**ICA Hints**

**Hint 1**

* Use sklearn.model\_selection and cv.split() for cross validation

**E3**

* Use a for loop to change the value n\_neighbors= and print the accuracy\_score(y\_test,y\_hat) for each value of K.

**Q3**

* The cost of measuring distance with truly sparse data is small--usually smaller than the cost of creating a tree. Sparse data also will not branch well in a KDTree because there are so many zeros. The number of "zero" comparisons will mean each tree needs to be very deep. Therefore the benefit of the KDTree is drastically reduced.

**E4**

* Use a for loop to test your different distance metrics
  + for d in ['l1','l2','cosine']:
    - clf = NearestCentroid(metric=d)

**Q5.**

* Total conditionals = # of features or columns \* number of classes
* There is one "prior" for each class.

**Q6**

* Sparse matrices are much harder to find realistic Gaussian models for because they always have a mean near zero.
* For sparse data, it is probably better (and faster) to use multinomial naive Bayes.
  + An argument can also be made for Bernoulli if binarizing the feature data helps to reduce the complexity of the problem.

**Q7**

* Use a for loop to test both classifiers.
  + for clf in [clf\_mnb,clf\_bnb]:
    - clf.fit(X\_train, y\_train)
* Change the parameter values in the code above to something like 0.001 for alpha and 0.02 for binarize.