

Decision Tree Tutorial Part 1 - IRIS

John Kelly

11 February 2019

Use C5.0 Algorithm on the IRIS Dataset

Step 1: Load required packages and libraries

```
library(C50)
data(iris)
np <- ceiling(0.1*nrow(iris))
np
```

```
## [1] 15
```

Step 2: Set Seed and create training and test datasets

```
set.seed(3)
test.index <- sample(1:nrow(iris), np)
iris.test <- iris[test.index, ]
iris.train <- iris[-test.index, ]
```

Step 3: Run C5.0 Algorithm

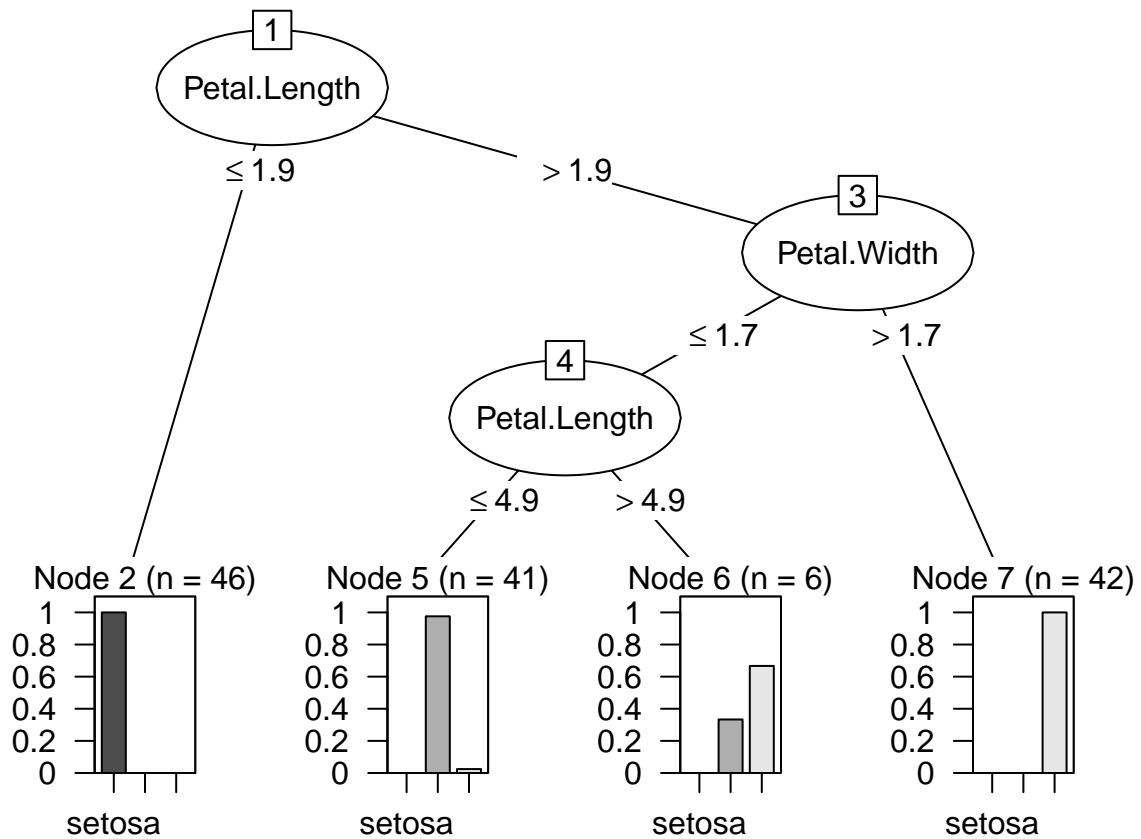
```
c <- C5.0Control(subset = FALSE,
                 bands = 0,
                 winnow = FALSE,
                 noGlobalPruning = FALSE,
                 CF = 0.25,
                 minCases = 2,
                 fuzzyThreshold = FALSE,
                 sample = 0,
                 seed = sample.int(4096, size = 1) -1L,
                 earlyStopping = TRUE
)
iris_treeModel <- C5.0(x = iris.train[, -5], y = iris.train$Species, control = c)
summary(iris_treeModel)
```

```
##
## Call:
## C5.0.default(x = iris.train[, -5], y = iris.train$Species, control = c)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Feb 11 17:52:16 2019
## -----
##
## Class specified by attribute `outcome'
##
## Read 135 cases (5 attributes) from undefined.data
##
## Decision tree:
```

```

##
## Petal.Length <= 1.9: setosa (46)
## Petal.Length > 1.9:
##   :...Petal.Width > 1.7: virginica (42)
##     Petal.Width <= 1.7:
##       :...Petal.Length <= 4.9: versicolor (41/1)
##         Petal.Length > 4.9: virginica (6/2)
##
##
## Evaluation on training data (135 cases):
##
##      Decision Tree
##      -----
##      Size      Errors
##
##      4      3( 2.2%)  <<
##
##
##      (a)  (b)  (c)  <-classified as
##      ----  ----  ----
##      46          (a): class setosa
##           40    2  (b): class versicolor
##           1    46  (c): class virginica
##
##
## Attribute usage:
##
## 100.00% Petal.Length
##  65.93% Petal.Width
##
##
## Time: 0.0 secs
plot(iris_treeModel)

```



Step 4: Test Output and Accuracy

```
test.output <- predict(iris_treeModel, iris.test[, -5], type = "class")
n <- length(test.output)
number = 0
for ( i in 1:n){
  if(test.output[i] == iris.test[i, 5])
  {
    number=number+1}
}
test.accuracy = number/n*100
test.accuracy
```

```
## [1] 93.33333
```