sms-spam-classifier

February 5, 2024

[242]:

```
import numpy as np
import pandas as pd
df = pd.read_csv('spam.csv', encoding='ISO-8859-1')
df.head()
```

[242]:

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

  Unnamed: 3 Unnamed: 4
0        NaN        NaN
1        NaN        NaN
2        NaN        NaN
3        NaN        NaN
4        NaN        NaN
```

[243]:

```
df.shape
```

[243]:

```
(5572, 5)
```

[244]:

```
# Data Cleaning
# EDA
# Text preprocessing
# Model Building
# Evaluation
# Improvement
# Website
# Deploy
```

1 Data Cleaning

[245]:

```
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
```

[246]:
```
df.columns
```

[246]:
```
Index(['v1', 'v2', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], dtype='object')
```

[247]:
```
#dropping last 3 columns
df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace=True)
```

[248]:
```
df.head(3)
```

[248]:
```
     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…
```

[249]:
```
# renaming the columns
df.rename(columns={'v1':'target','v2':'text'},inplace=True)
df.head(3)
```

[249]:
```
  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…
```

[250]:
```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df['target']=encoder.fit_transform(df['target'])
df.head()
```

[250]:
```
   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…
```

```
[251]: # missing values
       df.isnull().sum()
```

```
[251]: target    0
       text      0
       dtype: int64
```

```
[252]: # check your duplicated values
       df.duplicated().sum()
```

```
[252]: 403
```

```
[253]: # remove duplicates
       df=df.drop_duplicates(keep='first')
```

```
[254]: df.duplicated().sum()
```

```
[254]: 0
```

```
[255]: df.shape
```

```
[255]: (5169, 2)
```

2 EDA

```
[256]: df.head()
```

```
[256]:    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…
```

```
[257]: df['target'].unique()
```

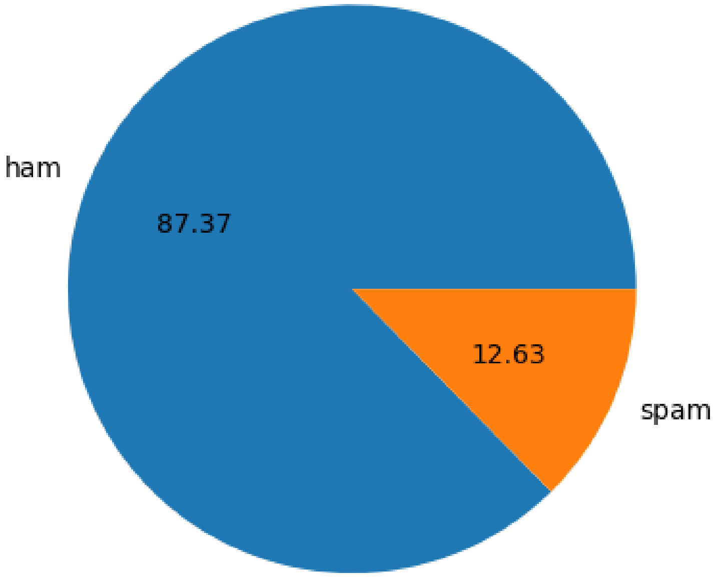
```
[257]: array([0, 1])
```

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

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

[259]:

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



[260]:

```
# data is imbalances
```

[261]:

```
!pip install nltk
```

```
Requirement already satisfied: nltk in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (3.8.1)


[notice] A new release of pip is available: 23.2.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip


Requirement already satisfied: click in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
nltk) (8.1.7)
Requirement already satisfied: joblib in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
nltk) (2023.12.25)
```

```
Requirement already satisfied: tqdm in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
nltk) (4.66.1)
Requirement already satisfied: colorama in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
click->nltk) (0.4.6)
```

[262]:

```
import nltk
```

[263]:

```
nltk.download('punkt')
```

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

[263]:

```
True
```

[264]:

```
# fetching number of characters in each instance
df['num_characters']=df['text'].apply(len)
```

[265]:

```
df.head()
```

[265]:

```
   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
```

[266]:

```
# fetching number of words in each instance
# df['text'].apply(lambda x:nltk.word_tokenize(x))
df['num_words']=df['text'].apply(lambda x:len(nltk.word_tokenize(x)))
df.head()
```

[266]:

```
   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

   num_words
0         24
1          8
2         37
3         13
4         15
```

[267]:

```
# fetching number of words in each instance
# df['text'].apply(lambda x:nltk.sent_tokenize(x))
df['num_sentences']=df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))
df.head()
```

[267]:

```
   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

   num_words  num_sentences
0         24              2
1          8              2
2         37              2
3         13              1
4         15              1
```

[268]:

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

[268]:

```
       num_characters    num_words  num_sentences
count     4516.000000  4516.000000    4516.000000
mean        70.459256    17.123782       1.820195
std         56.358207    13.493970       1.383657
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
```

[269]:

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

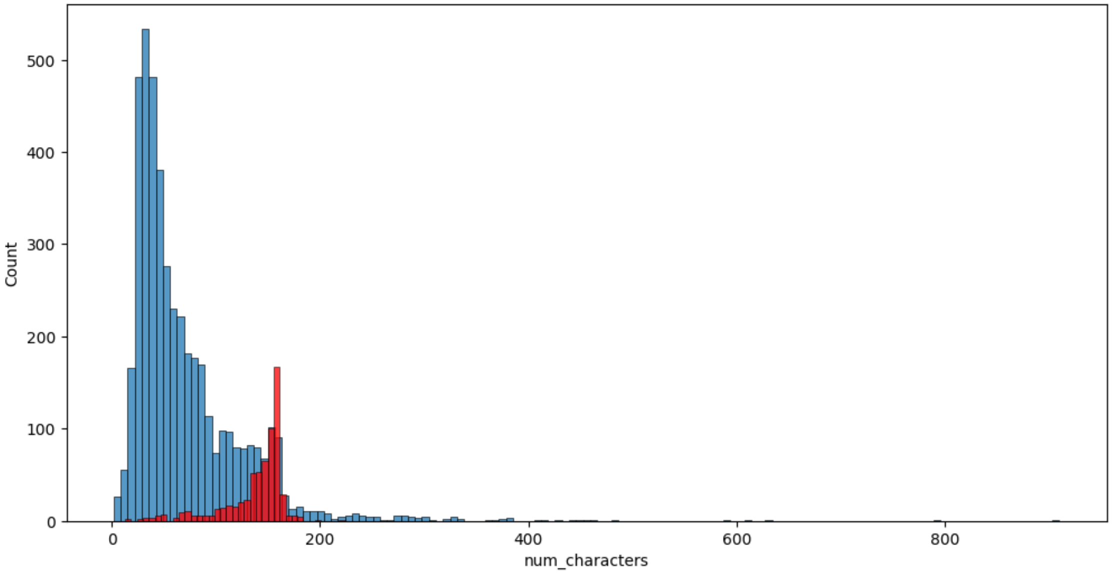
[269]:

```
       num_characters   num_words  num_sentences
count      653.000000  653.000000     653.000000
mean       137.891271   27.667688       2.970904
std         30.137753    7.008418       1.488425
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
```

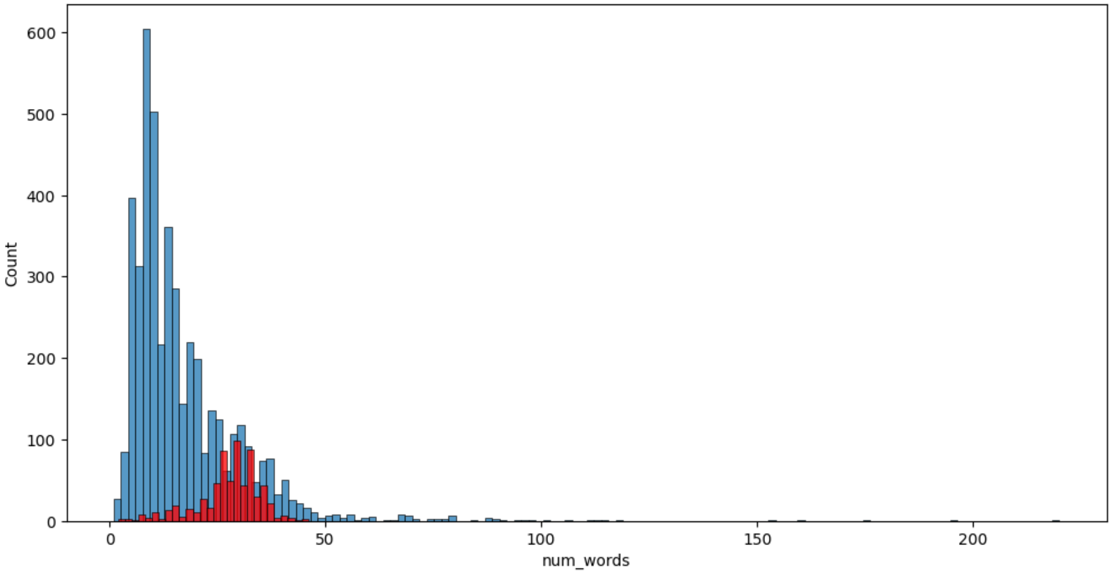
[270]:

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

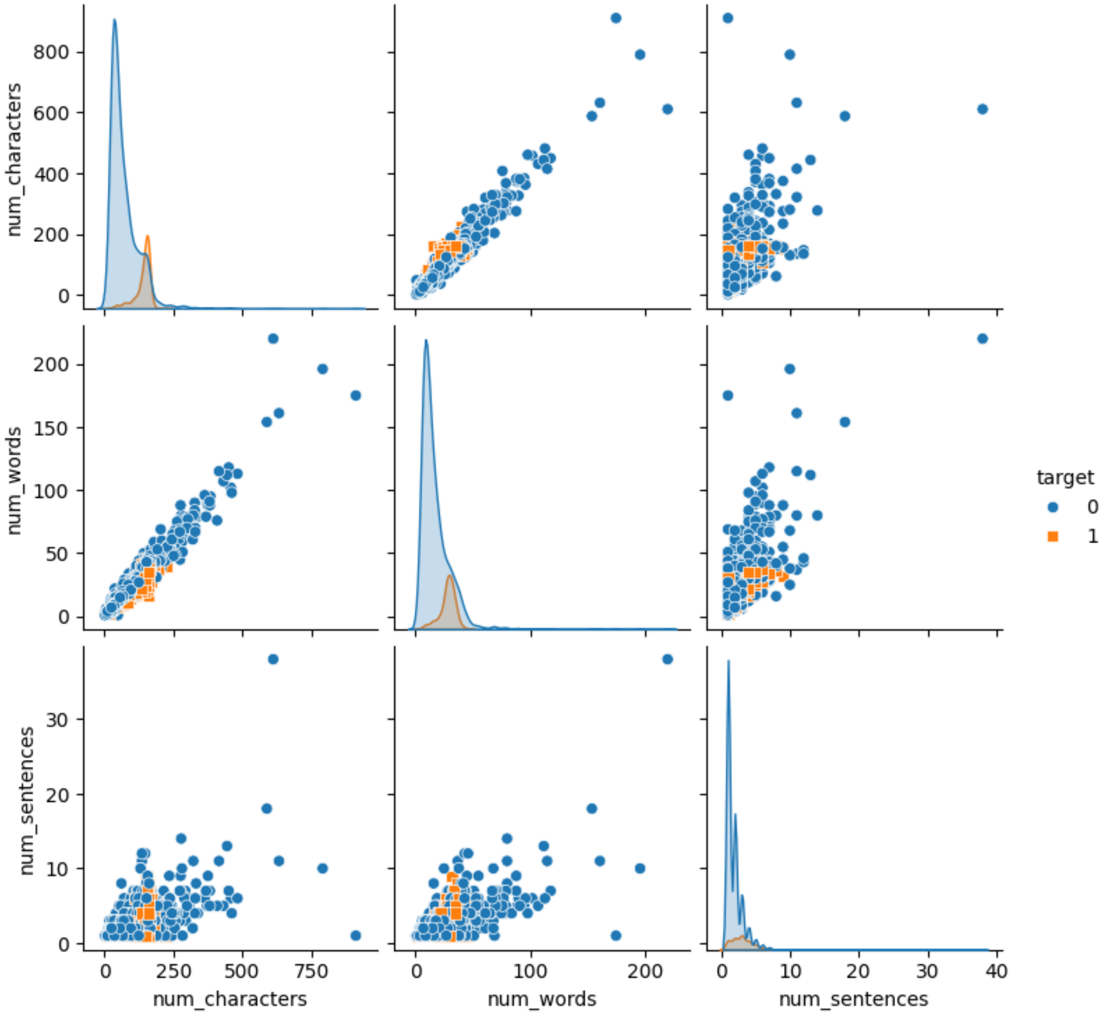
```
plt.show()
```



[271]:
```
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()
```

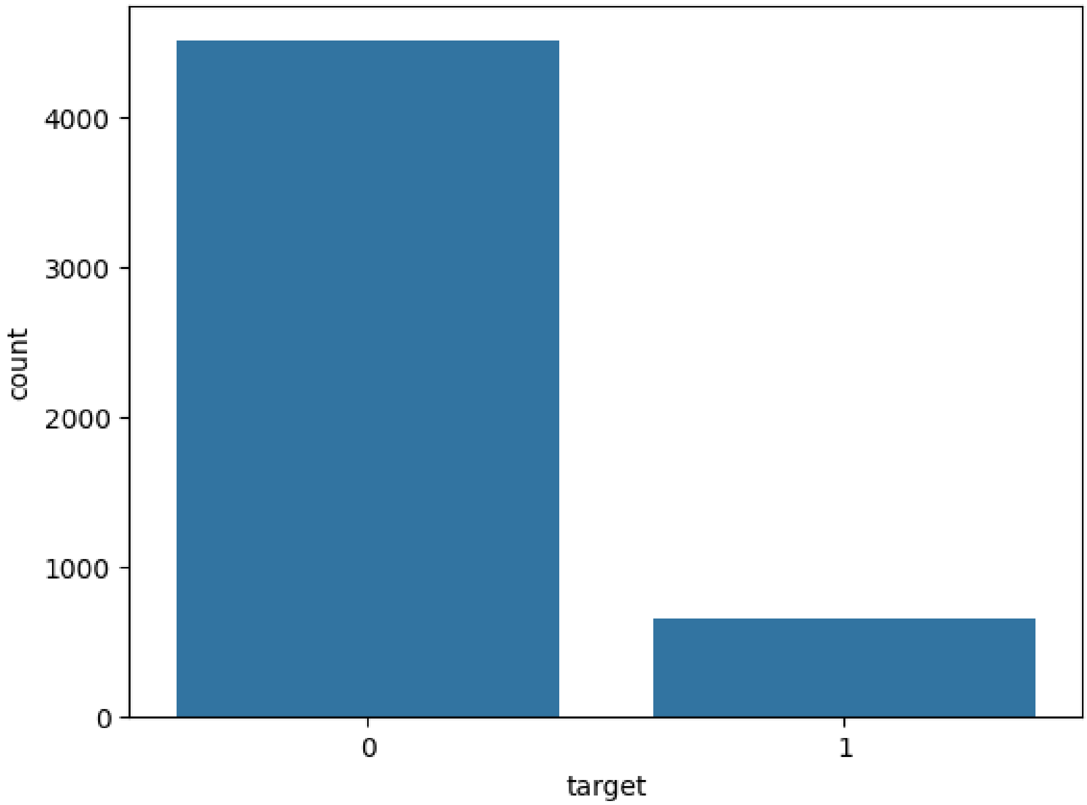


[272]:
```
sns.pairplot(df, hue='target', markers=["o", "s"])
plt.show()
```



[273]:
```
# constructing a Heat Map
# sns.heatmap(df.corr(),annot=True)
```

[274]:
```
sns.countplot(x='target',data=df)
plt.show()
```



3 Data Preprocessing

1. Lower Case
2. Tokenization
3. Removing Special Characters
4. Removing stop words and Punctuations
5. stemming

[275]:
```
nltk.download('stopwords')
```

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

[275]:
```
True
```

[276]:
```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()

from nltk.corpus import stopwords
stopwords.words('english')
```

```
import string
string.punctuation

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)
```

[277]:

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

[277]:

```
   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

   num_words  num_sentences                                  transformed_text
0         24              2  go jurong point crazi avail bugi n great world…
1          8              2                            ok lar joke wif u oni
2         37              2  free entri 2 wkli comp win fa cup final tkt 21…
3         13              1              u dun say earli hor u c alreadi say
4         15              1             nah think goe usf live around though
```

[278]:

```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (1.9.3)
Requirement already satisfied: numpy>=1.6.1 in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
wordcloud) (1.26.3)
Requirement already satisfied: pillow in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
wordcloud) (10.2.0)
Requirement already satisfied: matplotlib in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
wordcloud) (3.8.2)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->wordcloud) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->wordcloud) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->wordcloud) (4.47.2)
Requirement already satisfied: kiwisolver>=1.3.1 in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->wordcloud) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->wordcloud) (23.2)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->wordcloud) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in
c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from
python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)


[notice] A new release of pip is available: 23.2.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
```

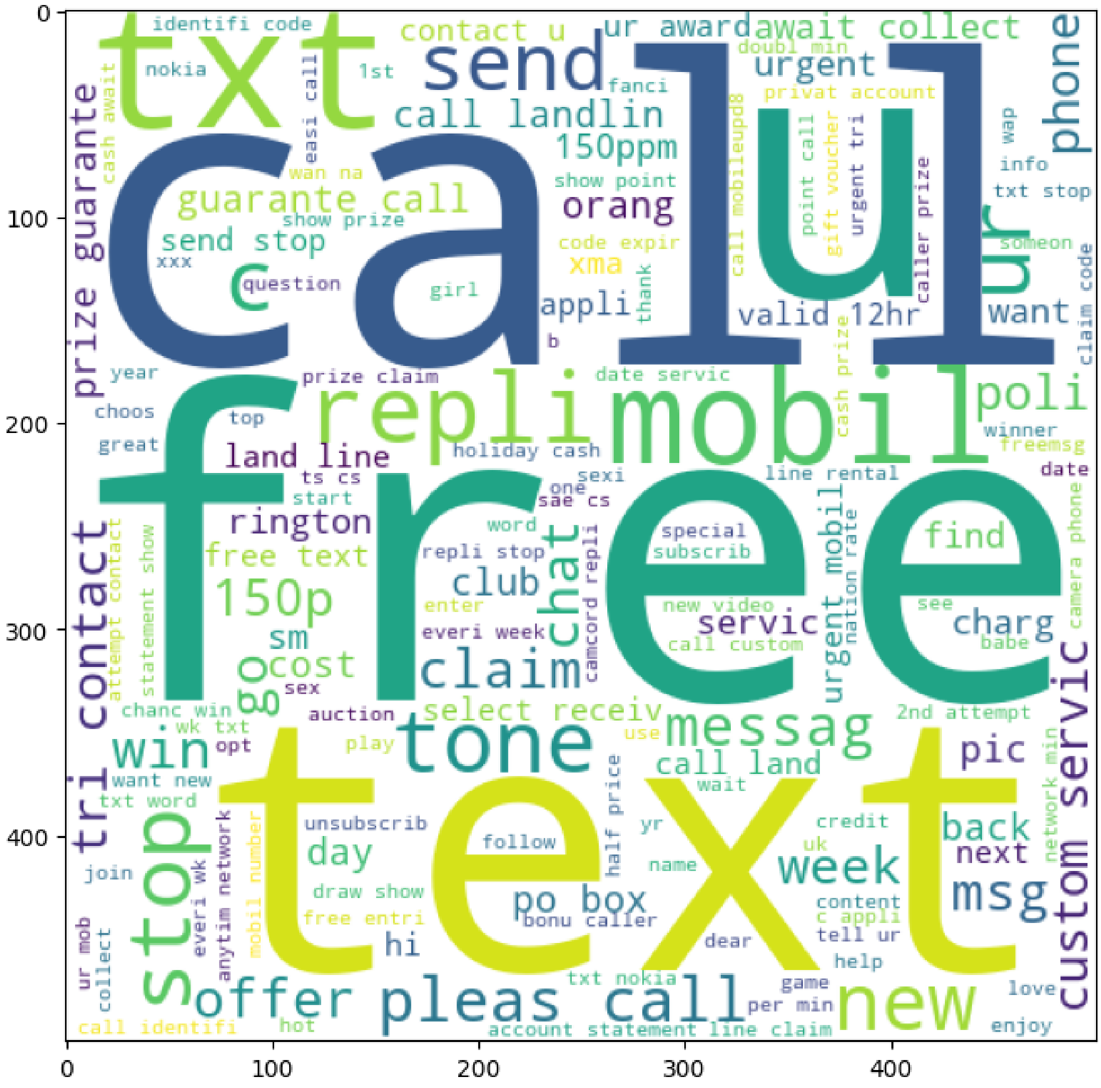
[279]:

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

[280]:

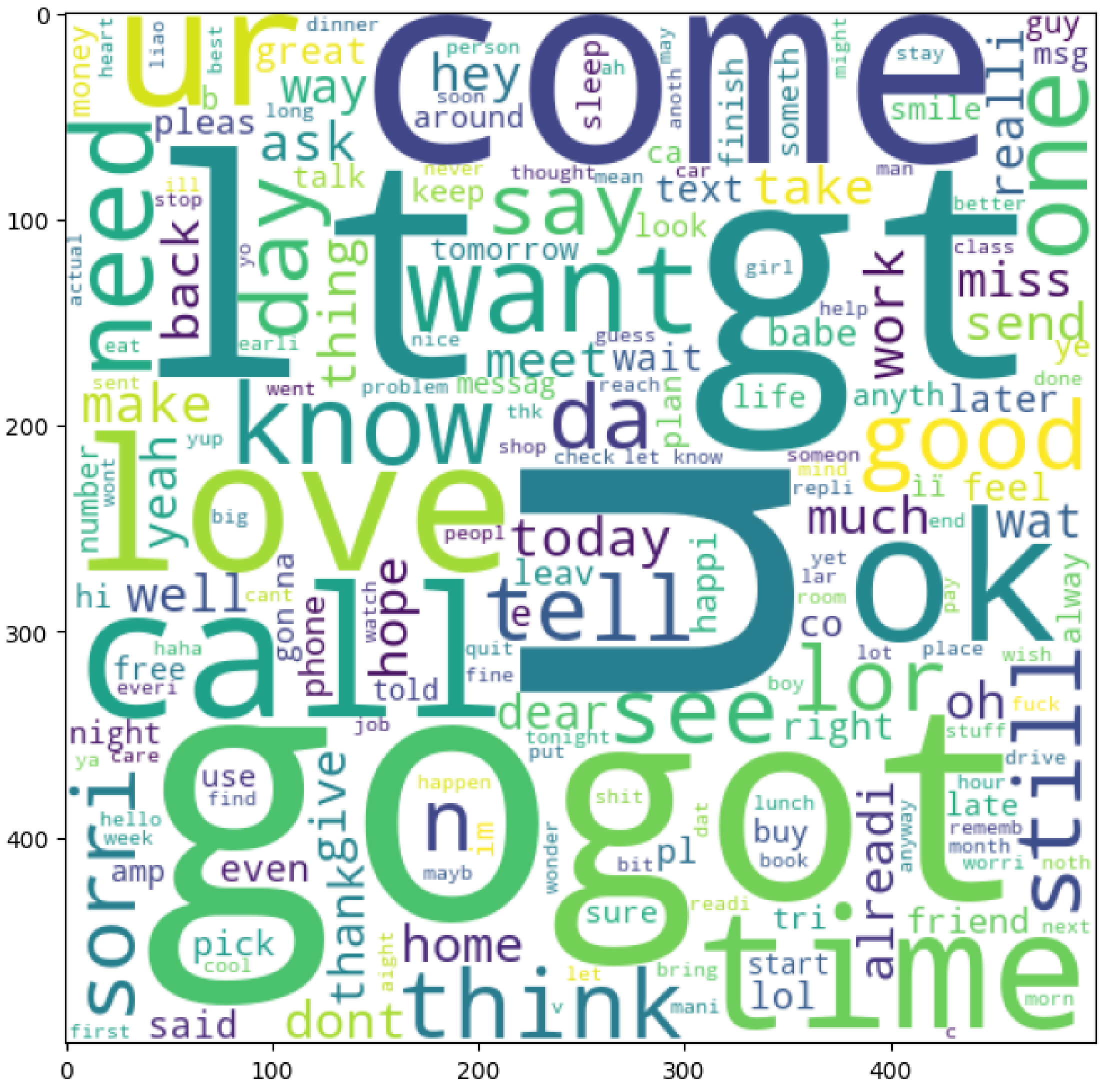
```python
spam_wc=wc.generate(df[df['target']==1]['transformed_text'].str.cat(sep=" "))
plt.figure(figsize=(12,8))
plt.imshow(spam_wc)
```

```
[280]: <matplotlib.image.AxesImage at 0x1b559953ed0>
```



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

```
[281]: <matplotlib.image.AxesImage at 0x1b552303150>
```



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

```
[285]: len(spam_corpus)
```

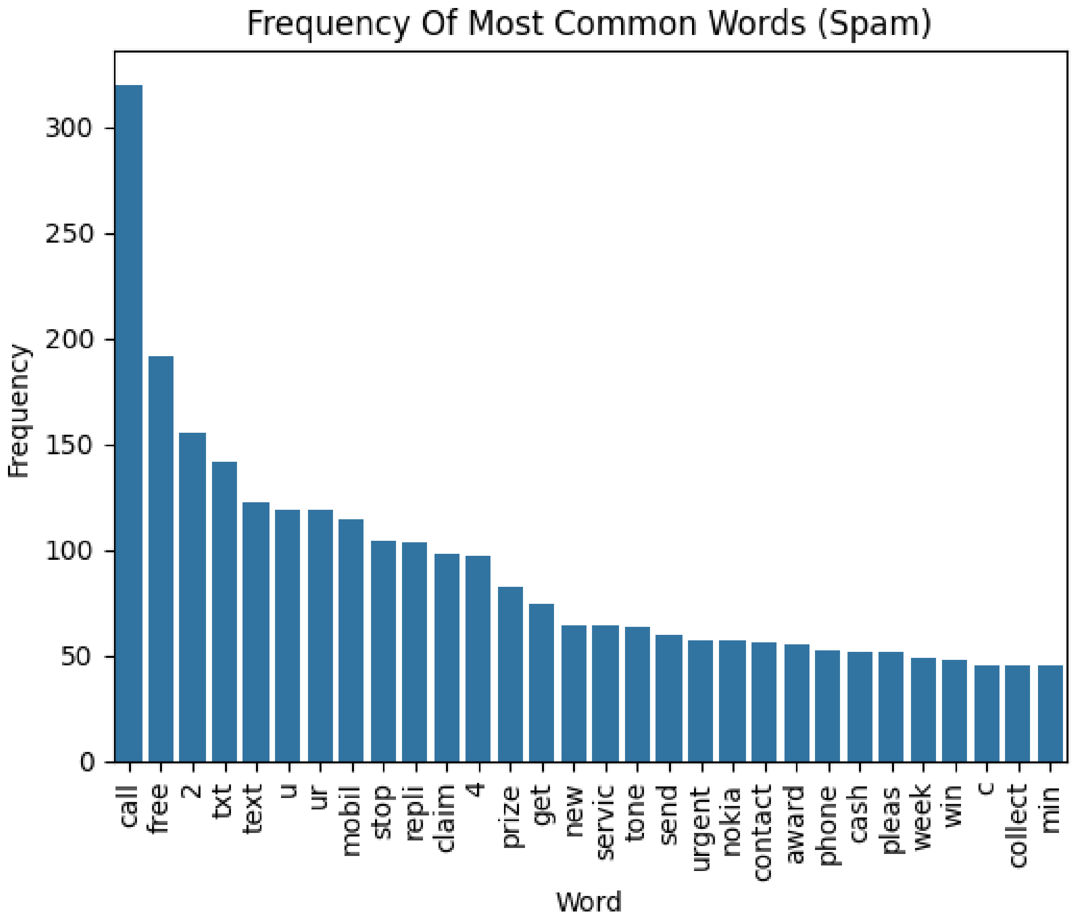
```
[285]: 9939
```

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

[297]:
```
len(ham_corpus)
```

[297]:
```
35404
```

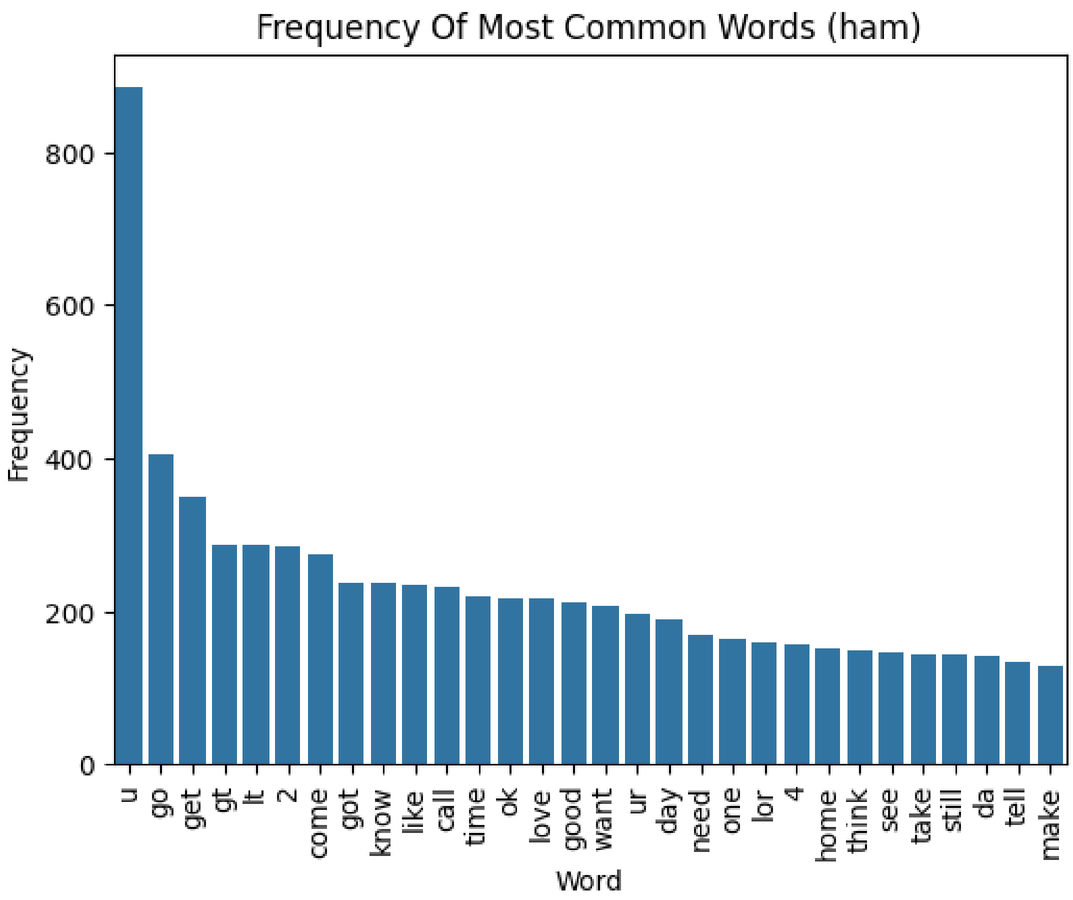
[290]:
```
from collections import Counter
wordFreq_spam = pd.DataFrame(Counter(spam_corpus).most_common(30),␣
 ↪columns=['Word', 'Frequency'])

# Now, use seaborn's barplot with keyword arguments for x and y
sns.barplot(x=wordFreq_spam['Word'], y=wordFreq_spam['Frequency'])
plt.xticks(rotation='vertical')
plt.title("Frequency Of Most Common Words (Spam)")
plt.show()
```



[287]:
```
wordFreq_spam.head()
```

```
[287]:    Word  Frequency
       0  call        320
       1  free        191
       2     2        155
       3   txt        141
       4  text        122
```

```
[291]: from collections import Counter
       wordFreq_ham = pd.DataFrame(Counter(ham_corpus).most_common(30),␣
        ↪columns=['Word', 'Frequency'])
       # Now, use seaborn's barplot with keyword arguments for x and y
       sns.barplot(x=wordFreq_ham['Word'], y=wordFreq_ham['Frequency'])
       plt.xticks(rotation='vertical')
       plt.title("Frequency Of Most Common Words (ham)")
       plt.show()
```



```
[289]: wordFreq_ham.head()
```

```
[289]:   Word  Frequency
       0    u        883
       1   go        404
       2  get        349
       3   gt        288
       4   lt        287
```

4 Model Building

```
[351]: from sklearn.feature_extraction.text import TfidfVectorizer
       tfidf = TfidfVectorizer(max_features=3000)
```

```
[352]: x=tfidf.fit_transform(df['transformed_text']).toarray()
```

```
[353]: x.shape
```

```
[353]: (5169, 3000)
```

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

```
[355]: y
```

```
[355]: array([0, 0, 1, …, 0, 0, 0])
```

```
[356]: from sklearn.model_selection import train_test_split
       from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

```
[357]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=3)
```

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

```
[359]: gnb=GaussianNB()
       mnb=MultinomialNB()
       bnb=BernoulliNB()
```

```
[360]: 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))
```

```
0.8646034816247582
[[773 121]
 [ 19 121]]
0.5
```

[361]:

```
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))
```

```
0.9806576402321083
[[893   1]
 [ 19 121]]
0.9918032786885246
```

[362]:

```
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))
```

```
0.9690522243713733
[[894   0]
 [ 32 108]]
1.0
```

[363]:

```
#tfidf ---> mnb
```

[366]:

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