Experiment-3B: Decision Tree

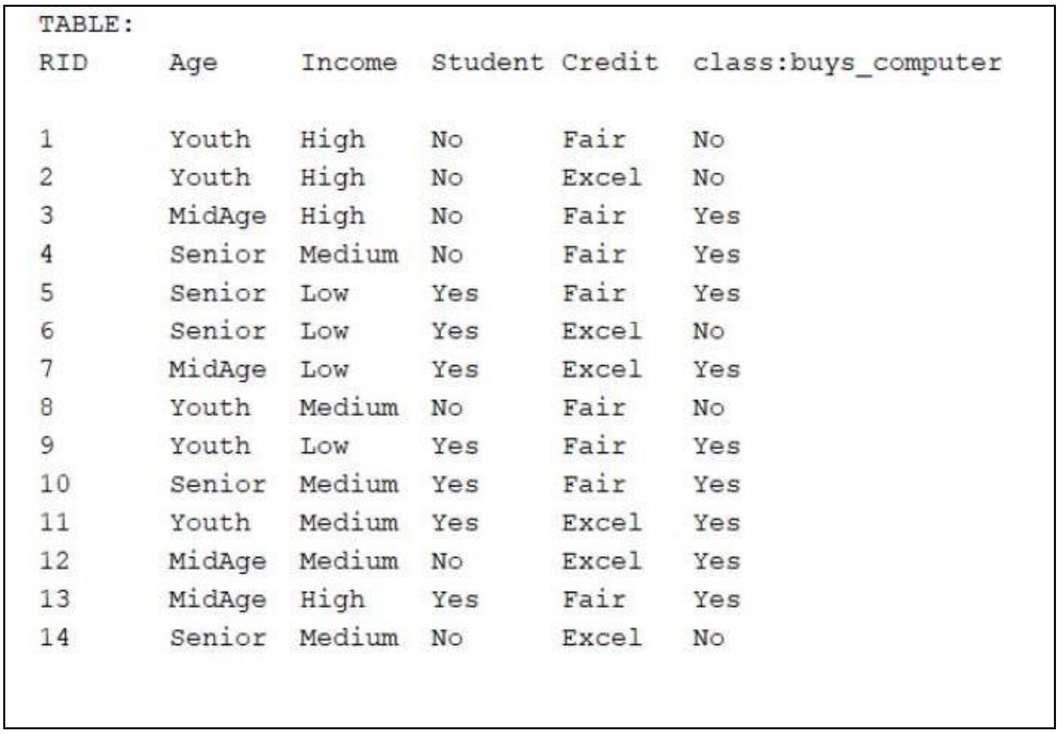
Source Code:

```
import java.util.*; class Atr{

String name;

String val[]; int

count[];

Atr(String name,String val[]){

this.name=name; this.val=val;

count=new int[val.length];

}

} class

DT1{

    String table[][]={

        {"1","Youth","High","No","Fair","No"},

        {"2","Youth","High","No","Excel","No"},

        {"3","MidAge","High","No","Fair","Yes"},

        {"4","Senior","Medium","No","Fair","Yes"},

        {"5","Senior","Low","Yes","Fair","Yes"},

        {"6","Senior","Low","Yes","Excel","No"},

        {"7","MidAge","Low","Yes","Excel","Yes"},

        {"8","Youth","Medium","No","Fair","No"},

        {"9","Youth","Low","Yes","Fair","Yes"},

        {"10","Senior","Medium","Yes","Fair","Yes"},

        {"11","Youth","Medium","Yes","Excel","Yes"},

        {"12","MidAge","Medium","No","Excel","Yes"},

        {"13","MidAge","High","Yes","Fair", "Yes"},
        {"14","Senior","Medium","No","Excel","No"},
    };
Atr arr[];
String s1[]={"Youth","MidAge","Senior"};
String s2[]={"High","Medium","Low"};
String s3[]={"Yes","No"};
String s4[]={"Fair","Excel"};
String s5[]={"Yes","No"}; int
barred[];
DT1(){
    System.out.println("TABLE:");
    System.out.println("RID\tAge\tIncome\tStudent\tCredit\tclass:buys_computer");
    System.out.println(); for(int i=0;i<table.length;i++){
        for(int j=0;j<6;j++)
            System.out.print(table[i][j]+"\t");
        System.out.println();
    }
    System.out.println(); arr=new
    Atr[5]; arr[0]=new
    Atr("Age",s1); arr[1]=new
    Atr("Income",s2); arr[2]=new
    Atr("Student",s3); arr[3]=new
    Atr("Credit_Rating",s4);
    arr[4]=new
    Atr("buys_computer",s5);
    barred=new int[5];
    solve(table); }
void solve(String table[][]){
    int yc=0,nc=0;
    double cI=0;
```

```
 double gain[]=new double[4]; for(int
 i=0;i<table.length;i++)
 if(table[i][5].equals("Yes"))
        yc++;
    else
        nc++;
cI=inf(yc,nc); for(int
i=1;i<5;i++)
    if(barred[i-1]!=1) gain[i-1]=gain(i,cI,table,yc,nc);
double max=0,pos=-1; for(int
i=0;i<gain.length;i++)
    if(gain[i]>max){
        max=gain[i];
        pos=i;
    }                                                           if(pos==-1){
        System.out.println(table[0][5].toUpperCase()); return;
}
barred[(int)pos]=1;
System.out.println();
System.out.println("CHOSEN ATTRIBUTE : "+arr[(int)pos].name);
if(check()==false){ int cc[]=new int[arr[(int)pos].val.length]; for(int
i=0;i<arr[(int)pos].val.length;i++){ for(int j=0;j<table.length;j++)
if(table[j][(int)(pos+1)].equals(arr[(int)pos].val[i])) cc[i]++;
    }
    String tab[][];
    int k=0;
    for(int i=0;i<arr[(int)pos].val.length;i++){
        k=0;
        tab=new String[cc[i]][6]; for(int j=0;j<table.length;j++)
        if(table[j][(int)(pos+1)].equals(arr[(int)pos].val[i]))
        tab[k++]=table[j];
```

```
                System.out.println("FOR ATTRIBUTE AND VALUE: "+arr[(int)pos].name+"
"+arr[(int)pos].val[i]);

                solve(tab);

                System.out.println("BACK TO ATTRIBUTE: "+arr[(int)pos].name);

            }

        } else

        return;

        }

        boolean check(){ for(int

            i=0;i<barred.length;i++)

            if(barred[i]==0)

                return false;

            return true;

        }

        double gain(int attr,double Inf,String table[][],int yc,int nc){

          double gain=0;

          gain=Inf-entropy(attr,table,yc,nc); return

          gain;

        }

        double entropy(int a,String table[][],int yc,int nc){

          double t1[][]=new double[arr[a-1].val.length][3]; for(int

          i=0;i<arr[a-1].val.length;i++){

            for(int j=0;j<table.length;j++)

                if(table[j][5].equals("Yes") && table[j][a].equals(arr[a-1].val[i]))

                    t1[i][0]++;

                      else if(table[j][5].equals("No") && table[j][a].equals(arr[a-1].val[i]))

                    t1[i][1]++;

            t1[i][2]=inf(t1[i][0],t1[i][1]);

          } double e=0; for(int

          i=0;i<t1.length;i++)

            e=e+(((t1[i][0]+t1[i][1])/(yc+nc))*t1[i][2]);
          return
        e; }
        double inf(double p,double n){ if(p==n)
                return 1;
            if(p==0 || n==0)

                return 0;
            double r=0;
            r=-(((p/(p+n))*Math.log((p/(p+n))))+((n/(p+n))*Math.log((n/(p+n)))));
            r=r/Math.log(2); return r;
        }
     }
     public class DecisionTree { public static
        void main(String[] args) {
DT1 obj=new DT1();
        }
}
```

Output:



```
TABLE:
RID     Age     Income  Student Credit  class:buys_computer

1       Youth   High    No      Fair    No
2       Youth   High    No      Excel   No
3       MidAge  High    No      Fair    Yes
4       Senior  Medium  No      Fair    Yes
5       Senior  Low     Yes     Fair    Yes
6       Senior  Low     Yes     Excel   No
7       MidAge  Low     Yes     Excel   Yes
8       Youth   Medium  No      Fair    No
9       Youth   Low     Yes     Fair    Yes
10      Senior  Medium  Yes     Fair    Yes
11      Youth   Medium  Yes     Excel   Yes
12      MidAge  Medium  No      Excel   Yes
13      MidAge  High    Yes     Fair    Yes
14      Senior  Medium  No      Excel   No
```