Classification of iris using logistic regression

Kevins_Kacha

2022-05-05

Relevant packages

Calling for the relevant libraries that will aid in our task.

Rpart helps in checking for the relationship that exist between the classes.

Rpart.plot aids in drawing the decision tree.

We wish to classify the species iris data based on the flower attributes, including the sepal.length, sepal.width, petal.length and petal.width using the decision tree or logistic regression.

```
library(rpart)
library(rpart.plot)
data("iris")
library(caret)
```

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

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

```
library(e1071)
```

Data manipulation

Based on our dataset iris the data is classified based in the species ie setosa , virginica and versicolor as below.

```
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
```

Randomisation of the data

Our aim is to mix the data up before subsettung the train data and the testing data.

We assign randomly generated numbers which are uniformly distributed and arrange them in asceding order to mix up the dataset.

According to the glimple below, the data is now mixed up.

```
set.seed(500)
g<-runif(nrow(iris))
iris_ran<-iris[order(g),]
head(iris_ran)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 74           6.1         2.8          4.7         1.2 versicolor
## 147          6.3         2.5          5.0         1.9  virginica
## 127          6.2         2.8          4.8         1.8  virginica
## 115          5.8         2.8          5.1         2.4  virginica
## 61           5.0         2.0          3.5         1.0 versicolor
## 78           6.7         3.0          5.0         1.7 versicolor
```

Fitting the model

We proceed and select first 100 rows as training data and fit a model on it using rpart using classification method.

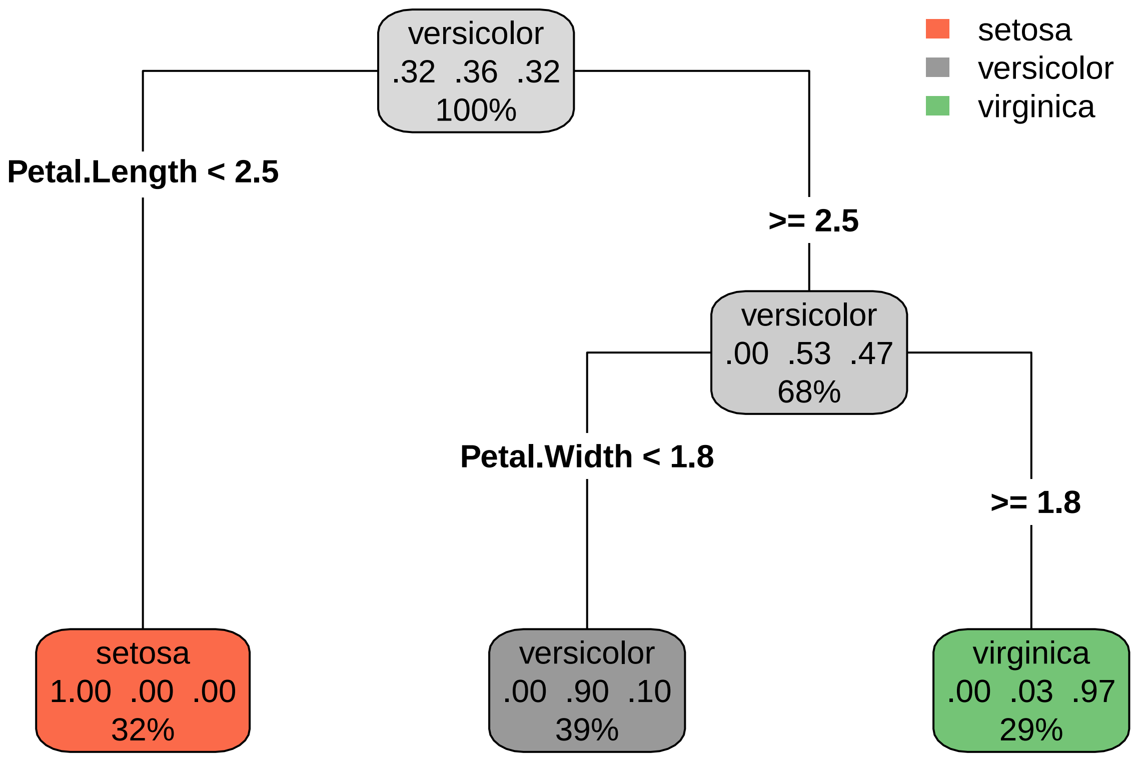
```
model1<-rpart(Species~., data =iris_ran[1:100,], method = "class")
model1
```

```
## n= 100
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 100 64 versicolor (0.32000000 0.36000000 0.32000000)
##   2) Petal.Length< 2.45 32  0 setosa (1.00000000 0.00000000 0.00000000) *
##   3) Petal.Length>=2.45 68 32 versicolor (0.00000000 0.52941176 0.47058824)
##     6) Petal.Width< 1.75 39  4 versicolor (0.00000000 0.89743590 0.10256410) *
##     7) Petal.Width>=1.75 29  1 virginica (0.00000000 0.03448276 0.96551724) *
```

The decision tree

The decision tree gives a clear picture of the classification based on the features evident in the model

```
rpart.plot(model1, type = 4, fallen.leaves = T, extra = 104 )
```



According to our plot , it is observed that the setosa species had petal.length less than 2.5, versicolor and virginica had petal.length >= 2.5. They only differ in petal width as versicolor is <1.8 and virginica is >= 1.8.

The sepal.length and sepal.width does not influence the classification of species. ## Testing the model

We tested the model on the remaining 50 rows to evaluate the goodness of fit.

```
model.predict<-predict(model1,iris_ran[101:150,], type = "class")
model.predict
```

```
##         10         60        128          2        103         15        125
##     setosa versicolor  virginica     setosa  virginica     setosa  virginica
##         13        146         62         87        150         92        116
##     setosa  virginica versicolor versicolor  virginica versicolor  virginica
##        143         22         58          5         26         37          9
##  virginica     setosa versicolor     setosa     setosa     setosa     setosa
##         43          1        118         36         63        101         45
##     setosa     setosa  virginica     setosa versicolor  virginica     setosa
##        109         75         86         12         28        106          8
##  virginica versicolor versicolor     setosa     setosa  virginica     setosa
##         39         65        144        121        114         82         59
##     setosa versicolor  virginica  virginica  virginica versicolor versicolor
##         83        122          3         96        130        129        105
## versicolor  virginica     setosa versicolor versicolor  virginica  virginica
##         52
## versicolor
## Levels: setosa versicolor virginica
```

Prediction accuracy

The function confusionmatrix in caTools helps to check the level of prediction accuracy.

Based on our model, the prediction accuracy is 98%. It predicted all setosa species correctly, out of 15 versicolor it predicted 14 of them correctly and lastly it predicted all the virginica species correctly.

```
confusionMatrix(iris_ran[101:150,5], reference = model.predict)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         18          0         0
##   versicolor      0         14         0
##   virginica       0          1        17
##
## Overall Statistics
##
##                Accuracy : 0.98
##                  95% CI : (0.8935, 0.9995)
##     No Information Rate : 0.36
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9699
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                   1.00            0.9333           1.0000
## Specificity                   1.00            1.0000           0.9697
## Pos Pred Value                1.00            1.0000           0.9444
## Neg Pred Value                1.00            0.9722           1.0000
## Prevalence                    0.36            0.3000           0.3400
## Detection Rate                0.36            0.2800           0.3400
## Detection Prevalence          0.36            0.2800           0.3600
## Balanced Accuracy             1.00            0.9667           0.9848
```