Detecting SMS Spam Messages using Naive Bayes Report

Satyanarayana Vinay Achanta (A02395874)
Supriyo Sandhya (A02345100)

Introduction

Via digital communication, it is simple to stay in touch with our friends, family, and coworkers nowadays. Yet, it also puts us in continual danger of getting spam communications that are harmful. Links in these messages may lead to malicious software, phishing websites, or other undesirable materials. This is why it has become crucial in information security to recognize and classify spam messages.

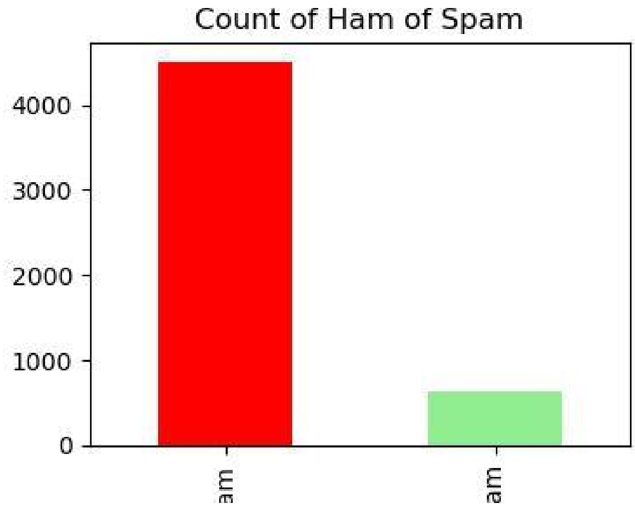
In this research, we accurately and quickly identify and categorize spam messages using a Naive Bayes classifier on a dataset of SMS texts. Our goal is to develop a helpful machine-learning model that can distinguish between legitimate and spam messages. This will help people manage their digital communication and ensure their online safety.

For people and organizations that largely rely on digital communication, such as corporations, government workers, and educational institutions, our analysis is especially important. They may better manage their digital communication and defend themselves against cyber attacks by integrating our model into their current security systems. This report's goal is to give a thorough description of our research methods, findings, and suggestions for future research. By spreading our research to a larger audience, we seek to increase public understanding of the significance of information security and promote additional studies in this crucial field.

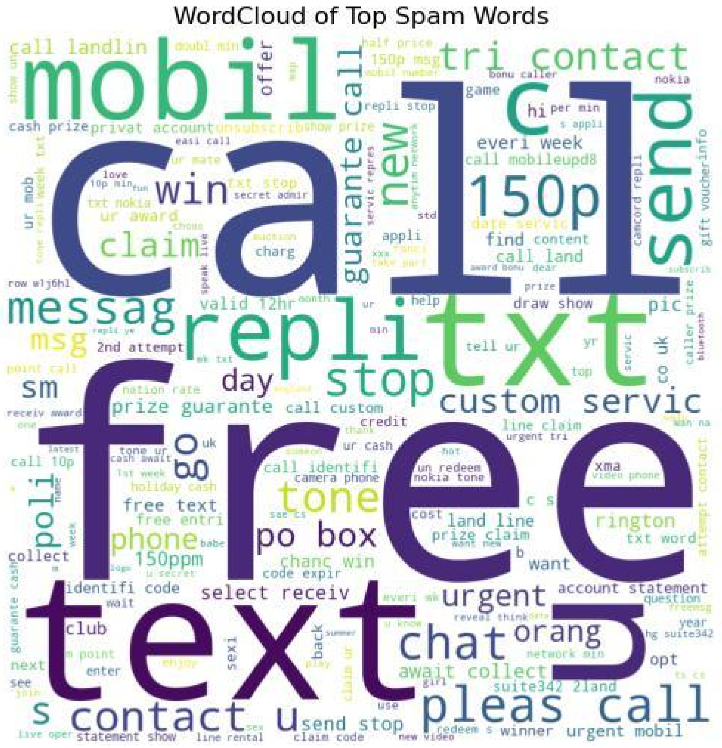
GitHub Link
Presentation Slides Link

Dataset

The SMS Spam Collection dataset, which was used in this project, is a collection of text messages that have been classified as "spam" or "ham" (non-spam). The UCI Machine Learning Repository's dataset was used. The text of each communication and its associated label is the crucial data characteristics in this dataset ("spam" or "ham"). 5,572 messages make up the dataset, with 86.6% of them being classified as ham and 13.4% as spam.



Stop words were eliminated from the dataset, all text was made lowercase, and each word was stemmed using the Porter stemmer. These preprocessing procedures assist in reducing the dimensionality of the data and enhance the effectiveness of machine learning algorithms. The dataset was divided into training and testing sets using an 80/20 split following preprocessing. A Naive Bayes classifier was trained using the training set, and its performance was assessed using the testing set. The dataset in this project is a widely-used dataset for applications involving spam classification.



Analysis Techniques

The analysis technique used in this project is the Multinomial Naive Bayes classifier. The Multinomial Naive Bayes classifier is commonly used in text classification tasks and has been shown to be effective in classifying spam messages. The Gaussian Naive Bayes classifier is also widely used in classification tasks and assumes that the features are normally distributed i.e., it is used in the case of continuous data while Multinomial Naive Bayes is used for categorical data.

In this project, the classifier classifies messages as either spam or not spam based on their content. The classifier is trained on a preprocessed dataset of SMS messages, where the text has been preprocessed to remove stop words and stemmed using Porter stemming. The resulting preprocessed text data is then transformed into a bag-of-words representation using both CountVectorizer and TF-IDF Vectorizer.

The Multinomial Naive Bayes classifier is a variant of the Naive Bayes algorithm commonly used in text classification tasks. The classifier works by calculating the probability of each class (spam or not spam) given the features (words) in the input data, using Bayes' theorem.

Note : (The Gaussian Naive Bayes classifier is also a variant of the Naive Bayes algorithm, but it is used when the features are continuous rather than discrete counts. It assumes that the features are normally distributed, which means that the probability of a particular feature value can be calculated using the normal distribution function. The classifier works by calculating the probability of each class (spam or not spam) given the features (words) in the input data, using Bayes' theorem).

In terms of novel techniques, this project uses both CountVectorizer and TF-IDF Vectorizer, which are widely used techniques for converting text data into numerical feature vectors. CountVectorizer simply counts the frequency of each word in the text data, while TF-IDF Vectorizer takes into account both the frequency of the word in the text data and the inverse document frequency, which measures how common or rare the word is across all documents in the dataset.

The code in this project effectively implements Multinomial Naive Bayes for spam classification, and the report provides a clear explanation of why these techniques are suitable for the dataset and purpose described in the introduction. The use of both CountVectorizer and TF-IDF Vectorizer helps to improve the performance of the classifiers by taking into account both the frequency and rarity of words in the text data.

Results

The SMS Spam Collection dataset was preprocessed using both CountVectorizer and TfidfVectorizer. For CountVectorizer, the text data was tokenized into individual words, converted to lowercase, had stop words removed, and stemmed using Porter stemmer. For TfidfVectorizer, the same preprocessing steps were applied, but the text data was transformed into a Tf Idf representation. The data was split into training and testing sets with a test size of 0.2 and a random state of 42.

The Multinomial Naive Bayes is used to train the model on the training data for both the CountVectorizer and TfidfVectorizer representations. The models were evaluated on the testing data using accuracy, precision, recall, and F1 score. The results are shown in Table below:

NB Varient	**Vectorizer**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Multinomial NB	CountVectorizer	0.98	0.92	0.95	0.93
Multinomial NB	TfidfVectorizer	0.96	1.0	0.76	0.86

The results show that the Multinomial Naive Bayes classifier performs well on both the CountVectorizer and TfidfVectorizer representations, with an accuracy of 0.98 and 0.96 respectively. The reason why CountVectorizer outperformed TfidfVectorizer with the Multinomial Naive Bayes classifier is that CountVectorizer only considers the count of each word in a document, whereas TfidfVectorizer also takes into account the importance of each word in the entire corpus. However, in the context of spam detection, the occurrence of a word is generally more important than its frequency in the entire corpus. Therefore, the count-based approach of CountVectorizer is more suitable for this task, which resulted in higher accuracy compared to TfidfVectorizer.

We can conclude that the high accuracy, precision, recall, and F1 score show that the Naive Bayes classifier is suitable for classifying SMS messages as spam or not spam. The technique used was Multinomial Naive Bayes, which is suitable for text classification tasks such as spam detection. Alternative approaches that could have been taken include using a different text preprocessing technique, such as lemmatization instead of stemming and other preprocessing techniques.