# Joshua Ewart, CSCI 481 Data Mining, HomeWork 3

#include libraries

import numpy as numPy

import sys

from sklearn import svm

from sklearn import metrics

#Get fold for train

train = sys.argv[1];

#Get folds for comparison

test1 = sys.argv[2];

test2 = sys.argv[3];

test3 = sys.argv[4];

#create data from files

#17 elements, first is class

#train

trainInfo = numPy.loadtxt(train, dtype= 'str', delimiter= ',');

#create numeric data

trainStr = trainInfo[:, 1:16];

trainData = trainStr.astype('float');

trainClasses = trainInfo[:, 0];

#test1

t1INFO = numPy.loadtxt(test1, dtype= 'str', delimiter= ',');

#create numeric data

t1STR = t1INFO[:, 1:16];

test1DATA = t1STR.astype('float');

test1CLASSES = t1INFO[:, 0];

#test2

t2INFO = numPy.loadtxt(test2, dtype= 'str', delimiter= ',');

#create numeric data

t2STR = t2INFO[:, 1:16];

test2DATA = t2STR.astype('float');

test2CLASSES = t2INFO[:, 0];

#test3

t3INFO = numPy.loadtxt(test3, dtype= 'str', delimiter= ',');

#create numeric data

t3STR = t3INFO[:, 1:16];

test3DATA = t3STR.astype('float');

test3CLASSES = t3INFO[:, 0];

#data read in. now use SVM

#first do linear

linCLASSIFY = svm.SVC(C = .25, kernel='linear');

linCLASSIFY.fit(trainData,trainClasses);

lin1Predict = linCLASSIFY.predict(test1DATA);

lin2Predict = linCLASSIFY.predict(test2DATA);

lin3Predict = linCLASSIFY.predict(test3DATA);

lin1Confusion = metrics.confusion\_matrix(test1CLASSES, lin1Predict);

lin2Confusion = metrics.confusion\_matrix(test2CLASSES, lin2Predict);

lin3Confusion = metrics.confusion\_matrix(test3CLASSES, lin3Predict);

print(lin1Confusion);

lin1Acc= float((lin1Confusion[0,0] +lin1Confusion[1,1]))/108;

lin2Acc= float((lin2Confusion[0,0] +lin2Confusion[1,1]))/108;

lin3Acc= float((lin3Confusion[0,0] +lin3Confusion[1,1]))/111;

LINAVG = float(lin1Acc+lin2Acc+lin3Acc)/3;

print(LINAVG);

print("lin^ gau[below]");

#now do Gaussian

GauCLASSIFY = svm.SVC(C = .25, kernel='rbf');

GauCLASSIFY.fit(trainData,trainClasses);

Gau1Predict = GauCLASSIFY.predict(test1DATA);

Gau2Predict = GauCLASSIFY.predict(test2DATA);

Gau3Predict = GauCLASSIFY.predict(test3DATA);

Gau1Confusion = metrics.confusion\_matrix(test1CLASSES, Gau1Predict);

Gau2Confusion = metrics.confusion\_matrix(test2CLASSES, Gau2Predict);

Gau3Confusion = metrics.confusion\_matrix(test3CLASSES, Gau3Predict);

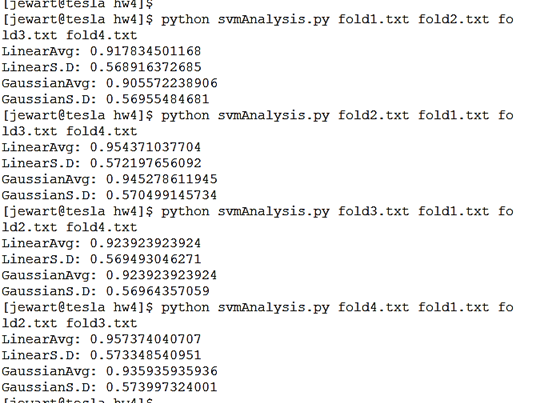
Gau1Acc= float((Gau1Confusion[0,0] +Gau1Confusion[1,1]))/108;

Gau2Acc= float((Gau2Confusion[0,0] +Gau2Confusion[1,1]))/108;

Gau3Acc= float((Gau3Confusion[0,0] +Gau3Confusion[1,1]))/111;

GAUAVG = float(Gau1Acc+Gau2Acc+Gau3Acc)/3;

print(GAUAVG);



1. Frequent Itemset results
   1. 402 sets
   2. 6 (62.5287), 16 (61.8391), 200 (61.3793), 3 200 (53.1034), 3 (58.1609), 14 6 (49.1954), 14 (57.0115), 8 200 (50.1149), 8 3 (49.4253), 8 (55.6322)
   3. 75
   4. 94

100 <- 4 10 14 (20.6897, 96.6667)

100 <- 4 10 (21.6092, 96.8085)

100 <- 4 5 16 (20.9195, 96.7033)

100 <- 4 5 14 16 (20.4598, 96.6292)

100 <- 4 5 10 (20.2299, 96.5909)

100 <- 4 16 (22.5287, 95.9184)

100 <- 4 14 16 (22.069, 95.8333)

100 <- 12 4 13 14 (27.5862, 95.8333)

100 <- 12 4 13 5 (27.8161, 95.8678)

100 <- 12 4 13 (28.2759, 95.935)

100 <- 5 16 6 (24.3678, 75.4717)

100 <- 5 16 (27.1264, 75.4237)

100 <- 5 14 6 (41.3793, 76.1111)

100 <- 5 14 16 6 (22.5287, 79.5918)

100 <- 5 14 16 (25.0575, 79.8165)

100 <- 5 14 (44.5977, 76.8041)

100 <- 5 10 6 (22.9885, 79)

100 <- 5 10 14 6 (21.6092, 79.7872)

100 <- 5 10 14 (22.9885, 81)

100 <- 5 10 (24.5977, 79.4393)

100 <- 13 14 16 (22.069, 75)

100 <- 13 5 6 (36.7816, 77.5)

100 <- 13 5 16 (21.1494, 78.2609)

100 <- 13 5 14 6 (35.1724, 77.7778)

100 <- 13 5 14 16 (20, 80.4598)

100 <- 13 5 14 (37.2414, 78.3951)

100 <- 13 5 (39.0805, 77.6471)

100 <- 4 6 (36.7816, 91.875)

100 <- 4 16 (22.5287, 95.9184)

100 <- 4 14 6 (34.7126, 92.053)

100 <- 4 14 16 (22.069, 95.8333)

100 <- 4 14 (38.6207, 92.2619)

100 <- 4 10 14 (20.6897, 96.6667)

100 <- 4 10 (21.6092, 96.8085)

100 <- 4 5 6 (35.8621, 92.3077)

100 <- 4 5 16 (20.9195, 96.7033)

100 <- 4 5 14 6 (34.023, 91.8919)

100 <- 4 5 14 16 (20.4598, 96.6292)

100 <- 4 5 14 (36.7816, 92.5)

100 <- 4 5 10 (20.2299, 96.5909)

100 <- 4 5 (38.6207, 92.8571)

100 <- 4 13 6 (31.2644, 92.6471)

100 <- 4 13 14 6 (29.8851, 93.0769)

100 <- 4 13 14 (31.954, 93.5252)

100 <- 4 13 5 6 (30.5747, 93.2331)

100 <- 4 13 5 14 6 (29.4253, 92.9688)

100 <- 4 13 5 14 (31.2644, 93.3824)

100 <- 4 13 5 (32.4138, 93.617)

100 <- 4 13 (33.3333, 93.1034)

100 <- 4 2 (20, 86.2069)

100 <- 4 (40.6897, 92.0904)

100 <- 12 6 (36.5517, 79.8742)

100 <- 12 16 6 (20, 81.6092)

100 <- 12 16 (22.069, 80.2083)

100 <- 12 14 6 (34.023, 81.7568)

100 <- 12 14 16 (20.2299, 85.2273)

100 <- 12 14 (36.3218, 81.6456)

100 <- 12 10 (20.9195, 81.3187)

100 <- 12 5 6 (32.4138, 87.234)

100 <- 12 5 14 6 (31.2644, 86.7647)

100 <- 12 5 14 (33.1034, 87.5)

100 <- 12 5 (34.2529, 87.9195)

100 <- 12 13 6 (30.3448, 85.6061)

100 <- 12 13 14 6 (28.9655, 86.5079)

100 <- 12 13 14 (30.5747, 86.4662)

100 <- 12 13 5 6 (28.7356, 88)

100 <- 12 13 5 14 6 (28.046, 87.7049)

100 <- 12 13 5 14 (29.4253, 88.2812)

100 <- 12 13 5 (30.1149, 88.5496)

100 <- 12 13 (31.954, 85.6115)

100 <- 12 4 6 (30.5747, 94.7368)

100 <- 12 4 14 6 (29.4253, 94.5312)

100 <- 12 4 14 (31.2644, 94.8529)

100 <- 12 4 5 6 (29.8851, 94.6154)

100 <- 12 4 5 14 6 (28.7356, 94.4)

100 <- 12 4 5 14 (30.5747, 94.7368)

100 <- 12 4 5 (31.7241, 94.9275)

100 <- 12 4 13 6 (26.8966, 95.7265)

100 <- 12 4 13 14 6 (26.2069, 95.614)

100 <- 12 4 13 14 (27.5862, 95.8333)

100 <- 12 4 13 5 6 (26.4368, 95.6522)

100 <- 12 4 13 5 14 6 (25.7471, 95.5357)

100 <- 12 4 13 5 14 (27.1264, 95.7627)

100 <- 12 4 13 5 (27.8161, 95.8678)

100 <- 12 4 13 (28.2759, 95.935)

100 <- 12 4 (32.4138, 95.0355)

100 <- 12 (39.3103, 78.9474)

200 <- 1 9 7 8 3 (23.4483, 99.0196)

200 <- 1 9 8 3 (25.2874, 98.1818)

200 <- 1 9 7 3 (24.1379, 98.0952)

200 <- 15 9 7 (27.1264, 98.3051)

200 <- 15 9 7 3 (24.8276, 98.1481)

200 <- 15 9 7 8 (26.4368, 98.2609)

200 <- 15 9 7 8 3 (24.3678, 98.1132)

200 <- 15 9 (29.8851, 97.6923)

200 <- 15 9 8 (28.9655, 97.619)

200 <- 11 8 3 (20.2299, 97.7273)

200 <- 3 6 (26.2069, 85.0877)

200 <- 3 16 (39.7701, 88.4393)

200 <- 3 (58.1609, 91.3043)

200 <- 8 6 (21.6092, 82.9787)

200 <- 8 16 (38.6207, 87.5)

200 <- 8 (55.6322, 90.0826)

200 <- 8 3 16 (34.2529, 92.6174)

200 <- 8 3 (49.4253, 94.4186)

200 <- 7 16 (39.5402, 79.6512)

200 <- 7 (54.9425, 83.682)

200 <- 7 3 16 (32.1839, 89.2857)

200 <- 7 3 (46.2069, 91.5423)

200 <- 7 8 16 (33.7931, 87.0748)

200 <- 7 8 (48.2759, 90)

200 <- 7 8 3 16 (29.6552, 92.2481)

200 <- 7 8 3 (42.9885, 94.1176)

200 <- 10 3 16 (22.9885, 83)

200 <- 10 3 (29.6552, 86.8217)

200 <- 10 8 16 (22.7586, 82.8283)

200 <- 10 8 (28.7356, 86.4)

200 <- 10 8 3 (25.2874, 90.9091)

200 <- 10 7 16 (22.9885, 75)

200 <- 10 7 (28.5057, 79.8387)

200 <- 10 7 3 (23.4483, 88.2353)

200 <- 10 7 8 (25.0575, 85.3211)

200 <- 10 7 8 3 (21.6092, 90.4255)

200 <- 9 16 (34.023, 87.8378)

200 <- 9 (47.5862, 90.8213)

200 <- 9 3 16 (29.6552, 93.0233)

200 <- 9 3 (41.3793, 95)

200 <- 9 8 16 (31.4943, 91.2409)

200 <- 9 8 (44.1379, 93.2292)

200 <- 9 8 3 16 (28.046, 95.082)

200 <- 9 8 3 (39.3103, 96.4912)

200 <- 9 7 16 (30.3448, 88.6364)

200 <- 9 7 (41.8391, 91.7582)

200 <- 9 7 3 16 (26.2069, 93.8596)

200 <- 9 7 3 (37.0115, 95.6522)

200 <- 9 7 8 16 (28.5057, 91.129)

200 <- 9 7 8 (39.7701, 93.6416)

200 <- 9 7 8 3 16 (25.0575, 95.4128)

200 <- 9 7 8 3 (35.6322, 96.7742)

200 <- 9 10 16 (20.4598, 83.1461)

200 <- 9 10 (24.3678, 85.8491)

200 <- 9 10 3 (20.9195, 91.2088)

200 <- 9 10 8 (22.069, 89.5833)

200 <- 9 10 7 (21.6092, 87.234)

200 <- 9 10 7 8 (20, 89.6552)

200 <- 2 3 (25.2874, 92.7273)

200 <- 2 8 (22.069, 94.7917)

200 <- 2 8 3 (20.4598, 95.5056)

200 <- 2 7 (20, 86.2069)

200 <- 1 16 (27.5862, 81.6667)

200 <- 1 (42.9885, 83.4225)

200 <- 1 3 16 (22.7586, 91.9192)

200 <- 1 3 (34.7126, 94.0397)

200 <- 1 8 16 (21.8391, 90.5263)

200 <- 1 8 (33.5632, 92.4658)

200 <- 1 8 3 16 (20.4598, 93.2584)

200 <- 1 8 3 (31.4943, 94.8905)

200 <- 1 7 16 (21.1494, 86.9565)

200 <- 1 7 (32.1839, 89.2857)

200 <- 1 7 3 (29.4253, 93.75)

200 <- 1 7 8 (29.6552, 93.0233)

200 <- 1 7 8 3 (28.046, 95.082)

200 <- 1 9 (27.8161, 95.8678)

200 <- 1 9 3 (26.2069, 97.3684)

200 <- 1 9 8 (26.6667, 96.5517)

200 <- 1 9 8 3 (25.2874, 98.1818)

200 <- 1 9 7 (25.2874, 96.3636)

200 <- 1 9 7 3 (24.1379, 98.0952)

200 <- 1 9 7 8 (24.5977, 97.1963)

200 <- 1 9 7 8 3 (23.4483, 99.0196)

200 <- 1 2 (20.2299, 81.8182)

200 <- 15 16 (27.3563, 90.7563)

200 <- 15 (40, 91.954)

200 <- 15 3 16 (23.6782, 94.1748)

200 <- 15 3 (34.7126, 95.3642)

200 <- 15 8 16 (24.1379, 94.2857)

200 <- 15 8 (34.7126, 96.0265)

200 <- 15 8 3 16 (22.069, 94.7917)

200 <- 15 8 3 (31.7241, 96.3768)

200 <- 15 7 16 (23.2184, 91.0891)

200 <- 15 7 (33.3333, 93.1034)

200 <- 15 7 3 16 (20, 95.4023)

200 <- 15 7 3 (29.6552, 96.124)

200 <- 15 7 8 16 (21.3793, 94.6237)

200 <- 15 7 8 (31.0345, 96.2963)

200 <- 15 7 8 3 (28.5057, 96.7742)

200 <- 15 9 16 (21.1494, 96.7391)

200 <- 15 9 (29.8851, 97.6923)

200 <- 15 9 3 (27.1264, 97.4576)

200 <- 15 9 8 16 (20.4598, 96.6292)

200 <- 15 9 8 (28.9655, 97.619)

200 <- 15 9 8 3 (26.4368, 97.3913)

200 <- 15 9 7 (27.1264, 98.3051)

200 <- 15 9 7 3 (24.8276, 98.1481)

200 <- 15 9 7 8 (26.4368, 98.2609)

200 <- 15 9 7 8 3 (24.3678, 98.1132)

200 <- 15 1 (22.5287, 95.9184)

200 <- 15 1 3 (21.1494, 96.7391)

200 <- 15 1 8 (20.9195, 96.7033)

200 <- 15 1 8 3 (20, 96.5517)

200 <- 11 6 (21.8391, 78.9474)

200 <- 11 16 (22.069, 86.4583)

200 <- 11 (34.4828, 86)

200 <- 11 3 (25.0575, 97.2477)

200 <- 11 8 (22.2989, 96.9072)

200 <- 11 8 3 (20.2299, 97.7273)

200 <- 11 7 (20, 94.2529)

200 <- 11 2 (20.4598, 84.2697)