Decision Tree

Ravi Kumar Tiwari

13 July 2016

Classification

Tree based algorithm

Tree based algorithms are used to model both quantitative and qualitative response.

3.1 Decision Tree

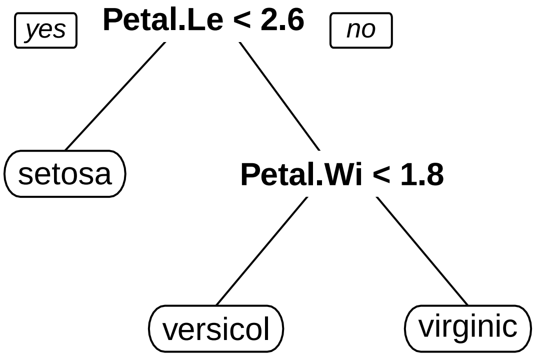
This approach involves dividing the predictor variables into smaller regions (using decision rules that combine to form a decision tree) such that response within these regions are (nearly) homogeneous. The response corresponding to a given observation is determined based on the region it falls into. For qualitative response, it is assigned the most dominant class of the region. For quantitative response, it is assigned the mean of the response values in the region

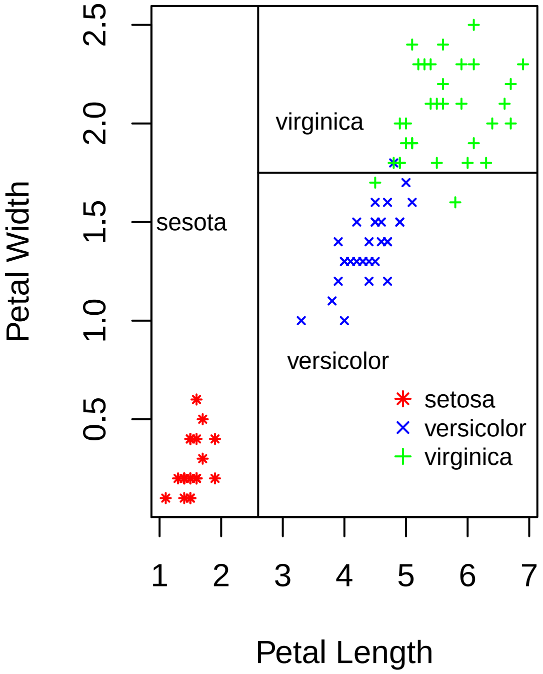
Example

Decision tree showing decision rules to determine the species of a flower based on its Sepal and Petal measurments. On the right, the regions that results from those rules are shown. We look at the data, first

```
iris[c(1,100,150),]
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 100          5.7         2.8          4.1         1.3 versicolor
## 150          5.9         3.0          5.1         1.8  virginica
```





Code to create decision tree model

```
## Load the required libraries
library(rpart)
library(rpart.plot)

## create data partition
set.seed(1)
inTrain <- sample(c(TRUE, FALSE), size = nrow(iris), replace = TRUE, prob = c(0.6,0.4))
trainData <- iris[inTrain,]
testData <- iris[!inTrain,1:4]
testClass <- iris[!inTrain,5]

## create the tree model and make prediction using the tree model
treeModel <- rpart(Species ~ ., data = trainData)
predClass <- predict(treeModel, newdata = testData, type = "class")

# Plot the tree
#rpart.plot(treeModel, type = 0)
```

code for making prediction using the decision tree

```
predTrainClass <- predict(treeModel, newdata = trainData, type = "class")
predTestClass <- predict(treeModel, newdata = testData, type = "class")
```

Evaluating the performance of the decision tree

```
## Training Data
table(predTrainClass, trainData$Species)  # Confusion Matrix
```

```
## 
## predTrainClass setosa versicolor virginica
##     setosa         27          0         0
##     versicolor      0         29         2
##     virginica       0          1        30
```

```
mean(predTrainClass == trainData$Species) # Prediction Accuracy
```

```
## [1] 0.9662921
```

```
## Test Data
table(predTestClass, testClass)           # Confusion Matrix
```

```
##               testClass
## predTestClass setosa versicolor virginica
##    setosa         23          0         0
##    versicolor      0         20         3
##    virginica       0          0        15
```

```
mean(predTestClass == testClass)
```

```
## [1] 0.9508197
```