Classification Process of Decision Tree Analysis on Heart Diseases

Lilian

13th December 2022

```
#LIBRARY
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
```

```
#import data
df_heart <- read.csv("C:/Users/LILIAN/Desktop/Data Science COVERED/Decision Tree/heart_disease_uci.csv"
```

```
#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

```
#remove column not needed
df_heart <- subset(df_heart, select = -c(id))
```

```
nrow(df_heart)
```

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

```r
ncol(df_heart)
```

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

```r
#RENAME COLUMNS
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
#TOTAL duplicated
sum(duplicated(df_heart))
```

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

```r
#see the duplicate rows in the dataframe
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
#remove duplicate rows
df_heart <- df_heart[ !duplicated(df_heart), ]
```

```r
# dealing with missing values
#df_heart <- na.omit(df_heart)
```

‘

```r
#view the category of some columns
table(df_heart$sex)
```

```
##
## Female   Male
##    193    725
```

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

```
##
##     Cleveland       Hungary   Switzerland VA Long Beach
##           304           292           123           199
```

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

```
##
##    asymptomatic atypical angina     non-anginal  typical angina
##             496             173             203              46
```

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

```
##
## FALSE  TRUE
##   690   138
```

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

```
##
##                    lv hypertrophy           normal st-t abnormality
##                2              188              550              178
```

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

```
##
## FALSE  TRUE
##   527   336
```

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

```
##
##             downsloping        flat   upsloping
##         307          63         345         203
```

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

```
##
##                        fixed defect            normal reversable defect
##               484                46               196               192
```

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

```
##
##   0   1   2   3   4
## 410 265 108 107  28
```

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

```
# column names
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"
```

```
#change the categorical variables to muneric
#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':    918 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          : logi  TRUE FALSE FALSE FALSE FALSE FALSE ...
##  $ 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     : logi  FALSE TRUE TRUE FALSE FALSE FALSE ...
##  $ 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 ...
```

```r
# preview the data summary
glimpse(df_heart)
```

```
## Rows: 918
## 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            <lgl> TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ 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       <lgl> FALSE, TRUE, TRUE, FALSE, FALSE, FALSE, FALSE, T~
## $ 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, ~
```

```r
# summary statistics
summary(df_heart)
```

```
##       age            sex               location          chest_pain_type
##  Min.   :28.00   Female:193   Cleveland    :304   asymptomatic   :496
##  1st Qu.:47.00   Male  :725   Hungary      :292   atypical angina:173
##  Median :54.00                Switzerland  :123   non-anginal    :203
##  Mean   :53.51                VA Long Beach:199   typical angina : 46
##  3rd Qu.:60.00
##  Max.   :77.00
##
##    resting_BP     cholesterol    blood_sugar          cardiographic_results
##  Min.   :  0.0   Min.   :  0.0   Mode :logical                    :  2
##  1st Qu.:120.0   1st Qu.:175.0   FALSE:690       lv hypertrophy  :188
##  Median :130.0   Median :223.0   TRUE :138       normal          :550
##  Mean   :132.1   Mean   :199.1   NA's :90        st-t abnormality:178
##  3rd Qu.:140.0   3rd Qu.:268.0
##  Max.   :200.0   Max.   :603.0
##  NA's   :59      NA's   :29
##  max_heart_rate  exercise_induced depression_induced     exercise_slope
##  Min.   : 60.0   Mode :logical    Min.   :-2.6000               :307
##  1st Qu.:120.0   FALSE:527        1st Qu.: 0.0000    downsloping: 63
##  Median :140.0   TRUE :336        Median : 0.5000    flat       :345
##  Mean   :137.5   NA's :55         Mean   : 0.8808    upsloping  :203
##  3rd Qu.:157.0                    3rd Qu.: 1.5000
##  Max.   :202.0                    Max.   : 6.2000
##  NA's   :55                       NA's   :62
##  nos_major_vessels            thermometer  predicted_values
##  Min.   :0.0000                     :484   Min.   :0.0000
##  1st Qu.:0.0000    fixed defect     : 46   1st Qu.:0.0000
##  Median :0.0000    normal           :196   Median :1.0000
##  Mean   :0.6764    reversable defect:192   Mean   :0.9956
```

```
##  3rd Qu.:1.0000                          3rd Qu.:2.0000
##  Max.   :3.0000                          Max.   :4.0000
##  NA's   :609
```

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.

```
#  DATA PARTITIONING USING 555
# 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 of the training columns
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"
```

```
#dimensions of the train and test data
print(dim(dfheart_train))
```

```
## [1] 727  15
```

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

```
## [1] 191  15
```

THE TRAIN SET HAS 727 ROWS AND TEST 191 ROWS

```
#DECISION TREE
#(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)
```

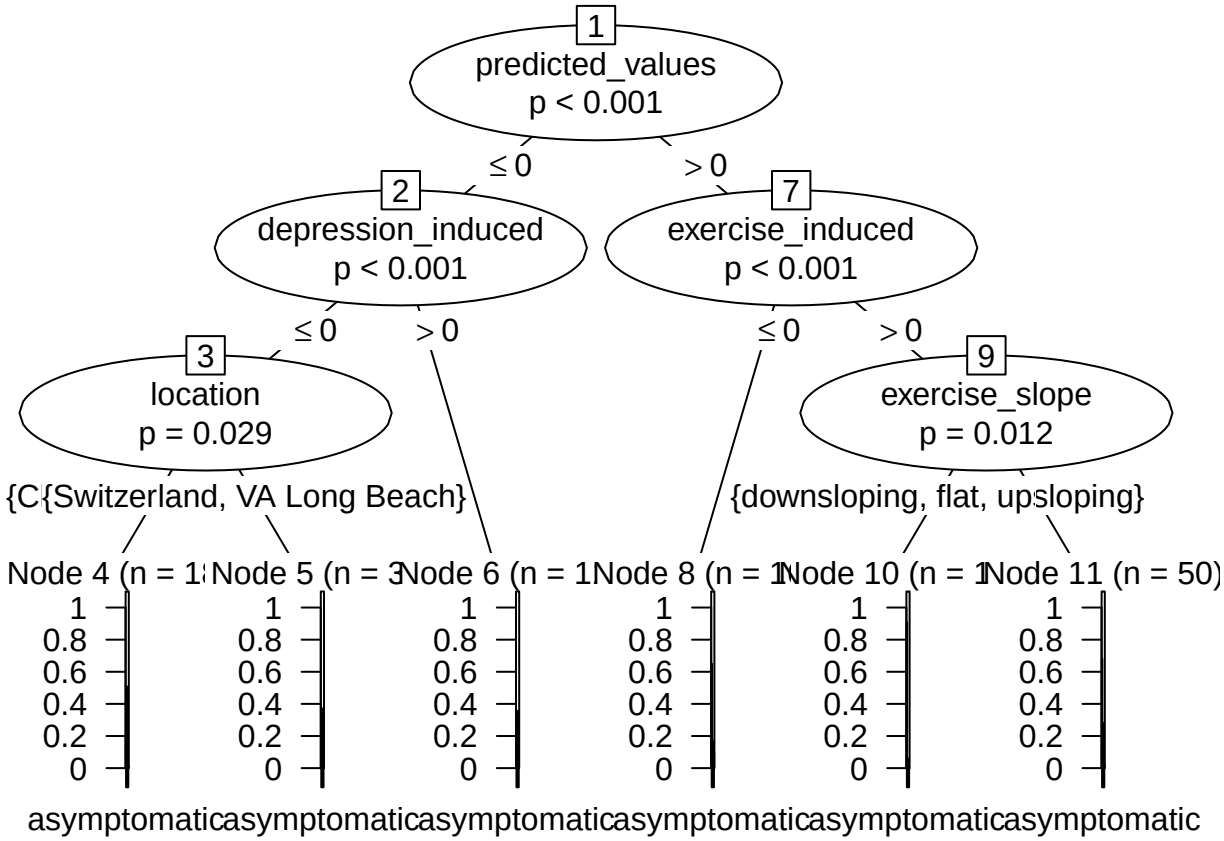
CLASSIFICATION DECISION TREE ANALYSIS

```
# select the independent variable
dfheart_tree <- ctree(chest_pain_type~., dfheart_train)
print(dfheart_tree)
```

```
##
##   Conditional inference tree with 6 terminal nodes
##
## Response:  chest_pain_type
## Inputs:  age, sex, location, resting_BP, cholesterol, blood_sugar, cardiographic_results, max_heart_
## Number of observations:  727
##
## 1) predicted_values <= 0; criterion = 1, statistic = 154.315
##   2) depression_induced <= 0; criterion = 1, statistic = 37.546
##     3) location == {Cleveland, Hungary}; criterion = 0.971, statistic = 25.928
##       4)*  weights = 185
##     3) location == {Switzerland, VA Long Beach}
##       5)*  weights = 35
##   2) depression_induced > 0
##     6)*  weights = 112
## 1) predicted_values > 0
##   7) exercise_induced <= 0; criterion = 1, statistic = 34.803
##     8)*  weights = 160
##   7) exercise_induced > 0
##     9) exercise_slope == {downsloping, flat, upsloping}; criterion = 0.988, statistic = 28.158
##       10)*  weights = 185
##     9) exercise_slope == {}
##       11)*  weights = 50
```

The output above shows the conditional inference tree with 4 terminal nodes, predicted values will be the top most node

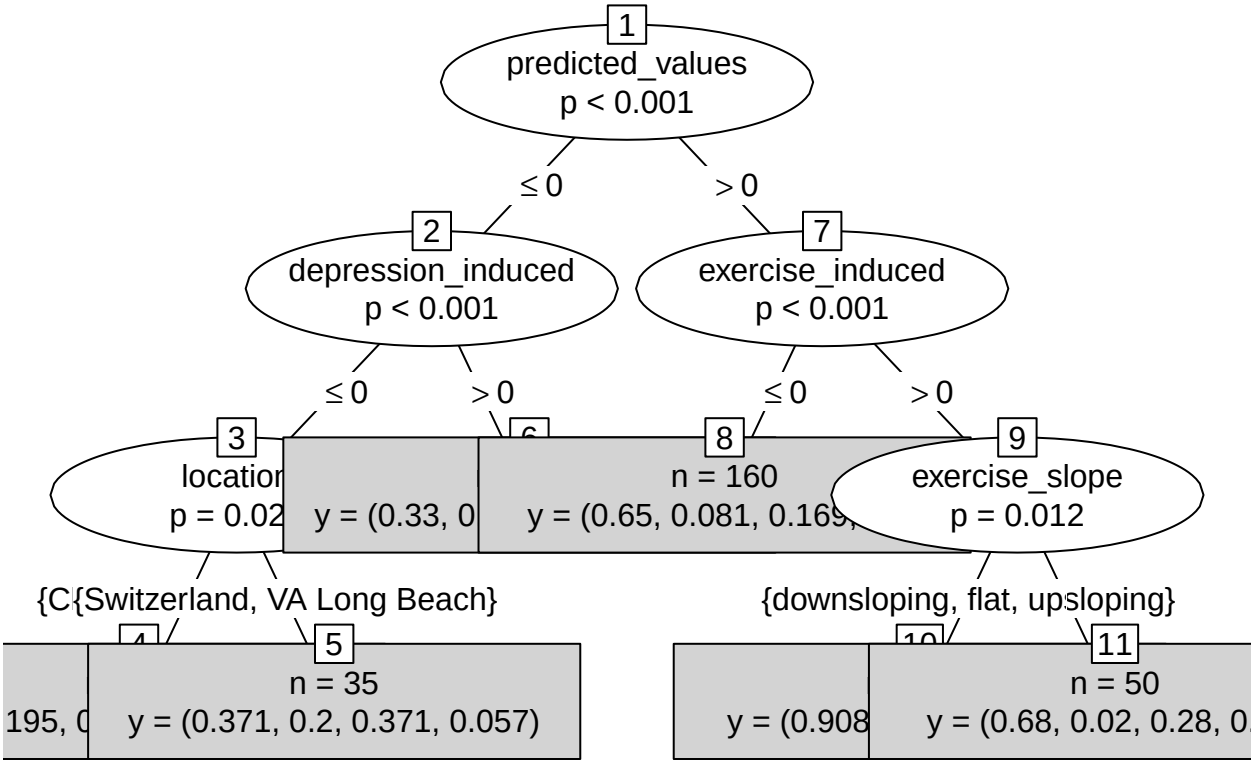
```
#VISUALIZAYION OF DECISION TREE
plot(dfheart_tree)
```



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

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

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.

```
#CONFUSION MATRIX - TRAIN DATA
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             319              27          65              19
##   atypical angina           36              94          52               3
##   non-anginal               37              21          40              14
##   typical angina             0               0           0               0
```

THE CONFUSION MATRIX PRINTED ABOVE, WE SEE THAT THERE ARE 319 DATA POINT BELONGING TO ASYMPOMATIC, 94 BELONGS T0 ATYPICAL ANGINA,NON ANGINAL 40, TYPICAL

ANGINA 0. THERE ARE 27 MISCLASSIFICATION ERROR BELONGING TO ATYPICAL ANGINA BUT PREDICTED TO BELONG TO ASYMPTOMATIC,65 IN non-anginal BUT PTREDICTED TO BELONG TO ASYMPTOMATIC,19 IN typical angina BUT PTREDICTED TO BELONG TO ASYMPTOMATIC.

```
#calcualte misclassification error ABOVE
1 - sum(diag(table1))/sum(table1)
```

```
## [1] 0.3768913
```

THE MISCLASSIFICATION ERROR IS 37.7%, WHICH MEANS THAT ACCURACY LEVEL IS 62.3%.

```
#CONFUSION MATRIX ON TEST DATA
p2 <- predict(dfheart_tree, dfheart_test)

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

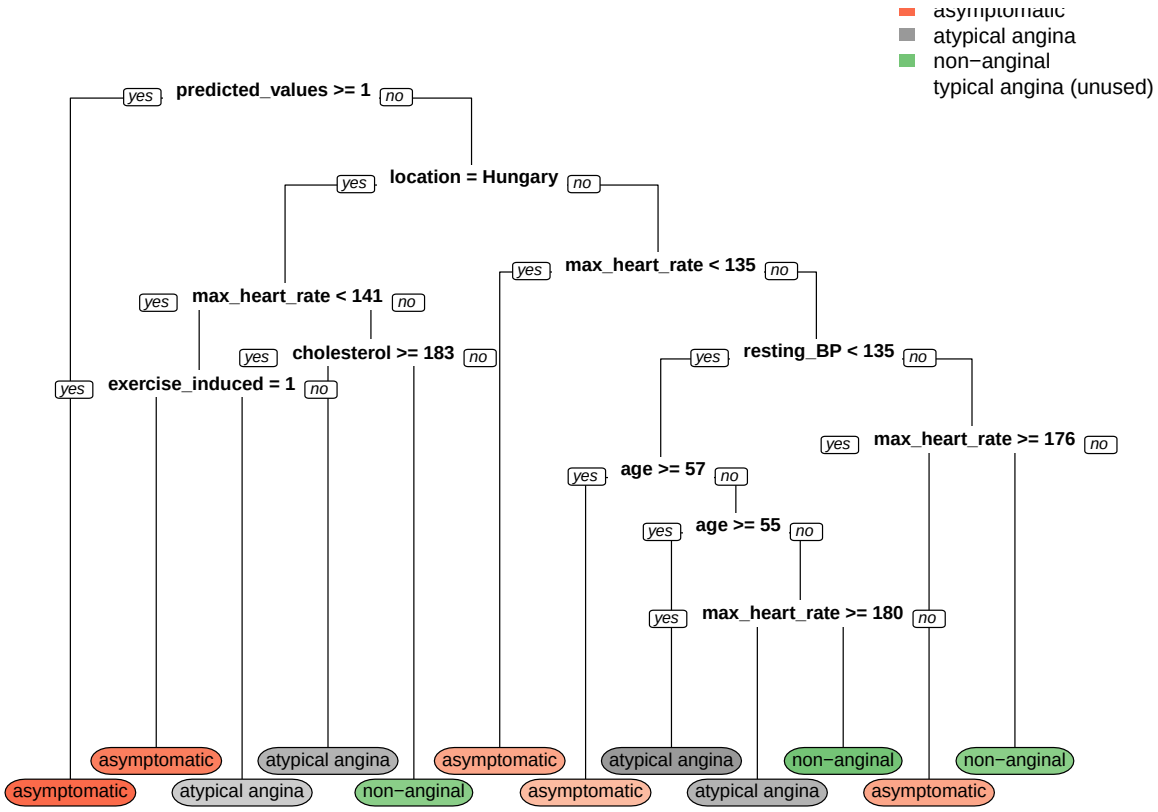
```
##                  Actual
## Predicted         asymptomatic atypical angina non-anginal typical angina
##   asymptomatic              86               5          22              4
##   atypical angina            8              18          14              4
##   non-anginal               10               8          10              2
##   typical angina             0               0           0              0
```

```
#MISCLASSIFICATION ERROR - TEST DATA
1 - sum(diag(table2))/sum(table2)
```

```
## [1] 0.4031414
```

THE MISCLASSIFICATION ERROR FOR TEST DATA IS 40.3%, WHICH MEANS THAT ACCURACY LEVEL IS 59.7%.

```
#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.

```
# TREE SPLITTING CRITERIA BASED COMPARISON to see if there is any prediction accuracy
# 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"))
```

```
#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"))
```

```
# MODEL EVALUATION ON TEST DATA
# 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

```
# HERE WE PREDICT THE CLASS LABELS OF THE TEST DATASET
# Generate class predictions on the test data using information-based splitting criteria

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

```
library(Metrics)
```

PREDICTION ACCURACY COMPARISON

```
#WE COMPARE THE ACCURACY OF THE MODELS
#compare classification accuracy on test data
accuracy(actual = dfheart_test$chest_pain_type,
         predicted = pred_dfheart_tree1)
```

```
## [1] 0.6335079
```

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

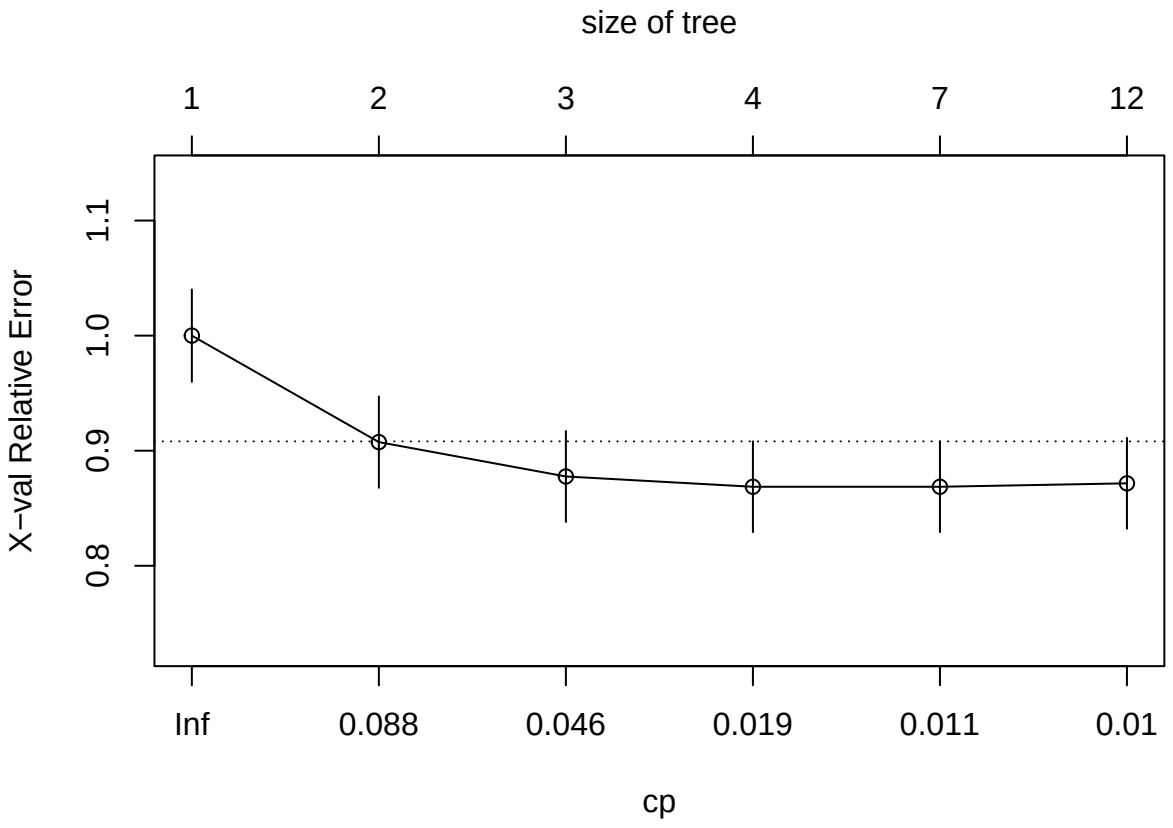
```
## [1] 0.6544503
```

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

```
#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

```
# PLOTTING THE ERROR VS COMPLEXITY PARAMETER
#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.

```
# GENERATING COMPLEXITY PARAMETER TABLE WITH "model$cptable"
print(dfheart_tree1$cptable)
```

```
##           CP nsplit rel error    xerror       xstd
## 1 0.10746269      0 1.0000000 1.0000000 0.04011931
## 2 0.07164179      1 0.8925373 0.9074627 0.03970041
## 3 0.02985075      2 0.8208955 0.8776119 0.03950077
## 4 0.01194030      3 0.7910448 0.8686567 0.03943463
## 5 0.01074627      6 0.7552239 0.8686567 0.03943463
## 6 0.01000000     11 0.7014925 0.8716418 0.03945700
```

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

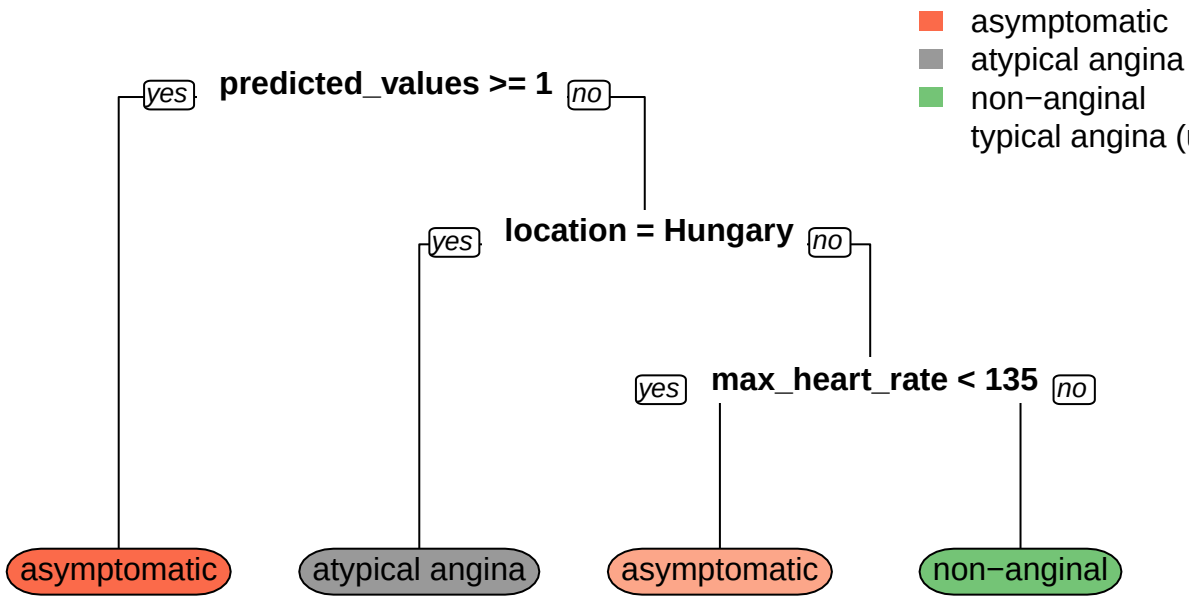
```
# OBTAINING AN OPTIMAL PRUNED MODEL
#HERE WE FILTER OUT THE OPTIMAL CP VALUE BY IDENTIFYING THE INDEX OF MINIMUM XERROR AND #BY SUPPLYING I

# 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

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

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



TO CHECK THE PRUNED TREE PERFORMANCE

```
# TO CHECK IF THE PRUNED TREE HAS SIMILAR PERFORMANCE WITH THE MAIN TREE
pred_dfheart_tree3 <- predict(object = dfheart_tree1_opt,
                newdata = dfheart_test,
                type = "class")
```

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

```
## [1] 0.6492147
```