Random Forest On Iris Data

Girdhar Gehlot

2023-04-13

R Markdown

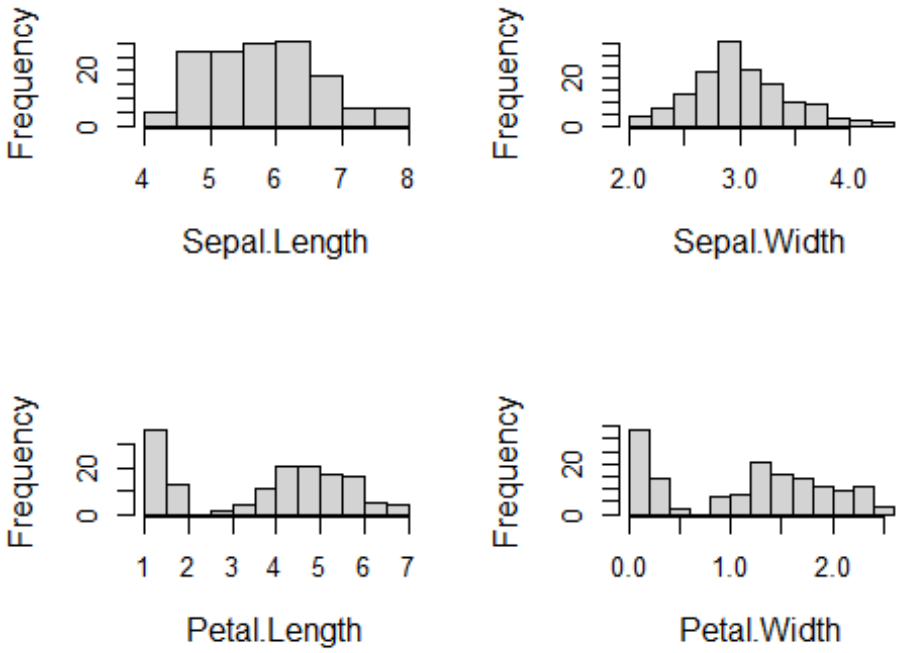
```
# loading libraries :-

library(randomForest)

## randomForest 4.7-1.1

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

library(caTools)

#view first six rows of airquality dataset
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

str(iris)

## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1
1 1 1 1 ...

# data exploring :-
summary(iris)

##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
```

```
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

```
# we can quickly see that sepals are both longer and wider than petals.

#find number of rows with missing values :-
colSums(is.na(iris))
```

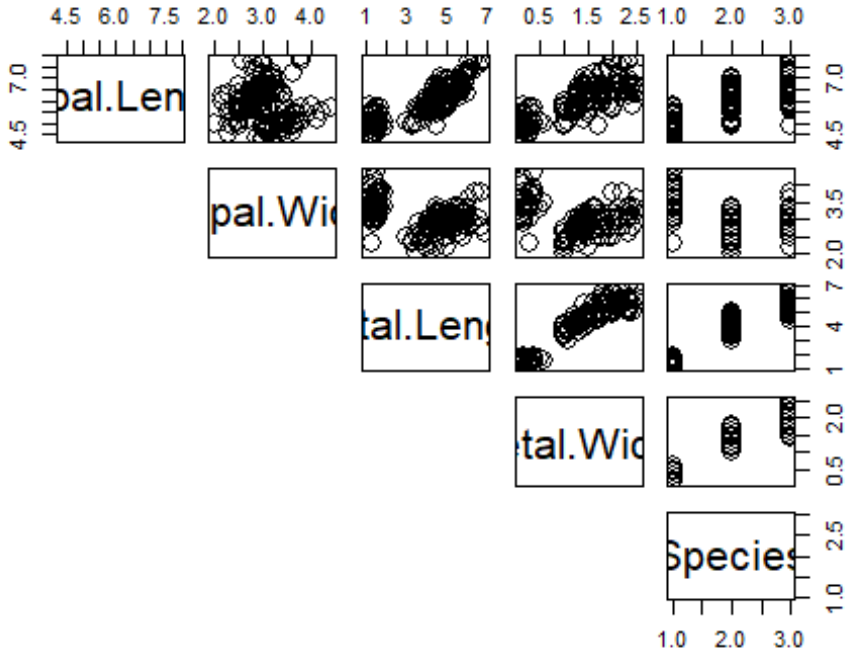
```
Sepal.Length      Sepal.Width   Petal.Length    Petal.Width      Species
           0             0            0              0              0
```

```
# To visualize and compare the distributions of the variables :-
par(mfrow=c(2,2))
for(i in 1:4){hist(iris[,i],xlab=colnames(iris[i]), cex.lab=1.2,
main="")}
```




```
# The histograms show that the petal variables are skewed and the sepal varia
bles are more symmetrical.

# put a legend in that area. The legend indicates that black represents setos
a, gray represents versicolor, and white represents virginica.
```

```
pairs(iris,lower.panel=NULL,cex=2,pch=21,cex.labels = 2,bg = c("black","grey"
,"white")[iris$species])
```




```
# The main diagonal cells, of course, have the names of the variables. Each n
onmain-diagonal cell represents the relationship between the variable in the
cell's row and the variable in the cell's column. So the cell in row 1, colum
n 2 plots the relationship between sepal.length and sepal.width. The cells in
column 5 show the relationships between each of the four measured variables a
nd species. In effect, they show the distributions of the measurements within
each species.


set.seed(700)

# data partitioning :-

split=sample.split(iris,SplitRatio=0.8)
split

## [1]  TRUE  TRUE  TRUE FALSE  TRUE

train_data<-subset(iris,split=="TRUE")
head(train_data)
```

```
##   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
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
## 7          4.6         3.4          1.4         0.3  setosa
```

```
test_data<-subset(iris,split=="FALSE")
head(test_data)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 4           4.6         3.1          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 14          4.3         3.0          1.1         0.1  setosa
## 19          5.7         3.8          1.7         0.3  setosa
## 24          5.1         3.3          1.7         0.5  setosa
## 29          5.2         3.4          1.4         0.2  setosa
```

```
#fit random forest model :-
model <- randomForest(formula = Species ~ Petal.Length + Petal.Width + Sepal.
Length + Sepal.Width,data = train_data,importance = TRUE)

#display fitted model :-
model
```

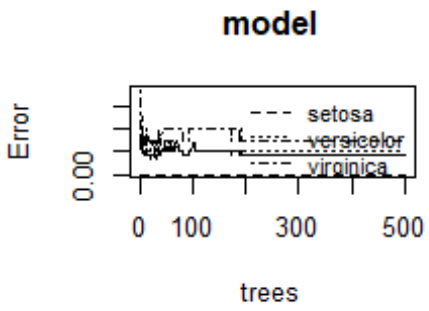
```
##
## Call:
##  randomForest(formula = Species ~ Petal.Length + Petal.Width +      Sepal.
Length + Sepal.Width, data = train_data, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 4.17%
## Confusion matrix:
##            setosa versicolor virginica class.error
## setosa         40          0         0       0.000
## versicolor      0         38         2       0.050
## virginica       0          3        37       0.075
```

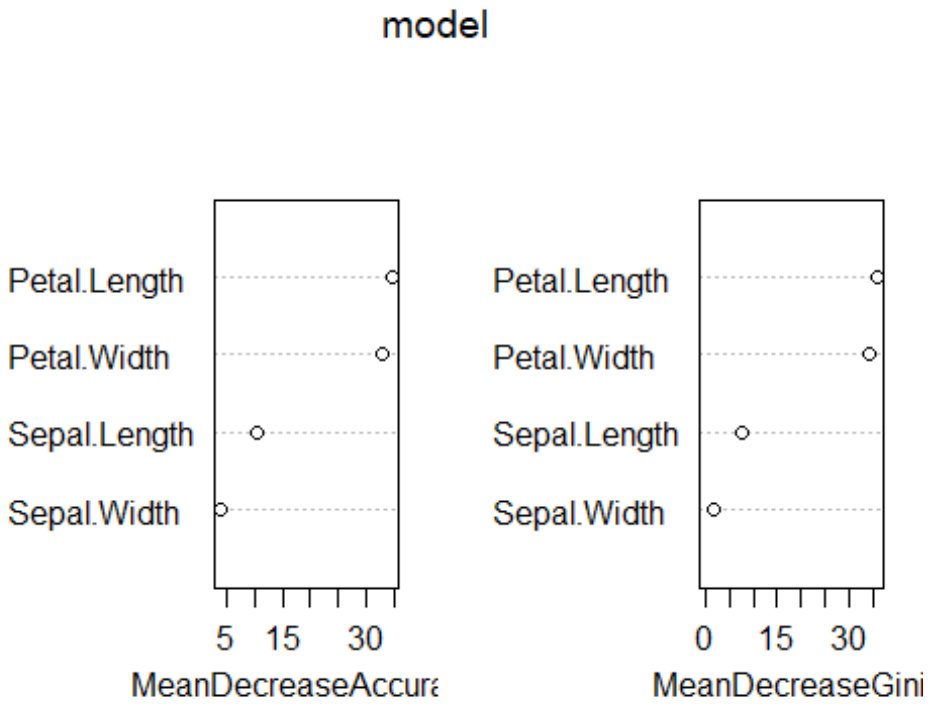
```
# accuracy of train_data iss :-

accuracy=sum(diag(model$confusion))/nrow(train_data)
accuracy
```

```
## [1] 0.9583333
```

```
# Plotting error :-

attach(iris)
```

```
plot(model, col = "black")

legend("topright", legend=c(levels(Species),"OOB"),
 lty = c("dashed","dotted","dotdash","solid"),
 cex=.8,bty = "n")

#produce variable importance plot
varImpPlot(model)
```






```
#use fitted random forest model to predict Ozone value of new observation
Species_pred=predict(model, newdata=test_data)
test_data$Species_pred=Species_pred
```

```r
confusion_matrix=table(test_data$Species,test_data$Species_pred)
confusion_matrix
```

```
##
##              setosa versicolor virginica
##   setosa         10          0         0
##   versicolor      0          9         1
##   virginica       0          0        10
```

```r
# checking accuracy of model on test data :-

accuracy=sum(diag(confusion_matrix))/nrow(test_data)
accuracy
```

```
## [1] 0.9666667
```