Decision Trees and Random Forest classifiers

R Mofidi

24/06/2020

Decision tree Analysis

Decision Trees are a class of tree like graph algorithms. A decision tree uses this tree like structure to illustrate the possible decisions and their consequences. It is one way of combining an algorithm and a decision support tool as well as machine learning paradigm. A decision tree consists of 3 constituents (no pun intended), a node which represents an attribute and a branch which represents the consequences of that attribute. Attributes can occur as a result of an active decision (decision nodes usually marked as a square), a chance node which is the consequences of an event outside of the decision makers' control and an end node which denotes one of the possible final outcomes of the decision tree algorithm. The findings of decision tree can be illustrated in a compact and easy to follow format using an influence diagram which describes the relationship between actions and consequences (1).

Decision tree classifiers utilise the topography of a decision tree to map observations about a target item to conclusions about the value of the target item. Where the target value has a finite set of values (ordinal or nominal). The process is known as a classification tree. If the Target value is a continuous variable it is called regression tree. The umbrella term Classification and Regression Tree (CART) is used to group both processes. It was devised by the distinguished American statistician Leo Breiman. The algorithms work from top down by choosing the variables which best splits a set of items into the intended classes. The algorithms created are simple to understand and require little preparation, It is possible to use categorical as well as continuous variables and create a white box model which is transparent to the user. It is possible to validate using statistical analysis such as a confusion matrix or receiver operator characteristic test.

The process involves a series of simple steps the first of which involves identifying and using the variable which best separates the dataset in accordance with the outcome of interest. This creates 2 leaves or childeren nodes. the remaining dataset in each leaves is divided in a similar manner until the groups are either too small or pure i.e. contain only a single outcome (pure).

The following example is created using a commonly encountered database in R known as the Iris database. This databse uses the width and length of the petal and sepal of iris floweres to classify the iris flowers into the 3 different species: 1- Setosa 2- Versicolor 3- Virginica

The follollowing code describes how decision trees (classification and Regression Trees) can be developed in R.

Installing the appropriate libraries and datasets

```r
data(iris)
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
library(ggplot2)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Separating the training and test datasets

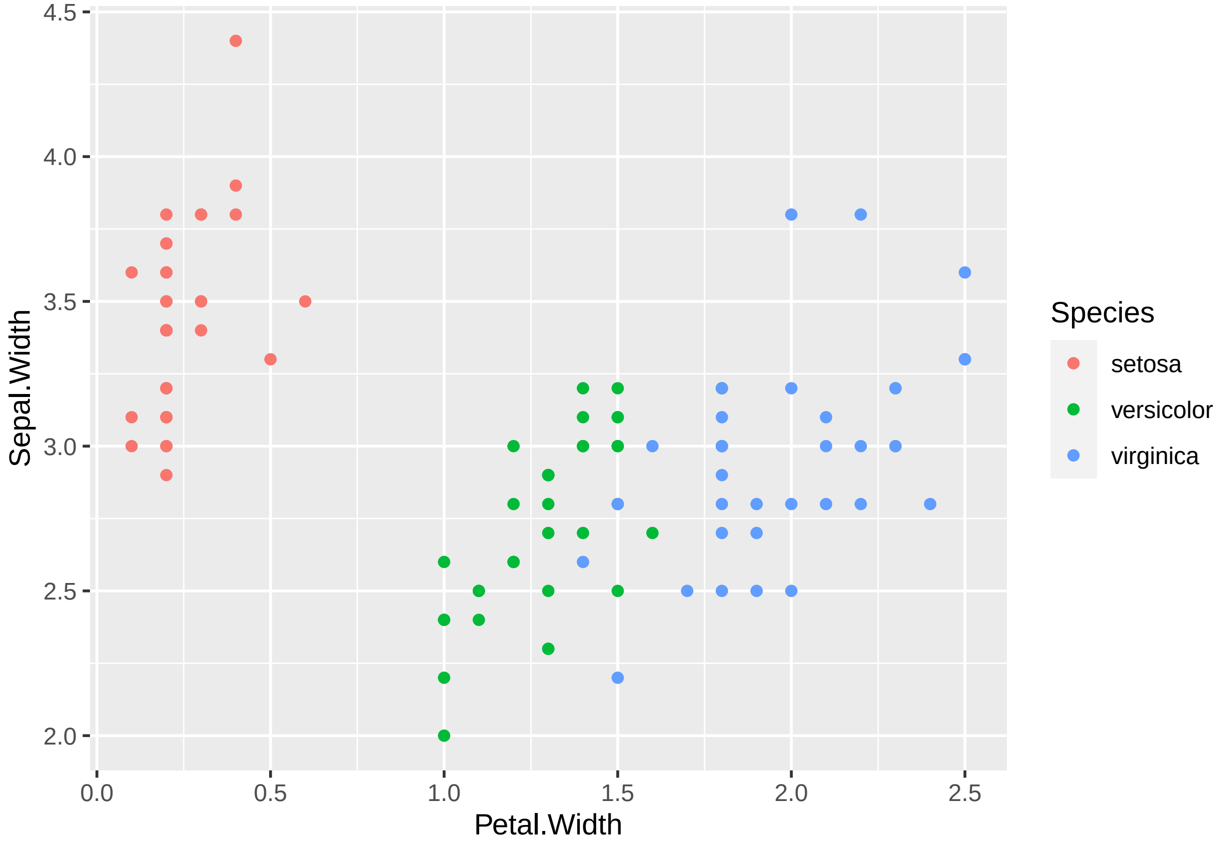
This creates a training dataset for developing the tree and a testing dataset for cross validation:

```
inTrain3<- createDataPartition(y=iris$Species, p=0.7, list=FALSE)
trainingDS<- iris[inTrain3,]
testingDS<- iris[-inTrain3,]
dim (trainingDS); dim(testingDS)
```

```
## [1] 105   5
```

```
## [1] 45  5
```

Viewing the data separation



Developing the decision tree "rpart" model.

```
library(caret)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lattice)
modfit<- train(Species~.,method="rpart", data=trainingDS)
print(modfit$finalModel)
```

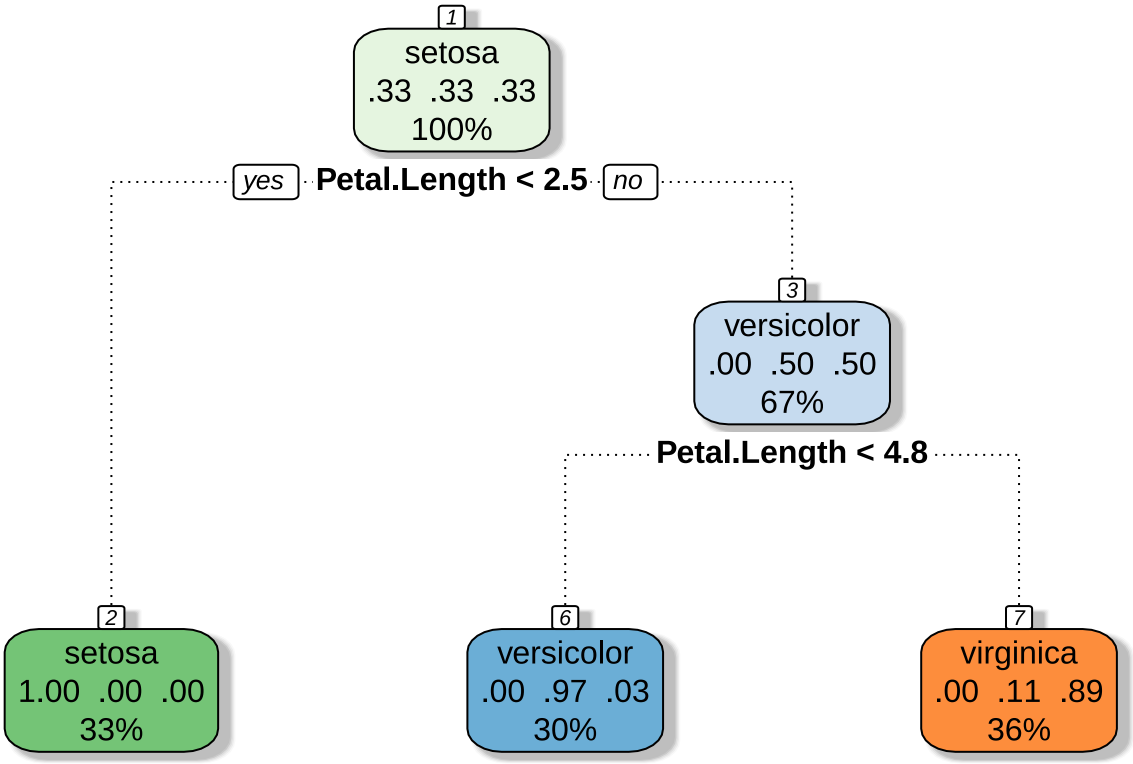
```
## n= 105
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 105 70 setosa (0.3333333 0.3333333 0.3333333)
##   2) Petal.Length< 2.45 35  0 setosa (1.0000000 0.0000000 0.0000000) *
##   3) Petal.Length>=2.45 70 35 versicolor (0.0000000 0.5000000 0.5000000)
##     6) Petal.Length< 4.75 32  1 versicolor (0.0000000 0.9687500 0.0312500) *
##     7) Petal.Length>=4.75 38  4 virginica (0.0000000 0.1052632 0.8947368) *
```

The Classification Tree

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```



Rattle 2020−Jun−24 23:35:30 Rachael

Crossvalidation of the decision tree model

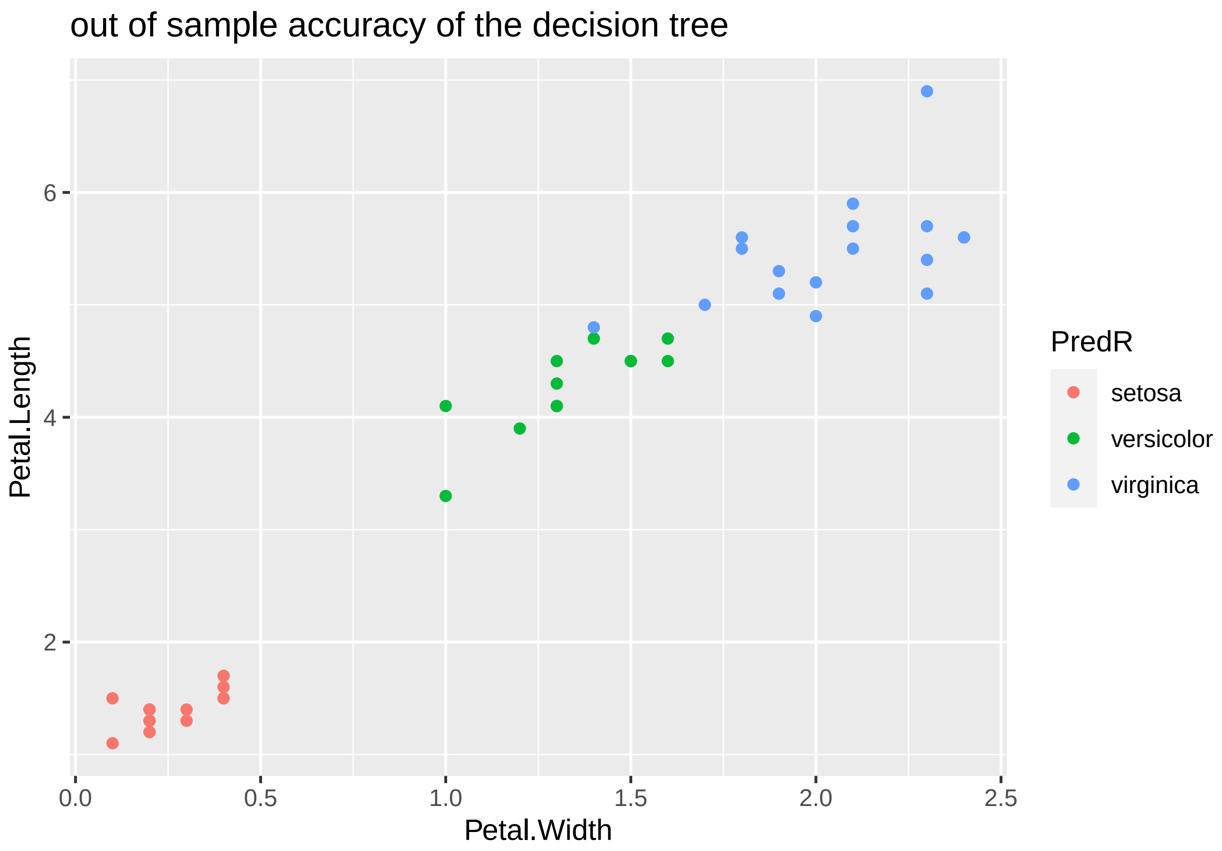
The process of crossvalidation is performed using the testing dataset i.e. the 30% if the dataset which was set aside from the training process and used to assess the ability of the decision tree to make out of sample predictions:

```r
predict(modfit, newdata=testingDS)
```

```
##  [1] setosa     setosa     setosa     setosa     setosa     setosa    
##  [7] setosa     setosa     setosa     setosa     setosa     setosa    
## [13] setosa     setosa     setosa     versicolor versicolor versicolor
## [19] versicolor versicolor virginica  virginica  versicolor versicolor
## [25] versicolor versicolor versicolor versicolor versicolor versicolor
## [31] virginica  virginica  virginica  virginica  virginica  virginica 
## [37] virginica  virginica  virginica  virginica  virginica  virginica 
## [43] virginica  virginica  virginica 
## Levels: setosa versicolor virginica
```

```r
PredR<- predict(modfit, newdata=testingDS)
summary(PredR)
```

```
##     setosa versicolor  virginica 
##         15         13         17
```



Decision trees are used extensively as classifiers and decision aids in many areas including healthcare. They provide a mixture of simplicity, utility and functionality. For many uses classification and regression trees are more than sufficient inluding many healthcare decision support models (2, 3)

Random Forests and Ensamble Classifiers

Ensemble Classifiers a consist of multiple classifiers which by themselves may be weak (i.e. have low predictive ability) but combining them into an ensemble improves their predictive ability significantly. A similar concept exists in statistics with the difference that machine learning the ensembles contain a finite set of constituent algorithms whilst a statistical ensemble is infinite. The design of this class of algorithm is based on ensemble theory which states that a trained ensemble of algorithms can represent a single supervised trained algorithm. In general an ensemble algorithm functions better if it contains a diverse set of constituent algorithms.

In fact the individual constituents of the ensemble do not necessarily have to be weak classifiers, being part of the ensemble means that they do not need to be complex in structure which in turn helps protect against over-fitting (over-training). They key to a successful ensemble is "stochastic discrimination" which is discussed later. There are a number of classes of ensemble classifiers in existence they include: • Bayes optimal classifiers • Bootstrap aggregating methods such as Random forest classifiers • Boosting: such as Adaboost

In this paper we discuss random forest classfiers

Random Forest classifiers

An example of ensemble classifiers is the random forest. Random forests were first designed by the American data scientist Tin Kam Ho in the 1990s. Random forests are made up of (large) number of tree classifiers which are added together in an ensemble in order to improve their classification ability. It is a bootstrap aggregating method which means each of these decision tree classifiers contributes to the eventual decision with an equal weight. (4)

Each tree acts as a weak classifier and together a large number of trees form a random forest and diversity of classifiers within the ensemble is the key to its performance.

If you remember in order to develop an ensemble classifier two assumptions will need to be fulfilled. The 2nd of these two assumptions is that each weak classifier should make its choice independently of the other classifiers. This concept is called "Stochastic discrimination". In order to fulfil this assumption each decision tree needs to undergo a different training process using the same training data. This is a dilemma as often the training data is scarce. One way of getting around this is randomisation (random selection) of training data he used to train each tree classifier. In order to develop a random forest during training process a number of hyperplanes are selected at random and are trained using a random sub set of they are available training data. The final classification is performed using a majority vote

The following is an example of a small random forest classifier created using the iris database described above:

###loading the required packages and database

```r
data(iris); library(ggplot2); library(randomForest);library(caret)
```

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

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:rattle':
##
##     importance
## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

###partitioning the data into training and testing sets

```
inTrainRF<-createDataPartition(y=iris$Species, p=0.7, list=FALSE)
training4<-iris[inTrainRF,]
testing4<-iris[-inTrainRF,]
```

Training the random forest classifier:

This involves setting the output variable as the Species and the rest of the variables as input variables.

```
modFit<- train(Species~.,data=training4, method="rf", prox=TRUE)
modFit
```

```
## Random Forest
##
## 105 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 105, 105, 105, 105, 105, 105, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.9547200  0.9310645
##   3     0.9516133  0.9263440
##   4     0.9381133  0.9060676
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

As you can see the in sample accuracy od the data is excellent and it overperforms most linear data models as well as Classification and regression trees. clearly concerns regarding over-training (overfitting the data exists. This is why cross validation and data visualization is important. This is what the testing sample "testing4" is used for. Random forests can be complicated in order to understand their underlying anatomy, it is possible to view the consituent trees making up the random forest classifier:

```
getTree(modFit$finalModel,k=2)
```

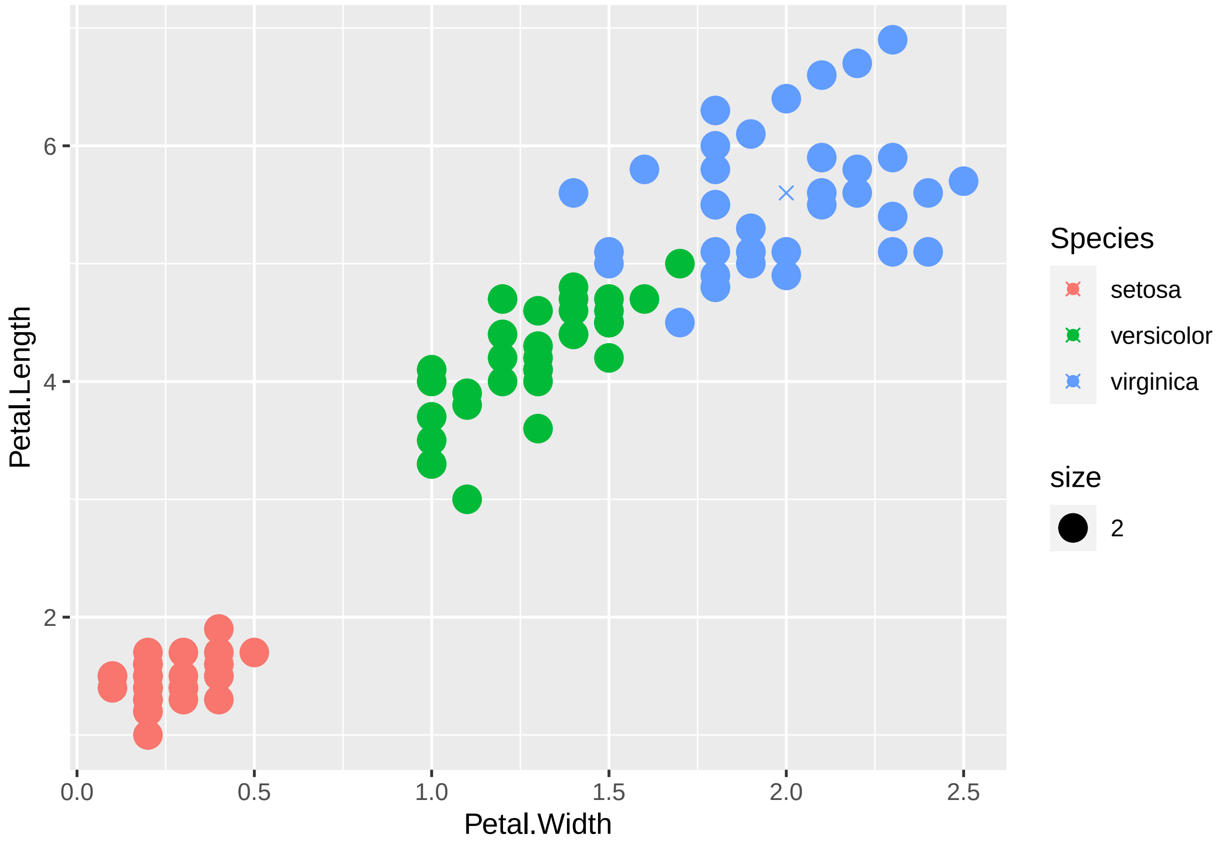
```
##   left daughter right daughter split var split point status prediction
## 1             2              3         3        4.85      1          0
## 2             4              5         3        2.45      1          0
## 3                6              7         1        6.60      1          0
## 4                0              0         0        0.00     -1          1
## 5                0              0         0        0.00     -1          2
## 6                0              0         0        0.00     -1          3
## 7                8              9         3        5.05      1          0
## 8                0              0         0        0.00     -1          2
## 9                0              0         0        0.00     -1          3
```

```
irisP<- classCenter(training4[,c(3,4)], training4$Species, modFit$finalModel$prox)
irisP<-as.data.frame(irisP);irisP$Species<- rownames(irisP)
P<- qplot(Petal.Width, Petal.Length, col=Species, size=2,Shape=4,data=training4)
```

Visualising the classifier

```
## Warning: Ignoring unknown parameters: Shape
```

```
P+geom_point(aes(x=Petal.Width, y=Petal.Length, col=Species), size=2, shape=4, data=irisP)
```

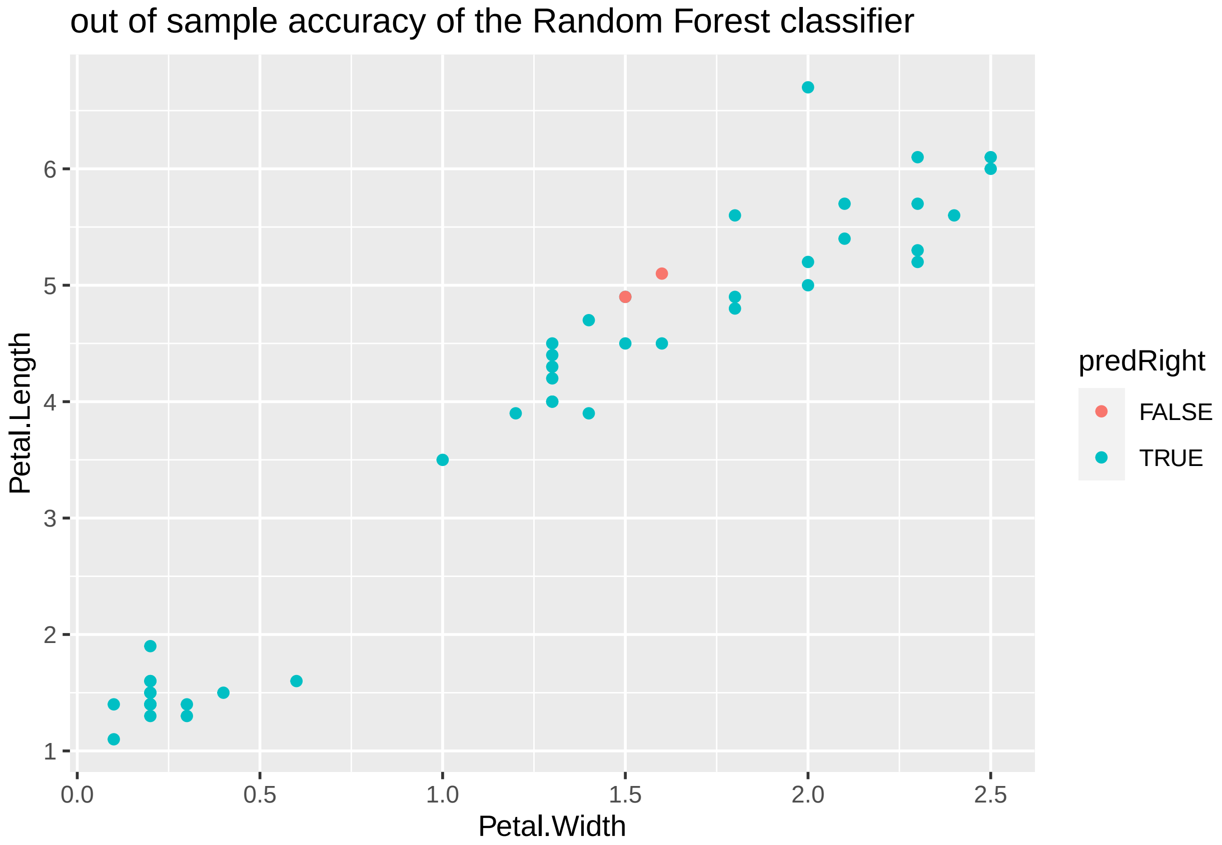


Making the prediction in the testing sample

This is the process of cross validation of the classifier.

```
pred<-predict(modFit, testing4)
testing4$predRight<-pred == testing4$Species
table(pred,testing4$Species)
```

```
##
## pred         setosa versicolor virginica
##   setosa         15          0         0
##   versicolor      0         13         0
##   virginica       0          2        15
```



Random forests are amongst the most powerful classifiers available and as you can see at a very basic level they are not too difficult to develop in R. Despite their complexity which can be a problem when dealing with large data sets, they are considered white-box classifiers. Well implemented random forests are more resiliant to overfitting Random forests have less variance than single decision trees.

Complexity when large ensambles are being used to analyse large datasets. This can be time consuming and complicated process. Because of this random forests are used less often than decision trees in healthcare informatics(5).

References

1- Lewis RJ. An introduction to classification and regression tree (CART) analysis. InAnnual meeting of the society for academic emergency medicine in San Francisco, California 2000, (Vol. 14).

2- McBride OM, Mofidi R, Griffiths GD, Dawson AR, Chalmers RT, Stonebridge PA. Development of a decision tree to streamline infrainguinal vein graft surveillance. Annals of vascular surgery;36:182-9.

3- Mofidi R, McBride OM, Green BR, Gatenby T, Walker P, Milburn S. Validation of a decision tree to streamline infrainguinal vein graft surveillance. Annals of vascular surgery 2017;40:216-22.

4- Prasad AM, Iverson LR, Liaw A. Newer classification and regression tree techniques: bagging and random forests for ecological prediction. Ecosystems. 2006;9(2):181-99.

5- Masetic Z, Subasi A. Congestive heart failure detection using random forest classifier. Computer methods and programs in biomedicine. 2016;130:54-64.