An introduction to statistical learning - Ch8 - Ex8

Thalles Quinaglia Liduares

24/03/2023

In the lab, a classification tree was applied to the Carseats data (islr r package) set after converting Sales into a qualitative response variable. Now we will seek to predict Sales using regression trees and related approaches, treating the response as a quantitative variable.

a. Split the data set into a training set and a test set.

Upload packages

```
library(ISLR)
library(caret)
```

```
## Carregando pacotes exigidos: ggplot2
```

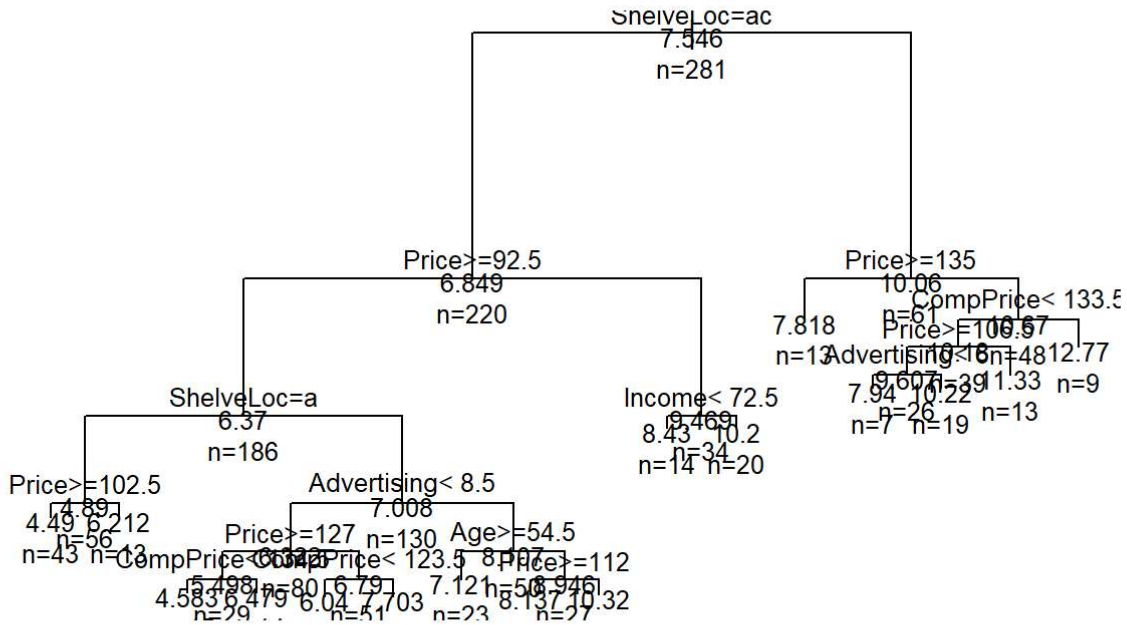
```
## Carregando pacotes exigidos: lattice
```

```
# set the seed

# split the data into a training set (70%) and a test set (30%)

trainIndex <- createDataPartition(Carseats$Sales, p = 0.7, list = FALSE)
training <- Carseats[trainIndex, ]
testing <- Carseats[-trainIndex, ]
```

b. Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain?

```
library(rpart)

# fit a regression tree to the training set

fit <- rpart(Sales ~ ., data = training, method = "anova")
```

```
# plot the tree

plot(fit)

text(fit, use.n = TRUE, all = TRUE, cex = 0.8)
```




```
# predict Sales on the test set

predictions <- predict(fit, testing)
```

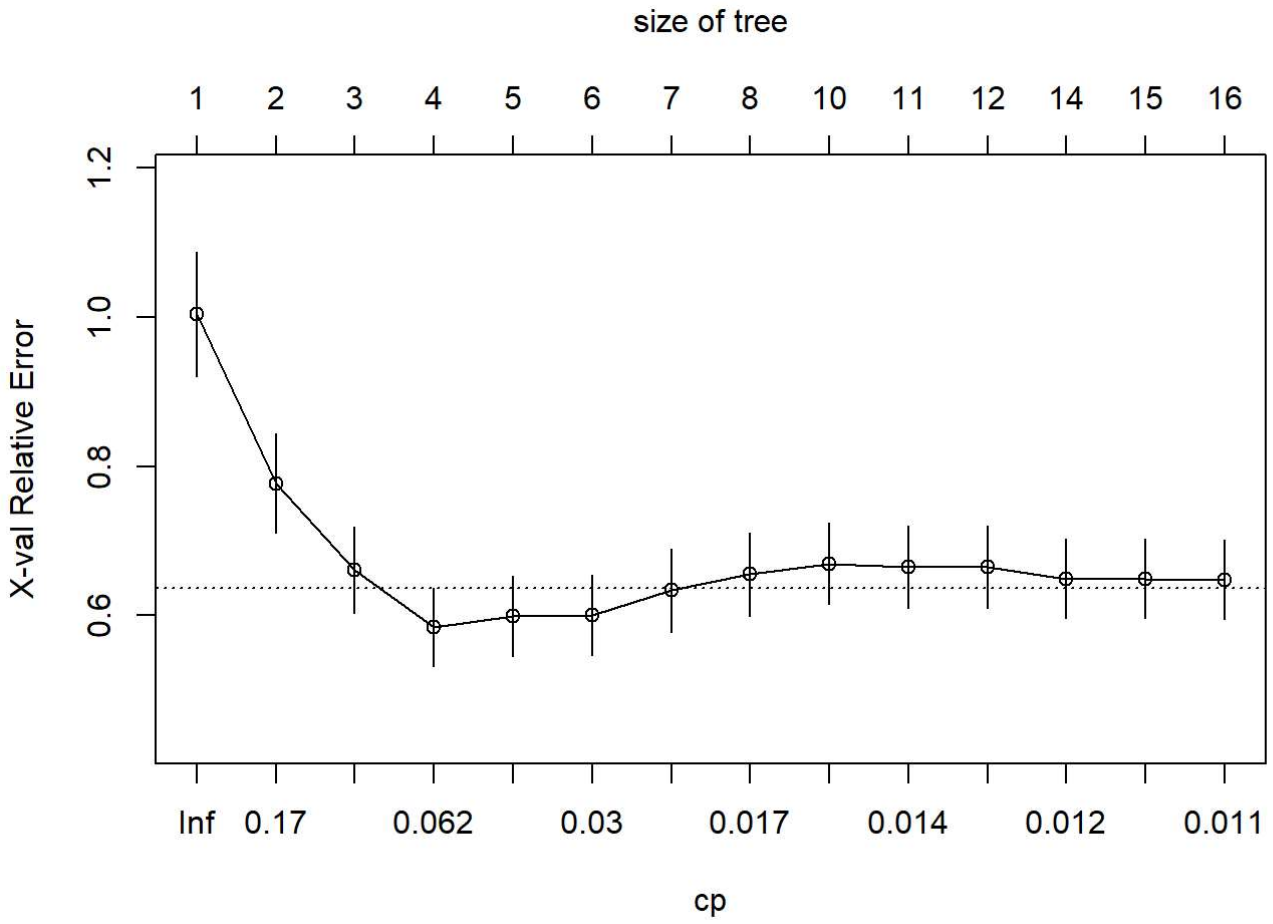
```
# calculate the test MSE

mse <- mean((testing$Sales - predictions)^2)

mse
```

```
## [1] 4.623136
```

c. Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?

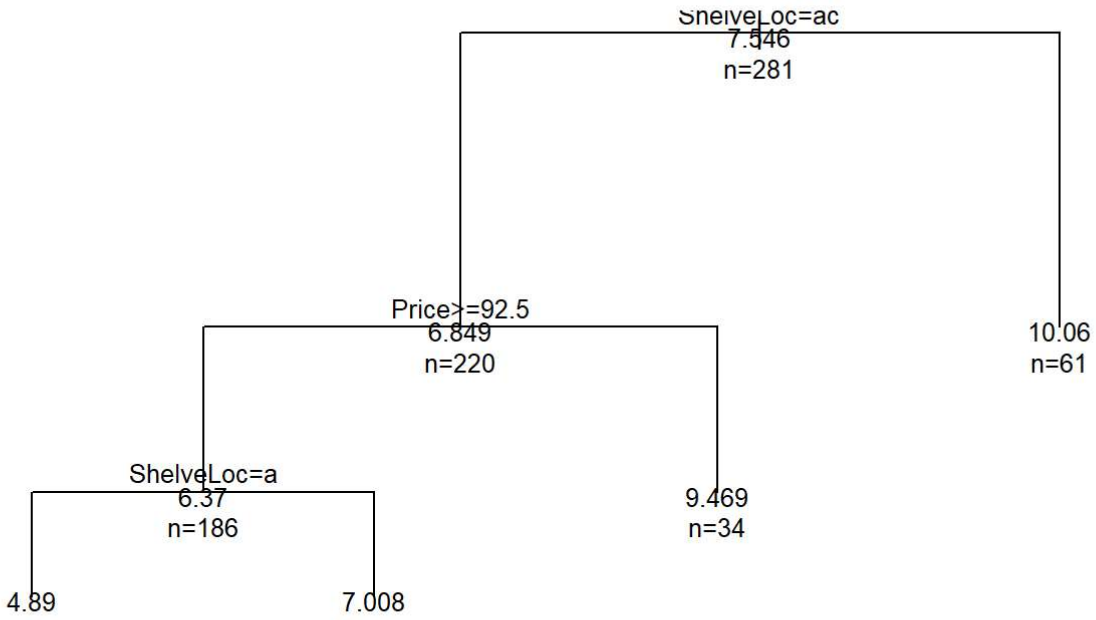
```
# fit a regression tree with cross-validation

fit.cv <- rpart(Sales ~ ., data = training, method = "anova",
                control = rpart.control(cp = 0.01),
                parms = list(split = "information"),
                cp = seq(0, 0.1, by = 0.001))
```

```
# plot the cross-validation error

plotcp(fit.cv)
```




```
# prune the tree

prune.fit <- prune(fit.cv, cp = fit.cv$cptable[which.min(fit.cv$cptable[, "xerror"]), "CP"])
```

```
# plot the pruned tree
plot(prune.fit)
text(prune.fit, use.n = TRUE, all = TRUE, cex = 0.8)
```




```
# predict Sales on the test set
predictions <- predict(prune.fit, testing)
```

```
# calculate the test MSE

mse <- mean((testing$Sales - predictions)^2)
mse
```

```
## [1] 6.058706
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

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

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

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

```
# fit a random forest model to the training set
fit <- randomForest(Sales ~ ., data = training, ntree = 500, mtry = 3)
```

```
# predict Sales on the test set
predictions <- predict(fit, testing)
```

```
# calculate the test MSE
mse <- mean((testing$Sales - predictions)^2)
mse
```

```
## [1] 3.361242
```