VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Machine Learning and Applications (CO5241)

Answer Sheet

A Practical Problem

Advisor(s):	NGUYỄN AN KHƯƠNG	
Student(s):	Krebs Luca Felix	2470475

HO CHI MINH CITY, MARCH 2025

Contents

List of Figures

List of Tables

Listings

1 Question 1

Calculate the information gain for splitting ***`CreditScore`*** *at 650 in a decision tree classification task, then explain why you would or would not choose this as the root node split.*

1.1 Step 1: Extract Relevant Data

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

Target attribute: `RiskLevel` (Low or High)
Split feature: `CreditScore` at 650

1.2 Step 2: Compute Total Entropy $H(S)$

We have:

- 4 samples labeled "Low"
- 4 samples labeled "High"

$$H(S) = -\left(\frac{4}{8}\log_2\frac{4}{8} + \frac{4}{8}\log_2\frac{4}{8}\right) = -\left(0.5\log_2 0.5 + 0.5\log_2 0.5\right) = 1.0$$

1.3 Step 3: Split at `CreditScore` ≤ 650

Left group (S_1):

$$\text{IDs: } 2, 4, 6, 8 \Rightarrow \texttt{CreditScore} \leq 650$$

`RiskLevel: High, High, High, High` ⇒ All "High"

$$H(S_1) = -1 \cdot \log_2(1) = 0$$

Right group (S_2):

$$\text{IDs: 1, 3, 5, 7} \Rightarrow \texttt{CreditScore} > 650$$

`RiskLevel: Low, Low, Low, Low` ⇒ All "Low"

$$H(S_2) = -1 \cdot \log_2(1) = 0$$

1.4 Step 4: Weighted Average Entropy After Split

$$H_{\text{after split}} = \frac{4}{8} \cdot H(S_1) + \frac{4}{8} \cdot H(S_2) = 0.5 \cdot 0 + 0.5 \cdot 0 = 0$$

1.5 Step 5: Information Gain

$$IG = H(S) - H_{\text{after split}} = 1.0 - 0 = \boxed{1.0}$$

1.6 Step 6: Interpretation

The calculated information gain for splitting the dataset at `CreditScore = 650` is 1.0. This value is the highest possible, indicating that the split completely separates the two classes, "Low" and "High" risk.

After the split:

- All individuals with a credit score less than or equal to 650 belong to the "High" risk group.
- All individuals with a credit score greater than 650 belong to the "Low" risk group.

This result shows that the feature `CreditScore`, when split at 650, perfectly distinguishes the risk levels in the training dataset. As a result, there is no uncertainty remaining in either group after the split.

Conclusion: Splitting on `CreditScore = 650` provides a highly informative division of the data. Therefore, it is a strong candidate to be used as the root node in the decision tree model.

2 Question 2

For a regression decision tree predicting `CreditScore`, calculate the variance reduction when splitting on `Age = 35`, and describe how this splitting criterion differs from information gain.

2.1 Step 1: Prepare Data

We use the training dataset from Question 1, removing any rows with missing `CreditScore`. All 8 records are complete.

2.2 Step 2: Split at Age = 35

- Left group (Age $\leq$ 35): IDs 1, 2, 4, 6, 8
- Right group (Age $>$ 35): IDs 3, 5, 7

2.3 Step 3: Calculate Variance

Let $\bar{x}$ be the mean of the credit scores.

- Total variance (before split):

$$\text{Var}_{\text{total}} = 3575.00$$

- Left group variance:

$$\text{Var}_{\text{left}} = 1576.00 \quad \text{(Group size: 5)}$$

- Right group variance:

$$\text{Var}_{\text{right}} = 822.22 \quad \text{(Group size: 3)}$$

2.4 Step 4: Compute Weighted Average Variance After Split

$$\text{Var}_{\text{after split}} = \frac{5}{8} \cdot 1576.00 + \frac{3}{8} \cdot 822.22 = 1293.33$$

2.5 Step 5: Variance Reduction

$$\text{Reduction} = 3575.00 - 1293.33 = \boxed{2281.67}$$

2.6 Step 6: Interpretation

The variance reduction of 2281.67 is quite substantial. This indicates that splitting the dataset at `Age = 35` helps reduce the spread in `CreditScore` values and could therefore improve the accuracy of the regression tree.

Difference from Classification Trees:
Classification trees use entropy and information gain to measure uncertainty in categorical outcomes. In contrast, regression trees work with numerical targets and aim to minimise variance, which reflects prediction error.

3 Question 3

Using both ***`CreditScore`*** *and* ***`Age`*** *patterns in the training data, determine the probability of T2 being High Risk given its missing* ***`Education`*** *value. Then propose a method to handle similar missing values in future cases.*

3.1 Step 1: Identify T2 Features

T2 has the following:

- Age = 30
- CreditScore = 645
- Education = missing

3.2 Step 2: Compare with Similar Training Samples

We define "similar" as:

- Age difference ≤ 3 years
- CreditScore difference ≤ 20 points

Matching training samples:

- ID 2: Age 28, CreditScore 650 – High Risk
- ID 6: Age 29, CreditScore 630 – High Risk
- ID 8: Age 33, CreditScore 640 – High Risk

3.3 Step 3: Estimate Risk Probabilities

Among the 3 similar training samples:

- High Risk: $3/3 = 100\%$
- Low Risk: $0/3 = 0\%$

$$P(\text{High Risk} \mid \text{Age} = 30, \text{CreditScore} = 645) = \boxed{1.00}$$

3.4 Step 4: Handling Missing Values

When a feature such as `Education` is missing, we can apply different strategies:

- **Mean or Median Imputation:** Replace with the average value from the dataset.
- **Similarity-Based Estimation:** Use nearby records based on available features only.
- **Predictive Models:** Train a model on complete data to estimate the missing value.
- **Omit Missing Attributes:** If the known features are strong predictors, proceed without imputation.

Conclusion: Based on the observed pattern, T2 has a high likelihood (1.00) of being High Risk. Using similarity-based probability estimation is a practical and interpretable solution in the presence of missing values.

4 Question 4

Implement batch gradient descent to find the optimal weights for predicting ***`CreditScore`*** *using* ***`Age`*** *as input. Starting with $\theta_0 = 500$, $\theta_1 = 5$, compute the cost function and one iteration of gradient descent updates using learning rate $\alpha = 0.01$. Interpret the direction of the parameter updates.*

4.1 Step 1: Model and Parameters

We use the linear regression model:

$$\hat{y} = \theta_0 + \theta_1 x$$

Initial values:

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

4.2 Step 2: Training Data

We use all 8 records with valid values for `Age` and `CreditScore`.

4.3 Step 3: Cost Function

Predicted values:

$$\hat{y}_i = 500 + 5 \cdot \texttt{Age}_i$$

Mean Squared Error:

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 = 878.12$$

4.4 Step 4: Gradient Computation

$$\frac{\partial J}{\partial \theta_0} = -1.25, \quad \frac{\partial J}{\partial \theta_1} = -263.75$$

4.5 Step 5: Parameter Updates

$$\theta_0 := 500 - 0.01 \cdot (-1.25) = \boxed{500.01}$$

$$\theta_1 := 5 - 0.01 \cdot (-263.75) = \boxed{7.64}$$

4.6 Step 6: Interpretation

Both gradients are negative, which means the current model underestimates the actual `CreditScore`. The algorithm increases both parameters. In the next iterations, this trend will continue until the model converges to values that minimise the error.