

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC XÃ HỘI VÀ NHÂN VĂN

NGHIÊM THỊ NGỌC THẢO 2256210057

**CÁC THUẬT TOÁN PHÂN LỚP DỰA TRÊN CÂY QUYẾT
ĐỊNH ỨNG DỤNG TRONG BÀI TOÁN TUYỂN DỤNG NHÂN SỰ**

TIỂU LUẬN ĐỒ ÁN LƯU TRỮ VÀ KHAI THÁC DỮ LIỆU

TP. Hồ Chí Minh, 2024

MỤC LỤC

MỤC LỤC	i
BẢNG MÔ TẢ CÁC THUẬT NGỮ	ii
DANH MỤC CÁC HÌNH	iii
DANH MỤC BẢNG	v
TÓM TẮT TIỂU LUẬN ĐỒ ÁN	vi
Chương 1 Khái quát về cây quyết định.....	1
1.1. Phân lớp dữ liệu dựa vào cây quyết định.....	1
1.2. Quá trình xây dựng cây quyết định.....	3
1.3. Các thuật toán cây quyết định.....	3
Chương 2 Một số thuật toán cây quyết định	5
2.1. Thuật toán ID3	5
2.2. Thuật toán C4.5	10
2.3. Thuật toán CART	12
Chương 3 Kết quả thực nghiệm.....	17
3.1. Đặt vấn đề	17
3.2. Thu thập dữ liệu	17
3.3. Tiền xử lý dữ liệu.....	18
3.4. Xây dựng cây quyết định.....	20
3.4.1. Phần mềm Weka	20
3.4.2. Python	23
DANH MỤC TÀI LIỆU THAM KHẢO	28

BẢNG MÔ TẢ CÁC THUẬT NGỮ

STT	Thuật ngữ tiếng Anh	Thuật ngữ tiếng Việt
1	Classification And Regression Tree	Thuật toán phân loại và hồi quy
2	Decision tree	Cây quyết định
3	Gini index	Độ thuần khiết
4	Information Gain	Độ lợi thông tin

DANH MỤC CÁC HÌNH

Hình 1. 1. Mô hình cây quyết định.....	1
Hình 2. 1. Cây quyết định phân tách nút đầu tiên	9
Hình 2. 2. Cây quyết định.....	9
Hình 2. 3. Cây quyết định thuật toán CART	16
Hình 3. 1. Dữ liệu sau khi tải về.....	18
Hình 3. 2. Dữ liệu sau khi loại bỏ một số cột.....	19
Hình 3. 3. Kiểm tra dữ liệu thiếu.....	19
Hình 3. 4. Giao diện phần mềm Weka	20
Hình 3. 5. Chọn các thuộc tính của dữ liệu sau khi tải lên.....	21
Hình 3. 6. Tải dữ liệu lên Weka	21
Hình 3. 7. Lựa chọn thuật toán cây quyết định	22
Hình 3. 8. Kết quả trả về sau khi chạy thuật toán C4.5.....	22
Hình 3. 9. Mô hình cây quyết định sử dụng thuật toán C4.5 trên Weka.....	23
Hình 3. 10. Tải thư viện và tệp dữ liệu.....	23
Hình 3. 11. Mã hóa các thuộc tính và gán giá trị x	24
Hình 3. 12. In thông tin dữ liệu và kiểm tra dữ liệu.....	24
Hình 3. 13. Gán giá trị y	25
Hình 3. 14. Tải thêm thư viện và tạo cây quyết định	25
Hình 3. 15. Đo độ chính xác của mô hình.....	25
Hình 3. 16. Tải các thư viện trực quan.....	26
Hình 3. 17. Vẽ cây quyết định.....	26

Hình 3. 18. Thêm điều kiện cắt tia cây quyết định.....	26
Hình 3. 19. Cây quyết định sau khi cắt tia	27

DANH MỤC BẢNG

Bảng 1. 1. Bảng dữ liệu tập huấn	2
Bảng 1. 2. Các thuộc tính làm điều kiện phân tách	3
Bảng 1. 3. Các thể hệ thuật toán cây quyết định	4
Bảng 2. 1. Bảng dữ liệu kết quả hồ sơ các ứng viên	6
Bảng 2. 2. Dữ liệu hồ sơ ứng viên mới	7
Bảng 2. 3. Thống kê số lượng phần tử	8
Bảng 2. 4. Giá trị kết quả dự đoán từ cây quyết định thuật toán ID3	10
Bảng 2. 5. Thống kê số lượng phần tử của thuộc tính <i>Tuoi</i>	14
Bảng 2. 6. Tính Gini index của các thuộc tính	15
Bảng 2. 7. Giá trị kết quả dự đoán từ cây quyết định thuật toán CART	16
Bảng 3. 1. Mô tả các thuộc tính thu thập	17

TÓM TẮT TIỂU LUẬN ĐỒ ÁN

Tiểu luận tập trung nghiên cứu các thuật toán cây quyết định – một trong những kỹ thuật phân lớp dữ liệu. Trong bối cảnh thị trường lao động ngày càng cạnh tranh như hiện nay, các doanh nghiệp cần có công cụ để hỗ trợ việc lựa chọn ứng viên phù hợp với công ty nhằm tiết kiệm thời gian cũng như nguồn lực. Mô hình cây quyết định sẽ giúp doanh nghiệp giải quyết vấn đề lựa chọn hồ sơ ứng viên ngay từ vòng đầu tiên.

Nội dung tiểu luận bao gồm 3 chương:

Chương 1: Khái quát về cây quyết định

Chương 2: Một số thuật toán cây quyết định

Chương 3: Kết quả thực nghiệm

Chương 1 Khái quát về cây quyết định

✍ Nội dung chương này sẽ trình bày về khái quát phân lớp dữ liệu dựa vào cây quyết định.

❖ Cây quyết định

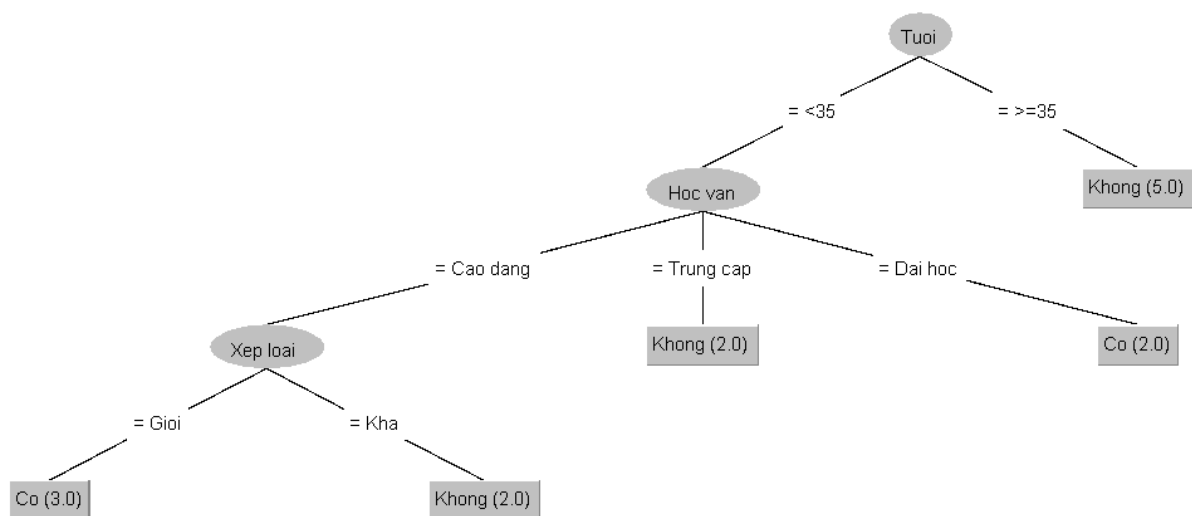
❖ Các thuật toán cây quyết định

1.1. Phân lớp dữ liệu dựa vào cây quyết định

Phân lớp dữ liệu là một kỹ thuật quan trọng trong lĩnh vực khai phá dữ liệu, được sử dụng để phân tích các tập dữ liệu nhằm xác định nhãn (label) của lớp (class) mà một đối tượng dữ liệu cụ thể thuộc về. Mục tiêu chính của kỹ thuật này là xây dựng các mô hình hoặc thuật toán có khả năng mô các đặc điểm của từng lớp dữ liệu, từ đó hỗ trợ việc phân loại hoặc dự đoán xu hướng của dữ liệu chưa biết.

Các kỹ thuật phân lớp dữ liệu có thể kể đến hồi quy logistic, cây quyết định, xác suất dựa vào định lý Bayes, mạng nơ-ron nhân tạo.

Trong đó, cây quyết định (Decision Tree) có cấu trúc dữ liệu dạng cây, được sử dụng để biểu diễn quá trình ra quyết định dựa trên các điều kiện được thiết lập từ các giá trị dữ liệu cụ thể. Cây quyết định bắt đầu từ nút gốc (root) đi qua các nút trung gian hay còn gọi là nút nội (internal node) và kết thúc tại các nút lá (leaf).



Hình 1. 1. Mô hình cây quyết định

Bảng 1. 1. Bảng dữ liệu tập huấn

STT	Tuoi	Hoc van	Xep loai	Gioi tinh	Ket qua
1	<35	Cao dang	Gioi	Nu	Co
2	<35	Cao dang	Gioi	Nam	Co
3	>=35	Trung cap	Gioi	Nu	Khong
4	>=35	Dai hoc	Gioi	Nam	Khong
5	>=35	Dai hoc	Kha	Nu	Khong
6	<35	Dai hoc	Kha	Nam	Co
7	<35	Trung cap	Kha	Nu	Khong
8	<35	Cao dang	Gioi	Nam	Co
9	<35	Cao dang	Kha	Nam	Khong
10	>=35	Dai hoc	Kha	Nam	Khong
11	<35	Cao dang	Kha	Nam	Khong
12	<35	Trung cap	Gioi	Nam	Khong
13	>=35	Trung cap	Kha	Nu	Khong
14	<35	Dai hoc	Gioi	Nam	Co

Cấu trúc cây quyết định bao gồm (Đặng Văn Nam et al., 2018):

- Nút trong/nội (internal node): biểu diễn một phép kiểm tra giá trị đối với các tập thuộc tính. Ví dụ trên *Tuoi*, *Hoc van*, *Xep loai* (hình elip) là nút trong.
- Nút gốc (root): cũng là một nút nội và là nút trên cùng của cây quyết định. Cây quyết định trên có thuộc tính *Tuoi* là nút gốc.
- Nút lá (leaf node): biểu diễn một lớp chứa nhãn. Nhãn *Co*, *Khong* biểu diễn bằng hình chữ nhật chính là nút lá.
- Nút nhánh (label): kết quả từ thuộc tính phân tách tương ứng.

1.2. Quá trình xây dựng cây quyết định

Cây quyết định được xây dựng theo cơ chế phân hoạch tập dữ liệu lớn ban đầu D thành các tập con D_j nhỏ hơn dựa vào điều kiện phân tách tại nút đang xét. Tiêu chí trong việc chọn thuộc tính phân tách dựa vào các độ đo như Gain, GainRatio hay Gini index sẽ được trình bày cụ thể ở Chương 2.

Trong các thuộc tính của tập dữ liệu sẽ có các điều kiện phân tách, 3 loại thuộc tính phân tách được mô tả tại bảng 1.2 dưới đây.

Bảng 1. 2. Các thuộc tính làm điều kiện phân tách

Thuộc tính phân tách	Định nghĩa	Ví dụ
Thuộc tính rời rạc	Thuộc tính có tập giá trị hữu hạn hoặc có thể đếm được, thường là số nguyên hoặc hạng mục.	<ul style="list-style-type: none">• Xếp loại (Trung bình, Khá, Giỏi)• Các loại xe (Xe máy, Oto, Xe buýt)
Thuộc tính liên tục	Thuộc tính có thể là giá trị nào trong một khoảng xác định, thường là số thực.	<ul style="list-style-type: none">• Chiều cao i (150.5 cm, 162.3 cm)• Nhiệt độ (36.5°C, 20.1°C)
Thuộc tính nhị phân	Thuộc tính chỉ có hai giá trị có thể xảy ra, thường được biểu diễn dưới dạng 0 và 1 hoặc True/False hoặc Yes/No.	<ul style="list-style-type: none">• Giới tính: Nam (0), Nữ (1)• Mua hàng (Có, Không)

1.3. Các thuật toán cây quyết định

Theo (Loh, 2015) phân chia các thuật toán cây quyết định thành 4 thể hệ được trình bày tại bảng 1.3. Trong đó, thuật toán ID3, C4.5 và CART là 3 thuật toán được biết đến rộng rãi khi sử dụng độ đo

Bảng 1. 3. Các thể hệ thuật toán cây quyết định

Các thể hệ	Các thuật toán
Thế hệ 1	AID (Morgan and Sonquist, 1963), THAID (Messenger and Mandell, 1972), CHAID (Kass, 1980)
Thế hệ 2	CART (Breiman et al., 1984) , RECPAM (Ciampi et al., 1988), Segal (1988, 1992), LeBlanc and Crowley (1992), Alexander and Grimshaw (1996), Zhang (1998), MVPART (De'ath, 2002), Su et al. (2004); ID3 (Quinlan, 1986), M5 (Quinlan, 1992), C4.5 (Quinlan, 1993); FACT (Loh and Vanichsetakul, 1988)
Thế hệ 3	QUEST (Loh and Shih, 1997), CRUISE (Kim and Loh, 2001, 2003), Bayesian CART (Chipman et al., 1998; Denison et al., 1998)
Thế hệ 4	GUIDE (Loh, 2002, 2009; Loh and Zheng, 2013; Loh et al., 2015), CTREE (Hothorn et al., 2006), MOB (Zeileis et al., 2008); Random forest (Breiman, 2001), TARGET (Fan and Gray, 2005; Gray and Fan, 2008), BART (Chipman et al., 2010)

Chương 2

Một số thuật toán cây quyết định

✍ Chương này giới thiệu chi tiết về các thuật toán cây quyết định bao gồm ID3, C4.5, và CART. Mỗi thuật toán sẽ được trình bày về ý nghĩa, các bước tính toán, và cách lựa chọn thuộc tính phân tách dựa trên các độ đo như Information Gain, Gain Ratio, và Gini Index

- ❖ Thuật toán ID3
- ❖ Thuật toán C4.5
- ❖ Thuật toán CART

2.1. Thuật toán ID3

Thuật toán ID3 (Iterative Dichotomiser 3) được phát triển bởi Quinlan (1986), đây là thuật toán sử dụng độ lợi thông tin (Information gain) để phân tích thuộc tính thông qua việc xác định tính thuần khiết (purity) lớn nhất hay tính trùng lặp (impurity)/ngẫu nhiên (randomness) của các phần tử là nhỏ nhất. (Trần Minh Quang, 2020)

Để tìm được độ lợi thông tin bước đầu cần thực hiện tính sự hỗn loạn (entropy) của tập D thông qua $Info(D)$ là lượng thông tin cần thiết để phân lớp một phần tử a từ tập D . $Info(D)$ được tính như sau:

$$Info(D) = -\sum_{i=1}^m p_i \log_2(p_i) \quad (1)$$

Trong đó p_i là xác suất để một phần tử thuộc về một lớp C_i trong tập D và được tính theo công thức:

$$p_i = \frac{|C_i \cap D|}{|D|} \quad (2)$$

Với thuộc tính A bất kỳ được chọn làm thuộc tính phân tách và để tính lượng thông tin cần phân lớp dựa trên thuộc tính A ($Info_A(D)$) công thức như sau:

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} Info(D_j) \quad (3)$$

Giải thích:

- Thuộc tính A được phân tách thành v điều kiện phân tách.
- D_j là tập con chứa giá trị v của thuộc tính A.
- Độ đo $Info_A(D)$ càng nhỏ càng tốt, bởi lẽ nó cho biết mức độ trùng lặp giữa các tập con tức là phần tử cùng thuộc 1 lớp có thể thuộc các tập con khác nhau.

Cuối cùng độ đo *Gain* (độ lợi thông tin) chính là sự chênh lệch giữa lượng thông tin gốc ($Info(D)$) và lượng thông tin mới sau khi phân hoạch dựa trên thuộc tính A ($Info_A(D)$). Thuộc tính được chọn phân tách khi *Gain* (ký hiệu $Gain_{thuộc\ tính}$) lớn nhất. Công thức tính dưới đây:

$$Gain_A = Info(D) - Info_A(D) \quad (4)$$

Ví dụ chọn thuộc tính phân tách dùng độ đo Information Gain:

Với tập dữ liệu D dưới đây, tính *Gain* và tạo cây quyết định, đưa ra quyết định cho hồ sơ của những ứng viên mới?

Bảng 2. 1. Bảng dữ liệu kết quả hồ sơ các ứng viên

STT	Tuoi	Hoc van	Xep loai	Gioi tinh	Ket qua
1	<35	Cao dang	Gioi	Nu	Co
2	<35	Cao dang	Gioi	Nam	Co
3	>=35	Trung cap	Gioi	Nu	Khong
4	>=35	Dai hoc	Gioi	Nam	Khong
5	>=35	Dai hoc	Kha	Nu	Khong
6	<35	Dai hoc	Kha	Nam	Co
7	<35	Trung cap	Kha	Nu	Khong
8	<35	Cao dang	Gioi	Nam	Co
9	<35	Cao dang	Kha	Nam	Khong
10	>=35	Dai hoc	Kha	Nam	Khong

11	<35	Cao dang	Kha	Nam	Khong
12	<35	Trung cap	Gioi	Nam	Khong
13	>=35	Trung cap	Kha	Nu	Khong
14	<35	Dai hoc	Gioi	Nam	Co

Bảng 2. 2. Dữ liệu hồ sơ ứng viên mới

STT	Tuoi	Hoc van	Xep loai	Gioi tinh	Ket qua
1	<35	Cao dang	Kha	Nu	Co/ Khong?
2	>=35	Cao dang	Gioi	Nam	Co/ Khong?
3	<35	Trung cap	Gioi	Nu	Co/ Khong?
4	>=35	Dai hoc	Gioi	Nam	Co/Khong?
5	<35	Dai hoc	Kha	Nam	Co/Khong?

Các bước tính toán

- (1) Tính $Info(D)$.
 - (2) Tính $Info_{thuộc tính}(D)$.
 - (3) Tính $Gain_{thuộc tính}$.
 - (4) So sánh $Gain_{thuộc tính}$ chọn ra thuộc tính có giá trị cao nhất làm thuộc tính phân tách.
-

(1): Tập dữ liệu D trên có 14 phần tử, trong đó 9 phần tử thuộc lớp “Khong” và 5 phần tử thuộc lớp “Co”. Áp dụng công thức (1) $Info(D)$ được tính như sau:

$$Info(D) = -\left(\frac{9}{14}\log_2\left(\frac{9}{14}\right) + \frac{5}{14}\log_2\left(\frac{5}{14}\right)\right) = 0.94$$

(2): Xét thuộc tính *Tuoi* làm thuộc tính phân tách, ta có 2 tập con $D_1 (<35)$ và $D_2 (>=35)$. Trong đó, tập con D_1 có 9 phần tử (5 phần tử thuộc lớp “Co”, 4 phần tử thuộc lớp “Khong”), tập con D_2 có 5 phần tử đều thuộc lớp “Khong”. Áp dụng công thức (3) ta được:

$$Info_{Tuoi}(D) = \frac{9}{14} \left(- \left(\frac{5}{9} \log_2 \frac{5}{9} + \frac{4}{9} \log_2 \frac{4}{9} \right) \right) + \frac{5}{14} \left(- \left(\frac{5}{5} \log_2 \frac{5}{5} \right) \right) = 0.63$$

Các thuộc tính *Hoc van*, *Xep loai*, *Gioi tinh* được tính tương tự như trên. Dưới đây là bảng tổng hợp đếm số lượng phần tử thuộc các lớp của từng thuộc tính.

Bảng 2. 3. Thống kê số lượng phần tử

Thuộc tính		D_j	Co	Khong
Tuoi	<35	9	5	4
	>=35	5	0	5
Hoc van	Trung cap	4	0	4
	Cao dang	5	3	2
	Dai hoc	5	2	3
Xep loai	Kha	7	1	6
	Gioi	7	4	3
Gioi tinh	Nam	9	4	5
	Nu	5	1	4

$$Info_{Hoc van}(D) = \frac{4}{14} \left(- \left(\frac{4}{4} \log_2 \frac{4}{4} \right) \right) + \frac{5}{14} \left(- \left(\frac{3}{5} \log_2 \frac{3}{5} + \frac{2}{5} \log_2 \frac{2}{5} \right) \right) + \frac{5}{14} \left(- \left(\frac{2}{5} \log_2 \frac{2}{5} + \frac{3}{5} \log_2 \frac{3}{5} \right) \right) = 0.69$$

$$Info_{Xep loai}(D) = \frac{7}{14} \left(- \left(\frac{1}{7} \log_2 \frac{1}{7} + \frac{6}{7} \log_2 \frac{6}{7} \right) \right) + \frac{7}{14} \left(- \left(\frac{4}{7} \log_2 \frac{4}{7} + \frac{3}{7} \log_2 \frac{3}{7} \right) \right) = 0.78$$

$$Info_{Gioi tinh}(D) = \frac{9}{14} \left(- \left(\frac{4}{9} \log_2 \frac{4}{9} + \frac{5}{9} \log_2 \frac{5}{9} \right) \right) + \frac{5}{14} \left(- \left(\frac{1}{5} \log_2 \frac{1}{5} + \frac{4}{5} \log_2 \frac{4}{5} \right) \right) = 0.89$$

(3): Áp dụng công thức (4) ta tính được Gain của các thuộc tính như sau:

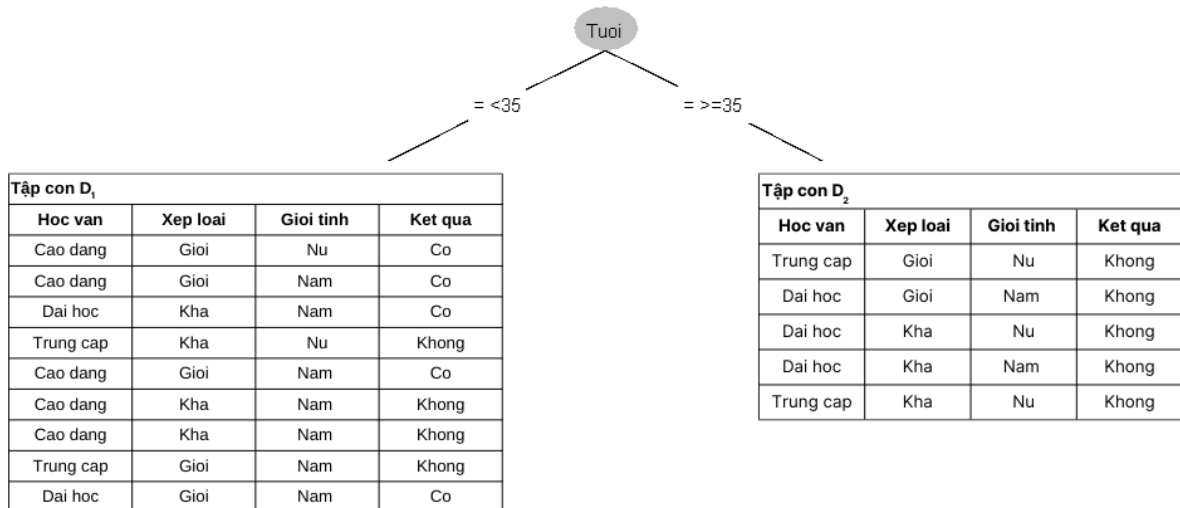
$$Gain_{Tuoi} = Info(D) - Info_{Tuoi}(D) = 0.94 - 0.63 = \mathbf{0.31}$$

$$Gain_{Hoc\ van} = Info(D) - Info_{Hoc\ van}(D) = 0.94 - 0.69 = 0.25$$

$$Gain_{Xep\ loai} = Info(D) - Info_{Xep\ loai}(D) = 0.94 - 0.78 = 0.16$$

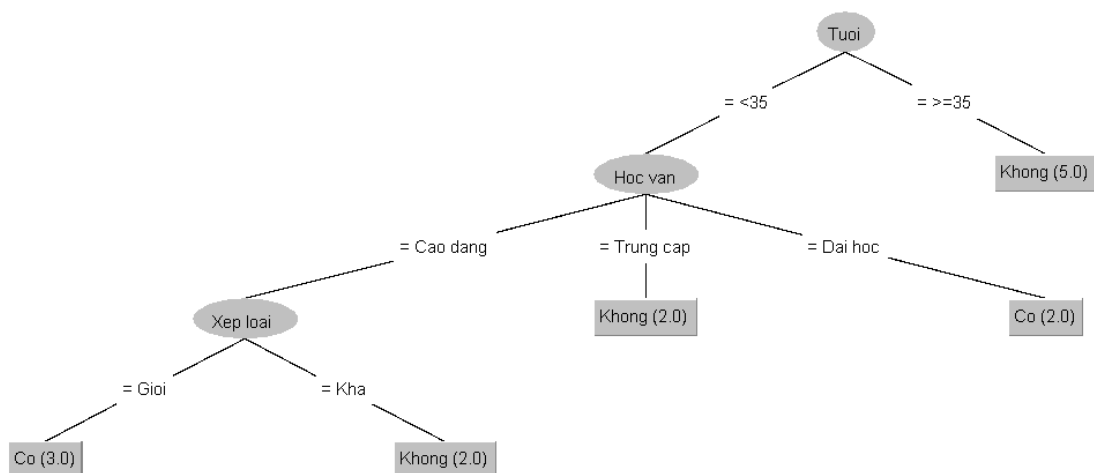
$$Gain_{Gioi\ tinh} = Info(D) - Info_{Gioi\ tinh}(D) = 0.94 - 0.89 = 0.05$$

(4): Như vậy, thuộc tính *Tuoi* có độ đo Information gain lớn nhất và được chọn làm thuộc tính phân tách đầu tiên được xem là nút gốc.



Hình 2. 1. Cây quyết định phân tách nút đầu tiên

Tiếp tục thực hiện lại các bước đối với từng tập con được phân hoạch cho đến khi tất cả phần tử thuộc về cùng 1 nhãn. Chẳng hạn, như tập con D₂ trên tất cả các phần tử đều thuộc về nhãn “*Khong*”. Sau cùng ta được cây quyết định như hình 2.2.



Hình 2. 2. Cây quyết định

Từ cây quyết định trên các tập luật sinh ra như sau

NẾU Tuổi = “>=35” THÌ “Khong”

NẾU Tuổi = “<35” VÀ Học van = “Đại học” THÌ “Co”

NẾU Tuổi = “<35” VÀ Học van = “Trung cấp” THÌ “Khong”

NẾU Tuổi = “<35” VÀ Học van = “Cao đẳng” VÀ Xếp loại = “Giỏi” THÌ “Co”

NẾU Tuổi = “<35” VÀ Học van = “Cao đẳng” VÀ Xếp loại = “Kha” THÌ “Khong”

Dựa vào tập luật trên ta có thể dự đoán kết quả hồ sơ của các ứng viên mới như sau:

Bảng 2. 4. Giá trị kết quả dự đoán từ cây quyết định thuật toán ID3

STT	Tuoi	Hoc van	Xep loai	Gioi tinh	Ket qua
1	<35	Cao dang	Kha	Nu	Khong
2	>=35	Cao dang	Giỏi	Nam	Khong
3	<35	Trung cap	Giỏi	Nu	Khong
4	>=35	Đại học	Giỏi	Nam	Khong
5	<35	Đại học	Kha	Nam	Co

2.2. Thuật toán C4.5

Thuật toán C4.5 là phiên bản cải tiến từ thuật toán ID3 được Quinlan(1993) phát triển. Thuật toán C4.5 sử dụng độ đo gain ratio (hệ số độ lợi thông tin) là một biến thể của Information gain nhằm khắc phục nhược điểm phân hoạch có kích thước quá nhỏ dẫn đến bùng nổ cây quyết định và mất thời gian xây dựng mô hình. Thuộc tính được chọn là nút là thuộc tính có GainRatio cao nhất.

Công thức tính:

$$GainRatio_A = \frac{Gain_A}{SplitInfo_A(D)} \quad (5)$$

Trong đó $SplitInfo_A(D)$ được tính theo công thức:

$$SplitInfo_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} \log_2 \left(\frac{|D_j|}{|D|} \right) \quad (6)$$

Ví dụ chọn thuộc tính phân tách sử dụng độ đo GainRatio

Sử dụng dữ liệu tại bảng 2.1 các bước tính toán như sau:

Các bước tính toán

- (1) Tính $Gain_{thuộc tính}$.
 - (2) Tính $SplitInfo_{thuộc tính}(D)$.
 - (3) Tính $GainRatio_{thuộc tính}$.
 - (4) So sánh $GainRatio_{thuộc tính}$ chọn ra thuộc tính có giá trị cao nhất làm thuộc tính phân tách.
-

(1) Ví dụ trên đã tính Gain của các thuộc tính và có:

$$Gain_{Tuoi} = 0.31$$

$$Gain_{Hoc van} = 0.25$$

$$Gain_{Xep loai} = 0.16$$

$$Gain_{Gioi tinh} = 0.05$$

(2) Tính $SplitInfo_{thuộc tính}(D)$

Thuộc tính *Tuoi* có 2 tập con (D_1 có 9 phần tử, D_2 có 5 phần tử), áp dụng công thức (6) ta được:

$$SplitInfo_{Tuoi}(D) = - \left(\frac{9}{14} \log_2 \left(\frac{9}{14} \right) + \frac{5}{14} \log_2 \left(\frac{5}{14} \right) \right) = 0.94$$

Tương tự, kết quả của các thuộc tính khác:

$$SplitInfo_{Hoc van}(D) = - \left(\frac{4}{14} \log_2 \left(\frac{4}{14} \right) + \frac{5}{14} \log_2 \left(\frac{5}{14} \right) + \frac{5}{14} \log_2 \left(\frac{5}{14} \right) \right) = 1.57$$

$$SplitInfo_{Xep\ loai}(D) = -\left(\frac{7}{14}\log_2\left(\frac{7}{14}\right) + \frac{7}{14}\log_2\left(\frac{7}{14}\right)\right) = 1$$

$$SplitInfo_{Gioi\ tinh}(D) = -\left(\frac{9}{14}\log_2\left(\frac{9}{14}\right) + \frac{5}{14}\log_2\left(\frac{5}{14}\right)\right) = 0.94$$

(3) Tính GainRatio của các thuộc tính trên:

$$GainRatio_{Tuoi} = \frac{Gain_{Tuoi}}{SplitInfo_{Tuoi}(D)} = \mathbf{0.329}$$

$$GainRatio_{Hoc\ van} = \frac{Gain_{Hoc\ van}}{SplitInfo_{Hoc\ van}(D)} = 0.159$$

$$GainRatio_{Xep\ loai} = \frac{Gain_{Xep\ loai}}{SplitInfo_{Xep\ loai}(D)} = 0.16$$

$$GainRatio_{Gioi\ tinh} = \frac{Gain_{Gioi\ tinh}}{SplitInfo_{Gioi\ tinh}(D)} = 0.05$$

(4) So sánh GainRatio

Như vậy, thuộc tính Tuổi là thuộc tính có *GainRatio* được chọn làm thuộc tính phân tách.

2.3. Thuật toán CART

Khác với 2 thuật toán ID3 và C4.5 trong thuật CART (Classification And Regression Tree) sử dụng độ đo Gini index để xây dựng cây quyết định dưới dạng phân nhánh nhị phân tức là mỗi nút chỉ có 2 nhánh. Công thức tính Gini của toàn tập dữ liệu:

$$Gini(D) = 1 - \sum_{j=1}^m p_i^2 \quad (7)$$

Khác với 2 độ đo Gain và GainRatio, Gini index là độ đo không thuần khiết của tập dữ liệu D dùng để đánh giá phân tách nhị phân. Xét thuộc tính A trong tập D có v giá trị, với S_A là tập con hay là điểm phân tách của thuộc tính A, để xác định điểm chia tốt nhất cần kiểm tra nhị phân dạng $A \in S_A$ và số lượng tập con với v giá trị được tính theo công thức:

$$Số\ lượng\ tập\ con = 2^v - 2 \quad (8)$$

Trong công thức trên số lượng tập con trừ đi 2 vì bao gồm cả tập toàn phần và tập rỗng. Ví dụ xét thuộc tính *Hoc van*, ta có $v = \{Trung\ cap, Cao\ dang, Dai\ hoc\}$ thì số lượng tập con là $2^3 = 8$ tập, liệt kê dưới đây:

$$\{Trung\ cap, Cao\ dang, Dai\ hoc\}$$

$\{Trung\ cap, Cao\ dang\}$

$\{Trung\ cap, Dai\ hoc\}$

$\{Cao\ dang, Dai\ hoc\}$

$\{Dai\ hoc\}$

$\{Cao\ dang\}$

$\{Trung\ cap\}$

$\{\}$

Tập toàn phần và tập rỗng được loại bỏ vì không thể tách nhị phân tập dữ liệu D thành 2 tập con khác nhau. Do đó, số lượng tập con dùng để phân tác là $2^3 - 2 = 6$.

Để xác định độ không thuần khiết của thuộc tính A khi phân tách ta tính $Gini_A(D)$ theo công thức:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \quad (9)$$

Với D_1 và D_2 là 2 tập được phân hoạch từ thuộc tính A. Trái ngược với 2 độ đo trước, thuộc tính được chọn phân tách khi Gini index của thuộc tính nào nhỏ nhất hoặc sự chênh lệch khi so với độ không thuần khiết ban đầu lớn nhất. Công thức tính độ chênh lệch Gini index:

$$\Delta Gini(A) = Gini(D) - Gini_A(D) \quad (10)$$

Ví dụ chọn thuộc tính phân tách sử dụng độ đo Gini Index

Tiếp tục sử dụng đề bài mục 2.1, các bước tính toán như sau:

Các bước tính toán

- (1) Tính $Gini(D)$
 - (2) Tính $Gini_{thuộc\ tính}(D)$. Chọn ra $Gini_{thuộc\ tính}(D)$ có giá trị nhỏ nhất.
 - (3) Tính $\Delta Gini(Thuộc\ tính)$.
 - (4) So sánh $\Delta Gini(Thuộc\ tính)$ chọn ra thuộc tính có giá trị cao nhất làm thuộc tính phân tách.
-

(1) Tính $Gini(D)$

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

(2) Tính $Gini_{thuộc tính}(D)$ theo công thức...

Xét thuộc tính Tuổi có 2 tập con $S_A = \{<35, \geq 35\}$:

Bảng 2. 5. Thống kê số lượng phần tử của thuộc tính Tuổi

Thuộc tính		D_j	Co	Khong	$S_{Thuộc tính}$	D_1	D_2
Tuoi	<35	9	5	4	{<35}	9 (5 Co, 4 Khong)	5 (5 Khong)
	≥ 35	5	0	5	{ ≥ 35 }	5 (5 Khong)	9 (5 Co, 4 Khong)

Với điều kiện “Tuoi $\in \{<35\}$ ” thì ta phân hoạch được D_1 (phần tử <35) gồm 9 phần tử và D_2 (phần tử ≥ 35) có 5 phần tử. Trong đó, tập D_1 có 5 phần tử thuộc lớp “Co”, 4 phần tử lớp “Khong”, tập D_2 có 5 phần tử đều thuộc lớp “Khong”. Áp dụng công thức.... ta tính được $Gini(D_1), Gini(D_2)$:

$$Gini(D_1) = 1 - \left(\frac{5}{9}\right)^2 - \left(\frac{4}{9}\right)^2 = 0.49$$

$$Gini(D_2) = 1 - \left(\frac{5}{5}\right)^2 = 0$$

Áp dụng công thức... ta tính được $Gini_{Tuoi}(D)$:

$$Gini_{Tuoi}(D) = \frac{|9|}{|14|} \times 0.49 + \frac{|5|}{|14|} \times 0 = 0.315$$

Với điều kiện “Tuoi $\in \{\geq 35\}$ ” làm tương tự như trên, tuy nhiên vì thuộc tính Tuổi chỉ có 2 giá trị nên khi phân hoạch thì Gini index của 2 giá trị sẽ bằng nhau.

Thực hiện tương tự ta có được bảng dưới đây, trong đó thuộc tính Học van cần chọn ra giá trị $Gini_A(D)$ nhỏ nhất để làm độ đo Gini index.

Bảng 2. 6. Tính Gini index của các thuộc tính

Thuộc tính		Co	Khong	S _{Thuộc tính}	D ₁	D ₂	Gini(D ₁)	Gini(D ₂)	Gini _A (D)
Hoc van	Trung cap	0	4	{Trung cap, Cao dang}	9	5	0.444	0.48	0.456
	Cao dang	3	2	{Trung cap, Dai hoc }	9	5	0.345	0.48	0.3932
	Dai hoc	2	3	{Cao dang, Dai hoc}	10	4	0.5	0	0.357
				{Dai hoc}	5	9	0.48	0.444	0.456
				{Cao dang}	5	9	0.48	0.345	0.3932
				{Trung cap}	4	10	0	0.5	0.357
Xep loai	Kha	1	6	{Kha}	7	7	0.244	0.489	0.366
	Gioi	4	3	{Gioi}	7	7	0.489	0.244	0.366
Gioi tinh	Nam	4	5	{Nam}	9	5	0.493	0.32	0.4312
	Nu	1	4	{Nu}	5	9	0.32	0.493	0.4312

Từ bảng trên ta rút ra được:

$$Gini_{Tuoi}(D) = 0.315$$

$$Gini_{Hoc\ van}(D) = 0.357$$

$$Gini_{Xep\ loai}(D) = 0.366$$

$$Gini_{Gioi\ tinh}(D) = 0.4312$$

(3) Tính $\Delta Gini(Thuộc\ tính)$.

$$\Delta Gini(Tuoi) = Gini(D) - Gini_{Tuoi}(D) = 0.459 - 0.315 = 0.144$$

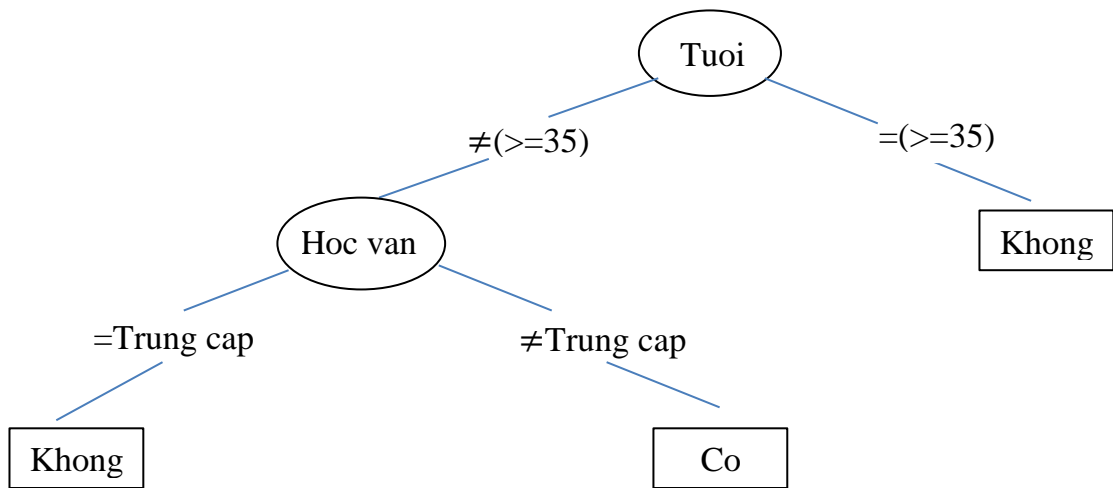
$$\Delta Gini(Hoc\ van) = Gini(D) - Gini_{Hoc\ van}(D) = 0.459 - 0.357 = 0.102$$

$$\Delta Gini(Xep\ loai) = Gini(D) - Gini_{Xep\ loai}(D) = 0.459 - 0.366 = 0.093$$

$$\Delta Gini(Gioi\ tinh) = Gini(D) - Gini_{Gioi\ tinh}(D) = 0.459 - 0.4312 = 0.0278$$

(4) So sánh $\Delta Gini(Thuộc\ tính)$. Chọn ra thuộc tính sự chênh lệch cao nhất.

Ta có thể thấy $\Delta Gini(Tuoi)$ có giá trị lớn nhất, do đó thuộc tính Tuoi sẽ là thuộc tính được phân tách đầu tiên. Tiếp tục thực hiện lại các bước trên cho nút tiếp theo, cuối cùng ta có cây quyết định như hình 2.3.



Hình 2. 3. Cây quyết định thuật toán CART

NẾU Tuổi = “>=35” THÌ “Không”

NẾU Tuổi ≠ “>=35” VÀ Học van = “Trung cap” THÌ “Không”

NẾU Tuổi ≠ “>=35” VÀ Học van ≠ “Trung cap” THÌ “Có”

Từ tập luật trên ta có thể dự đoán kết quả hồ sơ các ứng viên như sau

Bảng 2. 7. Giá trị kết quả dự đoán từ cây quyết định thuật toán CART

STT	Tuoi	Hoc van	Xep loai	Giới tính	Ket qua
1	<35	Cao dang	Kha	Nu	Co
2	>=35	Cao dang	Gioi	Nam	Khong
3	<35	Trung cap	Gioi	Nu	Khong
4	>=35	Dai hoc	Gioi	Nam	Khong
5	<35	Dai hoc	Kha	Nam	Co

Chương 3

Kết quả thực nghiệm

✍ Bài toán đặt ra nếu một công ty đang tuyển dụng nhân sự dựa vào các thuộc tính của nhân viên trước đây để tạo ra mô hình cây quyết định nhằm lọc ra những hồ sơ phù hợp. Chương này sẽ xây dựng cây quyết định sử dụng dữ liệu tập huấn thu thập từ Kaggle thông qua các công cụ như phần mềm Weka, Python.

- ❖ Xây dựng cây quyết định từ phần mềm Weka
- ❖ Xây dựng cây quyết định từ Python

3.1. Đặt vấn đề

Giả sử một công ty đang tuyển dụng cho vị trí lập trình viên, với một hồ sơ ứng viên sẽ bao gồm những thuộc tính quan trọng ảnh hưởng tới quyết định lựa chọn ứng viên đó. Với nhu cầu tuyển dụng của công ty sẽ có nhiều hồ sơ nộp vào đăng ký, vì vậy để hỗ trợ cho việc lựa chọn ứng viên phù hợp cần xây dựng mô hình từ dữ liệu lịch sử tuyển dụng để đưa ra quyết định lựa chọn ứng viên mới.

3.2. Thu thập dữ liệu

Dữ liệu tập huấn được lấy từ bộ dữ liệu [70k+ Job Applicants Data \(Human Resource\)](#) tại Kaggle (AyushTankha, n.d.). Đây là bộ dữ liệu chứa thông tin của 70 nghìn ứng viên xin việc liên quan đến Công nghệ Thông tin. Tuy nhiên, trong tiểu luận đồ án này chỉ sử dụng dữ liệu tập huấn của khoảng 1000 ứng viên. Các thuộc tính thu thập được mô tả tại bảng 3.1. Phần mềm sử dụng xây dựng cây quyết định là Weka và ngôn ngữ lập trình Python.

Bảng 3. 1. Mô tả các thuộc tính thu thập

STT	Tên thuộc tính	Giá trị thuộc tính	Ý nghĩa
1	Age	<35, >35	Cho biết độ tuổi của ứng viên

2	EdLevel	Undergraduate, PhD, Master, NoHigherEd, Other	Cho biết trình độ học vấn của ứng viên
3	Gender	Man, Woman, NonBinary	Cho biết giới tính của ứng viên
4	MainBranch	Dev, NotDev	Cho biết ứng viên có thuộc nhà phát triển chuyên nghiệp hay không
5	YearsCode	1, 2, 3,...	Thời gian mà ứng viên biết lập trình
6	ComputerSkills	1, 2, 3,...	Số lượng kỹ năng máy tính/ngôn ngữ lập trình của ứng viên
7	Employed	0,1	Kết quả hồ sơ (0: loại, 1: nhận)

3.3. Tiền xử lý dữ liệu

Dữ liệu tải về là tập dữ liệu Excel:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		Age	Accessibil	EdLevel	Employme	Gender	MentalHea	MainBranc	YearsCode	YearsCode	Country	PreviousSc	HaveWork	ComputerS	Employed	
2		0 <35	No	Master	1	Man	No	Dev	7		4 Sweden	51552	C++;Pytho	4		0
3		1 <35	No	Undergrad	1	Man	No	Dev	12		5 Spain	46482	Bash/Shell	12		1
4		2 <35	No	Master	1	Man	No	Dev	15		6 Germany	77290	C++;Java	7		0
5		3 <35	No	Undergrad	1	Man	No	Dev	9		6 Canada	46135	Bash/Shell	13		0
6		4 >35	No	PhD	0	Man	No	NotDev	40		30 Singapore	160932	C++;Pytho	2		0
7		5 <35	No	Master	1	Man	No	Dev	9		2 France	38915	JavaScript	5		0
8		6 >35	No	Master	1	Man	No	Dev	26		18 Germany	77831	C++;HTML	17		1
9		7 <35	No	Master	1	Man	No	NotDev	14		5 Switzerlan	81319	C++;Pytho	4		0
10		8 >35	No	Undergrad	1	Man	No	Dev	39		21 United King	68507	Python;Git	3		0
11		9 >35	No	Master	1	Man	No	Dev	20		16 Russian Fe	37752	Delphi;Java	6		0
12		10 <35	No	Undergrad	1	Man	Yes	Dev	4		2 Israel	122580	Bash/Shell	18		1
13		11 <35	No	Undergrad	1	Man	Yes	Dev	6		2 Turkey	11832	Assembly;C	13		1
14		12 <35	No	Master	1	Man	No	Dev	19		10 Germany	60535	C++;Java	5		0
15		13 <35	No	Undergrad	1	Man	No	Dev	8		0 United Stat	103000	Assembly;C	7		0
16		14 <35	No	Master	1	Man	No	NotDev	6		3 France	25944	Bash/Shell	5		0
17		15 >35	No	Undergrad	1	Man	Yes	Dev	22		15 Brazil	60480	C#;C++;Ja	16		1
18		16 <35	No	Other	1	Man	Yes	Dev	7		1 Bulgaria	20556	C++;SQL;C	5		0
19		17 <35	No	Undergrad	1	Man	No	Dev	12		6 Greece	25944	C#;HTML/C	25		1
20		18 >35	No	Master	1	Man	Yes	Dev	34		12 United King	64630	Bash/Shell	14		1
21		19 >35	No	Master	1	Man	No	Dev	21		16 Italy	54049	C#;HTML	17		1
22		20 <35	No	Other	1	Man	No	NotDev	5		3 Russian Fe	22644	Bash/Shell	12		0
23		21 >35	Yes	Other	1	Man	Yes	Dev	25		10 Canada	71850	C#;VBA;Gi	6		0
24		22 >35	No	Master	1	Man	No	Dev	20		10 Spain	58373	Bash/Shell	7		1
25		23 >35	No	Undergrad	1	Man	No	Dev	24		18 United King	129266	HTML/CSS	12		0
26		24 <35	No	Master	1	Man	No	NotDev	10		2 Netherland	25944	Bash/Shell	7		0
27		25 >35	No	Undergrad	1	Man	No	Dev	25		15 United Stat	105000	HTML/CSS	6		0

Hình 3. 1. Dữ liệu sau khi tải về

Sau khi tải dữ liệu về, thực hiện loại bỏ những thuộc tính không cần thiết.

	A	B	C	D	E	F	G	H	I	
1		Age	EdLevel	Gender	MainBranch	YearsCodePro	ComputerSkills	Employed		
2		0 <35	Master	Man	Dev	4	4	0		
3		1 <35	Undergrad	Man	Dev	5	12	1		
4		2 <35	Master	Man	Dev	6	7	0		
5		3 <35	Undergrad	Man	Dev	6	13	0		
5		4 >35	PhD	Man	NotDev	30	2	0		
7		5 <35	Master	Man	Dev	2	5	0		
3		6 >35	Master	Man	Dev	18	17	1		
9		7 <35	Master	Man	NotDev	5	4	0		
0		8 >35	Undergrad	Man	Dev	21	3	0		
1		9 >35	Master	Man	Dev	16	6	0		
2		10 <35	Undergrad	Man	Dev	2	18	1		
3		11 <35	Undergrad	Man	Dev	2	13	1		
4		12 <35	Master	Man	Dev	10	5	0		
5		13 <35	Undergrad	Man	Dev	0	7	0		
6		14 <35	Master	Man	NotDev	3	5	0		
7		15 >35	Undergrad	Man	Dev	15	16	1		
8		16 <35	Other	Man	Dev	1	5	0		
9		17 <35	Undergrad	Man	Dev	6	25	1		
0		18 >35	Master	Man	Dev	12	14	1		
1		19 >35	Master	Man	Dev	16	17	1		
2		20 <35	Other	Man	NotDev	3	12	0		
3		21 >35	Other	Man	Dev	10	6	0		
4		22 >35	Master	Man	Dev	10	7	1		
5		23 >35	Undergrad	Man	Dev	18	12	0		
6		24 <35	Master	Man	NotDev	2	7	0		
7		25 >35	Undergrad	Man	Dev	15	6	0		

Hình 3. 2. Dữ liệu sau khi loại bỏ một số cột

Sử dụng hàm COUNTIF() để đếm ô thiếu dữ liệu, kết quả trả về là không tức tập dữ liệu không có lỗi thiếu dữ liệu (missing data).

J2											
	A	B	C	D	E	F	G	H	I	J	K
1	Age	EdLevel	Gender	MainBranch	YearsCodePro	ComputerSkills	Employed				
2	<35	Master	Man	Dev	4	4	0			0	
3	<35	Undergrad	Man	Dev	5	12	1				
4	<35	Master	Man	Dev	6	7	0				
5	<35	Undergrad	Man	Dev	6	13	0				
6	>35	PhD	Man	NotDev	30	2	0				
7	<35	Master	Man	Dev	2	5	0				
8	>35	Master	Man	Dev	18	17	1				

Hình 3. 3. Kiểm tra dữ liệu thiếu

Chuyển dữ liệu tại thuộc tính Employed với 0 là No, 1 là Yes. Lấy 1000 dòng dữ liệu làm dữ liệu tập huấn.

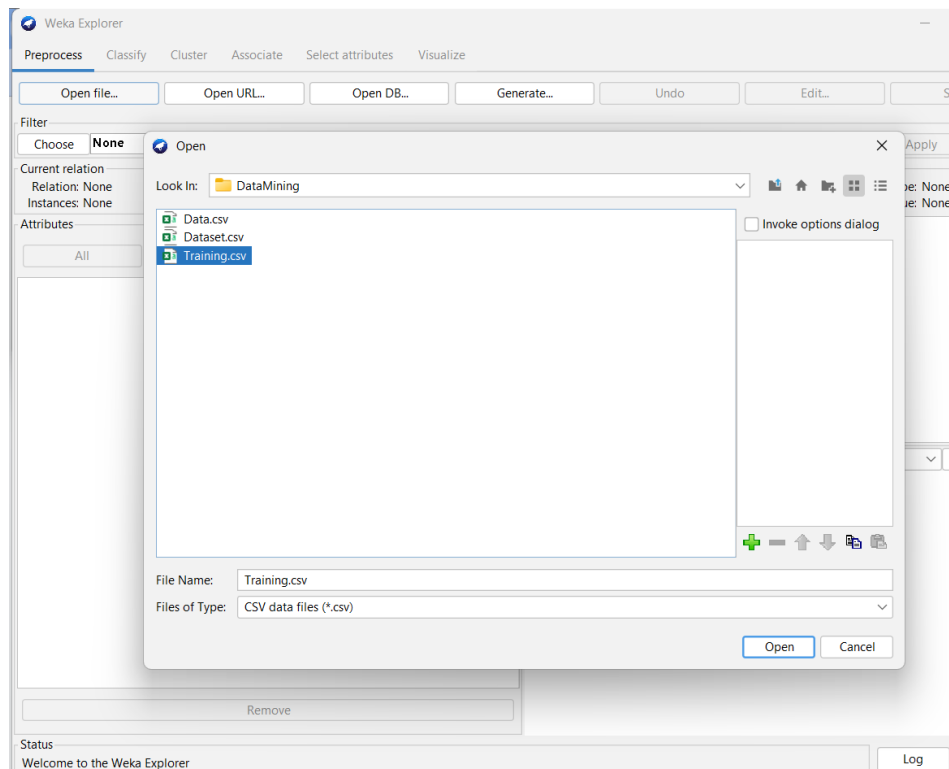
3.4. Xây dựng cây quyết định

3.4.1. Phần mềm Weka

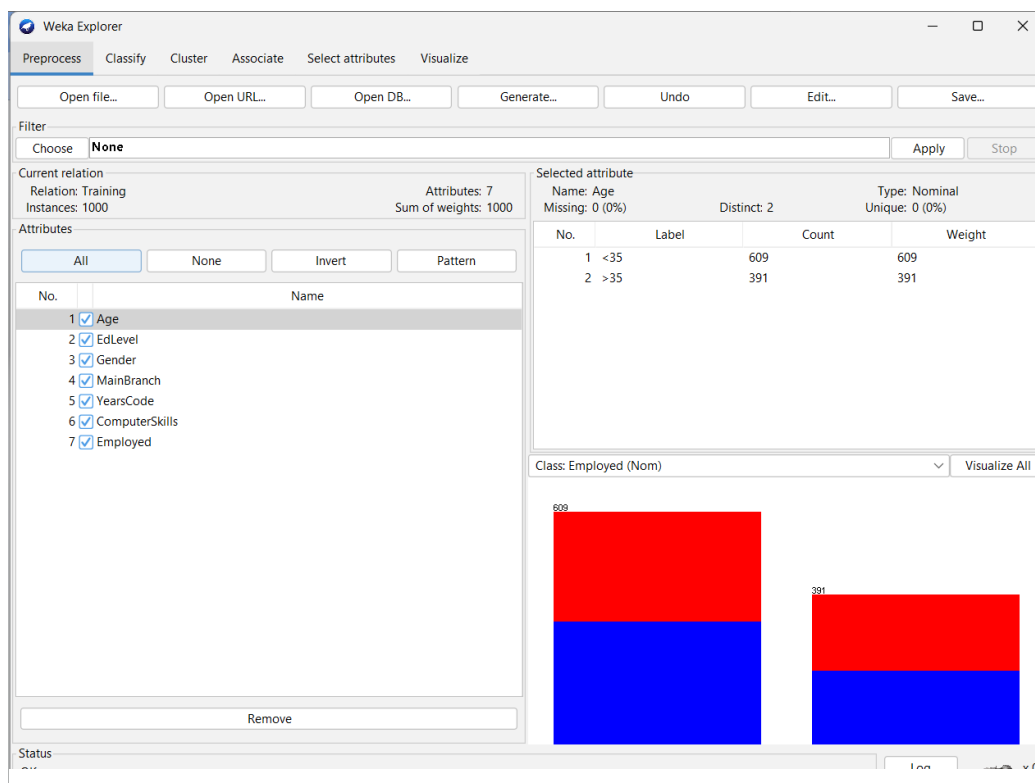
Phần mềm Weka là phần mềm mã nguồn mở do các nhà nghiên cứu của Đại học Waikato tại New Zealand phát triển vào khoảng năm 2011. Đây là phần mềm hỗ trợ khai thác dữ liệu với các công cụ phân tích dữ liệu được sử dụng để trích xuất thông tin và triển khai các sơ đồ máy học (Machine learning).



Hình 3. 4. Giao diện phần mềm Weka

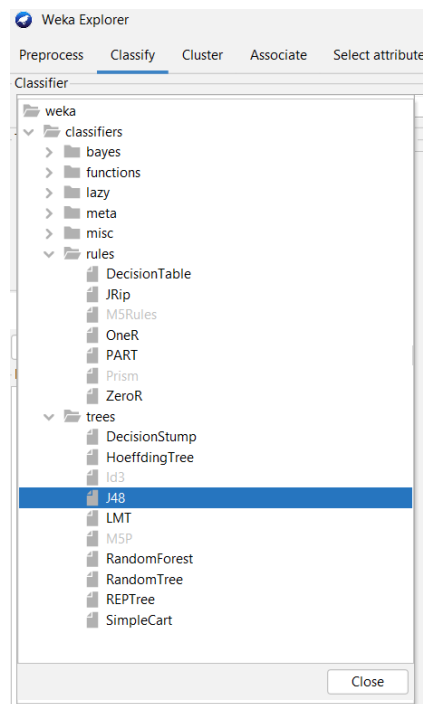


Hình 3. 6. Tải dữ liệu lên Weka

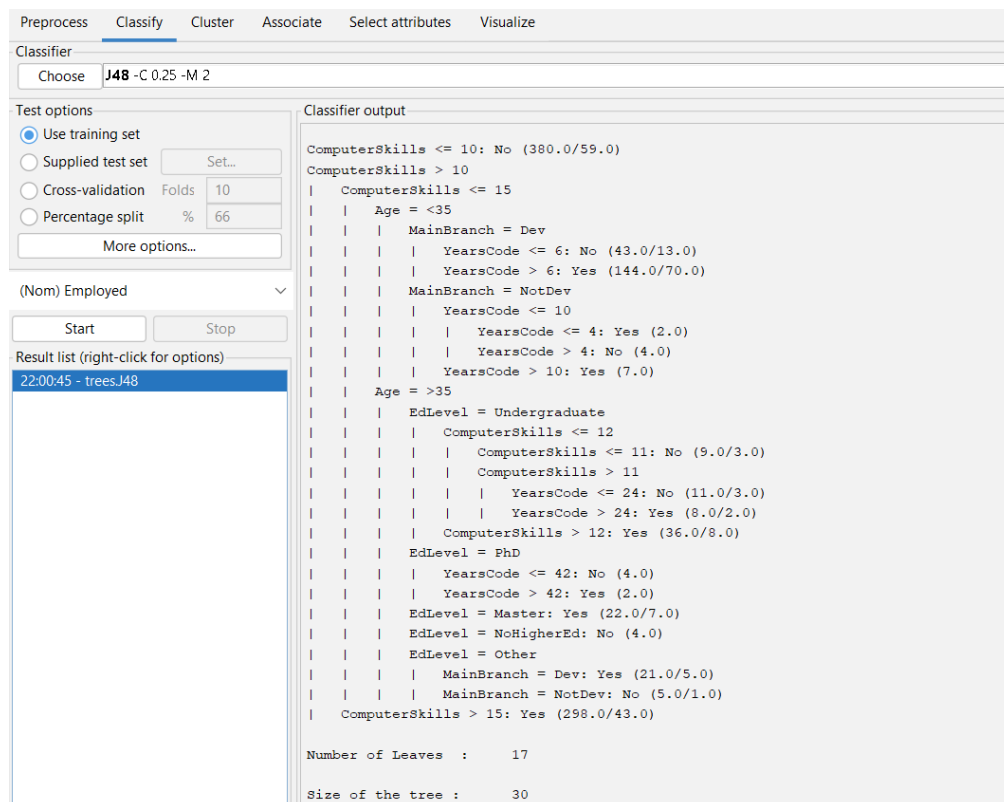


Hình 3. 5. Chọn các thuộc tính của dữ liệu sau khi tải lên

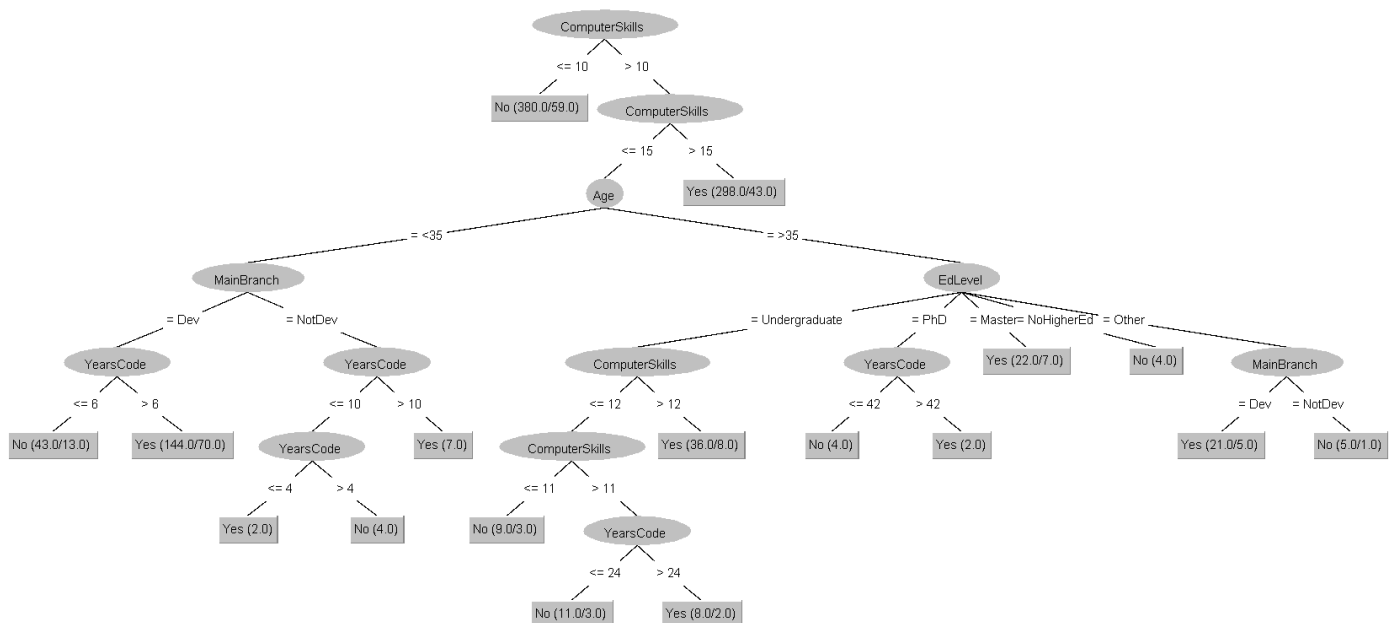
Sau đó vào tính năng Classify và chọn thuật toán trong mục trees.



Hình 3. 7. Lựa chọn thuật toán cây quyết định



Hình 3. 8. Kết quả trả về sau khi chạy thuật toán C4.5



Hình 3. 9. Mô hình cây quyết định sử dụng thuật toán C4.5 trên Weka

3.4.2. Python

Thuật toán sử dụng được tham khảo [tại đây](#) (Anny8910, n.d.). Và bài làm được đăng tải trên Github theo đường dẫn [này](#).

```
[1] import numpy as np
import pandas as pd
from sklearn import tree
from IPython.display import Image
import pydotplus
```

```
[2] from google.colab import files
uploaded = files.upload()
```

Chọn tệp Train_data.csv

- Train_data.csv(text/csv) - 32241 bytes, last modified: 14/1/2025 - 100% done

Saving Train_data.csv to Train_data.csv

```
[3] data = pd.read_csv('Train_data.csv')
data.head()
```

	Age	EdLevel	Gender	MainBranch	YearsCodePro	ComputerSkills	Employed
0	<35	Master	Man	Dev	4	4	No
1	<35	Undergraduate	Man	Dev	5	12	Yes
2	<35	Master	Man	Dev	6	7	No
3	<35	Undergraduate	Man	Dev	6	13	No
4	>35	PhD	Man	NotDev	30	2	No

Các bước tiếp theo: [Tạo mã bằng data](#) [Xem các đồ thị được đề xuất](#) [New interactive sheet](#)

Hình 3. 10. Tải thư viện và tệp dữ liệu

```
[4] data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Age                   1000 non-null  object
1   EdLevel               1000 non-null  object
2   Gender                1000 non-null  object
3   MainBranch            1000 non-null  object
4   YearsCodePro          1000 non-null  int64
5   ComputerSkills        1000 non-null  int64
6   Employed              1000 non-null  object
dtypes: int64(2), object(5)
memory usage: 54.8+ KB
```

```
[5] print("Dữ liệu thiếu: ", data.isnull().sum())
print("Dữ liệu trống: ",data.isna().values.any())
```

```
Dữ liệu thiếu: Age      0
EdLevel      0
Gender       0
MainBranch   0
YearsCodePro 0
ComputerSkills 0
Employed     0
dtype: int64
Dữ liệu trống: False
```

Hình 3. 12. In thông tin dữ liệu và kiểm tra dữ liệu

```
[6] from sklearn.preprocessing import LabelEncoder #Import LabelEncoder
label_encoder = LabelEncoder()

for col in ['Age', 'EdLevel', 'Gender', 'MainBranch', 'Employed']: # Loop through the columns you want to encode.
    if col in data.columns:
        data[col] = label_encoder.fit_transform(data[col]) # Apply label encoding to the correct columns.
```


```
#feature variables
x=data.drop(['Employed'], axis=1)
x
```

	Age	EdLevel	Gender	MainBranch	YearsCodePro	ComputerSkills
0	0	0	0	0	4	4
1	0	4	0	0	5	12
2	0	0	0	0	6	7
3	0	4	0	0	6	13
4	1	3	0	1	30	2
...
995	0	4	0	0	2	3
996	0	0	0	0	5	12
997	1	0	0	0	15	25
998	1	3	0	0	15	12
999	0	4	0	1	9	10

1000 rows x 6 columns

Hình 3. 11. Mã hóa các thuộc tính và gán giá trị x

```
[8] y=data.Employed
y
```



	Employed
0	0
1	1
2	0
3	0
4	0
...	...
995	0
996	1
997	1
998	0
999	0

1000 rows × 1 columns

dtype: int64

Hình 3. 13. Gán giá trị y

```
[9] from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=1)
```


```
[10] # Create Decision Tree classifier object
model = DecisionTreeClassifier()

# Train Decision Tree Classifier
model = model.fit(x_train,y_train)


#Predict the response for test dataset
y_pred = model.predict(x_test)
```

Hình 3. 14. Tải thêm thư viện và tạo cây quyết định

```
[11] #Evaluation using Accuracy score
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
print("Accuracy:",metrics.accuracy_score(y_test, y_pred)*100)
```

 Accuracy: 73.0

```
[12] #Evaluation using Confusion matrix
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

 array([[72, 30],
[24, 74]])

Hình 3. 15. Đo độ chính xác của mô hình

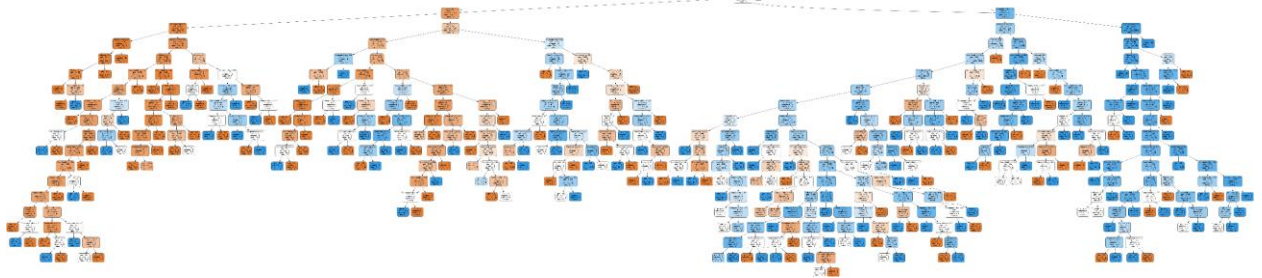

```
[13] #Import modules for Visualizing Decision trees
      from sklearn.tree import export_graphviz
      from io import StringIO
      from IPython.display import Image
      import pydotplus

[15] features=x.columns
      features

↗ Index(['Age', 'EdLevel', 'Gender', 'MainBranch', 'YearsCodePro',
        'ComputerSkills'],
        dtype='object')
```

Hình 3. 16. Tải các thư viện trực quan

```
[16] dot_data = StringIO()
      export_graphviz(model, out_file=dot_data, filled=True, rounded=True, special_characters=True, feature_names = features, class_names=['0', '1'])
      graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
      graph.write_png('recruitment.png')
      Image(graph.create_png())
```



Hình 3. 17. Vẽ cây quyết định

```
[17] # Create Decision Tree classifier object
      model = DecisionTreeClassifier(criterion="entropy", max_depth=3)

      # Train Decision Tree Classifier
      model = model.fit(x_train,y_train)

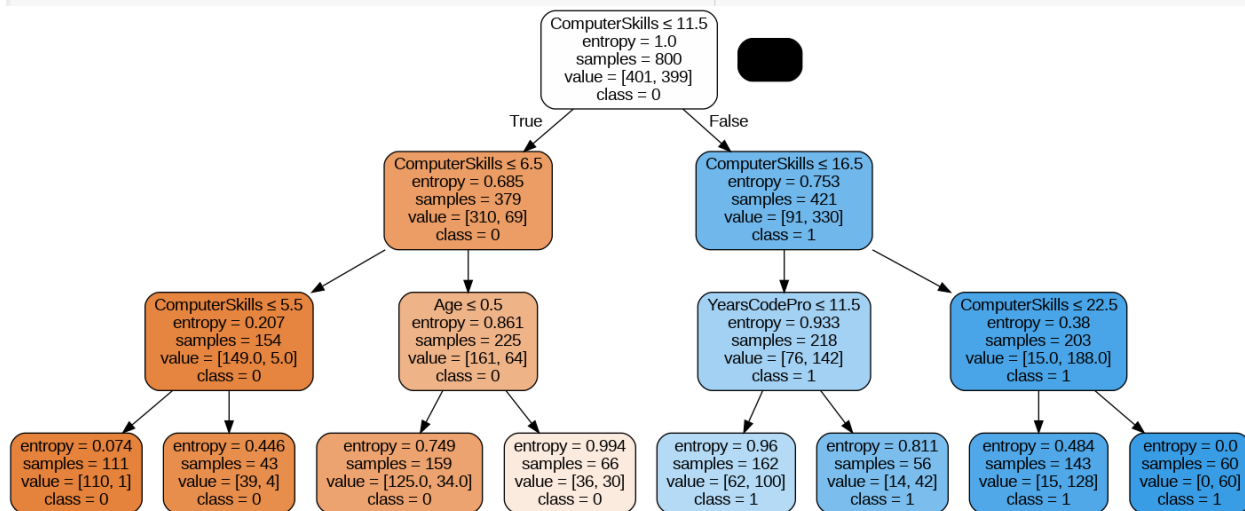
      #Predict the response for test dataset
      y_pred = model.predict(x_test)

      # Model Accuracy
      print("Accuracy:",metrics.accuracy_score(y_test, y_pred)*100)

↗ Accuracy: 81.0
```

Hình 3. 18. Thêm điều kiện cắt tĩa cây quyết định

```
#Better Decision Tree Visualisation
from io import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
dot_data = StringIO()
export_graphviz(model, out_file=dot_data, filled=True, rounded=True, special_characters=True, feature_names = features, class_names=['0', '1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('recruitment.png')
Image(graph.create_png())
```



Hình 3. 19. Cây quyết định sau khi cắt tỉa

DANH MỤC TÀI LIỆU THAM KHẢO

- 70k+ Job Applicants Data (Human Resource). (n.d.). Retrieved January 15, 2025, from <https://www.kaggle.com/datasets/ayushtankha/70k-job-applicants-data-human-resource/code>
- Đặng Văn Nam, Nguyễn Thị Phương Bắc, & Nguyễn Thị Hải Yến. (2018). *Nghiên cứu và ứng dụng cây quyết định trong bài toán tuyển dụng nhân sự*.
- Decision-Tree-Classification-on-Diabetes-Dataset/Diabetes_set_(Decision_tree).ipynb at master · Anny8910/Decision-Tree-Classification-on-Diabetes-Dataset · GitHub*. (n.d.). Retrieved January 15, 2025, from [https://github.com/Anny8910/Decision-Tree-Classification-on-Diabetes-Dataset/blob/master/Diabetes_set_\(Decision_tree\).ipynb](https://github.com/Anny8910/Decision-Tree-Classification-on-Diabetes-Dataset/blob/master/Diabetes_set_(Decision_tree).ipynb)
- Loh, W.-Y. (2015). *A Brief History of Classification and Regression Trees*. www.stat.wisc.edu/
- Trần Minh Quang. (2020). *Khai phá dữ liệu và kỹ thuật phân lớp*.