Decision_tree_Kyphosis.R

Sai

Sun Dec 25 21:16:28 2016

```
# library for Classification & Regression Trees
library(xtable)
# library for Ionosphere data
library(rpart)
library(gam)
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
## Loaded gam 1.12
```

```
data("kyphosis")
#Data on Children who have had Corrective Spinal Surgery
kyp<-kyphosis
str(kyp)
```

```
## 'data.frame':    81 obs. of  4 variables:
##  $ Kyphosis: Factor w/ 2 levels "absent","present": 1 1 2 1 1 1 1 1 1 2 ...
##  $ Age     : int  71 158 128 2 1 1 61 37 113 59 ...
##  $ Number  : int  3 3 4 5 4 2 2 3 2 6 ...
##  $ Start   : int  5 14 5 1 15 16 17 16 16 12 ...
```

```
v<-kyp$Kyphosis
table(v)
```

```
## v
##  absent present
##      64      17
```

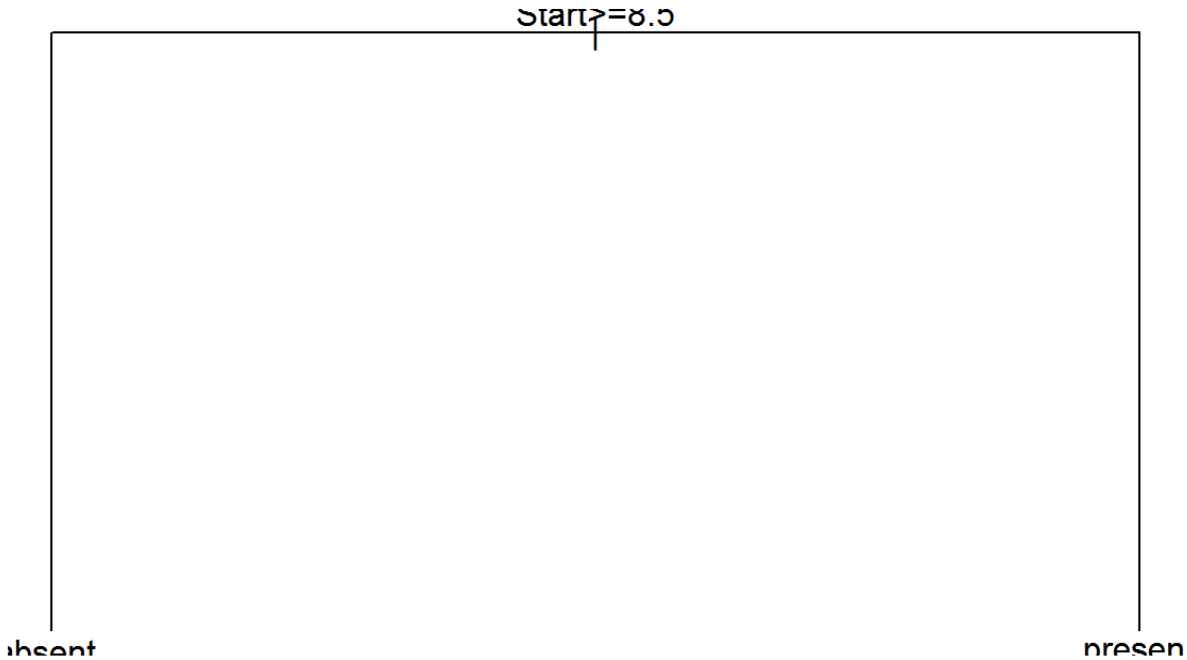
```
#set seed to ensure reproducible results
set.seed(250)
#spliting into training and test data sets in 3:1 ratio
kyp[,'train'] <- ifelse(runif(nrow(kyp))<0.75,1,0)
#separate training and test sets
train_kyp <- kyp[kyp$train==1,]
test_kyp <- kyp[kyp$train==0,]
#get column index of train flag
kyp_trainColNum <- grep('train',names(train_kyp))
str(train_kyp)
```

```
## 'data.frame':    64 obs. of  5 variables:
##  $ Kyphosis: Factor w/ 2 levels "absent","present": 1 2 1 1 2 1 1 1 1 1 ...
##  $ Age     : int  71 128 37 113 82 148 18 1 168 1 ...
##  $ Number  : int  3 4 3 2 5 3 5 4 3 3 ...
##  $ Start   : int  5 5 16 16 14 16 2 12 18 16 ...
##  $ train   : num  1 1 1 1 1 1 1 1 1 1 ...
```

```
str(test_kyp)
```

```
## 'data.frame':    17 obs. of  5 variables:
##  $ Kyphosis: Factor w/ 2 levels "absent","present": 1 1 1 1 1 2 2 1 1 1 ...
##  $ Age     : int  158 2 1 1 61 59 105 9 100 31 ...
##  $ Number  : int  3 5 4 2 2 6 6 5 3 3 ...
##  $ Start   : int  14 1 15 16 17 12 5 13 14 16 ...
##  $ train   : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
#Obtaining the train and test data set
#remove train flag column from train and test sets
train_kyp <- train_kyp[,-kyp_trainColNum]
test_kyp <- test_kyp[,-kyp_trainColNum]
#Get column index of predicted variable in dataset
typeColNum_kyp <- grep('Kyphosis',names(kyp))
#Constructing the required Decision tree model
rpart_model_kyp <- rpart(Kyphosis~.,data = train_kyp, method= 'class')
# Plotting the tree
plot(rpart_model_kyp)
text(rpart_model_kyp)
```




```
summary(rpart_model_kyp)
```

```
## Call:
## rpart(formula = Kyphosis ~ ., data = train_kyp, method = "class")
##   n= 64
##
##     CP nsplit rel error xerror     xstd
## 1 0.20      0       1.0      1 0.225924
## 2 0.01      1       0.8      1 0.225924
##
## Variable importance
##  Start Number
##     85     15
##
## Node number 1: 64 observations,    complexity param=0.2
##   predicted class=absent   expected loss=0.234375  P(node) =1
##     class counts:    49    15
##    probabilities: 0.766 0.234
##   left son=2 (47 obs) right son=3 (17 obs)
##   Primary splits:
##       Start  < 8.5  to the right, improve=5.797286, (0 missing)
##       Age    < 39.5 to the left,  improve=2.180267, (0 missing)
##       Number < 6.5  to the left,  improve=1.785793, (0 missing)
##   Surrogate splits:
##       Number < 6.5  to the left,  agree=0.781, adj=0.176, (0 split)
##
## Node number 2: 47 observations
##   predicted class=absent   expected loss=0.106383  P(node) =0.734375
##     class counts:    42     5
##    probabilities: 0.894 0.106
##
## Node number 3: 17 observations
##   predicted class=present  expected loss=0.4117647  P(node) =0.265625
##     class counts:     7    10
##    probabilities: 0.412 0.588
```

```
#Checking how good the model is
rpart_predict_kyp<- predict(rpart_model_kyp,test_kyp[,-typeColNum_kyp],type='class'
)
mn_kyp <- mean(rpart_predict_kyp==test_kyp$Kyphosis)
mn_kyp
```

```
## [1] 0.8823529
```

```
# Constructing the confusion matrix to find out the efficiency of the model
table(pred=rpart_predict_kyp,true=test_kyp$Kyphosis)
```

```
##          true
## pred      absent present
##   absent      14       1
##   present      1       1
```

```
# Applying the cost-complexity pruning
printcp(rpart_model_kyp)
```

```
##
## Classification tree:
## rpart(formula = Kyphosis ~ ., data = train_kyp, method = "class")
##
## Variables actually used in tree construction:
## [1] Start
##
## Root node error: 15/64 = 0.23438
##
## n= 64
##
##     CP nsplit rel error xerror    xstd
## 1 0.20      0       1.0      1 0.22592
## 2 0.01      1       0.8      1 0.22592
```

```
#Finding index of CP with lowest xerror
opt_kyp <- which.min(rpart_model_kyp$cptable[,'xerror'])
#Finding the values of CP
cp_kyp <- rpart_model_kyp$cptable[opt_kyp, 'CP' ]
cp_kyp
```

```
## [1] 0.2
```

```
# Pruning not required
```