In [2]:

```
import numpy as np
import pandas as pd
import sklearn

docs = pd.read_csv('example_train1.csv')
#text in column 1, classifier in column 2.
docs
```

Out[2]:

	Document	Class
0	techlov is a great educational institution.	education
1	Educational greatness depends on ethics	education
2	A story of great ethics and educational greatness	education
3	Sholey is a great cinema	cinema
4	good movie depends on good story	cinema

In [3]:

```
# convert label to a numerical variable
docs['Class'] = docs.Class.map({'cinema':0, 'education':1})
docs
```

Out[3]:

	Document	Class
0	techlov is a great educational institution.	1
1	Educational greatness depends on ethics	1
2	A story of great ethics and educational greatness	1
3	Sholey is a great cinema	0
4	good movie depends on good story	0

In [5]:

```
numpy_array = docs.to_numpy()
X = numpy_array[:,0]
Y = numpy_array[:,1]
Y = Y.astype('int')
print("X")
print(X)
print("Y")
print(Y)
```

```
X
['techlov is a great educational institution.'
 'Educational greatness depends on ethics'
 'A story of great ethics and educational greatness'
 'Sholey is a great cinema' 'good movie depends on good story']
Y
[1 1 1 0 0]
```

In [6]:

```
# create an object of CountVectorizer() class
from sklearn.feature_extraction.text import CountVectorizer
vec = CountVectorizer( )
```

In [7]:

```
vec.fit(X)
vec.vocabulary_
```

Out[7]:

```
{'techlov': 15,
 'is': 9,
 'great': 6,
 'educational': 3,
 'institution': 8,
 'greatness': 7,
 'depends': 2,
 'on': 12,
 'ethics': 4,
 'story': 14,
 'of': 11,
 'and': 0,
 'sholey': 13,
 'cinema': 1,
 'good': 5,
 'movie': 10}
```

In [8]:

```
# removing the stop words
vec = CountVectorizer(stop_words='english' )
vec.fit(X)
vec.vocabulary_
```

Out[8]:

```
{'techlov': 11,
 'great': 5,
 'educational': 2,
 'institution': 7,
 'greatness': 6,
 'depends': 1,
 'ethics': 3,
 'story': 10,
 'sholey': 9,
 'cinema': 0,
 'good': 4,
 'movie': 8}
```

In [9]:

```
# printing feature names
print(vec.get_feature_names())
print(len(vec.get_feature_names()))
```

```
['cinema', 'depends', 'educational', 'ethics', 'good', 'great', 'greatness', 'instit
ution', 'movie', 'sholey', 'story', 'techlov']
12
```

In [10]:

```
# another way of representing the features
X_transformed=vec.transform(X)
X_transformed
```

Out[10]:

```
<5x12 sparse matrix of type '<class 'numpy.int64'>'
        with 20 stored elements in Compressed Sparse Row format>
```

In [11]:

```
print(X_transformed)
```

```
  (0, 2)        1
  (0, 5)        1
  (0, 7)        1
  (0, 11)       1
  (1, 1)        1
  (1, 2)        1
  (1, 3)        1
  (1, 6)        1
  (2, 2)        1
  (2, 3)        1
  (2, 5)        1
  (2, 6)        1
  (2, 10)       1
  (3, 0)        1
  (3, 5)        1
  (3, 9)        1
  (4, 1)        1
  (4, 4)        2
  (4, 8)        1
  (4, 10)       1
```

In [12]:
```
# converting transformed matrix back to an array
# note the high number of zeros
X=X_transformed.toarray()
X
```

Out[12]:
```
array([[0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1],
       [0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0],
       [0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0],
       [1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0],
       [0, 1, 0, 0, 2, 0, 0, 0, 1, 0, 1, 0]], dtype=int64)
```

In [13]:
```
# converting matrix to dataframe
pd.DataFrame(X, columns=vec.get_feature_names())
```

Out[13]:

	cinema	depends	educational	ethics	good	great	greatness	institution	movie	sholey	story
0	0	0	1	0	0	1	0	1	0	0	0
1	0	1	1	1	0	0	1	0	0	0	0
2	0	0	1	1	0	1	1	0	0	0	1
3	1	0	0	0	0	1	0	0	0	1	0
4	0	1	0	0	2	0	0	0	1	0	1

In [ ]: