

```

1 # logRegTrainer.py
2 # Logistic Regression Trainer
3 import numpy
4 class logRegTrainer(object):
5     def __init__(self):
6         # Define step length and iteration numbers
7         self.eta = 0.0025
8         self.iteration = 10000
9     def sigmoid(self, x):
10        return 1/(1 + numpy.exp(-x))
11
12    def loadTrainDataSet(self, inputData):
13        # Train the model
14        self.theta = numpy.zeros((inputData.inputNum+1))
15        self.devLL = numpy.zeros((inputData.inputNum+1))
16        for i in range(0, self.iteration):
17            self.devLL = numpy.zeros((inputData.inputNum + 1))
18            if i%100 == 0:
19                print("Iteration #%d"%(i))
20
21            for k in range(0, inputData.rowNum):
22                # Calculate devLL
23                x_now = numpy.append(numpy.array([1]), inputData.rowInputList[k])
24                self.devLL += (inputData.rowOutputList[k] - self.sigmoid(numpy.sum(x_now*
self.theta)))*x_now
25                # Move one step
26                self.theta += self.eta*self.devLL
27
28
29
30    def loadTestDataSet(self, testData):
31        # Verify model with test data set
32        self.py = numpy.zeros(testData.rowNum)
33        self.ty = numpy.zeros(testData.rowNum)
34        self.correct = 0
35        self.correctRate = float(0)
36        for i in range(0, len(testData.rowInputList)):
37            x_now = numpy.append(numpy.array([1]), testData.rowInputList[i])
38            # Calculate P(Y = 1|X) based on current row of data
39            self.py[i] = self.sigmoid(numpy.sum(x_now* self.theta))
40
41            if self.py[i] > 0.5:
42                self.ty[i] = 1
43            else:
44                self.ty[i] = 0
45            if self.ty[i] == testData.rowOutputList[i][0]:
46                self.correct += 1
47
48        self.correctRate = self.correct / float(testData.rowNum)
49
50    # Print estimation accuracy
51    def printAccuracy(self):
52        print("Logistic: Accuracy = %f"%(self.correctRate))
53    # Print parameter list
54    def printTheta(self):
55        index = 0
56        for i in (self.theta):
57            print("Logistic Regression: theta[%d] = %f"%(index, i))
58            index += 1
59    # Print LL
60    def printLL(self, inputData):
61        self.LL = 0
62        self.LL0 = 0
63        for k in range(0, inputData.rowNum):
64            x_now = numpy.append(numpy.array([1]), inputData.rowInputList[k])
65            y = inputData.rowOutputList[k][0]
66            s = self.sigmoid(numpy.sum(x_now * self.theta))
67            s0 = self.sigmoid(0)
68            self.LL += (y*numpy.log(s))+(1-y)*numpy.log(1-s)
69            self.LL0 += (y * numpy.log(s0)) + (1 - y) * numpy.log(1 - s0)
70        print("LL = %f"%(self.LL))
71        print("LL0 = %f" % (self.LL0))

```

72  
73  
74  
75  
76