|  |  |
| --- | --- |
| **logo moi** | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**BÁO CÁO TRÍ TUỆ NHÂN TẠO**

**TÌM HIỂU VỀ RANDOM FOREST VÀ ÁP DỤNG VÀO BÀI TOÁN DỰ ĐOÁN KHẢ NĂNG MẮC BỆNH TIM MẠCH**

Ngành: **CÔNG NGHỆ THÔNG TIN**

***Nhóm sinh viên thực hiện:***

**Tên** **Lớp** **MSSV**

Nguyễn Thanh Siêu 21DTHE4 2180609116

Trần Việt Đức 21DTHE4 2180606946

Đặng Lê Ngọc Trạng 21DTHE4 2180608126

Giảng viên hướng dẫn: **TS. Vũ Thanh Hiền**

TP. Hồ Chí Minh, 2023

# **LỜI MỞ ĐẦU**

Trong những năm gần đây, các lĩnh vực nghiên cứu ngành Khoa Học Máy Tính phát triển hết sức mạnh mẽ, nhiều thuật toán ra đời với nhiều hướng nghiên cứu khác nhau. Trong đó Mechine Learning là một hướng nghiên cứu đã xuất hiện từ những năm 1959-1960 và đạt được nhiều thành tựu. Nhiều thuật toán học máy được ứng dụng trong thực tế và mang lại hiệu quả trong một số lĩnh vực như: nhận diện hình ảnh/ khuôn mặt, tự động nhận diện giọng nói, chữ viết, lĩnh vực tài chính – ngân hàng, chăm sóc sức khỏe,… Các thuật toán được dùng phổ biến như: Decision tree, mạng Nơ-ron nhân tạo, K-Mean, Random Forest,… Mỗi thuật toán đều có một số tham số và các tham số này ảnh hưởng rất lớn đến kết quả của thuật toán, vì vậy việc tối ưu các tham số là vô cùng cần thiết,

Trong đồ án này, mục tiêu của nhóm chúng tôi là tìm hiểu về Random Forest và áp dụng vào bài toán cụ thể “Bài toán dự đoán khả năng mắc bệnh tim mạch”. Đồ án tập trung về lịch sử, cơ sở lý thuyết, tham số và siêu tham số trong Random Forest và cách điều chỉnh nó, ưu và nhược điểm của thuật toán Random Forest, cùng một số ứng dụng thực tế của thuật toán Random Forest và cuối cùng là giải thuật của bài toán liên quan về Random Forest – bài toán dự đoán khả năng mắc bệnh tim mạch.

Tất cả những tham khảo từ các nghiên cứu liên quan đều được nêu nguồn gốc rõ ràng từ danh mục tài liệu tham khảo của đồ án. Trong đồ án , không có việc sao chép tài liệu, công trình nghiên cứu của người khác mà không chỉ rõ về tài liệu tham khảo. Mọi sao chép không hợp lệ chúng tôi xin cam đoan sẽ chịu toàn bộ trách nhiệm.

# **LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin chân thành cảm ơn TS.Vũ Thanh Hiền hướng dẫn đồ án môn học cho chúng em.

Sau thời gian học tập và tìm hiểu dưới sự hướng dẫn tận tình của thầy, chúng em đã rút ra được nhiều kinh nghiệm quý báu về khả năng phân tích và thiết kế cũng như kĩ năng làm việc mà không chỉ đơn giản là đọc trong sách, vở có thể có được và một lần nữa chúng em xin gửi lời cảm ơn sâu sắc đến Thầy đã dạy bảo và hướng dẫn những kiến thức chuyên môn cần có để chúng em áp dụng tốt nhất những gì đã được học trong suốt thời gian qua.

Trong quá trình thực hiện và làm báo cáo, do còn thiếu nhiều kinh nghiệm thực tế không tránh được những sai sót. Chúng em rất mong nhận được những ý kiến đóng góp của Thầy để giúp đỡ chúng em trong kĩnh vững này được hoàn thiện hơn. Đó là hành trang quý giá giúp chúng em hoàn thiện kiến thức của mình trong chặng đường sau này.

Chúng em xin chân thành cảm ơn và trân trọng kính chào!

# **NHẬN XÉT & ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN**

Qua quá trình học tập:

Giáo viên hướng dẫn có một số nhận xét, đánh giá như sau:

1/- Quá trình học tập

2/- Thực hiện báo cáo

Đánh giá chung

Kết quả đạt được:

Điểm đánh giá việc thực hiện báo cáo /10

*TP. HCM, Ngày tháng năm 20*

**GIẢNG VIÊN HƯỚNG DẪN**

**TS. Vũ Thanh Hiền**

# **MỤC LỤC**

[**LỜI MỞ ĐẦU** 2](#_Toc154271843)

[**LỜI CẢM ƠN** 3](#_Toc154271844)

[**NHẬN XÉT & ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN** 4](#_Toc154271845)

[**MỤC LỤC** 5](#_Toc154271846)

[**DANH MỤC HÌNH ẢNH** 6](#_Toc154271847)

[**CHƯƠNG 1: TỔNG QUAN** 7](#_Toc154271848)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** 8](#_Toc154271849)

[**CHƯƠNG 3: BÀI TOÁN DỰ ĐOÁN KHẢ NĂNG MẮC BỆNH TIM MẠCH** 9](#_Toc154271850)

[**CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 10](#_Toc154271851)

[**TÀI LIỆU THAM KHẢO** 11](#_Toc154271852)

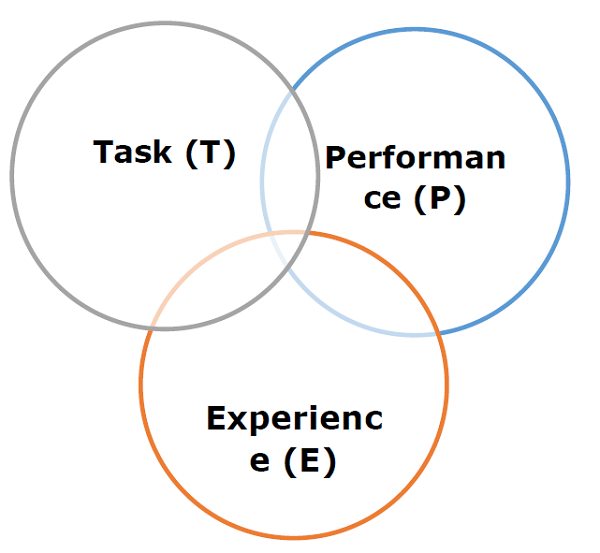
# **DANH MỤC HÌNH ẢNH**

# **CHƯƠNG 1: TỔNG QUAN**

1. **Giới thiệu về Machine Learning**

Chúng ta đang sống trong “thời đại dữ liệu” được làm giàu với sức mạnh tính toán tốt hơn với nhiều tài nguyên lưu trữ hơn. Dữ liệu thông tin này đang tăng lên từng ngày, nhưng thách thức thực sự là hiểu được tất cả dữ liệu. Các doanh nghiệp và tổ chức đang cố gắng giải quyết vấn đề này bằng cách xây dựng các hệ thống thông minh sử dụng các khái niệm và phương pháp luận từ Khoa học dữ liệu, Khai thác dữ liệu và Máy học.

Machine Learning là ngành khoa học máy tính nhằm nghiên cứu, phát triển các thuật toán, thuật giải với các mục đích đưa tri thức vào máy tính, cụ thể hơn là những thuật giải dựa trên các tập dữ liệu và rút ra các quy luật từ chúng, làm cho máy tính có thể giải được các bài toán mà các thuật toán bình thường khó có thể thực hiện như tìm kiếm, nhận dạng, dự đoán. Các thuật toán học máy được phân loại theo kết quả thuật toán. Các loại thuật toán thường được sử dụng như: học có giám sát (supervised learning), học không giám sát (unsupervised learning), học bán giám sát (semi-supervidsed learning), học tặng cường(reinforcement learning), học tiến hóa(evolutionary learning),… Machine Learning đang ngày càng phát triển, ngoài những thuật toán ra đời sớm thì sau này có nhiều thuật toán ra đời nhu: Mạng nơ-ron nhân tạo, Mạng Bayes, Học bằng quy trình Markov, Decision tree, K-Mean, SVM, Random Forest, Học luật bằng quy nạp,…



Hình 1.1. Mô hình học máy

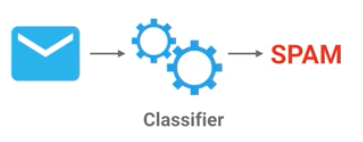
Machine Learning hiện nay được áp dụng rộng rãi và trong nhiều lĩnh vực khác nhau bao gồm máy truy tìm dữ liệu, chuẩn đoán y khoa, phát hiện thẻ tín dụng giả, phân tích thị trường chứng khoán, phân loại chuỗi DNA, nhận dạng tiếng nói và chữ vieetsm dịch tự động, trò chơi điện tử, cử động rô-bốt (robot locomotion).

Thuật toán Random Forest là một trong những thuật toán Machine Learning ra đời muộn hơn các thuật toán Machine Learning khác, chính vì vậy thuật toán Random Forest kế thừa được ưu điểm của các thuật toán khác đồng thời khắc phục được những hạn chế về mặt số lượng dữ liệu cũng như độ phức tạp của dữ liệu.

1. **Giới thiệu về Classification**

Phân loại trong học máy và thống kê là một trong phương pháp học có giám sát (supervised learning), trong đó chương trình máy tính học từ dữ liệu cung cấp cho nó và thực hiện các quan sát hoặc phân loại mới. Nó là quá trình phân loại một tập hợp dữ liệu nhất định thành các lớp. Nó có thể được thực hiện trên cả dữ liệu có cấu trúc và không có cấu trúc. Quá trình bắt đầu với việc dự đoán class/category của các điểm dữ liệu đã cho. Các lớp thường được gọi là target, lable hoặc categories.

Các vấn đề phân loại phổ biến nhất là: Nhận dạng giọng nói, nhận diện khuôn mặt, nhận dạng chữ viết tay, phát hiện email spam, phân loại tài liệu,… Nó có thể là bài toán phân loại nhị phân hoặc bài toán nhiều lớp.



Hình 1.2. Phân lớp nhận dạng email spam

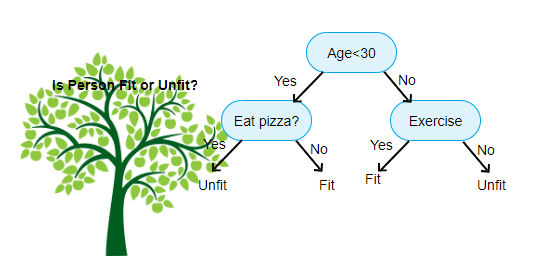
Có một loạt các thuật toán học máy để phân loại trong trong học máy: Naïve Bayes Classifier, Stochastic Gradient Descent, K-Nearest Neighbor, Decision Tree, Random forest,… Mỗi loại thuật toán sẽ có ưu nhược điểm riêng nhưng trong đồ án này chúng ta sẽ tập trung vào Random Forest sẽ được đề cập chương 2 và chương 3.

1. **Giới thiệu sơ lược Decision Tree**

Decision Tree là một trong những công cụ mạnh mẽ nhất của các thuật toán học tập có giám sát. Việc huấn luyện Decision Tree và sử dụng nó như một mô hình dự đoán, ánh xạ từ các quan sát về một sự vật/hiện tượng với các kết luận về giá trị mục tiêu của sự vật/hiện tượng.

Decision Tree có hai loại chính:

* Cây phân loại (Classification Tree): nếu y là một biến phân loại như: giới tính (nam hay nữ), kết quả một trận đấu (thắng hay thua).
* Cây hồi quy (Regression tree): ước lượng hàm đó có giá trị là số thực thay vì được sử dụng cho các nhiệm vụ phân loại. Ví dụ ước tính giá của một ngôi nhà hoặc khoảng thời gian một bệnh nhân nằm viện.



Hình 1.3. Ảnh minh họa Decision Tree

Ở chương 2, chúng ta sẽ xem thử Decision Tree có vai trò quan trọng như thế nào trong thuật toán Random Forest.

1. **Giới thiệu về thuật toán Random Forest**

Random Forest hay rừng ngẫu nhiên là một thuật toán học có giám sát được sử dụng cho cả phân loại cũng như hồi quy. Tuy nhiên, nó chủ yếu được sử dụng cho các bài toán phân loại. Như chúng ta đã biết một khu rừng được tạo thành từ cây cối và nhiều cây cối hơn có nghĩa là khu rừng vững chắc hơn. Tương tự, thuật toán Random Forest được tạo ra từ các cây quyết định trên các mẫu dữ liệu sau đó lấy dự đoán từ mỗi người trong số chúng và cuối cùng chọn giải pháp tốt nhất bằng cách bỏ phiếu. Đây là một phương pháp tổng hợp tốt hơn một cây quyết định đơn lẻ vì nó giảm sự phù hợp quá mức bằng cách lấy trung bình kết quả.



Hình 1.4. Hình ảnh giới thiệu về thuật toán Random Forest

Thuật toán đầu tiên cho các khu rừng ngẫu nhiên được tạo ra bởi Tin Kam Ho bằng cách sử dụng phương pháp không gian con ngẫu nhiên, theo công thức của Ho là một cách để thực hiện phương pháp “phân biệt ngẫu nhiên” để phân loại do Eugene Kleinberg đề xuất. Một phần mở rộng của thuật toán là được phát triển bởi Leo Breiman và Adele Cutler, người đã đăng kí “Random Forest” làm nhãn hiệu (kể từ năm 2019, thuộc sở hữu của Minitab, Inc). Phần mở rộng ý tưởng “bagging (đóng gói)” của Breiman và lựa chọn ngẫu nhiên các tính năng được Ho giới thiệu lần đầu tiên và sau là độc lập bởi Amit và German để xây dụng một tập hợp các Decision Tree với phương sai có kiểm soát.

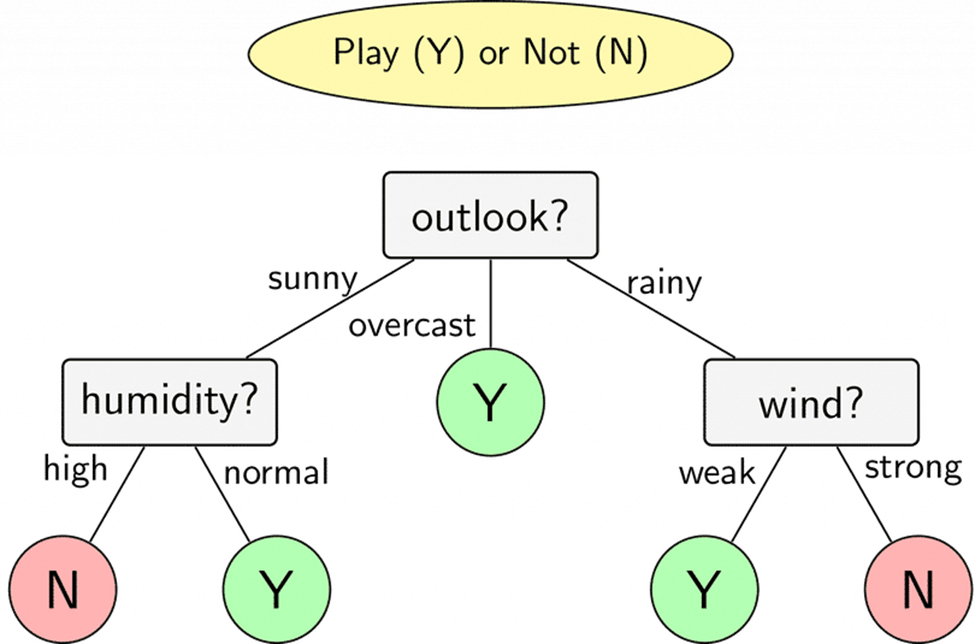
Random Forest được mô hình hóa như tập các cây phân lớp. Tuy nhiên Random Forest sử dụng các mẫu ngẫu nhiên cho các cây cũng như việc chọn lựa các thuộc tính ngẫu nhiên khi phân chia cây. Thuật toán Random Forest tỏ ra chính xác và nhanh hơn khi huấn luyện trên không gian dữ liệu lớn với nhiều thuộc tính, việc sử dụng kết quả dự đoán của tất cả các cây trong rừng khi phân lớp hoặc hồi quy giúp cho kết quả thuật toán chính xác hơn. Chúng ta sẽ tìm hiểu kĩ hơn về Random Forest ở chương 2.

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

1. **Decision Tree**
2. **Định nghĩa**

Trong lĩnh vực máy học, Decision Tree là một kiểu mô hình dự báo (predictive model), nghĩa là một ánh xạ từ các quan sát về một sự vật/hiện tượng tới các kết luận về giá trị mục tiêu của sự vật hiện tượng.

Decision Tree là mô hình Supervised Learning, có thể được áp dụng vào cả hai bài toán Classification và Regression. Mỗi một nút trong (internal node) tương ứng với một biến, đường nối giữa nó với con của nó thể hiện một giá trị cụ thể cho biến đó. Mỗi nút lá đại diện cho giá trị dự đoán của biến mục tiêu, cho trước các giá trị của biến được biểu diễn đường đi từ nút gốc tới nút lá đó. Kỹ thuật học máy dùng trong cây quyết định được gọi là học bằng cây quyết đinh, hay chỉ với cái tên ngắn gọn là cây quyết định.

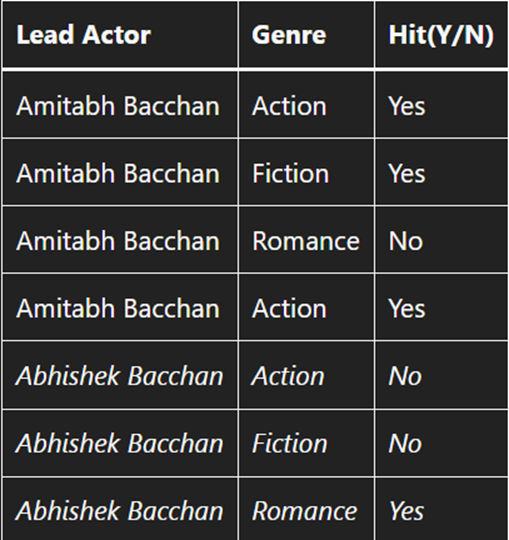


Hình 2.1. Mô hình Decision Tree quyết định đi chơi

1. **Một số thuật toán xây dựng Decision Tree**

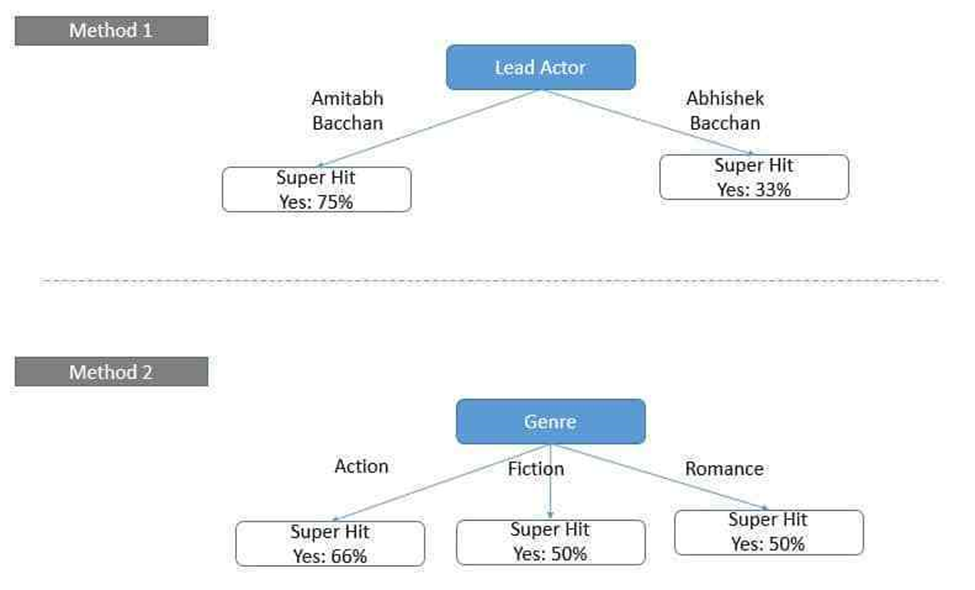
Thuật toán ID3(J.R.Quinlan 1933) sử dụng phương pháp tham lam tìm kiếm từ trên xuống thông qua không gian của các nhánh có thể không có backtracking. ID3 sử dụng Entropy và Information Gain để xây dựng Decision Tree.

Ví dụ: Bạn muốn xem xét sự thành công của một bộ phim thông qua hai yếu tố: Diễn viên chính của phim và thể loại phim.



Hình 2.2. Bảng quan sát sự thành công của một bộ phim

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



Hình 2.3. Mô hình thể hiện kết quả thành công của bộ phim

Qua sơ đồ trên ta thấy rõ, với phương pháp thứ nhất, ta phân loại được rõ ràng, trong khi phương pháp hai 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 cây quyết định sử dụng phân chia. Dưới đây là hai hệ số phổ biến là Information Gain và Gain Ratio (ngoài ra còn hệ số Gini)

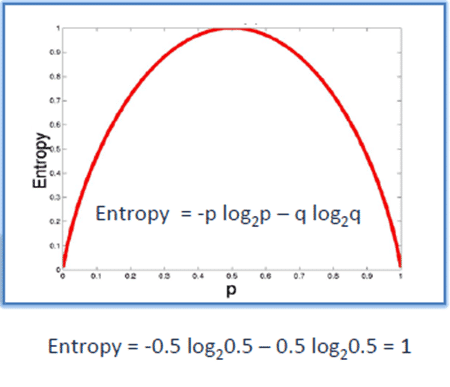
Entropy trong Decision Tree:

* Số lượng mong đợi các bit cần thiết để mã hóa thông tin về lớp của một thành viên rút ra một cách ngẫu nhiên từ tập S
* Trong trường hợp tối ưu, mã có độ dài ngắn nhất
* Trong lý thuyết thông tin, mã có độ dài tối ưu là mã gán (-log2p) bit cho thông điệp (message) có xác suất là p
* Trong trường hợp S là tập mẫu dữ liệu thì thành viên của S là một mẫu dữ liệu, mỗi mẫu dữ liệu thuộc một lớp có một giá trị phân lớp.
* Entropy có giá trị nằm trong đoạn [0,1]
  + Entropy (S) = 0 ⬄ tập mẫu S chỉ toàn mẫu thuộc cùng lớp hoặc S là thuần nhất
  + Entropy(S) = 1 ⬄ tập mẫu S có các mẫu thuộc các lớp khác nhau với độ pha trộn là cao nhất
  + 0 < Entropy(S) < 1 ⬄ tập mẫu S có số lượng mẫu thuộc các lớp khác nhau là không bằng nhau

Entropy cần để phân lớp một mẫu trong CSDL D là:

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

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



Hình 2.4. Hình vẽ biểu diễn sự thay đổi của hàm Entropy

Ta có thể thấy rằng, 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 đạt đỉnh cao nhất

Information Gain trong Decision Tree:

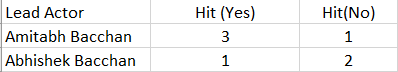
* Thuộc tính A gồm các giá trị {a1, a2,…,av}, và phân chia D thành v tập con {D1, D2,…,Dv}. Thông tin cần thiết để chia D theo A.
* Information Gain dựa trên phân chia theo thuộc tính A của tập D

Gain(A) = Info(D) – InfoA(D)

* Chọn thuộc tính với Information Gain lớn nhất là thuộc tính phân hạch

Với ví dụ về thành công của một bộ phim trên, ta tính được hệ số Entropy như sau:

* + - Trong 7 mẫu có 4 mẫu “Yes” và 3 mẫu “No”



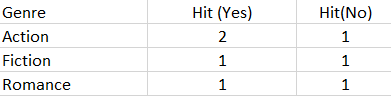
* + - * |D| = 7, n = 2, C1 = “Yes”, C2 = “No”
      * |C1,D| = 4, |C2,D| = 3
      * Entropy để phân lớp một mẫu trong D là

Info(D) = log2 log2 ≈ 0.99

* + - Tính Information Gain đối với “Lead Actor”
      * InfoLead Actor(D) = log2 log2 ) +

log2 log2 ) ≈ 0.69

* + - * Gain (Lead Actor) = Info(D) – InfoLead Actor(D) ≈ 0.99 -0.69 ≈ 0.3
    - Tính Information Gain đối với “Genre”



* + - * InfoGenre(D) = log2 log2 ) +

log2 log2 ) +

log2 log2 ) ≈ 0.55

* + - * GainGenre(D) = Info(D) - InfoGenre = 0.99 – 0.55 ≈ 0.44

🡺Chọn “Genre” làm thuộc tính phân hạch và được cây quyết định

C4.5 là thuật toán cải tiến ID3

* Độ đo Information Gain có xu hướng thiên vị cho các thuộc tính có nhiều giá trị
  + Một số trường hợp các mẫu chia thuần nhất và không có ích cho việc phân lớp
  + Cần chuẩn hóa độ đo Information Gain
* C4.5 sử dụng độ đo Gain Ratio để khắc phục vấn đề của ID3

GainRatio(A) = với

* Chọn thuộc tính với độ đo Gain Ratio lớn nhất là thuộc tính phân hoạch áp dụng với ví dụ trên ta có

SplitInfoGenre(D) = ≈ 0.98

* GainRatio(Lead Actor) = ≈ 0.58

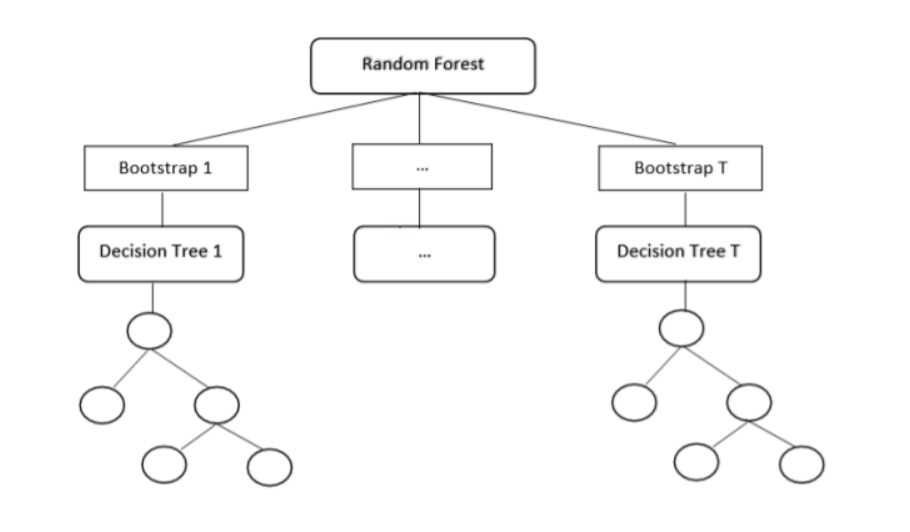
1. **Đặc điểm của Decision Tree**

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược 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 dữ liệu missing, không cần chuẩn hóa hoặc biến tạo 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 kiểm tra thống kê. * Có khả năng là việc với dữ liệu lớn. | Kèm với đó, cây quyết định cũng có những nhược điểm:   * Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu của bạn. Thâm chí với một sự thay đổi nhỏ trong toàn bộ dữ liệu, cấu trúc mô hình cây quyết định thay đổi hoàn toàn. * Cây quyết định hay gặp vấn đề overfitting. |

1. **Thuật Toán Random Forest**
2. **Định nghĩa**

Random Forests hay Random Decision Forests là thuật toán học máy dựa trên kĩ thuật lắp ghép, kết hợp với các cây phân lớp. Random Forest xây dựng cây phân lớp bằng cách lựa chọn ngẫu nhiên một nhóm nhỏ các thuộc tính tại mỗi nút của cây để phân chia cho mức tiếp theo của cây phân lớp. Ngoài ra tập mẫu của mỗi cây cũng được lựa chọn ngẫu nhiên bằng phương pháp Bootstrap từ tập mẫu ban đầu. Số lượng các cây phân lớp trong rừng là không hạn chế và thuật toán sử dụng kết quả dự đoán của tất cả cây trong rừng làm kết quả cuối cùng của thuật toán.

1. **Mô hình thuật toán**



Hình 2.5 Mô hình thuật toán Random Forest

Các kí hiệu:

Random Forest

* + Rừng ngẫu nhiên: Tập cây phân lớp

Bootstrap i

* + Phương pháp Bootstrap tạo tập huấn luyện cho cây thứ *i*

Decision Tree i

* + Cây phân lớp thứ *i* trong rừng
  + Nút trong cây phân lớp

Thuật toán Random Forest được xây dựng từ nhiều Decision Tree, tuy nhiên mỗi Decision Tree sẽ khác nhau. Sau đó, kết quả dự đoán sẽ được tổng hợp từ những kết quả Decision Tree đó.

Quá trình hoạt động cơ bản:

* Tạo các tập con dữ liệu: Sử dụng Bootstrap Aggregating để tạo ra các tập con dữ liệu từ tập dữ liệu huấn luyện.
* Xây dựng cây quyết định: Với mỗi tập con dữ liệu được tạo ra, một Decision Tree được xây dựng.
* Sử dụng phương pháp voting: Mỗi cây quyết định được gán một trọng số dựa trên độ chính xác của nó trên tập dữ liệu huấn luyện. Random Forest sử dụng voting để đưa ra dự đoán cuối cùng dựa vào kết quả tính trung bình của trọng số.

1. **Thuật toán**
2. **Ví dụ minh họa-**
3. **Đặc điểm của Random Forest**
4. **Ứng dụng của thuật toán Random Forest**
5. **Phương pháp lấy mẫu Bootstrap**
6. **Giới thiệu Bootstrap**
7. **Thuật toán Bootstrap**
8. **Đặc điểm của Bootstrap**
9. **Bootstrap trong Random Forest**
10. **Xây dựng một quá trình phân loại (classification) bằng cách sử dụng ͌với Random Forest Classifier.**
11. **Tham số và siêu tham số trong model Random Forest**

# **CHƯƠNG 3: BÀI TOÁN DỰ ĐOÁN KHẢ NĂNG MẮC BỆNH TIM MẠCH**

# **CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

# **TÀI LIỆU THAM KHẢO**

# References

|  |  |
| --- | --- |
| [1] | "wikipedia," 23 20 2023. [Online]. Available: https://vi.wikipedia.org/wiki/H%E1%BB%8Dc\_m%C3%A1y. [Accessed 15 12 2023]. |
| [2] | T. V. Hào, "123docz.net," [Online]. Available: https://123docz.net/document/9898782-bao-cao-mon-lap-trinh-python-cho-may-hoc-random-forest-classifier.htm. [Accessed 1 12 2023]. |
| [3] | T. Mai, "TinoGroup," [Online]. Available: https://tino.org/vi/machine-learning-la-gi/. [Accessed 24 12 2023]. |
| [4] | "wikipedia," 3 1 2023. [Online]. Available: https://vi.wikipedia.org/wiki/C%C3%A2y\_quy%E1%BA%BFt\_%C4%91%E1%BB%8Bnh. [Accessed 24 12 2023]. |
| [5] | "iSolution," [Online]. Available: https://isolution.pro/vi/t/machine-learning-with-python/classification-algorithms-random-forest/thuat-toan-phan-loai-rung-ngau-nhien. [Accessed 24 12 2023]. |
| [6] | "Trí Tuệ Nhân Tạo.io," 06 06 2019. [Online]. Available: https://trituenhantao.io/kien-thuc/decision-tree/. [Accessed 25 12 2023]. |