|  | **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**  **KHOA CÔNG NGHỆ THÔNG TIN**    Description: C:\Documents and Settings\Administrator\Desktop\logo dai hoc_khong nen.png    **BÁO CÁO ĐỒ ÁN/CHUYÊN NGÀNH**  **<Chuẩn đoán bệnh & Độ hiệu quả của thuốc theo độ tuổi >**  **Giảng viên hướng dẫn: Bùi Tiến Đức**  **Sinh viên thực hiện: Võ Nhật Huy**  **MSSV: 2000004612**  **Chuyên ngành: Trí tuệ nhân tạo**  **Môn học: LẬP TRÌNH TRÍ TUỆ NHÂN TẠO**  **Khóa: 2020**  **Tp.HCM, tháng 08 năm 2022** |
| --- | --- |

Họ và Tên: Võ Nhật Huy

MSSV: 2000004612

Lớp: 20DTH1A

Email: 2000004612@nttu.edu.vn & [vonhathuy2203@gmail.com](mailto:vonhathuy2203@gmail.com)

Phone: 0388745079

Môn: Lập trình trí tuệ nhân tạo

Giảng viên hướng dẫn: Bùi Tiến Đức

Thời gian thực hiện: **01/07/2022** **đến 22/07/2022**

Nhiệm vụ/nội dung (mô tả chi tiết nội dung, yêu cầu, phương pháp… ) :

* **Tìm hiểu về mô hình cây quyết định**
* **Các thuật toán**

**Nội dung và yêu cầu đã được thông qua Bộ môn.**

*TP.HCM, ngày tháng năm*

| **TRƯỞNG BỘ MÔN**  *(Ký và ghi rõ họ tên)* | **GIÁO VIÊN HƯỚNG DẪN**  *(Ký và ghi rõ họ tên)* |
| --- | --- |

# **Lời Cảm Ơn**

Đầu tiên, em xin gửi lời cảm ơn chân thành đến Trường Đại học Nguyễn Tất Thành đã đưa ngành Trí Tuệ Nhân Tạo vào trương trình giảng dạy. Em cảm thấy thật may mắn khi một lần nữa được Thầy Bùi Tiến Đức giảng dạy cho chúng, đối với em sau thời gian học môn Lập Trình Trí Tuệ Nhân Tạo đó là hành trình của Thầy và trò cùng nhau tiến bước mà Thầy chính là người lái đò dẫn dắt và hướng chúng em đến với thành công của môn học. Em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn – Thầy Bùi Tiến Đức đã dạy dỗ, truyền đạt những kiến thức quý báu cho em trong suốt thời gian học tập vừa qua.

Bộ môn Lập Trình Trí Tuệ Nhân Tạo là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên nói chung và riêng bản thân em nói riêng. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ và hạn hẹp. Mặc dù em đã cố gắng hết sức nhưng chắc chắn bài báo của em khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong các thầy/cô chấm bài xem xét và góp ý để bài tiểu luận của em được hoàn thiệt hơn.

Kính chúc người Thầy của chúng em có nhiều sức khỏe, hạnh phúc, ngọn lửa nghề ngày càng cháy bỏng thành công trên con đường giảng dạy và dẫn dắt sinh viên.

Em xin chân thành cảm ơn!

# Lời Mở Đầu

Trong khoa học máy tính, trí tuệ nhân tạo hay AI ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): *artificial intelligence*), đôi khi được gọi là trí thông minh nhân tạo, là [trí thông minh](https://vi.wikipedia.org/wiki/Tr%C3%AD_th%C3%B4ng_minh) được thể hiện bằng [máy móc](https://vi.wikipedia.org/wiki/M%C3%A1y_m%C3%B3c), trái ngược với trí thông minh tự nhiên của con người. Thông thường, thuật ngữ "trí tuệ nhân tạo" thường được sử dụng để mô tả các máy móc (hoặc máy tính) có khả năng bắt chước các chức năng "nhận thức" mà con người thường phải liên kết với [tâm trí](https://vi.wikipedia.org/wiki/T%C3%A2m_tr%C3%AD), như "học tập" và "giải quyết vấn đề".

Khi máy móc ngày càng tăng khả năng, các nhiệm vụ được coi là cần "trí thông minh" thường bị loại bỏ khỏi định nghĩa về AI, một hiện tượng được gọi là [hiệu ứng AI](https://vi.wikipedia.org/w/index.php?title=Hi%E1%BB%87u_%E1%BB%A9ng_AI&action=edit&redlink=1). Một câu châm ngôn trong Định lý của Tesler nói rằng "AI là bất cứ điều gì chưa được thực hiện."Ví dụ, [nhận dạng ký tự quang học](https://vi.wikipedia.org/wiki/Nh%E1%BA%ADn_d%E1%BA%A1ng_k%C3%BD_t%E1%BB%B1_quang_h%E1%BB%8Dc) thường bị loại trừ khỏi những thứ được coi là AI, đã trở thành một công nghệ thông thường.Khả năng máy hiện đại thường được phân loại như AI bao gồm thành công hiểu lời nói của con người, cạnh tranh ở mức cao nhất trong [trò chơi chiến lược](https://vi.wikipedia.org/wiki/Tr%C3%B2_ch%C6%A1i_chi%E1%BA%BFn_l%C6%B0%E1%BB%A3c) (chẳng hạn như [cờ vua](https://vi.wikipedia.org/wiki/C%E1%BB%9D_vua) và [Go](https://vi.wikipedia.org/wiki/C%E1%BB%9D_v%C3%A2y)),xe hoạt động độc lập, định tuyến thông minh trong mạng phân phối nội dung, và mô phỏng quân sự.

Trong thế kỷ 21, các kỹ thuật AI đã trải qua sự hồi sinh sau những tiến bộ đồng thời về sức mạnh máy tính, [dữ liệu lớn](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u_l%E1%BB%9Bn) và hiểu biết lý thuyết; và kỹ thuật AI đã trở thành một phần thiết yếu của [ngành công nghệ](https://vi.wikipedia.org/w/index.php?title=C%C3%B4ng_nghi%E1%BB%87p_c%C3%B4ng_ngh%E1%BB%87&action=edit&redlink=1), giúp giải quyết nhiều vấn đề thách thức trong học máy, [công nghệ phần mềm](https://vi.wikipedia.org/wiki/C%C3%B4ng_ngh%E1%BB%87_ph%E1%BA%A7n_m%E1%BB%81m) và [nghiên cứu vận hành](https://vi.wikipedia.org/wiki/V%E1%BA%ADn_tr%C3%B9_h%E1%BB%8Dc).

Cho nên ngày hôm nay dưới sự dẫn dắt của thầy/cô và quá trình tìm hiểu học hỏi thêm ở các trang, nguồn thông tin đáng tin cậy, Em dựa theo đó để làm nên bài báo cáo đồ án giữa kì ngày hôm nay. Bằng ngôn ngữ lập trình Python và một số thư viện cần thiết cho việc xây dựng đồ án.

Python (phát âm tiếng Anh: [/ˈpaɪθɑːn/](https://en.wikipedia.org/wiki/WP:IPAE)) là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) bậc cao cho các mục đích lập trình đa năng, do [Guido van Rossum](https://vi.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1) tạo ra và lần đầu ra mắt vào năm [1991](https://vi.wikipedia.org/wiki/1990). Python được thiết kế với ưu điểm

mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o). Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Python luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất.

# Nhận Xét Của Giảng Viên Hướng Dẫn

…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..…………………………………………………………………………………………………..Điểm đồ án: ……………………………………………………………………………………

*TPHCM, Ngày …… tháng …… năm*

**Giáo viên hướng dẫn**

(Ký tên, đóng dấu)

# Mục Lục

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## 1.1 **Cây Quyết Định**

Cây Quyết định (DTs) là một phương pháp học tập có giám sát phi tham số được sử dụng để phân loại và hồi quy . Mục đích là tạo ra một mô hình dự đoán giá trị của một biến mục tiêu bằng cách tìm hiểu các quy tắc quyết định đơn giản được suy ra từ các tính năng dữ liệu. Một cây có thể được coi là một phép gần đúng không đổi từng mảnh.

## 1.2 ​​**Entropy**trong Cây quyết định (Decision Tree)

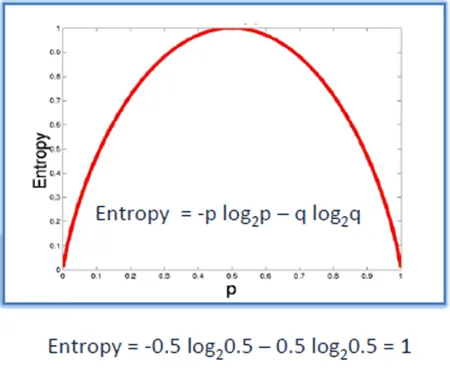
Entropy là một khái niệm khoa học được sử dụng lần đầu tiên trong nhiệt động lực học và sau đó được phổ biến trong các lĩnh vực khác như vật lý, hoá học, y sinh, lý thuyết thông tin,… Trong nhiệt động lực học thì Entropy là một đặc tính vật lý có thể đo lường được có sự liên kết với trạng thái hỗn loạn (disorder) hoặc không chắc chắn (uncertainty).

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 của phân phối này được định nghĩa là: H(p)= – ∑nn=1 pi log(pi)

Giả sử bạn tung một đồng xu, entropy sẽ được tính như sau:

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

Kết luận giá trị entropy cực tiểu đạt được khi phân phối **p** là tinh khiết nhất, tức phân phối hoàn toàn thuộc về một nhóm. Trái lại, entropy cực đại đạt được khi toàn bộ xác suất thuộc về các nhóm là bằng nhau. Một phân phối có entropy càng cao thì mức độ tinh khiết của phân phối đó sẽ càng thấp và ngược lại.

Như vậy về bản chất thì entropy là một thước đo về độ tinh khiết của phân phối xác suất. Dựa trên entropy chúng ta có thể đánh giá tính hiệu quả của câu hỏi ở mỗi node và quyết định xem đâu là câu hỏi hiệu quả hơn (có độ tinh khiết lớn hơn, entropy nhỏ hơn). Tiếp theo chúng ta sẽ cùng tìm hiểu giải thuật tìm kiếm tham lam (greedy) theo chiều từ trên xuống để xây dựng nên cây quyết định dựa trên hàm entropy.

## 1.3 **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 Information gain cao nhất.

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

•Bước 1: Tính toán hệ số 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 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:

*EntropyParent = -(0.57\*ln(0.57) + 0.43\*ln(0.43))* = 0.68

Hệ số 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*

*Entropyright= -(.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 với phương pháp chia thứ hai như sau:

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

*Entropymiddle = -(.5\*ln(0.5) + 0.5\*ln(0.5)) = 0.69*

*Entropyright = -(.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.

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

#### Ưu Điểm

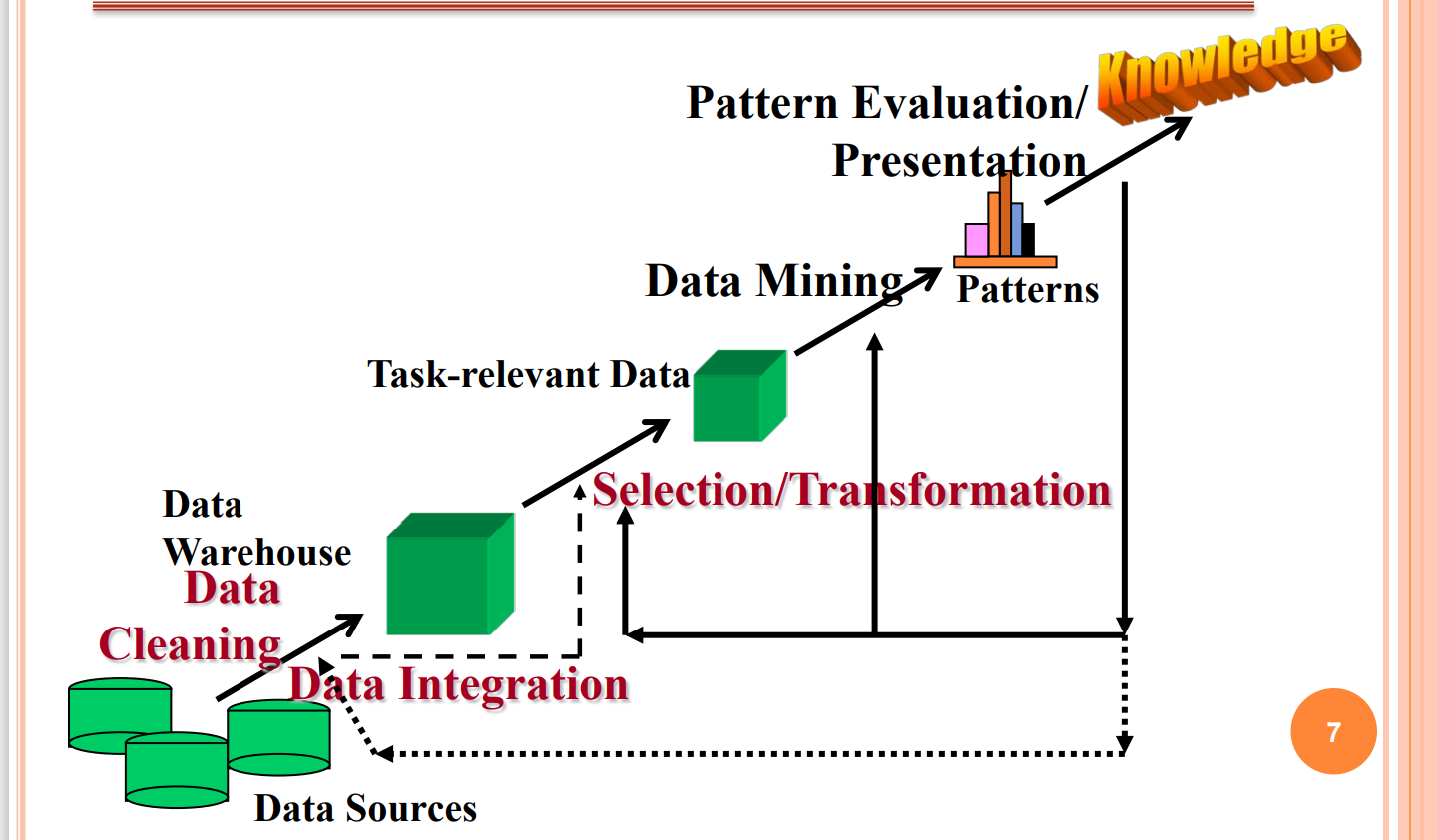
Cây quyết định là một thuật toán đơn giản và phổ biến. Thuật toán này được sử dụng rộng rãi bởi những lợi ích của nó:

* 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 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 là việc với dữ liệu lớn

#### Nhược Điểm

* 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](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/)

# CHƯƠNG 2: QUY TRÌNH KHAI PHÁ TRI THỨC



## **2.1 Quá trình phát hiện tri thức gồm các bước cơ bản sau:**

-Chọn lọc dữ liệu (selection): Đây là giai đoạn tập hợp các dữ liệu được khai thác từ một CSDl, một kho dữ liệu, thậm chí từ các nguồn ứng dụng web vào một CSDl riêng. chúng ta chỉ chọn ra những dữ liệu cần thiết cho các giai đoạn sau. Tuy nhiên, công việc thu gom dữ liệu vào một CSDl lớn thường rất khó khăn vì dữ liệu nằm ở khắp nơi và dạng tạo lập khác nhau.

-Tiền xử lý dữ liệu (preprocessing): Phần lớn các cSDl đều ít nhiều mang tính không nhất quán. Vì vậy khi gom dữ liệu rất có thể mắc một số lỗi như dữ liệu không đầy đủ, chặt chẽ và không lôgic (bị trùng lặp, giá trị bị sai lệch,...). Do đó cần phải được “tiền xử lý” trước khi khai phá dữ liệu nếu không sẽ gây nên những kết quả sai lệch nghiêm trọng.

-Chuyển đổi dữ liệu (transformation): Trong giai đoạn này dữ liệu sẽ được chuyển đổi về dạng thuận tiện để tiến hành các thuật toán khám phá dữ liệu. -Khai phá dữ liệu (Data mining): trong giai đoạn này ta sử dụng các kỹ thuật nhằm phát hiện ra các tri thức tiềm ẩn trong dữ liệu. một số kỹ thuật được sử dụng đó là: phân lớp, gom cụm, luật kết hợp, Đánh giá kết quả mẫu: Đây là giai đoạn cuối cùng trong tiến trình KDD.

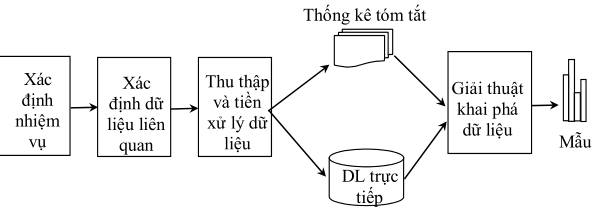
Trong giai đoạn này, các mẫu dữ liệu được chiết xuất bởi các phần mềm khai phá dữ liệu. Không phải bất cứ mẫu nào cũng đều có ích, thậm chí còn bị sai lệch. chính vì vậy, cần phải xác định và lựa chọn những tiêu chuẩn đánh giá sao cho sẽ chiết xuất ra các tri thức cần thiết. Nếu phát hiện tri thức là toàn bộ quá trình chiết xuất tri thức từ các CSDl thì KPDL là giai đoạn chủ yếu của quá trình đó.

Như trên đã trình bày, trong quá trình phát hiện tri thức, khâu KPDL được thực hiện sau các khâu tinh lọc và tiền xử lý dữ liệu, tức là việc khai phá để tìm ra các mẫu có ý nghĩa được tiến hành trên tập dữ liệu có hy vọng là sẽ thích hợp với nhiệm vụ khai phá đó chứ không phải là khai phá hết dữ liệu với một thời gian đủ dài để lấy được một mẫu không thực sự có ích như khái niệm trong thống kê trước đây. Vì vậy, KPDL thường bao gồm việc thử tìm mô hình phù hợp với tập dữ liệu và tìm kiếm các mẫu từ tập dữ liệu theo mô hình đó. chẳng hạn ta có mô hình là một luật kết hợp thì mẫu là các yếu tố tham gia cùng với các độ hỗ trợ (support) và độ tin cậy (confidence) trong các luật tương ứng. mặc dù các mẫu có thể được trích lọc từ bất kỳ CSDL nào nhưng chỉ có các mẫu được xem là đáng quan tâm xét theo một phương diện nào đó mới được coi là tri thức.

## **2.2 Khai phá tri thức là gì ?**

KPDL là một khái niệm ra đời vào những năm cuối của thập kỷ 80. Nó bao hàm một loạt các kỹ thuật nhằm phát hiện ra các thông tin có giá trị tiềm ẩn trong các tập dữ liệu lớn (các kho dữ liệu). Về bản chất, KPDL liên quan đến việc phân tích các dữ liệu và sử dụng các kỹ thuật để tìm ra các mẫu hình có tính chính quy (regularities) trong tập dữ liệu.

Khai phá dữ liệu là một tiến trình sử dụng các công cụ phân tích dữ liệu khác nhau để khám phá ra các mẫu dưới nhiều góc độ khác nhau nhằm phát hiện ra các mối quan hệ giữa các dữ kiện, đối tượng bên trong CSDL, kết quả của việc khai phá là xác định các mẫu hay các mô hình đang tồn tại bên trong, nhưng chúng nằm ẩn khuất ở các CSDL. Để từ đó rút trích ra được các mẫu, các mô hình hay các thông tin và tri thức từ các CSDL.



# CHƯƠNG 3 XÂY DỰNG MÔ HÌNH KHAI PHÁ TRI THỨC

## 3.1 Xác định mục tiêu cần giải quyết

Trong đồ án giữa kì này, tôi sẽ sử dụng một thuật toán học máy phổ biến ‘cây quyết định’. Tôi sẽ sử dụng thuật toán phân loại này để xây dựng mô hình từ dữ liệu lịch sử của bệnh nhân và phản ứng của họ với các loại thuốc khác nhau. Sau đó, tôi sử dụng cây quyết định đã được đào tạo để dự đoán loại bệnh nhân chưa biết hoặc để tìm một loại thuốc thích hợp cho bệnh nhân mới.

Sử dụng các thư viện sau:

* numpy (as np)
* pandas
* DecisionTreeClassifier from sklearn.tree

## 3.2 Giới thiệu về tập dữ liệu

Hãy tưởng tượng rằng bạn là một nhà nghiên cứu y tế đang biên soạn dữ liệu cho một nghiên cứu. Bạn đã thu thập dữ liệu về một nhóm bệnh nhân, tất cả đều bị bệnh giống nhau. Trong quá trình điều trị, mỗi bệnh nhân phản ứng với một trong 5 loại thuốc, Thuốc A, Thuốc B, Thuốc c, Thuốc x và y.

Một phần công việc của bạn là xây dựng một mô hình để tìm ra loại thuốc nào có thể thích hợp cho một bệnh nhân mắc bệnh tương tự trong tương lai. Các bộ đặc điểm của bộ dữ liệu này là Tuổi, Giới tính, Huyết áp và Cholesterol của bệnh nhân và mục tiêu là loại thuốc mà mỗi bệnh nhân đáp ứng.

Nó là một mẫu của bộ phân loại nhị phân và bạn có thể sử dụng phần đào tạo của tập dữ liệu để xây dựng cây quyết định, sau đó sử dụng nó để dự đoán phân loại của một bệnh nhân chưa biết hoặc để kê đơn cho một bệnh nhân mới.

## 3.3 Thiết lập cây quyết định

Bắt đầu sẽ sử dụng train/test split cho decision tree, thêm train\_test\_split từ sklearn.cross\_validation.

-train\_test\_split sẽ trả về 4 tham số khác nhau.

Tôi sẽ đặt tên cho chúng:

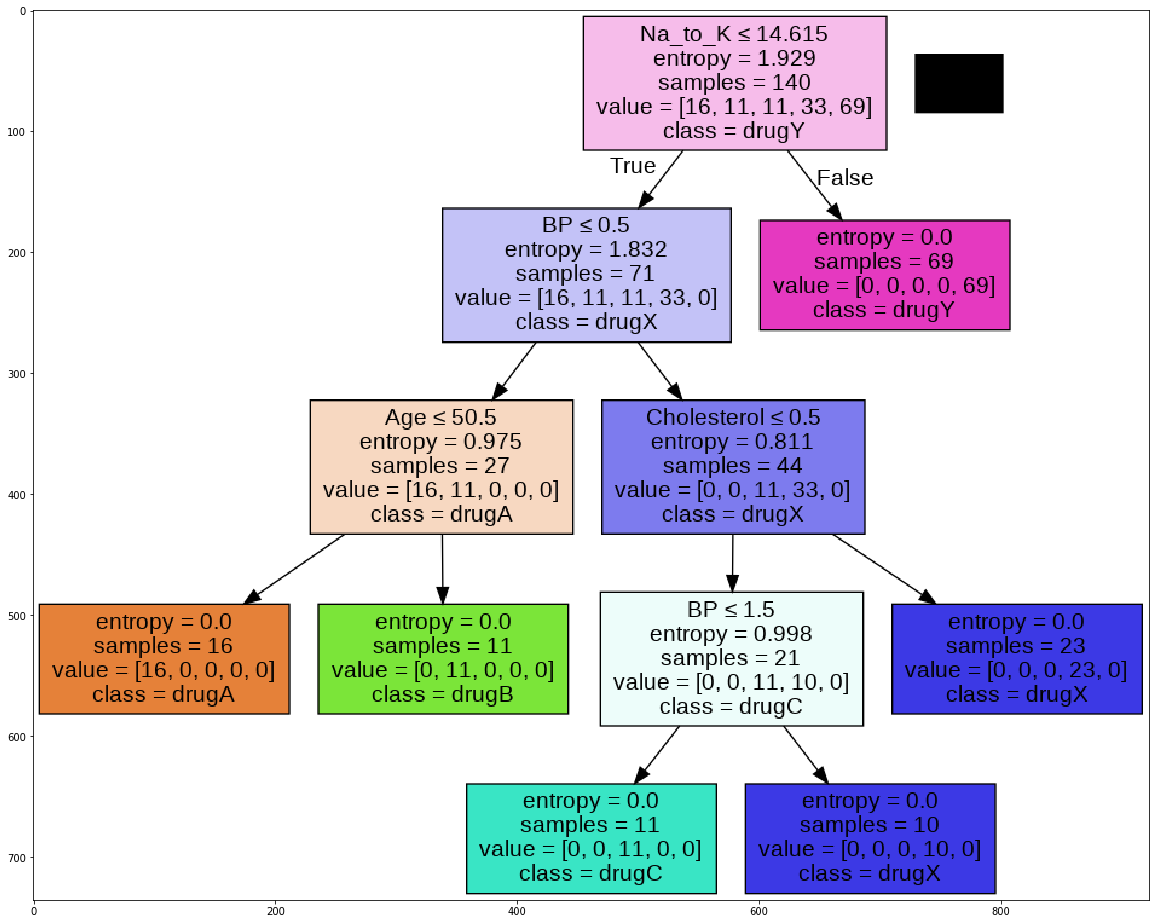
X\_trainset, X\_testset, y\_trainset, y\_testset

Train\_test\_split sẽ cần các tham số:X, y, test\_size = 0,3 và random\_state = 3.

X và y là các mảng được yêu cầu trước khi phân tách, test\_size đại diện cho tỉ lệ của tập dữ liệu thử nghiệm và random\_state đảm bảo rằng chúng ta có được các phân tách giống nhau.

Sau khi đã tạo ra các biến lúc này tiếp tục tạo nên một DecisionTreeClassifier và gọi là DrugTree. Bên trong cây quyết định vừa tạo chỉ định dùng “entropy"

để chúng ta thấy mức tăng thông tin của mỗi nút với max\_depth = 4.



Chúng ta thấy dù có thay đổi các giá trị khác như thế nào, miễn là 3 giá trị Na\_to\_K = 10, BP = 0,5, Age = 50.thoả mãn thì chúng ta đều thu được dự báo có nhãn là 1. Điều này cho thấy dự báo từ cây nhị phân đã tuân theo quy luật rẽ nhánh in đậm như bên trên

## 3.4 Đường biên phân chia của cây quyết định

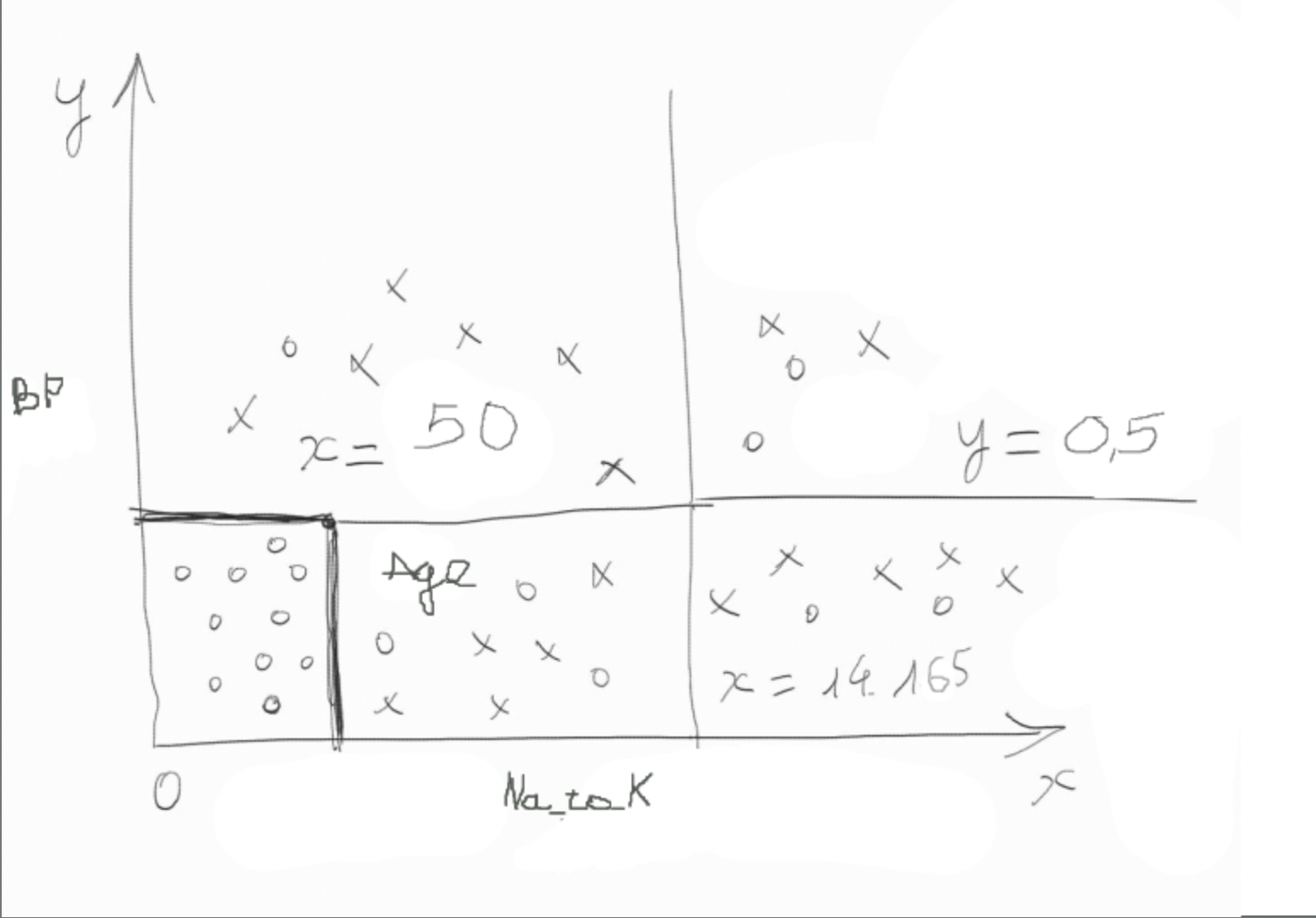
Đường biên phân chia của cây quyết định sẽ dựa trên kịch bản rẽ nhánh mà chúng ta lựa chọn. Gỉa định chúng ta đi từ node (Na\_to\_K <= 14.165) --> (BP <= 0,5) --> (Age <= 50). Khi đó đường biên phân chia là những đường thẳng trong hình bên dưới đi qua ngưỡng threshold.

Bước 1: Đường thẳng x=14,165 sẽ phân mặt phẳng thành 2 phần là x=<14,165 và x> 14,165. Theo phương án rẽ nhánh tại node gốc tương ứng với biến Na\_to\_K, chúng ta lựa chọn nửa mặt phẳng x=<14,165 nằm bên trái để dự báo nhãn.

Bước 2: Đối với nửa mặt phẳng x=<14,165 chúng ta lại xét tiếp trục y tương ứng với biến BP. Đường thẳng y=0,5 sẽ tiếp tục chia nửa mặt phẳng này thành hai phần là y>0,5 và y=< 0,5 Theo kịch bản chúng ta sẽ lựa chọn hình chữ nhật bị giới hạn bởi x>= 14,165 và y >= 0,5. Lúc này cây quyết định vẫn chưa kết thúc. Do mô hình có độ sâu là 3 nên chúng ta phải tiếp tục phân chia tiếp theo threshold của biến Age.

Đường thẳng x= 50 sẽ tiếp tục phân chia hình chữ nhật thu được ở bước 2 thành hai hình chữ nhật con. Ta chỉ lấy hình chữ nhật nằm bên trái. Đó chính là hình chữ nhật bị giới hạn bởi các cạnh in đậm.

Như vậy đường biên phân chia của cây quyết định là khá đơn giản và trực quan. Chúng ta không cần giải bài toán tối ưu như SVM hay Logistic để tìm ra được đường biên phân chia này mà đơn thuần chúng chỉ là một siêu phẳng đi qua một ngưỡng threshold cố định. Những mô hình có đường biên phân chia có thể giải thích được như vậy được gọi là white box models. Trái lại, trong machine learning tồn tại những mô hình mà chúng ta không thể giải thích được vì sao mô hình lại đưa ra quyết định như vậy.



## 3.5 Tìm kiếm tham lam & thuật toán ID3, CART

Giả sử số lượng biến của bạn là n rất lớn. Bạn muốn tạo ra một cây nhị phân với độ sâu tối đa là d. Số khả năng lựa chọn d biến (có xét đến tính thứ tự) từ m biến để tạo thành một cây nhị phân là chỉnh hợp And=d!(d−n)!. Khi d lớn thì đây là một con số rất lớn. Do đó rất khó để chúng ta tìm được đúng cây nhị phân tối ưu ngay chỉ trong một lần. Thay vào đó, một chiến lược hợp lý và khôn ngoan hơn là đi từng bước nhỏ và tìm cách lựa chọn câu hỏi tối ưu ở mỗi node. Chiến lược như vậy được gọi là tìm kiếm tham lam.

Ngoài ra quá trình lựa chọn này sẽ tiếp diễn một cách truy hồi (recursive) theo chiều từ trên xuống dưới cho đến khi đạt ngưỡng về độ sâu hoặc node cuối cùng hoàn toàn thuộc về một nhóm. Ở một số thuật toán, để hạn chế hiện tượng quá khớp chúng ta có thể dừng phân chia nếu chạm ngưỡng số lượng quan sát tối thiểu ở node lá hoặc giới hạn về độ sâu của nhánh (chúng ta sẽ làm rõ hơn điều này ở phần bên dưới).

Cách xây dựng cây quyết định theo phương pháp tìm kiếm tham lam và truy hồi dựa trên thuật toán ID3 bên dưới.

Thuật toán ID3 (viết tắt của Iterative Dichotomiser 3) là một giải thuật khá lâu đời được tạo ra bởi Ross Quinlan nhằm xây dựng cây quyết định phù hợp từ một bộ dữ liệu. Đây là giải thuật tiền đề mà dựa trên cơ sở đó, rất nhiều những giải thuật khác liên quan tới cây quyết định được kế thừa và phát triển:

* C4.5: Kế thừa của thuật toán ID3. Giải thuật này được sử dụng phổ biến trong machine learning và xử lý ngôn ngữ tự nhiên.
* CART: Viết tắt của cụm từ Classification And Regression Tree. Ưu điểm của nó là có thể sử dụng cho cả bài toán phân loại và hồi quy.
* CHAID: Sử dụng phân phối
* χ2
* để tự động tương tác phát hiện phân chia khi tính toán cây phân loại.
* MARS: Áp dụng hồi quy đa biến theo splines. Đây là một phương pháp hồi quy chia để trị, có thể loại bỏ ảnh hưởng của outliers.

Trong khuôn khổ bài viết này chúng ta sẽ tìm hiểu về thuật toán CART. Đây là thuật toán được sử dụng phổ biến nhất trong machine learning và có thể được sử dụng trong cả hai bài toán phân loại và hồi quy.

## 3.6 Điều kiện dừng để giảm quá khớp (overfitting)

Nếu chúng ta tiếp tục phân chia cây quyết định liên tục thì số lượng các quan sát ở mỗi node lá sẽ giảm dần. Cho tới một ngưỡng độ sâu p nào đó, số quan sát còn lại ở mỗi node lá sẽ rất nhỏ và thậm chí chỉ một vài quan sát. Các kết quả dự báo dựa trên tập mẫu rất nhỏ này không còn mang tính tổng quát và do đó hiện tượng quá khớp thường xảy ra. Để tránh hiện tượng quá khớp cũng như tiết kiệm chi phí tính toán, chúng ta sẽ dừng việc phân chia khi đạt một số điều kiện:

* Độ sâu của cây nhị phân chạm một ngưỡng tối thiểu.
* Số lượng các quan sát của một node lá đạt ngưỡng tối thiểu. Chẳng hạn như: 30 quan sát thuộc node lá cho bài toán phân loại nhị phân thì quyết định phân lớp là đủ tin cậy.
* Node lá hoàn toàn thuộc về một nhóm duy nhất. Tức node phân chia là hoàn toàn tinh khiết.
* Số lượng các node phân chia đạt ngưỡng.
* Số lượng các node lá đạt ngưỡng. Số lượng node lá càng nhiều thì mô hình càng trở nên phức tạp.
* hàm tin thu giảm dưới một ngưỡng rất nhỏ. Đồng nghĩa với việc phân chia thêm cũng không có nhiều ý nghĩa.

Ngoài các phương pháp giảm thiểu quá khớp nêu trên chúng ta còn có thể giảm thiểu quá khớp thông qua phương pháp cắt tỉa (pruning).