ML Project

rohan

2022-12-18

install.packages("party")

```
library(party)
```

```
## Warning: package 'party' was built under R version 4.2.2
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Warning: package 'strucchange' was built under R version 4.2.2
```

```
## Loading required package: zoo
```

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

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

```
## Loading required package: sandwich
```

```
## Warning: package 'sandwich' was built under R version 4.2.2
```

```
cardio_data<-read.csv("C:/Users/adari/Downloads/Cardio.csv", header= TRUE)

#Inspect the dataset
names(cardio_data)
```

```
##  [1] "BPM"   "APC"   "FMPS"  "UCPS"  "DLPS"  "SDPS"  "PDPS"  "ASTV"  "MSTV"
## [10] "ALTV"  "MLTV"  "Width" "Min"   "Max"   "NSP"
```

```
head(cardio_data)
```

```
##   BPM         APC FMPS        UCPS        DLPS SDPS        PDPS ASTV MSTV ALTV
## 1 120 0.000000000    0 0.000000000 0.000000000    0 0.000000000   73  0.5   43
## 2 132 0.006379585    0 0.006379585 0.003189793    0 0.000000000   17  2.1    0
## 3 133 0.003322259    0 0.008305648 0.003322259    0 0.000000000   16  2.1    0
## 4 134 0.002560819    0 0.007682458 0.002560819    0 0.000000000   16  2.4    0
## 5 132 0.006514658    0 0.008143322 0.000000000    0 0.000000000   16  2.4    0
## 6 134 0.001049318    0 0.010493179 0.009443861    0 0.002098636   26  5.9    0
##   MLTV Width Min Max NSP
## 1  2.4    64  62 126   2
## 2 10.4   130  68 198   1
## 3 13.4   130  68 198   1
## 4 23.0   117  53 170   1
## 5 19.9   117  53 170   1
## 6  0.0   150  50 200   3
```

```
tail(cardio_data)
```

```
##      BPM         APC        FMPS        UCPS        DLPS SDPS PDPS ASTV MSTV
## 2121 140 0.000000000 0.000000000 0.004975124 0.001243781    0    0   77  0.7
## 2122 140 0.000000000 0.000000000 0.007425743 0.000000000    0    0   79  0.2
## 2123 140 0.000774593 0.000000000 0.006971340 0.000000000    0    0   78  0.4
## 2124 140 0.000980392 0.000000000 0.006862745 0.000000000    0    0   79  0.4
## 2125 140 0.000678887 0.000000000 0.006109980 0.000000000    0    0   78  0.4
## 2126 142 0.001615509 0.001615509 0.008077544 0.000000000    0    0   74  0.4
##      ALTV MLTV Width Min Max NSP
## 2121   17  6.0    31 124 155   1
## 2122   25  7.2    40 137 177   2
## 2123   22  7.1    66 103 169   2
## 2124   20  6.1    67 103 170   2
## 2125   27  7.0    66 103 169   2
## 2126   36  5.0    42 117 159   1
```

```
summary(cardio_data)
```

```
##       BPM              APC                FMPS                UCPS
##  Min.   :106.0   Min.   :0.000000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:126.0   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.001876
##  Median :133.0   Median :0.001630   Median :0.000000   Median :0.004482
##  Mean   :133.3   Mean   :0.003170   Mean   :0.009474   Mean   :0.004357
##  3rd Qu.:140.0   3rd Qu.:0.005631   3rd Qu.:0.002512   3rd Qu.:0.006525
##  Max.   :160.0   Max.   :0.019284   Max.   :0.480634   Max.   :0.014925
##       DLPS               SDPS                PDPS                ASTV
##  Min.   :0.000000   Min.   :0.000e+00   Min.   :0.0000000   Min.   :12.00
##  1st Qu.:0.000000   1st Qu.:0.000e+00   1st Qu.:0.0000000   1st Qu.:32.00
##  Median :0.000000   Median :0.000e+00   Median :0.0000000   Median :49.00
##  Mean   :0.001885   Mean   :3.585e-06   Mean   :0.0001566   Mean   :46.99
##  3rd Qu.:0.003264   3rd Qu.:0.000e+00   3rd Qu.:0.0000000   3rd Qu.:61.00
##  Max.   :0.015385   Max.   :1.353e-03   Max.   :0.0053476   Max.   :87.00
##       MSTV              ALTV              MLTV              Width
##  Min.   :0.200   Min.   : 0.000   Min.   : 0.000   Min.   :  3.00
```

```
##  1st Qu.:0.700   1st Qu.: 0.000   1st Qu.: 4.600   1st Qu.: 37.00
##  Median :1.200   Median : 0.000   Median : 7.400   Median : 67.50
##  Mean   :1.333   Mean   : 9.847   Mean   : 8.188   Mean   : 70.45
##  3rd Qu.:1.700   3rd Qu.:11.000   3rd Qu.:10.800   3rd Qu.:100.00
##  Max.   :7.000   Max.   :91.000   Max.   :50.700   Max.   :180.00
##       Min              Max           NSP
##  Min.   : 50.00   Min.   :122   Min.   :1.000
##  1st Qu.: 67.00   1st Qu.:152   1st Qu.:1.000
##  Median : 93.00   Median :162   Median :1.000
##  Mean   : 93.58   Mean   :164   Mean   :1.304
##  3rd Qu.:120.00   3rd Qu.:174   3rd Qu.:1.000
##  Max.   :159.00   Max.   :238   Max.   :3.000
```

```
str(cardio_data)
```

```
## 'data.frame':    2126 obs. of  15 variables:
##  $ BPM  : int  120 132 133 134 132 134 134 122 122 122 ...
##  $ APC  : num  0 0.00638 0.00332 0.00256 0.00651 ...
##  $ FMPS : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ UCPS : num  0 0.00638 0.00831 0.00768 0.00814 ...
##  $ DLPS : num  0 0.00319 0.00332 0.00256 0 ...
##  $ SDPS : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PDPS : num  0 0 0 0 0 ...
##  $ ASTV : int  73 17 16 16 16 26 29 83 84 86 ...
##  $ MSTV : num  0.5 2.1 2.1 2.4 2.4 5.9 6.3 0.5 0.5 0.3 ...
##  $ ALTV : int  43 0 0 0 0 0 0 6 5 6 ...
##  $ MLTV : num  2.4 10.4 13.4 23 19.9 0 0 15.6 13.6 10.6 ...
##  $ Width: int  64 130 130 117 117 150 150 68 68 68 ...
##  $ Min  : int  62 68 68 53 53 50 50 62 62 62 ...
##  $ Max  : int  126 198 198 170 170 200 200 130 130 130 ...
##  $ NSP  : int  2 1 1 1 1 3 3 3 3 3 ...
```

```
#Check the dimension and number of points of the "cardio_data" dataset
nrow(cardio_data)
```

```
## [1] 2126
```

```
ncol(cardio_data)
```

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

```
dim(cardio_data)
```

```
## [1] 2126   15
```

```
#We need categorical (Factor) data to class variable for prediction, hence we should convert the NSP va
cardio_data$NSPF <-as.factor(cardio_data$NSP)
str(cardio_data)
```

```
## 'data.frame':    2126 obs. of  16 variables:
```

```
##  $ BPM  : int  120 132 133 134 132 134 134 122 122 122 ...
##  $ APC  : num  0 0.00638 0.00332 0.00256 0.00651 ...
##  $ FMPS : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ UCPS : num  0 0.00638 0.00831 0.00768 0.00814 ...
##  $ DLPS : num  0 0.00319 0.00332 0.00256 0 ...
##  $ SDPS : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PDPS : num  0 0 0 0 0 ...
##  $ ASTV : int  73 17 16 16 16 26 29 83 84 86 ...
##  $ MSTV : num  0.5 2.1 2.1 2.4 2.4 5.9 6.3 0.5 0.5 0.3 ...
##  $ ALTV : int  43 0 0 0 0 0 0 6 5 6 ...
##  $ MLTV : num  2.4 10.4 13.4 23 19.9 0 0 15.6 13.6 10.6 ...
##  $ Width: int  64 130 130 117 117 150 150 68 68 68 ...
##  $ Min  : int  62 68 68 53 53 50 50 62 62 62 ...
##  $ Max  : int  126 198 198 170 170 200 200 130 130 130 ...
##  $ NSP  : int  2 1 1 1 1 3 3 3 3 3 ...
##  $ NSPF : Factor w/ 3 levels "1","2","3": 2 1 1 1 1 3 3 3 3 3 ...
```

```
#train and validate(test) data from our data. Divide 80% Training and 20% Validation parts for implemen
set.seed(1234)
pd <-sample(2,nrow(cardio_data),replace=TRUE, prob=c(0.8,0.2))
train <-cardio_data[pd==1,]
validate <-cardio_data[pd==2,]
#Checking how many observations are in train and validate data sets
dim(train)
```

```
## [1] 1718   16
```

```
dim(validate)
```

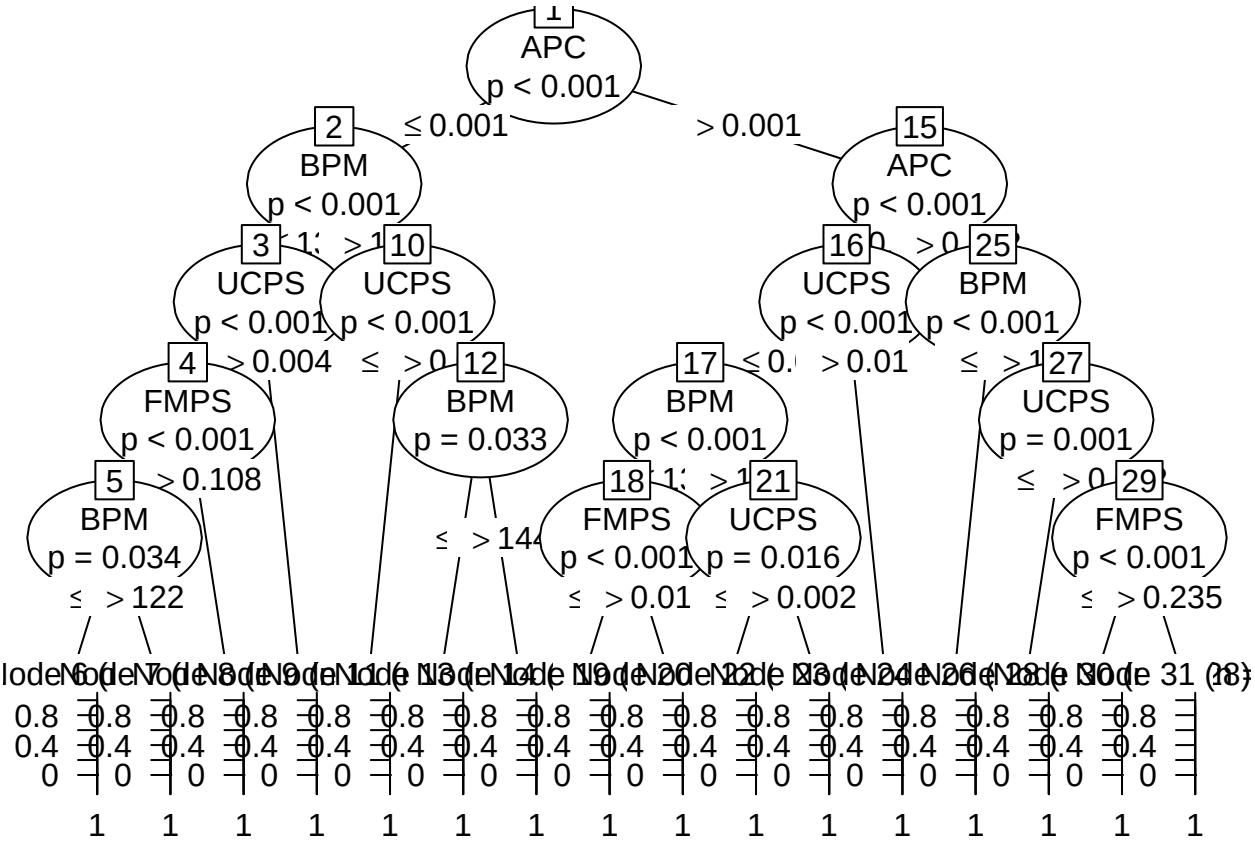
```
## [1] 408  16
```

```
#Now since we have train and validate data sets,we can implement Decision trees. Train the treeusing ct

cardio_tree <-ctree(NSPF~BPM+APC+FMPS+UCPS,data = train)

# Printing and plotting the tree
print(cardio_tree)
```

```
##
##   Conditional inference tree with 16 terminal nodes
##
## Response:  NSPF
## Inputs:  BPM, APC, FMPS, UCPS
## Number of observations:  1718
##
## 1) APC <= 0.000834028; criterion = 1, statistic = 263.403
##   2) BPM <= 136; criterion = 1, statistic = 131.511
##     3) UCPS <= 0.003809524; criterion = 1, statistic = 39.234
##       4) FMPS <= 0.1075897; criterion = 1, statistic = 19.346
##         5) BPM <= 122; criterion = 0.966, statistic = 9.491
##           6)*  weights = 51
##         5) BPM > 122
```

```
##           7)*  weights = 95
##       4) FMPS > 0.1075897
##         8)*  weights = 10
##     3) UCPS > 0.003809524
##       9)*  weights = 260
##   2) BPM > 136
##     10) UCPS <= 0.000834028; criterion = 1, statistic = 31.044
##       11)*  weights = 116
##     10) UCPS > 0.000834028
##       12) BPM <= 144; criterion = 0.967, statistic = 9.592
##         13)*  weights = 115
##       12) BPM > 144
##         14)*  weights = 83
## 1) APC > 0.000834028
##   15) APC <= 0.002209945; criterion = 1, statistic = 52.155
##     16) UCPS <= 0.009578544; criterion = 1, statistic = 27.577
##       17) BPM <= 136; criterion = 0.999, statistic = 16.591
##         18) FMPS <= 0.0121396; criterion = 1, statistic = 39.555
##           19)*  weights = 95
##         18) FMPS > 0.0121396
##           20)*  weights = 7
##       17) BPM > 136
##         21) UCPS <= 0.001838235; criterion = 0.984, statistic = 8.267
##           22)*  weights = 16
##         21) UCPS > 0.001838235
##           23)*  weights = 62
##     16) UCPS > 0.009578544
##       24)*  weights = 8
##   15) APC > 0.002209945
##     25) BPM <= 110; criterion = 1, statistic = 18.889
##       26)*  weights = 18
##     25) BPM > 110
##       27) UCPS <= 0.001668057; criterion = 0.999, statistic = 16.249
##         28)*  weights = 142
##       27) UCPS > 0.001668057
##         29) FMPS <= 0.2354892; criterion = 1, statistic = 19.471
##           30)*  weights = 628
##         29) FMPS > 0.2354892
##           31)*  weights = 12
```

```
plot(cardio_tree)
```




```
# check the prediction on train data
predict(cardio_tree)
```

```
##    [1] 1 1 1 1 3 3 1 1 1 2 2 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##   [38] 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 1
##   [75] 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
##  [112] 2 2 2 2 2 2 2 1 1 1 2 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [149] 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [260] 2 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 1 2 2 2 2 1 1 1 1 1 1 1 1 1
##  [297] 1 1 1 2 1 2 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
##  [334] 2 2 2 2 2 1 1 2 2 2 2 2 1 1 2 2 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1
##  [371] 1 1 1 1 1 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1
##  [408] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 2 2 1 1
##  [445] 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 1
##  [482] 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 2 2 2 1 1 2 2 1 1 1 1 1 1 1 1 1
##  [519] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 3 3 1 1 1 1 1 1 1 1 1 1 3 3 1 3 3 3
##  [556] 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [593] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2
##  [630] 2 2 2 2 2 2 1 2 2 2 1 2 2 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1
##  [667] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [704] 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [741] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [778] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [815] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
##  [852] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [889] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [926] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [963] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1000] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1037] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1074] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1111] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1148] 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 2 1 1 1 1 2 2 1 1 1 2 1 2 1 1 1 1 1 1 1
## [1185] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1222] 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1259] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1296] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3
## [1333] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1370] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1407] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1444] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1481] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1518] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1
## [1555] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1592] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1629] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1666] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1703] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## Levels: 1 2 3
```

```
tab<-table(predict(cardio_tree), train$NSPF)
print(tab)
```

```
##
##        1    2    3
##   1 1291   91  103
##   2   55  135   25
##   3    2    0   16
```

```
# Calculate classification accuracy

sum(diag(tab))/sum(tab)
```

```
## [1] 0.8393481
```

```
# classification error

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

```
## [1] 0.1606519
```

```
# validate the model on test data set

test_predict <-table(predict(cardio_tree, newdata= validate), validate$NSPF)
print(test_predict)
```

```
##
##       1   2   3
##   1 290  30  28
##   2  16  38   3
##   3   1   1   1
```

```
#Calculate classification accuracy

sum(diag(test_predict))/sum(test_predict)
```

```
## [1] 0.8063725
```

```
# classification error

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

```
## [1] 0.1606519
```