Assignment3

Chaitra , Ganesh , Avik

May 16 2019

1 Question 1

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


from liblinearutil import *
from liblinear import *
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math


# In[2]:


from bokeh.plotting import figure, show, Figure
from bokeh.models import ColumnDataSource, Label
from bokeh.models.glyphs import Text
from bokeh.palettes import Spectral3
from bokeh.layouts import row, column, gridplot

# tell bokeh to show the figures in the notebook
from bokeh.io import output_notebook
output_notebook()


# In[3]:


dataFrame = pd.read_csv('./data/DWH_Training.csv')
```

```
dataFrameTest = pd.read_csv('./data/DWH_test.csv')


# In[4]:


dataFrameTest.head()


# In[5]:


dataFrame['height(cm)'].min()
dataFrame['weight(kg)'].min()


# In[6]:


palette = ["SpringGreen", "Crimson"]
dataFrame['color'] = np.where(dataFrame['gender']==1,"SpringGreen", "Crimson")
dataFrame['legend_values'] = np.where(dataFrame['gender']==1,"Male", "Female")


#aff a figure
p=figure(x_axis_label='height(cm)',y_axis_label='weight(kg)', width=800, height=
          title='Relation between Male and Female',
         )

#render the graph
p.circle('height(cm)','weight(kg)',source=dataFrame,size=5,alpha=0.8,color='colo
#annotate the graphic
p.y_range.start = dataFrame['weight(kg)'].min()
p.x_range.start = dataFrame['height(cm)'].min()
#show(p)


# In[7]:


import scipy

dataFrameTrain = pd.read_csv('./data/DWH_Training.csv')
dataFrameTrain.head()

dataFrameTest = pd.read_csv('./data/DWH_Test.csv')
```

```
dataFrameTest.head()


# In[8]:



file  = open('./data/DWH_Train.data','a+')
for index, row in dataFrameTrain.iterrows():
    label = row['gender']
    i=1
    featureValue = str(label) + ' '
    for element in row[1:len(row)-1]:
        featureValue +=(str(i)+':'+str(element)+' ')
        i+=1
    file.write(featureValue+'\n')

file.close()

file  = open('./data/DWH_Test.data','a+')
for index, row in dataFrameTest.iterrows():
    label = row['gender']
    i=1
    featureValue = str(label) + ' '
    for element in row[1:len(row)-2]:
        featureValue +=(str(i)+':'+str(element)+' ')
        i+=1
    file.write(featureValue+'\n')

file.close()


# In[9]:


y, x = svm_read_problem('./data/DWH_Train.data')
yt, xt = svm_read_problem('./data/DWH_Test.data',  return_scipy = True)
len(y)
m = train(y[:200], x[:200], '-s 2 -c 1 -B 9') #Explicitly set Bias
p_label, p_acc, p_val = predict(y[100:], x[100:], m) #Training Acc
p_label, p_acc, p_val = predict(yt[40:], xt[40:], m) #Testing Acc

save_model('DWH_Train.model',m)
weight = open('DWH_Train.model').readlines()
load_m = load_model('DWH_Train.model')
print(weight)
```

```
# In[10]:



w = load_m.get_decfun()[0]
bias=load_m.get_decfun()[1]

hyperPlaneValues = []
for index, row in dataFrame.iterrows():
    x = row['height(cm)']
    elements = []
    elements.append((-(w[0]/w[1])*x)-(bias/w[1]))
    hyperPlaneValues.append(elements)

normalizedList = []
heightWeightMap1 = {'height':[],
                    'weight':[]
                   }

for row in hyperPlaneValues:
    normalizedElement = []

    for term in row:
        if term < 0.0:
            term = 0.0
        normalizedElement.append(term)
    normalizedList.append(normalizedElement)

i = 0
for index, row in dataFrame.iterrows():
    x = row['height(cm)']

    if normalizedList[i][0]!=0:
        heightWeightMap1['height'].append(x)
        heightWeightMap1['weight'].append(normalizedList[i][0])

    i+=1

p.line(x= heightWeightMap1['height'], y=heightWeightMap1['weight'], line_width=2
show(p)


# In[ ]:
```

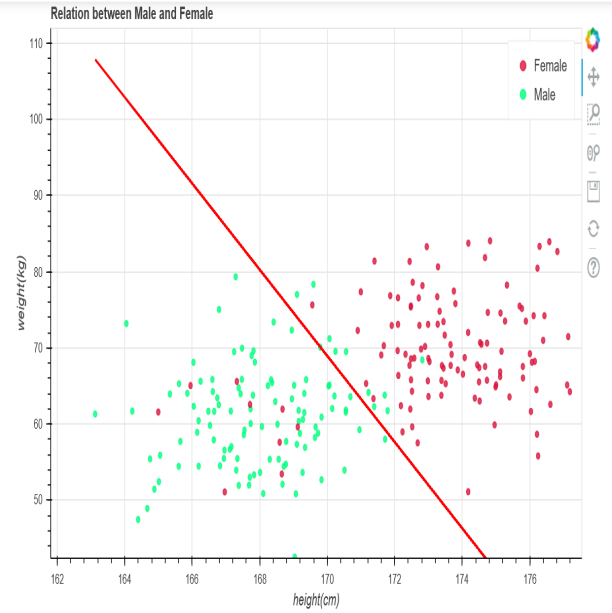

Figure 1: Hyperplane

```
Accuracy = 91.6038% (971/1060) (classification)
Accuracy = 88.6486% (164/185) (classification)
['solver_type L2R_L2LOSS_SVC\n', 'nr_class 2\n', 'label 1 -1\n', 'nr_feature 2\n', 'bias 9\n', 'w\n', '-0.1872976733720
\n', '-0.033099317173988688 \n', '3.7914488320344502 \n']
```

Figure 2: Training and Testing Accuracy

Sheet 1 Testing Accuracy: 57.777777 %
Sheet 2 Testing Accuracy: 75.777777 %
Sheet 3 Testing Accuracy: 88.6486 %

2 Question 2

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import pandas as pd


# In[2]:


dataFrameTrain = pd.read_csv('./data/INS_training.csv')
dataFrameTrain.head()


# In[3]:


dataFrameTest = pd.read_csv('./data/INS_test.csv')
dataFrameTest.head()


# In[4]:


file  = open('./data/featureFileTrain.csv','a+')
for index, row in dataFrameTrain.iterrows():
    label = row['target']
    labelNumber = label[len(label)-1]
    rowID = 'ex'+str(row['id'])
    i=1
    featureValue = str(labelNumber) + ' ' + str(rowID) + '|f '
    for element in row[1:len(row)-1]:
        if element != 0:
            featureValue +=(str(i)+':'+str(element)+' ')
        i+=1
    file.write(featureValue+'\n')
```

```
file.close()

file = open('./data/featureFileTest.csv','a+')
for index, row in dataFrameTest.iterrows():
    rowID = 'ex'+str(row['id'])
    featureValue = str(rowID) + '|f '
    i = 1
    for element in row[1:len(row)-1]:
        if element != 0:
            featureValue +=(str(i)+':'+str(element)+' ')
        i+=1
    file.write(featureValue+'\n')
file.close()
```

The Average Training Loss = 0.287637