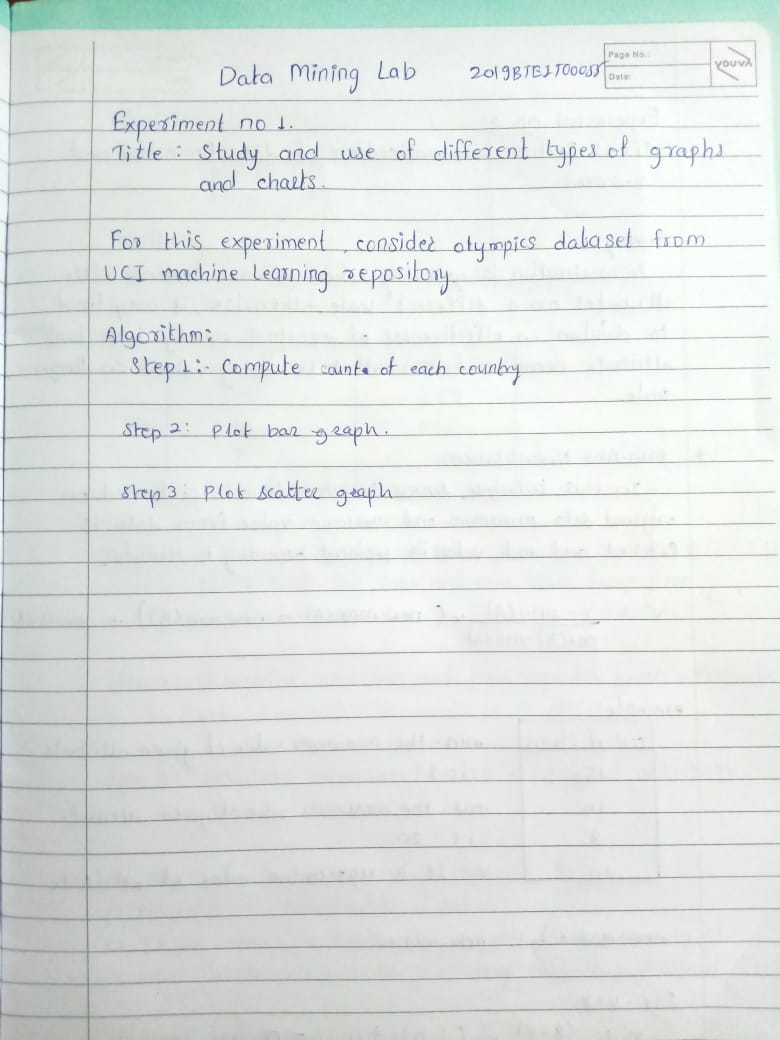
**Walchand College Of Engineering, Sangli**

**Information Technology**

**Data Mining Lab**

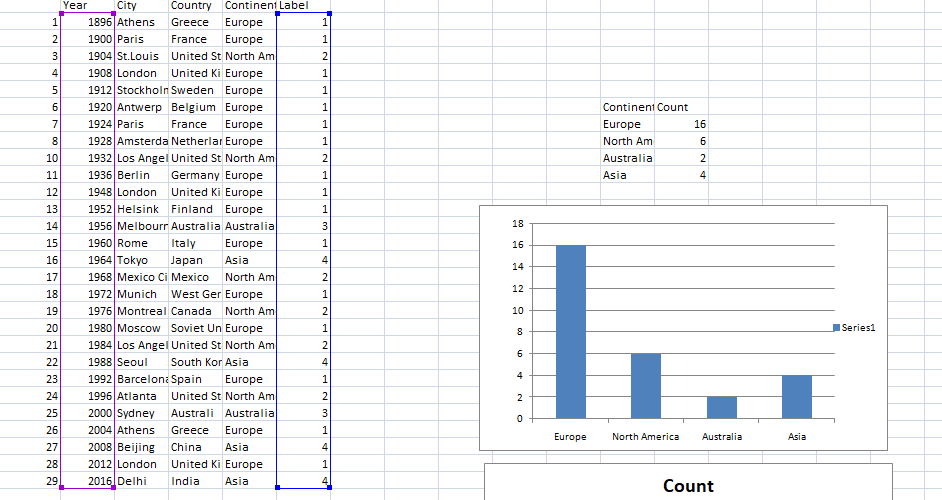
**PRN - 2019BTEIT00060**

**Bhushan Rajendra Jadhav**

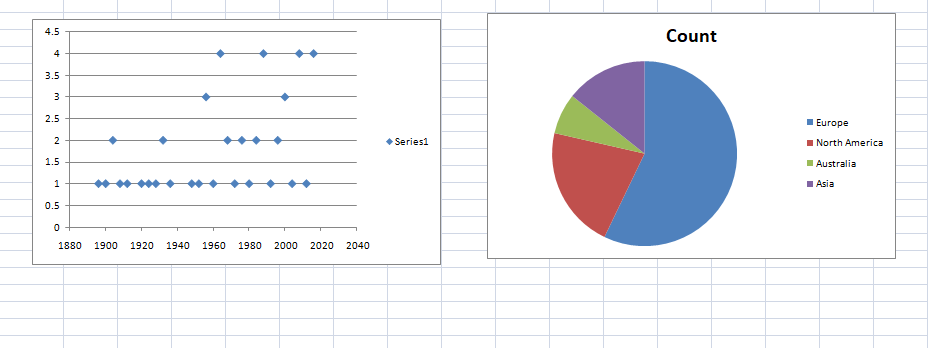
**Experiment No 1: Study and use of different types of graphs **

**Data Set(Winter Olympics game Hosted CIties)**

**Output 1: (Pie Chart)**

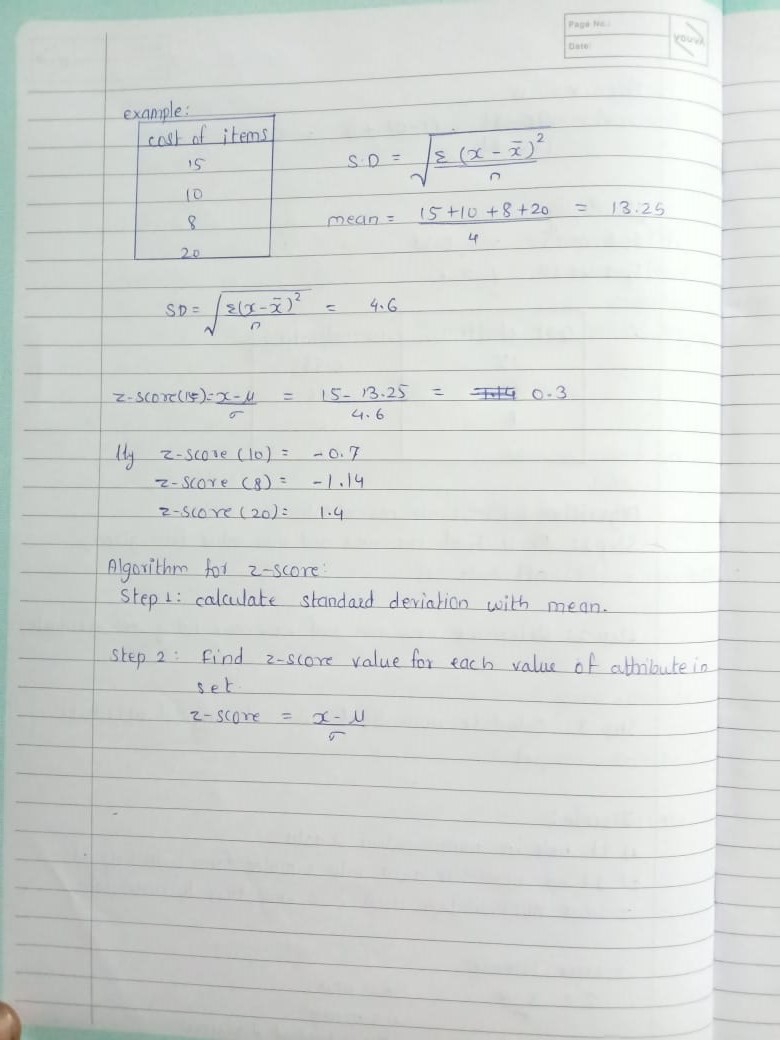
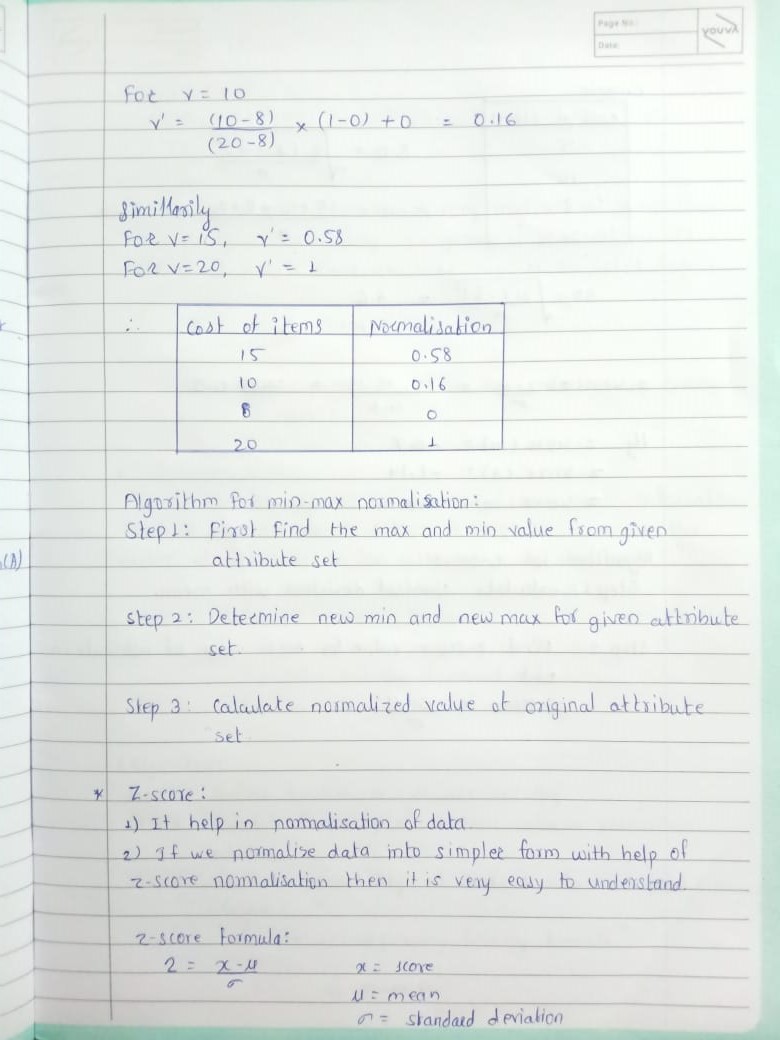
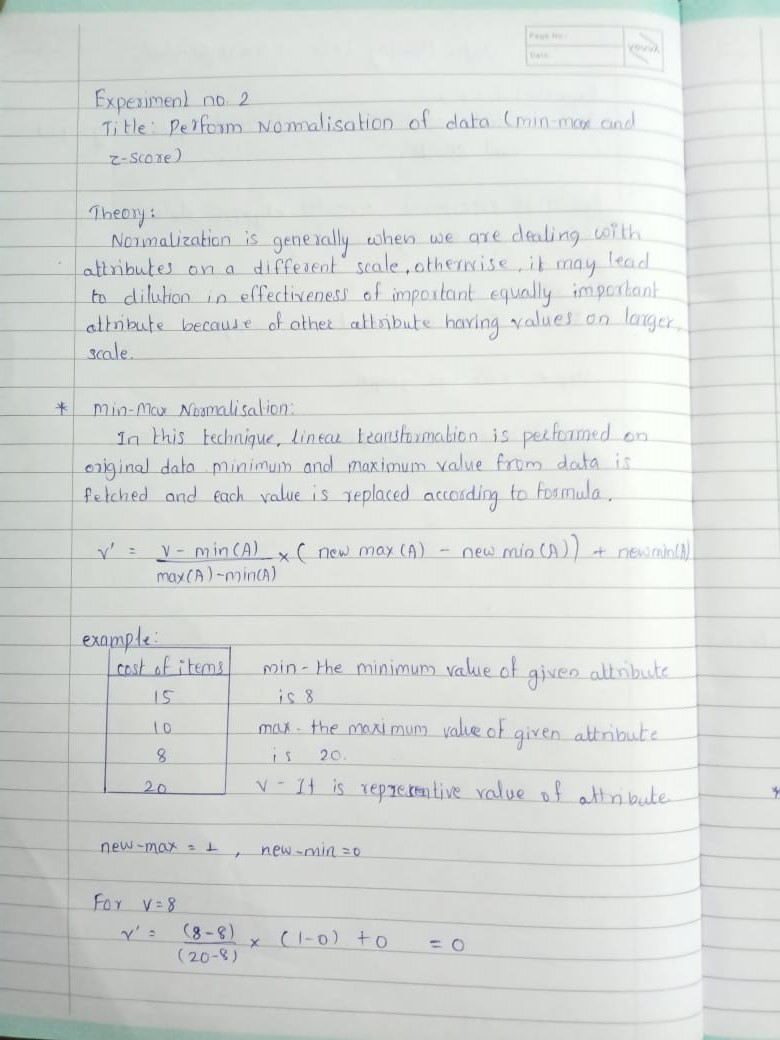
****

**2)bar chart and scatter plot**



**Conclusion: By using the Winter Olympics game Hosted CIties dataset, we are able to understand the various graphs, charts. Also I draw graph for data set by performing some calculations.**

**Experiment No.2**

**Perform Normalization of Data(Min-Max and Z-score)**

Program:

import java.util.\*;

import java.io.\*;

class MinMaxNorm {

public static void main(String[] args) throws Exception{

BufferedReader br = new BufferedReader(new FileReader("C://Users/Admin/Desktop/DMLAB/MinMaxNorm/Datafile.txt"));

String line = br.readLine();

String[] items = line.split(" ");

int [] arr = new int[items.length];

int i = 0;

for(String item : items) {

arr[i++] = Integer.parseInt(item);

}

System.out.println("Data : ");

System.out.println(Arrays.toString(arr));

int minValue = arr[0], maxValue = arr[0];

for(int item : arr) {

if(minValue > item) {

minValue = item;

} else if(maxValue < item) {

maxValue = item;

}

}

System.out.println("\nMin Value : " + minValue + "\nMax Value : " + maxValue);

int newMin = 0, newMax = 10;

System.out.println("\nNormalizing Data with max = "+ newMax +" & min = " + newMin);

double [] normArr = new double[arr.length];

i = 0;

for(int item : arr)

{

double a = (item - minValue) / ((maxValue - minValue) \* 1.0);

double b = newMax - newMin;

normArr[i++] = a \* b + newMin;

}

for(i=0; i<arr.length; i++)

{

System.out.printf(arr[i] + "---Normalized value--> %.4f\n", normArr[i]);

}

double mean = 0, sum = 0;

for(double item : normArr) {

sum += item;

}

int n = arr.length;

mean = sum / n;

double stdDev = 0.0;

double sqrsum = 0.0;

for(i = 0; i < n; i++)

{

sqrsum += (arr[i] - mean) \* (arr[i] - mean);

}

sqrsum /= n;

stdDev = Math.sqrt(sqrsum);

System.out.printf("\nMean : %.4f \nStandard Deviation : %.4f", mean, stdDev);

double zscore[] = new double[n];

for(i = 0;i < n; i++)

{

zscore[i] = (arr[i] - mean) / stdDev;

}

System.out.println("\nData item -------------> zscore");

for(i = 0; i < n; i++)

{

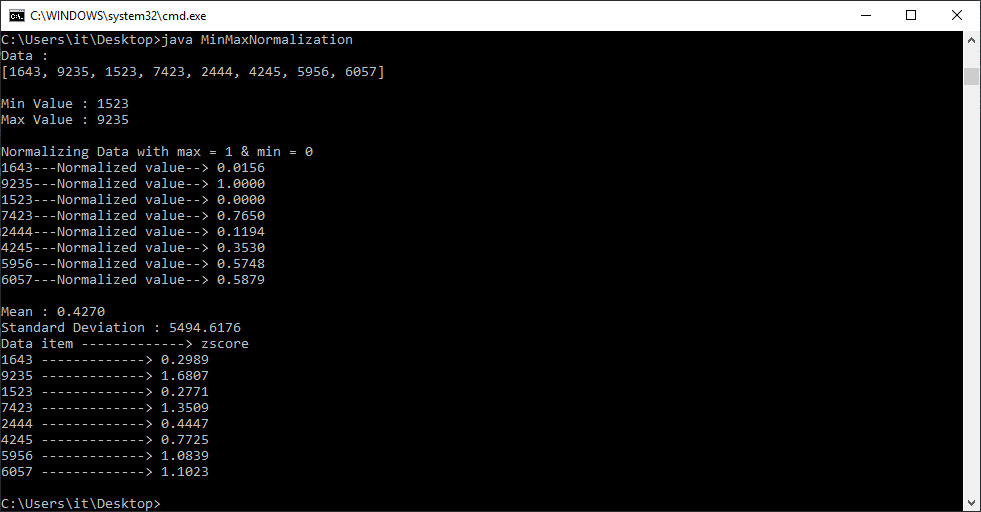
System.out.printf("%d -------------> %.4f\n", arr[i], zscore[i]);

}

}

}

Output

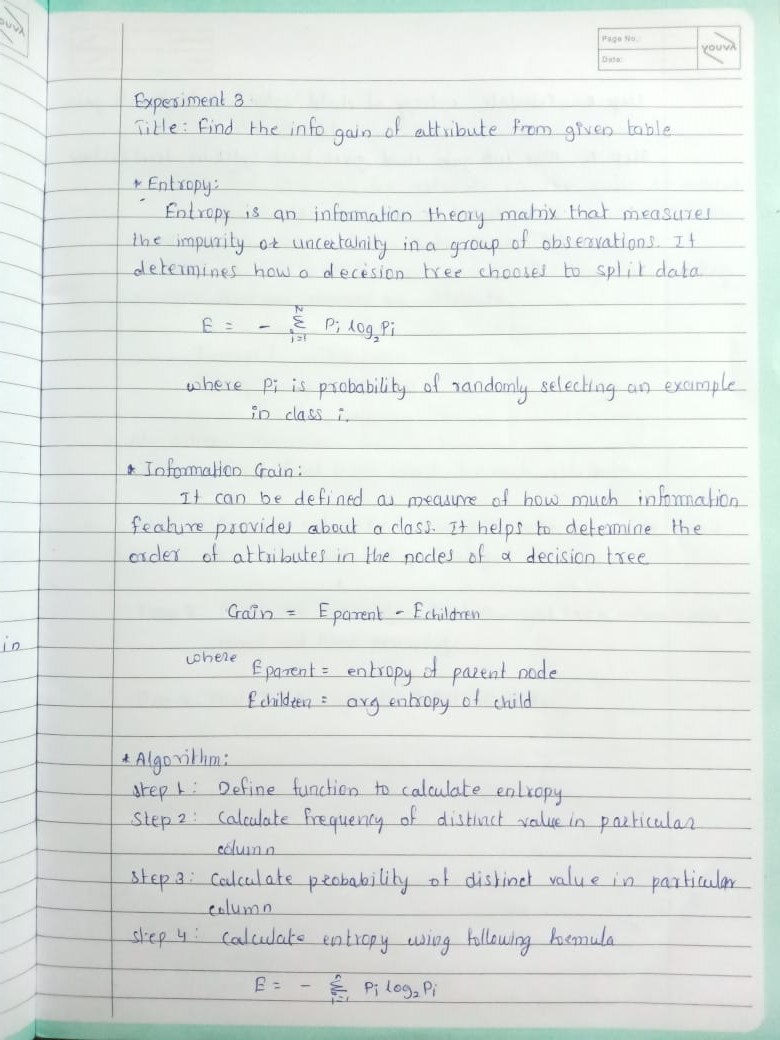


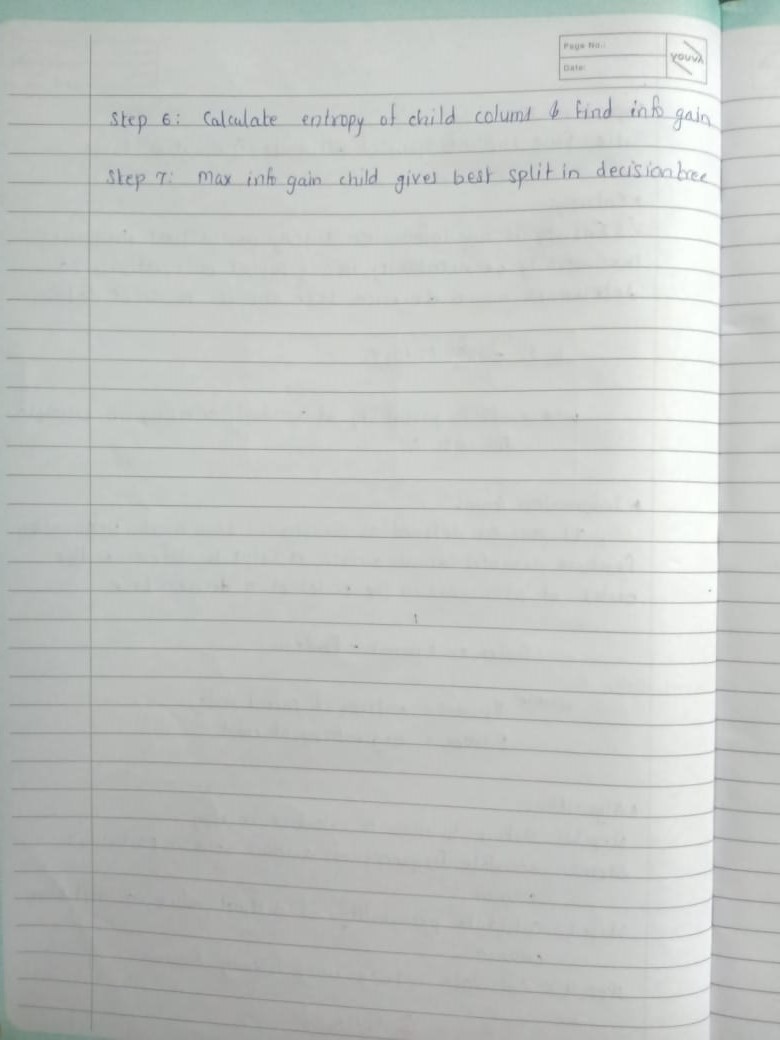
**Conclusion:**

**Successfully performed min-max and z-score normalization with code**

**implementation.**

**Experiment 3: Find Info Gain of an attribute from given data.**

****

****

Program :

import java.io.BufferedReader;

import java.io.FileReader;

import java.util.\*;

class InfoGain {

public static void main(String[] args) throws Exception {

BufferedReader br = new BufferedReader(new FileReader("C:/Users/Admin/Desktop/DMLAB/InfoGain/InfoGainData.csv"));

ArrayList<ArrayList<String>> data = new ArrayList<>();

int n = br.readLine().trim().split(",").length;

while (true) {

String str[] = new String[n];

try {

str = br.readLine().trim().split(",");

} catch (NullPointerException e) {

break;

}

ArrayList<String> temp = new ArrayList<>();

for (String s : str)

temp.add(s);

data.add(temp);

}

int posR = 0, negR = 0;

for (int i = 0; i < data.size(); i++) {

if (data.get(i).get(5).equals("Yes"))

posR++;

else

negR++;

}

double totRecord = data.size();

double entropy = -((1.0 \* posR / totRecord) \* (Math.log(1.0 \* posR / totRecord)) / Math.log(2)) - ((1.0 \* negR / totRecord) \* (Math.log(1.0 \* negR / totRecord) / Math.log(2)));

System.out.println("Entropy of play: "+entropy);

for(int i=0;i<data.get(0).size()-1;i++){

System.out.println("Info Gain of column "+i+": "+findAllEntropy(i,entropy,data));

}

}

private static double findAllEntropy(int i, double entropy, ArrayList<ArrayList<String>> data) {

Set<String> attribute=new HashSet<>();

for(ArrayList<String>x:data)

attribute.add(x.get(i));

Map<String,double[]>total=new HashMap<>();

for(String x:attribute){

total.put(x,new double[2]);

}

for(ArrayList<String> x:data){

if(x.get(5).equals(("Yes")))

total.get(x.get(i))[0]++;

else

total.get(x.get(i))[1]++;

}

double totalE=0.0;

for(Map.Entry<String,double[]> x:total.entrySet()){

double total1=x.getValue()[0]+x.getValue()[1];

if(x.getValue()[0]==0||x.getValue()[1]==0)

continue;

double temp= -((1.0 \* x.getValue()[0] / total1) \* (Math.log(1.0 \* x.getValue()[0] / total1)) / Math.log(2)) - ((1.0 \* x.getValue()[1]/ total1) \* (Math.log(1.0 \* x.getValue()[1] / total1) / Math.log(2)));

totalE+=(total1/ data.size())\*temp;

}

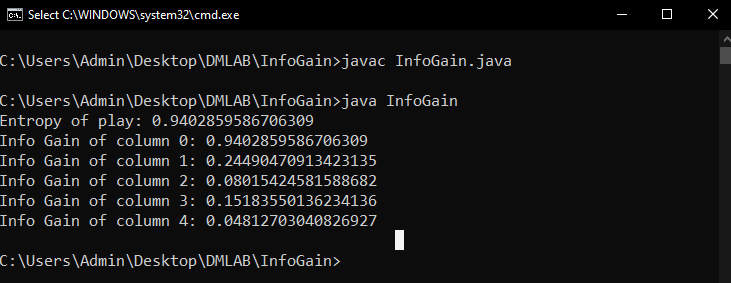
double finalE=entropy-totalE;

return finalE;

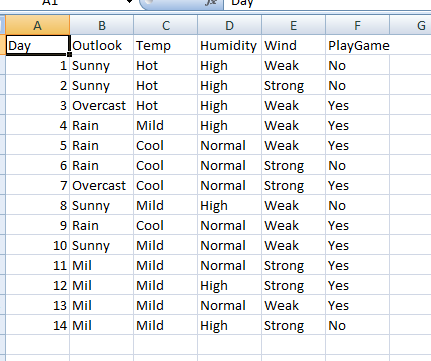
}

}

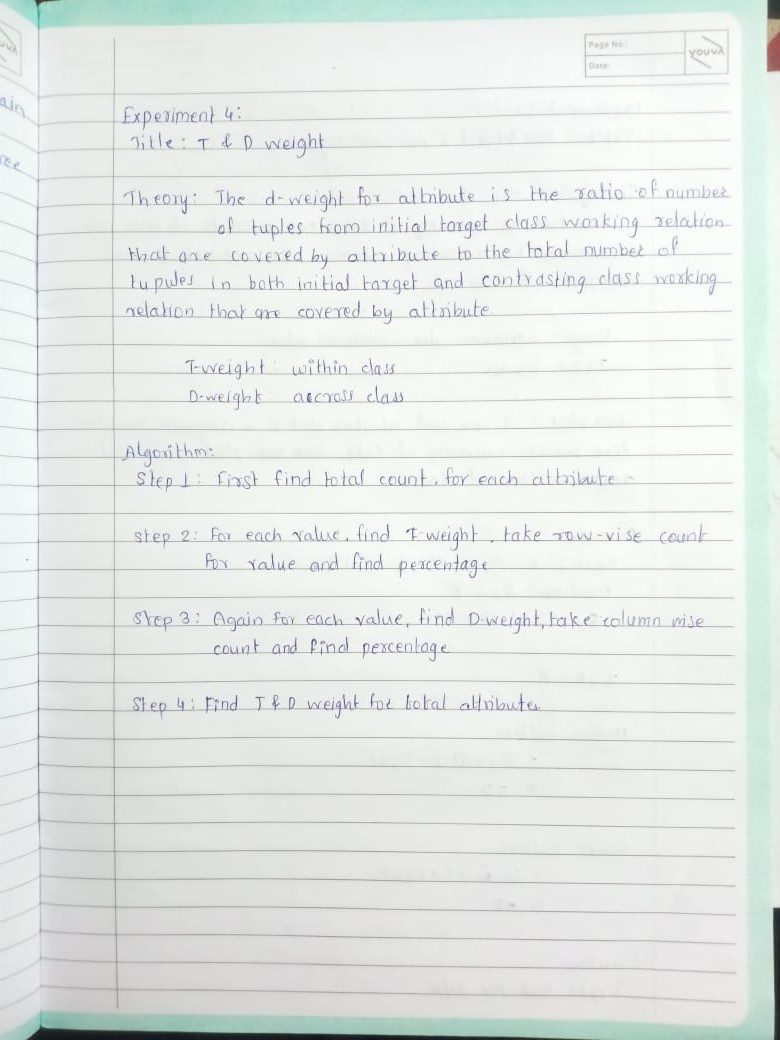
Output (InfoGain of all column)



Dataset

****

**Experiment No 4: find T and D weight of data**

****

Program :

import java.util.\*;

import java.io.\*;

class TDWeight {

public static void main(String[] args) throws Exception {

BufferedReader br = new BufferedReader(new FileReader("C://Users/Admin/Desktop/DMLAB/TDWeight/Data.csv"));

String[] col = br.readLine().trim().split(",");

int[][] data = new int[2][2];

for(int i = 0; i < 2; i++) {

String[] row = br.readLine().trim().split(",");

int j = 0;

for(int k = 1; k <= 2; k++ ) {

data[i][j++] = Integer.parseInt(row[k]);

}

}

int c1\_tot = 0;

int c2\_tot = 0;

int r1\_tot = 0;

int r2\_tot = 0;

for(int i = 0; i < 2; i++) {

for(int j = 0; j < 2; j ++) {

if(i == 0)

r1\_tot += data[i][j];

if(i == 1)

r2\_tot += data[i][j];

if(j == 0)

c1\_tot += data[i][j];

if(j == 1)

c2\_tot += data[i][j];

}

}

System.out.println("row 1 total: " + r1\_tot);

System.out.println("row 2 total: " + r2\_tot);

System.out.println("col 1 total: " + c1\_tot);

System.out.println("col 2 total: " + c2\_tot);

for(int i = 0; i < 2; i++) {

for(int j = 0; j < 2; j++) {

if(i == 0)

System.out.printf("\nt-weight for data " + data[i][j] + "is---> %.4f", (1.0 \* data[i][j] / r1\_tot) \* 100);

if(i == 1)

System.out.printf("\nt-weight for data " + data[i][j] + "is---> %.4f", (1.0 \* data[i][j] / r2\_tot) \* 100);

if(j == 0)

System.out.printf("\nd-weight for data " + data[i][j] + "is---> %.4f", (1.0 \* data[i][j] / c1\_tot) \* 100);

if(j == 1)

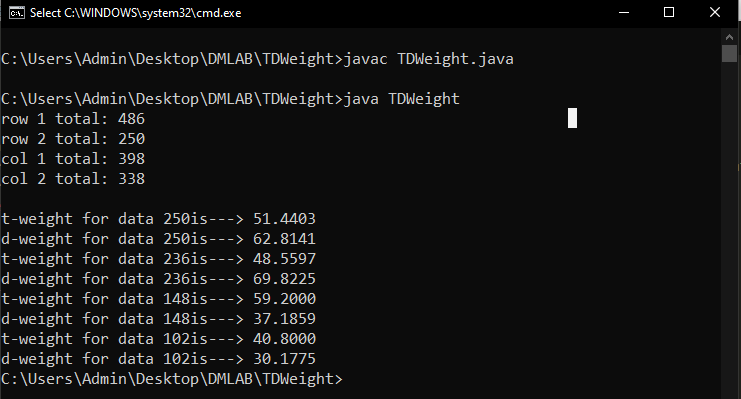
System.out.printf("\nd-weight for data " + data[i][j] + "is---> %.4f", (1.0 \* data[i][j] / c2\_tot) \* 100);

}

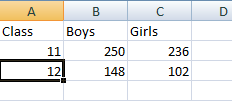
}

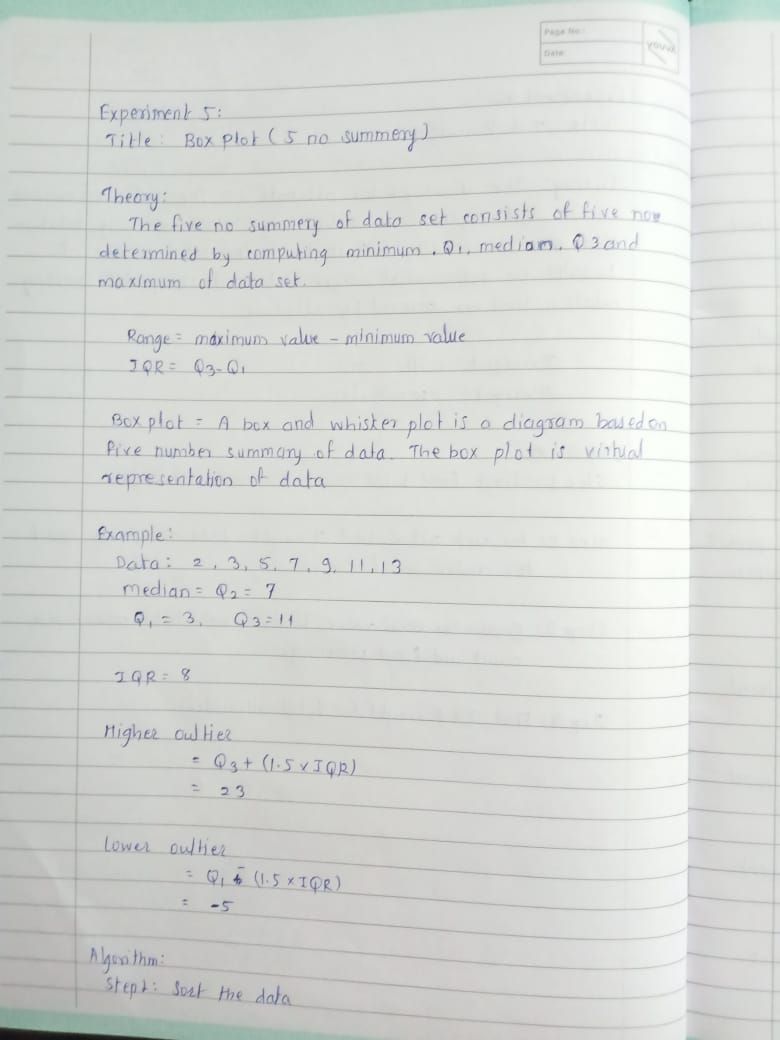
}

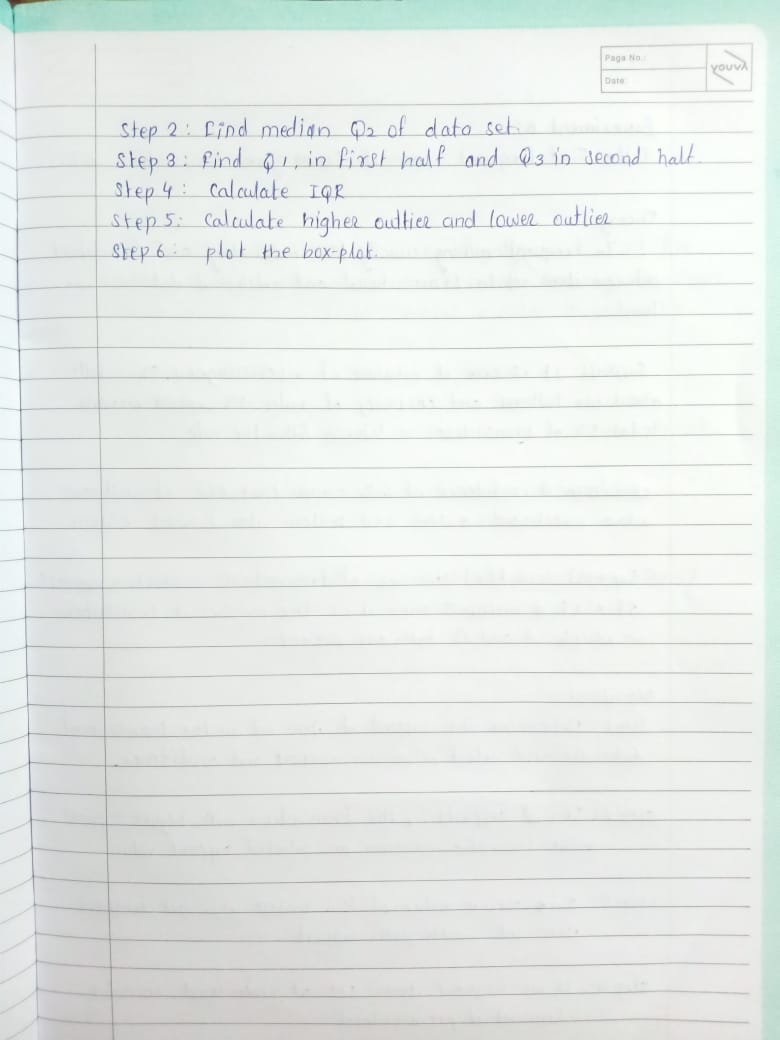
}

output:

DataSet(T and D weight)



**Experiment 5: Find 5 no summary of a dataset.**

****

**program:**

import java.io.BufferedReader;

import java.io.FileReader;

import java.util.Arrays;

public class BoxPlot {

public static void main(String[] args)throws Exception {

BufferedReader br=new BufferedReader(new FileReader("C://Users/Admin/Desktop/DMLAB/BoxPlot/Data.txt"));

String str[]=br.readLine().trim().split(" ");

int arr[]=new int[str.length];

int i=0;

for(String s:str){

arr[i++]=Integer.parseInt(s);

}

System.out.println("Input data");

System.out.println(Arrays.toString(arr));

System.out.println("Sorted data");

Arrays.sort(arr);

System.out.println(Arrays.toString(arr));

int n=arr.length;

System.out.println("1.MAX VALUE--------->"+arr[n-1]);

System.out.println("2.MIN VALUE--------->"+arr[0]);

System.out.println("3.MEDIAN---------->"+arr[(n+1)/2-1]);

System.out.println("4.QUARTILE 1--------->"+arr[(n+1)/4-1]);

System.out.println("5.QUARTILE 3--------->"+arr[3\*(n+1)/4-1]);

}

}

Output :