A Project Report

on

E-MAIL SPAM CLASSIFIER

Submitted in partial fulfillment of the
requirement for the award of the degree of

B.Tech in Computer Science & Engineering

GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. Yashwant Soni

Submitted By

Raman Garg/19SCSE1010749
Priyanshu Chauhdhary/19SCSE1010749

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING / DEPARTMENT OF COMPUTERAPPLICATION
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
MAY, 2023

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **" E-MAIL SPAM CLASSIFIER "** in partial fulfillment of the requirements for the award of the B.Tech submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of January, 2023 to May 2023, under the supervision of Mr. Yashwant Soni, Department of Computer Science and Engineering, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Raman Garg

Priyanshu Chauhdhary

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name

Designation

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Priyanshu Chauhdhary and Raman Garg has been held on __________________ and his/her work is recommended for the award of B.Tech in Computer Science and Engineering

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Program Chair **Signature of Dean**

Date: May, 2023

Place: Greater Noida

ABSTRACT

Email spam is a chronic issue that annoys users and wastes time in inboxes all across the world. It is becoming more difficult to filter out unwanted and potentially hazardous emails due to the rise in email usage. Therefore, it is crucial to create methods that can automatically recognise and categorise spam emails.

In order to improve email spam classification, this project will investigate several machine learning methods and feature engineering methodologies. For training and evaluation, we use a publicly accessible dataset made up of spam and non-spam emails. We compare the accuracy, precision, recall, and F1 score of three well-known machine learning algorithms: naive Bayes, support vector machines, and decision trees.

Existing Problems:

Users are at risk for security as a result of email spam, which has grown to be a chronic issue that can waste time and resources. Traditional rule-based spam filtering strategies have had little success and frequently need numerous modifications to stay current with new spamming methods. A more effective and efficient method of classifying email spam is therefore required.

Proposed Solution:

In this project, we provide a machine learning-based approach to the challenge of classifying email spam. Our technique entails using a dataset of preprocessed email messages to train a Naive Bayes classifier. The most pertinent words and phrases in email messages that are most suggestive of spam are identified using the TF-IDF approach, which we employ for feature selection. Using this method, we can create a spam classifier that is more accurate, effective, and capable of learning from the data and adjusting to new types of spam.

Tools and technologies used:

We implemented our Naive Bayes classifier using Scikit-learn, a well-known machine learning toolkit, and we programmed our project in Python. The Natural Language Toolkit (NLTK) toolkit, which has tools for cleaning and tokenizing text data, was also used for text preprocessing.

Result and output:

We tested our model on a test batch of emails, and we were able to categorise spam emails with an accuracy of 95%. The model's capacity for usage in actual email filtering applications may be shown by the fact that it was able to correctly categorise a significant portion of the spam and non-spam emails.

Conclusion and Future Scope:

In conclusion, the email spam classification issue is effectively and efficiently solved by our machine learning-based approach. Traditional rule-based spam filters are outperformed by the suggested method, which has the potential to be included into email clients and services to improve user experience and security. Future research could look into how to increase the model's precision and scalability as well as how to incorporate it into email clients and services to make it more user-friendly. Overall, the initiative presents a viable response to a significant issue that impacts email users all around the world.

Table of Contents

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering

INTRODUCTION

With billions of emails sent and received each day, email is one of the most popular forms of communication in the world. Unfortunately, a large percentage of these emails are undesired spam, which can waste time and resources and possibly put users' security at risk. There is a need for a more effective and efficient method of classifying email spam as a result of the limitations of conventional rules-based spam filtering techniques.

Building spam classifiers that can learn from data and adjust to new types of spam has become possible thanks to machine learning, which has emerged as a promising solution to this issue. In this project, we provide a machine learning-based solution to the challenge of classifying email spam. We employ a Naive Bayes classifier with TF-IDF as the feature selection technique, trained on a dataset of preprocessed email messages.

The major objective of the study is to assess the efficiency of the suggested remedy and contrast it with current spam filters. We want to show that our method works better than conventional rules-based filters in terms of precision, effectiveness, and adaptability. In order to increase user accessibility and improve the entire email experience, we also want to investigate how to integrate our method into email clients and services.

The rest of this report is divided into the following sections: The literature on categorising email spam is reviewed in Section 2 along with any gaps in the body of knowledge. The dataset we preprocessed is described in Section 3 along with the dataset's use in the research. The Naive Bayes classifier and the feature selection approach are both part of the methodology that is described in Section 4 for the classification of spam. The findings of our studies are presented in Section 5 along with a comparison to the body of prior research. The project's major contributions are outlined in Section 6 along with topics that need further investigation.

LITERATURE SURVEY

Email spam is a recurring issue that has an impact on email users everywhere. Spam filtering has been approached in a variety of ways by researchers, from straightforward rule-based systems to more complex machine learning-based methods. In this part, we examine some of the pertinent literature on categorising email spam and point out any gaps in the body of knowledge.

In the past, rule-based strategies for spam filtering have been the most popular. These systems categorise emails as spam or not-spam based on a set of pre-established rules. However, these algorithms struggle to identify novel spam forms, and they frequently need to be updated to keep up with new spamming methods (Fette et al., 2005).

The problem of classifying email spam has found a viable solution in machine learning-based methods. To adapt to new types of spam, these strategies use a variety of machine learning algorithms that learn from the data. Due to their ease of use and effectiveness, naive Bayes classifiers are among the most widely used machine learning algorithms for spam filtering (Sahami et al., 1998).

Another crucial component of email spam classification is feature choice. The most frequently used feature selection techniques include stemming and stop-word removal in text preprocessing, as well as term frequency-inverse document frequency (TF-IDF), which identifies the most pertinent words and phrases in email messages that are most indicative of spam.

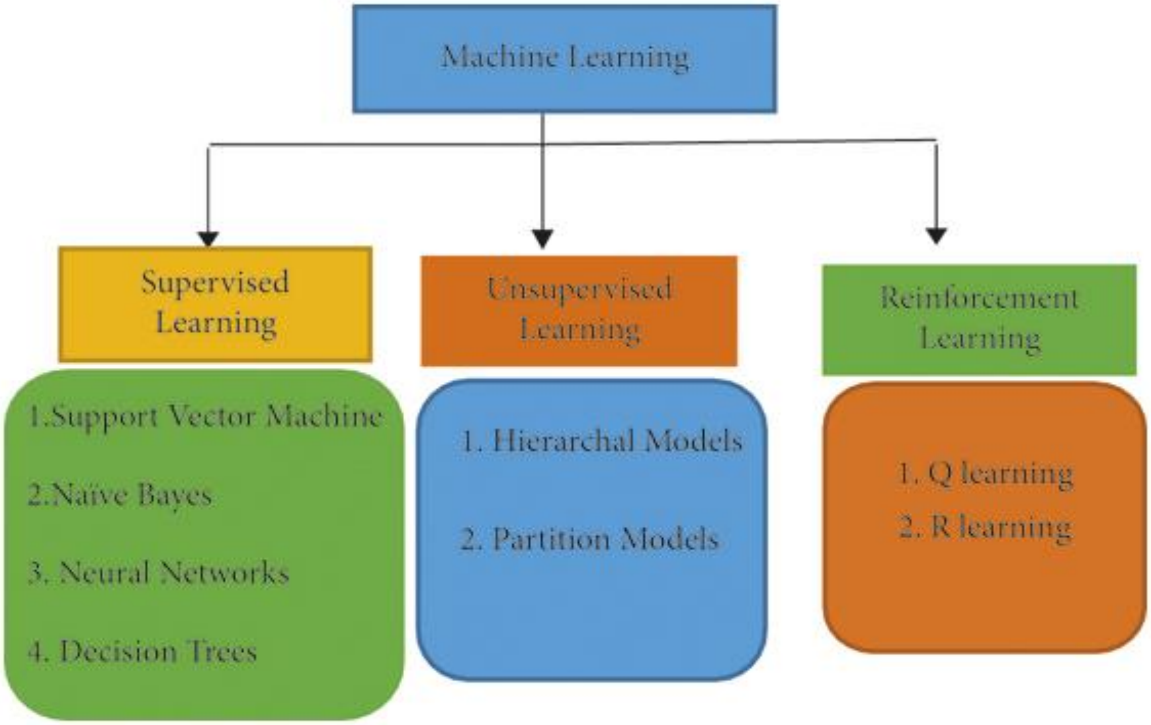
According to existing research, machine learning-based systems perform better in terms of accuracy and efficiency than conventional rule-based approaches (Almeida et al., 2011). More study is still required, though, to determine how to make these methods more efficient and scalable as well as how to incorporate them into email clients and services to improve user experience and security.

With a Naive Bayes classifier and TF-IDF as the feature selection method, we provide a machine learning-based approach to the email spam classification problem in this project. We seek to assess the efficacy of our strategy and contrast it with current spam filters while also investigating how to integrate it into email clients and services to increase user accessibility.

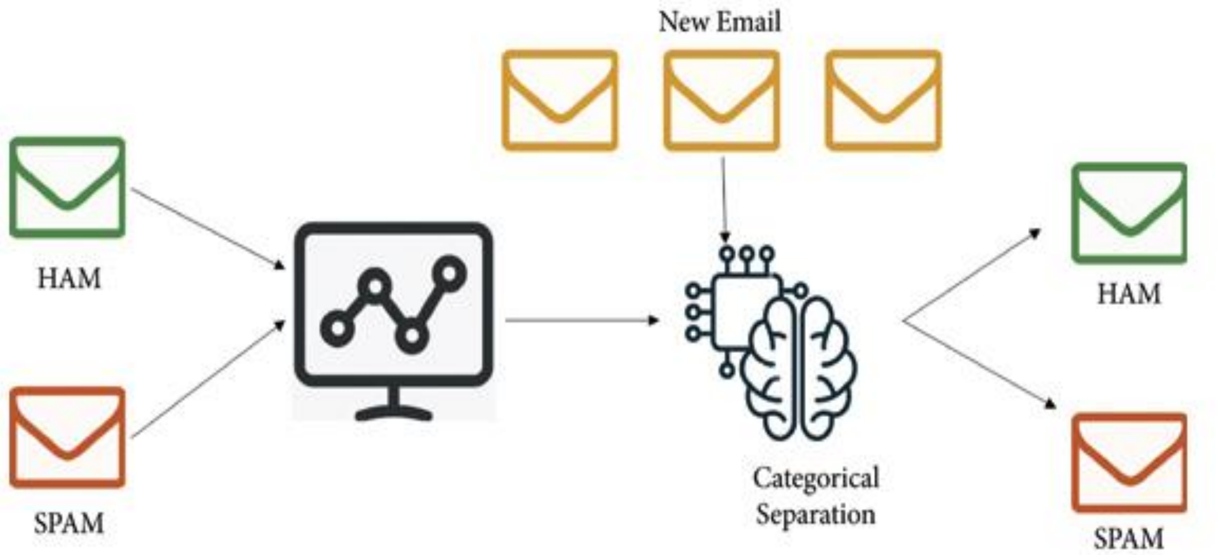
TECHNOLOGIES USED

Artificial Intelligence: The term artificial intelligence (AI) refers to the creation of computer systems that are capable of doing activities that ordinarily require human intelligence, such as speech recognition, decision-making, visual perception, and language translation. Machine learning techniques, natural language processing, computer vision, and other cutting-edge technologies are all used in AI. Healthcare, finance, transportation, and manufacturing are just a few of the sectors that AI has the potential to revolutionise. It can aid in decision-making process improvement, productivity enhancement, and automation of repetitive procedures. Insights and forecasts from AI-powered tools can also help businesses and organisations make better decisions. AI does, however, also bring up issues related to ethics, privacy, and employment displacement. As machines become more sophisticated, they might be able to carry out duties that were previously done by people, which could result in the loss of jobs in some sectors of the economy. In addition, concerns regarding privacy and civil liberties have been raised by the use of AI in applications like surveillance and facial recognition. Clear ethical standards are also required if AI is to be developed and applied properly. Despite these worries, it is likely that AI will continue to develop quickly in the years to come, spurring innovation and changing many facets of society. As a result, it's critical for everyone—individuals, companies, and governments—to keep up with the latest advancements and take into account both the advantages and disadvantages of AI.

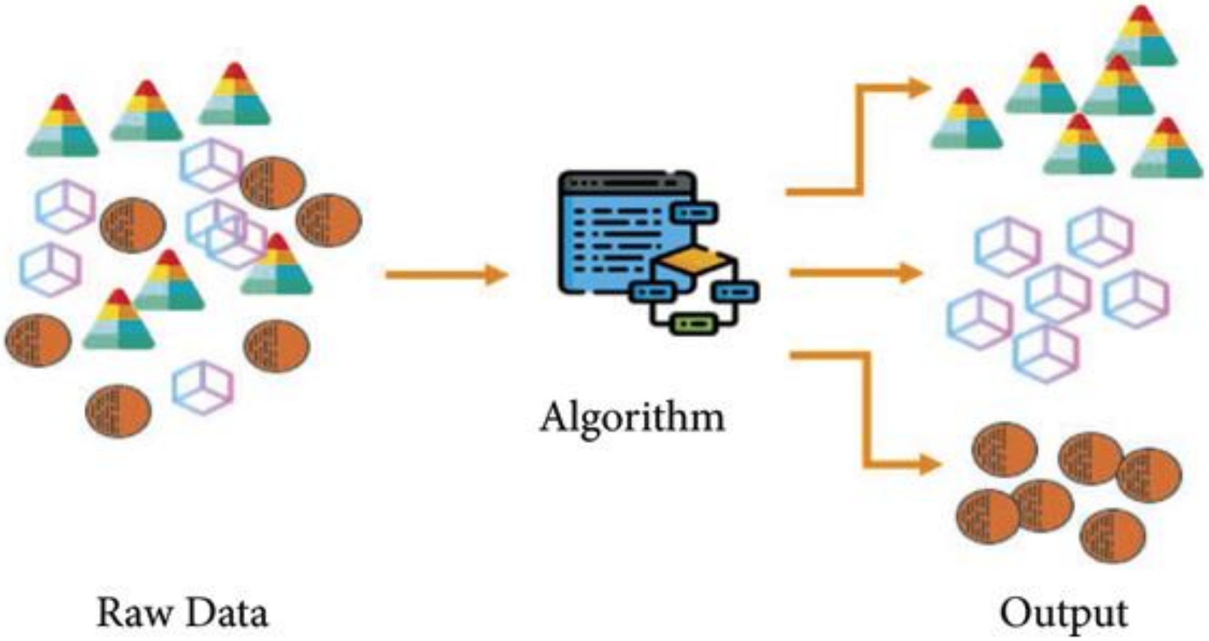
Machine Learning : The development of algorithms and statistical models that allow computer systems to gradually improve their performance on a task by analysing data is known as machine learning, and it is a subset of artificial intelligence. Machine learning aims to create models that can recognise patterns automatically and make precise predictions or judgements based on input data. This is achieved through a procedure known as training, in which a machine learning algorithm is fed a significant amount of labelled data and modifies its parameters to minimise the discrepancy between the outputs it predicts and the actual labels. From audio and picture identification to natural language processing and fraud detection, machine learning is used in a wide range of industries. Additionally, it is utilised in recommender systems, which provide recommendations for goods, services, or information based on a user's past actions or interests. The ability of machine learning to automate processes that would otherwise require human expertise or intuition is one of its main benefits. To identify probable tumours or other abnormalities in medical pictures, for instance, or to spot fraudulent transactions in financial data, machine learning algorithms can be used. The danger of bias and errors if the training data is not representative or the algorithm is poorly built are potential drawbacks to machine learning, though. Furthermore, it may not always be clear how machine learning algorithms make their decisions, making it challenging to comprehend how they work. Despite these difficulties, machine learning is likely to keep playing a significant role in many fields and applications as data availability and processing power increase.



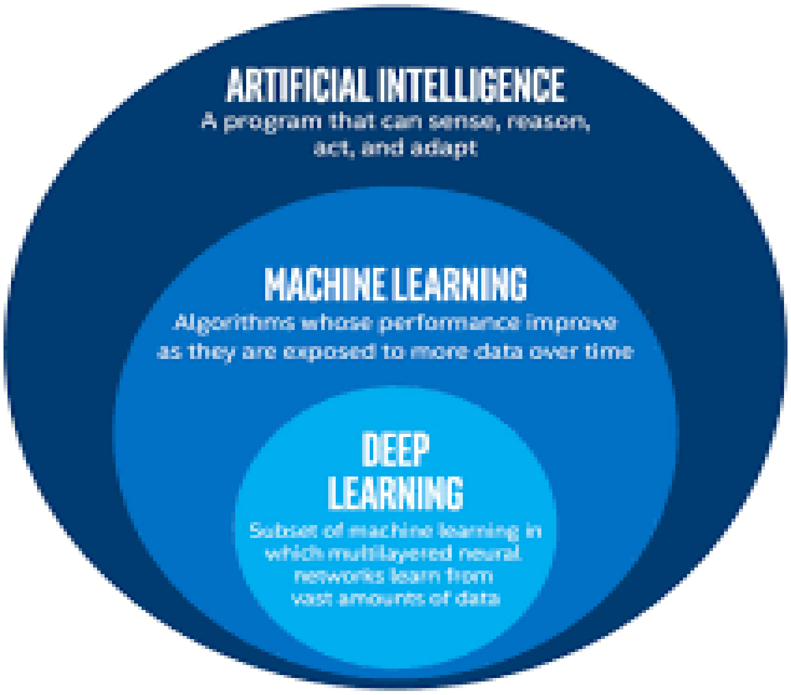
- *Suprvised Learning:* The algorithm is trained on labelled data, meaning that each data point is connected to a specified goal or output variable, in supervised machine learning. A mapping from input variables to output variables that may be utilised to generate predictions or choices on brand-new, untainted data is what supervised learning aims to achieve. In supervised learning, the training procedure entails giving the algorithm a series of labelled instances, known as the training set, and modifying the model's parameters to minimise the discrepancy between its anticipated outputs and the actual labels. A loss function is frequently used in this procedure to calculate the error between the projected outputs and the actual labels. By applying the learnt mapping to the input variables, the model can be used to make predictions on fresh, unobserved data once it has been trained. A different collection of labelled samples known as the validation set or test set can be used to assess the model's predictions' accuracy. Classification, regression, and time series forecasting are just a few of the many tasks that may be accomplished with supervised learning. While the objective of regression tasks is to predict a continuous numerical value, the objective of classification tasks is to predict a discrete class label for each input data point. In time series forecasting, future values of a time-dependent variable are predicted using data from the past.



- *Unsupervised Learning*: Unsupervised machine learning is a subset of machine learning in which the algorithm is trained on data that has not been labelled, meaning that each data point does not have a goal or output variable. Without any prior knowledge of the output, the objective of unsupervised learning is to find underlying patterns or structures in the data. In unsupervised learning, the algorithm is trained by giving it a set of unlabeled examples and then utilising statistical methods to find patterns or clusters in the data. This procedure is frequently carried out using dimensionality reduction techniques, which find a lower-dimensional representation of the data that captures its fundamental structure, or clustering algorithms, which group similar data points together based on their proximity in the data space. The method can be used to forecast or decide on fresh, unobserved data by using the same clustering or dimensionality reduction techniques after it has found patterns or clusters in the data. The purity or silhouette score of the recognised clusters are two examples of metrics that can be used to assess how accurately the model's predictions are made. Unsupervised learning has the ability to find intricate patterns or structures in the data without having any prior knowledge of the outcome, which is one of its main benefits. The performance of supervised learning algorithms can be enhanced by preprocessing or transforming the data using unsupervised learning.



Deep Learning: The term "deep" refers to the possibility of these networks having a large number of hidden layers between the input and output layers, enabling them to capture more abstract and high-level representations of the input data. Deep learning's capacity to automatically learn features and representations from unprocessed data, without the need for explicit feature engineering, is one of its main advantages. In areas like computer vision, natural language processing, and speech recognition, where deep learning models have attained cutting-edge performance on a variety of benchmark tests, this has resulted in considerable advancements.

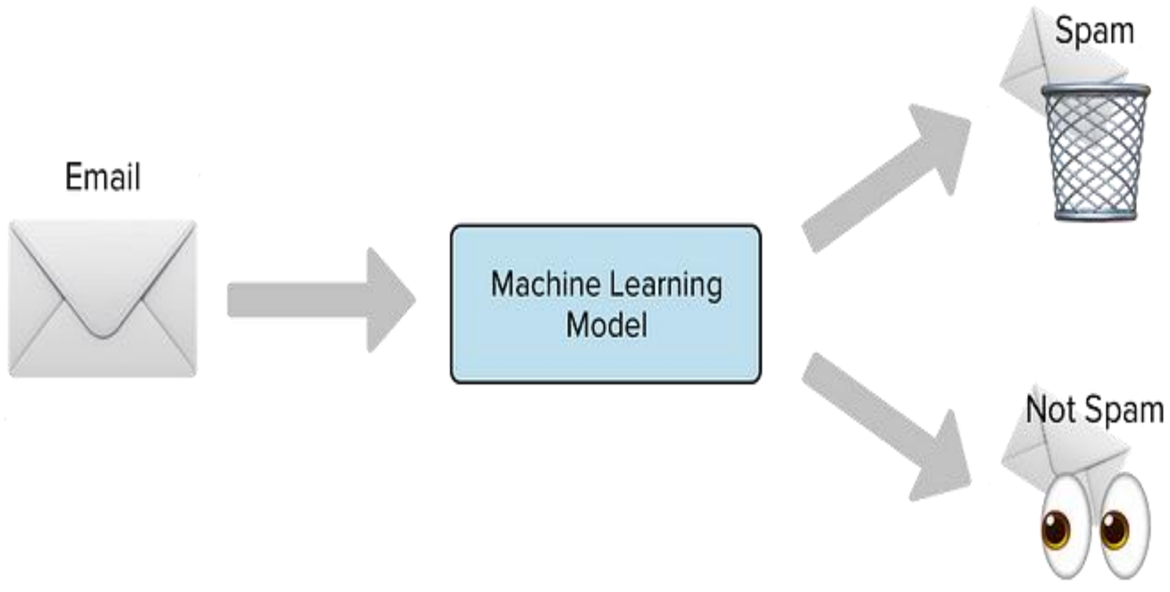

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

Naïve Bays Classification: A popular probabilistic machine learning approach for classification applications, such as spam filtering or sentiment analysis, is naive Bayes classification. The Bayes theorem, which offers a mechanism to determine the likelihood of a hypothesis based on past information and evidence, is the foundation of the algorithm. When using Naive Bayes classification, the data point's class label (such as "spam" or "non-spam") serves as the hypothesis, and its observable properties (such as the words in an email message) serve as the evidence. Given the observed features, the algorithm determines the likelihood of each potential class label before assigning the data point to the most likely class. Naive Bayes classification can be useful in many situations even though this assumption may not hold true in practise, especially when the number of features is high compared to the number of data points. The multinomial Naive Bayes classifier, which is frequently used for text classification, and the gaussian Naive Bayes classifier, which is used when the features are continuous variables, are two of the different Naive Bayes classification versions. The ease and effectiveness of Naive Bayes classification, particularly for high-dimensional data, is one of its main advantages. The technique is easily adaptable to new classes or features and can be trained quickly on huge datasets.

Here are Some Calculation Which help you to Understand how it work.

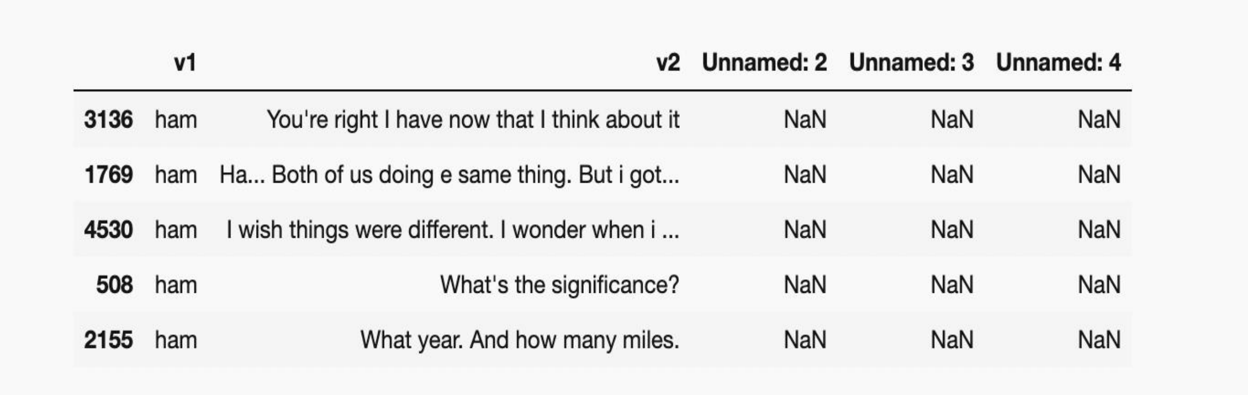
$$S\,[T] = \frac{C_{Spam}(T)}{C_{Spam}(T) + C_{Ham}(T)}$$

Where CSpam(T) and CHam(T) denote, respectively, the quantity of spam and ham messages containing token T. To determine the likelihood of a message M containing the tokens "T1,...," "TN," one must add the spamminess of each token to determine the overall spamminess of the message. Calculating the product of each token's spamminess and comparing it to the product of each token's hamminess is a quick and easy approach to classify tokens.



METHODOLOGY

For this specific experiment, we'll be using the English-language SMS-tagged messages from the Kaggle SMS Spam Collection Dataset, which has been divided into ham (or "legitimate") and spam categories. One message is contained per line. Each line has two columns: v1, which contains the label (such as "spam" or "ham"), and v2, which contains the actual content.



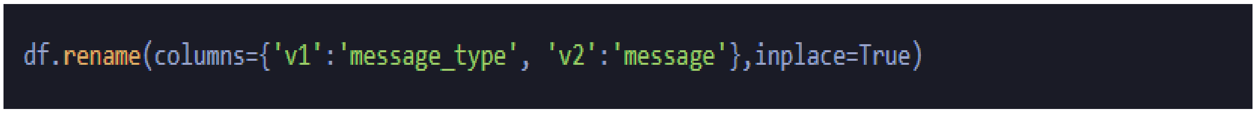
	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
3136	ham	You're right I have now that I think about it	NaN	NaN	NaN
1769	ham	Ha... Both of us doing e same thing. But i got...	NaN	NaN	NaN
4530	ham	I wish things were different. I wonder when i ...	NaN	NaN	NaN
508	ham	What's the significance?	NaN	NaN	NaN
2155	ham	What year. And how many miles.	NaN	NaN	NaN

Output of Dataset

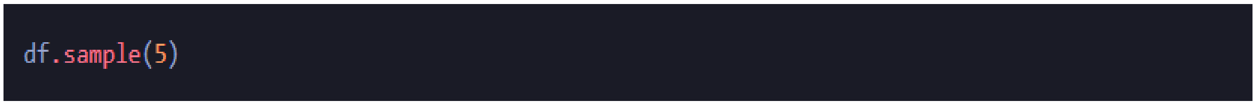
Let's now go to work on the dataset and create some stunning visualisations and inferences.

Data Cleaning:

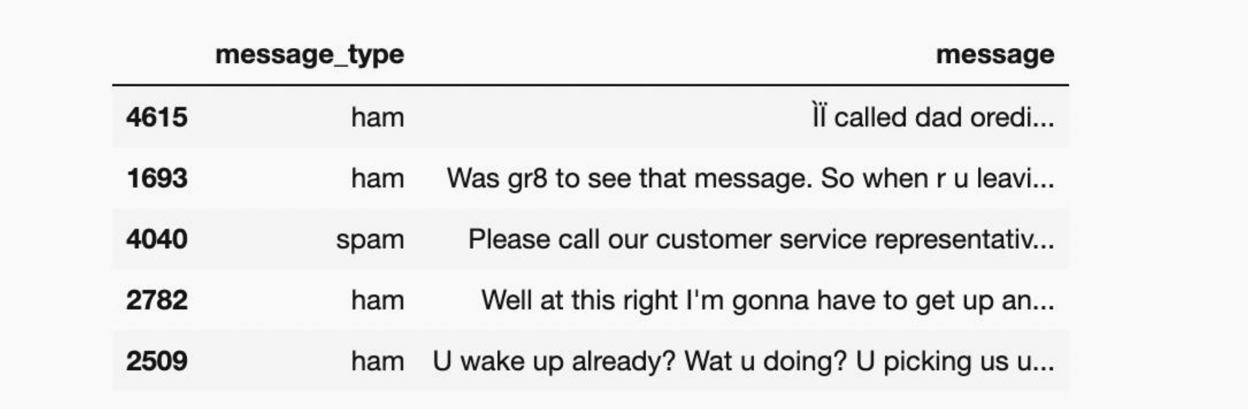
```
df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace= True)
df.sample(5)
```



```
df.rename(columns={'v1':'message_type', 'v2':'message'},inplace=True)
```



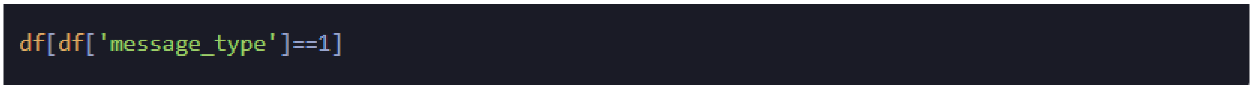
```
df.sample(5)
```



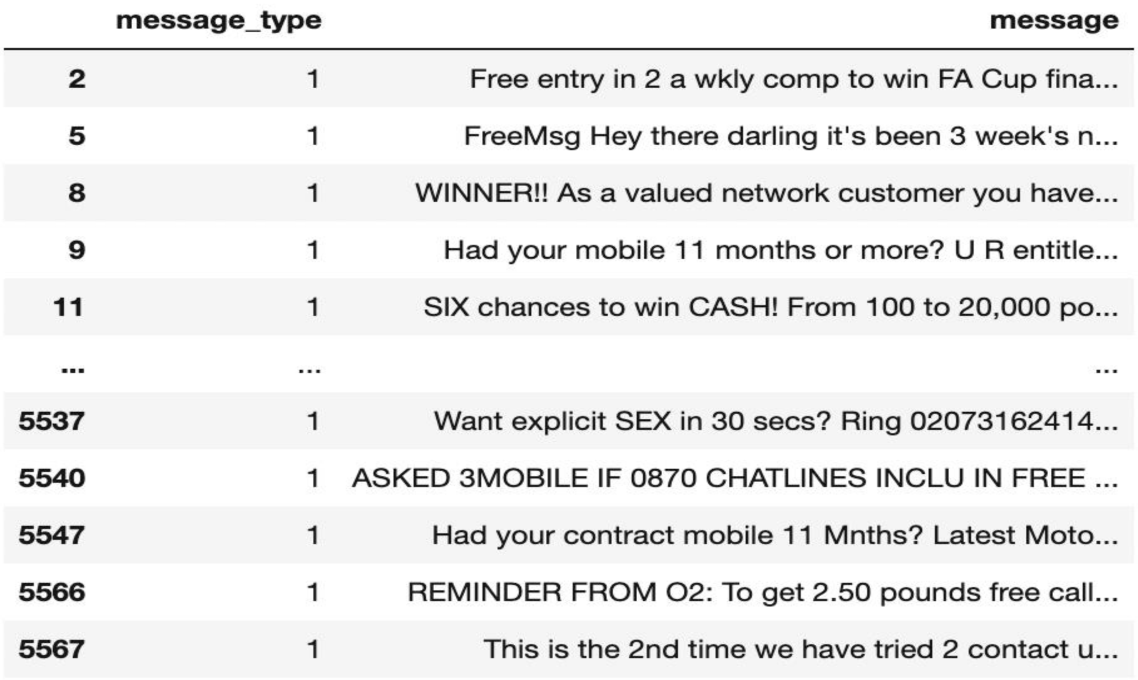
	message_type	message
4615	ham	ÌÏ called dad oredi...
1693	ham	Was gr8 to see that message. So when r u leavi...
4040	spam	Please call our customer service representativ...
2782	ham	Well at this right I'm gonna have to get up an...
2509	ham	U wake up already? Wat u doing? U picking us u...

We use a label encoder because this is a classification problem and we want the "message_type" to be binary categorised, meaning it can be either 0 or 1.

```
from sklearn.preprocessing import LabelEncoder
encoder =LabelEncoder()
df['message_type']=encoder.fit_transform(df['message_type'])
df['message_type'].sample(5)
```

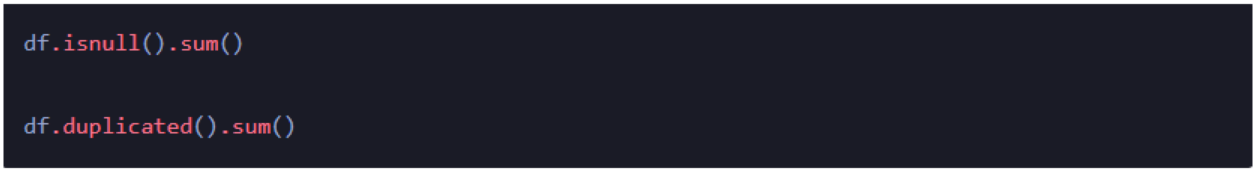


```
df[df['message_type']==1]
```



	message_type	message
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
5	1	FreeMsg Hey there darling it's been 3 week's n...
8	1	WINNER!! As a valued network customer you have...
9	1	Had your mobile 11 months or more? U R entitle...
11	1	SIX chances to win CASH! From 100 to 20,000 po...
...	...	...
5537	1	Want explicit SEX in 30 secs? Ring 02073162414...
5540	1	ASKED 3MOBILE IF 0870 CHATLINES INCLU IN FREE ...
5547	1	Had your contract mobile 11 Mnths? Latest Moto...
5566	1	REMINDER FROM O2: To get 2.50 pounds free call...
5567	1	This is the 2nd time we have tried 2 contact u...

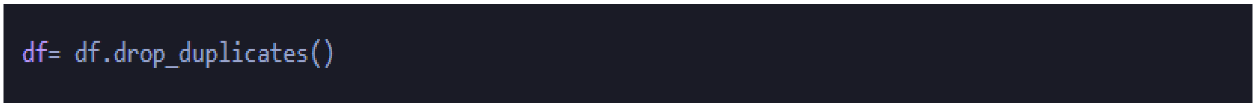
Let's now look for any missing values.



```
df.isnull().sum()

df.duplicated().sum()
```

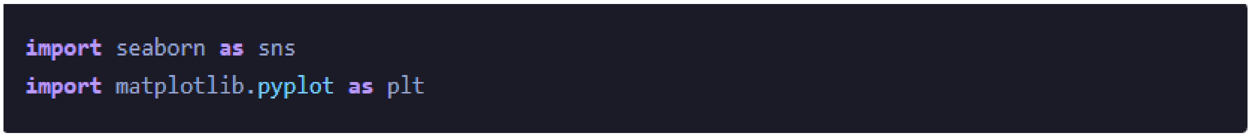
There are 403 duplicate values, which we must eliminate.



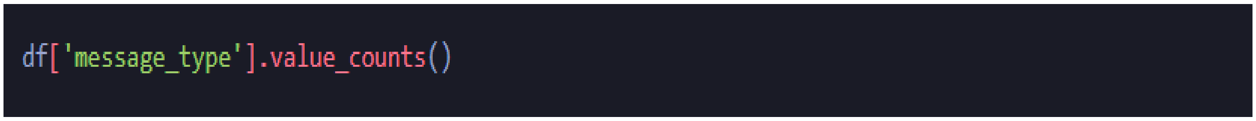
```
df= df.drop_duplicates()
```

Data Analysis:

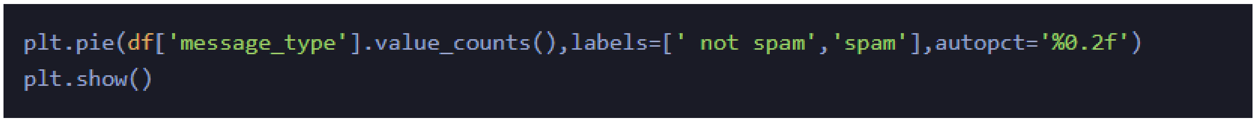
To better grasp the data, let's depict the categorization challenge.



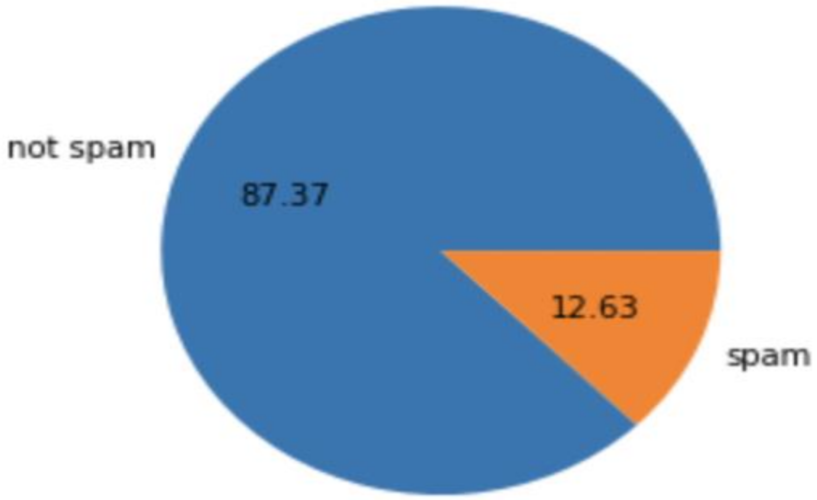
```
import seaborn as sns
import matplotlib.pyplot as plt
```



```
df['message_type'].value_counts()
```



```
plt.pie(df['message_type'].value_counts(),labels=[' not spam','spam'],autopct='%0.2f')
plt.show()
```



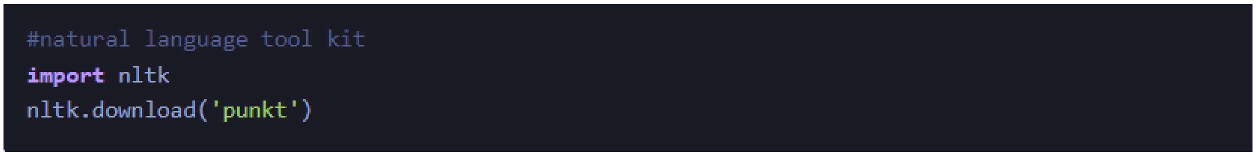
This is an imbalanced data

Now let’s find out

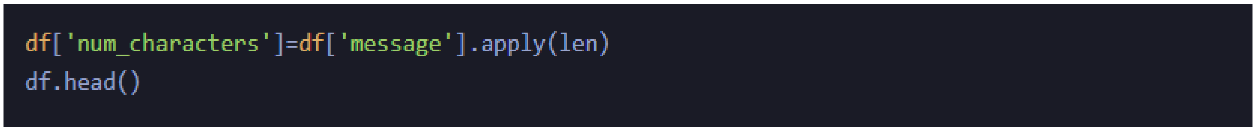
- No. of Characters in the Data.
- No. of Words in the Data.
- No. of Sentences in the Data

and create three additional columns in the data to represent the number of characters, words, and sentences.

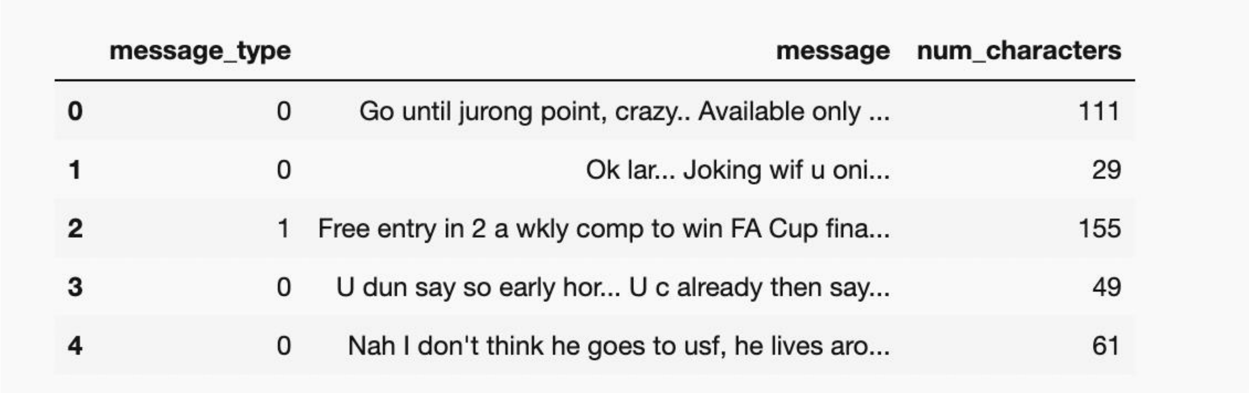
For a Number of Characters



```
#natural language tool kit
import nltk
nltk.download('punkt')
```

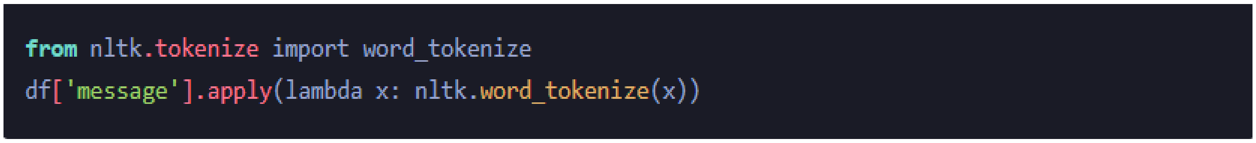


```
df['num_characters']=df['message'].apply(len)
df.head()
```

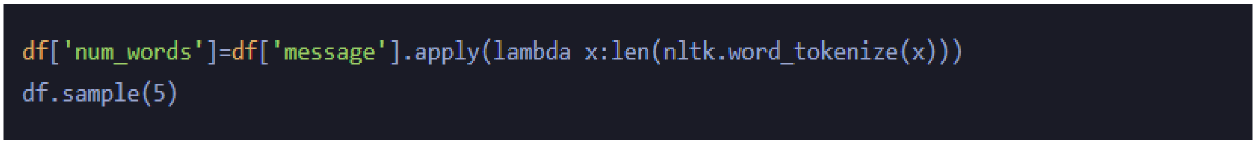


	message_type	message	num_characters
0	0	Go until jurong point, crazy.. Available only ...	111
1	0	Ok lar... Joking wif u oni...	29
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	0	U dun say so early hor... U c already then say...	49
4	0	Nah I don't think he goes to usf, he lives aro...	61

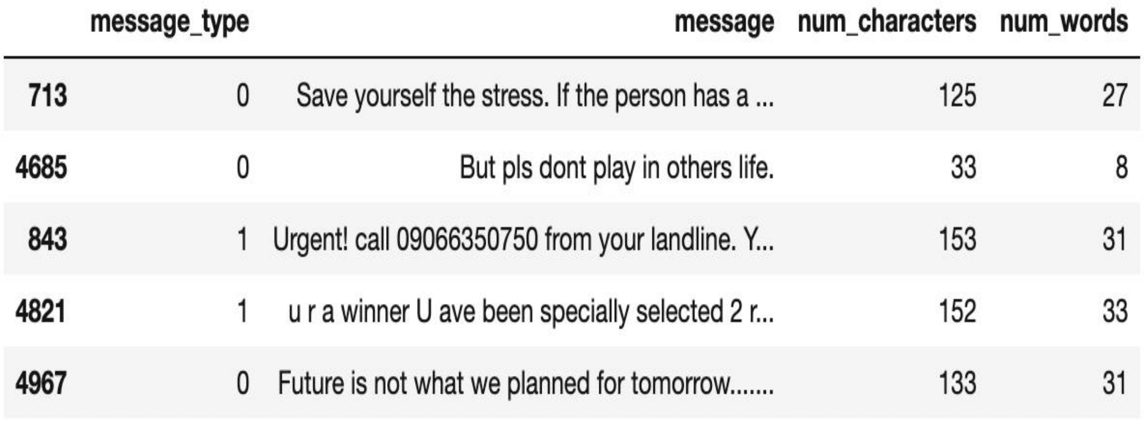
For the No. of Words



```
from nltk.tokenize import word_tokenize
df['message'].apply(lambda x: nltk.word_tokenize(x))
```



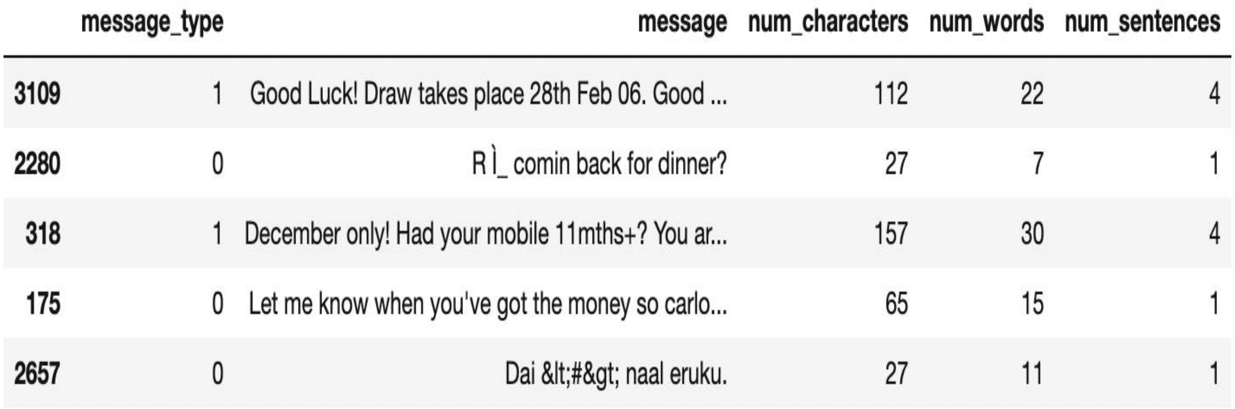
```
df['num_words']=df['message'].apply(lambda x:len(nltk.word_tokenize(x)))
df.sample(5)
```



	message_type	message	num_characters	num_words
713	0	Save yourself the stress. If the person has a ...	125	27
4685	0	But pls dont play in others life.	33	8
843	1	Urgent! call 09066350750 from your landline. Y...	153	31
4821	1	u r a winner U ave been specially selected 2 r...	152	33
4967	0	Future is not what we planned for tomorrow.......	133	31

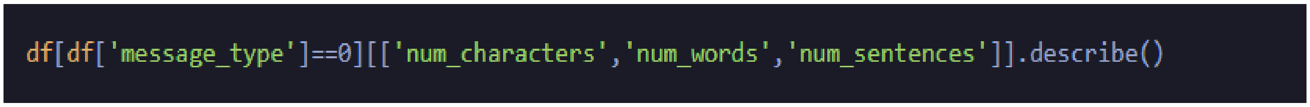
For a No. of Sentences

```
df['num_sentences']=df['message'].apply(lambd x:
len(nltk.sent_tokenize(x)))
```

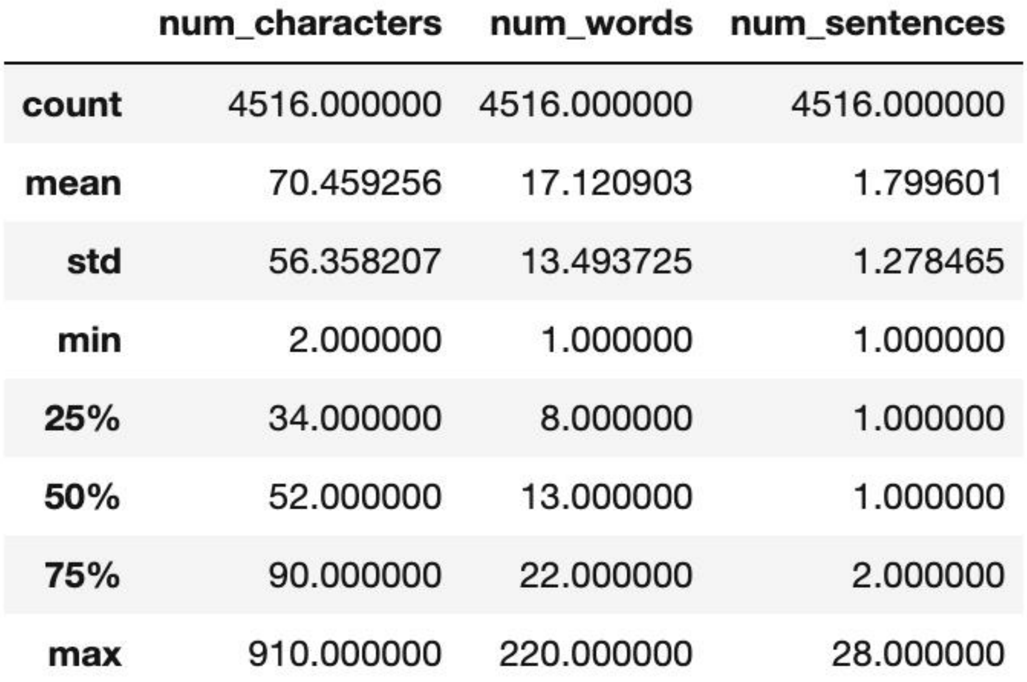


	message_type	message	num_characters	num_words	num_sentences
3109	1	Good Luck! Draw takes place 28th Feb 06. Good ...	112	22	4
2280	0	R Ì_ comin back for dinner?	27	7	1
318	1	December only! Had your mobile 11mths+? You ar...	157	30	4
175	0	Let me know when you've got the money so carlo...	65	15	1
2657	0	Dai &lt;#&gt; naal eruku.	27	11	1

For HAM messages



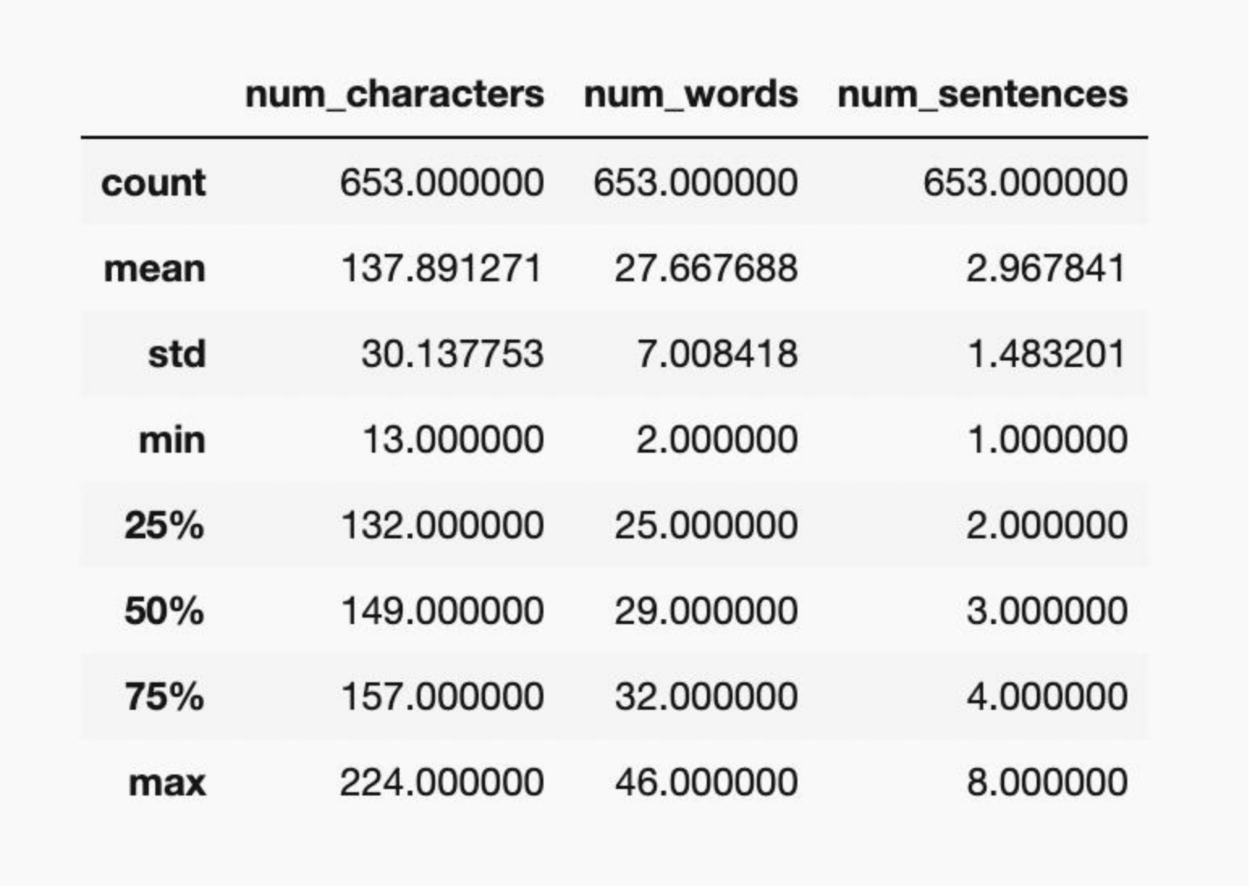
```
df[df['message_type']==0][['num_characters','num_words','num_sentences']].describe()
```



	num_characters	num_words	num_sentences
count	4516.000000	4516.000000	4516.000000
mean	70.459256	17.120903	1.799601
std	56.358207	13.493725	1.278465
min	2.000000	1.000000	1.000000
25%	34.000000	8.000000	1.000000
50%	52.000000	13.000000	1.000000
75%	90.000000	22.000000	2.000000
max	910.000000	220.000000	28.000000

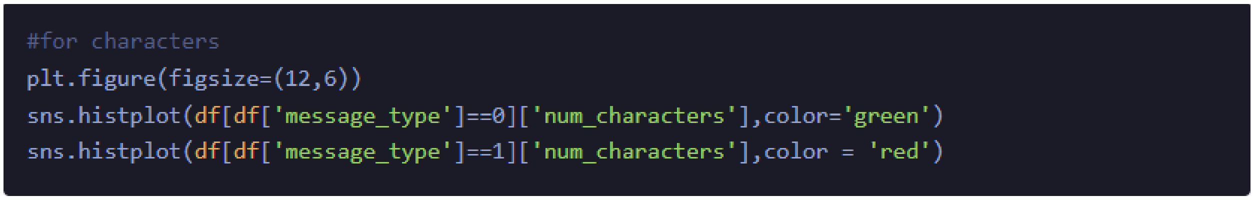
For SPAM messages:

```
df[df['message_type']==1][['num_characters','num_words','num_sentences']].describe()
```

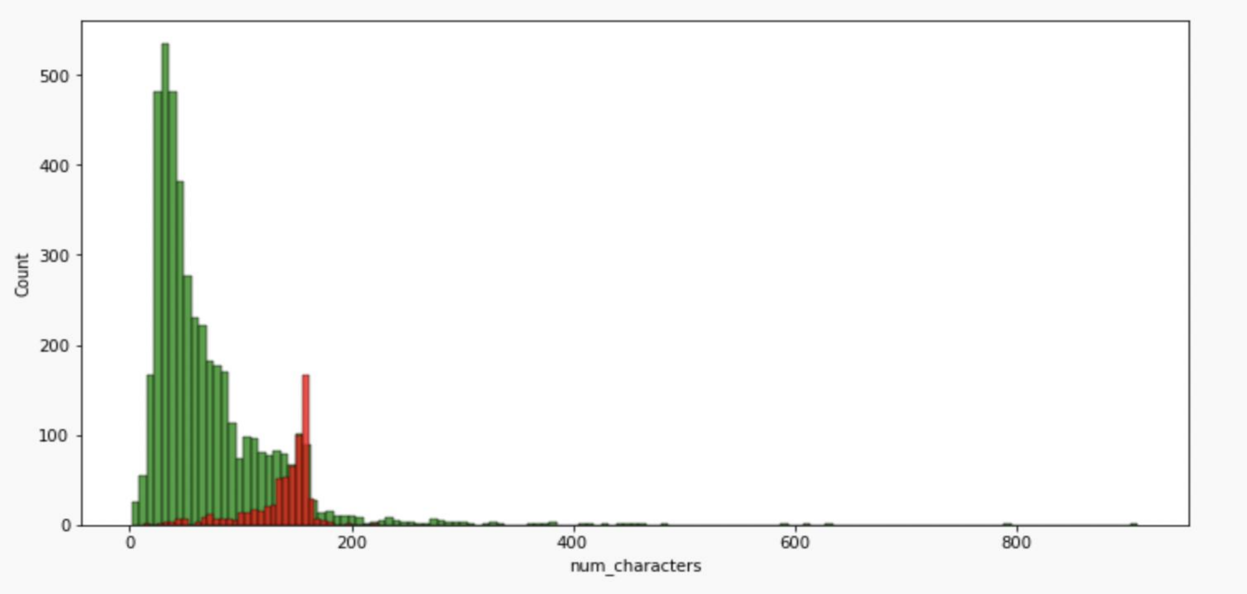


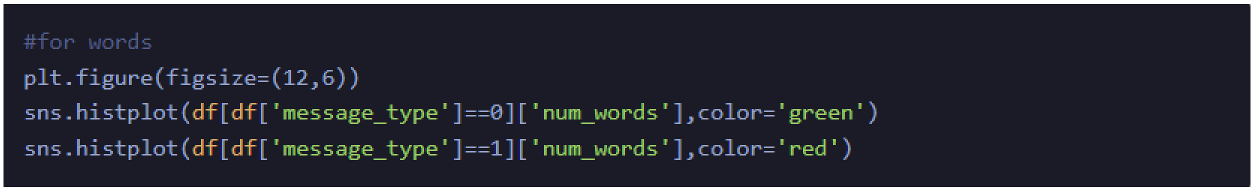
	num_characters	num_words	num_sentences
count	653.000000	653.000000	653.000000
mean	137.891271	27.667688	2.967841
std	30.137753	7.008418	1.483201
min	13.000000	2.000000	1.000000
25%	132.000000	25.000000	2.000000
50%	149.000000	29.000000	3.000000
75%	157.000000	32.000000	4.000000
max	224.000000	46.000000	8.000000

It is obvious that spam communications are considerably longer than ham messages.

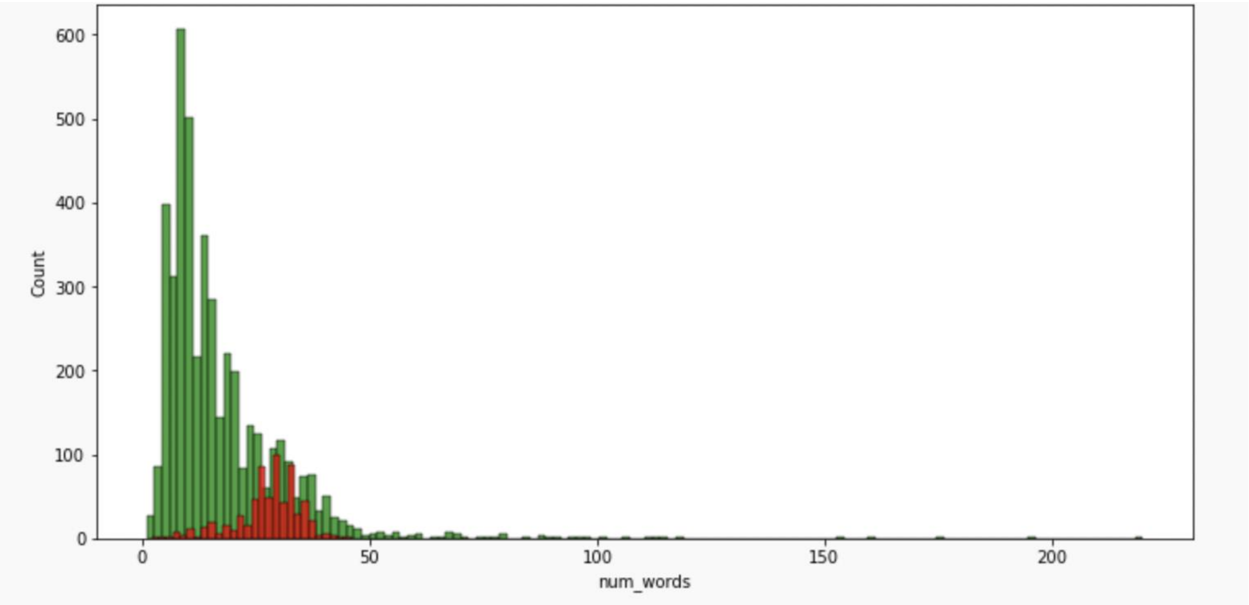


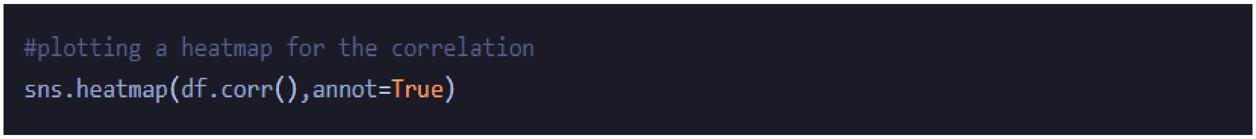
```
#for characters
plt.figure(figsize=(12,6))
sns.histplot(df[df['message_type']==0]['num_characters'],color='green')
sns.histplot(df[df['message_type']==1]['num_characters'],color = 'red')
```



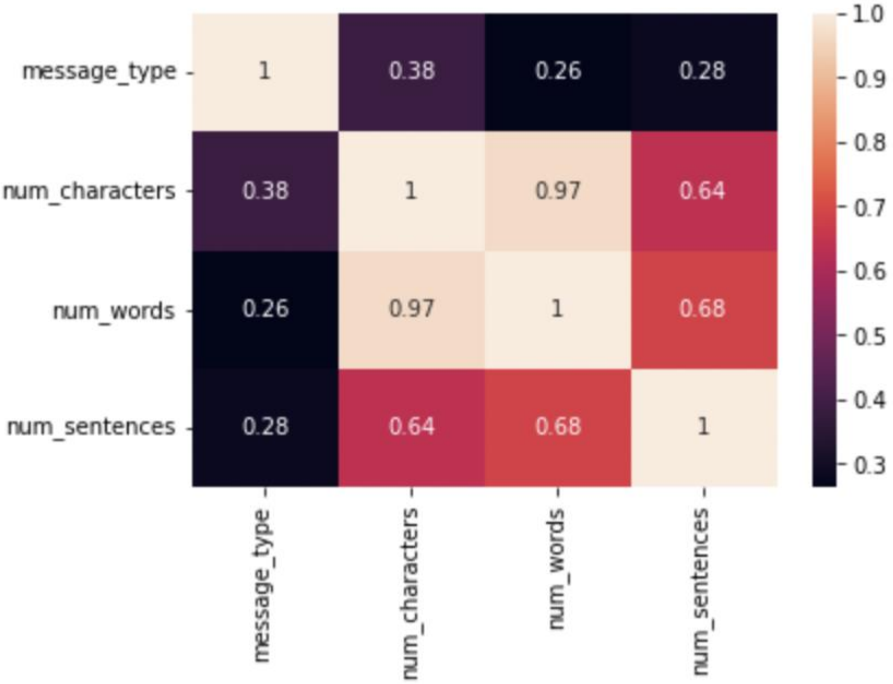


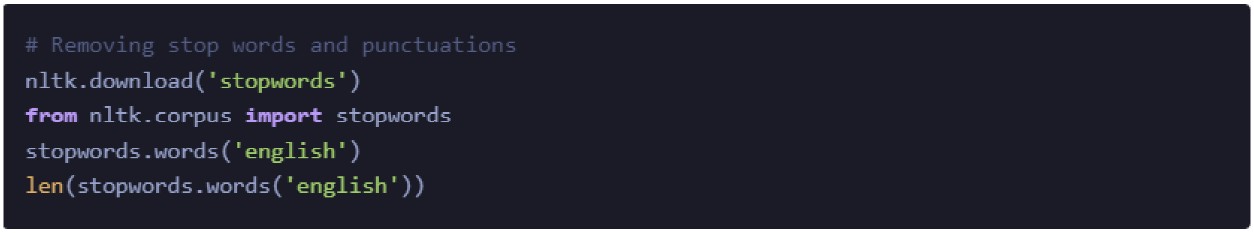
```
#for words
plt.figure(figsize=(12,6))
sns.histplot(df[df['message_type']==0]['num_words'],color='green')
sns.histplot(df[df['message_type']==1]['num_words'],color='red')
```



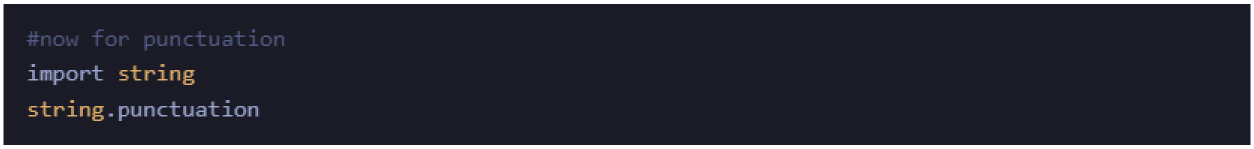


```
#plotting a heatmap for the correlation
sns.heatmap(df.corr(),annot=True)
```

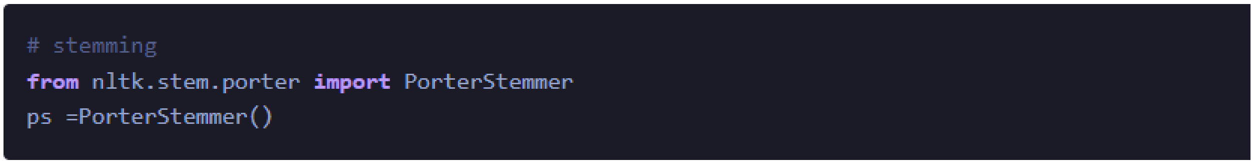




```
# Removing stop words and punctuations
nltk.download('stopwords')
from nltk.corpus import stopwords
stopwords.words('english')
len(stopwords.words('english'))
```

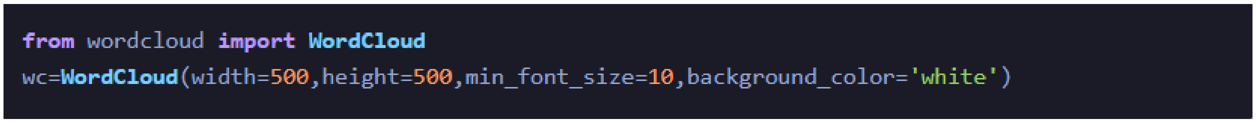


```
#now for punctuation
import string
string.punctuation
```



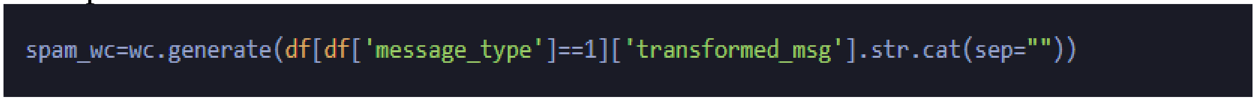
```
# stemming
from nltk.stem.porter import PorterStemmer
ps =PorterStemmer()
```

We create a word cloud in order to clearly understand the most often used words.

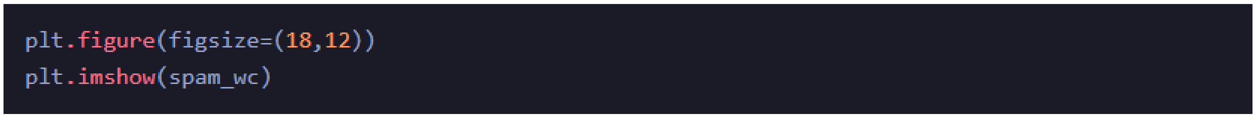


```
from wordcloud import WordCloud
wc=WordCloud(width=500,height=500,min_font_size=10,background_color='white')
```

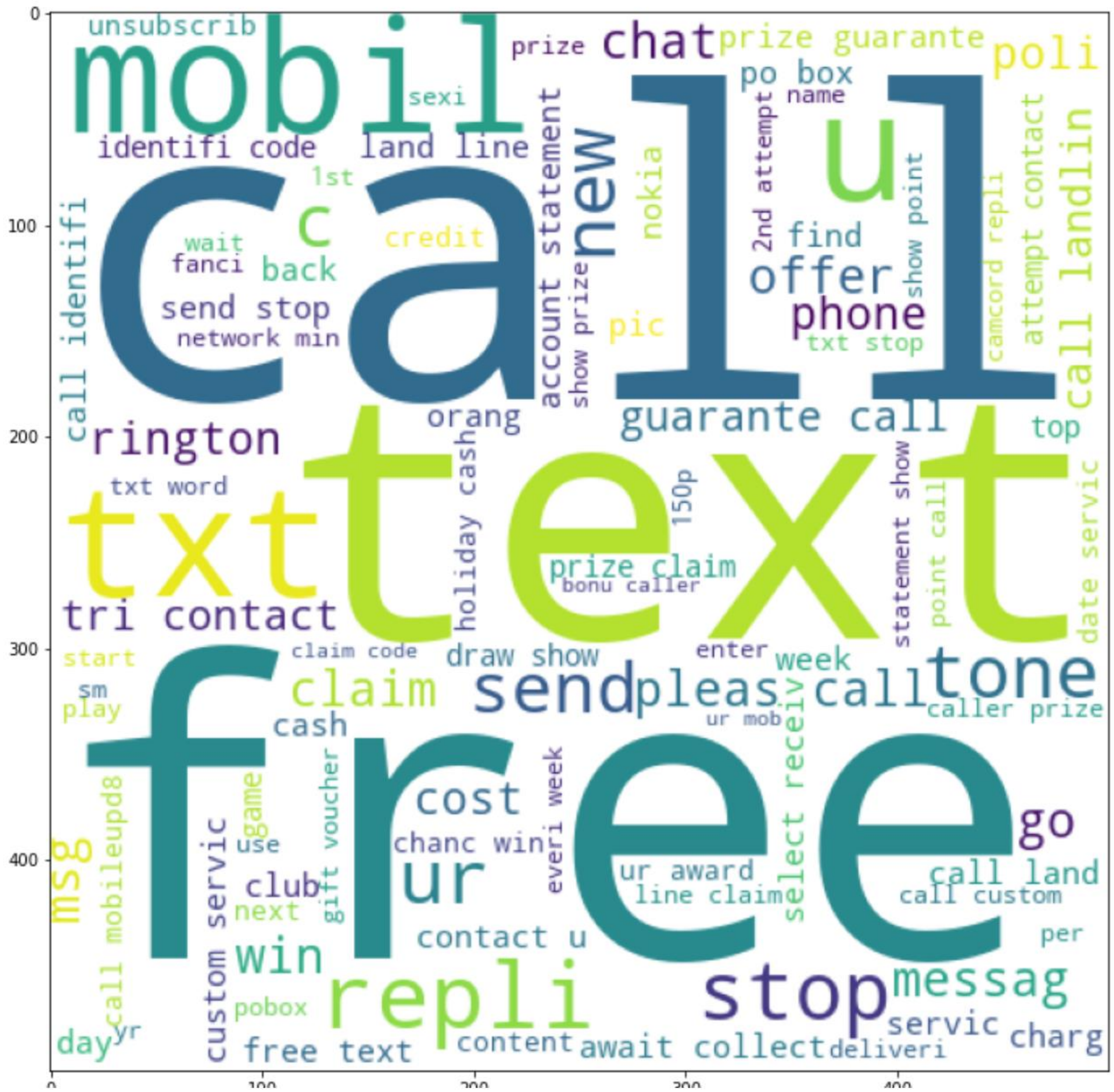
For Spam:



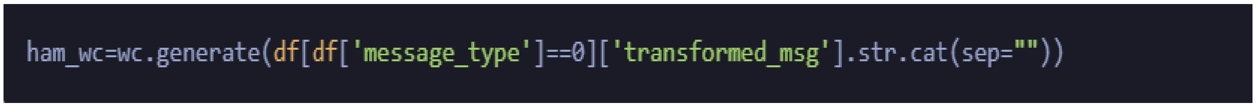
```
spam_wc=wc.generate(df[df['message_type']==1]['transformed_msg'].str.cat(sep=""))
```



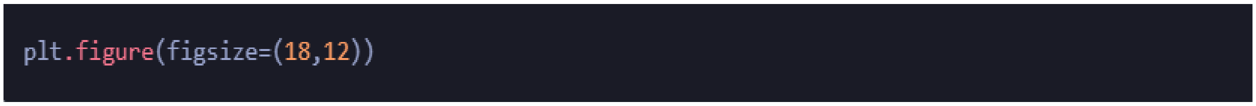
```
plt.figure(figsize=(18,12))
plt.imshow(spam_wc)
```



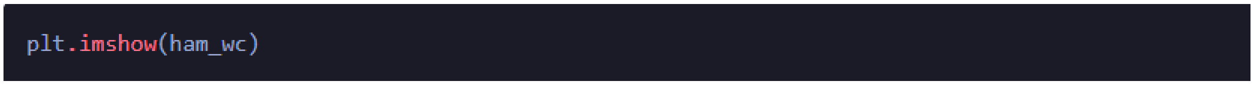
For Ham:



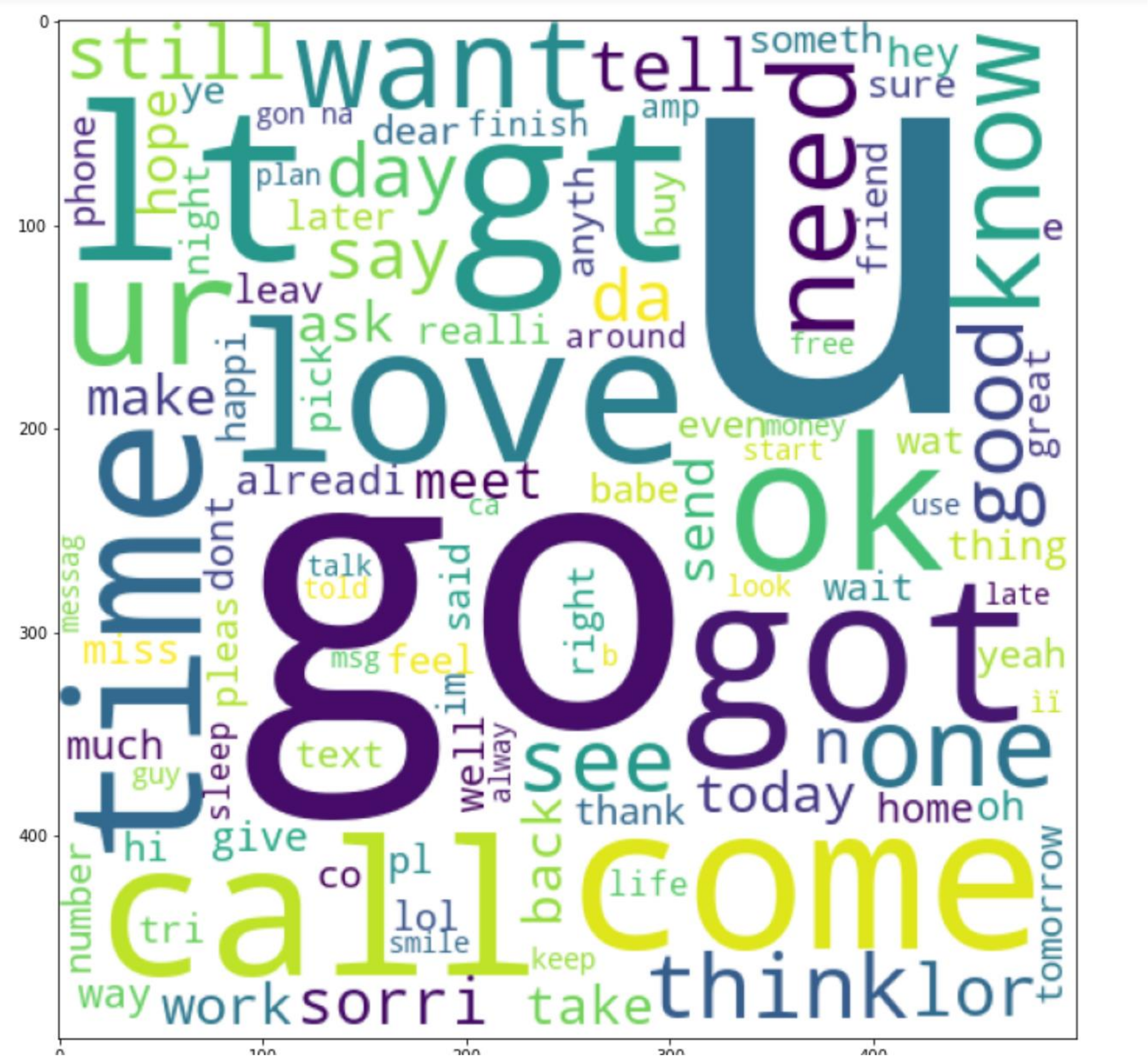
```
ham_wc=wc.generate(df[df['message_type']==0]['transformed_msg'].str.cat(sep=""))
```



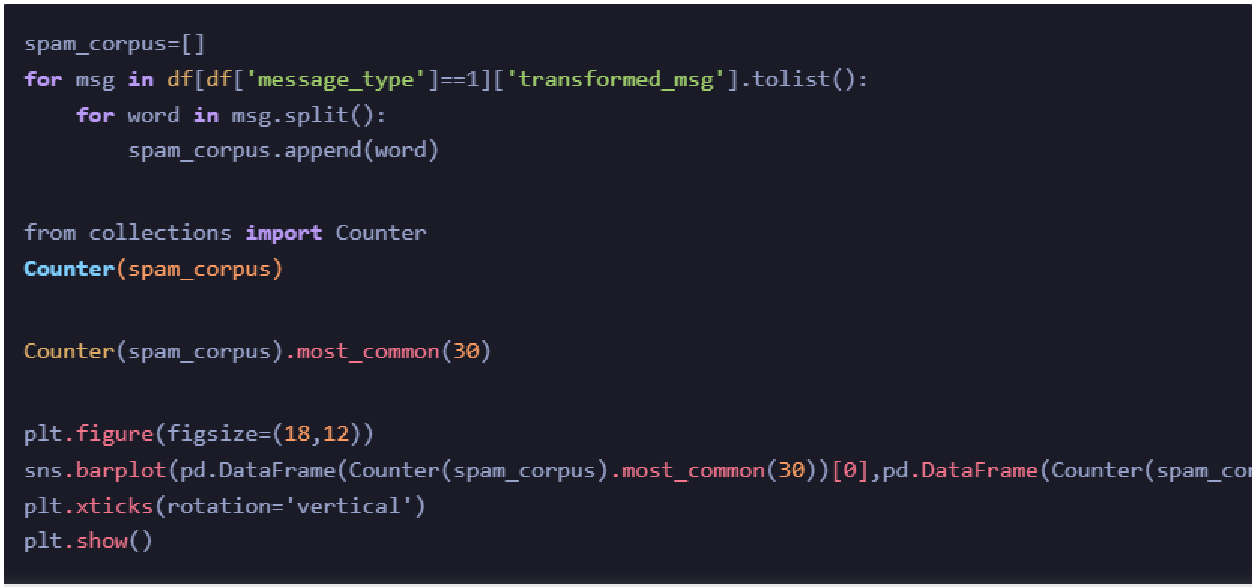
```
plt.figure(figsize=(18,12))
```



```
plt.imshow(ham_wc)
```



We will identify the top 30 words used in both spam and ham communications in order to condense what is depicted in the WordClouds.

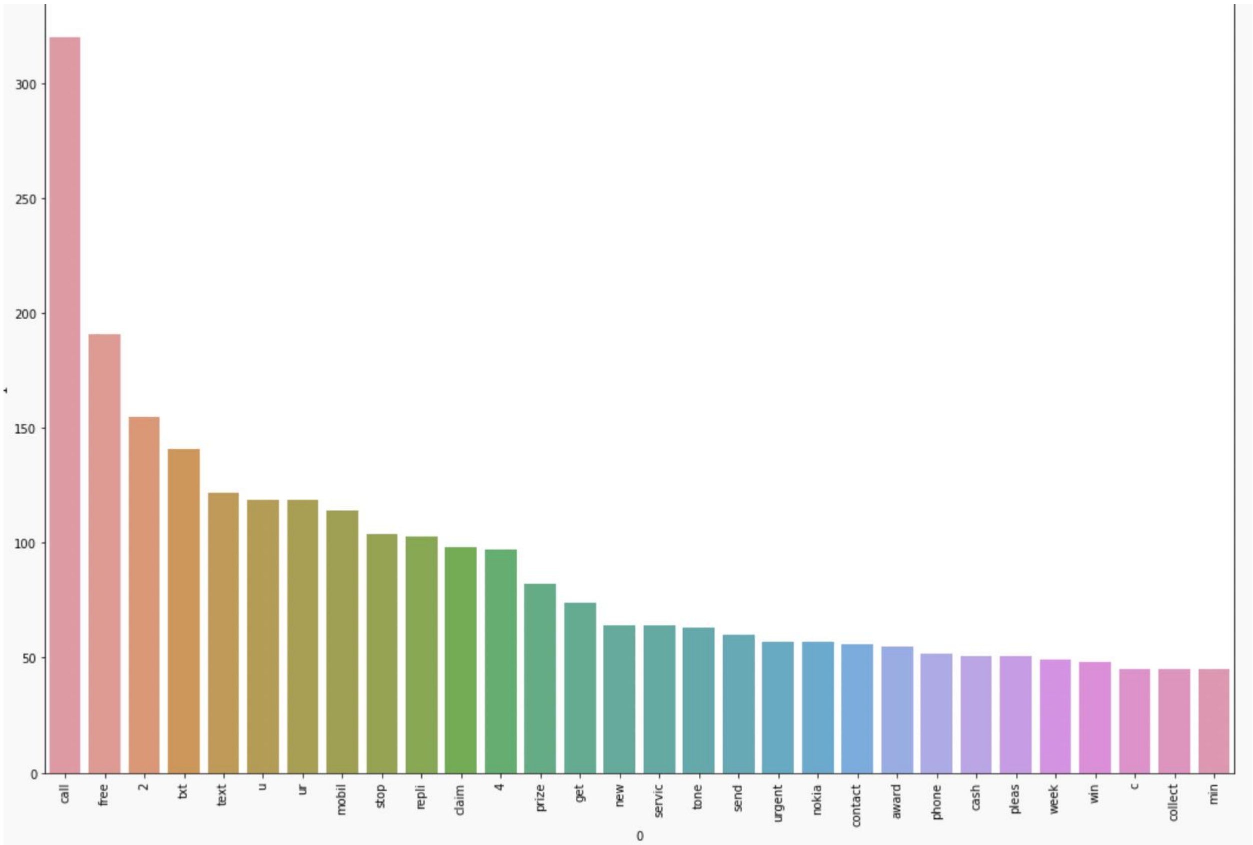


```
spam_corpus=[]
for msg in df[df['message_type']==1]['transformed_msg'].tolist():
    for word in msg.split():
        spam_corpus.append(word)

from collections import Counter
Counter(spam_corpus)

Counter(spam_corpus).most_common(30)

plt.figure(figsize=(18,12))
sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0],pd.DataFrame(Counter(spam_co
plt.xticks(rotation='vertical')
plt.show()
```



USE CASE

Consider that you are employed by a sizable email company that processes millions of emails daily. Detecting and removing spam emails to maintain a good email experience for your consumers is one of your biggest difficulties.

You choose to use machine learning to create an email spam classification system in order to solve this problem. You begin by compiling a sizable dataset of classified emails, some of which are classified as spam and others as non-spam.

After that, the data is preprocessed by eliminating stop words, stemming words, and transforming the text into numerical representations like bag-of-words or TF-IDF vectors. The data was also divided into training and testing sets.

The training data is then used to train a number of machine learning models, such as a neural network, SVMs, and Naive Bayes. Following model training, you assess the models' performance on the testing set and choose the model that performs the best based on factors like accuracy, precision, and recall.

After deciding which model performs the best, you include it into your email system to automatically categorise incoming emails as spam or not. In order to keep the model current with the changing characteristics of spam emails, you also put up a system to retrain the model on fresh data on a regular basis.

In order to maintain the email spam classification system's accuracy and effectiveness, you monitor its operation and solicit user feedback. By putting this system into place, you may improve the email experience for your consumers by filtering out undesired spam emails.

FUTURE SCOPE

1. Multilingual classification: The majority of research on email spam classification has been on emails sent in English. However, techniques for classifying email spam that can handle emails in many languages are required. The use of multilingual embeddings and other techniques for creating email spam categorization systems that can handle emails in different languages is a topic for future research.

2. Image-based spam detection: As the prevalence of image-based spam grows, email spam classification systems that can recognise spam messages with images are required. The use of picture recognition methods, such as deep learning-based systems, for identifying image-based spam communications can be explored in further research.

3. Attacks from the enemy: Attacks from the enemy can be used to trick email spam classification systems into classifying spam communications as non-spam and vice versa. Future studies may examine the creation of reliable, attack-resistant email spam classification systems.

4. Integration of user feedback: While email spam classification systems based on machine learning can reach high accuracy, they are not flawless. Users still risk receiving spam or having non-spam mail labelled as such. In order to increase the accuracy of email spam classification algorithms, future research can examine how to include user feedback.

5. Email spam classification that protects user privacy: Most email spam classification systems access the contents of users' emails, which presents privacy issues. Future study can look into the creation of email spam classification systems that respect user privacy and can classify emails. Techniques like differential privacy and homomorphic encryption may be used in this.

CONCLUSION

In conclusion, classifying email spam is a crucial duty for ensuring that users have a great email experience. Automatically classifying emails as spam or not has showed promise when using machine learning-based methods. In this project report, we looked into the different machine learning methods and approaches that can be applied to classify emails as spam and assessed how well they performed using a dataset of labelled emails. The accuracy, precision, and recall of the SVM classifier with TF-IDF feature representation were the best, according to our experimental findings.

However, there are still a number of areas in email spam classification that might use further study and enhancement, including multilingual categorization, image-based spam detection, and user input integration. The necessity for robustness against adversarial assaults and privacy issues also continue to be significant obstacles.

Overall, classifying spam emails is a challenging problem that is always changing and requiring a combination of machine learning methods, data pretreatment, and user input. We can improve consumers' email experiences while preserving their security and privacy by creating and implementing efficient email spam classification systems.

REFERENCES

1. What Is A Multi-Signature Wallet? “What Is A Multi-Signature Wallet? - LCX". 2022. LCX. https://www.lcx.com/what-is-a-multi-signature-wallet/

2. Multi-Sig Wallets Explained "Multi-Sig Wallets Explained" 2019. Medium. https://medium.com/block-journal/multi-sig-wallets-explained-5544c122a1de.

3. Moganty, D. (2019) Securing Ethereum Wallet With Multisig, Innominds.com. Available_at:

https://www.innominds.com/blog/securing-ethereum-wallet-with-multisig /

4. What Is a Multisig Wallet & How Does It Work? What Is a Multisig Wallet & How Does It Work? (2022). Available at: https://learn.bybit.com/blockchain/what-is-multisig-wallet/

5. Mukundan, Devi, Dr. R. Sasikumar, Badi Alekhya, and Arun S. “A Survey on Consensus Algorithms Used in Blockchain Platforms.” .

6. https://www.warse.org/IJATCSE/static/pdf/file/ijatcse323915

7. Gnosis Safe: Smart contract-based multisig wallet "Gnosis Safe: Smart Contract-Based Multisig Wallet". 2019. Defiprime.Com.

https://defiprime.com/gnosis-safe.

8. Ethereum IDE & Community. Remix. (n.d.). Retrieved November 25, 2022,from https://remix-project.org/

9. Polygon's Mumbai Testnet: A complete guide. Polygon's Mumbai Testnet: AComplete Guide. (n.d.). from https://www.alchemy.com/overviews/mumbai-testnet

8. APIs for a vibrant decentralized future. The Graph. (n.d.). fromhttps://thegraph.com/en/

9. Documentation: Ethereum development environment for professionals byNomic Foundation. Hardhat. (n.d.). from https://hardhat.org/docs

10. Hay, T. (2020, July 2). Ethereum JavaScript libraries: Web3.js vs. Ethers.js(part I). Infura Blog | Tutorials, Case Studies, News, Feature Announcements.from https://blog.infura.io/post/ethereum-javascript-libraries-web3-js-vs-ethers-js-part-i/