TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA: CÔNG NGHỆ THÔNG TIN

----֎----



**MÔN HỌC: PHÂN TÍCH DỮ LIỆU**

**ĐỀ TÀI: Phân tích những yếu tố ảnh hưởng đến mức độ tiêu thụ rượu của sinh viên**

**Giáo viên hướng dẫn: Phan Thành Huấn**

**Sinh viên thực hiện**

|  |  |  |
| --- | --- | --- |
| **STT** | **HỌ VÀ TÊN** | **MSSV** |
| 1 | Hồ Lâm Trường | 3121410540 |
| 2 | Phạm Dương Ngọc Thuận | 3121410483 |

Ngày 17, Tháng 12, Năm 2024

**MỤC LỤC**

[LỜI CẢM ƠN 4](#_Toc184659058)

[NHẬN XÉT CỦA GIÁO VIÊN 5](#_Toc184659059)

[BẢNG PHÂN CÔNG CÔNG VIỆC 6](#_Toc184659060)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỒ ÁN 7](#_Toc184659061)

[1.1 Lý do chọn đề tài 7](#_Toc184659062)

[1.2 Mục tiêu đồ án 7](#_Toc184659063)

[1.3 Mô tả tập dữ liệu 8](#_Toc184659064)

[CHƯƠNG 2:TIỀN XỬ LÝ DỮ LIỆU 14](#_Toc184659065)

[2.1 Làm sạch dữ liệu và chuẩn hóa dữ liệu 14](#_Toc184659066)

[2.1.1 Xóa các dữ liệu không hợp lệ 14](#_Toc184659067)

[2.1.2 Chuẩn hóa dữ liệu 17](#_Toc184659068)

[2.1.3 Kiểm tra các cột dữ liệu 18](#_Toc184659069)

[CHƯƠNG 3:PHÂN TÍCH DỮ LIỆU 20](#_Toc184659070)

[3.1 CÁC PHƯƠNG PHÁP SỬ DỤNG 20](#_Toc184659071)

[3.1.1. Phương pháp phân tích tương quan 20](#_Toc184659072)

[3.1.2.Phân tích hồi quy logistic 20](#_Toc184659073)

[3.1.3.Phân tích sử dụng Cây quyết định (Decision Tree regression) 20](#_Toc184659074)

[3.1.4.Phân tích sử dụng Rừng ngẫu nhiên (Random Forest) 20](#_Toc184659075)

[3.1.5.Phân tích sử dụng Máy vector hỗ trợ (Support Vector Machine) 21](#_Toc184659076)

[3.2 Tìm hiểu mối quan hệ của biến Absences(buổi vắng học) và Walc ( mức độ tiêu thụ rượu vào cuối tuần) 24](#_Toc184659077)

[3.2.1 Phân tích mô hình hồi quy logistic (Logistic Regression) 24](#_Toc184659078)

[3.2.2.Phân tích sử dụng Cây quyết định (Decision Tree) 28](#_Toc184659079)

[3.3.3.Phân tích sử dụng Rừng ngẫu nhiên (Random Forest) 32](#_Toc184659080)

[3.2.3.Phân tích sử dụng Máy vector hỗ trợ (Support Vector Machine) 34](#_Toc184659081)

[3.2.5 So sánh và kết luận giữa 4 phương pháp: Hồi quy Logistic, Cây Quyết Định, Rừng Ngẫu Nhiên và SVM. 37](#_Toc184659082)

[3.3 Tìm hiểu mối quan hệ của biến freetime(thời gian rãnh) và Walc ( mức độ tiêu thụ rượu vào cuối tuần) 39](#_Toc184659083)

[3.3.1 Phân tích mô hình hồi quy logistic (Logistic Regression) 39](#_Toc184659084)

[3.3.2.Phân tích sử dụng Cây quyết định (Decision Tree) 43](#_Toc184659085)

[3.3.3.Phân tích sử dụng Rừng ngẫu nhiên (Random Forest) 47](#_Toc184659086)

[3.3.4.Phân tích sử dụng Máy vector hỗ trợ (Support Vector Machine) 50](#_Toc184659087)

[3.3.5 So sánh và kết luận giữa 4 phương pháp: Hồi quy Logistic, Cây Quyết Định, Rừng Ngẫu Nhiên và SVM. 53](#_Toc184659088)

[3.4.Tìm hiểu mối quan hệ giữa goout và Walc 55](#_Toc184659089)

[3.4.1.Phân tích mô hình hồi quy (Logistic Regression hoặc Multinomial Logistic Regression) 55](#_Toc184659090)

[3.4.2.Phân tích sử dụng Cây quyết định (Decision Tree) 61](#_Toc184659091)

[3.4.3.Rừng Ngẫu Nhiên (Random Forest) 67](#_Toc184659092)

[3.4.4.Phân tích sử dụng Máy vector hỗ trợ (Support Vector Machine) 71](#_Toc184659093)

[3.4.5.So sánh và kết luận giữa 4 phương pháp: Hồi quy Logistic, Cây Quyết Định, Rừng Ngẫu Nhiên và SVM. 75](#_Toc184659094)

[3.5.Tìm hiểu mối quan hệ giữa activities và Walc 77](#_Toc184659095)

[3.5.1.Phân tích mô hình hồi quy (Logistic Regression hoặc Multinomial Logistic Regression) 77](#_Toc184659096)

[3.5.2.Phân tích sử dụng Cây quyết định (Decision Tree) 82](#_Toc184659097)

[3.5.3.Rừng Ngẫu Nhiên (Random Forest) 87](#_Toc184659098)

[3.5.4.Phân tích sử dụng Máy vector hỗ trợ (Support Vector Machine) 91](#_Toc184659099)

[3.5.5.So sánh và kết luận giữa 4 phương pháp: Hồi quy Logistic, Cây Quyết Định, Rừng Ngẫu Nhiên và SVM. 94](#_Toc184659100)

[3.6.Kết luận từ các mối quan hệ 97](#_Toc184659101)

[CHƯƠNG 4: XÂY DỰNG MÔ HÌNH DỰ ĐOÁN MỨC ĐỘ TIÊU THỤ RƯỢU ẢNH HƯỞNG 98](#_Toc184659102)

[1.Mô hình sử dụng để dự đoán mức độ tiêu thụ rượu ảnh hưởng 98](#_Toc184659103)

[2.Xây dựng mô hình Máy Véc-tơ Hỗ trợ (SVM) 99](#_Toc184659104)

[3.Xây dựng mô hình rừng ngẫu nhiên (Random Forest) 103](#_Toc184659105)

[4.So sánh mô hình SVM và Rừng Ngẫu nhiên 104](#_Toc184659106)

[CHƯƠNG 5: KẾT LUẬN 106](#_Toc184659107)

[1.Kết quả đạt được 106](#_Toc184659108)

[2.Hạn chế 107](#_Toc184659109)

[3.Hướng phát triển 107](#_Toc184659110)

[DANH MỤC TÀI LIỆU THAM KHẢO 109](#_Toc184659111)

[DANH MỤC HÌNH ẢNH 110](#_Toc184659112)

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn sâu sắc đến Thầy Phan Thành Huấn, người đã truyền đạt những kiến thức quý báu và hướng dẫn tận tình trong suốt quá trình học môn Phân tích Dữ liệu. Nhờ sự hướng dẫn chi tiết và những gợi ý thực tế từ Thầy, em đã có cơ hội hiểu rõ hơn về các phương pháp phân tích và ứng dụng dữ liệu trong nhiều lĩnh vực. Chính sự nhiệt huyết và tận tâm của Thầy đã khơi dậy niềm đam mê nghiên cứu trong em, đồng thời giúp em hoàn thành bài tiểu luận này. Em hy vọng rằng những kiến thức và kỹ năng mà em tiếp thu được sẽ là nền tảng vững chắc cho con đường học tập và nghề nghiệp của mình trong tương lai. Thông qua bài tiểu luận này, chúng em xin trình bày những gì chúng em đã tìm hiểu về môn học, và cụ thể là chúng em lựa chọn đề tài “Phân tích những yếu tố ảnh hưởng đến mức độ tiêu thụ rượu của sinh viên”.

Kiến thức là vô hạn nhưng sự tiếp nhận kiến thức của bản thân mỗi chúng em luôn hạn chế một cách nhất định. Do đó, trong quá trình hoàn thành bài tiểu luận, chắc chắn chúng em không tránh khỏi những thiếu sót, vậy nên chúng em rất mong nhận được những đóng góp đến từ thầy để bài tiểu luận của nhóm được hoàn thiện hơn.

Nhóm chúng em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, Ngày 17, Tháng 12, Năm 2024

# NHẬN XÉT CỦA GIÁO VIÊN

…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..…………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………..…………

……………………………………………………………………………………..………………………………………………………………………………………………..………………………………………………………………………………………………..………………………………………………………………………………………………..………………………………………………………………………………………………..………………………………………………………………………………………………..………………………………………………………………………………………………..…………………………………………………………………………………………..

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thành viên** | **Phần trăm tham gia hoàn thành đồ án (tổng 100%)** | **Công việc** |
| 1 | Hồ Lâm Trường | 50% | Tiền xử lý, xử lý các mô hình học máy |
| 2 | Phạm Dương Ngọc Thuận | 50% | Tiền xử lý, xử lý các mô hình học máy |

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỒ ÁN

## 1.1 Lý do chọn đề tài

Trong bối cảnh hiện nay, việc tiêu thụ rượu bia trong giới trẻ, đặc biệt là sinh viên, ngày càng trở thành vấn đề quan tâm của xã hội và các nhà nghiên cứu. Nhiều sinh viên, dù chưa đến độ tuổi trưởng thành đầy đủ về mặt kinh tế và xã hội, lại có xu hướng tiêu thụ rượu bia với mức độ cao. Hành vi này không chỉ ảnh hưởng đến sức khỏe và hiệu suất học tập mà còn tiềm ẩn nguy cơ tạo ra những hệ lụy nghiêm trọng về hành vi, tâm lý và tương lai nghề nghiệp. Vì vậy, việc phân tích những yếu tố tác động đến mức độ tiêu thụ rượu của sinh viên là một nghiên cứu có ý nghĩa cả về mặt khoa học lẫn thực tiễn.

Thông qua đề tài "Phân tích những yếu tố ảnh hưởng đến mức độ tiêu thụ rượu của sinh viên", em mong muốn làm rõ những yếu tố khách quan và chủ quan thúc đẩy hành vi này, từ đó giúp tìm ra các giải pháp hỗ trợ và định hướng tích cực cho sinh viên trong việc quản lý và kiểm soát việc sử dụng rượu bia.

Đề tài này cũng phù hợp với mục tiêu của môn học Phân tích Dữ liệu, bởi nó đòi hỏi khả năng thu thập, xử lý và phân tích dữ liệu một cách khoa học để đưa ra các kết luận đáng tin cậy. Với việc áp dụng các phương pháp phân tích dữ liệu, em kỳ vọng sẽ tìm ra mối quan hệ giữa các yếu tố như hoàn cảnh gia đình, áp lực học tập, sự ảnh hưởng từ bạn bè, và môi trường xã hội đến hành vi tiêu thụ rượu của sinh viên. Đây không chỉ là cơ hội để em áp dụng kiến thức đã học vào thực tế mà còn là dịp để đóng góp những góc nhìn mới mẻ và hữu ích cho các vấn đề đang tồn tại trong cộng đồng sinh viên

## 1.2 Mục tiêu đồ án

Mục tiêu của đồ án "Các yếu tố ảnh hưởng đến mức độ tiêu thụ rượu trong phạm vi sinh viên và học sinh" là nghiên cứu và phân tích các yếu tố ảnh hưởng đến thói quen tiêu thụ rượu của sinh viên và học sinh, nhằm hiểu rõ hơn về hành vi và các yếu tố xã hội, tâm lý liên quan đến việc sử dụng rượu trong nhóm đối tượng này. Đồ án không chỉ tập trung vào việc xây dựng mô hình dự đoán mà còn muốn khám phá những tác động của các yếu tố như tuổi tác, giới tính, tình trạng học tập, thói quen sinh hoạt, mức độ căng thẳng, hoạt động ngoại khóa, và môi trường bạn bè đến mức độ tiêu thụ rượu.

Trong bối cảnh sinh viên và học sinh, việc tiêu thụ rượu là một vấn đề xã hội và sức khỏe đáng lo ngại, đặc biệt là khi thói quen này có thể dẫn đến các hệ quả tiêu cực về sức khỏe, học tập và đời sống xã hội. Do đó, mục tiêu của đồ án là phát hiện và chỉ ra những yếu tố có khả năng gây ảnh hưởng mạnh mẽ đến việc tiêu thụ rượu trong nhóm đối tượng này, từ đó tạo ra cơ sở cho các chương trình giáo dục sức khỏe và can thiệp phòng ngừa.

Đồ án sẽ sử dụng mô hình học máy rừng ngẫu nhiên (Random Forest) để phân tích và dự đoán mức độ tiêu thụ rượu dựa trên các yếu tố đầu vào như hoạt động ngoại khóa, thời gian rảnh, mức độ căng thẳng trong học tập, tình trạng bạn bè và các yếu tố tâm lý khác. Mô hình này sẽ giúp xác định mối quan hệ giữa các yếu tố này và thói quen uống rượu của sinh viên, học sinh, từ đó cung cấp những thông tin quan trọng cho các cơ sở giáo dục và các tổ chức y tế.

Mô hình Random Forest sẽ giúp phân tích dữ liệu từ các cuộc khảo sát, từ đó chỉ ra các nhóm đối tượng có nguy cơ cao và các yếu tố cần tập trung trong công tác tuyên truyền và can thiệp. Đồ án còn đặt mục tiêu cung cấp những chỉ số và thông tin có giá trị, giúp các cơ sở giáo dục và các tổ chức y tế thiết kế các chương trình can thiệp, nâng cao nhận thức và khuyến khích sinh viên, học sinh có những thói quen tiêu thụ rượu lành mạnh và có trách nhiệm.

Cuối cùng, đồ án không chỉ mang tính chất nghiên cứu lý thuyết mà còn đóng góp vào việc xây dựng các chiến lược phòng ngừa và giáo dục sức khỏe hiệu quả trong môi trường học đường. Các kết quả từ đồ án có thể giúp các nhà quản lý giáo dục, các tổ chức y tế, và các bậc phụ huynh có cái nhìn rõ ràng hơn về các yếu tố ảnh hưởng đến việc tiêu thụ rượu của sinh viên, học sinh, từ đó đưa ra các biện pháp can thiệp và hỗ trợ phù hợp.

## 1.3 Mô tả tập dữ liệu

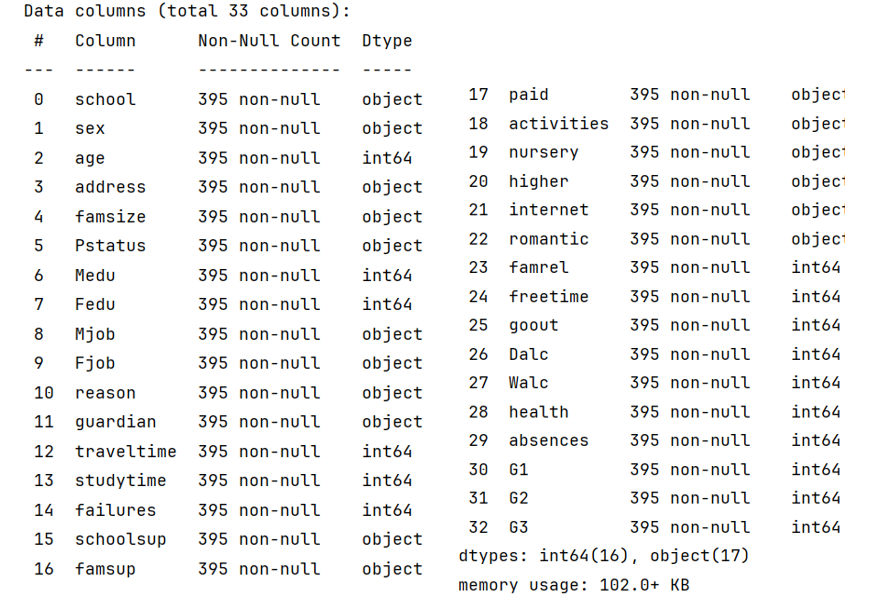
Bộ dữ liệu “student-mat.csv” từ Kaggle <https://www.kaggle.com/> cung cấp thông tin chi tiết về việc sinh viên tiêu thụ rượu tại các trường đại học ở Bồ Đào Nha. Bộ dữ liệu này không chỉ tập trung vào hành vi tiêu thụ rượu mà còn bao gồm các thông tin về môi trường, gia đình và xã hội của sinh viên. Mục đích của nghiên cứu là để hiểu rõ hơn về mối liên hệ giữa các yếu tố này và thói quen tiêu thụ rượu của sinh viên trong 1 ngày. Dữ liệu được thu thập từ các cuộc khảo sát trong cộng đồng học đường, mang lại cơ hội để nghiên cứu tác động của môi trường và gia đình đối với hành vi tiêu thụ rượu ở độ tuổi sinh viên.

Bộ dữ liệu bao gồm 395 quan sát trên tổng 33 biến về sinh viên của 2 trường đại học ở Bồ Đào Nha. Dưới đây là bảng chi tiết từng biến trên từng quan sát khác nhau

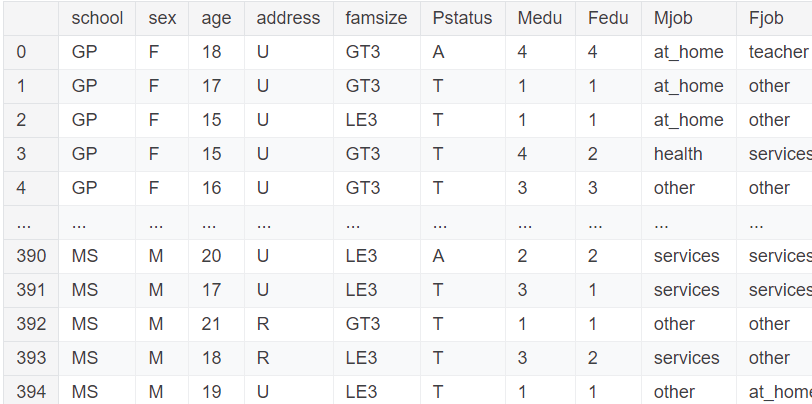
* **schools** - trường của học sinh (nhị phân: 'GP' - Gabriel Pereira hoặc 'MS' - Mousinho da Silveira)
* **sex** - giới tính của học sinh (nhị phân: 'F' - nữ hoặc 'M' - nam)
* **age** - tuổi của học sinh (số: từ 15 đến 22)
* **address** - loại địa chỉ nhà của học sinh (nhị phân: 'U' - thành thị hoặc 'R' - nông thôn)
* **famsize** - kích thước gia đình (nhị phân: 'LE3' - nhỏ hơn hoặc bằng 3 hoặc 'GT3' - lớn hơn 3)
* **Pstatus** - tình trạng sống chung của bố mẹ (nhị phân: 'T' - sống cùng nhau hoặc 'A' - sống xa nhau)
* **Medu** - trình độ học vấn của mẹ (số: 0 - không có, 1 - tiểu học (lớp 4), 2 - lớp 5 đến lớp 9, 3 - trung học phổ thông hoặc 4 - đại học)
* **Fedu** - trình độ học vấn của bố (số: 0 - không có, 1 - tiểu học (lớp 4), 2 - lớp 5 đến lớp 9, 3 - trung học phổ thông hoặc 4 - đại học)
* **Mjob** - công việc của mẹ (danh nghĩa: 'teacher' - giáo viên, 'health' - liên quan đến chăm sóc sức khỏe, 'services' - dịch vụ công (ví dụ: hành chính hoặc cảnh sát), 'at\_home' - nội trợ hoặc 'other' - khác)
* **Fjob** - công việc của bố (danh nghĩa: 'teacher' - giáo viên, 'health' - liên quan đến chăm sóc sức khỏe, 'services' - dịch vụ công (ví dụ: hành chính hoặc cảnh sát), 'at\_home' - nội trợ hoặc 'other' - khác)
* **reason** - lý do chọn trường này (danh nghĩa: gần 'home' - nhà, 'reputation' - danh tiếng trường, 'course' - lựa chọn khóa học hoặc 'other' - khác)
* **guardian** - người giám hộ của học sinh (danh nghĩa: 'mother' - mẹ, 'father' - bố hoặc 'other' - khác)
* **traveltime** - thời gian di chuyển từ nhà đến trường (số: 1 - 1 giờ)
* **studytime** - thời gian học hàng tuần (số: 1 - 10 giờ)
* **failures** - số lần học sinh thi trượt trước đó (số: n nếu 1<=n<3, nếu không thì là 4)
* **schoolsup** - hỗ trợ giáo dục bổ sung (nhị phân: có hoặc không)
* **famsup** - hỗ trợ học tập từ gia đình (nhị phân: có hoặc không)
* **paid** - lớp học thêm trong môn học chính (Toán hoặc Ngữ văn Bồ Đào Nha) (nhị phân: có hoặc không)
* **activities** - các hoạt động ngoại khóa (nhị phân: có hoặc không)
* **nursery** - có học mẫu giáo không (nhị phân: có hoặc không)
* **higher** - mong muốn học lên đại học (nhị phân: có hoặc không)
* **internet** - có truy cập Internet tại nhà (nhị phân: có hoặc không)
* **romantic** - có mối quan hệ tình cảm (nhị phân: có hoặc không)
* **famrel** - chất lượng mối quan hệ gia đình (số: từ 1 - rất tệ đến 5 - rất tốt)
* **freetime** - thời gian rảnh sau giờ học (số: từ 1 - rất ít đến 5 - rất nhiều)
* **goout** - tần suất đi chơi với bạn bè (số: từ 1 - rất ít đến 5 - rất nhiều)
* **Dalc** - mức độ tiêu thụ rượu trong ngày làm việc (số: từ 1 - rất ít đến 5 - rất nhiều)
* **Walc** - mức độ tiêu thụ rượu vào cuối tuần (số: từ 1 - rất ít đến 5 - rất nhiều)
* **health** - tình trạng sức khỏe hiện tại (số: từ 1 - rất tệ đến 5 - rất tốt)
* **absences** - số ngày vắng mặt tại trường (số: từ 0 đến 93)

Các điểm số này liên quan đến môn học, Toán hoặc Ngữ văn Bồ Đào Nha:

* **G1** - điểm học kỳ đầu tiên (số: từ 0 đến 20)
* **G2** - điểm học kỳ thứ hai (số: từ 0 đến 20)
* **G3** - điểm cuối cùng (số: từ 0 đến 20, là mục tiêu đầu ra).



Hình : thông tin tổng quan về dữ liệu



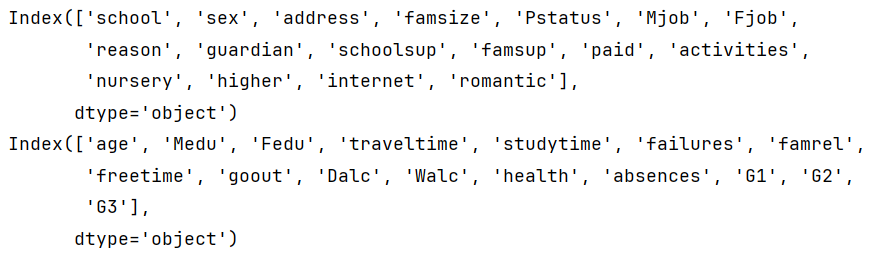
Hình : Tổng quan dữ liệu

Bộ dữ liệu bao gồm 2 loại thông tin chính: Biến phân loại và Biến số

A computer code with text

Description automatically generated

Hình : Đoạn code giúp phân biêt biến phân loại và biến số



Hình : Kết quả cho ra , Biến phân loại và Biến số trong tập dữ liệu

Bảng mô tả dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | School | Object | Ngôi trường sinh viên đang theo học (GP: Gabriel Pereira; MS: Mousinho da Silveria) |
| 2 | Sex | Object | Giới tính của sinh viên (F: nữ; M: nam) |
| 3 | Age | Int64 | tuổi của học sinh (số: từ 15 đến 22) |
| 4 | Address | Object | Nơi ở của sinh viên (U: thành thị; R: nông thôn) |
| 5 | Famsize | Object | kích thước gia đình (nhị phân: 'LE3' - nhỏ hơn hoặc bằng 3 hoặc 'GT3' - lớn hơn 3) |
| 6 | Pstatus | Object | tình trạng sống chung của bố mẹ (nhị phân: 'T' - sống cùng nhau hoặc 'A' - sống xa nhau) |
| 7 | Medu | Int64 | trình độ học vấn của mẹ (số: 0 - không có, 1 - tiểu học (lớp 4), 2 - lớp 5 đến lớp 9, 3 - trung học phổ thông hoặc 4 - đại học) |
| 8 | Fedu | Int64 | trình độ học vấn của bố (số: 0 - không có, 1 - tiểu học (lớp 4), 2 - lớp 5 đến lớp 9, 3 - trung học phổ thông hoặc 4 - đại học) |
| 9 | Mjob | Object | công việc của mẹ (danh nghĩa: 'teacher' - giáo viên, 'health' - liên quan đến chăm sóc sức khỏe, 'services' - dịch vụ công (ví dụ: hành chính hoặc cảnh sát), 'at\_home' - nội trợ hoặc 'other' - khác) |
| 10 | Fjob | Object | công việc của bố (danh nghĩa: 'teacher' - giáo viên, 'health' - liên quan đến chăm sóc sức khỏe, 'services' - dịch vụ công (ví dụ: hành chính hoặc cảnh sát), 'at\_home' - nội trợ hoặc 'other' - khác) |
| 11 | Reson | Object | lý do chọn trường này (danh nghĩa: gần 'home' - nhà, 'reputation' - danh tiếng trường, 'course' - lựa chọn khóa học hoặc 'other' - khác) |
| 12 | Guardian | Object | người giám hộ của học sinh (danh nghĩa: 'mother' - mẹ, 'father' - bố hoặc 'other' - khác) |
| 13 | Traveltime | Int64 | thời gian di chuyển từ nhà đến trường (số: 1 - 1 giờ) |
| 14 | Studytime | Int64 | thời gian học hàng tuần (số: 1 - 10 giờ) |
| 15 | Failures | Int64 | số lần học sinh thi trượt trước đó (số: n nếu 1<=n<3, nếu không thì là 4) |
| 16 | Schoolsup | Object | Hỗ trợ giáo dục bổ sung (yes: có; no: không) |
| 17 | Famsup | Object | Hỗ trợ giáo dục gia đình (yes: có; no: không) |
| 18 | Paid | Object | lớp học thêm trong môn học chính (Toán hoặc Ngữ văn Bồ Đào Nha) (nhị phân: có hoặc không) |
| 19 | Activities | Object | Hoạt động ngoại khóa (yes: có; no: không) |
| 20 | Nursery | Object | có học mẫu giáo không (nhị phân: có hoặc không) |
| 21 | Higher | Object | mong muốn học lên đại học (nhị phân: có hoặc không) |
| 22 | Internet | Object | Truy cập mạng tại nhà (yes: có; no: không) |
| 23 | Romantic | Object | Mối quan hệ tình cảm (yes: có; no: không) |
| 24 | Famrel | Int64 | chất lượng mối quan hệ gia đình (số: từ 1 - rất tệ đến 5 - rất tốt) |
| 25 | Freetime | Int64 | thời gian rảnh sau giờ học (số: từ 1 - rất ít đến 5 - rất nhiều) |
| 26 | Goout | Int64 | tần suất đi chơi với bạn bè (số: từ 1 - rất ít đến 5 - rất nhiều) |
| 27 | Dalc | Int64 | Mức tiêu thụ rượu trong 1 ngày (1: rất thấp; 2: thấp; 3: trung bình; 4: cao; 5: rất cao) |
| 28 | Walc | Int64 | mức độ tiêu thụ rượu vào cuối tuần (số: từ 1 - rất ít đến 5 - rất nhiều) |
| 29 | Health | Int64 | tình trạng sức khỏe hiện tại (số: từ 1 - rất tệ đến 5 - rất tốt) |
| 30 | Absences | Int64 | số ngày vắng mặt tại trường (số: từ 0 đến 93) |
| 31 | G1 | Int64 | điểm học kỳ đầu tiên (số: từ 0 đến 20) |
| 32 | G2 | Int64 | điểm học kỳ thứ hai (số: từ 0 đến 20) |
| 33 | G3 | Int64 | điểm cuối cùng (số: từ 0 đến 20, là mục tiêu đầu ra). |

# CHƯƠNG 2:TIỀN XỬ LÝ DỮ LIỆU

## 2.1 Làm sạch dữ liệu và chuẩn hóa dữ liệu

### **2.1.1 Xóa các dữ liệu không hợp lệ**

* **Tiến hành kiểm tra các dữ liệu trùng nhau:**

import pandas as pd

# Đường dẫn tệp dữ liệu

file\_path = 'D:\\python\\BAITAPDULIEU\\DULIEU\\student-mat-binary.csv'

data = pd.read\_csv(file\_path)

# Đếm số lượng hàng trùng lặp

duplicate\_count = data.duplicated().sum()

# Xóa các hàng trùng lặp

data\_cleaned = data.drop\_duplicates()

# Kiểm tra số lượng hàng trước và sau khi xóa trùng lặp

original\_row\_count = data.shape[0]

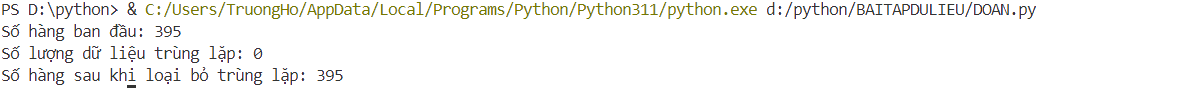
cleaned\_row\_count = data\_cleaned.shape[0]

print("Số hàng ban đầu:", original\_row\_count)

print("Số lượng dữ liệu trùng lặp:", duplicate\_count)

print("Số hàng sau khi loại bỏ trùng lặp:", cleaned\_row\_count)

**Kết quả:**



Hình : kết quả của làm sạch dữ liệu

**Diễn giải kết quả trên:**

* **Số hàng ban đầu (original\_row\_count)**: 395

Đây là tổng số hàng (dữ liệu) ban đầu trong tệp student-mat.csv.

* **Số lượng dữ liệu trùng lặp (duplicate\_count)**: 0

Trong dữ liệu ban đầu, có 0 hàng trùng lặp.

* **Số hàng sau khi loại bỏ trùng lặp (cleaned\_row\_count)**: **395** – Sau khi thực hiện bước loại bỏ trùng lặp, số hàng vẫn là 395, khẳng định rằng không có hàng nào bị loại bỏ vì không có dữ liệu trùng lặp.
* Tập dữ liệu của bạn có 395 hàng duy nhất, và không có hàng nào bị trùng lặp.
* **Kiểm tra có tồn tại các giá trị thiếu.**

import pandas as pd

# Đường dẫn tệp dữ liệu

file\_path = 'D:\\python\\BAITAPDULIEU\\DULIEU\\student-mat-binary.csv'

data = pd.read\_csv(file\_path)

# Kiểm tra giá trị thiếu

missing\_values\_count = data.isnull().sum() # Đếm số lượng giá trị thiếu trong mỗi cột

total\_missing\_values = missing\_values\_count.sum() # Tổng số giá trị thiếu trong toàn bộ dữ liệu

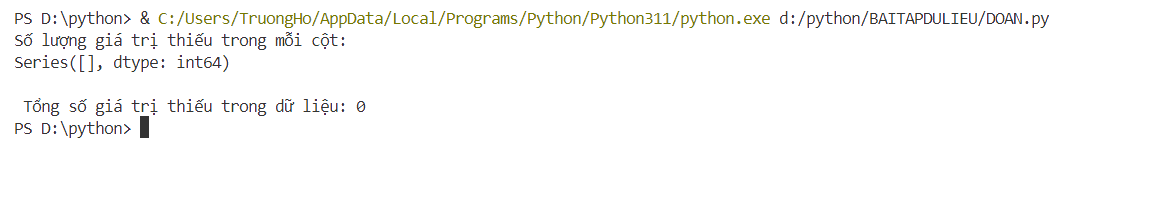
# In kết quả

print("Số lượng giá trị thiếu trong mỗi cột:")

print(missing\_values\_count[missing\_values\_count > 0])

print("\n Tổng số giá trị thiếu trong dữ liệu:", total\_missing\_values)

**Kết quả:**

****

Hình : kết quả kiểm tra giá trị thiếu

**Diễn giải kết quả trên:**

* **Số lượng giá trị thiếu trong mỗi cột:**

Dòng kết quả này trống, nghĩa là không có cột nào có giá trị thiếu. Điều này cho thấy tất cả các cột trong tập dữ liệu student-mat.csv đều đầy đủ và không có giá trị NaN hoặc giá trị thiếu.

* **Tổng số giá trị thiếu trong dữ liệu:**

Kết quả là 0, xác nhận rằng không có bất kỳ giá trị nào bị thiếu trong toàn bộ tập dữ liệu.

* **Kết luận:**Tập dữ liệu student-mat.csv hoàn toàn sạch sẽ về mặt dữ liệu thiếu. Điều này có nghĩa có thể tiến hành phân tích dữ liệu mà không cần xử lý thêm cho các giá trị thiếu, giúp quá trình phân tích trở nên dễ dàng và chính xác hơn.
* **Tìm các dữ liệu ngoại lai và xóa**

**import pandas as pd**

**# Tải dữ liệu**

**data = pd.read\_csv('D:\\python\\BAITAPDULIEU\\DULIEU\\student-mat-binary.csv')**

**# Hàm loại bỏ ngoại lai dựa trên IQR**

**def remove\_outliers(df, columns):**

**df\_cleaned = df.copy()**

**for column in columns:**

**Q1 = df\_cleaned[column].quantile(0.25)**

**Q3 = df\_cleaned[column].quantile(0.75)**

**IQR = Q3 - Q1**

**lower\_bound = Q1 - 1.5 \* IQR**

**upper\_bound = Q3 + 1.5 \* IQR**

**df\_cleaned = df\_cleaned[(df\_cleaned[column] >= lower\_bound) & (df\_cleaned[column] <= upper\_bound)]**

**return df\_cleaned**

**# Xác định các cột số liệu để loại bỏ ngoại lai**

**numerical\_columns = data.select\_dtypes(include=['int64', 'float64']).columns**

**# Loại bỏ ngoại lai khỏi tập dữ liệu**

**data\_no\_outliers = remove\_outliers(data, numerical\_columns)**

**# So sánh kích thước tập dữ liệu trước và sau khi loại bỏ ngoại lai**

**print("Kích thước ban đầu:", data.shape)**

**print("Kích thước sau khi loại bỏ ngoại lai:", data\_no\_outliers.shape)**

**# Lưu dữ liệu đã chuẩn hóa**

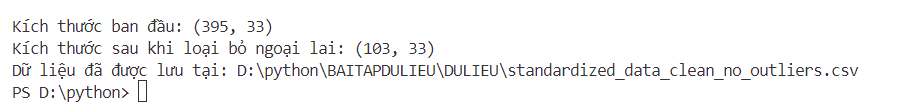
**output\_file\_path = 'D:\\python\\BAITAPDULIEU\\DULIEU\\standardized\_data\_clean\_no\_outliers.csv'**

**data\_no\_outliers.to\_csv(output\_file\_path, index=False)**

**# Trả về thông báo xác nhận**

**print("Dữ liệu đã được lưu tại:", output\_file\_path)**

**Kết quả:**

****

Hình : Kết quả của giá trị ngoại lai

**Diễn giải kết quả trên:**

**Kích thước ban đầu**:

* Dữ liệu gốc có **395 hàng** và **33 cột**.
* Đây là tập dữ liệu ban đầu trước khi xử lý ngoại lai.

**Kích thước sau khi loại bỏ ngoại lai**:

* Sau khi áp dụng phương pháp loại bỏ ngoại lai dựa trên IQR (Interquartile Range), chỉ còn **103 hàng** và **33 cột**.
* Điều này cho thấy có **292 hàng dữ liệu bị loại bỏ** do chứa các giá trị nằm ngoài khoảng cho phép.

### **2.1.2 Chuẩn hóa dữ liệu**

**import pandas as pd**

**from sklearn.preprocessing import MinMaxScaler**

**# Đường dẫn tới tệp dữ liệu gốc**

**file\_path ='D:\\python\\BAITAPDULIEU\\DULIEU\\student-mat-binary.csv'**

**# Tải dữ liệu**

**data = pd.read\_csv(file\_path)**

**# Sao chép dữ liệu để làm sạch**

**data\_clean = data.copy()**

**# Xác định các cột dạng chuỗi**

**string\_columns = data\_clean.select\_dtypes(include=['object']).columns**

**# Bắt đầu làm sạch dữ liệu chuỗi**

**data\_clean[string\_columns] = data\_clean[string\_columns].apply(lambda col: col.str.strip()) # Xóa khoảng trắng ở đầu và cuối**

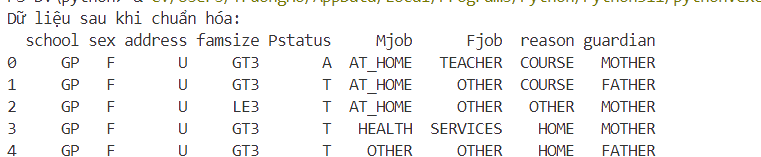
**data\_clean[string\_columns] = data\_clean[string\_columns].apply(lambda col: col.str.replace(r'\s+', ' ', regex=True)) # Thay thế nhiều khoảng trắng bằng một khoảng trắng**

**data\_clean[string\_columns] = data\_clean[string\_columns].apply(lambda col: col.str.replace(r'[^\x00-\x7F]+', '', regex=True)) # Loại bỏ ký tự không thuộc ASCII**

**data\_clean[string\_columns] = data\_clean[string\_columns].apply(lambda col: col.str.replace('#', '', regex=False)) # Loại bỏ ký tự '#'**

**data\_clean[string\_columns] = data\_clean[string\_columns].apply(lambda col: col.str.upper()) # Chuyển thành chữ in hoa**

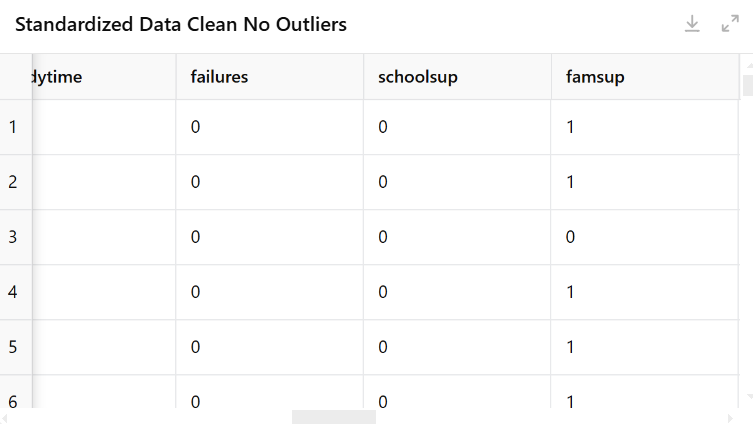
**Kết quả:**

****

Hình : Kết quả sau khi được chuẩn hóa

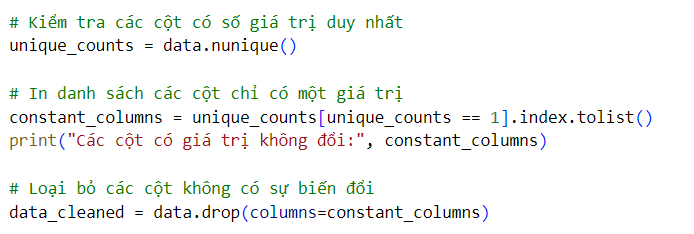
### **2.1.3 Kiểm tra các cột dữ liệu**

Tập dữ liệu sau khi lọc ra các giá trị ngoại lai và chuẩn hóa sẽ như sau:



Hình :Kết quả sau khi lọc các giá trị ngoại lai và chuẩn hóa

Sẽ có các cột chỉ tồn tại một giá trị nên tiến hành kiểm tra và xóa những dữ liệu đó đi vì các cột như failures, schoolsup, nursery, higher, và internet có thể chỉ chứa một giá trị duy nhất sau khi loại bỏ giá trị ngoại lai. Điều này xảy ra khi tất cả các giá trị khác của các cột đó bị loại bỏ, hoặc khi các cột ban đầu đã không có sự biến đổi lớn (nghĩa là cột có tính chất không đổi).

  
Kết quả:



Hình : Kết quả của loại bỏ các giá trị

# CHƯƠNG 3:PHÂN TÍCH DỮ LIỆU

## 3.1 CÁC PHƯƠNG PHÁP SỬ DỤNG

### **3.1.1. Phương pháp phân tích tương quan**

Phân tích tương quan là một phương pháp thống kê được sử dụng để đánh giá mối quan hệ giữa hai hoặc nhiều biến số. Phương pháp này xác định mức độ mà một biến thay đổi theo biến khác, được biểu thị thông qua hệ số tương quan (correlation coefficient). Giá trị hệ số nằm trong khoảng [−1,1]

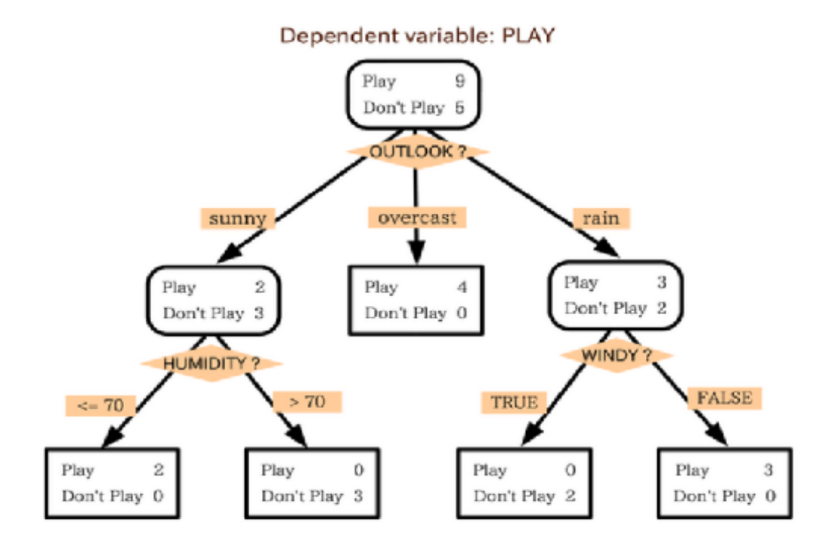
* 1: Tương quan thuận hoàn hảo (biến tăng đồng thời).
* −1: Tương quan nghịch hoàn hảo (một biến tăng, biến kia giảm).
* 0: Không có tương quan (không có mối quan hệ tuyến tính).

### **3.1.2.Phân tích hồi quy logistic**

Hồi quy logistic là một phương pháp thống kê và thuật toán học máy được sử dụng để phân loại nhị phân, hoạt động bằng cách mô hình hóa xác suất của một sự kiện xảy ra (lớp 1) dựa trên các biến đầu vào. Mô hình sử dụng hàm sigmoid để ánh xạ giá trị đầu ra vào khoảng [0,1][0, 1][0,1]. Nếu xác suất lớn hơn ngưỡng (thường là 0.5), mô hình dự đoán sự kiện xảy ra; ngược lại, sự kiện không xảy ra.

### **3.1.3.Phân tích sử dụng Cây quyết định (Decision Tree regression)**

Cây quyết định (Decision Tree): Cây quyết định là một kiểu mô hình dự báo, 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.



Hình : Cây quyết định

### **3.1.4.Phân tích sử dụng Rừng ngẫu nhiên (Random Forest)**

Rừng ngẫu nhiên (Random Forest): Rừng ngẫu nhiên hoặc rừng quyết định ngẫu nhiên là một phương pháp học tập tổng hợp để phân loại, hồi quy và các nhiệm vụ khác hoạt động bằng cách xây dựng vô số cây quyết định tại thời điểm đào tạo. Đối với các nhiệm vụ phân loại, đầu ra của rừng ngẫu nhiên là lớp được hầu hết các cây lựa chọn.

A diagram of fruit and fruit

Description automatically generated

Hình : Rừng ngẫu nhiên

### **3.1.5.Phân tích sử dụng Máy vector hỗ trợ (Support Vector Machine)**

Máy vector hỗ trợ là một thuật toán học máy được sử dụng cho cả phân loại và hồi quy. SVM hoạt động bằng cách tìm kiếm một siêu phẳng (hyperplane) tối ưu để phân chia dữ liệu thành các lớp khác nhau sao cho khoảng cách giữa siêu phẳng và các điểm gần nhất của mỗi lớp (biên cách ly) là lớn nhất. Các điểm dữ liệu gần siêu phẳng nhất được gọi là vector hỗ trợ (support vectors) và đóng vai trò quyết định trong việc định nghĩa siêu phẳng.

* **Phương pháp phân tích tương quan**

Đoạn lệnh để phân tích tương quan giữa các thuộc tính:

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Đọc dữ liệu từ file

data = pd.read\_csv('D:\\python\\BAITAPDULIEU\\DULIEU\\FINAL.csv')

# Tính ma trận tương quan (chỉ các cột số)

correlation\_matrix = data.corr(numeric\_only=True)

# Vẽ biểu đồ heatmap

plt.figure(figsize=(14, 12)) # Kích thước biểu đồ

sns.heatmap(correlation\_matrix,

annot=True, # Hiển thị giá trị trong mỗi ô

cmap='coolwarm', # Màu sắc

fmt=".2f", # Định dạng giá trị

linewidths=0.5, # Độ dày đường phân cách

cbar=True) # Hiển thị thanh màu

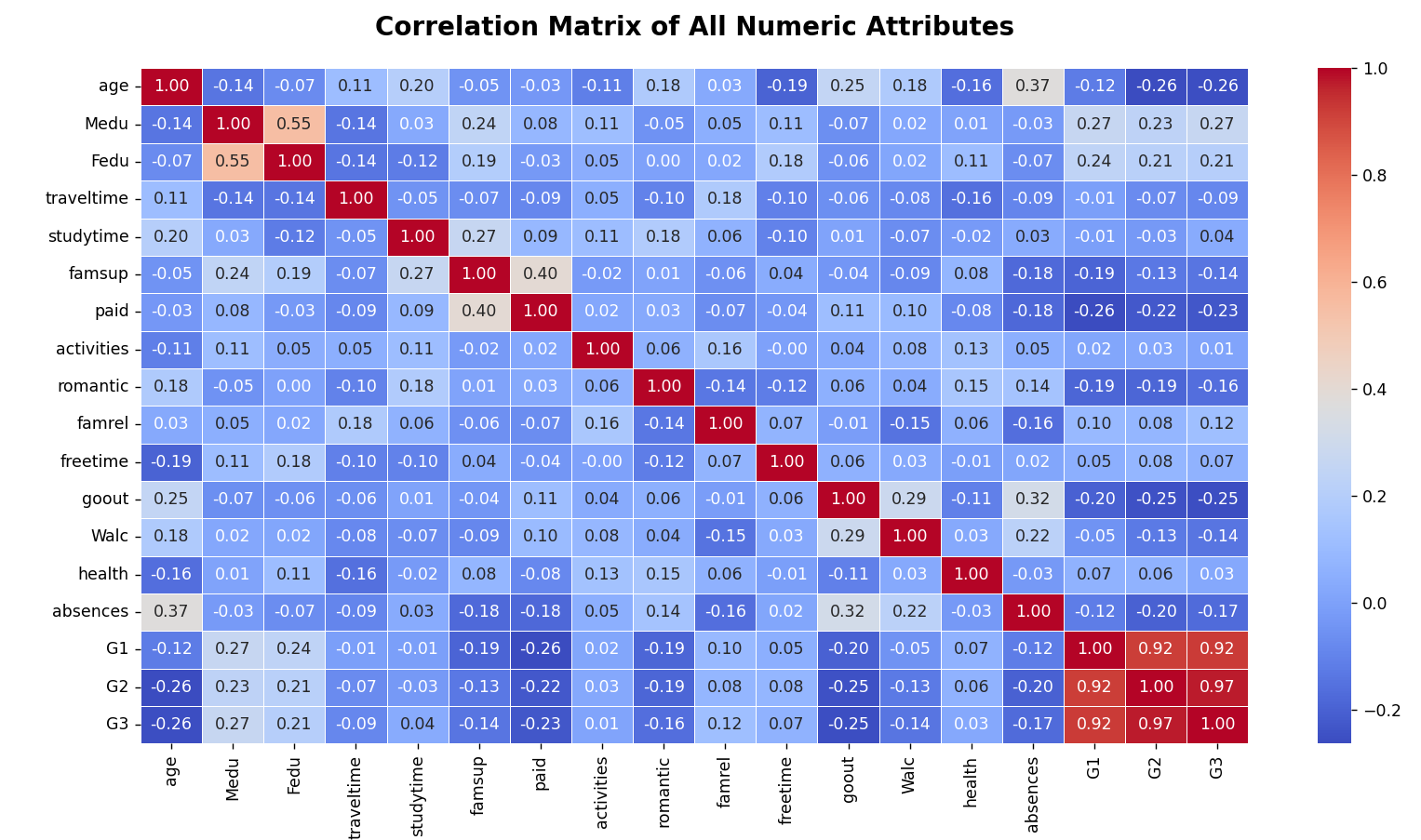
# Tiêu đề biểu đồ

plt.title("Correlation Matrix of All Numeric Attributes", fontsize=16, weight='bold', pad=20)

# Hiển thị biểu đồ

plt.show()

**Kết quả:**

****

Hình :Hình ảnh tương quan của các biến

Biểu đồ trên là heatmap thể hiện mối tương quan giữa các biến số liệu trong dataset.

* Màu sắc:
* Màu đỏ đậm: Tương quan dương mạnh.
* Màu xanh đậm: Tương quan âm mạnh.
* Màu nhạt: Tương quan yếu.

**Phân tích kết quả:**

**1.Biến số có liên quan mạnh đến nhau**

* **G1, G2, G3 (Điểm số học kỳ 1, 2, và cuối cùng):**
  + Tương quan rất mạnh (**0.92-0.97**) giữa các điểm số.
  + Điều này chỉ ra rằng điểm số qua các học kỳ có mối liên hệ chặt chẽ, điểm cao trong học kỳ đầu thường dẫn đến điểm cao ở các học kỳ sau.

**2. Biến Walc (Tiêu thụ rượu cuối tuần):**

* Có tương quan dương với:
  + **goout (Tần suất ra ngoài): 0.29**
    - Sinh viên đi ra ngoài thường xuyên có xu hướng tiêu thụ rượu cuối tuần nhiều hơn.
  + **absences (Số buổi vắng mặt): 0.22**
    - Tiêu thụ rượu có thể liên quan đến việc vắng mặt nhiều hơn trong lớp học.
* Tương quan thấp hoặc không đáng kể với các biến khác như:
  + freetime, health, activities.

**3. Biến số Medu và Fedu (Học vấn của cha mẹ):**

* **Tương quan cao giữa Medu và Fedu: 0.55**
  + Trình độ học vấn của cha mẹ có liên hệ đáng kể, có thể do môi trường sống hoặc sở thích chung.
* Học vấn của cha mẹ không ảnh hưởng lớn đến các biến như tiêu thụ rượu, vắng mặt hay điểm số.

**4. Biến absences (Số buổi vắng mặt):**

* Tương quan dương với:
  + **Walc (Tiêu thụ rượu): 0.22**
    - Sinh viên uống rượu thường có xu hướng nghỉ học nhiều hơn.
  + **goout (Tần suất ra ngoài): 0.32**
    - Việc ra ngoài nhiều có thể liên quan đến số ngày nghỉ học.

**5. Các yếu tố về sức khỏe và lối sống:**

* **health (Sức khỏe tự đánh giá):**
  + Tương quan thấp với hầu hết các biến, bao gồm cả Walc (0.03).
  + Không có mối liên hệ rõ ràng giữa sức khỏe và mức độ tiêu thụ rượu.
* **studytime (Thời gian học):**
  + Không liên quan đáng kể đến Walc hoặc các yếu tố khác.

**Kết luận:**

Các yếu tố như tần suất ra ngoài (goout) và số buổi nghỉ học (absences) có ảnh hưởng đáng kể đến mức độ tiêu thụ rượu cuối tuần (Walc).

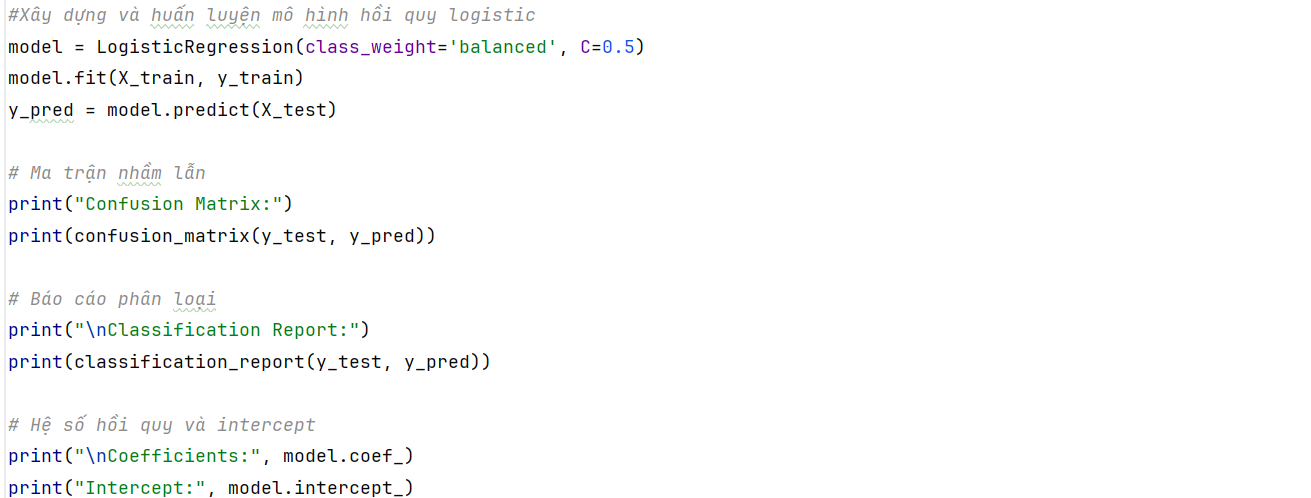
Điểm số qua các kỳ học có mối liên hệ rất mạnh, nhưng ít chịu ảnh hưởng bởi các yếu tố khác trong biểu đồ.

Sức khỏe và thời gian học không phải là yếu tố đáng kể trong việc tiêu thụ rượu hoặc điểm số.

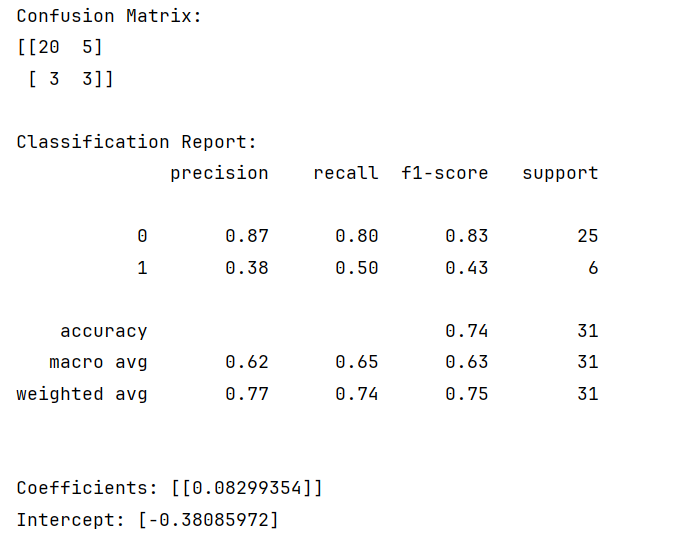
## 3.2 Tìm hiểu mối quan hệ của biến Absences(buổi vắng học) và Walc ( mức độ tiêu thụ rượu vào cuối tuần)

### **3.2.1 Phân tích mô hình hồi quy logistic (Logistic Regression)**

****

****

**Kết quả và giải thích:**

****

Hình :Kết quả của hồi quy Logistic

**1. Ma trận nhầm lẫn (Confusion Matrix):**

[[20 5] # 20 mẫu lớp 0 được dự đoán đúng, 5 mẫu bị dự đoán nhầm thành lớp 1. [ 3 3]] # 3 mẫu lớp 1 được dự đoán đúng, 3 mẫu bị nhầm thành lớp 0.

**2. Báo cáo phân loại (Classification Report):**

* Lớp 0 (Tiêu thụ rượu thấp):
  + Precision: 87% (Mô hình dự đoán đúng 87% các trường hợp là lớp 0).
  + Recall: 80% (Dự đoán đúng 80% các mẫu thực tế thuộc lớp 0).
  + F1-Score: 83% (Cân bằng giữa Precision và Recall).
* Lớp 1 (Tiêu thụ rượu cao):
  + Precision: 38% (Chỉ 38% dự đoán lớp 1 là đúng).
  + Recall: 50% (Dự đoán đúng 50% số mẫu thực tế lớp 1).
  + F1-Score: 43%.
* Độ chính xác tổng thể (Accuracy): 74%.

**3. Hệ số hồi quy (Coefficients & Intercept):**

* Hệ số absences: 0.0833
  + Mỗi ngày nghỉ học làm tăng nhẹ xác suất tiêu thụ rượu bia cao. Tuy nhiên, tác động này không mạnh, cho thấy mối quan hệ giữa số ngày nghỉ và việc tiêu thụ rượu bia là tương đối yếu.
* Intercept (Hằng số): -0.3820
  + Hằng số này cho biết xác suất cơ bản của việc tiêu thụ rượu bia cao khi số ngày nghỉ là 0.

**Mối quan hệ yếu giữa absences và Walc:**

Mô hình hồi quy logistic chỉ ra rằng số ngày nghỉ học (absences) có ảnh hưởng đến xác suất tiêu thụ rượu bia cao, nhưng **mối quan hệ này khá yếu**. Hệ số hồi quy (0.0833) cho thấy mỗi ngày nghỉ học chỉ làm tăng nhẹ xác suất tiêu thụ rượu bia cao, nhưng tác động này không lớn. Điều này có thể chỉ ra rằng số ngày nghỉ học không phải là yếu tố quyết định duy nhất hay mạnh mẽ nhất đối với việc tiêu thụ rượu bia của học sinh.

**Mô hình chưa nhận diện tốt lớp 1 (Tiêu thụ rượu bia cao):**

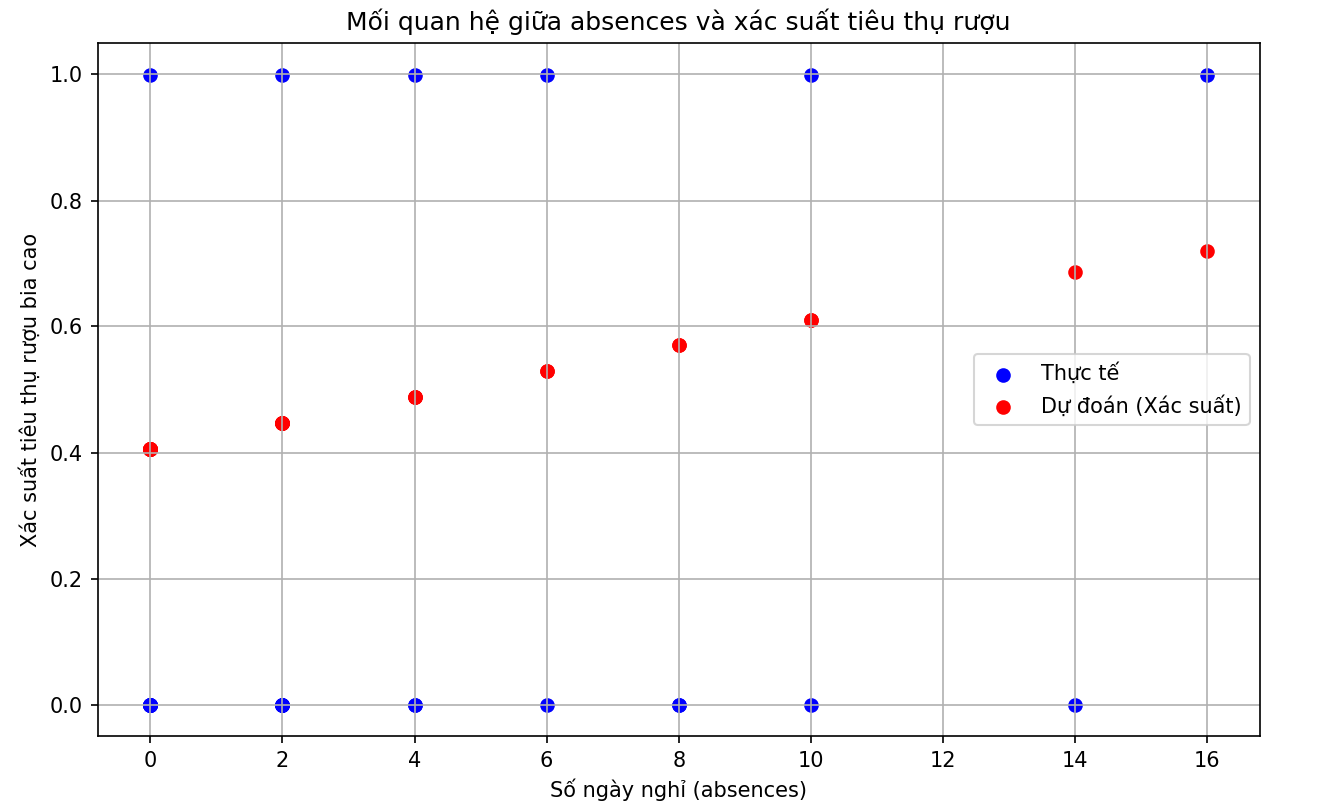
Mô hình không nhận diện tốt lớp thiểu số (lớp 1 - Tiêu thụ rượu bia cao). Chính xác hơn, mô hình chỉ dự đoán đúng 50% mẫu thuộc lớp 1, trong khi Precision cho lớp này chỉ đạt 38%. Điều này cho thấy mô hình gặp khó khăn trong việc phân loại học sinh có mức tiêu thụ rượu bia cao, mặc dù số ngày nghỉ học có thể là yếu tố liên quan.

**Ảnh hưởng yếu của số ngày nghỉ đến tiêu thụ rượu bia:**

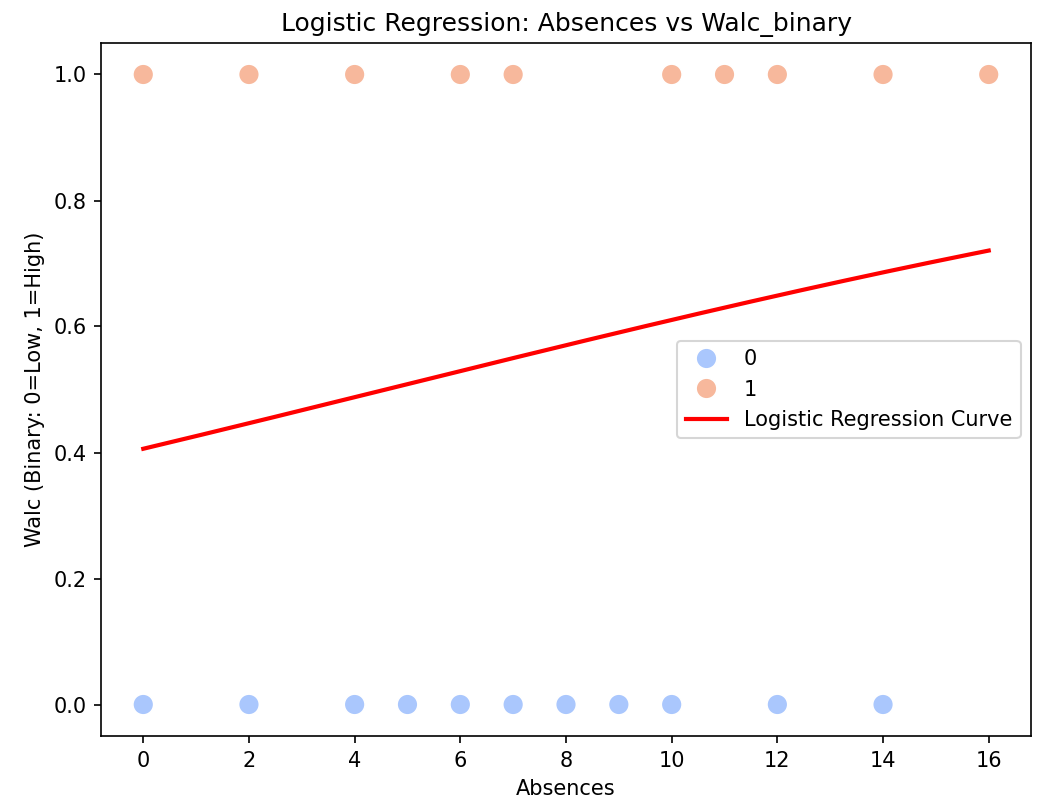
Mặc dù hệ số hồi quy cho thấy có mối quan hệ tích cực giữa số ngày nghỉ và tiêu thụ rượu bia (số ngày nghỉ học càng cao thì xác suất tiêu thụ rượu bia càng cao), mối quan hệ này không mạnh mẽ. Điều này có thể do nhiều yếu tố khác ảnh hưởng đến việc tiêu thụ rượu bia, chẳng hạn như môi trường sống, nhóm bạn, tâm lý cá nhân, hay các yếu tố xã hội và gia đình.

**Tiến hành vẽ sơ đồ**

****

****

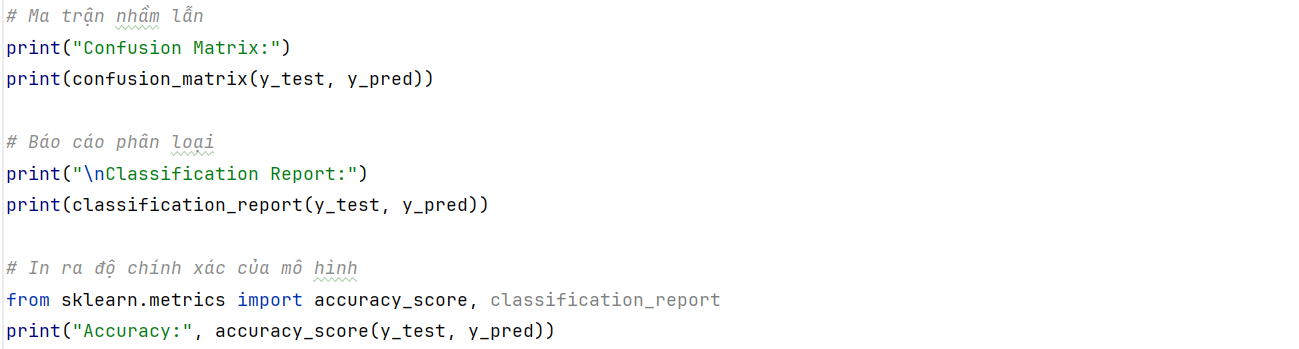
Hình :Biểu đồ mối quan hệ giữa absences và Walc

**

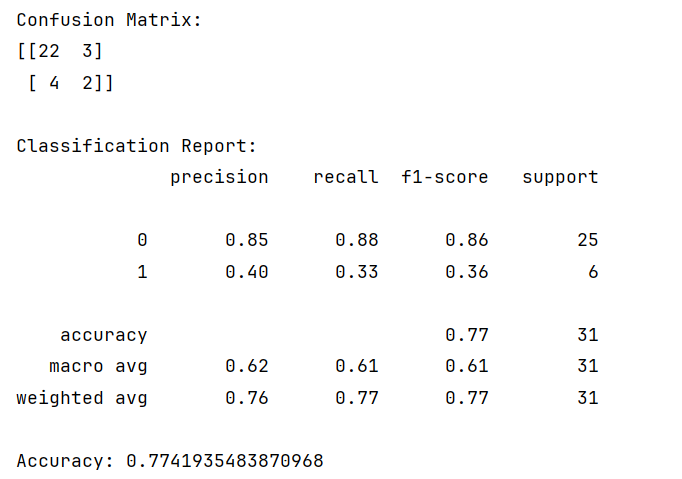
Hình :biểu đồ hồi quy giữa absences và Walc

### **3.2.2.Phân tích sử dụng Cây quyết định (Decision Tree)**





**Kết quả và giải thích:**

****

Hình :Kết quả của Cây quyết định

**1. Ma trận nhầm lẫn (Confusion Matrix):**

[[22 3] #22 mẫu lớp 0 được dự đoán đúng, 3 mẫu bị dự đoán nhầm thành lớp 1. [ 4 2]] #4 mẫu lớp 1 được dự đoán đúng,2 mẫu bị nhầm thành lớp 0.

**2. Báo cáo phân loại (Classification Report):**

* Lớp 0 (Tiêu thụ rượu thấp):
  + Precision: là 0.85, cho thấy mô hình khá chính xác khi dự đoán lớp này.
  + Recall: là 0.88, cho thấy mô hình rất tốt trong việc phát hiện các mẫu thuộc lớp 0.
  + F1-Score:là 0.86, cho thấy mô hình có sự cân bằng tốt giữa Precision và Recall đối với lớp 0.
* Lớp 1 (Tiêu thụ rượu cao):
  + Precision: là 0.40, khá thấp, nghĩa là có một tỷ lệ khá cao các dự đoán "tiêu thụ cao" bị sai (dự đoán là 1, nhưng thực tế là 0).
  + Recall: là 0.33, khá thấp, nghĩa là mô hình bỏ sót nhiều mẫu thực sự thuộc lớp 1.
  + F1-Score: là 0.36, thấp, cho thấy mô hình chưa thể dự đoán chính xác lớp 1 một cách hiệu quả.
* Độ chính xác tổng thể (Accuracy): 77%.

**Mối quan hệ yếu giữa absences và Walc:**

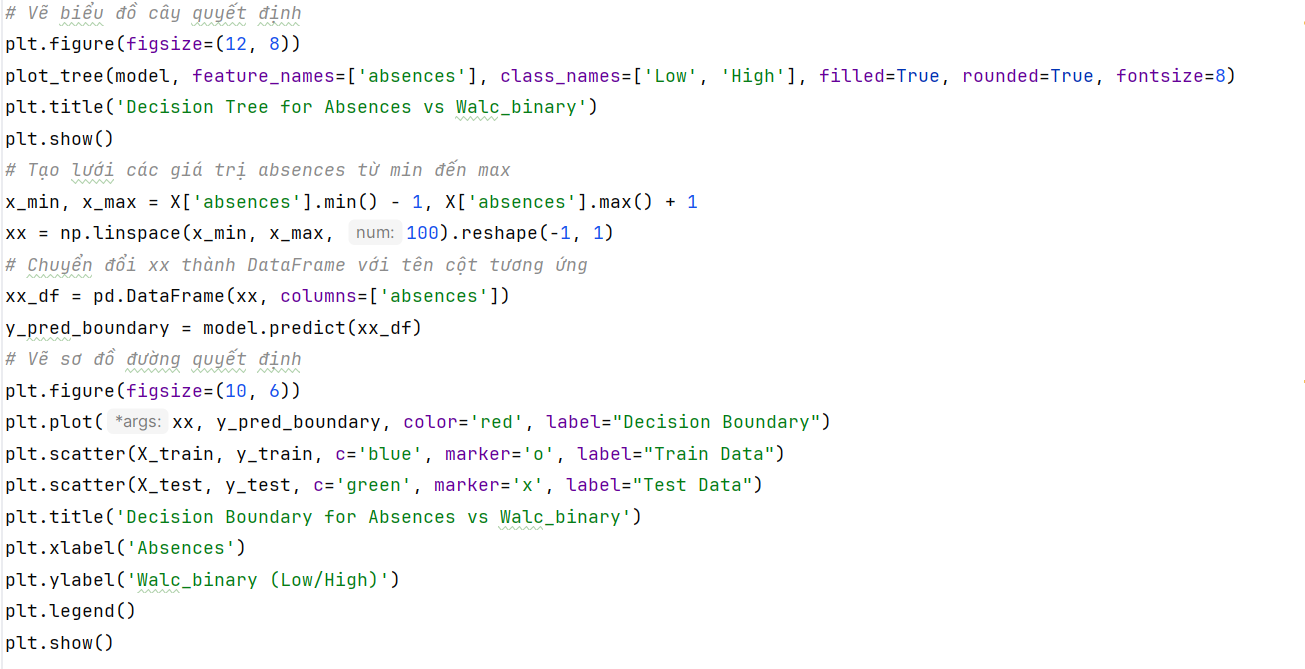
Kết quả của cây quyết định: Cây quyết định cho thấy biến "absences" (số buổi vắng mặt) là một yếu tố quan trọng trong việc phân loại mức tiêu thụ rượu (biến "Walc\_binary"). Các ngưỡng giá trị cụ thể của "absences" (ví dụ: ≤ 4.5, ≤ 10.5) được sử dụng để chia các nhóm, từ đó giúp xác định mức tiêu thụ rượu cao (High) hoặc thấp (Low). Mức tiêu thụ rượu cao (High) thường liên quan đến số buổi vắng mặt cao hơn, trong khi mức thấp (Low) chủ yếu thuộc về những người có số buổi vắng mặt ít.

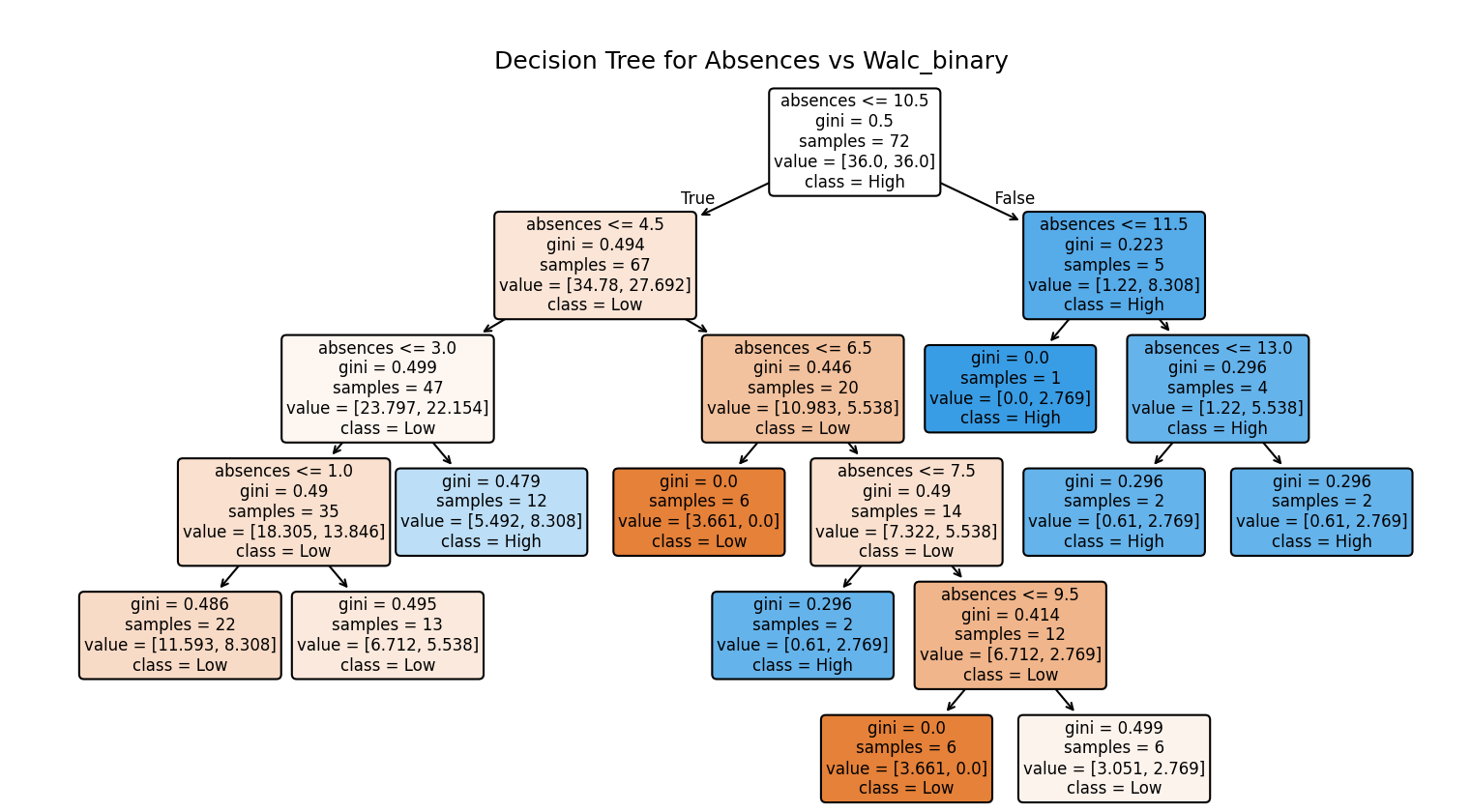
Đường biên quyết định (decision boundary) phản ánh rõ rằng mô hình cây quyết định phân chia rõ ràng hai nhóm Low và High dựa trên các khoảng giá trị của absences. Tuy nhiên, một số điểm dữ liệu (test data) có thể nằm ngoài dự đoán của mô hình, dẫn đến sai số.

**Ảnh hưởng yếu của số ngày nghỉ đến tiêu thụ rượu bia:**

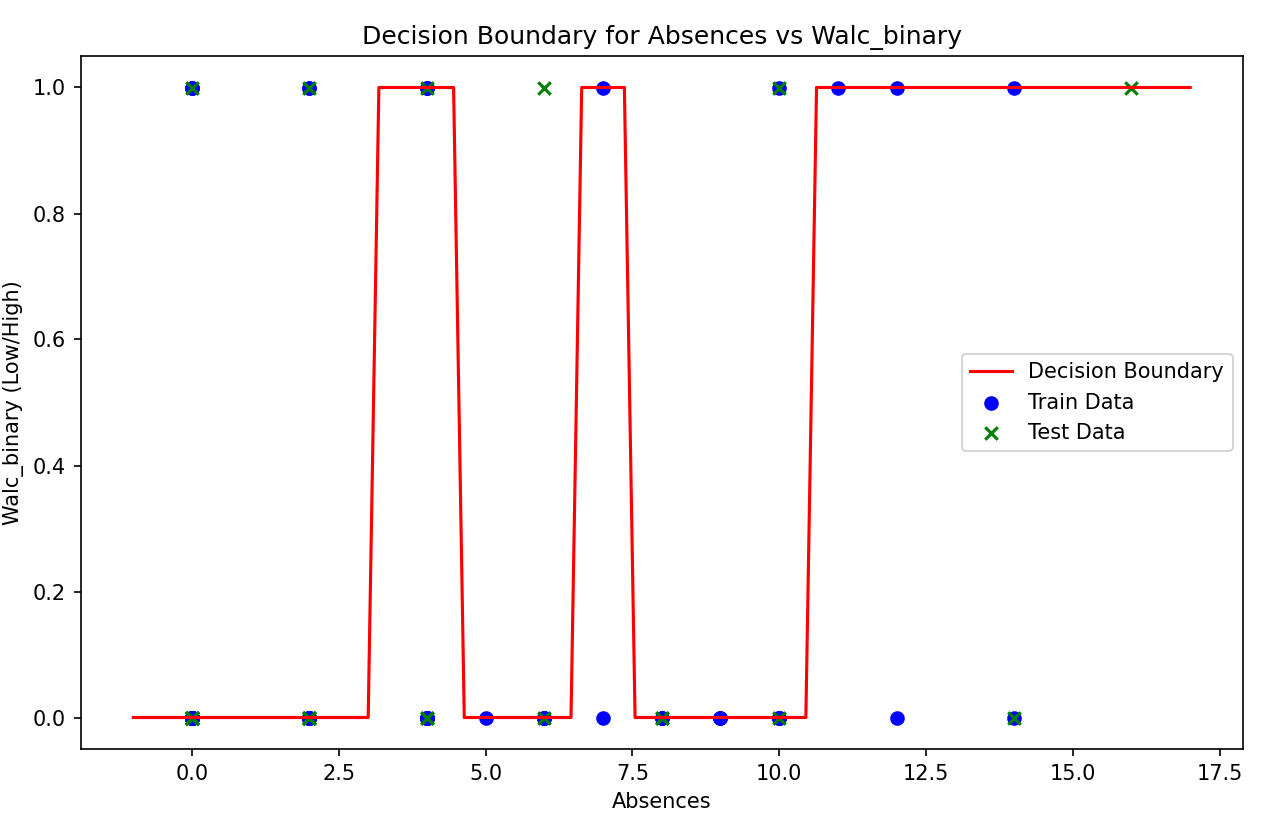
Số buổi vắng mặt (absences) có ảnh hưởng trực tiếp đến mức tiêu thụ rượu (Walc), với xu hướng: vắng mặt nhiều hơn có liên quan đến mức tiêu thụ rượu cao hơn. Tuy nhiên, mối quan hệ này không hoàn toàn mạnh mẽ và cần được kết hợp với các yếu tố khác để đưa ra dự đoán toàn diện hơn.

**Tiến hành vẽ sơ đồ**

****

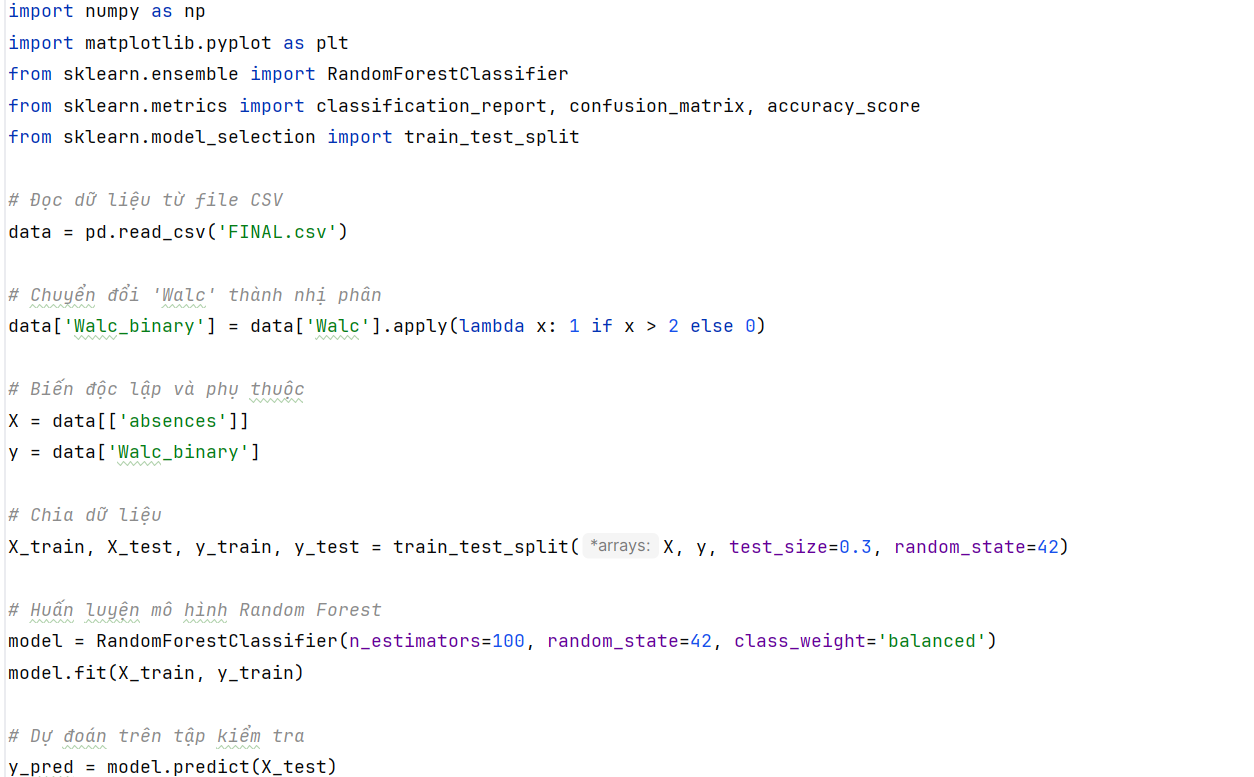


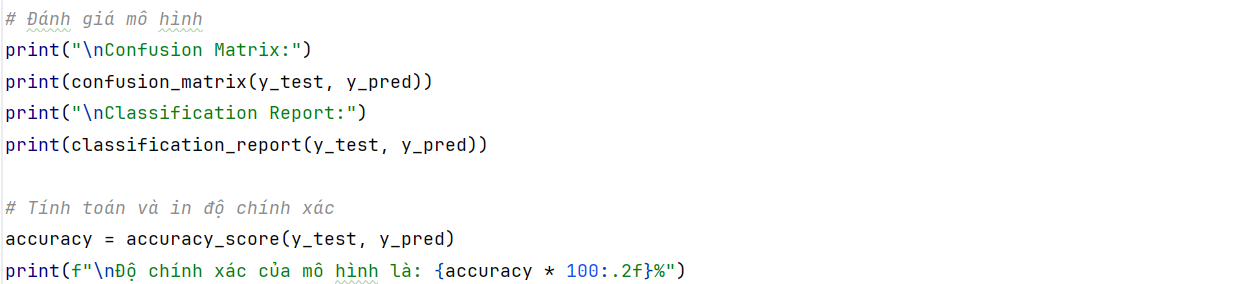
Hình :Cây quyết định giữa absences và Walc



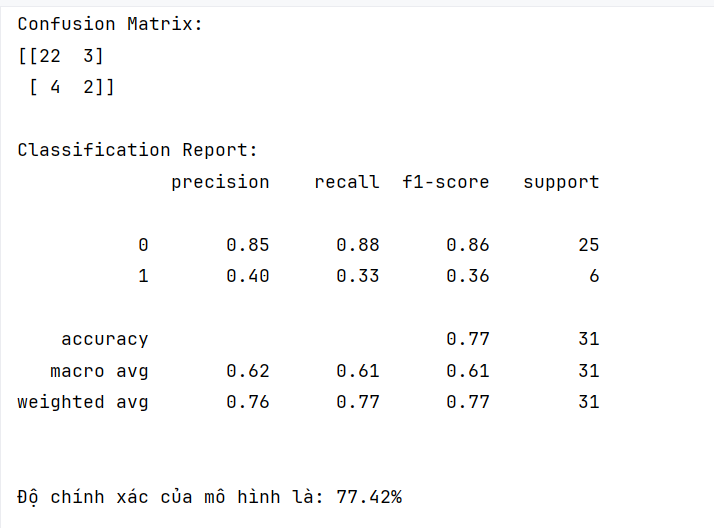
Hình :Sơ đồ đường quyết định cây quyết định

### **3.3.3.Phân tích sử dụng Rừng ngẫu nhiên (Random Forest)**





**Kết quả và giải thích:**

****

Hình :Kết quả của Rừng ngẫu nhiên

**1. Ma trận nhầm lẫn (Confusion Matrix):**

[[22 3] #22 mẫu lớp 0 được dự đoán đúng, 3 mẫu bị dự đoán nhầm thành lớp 1. [ 4 2]] #4 mẫu lớp 1 được dự đoán đúng,2 mẫu bị nhầm thành lớp 0.

**2. Báo cáo phân loại (Classification Report):**

**Lớp 0 (Tiêu thụ rượu thấp):**

* **Precision**: 85% (Dự đoán đúng 85% các trường hợp là lớp 0).
* **Recall**: 88% (Dự đoán đúng 88% số mẫu thực tế thuộc lớp 0).
* **F1-Score**: 86% (Hiệu suất trung bình giữa Precision và Recall).

**Lớp 1 (Tiêu thụ rượu cao):**

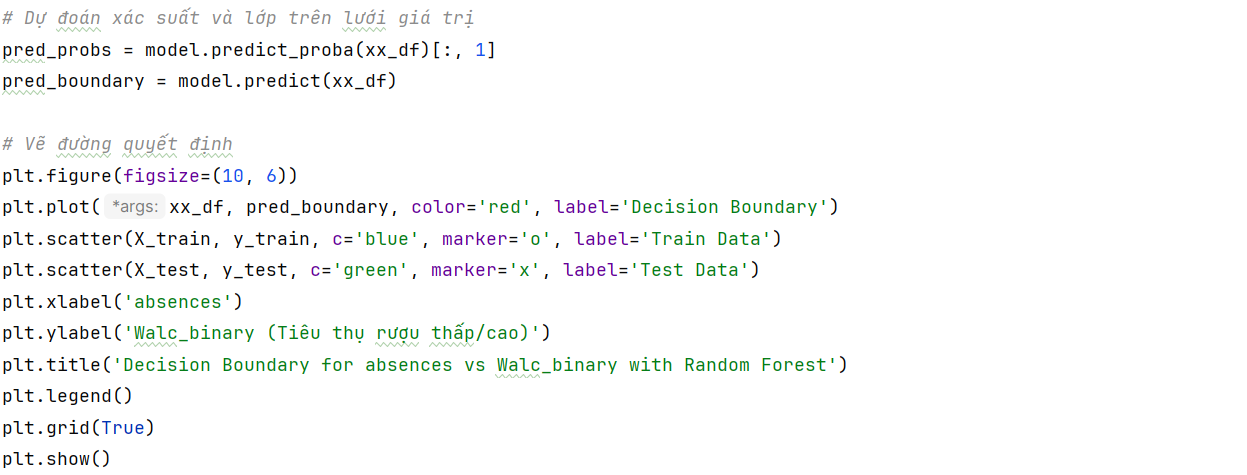
* **Precision**: 40% (Chỉ 40% dự đoán lớp 1 là chính xác).
* **Recall**: 33% (Dự đoán đúng 33% số mẫu thực tế là lớp 1).
* **F1-Score**: 36% (Hiệu suất thấp do Recall và Precision đều thấp).

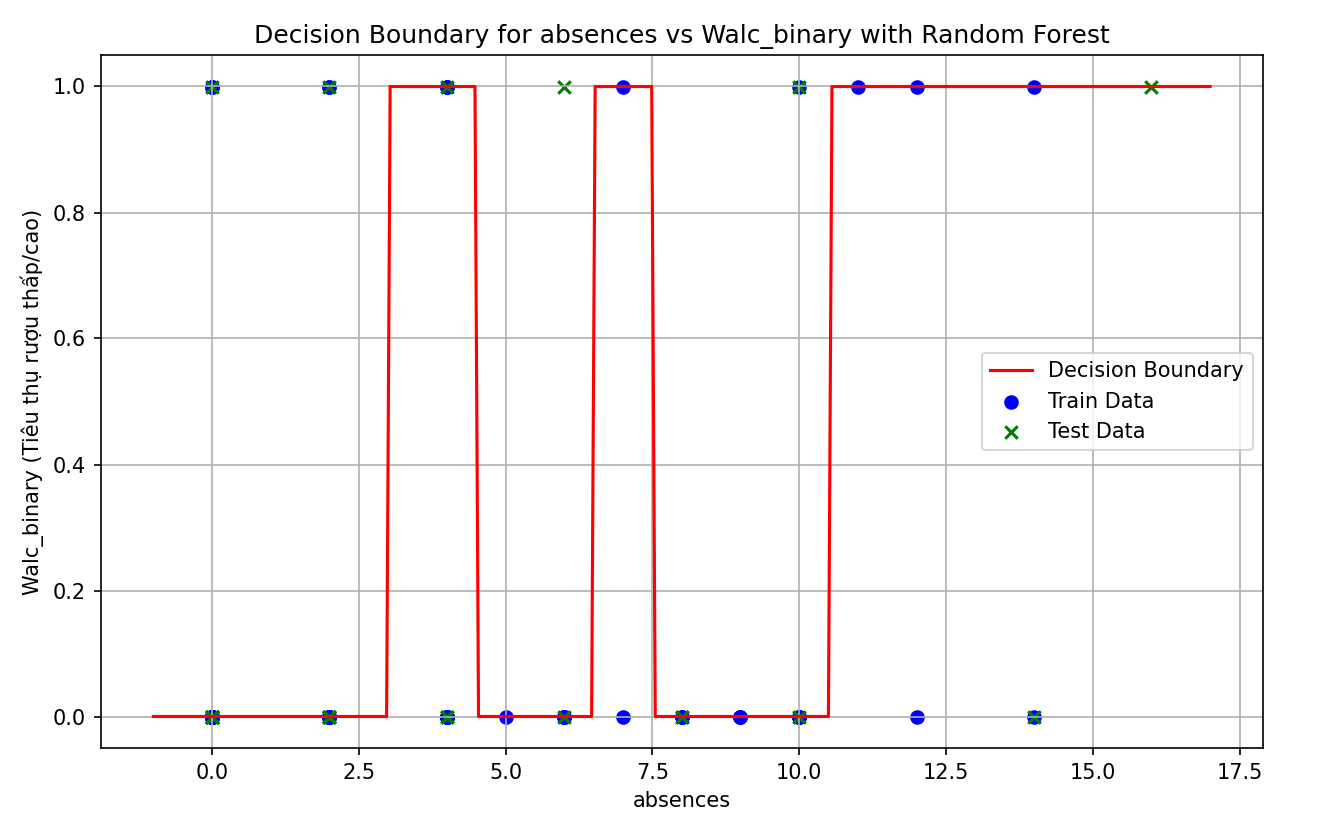
**Độ chính xác tổng thể (Accuracy):** **77,42%**.

**Ảnh hưởng yếu của số ngày nghỉ đến tiêu thụ rượu bia:**

Biến absences dường như có mối quan hệ nhất định với Walc, khi số buổi vắng mặt tăng lên thì khả năng tiêu thụ rượu cũng tăng.

**Tiến hành vẽ sơ đồ**

****



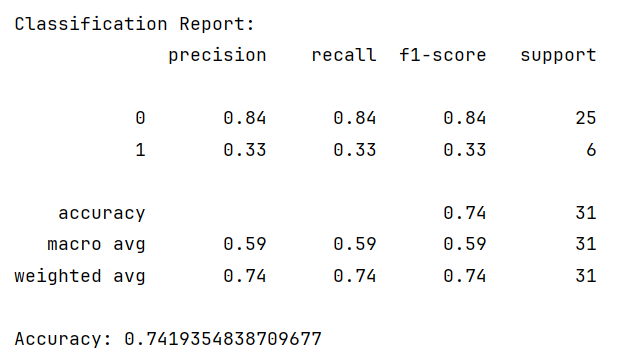
Hình :Sơ đồ đường quyết định rừng ngẫu nhiên

### **3.2.3.Phân tích sử dụng Máy vector hỗ trợ (Support Vector Machine)**





**Kết quả và giải thích:**



Hình :Kết quả của SVM

**1. Ma trận nhầm lẫn (Confusion Matrix):**

[[21 4]

[ 4 2]]

**21**: Số mẫu thuộc lớp "0" (tiêu thụ thấp) được dự đoán đúng.

**4**: Số mẫu thuộc lớp "0" bị dự đoán nhầm thành lớp "1".

**4**: Số mẫu thuộc lớp "1" (tiêu thụ cao) bị dự đoán nhầm thành lớp "0".

**2**: Số mẫu thuộc lớp "1" được dự đoán đúng

**2. Báo cáo phân loại (Classification Report):**

* Lớp 0 (Tiêu thụ rượu thấp):
  + Precision: 84% các dự đoán là chính xác khi mô hình dự đoán tiêu thụ thấp.
  + Recall: Mô hình tìm ra 84% trường hợp thuộc lớp tiêu thụ thấp.
  + F1-Score: Mô hình phát hiện được 84% trường hợp thuộc lớp tiêu thụ thấp.
* Lớp 1 (Tiêu thụ rượu cao):
  + Precision: 33% các dự đoán là chính xác khi mô hình dự đoán tiêu thụ cao.
  + Recall:Mô hình chỉ phát hiện được 33% trường hợp thuộc lớp tiêu thụ cao.
  + F1-Score: là 33%, thấp, cho thấy mô hình chưa thể dự đoán chính xác lớp 1 một cách hiệu quả.
* Độ chính xác tổng thể (Accuracy): 74%.

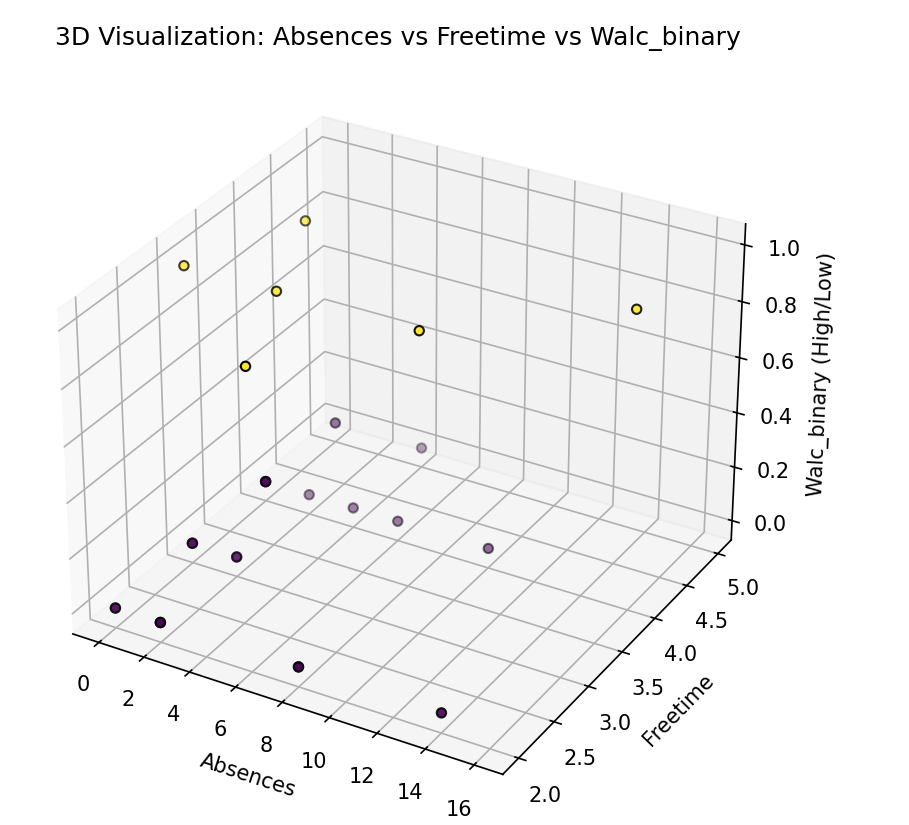
**Mối quan hệ giữa absences và Walc:**

Kết quả từ mô hình SVM cho thấy biến "absences" (số buổi vắng mặt) có ảnh hưởng đáng kể đến mức tiêu thụ rượu vào cuối tuần (biến Walc). Mức tiêu thụ rượu cao thường liên quan đến số buổi vắng mặt nhiều hơn, trong khi mức tiêu thụ thấp thường gắn liền với số buổi vắng mặt ít.

Mô hình SVM có khả năng phân chia rõ ràng hai nhóm tiêu thụ rượu dựa trên số ngày nghỉ. Tuy nhiên, do dữ liệu không cân bằng, mô hình khó nhận diện chính xác nhóm có mức tiêu thụ rượu cao, dẫn đến kết quả F1-score thấp cho nhóm này.

**Ảnh hưởng yếu của số ngày nghỉ đến tiêu thụ rượu bia:**

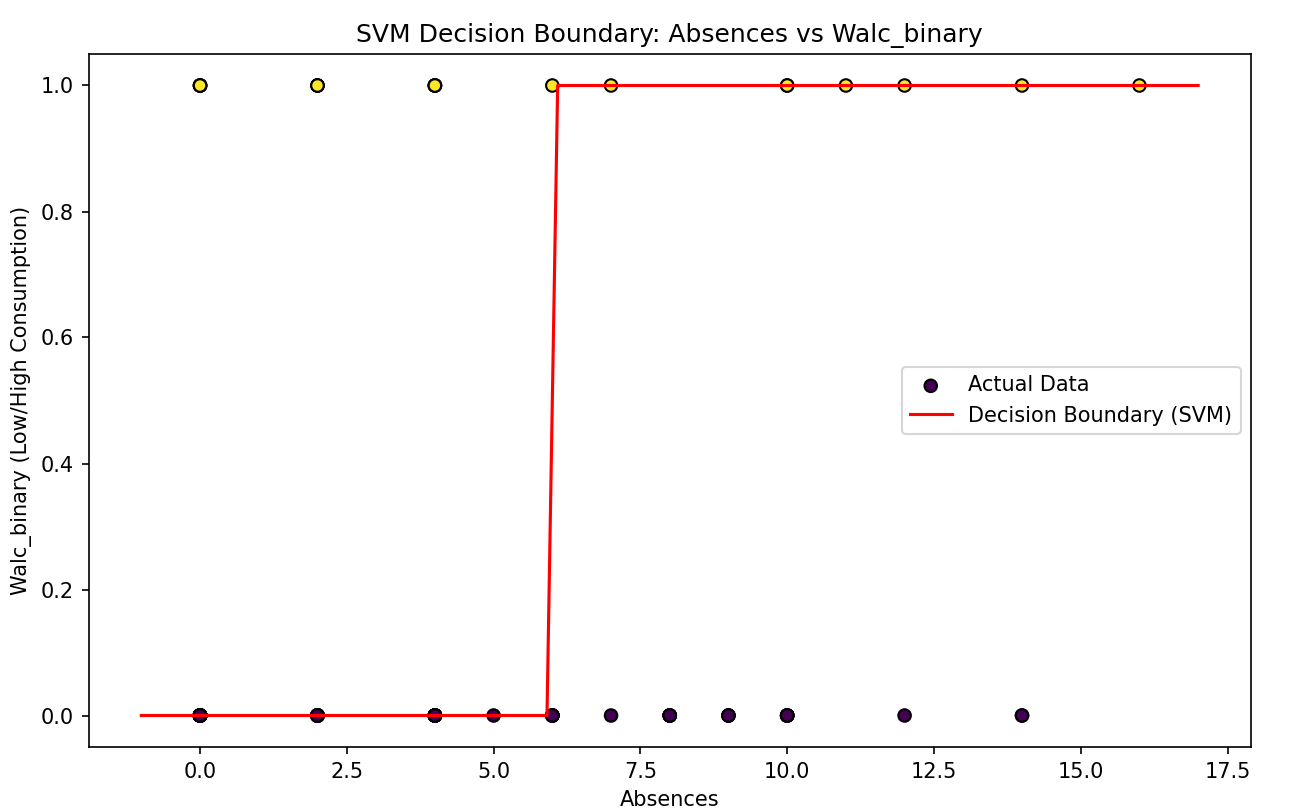
Số buổi vắng mặt có xu hướng tỷ lệ thuận với mức tiêu thụ rượu. Người có số buổi vắng mặt nhiều thường tiêu thụ rượu cao hơn. Điều này gợi ý rằng số ngày nghỉ có thể là một yếu tố dự báo quan trọng về mức độ tiêu thụ rượu vào cuối tuần.



Hình :biểu đồ 3D giữa absences và Walc và freetime

Biểu đồ 3D giữa các biến Absences (số buổi vắng), Walc (tiêu thụ rượu), và Freetime (thời gian rảnh) cho thấy rằng:

Các điểm dữ liệu có giá trị Absences thấp (với giá trị từ 0 đến 10) có xu hướng nằm gần nhau, và hầu hết là thuộc nhóm Walc thấp (màu vàng). Các điểm có giá trị Absences cao (trên 10) có xu hướng thuộc nhóm Walc cao (màu tím). Biểu đồ cũng cho thấy sự phân bổ giữa các nhóm Walc và sự ảnh hưởng của các biến khác như Freetime. Các điểm dữ liệu có thể tạo thành các nhóm riêng biệt dựa trên sự phân bố của các giá trị trong ba chiều này.



Hình :sơ đồ đường quyết định của mô hình SVM

Biểu đồ minh họa đường biên quyết định của mô hình SVM giữa số buổi vắng mặt (Absences) và mức tiêu thụ rượu (Walc\_binary). Dựa trên biểu đồ:

+Đường biên quyết định (Decision Boundary): Được xác định khá rõ ràng, ngăn cách hai nhóm tiêu thụ rượu (thấp và cao) dựa trên số buổi vắng mặt.

+Ngưỡng quyết định: Có vẻ mô hình quyết định phân loại nhóm tiêu thụ rượu cao khi số buổi vắng mặt vượt ngưỡng khoảng 5 buổi. Điều này cho thấy sự gia tăng đáng kể khả năng tiêu thụ rượu khi số ngày nghỉ vượt ngưỡng này.

+Phân bố dữ liệu: Các điểm dữ liệu nằm khá tập trung, nhưng cũng có một số điểm ngoại lệ gần ngưỡng phân loại, điều này có thể ảnh hưởng đến độ chính xác của mô hình

### **3.2.5 So sánh và kết luận giữa 4 phương pháp: Hồi quy Logistic, Cây Quyết Định, Rừng Ngẫu Nhiên và SVM.**

**1. Độ chính xác tổng thể (Accuracy):**

* **Cây quyết định:** 77.42%
* **Hồi quy logistic:** 74.19% (thấp nhất).
* **SVM:** 74.19% (cao nhất, ngang với Hồi quy logistic).
* **Rừng ngẫu nhiên:** 77.42% (cao nhất, ngang với Cây quyết định).

=> Cây quyết định và Rừng ngẫu nhiên có độ chính xác cao nhất (77.42%). Hồi quy Logistic và SVM có độ chính xác thấp hơn, nhưng vẫn tương đối gần nhau ở mức 74.19%.

**2. F1-Score (Trung bình macro):**

F1-Score macro trung bình đo lường sự cân bằng dự đoán giữa các lớp.

* **Cây quyết định:** 0.61
* **Hồi quy logistic:** 0.61
* **SVM:** 0.59
* **Rừng ngẫu nhiên:** 0.61

=> Tất cả các mô hình có F1-Score macro trung bình tương tự nhau (0.61 đối với Cây quyết định, Hồi quy Logistic và Rừng ngẫu nhiên), trong khi SVM có điểm thấp hơn một chút (0.59). Điều này cho thấy các mô hình đều khá cân bằng về khả năng phân loại các lớp, nhưng SVM có phần thua kém một chút về sự cân bằng giữa hai lớp.

**3. Khả năng dự đoán lớp khó (Lớp 1):**

Precision và Recall cho lớp 1 là chỉ số quan trọng để đánh giá khả năng mô hình phát hiện lớp ít phổ biến.

* **Cây quyết định:** Precision = 0.40, Recall = 0.33, F1 = 0.36
* **Hồi quy logistic:** Precision = 0.38, Recall = 0.50, F1 = 0.43
* **SVM:** Precision = 0.33, Recall = 0.33, F1 = 0.33
* **Rừng ngẫu nhiên:** Precision = 0.40, Recall = 0.33, F1 = 0.36

=> Các mô hình SVM và Hồi quy Logistic có kết quả dự đoán lớp 1 kém, với SVM có Precision và Recall thấp nhất. Cây quyết định và Rừng ngẫu nhiên có Precision cao hơn (0.40), nhưng không cải thiện đáng kể về Recall. Do đó, tất cả các mô hình đều cần cải thiện khả năng phân loại lớp ít phổ biến này.

**4. Độ cân bằng tổng thể giữa hai lớp (Macro Precision và Macro Recall):**

* **Cây quyết định:** Precision = 0.62, Recall = 0.61
* **Hồi quy logistic:** Precision = 0.77, Recall = 0.50 (không cân bằng).
* **SVM:** Precision = 0.59, Recall = 0.59
* **Rừng ngẫu nhiên:** Precision = 0.62, Recall = 0.61

=> Cây quyết định và Rừng ngẫu nhiên có độ cân bằng khá tốt giữa Precision và Recall (0.62, 0.61). SVM có Precision và Recall thấp hơn một chút (0.59). Hồi quy Logistic có sự mất cân bằng rõ rệt giữa Precision và Recall (Precision cao, Recall thấp).

**Kết luận:**

Mặc dù các mô hình đều cho thấy một mối quan hệ giữa absences và Walc, Walc dường như không phải là yếu tố quyết định duy nhất gây ra việc vắng mặt. Sự vắng mặt của sinh viên có thể bị ảnh hưởng bởi nhiều yếu tố khác, như tâm lý, động lực học tập, hay các yếu tố bên ngoài khác mà mô hình không được đưa vào phân tích. Walc có thể đóng vai trò là một yếu tố có ảnh hưởng, nhưng absences có thể là một kết quả phức tạp hơn và không chỉ do thói quen uống rượu vào cuối tuần mà còn liên quan đến các yếu tố khác trong môi trường học tập và cuộc sống của sinh viên.

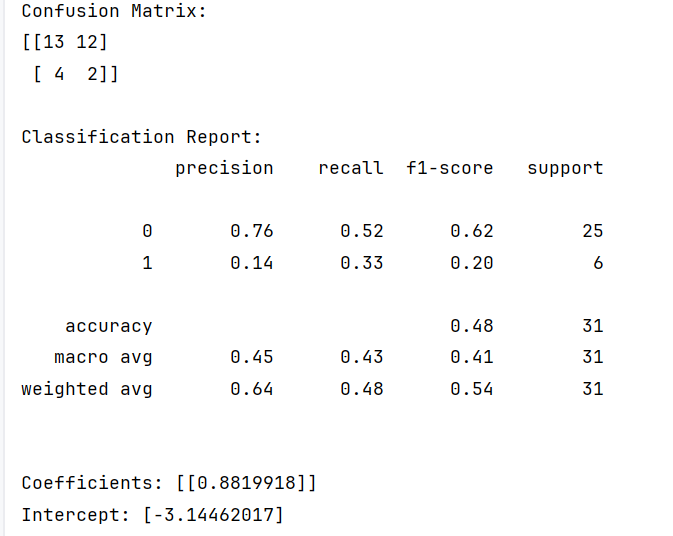
## 3.3 Tìm hiểu mối quan hệ của biến freetime(thời gian rãnh) và Walc ( mức độ tiêu thụ rượu vào cuối tuần)

### **3.3.1 Phân tích mô hình hồi quy logistic (Logistic Regression)**

****

****

**Kết quả và giải thích:**

****

Hình :Kết quả của hồi quy Logistic

**1. Ma trận nhầm lẫn (Confusion Matrix):**

[[13 12] # 13 mẫu lớp 0 (Tiêu thụ rượu bia thấp) được dự đoán đúng, 12 mẫu bị nhầm thành lớp 1.

[ 4 2]] # 2 mẫu lớp 1 (Tiêu thụ rượu bia cao) được dự đoán đúng, 4 mẫu bị nhầm thành lớp 0.

**2. Báo cáo phân loại (Classification Report):**

**Lớp 0 (Tiêu thụ rượu bia thấp):**

* **Precision:** 76% (Mô hình dự đoán đúng 76% các trường hợp là lớp 0).
* **Recall:** 52% (Dự đoán đúng 52% số mẫu thực tế thuộc lớp 0).
* **F1-Score:** 62% (Cân bằng giữa Precision và Recall).

**Lớp 1 (Tiêu thụ rượu bia cao):**

* **Precision:** 14% (Chỉ 14% các dự đoán lớp 1 là chính xác).
* **Recall:** 33% (Dự đoán đúng 33% số mẫu thực tế lớp 1).
* **F1-Score:** 20% (Hiệu suất mô hình thấp đối với lớp 1).

**Độ chính xác tổng thể (Accuracy):** 48%.

**3. Hệ số hồi quy (Coefficients & Intercept):**

* **Hệ số freetime:** 0.882 Thời gian rảnh rỗi cao hơn có xu hướng làm tăng xác suất tiêu thụ rượu bia. Hệ số dương này cho thấy mối quan hệ tích cực giữa thời gian rảnh và tiêu thụ rượu bia, nhưng tác động cụ thể cần được kiểm định thêm.
* **Intercept (Hằng số):** -3.145 Hằng số cho biết xác suất cơ bản của việc tiêu thụ rượu bia cao khi thời gian rảnh là 0.

**Mối quan hệ yếu giữa freetime và Walc:**

Mô hình hồi quy logistic chỉ ra rằng số ngày nghỉ học (absences) có ảnh hưởng đến xác suất tiêu thụ rượu bia cao, nhưng **mối quan hệ này khá yếu**. Hệ số hồi quy (0.0833) cho thấy mỗi ngày nghỉ học chỉ làm tăng nhẹ xác suất tiêu thụ rượu bia cao, nhưng tác động này không lớn. Điều này có thể chỉ ra rằng số ngày nghỉ học không phải là yếu tố quyết định duy nhất hay mạnh mẽ nhất đối với việc tiêu thụ rượu bia của học sinh.

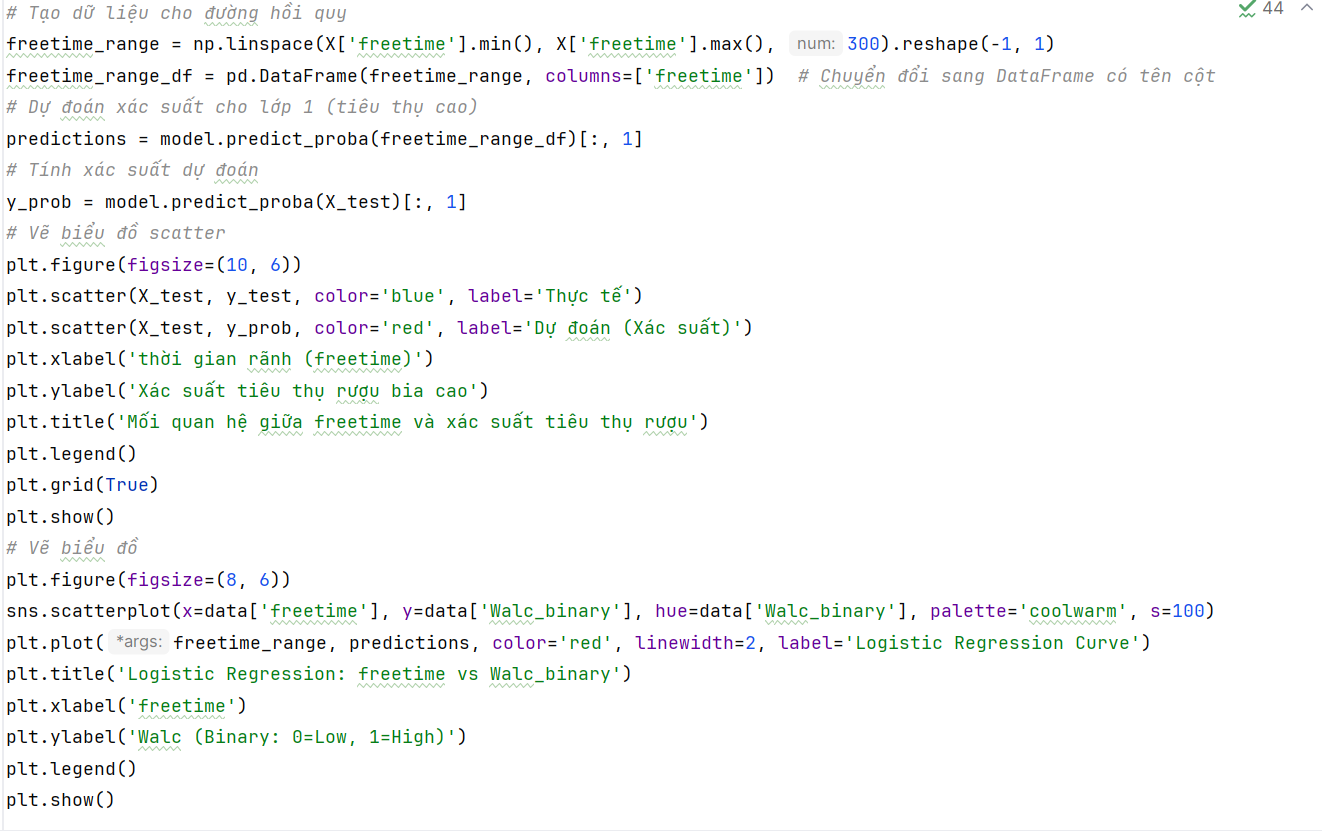
**Mô hình chưa nhận diện tốt lớp 1 (Tiêu thụ rượu bia cao):**

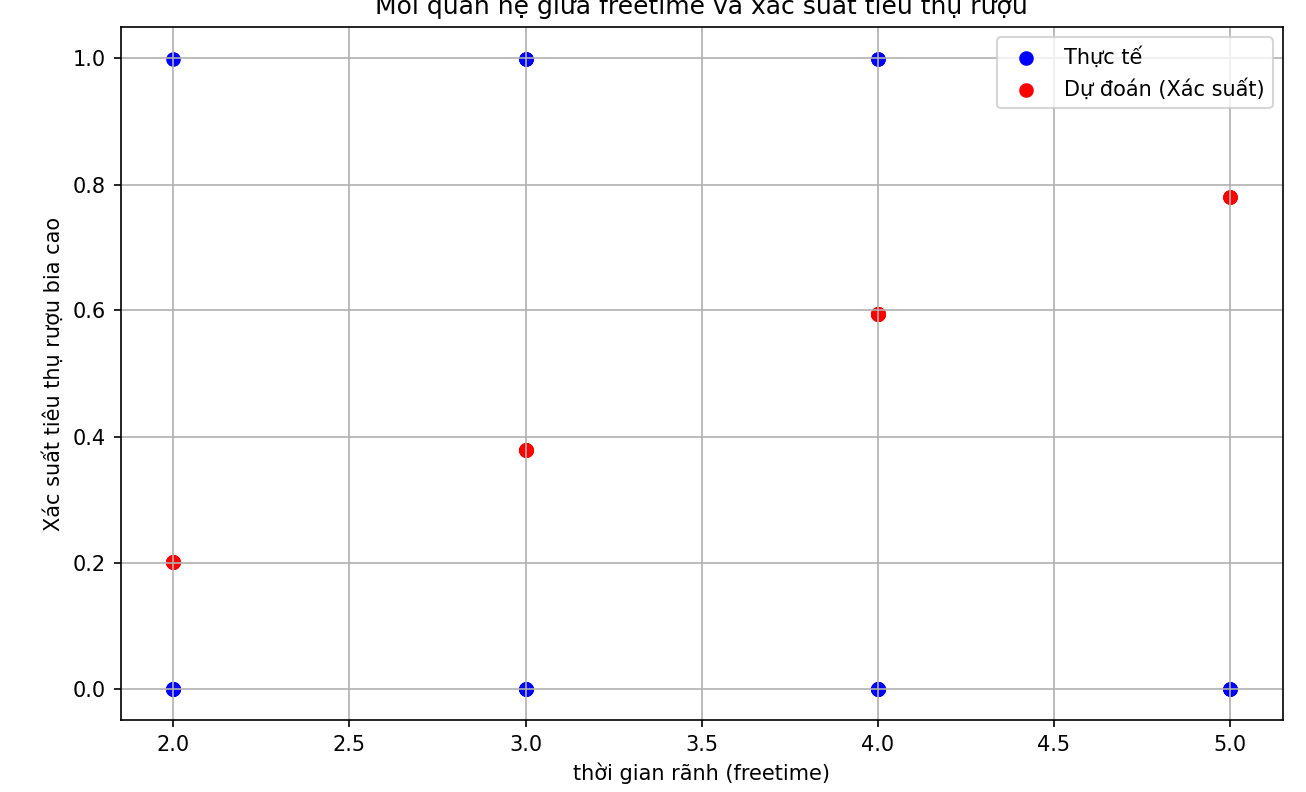
Mô hình chỉ dự đoán đúng 33% số mẫu thuộc lớp 1, và Precision chỉ đạt 14%. Điều này có thể là do mất cân bằng dữ liệu hoặc biến 'freetime' chưa đủ mạnh để phân biệt rõ ràng giữa hai lớp.

**Ảnh hưởng yếu của số ngày nghỉ đến tiêu thụ rượu bia:**

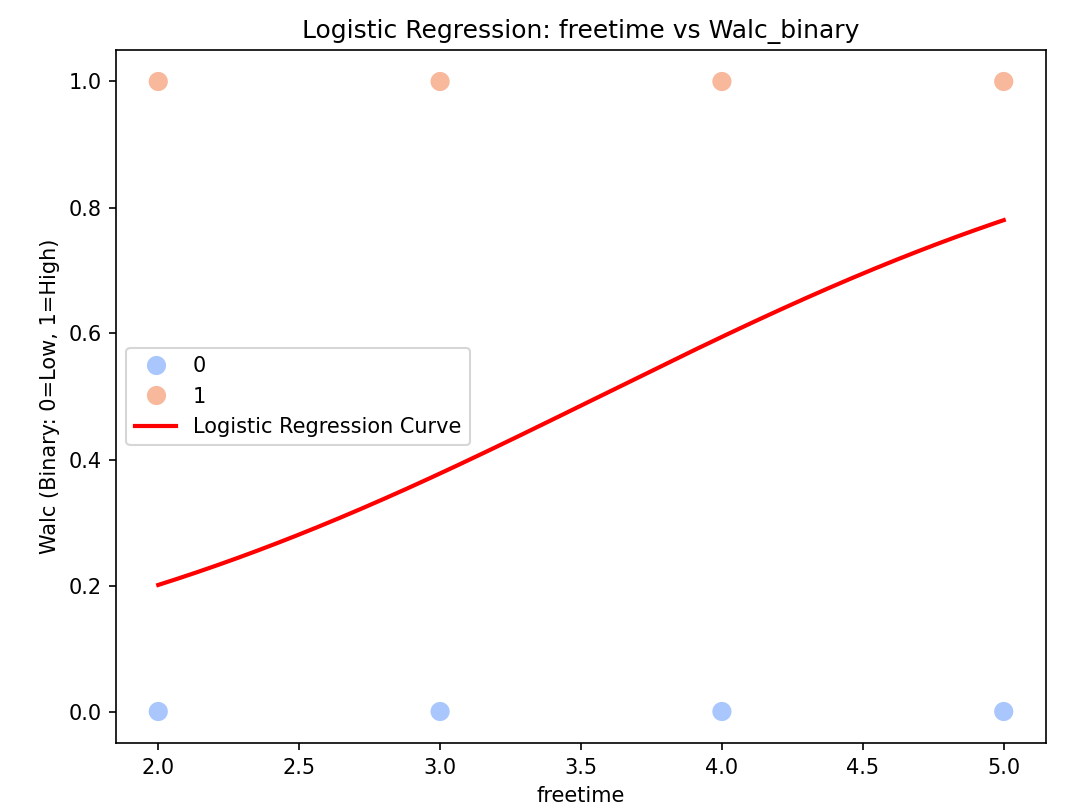
Mặc dù thời gian rảnh rỗi có ảnh hưởng tích cực đến việc tiêu thụ rượu bia, tác động này chưa mạnh mẽ. Điều này cho thấy cần xem xét thêm các yếu tố khác như môi trường sống, nhóm bạn, hay tâm lý cá nhân để cải thiện khả năng dự đoán.

**Tiến hành vẽ sơ đồ**

****

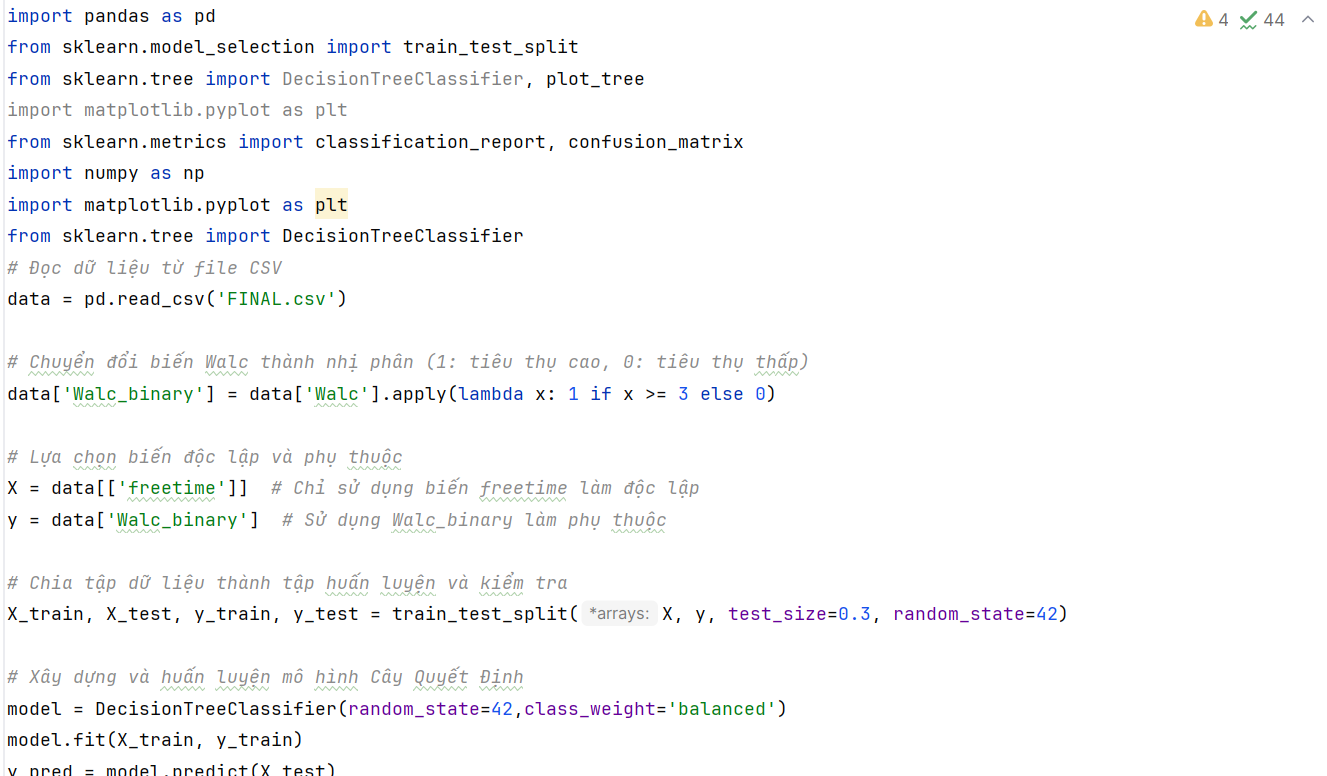
****

Hình :Biểu đồ mối quan hệ giữa freetime và Walc

**

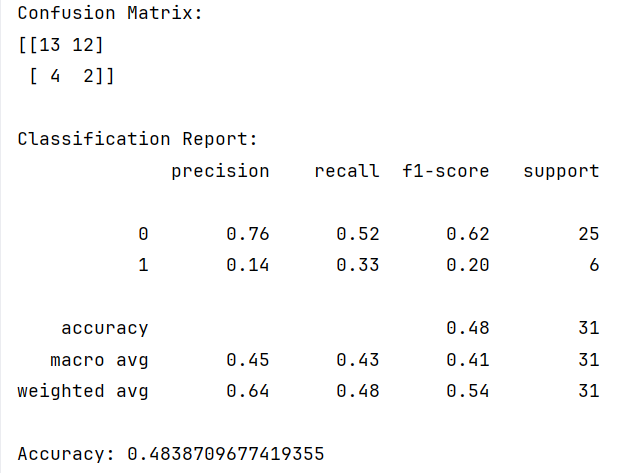
Hình :biểu đồ hồi quy giữa freetime và Walc

### **3.3.2.Phân tích sử dụng Cây quyết định (Decision Tree)**





**Kết quả và giải thích:**

****

Hình :Kết quả của Cây quyết định

**1. Ma trận nhầm lẫn (Confusion Matrix):**

[[13 12] # 13 mẫu lớp 0 (Tiêu thụ rượu thấp) được dự đoán đúng, 12 mẫu bị nhầm thành lớp 1.

[ 4 2]] # 2 mẫu lớp 1 (Tiêu thụ rượu cao) được dự đoán đúng, 4 mẫu bị nhầm thành lớp 0.

**2. Báo cáo phân loại (Classification Report):**

**Lớp 0 (Tiêu thụ rượu bia thấp):**

* **Precision:** 76% (Mô hình dự đoán đúng 76% các trường hợp là lớp 0).
* **Recall:** 52% (Dự đoán đúng 52% số mẫu thực tế thuộc lớp 0).
* **F1-Score:** 62% (Hiệu suất trung bình giữa Precision và Recall).

**Lớp 1 (Tiêu thụ rượu bia cao):**

* **Precision:** 14% (Chỉ 14% dự đoán lớp 1 là chính xác).
* **Recall:** 33% (Dự đoán đúng 33% số mẫu thực tế lớp 1).
* **F1-Score:** 20% (Hiệu suất thấp do Recall và Precision đều thấp).

**Độ chính xác tổng thể (Accuracy):** 48.39%.

**Mối quan hệ yếu giữa freetime và Walc:**

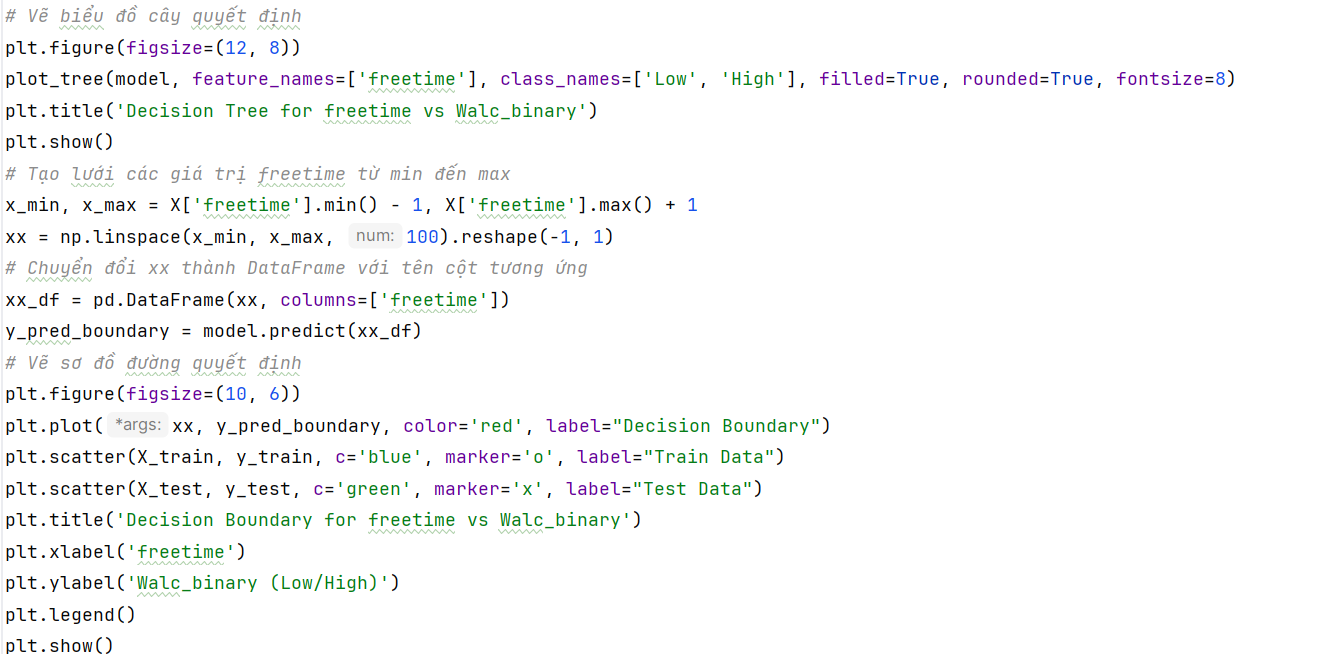
Kết quả của cây quyết định: Cây quyết định cho thấy biến "freetime" (số thời gian rãnh) là một yếu tố quan trọng trong việc phân loại mức tiêu thụ rượu (biến "Walc\_binary"). Các ngưỡng giá trị cụ thể của "absences" (ví dụ: ≤ 2.5, ≤4.5) được sử dụng để chia các nhóm, từ đó giúp xác định mức tiêu thụ rượu cao (High) hoặc thấp (Low). Mức tiêu thụ rượu cao (High) thường liên quan đến số buổi thời gian rãnh cao hơn, trong khi mức thấp (Low) chủ yếu thuộc về những người có mặt thời gian rãnh ít.

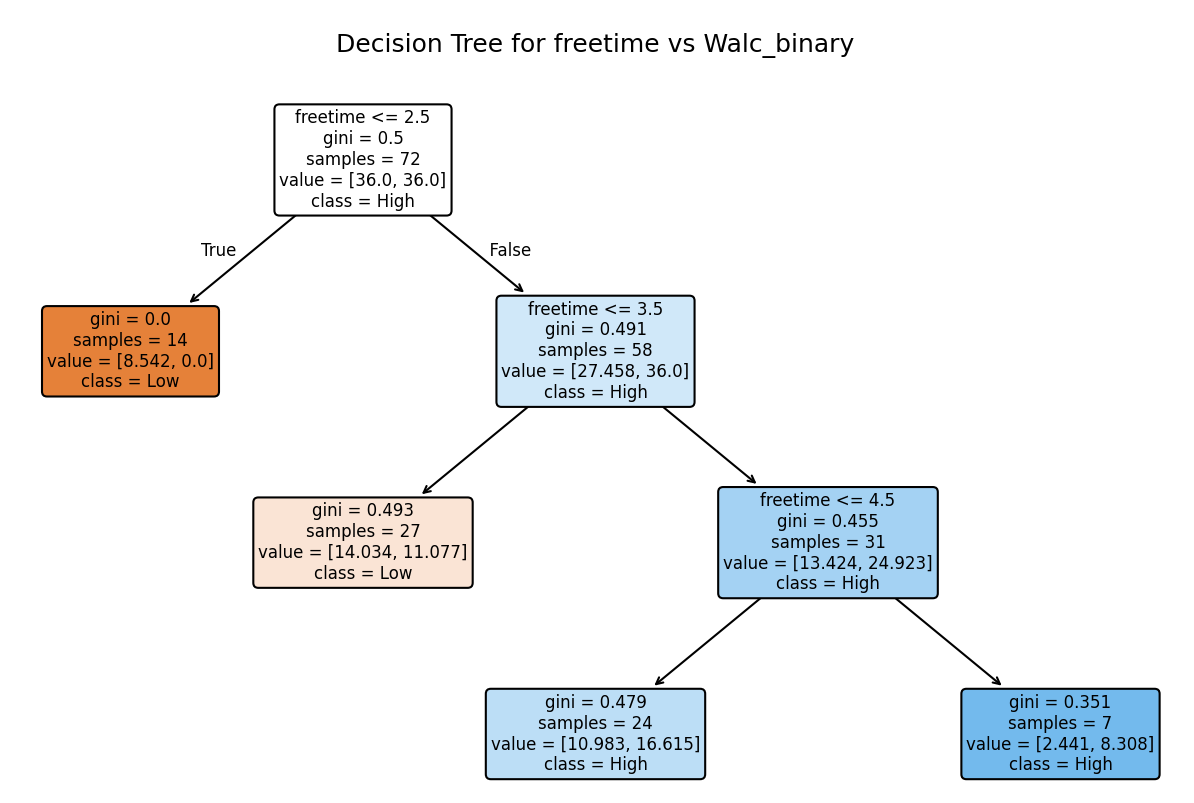
Đường biên quyết định (decision boundary) phản ánh rõ rằng mô hình cây quyết định phân chia rõ ràng hai nhóm Low và High dựa trên các khoảng giá trị của freetime. Tuy nhiên, một số điểm dữ liệu (test data) có thể nằm ngoài dự đoán của mô hình, dẫn đến sai số.

**Ảnh hưởng yếu của số ngày nghỉ đến tiêu thụ rượu bia:**

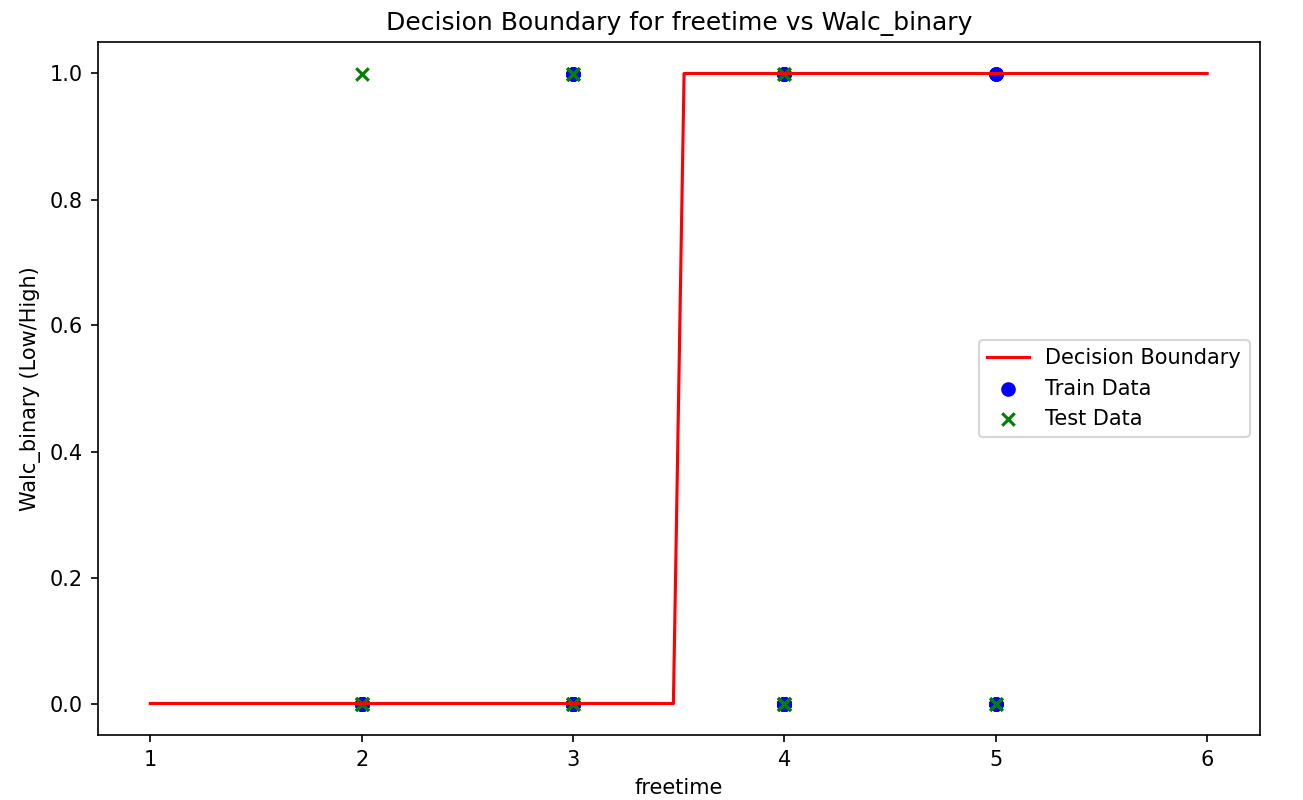
Số buổi rãnh (freetime) có ảnh hưởng trực tiếp đến mức tiêu thụ rượu (Walc), với xu hướng: vắng mặt nhiều hơn có liên quan đến mức tiêu thụ rượu cao hơn. Tuy nhiên, mối quan hệ này không hoàn toàn mạnh mẽ và cần được kết hợp với các yếu tố khác để đưa ra dự đoán toàn diện hơn.

**Tiến hành vẽ sơ đồ**

****



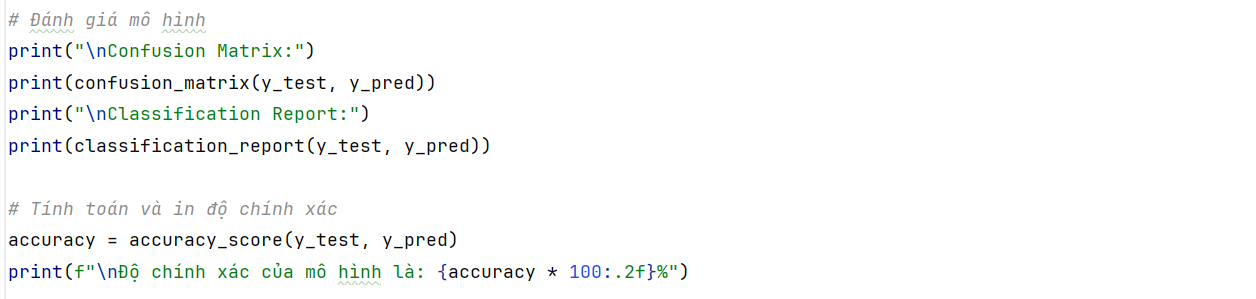
Hình :Cây quyết định giữa freetime và Walc



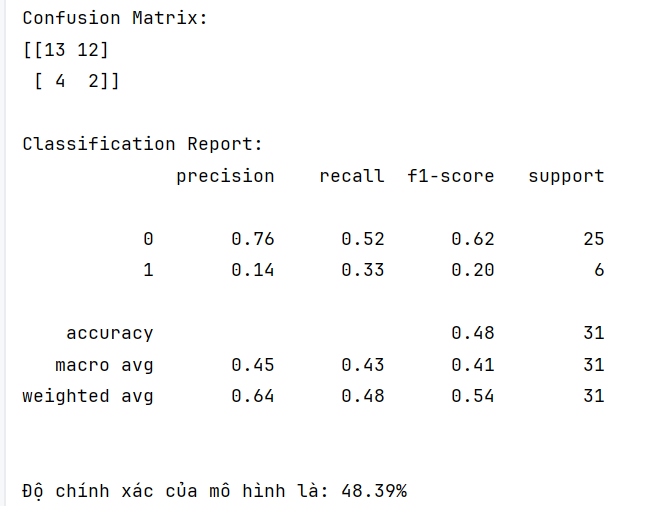
Hình :Sơ đồ đường quyết định cây quyết định

### **3.3.3.Phân tích sử dụng Rừng ngẫu nhiên (Random Forest)**





**Kết quả và giải thích:**

****

Hình :Kết quả của Rừng ngẫu nhiên

**1. Ma trận nhầm lẫn (Confusion Matrix):**

[[13 12] # 13 mẫu lớp 0 (Tiêu thụ rượu thấp) được dự đoán đúng, 12 mẫu bị nhầm thành lớp 1.

[ 4 2]] # 2 mẫu lớp 1 (Tiêu thụ rượu cao) được dự đoán đúng, 4 mẫu bị nhầm thành lớp 0.

**2. Báo cáo phân loại (Classification Report):**

**Lớp 0 (Tiêu thụ rượu thấp):**

* **Precision:** 76% (76% các dự đoán lớp 0 là chính xác).
* **Recall:** 52% (52% mẫu thực tế lớp 0 được dự đoán đúng).
* **F1-Score:** 62% (Hiệu suất trung bình giữa Precision và Recall).

**Lớp 1 (Tiêu thụ rượu cao):**

* **Precision:** 14% (Chỉ 14% dự đoán lớp 1 là chính xác).
* **Recall:** 33% (33% mẫu thực tế lớp 1 được dự đoán đúng).
* **F1-Score:** 20% (Hiệu suất thấp do Precision và Recall đều thấp).

**Độ chính xác tổng thể (Accuracy):** **48.39%**.

**Mối quan hệ yếu giữa freetime và Walc:**

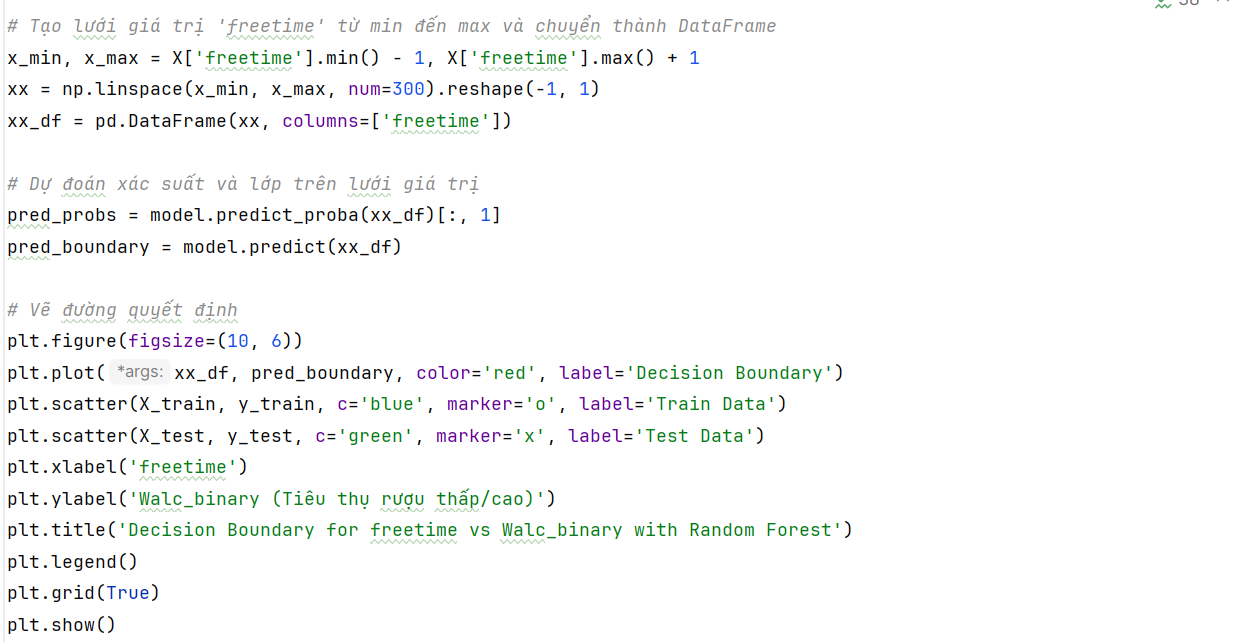
Tác động của freetime: Mô hình cho thấy mối liên hệ giữa thời gian rảnh (freetime) và mức tiêu thụ rượu. Những người có nhiều thời gian rảnh hơn có xu hướng tiêu thụ rượu cao hơn. Tuy nhiên, mức độ tác động yếu, dẫn đến khả năng phân loại không cao, đặc biệt với lớp tiêu thụ rượu cao.

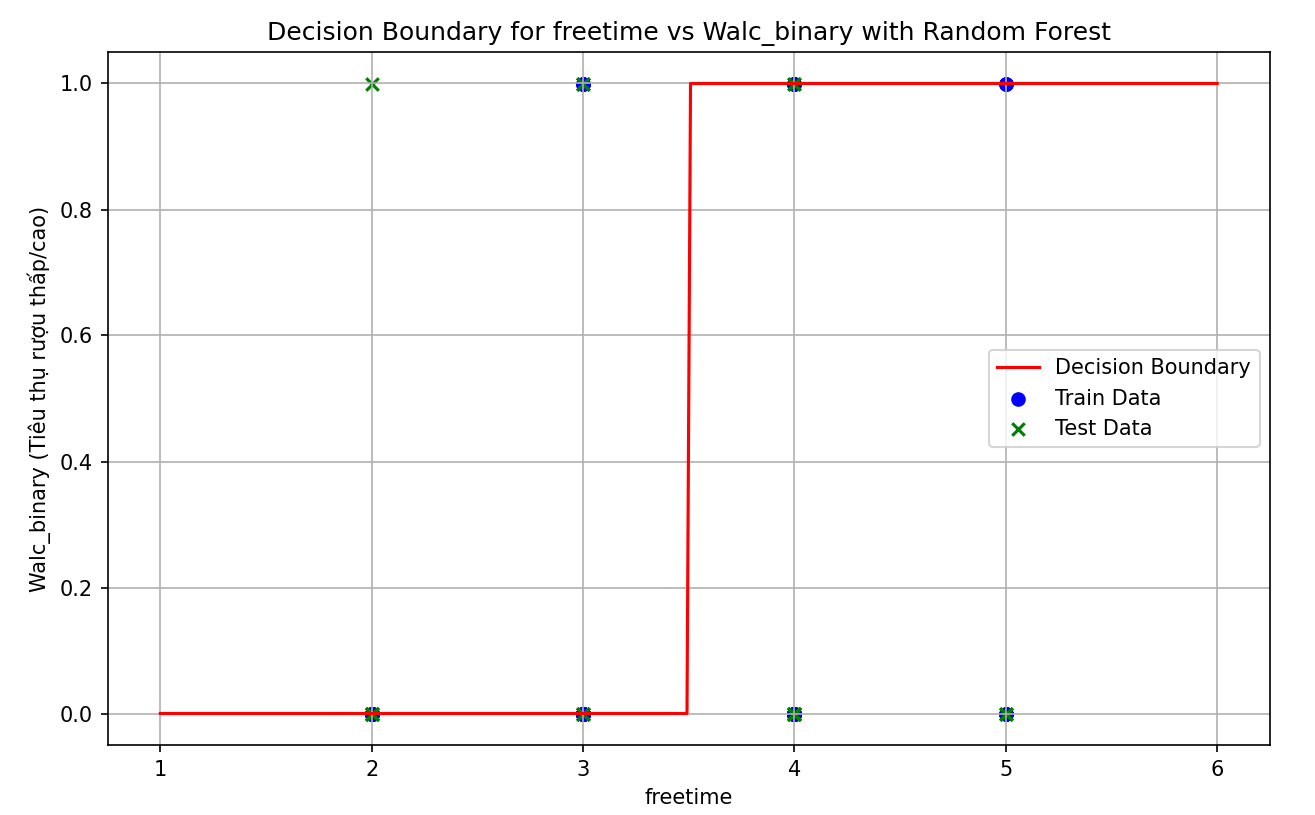
Đường biên quyết định (Decision Boundary): Đường biên quyết định giữa hai nhóm (Low và High) được xác định rõ dựa trên giá trị freetime. Một số điểm kiểm tra nằm ngoài vùng dự đoán của mô hình, dẫn đến sai số cao.

**Ảnh hưởng yếu của số ngày nghỉ đến tiêu thụ rượu bia:**

Mặc dù freetime có ảnh hưởng đến mức tiêu thụ rượu, mối quan hệ không đủ mạnh để dự đoán chính xác. Cần xem xét thêm các yếu tố khác (như age, absences, hoặc goout) để cải thiện hiệu suất mô hình. Hiệu suất hiện tại cho thấy mô hình chỉ phù hợp để đưa ra các dự đoán sơ bộ, chưa đạt độ tin cậy cao.

**Tiến hành vẽ sơ đồ**

****



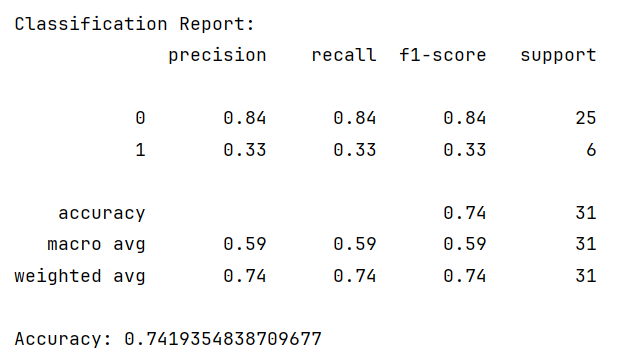
Hình :Sơ đồ đường quyết định rừng ngẫu nhiên

### **3.3.4.Phân tích sử dụng Máy vector hỗ trợ (Support Vector Machine)**





**Kết quả và giải thích:**



Hình :Kết quả của SVM

**1. Ma trận nhầm lẫn (Confusion Matrix):**

[[13 12] # 13 mẫu lớp 0 (Tiêu thụ rượu thấp) được dự đoán đúng, 12 mẫu bị nhầm thành lớp 1.

[ 4 2]] # 2 mẫu lớp 1 (Tiêu thụ rượu cao) được dự đoán đúng, 4 mẫu bị nhầm thành lớp 0.

**2. Báo cáo phân loại (Classification Report):**

**Lớp 0 (Tiêu thụ rượu bia thấp):**

* **Precision:** 76% (Mô hình dự đoán đúng 76% các trường hợp là lớp 0).
* **Recall:** 52% (Dự đoán đúng 52% số mẫu thực tế thuộc lớp 0).
* **F1-Score:** 62% (Cân bằng giữa Precision và Recall).

**Lớp 1 (Tiêu thụ rượu bia cao):**

* **Precision:** 14% (Chỉ 14% dự đoán lớp 1 là chính xác).
* **Recall:** 33% (Dự đoán đúng 33% số mẫu thực tế lớp 1).
* **F1-Score:** 20% (Hiệu suất kém do Precision và Recall thấp).

**Độ chính xác tổng thể (Accuracy):** 48.39%.

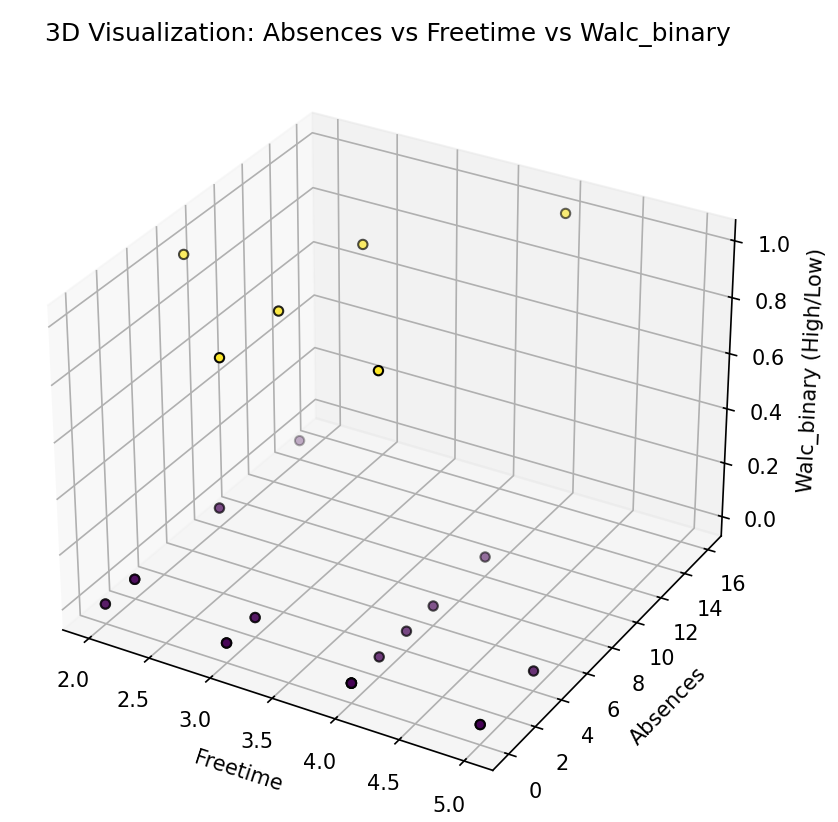
**Mối quan hệ giữa freetime và Walc:**

Kết quả từ mô hình SVM cho thấy biến **'freetime'** (thời gian rảnh rỗi) có ảnh hưởng đến mức tiêu thụ rượu vào cuối tuần (**'Walc'**). Mức tiêu thụ rượu cao có xu hướng xuất hiện nhiều hơn ở những người có thời gian rảnh rỗi cao, trong khi mức tiêu thụ thấp thường gắn liền với thời gian rảnh rỗi thấp.

Mặc dù **'freetime'** là yếu tố có ảnh hưởng, mô hình SVM gặp khó khăn trong việc phân chia rõ ràng hai nhóm do **dữ liệu không cân bằng**. Điều này dẫn đến việc mô hình khó nhận diện chính xác nhóm tiêu thụ rượu cao, thể hiện qua **Precision và F1-score thấp** cho lớp này.

**Ảnh hưởng yếu của số ngày nghỉ đến tiêu thụ rượu bia:**

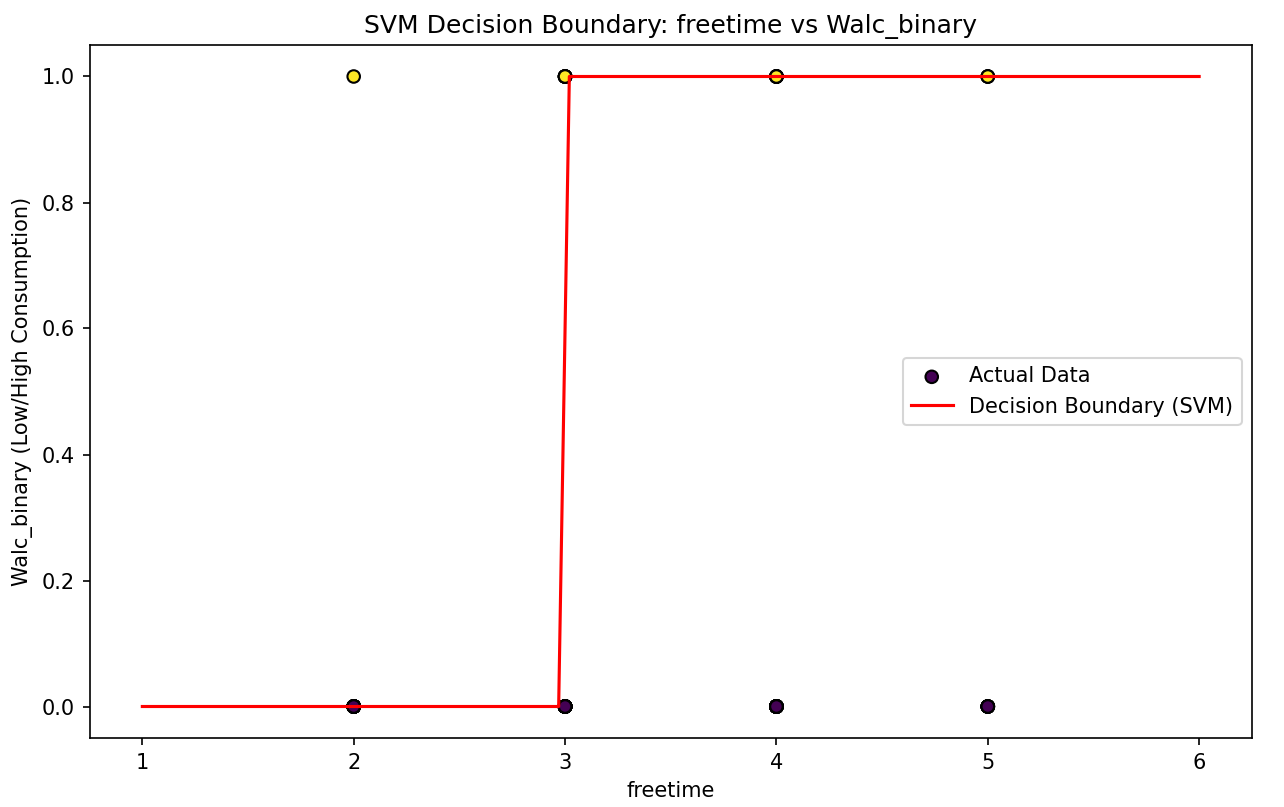
Mối quan hệ giữa thời gian rảnh rỗi và mức tiêu thụ rượu tồn tại nhưng không mạnh mẽ. Những người có nhiều thời gian rảnh rỗi có xu hướng tiêu thụ rượu cao hơn, nhưng điều này không phải yếu tố quyết định duy nhất.



Hình :biểu đồ 3D giữa absences và Walc và freetime

Biểu đồ 3D giữa các biến Freetime (thời gian rảnh), Walc (tiêu thụ rượu), và Absences (số buổi vắng) cho thấy rằng:

Các điểm dữ liệu có giá trị Freetime thấp (với giá trị từ 0 đến 10) có xu hướng nằm gần nhau, và hầu hết là thuộc nhóm Walc thấp (màu vàng). Các điểm có giá trị Freetime cao (trên 10) có xu hướng thuộc nhóm Walc cao (màu tím). Biểu đồ cũng cho thấy sự phân bổ giữa các nhóm Walc và sự ảnh hưởng của các biến khác như Absences me. Các điểm dữ liệu có thể tạo thành các nhóm riêng biệt dựa trên sự phân bố của các giá trị trong ba chiều này.



Hình :sơ đồ đường quyết định của mô hình SVM

Phần lớn các điểm dữ liệu được phân tách rõ ràng theo hai nhóm (Walc\_binary = 0 và Walc\_binary = 1). Tuy nhiên, một số điểm dữ liệu nằm gần ranh giới quyết định (decision boundary) của mô hình SVM, đặc biệt tại các giá trị freetime gần ngưỡng 3. Những điểm này có khả năng gây ra nhầm lẫn trong dự đoán, đặc biệt khi mô hình phải phân biệt giữa mức tiêu thụ rượu bia thấp và cao.

Các điểm gần đường biên cho thấy sự không chắc chắn trong việc phân loại và có thể ảnh hưởng đáng kể đến độ chính xác tổng thể của mô hình. Điều này đặc biệt quan trọng đối với nhóm thiểu số (nhóm có mức tiêu thụ rượu bia cao), vì bất kỳ lỗi phân loại nào cũng sẽ có tác động lớn hơn đến hiệu suất của mô hình trên toàn bộ tập dữ liệu. Việc xử lý những điểm này (ví dụ: sử dụng các kỹ thuật cân bằng lớp hoặc điều chỉnh mô hình) có thể cải thiện hiệu quả dự đoán.

### **3.3.5 So sánh và kết luận giữa 4 phương pháp: Hồi quy Logistic, Cây Quyết Định, Rừng Ngẫu Nhiên và SVM.**

**1. Độ chính xác tổng thể (Accuracy):**

* **Cây quyết định**: 48.39%
* **Hồi quy Logistic**: 48%
* **SVM**: 48.39%
* **Rừng ngẫu nhiên**: 48.39%

**=>** Tất cả các mô hình đều có độ chính xác tương tự nhau (≈48%), phản ánh rằng dữ liệu và đặc trưng đầu vào hiện tại chưa đủ mạnh để phân loại tốt.

**2. F1-Score (Trung bình macro):**

F1-Score macro trung bình đo lường sự cân bằng dự đoán giữa các lớp.

* **Cây quyết định**: 0.41
* **Hồi quy Logistic**: 0.41
* **SVM**: 0.41
* **Rừng ngẫu nhiên**: 0.41

**=>** Các mô hình đều đạt kết quả thấp, thể hiện sự cân bằng không tốt giữa hai lớp.

**3. Khả năng dự đoán lớp khó (Lớp 1):**

Precision và Recall cho lớp 1 là chỉ số quan trọng để đánh giá khả năng phát hiện lớp ít phổ biến.

* **Cây quyết định**: Precision = 0.14, Recall = 0.33, F1 = 0.20
* **Hồi quy Logistic**: Precision = 0.14, Recall = 0.33, F1 = 0.20
* **SVM**: Precision = 0.14, Recall = 0.33, F1 = 0.20
* **Rừng ngẫu nhiên**: Precision = 0.14, Recall = 0.33, F1 = 0.20

**=>** Không có mô hình nào nổi bật trong dự đoán lớp 1. Hiệu suất lớp này đều kém do mất cân bằng dữ liệu và số lượng mẫu ít.

**4. Độ cân bằng tổng thể giữa hai lớp (Macro Precision và Macro Recall):**

* **Cây quyết định**: Precision = 0.45, Recall = 0.43
* **Hồi quy Logistic**: Precision = 0.45, Recall = 0.43
* **SVM**: Precision = 0.45, Recall = 0.43
* **Rừng ngẫu nhiên**: Precision = 0.45, Recall = 0.43

**=>** Không có mô hình nào vượt trội. Độ cân bằng giữa hai lớp đều thấp.

**Kết luận:**

Freetime có xu hướng tác động dương đến Walc, nhưng tác động này không đủ mạnh hoặc ổn định để phân loại hoặc dự đoán. Các yếu tố khác ngoài freetime (chẳng hạn: áp lực xã hội, thu nhập, đặc điểm cá nhân) có thể quan trọng hơn trong việc xác định mức độ tiêu thụ rượu. Dữ liệu hiện tại không đủ mạnh để đưa ra kết luận chắc chắn, cần thêm thông tin hoặc đặc trưng khác để cải thiện mô hình và hiểu rõ hơn về mối quan hệ này.

## 3.4.Tìm hiểu mối quan hệ giữa goout và Walc

### **3.4.1.Phân tích mô hình hồi quy (Logistic Regression hoặc Multinomial Logistic Regression)**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.linear\_model import LogisticRegression**

**import numpy as np**

**# Đọc dữ liệu từ file CSV**

**data = pd.read\_csv('D:\\python\\BAITAPDULIEU\\DULIEU\\FINAL.csv')**

**# Chuyển đổi `Walc` thành 2 lớp (0 và 1)**

**data['Walc'] = data['Walc'].apply(lambda x: 1 if x > 2 else 0)**

**# Biến độc lập và phụ thuộc**

**X = data[['goout']] # Chọn biến độc lập goout (giữ nguyên)**

**y = data['Walc'] # Chọn biến phụ thuộc Walc (mức độ hài lòng)**

**# Chia dữ liệu thành tập huấn luyện và kiểm tra (70% huấn luyện, 30% kiểm tra)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)**

**# Khởi tạo mô hình Logistic Regression với phân loại nhị phân**

**model = LogisticRegression(solver='lbfgs')**

**# Huấn luyện mô hình**

**model.fit(X\_train, y\_train)**

**# Dự đoán trên tập kiểm tra**

**y\_pred = model.predict(X\_test)**

**# Đánh giá mô hình: Hiển thị classification report**

**print("\nClassification Report:")**

**print(classification\_report(y\_test, y\_pred, zero\_division=1))**

**# Tính độ chính xác (Accuracy)**

**accuracy = accuracy\_score(y\_test, y\_pred)**

**print(f'\nĐộ chính xác của mô hình: {accuracy \* 100:.2f}%')**

**# Hiển thị hệ số hồi quy và hệ số chặn của mô hình**

**print("\nHệ số hồi quy (Coefficients):", model.coef\_)**

**print("\nHệ số chặn (Intercept):", model.intercept\_)**

**# Tạo lưới điểm cho biến độc lập 'goout' để vẽ đường cong**

**X\_range = np.linspace(X['goout'].min() - 1, X['goout'].max() + 1, 300).reshape(-1, 1)**

**# Dự đoán xác suất cho mỗi giá trị của 'goout'**

**y\_prob = model.predict\_proba(X\_range)[:, 1] # Lấy xác suất cho lớp '1'**

**# Vẽ biểu đồ phân tán của dữ liệu**

**plt.figure(figsize=(8, 6))**

**sns.scatterplot(x=X['goout'], y=y, palette='viridis', s=100, marker='o', label='Dữ liệu thực tế')**

**# Vẽ đường cong hồi quy logistic (sigmoid curve)**

**plt.plot(X\_range, y\_prob, color='red', label='Đường cong hồi quy logistic', linewidth=2)**

**# Thiết lập tiêu đề và nhãn cho các trục**

**plt.title('Biểu đồ Hồi Quy Logistic: Xác Suất Dự Đoán với Đi chơi (Goout)')**

**plt.xlabel('Đi chơi (Goout)')**

**plt.ylabel('Xác suất dự đoán Walc (Mức độ hài lòng)')**

**# Hiển thị chú giải**

**plt.legend()**

**# Hiển thị biểu đồ**

**plt.show()**

**# Xuất ma trận nhầm lẫn (Confusion Matrix)**

**conf\_matrix = confusion\_matrix(y\_test, y\_pred)**

**print("\nMa trận nhầm lẫn:")**

**print(conf\_matrix)**

**# Trực quan hóa ma trận nhầm lẫn**

**plt.figure(figsize=(6, 5))**

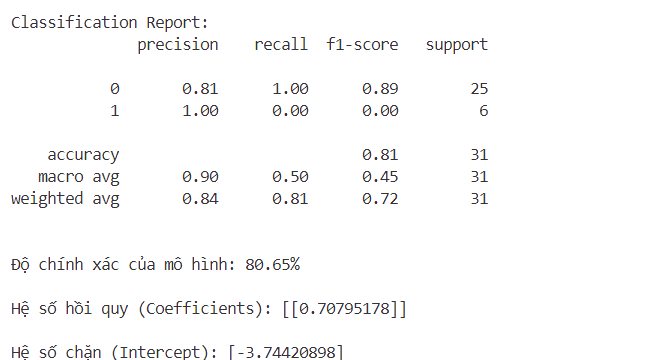
**sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Không hài lòng', 'Hài lòng'], yticklabels=['Không hài lòng', 'Hài lòng'])**

**plt.title('Ma Trận Nhầm Lẫn (Confusion Matrix)')**

**plt.xlabel('Dự đoán')**

**plt.ylabel('Thực tế')**

**plt.show()**

**Kết quả và diễn giải:  
**

Hình :Kết quả của hồi quy logistic

**Diễn giải kết quả:**

**1.Classification Report**

**a. Precision, Recall và F1-score**

* **Precision**: Đo lường độ chính xác của mô hình khi dự đoán một lớp nào đó. Tức là trong tất cả các lần mô hình dự đoán lớp đó, bao nhiêu lần là đúng.
  + **Lớp 0 (Tiêu thụ rượu thấp)**: Precision = 0.81 → Mô hình có độ chính xác 81% khi dự đoán người có mức tiêu thụ rượu thấp.
  + **Lớp 1 (Tiêu thụ rượu cao)**: Precision = 1.00 → Mô hình có độ chính xác 100% khi dự đoán người có mức tiêu thụ rượu cao.
* **Recall (Sensitivity)**: Đo lường khả năng mô hình phát hiện đúng tất cả các trường hợp thực tế của lớp. Tức là trong tất cả các mẫu thực tế của lớp này, bao nhiêu mẫu được mô hình dự đoán chính xác.
  + **Lớp 0 (Tiêu thụ rượu thấp)**: Recall = 1.00 → Mô hình phát hiện chính xác tất cả các trường hợp thực tế của lớp 0 (tiêu thụ rượu thấp).
  + **Lớp 1 (Tiêu thụ rượu cao)**: Recall = 0.00 → Mô hình không phát hiện được trường hợp nào của lớp 1 (tiêu thụ rượu cao). Điều này có nghĩa là mô hình không phân loại được bất kỳ ai là "tiêu thụ rượu cao" (lớp 1), mặc dù trong thực tế có 6 mẫu "tiêu thụ rượu cao".
* **F1-score**: Là chỉ số kết hợp giữa precision và recall. Đây là một phép đo toàn diện hơn về hiệu suất mô hình khi dữ liệu có sự mất cân bằng.
  + **Lớp 0 (Tiêu thụ rượu thấp)**: F1-score = 0.89 → Mô hình có hiệu suất rất tốt khi dự đoán lớp này.
  + **Lớp 1 (Tiêu thụ rượu cao)**: F1-score = 0.00 → Mô hình không có khả năng phân loại lớp này.

**b. Accuracy (Độ chính xác tổng thể):**

* Accuracy = 0.81 → Mô hình chính xác 81% trong tổng số mẫu thử nghiệm. Tuy nhiên, độ chính xác cao này chủ yếu do lớp 0 (tiêu thụ rượu thấp) chiếm ưu thế và mô hình dự đoán chính xác hầu hết các trường hợp này.

**c. Macro avg và Weighted avg:**

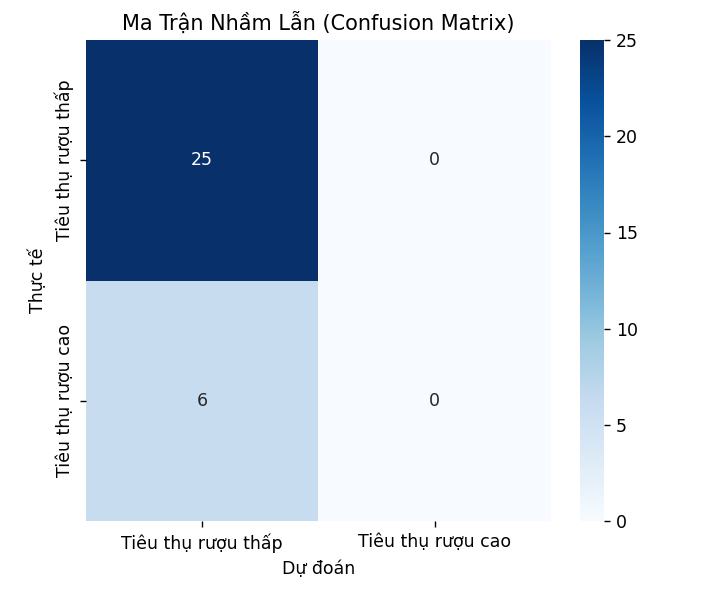
* **Macro avg**: Tính toán trung bình cho tất cả các lớp mà không cân nhắc tần suất của các lớp. Đây là trung bình của tất cả các chỉ số precision, recall và F1-score của các lớp:
  + **Precision (Macro avg)** = 0.90
  + **Recall (Macro avg)** = 0.50
  + **F1-score (Macro avg)** = 0.45
* **Weighted avg**: Tính toán trung bình có cân nhắc tần suất của các lớp. Tức là, các lớp có nhiều mẫu sẽ ảnh hưởng lớn hơn đến giá trị trung bình:
  + **Precision (Weighted avg)** = 0.84
  + **Recall (Weighted avg)** = 0.81
  + **F1-score (Weighted avg)** = 0.72

**2. Hệ số hồi quy (Coefficients):**

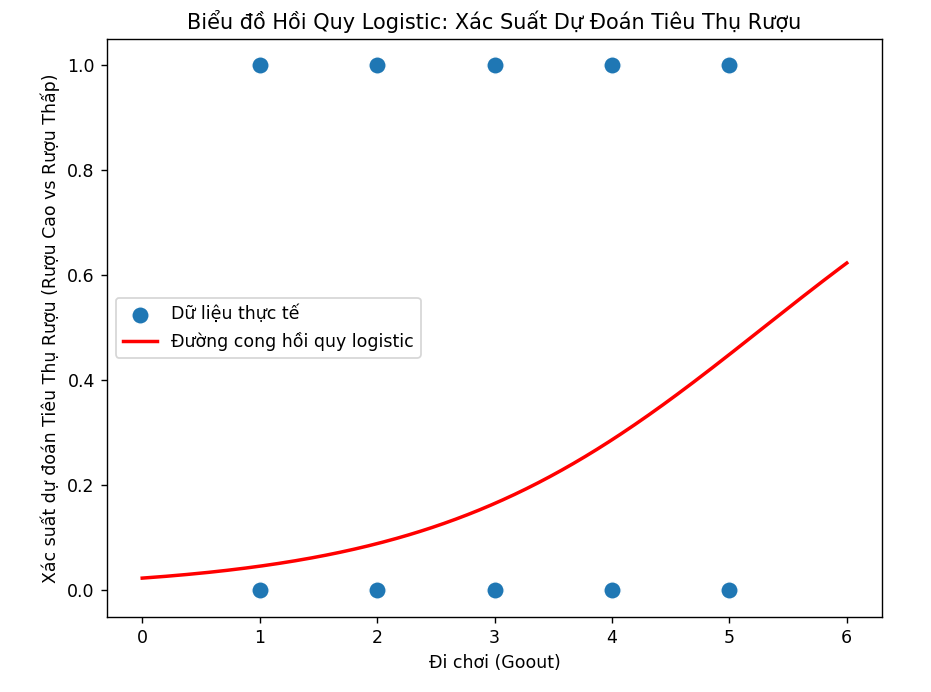
* **Hệ số hồi quy (Coefficient)** = 0.70795178 → Đây là hệ số của biến goout trong mô hình hồi quy logistic. Hệ số này cho biết mức độ thay đổi của log-odds (logarit của tỷ lệ xác suất) của việc một cá nhân thuộc lớp "tiêu thụ rượu cao" (lớp 1) khi giá trị goout tăng thêm một đơn vị.
  + Nếu goout tăng 1 đơn vị, tỷ lệ log-odds của việc người đó tiêu thụ rượu cao (lớp 1) tăng thêm 0.71.

**3. Hệ số chặn (Intercept):**

* **Hệ số chặn (Intercept)** = -3.74420898 → Đây là điểm mà đường hồi quy logistic cắt trục y. Cụ thể, hệ số chặn này có thể hiểu là log-odds của việc một cá nhân thuộc lớp "tiêu thụ rượu cao" (lớp 1) khi giá trị của goout bằng 0.
  + Trong trường hợp này, log-odds của việc thuộc lớp 1 khi goout = 0 là -3.74. Chuyển đổi log-odds sang xác suất sẽ cho biết rằng xác suất của việc người đó tiêu thụ rượu cao là rất thấp khi họ không đi chơi (goout = 0).



Hình :Ma trận nhầm lẫn



Hình :Biểu đồ trực quan hóa Hồi Quy logistic

Ma trận nhầm lẫn trong hình biểu thị kết quả của mô hình hồi quy logistic khi phân loại hai lớp: "Tiêu thụ rượu thấp" (0) và "Tiêu thụ rượu cao" (1). Dưới đây là ý nghĩa của từng giá trị trong ma trận:

**1. Ý nghĩa của các ô trong ma trận:**

* **Ô trên cùng bên trái (25)**:
  + Dự đoán: Tiêu thụ rượu thấp (0).
  + Thực tế: Tiêu thụ rượu thấp (0).
  + Số lượng: 25.
  + Ý nghĩa: Mô hình đã dự đoán đúng 25 trường hợp thuộc lớp "Tiêu thụ rượu thấp".
* **Ô trên cùng bên phải (0)**:
  + Dự đoán: Tiêu thụ rượu cao (1).
  + Thực tế: Tiêu thụ rượu thấp (0).
  + Số lượng: 0.
  + Ý nghĩa: Không có trường hợp nào bị mô hình dự đoán sai là "Tiêu thụ rượu cao" trong khi thực tế là "Tiêu thụ rượu thấp".
* **Ô dưới cùng bên trái (6)**:
  + Dự đoán: Tiêu thụ rượu thấp (0).
  + Thực tế: Tiêu thụ rượu cao (1).
  + Số lượng: 6.
  + Ý nghĩa: Mô hình đã dự đoán sai 6 trường hợp là "Tiêu thụ rượu thấp" trong khi thực tế là "Tiêu thụ rượu cao".
* **Ô dưới cùng bên phải (0)**:
  + Dự đoán: Tiêu thụ rượu cao (1).
  + Thực tế: Tiêu thụ rượu cao (1).
  + Số lượng: 0.
  + Ý nghĩa: Mô hình không dự đoán đúng bất kỳ trường hợp nào thuộc lớp "Tiêu thụ rượu cao".

**2. Đánh giá hiệu suất mô hình:**

* **Precision (Độ chính xác dự đoán cho mỗi lớp)**:
  + Lớp 0 (Tiêu thụ rượu thấp): Rất tốt, vì mô hình dự đoán đúng tất cả trường hợp của lớp này.
  + Lớp 1 (Tiêu thụ rượu cao): Precision không xác định vì không có dự đoán nào thuộc lớp 1 (mô hình không dự đoán được trường hợp "Tiêu thụ rượu cao").
* **Recall (Độ nhạy cho mỗi lớp)**:
  + Lớp 0 (Tiêu thụ rượu thấp): 1.00 (100%), vì tất cả các trường hợp thực tế của lớp 0 đều được dự đoán chính xác.
  + Lớp 1 (Tiêu thụ rượu cao): 0.00 (0%), vì không có trường hợp nào thuộc lớp 1 được mô hình dự đoán đúng.

**Kết luận của thuật toán:**

=> Mô hình đã đánh giá được mức độ tiêu thụ rượu thấp rất chính xác,cho thấy rằng việc ra ngoài đi chơi của các bạn học sinh sinh viên sẽ ít sử dụng rượu.Nhưng nếu chỉ dùng biến goout để kết luận mức độ tiêu thụ rượu thì không được chính xác và không đầy đủ.

### **3.4.2.Phân tích sử dụng Cây quyết định (Decision Tree)**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**from sklearn.tree import DecisionTreeClassifier, plot\_tree**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score**

**# Đọc dữ liệu từ file CSV**

**data = pd.read\_csv('D:\\python\\BAITAPDULIEU\\DULIEU\\FINAL.csv')**

**# Chuyển đổi 'Walc' thành 2 lớp: 0 là tiêu thụ rượu thấp, 1 là tiêu thụ rượu cao**

**data['Walc\_binary'] = data['Walc'].apply(lambda x: 1 if x > 2 else 0)**

**# Biến độc lập và phụ thuộc**

**X = data[['goout']] # Biến độc lập là activities**

**y = data['Walc\_binary'] # Biến phụ thuộc là Walc\_binary**

**# Chia dữ liệu thành tập huấn luyện và kiểm tra**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)**

**# Tạo và huấn luyện mô hình cây quyết định**

**model = DecisionTreeClassifier(criterion='entropy', random\_state=42, class\_weight='balanced')**

**model.fit(X\_train, y\_train)**

**# Dự đoán trên tập kiểm tra**

**y\_pred = model.predict(X\_test)**

**# Đánh giá mô hình**

**print("\nClassification Report:")**

**print(classification\_report(y\_test, y\_pred))**

**print("\nConfusion Matrix:")**

**print(confusion\_matrix(y\_test, y\_pred))**

**# Tính toán và in độ chính xác**

**accuracy = accuracy\_score(y\_test, y\_pred)**

**print(f"\nĐộ chính xác của mô hình là: {accuracy \* 100:.2f}%")**

**# Vẽ biểu đồ cây quyết định**

**plt.figure(figsize=(12, 8))**

**plot\_tree(model, feature\_names=['activities'], class\_names=['Tiêu thụ rượu thấp', 'Tiêu thụ rượu cao'],**

**filled=True, rounded=True, fontsize=10)**

**plt.title('Decision Tree for Activities vs Walc\_binary')**

**plt.show()**

**# Tạo lưới các giá trị của activities từ min đến max**

**x\_min, x\_max = X['goout'].min() - 1, X['activities'].max() + 1**

**xx = np.linspace(x\_min, x\_max, num=100).reshape(-1, 1)**

**# Chuyển đổi xx thành DataFrame**

**xx\_df = pd.DataFrame(xx, columns=['goout'])**

**# Dự đoán ranh giới quyết định**

**pred\_boundary = model.predict(xx\_df)**

**# Vẽ sơ đồ đường quyết định**

**plt.figure(figsize=(10, 6))**

**plt.plot(xx, pred\_boundary, color='red', label='Decision Boundary')**

**plt.scatter(X\_train, y\_train, c='blue', marker='o', label='Train Data')**

**plt.scatter(X\_test, y\_test, c='green', marker='x', label='Test Data')**

**plt.title('Decision Boundary for goout vs Walc\_binary')**

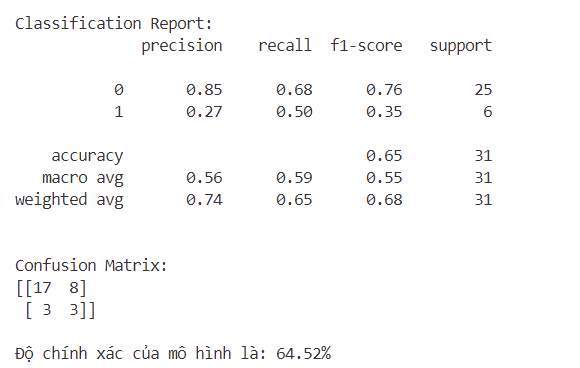
**plt.xlabel('goout')**

**plt.ylabel('Walc\_binary (Tiêu thụ rượu thấp/cao)')**

**plt.legend()**

**plt.show()**

**Kết quả và diễn giải:**

****

Hình :Kết quả của mô hình cây

**1.Classification Report**

* **Precision (Độ chính xác)**:
  + Đối với lớp 0 (Tiêu thụ rượu thấp), precision là 0.85. Điều này có nghĩa là 85% các mẫu được phân loại là lớp 0 thực sự thuộc lớp 0. Mô hình làm khá tốt trong việc không phân loại nhầm các mẫu tiêu thụ rượu thấp thành tiêu thụ rượu cao.
  + Đối với lớp 1 (Tiêu thụ rượu cao), precision là 0.27. Điều này có nghĩa là chỉ có 27% các mẫu được phân loại là lớp 1 thực sự thuộc lớp 1. Mô hình gặp khó khăn trong việc phân loại chính xác các mẫu tiêu thụ rượu cao.
* **Recall (Độ nhạy)**:
  + Đối với lớp 0, recall là 0.68. Mô hình nhận diện được 68% các mẫu thực sự thuộc lớp 0.
  + Đối với lớp 1, recall là 0.50. Mô hình nhận diện được 50% các mẫu thực sự thuộc lớp 1, tức là một nửa các mẫu tiêu thụ rượu cao được phân loại đúng.
* **F1-Score (Điểm F1)**:
  + Đối với lớp 0, F1-score là 0.76, cho thấy sự kết hợp khá tốt giữa precision và recall.
  + Đối với lớp 1, F1-score là 0.35, khá thấp, cho thấy mô hình gặp khó khăn trong việc phân loại chính xác lớp này.
* **Accuracy (Độ chính xác chung)**: Mô hình có accuracy là 64.52%, cho thấy mô hình phân loại đúng khoảng 65% tổng số mẫu.
* **Macro Average**: Trung bình của các chỉ số trên cho cả hai lớp, không tính đến tỉ lệ mẫu giữa các lớp:
  + **Precision**: **0.56**, chỉ ra rằng mô hình có hiệu suất không tốt trong việc phân loại các lớp, đặc biệt là với lớp 1.
  + **Recall**: **0.59**, cho thấy mô hình có khả năng nhận diện khá tốt, nhưng không quá xuất sắc đối với lớp 1.
  + **F1-score**: **0.55**, trung bình thấp, phản ánh rằng mô hình còn có nhiều điểm yếu trong việc phân loại đúng.
* **Weighted Average**: Trung bình có trọng số theo tỉ lệ mẫu trong từng lớp:
  + **Precision**: **0.74**, phản ánh rằng mô hình làm tốt khi phân loại lớp 0 (vì lớp này chiếm ưu thế hơn).
  + **Recall**: **0.65**, phản ánh khả năng nhận diện mẫu của mô hình, đặc biệt là lớp 0.
  + **F1-score**: **0.68**, cho thấy mô hình có hiệu suất tương đối ổn khi xét trên tổng thể.

**2. Confusion Matrix:**

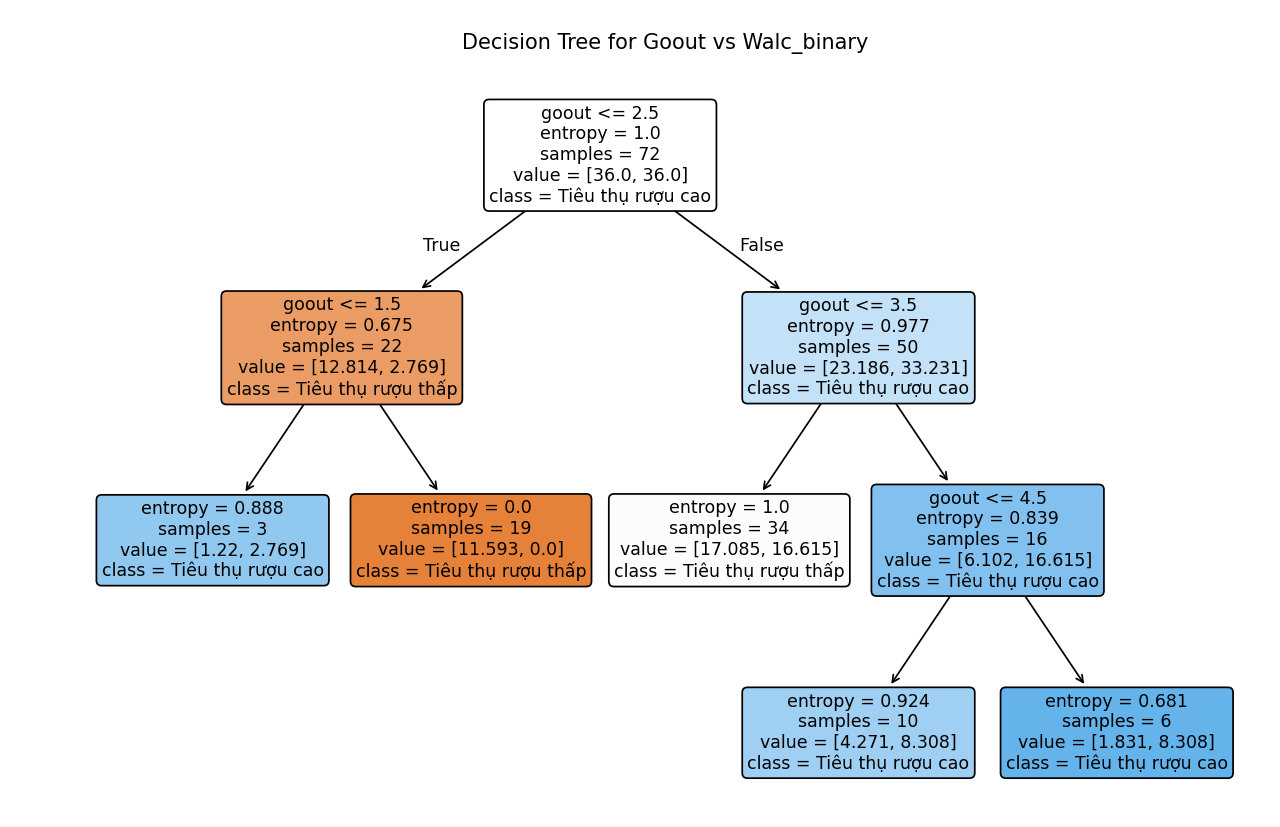
[17 8]

[3 3]

* **17**: Số lượng mẫu thuộc lớp 0 (Tiêu thụ rượu thấp) được mô hình phân loại đúng (True Negatives).
* **8**: Số lượng mẫu thuộc lớp 0 nhưng mô hình lại phân loại nhầm thành lớp 1 (False Positives).
* **3**: Số lượng mẫu thuộc lớp 1 (Tiêu thụ rượu cao) nhưng mô hình lại phân loại nhầm thành lớp 0 (False Negatives).
* **3**: Số lượng mẫu thuộc lớp 1 và được mô hình phân loại đúng (True Positives).

**3. Tổng kết:**

* Độ chính xác trung bình: Mô hình có độ chính xác 64.52%, không quá cao, nhưng có thể chấp nhận được trong trường hợp bài toán không đòi hỏi độ chính xác tuyệt đối. Tuy nhiên, mô hình vẫn còn những điểm yếu lớn trong việc phân loại chính xác lớp 1.
* Hiệu suất của lớp 0 (Tiêu thụ rượu thấp):
  + Mô hình khá tốt khi phân loại lớp 0, với precision và F1-score cao. Đặc biệt, precision là 0.85, cho thấy khả năng phân loại đúng lớp này của mô hình rất cao.
* Hiệu suất của lớp 1 (Tiêu thụ rượu cao):
  + Mô hình có precision và F1-score thấp đối với lớp 1, chỉ đạt 0.27 và 0.35. Điều này cho thấy mô hình gặp khó khăn trong việc phân loại chính xác lớp 1, dễ phân loại nhầm các mẫu lớp này thành lớp 0.
* Lớp không cân đối: Mặc dù mô hình đã cải thiện khả năng phân loại lớp 0 nhưng vẫn còn vấn đề trong việc phân loại lớp 1. Các kỹ thuật như SMOTE (Synthetic Minority Over-sampling Technique) hoặc undersampling lớp 0 có thể giúp cân bằng dữ liệu và cải thiện hiệu suất đối với lớp thiểu số.



Hình :Trực quan hóa cây

* + **Cây quyết định**:
    - Cây quyết định này phân chia dữ liệu thành 2 nhánh chính dựa vào giá trị của biến goout.
    - Nhánh đầu tiên (ở phía trên cây): Nếu goout nhỏ hơn hoặc bằng một giá trị cụ thể, ví dụ: goout <= 1.5, thì mô hình phân loại vào nhóm Tiêu thụ rượu thấp (0). Đây là một nhánh khá mạnh, phân loại rõ ràng các đối tượng với hành vi đi chơi ít hoặc không đi chơi (thấp).
    - Nhánh thứ hai (phía dưới cây): Nếu goout lớn hơn giá trị trên, ví dụ: goout > 1.5, thì mô hình phân loại vào nhóm Tiêu thụ rượu cao (1). Điều này cho thấy rằng những người có hành vi đi chơi nhiều hoặc thường xuyên sẽ có khả năng tiêu thụ rượu cao hơn.
  + **Đặc điểm của phân vùng**:
    - Đi chơi ít hoặc không đi chơi (goout ≤ 1.5): Những người này có xu hướng tiêu thụ rượu ít hoặc không uống rượu, mô hình quyết định gán vào nhóm 0 (rượu thấp).
    - Đi chơi nhiều (goout > 1.5): Những người này có xu hướng tiêu thụ rượu cao hơn, vì vậy mô hình quyết định gán vào nhóm 1 (rượu cao).

**Đánh giá mô hình thông qua biểu đồ cây:**

* **Hiệu quả phân chia**: Cây quyết định có vẻ hiệu quả trong việc phân chia giữa hai nhóm (rượu thấp và rượu cao) dựa trên hành vi đi chơi. Tuy nhiên, độ chính xác và độ phủ của mô hình có thể phụ thuộc vào độ phân tán của dữ liệu và cách mà "goout" ảnh hưởng đến hành vi tiêu thụ rượu.
* **Chất lượng phân vùng**: Biểu đồ cây cho thấy mô hình phân chia dữ liệu theo một ngưỡng rõ ràng, nhưng ngưỡng này có thể không đủ linh hoạt đối với các dữ liệu phức tạp hơn (ví dụ như nếu dữ liệu có nhiều yếu tố ảnh hưởng đến tiêu thụ rượu ngoài hành vi đi chơi).

### **3.4.3.Rừng Ngẫu Nhiên (Random Forest)**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score**

**# Đọc dữ liệu từ file CSV**

**data = pd.read\_csv('D:\\python\\BAITAPDULIEU\\DULIEU\\FINAL.csv')**

**# Chuyển đổi 'Walc' thành 2 lớp: 0 là tiêu thụ rượu thấp, 1 là tiêu thụ rượu cao**

**data['Walc\_binary'] = data['Walc'].apply(lambda x: 1 if x > 2 else 0)**

**# Biến độc lập và phụ thuộc**

**X = data[['goout']] # Biến độc lập là goout**

**y = data['Walc\_binary'] # Biến phụ thuộc là Walc\_binary**

**# Chia dữ liệu thành tập huấn luyện và kiểm tra**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)**

**# Tạo và huấn luyện mô hình Rừng Ngẫu Nhiên (Random Forest)**

**model = RandomForestClassifier(n\_estimators=100, random\_state=42, class\_weight='balanced')**

**model.fit(X\_train, y\_train)**

**# Dự đoán trên tập kiểm tra**

**y\_pred = model.predict(X\_test)**

**# Đánh giá mô hình**

**print("\nClassification Report:")**

**print(classification\_report(y\_test, y\_pred))**

**print("\nConfusion Matrix:")**

**print(confusion\_matrix(y\_test, y\_pred))**

**# Tính toán và in độ chính xác**

**accuracy = accuracy\_score(y\_test, y\_pred)**

**print(f"\nĐộ chính xác của mô hình là: {accuracy \* 100:.2f}%")**

**# Hiển thị đặc trưng quan trọng (Feature Importance)**

**features = model.feature\_importances\_**

**print("\nĐặc trưng quan trọng (Feature Importances):", features)**

**# Vẽ đồ thị đặc trưng quan trọng**

**plt.figure(figsize=(8, 6))**

**plt.barh(['goout'], features) # Đổi tên từ 'activities' thành 'goout'**

**plt.title('Feature Importance')**

**plt.xlabel('Feature Importance')**

**plt.show()**

**# Vẽ sơ đồ quyết định của Rừng Ngẫu Nhiên**

**plt.figure(figsize=(10, 6))**

**x\_min, x\_max = X['goout'].min() - 1, X['goout'].max() + 1 # Đổi từ 'activities' thành 'goout'**

**xx = np.linspace(x\_min, x\_max, num=100).reshape(-1, 1)**

**# Dự đoán trên lưới giá trị của 'goout'**

**pred\_boundary = model.predict(xx)**

**# Vẽ sơ đồ đường quyết định**

**plt.plot(xx, pred\_boundary, color='red', label='Decision Boundary')**

**plt.scatter(X\_train, y\_train, c='blue', marker='o', label='Train Data')**

**plt.scatter(X\_test, y\_test, c='green', marker='x', label='Test Data')**

**plt.title('Decision Boundary for Goout vs Walc\_binary with Random Forest')**

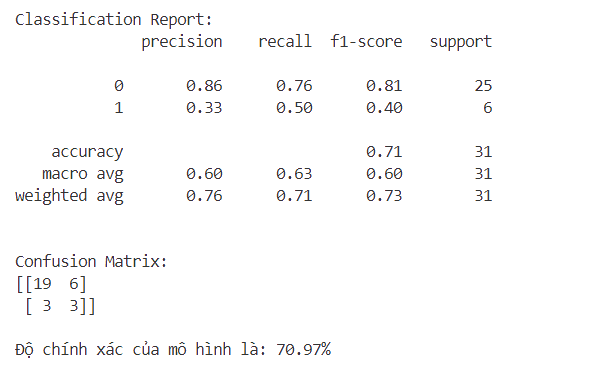
**plt.xlabel('Goout') # Đổi tên từ 'Activities' thành 'Goout'**

**plt.ylabel('Walc\_binary (Tiêu thụ rượu thấp/cao)')**

**plt.legend()**

**plt.show()**

**Kết quả và diễn giải:**



Hình :Kết quả từ mô hình Rừng ngẫu nhiên

**1. Classification Report:**

Dưới đây là các chỉ số cần chú ý:

* Precision: Độ chính xác của mô hình khi dự đoán là một lớp cụ thể (lớp 0 hoặc lớp 1).
  + Lớp 0 (Tiêu thụ rượu thấp): Precision = 0.86 → Mô hình chính xác 86% khi dự đoán là tiêu thụ rượu thấp.
  + Lớp 1 (Tiêu thụ rượu cao): Precision = 0.33 → Mô hình chỉ chính xác 33% khi dự đoán là tiêu thụ rượu cao.
* Recall: Độ nhạy của mô hình, nghĩa là khả năng mô hình nhận diện đúng tất cả các trường hợp thuộc mỗi lớp.
  + Lớp 0 (Tiêu thụ rượu thấp): Recall = 0.76 → Mô hình nhận diện đúng 76% các trường hợp tiêu thụ rượu thấp.
  + Lớp 1 (Tiêu thụ rượu cao): Recall = 0.50 → Mô hình chỉ nhận diện đúng 50% các trường hợp tiêu thụ rượu cao.
* F1-Score: Là trung bình hài hòa giữa precision và recall, là chỉ số tổng hợp.
  + Lớp 0 (Tiêu thụ rượu thấp): F1-score = 0.81 → Đây là một chỉ số khá tốt, nghĩa là mô hình đã đạt được sự cân bằng giữa precision và recall cho lớp này.
  + Lớp 1 (Tiêu thụ rượu cao): F1-score = 0.40 → Đây là chỉ số khá thấp, cho thấy mô hình chưa thực sự hiệu quả trong việc nhận diện lớp này.
* Macro Average:
  + Precision: 0.60
  + Recall: 0.63
  + F1-score: 0.60
  + Macro average cho thấy mô hình không có sự chênh lệch quá lớn về hiệu suất giữa hai lớp, nhưng nó vẫn chưa hoàn hảo.
* Weighted Average: Được tính dựa trên số lượng mẫu trong mỗi lớp.
  + Precision: 0.76
  + Recall: 0.71
  + F1-score: 0.73
  + Đây là các chỉ số tổng thể tốt, đặc biệt là Precision và F1-score.

**2. Confusion Matrix:**

Confusion Matrix là công cụ quan trọng để hiểu rõ hơn về cách mô hình phân loại các lớp.

[[19 6]

[ 3 3]]

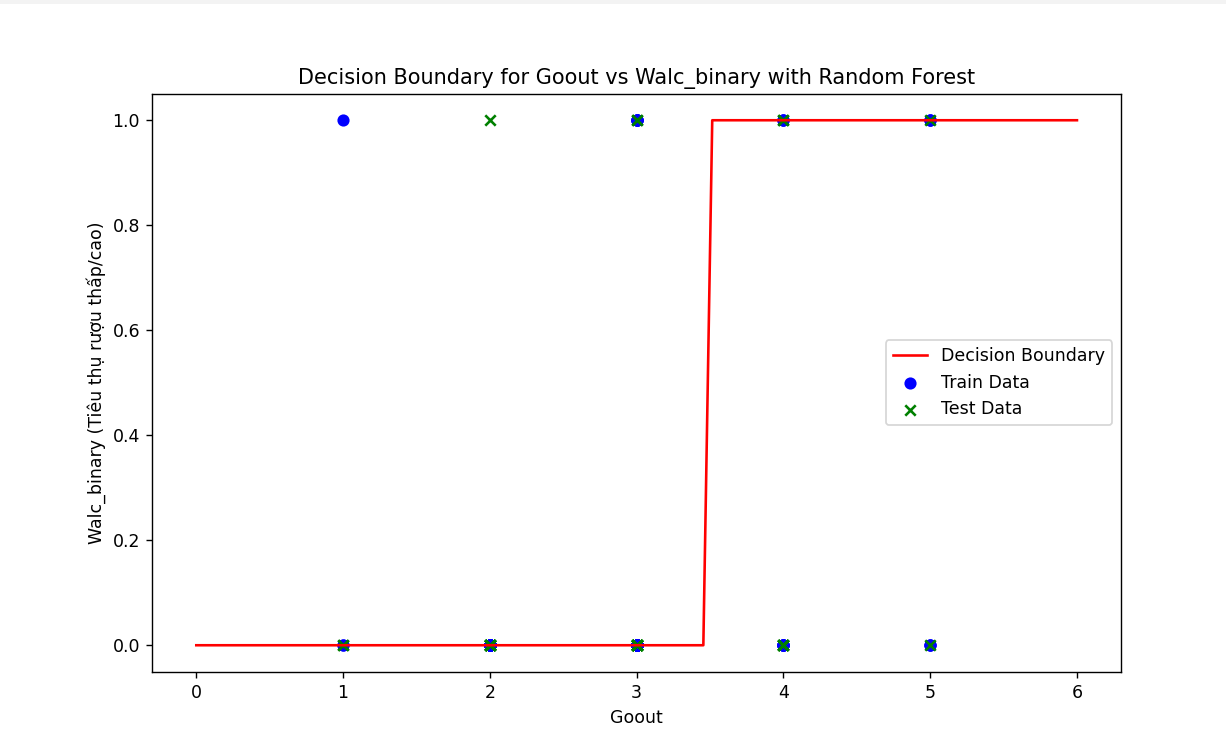
* Dòng 1 (Tiêu thụ rượu thấp):
  + 19 mẫu được phân loại chính xác là tiêu thụ rượu thấp (True Negatives - TN).
  + 6 mẫu bị phân loại sai là tiêu thụ rượu cao (False Positives - FP).
* Dòng 2 (Tiêu thụ rượu cao):
  + 3 mẫu bị phân loại sai là tiêu thụ rượu thấp (False Negatives - FN).
  + 3 mẫu được phân loại chính xác là tiêu thụ rượu cao (True Positives - TP).

**3. Độ chính xác (Accuracy):**

Độ chính xác chung của mô hình là 70.97%, có nghĩa là mô hình đã phân loại đúng 70.97% tổng số mẫu. Mặc dù đây là một kết quả tương đối tốt, nhưng ta cần chú ý rằng mô hình có sự chênh lệch lớn về hiệu quả giữa các lớp.

**Phân tích tổng quan:**

* Mô hình cho kết quả tốt khi phân loại lớp 0 (Tiêu thụ rượu thấp), với độ chính xác và recall cao (precision = 0.86 và recall = 0.76).
* Tuy nhiên, mô hình gặp khó khăn trong việc phân loại lớp 1 (Tiêu thụ rượu cao), với độ chính xác thấp (precision = 0.33) và recall thấp (recall = 0.50). Điều này có thể dẫn đến việc dự đoán sai nhiều mẫu thuộc lớp tiêu thụ rượu cao.



Hình :Biểu đồ trực quan hóa của rừng ngẫu nhiên

* Trục X (Goout - Đi chơi): Đại diện cho số lần đi chơi của cá nhân. Giá trị càng lớn thể hiện tần suất đi chơi cao hơn.
* Trục Y (Walc - Tiêu thụ rượu): Là nhãn nhị phân, với giá trị 0 là tiêu thụ rượu thấp và 1 là tiêu thụ rượu cao.
* Biểu đồ cho thấy sự phân bố của các giá trị thực tế và dự đoán. Dựa trên hình ảnh, có vẻ như hầu hết các điểm đều nằm gần giá trị 0 trên trục Y, cho thấy đa phần các trường hợp đều thuộc lớp "tiêu thụ rượu thấp".

### **3.4.4.Phân tích sử dụng Máy vector hỗ trợ (Support Vector Machine)**

**# Import các thư viện cần thiết**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.svm import SVC**

**from sklearn.metrics import classification\_report, confusion\_matrix**

**# Đọc dữ liệu từ file CSV**

**data = pd.read\_csv('D:\\python\\BAITAPDULIEU\\DULIEU\\FINAL.csv')**

**# Chuyển đổi Walc thành 2 lớp (0: Tiêu thụ rượu thấp, 1: Tiêu thụ rượu cao)**

**data['Walc\_binary'] = data['Walc'].apply(lambda x: 1 if x > 2 else 0)**

**# Lựa chọn biến độc lập và phụ thuộc**

**X = data[['goout']] # Chọn biến độc lập goout**

**y = data['Walc\_binary'] # Chọn biến phụ thuộc Walc\_binary**

**# Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra (70% huấn luyện, 30% kiểm tra)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)**

**# Huấn luyện mô hình SVM**

**model = SVC(kernel='linear', class\_weight='balanced', random\_state=42)**

**model.fit(X\_train, y\_train)**

**# Dự đoán trên tập kiểm tra**

**y\_pred = model.predict(X\_test)**

**# Ma trận nhầm lẫn**

**print("Confusion Matrix:")**

**conf\_matrix = confusion\_matrix(y\_test, y\_pred)**

**print(conf\_matrix)**

**# Báo cáo phân loại**

**print("\nClassification Report:")**

**print(classification\_report(y\_test, y\_pred))**

**# In ra độ chính xác của mô hình**

**from sklearn.metrics import accuracy\_score**

**accuracy = accuracy\_score(y\_test, y\_pred)**

**print("\nĐộ chính xác của mô hình SVM: {:.2f}%".format(accuracy \* 100))**

**# Vẽ biểu đồ trực quan hóa biên quyết định**

**# Tạo lưới để vẽ biên phân cách**

**xx = np.linspace(X['goout'].min() - 0.1, X['goout'].max() + 0.1, 100).reshape(-1, 1)**

**# Dự đoán nhãn cho từng điểm trên lưới**

**Z = model.decision\_function(xx)**

**# Vẽ biểu đồ**

**plt.figure(figsize=(8, 6))**

**plt.scatter(X['goout'], y, color='blue', s=100, marker='o', label='Dữ liệu thực tế')**

**plt.plot(xx, Z, color='red', label='Biên quyết định', linewidth=2)**

**# Thiết lập tiêu đề và nhãn trục**

**plt.title('Biểu đồ phân loại với SVM')**

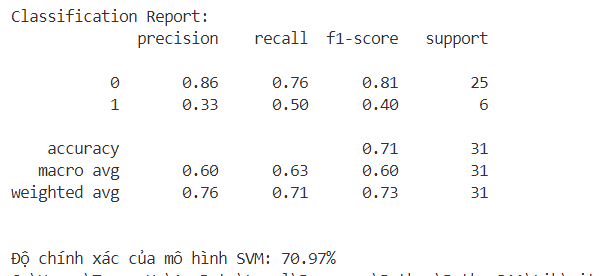
**plt.xlabel('Đi chơi (Goout)')**

**plt.ylabel('Tiêu thụ rượu vào buổi chiều (Walc)')**

**plt.legend()**

**plt.show()**

**Kết quả và diễn giải:**

****

Hình :Kết quả từ mô hình SVM

**1. Confusion Matrix:**

[[19 6]

[ 3 3]]

Ma trận nhầm lẫn mô tả số lượng các dự đoán đúng và sai của mô hình:

* **19** là số mẫu mà mô hình dự đoán chính xác là lớp 0 (Tiêu thụ rượu thấp).
* **6** là số mẫu mà mô hình dự đoán sai là lớp 1 (Tiêu thụ rượu cao), thực tế là lớp 0.
* **3** là số mẫu mà mô hình dự đoán sai là lớp 0, thực tế là lớp 1.
* **3** là số mẫu mà mô hình dự đoán chính xác là lớp 1 (Tiêu thụ rượu cao).

**2. Classification Report:**

* **Lớp 0 (Tiêu thụ rượu thấp)**:
  + **Precision**: 0.86: Trong số những dự đoán là lớp 0, 86% là đúng.
  + **Recall**: 0.76: Trong số những mẫu thực sự là lớp 0, 76% được dự đoán đúng là lớp 0.
  + **F1-score**: 0.81: Điểm hài hòa giữa precision và recall.
* **Lớp 1 (Tiêu thụ rượu cao)**:
  + **Precision**: 0.33: Trong số những dự đoán là lớp 1, chỉ có 33% là đúng.
  + **Recall**: 0.50: Trong số những mẫu thực sự là lớp 1, 50% được dự đoán đúng là lớp 1.
  + **F1-score**: 0.40: Điểm hài hòa giữa precision và recall.

**3. Accuracy:**

* Độ chính xác của mô hình là **70.97%**: Điều này có nghĩa là khoảng 71% số dự đoán của mô hình là đúng.

**4. Macro Average:**

* **Precision**: 0.60
* **Recall**: 0.63
* **F1-score**: 0.60
* Macro average tính toán các chỉ số này bằng cách lấy trung bình cộng các chỉ số của từng lớp, không phân biệt số lượng mẫu của mỗi lớp. Mô hình có kết quả khá đồng đều giữa các lớp.

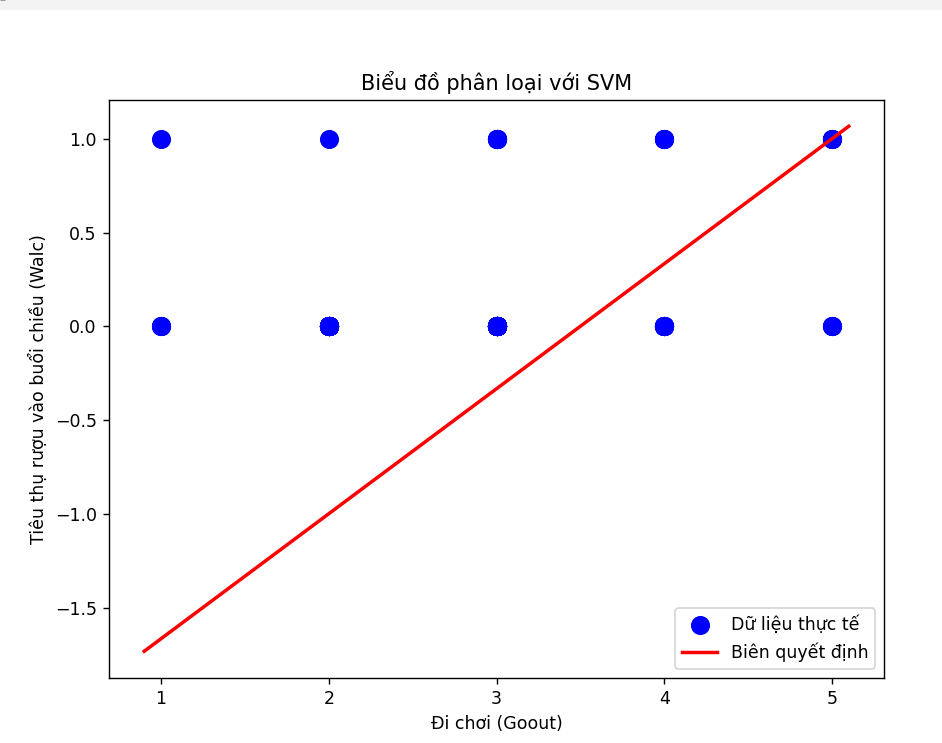
**5. Weighted Average:**

* **Precision**: 0.76
* **Recall**: 0.71
* **F1-score**: 0.73
* Weighted average tính toán các chỉ số này theo trọng số của từng lớp. Vì lớp 0 có số lượng mẫu lớn hơn lớp 1, nên điểm của lớp 0 ảnh hưởng mạnh đến điểm chung.

**Tổng kết:**

Mô hình SVM có độ chính xác tổng thể là **70.97%**, nhưng với lớp **1 (Tiêu thụ rượu cao)**, độ chính xác và recall còn thấp, điều này chỉ ra rằng mô hình gặp khó khăn trong việc phân loại đúng lớp này.

* **Lớp 0 (Tiêu thụ rượu thấp)** được phân loại tốt hơn với precision cao (0.86) và recall hợp lý (0.76).
* **Lớp 1 (Tiêu thụ rượu cao)** có độ chính xác thấp (precision 0.33), nhưng recall tốt hơn một chút (0.50), cho thấy mô hình có thể nhận diện một phần lớp này nhưng vẫn còn gặp khó khăn.



Hình :Biểu đồ trực quan hóa của mô hình SVM

### **3.4.5.So sánh và kết luận giữa 4 phương pháp: Hồi quy Logistic, Cây Quyết Định, Rừng Ngẫu Nhiên và SVM.**

**1. Độ chính xác tổng thể (Accuracy):**

* Cây quyết định: 64.52%
* Hồi quy logistic: 80.65% (cao nhất).
* SVM: 70.97%.
* Rừng ngẫu nhiên: 70.97%.

=> Hồi quy logistic đạt độ chính xác cao nhất, nhưng các mô hình SVM và Rừng ngẫu nhiên khá gần nhau.

**2. F1-Score (Trung bình macro):**

F1-Score macro trung bình đo lường sự cân bằng dự đoán giữa các lớp.

* Cây quyết định: 0.55 (thấp nhất).
* Hồi quy logistic: 0.45 (rất thấp do không dự đoán tốt lớp 1).
* SVM: 0.60 (cao nhất).
* Rừng ngẫu nhiên: 0.60 (cao nhất, ngang với SVM).

=> SVM và Rừng ngẫu nhiên đạt cân bằng tốt nhất giữa hai lớp.

**3. Khả năng dự đoán lớp khó (Lớp 1):**

Precision và Recall cho lớp 1 là chỉ số quan trọng để đánh giá khả năng mô hình phát hiện lớp ít phổ biến.

* Cây quyết định: Precision = 0.27, Recall = 0.50, F1 = 0.35.
* Hồi quy logistic: Precision = 1.00, Recall = 0.00, F1 = 0.00 (bỏ qua hoàn toàn lớp 1).
* SVM: Precision = 0.33, Recall = 0.50, F1 = 0.40.
* Rừng ngẫu nhiên: Precision = 0.33, Recall = 0.50, F1 = 0.40.

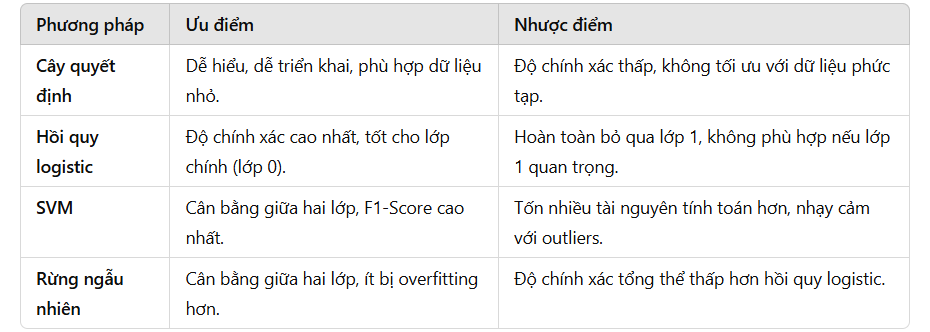
=> SVM và Rừng ngẫu nhiên cân bằng tốt hơn giữa Precision và Recall cho lớp 1. Hồi quy logistic không phù hợp nếu lớp 1 quan trọng.

**4. Độ cân bằng tổng thể giữa hai lớp (Macro Precision và Macro Recall):**

* Cây quyết định: Precision = 0.56, Recall = 0.59.
* Hồi quy logistic: Precision = 0.90, Recall = 0.50 (không cân bằng).
* SVM: Precision = 0.60, Recall = 0.63.
* Rừng ngẫu nhiên: Precision = 0.60, Recall = 0.63.

=> SVM và Rừng ngẫu nhiên tiếp tục dẫn đầu về độ cân bằng giữa hai lớp.

5. **Ưu và nhược điểm từng phương pháp**:



**Kết luận:**

1. **Nếu lớp 1 quan trọng và cần cân bằng giữa các lớp**:
   * SVM và Rừng ngẫu nhiên là lựa chọn tốt nhất.
   * Rừng ngẫu nhiên có thể ổn định hơn nếu dữ liệu lớn hoặc phức tạp.
2. **Nếu chỉ tập trung vào độ chính xác tổng thể và lớp chính (lớp 0)**:
   * Hồi quy logistic là lựa chọn tối ưu nhưng cần cải thiện khả năng dự đoán lớp 1 bằng cách điều chỉnh trọng số (class weight).
3. **Nếu cần giải pháp đơn giản và nhanh chóng**:
   * Cây quyết định phù hợp nhưng hiệu suất thấp hơn các phương pháp khác.

## 3.5.Tìm hiểu mối quan hệ giữa activities và Walc

### **3.5.1.Phân tích mô hình hồi quy (Logistic Regression hoặc Multinomial Logistic Regression)**

# Đọc dữ liệu từ file CSV

data = pd.read\_csv('D:\\python\\BAITAPDULIEU\\DULIEU\\FINAL.csv')

# Chuyển đổi 'Walc' thành 2 lớp (0 và 1)

data['Walc'] = data['Walc'].apply(lambda x: 1 if x > 2 else 0) # Nếu Walc > 2 thì 1, ngược lại 0

# Biến độc lập và phụ thuộc

X = data[['activities']] # Chọn biến độc lập activities

y = data['Walc'] # Chọn biến phụ thuộc Walc (mức độ hài lòng)

# Chia dữ liệu thành tập huấn luyện và kiểm tra (70% huấn luyện, 30% kiểm tra)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Khởi tạo mô hình Logistic Regression với phân loại nhị phân

model = LogisticRegression(solver='lbfgs')

# Huấn luyện mô hình

model.fit(X\_train, y\_train)

# Dự đoán trên tập kiểm tra

y\_pred = model.predict(X\_test)

# Đánh giá mô hình: Hiển thị classification report

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred, zero\_division=1))

# Tính độ chính xác (Accuracy)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f'\nĐộ chính xác của mô hình: {accuracy \* 100:.2f}%')

# Hiển thị hệ số hồi quy và hệ số chặn của mô hình

print("\nHệ số hồi quy (Coefficients):", model.coef\_)

print("\nHệ số chặn (Intercept):", model.intercept\_)

# Tạo lưới điểm cho biến độc lập 'activities' để vẽ đường cong

X\_range = np.linspace(X['activities'].min() - 1, X['activities'].max() + 1, 300).reshape(-1, 1)

# Dự đoán xác suất cho mỗi giá trị của 'activities'

y\_prob = model.predict\_proba(X\_range)[:, 1] # Lấy xác suất cho lớp '1'

# Vẽ biểu đồ phân tán của dữ liệu

plt.figure(figsize=(8, 6))

sns.scatterplot(x=X['activities'], y=y, palette='viridis', s=100, marker='o', label='Dữ liệu thực tế')

# Vẽ đường cong hồi quy logistic (sigmoid curve)

plt.plot(X\_range, y\_prob, color='red', label='Đường cong hồi quy logistic', linewidth=2)

# Thiết lập tiêu đề và nhãn cho các trục

plt.title('Biểu đồ Hồi Quy Logistic: Xác Suất Dự Đoán với Hoạt động (Activities)')

plt.xlabel('Hoạt động (Activities)')

plt.ylabel('Xác suất dự đoán Walc (Mức độ hài lòng)')

# Hiển thị chú giải

plt.legend()

# Hiển thị biểu đồ

plt.show()

# Xuất ma trận nhầm lẫn (Confusion Matrix)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

print("\nMa trận nhầm lẫn:")

print(conf\_matrix)

# Trực quan hóa ma trận nhầm lẫn

plt.figure(figsize=(6, 5))

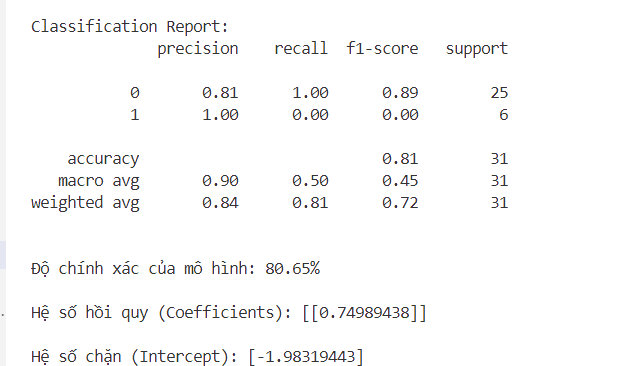
sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Không hài lòng', 'Hài lòng'], yticklabels=['Không hài lòng', 'Hài lòng'])

plt.title('Ma Trận Nhầm Lẫn (Confusion Matrix)')

plt.xlabel('Dự đoán')

plt.ylabel('Thực tế')

plt.show()

**Kết quả và diễn giải:  
**

Hình :Kết quả của hồi quy logistic

**1. Classification Report:**

* **Precision** (độ chính xác) của một lớp là tỷ lệ các dự đoán đúng (True Positives) trên tổng số dự đoán của lớp đó (True Positives + False Positives).
* **Recall** (độ nhớ) của một lớp là tỷ lệ các dự đoán đúng trên tổng số thực tế thuộc lớp đó (True Positives + False Negatives).
* **F1-score** là trung bình điều hòa của Precision và Recall, là một chỉ số tổng hợp để đánh giá mô hình phân loại.

**Chi tiết cho các lớp**:

* **Lớp 0 (tiêu thụ rượu thấp)**:
  + **Precision** = 0.81: 81% các dự đoán của mô hình về lớp này là đúng.
  + **Recall** = 1.00: Mô hình nhận diện tất cả các thực tế thuộc lớp 0 (Không hài lòng).
  + **F1-score** = 0.89: Điểm F1 của lớp này là khá cao, chứng tỏ mô hình có khả năng nhận diện lớp này rất tốt.
* **Lớp 1 (tiêu thụ rượu cao)**:
  + **Precision** = 1.00: Mô hình dự đoán chính xác 100% các trường hợp lớp này, tức là tất cả các dự đoán về lớp 1 đều đúng.
  + **Recall** = 0.00: Tuy nhiên, mô hình không thể nhận diện được bất kỳ trường hợp nào thực sự thuộc lớp 1, vì không có dự đoán đúng cho lớp này.
  + **F1-score** = 0.00: Điểm F1 rất thấp vì Recall bằng 0.

**Tổng kết:**

* Mô hình có độ chính xác cao đối với lớp 0 (không hài lòng) nhưng hoàn toàn không nhận diện được lớp 1 (hài lòng). Điều này cho thấy rằng mô hình đang nghiêng về việc dự đoán lớp 0 và bỏ qua lớp 1 hoàn toàn.

**2. Accuracy (Độ chính xác):**

* **Độ chính xác của mô hình** = 80.65%: Mô hình đúng 80.65% tổng số trường hợp trong tập kiểm tra.

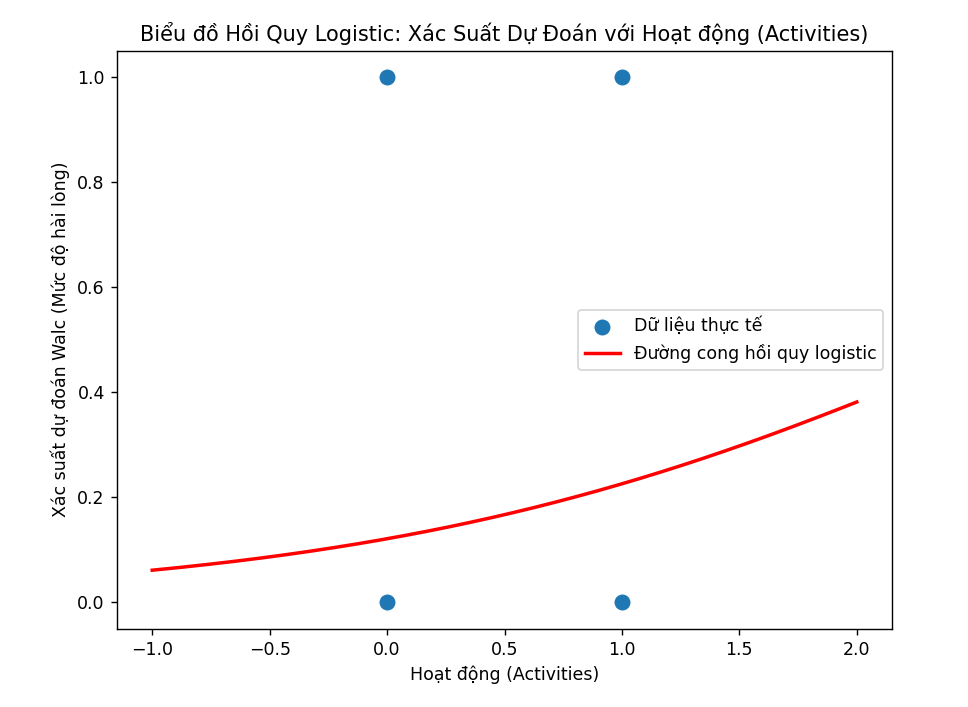
Tuy nhiên, độ chính xác này có thể gây hiểu lầm vì mô hình chỉ dự đoán chính xác lớp 0 mà không dự đoán được lớp 1. Vì vậy, mặc dù độ chính xác khá cao, nhưng mô hình không hoạt động hiệu quả với tất cả các lớp.

**3. Macro Average và Weighted Average:**

* **Macro avg**:
  + **Precision** = 0.90, **Recall** = 0.50, **F1-score** = 0.45: Đây là các chỉ số trung bình không trọng số, tính toán cho cả hai lớp (0 và 1). Precision cao (0.90) cho thấy mô hình mạnh trong việc dự đoán đúng lớp 0, nhưng Recall và F1-score thấp vì mô hình không nhận diện được lớp 1.
* **Weighted avg**:
  + **Precision** = 0.84, **Recall** = 0.81, **F1-score** = 0.72: Các chỉ số này tính trọng số cho mỗi lớp, với lớp 0 chiếm ưu thế vì có nhiều mẫu hơn. Mô hình có F1-score 0.72, cho thấy sự tổng quát tốt hơn nếu xét cả hai lớp.

**4. Hệ số hồi quy (Coefficients) và Hệ số chặn (Intercept):**

* **Hệ số hồi quy** = 0.7499: Hệ số này cho biết mức độ ảnh hưởng của biến độc lập activities (hoạt động) đối với xác suất dự đoán lớp 1 (hài lòng). Hệ số dương này cho thấy khi activities tăng lên, xác suất dự đoán lớp 1 (hài lòng) cũng tăng.
* **Hệ số chặn** = -1.9832: Hệ số chặn (Intercept) là giá trị mà mô hình sử dụng khi không có thông tin về activities. Đây là giá trị xác suất đối với lớp 0 (không hài lòng) khi activities = 0.



Hình :Kết quả trực quan biểu đồ hồi quy logistic

### **3.5.2.Phân tích sử dụng Cây quyết định (Decision Tree)**

**# Đọc dữ liệu từ file CSV**

**data = pd.read\_csv('D:\\python\\BAITAPDULIEU\\DULIEU\\FINAL.csv')**

**# Chuyển đổi 'Walc' thành 2 lớp: 0 là tiêu thụ rượu thấp, 1 là tiêu thụ rượu cao**

**data['Walc\_binary'] = data['Walc'].apply(lambda x: 1 if x > 2 else 0)**

**# Biến độc lập và phụ thuộc**

**X = data[['activities']] # Biến độc lập là activities**

**y = data['Walc\_binary'] # Biến phụ thuộc là Walc\_binary**

**# Chia dữ liệu thành tập huấn luyện và kiểm tra**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)**

**# Tạo và huấn luyện mô hình cây quyết định**

**model = DecisionTreeClassifier(criterion='entropy', random\_state=42, class\_weight='balanced')**

**model.fit(X\_train, y\_train)**

**# Dự đoán trên tập kiểm tra**

**y\_pred = model.predict(X\_test)**

**# Đánh giá mô hình**

**print("\nClassification Report:")**

**print(classification\_report(y\_test, y\_pred))**

**print("\nConfusion Matrix:")**

**print(confusion\_matrix(y\_test, y\_pred))**

**# Tính toán và in độ chính xác**

**accuracy = accuracy\_score(y\_test, y\_pred)**

**print(f"\nĐộ chính xác của mô hình là: {accuracy \* 100:.2f}%")**

**# Vẽ biểu đồ cây quyết định**

**plt.figure(figsize=(12, 8))**

**plot\_tree(model, feature\_names=['activities'], class\_names=['Tiêu thụ rượu thấp', 'Tiêu thụ rượu cao'],**

**filled=True, rounded=True, fontsize=10)**

**plt.title('Decision Tree for Activities vs Walc\_binary')**

**plt.show()**

**# Tạo lưới các giá trị của activities từ min đến max**

**x\_min, x\_max = X['activities'].min() - 1, X['activities'].max() + 1**

**xx = np.linspace(x\_min, x\_max, num=100).reshape(-1, 1)**

**# Chuyển đổi xx thành DataFrame**

**xx\_df = pd.DataFrame(xx, columns=['activities'])**

**# Dự đoán ranh giới quyết định**

**pred\_boundary = model.predict(xx\_df)**

**# Vẽ sơ đồ đường quyết định**

**plt.figure(figsize=(10, 6))**

**plt.plot(xx, pred\_boundary, color='red', label='Decision Boundary')**

**plt.scatter(X\_train, y\_train, c='blue', marker='o', label='Train Data')**

**plt.scatter(X\_test, y\_test, c='green', marker='x', label='Test Data')**

**plt.title('Decision Boundary for Activities vs Walc\_binary')**

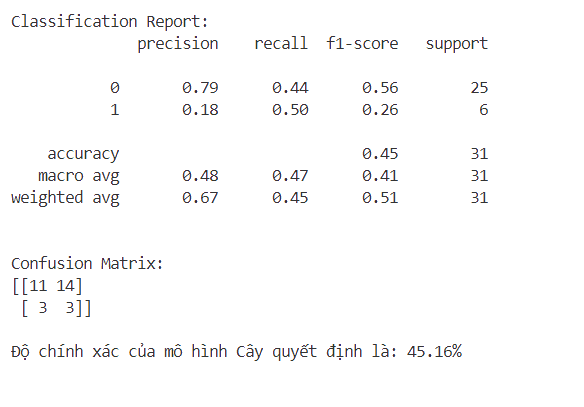
**plt.xlabel('Activities')**

**plt.ylabel('Walc\_binary (Tiêu thụ rượu thấp/cao)')**

**plt.legend()**

**plt.show()**

**Kết quả và diễn giải:**

****

Hình :Kết quả của mô hình cây quyết định

**1.Classification Report**

* **Precision**: Đây là tỷ lệ dự đoán chính xác đối với các mẫu được phân loại vào từng lớp.
  + **Precision cho lớp 0 (Tiêu thụ rượu thấp)**: 0.79 (79%)
    - Điều này có nghĩa là trong số các mẫu được dự đoán là "Tiêu thụ rượu thấp", 79% là chính xác.
  + **Precision cho lớp 1 (Tiêu thụ rượu cao)**: 0.18 (18%)
    - Điều này có nghĩa là trong số các mẫu được dự đoán là "Tiêu thụ rượu cao", chỉ có 18% là chính xác. Precision thấp cho lớp này là một vấn đề lớn, cho thấy mô hình có xu hướng dự đoán không chính xác đối với lớp này.
* **Recall**: Đây là tỷ lệ mẫu đúng bị mô hình phân loại đúng, so với tổng số mẫu thực tế trong lớp đó.
  + **Recall cho lớp 0 (Tiêu thụ rượu thấp)**: 0.44 (44%)
    - Điều này có nghĩa là chỉ có 44% số mẫu thực tế thuộc lớp 0 được mô hình phân loại đúng.
  + **Recall cho lớp 1 (Tiêu thụ rượu cao)**: 0.50 (50%)
    - Điều này có nghĩa là 50% số mẫu thực tế thuộc lớp 1 được phân loại đúng, cho thấy mô hình có thể nhận diện lớp này khá tốt hơn so với lớp 0.
* **F1-Score**: Là trung bình điều hòa của Precision và Recall.
  + **F1-Score cho lớp 0**: 0.56
    - Mức độ hài hòa giữa Precision và Recall cho lớp này là 56%, cho thấy mô hình hoạt động khá kém đối với lớp này.
  + **F1-Score cho lớp 1**: 0.26
    - Mức độ hài hòa giữa Precision và Recall cho lớp 1 rất thấp, chỉ có 26%, điều này chứng tỏ mô hình không nhận diện tốt lớp này.
* **Macro Average**: Trung bình không trọng số của Precision, Recall và F1-Score.
  + **Macro Average Precision**: 0.48
    - Precision trung bình cho tất cả các lớp là 48%, cho thấy mô hình vẫn không đạt được độ chính xác cao đối với cả hai lớp.
  + **Macro Average Recall**: 0.47
    - Recall trung bình cho tất cả các lớp là 47%, chỉ đạt mức trung bình.
  + **Macro Average F1-Score**: 0.41
    - F1-Score trung bình cho tất cả các lớp là 41%, đây là một chỉ số khá thấp.
* **Weighted Average**: Trung bình có trọng số của Precision, Recall và F1-Score, tính đến số lượng mẫu trong mỗi lớp.
  + **Weighted Average Precision**: 0.67
    - Do lớp 0 có số lượng mẫu lớn hơn lớp 1, mô hình cho kết quả chính xác cao hơn đối với lớp này, làm tăng Precision trung bình.
  + **Weighted Average Recall**: 0.45
    - Recall trung bình có trọng số thấp hơn một chút do lớp 1 có Recall tốt hơn lớp 0.
  + **Weighted Average F1-Score**: 0.51
    - F1-Score có trọng số khá thấp, phản ánh mô hình vẫn chưa đủ mạnh để đạt được kết quả tốt trên cả hai lớp.

**2. Confusion Matrix**

[[11 4]]

[[3 3]]

**Lớp 0 (Tiêu thụ rượu thấp)**:

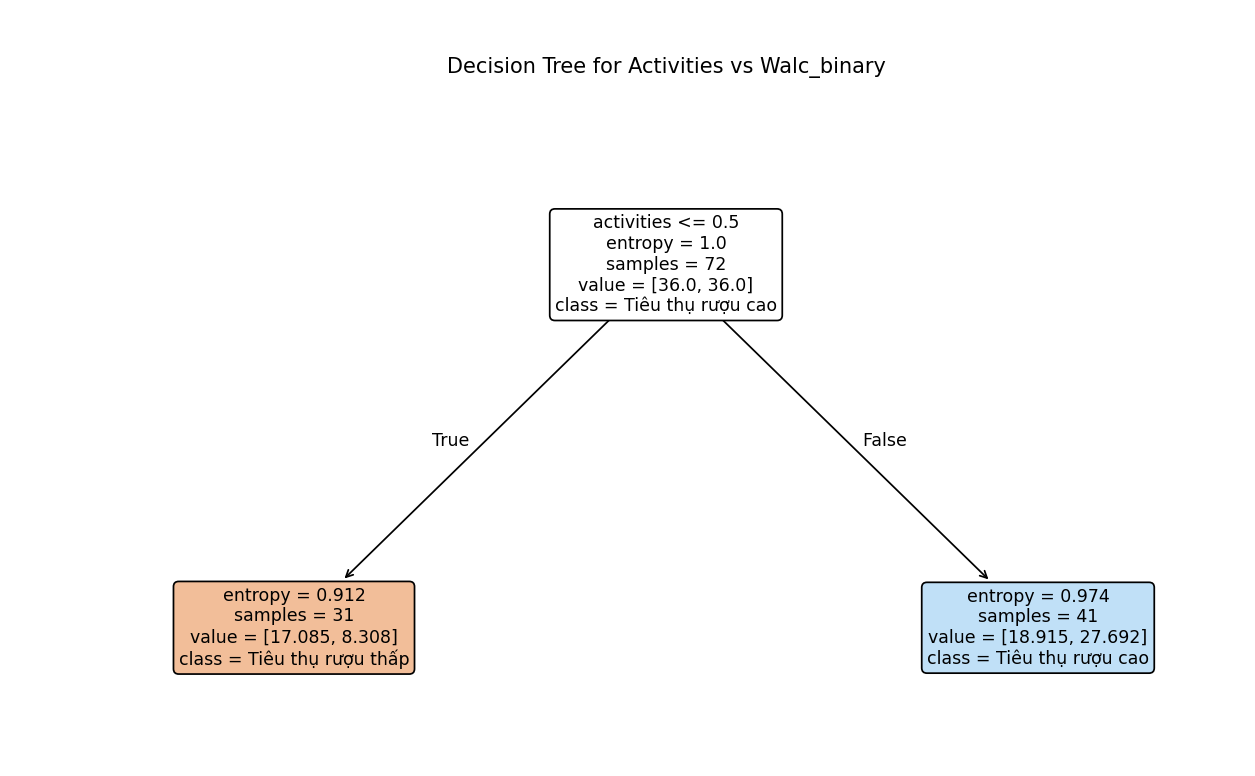
* Có 11 mẫu đúng được phân loại vào lớp này.
* Có 4 mẫu sai được phân loại thành lớp 1 (Tiêu thụ rượu cao), dẫn đến tỷ lệ Recall thấp cho lớp 0.

**Lớp 1 (Tiêu thụ rượu cao)**:

* Có 3 mẫu đúng được phân loại vào lớp này.
* Có 3 mẫu sai được phân loại vào lớp 0 (Tiêu thụ rượu thấp), nhưng do số lượng mẫu của lớp này rất nhỏ, tỷ lệ Precision và Recall cho lớp 1 đều thấp.

**3. Độ chính xác (Accuracy)**

* **Độ chính xác của mô hình**: 45.16%
  + Mô hình chỉ đúng khoảng 45% số mẫu, cho thấy độ chính xác thấp.
  + Điều này có thể là do sự không cân đối trong phân phối lớp (lớp 0 có số lượng mẫu lớn hơn nhiều so với lớp 1).



Hình :Trực quan hóa cây quyết định

**Ý nghĩa các thông số**

* **Entropy**:
  + Là độ đo mức độ hỗn loạn hoặc không chắc chắn của dữ liệu tại một nút.
  + Nút gốc (Entropy = 1.0): Phân phối dữ liệu đồng đều giữa hai lớp, tức là tại đây mô hình chưa phân biệt được lớp nào chiếm ưu thế.
  + Nhánh trái (Entropy = 0.912): Dữ liệu đã rõ ràng hơn, nhưng vẫn còn một số nhầm lẫn giữa hai lớp.
  + Nhánh phải (Entropy = 0.974): Nhánh này vẫn chứa nhiều sự không chắc chắn.
* **Value**:
  + Đại diện cho phân phối dữ liệu trong từng lớp tại mỗi nút.
  + Ví dụ:
    - **Nút gốc**: Value = [36.0, 36.0] => Dữ liệu chia đều giữa hai lớp.
    - **Nhánh trái**: Value = [17.085, 8.308] => Số lượng mẫu thuộc lớp Tiêu thụ rượu thấp lớn hơn hẳn lớp Tiêu thụ rượu cao.
    - **Nhánh phải**: Value = [18.915, 27.692] => Lớp Tiêu thụ rượu cao chiếm ưu thế.

**Phân tích và đánh giá**

* **Cây quyết định đơn giản**:
  + Cây này chỉ có 2 tầng, điều này cho thấy mô hình đã dừng lại sớm khi phân tách dữ liệu.
  + Mô hình phân chia dữ liệu dựa vào giá trị của biến activities nhưng chưa hoàn toàn tối ưu hóa việc tách lớp.
* **Vấn đề chính**:
  + **Entropy vẫn cao** tại các nhánh con (0.912 và 0.974), cho thấy dữ liệu tại đây vẫn chưa được phân chia rõ ràng. Điều này dẫn đến việc dự đoán không chính xác.
  + **Mất cân đối lớp**:
    - Nhánh trái có nhiều mẫu thuộc lớp Tiêu thụ rượu thấp.
    - Nhánh phải có nhiều mẫu thuộc lớp Tiêu thụ rượu cao.

### **3.5.3.Rừng Ngẫu Nhiên (Random Forest)**

# Đọc dữ liệu từ file CSV

data = pd.read\_csv('D:\\python\\BAITAPDULIEU\\DULIEU\\FINAL.csv')

# Chuyển đổi 'Walc' thành 2 lớp: 0 là tiêu thụ rượu thấp, 1 là tiêu thụ rượu cao

data['Walc\_binary'] = data['Walc'].apply(lambda x: 1 if x > 2 else 0)

# Biến độc lập và phụ thuộc

X = data[['activities']] # Biến độc lập là activities

y = data['Walc\_binary'] # Biến phụ thuộc là Walc\_binary

# Chia dữ liệu thành tập huấn luyện và kiểm tra

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Tạo và huấn luyện mô hình Rừng Ngẫu Nhiên (Random Forest)

model = RandomForestClassifier(n\_estimators=100, random\_state=42, class\_weight='balanced')

model.fit(X\_train, y\_train)

# Dự đoán trên tập kiểm tra

y\_pred = model.predict(X\_test)

# Đánh giá mô hình

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred))

print("\nConfusion Matrix:")

print(confusion\_matrix(y\_test, y\_pred))

# Tính toán và in độ chính xác

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"\nĐộ chính xác của mô hình là: {accuracy \* 100:.2f}%")

# Hiển thị đặc trưng quan trọng (Feature Importance)

features = model.feature\_importances\_

print("\nĐặc trưng quan trọng (Feature Importances):", features)

# Vẽ đồ thị đặc trưng quan trọng

plt.figure(figsize=(8, 6))

plt.barh(['activities'], features)

plt.title('Feature Importance')

plt.xlabel('Feature Importance')

plt.show()

# Vẽ sơ đồ quyết định của Rừng Ngẫu Nhiên

plt.figure(figsize=(10, 6))

x\_min, x\_max = X['activities'].min() - 1, X['activities'].max() + 1 # Đổi từ 'goout' thành 'activities'

xx = np.linspace(x\_min, x\_max, num=100).reshape(-1, 1)

# Dự đoán trên lưới giá trị của 'activities'

pred\_boundary = model.predict(xx)

# Vẽ sơ đồ đường quyết định

plt.plot(xx, pred\_boundary, color='red', label='Decision Boundary')

plt.scatter(X\_train, y\_train, c='blue', marker='o', label='Train Data')

plt.scatter(X\_test, y\_test, c='green', marker='x', label='Test Data')

plt.title('Decision Boundary for Activities vs Walc\_binary with Random Forest')

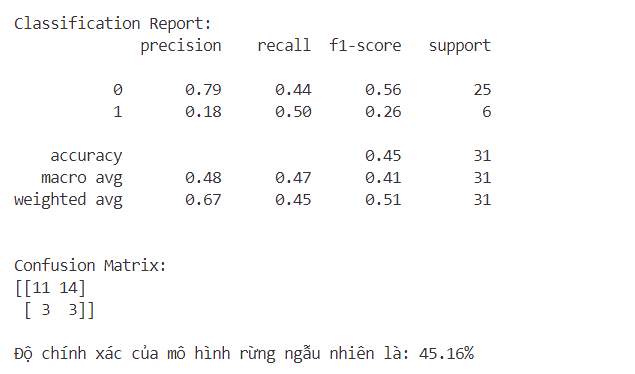
plt.xlabel('Activities')

plt.ylabel('Walc\_binary (Tiêu thụ rượu thấp/cao)')

plt.legend()

plt.show()

**Kết quả và diễn giải:**

****

Hình :Kết quả của mô hình ngẫu nhiên

**1. Classification Report:**

* Precision:
  + Cho lớp 0 (Tiêu thụ rượu thấp): Precision là 0.79, điều này có nghĩa là khi mô hình dự đoán lớp 0 (rượu thấp), thì 79% trong số đó thực sự là lớp 0.
  + Cho lớp 1 (Tiêu thụ rượu cao): Precision là 0.18, điều này cho thấy khi mô hình dự đoán lớp 1 (rượu cao), chỉ có 18% trong số đó thực sự là lớp 1.
* Recall:
  + Cho lớp 0 (Tiêu thụ rượu thấp): Recall là 0.44, có nghĩa là mô hình chỉ nhận diện được 44% các trường hợp thực sự thuộc lớp 0.
  + Cho lớp 1 (Tiêu thụ rượu cao): Recall là 0.50, nghĩa là mô hình nhận diện được 50% các trường hợp thực sự thuộc lớp 1.
* F1-Score:
  + F1-Score cho lớp 0 là 0.56, thể hiện một sự cân bằng giữa Precision và Recall, nhưng không quá cao, chỉ ra rằng mô hình vẫn còn nhiều thiếu sót khi dự đoán lớp này.
  + F1-Score cho lớp 1 là 0.26, khá thấp, cho thấy mô hình không thể dự đoán chính xác lớp 1 (Tiêu thụ rượu cao), đặc biệt khi Precision cũng rất thấp.
* Accuracy:
  + Độ chính xác chung là 45.16%, tức là khoảng 45% dự đoán của mô hình là chính xác. Tuy nhiên, con số này không phải là một dấu hiệu tốt vì nó có thể là kết quả của một sự thiên lệch trong dữ liệu (ví dụ như một lớp chiếm ưu thế hơn).
* Macro Average:
  + Precision: 0.48, Recall: 0.47, F1-Score: 0.41. Đây là trung bình của Precision, Recall và F1-Score cho cả hai lớp. Số liệu này cho thấy mô hình có hiệu suất kém, đặc biệt đối với lớp 1.
* Weighted Average:
  + Đây là sự tính toán trung bình có trọng số theo số lượng mẫu của từng lớp. Mặc dù Precision có vẻ ổn đối với lớp 0 (Tiêu thụ rượu thấp), nhưng Recall và F1-Score vẫn cho thấy mô hình không hiệu quả trong việc phân loại chính xác lớp 1.

**2. Confusion Matrix:**

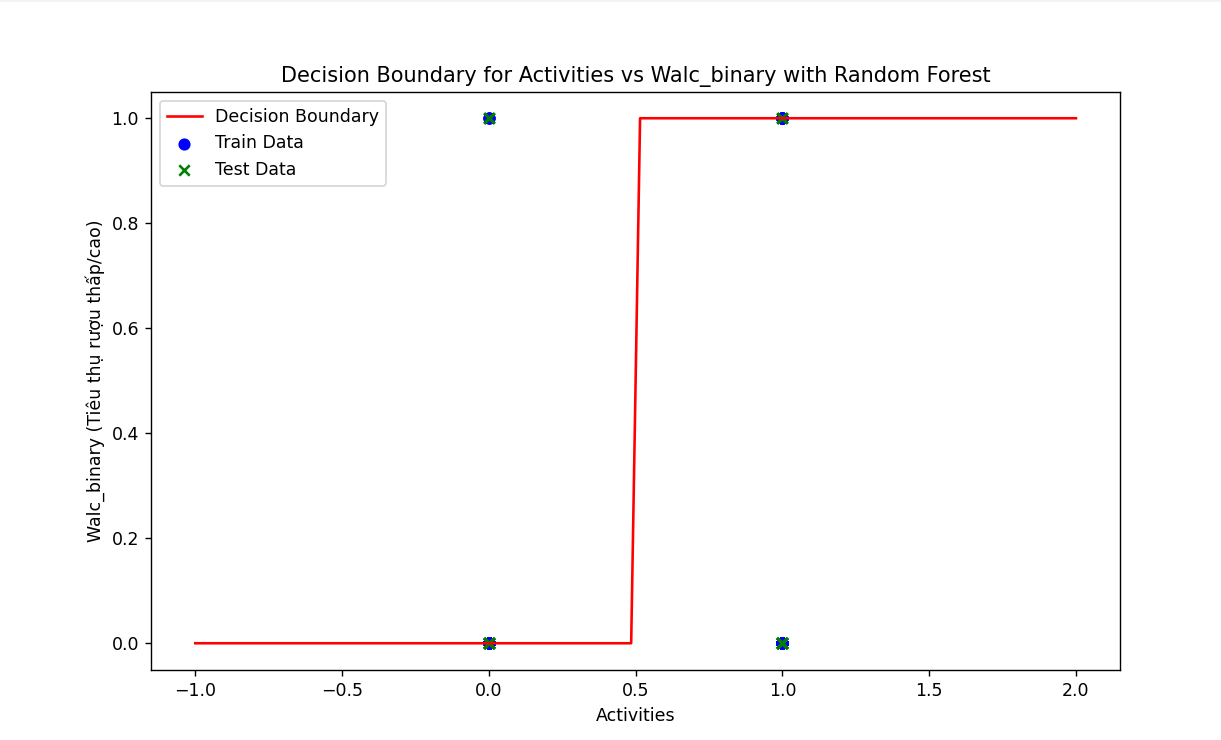
[[11 14]

[ 3 3]]

* True Negative (TN): 11 mẫu được phân loại đúng là lớp 0 (tiêu thụ rượu thấp).
* False Positive (FP): 14 mẫu bị phân loại sai là lớp 0 khi thực tế là lớp 1 (tiêu thụ rượu cao). Đây là một chỉ số quan trọng cho thấy mô hình thường xuyên nhầm lẫn với lớp 0 khi thực tế là lớp 1.
* False Negative (FN): 3 mẫu bị phân loại sai là lớp 1 khi thực tế là lớp 0. Mặc dù số lượng này ít, nhưng nó vẫn cho thấy mô hình chưa hoàn hảo.
* True Positive (TP): 3 mẫu được phân loại chính xác là lớp 1 (tiêu thụ rượu cao).

**3. Đánh giá chung:**

* Mô hình Rừng Ngẫu Nhiên có độ chính xác khá thấp, chỉ 45.16%. Điều này có thể là do sự không cân bằng trong dữ liệu, trong đó lớp 0 (tiêu thụ rượu thấp) có nhiều mẫu hơn lớp 1 (tiêu thụ rượu cao).
* Precision và Recall cho lớp 1 rất thấp, điều này cho thấy mô hình gặp khó khăn trong việc nhận diện lớp này, có thể vì lớp này hiếm gặp hơn trong dữ liệu huấn luyện.
* Feature Importance của biến activities có thể giúp chỉ ra tầm quan trọng của đặc trưng này trong quá trình phân loại, nhưng mô hình vẫn cần cải thiện để tối ưu hóa hiệu quả phân loại của lớp 1.

****

Hình :Biểu đồ trực quan hóa của mô hình rừng ngẫu nhiên

### **3.5.4.Phân tích sử dụng Máy vector hỗ trợ (Support Vector Machine)**

X = data[['activities']] # Chọn biến độc lập activities

y = data['Walc\_binary'] # Chọn biến phụ thuộc Walc\_binary

# Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra (70% huấn luyện, 30% kiểm tra)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Huấn luyện mô hình SVM

model = SVC(kernel='linear', class\_weight='balanced', random\_state=42)

model.fit(X\_train, y\_train)

# Dự đoán trên tập kiểm tra

y\_pred = model.predict(X\_test)

# Ma trận nhầm lẫn

print("Confusion Matrix:")

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

print(conf\_matrix)

# Báo cáo phân loại

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred))

# In ra độ chính xác của mô hình

from sklearn.metrics import accuracy\_score

accuracy = accuracy\_score(y\_test, y\_pred)

print("\nĐộ chính xác của mô hình SVM: {:.2f}%".format(accuracy \* 100))

# Vẽ biểu đồ trực quan hóa biên quyết định

# Tạo lưới để vẽ biên phân cách

xx = np.linspace(X['activities'].min() - 0.1, X['activities'].max() + 0.1, 100).reshape(-1, 1)

# Dự đoán nhãn cho từng điểm trên lưới

Z = model.decision\_function(xx)

# Vẽ biểu đồ

plt.figure(figsize=(8, 6))

plt.scatter(X['activities'], y, color='blue', s=100, marker='o', label='Dữ liệu thực tế')

plt.plot(xx, Z, color='red', label='Biên quyết định', linewidth=2)

# Thiết lập tiêu đề và nhãn trục

plt.title('Biểu đồ phân loại với SVM')

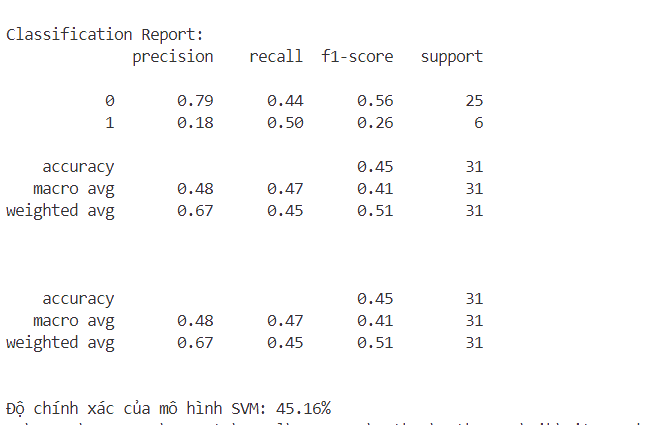
plt.xlabel('Hoạt động (Activities)')

plt.ylabel('Tiêu thụ rượu vào buổi chiều (Walc)')

plt.legend()

plt.show()

**Kết quả và diễn giải:**

****

Hình : Kết quả từ mô hình SVM

**1. Classification Report:**

* Precision: Đây là tỷ lệ các dự đoán dương tính đúng trong tất cả các dự đoán dương tính.
  + Lớp 0 (tiêu thụ rượu thấp) có precision là 0.79, tức là 79% các dự đoán rằng người ta có tiêu thụ rượu thấp là đúng.
  + Lớp 1 (tiêu thụ rượu cao) có precision là 0.18, tức là chỉ có 18% các dự đoán rằng người ta có tiêu thụ rượu cao là đúng. Đây là một chỉ số thấp, cho thấy mô hình không đủ mạnh để phân loại chính xác lớp này.
* Recall: Đây là tỷ lệ các trường hợp dương tính thực sự đã được mô hình nhận diện chính xác.
  + Lớp 0 có recall là 0.44, tức là mô hình chỉ nhận diện được 44% các trường hợp tiêu thụ rượu thấp thực sự.
  + Lớp 1 có recall là 0.50, tức là mô hình nhận diện được 50% các trường hợp tiêu thụ rượu cao thực sự.
* F1-Score: Đây là trung bình điều hòa giữa precision và recall, giúp đánh giá mô hình với sự cân bằng giữa hai chỉ số này.
  + Lớp 0 có F1-Score là 0.56, chỉ ra rằng độ chính xác và khả năng phát hiện của mô hình ở lớp này khá tốt, nhưng vẫn có thể cải thiện.
  + Lớp 1 có F1-Score là 0.26, thấp, phản ánh rằng mô hình chưa thể phân loại tốt lớp này.

**2. Accuracy:**

* Độ chính xác của mô hình (accuracy) là 45.16%, nghĩa là khoảng 45.16% dự đoán của mô hình là chính xác. Tuy nhiên, chỉ số này không phải là một chỉ số tốt trong trường hợp có sự mất cân bằng giữa các lớp (class imbalance), như trong trường hợp này, khi có ít hơn các mẫu cho lớp 1 (tiêu thụ rượu cao).

**3. Macro Average và Weighted Average:**

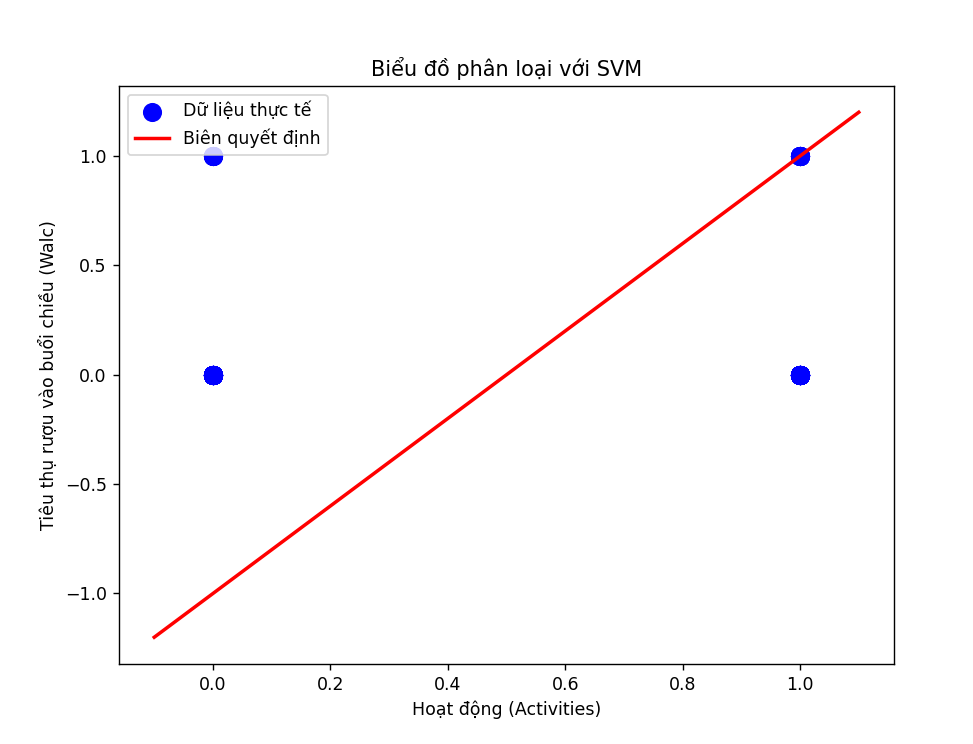
* Macro Average: Trung bình cộng giữa các chỉ số precision, recall, và F1-Score của các lớp, không cân nhắc tỉ lệ của mỗi lớp. Trong trường hợp này, macro average cho thấy mô hình không đạt được sự cân bằng giữa các lớp:
  + Precision là 0.48.
  + Recall là 0.47.
  + F1-Score là 0.41.
* Weighted Average: Trung bình có trọng số, tính đến tỉ lệ của mỗi lớp. Đây là chỉ số tổng hợp đánh giá mô hình với sự cân nhắc đến số lượng mẫu của từng lớp.
  + Weighted average cho thấy mô hình có sự ưu tiên hơn đối với lớp 0 vì có số lượng mẫu lớn hơn.

**4. Confusion Matrix:**

[[11 14]

[ 3 3]]

* + 11 là số trường hợp mà mô hình dự đoán chính xác lớp 0 là lớp 0.
  + 14 là số trường hợp mà mô hình dự đoán sai lớp 0 thành lớp 1.
  + 3 là số trường hợp mà mô hình dự đoán sai lớp 1 thành lớp 0.
  + 3 là số trường hợp mà mô hình dự đoán chính xác lớp 1 là lớp 1.

****

Hình :Biểu đồ trực quan hóa của mô hình SVM

### **3.5.5.So sánh và kết luận giữa 4 phương pháp: Hồi quy Logistic, Cây Quyết Định, Rừng Ngẫu Nhiên và SVM.**

**1. Logistic Regression**

* **Precision**:
  + Lớp 0: **0.81** (cao, dự đoán tốt lớp tiêu thụ rượu thấp).
  + Lớp 1: **1.00** (tuyệt đối nhưng mất cân bằng vì Recall của lớp này bằng 0).
* **Recall**:
  + Lớp 0: **1.00** (tất cả các mẫu thực sự thuộc lớp 0 đều được nhận diện chính xác).
  + Lớp 1: **0.00** (không nhận diện được lớp 1).
* **F1-Score**:
  + Lớp 0: **0.89** (rất tốt).
  + Lớp 1: **0.00** (không hiệu quả).
* **Accuracy**: **80.65%** (cao nhất trong các mô hình nhưng không cân bằng, bỏ qua lớp 1).
* **Nhận xét**: Mô hình Logistic Regression thiên về lớp 0 và không nhận diện được lớp 1 (tiêu thụ rượu cao).

**2. SVM**

* **Precision**:
  + Lớp 0: **0.79** (tương đối tốt).
  + Lớp 1: **0.18** (rất thấp, mô hình không dự đoán chính xác lớp 1).
* **Recall**:
  + Lớp 0: **0.44** (chỉ nhận diện được 44% mẫu thực tế thuộc lớp 0).
  + Lớp 1: **0.50** (nhận diện được 50% mẫu thực tế thuộc lớp 1).
* **F1-Score**:
  + Lớp 0: **0.56** (trung bình).
  + Lớp 1: **0.26** (rất thấp).
* **Accuracy**: **45.16%** (thấp).
* **Nhận xét**: Mô hình SVM hoạt động kém cân bằng và hiệu quả tổng thể thấp.

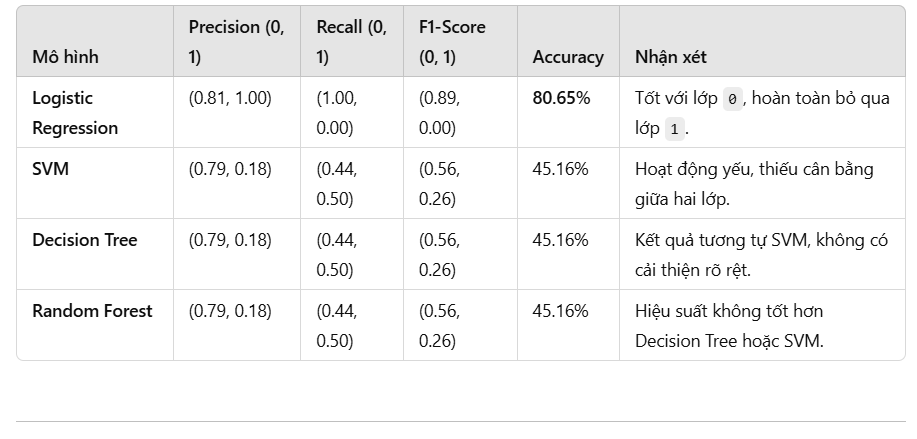
**3. Decision Tree**

* **Precision**:
  + Lớp 0: **0.79** (giống SVM, dự đoán tương đối chính xác lớp 0).
  + Lớp 1: **0.18** (thấp, dự đoán kém chính xác lớp 1).
* **Recall**:
  + Lớp 0: **0.44**.
  + Lớp 1: **0.50**.
* **F1-Score**:
  + Lớp 0: **0.56**.
  + Lớp 1: **0.26**.
* **Accuracy**: **45.16%** (tương đương SVM).
* **Nhận xét**: Mô hình Decision Tree có kết quả tương tự SVM, không cải thiện nhiều cho lớp 1.

**4. Random Forest**

* **Precision**:
  + Lớp 0: **0.79**.
  + Lớp 1: **0.18**.
* **Recall**:
  + Lớp 0: **0.44**.
  + Lớp 1: **0.50**.
* **F1-Score**:
  + Lớp 0: **0.56**.
  + Lớp 1: **0.26**.
* **Accuracy**: **45.16%** (giống Decision Tree và SVM).
* **Nhận xét**: Random Forest không mang lại sự cải thiện so với Decision Tree.

**So sánh tổng quát**



**Kết luận**

1. **Mô hình Logistic Regression**:
   * Có độ chính xác cao nhất (**80.65%**) nhưng không dự đoán được lớp 1. Điều này làm cho mô hình không phù hợp nếu lớp 1 quan trọng.
2. **Các mô hình SVM, Decision Tree, và Random Forest**:
   * Các mô hình này đều có hiệu suất thấp hơn (**45.16% accuracy**) và không đủ mạnh để phân loại tốt lớp 1 (precision thấp).
3. **Hạn chế chính**:
   * Dữ liệu có sự **mất cân bằng lớp**: Lớp 0 có nhiều mẫu hơn lớp 1, dẫn đến việc các mô hình thiên về lớp chiếm ưu thế.
   * **Precision và Recall thấp** cho lớp 1, đặc biệt là với các mô hình SVM, Decision Tree, và Random Forest.

## 3.6.Kết luận từ các mối quan hệ

Qua các mối quan hệ đã phân tích trên và đã áp dụng một số các thuật toán ta rút ra được như sau:

* Nếu chỉ tập trung vào lớp 0 (tiêu thụ rượu thấp):
  + Logistic Regression là lựa chọn tối ưu do độ chính xác và F1-Score cao với lớp 0.
* Nếu muốn cân bằng giữa hai lớp và cải thiện khả năng nhận diện lớp 1 (tiêu thụ rượu cao):
  + SVM hoặc Random Forest là lựa chọn phù hợp hơn vì chúng có khả năng nhận diện lớp 1 (Recall = 0.50), mặc dù Precision thấp.

**Đề xuất cuối cùng**

* Sử dụng Logistic Regression nếu mục tiêu chính là nhận diện chính xác lớp 0 và bạn không cần quan tâm đến lớp 1.
* Sử dụng Random Forest nếu bạn cần cân bằng hơn giữa các lớp và muốn cải thiện khả năng nhận diện lớp 1.

=>Do bài toán phân tích giữa mức độ tiêu thụ cao và thấp nên thuật toán tối ưu nhất để chọn là Random Forest

# CHƯƠNG 4: XÂY DỰNG MÔ HÌNH DỰ ĐOÁN MỨC ĐỘ TIÊU THỤ RƯỢU ẢNH HƯỞNG

Dựa trên các thuật toán phân tích nhóm, em đã chọn mô hình Rừng Ngẫu Nhiên (Random Forest) hoặc Máy Véc-tơ Hỗ trợ (SVM) để xây dựng mô hình dự đoán mức độ tiêu thụ rượu. Cả hai mô hình này đều là những thuật toán học máy mạnh mẽ và phổ biến trong phân tích dữ liệu, đặc biệt là trong các bài toán phân loại.

Mô hình Rừng Ngẫu Nhiên sử dụng tập hợp các cây quyết định (decision trees) được huấn luyện trên các mẫu dữ liệu khác nhau và kết hợp kết quả dự đoán từ các cây này để đưa ra dự đoán cuối cùng. Thuật toán này có khả năng xử lý tốt các dữ liệu có tính chất không tuyến tính và có thể tránh hiện tượng overfitting (quá khớp) nhờ vào sự kết hợp của nhiều cây quyết định.

Trong khi đó, SVM là một thuật toán phân loại mạnh mẽ, đặc biệt hữu ích trong các bài toán phân loại với không gian dữ liệu có chiều cao hoặc có sự phân tách rõ ràng giữa các lớp. SVM tìm kiếm một siêu phẳng tối ưu (hyperplane) để phân loại các điểm dữ liệu sao cho khoảng cách giữa các điểm của các lớp khác nhau là lớn nhất.

Trong mô hình này, các yếu tố như số ngày vắng mặt, hoạt động ngoại khóa, tần suất ra ngoài và thời gian rảnh rỗi được sử dụng làm các đặc trưng đầu vào (features), trong khi mức độ tiêu thụ rượu (Walc) là mục tiêu cần dự đoán. Việc áp dụng mô hình Rừng Ngẫu Nhiên hoặc SVM cho phép phân tích và dự đoán mức độ tiêu thụ rượu của học sinh hoặc sinh viên dựa trên các yếu tố xã hội và tâm lý. Mô hình này sẽ được huấn luyện và đánh giá với tập dữ liệu thực tế để đưa ra dự đoán chính xác về mức độ tiêu thụ rượu, từ đó hỗ trợ trong việc hiểu rõ hơn về hành vi và thói quen của nhóm đối tượng nghiên cứu.

## 1.Mô hình sử dụng để dự đoán mức độ tiêu thụ rượu ảnh hưởng

Trong bài toán dự đoán mức độ tiêu thụ rượu, chúng tôi sử dụng hai mô hình học máy phổ biến: Rừng Ngẫu Nhiên (Random Forest) và Máy Véc-tơ Hỗ trợ (SVM). Cả hai mô hình này đều có những đặc điểm riêng biệt, mang lại ưu điểm trong các bài toán phân loại và dự đoán, nhưng lại phù hợp với các loại dữ liệu và mục tiêu khác nhau.

* **Mô hình Rừng Ngẫu Nhiên (Random Forest)**

Rừng Ngẫu Nhiên là một mô hình học máy mạnh mẽ, dựa trên thuật toán Bagging (Bootstrap Aggregating). Nó xây dựng một tập hợp các cây quyết định (decision trees) từ những mẫu dữ liệu khác nhau và kết hợp kết quả dự đoán của các cây này để đưa ra dự đoán cuối cùng. Ưu điểm lớn nhất của mô hình Rừng Ngẫu Nhiên là khả năng giảm thiểu overfitting (quá khớp) nhờ vào việc sử dụng nhiều cây quyết định và kỹ thuật phân chia ngẫu nhiên. Điều này giúp mô hình này có thể xử lý tốt các dữ liệu không tuyến tính và các mối quan hệ phức tạp giữa các biến đầu vào. Bằng cách tạo ra các cây quyết định độc lập và kết hợp chúng lại, mô hình này mang lại kết quả ổn định và chính xác trong các bài toán phân loại.

Trong bài toán này, Rừng Ngẫu Nhiên giúp phân loại mức độ tiêu thụ rượu của học sinh hoặc sinh viên dựa trên các đặc trưng như số ngày vắng mặt, hoạt động ngoại khóa, tần suất ra ngoài và thời gian rảnh rỗi. Mô hình này có thể xử lý các mối quan hệ phức tạp giữa các yếu tố này và đưa ra dự đoán về mức độ tiêu thụ rượu một cách chính xác.

* **Mô hình Máy Véc-tơ Hỗ trợ (SVM)**

Máy Véc-tơ Hỗ trợ (SVM) là một thuật toán học máy mạnh mẽ và linh hoạt, đặc biệt nổi bật trong các bài toán phân loại với không gian dữ liệu có chiều cao. SVM tìm kiếm một siêu phẳng tối ưu (hyperplane) trong không gian đặc trưng sao cho khoảng cách giữa các điểm dữ liệu của các lớp khác nhau là lớn nhất. Điều này giúp SVM có thể phân loại các dữ liệu với độ chính xác cao, đặc biệt là trong các bài toán với dữ liệu phân tách rõ ràng giữa các lớp.

Một trong những đặc điểm nổi bật của SVM là khả năng xử lý tốt các dữ liệu không tuyến tính nhờ vào việc sử dụng các nhân (kernel), giúp chuyển dữ liệu vào không gian có chiều cao hơn và từ đó dễ dàng phân tách hơn. Đặc biệt, SVM thường được áp dụng khi dữ liệu có kích thước lớn và yêu cầu một mô hình có khả năng phân loại với độ chính xác cao. Trong bài toán này, SVM có thể giúp phân loại mức độ tiêu thụ rượu dựa trên các yếu tố xã hội và tâm lý của học sinh hoặc sinh viên, và có thể hoạt động hiệu quả khi các lớp dữ liệu có sự phân tách rõ ràng.

Cả hai mô hình, Rừng Ngẫu Nhiên và SVM, đều mang lại những ưu điểm riêng biệt cho bài toán dự đoán mức độ tiêu thụ rượu. Rừng Ngẫu Nhiên với sự kết hợp của nhiều cây quyết định giúp xử lý các dữ liệu có tính phức tạp cao, còn SVM lại đặc biệt mạnh mẽ khi đối mặt với các bài toán phân loại với dữ liệu phân tách rõ ràng. Việc áp dụng cả hai mô hình này giúp chúng tôi đánh giá độ chính xác của dự đoán và chọn ra mô hình phù hợp nhất cho bài toán nghiên cứu.

## 2.Xây dựng mô hình Máy Véc-tơ Hỗ trợ (SVM)

**Bước 1: Cài đặt các thư viện cần thiết**

Trước khi bắt đầu, bạn cần cài đặt các thư viện cần thiết. Dưới đây là các thư viện chính sử dụng trong mã:

* pip install pandas scikit-learn joblib
* **pandas**: Dùng để xử lý dữ liệu (đọc, lọc, chuyển đổi).
* **scikit-learn**: Thư viện máy học bao gồm SVM, chuẩn hóa dữ liệu, và đánh giá mô hình.
* **joblib**: Dùng để lưu mô hình đã huấn luyện vào tệp và tải lại mô hình.

**Bước 2: Đọc và Tiền xử lý Dữ liệu**

1. Đọc dữ liệu từ tệp CSV: Đọc dữ liệu từ một tệp CSV chứa các thông tin cần thiết cho mô hình. Trong trường hợp này, tệp được lưu trên máy tính là FINAL.csv.



1. Chọn các biến độc lập và biến mục tiêu: Bạn cần chỉ định các cột trong bộ dữ liệu là biến độc lập (features) và biến mục tiêu (target).

A close-up of a computer screen

Description automatically generated

1. Mã hóa biến phân loại: Nếu có các biến phân loại trong bộ dữ liệu, bạn cần mã hóa chúng thành các dạng số. Dưới đây là cách mã hóa biến phân loại thành dạng nhị phân và điền giá trị thiếu (nếu có).

A close-up of words

Description automatically generated

**Bước 3: Chia Dữ liệu thành Tập Huấn Luyện và Tập Kiểm Tra**

Dữ liệu sẽ được chia thành hai phần:

* **Tập huấn luyện (training set)**: Dùng để huấn luyện mô hình.
* **Tập kiểm tra (test set)**: Dùng để kiểm tra hiệu suất của mô hình.

A screenshot of a computer code

Description automatically generated

**Bước 4: Chuẩn hóa Dữ liệu**

SVM yêu cầu dữ liệu phải được chuẩn hóa hoặc chuẩn hóa trước khi đưa vào mô hình. Điều này giúp tăng hiệu suất của mô hình, đặc biệt là khi các đặc trưng có độ lớn khác nhau.

A screenshot of a computer

Description automatically generated

**Bước 5: Xây dựng Mô hình SVM**

Sử dụng **SVM** với kernel RBF (Radial Basis Function), vì kernel này thường được sử dụng khi bài toán là phi tuyến. Bạn cũng có thể tùy chỉnh các tham số khác như C (hệ số điều chỉnh) và gamma (tham số kernel).

A screenshot of a computer code

Description automatically generated

kernel='rbf': Chọn kernel RBF, giúp xử lý các bài toán phi tuyến.

C=1.0: Hệ số điều chỉnh, kiểm soát sự trade-off giữa độ chính xác của mô hình và mức độ phức tạp.

gamma='scale': Tự động tính giá trị gamma (tùy thuộc vào số lượng đặc trưng trong dữ liệu).

random\_state=42: Đảm bảo kết quả tái tạo được.

**Bước 6: Đánh giá Mô hình**

Sau khi mô hình đã được huấn luyện, bạn sẽ sử dụng tập kiểm tra để đánh giá hiệu suất của mô hình.

1. Dự đoán giá trị mục tiêu cho tập kiểm tra: Dự đoán kết quả cho các mẫu trong tập kiểm tra.

A close-up of a word

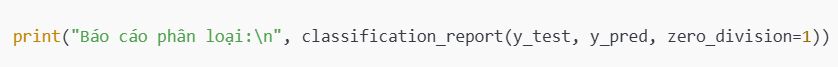
Description automatically generated

1. Đánh giá độ chính xác: Đo độ chính xác của mô hình bằng cách so sánh giá trị dự đoán với giá trị thực tế trong tập kiểm tra.

A close-up of a computer code

Description automatically generated

1. Báo cáo phân loại: Cung cấp các chỉ số đánh giá chi tiết như precision, recall, F1-score cho mỗi lớp trong bài toán phân loại.



**Bước 7: Lưu mô hình**

Sau khi mô hình đã được huấn luyện và đánh giá, bạn có thể lưu mô hình vào tệp để sử dụng lại trong tương lai mà không cần phải huấn luyện lại.

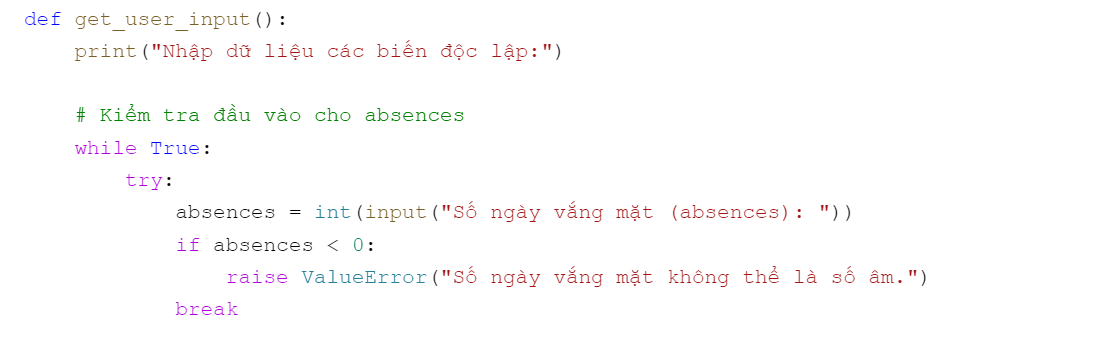
A computer code with text

Description automatically generated with medium confidence

**Bước 8: Nhập Dữ liệu Người Dùng và Dự đoán**

Để dự đoán cho người dùng, bạn có thể yêu cầu người dùng nhập dữ liệu và sử dụng mô hình đã huấn luyện để dự đoán.

1. **Nhập dữ liệu từ người dùng**: Bạn sẽ yêu cầu người dùng nhập các giá trị cho các biến độc lập và kiểm tra các giá trị nhập vào.



A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

1. **Dự đoán từ dữ liệu nhập vào**: Sau khi người dùng nhập dữ liệu, bạn sẽ chuẩn hóa các giá trị và sử dụng mô hình đã huấn luyện để dự đoán kết quả.

A screenshot of a computer code

Description automatically generated

**Kết quả của mô hình trên:**

**A screenshot of a computer

Description automatically generated**

Hình : Kết quả mô hình dự đoán SVM

## 3.Xây dựng mô hình rừng ngẫu nhiên (Random Forest)

Để tiến hành xây dựng mô hình ngẫu nhiên thì cũng làm tương tự mô hình SVM.

* **Huấn luyện mô hình Random Forest**

**A screenshot of a computer code

Description automatically generated**

**Kết quả:**

**A screenshot of a computer

Description automatically generated**

Hình : kết quả mô hình dự đoán rừng ngẫu nhiên

## 4.So sánh mô hình SVM và Rừng Ngẫu nhiên

Từ hai kết quả thu được ta tiến hành so sánh và chọn ra mô hình phù hợp cho tập dữ liệu:

* **Kết quả tổng quan**

**A screenshot of a graph

Description automatically generated**

**Mô hình SVM**

* **Độ chính xác thấp (48.39%)**: Mô hình SVM không phù hợp lắm với bài toán này, đặc biệt là khi dữ liệu không cân bằng.
* **Lớp 1 (Tiêu thụ rượu thấp)**:
  + Precision cao (85%), nhưng recall thấp (44%) => Dự đoán đúng nhiều mẫu thuộc lớp 1, nhưng bỏ sót nhiều trường hợp.
  + F1-score ở mức 0.58, hiệu suất vừa phải.
* **Lớp 2 (Tiêu thụ rượu cao)**:
  + Recall cao (67%), tức là phát hiện tốt các mẫu thực sự thuộc lớp 2.
  + Precision thấp (22%), dự đoán sai nhiều trường hợp không phải lớp 2 nhưng vẫn xếp vào lớp này.
  + F1-score ở mức 0.33, kém hiệu quả.

**Kết luận:** Mô hình SVM ưu tiên phát hiện lớp 2 (recall cao), nhưng tổng thể hoạt động chưa tốt, nhất là đối với dữ liệu bất cân bằng.

**Mô hình Random Forest**

* **Độ chính xác cải thiện đáng kể (64.52%)**: Random Forest thích hợp hơn cho bài toán với dữ liệu phức tạp và nhiều yếu tố ảnh hưởng.
* **Lớp 1 (Tiêu thụ rượu thấp)**:
  + Precision giữ vững ở mức 82%, không giảm nhiều so với SVM.
  + Recall tăng mạnh lên 72%, cho thấy mô hình phát hiện tốt hơn các trường hợp thực sự thuộc lớp 1.
  + F1-score tăng đáng kể lên 0.77 => hiệu suất tổng thể tốt hơn.
* **Lớp 2 (Tiêu thụ rượu cao)**:
  + Precision vẫn ở mức thấp (22%), giống như SVM, vì dữ liệu của lớp 2 ít.
  + Recall giảm xuống 33%, tức là phát hiện ít hơn so với SVM.
  + F1-score giảm nhẹ còn 0.27, nhưng không ảnh hưởng nhiều do lớp 2 có tỷ lệ nhỏ.

**Kết luận:** Random Forest hoạt động tốt hơn trên lớp 1 (đa số), nhưng cần cải thiện khả năng phát hiện lớp 2.

* **So sánh hiệu suất**
* Độ chính xác tổng thể: Random Forest vượt trội hơn SVM, đặc biệt trong việc phân loại chính xác các trường hợp thuộc lớp đa số (lớp 1).
* Khả năng phát hiện lớp 2: SVM có recall cao hơn cho lớp 2, nhưng precision rất thấp, khiến các dự đoán về lớp 2 thiếu tin cậy.
* Sự cân bằng giữa precision và recall: Random Forest cân bằng hơn ở lớp 1, làm tăng F1-score tổng thể.

**Kết luận :**

Random Forest phù hợp hơn với bài toán, đặc biệt khi dữ liệu có tính bất cân bằng.

SVM nhạy cảm hơn với các mẫu lớp thiểu số (lớp 2), nhưng không thực sự hiệu quả trong tổng thể.

=>Mô hình rừng ngẫu nhiên đang phù hợp với tập dữ liệu đang mất cân bằng của nhóm

# CHƯƠNG 5: KẾT LUẬN

## 1.Kết quả đạt được

Dựa trên kết quả phân tích dữ liệu, nhóm đã xây dựng mô hình Random Forest để đánh giá các yếu tố ảnh hưởng đến mức độ tiêu thụ rượu của học sinh. Mô hình Random Forest đạt độ chính xác 64.52%, cao hơn đáng kể so với mô hình SVM chỉ đạt 48.39%, cho thấy đây là mô hình phù hợp để phân loại và dự đoán trong nghiên cứu này. Kết quả chi tiết cho thấy mô hình có khả năng phân loại tốt các trường hợp tiêu thụ rượu thấp (lớp 1), với độ chính xác (precision) 82% và khả năng phát hiện (recall) đạt 72%, mang lại hiệu suất tổng thể cao hơn với F1-score đạt 0.77. Tuy nhiên, đối với các trường hợp tiêu thụ rượu cao (lớp 2), mô hình còn gặp khó khăn trong việc phát hiện chính xác, với recall chỉ đạt 33%.

Nhóm nhận thấy các yếu tố như số ngày vắng mặt, hoạt động ngoại khóa, tần suất ra ngoài, và thời gian rảnh rỗi có tác động nhất định đến mức độ tiêu thụ rượu của học sinh. Mô hình Random Forest không chỉ giúp nhóm đánh giá được tầm quan trọng của các yếu tố này mà còn hỗ trợ dự đoán chính xác hơn trong các trường hợp thực tế. Kết luận từ phân tích này sẽ là cơ sở để nhóm đề xuất các giải pháp phù hợp nhằm giảm thiểu tình trạng tiêu thụ rượu trong nhóm đối tượng nghiên cứu.

Trong quá trình nghiên cứu, nhóm cũng đã triển khai và phân tích các thuật toán hồi quy Logistic và Cây Quyết Định để tìm hiểu sâu hơn về mối quan hệ giữa các yếu tố cụ thể và mức độ tiêu thụ rượu (Walc). Các yếu tố được phân tích bao gồm: tần suất ra ngoài (goout), số ngày vắng mặt (absences), tham gia hoạt động ngoại khóa (activities), và thời gian rảnh rỗi (freetime).

* Hồi quy Logistic: Thuật toán này cho phép nhóm kiểm tra xác suất và mức độ ảnh hưởng của từng yếu tố đến xác suất tiêu thụ rượu. Kết quả cho thấy:
  + Goout: Tần suất ra ngoài có ảnh hưởng mạnh mẽ nhất, với xác suất tiêu thụ rượu tăng đáng kể khi học sinh ra ngoài thường xuyên.
  + Absences: Số ngày vắng mặt cũng có tác động đáng kể, khi học sinh vắng mặt nhiều hơn, mức độ tiêu thụ rượu cũng tăng.
  + Activities: Tham gia hoạt động ngoại khóa có mối liên hệ ngược chiều, tức là những học sinh tham gia hoạt động ngoại khóa thường xuyên có xu hướng tiêu thụ rượu thấp hơn.
  + Freetime: Thời gian rảnh rỗi có tác động ít hơn so với các yếu tố khác, nhưng vẫn cho thấy mối quan hệ tích cực với mức độ tiêu thụ rượu.
* Cây Quyết Định: Khi sử dụng Cây Quyết Định, nhóm có thể xác định rõ các ngưỡng giá trị cụ thể của từng yếu tố. Ví dụ:
  + Với goout, cây quyết định cho thấy nếu tần suất ra ngoài lớn hơn 4 (trên thang điểm 5), học sinh có khả năng cao rơi vào nhóm tiêu thụ rượu cao.
  + Với absences, nếu số ngày vắng mặt lớn hơn 7, nguy cơ tiêu thụ rượu cao cũng tăng lên rõ rệt.
  + Với activities, cây cho thấy nhóm không tham gia hoạt động ngoại khóa thường nằm trong nhóm tiêu thụ rượu cao.
* Qua các phân tích trên, nhóm nhận thấy rằng mỗi thuật toán đều có những ưu điểm riêng trong việc phân tích dữ liệu. Tuy nhiên, mô hình Random Forest vẫn được đánh giá cao hơn nhờ khả năng xử lý dữ liệu phức tạp và hiệu quả trong dự đoán. Các phát hiện từ hồi quy Logistic và Cây Quyết Định đã bổ sung giá trị, giúp nhóm hiểu rõ hơn về từng mối quan hệ cụ thể giữa các yếu tố đầu vào và mức độ tiêu thụ rượu, từ đó đưa ra những giải pháp phù hợp hơn cho nghiên cứu.

## 2.Hạn chế

Trong quá trình thực hiện nghiên cứu, nhóm nhận thấy một số hạn chế cần khắc phục để cải thiện kết quả. Trước hết, dữ liệu bị mất cân bằng giữa các lớp mức độ tiêu thụ rượu (low và high consumption), với số lượng mẫu của lớp "tiêu thụ thấp" chiếm ưu thế. Điều này ảnh hưởng đến hiệu quả của các thuật toán, đặc biệt là các mô hình như cây quyết định và hồi quy logistic, vốn nhạy cảm với sự mất cân bằng dữ liệu. Mặc dù nhóm đã áp dụng kỹ thuật SMOTE để cân bằng dữ liệu, nhưng phương pháp này vẫn có thể tạo ra dữ liệu tổng hợp không phản ánh chính xác đặc điểm thực tế, dẫn đến độ chính xác giảm trong thực tế.

Bên cạnh đó, một số yếu tố quan trọng khác ảnh hưởng đến mức độ tiêu thụ rượu, như thói quen sinh hoạt gia đình, môi trường xã hội, hoặc áp lực học tập, chưa được thu thập hoặc đưa vào phân tích. Việc chỉ tập trung vào các yếu tố như *goout*, *freetime*, *absences*, và *activities* có thể chưa đủ để phản ánh toàn diện các nguyên nhân ảnh hưởng. Ngoài ra, nhóm cũng nhận thấy việc lựa chọn mô hình chủ yếu dựa trên độ chính xác mà chưa đánh giá kỹ lưỡng các chỉ số khác như F1-score, precision, hay recall, nhất là trong trường hợp dữ liệu mất cân bằng.

## 3.Hướng phát triển

Trong tương lai, nhóm sẽ tập trung vào việc cải thiện độ chính xác của mô hình dự đoán mức độ tiêu thụ rượu bằng cách áp dụng các kỹ thuật và mô hình tiên tiến hơn. Đầu tiên, nhóm dự định thu thập thêm dữ liệu về các yếu tố ảnh hưởng chưa được xem xét, như thói quen gia đình, môi trường xã hội, và áp lực học tập, để có cái nhìn tổng thể và chính xác hơn về các yếu tố ảnh hưởng đến việc tiêu thụ rượu. Điều này có thể giúp mô hình hiểu rõ hơn về mối quan hệ giữa các yếu tố cá nhân và hành vi tiêu thụ rượu.

Thứ hai, nhóm sẽ thử nghiệm với các mô hình học sâu (Deep Learning), đặc biệt là các mạng nơ-ron nhân tạo (ANN), có khả năng nhận diện và học từ các mẫu phức tạp trong dữ liệu. Những mô hình này có thể xử lý tốt hơn với dữ liệu phi tuyến tính và có thể đưa ra dự đoán chính xác hơn trong các trường hợp có sự kết hợp giữa nhiều yếu tố tác động. Bên cạnh đó, việc cải thiện các chiến lược xử lý mất cân bằng dữ liệu, chẳng hạn như kết hợp SMOTE với các thuật toán học máy khác hoặc sử dụng kỹ thuật sampling thích ứng, sẽ giúp mô hình tránh được bias và cải thiện độ chính xác.

Ngoài ra, nhóm sẽ tiến hành đánh giá mô hình theo nhiều chỉ số khác nhau, như F1-score, precision, recall và AUC-ROC, để đảm bảo rằng mô hình không chỉ chính xác mà còn có khả năng dự đoán tốt các lớp ít dữ liệu, giúp cải thiện sự công bằng trong dự đoán. Việc xây dựng một hệ thống dự đoán tương tác, cho phép người dùng nhập các dữ liệu cá nhân và nhận kết quả dự đoán trực tiếp, sẽ là một bước đi quan trọng trong việc áp dụng mô hình vào thực tế và nâng cao tính ứng dụng của nghiên cứu.

Cuối cùng, nhóm chưa thử nghiệm với các mô hình tiên tiến hơn, chẳng hạn như Gradient Boosting, XGBoost hoặc mô hình deep learning, vốn có khả năng xử lý tốt hơn với các tập dữ liệu phức tạp và mất cân bằng. Các mô hình này, đặc biệt là XGBoost và Gradient Boosting, đã chứng minh được hiệu quả vượt trội trong việc xử lý các mối quan hệ phi tuyến tính và giảm thiểu hiện tượng overfitting khi làm việc với dữ liệu phức tạp. Đây là những hạn chế mà nhóm sẽ tiếp tục cải thiện trong các nghiên cứu tương lai nhằm nâng cao chất lượng và độ chính xác của kết quả. Bằng cách áp dụng các mô hình này, nhóm kỳ vọng sẽ có thể đưa ra dự đoán chính xác hơn và tạo ra một hệ thống phân tích đáng tin cậy hơn trong việc dự đoán các yếu tố ảnh hưởng đến mức độ tiêu thụ rượu.

# DANH MỤC TÀI LIỆU THAM KHẢO

1. Phan, T. H. (2024). Phân tích dữ liệu - Buổi 02, Tiền xử lý dữ liệu [PDF].
2. Phan, T. H. (2024). Phân tích dữ liệu - Buổi 03, Phân tích hồi quy [PDF].
3. Phan, T. H. (2024). Phân tích dữ liệu - Buổi 06,07, Phân tích luật kết hợp [PDF].
4. Danh, Nguyễn Thanh, Đậu Văn Ngọ, and Tạ Quốc Dũng. "Ứng dụng phương pháp hồi quy logistic xác định tổ hợp tối ưu các yếu tố ảnh hưởng và xây dựng bản đồ tai biến trượt lở đất huyện Khánh Vĩnh, tỉnh Khánh Hòa." *Science and Technology Development Journal* 20.K4-2017 (2017).
5. Thủy, Nguyễn Minh, and Nguyễn Thị Mỹ Tuyền. "Ứng dụng phương pháp phân tích thành phần chính, hồi quy logistic và giản đồ yêu thích trong đánh giá cảm quan sản phẩm sữa gạo." *Tạp chí Khoa học Đại học cần Thơ* 37 (2015): 11-20.
6. Đệ, Trần Cao, and Phạm Nguyên Khang. "Phân loại văn bản với máy học vector hỗ trợ và cây quyết định." *Tạp chí Khoa học Đại học cần Thơ* 21a (2012): 52-63.
7. Đức, Đảm Nguyễn, et al. "Phát triển mô hình học máy cây quyết định và cây quyết đinh xen kẽ thành lập bản đồ dự báo không gian sạt lở đất tại huyện Mường Nhé, tỉnh Điện Biên, Việt Nam." *Tạp chí điện tử Khoa học và Công nghệ Giao thông* (2022): 36-56.
8. Mammone, Alessia, Marco Turchi, and Nello Cristianini. "Support vector machines." *Wiley Interdisciplinary Reviews: Computational Statistics* 1.3 (2009): 283-289.

     9.Tập dữ liệu -truy cập <https://www.kaggle.com/code/karenzhang0701/student-alcohol-consumption-eda>

# DANH MỤC HÌNH ẢNH

[Hình 1 : thông tin tổng quan về dữ liệu 10](#_Toc184659004)

[Hình 2 : Tổng quan dữ liệu 11](#_Toc184659005)

[Hình 3 : Đoạn code giúp phân biêt biến phân loại và biến số 11](#_Toc184659006)

[Hình 4: Kết quả cho ra , Biến phân loại và Biến số trong tập dữ liệu 11](#_Toc184659007)

[Hình 5: kết quả của làm sạch dữ liệu 14](#_Toc184659008)

[Hình 6: kết quả kiểm tra giá trị thiếu 15](#_Toc184659009)

[Hình 7: Kết quả của giá trị ngoại lai 17](#_Toc184659010)

[Hình 8: Kết quả sau khi được chuẩn hóa 18](#_Toc184659011)

[Hình 9:Kết quả sau khi lọc các giá trị ngoại lai và chuẩn hóa 19](#_Toc184659012)

[Hình 10: Kết quả của loại bỏ các giá trị 19](#_Toc184659013)

[Hình 11: Cây quyết định 20](#_Toc184659014)

[Hình 12: Rừng ngẫu nhiên 21](#_Toc184659015)

[Hình 13:Hình ảnh tương quan của các biến 22](#_Toc184659016)

[Hình 14:Kết quả của hồi quy Logistic 25](#_Toc184659017)

[Hình 15:Biểu đồ mối quan hệ giữa absences và Walc 27](#_Toc184659018)

[Hình 16:biểu đồ hồi quy giữa absences và Walc 28](#_Toc184659019)

[Hình 17:Kết quả của Cây quyết định 29](#_Toc184659020)

[Hình 18:Cây quyết định giữa absences và Walc 31](#_Toc184659021)

[Hình 19:Sơ đồ đường quyết định cây quyết định 31](#_Toc184659022)

[Hình 20:Kết quả của Rừng ngẫu nhiên 32](#_Toc184659023)

[Hình 21:Sơ đồ đường quyết định rừng ngẫu nhiên 34](#_Toc184659024)

[Hình 22:Kết quả của SVM 35](#_Toc184659025)

[Hình 23:biểu đồ 3D giữa absences và Walc và freetime 36](#_Toc184659026)

[Hình 24:sơ đồ đường quyết định của mô hình SVM 37](#_Toc184659027)

[Hình 25:Kết quả của hồi quy Logistic 40](#_Toc184659028)

[Hình 26:Biểu đồ mối quan hệ giữa freetime và Walc 42](#_Toc184659029)

[Hình 27:biểu đồ hồi quy giữa freetime và Walc 43](#_Toc184659030)

[Hình 28:Kết quả của Cây quyết định 44](#_Toc184659031)

[Hình 29:Cây quyết định giữa freetime và Walc 46](#_Toc184659032)

[Hình 30:Sơ đồ đường quyết định cây quyết định 46](#_Toc184659033)

[Hình 31:Kết quả của Rừng ngẫu nhiên 48](#_Toc184659034)

[Hình 32:Sơ đồ đường quyết định rừng ngẫu nhiên 50](#_Toc184659035)

[Hình 33:Kết quả của SVM 51](#_Toc184659036)

[Hình 34:biểu đồ 3D giữa absences và Walc và freetime 52](#_Toc184659037)

[Hình 35:sơ đồ đường quyết định của mô hình SVM 53](#_Toc184659038)

[Hình 36:Kết quả của hồi quy logistic 57](#_Toc184659039)

[Hình 37:Ma trận nhầm lẫn 59](#_Toc184659040)

[Hình 38:Biểu đồ trực quan hóa Hồi Quy logistic 60](#_Toc184659041)

[Hình 39:Kết quả của mô hình cây 63](#_Toc184659042)

[Hình 40:Trực quan hóa cây 66](#_Toc184659043)

[Hình 41:Kết quả từ mô hình Rừng ngẫu nhiên 69](#_Toc184659044)

[Hình 42:Biểu đồ trực quan hóa của rừng ngẫu nhiên 71](#_Toc184659045)

[Hình 43:Kết quả từ mô hình SVM 73](#_Toc184659046)

[Hình 44:Biểu đồ trực quan hóa của mô hình SVM 75](#_Toc184659047)

[Hình 45:Kết quả của hồi quy logistic 79](#_Toc184659048)

[Hình 46:Kết quả trực quan biểu đồ hồi quy logistic 81](#_Toc184659049)

[Hình 47:Kết quả của mô hình cây quyết định 83](#_Toc184659050)

[Hình 48:Trực quan hóa cây quyết định 86](#_Toc184659051)

[Hình 49:Kết quả của mô hình ngẫu nhiên 89](#_Toc184659052)

[Hình 50:Biểu đồ trực quan hóa của mô hình rừng ngẫu nhiên 91](#_Toc184659053)

[Hình 51: Kết quả từ mô hình SVM 92](#_Toc184659054)

[Hình 52:Biểu đồ trực quan hóa của mô hình SVM 94](#_Toc184659055)

[Hình 53: Kết quả mô hình dự đoán SVM 103](#_Toc184659056)

[Hình 54: kết quả mô hình dự đoán rừng ngẫu nhiên 104](#_Toc184659057)