

ESE 5420 Homework 4

Mohammed Raza Syed - Penn ID: 37486255

Problem 1

a) Likelihood Function for a Single Observation

The distribution of Y_i depends on a , b , σ , and x_i . Of these, only a and b are unknown. Since $Y_i \sim ax_i + b + \varepsilon_i$ and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, it follows that

$$Y_i \sim \mathcal{N}(ax_i + b, \sigma^2).$$

The probability density function for a normally distributed random variable Y_i with mean $\mu = ax_i + b$ and variance σ^2 is:

$$f(y_i|a, b, x_i, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - (ax_i + b))^2}{2\sigma^2}\right).$$

This is the likelihood function for a single observation y_i , given the parameters a , b , x_i , and σ .

b) Likelihood and Log-Likelihood for the Dataset

For the entire dataset $(x_1, y_1), \dots, (x_n, y_n)$, the likelihood function is the product of the individual likelihoods for each observation, assuming independence:

$$L(a, b, \sigma) = \prod_{i=1}^n f(y_i|a, b, x_i, \sigma).$$

Substituting the expression for $f(y_i|a, b, x_i, \sigma)$ from part (a), we have:

$$L(a, b, \sigma) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - (ax_i + b))^2}{2\sigma^2}\right).$$

This simplifies to:

$$L(a, b, \sigma) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\sum_{i=1}^n \frac{(y_i - (ax_i + b))^2}{2\sigma^2}\right).$$

The log-likelihood function, $\log L(a, b, \sigma)$, is obtained by taking the natural logarithm of the likelihood function:

$$\log L(a, b, \sigma) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - (ax_i + b))^2.$$

c) Maximum Likelihood Estimates for a and b

To find the maximum likelihood estimates for a and b , we need to maximize the log-likelihood function with respect to these parameters. Since σ is known and constant, maximizing $\log L(a, b, \sigma)$ is equivalent to minimizing the sum of squared residuals:

$$\sum_{i=1}^n (y_i - (ax_i + b))^2.$$

Partial Derivative with Respect to a The partial derivative of the sum of squared residuals with respect to a is:

$$\frac{\partial}{\partial a} \sum_{i=1}^n (y_i - (ax_i + b))^2 = -2 \sum_{i=1}^n x_i (y_i - (ax_i + b)) = 0.$$

This equation simplifies to:

$$\sum_{i=1}^n x_i y_i = a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i.$$

Partial Derivative with Respect to b The partial derivative of the sum of squared residuals with respect to b is:

$$\frac{\partial}{\partial b} \sum_{i=1}^n (y_i - (ax_i + b))^2 = -2 \sum_{i=1}^n (y_i - (ax_i + b)) = 0.$$

This equation simplifies to:

$$\sum_{i=1}^n y_i = a \sum_{i=1}^n x_i + nb.$$

Solving for a and b Now we have two equations:

$$\sum_{i=1}^n x_i y_i = a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i,$$

$$\sum_{i=1}^n y_i = a \sum_{i=1}^n x_i + nb.$$

To solve for a and b , we introduce the sample means:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{and} \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i.$$

Using the second equation, we can rewrite it as:

$$\bar{y} = a\bar{x} + b.$$

Solving for b , we find:

$$b = \bar{y} - a\bar{x}.$$

Now, substitute $b = \bar{y} - a\bar{x}$ into the first equation:

$$\sum_{i=1}^n x_i y_i = a \sum_{i=1}^n x_i^2 + (\bar{y} - a\bar{x}) \cdot \sum_{i=1}^n x_i.$$

Expanding and simplifying, we get:

$$\sum_{i=1}^n x_i y_i - n\bar{y}\bar{x} = a \left(\sum_{i=1}^n x_i^2 - n\bar{x}^2 \right).$$

We can simplify each side as follows:

- For the left side, note that $\sum_{i=1}^n x_i y_i - n\bar{y}\bar{x}$ can be rewritten as:

$$\sum_{i=1}^n x_i y_i - n\bar{y}\bar{x} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}),$$

which represents the sum of the products of deviations of x and y from their means.

- For the right side, note that $\sum_{i=1}^n x_i^2 - n\bar{x}^2$ is the sum of squared deviations of x from its mean:

$$\sum_{i=1}^n x_i^2 - n\bar{x}^2 = \sum_{i=1}^n (x_i - \bar{x})^2.$$

Thus, our equation becomes:

$$\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = a \sum_{i=1}^n (x_i - \bar{x})^2.$$

Now, solving for a gives:

$$a = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}.$$

Finally, we substitute a back to find b :

$$b = \bar{y} - a\bar{x}.$$

Therefore, the maximum likelihood estimates for a and b are:

$$a = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2},$$

$$b = \bar{y} - a\bar{x}.$$

Problem 2

Given a set of points $\{(x_i, y_i)\}$ and the line equation $y = \beta_1 x + \beta_0$, we aim to minimize the sum of the squared perpendicular distances from these points to the line. This is formulated as minimizing:

$$L(\beta_0, \beta_1) = \sum_{i=1}^n \left(\frac{y_i - \beta_1 x_i - \beta_0}{\sqrt{1 + \beta_1^2}} \right)^2$$

Expanding the loss function for differentiation, we have:

$$L(\beta_0, \beta_1) = \frac{1}{1 + \beta_1^2} \sum_{i=1}^n (y_i - \beta_1 x_i - \beta_0)^2$$

The derivative of L with respect to β_0 while treating $\sqrt{1 + \beta_1^2}$ as a constant with respect to β_0 is:

$$\frac{\partial L}{\partial \beta_0} = \frac{1}{1 + \beta_1^2} \cdot 2 \sum_{i=1}^n -1 \cdot (y_i - \beta_1 x_i - \beta_0)$$

Setting the derivative equal to zero for minimization:

$$\begin{aligned} \sum_{i=1}^n (y_i - \beta_1 x_i - \beta_0) &= 0 \\ n\beta_0 + \beta_1 \sum_{i=1}^n x_i &= \sum_{i=1}^n y_i \\ \beta_0 &= \frac{\sum_{i=1}^n y_i - \beta_1 \sum_{i=1}^n x_i}{n} \end{aligned}$$

For β_1 , the derivative involves using the quotient rule:

$$\frac{\partial L}{\partial \beta_1} = \frac{(1 + \beta_1^2) \cdot \frac{\partial}{\partial \beta_1} (\sum_{i=1}^n (y_i - \beta_1 x_i - \beta_0)^2) - \sum_{i=1}^n (y_i - \beta_1 x_i - \beta_0)^2 \cdot 2\beta_1}{(1 + \beta_1^2)^2}$$

Expanding the derivative of the sum squared:

$$\frac{\partial}{\partial \beta_1} \left(\sum_{i=1}^n (y_i - \beta_1 x_i - \beta_0)^2 \right) = -2 \sum_{i=1}^n x_i (y_i - \beta_1 x_i - \beta_0)$$

Substituting this back into the derivative formula and simplifying:

$$\frac{\partial L}{\partial \beta_1} = \frac{-2(1 + \beta_1^2) \sum_{i=1}^n x_i (y_i - \beta_1 x_i - \beta_0) - 2\beta_1 \sum_{i=1}^n (y_i - \beta_1 x_i - \beta_0)^2}{(1 + \beta_1^2)^2}$$

Setting the derivative to zero and rearranging leads to:

$$(1 + \beta_1^2) \sum_{i=1}^n x_i (y_i - \beta_1 x_i - \beta_0) + \beta_1 \sum_{i=1}^n (y_i - \beta_1 x_i - \beta_0)^2 = 0$$

Now, substituting $\beta_0 = \frac{\sum_{i=1}^n y_i - \beta_1 \sum_{i=1}^n x_i}{n}$ into this equation, we obtain:

$$(1 + \beta_1^2) \sum_{i=1}^n x_i \left(y_i - \beta_1 x_i - \frac{\sum_{i=1}^n y_i - \beta_1 \sum_{i=1}^n x_i}{n} \right) + \beta_1 \sum_{i=1}^n \left(y_i - \beta_1 x_i - \frac{\sum_{i=1}^n y_i - \beta_1 \sum_{i=1}^n x_i}{n} \right)^2 = 0$$

This equation is now entirely in terms of β_1 , as desired. Solving for β_1 would typically require some complex methods

Problem 4

4.1

Since $P(X = x, Y = y) = P(Y = y)P(X = x|Y = y)$, we can find $P(X = x, Y = +1)$ and $P(X = x, Y = -1)$ as follows:

For $P(X = x, Y = +1)$:

$$P(X = x, Y = +1) = P(Y = +1) \cdot P(X = x|Y = +1)$$

Substituting the given values:

$$= \frac{3}{4} \cdot \frac{1}{2} \exp(-|x - 2|)$$

Simplifying, we get:

$$P(X = x, Y = +1) = \frac{3}{8} \exp(-|x - 2|)$$

For $P(X = x, Y = -1)$:

$$P(X = x, Y = -1) = P(Y = -1) \cdot P(X = x|Y = -1)$$

Substituting the given values:

$$= \frac{1}{4} \cdot \frac{1}{2} \exp(-|x + 2|)$$

Simplifying, we get:

$$P(X = x, Y = -1) = \frac{1}{8} \exp(-|x + 2|)$$

Thus, the expressions for $P(X = x, Y = y)$ are:

$$P(X = x, Y = +1) = \frac{3}{8} \exp(-|x - 2|),$$

$$P(X = x, Y = -1) = \frac{1}{8} \exp(-|x + 2|).$$

4.2

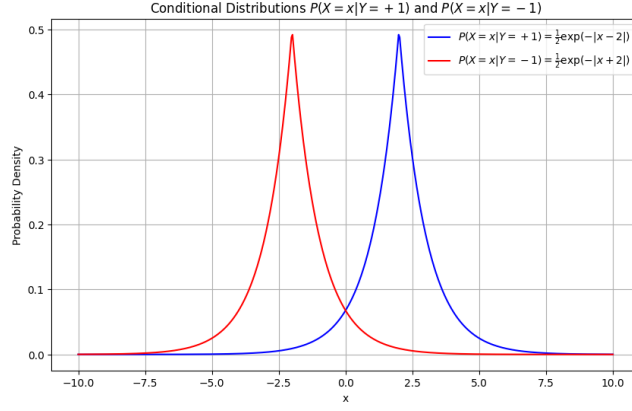


Figure 1: Conditional Distributions $P(X = x|Y = +1)$ and $P(X = x|Y = -1)$

4.3

To find the Bayes optimal classification rule, we want to assign $Y = +1$ if $P(Y = +1|X = x) > P(Y = -1|X = x)$ and $Y = -1$ otherwise.

Using Bayes' theorem, we have:

$$P(Y = +1|X = x) = \frac{P(X = x|Y = +1)P(Y = +1)}{P(X = x)}$$

$$P(Y = -1|X = x) = \frac{P(X = x|Y = -1)P(Y = -1)}{P(X = x)}$$

Since $P(X = x)$ is the same in both cases, we only need to compare $P(X = x|Y = +1)P(Y = +1)$ and $P(X = x|Y = -1)P(Y = -1)$. This leads to the decision rule:

Assign $Y = +1$ if $P(X = x|Y = +1)P(Y = +1) > P(X = x|Y = -1)P(Y = -1)$.

Given:

$$P(X = x|Y = +1) = \frac{1}{2} \exp(-|x - 2|),$$

$$P(X = x|Y = -1) = \frac{1}{2} \exp(-|x + 2|),$$

$$P(Y = +1) = \frac{3}{4}, \quad P(Y = -1) = \frac{1}{4}.$$

Our inequality becomes:

$$\frac{1}{2} \exp(-|x - 2|) \cdot \frac{3}{4} > \frac{1}{2} \exp(-|x + 2|) \cdot \frac{1}{4}.$$

We can cancel the common terms $\frac{1}{2}$ from both sides, resulting in:

$$3 \exp(-|x - 2|) > \exp(-|x + 2|).$$

Taking the natural logarithm on both sides gives:

$$\ln(3) > |x - 2| - |x + 2|.$$

Solving the Inequality

To solve $\ln(3) > |x - 2| - |x + 2|$, we need to handle the absolute values by considering different cases for x .

Case 1: $x \geq 2$ In this case:

$$|x - 2| = x - 2 \quad \text{and} \quad |x + 2| = x + 2.$$

Substituting into the inequality, we get:

$$\ln(3) > (x - 2) - (x + 2).$$

Simplifying, we have:

$$\ln(3) > -4.$$

Since $\ln(3) \approx 1.0986$, this inequality is always true for $x \geq 2$.

Case 2: $-2 \leq x < 2$ In this range:

$$|x - 2| = 2 - x \quad \text{and} \quad |x + 2| = x + 2.$$

Substituting, we get:

$$\ln(3) > (2 - x) - (x + 2).$$

Simplifying, we have:

$$\ln(3) > -2x.$$

Dividing both sides by -2 (and reversing the inequality):

$$x > \frac{-\ln(3)}{2} \approx -0.5493.$$

Thus, for this case, the inequality holds when $x > -0.5493$.

Case 3: $x < -2$ In this range:

$$|x - 2| = -x + 2 \quad \text{and} \quad |x + 2| = -x - 2.$$

Substituting, we get:

$$\ln(3) > (-x + 2) - (-x - 2).$$

Simplifying, we have:

$$\ln(3) > 4.$$

Since $\ln(3) \approx 1.0986 < 4$, this inequality is never true for $x < -2$.

Combining the Results

For $x \geq 2$, the inequality is always true. For $-0.5493 < x < 2$, the inequality holds. For $x < -0.5493$, the inequality does not hold.

The threshold value that separates the two regions is approximately $x = -0.5493$. Therefore, the Bayes optimal classification rule is:

Classify as $Y = +1$ if $x > -0.5493$, and as $Y = -1$ if $x < -0.5493$.

Verification

To confirm this threshold, let us evaluate the inequality $\ln(3) > |x-2| - |x+2|$ at sample values around the threshold.

1. For $x = -1$:

$$\begin{aligned} |x-2| &= 3, & |x+2| &= 1 \\ |x-2| - |x+2| &= 3 - 1 = 2 \end{aligned}$$

Since $\ln(3) \approx 1.0986 < 2$, the inequality does not hold for $x = -1$.

2. For $x = 0$:

$$\begin{aligned} |x-2| &= 2, & |x+2| &= 2 \\ |x-2| - |x+2| &= 2 - 2 = 0 \end{aligned}$$

Since $\ln(3) > 0$, the inequality holds for $x = 0$.

3. For $x = -0.6$:

$$\begin{aligned} |x-2| &= 2.6, & |x+2| &= 1.4 \\ |x-2| - |x+2| &= 2.6 - 1.4 = 1.2 \end{aligned}$$

Since $\ln(3) \approx 1.0986 < 1.2$, the inequality does not hold for $x = -0.6$.

4. For $x = -0.5$:

$$\begin{aligned} |x-2| &= 2.5, & |x+2| &= 1.5 \\ |x-2| - |x+2| &= 2.5 - 1.5 = 1.0 \end{aligned}$$

Since $\ln(3) > 1.0$, the inequality holds for $x = -0.5$.

These results confirm that the inequality holds for $x > -0.5493$ and does not hold for $x < -0.5493$. This verifies that the Bayes optimal classification rule is indeed:

- Classify as $Y = +1$ if $x > -0.5493$ - Classify as $Y = -1$ if $x < -0.5493$

4.4

The classification error is given by:

$$P(\text{error}) = P(X < -0.5493 \mid Y = +1) \cdot P(Y = +1) + P(X > -0.5493 \mid Y = -1) \cdot P(Y = -1)$$

Given:

$$\begin{aligned} P(X = x \mid Y = +1) &= \frac{1}{2} \exp(-|x-2|), & P(Y = +1) &= \frac{3}{4} \\ P(X = x \mid Y = -1) &= \frac{1}{2} \exp(-|x+2|), & P(Y = -1) &= \frac{1}{4} \end{aligned}$$

The CDF for a Laplace distribution centered at μ with scale parameter b is given by:

$$F(x; \mu, b) = \begin{cases} \frac{1}{2} \exp\left(\frac{x-\mu}{b}\right) & \text{if } x < \mu \\ 1 - \frac{1}{2} \exp\left(-\frac{x-\mu}{b}\right) & \text{if } x \geq \mu \end{cases}$$

For $Y = +1$ (center $\mu = 2$):

$$P(X < -0.5493 \mid Y = +1) = \frac{1}{2} \exp\left(\frac{-0.5493 - 2}{1}\right)$$

$$-0.5493 - 2 = -2.5493$$

$$P(X < -0.5493 \mid Y = +1) = \frac{1}{2} \exp(-2.5493)$$

For $Y = -1$ (center $\mu = -2$):

$$P(X > -0.5493 \mid Y = -1) = 1 - P(X \leq -0.5493 \mid Y = -1)$$

Since $-0.5493 > -2$,

$$P(X \leq -0.5493 \mid Y = -1) = 1 - \frac{1}{2} \exp\left(-\frac{-0.5493 + 2}{1}\right)$$

$$-0.5493 + 2 = 1.4507$$

$$P(X \leq -0.5493 \mid Y = -1) = 1 - \frac{1}{2} \exp(-1.4507)$$

$$P(X > -0.5493 \mid Y = -1) = \frac{1}{2} \exp(-1.4507)$$

Substituting values:

$$P(\text{error}) = \left(\frac{1}{2} \exp(-2.5493)\right) \cdot \frac{3}{4} + \left(\frac{1}{2} \exp(-1.4507)\right) \cdot \frac{1}{4}$$

Calculating:

$$\exp(-2.5493) \approx 0.0781, \quad \exp(-1.4507) \approx 0.2344$$

$$P(\text{error}) = (0.03905) \cdot 0.75 + (0.1172) \cdot 0.25$$

$$= 0.0293 + 0.0293 = 0.0586$$

Therefore, the probability of classification error is approximately:

$$P(\text{error}) \approx 0.0586 \text{ or } 5.86\%$$

4.5

Quadratic Discriminant Analysis (QDA) offers a more flexible approach compared to Linear Discriminant Analysis (LDA) by allowing different classes to have their own covariance matrices. Here are the main steps involved in training a QDA model.

- **Estimation of Parameters:**

- **Class Priors** (π_k): The prior probability of each class k is estimated as the proportion of training instances in that class. Mathematically:

$$\pi_k = \frac{N_k}{N}$$

where N_k is the number of instances in class k , and N is the total number of training instances.

- **Class Means** (μ_k): The mean vector for each class k represents the average feature values of instances in that class. It is calculated as:

$$\mu_k = \frac{1}{N_k} \sum_{i:y_i=k} x_i$$

where x_i are the feature vectors of instances that belong to class k .

- **Covariance Matrices** (Σ_k): Unlike LDA, QDA estimates a separate covariance matrix for each class k , allowing different variance and correlation structures for each class. The covariance matrix for class k is calculated as:

$$\Sigma_k = \frac{1}{N_k - 1} \sum_{i:y_i=k} (x_i - \mu_k)(x_i - \mu_k)^T$$

This matrix captures the spread and relationships between features within each class.

- **Parametric Model:** QDA assumes that each class k follows a multivariate normal (Gaussian) distribution with its own mean vector μ_k and covariance matrix Σ_k . Thus, the likelihood of a feature vector x given class k is:

$$p(x|y = k) = \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} \exp \left(-\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \right)$$

where d is the number of features, μ_k is the mean vector, and Σ_k is the covariance matrix for class k .

- **Discriminant Function:** In QDA, the decision rule is based on the log-likelihood ratio. The discriminant function for class k can be derived from

the log of the likelihood function for $p(x|y = k)$, along with the prior π_k . For a given class k , the discriminant function is:

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

This function combines the prior probability of each class and the fit of x under the class's Gaussian distribution.

- **Classification Rule:** To classify an observation x , QDA assigns it to the class k that maximizes the discriminant function $\delta_k(x)$. This is represented by:

$$\hat{y} = \arg \max_k \delta_k(x)$$

This rule selects the class k that gives the highest log-likelihood score for x , accounting for both the Gaussian distribution characteristics of each class and the class prior probabilities.

4.6

As $n \rightarrow \infty$, the parameters of the trained QDA model will converge to their true population values:

- **Class Prior Probabilities:**

$$P(Y = +1) = \frac{3}{4}, \quad P(Y = -1) = \frac{1}{4}$$

- **Class Means:**

$$\mu_{+1} = 2, \quad \mu_{-1} = -2$$

- **Class Variances:**

$$\sigma_{+1}^2 = 2, \quad \sigma_{-1}^2 = 2$$

Thus, the QDA model will accurately capture the underlying distributions for each class.

4.7

To simplify the QDA classifier using the provided discriminant function for one-dimensional data, we proceed as follows:

The QDA classifier assigns an input x to class $Y = k$ by maximizing the discriminant function:

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

where:

- μ_k is the mean of class k ,

- σ_k^2 is the variance of class k ,
- π_k is the prior probability of class k .

For this problem, we have two classes, $Y = +1$ and $Y = -1$, with:

- Means: $\mu_{+1} = 2$ and $\mu_{-1} = -2$,
- Variances: $\sigma_{+1}^2 = 2$ and $\sigma_{-1}^2 = 2$,
- Priors: $\pi_{+1} = \frac{3}{4}$ and $\pi_{-1} = \frac{1}{4}$.

We start by writing the discriminant functions for each class. For $Y = +1$:

$$\delta_{+1}(x) = -\frac{1}{2}\log(2) - \frac{(x-2)^2}{4} + \log\left(\frac{3}{4}\right)$$

For $Y = -1$:

$$\delta_{-1}(x) = -\frac{1}{2}\log(2) - \frac{(x+2)^2}{4} + \log\left(\frac{1}{4}\right)$$

The decision rule is to assign class $Y = +1$ if $\delta_{+1}(x) > \delta_{-1}(x)$. Substituting the expressions for $\delta_{+1}(x)$ and $\delta_{-1}(x)$ and simplifying gives:

$$-\frac{(x-2)^2}{4} + \log(3) > -\frac{(x+2)^2}{4}$$

We now expand the squared terms on each side. For the left side:

$$-\frac{(x-2)^2}{4} + \log(3) = -\frac{x^2 - 4x + 4}{4} + \log(3) = -\frac{x^2}{4} + x - 1 + \log(3)$$

For the right side:

$$-\frac{(x+2)^2}{4} = -\frac{x^2 + 4x + 4}{4} = -\frac{x^2}{4} - x - 1$$

Substituting back, we get:

$$-\frac{x^2}{4} + x - 1 + \log(3) > -\frac{x^2}{4} - x - 1$$

$$x - 1 + \log(3) > -x - 1$$

$$x + \log(3) > -x$$

$$2x + \log(3) > 0$$

$$x > -\frac{\log(3)}{2}$$

Thus, the simplified classification rule is: - Classify as $Y = +1$ if $x > -\frac{\log(3)}{2}$,
- Classify as $Y = -1$ if $x < -\frac{\log(3)}{2}$.

The decision boundary is at $x = -\frac{\log(3)}{2}$, which is approximately -0.5493. This threshold is derived based on maximizing the likelihood of each class given the discriminant function, resulting in a simple decision rule based on whether x is greater or less than this threshold.

- Classify as $Y = +1$ if $x > -0.5493$ - Classify as $Y = -1$ if $x < -0.5493$

4.8

In Part 7, the QDA classifier yielded a decision rule with a threshold at $x = -0.5493$, which matches the Bayes optimal decision boundary for this problem as computed in Q4.3

QDA performs optimally when we have many training data points, as the parameters estimated by QDA (class means, variances, and priors) converge to their true population values. Thus, with sufficient data, QDA approximates the Bayes optimal classifier closely.

In summary:

- **QDA performs optimally when there are many training data points**, as it captures the true class distributions accurately.
- In this problem, where the data distribution aligns with the QDA model assumptions, the QDA classifier achieves Bayes optimal performance even with finite data.

✓ Problem 3: Simple Linear Regression

In this question, you will implement simple linear regression from scratch. The dataset you will work with is called the Boston data set. You can find more information about the data set here:


<https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html>

You will use the pandas library to load the csv file into a dataframe:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

```
# read the csv file and load into a pandas dataframe
# make sure Boston.csv is in the same file path as this notebook
boston = pd.read_csv('Boston.csv')
```

```
# read the above link to learn more about what each of the columns indicate
boston.head()
```



	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	m
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	4.98	:
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	9.14	:
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	4.03	:
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	2.94	:
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	5.33	:

Simple linear regression builds a linear relationship between an input variable X and an output variable Y . We can define this linear relationship as follows:

$$Y = \beta_0 + \beta_1 X$$

- ✓ Objective: find the linear relationship between the proportion of non-retail business acres per town (indus) and the full-value property-tax rate per 10,000 dollars (tax)

So our equation will look like:

$$TAX = \beta_0 + \beta_1 INDUS$$

Here, the coefficient β_0 is the intercept, and β_1 is the scale factor or slope. How do we determine the values of these coefficients?

There are several different methods to do so, but we will focus on the Ordinary Least Squares (OLS) method. This method minimizes the sum of the squares of the differences between the observed dependent variable and those predicted by the linear function.

Recall that a residual is the difference between any data point and the line of regression. When we develop a regression model, we want the sum of the residuals squared to be minimized, indicating that the model is a close fit to the data.

$$\begin{aligned}RSS &= \sum_{i=1}^n (y_i - f(x_i))^2 \\ &= \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2\end{aligned}$$

This is the objective function we minimize to find β_0 and β_1 .

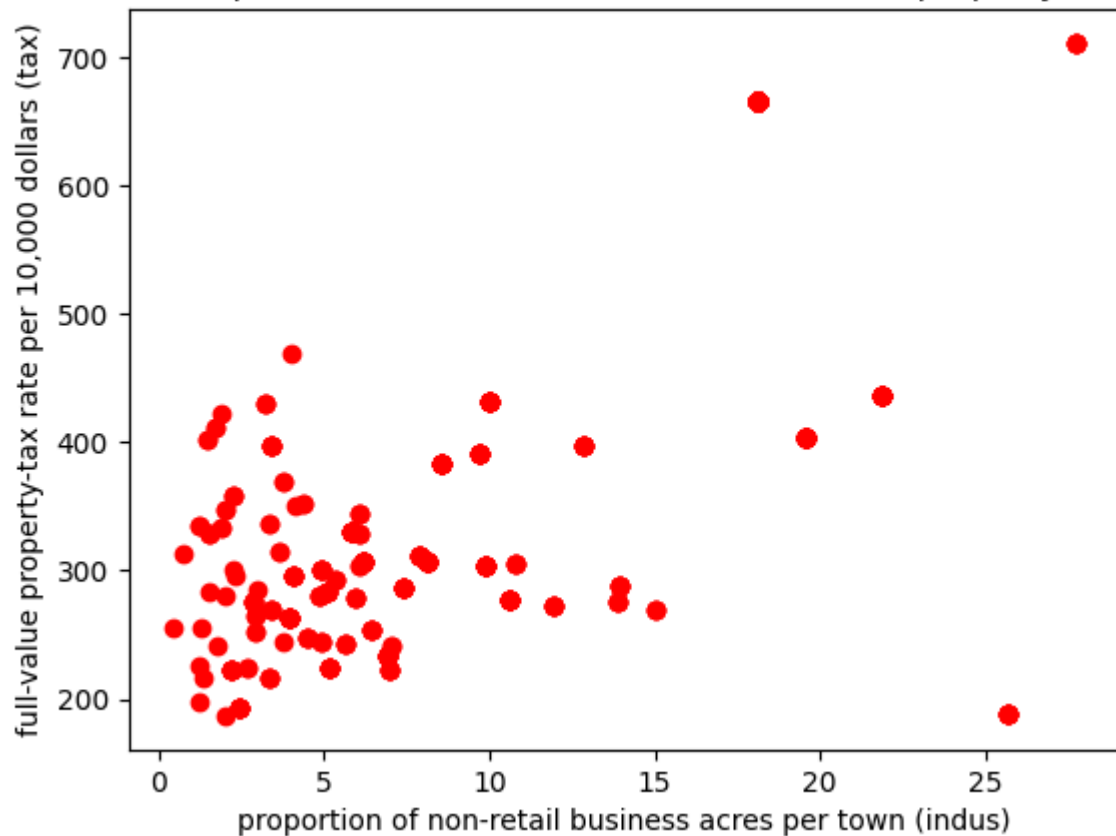
```
# set X to 'indus' and y to 'tax'
X = boston['indus']
y = boston['tax']
```

First, visualize the data by plotting X and y using matplotlib. Be sure to include a title and axis labels.

```
# TODO: display plot
plt.scatter(X,y, color='red')
# TODO: labels and title
plt.xlabel('proportion of non-retail business acres per town (indus)')
plt.ylabel('full-value property-tax rate per 10,000 dollars (tax)')
plt.title('Relationship between non-retail business acres and property tax rate')
plt.show()
```



Relationship between non-retail business acres and property tax rate



TODO: What do you notice about the relationship between the variables?

A: The plot shows no strong linear relationship. Most towns have low non-retail business acreage and a tax rate between 200-500, while towns with higher acreage show more variable tax rates.

Next, find the coefficients. The values for β_0 and β_1 are given by the following equations, where n is the total number of values. This derivation was done in class.

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$
$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

```
# TODO: implement function
def get_coeffs(X, y):
    """
    Params:
        X: the X vector
        y: the y vector
    Returns:
        a tuple (b0, b1)
    """
    x_mean = np.mean(X)
    y_mean = np.mean(y)
```



```

diff_x = X - x_mean
diff_y = y - y_mean

b1 = (np.sum(diff_x * diff_y))/np.sum(diff_x ** 2)
b0 = y_mean - (b1 * x_mean)
return b0,b1
raise NotImplementedError

```

```

# run cell to call function and display the regression line
# the values are rounded for display convenience
b0, b1 = get_coeffs(X, y)
print("Regression line: TAX = " + str(round(b0)) + " + " + str(round(b1)) + "*INDU

```

➡ Regression line: TAX = 211 + 18*INDUS

Plot the regression line overlayed on the real y-values.

```

# TODO: plot y-values
plt.scatter(X, y, color="red")

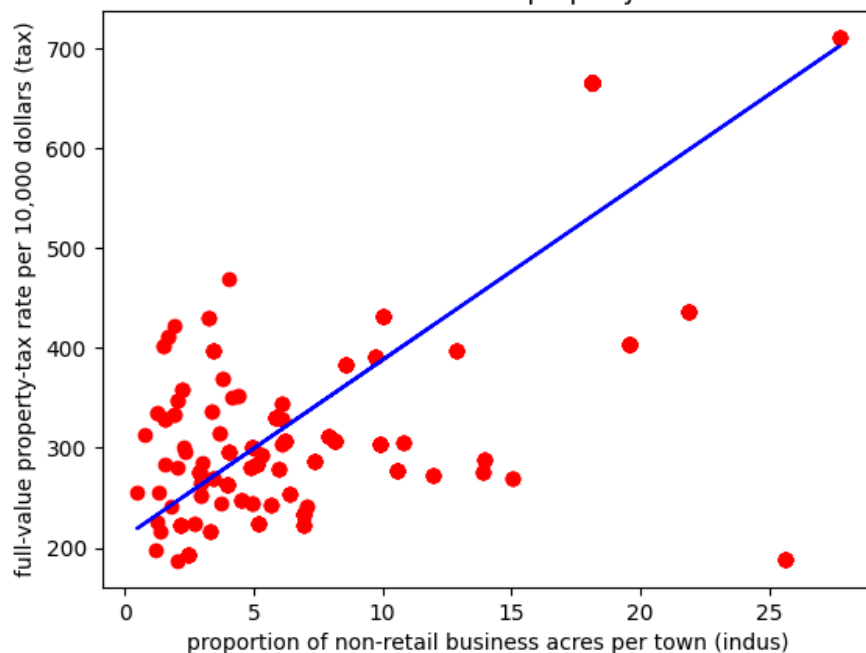
# TODO: plot regression line
pred_y = b0 + b1 * X
plt.plot(X, pred_y, color='blue')

# TODO: labels and title
plt.xlabel('proportion of non-retail business acres per town (indus)')
plt.ylabel('full-value property-tax rate per 10,000 dollars (tax)')
plt.title('Relationship between non-retail business acres and property tax rate a
plt.show()

```



Relationship between non-retail business acres and property tax rate after fitting a regression line



The line appears to fit the data, but first, let us find the RSS to evaluate this model. The RSS is used to measure the amount of variance in the data set that is not explained by the regression model. Recall that

$$RSS = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2$$

```
# TODO: implement function
def get_RSS(b0, b1, X, y):
    ...

    Params:
        b0: beta 0
        b1: beta 1
        X: X vector
        y: y vector
    Returns:
        residual sum of squares (RSS)
    ...

    diff = (y - (b0 + b1*X)) ** 2
    return np.sum(diff)
    raise NotImplementedError
```

```
# run this cell to print RSS
print("RSS:", get_RSS(b0, b1, X, y))
```



RSS: 6892554.224031559

We can also evaluate the model through the Root Mean Squared Error (RMSE) and the Coefficient of Determination (R^2 score).

- The RMSE is similar to the RSS, but provides a value with more interpretable units -- in our case, tax rate per 10,000 dollars.
- The R^2 value represents the proportion of the variance for the dependent variable that is explained by the independent variable.

Use the following equations to find the RMSE and R^2 score:

$$RMSE = \sqrt{\sum_{i=1}^n \frac{1}{n} (\hat{y}_i - y_i)^2}$$

$$R^2 = 1 - \frac{SS_r}{SS_t}$$

where

$$SS_t = \sum_{i=1}^n (y_i - \bar{y})^2$$

and

$$SS_r = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

```
# TODO: implement function
def get_RMSE(b0, b1, X, y):
    """
    Params:
        b0: beta 0
        b1: beta 1
        X: X vectore
        y: y vector
    Returns:
        rmse
    """
    pred_y = b0 + b1 * X
    diff_sum = np.sum((y - pred_y) ** 2)
    n = len(y)
    mse = diff_sum / n
    rmse = np.sqrt(mse)
    return rmse
    raise NotImplementedError

# run cell to print RMSE
print("RMSE: ", get_RMSE(b0, b1, X, y))
```

➡ RMSE: 116.71181887064391

```
# TODO: implement function
def get_R2(b0, b1, X, y):
    """
    Params:
        b0: beta 0
        b1: beta 1
        X: X vector
        y: y vector
    Returns:
        r2 score
    """
    pred_y = b0 + b1 * X
    sst = np.sum((y - np.mean(y))**2)
    ssr = np.sum((y - pred_y)**2)
    r2 = 1 - (ssr/sst)
    return r2
    raise NotImplementedError

# run cell to print RMSE
print("R2: ", get_R2(b0, b1, X, y))
```

⇒ R2: 0.5194952370037791

TODO: Analyze what the above R^2 score indicates about the model.

A: An R^2 of 0.519 indicates the model explains about 51.9% of the variance, suggesting a moderate fit. While it captures some relationship, nearly half of the variability remains unexplained,

Now, we will compare the above results with the results from using scikit-learn, a machine learning library in Python. Read the documentation (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html) to learn how to use this library. Return the R^2 score and RMSE.

```
# TODO: scikit learn function
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

def linear_regression_SKL(X, y):
    """
    Params:
        X: X vector
        y: y vector
    Returns:
        rmse and r2 as a tuple
    """
```