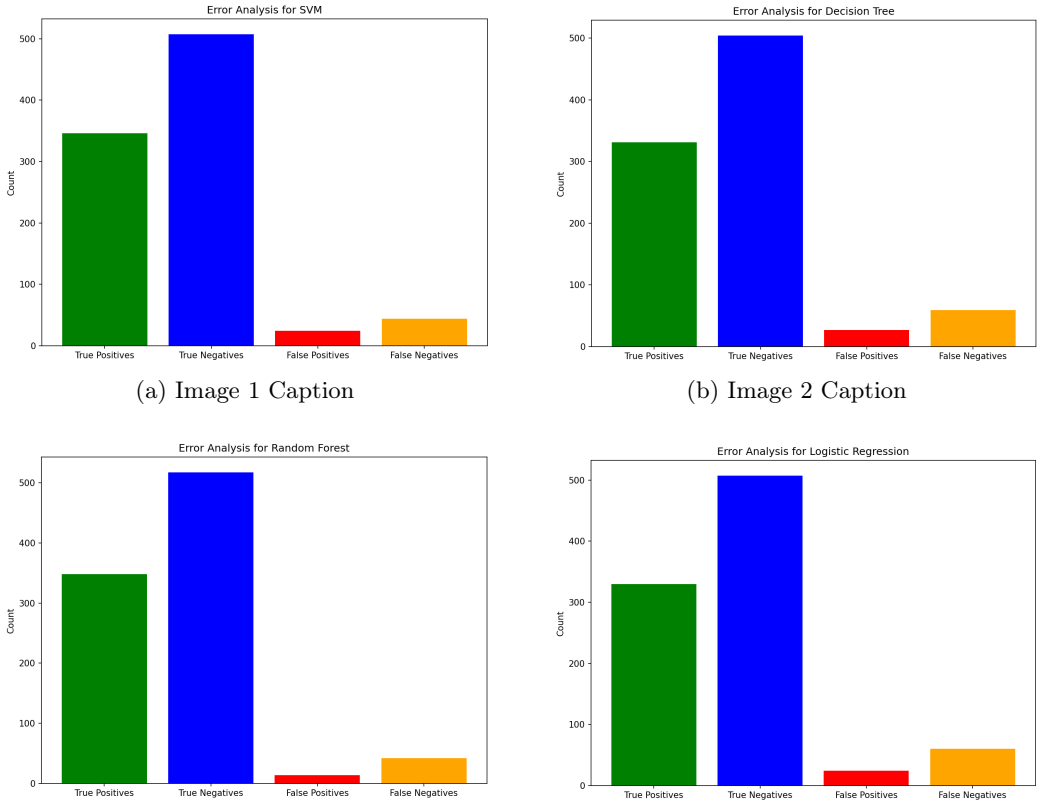
Table 2: Cross-Validation Results for Classification Models

Model	Mean Accuracy	Standard Deviation	Key Insights
Logistic Regression	0.8818	± 0.0043	Stable but moderate performance.
Decision Tree	0.9128	± 0.0079	Higher accuracy, with slight variability.
Random Forest	**0.9511**	± 0.0093	Best performance among models.
SVM	0.9285	± 0.0123	Competitive but with higher variability.

5.1 Key Findings

- **Best Performance:** Random Forest achieved the highest cross-validation accuracy (95.11%), with a standard deviation of ± 0.0093. This indicates high accuracy and consistency across folds.
- **Second-Best Model:** SVM achieved competitive performance with an accuracy of 92.85%, but its higher standard deviation (± 0.0123) indicates greater sensitivity to the subsets used during validation.
- **Moderate Accuracy:** Decision Tree achieved a mean accuracy of 91.28% and demonstrated moderate consistency with a standard deviation of ± 0.0079.
- **Lowest Accuracy:** Logistic Regression exhibited the lowest accuracy (88.18%), but it was the most stable model with a standard deviation of ± 0.0043, making it suitable for scenarios prioritizing simplicity and stability.

5.2 Visualization of Results



(a) Image 1 Caption

(b) Image 2 Caption

(c) Image 3 Caption

(d) Image 4 Caption

Figure 10: Four images arranged with wider side margins and narrower space in the middle.

Model	Type I Error Rate (False Positives)	Type II Error Rate (False Negatives)
Logistic Regression	0.0452	0.1538
Decision Tree	0.0508	0.1513
Random Forest	0.0264	0.1077
SVM	0.0452	0.1128

Table 3: Type I and Type II Error Rates for Various Models

The table summarizes the cross-validation results for four classification models. Random Forest achieved the highest accuracy (95.11%) with low standard deviation, indicating superior performance and stability. SVM showed competitive accuracy (92.85%) but with higher variability.
Decision Tree demonstrated moderate accuracy (91.28%) with reasonable consistency, while Logistic Regression, despite having the lowest accuracy (88.18%), exhibited the highest stability, making it suitable for simpler scenarios.

6 Statistical Tests for Model Performance Metrics

6.1 Objective

The goal of this section is to evaluate the statistical significance of differences in classification accuracy among the models using t-tests and to summarize the results of bootstrap analysis for key metrics such as Accuracy, Precision, Recall and F1-score.

6.2 Results of t-Test for Accuracy

Comparison	t-statistic	p-value	Conclusion
Logistic vs Decision Tree	0.0873	0.9305	No statistically significant difference
Logistic vs Random Forest	-25.2761	<0.0001	Statistically significant difference
Logistic vs SVM	-4.9461	<0.0001	Statistically significant difference
Decision Tree vs Random Forest	-24.3078	<0.0001	Statistically significant difference
Decision Tree vs SVM	-4.8813	<0.0001	Statistically significant difference
Random Forest vs SVM	19.5636	<0.0001	Statistically significant difference

Table 4: Results of t-tests comparing model accuracy

The results indicate that Random Forest significantly outperforms Logistic Regression, Decision Tree, and SVM in terms of accuracy. SVM also exhibits significant differences when compared to Logistic Regression and Decision Tree. Logistic Regression and Decision Tree showed no statistically significant difference in their accuracy.

6.3 Bootstrap Analysis of Metrics

To complement the t-test results, we conducted a bootstrap analysis to estimate the mean, variance, and 95% confidence intervals (CI) for Accuracy, Precision, Recall and F1-score.

Metric	Model	Mean (95% CI)	Variance
Accuracy	Logistic Regression	0.9075 [0.8816, 0.9294]	0.0002
	Decision Tree	0.9073 [0.8843, 0.9295]	0.0002
	Random Forest	0.9460 [0.9310, 0.9637]	0.0001
	SVM	0.9164 [0.8908, 0.9371]	0.0002
Precision	Logistic Regression	0.9302 [0.9007, 0.9587]	0.0002
	Decision Tree	0.9262 [0.8815, 0.9604]	0.0004
	Random Forest	0.9700 [0.9536, 0.9866]	0.0001
	SVM	0.9350 [0.9112, 0.9602]	0.0002
Recall	Logistic Regression	0.8470 [0.8043, 0.8849]	0.0005
	Decision Tree	0.8510 [0.7952, 0.8910]	0.0006
	Random Forest	0.9013 [0.8652, 0.9360]	0.0003
	SVM	0.8644 [0.8173, 0.9085]	0.0006
F1-score	Logistic Regression	0.8865 [0.8569, 0.9122]	0.0002
	Decision Tree	0.8867 [0.8547, 0.9138]	0.0003
	Random Forest	0.9343 [0.9164, 0.9545]	0.0001
	SVM	0.8981 [0.8657, 0.9221]	0.0002

Table 5: Bootstrap analysis results for Accuracy, Precision, Recall and F1-score.

Random Forest consistently demonstrated the highest metrics with narrow confidence intervals, confirming its robustness and reliability. Logistic Regression and Decision Tree had similar performances in accuracy and F1-score, but their metrics were significantly lower than those of Random Forest.

6.4 Monte Carlo Analysis of Metrics

Monte Carlo simulations were conducted to evaluate the robustness and variability of the models across multiple random data splits. This approach provides insights into the stability of the models' performance metrics such as Accuracy, Precision, Recall and F1-Score.

Metric	Model	Mean (95% CI)	Variance
Accuracy	Logistic Regression	0.9097 [0.9004, 0.9252]	0.0001
	Decision Tree	0.9144 [0.8959, 0.9296]	0.0001
	Random Forest	0.9511 [0.9416, 0.9627]	0.0000
	SVM	0.9210 [0.9072, 0.9322]	0.0001
Precision	Logistic Regression	0.9263 [0.9019, 0.9450]	0.0002
	Decision Tree	0.9344 [0.8927, 0.9587]	0.0004
	Random Forest	0.9725 [0.9587, 0.9797]	0.0001
	SVM	0.9343 [0.9172, 0.9538]	0.0002
Recall	Logistic Regression	0.8540 [0.8350, 0.8805]	0.0002
	Decision Tree	0.8573 [0.8391, 0.8764]	0.0001
	Random Forest	0.9098 [0.8884, 0.9361]	0.0002
	SVM	0.8746 [0.8555, 0.8912]	0.0001
F1-Score	Logistic Regression	0.8886 [0.8690, 0.9042]	0.0001
	Decision Tree	0.8941 [0.8660, 0.9102]	0.0002
	Random Forest	0.9400 [0.9303, 0.9574]	0.0001
	SVM	0.9034 [0.8879, 0.9146]	0.0001

Table 6: Monte Carlo simulation results for Accuracy, Precision, Recall and F1-Score

The Monte Carlo results confirm the superior performance of the Random Forest model, which consistently achieved the highest Accuracy, Precision, Recall, and F1-Score with minimal variance. The Logistic Regression and SVM models exhibited moderate performance, while the Decision Tree model showed stable but slightly lower performance metrics compared to Random Forest. The narrow confidence intervals further emphasize the reliability of Random Forest in this classification task.

6.5 Summary and Implications

The statistical analyses confirm that the **Random Forest** model outperforms all other classifiers in terms of accuracy, precision, recall and F1-score. Its consistent performance across metrics, coupled with narrow confidence intervals, highlights its robustness and reliability for this classification task. The **Support Vector Machine** showed competitive results, the observed variability suggests it may require further optimization to achieve comparable consistency. The **Logistic Regression** and **Decision Tree** models served as stable baselines, yet they lacked the performance edge necessary for high-stakes classification scenarios. These results strongly support the selection of **Random Forest** as the primary model for deployment in this project, providing the required accuracy and robustness for reliable decision-making in real-world applications.

7 Receiver Operating Characteristic (ROC) Curve Analysis

To properly evaluate the classification performance of the models, we generated receiver operating characteristic (ROC) curves for logistic regression, decision tree, random forest and support vector machine (SVM). The ROC curve provides a visual representation of the trade-off between sensitivity (true positive rate) and specificity (false positive rate) for different classification thresholds. In addition, the area under the curve (AUC) serves as a summary metric, providing a single outcome to compare the performance of the model.

7.1 Summary of Results

- **Logistic Regression:** Achieved an AUC of **0.91**, which reflects solid performance for a baseline model. While it performs well in distinguishing classes, its limitations become apparent when compared to ensemble-based approaches.
- **Decision Tree:** With an AUC of **0.94**, this model demonstrates an impressive ability to handle nonlinear relationships in the data. It balances sensitivity and specificity effectively, making it a strong candidate for less complex problems.
- **Random Forest:** Stands out with the highest AUC of **0.97**. This score underscores the model's exceptional capability to handle complex patterns and provide reliable classifications, thanks to its ensemble nature.
- **SVM:** Achieved an AUC of **0.92**, which indicates strong performance. However, it is slightly outperformed by Random Forest, suggesting that while it is competitive, it may require more fine-tuning for optimal results.

7.2 Key Insights

The ROC analysis reveals several important takeaways:

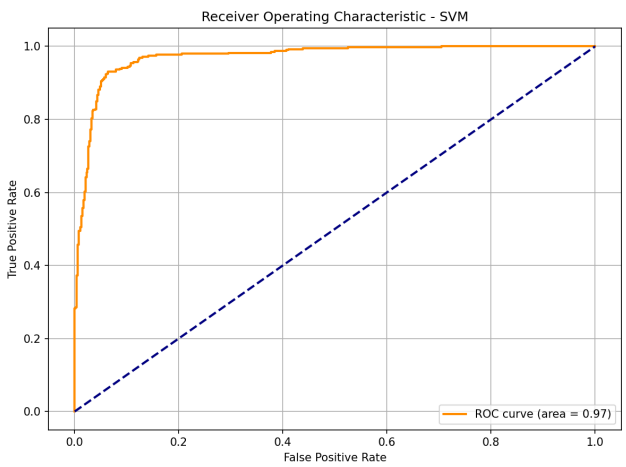
- **Random Forest is the clear leader.** Its AUC of 0.97 not only confirms its superior performance across various thresholds but also reinforces the findings from accuracy, precision, and F1-score metrics. This makes Random Forest the most reliable choice for this classification task.
- **Decision Tree and SVM are strong contenders.** Decision Tree's AUC of 0.94 highlights its ability to capture meaningful patterns without the complexity of ensemble methods. Similarly, SVM's AUC of 0.92 showcases its effectiveness, although its sensitivity to hyperparameters may require additional optimization in practical applications.
- **Logistic Regression as a baseline.** While Logistic Regression scored an AUC of 0.91, it serves as a robust and interpretable baseline. It may not excel in handling intricate relationships within the data but provides a solid starting point for simple scenarios.

7.3 Further Considerations

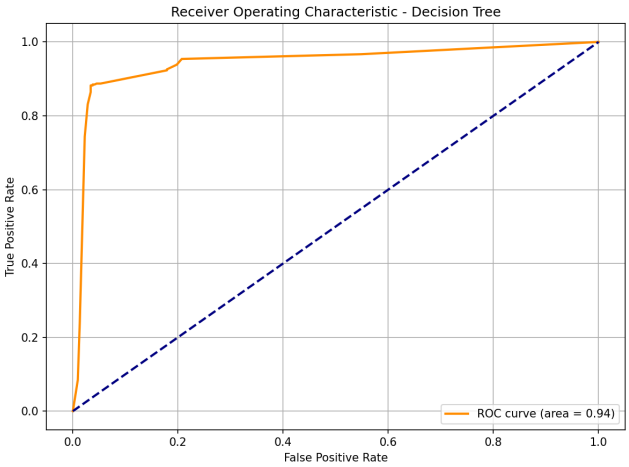
While the ROC curves and AUC scores provide a high-level view of model performance, it's crucial to consider the practical context of the task. For example:

- **Error trade-offs:** Random Forest minimizes incorrect classifications, but in scenarios where interpretability is critical (e.g., medical diagnosis), simpler models like Logistic Regression or Decision Tree may still be superior.
- **Threshold tuning:** AUC evaluates performance across all thresholds, but real-world applications it is often necessary to set a specific threshold to optimally balance sensitivity and specificity optimally.
- **Computational cost:** Random Forest's ensemble nature makes it computationally expensive compared to Logistic Regression or Decision Tree. This trade-off should be considered for large-scale deployments.

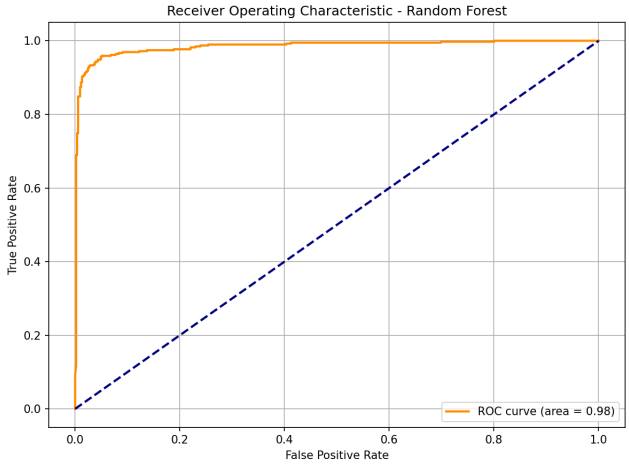
7.4 Visualizations



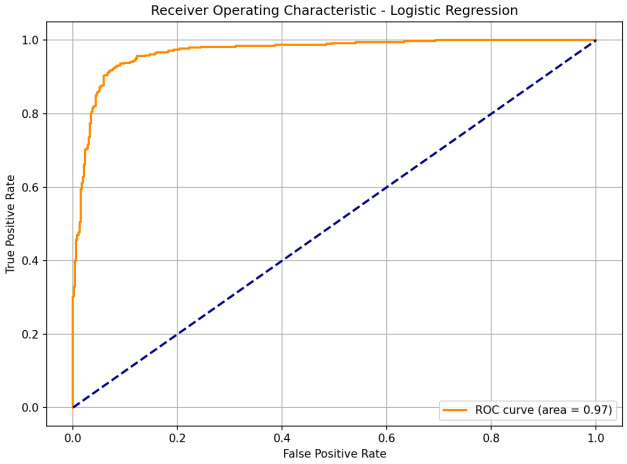
(a) Logistic Regression



(b) Decision Tree



(c) Random Forest



(d) Support Vector Machine

Figure 11: ROC Curves for Logistic Regression, Decision Tree, Random Forest, and SVM Models

7.5 Conclusion

Analysis of ROC curves and AUC values confirms that **Random Forest** is the most effective model for this classification task. Its ability to handle complex data patterns and maintain high sensitivity and specificity makes it the optimal choice for deployment. The Decision Tree and SVM models provide competitive alternatives, while Logistic Regression offers a robust and interpretable baseline. Future work could explore the combination of these models using ensemble techniques to further improve performance.

8 Conclusion

This project aimed to analyze the Spambase dataset to develop and evaluate classification models for distinguishing spam emails from legitimate (non-spam) emails. By employing statistical and machine learning techniques, we gained valuable insights into the characteristics of spam emails and the effectiveness of various classification approaches.

8.1 Key Findings

We can observe that:

- **Dataset Characteristics:** Features such as word frequencies, character frequencies, and capitalization patterns are highly predictive of email classification. However, the dataset exhibited challenges, including skewed distributions, outliers, and slight class imbalance, which required preprocessing to optimize model performance.
- **Hypothesis Testing:** Statistical tests confirmed significant differences between spam and non-spam emails for selected features, such as the frequency of the word *"free"* and the occurrence of the special character *"$"*. These differences highlight the importance of specific linguistic and formatting patterns in spam detection.
- **Model Evaluation:** Among the models tested, **Random Forest** emerged as the most effective classifier, achieving the highest accuracy, precision, recall, F1-score, and AUC. SVM also performed competitively, while Logistic Regression and Decision Tree served as interpretable but less robust baselines.
- **Statistical Robustness:** Bootstrap analysis and t-tests validated the reliability of the models' performance metrics, confirming the statistical significance of Random Forest's superiority.

8.2 Implications of the Analysis

The findings of this project have practical implications for the development of spam detection systems:

- Random Forest, due to its exceptional performance, is recommended for deployment in scenarios where classification accuracy is critical.
- Simpler models such as Logistic Regression may still be valuable in resource-constrained environments or when interpretability is a priority.
- The identified key features, such as word and character frequencies, can guide feature selection in future spam detection frameworks.

8.3 Limitations

Despite the successful outcomes, several limitations should be acknowledged:

- The dataset is relatively small and may not fully represent the diversity of real-world emails, potentially limiting generalizability.
- The models evaluated in this study are based on traditional machine learning algorithms. More advanced techniques, such as deep learning, could further enhance performance.
- Class imbalance, although minor, could still impact the models' ability to classify rare cases accurately.

8.4 Future Work

Building on this project, future research could explore the following directions:

- Testing the models on larger and more diverse datasets to validate their robustness in different contexts.
- Incorporating advanced techniques, such as deep neural networks or transformers, for improved classification performance.
- Investigating ensemble methods, such as stacking or boosting, to combine the strengths of multiple models.
- Exploring interpretability techniques, such as SHAP or LIME, to better understand the influence of individual features on model predictions.

8.5 Final Remarks

In conclusion, this project successfully demonstrated the application of statistical and machine learning methods to spam email classification. By leveraging insights from data exploration, hypothesis testing, and model evaluation, we identified effective approaches for building reliable spam detection systems. The Random Forest model stands out as the optimal solution, balancing high performance and robustness. With further refinement and testing, the methods and findings from this study could contribute to more advanced and practical anti-spam solutions in real-world applications.