Spam Email Filtering using NLP Techniques

Group 8

HUAYE ZHAN (301324797)
YAJUSHI GARG
SURENDRAPALSINGH JHIOUT (301378401)
GHIZLANE EZ-ZARRAD (301412844)

Outline of Project

1

Data Prepration

- Data Cleaning
- Understanding data
- Feature Engineering

2

Data Modelling

- Training
- Testing

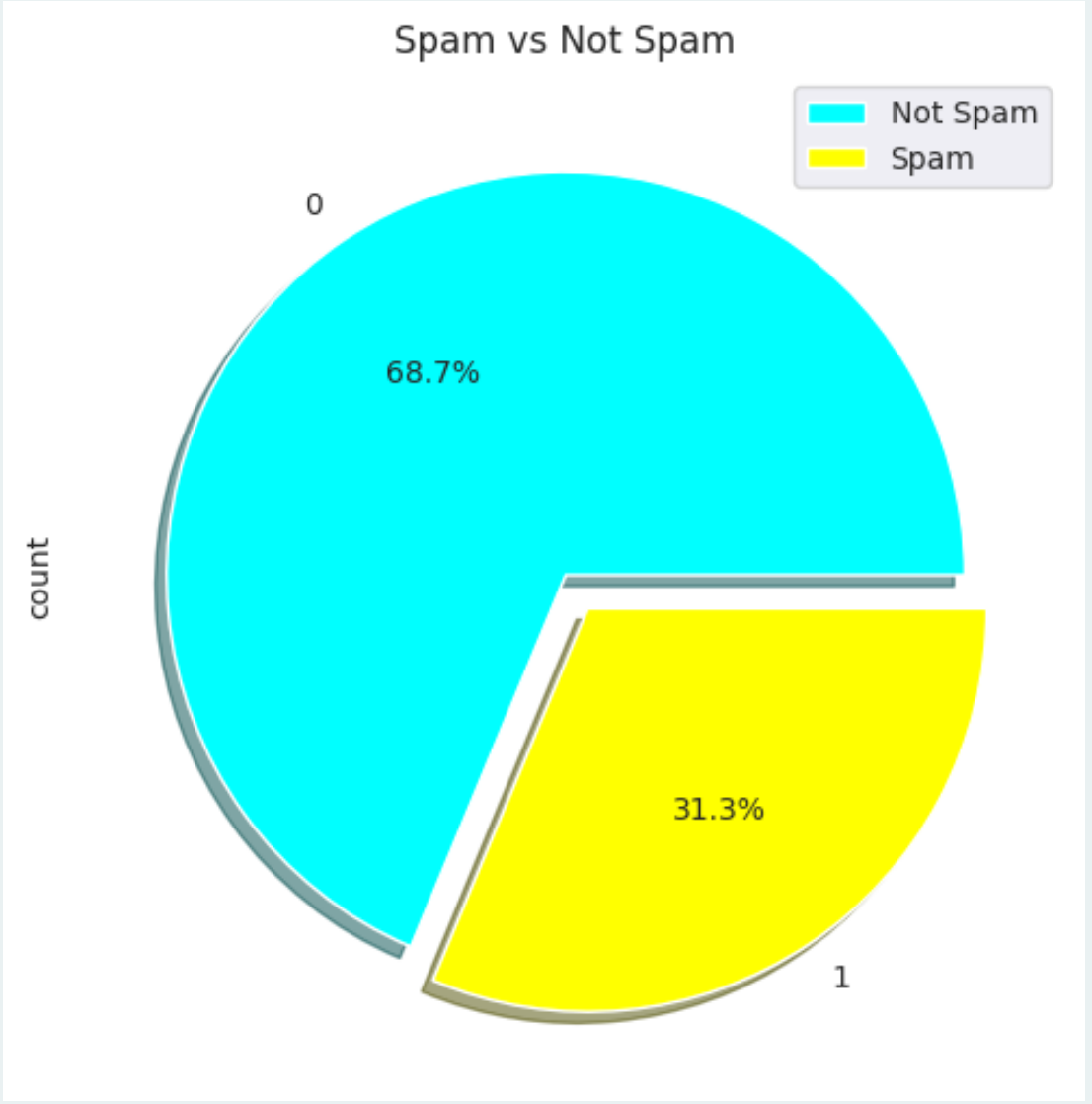
3

Evaluation & Comparison

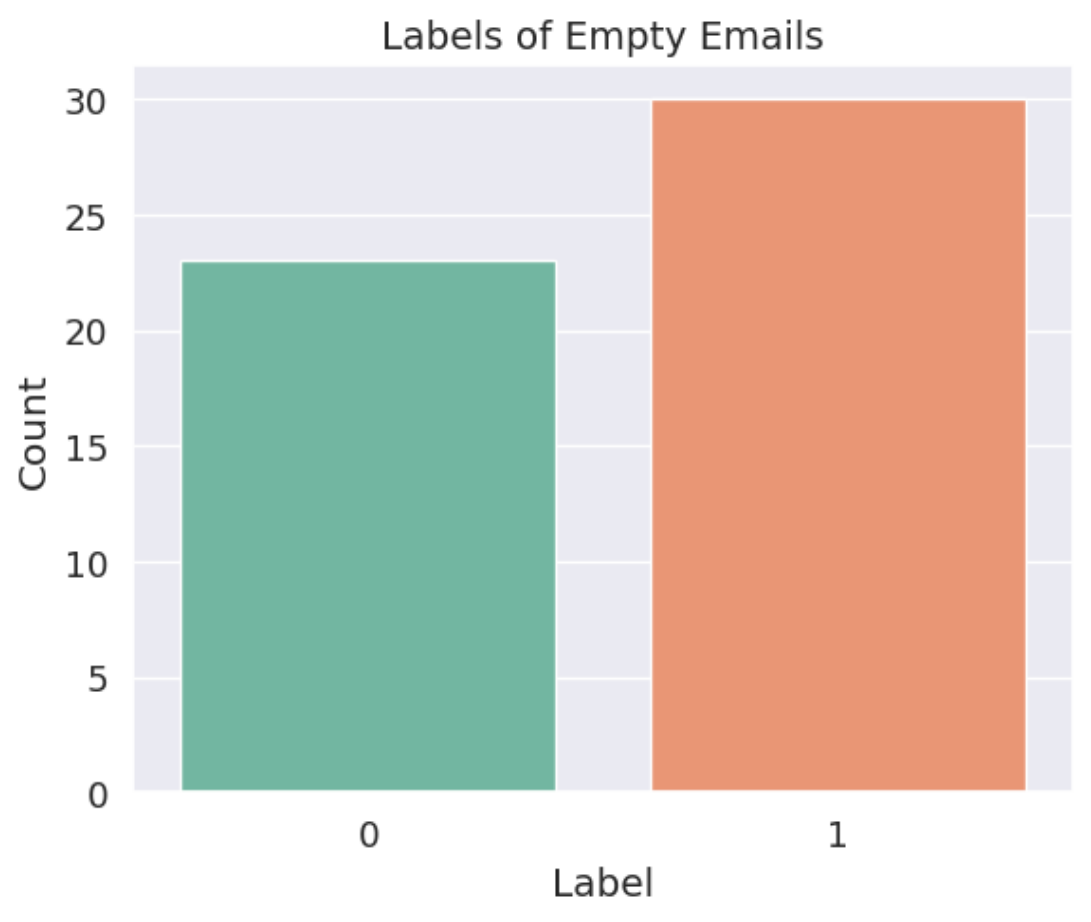
Original Data

TYPE	Not Spam	Spam	Missing Values
Body (count)	460	210	0
Label	0	1	0
Percentage	68.7%	31.3%	0%

Unnamed: 0	Body	Label
1470	empty	1
1471	As seen on NBC, CBS, CNN, and.....	1
1474	MR MABO LOKO\nBRANCH...	1
1475	Dear fork ,\nÂ Â Â Â Â Â Â...	1
666	empty	0



Duplicate Emails



460 Not Spam emails out of which 439 are unique.

210 Spam emails out of which 166 are unique.

"Empty" is the most popular Spam email with repetition of 30 times.

"Empty" is the most popular Not Spam email with repetition 18 times.

	Body	Label
2	empty	1
4	MR MABO LOKO\nBRANCH...	1
...	...	...
607	Empty	0
666	empty	0

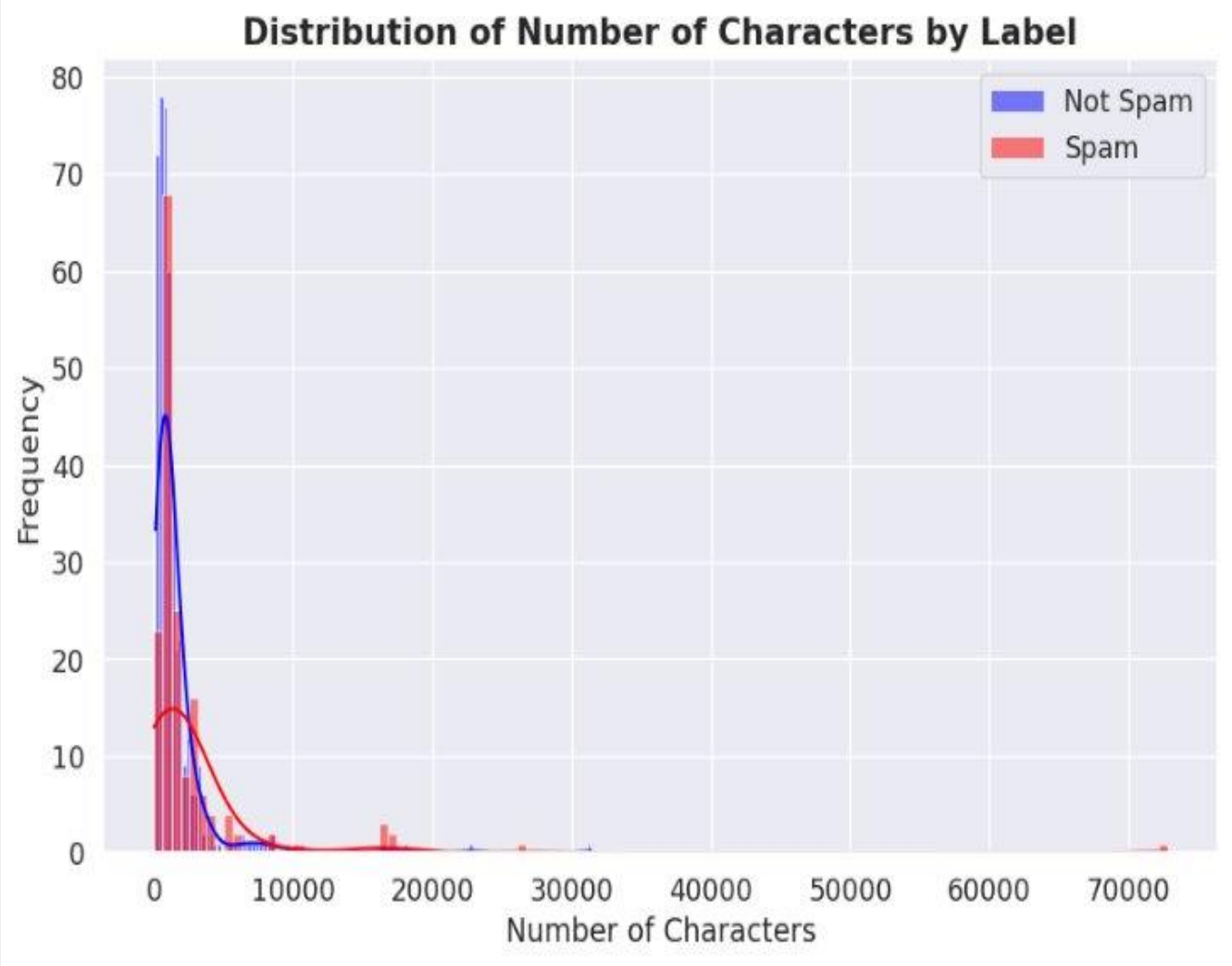
Frequency Distribution

Length of data

count	Min	Max
604	5	72778

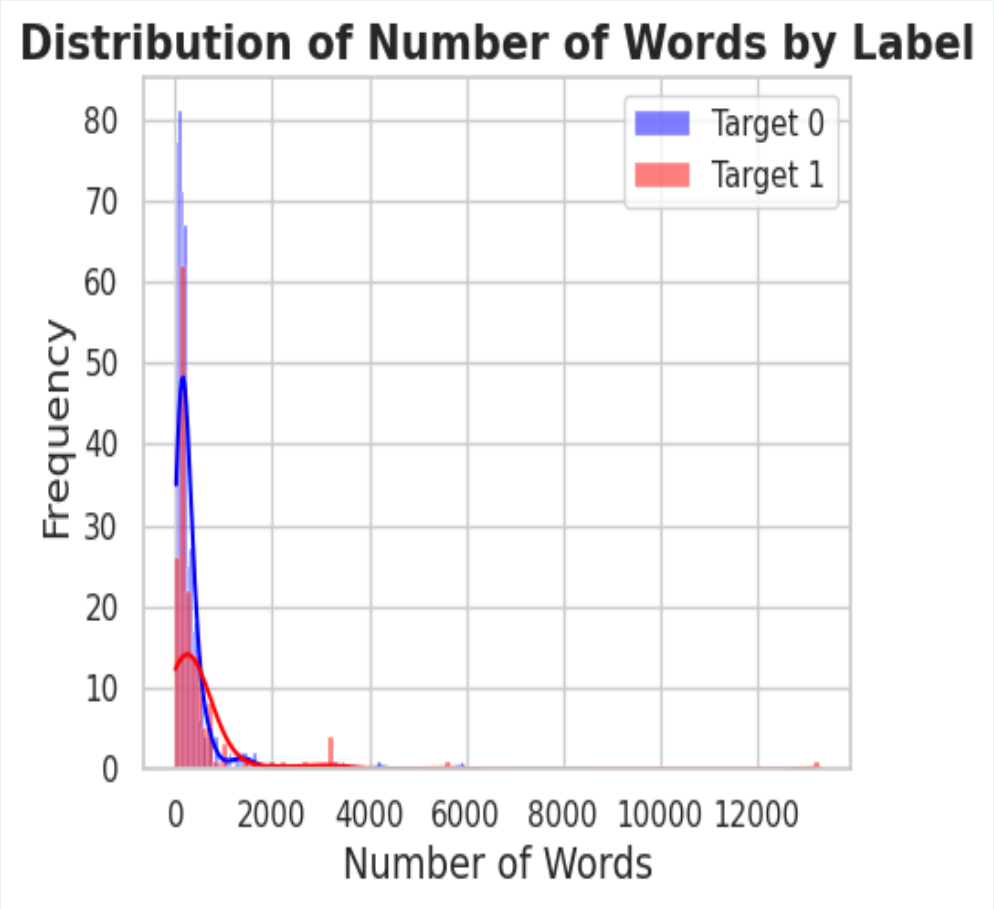
Longest Email

Name	Label	Length
Reply from Enenkio....	1	72778



Frequency Distribution (contd.)

	Label	num_characters	num_words	num_sentence
Label	1.000000	0.127215	0.119417	0.145500
num_characters	0.127215	1.000000	0.996518	0.940011
num_words	0.119417	0.996518	1.000000	0.950956
num_sentence	0.145500	0.940011	0.950956	1.000000

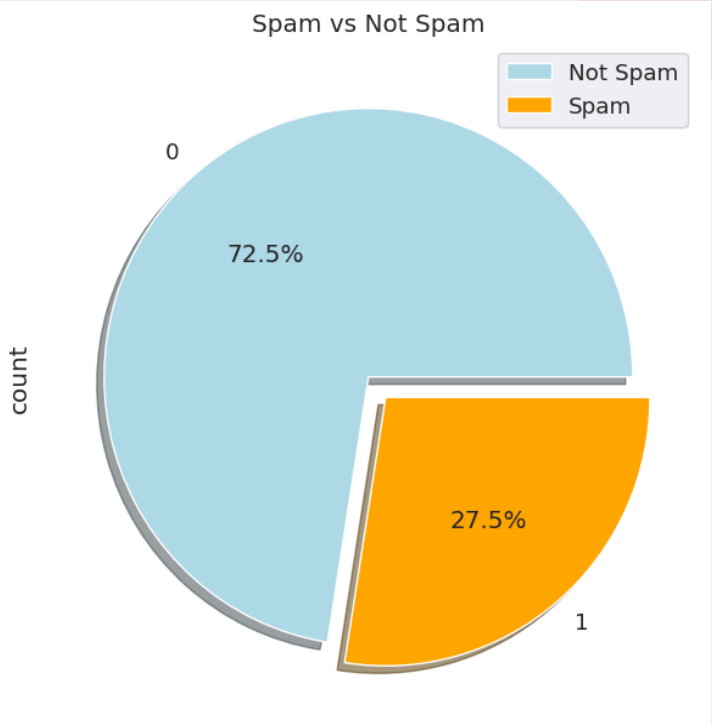


Data cleaning

Total length removal for Spam emails:
Original Length: 447308
Cleaned Length: 287501
Total Words Removed: 159807
Percentage of Words Removed: 35.73 %

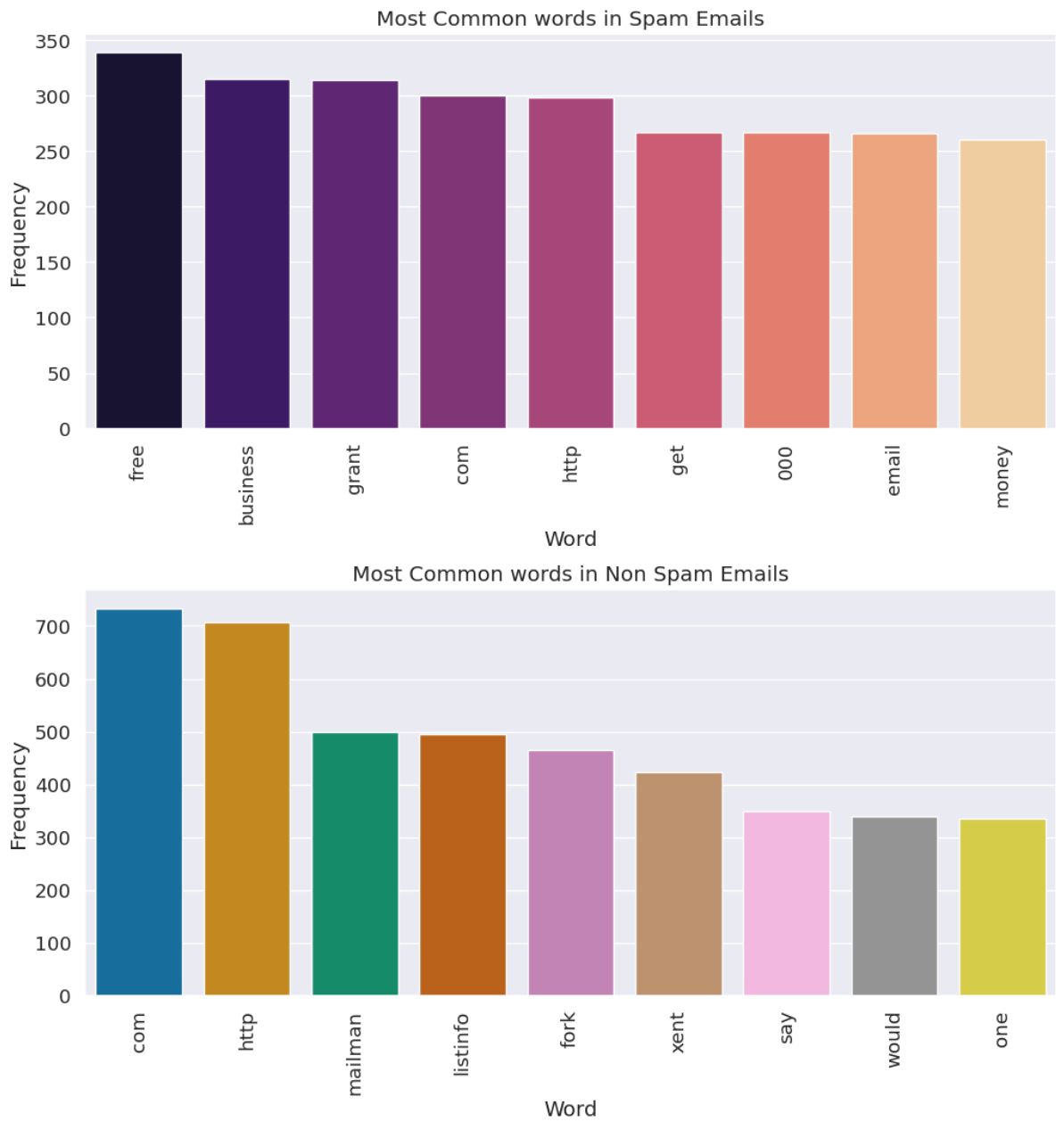
Total length removal for Not Spam emails:
Original Length: 671813
Cleaned Length: 287501
Total Words Removed: 243567
Percentage of Words Removed: 36.26 %

	Label	Count	Percentage
ORIGINAL DATA	No spam	438	68.7%
	Spam	166	31.3%
CLEANED DATA	No Spam	460	72.5%
	Spam	210	27.5%

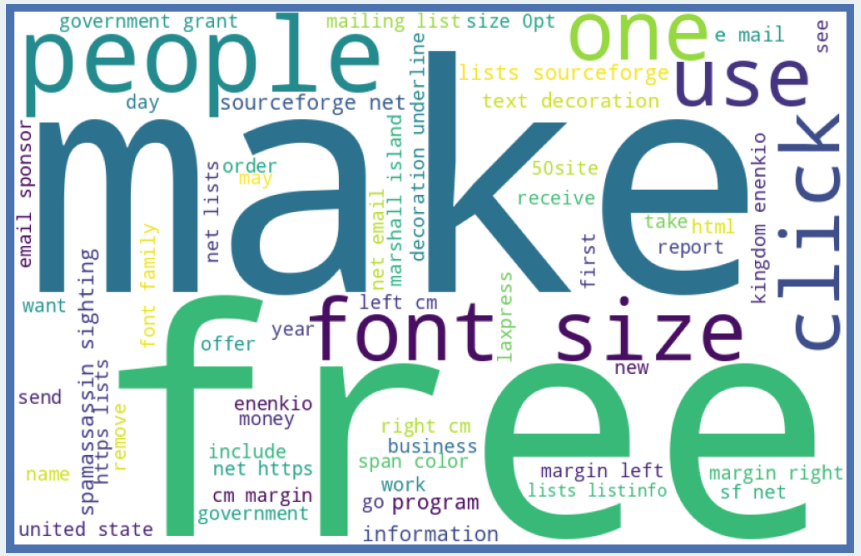


Body	Label	length	Clean_length	Body_cleaned
as seen on nbc, cbs, cnn, and even oprah! the ...	1	1022	707	see nbc cbs cnn even oprah health discovery ac...
mr mabo loko\nbranch officer,\nunited bank for...	1	2462	1483	mr mabo loko branch officer united bank africa...
dear fork ,\nâ â â â â â â â â â want \n ...	1	2838	1650	dear fork want harvest lot email address short...
"now\n is the time to take advantage of...	1	1477	894	time take advantage fall interest rate advanta...
\nâ \nfree personal and business grants\nâ " q...	1	16457	10808	free personal business grant qualify least 25 ...

WordCloud



SPAM



NOT SPAM

Vectorization of texts

Bag of words and Tf-Idf

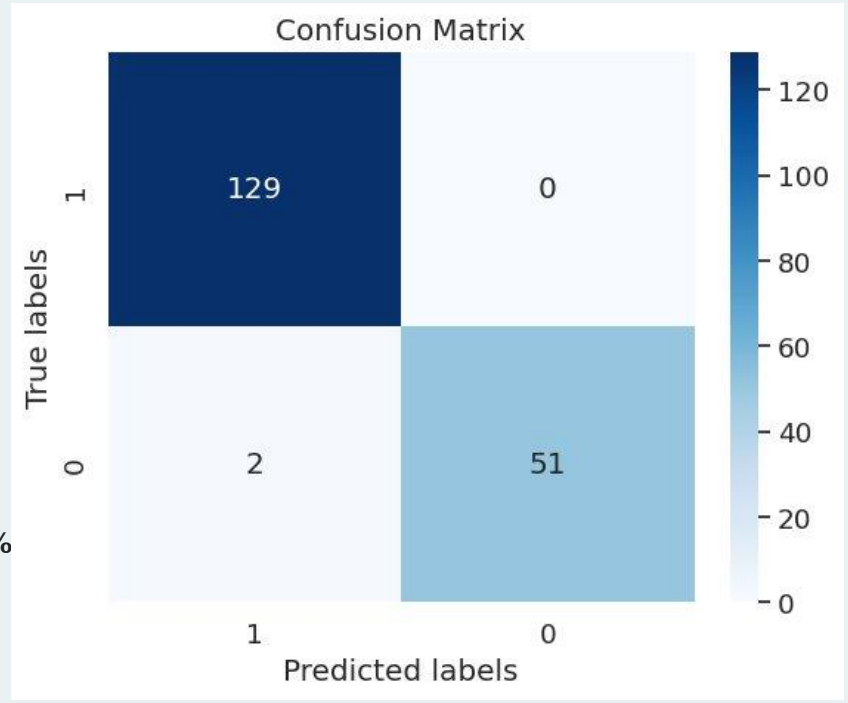
Token: "CALL"

```
OW          TF-IDF:
20 1        20 0.069908
21 1        21 0.044773
22 0        22 0.000000
23 0        23 0.000000
24 0        24 0.000000
25 0        25 0.000000
26 0        26 0.000000
27 1        27 0.019059
28 0        28 0.000000
29 0        29 0.000000
30 0        30 0.000000
31 0        31 0.000000
32 1        32 0.065489
```

Email 20: 129 words
Email 21: 154 words
Email27:678 words
Email 32: 97 words

Classifier Modeling : NAÏVEBAYES

BOW

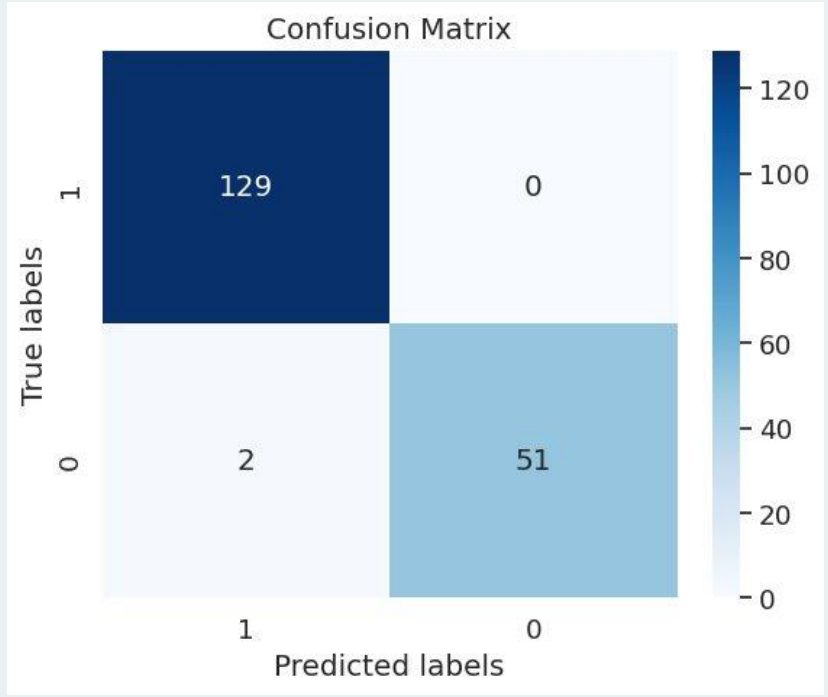


Precision: 100.00%

Recall: 96.23%

F1 Score: 98.08%

TF-IDF

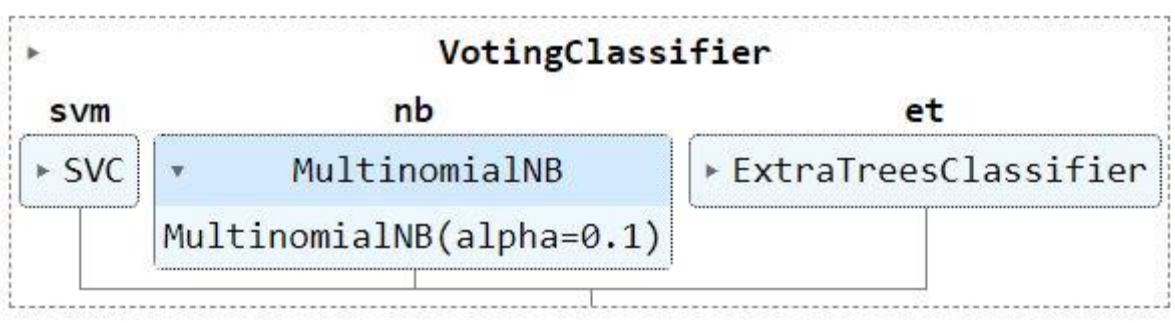


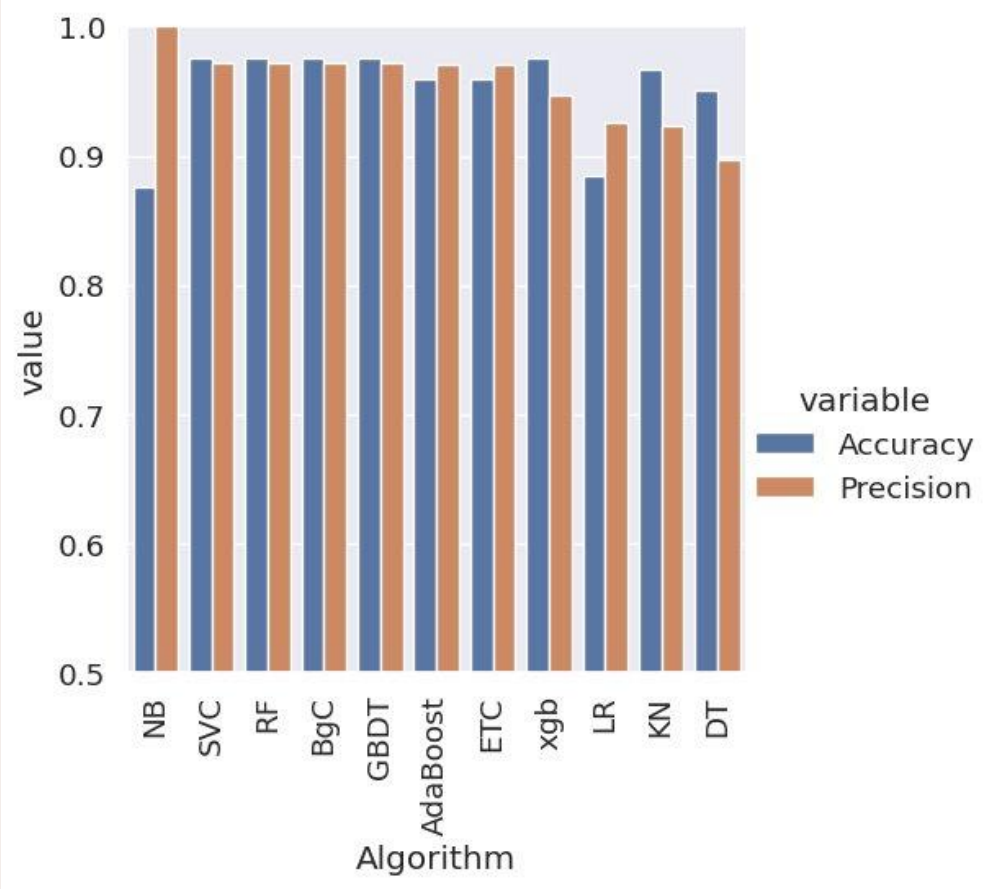
Precision: 100.00%

Recall: 96.23%

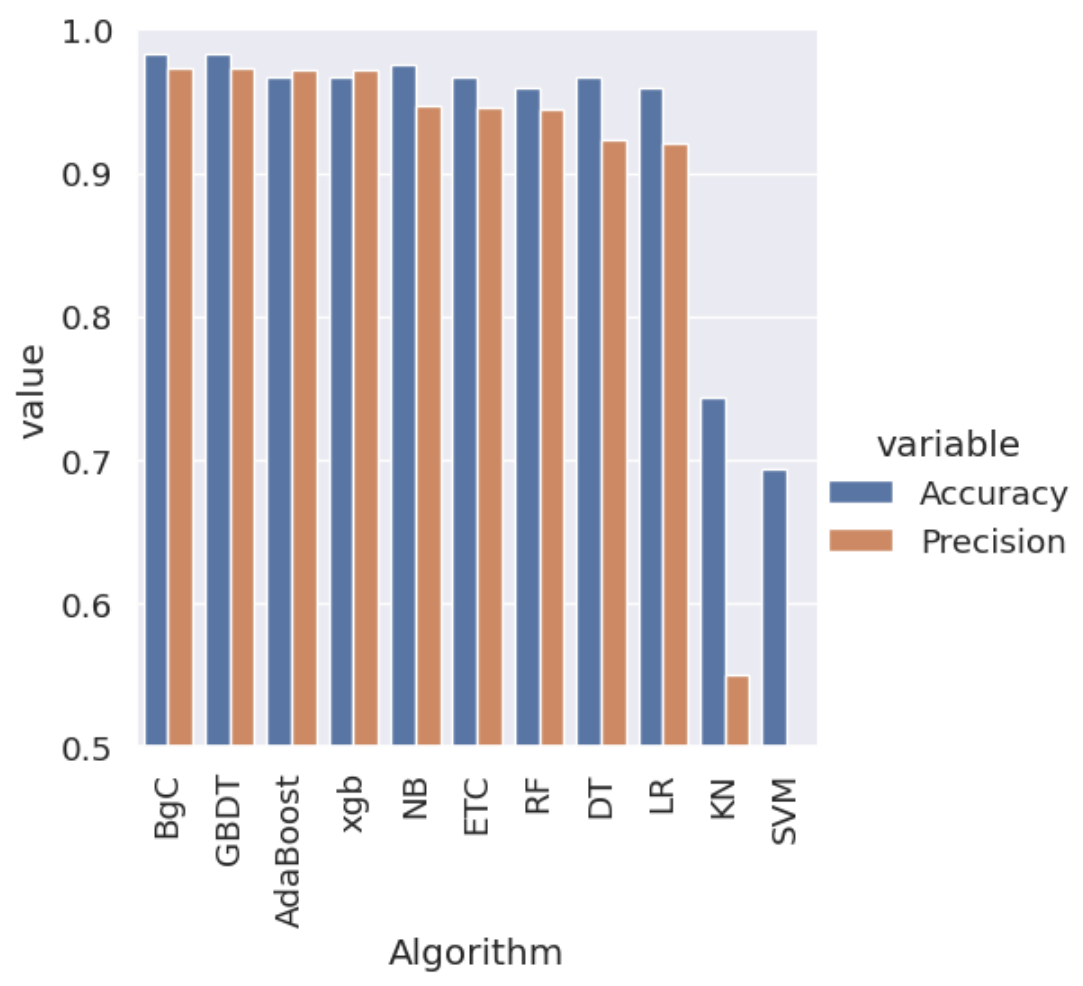
F1 Score: 98.08%

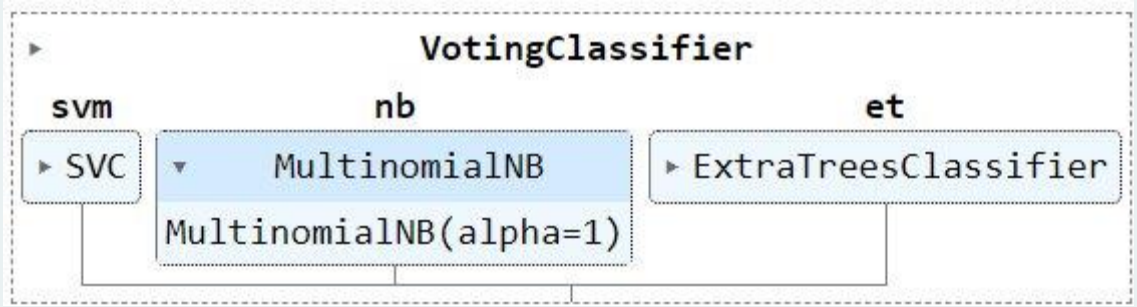
Evaluation And Comparison : TF-IDF





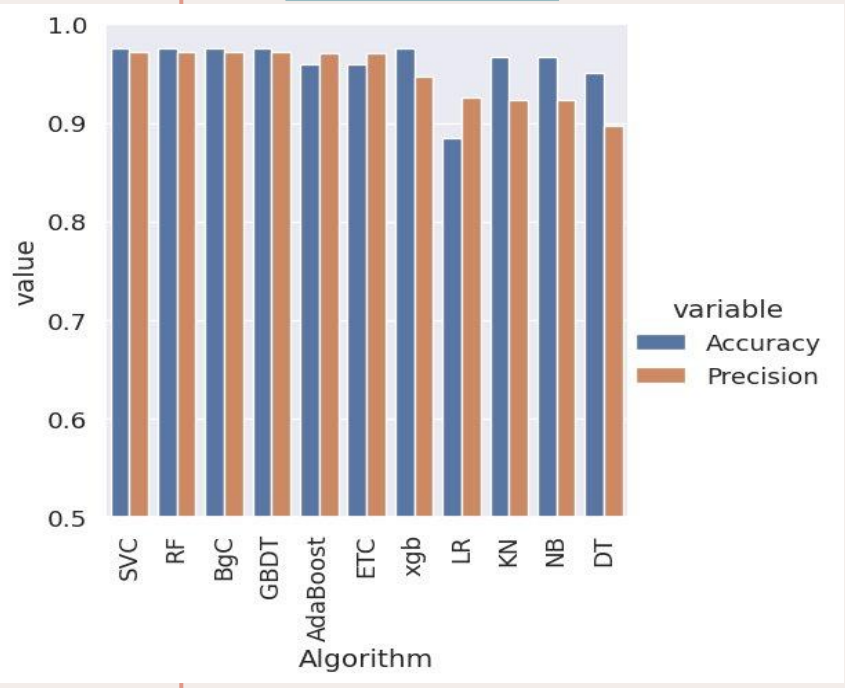
Evaluation And Comparison : BOW



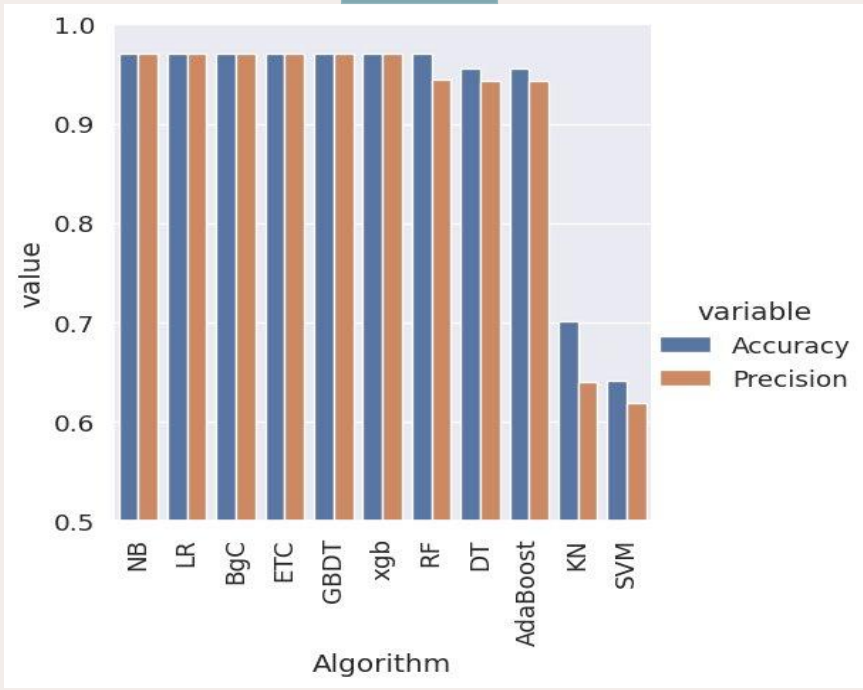


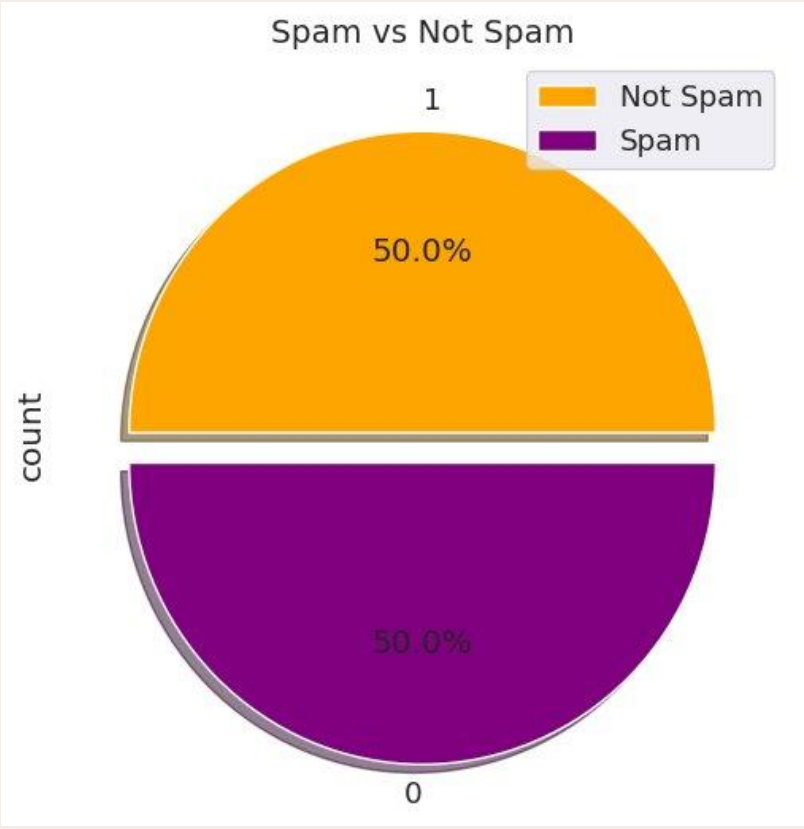
Evaluation Using Resampling

TF-IDF



BOW





Model	Accuracy Before/After	Precision Before/After
NB(TF-IDF)	0.876/0.967	1.000/0.923
NB(BOW)	0.975/0.970	0.947/0.971

Count of Label		
Label	0	1
Count	166	166

Thank you

Analysis Report

1. Understanding the data and its distribution:

The code loads the dataset containing email data using the Pandas library in Python. It begins by reading the CSV file named "Data8.csv" located at the specified path ("/content/sample_data/Data8.csv"). The pd.read_csv () function from the Pandas library is used for this purpose. Once the data is loaded into a DataFrame named nlp_group8, initial exploration is conducted to understand its structure and contents.

Data Overview: The dataset contains information about emails, including the body text and the label indicating whether it's spam or not. The initial exploration includes checking the data's shape, identifying any missing entries, and visualizing the distribution of labels through a pie chart.

Results: The dataset comprises 670 entries, each represented as a row in the DataFrame. There are three columns in the DataFrame:

- Unnamed: 0: This column serves as the index for each entry and contains integer values ranging from 0 to 669 having non-null values.
- Body: It contains the textual content of the emails and is represented as an object data type. There are no missing values in this column.
- Label: The Label column represents the classification label for each email. It contains integer values, where "0" typically denotes non-spam (or legitimate) emails, and "1" represents spam emails. Like the other columns, there are no missing values in this column, with all 670 entries having non-null values.

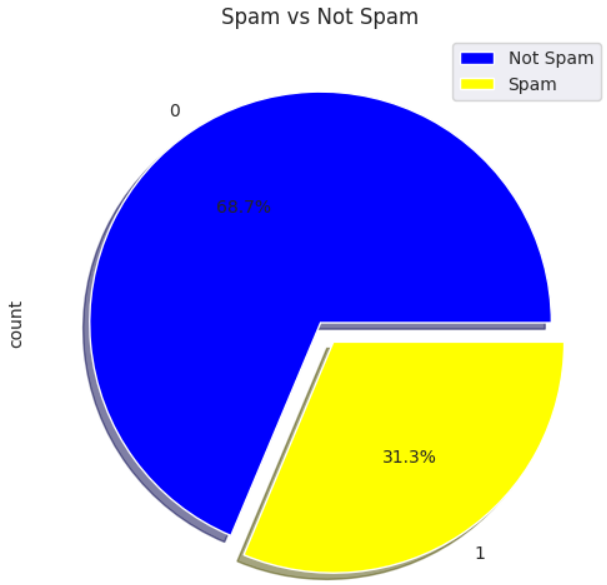
  Overall, the dataset appears to be well-structured, with no missing values across any of the columns. The shape of the dataset is (670, 3), indicating 670 rows and 3 columns.

Secondly, after dropping the unnecessary columns, the code groups the DataFrame by the "Label" column and computes descriptive statistics for each group. The result shows statistics such as count, unique values, most frequent value, and its frequency for both labels 0 (Not Spam) and 1 (Spam).

- Count: There are 460 emails labeled as not spam (0) and 210 emails labeled as spam (1).
- Unique: Among the emails labeled as not spam, there are 439 unique bodies, while among the spam emails, there are 166 unique bodies.

- Top: The most frequent body text among both not spam and spam emails is "empty". This suggests that a notable portion of the dataset consists of emails with no content or with the word "empty" in the body.
- Freq: The word "empty" appears 18 times among not spam emails and 30 times among spam emails, indicating its prevalence as the most common body text in both categories.

Finally, the code generates a pie chart to visually represent the distribution of labels in the DataFrame. It shows the proportion of Not Spam and Spam emails, making it easier to understand the class distribution visually.



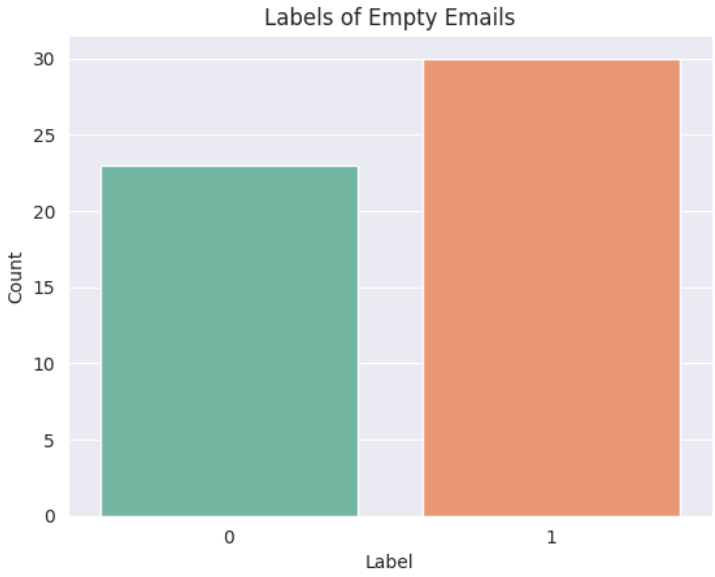
Explanation: The count indicates the number of instances for each label: 460 instances are labeled as 0 (Not Spam), and 210 instances are labeled as 1 (Spam). Hence, Not Spam emails (labeled as 0) constitute a larger portion of the dataset compared to Spam emails (labeled as 1).

2. Feature Engineering: Empty Emails, Duplicate Emails and Length Analysis

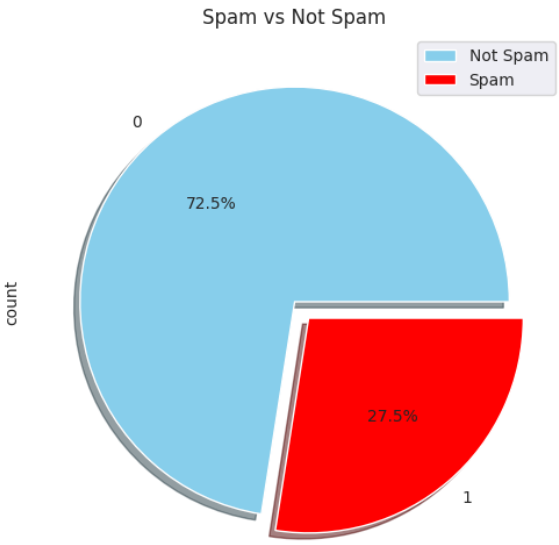
The code conducts feature engineering to extract useful information from the email bodies such as identification and handling of empty emails, analyzing the length of emails, identifying the longest email, and visualizing the distribution of email lengths based on spam and non-spam labels. Additionally, it calculates statistics such as the number of characters, words, and sentences in each email, providing insights into the structure of the emails.

Empty Emails: The code identifies and handles empty emails by filtering them out and visualizing the distribution of labels for these empty emails. It reveals that out of 460 not

spam emails, 439 are unique, while out of 210 spam emails, 166 are unique. Interestingly, the term "Empty" appears as the most frequent label for both spam and not spam emails, occurring 30 times in spam emails and 18 times in not spam emails. The graph sheds light on the prevalence of empty emails in the dataset and their distribution across different label categories, providing valuable insights into the composition of the email data.



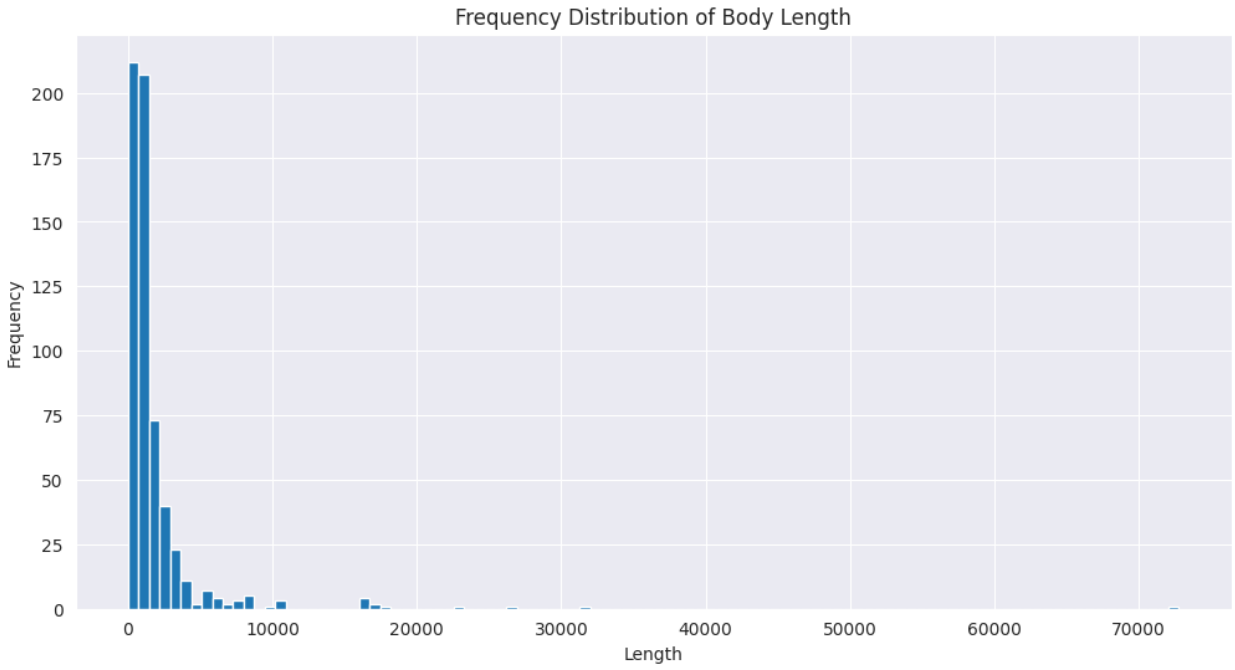
Duplicate Emails: The duplicate emails were identified and removed, resulting in the elimination of 65 duplicate entries. Subsequently, empty emails were filtered out, with 48 initially present. These empty emails were then labeled as spam (1), and one of them was randomly selected and labeled as such. Before cleaning the data, the dataset contained 670 entries with 438 labeled as not spam (0) and 166 labeled as spam (1). After removing empty emails, the dataset was refined to 604 entries. Notably, there were no missing values in the dataset. The final dataset consisted of 438 emails labeled as not spam (0) and 166 labeled as spam (1). This cleaning process resulted in a more balanced dataset with 72.52% of emails classified as not spam and 27.48% as spam. The pie chart visually illustrates this distribution, showing a reduction in the proportion of spam emails after data cleaning. This analysis highlights the effectiveness of data cleaning in achieving a more balanced and representative dataset for further analysis.



Length Analysis: In this part of analysis, the length of emails is analyzed, including identifying the longest email and its label. Histograms are plotted to visualize the distribution of email lengths for both spam and non-spam emails**.**

Firstly, a frequency distribution table has been created to describe the length of the e-mail body in the data set. In the graph below, x-axis depicts the body length of that email, and the y-axis illustrates the frequency of that email length.

As a result of this, it is seen that emails within length 0-10,000 words are most frequent with a very high chance of occurrence.



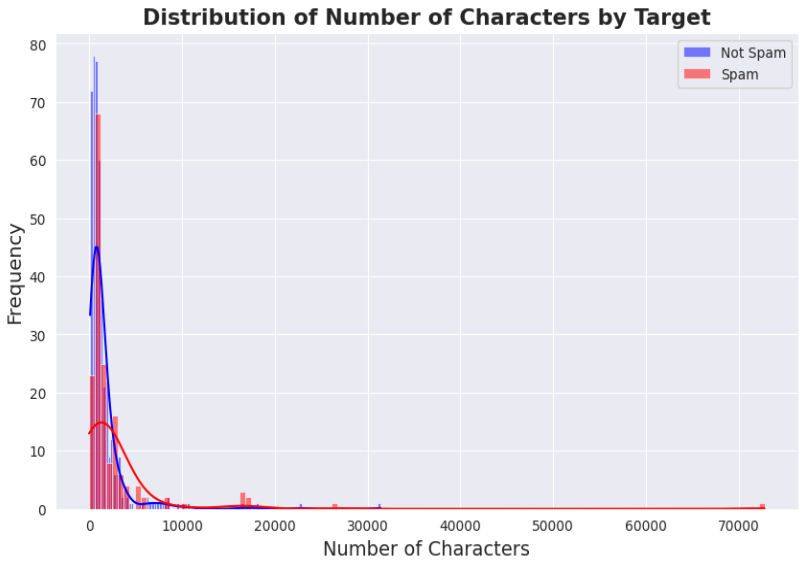
On further exploration of the data, it was analyzed that the lengthiest email was the one which had about 72778 characters and it had label for 1 which indicates that the probability of an e-mail to be spam is in direct correlation with the number of characters in that e-mail.

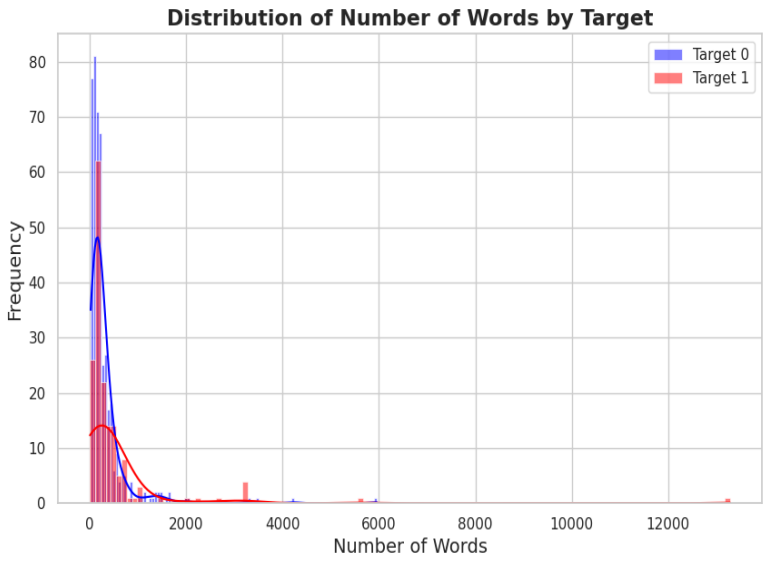
3. Data Visualization

On analyzing the statistics for spam and no spam emails based on the characters it has, it can be said that on an average the non-spam email has 1534 characters, 296 words, and 10 sentences whereas for the spam emails they have average length of about 2695 characters,

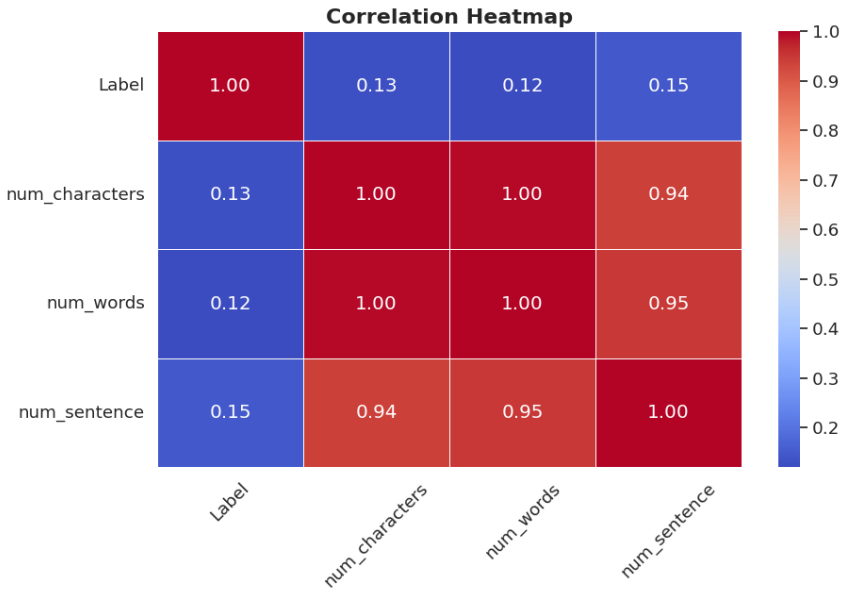
501 words, and 18 sentences. Though both the types of emails have significant variability in their length however for the non-spam emails, the shortest email has 108 characters, 15 words, and 1 sentence, while the longest email spans 31401 characters, 5956 words, and 225 sentences yet in case of spam emails the length is surprisingly as short as 5 characters, 1 word, and 1 sentence to as long as 72778 characters, 13288 words, and 368 sentences which clearly denotes that there are substantial differences between non-spam and spam emails in terms of length and complexity.

To elaborate, two different analysis has been done on the basis of character length distribution and the word count distribution for both categories of e-mails and a significant analysis has been done.





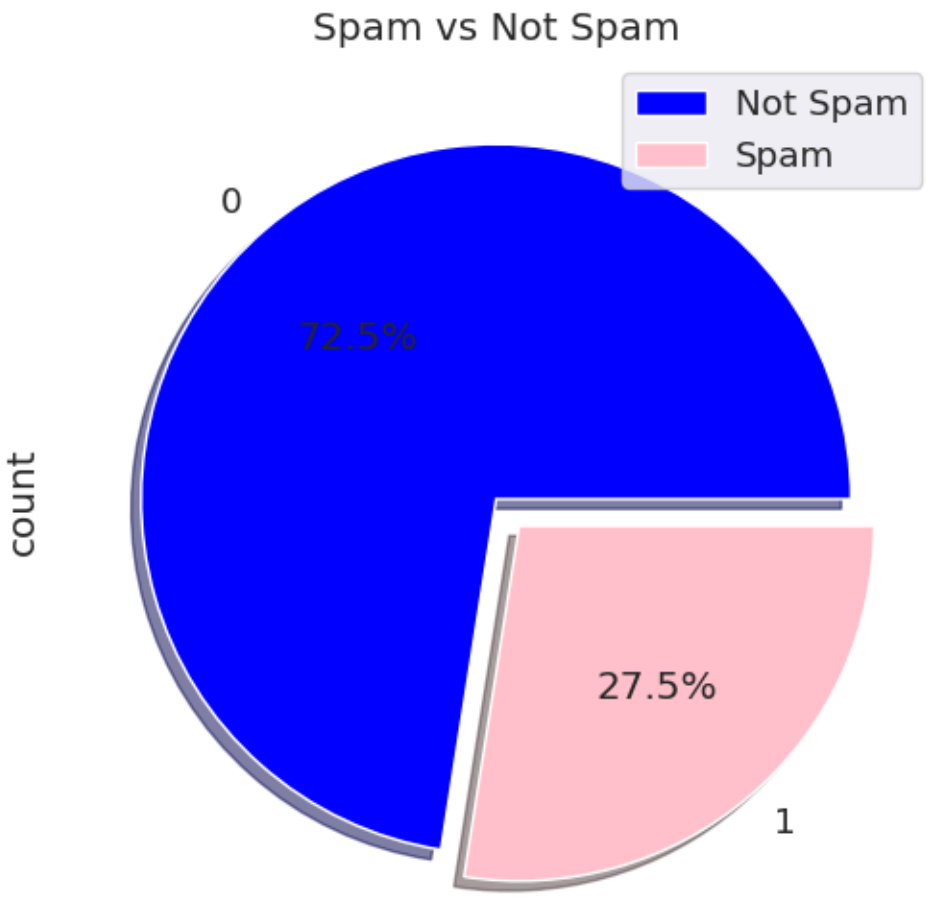
It is observed both the graphs are highly skewed and legitimate emails usually have shorter lengths and use words more consistently, meaning they stick to a similar number of words in each email. On the other hand, spam emails vary a lot in length and contain a wider range of words, sometimes having many more words than legitimate ones.



Before the text is pre-processed, the correlation heatmap is created that visually represents the correlation coefficients between different variables in the dataset. It uses colors like blue and red to represent how strong these relationships are. Blue means they're not very related, while red means they're strongly connected. Each square on the map has a number that tells us exactly how strong the relationship is. We look at things like the number of characters, words, and sentences in emails, and how they relate to each other and to whether the email is spam or not.

4. Text pre-processing

This is when the actual cleaning takes place such as removing lemmas, punctuation marks, stop words, special characters, symbols, whitespace and substituting those multi space emails with single space.



After cleaning the data, almost 15908 words were removed (35.73%) from the data of the spam emails that was originally found to be 44708 now it has been reduced to 287501.
Similarly, in the case of non-spam emails 36.26% of words were reduced after the cleaning.
Hence, the new percentage of emails that were spam is 27.5 and 72.5 for that of non-spam emails.

5. Vectorization of Texts

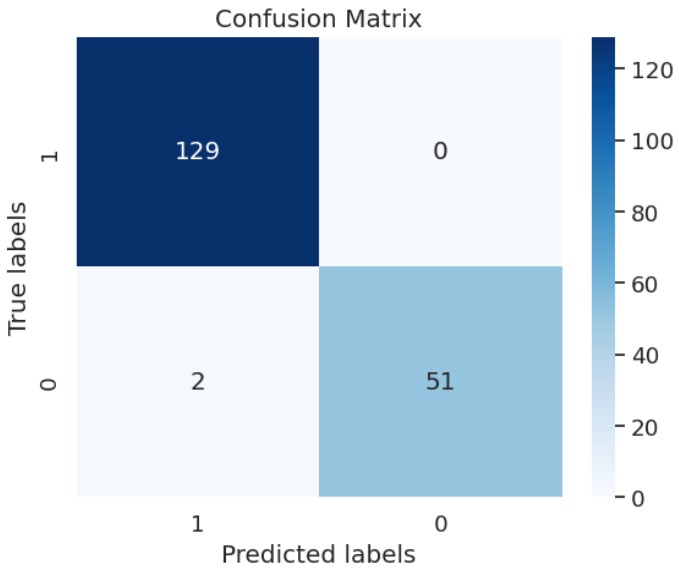
Vectorization has been done using two methods: Bag of words and TF-IDF. On printing the first 15 lines of bow_vector and tfidf_vector with column 'call', it was observed that ‘call' in lines 20, 21, 27, 32 has the same value of 1 in BoW vectorization. However, for TF-IDF, the value in line those lines are different. This means that TF-IDF considers the fact that email 32 has less tokens than other emails. This means that in relation to email 27, the word call is more important for

understanding the meaning of email 32 than in email 27. This example shows how TF-IDF captures information which cannot be found in BoW.

6. Classifier Modelling

A grid search approach is implemented to optimize the hyperparameters of a Multinomial Naive Bayes classifier trained on TF-IDF as well as Bag of Words features. Through this function, a systemic range of hyperparameters is approached, such as alpha, and it uses cross-validation to identify the combination that maximizes performance. This process enhances the model's predictive accuracy and generalization by fine-tuning parameters tailored to the dataset.

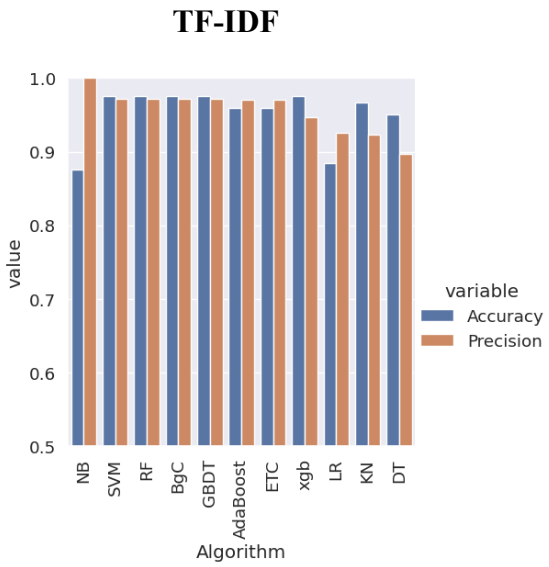
NB with TF-IDF: On calculating and plotting the classification metrics, we concluded that the confusion matrix correctly identified 129 non-spam emails (True Negatives) and 51 spam emails (True Positives), 0 non-spam emails as spam (False Positives) but misclassified and missed 2 spam emails (False Negatives) and as it accurately identified all spam emails, it achieves 100% precision. Recall reveals how well the model detects all spam emails. In this case, it caught 96.23% of them.

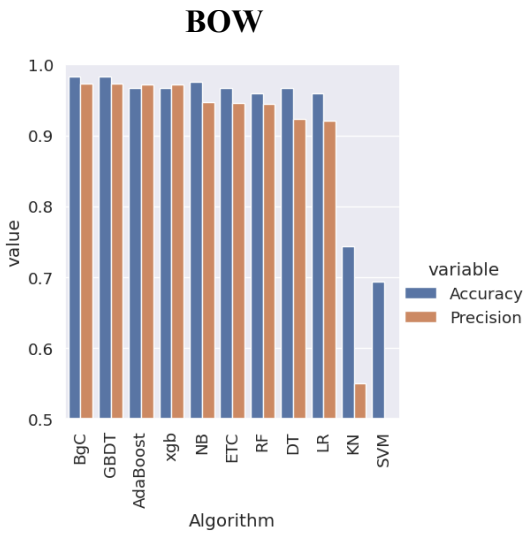


The F1 Score combines precision and recall, providing an overall model performance measure. Here, it's 98.08%, indicating high effectiveness in identifying spam while minimizing false alarms. These results signify the model's reliability in distinguishing between spam and non-spam emails.

When the classification was done with Bag of Words, surprisingly, Similar results were found.

Modelling of Tf-Idf and Bag of Words before undersampling:



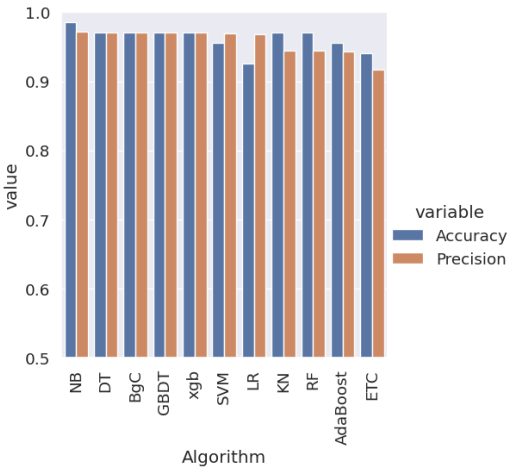


The main difference between Bag of Words and Tf-Idf modelling is how they handle feature representation for textual data. Tf-Idf provides a refined representation by assigning weights to words according to their importance both inside a text and throughout the corpus. In contrast, Bag of Words creates a simple matrix without considering the relative significance of words across the corpus, with each row representing a document and each column representing a single phrase. Bag of Words just portrays texts based on word frequencies, but Tf-Idf effectively conveys term significance by considering both local and global frequencies.

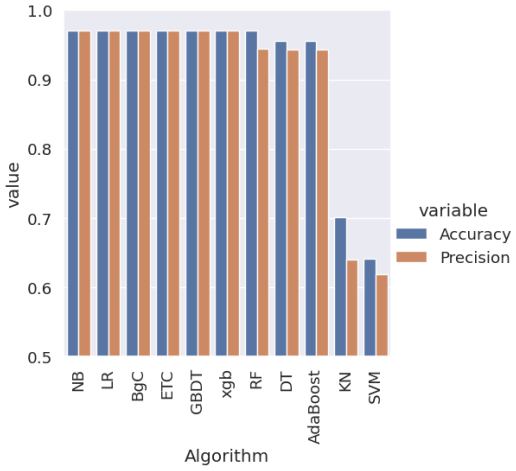
Moreover, there is a slight difference in both the accuracy and precision as for Tf- Idf it is 0.99 and 0.97 respectively and for Bow the values are 0.96 and 0.92 respectively.

Modelling of Tf-Idf and Bag of Words after undersampling:

TF-IDF



BOW



After resampling, both Tf-Idf and Bag of Words had improved performance in terms of accuracy and precision for most algorithms. It is noticed that the performance improvements were consistent across both feature representation methods, with algorithms such as NB, LR, BgC, and GBDT achieving higher accuracy and precision scores post-resampling. However, there were still discrepancies in performance among algorithms, with some achieving higher scores than others regardless of the feature representation method used. Therefore, resampling has resulted to overall performance enhancement, the choice of algorithm remains critical in determining the effectiveness of text classification tasks.