import libaries

In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import cross_val_score
```

read data

In [2]:
```python
df=pd.read_csv(r"C:\Users\Somay\Documents\Spam SMS Collection.csv",encoding="ISO-8859-1")
```

understanding the data

In [3]:
```python
print(df.columns)
```

```
Index(['ham', 'Go until jurong point, crazy.. Available only in bugis n great world la e buff
et... Cine there got amore wat...'], dtype='object')
```

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

Out[4]:

	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
0	ham	Ok lar... Joking wif u oni...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...
2	ham	U dun say so early hor... U c already then say...
3	ham	Nah I don't think he goes to usf, he lives aro...
4	spam	FreeMsg Hey there darling it's been 3 week's n...

In [5]:
```python
df.shape
```

Out[5]: (5571, 2)

In [6]:
```python
df.isnull().sum()
```

Out[6]:
```
ham
0
Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine ther
```

In [7]:
```python
df=df.rename(columns={'ham':'type','Go until jurong point, crazy.. Available only in bugis n
great world la e buffet... Cine there got amore wat...':'message'})
```

In [8]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5571 entries, 0 to 5570
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   type     5571 non-null   object
 1   message  5571 non-null   object
dtypes: object(2)
memory usage: 43.6+ KB
```

In [9]:
```python
df.head()
```

Out[9]:

	type	message
0	ham	Ok lar... Joking wif u oni...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...
2	ham	U dun say so early hor... U c already then say...
3	ham	Nah I don't think he goes to usf, he lives aro...
4	spam	FreeMsg Hey there darling it's been 3 week's n...

In [10]:
```python
df['type']=df['type'].map({'ham':0,'spam':1})
df
```

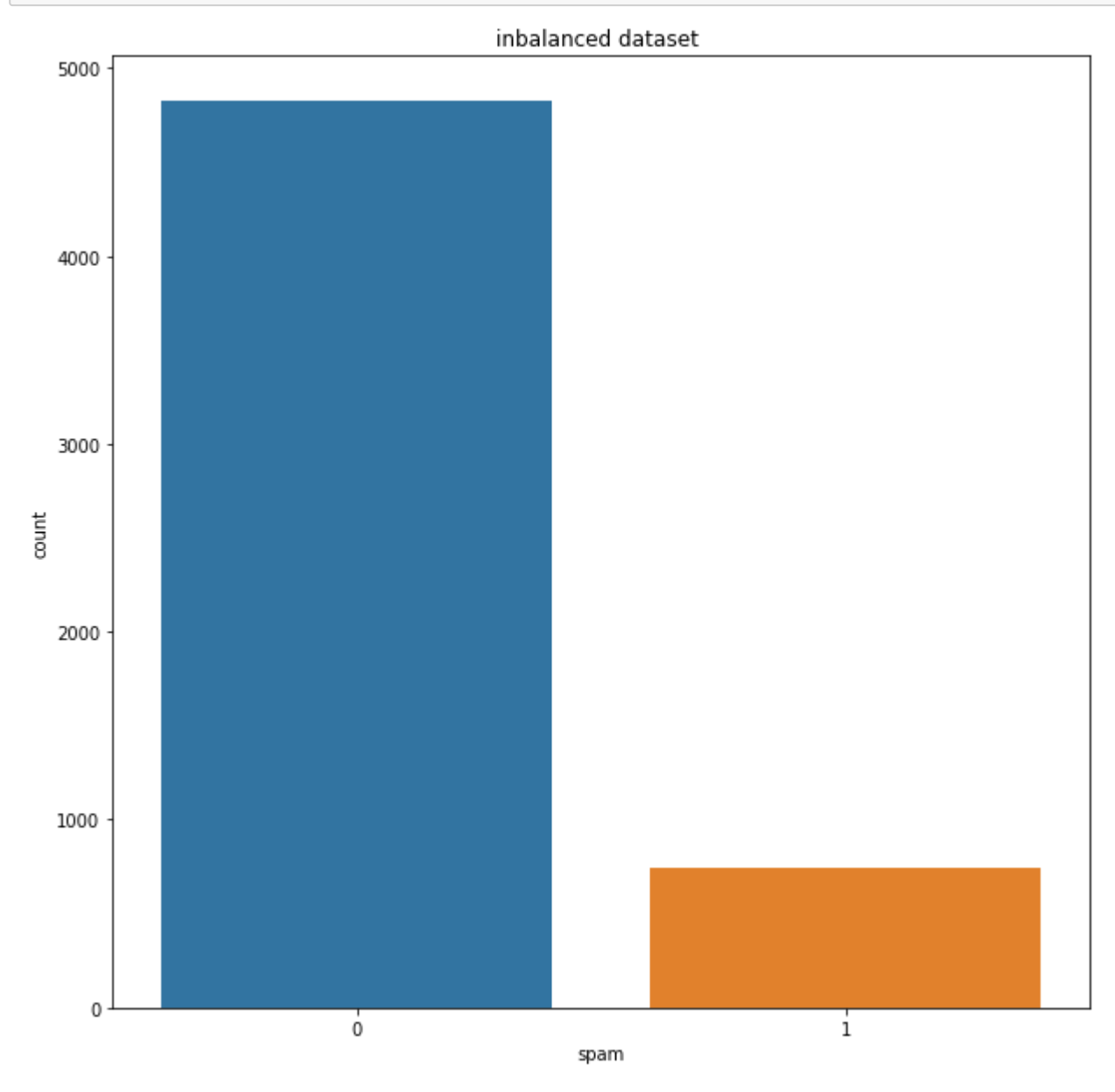
Out[10]:

	type	message
0	0	Ok lar... Joking wif u oni...
1	1	Free entry in 2 a wkly comp to win FA Cup fina...
2	0	U dun say so early hor... U c already then say...
3	0	Nah I don't think he goes to usf, he lives aro...
4	1	FreeMsg Hey there darling it's been 3 week's n...
...	...	...
5566	1	This is the 2nd time we have tried 2 contact u...
5567	0	Will ü b going to esplanade fr home?
5568	0	Pity, * was in mood for that. So...any other s...
5569	0	The guy did some bitching but I acted like i'd...
5570	0	Rofl. Its true to its name

5571 rows × 2 columns

data visualization for inbalanced dataset

In [11]:
```python
plt.figure(figsize=(10,10))
g=sns.countplot(x='type',data=df)
p=plt.title('inbalanced dataset ')
p=plt.xlabel('spam')
p=plt.ylabel('count')
```



In [12]:
```python
print('ham count',df['type'].value_counts()[0])
print('spam count',df['type'].value_counts()[1])
```

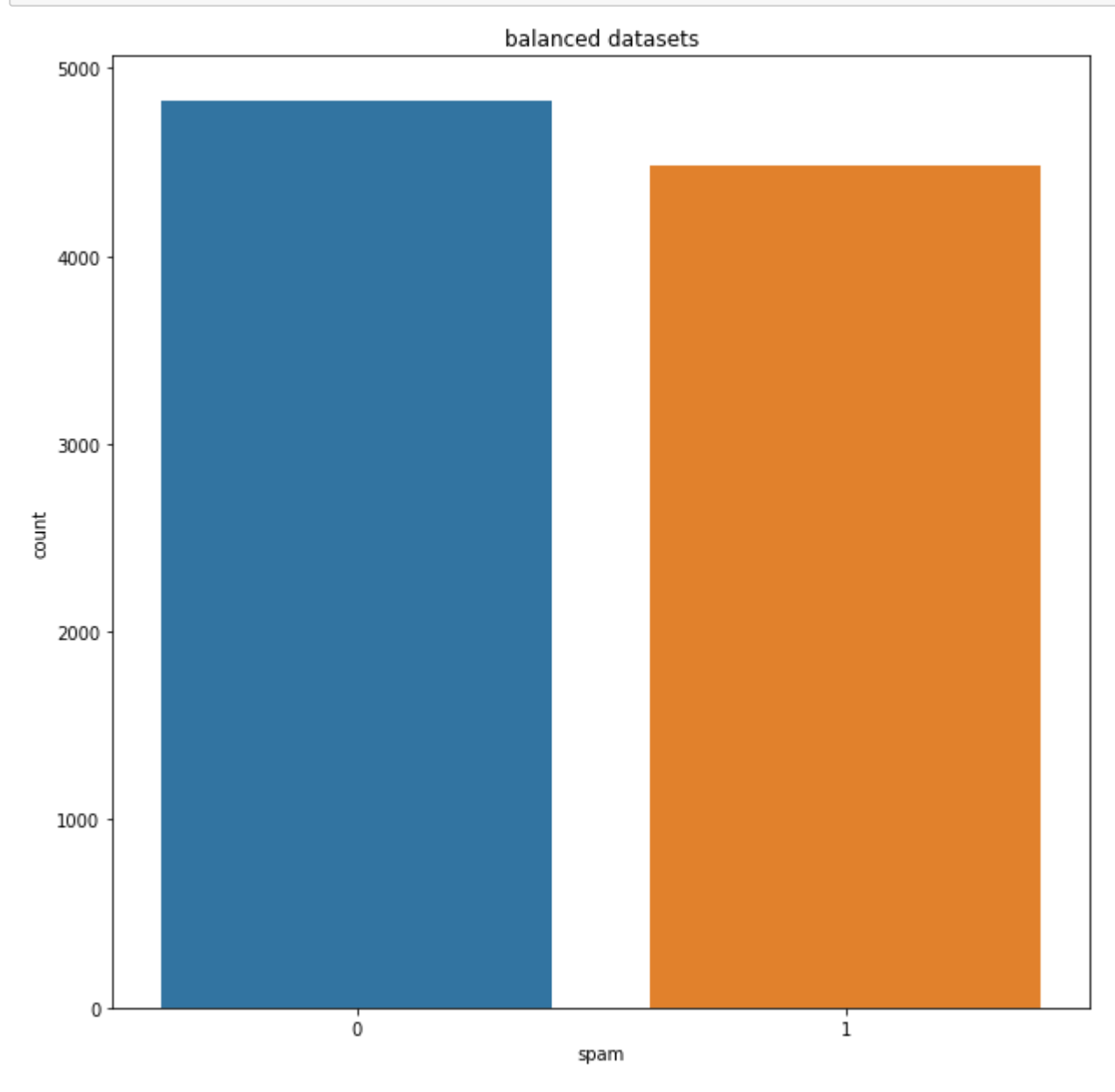
```
ham count 4824
spam count 747
```

In [13]:
```python
only_spam=df[df['type']==1]
count=int((df.shape[0]-only_spam.shape[0])/only_spam.shape[0])
for i in range(0,count-1):
    df=pd.concat([df,only_spam])
df.shape
```

Out[13]: (9306, 2)

data visualization for balanced dataset

In [14]:
```python
plt.figure(figsize=(10,10))
g=sns.countplot(x='type',data=df)
p=plt.title('balanced datasets')
p=plt.xlabel('spam')
p=plt.ylabel('count')
```



word count spam and ham message and visualize the graph

In [15]:
```python
df['word_count']=df['message'].apply(lambda x: len(x.split()))
```

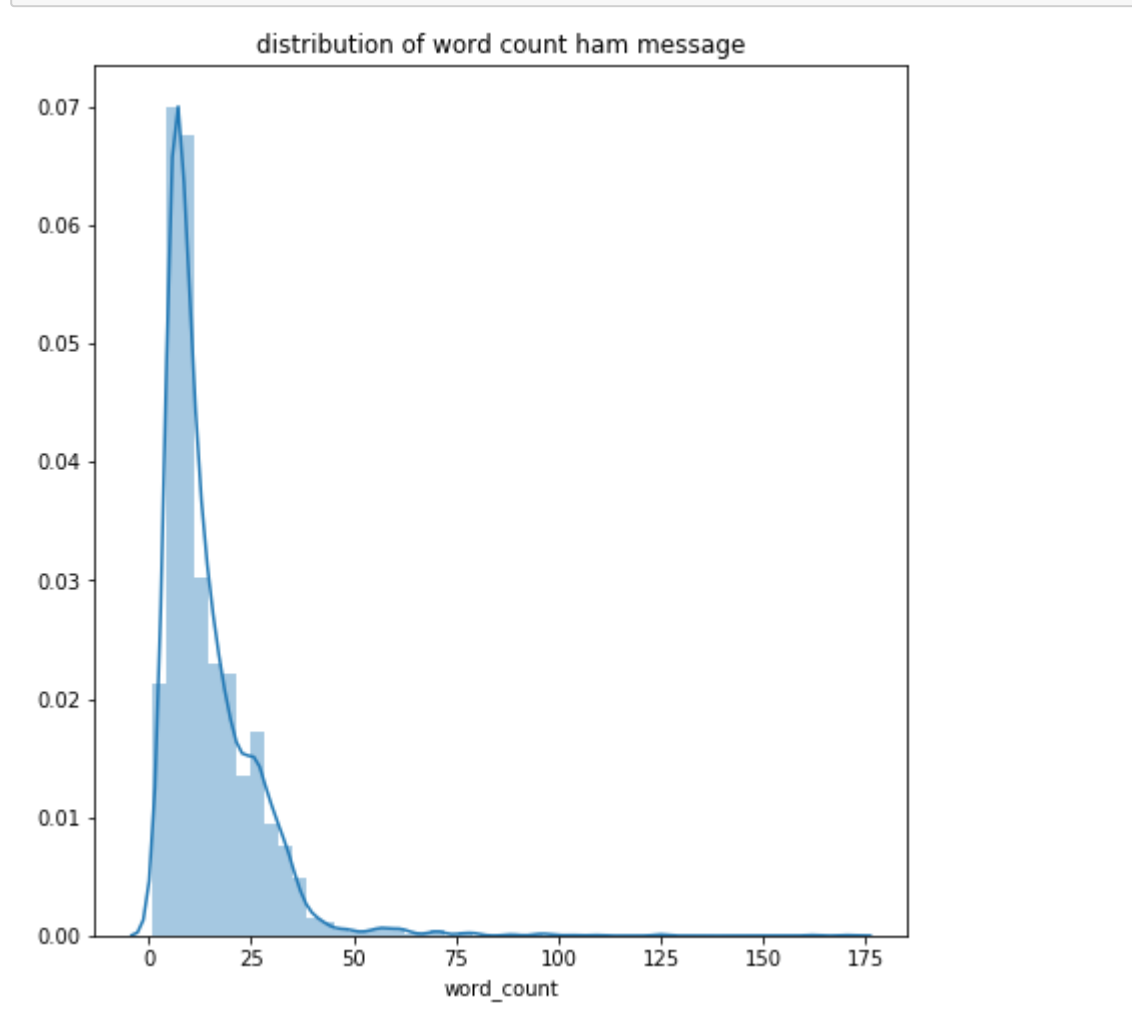
In [16]:
```python
df.head()
```

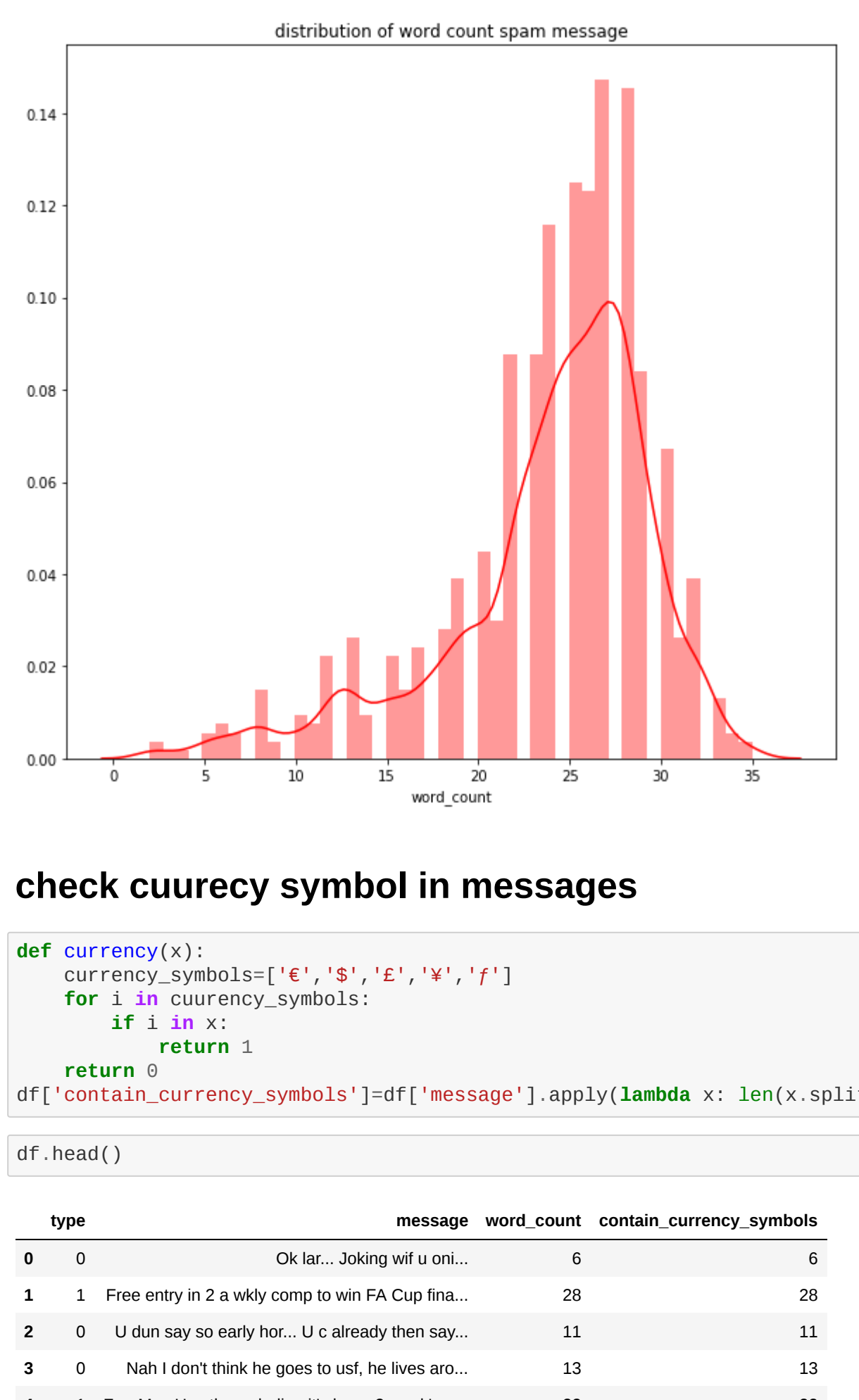
Out[16]:

	type	message	word_count
0	0	Ok lar... Joking wif u oni...	6
1	1	Free entry in 2 a wkly comp to win FA Cup fina...	28
2	0	U dun say so early hor... U c already then say...	11
3	0	Nah I don't think he goes to usf, he lives aro...	13
4	1	FreeMsg Hey there darling it's been 3 week's n...	32

In [17]:
```python
plt.figure(figsize=(16,8))
plt.subplot(1,2,1)
g=sns.distplot(a=df[df['type']==0].word_count)
p=plt.title('distribution of word count ham message')

plt.figure(figsize=(16,8))
plt.subplot(1,2,2)
g=sns.distplot(a=df[df['type']==1].word_count,color='red')
p=plt.title('distribution of word count spam message')
plt.tight_layout()
```





check cuurecy symbol in messages

In [18]:
```python
def currency(x):
    currency_symbols=['€','$','£','¥','ƒ']
    for i in cuurency_symbols:
        if i in x:
            return 1
    return 0
df['contain_currency_symbols']=df['message'].apply(lambda x: len(x.split()))
```

In [19]:
```python
df.head()
```

Out[19]:

	type	message	word_count	contain_currency_symbols
0	0	Ok lar... Joking wif u oni...	6	6
1	1	Free entry in 2 a wkly comp to win FA Cup fina...	28	28
2	0	U dun say so early hor... U c already then say...	11	11
3	0	Nah I don't think he goes to usf, he lives aro...	13	13
4	1	FreeMsg Hey there darling it's been 3 week's n...	32	32

data cleaning with lemmatized , tfidfvectorization and many more parameters

In [20]:
```python
corpus=[]
wl=WordNetLemmatizer()
for sms_string in list(df.message):
    message=re.sub(pattern='[a-zA-Z]', repl='', string=sms_string)
    message=message.lower()
    words=message.split()
    filtered_words=[word for word in words if word not in set(stopwords.words('english'))]
    lemmatized_words=[wl.lemmatize(word) for word in filtered_words]
    message=''.join(lemmatized_words)
    corpus.append(message)
```

In [21]:
```python
corpus[0:5]
```

Out[21]: ['......', "2212005.87121()&'0845281007518'", '......', "',", "'3'!'?!,£1.50"]

In [22]:
```python
tfidf=TfidfVectorizer(max_features=500)
vector=tfidf.fit_transform(corpus).toarray()
feature_names=tfidf.get_feature_names()
```

In [23]:
```python
X=pd.DataFrame(vector,columns=feature_names)
y=df['type']
```

model evaluation

In [24]:
```python
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42)
```

average and S.D score

In [25]:
```python
mn=MultinomialNB()
cv=cross_val_score(mn,X,y,scoring='f1',cv=10)
print("average score mn={}".format(round(cv.mean(), 3)))
print("standard deviation={}".format(round(cv.std(), 3)))
```

```
average score mn=0.745
standard deviation=0.008
```

classification reports

In [26]:
```python
mn.fit(X_train,y_train)
y_pred=mn.predict(X_test)
print("classification report of this model")
print(classification_report(y_test,y_pred))
```

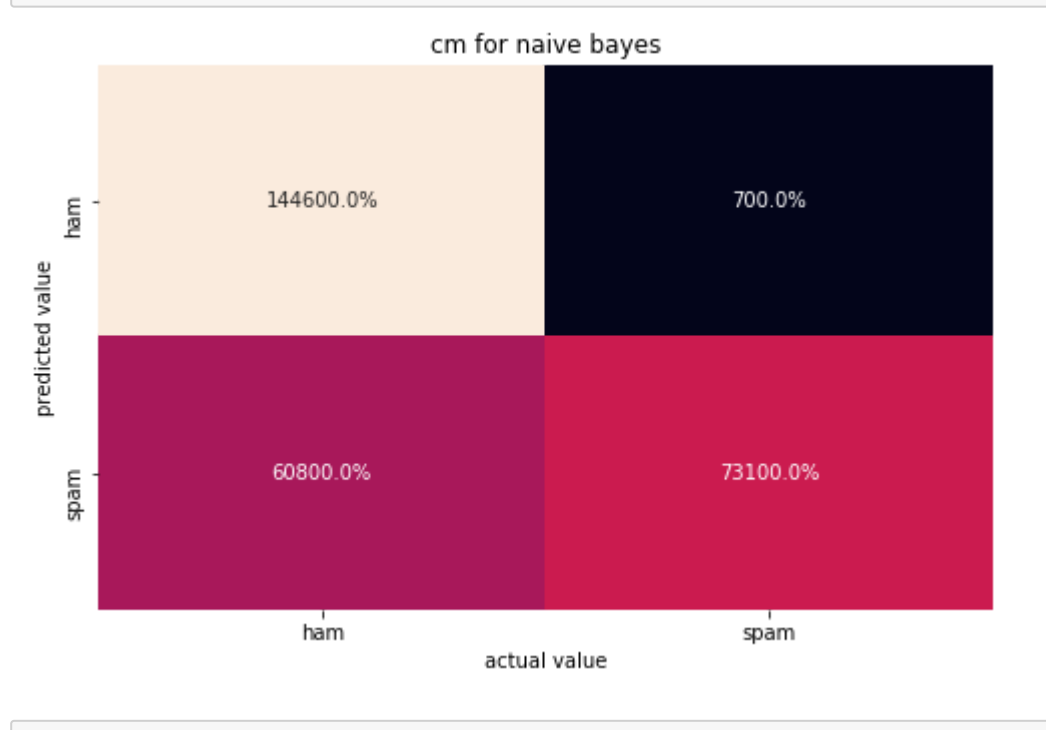
```
classification report of this model
              precision    recall  f1-score   support

           0       0.70      1.00      0.82      1453
           1       0.99      0.55      0.70      1339

    accuracy                           0.78      2792
   macro avg       0.85      0.77      0.76      2792
weighted avg       0.84      0.78      0.77      2792
```

confusion matrix

In [27]:
```python
cm=confusion_matrix(y_test,y_pred)
axis_labels=['ham','spam']
plt.figure(figsize=(8,5))
g=sns.heatmap(data=cm,annot=True,cbar=False, xticklabels=axis_labels , yticklabels=axis_labe
ls , fmt='.1%')
p=plt.xlabel('actual value')
p=plt.ylabel('predicted value')
```



In [ ]:

In [ ]:

In [ ]:

In [ ]: