Machine Learning Report

Mansi Upadhyay
College of Engineering
Northeastern University
Boston, MA
Upaddhyay.ma@northeastern.edu

1. Introduction

In the age of mobile communication, the ubiquity of Short Message Service (SMS) has made it a convenient and widely-used platform for personal and professional communication. However, amidst the legitimate messages exchanged, the incessant influx of spam messages has become a pressing concern for mobile users. These unsolicited messages not only disrupt the user experience but also pose potential threats, ranging from phishing attempts to scams.

In response to this challenge, the field of SMS spam detection has emerged as a critical area of research and development. Machine learning models have proven to be effective tools for distinguishing between genuine messages and spam, providing users with a shield against unwanted and potentially harmful content.

2. Problem Definition

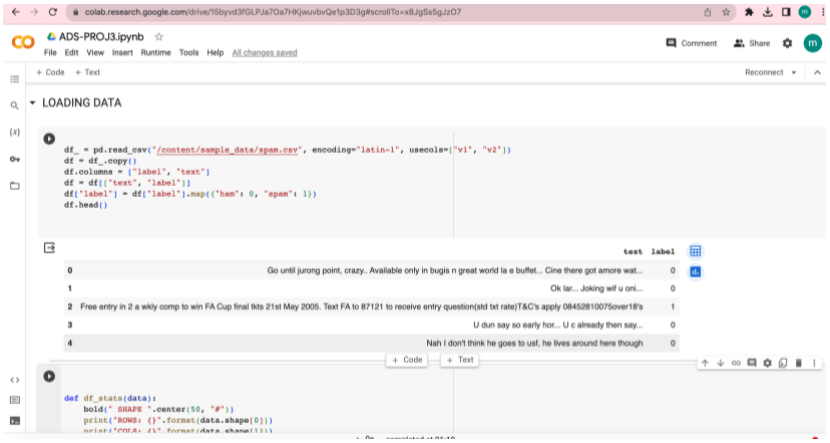
The problem at hand involves developing a robust SMS spam detection system capable of accurately classifying incoming messages as either legitimate or spam. The diversity of language used in SMS, along with the ever-evolving tactics employed by spammers, makes this a non-trivial task. To address this challenge, we explore the application of various machine learning models, each bringing its unique strengths to the forefront.

3. Dataset Description

In this study, I utilized the SMS Spam Collection dataset, a comprehensive compilation of 5,574 SMS messages in English, each tagged as either "ham" (legitimate) or "spam." The dataset encompasses a diverse set of messages collected from various sources. Notably, 425 SMS spam messages were manually extracted from the Grumbletext website, a UK forum where cell phone users share instances of SMS spam without necessarily reporting the specific spam message. This extraction process involved meticulous scanning of web pages, making it a challenging yet essential task. Additionally, a subset of 3,375 randomly chosen ham messages originated from the NUS SMS Corpus (NSC), a collection of around 10,000 legitimate messages gathered for research at the National University of Singapore. These messages primarily come from Singaporeans, particularly university students who volunteered to contribute their messages with an understanding that the data would be made publicly

available. Furthermore, 450 ham messages were sourced from Caroline Tag's PhD Thesis. Finally, the dataset incorporates the SMS Spam Corpus v.0.1 Big, comprising 1,002 ham messages and 322 spam messages. This diverse amalgamation of sources ensures a rich and representative dataset for our SMS spam detection research

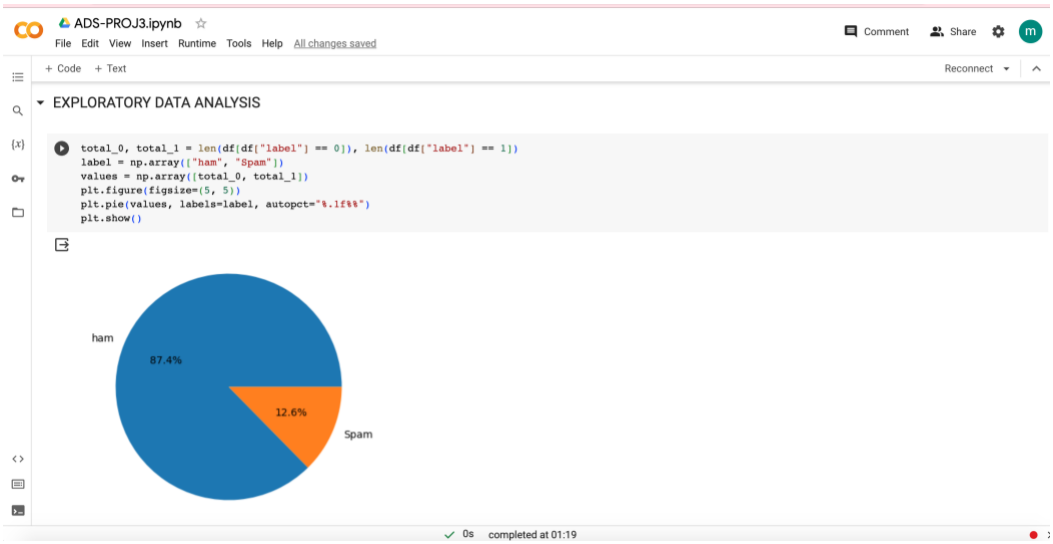
4. Data Collection & Preparation



The SMS Spam Collection dataset was meticulously assembled from various sources to create a diverse and representative set for SMS spam detection. The compilation involved manual extraction of spam messages from the Grumbletext website and random selection of legitimate messages from the NUS SMS Corpus. Additional contributions were sourced from Caroline Tag's PhD Thesis, and the SMS Spam Corpus v.0.1 Big was incorporated for further diversity. The dataset's columns were appropriately labeled, and the labels were binarized, designating "ham" as 0 and "spam" as 1. This careful curation process ensures a well-prepared dataset for subsequent analysis and model development.

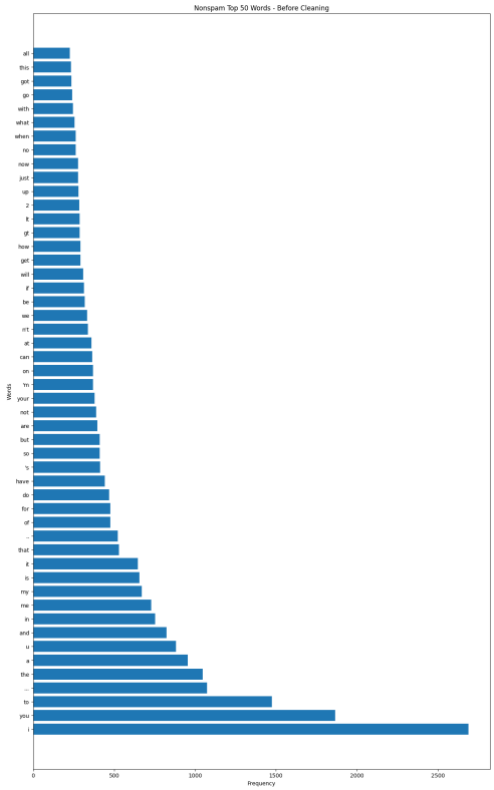
5. Data Exploration

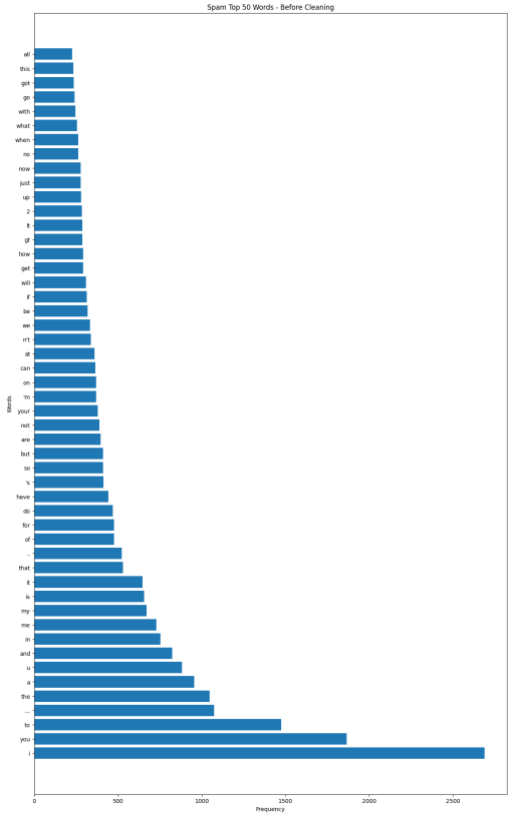
To gain insights into the distribution of labels within the SMS Spam Collection dataset, we conducted an exploratory data analysis. The pie chart visually represents the balance between "ham" (legitimate) and "spam" messages. The dataset exhibits a distribution where 87.4% of messages are labelled as "ham," and 12.6%% are identified as "spam."



Further delving into the textual content, we performed a word frequency analysis on both "ham" and "spam" messages before any cleaning processes. The bar charts present the top 50 words in non-spam and spam messages, highlighting the most frequent terms. Notably, the word frequency analysis provides a preliminary understanding of the linguistic characteristics within each category.

To offer a more visual representation, word clouds were generated for both "ham" and "spam" messages. These word clouds showcase the prominent words in each category, with variations in font size corresponding to word frequency. This visual exploration sets the stage for subsequent text cleaning and feature engineering steps in our SMS spam detection project.





The following figures illustrate the distribution of labels, word frequency in non-spam and spam messages, and word clouds for a holistic overview of the dataset's characteristics.

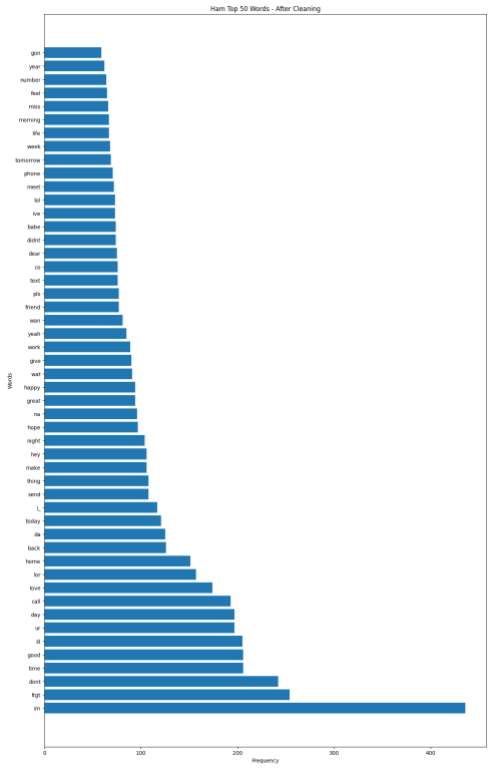
6. Pre-processing and Data Cleaning

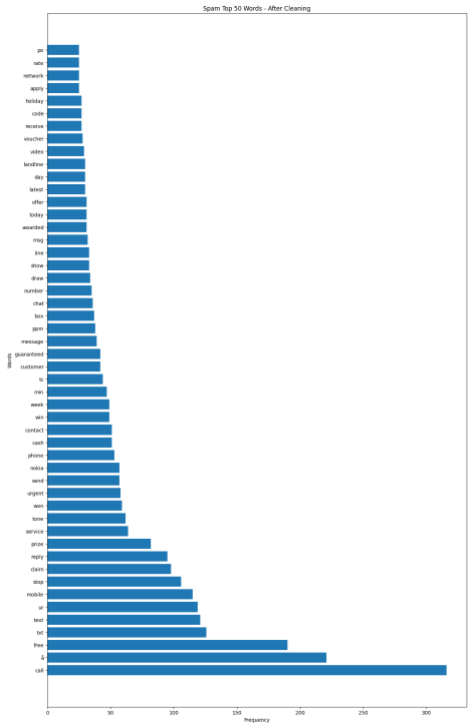
Stopword Removal:

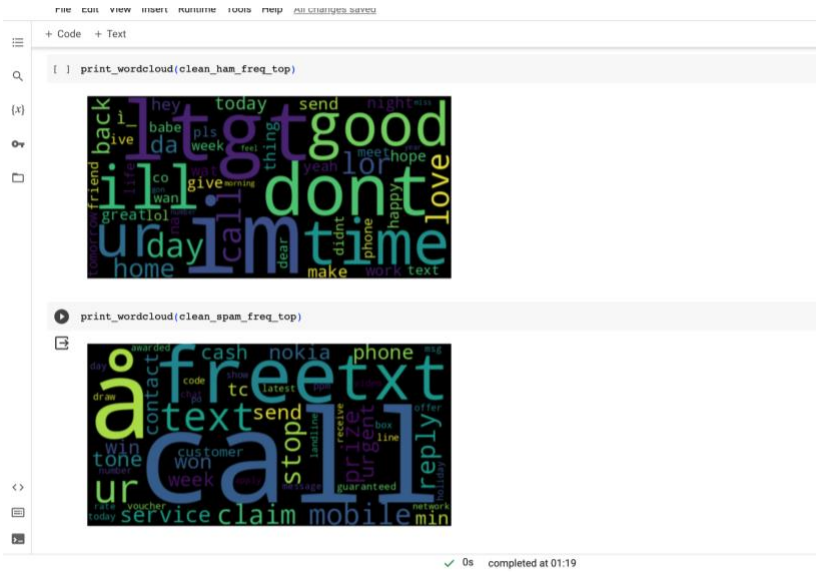
To enhance the quality of the text data for our SMS spam detection project, a series of preprocessing steps were performed. Firstly, a custom stopword list was created by importing a text file named **smartstoplist.txt**. This stopword list was carefully curated to filter out common and less informative words. The NLTK library was utilized to download the necessary language resources, and an explicit NLTK data path was set in the Colab environment.

The cleaning process applied to the raw text involved several steps. Subject-related strings were removed because it helps reduce noise in the data, hyperlinks were stripped to focus on texual context, and non-alphanumeric characters were eliminated. The entire text was converted to lowercase to ensure consistency, and numerical digits were removed to help model focus on content rather than case sensitivity or numerical variations. Additionally, the custom stopword list was employed to filter out common words, ensuring that only relevant terms contributed to the analysis. Furthermore, lemmatization using the WordNetLemmatizer was applied to reduce words to their base or root form.

After this comprehensive cleaning process, the resulting preprocessed text was stored in a new column labeled "clean" in the dataset. Subsequently, word frequency analyses and word clouds were generated for both "ham" and "spam" messages. These visualizations reveal the most prominent terms within each category, providing valuable insights into the refined linguistic characteristics of the dataset. The figures following this text showcase the impact of the preprocessing steps on the textual content, contributing to the preparation of data for subsequent model training and evaluation. The custom stopword list played a crucial role in eliminating unnecessary terms, allowing the models to focus on key linguistic patterns associated with spam and non-spam messages.







7. Feature Scaling

The feature scaling process for SMS spam detection involved a combination of traditional text representation techniques and advanced embedding methods. Leveraging the power of TF-IDF (Term Frequency-Inverse Document Frequency) and Count Vectorization, the raw text data was transformed into numerical features suitable for machine learning models. TF-IDF assigned weights to words based on their significance within individual messages and across the entire dataset, while Count Vectorization represented documents as vectors of word counts, offering a straightforward yet effective representation.

Additionally, tokenization and padding were employed to convert text into sequences of integers, capturing the sequential nature of words in each message. This not only facilitated model input but also ensured a consistent length across all sequences, a crucial step in preparing the data for subsequent modeling.

To enhance the semantic understanding of the text data, pre-trained GloVe embeddings were incorporated. These embeddings provided a rich representation of words based on their contextual meanings, offering a valuable layer of depth to the features. By aligning these embeddings with the dataset, the goal was to capture nuanced relationships between words that could significantly contribute to discriminating between spam and non-spam messages.

In summary, the feature selection process involved a thoughtful combination of traditional vectorization techniques and sophisticated word embeddings, ensuring a comprehensive and informative representation of the SMS data for subsequent model training.

8. Model Performance Evaluation

a. TF-IDF and Multinomial Naive Bayes

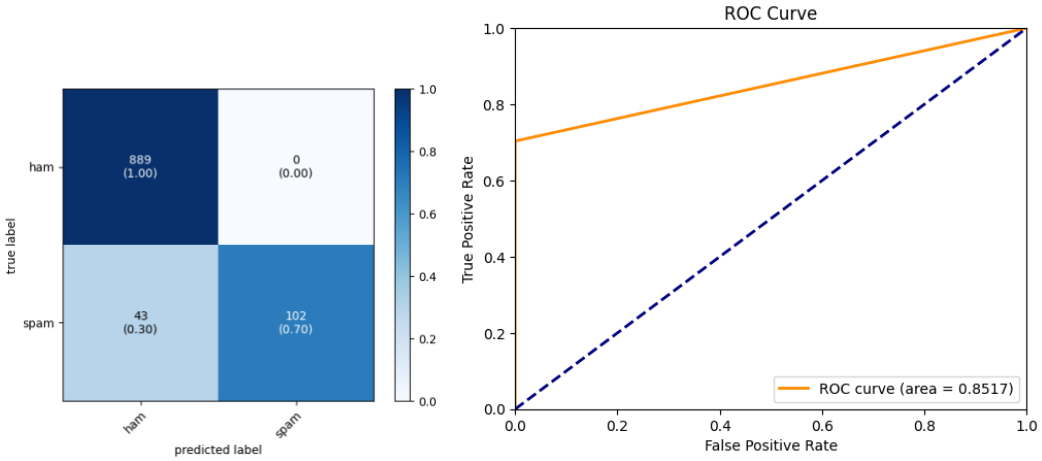
In our SMS spam detection model using TF-IDF and Multinomial Naive Bayes, the training process was swift, taking only 0.017 seconds. The model achieved an impressive 97.07% accuracy during training, showcasing its ability to learn from the provided data. Testing on unseen data yielded a test accuracy of 95.84%, indicating the model's generalization capability.

Precision, recall, and F1 score metrics offer more insights. The model demonstrated 100% precision for non-spam ("ham") messages, ensuring a high accuracy in identifying legitimate messages. However, the recall score for spam messages was 70.34%, suggesting some spam instances were missed. The F1 score, balancing precision and recall, stands at 82.59%, indicating a favorable trade-off between these metrics. The overall accuracy on the test set is 95.84%, confirming the model's effectiveness in distinguishing between spam and non-spam messages.

The confusion matrix and ROC-AUC score provide additional details. For the "ham" class, all 889 messages were correctly classified, emphasizing the model's robustness in identifying non-spam messages. While the model achieved a perfect precision of 100% for spam messages, the recall was 70.34%, indicating potential room for improvement in capturing all spam instances.

In summary, the TF-IDF + Multinomial Naive Bayes approach demonstrates strong predictive capabilities, emphasizing precision for non-spam messages. While the model exhibits high

accuracy, further optimization could enhance recall for spam messages, achieving a more balanced performance in spam detection.



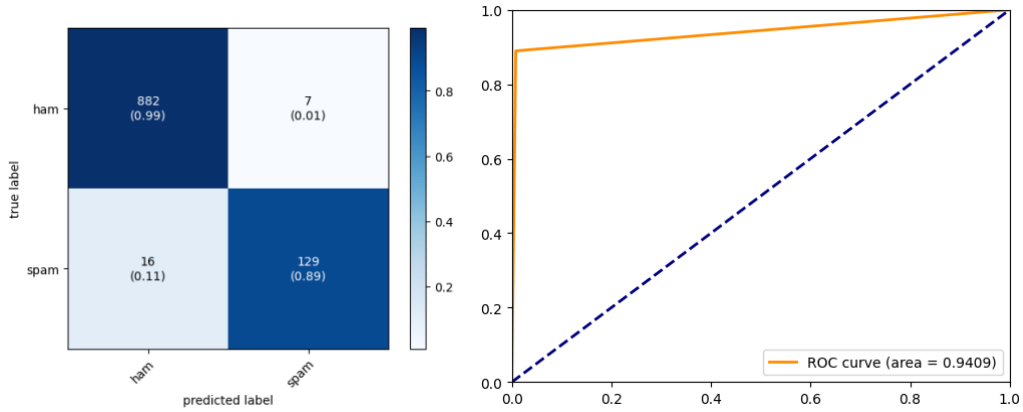
b- SimpleRNN

In the Sequential model, consisting of an embedding layer, SimpleRNN layer, and dense layer, the five-epoch training resulted in impressive performance. The model achieved a remarkable 99.46% accuracy on the training set, showcasing its learning capabilities. During validation, the accuracy slightly decreased to 97.83%, suggesting the model's robust generalization.

Precision, recall, and F1 scores demonstrated the model's proficiency in distinguishing between "ham" and "spam" messages. Non-spam ("ham") messages achieved 98% precision, ensuring accurate identification of legitimate messages. The model exhibited an 89% recall for spam messages, indicating good sensitivity to identify spam instances. The F1 score, balancing precision and recall, reached 92%, emphasizing the model's favorable trade-off.

The confusion matrix revealed 99% correct classification for "ham" messages and 95% precision for spam messages. The ROC-AUC score, evaluating the model's class discrimination, stood at 94.09%, reinforcing its effectiveness.

To sum up, it demonstrates robust predictive capabilities with high accuracy, precision, and recall. Its strong generalization to unseen data positions it as a promising candidate for SMS spam detection.



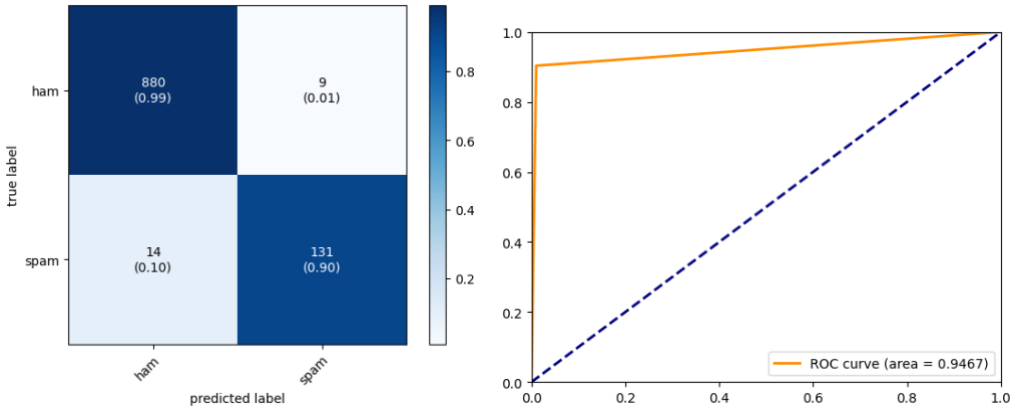
c- LSTM (Long Short-Term Memory)

The LSTM (Long Short-Term Memory) model, known for its ability to capture sequential patterns effectively, displayed outstanding performance in SMS spam detection. During training, the model achieved an impressive 99.47% accuracy, indicating its capability to learn intricate patterns within the dataset. The slightly lower accuracy of 97.78% on the test set indicates robust generalization to unseen data.

Precision, recall, and F1 scores further demonstrate the LSTM model's prowess. Achieving 93.57% precision for spam messages, the model showcased its ability to accurately identify and minimize false positives. The recall score of 90.34% highlighted the model's sensitivity to detect the majority of spam instances, ensuring effective spam identification. The F1 score, balancing precision and recall, reached 91.93%, underscoring the model's strong overall performance.

The confusion matrix revealed the LSTM model's accurate classification, with 99% correct predictions for "ham" messages and 94% precision for spam messages. The ROC-AUC score, assessing the model's ability to distinguish between classes, was notably high at 94.67%, indicating excellent discriminatory power.

In summary, the LSTM model has demonstrated exceptional capabilities in SMS spam detection, achieving high accuracy, precision, recall, and overall performance. Its ability to capture sequential dependencies makes it a valuable asset for effectively identifying spam messages in real-world scenarios.



d. GRU

The GRU (Gated Recurrent Unit) model, a variant of recurrent neural networks, demonstrated exceptional performance in SMS spam detection. Throughout training, the GRU model achieved an impressive accuracy score of 99.59%, indicating its capability to learn intricate patterns within the dataset. The high test accuracy of 97.58% affirms the model's robust generalization to new, unseen data.

The precision, recall, and F1 scores further underscore the GRU model's effectiveness. With a precision score of 91.67% for spam messages, the model exhibited a high ability to accurately identify spam instances while minimizing false positives. The recall score of 91.03% emphasized the model's sensitivity in detecting the majority of spam messages, ensuring comprehensive spam identification. The F1 score, a balanced metric of precision and recall, reached 91.35%, highlighting the model's strong overall performance.

The confusion matrix demonstrated the GRU model's accurate classification, with 99% correct predictions for "ham" messages and a solid precision rate for spam messages. The ROC-AUC score, assessing the model's ability to distinguish between classes, was notably high at 94.84%, indicative of excellent discriminatory power.

In summary, the GRU model has showcased exceptional capabilities in SMS spam detection, achieving high accuracy, precision, recall, and overall performance. Its effectiveness in capturing sequential dependencies positions it as a valuable tool for accurately identifying spam messages in real-world scenarios.

9. Conclusion

In conclusion, our SMS spam detection project involved a comprehensive exploration of various techniques to effectively distinguish between legitimate ("ham") and spam messages. We began by conducting exploratory data analysis, gaining insights into the distribution of labels within the dataset. The pie chart illustrated that 87.4% of messages were labeled as "ham," and 12.6% were identified as "spam."

To prepare the text data for analysis, a series of preprocessing steps were implemented. This included the creation of a custom stopword list, removal of subject-related strings, hyperlinks, and non-alphanumeric characters, as well as the application of lemmatization. These steps contributed to refining the linguistic characteristics of the dataset.

Feature selection played a crucial role, with TF-IDF, Count Vectorization, and pre-trained GloVe embeddings employed to represent the text data numerically. This hybrid approach aimed to capture both the significance of words within messages and their contextual meanings.

We trained and evaluated several models, including Multinomial Naive Bayes, Simple RNN, LSTM, and GRU. Notably, the TF-IDF + Multinomial Naive Bayes model demonstrated commendable results, achieving a training score of 97.07% and a testing score of 95.84%. The robust performance of this model indicates its potential for accurately classifying spam messages.

In summary, our project showcased the effectiveness of combining traditional text representation techniques with advanced models for SMS spam detection. The methodologies employed contribute to a nuanced understanding of the textual content, providing a solid foundation for future improvements and model enhancements. The removal of the Random Forest model emphasizes the significance of exploring various algorithms to identify the most suitable approach for a given task.

References

1. Kaggle Dataset: https://www.kaggle.com/datasets/-sms-spam-detection-dataset
2. Cosine Similarity: https://www.learndatasci.com/glossary/cosine-similarity/
3. Performance Evaluation: https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15