In [9]:

```
from sklearn.datasets import fetch_20newsgroups
```

In [10]:

```
categories = ['alt.atheism', 'soc.religion.christian', 'comp.graphics', 'sci.med']
news_train = fetch_20newsgroups(subset = 'train', categories = categories, shuffle = True)
news_test = fetch_20newsgroups(subset = 'test', categories = categories, shuffle = True)
news_train.target_names
```

Out[10]:

```
['alt.atheism', 'comp.graphics', 'sci.med', 'soc.religion.christian']
```

In [11]:

```
from sklearn.feature_extraction.text import CountVectorizer
count = CountVectorizer()
X_train = count.fit_transform(news_train.data)
X_train.shape
print(X_train)
```

```
  (0, 14887)    1
  (0, 29022)    1
  (0, 8696)     4
  (0, 4017)     2
  (0, 33256)    2
  (0, 21661)    3
  (0, 9031)     3
  (0, 31077)    1
  (0, 9805)     2
  (0, 17366)    1
  (0, 32493)    4
  (0, 16916)    2
  (0, 19780)    2
  (0, 17302)    2
  (0, 23122)    1
  (0, 25663)    1
  (0, 16881)    1
  (0, 16082)    1
  (0, 23915)    1
  (0, 32142)    5
  (0, 33597)    2
  (0, 20253)    1
  (0, 587)      1
  (0, 12051)    1
  (0, 5201)     1
  :     :
  (2256, 13740) 1
  (2256, 14662) 1
  (2256, 20201) 1
  (2256, 12443) 6
  (2256, 30325) 3
  (2256, 4610)  1
  (2256, 33844) 1
  (2256, 17354) 1
  (2256, 26998) 1
  (2256, 20277) 1
  (2256, 20695) 1
  (2256, 20702) 1
  (2256, 9649)  1
  (2256, 9086)  1
  (2256, 26254) 1
  (2256, 17133) 2
  (2256, 4490)  1
  (2256, 13720) 1
  (2256, 5016)  1
  (2256, 9632)  1
  (2256, 11824) 1
  (2256, 29993) 1
  (2256, 1298)  1
  (2256, 2375)  1
  (2256, 3921)  1
```

In [12]:

```
from sklearn.feature_extraction.text import TfidfTransformer
tfid = TfidfTransformer()
X_train_tfid = tfid.fit_transform(X_train)
X_train_tfid.shape
print(X_train_tfid)
```

```
  (0, 35416)    0.1348710554299733
  (0, 35312)    0.0312703097833574
  (0, 34775)    0.034481472140846715
  (0, 34755)    0.043341654399042764
  (0, 33915)    0.0999409997803694
  (0, 33597)    0.06567578043186388
  (0, 33572)    0.09313007554599557
  (0, 33256)    0.11819702490105698
  (0, 32493)    0.07283773941616518
  (0, 32391)    0.12806013119559947
  (0, 32270)    0.023871142738151236
  (0, 32142)    0.08865416253721688
  (0, 32135)    0.04910237380446671
  (0, 32116)    0.10218403421141944
  (0, 31915)    0.08631915131162177
  (0, 31077)    0.016797806021219684
  (0, 30623)    0.0686611288079694
  (0, 29022)    0.1348710554299733
  (0, 28619)    0.047271576160535234
  (0, 27836)    0.06899050810672397
  (0, 26175)    0.08497460943470851
  (0, 25663)    0.034290706362898604
  (0, 25361)    0.11947938145690981
  (0, 25337)    0.04935883383975408
  (0, 24677)    0.09796250319482307
  :     :
  (2256, 13720) 0.0969927054646086
  (2256, 13521) 0.06264742916622883
  (2256, 13498) 0.08574361554718753
  (2256, 12626) 0.047531848081675473
  (2256, 12443) 0.5533848656066114
  (2256, 11824) 0.12028503503707108
  (2256, 9649)  0.09916899209319778
  (2256, 9632)  0.11395377721095845
  (2256, 9338)  0.05248982587156077
  (2256, 9086)  0.1056108470261161
  (2256, 9072)  0.06784313233467505
  (2256, 7766)  0.030117682035074988
  (2256, 7743)  0.08847991007710146
  (2256, 6507)  0.26543973023130435
  (2256, 6430)  0.04825278674712813
  (2256, 5698)  0.032261754952242205
  (2256, 5529)  0.031207627784231296
  (2256, 5016)  0.12302132956698503
  (2256, 4938)  0.032031796462786304
  (2256, 4610)  0.0927607242624837
  (2256, 4490)  0.11395377721095845
  (2256, 3921)  0.13532523144219466
  (2256, 2375)  0.12625767908616808
  (2256, 1298)  0.12625767908616808
  (2256, 587)   0.06304633984640515
```

In [13]:

```
from sklearn.naive_bayes import MultinomialNB
a = MultinomialNB().fit(X_train_tfid, news_train.target)
```

In [14]:

```
X_test = count.transform(news_test.data)
X_test_tfid = tfid.transform(X_test)
predicted = a.predict(X_test_tfid)
print(predicted)
```

```
[2 2 3 ... 2 2 1]
```

In [15]:

```
doc = ['god is love', 'openGL on the GPU is fast']
X_new_counts = count.transform(doc)
X_new_tfid = tfid.transform(X_new_counts)
predicted = a.predict(X_new_tfid)
for x in predicted:
    print(x)
```

```
3
1
```

In [16]:

```
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
```

In [17]:

```
text_clf = Pipeline([('vect', TfidfVectorizer()), ('clf', MultinomialNB())])
text_clf.fit(news_train.data, news_train.target)
predicted = text_clf.predict(news_test.data)
```

In [18]:

```
from sklearn import metrics
from sklearn.metrics import accuracy_score
import numpy as np
```

In [19]:

```
print('Accuracy achieved is ' + str(np.mean(predicted == news_test.target)))
print(metrics.classification_report(news_test.target, predicted, target_names =
news_test.target_names)), metrics.confusion_matrix(news_test.target, predicted)
```

```
Accuracy achieved is 0.8348868175765646
                        precision    recall  f1-score   support

           alt.atheism       0.97      0.60      0.74       319
         comp.graphics       0.96      0.89      0.92       389
               sci.med       0.97      0.81      0.88       396
soc.religion.christian       0.65      0.99      0.78       398

              accuracy                           0.83      1502
             macro avg       0.89      0.82      0.83      1502
          weighted avg       0.88      0.83      0.84      1502
```

Out[19]:

```
(None,
 array([[192,   2,   6, 119],
        [  2, 347,   4,  36],
        [  2,  11, 322,  61],
        [  2,   2,   1, 393]], dtype=int64))
```

In [ ]: