regression-using-gredient-descent

November 10, 2024

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

[3]:
```
df = pd.read_csv('diabetes.csv')
```

[5]:
```
x = df['Glucose'].values
```

[6]:
```
x
```

[6]:
```
array([148,  85, 183,  89, 137, 116,  78, 115, 197, 125, 110, 168, 139,
       189, 166, 100, 118, 107, 103, 115, 126,  99, 196, 119, 143, 125,
       147,  97, 145, 117, 109, 158,  88,  92, 122, 103, 138, 102,  90,
       111, 180, 133, 106, 171, 159, 180, 146,  71, 103, 105, 103, 101,
        88, 176, 150,  73, 187, 100, 146, 105,  84, 133,  44, 141, 114,
        99, 109, 109,  95, 146, 100, 139, 126, 129,  79,   0,  62,  95,
       131, 112, 113,  74,  83, 101, 137, 110, 106, 100, 136, 107,  80,
       123,  81, 134, 142, 144,  92,  71,  93, 122, 163, 151, 125,  81,
        85, 126,  96, 144,  83,  95, 171, 155,  89,  76, 160, 146, 124,
        78,  97,  99, 162, 111, 107, 132, 113,  88, 120, 118, 117, 105,
       173, 122, 170,  84,  96, 125, 100,  93, 129, 105, 128, 106, 108,
       108, 154, 102,  57, 106, 147,  90, 136, 114, 156, 153, 188, 152,
        99, 109,  88, 163, 151, 102, 114, 100, 131, 104, 148, 120, 110,
       111, 102, 134,  87,  79,  75, 179,  85, 129, 143, 130,  87, 119,
         0,  73, 141, 194, 181, 128, 109, 139, 111, 123, 159, 135,  85,
       158, 105, 107, 109, 148, 113, 138, 108,  99, 103, 111, 196, 162,
        96, 184,  81, 147, 179, 140, 112, 151, 109, 125,  85, 112, 177,
       158, 119, 142, 100,  87, 101, 162, 197, 117, 142, 134,  79, 122,
        74, 171, 181, 179, 164, 104,  91,  91, 139, 119, 146, 184, 122,
       165, 124, 111, 106, 129,  90,  86,  92, 113, 111, 114, 193, 155,
       191, 141,  95, 142, 123,  96, 138, 128, 102, 146, 101, 108, 122,
        71, 106, 100, 106, 104, 114, 108, 146, 129, 133, 161, 108, 136,
       155, 119,  96, 108,  78, 107, 128, 128, 161, 151, 146, 126, 100,
       112, 167, 144,  77, 115, 150, 120, 161, 137, 128, 124,  80, 106,
       155, 113, 109, 112,  99, 182, 115, 194, 129, 112, 124, 152, 112,
       157, 122, 179, 102, 105, 118,  87, 180, 106,  95, 165, 117, 115,
       152, 178, 130,  95,   0, 122,  95, 126, 139, 116,  99,   0,  92,
```

```
137,  61,  90,  90, 165, 125, 129,  88, 196, 189, 158, 103, 146,
147,  99, 124, 101,  81, 133, 173, 118,  84, 105, 122, 140,  98,
 87, 156,  93, 107, 105, 109,  90, 125, 119, 116, 105, 144, 100,
100, 166, 131, 116, 158, 127,  96, 131,  82, 193,  95, 137, 136,
 72, 168, 123, 115, 101, 197, 172, 102, 112, 143, 143, 138, 173,
 97, 144,  83, 129, 119,  94, 102, 115, 151, 184,  94, 181, 135,
 95,  99,  89,  80, 139,  90, 141, 140, 147,  97, 107, 189,  83,
117, 108, 117, 180, 100,  95, 104, 120,  82, 134,  91, 119, 100,
175, 135,  86, 148, 134, 120,  71,  74,  88, 115, 124,  74,  97,
120, 154, 144, 137, 119, 136, 114, 137, 105, 114, 126, 132, 158,
123,  85,  84, 145, 135, 139, 173,  99, 194,  83,  89,  99, 125,
 80, 166, 110,  81, 195, 154, 117,  84,   0,  94,  96,  75, 180,
130,  84, 120,  84, 139,  91,  91,  99, 163, 145, 125,  76, 129,
 68, 124, 114, 130, 125,  87,  97, 116, 117, 111, 122, 107,  86,
 91,  77, 132, 105,  57, 127, 129, 100, 128,  90,  84,  88, 186,
187, 131, 164, 189, 116,  84, 114,  88,  84, 124,  97, 110, 103,
 85, 125, 198,  87,  99,  91,  95,  99,  92, 154, 121,  78, 130,
111,  98, 143, 119, 108, 118, 133, 197, 151, 109, 121, 100, 124,
 93, 143, 103, 176,  73, 111, 112, 132,  82, 123, 188,  67,  89,
173, 109, 108,  96, 124, 150, 183, 124, 181,  92, 152, 111, 106,
174, 168, 105, 138, 106, 117,  68, 112, 119, 112,  92, 183,  94,
108,  90, 125, 132, 128,  94, 114, 102, 111, 128,  92, 104, 104,
 94,  97, 100, 102, 128, 147,  90, 103, 157, 167, 179, 136, 107,
 91, 117, 123, 120, 106, 155, 101, 120, 127,  80, 162, 199, 167,
145, 115, 112, 145, 111,  98, 154, 165,  99,  68, 123,  91, 195,
156,  93, 121, 101,  56, 162,  95, 125, 136, 129, 130, 107, 140,
144, 107, 158, 121, 129,  90, 142, 169,  99, 127, 118, 122, 125,
168, 129, 110,  80, 115, 127, 164,  93, 158, 126, 129, 134, 102,
187, 173,  94, 108,  97,  83, 114, 149, 117, 111, 112, 116, 141,
175,  92, 130, 120, 174, 106, 105,  95, 126,  65,  99, 102, 120,
102, 109, 140, 153, 100, 147,  81, 187, 162, 136, 121, 108, 181,
154, 128, 137, 123, 106, 190,  88, 170,  89, 101, 122, 121, 126,
 93], dtype=int64)
```

[7]:
```
y = df['Outcome']
```

[8]:
```
y = np.array(y)
y
```

[8]:
```
array([1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0,
       1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1,
```

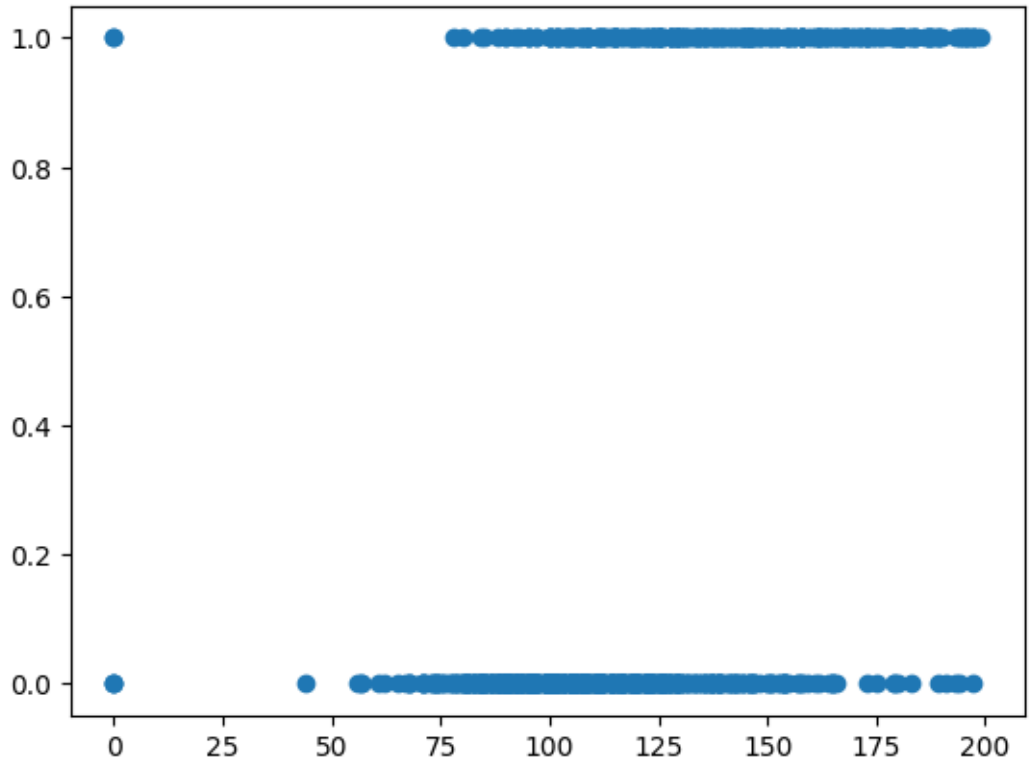
```
       0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0,
       1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0,
       1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0,
       0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0,
       1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0,
       0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1,
       0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0,
       1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0],
      dtype=int64)
```

[9]:
```
print(y)
```

```
[1 0 1 0 1 0 1 0 1 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 1 0 0 0 0 0
 1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 1 0 1 0
 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1
 1 0 0 1 1 1 0 0 0 1 0 0 0 0 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 0 1 0 1 0 0 0 0 0
 1 1 1 1 1 0 0 1 1 0 1 0 1 1 1 0 0 0 0 0 0 1 1 0 1 0 0 0 1 1 1 1 0 1 1 1 1
 0 0 0 0 0 1 0 0 1 1 0 0 0 1 1 1 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0
 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 0 1 0 0 0 1 1 1 0 0
 1 0 1 0 1 1 0 1 0 0 1 0 1 1 0 0 1 0 1 0 0 1 0 1 0 1 1 1 0 0 1 0 1 0 0 0 1
 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0 1 1 0 0 1 0 0 1 0 0 1
 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0 1 0 1 1 0 1 0 1 0 1
 0 1 1 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1
 1 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1
 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0
 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 0 1 0 1 0 1 0
```

```
1 0 0 1 0 0 1 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1 1 0
1 1 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 1 1 0 0 0 0 1 1
0 0 0 1 0 1 1 0 0 1 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1
1 0 0 1 0 0 1 0 1 1 1 0 0 1 1 1 0 1 0 1 0 1 0 0 0 0 0 1 0]
```

[10]:
```
plt.scatter(x,y)
plt.show()
```



[11]:
```
# y = b0 + b1*x
def sigmoid(y):

    return (1 / (1+ np.exp(-y)))
```

[12]:
```
def p(x,b0,b1):
    return ( sigmoid(b0 + b1*x) )
```

[13]:
```
def cost(y,x,b0,b1):
    n = len(x)
    c = 0
    for i in range(n):
```

```
        c += y[i]*np.log(p(x[i],b0,b1) + 0.00001) + (1-y[i]) * np.log(1 -␣
↪p(x[i],b0,b1)+0.0001)
    return -1*(c/n)
```

[14]:
```
def GD(x,y,epoch):
    lemda = 0.001
    b0 = 3.58
    b1 = 2.72
    counter = 1
    c =  cost(y,x,b0,b1)
  #  c = cost(y,x,b0,b1)
    while(True):
        c =  cost(y,x,b0,b1)
        b0_cost = 0
        b1_cost = 0
        for i in range(len(x)):
            b1_cost = b1_cost + (-y[i] + p(x[i],b0,b1)) * x[i]
            b0_cost = b0_cost +  (-y[i] + p(x[i],b0,b1))
        b1 = b1 - (1/len(x)) * lemda * b1_cost
        b0 = b0 - lemda * (1/len(x)) *b0_cost
        c1 = cost(y,x,b0,b1)

        print(cost(y,x,b0,b1))

        if(abs(c1-c) < 0.0000001):
            return (b0,b1)
```

[16]:
```
b0,b1 = GD(x,y,60)
```

```
5.974480542310062
5.9744781239728555
5.9744757057030355
5.9744732875006505
5.974470869365749
5.9744684512983826
5.974466033298596
5.974463615366442
5.974461197501969
5.974458779705223
5.974456361976256
5.974453944315116
5.974451526721854
5.974449109196517
5.9744466917391525
5.974444274349813
5.974441857028547
5.974439439775403
```

5.974437022590429
5.974434605473678
5.974432188425195
5.974429771445032
5.974427354533238
5.974424937689862
5.974422520914953
5.974420104208561
5.974417687570734
5.974415271001515
5.974412854500794
5.9744104380647824
5.97440802160376
5.974405602999884
5.974403131502967
5.974399337168357
5.974358225533185
5.972825439750338
5.816859347887854
2.2823222963404777
2.1953639641963396
1.1270804271221926
1.4643851979009028
1.4972976500385566
2.1265249847450716
1.1634911789100821
1.5680369369622207
1.4702916667532708
2.1012486930324057
1.1775290508822962
1.6052133304153442
1.4560240433702187
2.0868182053684623
1.185670685127808
1.6261183578613287
1.4471455817325314
2.077455914513933
1.1909915715645798
1.6395405792005466
1.4411479684373838
2.0709681076111255
1.1946904153364906
1.6487725442457304
1.4368930211892446
2.066289748342978
1.1973588916576987
1.6553931581633916
1.4337766334278756

2.0628277129416692
1.1993303291629005
1.6602722844436293
1.4314439496747484
2.0602207473878447
1.2008094781149898
1.6639348361254118
1.4296711294822233
2.0582344589628616
1.2019300470185639
1.666718706759893
1.4283093474499864
2.056709480425025
1.2027834036842555
1.6688520992909852
1.427255607193399
2.0555335221486004
1.2034341807611466
1.670494899530942
1.4264363736343497
2.054625244358883
1.2039293440979975
1.6717623795579526
1.425797842486081
2.0539244307996873
1.2043037504163518
1.6727395423844273
1.4252998724266812
2.05338572855987
1.2045837057765671
1.6734902032282648
1.4249120470545777
2.052974508051471
1.2047893290653242
1.6740629602119481
1.4246110309874236
2.0526640370410565
1.204936171344743
1.674495248126956
1.424378742592632
2.052433498828781
1.2050363551657708
1.6748161674573456
1.4242010592625334
2.052266569930523
1.2050993945234607
1.675048505844804
1.424066880187892

2.0521503789555617
1.2051327965018466
1.6752102120481593
1.4239674353705116
2.0520747316033154
1.2051425100558293
1.6753154894307778
1.42389576816227
2.0520315254934047
1.2051332654583804
1.6753756191911615
1.4238463425950882
2.0520143030124487
1.2051088340687586
1.6753995878637697
1.423814742075363
2.052017906192735
1.2050722290737
1.6753945706234779
1.4237974360285568
2.052038208122679
1.2050258618737408
1.6753663067626594
1.4237915977738496
2.0520719024839953
1.2049716647290378
1.6753193934904083
1.4237949614789231
2.0521163377108143
1.20491118746931
1.6752575171737727
1.423805709225822
2.052169385713638
1.2048456740907707
1.6751836362138735
1.4238223814740516
2.052229337581416
1.2047761236400985
1.6751001262338507
1.4238438058321912
2.0522948204723446
1.2047033387475843
1.6750088957053346
1.4238690402402268
2.052364731232916
1.2046279644031848
1.674911478261692
1.4238973275492715

2.0524381832795373
1.2045505189928618
1.6748091065439488
1.4239280591504107
2.052514464030997
1.2044714191750037
1.6747027713652267
1.423960745811222
2.05259300075758
1.2043909998411964
1.6745932691695729
1.423994994267815
2.052673333159389
1.2043095301457827
1.6744812401357425
1.42403048842195
2.0527550913337125
1.204227226386518
1.6743671987915265
1.4240669742286407
2.0528379780641433
1.2041442623592618
1.67425155862261
1.4241042475450614
2.052921754579317
1.2040607776848093
1.6741346518613478
1.4241421443576698
2.053006229099043
1.2039768845060217
1.6740167454029302
1.4241805329209012
2.053091247621459
1.2038926728749524
1.6738980536087509
1.4242193074329015
2.053176686512229
1.2038082150862508
1.6737787486060454
1.4242583829477355
2.0532624465435925
1.203723569162837
1.6736589685736252
1.4242976912825867
2.0533484481000017
1.2036387816594882
1.6735388244067588
1.4243371777255174

2.0534346273222983
1.203553889917529
1.6734184050779264
1.4243767983876792
2.0535209330073507
1.2034689238778338
1.6732977819480108
1.4244165180739843
2.0536073241153905
1.2033839075384558
1.6731770122328191
1.4244563085711361
2.053693767766372
1.203298860126174
1.673056141789789
1.424496147271504
2.053780237629827
1.2032137970381827
1.6729352073581698
1.4245360160670157
2.0538667126310575
1.2031287305985046
1.6728142383588427
1.424575900460549
2.0539531759118774
1.2030436706657734
1.6726932583406084
1.4246157888519397
2.054039613995965
1.2029586251212272
1.6725722861417232
1.4246556719646721
2.054126016118733
1.2028736002605944
1.6724513368227152
1.4246955423854788
2.054212373689409
1.2027886011086195
1.6723304224153033
1.424735394194782
2.054298679859349
1.20270363167144
1.6722095525234064
1.424775222670094
2.0543849291756504
1.2026186951391673
1.6720887348054856
1.4248150240481132

2.054471117303448
1.2025337940482363
1.6719679753614842
1.4248547953337865
2.0545572408030526
1.2024489304117092
1.671847279043125
1.4248945341473045
2.0546432969514075
1.2023641058236885
1.671726649702771
1.4249342386013604
2.0547292835989883
1.2022793215429373
1.6716060903927534
1.424973907202851
2.054815199055094
1.2021945785600028
1.6714856035253352
1.4250135387740093
2.054901041995992
1.2021098776508796
1.671365191000662
1.4250531323893334
2.054986811391337
1.2020252194200394
1.6712448543093774
1.4250926873250236
2.055072506445318
1.2019406043348508
1.6711245946147326
1.4251322030184574
2.055158126549492
1.2018560327531367
1.6710044128183004
1.4251716790358833
2.055243671245017
1.201771504945251
1.6708843096126296
1.4252111150464364
2.0553291401925944
1.2016870211116957
1.67076428552332
1.425250510801398
2.0554145331482
1.2016025813972286
1.6706443409426808
1.4252898661176385

2.055499849943903
1.2015181859021584
1.6705244761566718
1.4253291808643471
2.0555850904723765
1.201433834691331
1.6704046913664239
1.4253684549524215
2.0556702546745953
1.2013495278013882
1.6702849867054652
1.4254076883260138
2.0557553425300146
1.2012652652464466
1.6701653622533694
1.4254468809557472
2.0558403540486587
1.2011810470227544
1.6700458180468774
1.4254860328333459
2.055925289264881
1.2010968731123384
1.66992635408857
1.4255251439672596
2.056010148232352
1.2010127434859394
1.6698069703538885
1.425564214379283
2.0560949310200356
1.2009286581053213
1.669687666796712
1.4256032441017565
2.056179637709022
1.2008446169251976
1.669568443353854
1.425642233175397
2.056264268389966
1.2007606198946539
1.6694492999485906
1.4256811816475683
2.0563488231610987
1.200676666958362
1.6693302364934561
1.4257200895708493
2.056433302126565
1.2005927580575253
1.6692112528925627
1.4257589570019242

2.0565177053951316
1.2005088931306325
1.6690923490434135
1.4257977840007434
2.056602033079213
1.2004250721140395
1.6689735248382573
1.425836570629765
2.056686285294021
1.2003412949424679
1.6688547801652935
1.4258753169534202
2.056770462156916
1.2002575615493865
1.6687361149095812
1.4259140230376504
2.056854563786874
1.2001738718672958
1.668617528953755
1.425952688949586
2.056938590304131
1.2000902258279609
1.6684990221785345
1.4259913147572192
2.0570225418297747
1.2000066233626543
1.6683805944633228
1.4260299005291952
2.0571064184855588
1.1999230644022354
1.6682622456864742
1.4260684463346431
2.057190220393665
1.1998395488773352
1.6681439757255563
1.4261069522430414
2.05727394767653
1.1997560767184041
1.6680257844576445
1.4261454183240672
2.0573576004567173
1.1996726478558224
1.6679076717595336
1.4261838446475419
2.0574411788568487
1.1995892622199367
1.667789637507756
1.4262222312833643

2.057524682999491
1.1995059197411222
1.6676716815788122
1.4262605783014155
2.0576081130070905
1.199422620349815
1.6675538038492217
1.4262988857715486
2.0576914690019175
1.1993393639765357
1.6674360041955898
1.4263371537635487
2.057774751106073
1.1992561505519277
1.6673182824946926
1.4263753823470937
2.0578579594414146
1.1991729800067592
1.667200638623444
1.4264135715917383
2.057941094129501
1.1990898522719482
1.6670830724590342
1.4264517215669053
2.05802415529168
1.199006767278566
1.666965583878896
1.426489832341857
2.0581071430489324
1.1989237249578588
1.6668481727607196
1.426527903985699
2.058190057522
1.1988407252412425
1.6667308389825186
1.4265659365673582
2.0582728988312566
1.1987577680603136
1.6666135824226054
1.4266039301555908
2.05835566709681
1.1986748533468436
1.6664964029595797
1.4266418848189728
2.0584383624384053
1.198591981032784
1.6663793004723653
1.426679800625888

2.0585209849754507
1.1985091510503
1.6662622748402842
1.4267176776445274
2.0586035348270655
1.198426363331713
1.6661453259428969
1.4267555159429097
2.058686012112011
1.198343617809564
1.666028453660161
1.4267933155888401
2.058768416948681
1.198260914416567
1.6659116578723836
1.4268310766499315
2.0588507494551824
1.1981782530856397
1.665794938460157
1.4268687991936293
2.058933009749286
1.1980956337498745
1.6656782953044535
1.4269064832871676
2.059015197948391
1.1980130563425642
1.6655617282865547
1.4269441289975842
2.0590973141695987
1.1979305207971869
1.6654452372881139
1.4269817363917305
2.0591793585296583
1.1978480270474299
1.6653288221911255
1.427019305536254
2.059261331144985
1.1977655750271414
1.66521248287788
1.4270568364976315
2.059343232131676
1.19768316467039
1.665096219231077
1.4270943293421128
2.0594250616054537
1.197600795911417
1.6649800311337213
1.4271317841357787

2.0595068196817756
1.1975184686846483
1.6648639184691267
1.427169200944535
2.059588506475722
1.1974361829247022
1.6647478811209628
1.4272065798340614
2.059670122102078
1.197353938566385
1.6646319189732373
1.4272439208698684
2.0597516666752522
1.1972717355446951
1.6645160319103198
1.4272812241172776
2.0598331403093995
1.1971895737948002
1.6644002198168382
1.427318489641418
2.059914543118289
1.1971074532520734
1.6642844825778467
1.4273557175072205
2.059995875215374
1.1970253738520695
1.66416882007868
1.4273929077794343
2.0600771367138146
1.1969433355305121
1.664053232204995
1.4274300605226469
2.0601583277264317
1.1968613382233235
1.6639377188427726
1.4274671758012305
2.0602394483657496
1.196779381866584
1.6638222798783506
1.4275042536793745
2.0603204987439314
1.1966974663965915
1.6637069151983492
1.4275412942210968
2.0604014789728584
1.196615591749794
1.6635916246897675
1.4275782974902276

2.060482389164092
1.1965337578628301
1.6634764082398785
1.4276152635504094
2.0605632294288503
1.1964519646725298
1.6633612657363068
1.427652192465106
2.0606439998780863
1.196370212115881
1.6632461970669736
1.4276890842976224
2.060724700622423
1.196288500130051
1.6631312021201061
1.4277259391110428
2.060805331772146
1.1962068286524106
1.6630162807843198
1.4277627569682878
2.060885893437255
1.1961251976204759
1.6629014329484682
1.4277995379321011
2.060966385727444
1.196043606971963
1.6627866585017614
1.4278362820650659
2.061046808752109
1.1959620566447389
1.662671957333704
1.427872989429544
2.0611271626202843
1.195880546576871
1.662557329334148
1.4279096600877568
2.0612074474407804
1.1957990767065714
1.662442774393173
1.4279462941017538
2.061287663322059
1.19571764697225
1.6623282924012714
1.4279828915333788
2.0613678103722752
1.1956362573124844
1.662213883249178
1.428019452444307

2.061447888699294
1.1955549076660155
1.6620995468279727
1.4280559768960772
2.0615278984107026
1.1954735979717552
1.6619852830290032
1.4280924649500013
2.061607839613744
1.1953923281687937
1.6618710917439394
1.4281289166672577
2.0616877124154156
1.1953110981963786
1.661756972864756
1.4281653321088437
2.0617675169223855
1.1952299079939364
1.6616429262837238
1.4282017113355685
2.0618472532410252
1.195148757501072
1.661528951893451
1.4282380544080864
2.061926921477422
1.1950676466575254
1.661415049586784
1.4282743613868785
2.0620065217373735
1.1949865754032454
1.6613012192569283
1.4283106323322474
2.0620860541263766
1.1949055436783091
1.6611874607973378
1.4283468673043531
2.0621655187496657
1.1948245514229738
1.6610737741017723
1.4283830663631747
2.062244915712163
1.1947435985776655
1.6609601590643086
1.4284192295685019
2.062324245118476
1.194662685082974
1.660846615579316
1.4284553569799758

2.062403507072974
1.1945818108796586
1.6607331435414485
1.4284914486570794
2.0624827016797207
1.1945009759086196
1.6606197428456406
1.428527504659127
2.062561829042496
1.1944201801109349
1.6605064133871068
1.428563525045262
2.0626408892648054
1.1943394234278455
1.6603931550613977
1.428599509874462
2.0627198824498403
1.1942587058007479
1.660279967764315
1.4286354592055492
2.0627988087005558
1.1941780271712068
1.6601668513919572
1.4286713730971765
2.0628776681196013
1.1940973874809317
1.660053805840703
1.4287072516078434
2.0629564608093363
1.1940167866718097
1.6599408310072346
1.4287430947958686
2.063035186871865
1.193936224685885
1.6598279267885172
1.4287789027194224
2.0631138464089975
1.1938557014653373
1.6597150930817632
1.4288146754365274
2.0631924395222914
1.193775216952517
1.6596023297844944
1.4288504130050497
2.0632709663130195
1.193694771089944
1.6594896367944967
1.4288861154826498

2.063349426882166
1.193614363820277
1.6593770140098727
1.4289217829268797
2.0634278213304444
1.1935339950863377
1.659264461328969
1.428957415395118
2.063506149758331
1.1934536648311027
1.6591519786504116
1.4289930129445763
2.0635844122659965
1.1933733729976927
1.659039565873093
1.4290285756323302
2.0636626089533436
1.1932931195293954
1.6589272228962164
1.429064103515275
2.063740739920025
1.1932129043696549
1.6588149496192486
1.4290995966501585
2.0638188052653965
1.193132727462055
1.6587027459419044
1.4291350550936055
2.0638968050886057
1.193052588750323
1.6585906117641682
1.4291704789020407
2.0639747394884815
1.1929724881783643
1.6584785469863295
1.4292058681317477
2.0640526085635904
1.1928924256902143
1.6583665515089308
1.4292412228388855
2.0641304124122546
1.1928124012300696
1.6582546252327741
1.4292765430794159
2.064208151132535
1.192732414742272
1.6581427680589498
1.4293118289091948

```
2.064285824822226
1.1926524661713103
1.6580309798887987
1.429347080383885
2.064363433578856
1.192572555461824
1.657919260623905
1.429382297559035
2.064440977499706
1.1924926825586022
1.6578076101661756
1.429417480490015
2.0645184566817747
1.1924128474065798
1.6576960284177413
1.4294526292320555
2.064595871221827
1.1923330499508413
1.657584515280991
1.4294877438402571
2.0646732212163963
1.1922532901365996
1.6574730706585907
1.429522824369542
2.064750506761674
1.1921735679092451
1.6573616944534664
1.429557870874701
2.0648277279537046
1.192093883214287
1.6572503865687926
1.4295928834103797
2.064904884888194
1.192014235997387
1.6571391469080208
1.4296278620310587
2.0649819776606306
1.191934626204362
1.6570279753748416
1.4296628067911075
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Input In [16], in <cell line: 1>()
----> 1 b0,b1 = GD(x,y,1)

Input In [14], in GD(x, y, epoch)
```

```
     15 b1 = b1 - (1/len(x)) * lemda * b1_cost
     16 b0 = b0 - lemda * (1/len(x)) *b0_cost
---> 17 c1 = cost(y,x,b0,b1)
     19 print(cost(y,x,b0,b1))
     21 if(abs(c1-c) < 0.0000001):

Input In [13], in cost(y, x, b0, b1)
      3 c = 0
      4 for i in range(n):
----> 5     c += y[i]*np.log(p(x[i],b0,b1) + 0.00001) + (1-y[i]) * np.log(1 -␣
 ↪p(x[i],b0,b1)+0.0001)
      6 return -1*(c/n)

KeyboardInterrupt:
```

[ ]:
```
b0,b1 = GD(x,y,1)
```

[ ]:
```
def y_pred(b0,b1,x):
    l = []
    for i in range(len(x)):
        if(p(x[i],b0,b1) >= 0.5):
            l.append(1)
        else:
            l.append(0)
    return (np.array(l))
```

[ ]:
```
yPred = y_pred(b0,b1,x)
```

[ ]:
```
yPred
```