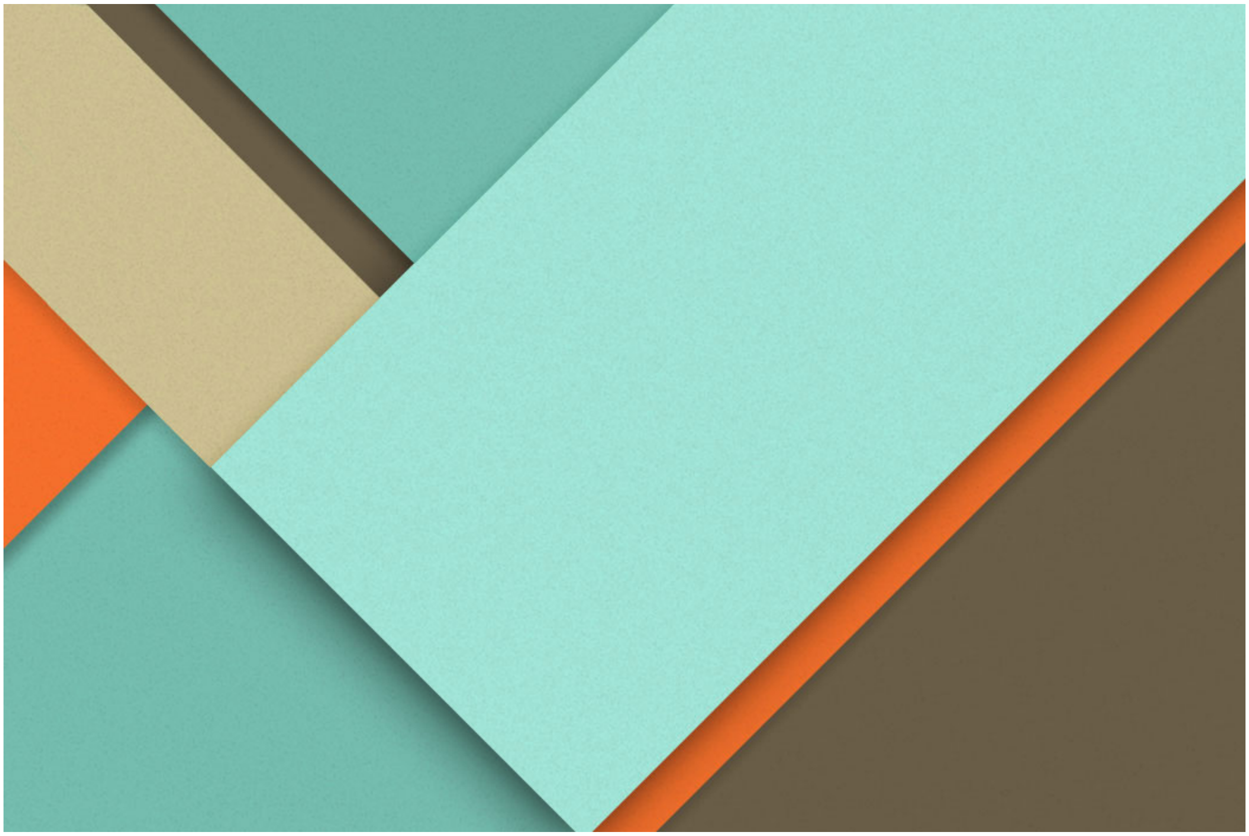


Lab 03 : Topics in Deep Learning

17th March 2019

Nanavati Tilak D.

201811001

Question 1 :

Use data in files "ex2data1-logistic.xls" and "ex2data2-logistic.xls" to perform logistic regression for each these data sets. Use 90% data points each set for training the regressor and remaining 10% for testing the accuracy of classification.

Goals

1. Finding the appropriate Theta Values
2. Finding and plotting Decision Boundaries.

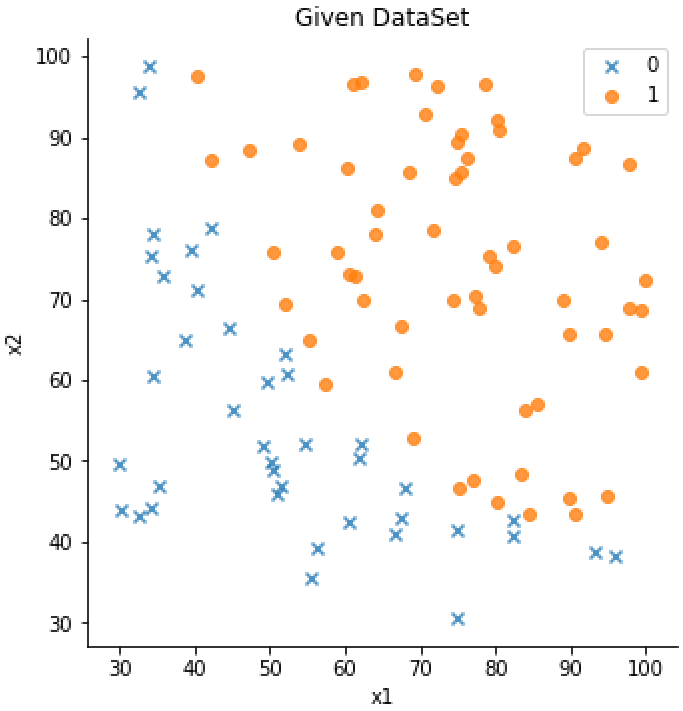
Code

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import math as m

dataset = pd.read_csv("ex2data1-logistic.csv")

sns.lmplot(x= "x1" , y= "x2" ,hue="y",data=dataset, fit_reg= False ,legend= False,markers=["x","o"])
plt.legend()
plt.title( "Given DataSet" )
plt.show()
```

Result



Data-points plotted given in the file.

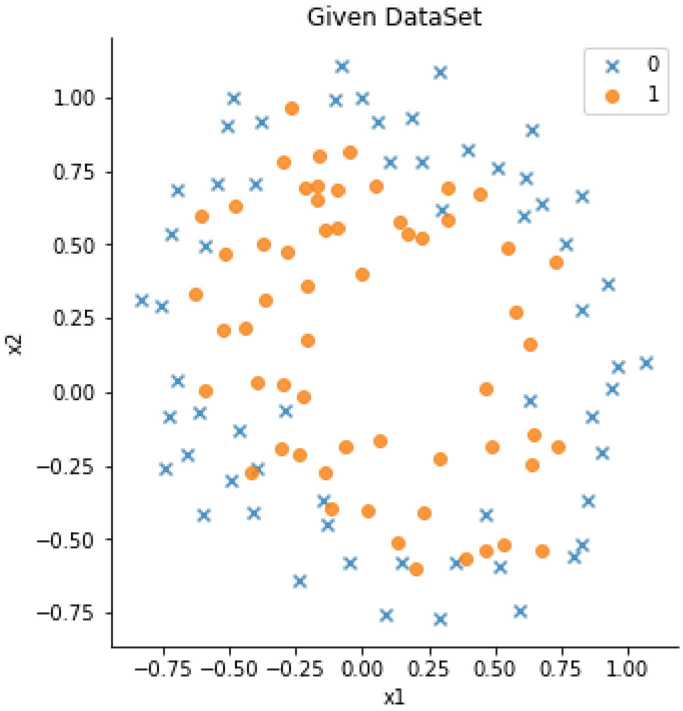
Code

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

dataset = pd.read_csv("ex2data2-logistic.csv")

sns.lmplot(x= "x1" , y= "x2" ,hue="y",data=dataset, fit_reg= False ,legend=
False,markers=["x","o"])
plt.legend()
plt.title( "Given DataSet" )
plt.show()
```

Result



Code

```
import pandas as pd

import math as m

import numpy as np


def gard_desc(m1,x,y,theta):


    for j in range(1500):

        alpha = 0.01

        A = []

        for i in range(m1):

            A.append(np.dot(x[i],theta))
        c = []
        for i in range(m1):
            c.append(1/(1+m.exp(-A[i])))

        T = []
        for i in range(m1):
            #print(c[i])
            if(c[i]!=1):
                T.append(y[i]*m.log(c[i]) + (1-y[i])*m.log(1-c[i]))
            else:
                T.append(0)

        B = [0,0,0]

        for i in range(m1):
            mul = (c[i]-y[i])
            x_m = pd.Series(x[i])
            B = (pd.Series(B) + (mul*x_m).tolist()).tolist()

        for i in range(len(theta)):
            theta[i] = theta[i] - (alpha*B[i])/m1

    return theta

dataset = pd.read_csv("ex2data1-logistic.csv")
theta = [0,0,0]

m1 = len(dataset["x1"])
```

```
x =[]
mn = [dataset["x1"].mean(),dataset["x1"].mean()]
std = [dataset["x1"].std(),dataset["x1"].std()]
for i in range(m1):
    l=[1]
    l.append((dataset.iloc[i,0]-mn[0])/std[0])
    l.append((dataset.iloc[i,1]-mn[1])/std[1])
    x.append(l)

#print(x)
y = list(dataset.iloc[:,2])
#print(y)
#print(np.dot(x[1],[1,1,1]))
theta_new = gard_desc(m1,x,y,theta)
print(theta_new)
```

#Theta Vallues : [0.5439106349462868, 1.5548382974288883, 1.4374899418749274]

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import math as m

dataset = pd.read_csv("ex2data1-logistic.csv")
y = dataset["y"]

mn = [dataset["x1"].mean(),dataset["x1"].mean()]
std = [dataset["x1"].std(),dataset["x1"].std()]

total = len(dataset["x1"])
```

```
for i in range(total):
    dataset.iloc[i,0] = (dataset.iloc[i,0]-mn[0])/std[0]
    dataset.iloc[i,1] = (dataset.iloc[i,1]-mn[1])/std[1]

ln = []
#print(dataset)
for i in range(total):
    c = theta[0]
    c = (c+theta[1]*dataset.iloc[i,0])/theta[2]
    ln.append(-c)
print(len(ln))
#print(ln)
sns.lmplot(x= "x1" , y= "x2" ,hue="y",data=dataset, fit_reg= False ,legend=
False,markers=["x","o"])
plt.legend()
plt.title("Given DataSet")
plt.plot(dataset["x1"],ln)
plt.show()
```

Result

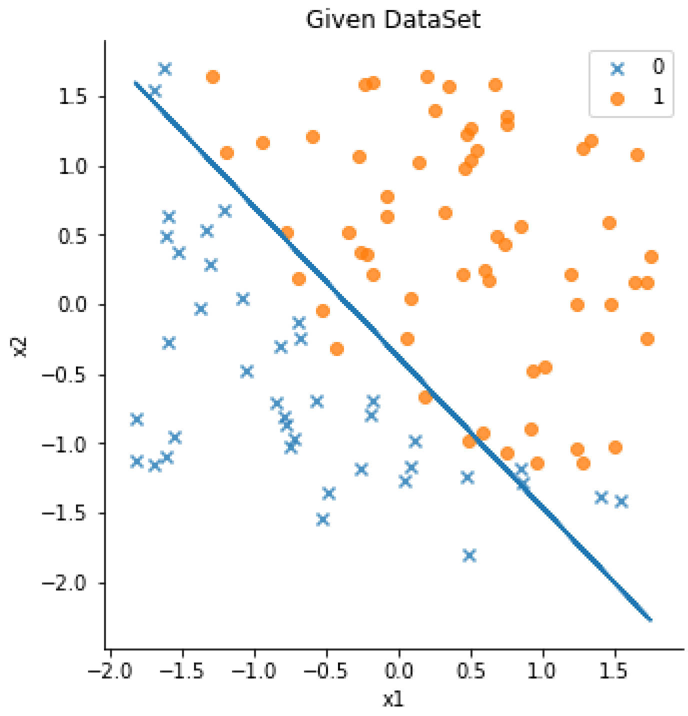


Fig. Decision Boundary for the data in File ex2data1-logistic.xls

Code

```
#for the NEXT Dataset , we are taking the function as:

#h_theta(x) = 1/(1+exp(-t0+t1*x1+t2*x2+t3*x1**2+t4*x2**2))


import pandas as pd
import math as m
import numpy as np

def gard_desc(m1,x,y,theta):

 for j in range(1000):
     alpha = 0.01

        A = []

        for i in range(m1):

            A.append(np.dot(x[i],theta))

                c = []

        for i in range(m1):

            c.append(1/(1+m.exp(-A[i])))
```

```
T = []
for i in range(m1):
    #print(c[i])
    if(c[i]!=1):
        T.append(y[i]*m.log(c[i]) + (1-y[i])*m.log(1-c[i]))
    else:
        T.append(0)
B = [0,0,0]
for i in range(m1):
    mul = (c[i]-y[i])
    x_m = pd.Series(x[i])
    B = (pd.Series(B) + (mul*x_m).tolist()).tolist()
for i in range(len(theta)):
    theta[i] = theta[i] - (alpha*B[i])/m1
```

```
  return theta

dataset = pd.read_csv("ex2data2-logistic.csv")

theta = [0,0,0]

#print(dataset)

m1 = len(dataset["x1"])

x =[]

mn = [dataset["x1"].mean(),dataset["x1"].mean()]

std = [dataset["x1"].std(),dataset["x1"].std()]

for i in range(m1):

    l=[1]

    l.append((dataset.iloc[i,0]**2))

    l.append((dataset.iloc[i,1]**2))

    x.append(l)
```

```
#print(x)

y = list(dataset.iloc[:,2])

#print(y)

#print(np.dot(x[1],[1,1,1]))

theta_new = gard_desc(m1,x,y,theta)

print(theta_new)
```

#Theta Values : [0.0979458932840007, -0.433528173448014 64, -0.3031778923057623]

```
#Now forming the the Decision Boundary...
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import math as m

dataset = pd.read_csv("ex2data2-logistic.csv")
y = dataset["y"]

total = len(dataset["x1"])

"""for i in range(total):
    dataset.iloc[i,0] = (dataset.iloc[i,0]-mn[0])/std[0]
    dataset.iloc[i,1] = (dataset.iloc[i,1]-mn[1])/std[1]"""

ln1 = []
ln2 = []
xd1 = np.arange(-0.48,0.48,0.001)
#print(dataset)
for i in range(len(xd1)):
    c = (-theta[0]-theta[1]*xd1[i]**2)/theta[2]

    #c = (c-theta[3]*dataset.iloc[i,0]**2)/theta[4]
    #print(c)
    if c>=0:
        ln1.append(m.sqrt(c))
        ln2.append(-m.sqrt(c))
    else:
        ln1.append(-m.sqrt(-c))
        ln2.append(m.sqrt(-c))

print(len(ln))
#print(ln)
sns.lmplot(x= "x1",y= "x2" ,hue="y",data=dataset, fit_reg= False ,legend= False,markers=["x","o"])
plt.legend()
plt.title("Given DataSet")
#plt.plot(dataset["x1"],ln)

plt.plot(xd1,ln1,'b')
plt.plot(xd1,ln2,'b')
```

Result

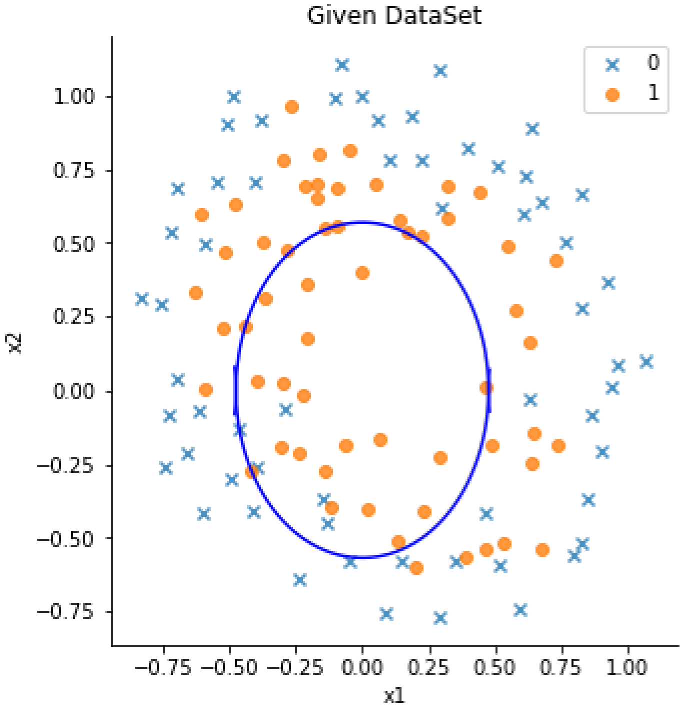


Fig. Decision Boundary for the data in File ex2data2-logistic.xls

Question 2 :

For testing the convexity / non-convexity of the cost function, consider one example from the first dataset. Now plot the cost function by varying the values of parameters (\theta) for logistic regression cost. Note that the hypothesis to be used is the sigmoid function in both the cases.

Code

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d, Axes3D

theta1 = np.arange(-10,10,2)
theta2 = np.arange(-10,10,2)
theta3 = np.arange(-10,10,2)
J = []

m1 = len(dataset["x1"])
x =[]

mn = [dataset["x1"].mean(),dataset["x1"].mean()]
std = [dataset["x1"].std(),dataset["x1"].std()]
for i in range(m1):
    l=[1]
    l.append((dataset.iloc[i,0]-mn[0])/std[0])
    l.append((dataset.iloc[i,1]-mn[1])/std[1])
    x.append(l)

J = []
for i in theta1:
    for j in theta2:
        J1 = []
        for k in theta3:
            A = []
            for i in range(m1):

                theta_c = []
                theta_c.append(i)
                theta_c.append(j)
                theta_c.append(k)

                A.append(np.dot(x[i],theta_c))

            c = []
            for i in range(m1):
                c.append(1/(1+m.exp(-A[i])))

            T = []
            for i in range(m1):
                #print(c[i])
                if(c[i]!=1):
                    T.append(y[i]*m.log(c[i]) + (1-y[i])*m.log(1-c[i]))
                else:
                    T.append(0)
        #print((-1/m1)*sum(T))
        J1.append((-1/m1)*sum(T))
    J.append(J1)

#print(J)
X, Y = np.meshgrid(theta2,theta3)
Z = np.asarray(J[0:10])
#print(Z)
fig = plt.figure()
ax = Axes3D(fig)
ax.plot_surface(X, Y, Z, rstride=1, cstride=1,cmap='viridis', edgecolor='green')
ax.set_xlabel('Theta2')
ax.set_ylabel('Theta3')
ax.set_zlabel('Cost Value J(Theta2,Theta3)');
```

Result

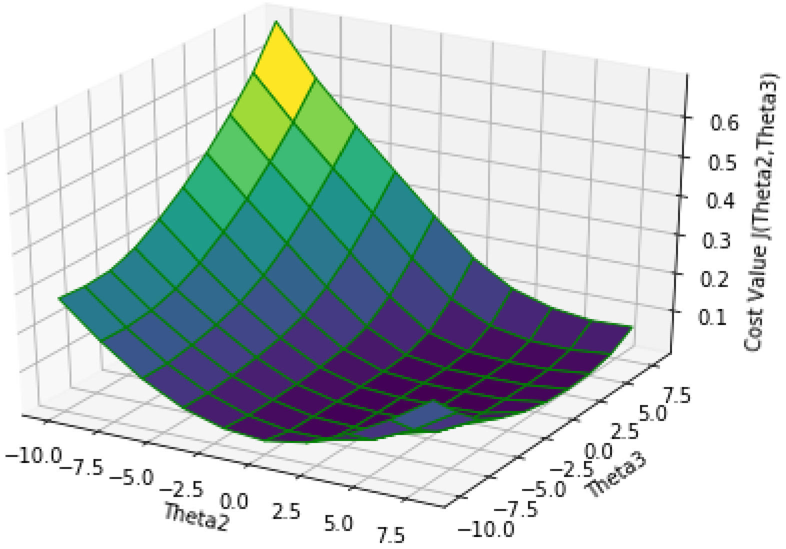


Fig. Cost Function Values keeping **Theta 1** constant and **Theta2** and **Theta3** variable