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)

---

### Cheat 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: Information Gain for Decision Trees

**Goal:** Determine whether splitting the `CreditScore` feature at 650 is a good root node in a decision tree.

## 1.1 Key Terminology

- **Entropy ($H$)**: A measure of impurity or disorder in a set.

$$H(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

  where $p_i$ is the proportion of class $i$ in dataset $S$, and $c$ is the number of classes.[2]

- **Information Gain (IG)**: Reduction in entropy after splitting a dataset.

$$IG(S, A) = H(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} H(S_v)$$

  where $A$ is an attribute and $S_v$ is the subset of $S$ where $A = v$ (or meets some condition, e.g., $\leq 650$).[3]

## 1.2 Steps to Compute Information Gain

1. Compute the total entropy of the training set $S$ using `RiskLevel` labels.
2. Split the dataset at `CreditScore` $= 650$ into:
   - $S_1$: `CreditScore` $\leq 650$
   - $S_2$: `CreditScore` $> 650$
3. Compute the entropy for $S_1$ and $S_2$.
4. Compute the weighted average entropy after the split:

$$H_{\text{after split}} = \frac{|S_1|}{|S|} H(S_1) + \frac{|S_2|}{|S|} H(S_2)$$

[7]

5. Calculate the information gain:

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

### 1.3 Interpretation

- A high information gain means the feature and threshold effectively reduce uncertainty.
- If $IG$ is low, choose a different feature or split value.[7]

### 1.4 Example Values (from Dataset)

- $H(S) = 1.0$
- $H(S_1) = 1.0$, $H(S_2) = 0.0$
- $H_{\text{after split}} = 0.5$
- $IG = 0.5$

**Conclusion:** Splitting on `CreditScore = 650` gives an information gain of 0.5, which significantly reduces entropy. It is a reasonable candidate for the root node split.

## 2 Question 2: Variance Reduction in Regression Trees

### 2.1 Goal

To determine whether splitting the dataset at `Age = 35` reduces the prediction error for `CreditScore` in a regression decision tree.

### 2.2 Key Concepts

- **Variance:** Measures how spread out numeric values are. A lower variance indicates that values are close to the mean.

$$\text{Var}(S) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

[6]

- **Variance Reduction:** Measures how much the variance decreases after a split.

$$\text{Reduction} = \text{Var}(S) - \left( \frac{|S_1|}{|S|} \cdot \text{Var}(S_1) + \frac{|S_2|}{|S|} \cdot \text{Var}(S_2) \right)$$

[6]

### 2.3 Example Calculation (from Dataset)

- Total variance before split: 3575.00
- Variance for group Age $\leq$ 35: 1576.00
- Variance for group Age $>$ 35: 822.22
- Weighted average variance after split: 1293.33
- Variance Reduction: 2281.67

### 2.4 Interpretation

A large variance reduction means the split improves the prediction of the numeric target. In this case, splitting on `Age = 35` substantially reduces the spread in `CreditScore`, making it a good candidate for a decision node in a regression tree.

**Note:** Regression trees aim to reduce numerical error, unlike classification trees which use entropy and information gain to split categorical outcomes. [5]

## 3 Question 3: Probability Estimation with Missing Values

### 3.1 Goal

To estimate the probability of the test sample T2 being "High Risk" using patterns in `Age` and `CreditScore`, even though the `Education` value is missing.

### 3.2 Key Concepts

- **Conditional Probability:** Estimate the likelihood of an outcome given observed features.

- **Similarity-Based Reasoning:** Use training records that are close to the test sample to make probabilistic predictions.

## 3.3 Steps

1. Identify the test sample with missing data (T2).
2. Compare only the available features (Age and CreditScore).
3. Select training records that are similar (within ±3 years in Age and ±20 points in CreditScore).
4. Count the frequency of each risk label in those similar samples.
5. Use these frequencies to estimate the probability.

## 3.4 Example Result (T2)

- Number of similar training samples: 3
- Probability of High Risk: 1.00
- Probability of Low Risk: 0.00

## 3.5 Handling Missing Values

- **Mean/Median Imputation:** Replace missing values with the average or most common value.
- **Similarity-Based Methods:** Use nearby samples in terms of known features.
- **Model-Based Prediction:** Use predictive models trained on complete data.
- **Ignore Missing Features:** Proceed using only the available attributes when feasible.

# 4 Question 4: Batch Gradient Descent (Linear Regression)

## 4.1 Goal

To perform one batch gradient descent step to optimise weights for predicting `CreditScore` using `Age` as input.

## 4.2 Key Concepts

- **Linear Regression Hypothesis:**

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

- **Mean Squared Error (MSE):**

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

[4]

- **Gradient Descent Updates:**

$$\theta_0 := \theta_0 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(\hat{y}^{(i)} - y^{(i)})$$

$$\theta_1 := \theta_1 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(\hat{y}^{(i)} - y^{(i)}) \cdot x^{(i)}$$

[1]

## 4.3 Example Output (from Dataset)

- Initial Cost: 878.12
- Gradient $\theta_0$: -1.25
- Gradient $\theta_1$: -263.75
- Updated $\theta_0$: 500.01

- Updated $\theta_1$: 7.64

### 4.4 Interpretation

Both gradients are negative, indicating that the initial predictions underestimate the true values. Therefore, the model increases both parameters to better fit the data. Repeating this process iteratively minimises the cost.

## References

[1] geeksforgeeks. Gradient descent in linear regression, 2024.

[2] geeksforgeeks. How to calculate entropy in decision tree?, 2024.

[3] geeksforgeeks. How to calculate information gain in decision tree?, 2024.

[4] geeksforgeeks. Mean squared error, 2024.

[5] Kushal Vala. How to get started with regression trees, 2025.

[6] Wikipedia. Decision tree learning, 2025.

[7] Victor Zhou. A simple explanation of information gain and entropy, 2022.