# 6: classify documents by bayesian classifier model

***load dataset***

```
from sklearn.datasets import fetch_20newsgroups

docs_train = fetch_20newsgroups(subset='train')

attributes = docs_train.target_names
```

***import necessary modules***

```
from sklearn.feature_extraction.text import CountVectorizer

from sklearn.feature_extraction.text import TfidfTransformer

from sklearn.naive_bayes import MultinomialNB
```

***create a pipeline for workflow***

```
from sklearn.pipeline import Pipeline

text_clf = Pipeline([  ('vect',  CountVectorizer()   ),
                       ('tfidf', TfidfTransformer()  ),
                       ('clf',   MultinomialNB()     ),
                    ])

text_clf.fit( docs_train.data, docs_train.target )
```

***make predictions over test dataset***

```
docs_test = fetch_20newsgroups( subset='test' )

predicted = text_clf.predict(docs_test.data)
```

***print accuracy***

```
from sklearn import metrics

percent =  round( metrics.accuracy_score( docs_test.target, predicted) * 100,  2)



print("Accuracy is {0}% ".format(percent))
```

```
Accuracy is 77.39%
```

***classify a few sentences***

```
statements = ['computers are incredible machines', 'microsoft and windows',  'India and christianity']

classifications = [ attributes[value] for value in text_clf.predict(statements) ]

print(classifications)
```

```
['comp.sys.mac.hardware', 'comp.os.ms-windows.misc', 'soc.religion.christian']
```

```
# just in case

# from sklearn import metrics
# print(metrics.classification_report(docs_test.target, predicted) )
```

```

```