**SVM Demo 2**

import pandas as pd

df = pd.read\_csv('iris.csv')

df = df.drop(['Id'],axis=1)

target = df['Species']

s = set()

for val in target:

s.add(val)

s = list(s)

rows = list(range(100,150))

df = df.drop(df.index[rows])

import matplotlib.pyplot as plt

x = df['SepalLengthCm']

y = df['PetalLengthCm']

setosa\_x = x[:50]

setosa\_y = y[:50]

versicolor\_x = x[50:]

versicolor\_y = y[50:]

plt.figure(figsize=(8,6))

plt.scatter(setosa\_x,setosa\_y,marker='+',color='green')

plt.scatter(versicolor\_x,versicolor\_y,marker='\_',color='red')

plt.show()

from sklearn.utils import shuffle

from sklearn.model\_selection import train\_test\_split

import numpy as np

**## Drop rest of the features and extract the target values**

df = df.drop(['SepalWidthCm','PetalWidthCm'],axis=1)

Y = []

target = df['Species']

for val in target:

if(val == 'Iris-setosa'):

Y.append(-1)

else:

Y.append(1)

df = df.drop(['Species'],axis=1)

X = df.values.tolist()

**## Shuffle and split the data into training and test set**

X, Y = shuffle(X,Y)

x\_train = []

y\_train = []

x\_test = []

y\_test = []

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, train\_size=0.9)

x\_train = np.array(x\_train)

y\_train = np.array(y\_train)

x\_test = np.array(x\_test)

y\_test = np.array(y\_test)

y\_train = y\_train.reshape(90,1)

y\_test = y\_test.reshape(10,1)

**## Support Vector Machine**

import numpy as np

train\_f1 = x\_train[:,0]

train\_f2 = x\_train[:,1]

train\_f1 = train\_f1.reshape(90,1)

train\_f2 = train\_f2.reshape(90,1)

w1 = np.zeros((90,1))

w2 = np.zeros((90,1))

epochs = 1

alpha = 0.0001

while(epochs < 10000):

y = w1 \* train\_f1 + w2 \* train\_f2

prod = y \* y\_train

print(epochs)

count = 0

for val in prod:

if(val >= 1):

cost = 0

w1 = w1 - alpha \* (2 \* 1/epochs \* w1)

w2 = w2 - alpha \* (2 \* 1/epochs \* w2)

else:

cost = 1 - val

w1 = w1 + alpha \* (train\_f1[count] \* y\_train[count] - 2 \* 1/epochs \* w1)

w2 = w2 + alpha \* (train\_f2[count] \* y\_train[count] - 2 \* 1/epochs \* w2)

count += 1

epochs += 1

from sklearn.metrics import accuracy\_score

**## Clip the weights**

index = list(range(10,90))

w1 = np.delete(w1,index)

w2 = np.delete(w2,index)

w1 = w1.reshape(10,1)

w2 = w2.reshape(10,1)

**## Extract the test data features**

test\_f1 = x\_test[:,0]

test\_f2 = x\_test[:,1]

test\_f1 = test\_f1.reshape(10,1)

test\_f2 = test\_f2.reshape(10,1)

**## Predict**

y\_pred = w1 \* test\_f1 + w2 \* test\_f2

predictions = []

for val in y\_pred:

if(val > 1):

predictions.append(1)

else:

predictions.append(-1)

print(accuracy\_score(y\_test,predictions))

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score

clf = SVC(kernel='linear')

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

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

print(accuracy\_score(y\_test,y\_pred))