Q1: Information Gain Calculation

Calculate the information gain. Given the training dataset with 8 records (4 **Low** risk and 4 **High** risk), the entropy of the parent node is:

$$E(\text{parent}) = -\sum_i p_i \log_2 p_i = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1$$

After splitting on `CreditScore` at 650, the dataset is divided into two groups:

- **Group A (CreditScore $\geq$ 650):** 5 records (4 Low, 1 High)
- **Group B (CreditScore $<$ 650):** 3 records (0 Low, 3 High)

Entropy for Group A:

$$p(\text{Low}) = \frac{4}{5}, \quad p(\text{High}) = \frac{1}{5}$$

$$E(A) = -\left(\frac{4}{5}\log_2\frac{4}{5} + \frac{1}{5}\log_2\frac{1}{5}\right)$$

Numerically, this gives:

$$E(A) \approx -\left(0.8 \times (-0.3219) + 0.2 \times (-2.3219)\right) \approx 0.722$$

Entropy for Group B: Since all records are High risk:

$$E(B) = 0$$

Weighted Entropy After the Split:

$$E_{\text{split}} = \frac{5}{8}E(A) + \frac{3}{8}E(B) = \frac{5}{8}(0.722) + \frac{3}{8}(0) \approx 0.451$$

Information Gain:

$$\text{Gain} = E(\text{parent}) - E_{\text{split}} = 1 - 0.451 \approx 0.549$$

Q2: Variance Reduction for Splitting on Age = 35

We consider the training dataset with CreditScore values:

$$\{720, 650, 750, 600, 780, 630, 710, 640\}$$

with corresponding Ages:

$$\{35, 28, 45, 31, 52, 29, 42, 33\}$$

Parent Node

The mean CreditScore is:

$$\mu = \frac{720 + 650 + 750 + 600 + 780 + 630 + 710 + 640}{8} = 685$$

The variance is computed as:

$$\sigma^2_{\text{parent}} = \frac{1}{8}\sum_{i=1}^{8}(x_i - 685)^2 = \frac{28600}{8} = 3575$$

Splitting on Age = 35

Group A (Age $\leq$ 35): Records: {1, 2, 4, 6, 8} with CreditScores:

$$\{720, 650, 600, 630, 640\}$$

The mean for Group A is:

$$\mu_A = \frac{720 + 650 + 600 + 630 + 640}{5} = 648$$

The variance for Group A is:

$$\sigma_A^2 = \frac{1}{5}\Big[(720-648)^2 + (650-648)^2 + (600-648)^2 + (630-648)^2 + (640-648)^2\Big] \approx 1576$$

Group B (Age $>$ 35): Records: {3, 5, 7} with CreditScores:

$$\{750, 780, 710\}$$

The mean for Group B is:

$$\mu_B = \frac{750 + 780 + 710}{3} \approx 746.67$$

The variance for Group B is:

$$\sigma_B^2 = \frac{1}{3}\Big[(750-746.67)^2 + (780-746.67)^2 + (710-746.67)^2\Big] \approx 822.22$$

Weighted Variance After Split

The weighted variance after the split is:

$$\sigma^2_{\text{split}} = \frac{5}{8}\sigma_A^2 + \frac{3}{8}\sigma_B^2 \approx \frac{5}{8}(1576) + \frac{3}{8}(822.22) \approx 1293.33$$

Variance Reduction

The variance reduction achieved by the split is:

$$\text{Reduction} = \sigma^2_{\text{parent}} - \sigma^2_{\text{split}} \approx 3575 - 1293.33 \approx 2281.67$$

Comparison with Information Gain in Classification

While **variance reduction** minimizes the mean squared error for continuous targets, **information gain** in classification trees measures the reduction in impurity (entropy) for categorical outcomes. Variance reduction focuses on reducing numerical dispersion, whereas information gain focuses on achieving purer class splits.

Q3: Predicting T2's Risk Level and Handling Missing Values

Part 1: Probability of T2 Being High Risk

Consider the training dataset with 8 records:

ID	Age	CreditScore	RiskLevel
1	35	720	Low
2	28	650	High
3	45	750	Low
4	31	600	High
5	52	780	Low
6	29	630	High
7	42	710	Low
8	33	640	High

The test record T2 has:

$$\text{Age} = 30, \quad \text{CreditScore} = 645, \quad \text{Education} = \text{missing}.$$

Since Education is missing, we rely on Age and CreditScore. We define similarity as:

$$\text{Absolute difference in Age} \leq 5 \quad \text{and} \quad \text{Absolute difference in CreditScore} \leq 25.$$

Based on this criterion, the similar training records are:

ID	Age	CreditScore	RiskLevel
2	28	650	High
6	29	630	High
8	33	640	High

Let:

$$n = \text{number of similar records} = 3, \quad n_H = \text{number of similar records with High Risk} = 3.$$

Then, the probability that T2 is High Risk is:

$$P(\text{High Risk} \mid \text{similar records}) = \frac{n_H}{n} = \frac{3}{3} = 1.$$

Part 2: Handling Missing Education Values

For future cases where the Education value is missing, several imputation methods can be used:

- **K-Nearest Neighbors (KNN) Imputation:** Estimate the missing Education value by averaging (or taking the mode, if categorical) the Education values of the most similar records, where similarity is determined using Age and CreditScore.
- **Regression Imputation:** Build a regression model that predicts Education based on other features (e.g., Age, CreditScore, and even RiskLevel) and use it to impute missing values.
- **Mean/Median Imputation:** Replace missing values with the mean or median Education value calculated from similar records or the entire dataset if appropriate.
- **Missingness Indicator:** Create an additional binary variable that indicates whether the Education value is missing. This allows the model to capture any information contained in the missingness itself.

In this example, because all similar records (based on Age and CreditScore) are classified as High Risk, T2 is predicted to be High Risk with probability 1, even though the Education value is missing.

Q4: Batch Gradient Descent for Predicting CreditScore Using Age

We use the hypothesis:

$$h(x) = \theta_0 + \theta_1 x,$$

with the training dataset:

Age (x)	CreditScore (y)
35	720
28	650
45	750
31	600
52	780
29	630
42	710
33	640

The initial parameters and learning rate are given by:

$$\theta_0 = 500, \quad \theta_1 = 5, \quad \alpha = 0.01,$$

and the number of records $m = 8$.

Step 1: Compute the Initial Cost

The cost function (Mean Squared Error) is defined as:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h(x_i) - y_i\right)^2 .$$

For each training example, the prediction is:

$$h(x_i) = 500 + 5x_i.$$

The errors and squared errors are computed as follows:

Record	x_i	$h(x_i)$	y_i	$e_i = h(x_i) - y_i$	e_i^2
1	35	$500 + 175 = 675$	720	$675 - 720 = -45$	2025
2	28	$500 + 140 = 640$	650	$640 - 650 = -10$	100
3	45	$500 + 225 = 725$	750	$725 - 750 = -25$	625
4	31	$500 + 155 = 655$	600	$655 - 600 = 55$	3025
5	52	$500 + 260 = 760$	780	$760 - 780 = -20$	400
6	29	$500 + 145 = 645$	630	$645 - 630 = 15$	225
7	42	$500 + 210 = 710$	710	$710 - 710 = 0$	0
8	33	$500 + 165 = 665$	640	$665 - 640 = 25$	625

Summing up the squared errors:

$$\sum_{i=1}^{8} e_i^2 = 2025 + 100 + 625 + 3025 + 400 + 225 + 0 + 625 = 7025.$$

Thus, the initial cost is:

$$J(\theta_0, \theta_1) = \frac{7025}{2 \cdot 8} = \frac{7025}{16} \approx 439.06.$$

Step 2: Compute the Gradients

The gradients are given by:

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^{m} (h(x_i) - y_i),$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{m}\sum_{i=1}^{m}(h(x_i) - y_i)x_i.$$

Calculations for θ_0:
The errors are: $-45, -10, -25, 55, -20, 15, 0, 25$. Their sum:

$$\sum_{i=1}^{8}(h(x_i) - y_i) = -45 - 10 - 25 + 55 - 20 + 15 + 0 + 25 = -5.$$

Thus,

$$\frac{\partial J}{\partial \theta_0} = \frac{-5}{8} \approx -0.625.$$

Calculations for θ_1:
Multiply each error by the corresponding Age:

$$\begin{aligned}
-45 \times 35 &= -1575, \\
-10 \times 28 &= -280, \\
-25 \times 45 &= -1125, \\
55 \times 31 &= 1705, \\
-20 \times 52 &= -1040, \\
15 \times 29 &= 435, \\
0 \times 42 &= 0, \\
25 \times 33 &= 825.
\end{aligned}$$

Summing these values:

$$-1575 - 280 - 1125 + 1705 - 1040 + 435 + 0 + 825 = -1055.$$

Then,

$$\frac{\partial J}{\partial \theta_1} = \frac{-1055}{8} \approx -131.875.$$

Step 3: Update the Parameters

Using the gradient descent update rule:

$$\theta_j := \theta_j - \alpha \frac{\partial J}{\partial \theta_j},$$

we update each parameter:

$$\theta_0^{\text{new}} = 500 - 0.01 \times (-0.625) = 500 + 0.00625 = 500.00625,$$
$$\theta_1^{\text{new}} = 5 - 0.01 \times (-131.875) = 5 + 1.31875 = 6.31875.$$

Interpretation

The update increases θ_1 significantly while leaving θ_0 almost unchanged. This indicates that the model requires a larger weight on the feature Age to better approximate the CreditScore. The negative gradients imply that our initial hypothesis predictions were generally lower than the actual values; hence, both parameters are adjusted upward to improve predictions.

Q5: Multiple Linear Regression for Predicting CreditScore using Age and Education

We consider a multiple linear regression model of the form

$$\hat{y} = \beta_0 + \beta_1 \cdot \text{Age} + \beta_2 \cdot \text{Education},$$

where $\hat{y}$ is the predicted CreditScore.

Training Data: Assume we have the following data after handling the missing Education value (for example, by imputing with the mean of the available values):

ID	Age	Education	CreditScore
1	35	16	720
2	28	14	650
3	45	14.57	750
4	31	12	600
5	52	18	780
6	29	14	630
7	42	16	710
8	33	12	640

We form the design matrix $\mathbf{X}$ by including an intercept term:

$$\mathbf{X} = \begin{bmatrix} 1 & 35 & 16 \\ 1 & 28 & 14 \\ 1 & 45 & 14.57 \\ 1 & 31 & 12 \\ 1 & 52 & 18 \\ 1 & 29 & 14 \\ 1 & 42 & 16 \\ 1 & 33 & 12 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 720 \\ 650 \\ 750 \\ 600 \\ 780 \\ 630 \\ 710 \\ 640 \end{bmatrix}.$$

The normal equation provides the solution for the coefficient vector $\boldsymbol{\beta}$ as:

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

Interpretation of Coefficients:

- β_0 (Intercept): The predicted CreditScore when both Age and Education are zero. Although extrapolating to Age $= 0$ or Education $= 0$ is not meaningful in practice, β_0 serves as a baseline level for the model.
- β_1 (Coefficient for Age): Represents the expected change in CreditScore associated with a one-year increase in Age, holding Education constant.
- β_2 (Coefficient for Education): Represents the expected change in CreditScore for each additional unit (e.g., year) of Education, holding Age constant.

In summary, after solving the normal equation, the fitted model

$$\hat{y} = \beta_0 + \beta_1 \cdot \text{Age} + \beta_2 \cdot \text{Education}$$

provides predictions for CreditScore. The coefficients indicate how sensitive the prediction is to changes in Age and Education.

Q6: Mean Squared Error and $\mathbf{R^2}$ Calculation for the Linear Regression Model

Consider the multiple linear regression model obtained in Q5:

$$\hat{y} = \beta_0 + \beta_1 \cdot \text{Age} + \beta_2 \cdot \text{Education},$$

where the model coefficients $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2]^T$ are computed using the normal equation:

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

Assume that we have a training dataset with m records:

$$\{(\text{Age}_i, \text{Education}_i, y_i) : i = 1, 2, \ldots, m\},$$

where y_i denotes the actual CreditScore for record i.

Mean Squared Error (MSE)

The mean squared error is given by:

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}\big(y_i - \hat{y}_i\big)^2,$$

where

$$\hat{y}_i = \beta_0 + \beta_1 \cdot \text{Age}_i + \beta_2 \cdot \text{Education}_i$$

is the predicted CreditScore for record i.

$\mathbf{R^2}$ Value

Let the mean of the actual CreditScore values be:

$$\bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i.$$

Then, we define:

- The total sum of squares (SST):

$$\text{SST} = \sum_{i=1}^{m}(y_i - \bar{y})^2,$$

- The residual sum of squares (SSR):

$$\text{SSR} = \sum_{i=1}^{m}(y_i - \hat{y}_i)^2.$$

The R^2 value is then calculated as:

$$R^2 = 1 - \frac{\text{SSR}}{\text{SST}}.$$

Assessment

A high R^2 (close to 1) indicates that the model explains most of the variance in the CreditScore values, meaning that linear regression is appropriate for this relationship. A low R^2 value suggests that the linear model might not be a good fit for the data.

Q7: Logistic Regression for Predicting RiskLevel

We model the probability that a record belongs to the positive class (e.g., RiskLevel = 1) using the logistic regression hypothesis:

$$h(\mathbf{x}) = g(z) = \frac{1}{1+e^{-z}}, \quad \text{with} \quad z = w_0 + w_1 x_1 + w_2 x_2,$$

where $x_1 =$ Age and $x_2 =$ CreditScore.

The given weights are:

$$w_0 = 0.5, \quad w_1 = -0.02, \quad w_2 = 0.01.$$

For test record T1, we have:

$$\text{Age} = 37, \quad \text{CreditScore} = 705.$$

Step 1: Calculate the Linear Combination z

$$z = 0.5 + (-0.02)(37) + 0.01(705).$$

Compute each term:

$$-0.02 \times 37 = -0.74, \quad 0.01 \times 705 = 7.05.$$

Thus,

$$z = 0.5 - 0.74 + 7.05 = 6.81.$$

Step 2: Compute the Logistic (Sigmoid) Prediction

$$h(\mathbf{x}) = \frac{1}{1+e^{-6.81}}.$$

Since $e^{-6.81}$ is very small, we have:

$$h(\mathbf{x}) \approx 0.9989.$$

Thus, the model predicts that T1 has approximately a 99.89% chance for the positive class.

Step 3: Compute the Cost Function for T1

The logistic regression cost function for a single example is:

$$J(w) = -\left[y \log\left(h(\mathbf{x})\right) + (1-y)\log\left(1 - h(\mathbf{x})\right) \right],$$

where y is the true label. Assume that for T1 the true label is $y = 1$ (i.e. the record belongs to the positive class).

Then the cost is:

$$J(w) = -\left[1 \cdot \log(0.9989) + (1-1) \cdot \log\left(1 - 0.9989\right)\right] = -\log(0.9989).$$

Numerically,

$$-\log(0.9989) \approx 0.0011.$$

Summary: For test record T1, the linear combination is $z = 6.81$, the logistic prediction is $h(\mathbf{x}) \approx 0.9989$, and assuming the true label $y = 1$, the cost function value is approximately 0.0011.

Q8: Gradient Vector Calculation in Logistic Regression and the Effect of Regularization

Recall the logistic regression hypothesis:

$$h(\mathbf{x}) = g(z) = \frac{1}{1+e^{-z}}, \quad \text{where} \quad z = w_0 + w_1 x_1 + w_2 x_2.$$

For test record T1 we have:

$$x_1 = \text{Age} = 37, \quad x_2 = \text{CreditScore} = 705,$$

with weights:

$$w_0 = 0.5, \quad w_1 = -0.02, \quad w_2 = 0.01.$$

Step 1: Compute the Prediction and Error

The linear combination is computed as:

$$z = 0.5 + (-0.02)(37) + (0.01)(705).$$

Calculating each term:

$$-0.02 \times 37 = -0.74, \quad 0.01 \times 705 = 7.05.$$

Thus,

$$z = 0.5 - 0.74 + 7.05 = 6.81.$$

The logistic prediction is:

$$h(\mathbf{x}) = \frac{1}{1+e^{-6.81}} \approx 0.9989.$$

Assuming for T1 the true label is $y = 1$, the error is:

$$\text{error} = h(\mathbf{x}) - y = 0.9989 - 1 \approx -0.0011.$$

Step 2: Compute the Unregularized Gradient Vector

For logistic regression, the gradient for each weight is given by:

$$\frac{\partial J}{\partial w_j} = \big(h(\mathbf{x}) - y\big) x_j,$$

where $x_0 = 1$ (for the intercept) and x_1, x_2 are the input features. Therefore:

$$\begin{aligned}
\frac{\partial J}{\partial w_0} &= (0.9989 - 1) \times 1 \approx -0.0011, \\
\frac{\partial J}{\partial w_1} &= (0.9989 - 1) \times 37 \approx -0.0011 \times 37 \approx -0.0407, \\
\frac{\partial J}{\partial w_2} &= (0.9989 - 1) \times 705 \approx -0.0011 \times 705 \approx -0.7755.
\end{aligned}$$

Thus, the unregularized gradient vector is approximately:

$$\nabla J = \begin{bmatrix} -0.0011 \\ -0.0407 \\ -0.7755 \end{bmatrix}.$$

Step 3: Effect of Regularization

In many practical cases, regularization (often using L2 regularization) is added to prevent overfitting. Under L2 regularization the cost function becomes:

$$J_{\text{reg}}(w) = J(w) + \frac{\lambda}{2}\sum_{j=1}^{n} w_j^2,$$

where λ is the regularization parameter and the sum is taken over all weights except for the intercept w_0.

The gradients are modified as follows:

$$\frac{\partial J_{\text{reg}}}{\partial w_j} = \begin{cases} (h(\mathbf{x}) - y)x_j, & \text{for } j = 0, \\ (h(\mathbf{x}) - y)x_j + \lambda w_j, & \text{for } j \geq 1. \end{cases}$$

Thus, for w_1 and w_2, the gradients become:

$$\begin{aligned} \frac{\partial J_{\text{reg}}}{\partial w_1} &= (h(\mathbf{x}) - y)x_1 + \lambda w_1 \approx -0.0407 + \lambda(-0.02), \\ \frac{\partial J_{\text{reg}}}{\partial w_2} &= (h(\mathbf{x}) - y)x_2 + \lambda w_2 \approx -0.7755 + \lambda(0.01). \end{aligned}$$

Why Regularization is Necessary

- **Preventing Overfitting:** Regularization discourages the model from learning overly complex or extreme weights by penalizing large values. This leads to a model that generalizes better to unseen data.
- **Handling High-Dimensionality/Multicollinearity:** In datasets with many features or correlated features, regularization can help stabilize the solution.
- **Improved Convergence:** Regularization can improve the convergence behavior of gradient descent by smoothing out the optimization landscape.

Summary: For test record T1, the unregularized gradient vector is

$$\nabla J \approx \begin{bmatrix} -0.0011 \\ -0.0407 \\ -0.7755 \end{bmatrix}.$$

With L2 regularization (with parameter λ), an additional term λw_j is added to the gradients for w_1 and w_2. This modification helps control the magnitude of the coefficients, reducing the risk of overfitting.