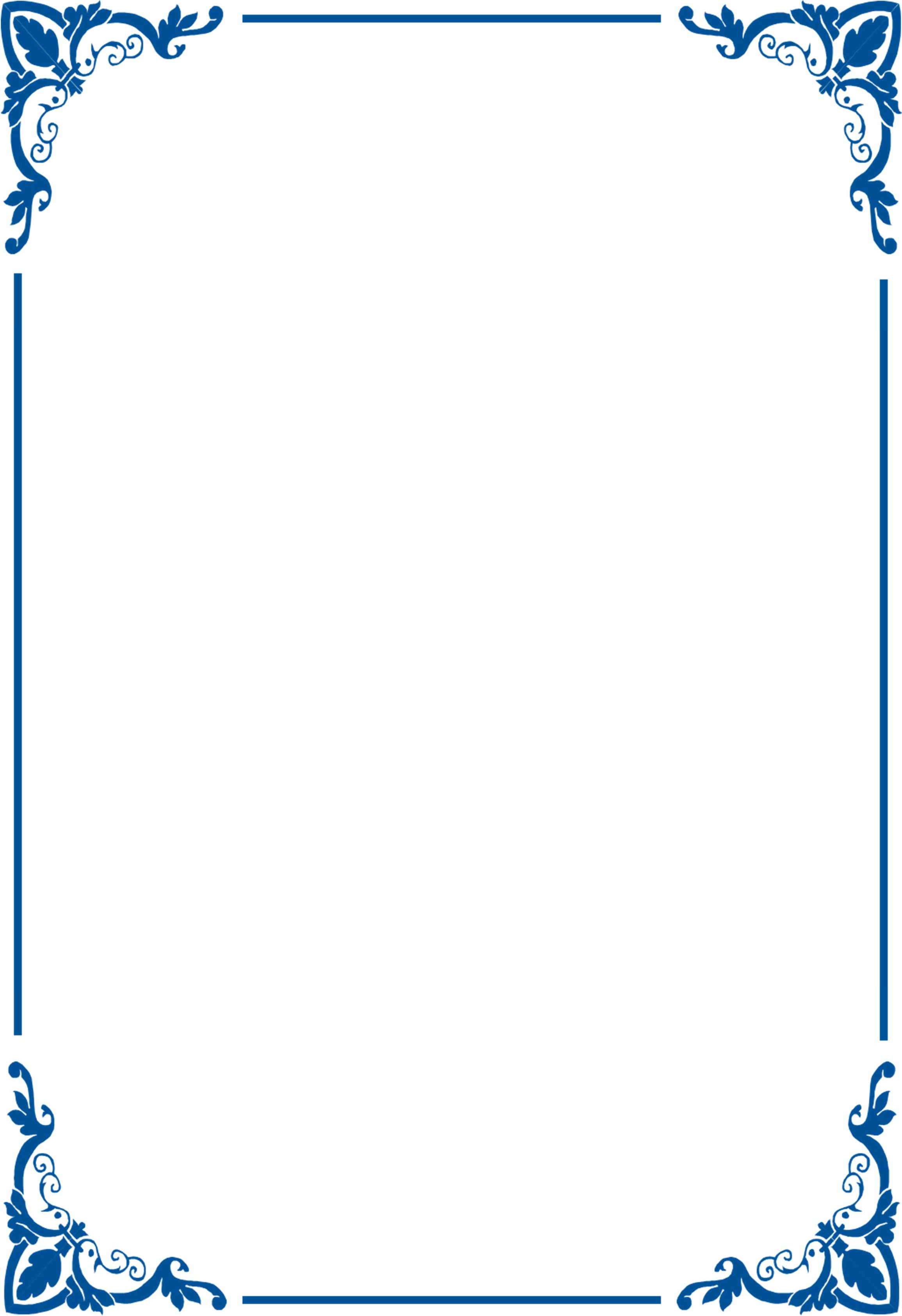
**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**MÔN HỌC:** KHAI THÁC DỮ LIỆU

Giảng viên hướng dẫn: **Ths. Mai Xuân Hùng**

**GV. Phạm Nguyễn Thanh Bình**

Tên đề tài: **DỰ ĐOÁN KHẢ NĂNG HỦY ĐẶT PHÒNG CỦA HAI KHÁCH SẠN TẠI BỒ ĐÀO NHA (2015 – 2017)**

Lớp: **IS252.021**

**Phạm Quốc Hùng - 19521579**

**Lê Trần Anh Quí - 21520094**

**Đặng Lưu Hà - 21520798**

**Trần Minh Quang - 21522519**

**NHẬN XÉT CỦA GIẢNG VIÊN**

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

A logo of a planet

Description automatically generated…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH 6](#_Toc167699864)

[DANH MỤC BẢNG BIỂU 10](#_Toc167699865)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 11](#_Toc167699866)

[1. Lý do chọn đề tài 11](#_Toc167699867)

[2. Giới thiệu nguồn dữ liệu 11](#_Toc167699868)

[3. Lọc dữ liệu 12](#_Toc167699869)

[4. Mô tả thuộc tính 15](#_Toc167699870)

[5. Giới thiệu bài toán 20](#_Toc167699871)

[CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU 22](#_Toc167699872)

[1. Trực quan hóa dữ liệu: 22](#_Toc167699873)

[1.1. Thống kê số lượng nhận phòng và hủy phòng: 22](#_Toc167699874)

[1.2. Thống kê loại hình khách sạn nào thu nhiều khách hàng nhất 23](#_Toc167699875)

[1.3. Thống kê doanh thu mỗi tháng trong tập dữ liệu 23](#_Toc167699876)

[1.4. Thống kê tỷ lệ hủy đặt phòng với từng loại hình đặt cọc 27](#_Toc167699877)

[1.5. Thống kê tỷ lệ phần trăm của các giá trị trong thuộc tính phân khúc thị trường với trạng thái hủy đặt phòng và nhận phòng. 27](#_Toc167699878)

[1.6. Tỷ lệ các trạng thái đăng ký khi hủy bỏ phòng thành công 29](#_Toc167699879)

[1.7. Thống kê tỷ lệ phòng bị hủy 29](#_Toc167699880)

[1.8. Kênh phân phối nào có tỷ lệ hủy cao nhất 30](#_Toc167699881)

[1.9. Thể hiện hệ số tương quan giữa giá trị hủy đặt phòng và các thuộc tính còn lại. 31](#_Toc167699882)

[2. Mô tả cơ bản về tập dữ liệu: 33](#_Toc167699883)

[3. Làm sạch dữ liệu: 36](#_Toc167699884)

[3.1. Xử lý dữ liệu bị nhiễu (outlier): 36](#_Toc167699885)

[3.2. Xử lí dữ liệu bị thiếu: 38](#_Toc167699886)

[3.3. Kiểm tra ý nghĩa của dữ liệu: 41](#_Toc167699887)

[3.4. Kiểm tra thuộc tính của dữ liệu: 43](#_Toc167699888)

[3.5. Xử lý dữ liệu bị trùng lặp: 44](#_Toc167699889)

[4. Thêm thuộc tính cho dữ liệu: 45](#_Toc167699890)

[5. Giảm thuộc tính của dữ liệu: 47](#_Toc167699891)

[6. Rời rạc hóa dữ liệu: 49](#_Toc167699892)

[6.1. Thuộc tính LeadTime: 50](#_Toc167699893)

[6.2. Thuộc tính Adults: 51](#_Toc167699894)

[6.3. Thuộc tính ADR: 52](#_Toc167699895)

[6.4. Thuộc tính StatuMinusArrivalDate: 52](#_Toc167699896)

[6.5. Thuộc tính TotalNights: 53](#_Toc167699897)

[6.6. Thuộc tính TotalPeople: 54](#_Toc167699898)

[7. Dán nhãn số cho các thuộc tính định tính: 55](#_Toc167699899)

[8. Lấy tập mẫu cho dữ liệu (Sampling data): 55](#_Toc167699900)

[CHƯƠNG 3: CÁC THUẬT TOÁN KHAI THÁC DỮ LIỆU 57](#_Toc167699901)

[1. Chuẩn bị dữ liệu để chạy các thuật toán: 57](#_Toc167699902)

[2. Thuật toán Cây quyết định (Decision Tree): 59](#_Toc167699903)

[2.1. Khái niệm: 59](#_Toc167699904)

[2.2. Thực hiện: 60](#_Toc167699905)

[2.2.1. Giải thuật Decision Tree ID3: 60](#_Toc167699906)

[2.2.2. Giải thuật Decision Tree Cart: 62](#_Toc167699907)

[3. Thuật toán Naïve Bayes: 65](#_Toc167699908)

[3.1. Khái niệm: 65](#_Toc167699909)

[3.2. Thực hiện: 66](#_Toc167699910)

[4. Thuật toán K - Nearest Neighbors: 68](#_Toc167699911)

[4.1. Khái niệm: 68](#_Toc167699912)

[4.2. Thực hiện: 68](#_Toc167699913)

[5. Thuật toán Neural Network: 70](#_Toc167699914)

[5.1. Khái niệm: 70](#_Toc167699915)

[5.2. Thực hiện: 70](#_Toc167699916)

[6. Thuật toán Random Forest: 72](#_Toc167699917)

[6.1. Khái niệm: 72](#_Toc167699918)

[6.2. Thực hiện: 72](#_Toc167699919)

[CHƯƠNG 4: ĐÁNH GIÁ CÁC THUẬT TOÁN 75](#_Toc167699920)

[1. Đánh giá ma trận nhầm lẫn: 75](#_Toc167699921)

[2. Đánh giá tỷ lệ TPR trên tỷ lệ FPR trong biểu đồ đường cong ROC: 76](#_Toc167699922)

[3. Đánh giá về thời gian chạy từng thuật toán: 77](#_Toc167699923)

[4. Đánh giá độ chính xác của thuật toán trên tập huấn luyện: 77](#_Toc167699924)

[5. Đánh giá độ chính xác của thuật toán trên tập kiểm thử: 78](#_Toc167699925)

[CHƯƠNG 5: KẾT LUẬN 80](#_Toc167699926)

[1. Ưu - Nhược điểm của từng thuật toán: 80](#_Toc167699927)

[1.1. Decision Tree 80](#_Toc167699928)

[1.2. Naïve Bayes 80](#_Toc167699929)

[1.3. K – Nearest Neighbor 81](#_Toc167699930)

[1.4. Neural Network 81](#_Toc167699931)

[1.5. Random Forest 81](#_Toc167699932)

[2. Hướng phát triển: 82](#_Toc167699933)

[BẢNG PHÂN CHIA CÔNG VIỆC 83](#_Toc167699934)

[BẢNG ĐÁNH GIÁ CÁC THÀNH VIÊN TRONG NHÓM 84](#_Toc167699935)

[TÀI LIỆU THAM KHẢO 84](#_Toc167699936)

DANH MỤC HÌNH ẢNH

[Hình 1. 1 Bộ dữ liệu Hotel Booking Demands 12](#_Toc167644705)

[Hình 1. 2 Thêm cột hotel vô 2 file csv 13](#_Toc167644706)

[Hình 1. 3 Gộp 2 file csv thành 1 file csv mới 13](#_Toc167644707)

[Hình 1. 4 Dữ liệu file hotel.csv 14](#_Toc167644708)

[Hình 1. 5 Dữ liệu file hotel.csv 15](#_Toc167644709)

[Hình 2. 1 Số lượng nhận và hủy phòng 22](#_Toc167647315)

[Hình 2. 2 Biểu đồ thể hiện số lượng nhận và hủy đặt phòng 22](#_Toc167647316)

[Hình 2. 3 Biểu đồ khách hàng từng khách sạn 23](#_Toc167647317)

[Hình 2. 4 Tạo biến lưu trữ theo từng loại hình khách sạn và nhận phòng thành công 23](#_Toc167647318)

[Hình 2. 5 Thống kê giá trị doanh thu trung bình theo từng tháng của khách sạn h1 24](#_Toc167647319)

[Hình 2. 6 Thống kê giá trị doanh thu trung bình theo từng tháng của khách sạn h2 25](#_Toc167647320)

[Hình 2. 7 Thống kê doanh thu trung bình của từng tháng của hai loại hình khách sạn 26](#_Toc167647321)

[Hình 2. 8 Biểu đồ doanh thu theo từng tháng của từng khách sạn 26](#_Toc167647322)

[Hình 2. 9 Biểu đồ thống kê tỷ lệ hủy đặt phòng với từng loại hình đặt cọc 27](#_Toc167647323)

[Hình 2. 10 Code để vẽ biểu đồ thống kê tỷ lệ phần trăm của các giá trị trong thuộc tính phân khúc thị trường với trạng thái hủy đặt phòng và nhận phòng 28](#_Toc167647324)

[Hình 2. 11 Biểu đồ thống kê tỷ lệ phần trăm của các giá trị trong thuộc tính phân khúc thị trường với trạng thái hủy đặt phòng và nhận phòng 28](#_Toc167647325)

[Hình 2. 12 Biểu đồ tỷ lệ các trạng thái đăng ký khi hủy bỏ phòng thành công 29](#_Toc167647326)

[Hình 2. 13 Biểu đồ thống kê tỷ lệ phòng bị hủy 30](#_Toc167647327)

[Hình 2. 14 Biểu đồ thống kê kênh phân phối nào có tỷ lệ hủy cao nhất 31](#_Toc167647328)

[Hình 2. 16 Biểu đồ hệ số tương quan giữa giá trị hủy đặt phòng và các thuộc tính còn lại 32](#_Toc167647329)

[Hình 2. 17 Thông tin tập dữ liệu ban đầu 33](#_Toc167647330)

[Hình 2. 18 Thông tin tập dữ liệu sau khi thay đổi kiểu dữ liệu 2 cột 34](#_Toc167647331)

[Hình 2. 19 Thống kê thông tin thuộc tính định lượng 35](#_Toc167647332)

[Hình 2. 20 Thống kê thông tin thuộc tính định tính 36](#_Toc167647333)

[Hình 2. 21 Biểu đồ Boxplot cho cột ADR 37](#_Toc167647334)

[Hình 2. 22 Loại bỏ cá ngoại lệ của cột ADR 37](#_Toc167647335)

[Hình 2. 23 Liệt kê các giá trị null 38](#_Toc167647336)

[Hình 2. 24 Tỉ lệ dữ liệu null 38](#_Toc167647337)

[Hình 2. 25 Liệt kê số giá trị và tỉ lệ null của của các cột 39](#_Toc167647338)

[Hình 2. 26 Xóa các dòng dữ liệu Children bị null 39](#_Toc167647339)

[Hình 2. 27 Xem thông tin của cột Country 40](#_Toc167647340)

[Hình 2. 28 Thay thế giá trị rỗng của Country bằng PRT 40](#_Toc167647341)

[Hình 2. 29 Biến đổi các giá trị null của Agent và Company thành giá trị mặc định 40](#_Toc167647342)

[Hình 2. 30 Liệt kê lại các giá trị null 41](#_Toc167647343)

[Hình 2. 31 Kiểm tra adults, children và babies có đồng thời bằng không 42](#_Toc167647344)

[Hình 2. 32 Xuất ra các dòng có adults, children và babies đồng thời bằng không 42](#_Toc167647345)

[Hình 2. 33 Xóa các dòng có 3 thuộc tính đó bằng 0 42](#_Toc167647346)

[Hình 2. 34 Liệt kê kiểu dữ liệu của các cột 43](#_Toc167647347)

[Hình 2. 35 Thay đổi dữ liệu từ float sang int 44](#_Toc167647348)

[Hình 2. 36 Thay đổi dữ liệu từ object sang date 44](#_Toc167647349)

[Hình 2. 37 Đếm số dòng trùng nhau 44](#_Toc167647350)

[Hình 2. 38 Xóa các dòng trùng lặp, chỉ giữ lại dòng đầu 44](#_Toc167647351)

[Hình 2. 39 Dữ liệu sau khi xóa các dòng trùng nhau 45](#_Toc167647352)

[Hình 2. 40 Tạo thuộc tính ArrivalDate 45](#_Toc167647353)

[Hình 2. 41 Tạo thuộc tính StatuMinusArrivalDate 46](#_Toc167647354)

[Hình 2. 42 46](#_Toc167647355)

[Hình 2. 43 Tạo thuộc tính RoomMatch 46](#_Toc167647356)

[Hình 2. 44 Tạo thuộc tính TotalPeople 46](#_Toc167647357)

[Hình 2. 45 Tạo thuộc tính HaveChild 47](#_Toc167647358)

[Hình 2. 46 Tính hệ số tương quan 47](#_Toc167647359)

[Hình 2. 47 Hiển thị các hệ số tương quan theo thứ tự giảm dần 48](#_Toc167647360)

[Hình 2. 48 Xóa các thuộc tính không cần thiết 49](#_Toc167647361)

[Hình 2. 49 Thống kê thuộc tính định lượng 50](#_Toc167647362)

[Hình 2. 50 Chia giỏ thuộc tính LeadTime 51](#_Toc167647363)

[Hình 2. 51 Chia dữ liệu theo giỏ thuộc tính LeadTime 51](#_Toc167647364)

[Hình 2. 52 Chia giỏ thuộc tính Adults 51](#_Toc167647365)

[Hình 2. 53 Chia dữ liệu theo giỏ thuộc tính Adults 52](#_Toc167647366)

[Hình 2. 54 Chia giỏ thuộc tính ADR 52](#_Toc167647367)

[Hình 2. 55 Chia dữ liệu theo giỏ thuộc tính ADR 52](#_Toc167647368)

[Hình 2. 56 Chia giỏ thuộc tính StatuMinusArrivalDate 53](#_Toc167647369)

[Hình 2. 57 Chia dữ liệu theo giỏ thuộc tính StatuMinusArrivalDate 53](#_Toc167647370)

[Hình 2. 58 Chia giỏ thuộc tính TotalNughts 53](#_Toc167647371)

[Hình 2. 59 Chia dữ liệu theo giỏ thuộc tính TotalNights 54](#_Toc167647372)

[Hình 2. 60 Chia giỏ thuộc tính TotalPeople 54](#_Toc167647373)

[Hình 2. 61 Chia dữ liệu theo giỏ thuộc tính TotalPeople 54](#_Toc167647374)

[Hình 2. 62 Dán nhãn số cho các thuộc tính định tính 55](#_Toc167647375)

[Hình 2. 63 Kiểm tra các giá trị trong IsCanceled 56](#_Toc167647376)

[Hình 2. 64 Thực hiện Sampling dữ liệu 56](#_Toc167647377)

[Hình 2. 65 Kết quả sau sampling dữ liệu 56](#_Toc167647378)

[Hình 3. 1 Loại bỏ các thuộc tính có tương quan cao ra khỏi thuộc tính 57](#_Toc167647379)

[Hình 3. 2 Chuẩn bị các mảng 58](#_Toc167647380)

[Hình 3. 3 Chuẩn bị các mảng dữ liệu lưu lại thời gian chạy, độ chính xác thuật toán 59](#_Toc167647381)

[Hình 3. 4 Hàm lưu ảnh cây quyết định ID3 60](#_Toc167647382)

[Hình 3. 5 Hàm để vẽ biểu đồ đường cong ROC cho Decision Tree ID3 60](#_Toc167647383)

[Hình 3. 6 Cài đặt thuật toán Decision Tree ID3 61](#_Toc167647384)

[Hình 3. 7 Kết quả chạy ID3 62](#_Toc167647385)

[Hình 3. 8 Mô hình cây quyết định ID3 62](#_Toc167647386)

[Hình 3. 9 Hàm lưu ảnh cây quyết định CART 63](#_Toc167647387)

[Hình 3. 10 Hàm để vẽ biểu đồ đường cong ROC cho Decision Tree Cart 63](#_Toc167647388)

[Hình 3. 11 Cài đặt thuật toán Decision Tree CART 64](#_Toc167647389)

[Hình 3. 12 Kết quả chạy CART 65](#_Toc167647390)

[Hình 3. 13 Mô hình cây quyết định CART 65](#_Toc167647391)

[Hình 3. 14 Hàm để vẽ biểu đồ đường cong ROC cho Naïve Bayes 66](#_Toc167647392)

[Hình 3. 15 Cài đặt thuật toán Naïve Bayes 67](#_Toc167647393)

[Hình 3. 16 Kết quả chạy Naïve Bayes 68](#_Toc167647394)

[Hình 3. 17 Hàm để vẽ biểu đồ đường cong ROC cho K - Nearest Neighbors 68](#_Toc167647395)

[Hình 3. 18 Cài đặt thuật toán K - Nearest Neighbors 69](#_Toc167647396)

[Hình 3. 19 Kết quả chạy K - Nearest Neighbors 70](#_Toc167647397)

[Hình 3. 20 Hàm để vẽ biểu đồ đường cong ROC cho Neural Network 70](#_Toc167647398)

[Hình 3. 21 Cài đặt thuật toán Neural Network 71](#_Toc167647399)

[Hình 3. 22 Kết quả chạy Neural Network 72](#_Toc167647400)

[Hình 3. 23 Hàm lưu ảnh cây ngẫu nhiên Random Forest 72](#_Toc167647401)

[Hình 3. 24 Hàm để vẽ biểu đồ đường cong ROC cho Random Forest 73](#_Toc167647402)

[Hình 3. 25 Cài đặt thuật toán Random Forest 73](#_Toc167647403)

[Hình 3. 26 Kết quả chạy Random Forest 74](#_Toc167647404)

[Hình 3. 27 Mô hình cây ngẫu nhiên 74](#_Toc167647405)

[Hình 4. 1 Ma trận nhầm lẫn của 6 thuật toán 75](#_Toc167647406)

[Hình 4. 2 Biểu đồ đường cong ROC 76](#_Toc167647407)

[Hình 4. 3 Thống kê thời gian chạy của từng thuật toán 77](#_Toc167647408)

[Hình 4. 4 Biểu đồ thời gian chạy của từng thuật toán 77](#_Toc167647409)

[Hình 4. 5 Thống kê độ chính xác trên tập train của từng thuật toán 78](#_Toc167647410)

[Hình 4. 6 Biểu đồ độ chính xác trên tập train của từng thuật toán 78](#_Toc167647411)

[Hình 4. 7 Thống kê độ chính xác trên tập test của từng thuật toán 79](#_Toc167647412)

[Hình 4. 8 Biểu đồ độ chính xác trên tập test của từng thuật toán 79](#_Toc167647413)

DANH MỤC BẢNG BIỂU

[Bảng 1. 1 Mô tả thuộc tính dữ liệu 20](#_Toc167647414)

1. TỔNG QUAN ĐỀ TÀI
   1. Lý do chọn đề tài

Ngành du lịch là một ngành quan trọng để phát triển kinh tế của một quốc gia, và việc đặt chỗ ở là một hoạt động không thể thiếu khi đi du lịch. Với một quốc gia có nhiều điểm đến thú vị như Bồ Đào Nha, dữ liệu về việc đặt phòng của các khách sạn sẽ là một tập dữ liệu lớn và đa dạng để phục vụ cho việc phân tích. Tìm hiểu về dữ liệu của các khách sạn sẽ đem lại những thông tin hữu ích để định hướng, phát triển khách sạn trong tương lai liên quan đến việc hủy phòng, đối tượng khách hàng, mức độ hài lòng của khách hàng, thời điểm thu hút khách, …

Chính vì lẽ đó, nhóm chúng em đã chọn bộ dữ liệu về nhu cầu đặt phòng của 2 khách sạn điển hình cho hai khu vực du lịch nổi tiếng. Một là khách sạn dạng resort ở Algarve, khách sạn thứ hai nằm ở thành phố Lisbon - thủ đô Bồ Đào Nha với đa dạng các thuộc tính về đặt phòng như thông số phòng, thời gian lưu trú, thời gian từ khi đặt phòng đến khi lưu trú, liệu đặt phòng có bị hủy hay không và cách thức đặt phòng được thực hiện, ...

* 1. Giới thiệu nguồn dữ liệu

Bộ dữ liệu nhóm sử dụng có tên là Hotel booking demand datasets được đăng tải trên trang web ScienceDirect bởi ba tác giả là Nuno Antonio, Ana de Almeida và Luis Nunes.

A screenshot of a computer

Description automatically generated

Hình 1. Bộ dữ liệu Hotel Booking Demands

Bộ dữ liệu này chứa thông tin về nhu cầu đặt phòng khách sạn tại Bồ Đào Nha. Khách sạn H1 là một khu nghỉ dưỡng tại Algarve và H2 là một khách sạn trong thành phố Lisbon. Cả hai tập dữ liệu của hai khách sạn đều có cấu trúc giống nhau, bao gồm 31 biến mô tả với 40.060 lượt đặt phòng của khách sạn H1 và 79.330 lượt đặt phòng của khách sạn H2. Mỗi lượt đặt phòng được biểu diễn trên một dòng dữ liệu.

Các lượt đặt phòng được ghi nhận từ 01/07/2015 đến 31/08/2017, bao gồm cả đặt phòng thành công và đặt phòng đã hủy.

Nguồn dữ liệu:

<https://www.sciencedirect.com/science/article/pii/S2352340918315191#s0005>

* 1. Lọc dữ liệu

Do bộ dữ liệu bao gồm 2 file csv và có quá nhiều thuộc tính mà nhóm em không sử dụng hết nên nhóm em sử dụng Python để thực hiện lọc và tiền xử lý dữ liệu.

**Bước 1**: Khai báo thư viện cần thiết, đọc file csv từ máy, thêm cột ‘hotel’ và xuất file csv.

A screenshot of a computer

Description automatically generated

Hình 1. Thêm cột hotel vô 2 file csv

**Bước 2**: Đọc 2 file vừa xuất từ bước 1, sau đó tiến hành gộp 2 bộ dữ liệu H1, H2 và xuất ra file csv mới.

A screenshot of a computer code

Description automatically generated

Hình 1. Gộp 2 file csv thành 1 file csv mới

***Kết quả sau khi xử lý:***

A screenshot of a computer screen

Description automatically generated

Hình 1. Dữ liệu file hotel.csv

A screenshot of a computer

Description automatically generated

Hình 1. Dữ liệu file hotel.csv

* 1. Mô tả thuộc tính

Dữ liệu gồm 119390 dòng dữ liệu và 32 cột thuộc tính chứa quan sát các hoạt động đặt phòng từ các khách sạn H1 và khách sạn H2.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| 1 | IsCanceled | boolean | Có hủy đặt phòng không. Nhận 2 giá trị:  0: không hủy  1: hủy |
| 1 | LeadTime | integer | Số ngày từ khi đặt phòng tới khi nhận phòng. |
| 2 | ArrivalDateYear | integer | Năm nhận phòng. |
| 3 | ArrivalDateMonth | string | Tháng nhận phòng trong năm. |
| 4 | ArrivalDateWeekNumber | integer | Tuần nhận phòng tính từ đầu năm. |
| 5 | ArrivalDateDayOfMonth | integer | Ngày nhận phòng trong tháng. |
| 6 | StayInWeenkendNights | integer | Số đêm lưu trú cuối tuần. |
| 7 | StayIn WeekNights | integer | Số đêm lưu trú trong tuần. |
| 8 | Adults | integer | Số lượng người lớn. |
| 9 | Children | integer | Số lượng trẻ em. |
| 10 | Babies | integer | Số lượng em bé. |
| 11 | Meal | string | Loại bữa ăn được phục vụ. Có 3 giá trị:  BB (Bed & Breakfast): Chỉ bao gồm bữa sáng.  HB (Half Board): Bữa sáng và bữa tối được bao gồm trong giá phòng. Trong một số trường hợp, bạn có thể chọn nhận bữa trưa thay vì bữa sáng - khách sạn sẽ xác nhận điều này khi khách hàng đến.  FB (Full Board): Được bao gồm bữa sáng, bữa trưa và buổi tối. |
| 12 | Country | string | Quốc tịch khách hàng. |
|  | MarketSegment | string | Thể hiện các phân khúc khách hàng của khách sạn:  Offline TA/TO (Travel Agency/ Tour Operator): khách hàng từ các đại lý du lịch và công ty điều hành tour.  Online TA: khách hàng từ các đại lý du lịch nhưng bằng hình thức online trực tuyến.  Aviation: khách hàng đến từ các hãng hàng không.  Complementary: khách hàng được hưởng ưu đãi miễn phí.  Corporate: khách hàng đến từ các công ty, doanh nghiệp.  Groups: khách hàng đến từ các hội nhóm, tổ chức.  Direct: khách hàng đặt phòng trực tiếp tại khách sạn.  Undefined: không xác định. |
|  | DistributionChannel | string | Kênh phân phối phòng khách sạn. Có ba loại:  TA/TO (Travel Agency/Tour Operator): là kênh phân phối thông qua các đại lý du lịch và công ty điều hành tour.  Direct: là kênh phân phối trực tiếp (khách hàng đặt phòng trực tiếp tại khách sạn)  Corporate: là kênh phân phối thông qua các công ty, doanh nghiệp  Undefined: không xác định. |
| 13 | IsRepeatedGuest | Boolean | Là khách quay lại. Có 2 giá trị:  0: khách tới lần đầu.  1: khách cũ. |
| 14 | PreviousCancellations | integer | Số lượng lần hủy đặt phòng trước đó. |
| 15 | PreviousBookingsNotCanceled | integer | Số lượng lần không hủy phòng trước đó. |
| 16 | ReservedRoomType | string | Loại phòng đã đặt. |
| 17 | AssignedRoomType | string | Loại phòng được nhận. |
| 18 | BookingChanges | integer | Số lần thay đổi thông tin đặt phòng. |
| 19 | DepositType | string | Loại hình đặt cọc trước:  No Deposit: không có tiền đặt cọc trước.  Non-Refund: một khoản đặt cọc đã được thực hiện bằng giá trị của tổng chi phí lưu trú.  Refundable: một khoản đặt cọc đã được thực hiện với giá trị dưới tổng chi phí lưu trú. |
| 20 | Agent | integer | Mã của đại lý đặt phòng. |
| 21 | Company | interger | Mã của công ty đặt phòng. |
| 22 | DaysInWaitingList | integer | Số ngày đặt chỗ nằm trong danh sách chờ trước khi được xác nhận với khách hàng. |
| 23 | CustomerType | string | Loại khách hàng:  Group: Khi khách hàng đặt phòng theo một nhóm.  Transient: Khách tạm trú là những khách lưu trú qua đêm không có ý định ở lại lâu dài.  Transient party: Khách tạm trú theo nhóm lưu trú qua đêm không có ý định ở lại lâu dài. |
| 24 | ADR | float | Doanh thu bình quân hằng ngày (Avarage Daily Rate) được tính bằng cách chia tổng của tất cả các giao dịch lưu trú cho tổng số đêm lưu trú. |
| 25 | RequiredCarParkingSpaces | integer | Số lượng chỗ đỗ xe ô tô được yêu cầu. |
| 26 | TotalOfSpecialRequests | integer | Tổng số lượng yêu cầu đặc biệt. |
| 27 | ReservationStatus | string | Trạng thái đặt phòng hiện tại:  Check-Out: Khách hàng đã check-in nhưng đã rời đi.  No-Show: Khách hàng không check-in và đã thông báo lý do cho khách san.  Cancel: Khách hàng hủy lượt đặt phòng. |
| 28 | ReservationStatusDate | datetime | Ngày mà trạng thái cuối cùng của lượt đặt phòng được cập nhật. |
| 29 | hotel | string | Tên khách sạn:  H1: Khách sạn ở resort.  H2: Khách sạn trong thành phố. |

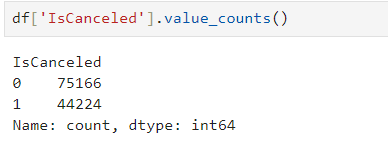
Bảng 1. Mô tả thuộc tính dữ liệu

* 1. Giới thiệu bài toán

Nhóm chúng em sẽ dự đoán mối quan hệ giữa khả năng hủy đặt phòng khách sạn **(IsCanceled)** với các thuộc tính khác để có thể khoanh vùng các đối tượng phục vụ thích hợp và đề ra các biện pháp phù hợp khi hủy đơn tăng cao.

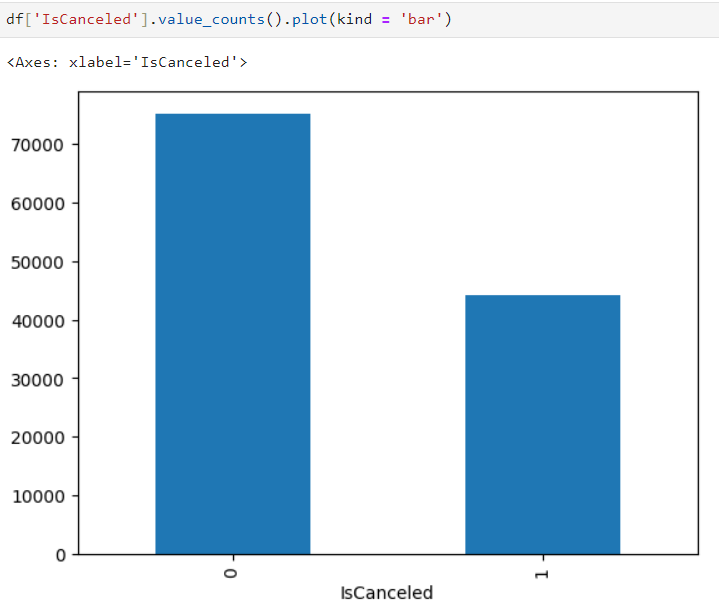
1. TIỀN XỬ LÝ DỮ LIỆU
   1. Trực quan hóa dữ liệu:
      1. Thống kê số lượng nhận phòng và hủy phòng:

* Số lượng trường hợp nhận và hủy phòng:



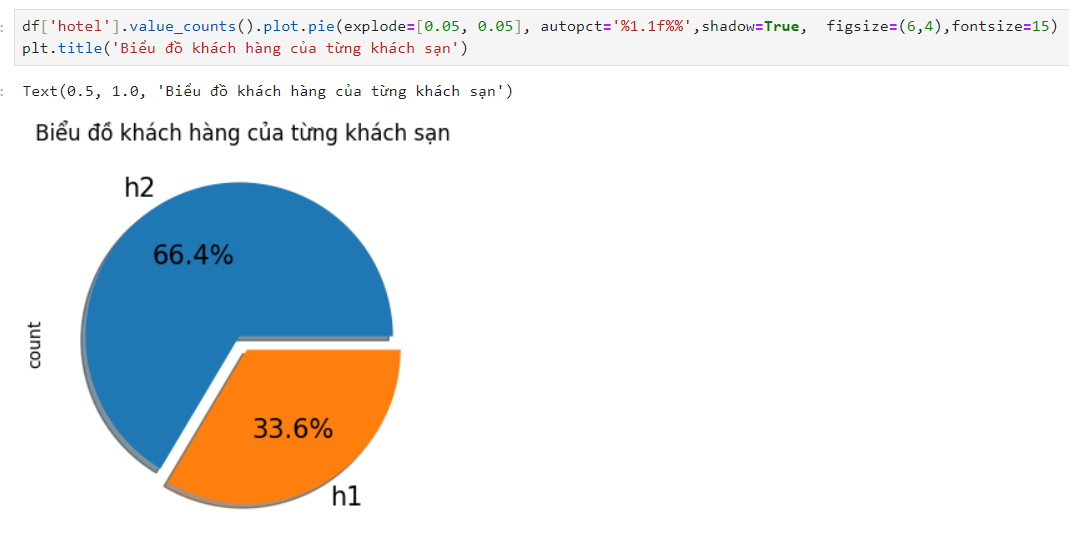
Hình 2. Số lượng nhận và hủy phòng

* Biểu đồ thể hiện:



Hình 2. Biểu đồ thể hiện số lượng nhận và hủy đặt phòng

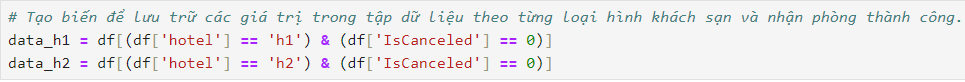
* ***Ta thấy***: Số lượng hủy nhận phòng chỉ bằng 2/3 số lượng nhận phòng.
  + 1. Thống kê loại hình khách sạn nào thu nhiều khách hàng nhất



Hình 2. Biểu đồ khách hàng từng khách sạn

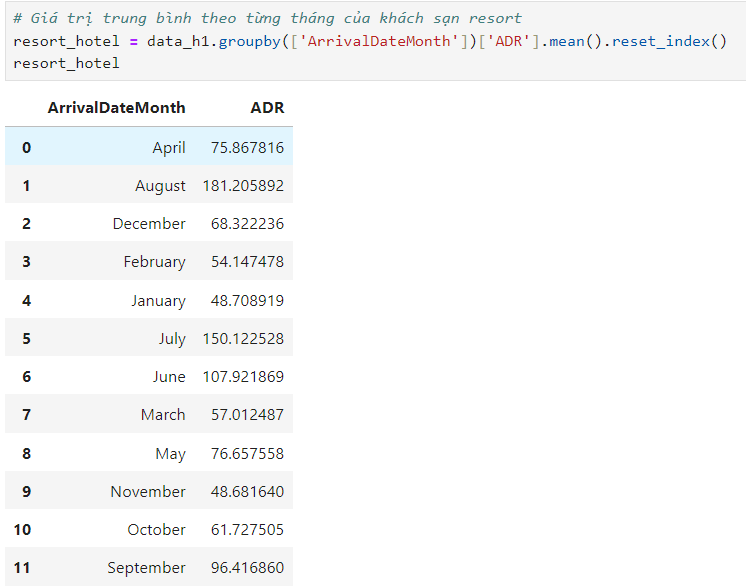
***Ta thấy***: Khách sạn thành phố (h2) được đặt phòng nhiều nhất.

* + 1. Thống kê doanh thu mỗi tháng trong tập dữ liệu
* Tạo biến để lưu trữ các giá trị trong tập dữ liệu theo từng loại hình khách sạn và nhận phòng thành công.

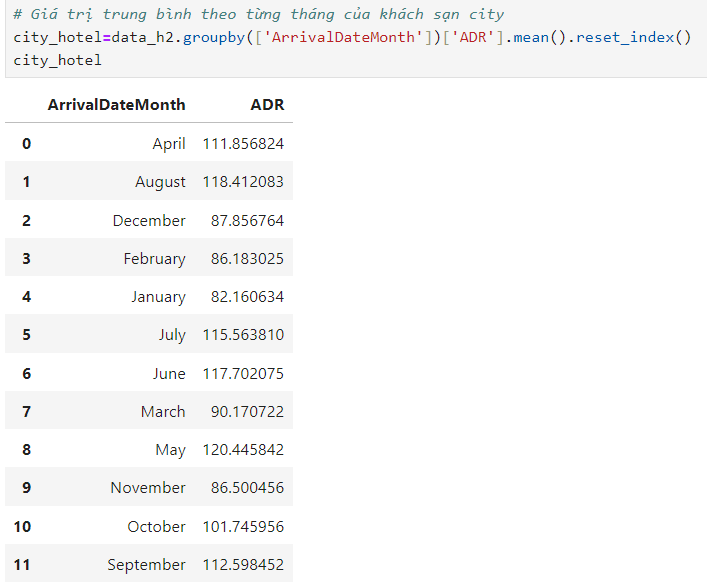


Hình 2. Tạo biến lưu trữ theo từng loại hình khách sạn và nhận phòng thành công

* Tính giá trị trung bình từng tháng của từng loại khách sạn.

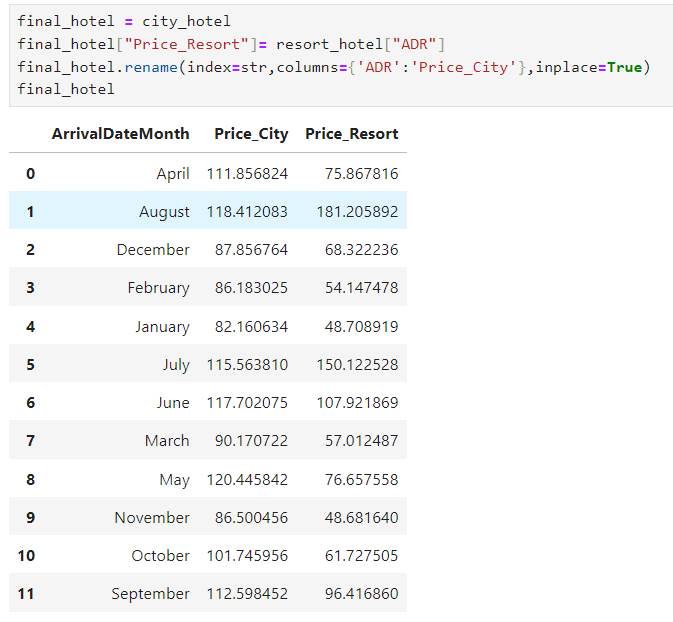


Hình 2. Thống kê giá trị doanh thu trung bình theo từng tháng của khách sạn h1



Hình 2. Thống kê giá trị doanh thu trung bình theo từng tháng của khách sạn h2

- Ghép hai **Dataframe** vừa tạo lại thành một tập **Dataframe** mới chứa doanh thu trung bình của từng tháng của hai loại hình khách sạn.



Hình 2. Thống kê doanh thu trung bình của từng tháng của hai loại hình khách sạn

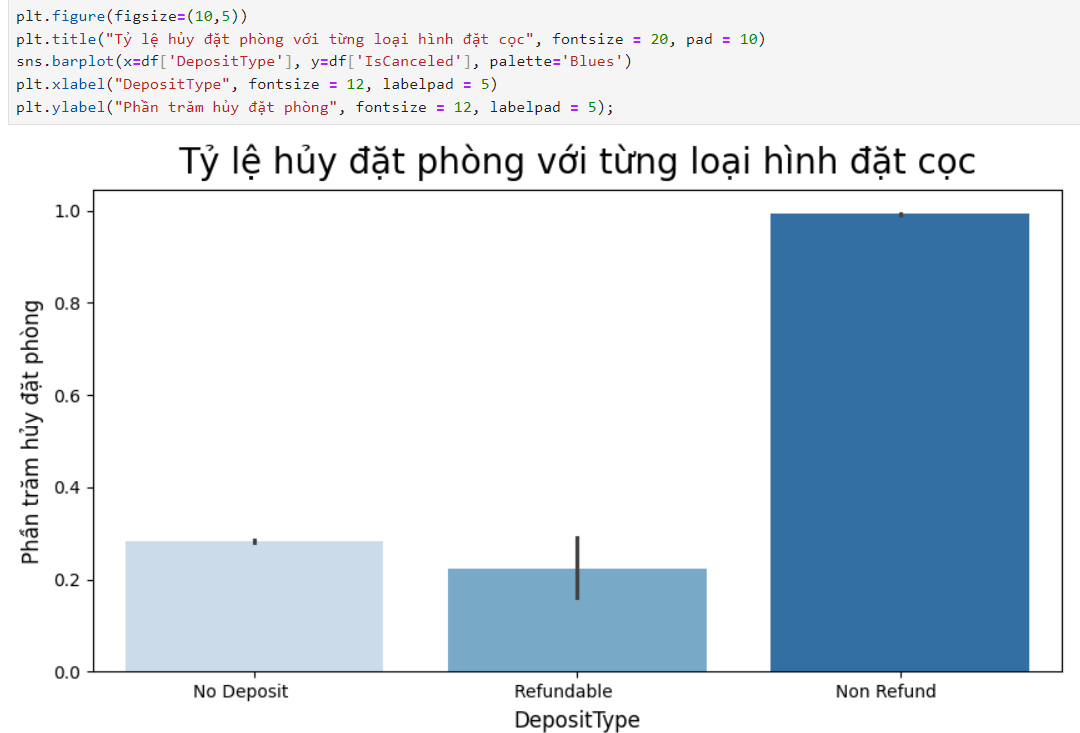
***- Kết quả:***



Hình 2. Biểu đồ doanh thu theo từng tháng của từng khách sạn

***Ta thấy:*** Doanh thu của khách sạn thành phố tương đối ổn định theo từng tháng trong từng năm. Doanh thu của khách sạn nghỉ dưỡng cao ở tháng 6, 7, 8 và thấp ở các tháng còn lại.

* + 1. Thống kê tỷ lệ hủy đặt phòng với từng loại hình đặt cọc



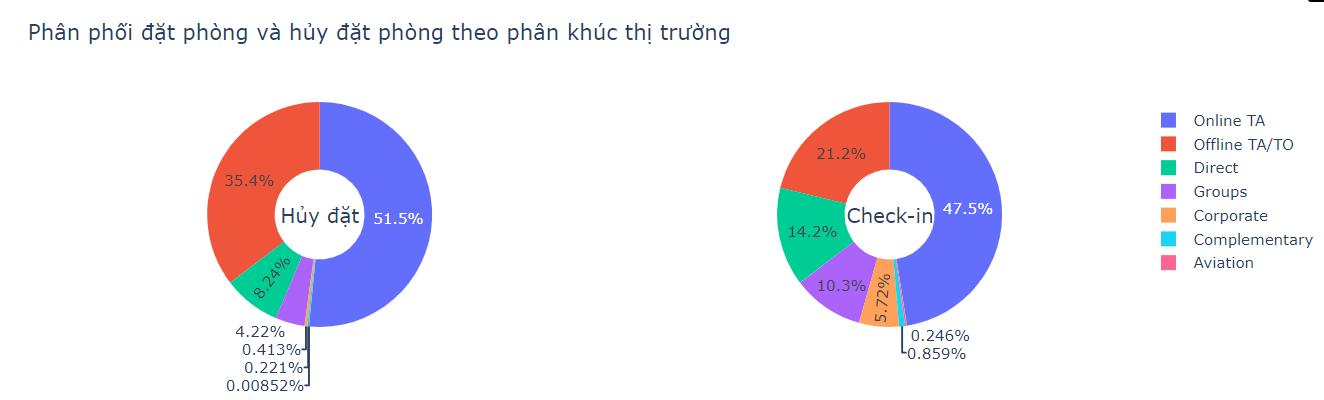
Hình 2. Biểu đồ thống kê tỷ lệ hủy đặt phòng với từng loại hình đặt cọc

***- Ta thấy:*** Lượng khách hàng hủy đặt cọc không nhận lại cọc có tỷ lệ không nhận phỏng cao nhất.

* + 1. Thống kê tỷ lệ phần trăm của các giá trị trong thuộc tính phân khúc thị trường với trạng thái hủy đặt phòng và nhận phòng.



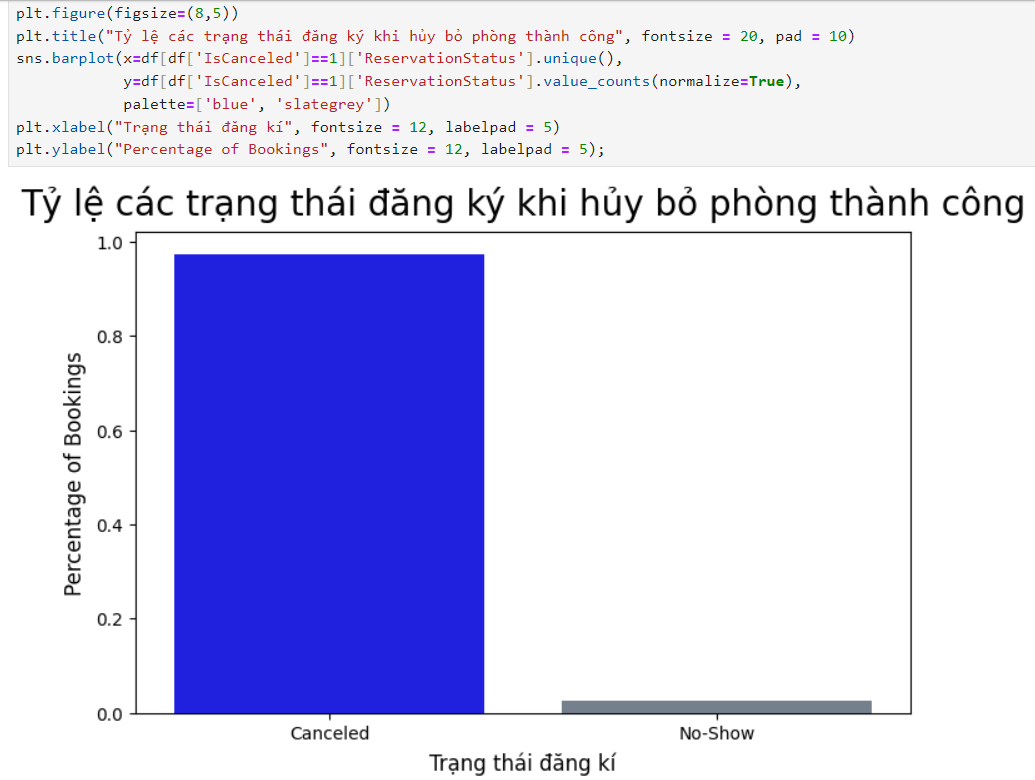
Hình 2. Code để vẽ biểu đồ thống kê tỷ lệ phần trăm của các giá trị trong thuộc tính phân khúc thị trường với trạng thái hủy đặt phòng và nhận phòng



Hình 2. Biểu đồ thống kê tỷ lệ phần trăm của các giá trị trong thuộc tính phân khúc thị trường với trạng thái hủy đặt phòng và nhận phòng

***Ta thấy:*** Tỷ lệ phân khúc thị trường trong việc nhận phòng và hủy phòng ở từng thị trường là tỷ lệ thuận với nhau. Và tỉ lệ đặt phòng và hủy phòng online là lớn nhất.

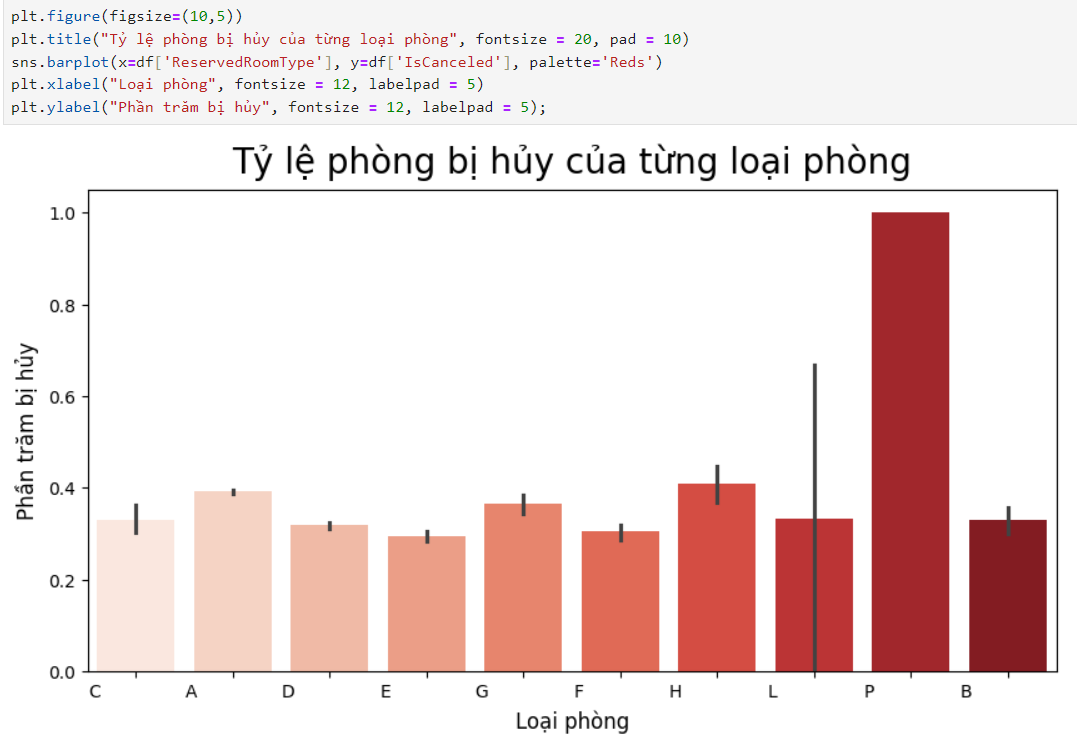
* + 1. Tỷ lệ các trạng thái đăng ký khi hủy bỏ phòng thành công



Hình 2. Biểu đồ tỷ lệ các trạng thái đăng ký khi hủy bỏ phòng thành công

***Ta thấy:*** Phần lớn các phòng bị hủy trước khi đến. Các dữ liệu chứa giá trị “**No – Show**” không đáng kể, có khả năng thuộc tính này sẽ không giúp ích được trong quá trình khai thác.

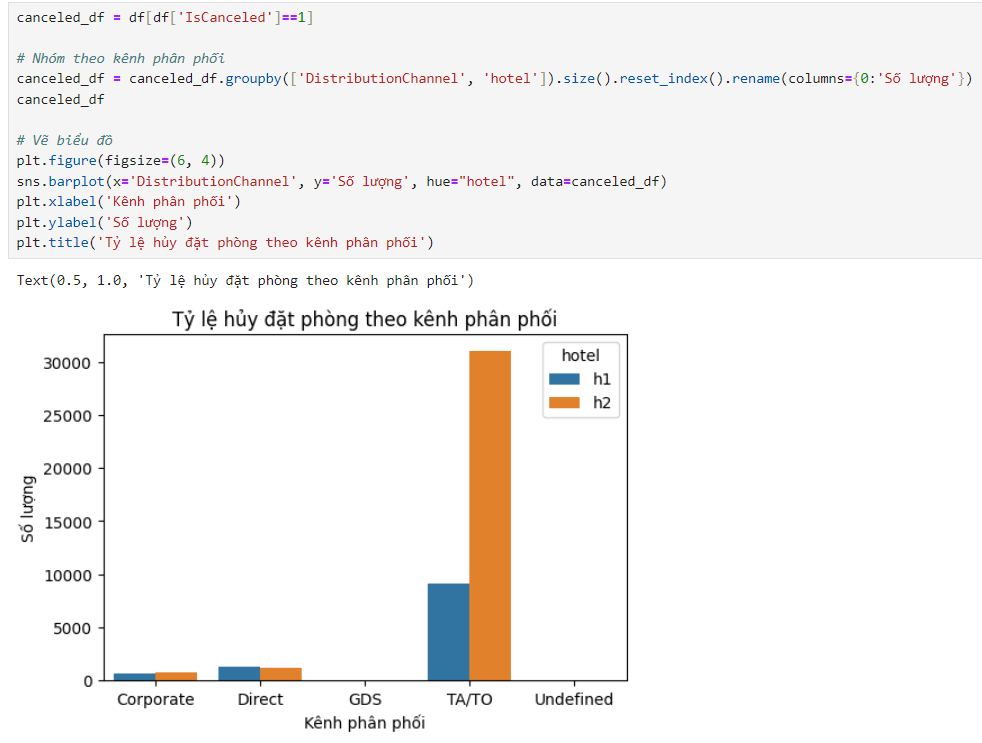
* + 1. Thống kê tỷ lệ phòng bị hủy



Hình 2. Biểu đồ thống kê tỷ lệ phòng bị hủy

***Ta thấy:*** Tỷ lệ khách hàng hủy đặt cao nhất đếu với phòng P và thấp nhất đối với phòng E.

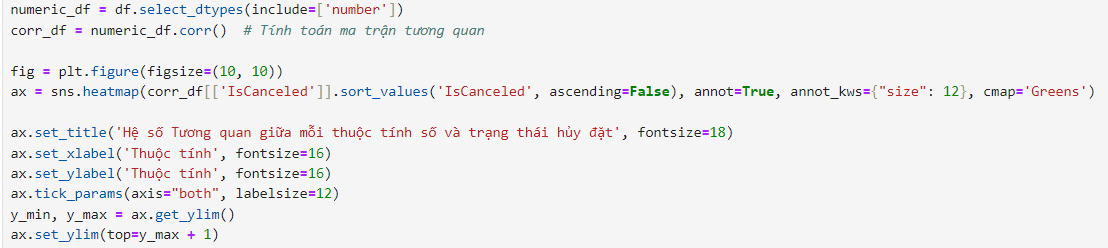
* + 1. Kênh phân phối nào có tỷ lệ hủy cao nhất



Hình 2. Biểu đồ thống kê kênh phân phối nào có tỷ lệ hủy cao nhất

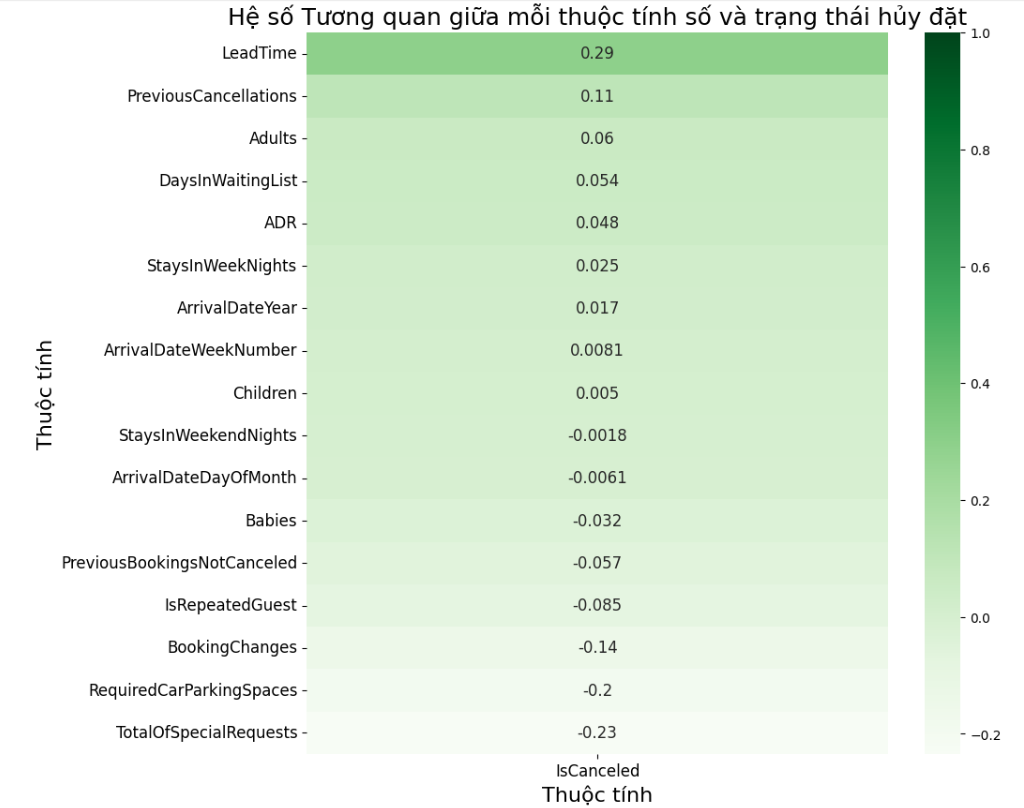
***Ta thấy:*** Kênh phân phối thông qua các đại lý du lịch và công ty điều hành tour có tỷ lệ bị hủy đặt phòng cao nhất ở cả 2 khách sạn.

* + 1. Thể hiện hệ số tương quan giữa giá trị hủy đặt phòng và các thuộc tính còn lại.
* Sử dụng hàm **heatmap** trong **Seaborn** để vẽ biểu đồ thể hiện sự tương quan giữa thuộc tính có hủy đặt phòng hay không và các thuộc tính khác trong tập dữ liệu. Biết rằng tương quan càng mạnh và đồng biến thì màu càng đậm và giảm dần về các tương quan mạnh nhưng nghịch biến với thuộc tính.



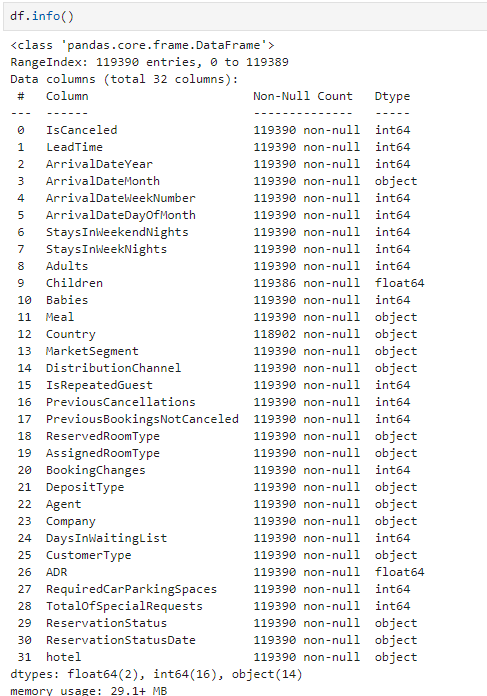
Hình 2. 15 Code để vẽ biểu đồ hệ số tương quan

* ***Kết quả:***



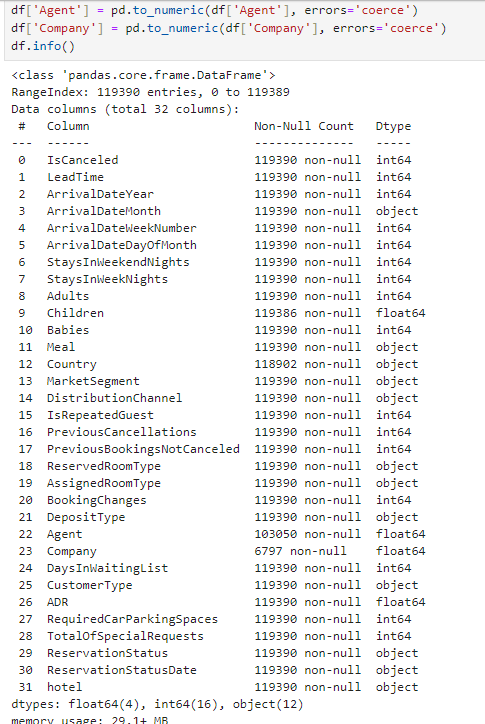
Hình 2. Biểu đồ hệ số tương quan giữa giá trị hủy đặt phòng và các thuộc tính còn lại

* ***Ta thấy:*** **LeadTime** có mối quan hệ tương quan cao nhất với việc có hủy đặt phòng. Có nghĩa là số ngày giữa thời điểm đặt phòng và ngày đến dự kiến tăng lên, khách hàng càng có nhiều thời gian hơn để hủy đặt phòng. **TotalOfSpecialRequests** có mối tương quan yếu với mục đích hủy đặt phòng. Có nghĩa rằng khách hàng càng có nhiều yêu cầu, đòi hỏi, gắn bó với khách sạn thì khả năng hủy đặt phòng càng thấp.
  1. Mô tả cơ bản về tập dữ liệu:
* Thông tin khái quát của dữ liệu:



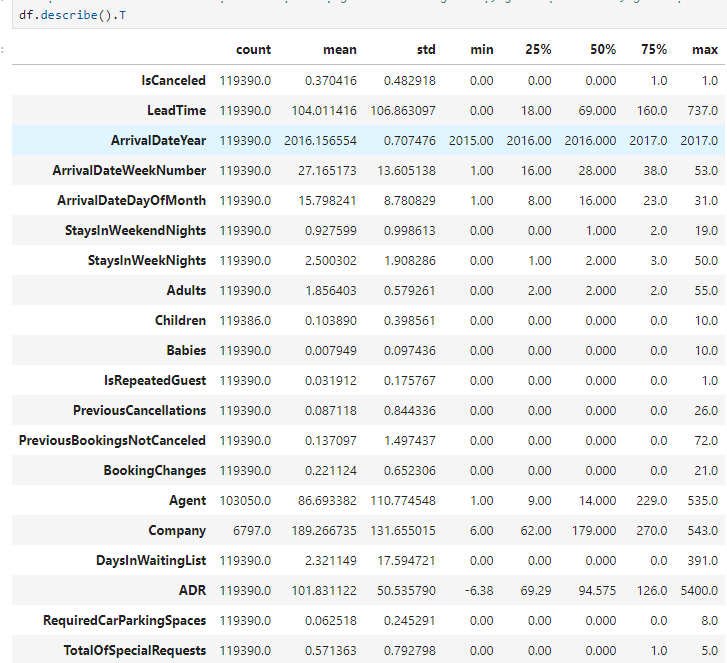
Hình 2. Thông tin tập dữ liệu ban đầu

Theo trên, nhóm thấy có 2 thuộc tính **Agent** và **Company** nhận các giá trị số nhưng lại được định dạng thành kiểu dữ liệu **object**. Nhóm sẽ tiến hành đặt lại kiểu dữ liệu cho 2 thuộc tính đó như sau:



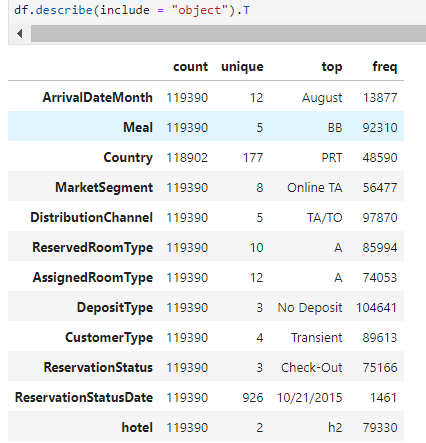
Hình 2. Thông tin tập dữ liệu sau khi thay đổi kiểu dữ liệu 2 cột

* ***Thuộc tính định lượng***: liệt kê các thông tin như: đếm số giá trị, giá trị lớn nhất, giá trị nhỏ nhất, trung bình, độ lệch chuẩn và tứ phân vị.



Hình 2. Thống kê thông tin thuộc tính định lượng

* ***Thuộc tính định tính***: liệt kê số giá trị, số lượng các giá trị khác nhau trong từng thuộc tính, giá trị có tần số xuất hiện cao nhất của thuộc tính đó và tần số của giá trị đó.



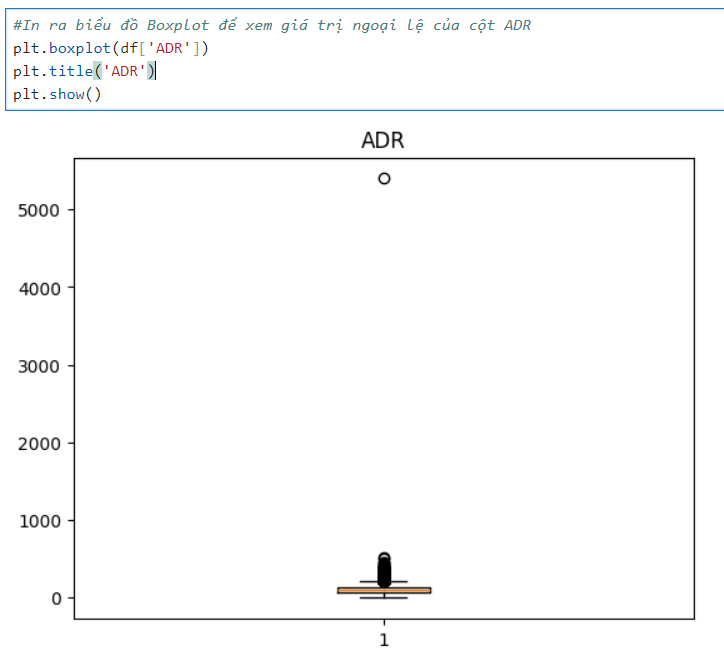
Hình 2. Thống kê thông tin thuộc tính định tính

* 1. Làm sạch dữ liệu:
     1. Xử lý dữ liệu bị nhiễu (outlier):

- Ngoại lệ (**Outlier**) chỉ một điểm dữ liệu có giá trị khác biệt đáng kể so với các điểm dữ liệu khác trong tập dữ liệu.

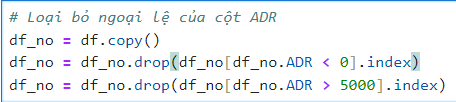
- Dựa vào bảng phân tích các thuộc tính định lượng trên, ta quan sát giá trị trung bình và giá trị nhỏ nhất hoặc lớn nhất của tất cả các cột, cột **ADR** có giá trị âm, và giá trị nhỏ nhất cực kỳ xa so với giá trị trung bình, điều này có nghĩa là giá trị nhỏ nhất là một ngoại lệ.

- Vẽ biểu đồ **Boxplot** để xem thông cột **ADR**:



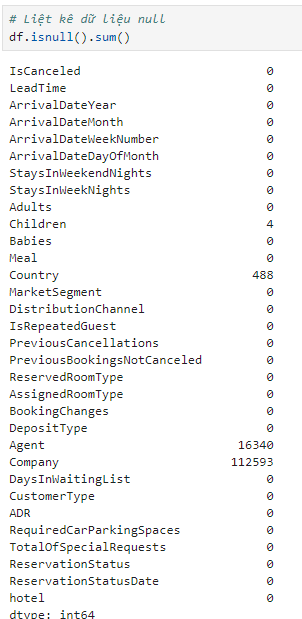
Hình 2. Biểu đồ Boxplot cho cột ADR

* ***Nhận xét:*** từ biểu đồ ta thấy, ngoại trừ giá trị âm thì cột **ADR** còn có một giá trị cực đại nằm xa so với các giá trị khác. Do đó giá trị cực đại đó cũng là 1 ngoại lệ.
* ***Cách xử lý***: xóa đi những giá trị ngoại lệ đó của cột ADR.



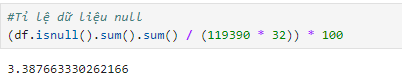
Hình 2. Loại bỏ cá ngoại lệ của cột ADR

* + 1. Xử lí dữ liệu bị thiếu:
* Liệt kê dữ liệu **NULL**:



Hình 2. Liệt kê các giá trị null

* Với tỉ lệ dữ liệu **NULL** trên toàn dữ liệu là:



Hình 2. Tỉ lệ dữ liệu null

* Tìm và sắp xếp các dữ liệu **Null**:



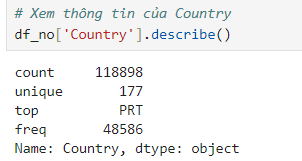
Hình 2. Liệt kê số giá trị và tỉ lệ null của của các cột

* ***Nhận xét***:
* Thuộc tính Children có 4 giá trị bị thiếu.
* Thuộc tính Country có 488 giá trị bị thiếu, chiếm 0,41% dữ liệu.
* Thuộc tính Agent có 16340 giá trị bị thiếu, chiếm 13,69% dữ liệu.
* Thuộc tính Company có 112593 giá trị bị thiếu, chiếm 94,31% dữ liệu.
* ***Cách xử lý***:
* Đối với thuộc tính Children, do số lượng giá trị Null quá ít với tỷ lệ xấp xỉ 0%, nên ta loại bỏ các dòng dữ liệu đó sẽ không ảnh hưởng đến kết quả của dữ liệu.



Hình 2. Xóa các dòng dữ liệu Children bị null

* Đối với thuộc tính Country, do có kiểu dữ liệu thuộc lớp object nên ta dùng giá trị xuất hiện nhiều nhất để điền vào những ô còn trống.

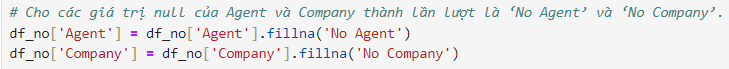


Hình 2. Xem thông tin của cột Country



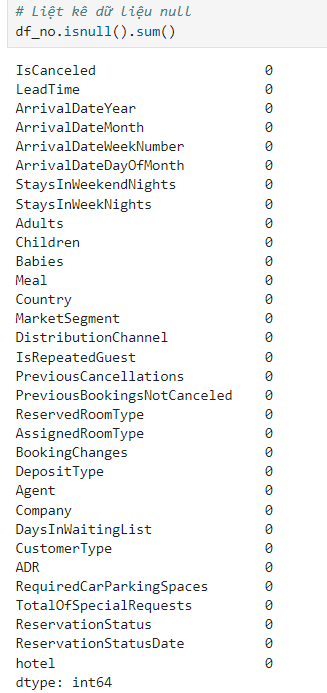
Hình 2. Thay thế giá trị rỗng của Country bằng PRT

* Đối với 2 thuộc tính Agent và Company do có số lượng giá trị null lớn chiếm lần lượt 13,69% và 94,31% trên tổng số dữ liệu nên ta sẽ cho các giá trị null thành lần lượt là ‘No Agent’ và ‘No Company’.



Hình 2. Biến đổi các giá trị null của Agent và Company thành giá trị mặc định

* ***Kết quả:***



Hình 2. Liệt kê lại các giá trị null

* + 1. Kiểm tra ý nghĩa của dữ liệu:
* Tại cùng một thời điểm thì các thuộc tính **adults, children** và **babies** có đồng thời bằng không hay không? Nếu có thì xóa các dòng dữ liệu đó.

A screenshot of a computer

Description automatically generated

Hình 2. Kiểm tra adults, children và babies có đồng thời bằng không

A screenshot of a computer

Description automatically generated

Hình 2. Xuất ra các dòng có adults, children và babies đồng thời bằng không

* ***Nhận xét:*** có 180 dòng dữ liệu nào mà cả 3 thuộc tính về người đều bằng 0.
* ***Cách xử lý***: xóa các dòng đó:



Hình 2. Xóa các dòng có 3 thuộc tính đó bằng 0

* + 1. Kiểm tra thuộc tính của dữ liệu:

A screenshot of a computer program

Description automatically generated

Hình 2. Liệt kê kiểu dữ liệu của các cột

* Ta nhận thấy có 1 số thuộc tính số lượng có kiểu dữ liệu là float trong khi các cột đó thể hiện giá trị thời gian hoặc là **yes/no** nên ta thay đổi kiểu dữ liệu cho các thuộc tính này.

A screen shot of a computer code

Description automatically generated

Hình 2. Thay đổi dữ liệu từ float sang int

* Tương tự, ta thấy thuộc tính **ReservationStatusDate** là kiểu dữ liệu ngày nhưng lại có kiểu dữ liệu là **object**, có thể gây ra ảnh hưởng trong quá trình phân tích nên ta tiếp tục thay đổi kiểu dữ liệu.



Hình 2. Thay đổi dữ liệu từ object sang date

* + 1. Xử lý dữ liệu bị trùng lặp:
* Kiểm tra dữ liệu có bị trùng lặp không.

A white rectangular object with black text

Description automatically generated

Hình 2. Đếm số dòng trùng nhau

* Ta thấy có 31980 dòng dữ liệu bị trùng lặp. Nhóm sẽ xử lý bằng cách xóa đi các dòng trùng lặp, chỉ giữ lại 1 dòng đầu trong các cặp trùng lặp.

A close-up of a white background

Description automatically generated

Hình 2. Xóa các dòng trùng lặp, chỉ giữ lại dòng đầu

* ***Kết quả:*** không còn dòng trùng lặp trong dữ liệu. Dữ liệu thu gọn còn 87224 dòng.

A screenshot of a computer

Description automatically generated

Hình 2. Dữ liệu sau khi xóa các dòng trùng nhau

* 1. Thêm thuộc tính cho dữ liệu:
* Tạo thuộc tính **ArrivalDate** là một ngày cụ thể theo format **%d/%m%Y**.

A screen shot of a computer

Description automatically generated

Hình 2. Tạo thuộc tính ArrivalDate

* Tạo thuộc tính **StatuMinusArrivalDate** là hiệu số giữa **ArrivalDate** và **ReservationStatusDate**.



Hình 2. Tạo thuộc tính StatuMinusArrivalDate

***Nhận xét:***

* Nếu khách hàng nhận phòng thì **status\_minus\_arrival\_date** là số ngày khách hàng nghỉ lại tại khách sạn.
* Nếu khách hàng hủy đặt phòng từ **status\_minus\_arrival\_date** là số ngày trước khi đến mà khách hàng đã hủy phòng.
* Tạo thuộc tính **TotalNights** là tổng số đêm ở vào ngày cuối tuần (**StaysInWeekendNights**) và ngày trong tuần (**StaysInWeekNights**).



Hình 2.

* Tạo thuộc tính **RoomMatch**: là sự kết hợp giữa hai thuộc tính **ReservedRoomType** và **AssignedRoomType** chỉ ra liệu giữa phòng đã đặt trước và phòng đã chỉ định của khách có giống nhau hay không.



Hình 2. Tạo thuộc tính RoomMatch

***Nhận xét:***

* Nếu phòng đã đặt trước và phòng đã chỉ định của khách giống nhau thì trả về 1.
* Nếu phòng đã đặt trước và phòng đã chỉ định của khách khác nhau thì trả về 0.
* Tạo thuộc tính **TotalPeople** là tổng số lượng người lấy từ các thuộc tính **adults, children, babies.**



Hình 2. Tạo thuộc tính TotalPeople

* Tạo thuộc tính **HaveChild** dựa trên các thuộc tính **Children, Babies** có số lượng nhiều hơn 0 trong tập dữ liệu.



Hình 2. Tạo thuộc tính HaveChild

* 1. Giảm thuộc tính của dữ liệu:
* Thực hiện tính toán hệ số tương quan giữa cột **IsCanceled** với các cột khác:

A computer screen shot of a computer code

Description automatically generated

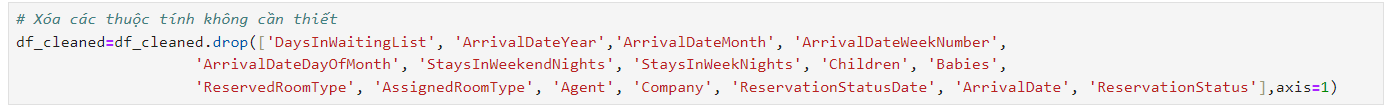
Hình 2. Tính hệ số tương quan

A screenshot of a computer

Description automatically generated

Hình 2. Hiển thị các hệ số tương quan theo thứ tự giảm dần

* Xóa thuộc tính **ArrivalDateYear**, **ArrivalDateMonth** và **ArrivalDateDayOfMonth** vì đã dùng các thuộc tính này để tạo **ArrivalDate**.
* Xóa thuộc tính **ArrivalDate, ReservationStatusDate** vì đã dùng các thuộc tính này để tạo **StatuMinusArrivalDate**.
* Xóa thuộc tính **ReservedRoomType** và **AssignedRoomType** vì đã tạo thuộc tính **RoomMatch**.
* Xóa thuộc tính **Children, Babies** vì đã dùng các thuộc tính này để tạo thuộc tính **TotalPeople** và **HaveChild**.
* Xóa thuộc tính **Agent** và **Company** vì chứa các dữ liệu không được định nghĩa cao.
* Xóa thuộc tính **StaysInWeekendNights** và **StaysInWeekNights** vì dùng các thuộc tính này để tạo **TotalNights**.
* Xóa thuộc tính **ArrivalDateWeekNumber** và **DaysInWaitingList** vì có mức tương quan rất thấp với thuộc tính **IsCanceled**.
* Xóa thuộc tính **ReservationStatus** vì có mức tương quan rất cao với thuộc tính **IsCanceled**.



Hình 2. Xóa các thuộc tính không cần thiết

* 1. Rời rạc hóa dữ liệu:
* Thống kê các thuộc tính định lượng:

A screenshot of a computer

Description automatically generated

Hình 2. Thống kê thuộc tính định lượng

* + 1. Thuộc tính LeadTime:

Chuyển đổi thuộc tính **LeadTime** về kiểu số nguyên và tạo ra các nhóm thời gian chia thời gian làm 4 phần nhỏ để thể hiện hành vi đặt phòng sớm hay muộn, sử dụng qcut để phân chia đổi ra ở mỗi nhóm (phương pháp làm mịn **binning**).

A computer screen shot of a computer

Description automatically generated

Hình 2. Chia giỏ thuộc tính LeadTime

- Các giỏ được đánh số theo số thứ tự từ 1 đến 4 cho các khoảng tăng dần:

A screenshot of a computer code

Description automatically generated

Hình 2. Chia dữ liệu theo giỏ thuộc tính LeadTime

* + 1. Thuộc tính Adults:

Chuyển đổi thuộc tính **Adults** về kiểu số nguyên và tạo ra các nhóm người lớn chia làm 2 phần nhỏ để thể hiện quy mô đoàn khách, sử dụng qcut để phân chia đổi ra ở mỗi nhóm (phương pháp làm mịn **binning**).

A white background with black text

Description automatically generated

Hình 2. Chia giỏ thuộc tính Adults

A screenshot of a computer code

Description automatically generated

Hình 2. Chia dữ liệu theo giỏ thuộc tính Adults

* + 1. Thuộc tính ADR:

Chuyển đổi thuộc tính **ADR** về kiểu số nguyên và tạo ra các nhóm doanh thu chia doanh thu làm 4 phần nhỏ, sử dụng **qcut** để phân chia đổi ra ở mỗi nhóm (phương pháp làm mịn binning).

A screenshot of a computer

Description automatically generated

Hình 2. Chia giỏ thuộc tính ADR

A screenshot of a computer code

Description automatically generated

Hình 2. Chia dữ liệu theo giỏ thuộc tính ADR

* + 1. Thuộc tính StatuMinusArrivalDate:

Chuyển đổi thuộc tính **StatuMinusArrivalDate** về kiểu số nguyên và tạo ra các nhóm theo số ngày đặt trước chia làm 3 phần nhỏ, sử dụng qcut để phân chia đổi ra ở mỗi nhóm (phương pháp làm mịn binning).

A screenshot of a computer

Description automatically generated

Hình 2. Chia giỏ thuộc tính StatuMinusArrivalDate

A close up of text

Description automatically generated

Hình 2. Chia dữ liệu theo giỏ thuộc tính StatuMinusArrivalDate

* + 1. Thuộc tính TotalNights:

Chuyển đổi thuộc tính **TotalNights** về kiểu số nguyên và tạo ra các nhóm tổng đêm ở chia làm 4 phần nhỏ đánh giá nhu cầu lưu trú của khách hàng, sử dụng **qcut** để phân chia đổi ra ở mỗi nhóm (phương pháp làm mịn binning).

A screenshot of a computer

Description automatically generated

Hình 2. Chia giỏ thuộc tính TotalNughts

A screen shot of a computer code

Description automatically generated

Hình 2. Chia dữ liệu theo giỏ thuộc tính TotalNights

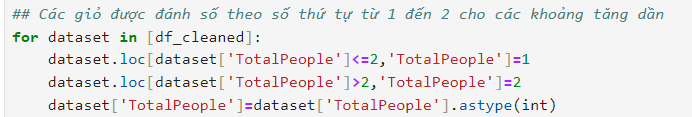
* + 1. Thuộc tính TotalPeople:

Chuyển đổi thuộc tính **TotalPeople** về kiểu số nguyên và tạo ra các nhóm tổng người ở chia làm 2 phần nhỏ, sử dụng qcut để phân chia đổi ra ở mỗi nhóm (phương pháp làm mịn binning).

A screenshot of a computer

Description automatically generated

Hình 2. Chia giỏ thuộc tính TotalPeople



Hình 2. Chia dữ liệu theo giỏ thuộc tính TotalPeople

* Các thuộc tính định lượng khác không rời rạc hóa dữ liệu vì:
* Với thuộc tính **IsCanceled**, **IsRepeatedGuest, RoomMatch, HaveChild** chỉ có 2 giá trị 1 và 0 nên không cần rời rạc hóa.
* Với các thuộc tính định lượng còn lại rời rạc hóa vì có quá nhiều giá trị trùng lặp (đặc biệt là 0) trong dữ liệu của thuộc tính.
  1. Dán nhãn số cho các thuộc tính định tính:
* Sử dụng hàm **Label Encoding** để thực hiện:

A screenshot of a computer

Description automatically generated

Hình 2. Dán nhãn số cho các thuộc tính định tính

* 1. Lấy tập mẫu cho dữ liệu (Sampling data):

Vì sự chênh lệch đáng kể giữa dữ liệu về việc nhận phòng và hủy phòng có thể dẫn đến bias (thiên vị) trong quá trình phát triển mô hình, chúng ta quyết định thực hiện một quy trình lọc ngẫu nhiên trên dữ liệu nhận phòng và chọn ra một số lượng mẫu tương đương với số lượng dữ liệu về hủy phòng. Điều này giúp đảm bảo rằng các mẫu dữ liệu sẽ đồng đều đại diện cho cả hai khía cạnh của quá trình đặt phòng và hủy phòng, tạo điều kiện thuận lợi cho quá trình xây dựng mô hình mà không gây ra sự thiên vị không mong muốn.

A screenshot of a computer program

Description automatically generated

Hình 2. Kiểm tra các giá trị trong IsCanceled

***Phương pháp thực hiện***:

A computer code with many colorful text

Description automatically generated with medium confidence

Hình 2. Thực hiện Sampling dữ liệu

***Kết quả***:

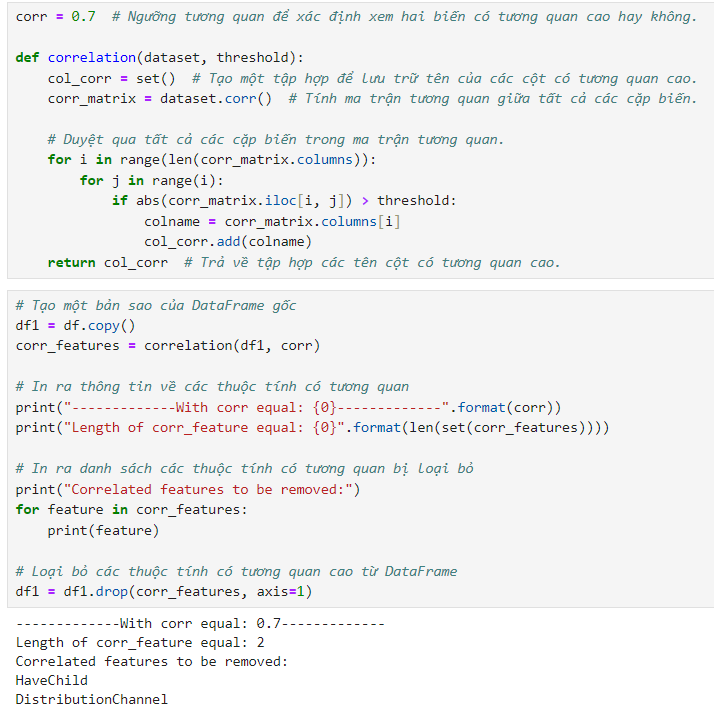
A screen shot of a computer

Description automatically generated

Hình 2. Kết quả sau sampling dữ liệu

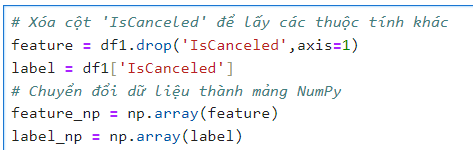
1. CÁC THUẬT TOÁN KHAI THÁC DỮ LIỆU
   1. Chuẩn bị dữ liệu để chạy các thuật toán:

* Loại bỏ các thuộc tính có tương quan cao ra khỏi thuộc tính.



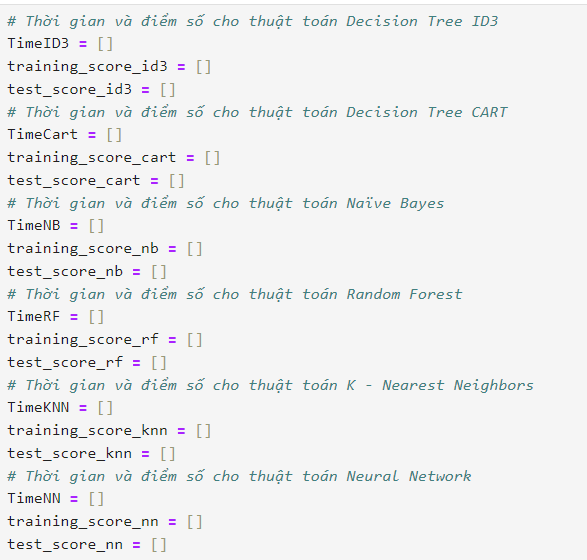
Hình 3. Loại bỏ các thuộc tính có tương quan cao ra khỏi thuộc tính

* Chuẩn bị mảng **feature\_np** chứa các thuộc tính của dữ liệu (không bao gồm nhãn) và một mảng **label\_np** chứa nhãn tương ứng.



Hình 3. Chuẩn bị các mảng

* Chuẩn bị các mảng dữ liệu lưu lại thời gian chạy, độ chính xác thuật toán qua các lần chạy:



Hình 3. Chuẩn bị các mảng dữ liệu lưu lại thời gian chạy, độ chính xác thuật toán

* 1. Thuật toán Cây quyết định (Decision Tree):
     1. Khái niệm:

- Thuật toán cây quyết định (**Decision Tree)** là một thuật toán học máy giám sát, không đối số, được sử dụng cho cả nhiệm vụ phân loại và hồi quy. Cây quyết định có cấu trúc cây phân cấp, bao gồm nút gốc, các nhánh, nút nội bộ và nút lá. Mục đích của thuật toán cây quyết định là tạo ra một mô hình huấn luyện có thể dự đoán lớp hoặc giá trị biến mục tiêu bằng cách học các quy tắc quyết định đơn giản từ dữ liệu đã biết trước đó (dữ liệu huấn luyện).

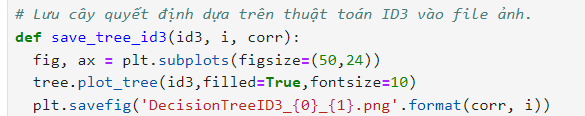
- Hai thuật toán nổi bật, quan trọng để tạo ra cây quyết định:

+ **ID3 (Iterative Dichotomiser 3**) là một thuật toán khá cổ điển trong việc xây dựng cây quyết định. Nó hoạt động bằng cách chọn thuộc tính tốt nhất để chia dữ liệu tại mỗi bước. **ID3** sử dụng thuật toán **Entropy** và **Information Gain** để đánh giá thuộc tính tốt nhất cho việc chia dữ liệu. Tuy nhiên, **ID3** có thể dễ dàng bị ảnh hưởng bởi nhiễu và không xử lý được dữ liệu liên tục.

+ **CART (Classification and Regression Trees)** là một thuật toán khác để xây dựng cây quyết định, nó có thể xây dựng cả cây phân loại (**classification**) và cây hồi quy (**regression**). **CART** sử dụng thuật toán **Gini Impurity** để đánh giá thuộc tính tốt nhất cho việc chia dữ liệu hoặc giá trị trung bình của các điểm dữ liệu để thực hiện hồi quy.

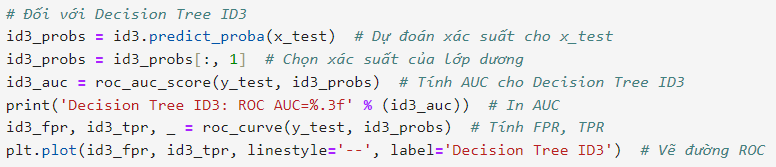
* + 1. Thực hiện:
       1. Giải thuật Decision Tree ID3:

- Chuẩn bị hàm lưu ảnh cây quyết định **ID3**.



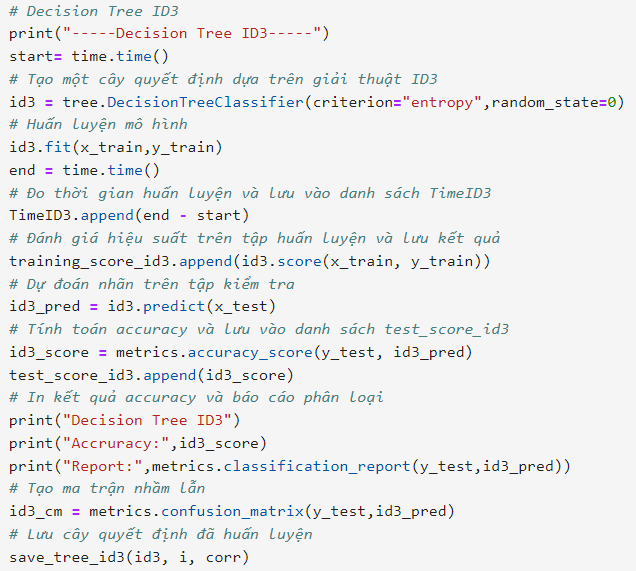
Hình 3. Hàm lưu ảnh cây quyết định ID3

- Chuẩn bị hàm để vẽ biểu đồ đường cong **ROC** cho **Decision Tree ID3**.



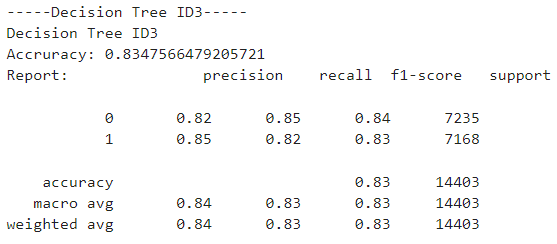
Hình 3. Hàm để vẽ biểu đồ đường cong ROC cho Decision Tree ID3

- Cài đặt thuật toán **Decision Tree ID3**.



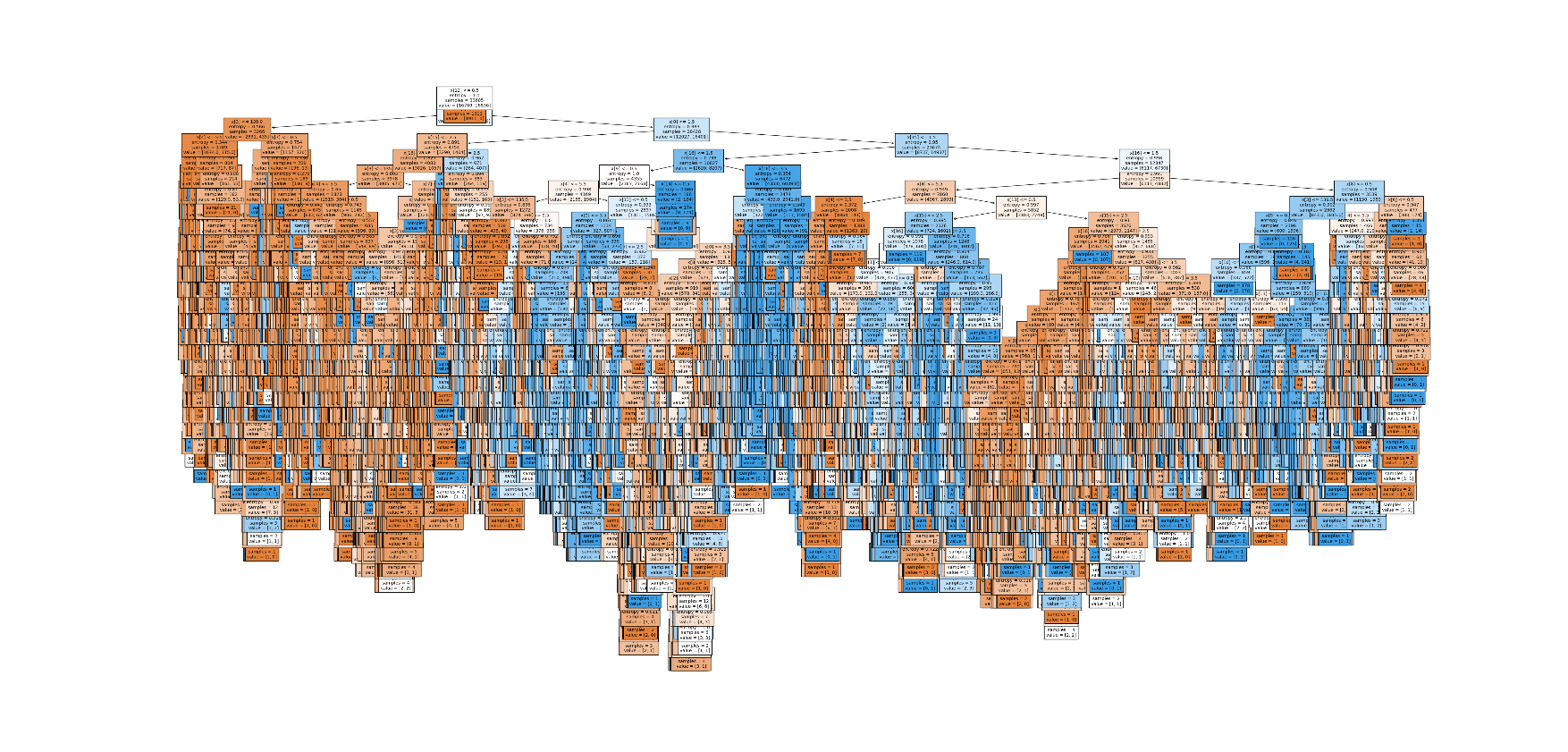
Hình 3. Cài đặt thuật toán Decision Tree ID3

- ***Kết quả chạy***:



Hình 3. Kết quả chạy ID3

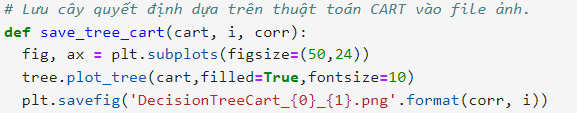
- Mô hình cây quyết định **ID3**:



Hình 3. Mô hình cây quyết định ID3

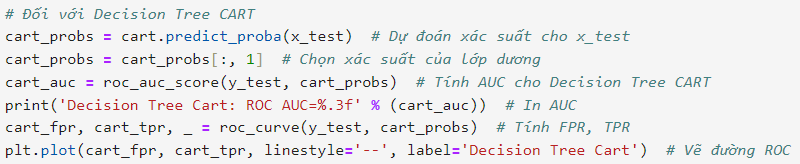
* + - 1. Giải thuật Decision Tree Cart:

- Chuẩn bị hàm lưu ảnh cây quyết định **CART**:



Hình 3. Hàm lưu ảnh cây quyết định CART

Chuẩn bị hàm để vẽ biểu đồ đường cong **ROC** cho **Decision Tree Cart.**



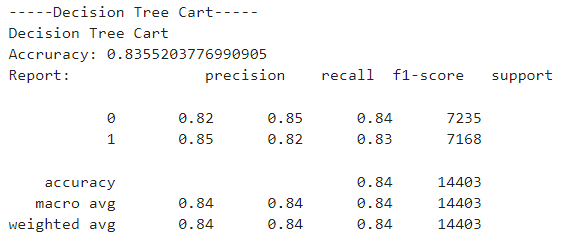
Hình 3. Hàm để vẽ biểu đồ đường cong ROC cho Decision Tree Cart

- Cài đặt thuật toán **Decision Tree CART**.



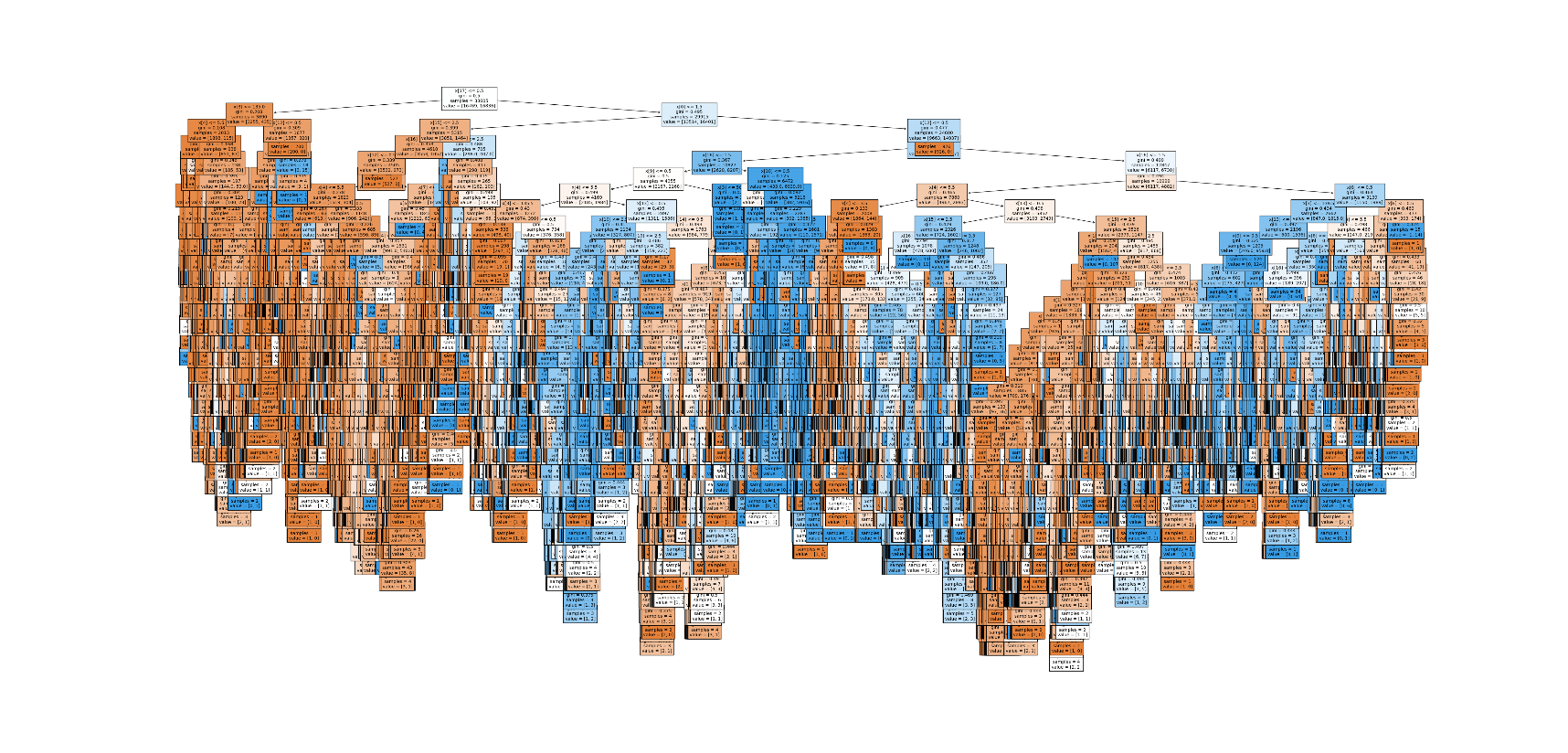
Hình 3. Cài đặt thuật toán Decision Tree CART

- ***Kết quả chạy:***



Hình 3. Kết quả chạy CART

***Mô hình cây quyết định CART***:



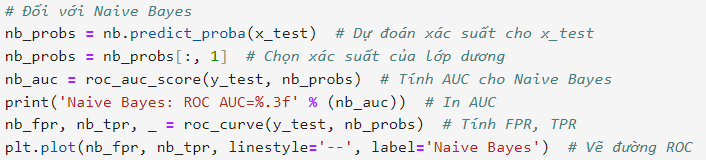
Hình 3. Mô hình cây quyết định CART

* 1. Thuật toán Naïve Bayes:
     1. Khái niệm:

- **Naïve Bayes** là một thuật toán phân loại cơ bản dựa trên Định lý **Bayes**. Nó giả định rằng tất cả các thuộc tính là độc lập với nhau khi đã biết nhãn lớp, điều này làm cho việc tính toán xác suất trở nên đơn giản. Mặc dù có giả định đơn giản này, **Naïve Bayes** thường hoạt động tốt trên các tập dữ liệu lớn với số lượng thuộc tính lớn. Thuật toán này đặc biệt hữu ích khi làm việc với các bài toán phân loại văn bản hoặc dữ liệu có cấu trúc đơn giản.

* + 1. Thực hiện:

- Chuẩn bị hàm để vẽ biểu đồ đường cong **ROC** cho **Naïve Bayes**.



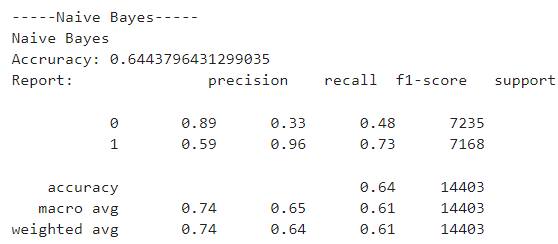
Hình 3. Hàm để vẽ biểu đồ đường cong ROC cho Naïve Bayes

- Cài đặt thuật toán **Naïve Bayes**.



Hình 3. Cài đặt thuật toán Naïve Bayes

- ***Kết quả chạy***:



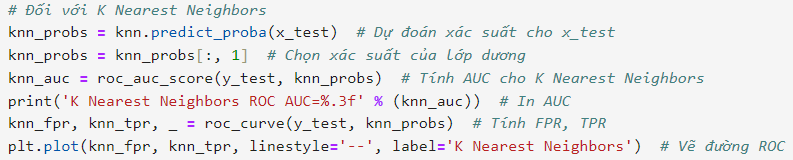
Hình 3. Kết quả chạy Naïve Bayes

* 1. Thuật toán K - Nearest Neighbors:
     1. Khái niệm:

- **K-Nearest** **Neighbors (KNN)** là một thuật toán phân loại không thông qua việc học mô hình mà dựa vào việc so sánh điểm dữ liệu mới với các điểm dữ liệu đã được gán nhãn trong tập huấn luyện. **KNN** sử dụng độ đo khoảng cách (thường là khoảng cách Euclidean) để tìm ra K điểm dữ liệu gần nhất và dùng đa số phiếu để quyết định nhãn của điểm dữ liệu mới. Thuật toán này đơn giản và dễ hiểu, nhưng có thể tốn nhiều thời gian khi phải tính toán khoảng cách với mỗi điểm dữ liệu trong tập huấn luyện.

* + 1. Thực hiện:

- Chuẩn bị hàm để vẽ biểu đồ đường cong **ROC** cho **K - Nearest Neighbors**.



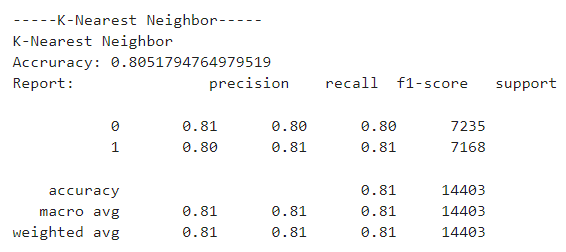
Hình 3. Hàm để vẽ biểu đồ đường cong ROC cho K - Nearest Neighbors

- Cài đặt thuật toán **K - Nearest Neighbors**.



Hình 3. Cài đặt thuật toán K - Nearest Neighbors

***- Kết quả chạy***:



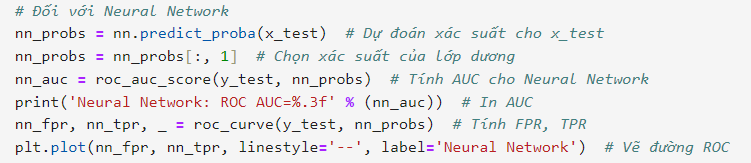
Hình 3. Kết quả chạy K - Nearest Neighbors

* 1. Thuật toán Neural Network:
     1. Khái niệm:

- Mạng nơ-ron (**Neural Network**) là một mô hình tính toán dựa trên cấu trúc của hệ thống nơ-ron trong não người. Mạng nơ-ron trong **Machine Learnin**g bao gồm các lớp nơ-ron liên kết với nhau qua các trọng số. Các thuật toán huấn luyện mạng nơ-ron như **Gradient Descent** được sử dụng để điều chỉnh các trọng số này sao cho mạng có thể học được cách biểu diễn dữ liệu và thực hiện các nhiệm vụ như phân loại hoặc dự đoán. **Neural Network** có thể mô hình hóa các mối quan hệ phức tạp trong dữ liệu, nhưng đòi hỏi nhiều dữ liệu huấn luyện và có thể cần nhiều thời gian và tài nguyên tính toán.

* + 1. Thực hiện:

- Chuẩn bị hàm để vẽ biểu đồ đường cong **ROC** cho **Neural Network**.



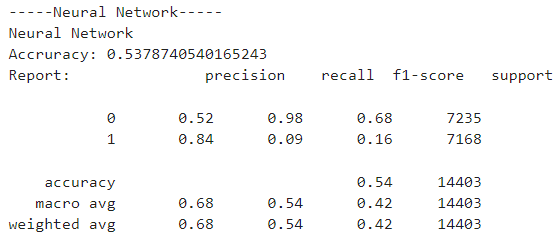
Hình 3. Hàm để vẽ biểu đồ đường cong ROC cho Neural Network

- Cài đặt thuật toán **Neural Network**.



Hình 3. Cài đặt thuật toán Neural Network

***- Kết quả chạy***:



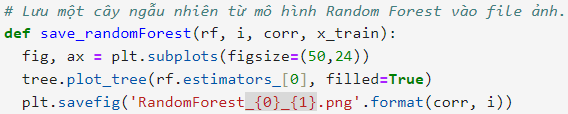
Hình 3. Kết quả chạy Neural Network

* 1. Thuật toán Random Forest:
     1. Khái niệm:

- **Random Forest** là một phương pháp học máy dựa trên việc kết hợp nhiều cây quyết định (**decision trees**) thành một mô hình dự đoán. Mỗi cây trong **Random Forest** được huấn luyện trên một tập dữ liệu con được lấy mẫu ngẫu nhiên từ tập dữ liệu huấn luyện ban đầu và sử dụng một số thuộc tính ngẫu nhiên trong quá trình xây dựng cây. Kết quả dự đoán cuối cùng là kết quả trung bình hoặc đa số phiếu của các cây con. **RandomForest** thường cho hiệu suất tốt trên nhiều loại dữ liệu và ít dễ bị **overfitting** so với một cây quyết định đơn lẻ. Đặc biệt, **Random Forest** có thể xử lý dữ liệu lớn, có nhiều thuộc tính và có khả năng xử lý nhiễu tốt.

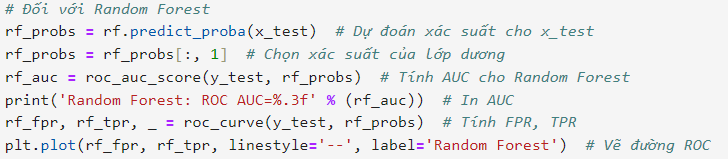
* + 1. Thực hiện:

- Chuẩn bị hàm lưu ảnh cây ngẫu nhiên **Random Forest**:



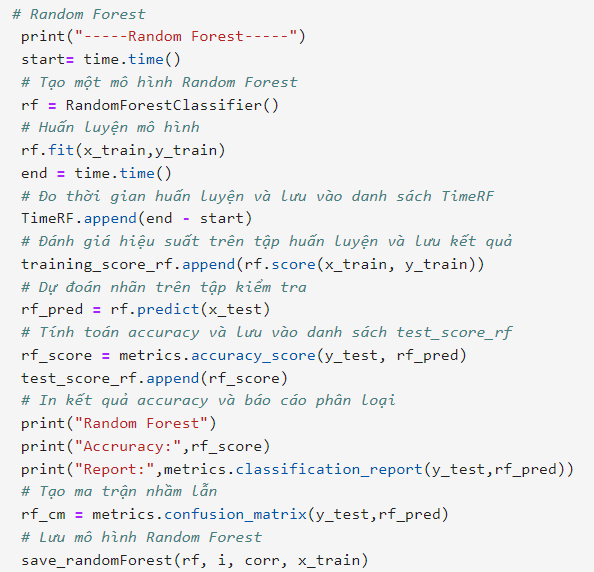
Hình 3. Hàm lưu ảnh cây ngẫu nhiên Random Forest

- Chuẩn bị hàm để vẽ biểu đồ đường cong **ROC** cho **Random Forest**.



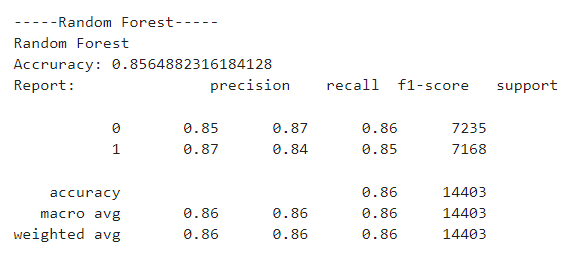
Hình 3. Hàm để vẽ biểu đồ đường cong ROC cho Random Forest

- Cài đặt thuật toán **Random Forest**.



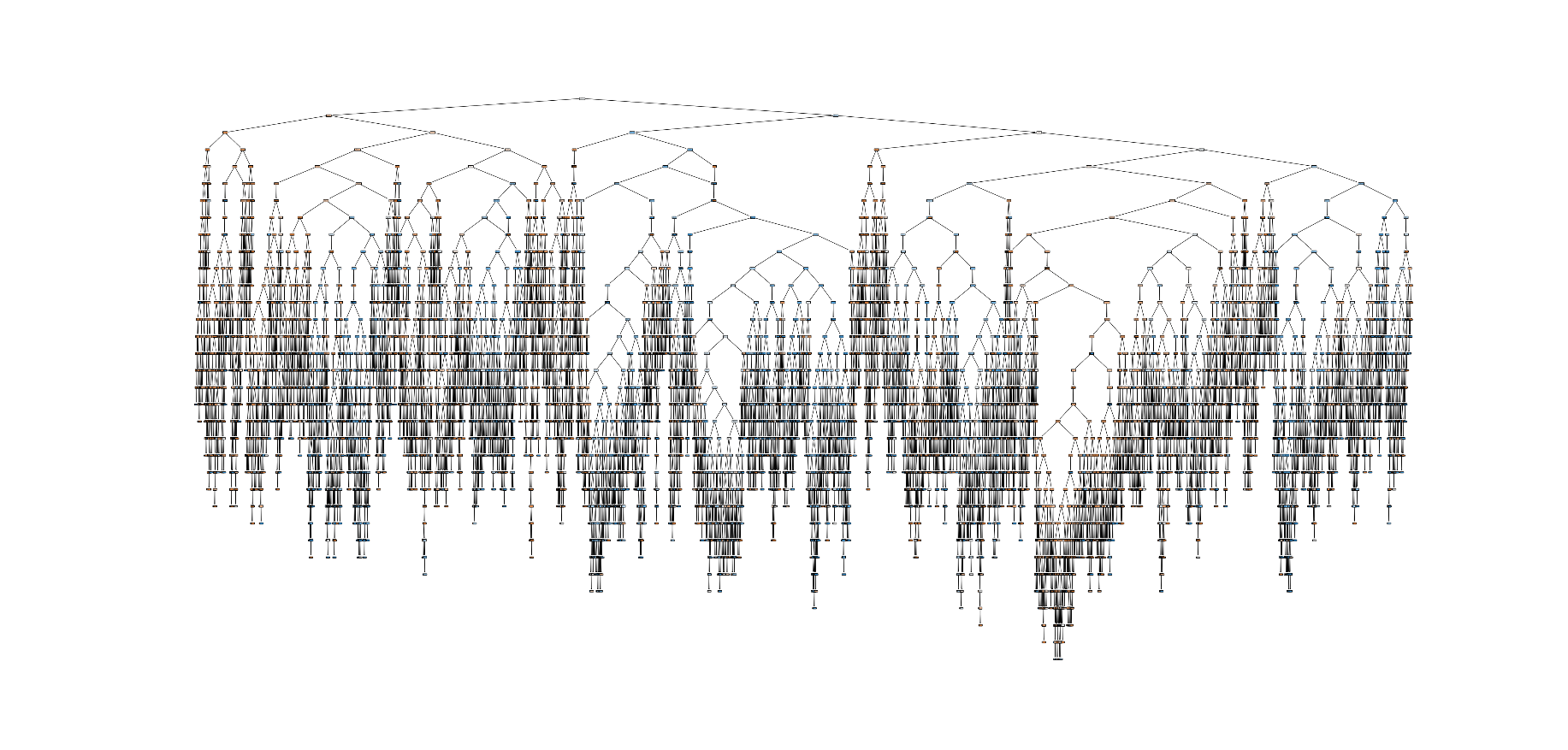
Hình 3. Cài đặt thuật toán Random Forest

- ***Kết quả chạy***:



Hình 3. Kết quả chạy Random Forest

***Mô hình cây ngẫu nhiên:***



Hình 3. Mô hình cây ngẫu nhiên

1. ĐÁNH GIÁ CÁC THUẬT TOÁN
   1. Đánh giá ma trận nhầm lẫn:

A screenshot of a computer

Description automatically generated

Hình 4. Ma trận nhầm lẫn của 6 thuật toán

***Nhận xét:*** Nhìn chung, ta thấy tỷ lệ **TPR (True Positive Rate**) và **FNR** (**False Negative Rate**) cao hơn **TNR (True Negative Rate**) và **FPR (False Positive Rate)** cho thấy mô hình phân loại có hiệu suất hợp lệ vì điều này phản ánh khả năng nhận biết chính xác của mô hình giữa các lớp phân loại, đặc biệt là khả năng nhận biết chính xác các trường hợp hủy đặt phòng.

* **TPR (Tỷ lệ đúng dương**): Tỷ lệ phần trăm của các dự đoán đúng là hủy đặt phòng (đã hủy) so với tổng số lượng thực sự là hủy đặt phòng. **TPR** cao cho thấy mô hình có khả năng nhận biết chính xác các trường hợp hủy đặt phòng, điều này rất quan trọng trong việc giảm thiểu rủi ro và đảm bảo lợi nhuận cho khách sạn.
* **FNR (Tỷ lệ sai âm):** Tỷ lệ phần trăm của các dự đoán sai là chưa hủy đặt phòng (đã hủy nhưng mô hình dự đoán chưa hủy) so với tổng số lượng thực sự là hủy đặt phòng. **FNR** thấp cho thấy mô hình ít bỏ sót các trường hợp hủy đặt phòng, điều này giúp khách sạn nắm bắt kịp thời các trường hợp hủy để có thể điều chỉnh và tối ưu hóa nguồn lực.
* **TNR (Tỷ lệ đúng âm):** Tương tự như **TPR** nhưng dành cho các dự đoán thuộc lớp chưa hủy đặt phòng. **TNR** cao cho thấy mô hình có khả năng nhận biết chính xác các trường hợp chưa hủy đặt phòng, giúp giảm thiểu số lượng dự đoán sai là hủy đặt phòng.
* **FPR (Tỷ lệ sai dương):** Tương tự như FNR nhưng dành cho các dự đoán thuộc lớp chưa hủy đặt phòng. **FPR** thấp cho thấy mô hình ít báo nhầm các trường hợp chưa hủy đặt phòng là hủy đặt phòng, giúp giảm thiểu số lượng dự đoán sai và cải thiện chất lượng dịch vụ khách hàng.
  1. Đánh giá tỷ lệ TPR trên tỷ lệ FPR trong biểu đồ đường cong ROC:

A graph of a graph

Description automatically generated with medium confidence

Hình 4. Biểu đồ đường cong ROC

***Nhận xét:***

* Đường cong **ROC** của thuật toán **Random Forest** nằm gần góc trên bên trái của biểu đồ nhất cho thấy mô hình của thuật toán này là tốt nhất.
* Đường cong **ROC** của thuật toán **Neural Network** nằm xa góc trên bên trái của biểu đồ nhất cho thấy mô hình của thuật toán này là tệ nhất.
  1. Đánh giá về thời gian chạy từng thuật toán:

A close-up of a computer screen

Description automatically generated

Hình 4. Thống kê thời gian chạy của từng thuật toán

A white background with black text

Description automatically generated

Hình 4. Biểu đồ thời gian chạy của từng thuật toán

***Nhận xét:***

* **Neural Network** tốn nhiều thời gian nhất.
* Thời gian chạy của **KNN** gần như bằng không do đặc tính **Lazy Learner** của chính nó.
  1. Đánh giá độ chính xác của thuật toán trên tập huấn luyện:

A white rectangular object with text

Description automatically generated

Hình 4. Thống kê độ chính xác trên tập train của từng thuật toán

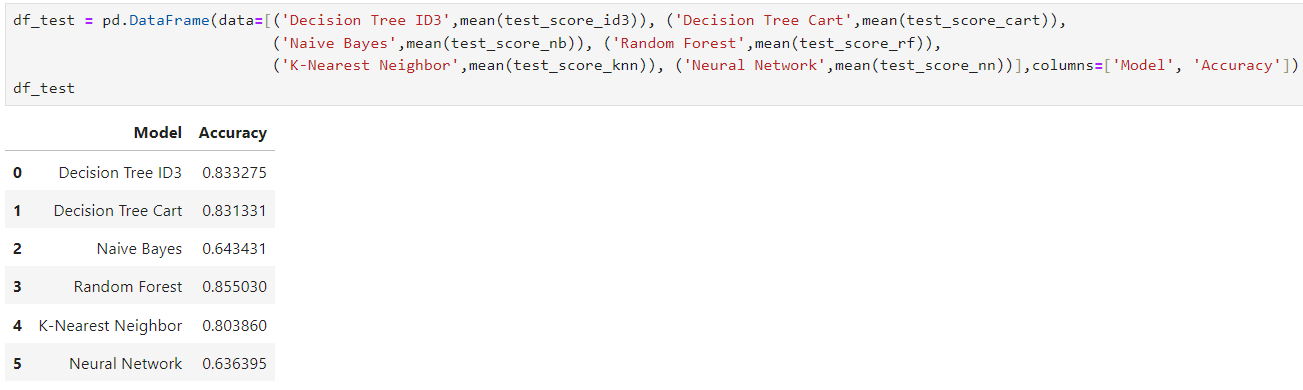
A graph of a bar

Description automatically generated with medium confidence

Hình 4. Biểu đồ độ chính xác trên tập train của từng thuật toán

***Nhận xét***:

* Độ chính xác của thuật toán cây quyết định **ID3, CART** và **Random Forest** cao nhất **(95,42%)**
* Độ chính xác của thuật toán **Neural Network** thấp nhất **(63,45%)**
  1. Đánh giá độ chính xác của thuật toán trên tập kiểm thử:



Hình 4. Thống kê độ chính xác trên tập test của từng thuật toán

A graph with blue squares

Description automatically generated with medium confidence

Hình 4. Biểu đồ độ chính xác trên tập test của từng thuật toán

***Nhận xét:***

* Độ chính xác của thuật toán cây quyết định **ID3, CART** và **Random Forest** cao nhất **(95,42%)**
* Độ chính xác của thuật toán **Neural Network** thấp nhất **(63,45%)**

==> Độ chính xác của các mô hình thuật toán trong kiểm thử gần như giống nhau với mô hình khi huấn luyện.

1. KẾT LUẬN
   1. Ưu - Nhược điểm của từng thuật toán:
      1. Decision Tree

***Ưu điểm:***

* Hiển thị rõ ràng**:** **Decision Tree** dễ hiểu và giải thích, giúp người dùng nắm bắt được quy luật mà thuật toán đã học từ dữ liệu.
* Không cần chuẩn hóa dữ liệu**:** Không yêu cầu tiền xử lý dữ liệu phức tạp như **Normalization** hoặc **Standardization**.
* Tính đơn giản**:** Thuật toán hoạt động dựa trên nguyên tắc cây quyết định, dễ dàng triển khai và duy trì.

***Nhược điểm:***

* Overfitting: Cây quyết định có thể quá phức tạp và dẫn đến **overfitting** nếu không được điều chỉnh đúng cách.
* Khả năng xử lý dữ liệu lớn hạn chế: Với lượng dữ liệu lớn, việc xây dựng và duy trì cây quyết định có thể trở nên khó khăn.
  + 1. Naïve Bayes

***Ưu điểm:***

* Hiệu quả với dữ liệu rời rạc:Đạt hiệu quả cao với dữ liệu dạng rời rạc, thường thấy trong dữ liệu khách sạn như loại phòng, ngày **check-in/check-out.**
* Khả năng xử lý nhanh: Thuật toán nhanh chóng và phù hợp cho việc phân loại lớn lượng dữ liệu.
* Không cần nhiều tính huống: Dữ liệu đầu vào không cần phải được chuẩn hóa trước khi sử dụng.

***Nhược điểm:***

* Giả định độc lập: Giả định rằng các đặc trưng là hoàn toàn độc lập với nhau, điều này không thực tế trong nhiều trường hợp.
* Độ chính xác thấp với dữ liệu liên quan: Khi các đặc trưng có mối liên hệ mạnh mẽ, **Naïve Bayes** có thể không hiệu quả.
  + 1. K – Nearest Neighbor

***Ưu điểm:***

* Thời gian chạy nhanh nhờ cơ chế **Lazy Learner**.
* Độ chính xác sau khi được chuẩn hóa đạt giá trị cao nhất.

***Khuyết điểm:***

* Độ chính xác giữa tập kiểm định và tập kiểm thử khi chưa chuẩn hóa dữ liệu có
* độ chênh lệch rất cao nên mô hình không phù hợp.
* KNN không hoạt động tốt trên tập dữ liệu lớn và kích thước cao. Hơn nữa, KNN
* còn nhạy cảm với nhiễu trong tập dữ liệu nên cần loại bỏ.
  + 1. Neural Network

***Ưu điểm:***

* Biểu đồ độ chính xác trên tập train của từng thuật toán **Neural Network** có thể học từ dữ liệu phức tạp và mô hình hóa các mối quan hệ gián tiếp giữa các biến.
* Cải thiện độ chính xác: Có thể đạt được độ chính xác cao hơn so với các thuật toán truyền thống.

***Nhược điểm:***

* Tốn nhiều thời gian chạy đối với cả ba trường hợp.
* Độ chính xác khi huấn luyện dữ liệu khá cao khi dữ liệu chưa được chuẩn hóa tuy nhiên độ chênh lệch với độ chính xác của tập thử nghiệm khá cao.
* Phụ thuộc vào phần cứng do cần bộ xử lý hoạt động cao khi xử lý song song trong mạng neural.
  + 1. Random Forest

***Ưu điểm:***

* Khả năng giảm thiểu overfitting: Bằng cách tạo ra nhiều cây quyết định và lấy trung bình, **Random Forest** giảm thiểu nguy cơ **overfitting**.
* Hiệu suất cao: Thường xuyên đạt được hiệu suất phân loại cao hơn so với các thuật toán đơn lẻ.
* Khả năng xử lý dữ liệu lớn: Tốt hơn so với **Decision Trees** và **KNN**.

***Nhược điểm:***

* Thời gian tính toán: Thời gian tính toán có thể dài hơn so với các thuật toán đơn giản hơn như **Naïve Bayes**.
* Khối lượng dữ liệu: Cần đủ lượng dữ liệu để đảm bảo hiệu suất tốt của mô hình.

==> Các thuật toán học máy như **Decision Tree, Naïve Bayes, K-Nearest Neighbor, Neural Network** và **Random Forest** đều mang lại những ưu và nhược điểm riêng biệt khi được áp dụng vào dự đoán khả năng hủy phòng của khách sạn. Việc lựa chọn thuật toán phù hợp đòi hỏi việc cân nhắc kỹ lưỡng dựa trên đặc điểm của dữ liệu và mục tiêu của dự án.

* 1. Hướng phát triển:
* Cập nhật và thử nghiệm với các mô hình học máy mới hơn, ví dụ như deep learning models, attention mechanisms, transfer learning.
* Phát triển ứng dụng hoặc trang web cho phép khách hàng đặt phòng trực tuyến, nhận thông báo và cập nhật về tình trạng đặt phòng của họ.

BẢNG PHÂN CHIA CÔNG VIỆC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Công việc | Phạm Quốc Hùng - 19521579 | Lê Trần Anh Quí - 21520094 | Trần Minh Quang - 21522519 | Đặng Lưu Hà - 21520798 |
| Chọn dataset, xác định đề tài | X | X | X | X |
| Thực hiện tổng quan đề tài |  |  |  | X |
| Trực quan hóa dữ liệu, Mô tả cơ bản về dữ liệu, làm sạch dữ liệu |  | X |  |  |
| Thêm, giảm, rời rạc hóa dữ liệu |  |  | X |  |
| Dán nhãn dữ liệu và lấy tập mẫu | X |  |  |  |
| Thực hiện thuật toán Decison Tree, Random Forest |  | X |  |  |
| Thực hiện thuật toán Naive Bayes và KNN |  |  | X | X |
| Thực hiện thuật toán Neural Network | X |  |  |  |
| Đánh giá các thuật toán |  | X | X |  |
| Ưu, nhược điểm và hướng phát triển | X |  |  |  |
| Viết báo cáo |  |  | X | X |

BẢNG ĐÁNH GIÁ CÁC THÀNH VIÊN TRONG NHÓM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thành viên | Phạm Quốc Hùng - 19521579 | Lê Trần Anh Quí - 21520094 | Trần Minh Quang - 21522519 | Đặng Lưu Hà - 21520798 |
| Điểm | 9 | 10 | 9 | 9 |

TÀI LIỆU THAM KHẢO

[1] Tài liệu lý thuyết của GV. Mai Xuân Hùng.

[2] Tài liệu thực hành của GV. Phạm Nguyễn Thanh Bình

[3] Random Forest Classification with Scikit-Learn: <https://www.datacamp.com/tutorial/random-forests-classifier-python>

[4] Binning Data in Pandas with cut and qcut: <https://datagy.io/pandas-cut-qcut/#:~:text=Pandas%20qcut%3A%20Binning%20Data%20into%20Equal-Sized%20Bins%20The,quantiles.%20This%20process%20is%20known%20as%20quantile-based%20discretization>