Question 1

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 29 13:00:46 2018

@author: tchat

"""

import sklearn

from sklearn.preprocessing import StandardScaler

import csv

import numpy as np

import math

import matplotlib.pyplot as plt

import copy

from sklearn.linear\_model import LinearRegression

from sklearn.multiclass import OneVsRestClassifier

from sklearn.linear\_model import Perceptron

from sklearn.metrics import accuracy\_score

from sklearn.svm import SVC

import matplotlib.pyplot as plt

from scipy.spatial.distance import cdist

from plotSVMBoundaries import plotDecBoundaries

#Import train files and the label files

with open(r'C:\Users\tchat\.spyder-py3\HW 10\HW10\_1\_csv\train\_x.csv') as feature\_train:

feature\_training = csv.reader(feature\_train)

feature\_list=[]

for row in feature\_training:

if len (row) !=0:

feature\_list = feature\_list + [row]

feat\_train\_data = np.array(feature\_list).astype("float")

with open(r'C:\Users\tchat\.spyder-py3\HW 10\HW10\_1\_csv\train\_y.csv') as label\_train:

label\_training = csv.reader(label\_train)

label\_list=[]

for row in label\_training:

if len (row) !=0:

label\_list = label\_list + [row]

label\_train = []

for row in label\_list:

label\_train.append(row[0])

label\_train = np.array(label\_train).astype("float")

#Train the Model

model = SVC(C=1, kernel='linear')

model.fit(feat\_train\_data, label\_train)

s = model.predict(feat\_train\_data)

acc= accuracy\_score(label\_train, s)

plotDecBoundaries(feat\_train\_data, label\_train, model)

Question 2 - a

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 29 13:00:46 2018

@author: tchat

"""

from sklearn.svm import SVC

from sklearn import svm

import csv

import numpy as np

import pandas as pd

from sklearn.metrics import accuracy\_score

from plotSVMBoundaries import plotDecBoundaries

import matplotlib.pyplot as plt

import matplotlib.image as mpimg

from scipy.spatial.distance import cdist

from sklearn.model\_selection import StratifiedKFold

from sklearn.utils import shuffle

import math

df1 = pd.read\_csv(r'C:\Users\tchat\.spyder-py3\HW 10\wine\_csv\feature\_train.csv',header=None)

x=df1.as\_matrix()

x=x[:,0:2:1]

df2 = pd.read\_csv(r'C:\Users\tchat\.spyder-py3\HW 10\wine\_csv\label\_train.csv',header=None)

y=df2.as\_matrix()

y=np.ravel(y)

x,y=shuffle(x,y)

all\_acc=[]

save\_avg\_acc=[]

skf=StratifiedKFold(n\_splits=5,shuffle=True)

cnt=1

for train\_index, test\_index in skf.split(x,y):

x\_train, x\_test=x[train\_index], x[test\_index]

y\_train, y\_test=y[train\_index], y[test\_index]

clf=SVC(C=1.0, kernel='rbf', gamma=1.0)

clf.fit(x\_train,y\_train)

y\_pred=clf.predict(x\_test)

acc=accuracy\_score(y\_test,y\_pred)

all\_acc.append(acc)

cnt+=1

mean=np.mean(all\_acc)

print('average cross validation accuracy: ', mean)

Question 2 - b

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 29 13:00:46 2018

@author: tchat

"""

from sklearn.svm import SVC

import numpy as np

import matplotlib.pyplot as plt

from plotSVMBoundaries import plotDecBoundaries

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import StratifiedKFold

from sklearn.utils import shuffle

wine\_feature\_train = r'C:\Users\tchat\.spyder-py3\HW 10\wine\_csv\feature\_train.csv'

wine\_label\_train = r'C:\Users\tchat\.spyder-py3\HW 10\wine\_csv\label\_train.csv'

X\_train = np.genfromtxt(wine\_feature\_train, delimiter=",")[:,0:2]

y\_train = np.genfromtxt(wine\_label\_train, delimiter=",")

skf = StratifiedKFold(n\_splits = 5, shuffle = True)

Cs = np.logspace(-3, 3, 50)

gammas = np.logspace(-3, 3, 50)

ACC = np.zeros((50,50))

DEV = np.zeros((50,50))

for i, gamma in enumerate(gammas):

for j, C in enumerate(Cs):

acc = []

for train\_index, dev\_index in skf.split(X\_train, y\_train):

X\_cv\_train, X\_cv\_dev = X\_train[train\_index], X\_train[dev\_index]

y\_cv\_train, y\_cv\_dev = y\_train[train\_index], y\_train[dev\_index]

clf = SVC(C = C, kernel = 'rbf', gamma = gamma, )

clf.fit(X\_cv\_train, y\_cv\_train)

acc.append(accuracy\_score(y\_cv\_dev, clf.predict(X\_cv\_dev)))

ACC[i,j] = np.mean(acc)

DEV[i,j] = np.std(acc)

plt.imshow(ACC, interpolation = 'nearest', cmap=plt.cm.Blues)

plt.colorbar()

i, j = np.argwhere(ACC == np.max(ACC))[0]

print('The best pair is C = ' + str(Cs[j]) + ' and gamma = ' + str(gammas[i]))

print('The mean Cross-Validation Accuracy for the best pair = ', ACC[i,j])

print('The Standard deviation for the best pair = ', DEV[i,j])

print('')

Question 2 – c,d

# -\*- coding: utf-8 -\*-

"""

Created on Thu Mar 29 13:00:46 2018

@author: tchat

"""

import numpy as np

import matplotlib.pyplot as plt

from plotSVMBoundaries import plotDecBoundaries

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import StratifiedKFold

from sklearn.utils import shuffle

wine\_feature\_train = r'C:\Users\tchat\.spyder-py3\HW 10\wine\_csv\feature\_train.csv'

wine\_feature\_test = r'C:\Users\tchat\.spyder-py3\HW 10\wine\_csv\feature\_test.csv'

wine\_label\_train = r'C:\Users\tchat\.spyder-py3\HW 10\wine\_csv\label\_train.csv'

wine\_label\_test = r'C:\Users\tchat\.spyder-py3\HW 10\wine\_csv\label\_test.csv'

X\_train = np.genfromtxt(wine\_feature\_train, delimiter=",")[:,0:2]

X\_test = np.genfromtxt(wine\_feature\_test, delimiter=",")[:,0:2]

y\_train = np.genfromtxt(wine\_label\_train, delimiter=",")

y\_test = np.genfromtxt(wine\_label\_test, delimiter=",")

skf = StratifiedKFold(n\_splits = 5, shuffle = True)

Cs = np.logspace(-3, 3, 50)

gammas = np.logspace(-3, 3, 50)

accuracy = np.zeros((50,50))

dev = np.zeros((50,50))

pair\_history = [0, 0]

ACC = []

DEV = []

for t in range(0, 20):

for i, gamma in enumerate(gammas):

for j, C in enumerate(Cs):

acc = []

for train\_index, dev\_index in skf.split(X\_train, y\_train):

X\_cv\_train, X\_cv\_dev = X\_train[train\_index], X\_train[dev\_index]

y\_cv\_train, y\_cv\_dev = y\_train[train\_index], y\_train[dev\_index]

clf = SVC(C = C, kernel = 'rbf', gamma = gamma, decision\_function\_shape = 'ovr')

clf.fit(X\_cv\_train, y\_cv\_train)

acc.append(accuracy\_score(y\_cv\_dev, clf.predict(X\_cv\_dev)))

accuracy[i,j] = np.mean(acc)

dev[i,j] = np.std(acc)

i, j = np.argwhere(accuracy == np.max(accuracy))[0]

pair\_history = np.vstack([pair\_history, [gammas[i], Cs[j]]])

ACC.append(accuracy[i,j])

DEV.append(dev[i,j])

pair\_history = pair\_history[1:]

print('The 20 chosen pairs = \n', pair\_history)

i = np.argwhere(ACC == np.max(ACC))[0]

print('The best pair is gamma = ' + str(pair\_history[int(i),0]) + ' and C = ' + str(pair\_history[int(i),1]))

print('The mean Cross-Validation Accuracy for the best pair = ', ACC[int(i)])

print('The Standard deviation for the best pair = ', DEV[int(i)])

print('')

# Part d

clf1 = SVC(C = pair\_history[int(i),0], kernel = 'rbf', gamma = pair\_history[int(i),1], decision\_function\_shape = 'ovr')

clf1.fit(X\_train, y\_train)

y\_pred = clf1.predict(X\_test)

acc = accuracy\_score(y\_test, y\_pred)

print('Test accuracy = ', acc)

plotDecBoundaries(X\_train, y\_train, clf1)