Create and train a Naive Bayes classifier in naive_bayes/nb_author_id.py. Use it to make predictions for the test set. What is the accuracy?

When training you may see the following error: UserWarning: Duplicate scores. Result may depend on feature ordering.There are probably duplicate features, or you used a classification score for a regression task. warn("Duplicate scores. Result may depend on feature ordering.")

This is a warning that two or more words happen to have the same usage patterns in the emails--as far as the algorithm is concerned, this means that two features are the same. Some algorithms will actually break (mathematically won't work) or give multiple different answers (depending on feature ordering) when there are duplicate features and sklearn is giving us a warning. Good information, but not something we have to worry about.

In [1]:
```
#!/usr/bin/python

"""
    This is the code to accompany the Lesson 1 (Naive Bayes) mini-project.

    Use a Naive Bayes Classifier to identify emails by their authors

    authors and labels:
    Sara has label 0
    Chris has label 1
"""

import sys
from time import time
sys.path.append("../tools/")
from email_preprocess import preprocess


### features_train and features_test are the features for the training
### and testing datasets, respectively
### labels_train and labels_test are the corresponding item labels
features_train, features_test, labels_train, labels_test = preprocess()

t0 = time()


#########################################################
### your code goes here ###
from sklearn.naive_bayes import GaussianNB
clas = GaussianNB()
clas.fit(features_train,labels_train)
acc = clas.score(features_test,labels_test)
print acc

#########################################################

print "training time:",round(time()-t0,3),"s"
```

```
C:\Users\Andrew\Anaconda3\envs\conda2\lib\site-packages\sklearn\cross_validatio
n.py:41: DeprecationWarning: This module was deprecated in version 0.18 in favo
r of the model_selection module into which all the refactored classes and funct
ions are moved. Also note that the interface of the new CV iterators are differ
ent from that of this module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)

no. of Chris training emails: 7936
no. of Sara training emails: 7884
0.9732650739476678
training time: 2.643 s
```

An important topic that we didn't explicitly talk about is the time to train and test our algorithms. Put in two lines of code, above and below the line fitting your classifier, like this:

t0 = time() < your clf.fit() line of code > print "training time:", round(time()-t0, 3), "s"

Put similar lines of code around the clf.predict() line of code, so you can compare the time to train the classifier and to make predictions with it. What is faster, training or prediction?

Prediction

We will compare the Naive Bayes timing to a couple other algorithms, so note down the speed and accuracy you get and we'll revisit this in the next mini-project.

In [ ]: