

Chapter 4: Demo

```
In [0]: # Load libraries
import numpy as np
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
```

```
In [0]: text_data = np.array(['I love Vietnam. Vietnam!',
                              'Vietnam is best',
                              'Hochiminh is the biggest city of Vietnam'])
```

```
In [0]: # Create target vector
y = np.array([0,0,1])
```

```
In [0]: # Create bag of words
count = CountVectorizer()
count.fit(text_data)
bag_of_words = count.transform(text_data)
bag_of_words
```

```
Out[3]: <3x9 sparse matrix of type '<class 'numpy.int64'>'
        with 12 stored elements in Compressed Sparse Row format>
```

```
In [0]: # Create feature matrix
X = bag_of_words.toarray()
X
```

```
Out[4]: array([[0, 0, 0, 0, 0, 1, 0, 0, 2],
               [1, 0, 0, 0, 1, 0, 0, 0, 1],
               [0, 1, 1, 1, 1, 0, 1, 1, 1]])
```

```
In [0]: # https://scikit-learn.org/stable/modules/generated/sklearn.naive\_bayes.MultinomialNB.html
# Create multinomial naive Bayes object
clf = MultinomialNB()

# Train model
model = clf.fit(X, y)
```

```
In [0]: import numpy as np
# Kiểm tra độ chính xác
print("The prediction accuracy is: ", model.score(X,y)*100,"%")
```

The prediction accuracy is: 100.0 %

```
In [0]: # Create new observation
test_data = np.array(['Vietnam is beautiful country',
                      'I live in Hochiminh city'])
X_test = count.transform(test_data)
X_test
```

```
Out[8]: <2x9 sparse matrix of type '<class 'numpy.int64'>'
        with 4 stored elements in Compressed Sparse Row format>
```

```
In [0]: X_test = X_test.toarray()
X_test
```

```
Out[9]: array([[0, 0, 0, 0, 1, 0, 0, 0, 1],
               [0, 0, 1, 1, 0, 0, 0, 0, 0]])
```

```
In [0]: # Predict new data
y_pred = model.predict(X_test)
y_pred
```

```
Out[10]: array([0, 1])
```

Save model to file

```
In [0]: # import pickle
# pkl_filename = "model.pkl"
# with open(pkl_filename, 'wb') as file:
#     pickle.dump(model, file)
```

```
In [0]: # # Đọc model
# with open(pkl_filename, 'rb') as file:
#     saved_model = pickle.load(file)
```