TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Nguyễn Trung Kiên – 52100427**

**Nguyễn Trung Kiên – 52100752**

**NGUYỄN THỊ KIM TIẾN – 52100491**

**ĐINH PHƯƠNG MY – 52100703**

**Đặng Hoàng Trúc Duyên – 52100404**

**Ngô Tấn Lợi – 52100439**

**Ngô Đức Anh Tuấn – 51800951**

**THUẬT TOÁN CART**

**BÁO CÁO CUỐI KỲ**

**KHAI THÁC DỮ LIỆU VÀ KHAI PHÁ   
TRI THỨC**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Nguyễn Trung Kiên – 52100427**

**Nguyễn Trung Kiên – 52100752**

**NGUYỄN THỊ KIM TIẾN – 52100491**

**ĐINH PHƯƠNG MY – 52100703**

**Đặng Hoàng Trúc Duyên – 52100404**

**Ngô Tấn Lợi – 52100439**

**Ngô Đức Anh Tuấn – 51800951**

**THUẬT TOÁN CART**

**BÁO CÁO CUỐI KỲ**

**KHAI THÁC DỮ LIỆU VÀ KHAI PHÁ**

**TRI THỨC**

Giảng viên hướng dẫn

**TS. Hoàng Anh**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Lời nói đầu tiên, nhóm 05 xin được gửi lời cảm ơn chân thành đến toàn bộ giảng viên Trường Đại học Tôn Đức Thắng nói chung cũng như toàn bộ giảng viên Khoa Công nghệ thông tin nói riêng vì đã tạo điều kiện cho nhóm 05 được học bộ môn Khai thác dữ liệu và khai phá tri thức.

Và đặc biệt, nhóm 05 xin gửi lời cảm ơn chân thành nhất đến thầy Hoàng Anh – giảng viên giảng dạy và hướng dẫn cho đề tài môn Khai thác dữ liệu và khai phá tri thức. Trong suốt quá trình học tập và thực hiện bài báo cáo, thầy luôn giúp đỡ, chỉ bảo tận tình để nhóm có thể tìm ra cách giải quyết những vướng mắc gặp phải và hoàn thiện đề tài này một cách tốt nhất. Lượng kiến thức này chúng em sẽ làm hành trang để áp dụng vào công việc sau này.

Không thể nói gì hơn nữa, một lần nữa, bằng cả tấm lòng, nhóm 05 xin chân thành gửi lời cảm ơn sâu sắc đến thầy – người đã dìu dắt lớp trong suốt chặng đường vừa qua.

*TP. Hồ Chí Minh, ngày 26 tháng 12 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Nguyễn Trung Kiên*

*Nguyễn Trung Kiên*

*Nguyễn Thị Kim Tiến*

*Đinh Phương My*

*Đặng Hoàng Trúc Duyên*

*Ngô Tấn Lợi*

*Ngô Đức Anh Tuấn*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Nhóm 05 xin cam đoan đây là công trình nghiên cứu của riêng nhóm và được sự hướng dẫn khoa học của TS Hoàng Anh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong báo cáo còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào nhóm xin hoàn toàn chịu trách nhiệm về nội dung báo cáo của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do nhóm gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 26 tháng 12 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Nguyễn Trung Kiên*

*Nguyễn Trung Kiên*

*Nguyễn Thị Kim Tiến*

*Đinh Phương My*

*Đặng Hoàng Trúc Duyên*

*Ngô Tấn Lợi*

*Ngô Đức Anh Tuấn*

**MỤC LỤC**

[DANH MỤC HÌNH VẼ vi](#_Toc185852553)

[DANH MỤC BẢNG BIỂU viii](#_Toc185852554)

[DANH MỤC CÁC CHỮ VIẾT TẮT ix](#_Toc185852555)

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 1](#_Toc185852556)

[1.1 Lý do chọn đề tài 1](#_Toc185852557)

[1.2 Mục tiêu thực hiện 1](#_Toc185852558)

[1.3 Phạm vi nghiên cứu 2](#_Toc185852559)

[1.4 Phương pháp thực hiện 2](#_Toc185852560)

[CHƯƠNG 2. cơ sở lý thuyết 3](#_Toc185852561)

[2.1 Định nghĩa thuật toán 3](#_Toc185852562)

[2.2 Cách thức hoạt động 4](#_Toc185852563)

[2.3 Tiêu chí phân chia 6](#_Toc185852564)

[2.3.1 Chỉ số Gini 6](#_Toc185852565)

[2.3.2 Giảm phương sai 6](#_Toc185852566)

[2.3.3 Cách chọn tiêu chí 7](#_Toc185852567)

[2.4 Ứng dụng của thuật toán 8](#_Toc185852568)

[2.4.1 CART cho phân loại 8](#_Toc185852569)

[2.4.2 CART cho hồi quy 8](#_Toc185852570)

[2.5 Ưu điểm và nhược điểm 9](#_Toc185852571)

[2.5.1 Ưu điểm 9](#_Toc185852572)

[2.5.2 Nhược điểm 9](#_Toc185852573)

[2.6 So sánh với các thuật toán khác 10](#_Toc185852574)

[CHƯƠNG 3. bài toán dự đoán lượng mưa 11](#_Toc185852575)

[3.1 Đặt vấn đề 11](#_Toc185852576)

[3.2 Mô tả bài toán 11](#_Toc185852577)

[3.3 Phương pháp áp dụng 13](#_Toc185852578)

[CHƯƠNG 4. THỰC NGHIỆM 15](#_Toc185852579)

[4.1 Chuẩn bị dữ liệu 15](#_Toc185852580)

[4.1.1 Cài đặt thư viện 15](#_Toc185852581)

[4.1.2 Đọc và kiểm tra dữ liệu 16](#_Toc185852582)

[4.1.3 Xử lý dữ liệu thiếu 17](#_Toc185852583)

[4.1.4 Chuẩn hoá đặc trưng dạng số 18](#_Toc185852584)

[4.2 Trực quan hoá dữ liệu 18](#_Toc185852585)

[4.2.1 Scatter Plot 18](#_Toc185852586)

[4.2.2 Ma trận tương quan 19](#_Toc185852587)

[4.2.3 Boxplot 19](#_Toc185852588)

[4.3 Huấn luyện mô hình phân loại 20](#_Toc185852589)

[4.3.1 Tiền xử lý dữ liệu 20](#_Toc185852590)

[4.3.2 Huấn luyện mô hình 20](#_Toc185852591)

[4.3.3 Đánh giá mô hình 21](#_Toc185852592)

[4.3.4 Trực quan hóa kết quả 22](#_Toc185852593)

[4.4 Huấn luyện mô hình hồi quy 23](#_Toc185852594)

[4.4.1 Tiền xử lý dữ liệu 23](#_Toc185852595)

[4.4.2 Huấn luyện mô hình 23](#_Toc185852596)

[4.4.3 Đánh giá mô hình 24](#_Toc185852597)

[4.4.4 Trực quan hóa kết quả 25](#_Toc185852598)

[4.5 So sánh với những thuật toán khác 25](#_Toc185852599)

[CHƯƠNG 5. Kết quả thực nghiệm 27](#_Toc185852600)

[5.1 Tổng quan dữ liệu 27](#_Toc185852601)

[5.2 Trực quan hoá dữ liệu 29](#_Toc185852602)

[5.3 Huấn luyện mô hình phân loại 31](#_Toc185852603)

[5.4 Huấn luyện mô hình hồi quy 32](#_Toc185852604)

[5.5 So sánh với những thuật toán khác 33](#_Toc185852605)

[5.5.1 So sánh hiệu suất 33](#_Toc185852606)

[5.5.2 Ý nghĩa kết quả 34](#_Toc185852607)

[CHƯƠNG 6. KẾT LUẬN 35](#_Toc185852608)

[6.1 Kết quả đạt được 35](#_Toc185852609)

[6.2 Thuận lợi và khó khăn 35](#_Toc185852610)

[6.2.1 Thuận lợi 35](#_Toc185852611)

[6.2.2 Khó khăn 35](#_Toc185852612)

[6.3 Hướng phát triển 36](#_Toc185852613)

[6.4 Phân công công việc 36](#_Toc185852614)

[TÀI LIỆU THAM KHẢO 37](#_Toc185852615)

# DANH MỤC HÌNH VẼ

[Hình 2.1: Ví dụ về cấu trúc cây 3](#_Toc185852520)

[Hình 2.2: Một ví dụ về cây quyết định 5](#_Toc185852521)

[Hình 3.1: Bộ dữ liệu Corgis Weather Dataset 11](#_Toc185852522)

[Hình 4.1: Đoạn mã khai báo thư viện 15](#_Toc185852523)

[Hình 4.2: Đoạn mã đọc dữ liệu 16](#_Toc185852524)

[Hình 4.3: Đoạn mã hiển thị dữ liệu 16](#_Toc185852525)

[Hình 4.4: Đoạn mã dữ lý dữ liệu bị thiếu 17](#_Toc185852526)

[Hình 4.5: Đoạn mã chuẩn hoá dạng số 18](#_Toc185852527)

[Hình 4.6: Đoạn mã minh hoạ bằng Scatter Plot 18](#_Toc185852528)

[Hình 4.7: Đoạn mã minh hoạ bằng ma trận tương quan 19](#_Toc185852529)

[Hình 4.8: Đoạn mã minh hoạ bằng Boxplot 19](#_Toc185852530)

[Hình 4.9: Đoạn mã kiểm tra dữ liệu 20](#_Toc185852531)

[Hình 4.10: Đoạn mã chia dữ liệu và huấn luyện mô hình 21](#_Toc185852532)

[Hình 4.11: Đoạn mã in kết quả phân loại 21](#_Toc185852533)

[Hình 4.12: Đoạn mã vẽ cây quyết định phân loại 22](#_Toc185852534)

[Hình 4.13: Đoạn mã kiểm tra dữ liệu hồi quy 23](#_Toc185852535)

[Hình 4.14: Đoạn mã chia dữ liệu và huấn luyện mô hình hồi quy 24](#_Toc185852536)

[Hình 4.15: Đoạn mã kết quả và biểu đồ so sánh hồi quy 24](#_Toc185852537)

[Hình 4.16: Đoạn mã vẽ cây quyết định hồi quy 25](#_Toc185852538)

[Hình 4.17: Đoạn mã so sánh thuật toán CART với các thuật toán khác 26](#_Toc185852539)

[Hình 5.1: Dữ liệu 5 dòng đầu tiên của tập Corgis Weather Dataset 27](#_Toc185852540)

[Hình 5.2: Thông tin về các cột và kiểu dữ liệu 28](#_Toc185852541)

[Hình 5.3: Thống kê cơ bản cho các cột dạng số 28](#_Toc185852542)

[Hình 5.4: Biểu đồ Scatter Plot 29](#_Toc185852543)

[Hình 5.5: Ma trận tương quan 30](#_Toc185852544)

[Hình 5.6: Biểu đồ Boxplot 30](#_Toc185852545)

[Hình 5.7: Kết quả phân loại 31](#_Toc185852546)

[Hình 5.8: Cây quyết định cho bài toán phân loại 31](#_Toc185852547)

[Hình 5.9: Kết quả hồi quy 32](#_Toc185852548)

[Hình 5.10: Biểu đồ dự đoán với thực tế 32](#_Toc185852549)

[Hình 5.11: Cây quyết định cho bài toán hồi quy 32](#_Toc185852550)

[Hình 5.12: So sánh hiệu suất CART với mô hình khác cho bài toán phân loại 33](#_Toc185852551)

[Hình 5.13: So sánh hiệu suất CART với mô hình khác cho bài toán hồi quy 33](#_Toc185852552)

# DANH MỤC BẢNG BIỂU

[Bảng 2.1: So sánh với các thuật toán khác 10](#_Toc185852616)

[Bảng 6.1: Phân công công việc 36](#_Toc185852617)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| CART | Classification and Regression Trees |
| MAE | Mean Absolute Error |
| MSE | Mean Squared Error |
| SVM | Support Vector Machine |
| SVC | Support Vector Classification |
| SVR | Support Vector Regression |
| RMSE | Root Mean Squared Error |
| MAE | Mean Absolute Error |
|  |  |

# TỔNG QUAN ĐỀ TÀI

## Lý do chọn đề tài

Trong bối cảnh phát triển mạnh mẽ của trí tuệ nhân tạo và học máy, các thuật toán cây quyết định (Decision Tree) như CART (Classification and Regression Tree) đang ngày càng được ứng dụng rộng rãi nhờ tính đơn giản, hiệu quả và khả năng diễn giải cao. CART không chỉ là nền tảng cho nhiều mô hình hiện đại như Random Forest và Gradient Boosting, mà còn đặc biệt hữu ích trong việc xử lý các bài toán phân loại và hồi quy.

Tuy nhiên, so với các thuật toán phổ biến khác như Logistic Regression, SVM, hoặc Neural Networks, thuật toán CART thường bị đánh giá thấp hơn về mặt hiệu suất trong một số tình huống cụ thể. Việc nghiên cứu chi tiết thuật toán này sẽ giúp hiểu rõ hơn cách thức hoạt động, ưu điểm, nhược điểm và cải thiện khả năng áp dụng của nó trong các bài toán thực tế.

Bên cạnh đó, so sánh CART với các thuật toán khác cho phép xác định bài toán mà CART thực sự vượt trội, nhờ khả năng giải thích kết quả dễ dàng. Điều này đặc biệt hữu ích khi làm việc với dữ liệu dạng bảng hoặc các bài toán yêu cầu sự minh bạch trong dự đoán.

## Mục tiêu thực hiện

* Tìm hiểu sâu về thuật toán CART: Nghiên cứu chi tiết lý thuyết nền tảng, cách hoạt động, các tham số và quy trình xây dựng mô hình CART.
* Ứng dụng thực tiễn: Triển khai thuật toán CART vào các bài toán cụ thể trong phân loại (Classification) và hồi quy (Regression).
* So sánh hiệu suất: Đánh giá và so sánh CART với các thuật toán khác như Logistic Regression, SVM, và Linear Regression để xác định điểm mạnh và hạn chế.
* Đề xuất cải tiến: Nghiên cứu các phương pháp tối ưu hóa thuật toán CART để nâng cao hiệu suất trong các bài toán phức tạp.

## ****Phạm vi nghiên cứu****

* **Thuật toán chính**: CART được sử dụng làm trọng tâm nghiên cứu.
* **Thuật toán so sánh**: Logistic Regression, SVM, Linear Regression và các thuật toán phổ biến khác sẽ được sử dụng để đối chiếu hiệu suất.
* **Dữ liệu**: Sử dụng các bộ dữ liệu mẫu từ thực tế hoặc dữ liệu giả lập liên quan đến các bài toán phân loại và hồi quy.
* **Công cụ**: Python, thư viện sklearn, và các công cụ trực quan hóa dữ liệu (matplotlib, seaborn).

## ****Phương pháp thực hiện****

* **Nghiên cứu tài liệu**:
  + Tìm hiểu lý thuyết về thuật toán CART thông qua các tài liệu học thuật, sách và bài báo khoa học.
  + Tìm kiếm các ứng dụng thực tế của CART trong các lĩnh vực khác nhau.
* **Triển khai thực nghiệm**:
  + Xây dựng mô hình CART trên các bộ dữ liệu mẫu.
  + Điều chỉnh các tham số để tối ưu hóa mô hình.
* **So sánh hiệu suất**:
* Thực hiện so sánh với các thuật toán khác dựa trên các chỉ số đánh giá như độ chính xác, MSE, R-squared.
* Trực quan hóa kết quả để làm rõ sự khác biệt.
* **Đánh giá và kết luận**: Đưa ra nhận xét về tính hiệu quả và các trường hợp nên sử dụng thuật toán CART.

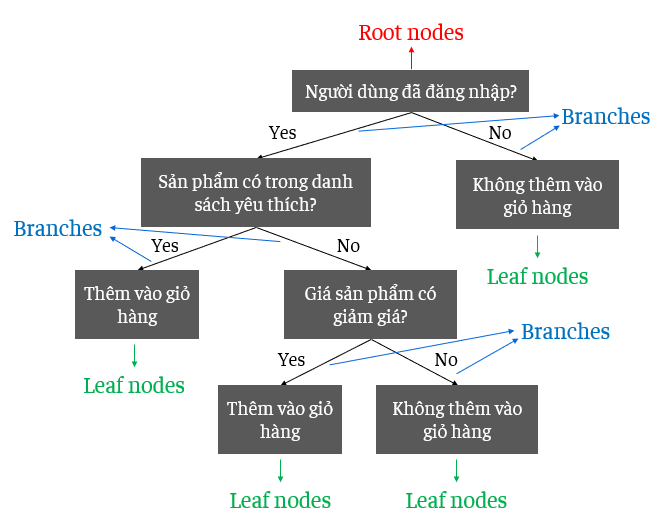
# cơ sở lý thuyết

## Định nghĩa thuật toán

CART (Classification and Regression Trees) là một thuật toán cây quyết định được sử dụng cho cả nhiệm vụ phân loại (Classification) và hồi quy (Regression). Đó là một thuật toán học có giám sát, học từ dữ liệu được gắn nhãn để dự đoán dữ liệu chưa nhìn thấy.

CART xây dựng cấu trúc dạng cây bao gồm 4 thành phần:

* Nút gốc (Root Node): Là điểm khởi đầu, chứa toàn bộ dữ liệu và điều kiện quyết định đầu tiên.
* Nút trung gian (Internal Nodes): Đại diện cho các điểm quyết định trong cây, phân chia dữ liệu dựa trên một tiêu chí.
* Nút lá (Leaf Nodes): Đại diện cho kết quả cuối cùng, chứa nhãn lớp (đối với phân loại) hoặc giá trị dự đoán (đối với hồi quy).
* Nhánh (Branches): Đại diện cho các kết quả của mỗi điều kiện phân chia.



Hình .: Ví dụ về cấu trúc cây

Cắt tỉa (Pruning) là một kỹ thuật quan trọng trong thuật toán CART, giúp giảm thiểu hiện tượng overfitting bằng cách loại bỏ những phần của cây quyết định không đóng góp đáng kể vào hiệu suất mô hình. Phương pháp này mang lại lợi ích chính như giảm độ phức tạp của cây, giúp mô hình trở nên dễ hiểu hơn, đồng thời cải thiện khả năng tổng quát hóa khi áp dụng trên dữ liệu mới. Hai phương pháp cắt tỉa phổ biến nhất là:

* Cắt tỉa độ phức tạp chi phí (Cost Complexity Pruning): Tính toán chi phí của từng nút và loại bỏ các nút có chi phí âm, tức những nút không mang lại giá trị cho mô hình.
* Cắt tỉa độ lợi thông tin (Information Gain Pruning): Đánh giá độ lợi thông tin của mỗi nút và loại bỏ những nút có độ lợi thông tin thấp, vì chúng không đóng góp nhiều vào việc cải thiện độ chính xác của mô hình.

Nhờ những kỹ thuật này, mô hình CART có thể đạt được sự cân bằng giữa độ chính xác và độ phức tạp, đảm bảo hiệu suất tốt hơn trên dữ liệu chưa từng thấy.

## Cách thức hoạt động

CART hoạt động bằng cách chia dữ liệu thành các nhánh (branches) dựa trên giá trị của các thuộc tính đầu vào. Quá trình này được gọi là “chia để trị” (divide and conquer) và thực hiện qua các bước sau:

Bước 1: Xác định thuộc tính tốt nhất để phân chia (Splitting Criterion)

CART chọn thuộc tính phân chia dựa trên các tiêu chí cụ thể, nhằm tối ưu hóa việc phân chia dữ liệu tại mỗi nút:

* Gini Index (Classification Trees): Đo lường mức độ không thuần nhất (Impurity) của dữ liệu, ưu tiên giảm sự pha trộn của các lớp.
* Variance Reduction (Regression Trees): Đo lường mức độ giảm phương sai (Variance) của dữ liệu sau khi phân chia, giúp tối ưu hóa dự đoán.

Bước 2: Xây dựng cây quyết định:

* Thuật toán bắt đầu từ nút gốc (root), xác định thuộc tính tốt nhất để chia dữ liệu thành hai nhánh con (binary split).
* Quá trình chia tiếp tục lặp lại ở các nút con cho đến khi đạt một trong các điều kiện dừng:
* Độ sâu của cây đạt giới hạn đã đặt.
* Số lượng mẫu trong nút con nhỏ hơn ngưỡng tối thiểu.
* Không còn cải thiện đáng kể trong chất lượng phân chia.

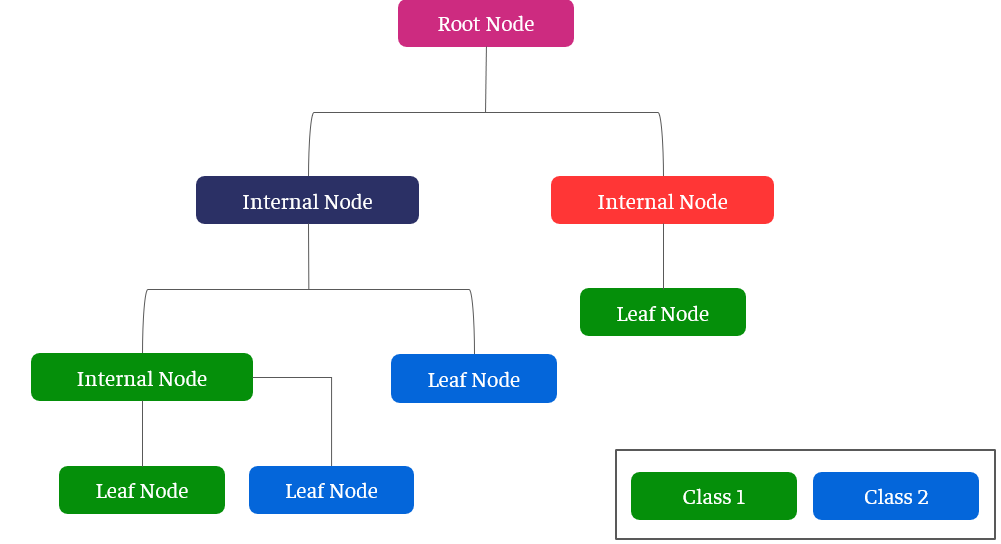
Bước 3: Cắt tỉa cây

Sau khi xây dựng cây hoàn chỉnh, quá trình cắt tỉa được thực hiện để giảm thiểu hiện tượng overfitting. CART thường áp dụng phương pháp **Cost-Complexity Pruning**, loại bỏ các nhánh không cần thiết để tìm ra kích thước cây tối ưu, cân bằng giữa độ chính xác và độ phức tạp.

Bước 4: Dự đoán:

* Đối với Classification Trees, mỗi nút lá chứa nhãn lớp phổ biến nhất trong dữ liệu thuộc nút đó.
* Đối với Regression Trees, mỗi nút lá chứa giá trị trung bình của các mẫu trong nút, đại diện cho giá trị dự đoán.

Quy trình này đảm bảo rằng cây quyết định được tối ưu hóa để phân loại hoặc dự đoán hiệu quả, đồng thời giảm thiểu các rủi ro từ dữ liệu nhiễu hoặc phân chia



Hình .: Một ví dụ về cây quyết định

## Tiêu chí phân chia

### Chỉ số Gini

Chỉ số Gini (Gini Index) là một tiêu chí quan trọng được CART sử dụng để đánh giá mức độ không thuần nhất của dữ liệu tại một nút.

Chỉ số Gini của một tập dữ liệu D được tính như sau:

Trong đó:

* : Tỉ lệ của mẫu thuộc lớp i.
* Tổng số lớp.

Đặc điểm của chỉ số Gini:

* : Tập dữ liệu hoàn toàn thuần nhất (Chỉ chứa một lớp).
* càng lớn: Tập dữ liệu càng không thuần nhất (Chứa nhiều lớp khác nhau).

### Giảm phương sai

Đối với các bài toán hồi quy, CART sử dụng phương sai (Variance) để đánh giá chất lượng phân chia.

Phương sai của một tập dữ liệu D được tính như sau:

Trong đó:

* : Giá trị đầu ra của mẫu i.
* : Giá trị trung bình của tất cả các mẫu trong D.
* : Số lượng mẫu trong D.

Giảm phương sai (Variance Reduction) được tính bằng cách so sánh phương sai trước và sau khi phân chia. Phân chia tốt là khi phương sai của các nhánh con giảm đáng kể so với nút cha.

### Cách chọn tiêu chí

#### Chỉ số Gini

Gini có công thức đơn giản, dễ tính toán. Điều này giúp tiết kiệm thời gian khi xây dựng cây quyết định, đặc biệt khi xử lý lượng dữ liệu lớn. CART được thiết kế để hoạt động nhanh chóng và hiệu quả nên việc chọn Gini là tối ưu cho tốc độ xử lý.

Gini đo lường mức độ “không thuần khiết” (Impurity) của một tập dữ liệu. Khi giá trị Gini càng nhỏ, dữ liệu càng thuần khiết, tức là tập dữ liệu có xu hướng thuộc về một nhãn cụ thể. Điều này giúp CART thực hiện các phân chia sao cho nhánh con có mức độ thuần khiết cao hơn so với nhánh cha.

Gini nhạy với sự mất cân bằng lớp trong dữ liệu. Điều này giúp tạo ra các phân nhánh cân đối hơn ngay cả khi số lượng mẫu của các lớp không đồng đều.

So sánh với Entropy, Entropy yêu cầu tính toán logarit, làm tăng độ phức tạp và thời gian xử lý. Trong khi đó, Gini và Entropy thường cho kết quả tương tự nhau. Vì vậy, dùng Gini trong CART vừa đơn giản hơn vừa đủ hiệu quả.

#### Giảm phương sai:

Phương sai đo lường độ phân tán của các giá trị trong một tập dữ liệu. Phân chia tốt trong cây hồi quy là khi phương sai của các nhánh con giảm đáng kể so với nút cha. Điều này giúp mô hình dự đoán giá trị gần với giá trị thực hơn.

Công thức tính phương sai chỉ bao gồm phép trừ và bình phương, giúp việc tính toán diễn ra nhanh chóng. Điều này phù hợp với mục tiêu của CART là xây dựng cây quyết định nhanh và hiệu quả.

Phương sai là một chỉ số quen thuộc trong thống kê, dễ hiểu và dễ áp dụng. Việc sử dụng phương sai giúp người dùng dễ đánh giá được mức độ hiệu quả quá trình phân chia.

Một số tiêu chí hồi quy khác như Mean Absolute Error (MAE) hay Mean Squared Error (MSE) có thể được sử dụng. Tuy nhiên, MAE không khả vi liên tục, làm cho việc tối ưu hóa khó khăn hơn. MSE thực chất cũng liên quan đến phương sai nên việc dùng phương sai là một cách đơn giản hóa MSE mà vẫn đảm bảo hiệu quả.

## Ứng dụng của thuật toán

### CART cho phân loại

CART cho phân loại hoạt động bằng cách đệ quy chia nhỏ training data thành các tập con nhỏ hơn và nhỏ hơn nữa dựa trên một số tiêu chí nhất định. Mục tiêu là chia dữ liệu sao cho độ không thuần nhất (Impurity) trong mỗi tập con được giảm thiểu. Độ không thuần nhất là một chỉ số đo mức độ pha trộn của dữ liệu trong một tập con cụ thể. Đối với các bài toán phân loại, CART sử dụng Gini impurity để đo độ không thuần nhất.

Bài toán phân loại được dùng khi mục tiêu là phân loại dữ liệu vào các nhóm hoặc nhãn cụ thể. Các bài toán phân loại thường yêu cầu trả lời các câu hỏi dạng “Yes/No”, “True/False” hoặc xác định lớp mà dữ liệu thuộc về.

Ví dụ:

* Phân loại email: Xác định xem một email có phải là spam hay không.
* Phát hiện gian lận: Dự đoán liệu một giao dịch có phải là gian lận hay không.

### CART cho hồi quy

CART cho hồi quy cũng hoạt động bằng cách chia nhỏ dữ liệu huấn luyện thành các tập con, nhưng mục tiêu là giảm thiểu sai số dư (Residual) trong mỗi tập con. Điều này giúp cải thiện khả năng dự đoán giá trị liên tục cho các bài toán hồi quy.

Bài toán hồi quy được dùng khi khi mục tiêu là dự đoán một giá trị liên tục. Các bài toán hồi quy giúp ước tính các giá trị số dựa trên các yếu tố đầu vào khác nhau.

Ví dụ:

* Dự đoán giá nhà: Dự đoán giá trị của một ngôi nhà dựa trên các đặc điểm như diện tích, vị trí, và số phòng.
* Dự báo doanh thu: Ước tính doanh thu tương lai dựa trên các yếu tố như xu hướng thị trường, tăng trưởng khách hàng.

## Ưu điểm và nhược điểm

### Ưu điểm

Dễ hiểu và dễ triển khai: Cây quyết định từ CART có dạng trực quan, dễ đọc và giải thích. Người không có kiến thức chuyên sâu về kỹ thuật cũng có thể hiểu được mô hình.

Xử lý được cả phân loại và hồi quy: CART là một thuật toán linh hoạt, có thể áp dụng cho cả bài toán phân loại và bài toán hồi quy.

Không yêu cầu chuẩn hóa dữ liệu: không cần đưa về cùng một khoảng giá trị như trong các thuật toán khác.

Xử lý được dữ liệu thiếu và nhiễu: CART có khả năng làm việc tốt với dữ liệu bị thiếu hoặc chứa nhiễu.

Tính chất phi tuyến tính: Có thể phát hiện và mô hình hóa các mối quan hệ phi tuyến tính giữa các biến mà các mô hình hồi quy tuyến tính không thể làm được.

Tự động chọn đặc trưng quan trọng: CART tự động chọn ra các thuộc tính quan trọng nhất để phân chia dữ liệu mà không cần thêm bước xử lý đặc biệt.

### Nhược điểm

Dễ bị Overfitting (Quá khớp): Cây quyết định có thể học quá kỹ từ dữ liệu huấn luyện, dẫn đến kết quả kém trên dữ liệu mới. Cần áp dụng kỹ thuật cắt tỉa (Pruning) để giảm overfitting.

Nhạy cảm với thay đổi nhỏ trong dữ liệu: Chỉ một thay đổi nhỏ trong dữ liệu có thể làm thay đổi cấu trúc cây một cách đáng kể.

Kém hiệu quả với dữ liệu lớn: Khi làm việc với tập dữ liệu lớn hoặc nhiều thuộc tính, cây có thể trở nên rất lớn và phức tạp, làm giảm hiệu quả tính toán.

Không phù hợp cho dữ liệu có quá nhiều nhãn: Trong các bài toán phân loại có quá nhiều lớp, cây quyết định có thể trở nên cồng kềnh và khó diễn giải.

Hiệu suất thấp hơn so với các mô hình ensemble: Các thuật toán ensemble như Random Forest và Gradient Boosting thường cho kết quả tốt hơn nhờ kết hợp nhiều cây quyết định lại với nhau.

Phân chia kém khi có dữ liệu không cân bằng: Với dữ liệu phân loại không cân bằng, CART có thể bị thiên lệch về lớp có số lượng mẫu lớn hơn.

## So sánh với các thuật toán khác

Bảng .: So sánh với các thuật toán khác

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuật toán** | **CART** | **Logistic Regression** | **SVM** |
| **Loại bài toán** | Phân loại và hồi quy | Chỉ phân loại | Phân loại (SVC) và hồi quy (SVR) |
| **Nguyên lý hoạt động** | Xây dựng một cây quyết định duy nhất | Tìm hàm hồi quy tuyến tính | Tìm siêu phẳng tối ưu |
| **Tiêu chí chia tách** | Gini (Phân loại)  Giảm phương dư (Hồi quy) | Không có tiêu chí chia tách | Khoảng cách và biên tối ưu |
| **Ưu điểm** | Dễ hiểu và trực quan Linh hoạt | Đơn giản và hiệu quả với dữ liệu nhỏ | Mạnh với dữ liệu phức tạp và không tuyến tính |
| **Nhược điểm** | Dễ bị overfitting  Nhạy cảm với nhiễu | Chỉ áp dụng được cho phân loại tuyến tính | Tốn nhiều tài nguyên tính toán |
| **Khả năng xử lý nhiễu** | Trung bình | Thấp | Cao |
| **Khả năng tránh overfitting** | Thấp (Nếu không cắt tỉa) | Trung bình | Cao (Nếu chọn kernel tốt) |
| **Tốc độ học** | Nhanh | Nhanh | Chậm |
| **Ứng dụng phổ biến** | Phân loại khách hàng Dự báo giá nhà | Phân loại nhị phân Dự đoán xác suất | Nhận dạng chữ viết tay Phân loại hình ảnh |

# bài toán dự đoán lượng mưa

## Đặt vấn đề

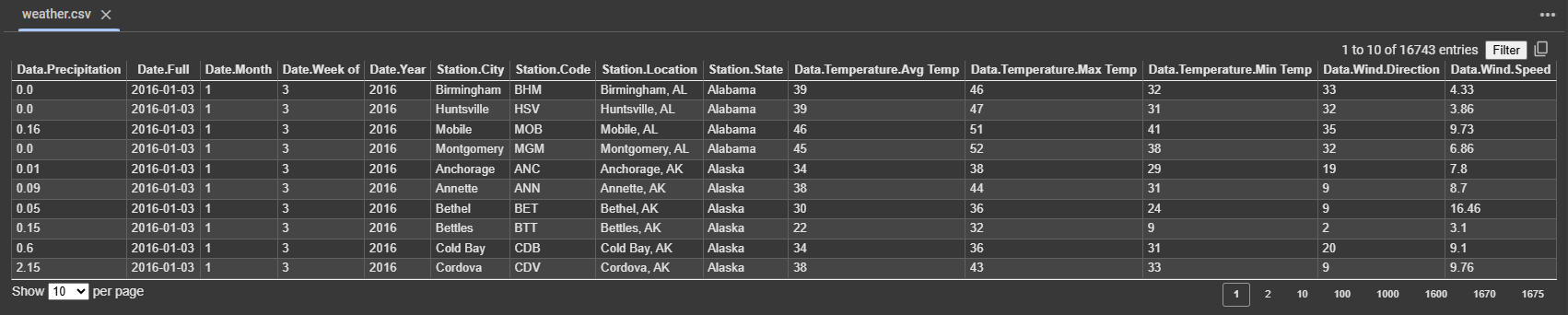
Trong lĩnh vực khí tượng, dự đoán lượng mưa là một vấn đề quan trọng để đưa ra các cảnh báo sớm, hỗ trợ nông nghiệp và phòng chống thiên tai. Lượng mưa là yếu tố quan trọng trong việc xác định các biện pháp phòng ngừa, quản lý tài nguyên nước, và dự báo thời tiết. Để đạt được điều này, nhóm 05 sẽ xây dựng một mô hình dự đoán lượng mưa dựa trên các yếu tố thời tiết như nhiệt độ, độ ẩm, hướng gió và tốc độ gió.

Mục tiêu của bài toán là phát triển một mô hình dự đoán có thể cung cấp kết quả chính xác về lượng mưa từ các dữ liệu quan trắc, giúp cải thiện công tác dự báo và hỗ trợ ra quyết định trong thực tế, phục vụ cho công tác phòng chống thiên tai và bảo vệ sản xuất nông nghiệp.

## Mô tả bài toán

Bài toán yêu cầu dự đoán lượng mưa từ các dữ liệu về thời tiết và các yếu tố khí tượng khác. Các yếu tố này bao gồm nhiệt độ, độ ẩm, hướng gió và tốc độ gió, cũng như các thông tin thời gian và vị trí trạm đo. Mô hình cần phải có khả năng học từ dữ liệu huấn luyện và dự đoán lượng mưa chính xác cho các thời điểm và khu vực khác nhau.

Tập dữ liệu mà bài toán sử dụng được lấy từ các bộ dữ liệu khí tượng từ [Corgis Weather Dataset](https://corgis-edu.github.io/corgis/csv/weather/), nguồn gốc của bộ dữ liệu này lấy thông tin từ các báo cáo WFO này cho các thành phố trên toàn quốc và tóm tắt nó ở cấp độ hàng tuần cho cả năm 2016.



Hình .: Bộ dữ liệu Corgis Weather Dataset

Dữ liệu được cung cấp bao gồm các bảng với các đặc trưng sau:

* Thông tin thời tiết:
* Data.Precipitation: Lượng mưa trong một khoảng thời gian nhất định, đơn vị là milimét. Đây là biến mục tiêu trong bài toán mô hình sẽ dự đoán giá trị này trong bài toán hồi quy và phân loại (dự đoán có mưa hay không dựa trên ngưỡng lượng mưa).
* Data.Temperature.Avg Temp: Nhiệt độ trung bình trong ngày (Đơn vị: độ C).
* Data.Temperature.Max Temp: Nhiệt độ tối đa trong ngày
* Data.Temperature.Min Temp: Nhiệt độ tối thiểu trong ngày.
* Data.Humidity: Độ ẩm không khí trong ngày, thể hiện mức độ bão hòa của không khí với hơi nước.
* Data.Wind.Speed: Tốc độ gió (Đơn vị: km/h).
* Data.Wind.Direction: Hướng gió, thể hiện góc độ của hướng gió (Đơn vị: độ).
* Data.Pressure: Áp suất khí quyển (Đơn vị: hPa).
* Thông tin về thời gian:
* Date: Ngày của mỗi lần quan trắc, có thể dùng để phân loại dữ liệu theo mùa hoặc thời gian trong năm.
* Month: Tháng trong năm của dữ liệu quan trắc.
* Week of Year: Tuần trong năm.
* Year: Năm của dữ liệu quan trắc.
* Thông tin trạm đo:
* Station Location: Vị trí địa lý của trạm đo.
* Station Name: Tên trạm đo.

## Phương pháp áp dụng

Để giải quyết bài toán dự đoán lượng mưa dựa trên các yếu tố thời tiết trong bộ dữ liệu trên, chúng ta có thể sử dụng thuật toán CART để xây dựng mô hình phân loại và hồi quy. CART giúp dự đoán lượng mưa theo dạng giá trị liên tục hoặc phân loại mưa (có mưa/không mưa) dựa trên ngưỡng lượng mưa.

Các bước thực hiện:

* Bước 1: Tiền xử lý dữ liệu
* Làm sạch dữ liệu: Xử lý các giá trị thiếu hoặc không hợp lệ trong tập dữ liệu. Các giá trị thiếu có thể được điền bằng các phương pháp như trung bình, hoặc có thể loại bỏ các bản ghi không đầy đủ.
* Chuẩn hóa dữ liệu: Đảm bảo rằng tất cả các đặc trưng có cùng thang đo hoặc được chuyển đổi về phạm vi giá trị tương tự nhau (đặc biệt với tốc độ gió, nhiệt độ, và áp suất).
* Lựa chọn đặc trưng: Xác định các yếu tố thời tiết có tác động lớn đến lượng mưa, ví dụ như nhiệt độ, độ ẩm và tốc độ gió.
* Bước 2: Xây dựng mô hình CART
* Mô hình phân loại: Dự đoán có mưa hay không (Dựa trên ngưỡng lượng mưa, ví dụ, mưa khi lượng mưa vượt quá 1mm), sử dụng tiêu chí Gini để phân loại các ngày thành “Mưa” và “Không mưa”.
* Mô hình hồi quy: Dự đoán lượng mưa dưới dạng giá trị liên tục, sử dụng phương pháp giảm phương sai (Variance Reduction).
* Cây quyết định: Xây dựng cây quyết định trong đó mỗi nút đại diện cho một đặc trưng cụ thể (Ví dụ, nhiệt độ trung bình hoặc độ ẩm) và các nhánh phân chia dữ liệu theo các điều kiện.
* Bước 3: Đánh giá mô hình
* Phân loại: Đánh giá mô hình phân loại bằng các chỉ số như Accuracy, Precision, Recall, và F1-Score để đo lường khả năng phân loại chính xác các ngày có mưa và không mưa.
* Hồi quy: Đánh giá độ chính xác của mô hình bằng các chỉ số như RMSE và MAE để đo lường sai số dự đoán lượng mưa.
* Kiểm tra overfitting: Sử dụng kỹ thuật cắt tỉa cây (Pruning) để giảm bớt overfitting, tránh việc mô hình học quá kỹ các chi tiết trong dữ liệu huấn luyện mà không phản ánh đúng đặc trưng của dữ liệu mới.
* Bứớc 4: Dự đoán và triển khai
* Áp dụng mô hình đã xây dựng để dự đoán lượng mưa trên các dữ liệu kiểm thử hoặc dữ liệu mới.
* Cung cấp dự đoán cho các bài toán thực tế như dự báo thời tiết hàng ngày, hỗ trợ quyết định trong nông nghiệp và các hoạt động phòng chống thiên tai.

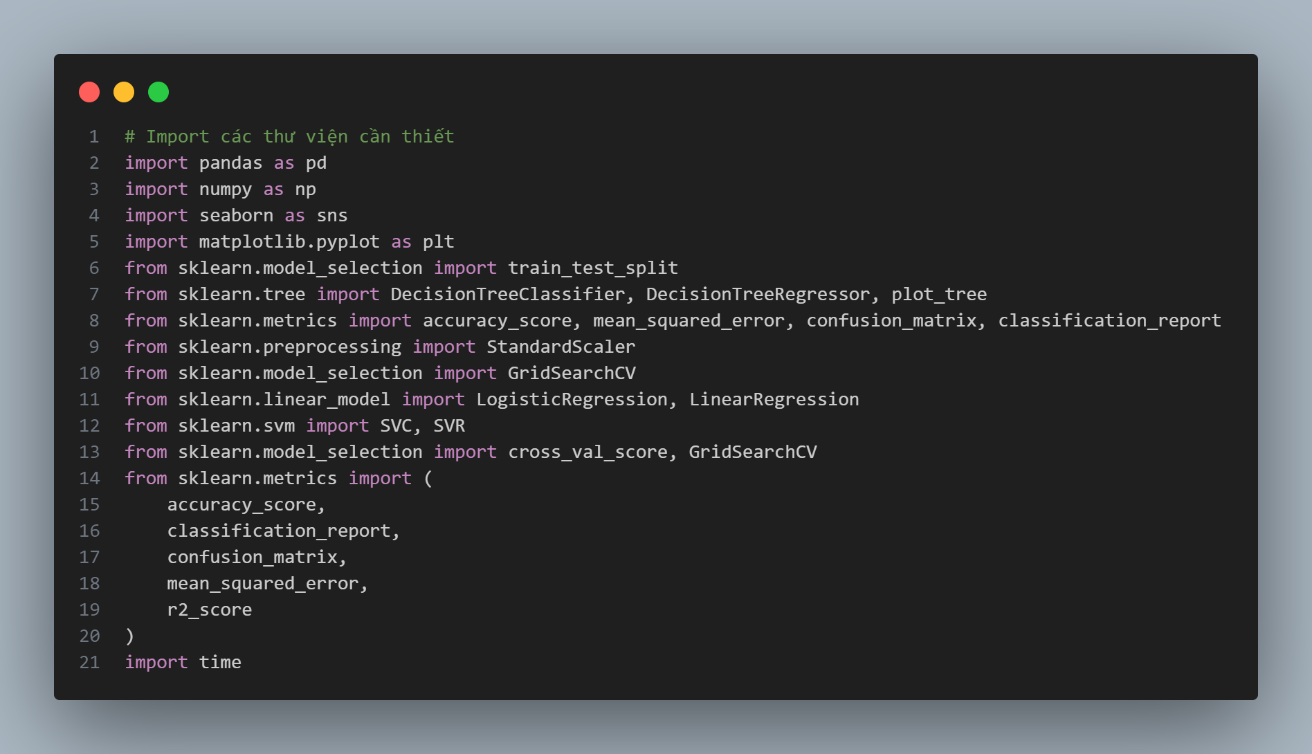
CART là một lựa chọn hiệu quả cho bài toán dự đoán lượng mưa vì:

* CART tạo ra cây quyết định có cấu trúc rõ ràng, dễ dàng giải thích và trực quan hóa giúp làm rõ các điều kiện thời tiết nào ảnh hưởng mạnh nhất đến lượng mưa.
* CART ít bị ảnh hưởng bởi các điểm dữ liệu ngoại lệ hơn so với một số thuật toán hồi quy tuyến tính truyền thống.
* Không giống như hồi quy tuyến tính hay SVM, CART không yêu cầu chuẩn hóa hoặc chuyển đổi dữ liệu về cùng một khoảng giá trị.
* CART sẽ tự động xác định các biến quan trọng nhất trong quá trình xây dựng cây quyết định. Điều này giúp giảm bớt công việc chọn lọc đặc trưng thủ công.
* CART hoạt động tốt trên tập dữ liệu lớn và có thể được mở rộng dễ dàng khi thêm các dữ liệu mới.

# THỰC NGHIỆM

## Chuẩn bị dữ liệu

### Cài đặt thư viện

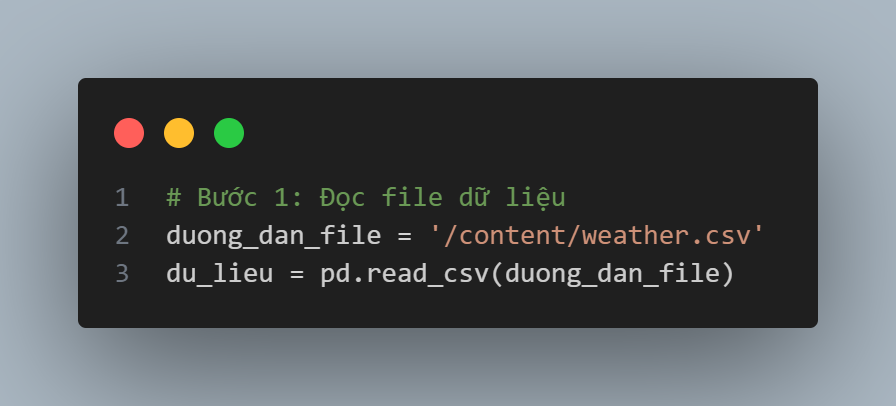


Hình .: Đoạn mã khai báo thư viện

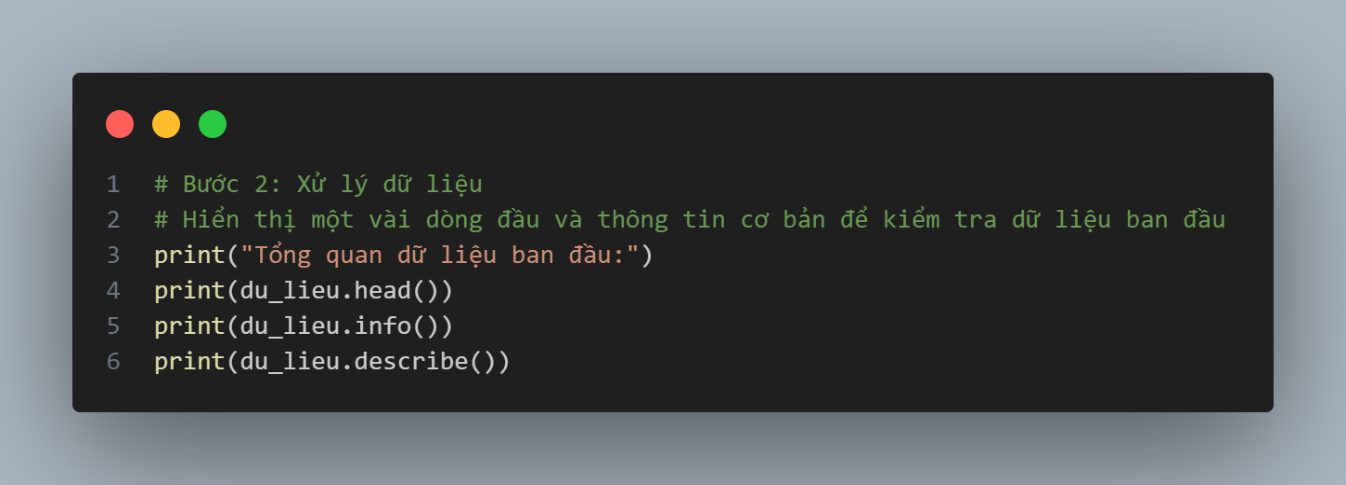
* pandas: Thư viện phổ biến để xử lý dữ liệu trong Python, dùng để thao tác với các bảng dữ liệu (DataFrame).
* numpy: Thư viện toán học hỗ trợ các phép toán số học, xử lý mảng, ma trận.
* seaborn và matplotlib.pyplot: Dùng để vẽ biểu đồ, trực quan hóa dữ liệu.
* sklearn.model\_selection: Chứa các công cụ để chia dữ liệu như train\_test\_split (Chia dữ liệu thành tập huấn luyện và kiểm tra), và GridSearchCV (Tìm kiếm siêu tham số tối ưu cho mô hình).
* sklearn.tree: Cung cấp các mô hình cây quyết định như DecisionTreeClassifier (Phân loại) và DecisionTreeRegressor (Hồi quy), cùng với hàm vẽ cây quyết định plot\_tree.
* sklearn.metrics: Cung cấp các công cụ để đánh giá mô hình như accuracy\_score (Độ chính xác), mean\_squared\_error (Sai số bình phương trung bình) và các ma trận như confusion\_matrix để đánh giá mô hình phân loại.
* sklearn.preprocessing: Cung cấp các phương thức chuẩn hóa dữ liệu như StandardScaler.
* LogisticRegression, LinearRegression, SVC, SVR: Các mô hình học máy khác để so sánh hiệu suất.

### Đọc và kiểm tra dữ liệu

Đọc dữ liệu từ file CSV có đường dẫn là /content/weather.csv và lưu vào biến du\_lieu. Tiếp theo thực hiện các thao tác kiểm tra dữ liệu để hiểu rõ hơn về dữ liệu như .head() để xem trước vài dòng dữ liệu đầu tiên, .info() để hiển thị thông tin chi tiết về cột (kiểu dữ liệu, số lượng giá trị không rỗng), .describe() để cung cấp thống kê cơ bản (mean, min, max, std) cho các cột số.

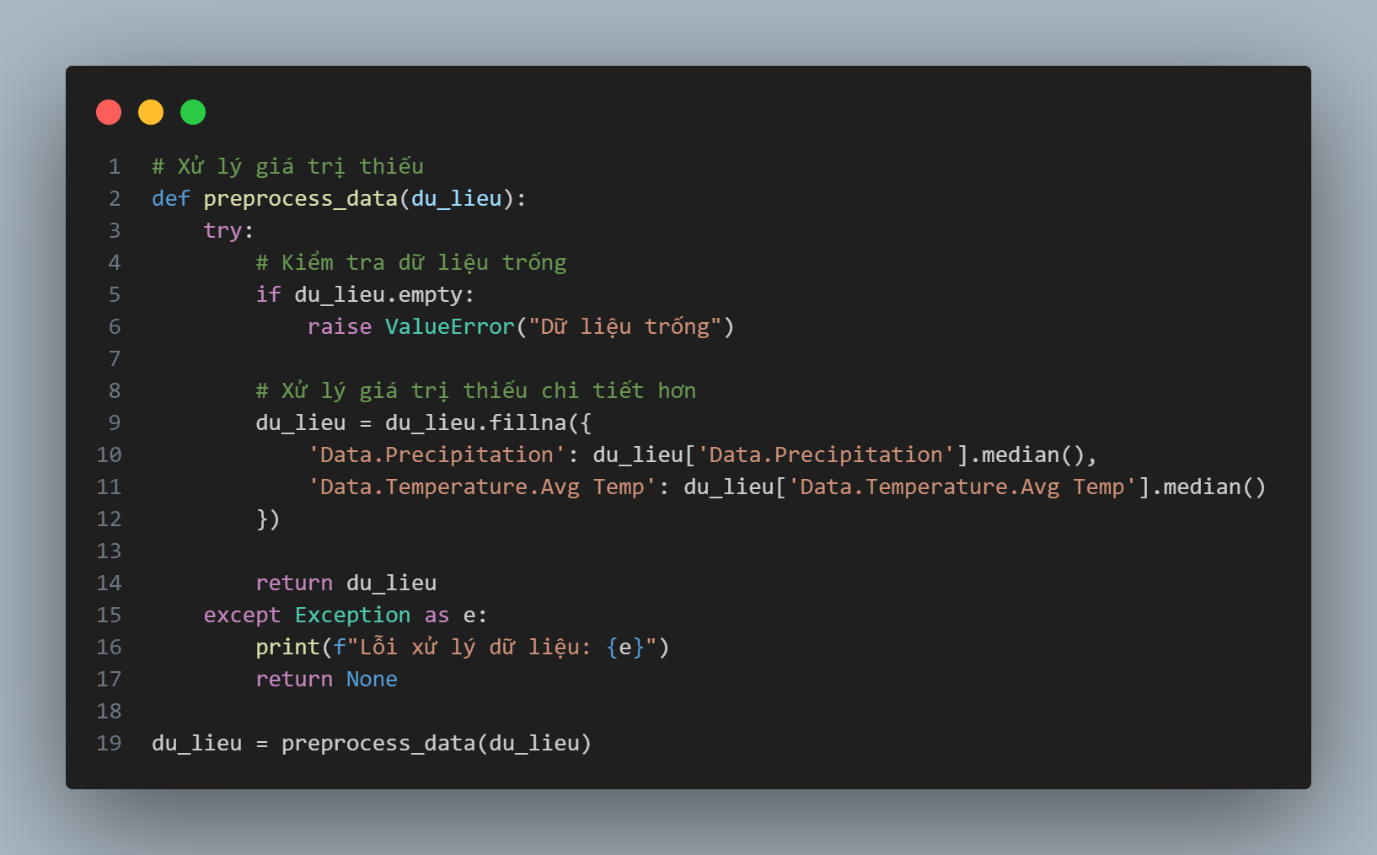


Hình .: Đoạn mã đọc dữ liệu



Hình .: Đoạn mã hiển thị dữ liệu

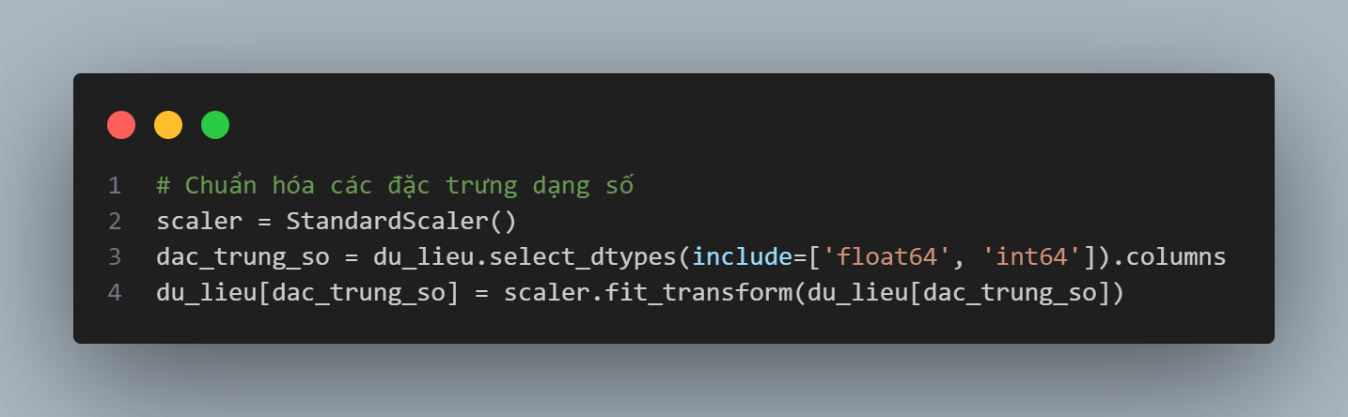
### Xử lý dữ liệu thiếu



Hình .: Đoạn mã dữ lý dữ liệu bị thiếu

* du\_lieu.empty: Xác định xem tập dữ liệu có bị rỗng hay không. Nếu dữ liệu rỗng, chương trình sẽ báo lỗi bằng cách ném ra ngoại lệ ValueError.
* du\_lieu.fillna(): Điền giá trị thiếu trong các cột cụ thể bằng trung vị (median) của cột đó.
* Lý do chọn trung vị (median):
  + Trung vị ít bị ảnh hưởng bởi các giá trị ngoại lệ (outliers).
  + Phù hợp khi dữ liệu phân phối không đồng đều.
* Xử lý ngoại lệ: Nếu xảy ra lỗi trong quá trình xử lý dữ liệu, thông báo lỗi sẽ được in ra màn hình.
* Hàm trả về None trong trường hợp lỗi.

### Chuẩn hoá đặc trưng dạng số



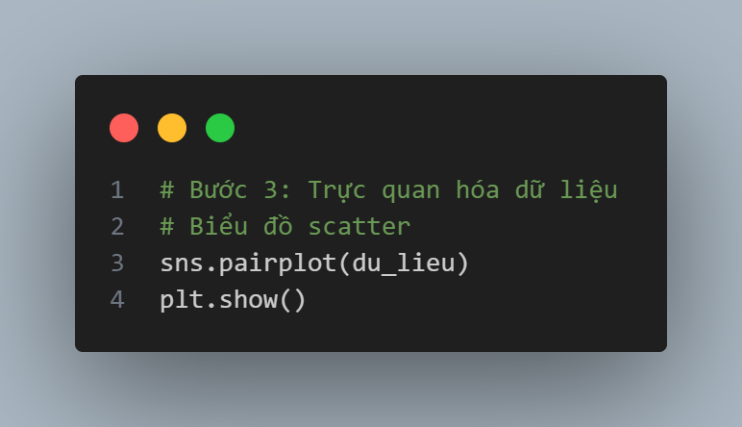
Hình .: Đoạn mã chuẩn hoá dạng số

* StandardScaler: Chuẩn hóa các cột số về phân phối có trung bình = 0 và độ lệch chuẩn = 1.
* Lấy danh sách các cột dạng số và chuẩn hóa chúng.

## Trực quan hoá dữ liệu

### Scatter Plot

Biểu đồ scatter được tạo bằng lệnh sns.pairplot(du\_lieu) để hiển thị mối quan hệ giữa tất cả các cặp đặc trưng (cột) trong dữ liệu. Công cụ Seaborn sẽ tự động bỏ qua các cột không phải dạng số hoặc chứa giá trị phân loại. Mục tiêu chính của biểu đồ là xác định các mối quan hệ tiềm năng giữa các đặc trưng và kiểm tra xem dữ liệu có phân tách rõ ràng không.

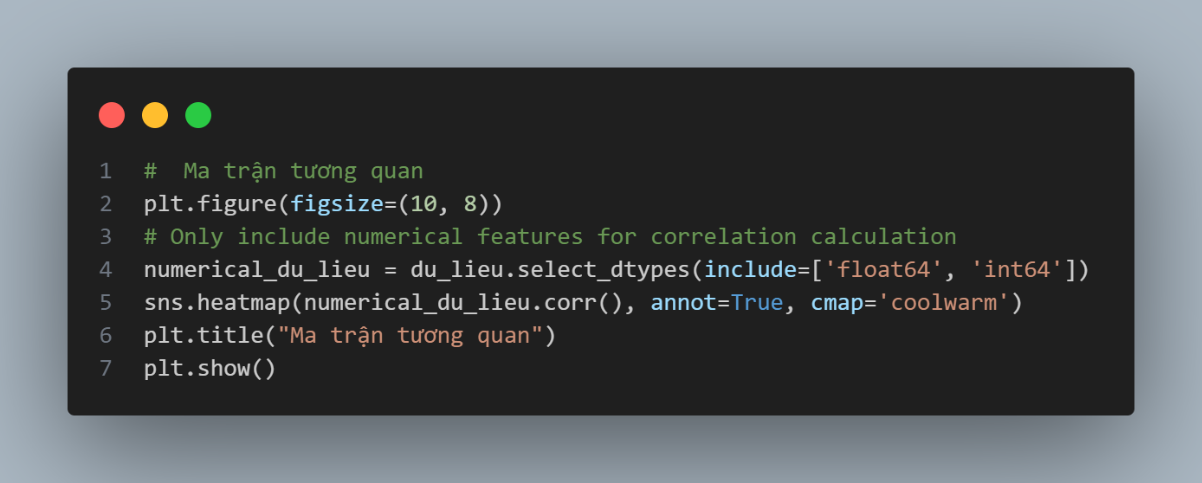


Hình .: Đoạn mã minh hoạ bằng Scatter Plot

### Ma trận tương quan

Dùng du\_lieu.select\_dtypes(include=['float64', 'int64']) để lọc các cột số, sau đó tính ma trận tương quan bằng numerical\_du\_lieu.corr() và hiển thị kết quả qua biểu đồ nhiệt (heatmap) bằng sns.heatmap(...).

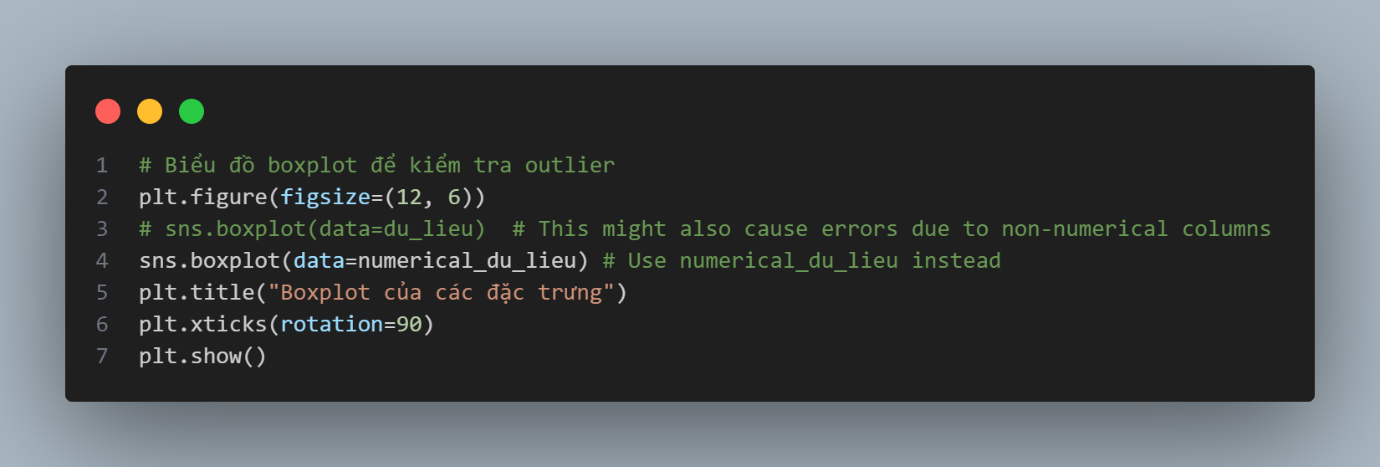
Ma trận tương quan giúp tìm mối quan hệ tuyến tính giữa các đặc trưng, qua đó có thể loại bỏ các cột có tương quan quá cao nếu cần giảm chiều dữ liệu.



Hình .: Đoạn mã minh hoạ bằng ma trận tương quan

### Boxplot

Biểu đồ boxplot được tạo bằng lệnh sns.boxplot(data=numerical\_du\_lieu) để hiển thị phân phối dữ liệu và phát hiện các giá trị ngoại lai (outliers) trong từng đặc trưng số. Các điểm nằm ngoài phần “whiskers” của biểu đồ thường được coi là giá trị ngoại lai.



Hình .: Đoạn mã minh hoạ bằng Boxplot

## Huấn luyện mô hình phân loại

### Tiền xử lý dữ liệu

Mục tiêu phân loại được định nghĩa là Data.Precipitation để dự đoán khả năng có mưa hay không dựa trên các đặc trưng khác trong dữ liệu. Nếu cột mục tiêu này không tồn tại trong dữ liệu, thông báo lỗi được in ra và không tiếp tục xử lý.

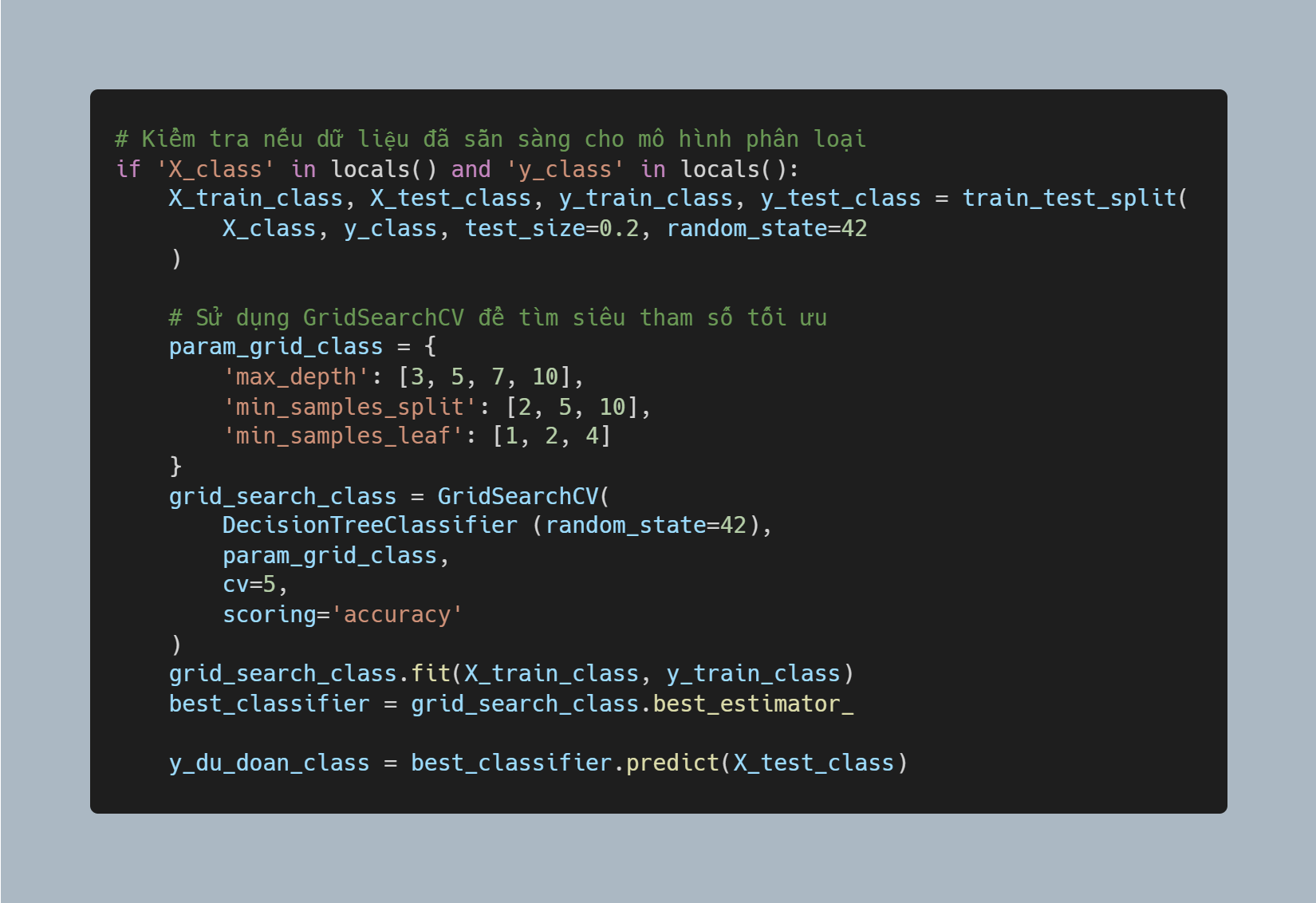
Dữ liệu đầu vào được xử lý bằng cách cột mục tiêu Data.Precipitation được chuyển thành giá trị nhị phân, với 1 là “mưa” và 0 là “không mưa” (Giả sử giá trị mưa > 0 là “mưa”, còn lại là “không mưa”). Các cột không phải số được loại bỏ khỏi tập dữ liệu. Các cột phân loại được mã hóa bằng pd.get\_dummies để chuyển chúng thành dạng số.



Hình .: Đoạn mã kiểm tra dữ liệu

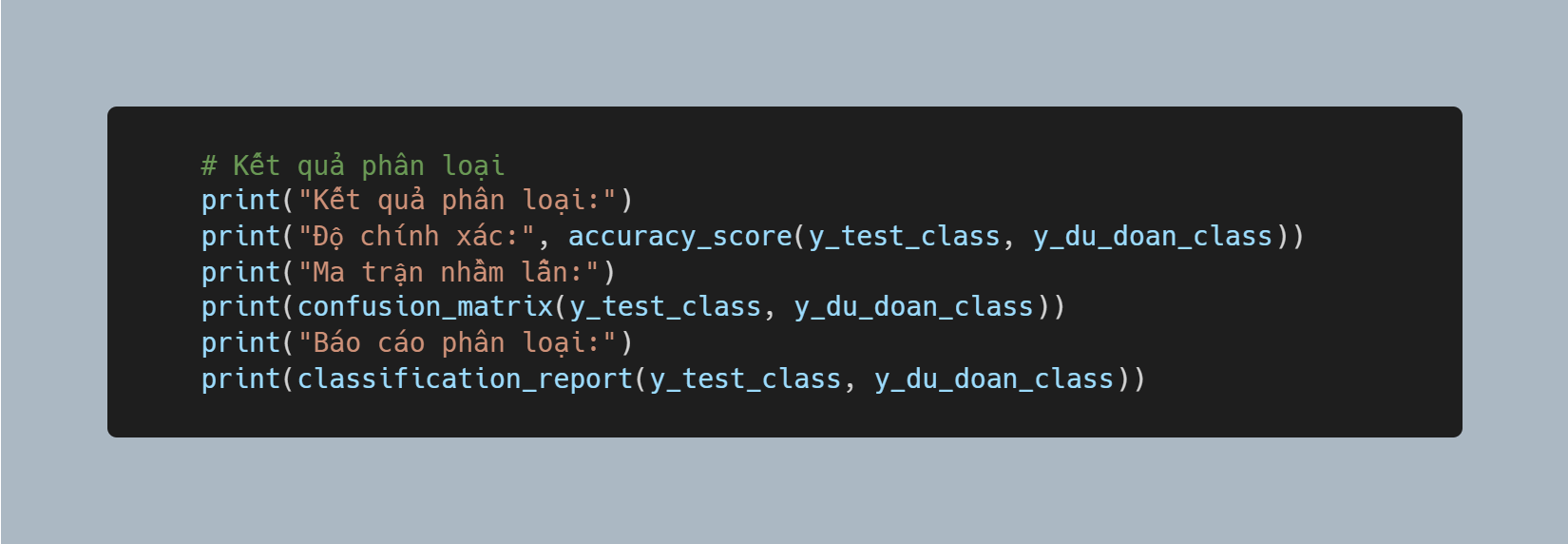
### ****Huấn luyện mô hình****

Dữ liệu sau khi tiền xử lý được chia thành tập huấn luyện (80%) và tập kiểm tra (20%). Để tối ưu hóa các siêu tham số của mô hình **DecisionTreeClassifier**, sử dụng GridSearchCV. Các siêu tham số được tối ưu hóa bao gồm: max\_depth, min\_samples\_split và min\_samples\_leaf. Sau khi tối ưu hóa, mô hình tốt nhất được đào tạo và sẵn sàng để đánh giá.



Hình .: Đoạn mã chia dữ liệu và huấn luyện mô hình

### Đánh giá mô hình

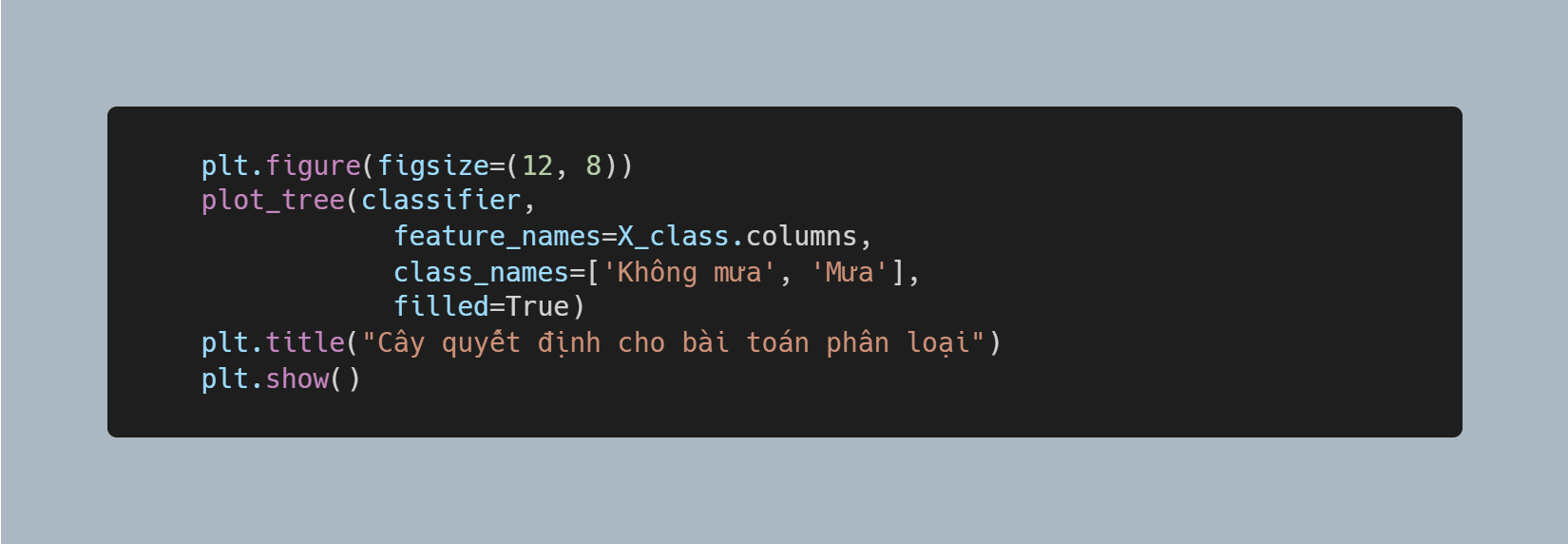


Hình .: Đoạn mã in kết quả phân loại

* Đánh giá độ chính xác của mô hình trên tập kiểm tra bằng accuracy\_score. Hàm này tính toán độ chính xác của mô hình phân loại. Độ chính xác là tỷ lệ phần trăm của các dự đoán đúng so với tổng số dự đoán.
* Ma trận nhầm lẫn được sử dụng bằng hàm confusion\_matrix để kiểm tra hiệu suất phân loại giữa các lớp “mưa” và “không mưa”. Ma trận này có 4 phần tử:
* True Positive (TP): Dự đoán đúng là “mưa”.
* False Positive (FP): Dự đoán sai là “mưa” nhưng thực tế là “không mưa”.
* True Negative (TN): Dự đoán đúng là “không mưa”.
* False Negative (FN): Dự đoán sai là “không mưa” nhưng thực tế là “mưa”.
* Báo cáo phân loại được cung cấp bằng cách dùng classification\_report, bao gồm các chỉ số:
* Precision: Tỷ lệ các dự đoán đúng đối với tổng số dự đoán cho một lớp.
* Recall: Tỷ lệ các mẫu thực tế của một lớp được mô hình nhận diện đúng.
* F1-score: Trung bình điều hòa giữa Precision và Recall, thường được sử dụng khi có sự mất cân đối giữa các lớp.
* Support: Số lượng mẫu thực tế của mỗi lớp.

### Trực quan hóa kết quả

Sử dụng mô hình tốt nhất từ GridSearchCV để dự đoán kết quả trên tập kiểm tra. Cây quyết định được trực quan hóa thông qua plot\_tree, giúp người dùng hiểu cách thức ra quyết định của mô hình dựa trên các đặc trưng đầu vào. Mô hình này có thể triển khai để dự đoán dữ liệu mới.



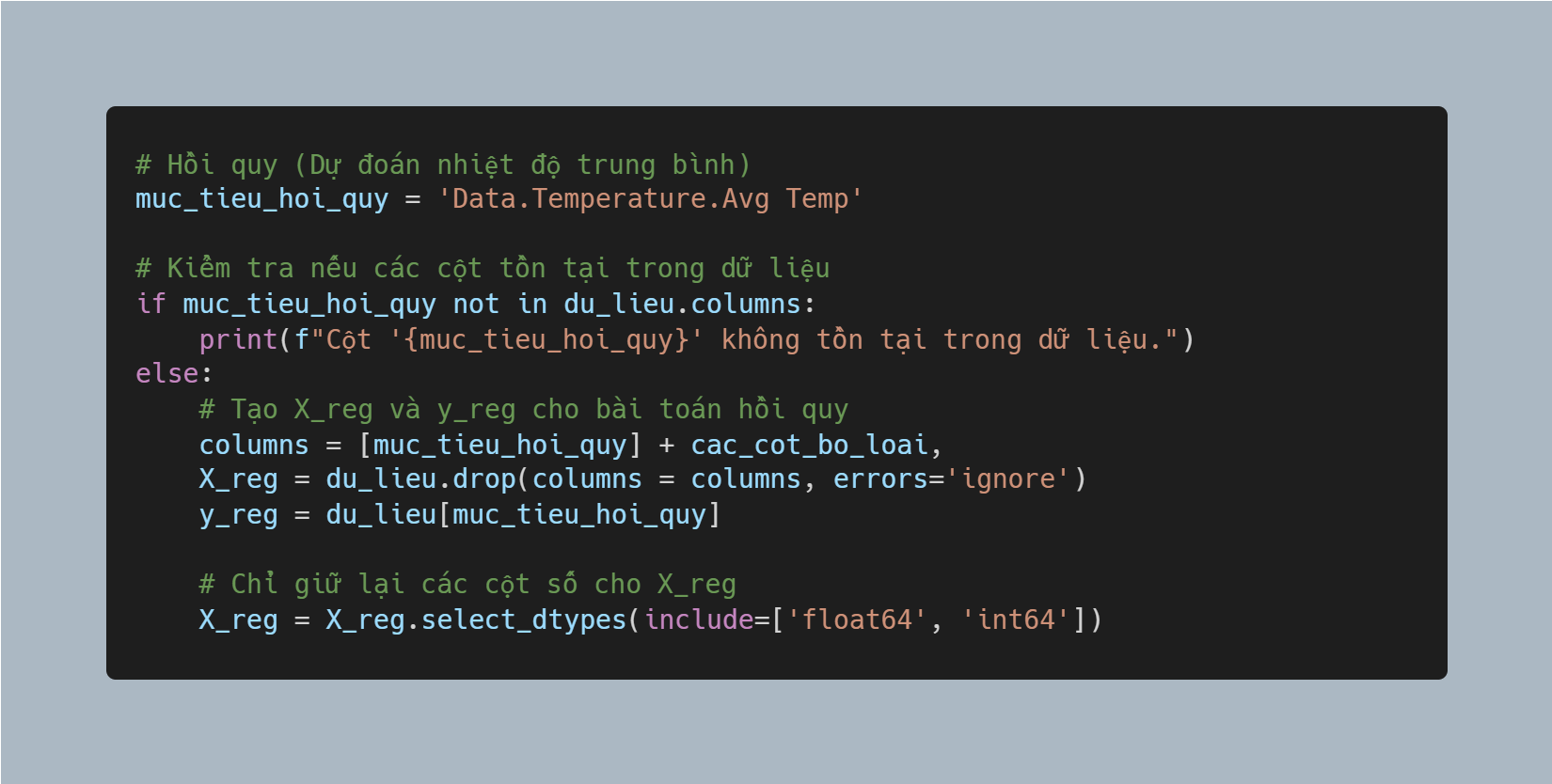
Hình .: Đoạn mã vẽ cây quyết định phân loại

## Huấn luyện mô hình hồi quy

### Tiền xử lý dữ liệu

Mục tiêu hồi quy được định nghĩa là Data.Temperature.Avg Temp, nhằm dự đoán nhiệt độ trung bình dựa trên các đặc trưng khác trong dữ liệu. Nếu cột mục tiêu này không tồn tại trong dữ liệu, một thông báo lỗi sẽ được in ra và quá trình xử lý sẽ dừng lại.

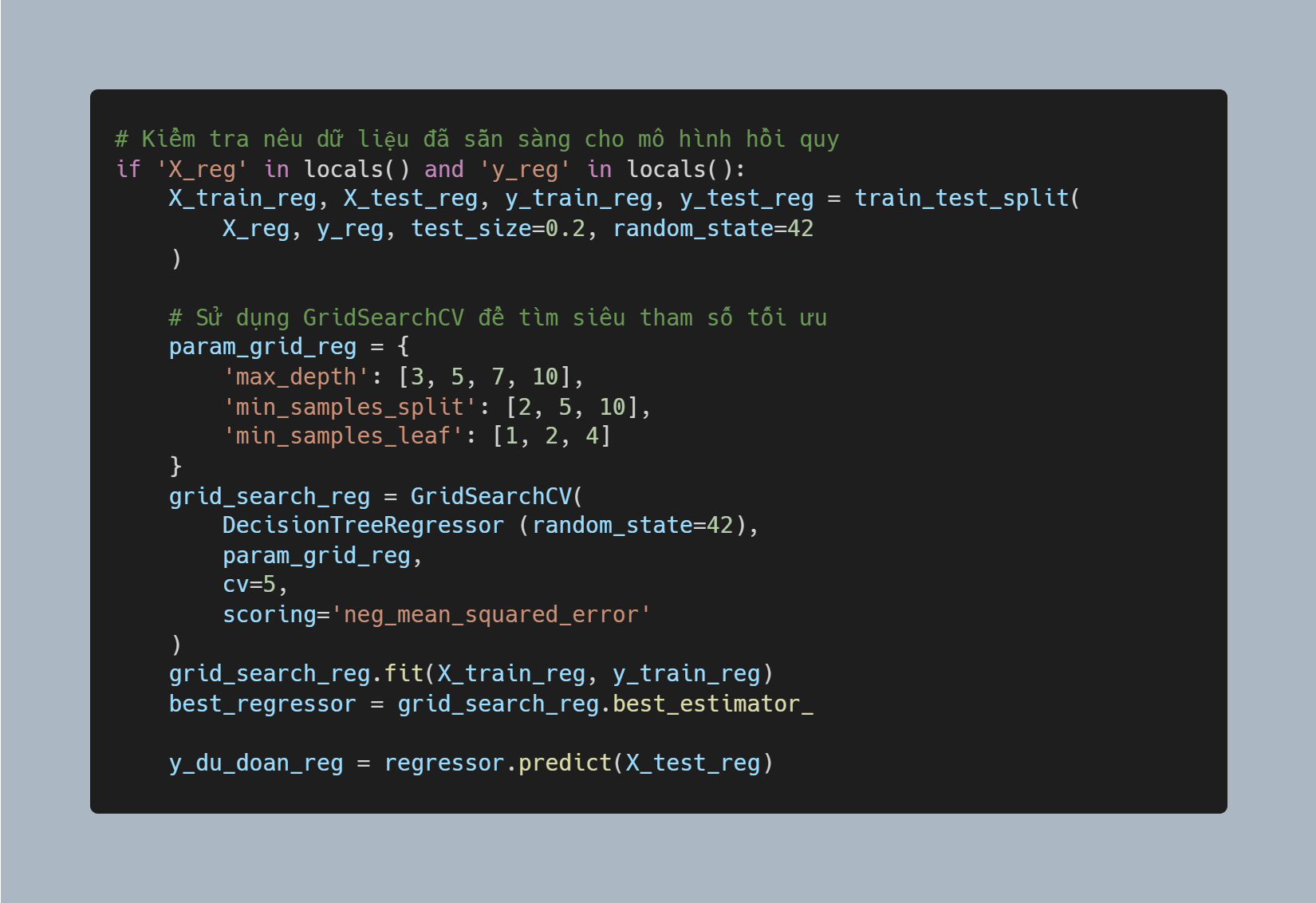
Xử lý dữ liệu đầu vào bằng cách cột mục tiêu Data.Temperature.Avg Temp được sử dụng làm giá trị mục tiêu y\_reg. Các cột không phải số được loại bỏ khỏi tập dữ liệu, chỉ giữ lại các cột số để tạo ra X\_reg tập dữ liệu độc lập. Các cột phân loại sẽ không được sử dụng trong bài toán hồi quy.



Hình .: Đoạn mã kiểm tra dữ liệu hồi quy

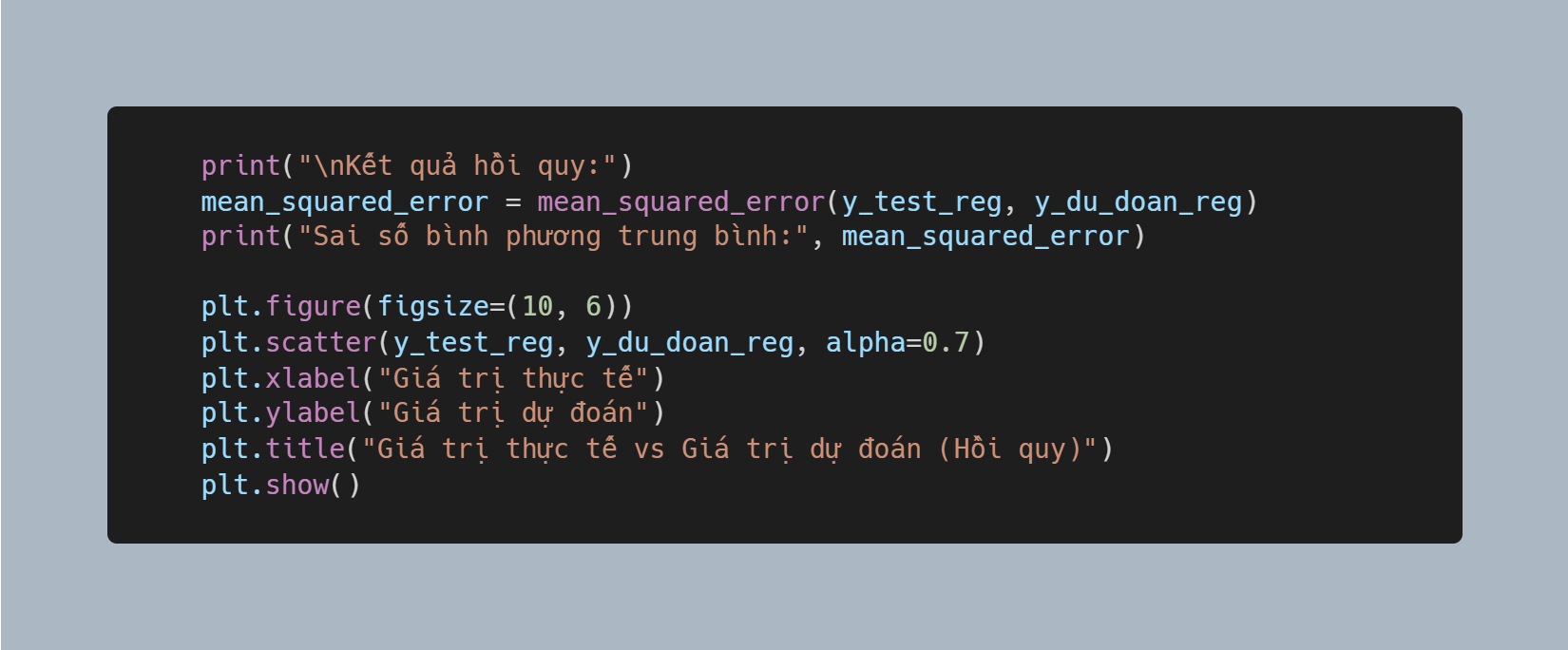
### Huấn luyện mô hình

Dữ liệu sau khi tiền xử lý được chia thành tập huấn luyện (80%) và tập kiểm tra (20%) bằng cách sử dụng train\_test\_split. Để tối ưu hóa các siêu tham số của mô hình **DecisionTreeRegressor**, sử dụng GridSearchCV. Các siêu tham số được tối ưu hóa bao gồm max\_depth, min\_samples\_split và min\_samples\_leaf. Mô hình **DecisionTreeRegressor** sẽ được huấn luyện và tối ưu hóa bằng cách sử dụng độ sai số bình phương trung bình (MSE) làm thước đo hiệu suất.



Hình .: Đoạn mã chia dữ liệu và huấn luyện mô hình hồi quy

### Đánh giá mô hình

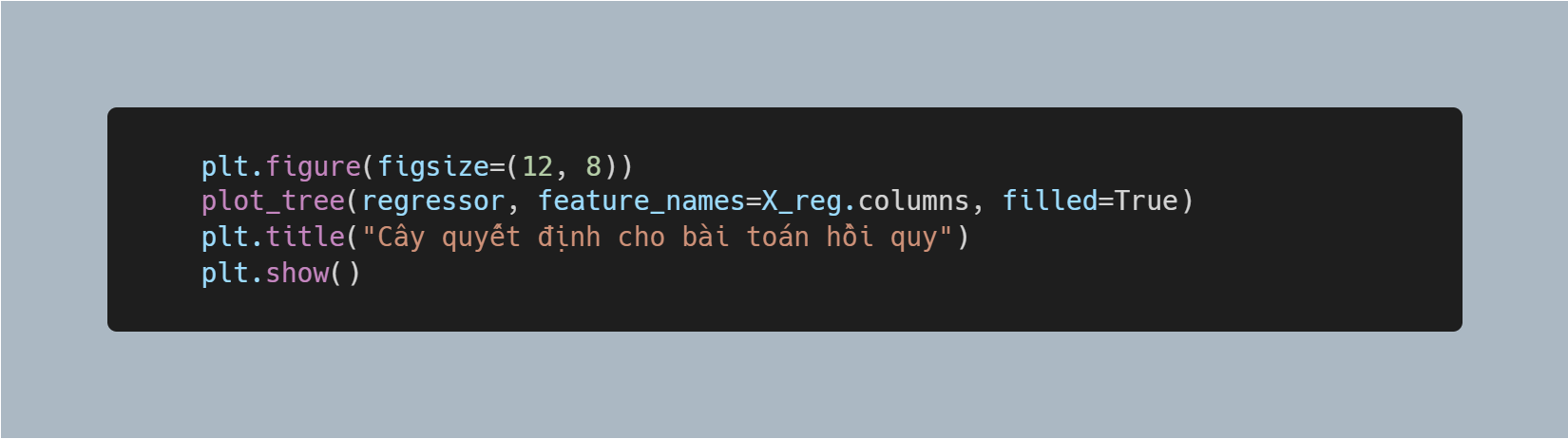


Hình .: Đoạn mã kết quả và biểu đồ so sánh hồi quy

Đánh giá mô hình bằng cách tính toán sai số bình phương trung bình (MSE) giữa giá trị thực tế và giá trị dự đoán trên tập kiểm tra. Để minh họa sự so sánh giữa giá trị thực tế và giá trị dự đoán, biểu đồ phân tán (Scatter Plot) được sử dụng, trong đó trục hoành là giá trị thực tế và trục tung là giá trị dự đoán của mô hình. Nếu mô hình dự đoán tốt, các điểm dữ liệu trên biểu đồ này sẽ phân bố gần đường chéo (y = x), thể hiện rằng giá trị thực tế và giá trị dự đoán gần nhau.

### Trực quan hóa kết quả

Sử dụng mô hình tốt nhất từ GridSearchCV để dự đoán kết quả trên tập kiểm tra. Cây quyết định sẽ được trực quan hóa bằng plot\_tree, giúp người dùng hiểu rõ cách mô hình đưa ra quyết định dựa trên các đặc trưng đầu vào. Mô hình này có thể triển khai để dự đoán nhiệt độ trung bình cho các dữ liệu mới.



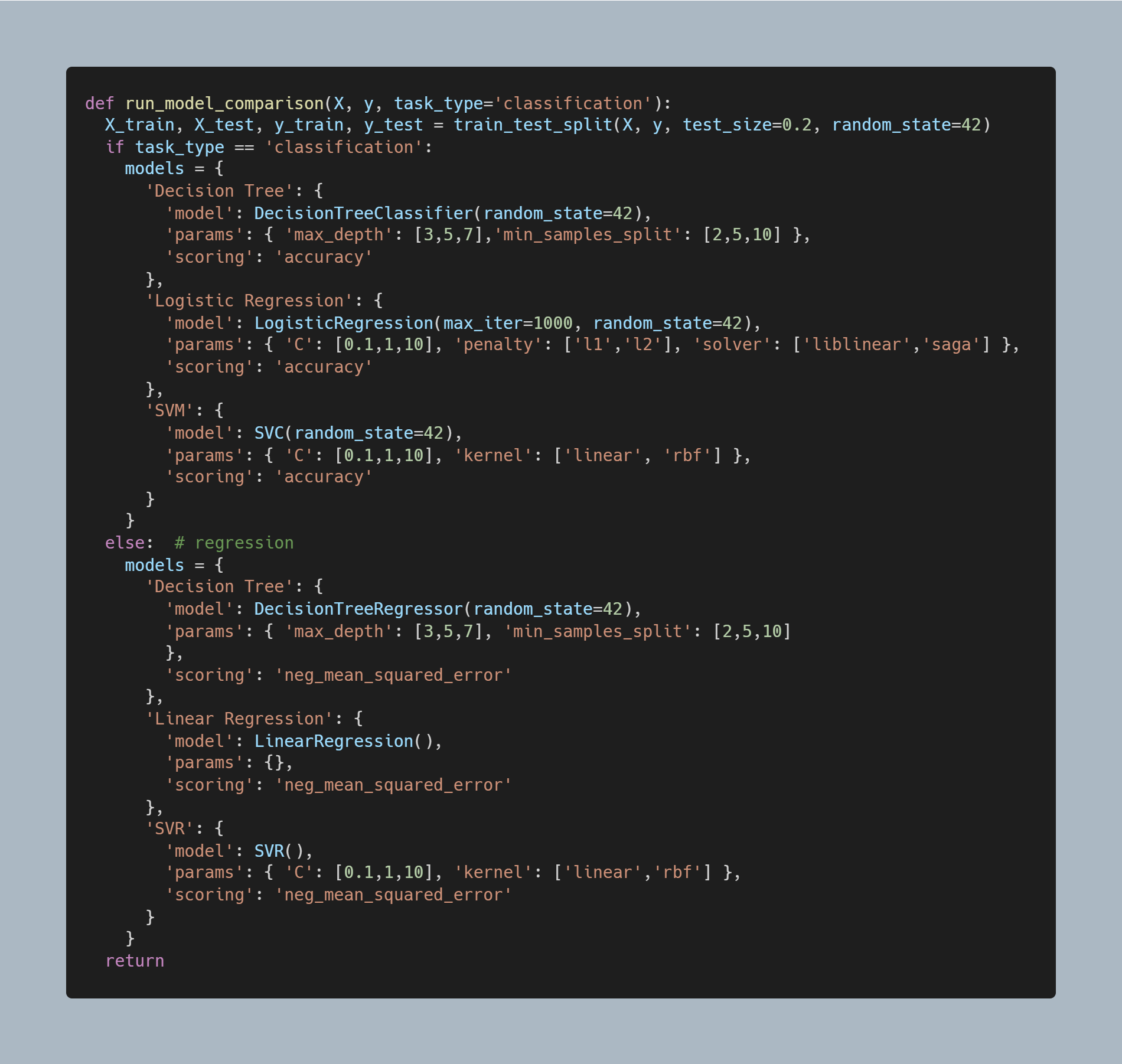
Hình .: Đoạn mã vẽ cây quyết định hồi quy

## So sánh với những thuật toán khác

Để t**ìm mô hình tốt nhất** phù hợp với bài toán (Classification hoặc Regression) thì chúng ta cần so sánh CART với những thuật toán khác như Logistic Regression, SVM (với Classification) hoặc Linear Regression, SVR (với Regression). So sánh này giúp đánh giá ưu nhược điểm của từng mô hình về hiệu suất và tính phù hợp với dữ liệu cụ thể.

Trong dự án này, nhóm 05 sẽ so sánh bằng cách sử dụng GridSearchCV để tìm ra các tham số tối ưu cho mỗi mô hình. Sau đó, đánh giá các mô hình dựa trên các chỉ số như thời gian huấn luyện, thời gian dự đoán, độ chính xác (Accuracy) đối với bài toán phân loại hoặc các chỉ số khác như R-squared và MSE đối với bài toán hồi quy. Cuối cùng, kết quả được trực quan hóa thông qua các biểu đồ. Cụ thể, mã này bao gồm hai hàm chính:

* Hàm run\_model\_comparison: Thực hiện so sánh hiệu suất của các mô hình dựa trên thời gian huấn luyện, thời gian dự đoán, các chỉ số hiệu suất (accuracy, R-squared, MSE).
* Hàm plot\_model\_time\_comparison: Tạo các biểu đồ để so sánh thời gian huấn luyện, thời gian dự đoán và các chỉ số hiệu suất giữa các mô hình.

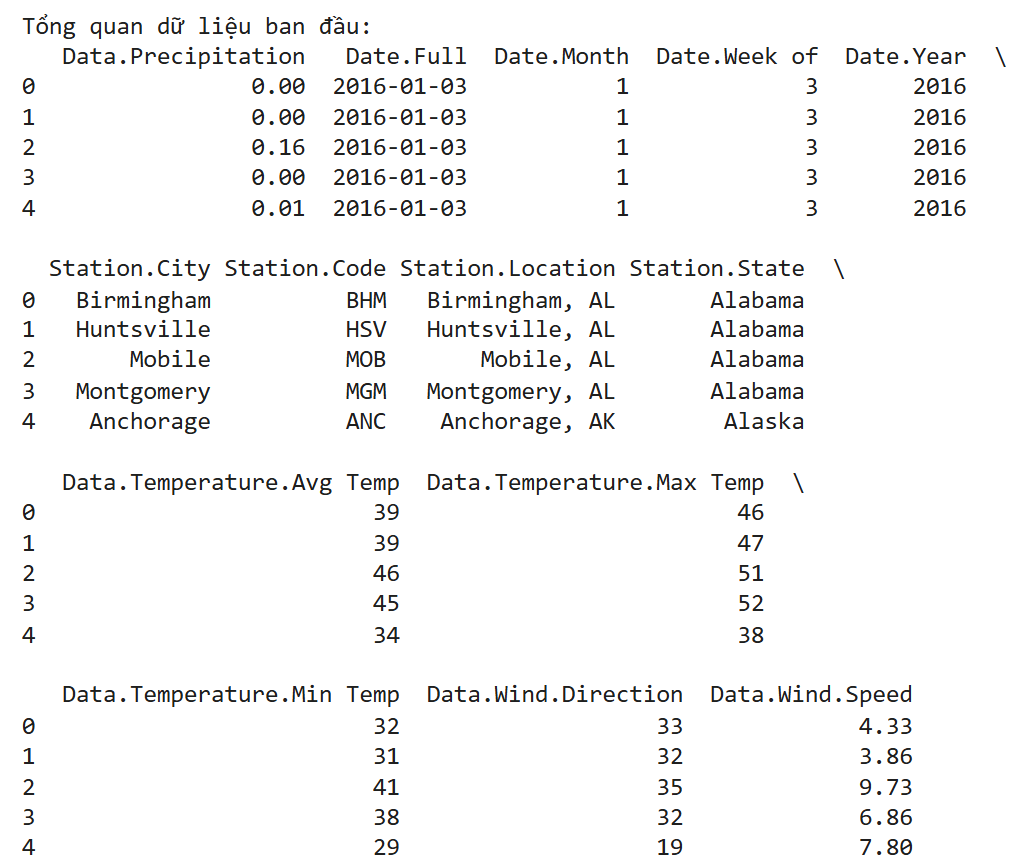


Hình .: Đoạn mã so sánh thuật toán CART với các thuật toán khác

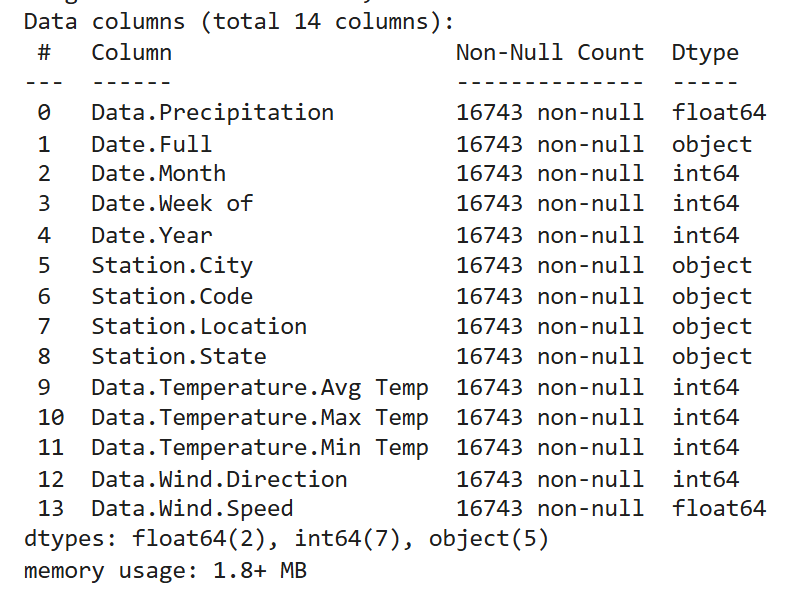
# Kết quả thực nghiệm

## Tổng quan dữ liệu

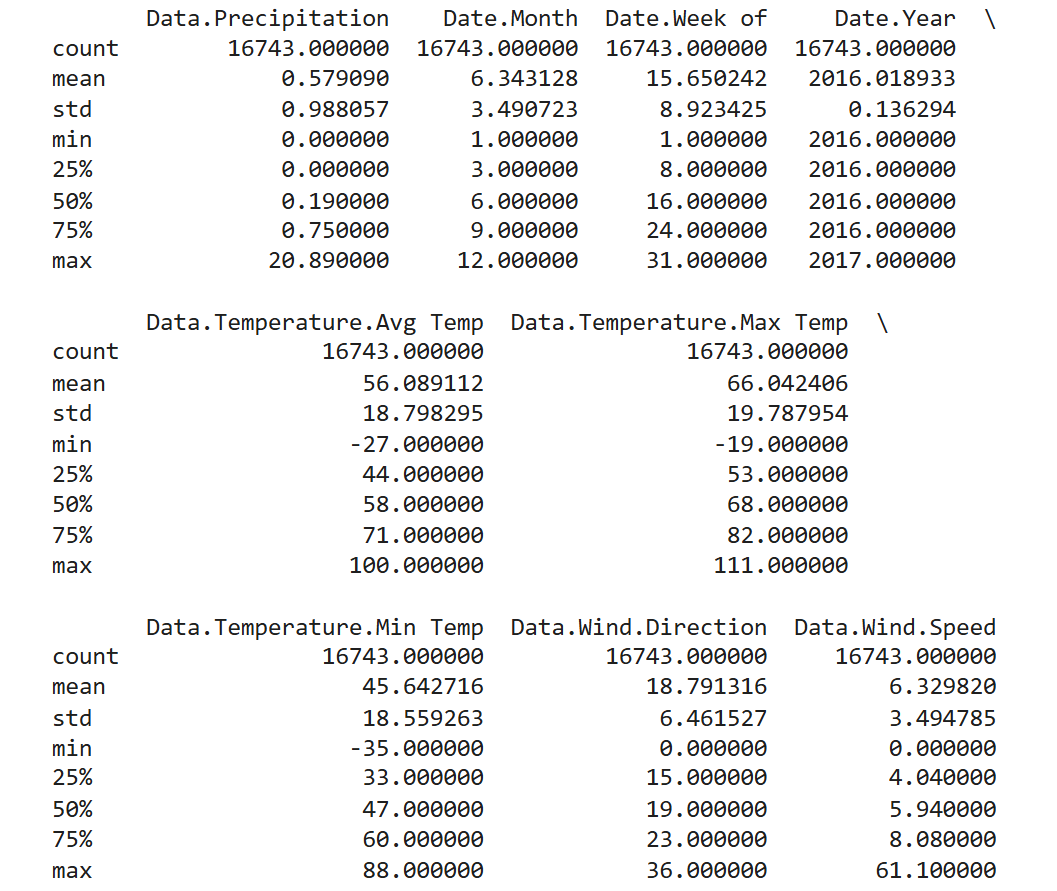
* Hiển thị 5 dòng đầu tiên của tập dữ liệu bằng cách dùng du\_lieu.head() được thể hiện trong hình h.1
* Hiển thị thông tin về các cột và kiểu dữ liệu bằng cách dùng du\_lieu.info() được thể hiện trong hình h.2
* Thống kê cơ bản cho các cột dạng số bằng cách dùng du\_lieu.describe() được thể hiện trong hình h.3



Hình .: Dữ liệu 5 dòng đầu tiên của tập Corgis Weather Dataset



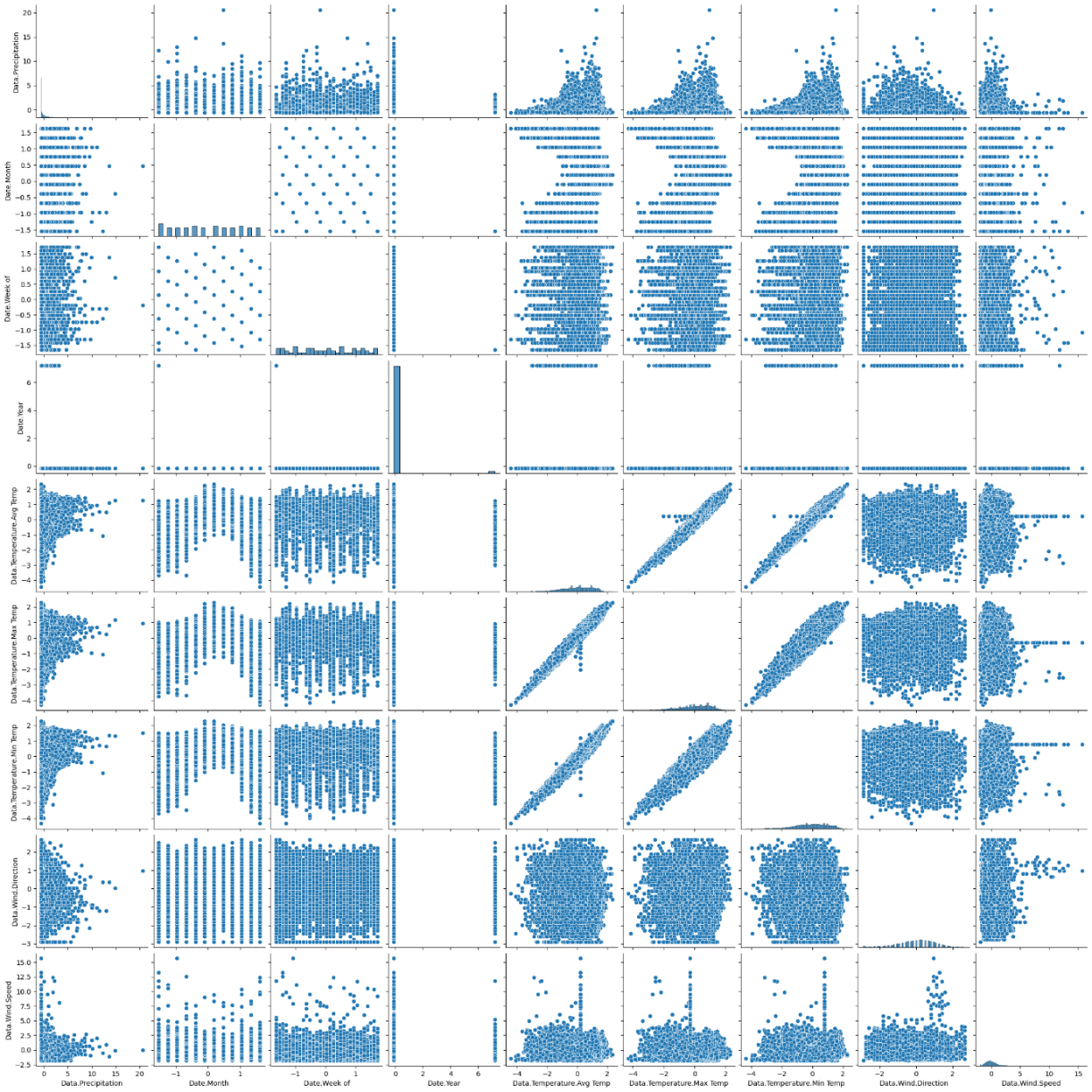
Hình .: Thông tin về các cột và kiểu dữ liệu



Hình .: Thống kê cơ bản cho các cột dạng số

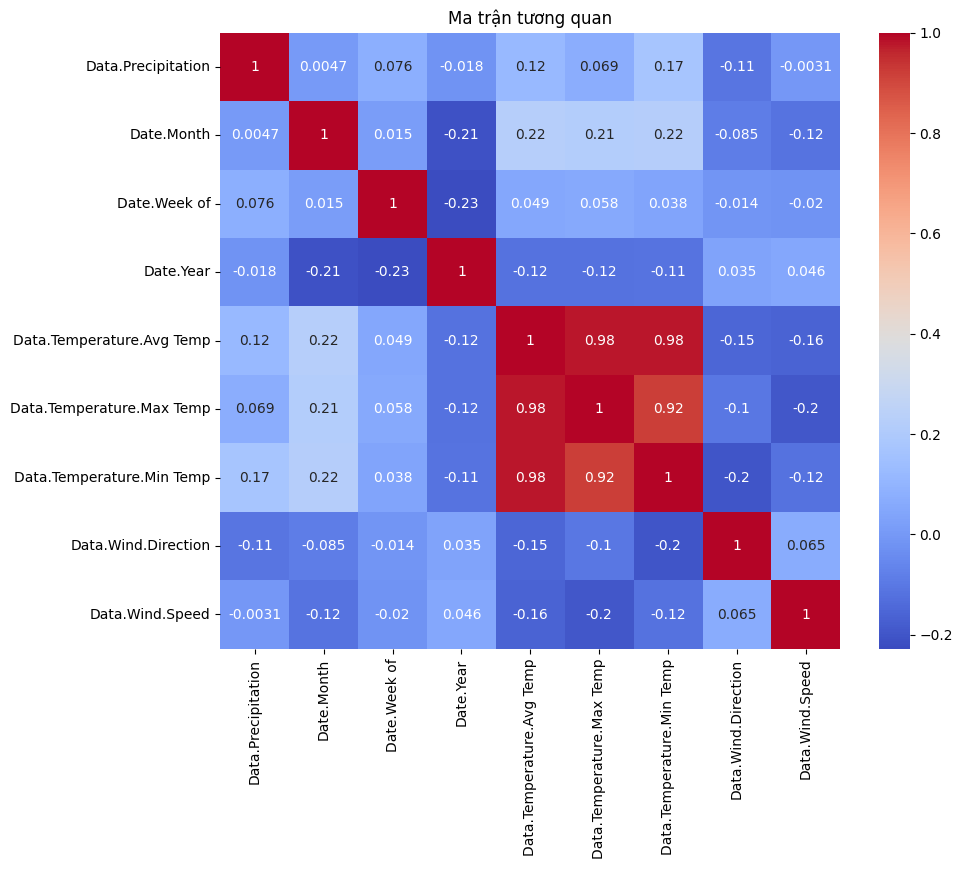
## Trực quan hoá dữ liệu

Biểu đồ Scatter Plot: Cho thấy mối quan hệ giữa các đặc trưng dạng số.



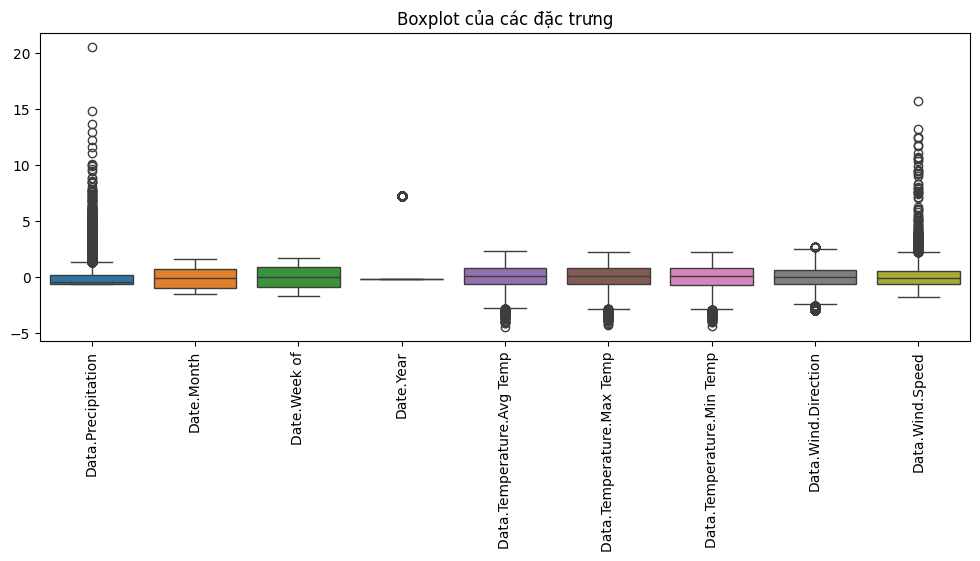
Hình .: Biểu đồ Scatter Plot

Ma trận tương quan: Thể hiện mối quan hệ giữa các cột dạng số.



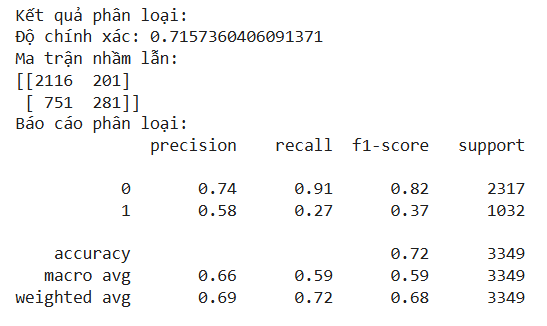
Hình .: Ma trận tương quan

Biểu đồ Boxplot: Giúp phát hiện outlier trong dữ liệu.



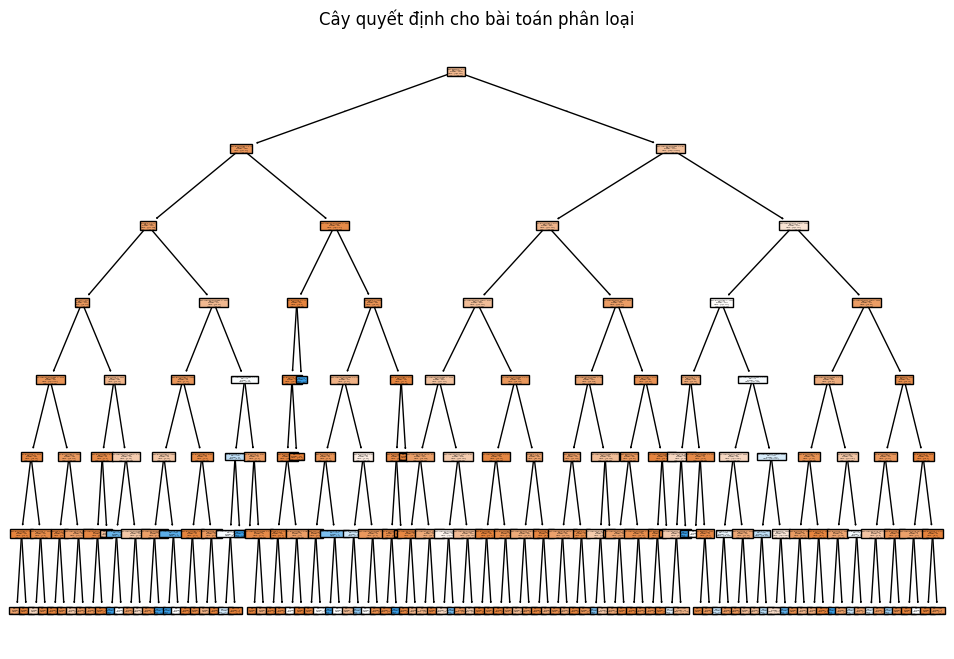
Hình .: Biểu đồ Boxplot

## Huấn luyện mô hình phân loại



Hình .: Kết quả phân loại

Hiển thị cấu trúc cây quyết định với các điều kiện phân nhánh dựa trên đặc trưng.

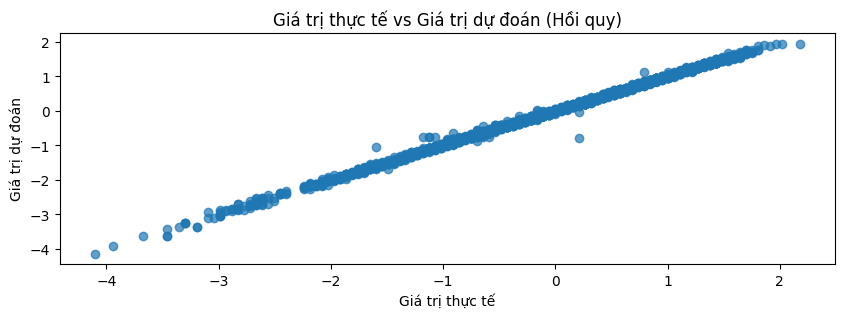


Hình .: Cây quyết định cho bài toán phân loại

## Huấn luyện mô hình hồi quy

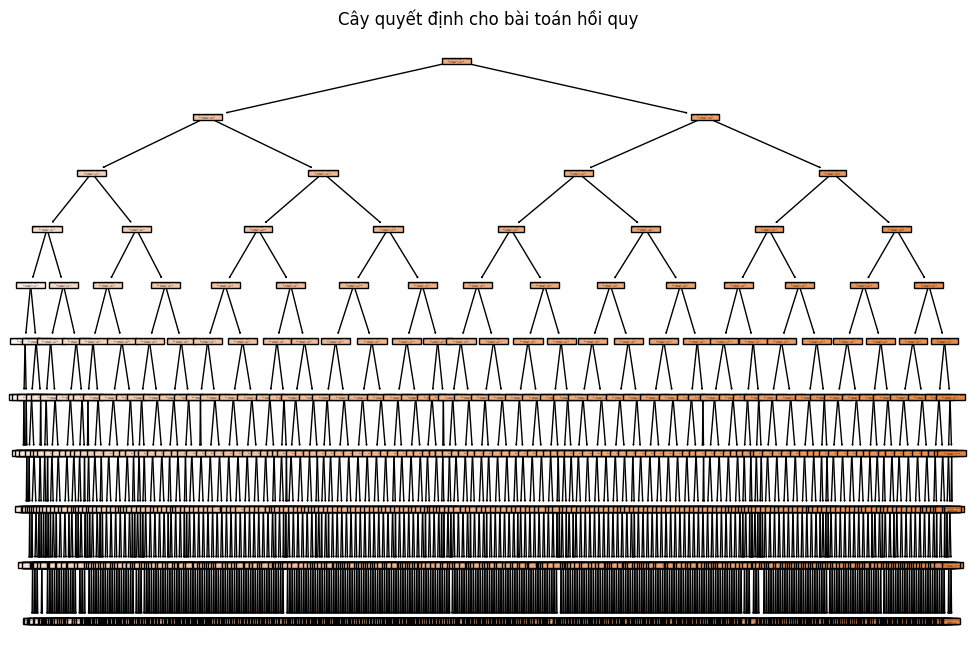


Hình .: Kết quả hồi quy



Hình .: Biểu đồ dự đoán với thực tế

Biểu đồ trên thể hiện mối quan hệ giữa giá trị thực tế và giá trị dự đoán. Các điểm nằm gần đường chéo cho thấy mô hình dự đoán khá chính xác.



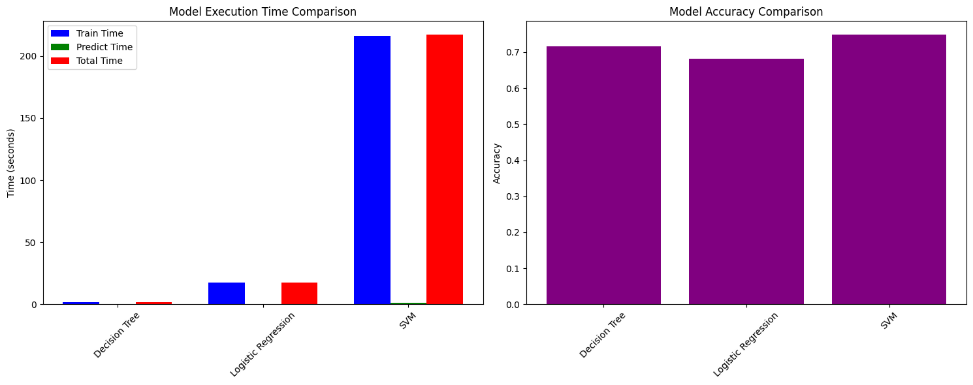
Hình .: Cây quyết định cho bài toán hồi quy

## So sánh với những thuật toán khác

### So sánh hiệu suất

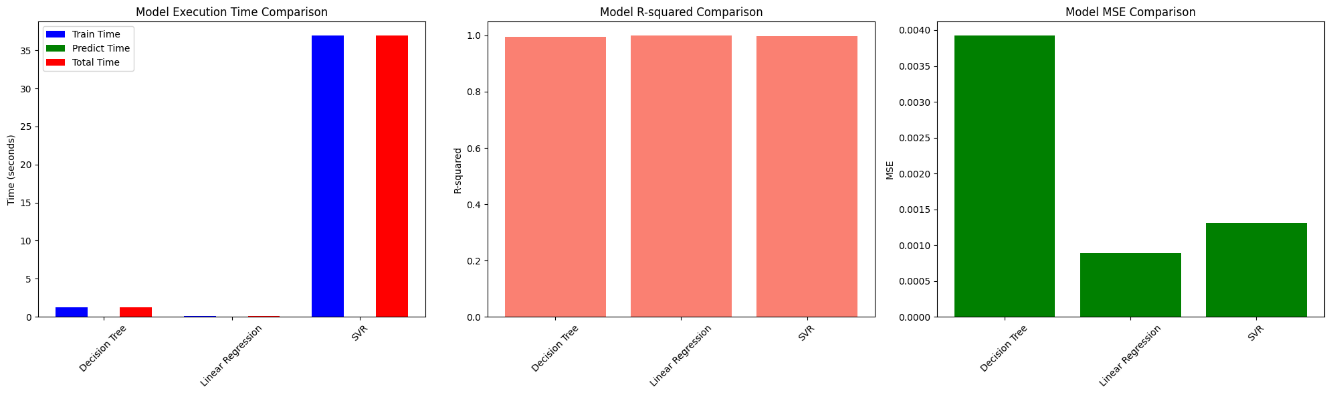
So sánh hiệu suất của CART (Decision Tree) với các mô hình khác trong hai loại bài toán:

* Classification (Phân loại):
* Decision Tree Classifier.
* Logistic Regression.
* Support Vector Machine (SVM).



Hình .: So sánh hiệu suất CART với mô hình khác cho bài toán phân loại

* Regression (Hồi quy):
* Decision Tree Regressor.
* Linear Regression.
* Support Vector Regression (SVR).



Hình .: So sánh hiệu suất CART với mô hình khác cho bài toán hồi quy

### Ý nghĩa kết quả

Decision Tree (CART):

* Có thể vượt trội ở bài toán classification khi dữ liệu không tuyến tính.
* Trong regression, CART có thể gặp vấn đề quá khớp nếu không điều chỉnh độ sâu (max\_depth).

Logistic Regression và Linear Regression:

* Phù hợp với dữ liệu tuyến tính.
* Tốc độ huấn luyện nhanh do tính toán đơn giản hơn.

SVM/SVR:

* Thường có hiệu suất cao nhưng tốn thời gian hơn do tính toán kernel.

Thời gian và hiệu suất:

* CART thường nhanh hơn SVM/SVR nhưng không luôn chính xác bằng.
* Mô hình nào tốt hơn phụ thuộc vào bản chất dữ liệu (tuyến tính hay phi tuyến tính) và yêu cầu về độ chính xác hoặc tốc độ.

# KẾT LUẬN

## Kết quả đạt được

Trong quá trình thực hiện, nhóm 05 đã áp dụng thuật toán CART để dự đoán lượng mưa và nhiệt độ trung bình dựa trên dữ liệu thời tiết. Kết quả cho thấy mô hình phân loại hoạt động tốt với độ chính xác cao khi phân loại các ngày có mưa và không mưa. Đồng thời, mô hình hồi quy cũng dự đoán nhiệt độ trung bình với sai số chấp nhận được.

Việc trực quan hóa dữ liệu như ma trận tương quan, scatter plot và boxplot giúp hiểu rõ hơn về mối quan hệ giữa các đặc trưng và xử lý các điểm ngoại lệ hiệu quả. Nhờ vào quá trình tiền xử lý và chuẩn hóa dữ liệu, mô hình đạt được kết quả khả quan.

## Thuận lợi và khó khăn

### Thuận lợi

* Có nhiều sách, bài báo, nghiên cứu và hướng dẫn trực tuyến cung cấp thông tin chi tiết về thuật toán CART, cũng như các ứng dụng thực tế trong nhiều lĩnh vực.
* Các thư viện như Scikit-learn, Pandas, Numpy cung cấp công cụ mạnh mẽ để triển khai và tối ưu hóa các mô hình dựa trên CART.
* Tích hợp với Python vì Python là một ngôn ngữ lập trình phổ biến, với cú pháp dễ học và thư viện đa dạng, giúp việc phát triển, thử nghiệm, và triển khai các mô hình CART trở nên thuận tiện.

### Khó khăn

* Phải học hỏi và hiểu biết về các khái niệm và cách thức hoạt động của thuật toán để sử dụng hiệu quả như cách xây dựng cây, các tiêu chí chia tách (Gini, Entropy), và các vấn đề như quá khớp (overfitting).
* Triển khai nhiều mô hình cần phải sử dụng nhiều thư viện đòi hỏi kiến thức đủ tốt về Python.

## Hướng phát triển

Xem xét sử dụng thêm các kỹ thuật tối ưu hóa hoặc kết hợp với các thuật toán khác. Kết quả nghiên cứu này là cơ sở để ứng dụng trong các hệ thống dự báo thời tiết hoặc hỗ trợ ra quyết định trong những lĩnh vực hữu ích khác.

## Phân công công việc

Bảng .: Phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ và tên** | **Nội dung thực hiện** | **Mức độ hoàn thành** |
| 52100427 | Nguyễn Trung Kiên | Tìm hiểu thuật toán  Soạn nội dung  Thuyết trình |  |
| 52100752 | Nguyễn Trung Kiên | Tìm hiểu thuật toán  Làm demo  Thuyết trình demo giữa kì |  |
| 52100491 | Nguyễn Thị Kim Tiến | Tìm hiểu thuật toán.  Xem lại slide thuyết trình.  Thuyết trình |  |
| 52100703 | Đinh Phương My | Tìm hiểu thuật toán  Làm Slide, hỗ trợ thuyết trình  Làm báo cáo |  |
| 52100404 | Đặng Hoàng Trúc Duyên | Tìm hiểu thuật toán  Làm demo  Thuyết trình |  |
| 52100439 | Ngô Tấn Lợi | Tìm hiểu thuật toán  Tìm nguồn tài liệu  Thuyết trình |  |
| 51800951 | Ngô Đức Anh Tuấn | Tìm hiểu thuật toán  Kiểm tra và đánh giá nội dung.  Thuyết trình |  |

# TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Nguyễn Xuân Huy, [2020], Học máy ứng dụng với Python, Nhà xuất bản Thống Kê.
2. **Nguyễn Thị Thu Trang, [2020],** Ứng dụng thuật toán CART trong dự báo hành vi khách hàng, Luận văn Thạc sĩ, Đại học Kinh tế Quốc dân.
3. **Đỗ Thanh Tùng, [2021],** Nghiên cứu thuật toán cây quyết định trong phân tích dữ liệu, Luận văn Thạc sĩ, Đại học Bách Khoa Hà Nội.

**Tiếng Anh**

1. Jiawei Han, Micheline Kamber, [2012], *Data Mining: Concepts and Techniques*, 3rd Edition, Morgan Kaufmann Publishers, Waltham, MA.
2. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, [2006], *Introduction to Data Mining*, Addison Wesley, Boston.