EMAIL/SMS SPAM CLASSIFICATION

Submitted by:

Ibrahim Abdul Shukoor

Internship – 30

ACKNOWLEDGMENT

I want to express my gratitude to my mentor, Mr. Kashif Mohd, who always supported me and helped me develop my talents. I want to thank Flip Robo Technologies for providing me with a platform to learn, comprehend, and carry out various projects like this one. Finally, I would want to express my sincere gratitude to Data Trained for teaching me cutting-edge Python, Statistics, and machine learning approaches.

INTRODUCTION

Email and SMS spam classification is important because it helps to filter out unwanted and potentially harmful messages from a person's inbox or messaging app. Without spam classification, a person's inbox or messaging app could become flooded with spam messages, which can be annoying and time-consuming to sort through. In addition, spam messages can sometimes contain malicious links or attachments that can pose a security threat if clicked on or opened. By classifying spam messages and separating them from legitimate messages, individuals and organizations can protect themselves and their devices from these threats. Spam classification can also help to protect privacy by preventing sensitive information from being shared with unauthorized parties.

I. Business Problem Framing

The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged according being ham (legitimate) or spam. The objective of the project is to processes the SMS/Emails and build an NLP model that could classify the messages as spam or ham (legitimate).

II. Conceptual Background of the Domain Problem

Following are the study's primary goals:

a. To understand the data
b. To clean the data (remove unwanted text, punctuations, lemmatizing etc…)
c. To visualize different traits that can have an impact on the target variable.
d. To create machine learning model that can predict the ratings
e. To select the ideal model and use it with the test data.

III. Review of Literature

Machine Learning: With the use of machine learning (ML), which is a form of artificial intelligence (AI), software programs can predict outcomes more accurately without having to be explicitly instructed to do so. In order to forecast new output values, machine learning algorithms use historical data as input. The kind of data that data scientists wish to predict determines the kind of algorithm they use.

The three categories of machine learning algorithms are:

i. Supervised Learning: In supervised learning, data scientists describe the variables they want the algorithm to look for connections between and provide the algorithms with labelled training data. The algorithm's input and output are both described.
ii. Unsupervised learning: Algorithms trained on unlabelled data are used in this sort of machine learning. The algorithm searches through data sets in search of any significant relationships. Both the input data that algorithms use to train and the predictions or suggestions they produce are predefined.
iii. Data scientists generally employ reinforcement learning to instruct a computer to carry out a multi-step procedure for which there are set rules. An algorithm is programmed by data scientists to fulfil a goal, and they provide it with positive or negative feedback as it determines how to do so. However, the algorithm typically chooses the course of action on its own.

An Artificial Neural Network (ANN): is a computational model that is inspired by the structure and function of the human brain. It consists of interconnected "neurons" that can process and transmit information. ANNs can be trained to perform a variety of tasks, including classification. In the context of spam classification, an ANN could be trained to identify patterns and features in emails that are indicative of spam and use this information to classify new emails as either spam or not spam. ANNs are particularly useful for tasks that require the processing of large amounts of data and the identification of complex patterns.

Naive Bayes Classifier: is a machine learning algorithm that is used for classifying items into one of two categories (e.g., spam or not spam). It is

based on the idea that certain words or features are more likely to occur in spam emails than in legitimate emails (also known as "ham"). The Naive Bayes classifier uses the probabilities of these words or features to determine the likelihood that a given email is spam.

IV. Motivation for the Problem Undertaken

There are several reasons to build a spam classification model:

- **To improve the user experience**: Spam emails can be annoying and time-consuming to sort through, so by building a spam classification model, we can help users to more easily filter out unwanted messages and focus on the emails that are most important to them.
- **To protect against security threats**: Spam emails can sometimes contain malicious links or attachments that can pose a security threat if clicked on or opened. By building a spam classification model, we can help to protect users from these threats.
- **To protect privacy**: Spam emails can sometimes contain sensitive information that is intended for someone else, or that has been obtained through nefarious means. By building a spam classification model, we can help to protect users' privacy by preventing sensitive information from being shared with unauthorized parties.
- **To improve email delivery rates**: If a significant proportion of emails sent to a particular email address are classified as spam, it can affect the delivery rate of legitimate emails. By building a spam classification model, we can help to improve the delivery rate of legitimate emails by accurately identifying and filtering out spam emails.
- **To reduce costs**: Spam emails can take up valuable resources, such as storage space and processing power. By building a spam classification model, we can help to reduce the costs associated with managing and storing large amounts of spam emails.

Analytical Problem Framing

I. Mathematical/ Analytical Modelling of the Problem

There are several methods for data analysis and model building, but the two that are used most frequently are as follows:

- Supervised learning, which includes classification and regression models.
- Unsupervised learning, which includes association rules and clustering methods

Classification Model: Classification models are used to look at how different variables relate to one another. When determining which independent factors have the most impact on dependent variables, classification models are frequently utilized to gather crucial information.

II. Data Sources and their formats

The data has been extracted from the Grumbletext Web site; It includes a collection of 5573 rows SMS spam messages. A subset of 3,375 SMS randomly chosen ham messages were extracted from NUS SMS Corpus (NSC) which is a dataset of about 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore.

Format: Comma Separated Values (CSV).

Data Types:

- object – 2

III. Data Pre-processing Done

The following actions were taken during the data pre-processing:

i. Three “Unnamed” columns were dropped from the dataset for containing 100% null values.
ii. Messages were cleaned:
    a. Emails were replaced with “email”
    b. Websites were replaced with “website”
    c. Currencies were replaced with “currency”

d. Removed alphanumeric characters
e. Phone numbers were replaced with “phonenumber”
f. Cleaned additional things such as trailing, leading white spaces, removed stop words.

iii. Tokenized and lemmatized the sentences.

iv. Performed Sentiment analysis (i.e; added Polarity and Subjectivity columns)

```
# Replace email addresses with 'email'
df['v2'] = df['v2'].str.replace(r'^.+@[^\.].*\.[a-z]{2,}$',
                                'email')

# Replace URLs with 'url'
df['v2'] = df['v2'].str.replace(r'^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\S*)?$',
                                'url')

# Replace money symbols with 'currency'
df['v2'] = df['v2'].str.replace(r'£|\$', 'currency')

# Replace 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumber'
df['v2'] = df['v2'].str.replace(r'^\(?[\d]{3}\)?[\s-]?[\d]{3}[\s-]?[\d]{4}$',
                                'phonenumber')

# Replace numbers with 'number'
df['v2'] = df['v2'].str.replace(r'\d+(\.\d+)?', 'number')

# Remove punctuation
df['v2'] = df['v2'].str.replace(r'[^\w\d\s]', ' ')

# Replace whitespace between terms with a single space
df['v2'] = df['v2'].str.replace(r'\s+', ' ')

# Remove leading and trailing whitespace
df['v2'] = df['v2'].str.replace(r'^\s+|\s+?$', '')

# removing escapes, tab spaces
df = df.replace(to_replace=[r"\\t|\\n|\\r","\t|\n|\r"], value=["",""], regex=True)
```

```
# remove stopwords
stop_words = set(stopwords.words('english') + ['u', 'ü', 'ur', '4', '2', 'im', 'dont', 'doin', 'ure','à'])
df['v2'] = df['v2'].apply(lambda x: ' '.join(term for term in x.split() if term not in stop_words))


# lemmatizing the rest of the data
w_tokenizer = nltk.tokenize.WhitespaceTokenizer()
lemmatizer = nltk.stem.WordNetLemmatizer()

def lemmatize_text(text):
    return [' '.join(str(lemmatizer.lemmatize(w)) for w in w_tokenizer.tokenize(text))]

df['Lemmatized Text'] = df.v2.apply(lemmatize_text)
df['Lemmatized Text'] = [str(x).strip("'[]'") for x in df['Lemmatized Text']]


df = df.drop('acc_punct_dict',axis=1)

# removing non-ascii words
df['Lemmatized Text'] = df['Lemmatized Text'].str.encode('ascii','ignore').str.decode('ascii')

# removing alpha-numeric characters
def cleanse(word):
    rx = re.compile(r"\D*\d")
    if rx.match(word):
        return ''
    return word
def remove_alphanumeric(strings):
    nstrings = [" ".join(filter(None, (
    cleanse(word) for word in string.split())))
                for string in strings.split()]
    str1 = ' '.join(nstrings)
    return str1

df['Lemmatized Text']= df['Lemmatized Text'].apply(remove_alphanumeric)
```

Figure 1: Data Pre-processing

I. Hardware/Software and Tools

i. Jupyter Notebook
ii. NumPy
iii. Pandas
iv. Matplotlib
v. Seaborn
vi. nltk
vii. string
viii. Regular Expressions
ix. TextBlob
x. Counter

Model/s Development and Evaluation

I. Testing and Evaluated (Algorithms)

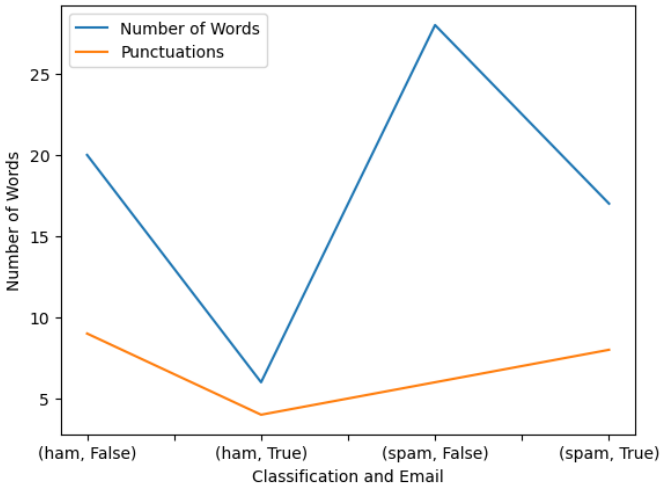
i. Naiver Bayes Classifier (98.42% accuracy)

II. Key Metrics for success in solving problem under consideration

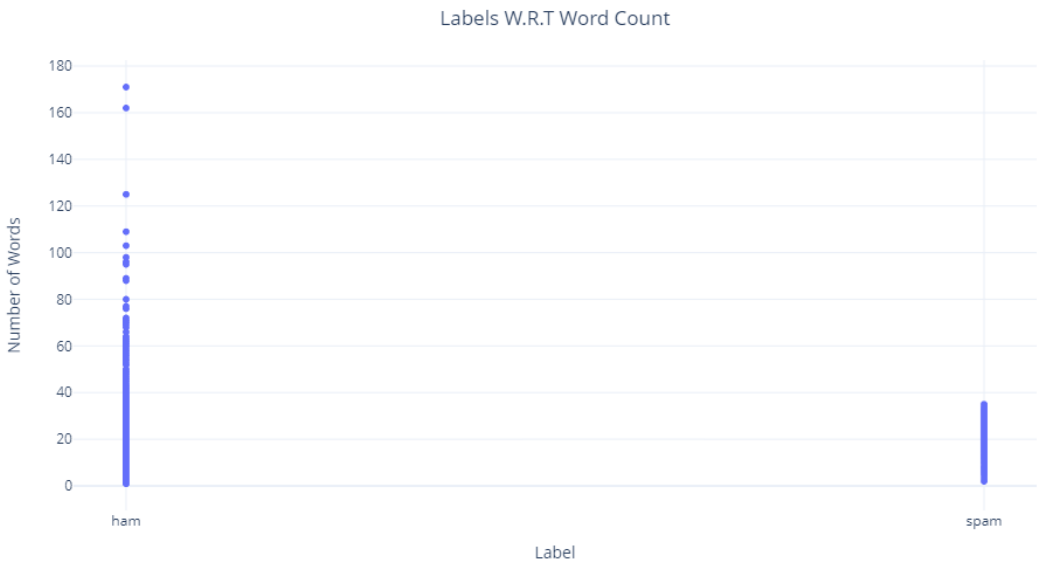
The measures utilized to assess the model's effectiveness included using the inbuilt accuracy function from TextBlob to determine the accuracy in the test data.

III. Visualizations

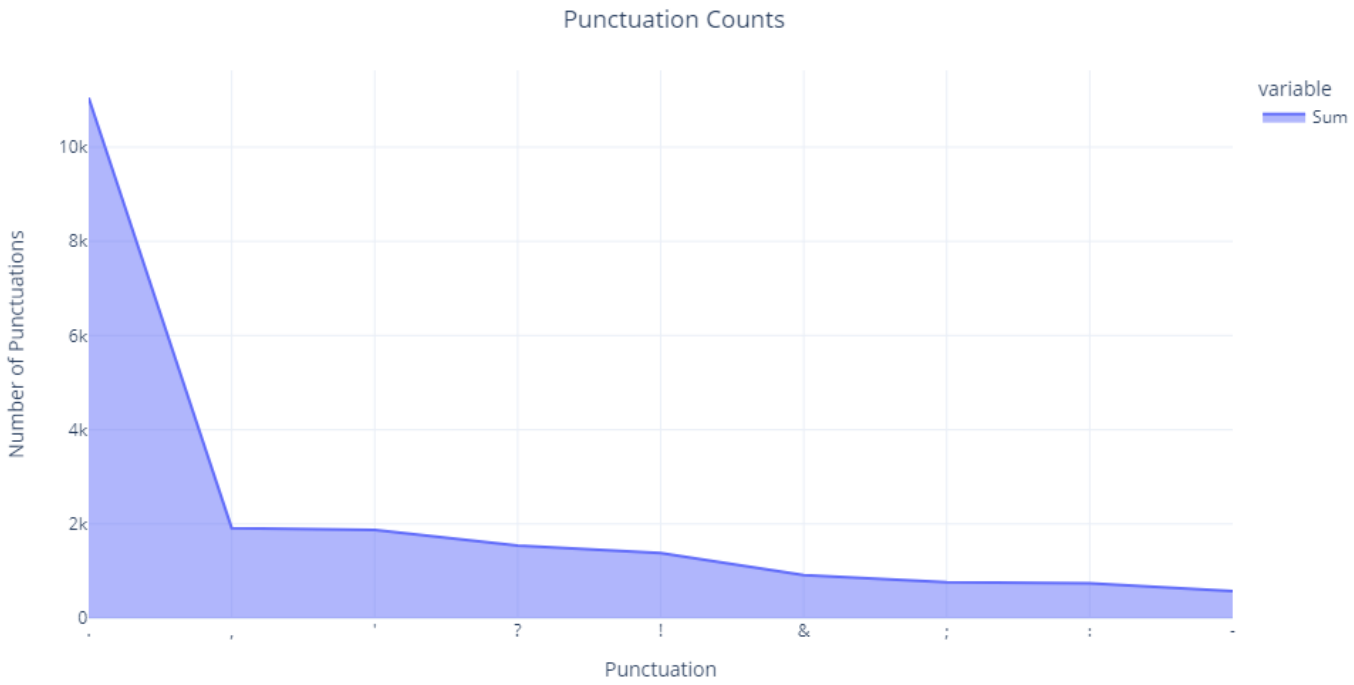
i. Plotting the Number of Sentences marked as spam or Ham



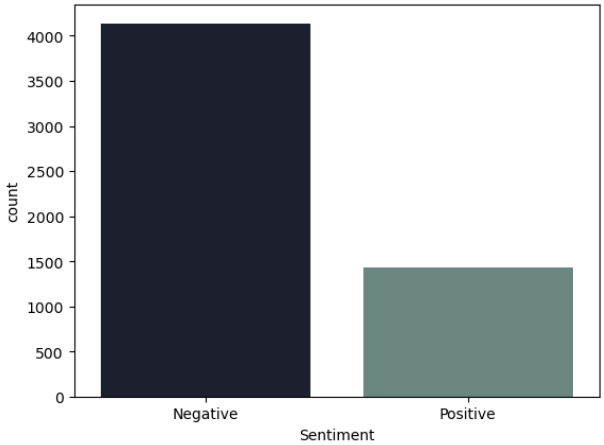
ii. Plotting the Labels with respect to the Word Count



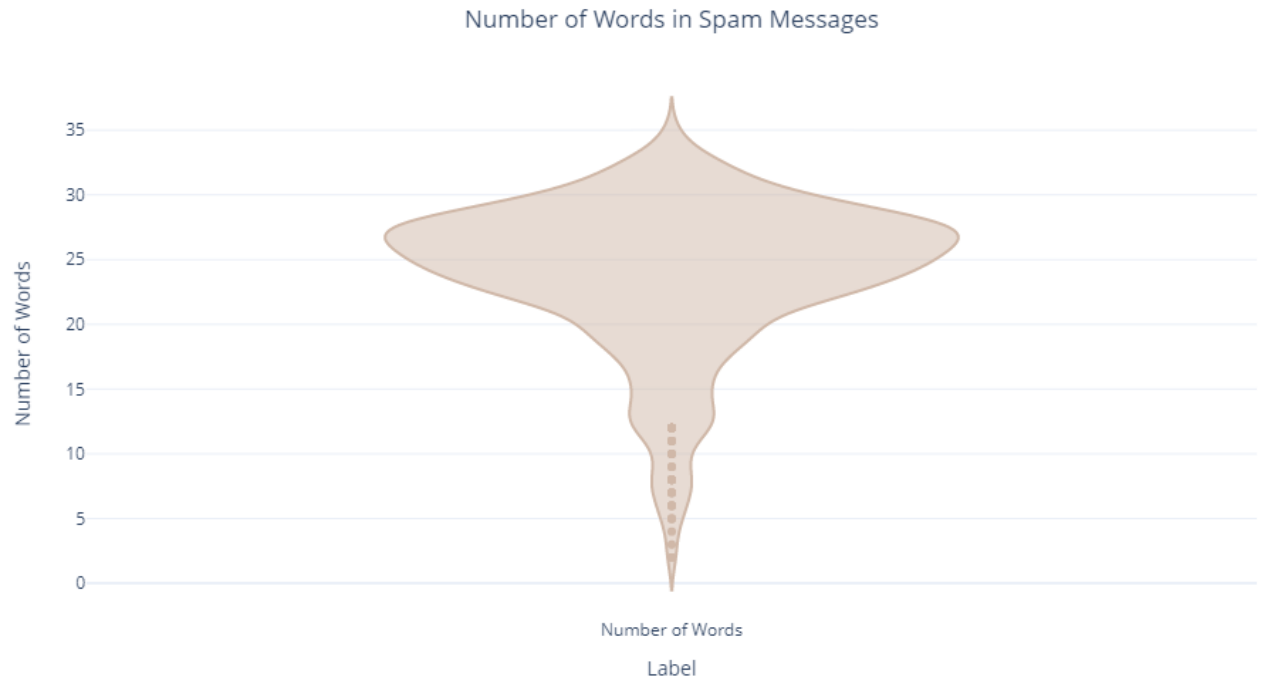
iii. Plotting the punctuation count in the data

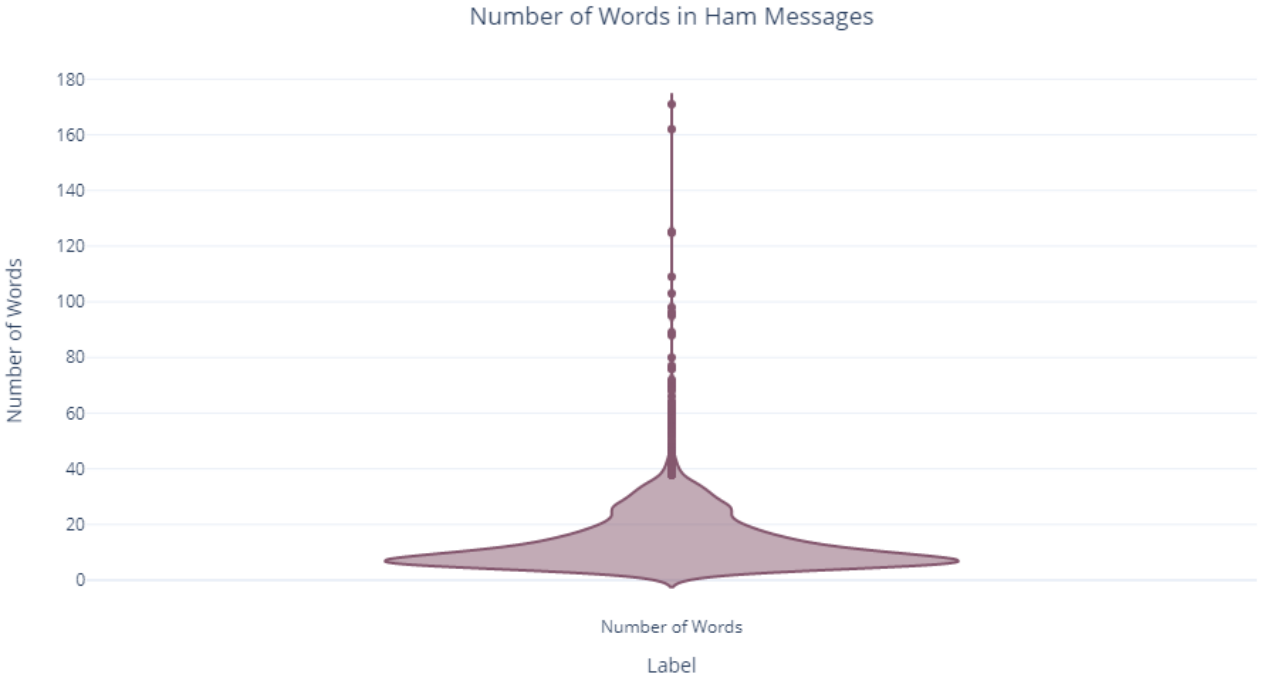


iv. Distribution of Sentiment

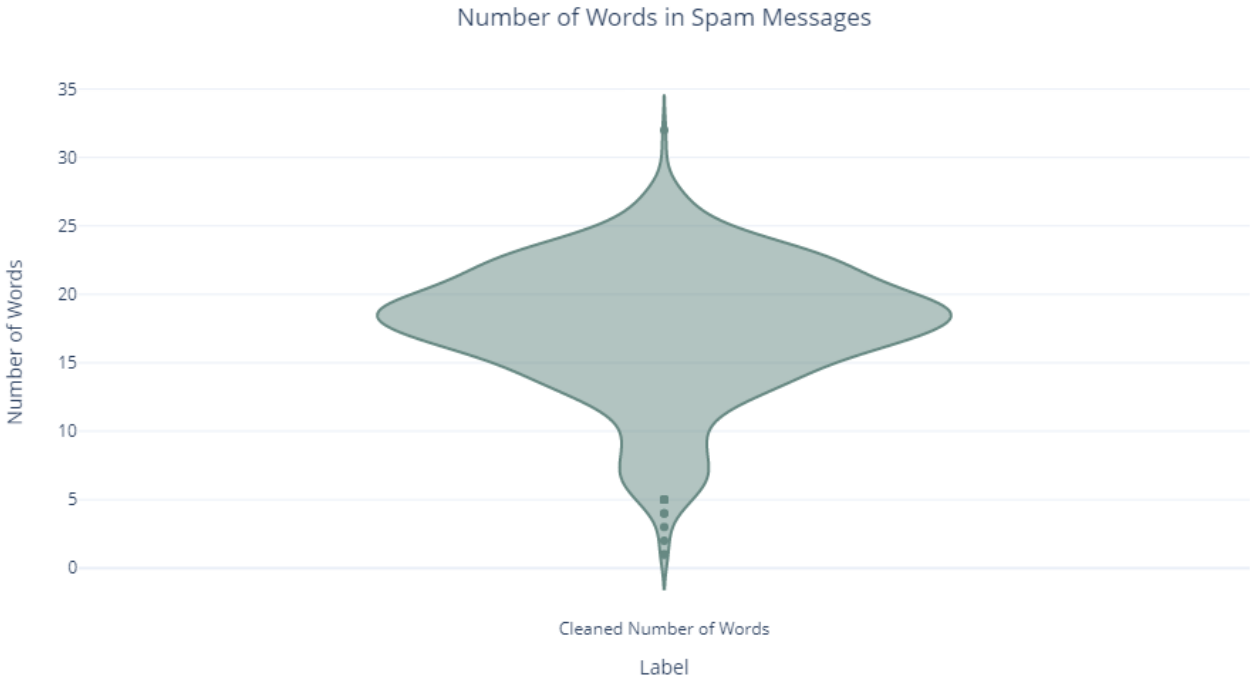


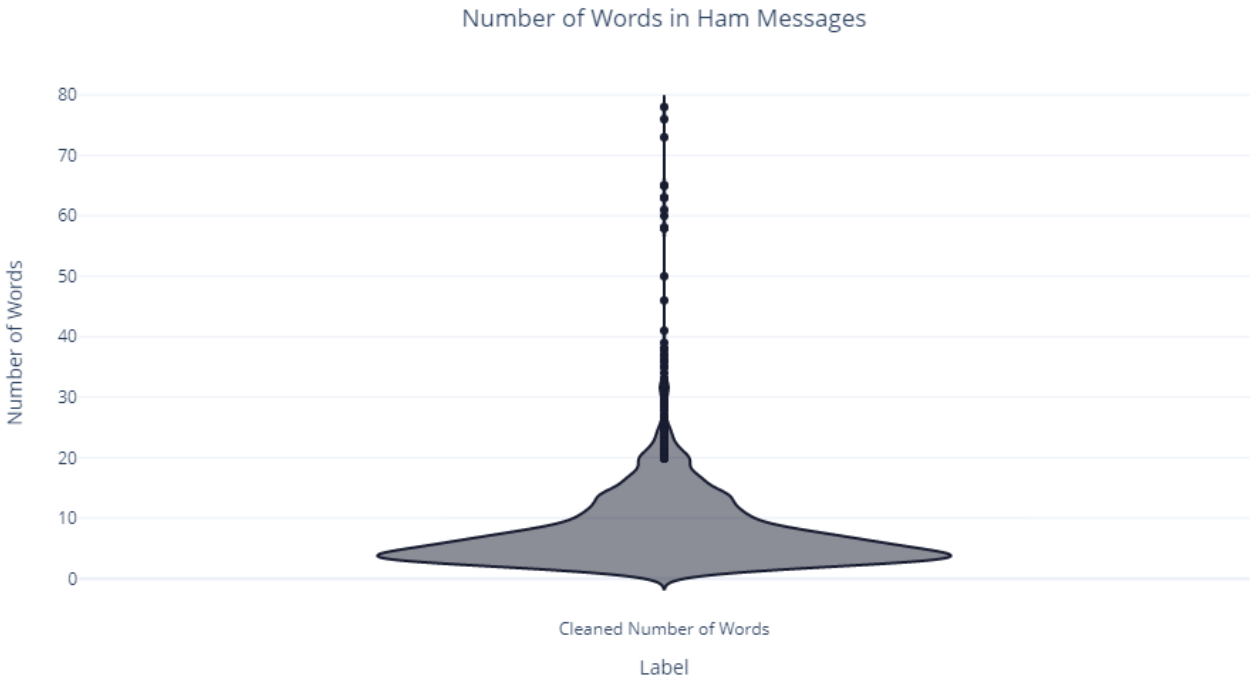
v. Distribution of Words in Spam & Ham messages (before cleaning)



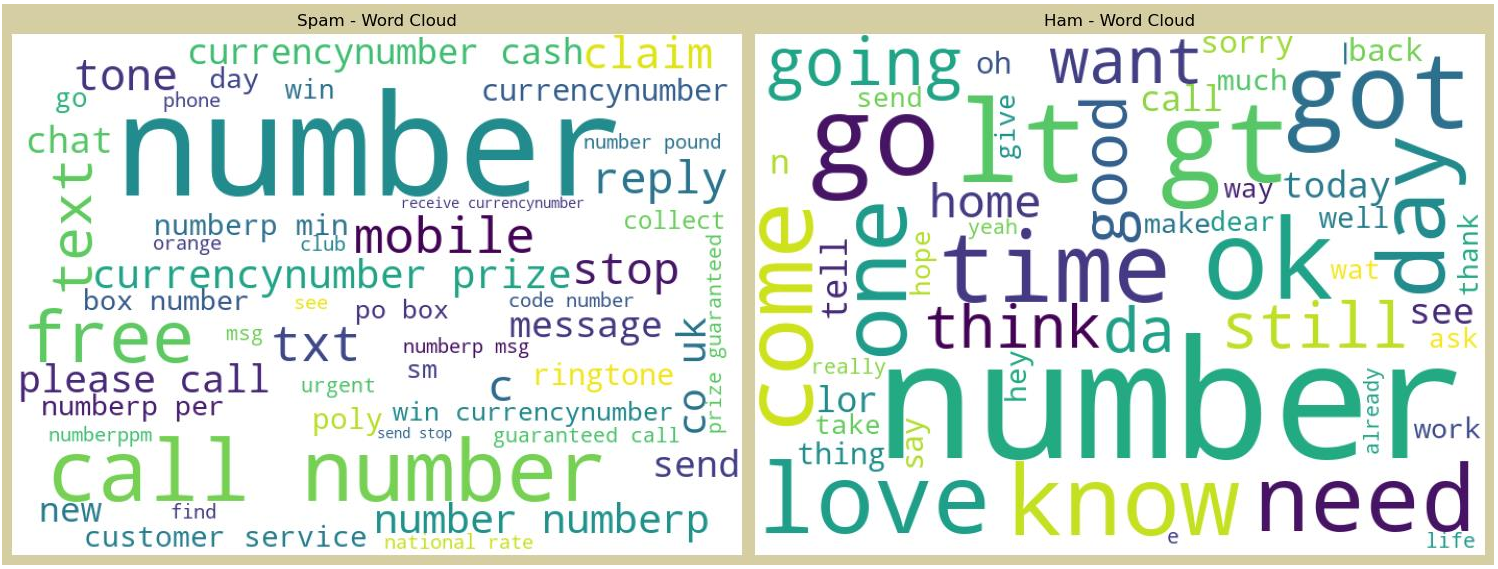


vi. Distribution of Words in Spam and Ham Messages (after cleaning).


Number of Words in Spam Messages
35
30
25
20
15
10
5
0
Number of Words
Cleaned Number of Words
Label


Number of Words in Ham Messages
80
70
60
50
40
30
20
10
0
Number of Words
Cleaned Number of Words
Label

vii. Word Cloud of the loud words in the dataset



viii. Percentage of spam and ham message based on sentiments

```
Positive Sentiment:
The percent of Hams: 81.8%
The percent of Spams: 18.2%

Negative Sentiment:
The percent of Hams: 88.26%
The percent of Spams: 11.74%
```

IV. Interpretation of the Results

a. Spam emails and messages tend to have more words than legitimate ones because spam emails are often sent only once, while legitimate emails and messages may be part of ongoing conversations between two people. Since spam emails are typically sent to many recipients, they often include more words in order to try and capture the recipient's attention and convince them to take some sort of action (e.g., click on a link, purchase a product). In contrast, legitimate emails and messages tend to be shorter because they are usually part of ongoing conversations between people who are already familiar with each other.

b. In spam messages, it is common to see a higher number of punctuation symbols such as @ and currency symbols like $ or £. This is because spam messages often contain emails or mention cash prizes as a way to entice people to fall for a scam.
c. During sentiment analysis, it was found that the majority of spam messages have a positive sentiment. This means that spam messages tend to use positive language and emotion-evoking words in an attempt to convince recipients to take some sort of action.

CONCLUSION

I. Learning Outcomes of the Study in respect of Data Science and Limitations of this work and Scope for Future Work

a. A larger dataset (with enhanced vocabulary from different regions) could be used to further the performance of the model.