Exercise 4. Task 8: Spam Classification with GLM and Trees

Daniil Ковеh

2025-10-30

Содержание

1. Теория

Сравниваем логистическую регрессию (генерализованную линейную модель) и классификационное дерево на одном тестовом наборе. Обе модели оценивают вероятность письма быть спамом, но делают это по-разному: логистическая регрессия — линейно, дерево — сегментируя пространство признаков. Проверяем точность, строим доверительные интервалы и используем тест Мак-Нимара для сравнения ошибочных классификаций.

2. Жизненный пример

Представьте антиспам-фильтр. Логистическая модель назначает вес каждому слову (сколько раз оно встречается в письме). Дерево разделяет письма по условиям вида «если слово free встречается чаще X и слово click чаще Y, то спам». Оба подхода обучаем на 3451 письмах и тестируем на 1000. В конце обсуждаем, есть ли основание считать, что одна модель лучше.

3. Академическое решение

```
library(ElemStatLearn) # загружаем данные
library(rpart) # деревья
library(caret) # инструменты для матриц
```

```
data("spam", package = "ElemStatLearn") # загружаем набор
set.seed(1234) # фиксируем генератор как в условии
test_index <- sample(nrow(spam), 1000) # выбираем тестовые индексы
spam_train <- spam[-test_index, ] # обучающая выборка
spam_test <- spam[test_index, ] # тестовая выборка
```

3.1. Логистическая регрессия

```
glm_fit <- glm(spam ~ ., data = spam_train, family = binomial(link = "logit")) # фитим GLM
glm_prob <- predict(glm_fit, newdata = spam_test, type = "response") # вероятности на тесте
glm_pred <- factor(ifelse(glm_prob > 0.5, "spam", "email"), levels = c("email", "spam")) # классификация
```

```
glm_conf <- confusionMatrix(glm_pred, spam_test$spam, positive = "spam") # матрица и метрики
glm_conf # выводим отчёт
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction email spam
##      email   584   51
##      spam     26  339
##
##                Accuracy : 0.923
##                  95% CI : (0.9047, 0.9388)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8363
##
##  Mcnemar's Test P-Value : 0.006237
##
##             Sensitivity : 0.8692
##             Specificity : 0.9574
##          Pos Pred Value : 0.9288
##          Neg Pred Value : 0.9197
##              Prevalence : 0.3900
##          Detection Rate : 0.3390
##    Detection Prevalence : 0.3650
##       Balanced Accuracy : 0.9133
##
##        'Positive' Class : spam
##
```

```
glm_ci <- prop.test(sum(glm_pred == spam_test$spam), length(glm_pred))$conf.int # доверительный интервал
glm_ci # выводим 95% ДИ
```

```
## [1] 0.9042638 0.9384014
## attr(,"conf.level")
## [1] 0.95
```

3.2. Классификационное дерево

```
tree_fit <- rpart(spam ~ ., data = spam_train, method = "class",
                  control = rpart.control(cp = 0.001)) # строим полное дерево
best_cp <- tree_fit$cptable[which.min(tree_fit$cptable[, "xerror"]), "CP"] # cp с минимальным xerror
tree_pruned <- prune(tree_fit, cp = best_cp) # подрезаем дерево
tree_pred <- predict(tree_pruned, newdata = spam_test, type = "class") # предсказания на тесте
tree_conf <- confusionMatrix(tree_pred, spam_test$spam, positive = "spam") # матрица
tree_conf # выводим отчёт
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction email spam
##      email   577   53
##      spam     33  337
##
##                Accuracy : 0.914
##                  95% CI : (0.8949, 0.9306)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8176
##
##  Mcnemar's Test P-Value : 0.04048
##
##             Sensitivity : 0.8641
##             Specificity : 0.9459
##          Pos Pred Value : 0.9108
##          Neg Pred Value : 0.9159
##              Prevalence : 0.3900
##          Detection Rate : 0.3370
##    Detection Prevalence : 0.3700
##       Balanced Accuracy : 0.9050
##
##        'Positive' Class : spam
##
```

```
tree_ci <- prop.test(sum(tree_pred == spam_test$spam), length(tree_pred))$conf.int # 95% ДИ точности
tree_ci # выводим интервал
```

```
## [1] 0.8944567 0.9302842
## attr(,"conf.level")
## [1] 0.95
```

3.3. Сравнение моделей

```
agreement <- table(GLM = glm_pred == spam_test$spam,
                   Tree = tree_pred == spam_test$spam) # таблица правильных классификаций
agreement # выводим таблицу
```

```
##        Tree
## GLM     FALSE TRUE
##   FALSE    46   31
##   TRUE     40  883
```

```
mcnemar.test(agreement) # тест Мак-Нимара
```

```
##
##  McNemar's Chi-squared test with continuity correction
##
## data:  agreement
## McNemar's chi-squared = 0.90141, df = 1, p-value = 0.3424
```

```
pred_confusion <- table(GLM = glm_pred, Tree = tree_pred) # таблица предсказанных классов
pred_confusion # выводим матрицу для понимания различий
```

```
##        Tree
## GLM     email spam
##   email   597   38
##   spam     33  332
```

3.4. Интерпретация

- Логистическая регрессия обычно показывает более высокую точность и узкий доверительный интервал.
- Дерево чуть менее точное, но интерпретация правил более наглядна.
- Тест Мак-Нимара сообщает, есть ли статистически значимая разница в ошибках; если p-значение небольшое, модели действительно отличаются.
- Конфузионная матрица предсказаний показывает, где модели расходятся (какие письма один классификатор пометил как спам, а другой нет).