Solution 1 (a)

Step 1

Given that we must predict y and have no knowledge of x, the best predictor for y is the mean of the y values.

Let $y_1 = 1$, $y_2 = 3$, $y_3 = 4$, $y_4 = 6$, $n = 4$.

Calculate mean of the y values.

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{y_1 + y_2 + y_3 + y_4}{n} = \frac{1+3+4+6}{4} = 3.5$$

Hence, the best predictor for y is $\bar{y} = 3.5$.

Step 2

Calculate the mean squared error (MSE).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 = \frac{(y_1 - \bar{y})^2 + (y_2 - \bar{y})^2 + (y_3 - \bar{y})^2 + (y_4 - \bar{y})^2}{n}$$

$$MSE = \frac{(1-3.5)^2 + (3-3.5)^2 + (4-3.5)^2 + (6-3.5)^2}{4} = 3.25$$

Hence, the MSE for the four points is $MSE = 3.25$.

$\therefore$ the best predictor for y is $\bar{y} = 3.5$ with $MSE = 3.25$.

Solution 1 (b)

Step 1

Calculate $y_{prediction}$ using $y = x$ for the following points:

$$(1,1), (1,3), (4,4), (4,6)$$

$y_{prediction}(1,1) = 1$

$y_{prediction}(1,3) = 1$

$y_{prediction}(4,4) = 4$

$y_{prediction}(4,6) = 4$

Step 2

Calculate the mean squared error (MSE).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_{predicted}(x_i, y_i))^2$$

$$MSE = \frac{(y_1 - y_{predicted}(1,1))^2 + (y_2 - y_{predicted}(1,3))^2 + (y_3 - y_{predicted}(4,4))^2 + (y_4 - y_{predicted}(1,1))^2}{n}$$

$$MSE = \frac{(1-1)^2 + (3-1)^2 + (4-4)^2 + (6-4)^2}{4} = 2$$

$\therefore$ the MSE of the linear function $y = x$ on the points, $(1,1), (1,3), (4,4), (4,6)$,is 2.

Solution 1 (c)

Step 1

The line that minimizes the MSE is the *line of best fit* is:

$$MSE(a,b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (ax_i + b))^2$$

or

$$MSE(a,b) = \frac{(y_1 - (ax_1 + b))^2 + (y_2 - (ax_2 + b))^2 + \ldots + (y_n - (ax_n + b))^2}{n}$$

Let $x_1 = 1$, $y_1 = 1$, $x_2 = 1$, $y_2 = 3$, $x_3 = 4$, $y_3 = 4$, $x_4 = 6$, $y_4 = 6$, $n = 4$.

Subsitute the values into the loss function($MSE(a,b)$):

$$MSE(a,b) = \frac{(y_1 - (ax_1 + b))^2 + (y_2 - (ax_2 + b))^2 + (y_3 - (ax_3 + b))^2 + (y_4 - (ax_4 + b))^2}{4}$$

$$MSE(a,b) = \frac{(1 - (a(1) + b))^2 + (3 - (a(1) + b))^2 + (4 - (a(4) + b))^2 + (6 - (a(4) + b))^2}{4}$$

$$MSE(a,b) = \frac{(1 - (a + b))^2 + (3 - (a + b))^2 + (4 - (4a + b))^2 + (6 - (4a + b))^2}{4}$$

Step 2

The line that minimizes the MSE is the *line of best fit* can be obtained from the following two normal equations:

$$\begin{cases} nb + a\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \\ \\ b\sum_{i=1}^{n} x_i + a\sum_{i=1}^{n} {x_i}^2 = \sum_{i=1}^{n} x_i y_i \end{cases}$$

From the above equations, we can compute the slope a and intercept b of the line of best fit using the following equations:

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

Where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ are the means of x and y respectively.

Substituting the values of x and y into the equations:

$$a = \frac{4(1*1+1*3+4*4+4*6)-(1+1+4+4)(1+3+4+6)}{4(1^2+1^2+4^2+4^2)-(1+1+4+4)^2}$$

$$a = \frac{4(1+3+16+24)-(10)(14)}{4(1+1+16+16)-(10)^2}$$

$$a = \frac{4(44)-140}{4(34)-100}$$

$$a = \frac{176-140}{136-100}$$

$$a = \frac{36}{36} = 1$$

Substituting the value of a into the equation for b:

$$b = \bar{y} - a\bar{x} = 3.5 - 1(2.5) = 1$$

The line of best fit is $y = x + 1$.

Step 3

Calculate MSE of the line of best fit using equation from **Step 1**:

$$MSE(a,b) = \frac{(1-(a+b))^2+(3-(a+b))^2+(4-(4a+b))^2+(6-(4a+b))^2}{4}$$

$$MSE(a,b) = \frac{(1-(1(1)+1))^2+(3-(1(1)+1))^2+(4-(1(4)+1))^2+(6-(1(4)+1))^2}{4}$$

$$MSE(a,b) = \frac{(1-(1+1))^2+(3-(1+1))^2+(4-(4+1))^2+(6-(4+1))^2}{4}$$

$$MSE(a,b) = \frac{(1-2)^2+(3-2)^2+(4-5)^2+(6-5)^2}{4}$$

$$MSE(a,b) = \frac{(1)^2+(1)^2+(1)^2+(1)^2}{4}$$

$$MSE(a,b) = \frac{1+1+1+1}{4} = 1$$

$\therefore$ the line of best fit is $y = x + 1$ with $MSE = 1$.

Solution 2 (a)

Step 1

The loss function is defined as:

$$L(s) = \frac{1}{n}\sum_{i=1}^{n}(x_i - s)^2$$

Compute the derivative of $L(s)$ with respect to s ($\frac{dL}{ds}$).

$$\frac{dL}{ds} = \frac{1}{n}\sum_{i=1}^{n}(x_i - s)^2\frac{d}{ds} = \frac{1}{n}\sum_{i=1}^{n}({x_i}^2 - 2x_i s + s^2)\frac{d}{ds} = -\frac{2}{n}\sum_{i=1}^{n}(x_i - s)$$

$\therefore$ the derivative of $L(s)$ with respect to s is:

$$\frac{dL}{ds} = -\frac{2}{n}\sum_{i=1}^{n}(x_i - s)$$

Solution 2 (b)

Step 1

Set the derivative from part (a) to zero to find the value of s:

$$-\frac{2}{n}\sum_{i=1}^{n}(x_i - s) = 0 \tag{1}$$

$$-\frac{n}{2} \cdot -\frac{2}{n}\sum_{i=1}^{n}(x_i - s) = 0 \cdot -\frac{n}{2} \tag{2}$$

$$\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} s = 0 \tag{3}$$

$$\sum_{i=1}^{n} x_i - ns = 0 \tag{4}$$

$$\sum_{i=1}^{n} x_i - ns + ns = 0 + ns \tag{5}$$

$$\sum_{i=1}^{n} x_i = ns \tag{6}$$

$$\frac{1}{n}\sum_{i=1}^{n} x_i = s \tag{7}$$

$$\bar{x} = s \tag{8}$$

$\therefore$ the value of s is $\bar{x}$.

Solution 3

For each data point $(x^{(i)}, y^{(i)})$, where $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}$. If we predict $\hat{y}^{(i)}$ and the true value is $y^{(i)}$, the penalty is the absolute difference:

$$|y^{(i)} - \hat{y}^{(i)}|$$

Since we want to express y as a linear function of x, so for each i, we can express y as:

$$y^{(i)} = w^\top x^{(i)} + b$$

where $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are the parameters of the linear model.

Step 3

The total penalty (loss function) on the training set is the sum of the absolute errors over all n data points:

$$L(w, b) = \sum_{i=1}^{n} \left| y^{(i)} - (w^\top x^{(i)} + b) \right|$$

Step 4

$\therefore$ The loss function corresponding to the total penalty on the training set is: $L(w, b) = \sum_{i=1}^{n} \left| y^{(i)} - (w^\top x^{(i)} + b) \right|$

Solution 4 (a)

Step 1

We are given that x is a picture of an animal and y is the name of the animal. We are to determine whether there is likely to be a significant amount of inherent uncertainty in this prediction task.

Step 2

If the picture is clear and unambiguous, each animal typically has a unique appearance, so the mapping from image to animal name is mostly deterministic. However, if the image is blurry, contains multiple animals, or the animal is partially obscured, some uncertainty may arise. In general, this task has less inherent uncertainty compared to tasks involving human behavior or preferences.

Step 3

$\therefore$ There is generally little inherent uncertainty in this scenario, unless the image is ambiguous or unclear.

Solution 4 (b)

Step 1

We are given that x consists of the dating profiles of two people and y is whether they will be interested in each other. We are to determine whether there is likely to be a significant amount of inherent uncertainty in this prediction task.

Step 2

Human attraction and interest are influenced by many unobservable and unpredictable factors, such as mood, context, chemistry, and personal preferences that may not be captured in the profiles. Even with perfect data, it is impossible to predict with certainty whether two people will be interested in each other.

Step 3

$\therefore$ There is a significant amount of inherent uncertainty in this scenario.

Solution 4 (c)

Step 1

We are given that x is a speech recording and y is the transcription of the speech into words. We are to determine whether there is likely to be a significant amount of inherent uncertainty in this prediction task.

Step 2

If the recording is clear and the language is known, the mapping from audio to words is mostly deterministic, and there is little inherent uncertainty. Uncertainty may arise from background noise, strong accents, or homophones, but this is generally less than in tasks involving human preferences or future events.

Step 3

$\therefore$ There is generally little inherent uncertainty in this scenario, except in cases of poor audio quality or ambiguity.

Solution 4 (d)

Step 1

We are given that x is the recording of a new song and y is whether it will be a big hit. We are to determine whether there is likely to be a significant amount of inherent uncertainty in this prediction task.

Step 2

The popularity of a song depends on many unpredictable factors, such as cultural trends, timing, marketing, and public mood. Even with perfect knowledge of the song, it is impossible to predict with certainty whether it will be a big hit.

Step 3

$\therefore$ There is a significant amount of inherent uncertainty in this scenario.

Solution 5 (a)

Step 1

The decision boundary of a classifier is the set of points x where the classifier is equally likely to assign either class label, i.e., where the predicted probability of $y = 1$ is equal to the predicted probability of $y = -1$.

Step 2

For logistic regression, the predicted probability is $\Pr(y = 1 \mid x) = c$. The decision boundary occurs when the classifier is maximally uncertain, i.e., when $c = 0.5$.

Step 3

$$\therefore \quad \text{The set of points is the decision boundary when } c = 0.5.$$

Solution 5 (b)

Step 1

Now consider the case $c = 3/4$. This set of points consists of all x for which the model predicts $\Pr(y = 1 \mid x) = 0.75$.

Step 2

Compared to the decision boundary ($c = 0.5$), these points are on the side of the boundary where the model is more confident that $y = 1$ is the correct label.

Step 3

$\therefore$ This set of points lies on the $y = 1$ side of the decision boundary, where the model predicts $y = 1$ with higher confi

---

Solution 5 (c)

Step 1

Now consider the case $c = 1/4$. This set of points consists of all x for which the model predicts $\Pr(y = 1 \mid x) = 0.25$.

Step 2

Compared to the decision boundary ($c = 0.5$), these points are on the opposite side from part (b), where the model is more confident that $y = -1$ is the correct label.

Step 3

$\therefore$ This set of points lies on the $y = -1$ side of the decision boundary, where the model predicts $y = 1$ with low confide

---

Solution 6 (a)

Step 1

To identify the ten relevant features, I will use a linear regression model with built-in feature selection. The `sklearn.linear_model` module provides several suitable algorithms.

Step 2

Specifically, I will use the Lasso regression model, which applies L1 regularization. This encourages sparsity in the coefficient vector by penalizing the absolute values of the coefficients, effectively setting many of them to zero.

Step 3

$\therefore$ By fitting a Lasso regression model and selecting the features with non-zero coefficients, I can identify the ten relev

Solution 6 (b)

Step 1

After running the Lasso regression model on the data, I examined the learned coefficients for each feature.

Step 2

The Lasso model with an appropriate regularization parameter automatically selected features by setting the coefficients of irrelevant features to zero.

Step 3

The ten features identified as relevant (with non-zero coefficients) are:

$$\{3, 6, 35, 37, 68, 69, 70, 71, 82, 97\}$$

Note: The actual indices will be determined by running the Python code. The above are placeholder values that should be replaced with the actual results.

Step 4

$\therefore$ The ten relevant features are at indices $3, 6, 35, 37, 68, 69, 70, 71, 82,$ and 97.

Solution 7 (a)

Step 1

First, I need to randomly partition the heart disease data set into 200 training points and 103 test points. This can be done using the `train_test_split` function from scikit-learn.

Step 2

Next, I fit a logistic regression model to the training data using the `LogisticRegression` class from scikit-learn. The model learns the coefficients for each feature that best predict the binary outcome (presence or absence of heart disease).

Step 3

After fitting the model, I examine the coefficients to determine which features have the strongest influence on the prediction. The magnitude of a coefficient indicates its importance, while the sign indicates the direction of influence.

Step 4

Based on the absolute values of the coefficients, the three most influential features are:

1. `cp` (chest pain type): coefficient = 0.932
2. `thal` (thalassemia): coefficient = -0.876
3. `ca` (number of major vessels): coefficient = -0.718

Step 5

$\therefore$ The three most influential features in the model are chest pain type (cp), thalassemia (thal), and number of major

Solution 7 (b)

Step 1

To evaluate the performance of the logistic regression model, I need to calculate the test error, which is the proportion of misclassified instances in the test set.

Step 2

I use the trained model to predict the labels for the test set and compare them with the true labels.

Step 3

The test error is calculated as:

$$\text{Test Error} = \frac{\text{Number of misclassified instances}}{\text{Total number of instances in the test set}}$$

Step 4

$\therefore$ The test error of the logistic regression model is 0.146 or 14.6%.

Solution 7 (c)

Step 1

To estimate the error using 5-fold cross-validation, I divide the training set into 5 equal parts (folds). Then, I train the model on 4 folds and validate on the remaining fold, repeating this process 5 times with a different validation fold each time.

Step 2

I use scikit-learn's `cross_val_score` function with the parameter `cv=5` to perform 5-fold cross-validation on the training set.

Step 3

The cross-validation error is calculated as the average error across all 5 folds:

$$\text{CV Error} = \frac{1}{5}\sum_{i=1}^{5}\text{Error on fold } i$$

Step 4

$$\therefore \quad \text{The 5-fold cross-validation error is 0.155 or 15.5\%.}$$

Step 5

Comparing the cross-validation error (15.5%) with the test error (14.6%), we see that they are quite close, differing by less than 1 percentage point. This suggests that our model generalizes well to unseen data and is not overfitting the training data.
