Classification Process of Decision Tree Analysis on Heart Diseases

Lilian

13th December 2022

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df_heart <- read.csv("C:/Users/LILIAN/Desktop/Decision Tree/heart_disease_uci.csv", stringsAsFactors=T)
```

```
options(knitr.duplicate.label = "allow")
```

```
head(df_heart,3)
```

```
##   id age  sex  dataset             cp trestbps chol   fbs        restecg
## 1  1  63 Male Cleveland typical angina      145  233  TRUE lv hypertrophy
## 2  2  67 Male Cleveland   asymptomatic      160  286 FALSE lv hypertrophy
## 3  3  67 Male Cleveland   asymptomatic      120  229 FALSE lv hypertrophy
##   thalch exang oldpeak       slope ca              thal num
## 1    150 FALSE     2.3 downsloping  0      fixed defect   0
## 2    108  TRUE     1.5        flat  3            normal   2
## 3    129  TRUE     2.6        flat  2 reversable defect   1
```

DATA CLEANING AND WRANGLING

```
df_heart <- subset(df_heart, select = -c(id))
```

```
nrow(df_heart)
```

```
## [1] 920
```

```r
ncol(df_heart)
```

```
## [1] 15
```

```r
df_heart <- df_heart %>%
  rename(chest_pain_type = cp,resting_BP = trestbps,cholesterol = chol,blood_sugar = fbs, cardiographic_
         exercise_induced = exang,depression_induced = oldpeak,
         exercise_slope = slope,nos_major_vessels = ca,thermometer = thal,
         predicted_values = num,location = dataset)
```

I RENAMED THE COLUMNS FOR CLARITY PURPOSE

```r
sum(duplicated(df_heart))
```

```
## [1] 2
```

```r
df_heart %>%
  group_by_all() %>%
  filter(n()>1) %>%
  ungroup()
```

```
## # A tibble: 4 x 15
##     age sex    location  chest~1 resti~2 chole~3 blood~4 cardi~5 max_h~6 exerc~7
##   <int> <fct>  <fct>     <fct>     <int>   <int> <lgl>   <fct>     <int> <lgl>
## 1    49 Female Hungary   atypic~     110      NA FALSE   normal      160 FALSE
## 2    49 Female Hungary   atypic~     110      NA FALSE   normal      160 FALSE
## 3    58 Male   VA Long ~ non-an~     150     219 FALSE   st-t a~     118 TRUE
## 4    58 Male   VA Long ~ non-an~     150     219 FALSE   st-t a~     118 TRUE
## # ... with 5 more variables: depression_induced <dbl>, exercise_slope <fct>,
## #   nos_major_vessels <int>, thermometer <fct>, predicted_values <int>, and
## #   abbreviated variable names 1: chest_pain_type, 2: resting_BP,
## #   3: cholesterol, 4: blood_sugar, 5: cardiographic_results,
## #   6: max_heart_rate, 7: exercise_induced
```

```r
df_heart <- df_heart[ !duplicated(df_heart), ]
```

```r
df_heart <- na.omit(df_heart)
```

‘ LEFT WITH 303 COLUMNS AND 15 ROWS

```r
table(df_heart$sex)
```

```
##
## Female   Male
##     97    206
```

```r
table(df_heart$location)
```

```
##
##     Cleveland       Hungary   Switzerland VA Long Beach
##           299             2             0             2
```

```r
table(df_heart$chest_pain_type)
```

```
##
##    asymptomatic atypical angina     non-anginal  typical angina
##             146              50              84              23
```

```r
table(df_heart$blood_sugar)
```

```
##
## FALSE  TRUE
##   259    44
```

```r
table(df_heart$cardiographic_results)
```

```
##
##                    lv hypertrophy           normal st-t abnormality
##                0              147              151                5
```

```r
table(df_heart$exercise_induced)
```

```
##
## FALSE  TRUE
##   202   101
```

```r
table(df_heart$exercise_slope)
```

```
##
##             downsloping        flat   upsloping
##           2          21         140         140
```

```r
table(df_heart$thermometer)
```

```
##
##                        fixed defect            normal reversable defect
##                 4                18               164               117
```

```r
table(df_heart$predicted_values)
```

```
##
##   0   1   2   3   4
## 163  56  36  35  13
```

OVERVIEW OF THE CATEGORIES OF EACH VARIABLE ARE LISTED BELOW: INTERPRET: SEX: female 97 male 206

LOCATION: Cleveland 299
Hungary 2
Switzerland 0 VA Long Beach 2

CHEST TYPE:asymptomatic 146
atypical angina 50 non-anginal 84
typical angina 23

BLOOD SUGAR:FALSE 259 TRUE 44

CARDIOGRAPHIC RESULTS: lv hypertrophy 147 normal 151 st-t abnormality 5

EXERCISE INDUCED: FALSE 202 TRUE 101

EXERCISE SLOPE: downsloping 21 flat 140 upsloping 140

THERMOMETER: fixed defect 4 normal 18 reversable 164 defect 117

PREDICTED ATTRIBUTES: 0 = 163 1 = 56 2 = 36 3 = 35 4 = 13

```
colnames(df_heart)
```

```
##  [1] "age"                    "sex"                    "location"
##  [4] "chest_pain_type"        "resting_BP"             "cholesterol"
##  [7] "blood_sugar"            "cardiographic_results" "max_heart_rate"
## [10] "exercise_induced"       "depression_induced"     "exercise_slope"
## [13] "nos_major_vessels"      "thermometer"            "predicted_values"
```

```
df_heart$blood_sugar <- as.integer(as.logical(df_heart$blood_sugar))
df_heart$exercise_induced <- as.integer(as.logical(df_heart$exercise_induced))
```

```
str(df_heart)
```

```
## 'data.frame':    303 obs. of  15 variables:
##  $ age                  : int  63 67 67 37 41 56 62 57 63 53 ...
##  $ sex                  : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 1 1 2 2 ...
##  $ location             : Factor w/ 4 levels "Cleveland","Hungary",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ chest_pain_type      : Factor w/ 4 levels "asymptomatic",..: 4 1 1 3 2 2 1 1 1 1 ...
##  $ resting_BP           : int  145 160 120 130 130 120 140 120 130 140 ...
##  $ cholesterol          : int  233 286 229 250 204 236 268 354 254 203 ...
##  $ blood_sugar          : int  1 0 0 0 0 0 0 0 0 1 ...
##  $ cardiographic_results: Factor w/ 4 levels "","lv hypertrophy",..: 2 2 2 3 2 3 2 3 2 2 ...
##  $ max_heart_rate       : int  150 108 129 187 172 178 160 163 147 155 ...
##  $ exercise_induced     : int  0 1 1 0 0 0 0 1 0 1 ...
##  $ depression_induced   : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
##  $ exercise_slope       : Factor w/ 4 levels "","downsloping",..: 2 3 3 2 4 4 2 4 3 2 ...
##  $ nos_major_vessels    : int  0 3 2 0 0 0 2 0 1 0 ...
##  $ thermometer          : Factor w/ 4 levels "","fixed defect",..: 2 3 4 3 3 3 3 3 4 4 ...
##  $ predicted_values     : int  0 2 1 0 0 0 3 0 2 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:615] 167 193 288 303 304 305 306 307 308 309 ...
##   ..- attr(*, "names")= chr [1:615] "167" "193" "288" "303" ...
```

```
glimpse(df_heart)
```

```
## Rows: 303
## Columns: 15
## $ age                   <int> 63, 67, 67, 37, 41, 56, 62, 57, 63, 53, 57, 56, ~
## $ sex                   <fct> Male, Male, Male, Male, Female, Male, Female, Fe~
## $ location              <fct> Cleveland, Cleveland, Cleveland, Cleveland, Clev~
```

```
## $ chest_pain_type       <fct> typical angina, asymptomatic, asymptomatic, non-~
## $ resting_BP            <int> 145, 160, 120, 130, 130, 120, 140, 120, 130, 140~
## $ cholesterol           <int> 233, 286, 229, 250, 204, 236, 268, 354, 254, 203~
## $ blood_sugar           <int> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, ~
## $ cardiographic_results <fct> lv hypertrophy, lv hypertrophy, lv hypertrophy, ~
## $ max_heart_rate        <int> 150, 108, 129, 187, 172, 178, 160, 163, 147, 155~
## $ exercise_induced      <int> 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, ~
## $ depression_induced    <dbl> 2.3, 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 0.6, 1.4, 3.1~
## $ exercise_slope        <fct> downsloping, flat, flat, downsloping, upsloping,~
## $ nos_major_vessels     <int> 0, 3, 2, 0, 0, 0, 2, 0, 1, 0, 0, 0, 1, 0, 0, 0, ~
## $ thermometer           <fct> fixed defect, normal, reversable defect, normal,~
## $ predicted_values      <int> 0, 2, 1, 0, 0, 0, 3, 0, 2, 1, 0, 0, 2, 0, 0, 0, ~
```

```
summary(df_heart)
```

```
##       age            sex              location          chest_pain_type
##  Min.   :29.00   Female: 97   Cleveland    :299   asymptomatic   :146
##  1st Qu.:48.00   Male  :206   Hungary      :  2   atypical angina: 50
##  Median :56.00                Switzerland  :  0   non-anginal    : 84
##  Mean   :54.51                VA Long Beach:  2   typical angina : 23
##  3rd Qu.:61.00
##  Max.   :77.00
##    resting_BP     cholesterol     blood_sugar          cardiographic_results
##  Min.   : 94.0   Min.   :  0.0   Min.   :0.0000                    :  0
##  1st Qu.:120.0   1st Qu.:211.0   1st Qu.:0.0000   lv hypertrophy  :147
##  Median :130.0   Median :240.0   Median :0.0000   normal          :151
##  Mean   :131.7   Mean   :245.5   Mean   :0.1452   st-t abnormality:  5
##  3rd Qu.:140.0   3rd Qu.:275.0   3rd Qu.:0.0000
##  Max.   :200.0   Max.   :564.0   Max.   :1.0000
##  max_heart_rate  exercise_induced depression_induced     exercise_slope
##  Min.   : 71.0   Min.   :0.0000   Min.   :0.000                 :  2
##  1st Qu.:132.0   1st Qu.:0.0000   1st Qu.:0.000     downsloping: 21
##  Median :152.0   Median :0.0000   Median :0.800     flat       :140
##  Mean   :149.2   Mean   :0.3333   Mean   :1.053     upsloping  :140
##  3rd Qu.:165.0   3rd Qu.:1.0000   3rd Qu.:1.600
##  Max.   :202.0   Max.   :1.0000   Max.   :6.200
##  nos_major_vessels            thermometer  predicted_values
##  Min.   :0.0000                     :  4   Min.   :0.0000
##  1st Qu.:0.0000    fixed defect     : 18   1st Qu.:0.0000
##  Median :0.0000    normal           :164   Median :0.0000
##  Mean   :0.6634    reversable defect:117   Mean   :0.9406
##  3rd Qu.:1.0000                            3rd Qu.:2.0000
##  Max.   :3.0000                            Max.   :4.0000
```

THE TABLE ABOVE SHOWS THE TEST STATISTICS OF THE VARIABLES:

The youngest age under review is 28 years while the oldest is 77 years.

The total number of female is 194 while male 726

There are 4 locations(Cleveland, Hungary, Switzerland,Long Beach) under survey with sample size of 304, 293, 123, 200 respectively.

There are 4 types of chest pain that people suffer from, the data shows that 496 people suffer from asymptomatic chest pain,174 from atypical angina,204 from non-anginal and 46 from typical angina.

The average mean of Blood pressure under survey is 132.1 while the maximum is 200.

cholesterol records the average mean of 199.1 while the maximum cholesterol recorded is 603.0

Average blood sugar level is 0.1663 while the maximum is 1.0000.

```
# THIS IS BECAUSE WHEN CARRYING OUT THE ANALYSIS,WE ARE ABLE TO GET EXACTLY SAME SAMPLE # IN THE TRAINI
set.seed(555)
ind_hd <- sample(2,
                nrow(df_heart),
                replace = TRUE,
                  prob = c(0.8, 0.2))
dfheart_train <- df_heart[ind_hd==1, ]
dfheart_test <- df_heart[ind_hd==2, ]
```

```
names(dfheart_train)
```

```
##  [1] "age"                    "sex"                    "location"
##  [4] "chest_pain_type"        "resting_BP"             "cholesterol"
##  [7] "blood_sugar"            "cardiographic_results" "max_heart_rate"
## [10] "exercise_induced"       "depression_induced"     "exercise_slope"
## [13] "nos_major_vessels"      "thermometer"            "predicted_values"
```

```
print(dim(dfheart_train))
```

```
## [1] 241  15
```

```
print(dim(dfheart_test))
```

```
## [1] 62 15
```

THE TRAIN SET HAS 241 ROWS AND TEST 62 ROWS

```
#(chest_pain_type,resting_BP,cholesterol,maximum_heart_rate,depression_induced)
library(party)
```

```
## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich
```

```
library(rpart)
library(rpart.plot)
```

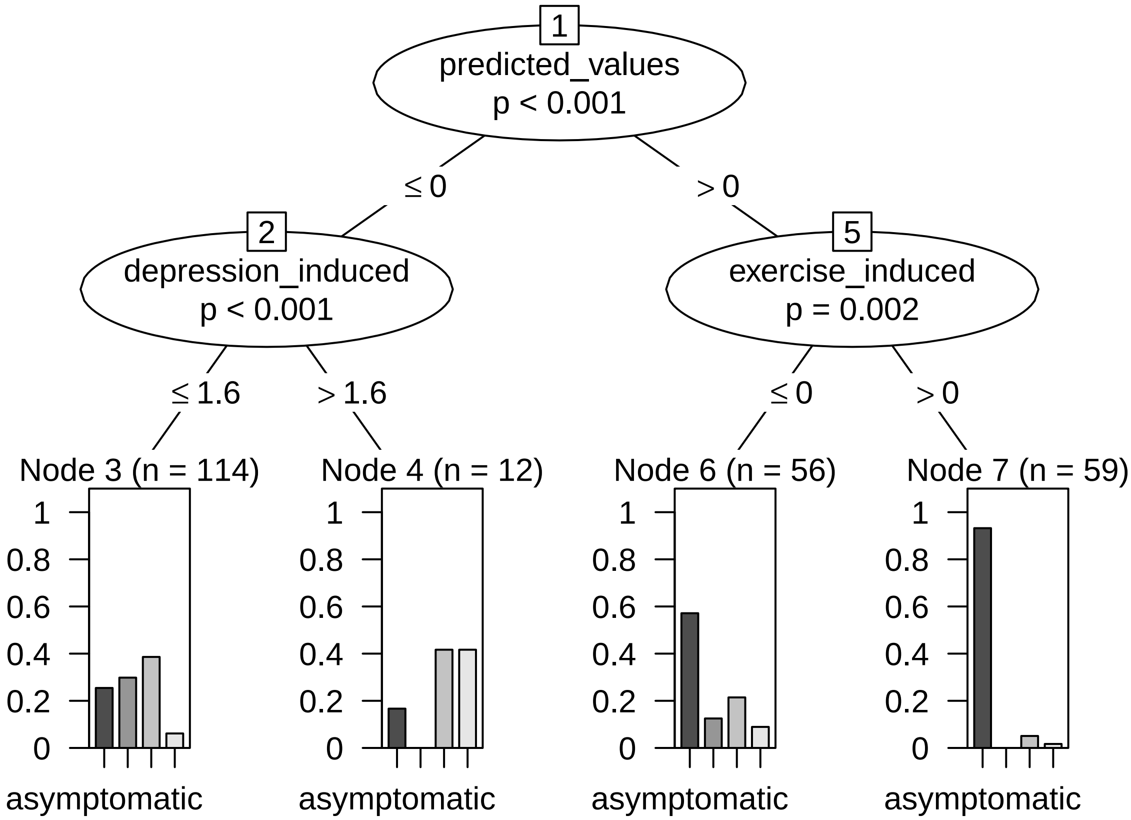
CLASSIFICATION DECISION TREE ANALYSIS

```
dfheart_tree <- ctree(chest_pain_type~., dfheart_train)
print(dfheart_tree)
```

```
##
##   Conditional inference tree with 4 terminal nodes
##
## Response:  chest_pain_type
## Inputs:  age, sex, location, resting_BP, cholesterol, blood_sugar, cardiographic_results, max_heart_r
## Number of observations:  241
##
## 1) predicted_values <= 0; criterion = 1, statistic = 55.564
##   2) depression_induced <= 1.6; criterion = 0.999, statistic = 23.246
##     3)*  weights = 114
##   2) depression_induced > 1.6
##     4)*  weights = 12
## 1) predicted_values > 0
##   5) exercise_induced <= 0; criterion = 0.998, statistic = 20.9
##     6)*  weights = 56
##   5) exercise_induced > 0
##     7)*  weights = 59
```

The output above shows the conditional inference tree with 5 terminal nodes, predicted values will be the top most node

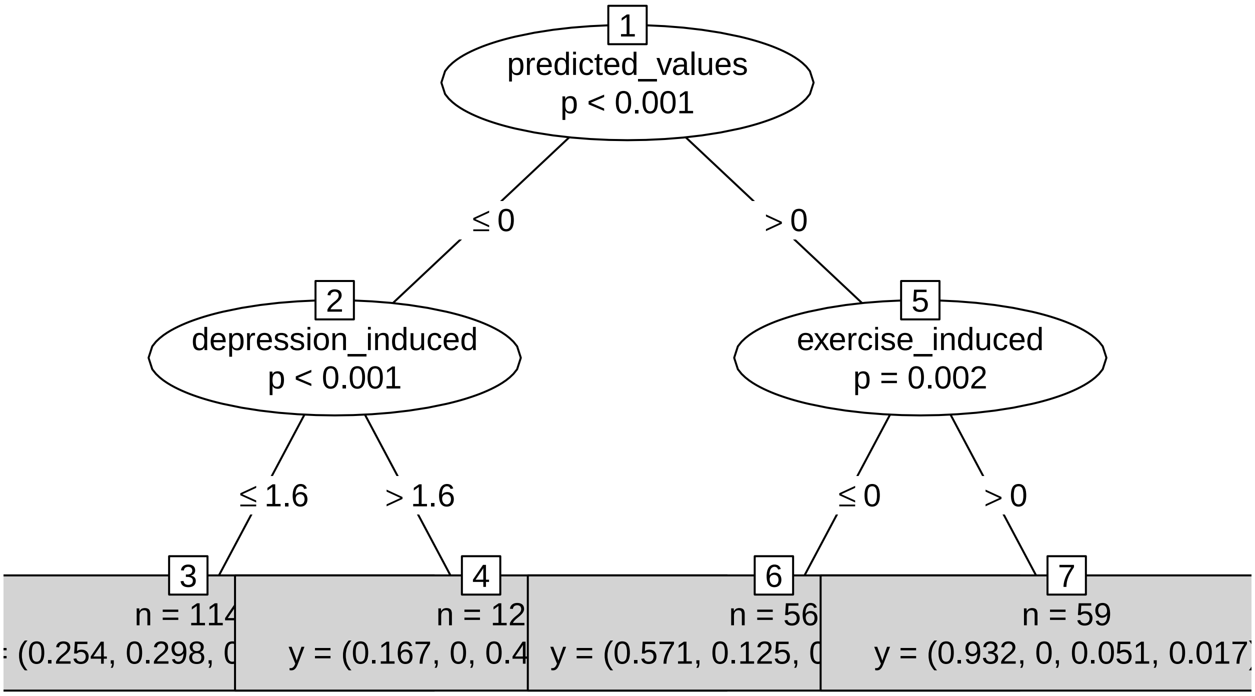
```
plot(dfheart_tree)
```



The root node which is the most important variable is at the top(predicted values), the nodes at the bottom are called terminal nodes, which helps us to take decisions based on the data and model. When Predicted values is < 0.001: then depression induced is the heart disease a patient suffers, if the depression induced is < 0.5,then the probability that any of the heart people will suffer any of the chest pain is high except asympomatic chest pain. If > 0.5, then. . . ..

When the predicted value is > 0.001, then the heart disease is exercise induced,if $P = 0.002$, it is unsloping, flat or downsloping,

```
plot(dfheart_tree,type = 'simple')
```



THIS IS PLOTTING ONLY THE NUMERIC VALUABLES FOR CLEARER VIEW

```
#to get the probability value
Predict(dfheart_tree, dfheart_train,type = 'prob')
```

```
## [[1]]
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[2]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[3]]
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[4]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[5]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[6]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[7]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[8]]
```

```
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[9]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[10]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[11]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[12]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[13]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[14]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[15]]
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[16]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[17]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[18]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[19]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[20]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[21]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[22]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[23]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[24]]
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[25]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[26]]
```

```
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[27]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[28]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[29]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[30]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[31]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[32]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[33]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[34]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[35]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[36]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[37]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[38]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[39]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[40]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[41]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[42]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[43]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[44]]
```

```
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[45]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[46]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[47]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[48]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[49]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[50]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[51]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[52]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[53]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[54]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[55]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[56]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[57]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[58]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[59]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[60]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[61]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[62]]
```

```
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[63]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[64]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[65]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[66]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[67]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[68]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[69]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[70]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[71]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[72]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[73]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[74]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[75]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[76]]
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[77]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[78]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[79]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[80]]
```

```
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[81]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[82]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[83]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[84]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[85]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[86]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[87]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[88]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[89]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[90]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[91]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[92]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[93]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[94]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[95]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[96]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[97]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[98]]
```

```
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[99]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[100]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[101]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[102]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[103]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[104]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[105]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[106]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[107]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[108]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[109]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[110]]
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[111]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[112]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[113]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[114]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[115]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[116]]
```

```
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[117]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[118]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[119]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[120]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[121]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[122]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[123]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[124]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[125]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[126]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[127]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[128]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[129]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[130]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[131]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[132]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[133]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[134]]
```

```
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[135]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[136]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[137]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[138]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[139]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[140]]
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[141]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[142]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[143]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[144]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[145]]
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[146]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[147]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[148]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[149]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[150]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[151]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[152]]
```

```
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[153]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[154]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[155]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[156]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[157]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[158]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[159]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[160]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[161]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[162]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[163]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[164]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[165]]
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[166]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[167]]
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[168]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[169]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[170]]
```

```
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[171]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[172]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[173]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[174]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[175]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[176]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[177]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[178]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[179]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[180]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[181]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[182]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[183]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[184]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[185]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[186]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[187]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[188]]
```

```
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[189]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[190]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[191]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[192]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[193]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[194]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[195]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[196]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[197]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[198]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[199]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[200]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[201]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[202]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[203]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[204]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[205]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[206]]
```

```
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[207]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[208]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[209]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[210]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[211]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[212]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[213]]
## [1] 0.1666667 0.0000000 0.4166667 0.4166667
##
## [[214]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[215]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[216]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[217]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[218]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[219]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[220]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[221]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[222]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[223]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[224]]
```

```
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[225]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[226]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[227]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[228]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[229]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[230]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
##
## [[231]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[232]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[233]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[234]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[235]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[236]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[237]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[238]]
## [1] 0.57142857 0.12500000 0.21428571 0.08928571
##
## [[239]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[240]]
## [1] 0.93220339 0.00000000 0.05084746 0.01694915
##
## [[241]]
## [1] 0.25438596 0.29824561 0.38596491 0.06140351
```

THE TABLE SHOW THE PROBABILITY THAT THE EACH LINE BELONG TO THE CHEST PAIN

TYPES RESPECTIVELY.BECAUSE THE asymptomatic HAS THE HIGHEST PROBABILITY IN ALL THE OBSERVATIONS.

```
p1 <- predict(dfheart_tree, dfheart_train)

#to store it in tables
table1 <- table(Predicted = p1, Actual = dfheart_train$chest_pain_type)

table1
```

```
##                  Actual
## Predicted         asymptomatic atypical angina non-anginal typical angina
##   asymptomatic              87               7          15              6
##   atypical angina            0               0           0              0
##   non-anginal               31              34          49             12
##   typical angina             0               0           0              0
```

THE CONFUSION MATRIX PRINTED ABOVE, WE SEE THAT THERE ARE 302 DATA POINT BELONGING TO ASYMPOMATIC, 103 BELONGS T0 ATYPICAL ANGINA,NON ANGINAL 30, TYPICAL ANGINA 0. THERE ARE 20 MISCLASSIFICATION ERROR BELONGING TO ATYPICAL ANGINA BUT PREDICTED TO BELONG TO ASYMPTOMATIC,60 IN non-anginal BUT PTREDICTED TO BELONG TO ASYMPTOMATIC,16 IN typical angina BUT PTREDICTED TO BELONG TO ASYMPTOMATIC.

```
1 - sum(diag(table1))/sum(table1)
```

```
## [1] 0.4356846
```

THE MISCLASSIFICATION ERROR IS ABOUT 4.3%, WHICH MEANS THAT ACCURACY LEVEL IS 95.7%.

```
p2 <- predict(dfheart_tree, dfheart_test)

table2 <- table(Predicted = p2, Actual = dfheart_test$chest_pain_type)
table2
```

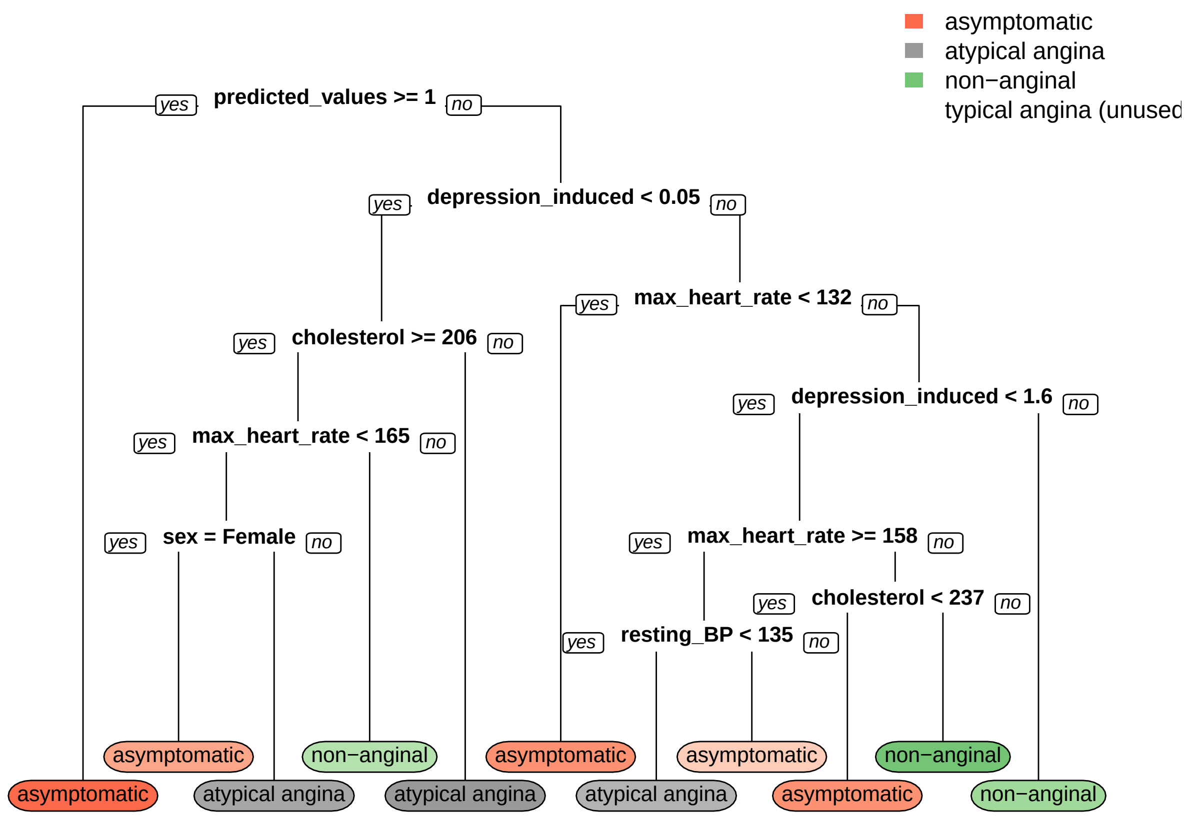
```
##                  Actual
## Predicted         asymptomatic atypical angina non-anginal typical angina
##   asymptomatic              19               2           3              1
##   atypical angina            0               0           0              0
##   non-anginal                9               7          17              4
##   typical angina             0               0           0              0
```

```
1 - sum(diag(table2))/sum(table2)
```

```
## [1] 0.4193548
```

THE MISCLASSIFICATION ERROR FOR TEST DATA IS 3.6%, WHICH MEANS THAT ACCURACY LEVEL IS 96.4%.

```
#TO GET A CLEARER PLOT FROM RPART  PACKAGE
dfheart_tree <- rpart(formula = chest_pain_type~.,
                 data = dfheart_train,
                 method = "class")

rpart.plot(x = dfheart_tree, yesno = 2,type =0, extra =0)
```



INTERPRETING THE ABOVE TREE: WHEN PREDICTED VALUE IS GEATER OR EUALS TO 1,IT IS DEPRESSION INDUCED AND IF THE PROBABILITY OF THE PREDICTED VALUES IS LESS THAN 1, THE CHEST PAIN TYPE IS ASYMPOMATIC.

AT THE SECOND STAGE NODE,THE PROBAILITY THAT DEPRESSION INDUCED ID $<$ 0.05 IS CHOLESTEROL BEING GREATER THAN OR EUAL TO 206 WHICH RESULTS TO ATYPICAL ANGINA CHEST PAIN.THE PROBABILITY OF CHOLESTEROL $<$ 206, MAKES THE MAXIMUM HEART RATE $<$ 165 WHICH IS IS THE PROBABILITY OF THE CHEST PAIN TYPE BEING NON_ANGINA. IF THE MAXIMUM HEAR RATE IS $<$ 165, THEN THE GENDER IS LIKELY TO BE FEMALE AND IF FEMALE, IT IS ASYMPTOMATIC CHEST PAIN TYPE, IF NOT FEMALE, IT IS ATYPICAL ANGINA.

```
# WE TRY TO COMPARE THE MODEL PERFOMANCE ON THE TEST SET AFTER TRAINING WITH DIFFERENT CRITERIA. THE 2

#model training based on gini-based splitting criteria

dfheart_tree1 <- rpart(formula = chest_pain_type~.,
                       data = dfheart_train,
                       method = "class",
                       parms = list(split = "gini"))
```

```r
#model training based on information gain-based splitting criteria

dfheart_tree2 <- rpart(formula = chest_pain_type~.,
                   data = dfheart_train,
                   method = "class",
                   parms = list(split = "information"))
```

```r
# Generate class predictions on the test data using gini-based splitting criteria

pred_dfheart_tree1 <- predict(object = dfheart_tree1,
              newdata = dfheart_test,
                  type = "class")
```

MODEL EVALUATION ON TEST DATA

```r
# Generate class predictions on the test data using information-based splitting criteria

pred_dfheart_tree2 <- predict(object = dfheart_tree2,
              newdata = dfheart_test,
              type = "class")
```

```r
library(Metrics)
```

PREDICTION ACCURACY COMPARISON

```r
#compare classification accuracy on test data
accuracy(actual = dfheart_test$chest_pain_type,
         predicted = pred_dfheart_tree1)
```

```
## [1] 0.5967742
```

```r
accuracy(actual = dfheart_test$chest_pain_type,
         predicted = pred_dfheart_tree2)
```

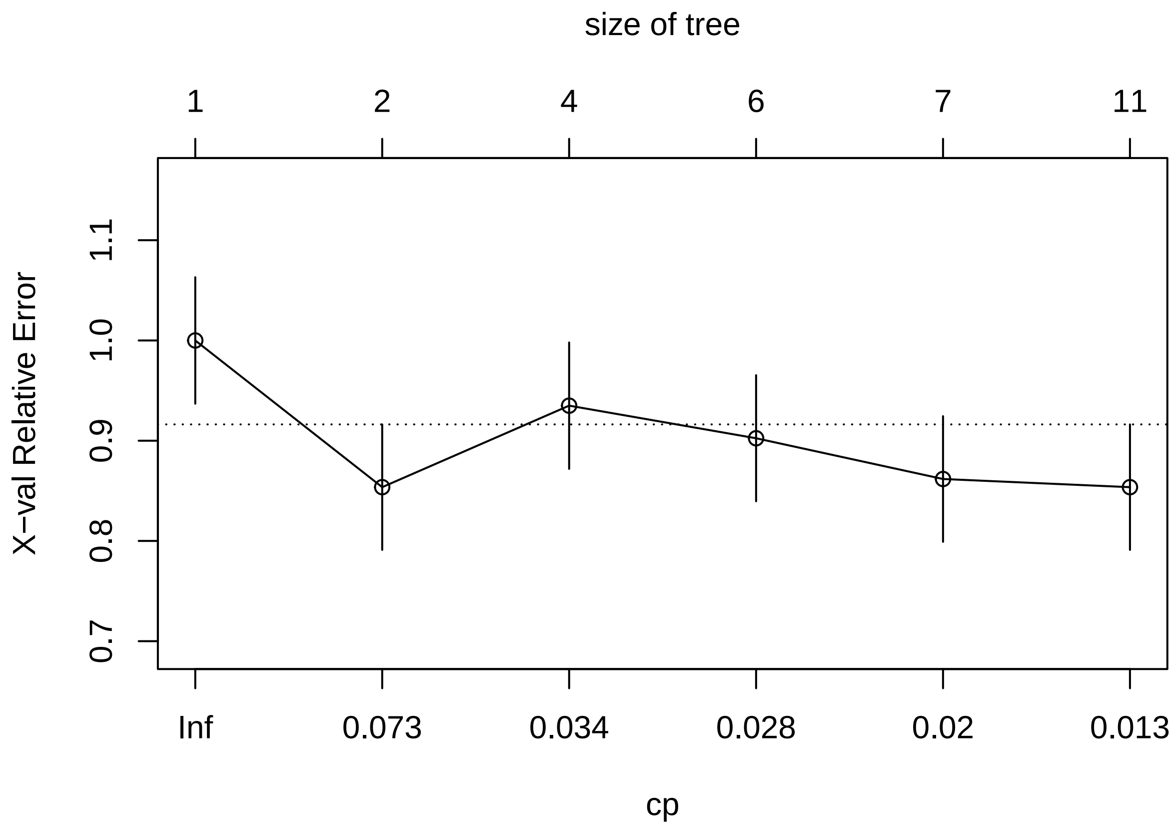
```
## [1] 0.5806452
```

THE ABOVE OUTPUT SHOWS THAT THE BEST SPLITTING MODEL ACCURACY IS gini based splitting criteria more than the information gain splitting.

```r
#because of the complicated nature of the tree, sometimes it runs into overfitting, you #will not be ab
#sometimes you need to prune the tree to simplify it
#to maximize the accuracy and minimize the error
```

DECISION TREE PRUNNING

```r
#plotting complex parameter(CP) table
plotcp(dfheart_tree1)
```



WITH THE USE OF LIBRARY CALLED "plotcp", COMPLEX PARAMETER (CP) CONTROLS THE SIZE OF THE DECISION TREE. IF THE COST OF ADDING ANOTHER VARIABLE TO THE DECISION TREE FROM THE CURRENT NODE IS ABOVE THE VALUE OF CP, THEN TREE BUILDING DOES NOT CONTINUE.

```
print(dfheart_tree1$cptable)
```

```
##           CP nsplit rel error    xerror       xstd
## 1 0.14634146      0 1.0000000 1.0000000 0.06309282
## 2 0.03658537      1 0.8536585 0.8536585 0.06258215
## 3 0.03252033      3 0.7804878 0.9349593 0.06304063
## 4 0.02439024      5 0.7154472 0.9024390 0.06290998
## 5 0.01626016      6 0.6910569 0.8617886 0.06264785
## 6 0.01000000     10 0.6260163 0.8536585 0.06258215
```

the above table shows that xerror is minimum with CP value of 0.

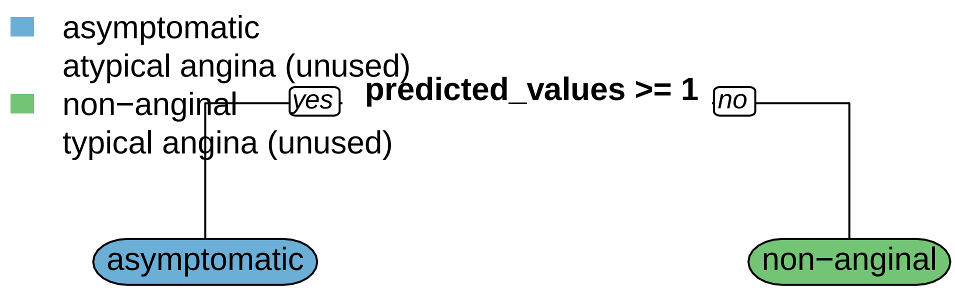
```
#HERE WE FILTER OUT THE OPTIMAL CP VALUE BY IDENTIFYING THE INDEX OF MINIMUM XERROR AND #BY SUPPLYING I

# Retrieve of optimal cp value based on cross_validated error
index <- which.min(dfheart_tree1$cptable[, "xerror"])
```

```
cp_optimal <- dfheart_tree1$cptable[index, "CP"]
```

USING PRUNE FUNCTION BY SUPPLYING OPTIMAL CP VALUE

```r
#pruning tree based on optimal cp value
dfheart_tree1_opt <- prune(tree = dfheart_tree1, cp = cp_optimal)
```

```r
#plot the optimized model
rpart.plot(x = dfheart_tree1_opt, yesno = 2, type = 0, extra = 0)
```



TO CHECK THE PRUNED TREE PERFORMANCE

```r
pred_dfheart_tree3 <- predict(object = dfheart_tree1_opt,
               newdata = dfheart_test,
               type = "class")
```

```r
#check the accuracy level
accuracy(actual = dfheart_test$chest_pain_type,
         predicted = pred_dfheart_tree3)
```

```
## [1] 0.5806452
```