

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

--------------------------------------

BÁO CÁO MÔN HỌC

**KHO DỮ LIỆU VÀ KHAI PHÁ DỮ LIỆU**

**Đề tài: Áp dụng thuật toán Decision Tree, Random Forest**

**và Logistic Regression vào việc đánh giá dự đoán bệnh ưng thư vú**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | Nguyễn Quỳnh Chi |
| **Nhóm lớp** | 04 |
| **Nhóm BTL** | 10 |
| **Sinh viên thực hiện:** | Nguyễn Công Hiệp – B20DCCN239 |
|  | Đỗ Nguyên Vũ – B20DCCN039 |
|  | Trần Văn Việt – B20DCCN733 |
|  | Lê Thành Trung – B20DCCN697 |
|  | Phạm Duy Hùng – B20DCCN299 |
|  | Ngô Đức Minh – B20DCCN434 |

**Hà Nội 2024**

**Lời mở đầu**

Trong thời đại ngày nay, với sự phát triển vượt bậc của công nghệ thông tin, các hệ thống thông tin có thể lưu trữ một khối lượng lớn dữ liệu về hoạt động hàng ngày của chúng. Từ khối dữ liệu này, các kỹ thuật trong Khai Phá Dữ Liệu và Máy Học có thể dùng để trích xuất những thông tin hữu ích mà chúng ta chưa biết. Các tri thức vừa học được có thể vận dụng để cải thiện hiệu quả hoạt động của hệ thống thông tin ban đầu.

Khai phá dữ liệu thường được xem là việc khám phá tri thức trong các cơ sở dữ liệu, là một quá trình trích xuất những thông tin ẩn, trước đây chưa biết và có khả năng hữu ích, dưới dạng các quy luật, ràng buộc, quy tắc trong cơ sở dữ liệu. Nói tóm lại, khai phá dữ liệu là một quá trình học tri thức mới từ những dữ liệu đã thu thập được.

Kỹ thuật khám phá tri thức và khai phá dữ liệu đã và đang được nghiên cứu, ứng dụng trong nhiều lĩnh vực khác nhau ở các nước trên thế giới, tại Việt Nam kỹ thuật này tương đối còn mới mẻ tuy nhiên cũng đang được nghiên cứu và dần đưa vào ứng dụng.

Khai phá dữ liệu (Data Mining) được coi là quá trình trích xuất các thông tin có giá trị tiềm ẩn bên trong lượng lớn dữ liệu được lưu trữ trong các CSDL, kho dữ liệu…

**MỤC LỤC**

[A. Đặt vấn đề 2](#_Toc167217547)

[1. Phân chia công việc 2](#_Toc167217548)

[2. Bài toán đặt ra 2](#_Toc167217549)

[B. Nội dung 2](#_Toc167217550)

[I. GIỚI THIỆU MÔ HÌNH 2](#_Toc167217551)

[1. Decision Tree 2](#_Toc167217552)

[2. Random Forest 3](#_Toc167217553)

[3. Logistic Regression 4](#_Toc167217554)

[4. Phương pháp giải quyết 4](#_Toc167217555)

[II. NÊU VỀ NGUỒN GỐC DỮ LIỆU 5](#_Toc167217556)

[1. Thu thập dữ liệu: 5](#_Toc167217557)

[2. Mô tả dữ liệu: 6](#_Toc167217558)

[3. Tiền xử lí dữ liệu 6](#_Toc167217559)

[III. QUÁ TRÌNH XỬ LÝ THUẬT TOÁN DECISION TREE 7](#_Toc167217560)

[1. Tổng quan về decision tree 7](#_Toc167217561)

[2. Thuật toán ID3 9](#_Toc167217562)

[3. Thuật toán C4.5 13](#_Toc167217563)

[4. Thuật toán CART 15](#_Toc167217564)

[5. Overfitting 18](#_Toc167217565)

[6. Cắt tỉa cây 19](#_Toc167217566)

[IV. QUÁ TRÌNH XỬ LÝ THUẬT TOÁN RANDOM FOREST 21](#_Toc167217567)

[1. Random Forest 21](#_Toc167217568)

[2. Ưu điểm của Random Forest 24](#_Toc167217569)

[3. Nhược điểm của Random Forest 24](#_Toc167217570)

[4. Xây dựng thuật toán Random Forest 25](#_Toc167217571)

[V. QUÁ TRÌNH XỬ LÝ THUẬT TOÁN LOGISTIC REGRESSION 27](#_Toc167217572)

[1. Mô hình Logistic Regression 27](#_Toc167217573)

[2. Các bước của Thuật toán Logistic Regression 29](#_Toc167217574)

[3. Code thuật toán 30](#_Toc167217575)

[VI. ĐÁNH GIÁ HIỆU QUẢ THUẬT TOÁN 32](#_Toc167217576)

[1. Các tiêu chí đánh giá 32](#_Toc167217577)

[2. Đánh giá 3 thuật toán 33](#_Toc167217578)

[3. Kết luận 36](#_Toc167217579)

1. **Đặt vấn đề**
2. Phân chia công việc

|  |  |  |
| --- | --- | --- |
| 1 | - Giới thiệu về mô hình Decision Tree, Random Forest, Logistic Regression (phân loại, phạm vi, bài toán sử dụng) - Giới thiệu về bài toán đang được đặt ra - Phương pháp giải quyết cho bài toán này | Hùng |
| 2 | - Phân chia công việc, quản lí tiến độ báo cáo của nhóm, chỉnh sửa báo cáo.  - Nêu về nguồn gốc dữ liệu  - Mô tả các đặc điểm, các cột dữ liệu của dataset | Hiệp  (Nhóm trưởng) |
| 3 | - Nêu các bước tiền xử lý - Code phần tiền xử lý |
| 4 | - Nêu quá trình xử lý thuật toán Logistic Regression  - Code triển khai thuật toán Logistic Regression - So sánh Sklearn | Vũ |
| 5 | - Nêu quá trình xử lý thuật toán Decision Tree - Code triển khai thuật toán Decision Tree  - So sánh Sklearn | Việt |
| 6 | - Nêu quá trình xử lý thuật toán Random Forest - Code triển khai thuật toán Random Forest  - So sánh Sklearn | Trung |
| 7 | - Viết phần đánh giá hiệu quả của thuật toán, nêu các yếu tố để đánh giá thuật toán  - Code phần đánh giá thuật toán đó. | Minh |
| 8 | - Làm slide báo cáo. - Làm doc báo cáo. | Việt + Vũ |
| 9 | - Đánh giá thuật toán, đưa ra kết luận | Tất cả thành viên |

1. Bài toán đặt ra

Ung thư vú là một trong những loại ung thư phổ biến nhất ảnh hưởng đến phụ nữ trên toàn thế giới. Việc chẩn đoán và điều trị sớm ung thư vú có thể cải thiện đáng kể tỷ lệ sống sót và chất lượng cuộc sống của bệnh nhân. Hiện nay bệnh ưng thư đã và đang là vấn đề mà mọi người quan tâm bởi vì nó sẽ ảnh hưởng trực tiếp đến con người. Yêu cầu đặt ra là phân tích và đánh giá chuẩn đoán tỉ lệ mắc bệnh ung thư vũ

1. **Nội dung**

## GIỚI THIỆU MÔ HÌNH

### Decision Tree

Cây quyết định 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. Mỗi một nút trong (*internal node*) tương ứng với một biến; đường nối giữa nó với nút 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 các biến được biểu diễn bởi 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 định, hay chỉ gọi với cái tên ngắn gọn là cây quyết định.

Học bằng cây quyết định cũng là một phương pháp thông dụng trong [khai phá dữ liệu](https://vi.wikipedia.org/wiki/Khai_ph%C3%A1_d%E1%BB%AF_li%E1%BB%87u). Khi đó, cây quyết định mô tả một cấu trúc cây, trong đó, các lá đại diện cho các phân loại còn cành đại diện cho các kết hợp của các thuộc tính dẫn tới phân loại đó. Một cây quyết định có thể được học bằng cách chia [tập hợp](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_h%E1%BB%A3p) nguồn thành các tập con dựa theo một kiểm tra giá trị thuộc tính. Quá trình này được lặp lại một cách đệ quy cho mỗi tập con dẫn xuất. Quá trình [đệ quy](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%87_quy) hoàn thành khi không thể tiếp tục thực hiện việc chia tách được nữa, hay khi một phân loại đơn có thể áp dụng cho từng phần tử của tập con dẫn xuất. Một bộ phân loại [rừng ngẫu nhiên](https://vi.wikipedia.org/w/index.php?title=R%E1%BB%ABng_ng%E1%BA%ABu_nhi%C3%AAn&action=edit&redlink=1) (*random forest*) sử dụng một số cây quyết định để có thể cải thiện tỉ lệ phân loại.

Cây quyết định cũng là một phương tiện có tính mô tả dành cho việc tính toán các [xác suất có điều kiện](https://vi.wikipedia.org/wiki/X%C3%A1c_su%E1%BA%A5t_c%C3%B3_%C4%91i%E1%BB%81u_ki%E1%BB%87n).

Cây quyết định có thể được mô tả như là sự kết hợp của các kỹ thuật toán học và tính toán nhằm hỗ trợ việc mô tả, phân loại và tổng quát hóa một tập dữ liệu cho trước.

Dữ liệu được cho dưới dạng: .

Biến phụ thuộc (*dependant variable*) *y* là biến mà chúng ta cần tìm hiểu, phân loại hay tổng quát hóa. *x1*, *x2*, *x3*... là các biến sẽ giúp ta thực hiện công việc đó.

* Phân loại cây quyết định:
  + Cây hồi quy (*Regression tree*) ước lượng các hàm giá có giá trị là [số thực](https://vi.wikipedia.org/wiki/S%E1%BB%91_th%E1%BB%B1c) thay vì được sử dụng cho các nhiệm vụ phân loại. (ví dụ: ước tính giá một ngôi nhà hoặc khoảng thời gian một bệnh nhân nằm viện).
  + Cây phân loại (*Classification tree*), nếu *y* là một biến phân loại như: giới tính (nam hay nữ), kết quả của một trận đấu (thắng hay thua).

### Random Forest

Random Forest là một thuật toán học máy bộ tổ hợp (ensemble) được sử dụng cho các tác vụ phân loại, hồi quy và các tác vụ khác liên quan đến dữ liệu. Nó kết hợp nhiều cây quyết định (decision trees) để tạo ra một mô hình dự đoán chính xác hơn và ít bị quá khớp hơn.

Thuật toán Random Forest hoạt động bằng cách tạo ra một tập hợp các cây quyết định ngẫu nhiên, mỗi cây được xây dựng trên một tập dữ liệu con được lấy ra từ tập dữ liệu gốc bằng cách sử dụng phương pháp lấy mẫu ngẫu nhiên (bootstrap sampling). Khi có một điểm dữ liệu mới cần được dự đoán, Random Forest sẽ truyền nó qua tất cả các cây quyết định và tính toán kết quả bằng cách xem xét các kết quả từ mỗi cây, sau đó đưa ra kết quả dự đoán cuối cùng.

Random Forest có nhiều ưu điểm như khả năng xử lý các tập dữ liệu lớn, giảm thiểu hiện tượng quá khớp và tăng tính tổng quát của mô hình. Nó cũng có thể được sử dụng để xác định tầm quan trọng của các đặc trưng trong dữ liệu. Tuy nhiên, Random Forest cũng có một số nhược điểm như đòi hỏi thời gian và tài nguyên tính toán lớn hơn so với một cây quyết định đơn lẻ.

### Logistic Regression

Hồi quy logistic là một kỹ thuật phân tích dữ liệu sử dụng toán học để tìm ra mối quan hệ giữa hai yếu tố dữ liệu. Sau đó, kỹ thuật này sử dụng mối quan hệ đã tìm được để dự đoán giá trị của những yếu tố đó dựa trên yếu tố còn lại. Dự đoán thường cho ra một số kết quả hữu hạn, như có hoặc không.

Ví dụ: giả sử ta muốn đoán xem khách truy cập trang web của mình sẽ nhấp vào nút thanh toán trong giỏ hàng của họ hay không. Phân tích hồi quy logistic xem xét hành vi của khách truy cập trước đây, chẳng hạn như thời gian dành cho trang web và số lượng các mặt hàng trong giỏ hàng. Quá trình phân tích này xác định rằng, trước đây, nếu khách truy cập dành hơn năm phút trên trang web và thêm hơn ba mặt hàng vào giỏ hàng, họ sẽ nhấp vào nút thanh toán. Nhờ vào thông tin này, sau đó, hàm hồi quy logistic có thể dự đoán hành vi của một khách mới truy cập trang web.

Dựa trên các loại, hồi quy logistic có thể được phân thành ba loại:

* **Nhị thức:** Trong hồi quy logistic nhị thức, chỉ có thể có hai loại biến phụ thuộc, chẳng hạn như 0 hoặc 1, Đạt hoặc Thất bại, v.v.
* **Đa thức:** Trong hồi quy logistic đa thức, có thể có 3 hoặc nhiều loại biến phụ thuộc không có thứ tự, chẳng hạn như “mèo”, “chó” hoặc “cừu”
* **Thứ tự:** Trong hồi quy logistic thứ tự, có thể có 3 loại biến phụ thuộc được sắp xếp theo thứ tự trở lên, chẳng hạn như “thấp”, “Trung bình” hoặc “Cao”.

### Phương pháp giải quyết

* **Thu thập dữ liệu**: Thu thập các dữ liệu liên quan đến bệnh ung thư vú.
* **Tiền xử lý dữ liệu**: Xử lý và làm sạch dữ liệu, loại bỏ các giá trị bị khuyết (missing values) hoặc nhiễu (outliers), chuyển đổi các dữ liệu dạng chuỗi thành dạng số và chuẩn hóa dữ liệu.
* **Phân chia dữ liệu**: Phân chia dữ liệu thành tập huấn luyện và tập kiểm tra.

Xây dựng mô hình cây quyết định dựa trên dữ liệu huấn luyện và các đặc trưng bệnh ung thư vú. Điều chỉnh các siêu tham số của thuật toán để tối ưu hoá hiệu suất dự đoán.

Đánh giá hiệu suất mô hình cây quyết định: Sử dụng tập kiểm tra để đánh giá hiệu suất của mô hình cây quyết định, đo lường các chỉ số như độ chính xác (accuracy), độ phủ (recall), độ chính xác dương tính (precision), và F1-score.

* **Xây dựng mô hình Random Forest**: Sử dụng thuật toán Random Forest để xây dựng một tập hợp các cây quyết định ngẫu nhiên. Điều chỉnh các siêu tham số của thuật toán để tối ưu hoá hiệu suất dự đoán.
* **Đánh giá hiệu suất mô hình Random Forest**: Sử dụng tập kiểm tra để đánh giá hiệu suất của mô hình Random Forest, đo lường các chỉ số như độ chính xác, độ phủ, độ chính xác dương tính, và F1-score.
* **Xây dựng mô hình Ligistic Regression**: Sử dụng thuật toán Ligistic Regression, Điều chỉnh các siêu tham số của thuật toán để tối ưu hoá hiệu suất dự đoán.
* **Đánh giá hiệu suất mô hình Logistic Regression**: Sử dụng tập kiểm tra để đánh giá hiệu suất của mô hình Ligistic Regression, đo lường các chỉ số như độ chính xác, độ phủ, độ chính xác dương tính, và F1-score.
* **So sánh hiệu suất của ba mô hình**: So sánh hiệu suất của hai mô hình cây quyết định và Random Forest, Ligistic Regression để xác định mô hình nào có hiệu suất dự đoán tốt hơn cho bài toán khai phá dữ liệu phân tích nồng độ không khí.

## NÊU VỀ NGUỒN GỐC DỮ LIỆU

### Thu thập dữ liệu:

* Dữ liệu được lấy từ: https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic
* Tên dataset: Breast Cancer Wisconsin (Diagnostic)
* Dữ liệu này bao gồm các đo lường chẩn đoán từ các mẫu ung thư vú. Tập dữ liệu được lấy từ các kho dữ liệu y tế đáng tin cậy và các cơ sở dữ liệu nghiên cứu. Nó bao gồm các đặc điểm lâm sàng như:

|  |  |  |
| --- | --- | --- |
| bán kính\_trung bình | bán kính\_se | bán kính\_lớn nhất |
| độ mịn\_trung bình | độ mịn\_se | độ mịn\_lớn nhất |
| chu vi\_trung bình | chu vi\_se | chu vi\_lớn nhất |
| diện tích\_trung bình | diện tích\_se | diện tích\_lớn nhất |
| độ mịn\_trung bình | độ mịn\_se | độ mịn\_lớn nhất |
| độ nén\_trung bình | độ nén\_se | độ nén\_lớn nhất |
| lõm\_trung bình | lõm\_se | lõm\_lớn nhất |
| số lõm\_trung bình | số lõm\_se | số lõm\_lớn nhất |
| đối xứng\_trung bình | đối xứng\_se | đối xứng\_lớn nhất |
| Kích thước fractal\_trung bình | kích thước fractal\_se | kích thước fractal\_lớn nhất |

### Mô tả dữ liệu:

Chủ đề “Chuẩn đoán bệnh Ung thư vú”: Ung thư vú là loại ung thư phổ biến nhất ở phụ nữ trên toàn thế giới. Nó chiếm 25% tổng số ca ung thư và ảnh hưởng đến hơn 2,1 triệu người chỉ trong năm 2015. Ung thư vú bắt đầu khi các tế bào trong vùng vú bắt đầu phát triển không kiểm soát. Những tế bào này thường tạo thành khối u có thể được nhìn thấy thông qua tia X hoặc cảm nhận được dưới dạng cục u trong vùng vú.

Sử dụng các thư viện và framework như Sklearn, Numpy, Pandas và Mô hình học máy (Random forest, Descision Tree Classification, Logistic regression,…) để so sánh hiệu quả

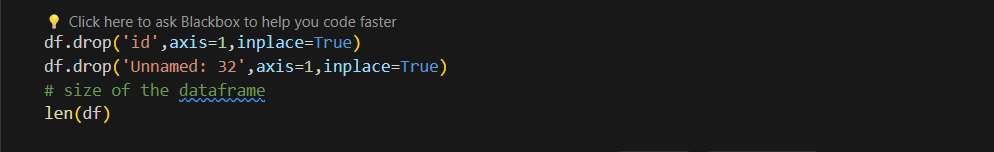
- Số bản ghi: 569

- Số lượng thuộc tính: 33

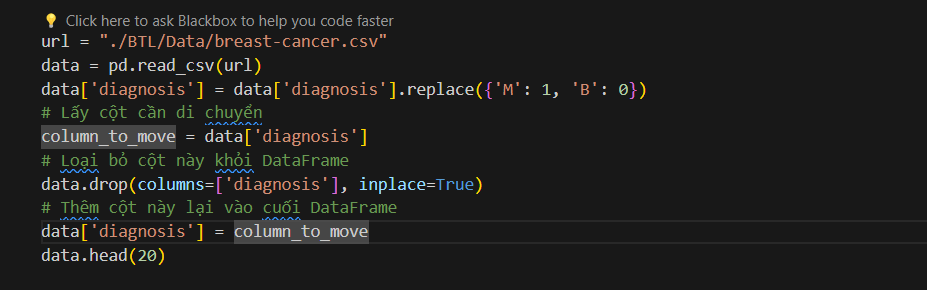
### Tiền xử lí dữ liệu

Tập dữ liệu “: Breast Cancer Wisconsin “cần loại bỏ các cột không hữu ích và xóa các hàng có chứa giá trị trống.

* Loại bỏ các cột không hữu ích như Unnamed, Id



* Di chuyển cột diagnosis xuống cuối DataFrame và chuẩn hóa dữ liệu.



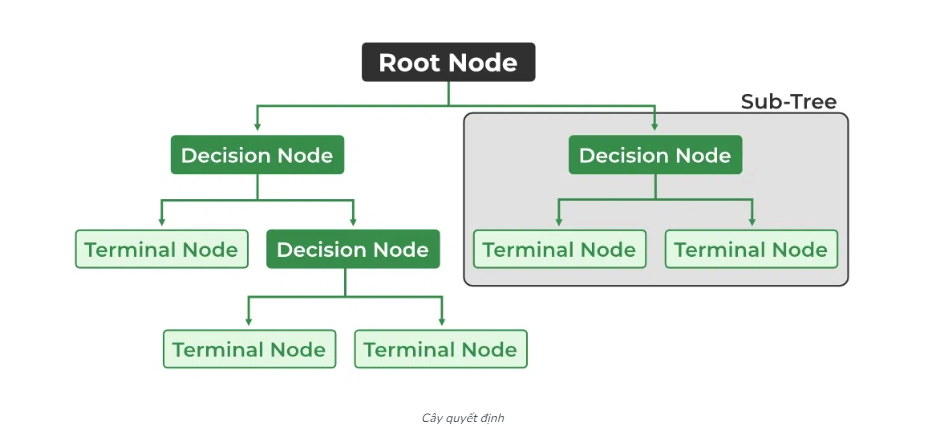
## QUÁ TRÌNH XỬ LÝ THUẬT TOÁN DECISION TREE

### Tổng quan về decision tree

Các thành phần cơ bản trong một cây quyết định sẽ bao gồm:

* Node (Nút): Là một điểm trên cây quyết định, có thể là nút gốc, nút lá hoặc nút trung gian. Nút được xác định bởi một đặc trưng và một ngưỡng giá trị của đặc trưng đó.
* Root Node (Gốc): Là nút đầu tiên trên cây quyết định, mà không có bất kỳ nút cha nào, một cây sẽ chỉ tồn tại duy nhất 1 nút lá
* Leaf Node (Nút lá): Là các nút cuối cùng trên cây quyết định, không có nút con.
* Internal Node (Nút trung gian) : Là các nút mà có ít nhất hai nút con (children nodes). Các nút trung gian được sử dụng để phân tách các quan sát dữ liệu dựa trên các đặc trưng (features) của chúng.
* Branch (Nhánh): Đường kết nối giữa các nút trên cây quyết định.

**Sơ đồ cây**



Ý tưởng thuật toán:

* Bước 1: Tìm Thuộc Tính và Giá Trị Ngưỡng Tốt Nhất+ Mục tiêu: Tìm thuộc tính (feature) và giá trị ngưỡng (threshold) tốt nhất để phân chia dữ liệu sao cho sai số của mô hình được giảm thiểu.

+ Cách thực hiện: Duyệt qua tất cả các thuộc tính và các giá trị ngưỡng có thể, tính toán chỉ số đánh giá (ví dụ: Gini, Entropy) để đo lường mức độ tốt của phân chia. Chọn thuộc tính và giá trị ngưỡng sao cho chỉ số đánh giá này là tốt nhất.

* Bước 2: Tách Dữ Liệu và Tạo Nhánh Con+ Mục tiêu: Phân chia dữ liệu thành hai phần dựa trên thuộc tính và giá trị ngưỡng đã chọn.+ Cách thực hiện: Tạo ra hai nhánh con, một nhánh chứa các điểm dữ liệu có giá trị thuộc tính nhỏ hơn hoặc bằng ngưỡng, và nhánh còn lại chứa các điểm dữ liệu có giá trị thuộc tính lớn hơn ngưỡng.
* Bước 3: Lặp Lại Quá Trình Trên Các Nhánh Con+ Mục tiêu: Tiếp tục phân chia các nhánh con cho đến khi đạt được điều kiện dừng.+ Điều kiện dừng: Có thể bao gồm:Số lượng bản ghi trong một nút không đủ lớn.Phân nhánh đã đạt độ sâu tối đa.Không có sự cải thiện đáng kể khi phân chia thêm.Các điểm dữ liệu trong một nhánh con đều có cùng một nhãn.
* Bước 4: Tính Toán Giá Trị Đầu Ra Cho Nút Dừng+ Mục tiêu: Xác định giá trị đầu ra của mỗi nút lá.+ Cách thực hiện: Lấy trung bình của các giá trị đầu ra (nhãn) của các điểm dữ liệu trong nút lá (đối với bài toán hồi quy) hoặc xác định nhãn phổ biến nhất (đối với bài toán phân loại).
* Bước 5: Sử Dụng Cây Quyết Định Để Dự Đoán+ Mục tiêu: Dự đoán nhãn cho các điểm dữ liệu mới.+ Cách thực hiện: Duyệt từ gốc của cây quyết định đến lá tương ứng với điểm dữ liệu mới, dựa trên thuộc tính và giá trị ngưỡng tại mỗi nút. Trả về giá trị hoặc nhãn của nút lá.

**Ưu điểm**

* Dễ hiểu và dễ trực quan hoá.
* Có thể xử lý cả dữ liệu số và dữ liệu phân loại.
* Có khả năng xử lý dữ liệu có nhiễu.

**Nhược điểm**

* Có thể gặp vấn đề với các tập dữ liệu phức tạp.
* Dễ bị overfitting nếu không được cắt tỉa cẩn thận.

### Thuật toán ID3

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

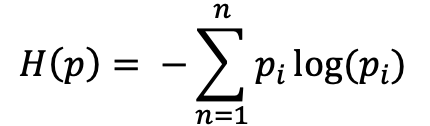
ID3 sử dụng Entropy và Information Gain để xây dựng một cây quyết định.

Entropy trong cây quyết định (Decision Tree)

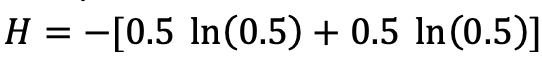
Entropy là thuật ngữ thuộc Nhiệt động lực học, là thước đo của sự biến đổi, hỗn loạn hoặc ngẫu nhiên. Năm 1948, Shannon đã mở rộng khái niệm Entropy sang lĩnh vực nghiên cứu, thống kê với công thức như sau:

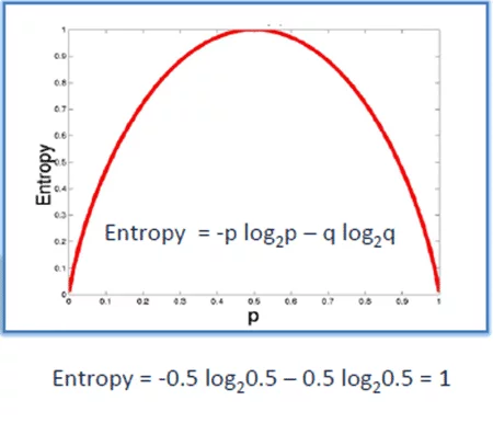
Với một phân phối xác suất của một biến rời rạc x có thể nhận ra n giá trị khác nhau x1, x2,..., xn.

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

Ký hiệu phân phối này là . Entropy của phân phối này được định nghĩa là:

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





Hình vẽ trên biểu diễn sự thay đổi của hàm Entropy. Ta có thể thấy rằng, entropy đạt tối đa khi xác suất xảy ra của 2 lớp bằng nhau.

· P tinh khiết: pi = 0 hoặc pi = 1

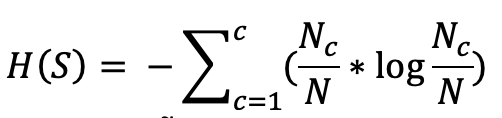
· P vẩn đục: pi = 0.5 khi đó hàm Entropy đạt đỉnh cao nhất

Information Gain trong Cây quyết định (Decision Tree)

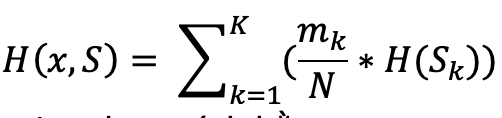
Information Gain dựa trên sự giảm của hàm Entropy khi tập dữ liệu được phân chia trên một thuộc tính. Để xây dựng một cây quyết định, ta phải tìm tất cả các thuộc tính trả về Information Gain cao nhất.

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 phần tử thuộc lớp c cho trước:



· 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 với số điểm trong mỗi child node lần lượt là , ta có:



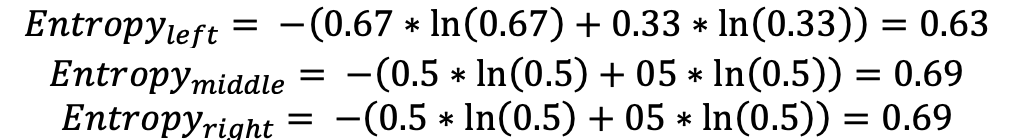
· Bước 3: Chỉ số Gain Information được tính bằng:



Với ví dụ trên, ta tính được hệ số Entropy như sau:



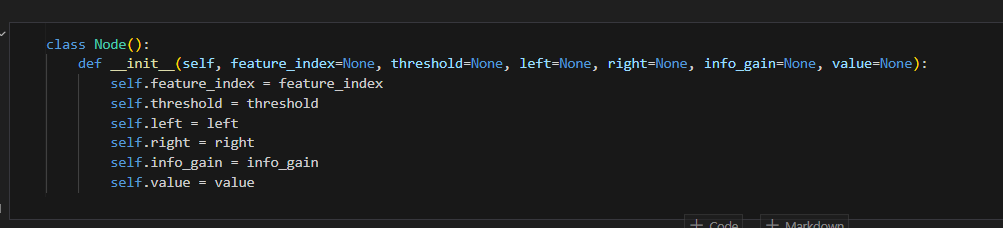
Hệ số Entropy theo phương pháp chia thứ nhất:

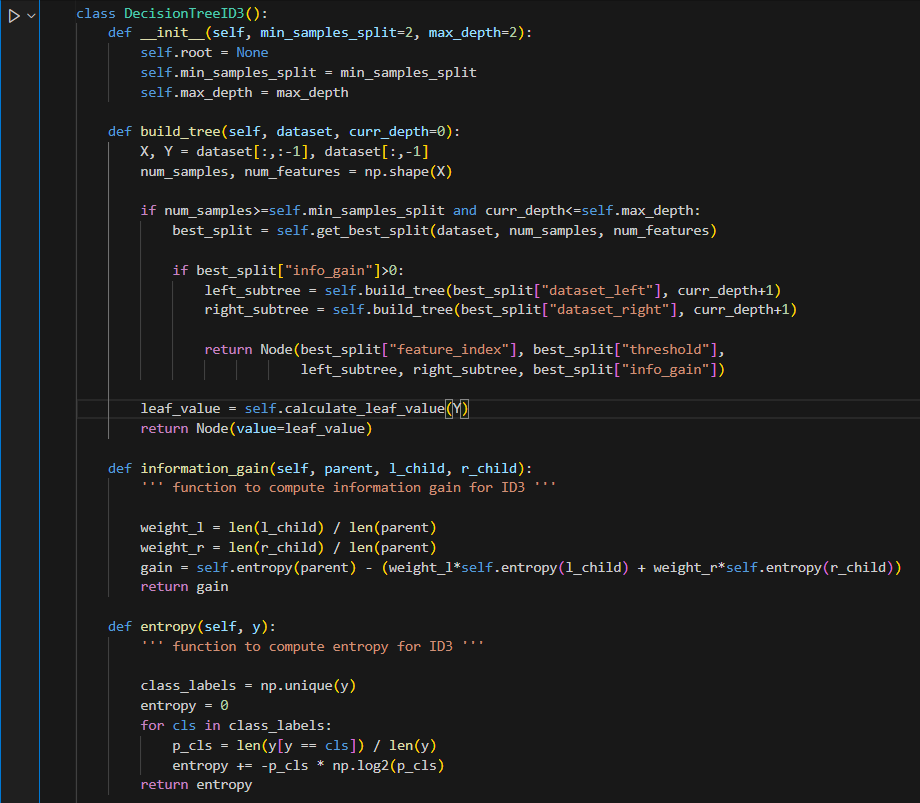


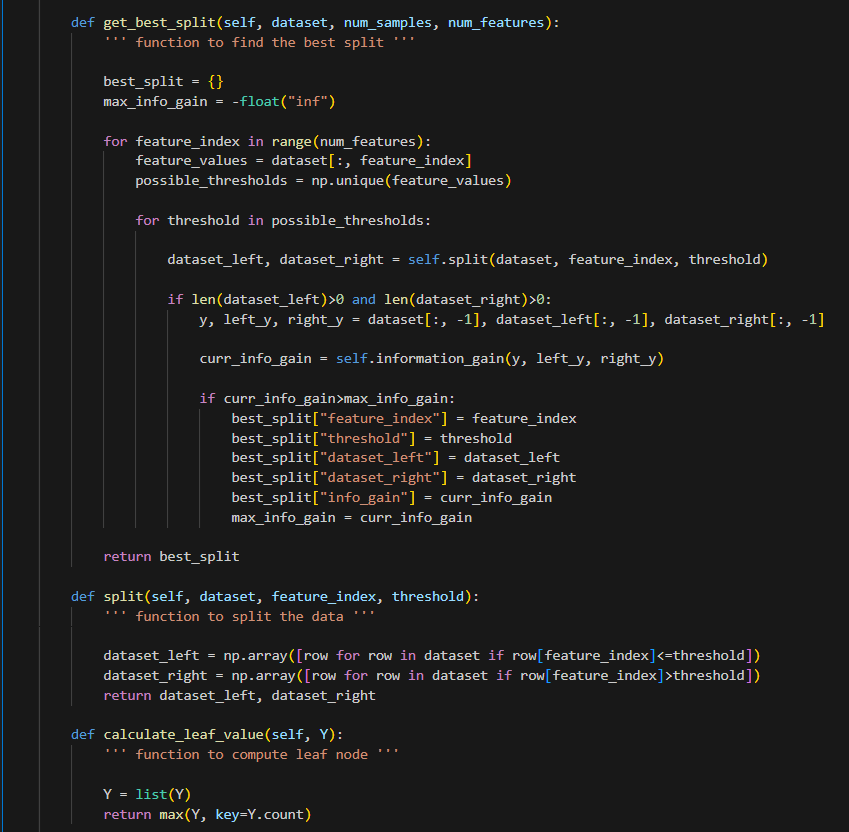
Hệ số Information Gain:

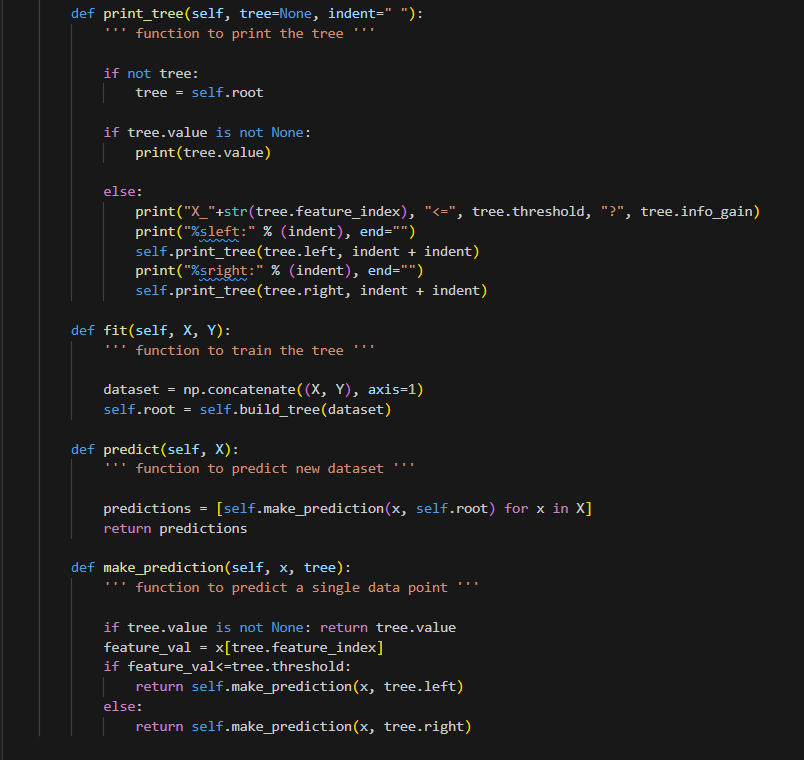


* Code thuật toán:

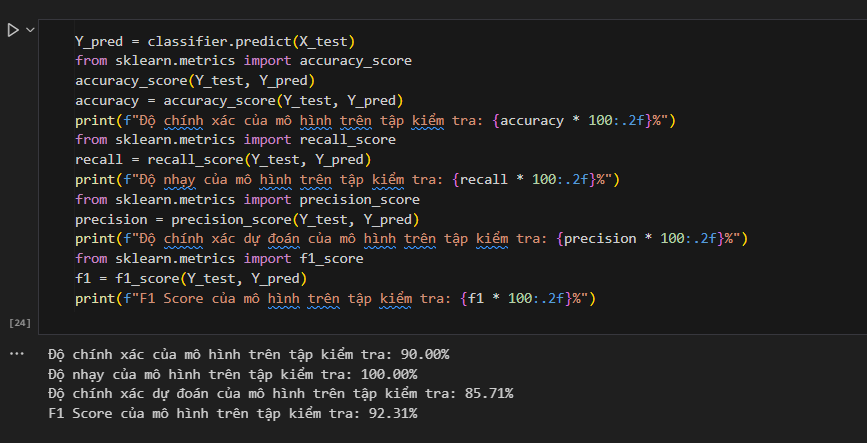
­­





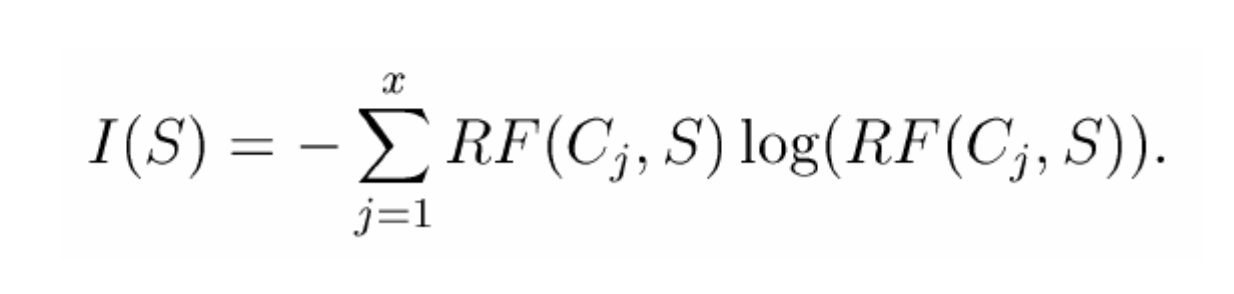


* Đánh giá thuật toán



### Thuật toán C4.5

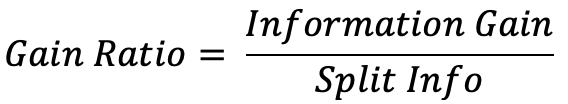
C4.5: Sử dụng information gain để chọn thuộc tính phân chia.



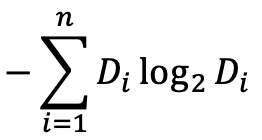
· Thuật toán C4.5 là thuật toán cải tiến của ID3.

· 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 sẽ chuẩn hoá information gain với giá trị thông tin phân tách (split information):

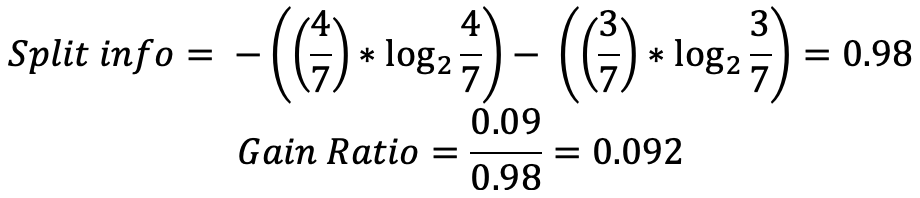


· Trong đó: Split Info được tính như sau:



· Giả sử chúng ta phân chia biến thành n nút con và 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:



· Tiêu chuẩn dừng

· Trong các thuật toán 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 free mà mọi điểm trong tập huấn luyện đều được dự đoán đúng (giả sử rằng không có 2 input giống nhau nào cho output khác nhau). Khi đó, cây có thể sẽ rất phức tạp (nhiều node) với nhiều node lá chỉ có 1 vài điểm dữ liệu.Như vậy, nhiều khả năng overfitting sẽ xảy ra.

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

o Nếu node đó có entropy = 0, tức mọi điểm trong node đều thuộc một class.

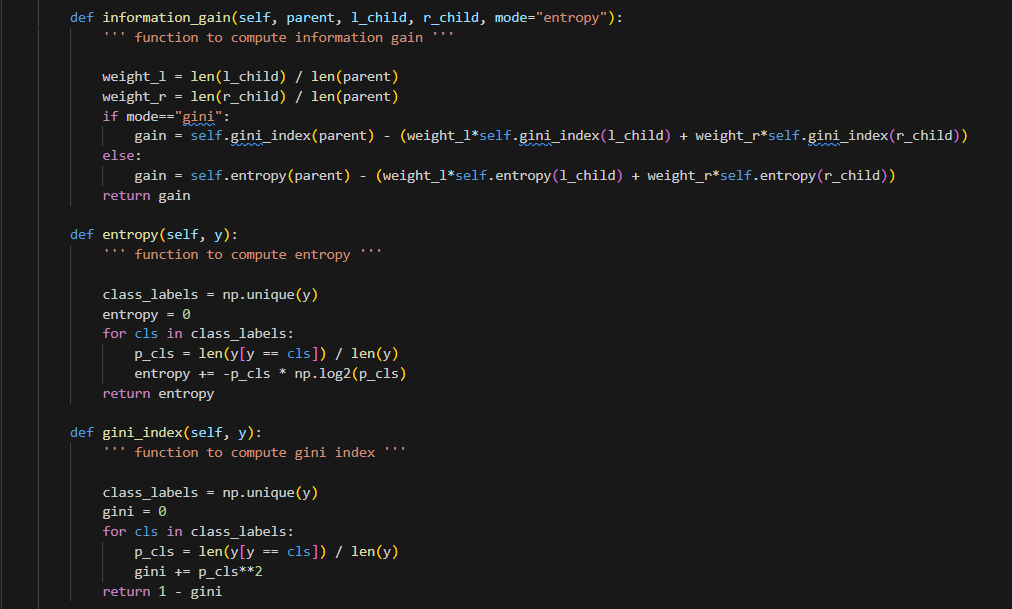
o Nếu node đó có số phần từ nhỏ hơn 1 ngưỡng nào đó. Trong trường hợp này, ta chấp nhận có một số điểm bị phân lớp sai để tránh overfitting. Class cho node lá này có thể được xác định dựa trên class chiếm đa số trong node.

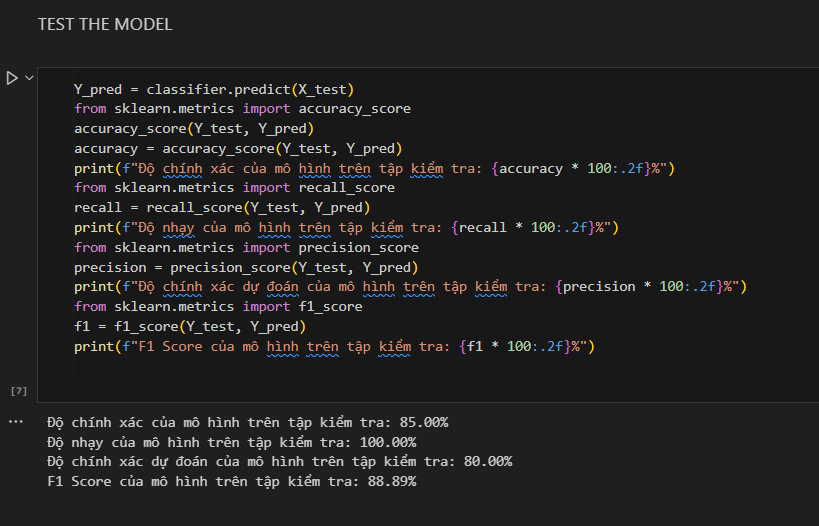
o Nếu khoảng cách từ node đó đến root node đạt tới 1 giá trị nào đó. Việc hạn chế chiều sâu của tree này làm giảm độ phức tạp của tree và phần nào giúp tránh overfitting.

o Nếu tổng số leaf node vượt quá một ngưỡng nào đó.

o Nếu công việc phân chia node đó không làm giảm entropy quá nhiều (information gain nhỏ hơn 1 ngưỡng nào đó).

· Ngoài ra, ta còn có phương pháp cắt tỉa cây.



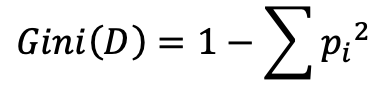


### Thuật toán CART

Thuật toán CART là thuật toán học máy phổ biến được sử dụng để xây dựng cây quyết định cho cả phân loại (classification) và hồi quy (regression). Thuật toán này được phát triển bởi Leo Breiman vào năm 1984 và được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau.

**Cách thức hoạt động:**

* Bước 1: Chọn thuộc tính phân chia:
* Sử dụng chỉ số Gini (Gini index) để đánh giá độ không tinh khiết của tập dữ liệu tại mỗi nút.
* Chọn thuộc tính có Gini index cao nhất để phân chia dữ liệu thành các tập con.
* Công thức tính Gini index:



D: Tập dữ liệu

pi: tỷ lệ phần trăm của các mẫu thuộc lớp I trong tập dữ liệu D.

* Bước 2: Lặp lại quá trình phân chia:
* Tiếp tục áp dụng bước 1 cho các tập con được tạo ra từ bước trước cho đến khi đáp ứng một trong các điều kiện dừng:

· Tất cả các mẫu dữ liệu tại 1 nút thuộc cùng 1 lớp (phân loại) hoặc cùng một giá trị (hồi quy).

· Độ sâu của dây đạt đến giá trị tối đa.

· Số lượng mẫu dữ liệu tại 1 nút nhỏ hơn giá trị tối thiểu.

* Bước 3: Cắt tỉa cây:
* Áp dụng kỹ thuật cắt tỉa (pruning) để loại bỏ các nhánh không cần thiết và tránh overfitting.
* Bước 4: Dự đoán:
* Sử dụng cây quyết định đã được xây dựng để dự đoán kết quả cho các mẫu dữ liệu mới.

**Hồi quy:**

* Trong trường hợp hồi quy, CART sử dụng phương pháp bình phương trung bình (least squares) để dự đoán giá trị mục tiêu cho các mẫu dữ liệu mới.

**Ví dụ thuật toán CART:**

Giả sử chúng ta có tập dữ liệu về các loại hoa với các thuộc tính như màu sắc, hình dạng và kích thước. Mục tiêu là xây dựng cây quyết định để phân loại các loại hoa.

Bước1: Tính toán Gini index cho mỗi thuộc tính.

Bước 2: Chọn thuộc tính có Gini index cao nhất để phân chia dữ liệu. Ví dụ, màu sắc được chọn là thuộc tính phân chia đầu tiên.

Bước 3: Lặp lại bước 1 và 2 cho các tập con được tạo ra từ bước trước.

Bước 4: Dừng việc phân chia khi tất cả các mẫu dữ liệu tại 1 nút thuộc cùng một loại hoa hoặc khi đạt đến độ sâu tối đa.

Bước 5: Cắt tỉa cây để loại bỏ các nhánh không cần thiết.

Bước 6: Sử dụng cây quyết định đã được xây dựng để dự đoán loại hoa cho các mẫu dữ liệu mới.

**Ưu điểm của thuật toán CART:**

* Dễ hiểu và trực quan: Cây quyết định được tạo bởi CART có cấu trúc đơn giản và dễ hiểu, giúp người dùng dễ dàng nắm bắt được logic của mình.
* Có thể xử lý cả dữ liệu số và dữ liệu phân loại: CART có thể xử lý nhiều loại dữ liệu khác nhau, bao gồm cả dữ liệu số (như tuổi, thu nhập) và dữ liệu phân loại (như màu sắc, giới tính).
* Khả năng giải thích cao: CART cung cấp khả năng giải thích cao cho người dùng, giúp họ hiểu được lý do đằng sau các dự đoán của mô hình.
* Hoạt động tốt với dữ liệu có nhiều thuộc tính: CART có thể xử lý hiệu quả dữ liệu có nhiều thuộc tính, giúp giảm nguy cơ overfitting.

**Nhược điểm của thuật toán CART:**

* Dễ bị overfitting: CART có thể dễ bị overfitting nếu không được điều chỉnh cẩn thận.
* Có thể không hiệu quả với dữ liệu có nhiễu: CART có thể không hoạt động hiệu quả với dữ liệu có nhiễu cao.
* Khả năng khái quát hoá thấp: CART có thể gặp khó khăn trong việc khái quát hoá sang dữ liệu mới chưa từng được nhìn thấy trước đây.

**So sánh CART với các thuật toán khác**

* **CART vs ID3:** ID3 là 1 thuật toán xây dựng cây quyết định sử dụng entropy để chọn thuộc tính phân chia. CART sử dụng Gini index để chọn thuộc tính phân chia, thường dẫn đến hiệu suất tốt hơn ID3.
* **CART vs C4.5:** C4.5 là 1 thuật toán cải tiến từ ID3, sử dụng information gain để chọn thuộc tính phân chia. C4.5 có thể hoạt động hiệu quả hơn CART trong một số trường hợp, nhưng thường phức tạp hơn và khó hiểu hơn.

### Overfitting

**Overfitting** là một hiện tượng trong học máy xảy ra khi mô hình học quá tốt từ dữ liệu huấn luyện, dẫn đến việc dự đoán kém hiệu quả trên dữ liệu mới. Nó giống như một học sinh thuộc lòng tất cả câu trả lời trong sách giáo khoa nhưng lại trượt bài kiểm tra bất ngờ vì không hiểu rõ các khái niệm cơ bản.

**Dấu hiệu của Overfitting:**

Độ chính xác cao trên tập huấn luyện, nhưng thấp trên tập kiểm tra: Đây là dấu hiệu rõ ràng nhất của overfitting.

Mô hình phức tạp: Mô hình có nhiều tham số hoặc cấu trúc phức tạp có thể dễ bị overfitting hơn.

Hệ số lỗi cao trên tập kiểm tra: Hệ số lỗi (như **RMSE, MAE**) trên tập kiểm tra tăng cao là dấu hiệu của overfitting.

**Nguyên nhân của Overfitting:**

Dưới mức dữ liệu: Mô hình không có đủ dữ liệu để học các khái niệm chung và có thể bắt đầu học các đặc điểm cụ thể của tập huấn luyện.

Mô hình quá mức phức tạp: Mô hình có nhiều tham số hoặc cấu trúc phức tạp có thể dễ dàng học thuộc lòng tập huấn luyện.

Thiếu dữ liệu đa dạng: Dữ liệu huấn luyện không bao gồm đủ các trường hợp đa dạng có thể dẫn đến việc mô hình học các đặc điểm cụ thể của tập dữ liệu.

**Phòng ngừa overfitting:**

Tăng kích thước tập huấn luyện: Cung cấp cho mô hình nhiều dữ liệu hơn giúp nó học các khái niệm chung tốt hơn và giảm nguy cơ học thuộc lòng.

Sử dụng kỹ thuật Regularization: kỹ thuật Regularization thêm 1 khoản phạt vào hàm mục tiêu của mô hình để ngăn chặn nó trở nên quá phức tạp.

Sử dụng kỹ thuật Early stopping: Theo dõi hiệu suất của mô hình trên tập kiểm tra và dừng quá trình huấn luyện khi hiệu suất bắt đầu giảm.

Sử dụng kỹ thuật Augmentation data: Tăng cường dữ liệu huấn luyện bằng cách thêm các biến thể mới của các mẫu dữ liệu hiện có.

Sử dụng mô hình đơn giản: Chọn mô hình có cấu trúc đơn giản hơn có thể giúp giảm nguy cơ overfitting.

### Cắt tỉa cây

1. **Pre-pruning:** Dừng việc xây dựng cây trước khi nó đạt đến độ sâu tối đa.

Pre-pruning, hay còn gọi là dừng sớm (early stopping), là một kỹ thuật được sử dụng để ngăn ngừa overfitting trong thuật toán Decision Tree. Overfitting xảy ra khi cây quyết định trở nên quá phức tạp và bắt đầu học thuộc các nhiễu trong dữ liệu đào tạo, dẫn đến hiệu quả kém trên dữ liệu mới.

**Hoạt động của Pre-pruning:**

Pre-pruning hoạt động bằng cách thiết lập các tiêu chí dừng trong quá trình xây dựng cây. Khi một nút đạt đến 1 tiêu chí dừng, thuật toán sẽ dừng việc phân chia các nút đó thành các nhánh con và thay vào đó chuyển đổi nó thành 1 lá. Lá này sẽ đại diện cho dự đoán cuối cùng cho các điểm dữ liệu thuộc về nhánh đó.

**Các tiêu chí dừng thường được sử dụng trong pre-pruning:**

* Độ sâu tối đa (Maximum Depth): Giới hạn độ sâu tối đa của cây. Khi một nút đạt đến độ sâu này, nó sẽ được chuyển thành lá.
* Số lượng mẫu tối thiểu tại 1 nút (Minimum Sample Split): Yêu cầu phải có 1 số lượng mẫu tối thiểu nhất định tại 1 nút trước khi nó được phân chia. Nếu số lượng mẫu nhỏ hơn ngưỡng này, nút sẽ được chuyển thành lá.
* Giảm thiểu độ không tinh khiết tối thiểu (Minimum Impurity Decrease): Chỉ chia 1 nút nếu việc phân chia dẫn đến giảm độ không tinh khiết (ví dụ như entropy hoặc Gini index) vượt quá một ngưỡng nhất định.

**Ưu điểm của Pre-pruning:**

* Ngăn ngừa overfitting.
* Giữ cho cây quyết định đơn giản và dễ hiểu hơn.
* Giảm thời gian đào tạo.

**Nhược điểm của Pre-pruning:**

* Có thể bỏ qua các phân chia hữu ích nếu ngưỡng dừng quá chặt chẽ.
* Cần điều chỉnh các tham số dừng (chẳng hạn như độ sâu tối đa) để đạt hiệu quả tốt nhất.

**Ví dụ Pre-pruning:** Giả sử chúng ta có tập dữ liệu về các loại hoa với các thuộc tính như màu sắc, hình dạng và kích thước. Mục tiêu là xây dựng cây quyết định để phân loại các loại hoa.

* Chọn 1 tiêu chí dừng, ví dụ độ sâu tối đa là 3.
* Bắt đầu xây dựng cây từ gốc.
* Tại mỗi nút, chọn thuộc tính có mức độ thông tin cao nhất để phân chia dữ liệu.
* Dừng việc phân chia 1 nút khi nó đạt đến độ sâu tối đa (3) hoặc khi tất cả các mẫu dữ liệu tại nút thuộc cùng một loại hoa.
* Cây quyết định được tạo ra sẽ có độ phức tạp thấp hơn và ít có khả năng overfitting hơn so với cây được xây dựng mà không có pre-pruning.

1. **Post-pruning:** Loại bỏ các nhánh không cần thiết sau khi cây đã được xây dựng.

**Post-pruning**, còn được gọi là cắt tỉa sau (backward pruning), là một kỹ thuật khác được sử dụng để chống lại overfitting trong cây quyết định. Giống như pre-pruning, nó nhằm mục đích đơn giản hoá và cải thiện khả năng tổng quát hoá.

**Cách thức hoạt động của Post-pruning:**

* Xây dựng cây đầy đủ: Khác với pre-pruning, thuật toán sẽ xây dựng hoàn chỉnh cây quyết định.
* Đánh giá các nhánh: Xét từng nhánh trong cây và đánh giá mức độ đóng góp của nó vào hiệu suất của mô hình. Các phép đo thường dùng để đánh giá nhánh bao gồm độ phức tạp của nhánh, độ giảm thiểu độ không tinh khiết và hiệu suất trên tập dữ liệu (xác minh – validation set).
* Xác minh (Validation set) là một tập dữ liệu riêng biệt được sử dụng để đánh giá hiệu suất của mô hình trên dữ liệu chưa nhìn thấy trước đó. Nó giúp tránh việc đánh giá quá mức (overfitting) trên tập dữ liệu huấn luyện.
* Loại bỏ nhánh không cần thiết: Dựa trên kết quả đánh giá, loại bỏ các nhánh được coi là không cần thiết cho việc dự đoán. Nhánh bị loại bỏ sẽ được thay thế bằng lá đại diện cho dự đoán phổ biến cho các mẫu dữ liệu thuộc về nhánh đó.

**Ưu điểm của Post-pruning:**

* Linh hoạt hơn pre-pruning vì nó có thể tận dụng tất cả thông tin có trong cây được xây dựng đầy đủ.
* Có thể loại bỏ các nhanh có hại 1 cách hiệu quả hơn nếu chúng được xác định rõ ràng trong cây đã hoàn thiện.

**Nhược điểm của Post-pruning:**

* Có thể tính toán tốn thời gian hơn so với pre-pruning vì cần đánh giá tất cả các nhánh trong cây.
* Nguy cơ loại bỏ các nhánh quan trọng nếu chiến lược đánh giá không đủ chính xác.

1. **Lựa chọn giữa Pre-pruning và Post-pruning:**

Pre-pruning thường được ưu tiên hơn vì nó đơn giản hơn về tính toán và có thể ngăn ngừa overfitting ngay từ đầu. Tuy nhiên, trong một số trường hợp, post-pruning có thể hiệu quả hơn nếu chiến lược đánh giá nhánh được thiết kế tốt. Thực tế, một số thuật toán Decision Tree kết hợp cả 2 kỹ thuật này.

Ví dụ Pre-pruning: Giả sử chúng ta có tập dữ liệu về các loại hoa với các thuộc tính như màu sắc, hình dạng và kích thước. Mục tiêu là xây dựng cây quyết định để phân loại các loại hoa.

* Xây dựng cây quyết định đầy đủ.
* Sử dụng tập dữ liệu xác nhận để đánh giá hiệu suất của từng nhánh cây.
* Loại bỏ các nhánh có độ phức tạp cao và không đóng góp đáng kể vào hiệu suất của mô hình.
* Cây quyết định sau khi cắt tỉa sẽ đơn giản hơn và có khả năng tổng quát hoá tốt hơn.

## QUÁ TRÌNH XỬ LÝ THUẬT TOÁN RANDOM FOREST

### Random Forest

Random forest tạo ra cây quyết định trên các mẫu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và kết quả cuối cùng dựa trên cơ chế major voting (bầu chọn) đối với bài toán phân loại và averaging (trung bình) đối với bài toán hồi quy.

Thuật toán Random Forest là một phương pháp thống kê mô hình hóa bằng máy (machine learning statistic) dùng để phục vụ mục đích phân loại, tính hồi quy và các nhiệm bằng cách xây dựng nhiều cây quyết định (Decision tree).

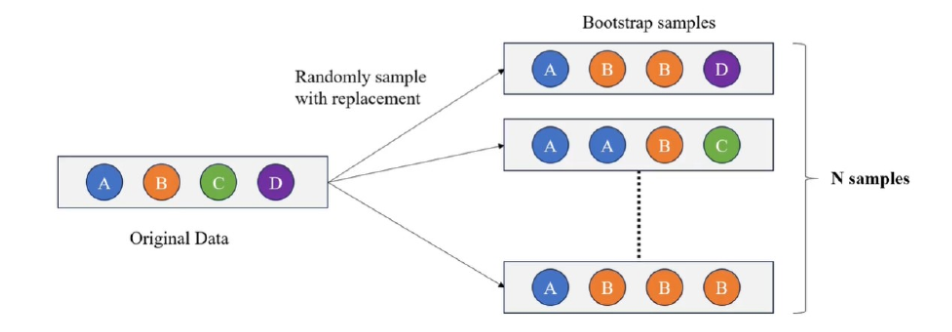
Random forest được sử dụng cho cả bài toán phân loại (classification) và hồi quy (regression). Random forest là một phương pháp dựa trên việc nhiều cây quyết định (decision trees) một cách hệ thống.



Random forest được cấu thành bởi nhiều cây quyết định. Các cây này cùng nhận đầu vào là một đối tượng và đưa ra quyết định về danh mục thuộc tính của đối tượng đó. Các quyết định này sẽ được tổng hợp lại lấy trung bình để chọn ra quyết định cuối cùng.

1.1. Phương pháp Bootstrap

Bootstrap là phương pháp lấy mẫu có hoàn lại (sampling with replacement). Phương pháp lấy mẫu có hoàn lại có nghĩa là một cá thể có thể xuất hiện nhiều lần trong một lần lấy mẫu. Phương pháp chủ yếu dùng để ước lượng lỗi chuẩn (standard errors), độ lệch (bias) và tính toán khoảng tin cậy (confidence interval) cho các tham số.



1.2. Phương pháp Bagging

Mô hình hoạt động của Bagging:

Bagging (Bootstrap aggregation) là tổng hợp các bootstrap sử dụng cách tiếp cận xây dựng mỗi bộ phận loại một cách độc lập với nhau. Bagging tạo ra các bộ phận loại từ các tập con có lặp từ tập mẫu ban đầu (sử dụng bootstrap lấy mẫu có hoàn lại) và một thuật toán học máy, mỗi tập mẫu sẽ tạo ra một bộ phận loại cơ bản. Sau đó sử dụng phương pháp bỏ phiếu để chọn kết quả cuối cùng của bộ kết hợp.

Thuật toán Bagging: Bagging tạo ra N tập huấn luyện được chọn có lặp từ tập dữ liệu huấn luyện ban đầu. Trong đó các mẫu huấn luyện có thể được chọn hơn một lần hoặc không được chọn lần nào. Từ mỗi tập huấn luyện mới, Bagging cho chạy với một thuật toán học máy L để sinh ra M bộ phận cơ bản hm. Tùy thuộc vào nhiệm vụ (tức là hồi quy hoặc phân loại), thì việc lấy giá trị trung bình hoặc phần lớn các dự đoán được lấy để tính toán ước tính chính xác hơn. Trong trường hợp hồi quy, giá trị trung bình của tất cả các kết quả đầu ra được dự đoán bởi các bộ phận riêng lẻ sẽ được lấy, điều này được gọi là bỏ phiếu mềm (soft voting). Đối với các bài toán phân loại, lớp có số phiếu bầu cao nhất được chấp nhận, điều này được gọi là bỏ phiếu cứng hoặc phiếu đa số (hard voting).

1.3. Học tập thể

Với bài toán phân loại hoặc hồi quy cụ thể, người ta thường có nhiều thuật toán học để khi xây dựng bộ học. Cùng một thuật toán, có thể chọn các tham số khác nhau sử dụng tập dữ liệu huấn luyện khác nhau nên cho các bộ phân loại khác nhau.

Khi thực hiện phân loại thì mỗi bộ huấn luyện theo thuật toán tương ứng có những lớp mẫu được phân loại tốt và tồi khác nhau. Kết hợp hợp lý các bộ phân loại có thể cho ta bộ phân loại mới có nhiều ưu điểm hơn,cách kết hợp này gọi là học máy tập thể.(ensemble learning).

Như vậy, mỗi cách học cho ta một bộ phân loại cơ sở, nhờ kết hợp các bộ phân loại thành phần có được mà ta có một bộ phân loại tốt hơn. Các bộ phân loại cơ sở này thường được xây dựng theo cách tiếp cận sau đây:

1) Dùng các thuật toán huấn luyện khác nhau. Các thuật toán này sử dụng các giả thuyết khác nhau về dữ liệu, các bộ học có thể phụ thuộc tham số hoặc không. Khi kết hợp các bộ học, ta được giải phóng khỏi các giả thiết áp đặt này.

2) Mỗi bộ học dùng cách chọn đặc trưng khác nhau. Chẳng hạn chúng ta dùng một thuật toán để phân biệt chữ viết tay nhưng cách chọn đặc trưng có thể là nội dung ảnh hayqua phép biến đổi nào đó.

3) Có thể sử dụng cùng một thuật toán nhưng có tham số khác nhau. Chẳng hạn đều sử dụng thuật toán k láng giềng gần nhất nhưng với số lượng cây k khác nhau.

4) Cùng một thuật toán nhưng sử dụng các tập dữ liệu huấn luyện khác nhau. Thông thường thì các bộ phân loại được xây dựng theo hai cách cách tiếp cận đầu có thời gian chạy khác nhau và bộ phân loại chính xác hơn thường đòi hỏi thời gian xử lý nhiều hơn.

### Ưu điểm của Random Forest

Trong thuật toán Decision Tree, khi xây dựng cây quyết định nếu để độ sâu của cây sẽ phân loại đúng hết các dữ liệu trong tập training dẫn đến mô hình có thể dự đoán tệ trên tập training (test), khi đó mô hình bị overfitting.

Thuật toán Random Forest sử dụng nhiều cây quyết định, mà mỗi cây quyết định không sử dụng tất cả dữ liệu tập training, cũng như không sử dụng tất cả các thuộc tính của dữ liệu để xây dựng cây nên mỗi cây có thể sẽ dự đoán không tốt, khi đó mỗi mô hình cây quyết định không bị overfitting mà có thể bị underfitting. Tuy nhiên kết quả cuối cùng của thuật toán Random Forest lại tổng hợp lại từ nhiều cây quyết định, thế nên thông tin từ các cây sẽ bổ sung thông tin cho nhau, dẫn đến mô hình có kết quả dự đoán tốt

### Nhược điểm của Random Forest

Mặc dù cố gắng giảm thiểu overfitting so với Decision tree, nhưng với số lượng cây lớn và số lượng đặc trưng nhiều thì vẫn có khả năng mô hình bị overfitting trên dữ liệu huấn luyện.

Phức tạp, không trực quan: Random Forest có khả năng đánh giá đặc trưng quan trọng, việc diễn giải các quyết định của mô hình có thể khó khăn.

Cần nhiều tài nguyên tính toán, quá trình dự đoán tốn nhiều thời gian.

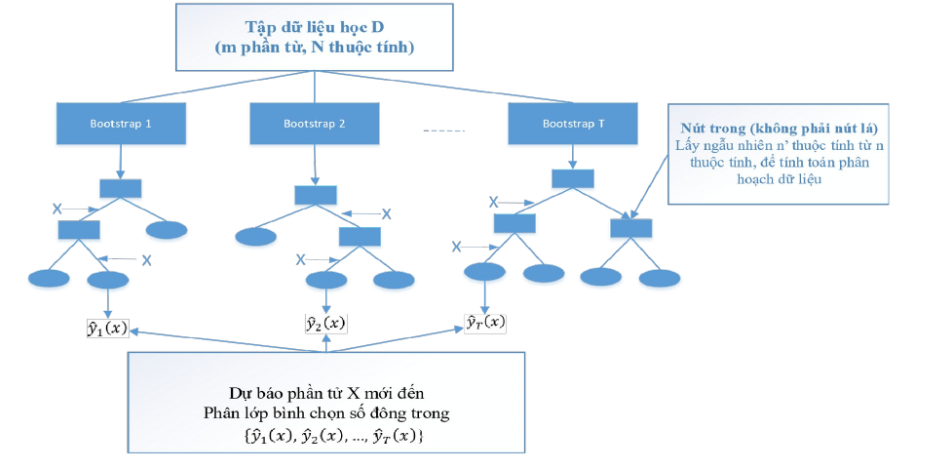
### Xây dựng thuật toán Random Forest

Thuật toán Random Forest sử dụng hai kỹ thuật bagging và bootstrapping đã được cải tiến. Bootstrapping là một phương pháp nổi tiếng được thực hiện như sau: từ một quần thể ban đầu lấy ra một mẫu D = (x1, x2, x3, .., xn) gồm n thành phần để tính toán các tham số mong muốn.

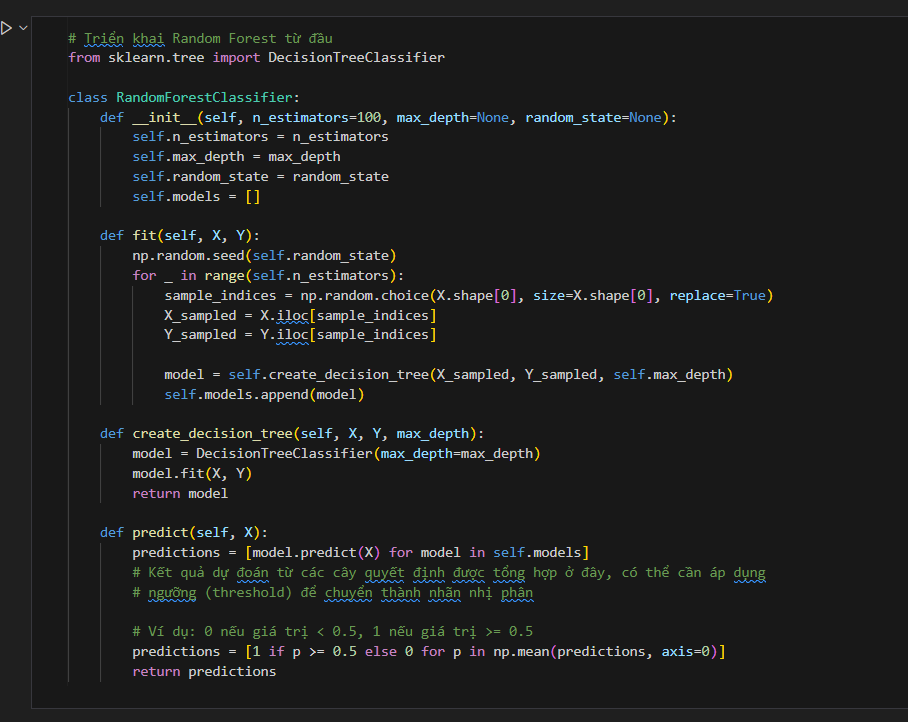
Bước 1: Từ tập dữ liệu huấn luyện D, ta tạo dữ liệu ngẫu nhiên (mẫu bootstrap).

Bước 2: Sử dụng các tập dữ liệu con dữ liệu lấy mẫu ngẫu nhiên D1, D2,...,Dk xây dựng nên các cây T1, T2,..., Tk.

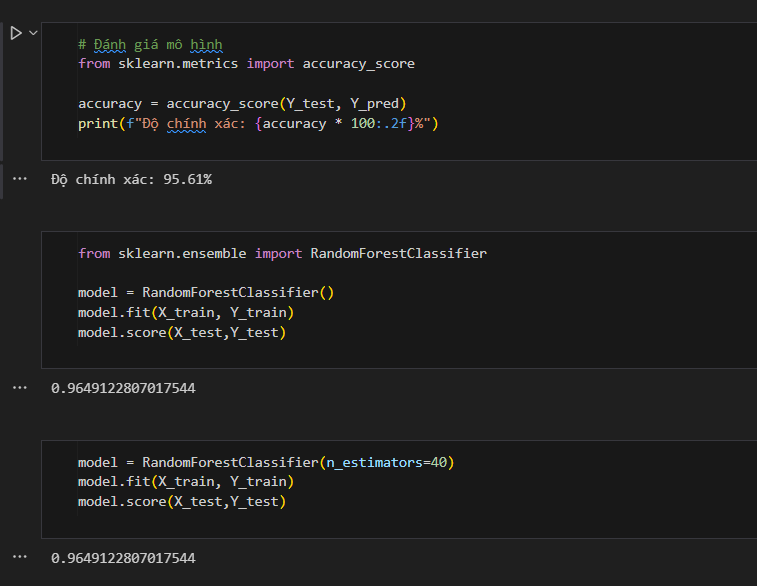
Bước 3: Kết hợp các cây: sử dụng chiến lược bình chọn theo số đông với bài toán phân loại hoặc lấy trung bình các giá trị dự đoán từ các cây với bài toán hồi quy.



* Code thuật toán



* Đánh giá mô hình



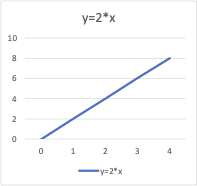
## QUÁ TRÌNH XỬ LÝ THUẬT TOÁN LOGISTIC REGRESSION

### Mô hình Logistic Regression

Để hiểu rõ về mô hình hồi quy logistic, trước tiên chúng ta phải hiểu các phương trình và biến.

**Phương trình**

Trong toán học, phương trình cho ta mối quan hệ giữa hai biến: *x* và *y*. Ta có thể sử dụng các phương trình hoặc hàm này để vẽ đồ thị theo trục x và trục y bằng cách nhập các giá trị khác nhau của *x* và *y*. Ví dụ: nếu ta vẽ đồ thị cho hàm *y* = 2\**x*, sẽ có một đường thẳng như hình dưới đây. Do đó hàm này còn được gọi là hàm tuyến tính.

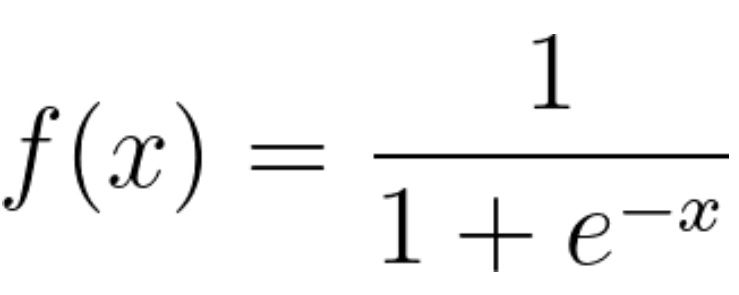


**Biến**

Trong thống kê, biến là các yếu tố dữ liệu hoặc thuộc tính có giá trị khác nhau. Bất kỳ phân tích nào cũng có một số biến nhất định là biến độc lập hoặc biến giải thích. Những thuộc tính này là nguyên nhân của một kết quả. Các biến khác là biến phụ thuộc hoặc biến đáp ứng; giá trị của chúng phụ thuộc vào các biến độc lập. Nhìn chung, hồi quy logistic khám phá cách các biến độc lập ảnh hưởng đến một biến phụ thuộc bằng cách xem xét các giá trị dữ liệu lịch sử của cả hai biến.

Trong ví dụ ở trên của chúng tôi, *x* được gọi là biến độc lập, biến dự đoán hoặc biến giải thích vì nó có một giá trị đã xác định. *Y* được gọi là biến phụ thuộc, biến kết quả hoặc biến đáp ứng vì giá trị của nó không xác định.

**Hàm hồi quy logistic**

****

Hồi quy logistic là một mô hình thống kê sử dụng hàm logistic, hay hàm logit trong toán học làm phương trình giữa *x* và *y*. Hàm logit ánh xạ *y* làm hàm sigmoid của *x*.

Nếu vẽ phương trình hồi quy logistic này, ta sẽ có một đường cong hình chữ S như hình dưới đây.

Như có thể thấy, hàm logit chỉ trả về các giá trị giữa 0 và 1 cho biến phụ thuộc, dù giá trị của biến độc lập là gì. Đây là cách hồi quy logistic ước tính giá trị của biến phụ thuộc. Phương pháp hồi quy logistic cũng lập mô hình phương trình giữa nhiều biến độc lập và một biến phụ thuộc.

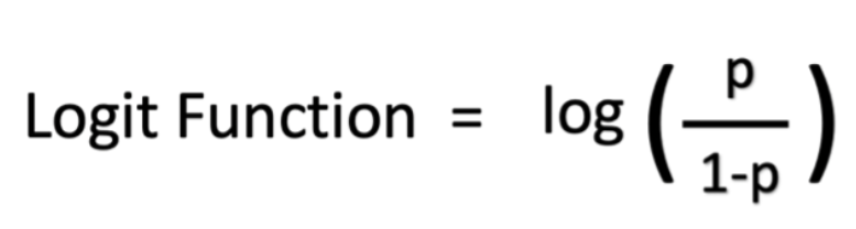
**Phân tích hồi quy logistic với nhiều biến độc lập**

Trong nhiều trường hợp, nhiều biến giải thích ảnh hưởng đến giá trị của biến phụ thuộc. Để lập mô hình các tập dữ liệu đầu vào như vậy, công thức hồi quy logistic phải giả định mối quan hệ tuyến tính giữa các biến độc lập khác nhau. Có thể sửa đổi hàm sigmoid và tính toán biến đầu ra cuối cùng như sau

*y* = *f*(β0 + β1*x*1 + β2*x*2+… βn*x*n)

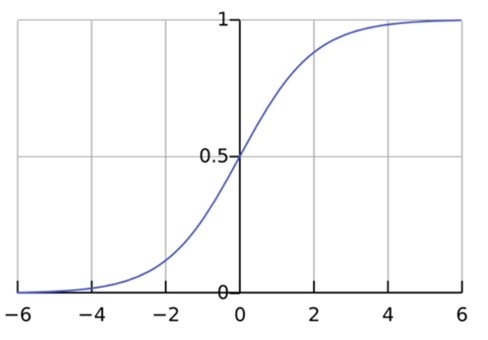
Ký hiệu β đại diện cho hệ số hồi quy. Mô hình logit có thể đảo ngược tính toán các giá trị hệ số này khi ta cho nó một tập dữ liệu thực nghiệm đủ lớn có các giá trị đã xác định của cả hai biến phụ thuộc và biến độc lập.

**Log của tỷ số odds**

****

Mô hình logit cũng có thể xác định tỷ số thành công trên thất bại hay log của tỷ số odds. Ví dụ: nếu ta đang chơi poker với bạn bè và thắng bốn ván trên mười ván, tỷ số chiến thắng của mình là bốn phần sáu, hoặc 4/6, và đó là tỷ số thành công trên thất bại của mình. Mặt khác, xác suất thắng là 4/10.

Về mặt toán học, tỷ số odds về mặt xác suất của ta là *p*/(1 - *p*) và log của tỷ số odds là log (*p*/(1 - *p*)). Ta có thể biểu diễn hàm logistic bằng log của tỷ số odds như hình dưới đây:

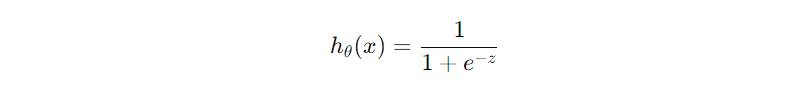


### Các bước của Thuật toán Logistic Regression

* Bước 1: Chuẩn Bị Dữ LiệuThu thập dữ liệu: Thu thập các điểm dữ liệu huấn luyện, mỗi điểm dữ liệu bao gồm các thuộc tính (features) và nhãn (label).Tiền xử lý dữ liệu: Chuẩn hóa hoặc chuẩn hóa các thuộc tính nếu cần thiết. Thêm một cột 1 vào ma trận thuộc tính để xử lý hệ số chặn (bias term).
* Bước 2: Khởi Tạo Trọng SốKhởi tạo trọng số (weights): Bắt đầu với các trọng số nhỏ ngẫu nhiên hoặc bằng 0 cho mỗi thuộc tính. Thêm một trọng số cho hệ số chặn.
* Bước 3: Xác Định Hàm Giả ThuyếtHàm giả thuyết (hypothesis): Sử dụng mô hình tuyến tính để tính toán giá trị logit

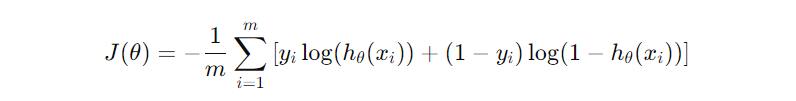


Sau đó, sử dụng hàm logistic để chuyển đổi logit thành xác suất:



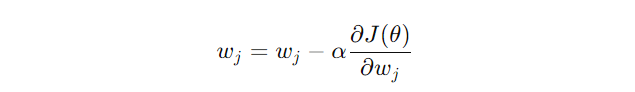
Trong đó, hθ(x) là xác suất dự đoán mà điểm dữ liệu thuộc lớp "1".

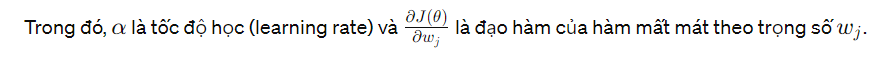
* Bước 4: Xác Định Hàm Mất MátHàm mất mát (loss function): Sử dụng hàm mất mát log-likelihood để đo lường độ chính xác của mô hình:



Trong đó, m là số lượng điểm dữ liệu, yi là nhãn thực tế của điểm dữ liệu thứ i.

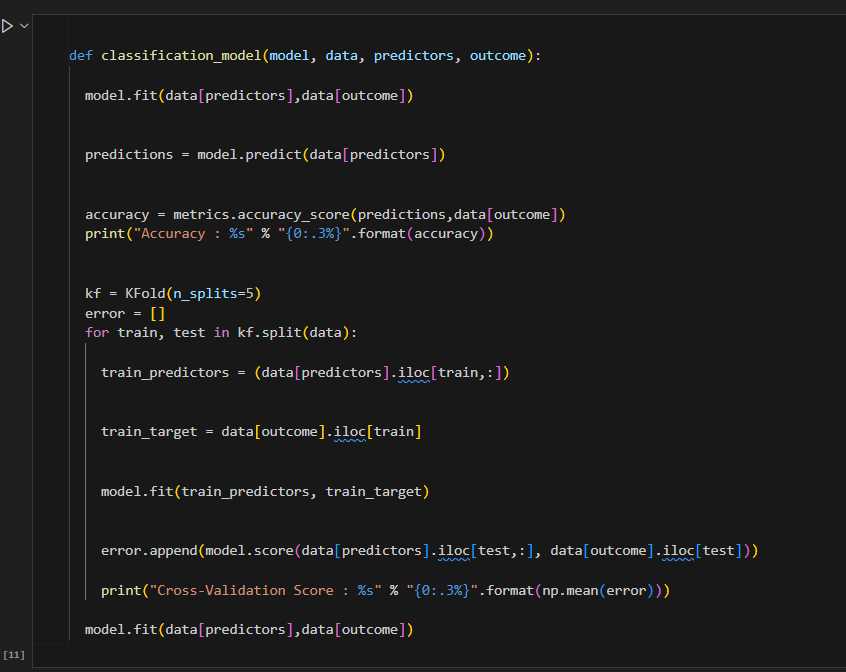
* Bước 5: Tối Ưu Hóa Trọng SốTối ưu hóa trọng số: Sử dụng phương pháp gradient descent để tối ưu hóa hàm mất mát và cập nhật trọng số:

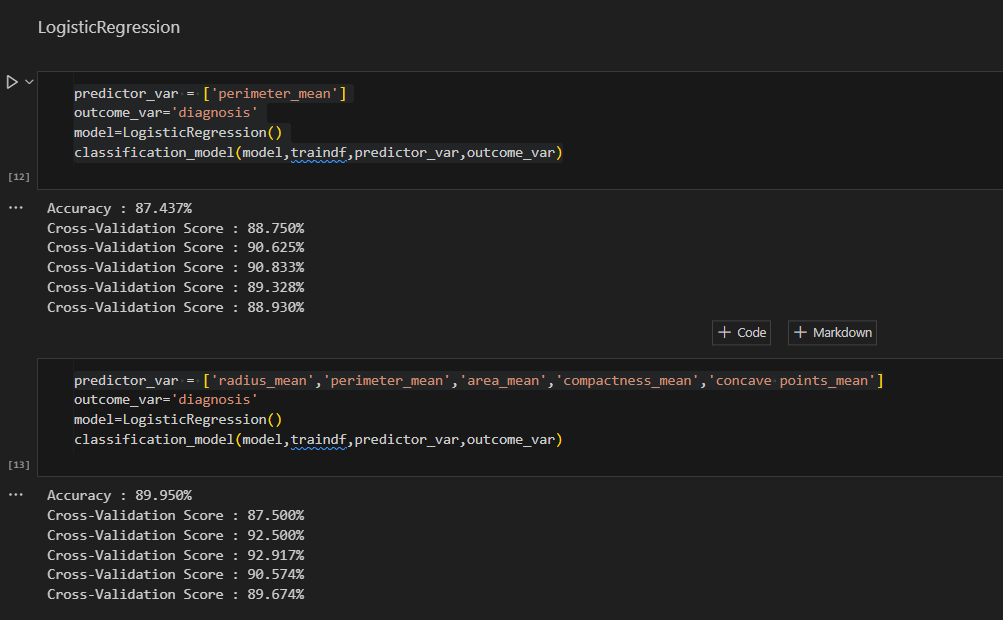




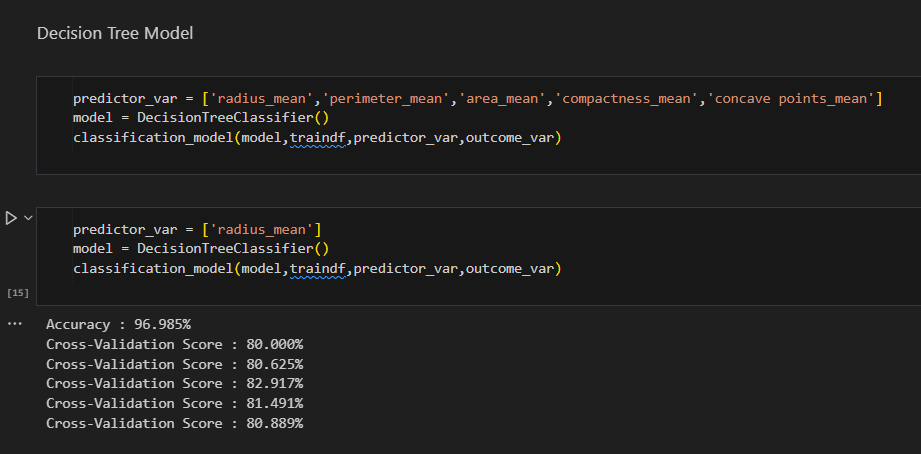
* Bước 6: Lặp Lại Quá Trình Tối ƯuLặp lại quá trình tối ưu: Lặp lại bước 5 cho đến khi hàm mất mát hội tụ hoặc đạt số lần lặp tối đa.
* Bước 7: Dự Đoán và Đánh GiáDự đoán: Sử dụng mô hình đã huấn luyện để dự đoán xác suất cho các điểm dữ liệu mới và phân loại chúng dựa trên ngưỡng xác suất (thông thường là 0.5).Đánh giá mô hình: Sử dụng các chỉ số như độ chính xác, độ nhạy, độ đặc hiệu, và AUC-ROC để đánh giá hiệu suất của mô hình.

### Code thuật toán

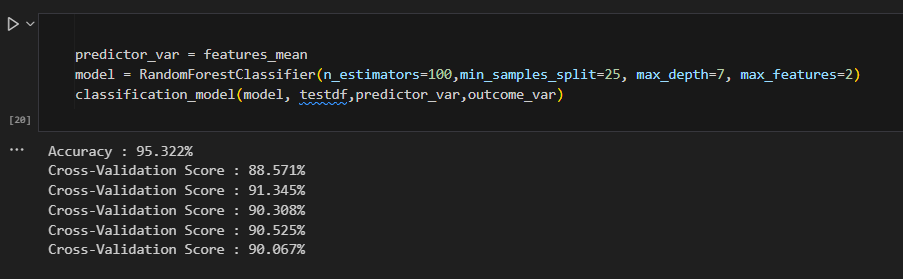




* So sánh với Decision Tree



* So sánh với Random Forest

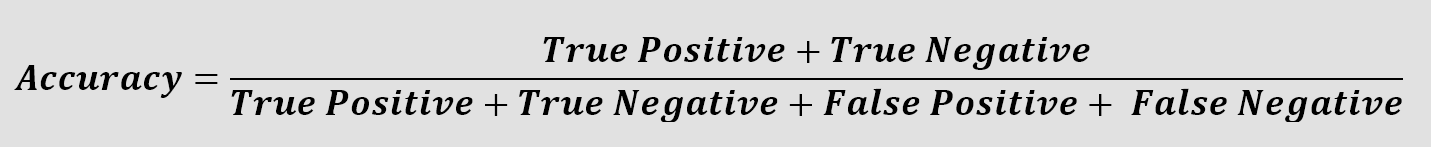


## ĐÁNH GIÁ HIỆU QUẢ THUẬT TOÁN

### Các tiêu chí đánh giá

1. Độ chính xác (Accuracy)

Đây là tiêu chí cơ bản để đánh giá hiệu suất của một thuật toán phân loại. Các thuật toán được so sánh dựa trên tỷ lệ dự đoán chính xác việc có hoặc không một bệnh nhân mắc bệnh ung thư vú trên toàn bộ tập dữ liệu.



Với:

· TP (True Positive): Số lượng trường hợp dương tính được dự đoán chính xác (bệnh nhân ung thư vú được dự đoán là ung thư vú).

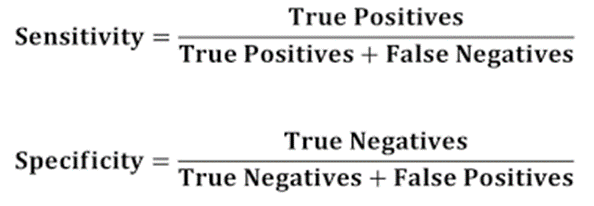
· TN (True Negative): Số lượng trường hợp âm tính được dự đoán chính + xác (người không bị ung thư vú được dự đoán là không bị ung thư vú).

· FP (False Positive): Số lượng trường hợp âm tính được dự đoán sai (người không bị ung thư vú được dự đoán là ung thư vú).

· FN (False Negative): Số lượng trường hợp dương tính được dự đoán sai (bệnh nhân ung thư vú được dự đoán là không bị ung thư vú).

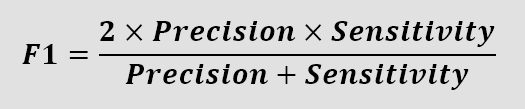
1. Sensitivity – Specificity

Khả năng dự đoán chính xác các trường hợp dương tính – âm tính (bệnh nhân ung thư vú). Chỉ số Sensitive đo lường khả năng của một mô hình phát hiện các trường hợp dương tính (bệnh nhân mắc bệnh), trong khi chỉ số Specificity đo lường khả năng loại trừ các trường hợp âm tính (bệnh nhân không mắc bệnh). Một mô hình tốt sẽ có cả hai chỉ số này đều cao.

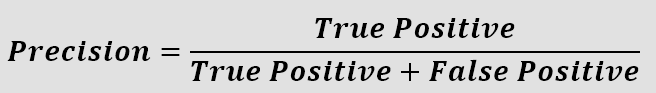


1. F1-score

Một phép đo tổng hợp của độ nhạy (sensitivity) và độ chính xác (precision) của một mô hình phân loại. F1 score cao cho logistic regression có thể chỉ ra rằng mô hình này có khả năng dự đoán chính xác cả các trường hợp mắc bệnh và không mắc bệnh một cách cân bằng; với decision tree có thể chỉ ra rằng mô hình này có khả năng tạo ra các quy tắc phân loại hiệu quả; và với random forest có thể chỉ ra rằng mô hình này có khả năng kết hợp hiệu quả giữa các cây quyết định để đưa ra dự đoán chính xác.



Với:



d) AUC (Area Under the ROC Curve)

Một thước đo hiệu suất của mô hình phân loại, thể hiện khả năng phân biệt giữa các trường hợp dương tính và âm tính của thuật toán. Diện tích hay giá trị AUC càng cao (> 0,5), khả năng phân biệt của mô hình càng tốt.



### Đánh giá 3 thuật toán

1. Logistic Regression

Dữ liệu chẩn đoán (diagnosis) được chuyển đổi từ 'M' (ác tính) và 'B' (lành tính) thành 1 và 0, tương ứng, để phù hợp với mô hình hồi quy logistic.

Hàm df.describe() được sử dụng để xem các thông số thống kê cơ bản của dữ liệu, giúp hiểu rõ hơn về phân phối của các đặc trưng.

Biểu đồ histogram của cột diagnosis được vẽ để xem xét sự phân bố của các trường hợp ác tính và lành tính trong tập dữ liệu.

Dữ liệu được chia thành hai tập: ‘traindf’ (dùng để huấn luyện mô hình) và ‘testdf’ (dùng để đánh giá mô hình).

Cụ thể hơn trong xây dựng mô hình, hàm ‘classification\_model’ được định nghĩa để huấn luyện mô hình hồi quy logistic. Các biến dự đoán (predictor\_var) được chọn là các đặc trưng trung bình của khối u. Biến mục tiêu (outcome\_var) là ‘diagnosis’ (chẩn đoán). Mô hình được huấn luyện trên tập ‘traindf’.

Mô hình Logistic Regression đạt độ chính xác khá cao trên tập dữ liệu lớn, khoảng 90%. Độ chính xác của mô hình trên tập kiểm tra là 90.452%. Kết quả này cho thấy mô hình có thể dự đoán và phân loại chính xác các trường hợp ung thư vú và lành tính trong tập huấn luyện.

Kỹ thuật Cross-Validation giúp đánh giá khả năng khái quát hóa của mô hình bằng cách chia tập huấn luyện thành nhiều phần (folds), lần lượt sử dụng từng phần để kiểm tra và phần còn lại để huấn luyện. Sau đó, kết quả được trung bình để có được độ chính xác tổng quát hơn. Trong đoạn code, kết quả các giá trị Cross-Validtion thu được sau khi xác thực chéo 5 folds (88.750%, 90.000%, 89.583%, 89.972%, 90.206%) được sử dụng và điểm số trung bình đạt được khoảng 89.972%. Điểm số này gần với độ chính xác trên tập huấn luyện, cho thấy mô hình có khả năng khái quát hóa tốt và không bị overfitting.

1. Decision Tree

Với ID3, sử dụng entropy để đo lường độ không chắc chắn của tập dữ liệu. Entropy càng cao thì độ không chắc chắn càng lớn. Về Information Gain, ID3 chọn thuộc tính có information gain lớn nhất (thuộc tính làm giảm entropy nhiều nhất sau khi phân chia) làm nút gốc của cây. Trong quá trình xây dựng cây, ID3 tiếp tục xây dựng cây bằng cách đệ quy chọn thuộc tính tốt nhất cho mỗi nút con, cho đến khi tất cả các mẫu trong một nút đều thuộc cùng một lớp hoặc đạt đến độ sâu tối đa (max\_depth = 2 trong trường hợp này).

Kết quả đánh giá:

* Độ chính xác (accuracy): 90% trên tập kiểm tra cho thấy mô hình dự đoán khá tốt kết quả chẩn đoán ung thư vú.
* Độ nhạy (recall): 100% trên tập kiểm tra cho thấy mô hình có khả năng phát hiện các trường hợp ung thư vú (dương tính thật) rất tốt, không bỏ sót trường hợp nào.
* Độ chính xác dự đoán (precision): 85.71% trên tập kiểm tra cho thấy rằng khi mô hình dự đoán một mẫu là dương tính, thì có khoảng 85.71% khả năng dự đoán đó là chính xác.
* F1 Score: 92.31% là trung bình điều hòa giữa độ chính xác và độ nhạy, cho thấy mô hình có sự cân bằng tốt giữa việc phát hiện đúng các trường hợp dương tính và giảm thiểu dự đoán sai.

Với C4.5, sử dụng gain ratio để khắc phục nhược điểm của ID3 khi ưu tiên các thuộc tính có nhiều giá trị. Gain ratio là information gain được chia cho entropy của chính thuộc tính đó. C4.5 có khả năng xử lý các thuộc tính liên tục bằng cách tìm ngưỡng tốt nhất để phân chia tập dữ liệu dựa trên các thuộc tính liên tục. C4.5 có thể cắt tỉa cây để giảm overfitting, giúp mô hình khái quát hóa tốt hơn trên dữ liệu mới. Quá trình xây dựng cây dừng lại khi tất cả các mẫu trong một nút thuộc cùng một lớp (lành tính hoặc ác tính), hoặc khi đạt đến độ sâu tối đa (max\_depth = 3 trong trường hợp này).

Kết quả đánh giá:

* Độ chính xác (accuracy): 85% trên tập kiểm tra cho thấy mô hình dự đoán khá tốt kết quả chẩn đoán ung thư vú.
* Độ nhạy (recall): 100% trên tập kiểm tra cho thấy mô hình có khả năng phát hiện các trường hợp ung thư vú (dương tính thật) rất tốt, không bỏ sót trường hợp nào.
* Độ chính xác dự đoán (precision): 80% trên tập kiểm tra cho thấy rằng khi mô hình dự đoán một mẫu là dương tính, thì có khoảng 80% khả năng dự đoán đó là chính xác.
* F1 Score: 88.89% là trung bình điều hòa giữa độ chính xác và độ nhạy, cho thấy mô hình có sự cân bằng tốt giữa việc phát hiện đúng các trường hợp dương tính và giảm thiểu dự đoán sai.

Mô hình C4.5 đạt hiệu quả tốt trong việc dự đoán ung thư vú, đặc biệt là khả năng phát hiện tất cả các trường hợp ác tính (độ nhạy 100%). Tuy nhiên, độ chính xác dự đoán chỉ đạt 80%, cho thấy cần cải thiện khả năng giảm thiểu việc dự đoán nhầm các trường hợp lành tính thành ác tính. Có thể cải thiện mô hình bằng cách tinh chỉnh các tham số như min\_samples\_split và max\_depth, hoặc sử dụng các kỹ thuật khác như cắt tỉa cây (pruning) để giảm overfitting.

1. Random Forest

Tạo lớp ‘RandomForestClassifier’: Khởi tạo với số lượng cây (n\_estimators), độ sâu tối đa của cây (max\_depth) và trạng thái ngẫu nhiên (random\_state). Hàm ‘fit’ huấn luyện từng cây quyết định trên các tập mẫu ngẫu nhiên từ dữ liệu. Hàm ‘create\_decision\_tree’, tạo cây quyết định sử dụng DecisionTreeClassifier từ scikit-learn. Hàm ‘predict’ Dự đoán bằng cách tổng hợp kết quả dự đoán từ tất cả các cây và sử dụng ngưỡng 0.5 để phân loại.

Huấn luyện và dự đoán: tạo một đối tượng RandomForestClassifier và huấn luyện mô hình trên tập huấn luyện (X\_train, Y\_train). Thực hiện dự đoán trên tập kiểm tra (X\_test) và lưu kết quả vào Y\_pred. Sử dụng scikit-learn: ta tạo một đối tượng RandomForestClassifier (có thể tùy chỉnh tham số). Huấn luyện mô hình và đánh giá độ chính xác trên tập kiểm tra (X\_test, Y\_test).

Kết quả độ chính xác (accuracy) đạt được khoảng 95.61% khi tự triển khai và 96.49% khi sử dụng scikit-learn. Độ chính xác cao cho thấy mô hình hoạt động tốt trong việc dự đoán ung thư vú dựa trên các đặc trưng được cung cấp. Mô hình Random Forest cho thấy hiệu quả tốt trong việc dự đoán ung thư vú, với độ chính xác trên 95%. Triển khai từ đầu giúp hiểu rõ hơn về thuật toán, trong khi sử dụng scikit-learn tiện lợi và nhanh chóng hơn.

### Kết luận

Dựa trên kết quả độ chính xác (accuracy) của 3 thuật toán có thể thấy rằng Random Forest với 95.61% khi tự triển khai và 96.49% khi sử dụng scikit-learn, có có hiệu suất tốt nhất trong việc dự đoán ung thư vú dựa trên tập dữ liệu này. Đối với Logistic Regression, thuật toán khá đơn giản để hiểu và triển khai. Nó có công thức toán học rõ ràng và việc huấn luyện mô hình thường nhanh chóng. Mô hình Logistic Regression có khả năng diễn giải cao, cung cấp các hệ số (coefficients) cho từng đặc trưng, giúp chúng ta hiểu được mức độ ảnh hưởng của từng đặc trưng đến kết quả dự đoán. Ví dụ, hệ số dương lớn cho biết đặc trưng đó làm tăng khả năng ung thư ác tính. Thuật toán còn có thể có thể xử lý tốt dữ liệu lớn vì nó không yêu cầu nhiều tài nguyên tính toán. Với Decision Tree, thuật toán phức tạp hơn Logistic Regression một chút, nhưng vẫn dễ hiểu và triển khai. Việc hình dung cây quyết định giúp dễ dàng giải thích quá trình ra quyết định của mô hình. Decision Tree có thể xử lý dữ liệu lớn, nhưng việc huấn luyện và dự đoán có thể chậm hơn so với Logistic Regression khi số lượng đặc trưng và mẫu dữ liệu tăng lên. Còn với Random Forest, thuật toán này phức tạp hơn hai thuật toán trên vì nó là một tập hợp của nhiều cây quyết định. Việc hiểu và tinh chỉnh các siêu tham số của Random Forest có thể đòi hỏi kiến thức chuyên sâu hơn. Mặc dù mỗi cây quyết định trong rừng ngẫu nhiên có thể được diễn giải, nhưng việc diễn giải toàn bộ mô hình Random Forest là khó khăn hơn. Tương tự như Decision Tree, Random Forest có thể xử lý dữ liệu lớn nhưng có thể chậm hơn Logistic Regression.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí** | **Logistic Regression** | **Decision Tree** | **Random Forest** |
| **Thực hiện đơn giản** | Cao | Trung bình | Thấp |
| **Khả năng diễn giải** | Cao | Cao | Thấp |
| **Độ chính xác tiềm năng** | Trung bình | Cao | Cao |
| **Xử lý dữ liệu lớn** | Tốt | Khá | Khá |