ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT

KHOA TOÁN KINH TẾ

BÀI TIỂU LUẬN CHƯƠNG IV

MÔN HỌC: THỐNG KÊ TRONG QUẢN TRỊ KINH DOANH VÀ MARKETING

Giảng viên hướng dẫn: Nguyễn Đình Uông
Sinh viên thực hiện: Vũ Thị Hạnh Dung MSSV: K204131871

Câu 1:
Tính toán chi tiết thuật toán BẰNG TAY và trình bày chi tiết các bước thực hiện để tạo ra cây quyết định (sử dụng ID3 hoặc CART Aglorithm) (5 điểm).

Trong ID3, chúng ta cần xác định thứ tự của thuộc tính cần được xem xét tại mỗi bước. Với các bài toán có nhiều thuộc tính và mỗi thuộc tính có nhiều giá trị khác nhau, việc tìm được nghiệm tối ưu thường là không khả thi. Thay vào đó, một phương pháp đơn giản thường được sử dụng là tại mỗi bước, một thuộc tính tốt nhất sẽ được chọn ra dựa trên một tiêu chuẩn nào đó (chúng ta sẽ bàn sớm). Với mỗi thuộc tính được chọn, ta chia dữ liệu vào các child node tương ứng với các giá trị của thuộc tính đó rồi tiếp tục áp dụng phương pháp này cho mỗi child node. Việc chọn ra thuộc tính tốt nhất ở mỗi bước như thế này được gọi là cách chọn greedy (tham lam). Cách chọn này có thể không phải là tối ưu, nhưng trực giác cho chúng ta thấy rằng cách làm này sẽ gần với cách làm tối ưu. Ngoài ra, cách làm này khiến cho bài toán cần giải quyết trở nên đơn giản hơn.

Trong ID3, tổng có trọng số của entropy tại các leaf-node sau khi xây dựng decision tree được coi là hàm mất mát của decision tree đó. Các trọng số ở đây tỉ lệ với số điểm dữ liệu được phân vào mỗi node. Công việc của ID3 là tìm các cách phân chia hợp lý (thứ tự chọn thuộc tính hợp lý) sao cho hàm mất mát cuối cùng đạt giá trị càng nhỏ càng tốt. Như đã đề cập, việc này đạt được bằng cách chọn ra thuộc tính sao cho nếu dùng thuộc tính đó để phân chia, entropy tại mỗi bước giảm đi một lượng lớn nhất. Bài toán xây dựng một decision tree bằng ID3 có thể chia thành các bài toán nhỏ, trong mỗi bài toán, ta chỉ cần chọn ra thuộc tính giúp cho việc phân chia đạt kết quả tốt nhất. Mỗi bài toán nhỏ này tương ứng với việc phân chia dữ liệu trong một non-leaf node.

TÍNH TOÁN CHI TIẾT THUẬT TOÁN

Tổng quan dữ liệu: Ban đầu ta chia các bộ dữ liệu
∗ Biến “Age”:
+ Số tháng < 55
+ 55 ≤ Số tháng < 111
+ Số tháng ≥ 111
∗ Biến “Number”:
+ Đốt sống liên quan < 3,5
+ 3,5 ≤ Đốt sống liên quan < 6,5
+ Đốt sống liên quan ≥ 6,5
∗ Biến “Start”:
+ Đốt sống đầu tiên được phẫu thuật < 8,5
+ 8,5 ≤ Đốt sống đầu tiên được phẫu thuật < 14,5
+ Đốt sống đầu tiên được phẫu thuật ≥ 14,5
Thuật toán ID3
∗ Với bộ dữ liệu Kyphosis:
P(absent) = 64/81 = 79,01% ;
P(present) = 17/81 = 20,99%

Dữ liệu sơ bộ

		Absent	Present
Age	X1< 55	26	3
	55 ≤ X1< 111	13	6

	X1 ≥ 111	25	8
Number	X2< 3,5	31	4
	3,5 ≤ X2< 6,5	30	9
	X2 ≥ 6,5	3	4
Start	X3< 8,5	8	11
	8,5 ≤ X3 < 14,5	27	6
	X3≥ 14,5	29	0

I. Bước 1: Deep 0

Lựa chọn 1: Biến Age

Tính Entropy (H):

$$H = -\sum_{k=1}^{m} p_k log_2(p_k)$$

Với

- $H_{age;age<35} = -\left(\frac{3}{29}\right) log_2\left(\frac{3}{29}\right) - \left(\frac{26}{29}\right) log_2\left(\frac{26}{29}\right) = 0.48$
- $H_{age;35\leq age<111} = -\left(\frac{6}{19}\right) log_2\left(\frac{6}{19}\right) - \left(\frac{13}{19}\right) log_2\left(\frac{13}{19}\right) = 0.9$
- $H_{age;age\geq 111} = -\left(\frac{8}{33}\right) log_2\left(\frac{8}{33}\right) - \left(\frac{25}{33}\right) log_2\left(\frac{25}{33}\right) = 0.8$
- $\boldsymbol{I_{age}} = P_{age;age<35} \times H_{age;age<35} + P_{age;35\leq age<111} \times H_{age;35\leq age<111} + P_{age;age\geq 111} \times H_{age;age\geq 111}$

$$=> \boldsymbol{I_{number}} = \frac{29}{81} \times 0.48 + \frac{19}{81} \times 0.9 + \frac{33}{81} \times 0.8 = 0.709$$

- $\boldsymbol{I_{age;no\ partion}} = -\left(\frac{64}{81}\right) log_2\left(\frac{64}{81}\right) - \left(\frac{17}{81}\right) log_2\left(\frac{17}{81}\right) = 0.741$
- $\boldsymbol{IG_{age}} = \boldsymbol{I_{age;no\ partion}} - \boldsymbol{I_{age}} = \mathbf{0.731 - 0.709 = 0.032}$

Lựa chọn 2 : Biến Number

- Tính Entropy (H):
- $H_{number\leq 3.5} = -\left(\frac{4}{35}\right) log_2\left(\frac{4}{35}\right) - \left(\frac{31}{35}\right) log_2\left(\frac{31}{35}\right) = 0.513$
- $H_{number;3.5\leq number<6.5} = -\left(\frac{9}{39}\right) log_2\left(\frac{9}{39}\right) - \left(\frac{30}{39}\right) log_2\left(\frac{30}{39}\right) = 0.78$
- $H_{numbe;number\geq 6.5} = -\left(\frac{4}{7}\right) log_2\left(\frac{3}{7}\right) - \left(\frac{4}{7}\right) log_2\left(\frac{3}{7}\right) = 0.8$
- $\boldsymbol{I_{number}} = P_{number;number<3.5} \times H_{numbe;number<3.5} + P_{number;3.5\leq number<6.5} \times H_{number;3.5\leq number<6.5} + P_{number;number\geq 6.5} \times H_{number;number\geq 6.5}$

$$=> \boldsymbol{I_{number}} = \frac{35}{81} \times 0.513 + \frac{39}{81} \times 0.78 + \frac{7}{81} \times 0.8 = 0.682$$

- $\boldsymbol{I_{number;no\ partion}} = -\left(\frac{64}{81}\right) log_2\left(\frac{64}{81}\right) - \left(\frac{17}{81}\right) log_2\left(\frac{17}{81}\right) = 0.741$

- $\boldsymbol{IG_{number}} = \boldsymbol{I_{number;no\ partion}} - \boldsymbol{I_{number}} = \mathbf{0.059}$

Lựa chọn 3 : Biến Start

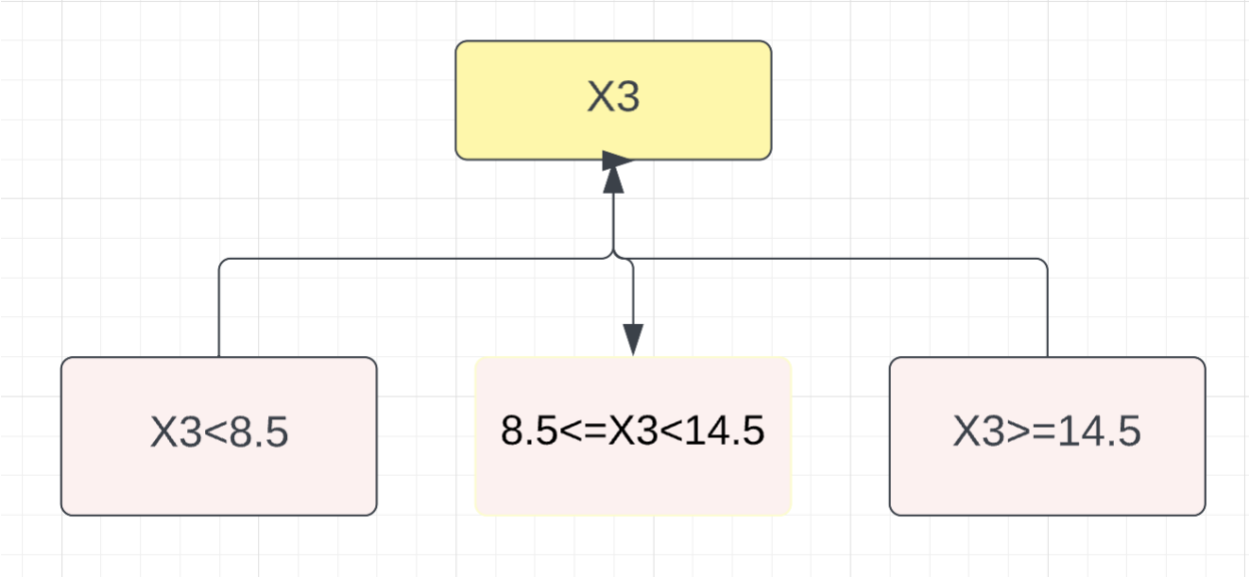
- Tính Entropy (H):
- $H_{start\ \leq 8.5} = -\left(\frac{11}{19}\right) log_2\left(\frac{11}{19}\right) - \left(\frac{8}{19}\right) log_2\left(\frac{8}{19}\right) = 0.982$
- $H_{start\ ;8.5 \leq start\ <14.5} = -\left(\frac{6}{33}\right) log_2\left(\frac{6}{33}\right) - \left(\frac{27}{33}\right) log_2\left(\frac{27}{33}\right) = 0.684$
- $H_{start\ ;start\ \geq 14.5} = 0$
- $\boldsymbol{I_{start}} = P_{start\ ;start\ <3.5} \times H_{start\ ;start\ <3.5} + P_{start\ ;3.5 \leq start\ <6.5} \times H_{start\ ;3.5 \leq start\ <6.5} + P_{start\ ;start\ \geq 6.5} \times H_{start\ ;start\ \geq 6.5}$

$$=> \boldsymbol{I_{start}} = \frac{19}{81} \times 0.982 + \frac{33}{81} \times 0.684 + \frac{29}{81} \times 0 = 0.682$$

- $\boldsymbol{I_{start\ ;no\ partion}} = -\left(\frac{64}{81}\right) log_2\left(\frac{64}{81}\right) - \left(\frac{17}{81}\right) log_2\left(\frac{17}{81}\right) = 0.741$
- $\boldsymbol{IG_{start}} = \boldsymbol{I_{start\ ;no\ partion}} - \boldsymbol{I_{start}} = \mathbf{0.232}$
- Tổng hợp ta có bảng Information Gain

Attribute	**Information Gain**
Age	0,032
Number	0,059
Start	0,232

- **Vì $\boldsymbol{IG_{start}}$ lớn nhất nên ta chọn "Start" làm biến gốc cho decision tree**



II. Bước 2: Deep 1

– "Start:< 8,5

		Absent	Present	Total
Age	X1< 55	3	3	6
	55 ≤ X1< 111	3	2	5
	X1 ≥ 111	2	6	8
Number	X2< 3,5	3	1	4
	3,5 ≤ X2< 6,5	4	6	10
	X2 ≥ 6,5	1	4	5

* "Start:< 8,5 ;Age

- $H_{start<8.5;age<55} = -\left(\frac{3}{6}\right) log_2\left(\frac{3}{6}\right) - \left(\frac{3}{6}\right) log_2\left(\frac{3}{6}\right) = 1$
- $H_{start<8.5;55\leq age<111} = -\left(\frac{3}{5}\right) log_2\left(\frac{3}{5}\right) - \left(\frac{2}{5}\right) log_2\left(\frac{2}{5}\right) = 0.971$
- $H_{start<8.5;age\geq 111} = -\left(\frac{2}{8}\right) log_2\left(\frac{2}{8}\right) - \left(\frac{6}{8}\right) log_2\left(\frac{6}{8}\right) = 0.811$
-
- $\boldsymbol{I}_{start<8.5;age} = P_{start<8.5;age<55} \times H_{start<8.5;age<55} + P_{start<8.5;35\leq age<111} \times H_{start<8.5;35\leq age<111} + P_{start<8.5;age\geq 111} \times H_{start<8.5;age\geq 111}$

$$=> \boldsymbol{I}_{start<8.5;age} = \frac{6}{19} \times 1 + \frac{5}{19} \times 0.971 + \frac{8}{19} \times 0.811 = 0.9127$$

- $\boldsymbol{I}_{start<8.5;age\ ;\boldsymbol{no\ partion}} = -\left(\frac{11}{19}\right) log_2\left(\frac{11}{19}\right) - \left(\frac{8}{19}\right) log_2\left(\frac{8}{19}\right) = 0.982$
- $\boldsymbol{IG}_{start<8.5;age} = \boldsymbol{I}_{start\ ;\boldsymbol{no\ partion}} - \boldsymbol{I}_{start} = \mathbf{0.982 - 0.9127 = 0.127}$

* "Start:< 8,5 ;Number

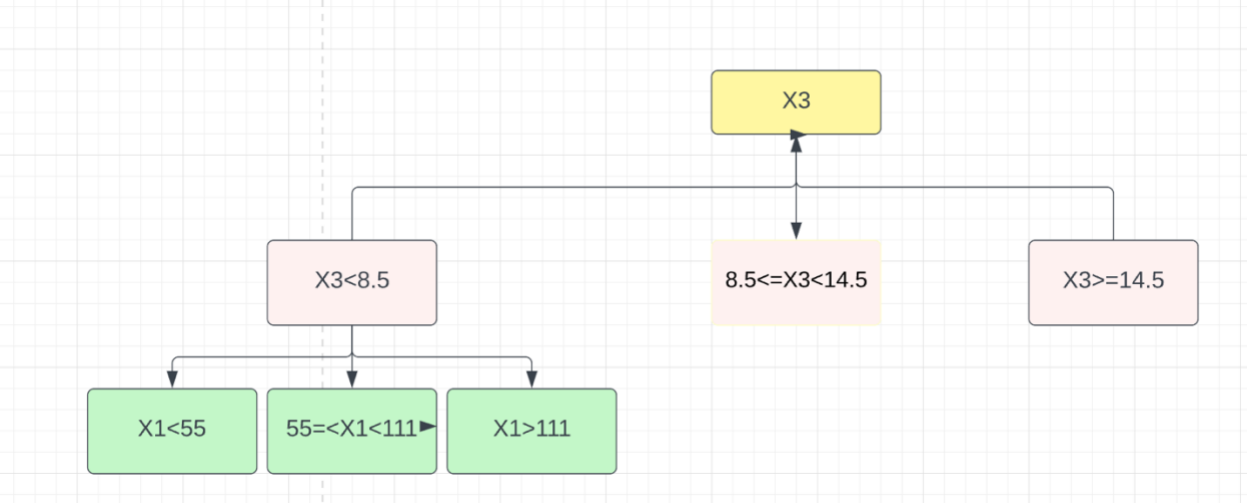
- $H_{start<8.5;number\leq 3.5} = -\left(\frac{3}{4}\right) log_2\left(\frac{3}{4}\right) - \left(\frac{1}{4}\right) log_2\left(\frac{1}{4}\right) = 0.811$
- $H_{start<8.5;3.5\leq number<6.5} = -\left(\frac{6}{10}\right) log_2\left(\frac{6}{10}\right) - \left(\frac{4}{10}\right) log_2\left(\frac{4}{10}\right) = 0.971$
- $H_{start<8.5;\ number\geq 6.5} = -\left(\frac{4}{5}\right) log_2\left(\frac{4}{5}\right) - \left(\frac{1}{5}\right) log_2\left(\frac{1}{5}\right) = 0.722$

$$\boldsymbol{I}_{start;number} = \frac{4}{19} \times 0.811 + \frac{10}{19} \times 0.971 + \frac{5}{19} \times 0.722 = 0.87179$$

- $\boldsymbol{I}_{start<8.5;number\ ;\boldsymbol{no\ partion}} = -\left(\frac{11}{19}\right) log_2\left(\frac{11}{19}\right) - \left(\frac{8}{19}\right) log_2\left(\frac{8}{19}\right) = 0.98194$
- $\boldsymbol{IG}_{start<8.5;number} = \boldsymbol{I}_{start<8.5;number\ ;\boldsymbol{no\ partion}} - \boldsymbol{I}_{start<8.5;number} = 0.11015$

Attribute	**Information Gain**
Age	0,127
Number	0,110

- **Vì $\boldsymbol{IG}_{start<8.5;age}$ lớn nhất nên ta chọn "AGE" làm** Decision Node **tiếp theo cho decision tree**



– "Start:< 8,5

		Absent	Present	Total
Age	X1< 55	12	0	12
	55 ≤ X1< 111	3	4	7
	X1 ≥ 111	12	2	14
Number	X2< 3,5	9	3	12
	3,5 ≤ X2< 6,5	16	3	19
	X2 ≥ 6,5	2	0	2

"8,5 ≤ Start < 14,5 ; Age"

- $H_{8.5\leq start<14.5;age<55} = 0$
- $H_{8.5\leq start<14.5;55\leq age<111} = -\left(\frac{4}{7}\right) log_2\left(\frac{4}{7}\right) - \left(\frac{3}{7}\right) log_2\left(\frac{3}{7}\right) = 0.985$
- $H_{8.5\leq start<14.5;age\geq 111} = -\left(\frac{2}{14}\right) log_2\left(\frac{2}{14}\right) - \left(\frac{12}{14}\right) log_2\left(\frac{12}{14}\right) = 0.592$
- $\boldsymbol{I}_{8.5\leq start<14.5;age} = \frac{12}{33} \times 0 + \frac{7}{33} \times 0.985 + \frac{14}{33} \times 0.592 = 0.46$

- $\boldsymbol{I}_{8.5\leq start<14.5;age\ ;\boldsymbol{no\ partion}} = -\left(\frac{6}{33}\right) log_2\left(\frac{6}{33}\right) - \left(\frac{27}{33}\right) log_2\left(\frac{27}{33}\right) = 0.684$
- $\boldsymbol{IG}_{8.5\leq start<14.5;age} = \boldsymbol{I}_{start\ ;\boldsymbol{no\ partion}} - \boldsymbol{I}_{start} = \mathbf{0.224}$
-

* "Start: $8.5 \leq X1 < 14.5$;Number

- $H_{8.5\leq start<14.5;number\leq 3.5} = -\left(\frac{3}{12}\right) log_2\left(\frac{3}{12}\right) - \left(\frac{9}{12}\right) log_2\left(\frac{9}{12}\right) = 0.8113$
- $H_{8.5\leq start<14.5;3.5\leq number<6.5} = -\left(\frac{3}{19}\right) log_2\left(\frac{3}{19}\right) - \left(\frac{16}{19}\right) log_2\left(\frac{16}{19}\right) = 0.6292$
- $H_{8.5\leq start<14.5;\ number\geq 6.5} = 0$

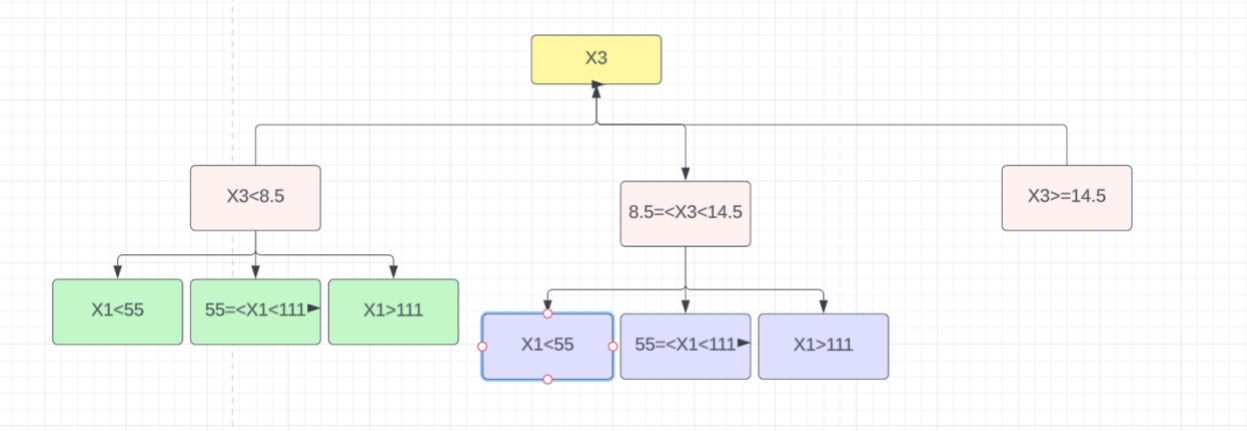
$$\boldsymbol{I}_{8.5\leq start<14.5;number} = \frac{12}{33} \times 0.8113 + \frac{19}{33} \times 0.6292 + 0 = 0.6573$$

- $I_{8.5\leq start<14.5;number;\textbf{no partion}} = -\left(\frac{6}{33}\right)log_2\left(\frac{6}{33}\right) - \left(\frac{27}{33}\right)log_2\left(\frac{27}{33}\right) = 0.684$
- $IG_{8.5\leq start<14.5;number} = I_{8.5\leq start<14.5;number;\textbf{no partion}} - I_{8.5\leq start<14.5;number} = 0.0267$

Attribute	**Information Gain**
Age	0,224
Number	0,0267

- **Vì $IG_{8.5\leq start<14.5;age}$ lớn nhất nên ta chọn "AGE" làm** Decision Node **tiếp theo cho decision tree**

- **Vì "Start ≥ 14, 5" :** ta nhận thấy rằng đều cho ra kết quả là "Absent". Vậy với nhánh "Start ≥ 14,5" sẽ cho ra kết quả cuối cùng là "Absent" và sẽ ngưng tính toán.
-



III. DECISION NODE 2

Number	X2< 3,5	13	0	13
	3,5 ≤ X2< 6,5	11	1	12
	X2 ≥ 6,5	2	2	4

– Age < 55

- $H_{age<55;number\leq 3.5} = 0$
- $H_{age<55;3.5\leq number<6.5} = -\left(\frac{1}{12}\right)log_2\left(\frac{1}{12}\right) - \left(\frac{11}{12}\right)log_2\left(\frac{11}{12}\right) = 0.4138$
- $H_{age<55;number\geq 6.5} = -\left(\frac{2}{4}\right)log_2\left(\frac{2}{4}\right) - \left(\frac{2}{4}\right)log_2\left(\frac{2}{4}\right) = 1$

$$=> I_{age<55;\,number} = \frac{13}{29}\times 0 + \frac{12}{29}\times 0.4183 + \frac{4}{29}\times 1 = 0.311$$

- $\boldsymbol{I}_{55\leq age<111;\boldsymbol{no\ partion}} = -\left(\frac{26}{29}\right) log_2\left(\frac{26}{29}\right) - \left(\frac{3}{29}\right) log_2\left(\frac{3}{29}\right) = 0.4798$
- $\boldsymbol{IG}_{55\leq age<111\ ;\boldsymbol{number}} = \boldsymbol{I}_{55\leq age<111;number\ ;\boldsymbol{no\ partion}} - \boldsymbol{I}_{55\leq age<111} = \mathbf{0.4798} - \mathbf{0.311} = \mathbf{0.168}$

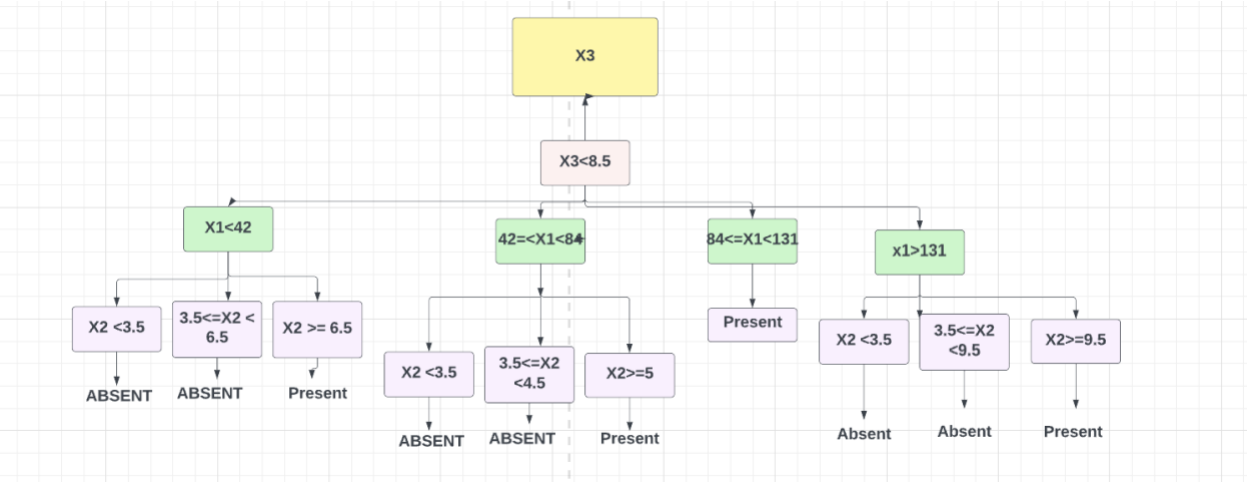
Number	X2< 3,5	6	1	7
	3,5 ≤ X2< 6,5	7	5	12
	X2 ≥ 6,5	0	0	0

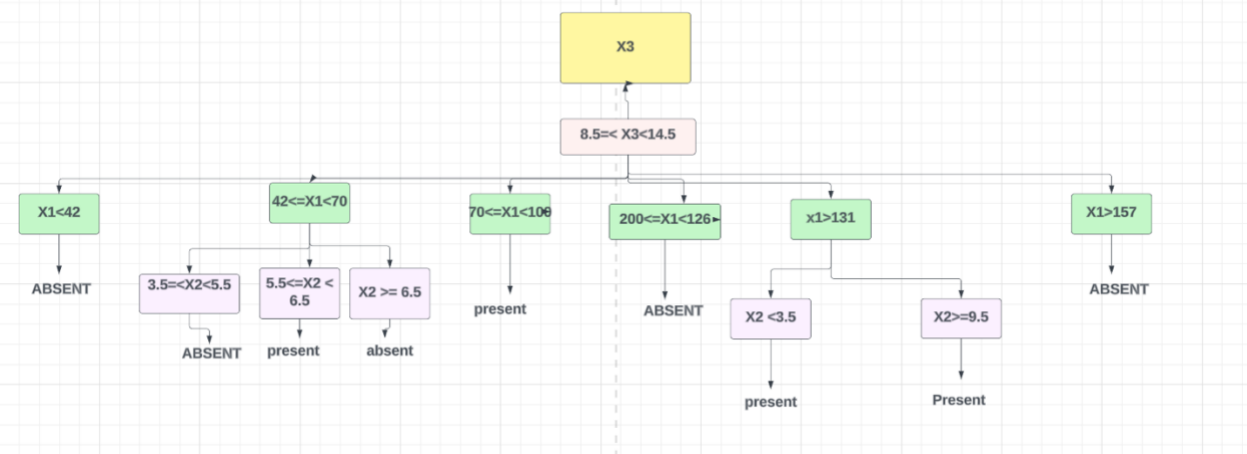
- Age : $55 \leq age < 111$

- $H_{55\leq age<111;number\leq 3.5} = -\left(\frac{6}{7}\right) log_2\left(\frac{6}{7}\right) - \left(\frac{1}{7}\right) log_2\left(\frac{1}{7}\right) = 0.5916$
- $H_{55\leq age<111;3.5\leq number<6.5} = -\left(\frac{7}{12}\right) log_2\left(\frac{7}{12}\right) - \left(\frac{5}{12}\right) log_2\left(\frac{5}{12}\right) = 0.9798$

$$=> \boldsymbol{I}_{\boldsymbol{age<55;\ number}} = \frac{7}{19} \times 0.5916 + \frac{12}{19} \times 0.9798 = 0.8367$$

- $\boldsymbol{I}_{age;\boldsymbol{no\ partion}} = -\left(\frac{13}{19}\right) log_2\left(\frac{13}{19}\right) - \left(\frac{6}{19}\right) log_2\left(\frac{6}{19}\right) = 0.8997$
- $\boldsymbol{IG}_{age} = \boldsymbol{I}_{age;\boldsymbol{no\ partion}} - \boldsymbol{I}_{age} = \mathbf{0.8997} - \mathbf{0.8367} = \mathbf{0.036}$





2.1 So sánh kết quả nhận được ở câu 1 với kết quả Decision Tree

Theo phần trình bày của Kaggle, tác giả chọn thuật toán CART với mức training là 70% và testing 30% , ở đây em chọn sử dụng thuật toán ID3 để đưa ra cây quyết định.
Theo tác giả chọn theo Cart, lấy biến " Start" làm biến gốc cho Decision tree và sau khi tính toán bằng thuật toán ID3, em cũng quyết định chọn biến "Start" là biến gốc vì có IG lớn nhất, phù hợp Về kết quả thì cả Python và tính tay đều đưa ra kết quả tương đồng, chênh lệch một vài khoảng giá trị tại biến Number nhưng không quá lớn và không ảnh hưởng đến cây quyết định.

2.2 Tiến hành code toàn bộ thuật toán Decision Tree, thực hiện training và test data, giải thích kết quả

Bước 1: Tiến hành chia dữ liệu thành hai phần theo tỷ lệ 70:30. Training 70% và testing 30%

```
X = df.drop('Kyphosis',axis = 1)
y = df['Kyphosis']
```

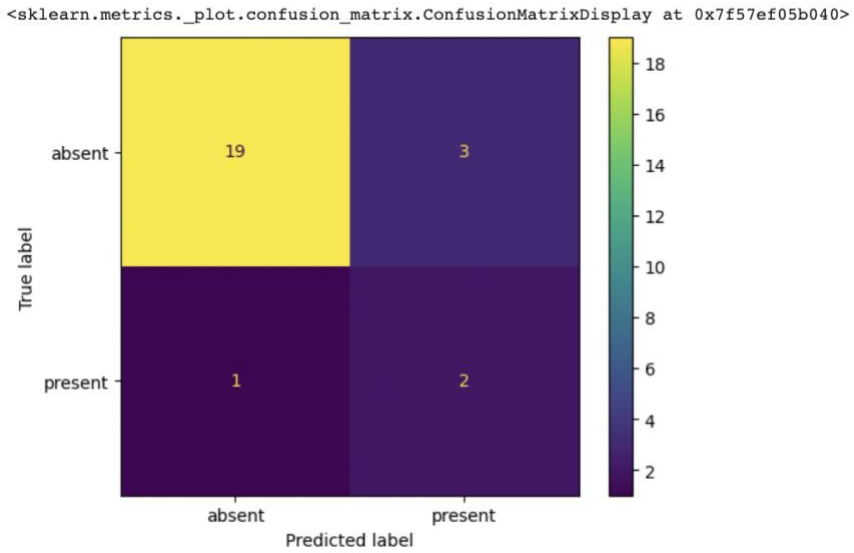
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

Bước 2: Đánh giá mô hình

	precision	recall	f1-score	support
absent	0.95	0.86	0.90	22
present	0.40	0.67	0.50	3
accuracy			0.84	25
macro avg	0.68	0.77	0.70	25
weighted avg	0.88	0.84	0.86	25

⇨ Mô hình có độ chính xác phân loại (Accuracy) bằng 84%, mô hình có độ chính xác khá cao . Ta tiếp tục vẽ Confusion Matrix

	precision	recall	f1-score	support
absent	0.95	0.86	0.90	22
present	0.40	0.67	0.50	3
accuracy			0.84	25
macro avg	0.68	0.77	0.70	25
weighted avg	0.88	0.84	0.86	25



Dựa vào biểu đồ nhiệt của Confusion Matrix, có thể thấy

- **True absent:** Số lượng dự đoán chính xác một người không bị gù cột sống là 19
- **True present:** Số lương dự đoán chính xác một người bị gù cột sống là 1
- **Fale absent:** Số lượng các dự đoán sai lệch là 3. Là khi mô hình dự đoán một người không bị gù cột sống nhưng người đó bị gù cột sống.
- **False present:** Số lượng các dự đoán sai lệch một cách gián tiếp là 2. Là khi mô hình dự đoán một người bị gù cột sống nhưng người đó không bị gù cột sống.

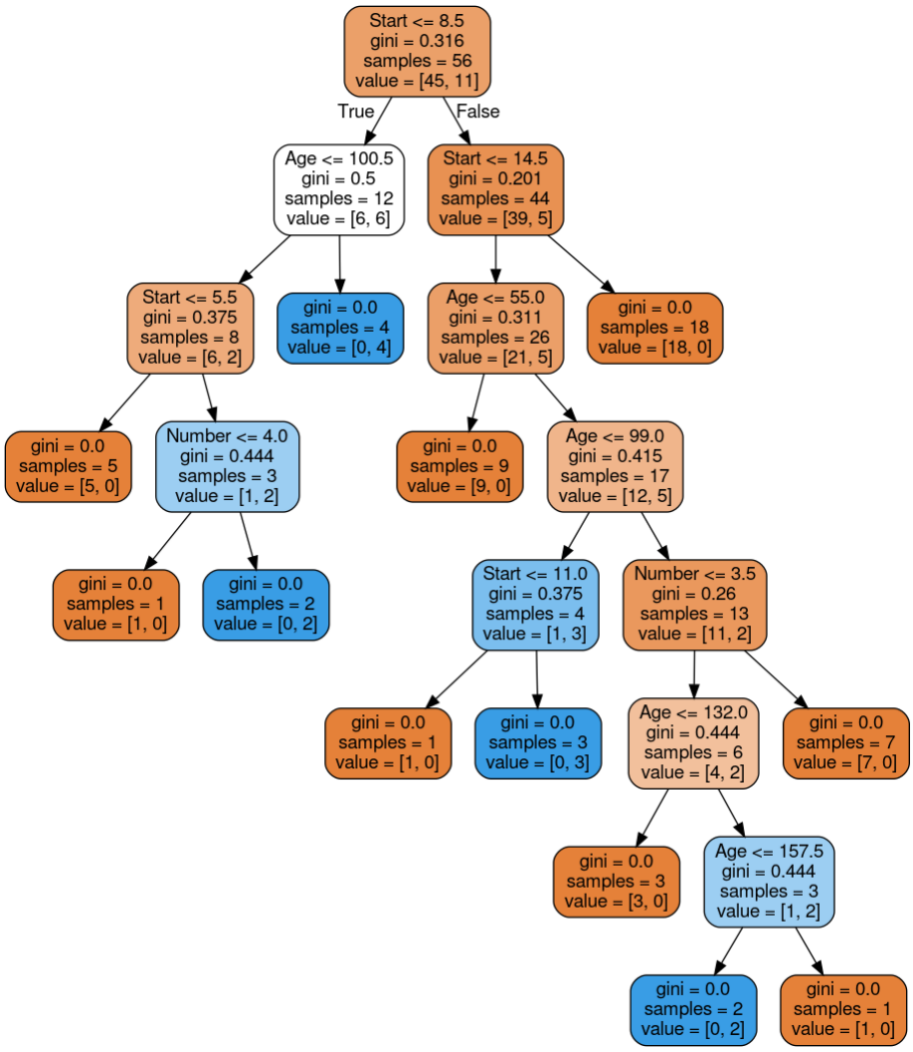
Bước 3: Vẽ Decision Tree

```
import os
os.environ["PATH"] += os.pathsep + 'C:/Program Files (x86)/Graphviz2.38/bin/'
```

```
dot_data = StringIO()
export_graphviz(dtree, out_file=dot_data,feature_names=features,filled=True,rounded=True)

graph = pydot.graph_from_dot_data(dot_data.getvalue())

Image(graph[0].create_png())
```



Câu 3: Random Forest

Random Forest (Rừng ngẫu nhiên) là một thuật toán học máy dựa trên việc kết hợp nhiều cây quyết định (Decision Trees) để tạo ra một dự đoán chính xác và ổn định hơn. Điều đặc biệt của Random Forest là nó sử dụng kỹ thuật "bagging" (bootstrap aggregating) và "random feature selection" để xây dựng các cây quyết định độc lập. Tuy nhiên mỗi cây quyết định sẽ khác nhau (có yếu tố random). Sau đó kết quả dự đoán được tổng hợp từ các cây quyết định. Ở bước huấn luyện thì mình sẽ xây dựng nhiều cây quyết định, các cây quyết định có thể khác nhau (phần sau mình sẽ nói mỗi cây được xây dựng như thế nào). Sau đó ở bước dự đoán, với một dữ liệu mới, thì ở mỗi cây quyết định mình sẽ đi từ trên xuống theo các node điều kiện để được các dự đoán, sau đó kết quả cuối cùng được tổng hợp từ kết quả của các cây quyết định.

-Xây dựng thuật toán Random Forest

Lấy ngẫu nhiên n dữ liệu từ bộ dữ liệu với kĩ thuật Bootstrapping, hay còn gọi là random sampling with replacement. Tức khi mình sample được 1 dữ liệu thì mình không bỏ dữ liệu đấy ra mà vẫn giữ lại trong tập dữ liệu ban đầu, rồi tiếp tục sample cho tới khi sample đủ n dữ liệu. Khi dùng kĩ thuật này thì tập dữ liệu mới của mình có thể có những dữ liệu bị trùng nhau.

Sau khi sample được n dữ liệu từ bước 1 thì mình chọn ngẫu nhiên ở k thuộc tính (k < n). Giờ mình được bộ dữ liệu mới gồm n dữ liệu và mỗi dữ liệu có k thuộc tính.
Dùng thuật toán Decision Tree để xây dựng cây quyết định với bộ dữ liệu ở bước 2.

Bước 1: Phân chia dữ liệu thành hai phần: features (X) gồm có các biến "Age", "Number", "Start" và target (y) với biến "Kyphosis". Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra để đánh giá hiệu suất của mô hình.

Bước 2:

- Import thư viện RandomForestClassifier từ sklearn.ensemble.
- Khởi tạo một đối tượng Random Forest với các tham số tuỳ chọn như số lượng cây (n_estimators), độ sâu tối đa của cây (max_depth), và số lượng đặc trưng được chọn ngẫu nhiên (max_features).
- Sử dụng phương thức fit() trên mô hình để huấn luyện với tập huấn luyện (X_train, y_train).

```
from sklearn.ensemble import RandomForestClassifier

rf_clf_model = RandomForestClassifier()
```

```
rfc = RandomForestClassifier(n_estimators=200)
```

```
rfc.fit(X_train,y_train)
```

Bước 3: Đánh giá mô hình

```
print(confusion_matrix(y_test,rfc_pred))
print('\n')
print(classification_report(y_test,rfc_pred))
```

```
[[20  2]
 [ 1  2]]


              precision    recall  f1-score   support

      absent       0.95      0.91      0.93        22
     present       0.50      0.67      0.57         3

    accuracy                           0.88        25
   macro avg       0.73      0.79      0.75        25
weighted avg       0.90      0.88      0.89        25
```

⇨ Mô hình có độ chính xác phân loại (Accuracy) bằng 88%, mô hình có độ chính xác cao hơn mô hình cây quyết định .

```
#  acc for decision Tree
dtree = round(dtree.score(X,y)*100,2)
print(dtree)
```

```
95.06
```

```
rfc = round(rfc.score(X,y)*100,2)
print(rfc)
```

```
96.3
```

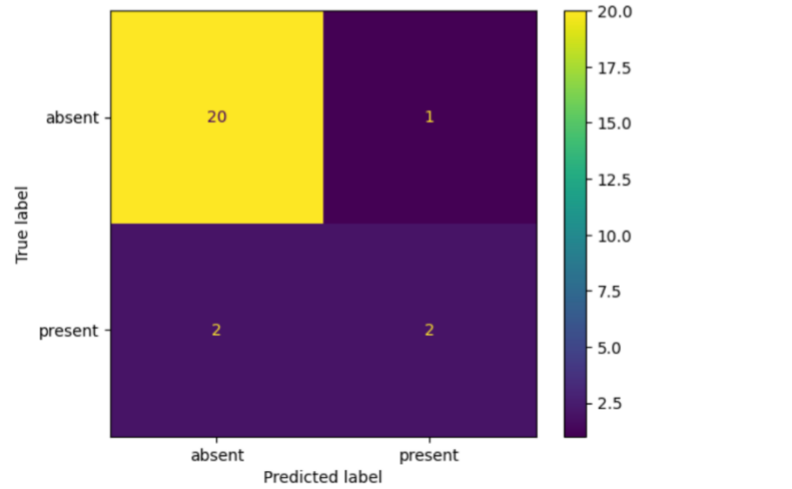
Chỉ số đánh giá cũng cho thấy mô hình Random Forest cũng cao hơn Decision Tree

```
              precision    recall  f1-score   support

      absent       0.91      0.95      0.93        21
     present       0.67      0.50      0.57         4

    accuracy                           0.88        25
   macro avg       0.79      0.73      0.75        25
weighted avg       0.87      0.88      0.87        25
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f582b945e70>
```



Kết luận: Với bài toán Kyphosis, mô hình **random forest** có thể hiệu quả hơn **decision tree** . Vì random forest kết hợp nhiều cây quyết định, nó có khả năng giảm overfitting và cung cấp dự đoán chính xác hơn trên tập kiểm tra. Tuy nhiên, để đưa ra đánh giá chính xác hơn về hiệu suất của hai mô hình trong bài toán cụ thể này, cần xem xét các yếu tố khác như kích thước dữ liệu, đặc điểm của thuộc tính và điều chỉnh tham số của các mô hình.