In [1]:

```python
import pandas as pd
import pandas as pd
```

In [2]:

```python
a=pd.read_csv(r"C:\Users\SHRADDHA TRIPATHI\OneDrive\Desktop\datasets\sp
```

In [3]:

```python
a.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 [4]:

```python
a.sample(10)
```

Out[4]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
1965	ham	Thanks. It was only from tescos but quite nice...	NaN	NaN	NaN
4416	ham	says that he's quitting at least5times a day ...	NaN	NaN	NaN
4881	ham	As usual u can call me ard 10 smth.	NaN	NaN	NaN
517	spam	Your credits have been topped up for http://ww...	NaN	NaN	NaN
4328	ham	1Apple/Day=No Doctor. 1Tulsi Leaf/Day=No Cance...	NaN	NaN	NaN
66	ham	Today is \song dedicated day..\" Which song wi...	NaN	NaN	NaN
4424	ham	Just now saw your message.it k da:)	NaN	NaN	NaN
1921	ham	No current and food here. I am alone also	NaN	NaN	NaN
1100	ham	You busy or can I come by at some point and fi...	NaN	NaN	NaN
4187	ham	Mm umma ask vava also to come tell him can pla...	NaN	NaN	NaN

In [5]:

```python
a.shape
```

Out[5]:

```
(5572, 5)
```

In [6]:

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

In [7]:

```
a.sample(5)
```

Out[7]:

	v1	v2
4693	ham	Pls give her the food preferably pap very slow...
2924	ham	Are you coming to day for class.
1218	ham	K..k..i'm also fine:)when will you complete th...
2315	ham	That's significant but dont worry.
3154	ham	Ok...

In [8]:

```
a.rename(columns={'v1':'target','v2':'data'},inplace=True)
a.sample(4)
```

Out[8]:

	target	data
1157	ham	He's an adult and would learn from the experie...
5390	ham	Nt joking seriously i told
2290	ham	HEY THERE BABE, HOW U DOIN? WOT U UP 2 2NITE L...
5052	ham	Lmao you know me so well...

In [9]:

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

In [10]:

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

In [11]:

```
a.head()
```

Out[11]:

	target	data
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]:

```
a.isnull().sum()
```

Out[12]:

```
target    0
data      0
dtype: int64
```

In [13]:

```
a.duplicated().sum()
```

Out[13]:

```
403
```

In [14]:

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

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

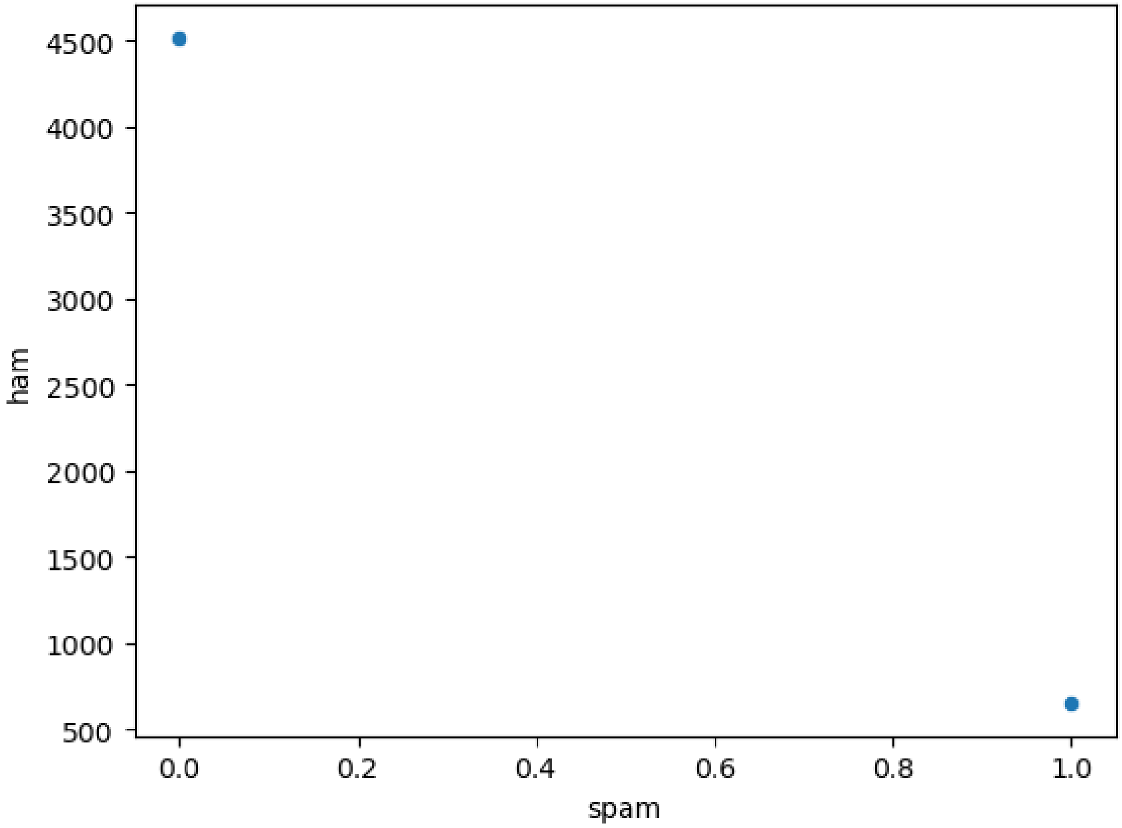
Out[15]:
```
0
```

In [16]:
```
## data analysis
```

In [17]:
```
a['target'].value_counts()
```

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

In [18]:
```
import matplotlib.pyplot as plt
import seaborn as sns
sns.scatterplot(a['target'].value_counts())
plt.xlabel('spam')
plt.ylabel('ham')
plt.show()
```

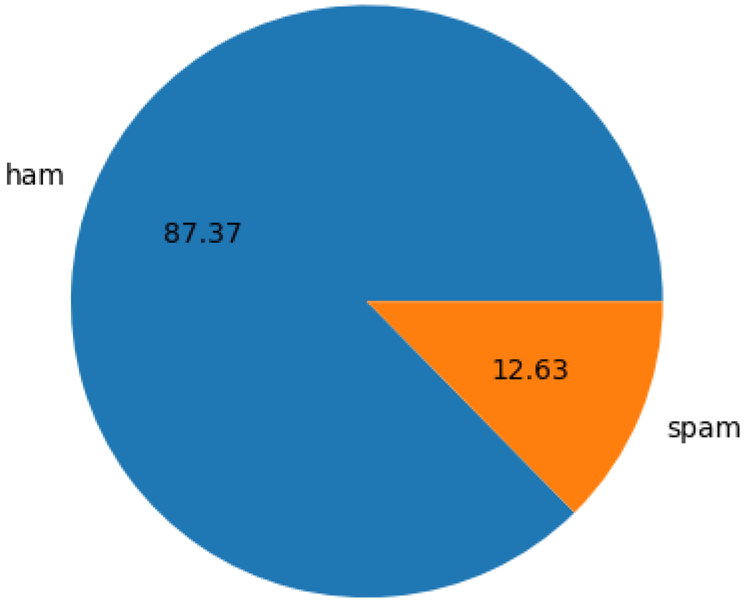


In [19]:

```
plt.pie(a['target'].value_counts(),labels=['ham','spam'],autopct="%0.2f
```

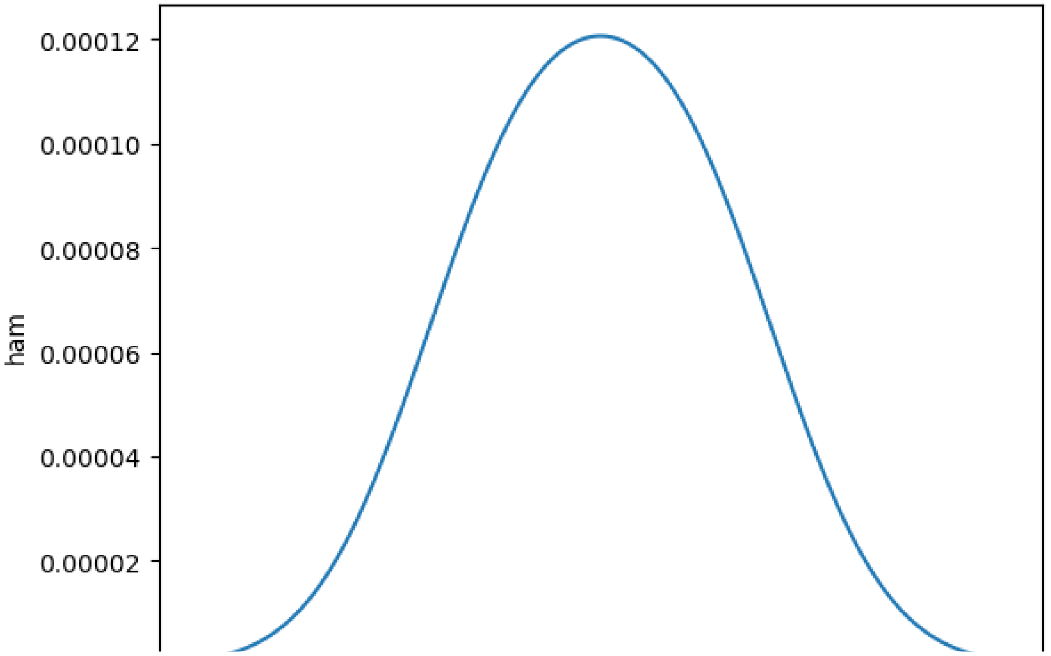
Out[19]:

```
([<matplotlib.patches.Wedge at 0x1d1c8cd5450>,
  <matplotlib.patches.Wedge at 0x1d1c8cd5360>],
 [Text(-1.0144997251399075, 0.4251944351600247, 'ham'),
  Text(1.014499764949479, -0.4251943401757036, 'spam')],
 [Text(-0.5533634864399495, 0.23192423736001344, '87.37'),
  Text(0.5533635081542612, -0.23192418555038377, '12.63')])
```



In [20]:

```
sns.kdeplot(a['target'].value_counts())
plt.xlabel('spam')
plt.ylabel('ham')
plt.show()
```



In [21]:

```
import nltk
```

In [22]:

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

```
[nltk_data] Error loading punkt: <urlopen error [WinError 10060] A
[nltk_data]     connection attempt failed because the connected party
[nltk_data]     did not properly respond after a period of time, or
[nltk_data]     established connection failed because connected host
[nltk_data]     has failed to respond>
```

Out[22]:

```
False
```

In [23]:

```
a['data']
```

Out[23]:

```
0       Go until jurong point, crazy.. Available only ...
1                           Ok lar... Joking wif u oni...
2       Free entry in 2 a wkly comp to win FA Cup fina...
3       U dun say so early hor... U c already then say...
4       Nah I don't think he goes to usf, he lives aro...
                              ...
5567    This is the 2nd time we have tried 2 contact u...
5568                Will Ì_ b going to esplanade fr home?
5569    Pity, * was in mood for that. So...any other s...
5570    The guy did some bitching but I acted like i'd...
5571                           Rofl. Its true to its name
Name: data, Length: 5169, dtype: object
```

In [24]:

```
a['data'].apply(len)## discribe total number of charecter in a row
```

Out[24]:

```
0       111
1        29
2       155
3        49
4        61
       ...
5567    161
5568     37
5569     57
5570    125
5571     26
Name: data, Length: 5169, dtype: int64
```

In [25]:

```
a['total_num_char']=a['data'].apply(len)
```

In [26]:

```
a.head()
```

Out[26]:

	target	data	total_num_char
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 [27]:
```
a['data'].apply(lambda x:nltk.word_tokenize(x))
```

Out[27]:
```
0       [Go, until, jurong, point, ,, crazy, .., Avail...
1                [Ok, lar, ..., Joking, wif, u, oni, ...]
2       [Free, entry, in, 2, a, wkly, comp, to, win, F...
3       [U, dun, say, so, early, hor, ..., U, c, alrea...
4       [Nah, I, do, n't, think, he, goes, to, usf, ,,...
                              ...
5567    [This, is, the, 2nd, time, we, have, tried, 2,...
5568     [Will, Ì_, b, going, to, esplanade, fr, home, ?]
5569    [Pity, ,, *, was, in, mood, for, that, ., So, ...
5570    [The, guy, did, some, bitching, but, I, acted,...
5571                  [Rofl, ., Its, true, to, its, name]
Name: data, Length: 5169, dtype: object
```

In [28]:
```
a['total_words']=a['data'].apply(lambda b:len(nltk.word_tokenize(b)))
```

In [29]:
```
a.head()
```

Out[29]:

	target	data	total_num_char	total_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 [30]:
```
a['total_sentence']=a['data'].apply(lambda c:len(nltk.sent_tokenize(c))
```

In [31]:
```
a.head()
```

Out[31]:

	target	data	total_num_char	total_words	total_sentence
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 [32]:

```
a[['total_num_char','total_words','total_sentence']].describe()
```

Out[32]:

	total_num_char	total_words	total_sentence
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 [33]:

```
a[a['target']==0]## 0=ham
```

Out[33]:

	target	data	total_num_char	total_words	total_sentence
0	0	Go until jurong point, crazy.. Available only ...	111	24	2
1	0	Ok lar... Joking wif u oni...	29	8	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
6	0	Even my brother is not like to speak with me. ...	77	18	2
...	...	...	...	...	...
5565	0	Huh y lei...	12	4	1
5568	0	Will Ì_ b going to esplanade fr home?	37	9	1
5569	0	Pity, * was in mood for that. So...any other s...	57	15	2
5570	0	The guy did some bitching but I acted like i'd...	125	27	1
5571	0	Rofl. Its true to its name	26	7	2

4516 rows × 5 columns

In [34]:

```
a[a['target']==0][['total_num_char','total_words','total_sentence']].de
```

Out[34]:

	total_num_char	total_words	total_sentence
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 [35]:

```


a[a['target']==1][['total_num_char','total_words','total_sentence']].de
```

Out[35]:

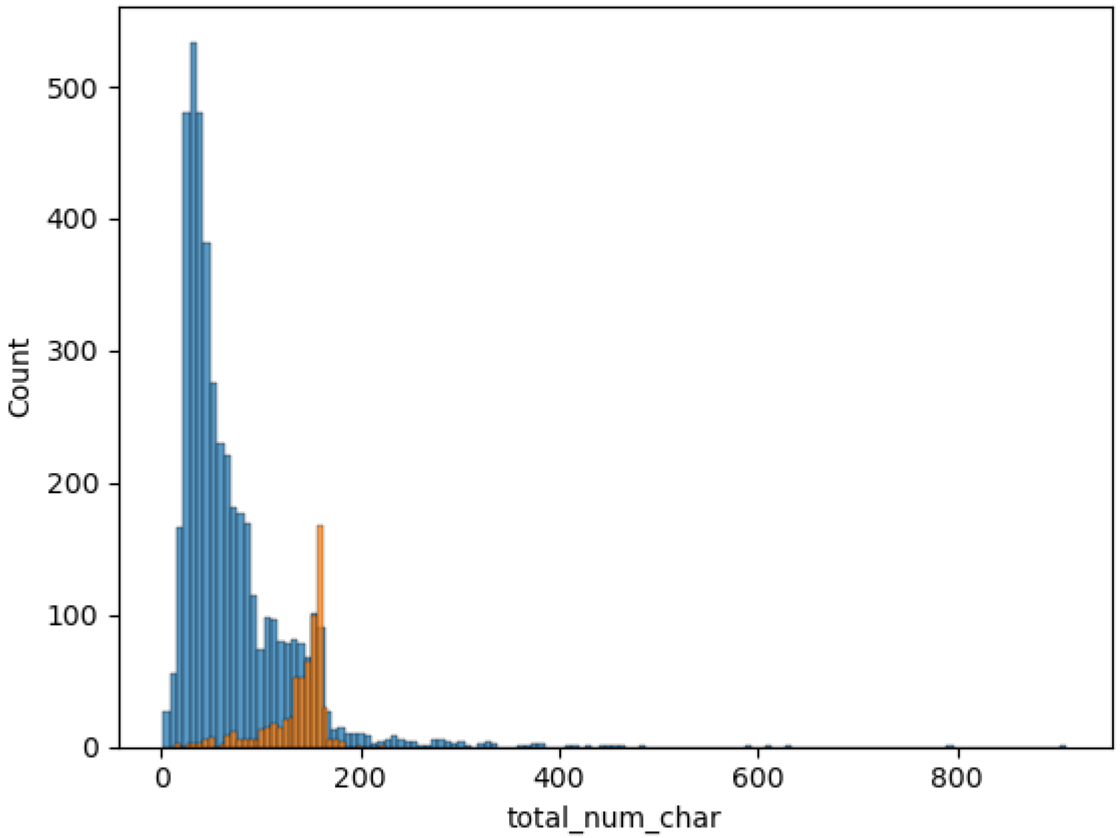
	total_num_char	total_words	total_sentence
count	653.000000	653.000000	653.000000
mean	137.891271	27.667688	2.967841
std	30.137753	7.008418	1.483201
min	13.000000	2.000000	1.000000
25%	132.000000	25.000000	2.000000
50%	149.000000	29.000000	3.000000
75%	157.000000	32.000000	4.000000
max	224.000000	46.000000	8.000000

In [36]:

```
sns.histplot(a[a['target']==0]['total_num_char'])
sns.histplot(a[a['target']==1]['total_num_char'])
```

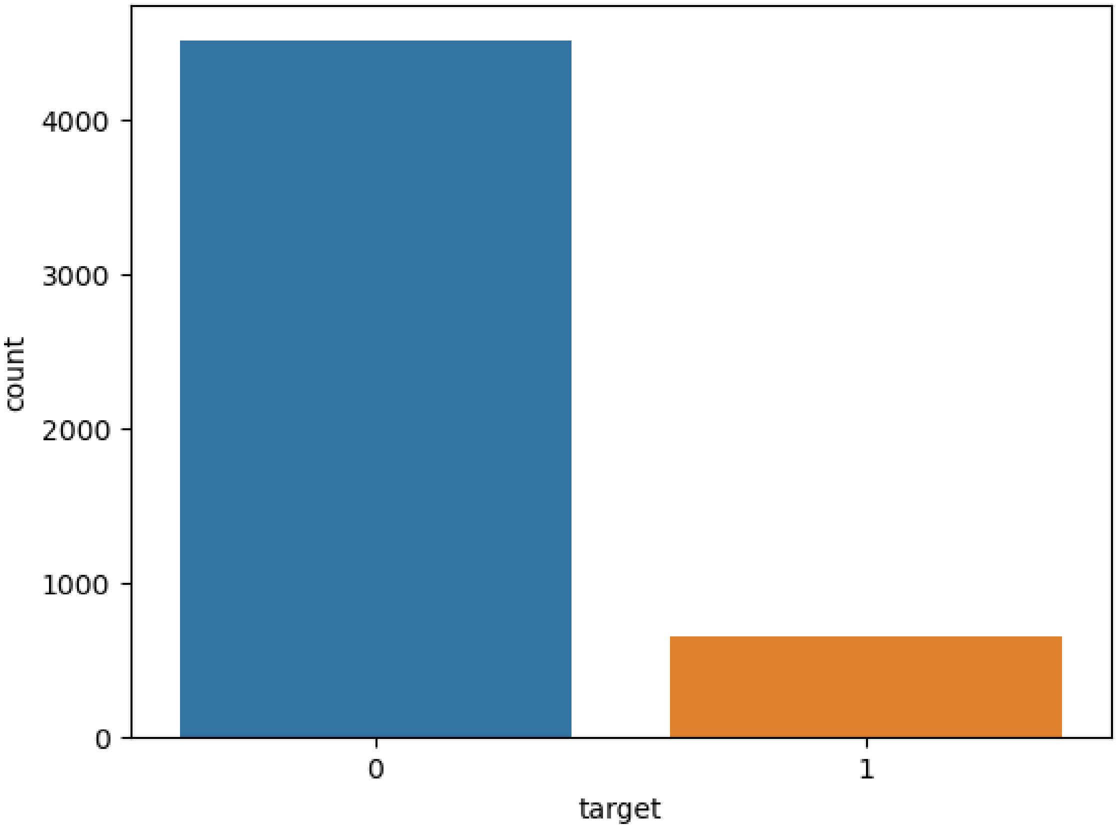
Out[36]:

```
<Axes: xlabel='total_num_char', ylabel='Count'>
```



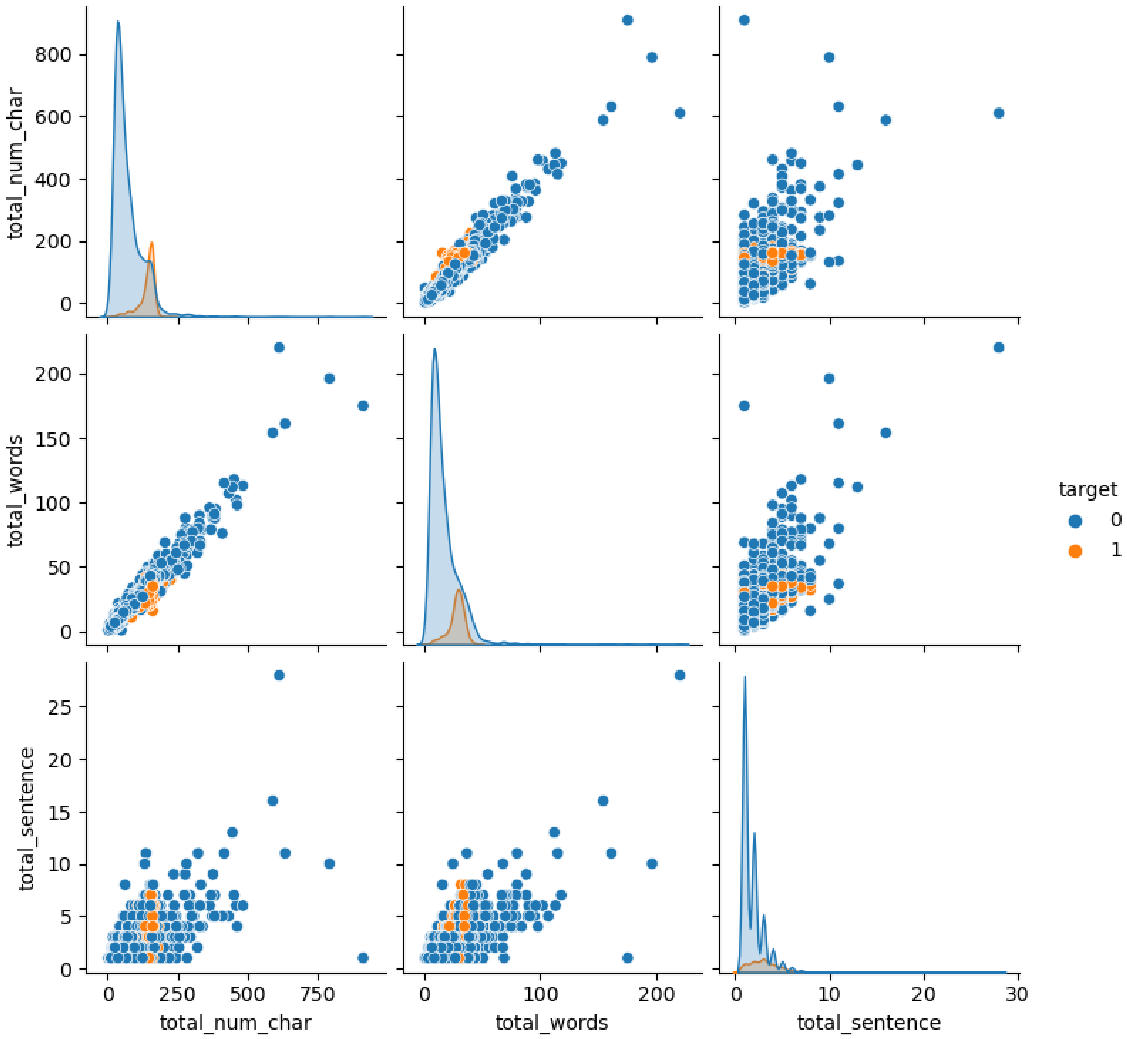
```
In [37]: sns.countplot(x=a['target'])
```

```
Out[37]: <Axes: xlabel='target', ylabel='count'>
```



```
In [38]: sns.pairplot(a,hue='target')
```

```
Out[38]: <seaborn.axisgrid.PairGrid at 0x1d1ce4069e0>
```



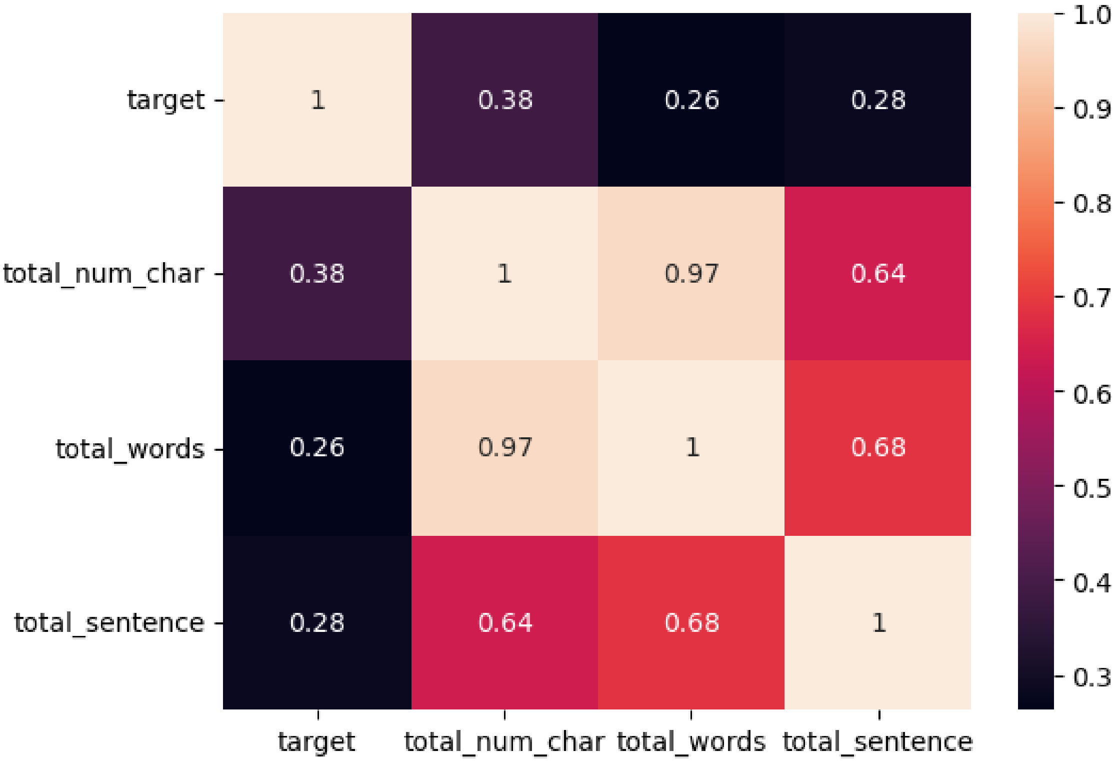
In [39]:

```
sns.heatmap(a.corr(),annot=True)
```

```
C:\Users\SHRADDHA TRIPATHI\AppData\Local\Temp\ipykernel_20116\3595583958.p
y:1: FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only val
id columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(a.corr(),annot=True)
```

Out[39]:

```
<Axes: >
```



In [44]:

```
import nltk
def transform(a):
    a=a.lower()
    a=nltk.word_tokenize(a)
    b=[]
    for i in a:
        if i.isalnum():
            b.append(i)
    a=b[:]
    b.clear()
    for i in a:
        if i not in stopwords.words('english') and i not in string.punc
            b.append(i)
    a=b[:]
    b.clear()
    for i in a:
        b.append(ps.stem(i))

    return"  ".join(b)
```

```
In [53]: transform('"Whats up bruv, hope you had a great break. Do have a reward
```

```
Out[53]: 'what  bruv  hope  great  break  reward  semest'
```

```
In [46]: import nltk
         nltk.download('stopwords')
         
```

```
[nltk_data] Error loading stopwords: <urlopen error [WinError 10060] A
[nltk_data]     connection attempt failed because the connected party
[nltk_data]     did not properly respond after a period of time, or
[nltk_data]     established connection failed because connected host
[nltk_data]     has failed to respond>
```

```
Out[46]: False
```

```
In [47]: from nltk.corpus import stopwords
         stopwords.words('english')
```

```
Out[47]: ['i',
          'me',
          'my',
          'myself',
          'we',
          'our',
          'ours',
          'ourselves',
          'you',
          "you're",
          "you've",
          "you'll",
          "you'd",
          'your',
          'yours',
          'yourself',
          'yourselves',
          'he',
          'him',
```

```
In [48]: import string
         string.punctuation
```

```
Out[48]: '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```
In [49]: a['data'][210]
```

```
Out[49]: "What's up bruv, hope you had a great break. Do have a rewarding semeste
         r."
```

In [50]:

```python
import nltk
nltk.download('PorterStemmer')
```

```
[nltk_data] Error loading PorterStemmer: <urlopen error [WinError
[nltk_data]     10060] A connection attempt failed because the
[nltk_data]     connected party did not properly respond after a
[nltk_data]     period of time, or established connection failed
[nltk_data]     because connected host has failed to respond>
```

Out[50]:

```
False
```

In [51]:

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

Out[51]:

```
'danc'
```

In [52]:

```python
a.head()
```

Out[52]:

	target	data	total_num_char	total_words	total_sentence
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 [54]:

```python
a['transform']=a['data'].apply(transform)
```

In [55]:

```python
a.head()
```

Out[55]:

	target	data	total_num_char	total_words	total_sentence	transform
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n grea...
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 fina...
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 [56]:
```
!pip install WordCloud
```

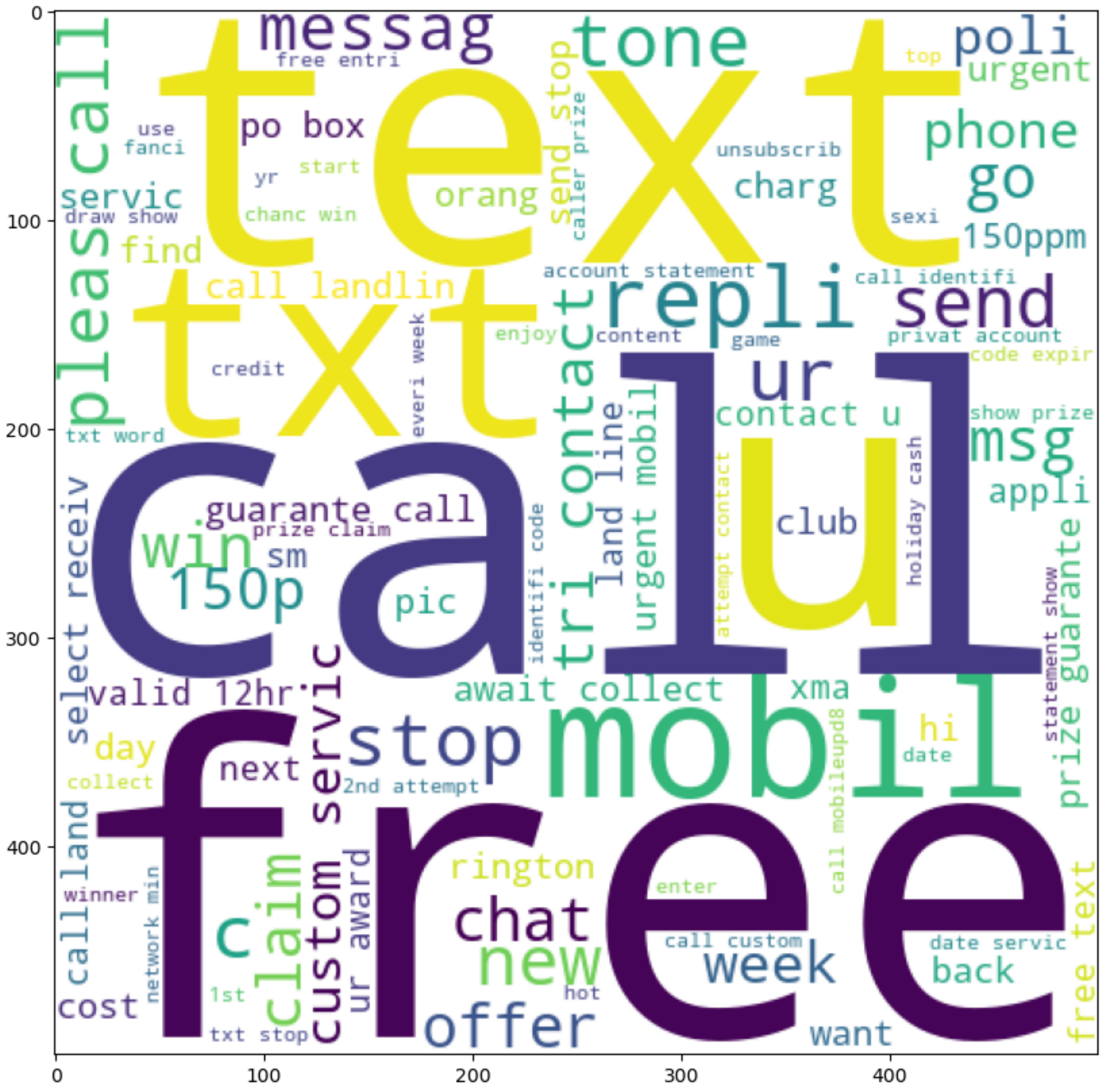
```
mingw32-c++.exe: error: /c: No such file or directory
mingw32-c++.exe: error: pip install WordCloud: No such file or directory
mingw32-c++.exe: fatal error: no input files
compilation terminated.
```

In [57]:
```
from wordcloud import WordCloud
wc=WordCloud(width=500,height=500,min_font_size=10,background_color='wh
```

In [58]:
```
spam_wc=wc.generate(a[a['target']==1]['transform'].str.cat(sep=" "))
```

In [59]:
```
plt.figure(figsize=(20,10))
plt.imshow(spam_wc)
```

Out[59]:
```
<matplotlib.image.AxesImage at 0x1d1cea3a890>
```



In [60]:
```
ham_wc=wc.generate(a[a['target']==0]['transform'].str.cat(sep=" "))
```

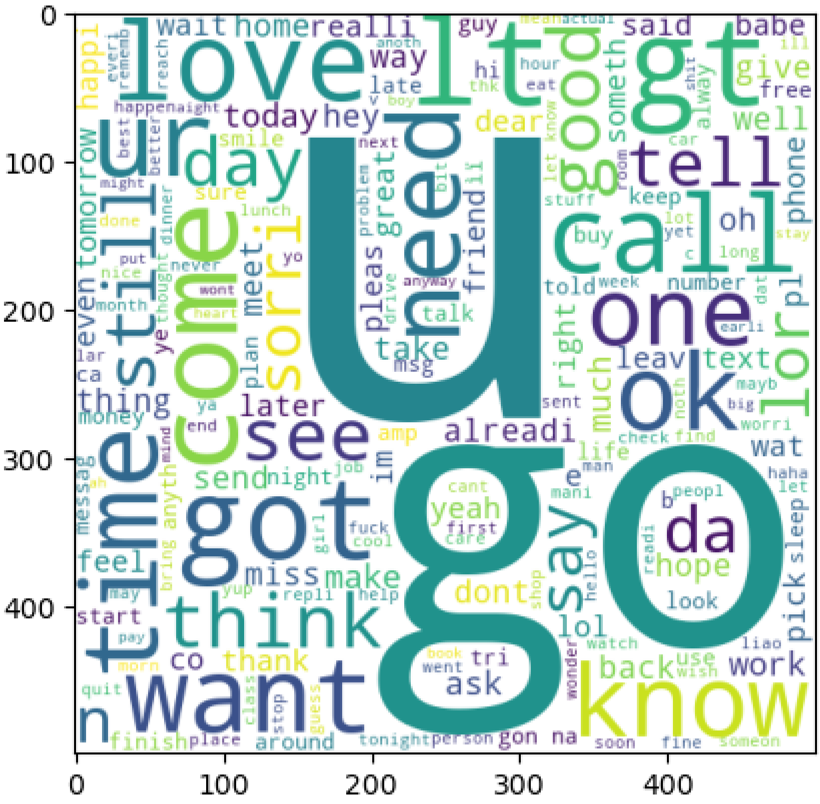
In [61]:

```

plt.imshow(ham_wc)
```

Out[61]:

```
<matplotlib.image.AxesImage at 0x1d1d13da230>
```



In [62]:

```
c=[]
for msg in a[a['target']==1]['transform'].tolist():
    for words in msg.split():
        c.append(words)
```

In [63]:

```
len(c)
```

Out[63]:

```
9939
```

In [64]:

```
from collections import Counter
Counter(c)
```

Out[64]:

```
Counter({'free': 191,
         'entri': 21,
         '2': 155,
         'wkli': 9,
         'comp': 8,
         'win': 48,
         'fa': 2,
         'cup': 3,
         'final': 14,
         'tkt': 2,
         '21st': 1,
         'may': 6,
         'text': 122,
         '87121': 2,
         'receiv': 31,
         'question': 9,
         'std': 6,
         'txt': 141,
         'rate': 26,
```

In [65]:

```
from collections import Counter
Counter(c).most_common(40)
```

Out[65]:

```
[('call', 320),
 ('free', 191),
 ('2', 155),
 ('txt', 141),
 ('text', 122),
 ('u', 119),
 ('ur', 119),
 ('mobil', 114),
 ('stop', 104),
 ('repli', 103),
 ('claim', 98),
 ('4', 97),
 ('prize', 82),
 ('get', 74),
 ('new', 64),
 ('servic', 64),
 ('tone', 63),
 ('send', 60),
 ('urgent', 57),
 ('nokia', 57),
 ('contact', 56),
 ('award', 55),
 ('phone', 52),
 ('cash', 51),
 ('pleas', 51),
 ('week', 49),
 ('win', 48),
 ('c', 45),
 ('collect', 45),
 ('min', 45),
 ('custom', 42),
 ('messag', 42),
 ('guarante', 42),
 ('per', 41),
 ('chat', 38),
 ('tri', 37),
 ('msg', 35),
 ('draw', 35),
 ('number', 35),
 ('cs', 35)]
```

In [66]:

```

pd.DataFrame(Counter(c).most_common(40))
```

Out[66]:

	0	1
0	call	320
1	free	191
2	2	155
3	txt	141
4	text	122
5	u	119
6	ur	119
7	mobil	114
8	stop	104
9	repli	103
10	claim	98
11	4	97
12	prize	82
13	get	74
14	new	64
15	servic	64
16	tone	63
17	send	60
18	urgent	57
19	nokia	57
20	contact	56
21	award	55
22	phone	52
23	cash	51
24	pleas	51
25	week	49
26	win	48
27	c	45
28	collect	45
29	min	45
30	custom	42
31	messag	42
32	guarante	42
33	per	41
34	chat	38
35	tri	37
36	msg	35
37	draw	35
38	number	35

	0	1
39	cs	35

In [67]:
```
from sklearn.feature_extraction.text import CountVectorizer
cv=CountVectorizer()
```

In [68]:
```
d=cv.fit_transform(a['transform']).toarray()
```

In [69]:
```
d
```

Out[69]:
```
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]], dtype=int64)
```

In [70]:
```
d.shape
```

Out[70]:
```
(5169, 6708)
```

In [71]:
```
e=a['target'].values
```

In [72]:
```
e
```

Out[72]:
```
array([0, 0, 1, ..., 0, 0, 0])
```

In [73]:
```
from sklearn.model_selection import train_test_split
```

In [74]:
```
X_train,X_test,y_train,y_test=train_test_split(d,e,test_size=0.3,random
```

In [75]:
```
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
```

In [76]:
```
g=GaussianNB()
m=MultinomialNB()
bn=BernoulliNB()

```

In [77]:
```
g.fit(X_train,y_train)
pre1=g.predict(X_test)

```

In [78]:
```
pre1
```

Out[78]:
```
array([0, 0, 0, ..., 0, 0, 0])
```

In [79]:
```
from sklearn.metrics import accuracy_score,confusion_matrix,precision_s
```

In [80]:

```
g.fit(X_train,y_train)
pre1=g.predict(X_test)
print(accuracy_score(y_test,pre1))
print(confusion_matrix(y_test,pre1))
print(precision_score(y_test,pre1))
```

```
0.8671824629271437
[[1165  169]
 [  37  180]]
0.5157593123209169
```

In [81]:

```
m.fit(X_train,y_train)
pre2=m.predict(X_test)
print(accuracy_score(y_test,pre2))
print(confusion_matrix(y_test,pre2))
print(precision_score(y_test,pre2))
```

```
0.9729206963249516
[[1306   28]
 [  14  203]]
0.8787878787878788
```

In [82]:

```
bn.fit(X_train,y_train)
pre3=bn.predict(X_test)
print(accuracy_score(y_test,pre3))
print(confusion_matrix(y_test,pre3))
print(precision_score(y_test,pre3))
```

```
0.9632495164410058
[[1330    4]
 [  53  164]]
0.9761904761904762
```

In [83]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

t=TfidfVectorizer(max_features=3000)
```

In [84]:

```
de=t.fit_transform(a['transform']).toarray()
```

In [85]:

```
ed=a['target'].values
```

In [86]:

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

In [87]:

```
X_train,X_test,y_train,y_test=train_test_split(de,ed,test_size=0.3,rand
```

In [88]:

```
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
```

In [89]:

```
gg=GaussianNB()
mm=MultinomialNB()
bnn=BernoulliNB()

```

In [90]:

```
gg.fit(X_train,y_train)
pree1=gg.predict(X_test)

```

In [91]:

```
pree1
```

Out[91]:

```
array([0, 0, 0, ..., 0, 0, 0])
```

In [92]:

```
from sklearn.metrics import accuracy_score,confusion_matrix,precision_s
```

In [93]:

```
gg.fit(X_train,y_train)
pre1=gg.predict(X_test)
print(accuracy_score(y_test,pree1))
print(confusion_matrix(y_test,pree1))
print(precision_score(y_test,pree1))
```

```
0.8671824629271437
[[1164  170]
 [  36  181]]
0.5156695156695157
```

In [94]:

```
mm.fit(X_train,y_train)
pree2=mm.predict(X_test)
print(accuracy_score(y_test,pree2))
print(confusion_matrix(y_test,pree2))
print(precision_score(y_test,pree2))
```

```
0.9671179883945842
[[1334    0]
 [  51  166]]
1.0
```

In [95]:

```
bnn.fit(X_train,y_train)
pree3=bnn.predict(X_test)
print(accuracy_score(y_test,pree3))
print(confusion_matrix(y_test,pree3))
print(precision_score(y_test,pree3))
```

```
0.9748549323017408
[[1331    3]
 [  36  181]]
0.9836956521739131
```

In [96]:

```
# tfidf mnn
```

In [97]:

```
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 [98]:

```
ss=SVC(kernel='sigmoid',gamma=1.0)
knn=KNeighborsClassifier()
mnn= MultinomialNB()
dtt=DecisionTreeClassifier(max_depth=5)
ll= LogisticRegression(solver='liblinear',penalty='l1')
rf=RandomForestClassifier(n_estimators=50,random_state=2)
bbc=BaggingClassifier(n_estimators=50,random_state=2)
abc=AdaBoostClassifier(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 [99]:

```
cls={
    'SVC':ss,
    'KNN':knn,
    'MNN':mnn,
    'dtt':dtt,
    'll':ll,
    'rf':rf,
    'bbc':bbc,
    'abc':abc,
    'etc':etc,
    'gbdt':gbdt,
    'xgb':xgb,
}
```

In [100]:

```
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 [101]:

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

```

Out[101]:

```
(0.9664732430689877, 0.9558011049723757)
```

In [102]:

```
accuracy_scores=[]
precision_scores=[]
for n,clf in cls.items():
    current_accuracy,current_precision=train_classifier(clf,X_train,y_t
    print("for ",n)
    print("Accuracy",current_accuracy)
    print("Precision",current_precision)
    accuracy_scores.append(current_accuracy)
    precision_scores.append(current_precision)


```

```
for  SVC
Accuracy 0.9664732430689877
Precision 0.9558011049723757
for  KNN
Accuracy 0.90715667311412
Precision 1.0
for  MNN
Accuracy 0.9671179883945842
Precision 1.0
for  dtt
Accuracy 0.9381044487427466
Precision 0.9230769230769231
for  ll
Accuracy 0.9464861379754996
Precision 0.9527027027027027
for  rf
Accuracy 0.965183752417795
Precision 1.0
for  bbc
Accuracy 0.9638942617666022
Precision 0.9497206703910615
for  abc
Accuracy 0.9580915538362347
Precision 0.9318181818181818
for  etc
Accuracy 0.9658284977433914
Precision 0.9939759036144579
for  gbdt
Accuracy 0.9381044487427466
Precision 0.9763779527559056
for  xgb
Accuracy 0.9613152804642167
Precision 0.9875776397515528
```

In [109]:

```
p=pd.DataFrame({'Algorithm':cls.keys(),'Accuracy':accuracy_scores,'Prec
```

In [110]:

```
p
```

Out[110]:

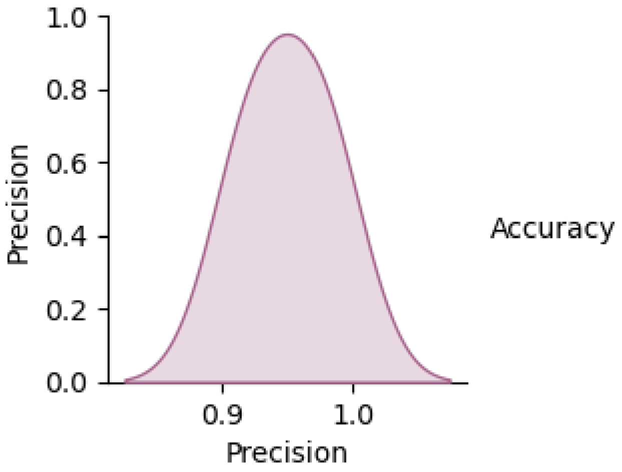
	Algorithm	Accuracy	Precision
1	KNN	0.907157	1.000000
2	MNN	0.967118	1.000000
5	rf	0.965184	1.000000
8	etc	0.965828	0.993976
10	xgb	0.961315	0.987578
9	gbdt	0.938104	0.976378
0	SVC	0.966473	0.955801
4	ll	0.946486	0.952703
6	bbc	0.963894	0.949721
7	abc	0.958092	0.931818
3	dtt	0.938104	0.923077

In [111]:

```
sns.pairplot(p,hue='Accuracy')
```

Out[111]:

```
<seaborn.axisgrid.PairGrid at 0x1d1d38f8250>
```

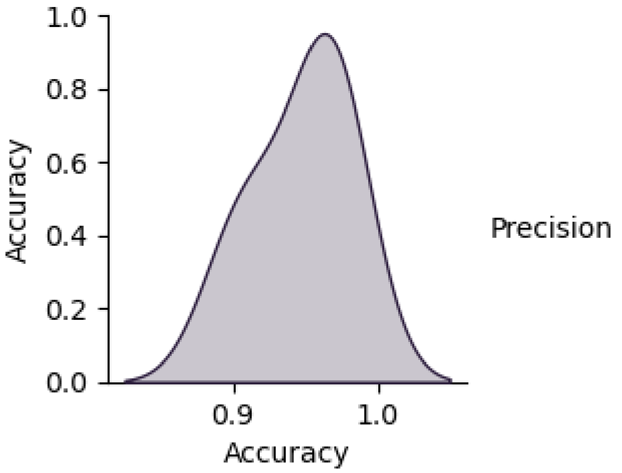


In [112]:

```
sns.pairplot(p,hue='Precision')
```

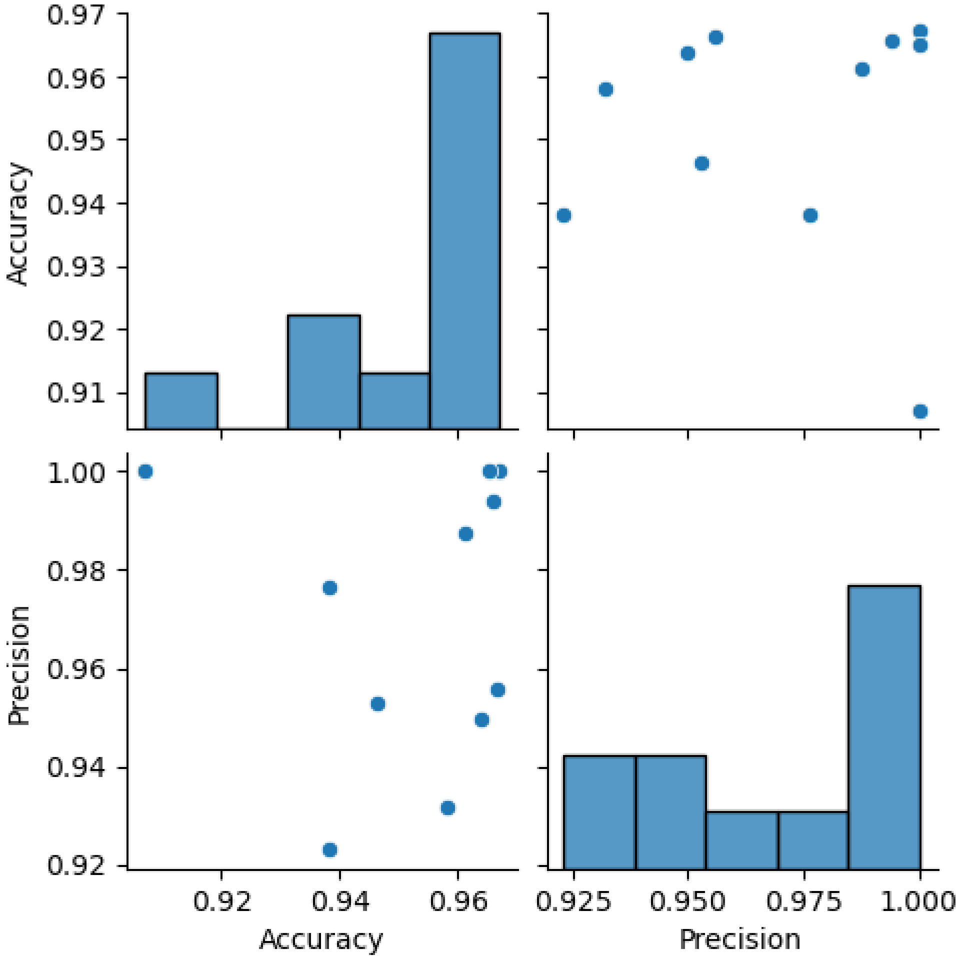
Out[112]:

```
<seaborn.axisgrid.PairGrid at 0x1d1d330af50>
```



```
In [113]: sns.pairplot(p)
```

```
Out[113]: <seaborn.axisgrid.PairGrid at 0x1d1d332f760>
```



```
In [114]: p1=pd.melt(p,id_vars='Algorithm')
```

In [115]:

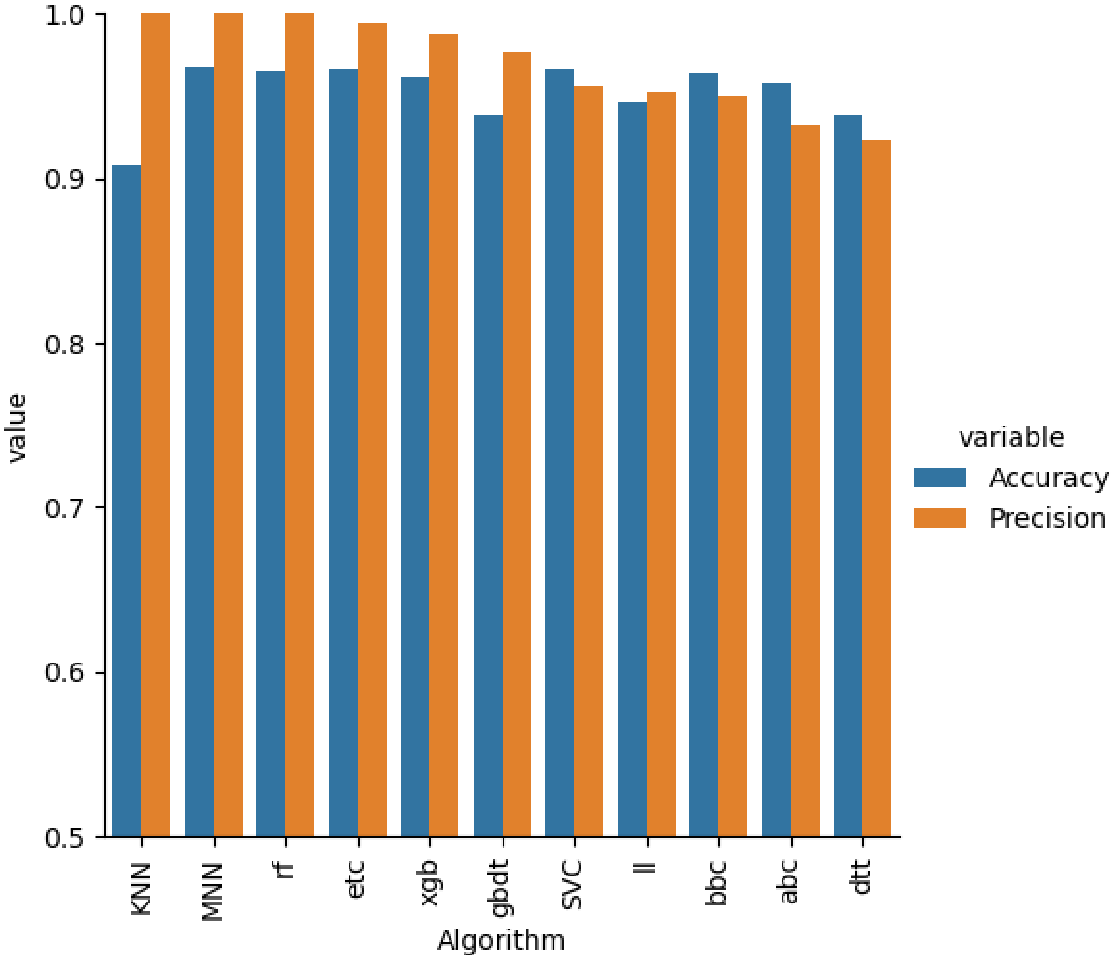
```
p1
```

Out[115]:

	Algorithm	variable	value
0	KNN	Accuracy	0.907157
1	MNN	Accuracy	0.967118
2	rf	Accuracy	0.965184
3	etc	Accuracy	0.965828
4	xgb	Accuracy	0.961315
5	gbdt	Accuracy	0.938104
6	SVC	Accuracy	0.966473
7	ll	Accuracy	0.946486
8	bbc	Accuracy	0.963894
9	abc	Accuracy	0.958092
10	dtt	Accuracy	0.938104
11	KNN	Precision	1.000000
12	MNN	Precision	1.000000
13	rf	Precision	1.000000
14	etc	Precision	0.993976
15	xgb	Precision	0.987578
16	gbdt	Precision	0.976378
17	SVC	Precision	0.955801
18	ll	Precision	0.952703
19	bbc	Precision	0.949721
20	abc	Precision	0.931818
21	dtt	Precision	0.923077

In [116]:

```
sns.catplot(x='Algorithm',y='value',
               hue='variable',data=p1,kind='bar',height=5)
plt.ylim(0.5,1.0)
plt.xticks(rotation='vertical')
plt.show()
```



In [117]:

```
svc=SVC(kernel='sigmoid',gamma=1.0,probability=True)
mnb=MultinomialNB()
#etc=ExtraTreesClassifier(n_estimators=50,random_state=2)
rf=RandomForestClassifier(n_estimators=50,random_state=2)
from sklearn.ensemble import VotingClassifier
```

In [118]:

```
voting=VotingClassifier(estimators=[('svm',svc),('nb',mnb),('rf',rf)],v
```

In [ ]:

```

```

In [119]:

```
#predicted = VotingClassifier.predict(x_test)
```

In [120]:

```
voting.fit(X_train,y_train)
```

Out[120]:

```
VotingClassifier(estimators=[('svm',
                              SVC(gamma=1.0, kernel='sigmoid',
                                  probability=True)),
                             ('nb', MultinomialNB()),
                             ('rf',
                              RandomForestClassifier(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 [121]:

```
y_pred=voting.predict(X_test)
print('Accuracy',accuracy_score(y_test,y_pred))
print('Precision',precision_score(y_test,y_pred))
```

```
Accuracy 0.97678916827853
Precision 1.0
```

In [122]:

```
import pickle
```

In [123]:

```
pickle.dump(t,open('Vectorizer.pkl','wb'))
pickle.dump(voting,open('model.pkl','wb'))
```

In [ ]:

```

```