Homework 2

Yuhuan Lyu

02, 2025

```
library(tidyverse)
library(ISLR)
library(ROCR)
library(MASS)
```

Linear Regression

Question 1.1

```
data(Auto)
Auto$origin <- factor(Auto$origin)

lm_fit <- lm(mpg ~ . - name, data = Auto)
summary(lm_fit)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.009 -2.078 -0.098  1.986 13.361
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.80e+01   4.68e+00   -3.84  0.00014 ***
## cylinders    -4.90e-01   3.21e-01   -1.52  0.12821
## displacement  2.40e-02   7.65e-03    3.13  0.00186 **
## horsepower   -1.82e-02   1.37e-02   -1.33  0.18549
## weight       -6.71e-03   6.55e-04  -10.24  < 2e-16 ***
## acceleration  7.91e-02   9.82e-02    0.81  0.42110
## year          7.77e-01   5.18e-02   15.01  < 2e-16 ***
## origin2       2.63e+00   5.66e-01    4.64  4.7e-06 ***
## origin3       2.85e+00   5.53e-01    5.16  3.9e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.31 on 383 degrees of freedom
## Multiple R-squared:  0.824,  Adjusted R-squared:  0.821
## F-statistic:  224 on 8 and 383 DF,  p-value: <2e-16
```

With a 0.01 threshold, we can reject the null hypothesis of no linear association since the overall F-statistic p-value is much smaller than 0.01.

Question 1.2

```
train_predictions <- predict(lm_fit, newdata = Auto)
train_mse <- mean((Auto$mpg - train_predictions)^2)
train_mse
```

```
## [1] 10.68
```

We cannot calculate the test MSE because we used the entire dataset for training without holding out a test set.

Question 1.3

```
range(Auto$year)
```

```
## [1] 70 82
```

```
new_car <- data.frame(
  cylinders = 4,
  displacement = 133,
  horsepower = 117,
  weight = 3250,
  acceleration = 29,
  year = 97,
  origin = factor(2),
  name = Auto$name[1]
)

predicted_mpg <- predict(lm_fit, newdata = new_car)
predicted_mpg
```

```
##     1
## 39.64
```

Question 1.4

```
coef_summary <- summary(lm_fit)$coefficients

japanese_diff <- coef_summary["origin3", "Estimate"]
european_diff <- coef_summary["origin2", "Estimate"]

japanese_diff
```

```
## [1] 2.853
```

```
european_diff
```

```
## [1] 2.63
```

Holding all other features fixed, Japanese cars have on average 2.85 mpg higher than American cars, and European cars have on average 2.63 mpg higher than American cars.

Question 1.5

```
hp_coef <- coef_summary["horsepower", "Estimate"]
change_30_hp <- 30 * hp_coef
change_30_hp
```

```
## [1] -0.5455
```

Holding all other predictor variables fixed, a 30-unit increase in horsepower is associated with a decrease of 0.55 mpg on average.

Algae Classification using Logistic regression

```
algae <- read_table2("algaeBloom.txt",
                     col_names = c('season','size','speed','mxPH','mnO2','Cl',
                                   'NO3','NH4','oPO4','PO4','Chla','a1','a2',
                                   'a3','a4','a5','a6','a7'),
                     na = "XXXXXXX")
```

```
## Warning: `read_table2()` was deprecated in readr 2.0.0.
## i Please use `read_table()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   season = col_character(),
##   size = col_character(),
##   speed = col_character(),
##   mxPH = col_double(),
##   mnO2 = col_double(),
##   Cl = col_double(),
##   NO3 = col_double(),
##   NH4 = col_double(),
##   oPO4 = col_double(),
##   PO4 = col_double(),
##   Chla = col_double(),
##   a1 = col_double(),
##   a2 = col_double(),
##   a3 = col_double(),
##   a4 = col_double(),
##   a5 = col_double(),
##   a6 = col_double(),
##   a7 = col_double()
## )
```

```
algae.transformed <- algae %>% mutate_at(vars(4:11), funs(log(.)))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
```

```
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
algae.transformed <- algae.transformed %>%
  mutate_at(vars(4:11),funs(ifelse(is.na(.),median(.,na.rm=TRUE),.)))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
# a1 == 0 means low
algae.transformed <- algae.transformed %>% mutate(a1 = factor(as.integer(a1 > 5), levels = c(0, 1)))

calc_error_rate <- function(predicted.value, true.value){
  return(mean(true.value != predicted.value))
}

set.seed(1)
test.indices <- sample(1:nrow(algae.transformed), 50)
algae.train <- algae.transformed[-test.indices,]
algae.test <- algae.transformed[test.indices,]
```

Question 2.1

Prove that the inverse of the logistic function is the logit function.

Starting with $p(z) = \frac{e^z}{1+e^z}$, we solve for z:

$$p(1+e^z) = e^z$$

$$p + pe^z = e^z$$

$$p = e^z - pe^z = e^z(1-p)$$

$$\frac{p}{1-p} = e^z$$

$$z = \ln\left(\frac{p}{1-p}\right)$$

Therefore, $z(p) = \ln\left(\frac{p}{1-p}\right)$ is indeed the inverse of the logistic function.

Question 2.2

Assume that $z = \beta_0 + \beta_1 x_1$, and $p = \text{logistic}(z)$.

How does the odds change when x_1 increases by 2?

The odds ratio is $\frac{p}{1-p} = e^z = e^{\beta_0+\beta_1 x_1}$. When x_1 increases by 2:

$$\frac{\text{odds}_{\text{new}}}{\text{odds}_{\text{old}}} = \frac{e^{\beta_0+\beta_1(x_1+2)}}{e^{\beta_0+\beta_1 x_1}} = e^{2\beta_1}$$

The odds are multiplied by a factor of $e^{2\beta_1}$.

If β_1 is negative, what value does p approach as $x_1 \to \infty$?

When $\beta_1 < 0$ and $x_1 \to \infty$, we have $z = \beta_0 + \beta_1 x_1 \to -\infty$, thus $p = \frac{e^z}{1+e^z} \to 0$.

What value does p approach as $x_1 \to -\infty$?

When $x_1 \to -\infty$, we have $z \to +\infty$, thus $p \to 1$.

Question 2.3

```
glm.fit <- glm(a1 ~ . - a2 - a3 - a4 - a5 - a6 - a7,
               family = binomial,
               data = algae.train)

train.probs <- predict(glm.fit, newdata = algae.train, type = "response")
train.preds <- ifelse(train.probs > 0.5, 1, 0)
train.error <- calc_error_rate(train.preds, algae.train$a1)

test.probs <- predict(glm.fit, newdata = algae.test, type = "response")
test.preds <- ifelse(test.probs > 0.5, 1, 0)
test.error <- calc_error_rate(test.preds, algae.test$a1)

train.error
```

```
## [1] 0.2
```

```
test.error
```

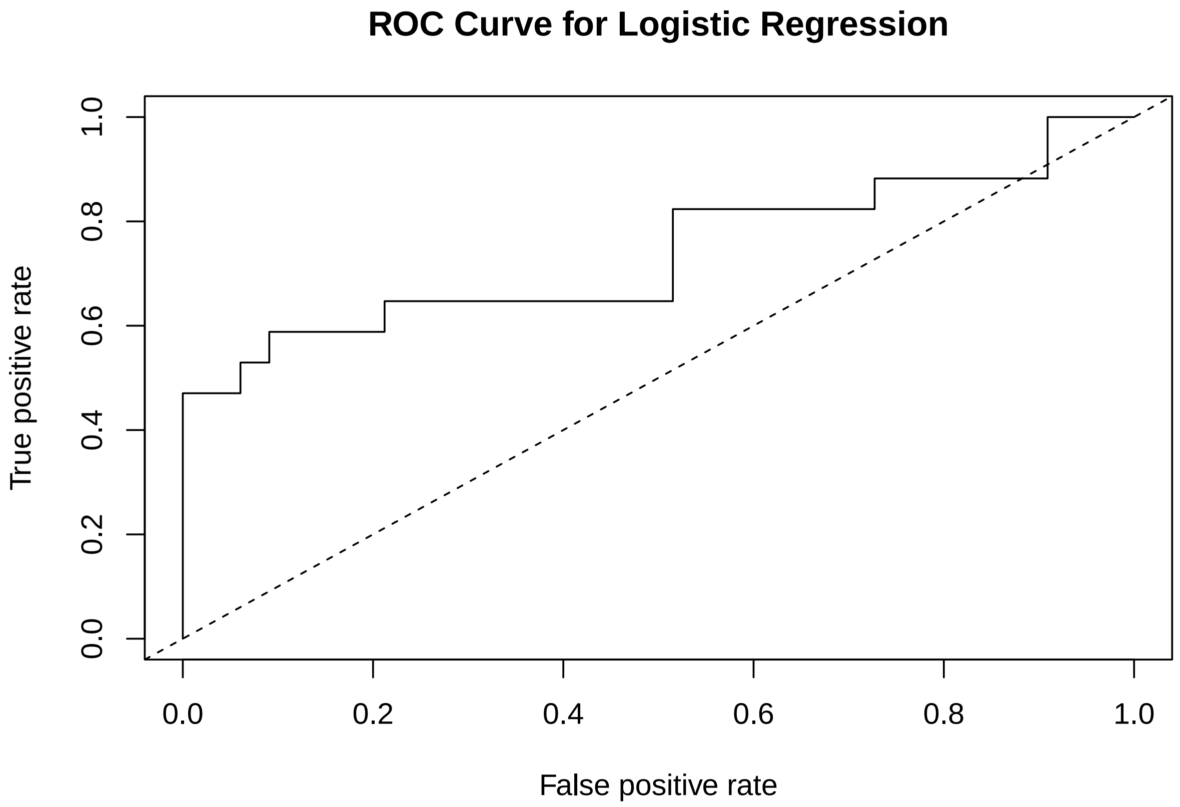
```
## [1] 0.3
```

The training error rate is 0.2 and the test error rate is 0.3.

Question 2.4

```
pred.obj <- prediction(test.probs, algae.test$a1)
perf.obj <- performance(pred.obj, "tpr", "fpr")

plot(perf.obj, main = "ROC Curve for Logistic Regression")
abline(0, 1, lty = 2)
```



```
auc <- performance(pred.obj, "auc")
auc.value <- auc@y.values[[1]]
auc.value
```

```
## [1] 0.738
```

The area under the ROC curve (AUC) is 0.738, indicating the model's discriminative ability.

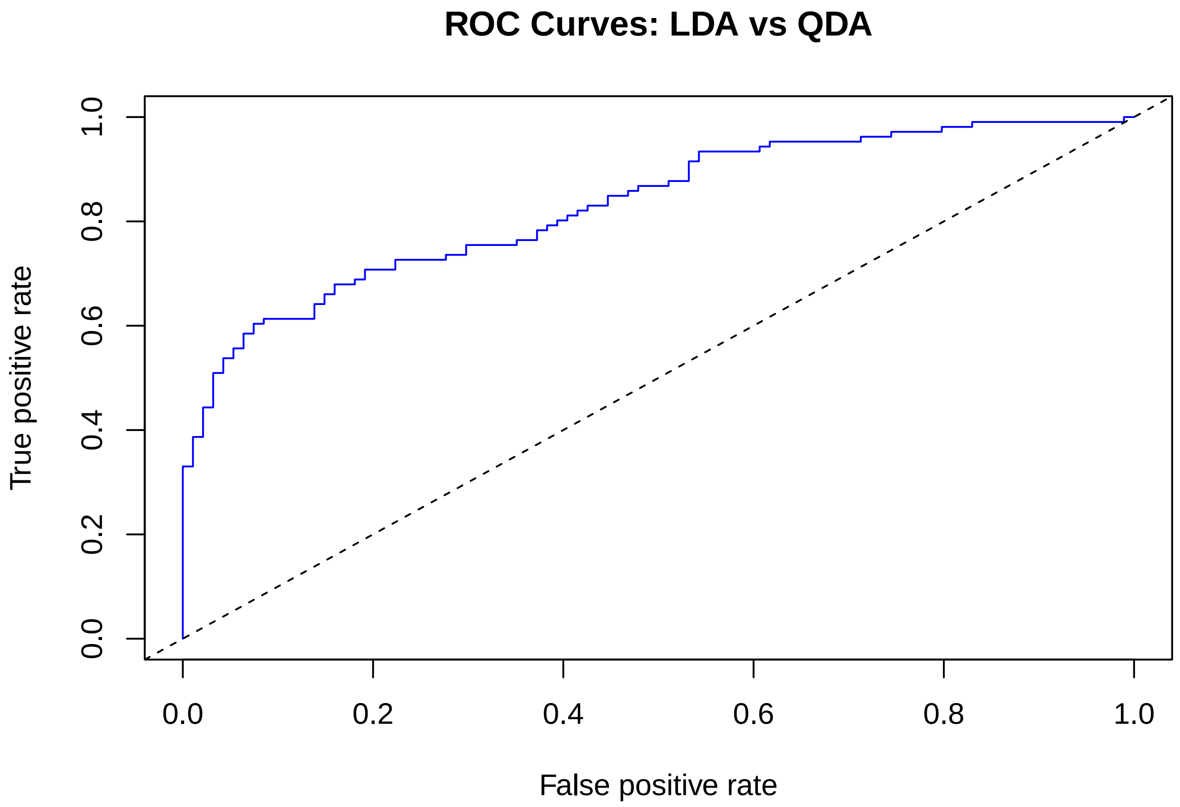
Algae Classification using Discriminant Analysis

Question 3.1

```
lda.fit <- lda(a1 ~ . - a2 - a3 - a4 - a5 - a6 - a7,
               data = algae.transformed,
               CV = TRUE)

lda.probs <- lda.fit$posterior[, 2]
lda.pred.obj <- prediction(lda.probs, algae.transformed$a1)
lda.perf.obj <- performance(lda.pred.obj, "tpr", "fpr")

plot(lda.perf.obj, main = "ROC Curves: LDA vs QDA", col = "blue")
abline(0, 1, lty = 2)
```



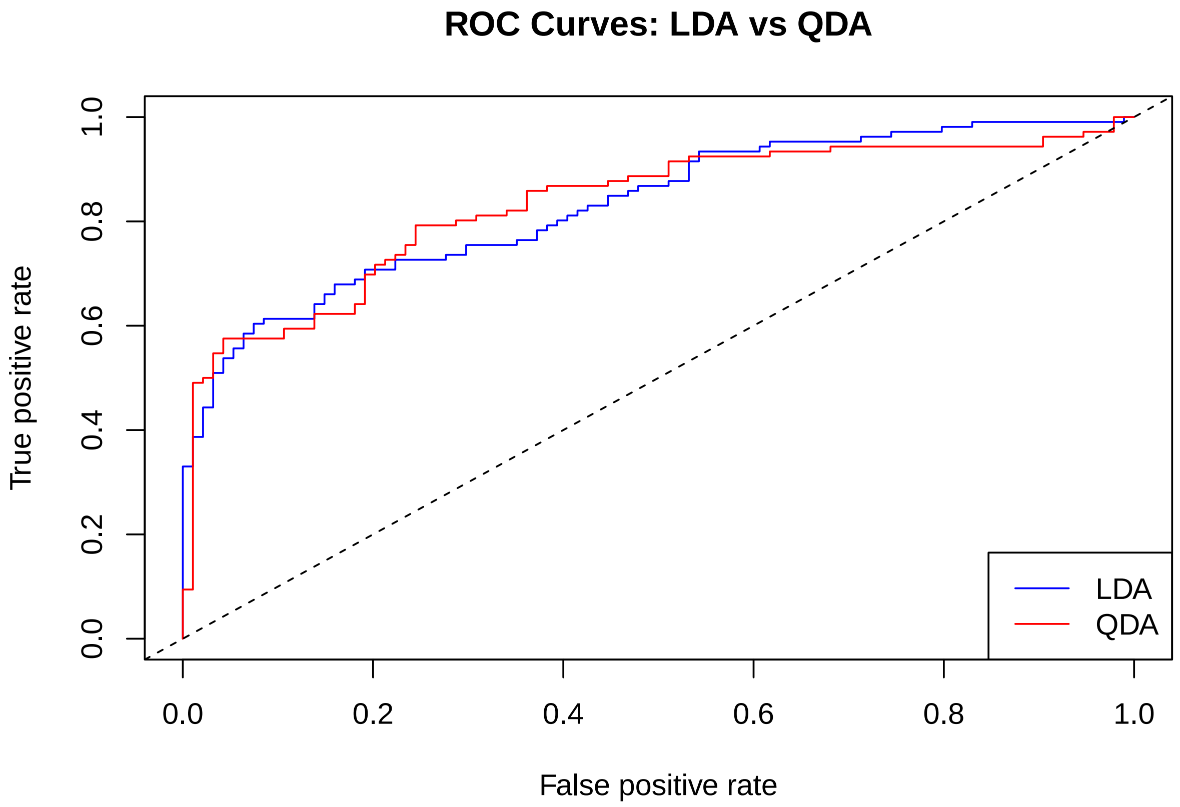
```
lda.auc <- performance(lda.pred.obj, "auc")
lda.auc.value <- lda.auc@y.values[[1]]
```

Question 3.2

```
qda.fit <- qda(a1 ~ . - a2 - a3 - a4 - a5 - a6 - a7,
               data = algae.transformed,
               CV = TRUE)

qda.probs <- qda.fit$posterior[, 2]
qda.pred.obj <- prediction(qda.probs, algae.transformed$a1)
qda.perf.obj <- performance(qda.pred.obj, "tpr", "fpr")

plot(lda.perf.obj, main = "ROC Curves: LDA vs QDA", col = "blue")
plot(qda.perf.obj, add = TRUE, col = "red")
abline(0, 1, lty = 2)
legend("bottomright", legend = c("LDA", "QDA"), col = c("blue", "red"), lty = 1)
```



```
qda.auc <- performance(qda.pred.obj, "auc")
qda.auc.value <- qda.auc@y.values[[1]]

lda.auc.value
```

```
## [1] 0.8305
```

```
qda.auc.value
```

```
## [1] 0.831
```

The LDA model has an AUC of 0.8305, while the QDA model has an AUC of 0.831. The QDA model performs better.

In terms of the bias-variance tradeoff, LDA assumes equal covariance matrices across classes (higher bias, lower variance), while QDA allows different covariance matrices (lower bias, higher variance). If the better-performing model is LDA, it suggests that the equal covariance assumption is reasonable and the reduced variance outweighs the additional bias. If QDA performs better, it indicates that the classes have sufficiently different covariance structures and the dataset is large enough to support the additional model complexity.

Fundamentals of the bootstrap

Question 4.1

For a sample of size n, each observation has a probability of $\frac{1}{n}$ of being selected in each draw. Since bootstrap sampling is with replacement and consists of n independent draws, the probability that observation j is **not** selected in any single draw is $1 - \frac{1}{n}$. Therefore, the probability that observation j is not in the bootstrap sample after n draws is:

$$P(\text{observation } j \text{ not in bootstrap sample}) = \left(1 - \frac{1}{n}\right)^n$$

Question 4.2

```
n <- 1000
prob_not_selected <- (1 - 1/n)^n
prob_not_selected
```

```
## [1] 0.3677
```

For $n = 1000$, the probability is approximately 0.3677, which is close to $e^{-1} \approx 0.368$.

Question 4.3

```
set.seed(123)
bootstrap_sample <- sample(1:1000, size = 1000, replace = TRUE)
unique_obs <- length(unique(bootstrap_sample))
missing_ratio <- 1 - unique_obs/1000
missing_ratio
```

```
## [1] 0.362
```

The ratio of missing observations is 0.362, which is close to the theoretical value of 0.3677.

Cross-validation estimate of test error

Question 5.1

```
data(Smarket)
dat <- subset(Smarket, select = -c(Year, Today))
dat$Direction <- ifelse(dat$Direction == "Up", 1, 0)

set.seed(123)
train_indices <- sample(1:nrow(dat), 700)
dat.train <- dat[train_indices, ]
dat.test <- dat[-train_indices, ]

glm.smarket <- glm(Direction ~ ., family = binomial, data = dat.train)

test.probs.smarket <- predict(glm.smarket, newdata = dat.test, type = "response")
test.preds.smarket <- ifelse(test.probs.smarket > 0.5, 1, 0)
test.error.smarket <- mean(test.preds.smarket != dat.test$Direction)

test.error.smarket
```

```
## [1] 0.4709
```

The test error rate using a single train-test split is 0.4709.

Question 5.2

```r
set.seed(123)
nfold <- 10
n <- nrow(dat)
foldid <- rep(1:nfold, each = ceiling(n/nfold))[sample(1:n)]

do.chunk <- function(chunkid, folddef, dat){
  train <- (folddef != chunkid)
  dat.train <- dat[train, ]
  dat.val <- dat[!train, ]

  fit.train <- glm(Direction ~ ., family = binomial, data = dat.train)

  pred.val <- predict(fit.train, newdata = dat.val, type = "response")
  pred.val <- ifelse(pred.val > 0.5, 1, 0)

  data.frame(fold = chunkid,
             val.error = mean(pred.val != dat.val$Direction))
}

cv.results <- lapply(1:nfold, do.chunk, folddef = foldid, dat = dat)
cv.results <- do.call(rbind, cv.results)

cv.error <- mean(cv.results$val.error)
cv.error
```

```
## [1] 0.5048
```

The estimated test error rate using 10-fold cross-validation is 0.5048.

Discriminant functions

We are asked to show that the decision boundary in QDA is quadratic by proving that $\hat{Y} = 1$ if and only if $\delta_1(x) - \delta_2(x) > M(\tau)$, where

$$\delta_k(x) = -\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) - \frac{1}{2}\log|\Sigma_k| + \log \pi_k$$

Proof:

Starting with Bayes rule:

$$\Pr(Y = 1 \mid X = x) = \frac{\pi_1 f_1(x)}{\pi_1 f_1(x) + \pi_2 f_2(x)}$$

We classify $\hat{Y} = 1$ when $\Pr(Y = 1 \mid X = x) > \tau$, which is equivalent to:

$$\frac{\pi_1 f_1(x)}{\pi_1 f_1(x) + \pi_2 f_2(x)} > \tau$$

This simplifies to:

$$\pi_1 f_1(x) > \tau[\pi_1 f_1(x) + \pi_2 f_2(x)]$$

$$(1 - \tau)\pi_1 f_1(x) > \tau \pi_2 f_2(x)$$

$$\frac{\pi_1 f_1(x)}{\pi_2 f_2(x)} > \frac{\tau}{1-\tau}$$

Taking logarithms:

$$\log \pi_1 + \log f_1(x) - \log \pi_2 - \log f_2(x) > \log\left(\frac{\tau}{1-\tau}\right)$$

For a multivariate normal density:

$$\log f_k(x) = -\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\Sigma_k| - \frac{p}{2}\log(2\pi)$$

Substituting and noting that the $-\frac{p}{2}\log(2\pi)$ terms cancel:

$$\begin{gathered} -\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1}(x-\mu_1) - \frac{1}{2}\log|\Sigma_1| + \log \pi_1 \\ - \left[-\frac{1}{2}(x-\mu_2)^T \Sigma_2^{-1}(x-\mu_2) - \frac{1}{2}\log|\Sigma_2| + \log \pi_2\right] \\ > \log\left(\frac{\tau}{1-\tau}\right) \end{gathered}$$

This is exactly:

$$\delta_1(x) - \delta_2(x) > M(\tau)$$

where $M(\tau) = \log\left(\frac{\tau}{1-\tau}\right)$.

Since $\delta_k(x)$ contains the quadratic form $(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)$, the decision boundary is indeed quadratic.

Decision threshold for $\tau = 1/2$:

When $\tau = 1/2$:

$$M(1/2) = \log\left(\frac{1/2}{1-1/2}\right) = \log(1) = 0$$

Therefore, with a probability threshold of 1/2, we classify $\hat{Y} = 1$ when $\delta_1(x) - \delta_2(x) > 0$.