**CSCI 626 Information Retrieval**

**New York Institute of Technology**

****

**Spam Email Detection System**

**Group - 10**

**Sharon - 1318510**

**Narsimha Reddy - 1318622**

**Jahnavi Vemuri - 1318900**

**Vishnu Ravikumar - 1312218**

# **Table of contents**

[**Abstract**](#_dddna2pc45a4) **2**

[**Overview**](#_wno56xau6od7) **3**

**Methodology****4**

[**Experiments and Results**](#_34qwdav96q9s) **10**

[**Future work**](#_xj9f3uho8cdq) **11**

[**Challenges**](#_eyjzkoij4a8n) **12**

[**Conclusion**](#_jc9vih9uq1c2) **12**

[**References 1**](#_3ic5z5ql447u)**3**

# 

# 

# 

# 

Abstract

E-mail has become a necessary component of our everyday life due to the quick growth of digital communication. Nevertheless, this ease of use has also made room for an increase in unwanted and harmful emails, or spam. Spam not only reduces productivity but can also present security risks and privacy violations. In light of this, creating efficient spam detection systems has become essential in the current digital environment.

Our project presents a comprehensive approach to classifying emails as spam or ham (non-spam) using machine learning techniques. The methodology includes balancing the dataset, preprocessing the text, and visualizing the data. Initially, a CSV file containing labeled email data is processed to separate spam and ham emails. An equal number of spam and ham emails are randomly selected and shuffled to create a balanced dataset.

Text preprocessing of data involves tokenization, removal of stopwords, stemming, and normalization, transforming the raw text into a format suitable for analysis. Visualization techniques, such as word clouds and pie charts, offer insights into the frequency and distribution of words within the dataset.

Three different classifiers—Naive Bayes, Support Vector Machine (SVM), and Random Forest—are trained using Term Frequency-Inverse Document Frequency (TF-IDF) features extracted from the text. Each model's performance is evaluated based on accuracy, precision, recall, F1 score, and ROC-AUC metrics.

The results indicate that the Random Forest classifier outperforms the others in terms of accuracy and general performance. The project not only highlights effective methods for handling imbalanced datasets and text preprocessing but also demonstrates the comparative strengths of different machine learning models in spam detection. The findings can inform future research and applications in email filtering and text classification systems.

Overview

Problem Statement:

Despite the continuous evolution of email filtering technologies, spam emails remain a pervasive problem that affects users worldwide. These emails encompass a range of malicious activities including phishing scams, fraudulent schemes, and intrusive advertisements. Such spam messages not only disrupt user communication but also pose severe risks to personal and financial security by attempting to steal sensitive information or deliver malware.

Challenges:

Spam emails, which are inherently diverse and often more sophisticated than ham emails, create several challenges for detection systems. A significant issue is the data imbalance where spam emails are outnumbered by legitimate ones, skewing the dataset. This imbalance can bias the model, leading to inaccurate classifications. Textual noise and the wide variety of spam content—ranging from phishing attempts to advertisements—complicate the preprocessing and feature extraction stages. Effective transformation of this raw text into meaningful numerical data using techniques like TF-IDF is crucial yet complex. Lastly, selecting and optimizing machine learning models that can generalize well to unseen data and accurately differentiate between spam and ham remains a challenging aspect of developing a reliable spam detection system.

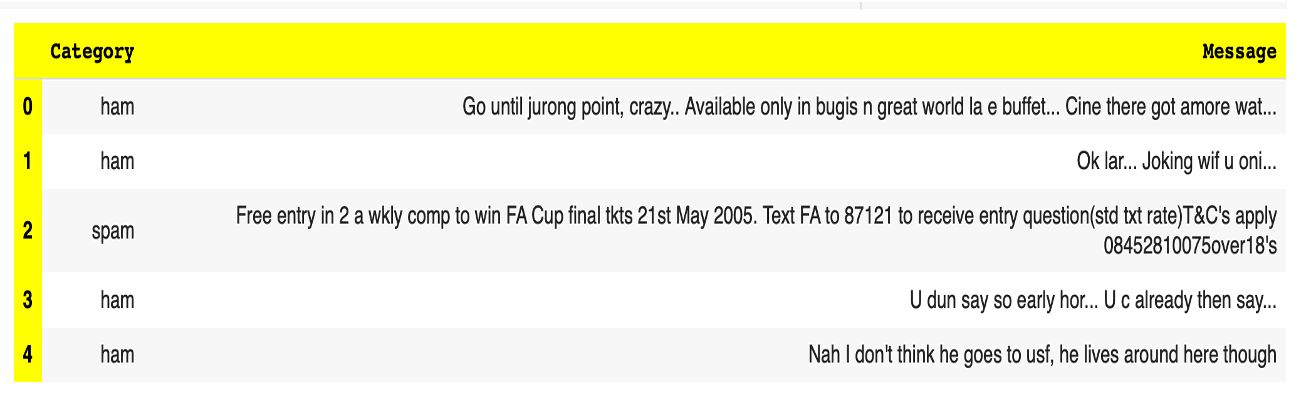
Objective:

The primary goal is to build a robust spam detection system using **Naive Bayes** and **Support Vector Machine (SVM)**algorithms. This involves balancing the dataset to mitigate the issue of data disparity between spam and ham emails and applying comprehensive text preprocessing to clean and standardize the email content. Key features will be extracted using **TF-IDF vectorization** to convert the text into a format suitable for machine learning. The project aims to train and evaluate both the Naive Bayes and SVM classifiers, comparing their effectiveness in detecting spam. Comprehensive performance metrics will be used to assess the accuracy, precision, recall, F1 score, and reliability of the detection system.

Methodology

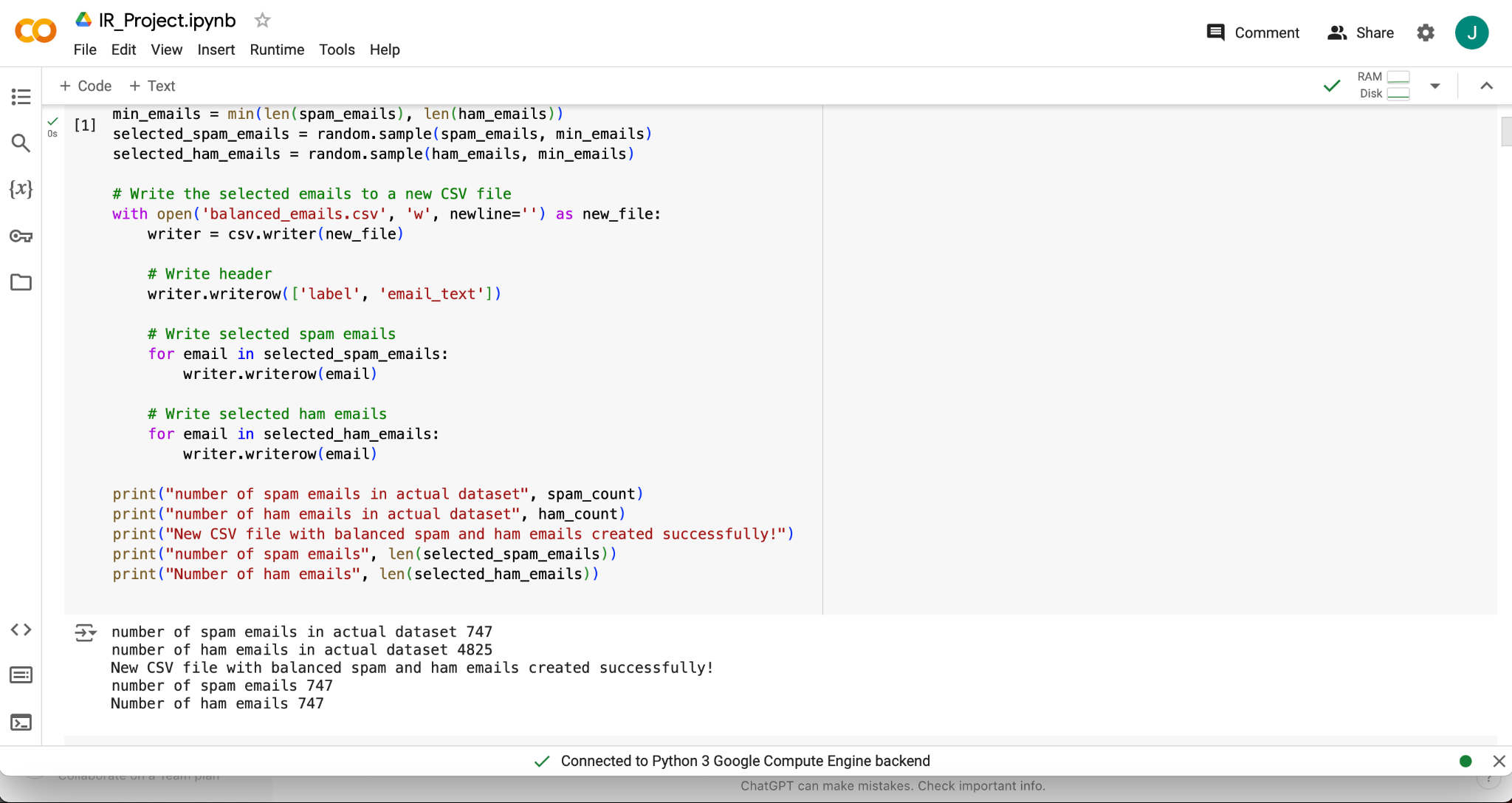
**Data Collection:**

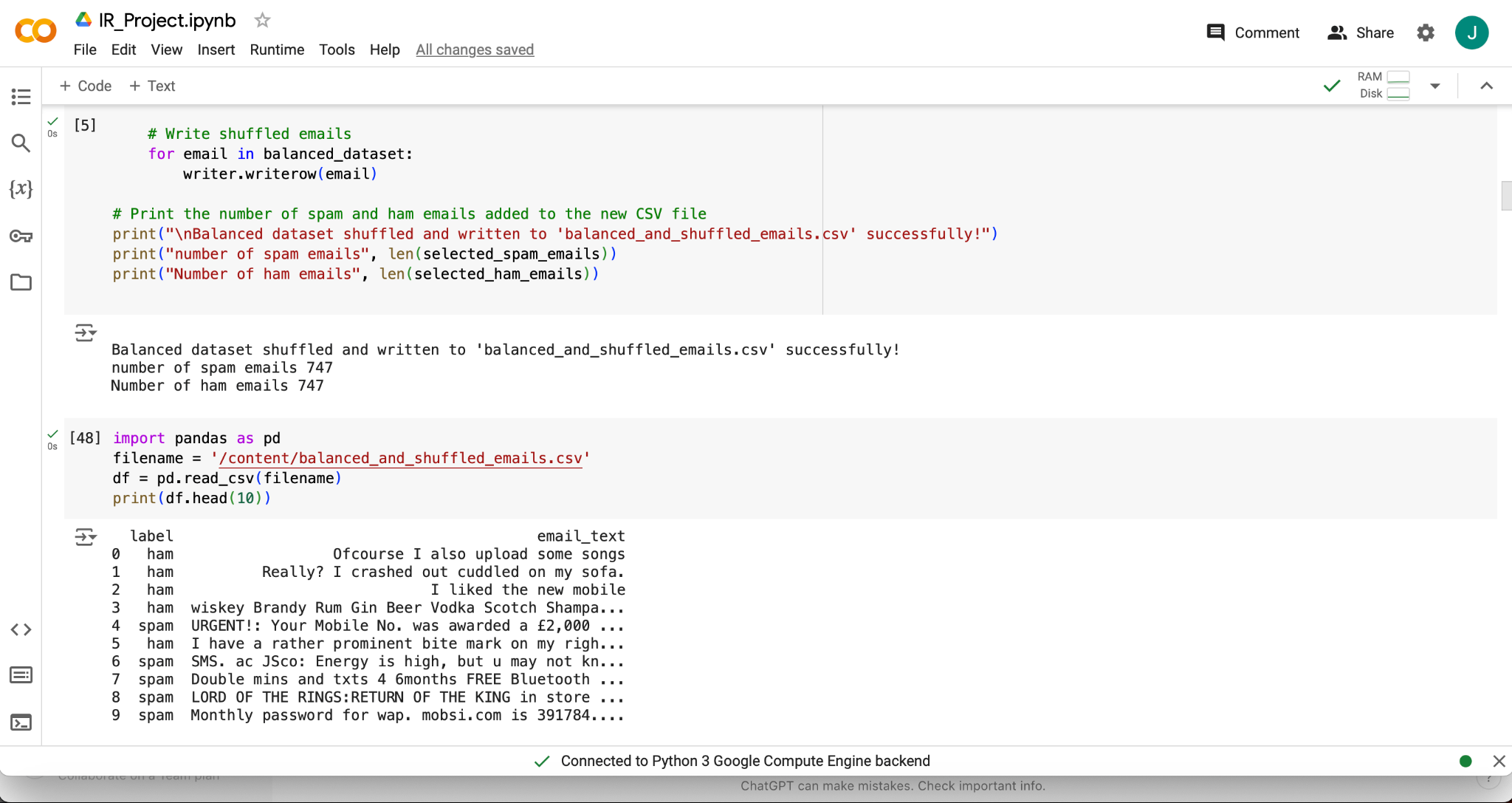
For our spam email detection project, we examined various datasets and chose one that best aligns with the problem we aim to address. The selected dataset, sourced from [Kaggle](https://www.kaggle.com/), is well-suited for developing and testing our spam detection models. It contains **5,573 rows** of email data, providing a diverse range of examples for robust model training. This dataset includes various types of spam emails such as phishing attempts, scams, fraud emails, malware distributors, and promotional advertising emails. The variety in the dataset ensures that the developed models can handle different forms of spam effectively, making it a comprehensive resource for tackling the challenges of email spam detection

<https://docs.google.com/spreadsheets/d/1RDg2u417pfRmcEzveRdnJpVV1Ck-ayIx8BUN-5FMrQc/edit#gid=632363594> 

Data Preprocessing:

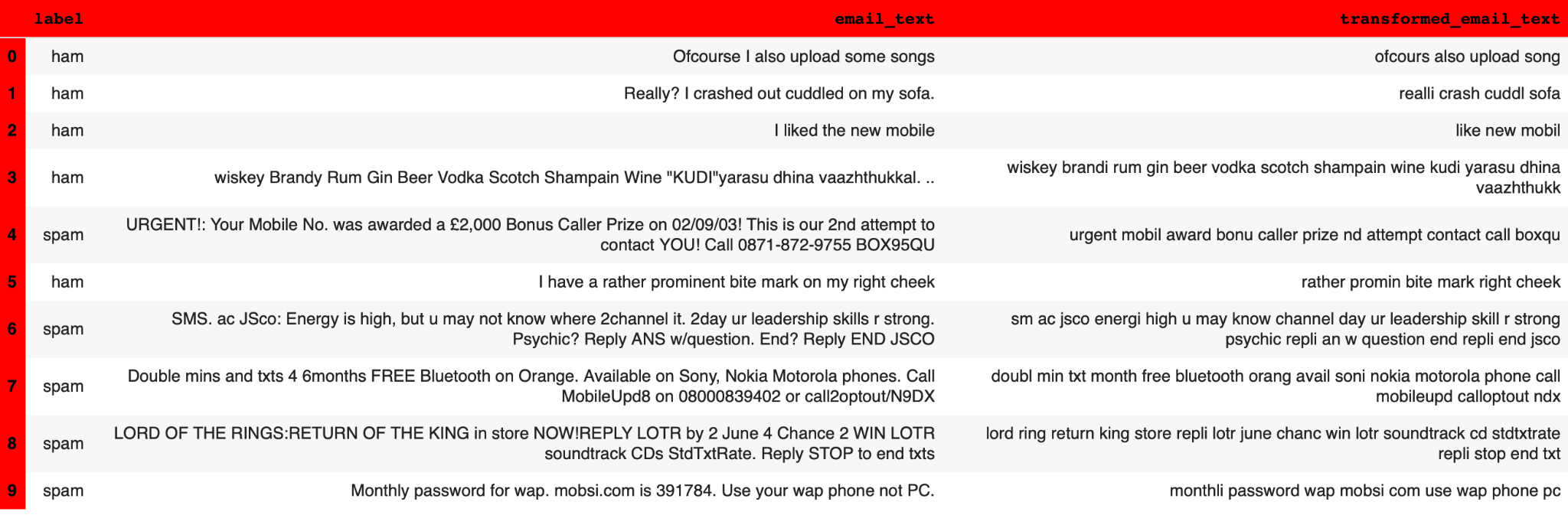
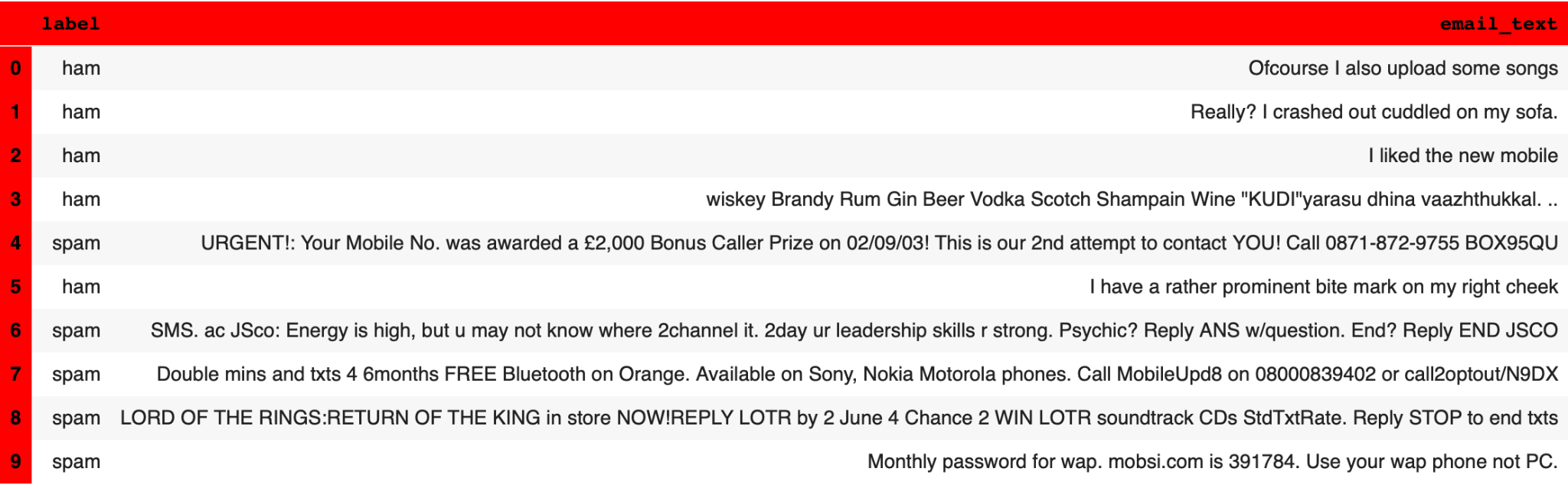
Data preprocessing prepares the dataset for effective spam detection by addressing imbalances and ensuring data cleanliness. Initially, the dataset is balanced by randomly selecting equal numbers of spam and ham emails, creating a fair representation for model training. This balanced subset is then shuffled to ensure randomness and prevent order biases. Following this, the emails are stored in a new dataset, ready for further analysis and text preprocessing. This step is crucial to avoid model bias and improve the reliability of subsequent text-based feature extraction and classification.





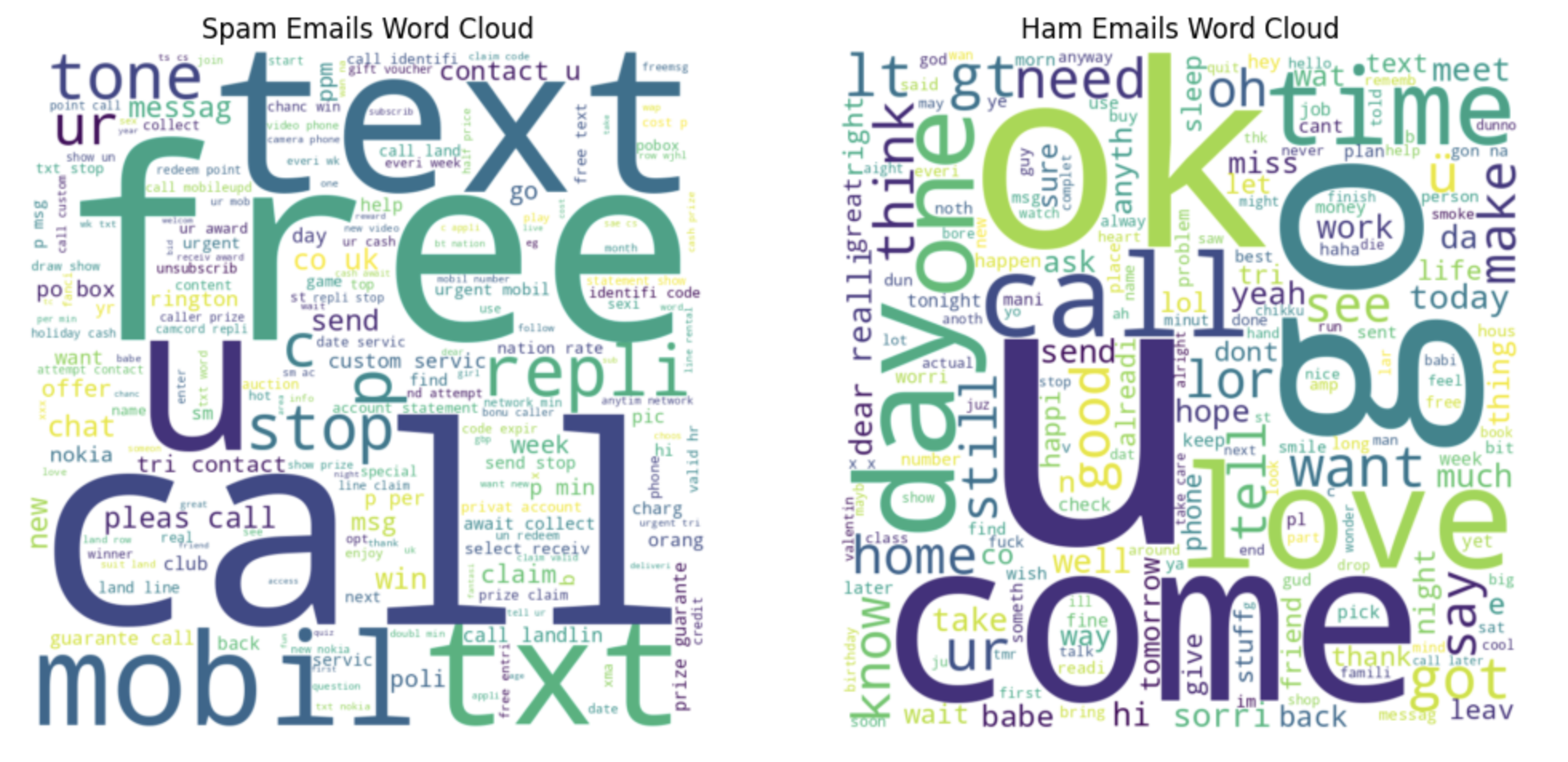
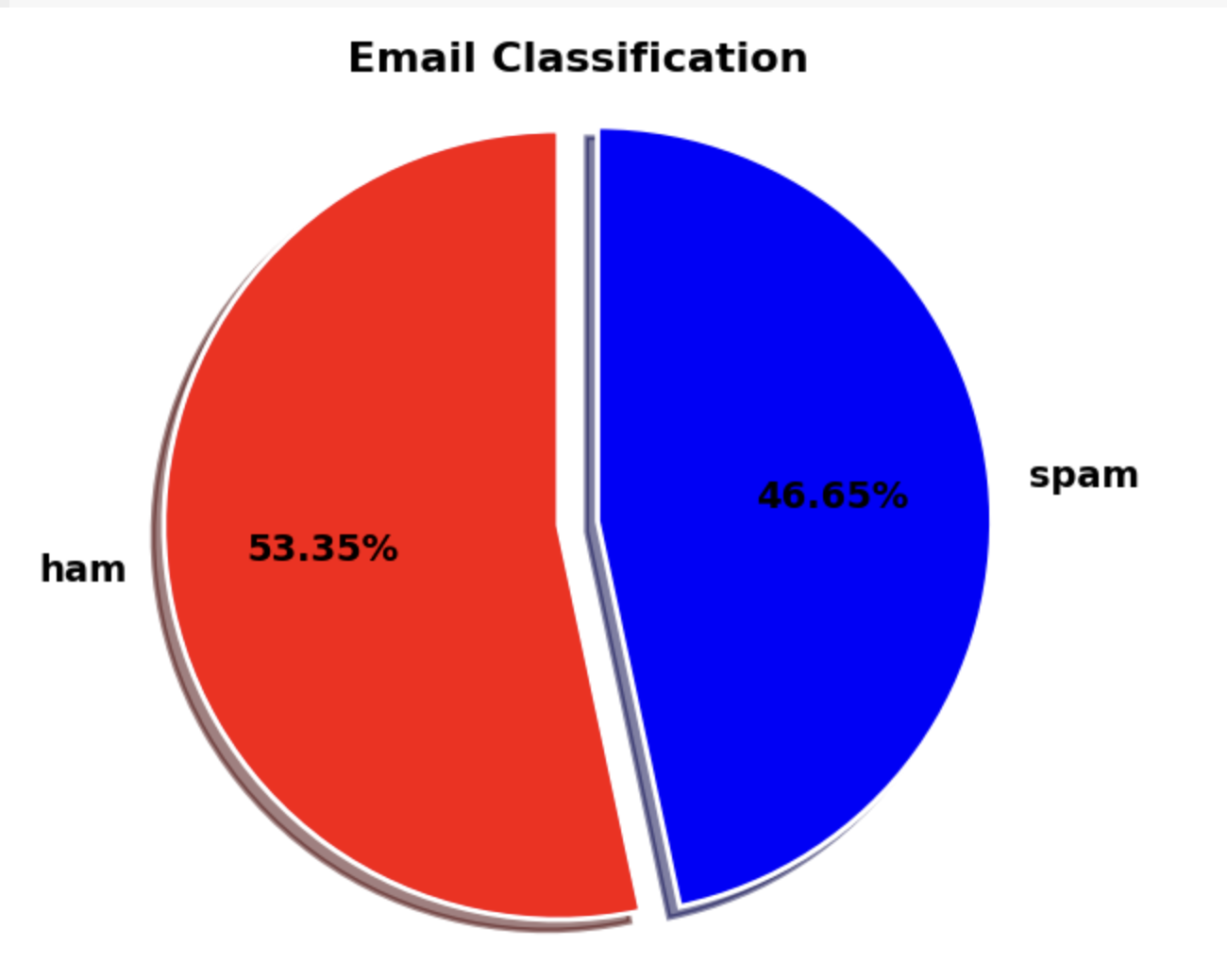
Text Preprocessing:

Text preprocessing refines raw email content into a structured and analyzable format, crucial for effective spam detection. The process begins with **tokenization**, breaking the text into individual words or tokens, followed by **normalization**, where the text is converted to lowercase, and punctuation, digits, and special characters are removed. **Stopwords**, which are common but non-informative words, are eliminated to focus on meaningful terms. Lastly, **stemming** reduces words to their root forms, consolidating different variations of the same word. This comprehensive cleaning and standardization of the text ensure that only relevant and consistent information is extracted, facilitating accurate feature extraction and improving the performance of machine learning models in distinguishing spam from ham emails.



Data Visualization:

Data visualization provides crucial insights into the dataset's composition and patterns in spam and ham emails. A **pie chart** depicts the distribution of spam and ham emails, helping to assess the dataset's balance. **Word clouds** for each category highlight frequently occurring terms, revealing common themes and vocabulary used in spam (e.g., phishing, scams, advertisements) and ham emails. These visualizations aid in understanding the dataset and inform the feature extraction and model training processes for effective spam detection.



Feature extraction:

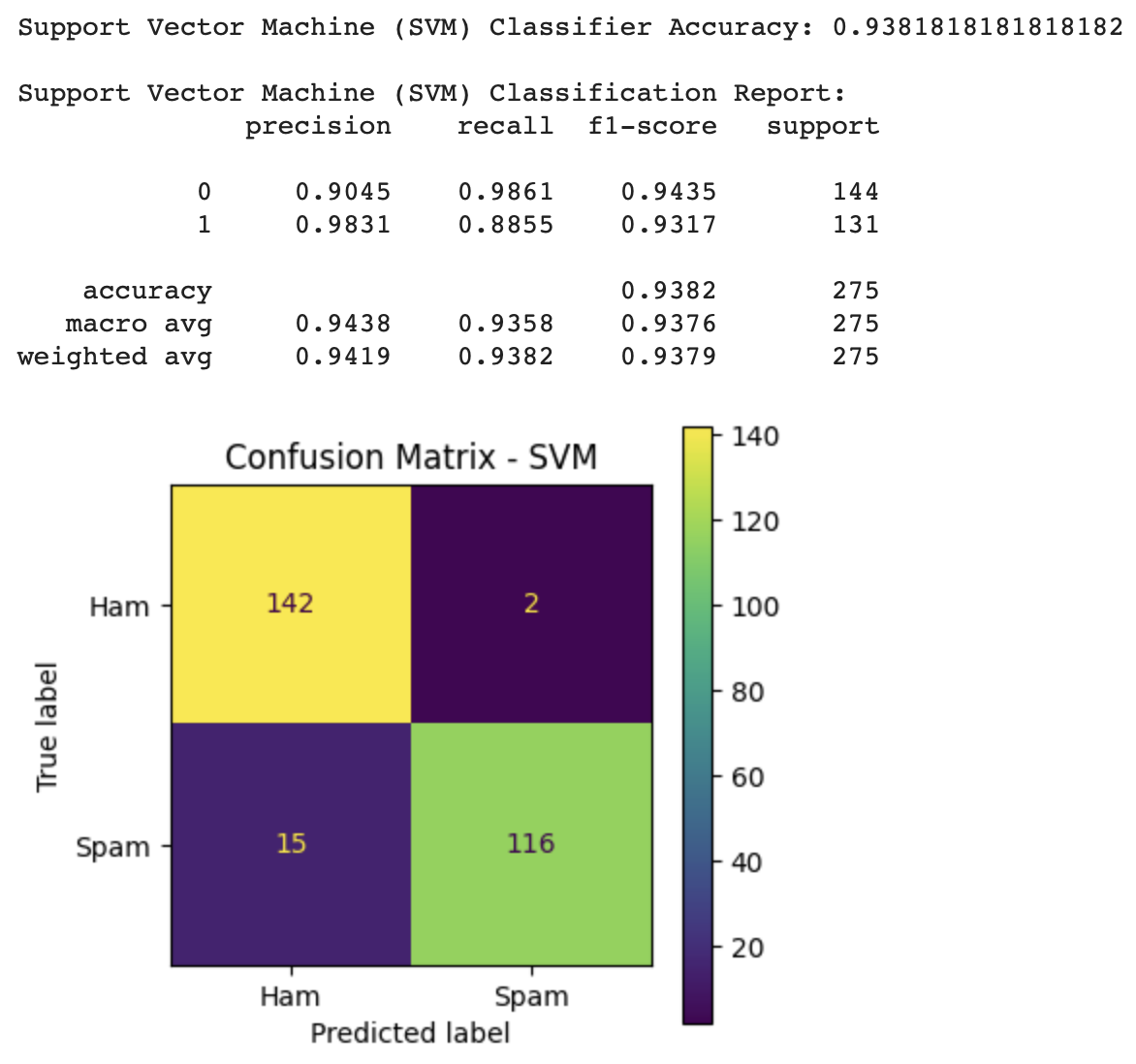
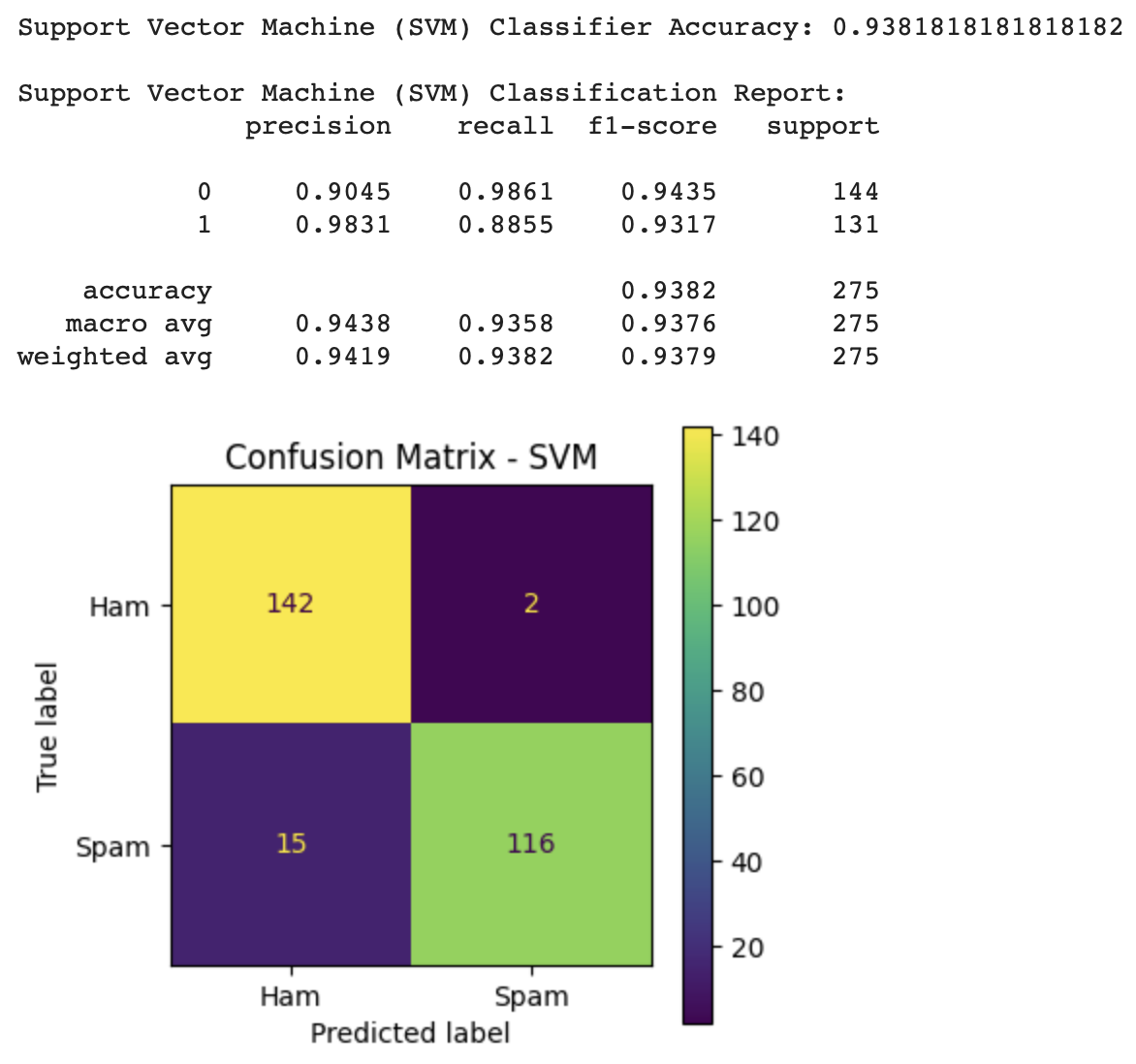
In the project, feature extraction is achieved through **Term Frequency-Inverse Document Frequency (TF-IDF) Vectorization**. This technique converts email text into numerical features by measuring the frequency of words (TF) and their uniqueness across the dataset (IDF). Using the TfidfVectorizer, the text data is transformed into a matrix of numerical values, with each feature representing the importance of a word. This method highlights key words and phrases typical of spam, enabling effective classification by machine learning models. The process ensures that the models can accurately differentiate between spam and ham emails based on the relevance of terms.

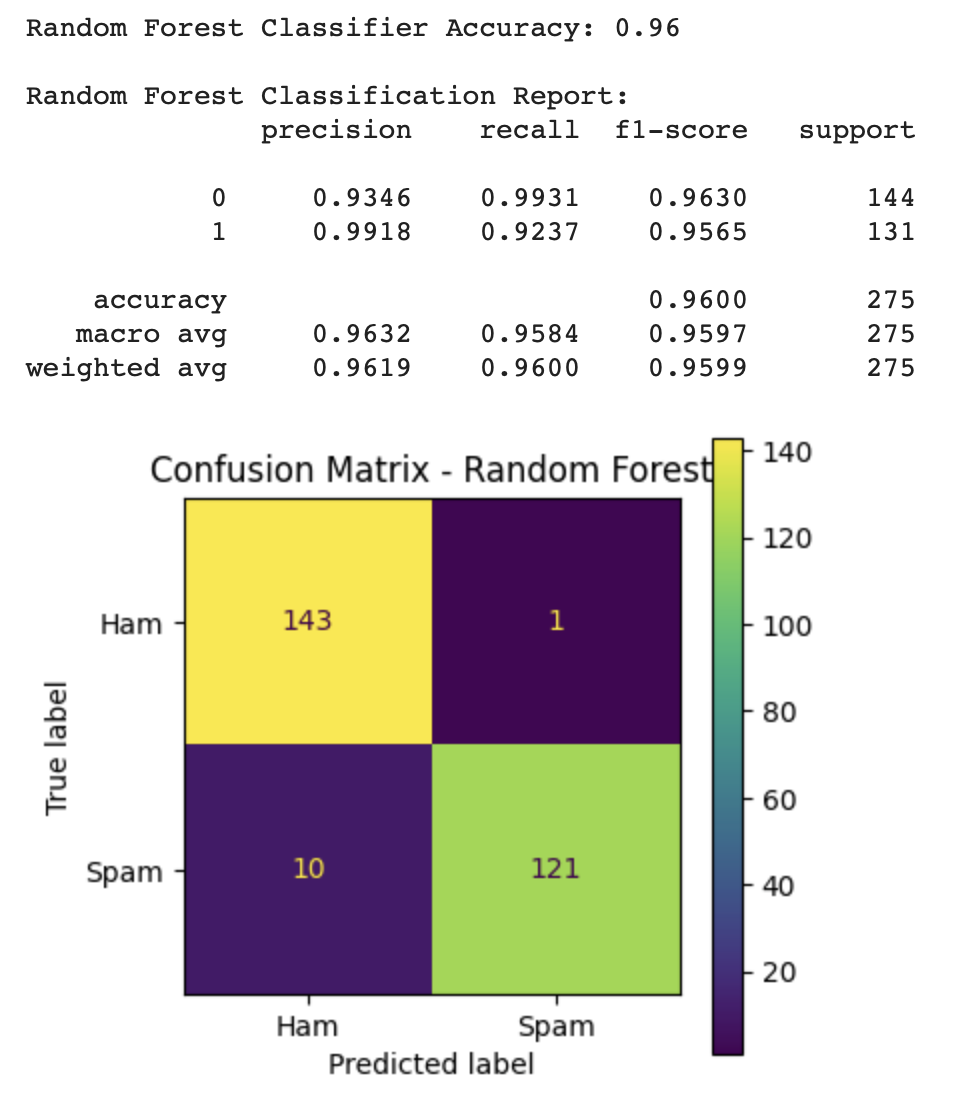
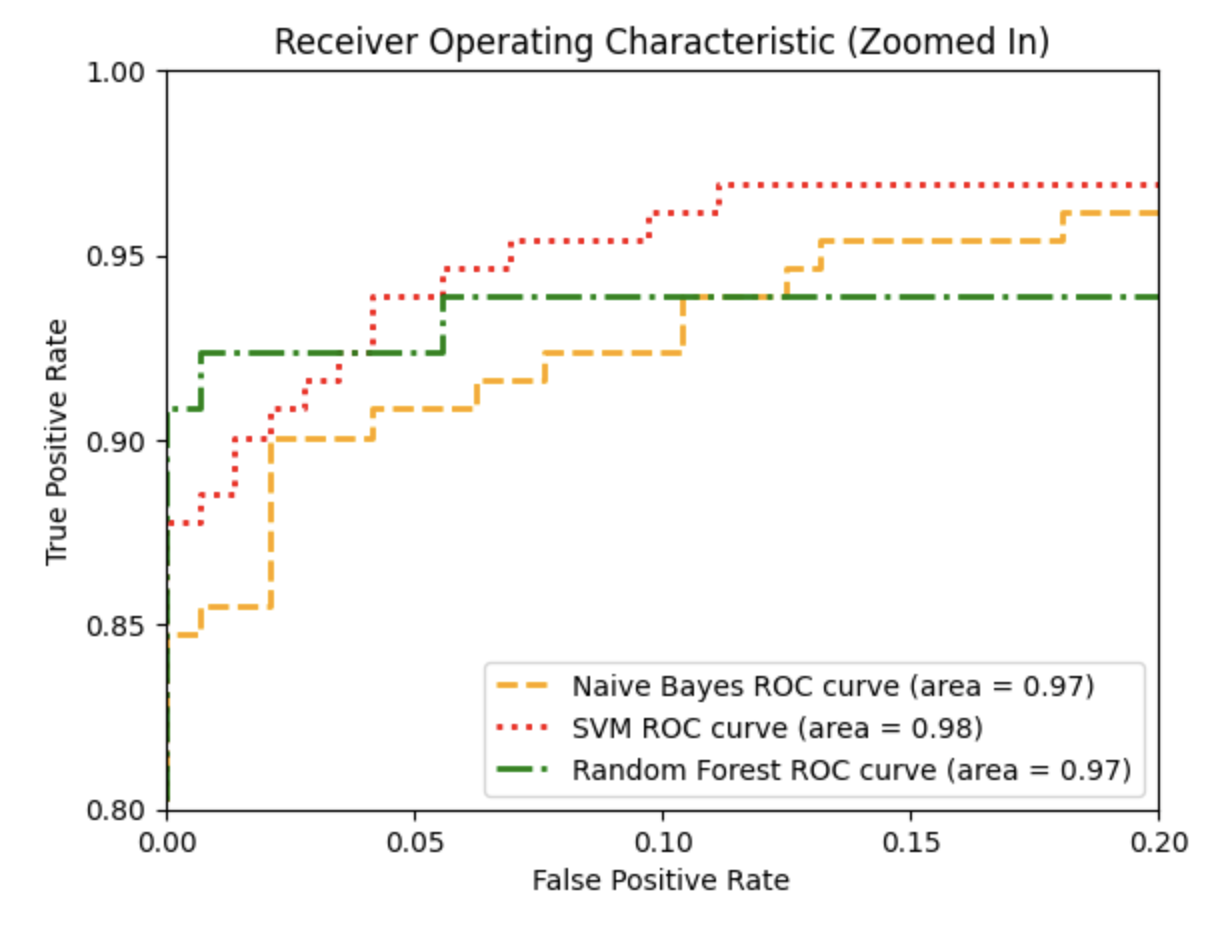
Model Training:

Model training involves developing and evaluating machine learning models to classify emails as spam or ham. The dataset is split into training and testing sets to ensure robust evaluation. Three classifiers are employed:

1. **Naive Bayes**, which uses a probabilistic approach effective for text data.
2. **Support Vector Machine (SVM)**, which uses a linear kernel to handle high-dimensional TF-IDF features and find the optimal separation between spam and ham.
3. **Random Forest**, which combines multiple decision trees for robust classification by averaging their outputs.

Each model is trained on TF-IDF vectorized text data, and their performance is assessed using accuracy, precision, recall, F1 score, ROC-AUC and confusion matrix. This evaluation helps in determining the most effective algorithm for accurate spam detection.

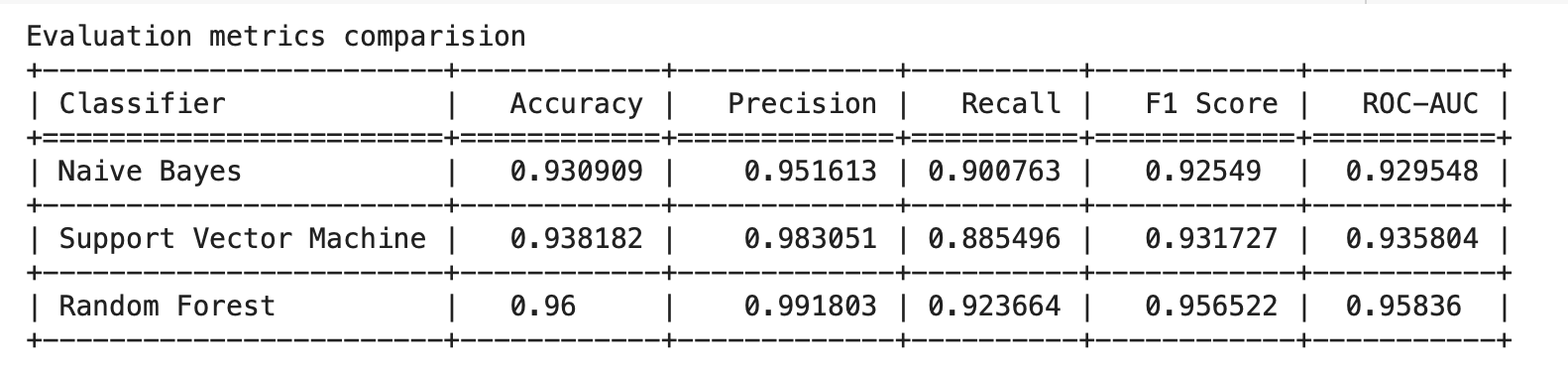
 

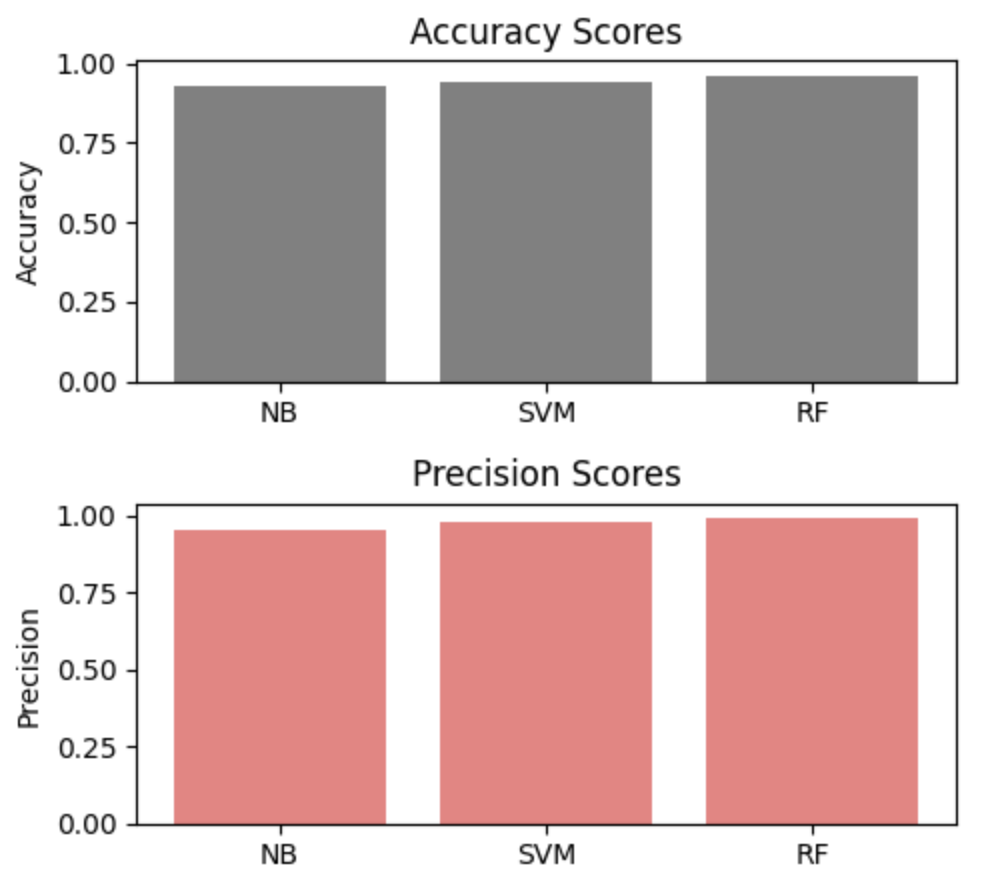
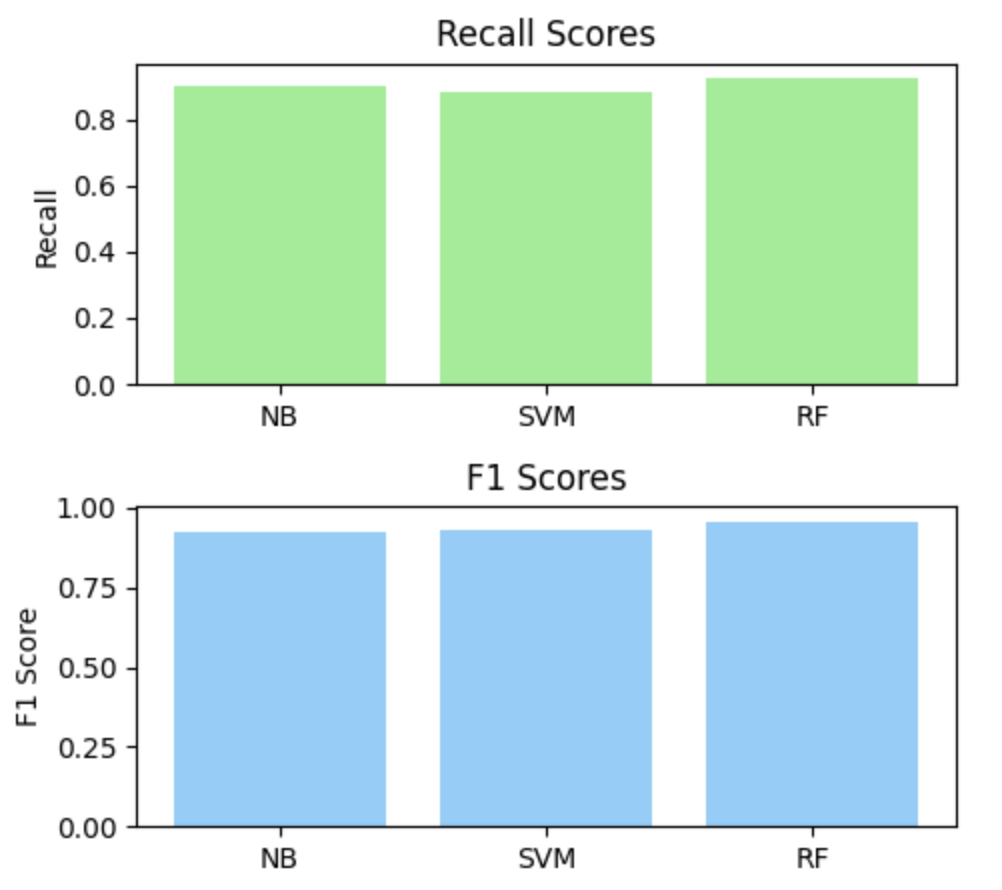
Evaluation Metrics:

In the project, various metrics are used to evaluate the performance of spam detection models:

1. **Accuracy:** Measures the proportion of correctly classified emails among all emails, giving an overall effectiveness of the model.
2. **Precision:** Indicates the proportion of true spam emails among those predicted as spam, reducing false positives.
3. **Recall (Sensitivity):** Represents the proportion of actual spam emails that were correctly identified, minimizing false negatives.
4. **F1 Score:** Combines precision and recall into a single metric, useful for evaluating models on imbalanced datasets.
5. **ROC-AUC Score:** Assesses the model's ability to distinguish between spam and ham across various thresholds, summarizing the trade-off between true positive and false positive rates.

These metrics collectively help in assessing and comparing the effectiveness of the Naive Bayes, SVM, and Random Forest models in classifying spam and ham emails.



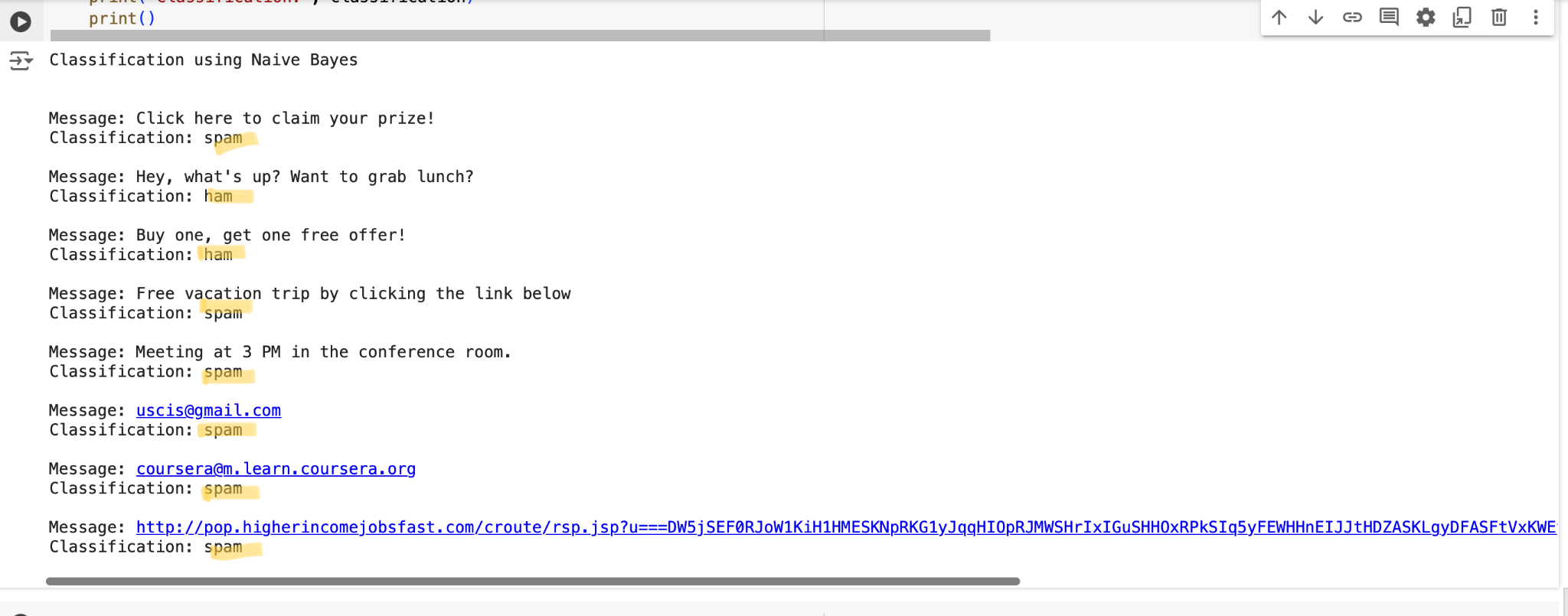
**Experiments and Results**

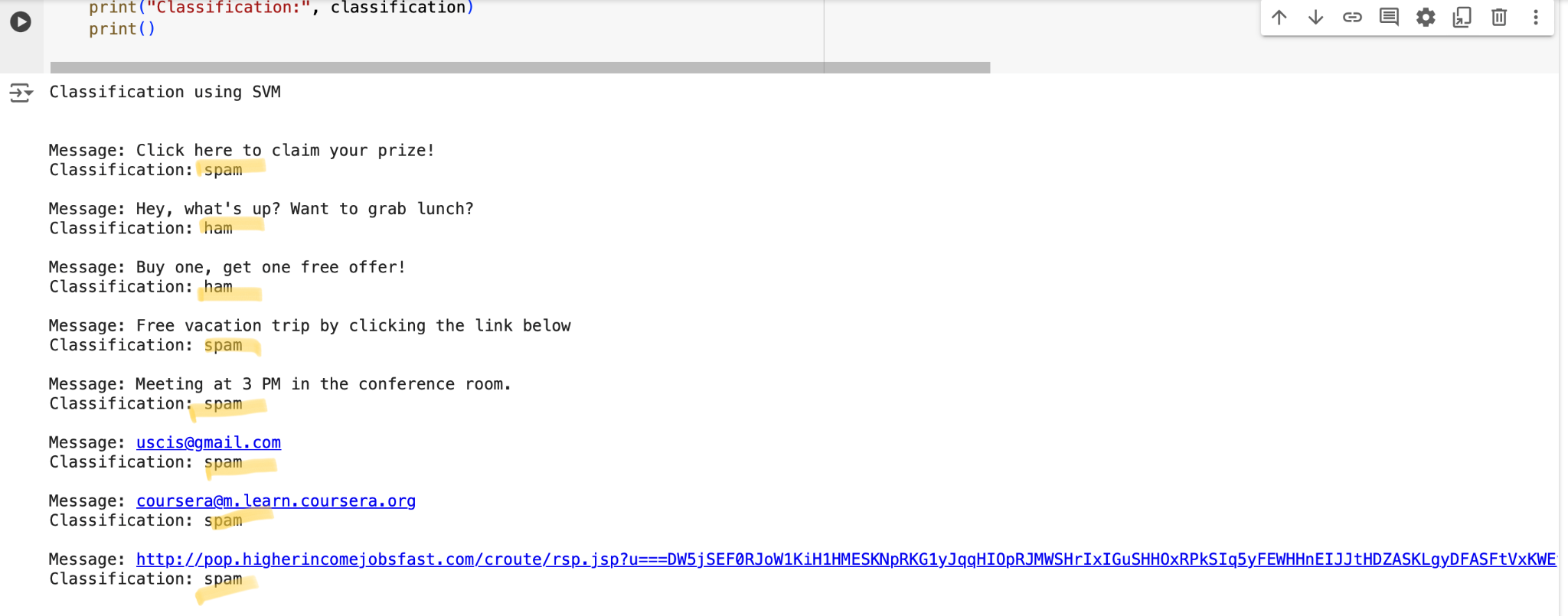
The spam or ham classification process converts real email content into numerical features using **TF-IDF Vectorization**and leverages trained machine learning models to determine whether an email is spam or ham.

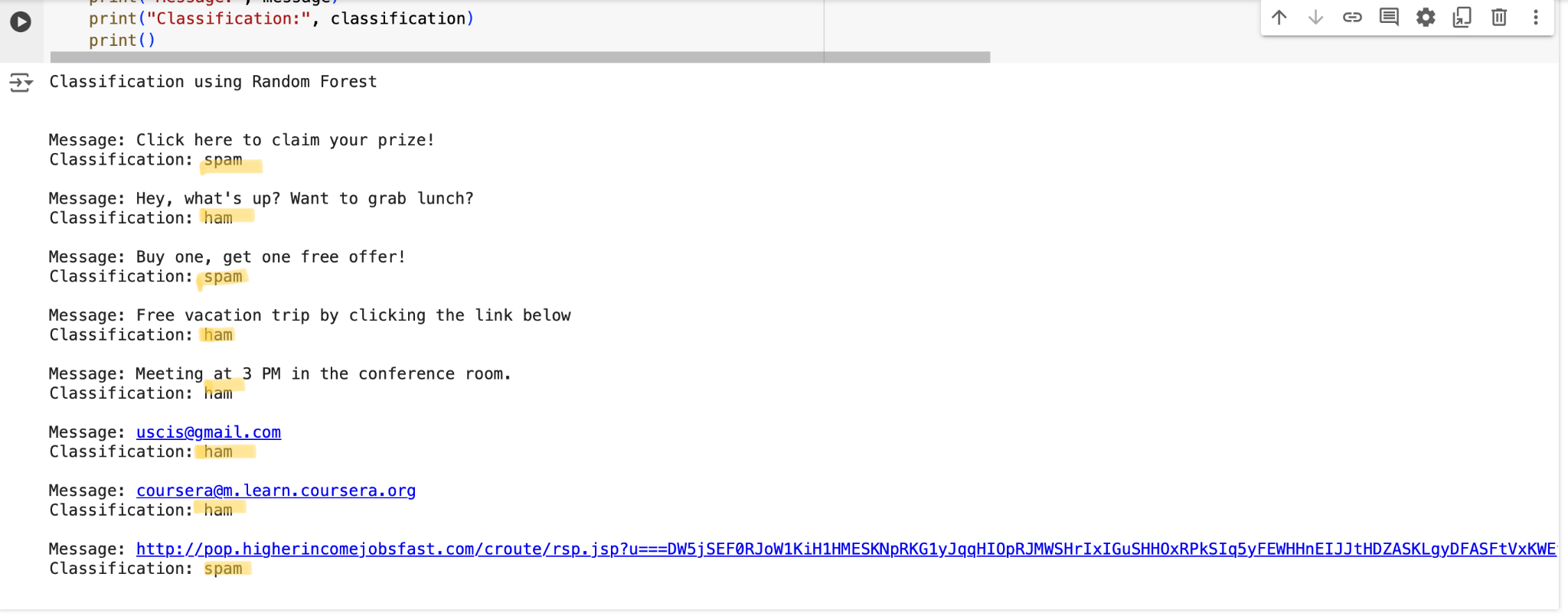
1. **TF-IDF Vectorization:**
   * Converts email text into numerical features based on word importance, making the text suitable for machine learning models.
2. **Model Training:**
   * Classifiers like **Naive Bayes**, **SVM**, and **Random Forest** are trained on these features to learn the distinctions between spam and ham emails.
3. **Email Classification:**
   * New emails are preprocessed, transformed into TF-IDF features, and classified by the trained models. The system predicts whether an email is spam or ham based on its content.

**Example:** A message like "URGENT!: Your Mobile No. was awarded a £2,000 Bonus Caller Prize!" would be processed and likely classified as spam, while a casual message like "Hey, what's up? Want to grab lunch?" would be classified as ham.

This process ensures accurate and effective classification of diverse real-world email content.







**Future work**

* **Advanced Feature Engineering:** To capture more subtle patterns in email text, investigate more advanced feature engineering techniques, such as word embeddings or deep learning-based methods.
* **Active Learning:** To reduce the quantity of labeled data needed for training while retaining high classification accuracy, use active learning algorithms to choose the most instructive emails for hand labeling.
* **Real-Time Detection:** Using streaming data processing techniques, create a real-time email spam detection system that can identify emails as spam or not as they are received.
* **Enhance the system's capacity:** To manage substantial email volumes effectively by optimizing its scalability and performance, especially in high-traffic email situations.

Challenges

* **Imbalanced Data:** Biassed models may result from an imbalanced amount of spam and non-spam emails in the dataset. To overcome this obstacle, methods like resampling or the application of suitable assessment measures are required.
* **Feature Selection:** It's critical to extract pertinent information from the email text that can be used to distinguish spam from legitimate emails. However, this process is difficult due to the sheer volume and variety of email content.
* **Computational Complexity:** To handle the computational load, processing and classifying a huge volume of emails in real-time calls for effective algorithms and infrastructure.
* **Privacy Issues:** When deploying spam detection technologies, privacy problems arise since email content frequently contains sensitive information. It's difficult to properly filter out spam while maintaining user privacy protection.

# 

Conclusion

The developed spam detection system demonstrated the capability to effectively classify emails as spam or ham using machine learning techniques. By addressing challenges related to data imbalance, textual noise, and feature extraction, and by employing robust evaluation metrics, we created a reliable and accurate spam detection solution. The comparative analysis of Naive Bayes, SVM, and Random Forest models provided valuable insights into their respective performances. Moving forward, incorporating advanced feature extraction techniques, real-time capabilities, and adaptive learning can further enhance the system’s effectiveness and adaptability to evolving spam tactics.

**Contributions**

Sharon- Data collection and project report

Jahnavi- Data preprocessing and project presentation

Vishnu Ravikumar- Model evaluation and project presentation

Narsimha reddy Bhogala- project report and model evaluation

References

1.Cormack, G. V., & Lynam, T. R. (2007). Email spam filtering: A systematic review. Foundations and Trends® in Information Retrieval, 1(4), 335-455.

2.Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. In AAAI Workshop on Learning for Text Categorization.

3.Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Paliouras, G. (2000). An evaluation of naive Bayesian anti-spam filtering. In Proceedings of the workshop on machine learning in the new information age (Vol. 66, pp. 9-17).

4.Carreras, X., & Marquez, L. (2001). Boosting trees for anti-spam email filtering. In Proceedings of the 16th International Conference on Machine Learning (ICML-2001) (Vol. 1, pp. 79-86).

5.Androutsopoulos, I., Paliouras, G., Michalopoulos, I., & Spyropoulos, C. D. (2010). Learning to filter unsolicited commercial email. ACM Computing Surveys (CSUR), 43(4), 25.

6.Osman, M. E. M., & Elragal, A. (2016). A review of email spam filtering techniques. Journal of Network and Computer Applications, 60, 82-101.