Different Random Forest Packages in R

Thiyanga Talagala

14 November 2017

```
library(knitr)
opts_chunk$set(tidy = TRUE)
```

randomForest Package

Data pre-processing

Split iris data to Training data and testing data

```
ind <- sample(2,nrow(iris),replace=TRUE,prob=c(0.7,0.3))
trainData <- iris[ind==1,]
testData <- iris[ind==2,]
```

1. Load randomForest

```
library(randomForest)
```

```
## randomForest 4.6-12
```

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

2. Generate Random Forest learning treee

```
iris_rf <- randomForest(Species~.,data=trainData,ntree=100,proximity=TRUE)
table(predict(iris_rf),trainData$Species)
```

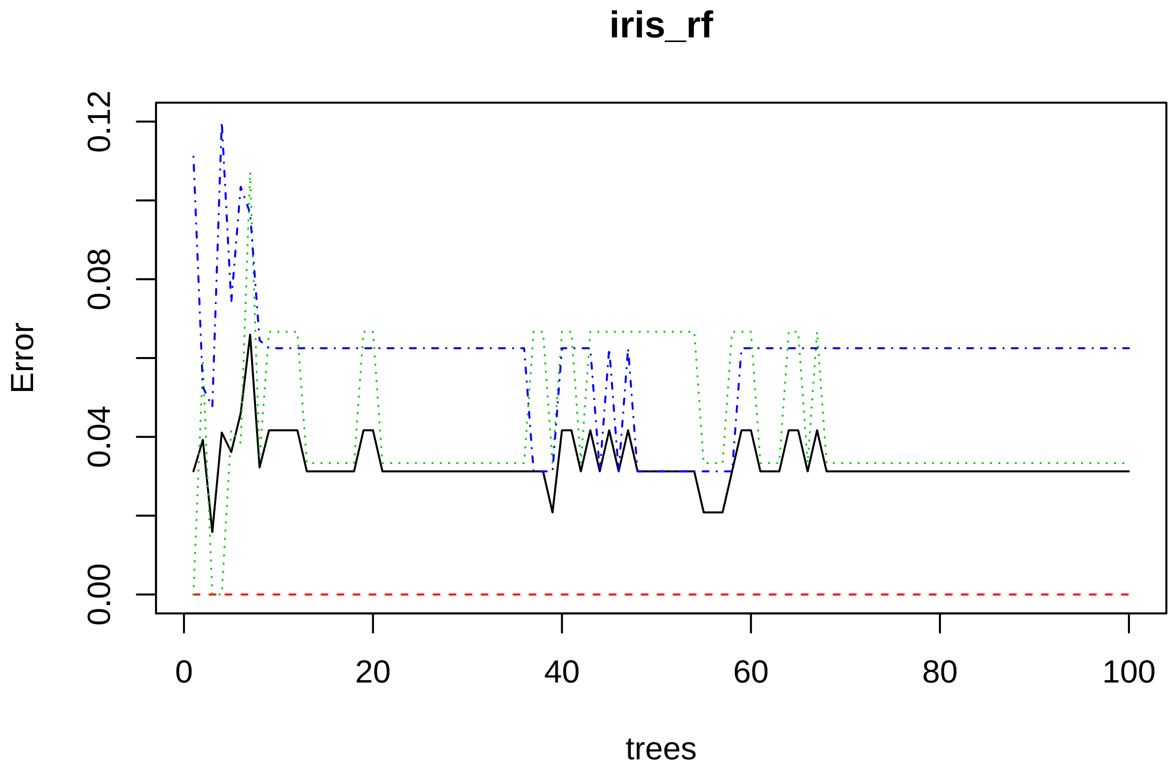
```
##
##               setosa versicolor virginica
##    setosa         34          0         0
##    versicolor      0         29         2
##    virginica       0          1        30
```

3. Try to print Random Forest model and see the importance features

```
print(iris_rf)
```

```
##
## Call:
##  randomForest(formula = Species ~ ., data = trainData, ntree = 100,      proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 3.12%
## Confusion matrix:
##            setosa versicolor virginica class.error
## setosa         34          0         0  0.00000000
## versicolor      0         29         1  0.03333333
## virginica       0          2        30  0.06250000
```

```
plot(iris_rf)
```

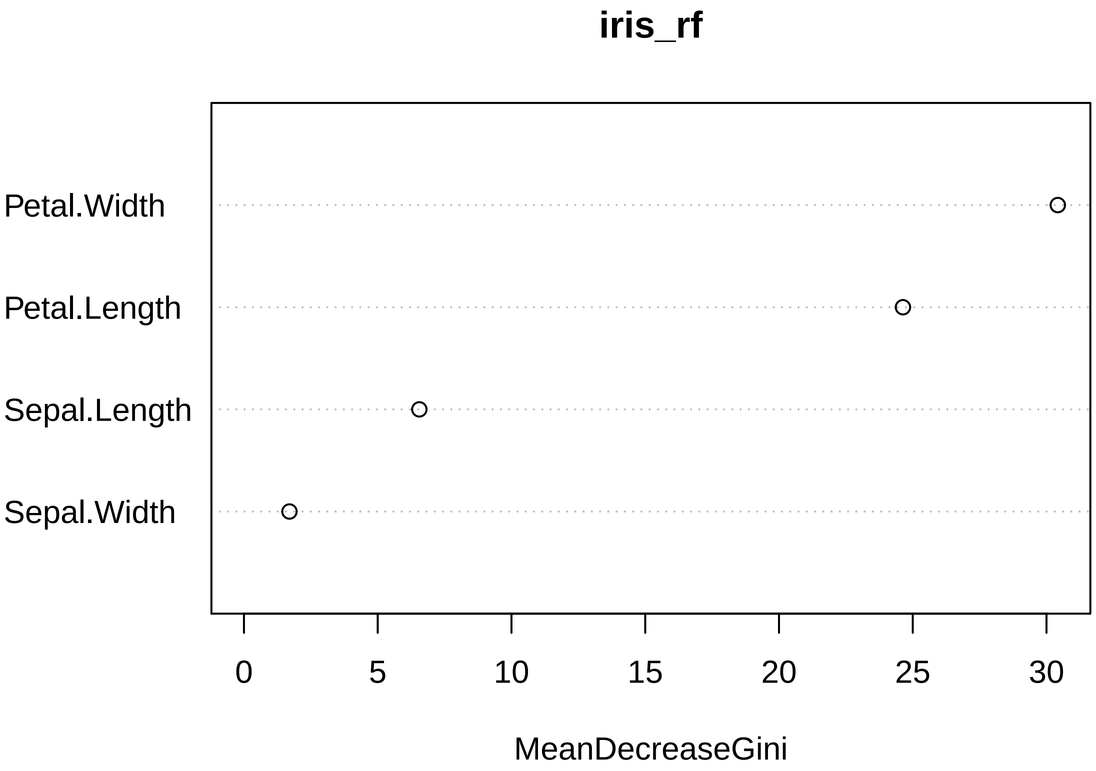


4. Assessing model fit

```
importance(iris_rf)
```

```
##              MeanDecreaseGini
## Sepal.Length         6.555624
## Sepal.Width          1.700689
## Petal.Length        24.633981
## Petal.Width         30.422414
```

```
varImpPlot(iris_rf)
```



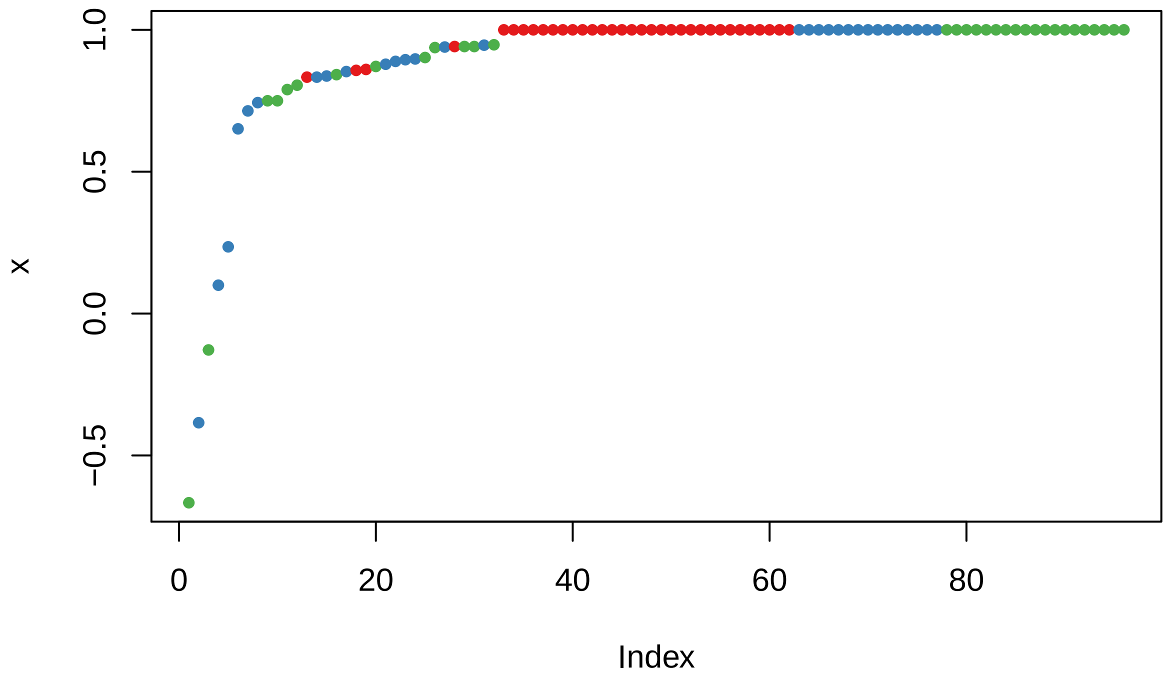
5. Predict the class labels for test data

```
irisPred<-predict(iris_rf,newdata=testData)
table(irisPred, testData$Species)
```

```
##
## irisPred     setosa versicolor virginica
##   setosa         16          0         0
##   versicolor      0         18         2
##   virginica       0          2        16
```

6. Try to see the margin, positive or negative, if positif it means correct classification

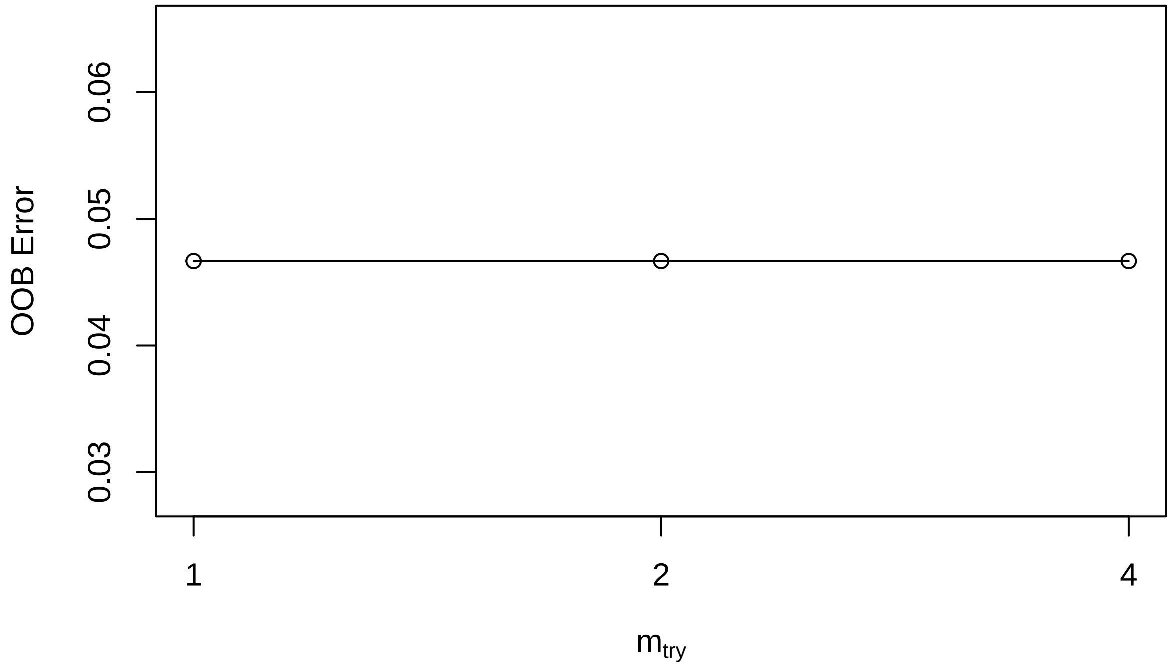
```
plot(margin(iris_rf,testData$Species))
```



7. Tune randomForest for the optimal mtry parameter

```
tune.rf <- tuneRF(iris[,-5],iris[,5], stepFactor=0.5)
```

```
## mtry = 2  OOB error = 4.67%
## Searching left ...
## mtry = 4     OOB error = 4.67%
## 0 0.05
## Searching right ...
## mtry = 1     OOB error = 4.67%
## 0 0.05
```



```
print(tune.rf)
```

```
##        mtry   OOBError
## 1.OOB    1 0.04666667
## 2.OOB    2 0.04666667
## 4.OOB    4 0.04666667
```