#P2-Q1

import numpy as np

from sklearn.naive\_bayes import GaussianNB

from sklearn import tree

from sklearn.metrics import confusion\_matrix

from sklearn import linear\_model

data = np.loadtxt('random\_classification.txt')

Y=data[:,-1]

X=data[:,:len(data[0])-1]

X\_train = X[:400,:]

X\_test = X[400:,:]

Y\_train = Y[:400]

Y\_test = Y[400:]

clf\_NB = GaussianNB()

clf\_DT = tree.DecisionTreeClassifier()

clf\_LR = linear\_model.LogisticRegression()

Y\_pred\_NB = clf\_NB.fit(X\_train, Y\_train).predict(X\_test)

Y\_pred\_DT = clf\_DT.fit(X\_train, Y\_train).predict(X\_test)

Y\_pred\_LR = clf\_LR.fit(X\_train, Y\_train).predict(X\_test)

meas\_NB = perf\_measure(Y\_test, Y\_pred\_NB)

meas\_DF = perf\_measure(Y\_test, Y\_pred\_DT)

meas\_LR = perf\_measure(Y\_test, Y\_pred\_LR) #final answer

def perf\_measure(y\_test, y\_pred):

CM = confusion\_matrix(y\_test, y\_pred)

TN = CM[0][0]

FN = CM[1][0]

TP = CM[1][1]

FP = CM[0][1]

sensitivity = float(TP)/float((TP+FN))

specificity = float(TN)/float((TN+FP))

acc = float((TP+TN))/float((TP+FP+TN+FN))

return (acc, sensitivity, specificity)

#P2-Q2

import numpy as np

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import confusion\_matrix

from sklearn import linear\_model

from sklearn import svm

data = np.loadtxt('breast\_cancer.txt')

Y=data[:,-1]

X=data[:,:len(data[0])-1]

X\_train = X[:500,:]

X\_test = X[500:,:]

Y\_train = Y[:500]

Y\_test = Y[500:]

clf\_NB = GaussianNB()

clf\_LR = linear\_model.LogisticRegression()

clf\_SVM = svm.SVC()

Y\_pred\_NB = clf\_NB.fit(X\_train, Y\_train).predict(X\_test)

Y\_pred\_LR = clf\_LR.fit(X\_train, Y\_train).predict(X\_test)

Y\_pred\_SVM = clf\_SVM.fit(X\_train, Y\_train).predict(X\_test)

meas\_NB = perf\_measure(Y\_test, Y\_pred\_NB)

meas\_LR = perf\_measure(Y\_test, Y\_pred\_LR)

meas\_SVM = perf\_measure(Y\_test, Y\_pred\_SVM)

#P2-Q3

import numpy as np

from sklearn import linear\_model

data = np.loadtxt('data\_lasso.txt')

Y=data[:,-1]

X=data[:,:len(data[0])-1]

lasso\_model = linear\_model.Lasso()

lasso\_model.fit(X,Y)

bar(range(len(lasso\_model.coef\_)),lasso\_model.coef\_)

xlabel("Index of feature")

ylabel("Coeff")

np.nonzero(lasso\_model.coef\_)

#P3. Only for random\_classification data. You can get breast\_cancer once you change the loading file name

from sklearn.cross\_validation import KFold

data = np.loadtxt('random\_classification.txt')

Y=data[:,-1]

X=data[:,:len(data[0])-1]

n\_sample = len(X[:,0])

kf = KFold(n\_sample,n\_folds=5)

sum\_scores = np.zeros((3,3))

for train, test in kf:

X\_train, X\_test, Y\_train, Y\_test = X[train], X[test], Y[train], Y[test]

Y\_pred\_NB = clf\_NB.fit(X\_train, Y\_train).predict(X\_test)

Y\_pred\_DT = clf\_DT.fit(X\_train, Y\_train).predict(X\_test)

Y\_pred\_LR = clf\_LR.fit(X\_train, Y\_train).predict(X\_test)

meas\_NB = perf\_measure(Y\_test, Y\_pred\_NB)

meas\_DF = perf\_measure(Y\_test, Y\_pred\_DT)

meas\_LR = perf\_measure(Y\_test, Y\_pred\_LR)

sum\_scores[0,:] += meas\_NB

sum\_scores[1,:] +=meas\_DF

sum\_scores[2,:] += meas\_LR

print sum\_scores/5