In [1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
```

In [2]:
```
df=pd.read_csv('dm_mid5.csv', header=0)
```

In [3]:
```
df.head()
```

Out[3]:

	CONTENT	CLASS
0	+447935454150 lovely girl talk to me xxx	1
1	I always end up coming back to this song 	0
2	my sister just received over 6,500 new <a rel=...	1
3	Cool	0
4	Hello I'am from Palastine	1

In [4]:
```
X = df['CONTENT']
y = df['CLASS']
```

In [5]:
```
# Note random_state
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=
0.25,
                                                    random_state=20)
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

```
(336,) (336,)
(112,) (112,)
```

In [6]:
```
vectorizer = CountVectorizer()
vectorizer.fit(X_train)
vocab = vectorizer.get_feature_names()
print('Vocabulary size:', len(vocab))
print(vocab[-10:])  # debug
```

```
Vocabulary size: 1325
['youtube', 'youtuber', 'youtubers', 'yrs', 'ytma', 'yuliya', 'yut
tx04oyqq', 'zesty', 'zip', 'zonepa']
```

In [7]:
```
X_train_bow = vectorizer.transform(X_train)
X_test_bow = vectorizer.transform(X_test)
```

In [8]:
```
model = MultinomialNB(alpha=1.0)
model.fit(X_train_bow, y_train)
```

Out[8]:
```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

In [9]:
```
test_score = model.score(X_test_bow, y_test)
print('Test accuracy:', test_score)
```

```
Test accuracy: 0.8482142857142857
```

Ans.

訓練データ数 # training data: 336
テストデータ数 # test data: 112
識別率 correct rate: 0.848

In [ ]: