Name :-Kunal R Agarwal

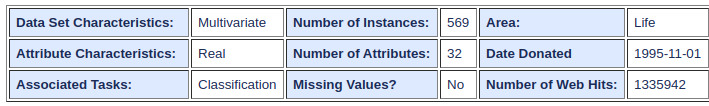
SRN No:-201700110

Roll No:-18

Question:-**Implement the McCulloch-Pitts model**

**Write a program using python**

**Input:-**



**Code:-**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

import sklearn.datasets

cancer = sklearn.datasets.load\_breast\_cancer()

data = pd.DataFrame(cancer.data,columns=cancer.feature\_names)

data["Class"] = cancer.target

x = data.drop('Class',axis=1)

y = data['Class']

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.2,random\_state=3,stratify = y)

x\_train\_binarized = x\_train.apply(pd.cut,bins = 2 ,labels = [1,0]).values

x\_test\_binarized = x\_test.apply(pd.cut,bins=2,labels =[1,0]).values

class MP\_Neuron:

def \_\_init\_\_(self):

self.b = 0

def Model(self,x):

return np.sum(x)>=self.b

def fit(self,x,y):

accuracy = {}

for b in range(x.shape[1] + 1):

self.b = b

yhat = []

for row in x:

yhat.append(self.Model(row))

accuracy[b] = accuracy\_score(yhat,y)

best\_b = max(accuracy,key = accuracy.get)

self.b = best\_b

return [accuracy,best\_b,accuracy[best\_b]]

def predict(self,x,y):

yhat = []

for row in x:

yhat.append(self.Model(row))

accuracy = accuracy\_score(y,yhat)

return accuracy

neuron = MP\_Neuron()

accuracy, best\_b, accuracy\_model = neuron.fit(x\_train\_binarized,y\_train)

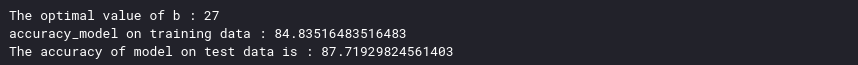
print("The optimal value of b :",best\_b)

print("accuracy\_model on training data :",accuracy\_model\*100)

accuracy = neuron.predict(x\_test\_binarized,y\_test)

print("The accuracy of model on test data is :",accuracy\*100)

**Output:-**

****