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

**KHOA CNTT**

----------

****

**BÁO CÁO MÔN HỌC**

**KHO DỮ LIỆU VÀ KHAI PHÁ DỮ LIỆU**

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

*vào việc đánh giá chất lượng không khí*

**Giảng viên hướng dẫn: Nguyễn Quỳnh Chi**

**Nhóm môn học: 01**

**Nhóm BTL: 07**

**Sinh viên thực hiện:**

1. Đỗ Hùng Cường B19DCCN078

2. Lê Huy Duy B19DCCN138

3. Nguyễn Như Mạnh B19DCCN421

4. Nguyễn Tuấn Anh B19DCCN030

5. Trần Đình Hoan B19DCCN270

Hà Nội, Tháng 5/2023

**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: 1](#_Toc135119364)

[1. Phân chia công việc: 1](#_Toc135119365)

[2. Bài toán đặt ra: 1](#_Toc135119366)

[3. Hướng giải quyết: 1](#_Toc135119367)

[B. Nội dung: 1](#_Toc135119368)

[I. Giới thiệu mô hình: 1](#_Toc135119369)

[1. Cây quyết định: 1](#_Toc135119370)

[2. Random forest: 2](#_Toc135119371)

[3. Bài toán đề ra: 3](#_Toc135119372)

[4. Phương pháp giải quyết: 3](#_Toc135119373)

[II. Dữ liệu bài toán: 4](#_Toc135119374)

[1. Thu thập dữ liệu: 4](#_Toc135119375)

[2. Mô tả dữ liệu: 4](#_Toc135119376)

[3. Tiền xử lý dữ liệu: 5](#_Toc135119377)

[III. Thuật toán: 5](#_Toc135119378)

[1. Thuật toán Decision Tree Regression: 6](#_Toc135119379)

[2. Thuật toán Random Forest: 9](#_Toc135119380)

[IV. Triển khai thuật toán: 10](#_Toc135119381)

[V. Kết quả: 18](#_Toc135119382)

[VI. Kết luận: 19](#_Toc135119383)

[VII. Tài liệu tham khảo: 19](#_Toc135119384)

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

|  |  |  |
| --- | --- | --- |
| 1 | - Giới thiệu về mô hình Cây quyết định, Random Forest (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 | T.Anh |
| 2 | - Nêu về nguồn gốc dữ liệu  - Mô tả các đặc điểm, các cột dữ liệu của dataset | Hoan |
| 3 | - Nêu các bước tiền xử lý  - Code phần tiền xử lý | Hoan |
| 4 | - Nêu quá trình xử lý thuật toán Decision Tree  - Code triển khai thuật toán Decision Tree | Cường |
| 5 | - Nêu quá trình xử lý thuật toán Random Forest  - Code triển khai thuật toán Random Forest | Duy |
| 6 | - 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. | Duy |
| 7 | - Làm slide báo cáo.  - Chỉnh sửa báo cáo. | Mạnh |
| 8 | - Đánh giá thuật toán, đưa ra kết luận | Tất cả thành viên |

1. **Bài toán đặt ra:**

Hiện nay vấn đề ô nhiễm không khí 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 được chất lượng không khí.

1. **Hướng giải quyết:**

Sử dụng các thuật toán hồi quy Desion Tree Regression và Random Forest để xác đinh đầu ra dựa trên các đặc trưng đầu vào. Đây là các thuật toán học máy có giám sát quan trọng và phổ biến.

1. **Nội dung:**
2. **Giới thiệu mô hình:**
3. Cây quyết định:

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).

1. 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.

1. Bài toán đề ra:

Trước tình trạng biến đổi khí hậu ngày càng trở nên nghiêm trọng và gây hại cho con người. Không những là thời tiết cực đoan mà không khí cũng ngày càng trở nên độc hại hơn với sức khỏe con người.

Bài toán đề ra ở đây phân tích nồng độ các khí trong không khí của thành phố Ahmedabad và đưa ra kết quả.

1. Phương pháp giải quyết:

* **Thu thập dữ liệu**: Thu thập các dữ liệu liên quan đến nồng độ không khí, bao gồm các đặc trưng như nhiệt độ, độ ẩm, tốc độ gió, áp suất không khí, và các yếu tố môi trường khác.
* **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 của nồng độ không khí. 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.
* **So sánh hiệu suất của hai mô hình**: So sánh hiệu suất của hai mô hình cây quyết định và Random Forest 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.

1. **Dữ liệu bài toán:**
2. **Thu thập dữ liệu:**

* Dữ liệu được lấy từ: <https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india>
* Tên dataset: Air Quality Data in India (2015 - 2020)

1. **Mô tả dữ liệu:**

Những dữ liệu chứa dữ liệu chất lượng không khí và AQI (Chỉ số chất lượng không khí) ở mức hàng giờ và hàng ngày của các trạm khác nhau trên nhiều thành phố ở Ấn Độ. Phân tích xác định được 12 chất khí.

- Số bản ghi: 29531

- Số lượng thuộc tính: 16

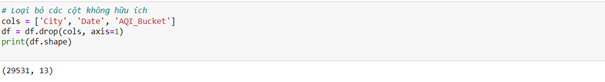
- Thuộc tính:

* City
* Date
* PM2.5
* PM10
* NO
* NO2
* NOx
* NH3
* CO
* SO2
* O3
* Benzene
* Toluene
* Xylene
* AQI
* AQI\_Bucket

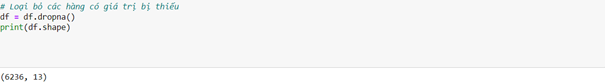
1. **Tiền xử lý dữ liệu:**

Tập dữ liệu “Air Quality Data in India (2015 - 2020)" 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 City, Date và AQI\_Bucket. Thì dữ liệu từ 16 cột còn lại 13 cột



* Loại bỏ các hàng có giá trị bị thiếu. Dữ liệu từ 29531 hàng còn lại 6236 hàng



1. **Thuật toán:**

Ý tưởng: Để dự đoán chỉ số chất lượng không khí (AQI) cho bộ dữ liệu của trang web trên, ta có thể sử dụng mô hình Random Forest và Decision Tree như sau:

+ Tiền xử lý dữ liệu: loại bỏ các giá trị thiếu, chuẩn hóa dữ liệu nếu cần thiết.

+ Tách tập huấn luyện và tập kiểm tra: sử dụng tập huấn luyện để huấn luyện mô hình và tập kiểm tra để kiểm tra độ chính xác của mô hình.

+ Huấn luyện mô hình Decision Tree: sử dụng tập huấn luyện để huấn luyện mô hình Decision Tree. Bạn có thể tinh chỉnh các siêu tham số như độ sâu cây, số lượng mẫu tối thiểu để phân chia một nút.

+ Huấn luyện mô hình Random Forest: sử dụng tập huấn luyện để huấn luyện mô hình Random Forest. Bạn có thể tinh chỉnh các siêu tham số như số lượng cây, độ sâu cây, số lượng mẫu tối thiểu để phân chia một nút và số lượng đặc trưng được chọn ngẫu nhiên cho mỗi cây.

+ Kiểm tra độ chính xác của mô hình: sử dụng tập kiểm tra để kiểm tra độ chính xác của mô hình Decision Tree và Random Forest. Bạn có thể sử dụng các chỉ số đánh giá như RMSE, R-square, MAE để đánh giá độ chính xác của mô hình.

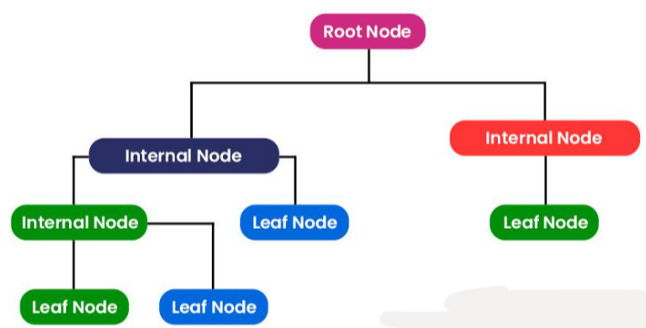
+ Sử dụng mô hình để dự đoán AQI cho các giá trị mới: sau khi huấn luyện mô hình, bạn có thể sử dụng mô hình để dự đoán AQI cho các giá trị mới bằng cách đưa các giá trị đó vào mô hình để lấy dự đoán.

1. Thuật toán Decision Tree Regression:

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:

* B1: Từ bộ dữ liệu ban đầu, tìm thuộc tính (feature) và giá trị ngưỡng (threshold) tốt nhất để chia dữ liệu thành hai phần (splitting) sao cho phân chia này giúp giảm thiểu sai số của mô hình.
* B2: Tách dữ liệu thành hai phần dựa trên thuộc tính (feature) và giá trị ngưỡng (threshold) được chọn, và tạo ra hai nhánh con của cây.
* B3: Lặp lại quá trình trên trên các nhánh con cho đến khi các điều kiện dừng được thỏa mãn, ví dụ như số lượng bản ghi trong một nút không còn đủ lớn hoặc phân nhánh đó đã đạt được độ sâu (depth) tối đa.
* B4: Khi một nút dừng lại, ta tính toán giá trị đầu ra cho nút đó bằng cách lấy trung bình (đối với bài toán hồi quy) của các giá trị đầu ra của các điểm dữ liệu trong nút.
* B5: Cuối cùng, ta có được một cây quyết định hoàn chỉnh, có thể sử dụng để dự đoán đầu ra cho các điểm dữ liệu mới bằng cách đi từ gốc (root) của cây đến lá (leaf) mà điểm dữ liệu đó thuộc về, sau đó trả về kết quả dựa theo giá trị (value) của lá đó.
* Nhận xét về thuật toán: Kết quả dự đoán có thể khác nhau trong các lần chạy thuật toán vì cây quyết định được học và xây dựng trên một tập dữ liệu cụ thể, và các cách chia cây quyết định khác nhau có thể được tạo ra bằng cách thay đổi thứ tự các thuộc tính và giá trị ngưỡng để chia. Ngoài ra, sự khác biệt trong các tham số của thuật toán như chiều sâu tối đa của cây cũng có thể ảnh hưởng đến kết quả dự đoán.
* Ưu điểm:
* Dễ hiểu và giải thích: Mô hình cây quyết định có cấu trúc rõ ràng và dễ hiểu. Việc giải thích dự đoán của mô hình cho các quyết định hồi quy cũng rất đơn giản, bằng cách theo dõi các quy tắc đơn giản từ các nút gốc đến nút lá.
* Khả năng tùy chỉnh mức độ phức tạp: Các siêu tham số trong mô hình cây quyết định như độ sâu cây, số lượng mẫu tối thiểu để tách một nút, số lượng mẫu tối thiểu để tạo một lá,.. được điều chỉnh để tăng hoặc giảm độ phức tạp của cây và tùy chỉnh mô hình cho từng bài toán cụ thể.
* Khả năng xác định các đặc trưng quan trọng: Các cây quyết định có thể đánh giá mức độ quan trọng của các đặc trưng trong mô hình, giúp người dùng có thể hiểu được tầm quan trọng của các đặc trưng và lựa chọn các đặc trưng quan trọng cho mô hình.
* Nhược điểm:
* Không ổn định: Một số thay đổi nhỏ trong dữ liệu đầu vào có thể dẫn đến sự thay đổi lớn trong cây quyết định, điều này làm cho mô hình khá không ổn định.
* Thời gian tính toán có thể là một nhược điểm đối với các tập dữ liệu lớn hoặc phức tạp. Vì quá trình xây dựng cây quyết định có thể đòi hỏi phải duyệt qua toàn bộ tập dữ liệu và tính toán các giá trị liên quan đến các nút trên cây. Tuy nhiên, với các tập dữ liệu nhỏ hoặc trung bình, quá trình huấn luyện và dự đoán của Decision Tree Regression vẫn khá nhanh chóng.

Điều kiện chia nhánh trong Decision Tree Regression được xác định bằng cách chọn một ngưỡng giá trị của một biến đầu vào (feature) để tạo thành các nhánh con sao cho sai số giữa các điểm dữ liệu và giá trị dự đoán trung bình trên mỗi nhánh con là nhỏ nhất. Khi tìm kiếm ngưỡng giá trị này, thuật toán sử dụng các độ đo khác nhau để đánh giá chất lượng phân chia, trong đó bao gồm cả sai số bình phương trung bình (MSE). Công thức của MSE như sau:

Trong đó:

là số mẫu trong tập huấn luyện.

: là trung bình các giá trị thực của các mẫu trong bộ huấn luyện

: là giá trị thực của các mẫu trong bộ huấn luyện

Độ đo MSE càng thấp thì mô hình càng tốt. Vì MSE đo lường mức độ sai lệch giữa giá trị dự đoán và giá trị thực tế của biến phụ thuộc. Khi chia dữ liệu thành các nhánh, mục tiêu là tìm ra những thuộc tính chia nhánh sao cho giá trị MSE là thấp nhất có thể. Khi giá trị MSE là thấp nhất, điều đó có nghĩa là mô hình của chúng ta có khả năng dự đoán tốt hơn. Do đó, chúng ta chọn MSE càng thấp để chia tách dữ liệu ra làm 2 nhánh để xây dựng cây quyết định hồi quy.

1. Thuật toán Random Forest:

* Random Forest là gì?

Random Forest là một thuật toán học máy dùng để giải quyết bài toán phân loại hoặc dự đoán giá trị số. Nó dựa trên một tập hợp các cây quyết định, gọi là cây quyết định ngẫu nhiên (Random Decision Tree), mà mỗi cây được huấn luyện trên một tập dữ liệu khác nhau nhưng có cùng cấu trúc.

Random Forest giải quyết vấn đề overfitting bằng cách giảm thiểu sự phụ thuộc quá mức vào các trường hợp dữ liệu riêng lẻ. Nó cũng giảm thiểu variance (độ lệch) bằng cách sử dụng nhiều cây quyết định và kết hợp chúng để đưa ra dự đoán.

Để huấn luyện mô hình Random Forest, trước hết, chúng ta phải chọn ngẫu nhiên một số lượng các mẫu từ tập dữ liệu và sử dụng chúng để huấn luyện mỗi cây quyết định. Chúng ta cũng cần chọn ngẫu nhiên một số lượng các đặc trưng từ tập dữ liệu cho mỗi cây quyết định sử dụng để huấn luyện.

Khi có một mẫu mới cần được dự đoán, các cây quyết định trong Random Forest được sử dụng để đưa ra dự đoán độc lập. Sau đó, dự đoán được đưa ra bằng cách lấy trung bình hoặc phương án đa số của các dự đoán của các cây quyết định.

Điều này giúp giảm thiểu sự ảnh hưởng của các trường hợp dữ liệu riêng lẻ và độ lệch của một cây quyết định đơn lẻ, do đó cải thiện độ chính xác của mô hình.

* Ưu điểm:

+ Cả hai mô hình đều được sử dụng rộng rãi và có thể cho kết quả tốt trong việc dự đoán các biến phụ thuộc trong dữ liệu.

+ Random Forest và Decision Tree có khả năng xử lý các tập dữ liệu lớn và cực kỳ phù hợp cho việc phân tích các tập dữ liệu với nhiều thuộc tính và giá trị bị thiếu.

+ Random Forest có khả năng chống lại tình trạng overfitting, giúp ngăn chặn việc mô hình quá khớp với dữ liệu huấn luyện.

+ Decision Tree là một mô hình khá dễ giải thích và hiểu, do đó giúp cho người dùng dễ dàng phân tích, kiểm tra và cải thiện mô hình.

* Nhược điểm:

+ Tuy nhiên, Random Forest có thể gặp vấn đề về khả năng giải thích kết quả dự đoán vì đó là kết quả của nhiều cây quyết định trong mô hình, không phải chỉ một cây như Decision Tree.

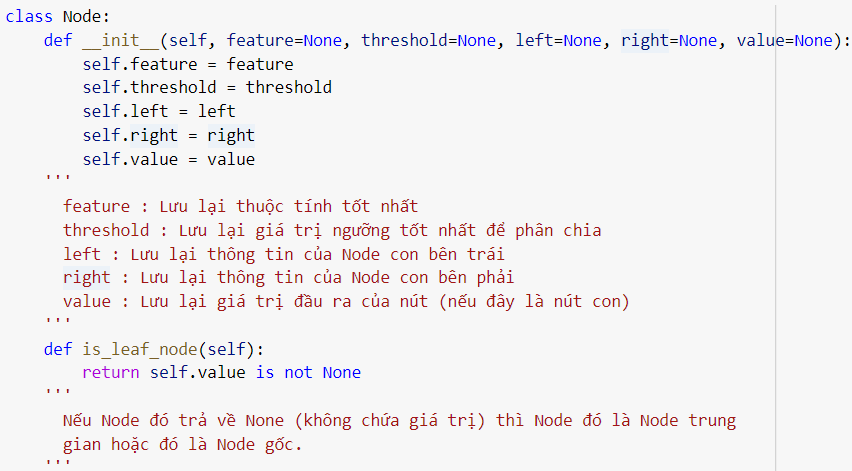
+ Decision Tree dễ bị tình trạng overfitting, khi có quá nhiều lá cây trong mô hình. Do đó, cần chú ý để điều chỉnh các tham số của mô hình để tránh tình trạng này.

+ Cả hai mô hình đều có thể bị ảnh hưởng bởi nhiễu trong dữ liệu, do đó cần phải xử lý nhiễu trước khi sử dụng mô hình để dự đoán kết quả chính xác hơn.

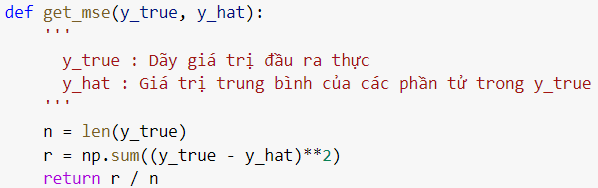
Dựa theo giá trị AQI sẽ kết luận chất lượng không khí tại thời điểm đó là tốt hay xấu, thông thường các giá trị này sẽ nằm trong khoảng từ 0 - 1000 :

* AQI (0 – 50) : Tốt
* 50 < AQI <= 100 : Khá tốt
* 100 < AQI <= 200 : Trung bình
* 200 < AQI <= 300 : Xấu
* 300 < AQI <= 400 : Rất xấu
* AQI > 400 : Nghiêm trọng

1. **Triển khai thuật toán:**
2. **Thuật toán Decision Tree Regression:**
3. Class Node (Đối tượng nút):

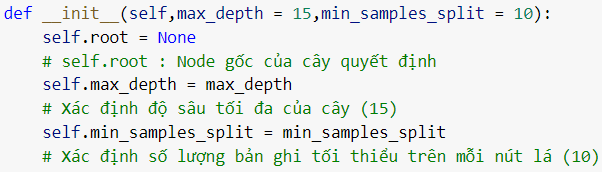
****

1. Hàm tính MSE (Mean Squared Error) :



1. Class Decision Tree Regression :

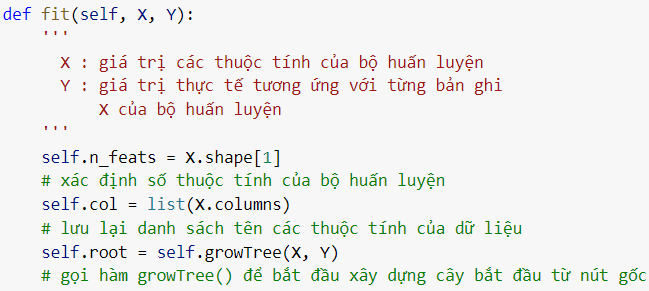
* Hàm khởi tạo :



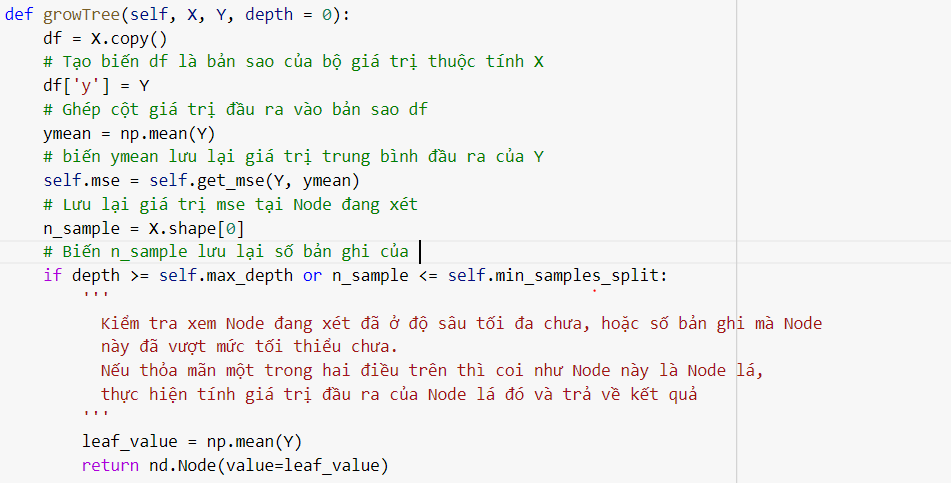
Khi áp dụng với từng bài toán cụ thể, ta có thể tùy chỉnh tham số max\_depth và min\_samples\_split để phù hợp hơn với mục tiêu kết quả đầu ra cũng như hiệu suất xử lý.

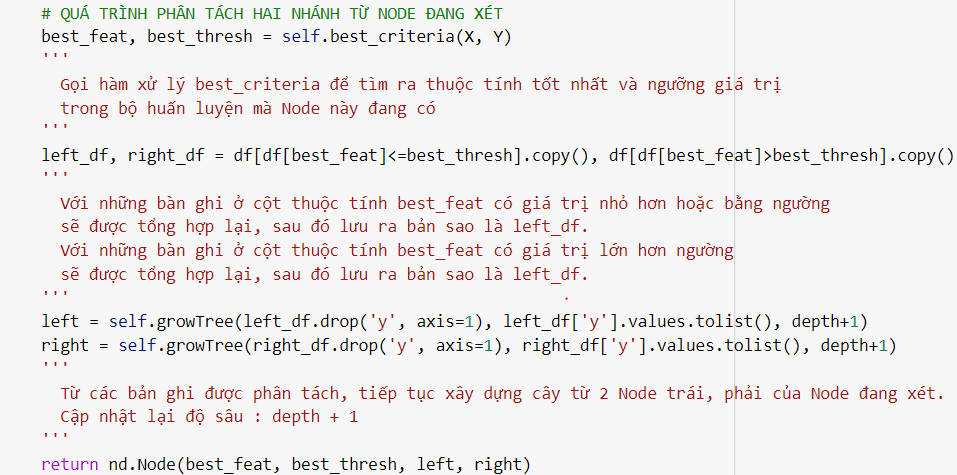
* Hàm xây dựng mô hình :

Hàm này sẽ xây dựng cây quyết định dựa trên dữ liệu huấn luyện đầu vào và các tham số được chỉ định.



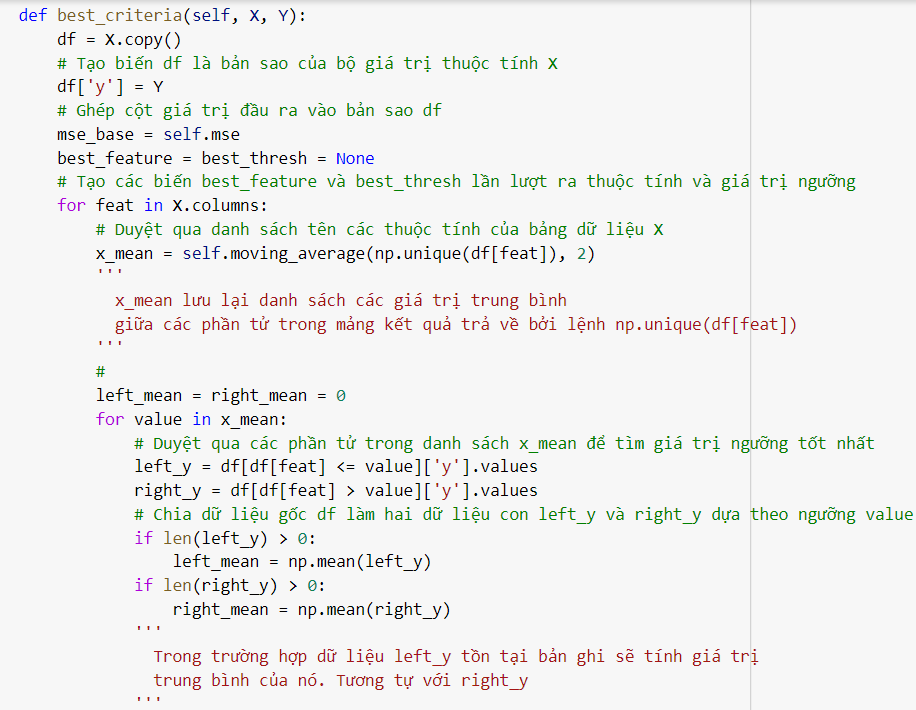
* Hàm phân nhánh :

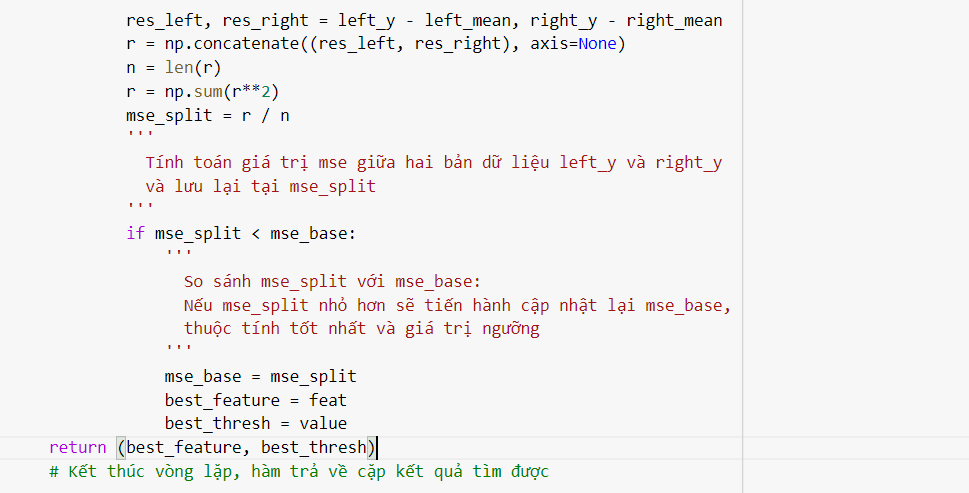




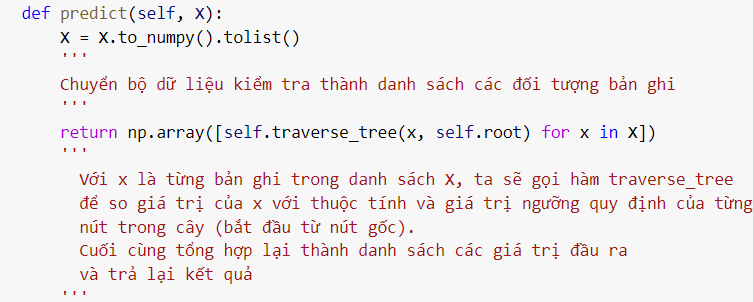
* Hàm lựa chọn thuộc tính tốt nhất và giá trị ngưỡng :



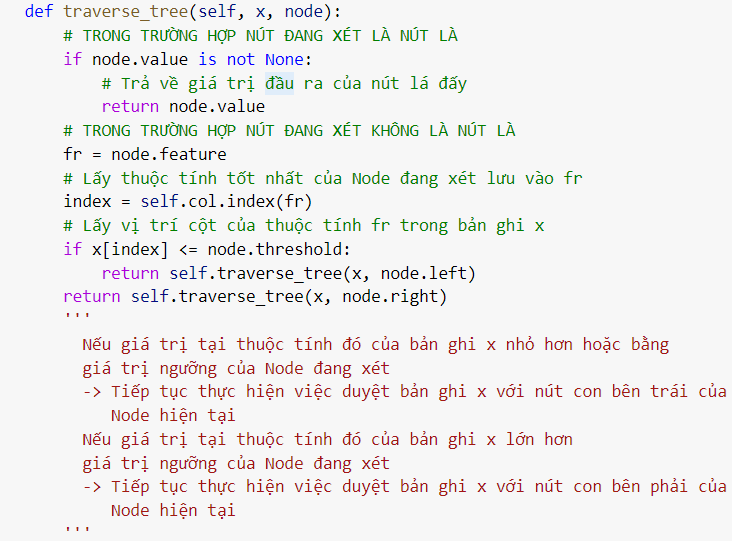




* Hàm dự đoán giá trị đầu ra cho mẫu dữ liệu mới

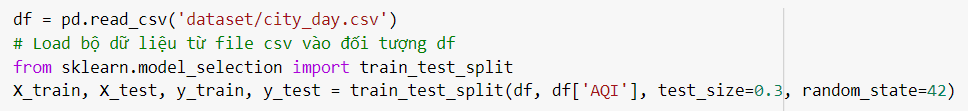


* Hàm duyệt cây :



1. Áp dụng vào bộ dữ liệu :

Từ bộ dữ liệu df, ta sẽ sử dụng hàm train\_test\_split từ thư viện sklearn.model\_selection để chia tập dữ liệu df thành 2 phần là tập huấn luyện (train) và tập kiểm tra (test). Chức năng chính của hàm train\_test\_split là trộn ngẫu nhiên dữ liệu và chia chúng thành 2 tập theo tỉ lệ được chỉ định bởi người dùng.

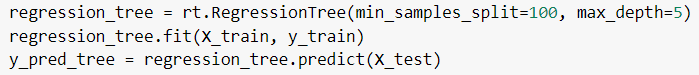


Bộ dữ liệu sẽ được chia ra thành các tập huấn luyện và kiểm tra theo tỉ lệ bản ghi là 80/20 (Tập huấn luyện chiếm 80% số bản ghi mấu, tập kiểm tra chiếm 20% số bản ghi mấu).

* X\_train : Tập dữ liệu huấn luyện chứa các giá trị thuộc tính
* X\_test : Tập dữ liệu kiểm tra cho chứa giá trị thuộc tính
* y\_train : Tập dữ liệu huấn luyện chứa các giá trị đầu ra
* y\_test : Tập dữ liệu kiểm tra chứa các giá trị đầu ra

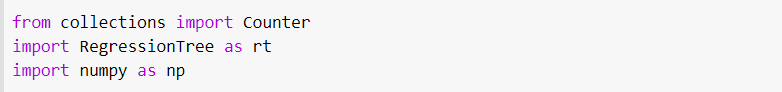
Sau đó là tạo ra đối tượng cây quyết định, huấn luyện và tiến hành đưa ra dự đoán như sau :

* Đầu tiên ta cần xác định các thông số về độ sâu tối đa và số bản ghi tối thiểu nằm trên mối nút lá lần lượt là 5 và 100.
* Từ bộ huấn luyện X\_train, y\_train, ta đưa vào mô hình thông qua hàm fit()
* Sau quá trình huấn luyện, ta cho bộ dữ liệu thuộc tính X\_test vào để mô hình đưa ra dự đoán theo từng bản ghi X\_test đó, kết quả là danh sách y\_pred\_tree. Và code minh họa như sau:

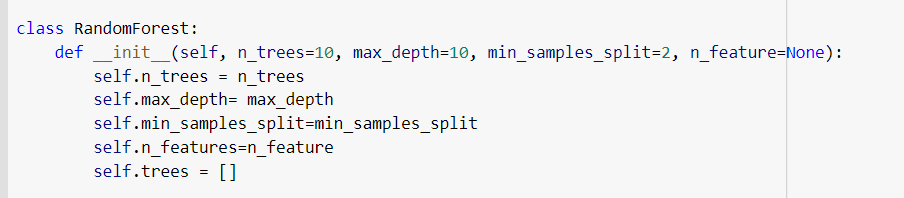


1. **Thuật toán Random Forest:**

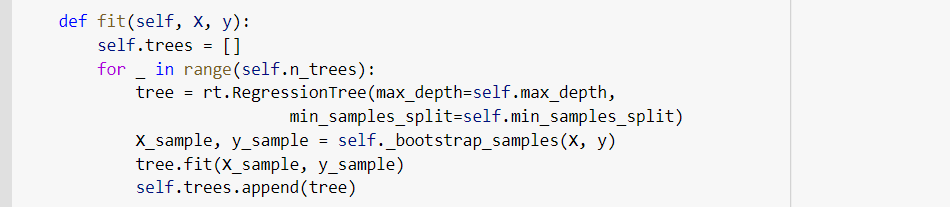
Đầu tiên khai báo các thư viện cần thiết được import vào, bao gồm: *collections* để đếm số lần xuất hiện của các phần tử, *numpy* để làm việc với mảng đa chiều và *RegressionTree* là module chứa cây hồi quy.



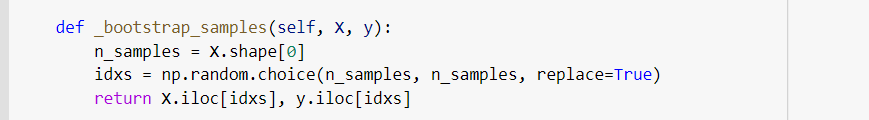
Lớp RandomForest được định nghĩa, có một số tham số được cấu hình trong hàm khởi tạo bao gồm: số lượng cây trong rừng (n\_trees), độ sâu tối đa của cây (max\_depth), số lượng mẫu tối thiểu cần phải có trong mỗi node để chia nhánh (min\_samples\_split) và số lượng đặc trưng được chọn ngẫu nhiên cho mỗi cây (n\_feature).



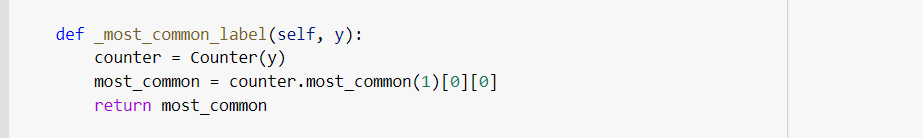
Phương thức fit được sử dụng để huấn luyện mô hình Random Forest, với đầu vào là ma trận đặc trưng X và vector nhãn y. Trong đó, mỗi cây trong rừng được tạo ra thông qua việc chọn ngẫu nhiên các mẫu và đặc trưng từ bộ dữ liệu huấn luyện, và sau đó fit vào mô hình RegressionTree.



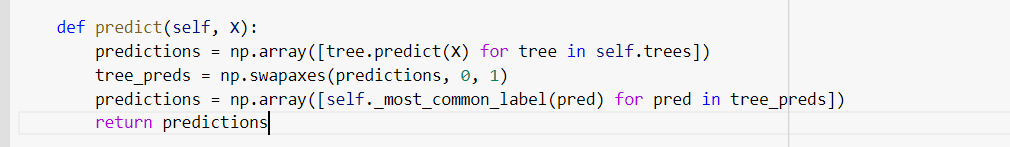
Phương thức \_bootstrap\_samples sử dụng kỹ thuật tái chọn mẫu với hoàn lại (bootstrap) để tạo ra một tập con các mẫu từ tập dữ liệu huấn luyện X và y.



Phương thức \_most\_common\_label dùng để trả về nhãn xuất hiện nhiều nhất trong một mảng nhãn đầu vào.

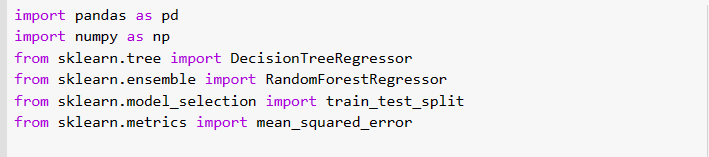


Phương thức predict dùng để dự đoán nhãn của các mẫu đầu vào X bằng cách trả về nhãn xuất hiện nhiều nhất của các dự đoán của từng cây trong rừng.

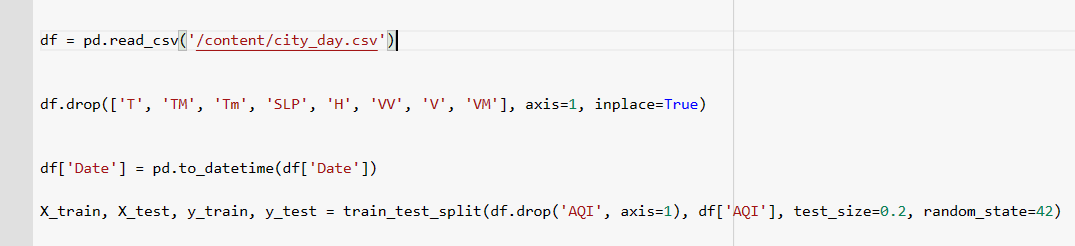


Xây dựng mô hình học máy trong Sklearn để mô hình hóa và dự đoán AQI bằng Decision Tree và Random Forest

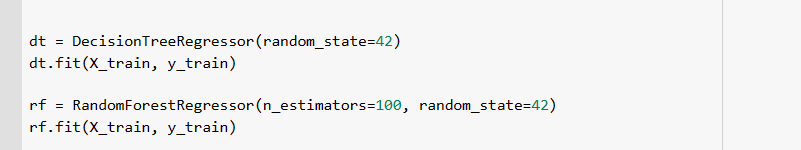
Ta import các thư viện cần thiết vào



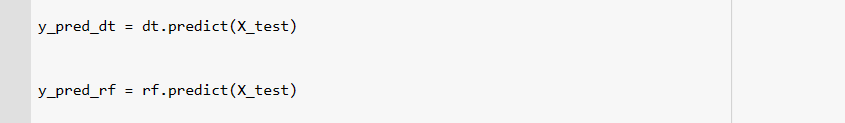
Ta đọc dữ liệu từ tập tin csv, loại bỏ các cột không cần thiết, chuyển đổi cột 'Date' thành kiểu dữ liệu ngày tháng và chia dữ liệu thành tập huấn luyện và tập kiểm tra



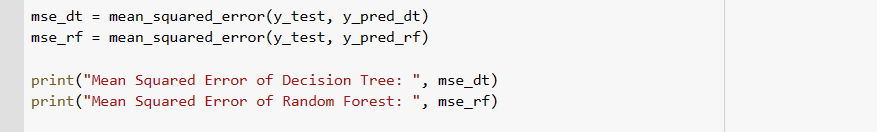
Sử dụng Decision Tree và Random Forest để huấn luyện mô hình



Dự đoán kết quả trên tập kiểm tra sử dụng Decision Tree và Random Forest



Tính toán độ lỗi trung bình trên tập kiểm tra sử dụng Decision Tree và Random Forest

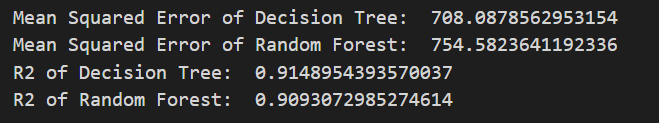


1. **Kết quả:**

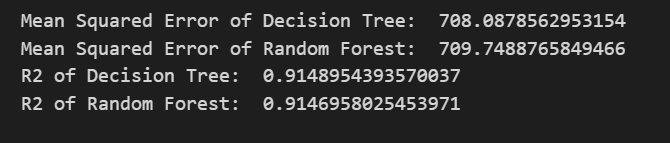
**1. Kết quả của code so với kết quả thực tế**

|  |  |
| --- | --- |
| **Thực tế** | **Dự đoán** |
| 90 | 85.722222 |
| 357 | 409.919355 |
| 106 | 103.632911 |
| 103 | 103.632911 |
| 118 | 113.534247 |
| 120 | 123.147059 |
| 150 | 159.710526 |
| 74 | 77.411765 |
| 597 | 409.919355 |
| 82 | 85.722222 |
| 38 | 43.947917 |
| 132 | 132.075758 |
| 128 | 132.075758 |
| 159 | 159.710526 |
| 128 | 132. 075758 |
| 249 | 237.558824 |
| 103 | 103.632911 |
| 102 | 103.632911 |
| 63 | 59.177778 |
| 127 | 123.147059 |

* Kết quả khi sử dụng thư viện:



* Kết quả mô hình code:



1. **Kết luận:**

Decision Tree:

* Decision Tree có thể được xây dựng dễ dàng và hiệu quả.
* Nếu không được điều chỉnh cẩn thận, Decision Tree có nguy cơ cao bị overfitting dữ liệu huấn luyện.
* Decision Tree thường thực hiện tốt trên dữ liệu có cấu trúc đơn giản và không cân bằng.
* Khi số lượng đặc trưng lớn và phức tạp, Decision Tree có thể không cho kết quả tốt như các thuật toán học máy khác.

Random Forest:

* Random Forest là một phương pháp ensemble learning dựa trên Decision Tree, do đó nó kết hợp sự mạnh mẽ của nhiều cây quyết định.
* Random Forest có khả năng giảm overfitting và cung cấp tính tổng quát hóa cao hơn so với Decision Tree đơn lẻ.
* Với một tập dữ liệu lớn và phức tạp, Random Forest thường cho kết quả tốt và ổn định hơn Decision Tree.
* Random Forest có khả năng xếp hạng quan trọng đặc trưng, giúp đánh giá ảnh hưởng của các đặc trưng đối với kết quả dự đoán.

1. **Tài liệu tham khảo:**

Bài giảng môn Kho dữ liệu và khai phá kho dữ liệu – GV Nguyễn Quỳnh Chi.