다변량분석 과제 6

2015170378

정은영

전체 데이터셋을 임의로 200 명이 포함된 Training dataset 과 103 명 Validation dataset 으로 구분한 뒤 다음 각 물음에 답하시오. 분류 성능을 평가/비교할 때는 TPR, TNR, Precision, Accuracy, BCR, F1- Measure 를 모두 고려하여 의견을 서술하시오.

```r
#Assignment6_2015170378 정은영

# Performance Evaluation Function ---------------------------------------
perf_eval <- function(cm){

  # True positive rate: TPR (Recall)
  TPR <- cm[2,2]/sum(cm[2,])
  # Precision
  PRE <- cm[2,2]/sum(cm[,2])
  # True negative rate: TNR
  TNR <- cm[1,1]/sum(cm[1,])
  # Simple Accuracy
  ACC <- (cm[1,1]+cm[2,2])/sum(cm)
  # Balanced Correction Rate
  BCR <- sqrt(TPR*TNR)
  # F1-Measure
  F1 <- 2*TPR*PRE/(TPR+PRE)

  return(c(TPR, PRE, TNR, ACC, BCR, F1))
}

# Performance table
Perf.Table <- matrix(0, nrow = 2, ncol = 6)
rownames(Perf.Table) <- c("CART(pruned 'x')","CART(pruned 'o')")
colnames(Perf.Table) <- c("TPR", "Precision", "TNR", "Accuracy", "BCR", "F1-Measure")
```

-> 분류 성능 평가 matrix 생성 및 함수 생성

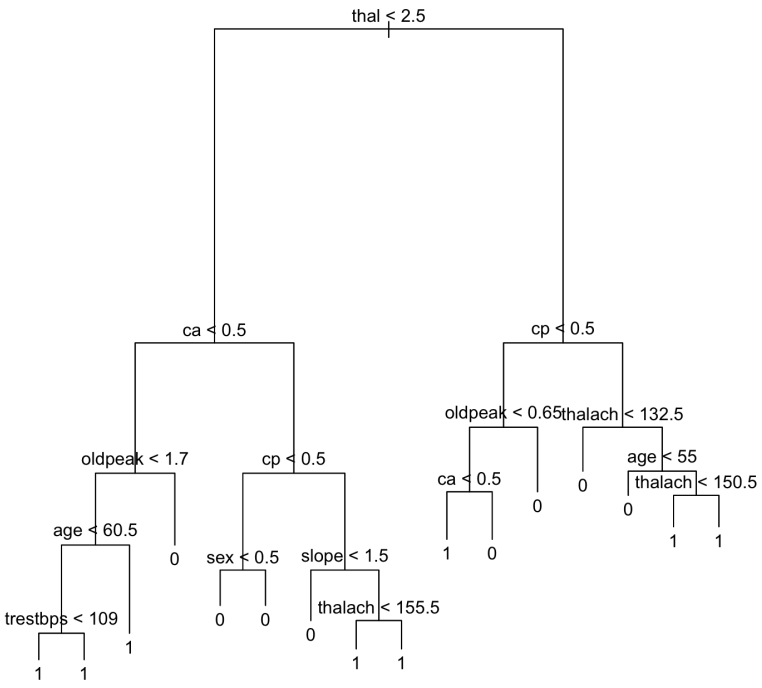
```r
# Load the data & Preprocessing
df <- read.csv("heart.csv")
input.idx <- c(1:13)
target.idx <- 14

df.input <- df[,input.idx]
df.target <- as.factor(df[,target.idx])

df.data <- data.frame(df.input, df.target)

nR <- nrow(df)
nR

set.seed(12345) #랜덤 알고리즘 지정
trn.idx <- sample(1:nR, 200) #train data 를 임의로 200개 뽑기
```

-> 전체 데이터셋을 임의로 200 명이 포함된 Training dataset 과 103 명 Validation dataset 으로 구분

[Q1] 실습 시간에 사용한 "tree"package 를 사용하여 ClassificationTree 를 학습한 결과물을 Plotting 하고 이에 대한 해석을 수행하시오. 또한 해당 Tree 를 pruning 을 수행하지 않은 상태에서 Validation dataset 에 대한 분류 성능을 평가하시오.

```
#[Q1]
install.packages("tree")
library(tree)

tree.trn <- data.frame(df.input[trn.idx,], HeartYN = df.target[trn.idx])
tree.tst <- data.frame(df.input[-trn.idx,], HeartYN = df.target[-trn.idx])

# Training the tree
CART.model <- tree(HeartYN ~ ., tree.trn)
summary(CART.model)

# Plot the tree
plot(CART.model)
text(CART.model, pretty = 1)

# Prediction
CART.prey <- predict(CART.model, tree.tst, type = "class")
CART.cfm <- table(tree.tst$HeartYN, CART.prey)
CART.cfm

# Evaluation
Perf.Table[1,] <- perf_eval(CART.cfm)
Perf.Table
```

-> tree package 를 이용하여 classification tree 학습 및 plotting, predict 한 뒤 evaluation.



각 Column 이 뜻하는 바는 다음과 같다. age(age in years), sex(1 = male; 0 = female), cp(chest pain type), trestbps(resting blood pressure (in mm Hg on admission to the hospital)), chol(serum cholestoral in mg/dl), fbs((fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)), restecg(resting electrocardiographic results), thalach(maximum heart rate achieved), exang(exercise induced angina (1 = yes; 0 = no)), oldpeak(ST depression induced by exercise relative to rest), slope(the slope of the peak exercise ST segment), ca(number of major vessels (0-3) colored by flouroscopy), thal(3 = normal; 6 = fixed defect; 7 = reversable defect), target(1 or 0)

연령, 성별, 흉통유형(4 가지 값), 휴식혈압, 혈청칼슘농도, 공복혈당, 휴식심전도결과(0,1,2), 최대 심박수 달성, 운동성 협심증(0,1), oldpeak(휴식과 운동에 의해 유도된 ST 우울증), 피크 운동 ST 의 기울기, 형광투시법으로 채색된 주요혈관(0-3)의 수, thal(원본 데이터에서,3:보통, 6:fixed defect, 7: reversable defect, 이 데이터에서는 1,2,3 으로 기록), 타켓(1:심장질환, 0:존재 x)

따라서 위의 classification tree 를 보았을 때, 하나를 예를 들자면 thal 이 2.5 보다 작고, 즉 정상범위, ca 가 0.5 보다 작고, 즉 형광투시법으로 채색된 주요혈관의 수가 0, oldpeak 가 1.7 보다 작고, 나이가 60.5 세보다 작고, trestbps(휴식혈압)이 109 보다 낮을 때 심장병이 발병한다고 분류한다. 하지만 이 classification tree 는 full tree 로 (leaf node 16 개) overfitting 의 가능성이 있다. tree 에서는 9 가지 변수를 사용하여 분류하였다. 또한 tree 를 자세히 보면 하위 sex 의 경우 부등호에 상관없이 즉, 여자 남자에 상관없이 결과값이 같음을 확인할 수 있다.

다음은 validation data 를 가지고 시행한 예측에서의 confusion matrix 이다.

```
  CART.prey
    0  1
0 31 11
1 15 46
```

위의 모델으로 prediction 을 해본 결과 심장병 발병시 발병했다고 분류한 것은 46 건, 발병 시 발병 안했다고 분류한 건수는 15 건, 발병 안했는데 했다고 한 건수는 11 건, 안했는데 안했다고 한 건수는 31 건이다. 잘못 판단한 건수의 약 3 배가 제대로 판단한 건수로 생각보다 오류가 높았다.

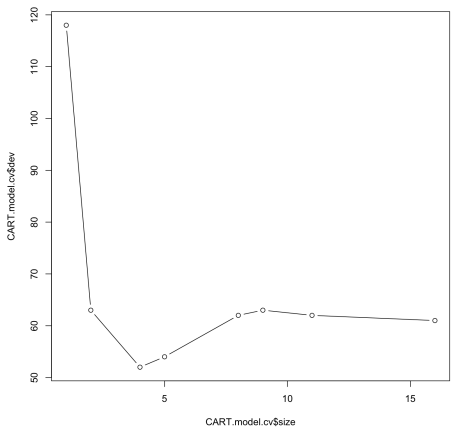
```
> Perf.Table
                        TPR Precision       TNR  Accuracy       BCR F1-Measure
CART(pruned 'x') 0.7540984 0.8070175 0.7380952 0.7475728 0.7460539   0.779661
```

evaluation 을 하기 위해 각각을 구해본 결과는 다음과 같다. 발병했다고 예측했을 때 실제 발병한지(Precision)은 약 80.7%이며, 실제 발병했을 때 발병했다고 예측한 경우(TPR)은 75.4%, 실제 발병하지 않았는데 발병하지 않았다고 예측한 경우(TNR)은 약 73.8%, 전체 중에 제대로 예측할 확률(Accuracy)는 74.8%, 발병하지 않았거나 했을 경우 하나만 잘 예측하고 하나는 잘 예측하지 못할 경우를 대비한 인자인(BCR)은 74.6%, 데이터셋의 불균형을 맞춰주는 FI 측정지표는 80.0%정도를 기록하였다.

[Q2] 앞에서 생성한 Tree 에 대해서 적절한 Pruning 을 수행한 뒤 결과물을 Plotting 하고 이에 대한 해석을 수행하시오. Pruning 전과 후에 Split 에 사용된 변수는 어떤 변화가 있는가? Validation dataset 에 대한 분류 성능을 평가하고 [Q1]의 결과와 비교해보시오.

```
#[Q2]
# Find the best tree
set.seed(12345)
CART.model.cv <- cv.tree(CART.model, FUN = prune.misclass)

# Plot the pruning result
plot(CART.model.cv$size, CART.model.cv$dev, type = "b")
CART.model.cv #best 찾기
```

best tree 를 찾기 위해 size 와 deviation 으로 그래프를 그려보고 확인해본다. k-fold Cross-validation 방법을 사용해서 train 셋을 여러번 쪼개서 테스트 한 다음 분산이 가장 낮은 가지의 수를 선택한다.



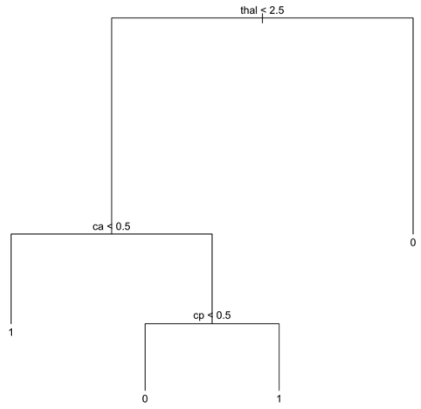

```
> CART.model.cv #best 찾기
$size
[1] 16 11  9  8  5  4  2  1

$dev
[1]  61  62  63  62  54  52  63 118

$k
[1] -Inf  0.0  0.5  1.0  2.0  3.0  7.0 45.0

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
```

size 가 4 일 때 dev 가 52 로 가장 작기 때문에 size 를 4 로 선택한다.

```
# Select the final model
CART.model.pruned <- prune.misclass(CART.model, best = 4)
plot(CART.model.pruned) # 4개만 나타남!
text(CART.model.pruned, pretty = 1)

# Prediction
CART.prey <- predict(CART.model.pruned, tree.tst, type = "class")
CART.cfm <- table(tree.tst$HeartYN, CART.prey)
CART.cfm

Perf.Table[2,] <- perf_eval(CART.cfm)
Perf.Table
```

pruning 을 진행하고, plot 을 그려보면, 총 4 개의 leaf node 가 나오게 된다.

Pruning 전의 tree 는 9 개의 변수로 분류하였는데, 이는 thal, ca, cp 세개의 변수로 심장병 발병 여부가 분류된다.



따라서 옆의 classification tree 를 보았을 때, thal 이 2.5 보다 작고(1,2) 일 때, ca(형광투시법으로 채색된 주요혈관의 수)가 0.5 보다 작으면 발병, 1 개 이상이고, cp(흉통유형)이 '0'이면 발병하지 않는다고 분류, 다른 3 가지 유형에 대해서는 발병한다고 분류한다. 또한 thal 이 3 일 때 발병하지 않는다고 분류한다.

따라서 Pruning 된 트리로 predict 하면 다음과 같은 confusion matrix 가 그려진다.

```
> CART.cfm
   CART.prey
     0  1
  0 37  5
  1 14 47
```

이는 pruning 전의 Matrix 와 비교했을 때, 잘못 예측할 건수가 적어졌다는 것을 알 수 있다.

아래의 preference table 을 볼 때 평가지표들이 모두 증가했음을 알 수 있다. 위의 문제와 비교하여 pruning 후가 성능이 좋기 때문에 사용한다.

```
> Perf.Table
                       TPR Precision       TNR  Accuracy       BCR F1-Measure
CART(pruned 'x') 0.7540984 0.8070175 0.7380952 0.7475728 0.7460539  0.7796610
CART(pruned 'o') 0.7704918 0.9038462 0.8809524 0.8155340 0.8238729  0.8318584
```

[Q3] "tree" package 이외에 R 에서 Classification Tree 를 학습할 수 있는 package 를 최소 세 개 이상 찾아서 [Q1]과 [Q2]에서 사용한 데이터셋과 동일한 데이터셋으로 Classification Tree 를 학습하고 분류 성능 을 평가해 보시오. 각 package 에 대해서 아래 사항들에 대해서 개별적으로 답하시오.

```
89  #[Q3]
90
91  #Preference Matrix
92  Perf.Table_2 <- matrix(0, nrow = 4, ncol = 6)
93  rownames(Perf.Table_2) <- c("tree","rpart","RWeka","party")
94  colnames(Perf.Table_2) <- c("TPR", "Precision", "TNR", "Accuracy", "BCR", "F1-Measure")
95
96  Perf.Table_2[1,] <- perf_eval(CART.cfm)
97  Perf.Table_2
98
99  #[Q3-1] 사용한 패키지의 이름
100 install.packages("RWeka")
101 install.packages("rpart")
102 install.packages("party")
103 install.packages("partykit") #J48모델 그릴때 필요
104 library(rpart)
105 library(RWeka)
106 library(party)
107 library(partykit)
```

[Q3-1] 사용한 패키지의 이름

[Q3-2] 사용한 패키지가 Classification Tree 를 학습할 때 사용자가 지정할 수 있는 옵션의 종류와 의미

[Q3-3] 본인이 실제로 옵션을 변화시켜 가면서 학습한 Classification Tree 들의 차이점 및 패키지별로 최종적으로 선정한 Best model 에 대한 설명

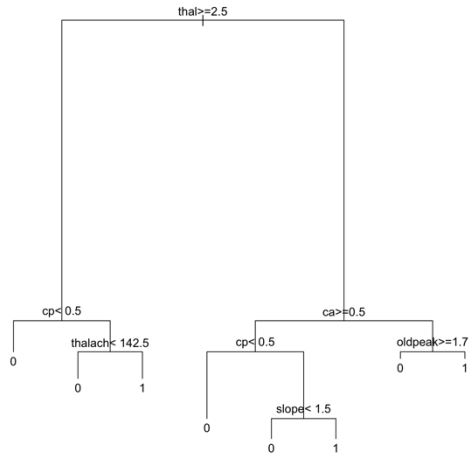
[Q3-4] 각 패키지에서 제공하는 Tree plot 및 (가능할 경우) 다른 시각화 package 를 사용하여 도시한 Tree plot 들 간의 비교

[Q3-5] 각 패키지에서 제공한 Classification tree 들의 분류 성능 비교 및 논의

>>각각 문제에 대해 주석처리하여 부분을 제시하였고, package 별로 코드를 작성하였다. 비교는 중간 중간과 마지막 둘 다 적도록 하겠다.

 # rpart package (CART)

```
#####rpart#####
#rpart model [Q3-1] package name
rpart.model <- rpart(HeartYN ~ ., data = tree.trn, method = "class")
plot(rpart.model)
text(rpart.model)
```



pruning 전의 tree 는 다음과 같으며, 하나의 예를 들면, thal value 가 2.5 보다 작고(즉 1,2), ca(투시법으로 채색된 주요혈관)이 0.5 보다 크며, 흉통유형이 0 이면 발병하지 않는다고 분류한다.

또한 leaf node 가 8 개인데 과적합의 가능성이 있어 pruning 을 진행한다.

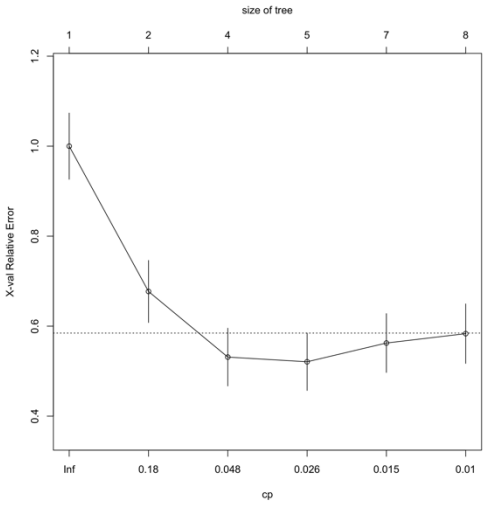
```
#pruning rpart
printcp(rpart.model)
plotcp(rpart.model)
```

```
> printcp(rpart.model)

Classification tree:
rpart(formula = HeartYN ~ ., data = tree.trn, method = "class")

Variables actually used in tree construction:
[1] ca      cp      oldpeak slope   thal    thalach

Root node error: 96/200 = 0.48

n= 200

        CP nsplit rel error  xerror     xstd
1 0.468750      0   1.00000 1.00000 0.073598
2 0.072917      1   0.53125 0.67708 0.068998
3 0.031250      3   0.38542 0.53125 0.064208
4 0.020833      4   0.35417 0.52083 0.063789
5 0.010417      6   0.31250 0.56250 0.065401
6 0.010000      7   0.30208 0.58333 0.066144
```

옆의 결과와 아래의 plot 을 보면,

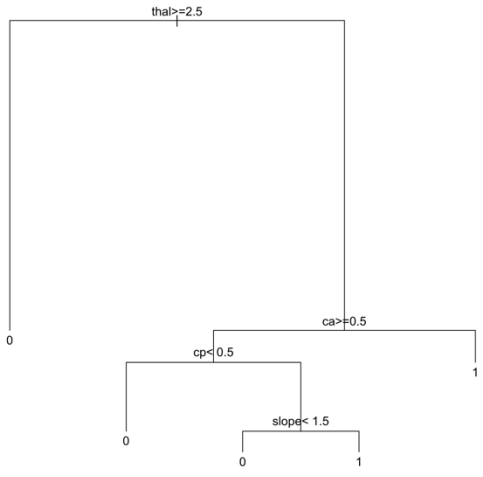
size 가 5 일 때 X-val relative error 가 가장 작게 나타난다.

따라서 size 5 를 선택한다.



옆의 그림을 보고, size 를 5 로 결정한다. 위의 tree pacakage 에서는 best=5 라고 option 을 직접 지정하여 결정하였는데, 옆의 그래프에서 best 를 결정했기 때문에 이 패키지에서는 option 을 하나만 지정하여 결과를 구했다. 또한 수업시간에 교수님이 best 를 나타내는 방식을 다르게 해봤으면 좋겠다 하셔서 which.min 을 통해 rpart 모델에서 xerror 가 가장 작을 때의 CP 로 지정했다.

```
118  #[Q3-2] option_ 교수님께서 다른 방식으로 변환해 보면 좋다고 하셨다, [Q3-3]비교,best
119  rpart.model.pruned <- prune(rpart.model, cp= rpart.model$cptable[which.min(rpart
     .model$cptable[,"xerror"]),"CP"])
120
121  #[Q3-4] plot
122  plot(rpart.model.pruned)
123  text(rpart.model.pruned)
```



결과는 왼쪽과 같으며, thal, ca, cp, slope 네개의 변수를 사용하여 분류한다. best model 의 leaf node 는 5 개이다.

thal 이 2.5 보다 클 때, 즉 value 가 3 일때 발병하지 않으며, 1,2 일 때 중에 채색된 혈관 수가 1 개 이상이고 흉통유형이 0 이면 발병하지 않으며, 채색된 흉통유형이 1,2,3 일 때, slope 즉 최대운동시 ST 의 기울기가 1.5 보다 작으면 발병하지 않지만 크면 발병한다고 분류한다. 또한 채색혈관수가 0 개이면 발병한다고 분류한다.

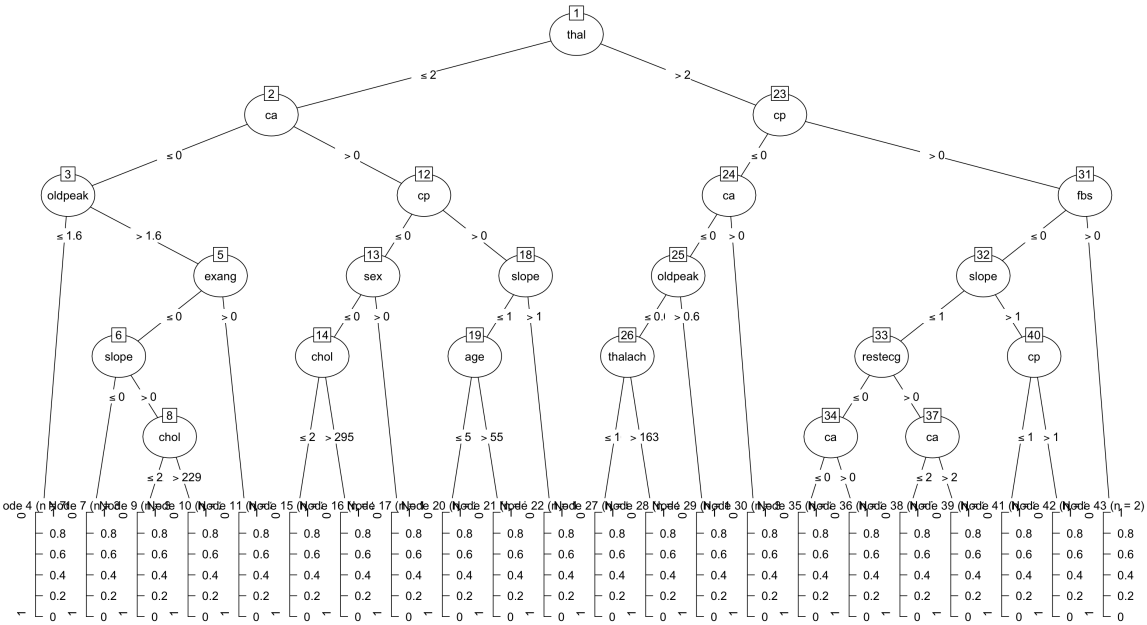
```
> rpart.cfm      > Perf.Table_2
   rpart.prey                TPR Precision       TNR  Accuracy       BCR F1-Measure
     0  1       tree  0.7704918 0.9038462 0.8809524 0.8155340 0.8238729  0.8318584
  0 37  5       rpart 0.7540984 0.9019608 0.8809524 0.8058252 0.8150612  0.8214286
  1 15 46
```

confusion matrix 와 평가지표들을 봤을 때, rpart 를 사용한 tree 모델이 tree 패키지를 사용한 모델보다 약간씩 성능이 떨어짐을 볼 수 있다. tree 패키지를 사용한 모델의 leaf node 수는 4 개로 rpart 를 사용한 모델보다 간단함에도 분류가 잘 됨을 알 수 있다.

 # RWeka package

```
#####RWeka#####
#RWeka model [Q3-1] package name
RWeka.model <- J48(HeartYN ~ ., data = tree.trn)
if(require("party",quietly = TRUE)){plot(RWeka.model)}

#[Q3-2] option
C = c(0.4, 0.25, 0.05) #C: <pruning confidence>, Set confidence threshold for pruning. (default 0.25)
M = c(2,10,20) # M: <minimum number of instances> Set minimum number of instances per leaf. (default 2)

Perf.Table_3 <- matrix(0, nrow = 9, ncol = 6)
rNames <- c()
colnames(Perf.Table_3) <- c("TPR", "Precision", "TNR", "Accuracy", "BCR", "F1-Measure")
```

RWeka model 의 J48 함수는 C4.5 알고리즘을 구현하는 함수이다.



pruning 없이 tree 를 구성할 경우 다음과 같이 나온다. C4.5 에서 중요한 parameter 두개를 뽑아 조정해 보기 위해 parameter C, M 을 조정한다. C 는 pruning 시의 confidence threshold 값이며, default 값은 0.25 이다. 또한 M 은 한 leaf node 에 들어가는 최소한의 instance 개수이며, defalut 값이 2 이다. 따라서 이러한 옵션을 C 3 가지, M 3 가지해서 총 9 가지 경우에 대해 학습과 예측을 통해서 모두 구해보았다. 즉, C=0.4, 0.25, 0.05, M=2, 10, 20 으로 바꾸면서 확인한다. for 문을 통해 9 가지 경우의 수에 대해 학습하고 평가를 해본다.

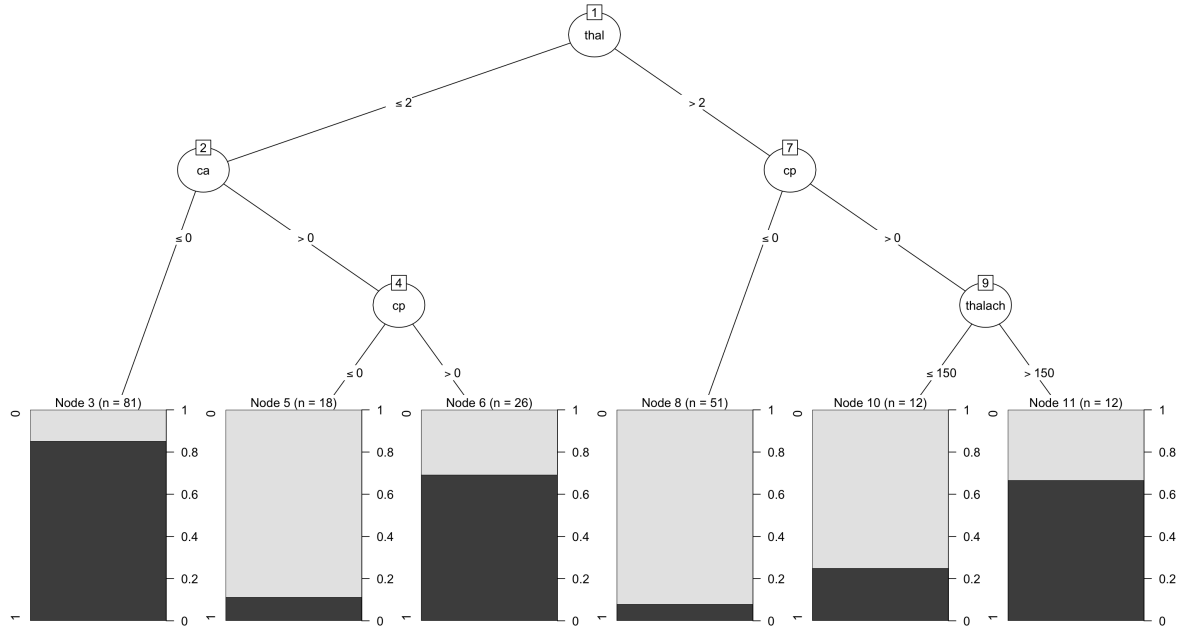
```
for(i in 1:3){
  for(j in 1:3){
    rNames <- c(rNames, paste("C:",C[i],",M:",M[j]))
    RWeka.model.pruned <- J48(HeartYN ~ ., data = tree.trn, control =
Weka_control(C = C[i], M = M[j]))
    if(require("party",quietly = TRUE)){plot(RWeka.model.pruned)}
    RWeka.prey <- predict(RWeka.model.pruned, tree.tst)
    RWeka.cfm <- table(tree.tst$HeartYN, RWeka.prey)
    RWeka.cfm
    Perf.Table_3[(3*(i-1)+j),] <- perf_eval(RWeka.cfm)
  }
}
rownames(Perf.Table_3) <- rNames
Perf.Table_3
#[Q3-3]비교,best
#C: 0.25, M:10 일 때 가장 높음.
```

R-script 에서 각각의 경우의 tree 를 확인 가능하며, best model 을 뽑기 위해 perference table 을 살펴본다.

```
> Perf.Table_3
                    TPR Precision       TNR  Accuracy       BCR F1-Measure
C: 0.4 ,M: 2   0.7377049 0.8035714 0.7380952 0.7378641 0.7379001  0.7692308
C: 0.4 ,M: 10  0.8524590 0.8666667 0.8095238 0.8349515 0.8307141  0.8595041
C: 0.4 ,M: 20  0.7704918 0.8703704 0.8333333 0.7961165 0.8012968  0.8173913
C: 0.25 ,M: 2  0.8360656 0.8225806 0.7380952 0.7961165 0.7855546  0.8292683
C: 0.25 ,M: 10 0.9016393 0.8730159 0.8095238 0.8640777 0.8543410  0.8870968
C: 0.25 ,M: 20 0.7704918 0.8703704 0.8333333 0.7961165 0.8012968  0.8173913
C: 0.05 ,M: 2  0.7704918 0.9038462 0.8809524 0.8155340 0.8238729  0.8318584
C: 0.05 ,M: 10 0.7704918 0.9038462 0.8809524 0.8155340 0.8238729  0.8318584
C: 0.05 ,M: 20 0.7704918 0.8703704 0.8333333 0.7961165 0.8012968  0.8173913
```

C: 0.25, M:10 일 때 평가지표들이 제일 높기 때문에 best model 으로 설정하였다. default 값일 때와 M 이 다르게 나왔다. best model 에 대해서 한번 더 확인해 보는 코드는 아래와 같다.

```
#best model
RWeka.model.pruned <- J48(HeartYN ~ ., data = tree.trn, control =
Weka_control(C = 0.25, M = 10))
#[Q3-4] plot
if(require("party",quietly = TRUE)){plot(RWeka.model.pruned)}
RWeka.prey <- predict(RWeka.model.pruned, tree.tst)
RWeka.cfm <- table(tree.tst$HeartYN, RWeka.prey)
RWeka.cfm

#[Q3-5] 분류성능
Perf.Table_2[3,] <- perf_eval(RWeka.cfm)
Perf.Table_2
```



tree 를 구성하면 다음과 같은데, 검은색 부분이 발병, 회색부분이 발병 안함으로 분류한 것이다. 즉, 하나의 예를 들자면 thal 이 1,2 일 때, 즉 fixed, reversable defect 일 때, ca 즉 채색된 혈관이 없을 때 발병한다고 분류한다. 채색된 혈관이 있을 때는 가슴통증 유형이 0 이 아닐 때 발병한다. 이런식으로 해석할 수 있다. leaf node 의 수는 6 개로 결정되었다.

```
> RWeka.cfm
   RWeka.prey
     0  1
  0 34  8
  1  6 55
```

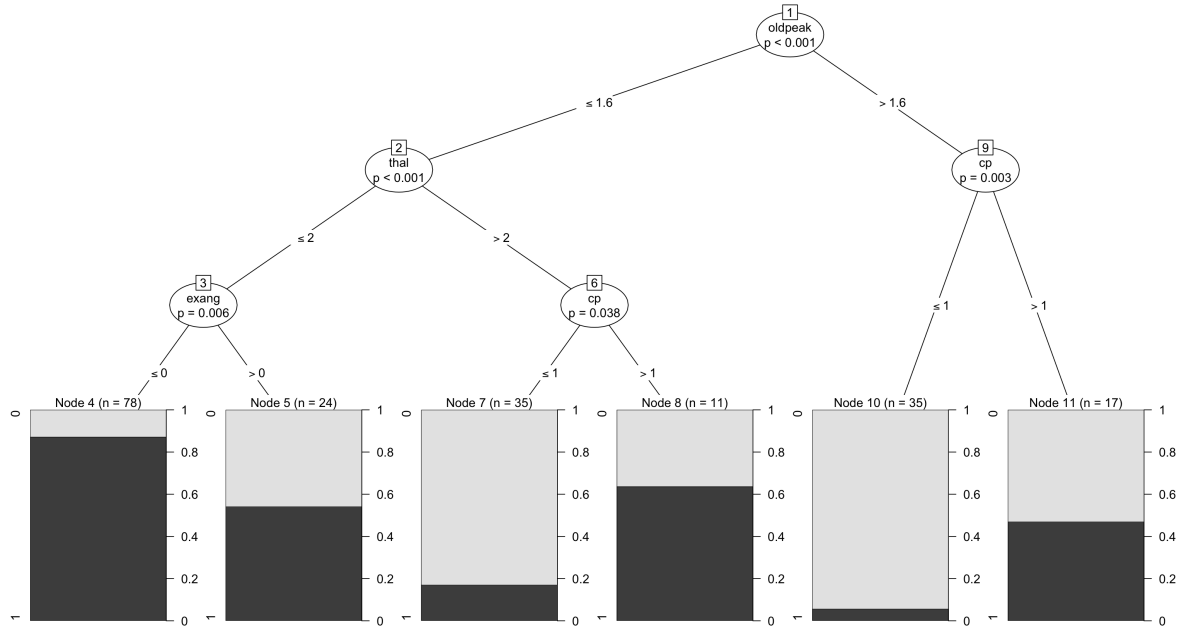
```
> Perf.Table_2
                TPR Precision       TNR  Accuracy       BCR F1-Measure
tree  0.7704918 0.9038462 0.8809524 0.8155340 0.8238729  0.8318584
rpart 0.7540984 0.9019608 0.8809524 0.8058252 0.8150612  0.8214286
RWeka 0.9016393 0.8730159 0.8095238 0.8640777 0.8543410  0.8870968
```

결정된 best 모델의 confusion matrix 를 보면, 잘못 예측된 instance 의 수가 작으며, tree, rpart 와 비교하였을 때 분류성능이 TPR 은 매우 높지만, precision 과 TNR 은 작고, accuracy 는 크며, BCR 은 크고, FI-measure 는 크다. 데이터의 불균형성을 고려한 측정지표인 FI 를 봤을 때 높기 때문에 우수하다는 결론을 내린다.

 # party package

```
174 #####party#####
175 #party model [Q3-1] package name
176 party.model <- ctree(HeartYN ~ ., data = tree.trn)
177 #[Q3-2] option, [Q3-3]비교,best
178 # party패키지는 가지치기를 significance를 사용해서 하기 때문에 별도의 pruning 과정이 필요
    없다.
179 #[Q3-4] plot
180 plot(party.model)
```

party package 의 ctree 함수로 모델을 구성한다. party 패키지는 가지치기를 significance 를 사용해서 하기 때문에 별도의 pruning 과정이 필요 없다는 것을 검색을 통해 알았다. 따라서 별도의 옵션(parameter)조절 없이 진행한다.

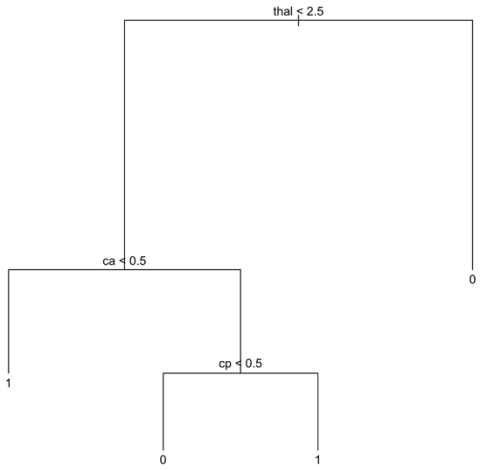


tree 를 구성하면 다음과 같으며 가장 왼쪽에 있는 Leaf node 로 예를 들면 old peak 즉 휴식에 대한 운동으로 이끌어진 ST 우울증이 1.6 이하이고, thal 이 1,2 유형이고, exang 이 운동유발성협심증이 없으면 발병한다. 있어도 발병한다. 또한 thal 이 normal 일 때 cp, 즉 가슴통증의 유형이 0,1 이면 발병하지 않는다, 또한 2,3 이면 발병한다고 분류한다. oldpeak 가 1.6 초과일 때, 가슴통증 유형이 0,1 이면 발병하지 않으며, 2,3 이어도 발병하지 않는다고 분류하겠다.

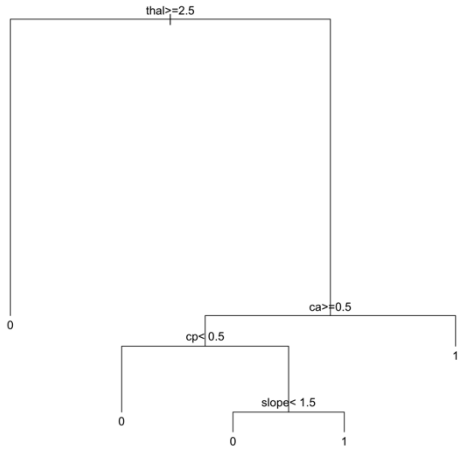
```
> party.cfm
   party.prey
     0  1
  0 30 12
  1 10 51
```

```
> Perf.Table_2
            TPR Precision       TNR  Accuracy       BCR F1-Measure
tree  0.7704918 0.9038462 0.8809524 0.8155340 0.8238729  0.8318584
rpart 0.7540984 0.9019608 0.8809524 0.8058252 0.8150612  0.8214286
RWeka 0.9016393 0.8730159 0.8095238 0.8640777 0.8543410  0.8870968
party 0.8360656 0.8095238 0.7142857 0.7864078 0.7727805  0.8225806
```

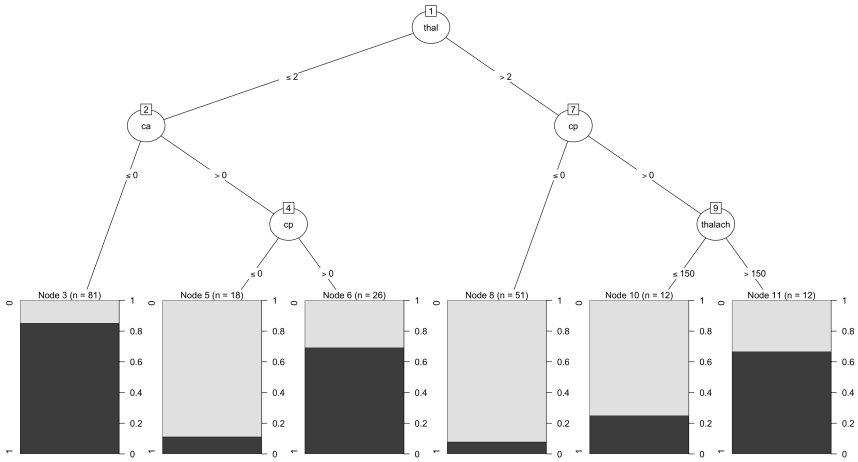
confusion matrix 는 다음과 같고, 평가지표를 확인하면 다음과 같다. party 모델의 성능은 대부분 다른 모델에 비해 살짝 좋지 않게 나타났다. 종합적으로, 개인적인 의견으로 4 개의 package 를 비교했을 때, RWeka 의 confidence threshold 값을 0.25 로, 하나의 leaf node 에 들어가는 최소 개수를 10 개로 한 모델이 성능이 가장 좋기 때문에 이 모델을 선택할 것이다. 다음은 사용한 모든 모델에 사용된 변수를 비교하기 위해 plot 을 한꺼번에 모아놓고 비교한 것이다.



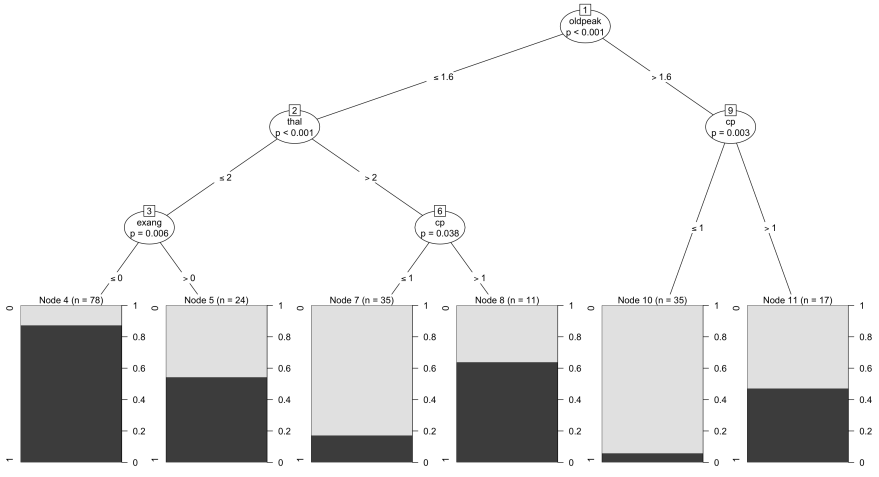
<tree>



<rpart>

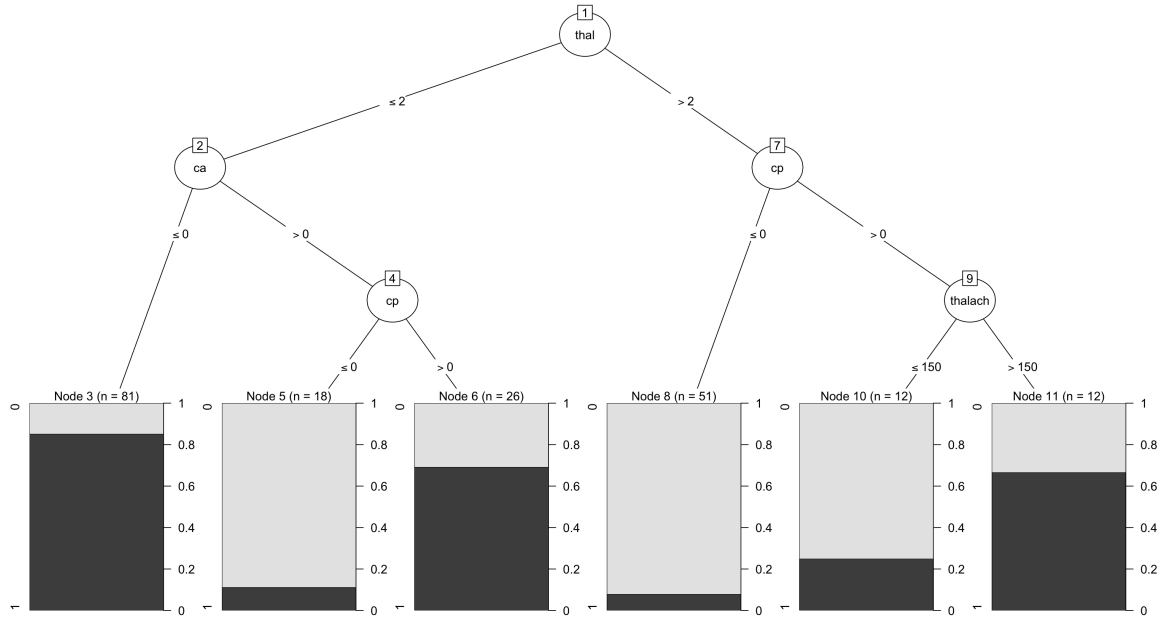


<RWeka 大>



<party>

rpart 는 tree 와 매우 유사하다. CART 이기 때문인 것으로 보인다. tree package 모델에서 slope 변수만 추가되어 분류한다. RWeka 도 살짝 유사한데, tree package 모델에서 thal 이 2.5 이상일 때 cp, thalach 변수들을 사용하여 leaf 가 3 개가 되도록 더 나누었다. party 모델은 다른 모델들과 다르게 oldpeak 변수로 처음에 나눈다. 교수님의 수업 말씀에 따르면 처음에 있는 classification 변수가 중요할 가능성이 높을 확률이 크다고 하셨다. party 모델은 oldpeak 변수를 심장병 발병에 대한 분류변수로 중요하게 배치하였다. 또한 다른 모델과 다르게 특이하게 exang 변수가 들어간다. 결과적으로 분류성능 평가를 통해 결정한 모델은 RWeka 모델이며, 즉 C4.5 알고리즘을 통해 구한 모델이고 다음과 같다.



thal, ca, cp, thalach 변수가 사용되었으며 이 트리를 통해 발병 여부를 분류, 예측할 것이다.