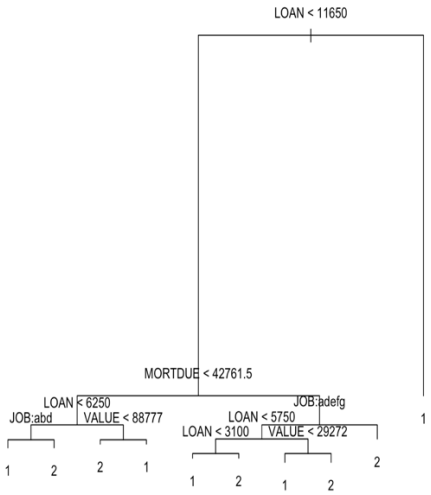
```
> install.packages("reshape")
> library(reshape)
> install.packages("RWeka")
> library(RWeka)
> install.packages("caret")
> library(caret)

> hmeqtrain<-read.csv("hmeqN_train.csv",header = TRUE, fileEncoding = "euc-kr")
> str(hmeqtrain)
'data.frame':	2000 obs. of  7 variables:
 $ ID      : int  4952 5546 938 277 5204 5762 2354 2896 76 4700 ...
 $ BAD     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN    : int  26100 35000 9100 5700 28100 44000 14200 16000 3900 24800 ...
 $ MORTDUE: num  73525 391000 17218 58400 61000 ...
 $ VALUE   : num  89870 505000 36721 75000 99000 ...
 $ REASON : chr  "DebtCon" "DebtCon" "DebtCon" "HomeImp" ...
 $ JOB     : chr  "Office" "ProfExe" "Other" "ProfExe" ...

> head(hmeqtrain,3);tail(hmeqtrain,3)
 ID BAD  LOAN MORTDUE   VALUE   REASON      JOB
1 4952   1 26100   73525   89870 DebtCon   Office
2 5546   1 35000  391000 505000 DebtCon ProfExe
3  938   1  9100   17218   36721 DebtCon    Other
        ID BAD LOAN MORTDUE VALUE   REASON      JOB
1998 936   2 9100   80719 86782 HomeImp    Other
1999 209   2 5100   62702 79784 HomeImp    Other
2000 399   2 6500   80739 97630 DebtCon ProfExe

> hmeqtest<-read.csv("hmeqN_test.csv",header = TRUE, fileEncoding = "euc-kr")
> str(hmeqtest)
'data.frame':	378 obs. of  7 variables:
 $ ID      : int  5632 675 3234 3537 3804 926 462 2229 4770 184 ...
 $ BAD     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN    : int  38700 8000 17300 18600 20000 9000 6900 13700 25000 5000 ...
 $ MORTDUE: num  119847 37871 73000 64248 60336 ...
 $ VALUE   : num  162365 89870 95000 82690 132430 ...
 $ REASON : chr  "HomeImp" "HomeImp" "DebtCon" "DebtCon" ...
 $ JOB     : chr  "ProfExe" "ProfExe" "Other" "Mgr" ...

> hmeqtrain<-as.data.frame(hmeqtrain)
> BAD<-as.factor(hmeqtrain[,2])
> LOAN<-as.numeric(hmeqtrain[,3])
> hmeqtrain<-cbind(BAD,LOAN,hmeqtrain[,-2:-3])
> hmeqtrain$REASON<-as.factor(hmeqtrain$REASON)
> hmeqtrain$JOB<-as.factor(hmeqtrain$JOB)
> str(hmeqtrain)
'data.frame':	2000 obs. of  7 variables:
 $ BAD     : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
```

```
 $ LOAN    : num   26100 35000 9100 5700 28100 44000 14200 16000 3900 24800 ...
 $ ID      : int   4952 5546 938 277 5204 5762 2354 2896 76 4700 ...
 $ MORTDUE: num   73525 391000 17218 58400 61000 ...
 $ VALUE   : num   89870 505000 36721 75000 99000 ...
 $ REASON : Factor w/ 3 levels "DebtCon","HomeImp",..: 1 1 1 2 1 1 1 1 2 1 ...
 $ JOB     : Factor w/ 7 levels "Mgr","missing",..: 3 5 4 5 1 4 5 4 4 1 ...
```

```
> hmeqtest<-as.data.frame(hmeqtest)
> BAD<-as.factor(hmeqtest[,2])
> LOAN<-as.numeric(hmeqtest[,3])
> hmeqtest<-cbind(BAD,LOAN,hmeqtest[,-2,-3])
> hmeqtest$REASON<-as.factor(hmeqtest$REASON)
> hmeqtest$JOB<-as.factor(hmeqtest$JOB)
> str(hmeqtest)
```

```
$ BAD     : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN    : num   38700 8000 17300 18600 20000 9000 6900 13700 25000 5000 ...
 $ ID      : int   5632 675 3234 3537 3804 926 462 2229 4770 184 ...
 $ LOAN    : int   38700 8000 17300 18600 20000 9000 6900 13700 25000 5000 ...
 $ MORTDUE: num   119847 37871 73000 64248 60336 ...
 $ VALUE   : num   162365 89870 95000 82690 132430 ...
 $ REASON : Factor w/ 3 levels "DebtCon","HomeImp",..: 2 2 1 1 1 1 2 1 1 1 ...
 $ JOB     : Factor w/ 7 levels "Mgr","missing",..: 5 5 4 1 4 4 6 5 6 4 ...
```

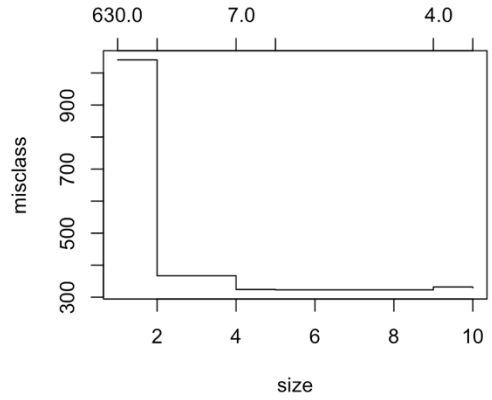
```
'data.frame':       378 obs. of   8 variables:
> library(tree)
> tree1<-tree(BAD~.-ID,data=hmeqtrain,split=c("deviance"), na.action=na.pass,
+                  control=tree.control(nobs=nrow(hmeqtrain), minsize=10, mindev=0.005))
> plot(tree1)
> text(tree1)
```



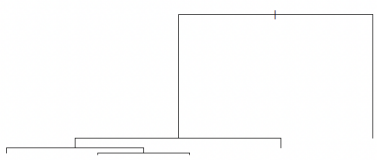
> print(tree1)

```
node), split, n, deviance, yval, (yprob)   * denotes terminal node

 1) root 2000 2773.000 1 ( 0.50000 0.50000 )
   2) LOAN < 11650 1367 1590.000 2 ( 0.26847 0.73153 )
     4) MORTDUE < 42761.5 278   385.300 1 ( 0.51079 0.48921 )
       8) LOAN < 6250 81    74.580 1 ( 0.82716 0.17284 )
        16) JOB: Mgr,missing,Other 63    24.120 1 ( 0.95238 0.04762 ) *
        17) JOB: Office,ProfExe,Self 18    24.060 2 ( 0.38889 0.61111 ) *
       9) LOAN > 6250 197   261.800 2 ( 0.38071 0.61929 )
        18) VALUE < 88777 190   247.800 2 ( 0.35789 0.64211 ) *
        19) VALUE > 88777 7      0.000 1 ( 1.00000 0.00000 ) *
     5) MORTDUE > 42761.5 1089 1110.000 2 ( 0.20661 0.79339 )
      10) JOB: Mgr,Other,ProfExe,Sales,Self 810   914.300 2 ( 0.25185 0.74815 )
        20) LOAN < 5750 119   164.800 1 ( 0.52101 0.47899 )
          40) LOAN < 3100 16      7.481 1 ( 0.93750 0.06250 ) *
          41) LOAN > 3100 103   142.000 2 ( 0.45631 0.54369 ) *
        21) LOAN > 5750 691   702.000 2 ( 0.20550 0.79450 )
          42) VALUE < 29272 9      0.000 1 ( 1.00000 0.00000 ) *
          43) VALUE > 29272 682   673.000 2 ( 0.19501 0.80499 ) *
      11) JOB: missing,Office 279   149.000 2 ( 0.07527 0.92473 ) *
   3) LOAN > 11650 633      0.000 1 ( 1.00000 0.00000 ) *
```

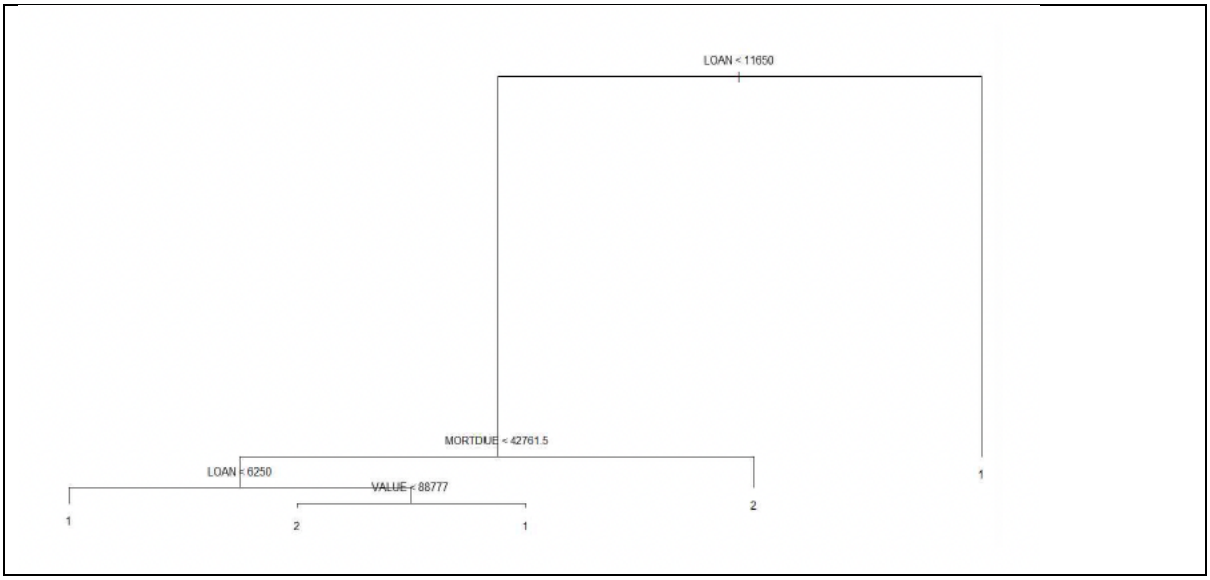
> cv.tree1<-cv.tree(tree1,FUN=prune.misclass,K=10)
> plot(cv.tree1)



> tree1<-prune.misclass(tree1,best=5)
> plot(tree1)



> text(tree1)



> summary(tree1)

```
Classification tree:
snip.tree(tree = tree1, nodes = c(8L, 5L))
Variables actually used in tree construction:
[1] "LOAN"    "MORTDUE" "VALUE"
Number of terminal nodes:  5
Residual mean deviance:  0.7178 = 1432 / 1995
Misclassification error rate: 0.1535 = 307 / 2000
```

> tree2<-predict(tree1,hmeqtest[,-1,-3],type="class")
> summary(tree2)

```
1   2
141 237
```

> print(tree2)

```
[1] 1 1 1 1 1 2 2 1 1 1 1 1 1 2 1 2 2 1 1 1 1 2 1 1 2 1 1 1 1
 [30] 1 2 1 2 1 2 1 1 1 1 1 2 2 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2
 [59] 1 1 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 2 2 1
 [88] 1 1 2 1 1 1 1 2 1 2 1 1 1 1 2 2 1 2 1 2 1 1 1 2 1 2 2 1 2
[117] 1 1 1 1 1 1 2 2 1 1 1 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 1 1 1
[146] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 1 2 1 2 2 1 2 2 1 1
[175] 2 2 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
[204] 2 2 2 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
[233] 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[262] 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 2 2 2
[291] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[320] 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 2 2 2
[349] 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 2 2 2
[378] 2
```

```
Levels: 1 2
```

```
> tree3<-cbind(hmeqtest[,1],tree2)
> tree3<-as.data.frame(tree3)
> print(tree3)
```

```
     V1 tree2
1     1       1
2     1       1
...
377   2       2
378   2       2
```

```
     V1 tree2
1     1       1
2     1       1
3     1       1
375   2       2
376   2       2
377   2       2
378   2       2
```

```
> with(tree3,table(V1,tree2))
```

```
tree2
V1      1    2
   1 138   51
   2   3 186
```

```
> library(writexl)
> writexl::write_xlsx(tree3,path="tree3.xlsx")
```

2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	2
8	1	2
9	1	1
10	1	1
11	1	1

```
> tree2<-predict(tree1,hmeqtest[,-1,-3],type="vector")
> print(tree2)
```

```
1             2
1    1.0000000 0.0000000
2    1.0000000 0.0000000
```

```
...
377 0.2066116 0.7933884
378 0.2066116 0.7933884
```

```
> hmeqtrain<-read.csv("hmeqN_train.csv",header = TRUE, fileEncoding = "euc-kr")
> str(hmeqtrain)
```

```
'data.frame':	2000 obs. of  7 variables:
 $ ID      : int  4952 5546 938 277 5204 5762 2354 2896 76 4700 ...
 $ BAD     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN    : int  26100 35000 9100 5700 28100 44000 14200 16000 3900 24800 ...
 $ MORTDUE: num  73525 391000 17218 58400 61000 ...
 $ VALUE   : num  89870 505000 36721 75000 99000 ...
 $ REASON : chr  "DebtCon" "DebtCon" "DebtCon" "HomeImp" ...
 $ JOB     : chr  "Office" "ProfExe" "Other" "ProfExe" ...
```

```
> head(hmeqtrain,3);tail(hmeqtrain,3)
```

```
ID BAD  LOAN MORTDUE  VALUE  REASON      JOB
1 4952   1 26100   73525  89870 DebtCon  Office
2 5546   1 35000  391000 505000 DebtCon ProfExe
3  938   1  9100   17218  36721 DebtCon   Other
       ID BAD LOAN MORTDUE VALUE  REASON      JOB
1998 936   2 9100   80719 86782 HomeImp   Other
1999 209   2 5100   62702 79784 HomeImp   Other
2000 399   2 6500   80739 97630 DebtCon ProfExe
```

```
> hmeqtest <-read.csv("hmeqN_test.csv",header = TRUE, fileEncoding = "euc-kr")
> str(hmeqtest)
```

```
'data.frame':	378 obs. of  7 variables:
 $ ID      : int  5632 675 3234 3537 3804 926 462 2229 4770 184 ...
 $ BAD     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN    : int  38700 8000 17300 18600 20000 9000 6900 13700 25000 5000 ...
 $ MORTDUE: num  119847 37871 73000 64248 60336 ...
 $ VALUE   : num  162365 89870 95000 82690 132430 ...
 $ REASON : chr  "HomeImp" "HomeImp" "DebtCon" "DebtCon" ...
 $ JOB     : chr  "ProfExe" "ProfExe" "Other" "Mgr" ...
```

```
> library(reshape)
> hmeqtrain<-as.data.frame(hmeqtrain)
> BAD<-as.factor(hmeqtrain[,2])
> LOAN<-as.numeric(hmeqtrain[,3])
> hmeqtrain<-cbind(BAD,LOAN,hmeqtrain[,-2:-3])
> hmeqtrain$REASON<-as.factor(hmeqtrain$REASON)
> hmeqtrain$JOB<-as.factor(hmeqtrain$JOB)
> str(hmeqtrain)
```

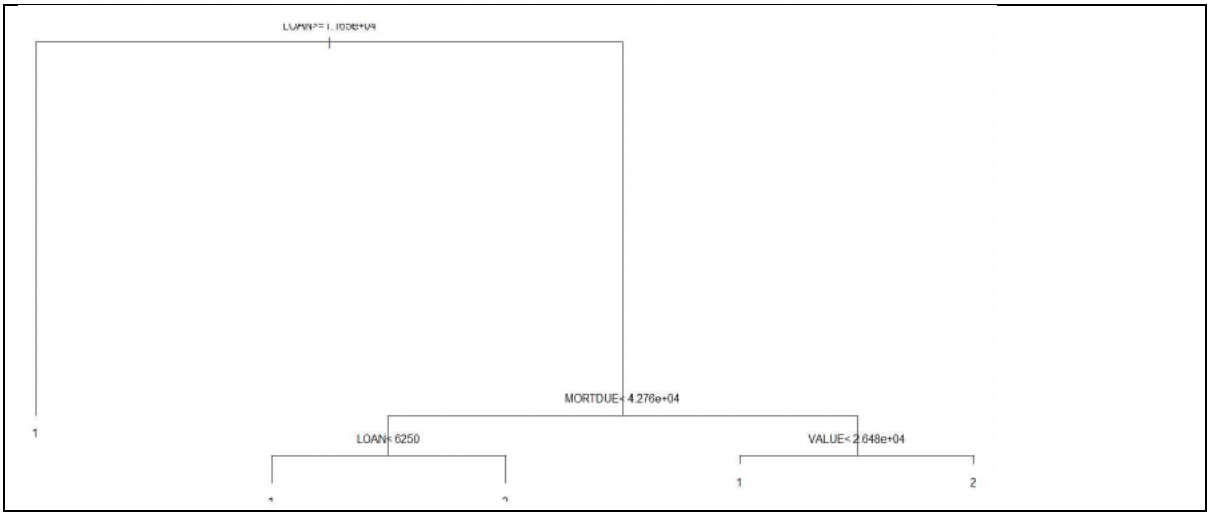
```
'data.frame':	2000 obs. of  7 variables:
 $ BAD     : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
```

```
 $ LOAN     : num   26100 35000 9100 5700 28100 44000 14200 16000 3900 24800 ...
 $ ID       : int   4952 5546 938 277 5204 5762 2354 2896 76 4700 ...
 $ MORTDUE: num   73525 391000 17218 58400 61000 ...
 $ VALUE    : num   89870 505000 36721 75000 99000 ...
 $ REASON : Factor w/ 3 levels "DebtCon","HomeImp",..: 1 1 1 2 1 1 1 1 2 1 ...
 $ JOB      : Factor w/ 7 levels "Mgr","missing",..: 3 5 4 5 1 4 5 4 4 1 ...
```

```
> hmeqtest<-as.data.frame(hmeqtest)
> BAD<-as.factor(hmeqtest[,2])
> LOAN<-as.numeric(hmeqtest[,3])
> hmeqtest<-cbind(BAD,LOAN,hmeqtest[,-2:-3])
> hmeqtest$REASON<-as.factor(hmeqtest$REASON)
> hmeqtest$JOB<-as.factor(hmeqtest$JOB)
> str(hmeqtest)
```

```
'data.frame':       378 obs. of  7 variables:
 $ BAD      : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN     : num   38700 8000 17300 18600 20000 9000 6900 13700 25000 5000 ...
 $ ID       : int   5632 675 3234 3537 3804 926 462 2229 4770 184 ...
 $ MORTDUE: num   119847 37871 73000 64248 60336 ...
 $ VALUE    : num   162365 89870 95000 82690 132430 ...
 $ REASON : Factor w/ 3 levels "DebtCon","HomeImp",..: 2 2 1 1 1 1 2 1 1 1 ...
 $ JOB      : Factor w/ 7 levels "Mgr","missing",..: 5 5 4 1 4 4 6 5 6 4 ...
```

```
> library(rpart)
> hmeqrpart<-rpart(BAD~.-ID,data=hmeqtrain,method="class",control=rpart.control(minsplit=10,maxdepth=5))
> plot(hmeqrpart); text(hmeqrpart)
```




```
> print(hmeqrpart)
```

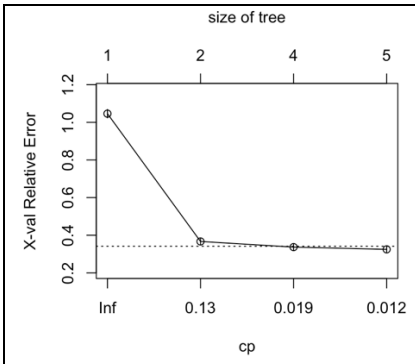
```
n= 2000

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 2000 1000 1 (0.5000000 0.5000000)
```

```
    2) LOAN>=11650 633       0 1 (1.0000000 0.0000000) *
    3) LOAN< 11650 1367    367 2 (0.2684711 0.7315289)
      6) MORTDUE< 42761.5 278    136 1 (0.5107914 0.4892086)
       12) LOAN< 6250 81      14 1 (0.8271605 0.1728395) *
       13) LOAN>=6250 197      75 2 (0.3807107 0.6192893) *
      7) MORTDUE>=42761.5 1089    225 2 (0.2066116 0.7933884)
       14) VALUE< 26475 16       1 1 (0.9375000 0.0625000) *
       15) VALUE>=26475 1073    210 2 (0.1957130 0.8042870) *
```

> printcp(hmeqrpart)

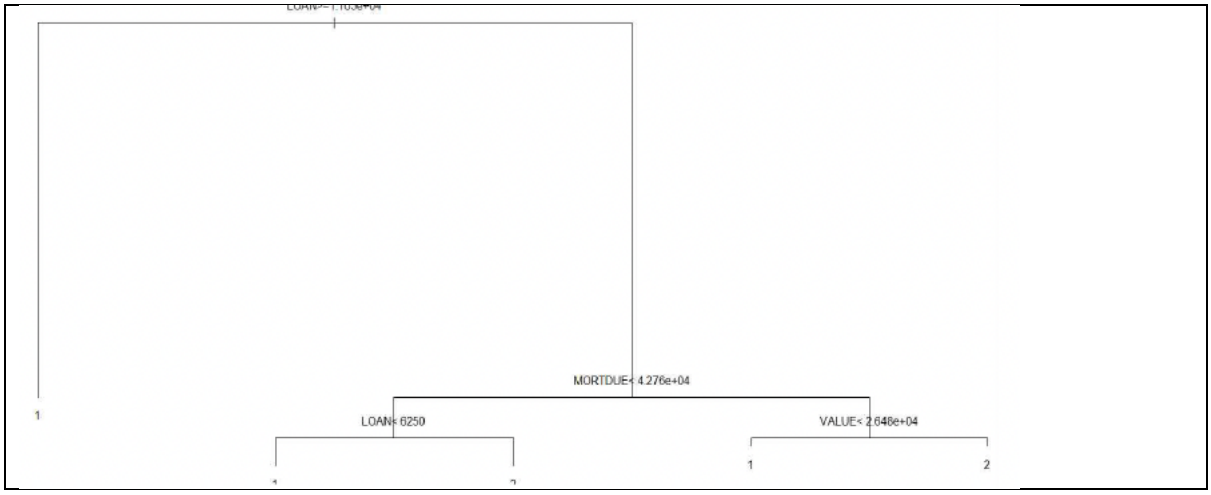
```
Classification tree:
rpart(formula = BAD ~ . - ID, data = hmeqtrain, method = "class",
      control = rpart.control(minsplit = 10, maxdepth = 5))

Variables actually used in tree construction:
[1] LOAN      MORTDUE VALUE

Root node error: 1000/2000 = 0.5

n= 2000

        CP nsplit rel error xerror        xstd
1 0.6330      0       1.000   1.046 0.022337
2 0.0265      1       0.367   0.367 0.017311
3 0.0140      3       0.314   0.337 0.016740
4 0.0100      4       0.300   0.325 0.016498
```

>plotcp(hmeqrpart)



> hmeqrpart<-prune(hmeqrpart,cp=hmeqrpart$cptable[which.min(hmeqrpart$cptable[,"xerror"]),"CP"])

> plot(hmeqrpart);text(hmeqrpart)




```
> hmeqpred<-predict(hmeqrpart,hmeqtest[,-1,-3],type="class")
> summary(hmeqpred)
```

```
1   2
140 238
```

```
> tree3<-cbind(hmeqtest[,1],hmeqpred)
> tree3<-as.data.frame(tree3)
> print(tree3)
```

```
V1 hmeqpred
1      1            1
2      1            2
...
377    2            2
378    2            2
```

```
> with(tree3,table(V1,hmeqpred))
```

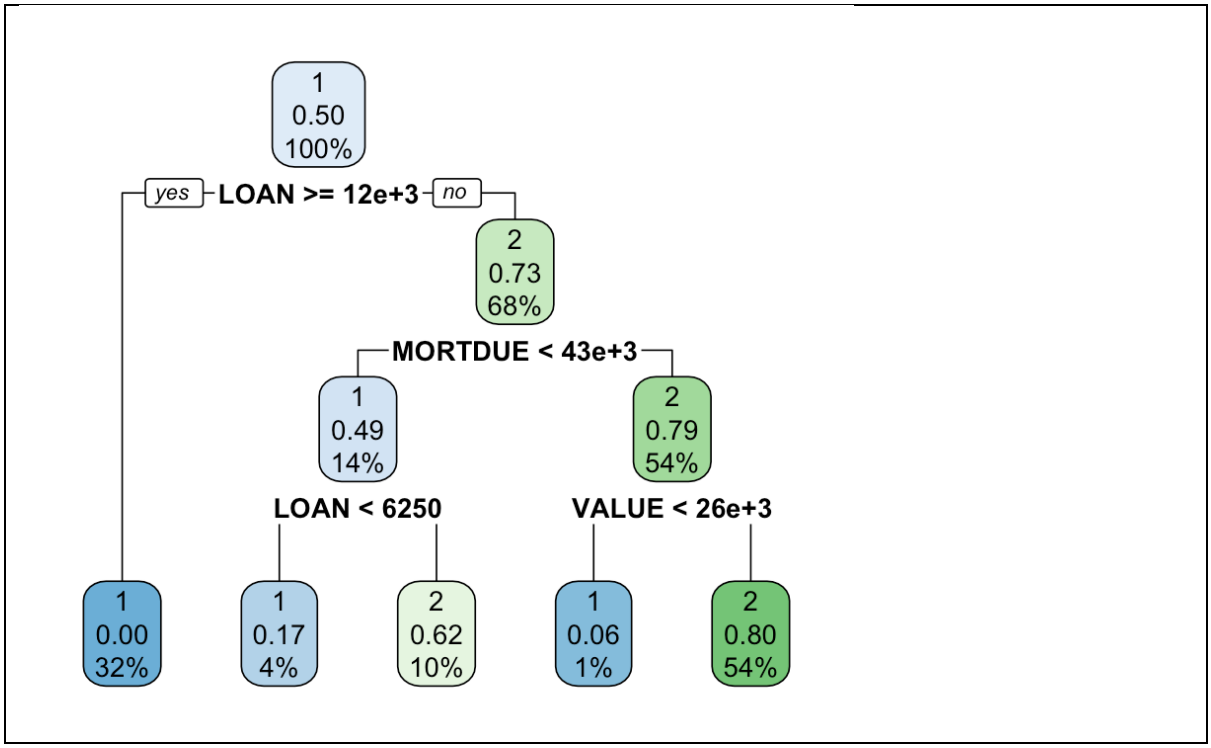
```
hmeqpred
V1      1    2
   1 137   52
   2    3 186
```

```
> hmeqpred<-predict(hmeqrpart,hmeqtest[,-1,-3],type="prob")
> print(hmeqpred)
```

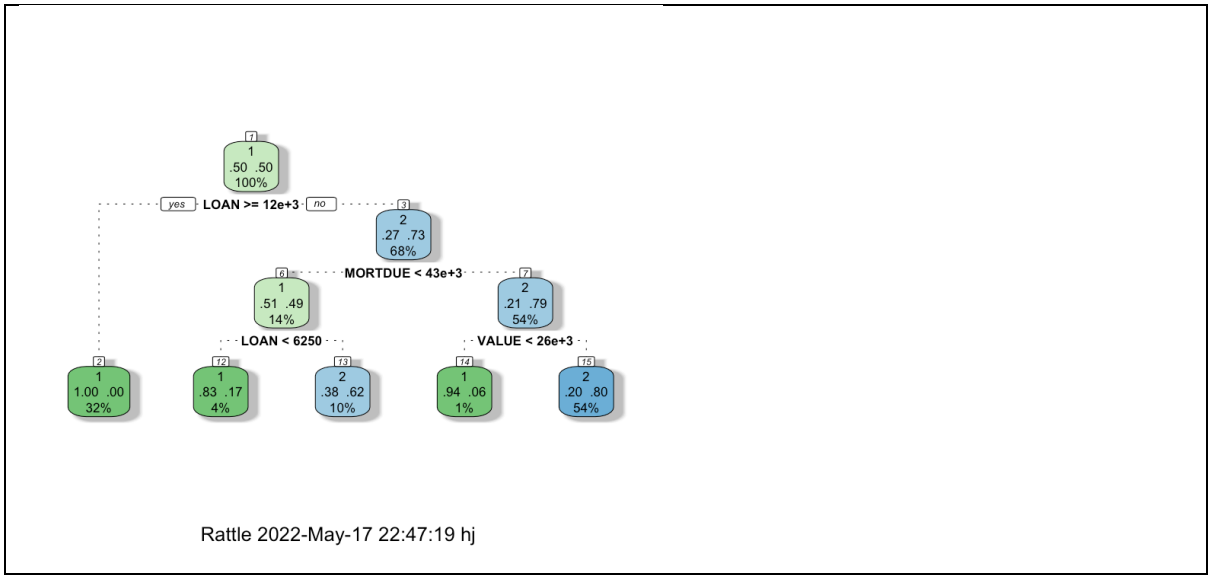
```
1             2
1    1.0000000 0.0000000
2    0.3807107 0.6192893
...
377 0.1957130 0.8042870
378 0.1957130 0.8042870
```

```
> library(rattle)
> library(rpart.plot)
> library(RColorBrewer)
```

> rpart.plot(hmeqrpart)



> fancyRpartPlot(hmeqrpart)



> hmeqtrain<-read.csv("hmeqN_train.csv",header = TRUE, fileEncoding = "euc-kr")
> str(hmeqtrain)

```
'data.frame':	2000 obs. of  7 variables:
 $ ID     : int  4952 5546 938 277 5204 5762 2354 2896 76 4700 ...
 $ BAD    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN   : int  26100 35000 9100 5700 28100 44000 14200 16000 3900 24800 ...
 $ MORTDUE: num  73525 391000 17218 58400 61000 ...
 $ VALUE  : num  89870 505000 36721 75000 99000 ...
 $ REASON : chr  "DebtCon" "DebtCon" "DebtCon" "HomeImp" ...
```

```
 $ JOB      : chr   "Office" "ProfExe" "Other" "ProfExe" ...
```

```
> head(hmeqtrain,3);tail(hmeqtrain,3)
```

```
ID BAD   LOAN MORTDUE   VALUE   REASON       JOB
1 4952    1 26100    73525   89870 DebtCon   Office
2 5546    1 35000   391000 505000 DebtCon ProfExe
3   938    1   9100    17218   36721 DebtCon    Other
        ID BAD LOAN MORTDUE VALUE   REASON       JOB
1998 936    2 9100    80719 86782 HomeImp    Other
1999 209    2 5100    62702 79784 HomeImp    Other
2000 399    2 6500    80739 97630 DebtCon ProfExe
```

```
> hmeqtest <-read.csv("hmeqN_test.csv”,header = TRUE, fileEncoding = "euc-kr")
> str(hmeqtest)
```

```
'data.frame':       378 obs. of   7 variables:
 $ ID       : int   5632 675 3234 3537 3804 926 462 2229 4770 184 ...
 $ BAD      : int   1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN     : int   38700 8000 17300 18600 20000 9000 6900 13700 25000 5000 ...
 $ MORTDUE: num   119847 37871 73000 64248 60336 ...
 $ VALUE    : num   162365 89870 95000 82690 132430 ...
 $ REASON : chr   "HomeImp" "HomeImp" "DebtCon" "DebtCon" ...
 $ JOB      : chr   "ProfExe" "ProfExe" "Other" "Mgr" ...
```

```
> library(reshape)
> hmeqtrarin<-as.data.frame(hmeqtrain)
> BAD<-as.factor(hmeqtrain[,2])
> LOAN<-as.numeric(hmeqtrain[,3])
> hmeqtrain<-cbind(BAD,LOAN,hmeqtrain[,-2:-3])
> hmeqtrain$REASON<-as.factor(hmeqtrain$REASON)
> hmeqtrain$JOB<-as.factor(hmeqtrain$JOB)
> str(hmeqtrain)
```

```
'data.frame':       2000 obs. of   7 variables:
 $ BAD      : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN     : num   26100 35000 9100 5700 28100 44000 14200 16000 3900 24800 ...
 $ ID       : int   4952 5546 938 277 5204 5762 2354 2896 76 4700 ...
 $ MORTDUE: num   73525 391000 17218 58400 61000 ...
 $ VALUE    : num   89870 505000 36721 75000 99000 ...
 $ REASON : Factor w/ 3 levels "DebtCon","HomeImp",..: 1 1 1 2 1 1 1 1 2 1 ...
 $ JOB      : Factor w/ 7 levels "Mgr","missing",..: 3 5 4 5 1 4 5 4 4 1 ...
```

```
> hmeqtest<-as.data.frame(hmeqtest)
> BAD<-as.factor(hmeqtest[,2])
> LOAN<-as.numeric(hmeqtest[,3])
> hmeqtest<-cbind(BAD,LOAN,hmeqtest[,-2:-3])
> hmeqtest$REASON<-as.factor(hmeqtest$REASON)
> hmeqtest$JOB<-as.factor(hmeqtest$JOB)
> str(hmeqtest)
```

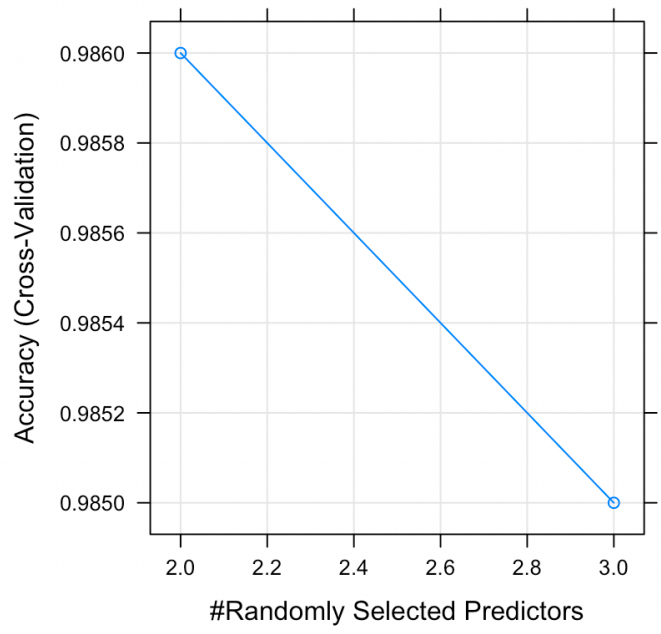
```
'data.frame':       378 obs. of   7 variables:
```

```
 $ BAD      : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN     : num  38700 8000 17300 18600 20000 9000 6900 13700 25000 5000 ...
 $ ID       : int  5632 675 3234 3537 3804 926 462 2229 4770 184 ...
 $ MORTDUE: num  119847 37871 73000 64248 60336 ...
 $ VALUE    : num  162365 89870 95000 82690 132430 ...
 $ REASON : Factor w/ 3 levels "DebtCon","HomeImp",..: 2 2 1 1 1 1 2 1 1 1 ...
 $ JOB      : Factor w/ 7 levels "Mgr","missing",..: 5 5 4 1 4 4 6 5 6 4 ...
```

```
> library(RWeka)
> library(caret)
> train<-createFolds(hmeqtrain$BAD,k=10)
> C45Fit<-train(BAD~LOAN+MORTDUE+VALUE+REASON+JOB,data=hmeqtrain,
+                     trControl=trainControl(method="cv",indexOut=train))
> C45Fit
```

```
Random Forest

2000 samples
   5 predictor
   2 classes: '1', '2'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1800, 1800, 1800, 1800, 1800, 1800, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.8815     0.763
  6     0.9870     0.974
  11    0.9860     0.972

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 6.
```

```
> plot(C45Fit)
```



```
> C45Fit$finalModel
Call:
 randomForest(x = x, y = y, mtry = min(param$mtry, ncol(x)))
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 6

        OOB estimate of  error rate: 12.7%
Confusion matrix:
     1   2 class.error
1 805 195       0.195
2  59 941       0.059
> results<-predict(object=C45Fit,newdata=hmeqtest,type="raw")
> table(hmeqtest$BAD,results)
   results
      1   2
  1 158  31
  2  19 170
```

```
> results<-predict(object=C45Fit,newdata=hmeqtest,type="prob")
> print(results)
1       2
1   1.000 0.000
2   0.666 0.334
...
377 0.234 0.766
378 0.062 0.938
```

```
> hmeqtrain <-read.csv("hmeqN_train.csv",header = TRUE, fileEncoding = "euc-kr")
> str(hmeqtrain)
'data.frame':       2000 obs. of  7 variables:
 $ ID      : int  4952 5546 938 277 5204 5762 2354 2896 76 4700 ...
 $ BAD     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN    : int  26100 35000 9100 5700 28100 44000 14200 16000 3900 24800 ...
 $ MORTDUE: num  73525 391000 17218 58400 61000 ...
 $ VALUE   : num  89870 505000 36721 75000 99000 ...
 $ REASON : chr  "DebtCon" "DebtCon" "DebtCon" "HomeImp" ...
 $ JOB     : chr  "Office" "ProfExe" "Other" "ProfExe" ...
```

```
> head(hmeqtrain,3);tail(hmeqtrain,3)
ID BAD  LOAN MORTDUE   VALUE   REASON      JOB
1 4952   1 26100    73525   89870 DebtCon   Office
2 5546   1 35000   391000 505000 DebtCon ProfExe
3  938   1  9100    17218   36721 DebtCon    Other
        ID BAD LOAN MORTDUE VALUE   REASON      JOB
1998 936   2 9100    80719 86782 HomeImp    Other
1999 209   2 5100    62702 79784 HomeImp    Other
```

```
2000 399     2 6500     80739 97630 DebtCon ProfExe
```

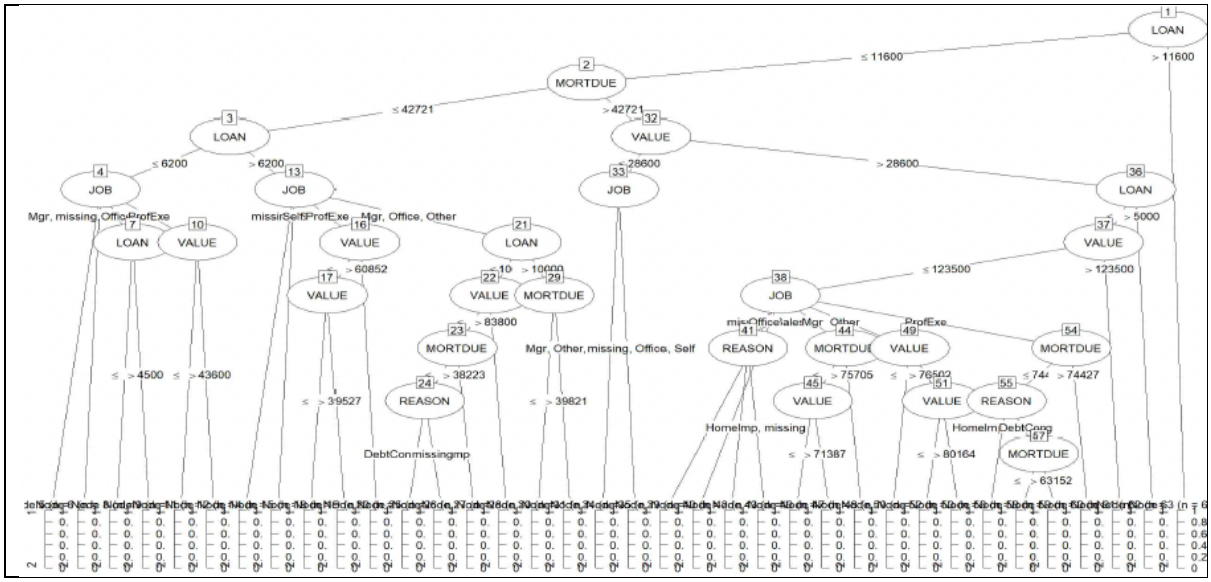
```
> hmeqtest <-read.csv("hmeqN_test.csv",header = TRUE, fileEncoding = "euc-kr")
> str(hmeqtest)
'data.frame':       378 obs. of    7 variables:
 $ ID       : int   5632 675 3234 3537 3804 926 462 2229 4770 184 ...
 $ BAD      : int   1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN     : int   38700 8000 17300 18600 20000 9000 6900 13700 25000 5000 ...
 $ MORTDUE: num   119847 37871 73000 64248 60336 ...
 $ VALUE    : num   162365 89870 95000 82690 132430 ...
 $ REASON : chr   "HomeImp" "HomeImp" "DebtCon" "DebtCon" ...
 $ JOB      : chr   "ProfExe" "ProfExe" "Other" "Mgr" ...
```

```
> library(reshape)
> hmeqtrain<-as.data.frame(hmeqtrain)
> BAD<-as.factor(hmeqtrain[,2])
> LOAN<-as.numeric(hmeqtrain[,3])
> hmeqtrain<-cbind(BAD,LOAN,hmeqtrain[,-2,-3])
> hmeqtrain$REASON<-as.factor(hmeqtrain$REASON)
> hmeqtrain$JOB<-as.factor(hmeqtrain$JOB)
> str(hmeqtrain)
'data.frame':       2000 obs. of    8 variables:
 $ BAD      : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN     : num   26100 35000 9100 5700 28100 44000 14200 16000 3900 24800 ...
 $ ID       : int   4952 5546 938 277 5204 5762 2354 2896 76 4700 ...
 $ LOAN     : int   26100 35000 9100 5700 28100 44000 14200 16000 3900 24800 ...
 $ MORTDUE: num   73525 391000 17218 58400 61000 ...
 $ VALUE    : num   89870 505000 36721 75000 99000 ...
 $ REASON : Factor w/ 3 levels "DebtCon","HomeImp",..: 1 1 1 2 1 1 1 1 2 1 ...
 $ JOB      : Factor w/ 7 levels "Mgr","missing",..: 3 5 4 5 1 4 5 4 4 1 ...
```

```
> hmeqtest<-as.data.frame(hmeqtest)
> BAD<-as.factor(hmeqtest[,2])
> LOAN<-as.numeric(hmeqtest[,3])
> hmeqtest<-cbind(BAD,LOAN,hmeqtest[,-2,-3])
> str(hmeqtest)
'data.frame':       378 obs. of    8 variables:
 $ BAD      : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ LOAN     : num   38700 8000 17300 18600 20000 9000 6900 13700 25000 5000 ...
 $ ID       : int   5632 675 3234 3537 3804 926 462 2229 4770 184 ...
 $ LOAN     : int   38700 8000 17300 18600 20000 9000 6900 13700 25000 5000 ...
 $ MORTDUE: num   119847 37871 73000 64248 60336 ...
 $ VALUE    : num   162365 89870 95000 82690 132430 ...
 $ REASON : chr   "HomeImp" "HomeImp" "DebtCon" "DebtCon" ...
 $ JOB      : chr   "ProfExe" "ProfExe" "Other" "Mgr" ...
```

```
> library(C50)
> library(printr)
```

> tree1<-C5.0(BAD~LOAN+MORTDUE+VALUE+REASON+JOB,data=hmeqtrain)
> plot(tree1)



> summary(tree1)

```
Call:
C5.0.formula(formula = BAD ~ LOAN + MORTDUE + VALUE + REASON + JOB, data
  = hmeqtrain, trials = 10)


C5.0 [Release 2.07 GPL Edition]     Tue May 17 23:13:00 2022
-------------------------------

Class specified by attribute `outcome'

Read 2000 cases (6 attributes) from undefined.data

-----  Trial 0:  -----

Decision tree:

LOAN > 11600: 1 (633)
LOAN <= 11600:
:...MORTDUE <= 42721:
    :...LOAN <= 6200:
    :   :...JOB = Self: 2 (6/1)
    :   :   JOB in {Mgr,missing,Other,Sales}: 1 (63/3)
    :   :   JOB = Office:
    :   :   :...LOAN <= 4500: 1 (3)
    :   :   :   LOAN > 4500: 2 (3)
    :   :   JOB = ProfExe:
    :   :   :...VALUE <= 43600: 2 (3)
    :   :       VALUE > 43600: 1 (3)
    :   LOAN > 6200:
    :   :...JOB in {missing,Sales}: 2 (8)
    :       JOB = Self: 1 (6/1)
    :       JOB = ProfExe:
    :       :...VALUE > 60852: 2 (16)
    :       :   VALUE <= 60852:
    :       :   :...VALUE <= 39527: 2 (4)
    :       :       VALUE > 39527: 1 (4)
    :       JOB in {Mgr,Office,Other}:
    :       :...LOAN > 10000:
    :           :...MORTDUE <= 39821: 2 (33/3)
```

```
:                :    MORTDUE > 39821: 1 (6/1)
:                LOAN <= 10000:
:                :...VALUE > 83800: 1 (7)
:                    VALUE <= 83800:
:                    :...MORTDUE > 38223: 2 (25/2)
:                        MORTDUE <= 38223:
:                        :...REASON in {DebtCon,HomeImp}: 1 (85/36)
:                            REASON = missing: 2 (3)
...
Trial            Decision Tree
-----          ----------------
               Size        Errors

   0             36   219(10.9%)
   1             18   330(16.5%)
   2             30   310(15.5%)
   3             18   359(17.9%)
   4             31   365(18.2%)
   5             16   412(20.6%)
   6             36   415(20.8%)
   7             30   374(18.7%)
   8             37   246(12.3%)
   9             30   278(13.9%)
boost                142( 7.1%)    <<


                (a)    (b)      <-classified as
               ----   ----
                875    125      (a): class 1
                 17    983      (b): class 2


            Attribute usage:

            100.00%    LOAN
             68.35%    MORTDUE
             68.35%    VALUE
             68.35%    JOB
             61.90%    REASON


Time: 0.0 secs
```

```
> results<-predict(object=tree1,newdata=hmeqtest,type="class")
> table(hmeqtest$BAD,results)
results
      1   2
  1 156  33
  2  20 169
```

```
> results<-predict(object=tree1,newdata=hmeqtest,type="prob")
> print(results)
1            2
1   1.00000000 0.0000000
2   0.70996726 0.2900327
...
```

```
377 0.39892986 0.6010701
378 0.00000000 1.0000000
```

```
> tree1<-C5.0(BAD~LOAN+MORTDUE+VALUE+REASON+JOB,data=hmeqtrain,trials=10)
```

```
> summary(tree1)
```

```
Call:
C5.0.formula(formula = BAD ~ LOAN + MORTDUE + VALUE + REASON + JOB, data
  = hmeqtrain, trials = 10)


C5.0 [Release 2.07 GPL Edition]     Tue May 17 23:15:00 2022
-------------------------------

Class specified by attribute `outcome'

Read 2000 cases (6 attributes) from undefined.data

-----  Trial 0:  -----

Decision tree:

LOAN > 11600: 1 (633)
LOAN <= 11600:
:...MORTDUE <= 42721:
    :...LOAN <= 6200:
    :   :...JOB = Self: 2 (6/1)
    :   :   JOB in {Mgr,missing,Other,Sales}: 1 (63/3)
    :   :   JOB = Office:
    :   :   :...LOAN <= 4500: 1 (3)
    :   :   :   LOAN > 4500: 2 (3)
    :   :   JOB = ProfExe:
    :   :   :...VALUE <= 43600: 2 (3)
    :   :       VALUE > 43600: 1 (3)
    :   LOAN > 6200:
    :   :...JOB in {missing,Sales}: 2 (8)
    :       JOB = Self: 1 (6/1)
    :       JOB = ProfExe:
    :       :...VALUE > 60852: 2 (16)
    :       :   VALUE <= 60852:
    :       :   :...VALUE <= 39527: 2 (4)
    :       :       VALUE > 39527: 1 (4)
    :       JOB in {Mgr,Office,Other}:
    :       :...LOAN > 10000:
    :           :...MORTDUE <= 39821: 2 (33/3)
    :           :   MORTDUE > 39821: 1 (6/1)
    :           LOAN <= 10000:
    :           :...VALUE > 83800: 1 (7)
    :               VALUE <= 83800:
    :               :...MORTDUE > 38223: 2 (25/2)
    :                   MORTDUE <= 38223:
    :                   :...REASON in {DebtCon,HomeImp}: 1 (85/36)
    :                       REASON = missing: 2 (3)
    MORTDUE > 42721:
    :...VALUE <= 28600:
        :...JOB in {Mgr,Other,ProfExe,Sales,Self}: 1 (19/2)
        :   JOB in {missing,Office}: 2 (6)
        VALUE > 28600:
        :...LOAN > 5000: 2 (960/163)
            LOAN <= 5000:
            :...VALUE > 123500: 1 (6)
                VALUE <= 123500:
```

```
                    :...JOB in {missing,Sales}: 2 (15/2)
                        JOB = Self: 1 (1)
                        JOB = Office:
                        :...REASON = DebtCon: 1 (2)
                        :    REASON in {HomeImp,missing}: 2 (15)
                        JOB = Mgr:
                        :...MORTDUE > 75705: 1 (4)
                        :    MORTDUE <= 75705:
                        :    :...VALUE <= 71387: 1 (2)
                        :         VALUE > 71387: 2 (9)
                        JOB = Other:
                        :...VALUE <= 76502: 1 (15/1)
                        :    VALUE > 76502:
                        :    :...VALUE <= 80164: 2 (5)
                        :         VALUE > 80164: 1 (9/3)
                        JOB = ProfExe:
                        :...MORTDUE > 74427: 2 (6)
                            MORTDUE <= 74427:
                            :...REASON in {HomeImp,missing}: 1 (6/1)
                                REASON = DebtCon:
                                :...MORTDUE <= 63152: 2 (6)
                                    MORTDUE > 63152: 1 (3)

-----  Trial 1:  -----

Decision tree:

LOAN > 11600: 1 (494.2)
LOAN <= 11600:
:...JOB in {missing,Office}:
    :...JOB = missing: 2 (88.3/7.9)
    :    JOB = Office:
    :    :...MORTDUE > 63152: 2 (95.9/13.9)
    :        MORTDUE <= 63152:
    :        :...VALUE <= 66500: 2 (82.5/16.8)
    :             VALUE > 66500: 1 (50.9/20)
    JOB in {Mgr,Other,ProfExe,Sales,Self}:
    :...LOAN <= 7500:
        :...LOAN <= 3100: 1 (29.8/3.6)
        :    LOAN > 3100:
        :    :...LOAN > 7400: 1 (41.2/6.7)
        :        LOAN <= 7400:
        :        :...JOB in {Mgr,Sales,Self}: 1 (96.5/33.2)
        :            JOB = ProfExe: 2 (84.9/42)
        :            JOB = Other:
        :            :...MORTDUE <= 67366: 1 (164.5/67.4)
        :                MORTDUE > 67366: 2 (33.3/3.6)
        LOAN > 7500:
        :...LOAN > 9900:
            :...LOAN <= 10000: 1 (77.2/22.1)
            :    LOAN > 10000: 2 (294.8/116.3)
            LOAN <= 9900:
            :...LOAN > 9100: 2 (135.4/31)
                LOAN <= 9100:
                :...LOAN <= 7800: 2 (19.2/1.6)
                    LOAN > 7800:
                    :...LOAN <= 8000: 1 (41.8/15.6)
                        LOAN > 8000:
                        :...LOAN <= 8800: 2 (116.5/32.9)
                            LOAN > 8800: 1 (53.2/23.4)
...
Evaluation on training data (2000 cases):

Trial           Decision Tree
```

```
-----           ----------------
                Size        Errors

   0              36   219(10.9%)
   1              18   330(16.5%)
   2              30   310(15.5%)
   3              18   359(17.9%)
   4              31   365(18.2%)
   5              16   412(20.6%)
   6              36   415(20.8%)
   7              30   374(18.7%)
   8              37   246(12.3%)
   9              30   278(13.9%)
boost                142( 7.1%)    <<


                (a)    (b)     <-classified as
                ----   ----
                 875    125     (a): class 1
                  17    983     (b): class 2


             Attribute usage:

             100.00%    LOAN
              68.35%    MORTDUE
              68.35%    VALUE
              68.35%    JOB
              61.90%    REASON


Time: 0.0 secs
```

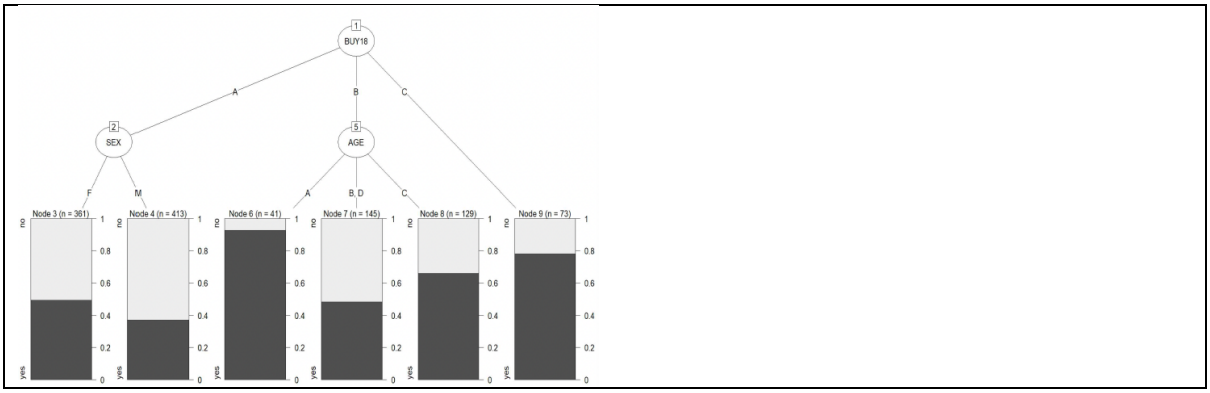
```
> results<-predict(object=tree1,newdata=hmeqtest,type="class")
> table(hmeqtest$BAD,results)
```

```
results
       1    2
  1 156   33
  2   20 169
```

```
> buytrain <-read.csv("buy_train_chaid.csv",header = TRUE, fileEncoding = "euc-kr")
> buytrain$RESPOND<-as.factor(buytrain$RESPOND)
> buytrain$AGE<-as.factor(buytrain$AGE)
> buytrain$SEX<-as.factor(buytrain$SEX)
> buytrain$MARRIED<-as.factor(buytrain$MARRIED)
> buytrain$BUY18<-as.factor(buytrain$BUY18)
> str(buytrain)
```

```
'data.frame':       1162 obs. of    6 variables:
 $ ID       : int   1 2 3 4 5 6 7 8 9 10 ...
 $ RESPOND: Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ AGE      : Factor w/ 4 levels "A","B","C","D": 2 1 2 3 2 4 1 3 2 3 ...
 $ SEX      : Factor w/ 2 levels "F","M": 1 1 2 2 1 2 1 2 1 1 ...
 $ MARRIED: Factor w/ 2 levels "MA","notMA": 2 2 1 1 2 1 2 1 2 1 ...
 $ BUY18   : Factor w/ 3 levels "A","B","C": 1 1 2 1 2 1 2 1 1 1 ...
```

```
> buytest <-read.csv("buy_test_chaid.csv",header = TRUE, fileEncoding = "euc-kr")
> buytest$RESPOND<-as.factor(buytest$RESPOND)
> buytest$AGE<-as.factor(buytest$AGE)
> buytest$SEX<-as.factor(buytest$SEX)
> buytest$MARRIED<-as.factor(buytest$MARRIED)
> buytest$BUY18<-as.factor(buytest$BUY18)
> str(buytest)
'data.frame':	300 obs. of  6 variables:
 $ ID      : int  1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 ...
 $ RESPOND: Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ AGE     : Factor w/ 4 levels "A","B","C","D": 4 4 4 3 2 1 2 2 4 4 ...
 $ SEX     : Factor w/ 2 levels "F","M": 2 1 2 2 2 2 2 1 2 2 ...
 $ MARRIED: Factor w/ 2 levels "MA","notMA": 2 1 1 1 2 1 2 2 1 1 ...
 $ BUY18   : Factor w/ 3 levels "A","B","C": 1 2 2 1 3 1 1 1 2 1 ...

> library(partykit)
Loading required package: grid
Loading required package: libcoin
Loading required package: mvtnorm

> library(CHAID)
> ctrl<-chaid_control(minsplit=20,minprob=0.1)
> chaidresult<-chaid(RESPOND~AGE+SEX+MARRIED+BUY18,data=buytrain,control=ctrl)
> print(chaidresult)
Model formula:
RESPOND ~ AGE + SEX + MARRIED + BUY18

Fitted party:
[1] root
|   [2] BUY18 in A
|   |   [3] SEX in F: no (n = 361, err = 49.3%)
|   |   [4] SEX in M: no (n = 413, err = 37.0%)
|   [5] BUY18 in B
|   |   [6] AGE in A: yes (n = 41, err = 7.3%)
|   |   [7] AGE in B, D: no (n = 145, err = 48.3%)
|   |   [8] AGE in C: yes (n = 129, err = 34.1%)
|   [9] BUY18 in C: yes (n = 73, err = 21.9%)

Number of inner nodes:    3
Number of terminal nodes: 6

> plot(chaidresult)
```




```
> tree2<-predict(chaidresult,buytest[,-1,-2])
> summary(tree2)
```

```
no yes
230  70
```

```
> tree3<-cbind(buytest[,2],tree2)
> tree3<-as.data.frame(tree3)
> with(tree3,table(V1,tree2))
```

```
tree2
V1        1    2
  no   119   31
  yes 111   39
```

```
> tree2<-predict(chaidresult,buytest[,-1,-2],type='prob')
> print(tree2)
```

```
no          yes
1    0.62953995 0.3704600
2    0.51724138 0.4827586
...
299 0.62953995 0.3704600
300 0.62953995 0.3704600
```