

```
In [1]: # Load libraries
import numpy as np
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
```

```
In [2]: text_data = np.array(['I love Vietnam. Vietnam!',
                              'Vietnam is best',
                              'Hochiminh is the biggest city of Vietnam'])
```

```
In [3]: # 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 [4]: # 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]], dtype=int64)
```

```
In [5]: # Create target vector
y = np.array([0,0,1])
```

```
In [6]: # Create multinomial naive Bayes object with prior probabilities of each class
clf = MultinomialNB(class_prior=[0.25, 0.5])

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

```
In [7]: # 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[7]: <2x9 sparse matrix of type '<class 'numpy.int64'>'
        with 4 stored elements in Compressed Sparse Row format>
```

```
In [8]: # Predict new data
y_pred = model.predict(X_test)
y_pred
```

```
Out[8]: array([0, 1])
```

```
In [ ]:
```

