In [58]:
```
pip install wordcloud
```

```
Defaulting to user installation because normal site-packages is not writeableNote: you m
ay need to restart the kernel to use updated packages.
  WARNING: The script wordcloud_cli.exe is installed in 'C:\Users\amits\AppData\Roaming
\Python\Python310\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use
--no-warn-script-location.
Collecting wordcloud
  Downloading wordcloud-1.9.2-cp310-cp310-win_amd64.whl (152 kB)
     ---------------------------------------- 152.1/152.1 kB 1.1 MB/s eta 0:00:00
Requirement already satisfied: pillow in d:\anaoncda\lib\site-packages (from wordcloud)
(9.4.0)
Requirement already satisfied: matplotlib in d:\anaoncda\lib\site-packages (from wordclo
ud) (3.7.0)
Requirement already satisfied: numpy>=1.6.1 in d:\anaoncda\lib\site-packages (from wordc
loud) (1.23.5)
Requirement already satisfied: cycler>=0.10 in d:\anaoncda\lib\site-packages (from matpl
otlib->wordcloud) (0.11.0)
Requirement already satisfied: pyparsing>=2.3.1 in d:\anaoncda\lib\site-packages (from m
atplotlib->wordcloud) (3.0.9)
Requirement already satisfied: packaging>=20.0 in d:\anaoncda\lib\site-packages (from ma
tplotlib->wordcloud) (22.0)
Requirement already satisfied: fonttools>=4.22.0 in d:\anaoncda\lib\site-packages (from
matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: python-dateutil>=2.7 in d:\anaoncda\lib\site-packages (fr
om matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in d:\anaoncda\lib\site-packages (from
matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: contourpy>=1.0.1 in d:\anaoncda\lib\site-packages (from m
atplotlib->wordcloud) (1.0.5)
Requirement already satisfied: six>=1.5 in d:\anaoncda\lib\site-packages (from python-da
teutil>=2.7->matplotlib->wordcloud) (1.16.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.2
```

In [117…
```
pip install xgboost
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting xgboost
  Downloading xgboost-1.7.5-py3-none-win_amd64.whl (70.9 MB)
     ---------------------------------------- 70.9/70.9 MB 2.6 MB/s eta 0:00:00
Requirement already satisfied: scipy in d:\anaoncda\lib\site-packages (from xgboost) (1.
10.0)
Requirement already satisfied: numpy in d:\anaoncda\lib\site-packages (from xgboost) (1.
23.5)
Installing collected packages: xgboost
Successfully installed xgboost-1.7.5
Note: you may need to restart the kernel to use updated packages.
```

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

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

Out[2]:

	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 [3]:
```
df.shape
```

Out[3]:
```
(5572, 5)
```

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
v1              0
v2              0
Unnamed: 2    5522
Unnamed: 3    5560
Unnamed: 4    5566
dtype: int64
```

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

Out[5]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
74	ham	U can call me now...	NaN	NaN	NaN
4470	ham	Wa... U so efficient... Gee... Thanx...	NaN	NaN	NaN
3753	spam	Bloomberg -Message center +447797706009 Why wa...	NaN	NaN	NaN
2227	ham	Oh k.k..where did you take test?	NaN	NaN	NaN
5077	spam	Do you want a New Nokia 3510i colour phone Del...	NaN	NaN	NaN

1.Data cleaning

In [6]:
```
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 [7]:
```
df.drop(columns = ['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace = True)
```

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

Out[8]:

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

In [9]:
```
df.rename(columns={'v1':'target','v2':'text'},inplace=True)
df.sample(5)
```

Out[9]:

	target	text
1071	spam	URGENT! We are trying to contact U. Todays dra...
4343	ham	Hi:)did you asked to waheeda fathima about leave?
353	ham	Yo you guys ever figure out how much we need f...
4390	ham	The greatest test of courage on earth is to be...
4696	ham	Okey dokey, i□Û÷ll be over in a bit just sorti...

In [10]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

In [11]:
```
df['target'] = encoder.fit_transform(df['target'])
```

In [12]:
```
df.head()
```

Out[12]:

	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 [13]:
```
df.isnull().sum()
```

Out[13]:
```
target    0
text      0
dtype: int64
```

In [14]:
```
# check for duplicated values
df.duplicated().sum()
```

Out[14]:
```
403
```

In [15]:
```
df = df.drop_duplicates(keep = 'first')
```

2.EDA

In [16]:
```
df.head()
```

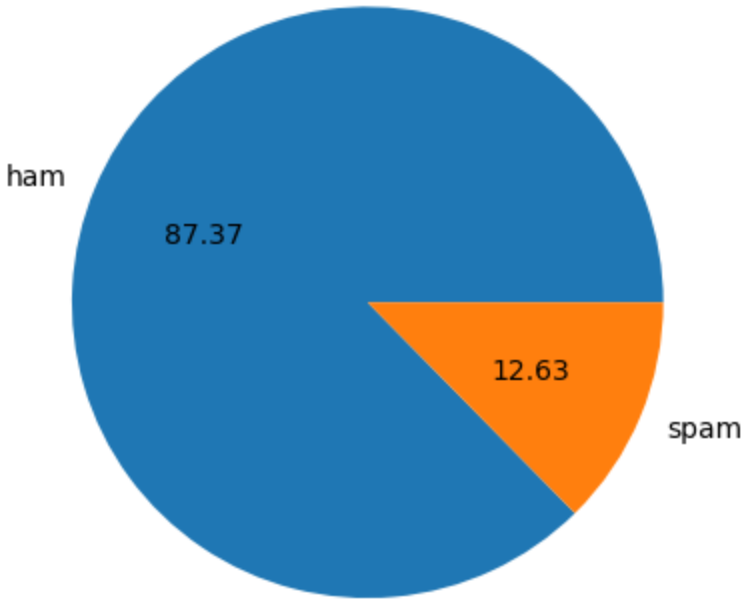
Out[16]:

	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 [17]:
```
df['target'].value_counts()
```

Out[17]:
```
0    4516
1     653
Name: target, dtype: int64
```

In [18]:
```
plt.pie(df['target'].value_counts(),labels = ['ham','spam'],autopct="%0.2f")
plt.show()
```



In [19]:
```
# Data is imbalanced
import nltk
```

In [20]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\amits\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	18.453279	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 [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	17.120903	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 [29]:
```
df[df['target']==1][['num_characters','num_words','num_sentences']].describe()
```

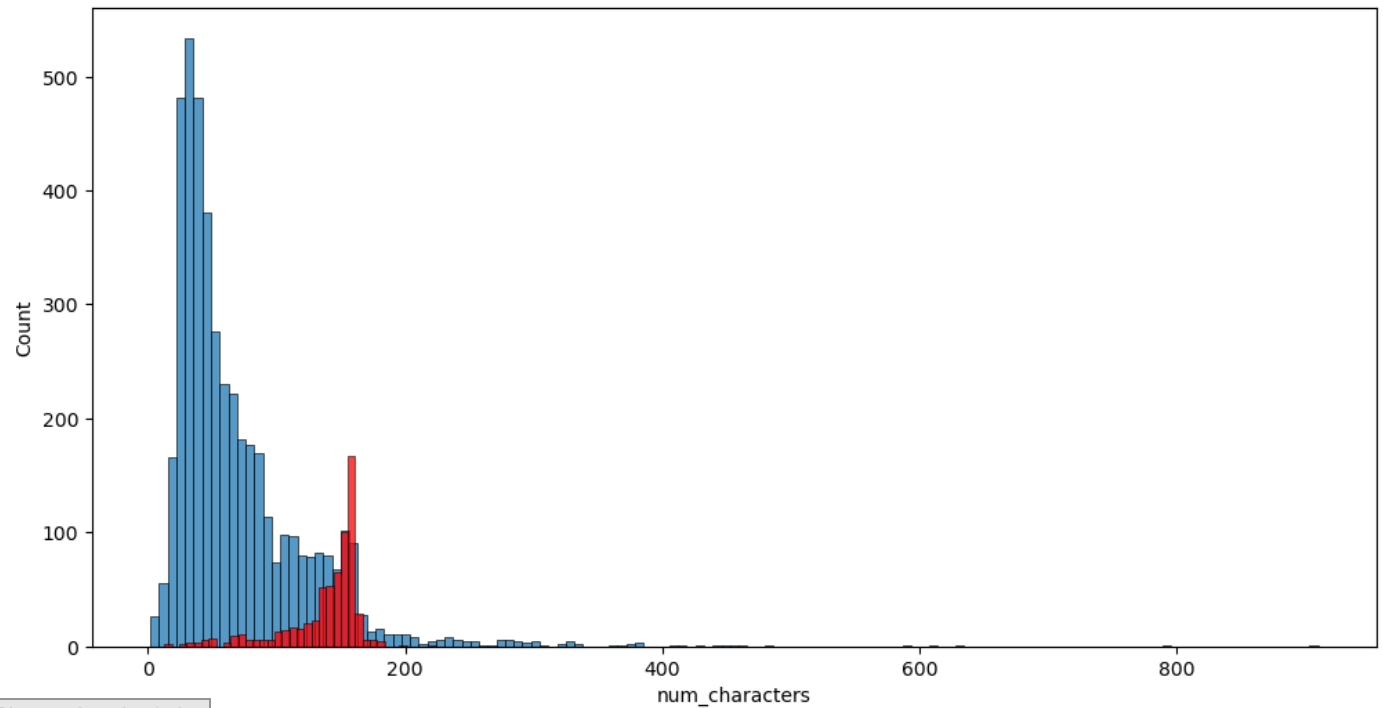
Out[29]:

	num_characters	num_words	num_sentences
count	653.000000	653.000000	653.000000
mean	137.891271	27.667688	2.967841
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 [30]:
```
import seaborn as sns
```

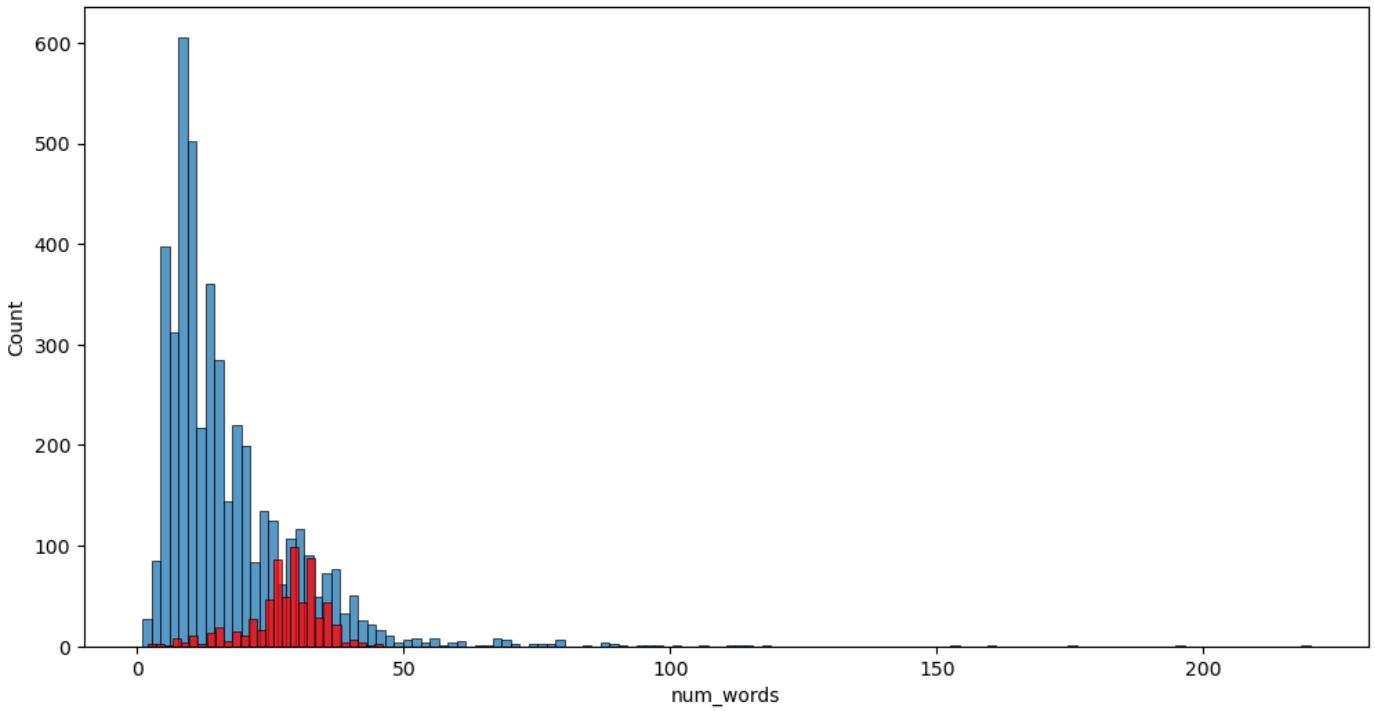
In [31]:
```
plt.figure(figsize=(12,6))
sns.histplot(df[df['target'] == 0]['num_characters'])
sns.histplot(df[df['target'] == 1]['num_characters'],color='red')
```

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



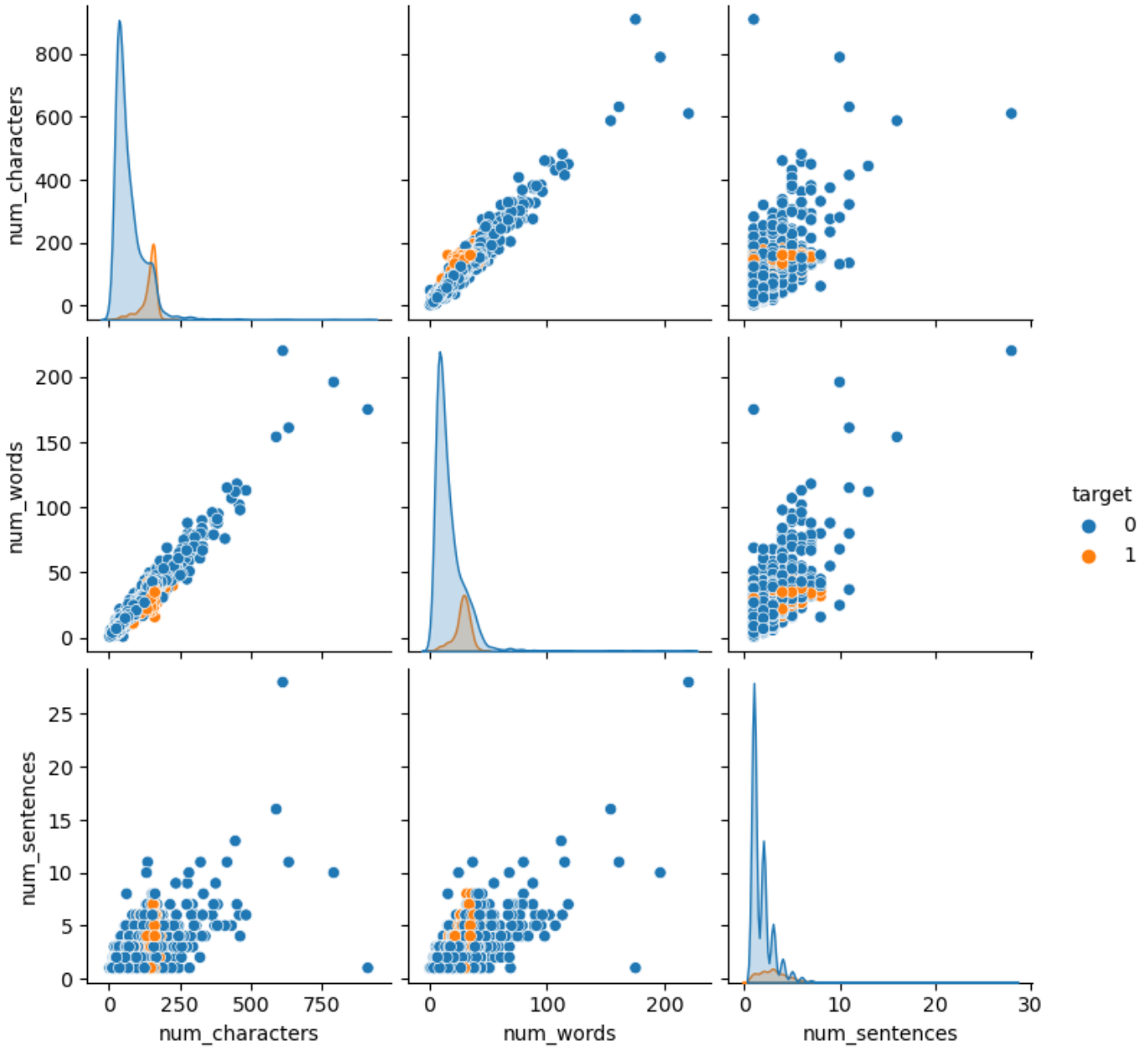
In [32]:
```
plt.figure(figsize=(12,6))
sns.histplot(df[df['target'] == 0]['num_words'])
sns.histplot(df[df['target'] == 1]['num_words'],color='red')
```

Out[32]:
```
<Axes: xlabel='num_words', ylabel='Count'>
```



In [33]:
```
sns.pairplot(df,hue='target')
```

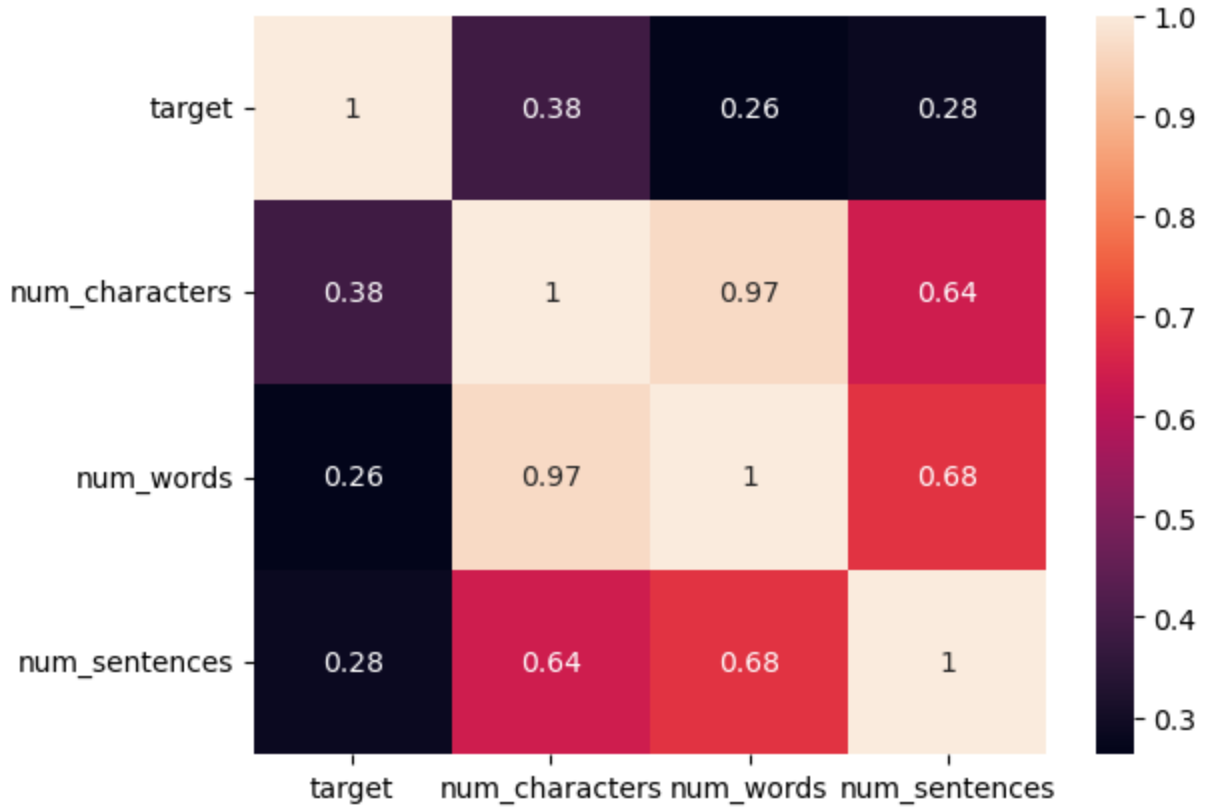
Out[33]:
```
<seaborn.axisgrid.PairGrid at 0x15c87145ff0>
```



In [34]:
```
sns.heatmap(df.corr(),annot = True)
```

```
C:\Users\amits\AppData\Local\Temp\ipykernel_11324\2221401063.py:1: FutureWarning: The de
fault value of numeric_only in DataFrame.corr is deprecated. In a future version, it wil
l default to False. Select only valid columns or specify the value of numeric_only to si
lence this warning.
  sns.heatmap(df.corr(),annot = True)
```

Out[34]:
```
<Axes: >
```



3. Data Preprocessing

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

In [49]:
```
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)
```

```
xt('Go until jurong point, crazy.. Available only in bugis n great world la
```

Out[52]:
```
'go jurong point crazi avail bugi n great world la e buffet cine got amor wat'
```

In [48]:
```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
ps.stem('Dancing')
```

Out[48]:
```
'danc'
```

In [51]:
```
df['text'][0]
```

Out[51]:
```
'Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cin
e there got amore wat...'
```

In [53]:
```
df['transform_text'] = df['text'].apply (transform_text)
```

In [54]:
```
df.head()
```

Out[54]:

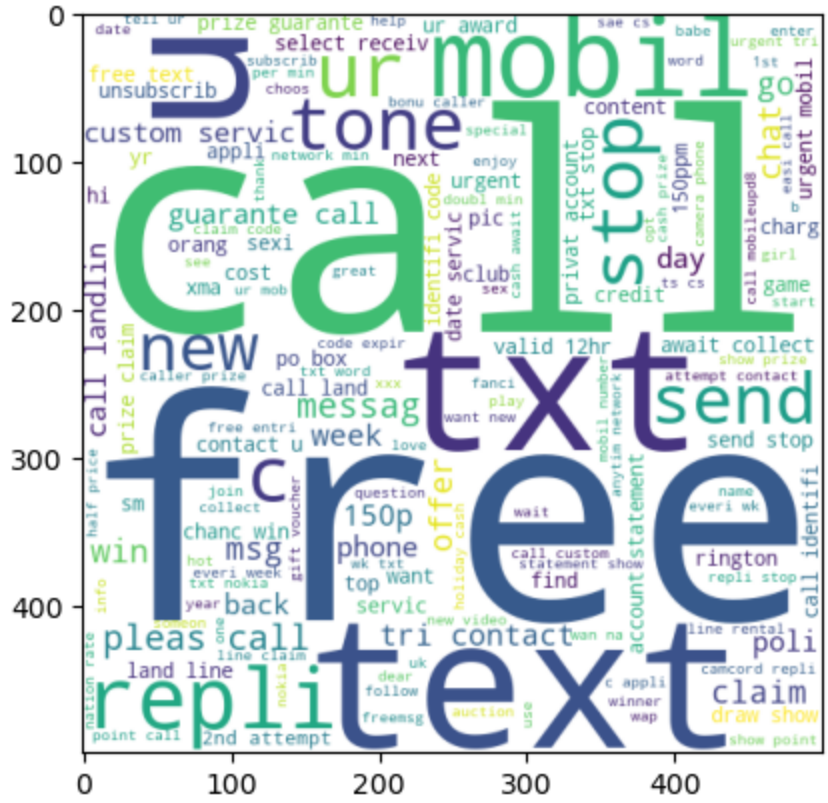
	target	text	num_characters	num_words	num_sentences	transform_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	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 [59]:
```
from wordcloud import WordCloud
wc = WordCloud(width=500,height=500,min_font_size=10,background_color='white')
```

In [60]:
```
spam_wc = wc.generate(df[df['target']==1]['transform_text'].str.cat(sep = " "))
```

In [64]:
```
 plt.imshow(spam_wc)
```

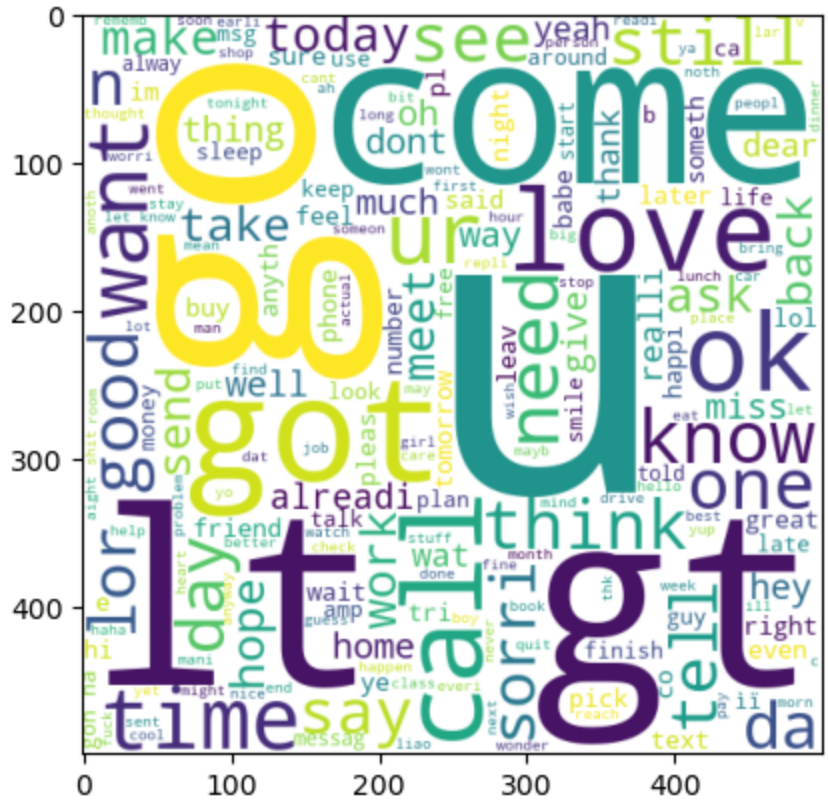
Out[64]:
```
<matplotlib.image.AxesImage at 0x15c8d8af580>
```



In [65]:
```
ham_wc = wc.generate(df[df['target']==0]['transform_text'].str.cat(sep = " "))
```

In [66]:
```
plt.imshow(ham_wc)
```

Out[66]:
```
<matplotlib.image.AxesImage at 0x15c8cc49b70>
```



In [67]:
```
df.head()
```

Out[67]:

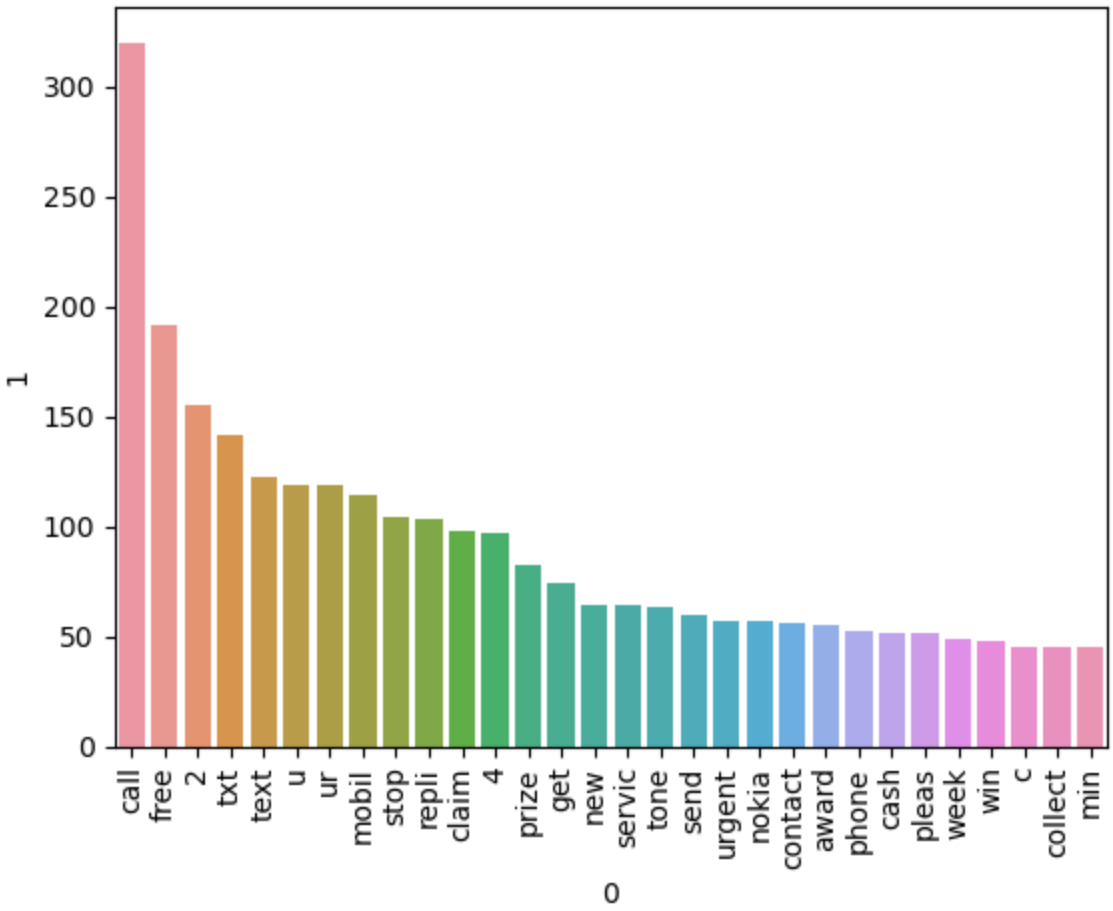
	target	text	num_characters	num_words	num_sentences	transform_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	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 [70]:
```
spam_corpus = []
for msg in df[df['target']==1]['transform_text'].tolist():
    for word in msg.split():
        spam_corpus.append(word)
```

In [71]:
```
len(spam_corpus)
```

Out[71]:
```
9939
```

In [80]:
```
from collections import Counter
sns.barplot(x=pd.DataFrame(Counter(spam_corpus).most_common(30))[0], y=pd.DataFrame(Coun
plt.xticks(rotation='vertical')
plt.show()
```

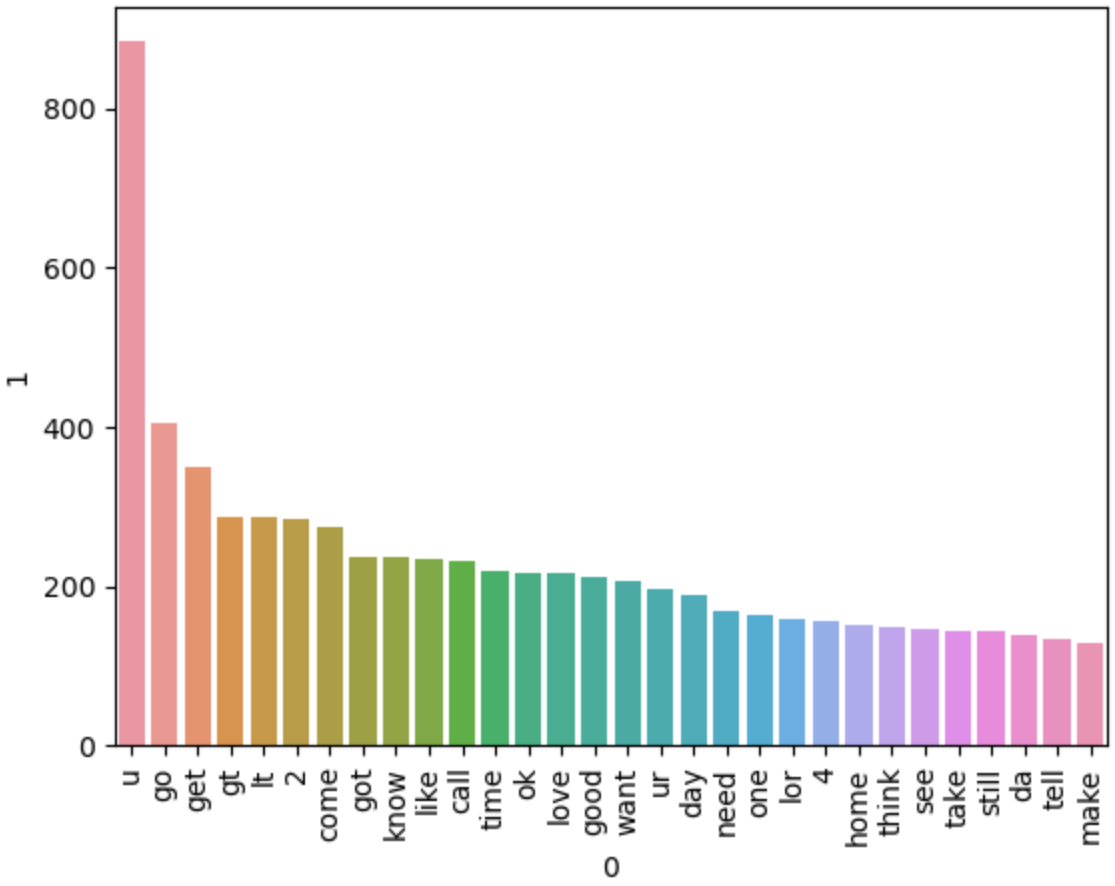


In [85]:
```
ham_corpus = []
for msg in df[df['target'] == 0]['transform_text'].tolist():
    for word in msg.split():
        ham_corpus.append(word)
```

In [86]:
```
len(ham_corpus)
```

Out[86]:
```
35394
```

In [87]:
```
from collections import Counter
sns.barplot(x=pd.DataFrame(Counter(ham_corpus).most_common(30))[0], y=pd.DataFrame(Count
plt.xticks(rotation='vertical')
plt.show()
```



4.MODEL BUILDING

In [166…
```
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
cv = CountVectorizer()
tfidf = TfidfVectorizer(max_features=3000)
```

In [167…
```
X = tfidf.fit_transform(df['transform_text']).toarray()
```

In [168…
```
X
```

Out[168]:
```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

In [169…
```
X.shape
```

Out[169]:
```
(5169, 3000)
```

In [170…
```
y = df['target'].values
```

In [171…
```
y
```

Out[171]:
```
array([0, 0, 1, ..., 0, 0, 0])
```

In [172…
```
from sklearn.model_selection import train_test_split
```

In [173…
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.2,random_state = 2)
```

In [174…
```
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

In [175…
```
gnb = GaussianNB()
mnb = MultinomialNB()
bnb = BernoulliNB()
```

In [176…
```
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 [177…
```
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 [115…
```
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 [ ]:
```
# tfidf - MNB
```

In [118…
```
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 [119…
```
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 [120…
```
clfs = {
    'SVC' : svc,
    'KN' : knc,
    'NB': mnb,
    'DT': dtc,
    'LR': lrc,
    'RF': rfc,
    'AdaBoost': abc,
    'BgC': bc,
    'ETC': etc,
    'GBDT':gbdt,
    'xgb':xgb
}
```

In [121…
```
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 [122…
```
train_classifier(svc,X_train,y_train,X_test,y_test)
```

Out[122]:
```
(0.9758220502901354, 0.9747899159663865)
```

In [123…
```
accuracy_scores = []
precision_scores = []

for name,clf in clfs.items():

    current_accuracy,current_precision = train_classifier(clf, X_train,y_train,X_test,y_

    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 -  0.9274661508704062
Precision -  0.8118811881188119
For  LR
Accuracy -  0.9584139264990329
Precision -  0.9702970297029703
For  RF
Accuracy -  0.9748549323017408
Precision -  0.9827586206896551
For  AdaBoost
Accuracy -  0.960348162475822
Precision -  0.9292035398230089
For  BgC
Accuracy -  0.9574468085106383
Precision -  0.8671875
For  ETC
Accuracy -  0.9748549323017408
Precision -  0.9745762711864406
For  GBDT
Accuracy -  0.9477756286266924
Precision -  0.92
For  xgb
Accuracy -  0.971953578336557
Precision -  0.943089430894309
```

In [124…
```
performance_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy':accuracy_scores,'Preci
```

In [125…
```
performance_df
```

Out[125]:

	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
10	xgb	0.971954	0.943089
6	AdaBoost	0.960348	0.929204
9	GBDT	0.947776	0.920000
7	BgC	0.957447	0.867188
3	DT	0.927466	0.811881

Model Improve

In [130…
```
df.head()
```

Out[130]:

	target	text	num_characters	num_words	num_sentences	transform_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 [147…
```
temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_max_ft_3000':accuracy_scores,'
```

In [148…
```
temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_scaling':accuracy_scores,'Prec
```

In [149…
```
new_df = performance_df.merge(temp_df,on='Algorithm')
```

In [150…
```
new_df_scaled = new_df.merge(temp_df,on='Algorithm')
```

In [151…
```
temp_df = pd.DataFrame({'Algorithm':clfs.keys(),'Accuracy_num_chars':accuracy_scores,'Pr
```

In [152…
```
new_df_scaled.merge(temp_df,on='Algorithm')
```

Out[152]:

	Algorithm	Accuracy	Precision	Accuracy_scaling_x	Precision_scaling_x	Accuracy_scaling_y	Precision_sc
0	KN	0.905222	1.000000	0.905222	1.000000	0.905222	1.
1	NB	0.970986	1.000000	0.970986	1.000000	0.970986	1.
2	RF	0.974855	0.982759	0.974855	0.982759	0.974855	0.
3	SVC	0.975822	0.974790	0.975822	0.974790	0.975822	0.
4	ETC	0.974855	0.974576	0.974855	0.974576	0.974855	0.
5	LR	0.958414	0.970297	0.958414	0.970297	0.958414	0.
6	xgb	0.971954	0.943089	0.971954	0.943089	0.971954	0.
7	AdaBoost	0.960348	0.929204	0.960348	0.929204	0.960348	0.
8	GBDT	0.947776	0.920000	0.947776	0.920000	0.947776	0.
9	BgC	0.957447	0.867188	0.957447	0.867188	0.957447	0.
10	DT	0.927466	0.811881	0.927466	0.811881	0.927466	0.

Voting classifier

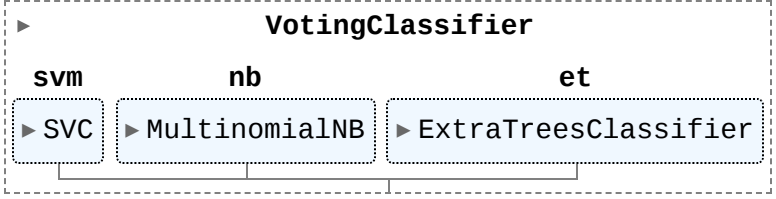
In [156…
```
svc = SVC(kernel = 'sigmoid',gamma = 1.0,probability = True)
mnb = MultinomialNB()
etc = ExtraTreesClassifier(n_estimators=50, random_state=2)

from sklearn.ensemble import VotingClassifier
```

In [157…
```
voting = VotingClassifier(estimators = [('svm', svc), ('nb', mnb), ('et', etc)],voting='
```

In [158…
```
voting.fit(X_train,y_train)
```

Out[158]:



In [159…
```
y_pred = voting.predict(X_test)
print("Accuracy",accuracy_score(y_test,y_pred))
print("Precision",precision_score(y_test,y_pred))
```

```
Accuracy 0.9738878143133463
Precision 1.0
```

Applying stacking

In [160…
```
estimators=[('svm', svc), ('nb', mnb), ('et', etc)]
final_estimator=RandomForestClassifier()
```

In [161…
```
from sklearn.ensemble import StackingClassifier
```

In [162…
```
clf = StackingClassifier(estimators=estimators, final_estimator=final_estimator)
```

In [165…
```
# clf.fit(X_train,y_train)
# y_pred = clf.predict(X_test)
# print("Accuracy",accuracy_score(y_test,y_pred))
# print("Precision",precision_score(y_test,y_pred))
```

In [180…
```
import pickle
pickle.dump(tfidf,open('vectorizer.pkl','wb'))
pickle.dump(mnb,open('model.pkl','wb'))
```