Final year project

Spam detection

Step:-

- Data cleaning
- EDA
- Text preprocessing
- Model building
- Evaluation
- Improvement
- website
- deploy

In [1]:
```
#Loading required libraries
import pandas as pd
import numpy as np
```

In [2]:
```
#Loading the Ham/Spam data set
data=pd.read_csv("spam.csv", encoding="latin-1")
```

In [3]:
```
#Having a glance at the first five records of the data set
data.head()
```

Out[3]:

	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 [4]:
```
data.shape
```

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

Data Cleaning

In [5]:
```
data.columns
```

Out[5]:
```
Index(['v1', 'v2', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], dtype='objec
t')
```

In [6]:
```
# drop last 3 cols
data.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)
```

In [7]:
```
data.sample(5)
```

Out[7]:

	v1	v2
4435	ham	House-Maid is the murderer, coz the man was mu...
1972	ham	Yes but can we meet in town cos will go to gep...
5417	ham	Nope. I just forgot. Will show next week
2467	ham	Is there coming friday is leave for pongal?do ...
3238	ham	Am okay. Will soon be over. All the best

In [8]:
```
# Renaming the cols
data.rename(columns={'v1':'target','v2':'text'},inplace=True)
data.sample(5)
```

Out[8]:

	target	text
1091	ham	Please da call me any mistake from my side sor...
2880	ham	Printer is cool. I mean groovy. Wine is groovying
1658	spam	RGENT! This is the 2nd attempt to contact U!U ...
332	spam	Call Germany for only 1 pence per minute! Call...
2875	ham	Fuck cedar key and fuck her (come over anyway ...

In [9]:
```
#Level encoder use
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

In [10]:
```
data['target']=encoder.fit_transform(data['target'])
```

In [11]:
```
data.head()
```

Out[11]:

	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 [12]:
```
# missing values
data.isnull().sum()
```

Out[12]:
```
target    0
text      0
dtype: int64
```

In [13]:
```
# check for duplicte values
data.duplicated().sum()
```

Out[13]:
```
403
```

In [14]:
```
# remove duplicates
data=data.drop_duplicates(keep='first')
```

In [15]:
```
data.duplicated().sum()
```

Out[15]:
```
0
```

In [16]:
```
data.shape
```

Out[16]:
```
(5169, 2)
```

2.EDA

In [17]:
```
data.head()
```

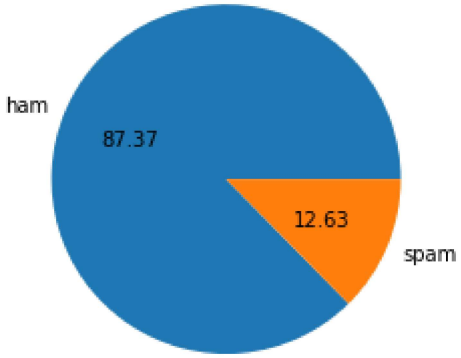
Out[17]:

	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 [18]:
```
data['target'].value_counts()
```

Out[18]:
```
0    4516
1     653
Name: target, dtype: int64
```

In [19]:
```
import matplotlib.pyplot as plt
plt.pie(data["target"].value_counts(),labels=['ham','spam'],autopct="%0.2f")
plt.show()
```



In [20]:
```
# Data is imbalanced
```

In [21]:
```
import nltk
```

In [22]:
```
nltk.download("punkt")
```

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

Out[22]: True

In [23]:
```
data['num_characters']=data['text'].apply(len)
```

In [24]:
```
data.head()
```

Out[24]:

	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 [25]:
```
# num of words
data['num_words']=data['text'].apply(lambda x:len(nltk.word_tokenize(x)))
```

In [26]:
```
data.head()
```

Out[26]:

	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 [27]:
```
data['num_sentences']=data['text'].apply(lambda x:len(nltk.sent_tokenize(x)))
```

In [28]:
```
data.head()
```

Out[28]:

	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 [29]:
```
data[['num_characters','num_words','num_sentences']].describe()
```

Out[29]:

	num_characters	num_words	num_sentences
count	5169.000000	5169.000000	5169.000000
mean	78.977945	18.455407	1.961308
std	58.236293	13.322448	1.432583
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	38.000000

In [30]:
```
data[['num_characters','num_words','num_sentences']].describe()
```

Out[30]:

	num_characters	num_words	num_sentences
count	5169.000000	5169.000000	5169.000000
mean	78.977945	18.455407	1.961308
std	58.236293	13.322448	1.432583
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	38.000000

In [31]:
```
#spam
data[data['target']==1][['num_characters','num_words','num_sentences']].descr
```

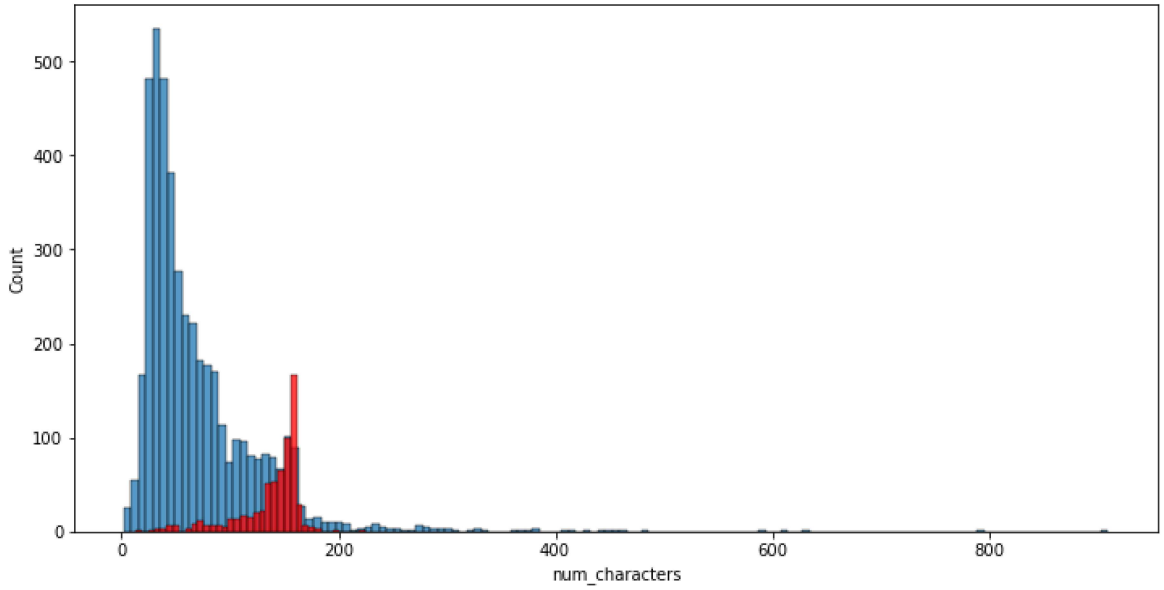
Out[31]:

	num_characters	num_words	num_sentences
count	653.000000	653.000000	653.000000
mean	137.891271	27.667688	2.969372
std	30.137753	7.008418	1.488910
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	9.000000

In [32]:
```
import seaborn as sns
```

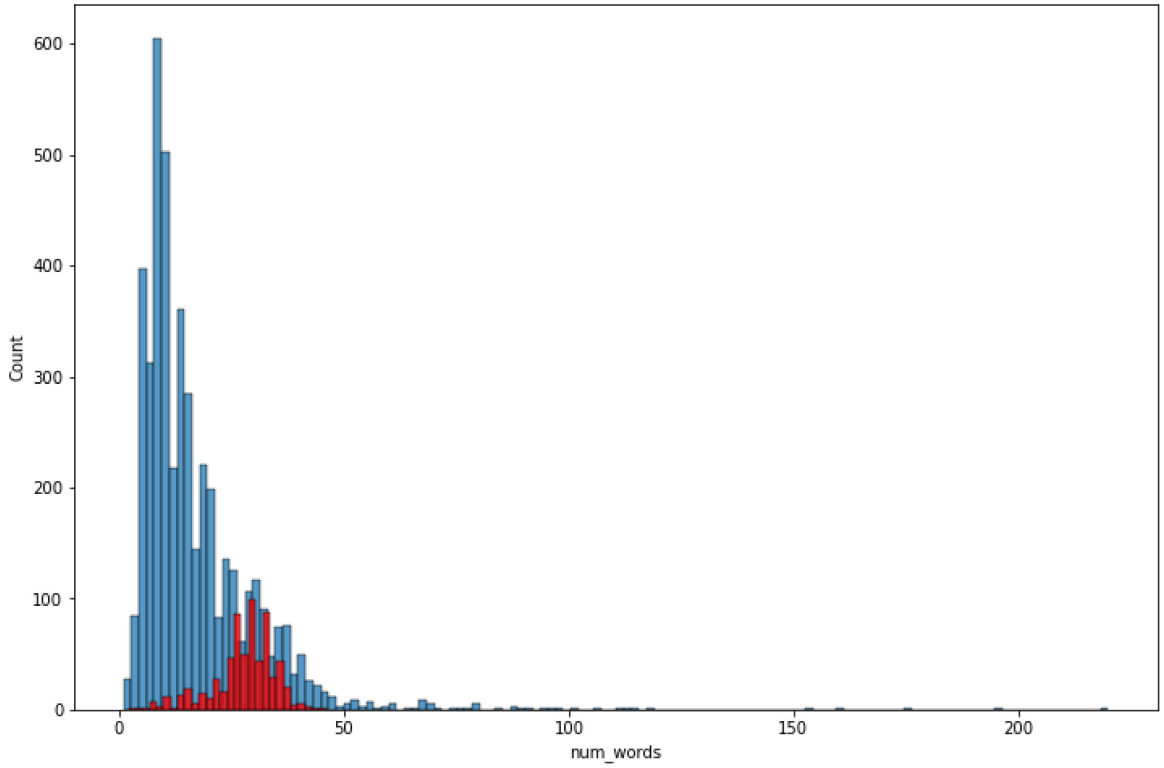
In [33]:
```
plt.figure(figsize=(12,6))
sns.histplot(data[data['target']==0]['num_characters'])
sns.histplot(data[data['target']==1]['num_characters'],color='red')
```

Out[33]: <AxesSubplot:xlabel='num_characters', ylabel='Count'>



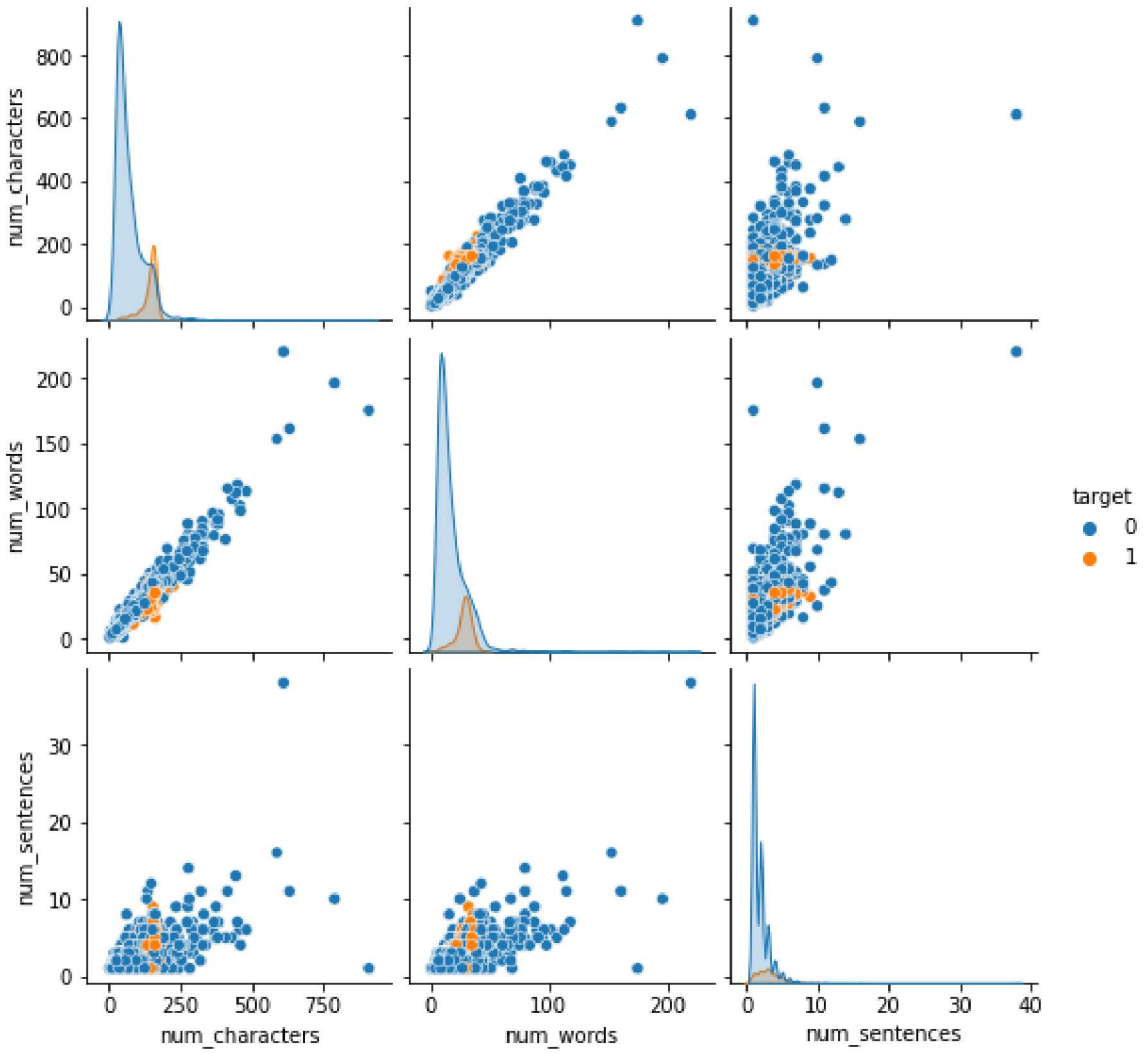
In [34]:
```
plt.figure(figsize=(12,8))
sns.histplot(data[data['target']==0]['num_words'])
sns.histplot(data[data['target']==1]['num_words'],color='red')
```

Out[34]: <AxesSubplot:xlabel='num_words', ylabel='Count'>



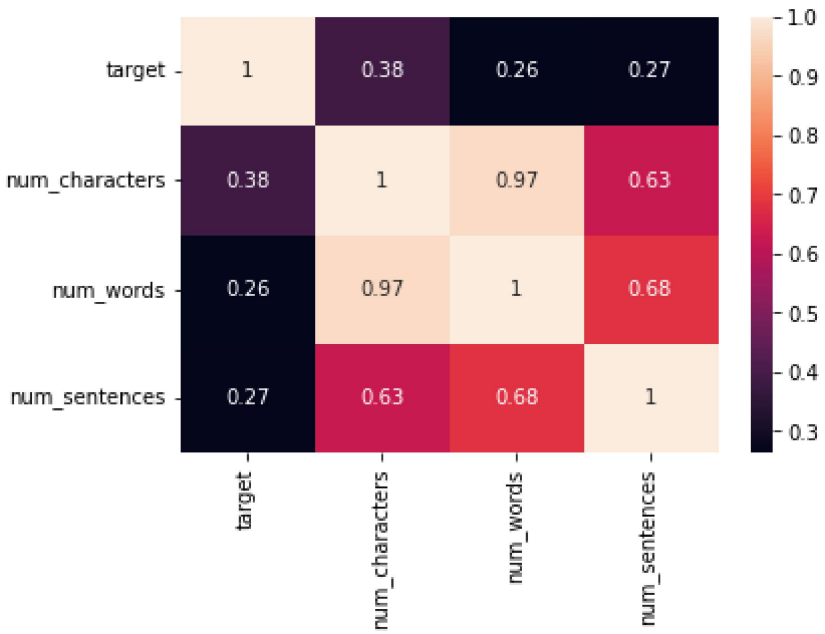
In [35]:
```
sns.pairplot(data,hue='target')
```

Out[35]: <seaborn.axisgrid.PairGrid at 0x278291eed60>



In [36]:
```
sns.heatmap(data.corr(),annot=True)
```

Out[36]: <AxesSubplot:>



3.Data Preprocessing

- Lower case
- Tokenization
- Removing special charecters
- Removing stop words and punctuation
- Stemming

In [37]:
```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
ps.stem('dancing')
```

Out[37]: 'danc'

In [38]:
```
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.punctuatic
            y.append(i)
    text = y[:]
    y.clear()

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

    return " ".join(y)
```

In [39]:
```
from nltk.corpus import stopwords
stopwords.words('english')
```

Out[39]:
```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
```

In [40]:
```
import string
string.punctuation
```

Out[40]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [41]:
```
transform_text('Go until jurong point, crazy.. Available only in bugis n grea
```

Out[41]:
```
'go jurong point crazi avail bugi n great world la e buffet cine got amor w
at'
```

In [42]:
```
data['text'][10]
```

Out[42]:
```
"I'm gonna be home soon and i don't want to talk about this stuff anymore t
onight, k? I've cried enough today."
```

In [43]:
```
data['transformed_text']=data['text'].apply(transform_text)
```

In [44]:
```
data.head()
```

Out[44]:

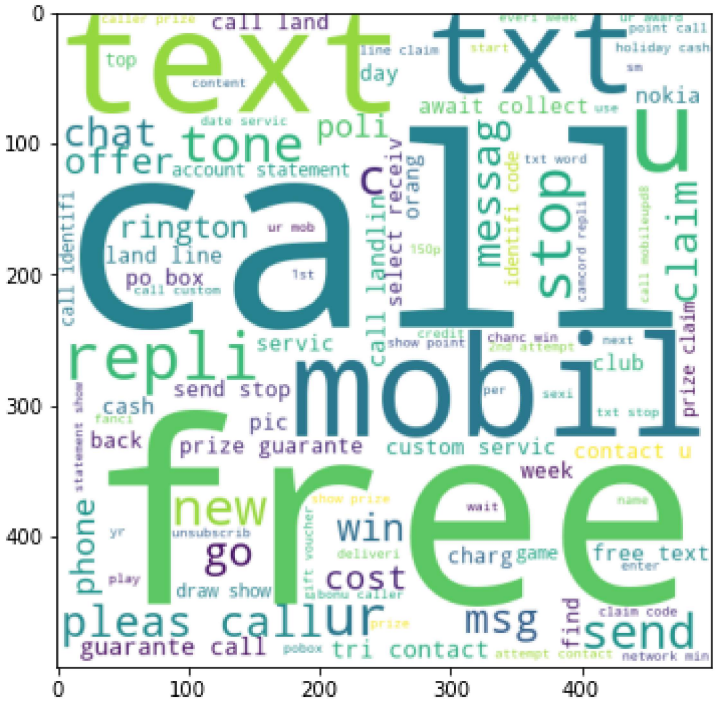
	target	text	num_characters	num_words	num_sentences	transformed_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	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 [45]:
```
from wordcloud import WordCloud
wc = WordCloud(width=500,height=500,min_font_size=10,background_color='white'
```

In [46]:
```
spam_wc = wc.generate(data[data['target']== 1]['transformed_text'].str.cat(se
```

In [47]:
```
plt.figure(figsize=(15,6))
plt.imshow(spam_wc)
```

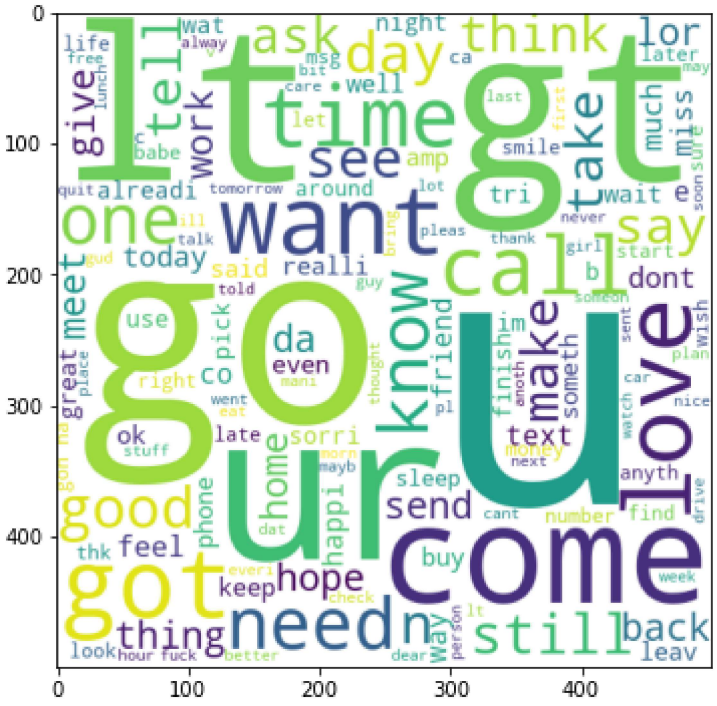
Out[47]: <matplotlib.figure.AxesImage at 0x2782ce0e310>



In [48]:
```
ham_wc = wc.generate(data[data['target']== 0]['transformed_text'].str.cat(sep
```

In [49]:
```
plt.figure(figsize=(15,6))
plt.imshow(ham_wc)
```

Out[49]: <matplotlib.image.AxesImage at 0x2782ce34400>



In [50]:
```
data.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

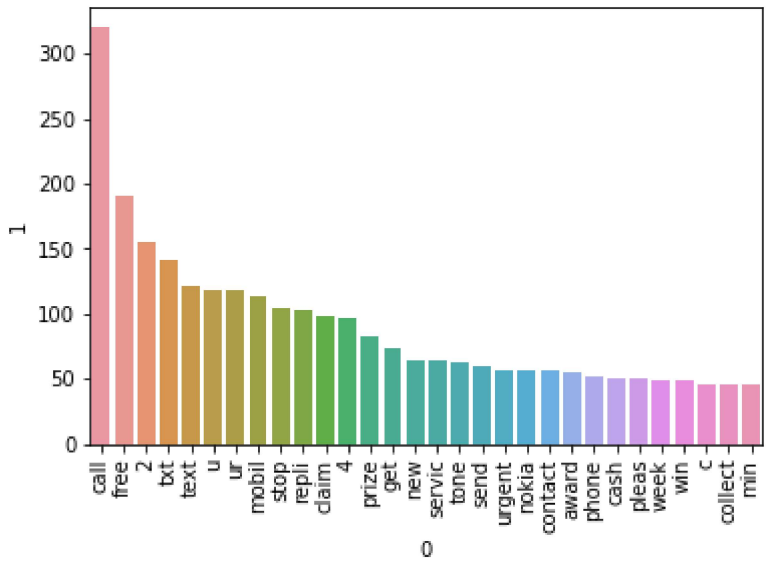
In [51]:
```
spam_corpus = []
for msg in data[data['target']==1]['transformed_text'].tolist():
    for word in msg.split():
        spam_corpus.append(word)
```

In [52]:
```
len(spam_corpus)
```

Out[52]: 9939

In [53]:
```
from collections import Counter
sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0],pd.DataFram
plt.xticks(rotation='vertical')
plt.show()
```

```
C:\Users\Mohini Tyagi\anaconda3\lib\site-packages\seaborn\_decorators.py:3
6: FutureWarning: Pass the following variables as keyword args: x, y. From
version 0.12, the only valid positional argument will be `data`, and passin
g other arguments without an explicit keyword will result in an error or mi
sinterpretation.
  warnings.warn(
```



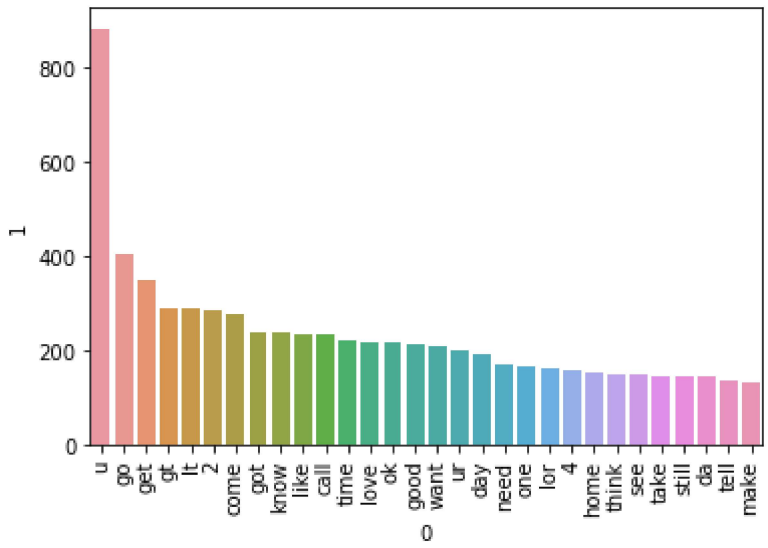
In [54]:
```
ham_corpus = []
for msg in data[data['target']==0]['transformed_text'].tolist():
    for word in msg.split():
        ham_corpus.append(word)
```

In [55]:
```
len(ham_corpus)
```

Out[55]: 35402

In [56]:
```
from collections import Counter
sns.barplot(pd.DataFrame(Counter(ham_corpus).most_common(30))[0],pd.DataFrame
plt.xticks(rotation='vertical')
plt.show()
```

```
C:\Users\Mohini Tyagi\anaconda3\lib\site-packages\seaborn\_decorators.py:3
6: FutureWarning: Pass the following variables as keyword args: x, y. From
version 0.12, the only valid positional argument will be `data`, and passin
g other arguments without an explicit keyword will result in an error or mi
sinterpretation.
  warnings.warn(
```



In [57]:
```
# Text Vectorization
# using Bag of Words
data.head()
```

Out[57]:

	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

4. Model Building

In [58]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

In [59]:
```
from sklearn.model_selection import train_test_split
```

In [60]:
```
X=data['text']
y=data['target']
```

In [61]:
```
X.shape
```

Out[61]:
```
(5169,)
```

In [62]:
```
y.shape
```

Out[62]:
```
(5169,)
```

In [63]:
```
data.isnull().sum()
```

Out[63]:
```
target              0
text                0
num_characters      0
num_words           0
num_sentences       0
transformed_text    0
dtype: int64
```

In [64]:
```
cv=CountVectorizer()
```

In [65]:
```
X=cv.fit_transform(X)
```

In [66]:
```
x_train, x_test,y_train, y_test=train_test_split(X,y, test_size=0.2, random_s
```

In [67]:
```
x_train.shape
```

Out[67]:
```
(4135, 8672)
```

In [68]:
```
x_test.shape
```

Out[68]:
```
(1034, 8672)
```

In [69]:
```
from sklearn.naive_bayes import MultinomialNB
```

In [70]:
```
model=MultinomialNB()
```

In [71]:
```
model.fit(x_train, y_train)
```

Out[71]:
```
MultinomialNB()
```

In [72]:
```
model.score(x_test, y_test)
```

Out[72]:
```
0.9825918762088974
```

In [73]:
```
msg="You Won 500$"
data = [msg]
vect = cv.transform(data).toarray()
my_prediction = model.predict(vect)
```

In [74]:
```
vect
```

Out[74]:
```
array([[0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

In [75]:
```
import pickle
pickle.dump(model, open('spam.pkl','wb'))
model1 = pickle.load(open('spam.pkl','rb'))
```

In [ ]:
```

```

In [76]:
```
from win32com.client import Dispatch
```

In [77]:
```
def speak(text):
    speak=Dispatch(("SAPI.SpVoice"))
    speak.Speak(text)
```

In [78]:
```
def result(msg):
    data = [msg]
    vect = cv.transform(data).toarray()
    my_prediction = model1.predict(vect)
    if my_prediction[0]==1:
        speak("This is a Spam mail")
        print("This is a Spam mail")
    else:
        speak("This is not a Spam mail")
        print("This is not a Spam mail")
```

In [79]:
```
import tkinter as tk
```

In [ ]:

```
root=tk.Tk()
root.geometry("200x200")
l2=tk.Label(root, text="Email Spam Classification Application")
l2.pack()
l1=tk.Label(root, text="Enter Your Message:")
l1.pack()
text=tk.Entry(root)
text.pack()
def result():
    data = [text.get()]
    vect = cv.transform(data).toarray()
    my_prediction = model1.predict(vect)
    if my_prediction[0]==1:
        speak("This is a Spam mail")
        print("This is a Spam mail")
    else:
        speak("This is not a Spam mail")
        print("This is not a Spam mail")
B=tk.Button(root, text="Click", command=result)
B.pack()

root.mainloop()
```

```
This is not a Spam mail
```

In [ ]:

In [ ]: