In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df=pd.read_csv('spam.csv',encoding='ISO-8859-1')
```

In [3]:
```
df.sample(5)
```

Out[3]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
3048	ham	Lol they don't know about my awesome phone. I ...	NaN	NaN	NaN
4806	spam	PRIVATE! Your 2004 Account Statement for 07849...	NaN	NaN	NaN
1848	ham	I dont want to hear philosophy. Just say what ...	NaN	NaN	NaN
609	ham	It's fine, imma get a drink or somethin. Want ...	NaN	NaN	NaN
1634	spam	Eerie Nokia tones 4u, rply TONE TITLE to 8007 ...	NaN	NaN	NaN

In [4]:
```
df.shape
#we got 5572 msg with 5 column with null values
```

Out[4]:
```
(5572, 5)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

Steps:

1. Data cleaning
2. EDA
3. text preprocessing
4. Model building
5. Evaluation
6. Improvement
7. website
8. deployment

1. Data cleaning

In [6]:
```
#drop last three column as they are containing null values
df.head()
```

Out[6]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

In [7]:
```
df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],axis=1,inplace=True)
```

In [8]:
```
df.head()
```

Out[8]:

	v1	v2
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...

In [9]:
```
#rename the column name to meaningful name
df.rename(columns={'v1':'target','v2':'text'},inplace=True)
```

In [10]:
```
df.head()
```

Out[10]:

	target	text
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...

In [11]:
```
#convert target variable to number - as machine is capable to understand values in num

# we are using label encoder here as we have only 2 category in target column(ham/spam
#into 0 and 1

from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
```

In [12]:
```
df['target']=encoder.fit_transform(df['target'])
```

In [13]:
```
df.head()
```

Out[13]:

	target	text
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]:
```
#Now finding the missing value in dataframe
df.isnull().sum()
```

Out[14]:
```
target    0
text      0
dtype: int64
```

In [15]:
```
#Find the duplicate value in dataframe
df.duplicated().sum()
```

Out[15]:
```
403
```

In [16]:
```
#remove duplicates
df.drop_duplicates(inplace=True)
```

In [17]:
```
df.duplicated().sum()
```

Out[17]:
```
0
```

In [18]:
```
df.shape
```

Out[18]:
```
(5169, 2)
```

2. EDA

In [19]:
```
df.head()
```

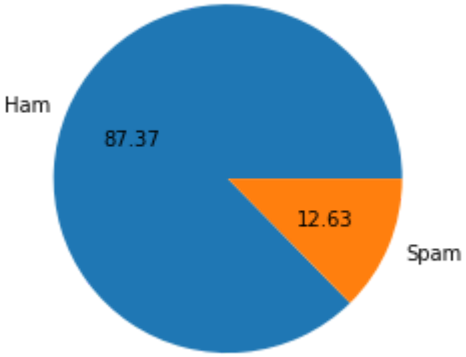
Out[19]:

	target	text
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 [20]:
```
#lets find the distribution of target values
df['target'].value_counts()
```

Out[20]:
```
0    4516
1     653
Name: target, dtype: int64
```

In [21]:
```
#lets visualize it in pie chart
import matplotlib.pyplot as plt
plt.pie(df['target'].value_counts(),labels=['Ham','Spam'],autopct='%0.2f')
plt.show()

# so our data is imbalanced
```



In [22]:
```
#NLTK is a leading platform for building Python programs to work with human language d
#To install NLTK use !pip install nltk in cmd
import nltk
```

In [23]:
```
nltk.download('punkt')
#This tokenizer divides a text into a list of sentences by using an unsupervised algor
#for abbreviation words, collocations, and words that start sentences.
```

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

Out[23]:
```
True
```

In [24]:
```
df.head(1)
```

Out[24]:

	target	text
0	0	Go until jurong point, crazy.. Available only ...

In [25]:
```
#count no of char in text data for each rows
(df['text']).apply(len)
```

Out[25]:
```
0       111
1        29
2       155
3        49
4        61
       ...
5567    161
5568     37
5569     57
5570    125
5571     26
Name: text, Length: 5169, dtype: int64
```

In [26]:
```
#Let add no of char in main dataframe
df['num_characters']=df['text'].apply(len)
```

In [27]:
```
df.head()
```

Out[27]:

	target	text	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 [28]:
```
#Find no of words in dataframe using NLTK - word_tokenize and add in dataframe
df['num_words']=df['text'].apply(lambda x:len(nltk.word_tokenize(x)))
```

In [29]:
```
df.head()
```

Out[29]:

	target	text	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 [30]:
```
#Find no of sentences in dataframe using NLTK - sent_tokenize and add in dataframe
df['num_sentences']=df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))
```

In [31]:
```
df.head()
```

Out[31]:

	target	text	num_characters	num_words	num_sentences
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 [32]:
```
df[['num_characters','num_words','num_sentences']].describe()
```

Out[32]:

	num_characters	num_words	num_sentences
count	5169.000000	5169.000000	5169.000000
mean	78.977945	18.453279	1.947185
std	58.236293	13.324793	1.362406
min	2.000000	1.000000	1.000000
25%	36.000000	9.000000	1.000000
50%	60.000000	15.000000	1.000000
75%	117.000000	26.000000	2.000000
max	910.000000	220.000000	28.000000

In [33]:
```
#Find the descriptive statistical summary when sms is ham

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

Out[33]:

	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

In [34]:
```
#Find the descriptive statistical summary when sms is spam

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

Out[34]:

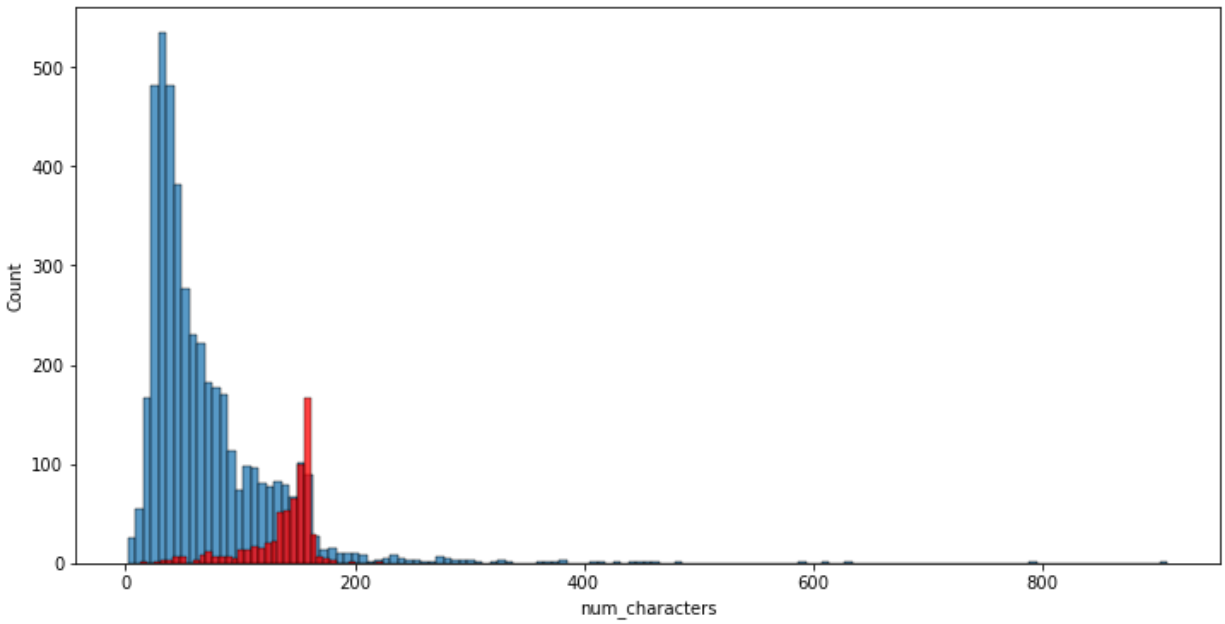
	num_characters	num_words	num_sentences
count	653.000000	653.000000	653.000000
mean	137.891271	27.667688	2.967841
std	30.137753	7.008418	1.483201
min	13.000000	2.000000	1.000000
25%	132.000000	25.000000	2.000000
50%	149.000000	29.000000	3.000000
75%	157.000000	32.000000	4.000000
max	224.000000	46.000000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

By checking above values, we can say that avg length of (character/words/sentences) are more in spam message as compared to Ham.

we can see outliers in ham messages

Important Visualization

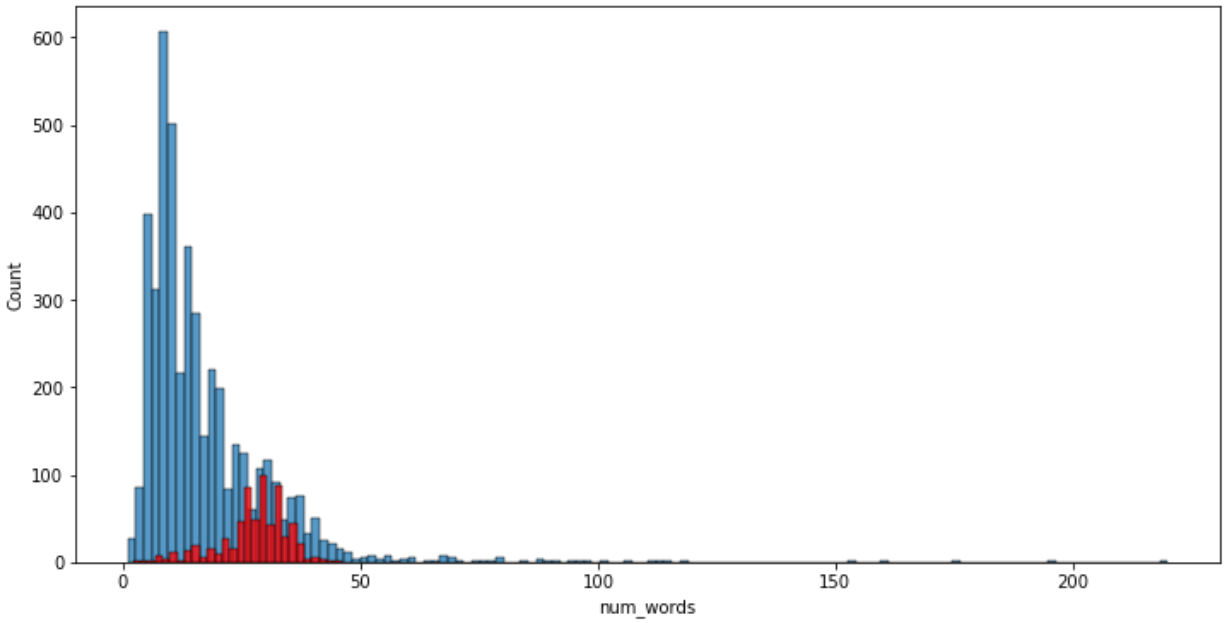
In [35]:
```
import seaborn as sns

# do visulization for num_chars for Ham and Spam

plt.figure(figsize=(12,6))
sns.histplot(df[df['target']==0]['num_characters'])
sns.histplot(df[df['target']==1]['num_characters'],color='red')
plt.show()
```



we can say num of char are more in spam message but they are less in number.

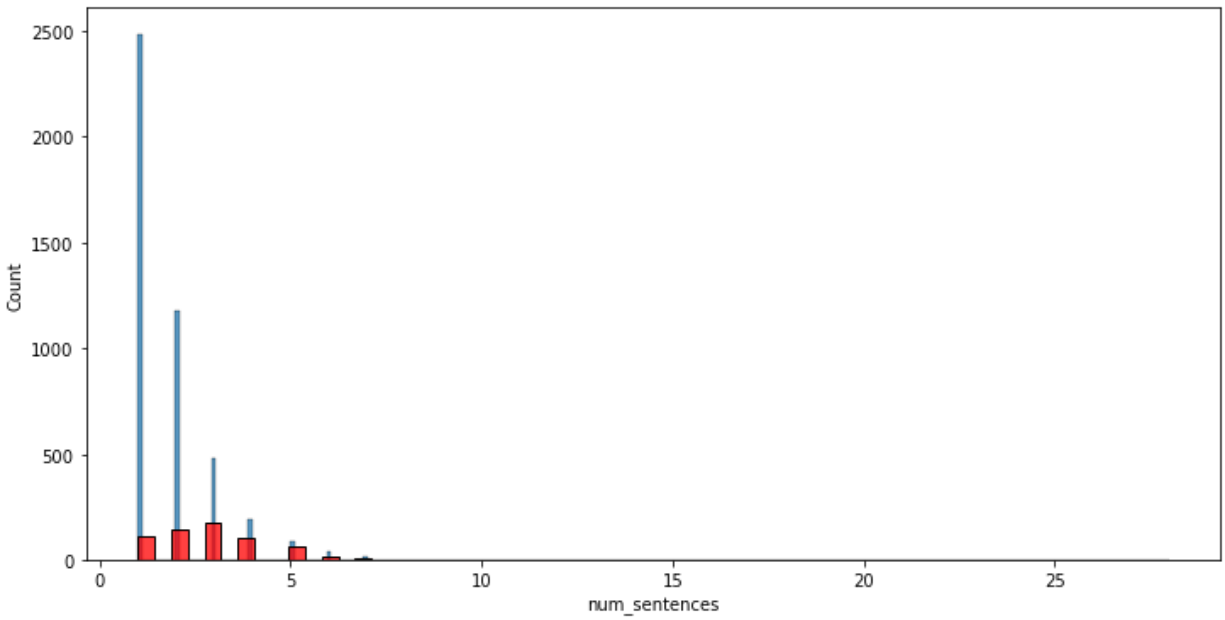
In [36]:
```
# do visulization for num_words for Ham and Spam
```

In [37]:
```
plt.figure(figsize=(12,6))

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



we can say num of words are more in spam message but they are less in number.

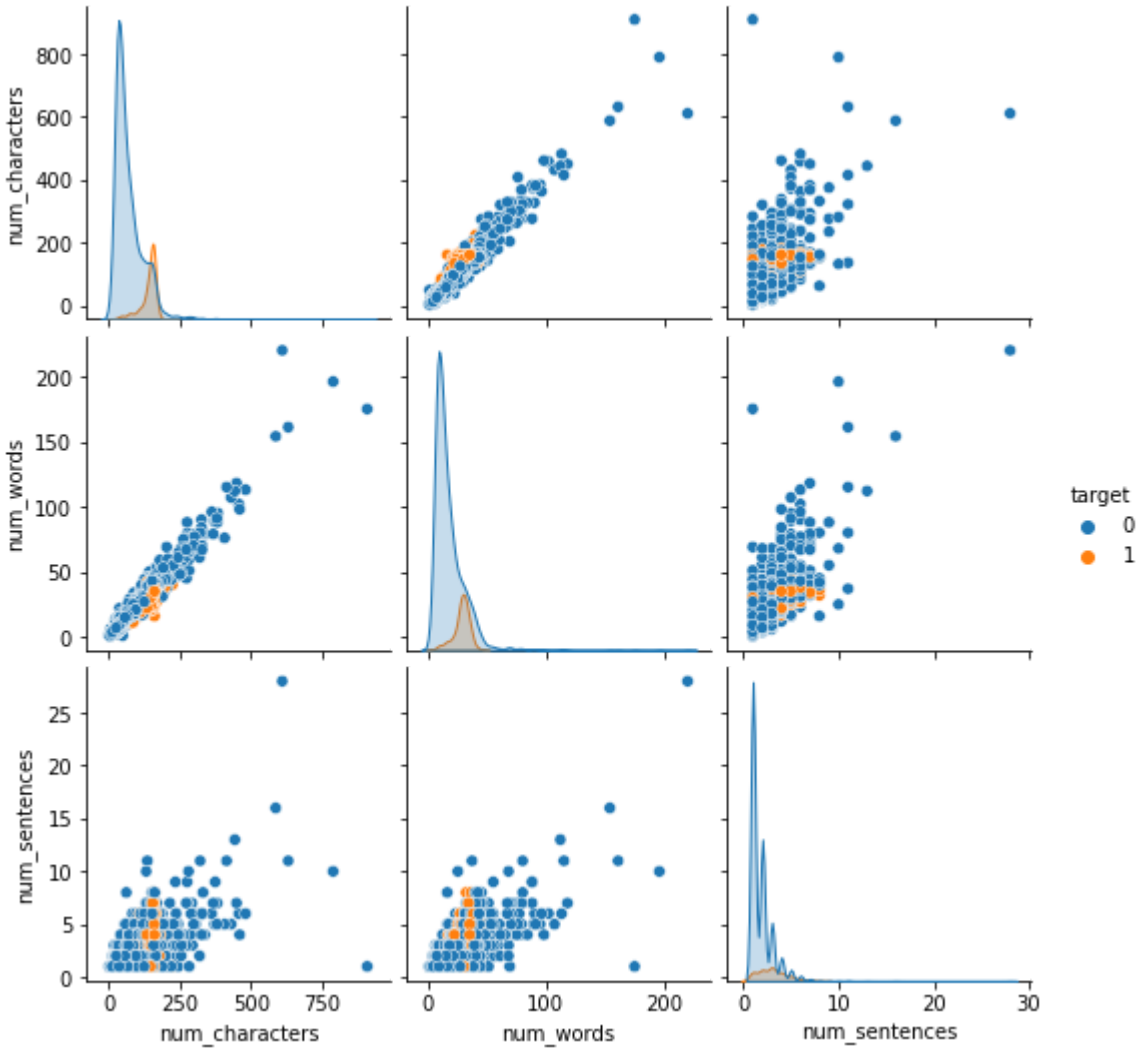
In [38]:
```
# do visulization for num_sentences for Ham and Spam
plt.figure(figsize=(12,6))

sns.histplot(df[df['target']==0]['num_sentences'],)
sns.histplot(df[df['target']==1]['num_sentences'],color='red')
plt.show()
```



we can say, for spam messages, count of sentences are more when number of sentences is 3

In [179…
```
#plot pair plot to see dependency b/w variables and target values

plt.figure(figsize=(10,8))
sns.pairplot(df,hue='target')
plt.show()
```

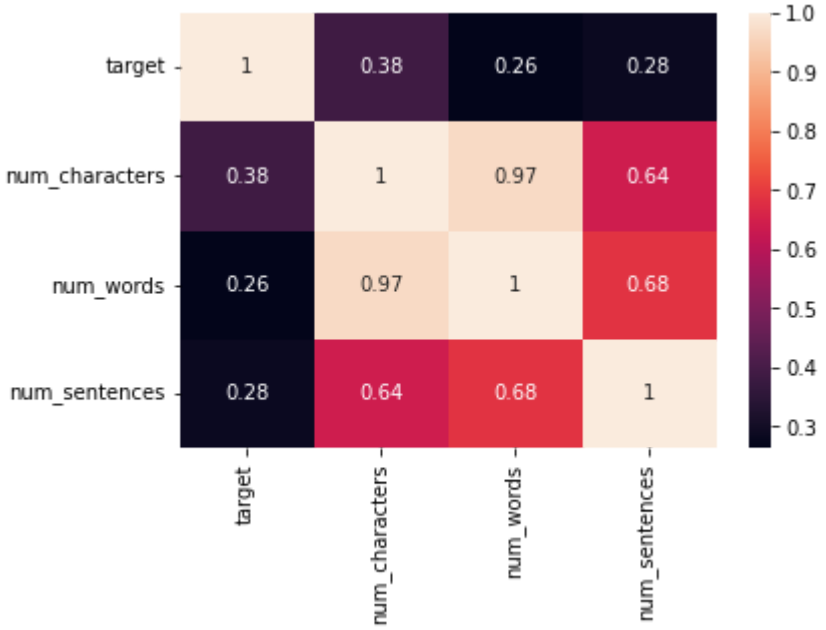
```
<Figure size 720x576 with 0 Axes>
```



number of words and number of charcters are showing linear relationship. It means more correlation among independent variables.

In [40]:
```
#lets further check the heatmap to correlation b/w all variables
sns.heatmap(df.corr(),annot=True)
```

Out[40]:
```
<AxesSubplot:>
```



num_words and num_characters are showing very high correlation num_char and num_words with num_sentences are also showing high corrleation we need to take care in case of model deployment

3. Data Preprocessing

Below are the steps we are going to perform in data preprocessing steps Lower case Tokenization Removing special characters Removing stop words and punctuation Stemming we will create function which will handle all above tasls at once

In [41]:
```
df.head()
```

Out[41]:

	target	text	num_characters	num_words	num_sentences
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 [42]:
```
import nltk
#way to include stopwords which will use in function
nltk.download('stopwords')
from nltk.corpus import stopwords
stopwords.words('ENGLISH')
```

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

Out[42]:
```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
 'his',
 'himself',
 'she',
 "she's",
 'her',
 'hers',
 'herself',
 'it',
 "it's",
 'its',
 'itself',
 'they',
 'them',
 'their',
 'theirs',
 'themselves',
 'what',
 'which',
 'who',
 'whom',
 'this',
 'that',
 "that'll",
 'these',
 'those',
 'am',
 'is',
 'are',
 'was',
 'were',
 'be',
 'been',
 'being',
 'have',
 'has',
 'had',
 'having',
 'do',
 'does',
 'did',
 'doing',
```

```
'a',
'an',
'the',
'and',
'but',
'if',
'or',
'because',
'as',
'until',
'while',
'of',
'at',
'by',
'for',
'with',
'about',
'against',
'between',
'into',
'through',
'during',
'before',
'after',
'above',
'below',
'to',
'from',
'up',
'down',
'in',
'out',
'on',
'off',
'over',
'under',
'again',
'further',
'then',
'once',
'here',
'there',
'when',
'where',
'why',
'how',
'all',
'any',
'both',
'each',
'few',
'more',
'most',
'other',
'some',
'such',
'no',
'nor',
'not',
'only',
```

```
'own',
'same',
'so',
'than',
'too',
'very',
's',
't',
'can',
'will',
'just',
'don',
"don't",
'should',
"should've",
'now',
'd',
'll',
'm',
'o',
're',
've',
'y',
'ain',
'aren',
"aren't",
'couldn',
"couldn't",
'didn',
"didn't",
'doesn',
"doesn't",
'hadn',
"hadn't",
'hasn',
"hasn't",
'haven',
"haven't",
'isn',
"isn't",
'ma',
'mightn',
"mightn't",
'mustn',
"mustn't",
'needn',
"needn't",
'shan',
"shan't",
'shouldn',
"shouldn't",
'wasn',
"wasn't",
'weren',
"weren't",
'won',
"won't",
'wouldn',
"wouldn't"]
```

In [ ]:
```
```

In [43]:
```
from nltk.stem.porter import PorterStemmer
#way to include stem which will use in function
ps=PorterStemmer()
ps.stem('standalsone')
```

Out[43]:
```
'standalson'
```

In [44]:
```
import string
#way to include punctuation which will use in function
string.punctuation
```

Out[44]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

Our Preprocessing Function

In [45]:
```
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)
```

In [180…
```
#testing of above funcion
transform_text('how are you and how you are doing, come at home for dancing, i loved y
```

Out[180]:
```
'come home danc love sinc past'
```

In [181…
```
transform_text('Did you like my words on ML presentation?')
```

Out[181]:
```
'like word ml present'
```

In [48]:
```
#apply our function to all our text messages and stored in dataframe
df['transformed_text']=df['text'].apply(transform_text)
```

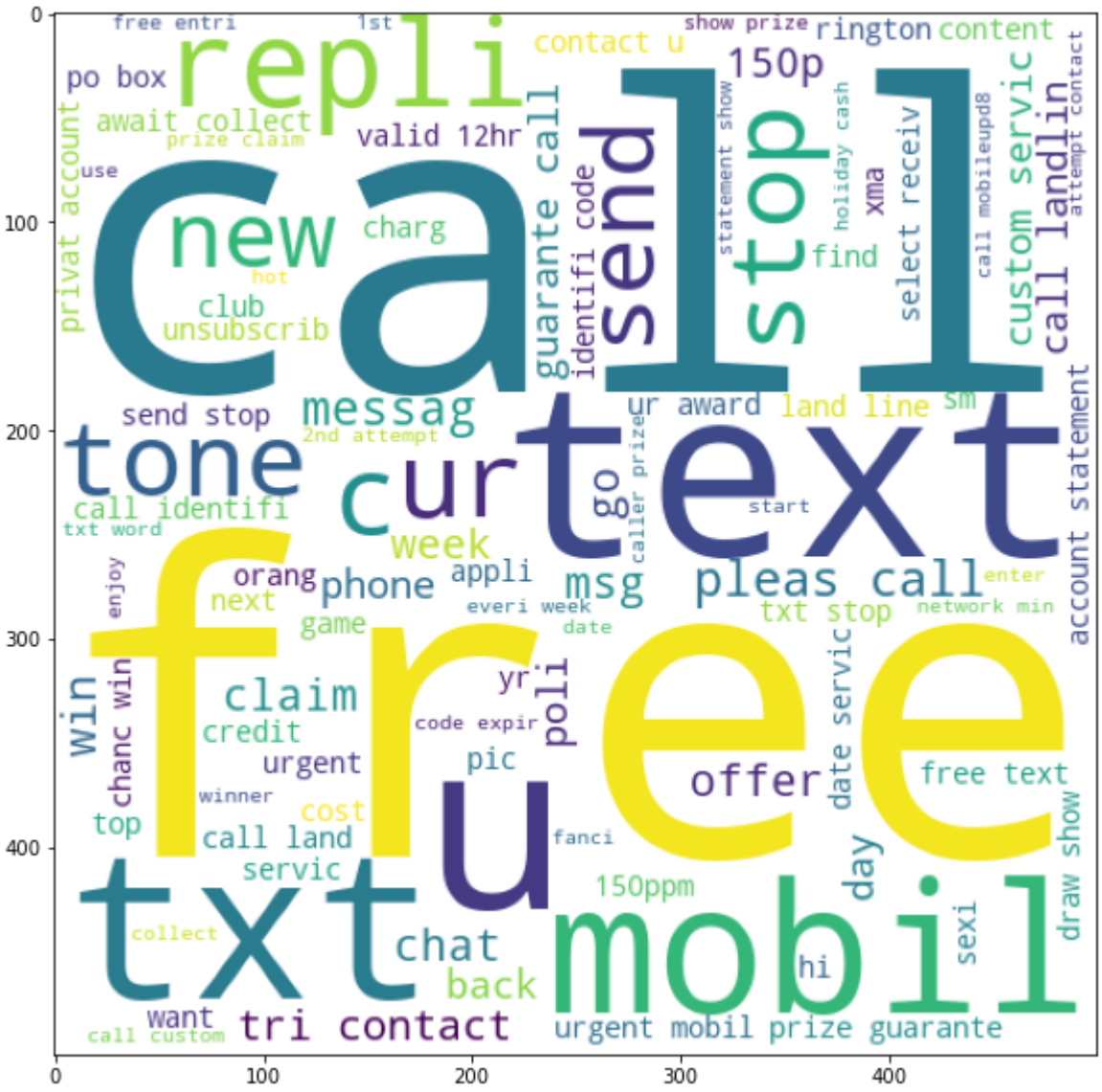
In [50]:
```
df.head()
```

Out[50]:

	target	text	num_characters	num_words	num_sentences	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 [51]:

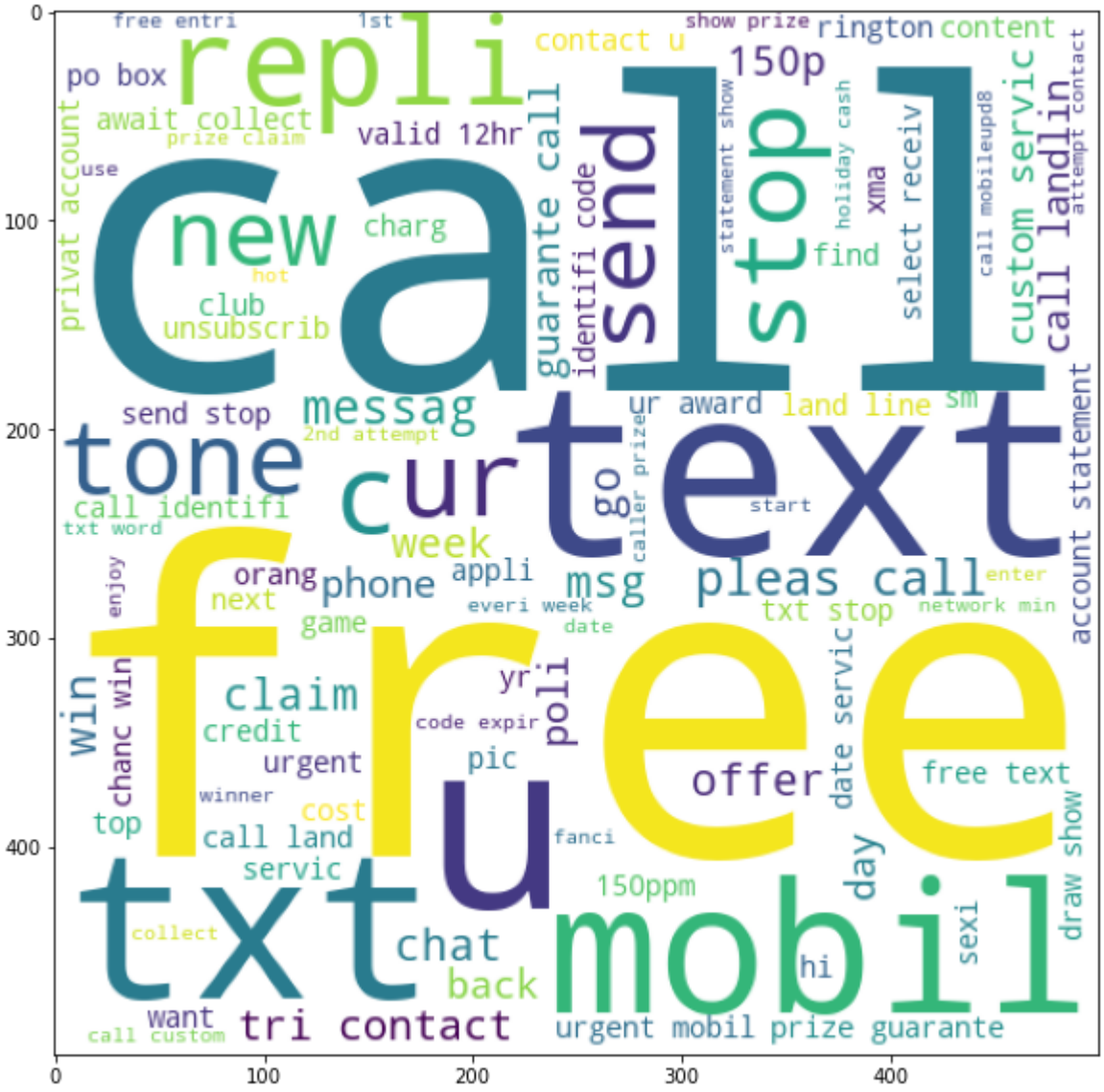
```
#!pip install wordcloud
# let see the wordcloud if messages are spam
from wordcloud import WordCloud
wc=WordCloud(height=500,width=500,min_font_size=10,background_color='white')

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


plt.figure(figsize=(10,10))
plt.imshow(spam_wc)
plt.show()
```



Mostly word are spam words

In [182…

```
from wordcloud import WordCloud
# let see the wordcloud if messages are Ham

wc=WordCloud(height=500,width=500,min_font_size=10,background_color='white')

ham_wc = wc.generate(df[df['target']==0]['transformed_text'].str.cat(sep=' '))


plt.figure(figsize=(10,10))
plt.imshow(spam_wc)
plt.show()
```



In [53]:
```
#find all words in spam messages
spam_corpus = []
for msg in df[df['target']==1]['transformed_text'].tolist():
    for word in msg.split():

spam_corpus
```

Out[53]:
```
['free',
 'entri',
 '2',
 'wkli',
 'comp',
 'win',
 'fa',
 'cup',
 'final',
 'tkt',
 '21st',
 'may',
 'text',
 'fa',
 '87121',
 'receiv',
 'entri',
 'question',
 'std',
 'txt',
 'rate',
 'c',
 'appli',
 '08452810075over18',
 'freemsg',
 'hey',
 'darl',
 '3',
 'week',
 'word',
 'back',
 'like',
 'fun',
 'still',
 'tb',
 'ok',
 'xxx',
 'std',
 'chg',
 'send',
 'rcv',
 'winner',
 'valu',
 'network',
 'custom',
 'select',
 'receivea',
 'prize',
 'reward',
 'claim',
 'call',
 'claim',
 'code',
 'kl341',
 'valid',
 '12',
 'hour',
 'mobil',
 '11',
 'month',
```

```
'u',
'r',
'entitl',
'updat',
'latest',
'colour',
'mobil',
'camera',
'free',
'call',
'mobil',
'updat',
'co',
'free',
'08002986030',
'six',
'chanc',
'win',
'cash',
'100',
'pound',
'txt',
'csh11',
'send',
'cost',
'6day',
'tsandc',
'appli',
'repli',
'hl',
'4',
'info',
'urgent',
'1',
'week',
'free',
'membership',
'prize',
'jackpot',
'txt',
'word',
'claim',
'81010',
'c',
'lccltd',
'pobox',
'4403ldnw1a7rw18',
'xxxmobilemovieclub',
'use',
'credit',
'click',
'wap',
'link',
'next',
'txt',
'messag',
'click',
'http',
'england',
'v',
```

```
'macedonia',
'dont',
'miss',
'news',
'txt',
'ur',
'nation',
'team',
'87077',
'eg',
'england',
'87077',
'tri',
'wale',
'scotland',
'poboxox36504w45wq',
'thank',
'subscript',
'rington',
'uk',
'mobil',
'charg',
'pleas',
'confirm',
'repli',
'ye',
'repli',
'charg',
'07732584351',
'rodger',
'burn',
'msg',
'tri',
'call',
'repli',
'sm',
'free',
'nokia',
'mobil',
'free',
'camcord',
'pleas',
'call',
'08000930705',
'deliveri',
'tomorrow',
'sm',
'ac',
'sptv',
'new',
'jersey',
'devil',
'detroit',
'red',
'wing',
'play',
'ice',
'hockey',
'correct',
'incorrect',
```

```
'end',
'repli',
'end',
'sptv',
'congrat',
'1',
'year',
'special',
'cinema',
'pass',
'2',
'call',
'09061209465',
'c',
'suprman',
'v',
'matrix3',
'starwars3',
'etc',
'4',
'free',
'150pm',
'dont',
'miss',
'valu',
'custom',
'pleas',
'advis',
'follow',
'recent',
'review',
'mob',
'award',
'bonu',
'prize',
'call',
'09066364589',
'urgent',
'ur',
'award',
'complimentari',
'trip',
'eurodisinc',
'trav',
'aco',
'entry41',
'claim',
'txt',
'di',
'87121',
'morefrmmob',
'shracomorsglsuplt',
'10',
'ls1',
'3aj',
'hear',
'new',
'come',
'ken',
'stuff',
```

```
'pleas',
'call',
'custom',
'servic',
'repres',
'0800',
'169',
'6031',
'guarante',
'cash',
'prize',
'free',
'rington',
'wait',
'collect',
'simpli',
'text',
'password',
'85069',
'verifi',
'get',
'usher',
'britney',
'fml',
'gent',
'tri',
'contact',
'last',
'weekend',
'draw',
'show',
'prize',
'guarante',
'call',
'claim',
'code',
'k52',
'valid',
'12hr',
'150ppm',
'winner',
'u',
'special',
'select',
'2',
'receiv',
'4',
'holiday',
'flight',
'inc',
'speak',
'live',
'oper',
'2',
'claim',
'privat',
'2004',
'account',
'statement',
'07742676969',
```

```
'show',
'786',
'unredeem',
'bonu',
'point',
'claim',
'call',
'08719180248',
'identifi',
'code',
'45239',
'expir',
'urgent',
'mobil',
'award',
'bonu',
'caller',
'prize',
'final',
'tri',
'contact',
'u',
'call',
'landlin',
'09064019788',
'box42wr29c',
'150ppm',
'today',
'voda',
'number',
'end',
'7548',
'select',
'receiv',
'350',
'award',
'match',
'pleas',
'call',
'08712300220',
'quot',
'claim',
'code',
'4041',
'standard',
'rate',
'app',
'sunshin',
'quiz',
'wkli',
'q',
'win',
'top',
'soni',
'dvd',
'player',
'u',
'know',
'countri',
'algarv',
```

```
'txt',
'ansr',
'sp',
'tyron',
'want',
'2',
'get',
'laid',
'tonight',
'want',
'real',
'dog',
'locat',
'sent',
'direct',
'2',
'ur',
'mob',
'join',
'uk',
'largest',
'dog',
'network',
'bt',
'txting',
'gravel',
'69888',
'nt',
'ec2a',
'150p',
'rcv',
'msg',
'chat',
'svc',
'free',
'hardcor',
'servic',
'text',
'go',
'69988',
'u',
'get',
'noth',
'u',
'must',
'age',
'verifi',
'yr',
'network',
'tri',
'freemsg',
'repli',
'text',
'randi',
'sexi',
'femal',
'live',
'local',
'luv',
'hear',
```

```
'netcollex',
'ltd',
'08700621170150p',
'per',
'msg',
'repli',
'stop',
'end',
'custom',
'servic',
'annonc',
'new',
'year',
'deliveri',
'wait',
'pleas',
'call',
'07046744435',
'arrang',
'deliveri',
'winner',
'u',
'special',
'select',
'2',
'receiv',
'cash',
'4',
'holiday',
'flight',
'inc',
'speak',
'live',
'oper',
'2',
'claim',
'0871277810810',
'stop',
'bootydeli',
'invit',
'friend',
'repli',
'see',
'stop',
'send',
'stop',
'frnd',
'62468',
'bangbab',
'ur',
'order',
'way',
'u',
'receiv',
'servic',
'msg',
'2',
'download',
'ur',
'content',
```

```
'u',
'goto',
'wap',
'bangb',
'tv',
'ur',
'mobil',
'menu',
'urgent',
'tri',
'contact',
'last',
'weekend',
'draw',
'show',
'prize',
'guarante',
'call',
'claim',
'code',
's89',
'valid',
'12hr',
'pleas',
'call',
'custom',
'servic',
'repres',
'freephon',
'0808',
'145',
'4742',
'guarante',
'cash',
'prize',
'uniqu',
'enough',
'find',
'30th',
'august',
'500',
'new',
'mobil',
'2004',
'must',
'go',
'txt',
'nokia',
'89545',
'collect',
'today',
'2optout',
'u',
'meet',
'ur',
'dream',
'partner',
'soon',
'ur',
'career',
```

```
'2',
'flyng',
'start',
'2',
'find',
'free',
'txt',
'horo',
'follow',
'ur',
'star',
'sign',
'horo',
'ari',
'text',
'meet',
'someon',
'sexi',
'today',
'u',
'find',
'date',
'even',
'flirt',
'join',
'4',
'10p',
'repli',
'name',
'age',
'eg',
'sam',
'25',
'18',
'recd',
'thirtyeight',
'penc',
'u',
'447801259231',
'secret',
'admir',
'look',
'2',
'make',
'contact',
'r',
'reveal',
'think',
'ur',
'09058094597',
'congratul',
'ur',
'award',
'500',
'cd',
'voucher',
'125gift',
'guarante',
'free',
'entri',
```

```
'2',
'100',
'wkli',
'draw',
'txt',
'music',
'87066',
'tnc',
'tri',
'contact',
'repli',
'offer',
'video',
'handset',
'750',
'anytim',
'network',
'min',
'unlimit',
'text',
'camcord',
'repli',
'call',
'08000930705',
'hey',
'realli',
'horni',
'want',
'chat',
'see',
'nake',
'text',
'hot',
'69698',
'text',
'charg',
'150pm',
'unsubscrib',
'text',
'stop',
'69698',
'ur',
'rington',
'servic',
'chang',
'25',
'free',
'credit',
'go',
'choos',
'content',
'stop',
'txt',
'club',
'stop',
'87070',
'club4',
'po',
'box1146',
'mk45',
```

```
'2wt',
'rington',
'club',
'get',
'uk',
'singl',
'chart',
'mobil',
'week',
'choos',
'top',
'qualiti',
'rington',
'messag',
'free',
'charg',
'hmv',
'bonu',
'special',
'500',
'pound',
'genuin',
'hmv',
'voucher',
'answer',
'4',
'easi',
'question',
'play',
'send',
'hmv',
'86688',
'info',
'custom',
'may',
'claim',
'free',
'camera',
'phone',
'upgrad',
'pay',
'go',
'sim',
'card',
'loyalti',
'call',
'0845',
'021',
'end',
'c',
'appli',
'sm',
'ac',
'blind',
'date',
'4u',
'rodds1',
'aberdeen',
'unit',
'kingdom',
```

```
'check',
'http',
'sm',
'blind',
'date',
'send',
'hide',
'themob',
'check',
'newest',
'select',
'content',
'game',
'tone',
'gossip',
'babe',
'sport',
'keep',
'mobil',
'fit',
'funki',
'text',
'wap',
'82468',
'think',
'ur',
'smart',
'win',
'week',
'weekli',
'quiz',
'text',
'play',
'85222',
'cs',
'winnersclub',
'po',
'box',
'84',
'm26',
'3uz',
'decemb',
'mobil',
'entitl',
'updat',
'latest',
'colour',
'camera',
'mobil',
'free',
'call',
'mobil',
'updat',
'co',
'free',
'08002986906',
'call',
'germani',
'1',
'penc',
```

```
'per',
'minut',
'call',
'fix',
'line',
'via',
'access',
'number',
'0844',
'861',
'85',
'prepay',
'direct',
'access',
'valentin',
'day',
'special',
'win',
'quiz',
'take',
'partner',
'trip',
'lifetim',
'send',
'go',
'83600',
'rcvd',
'fanci',
'shag',
'txt',
'xxuk',
'suzi',
'txt',
'cost',
'per',
'msg',
'tnc',
'websit',
'x',
'ur',
'current',
'500',
'pound',
'maxim',
'ur',
'send',
'cash',
'86688',
'cc',
'08708800282',
'xma',
'offer',
'latest',
'motorola',
'sonyericsson',
'nokia',
'free',
'bluetooth',
'doubl',
'min',
```

```
'1000',
'txt',
'orang',
'call',
'mobileupd8',
'08000839402',
'discount',
'code',
'rp176781',
'stop',
'messag',
'repli',
'stop',
'custom',
'servic',
'08717205546',
'thank',
'rington',
'order',
'refer',
't91',
'charg',
'gbp',
'4',
'per',
'week',
'unsubscrib',
'anytim',
'call',
'custom',
'servic',
'09057039994',
'doubl',
'min',
'txt',
'4',
'6month',
'free',
'bluetooth',
'orang',
'avail',
'soni',
'nokia',
'motorola',
'phone',
'call',
'mobileupd8',
'08000839402',
'4mth',
'half',
'price',
'orang',
'line',
'rental',
'latest',
'camera',
'phone',
'4',
'free',
'phone',
```

```
'11mth',
'call',
'mobilesdirect',
'free',
'08000938767',
'updat',
'or2stoptxt',
'free',
'rington',
'text',
'first',
'87131',
'poli',
'text',
'get',
'87131',
'true',
'tone',
'help',
'0845',
'2814032',
'16',
'1st',
'free',
'tone',
'txt',
'stop',
'100',
'date',
'servic',
'cal',
'l',
'09064012103',
'box334sk38ch',
'free',
'entri',
'weekli',
'competit',
'text',
'word',
'win',
'80086',
'18',
'c',
'send',
'logo',
'2',
'ur',
'lover',
'2',
'name',
'join',
'heart',
'txt',
'love',
'name1',
'name2',
'mobno',
'eg',
'love',
```

```
 'adam',
 'eve',
 '07123456789',
 '87077',
 'yahoo',
 'pobox36504w45wq',
 'txtno',
 '4',
 'ad',
 '150p',
 'someon',
 'contact',
 'date',
 'servic',
 'enter',
 'phone',
 'fanci',
 'find',
 'call',
 'landlin',
 '09111032124',
 'pobox12n146tf150p',
 'urgent',
 'mobil',
 'number',
 'award',
 'prize',
 'guarante',
 'call',
 '09058094455',
 'land',
 'line',
 'claim',
 'valid',
 '12hr',
 'congrat',
 'nokia',
 '3650',
 'video',
 'camera',
 ...]
```

In [54]:
```
#length of spam words
len(spam_corpus)
```

Out[54]:
```
9939
```

In [55]:
```
#convert all words in key value pair - it is showing how much time the word is repeate
from collections import Counter
Counter(spam_corpus)
```

Out[55]:
```
Counter({'free': 191,
         'entri': 21,
         '2': 155,
         'wkli': 9,
         'comp': 8,
         'win': 48,
         'fa': 2,
         'cup': 3,
         'final': 14,
         'tkt': 2,
         '21st': 1,
         'may': 6,
         'text': 122,
         '87121': 2,
         'receiv': 31,
         'question': 9,
         'std': 6,
         'txt': 141,
         'rate': 26,
         'c': 45,
         'appli': 24,
         '08452810075over18': 1,
         'freemsg': 14,
         'hey': 5,
         'darl': 2,
         '3': 20,
         'week': 49,
         'word': 21,
         'back': 20,
         'like': 12,
         'fun': 8,
         'still': 5,
         'tb': 1,
         'ok': 5,
         'xxx': 10,
         'chg': 2,
         'send': 60,
         'rcv': 2,
         'winner': 13,
         'valu': 8,
         'network': 26,
         'custom': 42,
         'select': 26,
         'receivea': 1,
         'prize': 82,
         'reward': 9,
         'claim': 98,
         'call': 320,
         'code': 27,
         'kl341': 1,
         'valid': 21,
         '12': 3,
         'hour': 4,
         'mobil': 114,
         '11': 3,
         'month': 5,
         'u': 119,
         'r': 24,
         'entitl': 6,
         'updat': 14,
```

```
 'latest': 30,
 'colour': 14,
 'camera': 23,
 'co': 3,
 '08002986030': 1,
 'six': 2,
 'chanc': 22,
 'cash': 51,
 '100': 14,
 'pound': 19,
 'csh11': 1,
 'cost': 24,
 '6day': 1,
 'tsandc': 1,
 'repli': 103,
 'hl': 3,
 '4': 97,
 'info': 11,
 'urgent': 57,
 '1': 28,
 'membership': 1,
 'jackpot': 1,
 '81010': 1,
 'lccltd': 1,
 'pobox': 11,
 '4403ldnw1a7rw18': 1,
 'xxxmobilemovieclub': 1,
 'use': 12,
 'credit': 18,
 'click': 5,
 'wap': 9,
 'link': 6,
 'next': 16,
 'messag': 42,
 'http': 18,
 'england': 7,
 'v': 4,
 'macedonia': 1,
 'dont': 9,
 'miss': 8,
 'news': 8,
 'ur': 119,
 'nation': 11,
 'team': 2,
 '87077': 8,
 'eg': 11,
 'tri': 37,
 'wale': 1,
 'scotland': 1,
 'poboxox36504w45wq': 1,
 'thank': 16,
 'subscript': 5,
 'rington': 30,
 'uk': 20,
 'charg': 22,
 'pleas': 51,
 'confirm': 1,
 'ye': 12,
 '07732584351': 1,
 'rodger': 1,
```

```
'burn': 1,
'msg': 35,
'sm': 22,
'nokia': 57,
'camcord': 15,
'08000930705': 16,
'deliveri': 18,
'tomorrow': 10,
'ac': 4,
'sptv': 2,
'new': 64,
'jersey': 1,
'devil': 1,
'detroit': 1,
'red': 1,
'wing': 1,
'play': 14,
'ice': 1,
'hockey': 1,
'correct': 4,
'incorrect': 1,
'end': 24,
'congrat': 6,
'year': 9,
'special': 16,
'cinema': 2,
'pass': 5,
'09061209465': 2,
'suprman': 2,
'matrix3': 2,
'starwars3': 2,
'etc': 6,
'150pm': 4,
'advis': 2,
'follow': 10,
'recent': 4,
'review': 3,
'mob': 20,
'award': 55,
'bonu': 17,
'09066364589': 1,
'complimentari': 11,
'trip': 3,
'eurodisinc': 1,
'trav': 1,
'aco': 1,
'entry41': 1,
'di': 1,
'morefrmmob': 1,
'shracomorsglsuplt': 1,
'10': 7,
'ls1': 1,
'3aj': 1,
'hear': 3,
'come': 5,
'ken': 1,
'stuff': 1,
'servic': 64,
'repres': 6,
'0800': 9,
```

```
'169': 1,
'6031': 1,
'guarante': 42,
'wait': 17,
'collect': 45,
'simpli': 2,
'password': 2,
'85069': 1,
'verifi': 2,
'get': 74,
'usher': 1,
'britney': 1,
'fml': 1,
'gent': 2,
'contact': 56,
'last': 12,
'weekend': 13,
'draw': 35,
'show': 33,
'k52': 3,
'12hr': 16,
'150ppm': 29,
'holiday': 27,
'flight': 6,
'inc': 7,
'speak': 7,
'live': 22,
'oper': 11,
'privat': 16,
'2004': 7,
'account': 18,
'statement': 15,
'07742676969': 1,
'786': 3,
'unredeem': 3,
'point': 15,
'08719180248': 1,
'identifi': 15,
'45239': 2,
'expir': 17,
'caller': 12,
'landlin': 30,
'09064019788': 1,
'box42wr29c': 1,
'today': 33,
'voda': 5,
'number': 35,
'7548': 1,
'350': 2,
'match': 14,
'08712300220': 6,
'quot': 6,
'4041': 1,
'standard': 6,
'app': 4,
'sunshin': 5,
'quiz': 9,
'q': 2,
'top': 14,
'soni': 5,
```

```
'dvd': 5,
'player': 10,
'know': 19,
'countri': 2,
'algarv': 1,
'ansr': 2,
'sp': 6,
'tyron': 2,
'want': 31,
'laid': 4,
'tonight': 3,
'real': 12,
'dog': 12,
'locat': 5,
'sent': 10,
'direct': 10,
'join': 18,
'largest': 4,
'bt': 5,
'txting': 7,
'gravel': 1,
'69888': 2,
'nt': 1,
'ec2a': 4,
'150p': 26,
'chat': 38,
'svc': 1,
'hardcor': 2,
'go': 32,
'69988': 1,
'noth': 4,
'must': 5,
'age': 9,
'yr': 14,
'randi': 2,
'sexi': 14,
'femal': 1,
'local': 5,
'luv': 5,
'netcollex': 4,
'ltd': 11,
'08700621170150p': 2,
'per': 41,
'stop': 104,
'annonc': 1,
'07046744435': 1,
'arrang': 2,
'0871277810810': 1,
'bootydeli': 1,
'invit': 10,
'friend': 13,
'see': 16,
'frnd': 5,
'62468': 5,
'bangbab': 1,
'order': 17,
'way': 1,
'download': 7,
'content': 14,
'goto': 5,
```

```
'bangb': 1,
'tv': 3,
'menu': 1,
's89': 1,
'freephon': 4,
'0808': 1,
'145': 1,
'4742': 1,
'uniqu': 2,
'enough': 2,
'find': 21,
'30th': 1,
'august': 2,
'500': 19,
'89545': 3,
'2optout': 4,
'meet': 5,
'dream': 2,
'partner': 5,
'soon': 2,
'career': 1,
'flyng': 1,
'start': 12,
'horo': 2,
'star': 3,
'sign': 2,
'ari': 1,
'someon': 13,
'date': 26,
'even': 7,
'flirt': 5,
'10p': 13,
'name': 15,
'sam': 2,
'25': 3,
'18': 20,
'recd': 3,
'thirtyeight': 1,
'penc': 3,
'447801259231': 1,
'secret': 7,
'admir': 7,
'look': 9,
'make': 11,
'reveal': 12,
'think': 11,
'09058094597': 1,
'congratul': 12,
'cd': 11,
'voucher': 28,
'125gift': 2,
'music': 15,
'87066': 10,
'tnc': 7,
'offer': 33,
'video': 29,
'handset': 6,
'750': 17,
'anytim': 12,
'min': 45,
```

```
 'unlimit': 10,
 'realli': 1,
 'horni': 7,
 'nake': 1,
 'hot': 13,
 '69698': 2,
 'unsubscrib': 17,
 'chang': 1,
 'choos': 9,
 'club': 19,
 '87070': 1,
 'club4': 1,
 'po': 24,
 'box1146': 1,
 'mk45': 1,
 '2wt': 1,
 'singl': 5,
 'chart': 3,
 'qualiti': 2,
 'hmv': 4,
 'genuin': 1,
 'answer': 6,
 'easi': 10,
 '86688': 14,
 'phone': 52,
 'upgrad': 4,
 'pay': 2,
 'sim': 2,
 'card': 4,
 'loyalti': 6,
 '0845': 3,
 '021': 2,
 'blind': 2,
 '4u': 4,
 'rodds1': 1,
 'aberdeen': 1,
 'unit': 1,
 'kingdom': 1,
 'check': 4,
 'hide': 1,
 'themob': 5,
 'newest': 3,
 'game': 20,
 'tone': 63,
 'gossip': 1,
 'babe': 10,
 'sport': 7,
 'keep': 5,
 'fit': 1,
 'funki': 1,
 '82468': 2,
 'smart': 2,
 'weekli': 20,
 '85222': 1,
 'cs': 35,
 'winnersclub': 1,
 'box': 26,
 '84': 1,
 'm26': 3,
 '3uz': 3,
```

```
'decemb': 4,
'08002986906': 3,
'germani': 3,
'minut': 12,
'fix': 2,
'line': 33,
'via': 3,
'access': 8,
'0844': 2,
'861': 2,
'85': 2,
'prepay': 2,
'valentin': 3,
'day': 26,
'take': 16,
'lifetim': 2,
'83600': 3,
'rcvd': 5,
'fanci': 12,
'shag': 2,
'xxuk': 1,
'suzi': 1,
'websit': 1,
'x': 8,
'current': 11,
'maxim': 7,
'cc': 6,
'08708800282': 1,
'xma': 15,
'motorola': 10,
'sonyericsson': 5,
'bluetooth': 7,
'doubl': 13,
'1000': 9,
'orang': 24,
'mobileupd8': 13,
'08000839402': 12,
'discount': 6,
'rp176781': 1,
'08717205546': 1,
'refer': 4,
't91': 1,
'gbp': 1,
'09057039994': 1,
'6month': 1,
'avail': 2,
'4mth': 2,
'half': 12,
'price': 15,
'rental': 11,
'11mth': 3,
'mobilesdirect': 2,
'08000938767': 2,
'or2stoptxt': 2,
'first': 6,
'87131': 3,
'poli': 22,
'true': 2,
'help': 12,
'2814032': 1,
```

```
'16': 18,
'1st': 19,
'cal': 1,
'l': 1,
'09064012103': 1,
'box334sk38ch': 2,
'competit': 2,
'80086': 2,
'logo': 6,
'lover': 2,
'heart': 3,
'love': 10,
'name1': 2,
'name2': 2,
'mobno': 2,
'adam': 2,
'eve': 5,
'07123456789': 2,
'yahoo': 2,
'pobox36504w45wq': 5,
'txtno': 2,
'ad': 4,
'enter': 15,
'09111032124': 1,
'pobox12n146tf150p': 1,
'09058094455': 1,
'land': 16,
'3650': 1,
'09066382422': 1,
'ave': 2,
'3min': 3,
'vari': 4,
'close': 3,
'300603': 1,
'post': 4,
'bcm4284': 1,
'ldn': 7,
'wc1n3xx': 3,
'loan': 3,
'purpos': 2,
'homeown': 1,
'tenant': 2,
'welcom': 8,
'previous': 1,
'refus': 2,
'1956669': 1,
'upgrdcentr': 1,
'0207': 2,
'153': 2,
'26th': 1,
'juli': 2,
'okmail': 1,
'dear': 16,
'dave': 2,
'notic': 3,
'tenerif': 7,
'5000': 4,
'09061743806': 2,
'tc': 7,
'sae': 18,
```

```
'box326': 2,
'cw25wx': 4,
'moan': 1,
'69888nyt': 1,
'activ': 5,
'term': 8,
'condit': 4,
'visit': 5,
'09050002311': 1,
'b4280703': 1,
'40gb': 6,
'ipod': 10,
'mp3': 5,
'83355': 4,
'ibhltd': 3,
'ldnw15h': 3,
'boltblu': 1,
'mono': 4,
'poly3': 1,
'cha': 2,
'slide': 1,
'yeah': 1,
'slow': 1,
'jamz': 1,
'toxic': 1,
'renew': 3,
'pin': 1,
'tgxxrz': 1,
'2nd': 19,
'attempt': 22,
'box95qu': 3,
'worth': 8,
'85023': 4,
'savamob': 8,
'member': 6,
'sub': 5,
'unsub': 3,
'reciev': 1,
'within': 4,
'24hr': 2,
'channel': 1,
'teletext': 1,
'pg': 1,
'2003': 13,
'07815296484': 1,
'800': 13,
'08718738001': 2,
'41782': 1,
'monthlysubscript': 1,
'csc': 1,
'web': 2,
'age16': 6,
'2stop': 1,
'call09050000327': 2,
'us': 7,
'ring': 5,
'09050005321': 1,
'150': 9,
'textand': 1,
'08002988890': 1,
```

```
'shop': 10,
'spree': 6,
'custcar': 7,
'08715705022': 3,
'2000': 2,
'08712402050': 2,
'10ppm': 2,
'ag': 2,
'promo': 2,
'07753741225': 1,
'08715203677': 1,
'42478': 1,
'import': 11,
'announc': 5,
'542': 3,
'0825': 1,
'xclusiv': 1,
'clubsaisai': 1,
'2morow': 1,
'soire': 1,
'zouk': 1,
'nichol': 1,
'rose': 1,
'ladi': 3,
'22': 1,
'kick': 3,
'euro2004': 3,
'kept': 2,
'result': 2,
'daili': 3,
'remov': 7,
'83222': 2,
'textbuddi': 2,
'guy': 5,
'area': 8,
'25p': 7,
'search': 2,
'postcod': 3,
'one': 10,
'89693': 2,
'vodafon': 5,
'4882': 1,
'09064019014': 1,
'holder': 6,
'pc': 6,
'ts': 9,
'80062': 5,
'08715203694': 1,
'40533': 2,
'rstm': 2,
'sw7': 2,
'3ss': 2,
'88800': 1,
'89034': 1,
'premium': 2,
'08718711108': 1,
'sun0819': 1,
'hello': 4,
'seem': 1,
'cool': 1,
```

```
'gr8': 7,
'20': 3,
'everi': 26,
'wk': 17,
'opt': 10,
'08452810071': 1,
'hi': 15,
'sue': 2,
'old': 2,
'work': 3,
'lapdanc': 1,
'sex': 10,
'bedroom': 1,
'textoper': 3,
'g2': 1,
'1da': 1,
'150ppmsg': 1,
'forward': 4,
'448712404000': 1,
'08712404000': 1,
'immedi': 5,
'fantast': 7,
'deck': 1,
'alert': 5,
'08714712388': 1,
'09071512433': 2,
'b4': 6,
'050703': 2,
'csbcm4235wc1n3xx': 2,
'callcost': 2,
'mobilesvari': 2,
'50': 3,
'08714712394': 1,
'email': 2,
'alertfrom': 1,
'jeri': 1,
'stewarts': 1,
'2kbsubject': 1,
'prescripiton': 1,
'drvgsto': 1,
'listen': 3,
'123': 1,
'nokia6650': 2,
'txtauction': 5,
'81151': 2,
'4t': 3,
'ctxt': 2,
'subscrib': 9,
'best': 7,
'helplin': 2,
'08706091795': 2,
'realiz': 1,
'40': 2,
'thousand': 1,
'run': 1,
'around': 2,
'tattoo': 1,
'premier': 2,
'romant': 1,
'pari': 3,
```

```
'night': 7,
'book': 6,
'08704439680t': 1,
'unclaim': 1,
'09066368327': 1,
'claimcod': 1,
'm39m51': 1,
'citi': 3,
'break': 5,
'could': 7,
'summer': 7,
'store': 7,
'88039': 2,
'skilgm': 3,
'tscs087147403231winawk': 2,
'0578': 2,
'ever': 3,
'thought': 1,
'good': 12,
'life': 3,
'perfect': 1,
'commun': 2,
'5': 8,
'polyphon': 4,
'087018728737': 1,
'toppoli': 1,
'tune': 1,
'subpoli': 2,
'81618': 1,
'pole': 1,
'08718727870': 2,
'14thmarch': 1,
'availa': 1,
'pobox84': 4,
'm263uz': 2,
'no1': 5,
'8077': 1,
'tell': 13,
'mate': 8,
'36504': 4,
'w45wq': 3,
'cashto': 2,
'08000407165': 2,
'getstop': 2,
'88222': 2,
'php': 2,
'rg21': 1,
'4jx': 1,
'either': 8,
'gift': 16,
'outbid': 1,
'simonwatson5120': 1,
'shinco': 1,
'plyr': 1,
'bid': 8,
'notif': 1,
'smsservic': 1,
'yourinclus': 1,
'pl': 8,
'3qxj9': 3,
```

```
'extra': 6,
'9ae': 3,
'alfi': 3,
'moon': 3,
'children': 3,
'need': 11,
'song': 3,
'm8': 4,
'chariti': 9,
'8007': 19,
'zed': 6,
'08701417012': 3,
'profit': 3,
'cust': 3,
'care': 6,
'07821230901': 1,
'five': 3,
'08002888812': 2,
'wed': 2,
'09066350750': 2,
'ibiza': 4,
'await': 24,
'434': 3,
'sk3': 3,
'8wp': 3,
'ppm': 4,
'talk': 5,
'fall': 1,
'world': 2,
'discreet': 2,
'vip': 4,
'83110': 1,
'suppli': 2,
'virgin': 2,
'record': 6,
'mysteri': 1,
'09061104283': 1,
'approx': 1,
'07808': 1,
'xxxxxx': 1,
'08719899217': 1,
'41685': 2,
'posh': 1,
'bird': 1,
'chap': 1,
'user': 8,
'trial': 1,
'prod': 1,
'champney': 1,
'put': 1,
'address': 3,
'dob': 1,
'asap': 6,
'ta': 1,
'0721072': 1,
'till': 2,
'drop': 1,
'10k': 2,
'5k': 1,
'travel': 1,
```

```
'ntt': 7,
'cr01327bt': 1,
'fixedlin': 1,
'liverpool': 2,
'mid': 1,
'09058094565': 2,
'remind': 2,
'alreadi': 1,
'paid': 1,
'mymobi': 1,
'lastest': 1,
'stereophon': 1,
'marley': 1,
'dizze': 1,
'racal': 1,
'libertin': 1,
'stroke': 1,
'nookii': 1,
'bookmark': 1,
'januari': 1,
'male': 3,
'sale': 4,
'gay': 6,
'cheaper': 2,
'cheap': 3,
'peak': 2,
'08712460324': 8,
'money': 4,
'lucki': 8,
'88600': 2,
'give': 8,
'away': 4,
'box403': 1,
'w1t1ji': 1,
'matthew': 1,
'09063440451': 1,
'lux': 2,
'ppm150': 1,
'box334': 1,
'sk38xh': 4,
'09061749602': 1,
'528': 1,
'hp20': 1,
'1yf': 1,
'touch': 1,
'folk': 1,
'compani': 3,
'enjoy': 12,
'08718720201': 5,
'filthi': 2,
'stori': 1,
'girl': 9,
'09050001808': 1,
'm95': 1,
'valid12hr': 2,
'3g': 3,
'videophon': 3,
'09063458130': 2,
'videochat': 3,
'wid': 3,
```

```
'java': 3,
'dload': 3,
'polyph': 2,
'nolin': 3,
'rentl': 3,
'panason': 1,
'bluetoothhdset': 1,
'doublemin': 1,
'doubletxt': 1,
'contract': 4,
'guess': 6,
'somebodi': 2,
'secretli': 2,
'wan': 11,
'na': 10,
'09065394514': 1,
'datebox1282essexcm61xn': 2,
'09058097218': 1,
'6': 5,
'ls15hb': 2,
'bloke': 1,
'zoe': 1,
'kickoff': 1,
'inform': 9,
'euro': 2,
'eastend': 1,
'flower': 2,
'dot': 1,
'compar': 1,
'violet': 1,
'tulip': 1,
'lili': 1,
'e': 1,
'f': 3,
'84025': 2,
'lot': 2,
'peopl': 3,
'regist': 5,
'replys150': 1,
'ask': 5,
'cant': 3,
'09058091854': 1,
'box385': 1,
'm6': 1,
'6wu': 1,
'09050003091': 2,
'c52': 2,
'xchat': 5,
'sipix': 3,
'digit': 6,
'09061221061': 1,
'28day': 1,
'box177': 1,
'm221bp': 1,
'2yr': 1,
'warranti': 1,
'p': 2,
'09061790121': 2,
'3030': 1,
'b': 10,
```

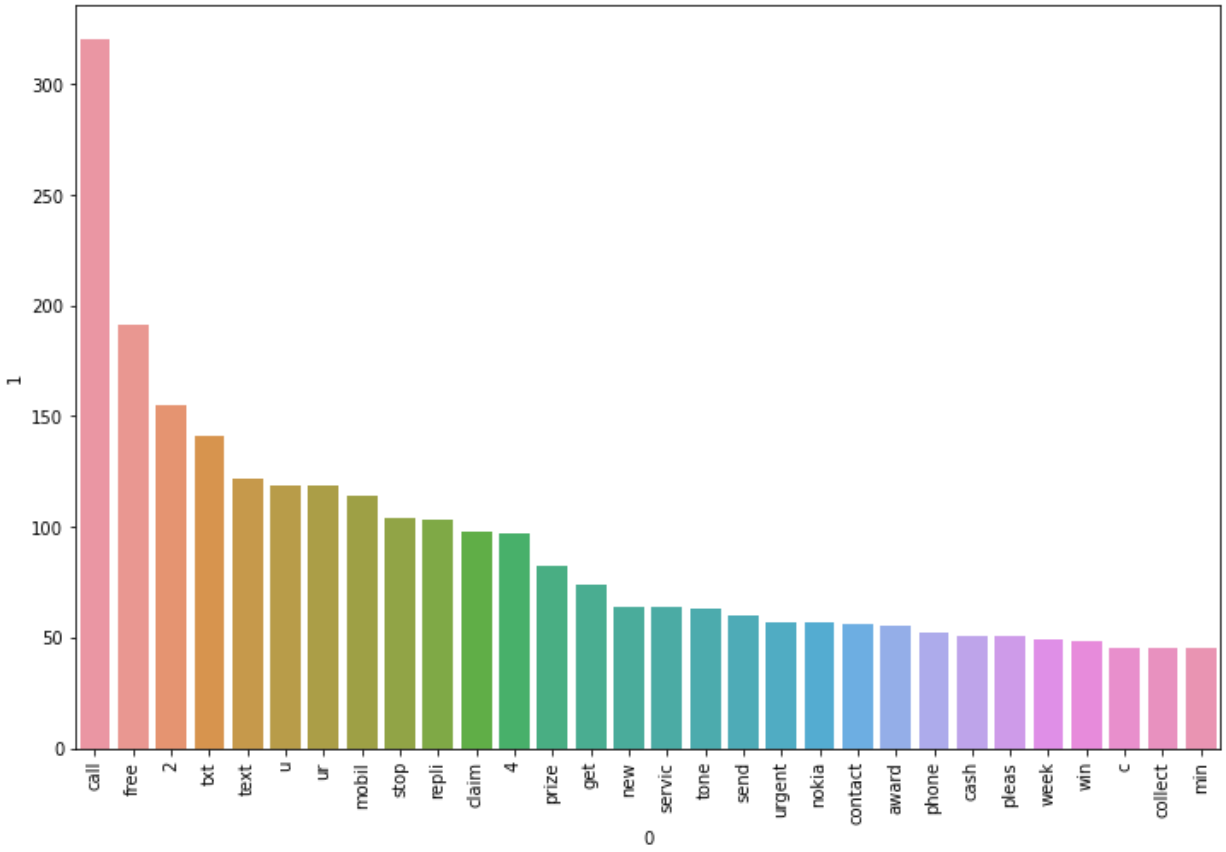
```
'receipt': 3,
'an': 6,
'elvi': 1,
'presley': 1,
'birthday': 1,
'o2': 5,
'log': 7,
'onto': 7,
'surpris': 5,
'449050000301': 1,
'09050000301': 1,
'bore': 3,
'speed': 1,
'speedchat': 2,
'80155': 1,
'em': 1,
'swap': 1,
'chatter': 1,
'chat80155': 1,
'rcd': 1,
'08000776320': 2,
'part': 8,
'survey': 1,
'yesterday': 3,
'howev': 1,
'wish': 2,
'80160': 1,
'hmv1': 1,
'forget': 2,
'place': 4,
'mani': 3,
'request': 2,
'08707808226': 1,
'let': 3,
'notifi': 1,
'luck': 6,
'futur': 3,
'market': 1,
'84122': 1,
'08450542832': 1,
...})
```

In [56]:
```
#fitler top 30 spam words
Counter(spam_corpus).most_common(30)
```

Out[56]:
```
[('call', 320),
 ('free', 191),
 ('2', 155),
 ('txt', 141),
 ('text', 122),
 ('u', 119),
 ('ur', 119),
 ('mobil', 114),
 ('stop', 104),
 ('repli', 103),
 ('claim', 98),
 ('4', 97),
 ('prize', 82),
 ('get', 74),
 ('new', 64),
 ('servic', 64),
 ('tone', 63),
 ('send', 60),
 ('urgent', 57),
 ('nokia', 57),
 ('contact', 56),
 ('award', 55),
 ('phone', 52),
 ('cash', 51),
 ('pleas', 51),
 ('week', 49),
 ('win', 48),
 ('c', 45),
 ('collect', 45),
 ('min', 45)]
```

In [57]:
```
#see the count od top 30 words in spam messages
plt.figure(figsize=(12,8))
sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0] , pd.DataFrame(Count

plt.xticks(rotation='vertical')

plt.show()
```

```
C:\Users\rupeshv\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variables as keyword args: x, y. From version 0.12, the only va
lid positional argument will be `data`, and passing other arguments without an explic
it keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [58]:
```
ham_corpus = []
for msg in df[df['target']==0]['transformed_text'].tolist():
    for word in msg.split():
        ham_corpus.append(word)

ham_corpus
```

Out[58]:
```
['go',
 'jurong',
 'point',
 'crazi',
 'avail',
 'bugi',
 'n',
 'great',
 'world',
 'la',
 'e',
 'buffet',
 'cine',
 'got',
 'amor',
 'wat',
 'ok',
 'lar',
 'joke',
 'wif',
 'u',
 'oni',
 'u',
 'dun',
 'say',
 'earli',
 'hor',
 'u',
 'c',
 'alreadi',
 'say',
 'nah',
 'think',
 'goe',
 'usf',
 'live',
 'around',
 'though',
 'even',
 'brother',
 'like',
 'speak',
 'treat',
 'like',
 'aid',
 'patent',
 'per',
 'request',
 'mell',
 'oru',
 'minnaminungint',
 'nurungu',
 'vettam',
 'set',
 'callertun',
 'caller',
 'press',
 '9',
 'copi',
 'friend',
```

```
'callertun',
'gon',
'na',
'home',
'soon',
'want',
'talk',
'stuff',
'anymor',
'tonight',
'k',
'cri',
'enough',
'today',
'search',
'right',
'word',
'thank',
'breather',
'promis',
'wont',
'take',
'help',
'grant',
'fulfil',
'promis',
'wonder',
'bless',
'time',
'date',
'sunday',
'oh',
'k',
'watch',
'eh',
'u',
'rememb',
'2',
'spell',
'name',
'ye',
'v',
'naughti',
'make',
'v',
'wet',
'fine',
'thatåõ',
'way',
'u',
'feel',
'thatåõ',
'way',
'gota',
'b',
'serious',
'spell',
'name',
'go',
'tri',
```

```
'2',
'month',
'ha',
'ha',
'joke',
'pay',
'first',
'lar',
'da',
'stock',
'comin',
'aft',
'finish',
'lunch',
'go',
'str',
'lor',
'ard',
'3',
'smth',
'lor',
'u',
'finish',
'ur',
'lunch',
'alreadi',
'fffffffff',
'alright',
'way',
'meet',
'sooner',
'forc',
'eat',
'slice',
'realli',
'hungri',
'tho',
'suck',
'mark',
'get',
'worri',
'know',
'sick',
'turn',
'pizza',
'lol',
'lol',
'alway',
'convinc',
'catch',
'bu',
'fri',
'egg',
'make',
'tea',
'eat',
'mom',
'left',
'dinner',
'feel',
```

```
'love',
'back',
'amp',
'pack',
'car',
'let',
'know',
'room',
'ahhh',
'work',
'vagu',
'rememb',
'feel',
'like',
'lol',
'wait',
'still',
'clear',
'sure',
'sarcast',
'x',
'want',
'live',
'us',
'yeah',
'got',
'2',
'v',
'apologet',
'n',
'fallen',
'actin',
'like',
'spoilt',
'child',
'got',
'caught',
'till',
'2',
'wo',
'go',
'badli',
'cheer',
'k',
'tell',
'anyth',
'fear',
'faint',
'housework',
'quick',
'cuppa',
'yup',
'ok',
'go',
'home',
'look',
'time',
'msg',
'xuhui',
'go',
```

```
'learn',
'2nd',
'may',
'lesson',
'8am',
'oop',
'let',
'know',
'roommat',
'done',
'see',
'letter',
'b',
'car',
'anyth',
'lor',
'u',
'decid',
'hello',
'saturday',
'go',
'text',
'see',
'decid',
'anyth',
'tomo',
'tri',
'invit',
'anyth',
'pl',
'go',
'ahead',
'watt',
'want',
'sure',
'great',
'weekend',
'abiola',
'forget',
'tell',
'want',
'need',
'crave',
'love',
'sweet',
'arabian',
'steed',
'mmmmmm',
'yummi',
'see',
'great',
'hope',
'like',
'man',
'well',
'endow',
'lt',
'gt',
'inch',
'call',
```

```
'messag',
'miss',
'call',
'get',
'hep',
'b',
'immunis',
'nigeria',
'fair',
'enough',
'anyth',
'go',
'yeah',
'hope',
'tyler',
'ca',
'could',
'mayb',
'ask',
'around',
'bit',
'u',
'know',
'stubborn',
'even',
'want',
'go',
'hospit',
'kept',
'tell',
'mark',
'weak',
'sucker',
'hospit',
'weak',
'sucker',
'think',
'first',
'time',
'saw',
'class',
'gram',
'usual',
'run',
'like',
'lt',
'gt',
'half',
'eighth',
'smarter',
'though',
'get',
'almost',
'whole',
'second',
'gram',
'lt',
'gt',
'k',
'fyi',
```

```
'x',
'ride',
'earli',
'tomorrow',
'morn',
'crash',
'place',
'tonight',
'wow',
'never',
'realiz',
'embarass',
'accomod',
'thought',
'like',
'sinc',
'best',
'could',
'alway',
'seem',
'happi',
'sorri',
'give',
'sorri',
'offer',
'sorri',
'room',
'embarass',
'know',
'mallika',
'sherawat',
'yesterday',
'find',
'lt',
'url',
'gt',
'sorri',
'call',
'later',
'meet',
'tell',
'reach',
'ye',
'gauti',
'sehwag',
'odi',
'seri',
'gon',
'na',
'pick',
'1',
'burger',
'way',
'home',
'ca',
'even',
'move',
'pain',
'kill',
'ha',
```

```
'ha',
'ha',
'good',
'joke',
'girl',
'situat',
'seeker',
'part',
'check',
'iq',
'sorri',
'roommat',
'took',
'forev',
'ok',
'come',
'ok',
'lar',
'doubl',
'check',
'wif',
'da',
'hair',
'dresser',
'alreadi',
'said',
'wun',
'cut',
'v',
'short',
'said',
'cut',
'look',
'nice',
'today',
'dedic',
'day',
'song',
'u',
'dedic',
'send',
'ur',
'valuabl',
'frnd',
'first',
'rpli',
'plane',
'give',
'month',
'end',
'wah',
'lucki',
'man',
'save',
'money',
'hee',
'finish',
'class',
'hi',
'babe',
```

```
'im',
'home',
'wan',
'na',
'someth',
'xx',
'k',
'k',
'perform',
'u',
'call',
'wait',
'machan',
'call',
'free',
'that',
'cool',
'gentleman',
'treat',
'digniti',
'respect',
'like',
'peopl',
'much',
'shi',
'pa',
'oper',
'lt',
'gt',
'still',
'look',
'job',
'much',
'ta',
'earn',
'sorri',
'call',
'later',
'call',
'ah',
'ok',
'way',
'home',
'hi',
'hi',
'place',
'man',
'yup',
'next',
'stop',
'call',
'later',
'network',
'urgnt',
'sm',
'real',
'u',
'get',
'yo',
'need',
```

```
'2',
'ticket',
'one',
'jacket',
'done',
'alreadi',
'use',
'multi',
'ye',
'start',
'send',
'request',
'make',
'pain',
'came',
'back',
'back',
'bed',
'doubl',
'coin',
'factori',
'got',
'ta',
'cash',
'nitro',
'realli',
'still',
'tonight',
'babe',
'ela',
'il',
'download',
'come',
'wen',
'ur',
'free',
'yeah',
'stand',
'close',
'catch',
'someth',
'sorri',
'pain',
'ok',
'meet',
'anoth',
'night',
'spent',
'late',
'afternoon',
'casualti',
'mean',
'done',
'stuff42moro',
'includ',
'time',
'sheet',
'sorri',
'smile',
'pleasur',
```

```
'smile',
'pain',
'smile',
'troubl',
'pour',
'like',
'rain',
'smile',
'sum1',
'hurt',
'u',
'smile',
'becoz',
'someon',
'still',
'love',
'see',
'u',
'smile',
'havent',
'plan',
'buy',
'later',
'check',
'alreadi',
'lido',
'got',
'530',
'show',
'e',
'afternoon',
'u',
'finish',
'work',
'alreadi',
'watch',
'telugu',
'movi',
'wat',
'abt',
'u',
'see',
'finish',
'load',
'loan',
'pay',
'hi',
'wk',
'ok',
'hol',
'ye',
'bit',
'run',
'forgot',
'hairdress',
'appoint',
'four',
'need',
'get',
'home',
```

```
'n',
'shower',
'beforehand',
'caus',
'prob',
'u',
'ham',
'pleas',
'text',
'anymor',
'noth',
'els',
'say',
'okay',
'name',
'ur',
'price',
'long',
'legal',
'wen',
'pick',
'u',
'ave',
'x',
'am',
'xx',
'still',
'look',
'car',
'buy',
'gone',
'4the',
'drive',
'test',
'yet',
'wow',
'right',
'mean',
'guess',
'gave',
'boston',
'men',
'chang',
'search',
'locat',
'nyc',
'someth',
'chang',
'cuz',
'signin',
'page',
'still',
'say',
'boston',
'umma',
'life',
'vava',
'umma',
'love',
'lot',
```

```
'dear',
'thank',
'lot',
'wish',
'birthday',
'thank',
'make',
'birthday',
'truli',
'memor',
'aight',
'hit',
'get',
'cash',
'would',
'ip',
'address',
'test',
'consid',
'comput',
'minecraft',
'server',
'know',
'grumpi',
'old',
'peopl',
'mom',
'like',
'better',
'lie',
'alway',
'one',
'play',
'joke',
'dont',
'worri',
'guess',
'busi',
'plural',
'noun',
'research',
'go',
'ok',
'wif',
'co',
'like',
'2',
'tri',
'new',
'thing',
'scare',
'u',
'dun',
'like',
'mah',
'co',
'u',
'said',
'loud',
'wa',
```

```
'ur',
'openin',
'sentenc',
'formal',
'anyway',
'fine',
'juz',
'tt',
'eatin',
'much',
'n',
'puttin',
'weight',
'haha',
'anythin',
'special',
'happen',
'enter',
'cabin',
'pa',
'said',
'happi',
'boss',
'felt',
'special',
'askd',
'4',
'lunch',
'lunch',
'invit',
'apart',
'went',
'goodo',
'ye',
'must',
'speak',
'friday',
'ratio',
'tortilla',
'need',
'hmm',
'uncl',
'inform',
'pay',
'school',
'directli',
'pl',
'buy',
'food',
'new',
'address',
'pair',
'malarki',
'go',
'sao',
'mu',
'today',
'done',
'12',
'ìï',
```

```
'predict',
'wat',
'time',
'finish',
'buy',
'good',
'stuff',
'know',
'yetund',
'sent',
'money',
'yet',
'sent',
'text',
'bother',
'send',
'dont',
'involv',
'anyth',
'impos',
'anyth',
'first',
'place',
'apologis',
'room',
'hey',
'girl',
'r',
'u',
'hope',
'u',
'r',
'well',
'del',
'r',
'bak',
'long',
'time',
'c',
'give',
'call',
'sum',
'time',
'lucyxx',
'k',
'k',
'much',
'cost',
'home',
'dear',
'call',
'accomod',
'first',
'answer',
'question',
'haf',
'msn',
'yiju',
'call',
'meet',
```

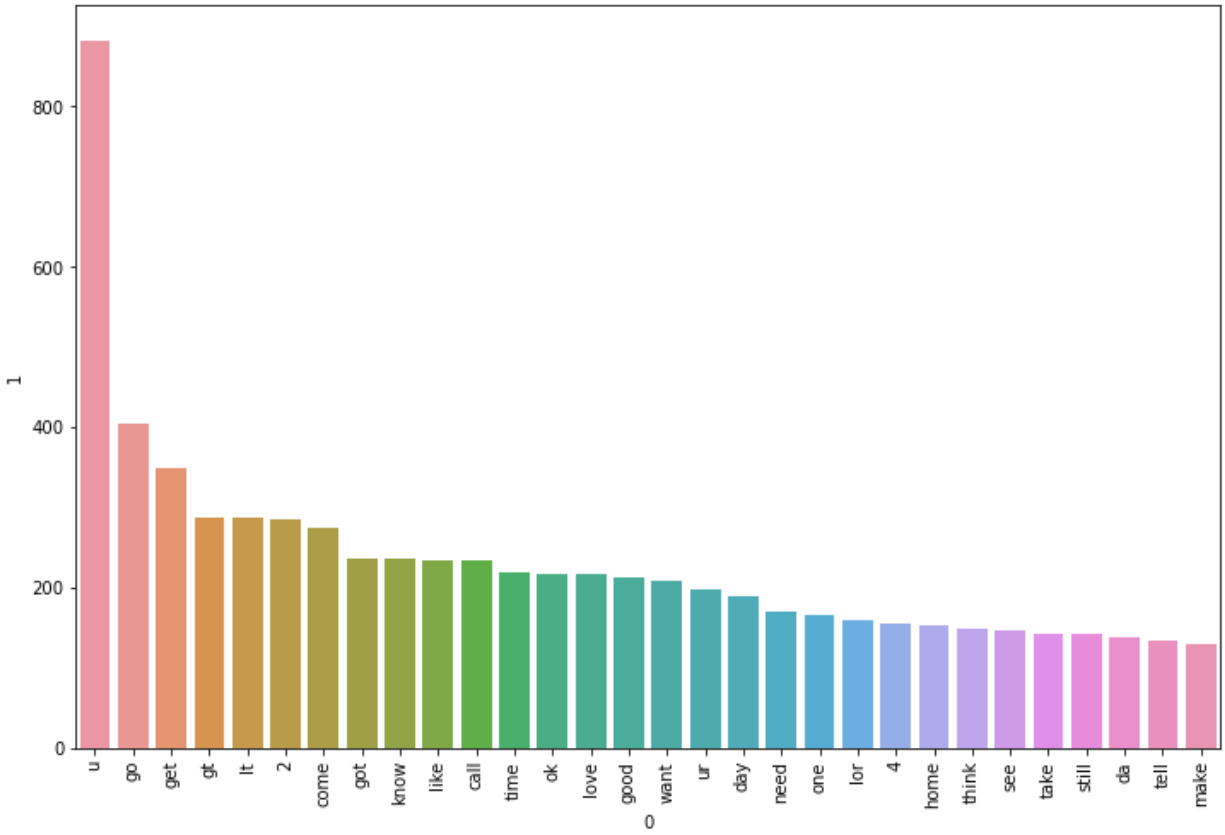
```
'check',
'room',
'befor',
'activ',
'got',
'c',
'lazi',
'type',
'forgot',
'lect',
'saw',
'pouch',
'like',
'v',
'nice',
'k',
'text',
'way',
'sir',
'wait',
'mail',
'swt',
'thought',
'get',
'tire',
'littl',
'thing',
'4',
'lovabl',
'person',
'coz',
'somtim',
'littl',
'thing',
'occupi',
'biggest',
'part',
'heart',
'gud',
'ni8',
'know',
'pl',
'open',
'back',
'ye',
'see',
'ya',
'dot',
'what',
'staff',
'name',
'take',
'class',
'us',
'call',
'check',
'life',
'begin',
'qatar',
'pl',
```

```
 'pray',
 'hard',
 'k',
 'delet',
 'contact',
 'sindu',
 'got',
 'job',
 'birla',
 'soft',
 'wine',
 'flow',
 'never',
 'yup',
 'thk',
 'cine',
 'better',
 'co',
 'need',
 '2',
 'go',
 '2',
 'plaza',
 'mah',
 'ok',
 'ur',
 'typic',
 'repli',
 'everywher',
 'dirt',
 'floor',
 'window',
 'even',
 'shirt',
 'sometim',
 'open',
 'mouth',
 'come',
 'flow',
 'dream',
 ...]
```

In [59]:
```
len(ham_corpus)
```

Out[59]:
```
35394
```

In [60]:
```
plt.figure(figsize=(12,8))
sns.barplot(x=pd.DataFrame(Counter(ham_corpus).most_common(30))[0] , y=pd.DataFrame(Co

plt.xticks(rotation='vertical')

plt.show()
```



All above top 30 words were also part of world clouds

4. Model Building

CountVectorizer

Scikit-learn's CountVectorizer is used to convert a collection of text documents to a vector of term/token counts. It also enables the pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text.

In [61]:
```
from sklearn.feature_extraction.text import CountVectorizer
cv=CountVectorizer()
```

In [62]:
```
X=cv.fit_transform(df['transformed_text']).toarray()
```

In [63]:
```
X.shape
```

Out[63]:
```
(5169, 6708)
```

In [64]:
```
y=df['target'].values
```

In [65]:
```
y
```

Out[65]:
```
array([0, 0, 1, ..., 0, 0, 0])
```

In [66]:
```
y.shape
```

Out[66]:
```
(5169,)
```

In [67]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=.2,random_state=2)

from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

In [68]:
```
from sklearn.naive_bayes import BernoulliNB,GaussianNB,MultinomialNB

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

In [69]:
```
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))
```

```
0.8800773694390716
[[792 104]
 [ 20 118]]
0.5315315315315315
```

In [70]:
```
mnb.fit(X_train,y_train)
y_pred2 = mnb.predict(X_test)
print(accuracy_score(y_test,y_pred2))
print(confusion_matrix(y_test,y_pred2))
print(precision_score(y_test,y_pred2))
```

```
0.9642166344294004
[[871  25]
 [ 12 126]]
0.8344370860927153
```

In [71]:
```
bnb.fit(X_train,y_train)
y_pred3 = bnb.predict(X_test)
print(accuracy_score(y_test,y_pred3))
print(confusion_matrix(y_test,y_pred3))
print(precision_score(y_test,y_pred3))
```

```
0.9700193423597679
[[893   3]
 [ 28 110]]
0.9734513274336283
```

tfidfvectorizer

Above model performancne can be further optimized. lets use TFIDF with max_features is equal to 3000, a as it is working well here

In [173…
```
from sklearn.feature_extraction.text import TfidfVectorizer
#tfidf=TfidfVectorizer()
```

```
#now second time using max_features = 3000
tfidf=TfidfVectorizer(max_features=3000)
```

In [175…
```
tfidf.fit_transform(df['transformed_text']).toarray()
```

Out[175]:
```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

In [174…
```
X=tfidf.fit_transform(df['transformed_text']).toarray()
```

In [171…
```
X.shape
```

Out[171]:
```
(5169, 3000)
```

In [113…
```
y=df['target'].values
```

In [76]:
```
y
```

Out[76]:
```
array([0, 0, 1, ..., 0, 0, 0])
```

In [77]:
```
y.shape
```

Out[77]:
```
(5169,)
```

In [114…
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=.2,random_state=2)

from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

In [115…
```
from sklearn.naive_bayes import BernoulliNB,GaussianNB,MultinomialNB

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

In [116…
```
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))
```

```
0.8694390715667312
[[788 108]
 [ 27 111]]
0.5068493150684932
```

In [176…
```
mnb.fit(X_train,y_train)
y_pred2 = mnb.predict(X_test)
print(accuracy_score(y_test,y_pred2))
print(confusion_matrix(y_test,y_pred2))
print(precision_score(y_test,y_pred2))
```

```
0.9709864603481625
[[896   0]
 [ 30 108]]
1.0
```

Here false positive count is zero and model is behaving very good and precision is also 1

In [ ]:

In [118…
```
bnb.fit(X_train,y_train)
y_pred3 = bnb.predict(X_test)
print(accuracy_score(y_test,y_pred3))
print(confusion_matrix(y_test,y_pred3))
print(precision_score(y_test,y_pred3))
```

```
0.9835589941972921
[[895   1]
 [ 16 122]]
0.991869918699187
```

so in tfidf - we see MNB is peforming better - with false positive rate to 0 - very good precision

In [83]:
```
#other different model
```

In [86]:
```
!pip install xgboost
```

```
Collecting xgboost
  Downloading xgboost-1.7.4-py3-none-win_amd64.whl (89.1 MB)
Requirement already satisfied: scipy in c:\users\rupeshv\anaconda3\lib\site-packages
(from xgboost) (1.7.3)
Requirement already satisfied: numpy in c:\users\rupeshv\anaconda3\lib\site-packages
(from xgboost) (1.21.5)
Installing collected packages: xgboost
Successfully installed xgboost-1.7.4
Collecting xgboost
  Downloading xgboost-1.7.4-py3-none-win_amd64.whl (89.1 MB)
Requirement already satisfied: numpy in c:\users\rupeshv\anaconda3\lib\site-packages
(from xgboost) (1.21.5)
Requirement already satisfied: scipy in c:\users\rupeshv\anaconda3\lib\site-packages
(from xgboost) (1.7.3)
Installing collected packages: xgboost
Successfully installed xgboost-1.7.4
```

In [119…
```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
```

In [120…
```
svc = SVC(kernel='sigmoid', gamma=1.0)
knc = KNeighborsClassifier()
mnb = MultinomialNB()
dtc = DecisionTreeClassifier(max_depth=5)
lrc = LogisticRegression(solver='liblinear', penalty='l1')
rfc = RandomForestClassifier(n_estimators=50, random_state=2)
abc = AdaBoostClassifier(n_estimators=50, random_state=2)
bc = BaggingClassifier(n_estimators=50, random_state=2)
etc = ExtraTreesClassifier(n_estimators=50, random_state=2)
gbdt = GradientBoostingClassifier(n_estimators=50,random_state=2)
xgb = XGBClassifier(n_estimators=50,random_state=2)
```

In [121…
```
clfs = {
    'SVC' : svc,
    'KN' : knc,
    'NB': mnb,
    'DT': dtc,
    'LR': lrc,
    'RF': rfc,
    'AdaBoost': abc,
    'BgC': bc,
    'ETC': etc,
    'GBDT':gbdt,
    'xgb':xgb
}
```

In [122…
```
def train_classifier(clf,X_train,X_test,y_train,y_test):
    clf.fit(X_train,y_train)
    y_pred=clf.predict(X_test)
    accuracy=accuracy_score(y_test,y_pred)
    precision=precision_score(y_test,y_pred)

    return accuracy,precision
```

In [178…
```
train_classifier(mnb,X_train,X_test,y_train,y_test)
```

Out[178]:
```
(0.9709864603481625, 1.0)
```

In [124…
```
accuracy_scores = []
precision_scores = []

for name, clf in clfs.items():
    current_accuracy,current_precision=train_classifier(clf,X_train,X_test,y_train,y_t

    print("for ",name)
    print("Accuracy - ", current_accuracy)
    print("Precision -", current_precision)

    accuracy_scores.append(current_accuracy)
    precision_scores.append(current_precision)
```

```
for  SVC
Accuracy -  0.9758220502901354
Precision - 0.9747899159663865
for  KN
Accuracy -  0.9052224371373307
Precision - 1.0
for  NB
Accuracy -  0.9709864603481625
Precision - 1.0
for  DT
Accuracy -  0.9294003868471954
Precision - 0.8282828282828283
for  LR
Accuracy -  0.9584139264990329
Precision - 0.9702970297029703
for  RF
Accuracy -  0.9748549323017408
Precision - 0.9827586206896551
for  AdaBoost
Accuracy -  0.960348162475822
Precision - 0.9292035398230089
for  BgC
Accuracy -  0.9574468085106383
Precision - 0.8671875
for  ETC
Accuracy -  0.9748549323017408
Precision - 0.9745762711864406
for  GBDT
Accuracy -  0.9477756286266924
Precision - 0.92
for  xgb
Accuracy -  0.971953578336557
Precision - 0.943089430894309
```

In [95]:
```
import pandas as pd
```

In [100…
```
performance_df=pd.DataFrame({'Algorithms':clfs.keys(),'Accuracy':accuracy_scores,'Prec
```

In [105…
```
performance_df.reset_index(drop='index')
```

Out[105]:

	Algorithms	Accuracy	Precision
0	KN	0.900387	1.000000
1	NB	0.959381	1.000000
2	RF	0.973888	1.000000
3	ETC	0.975822	0.982906
4	SVC	0.972921	0.974138
5	AdaBoost	0.961315	0.945455
6	LR	0.951644	0.940000
7	xgb	0.969052	0.934426
8	GBDT	0.952611	0.923810
9	BgC	0.958414	0.862595
10	DT	0.935203	0.838095

In [107…
```
performance_df1 = pd.melt(performance_df, id_vars = "Algorithms")
```

In [108…
```
performance_df1
```

Out[108]:

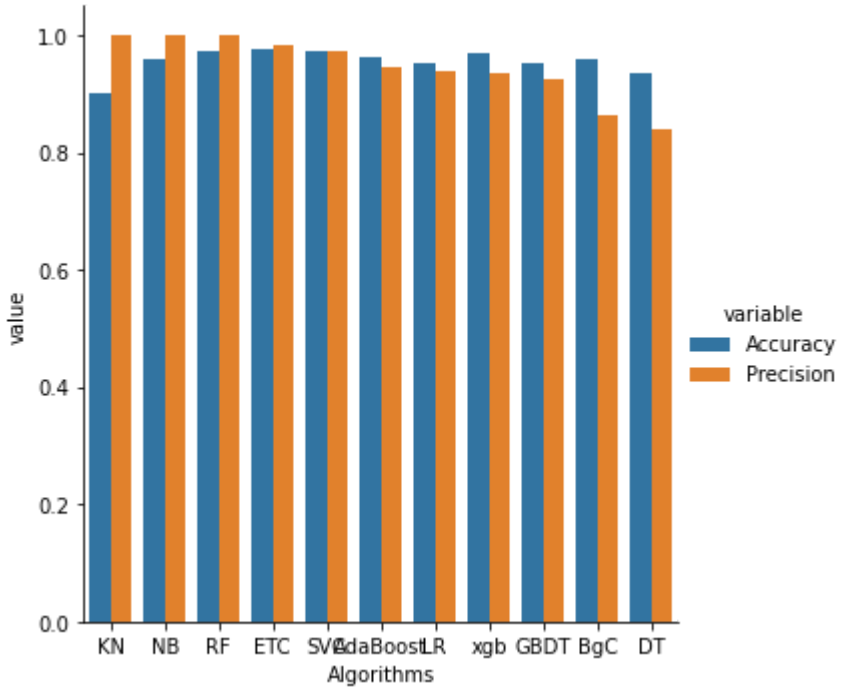
	Algorithms	variable	value
0	KN	Accuracy	0.900387
1	NB	Accuracy	0.959381
2	RF	Accuracy	0.973888
3	ETC	Accuracy	0.975822
4	SVC	Accuracy	0.972921
5	AdaBoost	Accuracy	0.961315
6	LR	Accuracy	0.951644
7	xgb	Accuracy	0.969052
8	GBDT	Accuracy	0.952611
9	BgC	Accuracy	0.958414
10	DT	Accuracy	0.935203
11	KN	Precision	1.000000
12	NB	Precision	1.000000
13	RF	Precision	1.000000
14	ETC	Precision	0.982906
15	SVC	Precision	0.974138
16	AdaBoost	Precision	0.945455
17	LR	Precision	0.940000
18	xgb	Precision	0.934426
19	GBDT	Precision	0.923810
20	BgC	Precision	0.862595
21	DT	Precision	0.838095

In [109…

```
sns.catplot(x='Algorithms',y='value',data=performance_df1,kind='bar',hue='variable')
```

Out[109]:

```
<seaborn.axisgrid.FacetGrid at 0x28d86411430>
```



change the max_features parameters of tfidf

In [137…
```
temp_df = pd.DataFrame({'Algorithms':clfs.keys(),'Accuracy_max_ft_3000':accuracy_score
temp_df = pd.DataFrame({'Algorithms':clfs.keys(),'Accuracy_scaling':accuracy_scores,'P
```

In [138…
```
temp_df
```

Out[138]:

	Algorithms	Accuracy_scaling	Precision_scaling
1	KN	0.905222	1.000000
2	NB	0.970986	1.000000
5	RF	0.974855	0.982759
0	SVC	0.975822	0.974790
8	ETC	0.974855	0.974576
4	LR	0.958414	0.970297
10	xgb	0.971954	0.943089
6	AdaBoost	0.960348	0.929204
9	GBDT	0.947776	0.920000
7	BgC	0.957447	0.867188
3	DT	0.929400	0.828283

In [131…
```
new_df = performance_df.merge(temp_df,on='Algorithms')
```

In [132…
```
new_df
```

Out[132]:

	Algorithms	Accuracy	Precision	Accuracy_max_ft_3000	Precision_max_ft_3000
0	KN	0.900387	1.000000	0.905222	1.000000
1	NB	0.959381	1.000000	0.970986	1.000000
2	RF	0.973888	1.000000	0.974855	0.982759
3	ETC	0.975822	0.982906	0.974855	0.974576
4	SVC	0.972921	0.974138	0.975822	0.974790
5	AdaBoost	0.961315	0.945455	0.960348	0.929204
6	LR	0.951644	0.940000	0.958414	0.970297
7	xgb	0.969052	0.934426	0.971954	0.943089
8	GBDT	0.952611	0.923810	0.947776	0.920000
9	BgC	0.958414	0.862595	0.957447	0.867188
10	DT	0.935203	0.838095	0.929400	0.828283

In [139…

```
new_df_scaled = new_df.merge(temp_df,on='Algorithms')
```

In [158…

```
new_df
```

Out[158]:

	Algorithms	Accuracy	Precision	Accuracy_max_ft_3000	Precision_max_ft_3000
0	KN	0.900387	1.000000	0.905222	1.000000
1	NB	0.959381	1.000000	0.970986	1.000000
2	RF	0.973888	1.000000	0.974855	0.982759
3	ETC	0.975822	0.982906	0.974855	0.974576
4	SVC	0.972921	0.974138	0.975822	0.974790
5	AdaBoost	0.961315	0.945455	0.960348	0.929204
6	LR	0.951644	0.940000	0.958414	0.970297
7	xgb	0.969052	0.934426	0.971954	0.943089
8	GBDT	0.952611	0.923810	0.947776	0.920000
9	BgC	0.958414	0.862595	0.957447	0.867188
10	DT	0.935203	0.838095	0.929400	0.828283

In [141…

```
# Voting Classifier combine multiple algo
svc = SVC(kernel='sigmoid', gamma=1.0,probability=True)
mnb = MultinomialNB()
etc = ExtraTreesClassifier(n_estimators=50, random_state=2)

from sklearn.ensemble import VotingClassifier
```

In [142…

```
voting = VotingClassifier(estimators=[('svm', svc), ('nb', mnb), ('et', etc)],voting='
```

In [159…

```
voting.fit(X_train,y_train)
```

Out[159]:
```
VotingClassifier(estimators=[('svm',
                              SVC(gamma=1.0, kernel='sigmoid',
                                  probability=True)),
                             ('nb', MultinomialNB()),
                             ('et',
                              ExtraTreesClassifier(n_estimators=50,
                                                   random_state=2))],
                 voting='soft')
```

In [144…
```
y_pred = voting.predict(X_test)
print("Accuracy",accuracy_score(y_test,y_pred))
print("Precision",precision_score(y_test,y_pred))
```

```
Accuracy 0.9816247582205029
Precision 0.9917355371900827
```

We have tried lot of classifer. but we can say MNB is performing better. so we will save the model here. so that there is no requirement to run it again and again

we have used joblib and pickle. But joblib is easy to work as it is not involving file handling part.

This step was used to save the model and will used in deployment.

In [177…
```
import joblib

joblib.dump(mnb, 'model1.pkl')
joblib.dump(tfidf,'vectorizer1.pkl')
```

Out[177]:
```
['vectorizer1.pkl']
```

In [162…
```
import pickle
pickle.dump(tfidf,open('vectorizer.pkl','wb'))
pickle.dump(mnb,open('model.pkl','wb'))
```

In [160…
```
pd.read_pickle('model.pkl')
```

Out[160]:
```
MultinomialNB()
```

In [149…
```
pd.read_pickle('vectorizer.pkl')
```

Out[149]:
```
TfidfVectorizer(max_features=3000)
```

In [148…
```
pd.read_pickle('nitish_trainer_files/model.pkl')
```

Out[148]:
```
MultinomialNB()
```

In [157…
```
import sklearn
print('The scikit-learn version is {}.'.format(sklearn.__version__))
```

```
The scikit-learn version is 1.0.2.
```

Our jupyter notebook version will match with our project venv version in pycharm

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: