from sklearn.datasets import fetch\_20newsgroups

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import classification\_report

import numpy as np

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

twenty\_train = fetch\_20newsgroups(subset='train',categories=categories,shuffle=True)

twenty\_test = fetch\_20newsgroups(subset='test',categories=categories,shuffle=True)

print(len(twenty\_train.data))

print(len(twenty\_test.data))

print(twenty\_train.target\_names)

print("\n".join(twenty\_train.data[0].split("\n")))

print(twenty\_train.target[0])

from sklearn.feature\_extraction.text import CountVectorizer

count\_vect = CountVectorizer()

X\_train\_tf = count\_vect.fit\_transform(twenty\_train.data)

from sklearn.feature\_extraction.text import TfidfTransformer

tfidf\_transformer = TfidfTransformer()

X\_train\_tfidf = tfidf\_transformer.fit\_transform(X\_train\_tf)

X\_train\_tfidf.shape

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import accuracy\_score

from sklearn import metrics

mod = MultinomialNB()

mod.fit(X\_train\_tfidf, twenty\_train.target)

X\_test\_tf = count\_vect.transform(twenty\_test.data)

X\_test\_tfidf = tfidf\_transformer.transform(X\_test\_tf)

predicted = mod.predict(X\_test\_tfidf)

print("Accuracy:", accuracy\_score(twenty\_test.target, predicted))

print(classification\_report(twenty\_test.target,predicted,target\_names=twenty\_test.target\_names))

print("confusion matrix is \n",metrics.confusion\_matrix(twenty\_test.target, predicted))