Project 5 Report

Group 9: Preston Hall, Landon Taylor, Md. Robiul Islam

Introduction

The aim of this project was to analyze text message data labeled spam or not spam. We then trained a Multinomial Naive Bayes Classifier to make predictions on whether or not a message is spam. This type of analysis has many important uses for different stakeholders. Companies who create spam detection software may find our analysis particularly useful, as well as business personnel who rely on texting to discuss important work events, or even just casual users who would rather not see spam messages. Through this analysis we wanted to find features that would be the best predictors of spam messages, including the most common words, message length, and other factors.

Our Google slides can be found here, and our GitHub repository can be found here.

Dataset

Our dataset was obtained from the UC Irvine machine learning repository archive, found here. According to the researchers, the spam messages in the dataset were found on the Grumbletext web forum, and the other messages were extracted from NUS SMS Corpus, a separate dataset compiled at the National University of Singapore's Computer Science Department. The dataset we used contains a series of SMS message data, in the form of a label of either "spam" or "ham", followed by a message that presumably fits that description. The dataset has a total of 5574 SMS messages, of which about 13.4% was labeled spam. The dataset comes with no heading.

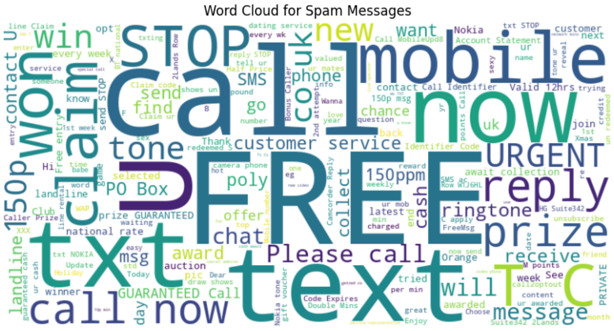
Data Cleaning and Preparation: At first it was converted to a DataFrame with two columns named label and message. After that label values ['ham', 'spam'] were mapped to numeric value 0 or 1 as classification type.

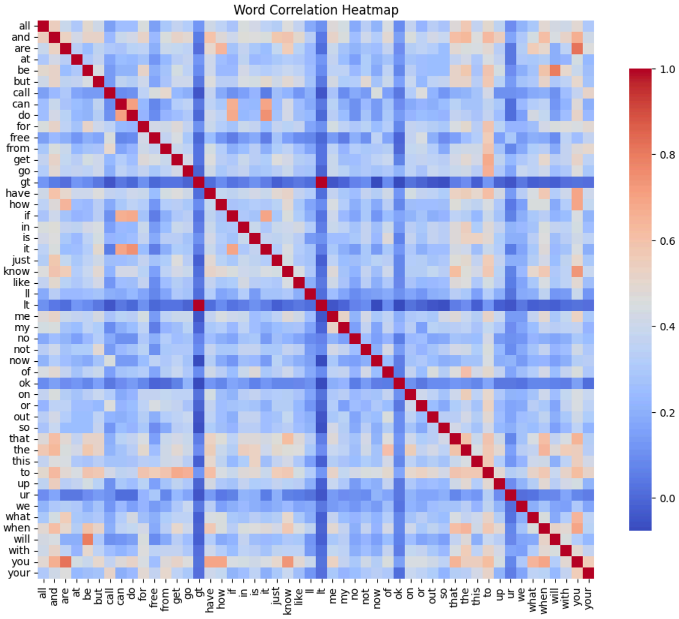
Feature Extraction: For feature extraction and converting the messages to quantitative values we used TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF is a statistical measure that quantifies the importance of a term in a document relative to a corpus of documents.

Analysis Technique

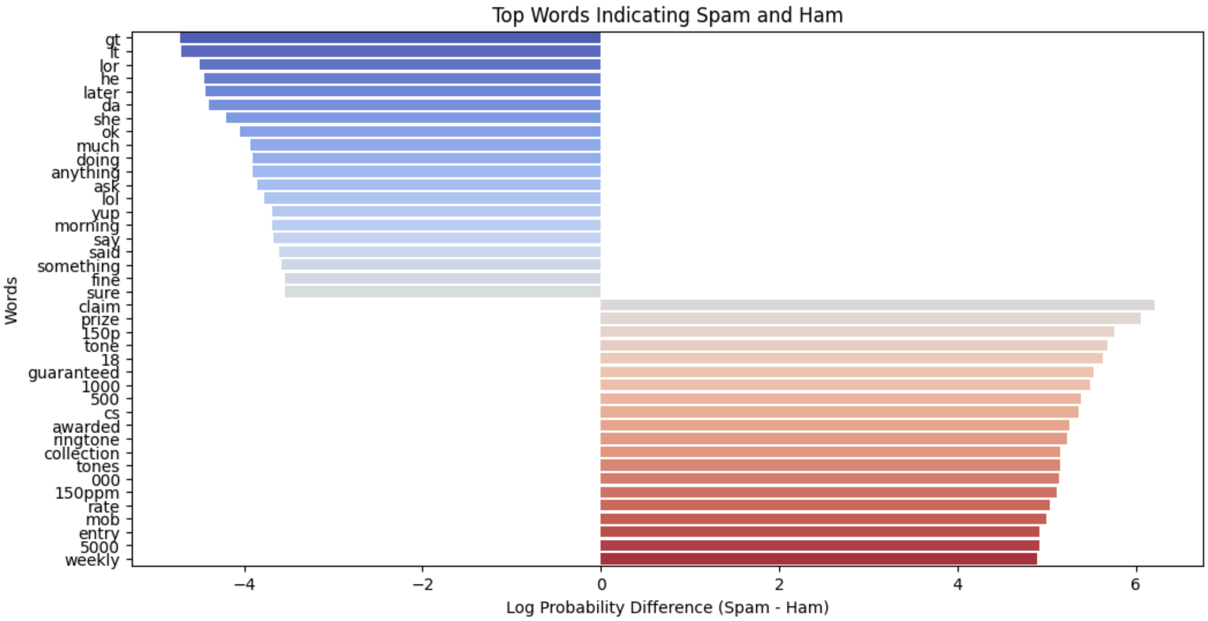
After preparing our dataset, we partitioned it into a training set (70% of the data) and a testing set (30% of the data). We found that the dataset contained many words with a high correlation to other words, and a sample of our findings appears to the right. Despite this, we believe that a Naive Bayesian approach is a promising method to apply to our dataset.

After fitting the data to the training portion of the model, we made predictions for the classification of messages in the testing portion. We found that our predictions are highly reliable, since spam messages tend to contain similar sets of words. Supporting this theory, we created the following word cloud indicating the most popular words in spam messages.

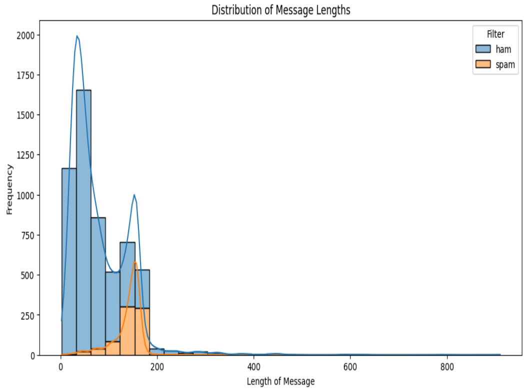




We further confirmed by intuition that our method was effective, as the top words indicating spam are words we tend not to use in our own conversations, and we have seen many of them in spam messages in personal applications.



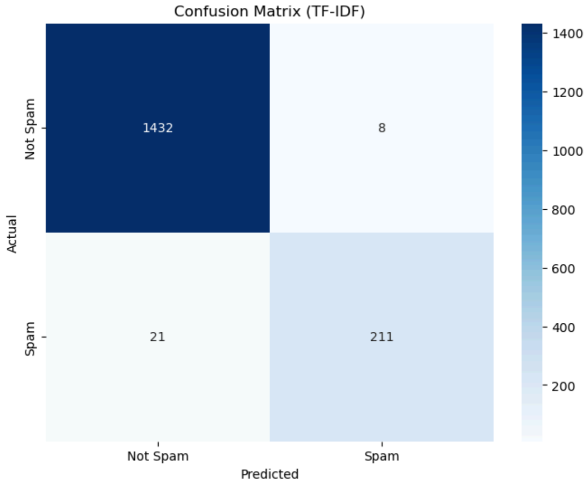
We considered other factors, such as message length, but we found that both spam and ham messages see a similar pattern for message lengths, around 180 characters. Very short (under 100 characters) messages tended to be ham, however.



Results

The Naive Bayes classifier was applied to classify SMS messages as either spam or ham (not spam), and the model demonstrated excellent performance across several evaluation metrics.

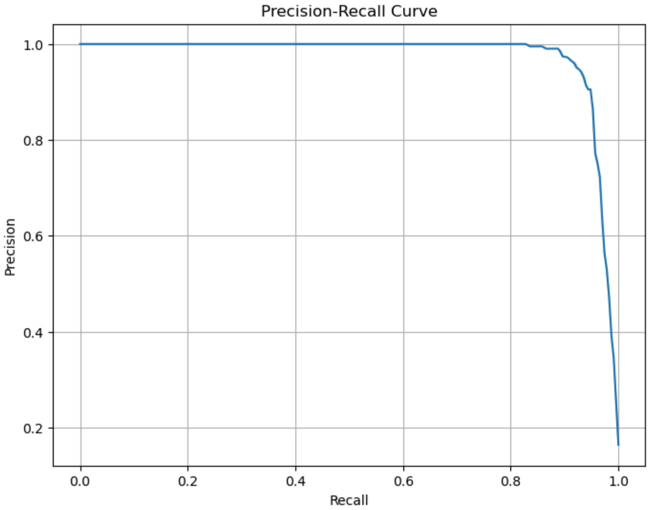
Classification Metrics



The classification report provides detailed insights into the model’s performance:

- **Class 0 (Ham)**:
  - Precision: 0.99
  - Recall: 0.99
  - F1-Score: 0.99
  - Support: 1,440
- **Class 1 (Spam)**:
  - Precision: 0.96
  - Recall: 0.91
  - F1-Score: 0.94
  - Support: 232

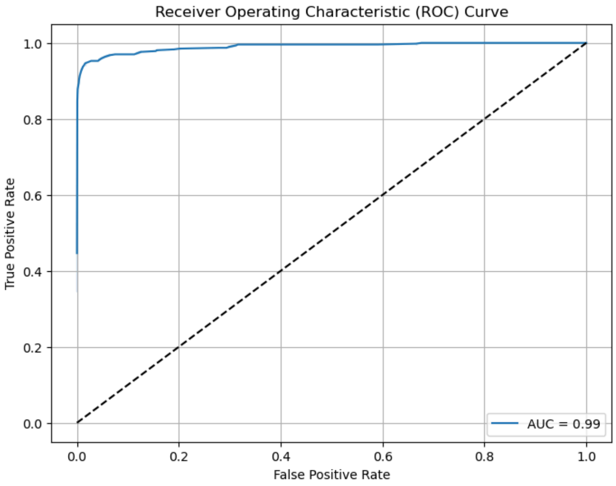
The overall **accuracy** of the model was **98%**, reflecting its ability to correctly classify the majority of the messages. The macro average F1-score of 0.96 and weighted average F1-score of 0.98 further indicate the model's strong generalization across both classes.



Confusion Matrix

The confusion matrix provided additional insight into the classifier's performance:

The model exhibited a low number of misclassifications, with only 8 legitimate (ham) messages misclassified as spam, and 21 spam messages misclassified as ham. This supports the high precision and recall scores observed, especially for the spam class.

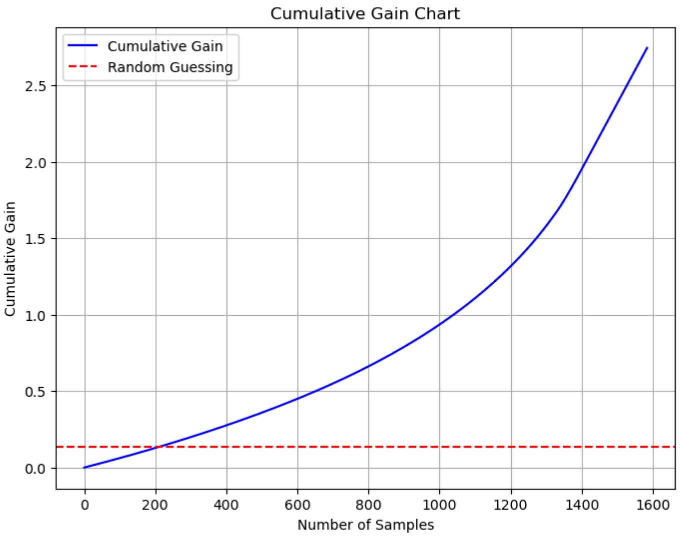


Precision-Recall Curve

The Precision-Recall (PR) curve showed the model's strong ability to balance precision and recall. The precision remained high (close to 1.0) for most values of recall, with a slight trade-off only at the highest recall values. This indicates that the classifier maintains a high level of accuracy in classifying both spam and ham, even as recall increases.

ROC Curve

The provided receiver operating characteristic (ROC) curve visually represents the trade-off between true positive rate (TPR) and false positive rate (FPR) for a classification model. It helps assess the model's performance in distinguishing between positive and negative instances while considering the balance between sensitivity and specificity.
AUC (Area Under the Curve) score of 99% Quantifies the overall performance of the model.



Lift Chart

Comparison to Random Guessing: The model significantly outperformed random guessing as the amount of training data increased, demonstrating its effectiveness.

Alternative Approach of Feature Selection: Without taking all the features into account it could be done that most frequent terms over the whole document have been removed and also rare terms could be removed for selecting important features. Then use these features to classify the message. Although we have tried with this approach but the test result was a little bit lower.