Group Assignment 3

Part A (Textbook Chapter 4.8 Exercises: Q1, Q6, Q8)

Problem 1. *Using a little bit of algebra, prove that (4.2) is equivalent to (4.3). In other words, the logistic function representation and the logit representation for the logistic regression model are equivalent.*

Answer. Solution:
Let

$$p(X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X)}}.$$

We want to show this is equivalent to

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X.$$

Proof:

$$p(X) = \frac{1}{1 + e^{-z}}, \quad \text{where } z = \beta_0 + \beta_1 X.$$

Then

$$1 - p(X) = 1 - \frac{1}{1 + e^{-z}} = \frac{1 + e^{-z} - 1}{1 + e^{-z}} = \frac{e^{-z}}{1 + e^{-z}}.$$

Hence,

$$\frac{p(X)}{1 - p(X)} = \frac{\frac{1}{1+e^{-z}}}{\frac{e^{-z}}{1+e^{-z}}} = \frac{1}{e^{-z}} = e^{z}.$$

Taking the natural logarithm on both sides,

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \log(e^{z}) = z = \beta_0 + \beta_1 X.$$

Thus, (4.2) and (4.3) are indeed equivalent.

Problem 6. *Logistic Regression Probability Estimate*

Answer. Solution:

(a) Given that $\beta_0 = -6$, $\beta_1 = 0.05$, and $\beta_2 = 1$, we estimate the probability of getting an A for a student studying 40 hours with a GPA of 3.5 as follows:

$$P(Y=1) = \frac{1}{1+e^{-(-6+0.05\cdot 40+1\cdot 3.5)}} = \frac{1}{1+e^{-(-6+2+3.5)}} = \frac{1}{1+e^{-0.5}} \approx 0.378 \text{ (37.8\%)}.$$

(b) To find the number of hours a student with a GPA of 3.5 needs to study to have a 50% chance of getting an A, we set $P(Y=1) = 0.5$ and solve for X_1:

$$0.5 = \frac{1}{1+e^{-(-6+0.05\cdot X_1+1\cdot 3.5)}}$$

Simplifying:

$$\begin{aligned}
0.5(1+e^{-(-6+0.05\cdot X_1+3.5)}) &= 1 \\
1+e^{-(-6+0.05\cdot X_1+3.5)} &= 2 \\
e^{-(-6+0.05\cdot X_1+3.5)} &= 1 \\
-(-6+0.05\cdot X_1+3.5) &= 0 \\
6-0.05\cdot X_1-3.5 &= 0 \\
2.5-0.05\cdot X_1 &= 0 \\
-0.05\cdot X_1 &= -2.5 \\
X_1 &= \frac{2.5}{0.05} = 50
\end{aligned}$$

Therefore, the student would need to study 50 hours to have a 50% chance of getting an A in the class.

Problem 8. *Comparison of Logistic Regression and K-Nearest Neighbors*

Answer. Solution:

We have two classification methods:

1. **Logistic Regression:**
   - Training error: 20%
   - Test error: 30%
2. **1-Nearest Neighbor (K=1):**
   - Average error: 18%

Even though KNN (K=1) has a lower average error, it is prone to overfitting and does not generalize well. Logistic regression, despite having a higher test error, is more stable and interpretable.

Thus, logistic regression is the better choice for classifying new observations in this case. However, using a better K value (e.g., K=5 or K=10) for KNN might improve its performance.

Part B (Stock Market Data: Logistic Regression & LDA)

Problem 1. *(a)–(d) Logistic Regression on the Stock Market Data*

Answer. Solution:

(a) Compute the testing error rate using all predictors Lag1, Lag2, Lag3, Lag4, Lag5.

(b) Identify which predictors can be removed to reduce the testing error (based on p-values or other criteria).

(c) Recompute the testing error after removing the less significant predictors.

(d) Given Lag1 $= 2.1$ and Lag2 $= -0.5$, calculate the predicted probability of the market going up.

Problem 2. *(a)–(c) LDA on the Stock Market Data*

Answer. Solution:

(a) Calculate $\Pr(Y = \text{UP})$ and $\Pr(Y = \text{DOWN})$ based on the training set.

(b) Compute the mean vector of $\mathbf{X}$ (the predictors) for each class (UP vs. DOWN).

(c) Discuss whether using a 70% posterior probability threshold ($\Pr(Y = \text{UP}|\mathbf{X} = x) \geq 0.70$) is feasible or advisable for predicting a market increase.

Appendix: Screenshots

```
> # Load necessary libraries
> library(ISLR)  # Contains the Stock Market dataset
> library(MASS)  # For LDA
> # Load the data
> data(Smarket)
> # Split data into training (Year < 2005) and testing (Year == 2005)
> train <- Smarket$Year < 2005
> test <- Smarket$Year == 2005
> # Logistic Regression with all predictors
> logit_model <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5,
+                    data = Smarket, subset = train, family = binomial)
> # Predict on test data
> logit_probs <- predict(logit_model, Smarket[test, ], type = "response")
> logit_pred <- ifelse(logit_probs > 0.5, "Up", "Down")
> # Compute the testing error rate
> logit_error_rate <- mean(logit_pred != Smarket$Direction[test])
> print(paste("Test Error Rate (All Predictors):", logit_error_rate))
[1] "Test Error Rate (All Predictors): 0.412698412698413"
> # Remove least significant predictors
> logit_model_reduced <- glm(Direction ~ Lag1 + Lag2,
+                            data = Smarket, subset = train, family = binomial)
> # Predict on test data using reduced model
> logit_probs_reduced <- predict(logit_model_reduced, Smarket[test, ], type = "response")
> logit_pred_reduced <- ifelse(logit_probs_reduced > 0.5, "Up", "Down")
> # Compute the new testing error rate
> logit_error_rate_reduced <- mean(logit_pred_reduced != Smarket$Direction[test])
> print(paste("Test Error Rate (Reduced Model):", logit_error_rate_reduced))
[1] "Test Error Rate (Reduced Model): 0.44047619047619"
> # Compute predicted probability for given Lag1 = 2.1, Lag2 = -0.5
> new_data <- data.frame(Lag1 = 2.1, Lag2 = -0.5)
> predicted_prob <- predict(logit_model_reduced, new_data, type = "response")
> print(paste("Predicted Probability of Market Going Up:", predicted_prob))
[1] "Predicted Probability of Market Going Up: 0.484419143967993"
> # Linear Discriminant Analysis (LDA)
> lda_model <- lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)
> lda_pred <- predict(lda_model, Smarket[test, ])
> lda_class <- lda_pred$class
> # Compute LDA error rate
> lda_error_rate <- mean(lda_class != Smarket$Direction[test])
> print(paste("LDA Test Error Rate:", lda_error_rate))
[1] "LDA Test Error Rate: 0.44047619047619"
```

Figure 1: Screenshot 1

```
> # Compute prior probabilities
> lda_prior <- lda_model$prior
> print("Prior Probabilities:")
[1] "Prior Probabilities:"
> print(lda_prior)
    Down        Up 
0.491984 0.508016 
> # Compute class means
> lda_means <- lda_model$means
> print("Class Means:")
[1] "Class Means:"
> print(lda_means)
            Lag1        Lag2
Down  0.04279022  0.03389409
Up   -0.03954635 -0.03132544
> # Assessing the 70% posterior probability threshold
> posterior_probs <- lda_pred$posterior
> pred_high_confidence <- ifelse(posterior_probs[, "Up"] > 0.7, "Up", "Down")
> print("Predictions with 70% Posterior Probability Threshold:")
[1] "Predictions with 70% Posterior Probability Threshold:"
> print(table(pred_high_confidence))
pred_high_confidence
Down 
 252 
```

Figure 2: Screenshot 2

Data	
lda_means	num [1:2, 1:2] 0.0428 -0.0395 0.0339 -0.0313
lda_model	List of 10
lda_pred	List of 3
logit_model	List of 30
logit_model_reduced	List of 30
new_data	1 obs. of 2 variables
posterior_probs	num [1:252, 1:2] 0.49 0.479 0.467 0.474 0.493 ...
Smarket	1250 obs. of 9 variables
Values	
lda_class	Factor w/ 2 levels "Down","Up": 2 2 2 2 2 2 2 2 2 2 ...
lda_error_rate	0.44047619047619
lda_prior	Named num [1:2] 0.492 0.508
logit_error_rate	0.412698412698413
logit_error_rate_reduc…	0.44047619047619
logit_pred	chr [1:252] "Up" "Up" "Up" "Up" "Up" "Up" "Up" "Up" "Up" "Up" "…
logit_pred_reduced	chr [1:252] "Up" "Up" "Up" "Up" "Up" "Up" "Up" "Up" "Up" "Up" "…
logit_probs	Named num [1:252] 0.512 0.52 0.533 0.524 0.503 ...
logit_probs_reduced	Named num [1:252] 0.51 0.521 0.533 0.526 0.507 ...
pred_high_confidence	chr [1:252] "Down" "Down" "Down" "Down" "Down" "Down" "Down" "D…
predicted_prob	Named num 0.484
test	logi [1:1250] FALSE FALSE FALSE FALSE FALSE FALSE ...
train	logi [1:1250] TRUE TRUE TRUE TRUE TRUE TRUE ...

Figure 3: Screenshot 3