In [1]:

```
# create sms spam message detector using naive buyers ML algorithm
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
# reading and loading csv file
df = pd.read_csv("SMSCollectione.csv")
```

In [4]:

```
#shape of the data set
df.shape
```

Out[4]:

```
(5572, 2)
```

In [5]:

```
#data cleaning
```

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Class   5572 non-null   object
 1   sms     5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

In [7]:

```
df.sample
```

Out[7]:

```
<bound method NDFrame.sample of      Class
sms
0       ham  Go until jurong point, crazy.. Available only ...
1       ham                      Ok lar... Joking wif u oni...
2      spam  Free entry in 2 a wkly comp to win FA Cup fina...
3       ham  U dun say so early hor... U c already then say...
4       ham  Nah I don't think he goes to usf, he lives aro...
...     ...                                                ...
5567   spam  This is the 2nd time we have tried 2 contact u...
5568    ham              Will ü b going to esplanade fr home?
5569    ham  Pity, * was in mood for that. So...any other s...
5570    ham  The guy did some bitching but I acted like i'd...
5571    ham                        Rofl. Its true to its name

[5572 rows x 2 columns]>
```

In [8]:

```
# Rename the data columns name class and sms
df.rename(columns={'class':'target','v2':'text'},inplace=True)
df.sample(10)
```

Out[8]:

	Class	sms
4424	ham	alright. Thanks for the advice. Enjoy your nig...
982	ham	Reckon need to be in town by eightish to walk ...
1303	ham	FRAN I DECIDED 2 GO N E WAY IM COMPLETELY BROK...
3606	ham	Jordan got voted out last nite!
4075	ham	A lot of this sickness thing going round. Take...
3854	ham	Dont worry, 1 day very big lambu ji vl come..t...
4702	ham	I liked the new mobile
3800	ham	Actually nvm, got hella cash, we still on for ...
1488	ham	I told your number to gautham..
4033	ham	I'm very happy for you babe ! Woo hoo party on...

In [9]:

```
#to diffrenciate ham and spame in the class using lable encoder in 0 and 1
```

In [10]:

```
from sklearn.preprocessing import LabelEncoder
```

In [11]:

```
encoder= LabelEncoder()
```

In [12]:

```
df['Class']=encoder.fit_transform(df['Class'])
```

In [13]:

```
df.head()
```

Out[13]:

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

In [14]:

```
#missing values
```

In [15]:

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

Out[15]:

```
Class    0
sms      0
dtype: int64
```

In [16]:

```
#cheak duplicate values
```

In [17]:

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

Out[17]:

```
403
```

In [18]:

```
# drop duplicatealues
```

In [19]:

```
df=df.drop_duplicates(keep="first")
```

In [20]:

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

Out[20]:

```
0
```

In [21]:

```
df.shape
```

Out[21]:

```
(5169, 2)
```

In [22]:

```
# EDA(explolatry data analysis) data analysis
```

In [23]:

```
df.head()
```

Out[23]:

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

In [24]:

```
df["Class"].value_counts()
```

Out[24]:

```
0    4516
1     653
Name: Class, dtype: int64
```

In [ ]:

```
```

In [25]:

```
import seaborn as sns
```

In [26]:

```
sns.scatterplot(df['Class'],df['sms'])
```

```
C:\Users\HP\anaconda4\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[26]:

```
<AxesSubplot:xlabel='Class', ylabel='sms'>
```

```
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:240: RuntimeWarning: Glyph 146 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:240: RuntimeWarning: Glyph 148 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:240: RuntimeWarning: Glyph 145 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:240: RuntimeWarning: Glyph 147 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:240: RuntimeWarning: Glyph 150 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:240: RuntimeWarning: Glyph 9 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:240: RuntimeWarning: Glyph 37413 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:240: RuntimeWarning: Glyph 12328 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:203: RuntimeWarning: Glyph 146 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:203: RuntimeWarning: Glyph 148 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:203: RuntimeWarning: Glyph 145 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:203: RuntimeWarning: Glyph 147 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:203: RuntimeWarning: Glyph 150 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:203: RuntimeWarning: Glyph 9 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
y:203: RuntimeWarning: Glyph 37413 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\HP\anaconda4\lib\site-packages\matplotlib\backends\backend_agg.p
```

```
y:203: RuntimeWarning: Glyph 12328 missing from current font.
```

In [ ]:

```
#data is imbalanced
```

In [28]:

```
import nltk
```

In [29]:

```
nltk.download("punkt")
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[29]:

```
True
```

In [30]:

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

```
C:\Users\HP\AppData\Local\Temp/ipykernel_11388/4217142489.py:1: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  df['num_characters']=df['sms'].apply(len)
```

In [31]:

```
df.head()
```

Out[31]:

	Class	sms	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

In [32]:

```
# to convert sms it into words using nltk and tokenizing the number
```

In [33]:

```
#num words
df['num_words']=df['sms'].apply(lambda x:len(nltk.word_tokenize(x)))
```

```
C:\Users\HP\AppData\Local\Temp/ipykernel_11388/777086027.py:2: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  df['num_words']=df['sms'].apply(lambda x:len(nltk.word_tokenize(x)))
```

In [34]:

```
df.head()
```

Out[34]:

	Class	sms	num_characters	num_words
0	0	Go until jurong point, crazy.. Available only ...	111	24
1	0	Ok lar... Joking wif u oni...	29	8
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37
3	0	U dun say so early hor... U c already then say...	49	13
4	0	Nah I don't think he goes to usf, he lives aro...	61	15

In [35]:

```
# convert sms messages to words
```

In [36]:

```
df['sms'].apply(lambda x:nltk.sent_tokenize(x))
```

Out[36]:

```
0       [Go until jurong point, crazy.., Available onl...
1                         [Ok lar..., Joking wif u oni...]
2       [Free entry in 2 a wkly comp to win FA Cup fin...
3       [U dun say so early hor... U c already then sa...
4       [Nah I don't think he goes to usf, he lives ar...
                              ...
5567    [This is the 2nd time we have tried 2 contact ...
5568                [Will ü b going to esplanade fr home?]
5569    [Pity, * was in mood for that., So...any other...
5570    [The guy did some bitching but I acted like i'...
5571                          [Rofl., Its true to its name]
Name: sms, Length: 5169, dtype: object
```

In [37]:

```
df["sentence"]=df['sms'].apply(lambda x:len(nltk.sent_tokenize(x)))
```

```
C:\Users\HP\AppData\Local\Temp/ipykernel_11388/771654530.py:1: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  df["sentence"]=df['sms'].apply(lambda x:len(nltk.sent_tokenize(x)))
```

In [38]:

```
df.head()
```

Out[38]:

	Class	sms	num_characters	num_words	sentence
0	0	Go until jurong point, crazy.. Available only ...	111	24	2
1	0	Ok lar... Joking wif u oni...	29	8	2
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2
3	0	U dun say so early hor... U c already then say...	49	13	1
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1

In [39]:

```
df[['num_characters','num_words','sentence']].describe()
```

Out[39]:

	num_characters	num_words	sentence
count	5169.000000	5169.000000	5169.000000
mean	79.344554	18.593151	1.969627
std	58.437457	13.400486	1.443078
min	2.000000	1.000000	1.000000
25%	36.000000	9.000000	1.000000
50%	61.000000	15.000000	1.000000
75%	119.000000	26.000000	2.000000
max	910.000000	220.000000	38.000000

In [40]:

```
#spam
df[df['Class']==1][['num_characters','num_words','sentence']].describe()
```

Out[40]:

	num_characters	num_words	sentence
count	653.000000	653.000000	653.000000
mean	137.704441	27.762634	2.984686
std	29.821348	6.993008	1.495313
min	13.000000	2.000000	1.000000
25%	132.000000	25.000000	2.000000
50%	148.000000	29.000000	3.000000
75%	157.000000	32.000000	4.000000
max	223.000000	46.000000	9.000000

In [41]:

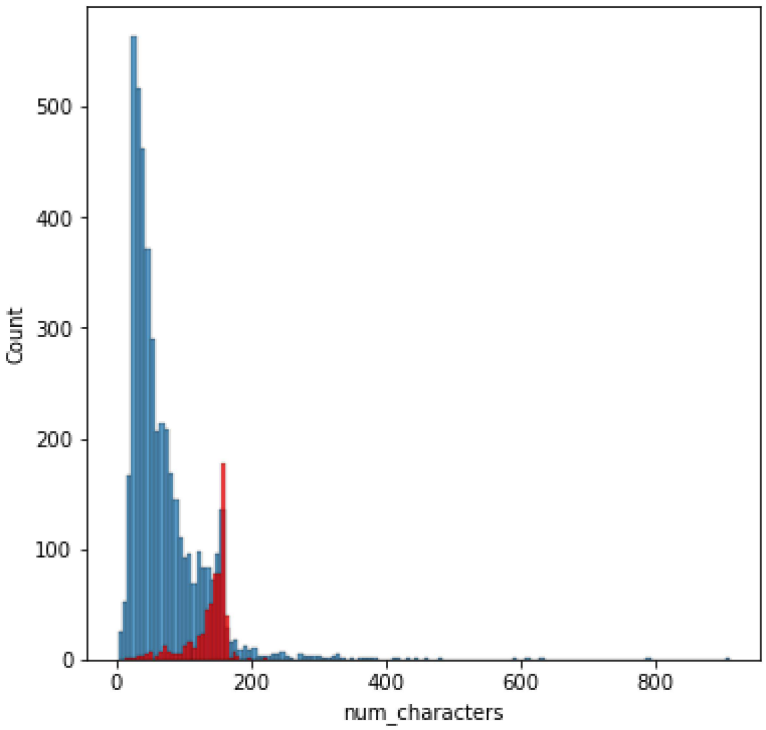
```
import seaborn as sns
```

In [42]:

```
plt.figure(figsize=(6,6))
sns.histplot(df[df['Class']==0]['num_characters'])
sns.histplot(df[df['Class']==1]['num_characters'],color='red')
```

Out[42]:

```
<AxesSubplot:xlabel='num_characters', ylabel='Count'>
```

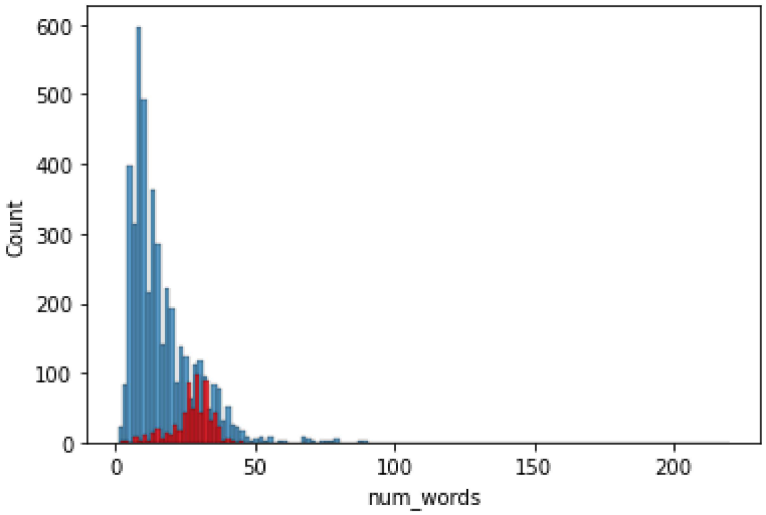


In [43]:

```
sns.histplot(df[df['Class']==0]['num_words'])
sns.histplot(df[df['Class']==1]['num_words'],color='red')
```

Out[43]:

```
<AxesSubplot:xlabel='num_words', ylabel='Count'>
```

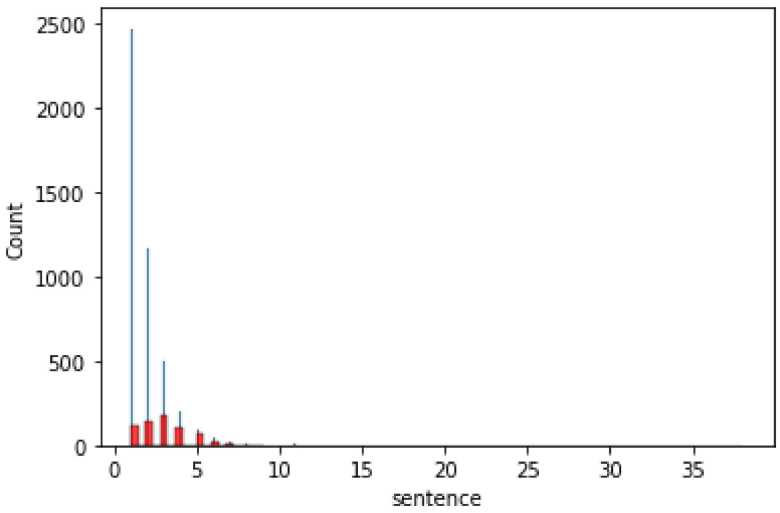


In [44]:

```
sns.histplot(df[df['Class']==0]['sentence'])
sns.histplot(df[df['Class']==1]['sentence'],color='red')
```

Out[44]:

```
<AxesSubplot:xlabel='sentence', ylabel='Count'>
```

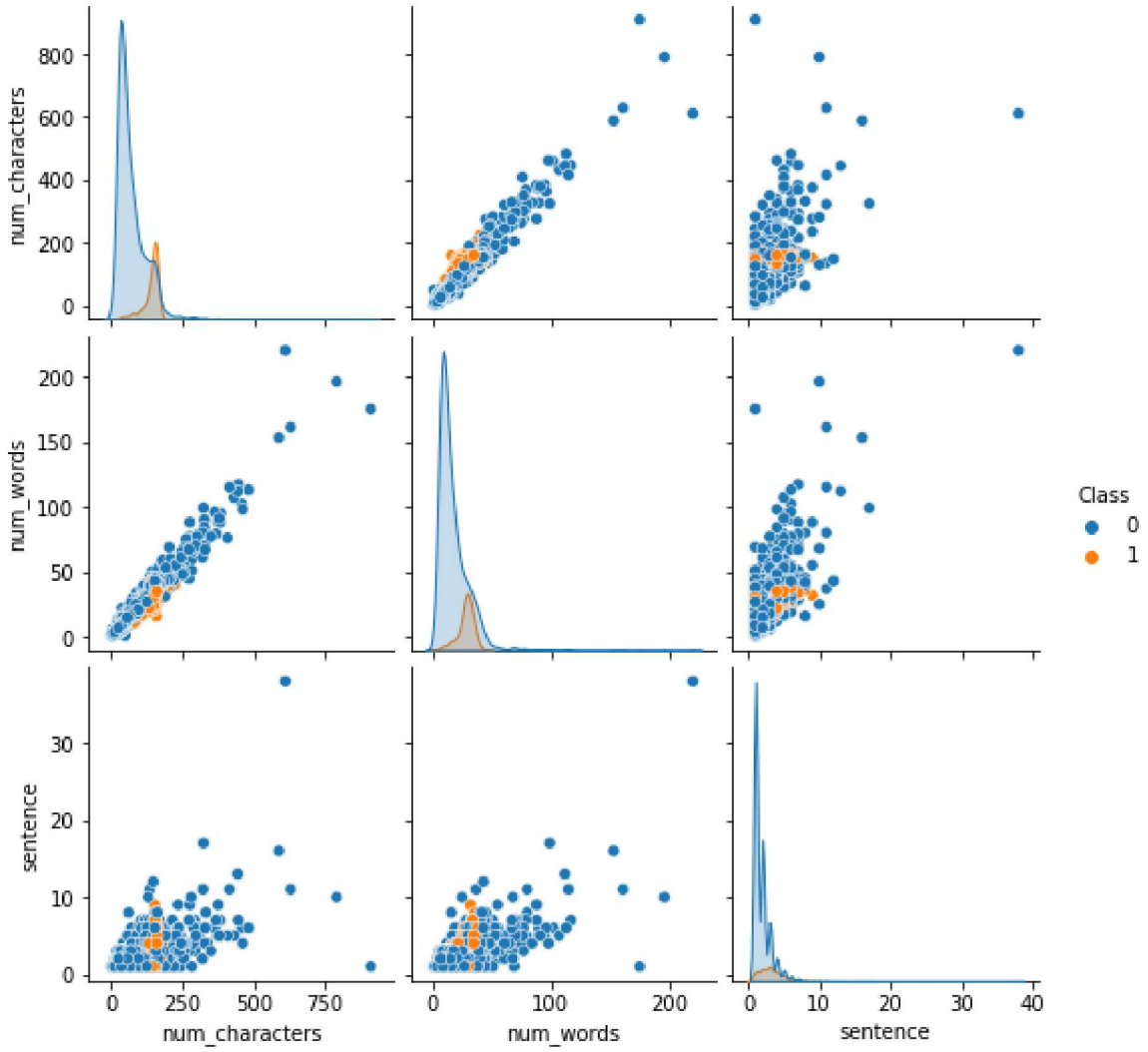


In [45]:

```
# analysing the pairplot and also then
sns.pairplot(df,hue='Class')
```

Out[45]:

```
<seaborn.axisgrid.PairGrid at 0x29b5518bd00>
```



In [46]:

```
df.corr()
```

Out[46]:

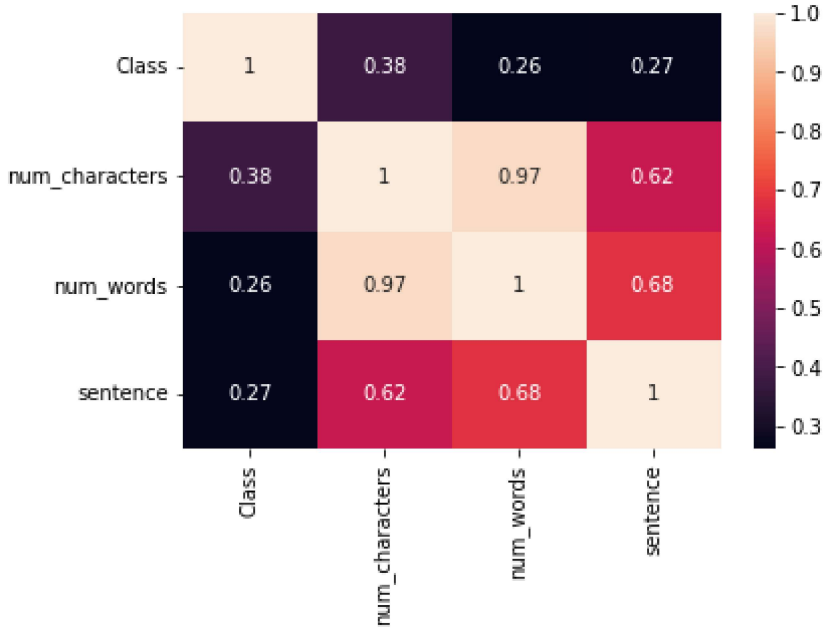
	Class	num_characters	num_words	sentence
Class	1.000000	0.379791	0.260223	0.267500
num_characters	0.379791	1.000000	0.966027	0.624405
num_words	0.260223	0.966027	1.000000	0.679939
sentence	0.267500	0.624405	0.679939	1.000000

In [47]:

```
sns.heatmap(df.corr(),annot=True)
```

Out[47]:

<AxesSubplot:>



In [48]:

```
#data preprocessing
#lower case
#Tokenization
#remove special characters y
#remove stop words and puncutation
#stemming
```

In [49]:

```
import string
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
from nltk.stem.porter import PorterStemmer
ps=  PorterStemmer()

def transform_text(text):
    text=text.lower()
    text = nltk.word_tokenize(text)

    y=[]
    for i in text:
        if i.isalnum():
            y.append(i)

    text = y[:]
    y.clear()

    for i in text :
        if i not in stopwords.words('english')  and i not in string.punctuation:
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        y.append(ps.stem(i))

    return " ".join(y)
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [50]:

```
transform_text('I loved the youtube videos on machine learning')
```

Out[50]:

```
'love youtub video machin learn'
```

In [51]:

```
transform_text(df['sms'][10])
```

Out[51]:

```
'gon na home soon want talk stuff anymor tonight k cri enough today'
```

In [52]:

```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[52]:

```
True
```

In [53]:

```
x=stopwords.words('english')
```

In [58]:

```
df['transformed_text']=df['sms'].apply(transform_text)
```

```
C:\Users\HP\AppData\Local\Temp/ipykernel_11388/3094600450.py:1: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  df['transformed_text']=df['sms'].apply(transform_text)
```

In [55]:

```
df.head()
```

Out[55]:

	Class	sms	num_characters	num_words	sentence	transformed_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	49	13	1	u dun say earli hor u c alreadi say
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1	nah think goe usf live around though

In [56]:

```
# stemming
```

In [57]:

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

In [59]:

```
# wordcloud
pip install wordcloud
```

```
  File "C:\Users\HP\AppData\Local\Temp/ipykernel_14712/1709129939.py", line
2
    pip install wordcloud
        ^
SyntaxError: invalid syntax
```

In [59]:

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

In [60]:

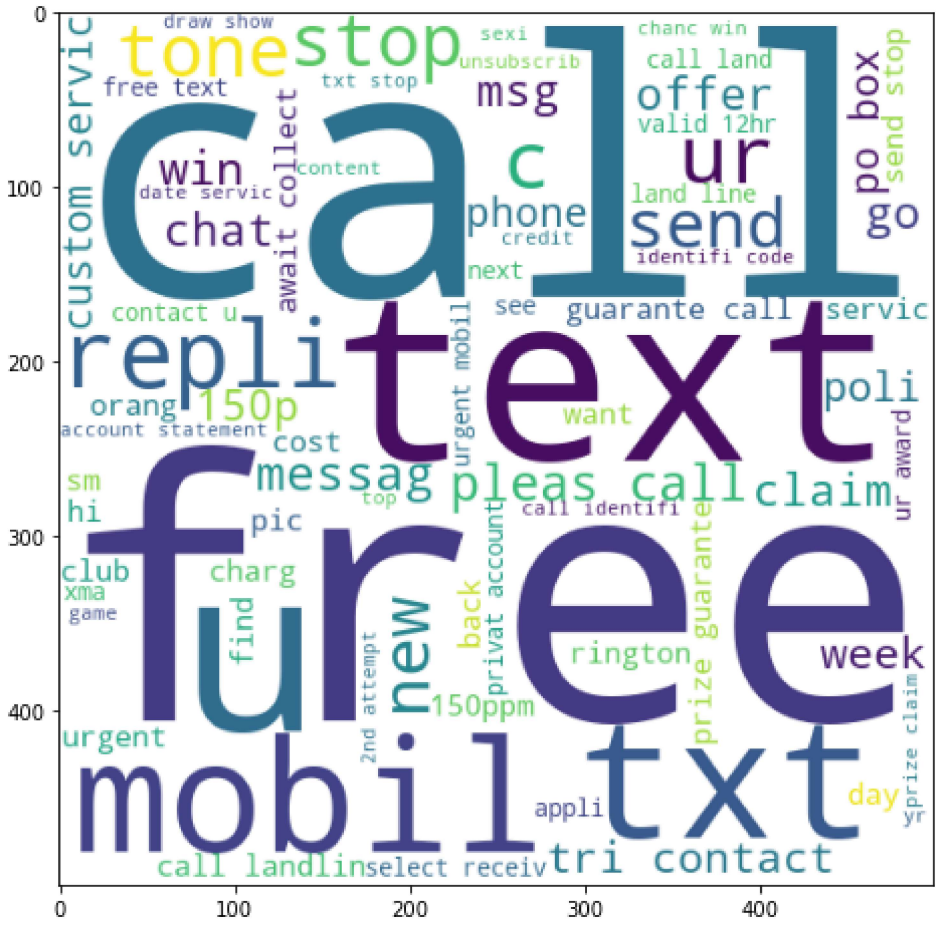
```
spam_wc=wc.generate(df[df['Class']==1]['transformed_text'].str.cat(sep=' '))
```

In [61]:

```
# words which most in spam message
spam_wc=wc.generate(df[df['Class']==1]['transformed_text'].str.cat(sep=' '))
plt.figure(figsize=(12,8))
plt.imshow(spam_wc)
```

Out[61]:

```
<matplotlib.image.AxesImage at 0x29b5746e7c0>
```

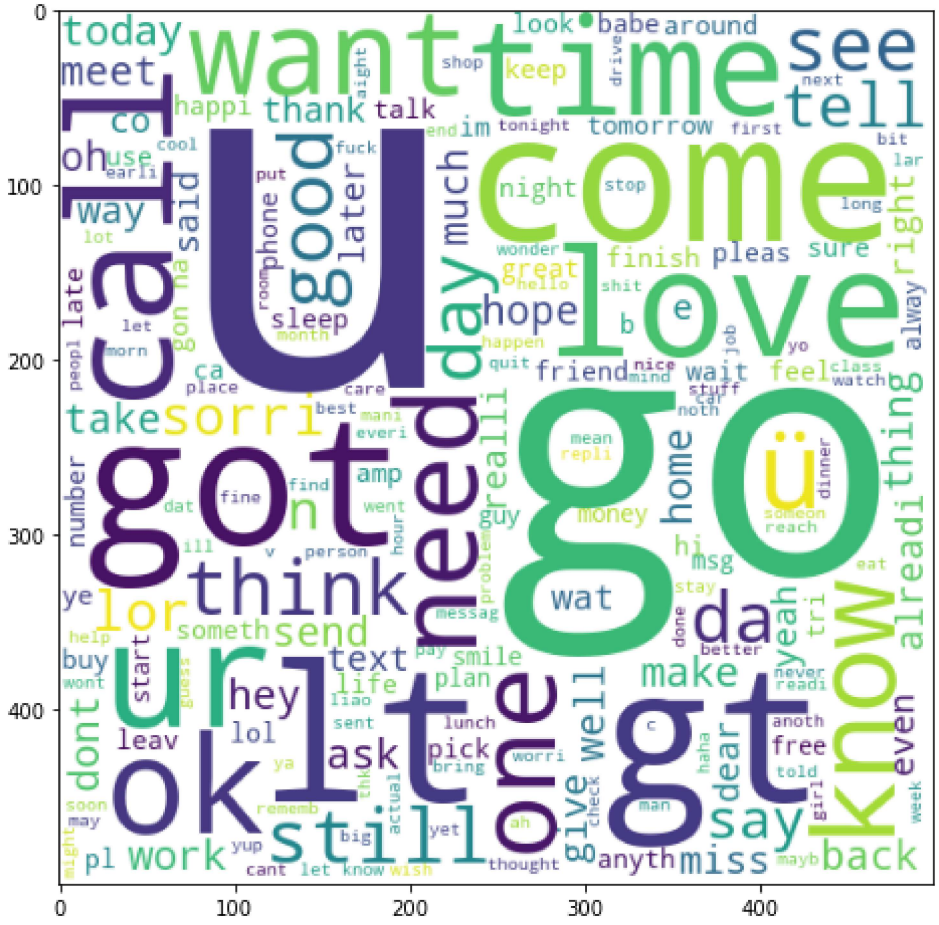


In [62]:

```
ham_wc=wc.generate(df[df['Class']==0]['transformed_text'].str.cat(sep=' '))
plt.figure(figsize=(12,8))
plt.imshow(ham_wc)
```

Out[62]:

```
<matplotlib.image.AxesImage at 0x29b58d32430>
```



In [63]:

```
df.head()
```

Out[63]:

	Class	sms	num_characters	num_words	sentence	transformed_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	49	13	1	u dun say earli hor u c alreadi say
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1	nah think goe usf live around though

In [64]:

```
# top words and top 30 words in spam message
```

In [65]:

```
# now we have to convert words to a list to wordlist

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

In [66]:

```
spam_corpus
```

```
 'camera',
 'free',
 'call',
 'mobil',
 'updat',
 'co',
 'free',
 '08002986030',
 'six',
 'chanc',
 'win',
 'cash',
 '100',
 'pound',
 'txt',
 'csh11',
 'send',
 'cost',
 '6day',
 'tsandc',
```

In [67]:

```
len(spam_corpus)
```

Out[67]:

```
9982
```

In [68]:

```
# to create a dictionary which will tell about the words in the dictionary
from collections import Counter
Counter(spam_corpus).most_common(30)
```

Out[68]:

```
[('call', 320),
 ('free', 191),
 ('2', 155),
 ('txt', 141),
 ('text', 122),
 ('u', 120),
 ('ur', 119),
 ('mobil', 114),
 ('stop', 108),
 ('repli', 103),
 ('claim', 98),
 ('4', 97),
 ('prize', 82),
 ('get', 74),
 ('new', 64),
 ('servic', 64),
 ('tone', 63),
 ('send', 61),
 ('urgent', 58),
 ('nokia', 57),
 ('contact', 56),
 ('award', 55),
 ('phone', 52),
 ('cash', 51),
 ('pleas', 51),
 ('week', 49),
 ('win', 48),
 ('c', 45),
 ('collect', 45),
 ('min', 45)]
```

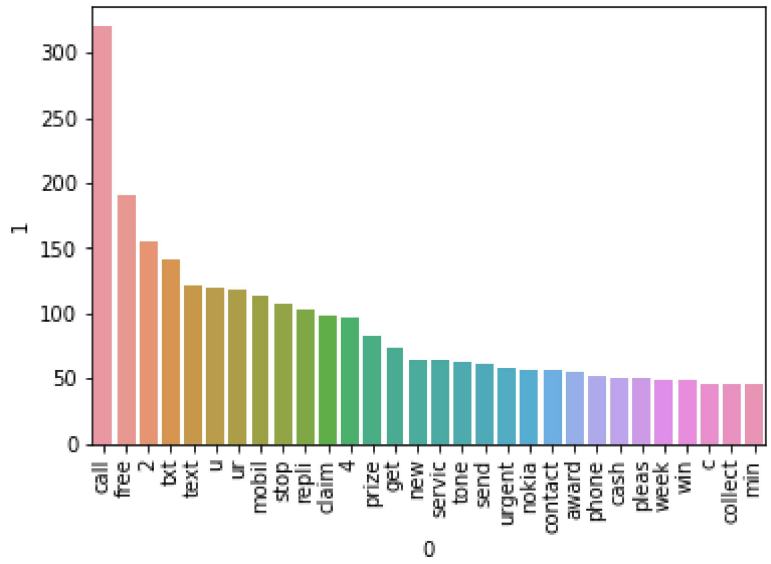
In [69]:

```
from collections import Counter
sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0],pd.DataFrame(Counter(spam
plt.xticks(rotation="vertical")
plt.show
```

```
C:\Users\HP\anaconda4\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[69]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [70]:

```
# now we have to convert words to a list to wordlist

ham_corpus=[]
for msg in df[df['Class']==0]['transformed_text'].tolist():
    for  word in msg.split():
        ham_corpus.append(word)
```

In [71]:

```
len(ham_corpus)
```

Out[71]:

```
35937
```

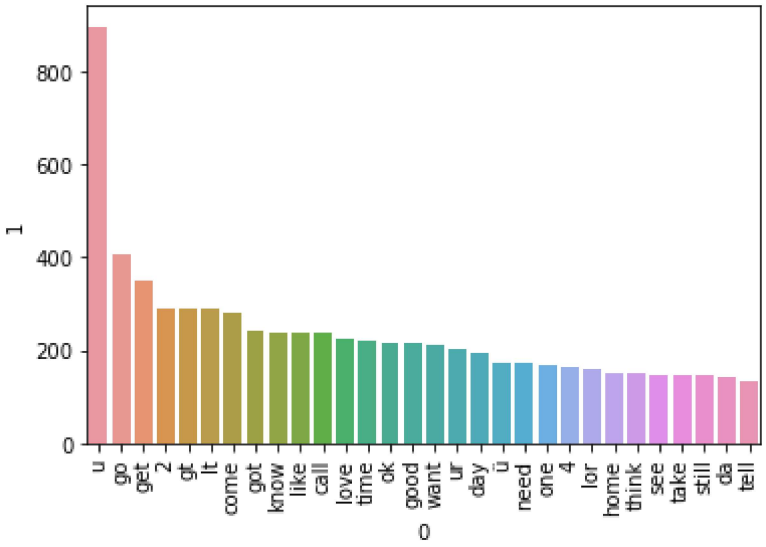
In [72]:

```
from collections import Counter
sns.barplot(pd.DataFrame(Counter(ham_corpus).most_common(30))[0],pd.DataFrame(Counter(ham_c
plt.xticks(rotation="vertical")
plt.show
```

```
C:\Users\HP\anaconda4\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variables as keyword args: x, y. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[72]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [73]:

```
# model building
```

In [74]:

```
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
cv = CountVectorizer()
tfidf = TfidfVectorizer(max_features=3000)
```

In [75]:

```
X = tfidf.fit_transform(df['transformed_text']).toarray()
```

In [76]:

```
X.shape
```

Out[76]:

```
(5169, 3000)
```

In [77]:

```
Y = df['Class'].values
```

In [78]:

```
Y
```

Out[78]:

```
array([0, 0, 1, ..., 0, 0, 0])
```

In [79]:

```
from sklearn.model_selection import train_test_split
```

In [80]:

```
 X_train,X_test,Y_train,Y_test = train_test_split(X,Y, test_size=0.2,random_state=2 )
```

In [81]:

```


from sklearn.naive_bayes import GaussianNB, MultinomialNB,BernoulliNB
from sklearn.metrics import accuracy_score, confusion_matrix,precision_score
```

In [82]:

```
gnb = GaussianNB()
mnb = MultinomialNB()
bnb = BernoulliNB()
```

In [83]:

```
gnb.fit(X_train,Y_train)
Y_pred1= gnb.predict(X_test)
print(accuracy_score(Y_test,Y_pred1))
print(confusion_matrix(Y_test,Y_pred1))
print(precision_score(Y_test,Y_pred1))
```

```
0.8617021276595744
[[788 120]
 [ 23 103]]
0.4618834080717489
```

In [84]:

```
mnb.fit(X_train,Y_train)
Y_pred1= mnb.predict(X_test)
print(accuracy_score(Y_test,Y_pred1))
print(confusion_matrix(Y_test,Y_pred1))
print(precision_score(Y_test,Y_pred1))
```

```
0.9690522243713733
[[908   0]
 [ 32  94]]
1.0
```

In [85]:

```
bnb.fit(X_train,Y_train)
Y_pred1= bnb.predict(X_test)
print(accuracy_score(Y_test,Y_pred1))
print(confusion_matrix(Y_test,Y_pred1))
print(precision_score(Y_test,Y_pred1))
```

```
0.9748549323017408
[[906   2]
 [ 24 102]]
0.9807692307692307
```

In [86]:

```
# tfidf and mnb is the combination
```

In [ ]:

```
# voting classifier is adding/combination of all the best models
```

In [87]:

```
import pickle
pickle.dump(tfidf,open('vectorizer.pkl','wb'))
pickle.dump(mnb,open('model.pkl','wb'))
```