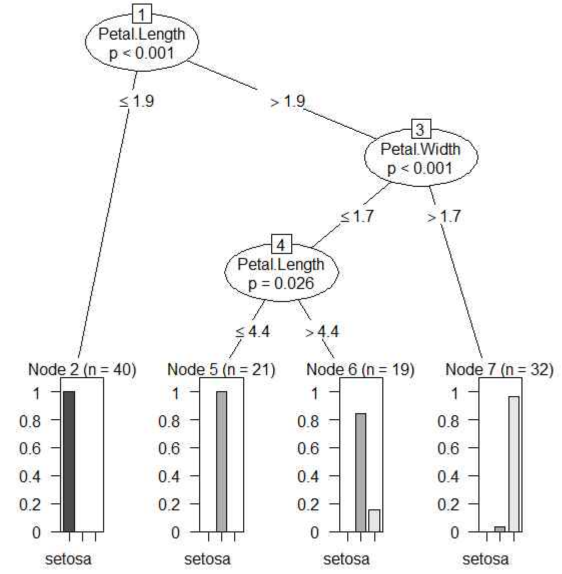
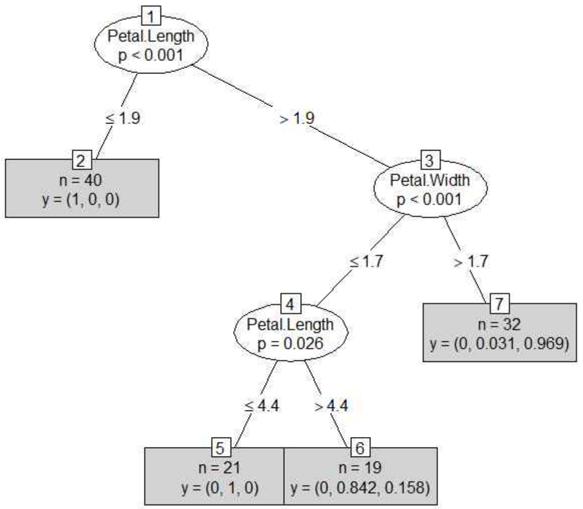
데이터 사이언스 과제6

< 1. Decision Trees and Random Forest >

```
> #1) Decision Trees with Package party
> str(iris)
'data.frame':	150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1
1 1 ...
> set.seed(1234)
> ind <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7, 0.3))
> trainData <- iris[ind==1,]
> testData <- iris[ind==2,]
>
> #install.packages("party")
> library(party)
필요한 패키지를 로딩중입니다: grid
필요한 패키지를 로딩중입니다: mvtnorm
필요한 패키지를 로딩중입니다: modeltools
필요한 패키지를 로딩중입니다: stats4
필요한 패키지를 로딩중입니다: strucchange
필요한 패키지를 로딩중입니다: zoo

다음의 패키지를 부착합니다: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric

필요한 패키지를 로딩중입니다: sandwich
> myFormula <- Species ~ Sepal.Length + Sepal.Width + Petal.Length +
Petal.Width
> iris_ctree <- ctree(myFormula, data=trainData)
> #check the prediction
> table(predict(iris_ctree), trainData$Species)

             setosa versicolor virginica
  setosa         40          0         0
  versicolor      0         37         3
  virginica       0          1        31
>
> print(iris_ctree)

	 Conditional inference tree with 4 terminal nodes

Response:  Species
Inputs:  Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
Number of observations:  112

1) Petal.Length <= 1.9; criterion = 1, statistic = 104.643
  2)*  weights = 40
1) Petal.Length > 1.9
  3) Petal.Width <= 1.7; criterion = 1, statistic = 48.939
    4) Petal.Length <= 4.4; criterion = 0.974, statistic = 7.397
      5)*  weights = 21
    4) Petal.Length > 4.4
      6)*  weights = 19
  3) Petal.Width > 1.7
    7)*  weights = 32
> plot(iris_ctree)
```



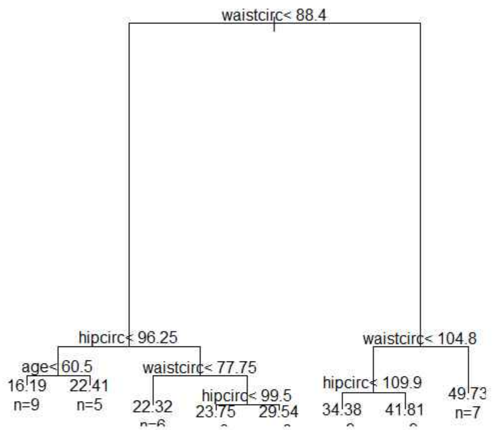
```
> plot(iris_ctree, type="simple")
```



```
> #predict on test data
> testPred <- predict(iris_ctree, newdata=testData)
> table(testPred, testData$Species)

testPred      setosa versicolor virginica
  setosa          10          0         0
  versicolor       0         12         2
  virginica        0          0        14
>
> #2) Decision Trees with Package rpart
> #data("bodyfat", package = "mboost") #더이상 mboost에 존재X, TH.data 패키지에 존재O
> #install.packages("TH.data")
> library(TH.data)
필요한 패키지를 로딩중입니다: survival
필요한 패키지를 로딩중입니다: MASS

다음의 패키지를 부착합니다: ‘TH.data’

The following object is masked from ‘package:MASS’:

    geyser

> data("bodyfat")
> dim(bodyfat)
[1] 71 10
> attributes(bodyfat)
$`names`
 [1] "age"          "DEXfat"       "waistcirc"    "hipcirc"
 [5] "elbowbreadth" "kneebreadth"  "anthro3a"     "anthro3b"
 [9] "anthro3c"     "anthro4"
```

```
$row.names
 [1] "47"  "48"  "49"  "50"  "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"
[13] "59"  "60"  "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"
[25] "71"  "72"  "73"  "74"  "75"  "76"  "77"  "78"  "79"  "80"  "81"  "82"
[37] "83"  "84"  "85"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"
[49] "95"  "96"  "97"  "98"  "99"  "100" "101" "102" "103" "104" "105" "106"
[61] "107" "108" "109" "110" "111" "112" "113" "114" "115" "116" "117"

$class
[1] "data.frame"

> bodyfat[1:5,]
   age DEXfat waistcirc hipcirc elbowbreadth kneebreadth anthro3a
anthro3b
47  57  41.68     100.0   112.0          7.1         9.4     4.42     4.95
48  65  43.29      99.5   116.5          6.5         8.9     4.63     5.01
49  59  35.41      96.0   108.5          6.2         8.9     4.12     4.74
50  58  22.79      72.0    96.5          6.1         9.2     4.03     4.48
51  60  36.42      89.5   100.5          7.1        10.0     4.24     4.68
   anthro3c anthro4
47     4.50    6.13
48     4.48    6.37
49     4.60    5.82
50     3.91    5.66
51     4.15    5.91
>
> set.seed(1234)
> ind <- sample(2, nrow(bodyfat), replace=TRUE, prob=c(0.7, 0.3))
> bodyfat.train <- bodyfat[ind==1,]
> bodyfat.test <- bodyfat[ind==2,]
> #train a decision tree
> library(rpart)

다음의 패키지를 부착합니다: ‘rpart’

The following object is masked from ‘package:survival’:

    solder

> myFormula <- DEXfat ~ age + waistcirc + hipcirc + elbowbreadth +
kneebreadth
> bodyfat_rpart <- rpart(myFormula, data=bodyfat.train,
control=rpart.control(minsplit=10))
> attributes(bodyfat_rpart)
$`names`
 [1] "frame"               "where"               "call"
 [4] "terms"               "cptable"             "method"
 [7] "parms"               "control"             "functions"
[10] "numresp"             "splits"              "variable.importance"
[13] "y"                   "ordered"

$xlevels
```

```
named list()

$`class`
[1] "rpart"

>
> print(bodyfat_rpart$cptable)
          CP nsplit  rel error    xerror       xstd
1 0.67272638      0 1.00000000 1.0194546 0.18724382
2 0.09390665      1 0.32727362 0.4415438 0.10853044
3 0.06037503      2 0.23336696 0.4271241 0.09362895
4 0.03420446      3 0.17299193 0.3842206 0.09030539
5 0.01708278      4 0.13878747 0.3038187 0.07295556
6 0.01695763      5 0.12170469 0.2739808 0.06599642
7 0.01007079      6 0.10474706 0.2693702 0.06613618
8 0.01000000      7 0.09467627 0.2695358 0.06620732
>
> print(bodyfat_rpart)
n= 56

node), split, n, deviance, yval
      * denotes terminal node

 1) root 56 7265.0290000 30.94589
   2) waistcirc< 88.4 31  960.5381000 22.55645
     4) hipcirc< 96.25 14  222.2648000 18.41143
       8) age< 60.5 9   66.8809600 16.19222 *
       9) age>=60.5 5   31.2769200 22.40600 *
     5) hipcirc>=96.25 17  299.6470000 25.97000
      10) waistcirc< 77.75 6   30.7345500 22.32500 *
      11) waistcirc>=77.75 11  145.7148000 27.95818
        22) hipcirc< 99.5 3    0.2568667 23.74667 *
        23) hipcirc>=99.5 8   72.2933500 29.53750 *
   3) waistcirc>=88.4 25 1417.1140000 41.34880
     6) waistcirc< 104.75 18  330.5792000 38.09111
      12) hipcirc< 109.9 9   68.9996200 34.37556 *
      13) hipcirc>=109.9 9   13.0832000 41.80667 *
     7) waistcirc>=104.75 7  404.3004000 49.72571 *
>
> plot(bodyfat_rpart)
> text(bodyfat_rpart, use.n=T)
```

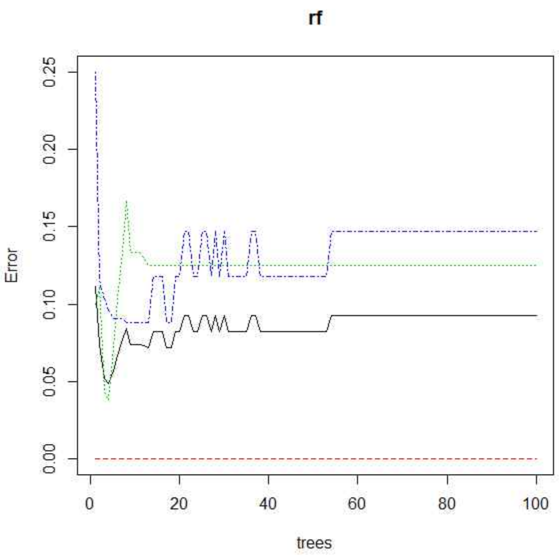


```
> opt <- which.min(bodyfat_rpart$cptable[,"xerror"])
> cp <- bodyfat_rpart#cptable[opt, "CP"]
>
> bodyfat_prune <- prune(bodyfat_rpart, cp = cp)
Error in prune.rpart(bodyfat_rpart, cp = cp) :
  (리스트) 객체는 유형 'double'로 강제형변환 될 수 없습니다
추가정보: 경고메시지(들):
In ff$complexity <= cp : 두 객체의 길이가 서로 배수관계에 있지 않습니다
```

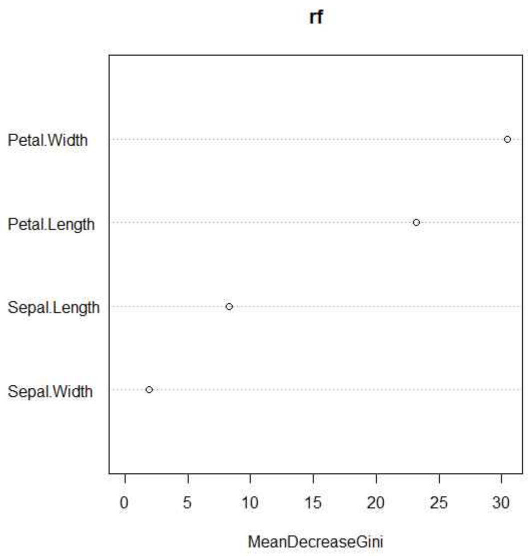
(오류 발생)

```
> #print(bodyfat_prune)
>
> #plot(bodyfat_prune)
> #text(bodyfat, use.n=T)
>
> #DEXfat_pred <- predict(bodyfat_prune, newdata=bodyfat.test)
> #xlim <- range(bodyfat$DEXfat)
> #plot(DEXfat_pred ~ DEXfat, data=bodyfat.test, xlab="Observed",
ylab="Predicted", ylim=xlim, xlim=xlim)
> #abline(a=0, b=1)
>
> #3) Random Forest
> ind <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7, 0.3))
> trainData <- iris[ind==1,]
> testData <- iris[ind==2,]
>
> #install.packages("randomForest")
> library(randomForest)
> rf <- randomForest(Species ~ ., data=trainData, ntree=100,
proximity=TRUE)
```

```
> table(predict(rf), trainData$Species)

             setosa versicolor virginica
  setosa         31          0         0
  versicolor      0         28         5
  virginica       0          4        29
>
> print(rf)

Call:
 randomForest(formula = Species ~ ., data = trainData, ntree = 100,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 2

        OOB estimate of  error rate: 9.28%
Confusion matrix:
           setosa versicolor virginica class.error
setosa         31          0         0   0.0000000
versicolor      0         28         4   0.1250000
virginica       0          5        29   0.1470588
>
> attributes(rf)
$`names`
 [1] "call"            "type"            "predicted"       "err.rate"
 [5] "confusion"       "votes"           "oob.times"       "classes"
 [9] "importance"      "importanceSD"    "localImportance" "proximity"
[13] "ntree"           "mtry"            "forest"          "y"
[17] "test"            "inbag"           "terms"

$class
[1] "randomForest.formula" "randomForest"

>
> plot(rf)
```

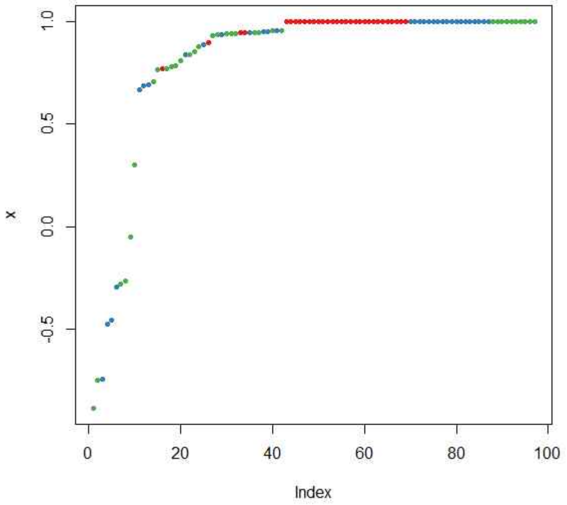


```
> importance(rf)
             MeanDecreaseGini
Sepal.Length         8.292946
Sepal.Width          1.962306
Petal.Length        23.159479
Petal.Width         30.431657
> varImpPlot(rf)
```

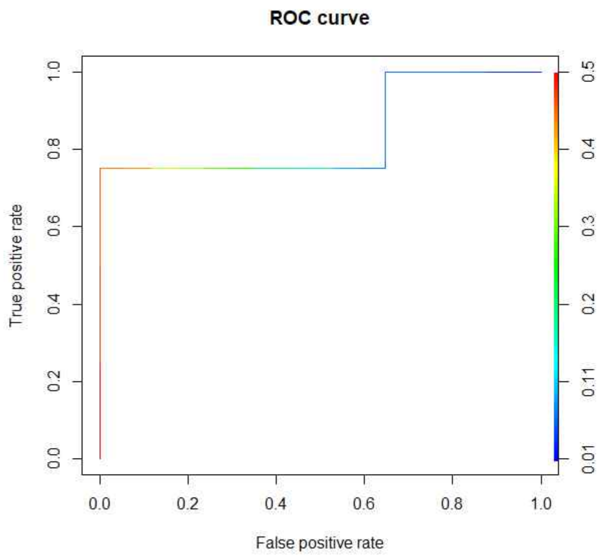


```
> irisPred <- predict(rf, newdata=testData)
> table(irisPred, testData$Species)
```

```
irisPred      setosa  versicolor  virginica
  setosa          19           0          0
  versicolor       0          18          0
  virginica        0           0         16
> plot(margin(rf, testData$Species))
```



```
> #4) ROCR 패키지로 성과분석
> #install.packages("party")
> #install.packages("ROCR")
> library(rpart)
> x <- kyphosis[sample(1:nrow(kyphosis), nrow(kyphosis), replace=F), ]
> x.train <- kyphosis[1:floor(nrow(x)*.75), ]
> x.evaluate <- kyphosis[(floor(nrow(x)*.75)+1):nrow(x), ]
> library(party)
> x.model <- cforest(Kyphosis ~ Age + Number + Start, data=x.train,
+   control = cforest_unbiased(mtry=3))
> #x.model <- ctree(Kyphosis ~ Age + Number + Start, data=x.train)
> #plot(x.model)
> x.evaluate$prediction <- predict(x.model, newdata=x.evaluate)
> x.evaluate$correct <- x.evaluate$prediction == x.evaluate$Kyphosis
> print(paste("% of predicted classifications correct",
mean(x.evaluate$correct)))
[1] "% of predicted classifications correct 0.80952380952381"
> x.evaluate$probabilities <- 1-unlist(treeresponse(x.model,
newdata=x.evaluate), use.names=F)[seq(1, nrow(x.evaluate)*2,2)]
> library(ROCR)
> pred <- prediction(x.evaluate$probabilities, x.evaluate$Kyphosis)
> perf <- performance(pred, "tpr", "fpr")
> plot(perf, main="ROC curve", colorize=T)
```


ROC curve
True positive rate
False positive rate