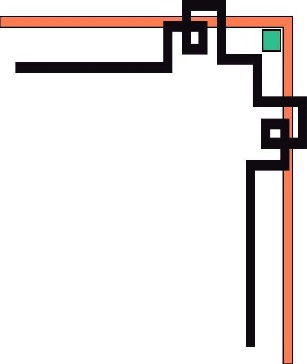
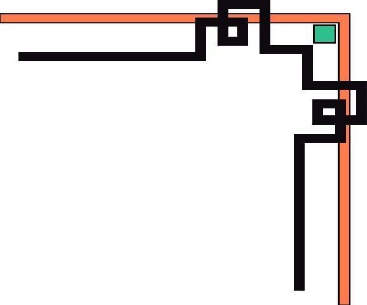
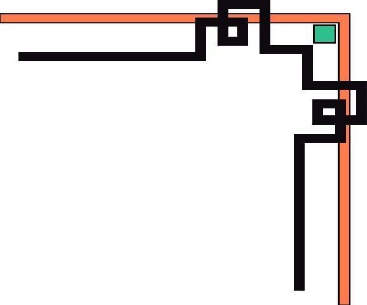
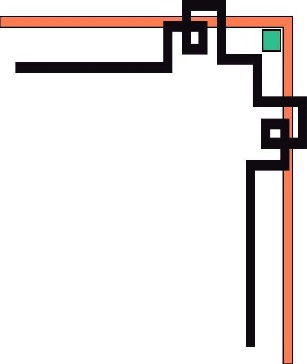
Tt (13)



**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO ĐỒ ÁN**

**MÔN HỌC**

**KHAI THÁC DỮ LIỆU**

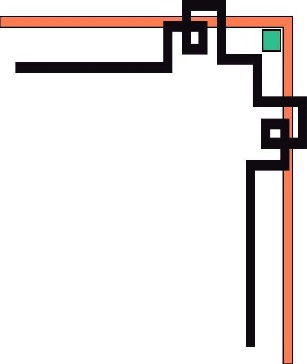
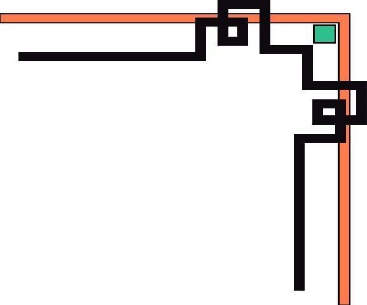
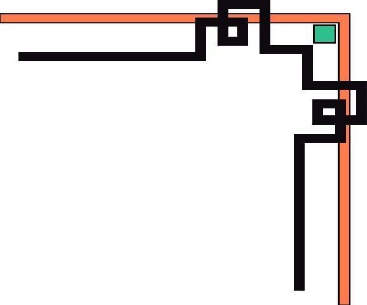
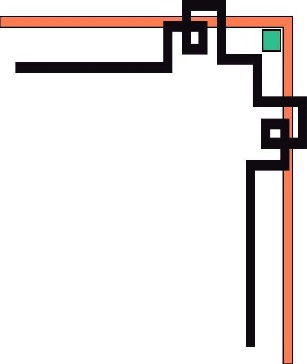


**Đề Tài**

**CÂY QUYẾT ĐỊNH**

**TP. HỒ CHÍ MINH, THÁNG 11 NĂM 2021**

Tt (13)



**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO ĐỒ ÁN**

**MÔN HỌC**

**KHAI THÁC DỮ LIỆU**



**GVHD : Vũ Thanh Nguyên**

**Nhóm :**

|  |  |
| --- | --- |
| **Họ Và Tên** | **MSSV** |
| **1. Lê Lưu Hoàng Nhân** | **2001190186** |
| **2. Trần Hiếu Nghĩa** | **2001191901** |
| **3. Trần Văn Hiền** | **2001190525** |
| **4. Bùi Đức Huy** | **2001190561** |

**TP. HỒ CHÍ MINH, THÁNG 11 NĂM 2021**

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 1](#_Toc86670083)

[**LỜI CAM ĐOAN** 2](#_Toc86670084)

[**LỜI CẢM ƠN** 3](#_Toc86670085)

[**TỔNG QUAN VỀ CÂY QUYẾT ĐỊNH** 4](#_Toc86670086)

[**1. Giới thiệu về cây quyết định.** 4](#_Toc86670087)

[**1.1 Khái niệm về cây quyết định.** 4](#_Toc86670088)

[**1.2 Ưu điểm và hạn chế của cây quyết định.** 5](#_Toc86670089)

[**2. Các thuật toán của cây quyết định.** 6](#_Toc86670090)

[**2.1 Thuật toán ID3.** 7](#_Toc86670091)

[**2.1.1 Tổng quan về ID3.** 7](#_Toc86670092)

[***2.1.2 Entropy trong Cây quyết định (Decision Tree).*** 9](#_Toc86670093)

[**2.1.3** ***Information Gain* *trong Cây quyết định (Decision Tree).*** 10](#_Toc86670094)

[**2.2 Thuật toán C4.5** 11](#_Toc86670095)

[**2.3** **Tiêu chuẩn dừng.** 13](#_Toc86670096)

[**3. Demo.** 13](#_Toc86670097)

[**KẾT LUẬN** 15](#_Toc86670098)

[**1. Đánh giá.** 15](#_Toc86670099)

[**1.1 Lý thuyết.** 15](#_Toc86670100)

[**1.2 Ứng dụng.** 15](#_Toc86670101)

[**2. Hướng phát triển.** 15](#_Toc86670102)

[**TÀI LIỆU THAM KHẢO** 16](#_Toc86670103)

**LỜI MỞ ĐẦU**

Trong nhiều năm qua, cùng với sự phát triển của công nghệ thông tin và ủng dụng của công nghệ thông tin trong nhiều lĩnh vực của đời sống xã hội, thì lượng dữ liệu được các cơ quan thu thập và lưu trữ ngày một nhiều lên Người ta lưu trữ những dữ liệu này vì cho rằng nó ẩn chứa những giá trị nhất định nào đó. Tuy nhiên theo thống kê thì chỉ có một lượng nhỏ của những dữ liệu này khoảng từ 5% đến 10%) là luôn được phân tích, số còn lại họ không biết sẽ phải làm gì và có thể làm gì với những dữ liệu này, nhưng họ vẫn tiếp tục thu thập và lưu trữ và hy vọng những dữ liệu này sẽ cung cấp cho họ những thông tin quý giá một cách nhanh chóng để đưa ra những quyết định kịp thời vào một lúc nào đó. Chính vì vậy, các phương pháp quản trị và khai thác cơ sở dữ liệu truyền thống ngày càng không đáp ứng được thực tế đã làm phát triển một khuynh hưởng kỹ thuật mới đó là Kỹ thuật phát hiện tri thức và khai phá dữ liệu KDD - Knowledge Discover and Data Mining).

Kỹ thuật phát hiện tri thức và khai phá dữ liệu đã và đang được nghiên cứu, ứng dụng trong nhiều lĩnh vực khác nhau trên thế giới, tại Việt Nam kỹ thuật này còn tương đối mới mẻ tuy nhiên cũng đang được nghiên cứu và bắt đầu đưa vào một số ứng dụng thực tế. Vì vậy, hiện nay ở nước ta vấn đề phát hiện tri thức và | khai phá dữ liệu đang thu hút được sự quan tâm của nhiều người và nhiều công ty.

**LỜI CAM ĐOAN**

*Chúng em cam đoan:* Báo Cáo đồ án cuối kỳ với Đề Tài **“Cây quyết định”**. Chúng em đã từng bước nghiên cứu và vận dụng các kiến thức đã được học để tìm hiểu, phân tích và xây dựng đáp ứng tương đối một số các yêu cầu đặt ra.

*Chúng em xin chịu mọi trách nhiệm về Đồ Án báo cáo của mình!*

**LỜI CẢM ƠN**

Lời đầu tiên cho phép chúng em gửi lời cảm ơn tới toàn thể các thầy cô giáo trong Khoa Công Nghệ Tông Tin - Trường Đại học Công Nghiệp Thực Phẩm TP.HCM, những người đã hết mình truyền đạt và chỉ dẫn cho chúng em những kiến thức, những bài học quý báu và bổ ích trong suốt thời gian học vừa qua.

Để hoàn thành được đồ án này, đặt biệt chúng em xin được bày tỏ sự tri ân và xin chân thành cảm ơn giảng viên Gv.Vũ Thanh Nguyên người trực tiếp hướng dẫn, chỉ bảo chúng em trong suốt quá trình học tập và thực hành để hoàn thành đồ án này.

Trong quá trình thực hành và làm báo cáo do năng lực, kiến thức, trình độ bản thân còn hạn hẹp nên không tránh khỏi những thiếu sót. Chúng em mong nhận được sự thông cảm và những ý kiến đóng góp của quý thầy cô và các bạn.

*Chúng em xin chân thành cảm ơn !*

**TỔNG QUAN VỀ CÂY QUYẾT ĐỊNH**

**1. Giới thiệu về cây quyết định.**

Trong cuộc sống hàng ngày, chúng ta vẫn đang sử dụng phương pháp [Decision Tree](https://trituenhantao.io/tu-dien-thuat-ngu/decision-tree/) (Cây quyết định). Chẳng hạn, đến siêu thị mua sữa cho cả gia đình. Câu đầu tiên trong đầu sẽ là: Chúng ta cần mua bao nhiêu sữa?

Ta sẽ xác định: Nếu là ngày thường thì gia đình sẽ sử dụng hết 1 lít sữa, còn cuối tuần thì sẽ là 1,5 lít. Như vậy, dựa theo ngày, ta sẽ quyết định lượng thực phẩm cần mua cho gia đình mình.

Đó chính là một dạng của cây quyết định nhị phân.

**1.1 Khái niệm về cây quyết định.**

Cây quyết định ([Decision Tree](https://trituenhantao.io/tu-dien-thuat-ngu/decision-tree/)) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượngncó 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.

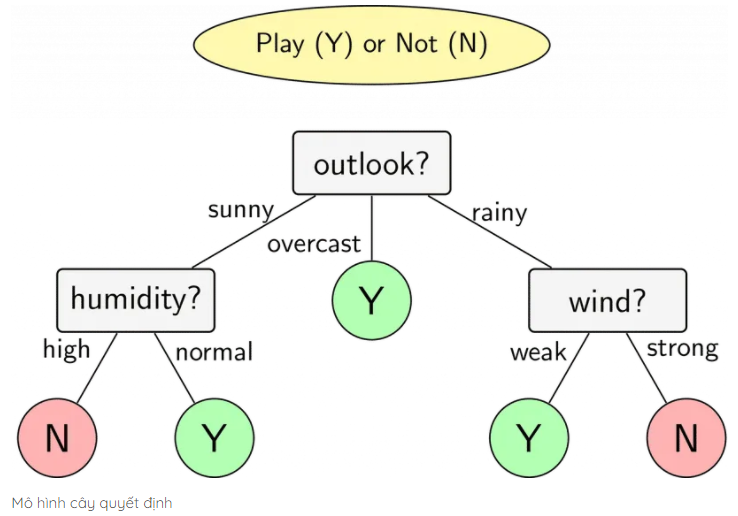
Tóm lại, cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết.

Xét một ví dụ 1 kinh điển về cây quyết định. Giả sử dựa theo thời tiết mà các bạn nam sẽ quyết định đi đá bóng hay không?

Những đặc điểm ban đầu là:

* Thời tiết
* Độ ẩm
* Gió

Dựa vào những thông tin trên, bạn có thể xây dựng được mô hình như sau:



Dựa theo mô hình trên, ta thấy:

Nếu trời nắng, độ ẩm bình thường thì khả năng các bạn nam đi chơi bóng sẽ cao. Còn nếu trời nắng, độ ẩm cao thì khả năng các bạn nam sẽ không đi chơi bóng.

Lưu ý cây quyết định trên không có sự tham gia của thuộc tính “thu nhập” trong thành phần cây, các thuộc tính như vậy được gọi chung là các thuộc tính dư thừa bởi vì các thuộc tính này không ảnh hưởng đến quá trình xây dựng mô hình của cây.

Các thuộc tính tham gia vào quá trình phân lớp thông thường có các giá trị liên tục hay còn gọi là kiểu số (ordered or numeric values) hoặc kiểu rời rạc hay còn gọi là kiểu dữ liệu phân loại (unordered or category values). Ví dụ kiểu dữ liệu lương biểu diễn bằng số thực là kiểu dữ liệu liên tục, kiểu dữ liệu giới tính là kiểu dữ liệu rời rạc (có thể rời rạc hóa thuộc tính giới tính một cách dễ dàng).

**1.2 Ưu điểm và hạn chế của cây quyết định.**

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

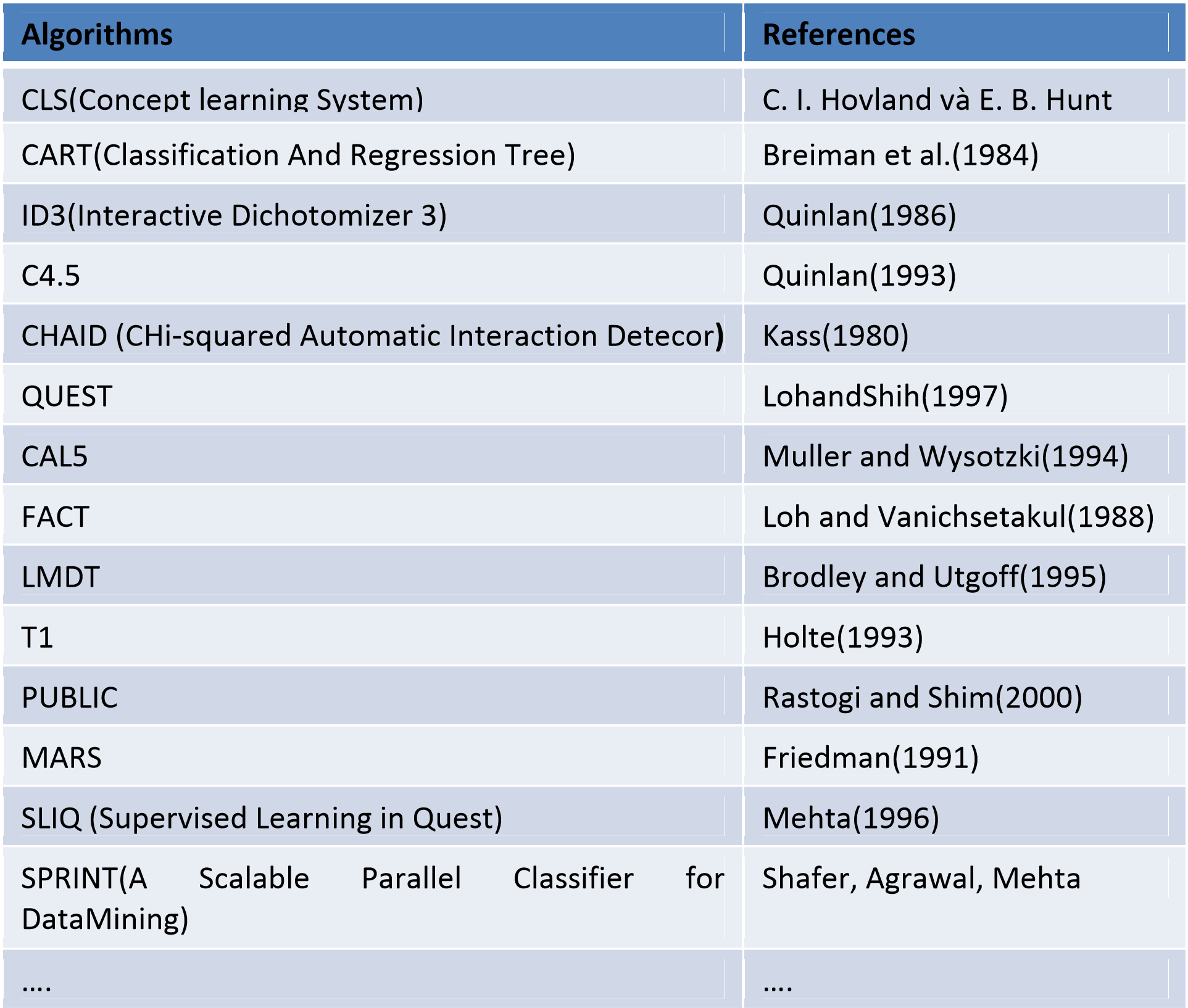
* Cây quyết định tự giải thích và khi được gắn kết lại, chúng có thể dễ dàng tự sinh ra. Nói cách khác, nếu cây quyết định mà có số lượng nút lá vừa phải thì người không chuyên cũng dễ dàng hiểu được nó. Hơn nữa, cây quyết định cũng có thể chuyển sang tập luật. Vì vậy, cây quyết định được xem như là dễ hiểu.
* Cây quyết định có thể xử lý cả thuộc tính tên và số đầu vào.
* Thể hiện của cây quyết định là đủ đa dạng để biểu diễn cho bất kỳ giá trị rời rạc nào.
* Cây quyết định có khả năng xử lý các bộ dữ liệu mà có thể gây ra lỗi.
* Cây quyết định có khả năng xử lý các bộ dữ liệu mà có giá trị rỗng.
* Cây quyết định được xem như là một phương pháp phi tham số. Điều này có nghĩa là cây quyết định không có giả định về sự phân chia bộ nhớ và cấu trúc phân lớp.

Bên cạnh đó, cây quyết định cũng có những bất lợi sau đây:

* Hầu hết các thuật toán (như ID3 hoặc C4.5) bắt buộc các thuộc tính mục tiêu phải là các giá trị rời rạc.
* Các đặc tính liên quan của cây quyết định dẫn đến những khó khăn khác như là độ nhạy với tập huấn luyện, các thuộc tính không phù hợp, nhiễu.
* Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu của bạn. Thạm 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](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/).

**2. Các thuật toán của cây quyết định.**

Với tiêu chí xây dựng cây quyết định ngày càng đơn giản, cho độ chính xác phân lớp cao, chi phí thấp, có khả năng mở rộng,… thì có rất nhiều tác giả đã cho ra đời các thuật toán ngày càng tối ưu hơn. Một số thuật toán tiêu biểu sau:



Trong phạm vi đồ án môn học này chúng ta sẽ nghiên cứu 2 thuật toán chính gồm thuật toán ID3 và C4.5.

**2.1 Thuật toán ID3.**

**2.1.1 Tổng quan về ID3.**

Thuật toán ID3 được phát biểu bởi tác giả Quinlan (trường đại học Syney, Australia) và được công bố vào cuối thập niên 70 của thế kỷ 20. Sau đó, thuật toán này được giới thiệu và trình bày trong mục Induction on decision trees, machine learning năm 1986. ID3 được xem như là một cải tiến của CLS với khả năng lựa chọn thuộc tính tốt nhất để tiếp tục triển khai cây tại mỗi bước. ID3 xây dựng cây quyết định từ trên- xuống (top - down). ID3 sử dụng độ đo Information Gainđể đo tính hiệu quả của các thuộc tính phân lớp. Trong quá trình xây dựng cây quyết định theo thuật toán ID3 tại mỗi bước phát triển cây, thuộc tính được chọn để triển khai là thuộc tính có giá trị Gain lớn nhất.

Hàm xây dựng cây quyết định trong thuật toán ID3:

*Function induce\_tree(tập\_ví\_dụ, tập\_thuộc\_tính)*

*begin*

*if mọi ví dụ trong tập\_ví\_dụ đều nằm trong cùng một lớp then*

*return một nút lá được gán nhãn bởi lớp đó*

*else if tập\_thuộc\_tính là rỗng then*

*return nút lá được gán nhãn bởi tuyển của tất cả các lớp trong tập\_ví\_dụ*

*else begin*

*chọn một thuộc tính P, lấy nó làm gốc cho cây hiện tại;*

*xóa P ra khỏi tập\_thuộc\_tính;*

*với mỗi giá trị V của P*

*begin*

*tạo một nhánh của cây gán nhãn V;*

*Đặt vào phân\_vùng các ví dụ trong tập\_ví\_dụ có giá trị V*

*tại thuộc tính P;*

*Gọi induce\_tree(phân\_vùng , tập\_thuộc\_tính), gắn kết quả*

*vào nhánh V*

*end*

*end*

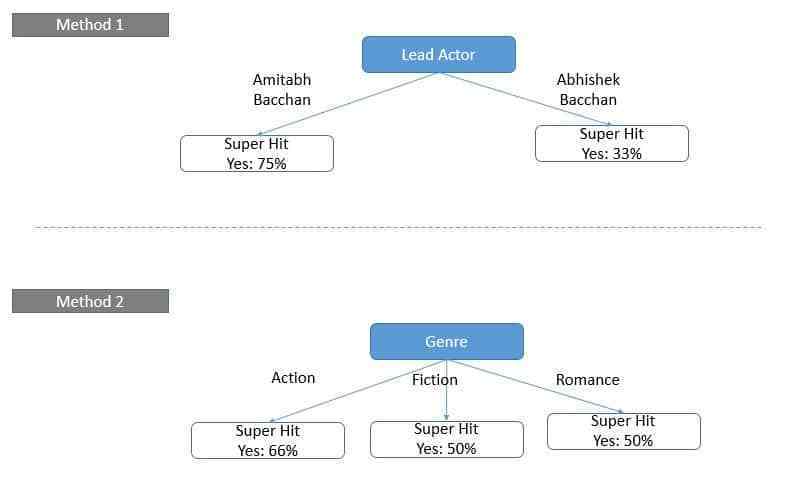
*end*

Xét ví dụ:

Muốn xem xét sự thành công của một bộ phim thông qua hai yếu tố: diễn viên chính của phim và thể loại phim:

| **Lead Actor** | **Genre** | **Hit(Y/N)** |
| --- | --- | --- |
| Amitabh Bacchan | Action | Yes |
| Amitabh Bacchan | Fiction | Yes |
| Amitabh Bacchan | Romance | No |
| Amitabh Bacchan | Action | Yes |
| Abhishek Bacchan | Action | No |
| Abhishek Bacchan | Fiction | No |
| Abhishek Bacchan | Romance | Yes |

Giả sử, bạn muốn xác định độ thành công của bộ phim chỉ trên 1 yếu tố, bạn sẽ có hai cách thực hiện sau: qua diễn viên chính của phim và qua thể loại phim.



Qua sơ đồ, ta có thể thấy rõ ràng ràng, với phương pháp thứ nhất, ta phân loại được rõ ràng, trong khi phương pháp thứ hai, ta có một kết quả lộn xộn hơn. Và tương tự, cây quyết định sẽ thực hiện như trên khi thực hiện việc chọn các biến.

Có rất nhiều hệ số khác nhau mà phương pháp cây quyết định sử dụng để phân chia. Dưới đây, tôi sẽ đưa ra hai hệ số phổ biến là **Information Gain** và **Gain Ratio** (ngoài ra còn hệ số Gini).

**2.1.2 Entropy trong Cây quyết định (Decision Tree).**

[Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) là thuật ngữ thuộc Nhiệt động lực học, là thước đo của sự biến đổi, hỗn loạn hoặc ngẫu nhiên. Năm 1948, Shannon đã mở rộng khái niệm [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) sang lĩnh vực nghiên cứu, thống kê với công thức như sau:

Với một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau x1,x2,…,xn.

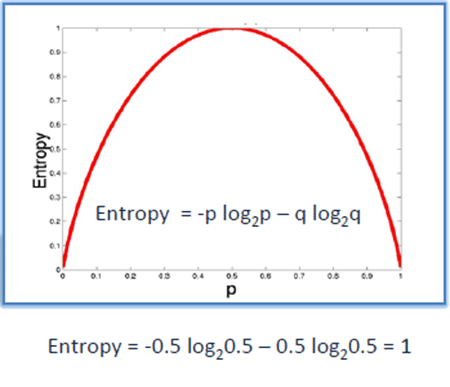
Giả sử rằng xác suất để x nhận các giá trị này là pi=p(x=xi).

Ký hiệu phân phối này là p=(p1 ,p2 ,…,pn). [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) của phân phối này được định nghĩa là:

H(p)=  – ∑nn=1 pi log(pi)

Giả sử bạn tung một đồng xu, [entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) sẽ được tính như sau:

H = -[0.5 ln(0.5) + 0.5 ln(0.5)]



Hình vẽ trên biểu diễn sự thay đổi của hàm [entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/). Ta có thể thấy rằng, [entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) đạt tối đa khi xác suất xảy ra của hai lớp bằng nhau.

* P tinh khiết: pi = 0 hoặc pi = 1
* P vẩn đục: pi = 0.5, khi đó hàm [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) đạt đỉnh cao nhất
  + 1. **Information Gain trong Cây quyết định (Decision Tree).**

Information Gain dựa trên sự giảm của hàm [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) khi tập dữ liệu được phân chia trên một thuộc tính. Để xây dựng một cây quyết định, ta phải tìm tất cả thuộc tính trả về Infomation gain cao nhất.

Để xác định các nút trong mô hình cây quyết định, ta thực hiện tính Infomation Gain tại mỗi nút theo trình tự sau:

**Bước 1**: Tính toán hệ số [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) của biến mục tiêu S có N phần tử với Nc phần tử thuộc lớp c cho trước:

H(S)=  – ∑cc=1 (Nc/N) log(Nc/N)

**Bước 2**: Tính hàm số [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) tại mỗi thuộc tính: với thuộc tính x, các điểm dữ liệu trong S được chia ra K child node S1, S2, …, SK với số điểm trong mỗi child node lần lượt là m1, m2 ,…, mK , ta có:

H(x, S) = ∑Kk=1 (mk / N) \* H(Sk )

**Bước 3**: Chỉ số Gain Information được tính bằng:

**G(x, S) = H(S) – H(x,S)**

Với ví dụ 2 trên, ta tính được hệ số [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) như sau:

[Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/)Parent= -(0.57\*ln(0.57) + 0.43\*ln(0.43)) = 0.68

Hệ số [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) theo phương pháp chia thứ nhất:

[Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/)left= -(.75\*ln(0.75) + 0.25\*ln(0.25))  = 0.56[Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/)right= -(.33\*ln(0.33) + 0.67\*ln(0.67)) = 0.63

Ta có thể tính hệ số **Information Gain** như sau:

Information Gain = 0.68 – (4\*0.56 + 3\*0.63)/7 = 0.09

Hệ số [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) với phương pháp chia thứ hai như sau:

E[ntropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/)left= -(.67\*ln(0.67) + 0.33\*ln(0.33))  = 0.63[Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/)middle= -(.5\*ln(0.5) + 0.5\*ln(0.5))  = 0.69

[Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/)right= -(.5\*ln(0.5) + 0.5\*ln(0.5))  = 0.69

Hệ số **Information Gain**:

Information Gain = 0.68 – (3\*0.63 + 2\*0.69 + 2\*0.69)/7 = 0.02

So sánh kết quả, ta thấy nếu chia theo phương pháp 1 thì ta được giá trị hệ số Information Gain lớn hơn gấp 4 lần so với phương pháp 2. Như vậy, giá trị thông tin ta thu được theo phương pháp 1 cũng nhiều hơn phương pháp 2.

**Nhận xét:** Với việc tính toán giá trị Gain để lựa chọn thuộc tính tối ưu cho việc triển khai cây, thuật toán ID3 được xem là một cải tiến của thuật toán CLS. Tuy nhiên thuật toán ID3 còn các vấn đề chưa được giải quyết như sau:

* Vấn đề *overfitting.*
* Độ đo Information Gain chưa thật sự tốt vì còn thiên về các thuộc tính có nhiều giá trị.
* Xử lý các thuộc tính có kiểu giá trị liên tục (ví dụ như kiểu số thực).
* Xử lý các bộ học thiếu giá thuộc tính (missing-value attributes).
* Xử lý các thuộc tính có chi phí (cost) khác nhau.

**2.2 Thuật toán C4.5**

Thuật toán C4.5 cũng được tác giả Quinlan phát triển và công bố vào năm 1996. Thuật toán này là một thuật toán được cải tiến từ thuật toán ID3 và giải quyết hầu hết các vấn đề mà ID3 chưa giải quyết như đã nêu trên. Nó thực hiện phân lớp tập mẫu dữ liệu theo chiến lược ưu tiên theo chiều sâu (Depth - First).

Trong thuật toán ID3, Information Gain được sử dụng làm độ đo. Tuy nhiên, phương pháp này lại ưu tiên những thuộc tính có số lượng lớn các giá trị mà ít xét tới những giá trị nhỏ hơn. Do vậy, để khắc phục nhược điểm trên, ta sử dụng độ đo Gain Ratio (trong thuật toán C4.5) như sau:

Đầu tiên, ta chuẩn hoá information gain với trị thông tin phân tách (split information):

https://i1.wp.com/trituenhantao.io/wp-content/uploads/2020/04/Formula-10.jpg?w=810&ssl=1

Trong đó: Split Info được tính như sau:

https://i2.wp.com/trituenhantao.io/wp-content/uploads/2020/04/Formula-11.jpg?w=810&ssl=1

Giả sử chúng ta phân chia biến thành n nút cón và Di đại diện cho số lượng bản ghi thuộc nút đó. Do đó, hệ số Gain Ratio sẽ xem xét được xu hướng phân phối khi chia cây.

Áp dụng cho ví dụ trên và với cách chia thứ nhất, ta có

Split Info = – ((4/7)\*log2(4/7)) – ((3/7)\*log2(3/7)) = 0.98

Gain Ratio = 0.09/0.98 = 0.092

Thuật toán xây dựng cây quyết định C4.5:

*Function xay\_dung\_cay(T)*

*{*

*<Tính toán tần xuất các giá trị trong các lớp của T>;*

*If <Kiểm tra các mẫu, nếu thuộc cùng một lớp hoặc có rất ít mẫu khác lớp>Then <Trả về 1 nút lá>*

*Else <Tạo một nút quyết định N>;*

*For <Với mỗi thuộc tính A> Do <Tính giá trị Gain(A)>;*

*<Tại nút N, thực hiện việc kiểm tra để chọn ra thuộc tính có giá trị Gain tốt nhất (lớn nhất). Gọi N.test là thuộc tính có Gain lớn nhất>; If <Nếu N.test là thuộc tính liên tục> Then <Tìm ngưỡng cho phép tách của N.test>;*

*For <Với mỗi tập con T` được tách ra từ tập T> Do ( T` được tách ra theo quy tắc:*

* *Nếu N.test là thuộc tính liên tục tách theo ngưỡng ở bước 5*
* *Nếu N.test là thuộc tính phân loại rời rạc tách theo các giá trị của thuộc tính này. )*

*{ If <Kiểm tra, nếu T' rỗng>} Then*

*<Gán nút con này của nút N là nút lá>;*

*Else*

*<Gán nút con này là nút được trả về bằng cách gọi đệ qui lại đối với hàm xay\_dung\_cay(T'), với tập T'>; }*

*<Tính toán các lỗi của nút N>;*

*<Trả về nút N>;*

*}*

* 1. **Tiêu chuẩn dừng.**

Trong các thuật toán [Decision tree](https://trituenhantao.io/tu-dien-thuat-ngu/decision-tree/), với phương pháp chia trên, ta sẽ chia mãi các node nếu nó chưa tinh khiết. Như vậy, ta sẽ thu được một tree mà mọi điểm trong tập huấn luyện đều được dự đoán đúng (giả sử rằng không có hai input giống nhau nào cho output khác nhau). Khi đó, cây có thể sẽ rất phức tạp (nhiều node) với nhiều leaf node chỉ có một vài điểm dữ liệu. Như vậy, nhiều khả năng [overfitting](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/) sẽ xảy ra.

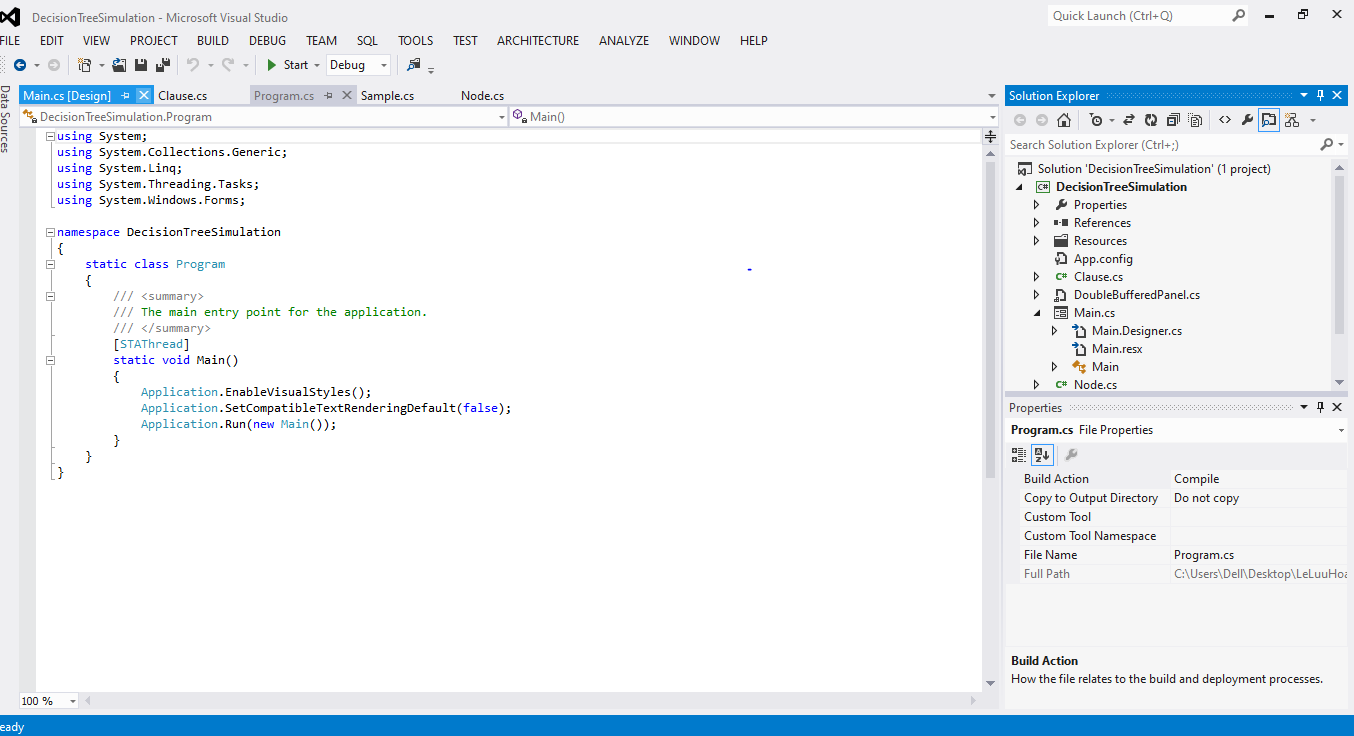
Để tránh trường họp này, ta có thể dừng cây theo một số phương pháp sau đây:

* Nếu node đó có [entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) bằng 0, tức mọi điểm trong node đều thuộc một class.
* Nếu node đó có số phần tử nhỏ hơn một ngưỡng nào đó. Trong trường hợp này, ta chấp nhận có một số điểm bị phân lớp sai để tránh [overfitting](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/). Class cho leaf node này có thể được xác định dựa trên class chiếm đa số trong node.
* Nếu khoảng cách từ node đó đến root node đạt tới một giá trị nào đó. Việc hạn chế chiều sâu của tree này làm giảm độ phức tạp của tree và phần nào giúp tránh [overfitting](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/).
* Nếu tổng số leaf node vượt quá một ngưỡng nào đó.
* Nếu việc phân chia node đó không làm giảm [entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) quá nhiều (information gain nhỏ hơn một ngưỡng nào đó).

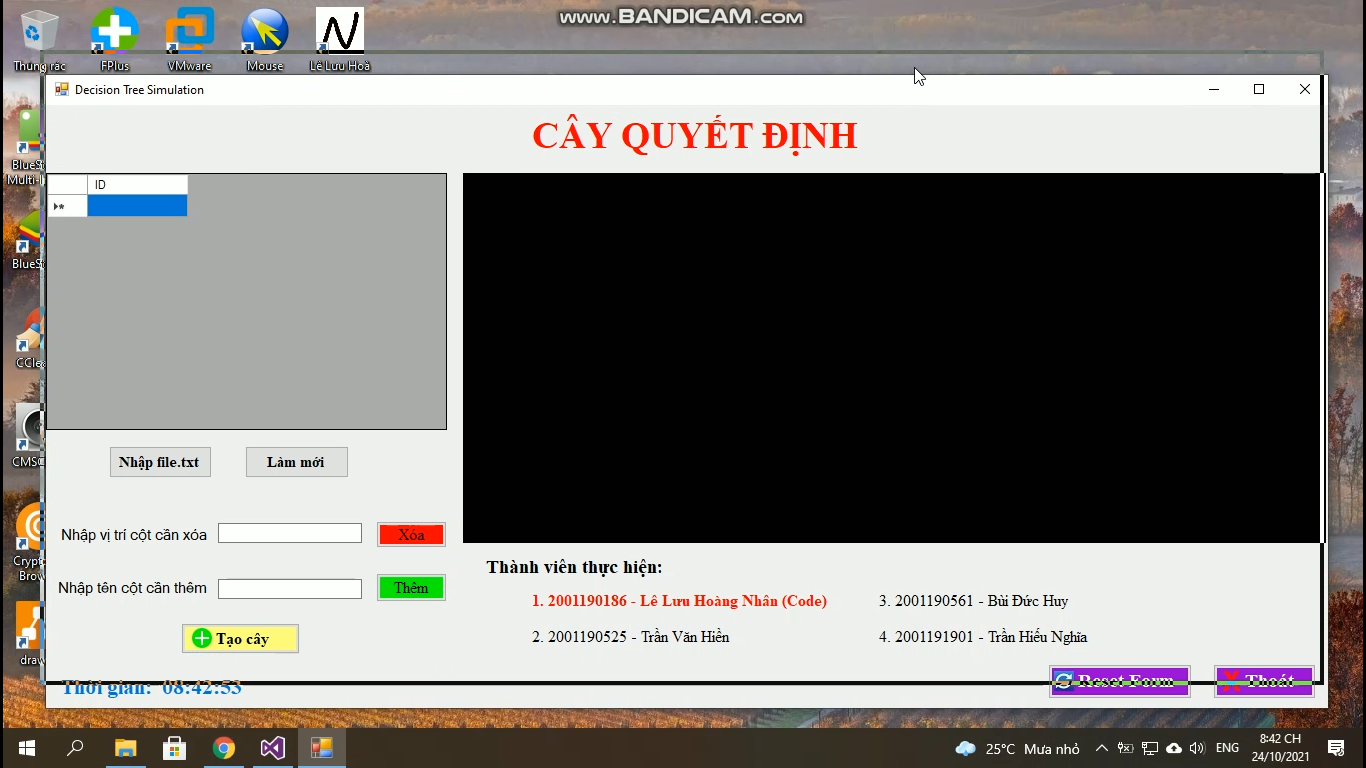
Ngoài ra, ta còn có phương pháp [cắt tỉa cây](https://en.wikipedia.org/wiki/Decision_tree_pruning).

**3. Demo.**

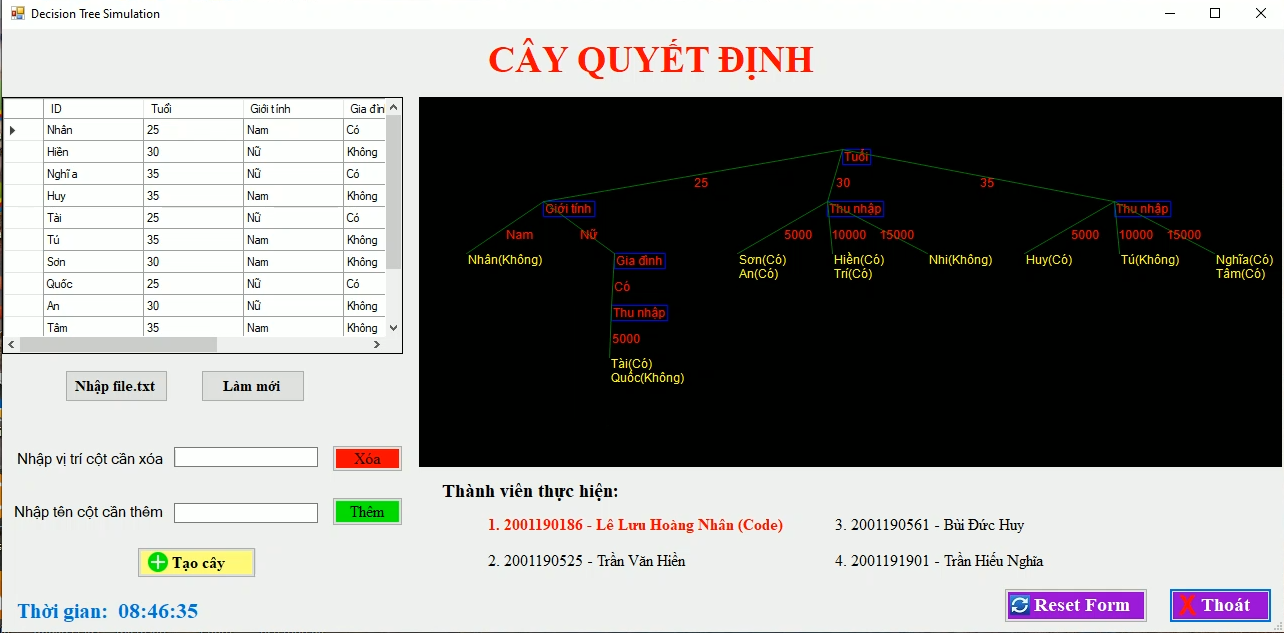
Code



Giao diện ứng dụng



Tạo cây quyết định



**KẾT LUẬN**

**1. Đánh giá.**

Qua quá trình nghiên cứu và tìm hiểu các vấn đề liên quan tới Decision Tree và cơ bản hoàn thành đề tài và đạt được một số kết quả như sau:

**1.1 Lý thuyết.**

**-** Tìm hiểu được nhiều tài liệu hay và bổ ích liên quan tới Decision Tree.

- Nắm được những nội dung cơ bản về Decision Tree,các ưu nhược điểm.

- Nắm được các thuật toán về cây quyết định.

**1.2 Ứng dụng.**

- Xây dựng được chương trình demo cho cây quyết định.

**2. Hướng phát triển.**

- Nghiên cứu thêm một số thuật toán mới về cây quyết định, tìm hiểu sâu hơn về các thuật toán đã nghiên cứu.

- Xây dựng được những chương trình ứng dụng phứt tạp và có tính thực tế hơn bằng cây quyết định.

**TÀI LIỆU THAM KHẢO**

[1] "Cây Quyết Định ([Decision Tree](https://trituenhantao.io/tu-dien-thuat-ngu/decision-tree/))," Trí tuệ nhân tạo, Ngày xuất bản: 06/06/2019, URL: <https://trituenhantao.io/kien-thuc/decision-tree/>, Ngày truy cập: 31/10/2021.

[2] Wikipedia - Bách khoa toàn thư mở - Cây quyết định. http://en.wikipedia.org/wiki/Decision tree

[3] Nguyễn Thị Hạnh, T.S Hồ Cẩm Hà, *“Khai phá dữ liệu bằng cây quyết định”,* 2008

[4] Nguyễn Thị Thùy Linh, *“Nghiên cứu các thuật toán phân lớp dữ liệu dựa trên cây quyết định”*, 2005

[5] <http://www.cs.waikato.ac.nz/ml/weka/index_documentation.html>

[6] <http://repository.seasr.org/Datasets/UCI>