

ĐẠI HỌC UEH
TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ
KHOA CÔNG NGHỆ THÔNG TIN KINH
DOANH

UEH
UNIVERSITY

ĐỒ ÁN
MÔN: BIỂU DIỄN TRỰC QUAN DỮ LIỆU
ĐỀ TÀI: Starbucks Satisfaction Survey

Giảng viên hướng dẫn: Nguyễn An Té

Mã học phần: 22C1INF50908201

Nhóm sinh viên thực hiện: Nhóm 6

Nguyễn Đình Đại Nhơn	:	31201024507
Huỳnh Thị Cẩm Nhung	:	31201024508
Lê Thị Tuyết Nhung	:	31201024509
Đoàn Vũ Minh Thanh	:	31201020910
Đoàn Anh Thư	:	31201024519
Huỳnh Trần Anh Thy	:	31201024522

TP Hồ Chí Minh, ngày 17 tháng 12 năm 2022

MỤC LỤC NỘI DUNG

CHƯƠNG 1. TỔNG QUAN.....	9
1.1 Giới thiệu về tài	9
1.2 Tổng quan về bộ dữ liệu.....	9
CHƯƠNG 2. TIỀN XỬ LÝ DỮ LIỆU.....	11
2.1. Xử lý các cột dữ liệu bị thiếu	11
CHƯƠNG 3. THỐNG KÊ MÔ TẢ	20
3.1. Kiểm tra các đại lượng trung tâm.....	20
3.2. Kiểm tra hình dáng phân phối của bộ dữ liệu	22
3.3. Kiểm tra tính tương quan giữa các feature	38
3.4. Độ phân tán	40
CHƯƠNG 4. BIỂU DIỄN TRỰC QUAN DỮ LIỆU	43
4.1. Biểu đồ thể hiện mức độ trung thành của từng độ tuổi ứng với từng mức thu nhập	43
4.2. Biểu đồ thể hiện mức độ chi trả tiền của người những người thường xuyên đến cửa hàng	44
4.3. Mức độ ảnh hưởng của wifi đối với nhận định chất lượng dịch vụ	47
4.4. Biểu đồ đánh giá chất lượng dịch vụ theo từng độ tuổi	48
4.5. Biểu đồ phân tích đánh giá chất lượng cửa hàng	50
4.6. Biểu đồ đánh giá sự phân bổ dữ liệu giữa đánh giá về mức giá và trung bình dành thời gian bao nhiêu cho một lần ghé thăm cửa hàng.	52
4.7. Biểu đồ thể hiện tỷ lệ thời gian khách hàng ở lại quán so với sự hài lòng về Starbucks:..	53
4.8. Biểu đồ thể hiện tỷ lệ thời gian khách hàng ở lại quán so với sự không hài lòng về Starbucks: ..	55
4.9. Tỷ lệ khách hàng ở lại quán so với phương thức mua nước tại Starbucks	57
4.10. Biểu đồ thể hiện số lượng khách hàng của từng nhóm tuổi	59
4.12. Các tiêu chí đánh giá theo thang điểm 1-5	63
4.13. Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng giới tính:	66
4.14. Biểu đồ thể hiện tỷ lệ về tần suất ghé thăm của khách.....	67
4.15. Biểu đồ thể hiện đánh giá mức tiền theo từng nhóm khách hàng	69
4.16. Biểu đồ thể hiện tỷ lệ thời gian của từng khách hàng ở lại Starbucks	70
4.17. Biểu đồ thể hiện tần suất ghé thăm Starbucks của từng nhóm khách hàng	71
4.18. Biểu đồ thể hiện mức độ chi trả của khách hàng đến Starbucks theo từng nhóm tuổi	74
CHƯƠNG 5. GIẢM CHIỀU DỮ LIỆU	76
5.1. Phân phối chuẩn, One-Way ANOVA	76
5.1.1 Kiểm định điều kiện phân phối chuẩn tất cả các biến (Shapiro):	76

5.1.2 One-way ANOVA:	78
5.1.3. Hậu kiểm Tukey HSD:	79
5.2. Chi-squared Test	80
5.3. PCA	87
TÀI LIỆU THAM KHẢO	97
BÁNG PHÂN CÔNG	98

MỤC LỤC BIỂU ĐỒ

Biểu đồ 1: Biểu đồ thanh thể hiện giá trị trung bình các biến	20
Biểu đồ 2: Biểu đồ boxplot kiểm tra độ phân tán của bộ dữ liệu	21
Biểu đồ 3: Biểu đồ phân phối của feature gender	23
Biểu đồ 4: Biểu đồ phân phối của feature age	23
Biểu đồ 5: Biểu đồ phân phối của feature status	24
Biểu đồ 6: Biểu đồ phân phối của feature income	24
Biểu đồ 7: Biểu đồ phân phối của feature visitNo	25
Biểu đồ 8: Biểu đồ phân phối của feature Method	25
Biểu đồ 9: Biểu đồ phân phối của feature timeSpend	26
Biểu đồ 10: Biểu đồ phân phối của feature location	26
Biểu đồ 11: Biểu đồ phân phối của feature membershipCard	27
Biểu đồ 12: Biểu đồ phân phối của feature itemPurchaseCoffee	27
Biểu đồ 13: Biểu đồ phân phối của feature itemPurchaseCold	28
Biểu đồ 14: Biểu đồ phân phối của feature itemPurchasePastries	28
Biểu đồ 15: Biểu đồ phân phối của feature itemPurchaseJuices	29
Biểu đồ 16: Biểu đồ phân phối của feature itemPurchaseSandwiches	29
Biểu đồ 17: Biểu đồ phân phối của feature spendPurchase	30
Biểu đồ 18: Biểu đồ phân phối của feature productRate	30
Biểu đồ 19: Biểu đồ phân phối của feature priceRate	31
Biểu đồ 20: Biểu đồ phân phối của feature promoRate	31
Biểu đồ 21: Biểu đồ phân phối của feature ambianceRate	32
Biểu đồ 22: Biểu đồ phân phối của feature wifiRate	32
Biểu đồ 23: Biểu đồ phân phối của feature serviceRate	33
Biểu đồ 24: Biểu đồ phân phối của feature chooseRate	33
Biểu đồ 25: Biểu đồ phân phối của feature loyal	34
Biểu đồ 26: Biểu đồ heatmap thể hiện tương quan giữa các biến	39
Biểu đồ 27: Biểu đồ phân tán giữa itemPurchaseCold và itemPurchaseCoffee	40
Biểu đồ 28: Biểu đồ phân tán giữa priceRate và productRate	41
Biểu đồ 29: Biểu đồ phân tán giữa ambianceRate và productRate	41
Biểu đồ 30: Biểu đồ phân tán giữa serviceRate và ambianceRate	42
Biểu đồ 31: Biểu đồ phân tán giữa serviceRate và wifiRate	42
Biểu đồ 32: Biểu đồ thể hiện mức độ trung thành của từng độ tuổi ứng với từng mức thu nhập	43
Biểu đồ 33: Biểu đồ thể hiện mức độ chi trả tiền của người nhũng người thường xuyên đến cửa hàng	46
Biểu đồ 34: Biểu đồ phân tán thể hiện mức độ ảnh hưởng của wifi đối với nhận định chất lượng dịch vụ	48
Biểu đồ 35: Biểu đồ đánh giá chất lượng dịch vụ theo từng độ tuổi	50
Biểu đồ 36: Biểu đồ thanh phân tích đánh giá chất lượng cửa hàng	52
Biểu đồ 37: Biểu đồ đánh giá sự phân bố dữ liệu giữa đánh giá về mức giá và trung bình dành thời gian bao nhiêu cho một lần ghé thăm cửa hàng	53
Biểu đồ 38: Biểu đồ thể hiện sự hài lòng của KH so với thời gian ở lại quán	55
Biểu đồ 39: Biểu đồ thể hiện sự không hài lòng của KH so với thời gian ở lại quán	56
Biểu đồ 40: Biểu đồ thể hiện lệ khách hàng ở lại quán so với phương thức mua nước tại Starbucks	58
Biểu đồ 41: Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi	61

Biểu đồ 42: Biểu đồ thể hiện Tỷ lệ mức độ ghé thăm cửa hàng theo từng nhóm khoảng cách ...	62
Biểu đồ 43: Biểu đồ thể hiện số lượt đánh giá các tiêu chí theo thang điểm từ 1 đến 5.....	65
Biểu đồ 44: Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng giới tính	67
Biểu đồ 45: Biểu đồ thể hiện tỷ lệ về tần suất ghé thăm của khách.....	68
Biểu đồ 46: Biểu đồ thể hiện đánh giá mức tiền theo từng nhóm khách hàng	70
Biểu đồ 47: Biểu đồ thể hiện tỷ lệ thời gian của từng khách hàng ở lại Starbucks	71
Biểu đồ 48: Biểu đồ thể hiện tần suất ghé thăm Starbucks của từng nhóm khách hàng	73
Biểu đồ 49: Biểu đồ thể hiện mức độ chi trả của khách hàng đến Starbucks theo từng nhóm tuổi	75
Biểu đồ 50: Biểu đồ sau khi chuẩn hóa dữ liệu	96

MỤC LỤC HÌNH

Hình 1: Cột của bộ dữ liệu gốc	11
Hình 2: Code Python thay thế các giá trị bị thiếu thành giá trị “none”	11
Hình 3: Code Python xóa các cột không cần thiết trong bảng dữ liệu.....	11
Hình 4: Code Python đọc ra những cột cần chuẩn hóa.....	12
Hình 5: Kết quả những cột cần chuẩn hóa	12
Hình 6: Code Python chuẩn hóa các cột đã đọc và định dạng lại kiểu dữ liệu.....	12
Hình 7: Code Python xử lý dữ liệu	13
Hình 8: Code Python đếm các cách thức được ghi nhận từ khách hàng	13
Hình 9: Code Python sắp xếp lại theo thứ tự tăng dần	14
Hình 10: Code Python tách tên các phương thức và gom chúng vào trong 1 danh sách.....	14
Hình 11: Danh sách các phương thức	14
Hình 12: Code Python thêm cột promomethod_list vào bảng dữ liệu hiện tại.....	14
Hình 13: Code Python xóa cột “19. How do you come to hear of promotions at Starbucks? Check all that apply.” và cột “count”	15
Hình 14: Code Python xử lý dữ liệu và tách cột “10. What do you most frequently purchase at Starbucks?”	15
Hình 15: Code Python xem kết quả xử lý	16
Hình 16: Code Python xóa cột “10. What do you most frequently purchase at Starbucks?” và cột “count”.....	16
Hình 17: Code Python sắp xếp lại cột theo thứ tự câu hỏi.....	17
Hình 18: Code Python đổi tên các cột	18
Hình 19: Code Python sắp xếp lại các index cho phù hợp	19
Hình 20: Code Python kiểm tra giá trị trung bình	20
Hình 21: Code Python kiểm tra độ phân tán của bộ dữ liệu	21
Hình 22: Code Python kiểm tra hình dáng phân phối của bộ dữ liệu	22
Hình 23: Code Python vẽ biểu đồ parallel sets biến age, income, loyal	43
Hình 24: Code Python tạo bảng gồm cột visitNo và cột spendPurcse và tính tổng theo 2 thuộc tính thành cột count.....	45
Hình 25: Code Python vẽ biểu đồ phân tán mức độ ảnh hưởng của wifi đối với nhận định chất lượng dịch vụ	47
Hình 26: Code Python tạo bảng chéo thể hiện sự tương quan giữa biến “age” và biến “serviceRate”	49
Hình 27: Code Python tạo biểu đồ thanh đánh giá chất lượng dịch vụ theo từng độ tuổi	49
Hình 28: Code Python định nghĩa các hàm plot_bar, organize_plot_data và generate_plot và vẽ biểu đồ.....	51
Hình 29: Code Python vẽ biểu đồ thể hiện tỷ lệ thời gian khách hàng ở lại quán so với sự hài lòng về Starbucks	54
Hình 30: Code Python vẽ biểu đồ thể hiện sự không hài lòng của KH so với thời gian ở lại quán	55
Hình 31: Code Python gom nhóm các phương thức mua nước và thời gian khách hàng ở lại quán	57
Hình 32: Code Python tổng hợp số lượng khách hàng theo nhóm tuổi	60
Hình 33: Code Python khai báo biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi	60
Hình 34: Khai báo biểu đồ Tỷ lệ mức độ ghé thăm cửa hàng theo từng nhóm khoảng cách	61
Hình 35: Xử lý dữ liệu điểm đánh giá các tiêu chí	64
Hình 36: Khai báo biểu đồ cột chồng cho điểm các tiêu chí đánh giá.....	65

Hình 37: Code Python tạo bảng chéo để tổng hợp số lượng khách hàng đối với từng giới tính..	66
Hình 38: Code Python vẽ biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng giới tính	66
Hình 39: Code Python trình bày dữ liệu của ta có tần suất ghé thăm - “visitNo” của khách như sau	68
Hình 40: Tạo bảng chéo để tổng hợp số lượng khách hàng đối với nhóm khách.....	69
Hình 41: Code Python vẽ biểu đồ thể hiện đánh giá mức tiền theo từng nhóm khách hàng	69
Hình 42: Code Python vẽ biểu đồ thể hiện tỷ lệ thời gian của từng khách hàng ở lại Starbucks .	70
Hình 43: Code Python tạo bảng bao gồm các cột “status” (nhóm khách hàng), “visitNo” (ghé thăm) và “Counts” (đếm tần số xuất hiện).....	72
Hình 44: Code Python vẽ Biểu đồ thể hiện tần suất ghé thăm Starbucks của từng nhóm khách hàng.....	72
Hình 45: Code Python tạo một bảng chéo crosstab bao gồm các cột “age”, “spendPurchase”....	74
Hình 46: Code Python vẽ biểu đồ thể hiện mức độ chi trả của khách hàng đến Starbucks theo từng nhóm tuổi	74
Hình 47: Code Python chuyển dạng dữ liệu	76
Hình 48: Code Python kiểm định Shapiro dựa trên Ordinary Least Squares (OLS) model	77
Hình 49: Kết quả Kiểm định Shapiro dựa trên Ordinary Least Squares (OLS) model	77
Hình 50: Code Python tạo bảng ANOVA để biểu thị các nguồn sai số và mức độ tự do liên quan của các biến	78
Hình 51: Code Python hậu kiểm Tukey HSD.....	79
Hình 52: Code Python xây dựng một biến chứa list các columns cần kiểm định.	81
Hình 53: Code Python tạo 1 DataFrame chứa các cột “Feature 1”, “Feature 2” và “p-value”....	81
Hình 54: Code Python tạo biến alpha = 0.05 và biến confidence_level = (1 - alpha).	81
Hình 55: Code Python tạo vòng lặp for để tìm giá trị p-value giữa các biến phân loại.....	82
Hình 56: Các kết quả sau khi lấy p-value so sánh với alpha.	82
Hình 57: Code Python loại bỏ biến target khỏi dữ liệu	87
Hình 58: Code Python truyền bộ dữ liệu sau khi lọc vào mô hình PCA	88
Hình 59: Code Python vẽ đồ thị biểu diễn % phương sai tích lũy theo số features.....	88
Hình 60: Đồ thị biểu diễn % phương sai tích lũy theo số features	89
Hình 61: Code Python tìm số k - cách 2	90
Hình 62: Code Python phân tích chi tiết theo các ngưỡng phương sai từ 50% đến 99%	90
Hình 63: Code Python biểu diễn trực quan dữ liệu với k = 17 – KHÔNG phân lớp.....	91
Hình 64: Biểu đồ biểu diễn trực quan dữ liệu với k = 17	92
Hình 65: Code Python biểu diễn trực quan dữ liệu với k = 17 – CÓ phân lớp	92
Hình 66: Biểu diễn trực quan dữ liệu với k = 17 – CÓ phân lớp.....	94
Hình 67: Code Python chuẩn hóa bộ dữ liệu đã giảm chiều.....	95

MỤC LỤC BẢNG

Bảng 1: Bảng dữ liệu sau khi xử lý label encoding	13
Bảng 2: Bảng dữ liệu sau khi xử lý promomethod	15
Bảng 3: Bảng dữ liệu sau khi xử lý itempurchase	16
Bảng 4: Bảng dữ liệu sau khi đổi tên cột	18
Bảng 5: Bảng chứa các dữ liệu hoàn chỉnh	19
Bảng 6: Bảng gồm cột visitNo và cột spendPurcse và tính tổng theo 2 thuộc tính thành cột count	45
Bảng 7: Bảng chéo lấy tên servicerate_crosstab thể hiện sự tương quan giữa biến “age” và biến “serviceRate”	49
Bảng 8: Bảng dữ liệu sau khi gom nhóm các phương thức mua nước và thời gian khách hàng ở lại quán	58
Bảng 9: Bảng ANOVA biểu thị các nguồn sai số và mức độ tự do liên quan của các biến	79
Bảng 10: Bảng so sánh các cặp mẫu bằng phương pháp Tukey HSD	80
Bảng 11: Ma trận B_T	93

CHƯƠNG 1. TỔNG QUAN

1.1 Giới thiệu về tài

Starbucks là một chuỗi các cửa hàng cà phê lớn nhất thế giới có trụ sở tại Washington, Mỹ. Được thành lập vào năm 1971, qua nhiều năm phát triển và chinh phục khách hàng, đến nay Starbucks được đánh giá cao cả về mặt chất lượng sản phẩm và dịch vụ, và được phân loại vào nhóm cửa hàng cà phê cao cấp. Để có thể hiểu rõ hơn khách hàng cảm nhận những gì về Starbucks, và liệu trong thực tế, Starbucks có thực sự là một thương hiệu đem đến những trải nghiệm tuyệt vời xứng đáng với số tiền trung bình khoảng \$2,75- \$3 khách hàng bỏ ra cho 1 phần thức uống, nhóm quyết định phân tích các kết quả từ một cuộc khảo sát về hành vi mua sắm của 122 khách hàng đối với Starbucks tại Malaysia.

1.2 Tổng quan về bộ dữ liệu

Bộ dữ liệu có tên là “*Starbucks Customer Survey*”, được đăng tải bởi MAHIRA HAMZAH trên trang Kaggle. Bộ dữ liệu gồm 122 quan sát, với 20 câu hỏi thuộc 3 nhóm vấn đề chính là: nhân khẩu học của khách hàng, hành vi mua sắm hiện tại của khách hàng tại Starbucks, các đặc điểm và cơ sở vật chất tại Starbucks tác động đến hành vi của khách hàng. Bộ dữ liệu gồm 22 cột, đó là:

- ***Timestamp***: thời gian ghi nhận câu trả lời khảo sát từ khách hàng
- ***1. Your gender***: giới tính khách hàng (Male; Female)
- ***2. Your age***: tuổi của khách hàng (Below 20; From 20 to 29; From 30 to 39; 40 and above)
- ***3. Are you currently....***: tình trạng công việc của khách hàng (Student; Self-Employed; Employed; Housewife)
- ***4. What is your annual income?***: thu nhập trung bình của khách hàng (Less than RM25,000; RM25,000 – RM50,000; RM50,000 – RM100,000; RM100,000 – RM150,000; More than RM150,000)
- ***5. How often do you visit Starbucks?***: tần suất ghé thăm cửa hàng của khách hàng trong 1 năm (Daily; Weekly; Monthly; Never)
- ***6. How do you usually enjoy Starbucks?***: phương thức dùng sản phẩm của khách hàng (Dine In; Drive-thru; Take away; Never; Others)
- ***7. How much time do you normally spend during your visit?***: thời gian khách hàng ở lại cửa hàng (Below 30 mins; 30 mins to 1h; 1h to 2h; 2h to 3h; More than 3h)
- ***8. The nearest Starbucks's outlet to you is...?***: khoảng cách từ chỗ ở/vị trí của khách hàng đến cửa hàng Starbucks gần nhất (Within 1 km; 1 km to 3 km; More than 3 km)

- **9. Do you have Starbucks membership card?:** khách hàng có đăng ký thẻ thành viên của Starbucks không (Yes; No)
- **10. What do you most frequently purchase at Starbucks?:** khách hàng đã từng sử dụng các sản phẩm nào của Starbucks - cà phê, đồ uống lạnh, bánh mặn, nước ép, bánh mì kẹp, các loại sản phẩm khác (Yes; No)
- **11. On average, how much would you spend at Starbucks per visit?:** số tiền trung bình trong 1 lần mua của khách hàng (Zero; Less than RM20; RM 20 to RM40; More than RM40)
- **7 cột Rate:** đánh giá điểm từ 1-5 của khách hàng về các yếu tố liên quan đến cửa hàng - sản phẩm, giá, chương trình quảng bá, không gian quán, chất lượng wifi, chất lượng dịch vụ, lựa chọn Starbucks là điểm đến khi bàn công việc hoặc tụ tập với bạn bè (1: Very Bad; 5: Excellent)
- **19. How do you come to hear of promotions at Starbucks? Check all that apply.:** khách hàng đã nhận được những thông tin quảng bá sản phẩm của Starbucks qua những kênh nào - ứng dụng Starbucks, mạng xã hội, thư điện tử, các trang web giao dịch, truyền miệng, trưng bày tại cửa hàng, bảng quảng cáo, các kênh khác,... (Yes; No)
- **20. Will you continue buying at Starbucks?:** khách hàng có muốn trung thành với Starbucks không (Yes; No)

 Trong bài báo cáo này, nhóm sẽ thực hiện các công việc sau:

- Tiền xử lý bộ dữ liệu: xử lý các giá trị bị thiếu, tách cột “10. What do you most frequently purchase at Starbucks?” và cột “19. How do you come to hear of promotions at Starbucks? Check all that apply.”
- Thống kê mô tả các nhằm tìm ra những đặc điểm nổi bật của bộ dữ liệu như hình dáng phân phối, sự tương quan giữa các biến,...
- Sau đó, nhóm thực hiện trực quan hóa dữ liệu qua các dạng biểu đồ phù hợp đã được học
- Cuối cùng, nhóm tiến hành một số kiểm định như One-way ANOVA, Chi-squared, PCA nhằm tìm ra các biến có độ tương quan cao và giảm chiều dữ liệu.

CHƯƠNG 2. TIỀN XỬ LÝ DỮ LIỆU

2.1. Xử lý các cột dữ liệu bị thiếu

Nhóm sẽ kiểm tra dữ liệu bị thiếu trên các cột của bộ dữ liệu gốc:

[] len(data)-data.count()	
Timestamp	0
1. Your Gender	0
2. Your Age	0
3. Are you currently....?	0
4. What is your annual income?	0
5. How often do you visit Starbucks?	0
6. How do you usually enjoy Starbucks?	1
7. How much time do you normally spend during your visit?	0
8. The nearest Starbucks's outlet to you is...?	0
9. Do you have Starbucks membership card?	0
10. What do you most frequently purchase at Starbucks?	0
11. On average, how much would you spend at Starbucks per visit?	0
12. How would you rate the quality of Starbucks compared to other brands (Coffee Bean, Old Town White Coffee..) to be:	0
13. How would you rate the price range at Starbucks?	0
14. How important are sales and promotions in your purchase decision?	0
15. How would you rate the ambiance at Starbucks? (lighting, music, etc...)	0
16. You rate the WiFi quality at Starbucks as..	0
17. How would you rate the service at Starbucks? (Promptness, friendliness, etc..)	0
18. How likely you will choose Starbucks for doing business meetings or hangout with friends?	0
19. How do you come to hear of promotions at Starbucks? Check all that apply.	1
20. Will you continue buying at Starbucks?	0
dtype: int64	

Hình 1: Cột của bộ dữ liệu gốc

Từ kết quả trên, nhóm nhìn thấy chỉ có cột “How do you come to hear of promotions at Starbucks? Check all that apply.” và cột “How much time do you normally spend during your visit?” chỉ có 1 dòng giá trị bị thiếu dữ liệu. Để đảm bảo cho các cột không bị thiếu số lượng khi kiểm định hoặc vẽ biểu đồ trực quan nên nhóm sẽ thay thế các giá trị bị thiếu thành giá trị “none”.

▶ data = data.fillna('none')

Hình 2: Code Python thay thế các giá trị bị thiếu thành giá trị “none”

Và xóa các cột không cần thiết trong bảng dữ liệu

▶ data = data.drop(columns = ['Timestamp']) # loại bỏ cột timestamp

Hình 3: Code Python xóa các cột không cần thiết trong bảng dữ liệu

✚ Label encoding:

Vì các dữ liệu trong các cột đều được chia theo các khoảng giá trị, để thuận tiện cho việc kiểm định và đọc dữ liệu thông qua biểu đồ một cách dễ dàng và dễ hiểu nên nhóm đã quyết định chuẩn hóa toàn bộ những cột dữ liệu.

Đọc ra những cột cần chuẩn hóa:

```

1 cols = list(data.columns)

1 cols_names = [i for i in cols if i not in [cols[9],cols[18]]]
2 cols_names = [i for i in cols_names if i not in cols_names[10:17]]

1 cols_names

```

Hình 4: Code Python đọc ra những cột cần chuẩn hóa

➤ Kết cho ra nhóm thu được là:

```

1 cols_names

['1. Your Gender',
 '2. Your Age',
 '3. Are you currently....?',
 '4. What is your annual income?',
 '5. How often do you visit Starbucks?',
 '6. How do you usually enjoy Starbucks?',
 '7. How much time do you normally spend during your visit?',
 "8. The nearest Starbucks's outlet to you is...?",
 '9. Do you have Starbucks membership card?',
 '11. On average, how much would you spend at Starbucks per visit?',
 '20. Will you continue buying at Starbucks?']

```

Hình 5: Kết quả những cột cần chuẩn hóa

Để thực hiện mã hóa dữ liệu, nhóm sử dụng hàm `OrdinalEncoder()` từ thư viện `sklearn.preprocessing`. Sau đó, nhóm tiến hành chuẩn hóa các cột đã đọc và định dạng lại kiểu dữ liệu là số nguyên (`int`):

```

▶ from sklearn.preprocessing import OrdinalEncoder

ord_enc = OrdinalEncoder()
for i in cols_names:
    data[i] = ord_enc.fit_transform(data[[i]]).astype('int')

```

Hình 6: Code Python chuẩn hóa các cột đã đọc và định dạng lại kiểu dữ liệu

➤ Nhóm thu được toàn bộ dữ liệu trong bảng như sau:

1. Your Gender	2. Your Age	3. Are you currently...?	4. What is your annual income?	5. How often do you visit Starbucks?	6. How do you usually enjoy Starbucks?	7. How much time do you normally spend during your visit?	8. The nearest Starbucks' outlet to you is...?	9. Do you have Starbucks membership card?	10. What do you most frequently purchase at Starbucks?	11. On average, how much would you spend at Starbucks per visit?	12. How would you rate the quality of Starbucks compared to other brands (Coffee Bean, Old Town White Coffee...) to be:	13. How would you rate the price range at Starbucks?	14. How important are sales and promotions in your purchase decision?	15. How would you rate the ambience at Starbucks? (Promptness, lighting, music, etc...)	16. You rate the WiFi quality at Starbucks as..	17. How would you rate the service at Starbucks? (Promptness, friendliness, etc.,)	18. How likely you will choose Starbucks for doing business meetings or hangout with friends?	19. How do you come to hear of promotions at Starbucks? Check all that apply.	20. Will you continue buying at Starbucks?
0	0	2	3	0	3	0	3	2	Coffee	1	4	3	5	5	4	4	3	Starbucks Website/Apps/Social Media/Emails/Deals	1
1	0	2	3	0	3	6	0	0	Cold drinks/Pastries	1	4	3	4	4	4	5	2	Social Media/In Store displays	1
2	1	2	0	0	1	0	3	1	Coffee	1	4	3	4	4	4	4	3	In Store displays/Billboards	1
3	0	2	3	0	3	6	0	1	Coffee	1	2	1	4	3	3	3	3	Through friends and word of mouth	0
4	1	2	3	0	1	6	3	0	Coffee/Sandwiches	0	3	3	4	2	2	3	3	Starbucks Website/Apps/Social Media	1
...	
117	1	0	2	3	1	0	1	0	Coffee	0	3	3	5	3	2	4	4	Website/Apps/Social Media	1
118	1	2	0	0	1	0	1	0	Coffee;Cold drinks;Juices;Pastries;Sandwiches	2	5	5	5	5	5	5	5	Starbucks Website/Apps/Social Media/Emails/Deals	1
119	1	2	3	0	3	0	3	0	Coffee;Cold drinks	1	3	2	4	3	3	3	4	Social Media/Through friends and word of mouth...	0
120	0	2	0	0	3	6	0	2	Coffee	1	4	4	4	4	4	4	4	Social Media/Through friends and word of mouth...	1
121	1	2	0	4	3	0	3	0	Coffee	1	1	1	5	4	3	3	2	In Store displays	0

122 rows x 20 columns

Bảng 1: Bảng dữ liệu sau khi xử lý label encoding

Split promomethod:

Trong bộ dữ liệu có chứa cột “How do you come to hear of promotions at Starbucks? Check all that apply.” là các ghi nhận của khách hàng về thông tin biết được các chương trình khuyến mãi tại Starbucks. Và các dữ liệu trên cột này là chứa tất cả rất nhiều thông tin mà khách hàng biết được. Nên nhóm tiến hành kiểm tra có tất cả bao nhiêu thông tin được ghi nhận từ khách hàng của cột này.

Các lựa chọn đang ngăn cách nhau bởi dấu “;” và dính liền nhau, không có sự ngăn cách nên sẽ gặp khó khăn trong lúc đếm các lựa chọn bằng các hàm xử lý chuỗi. Vì thế, nhóm sẽ bắt đầu xử lý dữ liệu ở cột này bằng cách thay tự “;” bằng cụm ký tự “;”; để có thể tách thành các chuỗi con để dễ dàng đếm các phương thức mà khách hàng nhận thông tin quảng bá, khuyến mãi:

```
▶ data['19. How do you come to hear of promotions at Starbucks? Check all that apply.']= \
data['19. How do you come to hear of promotions at Starbucks? Check all that apply.'].str.replace(';','; ')
```

Hình 7: Code Python xử lý dữ liệu

Sau đó nhóm sẽ dùng hàm str.len() đếm các cách thức được ghi nhận từ khách hàng

```
▶ data['count']= data['19. How do you come to hear of promotions at Starbucks? Check all that apply.'].str.len()
data
```

Hình 8: Code Python đếm các cách thức được ghi nhận từ khách hàng

Sau đó, nhóm sẽ sắp xếp lại theo thứ tự tăng dần để xem số phương thức mà khách hàng nhận thông báo nhiều nhất là bao nhiêu, lấy câu trả lời của khách hàng đó để đặt tên cho các cột phương thức khi nhóm tiến hành tách cột.

```
[24] data.sort_values('count', inplace = True)
```

Hình 9: Code Python sắp xếp lại theo thứ tự tăng dần

Kết quả cho thấy phần tử thứ 118 chọn nhiều phương thức nhận thông tin quảng cáo nhất, nên nhóm sẽ tách tên các phương thức và gom chúng vào trong 1 danh sách (list) tên promethod_list qua hàm.split("; ")

```
[19] promethod_list = data['19. How do you come to hear of promotions at Starbucks? Check all that apply.'][118].split('; ')
```

Hình 10: Code Python tách tên các phương thức và gom chúng vào trong 1 danh sách

➤ Nhóm thu được kết quả như sau:

```
promethod_list
['Starbucks Website/Apps',
 'Social Media',
 'Emails',
 'Deal sites (fave, iprice, etc...)',
 'Through friends and word of mouth',
 'In Store displays',
 'Billboards']
```

Hình 11: Danh sách các phương thức

Vậy nhóm thu được 6 cách phổ biến mà khách hàng nhận được thông tin chương trình khuyến mãi của Starbuck. Nhóm xây dựng một hàm tên check() với 2 tham số đầu vào x, i, lần lượt là tên bảng dữ liệu và tên phương thức cần xét. Hàm dùng để kiểm tra liệu tên phương thức của cột đang xét có trùng với 1 chuỗi con nào đó trong chuỗi câu trả lời lưu ở cột “19. How do you come to hear of promotions at Starbucks? Check all that apply.”; nếu trùng hàm sẽ trả về giá trị 1, ngược lại sẽ trả về giá trị 0. Định nghĩa hàm xong, nhóm sẽ thực hiện vòng lặp for với tất cả tên phương thức đã lưu trong danh sách promethod_list trước đó: nhóm sẽ thêm 1 cột dữ liệu mới vào bảng dữ liệu data hiện tại; cột này sẽ lưu lại kết quả hàm kiểm tra vừa được định nghĩa ở trên với tên phương thức i đang xét qua, sử dụng lệnh lambda và gọi hàm kiểm tra check() với 2 tham số đầu vào và x(bảng dữ liệu data) và i (tên phương thức đang xét).

```
def check(x,i):
    return 1 if x['19. How do you come to hear of promotions at Starbucks? Check all that apply.'].find(i)!=-1 else 0

for i in promethod_list:
    data[i] = data.apply(lambda x: check(x,i), axis=1)
```

Hình 12: Code Python thêm cột promethod_list vào bảng dữ liệu hiện tại

➤ Nhóm sẽ thu được kết quả bảng dữ liệu mới như sau:

	Starbucks Website/Apps	Social Media	Emails	Deal sites (fave, price, etc...)	Through friends and word of mouth	In Store displays	Billboards
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
...
0	1	0	1	1	0	0	0
1	1	0	0	1	1	1	0
1	1	0	1	1	0	0	0

Bảng 2: Bảng dữ liệu sau khi xử lý promomethod

Khi đã tách cột thành công, nhóm sẽ xóa cột “19. How do you come to hear of promotions at Starbucks? Check all that apply.” và cột “count”.

```
▶ data = data.drop(columns = ['19. How do you come to hear of promotions at Starbucks? Check all that apply.', 'count'])
```

Hình 13: Code Python xóa cột “19. How do you come to hear of promotions at Starbucks? Check all that apply.” và cột “count”

❖ Split itempurchase:

Tương tự như trên, trong bộ dữ liệu có cột “10. What do you most frequently purchase at Starbucks?” là chứa các thông tin về loại sản phẩm khách hàng đến Starbuck thường mua nhất. Nhóm cũng tiến hành các bước xử lý dữ liệu và tách cột như với cột “19. How do you come to hear of promotions at Starbucks? Check all that apply.”

```
[38] data[ '10. What do you most frequently purchase at Starbucks?' ] = data[ '10. What do you most frequently purchase at Starbucks?' ].str.replace(';', ', ')
[39] data['count'] = data[ '10. What do you most frequently purchase at Starbucks?' ].str.len()
[40] values = {"count":0}
      data = data.fillna(value=values)
[41] data.sort_values('count', ascending = False, inplace = True)
```

Hình 14: Code Python xử lý dữ liệu và tách cột “10. What do you most frequently purchase at Starbucks?”

Và thu được phản hồi chủ yếu từ ghi nhận của khách hàng có kết quả như sau:

```
[43] promethod_list = data[ '10. What do you most frequently purchase at Starbucks?'][118].split('; ')
[44] promethod_list
['Coffee', 'Cold drinks', 'Juices', 'Pastries', 'Sandwiches']
```

Hình 15: Code Python xem kết quả xử lý

```
[45] def check(x,i):
    return 1 if x[ '10. What do you most frequently purchase at Starbucks?'].find(i)!=-1 else 0
[46] for i in promethod_list:
    data[i] = data.apply(lambda x: check(x,i), axis=1)
```

➤ Sau khi xử lý, nhóm sẽ thu được một bộ dữ liệu mới như sau:

s	Coffee	Cold drinks	Juices	Pastries	Sandwiches
1	1	1	1	1	1
0	1	1	0	1	1
0	1	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	0

Bảng 3: Bảng dữ liệu sau khi xử lý itempurchase

Cuối cùng nhóm cũng xóa cột “10. What do you most frequently purchase at Starbucks?” và cột “count”.

```
▶ data = data.drop(columns = [ '10. What do you most frequently purchase at Starbucks?', 'count'])
```

Hình 16: Code Python xóa cột “10. What do you most frequently purchase at Starbucks?” và cột “count”.

Rename columns

Sau khi các dữ liệu đã được mã hóa các thông tin thì nhóm sẽ đổi tên cột cho phù hợp cho việc sử dụng lại các tên cột trong quá trình xử lý sau này. Đầu tiên, nhóm sẽ sắp xếp lại cột theo thứ tự câu hỏi bằng hàm.reindex():

```
▶ new_cols = ['1. Your Gender',
    '2. Your Age',
    '3. Are you currently....?',
    '4. What is your annual income?',
    '5. How often do you visit Starbucks?',
    '6. How do you usually enjoy Starbucks?',
    '7. How much time do you normally spend during your visit?',
    "8. The nearest Starbucks's outlet to you is...?",
    '9. Do you have Starbucks membership card?',
    'Coffee',
    'Cold drinks',
    'Juices',
    'Pastries',
    'Sandwiches',
    '11. On average, how much would you spend at Starbucks per visit?',
    '12. How would you rate the quality of Starbucks compared to other brands (Coffee Bean, Old Town White Coffee..) to be:',
    '13. How would you rate the price range at Starbucks?',
    '14. How important are sales and promotions in your purchase decision?',
    '15. How would you rate the ambiance at Starbucks? (lighting, music, etc...)',
    '16. You rate the WiFi quality at Starbucks as..',
    '17. How would you rate the service at Starbucks? (Promptness, friendliness, etc..)',
    '18. How likely you will choose Starbucks for doing business meetings or hangout with friends?',
    'Starbucks Website/Apps',
    'Social Media',
    'Emails',
    'Deal sites (fave, iprice, etc...)',
    'Through friends and word of mouth',
    'In Store displays',
    'Billboards',
    '20. Will you continue buying at Starbucks?'
]
```

Hình 17: Code Python sắp xếp lại cột theo thứ tự câu hỏi

```
[▶] data = data.reindex(columns=new_cols)
```

Vì các tên cột hiện tại quá dài, gây bất tiện trong quá trình làm bài nên nhóm sẽ đổi tên các cột ngắn gọn hơn qua hàm.rename(). Tham số trong hàm.rename() là một từ điển dict với khóa (key) là tên cột hiện tại và giá trị (value) là tên cột mới. Danh sách tên các cột mới nhóm lưu trong 1 danh sách (list) tên new_cols_name. Nhóm sử dụng hàm dict(zip()) để lập từ điển với 2 danh sách khóa và giá trị.

```

❶ new_cols_name = ['gender',
    'age',
    'status',
    'income',
    'visitNo',
    'method',
    'timeSpend',
    'location',
    'membershipCard',
    'itemPurchaseCoffee',
    'itemPurchaseCold',
    'itemPurchasePastries',
    'itemPurchaseJuices',
    'itemPurchaseSandwiches',
    'spendPurchase',
    'productRate',
    'priceRate',
    'promoRate',
    'ambianceRate',
    'wifiRate',
    'serviceRate',
    'chooseRate',
    'promoMethodApp',
    'promoMethodSoc',
    'promoMethodEmail',
    'promoMethodDeal',
    'promoMethodFriend',
    'promoMethodDisplay',
    'promoMethodBillboard',
    'loyal']

```

Hình 18: Code Python đổi tên các cột

```
[52] cols_rename = dict(zip(new_cols, new_cols_name))
```

```
[53] data = data.rename(columns = cols_rename)
```

➤ Nhóm sẽ thu được kết quả như sau:

	gender	age	status	income	visitNo	method	timeSpend	location	membershipCard	itemPurchaseCoffee	...	serviceRate	chooseRate	promoMethodApp	promoMethodSoc	promoMethodEmail
118	1	2	0	0	1	0	1	0	1	1	...	4	4	1	1	1
66	0	2	2	3	1	0	0	2	1	1	...	3	3	1	1	1
24	1	0	2	2	3	0	1	1	0	1	...	2	4	0	0	0
23	0	2	3	0	1	1	1	0	1	0	...	3	2	0	1	0
52	0	0	1	0	1	6	0	1	1	0	...	4	4	1	1	0
...
58	1	1	3	0	1	0	0	0	0	1	...	2	2	0	1	0
22	1	2	0	3	3	0	4	2	0	1	...	3	2	0	0	0
67	0	2	2	0	2	5	0	1	0	0	...	2	2	0	0	0
108	1	2	3	0	2	7	0	2	0	0	...	2	3	0	0	0

Bảng 4: Bảng dữ liệu sau khi đổi tên cột

Cuối cùng nhóm sẽ sắp xếp lại các index cho phù hợp

```
[55] data = data.sort_index()
```

Hình 19: Code Python sắp xếp lại các index cho phù hợp

➤ Sau đó nhóm sẽ thu được một bảng chứa các dữ liệu hoàn chỉnh như sau:

	gender	age	status	income	visitNo	method	timeSpend	location	membershipCard	itemPurchaseCoffee	...	serviceRate	chooseRate	promoMethodApp	promoMethodSoc	promoMethodEmail
0	0	2	3	0	3	0	3	2	1	1	...	3	2	1	1	1
1	0	2	3	0	3	6	0	0	1	0	...	4	1	0	1	C
2	1	2	0	0	1	0	3	1	1	1	...	3	2	0	0	C
3	0	2	3	0	3	6	0	1	0	0	1	...	2	2	0	0
4	1	2	3	0	1	6	3	0	0	0	1	...	2	2	1	C
...
117	1	0	2	3	1	0	1	0	1	1	...	3	3	1	1	C
118	1	2	0	0	1	0	1	0	1	1	...	4	4	1	1	1
119	1	2	3	0	3	0	3	0	0	0	1	...	2	3	0	C
120	0	2	0	0	3	6	0	2	0	0	1	...	3	3	0	1
121	1	2	0	4	3	0	3	0	0	0	1	...	2	1	0	C

Bảng 5: Bảng chứa các dữ liệu hoàn chỉnh

❖ Ý nghĩa các giá trị mã hóa ở từng cột như sau:

- **gender:** giới tính khách hàng (0 - female; 1 - male)
- **age:** tuổi của khách hàng (0: 40 and above; 1: Below 20; 2: From 20 to 29; 3: From 30 to 39)
- **status:** tình trạng công việc của khách hàng (0: Employed; 1: Housewife; 2: Self-employed; 3: Student)
- **income:** thu nhập trung bình của khách hàng (0: Less than RM25,000; 1: More than RM150,000; 2: RM100,000 – RM150,000; 3: RM50,000 – RM100,000; 4: RM25,000 – RM50,000)
- **visitNo:** tần suất ghé thăm cửa hàng của khách hàng trong 1 năm (0: Daily; 1: Monthly; 2: Never; 3: Rarely; 4: Weekly)
- **method:** phương thức dùng sản phẩm của khách hàng (0: Dine In; 1: Drive-thru; 2: I dont like coffee; 3: never; 4: never ; 5: none; 6: Take away; 7: Never; 8: Never Buy)
- **timeSpend:** thời gian khách hàng ở lại cửa hàng (0: Below 30 mins; 1: 1h to 2h; 2: 2h to 3h; 3: 30 mins to 1h; 4: More than 3h)
- **location:** khoảng cách từ chỗ ở/vị trí của khách hàng đến cửa hàng Starbucks gần nhất (0: 1km to 3km; 1: More than 3km; 2: Within 1km)
- **membershipCard:** khách hàng có đăng ký thẻ thành viên của Starbucks không (0: no; 1: yes)
- **6 cột itemPurchase:** khách hàng đã từng sử dụng các sản phẩm nào của Starbucks - cà phê, đồ uống lạnh, bánh mặn, nước ép, bánh mì kẹp, các loại sản phẩm khác (0: no; 1: yes)
- **spendPurchase:** số tiền trung bình trong 1 lần mua của khách hàng (1: “Less than RM20”; 0: “Around RM20 - RM40”; 2: “More than RM40”; 3: “Zero”)

- **7 cột Rate:** đánh giá điểm từ 1-5 của khách hàng về các yếu tố liên quan đến cửa hàng - sản phẩm, giá, chương trình quảng bá, không gian quán, chất lượng wifi, chất lượng dịch vụ, lựa chọn Starbucks là điểm đến khi bàn công việc hoặc tụ tập với bạn bè (1: Very Bad; 5: Excellent)
- **8 cột promomethod:** khách hàng đã nhận được những thông tin quảng bá sản phẩm của Starbucks qua những kênh nào - ứng dụng Starbucks, mạng xã hội, thư điện tử, các trang web giao dịch, truyền miệng, trưng bày tại cửa hàng, bảng quảng cáo, các kênh khác,... (0: no; 1: yes)
- **loyal:** khách hàng có muôn trung thành với Starbucks không (0: no; 1: yes)

CHƯƠNG 3. THỐNG KÊ MÔ TẢ

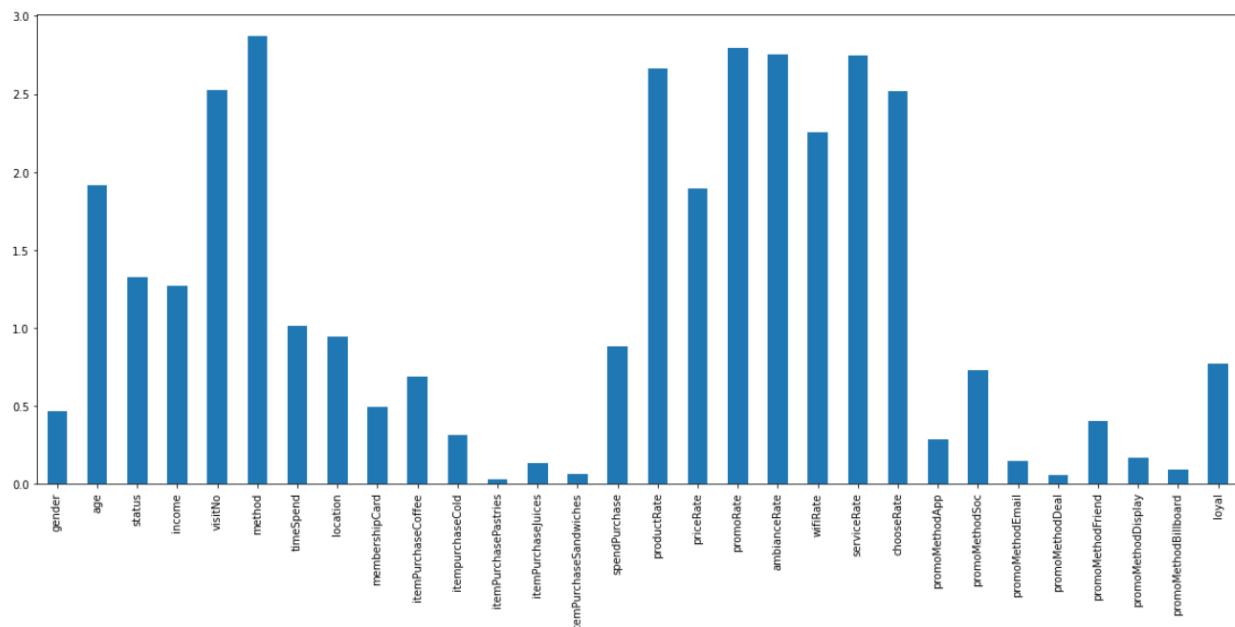
3.1. Kiểm tra các đại lượng trung tâm

```
[53] data[list(data.columns)].mean().plot(kind='bar')
plt.gcf().set_size_inches(20,8)
```

Hình 20: Code Python kiểm tra giá trị trung bình

Ta sử dụng hàm mean() để kiểm tra giá trị trung bình của các biến trong bộ dữ liệu rơi vào con số nào. Sau đó sử dụng hàm plot của thư viện matplotlib trực quan kết quả để có thể có cái nhìn tổng quan và dễ dàng hơn đối với các giá trị trung bình. Ở đây, nhóm sử dụng loại biểu đồ cột (kind = ‘bar’) để vẽ biểu đồ trực quan

➤ Kết quả thu được như sau:



Biểu đồ 1: Biểu đồ thanh thể hiện giá trị trung bình các biến

Ở đây, các biến đều được mã hóa với các định nghĩa tương ứng với các giá trị số khác nhau.

Chẳng hạn như đối với biến Gender sẽ được mã hóa thành 2 giá trị là 0 (Nam) và 1 (Nữ), theo quan sát biểu đồ, giá trị trung bình của biến Gender nằm ở mức gần 0.5. Điều này có thể được hiểu rằng số lượng Nam ghé tới cửa hàng thường xuyên và nhiều hơn Nữ một chút.

Tương tự với biến Age, giá trị trung bình nằm gần mức 2, có thể hiểu rằng các giá trị 2 được mã hóa tương ứng với độ tuổi từ 20 đến 29 sẽ có tỉ lệ ghé quán nhiều hơn so với các độ tuổi khác. Điều này khá dễ hiểu, vì Starbucks là một thương hiệu khá có tiếng và có giá thành khá cao, vì vậy sẽ thu hút những người đang ở độ tuổi có nhu cầu cao về sự thể hiện bản thân và những người có thu nhập tương đối ổn định.

Đối với các biến được mã hóa 0 (No) và 1 (Yes) thì giá trị trung bình sẽ phần nhiều là dưới 0.5. Riêng chỉ có biến *membershipCard*, *itemPurchaseCoffee*, *spendPurchase*, *promoMethodSoc* và *loyal* sẽ có nhiều Yes (hài lòng) hơn.

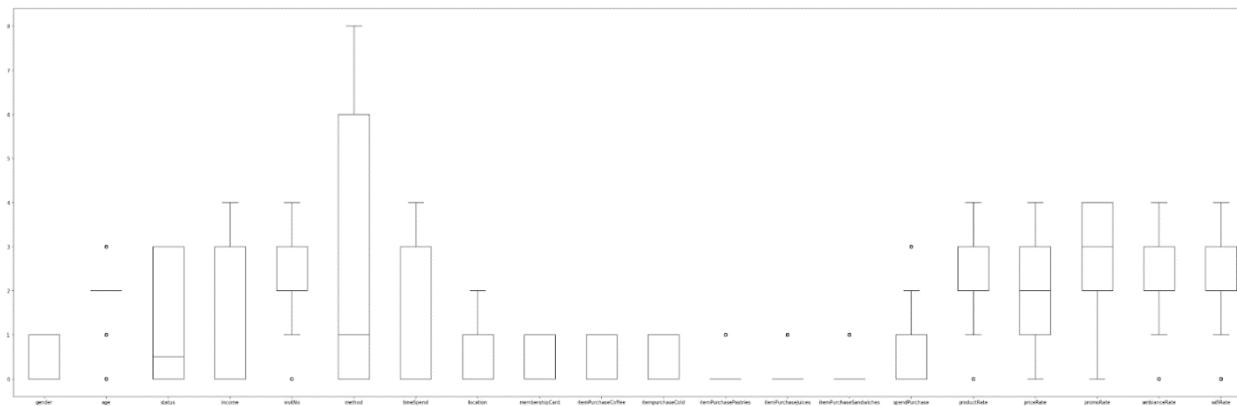
➊ Độ phân tán

Nhóm tiếp tục sử dụng hàm plot của thư viện matplotlib để vẽ biểu đồ boxplot (king= ‘box’) để kiểm tra độ phân tán của bộ dữ liệu.

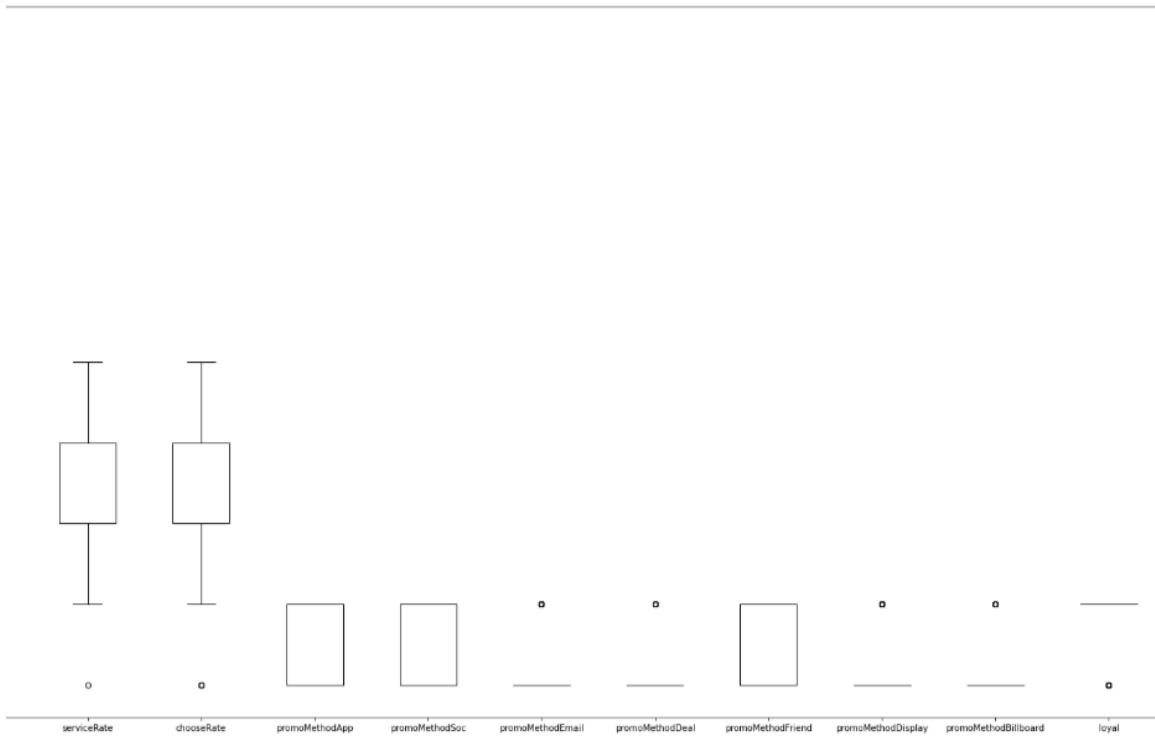
```
[61] data.plot(kind='box', color='black')
     plt.gcf().set_size_inches(70, 15)
```

Hình 21: Code Python kiểm tra độ phân tán của bộ dữ liệu

➤ Kết quả thu được như sau:



Biểu đồ 2: Biểu đồ boxplot kiểm tra độ phân tán của bộ dữ liệu



Theo như quan sát biểu đồ, ở tại các biến như sau sẽ xuất hiện Outliers (giá trị bất thường):

- **Age:** hầu hết sẽ tập trung vào 2, nhưng vẫn tồn tại 3 outliers tại 0,1 và 3. Điều này cho biết rằng trong dữ liệu 122 người ghé cửa hàng hầu hết sẽ có độ tuổi từ 20 tới 29, trong đó có 1 người dưới 20, 1 người từ 40 trở lên và 1 người sẽ có độ tuổi từ 30 tới 39.
- **itemPurchasePastries, itemPurchaseJuices, itemPurchaseSandwiches** sẽ có 1 outlier rơi vào giá trị 1 ở mỗi biến. Có khá nhiều người không thích mặt hàng này.
- **Các biến Rate** đánh giá từ productRate đến chooseRate đa số là đều có nhiều đánh giá tốt, vẫn tồn tại outlier nhưng không đáng kể.
- **Các biến promo** mã khuyến mãi, đa số là không hài lòng với các khuyến mãi tại cửa hàng. Có lẽ đây là một cửa hàng khá ít mã giảm giá hay khuyến mãi hấp dẫn.
- **Biến loyal** tập trung vào giá trị 1, hầu hết mọi người đều là khách hàng trung thành của cửa hàng. Tuy nhiên vẫn tồn tại 1 outlier.

3.2. Kiểm tra hình dáng phân phối của bộ dữ liệu

Ta sử dụng biểu đồ Histogram với các feature trong data.columns.

```

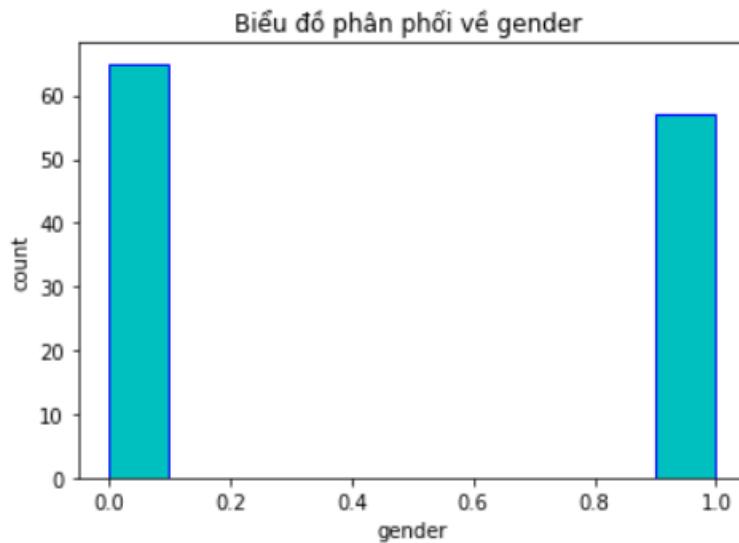
for i in data.columns:
    np.random.seed(1)
    plt.xlabel(i)
    plt.ylabel('count')
    plt.title(f'Biểu đồ phân phối về {i} ')
    plt.hist(data[i], color='c', edgecolor='b')
    plt.show()

```

Hình 22: Code Python kiểm tra hình dáng phân phối của bộ dữ liệu

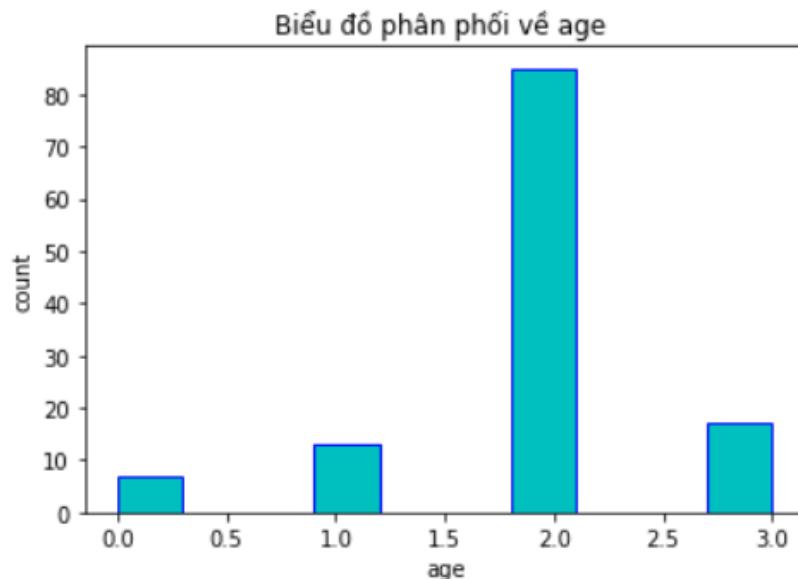
➤ Ta được các biểu đồ sau:

✚ Biểu đồ phân phối của feature gender



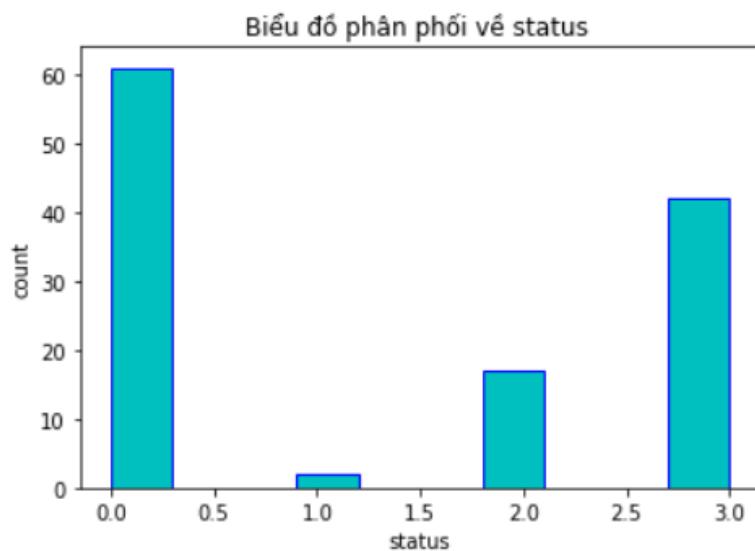
Biểu đồ 3: Biểu đồ phân phối của feature gender

✚ Biểu đồ phân phối của feature age



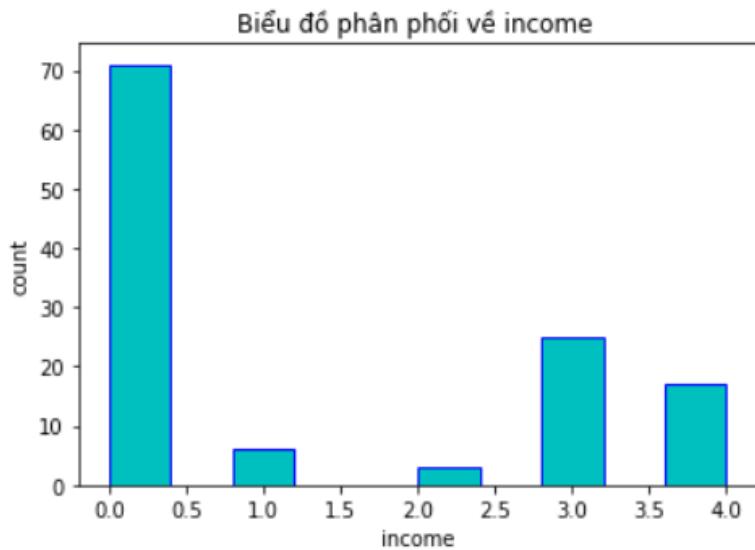
Biểu đồ 4: Biểu đồ phân phối của feature age

✚ Biểu đồ phân phối của feature status



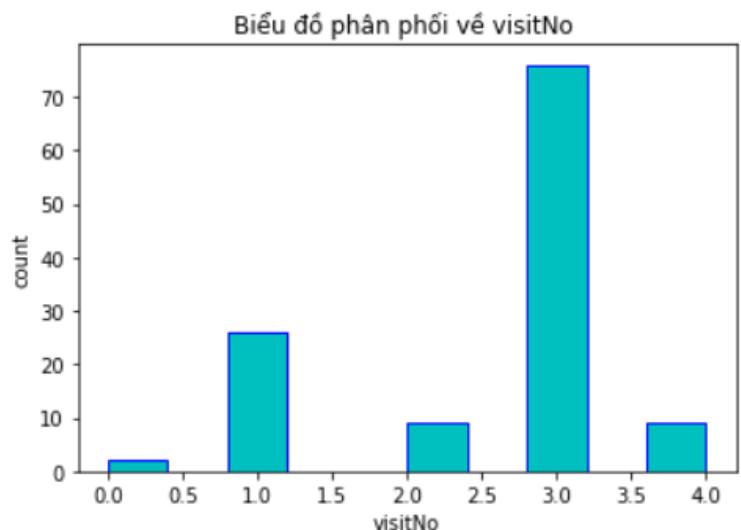
Biểu đồ 5: Biểu đồ phân phối của feature status

✚ **Biểu đồ phân phối của feature income**



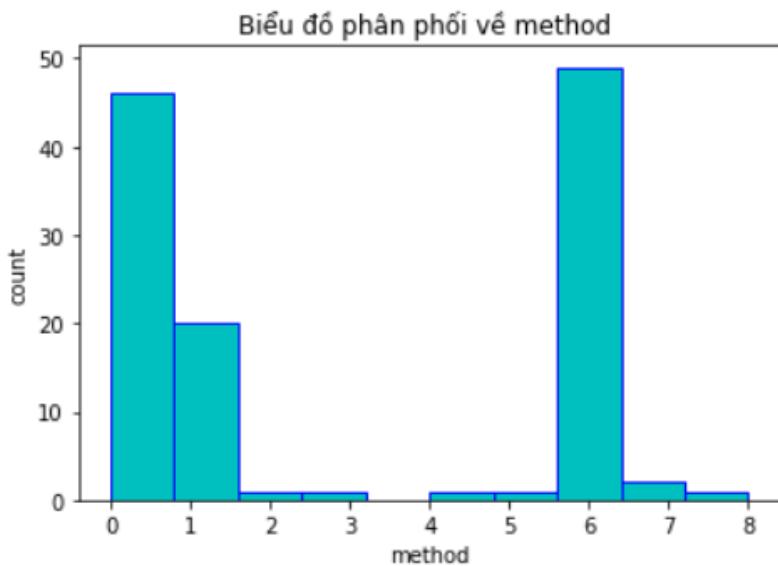
Biểu đồ 6: Biểu đồ phân phối của feature income

✚ **Biểu đồ phân phối của feature visitNo**



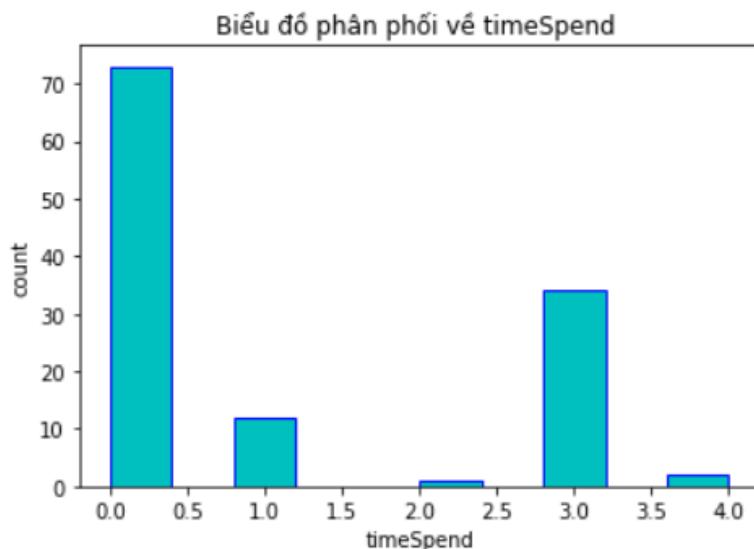
Biểu đồ 7: Biểu đồ phân phối của feature visitNo

✚ **Biểu đồ phân phối của feature Method**



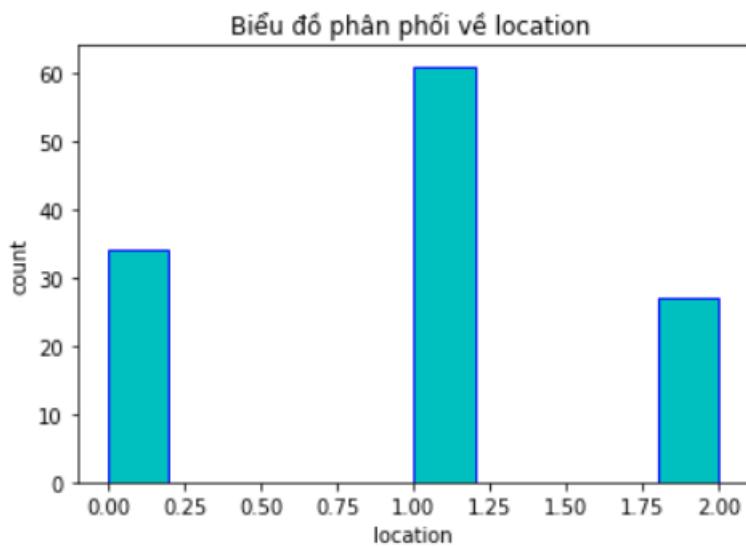
Biểu đồ 8: Biểu đồ phân phối của feature Method

✚ **Biểu đồ phân phối của feature timeSpend**



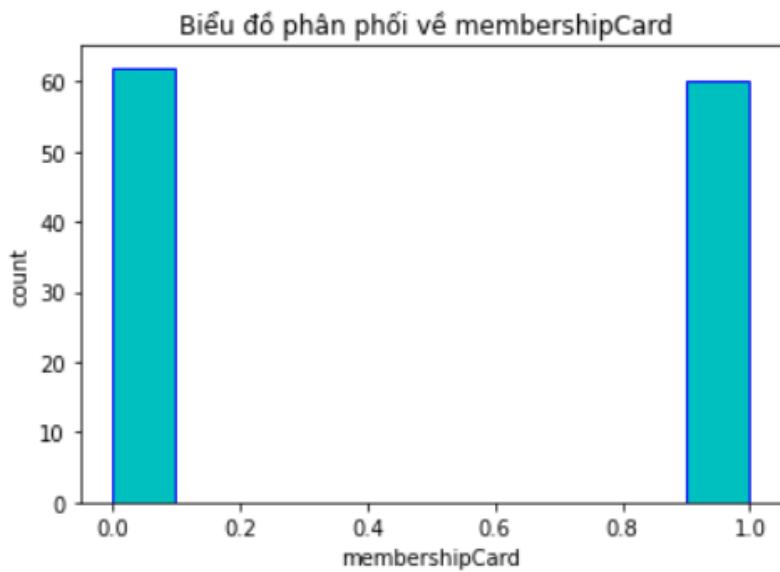
Biểu đồ 9: Biểu đồ phân phối của feature timeSpend

➊ **Biểu đồ phân phối của feature location**



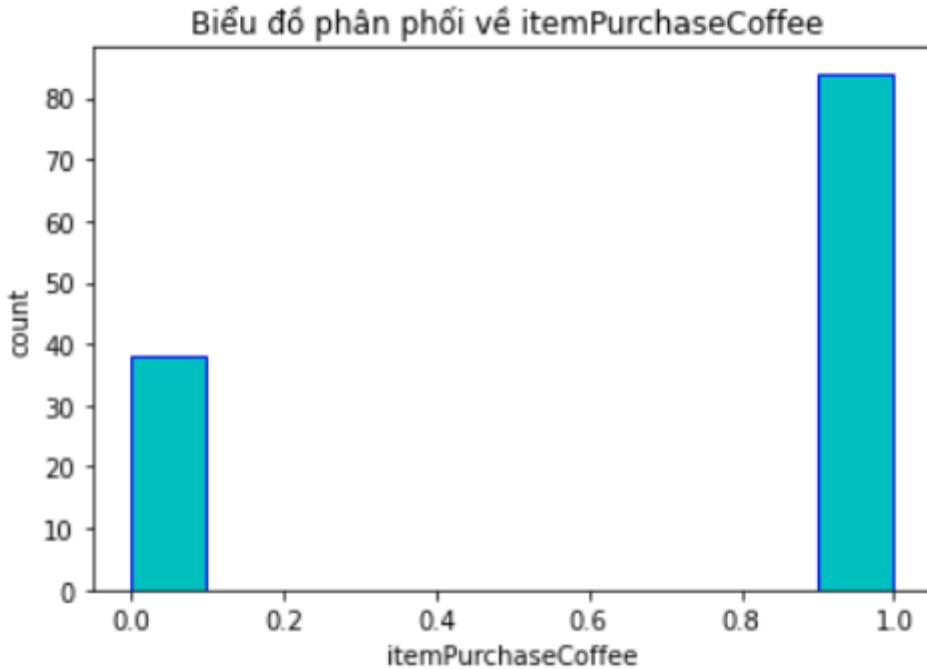
Biểu đồ 10: Biểu đồ phân phối của feature location

➋ **Biểu đồ phân phối của feature membershipCard**



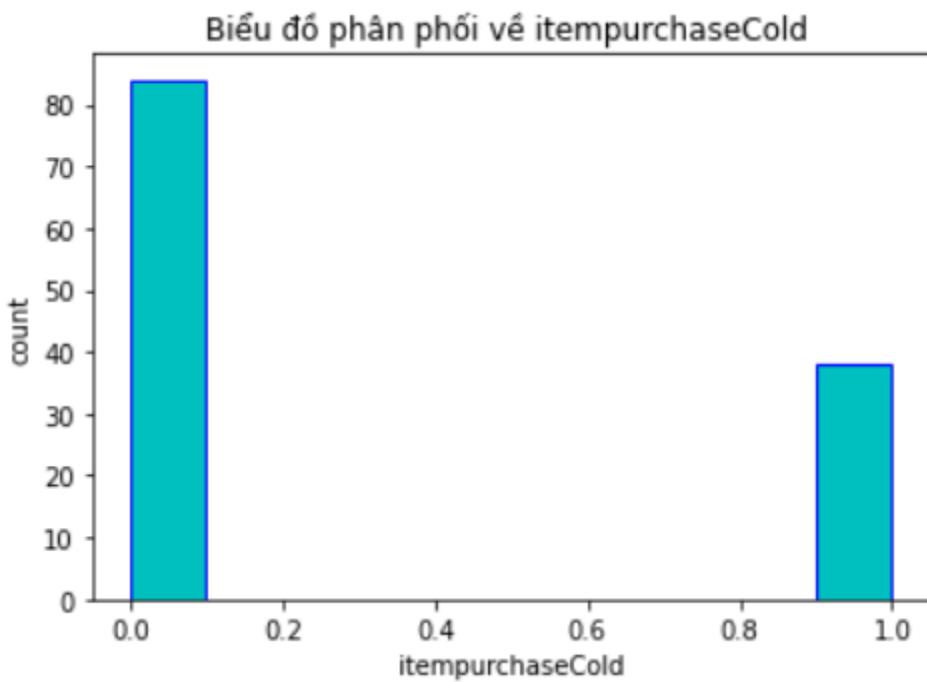
Biểu đồ 11: Biểu đồ phân phối của feature membershipCard

✚ **Biểu đồ phân phối của feature itemPurchaseCoffee**



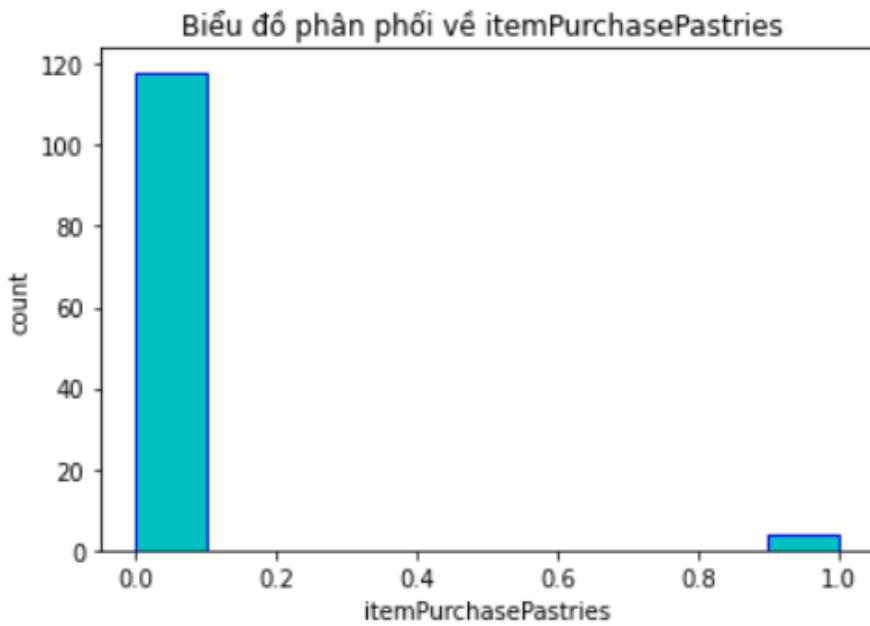
Biểu đồ 12: Biểu đồ phân phối của feature itemPurchaseCoffee

✚ **Biểu đồ phân phối của feature itemPurchaseCold**



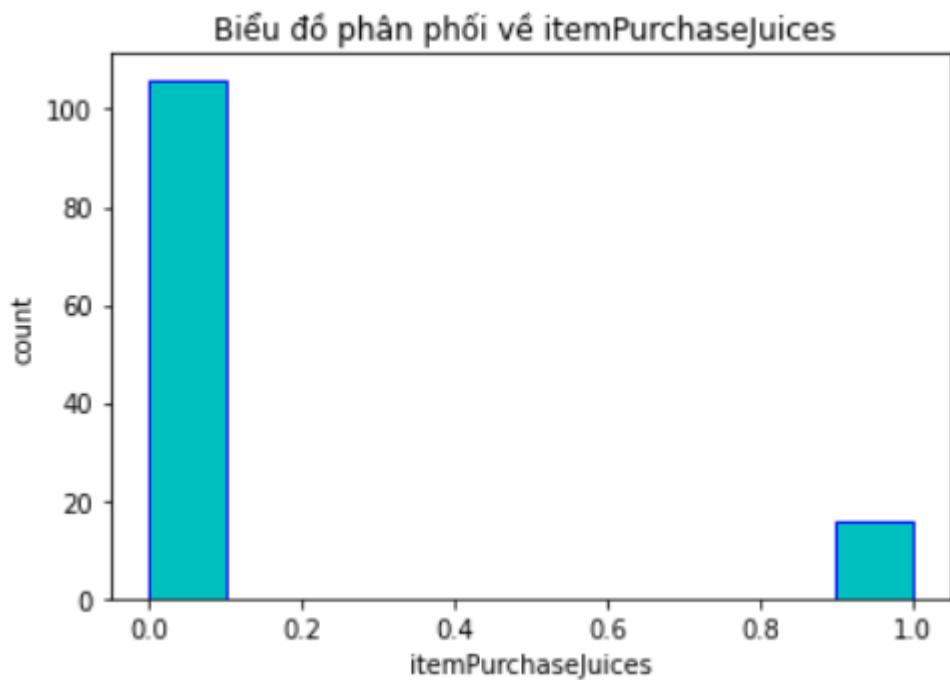
Biểu đồ 13: Biểu đồ phân phối của feature itemPurchaseCold

✚ **Biểu đồ phân phối của feature itemPurchasePastries**



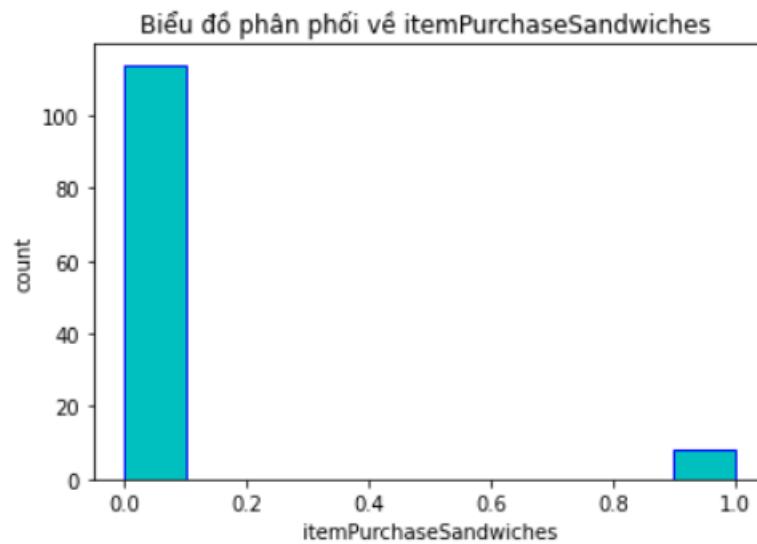
Biểu đồ 14: Biểu đồ phân phối của feature itemPurchasePastries

✚ **Biểu đồ phân phối của feature itemPurchaseJuices**



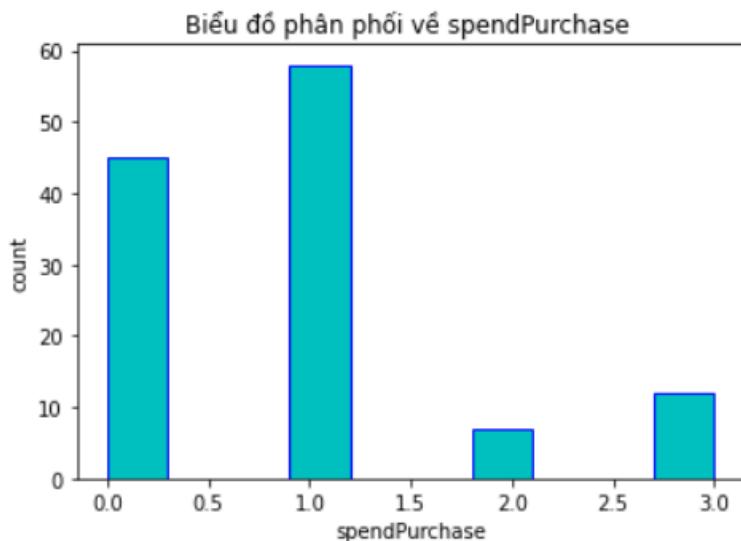
Biểu đồ 15: Biểu đồ phân phối của feature itemPurchaseJuices

✚ **Biểu đồ phân phối của feature itemPurchaseSandwiches**



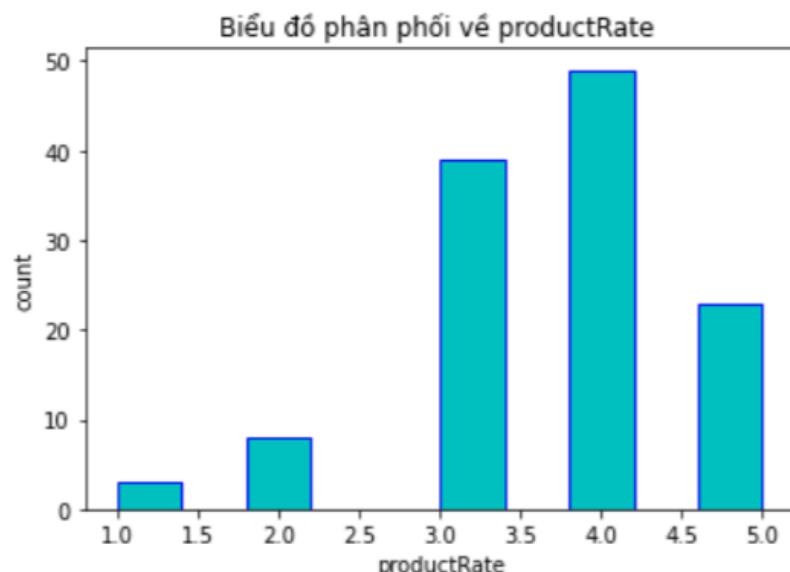
Biểu đồ 16: Biểu đồ phân phối của feature itemPurchaseSandwiches

✚ **Biểu đồ phân phối của feature spendPurchase**



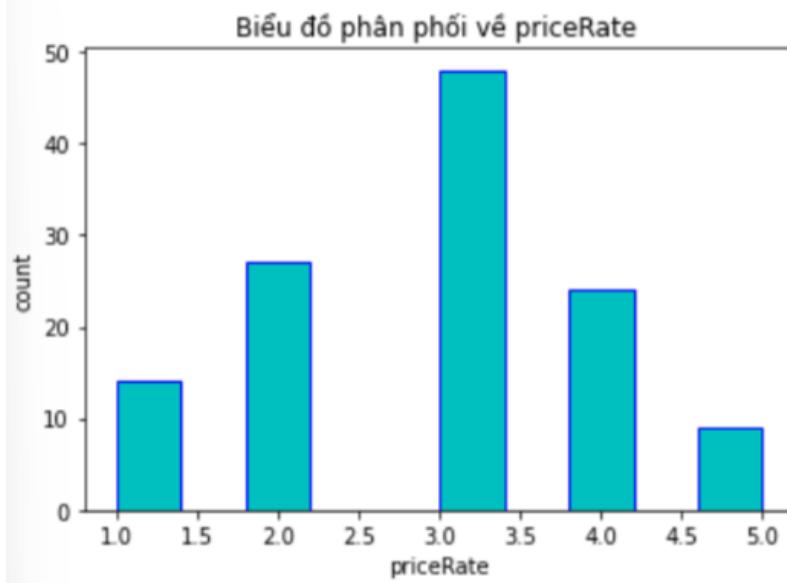
Biểu đồ 17: Biểu đồ phân phối của feature spendPurchase

✚ **Biểu đồ phân phối của feature productRate**



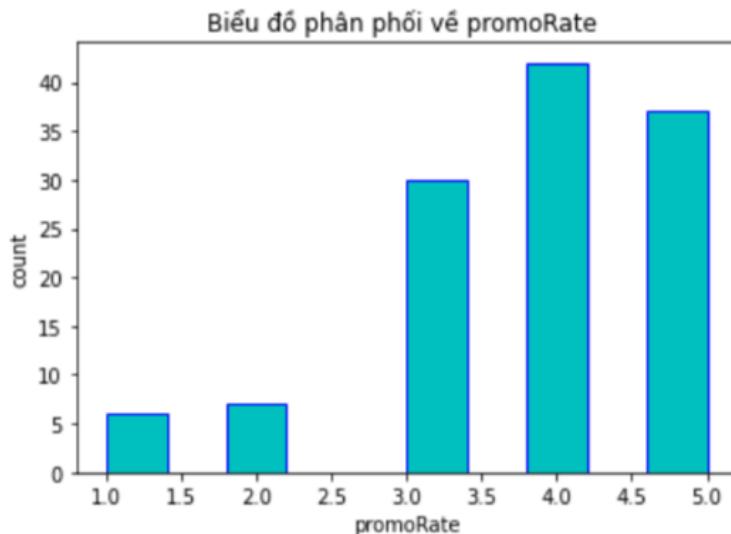
Biểu đồ 18: Biểu đồ phân phối của feature productRate

✚ **Biểu đồ phân phối của feature priceRate**



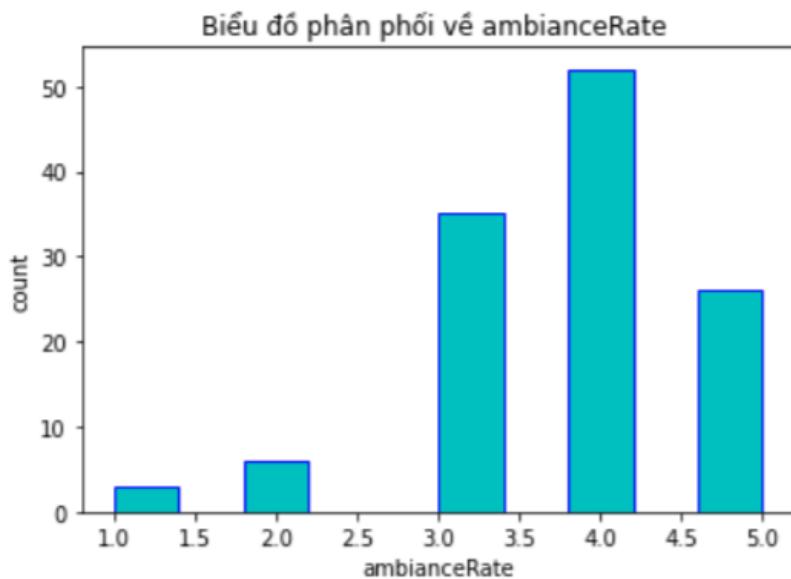
Biểu đồ 19: Biểu đồ phân phối của feature priceRate

✚ **Biểu đồ phân phối của feature promoRate**



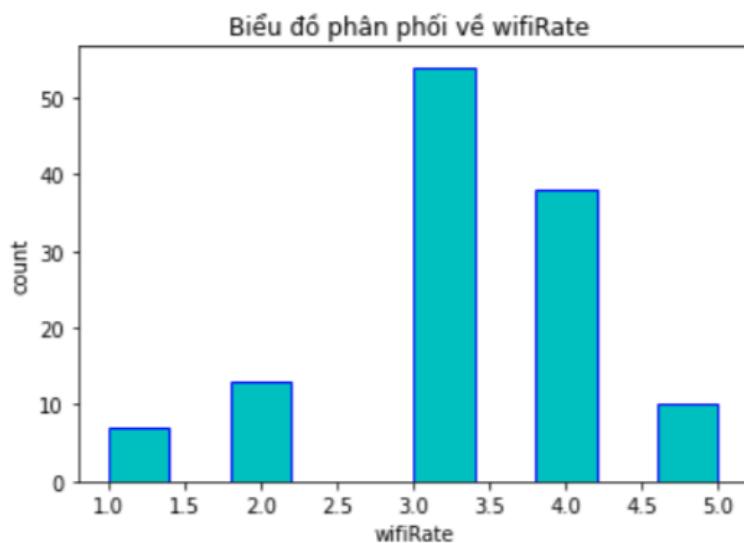
Biểu đồ 20: Biểu đồ phân phối của feature promoRate

✚ **Biểu đồ phân phối của feature ambianceRate**



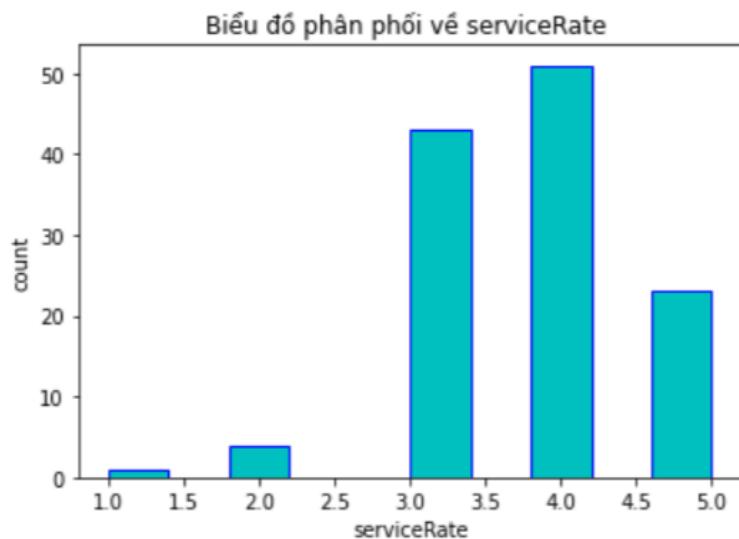
Biểu đồ 21: Biểu đồ phân phối của feature ambianceRate

✚ **Biểu đồ phân phối của feature wifiRate**



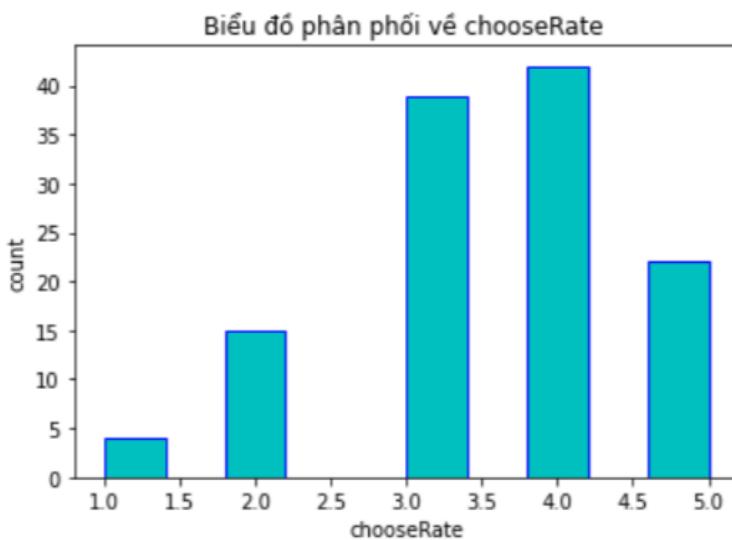
Biểu đồ 22: Biểu đồ phân phối của feature wifiRate

✚ **Biểu đồ phân phối của feature serviceRate**



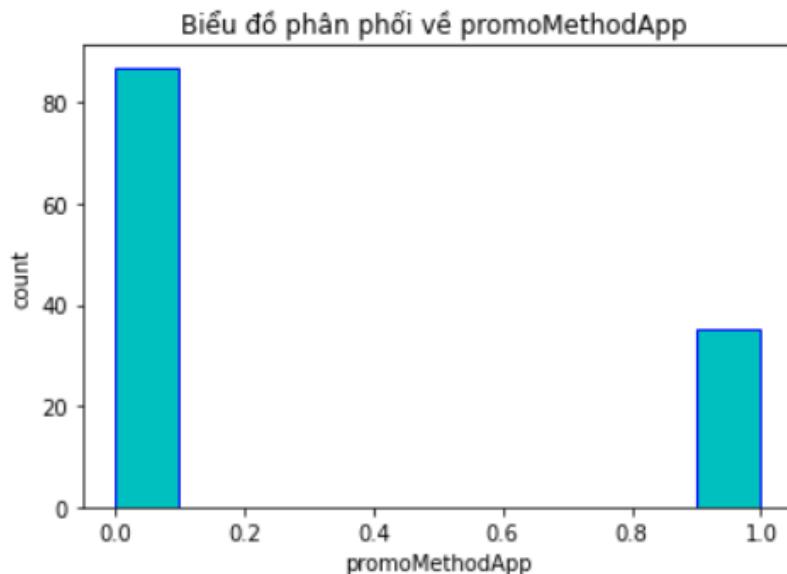
Biểu đồ 23: Biểu đồ phân phối của feature serviceRate

✿ **Biểu đồ phân phối của feature chooseRate**



Biểu đồ 24: Biểu đồ phân phối của feature chooseRate

✿ **Biểu đồ phân phối của feature loyal**



Biểu đồ 25: Biểu đồ phân phối của feature loyal

Nhận Xét: Vì mỗi biến feature đều có mức xếp hạng riêng biệt nên phần biểu đồ có sự khác biệt về số lượng cột và đơn vị.

1. feature gender không có sự chênh lệch giữa 2 giá trị nam và nữ (0 nữ, 1: nam), cụ thể :
 - Số người nữ là 65 người
 - Số người nam là 57 người
2. feature age có sự chênh lệch vượt trội là những người có độ tuổi từ 20 đến 29 tuổi còn lại khá tương đồng nhau, cụ thể:
 - số người có độ tuổi dưới 20 tuổi là 13 người
 - số người có độ tuổi từ 20 đến 29 tuổi là 85 người
 - số người có độ tuổi từ 30 đến 39 tuổi là 17 người
 - số người có độ tuổi trên 40 tuổi là 7 người
3. feature status có sự chênh lệch giữa 2 giá trị học sinh và người làm thuê so với 2 giá trị còn lại, cụ thể:
 - số người đang là học sinh là 42 người
 - số người đang là chủ hộ kinh doanh đến 29 tuổi là 17 người
 - số người đang là người làm thuê là 61 người
 - số người đang là nội trợ là 2 người
4. feature income có sự chênh lệch vượt trội là những người có nguồn thu nhập thấp hơn 25000 RM so với các giá trị còn lại, cụ thể:
 - số người có thu nhập dưới 25000 RM là 71 người

- số người có thu nhập từ 25000 đến 50000 RM là 25 người
- số người có thu nhập từ 50000 đến 100000 RM là 17 người
- số người có thu nhập từ 100000 đến 150000 RM là 3 người
- số người có thu nhập trên 150000 RM là 6 người

5. feature visitNo có sự chênh lệch vượt trội là những người ghé thăm hàng tháng so với các giá trị còn lại, cụ thể:

- số người hiếm khi ghé thăm là 2 người
- số người ghé thăm hàng ngày là 9 người
- số người ghé thăm mỗi tuần là 26 người
- số người ghé thăm mỗi tháng là 76 người

6. feature method khá tương đồng giữa các giá trị những người ở lại quán, cầm tay mang đi, lái xe mang đi, cao hơn hẳn các giá trị còn lại, cụ thể :

- số người ở lại quán là 46 người
- số người lái xe mang đi là 20 người
- số người cầm tay mang đi là 49 người
- số người khác là 7 người

7. feature timespend có sự giảm dần về thời gian ở lại quán với các mốc thời gian, cụ thể:

- số người có thời gian ở lại quán dưới 30 phút là 73 người
- số người có thời gian ở lại quán từ 30 phút đến 1 giờ là 34 người
- số người có thời gian ở lại quán từ 1 đến 2 giờ là 12 người
- số người có thời gian ở lại quán từ 2 đến 3 giờ là 1 người
- số người có thời gian ở lại quán trên 3 giờ là 2 người

8. feature location có sự tương đồng và tăng dần về khoảng cách đến cửa hàng, cụ thể :

- số người có khoảng cách đến cửa hàng dưới 1km là 27 người
- số người có khoảng cách đến cửa hàng từ 1 đến 3 km là 34 người
- số người có khoảng cách đến cửa hàng trên 3 km là 61 người

9. feature MembershipCard không có sự chênh lệch giữa 2 giá trị, cụ thể:

- số người có thẻ thành viên là 60 người
- số người không có thẻ thành viên là 62 người

10. feature itemPurchaseCoffee có sự chênh lệch giữa 2 giá trị, cụ thể:

- Số người gọi cà phê là 84 người
- Số người không gọi cà phê là 38 người

11. feature itemPurchaseCold có sự chênh lệch giữa 2 giá trị, cụ thể:

- Số người gọi nước lạnh là 84 người
- Số người không gọi nước lạnh là 38 người

12. feature itemPurchasePastries có sự chênh lệch giữa 2 giá trị, cụ thể:

- Số người gọi bánh ngọt là 4 người
- Số người không gọi bánh ngọt là 118 người

13. feature itemPurchasePastries có sự chênh lệch giữa 2 giá trị, cụ thể:

- Số người gọi nước ép là 16 người
- Số người không gọi nước ép là 106 người

14. feature itemPurchasePastries có sự chênh lệch giữa 2 giá trị, cụ thể:

- Số người gọi bánh mì sandwich là 8 người
- Số người không gọi bánh mì sandwich là 114 người

15. feature SpendPurchase có sự chênh lệch giữa 2 giá trị (dưới 20RM và từ 20 đến 40 RM) so với 2 giá trị còn lại, cụ thể:

- số người đến cửa hàng và không chi trả thứ gì là 7 người
- số người đến cửa hàng và chi số tiền dưới 20 RM là 58 người
- số người đến cửa hàng và chi số tiền từ 20 đến 40 RM là 45 người
- số người đến cửa hàng và chi số tiền trên 40 RM là 12 người

16. feature productRate có sự đánh giá của khách hàng tập trung ở mức bình thường, tốt, rất tốt, cụ thể:

- số người đánh giá chất lượng sản phẩm rất tệ là 3 người
- số người đánh giá chất lượng sản phẩm tệ là 8 người
- số người đánh giá chất lượng sản phẩm bình thường là 39 người
- số người đánh giá chất lượng sản phẩm tốt là 49 người
- số người đánh giá chất lượng sản phẩm rất tốt là 23 người

17. feature priceRate có sự đánh giá của khách hàng tập trung ở mức tệ, bình thường, tốt, cụ thể:

- số người đánh giá chất lượng sản phẩm rất tệ là 14 người
- số người đánh giá chất lượng sản phẩm tệ là 27 người
- số người đánh giá chất lượng sản phẩm bình thường là 48 người
- số người đánh giá chất lượng sản phẩm tốt là 24 người
- số người đánh giá chất lượng sản phẩm rất tốt là 9 người

18. feature promoRate có sự đánh giá của khách hàng tập trung ở mức tệ, bình thường, tốt, cụ thể:

- số người đánh giá chất lượng sản phẩm rất tệ là 6 người
- số người đánh giá chất lượng sản phẩm tệ là 7 người
- số người đánh giá chất lượng sản phẩm bình thường là 30 người
- số người đánh giá chất lượng sản phẩm tốt là 42 người
- số người đánh giá chất lượng sản phẩm rất tốt là 37 người

19. feature amnianceRate có sự đánh giá của khách hàng tập trung ở mức tệ, bình thường, tốt, cụ thể:

- số người đánh giá chất lượng sản phẩm rất tệ là 3 người
- số người đánh giá chất lượng sản phẩm tệ là 6 người
- số người đánh giá chất lượng sản phẩm bình thường là 35 người
- số người đánh giá chất lượng sản phẩm tốt là 52 người
- số người đánh giá chất lượng sản phẩm rất tốt là 26 người

20. feature wifiRate có sự đánh giá của khách hàng tập trung ở mức bình thường, tốt, cụ thể:

- số người đánh giá chất lượng sản phẩm rất tệ là 7 người
- số người đánh giá chất lượng sản phẩm tệ là 13 người
- số người đánh giá chất lượng sản phẩm bình thường là 54 người
- số người đánh giá chất lượng sản phẩm tốt là 38 người
- số người đánh giá chất lượng sản phẩm rất tốt là 10 người

21. feature serviceRate có sự đánh giá của khách hàng tập trung ở mức bình thường, tốt, rất tốt, cụ thể:

- số người đánh giá chất lượng sản phẩm rất tệ là 1 người
- số người đánh giá chất lượng sản phẩm tệ là 4 người
- số người đánh giá chất lượng sản phẩm bình thường là 43 người
- số người đánh giá chất lượng sản phẩm tốt là 51 người
- số người đánh giá chất lượng sản phẩm rất tốt là 23 người

22. feature chooseRate có sự đánh giá của khách hàng tập trung ở mức bình thường, tốt, rất tốt, cụ thể:

- số người đánh giá chất lượng sản phẩm rất tệ là 4 người
- số người đánh giá chất lượng sản phẩm tệ là 15 người
- số người đánh giá chất lượng sản phẩm bình thường là 39 người
- số người đánh giá chất lượng sản phẩm tốt là 42 người
- số người đánh giá chất lượng sản phẩm rất tốt là 22 người

23. feature promoMethodApp có sự chênh lệch giữa 2 giá trị, cụ thể :

- số người quan tâm là 35 người
- số người không quan tâm là 87 người

24. feature promoMethodSoc có sự chênh lệch giữa 2 giá trị, cụ thể:

- số người quan tâm là 33 người
- số người không quan tâm là 89 người

25. feature promoMethodEmail có sự chênh lệch giữa 2 giá trị, cụ thể:

- số người quan tâm là 18 người
- số người không quan tâm là 104 người

25. feature promoMethodDeal có sự chênh lệch giữa 2 giá trị cụ thể:

- số người quan tâm là 7 người
- số người không quan tâm là 115 người

26. feature promoMethodFriend có sự chênh lệch giữa 2 giá trị, cụ thể

- số người quan tâm là 49 người
- số người không quan tâm là 73 người

27. feature promoMethodDisplay có sự chênh lệch giữa 2 giá trị, cụ thể:

- số người quan tâm là 21 người
- số người không quan tâm là 101 người

28. feature promoMethodBillboard có sự chênh lệch giữa 2 giá trị, cụ thể:

- số người quan tâm là 11 người
- số người không quan tâm là 111 người

29. feature loyal không có sự chênh lệch giữa 2 giá trị, cụ thể:

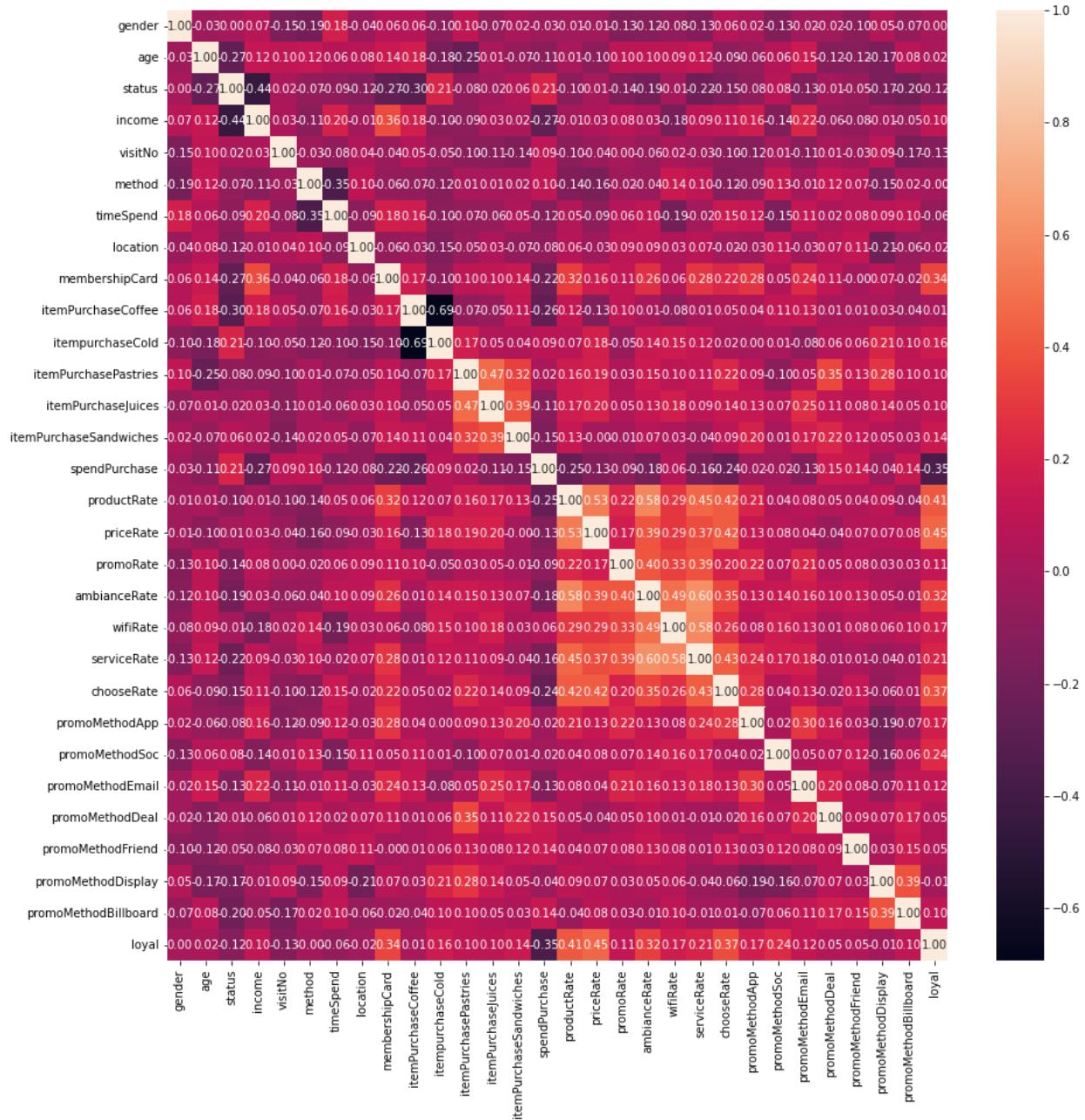
- số người trung thành với cửa hàng là 94 người
- số người không trung thành với cửa hàng là 28 người

3.3. Kiểm tra tính tương quan giữa các feature

Vẽ biểu đồ heatmap các feature sau khi lọc bỏ các feature có tương quan = 1 tạo thành list lst.

```
# Biểu đồ tương quan giữa các feature sau khi lọc bỏ các feature tương quan cực đại(max = 1)
lst = ['Id', 'gender', 'age', 'status', 'income', 'visitNo',
       'method', 'timeSpend', 'location', 'membershipCard',
       'spendPurchase', 'productRate', 'priceRate', 'promoRate',
       'ambianceRate', 'wifiRate', 'serviceRate', 'chooseRate', 'loyal'
      ]
plt.subplots(figsize=(15,15))
sns.heatmap(data[lst].corr(), annot = True, fmt = ".2f")
plt.show()
```

➤ *Biểu đồ thu được:*



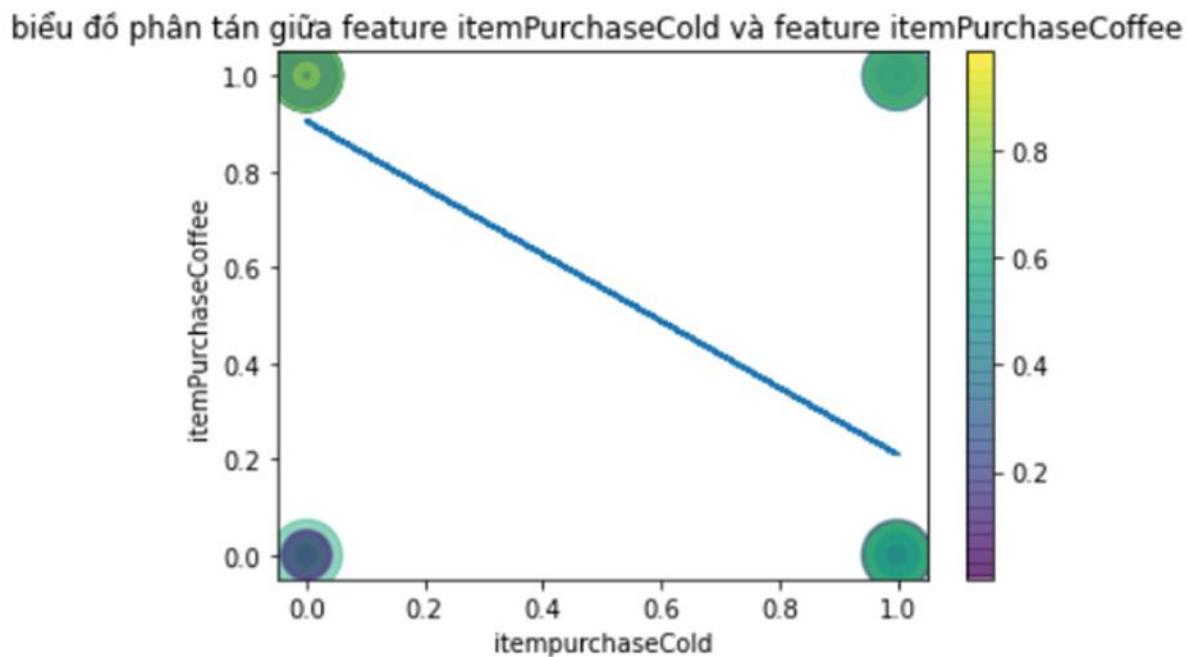
Biểu đồ 26: Biểu đồ heatmap thể hiện tương quan giữa các biến

Nhận Xét: Nhìn vào biểu đồ ta thấy:

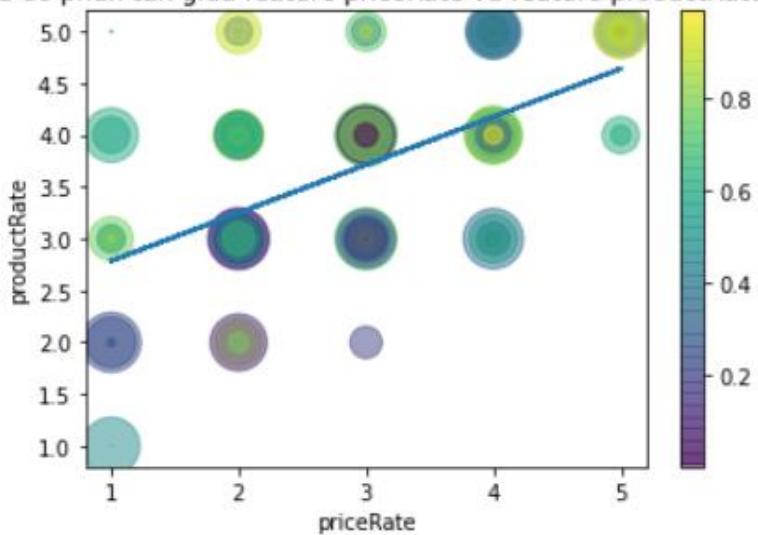
- Đa số các feature không tương quan nhau hoặc nếu có thì chỉ số tương quan rất bé ~ 0
- Một số cặp feature tương quan nghịch như:
 - feature itemPurchaseCold và feature itemPurchaseCoffee ~ - 0.69
- Một số cặp feature tương quan thuận như:
 - feature priceRate và feature productRate ~ 0.53
 - feature ambianceRate và feature productRate ~ 0.58
 - feature serviceRate và feature ambianceRate ~ 0.6
 - feature serviceRate và feature wifiRate ~ 0.58

3.4. Độ phân tán

Từ các cặp tương quan mạnh ta tiến hành vẽ các biểu đồ phân tán:

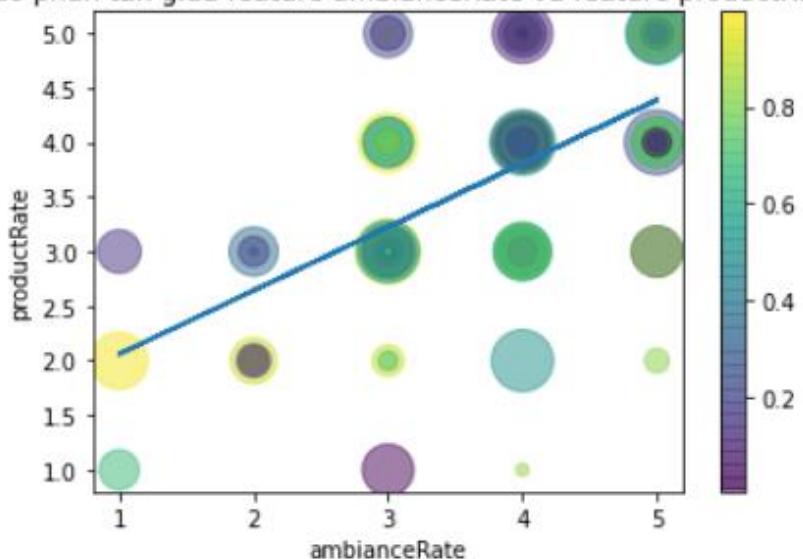


biểu đồ phân tán giữa feature priceRate và feature productRate



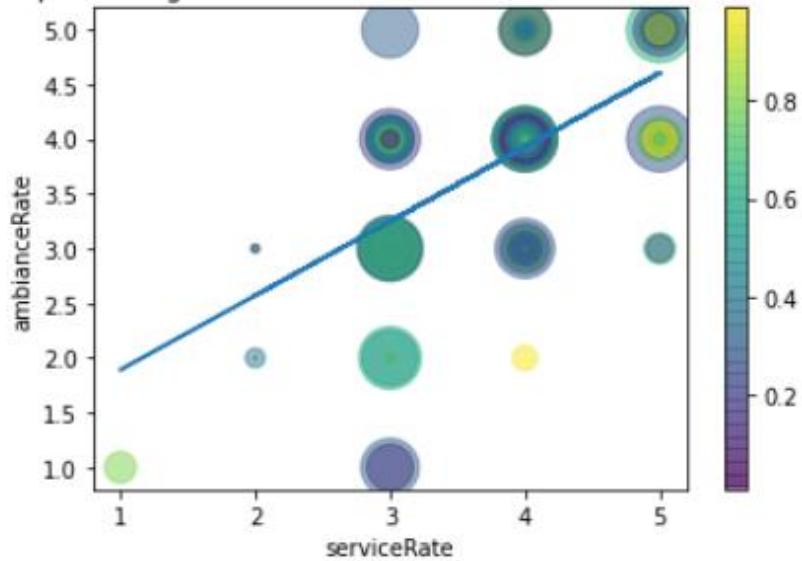
Biểu đồ 28: Biểu đồ phân tán giữa priceRate và productRate

biểu đồ phân tán giữa feature ambianceRate và feature productRate



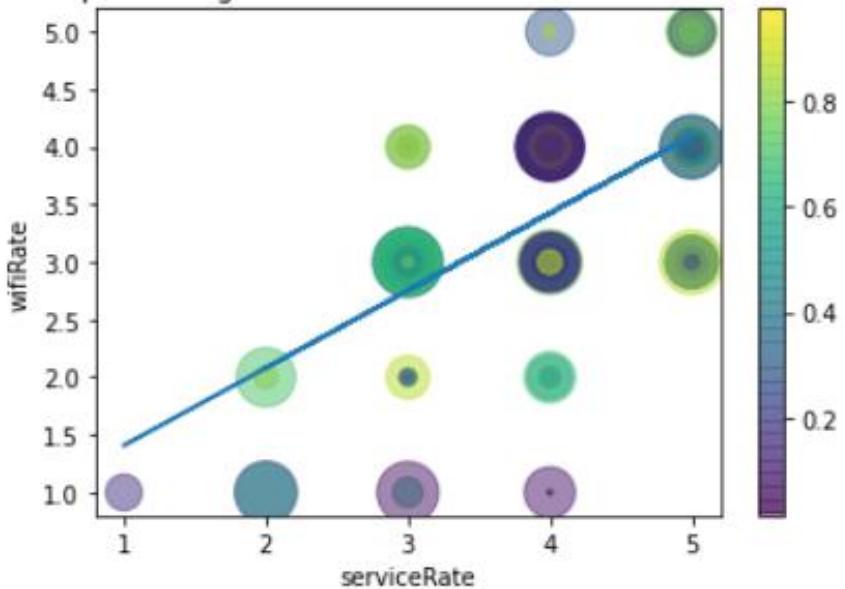
Biểu đồ 29: Biểu đồ phân tán giữa ambianceRate và productRate

biểu đồ phân tán giữa feature serviceRate và feature ambianceRate



Biểu đồ 30: Biểu đồ phân tán giữa serviceRate và ambianceRate

biểu đồ phân tán giữa feature serviceRate và feature wifiRate



Biểu đồ 31: Biểu đồ phân tán giữa serviceRate và wifiRate

CHƯƠNG 4. BIỂU ĐỒ THỂ HIỆN MỨC ĐỘ TRUNG THÀNH CỦA TỪNG ĐỘ TUỔI ỨNG VỚI TỪNG MỨC THU NHẬP

4.1. Biểu đồ thể hiện mức độ trung thành của từng độ tuổi ứng với từng mức thu nhập

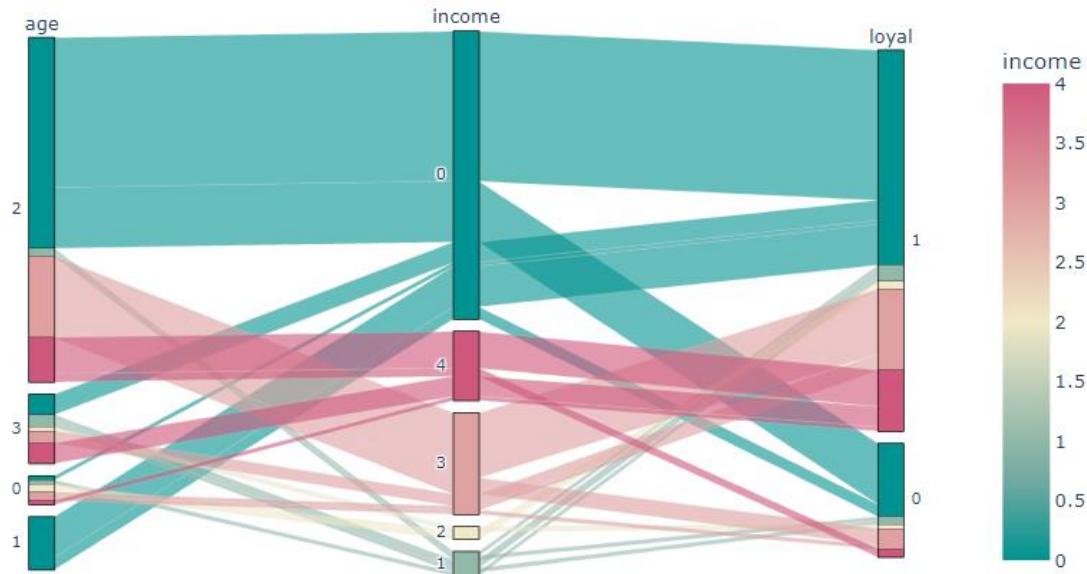
Vì là cửa hàng nổi tiếng và có giá thành cao nên mức thu nhập ảnh hưởng không ít đến mức độ trung thành của khách hàng bên cạnh đó mức thu nhập cũng bị ảnh hưởng bởi độ tuổi vô hình chung độ tuổi cũng có tác động đến mức độ trung thành.

Với mức thang đo rộng của biến income, ta sẽ xem xét 3 biến age, income và loyal với thang đo income. Nhóm sử dụng thư viện Plotly để vẽ biểu đồ parallel sets.

```
df=pd.DataFrame(data[['age','income','loyal']])
fig = px.parallel_categories(df,color = 'income', dimensions=['age', 'income', 'loyal'],
                             color_continuous_scale=px.colors.diverging.Tealrose,
                             color_continuous_midpoint=2,
                             title=' Biểu đồ thể hiện mức độ trung thành của từng độ tuổi ứng với từng mức thu nhập'
)
fig.show()
```

Hình 23: Code Python vẽ biểu đồ parrallel sets biến age, income, loyal

Biểu đồ thể hiện mức độ trung thành của từng độ tuổi ứng với từng mức thu nhập



Biểu đồ 32: Biểu đồ thể hiện mức độ trung thành của từng độ tuổi ứng với từng mức thu nhập

Tùy theo từng độ khoảng thu nhập và độ tuổi khác nhau mà dẫn đến sự trung thành đối với cửa hàng cũng khác nhau:

- Đối với mức độ tuổi 0 (dưới 20 tuổi)
 - Tất cả đều ở mức thu nhập dưới 25000 RM và có đến 2/3 là khách hàng trung thành, chỉ 1/3 là khách hàng không trung thành
- Đối với mức độ tuổi 3 (trên 40 tuổi)
 - Mức thu nhập của khách hàng trải khắp các khoảng và có đến 6/7 là khách hàng trung thành, chỉ 1/7 là khách hàng không trung thành
- Đối với mức độ tuổi 2 (từ 30 đến 40 tuổi)
 - Có 1/3 số khách hàng thuộc mức thu nhập dưới 25000 RM, 1/3 số khách hàng thuộc mức thu nhập từ 50000 đến 100000 RM, 1/3 số khách hàng có mức thu nhập trải khắp các mức còn lại
 - Đa số đều là khách hàng trung thành, tỉ lệ khách hàng trung thành chiếm ~ 90%
- Đối với mức độ tuổi 1 (từ 20 đến 29 tuổi)
 - Mức thu nhập dưới 25000 RM có tỉ lệ giữa khách hàng trung thành và khách hàng không trung thành là 3:1
 - Mức thu nhập từ 25000 đến 50000 RM có tỉ lệ giữa khách hàng trung thành và khách hàng không trung thành là 6:1
 - Mức thu nhập từ 50000 đến 100000 RM có tỉ lệ giữa khách hàng trung thành và khách hàng không trung thành là xấp xỉ 5:1
 - Mức thu nhập trên 150000 RM có 100% đều là khách hàng trung thành

4.2. Biểu đồ thể hiện mức độ chi trả tiền của người những người thường xuyên đến cửa hàng

Vì Starbuck là một cửa hàng đồ uống nổi tiếng thành ra giá thành khá cao vậy nên với tần suất xuất hiện tại cửa hàng để thưởng thức một tách trà hay cái bánh càng nhiều thì sẽ ảnh hưởng đến số tiền chi ra mỗi lần đến cửa hàng như thế nào?

Tạo bảng gồm cột visitNo và cột spendPurcse và tính tổng theo 2 thuộc tính thành cột count

```

df=pd.DataFrame(data[['visitNo','spendPurchase']])
dfgb=df.groupby(by=['visitNo','spendPurchase'])
s=dfgb.size()
df_gb=s.reset_index(name='counts')
print(df_gb)

```

Hình 24: Code Python tạo bảng gồm cột visitNo và cột spendPurchase và tính tổng theo 2 thuộc tính thành cột count

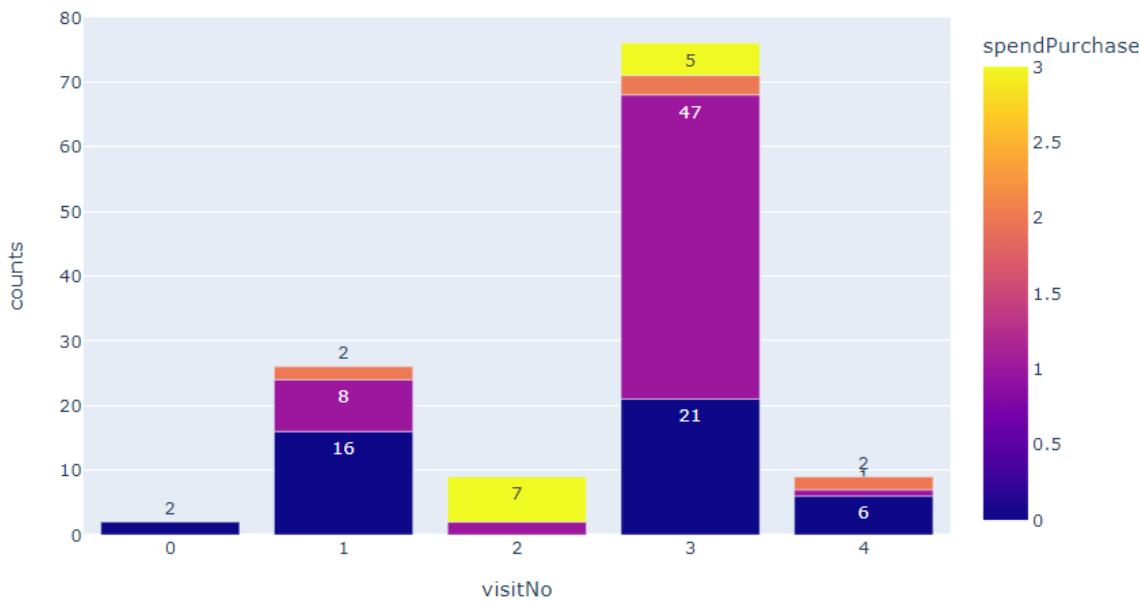
➤ *Ta có bảng:*

	visitNo	spendPurchase	counts
0	0	2	2
1	1	1	1
2	1	2	6
3	1	3	2
4	2	1	8
5	2	2	16
6	2	3	2
7	3	0	5
8	3	1	47
9	3	2	21
10	3	3	3

Bảng 6: Bảng gồm cột visitNo và cột spendPurchase và tính tổng theo 2 thuộc tính thành cột count

Sử dụng biểu đồ Bar từ thư viện plotly với biến visitNo làm trục hoành, biến counts làm trục tung và phân biệt loại bằng cách sử dụng màu là biến spendPurchase.

Biểu đồ thể hiện mức độ chi trả tiền của người những người thường xuyên đến cửa hàng



Biểu đồ 33: Biểu đồ thể hiện mức độ chi trả tiền của người những người thường xuyên đến cửa hàng

Nhận Xét:

1. Toàn bộ những người hiếm khi đến cửa hàng sẽ chi từ 20 đến 40 RM
2. Với những người ghé thăm mỗi ngày, phần lớn chi mua từ 20 đến 40 RM:
 - Phần lớn chi từ 20 đến 40 RM chiếm xấp xỉ 66.7%
 - Số khác chi từ hơn 40 RM chiếm xấp xỉ 22.2%
 - Còn lại chi dưới 20 RM chiếm xấp xỉ 11.1%
3. Với những người ghé thăm mỗi tuần, phần lớn chi mua từ 20 đến 40 RM :
 - Phần lớn từ 20 đến 40RM chiếm xấp xỉ 61.5%
 - Số khách chi dưới 20 RM chiếm xấp xỉ 30.8%
 - Còn lại chi trên 40RM chiếm xấp xỉ 7.7%
4. Với những người đến thăm mỗi tháng, phần lớn chi mua dưới 20 RM:
 - Phần lớn chi dưới 20 RM chiếm xấp xỉ 61.8%
 - Số khác chi từ 20 đến 40 RM chiếm xấp xỉ 27.6%
 - Đặc biệt không chi mua bất cứ thứ gì chiếm xấp xỉ 6.6%
 - Còn lại chi trên 40 RM chiếm xấp xỉ 4%

- **Kết luận:** Những người có tiền suýt ghé thăm cửa hàng càng lớn sẽ chi mức tiền vừa phải; còn những người có tiền suýt ghé thăm cửa hàng càng ít sẽ chi mức tiền ít hơn riêng trường hợp hiếm khi ghé thăm cửa hàng, khách hàng thường có xu hướng chi số tiền tầm trung như những người ghé thăm cửa hàng tiền suýt cao.

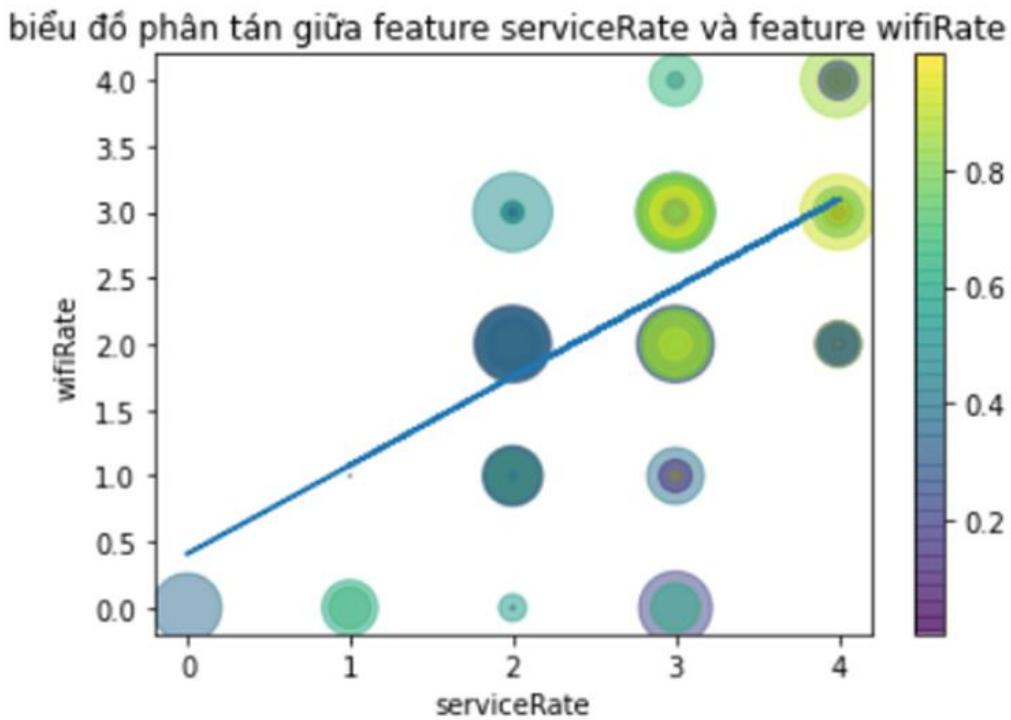
4.3. Mức độ ảnh hưởng của wifi đối với nhận định chất lượng dịch vụ

Khách hàng đến cửa hàng tập trung ở độ tuổi trẻ, năng động với công nghệ vậy nên internet là không thể thiếu. Vậy nên chất lượng wifi có mối quan hệ mật thiết với nhận định chất lượng dịch vụ của cửa hàng.

Vẽ biểu đồ Scatter từ thư viện matplotlib với biến x là thuộc tính serviceRate và biến y là thuộc tính wifiRate kết hợp đường hồi quy tuyến tính là biểu đồ Line từ thư viện matplotlib, đường thẳng được nối từ các cặp biến tương ứng. Ta sử dụng cột màu và lấy đó làm mức độ tương đồng của 2 thuộc tính.

```
x = data['serviceRate']
y = data['wifiRate']
m, b = np.polyfit(x, y, 1)
plt.plot(x, m*x+b)
colors = np.random.rand(N)
area = (30 * np.random.rand(N))**2
plt.xlabel('serviceRate')
plt.ylabel('wifiRate')
plt.title('biểu đồ phân tán giữa feature serviceRate và feature wifiRate ')
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.colorbar()
plt.show()
```

Hình 25: Code Python vẽ biểu đồ phân tán mức độ ảnh hưởng của wifi đối với nhận định chất lượng dịch vụ



Biểu đồ 34: Biểu đồ phân tán thể hiện mức độ ảnh hưởng của wifi đối với nhận định chất lượng dịch vụ

Nhận Xét:

Với đường hồi quy tuyến tính như hình ta có thể kết luận 2 thuộc tính wifiRate và serviceRate có mối tương quan thuận.

Với mỗi điểm có các lớp màu khác nhau, xét cặp số (wifiRate;serviceRate):

- Các mức xuất hiện chỉ số tương đồng rất cao (màu vàng): (3;3), (3;4)
- Các mức xuất hiện chỉ số tương đồng cao (màu xanh): (2;3), (0;1), (2;3), (4;3)
- Các điểm còn lại điểm tương đồng không cao

➤ **Kết luận:** Cho thấy nếu chất lượng wifi đảm bảo tốt thì cửa hàng sẽ dễ nhận được đánh giá tốt về mặt dịch vụ và ngược lại.

4.4. Biểu đồ đánh giá chất lượng dịch vụ theo từng độ tuổi

Các độ tuổi khác nhau sẽ có những đánh giá và yêu cầu khác nhau đối với chất lượng dịch vụ. Ở đây, ta có biến serviceRate - đây là một biến thể hiện đánh giá dịch vụ từ khách hàng theo thang điểm từ 0 đến 4 (tốt đến rất tốt).

Nhóm sẽ tiến hành tạo một dữ liệu bảng chéo lấy tên servicerate_crosstab thể hiện sự tương quan giữa biến “age” và biến “serviceRate” bằng hàm crosstab. Sau đó dùng hàm rename

để đổi tên của các mã hóa trong biến “age” tương ứng với các tên 0:'40 and above',1:'Below 20',2:'From 20 to 29',3:'From 30 to 39'.

```
[ ] servicerate_crosstab=pd.crosstab(data['age'], data['serviceRate'], normalize='index')
servicerate_crosstab=servicerate_crosstab.rename(index={0:'40 and above',1:'Below 20',2:'From 20 to 29',3:'From 30 to 39'})
servicerate_crosstab
```

Hình 26: Code Python tạo bảng chéo thể hiện sự tương quan giữa biến “age” và biến “serviceRate”

➤ Ta thu được bảng sau:

serviceRate	0	1	2	3	4
age					
40 and above	0.000000	0.000000	0.428571	0.428571	0.142857
Below 20	0.000000	0.076923	0.538462	0.230769	0.153846
From 20 to 29	0.011765	0.035294	0.329412	0.447059	0.176471
From 30 to 39	0.000000	0.000000	0.294118	0.411765	0.294118

Bảng 7: Bảng chéo lấy tên servicerate_crosstab thể hiện sự tương quan giữa biến “age” và biến “serviceRate”

Sau khi có được dữ liệu bảng chéo, ta sử dụng hàm plot và loại biểu đồ là bar plot (kind = ‘bar’) để vẽ biểu đồ thể hiện sự đánh giá chất lượng dịch vụ theo các nhóm tuổi dựa trên bảng chéo đó. Với x là biến ‘Age’, y là tần suất của các giá trị tương ứng.

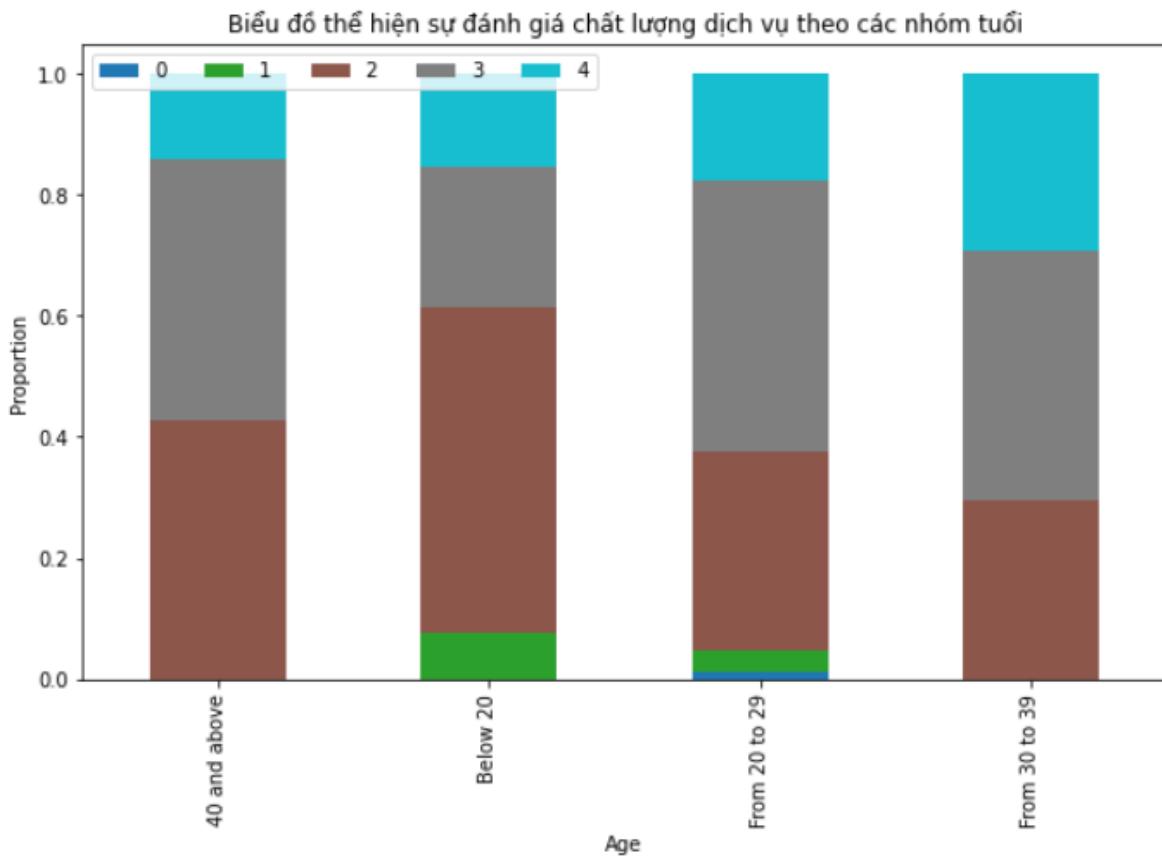
```
servicerate_crosstab.plot(kind='bar',
                           stacked=True,
                           colormap='tab10',
                           figsize=(10, 6))

plt.legend(loc="upper left", ncol=5)
plt.xlabel("Age")
plt.ylabel("Proportion")
plt.title("Biểu đồ thể hiện sự đánh giá chất lượng dịch vụ theo các nhóm tuổi")

plt.show()
```

Hình 27: Code Python tạo biểu đồ thanh đánh giá chất lượng dịch vụ theo từng độ tuổi

➤ Biểu đồ thu được như sau:



Biểu đồ 35: Biểu đồ đánh giá chất lượng dịch vụ theo từng độ tuổi

Nhận Xét: Dựa trên biểu đồ thu được ta có thể thấy rằng những người từ độ tuổi 30-39 sẽ có nhiều đánh giá rất tích cực với nhiều thang điểm 4 hơn hẳn so với các độ tuổi khác, độ tuổi 40 trở lên và 20-29 cũng có nhiều đánh giá tích cực không kém với khá nhiều thang điểm 3. Với độ tuổi dưới 20, chất lượng dịch vụ ở Starbucks chỉ nằm ở mức 2 là được đánh giá nhiều. Ngoài ra, vẫn còn tồn tại một số đánh giá 0 (rất tệ) ở độ tuổi từ 20-29 dù đây là độ tuổi có nhiều người ghé cửa hàng nhất.

4.5. Biểu đồ phân tích đánh giá chất lượng cửa hàng

Việc phân tích đánh giá chất lượng cửa hàng sẽ đem lại cái nhìn tổng quan cho mọi người. Ở đây nhóm sẽ sử dụng các dữ liệu về Rate để thống kê lại sau đó tiến hành vẽ biểu đồ cột.

Đầu tiên, tạo một cột index bằng hàm `reset_index()`. Sau đó định nghĩa các hàm `plot_bar`, `organize_plot_data` và `generate_plot`. Với hàm `organize` sẽ thực hiện việc đếm các giá trị xem thử có bao nhiêu giá trị tương ứng với thang điểm đánh giá, sau đó sẽ tiến hành sử dụng hàm `generate_plot` để truyền vào hàm `plt.bar` và đặt tiêu đề cho biểu đồ. Cuối cùng hàm `plot_bar` sẽ truyền vào cả 2 hàm vừa được tạo ở trên để cấu tạo nên một biểu đồ hoàn chỉnh. Tới đây, việc

của chúng ta chỉ là thiết lập đưa hàm plot_bar vào code và truyền vào các feature tương ứng với các đánh giá.

```
[ ] data = data.reset_index()
def plot_bar(feature):
    plot_data = organize_plot_data(feature)
    generate_plot(plot_data,feature)

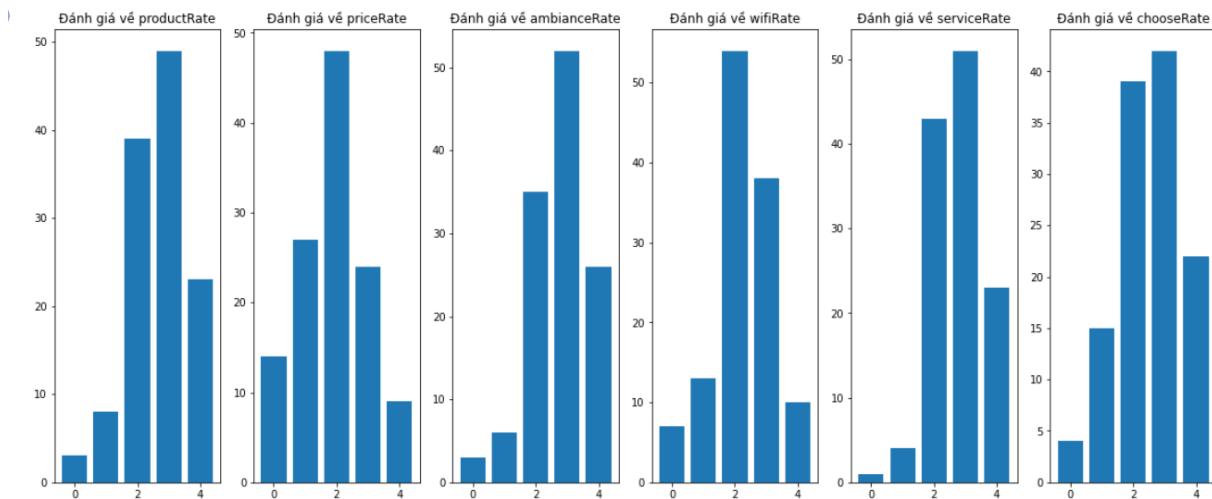
def organize_plot_data(feature):
    plot_data = data[['index',feature]]
    plot_data = plot_data.groupby(feature).count()
    plot_data = plot_data.reset_index()
    plot_data.columns = [feature,'Counts']
    return plot_data

def generate_plot(plot_data,feature):
    plt.bar(x = plot_data[feature], height = plot_data['Counts'])
    plt.title('Đánh giá về {}'.format(feature))

[ ] plt.figure(figsize=(20,8))
plt.subplot(1,6,1)
plot_bar(feature='productRate')
plt.subplot(1,6,2)
plot_bar(feature ='priceRate')
plt.subplot(1,6,3)
plot_bar(feature ='ambianceRate')
plt.subplot(1,6,4)
plot_bar(feature ='wifiRate')
plt.subplot(1,6,5)
plot_bar(feature ='serviceRate')
plt.subplot(1,6,6)
plot_bar(feature ='chooseRate')
```

Hình 28: Code Python định nghĩa các hàm plot_bar, organize_plot_data và generate_plot và vẽ biểu đồ

➤ **Biểu đồ thu được:**



Biểu đồ 36: Biểu đồ thanh phân tích đánh giá chất lượng cửa hàng

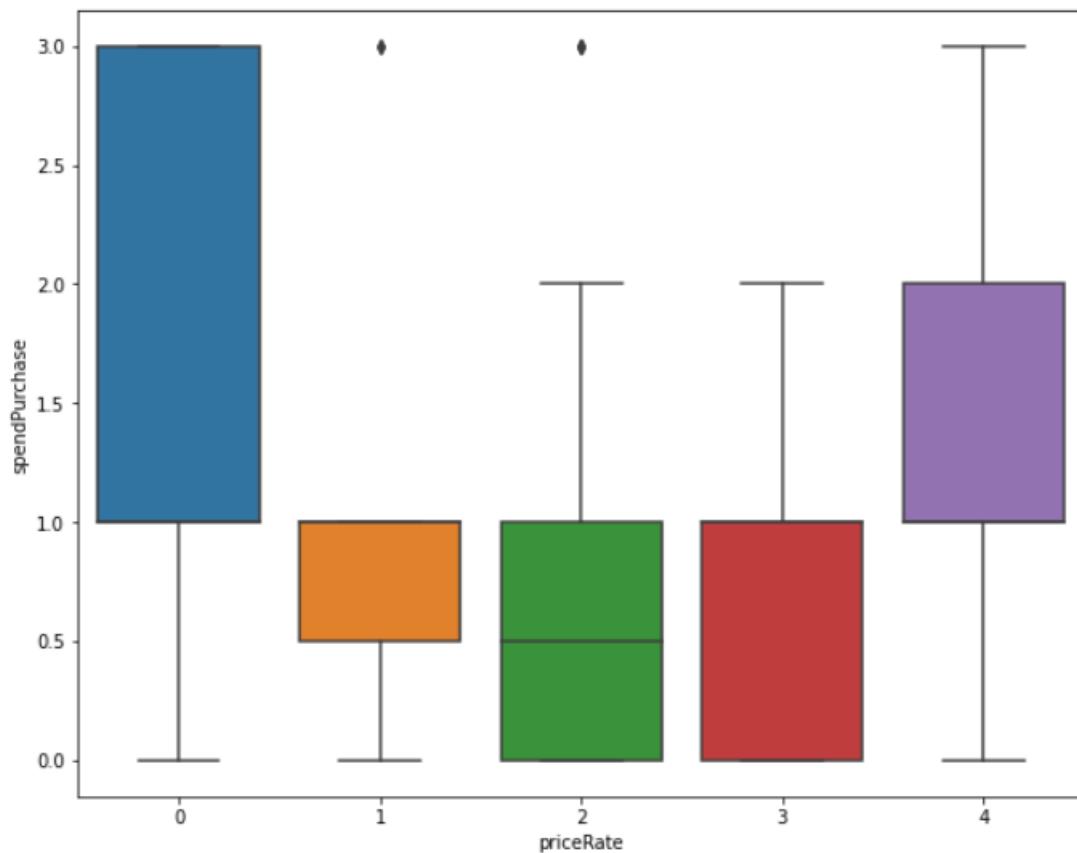
Nhận Xét: Qua biểu đồ trên ta có thấy thấy rằng có rất ít đánh giá 0 và hầu hết đều sẽ có đánh giá từ 2 trở lên trên toàn bộ 5 biểu đồ. Có thể nhận định rằng Starbucks có một chất lượng dịch vụ khá tốt đối với nhiều khách hàng.

4.6. Biểu đồ đánh giá sự phân bổ dữ liệu giữa đánh giá về mức giá và trung bình dành thời gian bao nhiêu cho một lần ghé thăm cửa hàng.

Sử dụng thư viện seaborn và hàm boxplot để tiến hành trực quan kết quả để có một cái nhìn rõ ràng hơn về sự phân bổ này.

```
plt.figure(figsize=(10,8))
sns.boxplot(x="priceRate",y="spendPurchase",data=data)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f55ffe6da60>



Biểu đồ 37: Biểu đồ đánh giá sự phân bố dữ liệu giữa đánh giá về mức giá và trung bình dành thời gian bao nhiêu cho một lần ghé thăm cửa hàng.

Nhận Xét:

- Đối với những đánh giá ở mức 0, 1, 2 dữ liệu tập trung ở 1-0, có thể hiểu rằng thường khách hàng dành ít thời gian hơn và tồn tại outlier không dành thời gian cho một lần ghé thăm.
- Đối với những đánh giá ở mức 3, dữ liệu tập trung ở 0 - 1.0, đây là mức thể hiện thời gian thời gian tương đối dành cho cửa hàng.
- Đối với những đánh giá ở mức 4, dữ liệu tập trung 1.0-2.0 ở mức cân bằng, đây là mức thể hiện thời gian tương đối và nhiều dành cho cửa hàng.

4.7. Biểu đồ thể hiện tỷ lệ thời gian khách hàng ở lại quán so với sự hài lòng về Starbucks:

Trong thực tế, khi một khách hàng yêu thích một quán nước (nổi tiếng hoặc bình dân) ở Việt Nam, thường mọi người có xu hướng ngồi lại quán khá lâu (từ 2 giờ đồng hồ trở lên). Và có thể tỷ lệ khách hàng ngồi lại quán trên 3 giờ đồng hồ chiếm ưu thế.

Ở trong bộ dữ liệu này, biến ‘loyal’ thể hiện sự đánh giá về mức độ hài lòng của khách hàng khi đến Starbucks, gồm 2 loại dữ liệu: 1 - thể hiện khách hàng hài lòng, 0 - thể hiện khách hàng không hài lòng. Và biến ‘timeSpend’ thể hiện các mốc thời gian khách hàng ở lại quán được chia theo: 0 - không quá 30 phút, 1- từ 30 phút đến 1h, 2 - 1h đến 2h, 3 - 2h đến 3h, 4 - hơn 3h. *Vậy tỷ lệ khách hàng đánh giá là hài lòng (loyal = 1) có thời gian ở lại quán hơn 3 giờ là chiếm ưu thế không?*

Ta sẽ lọc các dữ liệu theo yêu cầu và vẽ biểu đồ để trực quan hóa dữ liệu:

```
▶ # Số lượng khách hàng hài lòng về Starbucks so với thời gian khách hàng ở lại quán
df = data[data['loyal']==1].groupby('timeSpend')['loyal'].count().to_frame()
display(df)

# Biểu diễn dưới dạng biểu đồ
plt.pie(df['loyal'], labels=df.index, autopct='%1.2f%%', radius=1.2)
plt.title('Biểu đồ thể hiện sự hài lòng của KH\nso với thời gian ở lại quán'
          , color='#8B3A3A', fontsize=14)
plt.legend(title='Ghi chú:')
plt.show()
```

Hình 29: Code Python vẽ biểu đồ thể hiện tỷ lệ thời gian khách hàng ở lại quán so với sự hài lòng về Starbucks

➤ Kết quả được xuất ra dưới dạng bảng:

timeSpend	loyal
0	56
1	11
2	1
3	26

➤ Kết quả biểu diễn trực quan:



Biểu đồ 38: Biểu đồ thể hiện sự hài lòng của KH so với thời gian ở lại quán

Nhận Xét: Từ các kết quả trên, ta thấy khi khách hàng đánh giá là hài lòng nhưng thời gian khách ở lại quán không quá 30 phút chiếm tỷ lệ cao nhất (59.57%). Và thời gian để khách ở lại từ 30 phút đến 1 giờ (27.66%) chiếm tỷ lệ cũng khá cao, vì khách hàng đánh giá là hài lòng mà chỉ ở lại quán không quá 1 giờ, rất khác so với thực tế mà chúng ta nhận định. Còn từ 2 đến 3 giờ chỉ chiếm có 1.06%, đây là con số rất nhỏ so với tổng thể khách hàng đánh giá là hài lòng. Chúng ta có thể nói khách hàng yêu thích quán nhưng không nhất thiết là sẽ ngồi tại quán lâu, mà đánh giá cao các dịch vụ về chất lượng. *Vậy còn nếu khách hàng đánh giá không hài lòng với quán thì tỷ lệ về thời gian có gì thay đổi không?*

4.8. Biểu đồ thể hiện tỷ lệ thời gian khách hàng ở lại quán so với sự không hài lòng về Starbucks:

Ta tiếp tục lọc các dữ liệu và vẽ biểu đồ để trực quan hóa dữ liệu:

```
# Số lượng khách hàng không hài lòng về Starbucks so với thời gian khách hàng ở lại quán
df1 = data[data['loyal']==0].groupby('timeSpend')['loyal'].count().to_frame()
display(df1)

# Biểu diễn dưới dạng biểu đồ
plt.pie(df1['loyal'], labels=df1.index, autopct='%.1f%%', radius=1.2)
plt.title('Biểu đồ thể hiện sự không hài lòng của KH\n so với thời gian ở lại quán'
          , color='#8B3A3A', fontsize=14)
plt.legend(title='Ghi chú:')
plt.show()
```

Hình 30: Code Python vẽ biểu đồ thể hiện sự không hài lòng của KH so với thời gian ở lại quán

➤ **Kết quả được xuất ra dưới dạng bảng:**

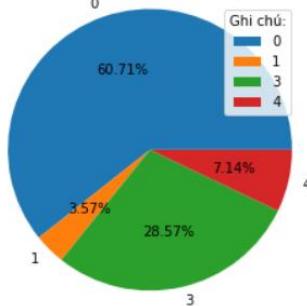
loyal

timeSpend

0	17
1	1
3	8
4	2

➤ *Kết quả biểu diễn trực quan:*

Biểu đồ thể hiện sự không hài lòng của KH so với thời gian ở lại quán



Biểu đồ 39: Biểu đồ thể hiện sự không hài lòng của KH so với thời gian ở lại quán

Nhận Xét: Từ kết quả khách hàng đánh giá không hài lòng với quán nước, ta thấy số lượng có thể giảm đi ở mục đánh giá này, nhưng tỷ lệ khách ở lại quán không quá 30 phút vẫn chiếm rất cao (60.71%). Và khi này tỷ lệ khách hàng ở lại từ 30 phút đến 1 giờ vẫn chiếm tỷ lệ rất cao (28.57%), tỉ lệ tăng lên không đáng kể so với khi khách hàng đánh giá là hài lòng nhưng vẫn chiếm phần tương đối cao. Khi này, thì khách hàng ở lại quán hơn 3 giờ lại có phần cao hơn khi khách hàng đánh giá là hài lòng, vậy mình có thể nói là do khách hàng ở lại trải nghiệm các dịch vụ về phục vụ và view ở quán thì khách mới đánh giá là không hài lòng (đây là nhận định cá nhân, chúng ta cần kiểm định sau).

Dựa vào 2 bảng số thể hiện số lượng khách hàng đánh giá về Starbucks theo thời gian ở lại quán (biến timeSpend) thì ta thấy số lượng khách đánh giá không hài lòng (số lượng: 28) ít hơn là khách hài lòng (số lượng: 94) về quán ($94 > 28$), như vậy ta có thể cho rằng Starbuck rất được mọi người yêu thích.

Dựa vào 2 biểu đồ trên, thì ta thấy được mặc dù khách hàng đánh giá hài lòng hay không hài lòng về Starbucks vì số lượng khách ở lại quán dưới 1 giờ đều chiếm số lượng lớn (lớn hơn 50% trên tổng thể). Điều này cho thấy khách hàng không ngồi ở lại quán và có thể hình thức mua nước tại quán là mang về.

Từ hai nhận xét trên thì ta có thể cho rằng thời gian ở lại quán không ảnh hưởng nhiều đến với sự đánh giá mức độ hài lòng của khách hàng. Dù cho số lượng khách có hài lòng về quán chiếm số lượng nhiều nhất.

Từ những điều trên, ta có thể cho rằng biến timeSpend không ảnh hưởng đến sự đánh giá của khách hàng. Và tiếp sau, ta sẽ thử kiểm định xem hai biến này có thực sự không tương quan hay không?

Ta thực hiện một phép kiểm định giữa biến ‘loyal’ và biến ‘timeSpend’:

```
▶ r, pvalue = stats.pearsonr(data['timeSpend'], data['loyal'])
print(f'Hệ số tương quan = {r:.4f}, p_values = {pvalue:.4f}')
```

⇨ Hệ số tương quan = -0.0648, p_values = 0.4786

Giả thiết kiểm định của ta là: H0: 2 biến không tương quan với nhau

Nhận Xét: Ta thấy trị số p value rất cao (xấp xỉ 48%) nên không bác bỏ H0, và hai biến này không tương quan với nhau. Và hệ số tương quan xấp xỉ tiến về 0 nên hai biến có không có tương quan, nên ta kết luận rằng hai biến loyal và biến timeSpend không tương quan với nhau. Như vậy thời gian khách hàng ở lại lâu hay ít không ảnh hưởng gì đến đánh giá sự hài lòng của khách hàng.

4.9. Tỷ lệ khách hàng ở lại quán so với phương thức mua nước tại Starbucks

Từ nhận định ở trên, khách hàng không đánh giá mức độ hài lòng của mình dựa theo thời gian ở lại quán, vậy số lượng khách ở lại quán (timeSpend) lâu có phụ thuộc đến phương thức mua nước tại quán (method) hay không khi khách hàng không hài lòng về quán? Các phương thức mua nước tại quán bao gồm: 0 - Uống tại quán, 1 - Gọi nước bên ngoài quầy dành cho khách đi xe hơi, 2 - Mang đi, 3 - Chưa bao giờ mua nước, 4 - Phương thức khác (như là được tặng hoặc là đặt giao đến). Ta tiến hành kiểm chứng các nhận định trên.

Gom nhóm các phương thức mua nước và thời gian khách hàng ở lại quán:

```
▶ df3 = data[data['loyal']==0].groupby(['method','timeSpend'])['loyal'].count().to_frame()
df3 = df3.reset_index()
df3
```

Hình 31: Code Python gom nhóm các phương thức mua nước và thời gian khách hàng ở lại quán

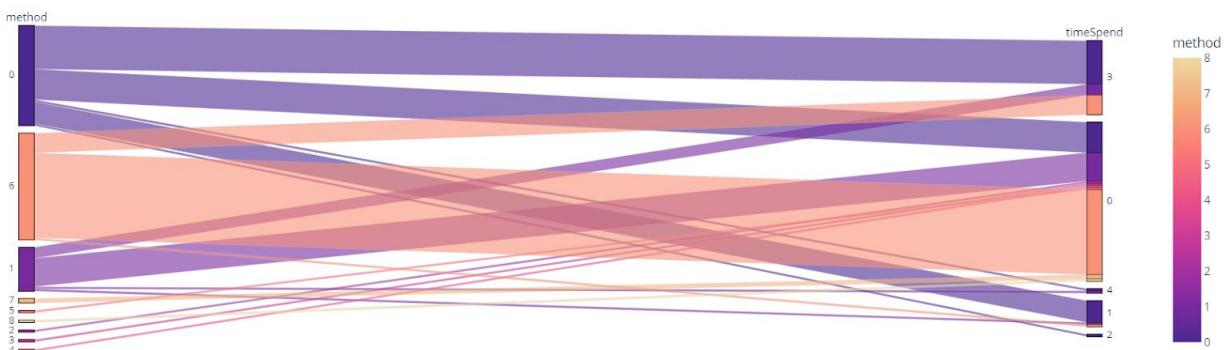
➤ **Ta thu được kết quả:**

	method	timeSpend	loyal
0	0	0	4
1	0	1	1
2	0	3	6
3	0	4	1
4	1	0	1
5	1	3	1
6	1	4	1
7	4	0	1
8	5	0	1
9	6	0	8
10	6	3	1
11	7	0	1
12	8	0	1

Bảng 8: Bảng dữ liệu sau khi gom nhóm các phương thức mua nước và thời gian khách hàng ở lại quán

Dựa vào bảng dữ liệu đã được xử lý, ta nhìn thấy được khi method là 6 (tức là chưa bao giờ mua nước tại quán - 'I don't like coffee') có thời gian ở lại quán dưới 30 phút chiếm số lượng cao nhất (8), có thể người này ghé qua mua nước cùng bạn bằng hình thức mang về. Và số lượng khách mua nước và uống ở tại quán (method = 0) và thời gian ở lại quán trên 3 giờ cũng có số lượng rất cao, rất phù hợp với thực tế. Ta sẽ hiểu rõ hơn thông qua biểu đồ bên dưới:

Biểu đồ thể hiện tỷ lệ của KH hài lòng về quán của cách thức mua nước và thời gian ở lại quán



Biểu đồ 40: Biểu đồ thể hiện lệ khách hàng ở lại quán so với phương thức mua nước tại Starbucks

Nhận Xét: Từ biểu đồ trên, thời gian ở lại bao gồm 0, 1, 2, 3 được chia tỷ lệ rất hợp lý dựa trên thực tế về các phương thức mua nước ở quán.

Còn khách hàng ở lại không quá 2 giờ thì được chia đều cho các phương thức tùy chọn của khách hàng. Ta thấy khi method = 6 (tức là khách hàng không thích dùng cafe hoặc là không sử dụng cái dịch vụ sản phẩm hoặc phục vụ ở đây) thì thời gian ở lại quán (timeSpend) không quá 30 phút là chiếm tỷ lệ nhiều nhất, còn các phần còn lại thì được chia nhỏ ra.

Còn khi method = 0 (tức là khách hàng ngồi lại quán) và phần tỷ lệ của thời gian ở lại quán trên 3h là chiếm ưu thế (timeSpend = 3) và cũng có phần khách hàng ngồi từ 1 đến 2 giờ (timeSpend = 1).

Từ hai điều ta nhận định trên thì ta thấy đường gần như hai biến này tương quan với nhau, vì những tỷ lệ đưa ra rất phù hợp với thực tế. Và để chính xác hơn về kết quả, ta sẽ tiến hành kiểm định hai biến ‘method’ và biến ‘timeSpend’ có tương quan với nhau không?

Ta sẽ kiểm định giả thiết của hai biến timeSpend và biến method với giả thiết:

- H0: hai biến không có tương quan với nhau



```
r, pvalue = stats.pearsonr(data['timeSpend'], data['method'])  
print(f'Hệ số tương quan = {r:.4f}, p_values = {pvalue:.4f}')
```

↪ Hệ số tương quan = -0.3520, p_values = 0.0001

Nhận Xét: Từ trị số p value rất nhỏ ta bác bỏ H0: hai biến không tương quan với nhau. Suy ra, hai biến này tương quan với nhau.

4.10. Biểu đồ thể hiện số lượng khách hàng của từng nhóm tuổi

- Số lượng khách hàng thuộc các nhóm tuổi:

Để xem xét nhóm tuổi nào có số lượng khách hàng của Starbucks nhiều nhất, ta dùng biểu đồ bong bóng (bubble chart) để thể hiện độ lớn của các nhóm tuổi.

Trước khi vẽ biểu đồ, ta tổng hợp số lượng các khách hàng ở mỗi nhóm tuổi thông qua hàm.groupby() ở cột ‘age’ và trả về kết quả là số lượng phần tử của từng nhóm.count(). Ta tiến hành sắp xếp lại dữ liệu ở cột count theo thứ tự tăng dần qua hàm.sort_values(inplace=True) để dễ dàng cho việc định hình độ lớn bong bóng trên đồ thị tương ứng với các điểm dữ liệu.

```

1 data_age = pd.DataFrame(index=[0,1,2,3])
2 data_gb1 = data.groupby ('age')['age'].count()
3 data_age['Age'] = data_gb1

1 data_age.sort_values(by = ['Age'], inplace = True)
2 display(data_age)

```

Age

3	7
0	10
2	17
1	79

Hình 32: Code Python tổng hợp số lượng khách hàng theo nhóm tuổi

Khi đã có số liệu, ta dùng biểu đồ scatter qua hàm go.Scatter trong thư viện Plotly.graph_objects. Các tham số cần khai báo gồm:

- x=data_age.index
- y=data_age['Age']
- mode='markers': sử dụng hình tròn để biểu diễn điểm dữ liệu
- Điều chỉnh kích thước, màu sắc cho các điểm dữ liệu: marker=dict(color=['rgb(93, 164, 214)', 'rgb(255, 144, 14)', 'rgb(44, 160, 101)', 'rgb(255, 65, 54)'], opacity=[1, 0.8, 0.6, 0.4], size=[40, 60, 80, 100])

```

import plotly.graph_objects as go

fig = go.Figure(data=[go.Scatter(
    x=data_age.index, y=data_age['Age'],
    mode='markers',
    marker=dict(
        color=['rgb(93, 164, 214)', 'rgb(255, 144, 14)',
               'rgb(44, 160, 101)', 'rgb(255, 65, 54)'],
        opacity=[1, 0.8, 0.6, 0.4],
        size=[40, 60, 80, 100]
    )
)])
fig.update_xaxes(title_text="Age Group")
fig.update_yaxes(title_text="Number of Customers")
fig.update_layout(title_text='Biểu đồ thể hiện số lượng khách hàng của từng nhóm tuổi', title_x=0.5
                  )
fig.show()

```

Hình 33: Code Python khai báo biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi



Biểu đồ 41: Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi

Age Group:

- 0: Below 20
- 1: From 20 to 29
- 2: From 30 to 39
- 3: 40 and above

Nhận Xét: Nhìn vào biểu đồ có thể thấy, phần lớn khách hàng đến cửa hàng nằm trong khoảng độ tuổi 20 đến 29 (79/113). Đây là độ tuổi lao động, thích khám phá và có nhu cầu cao về thể hiện giá trị bản thân, nên thương hiệu có danh tiếng là nơi thu hút họ đến và trải nghiệm. Trong khi đó, độ tuổi từ 40 trở lên chiếm số ít trong nhóm khách hàng, chỉ có 7/113 là thuộc nhóm tuổi này.

4.11. Biểu đồ thể hiện tỷ lệ mức độ ghé thăm cửa hàng theo từng nhóm khoảng cách

- Tỷ lệ mức độ ghé thăm cửa hàng theo từng nhóm khoảng cách:

Để biểu diễn tỷ lệ mức độ ghé thăm cửa hàng của khách hàng theo từng nhóm khoảng cách địa lý, nhóm sử dụng biểu đồ hình tròn ghép (nested pie) qua hàm.sunburst() trong thư viện Plotly Express. Các tham số cần thiết gồm:

- Bảng dữ liệu sẽ sử dụng: data
- Các lớp dữ liệu: path=["visitNo","location"] (thứ tự từ trong ra ngoài)

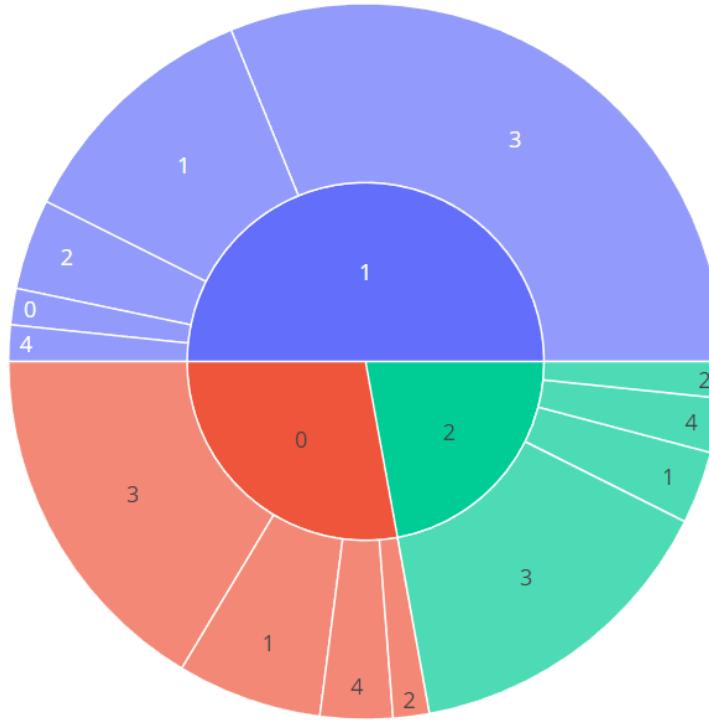
```

1 fig = px.sunburst(data,
2                     path=["location","visitNo"]
3                     )
4 fig.update_layout(title_text='Biểu đồ thể hiện tỷ lệ mức độ ghé thăm cửa hàng theo từng nhóm khoảng cách', title_x=0.5)
5 fig.show()

```

Hình 34: Khai báo biểu đồ Tỷ lệ mức độ ghé thăm cửa hàng theo từng nhóm khoảng cách

Biểu đồ thể hiện tỷ lệ mức độ ghé thăm cửa hàng theo từng nhóm khoảng cách



Biểu đồ 42: Biểu đồ thể hiện Tỷ lệ mức độ ghé thăm cửa hàng theo từng nhóm khoảng cách

Vòng tròn bên trong thể hiện tỷ lệ giữa các mức độ ghé thăm cửa hàng

- 0: Daily
- 1: Monthly
- 2: Never
- 3: Rarely
- 4: Weekly

Vòng tròn bên ngoài thể hiện tỷ lệ giữa các nhóm khoảng cách từ vị trí khách hàng đến cửa hàng gần nhất, tương ứng với từng mức độ ghé thăm cửa hàng:

- 0: Within 1km
- 1: 1km to 3km
- 2: More than 3km

Nhận Xét: Có thể thấy trong nhóm khách hàng tham gia khảo sát, hơn 50% khách hàng không thường xuyên ghé cửa hàng, họ chỉ đến hàng tháng - mức độ ghé cửa hàng ít nhất. Khi tìm hiểu về khoảng cách địa lý, yếu tố này không ảnh hưởng đến tần suất đến cửa hàng của nhóm những người chỉ ghé 1 lần/tháng này, vì tỷ lệ giữa 3 mức khoảng cách là như nhau.

Đối với nhóm khách hàng ghé cửa hàng 1 lần/tuần, hơn 1 nửa trong nhóm này cho biết khoảng cách từ vị trí của họ (nhà/nơi làm việc/...) đến cửa hàng gần nhất là trên 3km, 1 khoảng cách khá lớn và tồn tại nhiều thời gian di chuyển.

Là 1 cửa hàng nổi tiếng và có mức giá cao, vì thế tỷ lệ khách hàng trung thành ghé cửa hàng hằng ngày là không cao. Trong mẫu khảo sát này, chỉ có 9/113 người ghé cửa hàng mỗi ngày, phần lớn là những người ở gần cửa hàng (dưới 3km) (7/9 người thuộc nhóm này).

4.12. Các tiêu chí đánh giá theo thang điểm 1-5

Trước khi vẽ biểu đồ, ta tổng hợp số lượng đánh giá ở mỗi thang điểm với từng tiêu chí qua vòng lặp for cho từng cột tiêu chí đánh giá và hàm groupby() áp dụng trên từng cột, trả về kết quả là số lượng phần tử của từng nhóm.count(). Sau khi tổng hợp sẽ có một số ô bị thiếu dữ liệu, nên nhóm sẽ thay thế các chỗ NaN thành số 0 và chuyển đổi kiểu dữ liệu về dạng số nguyên.astype('Int64').

```

1 data_rate = pd.DataFrame(index=[1,2,3,4,5])
2 for i in range (col.index('productRate'), col.index('chooseRate')):
3     data_gb = data.groupby (by = [col[i]])[col[i]].count()
4     data_rate[col[i]] = data_gb
5 display(data_rate)

```

	productRate	priceRate	promoRate	ambianceRate	wifiRate	serviceRate
1	1	12	4	1	6	NaN
2	8	25	6	5	13	4.0
3	33	44	26	31	47	36.0
4	48	23	41	50	37	50.0
5	23	9	36	26	10	23.0

```
1 data_rate = data_rate.fillna(0)
```

```

1 col2 = list(data_rate.columns)
2 for i in col2:
3     if type(data_rate[i]) != 'int64':
4         data_rate[i] = data_rate[i].astype('int64')

```

```
1 data_rate.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 5 entries, 1 to 5
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  -- 
 0   productRate  5 non-null    int64  
 1   priceRate    5 non-null    int64  
 2   promoRate    5 non-null    int64  
 3   ambianceRate 5 non-null    int64  
 4   wifiRate     5 non-null    int64  
 5   serviceRate  5 non-null    int64  

```

Hình 35: Xử lý dữ liệu điểm đánh giá các tiêu chí

Với bảng dữ liệu này, nhóm sử dụng biểu đồ cột chồng (stacked bar plot) thông qua vòng lặp for cho các cột, hàm.add_trace(go.Bar()) trong thư viện Plotly. Trục hoành sẽ là thang điểm từ 1 đến 5, trục tung sẽ là số lượng các đánh giá tương ứng, màu sắc sẽ dùng để phân loại các tiêu chí.

```

import plotly.graph_objects as go
x = data_rate.index

fig = go.Figure()
for i in range (0,6):
    fig.add_trace(go.Bar(x=x, y = data_rate[col2[i]], name = col2[i]))

fig.update_layout(barmode='stack', title_text='Biểu đồ thể hiện số lượt đánh giá các tiêu chí theo từng thang điểm từ 1-5', title_x=0.5,
                  xaxis=dict(title='Rate point', titlefont_size=16, tickfont_size=14),
                  yaxis=dict(title='Number of votes',
                            titlefont_size=16,
                            tickfont_size=14))
)
fig.show()

```

Hình 36: Khai báo biểu đồ cột chồng cho điểm các tiêu chí đánh giá



Biểu đồ 43: Biểu đồ thể hiện số lượt đánh giá các tiêu chí theo thang điểm từ 1 đến 5

Nhận Xét: Là một cửa hàng lâu đời, chuyên nghiệp, thuộc phân khúc cao nên cửa hàng được kì vọng sẽ đem lại những trải nghiệm, dịch vụ tốt nhất cho khách hàng. Nhìn chung, cửa hàng nhận được các đánh giá tích cực, có thể thấy mức điểm 3-4 là phổ biến nhất.

4 là mức điểm được chọn nhiều nhất, trong đó tiêu chí dịch vụ (Service) và không khí tại cửa hàng (Ambiance) chiếm nhiều số vote nhất (50 lượt bình chọn). Thái độ niềm nở, chuyên nghiệp của nhân viên và cách bài trí, âm nhạc tại cửa hàng là những điều được cửa hàng chú trọng để làm hài lòng trải nghiệm tại cửa hàng của khách hàng.

Hơn nữa, yếu tố dịch vụ và không khí tại cửa hàng được xem là đặc trưng khi nhắc tới thương hiệu. Nhìn vào biểu đồ có thể thấy đánh giá điểm tốt (3-5 điểm) của yếu tố dịch vụ chiếm số lượng lớn so với các tiêu chí khác cùng thang điểm.

Tuy nhiên, vì là thương hiệu thuộc phân cấp cao nên giá thành của các sản phẩm sẽ cao hơn mặt bằng chung, có những sản phẩm rất đơn giản nhưng lại mắc hơn 2-3 lần giá bình thường. Do vậy, điểm đánh giá của tiêu chí giá chiếm số lượng lớn trong thang điểm 1, 2.

4.13. Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng giới tính:

Biên giới tính ‘gender’ được hiển thị trên trục hoành, đối với mỗi giới tính có 4 thanh tương ứng với 4 nhóm tuổi khách hàng. Biên ‘age’ mang tính liên tục nên màu sắc được sử dụng theo thang màu tuần tự.

Ở đây ta dùng hàm pandas.crosstab để tạo bảng chéo (một bảng 2 chiều) để tổng hợp số lượng khách hàng đối với từng giới tính. Cũng như phân tích mối quan hệ giữa giới tính và nhóm tuổi khách hàng.

Dữ liệu của ta có giới tính - “gender” như sau:

- 0: nữ - “female”
- 1: nam - “male”

Nhóm tuổi - “age” như sau:

- 0: Từ 40 trở lên
- 1: Dưới 20
- 2: Từ 20 đến 29 tuổi
- 3: Từ 30 đến 39 tuổi

```
myfield1 = data['gender']
myfield2 = data['age']
cross = pd.crosstab(myfield1,myfield2)
cross
```

age	0	1	2	3
gender				
0	1	7	52	5
1	6	6	33	12

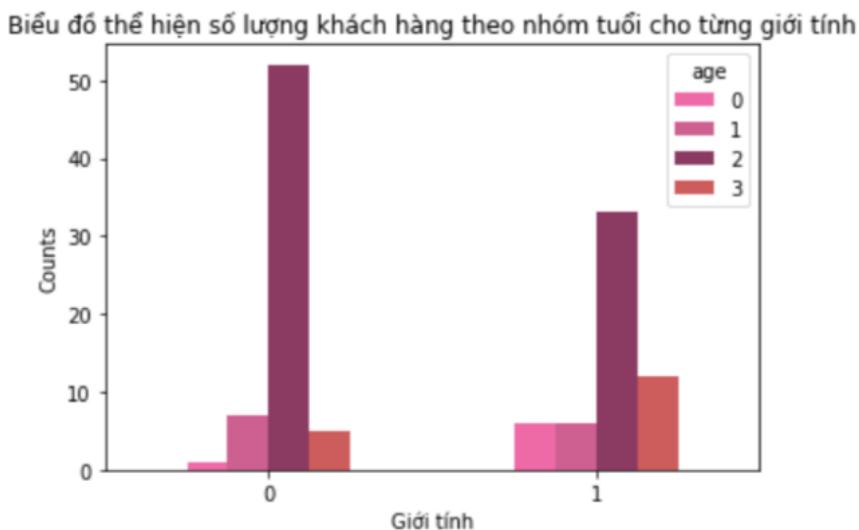
Hình 37: Code Python tạo bảng chéo để tổng hợp số lượng khách hàng đối với từng giới tính

Sau đó ta tạo biểu đồ bằng hàm datafram.plot.bar

```
barplot = cross.plot.bar(color=[(238/255,106/255,167/255), (205/255,96/255,144/255),(139/255,58/255,98/255),(205/255,92/255,92/255)],rot=0)
plt.title('Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng giới tính')
plt.xlabel('Giới tính')
plt.ylabel('Counts')
```

Hình 38: Code Python vẽ biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng giới tính

➤ *Ta thu được biểu đồ sau:*



Biểu đồ 44: Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng giới tính

Nhận Xét: Biểu đồ trên cho ta thấy rằng khách hàng theo giới tính nam và nữ đều tập trung phần lớn ở nhóm tuổi số 2 từ 20 đến 29 tuổi, đặc biệt là người nữ (52/122). Đây là đối tượng khách hàng chính mà các quán trà sữa/ cà phê thường hướng tới.

4.14. Biểu đồ thể hiện tỷ lệ về tần suất ghé thăm của khách

Sử dụng biểu đồ thể hiện tần suất ghé thăm của khách, diện tích mỗi phần tỷ lệ với giá trị mà nó biểu thị. Dữ liệu của ta có tần suất ghé thăm - “visitNo” của khách như sau:

- Số người ghé thăm mỗi ngày - “daily” là 2 người, ứng với 1.64%
- Số người ghé thăm mỗi tháng - “monthly” là 26 người, ứng với 21.31%
- Số người không bao giờ ghé thăm - “never” là 9 người, ứng với 7.38%
- Số người hiếm khi ghé thăm - “rarely” là 76 người, ứng với 62.3%
- Số người ghé thăm mỗi tuần - “weekly” là 9 người, ứng với 7.38%

```

visitNo = data['visitNo'].value_counts().sort_index()
print(visitNo, '\n')
visitNopct = data['visitNo'].value_counts()/sum(data['visitNo'].value_counts()) * 100
print('Tỷ lệ tần suất ghé thăm của khách')
print(visitNopct.sort_index())

0      2
1     26
2      9
3    76
4      9
Name: visitNo, dtype: int64

Tỷ lệ tần suất ghé thăm của khách
0      1.639344
1     21.311475
2      7.377049
3    62.295082
4      7.377049
Name: visitNo, dtype: float64

```

Hình 39: Code Python trình bày dữ liệu của ta có tần suất ghé thăm - “visitNo” của khách như sau

Biểu đồ tròn sẽ phù hợp nhất vì ta đang sử dụng 1 biến phân loại (tần suất ghé thăm - “visitNo”), với 5 nhóm:never, rarely, daily, weekly, monthly. Ta dùng hàm plot.pie từ thư viện pandas để tạo biểu đồ

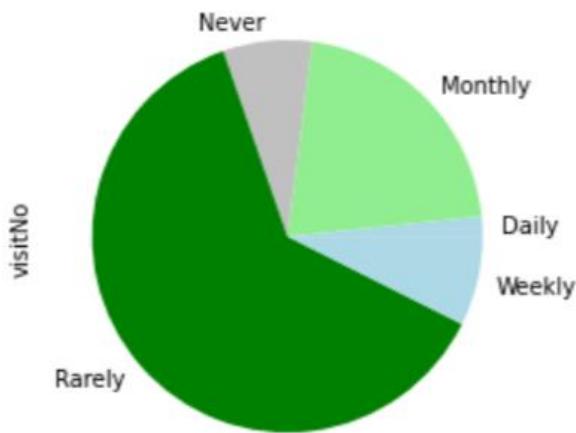
```

my_labels = ['Daily', 'Monthly', 'Never', 'Rarely', 'Weekly']
my_colors = ['lightblue','lightgreen',
             'silver','green']
visitNo.plot.pie (title="Biểu đồ thể hiện tỷ lệ về tần suất ghé thăm của khách ",y ='visitNo',
                  fontsize=10,labels=my_labels, colors=my_colors)

```

➤ Ta thu được biểu đồ sau:

Biểu đồ thể hiện tỷ lệ về tần suất ghé thăm của khách



Biểu đồ 45: Biểu đồ thể hiện tỷ lệ về tần suất ghé thăm của khách

Nhận Xét: Nhìn chung tần suất khách hàng ghé thấp có thể vì đây là thương hiệu khá đắt. Đặc biệt với sản phẩm có nhu cầu cao (như trà sữa, cà phê) thì việc cạnh tranh khó khăn và tần suất khách hàng ghé thăm bị ảnh hưởng là điều dễ hiểu

4.15. Biểu đồ thể hiện đánh giá mức tiền theo từng nhóm khách hàng

Dùng hàm pandas.crosstab để tạo bảng chéo (một bảng 2 chiều) để tổng hợp số lượng khách hàng đối với nhóm khách. Cũng như phân tích mối quan hệ giữa nhóm khách và đánh giá của họ về giá tiền.

Dữ liệu của ta có nhóm khách hàng - “status” như sau:

- 0: Có việc làm - “Employed”
- 1: Nội trợ - “Housewife”
- 2: Tự kinh doanh - “Self-Employed”
- 3: Học sinh - “Student”

Đánh giá về giá tiền - “priceRate”:

- 1: Rất tệ - “Very bad”
- 4: Tuyệt vời - “excellent”

```
myfield1 = data['status']
myfield2 = data['priceRate']
cross = pd.crosstab(myfield1,myfield2)
```

```
cross
```

		priceRate	0	1	2	3	4
		status	0	1	2	3	4
status	0	8	14	21	15	3	
	1	0	1	0	1	0	
	2	1	2	9	2	3	
	3	5	10	18	6	3	

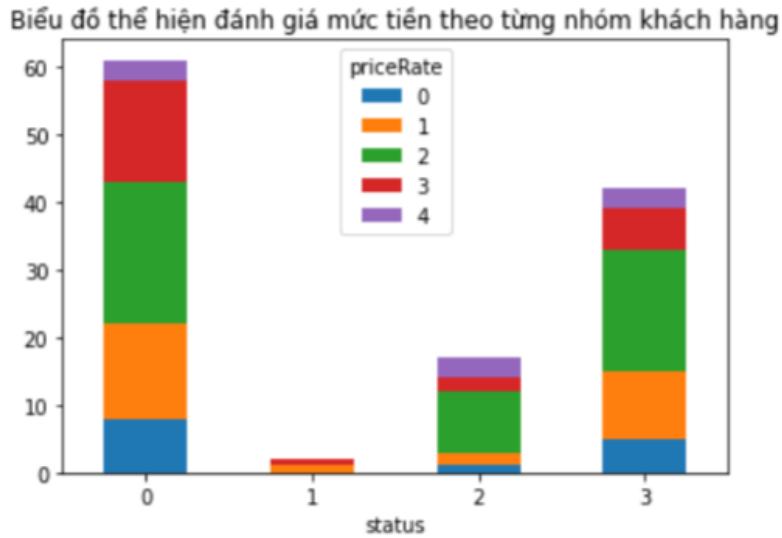
Hình 40: Tạo bảng chéo để tổng hợp số lượng khách hàng đối với nhóm khách

Tạo biểu đồ thanh chồng không chỉ biểu diễn trực quan mối quan hệ từng nhóm đối với tổng số khách hàng, mà còn so sánh riêng số lượng theo đánh giá giá tiền - “priceRate” trong từng nhóm khách. Ta thực hiện biểu đồ thanh chồng bằng hàm plot từ thư viện pandas

```
pl = cross.plot(kind="bar",title = "Biểu đồ thể hiện đánh giá mức tiền theo từng nhóm khách hàng", stacked =True , rot = 0)
```

Hình 41: Code Python vẽ biểu đồ thể hiện đánh giá mức tiền theo từng nhóm khách hàng

➤ Thu được biểu đồ sau:



Biểu đồ 46: Biểu đồ thể hiện đánh giá mức tiền theo từng nhóm khách hàng

Nhận Xét: Nhìn chung khách hàng chủ yếu là đối tượng "Employed" - những người có việc làm và có thu nhập ổn định để sử dụng loại đồ uống khá đắt này còn đối tượng nội trợ chiếm số lượng ít nhất. Ở các nhóm khách hàng ta đều thấy được họ có đánh giá mức trung bình cho thương hiệu trà sữa này, cụ thể mức điểm 2 nhận 48/122 lượt đánh giá (gần 39%)

4.16. Biểu đồ thể hiện tỷ lệ thời gian của từng khách hàng ở lại Starbucks

Hiện nay, việc ngồi lại ở các cửa hàng coffee, trà sữa,...hàng giờ đồng hồ đã là hình ảnh khá quen thuộc với người dân Việt Nam, đặc biệt là những sinh viên, người có việc làm,... Thế nhưng liệu cửa hàng nào cũng được như vậy hay không? Chúng ta sẽ tìm hiểu điều này thông qua biểu đồ hình tròn dưới đây.

Ta dùng thư viện `plotly.graph_objects` và dùng hàm `Pie` để vẽ biểu đồ.

```
[68] import plotly.graph_objects as go
df = data['timeSpend'].value_counts()
df = pd.DataFrame({'Time':df.index, 'Counts':df.values})

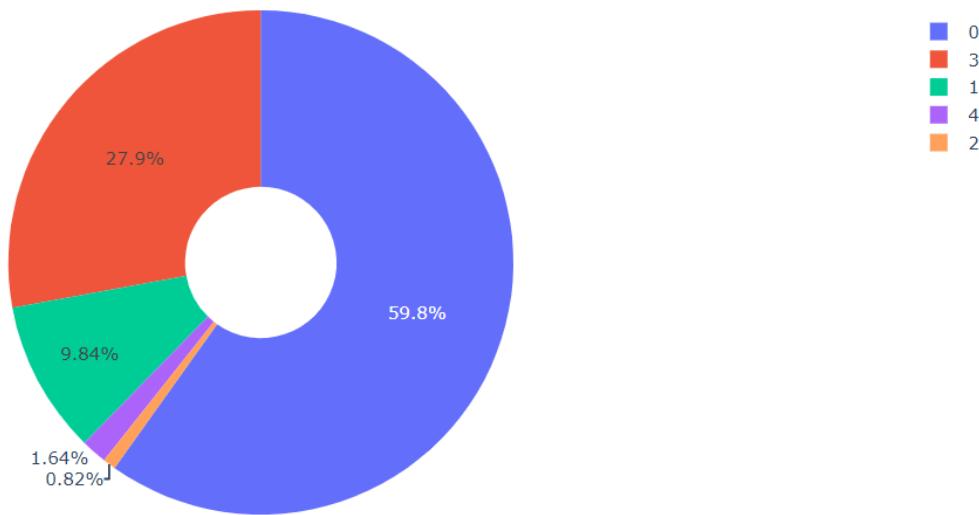
labels = df['Time']
values = df['Counts']

fig = go.Figure(data = [go.Pie(labels = labels, values = values, hole = 0.3)])
fig.update_layout(title_text = 'Biểu đồ tỷ lệ thời gian của từng khách hàng ở lại Starbucks', title_x = 0.5)
fig.show();
```

Hình 42: Code Python vẽ biểu đồ thể hiện tỷ lệ thời gian của từng khách hàng ở lại Starbucks

➤ **Kết quả thu được:**

Biểu đồ tỷ lệ thời gian của từng khách hàng ở Starbucks



Biểu đồ 47: Biểu đồ thể hiện tỷ lệ thời gian của từng khách hàng ở lại Starbucks

TimeSpend:

- 0: Below 30 mins (dưới 30 phút)
- 1: 1h to 2h (từ 1 giờ đến 2 giờ)
- 2: 2h to 3h (từ 2 giờ đến 3 giờ)
- 3: 30 mins to 1h (từ 30 phút đến 1 giờ)
- 4: More than 3h (từ 3 giờ trở lên)

Nhận Xét: Qua biểu đồ trên ta thấy tỷ lệ khách hàng ở lại Starbucks uống nước lần lượt là:

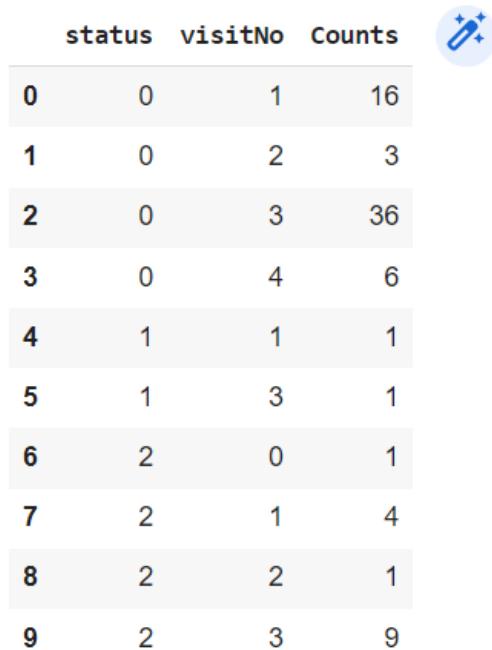
- Dưới 30 phút có tỷ lệ chiếm khoảng 59.8% → Chiếm tỷ trọng lớn nhất
- 1 giờ - 2 giờ có tỷ lệ chiếm khoảng 27.9%
- 2 giờ - 3 giờ có tỷ lệ chiếm khoảng 9.84%
- 30 phút - 1 giờ có tỷ lệ chiếm khoảng 1.64%
- 3 giờ trở lên có tỷ lệ chiếm khoảng 0.82% → Chiếm tỷ trọng nhỏ nhất

=> **Kết luận:** Phần đông khách hàng thường ở lại không quá 30 phút khi đến với Starbucks. Điều này cho thấy họ đến để mua rồi về chứ không ở lại thưởng thức không gian cửa hàng.

4.17. Biểu đồ thể hiện tần suất ghé thăm Starbucks của từng nhóm khách hàng

Sử dụng hàm groupby để tạo bảng bao gồm các cột “status” (nhóm khách hàng), “visitNo” (ghé thăm) và “Counts” (đếm tần số xuất hiện). Thực hiện những câu lệnh dưới đây ta sẽ thu được bảng tương tự.

```
[69] df1 = data.groupby(by = ['status','visitNo'])
g = df1.size()
df2 = g.reset_index(name = 'Counts')
df2.head(10)
```



	status	visitNo	Counts
0	0	1	16
1	0	2	3
2	0	3	36
3	0	4	6
4	1	1	1
5	1	3	1
6	2	0	1
7	2	1	4
8	2	2	1
9	2	3	9

Hình 43: Code Python tạo bảng bao gồm các cột “status” (nhóm khách hàng), “visitNo” (ghé thăm) và “Counts” (đếm tần số xuất hiện).

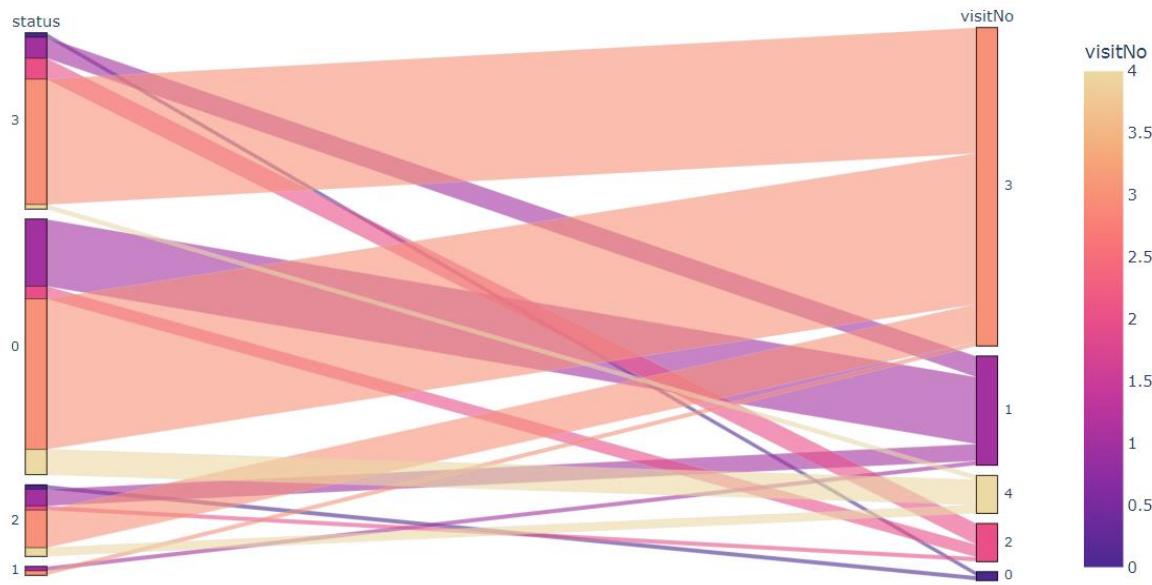
Dùng hàm parallel_categories để vẽ biểu đồ như yêu cầu với dữ liệu là data ban đầu và dimensions là “status” và “visitNo”.

```
▶ fig = px.parallel_categories(data,dimensions = ['status', 'visitNo'],
                                color = 'visitNo', color_continuous_scale = "agsunset",
                                title = 'Biểu đồ thể hiện tần suất ghé thăm Starbucks của từng nhóm khách hàng',
                                width = 1000, height = 600)
fig.show();
```

Hình 44: Code Python vẽ Biểu đồ thể hiện tần suất ghé thăm Starbucks của từng nhóm khách hàng

➤ Kết quả thu được:

Biểu đồ thể hiện tần suất ghé thăm Starbucks của từng nhóm khách hàng



Biểu đồ 48: Biểu đồ thể hiện tần suất ghé thăm Starbucks của từng nhóm khách hàng

Status:

- 0: Employed
- 1: Housewife
- 2: Self-employed
- 3: Student

VisitNo:

- 0: Daily
- 1: Monthly
- 2: Never
- 3: Rarely
- 4: Weekly

Nhận Xét:

- Đối với nhóm khách hàng 0 (Employed): Có 61 khách hàng đến cửa hàng mua thức uống. Phân lớn khoảng 59% khách hàng hiếm khi ghé thăm thương hiệu Starbucks, 26.2% khách hàng ghé thăm hàng tháng.
- Đối với nhóm khách hàng 1 (Housewife): Chỉ có 2 vị khách trong tổng số 122 khách mua thức uống tại cửa hàng. Trong đó 1 khách hiếm khi đến thăm Starbucks, còn 1 khách đến thăm hàng tháng

- Đối với nhóm khách hàng 2 (Self-employed): Có 9 khách hàng hiếm khi ghé thăm Starbucks (chiếm 56.25%), 4 khách ghé thăm hàng tháng (chiếm 25%)
- Đối với nhóm khách hàng 3 (Student): Có 41 khách hàng trong 122 khách đến với Starbucks thuộc nhóm này. Phần lớn khách hàng hiếm khi đến với cửa hàng để mua thức uống

=> **Kết luận:** Cần đánh mạnh phát triển vào phân khúc nhóm khách hàng Employed (có việc làm) bởi họ có thu nhập ổn định so với mức giá cao hơn thị trường thức uống của Starbucks. Họ sẵn sàng chi tiêu cho việc ăn uống tại thương hiệu nổi tiếng này.

4.18. Biểu đồ thể hiện mức độ chi trả của khách hàng đến Starbucks theo từng nhóm tuổi

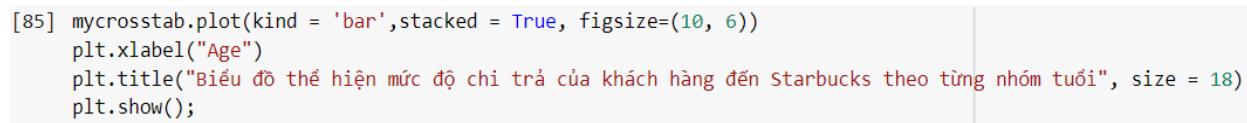
Tạo một bảng chéo crosstab bao gồm các cột “age”, “spendPurchase”. Ta thu được kết quả như bên dưới.

```
[71] mycol1 = data['age']
    mycol2 = data['spendPurchase']
    mycrosstab = pd.crosstab(mycol1,mycol2)
    mycrosstab = mycrosstab.rename(index = {0:'40 and above',1:'Below 20',2:'From 20 to 29',3:'From 30 to 39'})
    mycrosstab
```

spendPurchase	0	1	2	3
age				
40 and above	4	1	2	0
Below 20	2	9	0	2
From 20 to 29	29	42	4	10
From 30 to 39	10	6	1	0

Hình 45: Code Python tạo một bảng chéo crosstab bao gồm các cột “age”, “spendPurchase”.

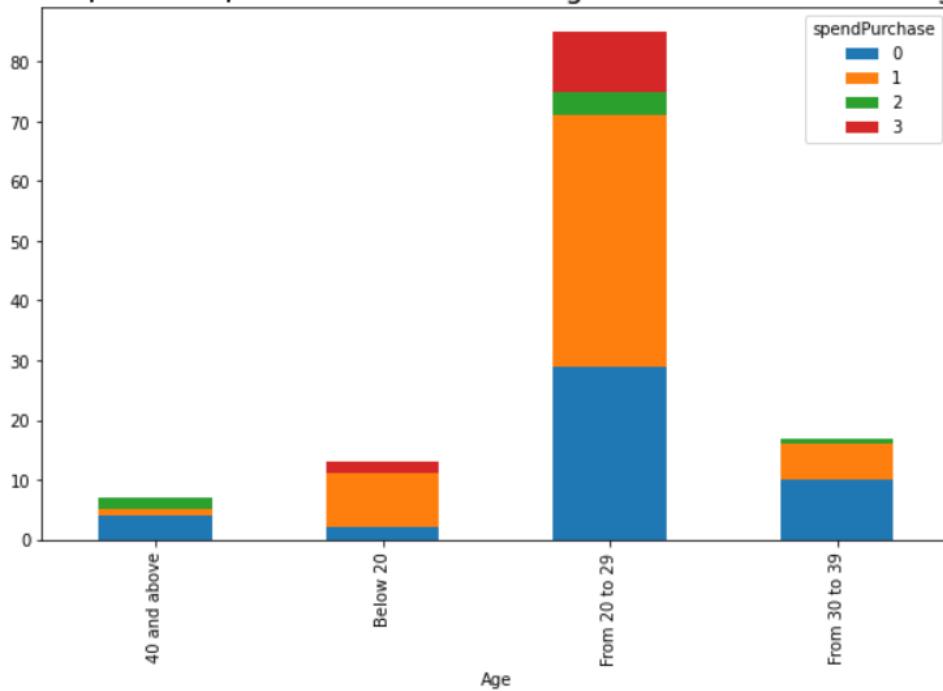
Sử dụng hàm bar kết hợp cùng bảng crosstab đã tạo để vẽ biểu đồ.



Hình 46: Code Python vẽ biểu đồ thể hiện mức độ chi trả của khách hàng đến Starbucks theo từng nhóm tuổi

➤ **Kết quả thu được:**

Biểu đồ thể hiện mức độ chi trả của khách hàng đến Starbucks theo từng nhóm tuổi



Biểu đồ 49: Biểu đồ thể hiện mức độ chi trả của khách hàng đến Starbucks theo từng nhóm tuổi

SpendPurchase:

- 0: Zero
- 1: Less than RM20
- 2: RM20 to RM40
- 3: More than RM40

Nhận Xét:

- Đối với độ tuổi dưới 20 (Below 20): độ tuổi này kinh tế chưa ổn định nên mức chi trả đa phần là ở mức điểm 1 (dưới 20RM)
- Đối với độ tuổi từ 20 - 29 (From 20 to 29): độ tuổi thuộc nhóm tuổi lao động, đây cũng là nhóm đối tượng khách hàng tiềm năng mà thương hiệu nhắm đến. Có thể thấy khoảng 34.1% khách hàng chịu chi tiêu cho bữa ăn uống của mình từ 20RM - 40RM (mức điểm 2), 49.4% khách chi trả dưới 20RM (mức điểm 1). Đặc biệt có xuất hiện các khách hàng chịu chi trả hơn 40RM (mức điểm 3).
- Đối với độ tuổi từ 30 - 39 (From 30 to 39): độ tuổi này dành khá ít thời gian lui tới các cửa hàng coffee, trà sữa,...vì họ còn phải chăm lo cho gia đình. Điều này có thể thấy chỉ có khoảng 6 khách hàng đến với cửa hàng chi trả ở mức dưới 20RM (mức điểm 1). Còn lại phần lớn là không đến mua.

- Đối với độ tuổi 40 trở lên: độ tuổi này là độ tuổi có phân khúc khách hàng thấp nhất trong 4 nhóm khách hàng. Họ không chịu bỏ tiền ra chi trả cho các buổi coffee ở thương hiệu này.

CHƯƠNG 5. GIẢM CHIỀU DŨ LIỆU

5.1. Phân phối chuẩn, One-Way ANOVA

5.1.1 Kiểm định điều kiện phân phối chuẩn tất cả các biến (Shapiro):

☞ Các giả thuyết kiểm định SHAPIRO

H0: gender, age, status, income, visitNo, method, timeSpend, location, membershipCard, itemPurchaseCoffee, itemPurchaseCold, itemPurchasePastries, itemPurchaseJuices, itemPurchaseSandwiches, spendPurchase, productRate, priceRate, promoRate, ambianceRate, wifiRate, serviceRate, chooseRate, promoMethodApp, promoMethodSoc, promoMethodEmail, promoMethodDeal, promoMethodFriend, promoMethodDisplay, promoMethodBillboard, loyal ~ Norm(Mu, Sigma)

Ha: gender, age, status, income, visitNo, method, timeSpend, location, membershipCard, itemPurchaseCoffee, itemPurchaseCold, itemPurchasePastries, itemPurchaseJuices, itemPurchaseSandwiches, spendPurchase, productRate, priceRate, promoRate, ambianceRate, wifiRate, serviceRate, chooseRate, promoMethodApp, promoMethodSoc, promoMethodEmail, promoMethodDeal, promoMethodFriend, promoMethodDisplay, promoMethodBillboard, loyal **KHÔNG** phải phân phối chuẩn

☞ Chuyển dạng dữ liệu:

Theo quy định của statsmodels: tên cột là feature, không phải là giá trị nên chuyển dạng dữ liệu về Cơ sở dữ liệu. Vậy ta dùng hàm melt của thư viện pandas để xoay khung dữ liệu (dataframe) để các cột được liệt kê thành hàng. Trong đó “id_vars” được cho là biến định danh, ta đặt tên là "index". Còn lại 30 cột được coi là biến đo lường, không được xoay vào trực hàng mà đưa vào danh sách “value_vars”.

```
df_melt = pd.melt(data.reset_index(), id_vars = ['index'],
                   value_vars = ['gender','age','status','income', 'visitNo', 'method', 'timeSpend', 'location',
                                 'membershipCard', 'itemPurchaseCoffee', 'itemPurchaseCold', 'itemPurchasePastries', 'itemPurchaseJuices',
                                 'itemPurchaseSandwiches', 'spendPurchase', 'productRate', 'priceRate', 'promoRate', 'ambianceRate', 'wifiRate',
                                 'serviceRate', 'chooseRate', 'promoMethodApp', 'promoMethodSoc', 'promoMethodEmail', 'promoMethodDeal', 'promoMethodFriend',
                                 'promoMethodDisplay', 'promoMethodBillboard', 'loyal'])
```

Hình 47: Code Python chuyển dạng dữ liệu

Ta có 30 features của 122 đối tượng nên sau khi chuyển dạng dữ liệu ta thu được danh sách 3660 dòng:

	index	variable	value
0	0	gender	0
1	1	gender	0
2	2	gender	1
3	3	gender	0
4	4	gender	1
...
3655	117	loyal	1
3656	118	loyal	1
3657	119	loyal	0
3658	120	loyal	1
3659	121	loyal	0

[3660 rows x 3 columns]

☞ *Kiểm định Shapiro dựa trên Ordinary Least Squares (OLS) model:*

Ta đặt tên cho các cột của bảng đã được chuyển dạng Cơ sở dữ liệu: Cột "index" chứa chỉ số các đối tượng, cột "features" chứa tên thuộc tính và "value" chứa các giá trị.

Tiếp theo sử dụng hàm OLS từ thư viện statsmodels để thực hiện hồi quy OLS bằng cách lần lượt truyền vào biến phụ thuộc và biến độc lập. Đồng thời tích hợp gọi phương thức fit để khớp đường hồi quy với dữ liệu

Cuối cùng thực hiện kiểm định xem 30 features đã cho có thuộc phân phối chuẩn hay không bằng kiểm định Shapiro-Wilk. Ta gọi hàm shapiro() từ thư viện scipy để kiểm tra.

```
df_melt.columns = ['index', 'features', 'value']
model = ols('value ~ C(features)', data = df_melt).fit()
shapiro, p = st.shapiro(model.resid)
print(f'Tri thong ke Shapiro = {shapiro:.4f}, tri so p = {p:.4f}')
```

Hình 48: Code Python kiểm định Shapiro dựa trên Ordinary Least Squares (OLS) model

➤ *Kết quả như sau:*

Tri thong ke Shapiro = 0.949101, tri so p = 0.000000

Hình 49: Kết quả Kiểm định Shapiro dựa trên Ordinary Least Squares (OLS) model

Nhận Xét: Trị số $p = 0 < \alpha = 0.05$ cho nên ta bác bỏ giả thuyết H_0 . Như vậy tất cả 18 biến không phải phân phối chuẩn

5.1.2 One-way ANOVA:

Đối với One-way ANOVA, đây là một kiểm định xem liệu có sự khác biệt nào giữa các biến hay không. Vì vậy giả thuyết được đưa ra ở đây là:

- H_0 : Không có sự khác biệt nào giữa các biến.
- H_1 : Có ít nhất một biến khác với các biến còn lại (có sự khác biệt giữa các biến)

Đầu tiên, ta tiến hành tạo các mảng dữ liệu tương ứng với các biến bằng thư viện `scipy.stats` và hàm `f_oneway` để có thể xác định trị số p (được ước tính từ trị số thống kê F và mức độ tự do). Sau khi có được trị số F , ta bắt đầu so sánh với $\alpha = 0.05$ xem liệu có thể bác bỏ H_0 hay không.

Tiếp theo, tạo bảng ANOVA để biểu thị các nguồn sai số và mức độ tự do liên quan của chúng.

```
alpha = .05
## F-statistic và p-value
f, p = stats.f_oneway(data.gender, data.age, data.status, data.income, data.visitNo, data.method, data.timeSpend, data.location,
                      data.membershipCard, data.itemPurchaseCoffee, data.itemPurchaseCold, data.itemPurchasePastries, data.itemPurchaseJuices,
                      data.itemPurchaseSandwiches, data.spendPurchase, data.productRate, data.priceRate, data.promoRate, data.ambianceRate, data.wifiRate,
                      data.serviceRate, data.chooseRate, data.promoMethodApp, data.promoMethodSoc, data.promoMethodEmail, data.promoMethodDeal, data.promoMethodFriend,
                      data.promoMethodDisplay, data.promoMethodBillboard, data.loyal)
print(f'* Trị số p = {p:.2f}, alpha = {alpha:.2f}*)

if (p < alpha):
    print('* Trị số p < alpha cho nên bác bỏ H0 ==> CÓ sự khác biệt giữa các biến')
else:
    print('* KHÔNG bác bỏ H0 ==> KHÔNG có sự khác biệt giữa các biến')

## Tạo ANOVA table
anova_table = sm.stats.anova_lm(model, typ = 2)
print('\n* ANOVA table\n', anova_table)

anova_table = sm.stats.anova_lm(model, typ = 1) # có thêm cột mean_squared
print('\n* ANOVA table\n', anova_table)
```

Hình 50: Code Python tạo bảng ANOVA để biểu thị các nguồn sai số và mức độ tự do liên quan của các biến

➤ *Ta thu được kết quả:*

```

* Trị số p = 0.00, alpha = 0.05
* Trị số p < alpha cho nên bác bỏ H0 ==> CÓ sự khác biệt giữa các biến

* ANOVA table
      sum_sq      df          F   PR(>F)
C(features) 3724.962022  29.0 145.200092  0.0
Residual     3211.172131 3630.0        NaN      NaN

* ANOVA table
      df      sum_sq      mean_sq          F   PR(>F)
C(features) 29.0 3724.962022 128.446966 145.200092  0.0
Residual    3630.0 3211.172131  0.884620        NaN      NaN

```

Bảng 9: Bảng ANOVA biểu thị các nguồn sai số và mức độ tự do liên quan của các biến

Trị số p ở đây = 0 và nhỏ hơn alpha cho nên ta bác bỏ H0 và ghi nhận rằng Có sự khác biệt giữa các biến. Bảng ANOVA table ở đây sẽ cho ta thấy rõ các kết quả của các tính toán liên quan đến tổng bình phương, bậc tự do và giá trị thống kê F.

5.1.3. Hậu kiểm Tukey HSD:

Việc áp dụng One-way ANOVA cho ta biết có sự khác biệt nào đó xảy ra giữa các biến, tuy nhiên lại không ta biết cụ thể những nhóm biến độc lập nào đang khác nhau. Để tìm ra chính xác những nhóm nào khác nhau, chúng ta phải tiến hành phân tích sâu bằng "post hoc test". Một trong những kiểm tra được sử dụng phổ biến nhất là hậu kiểm Tukey HSD. Phương pháp này giúp ta so sánh theo cặp các mẫu độc lập (30 biến độc lập - tạo thành 435 cặp mẫu). Ta sẽ dùng hàm pairwise_tukeyhsd từ thư viện stastmodels.

```

m_comp = pairwise_tukeyhsd(endog = df_melt['value'], groups = df_melt['features'], alpha = 0.05)
print(m_comp)
#dòng nào true thì cặp đó khác, false thì giống

```

Hình 51: Code Python hậu kiểm Tukey HSD

➤ Kết quả khi ta cắt ra 1 phần từ bảng so sánh:

Multiple Comparison of Means - Tukey HSD, FWER=0.05							
group1	group2	meandiff	p-adj	lower	upper	reject	
age	ambianceRate	0.8361	0.001	0.3843	1.2879	True	
age	chooseRate	0.5984	0.001	0.1466	1.0502	True	
age	gender	-1.4508	0.001	-1.9026	-0.999	True	
age	income	-0.6475	0.001	-1.0993	-0.1957	True	
age	itemPurchaseCoffee	-1.2295	0.001	-1.6813	-0.7777	True	
age	itemPurchaseJuices	-1.7869	0.001	-2.2387	-1.3351	True	
age	itemPurchasePastries	-1.8852	0.001	-2.337	-1.4335	True	
age	itemPurchaseSandwiches	-1.8525	0.001	-2.3043	-1.4007	True	
age	itemPurchaseCold	-1.6066	0.001	-2.0584	-1.1548	True	
age	location	-0.9754	0.001	-1.4272	-0.5236	True	
age	loyal	-1.1475	0.001	-1.5993	-0.6957	True	
age	membershipCard	-1.4262	0.001	-1.878	-0.9744	True	
age	method	0.9508	0.001	0.499	1.4026	True	
age	priceRate	-0.0246	0.9	-0.4764	0.4272	False	
age	productRate	0.7459	0.001	0.2941	1.1977	True	
age	promoMethodApp	-1.6311	0.001	-2.0829	-1.1794	True	
age	promoMethodBillboard	-1.8279	0.001	-2.2797	-1.3761	True	
age	promoMethodDeal	-1.8607	0.001	-2.3125	-1.4089	True	
age	promoMethodDisplay	-1.7459	0.001	-2.1977	-1.2941	True	
age	promoMethodEmail	-1.7705	0.001	-2.2223	-1.3187	True	
age	promoMethodFriend	-1.5164	0.001	-1.9682	-1.0646	True	
age	promoMethodSoc	-1.1885	0.001	-1.6403	-0.7367	True	
age	promoRate	0.877	0.001	0.4253	1.3288	True	
age	serviceRate	0.8279	0.001	0.3761	1.2797	True	
age	spendPurchase	-1.0328	0.001	-1.4846	-0.581	True	
age	status	-0.5902	0.001	-1.042	-0.1384	True	
age	timeSpend	-0.9016	0.001	-1.3534	-0.4498	True	
age	visitNo	0.6066	0.001	0.1548	1.0584	True	
age	wifiRate	0.3361	0.5438	-0.1157	0.7879	False	

Bảng 10: Bảng so sánh các cặp mẫu bằng phương pháp Tukey HSD

Nhận Xét: Có sự khác biệt về ý nghĩa thống kê của 305 cặp mẫu nhưng không có sự khác biệt về ý nghĩa thống kê của 130 cặp mẫu

5.2. Chi-squared Test

- + **Các cột cần kiểm định là:** “gender”, “age”, “status”, “income”, “visitNo”, “method”, “timeSpend”, “location”, “membershipCard”, “spendPurchase”, “productRate”, “priceRate”, “promoRate”, “ambianceRate”, “wifiRate”, “serviceRate”, “chooseRate”.

Xây dựng một biến chứa list các columns cần kiểm định.

```
[ ] from scipy.stats import chi2_contingency

[ ] col = list(data.columns)
col_chi = col[col.index('gender'):col.index('membershipCard')+1] +(col[col.index('spendPurchase'):col.index('chooseRate')+1])

[ ] col_chi
['gender',
'age',
'status',
'income',
'veisitNo',
'method',
'timeSpend',
'location',
'membershipCard',
'spendPurchase',
'productRate',
'priceRate',
'promoRate',
'ambianceRate',
'wifiRate',
'serviceRate',
'chooseRate']
```

Hình 52: Code Python xây dựng một biến chứa list các columns cần kiểm định.

Tạo 1 DataFrame chứa các cột “Feature 1”, “Feature 2” và “p-value”.

Trong đó:

- Cột Feature 1 và Feature 2: cột chứa 2 biến phân loại mà ta cần kiểm định
- Cột p-value: chứa giá trị p-value tính được dùng để so sánh với alpha để bác bỏ H0 hay chấp nhận H0

```
[ ] chi_sqr_result = pd.DataFrame(columns=[ 'Feature 1', 'Feature 2', 'p-value'])
```

Hình 53: Code Python tạo 1 DataFrame chứa các cột “Feature 1”, “Feature 2” và “p-value”.

Để có thể bác bỏ H0 thì cần so sánh giá trị giữa p-value và alpha đã cho. Theo lý thuyết, số alpha được gọi là mức ý nghĩa (là ngưỡng được chọn để quyết định ý nghĩa), còn $(1 - \alpha)$ gọi là độ tin cậy.

Tạo biến $\alpha = 0.05$ và biến $confidence_level = (1 - \alpha)$.

```
[ ] alpha = .05
confidence_level = (1 - alpha)
```

Hình 54: Code Python tạo biến $\alpha = 0.05$ và biến $confidence_level = (1 - \alpha)$.

Ta tạo vòng lặp for để tìm giá trị p-value giữa các biến phân loại.

```
[ ] for i in range (0, len(col_chi)-1):
    for j in range (i+1, len(col_chi)):
        table = pd.crosstab(data[col_chi[i]], data[col_chi[j]]) #tạo bảng tương quan giữa 2 biến phân loại
        stat, p, dof, expected = chi2_contingency(table) #kiểm định chi-squared
        chi_sqr_result.loc[len(chi_sqr_result)] = [col_chi[i], col_chi[j], round(p, 3)] #thêm kết quả p-value tương đương với cặp biến tương quan
```

Hình 55: Code Python tạo vòng lặp for để tìm giá trị p-value giữa các biến phân loại

Đối với vòng lặp for đầu tiên dùng để tìm biến “Feature 1”. Vòng lặp for thứ 2 dùng để tìm biến “Feature 2”.

Tiếp theo dùng hàm crosstab tạo “table” - bảng tương quan giữa 2 biến phân loại tìm được trong vòng lặp for.

Sử dụng hàm chi2_contingency có sẵn trong thư viện `scipy.stats` để thực hiện các kiểm định Chi-Squared với bộ dữ liệu là bảng `table` vừa tạo. Sau khi kiểm định xong lưu các kết quả vào biến “`chi_sqr_result`”.

Tạo cột “*Test Result*” để hiện thị các kết quả sau khi lấy **p-value** so sánh với `alpha`.

- Nếu `p-value < alpha` → bác bỏ H_0
- Ngược lại → chưa đủ cơ sở để bác bỏ H_0

➤ Kết quả thu được:

```
[ ] chi_sqr_result['Test Result'] = chi_sqr_result.apply(lambda x: 'bác bỏ H0' if x['p-value'] < alpha else 'Chưa đủ cơ sở để bác bỏ H0', axis=1)

[ ] chi_sqr_result
```

	Feature 1	Feature 2	p-value	Test Result
0	gender	age	0.016	bác bỏ H0
1	gender	status	0.238	Chưa đủ cơ sở để bác bỏ H0
2	gender	income	0.009	bác bỏ H0
3	gender	visitNo	0.160	Chưa đủ cơ sở để bác bỏ H0
4	gender	method	0.231	Chưa đủ cơ sở để bác bỏ H0
...
131	ambianceRate	serviceRate	0.000	bác bỏ H0
132	ambianceRate	chooseRate	0.000	bác bỏ H0
133	wifiRate	serviceRate	0.000	bác bỏ H0
134	wifiRate	chooseRate	0.006	bác bỏ H0
135	serviceRate	chooseRate	0.000	bác bỏ H0

136 rows × 4 columns

Hình 56: Các kết quả sau khi lấy p-value so sánh với `alpha`.

```
chi_sqr_result[chi_sqr_result['Test Result']=='bắc bỏ H0']
```

	Feature 1	Feature 2	p-value	Test Result
0	gender	age	0.016	bắc bỏ H0
2	gender	income	0.009	bắc bỏ H0
5	gender	timeSpend	0.015	bắc bỏ H0
16	age	status	0.000	bắc bỏ H0
17	age	income	0.000	bắc bỏ H0
23	age	spendPurchase	0.028	bắc bỏ H0
31	status	income	0.000	bắc bỏ H0
36	status	membershipCard	0.012	bắc bỏ H0
37	status	spendPurchase	0.025	bắc bỏ H0
47	income	timeSpend	0.010	bắc bỏ H0
49	income	membershipCard	0.002	bắc bỏ H0
50	income	spendPurchase	0.014	bắc bỏ H0
56	income	serviceRate	0.011	bắc bỏ H0
58	visitNo	method	0.000	bắc bỏ H0
61	visitNo	membershipCard	0.001	bắc bỏ H0
62	visitNo	spendPurchase	0.000	bắc bỏ H0
63	visitNo	productRate	0.001	bắc bỏ H0

68	visitNo	serviceRate	0.040	bác bỏ H0
73	method	spendPurchase	0.000	bác bỏ H0
74	method	productRate	0.000	bác bỏ H0
76	method	promoRate	0.000	bác bỏ H0
77	method	ambianceRate	0.000	bác bỏ H0
79	method	serviceRate	0.000	bác bỏ H0
80	method	chooseRate	0.019	bác bỏ H0
88	timeSpend	wifiRate	0.049	bác bỏ H0
91	location	membershipCard	0.001	bác bỏ H0
100	membershipCard	spendPurchase	0.003	bác bỏ H0
101	membershipCard	productRate	0.008	bác bỏ H0
104	membershipCard	ambianceRate	0.028	bác bỏ H0
106	membershipCard	serviceRate	0.042	bác bỏ H0
108	spendPurchase	productRate	0.017	bác bỏ H0
109	spendPurchase	priceRate	0.000	bác bỏ H0
111	spendPurchase	ambianceRate	0.018	bác bỏ H0
113	spendPurchase	serviceRate	0.032	bác bỏ H0
114	spendPurchase	chooseRate	0.029	bác bỏ H0
115	productRate	priceRate	0.000	bác bỏ H0

116	productRate	promoRate	0.000	bác bỏ H0
117	productRate	ambianceRate	0.000	bác bỏ H0
118	productRate	wifiRate	0.003	bác bỏ H0
119	productRate	serviceRate	0.000	bác bỏ H0
120	productRate	chooseRate	0.000	bác bỏ H0
121	priceRate	promoRate	0.018	bác bỏ H0
122	priceRate	ambianceRate	0.000	bác bỏ H0
123	priceRate	wifiRate	0.032	bác bỏ H0
124	priceRate	serviceRate	0.001	bác bỏ H0
125	priceRate	chooseRate	0.000	bác bỏ H0
126	promoRate	ambianceRate	0.000	bác bỏ H0
127	promoRate	wifiRate	0.025	bác bỏ H0
128	promoRate	serviceRate	0.000	bác bỏ H0
129	promoRate	chooseRate	0.017	bác bỏ H0
130	ambianceRate	wifiRate	0.000	bác bỏ H0
131	ambianceRate	serviceRate	0.000	bác bỏ H0
132	ambianceRate	chooseRate	0.000	bác bỏ H0
133	wifiRate	serviceRate	0.000	bác bỏ H0
134	wifiRate	chooseRate	0.006	bác bỏ H0
135	serviceRate	chooseRate	0.000	bác bỏ H0

```
[ ] chi_sqr_result[chi_sqr_result['Test Result'] !='bác bỏ H0']
```

	Feature 1	Feature 2	p-value	Test Result
1	gender	status	0.238	Chưa đủ cơ sở để bác bỏ H0
3	gender	visitNo	0.160	Chưa đủ cơ sở để bác bỏ H0
4	gender	method	0.231	Chưa đủ cơ sở để bác bỏ H0
6	gender	location	0.384	Chưa đủ cơ sở để bác bỏ H0
7	gender	membershipCard	0.594	Chưa đủ cơ sở để bác bỏ H0
...
103	membershipCard	promoRate	0.800	Chưa đủ cơ sở để bác bỏ H0
105	membershipCard	wifiRate	0.273	Chưa đủ cơ sở để bác bỏ H0
107	membershipCard	chooseRate	0.172	Chưa đủ cơ sở để bác bỏ H0
110	spendPurchase	promoRate	0.281	Chưa đủ cơ sở để bác bỏ H0
112	spendPurchase	wifiRate	0.584	Chưa đủ cơ sở để bác bỏ H0

80 rows × 4 columns

Nhận Xét: Từ bảng số liệu thu được sau khi thực hiện kiểm định Chi-squared ta có:

- Các dòng kiểm định cho ra kết quả là “bác bỏ H0” → Điều này chứng minh rằng 2 biến phân loại đó có mối liên kết với nhau (phụ thuộc nhau).
- Các dòng kiểm định cho ra kết quả là “chưa đủ cơ sở để bác bỏ H0” → Điều này chứng minh rằng 2 biến phân loại đó không có mối liên kết với nhau (độc lập nhau).
- ❖ Để giảm chiều dữ liệu ta có thể loại bỏ các cột có sự phụ thuộc nhiều với các cột khác. Ví dụ: biến “method” phụ thuộc với các biến “spendPurchase”, “productRate”, “promoRate”, “ambianceRate”, “serviceRate”, “chooseRate” (vì có p-value < alpha = 0.05)

method	spendPurchase	0.000	bác bỏ H0
method	productRate	0.000	bác bỏ H0
method	promoRate	0.000	bác bỏ H0
method	ambianceRate	0.000	bác bỏ H0
method	serviceRate	0.000	bác bỏ H0
method	chooseRate	0.019	bác bỏ H0

→ Ta có thể bỏ lược bỏ đi cột “method” để giảm số chiều dữ liệu mà không ảnh hưởng quá nhiều đến các phân tích dữ liệu khác.

==> Một số cột có thể suy nghĩ để lược bỏ bớt nhằm giảm chiều dữ liệu là: “productRate”, “method”, “priceRate”, “promoRate”,...

5.3. PCA

Đặt biến “loyal” làm target cho bộ dữ liệu, rồi lọc bỏ biến loyal khỏi bộ dữ liệu. Liệt kê số lượng các cột và tên các cột có trong bộ dữ liệu:

```
[ ] target = 'loyal'
print('* Biến phân lớp:', target)

# Danh sách các features
nb_features = data.shape[1] - 1
features = data.columns[:nb_features]
print('* Số lượng features = %d' %nb_features)
print(' Các features: ', ', '.join(features))

* Biến phân lớp: loyal
* Số lượng features = 18
Các features: Id, gender, age, status, income, visitNo, method, timeSpend, location, membershipCard, spendPurchase, productRate, priceRate, promoRate, ambianceRate, wifiRate, serviceRate
```

Loại bỏ biến target khỏi dữ liệu:

```
[ ] x = data.drop(columns = ['loyal'])
```

Hình 57: Code Python loại bỏ biến target khỏi dữ liệu

Cách 1:

Truyền bộ dữ liệu sau khi lọc vào mô hình PCA

```

pca = PCA().fit(x)

points = np.cumsum(pca.explained_variance_ratio_) * 100
points = np.insert(points, 0, 0) # Thêm điểm k = 0, variance = 0
x_i = np.arange(0, nb_features + 1)
y_i = (points)//0.01/100

```

Hình 58: Code Python truyền bộ dữ liệu sau khi lọc vào mô hình PCA

Tính toán phương sai tích lũy của từng số chiều dữ liệu. Ta được mảng y_i chứa phương sai tích lũy theo từng số chiều trong x_i

Biểu diễn từng cặp số (x_i; y_i)

```

## Vẽ đồ thị biểu diễn % phương sai tích lũy theo số features
##--> chọn k theo điểm "gãy"
import matplotlib.pyplot as plt

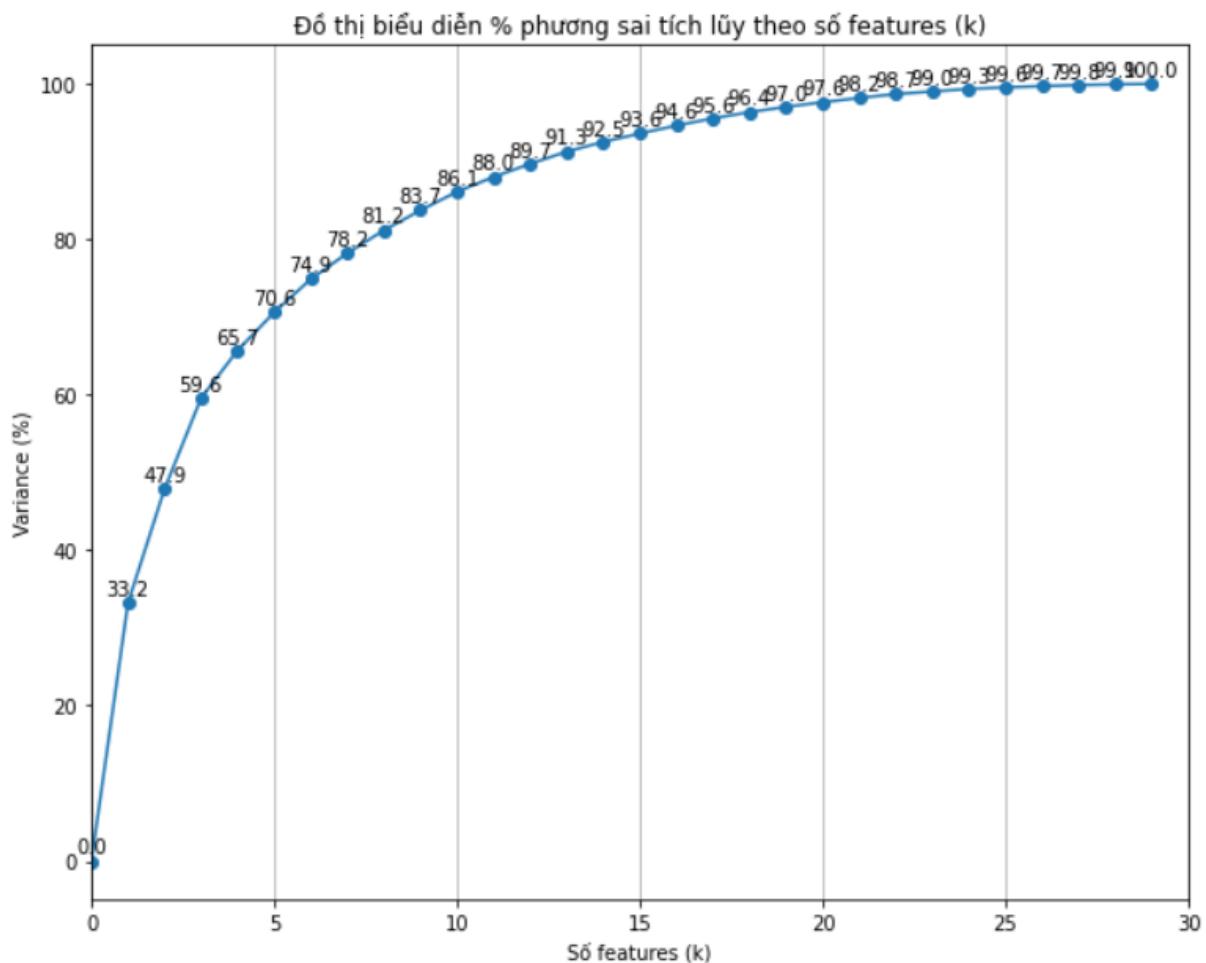
# Các điểm dữ liệu
points = np.cumsum(pca.explained_variance_ratio_) * 100
points = np.insert(points, 0, 0) # Thêm điểm k = 0, variance = 0
x_i = np.arange(0, nb_features + 1)
y_i = (points)//0.01/100

plt.figure(figsize = (10, 8))
plt.plot(points, marker = 'o')
plt.xlabel('Số features (k)')
plt.ylabel('Variance (%)')
plt.title('Đồ thị biểu diễn % phương sai tích lũy theo số features (k)')
plt.xlim([0, nb_features + 1])
plt.grid(axis = 'x')
for i in x_i:
    plt.text(i, y_i[i] + 1, round(y_i[i], 1), ha = 'center', va = 'baseline')
    # tung độ của text cao hơn point 1 đơn vị

plt.show()

```

Hình 59: Code Python vẽ đồ thị biểu diễn % phương sai tích lũy theo số features



Hình 60: Đồ thị biểu diễn % phương sai tích lũy theo số features

Nhận Xét: từ biểu đồ ta thấy tại k=17 đạt được phương sai tích lũy > 95% và khi tăng số chiều k % phương sai tích lũy tăng không nhiều

Cách 2:

```
[121] from sklearn.decomposition import PCA

## Giả sử muốn giữ lại tối thiểu 95% (đã định một ngưỡng)
threshold = .95
percent   = threshold * 100

## Áp dụng PCA với ngưỡng đã chọn
pca = PCA(threshold)
pca.fit_transform(x)

## Giá trị k thu được, với phương sai tích lũy tương ứng
k   = pca.n_components_
var = sum(pca.explained_variance_ratio_) * 100
print('  * Muốn phương sai tích lũy >= %.1f%%' %percent, 'thì k >= %d' %k, '--> %.1f%%' %var)

* Muốn phương sai tích lũy >= 95.0% thì k >= 17 --> 95.6%
```

Hình 61: Code Python tìm số k - cách 2

```
▶ ## Kiểm chứng: Phân tích chi tiết theo các ngưỡng phương sai từ 50% đến 99%
A = np.array([.5, .6, .7, .8, .9, .95, .99])
for t in A:
    percent = t * 100
    pca     = PCA(t)

    pca.fit(x)
    k   = pca.n_components_
    var = sum(pca.explained_variance_ratio_) * 100
    print('- Muốn phương sai tích lũy >= %.1f%%' %percent, 'thì k >= %2d' %k, '(var ~ %.1f%%)' %var)

→ - Muốn phương sai tích lũy >= 50.0% thì k >= 3 (var ~ 59.6%)
- Muốn phương sai tích lũy >= 60.0% thì k >= 4 (var ~ 65.7%)
- Muốn phương sai tích lũy >= 70.0% thì k >= 5 (var ~ 70.6%)
- Muốn phương sai tích lũy >= 80.0% thì k >= 8 (var ~ 81.2%)
- Muốn phương sai tích lũy >= 90.0% thì k >= 13 (var ~ 91.3%)
- Muốn phương sai tích lũy >= 95.0% thì k >= 17 (var ~ 95.6%)
- Muốn phương sai tích lũy >= 99.0% thì k >= 23 (var ~ 99.0%)
```

Hình 62: Code Python phân tích chi tiết theo các ngưỡng phương sai từ 50% đến 99%

→ Dựa vào 2 kết quả trên ta giảm chiều dữ liệu với k = 17

```

## Biểu diễn trực quan dữ liệu với k = 17
k    = 17
pca = PCA(k)
pca.fit(x)

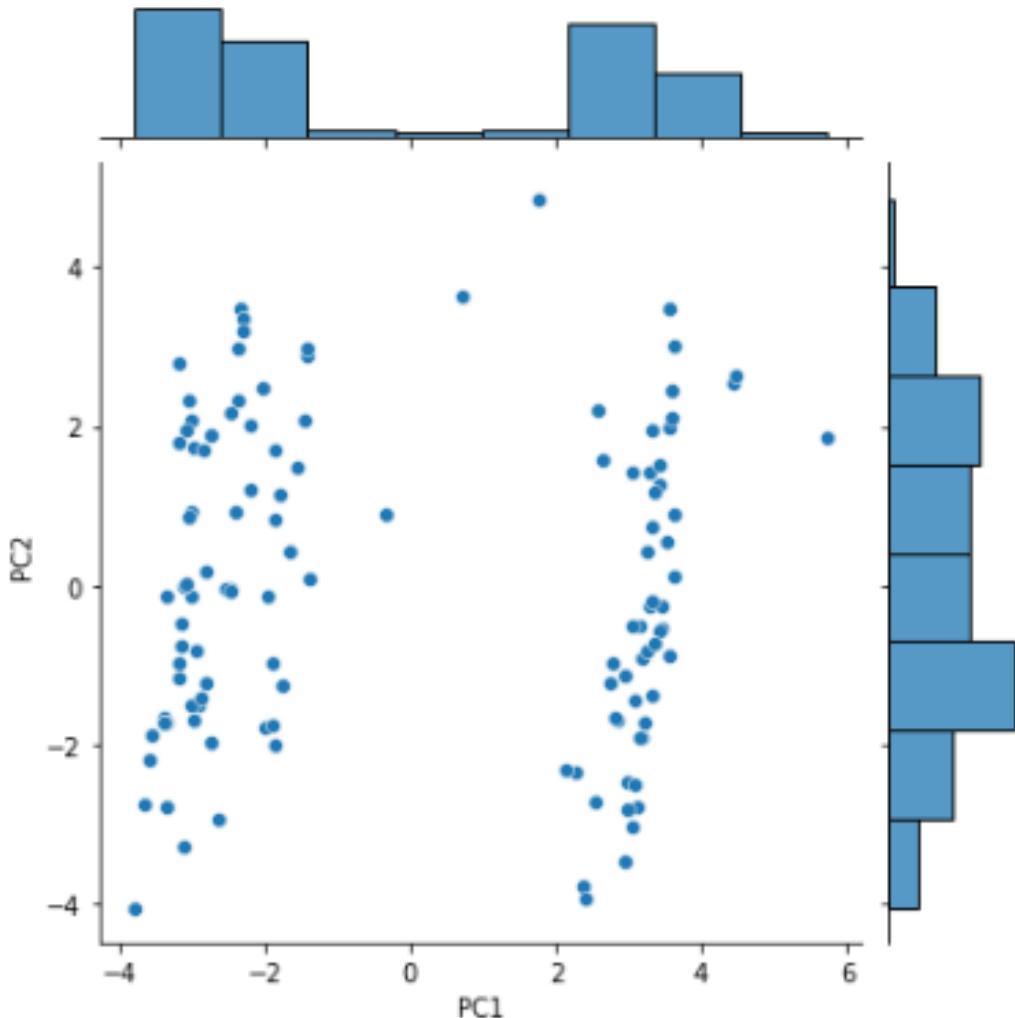
## Gán tên cho các Principal Components
PC_name = ['PC1', 'PC2', 'PC3', 'PC4',
            'PC5', 'PC6', 'PC7', 'PC8',
            'PC9', 'PC10', 'PC11', 'PC12',
            'PC13', 'PC14', 'PC15', 'PC16',
            'PC17']

## Chiếu dữ liệu vào không gian mới (Transform data)
B      = pca.transform(x)
principalDf = pd.DataFrame(data = B, columns = PC_name)

## Biểu diễn trực quan dữ liệu (KHÔNG phân lớp)
import seaborn         as sns
plt.figure(figsize = (8, 8))
sns.jointplot(x = PC_name[0],
               y = PC_name[1],
               data = principalDf)
plt.show()

```

Hình 63: Code Python biểu diễn trực quan dữ liệu với $k = 17$ – KHÔNG phân lớp



Hình 64: Biểu đồ biểu diễn trực quan dữ liệu với $k = 17$

```
▶ ## Lấy cột phân lớp (Class) trong file dữ liệu
target = 'loyal'
y = np.array(data.loyal)
y = pd.DataFrame(data = y, columns = [target])

## Ghép cột phân lớp (Class) vào ma trận PCA
finalDf = pd.concat([principalDf, y], axis = 1)
print('\n* Ma trận B_T (có thêm biến phân lớp Class)')
print(finalDf.head(), '\n')

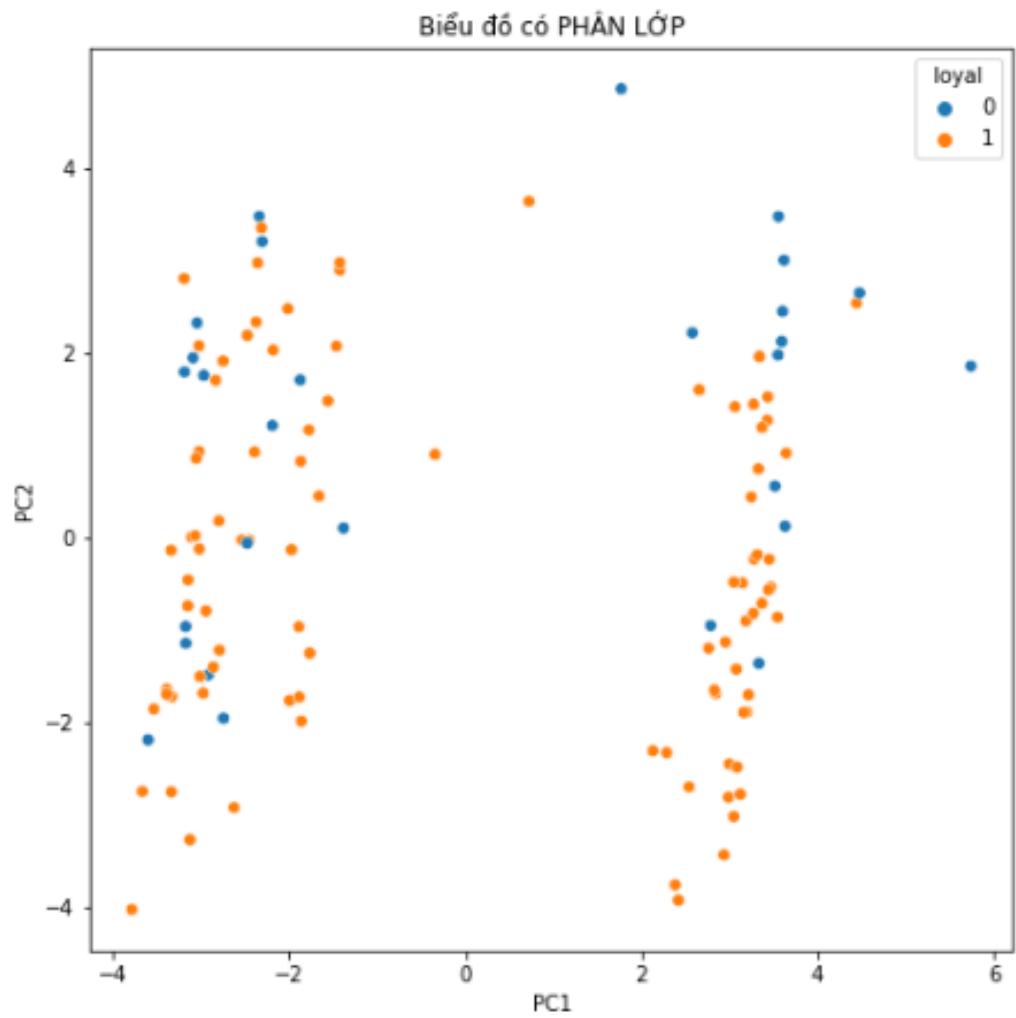
## Biểu diễn trực quan dữ liệu (có PHÂN LỚP)
plt.figure(figsize = (8, 8))
plt.title('Biểu đồ có PHÂN LỚP')
sns.scatterplot(x = PC_name[0], y = PC_name[1], data = finalDf, hue = target, legend = 'full')
plt.show()
```

Hình 65: Code Python biểu diễn trực quan dữ liệu với $k = 17$ – CÓ phân lớp

* Ma trận B_T (có thêm biến phân lớp Class)

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	\
0	-3.023493	0.927661	-1.719289	1.853745	0.671917	-1.738214	-0.071695	
1	3.423904	1.266957	-1.514664	-0.201174	-0.338070	-1.239529	-0.210649	
2	-3.021978	-0.127394	-0.543171	1.812899	1.554075	1.385226	0.745270	
3	3.598401	2.447354	1.053994	0.261417	0.374184	-1.113172	0.392140	
4	2.651519	1.596800	1.058894	2.882588	-1.786926	0.285546	1.175595	
	PC8	PC9	PC10	PC11	PC12	PC13	PC14	\
0	-0.715051	-0.725439	0.188135	0.902888	0.261586	-0.228576	-0.234137	
1	-0.193627	-1.528734	0.043018	-1.150350	-0.512985	0.478859	0.210881	
2	0.710851	-0.973631	0.232573	-0.545401	0.498722	0.031291	-0.911405	
3	-0.673172	0.882871	-0.679177	0.024768	-0.136387	0.182655	0.129856	
4	-0.932200	0.601068	0.854198	-1.459697	0.695517	-0.469999	-0.354991	
	PC15	PC16	PC17	loyal				
0	0.054983	0.747057	-0.204330	1				
1	-0.830747	0.179970	-0.677264	1				
2	-0.502222	-0.233073	0.269146	1				
3	0.601668	-0.148867	0.072461	0				
4	-0.491145	0.415505	-0.184593	1				

Bảng 11: Ma trận B_T



Hình 66: Biểu diễn trực quan dữ liệu với $k = 17$ – CÓ phân lớp

Ở đây ta thấy kết quả phân lớp chưa được hợp lý, vì có những chấm xanh trộn lẫn trong chấm màu cam. Vì thế ta tiếp tục chuẩn hóa bộ dữ liệu đã giảm chiều để cho ra kết quả tốt hơn

```

▶ pca_norm = PCA(k)

## Chuẩn hóa dữ liệu
from sklearn.preprocessing import StandardScaler
data_norm = pd.DataFrame(StandardScaler().fit_transform(data)) # tự động loại cột class

## Áp dụng PCA
pca_norm.fit(data_norm)

## Transform data
B_norm = pca_norm.transform(data_norm)
principalDf_norm = pd.DataFrame(data = B_norm, columns = PC_name)

## Lấy cột phân lớp (Class) trong file dữ liệu
y = np.array(data.loyal)
y = pd.DataFrame(data = y, columns = [target])

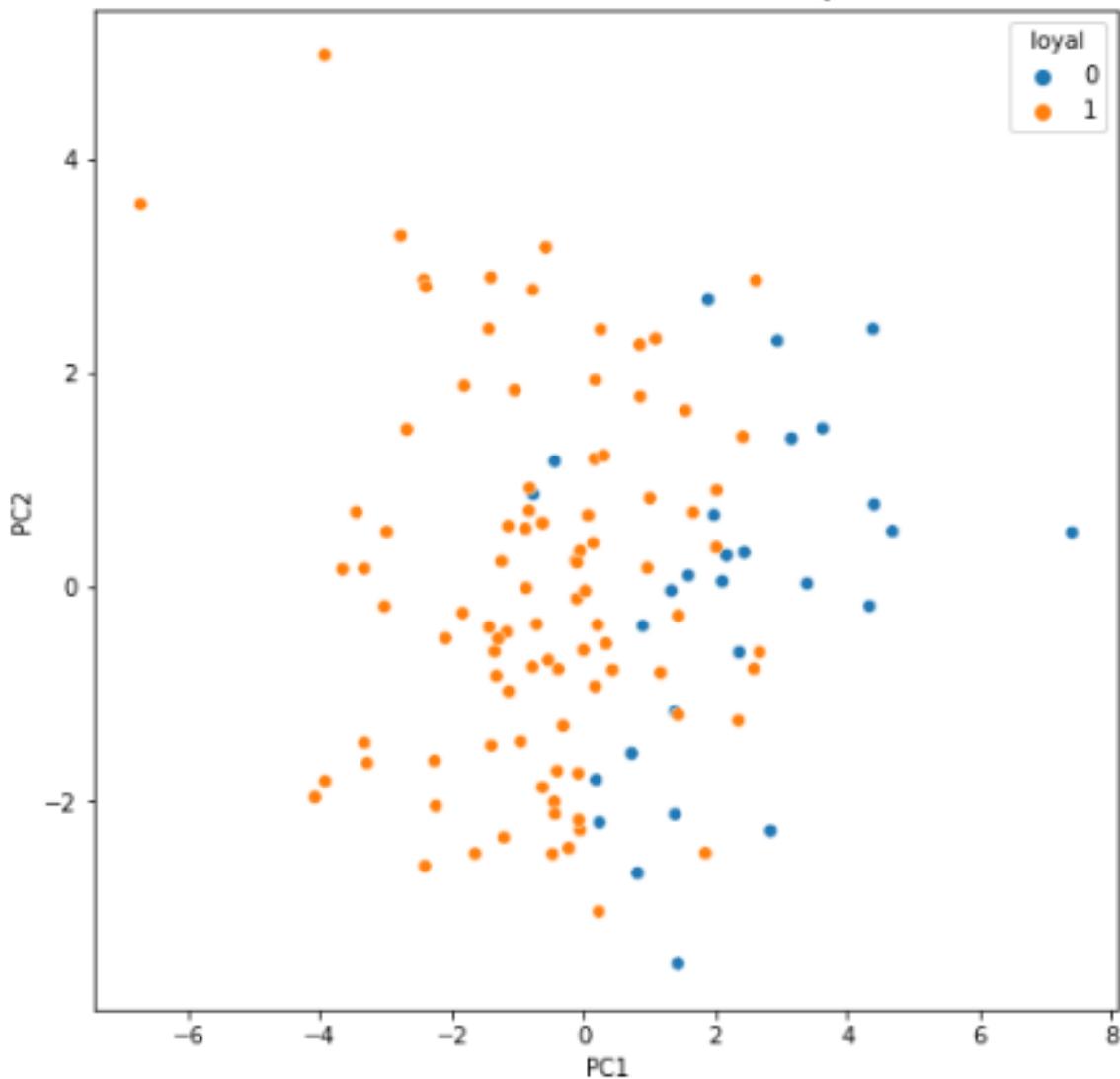
## Ghép cột phân lớp (Class) vào ma trận PCA
finalDf_norm = pd.concat([principalDf_norm, y], axis = 1)

## Biểu diễn trực quan dữ liệu (có PHÂN LỚP)
plt.figure(figsize = (8, 8))
plt.title('Biểu đồ sau khi CHUẨN HÓA dữ liệu')
sns.scatterplot(x = PC_name[0], y = PC_name[1], data = finalDf_norm, hue = target, legend = 'full')
plt.show()

```

Hình 67: Code Python chuẩn hóa bộ dữ liệu đã giảm chiều

Biểu đồ sau khi CHUẨN HÓA dữ liệu



Biểu đồ 50: Biểu đồ sau khi chuẩn hóa dữ liệu

Nhận Xét: Sau khi đã chuẩn hóa, ta tiếp tục vẽ biểu đồ thể hiện độ phân tán của dữ liệu, ở đây ta thấy các chấm được phân biệt nhau cũng rõ khi nhìn từ mắt thường. Vậy từ bộ dữ liệu gốc, thông qua 2 cách giảm chiều dữ liệu, ta quan sát được bộ dữ liệu có thể giảm từ 30 chiều thành 17 chiều để có thể xác định đinh biên phân lớp loyal.

TÀI LIỆU THAM KHẢO

Statsmodels (n.d.). Scikit, no tears. Retrieved December 16, 2022, from <https://learn-scikit.oneoffcoder.com/statsmodels.html>

Scipy.stats.shapiro (n.d.). Scipy. Retrieved December 16, 2022, from <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.shapiro.html>

Using pandas crosstab to create a bar plot (n.d.). Geeksforgeeks. Retrieved December 16, 2022, from <https://www.geeksforgeeks.org/using-pandas-crosstab-to-create-a-bar-plot/>

Pipis, G. (2020, October 24). *How to Run the Chi-Square Test in Python*. Medium. Retrieved December 17, 2022, from <https://medium.com/swlh/how-to-run-chi-square-test-in-python-4e9f5d10249d>

Starbucks Customer Survey. (n.d.). Kaggle. Retrieved December 17, 2022, from <https://www.kaggle.com/datasets/mahirahmzh/starbucks-customer-retention-malaysia-survey?select=Starbucks+satisfactory+survey+encode+cleaned.csv>

Statistical charts in Python. (n.d.). Plotly. Retrieved December 17, 2022, from <https://plotly.com/python/statistical-charts/>

Bar charts in Python. (n.d.). Plotly. Retrieved December 17, 2022, from <https://plotly.com/python/bar-charts/>

Parallel Categories Diagram in Python. (n.d.). Plotly. Retrieved December 17, 2022, from <https://plotly.com/python/parallel-categories-diagram/>

BẢNG PHÂN CÔNG

Tên thành viên	Công việc	Mức độ tham gia
Nguyễn Đình Đại Nhơn	<ul style="list-style-type: none"> Thống kê mô tả 3 biểu đồ PCA 	Hoàn thành đúng hạn: 100%
Huỳnh Thị Cẩm Nhung	<ul style="list-style-type: none"> Thống kê mô tả 3 biểu đồ One-way ANOVA 	<ul style="list-style-type: none"> Biểu đồ: trễ nửa ngày One-way ANOVA: trễ 1 ngày <p>→ 80%</p>
Lê Thị Tuyết Nhung	<ul style="list-style-type: none"> Tiền xử lý dữ liệu (label encoding) 3 biểu đồ PCA 	Hoàn thành đúng hạn: 100%
Đoàn Vũ Minh Thanh	<ul style="list-style-type: none"> Giới thiệu đề tài Tiền xử lý dữ liệu (trừ label encoding) 3 biểu đồ Kiểm định Chi-square (code) Nộp bài 	Hoàn thành đúng hạn: 100%
Đoàn Anh Thư	<ul style="list-style-type: none"> 3 biểu đồ Kiểm định phân phối chuẩn + hậu kiểm Tổng hợp báo cáo 	Hoàn thành đúng hạn: 100%
Huỳnh Trần Anh Thy	<ul style="list-style-type: none"> 3 biểu đồ Kiểm định Chi-square (giải thích) Tổng hợp báo cáo 	<ul style="list-style-type: none"> Biểu đồ: trễ nửa ngày Còn lại: đúng hạn <p>→ 90%</p>