**CS 5900/STAT 46700 Topics in Data Science Spring 2025**

**Homework 3- Solution**

**Q.N. 1)** The result of 15 students enrolled in data mining course are provided in the table below. It also provides few other categorical variables:

* Course: whether enrolled in other courses (Yes/No)
* Background: whether student is from a Math, computer science (CS) or other background
* Working: whether student working (W) or not working (NW)

The target variable is Result a binary (Pass/Fail) variable and the other variables are predictor variables.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Result | Course | Background | Working |
| 1 | Pass | Yes | Math | NW |
| 2 | Fail | No | Math | W |
| 3 | Fail | Yes | Math | W |
| 4 | Pass | Yes | CS | NW |
| 5 | Fail | No | Other | W |
| 6 | Fail | Yes | Other | W |
| 7 | Pass | Yes | Math | NW |
| 8 | Pass | Yes | CS | NW |
| 9 | Pass | Yes | Math | W |
| 10 | Pass | Yes | CS | W |
| 11 | Pass | Yes | CS | W |
| 12 | Pass | Yes | Math | NW |
| 13 | Fail | Yes | Other | W |
| 14 | Fail | No | Other | NW |
| 15 | Fail | No | Math | W |

1. Calculate the entropy of the Result.
2. Identify the root node of the above data by calculating the information gain.
3. Construct a decision tree for the subject data using R.

*Solution: We can use R code below to calculate the entropy*

> Q1=read.csv("C:\\Users\\aryalg\\course.csv")

> head(Q1,3)

Result Course Background Working

1 Pass Yes Math NW

2 Fail No Math W

3 Fail Yes Math W

> attach(Q1)

> library(DescTools)

> Entropy(table(Result))

[1] 0.9967916

*Therefore, the Entropy measure of the Result is 0.9967916*

*b) In order to determine the root node of the data we calculate the information gain using the R code below. Based on the calculation it has been determined that* ***Background variable is the root node of the tree with highest information gain.***

> Co=xtabs(~Result+Course)

> Co

Course

Result No Yes

Fail 4 3

Pass 0 8

> CourseYes=c(3,8)

> CourseNo=c(4,0)

> IGCo= Entropy(table(Result))-(11/15\*Entropy(CourseYes)+4/15\*Entropy(CourseNo))

> IGCo

[1] 0.3768676

> ####

> Backg=xtabs(~Result+Background)

> Backg

Background

Result CS Math Other

Fail 0 3 4

Pass 4 4 0

> BackgCS=c(0,4)

> BackMath=c(3,4)

> BackOther=c(4,0)

> IGBackg= Entropy(table(Result))-(4/15\*Entropy(BackgCS)+7/15\*Entropy(BackMath)+4/15\*Entropy(BackOther))

> IGBackg

[1] 0.5370185

> #####

> Work=xtabs(~Result+Working)

> Work

Working

Result NW W

Fail 1 6

Pass 5 3

> WorkNW=c(1,5)

> WorkW=c(6,3)

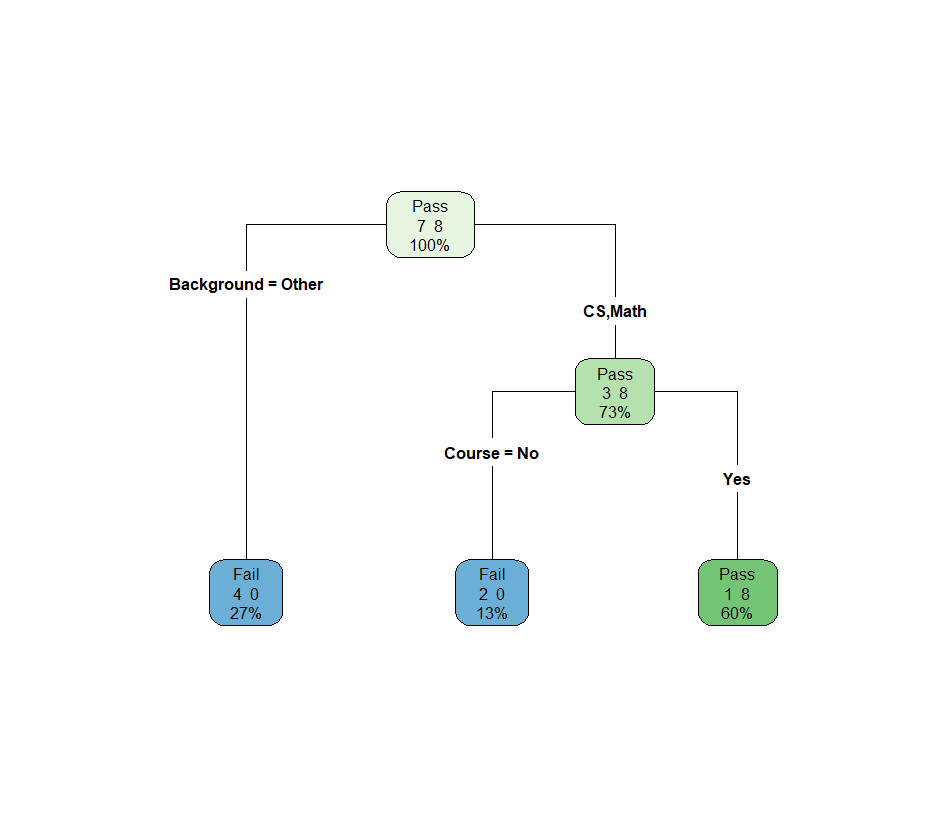
> IGWork= Entropy(table(Result))-(6/15\*Entropy(WorkNW)+9/15\*Entropy(WorkW))

> IGWork

[1] 0.1858052

c.) *We used R code below to create the decision tree*

|  |
| --- |
| > library(rpart)  > library(rpart.plot)  > model=rpart(Result~Background+Course+Working,data=Q1,method="class",  control= rpart.control(minsplit=1,cp=0))  > rpart.plot(model,type=4,extra=101) |
|  |
| |  | | --- | |  | |



***Note that the working variable is not included, however if you would like to add you can do so using R code below (It is not necessary)***

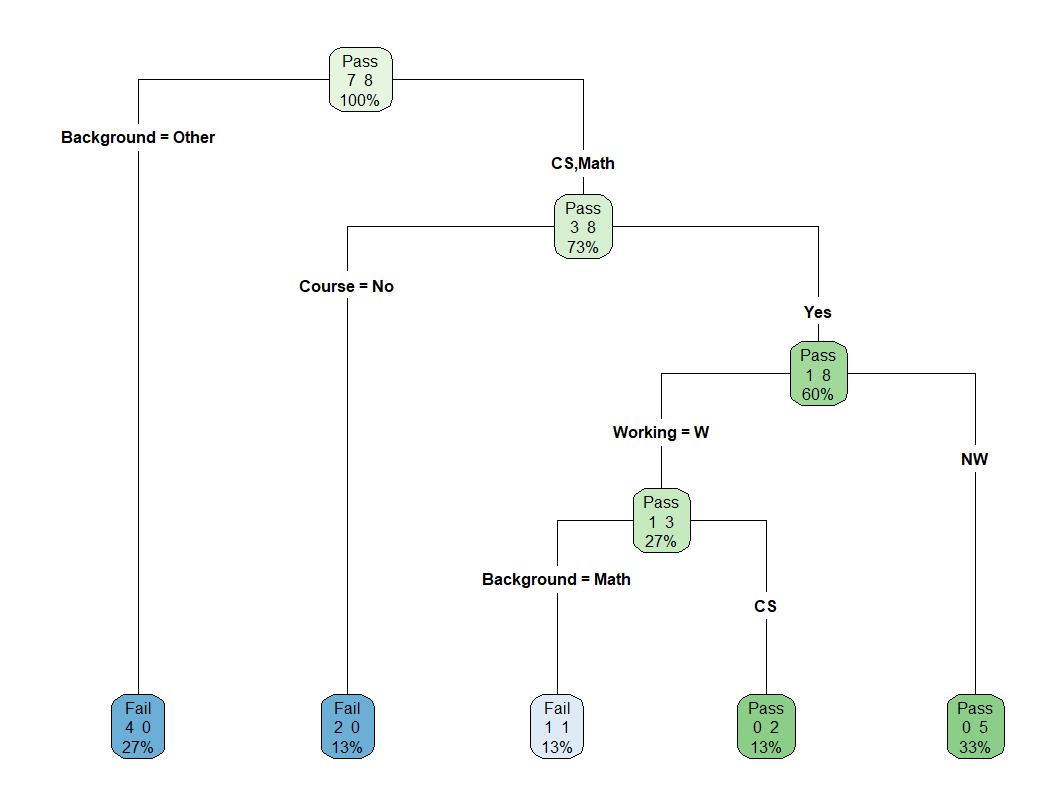
> library(rpart)

> library(rpart.plot)

> model=rpart(Result~Background+Course+Working,data=Q1,method="class",

control= rpart.control(minsplit=1,cp=-1))

> rpart.plot(model,type=4,extra=101)



**Q.N. 2)** Consider the dataset OJ (Orange Juice) available in ISLR package. It describes the purchasing habit of the customer either purchased Citrus Hill or Minute Maid Orange Juice. Several characteristics of the customer and product are recorded.

a) Import the dataset in R and print the variable names.

b) Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.

c) Fit a tree to the training data, with Purchase as the response and the other variables as predictors. Use the summary() function to produce summary statistics about the tree, and describe the results obtained. How many terminal nodes does the tree have?

d) Create a plot of the tree and interpret the results.

e) Predict the response on the test data, and produce a confusion matrix comparing the test labels to the predicted test labels. What is the test error rate?

f) Apply the cv.tree() function to the training set in order to determine the optimal tree size. What is the optimal cp value?

*Solution:*

1. We used R code below to import the data

> library(ISLR)

> data(OJ)

> names(OJ)

[1] "Purchase" "WeekofPurchase" "StoreID" "PriceCH" "PriceMM" "DiscCH"

[7] "DiscMM" "SpecialCH" "SpecialMM" "LoyalCH" "SalePriceMM" "SalePriceCH"

[13] "PriceDiff" "Store7" "PctDiscMM" "PctDiscCH" "ListPriceDiff" "STORE"

1. *We used R code below to create the training data and testing data*

> set.seed(500)

> train\_indices = sample(1:nrow(OJ), 800)

> train= OJ[train\_indices, ]

> test= OJ[-train\_indices, ]

> dim(train)

[1] 800 18

> dim(test)

[1] 270 18

1. *Based on the R code below there are 8 terminal nodes. The tree has misclassified 137 records which is rate of 0.1713, meaning it correctly classifies 82.87% of observations. The residual mean deviance is 0.7553.*

> library(tree)

> attach(OJ)

> treefit=tree(Purchase~.,data=train)

> summary(treefit)

Classification tree:

tree(formula = Purchase ~ ., data = train)

Variables actually used in tree construction:

[1] "LoyalCH" "WeekofPurchase" "PriceDiff"

Number of terminal nodes: 8

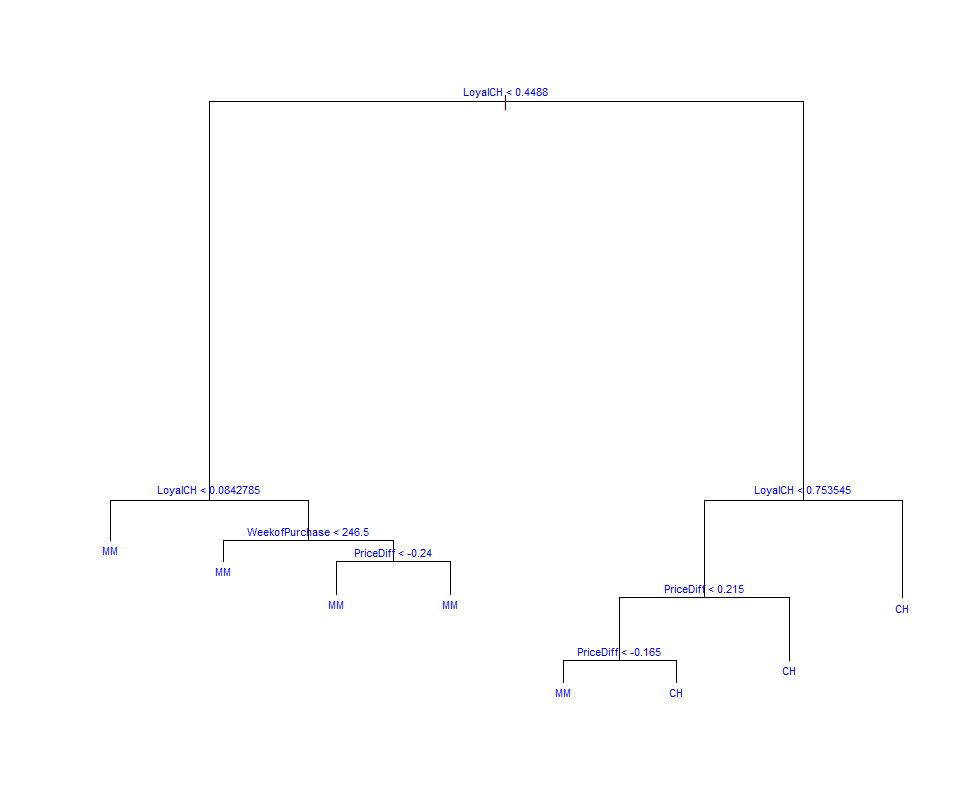
Residual mean deviance: 0.7553 = 598.2 / 792

Misclassification error rate: 0.1713 = 137 / 800

1. We used R code below to create the tree

> plot(treefit)

> text(treefit, col="blue", cex=0.7)



1. *We can use R code below to create the confusion matrix*

> pred=predict(treefit,test,type='class')

> table(pred)

pred

CH MM

180 90

> actual=test$Purchase

> table(actual)

actual

CH MM

182 88

> library(gmodels)

> CrossTable(pred,actual)

Cell Contents

|-------------------------|

| N |

| Chi-square contribution |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 270

| actual

pred | CH | MM | Row Total |

-------------|-----------|-----------|-----------|

CH | 159 | 21 | 180 |

| 11.693 | 24.184 | |

| 0.883 | 0.117 | 0.667 |

| 0.874 | 0.239 | |

| 0.589 | 0.078 | |

-------------|-----------|-----------|-----------|

MM | 23 | 67 | 90 |

| 23.386 | 48.367 | |

| 0.256 | 0.744 | 0.333 |

| 0.126 | 0.761 | |

| 0.085 | 0.248 | |

-------------|-----------|-----------|-----------|

Column Total | 182 | 88 | 270 |

| 0.674 | 0.326 | |

-------------|-----------|-----------|-----------|

*Error rate= (23+21)/270=0.1629.*

1. *Based on the R output below the optimal tree size is 7*

|  |
| --- |
| > cv=cv.tree(treefit)  > plot(cv, type="b")  > cv  $size  [1] 8 7 5 4 3 2 1  $dev  [1] 791.4973 748.9963 759.1515 784.8441 804.2861 843.6098 1088.2811  $k  [1] -Inf 15.94507 19.19996 29.17215 45.12013 70.41870 286.41453 |
|  |
| |  | | --- | |  | |

*In order to determine the optimum cp value we need to use rpart function to create the tree. The optimal cp value is 0.01.*

> treefit=rpart(Purchase~.,data=train)

> printcp(treefit)

Classification tree:

rpart(formula = Purchase ~ ., data = train)

Variables actually used in tree construction:

[1] LoyalCH PctDiscMM PriceDiff SpecialCH WeekofPurchase

Root node error: 329/800 = 0.41125

n= 800

CP nsplit rel error xerror xstd

1 0.516717 0 1.00000 1.00000 0.042303

2 0.021277 1 0.48328 0.51064 0.035016

3 0.015198 4 0.41945 0.50152 0.034785

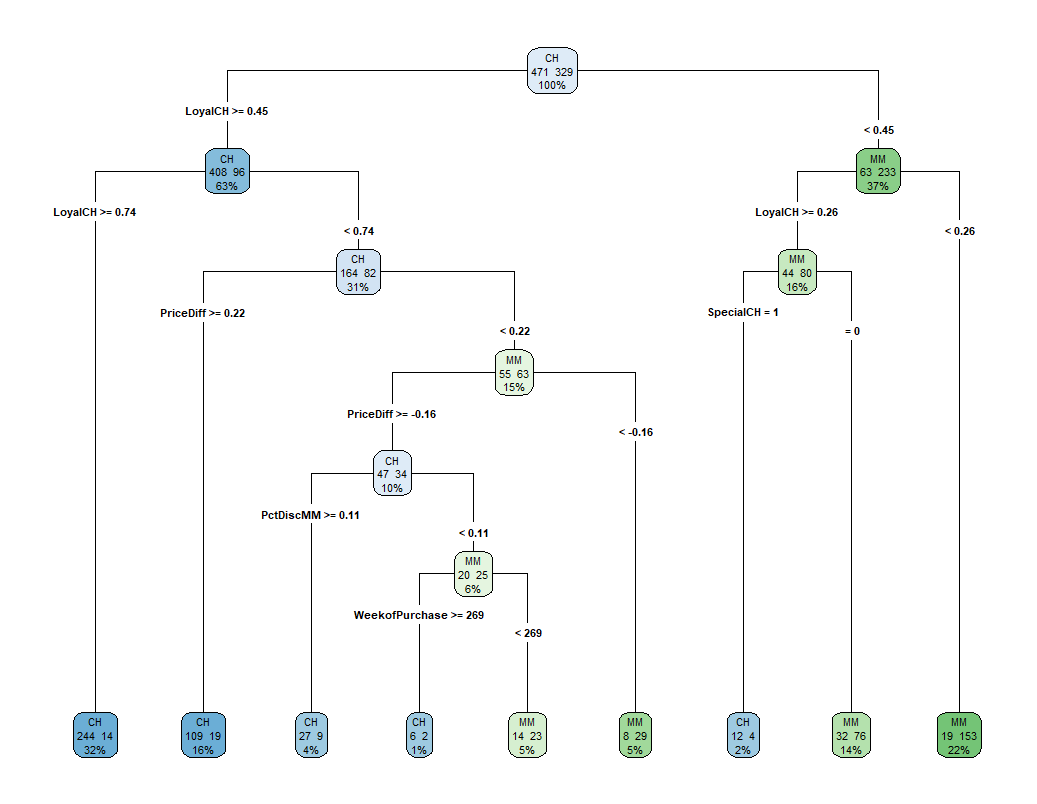
4 0.012158 5 0.40426 0.50456 0.034862

5 0.010000 8 0.36778 0.51672 0.035168

*Using the cp value of 0.01 and rprat plot we have the following decision tree*

> treefit=rpart(Purchase~.,data=train,method="class", cp=0.01)

> rpart.plot(treefit, type=4, extra=101)

**