STA 336: Statistical Machine Learning

Homework 4

Rento Saijo

February 24, 2026

Table of Contents

Disclosure

GPT-5.3-Codex was used to create the `YAML` portion and some `LaTeX` code to format the text/equations nicely. Page formatting code was also provided by Derin Gezgin. In the setup chunk, libraries were loaded and some helper functions were defined including but not limited to `table_latex()` and `with_family()`. See the original `RMD` file here for more details.

Problem 1

PROBLEM 1

Use the full dataset and build a model using a single predictor on student. What is the mathematical form of the model?

Cell 1

```r
# Load data.
default_df <- ISLR2::Default

# Fit student-only logistic model.
model_q1 <- stats::glm(default ~ student, data = default_df, family = stats::binomial)

# Build coefficient table.
coef_q1_tbl <- tibble::tibble(
  term     = names(stats::coef(model_q1)),
  estimate = as.numeric(stats::coef(model_q1))
) %>%
  dplyr::mutate(estimate = round(estimate, 6))

# Print coefficient table.
coef_q1_tbl %>%
  table_latex(
    col_names = c('Term', 'Estimate'),
    caption   = 'Logistic model coefficients with student only.'
  )
```

Table 1: Logistic model coefficients with student only.

Term	Estimate
(Intercept)	-3.504128
studentYes	0.404887

The fitted model is

$$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -3.504128 + 0.404887 \cdot \mathrm{I}\{\texttt{student} = \texttt{Yes}\},$$

where $\hat{p} = \Pr(\texttt{default} = \texttt{Yes} \mid \texttt{student})$. Equivalently,

$$\hat{p} = \frac{\exp(-3.504128 + 0.404887 \cdot \mathrm{I}\{\texttt{student} = \texttt{Yes}\})}{1 + \exp(-3.504128 + 0.404887 \cdot \mathrm{I}\{\texttt{student} = \texttt{Yes}\})}.$$

Problem 2

Problem 2

How would you interpret the coefficient of student on the model of Problem 1?

The coefficient on `studentYes` is 0.404887, so the log-odds of default are higher for students than non-students by 0.404887 in this one-predictor model. The corresponding odds ratio is

$$\exp(0.404887) = 1.4991.$$

Therefore, the model estimates that students have about $1.50\times$ the odds of default relative to non-students.

Problem 3

Problem 3

What is the confusion matrix when you use the cutoff point to be 0.5? What is the error rate of prediction?

Cell 2

```r
# Build predictions with 0.5 cutoff.
prob_q1   <- stats::predict(model_q1, type = 'response')
pred_q1   <- factor(ifelse(prob_q1 > 0.5, 'Yes', 'No'), levels = c('No', 'Yes'))
actual_q1 <- factor(default_df$default, levels = c('No', 'Yes'))

# Build confusion matrix.
cm_q1     <- table(actual = actual_q1, predicted = pred_q1)
cm_q1_tbl <- as.data.frame.matrix(cm_q1) %>%
  tibble::rownames_to_column(var = 'Actual')

# Compute error.
error_q1 <- mean(pred_q1 != actual_q1)
error_q1_tbl <- tibble::tibble(
  metric = 'Prediction error rate',
  value  = round(error_q1, 4)
)

# Print tables.
cm_q1_tbl %>%
  table_latex(
    col_names = c('Actual', 'Predicted No', 'Predicted Yes'),
    caption   = 'Confusion matrix for student-only model (cutoff = 0.5).'
  )
```

Table 2: Confusion matrix for student-only model (cutoff = 0.5).

Actual	Predicted No	Predicted Yes
No	9667	0
Yes	333	0

Cell 3

```r
error_q1_tbl %>%
  table_latex(
    col_names = c('Metric', 'Value'),
    caption   = 'Error rate for student-only model (cutoff = 0.5).'
  )
```

Table 3: Error rate for student-only model (cutoff = 0.5).

Metric	Value
Prediction error rate	0.0333

At cutoff 0.5, this model predicts every observation as `No`. The prediction error rate is $\boxed{0.0333}$ (about 3.33%).

Problem 4

Problem 4

Use the full dataset and build a model using a single predictor on balance. What is the mathematical form of the model?

Cell 4

```r
# Fit balance-only logistic model.
model_q4 <- stats::glm(default ~ balance, data = default_df, family = stats::binomial)

# Build coefficient table.
coef_q4_tbl <- tibble::tibble(
  term     = names(stats::coef(model_q4)),
  estimate = as.numeric(stats::coef(model_q4))
) %>%
  dplyr::mutate(estimate = round(estimate, 6))

# Print coefficient table.
coef_q4_tbl %>%
  table_latex(
    col_names = c('Term', 'Estimate'),
    caption   = 'Logistic model coefficients with balance only.'
  )
```

Table 4: Logistic model coefficients with balance only.

Term	Estimate
(Intercept)	-10.651331
balance	0.005499

The fitted model is

$$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -10.651331 + 0.005499\,\texttt{balance},$$

where $\hat{p} = \Pr(\texttt{default} = \texttt{Yes} \mid \texttt{balance})$. Equivalently,

$$\hat{p} = \frac{\exp(-10.651331 + 0.005499\,\texttt{balance})}{1 + \exp(-10.651331 + 0.005499\,\texttt{balance})}.$$

Problem 5

Problem 5

How would you interpret the coefficient of balance on the model of Problem 4?

The coefficient on `balance` is 0.005499, so each one-unit increase in balance increases the log-odds of default by 0.005499. In odds terms,

$$\exp(0.005499) = 1.0055,$$

so each $1 increase in balance multiplies the odds of default by about 1.0055.

Problem 6

Problem 6

What is the confusion matrix of the model in Problem 4 when you use the cutoff point to be 0.5? What is the error rate of prediction?

Cell 5

```r
# Build predictions with 0.5 cutoff.
prob_q4   <- stats::predict(model_q4, type = 'response')
pred_q4   <- factor(ifelse(prob_q4 > 0.5, 'Yes', 'No'), levels = c('No', 'Yes'))
actual_q4 <- factor(default_df$default, levels = c('No', 'Yes'))

# Build confusion matrix.
cm_q4 <- table(actual = actual_q4, predicted = pred_q4)
cm_q4_tbl <- as.data.frame.matrix(cm_q4) %>%
  tibble::rownames_to_column(var = 'Actual')

# Compute error.
error_q4 <- mean(pred_q4 != actual_q4)
error_q4_tbl <- tibble::tibble(
  metric = 'Prediction error rate',
  value  = round(error_q4, 4)
)

# Print tables.
cm_q4_tbl %>%
  table_latex(
    col_names = c('Actual', 'Predicted No', 'Predicted Yes'),
    caption   = 'Confusion matrix for balance-only model (cutoff = 0.5).'
  )
```

Table 5: Confusion matrix for balance-only model (cutoff = 0.5).

Actual	Predicted No	Predicted Yes
No	9625	42
Yes	233	100

Cell 6

```r
error_q4_tbl %>%
  table_latex(
    col_names = c('Metric', 'Value'),
    caption   = 'Error rate for balance-only model (cutoff = 0.5).'
  )
```

Table 6: Error rate for balance-only model (cutoff = 0.5).

Metric	Value
Prediction error rate	0.0275

The balance-only model has prediction error $\boxed{0.0275}$ (about 2.75%).

Problem 7

Problem 7

Split the data into halves and use one half as the training and the other half as the test. Fit a model on the training data using `student`, `balance`, and `income` as predictors.

Cell 7

```r
# Create train/test split.
set.seed(20060527)
n_default     <- nrow(default_df)
train_idx     <- sample(seq_len(n_default), n_default / 2)
default_train <- default_df[train_idx, ]
default_test  <- default_df[-train_idx, ]

# Fit training model.
model_q7 <- stats::glm(
  default ~ student + balance + income,
  data   = default_train,
  family = stats::binomial
)

# Build split table.
split_q7_tbl <- tibble::tibble(
  data_set     = c('Training', 'Test'),
  observations = c(nrow(default_train), nrow(default_test))
)

# Build coefficient table.
coef_q7_tbl <- as.data.frame(stats::coef(summary(model_q7))) %>%
  tibble::rownames_to_column(var = 'term') %>%
  dplyr::rename(
    estimate  = Estimate,
    std_error = `Std. Error`,
    z_value   = `z value`,
    p_value   = `Pr(>|z|)`
  ) %>%
  dplyr::mutate(dplyr::across(-term, ~ round(.x, 6)))

# Create probabilities for later questions.
prob_train_q7 <- stats::predict(model_q7, newdata = default_train, type = 'response')
prob_test_q7  <- stats::predict(model_q7, newdata = default_test, type = 'response')

# Print tables.
split_q7_tbl %>%
  table_latex(
    col_names = c('Data set', 'Observations'),
    caption   = 'Train/test split sizes.'
  )
```

Table 7: Train/test split sizes.

Data set	Observations
Training	5000
Test	5000

Cell 8

```
coef_q7_tbl %>%
  table_latex(
    col_names = c('Term', 'Estimate', 'Std. Error', 'z value', 'Pr(>|z|)'),
    caption   = 'Logistic regression coefficient summary.'
  )
```

Table 8: Logistic regression coefficient summary.

Term	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-11.362494	0.729852	-15.568209	0.000000
studentYes	-0.439823	0.345068	-1.274598	0.202452
balance	0.005885	0.000340	17.319003	0.000000
income	0.000008	0.000012	0.693137	0.488224

Problem 8

Problem 8

What is the training error of the model in Problem 7?

Cell 9

```r
# Build training predictions.
pred_train_q7   <- factor(ifelse(prob_train_q7 > 0.5, 'Yes', 'No'), levels = c('No', 'Yes'))
actual_train_q7 <- factor(default_train$default, levels = c('No', 'Yes'))

# Compute training error.
train_error_q7  <- mean(pred_train_q7 != actual_train_q7)
train_error_tbl <- tibble::tibble(
  metric = 'Training error rate',
  value  = round(train_error_q7, 4)
)

# Print training error table.
train_error_tbl %>%
  table_latex(
    col_names = c('Metric', 'Value'),
    caption   = 'Training error with cutoff = 0.5.'
  )
```

Table 9: Training error with cutoff = 0.5.

Metric	Value
Training error rate	0.0264

The training error is $\boxed{0.0264}$ (about 2.64%).

Problem 9

Problem 9

What is the test error of the model in Problem 7?

Cell 10

```r
# Build test predictions.
pred_test_q7   <- factor(ifelse(prob_test_q7 > 0.5, 'Yes', 'No'), levels = c('No', 'Yes'))
actual_test_q7 <- factor(default_test$default, levels = c('No', 'Yes'))

# Compute test error.
test_error_q7  <- mean(pred_test_q7 != actual_test_q7)
test_error_tbl <- tibble::tibble(
  metric = 'Test error rate',
  value  = round(test_error_q7, 4)
)

# Print test error table.
test_error_tbl %>%
  table_latex(
    col_names = c('Metric', 'Value'),
    caption   = 'Test error with cutoff = 0.5.'
  )
```

Table 10: Test error with cutoff = 0.5.

Metric	Value
Test error rate	0.0266

The test error is $\boxed{0.0266}$ (about 2.66%).

Problem 10

Problem 10

Draw the ROC curve for model in Problem 7 and find the best cutoff point. What does the best cutoff point mean?

Cell 11

```r
# Build ROC.
roc_q7 <- Epi::ROC(
  form = default ~ student + balance + income,
  data = default_train,
  plot = 'ROC',
  MX   = TRUE
)
```

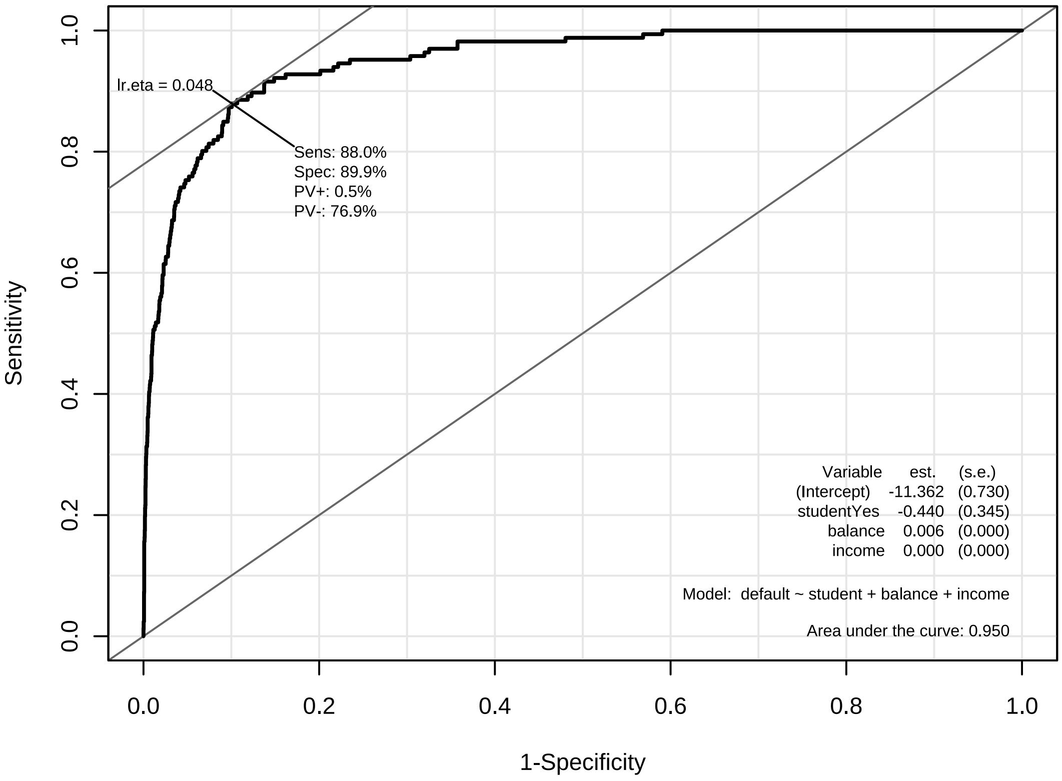

Figure 1: ROC curve on training data for the multivariable model.

Cell 12

```r
# Extract ROC grid.
roc_q7_res <- roc_q7$res

# Find best cutoff with MX criterion.
```

```r
best_row_q7 <- roc_q7_res %>%
  dplyr::mutate(mx_value = sens + spec) %>%
  dplyr::filter(mx_value == max(mx_value)) %>%
  dplyr::arrange(dplyr::desc(lr.eta)) %>%
  dplyr::slice(1)
best_cutoff_q7 <- best_row_q7$lr.eta[[1]]

# Build best-cutoff table.
best_cutoff_tbl <- tibble::tibble(
  metric = c(
    'Best cutoff',
    'Sensitivity',
    'Specificity',
    'False positive rate',
    'MX value (sensitivity + specificity)',
    'AUC'
  ),
  value  = round(c(
    best_cutoff_q7,
    best_row_q7$sens[[1]],
    best_row_q7$spec[[1]],
    1 - best_row_q7$spec[[1]],
    best_row_q7$mx_value[[1]],
    roc_q7$AUC
  ), 4)
)

# Print best-cutoff table.
best_cutoff_tbl %>%
  table_latex(
    col_names = c('Metric', 'Value'),
    caption   = 'Best cutoff chosen from training ROC.'
  )
```

Table 11: Best cutoff chosen from training ROC.

Metric	Value
Best cutoff	0.0477
Sensitivity	0.8795
Specificity	0.8995
False positive rate	0.1005
MX value (sensitivity + specificity)	1.7790
AUC	0.9498

Using (`Epi::ROC` with `MX = TRUE`), the best cutoff is $\boxed{0.0477}$. This is the threshold that maximizes sensitivity + specificity on the training data.

Problem 11

Problem 11

Use the best cutoff point on the test data. What is the test error now?

Cell 13

```r
# Build test predictions with best cutoff.
pred_test_best_q7 <- factor(ifelse(prob_test_q7 > best_cutoff_q7, 'Yes', 'No'), levels = c('No', 'Yes'))

# Compute test error with best cutoff.
test_error_best_q7 <- mean(pred_test_best_q7 != actual_test_q7)
test_error_best_tbl <- tibble::tibble(
  metric = 'Test error rate (best cutoff)',
  value = round(test_error_best_q7, 4)
)

# Print test error table.
test_error_best_tbl %>%
  table_latex(
    col_names = c('Metric', 'Value'),
    caption = 'Test error using best cutoff from training ROC.'
  )
```

Table 12: Test error using best cutoff from training ROC.

Metric	Value
Test error rate (best cutoff)	0.1016

Using the best cutoff from Problem 10, the test error is $\boxed{0.1016}$ (about 10.16%).

Problem 12

Problem 12

For a student, with 0 balance, and 20000 income, what is the predicted probability of default? Use the model on training data.

Cell 14

```r
# Build student profile.
new_student_q12 <- data.frame(
  student = factor('Yes', levels = levels(default_train$student)),
  balance = 0,
  income  = 20000
)

# Compute predicted probability.
p_q12 <- as.numeric(stats::predict(model_q7, newdata = new_student_q12, type = 'response'))
q12_tbl <- tibble::tibble(
  metric = 'Predicted probability',
  value  = signif(p_q12, 6)
)

# Print probability table.
q12_tbl %>%
  table_latex(
    col_names = c('Metric', 'Value'),
    caption   = 'Predicted probability for student profile.'
  )
```

Table 13: Predicted probability for student profile.

Metric	Value
Predicted probability	0.0000088

The predicted probability is $\boxed{0.00000882}$.

Problem 13

Problem 13

For a non-student, with 0 balance, and 20000 income, what is the predicted probability of default? Use the model on training data.

Cell 15

```r
# Build non-student profile.
new_nonstudent_q13 <- data.frame(
  student = factor('No', levels = levels(default_train$student)),
  balance = 0,
  income  = 20000
)

# Compute predicted probability.
p_q13 <- as.numeric(stats::predict(model_q7, newdata = new_nonstudent_q13, type = 'response'))
q13_tbl <- tibble::tibble(
  metric = 'Predicted probability',
  value  = signif(p_q13, 6)
)

# Print probability table.
q13_tbl %>%
  table_latex(
    col_names = c('Metric', 'Value'),
    caption   = 'Predicted probability for non-student profile.'
  )
```

Table 14: Predicted probability for non-student profile.

Metric	Value
Predicted probability	0.0000137

The predicted probability is $\boxed{0.00001369}$.

Problem 14

Problem 14

What is the difference between the predicted values of Problems 12 and 13? Does this difference come from the coefficient of the student variable? How are they related?

Cell 16

```r
# Compute difference and student effect.
diff_q14 <- p_q12 - p_q13
beta_student_q14 <- as.numeric(stats::coef(model_q7)['studentYes'])
odds_ratio_student_q14 <- exp(beta_student_q14)

# Build relation table.
q14_tbl <- tibble::tibble(
  metric = c(
    'p(student = Yes) - p(student = No)',
    'student coefficient (log-odds scale)',
    'student odds ratio'
  ),
  value = c(diff_q14, beta_student_q14, odds_ratio_student_q14)
) %>%
  dplyr::mutate(value = signif(value, 6))

# Print relation table.
q14_tbl %>%
  table_latex(
    col_names = c('Metric', 'Value'),
    caption   = 'Probability difference and student coefficient relation.'
  )
```

Table 15: Probability difference and student coefficient relation.

Metric	Value
p(student = Yes) - p(student = No)	-0.0000049
student coefficient (log-odds scale)	-0.4398230
student odds ratio	0.6441500

The difference is $\boxed{p_{12} - p_{13} = -4.87 \times 10^{-6}}$, so the student profile has a slightly lower predicted default probability here. This difference is directly related to the coefficient. Let

$$r = \frac{p_{12}/(1-p_{12})}{p_{13}/(1-p_{13})} = e^{\hat{\beta}_{\texttt{studentYes}}} = 0.6442.$$

Solving this for p_{12} in terms of p_{13},

$$p_{12} = \frac{r\,p_{13}}{1 - p_{13} + r\,p_{13}},$$

so

$$p_{12} - p_{13} = \frac{p_{13}(1 - p_{13})(r - 1)}{1 - p_{13} + r\, p_{13}}.$$

This shows the probability difference comes from the student coefficient via $r = e^{\hat{\beta}_{\texttt{studentYes}}}$. Plugging in $r = 0.6442$ and $p_{13} = 1.368903 \times 10^{-5}$ gives

$$p_{12} - p_{13} \approx -4.87 \times 10^{-6},$$

which matches the observed difference.