Assaf Bitton

We will be predicting whether the housing price is expensive or not using the `sahp` dataset in the **r02pro** package.

You can run the following code to prepare the analysis.

```
library(r02pro)     #INSTALL IF NECESSARY
```

```
## Warning: package 'r02pro' was built under R version 4.4.3
```

```
library(tidyverse)  #INSTALL IF NECESSARY
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
## Warning: package 'tidyr' was built under R version 4.4.2
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.4.3
```

```
my_sahp <- sahp %>%
  na.omit() %>%
  mutate(expensive = sale_price > median(sale_price)) %>%
  dplyr::select(gar_car, liv_area, oa_qual, expensive)
my_sahp_train <- my_sahp[1:100, ]
my_sahp_test <- my_sahp[-(1:100), ]
```

Please answer the following questions.

*Q1*

a. Using the training data `my_sahp_train` to fit a logistic regression model of `expensive` on each variable (`gar_car`, `liv_area`, `oa_qual`) separately. For each logistic regression, compute the training and test error. Which variable leads to the smallest training error? Which variable leads to the smallest test error?

**Solution:**

```
str(my_sahp_train$expensive)
```

```
##  logi [1:100] FALSE FALSE TRUE FALSE TRUE FALSE ...
```

```
my_sahp_train$expensive <-
  as.factor(my_sahp_train$expensive)
my_sahp_test$expensive <-
  as.factor(my_sahp_test$expensive)
str(my_sahp_train$expensive)
```

```
##  Factor w/ 2 levels "FALSE","TRUE": 1 1 2 1 2 1 1 1 1 1 ...
```

```
str(my_sahp_test$expensive)
```

```
##  Factor w/ 2 levels "FALSE","TRUE": 1 2 1 1 1 1 1 2 2 2 ...
```

```
glm_expensive_gar <-
  glm(expensive ~ gar_car,
      data = my_sahp_train,
    family = "binomial"
    )

glm_expensive_liv <-
  glm(expensive ~ liv_area,
      data = my_sahp_train,
    family = "binomial"
    )
glm_expensive_oa <-
  glm(expensive ~ oa_qual,
      data = my_sahp_train,
    family = "binomial"
    )
summary(glm_expensive_gar)
```

```
##
## Call:
## glm(formula = expensive ~ gar_car, family = "binomial", data = my_sahp_tra
in)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.2279     1.4330  -3.648 0.000264 ***
## gar_car       2.7339     0.7298   3.746 0.000180 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 137.63  on 99  degrees of freedom
## Residual deviance: 102.24  on 98  degrees of freedom
## AIC: 106.24
##
## Number of Fisher Scoring iterations: 6
```

```
summary(glm_expensive_liv)
```

```
##
## Call:
## glm(formula = expensive ~ liv_area, family = "binomial", data = my_sahp_tr
ain)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.8172511  0.9550756  -3.997 6.42e-05 ***
## liv_area     0.0024880  0.0006447   3.859 0.000114 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 137.63  on 99  degrees of freedom
## Residual deviance: 115.98  on 98  degrees of freedom
## AIC: 119.98
##
## Number of Fisher Scoring iterations: 4
```

```
summary(glm_expensive_oa)
```

```
##
## Call:
## glm(formula = expensive ~ oa_qual, family = "binomial", data = my_sahp_tra
in)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.7964     1.7314  -5.080 3.77e-07 ***
## oa_qual       1.4078     0.2835   4.967 6.82e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 137.63  on 99  degrees of freedom
## Residual deviance:  94.05  on 98  degrees of freedom
## AIC: 98.05
##
## Number of Fisher Scoring iterations: 5
```

```
#Training Error Garage
pred_gar_train <- predict(glm_expensive_gar,
  newdata = my_sahp_train,
  type = "response")

class_labels <- levels(my_sahp_train$expensive)
c0 <- class_labels[1]
c1 <- class_labels[2]
pred_gar_train_label <- ifelse(pred_gar_train
  > 0.5, c1, c0)

gar_tr_error <-
  mean(pred_gar_train_label != my_sahp_train$expensive)

gar_tr_error
```

```
## [1] 0.29
```

```r
#Test Error Garage
pred_gar_te <- predict(glm_expensive_gar,
  newdata = my_sahp_test,
  type = "response")

pred_gar_test_label <- ifelse(pred_gar_te
  > 0.5, c1, c0)

gar_te_error <-
  mean(pred_gar_test_label != my_sahp_test$expensive)

gar_te_error
```

```
## [1] 0.1774194
```

```r
#Training Error Liv
pred_liv_train <- predict(glm_expensive_liv,
    newdata = my_sahp_train,
    type = "response")

pred_liv_train_label <- ifelse(pred_liv_train
  > 0.5,
  c1,
  c0)

liv_tr_error <-
  mean(pred_liv_train_label != my_sahp_train$expensive)

liv_tr_error
```

```
## [1] 0.35
```

```r
#Test Error Liv
pred_liv_te <- predict(glm_expensive_liv,
  newdata = my_sahp_test,
  type = "response")

pred_liv_test_label <- ifelse(pred_liv_te
  > 0.5,
  c1,
  c0)

liv_te_error <-
  mean(pred_liv_test_label != my_sahp_test$expensive)

liv_te_error
```

```
## [1] 0.2580645
```

```r
#Training Error OA
pred_oa_train <- predict(glm_expensive_oa,
  newdata = my_sahp_train,
  type = "response")

pred_oa_train_label <- ifelse(pred_oa_train
  > 0.5, c1, c0)

oa_tr_error <-
  mean(pred_oa_train_label != my_sahp_train$expensive)

oa_tr_error
```

```
## [1] 0.23
```

```r
#Test Error OA
pred_oa_te <- predict(glm_expensive_oa,
  newdata = my_sahp_test,
  type = "response")

pred_oa_test_label <- ifelse(pred_oa_te
  > 0.5,
  c1,
  c0)
oa_te_error <-
  mean(pred_oa_test_label != my_sahp_test$expensive)

oa_te_error
```

```
## [1] 0.3064516
```

```r
Tr_Te_Table <- data.frame(
  Variables = c("gar_car", "liv_area", "oa_qual"),
  Training_Error = c(gar_tr_error, liv_tr_error, oa_tr_error),
  Test_error = c(gar_te_error, liv_te_error, oa_te_error)
)

Tr_Te_Table
```

```
##   Variables Training_Error Test_error
## 1   gar_car           0.29  0.1774194
## 2  liv_area           0.35  0.2580645
## 3   oa_qual           0.23  0.3064516
```

- The variable **oa_qual** has the smallest **training error**.
- The variable **gar_car** has the smallest **test error**.

b. Using the training data `my_sahp_train` to fit a logistic regression model of `expensive` on all three variables (`gar_car`, `liv_area`, `oa_qual`). Compute the training and test error. How do the result compare with part a.

**Solution:**

```r
#Training Error
glm_expensive_all <- glm(expensive ~
  gar_car + liv_area + oa_qual,
  data = my_sahp_train,
  family = "binomial")

summary(glm_expensive_all)
```

```
##
## Call:
## glm(formula = expensive ~ gar_car + liv_area + oa_qual, family = "binomial
",
##     data = my_sahp_train)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.339e+01  2.899e+00  -4.621 3.83e-06 ***
## gar_car      2.066e+00  8.146e-01   2.535 0.011229 *
## liv_area     1.922e-03  8.585e-04   2.239 0.025176 *
## oa_qual      1.083e+00  3.109e-01   3.484 0.000494 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 137.628  on 99  degrees of freedom
## Residual deviance:  76.806  on 96  degrees of freedom
## AIC: 84.806
##
## Number of Fisher Scoring iterations: 6
```

```r
pred_all_train <- predict(glm_expensive_all,
  newdata = my_sahp_train,
  type = "response")

pred_all_train_label <- ifelse(pred_all_train
  > 0.5,
  c1,
  c0)

all_tr_error <- mean(pred_all_train_label !=
  my_sahp_train$expensive)

all_tr_error
```

```
## [1] 0.21
```

```r
#Test Error
pred_all_te <- predict(glm_expensive_all,
  newdata = my_sahp_test,
  type = "response")

pred_all_test_label <- ifelse(pred_all_te
  > 0.5,
  c1,
  c0)

all_te_error <- mean(pred_all_test_label !=
  my_sahp_test$expensive)

all_te_error
```

```
## [1] 0.1612903
```

```r
Tr_Te_Table_all <- data.frame(
  Model = "All variables",
  Training_Error = all_tr_error,
  Test_Error = all_te_error)

Tr_Te_Table_all
```

```
##           Model Training_Error Test_Error
## 1 All variables           0.21  0.1612903
```

```r
Tr_Te_Table
```

```
##   Variables Training_Error Test_error
## 1   gar_car           0.29  0.1774194
## 2  liv_area           0.35  0.2580645
## 3   oa_qual           0.23  0.3064516
```

- The model using all three variables has the **lowest training error** and the **lowest test error**. The decrease in the test error only dropped slightly compared to the lowest test error of the single variable models, gar_car.
- This shows that the **predictive power** of our model of **all three variables** is **more effective** than our model with ***only one variable***.

*Q2*

Using the training data `my_sahp_train` to fit LDA and QDA models of `expensive` on all three variables (`gar_car`, `liv_area`, `oa_qual`). Compute the training and test error. How do the results compare with Q1?

**Solution:**

```r
library(MASS)
#LDA MODEL
lda_expensive_all <- lda(expensive ~
  gar_car + liv_area + oa_qual,
  data = my_sahp_train)
lda_expensive_all
```

```
## Call:
## lda(expensive ~ gar_car + liv_area + oa_qual, data = my_sahp_train)
##
## Prior probabilities of groups:
## FALSE  TRUE
##  0.55  0.45
##
## Group means:
##        gar_car liv_area  oa_qual
## FALSE 1.381818 1275.255 5.418182
## TRUE  2.133333 1686.333 7.000000
##
## Coefficients of linear discriminants:
##                    LD1
## gar_car  0.6842232584
## liv_area 0.0005864637
## oa_qual  0.5960637814
```

```r
lda_train_label <- predict(lda_expensive_all,
  newdata = my_sahp_train)$class

lda_tr_error <-mean(lda_train_label !=
  my_sahp_train$expensive)
lda_tr_error
```

```
## [1] 0.17
```

```r
lda_pred_test <- predict(lda_expensive_all,
  newdata = my_sahp_test)

lda_te_error <- mean(lda_pred_test$class !=
  my_sahp_test$expensive)
lda_te_error
```

```
## [1] 0.2419355
```

```r
#QDA MODEL

qda_expensive_all <- qda(expensive ~
  gar_car + liv_area + oa_qual,
  data = my_sahp_train)
qda_expensive_all
```

```
## Call:
## qda(expensive ~ gar_car + liv_area + oa_qual, data = my_sahp_train)
##
## Prior probabilities of groups:
## FALSE  TRUE
##  0.55  0.45
##
## Group means:
##        gar_car liv_area  oa_qual
## FALSE 1.381818 1275.255 5.418182
## TRUE  2.133333 1686.333 7.000000
```

```
qda_train_label <- predict(qda_expensive_all,
  newdata = my_sahp_train)$class

qda_tr_error <- mean(qda_train_label !=
  my_sahp_train$expensive)
qda_tr_error
```

```
## [1] 0.16
```

```
qda_test_label <- predict(qda_expensive_all,
  newdata = my_sahp_test)$class

qda_te_error <- mean(qda_test_label !=
  my_sahp_test$expensive)
qda_te_error
```

```
## [1] 0.1774194
```

```
Tr_Te_Table_LDA_QDA <- data.frame(
  Model = c("LDA","QDA"),
  Training_Error = c(lda_tr_error,
  qda_tr_error),
  Test_Error = c(lda_te_error,
  qda_te_error)
)

Tr_Te_Table_LDA_QDA
```

```
##   Model Training_Error Test_Error
## 1   LDA           0.17  0.2419355
## 2   QDA           0.16  0.1774194
```

- Both the **LDA** and **QDA** models have a **lower training error** than the **logistic regression model**.

-The **QDA** model has the **lowest training error** at **0.16** and also a **lower test error** than **LDA**.

-The **LDA test error** is higher at **0.242**, which means it generalizes worse than **QDA**.

-The **logistic regression** and **QDA** have similar test errors at **~ 0.161 and ~ 0.177** respectively. This means they have **comparable** predictive capabilities. **LDA's test error** is slightly higher.

*Q3*

Q6 in Chapter 4 of ISLRv2 (Page 191). Hint: Use equation (4.7) on Page 137 of ISLRv2.

**Solution:**

```
#Part A
beta0 <- -6
beta1 <- 0.05
beta2 <- 1

X1 <- 40
X2 <-3.5

log_odds <- beta0 + beta1 * X1 + beta2 * X2

p <- exp(log_odds) / (1 + exp(log_odds))

log_odds

## [1] -0.5

p

## [1] 0.3775407

#Part B
given_p <- 0.5
log_odds_target <- log(given_p / (1 - given_p))
log_odds_target

## [1] 0

hours_needed <- -(beta0 + beta2 * X2) / beta1
hours_needed

## [1] 50
```

- ***(For A):*** We are given the logistic regression model on page **137**. I assigned **X1 to hours studied, X2 to GPA**. The estimated coefficients are **Beta 0 = -1, Beta 1 = 0.05, and Beta 2 = 1**.
- ***(For A):*** For a student who studies 40 hours and has a GPA of 3.5, the estimated log-odds of getting an A are **log-odds = -6 + 0.05(40) + 1(3.5) = -0.5**.
- ***(For A):*** The corresponding probability is **p = $(e^{(-0.5))/(1+e}$(-0.5)) = 0.3775**. Therefore, the estimated probability that this student gets an A is around 37.8%.

- ***(For B):*** To find how many hours the student (given a GPA of 3.5) needs to study to have a 50% chance of getting an A, I set **p = 0.5**.
- ***(For B):*** Using the equation on **page 137**, I assigned **log(p/(1-p)) = 0**, therefore **0 = -6 + 0.5X1 + 1(3.5)**. Solving for **X1**, I got **- ((-6 + 3.5)/0.05) = 50**
- ***(For B):*** Therefore, the student must study for about 50 hours to have a 50% chance of getting an A in the class.