

```

5) import java.util.*;
import java.io.*;
class temp{
double p;
String q,r;
}
class naive
{
public static double probability(ArrayList<temp> al,String x)
{
    double sum = 0;
    for(temp z:al)
        if((z.q).equals(x))
            sum++;
    return sum / al.size();
}
public static double mean(ArrayList<temp> al)
{
    double sum = 0;
    for (temp z:al) {
        sum += z.p;
    }
    return sum / al.size();
}
public static double sd(ArrayList<temp> al,double mean)
{
    double sum = 0;
    int x = al.size();
    for (temp z:al) {
        sum += Math.pow((z.p-mean),2);
    }
    return Math.sqrt(sum/(x*(x-1)));
}
public static void main(String args[]) throws FileNotFoundException,
IOException
{
    BufferedReader csv = new BufferedReader(new FileReader(new
    File("data.csv")));
    String data = csv.readLine();
    ArrayList<temp> y = new ArrayList<>();
    ArrayList<temp> n = new ArrayList<>();
    int i=0;
    while(data != null)
    {
        String[] dataarray = data.split(",");
        temp res = new temp();
        res.p = Double.parseDouble(dataarray[0]);
        res.q = dataarray[1];
        res.r = dataarray[2];
        if(dataarray[2].equals("Y"))
            y.add(res);
        else
            n.add(res);
        data = csv.readLine();
    }
}

```

```

System.out.println("Enter weight(1.0 - 60.0) and shirt size(S,M,L)
to find its class");
Scanner in = new Scanner(System.in);
double x1 = in.nextDouble();
String x2 = in.next();
//initial probability
int t = y.size() + n.size();
double p_y = (double)y.size()/(double)t;
double p_n = (double)n.size()/(double)t;
//for numeric attributes
double p1,p2,p3,p4,exp;
double mean1 = mean(y);
double mean2 = mean(n);
double sd1 = sd(y,mean1);
double sd2 = sd(n,mean2);
//probability for numeric = (1/(root(2*PI)*sd))exponential(-
square(value-mean)/(2*sd))
//refer class notes for formula
exp = Math.pow((x1 - mean1), 2)/(2 * Math.pow(sd1,2));
p3 = Math.exp(-1*exp)/(Math.sqrt(2 * 3.14) * sd1);
exp = Math.pow((x1 - mean2), 2)/(2 * Math.pow(sd2,2));
p4 = Math.exp(-1*exp)/(Math.sqrt(2 * 3.14) * sd2);
//for non-numeric attributes
p1 = probability(y,x2);
p2 = probability(n,x2);
double res1,res2;
res1 = p1*p3*p_y;
res2 = p2*p4*p_n;
//results
System.out.println("--Class Y--");
System.out.println("Mean :"+mean1+ "\tStandard deviation:  "+sd1);
System.out.println("size :S given class"+i+":
"+probability(y,"S"));
System.out.println("size :M given class"+i+":
"+probability(y,"M"));
System.out.println("size :L given class"+i+":
"+probability(y,"L")+"\n");

System.out.println("--Class N--");
System.out.println("Mean :"+mean2+ "\tStandard deviation:  "+sd2);
System.out.println("size :S given class"+i+":
"+probability(n,"S"));
System.out.println("size :M given class"+i+":
"+probability(n,"M"));
System.out.println("size :L given class"+i+":
"+probability(n,"L")+"\n");

if(res1>res2)
    System.out.println("Class for weight "+x1+" and shirt size
"+x2+" is : Y");
else
    System.out.println("Class for weight "+x1+" and shirt size
"+x2+" is : N");
}

```

Dataset:

20	S	Y
23	L	N
13	M	Y
24	S	Y
45	M	Y
35	M	N
23	S	N
45	M	Y
20	S	Y
23	S	N
13	M	Y
24	M	Y
45	S	Y
35	M	N
23	S	N
45	L	N

```
6) import java.io.*;
import java.util.*;
class comp{
double a;
String b;
int x,y,z;
public comp(double p,String q,int r,int s,int t){
a=p;b=q;x=r;y=s;z=t;
}
}
class prog6
{
public static void main(String args[]) throws FileNotFoundException,
IOException
{
BufferedReader csv = new BufferedReader(new FileReader(new
File("data.csv")));
String data = csv.readLine();
int a[] = new int[3];
int b[] = new int[3];
Scanner in = new Scanner(System.in);
System.out.println("Enter x,y,z to classify in Class A,B,C");
a[0] = in.nextInt();
a[1] = in.nextInt();
a[2] = in.nextInt();
System.out.println("enter number of nearest neighbours");
int k = in.nextInt();
int i;
ArrayList<comp> al = new ArrayList<>();
while(data != null)
{
String[] dataarray = data.split(",");
int sum=0;
```

```

        for(i=0;i<3;i++)
            b[i] = Integer.parseInt(dataarray[i]);
        sum = (a[0]-b[0])*(a[0]-b[0]) + (a[1]-b[1])*(a[1]-b[1]) +
(a[2]-b[2])*(a[2]-b[2]);
        double res = Math.sqrt(sum);
        comp temp = new comp(res,dataarray[3],b[0],b[1],b[2]);
        al.add(temp);
        data = csv.readLine();
    }
    Collections.sort(al, new Comparator<comp>() {
        @Override public int compare(comp p1, comp p2) {
            if(p1.a == p2.a)
                return 0;
            else if(p1.a > p2.a)
                return 1;
            else
                return -1;
        } });
    int x[] = new int[3];
    System.out.println(k+" -nearest neighbour are:");
    for(comp temp:al)
    {
        if(k==0)
            break;
        System.out.println(temp.x+" "+temp.y+" "+temp.z+"
"+temp.b);
        k--;
        String c = temp.b;
        if(c.equals("A"))
            x[0]++;
        else if(c.equals("B"))
            x[1]++;
        else
            x[2]++;
    }
    char res;
    int r = Math.max(x[0],Math.max(x[1],x[2]));
    if(x[0] == r)
        res = 'A';
    else if(x[1] == r)
        res = 'B';
    else
        res = 'C';
    System.out.println("classifier for "+a[0]+" "+a[1]+" "+a[2]+" is
:"+res);
}
}

```

Dataset:

1	2	3	A
2	4	7	B
3	7	5	C
1	7	3	A
1	6	9	B
2	2	2	B
7	6	3	C

1	1	6	A
7	8	4	A
1	8	3	C

```

7) import java.util.*;
import java.io.*;
class temp{
double x,y,z;
}
class prog7
{
public static void main(String args[]) throws FileNotFoundException,
IOException
{
BufferedReader csv = new BufferedReader(new FileReader(new
File("data.csv")));
String data = csv.readLine();
Random rand = new Random();
ArrayList<temp> al = new ArrayList<>();
double[] weight = new double[3];
double rate = 0.02;
while(data != null)
{
String[] dataarray = data.split(",");
temp v = new temp();
v.x = Double.parseDouble(dataarray[0]);
v.y = Double.parseDouble(dataarray[1]);
v.z = Double.parseDouble(dataarray[2]);
al.add(v);
data = csv.readLine();
}
weight[0] = rand.nextInt(32767*2)-32767;
weight[1] = rand.nextInt(32767*2)-32767;
for(int i =0;i<10000;i++)
{
for(int j=0;j<al.size()-1;j++)
{
int res=0;
double sum =
weight[0]+weight[1]*(al.get(j)).x+weight[2]*(al.get(j)).y;
if(sum>0){res=1;}
double error = (al.get(j)).z-res;
weight[0]=weight[0]+error*rate;
weight[1]=weight[1]+(rate*error*(al.get(j)).x);
weight[2]=weight[2]+(rate*error*(al.get(j)).y);
}
}
System.out.println("enter x,y test data");
Scanner in = new Scanner(System.in);
double p = in.nextDouble();
double q = in.nextDouble();
double sum1 = weight[0]+weight[1]*p+weight[2]*q;
if(sum1>0)
System.out.println("predicted :1.0");
}

```

```

else
    System.out.println("predicted :0.0");
System.out.println("final weight[0] :"+weight[0]+"\nfinal weight[1]
:"+weight[1]+"\n final weight[2] :"+weight[2]);
}
}

```

Dataset:

2.7810836	2.550537003	0
1.465489372	2.362125076	0
3.396561688	4.400293529	0
1.38807019	1.850220317	0
7.673756466	3.508563011	1
8.675418651	-0.242068655	1
5.332441248	2.088626775	1
6.922596716	1.77106367	1
3.06407232	3.005305973	0

```

8) import java.io.*;
import java.util.*;
//y=Mx+b;
class prog8
{
public static void main(String args[])
{
    double M=-1000;
    double B=-1000;
    //y=5x+2
    double mean_x=0,mean_y=0,cov_xy=0,var_x=0;
    double[] x = { 1, 2, 3, 4, 5, 6, 7, 8 };
    double[] y = { 7,12,17,22,27,32,37,42};
    for(int i=0;i<x.length;i++)
    {
        mean_x = mean_x+x[i];
        mean_y = mean_y+y[i];
    }
    mean_x = mean_x/x.length;
    mean_y = mean_y/x.length;
    for(int i=0;i<x.length;i++)
    {
        var_x += Math.pow(x[i] - mean_x, 2);
        cov_xy += (x[i] - mean_x) * (y[i] - mean_y);
    }
    var_x = var_x/x.length;
    cov_xy = cov_xy/x.length;
    M = cov_xy/var_x;
    B = mean_y-M*mean_x;
    System.out.println("enter the value for x");
    Scanner in = new Scanner(System.in);
    double r = in.nextInt();
    double res = r*M+B;
    System.out.println("Predicted output for "+r+" : "+res);
} }

```

```

9) import java.util.*;
import java.io.*;
class record{
double attr1=0,attr2=0;
}
class prog9
{
public static record calc(ArrayList<record> bl)
{
    record a = new record();
    double x=0,y=0;
    for(record b:bl){
        x=x+b.attr1;
        y=y+b.attr2;
    }
    a.attr1=x/bl.size();
    a.attr2=y/bl.size();
    return a;
}
public static double dist(double p,double q,double r,double s)
{
    double res = Math.pow((p-r),2)+Math.pow((q-s),2);
    return Math.sqrt(res);
}
public static void main(String args[]) throws FileNotFoundException,
IOException
{
    BufferedReader csv = new BufferedReader(new FileReader(new
File("data.csv")));
    String data = csv.readLine();
    ArrayList<record> al = new ArrayList<>();
    while(data != null)
    {
        String[] dataarray = data.split(",");
        record rc = new record();
        rc.attr1 = Double.parseDouble(dataarray[0]);
        rc.attr2 = Double.parseDouble(dataarray[1]);
        al.add(rc);
        data = csv.readLine();
    }
    ArrayList<record> centroid = new ArrayList<>();
    centroid.add(al.get(0));
    centroid.add(al.get(3));
    centroid.add(al.get(6));
    ArrayList<record> c1 = new ArrayList<>();
    ArrayList<record> c2 = new ArrayList<>();
    ArrayList<record> c3 = new ArrayList<>();
    for(int i=0;i<100;i++)
    {
        c1.clear();c2.clear();c3.clear();
        for(record r:al)
        {
            ArrayList<Double> t = new ArrayList<>();
            double p =
dist(r.attr1,r.attr2,(centroid.get(0)).attr1,(centroid.get(0)).attr2
);

```

```

        double q =
dist(r.attr1,r.attr2,(centroid.get(1)).attr1,(centroid.get(1)).attr2
);
        double s =
dist(r.attr1,r.attr2,(centroid.get(2)).attr1,(centroid.get(2)).attr2
);
        t.add(p);
        t.add(q);
        t.add(s);
        double res = Collections.min(t);
        if(res == p){c1.add(r);}
        if(res == q){c2.add(r);}
        if(res == s){c3.add(r);}
    }
    centroid.clear();
    centroid.add(calc(c1));
    centroid.add(calc(c2));
    centroid.add(calc(c3));
}
System.out.println("-----0th cluster-----");
for(record r:c1)
    System.out.println(r.attr1+" "+r.attr2);
System.out.println("-----1th cluster-----");
for(record r:c2)
    System.out.println(r.attr1+" "+r.attr2);
System.out.println("-----2th cluster-----");
for(record r:c3)
    System.out.println(r.attr1+" "+r.attr2);
System.out.println("-----centroid-----");
for(int i=0;i<3;i++)

    System.out.println((centroid.get(i)).attr1+","+centroid.get(i
)).attr2);
}
}

```

Dataset:

2	10
2	5
8	4
5	8
7	5
6	4
1	2
4	9
3	9

```

10) import java.util.*;
import java.io.*;

class prog10
{
public static void main(String args[])
{
float[][] mat = new float[2][2];

```



```

int n=0;
float res;
Scanner in = new Scanner(System.in);
System.out.println("enter confusion matrix");
for(int i=0;i<2;i++)
{
    for(int j=0;j<2;j++)
    {
        int x = in.nextInt();
        mat[i][j]=x;
        n=n+x;
    }
}

res = mat[0][0]*100/(mat[0][0]+mat[0][1]);
System.out.println("Sensitivity :"+res+"%");
res = mat[1][1]*100/(mat[1][0]+mat[1][1]);
System.out.println("Specificity :"+res+"%");
res = mat[0][0]*100/(mat[0][0]+mat[0][1]);
System.out.println("recall :"+res+"%");
res = mat[0][0]*100/(mat[0][0]+mat[1][0]);
System.out.println("Precision :"+res+"%");
res = (mat[0][0]+mat[1][1])*100/n;
System.out.println("Weighted Accuracy :"+res+"%");
}
}

```

R Programs

4)

```

library(rpart) #install.packages("rpart")
library(datasets)
library(e1071) #install.packages("e1071")
library(rpart.plot) #install.packages("rpart.plot")
str(iris)
ind <- sample(2, nrow(iris),replace = TRUE, prob = c(0.5,0.5))
trainData <- iris[ind ==1,]
testData <- iris[ind ==2,]
my_plot <- rpart( Species ~ Sepal.Length + Sepal.Width + Petal.Length +
Petal.Width ,trainData,
                method = "class")
summary(my_plot)
rpart.plot::rpart.plot(my_plot)
res <- predict(my_plot,testData)
table(res,testData$Species)
summary(res)
print(res)
pred <- naiveBayes(Species ~.,data = trainData,laplace = 1)
#mosaicplot(pred)
ans <- predict(pred,testData)
table(ans,testData$Species)
summary(ans)

```

```
5) library(datasets)
car_data = cars
ind_cardata <- sample(1:nrow(cars), 0.8*nrow(cars))
train_cardata <- car_data[ind_cardata,]
test_cardata <- car_data[-ind_cardata,]
head(car_data)
lfit <- lm(dist ~ speed , train_cardata)
test_cardata$dist<-NULL
# predicting for speed =25
df <- data.frame(speed = 25)
lipre <- predict.lm(lfit,df)
print(lipre)
plot(car_data$speed,car_data$dist,xlab = "speed",ylab = "Distance")
abline(lfit)
```

```
6) library(datasets)
library(cluster)
library(factoextra) #install.packages("factoextra")
lindata <- iris
lindata$Species<-NULL
d<- scale(dist(lindata,method = "euclidian"))
kfit <- kmeans(d,3)
hfit <- hkmeans(d,3)
fviz_cluster(hfit, lindata, geom = "point")
fviz_cluster(kfit, lindata, geom = "point")
```
