|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| sepal\_length | sepal\_width | petal\_length | petal\_width | class |  |
| **TRAINING DATA** 5.1 | 3.5 | 1.4 | 0.2 | 0 | Setosa |
| 4.8 | 3 | 1.4 | 0.3 | 0 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 5.1 | 3.8 | 1.9 | 0.4 | 0 |  |
| 6.7 | 3 | 5 | 1.7 | 1 | Versicolour |
| 5.4 | 3.4 | 1.5 | 0.4 | 0 |  |
| 5.5 | 2.4 | 3.7 | 1 | 1 |  |
| 6.2 | 2.2 | 4.5 | 1.5 | 1 |  |
| 6 | 3.4 | 4.5 | 1.6 | 1 |  |
|  |  |  |  |  |  |
| 5.5 | 2.4 | 3.8 | 1.1 | 1 |  |
|  |  |  |  |  |  |
| 5.6 | 3 | 4.5 | 1.5 | 1 |  |
| 5.4 | 3.9 | 1.3 | 0.4 | 0 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 4.5 | 2.3 | 1.3 | 0.3 | 0 |  |
| 5.5 | 3.5 | 1.3 | 0.2 | 0 |  |
| 5.5 | 4.2 | 1.4 | 0.2 | 0 |  |
| 6.1 | 2.9 | 4.7 | 1.4 | 1 |  |
| 5.1 | 2.5 | 3 | 1.1 | 1 |  |
| 6.9 | 3.1 | 4.9 | 1.5 | 1 |  |
| 6.2 | 2.9 | 4.3 | 1.3 | 1 |  |
| 5.9 | 3.2 | 4.8 | 1.8 | 1 |  |
| 4.9 | 2.4 | 3.3 | 1 | 1 |  |
|  |  |  |  |  |  |
| 5.3 | 3.7 | 1.5 | 0.2 | 0 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 5 | 3.4 | 1.6 | 0.4 | 0 |  |
| 6.4 | 3.2 | 4.5 | 1.5 | 1 |  |
| 5.2 | 2.7 | 3.9 | 1.4 | 1 |  |
| 4.9 | 3 | 1.4 | 0.2 | 0 |  |
| 5.6 | 2.7 | 4.2 | 1.3 | 1 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 5.1 | 3.7 | 1.5 | 0.4 | 0 |  |
| 4.9 | 3.1 | 1.5 | 0.1 | 0 |  |
| 5.5 | 2.6 | 4.4 | 1.2 | 1 |  |
| 5.2 | 3.4 | 1.4 | 0.2 | 0 |  |
| 4.4 | 2.9 | 1.4 | 0.2 | 0 |  |
|  |  |  |  |  |  |
| 6 | 2.7 | 5.1 | 1.6 | 1 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 4.8 | 3 | 1.4 | 0.1 | 0 |  |
| 5.7 | 3 | 4.2 | 1.2 | 1 |  |
| 6.1 | 2.8 | 4 | 1.3 | 1 |  |
| 5.1 | 3.8 | 1.6 | 0.2 | 0 |  |
| 5 | 2.3 | 3.3 | 1 | 1 |  |
| 5.4 | 3 | 4.5 | 1.5 | 1 |  |
| 5 | 3 | 1.6 | 0.2 | 0 |  |
| 5.2 | 3.5 | 1.5 | 0.2 | 0 |  |
|  |  |  |  |  |  |
| 5.8 | 2.7 | 4.1 | 1 | 1 |  |
| 4.6 | 3.2 | 1.4 | 0.2 | 0 |  |
| 5 | 3.2 | 1.2 | 0.2 | 0 |  |
|  |  |  |  |  |  |
| 5.7 | 2.8 | 4.5 | 1.3 | 1 |  |
|  |  |  |  |  |  |
| 5.1 | 3.3 | 1.7 | 0.5 | 0 |  |
| 4.7 | 3.2 | 1.6 | 0.2 | 0 |  |
|  |  |  |  |  |  |
| 5.8 | 2.6 | 4 | 1.2 | 1 |  |
| 5.1 | 3.5 | 1.4 | 0.3 | 0 |  |
| ---------------- **TEST DATA** |  |  |  |  |  |
| 5.6 | 2.9 | 3.6 | 1.3 | 1 |  |
| 5 | 3.4 | 1.5 | 0.2 | 0 |  |
| 6.1 | 2.8 | 4.7 | 1.2 | 1 |  |
| 5.9 | 3 | 4.2 | 1.5 | 1 |  |
|  |  |  |  |  |  |
| 5 | 3.6 | 1.4 | 0.2 | 0 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 6.4 | 2.9 | 4.3 | 1.3 | 1 |  |
| 5 | 3.5 | 1.6 | 0.6 | 0 |  |
| 4.4 | 3.2 | 1.3 | 0.2 | 0 |  |
|  |  |  |  |  |  |
| 5 | 2 | 3.5 | 1 | 1 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 5.4 | 3.7 | 1.5 | 0.2 | 0 |  |
| 6.3 | 2.3 | 4.4 | 1.3 | 1 |  |
| 4.8 | 3.4 | 1.6 | 0.2 | 0 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 5.2 | 4.1 | 1.5 | 0.1 | 0 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 4.8 | 3.1 | 1.6 | 0.2 | 0 |  |
| 5.6 | 2.5 | 3.9 | 1.1 | 1 |  |

Sepal length

5.1,4.8,5.1,6.7,5.4,5.5,6.2,6.0,5.5,5.6,5.4,4.5,5.5,5.5,6.1,5.1,6.9,6.2,5.9,4.9,5.3,5.0,6.4,5.2,4.9,5.6,5.1,4.9,5.5,5.2,4.4,6.0,4.8,5.7,6.1,5.1,5.0,5.4,5.0,5.2,5.8,4.6,5.0,5.7,5.1,4.7,5.8,5.1,5.6

Sepal width

{3.5,3.0,3.8,3.0,3.4,2.4,2.2,3.4,2.4,3.0,3.9,2.3,3.5,4.2,2.9,2.5,3.1,2.9,  
 3.2,2.4,3.7,3.4,3.2,2.7,3.0,2.7,3.7,3.1,2.6,3.4,2.9,2.7,3.0,3.0,2.8,  
 3.8,2.3,3.0,3.0,3.5,2.7,3.2,3.2,2.8,3.3,3.2,2.6,3.5

Petal length

1.4,1.4,1.9,5.0,1.5,3.7,4.5,4.5,3.8,4.5,1.3,1.3,1.3,1.4,4.7,3.0,4.9,4.3,  
 4.8,3.3,1.5,1.6,4.5,3.9,1.4,4.2,1.5,1.5,4.4,1.4,1.4,5.1,1.4,4.2,4.0,  
 1.6,3.3,4.5,1.6,1.5,4.1,1.4,1.2,4.5,1.7,1.6,4.0,1.4

Petal Width

0.2,0.3,0.4,1.7,0.4,1.0,1.5,1.6,1.1,1.5,0.4,0.3,0.2,0.2,1.4,1.1,1.5,1.3,  
 1.8,1.0,0.2,0.4,1.5,1.4,0.2,1.3,0.4,0.1,1.2,0.2,0.2,1.6,0.1,1.2,1.3,  
 0.2,1.0,1.5,0.2,0.2,1.0,0.2,0.2,1.3,0.5,0.2,1.2,0.3

Y

0,0,0,1,0,1,1,1,1,1,0,0,0,0,1,1,1,1,1,1,0,0,1,1,0,1,0,0,1,0,0,1,0,1,1,0,  
 1,1,0,0,1,0,0,1,0,0,1,0

//  
// Created by Pushkal Shukla on 16-04-2021.  
//  
//  
// Created by Pushkal Shukla on 16-04-2021.  
//  
#include<bits/stdc++.h> // header file for all c++ libraries  
using namespace std; // stdout library for printing values  
bool custom\_sort(double a, double b) /\* this custom sort function is defined to  
 sort on basis of min absolute value or error\*/  
{  
 double a1=abs(a-0);  
 double b1=abs(b-0);  
 return a1<b1;  
}  
int main()  
{  
/\*Intialization Phase\*/  
 double SL[] = {5.1,4.8,5.1,6.7,5.4,5.5,6.2,6.0,5.5,5.6,5.4,4.5,5.5,5.5,6.1,5.1,6.9,6.2,  
 5.9,4.9,5.3,5.0,6.4,5.2,4.9,5.6,5.1,4.9,5.5,5.2,4.4,6.0,4.8,5.7,6.1,  
 5.1,5.0,5.4,5.0,5.2,5.8,4.6,5.0,5.7,5.1,4.7,5.8,5.1};  
 double SW[] = {3.5,3.0,3.8,3.0,3.4,2.4,2.2,3.4,2.4,3.0,3.9,2.3,3.5,4.2,2.9,2.5,3.1,2.9,  
 3.2,2.4,3.7,3.4,3.2,2.7,3.0,2.7,3.7,3.1,2.6,3.4,2.9,2.7,3.0,3.0,2.8,  
 3.8,2.3,3.0,3.0,3.5,2.7,3.2,3.2,2.8,3.3,3.2,2.6,3.5};  
 double PL[] = {1.4,1.4,1.9,5.0,1.5,3.7,4.5,4.5,3.8,4.5,1.3,1.3,1.3,1.4,4.7,3.0,4.9,4.3,  
 4.8,3.3,1.5,1.6,4.5,3.9,1.4,4.2,1.5,1.5,4.4,1.4,1.4,5.1,1.4,4.2,4.0,  
 1.6,3.3,4.5,1.6,1.5,4.1,1.4,1.2,4.5,1.7,1.6,4.0,1.4};  
 double PW[] = {0.2,0.3,0.4,1.7,0.4,1.0,1.5,1.6,1.1,1.5,0.4,0.3,0.2,0.2,1.4,1.1,1.5,1.3,  
 1.8,1.0,0.2,0.4,1.5,1.4,0.2,1.3,0.4,0.1,1.2,0.2,0.2,1.6,0.1,1.2,1.3,  
 0.2,1.0,1.5,0.2,0.2,1.0,0.2,0.2,1.3,0.5,0.2,1.2,0.3};  
 double y[] = {0,0,0,1,0,1,1,1,1,1,0,0,0,0,1,1,1,1,1,1,0,0,1,1,0,1,0,0,1,0,0,1,0,1,1,0,  
 1,1,0,0,1,0,0,1,0,0,1,0};  
 vector<double>error; // for storing the error values  
 double err; // for calculating error on each stage  
 double b0 = 0; // initializing b0  
 double SL1 = 0; // initializing b1  
 double SW1 = 0;  
 double PL1 = 0;  
 double PW1 = 0;  
 double alpha = 0.01; // initializing our learning rate  
 double e = 2.71828;  
  
/\*Training Phase\*/  
 for (int i = 0; i < 192; i ++) { //Since there are 48 values in our dataset and we want to run for 4 epochs so total for loop run 192 times  
 int idx = i % 48; //for accessing index after every epoch  
 double p = -(b0 + SL1\* SL[idx]+ SW1\* SW[idx]+ PL1\* PL[idx]+ PW1\* PW[idx]); //making the prediction  
 double pred = 1/(1+ pow(e,p)); //calculating final prediction applying sigmoid  
 err = y[idx]-pred; //calculating the error  
 b0 = b0 - alpha \* err\*pred \*(1-pred)\* 1.0; //updating b0  
 SL1 = SL1 + alpha \* err \* pred\*(1-pred) \* SL[idx];//updating b1  
 SW1 = SW1 + alpha \* err \* pred\*(1-pred) \* SW[idx];  
 PL1 = SW1 + alpha \* err \* pred\*(1-pred) \* PL[idx];  
 PW1 = SW1 + alpha \* err \* pred\*(1-pred) \* PW[idx];//updating b2  
 cout<<"B0="<<b0<<" "<<"SL1="<<SL1<<" "<<"SW1="<<SW1<<" "<<"PL1="<<PL1<<" "<<"PW1="<<PW1<<" "  
 <<" error="<<err<<endl;// printing values after every step  
 error.push\_back(err);  
 }  
 sort(error.begin(),error.end(),custom\_sort);//custom sort based on absolute error difference  
 cout<<"Final Values are: "<<"B0="<<b0<<" "<<"SL1="<<SL1<<" "<<"SW1="<<SW1<<" "<<"PL1="<<PL1<<" "  
 <<"PW1"<<PW1<<" "<<" error="<<error[0];  
  
/\*Testing Phase\*/  
 double test1,test2,test3,test4; //enter test SL,SW,PL and PW  
 cin>>test1>>test2>>test3>>test4;  
 double pred = b0 + SL1\*test1 + SW1\*test2 + PL1\*test3 + PW1\*test4; //make prediction  
 cout<<"The value predicted by the model= "<<pred<<endl;  
 if(pred>0.5)  
 pred=1;  
 else  
 pred=0;  
 cout<<"The class predicted by the model= "<<pred;  
}