**Problem Statement:**

**Use decision trees to prepare a model on fraud data**

**treating those who have taxable\_income <= 30000 as "Risky" and others are "Good".**

**Ans:**

|  |
| --- |
| > fraud<- read.csv(file.choose())  > View(fraud)  > attach(fraud)  > str(fraud)  'data.frame': 600 obs. of 6 variables:  $ Undergrad : chr "NO" "YES" "NO" "YES" ...  $ Marital.Status : chr "Single" "Divorced" "Married" "Single" ...  $ Taxable.Income : int 68833 33700 36925 50190 81002 33329 83357 62774 83519 98152 ...  $ City.Population: int 50047 134075 160205 193264 27533 116382 80890 131253 102481 155482 ...  $ Work.Experience: int 10 18 30 15 28 0 8 3 12 4 ...  $ Urban : chr "YES" "YES" "YES" "YES" ...  > summary(fraud)  Undergrad Marital.Status  Length:600 Length:600  Class :character Class :character  Mode :character Mode :character        Taxable.Income City.Population  Min. :10003 Min. : 25779  1st Qu.:32872 1st Qu.: 66967  Median :55074 Median :106494  Mean :55208 Mean :108747  3rd Qu.:78612 3rd Qu.:150114  Max. :99619 Max. :199778  Work.Experience Urban  Min. : 0.00 Length:600  1st Qu.: 8.00 Class :character  Median :15.00 Mode :character  Mean :15.56  3rd Qu.:24.00  Max. :30.00  > colSums(is.na(fraud))  Undergrad Marital.Status Taxable.Income  0 0 0  City.Population Work.Experience Urban  0 0 0  > hist(Taxable.Income)    >  > #converting taxable.income data to categorical type  > tax\_cat <- ifelse(fraud$Taxable.Income<=30000,"risky","good")  > data1 <- data.frame(tax\_cat,fraud[,-3])  > View(data1)  >  > #data partition  > set.seed(100)  > splittax <- createDataPartition(data1$tax\_cat,p=0.75,list = F)  > traintax <- data1[splittax,]  > testtax <- data1[-splittax,]  > str(traintax)  'data.frame': 450 obs. of 6 variables:  $ tax\_cat : chr "good" "good" "good" "good" ...  $ Undergrad : chr "NO" "YES" "NO" "YES" ...  $ Marital.Status : chr "Single" "Divorced" "Married" "Single" ...  $ City.Population: int 50047 134075 160205 193264 27533 116382 80890 102602 94875 86649 ...  $ Work.Experience: int 10 18 30 15 28 0 8 19 6 16 ...  $ Urban : chr "YES" "YES" "YES" "YES" ...  > str(testtax)  'data.frame': 150 obs. of 6 variables:  $ tax\_cat : chr "good" "good" "good" "risky" ...  $ Undergrad : chr "YES" "NO" "YES" "NO" ...  $ Marital.Status : chr "Single" "Single" "Divorced" "Divorced" ...  $ City.Population: int 131253 102481 155482 148033 34551 29106 155342 169128 41863 183767 ...  $ Work.Experience: int 3 12 4 14 29 7 14 15 30 1 ...  $ Urban : chr "YES" "YES" "YES" "YES" ...  > #model building  >  > #model1 using party package  > model1 <- ctree(as.factor(traintax$tax\_cat)~City.Population+Work.Experience,data = traintax)  > pred1<-predict(model1,testtax)  > table(pred1,testtax$tax\_cat)    pred1 good risky  good 119 31  risky 0 0  > mean(pred1==testtax$tax\_cat) #accuracy= 79.33%  [1] 0.7933333  >  > #model2 using c50 package  > model2 <- C5.0(as.factor(tax\_cat)~.,data = traintax,trails=100)  > pred2 <- predict(model2,testtax)  > table(pred2,testtax$tax\_cat)    pred2 good risky  good 119 31  risky 0 0  > mean(pred2==testtax$tax\_cat) #accuracy= 79.33%  [1] 0.7933333  > CrossTable(pred2,testtax$tax\_cat)    Cell Contents  |-------------------------|  | N |  | N / Table Total |  |-------------------------|    Total Observations in Table: 150    | testtax$tax\_cat  pred2 | good | risky | Row Total |  -------------|-----------|-----------|-----------|  good | 119 | 31 | 150 |  | 0.793 | 0.207 | |  -------------|-----------|-----------|-----------|  Column Total | 119 | 31 | 150 |  -------------|-----------|-----------|-----------|    >  > #model3 using rpart package  > model3 <- rpart(tax\_cat~.,data = traintax,method ='class')  > plot(model3)  > text(model3,pretty = 0)  > fancyRpartPlot(model3,cex=0.5,type = 2)    > pred3 <- predict(model3,testtax,type = 'class')  > table(pred3,testtax$tax\_cat)    pred3 good risky  good 115 29  risky 4 2  > mean(pred3==testtax$tax\_cat) #accuracy= 78%  [1] 0.78  >  > #bagging method  > acc4 <- c()  > for(i in 1:100){  + print(i)  + splits <- createDataPartition(data1$tax\_cat,p=0.85,list = F)  + trains <- data1[splits,]  + tests <- data1[-splits,]  + model4 <- rpart(trains[,-6],as.factor(trains$tax\_cat))  + pred4 <- predict(model4,tests,type ='class' )  + a4 <- table(pred4,tests$tax\_cat)  + acc4 <- c(acc4,sum(diag(a4))/sum(a4))  + }  [1] 1  [1] 2  [1] 3  [1] 4  [1] 5  [1] 6  [1] 7  [1] 8  [1] 9  [1] 10  [1] 11  [1] 12  [1] 13  [1] 14  [1] 15  [1] 16  [1] 17  [1] 18  [1] 19  [1] 20  [1] 21  [1] 22  [1] 23  [1] 24  [1] 25  [1] 26  [1] 27  [1] 28  [1] 29  [1] 30  [1] 31  [1] 32  [1] 33  [1] 34  [1] 35  [1] 36  [1] 37  [1] 38  [1] 39  [1] 40  [1] 41  [1] 42  [1] 43  [1] 44  [1] 45  [1] 46  [1] 47  [1] 48  [1] 49  [1] 50  [1] 51  [1] 52  [1] 53  [1] 54  [1] 55  [1] 56  [1] 57  [1] 58  [1] 59  [1] 60  [1] 61  [1] 62  [1] 63  [1] 64  [1] 65  [1] 66  [1] 67  [1] 68  [1] 69  [1] 70  [1] 71  [1] 72  [1] 73  [1] 74  [1] 75  [1] 76  [1] 77  [1] 78  [1] 79  [1] 80  [1] 81  [1] 82  [1] 83  [1] 84  [1] 85  [1] 86  [1] 87  [1] 88  [1] 89  [1] 90  [1] 91  [1] 92  [1] 93  [1] 94  [1] 95  [1] 96  [1] 97  [1] 98  [1] 99  [1] 100  > acc4  [1] 0.7977528 0.7977528 0.7977528 0.7977528  [5] 0.7977528 0.7977528 0.7977528 0.7977528  [9] 0.7977528 0.7977528 0.7977528 0.7977528  [13] 0.7977528 0.7977528 0.7977528 0.7977528  [17] 0.7977528 0.7977528 0.7977528 0.7977528  [21] 0.7977528 0.7977528 0.7977528 0.7977528  [25] 0.7977528 0.7977528 0.7977528 0.7977528  [29] 0.7977528 0.7977528 0.7977528 0.7977528  [33] 0.7977528 0.7977528 0.7977528 0.7977528  [37] 0.7977528 0.7977528 0.7977528 0.7977528  [41] 0.7977528 0.7977528 0.7977528 0.7977528  [45] 0.7977528 0.7977528 0.7977528 0.7977528  [49] 0.7977528 0.7977528 0.7977528 0.7977528  [53] 0.7977528 0.7977528 0.7977528 0.7977528  [57] 0.7977528 0.7977528 0.7977528 0.7977528  [61] 0.7977528 0.7977528 0.7977528 0.7977528  [65] 0.7977528 0.7977528 0.7977528 0.7977528  [69] 0.7977528 0.7977528 0.7977528 0.7977528  [73] 0.7977528 0.7977528 0.7977528 0.7977528  [77] 0.7977528 0.7977528 0.7977528 0.7977528  [81] 0.7977528 0.7977528 0.7977528 0.7977528  [85] 0.7977528 0.7977528 0.7977528 0.7977528  [89] 0.7977528 0.7977528 0.7977528 0.7977528  [93] 0.7977528 0.7977528 0.7977528 0.7977528  [97] 0.7977528 0.7977528 0.7977528 0.7977528  > summary(acc4)  Min. 1st Qu. Median Mean 3rd Qu. Max.  0.7978 0.7978 0.7978 0.7978 0.7978 0.7978 |
|  |
| |  | | --- | | > | |