Analise Multivariada II

Luciane Alcoforado

setembro de 2018

Universidade Federal Fluminense
Instituto de Matemática e Estatística
Departamento de Estatística

1 Aula 9 - Árvore de regressão

1.1 Arvores de regressão

Dados utilizados: iris; pacote **tree**, função *tree*

Vamos dividir o conjunto de dados em dois: treinamento e teste

```
set.seed(100)
alpha     <- 0.7 # porcentagem do conjunto de treinamento
inTrain   <- sample(1:nrow(iris), alpha * nrow(iris))
train.set <- iris[inTrain,]
test.set  <- iris[-inTrain,]
```

```
library(rpart)
library(rpart.plot)
arv1=rpart(Species~., data=train.set)
arv1
```

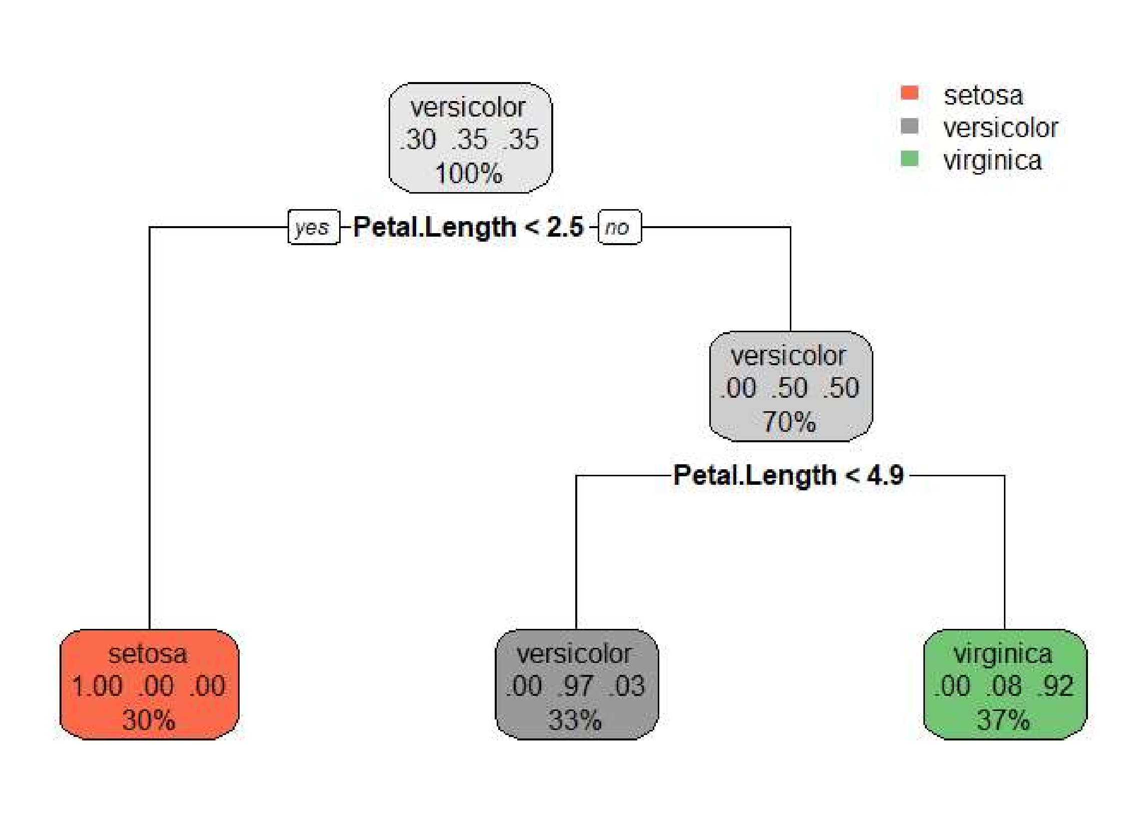
```
## n= 105
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 105 68 versicolor (0.29524 0.35238 0.35238)
##   2) Petal.Length< 2.45 31  0 setosa (1.00000 0.00000 0.00000) *
##   3) Petal.Length>=2.45 74 37 versicolor (0.00000 0.50000 0.50000)
##     6) Petal.Length< 4.85 35  1 versicolor (0.00000 0.97143 0.02857) *
##     7) Petal.Length>=4.85 39  3 virginica (0.00000 0.07692 0.92308) *
```

```r
summary(arv1)
```

```
## Call:
## rpart(formula = Species ~ ., data = train.set)
##   n= 105
##
##       CP nsplit rel error xerror    xstd
## 1 0.4706      0   1.00000 1.2353 0.06028
## 2 0.0100      2   0.05882 0.1471 0.04423
##
## Variable importance
## Petal.Length  Petal.Width Sepal.Length  Sepal.Width
##           35           32           21           13
##
## Node number 1: 105 observations,    complexity param=0.4706
##   predicted class=versicolor  expected loss=0.6476  P(node) =1
##     class counts:    31    37    37
##    probabilities: 0.295 0.352 0.352
##   left son=2 (31 obs) right son=3 (74 obs)
##   Primary splits:
##       Petal.Length < 2.45 to the left,  improve=32.77, (0 missing)
##       Petal.Width  < 0.75 to the left,  improve=32.77, (0 missing)
##       Sepal.Length < 5.45 to the left,  improve=20.94, (0 missing)
##       Sepal.Width  < 3.35 to the right, improve=12.78, (0 missing)
##   Surrogate splits:
##       Petal.Width  < 0.75 to the left,  agree=1.000, adj=1.000, (0 split)
##       Sepal.Length < 5.45 to the left,  agree=0.914, adj=0.710, (0 split)
##       Sepal.Width  < 3.35 to the right, agree=0.848, adj=0.484, (0 split)
##
## Node number 2: 31 observations
##   predicted class=setosa      expected loss=0  P(node) =0.2952
##     class counts:    31     0     0
##    probabilities: 1.000 0.000 0.000
##
## Node number 3: 74 observations,    complexity param=0.4706
##   predicted class=versicolor  expected loss=0.5  P(node) =0.7048
##     class counts:     0    37    37
##    probabilities: 0.000 0.500 0.500
##   left son=6 (35 obs) right son=7 (39 obs)
##   Primary splits:
##       Petal.Length < 4.85 to the left,  improve=29.520, (0 missing)
##       Petal.Width  < 1.75 to the left,  improve=29.520, (0 missing)
##       Sepal.Length < 6.15 to the left,  improve= 8.022, (0 missing)
##       Sepal.Width  < 2.45 to the left,  improve= 2.523, (0 missing)
##   Surrogate splits:
##       Petal.Width  < 1.65 to the left,  agree=0.905, adj=0.800, (0 split)
##       Sepal.Length < 6.25 to the left,  agree=0.743, adj=0.457, (0 split)
##       Sepal.Width  < 2.65 to the left,  agree=0.635, adj=0.229, (0 split)
##
```

```
## Node number 6: 35 observations
##    predicted class=versicolor  expected loss=0.02857  P(node) =0.3333
##      class counts:     0    34     1
##     probabilities: 0.000 0.971 0.029
##
## Node number 7: 39 observations
##    predicted class=virginica   expected loss=0.07692  P(node) =0.3714
##      class counts:     0     3    36
##     probabilities: 0.000 0.077 0.923
```

```
rpart.plot(arv1)
```



Outro pacote: tree, diferencial: fornece uma visualização de acordo com a partição

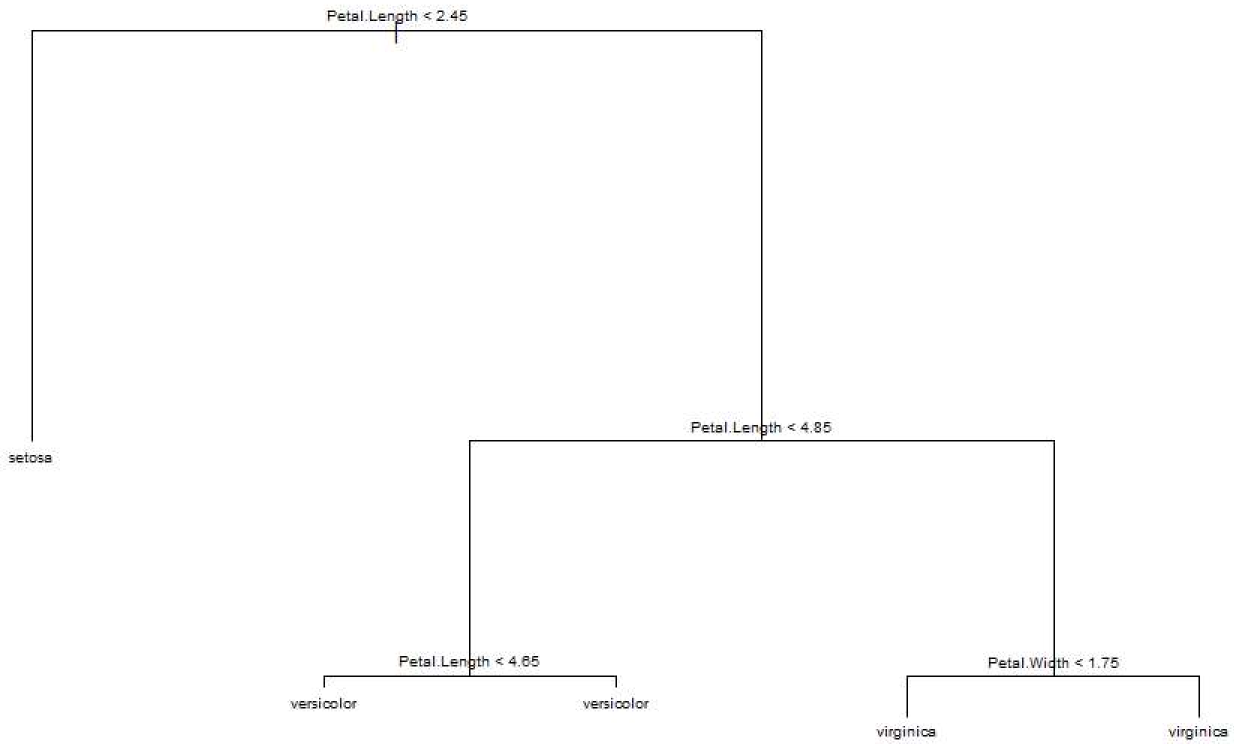
```
library(tree)

arv2 <- tree(Species ~ ., data=train.set)
arv2
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
```

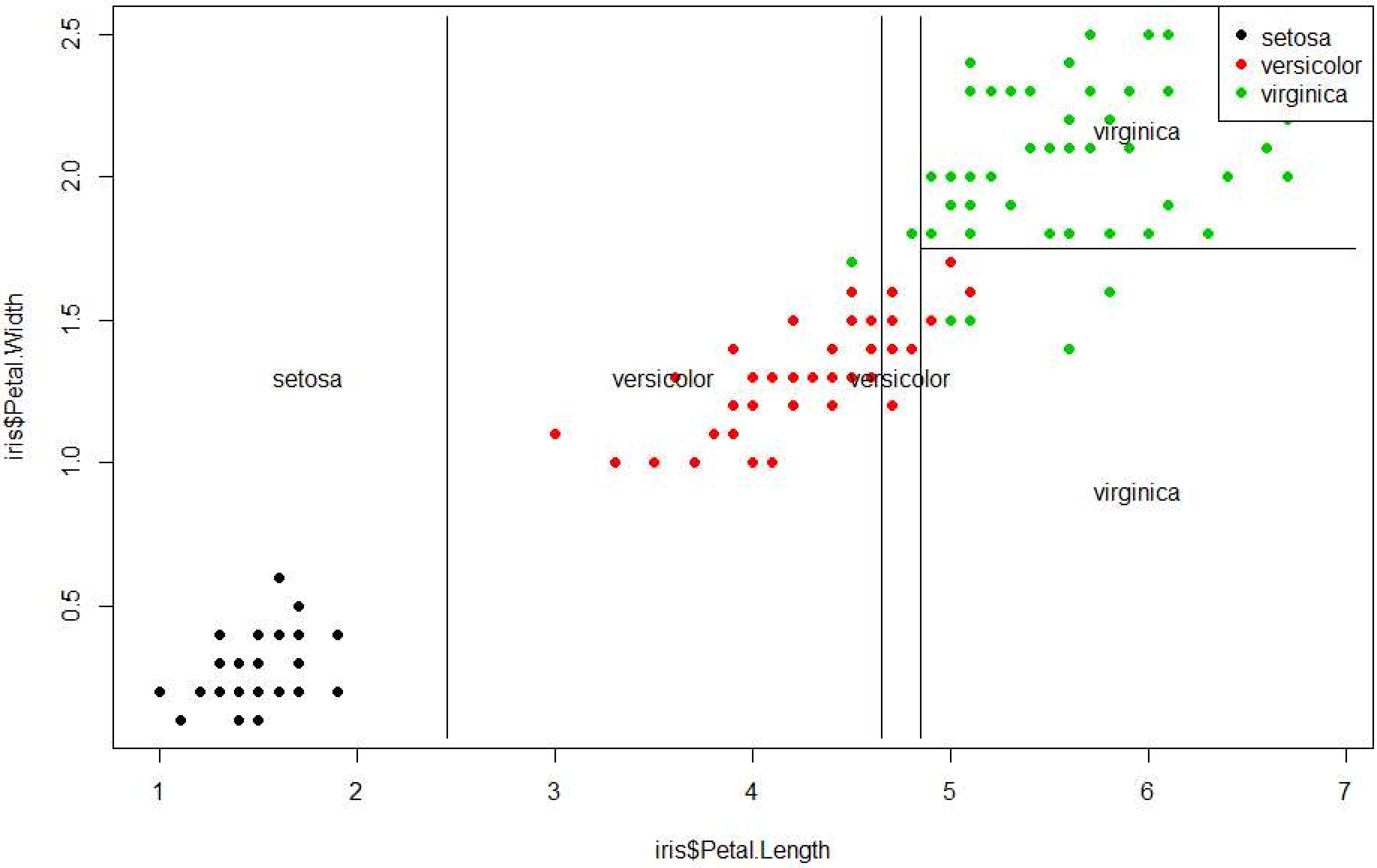
```
## 1) root 105 200 versicolor ( 0.30 0.35 0.35 )
##   2) Petal.Length < 2.45 31   0 setosa ( 1.00 0.00 0.00 ) *
##   3) Petal.Length > 2.45 74 100 versicolor ( 0.00 0.50 0.50 )
##     6) Petal.Length < 4.85 35   9 versicolor ( 0.00 0.97 0.03 )
##      12) Petal.Length < 4.65 29   0 versicolor ( 0.00 1.00 0.00 ) *
##      13) Petal.Length > 4.65 6   5 versicolor ( 0.00 0.83 0.17 ) *
##     7) Petal.Length > 4.85 39  20 virginica ( 0.00 0.08 0.92 )
##      14) Petal.Width < 1.75 6   8 virginica ( 0.00 0.50 0.50 ) *
##      15) Petal.Width > 1.75 33   0 virginica ( 0.00 0.00 1.00 ) *
```

```
plot(arv2)
text(arv2, cex=0.6)
```



Outra forma de visualizar a árvore:

```
plot(iris$Petal.Length,iris$Petal.Width,  pch=19, col=as.numeric(iris$Species))
partition.tree(arv2, label="Species", add=TRUE)
legend("topright",legend=unique(iris$Species), col=unique(as.numeric(iris$Species)), pch=19)
```



2 Exercício

Compare os modelos obtidos utilizando a matriz de confusão, monte uma tabela com os modelos e as medidas de acurácia, sensibilidade e especificidade.

3 Trabalho para avaliação

Organize um banco de dados e aplique a técnica de árvore de regressão. Descreva o contexto, qual o objetivo da classificação e os resultados. Apresentar na aula do dia 11/10.