In [2]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
df = pd.read_csv('C:/Users/dhpat/OneDrive/Desktop/test jupyter/Spam_mail_classifier/spam
```

In [4]:
```
df.head()
```

Out[4]:

| | 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 [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

In [6]:
```
df.head()
```

Out[6]:

| | 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 [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)
```

In [10]:
```
df.sample(5)
```

Out[10]:

| | target | text |
|---|---|---|
| **4225** | ham | Ok thats cool. Its , just off either raglan rd... |
| **5055** | ham | Goodnight da thangam I really miss u dear. |
| **881** | ham | see, i knew giving you a break a few times wou... |
| **3350** | ham | At what time are you coming. |
| **2228** | ham | Those were my exact intentions |

In [11]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

In [12]:
```
df['target'] = encoder.fit_transform(df['target'])
```

In [13]:
```
df.head()
```

Out[13]:

| | 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 [14]:
```
df.isnull().sum()
```

Out[14]:
```
target    0
text      0
dtype: int64
```

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

Out[15]:
```
403
```

In [16]:
```
df = df.drop_duplicates(keep='first')
```

In [17]:
```
df.duplicated().sum()
```

Out[17]:
```
0
```

In [18]:
```
df.shape
```

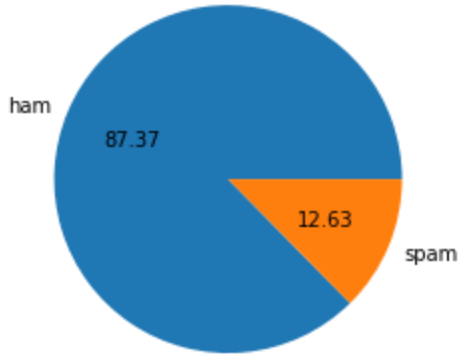
Out[18]:
```
(5169, 2)
```

# 2. EDA

In [19]:
```
df['target'].value_counts()
```

Out[19]:
```
0    4516
1     653
Name: target, dtype: int64
```

In [20]:
```
import matplotlib.pyplot as plt
plt.pie(df['target'].value_counts(), labels=['ham','spam'],autopct="%0.2f")
plt.show()
```




In [22]:
```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\dhpat\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

Out[22]:
```
True
```

In [23]:
```
df['num_characters'] = df['text'].apply(len)
```

In [24]:
```
df.head()
```

Out[24]:

| | 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 [25]:
```
# num of words
df['num_words'] = df['text'].apply(lambda x:len(nltk.word_tokenize(x)))
```

In [26]:
```
df.head()
```

Out[26]:

| | 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 [27]:
```
df['num_sentences'] = df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))
```

In [28]:
```
df[['num_characters','num_words','num_sentences']].describe()
```

Out[28]:

| | num_characters | num_words | num_sentences |
|---|---|---|---|
| **count** | 5169.000000 | 5169.000000 | 5169.000000 |
| **mean** | 78.923776 | 18.456761 | 1.966531 |
| **std** | 58.174846 | 13.325633 | 1.449833 |
| **min** | 2.000000 | 1.000000 | 1.000000 |
| **25%** | 36.000000 | 9.000000 | 1.000000 |
| **50%** | 60.000000 | 15.000000 | 1.000000 |
| **75%** | 117.000000 | 26.000000 | 2.000000 |
| **max** | 910.000000 | 220.000000 | 38.000000 |

In [29]:
```
# ham
df[df['target'] == 0][['num_characters','num_words','num_sentences']].describe()
```

Out[29]:

| | num_characters | num_words | num_sentences |
|---|---|---|---|
| **count** | 4516.000000 | 4516.000000 | 4516.000000 |
| **mean** | 70.456820 | 17.123782 | 1.820195 |
| **std** | 56.356802 | 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 |

In [30]:
```
#spam
df[df['target'] == 1][['num_characters','num_words','num_sentences']].describe()
```

Out[30]:

| | num_characters | num_words | num_sentences |
|---|---|---|---|
| **count** | 653.000000 | 653.000000 | 653.000000 |
| **mean** | 137.479326 | 27.675345 | 2.978560 |
| **std** | 30.014336 | 7.011513 | 1.493185 |
| **min** | 13.000000 | 2.000000 | 1.000000 |
| **25%** | 131.000000 | 25.000000 | 2.000000 |
| **50%** | 148.000000 | 29.000000 | 3.000000 |
| **75%** | 157.000000 | 32.000000 | 4.000000 |
| **max** | 223.000000 | 46.000000 | 9.000000 |

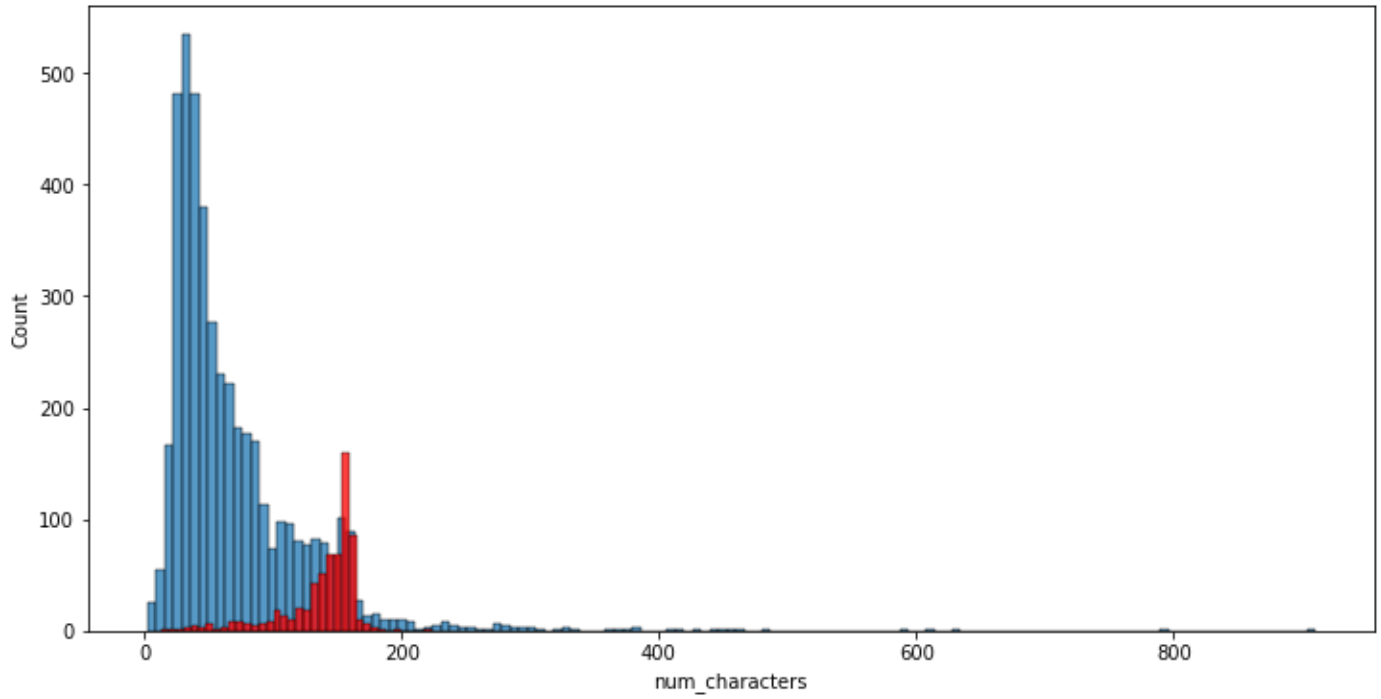
In [31]:
```
import seaborn as sns
```

In [32]:
```
plt.figure(figsize=(12,6))
```

```
sns.histplot(df[df['target'] == 0]['num_characters'])
sns.histplot(df[df['target'] == 1]['num_characters'],color='red')
```

Out[32]:

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

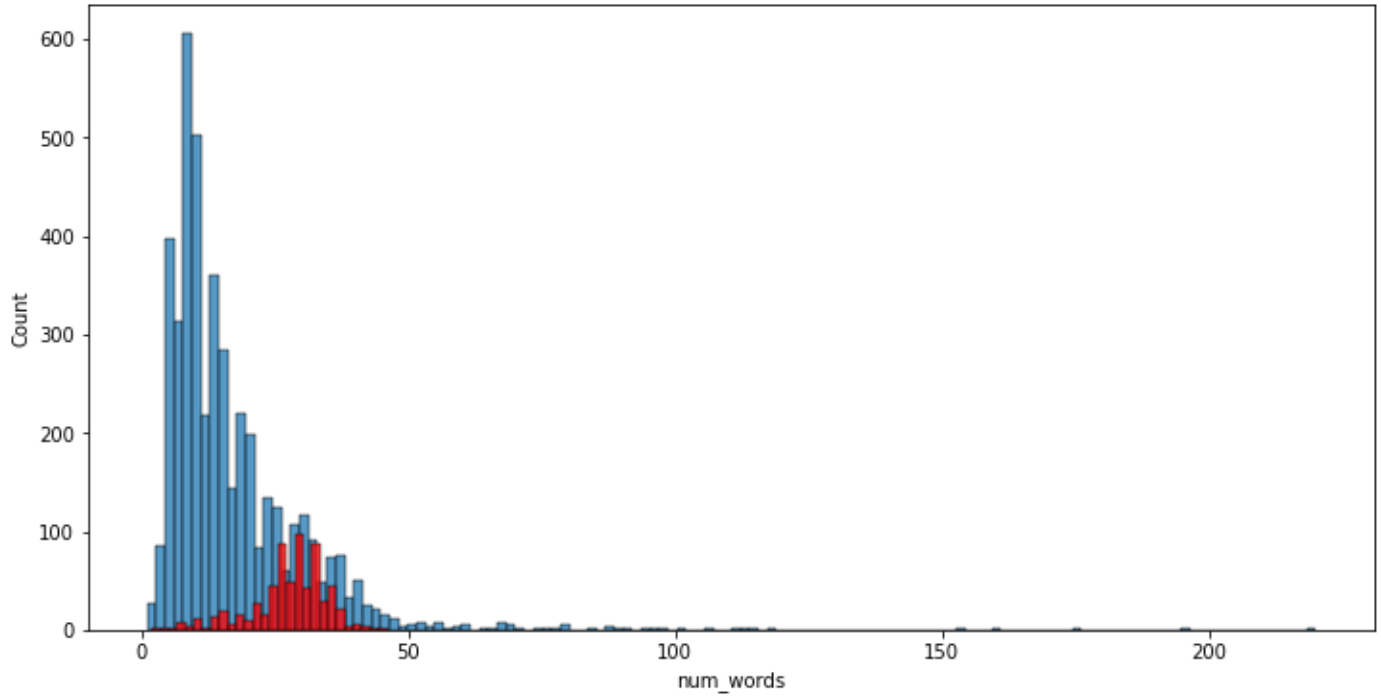



In [33]:

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

Out[33]:

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

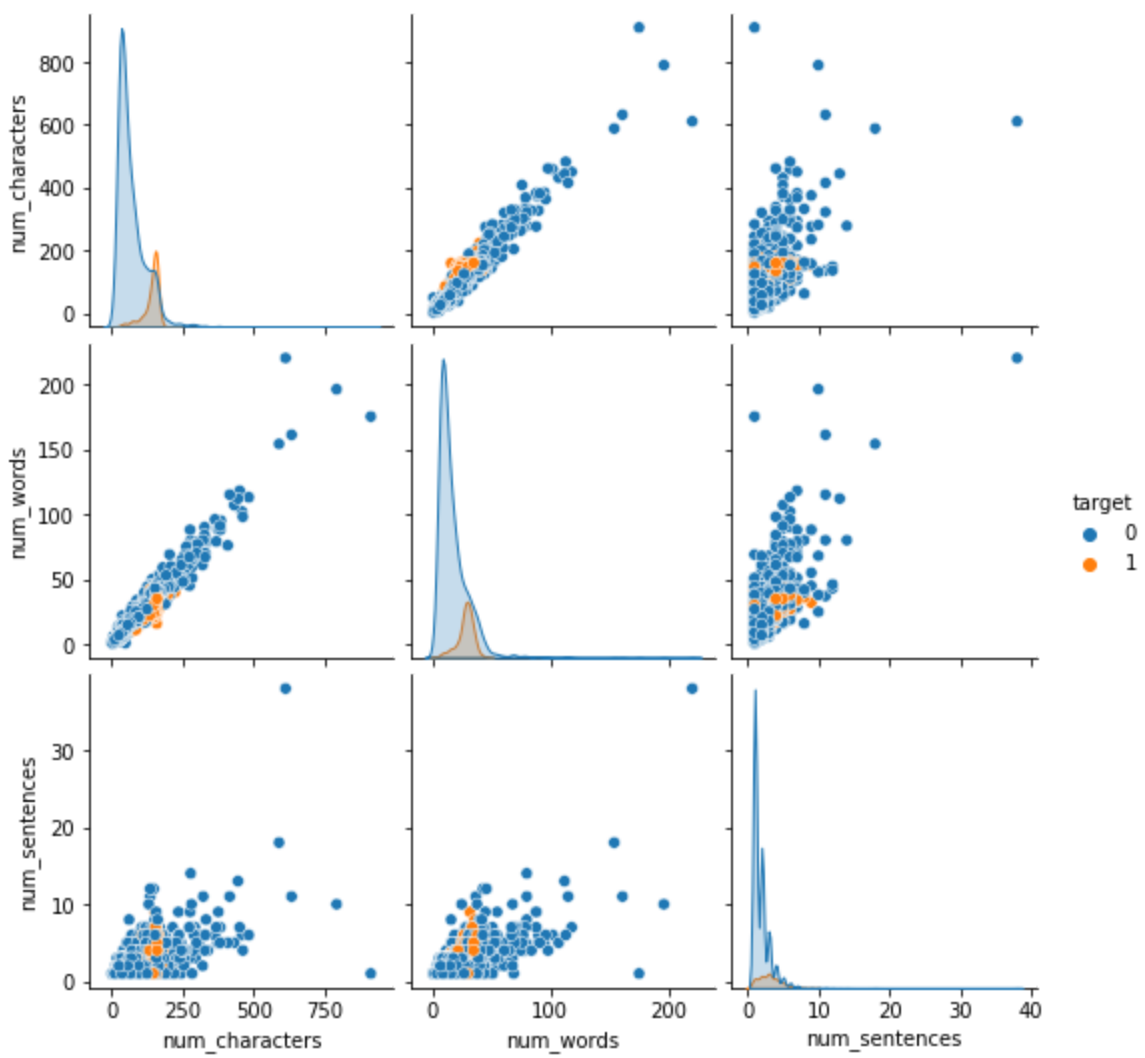



In [34]:

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

Out[34]:

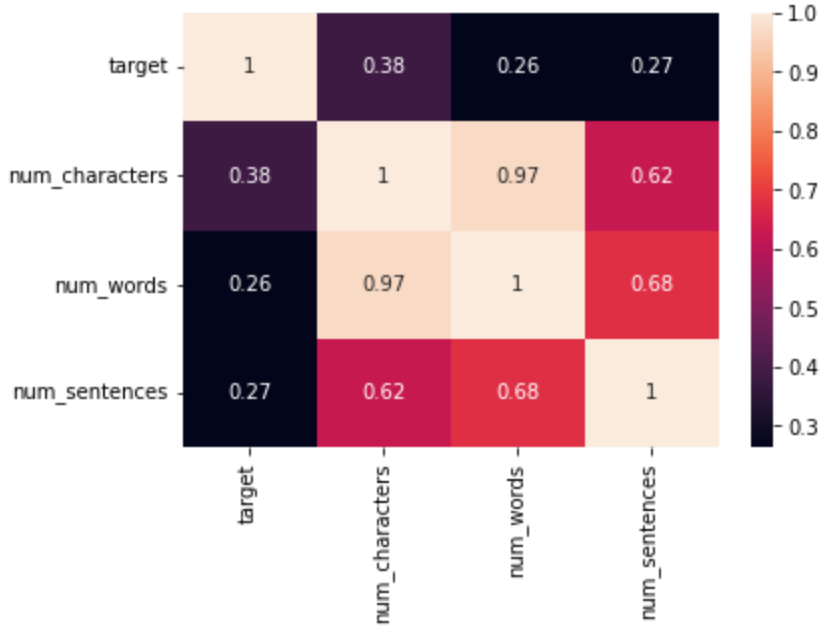
```
<seaborn.axisgrid.PairGrid at 0x1ac3871eb30>
```




In [35]:
```
sns.heatmap(df.corr(),annot=True)
```

```
C:\Users\dhpat\AppData\Local\Temp\ipykernel_8844\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[35]:
```
<AxesSubplot: >
```



# 3. Data Preprocessing

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

In [53]:
```
from nltk.corpus import stopwords
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\dhpat\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

Out[53]:
```
True
```

In [54]:
```
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)
```

In [52]:
```
df['text'][10]
```

Out[52]:
```
"I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k?
I've cried enough today."
```

In [57]:
```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
ps.stem('loving')
```

Out[57]:
```
'love'
```

In [ ]:

In [ ]:

In [ ]: