Practical 6:
(for Lecture 6: Classification)

Q1. K-NN

Check out the provided program on k-NN, applied to the famous Iris Data set Two parameters are interesting: (i) the ***split*** which is the size of the training subset v test subset (split = .67) means roughly 2/3rds training, 1/3rd testing, (ii) ***k*** which is the size of nearest neighbours. Note, the ***Accuracy*** of model is determined for test sets; it measures how well it classifications work for the unseen set.

Taking ideas about cross-validation into account; you job is first to systematically vary the size of the ***split*** ; exploring a decent number of other values for it >0.0 and <0.9 (to be chosen by you). Also to systematically vary ***k*** on five selected values between 1 and 20.

Then plot the accuracy in a graph for these parameter changes
Now, using this data set, describe an algorithm for doing a 5-fold cross validation on this data set. See can you implement it in Python.

Q2. Bayes Classifiers

Have a look at the nltk Bayes classifier that does the prediction of male/female names based on the last letter in the name.

Think of a new feature that you could extract from the data-set; define a fn for it (modifying **`gender_features`**) and see what is learned from it in the classification.

Compare the accuracy of your feature versus that of the last_letter feature and discuss.

Q3. SVMs

So, this is just a quick use of an SVM; to show you how easy it is. It involves learning digit recognition.

NB to install packages: sklearn, matplotlib, scipy; (problems around installing scipy in on mac with Pycharm; but works with "port install py34-scipy"

Now run three different training configurations and then test the outputs for 5-10 different digits; report results.