**Practical no –7**

**Code:**

class Perceptron :

def AdaBoost(self, examples, K):

w = []

N = len(examples[1])

y = examples[1]

for i in range (0,N):

w.append(1/N)

print("original w:", w)

h = []

for k in range (0,K) :

print("K=", k+1)

h.append([])

h = self.L(examples,w)

error = 0

for j in range(0, N) :

#print("pair :", h[j], ":" ,y[j])

#there is huge difference in the h result and actual answer

if( h[j] != y[j] ):

error = error + w[j]

print("error :", error)

for j in range(0, N) :

if( h[j] == y[j] ):# this condition never becomes true in current example

w[j] = w[j] \* error / (1-error) # never gets executed

self.normalize(w)

print(w)

print("result using final w :")

h = self.L(examples,w)

print(h)

#not sure what is to be done with z

def normalize(self, w): # not sure how to normalize

for t in range (0,len(w)) :

normalizer = 1 / float( sum(w) )

w = [x \* normalizer for x in w]

def L(self, ex, w) : # Learning function

hresult= []

for i in range(0 , len(ex[1])):

hresult.append(0)

hresult[i] = hresult[i] + ( w[i]\*ex[0][i] )

return hresult

# the first list x-vector indicates a customer's number in the queue of the waiting list say 1,2...

**Roll no – 17,19**

the second list y-vector indicates amount in minutes the customer will have to wait for the corresponding queue number

ex = [ [1,2,3,4,5,6] , [15,20,30,40,45,60] ]

k = 30

p = Perceptron()

p.AdaBoost(ex,k)

# we are no where close to the actual results

**Output:**

original w: [0.16666666666666666, 0.16666666666666666, 0.16666666666666666, 0.16666666666666666, 0.16666666666666666, 0.16666666666666666]

K= 1

error : 0.9999999999999999

[0.16666666666666666, 0.16666666666666666, 0.16666666666666666, 0.16666666666666666, 0.16666666666666666, 0.16666666666666666]

K= 2

error : 0.9999999999999999

[0.16666666666666666, 0.16666666666666666, 0.16666666666666666, 0.16666666666666666, 0.16666666666666666, 0.16666666666666666]

..

…

…

Attempt : 53

Ans : [0.5, 1.1, 1.2, 1.9567633533440096e+27]

result of act fun: [0, 1, 1, 1]

In Attempt number 53 , i got it! I think i have learnt enough. Your w's are --

w 0 --> [0.5, 0.7, 0.7, 6.522544511146698e+26]

w 1 --> [0.30000000000000004, 0.5, 0.5, 6.522544511146698e+26]

w 2 --> [0.20000000000000007, 0.4, 0.4, 6.522544511146698e+26]

Example 3 -->

Attempt : 1

Ans : [0.2, 0.7, 0.5, 1.0]

result of act fun: [0, 1, 0, 1]

Attempt : 2

Ans : [0.203, 0.7, 0.532, 1.045]

result of act fun: [0, 1, 1, 1]

In Attempt number 2 , i got it! I think i have learnt enough. Your w's are --

w 0 --> [0.203, 0.2, 0.21600000000000003, 0.21500000000000002]

w 1 --> [0.303, 0.3, 0.316, 0.315]

w 2 --> [0.503, 0.5, 0.516, 0.515]