­­

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**

Lớp ATTN2023

**Giảng viên hướng dẫn:** TS Nguyễn Thanh Sơn

NGHIÊN CỨU VỀ CẤU TRÚC DỮ LIỆU

DECISION TREE - CLASSIFICATION

**CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**ĐỒ ÁN CUỐI KÌ**

**Thời gian thực hiện:** 4/4/2024 – 17/4/2024

**Nội dung thực hiện**

**Sinh viên thực hiện:** Đoàn Việt Khải

**MSSV:** 23520673

**MỤC LỤC**

[1.Nội dung 1 2](#_Toc82987684)

[a. Nội dung a 2](#_Toc82987685)

[2.Nội dung 2 2](#_Toc82987686)

[a. Nội dung a 2](#_Toc82987687)

# GIỚI THIỆU CHUNG VỀ DECISION TREE

Trong cuộc sống, có rất nhiều tình huống chúng ta quan sát, suy nghĩ và đưa ra quyết định bằng cách đặt câu hỏi. Chẳng hạn như bản thân em, mỗi lần đến thời gian dùng bữa, trong thâm tâm em luôn xuất hiện câu hỏi “Hôm nay ăn gì?”

Em sẽ xác định:

* Nếu là đầu tháng thì đi ăn lẩu với bạn
* Nếu là cuối tháng thì ăn cơm chay

Như vậy, dựa theo ngày em sẽ quyết định được ngày hôm đó mình ăn món gì.

Đây chính là một dạng cơ bản của cây quyết định (decision tree)

## Các khái niệm

### Định nghĩa decision tree:

Cây quyết định (decision tree) là một cấu trúc dữ liệu dạng phân cấp được dùng để phân lớp các đối tượng, trong đó mỗi nút bên trong biểu thị một tính năng, các nhánh biểu thị các quy tắc và các nút lá biểu thị kết quả.

Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary) , Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.

Vậy hiểu đơn giản là khi ta có dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết.

Xét ví dụ: Các đặc điểm của chung cư ảnh hưởng đến quyết định người thuê

* Những đặc tính ban đầu là:

+) Nội thất (full hay không)

+) Điện, nước (giá cả như thế nào)

+) Tường (cách âm hay không)

* Xây dựng được decision tree từ đặc tính trên:
* List
* List

## Phân loại

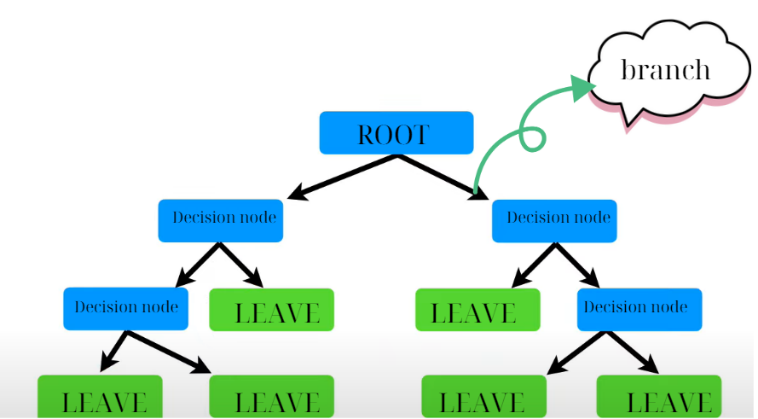
Decision tree là một thuật toán máy học có giám sát (supervised learning) dùng cho cả 2 dạng bài toán hồi quy (regression) và phân loại (classification). Tuy nhiên trong báo cáo đồ án này em chỉ tập trung ở bài toán phân loại (classification)

# CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

## Cấu trúc dữ liệu

### Tổng quan

Về cơ bản, cây quyết định gồm các phần sau:

* Nút gốc: Là nút trên cùng, khởi đầu cho quá trình quyết định
* Nút quyết định (nút nội bộ): Là nút tượng trưng cho sự lựa chọn, liên quan đến đầu vào (input)
* Nút lá: Là nút không có nút con, đại diện cho kết quả (label hoặc số)
* Kết nối giữa các nút là các nhánh, tùy theo cách kết hợp mà cây có độ phức tạp khác nhau

### Xây dựng decision tree

Cây quyết định được xây dựng bằng cách phân vùng đệ quy, bắt đầu từ nút gốc (nút cha đầu tiên), biểu diễn nút con trái và nút con phải, từ hai nút con, tiếp tục phân tách thêm để được cây hoàn chỉnh.

Tuy nhiên, nếu phân tách không điểm dừng thì sẽ dẫn đến cây có độ cao rất sâu và khi áp dụng vào các bài toán phân loại sẽ dẫn đến hiện tượng overfitting. Do đó, cần dựa vào thuộc tính phân lớp của cây để tìm độ cao phù hợp.

Tiến trình kết thúc, khi thỏa một trong các điều kiện sau:

* Tất cả các mẫu của một nút cho trước đều thuộc cùng một lớp
* Không còn thuộc tính mà mẫu dựa vào để phân tách xa hơn
* Không còn mẫu nào cho nhánh test\_attribute = ai (nhánh mà thuộc tính được kiểm tra)

### Những điểm nổi bật của decision tree

* Không đưa ra giả định cụ thể về hình dạng và phân phối dữ liệu: Do decision tree không sử dụng tham số, thay vào đó nó tạo ra các quy tắc quyết định dựa trên tính năng của dữ liệu, dẫn đến kết quả trực quan hơn.
* Xử lí được biến liên tục và phân loại:

+) Biến liên tục: chọn “ngưỡng” và chia dữ liệu thành 2 nhóm dựa trên ngưỡng

Ví dụ: nếu thuộc tính là “tuổi”, thì cây sẽ đưa ra quy tắc: “Nếu tuối lớn hơn (hoặc bé hơn) thì…”

+) Biến phân loại: dựa trên giá trị phân loại để chia thành 2 nhóm

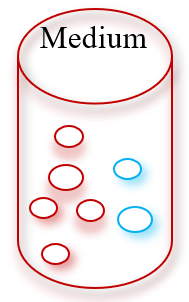
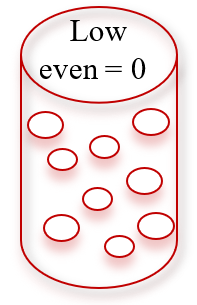
Ví dụ: nếu thuộc tính là “giới tính”, thì cây sẽ đưa ra quy tắc “Nếu giới tính là nam (hoặc là nữ) thì …”

* Xử lí được dữ liệu không cân bằng: bằng cách đánh giá tầm quan trọng của từng nút dựa trên sự phân lớp

## Thuộc tính của decision tree

### Gini

Chỉ số Gini (gini index), còn gọi là Gini Impurity, là một phương pháp để xác định thuộc tính phù hợp trong việc phân chia nút bằng cách đo “độ không tinh khiết” của một tập dữ liệu subset. Chỉ số Gini càng cao tức là càng có nhiều lớp khác nhau trong một tập dữ liệu và ngược lại.



**Định nghĩa theo toán học:** , trong đó:

+) classes: số lớp trong subset

+) 

+) : xác suất mà lớp i có trong subset

**Áp dụng Gini index để xây dựng cây:**

+) Đối với mỗi feature trong tập dữ liệu hoặc mỗi ngưỡng, tính giá trị trung bình của trọng số có chỉ số gini:

Total average of gini index  (: là xác suất lớp i có trong subset)

+) Lấy feature có gini index thấp nhất để chia nút

+) Lặp lại quá trình này đến khi đáp ứng tiêu chí dừng *(1.b)*

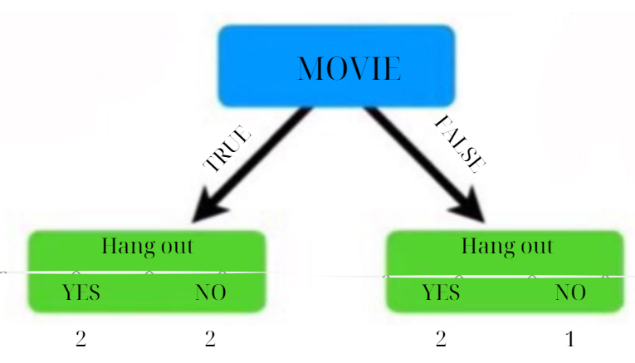
**Xét ví dụ:** *Dựa vào Gini index, xây dựng decision tree cho bộ dữ liệu sau*

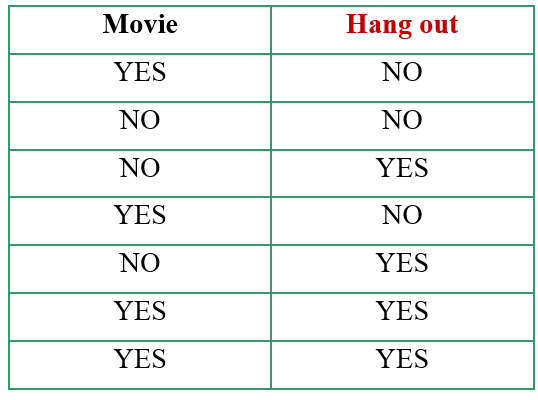
|  |  |  |  |
| --- | --- | --- | --- |
| **Movie** | **Eat** | **Fee** | **Hang out** |
| YES | YES | 70 | NO |
| NO | YES | 100 | NO |
| NO | NO | 120 | YES |
| YES | NO | 200 | NO |
| NO | YES | 250 | YES |
| YES | YES | 340 | YES |
| YES | NO | 500 | YES |

Đầu tiên, ta xem thử liệu “movie”, “eat” hay “fee” sẽ là question của root, bằng cách từ từng feature riêng biệt mà dự đoán label “hang out”.

Để làm được điều này, ta tạo cây đơn giản chỉ để hỏi xem có những ai thích từng feature, từ đó đưa ra số người “hang out”

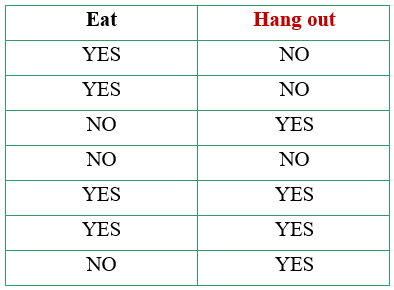
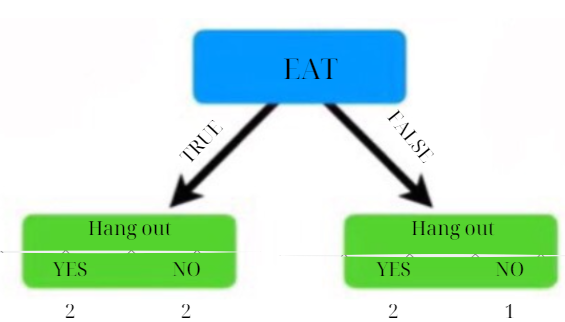
Vì “movie” và “eat” là biến phân loại nên ta sẽ tính trước, còn “fee” là biến liên tục nên ta xử lí sau

* Movie





* Eat





Áp dụng cách tính Gini index đã được trình bày ở trên:

+) Gini index of movie: 



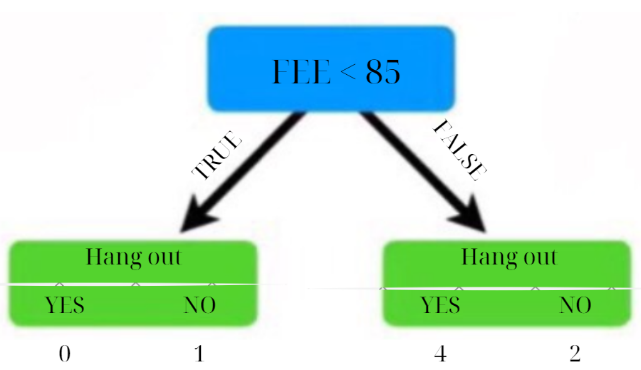
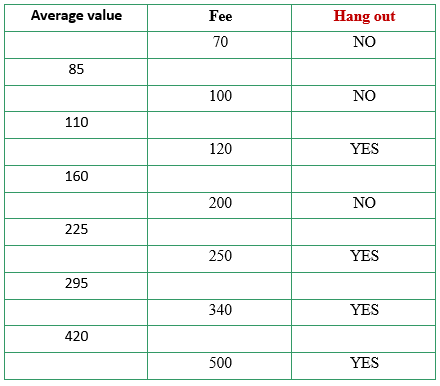




Total Gini index (movie) 

+) Tương tự: Total Gini index (eat) 

Đối với “fee”, do là biến liên tục nên ta sẽ chọn “ngưỡng” bằng cách tính giá trị trung bình của 2 giá trị kề nhau, ứng với mỗi giá trị trung bình ta cũng tính gini index như trên

* Fee



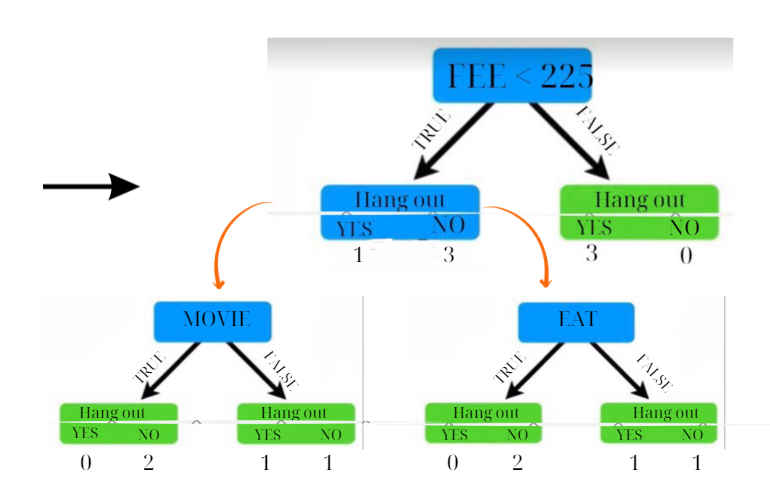
*(Những lớp khác tương tự)*

+) Total Gini index (fee < 85)  +) Total Gini index (fee < 110) 

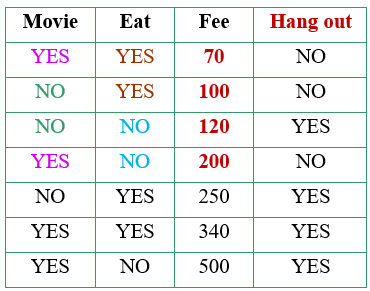
+) Total Gini index (fee < 160)  +) Total Gini index (fee < 225) 

+) Total Gini index (fee < 295)  +) Total Gini index (fee < 420) 

Vì (fee < 225) có gini index nhỏ nhất trong biến liên tục, nên chỉ cần chọn (fee < 225) là được, so sánh với (movie) và (eat) thì (fee < 225) vẫn nhỏ nhất nên root sẽ là (fee < 225)

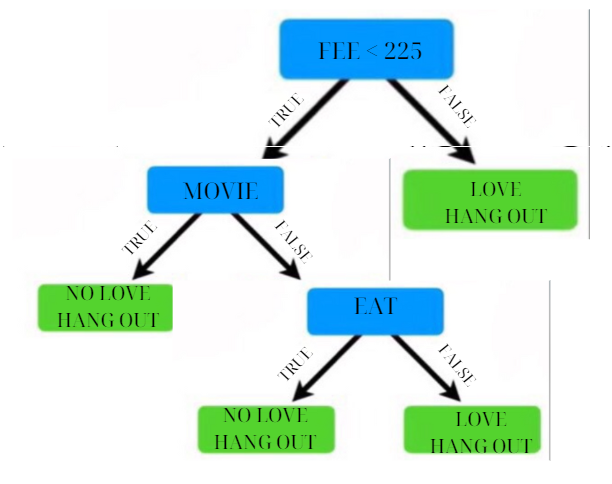
Ta tiếp tục so sánh gini index của “movie” và “eat” trong trường hợp root là

fee < 225, để chọn ra thuộc tính tối ưu



+) Total Gini index (movie) = Total Gini index (eat) = 0,25,

Chọn thuộc tính nào cũng được

Mô hình decision tree hoàn chỉnh:

### Entropy & Information Gain

Giống Gini index, Entropy cũng là một tiêu chuẩn để đo độ “tinh khiết” của dữ liệu, entropy càng cao thì dữ liệu càng “hỗn loạn” (có nhiều lớp) và ngược lại entropy thấp thì ít phân lớp hơn

**Định nghĩa theo toán học:** , trong đó:

+) classes: số lớp trong subset

+) 

+) : xác suất mà lớp i có trong subset

Mục đích của tìm entropy là xác định thuộc tính phân tách để nhận lượng thông tin là nhiều nhất, từ đây nảy sinh khái niệm ***Information Gain***

Information Gain là thước đo lượng thông tin có được từ một thuộc tính trong một phân lớp, entropy càng thấp thì information gain càng cao và ngược lại

**Định nghĩa theo toán học:** ,

với Ti là bộ dữ liệu được phân vào subset thứ i

### Log - loss

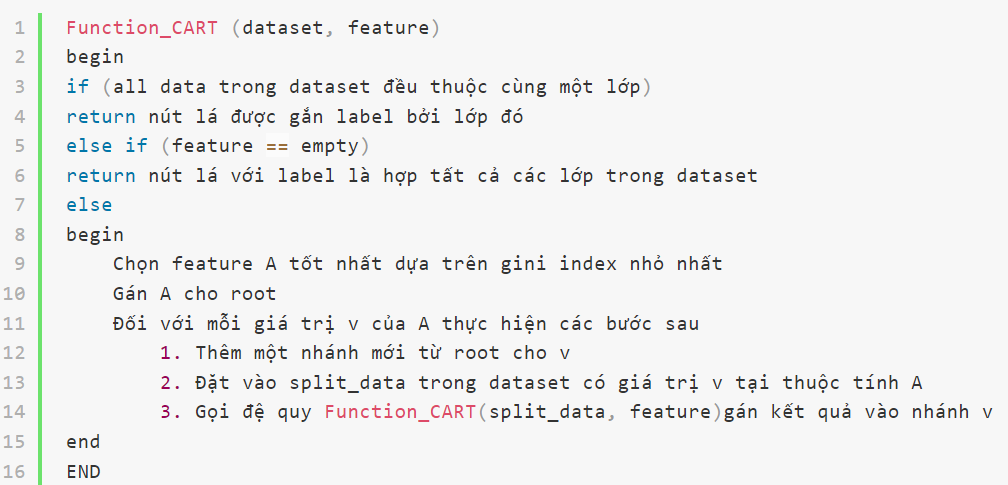
* Log-loss, còn gọi là cross-entropy loss, là một hàm mất mát được sử dụng trong các mô hình phân loại nhị phân và đa lớp. Trong decision tree, log-loss được sử dụng như một tiêu chí để đánh giá chất lượng phân tách của một thuộc tính.
* Cụ thể, log-loss đo lường sự khác biệt giữa dự đoán của mô hình và giá trị thực tế. Nếu mô hình dự đoán chính xác hoàn toàn (tức là dự đoán 1 cho các mẫu thuộc lớp 1 và 0 cho các mẫu thuộc lớp 0), log-loss sẽ bằng 0. Ngược lại, nếu mô hình dự đoán sai hoàn toàn, log-loss sẽ tiến về vô cùng. Tuy nhiên, decision tree thông thường sẽ sử dụng Gini và Entropy nhiều hơn là log-loss.

## Thuật toán

### Trong phạm vi đồ án, em xin trình bày 2 thuật toán cụ thể là CART và ID3

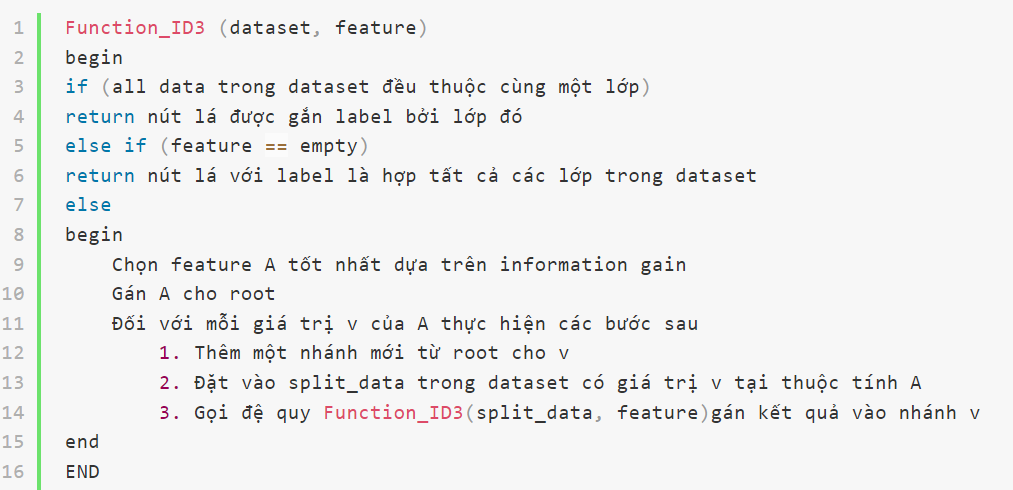
### Thuật toán CART (Classification and Regresstion Tree)

Là một thuật toán dùng được cho cả tác vụ phân loại và hồi quy, được công bố bởi Leo Breiman, Jerome Friedman, Richard Olshen và Charles Stone vào năm 1984. CART sử dụng cách tiếp cận “tham lam” để phân chia dữ liệu tại mỗi nút. Nó đánh giá tất cả các phép chia và chọn cách chia có độ “hỗn loạn” nhỏ nhất. CART dùng Gini index làm thước đo để phân chia dữ liệu. Dưới đây là Pseudo code của CART



### Thuật toán ID3 (Iterative Dichotomiser 3)

Là một thuật toán cơ bản trong việc chọn đặc tính tốt nhất để phân vùng dữ liệu, được phát triển bởi Ross Quinlan vào những năm 1980. ID3 hoạt động theo cách đệ quy, nó xây dựng cây từ trên xuống (top-down) và sử dụng entropy làm thước đo để phân chia dữ liệu một cách hiệu quả nhất. Trong quá trình xây dựng cây, thuộc tính được chọn là thuộc tính có information gain lớn nhất. Dưới đây là pseudo code của ID3



## Phân tích ưu và nhược điểm của decision tree

### Ưu điểm

* Giống hệt quy trình mà con người tuân theo khi đưa ra quyết định nên decison tree khá dễ hiểu
* Là thuật toán giám sát phi tham số, do đó không có giả định về phân chia bộ nhớ và cấu trúc phân lớp
* Yêu cầu xử lí dữ liệu ít hơn so với các thuật toán khác
* Có thể xử lí cả thuộc tính tên và số đầu vào
* Có khả năng xử lí các bộ dữ liệu gây lỗi và những bộ dữ liệu rỗng (missing data)

### Nhược điểm

* Hầu hết các thuật toán (điển hình như ID3, C4.5) đều yêu cầu các thuộc tính mục tiêu phải là các giá trị rời rạc.
* Cần dựa vào tiêu chí phân tách để xây dựng cây, do đó nếu không tìm được tiêu chí phân tách phù hợp sẽ dẫn đến cây quá phức tạp, gây ra hiện tượng overfitting, giảm hiệu suất phân loại trên những dữ liệu mới.
* Nhạy cảm, chỉ cần một thay đổi nhỏ trong bộ dữ liệu cũng có thể dẫn đến thay đổi cấu trúc mô hình.

# XÂY DỰNG MÔ HÌNH PHÂN LOẠI

## Dataset

Tiến hành xây dựng model và thử nghiệm với dataset [IRIS](https://www.kaggle.com/datasets/uciml/iris?resource=download)

## Tiền xử lí dữ liệu (preprocessing data)

### Nội dung 2a

Đây là nội dung trình bày

## Áp dụng thuật toán decision tree

### Nội dung 1a

Đây là nội dung trình bày.

* List
* List

## So sánh

### Nội dung 2a

Đây là nội dung trình bày

## Đánh giá, kết luận, cải tiến (nếu có)