LOGISTIC REGRESSION ON SYNTHETIC PILOT TRAINING DATA

Logistic regression on synthetic pilot training data

In this lab you will fit logistic regression models on synthetic data describing pilot training outcomes in a flight simulator. The response variable is binary:

- `success = 1` (pilot passed the simulator task),
- `success = 0` (pilot failed the simulator task).

Predictors include:

- `exam_score` (continuous),
- `prev_sim_training` (factor: whether the pilot had previous simulator training).

1. Why logistic regression?

When the response is binary, linear regression is not appropriate because predicted values can fall outside `[0, 1]`. Logistic regression models the probability of success:

$$P(success = 1 \mid x) = p,$$

with the logit link:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \cdot exam\_score + \beta_2 \cdot prev\_sim\_training.$$

In R:

```
glm(success ~ exam_score + prev_sim_training, family = binomial, data = df)
```

2. Create synthetic pilot data

We generate a realistic synthetic dataset for trainees.

```
set.seed(2026)
n <- 400

# Continuous predictor: theoretical exam score (0-100)
exam_score <- pmin(100, pmax(0, rnorm(n, mean = 68, sd = 12)))

# Factor predictor: prior simulator training
prev_sim_training <- sample(c("no", "yes"), size = n, replace = TRUE, prob = c
prev_sim_training <- factor(prev_sim_training, levels = c("no", "yes"))

# True data-generating logistic model (unknown in practice)
# Higher exam score and prior simulator training increase success chance.
eta <- -7.0 + 0.09 * exam_score + 1.1 * (prev_sim_training == "yes")
p_success <- 1 / (1 + exp(-eta))

# Binary outcome
success <- rbinom(n, size = 1, prob = p_success)

pilot_df <- data.frame(
  exam_score = exam_score,
  prev_sim_training = prev_sim_training,
  success = success
)

head(pilot_df)
str(pilot_df)
```

Quick check of success rate:

```
mean(pilot_df$success)
table(pilot_df$success)
```

3. Basic summaries and exploratory plots

3.1 Summaries by outcome

```
tapply(pilot_df$exam_score, pilot_df$success, summary)
table(pilot_df$prev_sim_training, pilot_df$success)
prop.table(table(pilot_df$prev_sim_training, pilot_df$success), margin =
```

3.2 Visualize score by success

```
boxplot(exam_score ~ success, data = pilot_df,
        names = c("fail (0)", "success (1)"),
        col = c("tomato", "lightgreen"),
        ylab = "Exam score",
        main = "Exam score by simulator outcome")
```

3.3 Success rate by prior training

```
tab <- table(pilot_df$prev_sim_training, pilot_df$success)
success_rate <- prop.table(tab, margin = 1)[, "1"]
success_rate
```

4. Fit logistic regression models

4.1 Model A: continuous predictor only

```
m_score <- glm(success ~ exam_score,
               family = binomial,
               data = pilot_df)
summary(m_score)
```

4.2 Model B: factor predictor only

```
m_train <- glm(success ~ prev_sim_training,
               family = binomial,
               data = pilot_df)
summary(m_train)
```

4.3 Model C: both predictors (continuous + factor)

```
m_both <- glm(success ~ exam_score + prev_sim_training,
              family = binomial,
              data = pilot_df)
summary(m_both)
```

Interpretation reminder:

- positive coefficient -> higher log-odds of success,
- negative coefficient -> lower log-odds of success.

5. Odds ratios and confidence intervals

For logistic regression, `exp(coef)` gives odds ratios.

```
exp(coef(m_both))
exp(confint(m_both))
```

Interpretation examples:

- `exp(beta_exam_score)` = multiplicative change in odds for a 1-point score increase.
- `exp(beta_prev_sim_trainingyes)` = odds ratio for "yes" vs "no" prior training.

6. Predicted probabilities

6.1 Predict for selected pilot profiles

```
new_pilots <- data.frame(
  exam_score = c(50, 65, 80, 80),
  prev_sim_training = factor(c("no", "no", "no", "yes"), levels = c("no"
)

predict(m_both, newdata = new_pilots, type = "response")
```

6.2 Probability curves by score and training group

```
score_grid <- seq(30, 100, by = 1)

pred_no <- data.frame(
  exam_score = score_grid,
  prev_sim_training = factor("no", levels = c("no", "yes"))
)

pred_yes <- data.frame(
  exam_score = score_grid,
  prev_sim_training = factor("yes", levels = c("no", "yes"))
)

p_no <- predict(m_both, newdata = pred_no, type = "response")
p_yes <- predict(m_both, newdata = pred_yes, type = "response")

plot(score_grid, p_no, type = "l", lwd = 2, col = "red",
     ylim = c(0, 1), xlab = "Exam score", ylab = "Predicted probability
     main = "Predicted success probability")
lines(score_grid, p_yes, lwd = 2, col = "blue")
legend("topleft", legend = c("no prior simulator training", "yes prior s
       col = c("red", "blue"), lwd = 2, bty = "n")
```

7. Compare models

7.1 Information criteria

```
AIC(m_score, m_train, m_both)
```

7.2 Likelihood-ratio tests

```
anova(m_score, m_both, test = "Chisq")
anova(m_train, m_both, test = "Chisq")
```

7.3 Simple classification check (threshold 0.5)

```
p_hat <- predict(m_both, type = "response")
y_hat <- ifelse(p_hat >= 0.5, 1, 0)

table(predicted = y_hat, observed = pilot_df$success)
mean(y_hat == pilot_df$success)  # accuracy
```

8. Tasks

Task 8.1 - Build and interpret

1. Fit `m_score`, `m_train`, and `m_both`.
2. Identify which coefficients are statistically significant in each model.
3. Explain in plain language how exam score and prior training affect success probability.

Task 8.2 - Odds ratios

1. Compute `exp(coef(m_both))`.
2. Interpret:
   - odds ratio for `exam_score`,

- odds ratio for `prev_sim_trainingyes`.

Task 8.3 - Predictions

1. Predict success probability for:
   - score 55, no training,
   - score 55, yes training,
   - score 85, no training,
   - score 85, yes training.
2. Comment how each predictor changes predictions.

Task 8.4 - Sensitivity experiment

Change data-generating parameters and re-run the lab:

- increase/decrease coefficient for `exam_score`,
- increase/decrease training effect,
- increase sample size `n`.

For each change, observe:

1. coefficient estimates,
2. p-values,
3. confidence interval widths,
4. classification accuracy.

9. Optional extension: interaction effect

Test whether exam score effect differs by training group:

```
m_int <- glm(success ~ exam_score * prev_sim_training,
             family = binomial,
             data = pilot_df)
summary(m_int)
anova(m_both, m_int, test = "Chisq")
```

Interpretation idea:

- if interaction is significant, the slope of exam score is different for trained vs untrained pilots.

10. Wrap-up

This lab demonstrates logistic regression with:

- a binary response,
- one continuous predictor (`exam_score`),
- one factor predictor (`prev_sim_training`).

Using synthetic data lets you control the true mechanism and directly observe how model outputs (coefficients, p-values, and predicted probabilities) respond.