In [1]:

```
import numpy as np
import pandas as pd
```

```
C:\Users\sanja\anaconda3\lib\site-packages\pandas\core\computation\expres
sions.py:21: UserWarning: Pandas requires version '2.8.0' or newer of 'nu
mexpr' (version '2.7.3' currently installed).
  from pandas.core.computation.check import NUMEXPR_INSTALLED
C:\Users\sanja\anaconda3\lib\site-packages\pandas\core\arrays\masked.py:6
2: UserWarning: Pandas requires version '1.3.4' or newer of 'bottleneck'
(version '1.3.2' currently installed).
  from pandas.core import (
```

In [2]:

```
df = pd.read_csv('spam.csv', encoding='latin-1')
```

In [3]:

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

Out[3]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
899	spam	Your free ringtone is waiting to be collected....	PO Box 5249	MK17 92H. 450Ppw 16"	NaN
597	spam	You have an important customer service announc...	NaN	NaN	NaN
2259	ham	Ill call you evening ill some ideas.	NaN	NaN	NaN
3755	ham	Yes:)here tv is always available in work place..	NaN	NaN	NaN
3374	ham	:)	NaN	NaN	NaN

In [4]:

```
df.shape
```

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
```

In [6]:

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

In [7]:

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

Out[7]:

	v1	v2
4635	ham	K k pa Had your lunch aha.
4797	ham	Just come home. I don't want u to be miserable
1954	ham	Good night. Am going to sleep.
2379	ham	Good evening Sir, hope you are having a nice d...
2950	ham	Hey now am free you can call me.

In [8]:

```
df.rename(columns={'v1':'target','v2':'text'},inplace=True)
df.sample(5)
```

Out[8]:

	target	text
3582	ham	I sent your maga that money yesterday oh.
1655	ham	At 7 we will go ok na.
1760	ham	Nt yet chikku..simple habba..hw abt u?
4	ham	Nah I don't think he goes to usf, he lives aro...
3287	spam	Someone U know has asked our dating service 2 ...

In [9]:

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

In [10]:

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

In [11]:

```
df.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]:

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

Out[12]:

```
target    0
text      0
dtype: int64
```

In [13]:

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

Out[13]:

```
403
```

In [14]:

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

In [15]:

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

Out[15]:

```
0
```

In [16]:

```
df.shape
```

Out[16]:

```
(5169, 2)
```

In [17]:

```
df.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]:

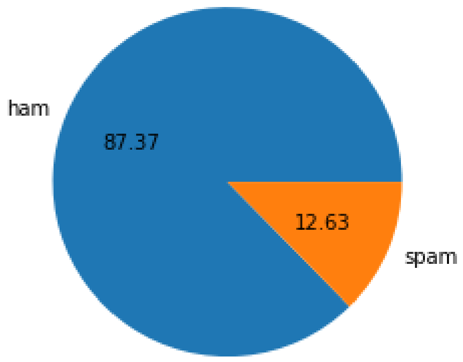
```
df['target'].value_counts()
```

Out[18]:

```
target
0    4516
1     653
Name: count, dtype: int64
```

In [19]:

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



In [20]:

```
import nltk
!pip install nltk
nltk.download('punkt')
```

```
Requirement already satisfied: nltk in c:\users\sanja\anaconda3\lib\site-
packages (3.6.5)
Requirement already satisfied: click in c:\users\sanja\anaconda3\lib\site
-packages (from nltk) (8.0.3)
Requirement already satisfied: joblib in c:\users\sanja\anaconda3\lib\sit
e-packages (from nltk) (1.1.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\sanja\anaconda
3\lib\site-packages (from nltk) (2021.8.3)
Requirement already satisfied: tqdm in c:\users\sanja\anaconda3\lib\site-
packages (from nltk) (4.62.3)
Requirement already satisfied: colorama in c:\users\sanja\anaconda3\lib\s
ite-packages (from click->nltk) (0.4.6)

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

Out[20]:

```
True
```

In [21]:

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

In [22]:

```
df.head()
```

Out[22]:

	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 [23]:

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

In [24]:

```
df.head()
```

Out[24]:

	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 [25]:

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

In [26]:

```
df.head()
```

Out[26]:

	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 [27]:

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

Out[27]:

	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 [28]:

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

Out[28]:

	num_characters	num_words	num_sentences
count	4516.000000	4516.000000	4516.000000
mean	70.459256	17.123339	1.815545
std	56.358207	13.491315	1.364098
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	38.000000

In [29]:

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

Out[29]:

	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

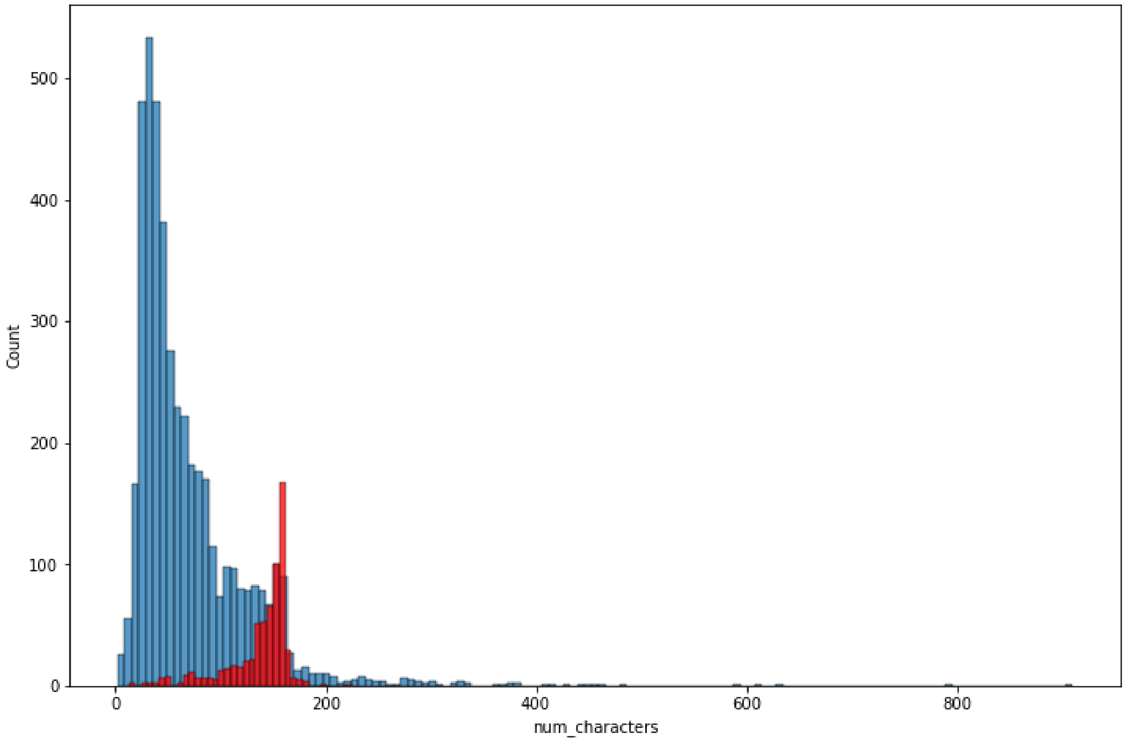
In [30]:

```
import seaborn as sns
plt.figure(figsize=(12,8))
sns.histplot(df[df['target'] == 0]['num_characters'])
sns.histplot(df[df['target'] == 1]['num_characters'],color='red')
```

```
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119: Futu
reWarning: use_inf_as_na option is deprecated and will be removed in a fu
ture version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119: Futu
reWarning: use_inf_as_na option is deprecated and will be removed in a fu
ture version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[30]:

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



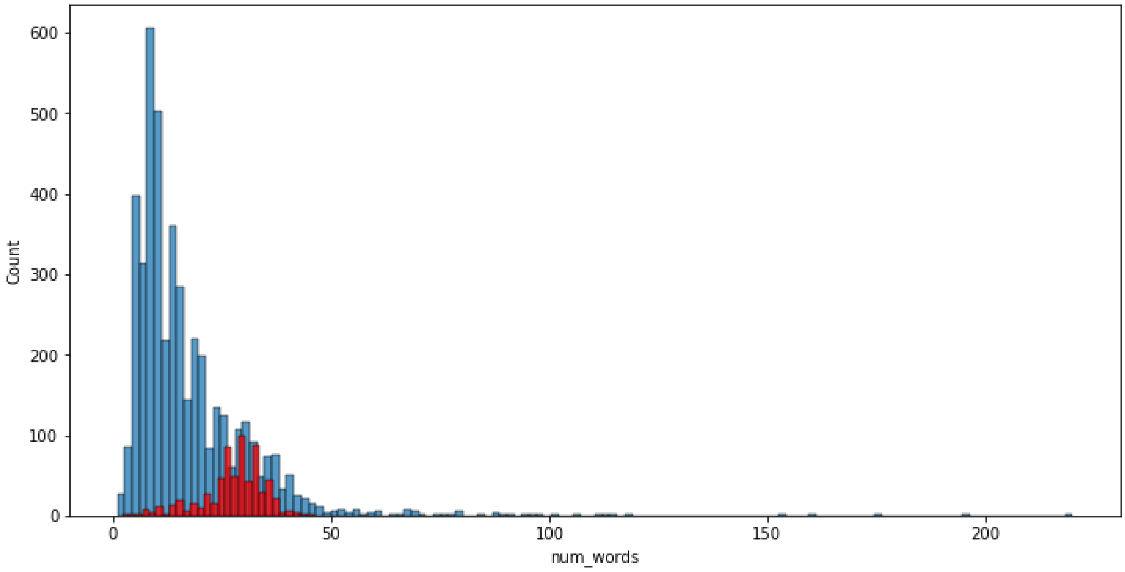
In [31]:

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

```
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119: Futu
reWarning: use_inf_as_na option is deprecated and will be removed in a fu
ture version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119: Futu
reWarning: use_inf_as_na option is deprecated and will be removed in a fu
ture version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[31]:

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



In [32]:

```
sns.pairplot(df,hue='target')
```

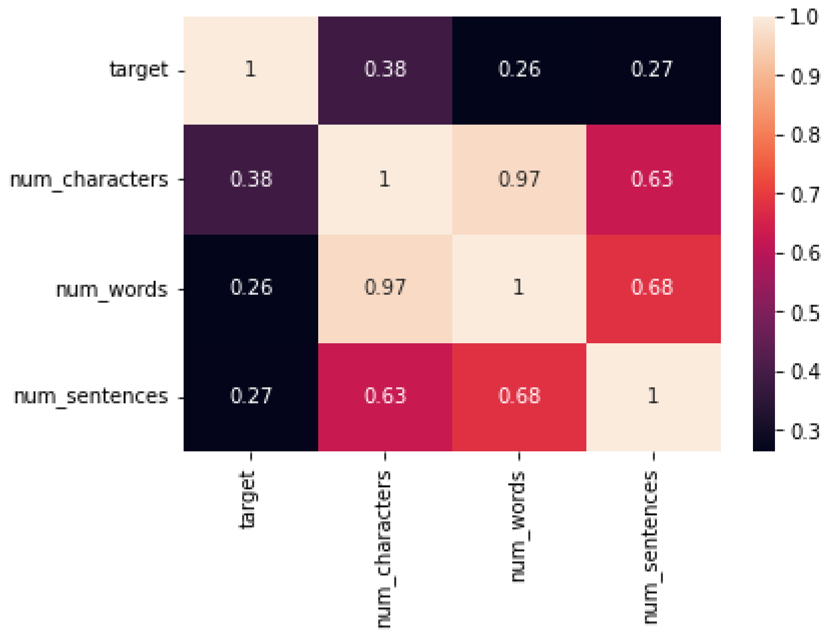
```
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: F
utureWarning: is_categorical_dtype is deprecated and will be removed i
n a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: F
utureWarning: is_categorical_dtype is deprecated and will be removed i
n a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: F
utureWarning: is_categorical_dtype is deprecated and will be removed i
n a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: F
utureWarning: is_categorical_dtype is deprecated and will be removed i
n a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: F
utureWarning: is_categorical_dtype is deprecated and will be removed i
n a future version. Use isinstance(dtype, CategoricalDtype) instead
```

In [33]:

```
df1=df.drop(['text'],axis=1)
sns.heatmap(df1.corr(),annot=True)
```

Out[33]:

```
<AxesSubplot:>
```



In [34]:

```
import string
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer


nltk.download('stopwords')
nltk.download('punkt')

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

In [35]:

```
df['text'][10]
```

Out[35]:

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

In [36]:

```
transform_text("I'm gonna be home soon and i don't want to talk about this stuff anymore
```

Out[36]:

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

In [37]:

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

Out[37]:

```
'love'
```

In [38]:

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

In [39]:

```
df.head()
```

Out[39]:

	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 [ ]:

```

```

In [40]:

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

In [41]:

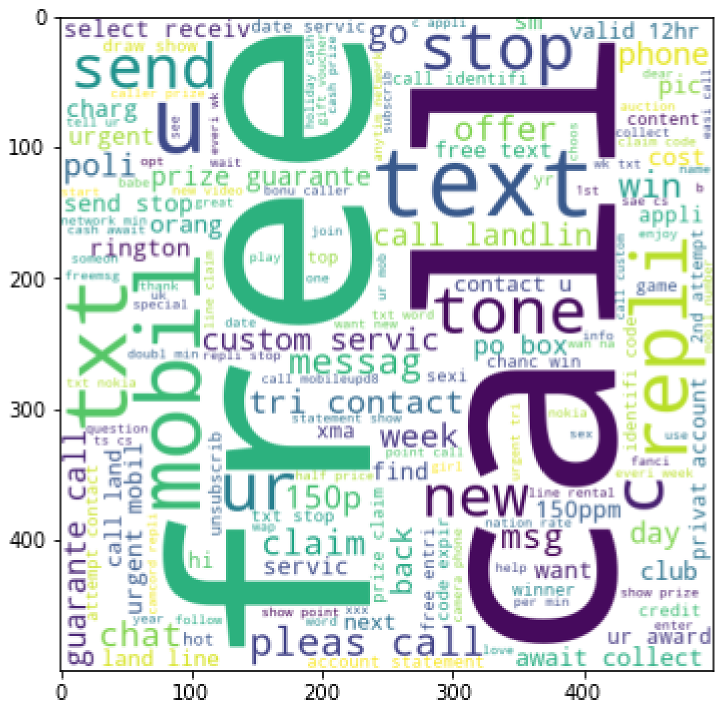
```
spam_wc = wc.generate(df[df['target'] == 1]['transformed_text'].str.cat(sep=" "))
spam_wc = wc.recolor(colormap='viridis', random_state=42)
```

In [42]:

```
plt.figure(figsize=(15,6))
plt.imshow(spam_wc)
```

Out[42]:

```
<matplotlib.image.AxesImage at 0x177264e8280>
```

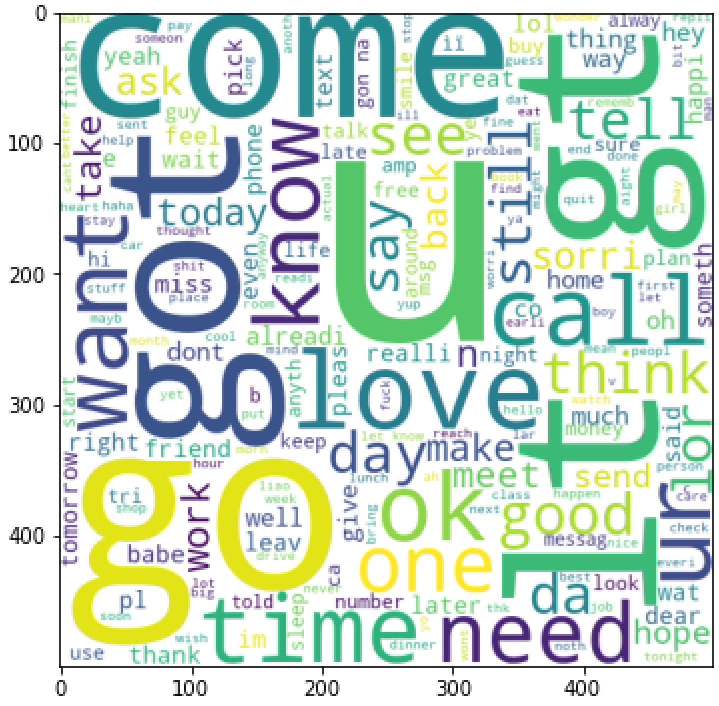


In [43]:

```
ham_wc = wc.generate(df[df['target'] == 0]['transformed_text'].str.cat(sep=" "))
plt.figure(figsize=(15,6))
plt.imshow(ham_wc)
```

Out[43]:

```
<matplotlib.image.AxesImage at 0x177269e6ac0>
```



In [44]:

```
df.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]:

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

In [46]:

```
len(spam_corpus)
```

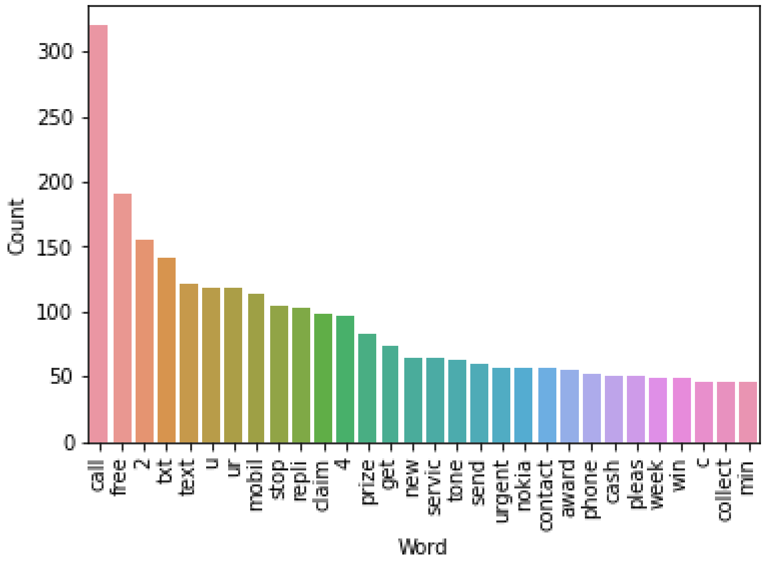
Out[46]:

```
9939
```

In [47]:

```
from collections import Counter
word_counts = Counter(spam_corpus)
common_words_df = pd.DataFrame(word_counts.most_common(30), columns=['Word', 'Count'])
sns.barplot(x='Word', y='Count', data=common_words_df)
plt.xticks(rotation='vertical')
plt.show()
```

```
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



In [48]:

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

In [49]:

```
len(ham_corpus)
```

Out[49]:

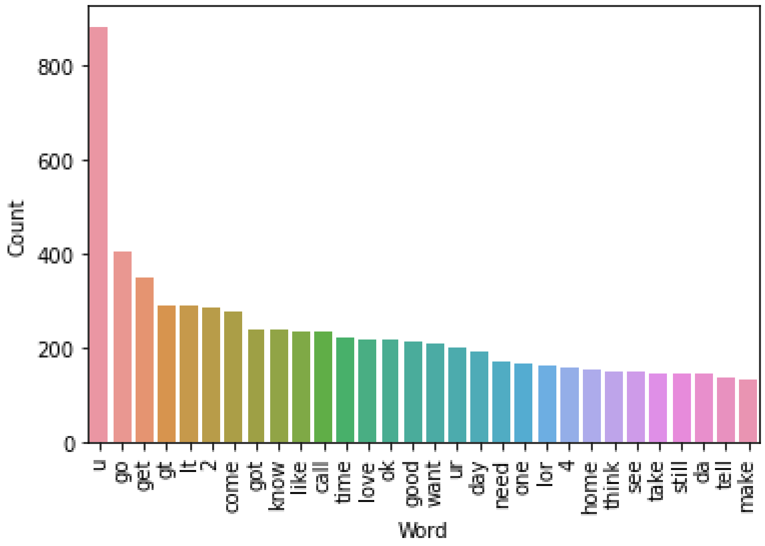
```
35402
```

In [50]:

```
from collections import Counter

word_counts = Counter(ham_corpus)
common_words_df = pd.DataFrame(word_counts.most_common(30), columns=['Word', 'Count'])
sns.barplot(x='Word', y='Count', data=common_words_df)
plt.xticks(rotation='vertical')
plt.show()
```

```
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



In [51]:

```
df.head()
```

Out[51]:

	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 [52]:

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

In [53]:

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

In [54]:

```
X.shape
```

Out[54]:

```
(5169, 3000)
```

In [55]:

```
y = df['target'].values
```

In [56]:

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

In [57]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

In [58]:

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

In [59]:

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

In [60]:

```
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 [61]:

```
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
```

In [62]:

```
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
```

In [63]:

```
pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\sanja\anaconda3\lib\si
te-packages (1.7.6)
Requirement already satisfied: numpy in c:\users\sanja\anaconda3\lib\site
-packages (from xgboost) (1.22.4)
Requirement already satisfied: scipy in c:\users\sanja\anaconda3\lib\site
-packages (from xgboost) (1.7.1)
Note: you may need to restart the kernel to use updated packages.
```

In [64]:

```
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 [65]:

```
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 [66]:

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

In [67]:

```
def train_classifier(clf,X_train,y_train,X_test,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 [68]:

```
train_classifier(svc,X_train,y_train,X_test,y_test)
```

Out[68]:

```
(0.9758220502901354, 0.9747899159663865)
```

In [69]:

```
accuracy_scores = []
precision_scores = []

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

    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 -  0.9284332688588007
Precision -  0.82
For  LR
Accuracy -  0.9584139264990329
Precision -  0.9702970297029703
For  RF
Accuracy -  0.9758220502901354
Precision -  0.9829059829059829
For  AdaBoost
Accuracy -  0.960348162475822
Precision -  0.9292035398230089
For  BgC
Accuracy -  0.9584139264990329
Precision -  0.8682170542635659
For  ETC
Accuracy -  0.9748549323017408
Precision -  0.9745762711864406
For  GBDT
Accuracy -  0.9468085106382979
Precision -  0.9191919191919192
For  xgb
Accuracy -  0.9671179883945842
Precision -  0.9333333333333333
```

In [70]:

```
performance_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy':accuracy_scores,'Preci
performance_df
```

Out[70]:

	Algorithm	Accuracy	Precision
1	KN	0.905222	1.000000
2	NB	0.970986	1.000000
5	RF	0.975822	0.982906
0	SVC	0.975822	0.974790
8	ETC	0.974855	0.974576
4	LR	0.958414	0.970297
10	xgb	0.967118	0.933333
6	AdaBoost	0.960348	0.929204
9	GBDT	0.946809	0.919192
7	BgC	0.958414	0.868217
3	DT	0.928433	0.820000

In [71]:

```
performance_df1 = pd.melt(performance_df, id_vars = "Algorithm")
performance_df1
```

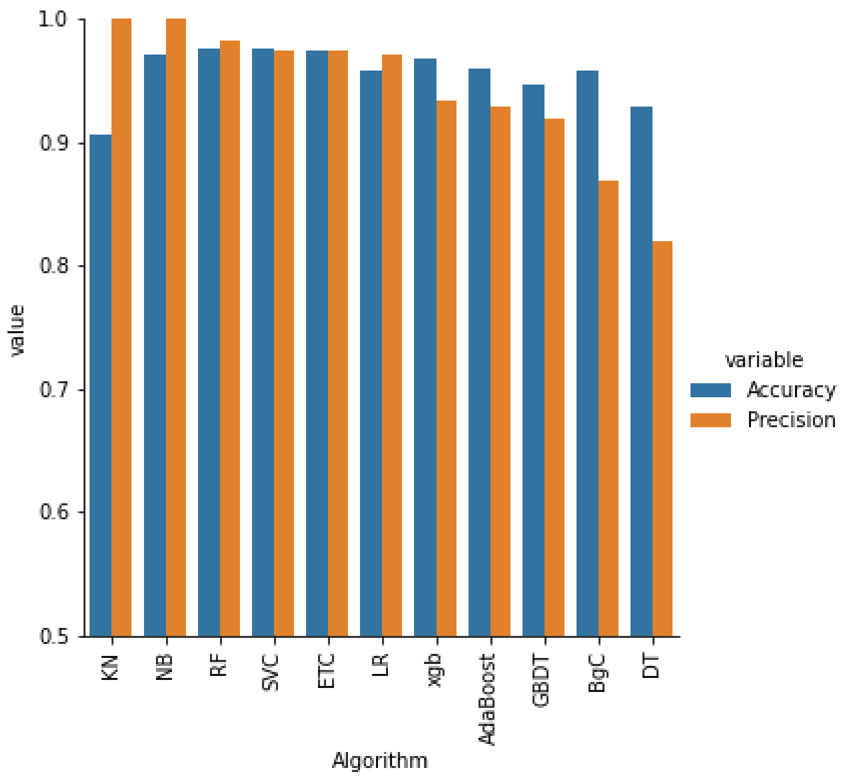
Out[71]:

	Algorithm	variable	value
0	KN	Accuracy	0.905222
1	NB	Accuracy	0.970986
2	RF	Accuracy	0.975822
3	SVC	Accuracy	0.975822
4	ETC	Accuracy	0.974855
5	LR	Accuracy	0.958414
6	xgb	Accuracy	0.967118
7	AdaBoost	Accuracy	0.960348
8	GBDT	Accuracy	0.946809
9	BgC	Accuracy	0.958414
10	DT	Accuracy	0.928433
11	KN	Precision	1.000000
12	NB	Precision	1.000000
13	RF	Precision	0.982906
14	SVC	Precision	0.974790
15	ETC	Precision	0.974576
16	LR	Precision	0.970297
17	xgb	Precision	0.933333
18	AdaBoost	Precision	0.929204
19	GBDT	Precision	0.919192
20	BgC	Precision	0.868217
21	DT	Precision	0.820000

In [72]:

```
sns.catplot(x = 'Algorithm', y='value', hue = 'variable',data=performance_df1, kind='bar
plt.ylim(0.5,1.0)
plt.xticks(rotation='vertical')
plt.show()
```

```
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\sanja\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498: Futu
reWarning: is_categorical_dtype is deprecated and will be removed in a fu
ture version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



In [73]:

```
temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_max_ft_3000':accuracy_scores,'
```

In [75]:

```
temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_scaling':accuracy_scores,'Prec
```

In [76]:

```
new_df = performance_df.merge(temp_df,on='Algorithm')
```

In [77]:

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

In [78]:

```
temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_num_chars':accuracy_scores,'Pr
```

In [79]:

```
new_df_scaled.merge(temp_df,on='Algorithm')
```

Out[79]:

	Algorithm	Accuracy	Precision	Accuracy_scaling_x	Precision_scaling_x	Accuracy_scali
0	KN	0.905222	1.000000	0.905222	1.000000	0.90
1	NB	0.970986	1.000000	0.970986	1.000000	0.97
2	RF	0.975822	0.982906	0.975822	0.982906	0.97
3	SVC	0.975822	0.974790	0.975822	0.974790	0.97
4	ETC	0.974855	0.974576	0.974855	0.974576	0.97
5	LR	0.958414	0.970297	0.958414	0.970297	0.95
6	xgb	0.967118	0.933333	0.967118	0.933333	0.96
7	AdaBoost	0.960348	0.929204	0.960348	0.929204	0.96
8	GBDT	0.946809	0.919192	0.946809	0.919192	0.94
9	BgC	0.958414	0.868217	0.958414	0.868217	0.95
10	DT	0.928433	0.820000	0.928433	0.820000	0.92

In [80]:

```
# Voting Classifier
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 [81]:

```
voting = VotingClassifier(estimators=[('svm', svc), ('nb', mnb), ('et', etc)],voting='sc
voting.fit(X_train,y_train)
```

Out[81]:

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

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [82]:

```
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
```

In [83]:

```
# Applying stacking
estimators=[('svm', svc), ('nb', mnb), ('et', etc)]
final_estimator=RandomForestClassifier()
```

In [84]:

```
from sklearn.ensemble import StackingClassifier
```

In [85]:

```
clf = StackingClassifier(estimators=estimators, final_estimator=final_estimator)
clf.fit(X_train,y_train)
```

Out[85]:

```
StackingClassifier(estimators=[('svm',
                                SVC(gamma=1.0, kernel='sigmoid',
                                    probability=True)),
                               ('nb', MultinomialNB()),
                               ('et',
                                ExtraTreesClassifier(n_estimators=50,
                                                     random_state=2))],
                   final_estimator=RandomForestClassifier())
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [86]:

```
y_pred = clf.predict(X_test)
print("Accuracy",accuracy_score(y_test,y_pred))
print("Precision",precision_score(y_test,y_pred))
```

```
Accuracy 0.9816247582205029
Precision 0.9541984732824428
```

In [87]:

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

In [ ]:

```

```