

# **Trường Đại học Công Nghiệp Hà Nội**

## **Khoa Công Nghệ Thông Tin**



### **Báo Cáo Bài Tập Lớn**

### **Học Phần Trí Tuệ Nhân Tạo**

#### **ĐỀ TÀI**

## **“TÌM HIỂU CÂY QUYẾT ĐỊNH VÀ ỨNG DỤNG TRONG BÀI TOÁN ĐÁNH GIÁ CHẤT LƯỢNG XE Ô TÔ”**

**Giảng viên:** TS. Trần Hùng Cường.

**Nhóm:** 3 - **Lớp:** 20211IT6043003 - K14.

**Sinh viên thực hiện:**

- 1. Cao Văn Nhật**
- 2. Trần Huy Cảnh**
- 3. Tạ Văn Hiếu**
- 4. Vũ Văn Khánh**
- 5. Hoàng Văn Mạnh**

Hà Nội, 2021

# MỤC LỤC

<b>CHƯƠNG 1: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO</b>	<b>3</b>
<b>TRÍ TUỆ NHÂN TẠO LÀ GÌ?</b>	<b>3</b>
<b>MỤC ĐÍCH CỦA TRÍ TUỆ NHÂN TẠO</b>	<b>3</b>
<b>MỘT SỐ ỨNG DỤNG TRÍ TUỆ NHÂN TẠO</b>	<b>3</b>
<b>PHÂN BIỆT AI, MACHINE LEARNING, DEEP LEARNING</b>	<b>4</b>
<b>AI (TRÍ TUỆ NHÂN TẠO)</b>	<b>5</b>
<b>MACHINE LEARNING (HỌC MÁY)</b>	<b>6</b>
<b>DEEP LEARNING (HỌC SÂU)</b>	<b>7</b>
<b>CHƯƠNG 2: CÂY QUYẾT ĐỊNH</b>	<b>8</b>
<b>2.1. CÂY QUYẾT ĐỊNH</b>	<b>8</b>
<b>2.2. THUẬT TOÁN TẠO CÂY QUYẾT ĐỊNH</b>	<b>8</b>
<b>2.3. VÍ DỤ MINH HOẠ</b>	<b>12</b>
<b>CHƯƠNG 3: ỨNG DỤNG TRONG BÀI TOÁN ĐÁNH GIÁ CHẤT LƯỢNG XE Ô TÔ</b>	<b>16</b>
<b>3.1. BÀI TOÁN</b>	<b>16</b>
<b>3.2. CƠ SỞ DỮ LIỆU</b>	<b>16</b>
<b>3.3. MỘT SỐ KẾT QUẢ</b>	<b>17</b>
<b>KẾT LUẬN</b>	<b>27</b>
<b>TÀI LIỆU THAM KHẢO</b>	<b>28</b>

# Mở đầu

Ngày nay công nghệ được ứng dụng trong hầu hết các lĩnh vực của đời sống. Bên cạnh những cách làm truyền thống cũng đã xuất hiện những kỹ thuật công nghệ mới được áp dụng và đem lại hiệu quả đáng kể. Với lượng thông tin lớn, những bài toán có độ phức tạp cao vấn đề đặt ra là làm thế nào để phát hiện tri thức, đưa ra lời giải mà thời gian thực hiện có thể chấp nhận được. Một trong số các kỹ thuật được sử dụng đó chính là trí tuệ nhân tạo. Báo cáo này sẽ giúp bạn đọc nắm được những khái niệm cơ bản, những kỹ thuật cũng như triển khai ứng dụng của lĩnh vực này vào giải quyết bài toán đánh giá chất lượng xe ô tô trong thực tế gồm: Chương một đưa ra tổng quan về trí tuệ nhân tạo. Chương hai là khái quát về cây quyết định và ví dụ minh họa với bài toán đánh giá ô tô. Chương cuối cùng là ứng dụng vào bài toán đánh giá ô tô trong thực tế dựa trên những trường dữ liệu có sẵn để hỗ trợ người sử dụng ra quyết định có nên mua hay là không.

# CHƯƠNG 1: TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO

## 1.1. TRÍ TUỆ NHÂN TẠO LÀ GÌ?

- *AI - Artificial Intelligence* hay còn gọi là *Trí tuệ nhân tạo* là một ngành khoa học, kỹ thuật chế tạo máy móc thông minh, đặc biệt là các chương trình máy tính thông minh.
- AI được thực hiện bằng cách nghiên cứu cách suy nghĩ của con người, cách con người học hỏi, quyết định và làm việc trong khi giải quyết một vấn đề nào đó, và sử dụng những kết quả nghiên cứu này như một nền tảng để phát triển các phần mềm và hệ thống thông minh, từ đó áp dụng vào các mục đích khác nhau trong cuộc sống. Nói một cách dễ hiểu thì AI là việc sử dụng, phân tích các dữ liệu đầu vào nhằm đưa ra sự dự đoán rồi đi đến quyết định cuối cùng.

## 1.2. MỤC ĐÍCH CỦA TRÍ TUỆ NHÂN TẠO

- Tạo ra các hệ thống chuyên gia - là các ứng dụng máy tính được phát triển để giải quyết các vấn đề phức tạp trong một lĩnh vực cụ thể, ở mức độ thông minh và chuyên môn của con người.
- Thực hiện trí thông minh của con người trong máy móc - Tạo ra các hệ thống có thể hiểu, suy nghĩ, học hỏi và hành xử như con người.

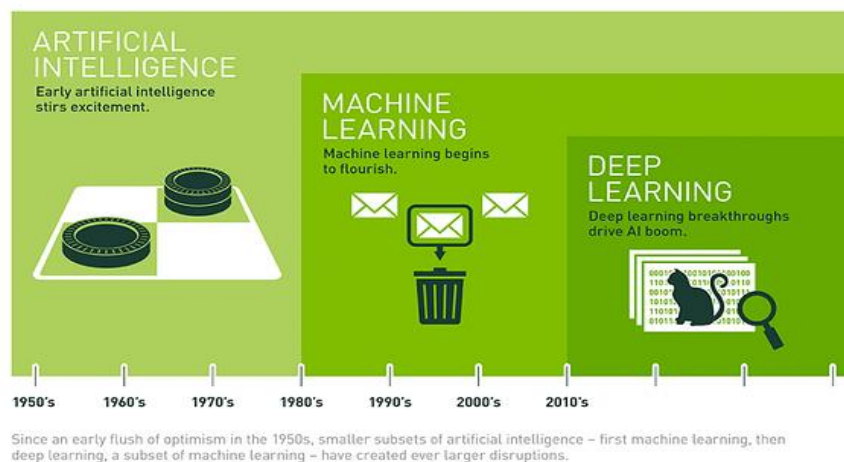
## 1.3. MỘT SỐ ỨNG DỤNG TRÍ TUỆ NHÂN TẠO

- Quản trị: Các hệ thống AI trợ giúp các công việc hành chính hàng ngày, để giảm thiểu lỗi của con người và tối đa hóa hiệu quả.
- Điều trị từ xa: Đối với các tình huống không khẩn cấp, bệnh nhân có thể liên hệ với hệ thống AI của bệnh viện để phân tích các triệu chứng của họ, nhập các dấu hiệu quan trọng của họ và đánh giá xem có cần phải chăm sóc y tế hay không.

Điều này làm giảm khối lượng công việc của các chuyên gia y tế bằng cách chỉ đưa các trường hợp quan trọng đến họ.

- Hỗ trợ chuẩn đoán: Thông qua thị giác máy tính và mạng lưới thần kinh tích chập, AI hiện có khả năng đọc quét hình ảnh cộng hưởng từ để kiểm tra khối u và sự phát triển ác tính khác của nó, với tốc độ nhanh hơn so với các bác sĩ x-quang và sai số thấp hơn đáng kể.
- Phẫu thuật có sự trợ giúp của robot: Robot phẫu thuật có sai số rất nhỏ và có thể thực hiện phẫu thuật suốt ngày đêm mà không bị kiệt sức.
- Giám sát các chỉ số quan trọng. Ngoài ra còn rất nhiều những ứng dụng trong các lĩnh vực khác trong đời sống như nhận diện khuôn mặt, nhận diện giọng nói, ô tô tự lái...

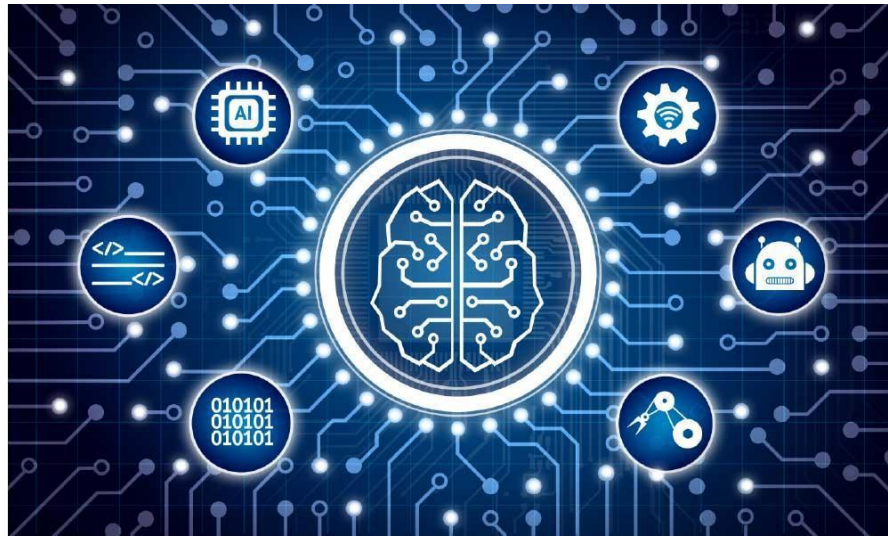
## 1.4. PHÂN BIỆT AI, MACHINE LEARNING, DEEP LEARNING



Hình 1.1: Phân biệt AI, Machine Learning và Deep Learning.

### 1.4.1. AI (TRÍ TUỆ NHÂN TẠO)

Trí tuệ nhân tạo là trí tuệ máy móc được tạo ra bởi con người. Trí tuệ này có thể tư duy, suy nghĩ, học hỏi,... như con người. Xử lý dữ liệu ở mức độ rộng hơn, quy mô hơn, hệ thống, khoa học và nhanh hơn so với con người.

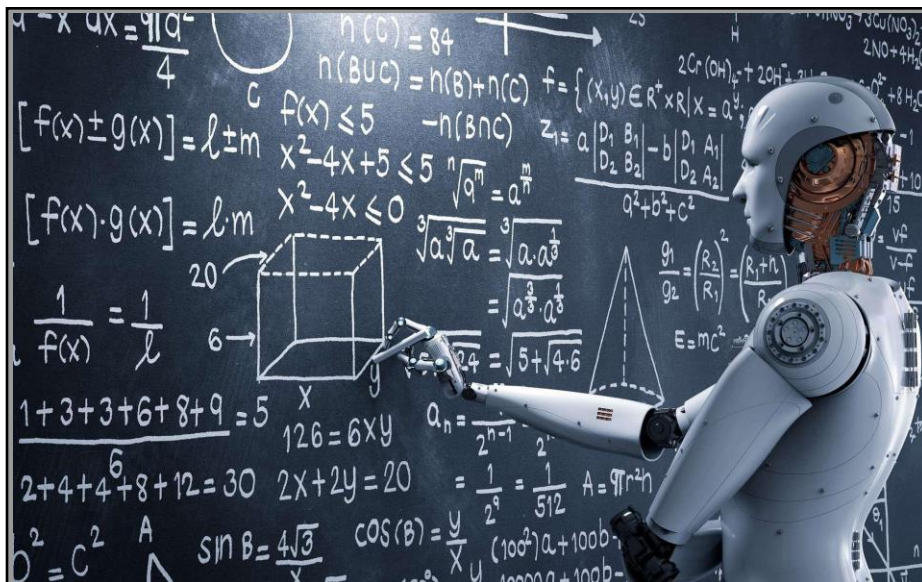


Hình 1.2: AI.

AI có ba mức độ khác nhau:

- Narrow AI: Trí tuệ nhân tạo được cho là hẹp khi máy có thể thực hiện một nhiệm vụ cụ thể tốt hơn so với con người. Nghiên cứu hiện tại về AI hiện đang ở cấp độ này.
- General AI: Trí tuệ nhân tạo đạt đến trạng thái chung khi nó có thể thực hiện bất kỳ nhiệm vụ sử dụng trí tuệ nào có cùng độ chính xác như con người.
- Strong AI: AI rất mạnh khi nó có thể đánh bại con người trong nhiều nhiệm vụ cụ thể.

### 1.4.2. MACHINE LEARNING (HỌC MÁY)



Hình 1.3: Machine learning.

- Machine Learning còn được gọi là học máy. Bạn có thể viết ứng dụng có AI mà không sử dụng học máy, nhưng bạn phải viết cả triệu triệu dòng code để xây dựng các trường hợp xảy ra.
- Học máy là cách để có được AI, máy tự học mà không cần phải code nhiều như không có học máy. Nói cách khác, nếu AI là mục tiêu thì học máy là phương tiện để đạt được mục tiêu đó.
- Máy sẽ được “học” bằng cách train nó với một lượng dữ liệu khổng lồ với một thuật toán, thuật toán có khả năng điều chỉnh và xây dựng model. Tuy nhiên, nếu như trong training dữ liệu có ngôn ngữ khác trong thực tế (tiếng Việt thay vì tiếng Anh...) thì rất có thể máy sẽ dự báo không chính xác nữa.

### 1.4.3. DEEP LEARNING (HỌC SÂU)



*Hình 1.4: Deep learning.*

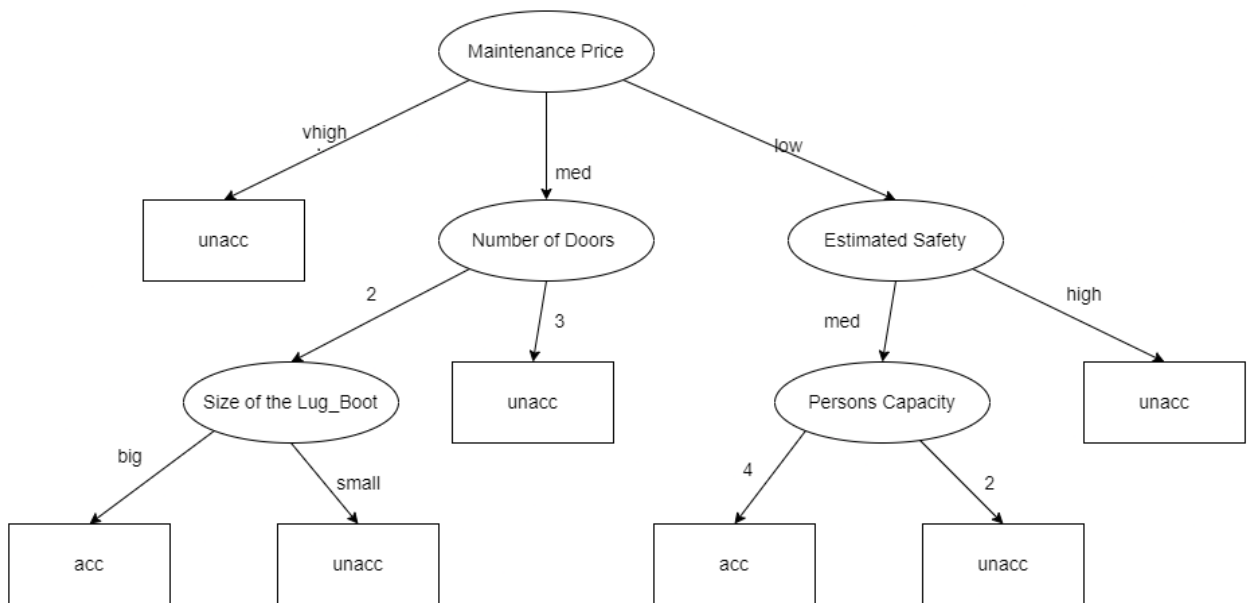
- Deep Learning được bắt nguồn từ thuật toán Neural network của AI, là một ngành nhỏ của Machine Learning.
- Deep learning tập trung giải quyết các vấn đề liên quan đến mạng thần kinh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, tầm nhìn máy tính và xử lý ngôn ngữ tự nhiên.
- Trí tuệ nhân tạo có thể được hiểu đơn giản là được cấu thành từ các lớp xếp chồng lên nhau, trong đó mạng thần kinh nhân tạo nằm ở dưới đáy, Machine learning nằm ở tầng tiếp theo và Deep learning nằm ở tầng trên cùng.



# CHƯƠNG 2: CÂY QUYẾT ĐỊNH

## 2.1. CÂY QUYẾT ĐỊNH

Cây quyết định được dùng để đưa ra tập luật if – then nhằm mục đích dự báo, giúp con người nhận biết về tập dữ liệu. Cây quyết định cho phép phân loại đối tượng tùy thuộc vào các điều kiện tại các nút trong cây, bắt đầu từ gốc cây tới các nút sát lá- Nút xác định phân loại đối tượng. Mỗi nút trong của cây xác định điều kiện đối với thuộc tính mô tả của đối tượng. Mỗi nhánh tương ứng với điều kiện: Nút (thuộc tính) bằng giá trị nào đó. Đối tượng được phân loại nhờ tích hợp các điều kiện bắt đầu từ nút gốc của cây và các thuộc tính mô tả với giá trị của thuộc tính đối tượng.



Hình 2.1. Một ví dụ về cây quyết định đánh giá chất lượng ô tô như thế nào thì phù hợp với việc mua hay không mua ô tô

## 2.2. THUẬT TOÁN TẠO CÂY QUYẾT ĐỊNH

Xét bảng dữ liệu  $T = (A, D)$  trong đó  $A = \{A_1, A_2, \dots, A_n\}$  là tập thuộc tính dẫn xuất,  $D = \{r_1, r_2, \dots, r_n\}$  là thuộc tính mục tiêu. Vấn đề đặt ra là trong tập thuộc tính  $A$  ta phải chọn thuộc tính nào để phân hoạch? Một trong các phương pháp đó là dựa vào độ lợi thông tin. Hay còn gọi là thuật giải ID3.

Lựa chọn chủ yếu trong giải thuật ID3 là chọn thuộc tính nào để đưa vào mỗi nút trong cây. Ta sẽ chọn thuộc tính phân rã tập mẫu tốt nhất. Thước đo độ tốt của việc chọn lựa thuộc tính là gì? Ta cần xác định một độ đo thống kê, gọi là thông tin thu được, đánh giá từng thuộc tính được chọn tốt như thế nào còn phụ thuộc vào việc phân loại mục tiêu của tập mẫu. ID3 sử dụng thông tin thu được đánh giá để chọn ra thuộc tính cho mỗi bước giữa những thuộc tính ứng viên, trong quá trình phát triển cây.

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

Để đánh giá chính xác thông tin thu được, dùng Entropy(S): Độ bất định (độ pha trộn/độ hỗn tạp) của S liên quan đến sự phân loại đang xét.

Trong đó  $p_i$  là xác suất xuất hiện trạng thái  $i$  của hệ thống. Theo lý thuyết thông tin: mã có độ dài tối ưu là mã gán  $-\log_2 p$  bits cho thông điệp có xác suất là  $p$ . S là một tập huấn luyện.

Nếu gọi  $p_{\oplus}$  là xác suất xuất hiện các ví dụ dương trong tập S,  $p_{\ominus}$  là xác suất xuất hiện các ví dụ âm trong tập S. Entropy đo độ bất định của tập S sẽ là:

$$Entropy(S) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

$$\text{Quy định } 0 \cdot \log 0 = 0$$

Chẳng hạn với tập S gồm 14 mẫu có chung một vài giá trị logic gồm 9 mẫu dương và 5 mẫu âm. Khi đó đại lượng Entropy của tập S liên quan đến sự phân loại logic này là:

$$Entropy([9+, 5-]) = - (9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0,940$$

### Chú ý:

Đại lượng Entropy = 0 nếu tất cả thành viên của tập S cùng thuộc một lớp (vì nếu tất cả là dương ( $P+ = 1$ ), do đó  $P- = 0$ ,  $Entropy(S) = 1\log_2 1 + 0\log_2 0 = 0$ ).

Đại lượng Entropy(S) = 1 khi tập S chứa tỉ lệ tập mẫu âm và mẫu dương là như nhau. Nếu tập S chứa tập mẫu âm và tập mẫu dương có tỉ lệ  $P+$  khác  $P-$  thì  $Entropy(S) \in (0,1)$ .

Dựa trên sự xác định entropy, ta tính  $\text{Gain}(S, A) = \text{Lượng giảm entropy mong đợi}$  qua việc chia các ví dụ theo thuộc tính A

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

## **Ưu/ nhược điểm của thuật toán cây quyết định:**

### **Ưu điểm:**

Cây quyết định là một thuật toán đơn giản và phổ biến. Thuật toán này được sử dụng rộng rãi với những lợi ích của nó:

- + Mô hình sinh ra các quy tắc dễ hiểu cho người đọc, tạo ra bộ luật với mỗi nhánh lá là một luật của cây.
- + Dữ liệu đầu vào có thể là dữ liệu missing, không cần chuẩn hóa hoặc tạo biến giả
- + Có thể làm việc với cả dữ liệu số và dữ liệu phân loại
- + Có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê
- + Có khả năng làm việc với dữ liệu lớn

### **Nhược điểm:**

Kèm với đó, cây quyết định cũng có những nhược điểm cụ thể:

- Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu của bạn. Thậm chí, với một sự thay đổi nhỏ trong bộ dữ liệu, cấu trúc mô hình cây quyết định có thể thay đổi hoàn toàn.
- Cây quyết định hay gặp vấn đề **overfitting** (Overfitting là hiện tượng mô hình ghi nhớ quá tốt dữ liệu huấn luyện và phụ thuộc vào nó, việc này khiến cho mô hình không thể tổng quát hóa các quy luật để hoạt động với dữ liệu chưa từng được chứng kiến).

## 2.3. VÍ DỤ MINH HOẠ

Xem xét nhiệm vụ đánh giá chất lượng ô tô được đưa ra bởi tập mẫu dưới đây, thuộc tính mục tiêu ở đây là: Car acceptability có giá trị là acc hoặc unacc, giá trị thuộc tính này dự đoán dựa vào các thuộc tính mô tả

A	B	C	D	E	F	G
Car Acceptability	Buying Price	Maintenance Price	Number of Doors in the Car	Persons Capacity	Size of the Lug_Boot	Estimated Safety of the Car
unacc	vhigh	vhigh	2	2	small	low
unacc	vhigh	vhigh	2	2	small	med
unacc	vhigh	vhigh	2	4	small	low
unacc	vhigh	vhigh	3	2	small	high
unacc	vhigh	vhigh	3	2	med	low
acc	vhigh	med	2	4	big	med
acc	vhigh	med	2	4	big	high
unacc	vhigh	med	2	4	small	med
unacc	vhigh	med	3	4	small	low
unacc	vhigh	med	3	4	small	med
acc	vhigh	low	3	4	big	high
acc	vhigh	low	4	4	big	med
acc	vhigh	low	4	4	big	high
unacc	vhigh	low	4	2	big	med

Giải quyết bước đầu tiên của giải thuật, tạo nút đỉnh của cây quyết định. Nên đưa thuộc tính nào vào cây đầu tiên? ID3 xác định thông tin thu được cho mỗi thuộc tính ứng cử (Buying price, Maintenances price, Number of doors, Persons capacity, Size of Lug\_Boot, Estimated safety of the car) sau đó chọn một trong số đó mà có thông tin thu được cao nhất.

$$IE(S) = -5/14 \log_2(5/14) - 9/14 \log_2(9/14) = 0.94$$

### Buying Price

$$IE[vhigh] = 0 \text{ (loại)}$$

### Maintenances Price

$$IE(vHigh) = 0$$

$$IE(\text{med}) = -2/5 \log_2(2/5) - 3/5 \log_2(3/5) = 0,97$$

$$IE(\text{low}) = -3/4 \log_2(3/4) - 1/4 \log_2(1/4) = 0,81$$

$$IG(\text{Maintenance Price}) = 0.94 - 0 - (5/14 * 0.97) - (4/14 * 0,81) = 0,36$$

### **Number of Doors in the Car**

$$IE(2) = -2/6 \log_2(2/6) - 4/6 \log_2(4/6) = 0,92$$

$$IE(3) = -1/5 \log_2(1/5) - 4/5 \log_2(4/5) = 0,72$$

$$IE(4) = -2/3 \log_2(2/3) - 1/3 \log_2(1/3) = 0,92$$

$$\begin{aligned} IG(\text{Number of Doors in the Car}) &= 0.94 - (6/14 * 0.92) - (5/14 * 0.72) - (3/14 * 0.92) \\ &= 0,09 \end{aligned}$$

### **Persons Capacity**

$$IE(2) = 0$$

$$IE(4) = -5/9 \log_2(5/9) - 4/9 \log_2(4/9) = 0,99$$

$$IG(\text{Persons Capacity}) = 0.94 - 0 - (9/14 * 0.99) = 0,3$$

### **Size of the Lug\_Boot**

$$IE(\text{small}) = 0$$

$$IE(\text{med}) = 0$$

$$IE(\text{big}) = -5/6 \log_2(5/6) - 1/6 \log_2(1/6) = 0,65$$

$$IG(\text{Size of the Lug_Boot}) = 0.94 - 0 - 0 - (6/14 * 0.65) = 0,3$$

### **Estimated Safety of the Car**

$$IE(\text{low}) = 0$$

$$IE(\text{med}) = -2/6 \log_2(2/6) - 4/6 \log_2(4/6) = 0,92$$

$$IE(\text{heigh}) = -3/4 \log_2(3/4) - 1/4 \log_2(1/4) = 0,8$$

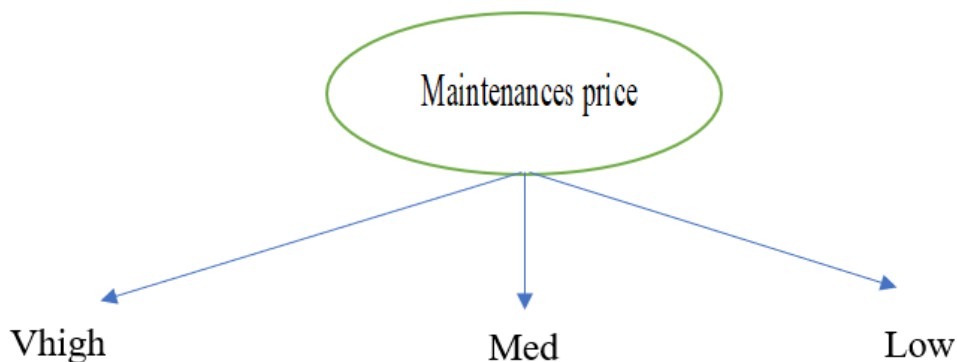
$$IG(\text{Estimated Safety of the Car}) = 0.94 - 0 - (6/14 * 0.92) - (4/14 * 0.8) = 0,31$$

**Giá trị thông tin thu được cho mỗi thuộc tính là:**

- +  $\text{Gain}(S, \text{Buying price}) = 0$
- +  $\text{Gain}(S, \text{Maintenances price}) = 0.36$
- +  $\text{Gain}(S, \text{Number of doors}) = 0.09$
- +  $\text{Gain}(S, \text{Person capacity}) = 0.3$
- +  $\text{Gain}(S, \text{Size of Lug\_boot}) = 0.3$
- +  $\text{Gain}(S, \text{Estimated safety}) = 0.31$

Trong đó tập S là tập mẫu ở bảng trên.

Theo đánh giá thông tin thu được, thuộc tính Maintenances price cung cấp dự đoán tốt nhất về thuộc tính mục tiêu “acc” trên tập mẫu. Do đó, thuộc tính “Maintenances price” được chọn là thuộc tính quyết định cho nút gốc, nhánh được tạo ra dưới nút gốc tương ứng với mỗi giá trị của thuộc tính thời tiết (vhigh, med, low) cùng với tập mẫu sẽ thêm vào mỗi nút con mới.

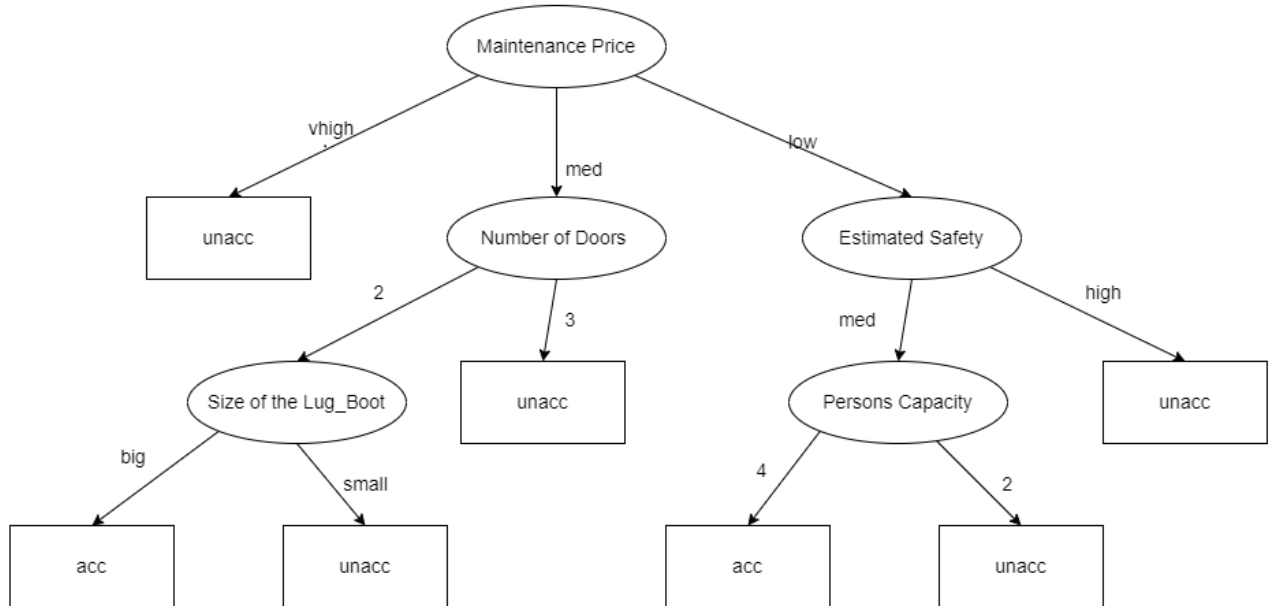


*Hình 2.2. Cây quyết định sau lần phân hoạch đầu tiên*

Mọi mẫu mà có Maintenances price = ‘vhigh’ thì là mẫu âm với thuộc tính Car accestability. Do vậy nút này trở thành nút lá với sự phân loại thuộc tính Car accestability = ‘unacc’. Trái lại với những nút con tương ứng với Maintenances price = ‘med’ và Maintenances price = ‘low’ có giá trị Entropy  $\neq 0$  và cây quyết định sẽ phát triển xa hơn dưới những nút này.

Quá trình chọn thuộc tính mới để phân loại tập mẫu lặp lại cho mỗi nút con. Lúc này chỉ sử dụng những mẫu có liên quan tới nút này. Những thuộc tính mô tả có sự kết hợp chặt chẽ hơn trong cây đã được ngăn chặn. Bởi vậy mà bất kì thuộc tính đưa ra nào có thể xuất hiện theo bất kì nhánh nào của cây. Quá trình xử lý còn tiếp cho mỗi

nút lá mới cho đến khi hai điều kiện sau thoả mãn: Tập thuộc tính rỗng (mọi thuộc tính đều đã nằm dọc theo những nhánh của cây) hoặc tất cả những mẫu có liên quan với nút lá này có cùng giá trị thuộc tính mục tiêu (giá trị entropy của chúng = 0).



*Hình 2.3: Cây quyết định sau các lần phân hoạch.*



# CHƯƠNG 3: ỨNG DỤNG TRONG BÀI TOÁN ĐÁNH GIÁ CHẤT LƯỢNG XE Ô TÔ

## 3.1. BÀI TOÁN

Trong bài toán này chúng tôi sẽ khám phá và tìm hiểu hoạt động của thuật toán cây quyết định dựa trên tập dữ liệu đánh giá xe ô tô.

Đầu vào của các ví dụ trong tập dữ liệu bao hàm tất cả các khả năng và đối với mỗi giá trị đầu vào có thể xảy ra, chỉ có một câu trả lời để dự đoán (do đó, hai ví dụ có cùng giá trị đầu vào sẽ không bao giờ có dự đoán khác nhau). Từ đó ta sẽ có được các kết quả dự đoán chất lượng của xe ô tô với độ chính xác cao.

- Web lấy dữ liệu: <https://archive.ics.uci.edu/ml/datasets/car+evaluation>

## 3.2. CƠ SỞ DỮ LIỆU

- Cơ sở dữ liệu đánh giá ô tô được bắt nguồn từ một mô hình quyết định phân cấp đơn giản ban đầu được phát triển để chứng minh DEX, M. Bohanec, V. Rajkovic: Hệ thống chuyên gia ra quyết định. Mô hình đánh giá xe ô tô sẽ có cấu trúc sau:

Chất lượng của xe ô tô:

- **PRICE:** Giá tổng thể PRICE:
  - + **Buying:** giá mua.
  - + **Maint:** giá bảo trì.
- **TECH:** Đặc điểm kỹ thuật:
  - + **COMFORT:** Độ thoải mái.
    - **Doors:** số lượng cửa.
    - **Persons:** sức chứa bao nhiêu người.
    - **Lug\_Boot:** kích thước cốp đựng hành lý.
  - + **Safety:** Ước tính mức an toàn của xe.
- Các thuộc tính đầu vào được in bằng chữ thường. Bên cạnh khái niệm mục tiêu (CAR), mô hình bao gồm ba khái niệm trung gian: PRICE, TECH, COMFORT.

Mọi khái niệm đều ở trong mô hình ban đầu liên quan đến con cháu cấp thấp hơn của nó bằng một tập hợp các ví dụ.

- Cơ sở dữ liệu đánh giá ô tô chứa các ví dụ với thông tin cấu trúc bị loại bỏ, tức là liên quan trực tiếp CAR với thuộc tính đầu vào: **Buying price, Maintenance price, Number of doors, Capacity, Size of luggage boot, Safety.**
- Do cấu trúc khái niệm cơ bản đã biết, cơ sở dữ liệu này có thể đặc biệt hữu ích cho việc thử nghiệm các phương pháp phát hiện cấu trúc và quy nạp có tính xây dựng.
- Gồm có 6 trường dữ liệu trong cơ sở dữ liệu đánh giá ô tô:
  - Car Acceptability, Buying price, Maintenance price, Number of doors, Capacity, Size of luggage boot, Safety

### 3.3. MỘT SỐ KẾT QUẢ

- Mã hóa các biến phân loại thành số bằng các kỹ thuật khác nhau:

```
!pip install category_encoders
```

- Khai báo các thư viện:

```
import pandas as pd
import seaborn as sns
import category_encoders as ce
from sklearn.model_selection import train_test_split
import graphviz
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, plot_confusion_matrix
```

- + Thư viện pandas sử dụng cho việc thao tác với dữ liệu
- + Thư viện seaborn sử dụng cho việc xuất ra các biểu đồ.
- + Category\_encoders sử dụng nhằm mã hóa dữ liệu.
- + Train\_test\_split sử dụng nhằm mục đích chia tập train\_data, val\_data.
- + Graphviz sử dụng để vẽ cây quyết định, lưu hình ảnh cây.
- + Khai báo cây: `from sklearn import tree`
- + Khai báo thuật toán cây quyết định:  
`from sklearn.tree import DecisionTreeClassifier`



- Sử dụng pandas đọc dữ liệu từ file .csv


```
data=pd.read_csv('Car_Evaluation_Data.csv')
```

- Đặt tên cho các cột:

```
data.columns = ['Car Acceptability', 'Buying Price', 'Maintenance Price',  
'Number of Doors', 'Capacity', 'Size of Luggage Boot', 'Safety']
```

- Xem 10 dòng dữ liệu đầu tiên:

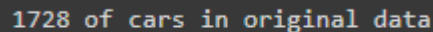
```
data.head(10)
```



	Car Acceptability	Buying Price	Maintenance Price	Number of Doors	Capacity	Size of Luggage Boot	Safety
0	unacc	vhigh	vhigh	2	2	small	low
1	unacc	vhigh	vhigh	2	2	small	med
2	unacc	vhigh	vhigh	2	2	small	high
3	unacc	vhigh	vhigh	2	2	med	low
4	unacc	vhigh	vhigh	2	2	med	med
5	unacc	vhigh	vhigh	2	2	med	high
6	unacc	vhigh	vhigh	2	2	big	low
7	unacc	vhigh	vhigh	2	2	big	med
8	unacc	vhigh	vhigh	2	2	big	high
9	unacc	vhigh	vhigh	2	4	small	low

- In ra xem dữ liệu có bao nhiêu dòng:

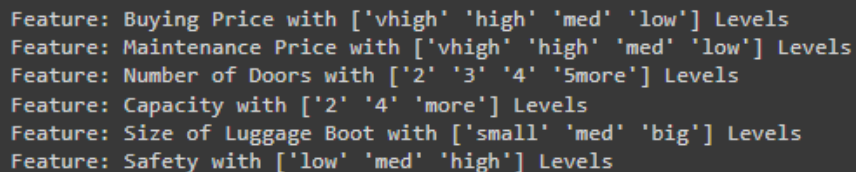
```
print(str(len(data.index)) + " of cars in original data")
```



```
1728 of cars in original data
```

- In ra các giá trị khác nhau của một cột:

```
def show(data):  
    for i in data.columns[1:]:  
        print("Feature: {} with {} Levels".format(i,data[i].unique()))  
show(data)
```



```
Feature: Buying Price with ['vhigh' 'high' 'med' 'low'] Levels  
Feature: Maintenance Price with ['vhigh' 'high' 'med' 'low'] Levels  
Feature: Number of Doors with ['2' '3' '4' '5more'] Levels  
Feature: Capacity with ['2' '4' 'more'] Levels  
Feature: Size of Luggage Boot with ['small' 'med' 'big'] Levels  
Feature: Safety with ['low' 'med' 'high'] Levels
```

- Xem tổng quan dữ liệu:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1728 entries, 0 to 1727
Data columns (total 7 columns):
 #   Column                      Non-Null Count  Dtype
---  -
 0   Car Acceptability           1728 non-null   object
 1   Buying Price                 1728 non-null   object
 2   Maintenance Price           1728 non-null   object
 3   Number of Doors              1728 non-null   object
 4   Capacity                     1728 non-null   object
 5   Size of Luggage Boot        1728 non-null   object
 6   Safety                       1728 non-null   object
dtypes: object(7)
memory usage: 94.6+ KB
```

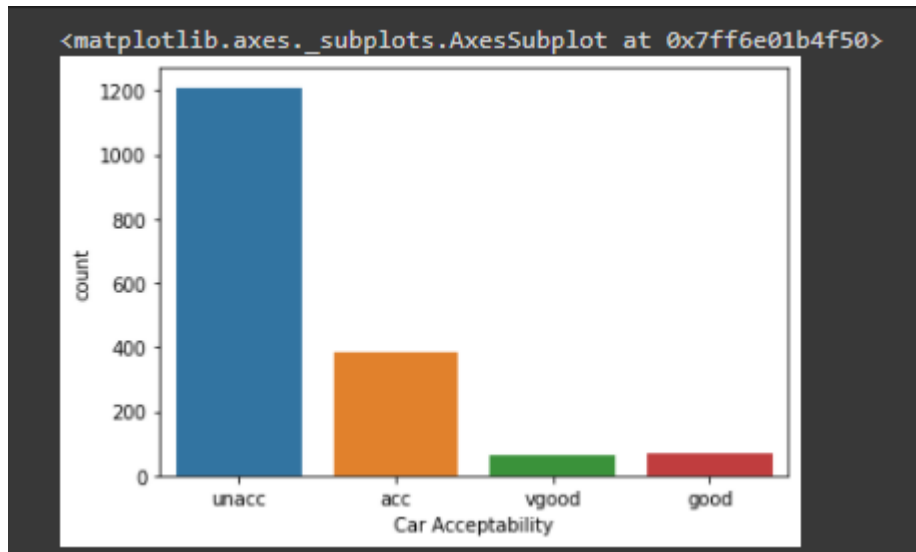
- Xem kiểu dữ liệu của các cột:

```
data.dtypes
```

```
Car Acceptability    object
Buying Price          object
Maintenance Price     object
Number of Doors       object
Capacity              object
Size of Luggage Boot  object
Safety                object
dtype: object
```

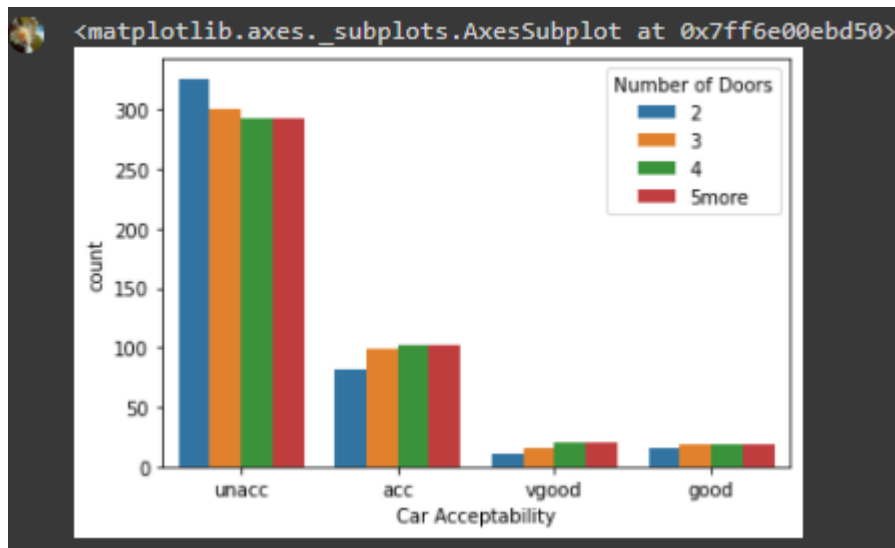
- Vẽ biểu đồ về số lượng dòng dữ liệu của cột **Car Acceptability**:

```
sns.countplot(x='Car Acceptability', data=data)
```



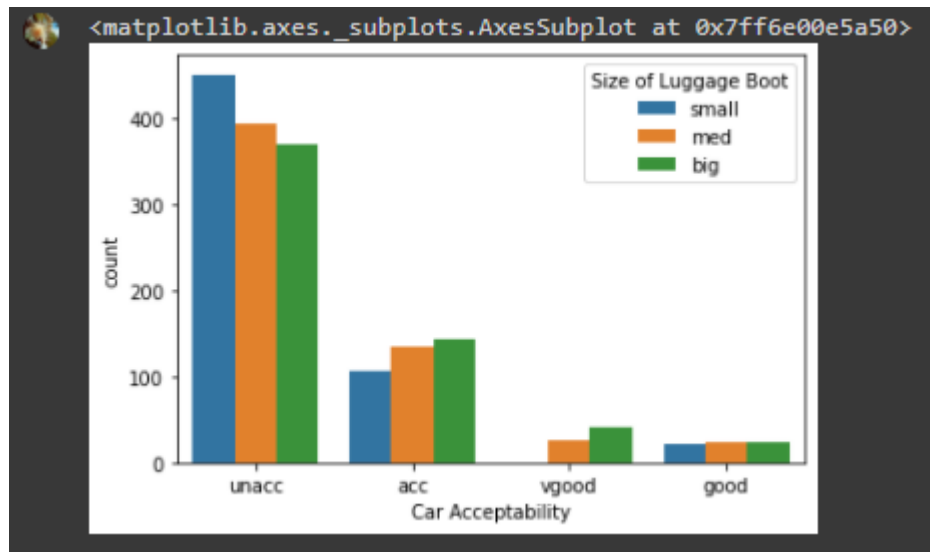
- Vẽ biểu đồ mối liên hệ giữa **Car Acceptability** với **Number of Doors**

```
sns.countplot(x='Car Acceptability', hue='Number of Doors', data=data)
```



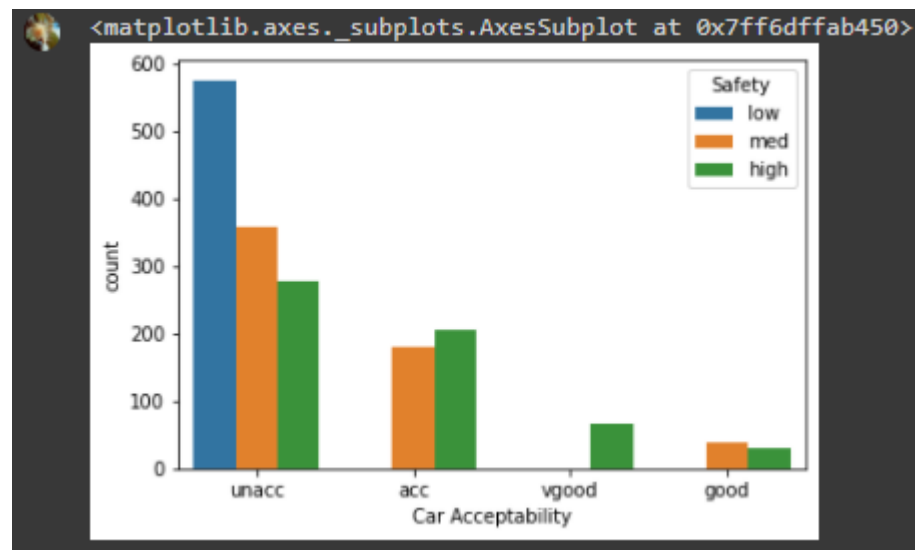
- Vẽ biểu đồ mối liên hệ giữa **Car Acceptability** với **Size of Luggage Boot**

```
sns.countplot(x="Car Acceptability", hue="Size of Luggage Boot", data=dat)
```



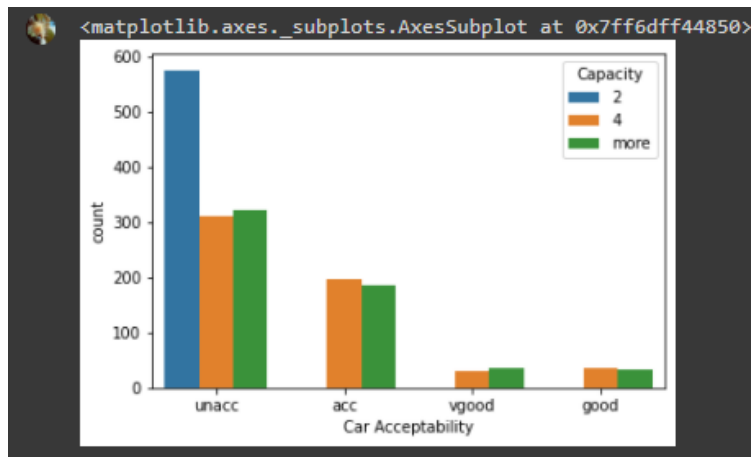
- Vẽ biểu đồ mối liên hệ giữa **Car Acceptability** với **Safety**

```
sns.countplot(x="Car Acceptability", hue="Safety", data=data)
```



- Vẽ biểu đồ mối liên hệ giữa **Car Acceptability** với **Capacity**:

```
sns.countplot(x="Car Acceptability", hue="Capacity", data=data)
```



- Mã hóa dữ liệu:

```
encoder = ce.OrdinalEncoder(cols = ['Car Acceptability', 'Buying Price', 'Maintenance Price', 'Number of Doors', 'Capacity', 'Size of Luggage Boot', 'Safety'])
```

```
data = encoder.fit_transform(data)
```

```
data.head(15)
```

	Car Acceptability	Buying Price	Maintenance Price	Number of Doors	Capacity	Size of Luggage Boot	Safety
0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2
2	1	1	1	1	1	1	3
3	1	1	1	1	1	2	1
4	1	1	1	1	1	2	2
5	1	1	1	1	1	2	3
6	1	1	1	1	1	3	1
7	1	1	1	1	1	3	2
8	1	1	1	1	1	3	3
9	1	1	1	1	2	1	1
10	1	1	1	1	2	1	2
11	1	1	1	1	2	1	3
12	1	1	1	1	2	2	1
13	1	1	1	1	2	2	2
14	1	1	1	1	2	2	3



- Xóa cột Car Acceptability:

```
x = data.drop(['Car Acceptability'], axis = 1)
```

- Lấy dữ liệu cột Car Acceptability:

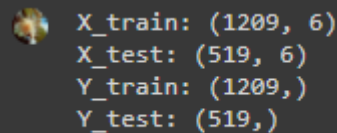
```
y = data['Car Acceptability']
```

- Chia dữ liệu 2 tập: train\_data, val\_data. Với train\_data=0.7 val\_data=0.3

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3,  
random_state = 42)
```

- Xem chiều của dữ liệu:

```
print("X_train: {}".format(x_train.shape))  
print("X_test: {}".format(x_test.shape))  
print("Y_train: {}".format(y_train.shape))  
print("Y_test: {}".format(y_test.shape))
```



```
X_train: (1209, 6)  
X_test: (519, 6)  
Y_train: (1209, )  
Y_test: (519, )
```

- Tạo cây quyết định Entropy với độ sâu là 4

```
clf_en = DecisionTreeClassifier(criterion='entropy', max_depth=4, random_s  
tate=48)
```

- Huấn luyện:

```
clf_en.fit(x_train, y_train)
```

- Dự đoán đầu ra:

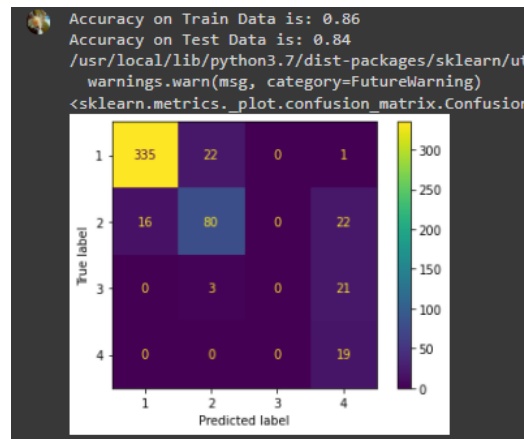
```
yp_train = clf_en.predict(x_train)  
yp_test = clf_en.predict(x_test)
```

- Xem điểm số với 2 tập: train, test

```
print("Accuracy on Train Data is: {}".format(round(accuracy_score(y_train,  
yp_train),2)))  
print("Accuracy on Test Data is: {}".format(round(accuracy_score(y_test,yp  
_test),2)))
```

- Xem confusion matrix:

```
plot_confusion_matrix(clf_en, x_test, y_test)
```



- So sánh kết quả dự đoán với kết quả thực tế:

```
for i in range(len(yp_test)):
    print('Predict: {} \nTruth: {} \n\n'.format([yp_test[i]], y_test[i:i+1]))
```

```
Predict: [1]
Truth: 599 1
Name: Car Acceptability, dtype: int64

Predict: [4]
Truth: 1201 2
Name: Car Acceptability, dtype: int64

Predict: [1]
Truth: 628 1
Name: Car Acceptability, dtype: int64

Predict: [2]
Truth: 1498 2
Name: Car Acceptability, dtype: int64

Predict: [1]
Truth: 1263 1
Name: Car Acceptability, dtype: int64

Predict: [2]
Truth: 931 2
Name: Car Acceptability, dtype: int64

Predict: [1]
Truth: 23 1
Name: Car Acceptability, dtype: int64

Predict: [1]
Truth: 844 1
Name: Car Acceptability, dtype: int64

Predict: [2]
Truth: 964 1
Name: Car Acceptability, dtype: int64
```

```
Predict: [1]
Truth: 298 1
Name: Car Acceptability, dtype: int64

Predict: [1]
Truth: 529 1
Name: Car Acceptability, dtype: int64

Predict: [1]
Truth: 1649 1
Name: Car Acceptability, dtype: int64

Predict: [1]
Truth: 1190 1
Name: Car Acceptability, dtype: int64

Predict: [2]
Truth: 1507 2
Name: Car Acceptability, dtype: int64

Predict: [1]
Truth: 548 1
Name: Car Acceptability, dtype: int64

Predict: [2]
Truth: 371 2
Name: Car Acceptability, dtype: int64

Predict: [2]
Truth: 1340 2
Name: Car Acceptability, dtype: int64

Predict: [1]
Truth: 736 1
Name: Car Acceptability, dtype: int64
```

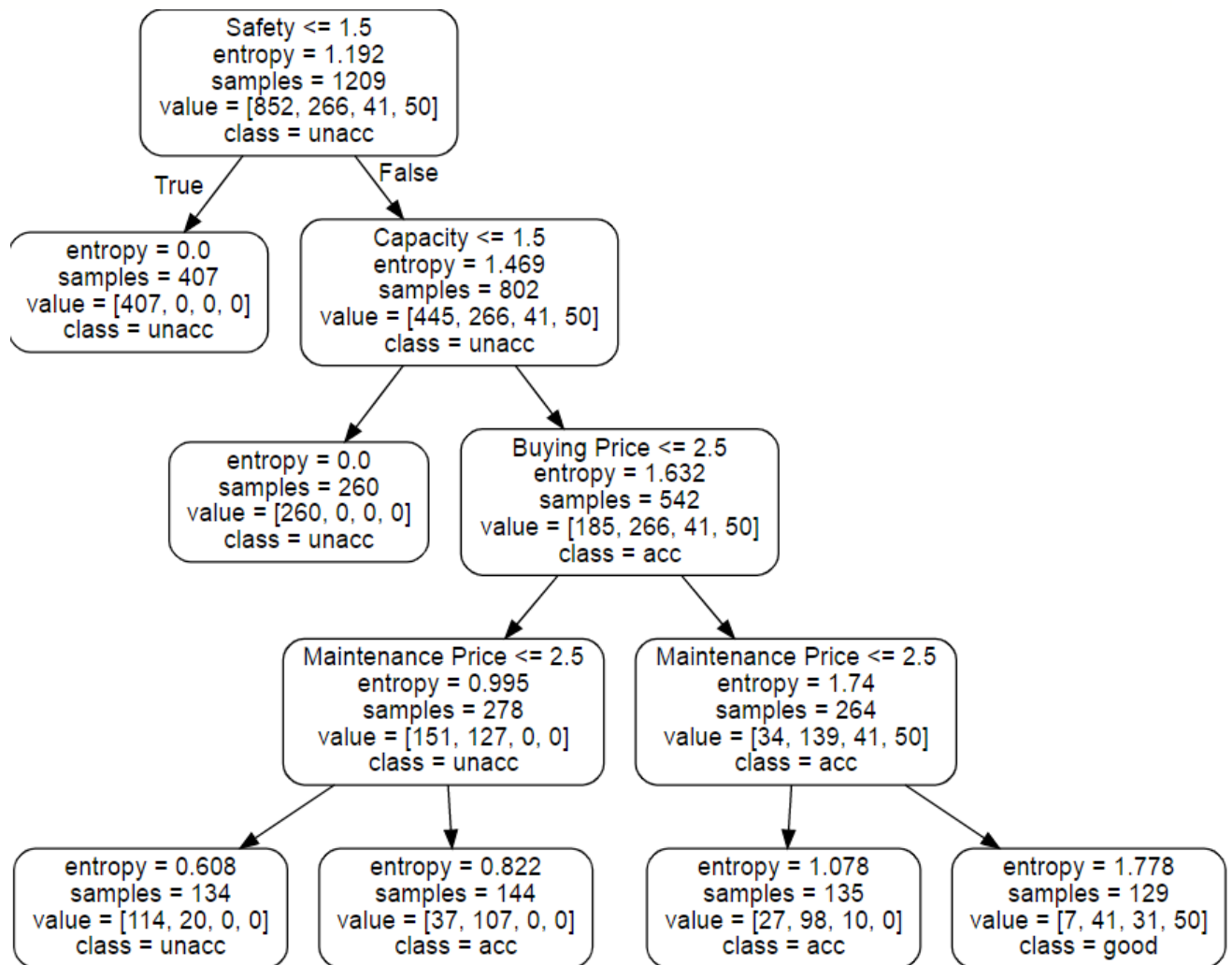
- Vẽ biểu đồ cây quyết định:

```
dot_data = tree.export_graphviz(clf_en, out_file=None,
                                feature_names=x_train.columns,
                                class_names=['unacc', 'acc', 'vgood', 'good'],
                                filled=True, rounded=True,
                                special_characters=True)
```

```
graph = graphviz.Source(dot_data)
```

- Lưu ảnh cây:

```
graph.render("entropy.jpg")
```



# KẾT LUẬN

Với việc công nghệ ngày càng được ứng dụng trong hầu hết các lĩnh vực của đời sống. Những cách làm theo truyền thống cũng dần được thay đổi và ứng dụng các kỹ thuật công nghệ đem lại hiệu quả đáng kể. Với lượng thông tin lớn, những bài toán có độ phức tạp cao vấn đề đặt ra là làm thế nào để phát hiện tri thức, đưa ra lời giải mà thời gian thực hiện có thể chấp nhận được. Trải qua quá trình cùng nhau làm việc nhóm và tìm hiểu thực hiện báo cáo. So với yêu cầu của đề tài đưa ra thì nhóm đã cơ bản nắm được tổng quan về trí tuệ nhân tạo, hiểu thêm về cây quyết định và ứng dụng vào bài toán thực tế. Biết cách làm sao sử dụng các tài liệu có sẵn từ trên mạng, cách để áp dụng kiến thức đã học vào một bài toán trong thực tế. Qua quá trình tìm hiểu cùng nhau các thành viên trong nhóm cũng hoàn thiện được một phần khả năng làm việc nhóm với nhau.

Qua đây chúng em xin gửi lời cảm ơn thầy Trần Hùng Cường đã tận tình giúp đỡ, hướng dẫn chúng em hoàn thành đề tài này. Tuy nhiên do trình độ và kiến thức của chúng em còn hạn chế nên không tránh khỏi những thiếu sót, chúng em rất mong nhận được những góp ý và bổ sung của thầy cô và các bạn để đề tài của chúng em được hoàn thiện hơn.

# TÀI LIỆU THAM KHẢO

- [1] <https://archive.ics.uci.edu/ml/datasets/car+evaluation>
- [2] [https://github.com/cvnhat/car/blob/main/Car\\_Evaluation.ipynb](https://github.com/cvnhat/car/blob/main/Car_Evaluation.ipynb)