Decision Trees

Ethan Duan

May 7, 2019

Decision Tress

```
install.packages('rpart')  library(rpart)
```

Using the build in data Kyphosis

```
# this is built in R
str(kyphosis)
```

```
## '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 ...
```

```
head(kyphosis)
```

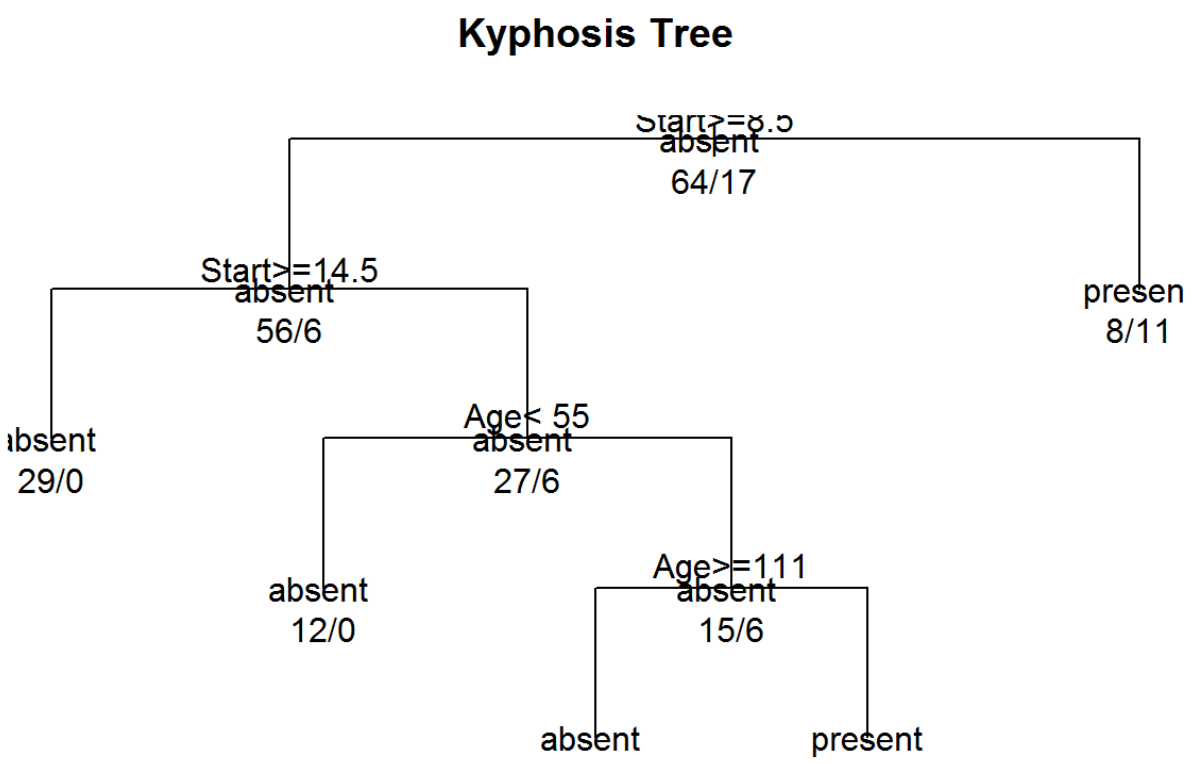
```
##   Kyphosis Age Number Start
## 1   absent  71      3     5
## 2   absent 158      3    14
## 3  present 128      4     5
## 4   absent   2      5     1
## 5   absent   1      4    15
## 6   absent   1      2    16
```

```
# Build Tree Model
tree <- rpart(Kyphosis ~ ., method = 'class', data = kyphosis)

# many functions to viusualzie
printcp(tree)
```

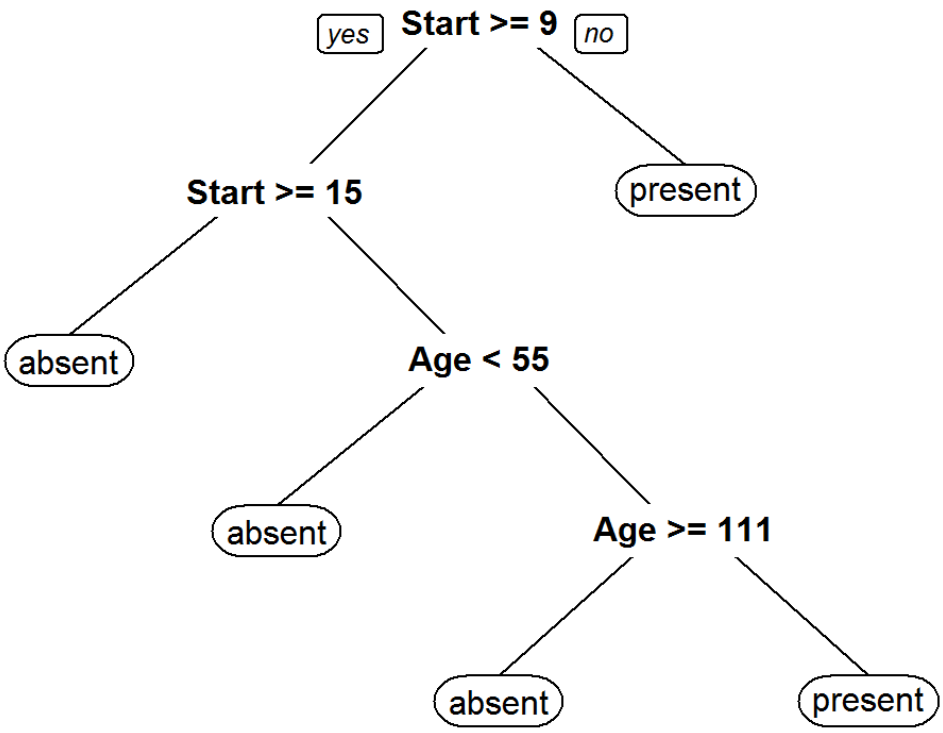
```
##
## Classification tree:
## rpart(formula = Kyphosis ~ ., data = kyphosis, method = "class")
##
## Variables actually used in tree construction:
## [1] Age   Start
##
## Root node error: 17/81 = 0.20988
##
## n= 81
##
##         CP nsplit rel error  xerror    xstd
## 1 0.176471      0   1.00000 1.00000 0.21559
## 2 0.019608      1   0.82353 0.88235 0.20565
## 3 0.010000      4   0.76471 0.88235 0.20565
```

```
plot(tree, uniform = T, main='Kyphosis Tree')
text(tree, use.n = T, all=T)
```



There's a easier way `install.packages('rpart.plot')`

```
# install.packages('rpart.plot')
library(rpart.plot)
prp(tree)
```



Random Forest

```
# install.packages('randomForest')
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
rf.model <- randomForest(Kyphosis~. , data = kyphosis)
print(rf.model)
```

```
##
## Call:
##  randomForest(formula = Kyphosis ~ ., data = kyphosis)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 19.75%
## Confusion matrix:
##         absent present class.error
## absent      59       5   0.0781250
## present     11       6   0.6470588
```

```
rf.model$predicted
```

```
##       1       2       3       4       5       6       7       8       9
## present  absent present  absent  absent  absent  absent  absent  absent
##      10      11      12      13      14      15      16      17      18
##  absent  absent  absent  absent  absent  absent  absent  absent  absent
##      19      20      21      22      23      24      25      26      27
##  absent  absent  absent present  absent present  absent  absent  absent
##      28      29      30      31      32      33      34      35      36
##  absent  absent  absent  absent  absent  absent  absent  absent  absent
##      37      38      39      40      41      42      43      44      45
##  absent  absent  absent present  absent  absent present  absent  absent
##      46      47      48      49      50      51      52      53      54
##  absent  absent  absent  absent  absent present  absent  absent  absent
##      55      56      57      58      59      60      61      62      63
##  absent  absent  absent present present  absent  absent present  absent
##      64      65      66      67      68      69      70      71      72
##  absent  absent  absent  absent  absent  absent  absent  absent  absent
##      73      74      75      76      77      78      79      80      81
##  absent  absent  absent  absent  absent  absent  absent present  absent
## Levels: absent present
```

```
rf.model$ntree
```

```
## [1] 500
```

```
rf.model$confusion
```

```
##         absent present class.error
## absent      59       5   0.0781250
## present     11       6   0.6470588
```