Aim: - To implement and observe output of One Node NN

CODE: -

import pandas as pd

import numpy as np

import math

import random

import sklearn

data = pd.read\_csv("diabetes.csv")

data.insert(0,'BIAS',1)

inp = np.array(data.iloc[:,:-1])

outp = np.array(data.iloc[:,-1])

learning\_rate = 1

print('INPUT: ',inp)

print('OUTPUT: ',outp)

print('LEARNING RATE: ',learning\_rate)

print('\n')

weights = np.random.uniform(low = -0.5,high = 0.5,size = (9,))

np.transpose(weights)

print('initial weights(random): ',weights)

def feedForward(input\_values):

sum\_value = (np.dot(input\_values,weights))

if sum\_value>0:

sum\_value = 1

else:

sum\_value = 0

return sum\_value

def wtupdate(weights, predicted\_output, actual\_output, inp):

for i in range(0,len(weights)):

weights[i] = weights[i] + learning\_rate\*(actual\_output - predicted\_output)\*inp[i]

return weights

epoch = 1000

for j in range(0,epoch):

error = 0

for i in range(0,len(inp)):

predicted\_output = feedForward(inp[i])

if predicted\_output!=outp[i]:

weights = wtupdate(weights, predicted\_output, outp[i], inp[i])

error = error+abs(predicted\_output-outp[i])

accuracy = (len(outp)-error)/len(outp)

print('ITERATION {}:'.format(j),weights,accuracy,'\n')

if error ==0:

break

if error != 0:

print('Unable to linearly seperate the function')

else:

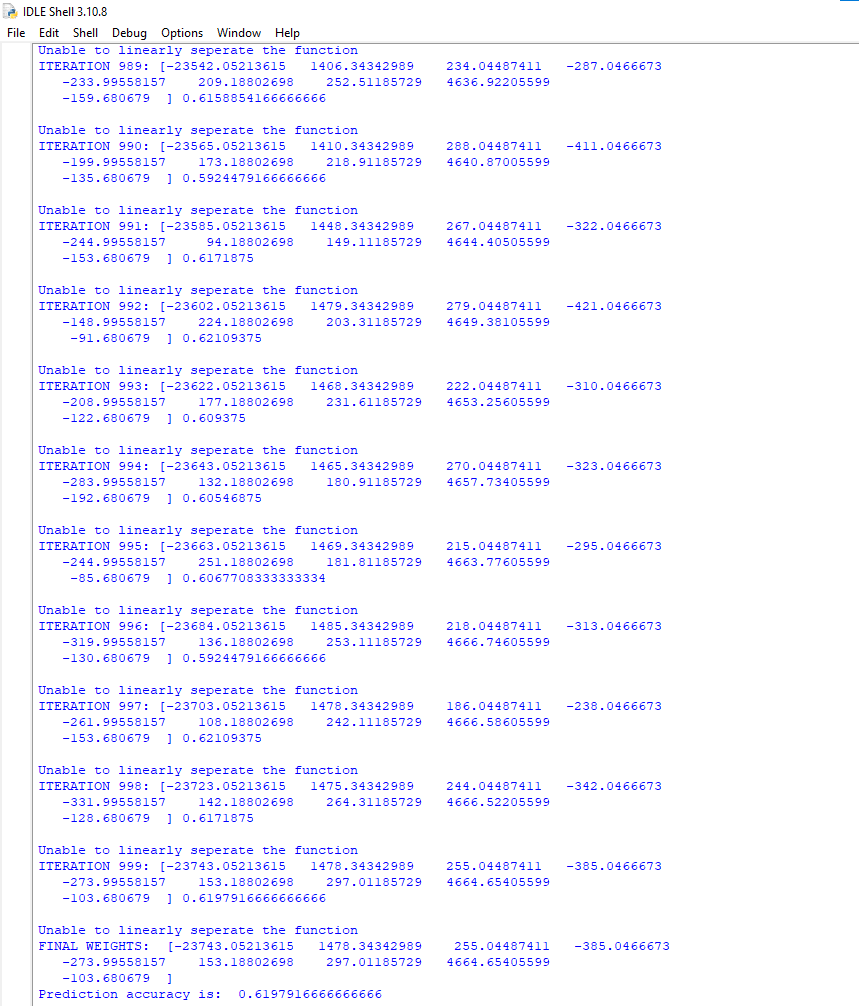
print('The function is linearly separable')

print('FINAL WEIGHTS: ', weights)

acc = accuracy.mean()

print('Prediction accuracy is: ',acc)

Output: -



Result: -

Hence verified and got the output of the tasks on IDLE Python.