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

In [2]:
```
# List of possible encodings to try
encodings= ['utf_8','latin1','ISO-8859-1','cp1252']
file_path='spam.csv' # change this to the path of your CSV file
#Attempt to read the CSV file with different encodings
for encoding in encodings:
    try:
        df= pd.read_csv(file_path,encoding=encoding)
        print(f'file successfully read with encoding: {encoding}')
        break # stop the loop if successful
    except UnicodeDecodeError:
        print(f'Failed to read with encoding: {encoding}')
        continue # Try the next encoding

        # If the loop completes without success, df wil not be defined
if 'df' in locals():
    print("CSV file has been successfully loaded.")
else:
    print("All encoding attempts failed. UNable to read the CSV file.")
```

```
Failed to read with encoding: utf_8
file successfully read with encoding: latin1
CSV file has been successfully loaded.
```

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

Out[3]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
1696	ham	Sorry man, my stash ran dry last night and I c...	NaN	NaN	NaN
3797	ham	Feb &lt;#&gt; is \I LOVE U\" day. Send dis t...	NaN	NaN	NaN
492	ham	Sorry,in meeting I'll call later	NaN	NaN	NaN
5199	ham	Ugh my leg hurts. Musta overdid it on mon.	NaN	NaN	NaN
1325	ham	Yeah jay's sort of a fucking retard	NaN	NaN	NaN

In [4]:
```
df.columns
```

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

# 1.Data Cleaning

In [5]:
```
#drop last 3 cols
df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace= True)
```

In [6]:
```
df.sample(5)
```

Out[6]:

	v1	v2
1276	ham	Can do lor...
5525	ham	I want to tell you how bad I feel that basical...
4716	ham	K will do, addie &amp; I are doing some art so...
1495	ham	Hey gals.. Anyone of u going down to e driving...
5246	ham	Haven't eaten all day. I'm sitting here starin...

In [7]:
```
#renaming the cols
df.rename(columns={'v1':'target','v2':'text'},inplace=True)
```

In [8]:
```
df.sample(10)
```

Out[8]:
```
      target                                               text
3345     ham        U're welcome... Caught u using broken english ...
4412     ham       Sad story of a Man - Last week was my b'day. M...
3896     ham                     No. Thank you. You've been wonderful
  88     ham                      I'm really not up to it still tonight babe
3036     ham           Cos darren say Ì_ considering mah so i ask Ì_...
1120     ham                  Cancel cheyyamo?and get some money back?
2420     ham              Oic... Then better quickly go bathe n settle d...
4557     ham   PISS IS TALKING IS SOMEONE THAT REALISE U THAT...
3620     ham             That means from february to april i'll be gett...
3649     ham         We are hoping to get away by 7, from Langport....
```

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)
```

EDA

In [17]:
```
df.head(10)
```

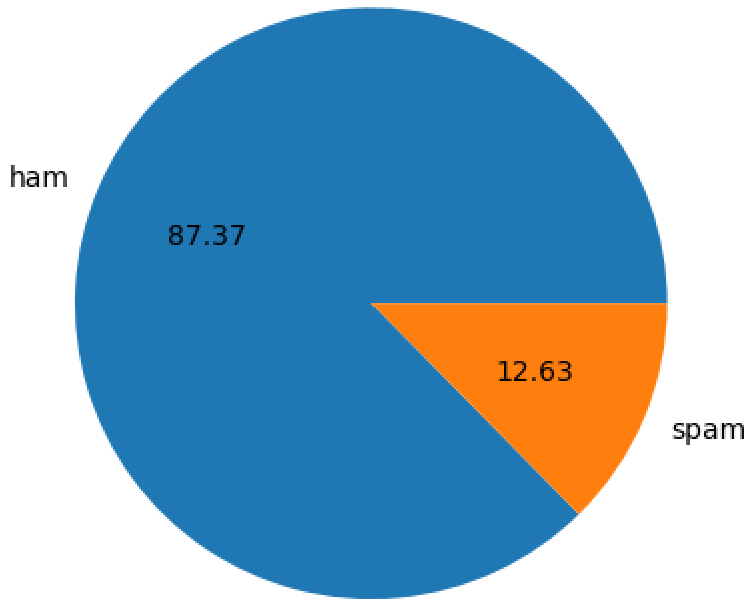
Out[17]:

	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...
5	1	FreeMsg Hey there darling it's been 3 week's n...
6	0	Even my brother is not like to speak with me. ...
7	0	As per your request 'Melle Melle (Oru Minnamin...
8	1	WINNER!! As a valued network customer you have...
9	1	Had your mobile 11 months or more? U R entitle...

In [18]:
```
df['target'].value_counts()
```

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

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



In [20]:
```python
import nltk
!pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\user\anaconda3\lib\site-packages (3.7)
Requirement already satisfied: regex>=2021.8.3 in c:\users\user\anaconda3\lib\site-packages (from nltk) (2022.7.9)
Requirement already satisfied: tqdm in c:\users\user\anaconda3\lib\site-packages (from nltk) (4.64.1)
Requirement already satisfied: joblib in c:\users\user\anaconda3\lib\site-packages (from nltk) (1.1.1)
Requirement already satisfied: click in c:\users\user\anaconda3\lib\site-packages (from nltk) (8.0.4)
Requirement already satisfied: colorama in c:\users\user\anaconda3\lib\site-packages (from click->nltk) (0.4.6)
```

In [21]:
```
nltk.download('punkt')
```

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

Out[21]:
```
True
```

In [22]:
```
df['num_characters'] = df['text'].apply(len) #number of char
```

In [23]:
```
#number of words
df['num_words'] = df['text'].apply(lambda x:len(nltk.word_tokenize(x)))#words count
```

In [24]:
```
df['num_sentences'] = df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))#words count
```

In [25]:
```
df.head(10)
```

Out[25]:

	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
5	1	FreeMsg Hey there darling it's been 3 week's n...	148	39	4
6	0	Even my brother is not like to speak with me. ...	77	18	2
7	0	As per your request 'Melle Melle (Oru Minnamin...	160	31	2
8	1	WINNER!! As a valued network customer you have...	158	32	5
9	1	Had your mobile 11 months or more? U R entitle...	154	31	3

In [26]:
```
df[['num_characters','num_words','num_sentences']].describe()
```

Out[26]:

	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 [27]:
```
#targetting ham
df[df['target']== 0][['num_characters','num_words','num_sentences']].describe()
```

Out[27]:

	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

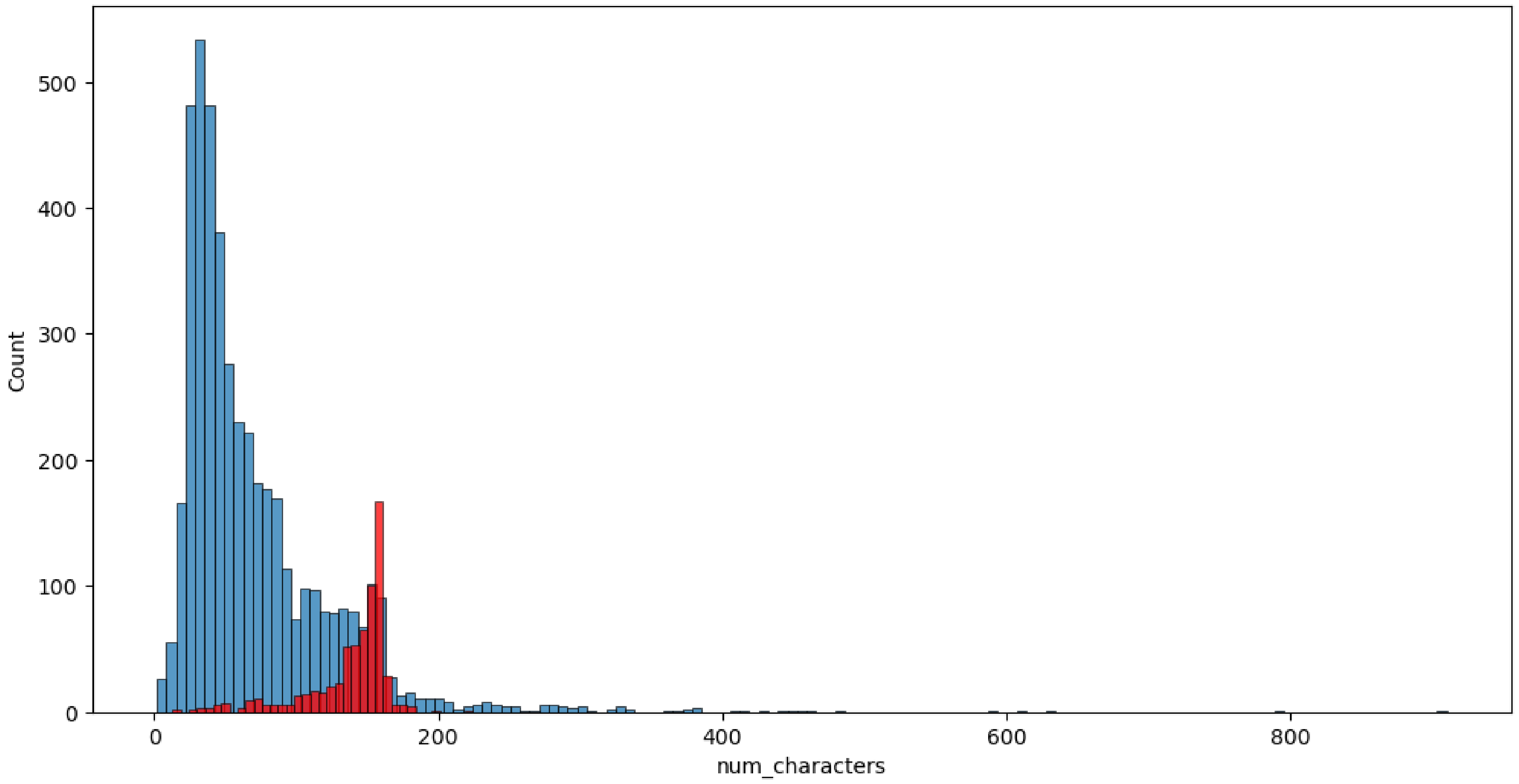
In [28]:
```
#targetting spam
df[df['target']== 1][['num_characters','num_words','num_sentences']].describe()
```

Out[28]:
```
       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
```

In [29]:
```
import seaborn as sns
```

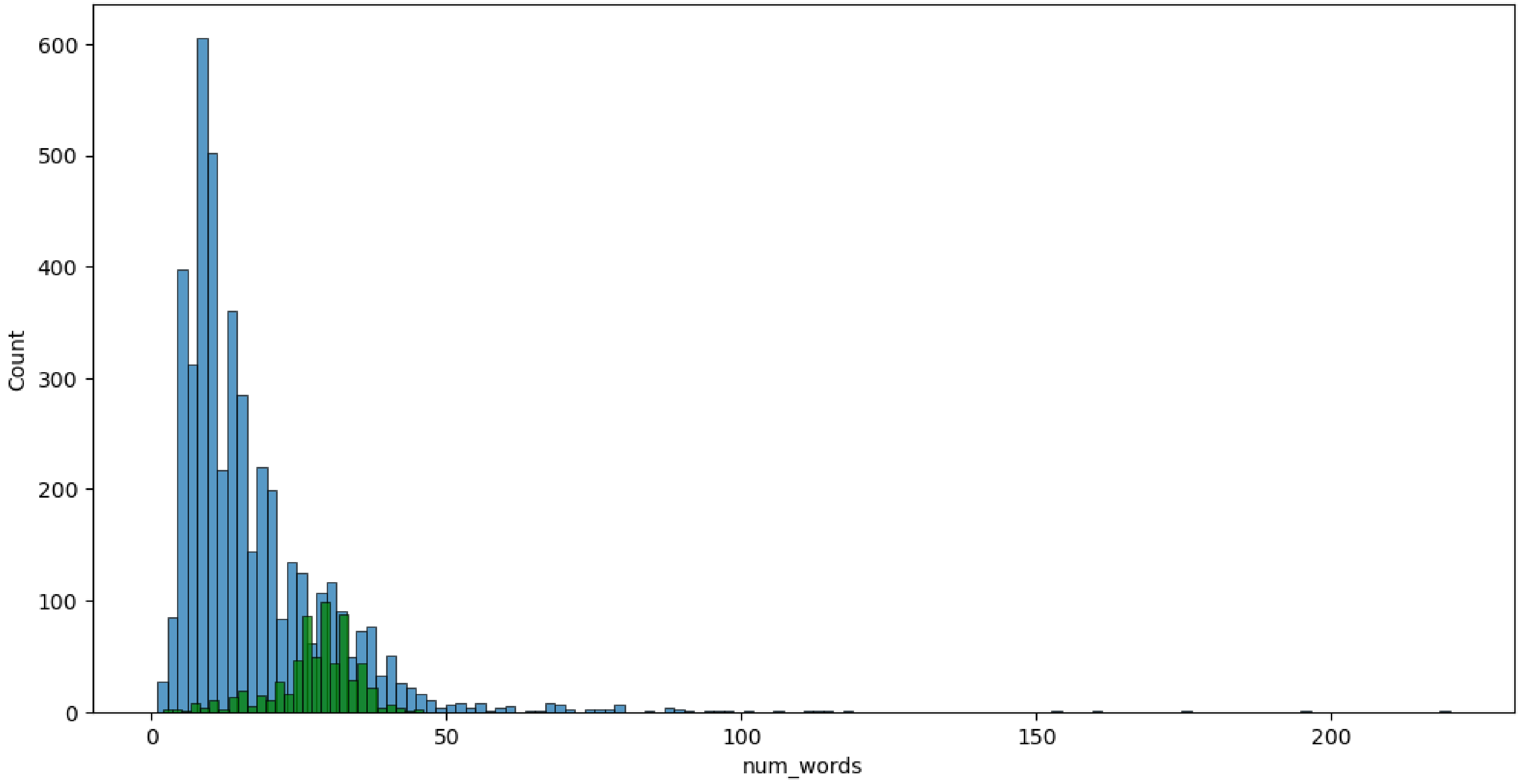
In [30]:
```
plt.figure(figsize=(12,6))
sns.histplot(df[df['target']==0]['num_characters']) #ham
sns.histplot(df[df['target']==1]['num_characters'],color='red') #spam
```

Out[30]:
```
<Axes: xlabel='num_characters', ylabel='Count'>
```



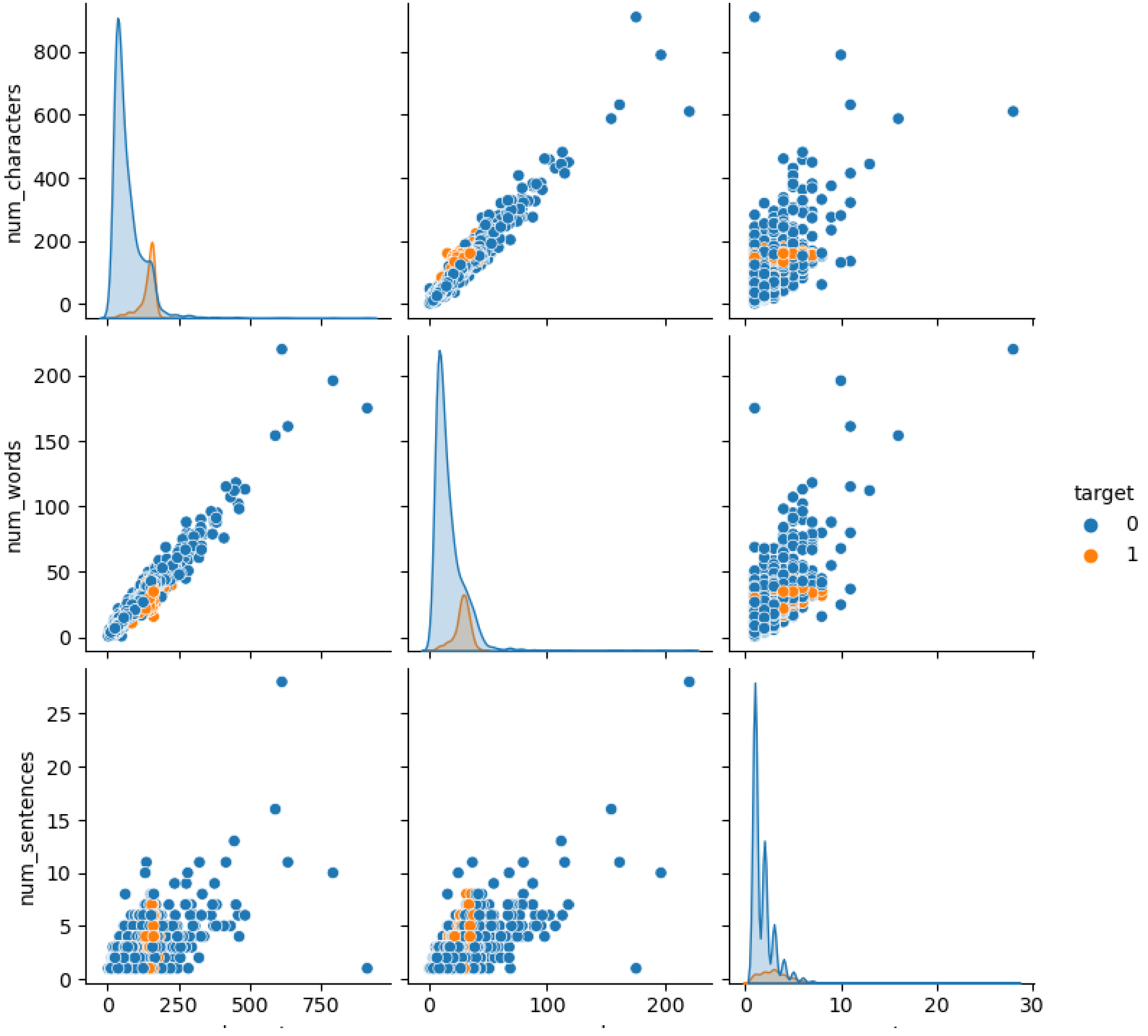
In [31]:
```
plt.figure(figsize=(12,6))
sns.histplot(df[df['target']==0]['num_words']) #ham
sns.histplot(df[df['target']==1]['num_words'],color='green') #spam
```

Out[31]:
```
<Axes: xlabel='num_words', ylabel='Count'>
```



```
In [32]: sns.pairplot(df,hue='target')
```

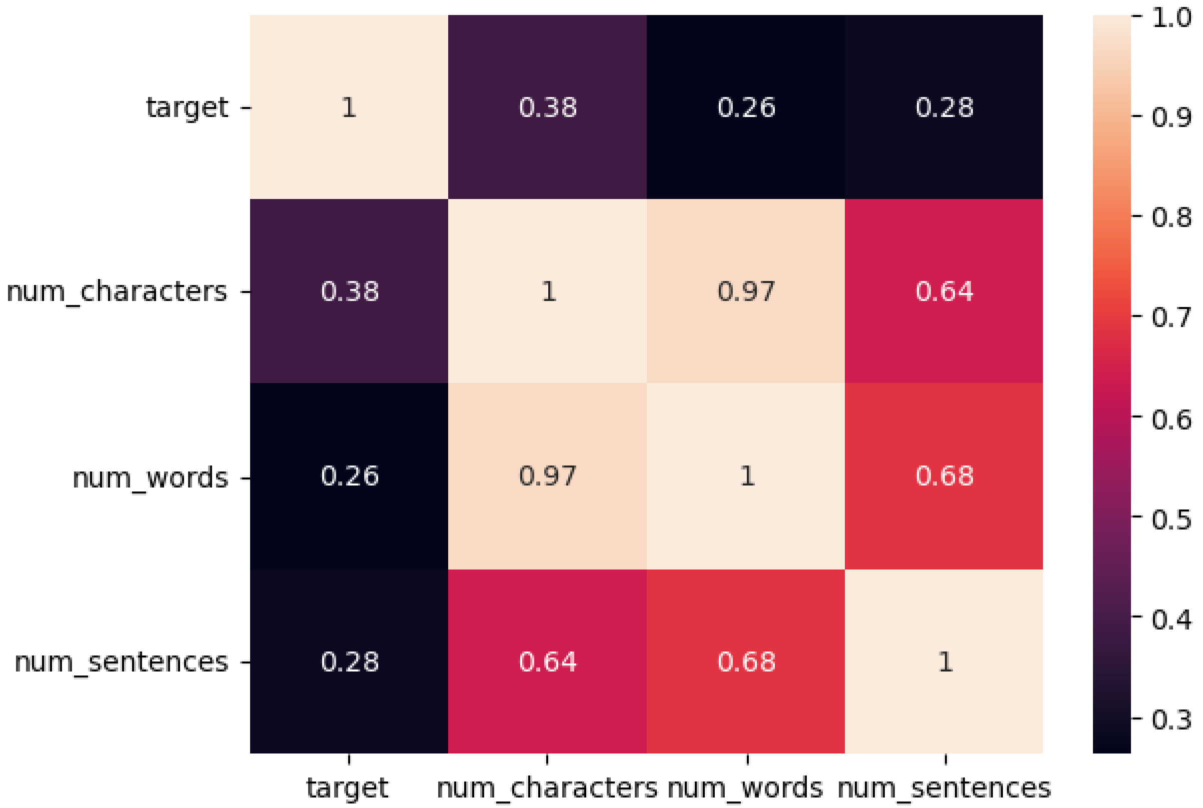
```
Out[32]: <seaborn.axisgrid.PairGrid at 0x2306674f850>
```


num_characters
num_words
num_sentences
target
● 0
● 1
800
600
400
200
0
150
100
50
25
20
15
10
5
250
500
750
30

In [33]:
```
sns.heatmap(df.corr(),annot=True)
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_4236\4277794465.py:1: FutureWarning: The default value of numeric_only in
DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  sns.heatmap(df.corr(),annot=True)
```

Out[33]:
```
<Axes: >
```



Data Preprocessing

LOWER CASE

TOKENIZATION

REMOVING SPECIAL CHARACTERS

REMOVING STOP WORDS AND PUNCTUATION

STEMMING

In [35]:
```
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import string
```

In [36]:
```
nltk.download('stopwords') #You may need to download the stopwords dataset
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)

transformed_text = transform_text("I'm gonna be home soon")

print(transformed_text)
```

```
gon na home soon

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

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

Out[37]:
```
'walk'
```

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 [47]:
```
from wordcloud import WordCloud
wc= WordCloud(width=500, height=500, min_font_size=10,background_color='black')
```

In [41]:
```
pip install wordcloud
```

```
Collecting wordcloud
  Downloading wordcloud-1.9.2-cp310-cp310-win_amd64.whl (152 kB)
     -------------------------------------- 152.1/152.1 kB 1.8 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.6.1 in c:\users\user\anaconda3\lib\site-packages (from wordcloud) (1.23.5)
Requirement already satisfied: pillow in c:\users\user\anaconda3\lib\site-packages (from wordcloud) (9.4.0)
Requirement already satisfied: matplotlib in c:\users\user\anaconda3\lib\site-packages (from wordcloud) (3.7.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\user\anaconda3\lib\site-packages (from matplotlib->wordcl
oud) (1.0.5)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\user\anaconda3\lib\site-packages (from matplotlib->wordc
loud) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\user\anaconda3\lib\site-packages (from matplotlib->wordclo
ud) (22.0)
Requirement already satisfied: cycler>=0.10 in c:\users\user\anaconda3\lib\site-packages (from matplotlib->wordcloud)
(0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\user\anaconda3\lib\site-packages (from matplotlib->wordc
loud) (4.25.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\user\anaconda3\lib\site-packages (from matplotlib->wordcl
oud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\user\anaconda3\lib\site-packages (from matplotlib->wo
rdcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\user\anaconda3\lib\site-packages (from python-dateutil>=2.7->matp
lotlib->wordcloud) (1.16.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.2
Note: you may need to restart the kernel to use updated packages.
```

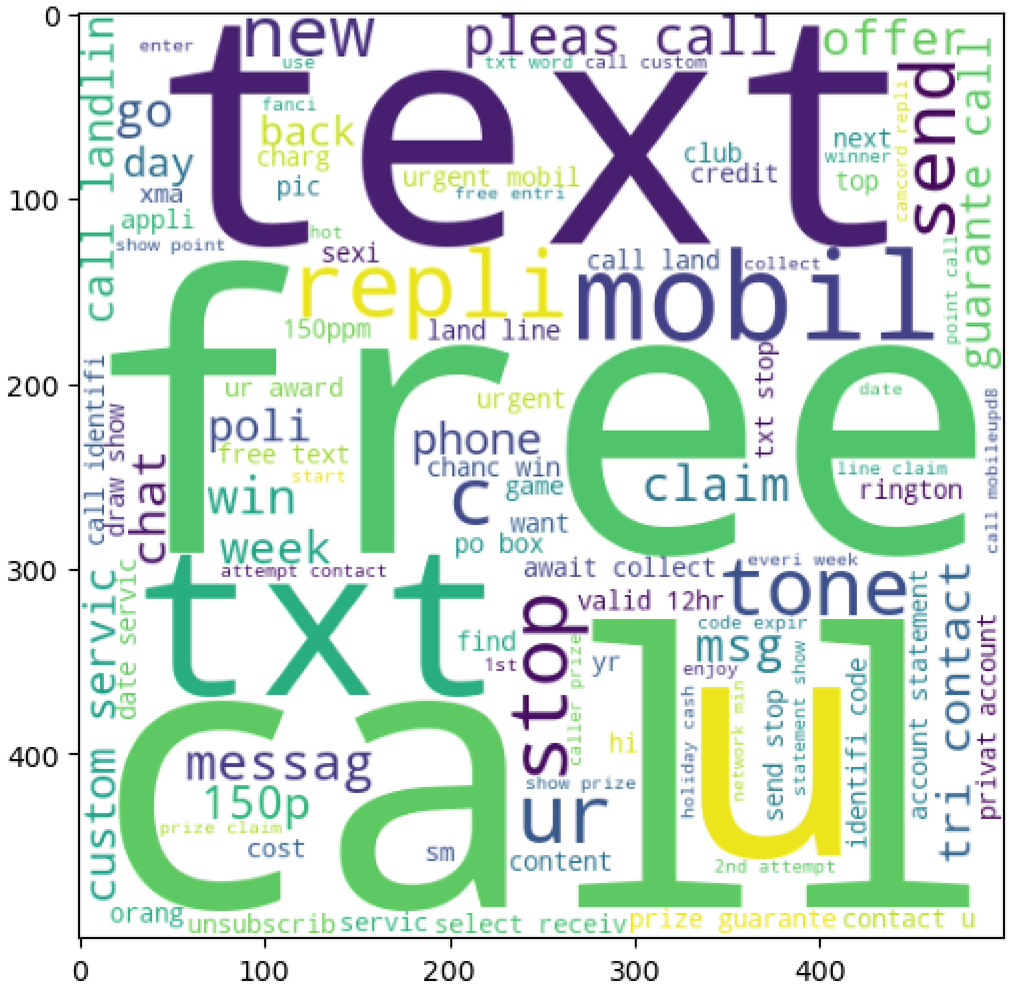
In [44]:
```
# Assuming you have a variable 'text_data' containing your text data
text_data = "Your text goes here."

# Generate the WordCloud
wordcloud = wc.generate(text_data)

# You can display the WordCloud using matplotlib or save it to a file
import matplotlib.pyplot as plt

# Display the WordCloud using matplotlib
plt.figure(figsize=(8, 8), facecolor=None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)

# Save the WordCloud to a file
wordcloud.to_file("your_wordcloud.png")

# Show the plot
plt.show()
```

goes
text

```
In [45]: spam_wc = wc.generate(df[df['target']==1]['transformed_text'].str.cat(sep= " "))
```

In [46]:
```
plt.figure(figsize=(15,6))
plt.imshow(spam_wc)
```

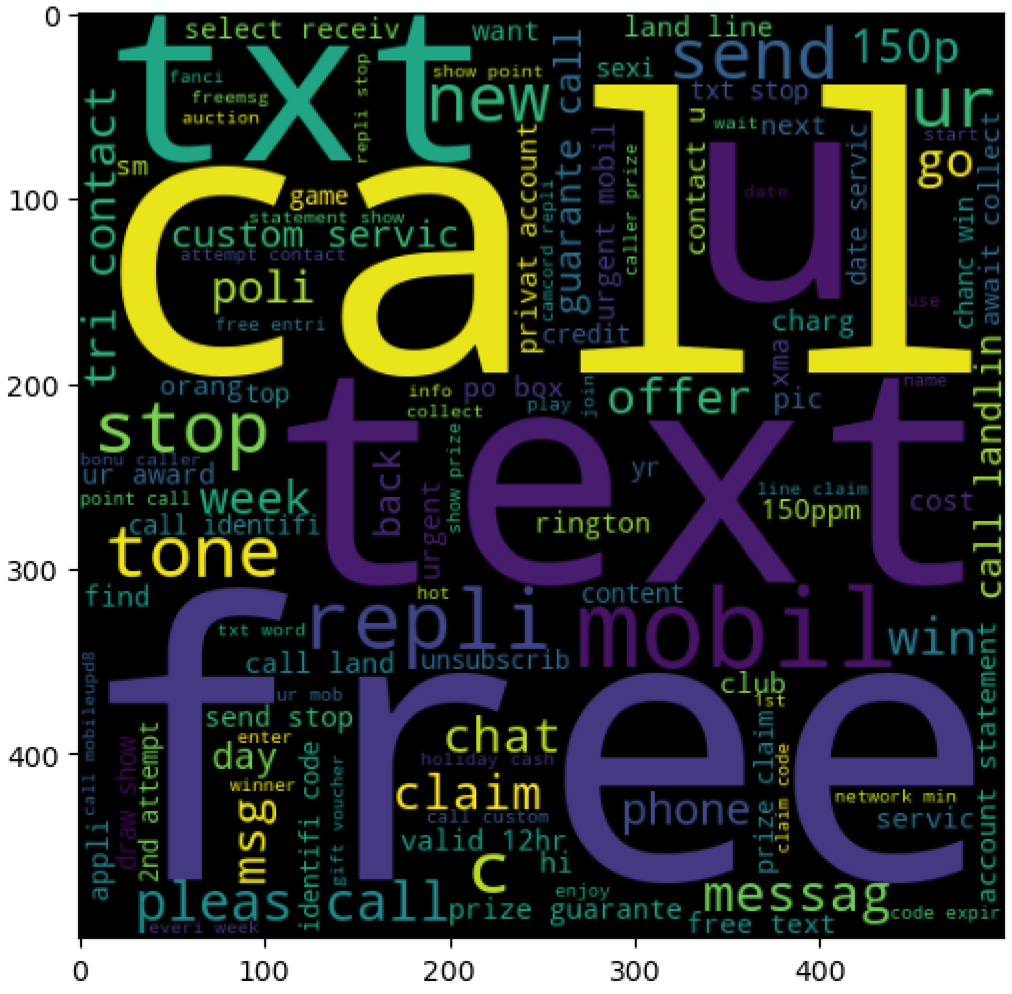
Out[46]:
```
<matplotlib.image.AxesImage at 0x2306a449150>
```



In [48]:
```
ham_wc = wc.generate(df[df['target']==1]['transformed_text'].str.cat(sep= " "))
```

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

Out[49]:
```
<matplotlib.image.AxesImage at 0x2306aa8c310>
```



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

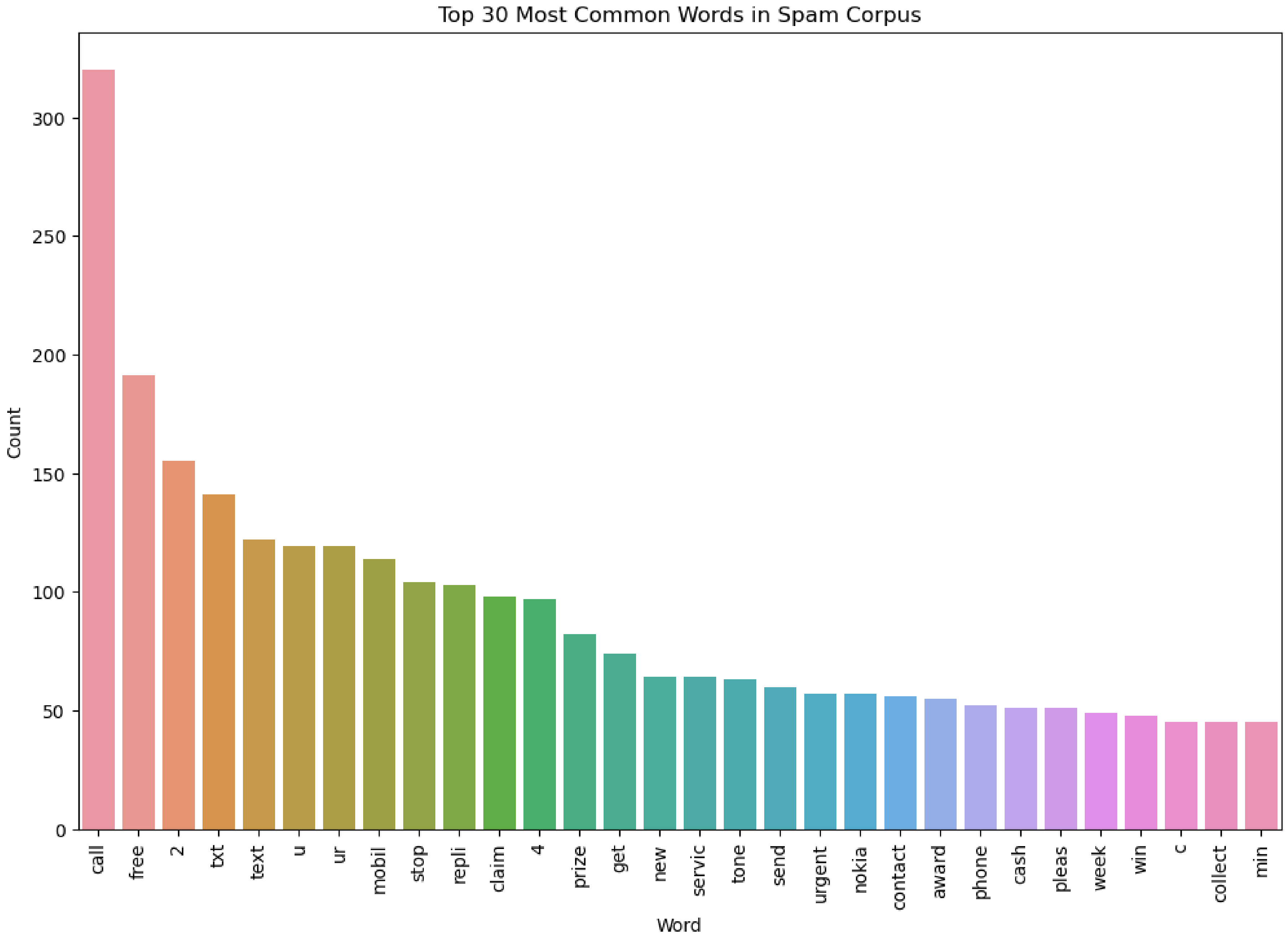
In [52]:
```
len(spam_corpus)
```

Out[52]:
```
9939
```

In [56]:
```
from collections import Counter
sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0],pd.DataFrame(Counter(spam_corpus).most_common(30))[1
plt.xticks(rotation='vertical')
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[56], line 2
      1 from collections import Counter
----> 2 sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0],pd.DataFrame(Counter(spam_corpus).most_comm
on(30))[1])
      3 plt.xticks(rotation='vertical')
      4 plt.show()

TypeError: barplot() takes from 0 to 1 positional arguments but 2 were given
```

In [58]:
```
from collections import Counter


# Created a Counter object to count word frequencies
word_counter = Counter(spam_corpus)

# Created a DataFrame with the most common 30 words and their counts
common_words_df = pd.DataFrame(word_counter.most_common(30), columns=['Word', 'Count'])

# Created a bar plot using Seaborn
plt.figure(figsize=(12, 8))
sns.barplot(x='Word', y='Count', data=common_words_df)
plt.xticks(rotation='vertical')  # Corrected the typo here
plt.title('Top 30 Most Common Words in Spam Corpus')
plt.show()
```


Top 30 Most Common Words in Spam Corpus
Count
300
250
200
150
100
50
0
call
free
2
txt
text
u
ur
mobil
stop
repli
claim
4
prize
get
new
servic
tone
send
urgent
nokia
contact
award
phone
cash
pleas
week
win
c
collect
min
Word

In [59]:
```
#Text Vectorization
#using Bag of Words
df.head()
```

Out[59]:

	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.Building the model

In [61]:
```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

# Create a CountVectorizer
cv = CountVectorizer()

# Create a TfidfVectorizer with a maximum of 3000 features
tfidf = TfidfVectorizer(max_features=3000)
```

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

In [63]:
```
#from sklearn.preprocessing import MinMaxScaler
#Scaler= MinMaxScaler()
#X= scaler.fit_transform(X)

# appending  the num_character col to X
#X = np.hstack((X,df['num_characters'].values.reshape(-1,1)))
```

In [64]:
```
X.shape
```
Out[64]:
```
(5169, 3000)
```

In [65]:
```
y = df['target'].values
```

In [66]:
```
from sklearn.naive_bayes import GaussianNB, MultinomialNB,BernoulliNB
from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

In [67]:
```
gnb= GaussianNB()
mnb= MultinomialNB()
bnb= BernoulliNB()
```

In [71]:
```
from sklearn.model_selection import train_test_split
```

In [72]:
```
X_train,X_test,y_train,y_test= train_test_split(X,y,test_size=0.2,random_state=2)
```

In [73]:
```
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 [74]:
```
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 [75]:
```
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 [77]:
```
!pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\user\anaconda3\lib\site-packages (2.0.2)
Requirement already satisfied: scipy in c:\users\user\anaconda3\lib\site-packages (from xgboost) (1.10.0)
Requirement already satisfied: numpy in c:\users\user\anaconda3\lib\site-packages (from xgboost) (1.23.5)
```

In [79]:
```
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 [80]:
```
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 [81]:
```
clfs={
    'SVC': svc,
    'KN': knc,
    'NB': mnb,
    'DT': dtc,
    'LR': lrc,
    'RF': rfc,
    'AdaBoost': abc,
    'BgC': bc,
    'ETC': etc,
    'GBDT': gbdt,
    'xgb': xgb
}
```

In [84]:
```
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 [83]:
```
train_classifier(svc,X_train,y_train,X_test,y_test)
```

Out[83]:
```
(0.9758220502901354, 0.9747899159663865)
```

In [85]:
```
accuracy_scores = []
precision_scores = []

for name,clf in clfs.items():

    current_accuracy,current_precision= train_classifier(clf,X_train,y_train,X_test,y_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 - 0.9332688588007737
Precision - 0.8415841584158416
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 - 0.9661508704061895
Precision - 0.9256198347107438
```

In [ ]:
```
performance_df= pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy':accuracy_scores,'Precision':precision_scores}).sort_v
```

In [88]:
```
performance_df
```

Out[88]:

	Algorithm	Accuracy	Precision
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
6	AdaBoost	0.960348	0.929204
10	xgb	0.966151	0.925620
9	GBDT	0.947776	0.920000
7	BgC	0.957447	0.867188
3	DT	0.933269	0.841584

In [89]:
```
performance_df1= pd.melt(performance_df, id_vars= "Algorithm")
```

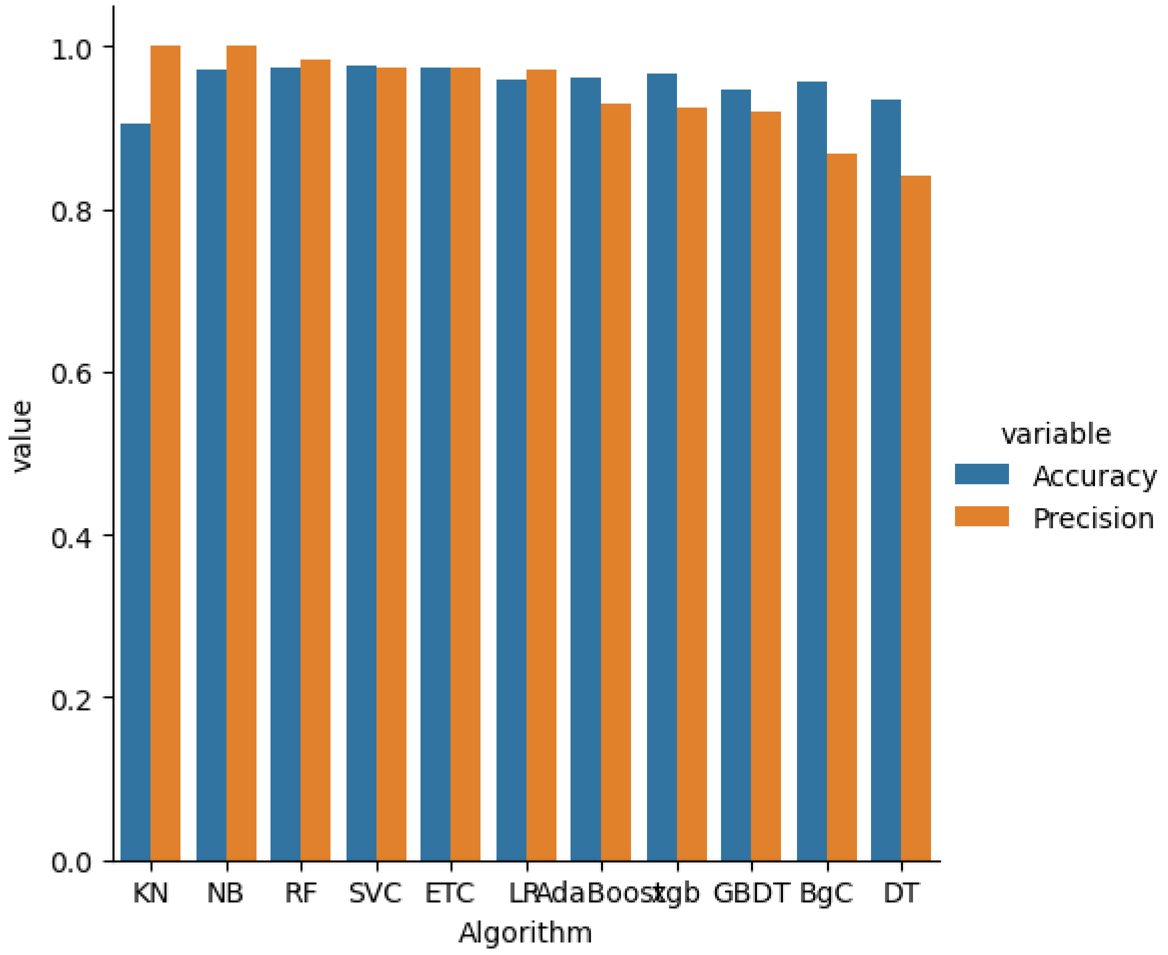
In [90]:
```
performance_df1
```

Out[90]:

	Algorithm	variable	value
0	KN	Accuracy	0.905222
1	NB	Accuracy	0.970986
2	RF	Accuracy	0.974855
3	SVC	Accuracy	0.975822
4	ETC	Accuracy	0.974855
5	LR	Accuracy	0.958414
6	AdaBoost	Accuracy	0.960348
7	xgb	Accuracy	0.966151
8	GBDT	Accuracy	0.947776
9	BgC	Accuracy	0.957447
10	DT	Accuracy	0.933269
11	KN	Precision	1.000000
12	NB	Precision	1.000000
13	RF	Precision	0.982759
14	SVC	Precision	0.974790
15	ETC	Precision	0.974576
16	LR	Precision	0.970297
17	AdaBoost	Precision	0.929204
18	xgb	Precision	0.925620
19	GBDT	Precision	0.920000
20	BgC	Precision	0.867188
21	DT	Precision	0.841584

In [92]:
```
sns.catplot(x="Algorithm",y="value",
           hue= "variable",data= performance_df1,kind="bar",height=5)
```

Out[92]:
```
<seaborn.axisgrid.FacetGrid at 0x2306a6a4ac0>
```



In [93]:
```
# model improve
#1. Change the max_features parameter of TfIdf
```

In [94]:
```
temp_df= pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_max_ft_3000':accuracy_scores,'Precision_max_ft_3000': precisi
```

In [95]:
```
new_df= performance_df.merge(temp_df,on= 'Algorithm')
```

In [96]:
```
new_df_scaled= new_df.merge(temp_df,on= 'Algorithm')
```

In [97]:
```
temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_num_chars':accuracy_scores,'Precision_num_chars': precision_
```

In [98]:
```
new_df_scaled.merge(temp_df,on= 'Algorithm')
```

Out[98]:

	Algorithm	Accuracy	Precision	Accuracy_max_ft_3000_x	Precision_max_ft_3000_x	Accuracy_max_ft_3000_y	Precision_max_ft_3000_y	Accura
0	KN	0.905222	1.000000	0.905222	1.000000	0.905222	1.000000	
1	NB	0.970986	1.000000	0.970986	1.000000	0.970986	1.000000	
2	RF	0.974855	0.982759	0.974855	0.982759	0.974855	0.982759	
3	SVC	0.975822	0.974790	0.975822	0.974790	0.975822	0.974790	
4	ETC	0.974855	0.974576	0.974855	0.974576	0.974855	0.974576	
5	LR	0.958414	0.970297	0.958414	0.970297	0.958414	0.970297	
6	AdaBoost	0.960348	0.929204	0.960348	0.929204	0.960348	0.929204	
7	xgb	0.966151	0.925620	0.966151	0.925620	0.966151	0.925620	
8	GBDT	0.947776	0.920000	0.947776	0.920000	0.947776	0.920000	
9	BgC	0.957447	0.867188	0.957447	0.867188	0.957447	0.867188	
10	DT	0.933269	0.841584	0.933269	0.841584	0.933269	0.841584	

In [103]:
```
#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 [104]:
```
voting = VotingClassifier(estimators=[('svm',svc),('nb',mnb),('et',etc)],voting='soft')
```

In [105]:
```
voting.fit(X_train,y_train)
```

Out[105]:
```
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 [106]:
```
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 [107]:
```
print('hello')
```

```
hello
```

```
In [ ]:
```