Exercise 6: Spam or ham

Cho dữ liệu spam.csv

Yêu cầu: đọc dữ liệu về, chuẩn hóa dữ liệu (nếu cần) và áp dụng thuật toán Naive Bayes để thực hiện việc dự đoán khả năng email là spam hay không dựa trên các thuộc tính v2

1. Tạo X_train, X_test, y_train, y_test từ dữ liệu đọc được với tỷ lệ dữ liệu test là 0.2
2. Áp dụng thuật toán Naive Bayer => kết quả
3. Cho dữ liệu Test x_new = np.array(['Dear Ms. Phuong. I will come on time.',

```
                'URGENT! We are trying to contact you. Today is the last d
    ay of sale. Discount up to 50%'])
```

Cho biết kết quả

In [24]:
```
# Load libraries
import numpy as np
import pandas as pd
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
```

In [25]:
```
# import some data to play with
data = pd.read_csv("spam.csv", encoding='latin-1')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
v1            5572 non-null object
v2            5572 non-null object
Unnamed: 2    50 non-null object
Unnamed: 3    12 non-null object
Unnamed: 4    6 non-null object
dtypes: object(5)
memory usage: 217.7+ KB
```

In [26]:
```
data['v1'].head()
```

Out[26]:
```
0     ham
1     ham
2    spam
3     ham
4     ham
Name: v1, dtype: object
```

In [27]:
```
source = data['v2']
type(source)
```

Out[27]:
```
pandas.core.series.Series
```

In [28]:
```
source[:5]
```

Out[28]:
```
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...
Name: v2, dtype: object
```

In [29]:
```
target = data['v1']
type(target)
```

Out[29]:
```
pandas.core.series.Series
```

In [30]:
```
target = target.replace("ham", 1)
```

In [31]:
```
target = target.replace("spam", 0)
```

In [32]:
```
target[:5]
```

Out[32]:
```
0    1
1    1
2    0
3    1
4    1
Name: v1, dtype: int64
```

In [33]:
```
text_data = np.array(source)
text_data
```

Out[33]:
```
array(['Go until jurong point, crazy.. Available only in bugis n great world la
e buffet... Cine there got amore wat...',
       'Ok lar... Joking wif u oni...',
       "Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Tex
t FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075ove
r18's",
       ..., 'Pity, * was in mood for that. So...any other suggestions?',
       "The guy did some bitching but I acted like i'd be interested in buying
something else next week and he gave it to us for free",
       'Rofl. Its true to its name'], dtype=object)
```

In [34]:
```
target_data = np.array(target)
target_data
```

Out[34]:
```
array([1, 1, 0, ..., 1, 1, 1], dtype=int64)
```

In [35]:
```
# Create bag of words
count = CountVectorizer()
count.fit(text_data)
bag_of_words = count.transform(text_data)
bag_of_words
```

Out[35]:
```
<5572x8672 sparse matrix of type '<class 'numpy.int64'>'
        with 73916 stored elements in Compressed Sparse Row format>
```

In [36]:
```
# Create feature matrix
X = bag_of_words.toarray()
X
```

Out[36]:
```
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 [37]:
```
X.shape
```

Out[37]:
```
(5572, 8672)
```

In [38]:
```
# Create target vector
y = np.array(target)
```

In [39]:
```
y.shape
```

Out[39]:
```
(5572,)
```

In [40]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
```

In [41]:
```
# Create multinomial naive Bayes object with prior probabilities of each class
clf = MultinomialNB()

# Train model
model = clf.fit(X_train, y_train)
```

In [42]:
```
y_pred = clf.predict(X_test)
```

In [43]:
```
print('score Scikit learn: ', model.score(X_test,y_test))
```

```
score Scikit learn:  0.9775784753363229
```

In [44]:
```
from sklearn.metrics import accuracy_score
# Kiểm tra độ chính xác
print("Accuracy is ", accuracy_score(y_test,y_pred)*100,"%")
```

```
Accuracy is  97.75784753363229 %
```

In [45]:
```
x_new = np.array(['Dear Ms. Phuong. I will come on time.',
                  'URGENT! We are trying to contact you. Today is the last day of
x_new = count.transform(x_new)
```

In [46]:
```
# Predict new observation's class
y_pred = model.predict(x_new)
y_pred
```

Out[46]:
```
array([1, 0], dtype=int64)
```

In [ ]:
```

```