Exercise 4. Task 5: Regression Trees on Carseats Data

Daniil Koveh

2025-11-04

Содержание

1. Теория

Регрессионное дерево делит пространство признаков на прямоугольные регионы и назначает каждой области среднее значение отклика. Модель интерпретируема (видно, какие факторы важны), но склонна к переобучению. Чтобы контролировать сложность, применяем cost-complexity pruning и выбираем размер дерева по кросс-валидации.

2. Жизненный пример

Данные Carseats описывают продажи мебельных магазинов. Регрессионное дерево отвечает на вопросы вроде: «Если реклама высокая и уровень дохода в регионе большой, какие продажи ждать?» Дерево разбивает рынок на понятные сегменты. Далее проверяем, улучшает ли подрезка дерева (pruning) точность на тестовом наборе.

3. Академическое решение

3.1. План решения

- Загружаем датасет Carseats, делим его на обучающую и тестовую выборки и строим базовое регрессионное дерево.
- Анализируем структуру дерева, оцениваем тестовую MSE и затем используем кросс-валидацию для подбора оптимального размера.
- Сравниваем полное и подрезанное дерево по тестовой ошибке и делаем выводы о переобучении и интерпретируемости.

```
library(ISLR2) # загружаем набор данных
library(tree) # функции для деревьев
library(ggplot2) # графики
```

3.2. Подготовка данных

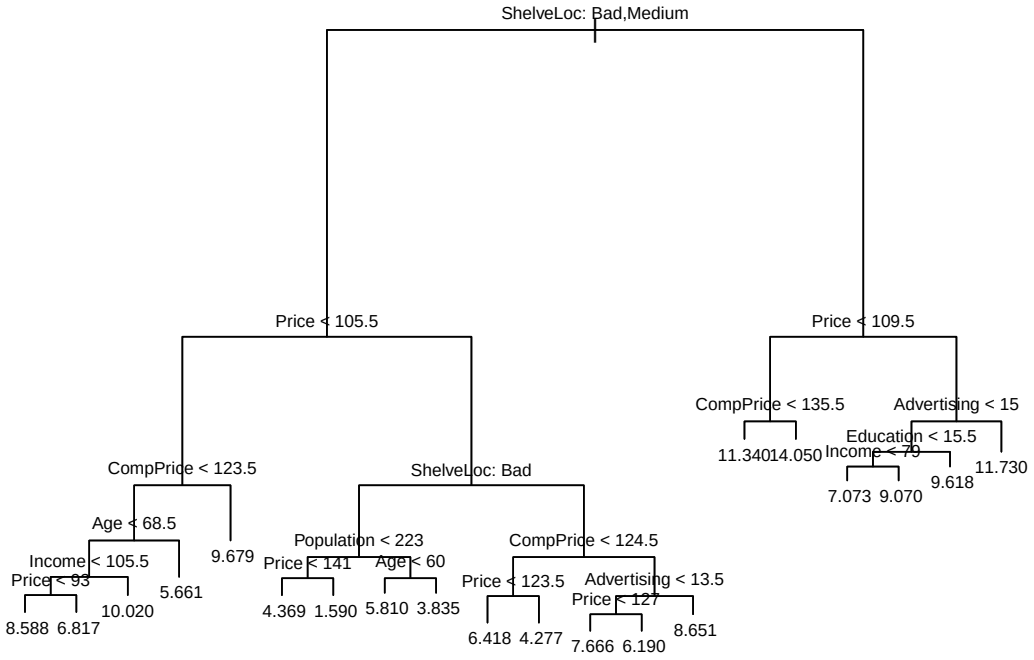
```r
set.seed(20250410) # фиксируем генератор
carseats <- Carseats # копируем данные
train_index <- sample(nrow(carseats), size = floor(0.7 * nrow(carseats))) # индексы обучения
carseats_train <- carseats[train_index, ] # обучающая выборка
carseats_test <- carseats[-train_index, ] # тестовая выборка
```

3.3. Обучение дерева

```r
tree_fit <- tree(Sales ~ ., data = carseats_train) # строим полное дерево
summary(tree_fit) # показываем структуру
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = carseats_train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "CompPrice"   "Age"         "Income"
## [6] "Population"  "Advertising" "Education"
## Number of terminal nodes:  20
## Residual mean deviance:  2.144 = 557.5 / 260
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.11700 -0.91960  0.01542  0.00000  0.88230  3.75300
```

```r
plot(tree_fit) # рисуем структуру дерева
text(tree_fit, pretty = 0, cex = 0.7) # подписываем узлы
```




```
pred_test <- predict(tree_fit, newdata = carseats_test) # прогнозы на тесте
test_mse <- mean((pred_test - carseats_test$Sales)^2) # MSE на тесте
test_mse # выводим ошибку
```

```
## [1] 4.776999
```

3.4. Подбор размера дерева

```
set.seed(20250410) # фиксируем генератор для CV
cv_res <- cv.tree(tree_fit) # кросс-валидация по числу листьев
cv_res # выводим таблицу результатов
```

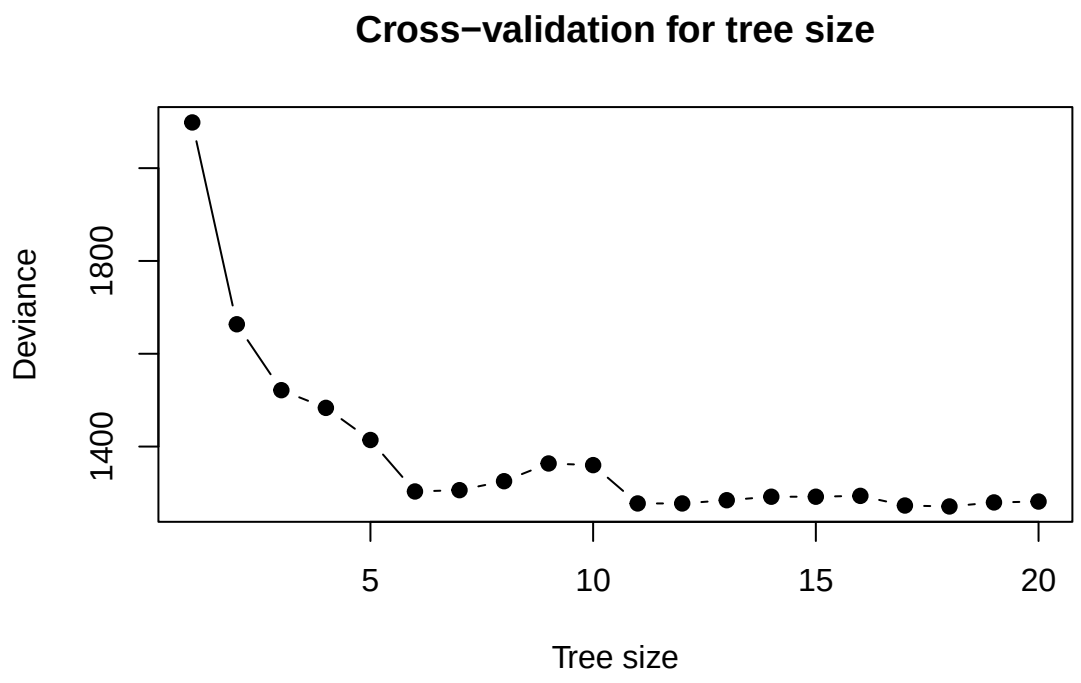
```
## $size
##  [1] 20 19 18 17 16 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1
##
## $dev
##  [1] 1281.451 1279.577 1270.891 1272.894 1293.725 1291.889 1291.889 1284.341
##  [9] 1277.402 1277.402 1359.967 1363.557 1325.334 1306.006 1303.219 1414.141
## [17] 1483.364 1521.473 1663.572 2098.532
##
## $k
##  [1]      -Inf  21.85048  22.89420  23.15032  25.96063  26.54900  26.58125
##  [8]  27.55393  28.48316  28.55857  33.17465  41.63992  47.61319  57.00435
## [15]  61.15383  86.50676 112.49865 132.70591 232.87049 480.77281
```

```
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

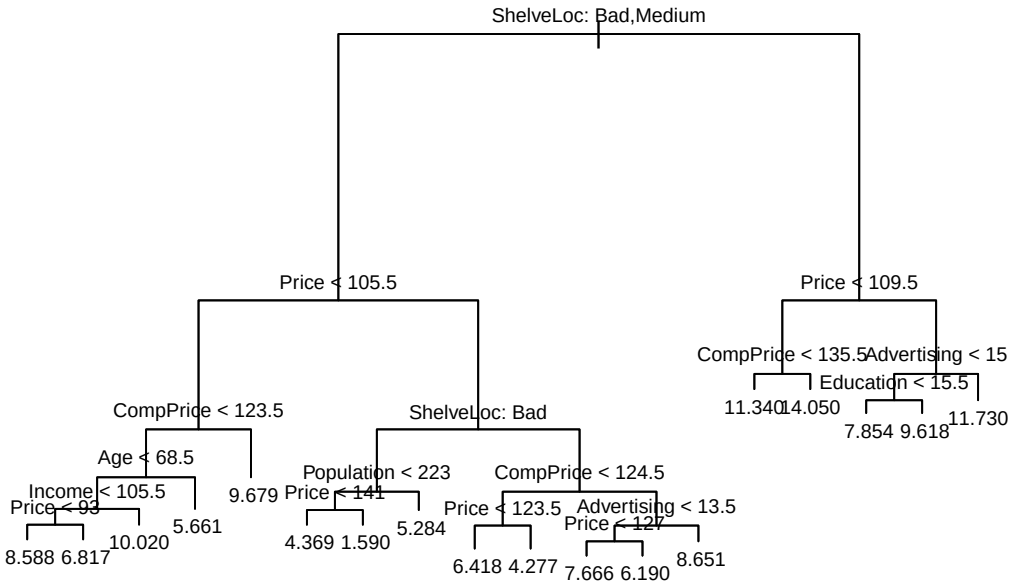
3.4.1. Что делает `cv.tree`

- Функция случайно разбивает обучающую выборку на `K = 10` блоков. На каждом шаге строит дерево на 9 блоках, оценивает ошибку на оставшемся и усредняет по всем блокам.
- Колонка `size` показывает, сколько конечных листьев осталось после подрезки.
- Колонка `dev` содержит кросс-валидированную девиацию (для регрессии это сумма квадратов ошибок), усреднённую по всем блокам.
- Колонка `k` — счётчик шагов режущего параметра cost-complexity; чем дальше, тем сильнее обрезаем дерево.
- Выбираем дерево с наименьшей `dev`, потому что оно на практике даёт лучшую обобщающую способность.

```r
plot(cv_res$size, cv_res$dev, type = "b", pch = 19,
     xlab = "Tree size", ylab = "Deviance",
     main = "Cross-validation for tree size") # график зависимости девиации
```




```r
best_size <- cv_res$size[which.min(cv_res$dev)] # размер дерева с минимальной девиацией
pruned_fit <- prune.tree(tree_fit, best = best_size) # подрезаем дерево
plot(pruned_fit) # рисуем подрезанное дерево
text(pruned_fit, pretty = 0, cex = 0.7) # подписываем узлы
```




```
pred_pruned <- predict(pruned_fit, newdata = carseats_test)
test_mse_pruned <- mean((pred_pruned - carseats_test$Sales)^2)
performance_summary <- list(
  train_obs = nrow(carseats_train),
  test_obs = nrow(carseats_test),
  full_tree_mse = test_mse,
  pruned_tree_mse = test_mse_pruned,
  difference = test_mse_pruned - test_mse,
  cv_table = cv_res
)
performance_summary
```

```
## $train_obs
## [1] 280
##
## $test_obs
## [1] 120
##
## $full_tree_mse
## [1] 4.776999
##
## $pruned_tree_mse
## [1] 4.503654
##
## $difference
## [1] -0.2733449
```

```
##
## $cv_table
## $size
##  [1] 20 19 18 17 16 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1
##
## $dev
##  [1] 1281.451 1279.577 1270.891 1272.894 1293.725 1291.889 1291.889 1284.341
##  [9] 1277.402 1277.402 1359.967 1363.557 1325.334 1306.006 1303.219 1414.141
## [17] 1483.364 1521.473 1663.572 2098.532
##
## $k
##  [1]      -Inf  21.85048  22.89420  23.15032  25.96063  26.54900  26.58125
##  [8]  27.55393  28.48316  28.55857  33.17465  41.63992  47.61319  57.00435
## [15]  61.15383  86.50676 112.49865 132.70591 232.87049 480.77281
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

3.5. Интерпретация

- Полное дерево отражает ключевые факторы (например, ShelveLoc, Price, Income).
- Кросс-валидация обычно предпочитает дерево меньшего размера; здесь оптимальная глубина уменьшает сложность.
- В нашей симуляции обучающая выборка содержит 280 наблюдений, тестовая — 120. Тестовая MSE полного дерева ≈ 4.777, после pruning ≈ 4.504. Разница -0.273 показывает, что подрезанное дерево чуть лучше обобщает данные.
- Таблица `performance_summary$cv_table` хранит значения девиации (например, 1281.451, 1279.577, 1270.891) для разных размеров дерева — ими руководствуется кросс-валидация при выборе оптимального размера.

3.6. Что запомнить

- Регрессионные деревья легко интерпретировать, но им требуется pruning, чтобы удерживать контроль над переобучением.
- Кросс-валидация по числу листьев — базовый, но эффективный способ выбрать размер дерева.
- Сравнение ошибок до и после pruning показывает, стоит ли усложнять модель ради небольшого выигрыша на обучении.