

ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN - TIN

—o0o—



HỆ HỖ TRỢ QUYẾT ĐỊNH

BÀI TẬP LỚN

Đề tài: Purchase Behavior Prediction

Giảng viên hướng dẫn: TS. Trần Ngọc Thăng

Nhóm sinh viên thực hiện: Bùi Minh Châu 20210110

Nguyễn Thị Vang 20216902

Mã lớp: 150331

Hà Nội, tháng 6 năm 2024

Mục lục

Lời mở đầu	1
Danh sách hình vẽ	3
1 Cơ sở lý thuyết	5
1.1 Thương mại điện tử	5
1.1.1 Khái niệm	5
1.1.2 Tầm quan trọng của thương mại điện tử	5
1.1.3 Vai trò Machine Learning, Ai trong ngành thương mại điện tử	7
1.2 Hành vi mua hàng của người tiêu dùng	7
1.2.1 Khái niệm	7
1.2.2 Tầm quan trọng của phân tích hành vi mua hàng của người tiêu dùng	7
1.2.3 Vai trò của công nghệ trong phân tích hành vi mua hàng của người tiêu dùng	8
1.3 Mô hình hồi quy logistic	9
1.3.1 Khái niệm	9
1.3.2 Thuật toán	9
1.4 Mô hình K-Nearest Neighbors(KNN)	10
1.4.1 Khái niệm	10
1.4.2 Thuật toán	11
1.5 Mô hình Support Vector Machine (SVM)	12
1.5.1 Khái niệm	12
1.5.2 Thuật toán	12
1.6 Mô hình Gaussian Naive Bayes	13
1.6.1 Naive Bayes	13
1.6.2 Gaussian Naïve Bayes	13
1.6.3 Thuật toán	14

1.7	Mô hình Decision Tree	15
1.7.1	Khái niệm	15
1.7.2	Thuật toán	16
1.8	Mô hình Random Forest Regression	19
1.8.1	Khái niệm	19
1.8.2	Thuật toán	19
1.9	Mô hình XGBoost	20
1.9.1	Khái niệm	20
1.9.2	Thuật toán	22
1.10	Mô hình CatBoost	23
1.10.1	Khái niệm	23
1.10.2	Thuật toán	23
1.11	Mô hình LightGBM	24
1.11.1	Khái niệm	24
1.11.2	Thuật toán	25
1.12	Mô hình MLP	25
1.12.1	Khái niệm	25
1.12.2	Thuật toán	26
2	Bài toán Dự đoán hành vi mua hàng của khách hàng	27
2.1	Phân tích bài toán	27
2.1.1	Bài toán nghiệp vụ	27
2.1.2	Bài toán kỹ thuật	28
2.2	Mô hình đề xuất	29
3	Thực nghiệm và kết quả	30
3.1	Dữ liệu	30
3.1.1	Mô tả bộ dữ liệu	30
3.1.2	Trực quan hóa và khám phá dữ liệu	33
3.1.3	Tiền xử lý dữ liệu	55
3.2	Các phương pháp đánh giá mô hình	58
3.3	Kết quả và chỉ số đánh giá	61
3.3.1	Bộ dữ liệu 1	61
3.3.2	Bộ dữ liệu 2	68
3.4	Thống kê và phân tích lỗi	71
4	Đóng gói mô hình	76
4.1	Giao diện chương trình	76

4.2 Kịch bản chương trình	76
Kết luận	78
Tài liệu tham khảo	78
Checklist	80

Lời mở đầu

Thời đại Internet lên ngôi, sử dụng Internet để đáp ứng nhu cầu tìm kiếm thông tin là một xu hướng quan trọng trong hành vi mua sắm hiện đại. Người tiêu dùng thường tìm kiếm thông tin về sản phẩm hoặc dịch vụ trên trang web của Doanh nghiệp để đáp ứng nhanh chóng và thuận tiện cho những thắc mắc của họ. Các nghiên cứu chỉ ra rằng đối với nhóm khách hàng từ 16 – 64 tuổi có xu hướng ghé thăm website chính của Thương hiệu để tìm hiểu về sản phẩm trước khi quyết định mua hàng. Nhận thấy tầm quan trọng của việc hành vi mua hàng của khách hàng, nhóm đã chọn đề tài "Dự đoán hành vi của khách hàng" để nghiên cứu và tìm hiểu.

Báo cáo được chia thành bốn chương chính:

- **Chương 1: Cơ sở lý thuyết** cung cấp cái nhìn tổng quan về thương mại điện tử, trong đó có hành vi mua hàng của người tiêu dùng và trình bày lý thuyết các mô hình dự đoán bao gồm: Logistic Regression, K-Nearest Neighbors, Support Vector Classification, Gaussian Naive Bayes, Decision Tree, Random Forest, XGBoost và CatBoost
- **Chương 2: Bài toán dự đoán hành vi khách hàng** sẽ tập trung ứng dụng các mô hình đã giới thiệu để dự đoán hành vi khách hàng, bao gồm phân tích dữ liệu và xử lý.
- **Thực nghiệm và kết quả** trình bày quá trình thực hiện các mô hình, phương pháp đánh giá mô hình bao gồm: Precision rate, Recall rate, F1 score, Support. Đồng thời, đưa ra kết quả thực hiện mô hình và thống kê, phân tích lỗi của mô hình.
- **Đóng gói mô hình** trình bày kết quả đóng gói mô hình bao gồm giao diện và kịch bản chương trình giúp người dùng dễ dàng sử dụng và triển khai mô hình thực tế.

Cuối cùng, chúng em xin gửi lời cảm ơn đến **thầy Trần Ngọc Thăng**, người đã trực tiếp giảng dạy bộ môn Hệ hỗ trợ quyết định và cũng là người luôn

sát cánh đồng hành cùng chúng em trong hành trình truyền tải tri thức. Cảm ơn thầy đã truyền tải cho chúng em rất nhiều kiến thức hay và bổ ích. Trong quá trình hoàn thành báo cáo, mặc dù chúng em đã cố gắng hoàn thiện nhưng không thể tránh khỏi những thiếu sót. Chúng em mong nhận được sự góp ý, nhận xét của thầy để bài báo cáo của chúng em được hoàn thiện hơn.

Danh sách hình vẽ

3.1	Tổng quan bộ dữ liệu	33
3.2	Thông tin thống kê	33
3.3	Số giá trị null từng cột	34
3.4	Biểu đồ heatmap thể hiện độ tương quan giữa các biến	35
3.5	Biểu đồ thống kê Review	36
3.6	Biểu đồ thống kê tần suất mua hàng theo từng giới tính	36
3.7	Biểu đồ thống kê tần suất mua hàng theo từng giới tính	37
3.8	Tổng quan bộ dữ liệu	38
3.9	Hiện thị giá trị null	38
3.10	Số liệu thống kê mô tả	39
3.11	Chuyển đổi dtype từ boolean thành chuỗi.	40
3.12	Số giá trị của biến mục tiêu	40
3.13	Phân tích phần trăm của tỷ lệ Revenue.	40
3.14	Tổng số khách hàng đã không mua và mua.	41
3.15	Phân phối theo tháng	41
3.16	Phân phối theo Operating Systems	42
3.17	Phân phối theo Browser	42
3.18	Phân phối theo Region	43
3.19	Phân phối theo Traffic Type	43
3.20	Phân phối theo Visitor Type	44
3.21	Phân phối theo tuần	44
3.22	Phân phối theo Revenue	45
3.23	Revenue theo Administrative	45
3.24	Revenue theo Administrative_Duration	46
3.25	Revenue theo Informational	46
3.26	Revenue theo Informational_Duration	47
3.27	Revenue theo ProductRelated	47
3.28	Revenue theo ProductRelated_Duration	48
3.29	Revenue theo BounceRates	48

3.30	Revenue theo ExitRates	49
3.31	Revenue theo PageValues	49
3.32	Revenue theo SpecialDay	50
3.33	Sự phân bố của các biến số	52
3.34	Kiểm tra mối quan hệ giữa các biến số.	53
3.35	Kiểm tra đa cộng tuyến.	53
3.36	Kiểm tra tính đa cộng tuyến sau khi loại bỏ ProductRelation và ExitRates	54
3.37	Phát hiện các ngoại lệ.	55
3.38	Kết quả sau khi xóa cột "Timestamp"	56
3.39	Kết quả sau khi phân nhóm tuổi	56
3.40	Kết quả sau khi phân nhóm tuổi	57
3.41	Kết quả chọn ra 10 đặc trưng tốt nhất	58
3.42	Kết quả sau khi chuẩn hóa	58
3.43	Mô hình hồi quy logistic	62
3.44	Mô hình SVC	63
3.45	Mô hình KNN	63
3.46	Mô hình GNB	64
3.47	Mô hình cây quyết định	65
3.48	Mô hình rừng ngẫu nhiên	66
3.49	Mô hình XGBoost	67
3.50	Mô hình CatBoost	67
3.51	Mô hình hồi quy logistic	68
3.52	Mô hình SVC	68
3.53	Mô hình cây quyết định	69
3.54	Mô hình rừng ngẫu nhiên	69
3.55	Mô hình Gradient Boosting	70
3.56	Mô hình LGBM	70
3.57	Mô hình MLP	71

Chương 1

Cơ sở lý thuyết

1.1 Thương mại điện tử

1.1.1 Khái niệm

Thương mại điện tử (eCommerce) là sự mua bán sản phẩm hay dịch vụ trên các hệ thống điện tử như Internet và các mạng máy tính.

Thương mại điện tử là gì Theo WTO: Thương mại điện tử (hay thương mại trực tuyến) bao gồm việc sản xuất, quảng cáo, bán hàng và phân phối sản phẩm được mua bán và thanh toán trên mạng Internet, nhưng được giao nhận một cách hữu hình, cả các sản phẩm giao nhận cũng như những thông tin số hoá thông qua mạng Internet. Việc mua bán hàng hóa trên Shopee, Lazada hoặc qua website thương mại là các ví dụ nổi bật về thương mại điện tử.

Ngoài việc hiểu về hệ thống thương mại điện tử là gì, doanh nghiệp cũng cần nhớ rõ về các hoạt động chủ yếu của thương mại điện tử gồm những gì để có thể ứng dụng phù hợp nhất. Các hoạt động này bao gồm:

- Mua bán và trao đổi hàng hóa, dịch vụ trực tuyến
- Mua bán vé trực tuyến
- Thanh toán online
- Chăm sóc và hỗ trợ khách hàng online

1.1.2 Tầm quan trọng của thương mại điện tử

Với người tiêu dùng

- **Đa dạng lựa chọn:** nhiều sự lựa chọn cả trong, ngoài nước, tìm kiếm thông tin sản phẩm trong vài giây

- **Đa dạng lựa chọn:** nhiều sự lựa chọn cả trong, ngoài nước, tìm kiếm thông tin sản phẩm trong vài giây.
- **Giá cả cạnh tranh:** dễ dàng so sánh giá cả sản phẩm ở tất cả thị trường khác nhau, nhiều chương trình khuyến mãi, mã giảm giá.
- **Tiết kiệm thời gian:** khi sử dụng thương mại điện tử có thể chủ động tìm kiếm, chọn lựa mọi lúc, mọi nơi, tiến hành các giao dịch mua bán 24 giờ mỗi ngày.
- **Giao hàng nhanh hơn:** Thương mại điện tử ra đời đã tạo ra nhiều sản phẩm số hóa như phần mềm, các file hình ảnh có thể dễ dàng tìm kiếm, tải về và xem. Cũng như nhờ việc sử dụng internet mà người tiêu dùng có thể theo dõi được đơn hàng từ khi sản xuất cho tới khi hàng đang trên đường vận chuyển bằng đường bưu điện.
- **Giao dịch mọi lúc, mọi nơi:** Thương mại điện tử cho phép người tiêu dùng có thể tiến hành các giao dịch mua bán 24 giờ mỗi ngày, liên tục các ngày suốt cả năm từ bất cứ nơi nào.
- **Mua hàng với số lượng lớn với giá cả cạnh tranh:** Nhờ sử dụng internet mà khách hàng nhanh chóng tìm kiếm được thông tin về những chương trình khuyến mại, giảm giá mua hàng từ các nhà bán lẻ khác nhau trên toàn cầu. Ngoài ra, thương mại điện tử còn cho phép các khách hàng cá nhân có thể đặt một đơn hàng với số lượng lớn với giá cả cạnh tranh.
- **Chia sẻ kinh nghiệm, trải nghiệm:** Thương mại điện tử cho phép người tiêu dùng có thể trao đổi ý kiến cũng như chia sẻ kinh nghiệm trên các diễn đàn, trang web mua bán.

Với người bán

- **Mở rộng quy mô thị trường:** Thị trường toàn cầu không biên giới, tự tham gia khâu chuẩn bị, chụp hình, đăng thông tin sản phẩm.
- **Chi phí hợp lý:** Tự động hóa các quy trình làm việc ví dụ như thanh toán qua các cổng điện tử, không mất chi phí mặt bằng, quảng cáo tiếp thị, thuê nhân công.
- **Giữ liên lạc với khách hàng:** Nâng cao hiệu quả quan hệ khách hàng và phân tích hành vi người tiêu dùng để tiếp tục bán hàng.
- **Dịch vụ 24/7/365:** Tăng số lượng đơn đặt hàng mà doanh nghiệp nhận được vì khách hàng thích một cửa hàng “luôn mở”.

Với xã hội

- Nâng cao tính cộng đồng.
- Nâng cao chất lượng cuộc sống.
- Tiết kiệm thời gian mua bán, đi lại.
- Hội nhập với nền kinh tế thế giới.

1.1.3 Vai trò Machine Learning, Ai trong ngành thương mại điện tử

- **Đề xuất được cá nhân hóa:** Hệ thống AI phân tích dữ liệu người dùng để điều chỉnh đề xuất sản phẩm, giúp việc mua sắm trở nên thuận lợi và hấp dẫn hơn
- **Nâng cao dịch vụ khách hàng:** Chatbot và trợ lý ảo được hỗ trợ bởi AI, đảm bảo chất lượng dịch vụ 24/7 hỗ trợ phản hồi thông tin ngay lập tức, đảm bảo người mua hàng có trải nghiệm liền mạch.
- **Phân tích dự đoán:** Thuật toán ML có thể dự báo xu hướng bán hàng nhờ việc phân tích dữ liệu được thu thập, cho phép doanh nghiệp đưa ra quyết định và chiến lược sáng suốt.
- **Cải thiện quản lý hàng tồn kho:** Nhờ việc phân tích và đưa ra dự đoán, doanh nghiệp có thể quản lý hiệu quả tồn kho, giúp giảm thiểu lãng phí và tiết kiệm chi phí cho doanh nghiệp.

1.2 Hành vi mua hàng của người tiêu dùng

1.2.1 Khái niệm

Hành vi mua hàng của người tiêu dùng là cách mà người dùng tiếp cận, lựa chọn, mua sắm và sử dụng sản phẩm hoặc dịch vụ để thỏa mãn nhu cầu của họ. Hành vi mua hàng của người tiêu dùng có thể thay đổi dựa trên từng cá nhân, thời điểm và ngữ cảnh cụ thể, từ đó tạo ra sự đa dạng trong cách mà người tiêu dùng quyết định mua sắm.

1.2.2 Tầm quan trọng của phân tích hành vi mua hàng của người tiêu dùng

Việc hiểu rõ hành vi mua hàng của người tiêu dùng là vô cùng quan trọng đối với các doanh nghiệp. Thông qua việc nghiên cứu hành vi mua hàng, doanh

ng nghiệp có thể đưa ra các chiến lược marketing phù hợp để thu hút và giữ chân khách hàng.

Xác định khách hàng mục tiêu thông qua nghiên cứu thị trường mục tiêu
Xác định khách hàng mục tiêu thông qua nghiên cứu thị trường mục tiêu Dưới đây là một số lý do tại sao doanh nghiệp cần phân tích hành vi mua hàng của người tiêu dùng:

- **Hiểu rõ nhu cầu và mong muốn của khách hàng:** Nhờ việc phân tích hành vi mua hàng, nhà quản lý có thể hiểu rõ hơn về những gì khách hàng đang tìm kiếm, từ đó đưa ra các sản phẩm và dịch vụ đáp ứng nhu cầu của họ.
- **Xác định thị trường mục tiêu:** Thị trường mục tiêu là tập hợp những khách hàng mà doanh nghiệp muốn nhắm đến. Doanh nghiệp cần phân tích hành vi mua hàng của người tiêu dùng để khoanh vùng thị trường mục tiêu của mình.
- **Xây dựng các chiến lược marketing hiệu quả:** Các chiến lược marketing hiệu quả cần được xác định dựa trên nhu cầu và mong muốn của khách hàng. Để đưa ra các chiến dịch truyền thông, tiếp thị hiệu quả cùng các chương trình khuyến mãi thu hút khách hàng.
- **Cải thiện trải nghiệm khách hàng:** Trải nghiệm khách hàng là một yếu tố quan trọng quyết định sự thành công của doanh nghiệp. Bằng việc thấu hiểu nhu cầu khách hàng, doanh nghiệp có thể xác định những điểm cần cải thiện và mang lại trải nghiệm tốt hơn.
- **Tăng cường lòng trung thành của khách hàng:** Phân tích hành vi mua hàng giúp doanh nghiệp hiểu rõ cách thức khách hàng tương tác với thương hiệu và sản phẩm của mình. Từ đó giúp doanh nghiệp cải thiện dịch vụ chăm sóc, mang lại giá trị và sự hài lòng cho khách hàng.

1.2.3 Vai trò của công nghệ trong phân tích hành vi mua hàng của người tiêu dùng

Công nghệ đóng vai trò vô cùng quan trọng trong việc phân tích hành vi khách hàng trong nhiều khía cạnh khác nhau. Dưới đây là một số vai trò quan trọng của công nghệ trong việc này:

1. **Thu thập dữ liệu:** Công nghệ cho phép tổng hợp và thu thập dữ liệu từ nhiều nguồn khác nhau như trang web, ứng dụng di động, mạng xã hội,

email, và các hệ thống CRM (Customer Relationship Management). Việc thu thập dữ liệu chi tiết và đa dạng giúp các doanh nghiệp hiểu rõ hơn về hành vi của khách hàng.

2. **Phân tích và khai thác dữ liệu:** Công nghệ phân tích dữ liệu như các thuật toán máy học và học sâu có thể giúp phát hiện ra các xu hướng và mẫu hành vi khách hàng. Ví dụ, việc áp dụng các mô hình học máy để dự đoán hành vi mua hàng, phân khúc khách hàng, hoặc đề xuất sản phẩm phù hợp.
3. **Đánh giá hiệu quả chiến dịch marketing:** Công nghệ cho phép đo lường và phân tích hiệu quả của các chiến dịch marketing dựa trên dữ liệu về hành vi khách hàng. Điều này giúp các doanh nghiệp hiểu được cái gì hoạt động và cái gì không hoạt động trong các chiến dịch tiếp thị của họ.
4. **Cải thiện trải nghiệm khách hàng:** Công nghệ giúp cá nhân hóa trải nghiệm khách hàng thông qua việc cung cấp các đề xuất sản phẩm cá nhân hóa, dịch vụ chăm sóc khách hàng tự động và hiệu quả hơn.
5. **Dự báo và tối ưu hóa:** Công nghệ cho phép dự báo hành vi khách hàng trong tương lai và tối ưu hóa các chiến lược kinh doanh dựa trên những dự báo này.

Tóm lại, công nghệ không chỉ giúp các doanh nghiệp hiểu sâu hơn về hành vi khách hàng mà còn giúp tăng cường hiệu quả và tính cá nhân hóa trong các chiến lược kinh doanh và tiếp thị.

1.3 Mô hình hồi quy logistic

1.3.1 Khái niệm

Hồi quy Logistic là một mô hình thống kê được sử dụng để phân loại nhị phân, tức dự đoán một đối tượng thuộc vào một trong hai nhóm. Hồi quy Logistic làm việc dựa trên nguyên tắc của hàm sigmoid – một hàm phi tuyến tự chuyển đầu vào của nó thành xác suất thuộc về một trong hai lớp nhị phân.

1.3.2 Thuật toán

Hồi quy Logistic hoạt động dựa trên hàm Sigmoid, được biểu diễn như sau:

$$S(z) = \frac{1}{1 + e^{-z}}$$

Hàm Sigmoid nhận đầu vào là một giá trị z bất kỳ, và trả về đầu ra là một giá trị xác suất nằm trong khoảng $[0,1]$. Khi áp dụng vào mô hình Hồi quy Logistic với đầu vào là ma trận dữ liệu \mathbf{X} và trọng số \mathbf{w} , ta có:

$$z = \mathbf{X}\mathbf{w}$$

Việc huấn luyện của mô hình là tìm ra bộ trọng số \mathbf{w} sao cho đầu ra dự đoán của hàm Sigmoid gần với kết quả thực tế nhất. Để làm được điều này, ta sử dụng hàm mất mát (Loss Function) để đánh giá hiệu năng của mô hình. Mô hình càng tốt khi hàm mất mát càng nhỏ. **Hàm mất mát (Loss Function)** là một hàm số được sử dụng để đo lường mức độ lỗi mà mô hình của chúng ta tạo ra khi dự đoán các kết quả từ dữ liệu đầu vào. Trong bài toán Hồi quy Logistic, chúng ta sử dụng hàm mất mát Cross-Entropy (còn gọi là Log Loss) để đánh giá hiệu năng của mô hình.

Hàm mất mát Cross-Entropy được định nghĩa như sau:

$$L(\mathbf{w}) = -\frac{1}{n} \sum_{i=1}^n [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$$

Trong đó:

- n : số lượng mẫu dữ liệu trong tập huấn luyện.
- y_i : giá trị thực tế của đầu ra thứ i .
- p_i : xác suất dự đoán thuộc lớp 1 của mô hình cho đầu vào thứ i .

Hàm Cross-Entropy đo lường khoảng cách giữa hai phân phối xác suất y_i và p_i . Khi mô hình dự đoán chính xác, tức là nếu $y_i = 1$, thì p_i càng gần 1, và nếu $y_i = 0$, thì p_i càng gần 0. Sau đó, hàm mất mát sẽ tiến gần về 0.

Trong quá trình huấn luyện, chúng ta tìm cách cập nhật bộ trọng số \mathbf{w} sao cho giá trị hàm mất mát Cross-Entropy đạt giá trị nhỏ nhất, dẫn đến một mô hình dự đoán tốt nhất.

Để tìm giá trị tối ưu cho bộ trọng số \mathbf{w} , chúng ta có thể sử dụng kỹ thuật **Gradient Descent**. Tại mỗi bước lặp, chúng ta cập nhật \mathbf{w} theo phương pháp đạo hàm của hàm mất mát $L(\mathbf{w})$ theo \mathbf{w} .

1.4 Mô hình K-Nearest Neighbors(KNN)

1.4.1 Khái niệm

KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng

của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới. Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

1.4.2 Thuật toán

K-Nearest Neighbors (KNN) là một thuật toán học máy giám sát được sử dụng cho cả phân loại và hồi quy. Dưới đây là các bước cơ bản để thực hiện phân loại bằng KNN:

1. Bước 1: Định nghĩa tập dữ liệu

- Gọi D là tập các điểm dữ liệu đã được gán nhãn.
- Gọi A là dữ liệu chưa được phân loại.

2. Bước 2: Tính khoảng cách Đo khoảng cách từ dữ liệu mới A đến tất cả các dữ liệu khác đã được phân loại trong D . Có nhiều cách để đo khoảng cách, bao gồm:

- Khoảng cách Euclidian

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- Khoảng cách Manhattan

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

- Khoảng cách Minkowski

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

- Khoảng cách Trọng số

$$d(x, y) = \sqrt{\sum_{i=1}^n w_i (x_i - y_i)^2}$$

Trong đó w_i là trọng số của thuộc tính i .

3. Bước 3: Chọn K Chọn K (K là tham số mà bạn định nghĩa) khoảng cách nhỏ nhất.
4. Bước 4: Kiểm tra lớp Kiểm tra danh sách các lớp có khoảng cách ngắn nhất và đếm số lượng của mỗi lớp xuất hiện.
5. Bước 5: Lấy đúng lớp (lớp xuất hiện nhiều lần nhất).
6. Bước 6: Lớp của dữ liệu mới là lớp mà bạn đã nhận được ở bước 5.

1.5 Mô hình Support Vector Machine (SVM)

1.5.1 Khái niệm

Support Vector Machine (SVM) là một thuật toán học máy có giám sát, nó có thể sử dụng cho cả việc phân loại (classification) và hồi quy (regression). Tuy nhiên, SVM được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta biểu diễn dữ liệu dưới dạng các điểm trong không gian n chiều (với n là số lượng các đặc trưng hay tính năng của dữ liệu). Giá trị của mỗi đặc trưng sẽ tương ứng với một trong các tọa độ của điểm đó.

1.5.2 Thuật toán

Siêu phẳng (Hyperplane)

Một siêu phẳng trong không gian n chiều được biểu diễn bởi phương trình:

$$w \cdot x + b = 0$$

Trong đó:

- w là vector trọng số.
- x là vector đầu vào.
- b là hằng số dịch chuyển (bias).

Lề (Margin)

Lề của một siêu phẳng là khoảng cách giữa siêu phẳng đó và các điểm dữ liệu gần nhất từ cả hai lớp. Mục tiêu của SVM là tìm ra siêu phẳng tối ưu sao cho lề này được tối đa hóa. Khoảng cách từ một điểm dữ liệu x đến siêu phẳng được tính bằng:

$$\frac{|w \cdot x + b|}{\|w\|}$$

Bài toán tối ưu hóa SVM

Để tìm siêu phẳng tối ưu, ta cần giải bài toán tối ưu hóa sau:

$$\min_{w,b} \frac{1}{2} \|w\|^2$$

với ràng buộc:

$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$

Bài toán này có thể được giải bằng phương pháp nhân tử Lagrange và các kỹ thuật tối ưu hóa khác.

SVM với hạt nhân (Kernel)

Trong nhiều trường hợp, dữ liệu không thể phân tách tuyến tính trong không gian đầu vào. SVM sử dụng kỹ thuật hạt nhân (kernel trick) để ánh xạ dữ liệu vào một không gian đặc trưng cao chiều hơn, nơi dữ liệu có thể phân tách tuyến tính. Một số hàm hạt nhân phổ biến bao gồm:

- Hạt nhân tuyến tính (Linear kernel): $K(x_i, x_j) = x_i \cdot x_j$
- Hạt nhân Gaussian (RBF kernel): $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$
- Hạt nhân bậc đa thức (Polynomial kernel): $K(x_i, x_j) = (x_i \cdot x_j + c)^d$

1.6 Mô hình Gaussian Naive Bayes

1.6.1 Naive Bayes

Gaussian Naive Bayes là một giải thuật thuộc lớp giải thuật thống kê, nó có thể dự đoán xác suất của một phần tử dữ liệu thuộc vào một lớp là bao nhiêu. Gaussian Naive Bayes được dựa trên định lý Bayes (định lý được đặt theo tên tác giả của nó là Thomas Bayes)

1.6.2 Gaussian Naïve Bayes

Naive Bayes có thể được mở rộng thành các thuộc tính có giá trị thực, phổ biến nhất bằng cách giả sử phân phối Gaussian.

Phần mở rộng này của Naive Bayes được gọi là Gaussian Naive Bayes. Các hàm khác có thể được sử dụng để ước tính phân phối dữ liệu, nhưng Gaussian (hoặc phân phối chuẩn) là dễ nhất để làm việc vì bạn chỉ cần ước tính giá trị trung bình và độ lệch chuẩn từ dữ liệu đào tạo của bạn.

1.6.3 Thuật toán

Naive Bayes

1. Gọi D là tập dữ liệu huấn luyện, trong đó mỗi phần tử dữ liệu X được biểu diễn bằng một vector chứa n giá trị thuộc tính $A_1, A_2, \dots, A_n = \{x_1, x_2, \dots, x_n\}$.
2. Giả sử có m lớp C_1, C_2, \dots, C_m . Cho một phần tử dữ liệu X , bộ phân lớp sẽ gán nhãn cho X là lớp có xác suất hậu nghiệm lớn nhất. Cụ thể, bộ phân lớp Bayes sẽ dự đoán X thuộc vào lớp C_i nếu và chỉ nếu:

$$P(C_i|X) > P(C_j|X) \quad (1 \leq i, j \leq m, i \neq j)$$

Giá trị này sẽ được tính dựa trên Định lý Bayes.

3. Để tìm xác suất lớn nhất, ta nhận thấy các giá trị $P(X)$ là giống nhau với mọi lớp nên không cần tính. Do đó ta chỉ cần tìm giá trị lớn nhất của $P(X|C_i) \cdot P(C_i)$. Chú ý rằng $P(C_i)$ được ước lượng bằng $\frac{|D_i|}{|D|}$, trong đó D_i là tập các phần tử dữ liệu thuộc lớp C_i . Nếu xác suất tiên nghiệm $P(C_i)$ không xác định được thì ta coi chúng bằng nhau $P(C_1) = P(C_2) = \dots = P(C_m)$, khi đó ta chỉ cần tìm giá trị $P(X|C_i)$ lớn nhất.
4. Khi số lượng các thuộc tính mô tả dữ liệu là lớn thì chi phí tính toán $P(X|C_i)$ là rất lớn, do đó có thể giảm độ phức tạp của thuật toán Naive Bayes giả thiết các thuộc tính độc lập nhau. Khi đó ta có thể tính:

$$P(X|C_i) = P(x_1|C_i) \cdot P(x_2|C_i) \cdot \dots \cdot P(x_n|C_i)$$

Gaussian Naive Bayes

Với Gaussian Naive Bayes, giả sử các giá trị thuộc tính x_1, x_2, \dots, x_n đều tuân theo phân phối chuẩn (Gaussian distribution). Xác suất có điều kiện $P(x_k|C_i)$ được tính như sau:

$$P(x_k|C_i) = \frac{1}{\sqrt{2\pi\sigma_{C_i}^2}} \exp\left(-\frac{(x_k - \mu_{C_i})^2}{2\sigma_{C_i}^2}\right)$$

Trong đó:

- μ_{C_i} là giá trị trung bình của thuộc tính x_k cho lớp C_i .

- σ_{C_i} là độ lệch chuẩn của thuộc tính x_k cho lớp C_i .

Từ đó, xác suất $P(X|C_i)$ có thể được tính bằng tích của các xác suất có điều kiện cho từng thuộc tính:

$$P(X|C_i) = \prod_{k=1}^n P(x_k|C_i)$$

Cuối cùng, xác suất hậu nghiệm $P(C_i|X)$ được tính dựa trên Định lý Bayes:

$$P(C_i|X) \propto P(X|C_i) \cdot P(C_i)$$

Bộ phân lớp Gaussian Naive Bayes sẽ gán nhãn cho X là lớp C_i nếu và chỉ nếu:

$$P(C_i|X) > P(C_j|X) \quad (1 \leq i, j \leq m, i \neq j)$$

1.7 Mô hình Decision Tree

1.7.1 Khái niệm

Đây là một trong những thuật toán phổ biến nhất được sử dụng hiện nay. Cây quyết định là thuật toán học có giám sát, dùng để phân loại các vấn đề. Thuật toán có thể thực hiện cho cả biến phân loại và biến liên tục. Trong thuật toán này, ta chia dữ liệu thành 2 hoặc nhiều lớp dựa trên phân loại theo các thuộc tính/biến quan trọng.

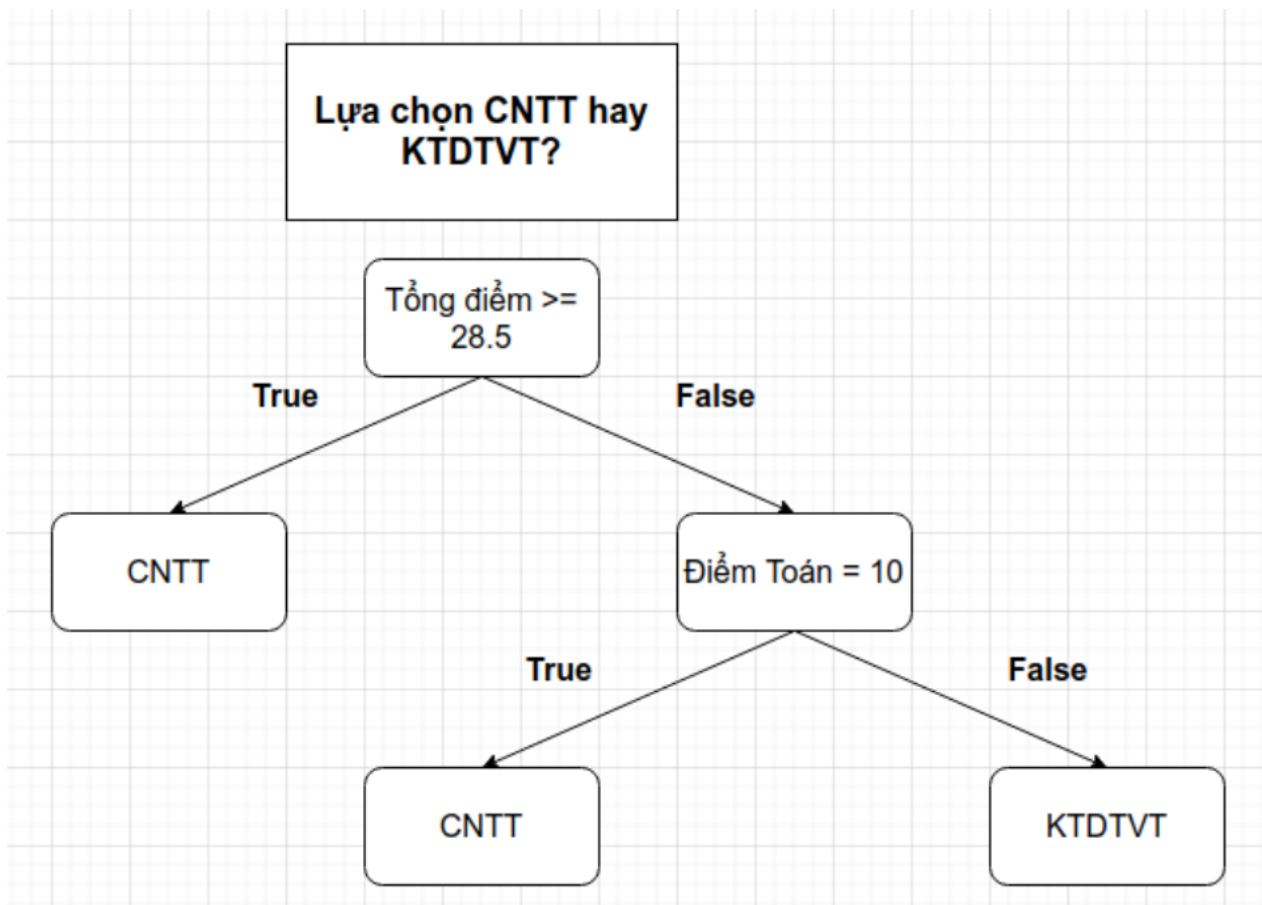
Vậy cây quyết định là gì? Bản chất của cây quyết định là một đồ thị có hướng được sử dụng cho việc ra quyết định. Lấy ví dụ, sau khi biết điểm thi tốt nghiệp THPT, bạn muốn xây dựng một chiến lược đăng kí ngành học bằng một loạt các lựa chọn:

Nếu tổng ba môn của bạn là lớn hơn 28.5 bạn sẽ nộp vào ngành CNTT.

Trái lại, nếu điểm thi của bạn nhỏ hơn hoặc bằng 28.5 thì vẫn còn cơ hội cho bạn nếu điểm Toán cao vì điểm Toán có hệ số nhân là 2. Do đó bạn quyết định vẫn lựa chọn CNTT nếu điểm Toán được 10. Trường hợp còn lại bạn đăng ký vào ngành KTĐTVT.

Tập hợp các câu hỏi và lựa chọn của bạn có thể ở trên được khái quát thành một cây quyết định:

Cây quyết định ở sơ đồ trên còn được gọi là cây quyết định nhị phân vì một câu hỏi chỉ có hai phương án là True hoặc False. Trên thực tế có thể có những dạng cây quyết định khác nhiều hơn hai phương án cho một câu hỏi.



Chúng ta có một số khái niệm liên quan tới cây quyết định:

- **Node gốc (root node):** Là node ở vị trí đầu tiên của cây quyết định. Mọi phương án đều bắt nguồn từ node này. Ở ví dụ trên là "Tổng điểm ≥ 28.5 ".
- **Node cha (parent node):** Là node mà có thể rẽ nhánh xuống những node khác bên dưới. Node bên dưới được gọi là node con.
- **Node con (child node):** Là những node tồn tại dưới node cha.
- **Node lá (leaf node):** Là node cuối cùng của một quyết định. Tại đây chúng ta thu được kết quả dự báo. Node lá ở vị trí cuối cùng nên sẽ không có node con.
- **Node quyết định (non-leaf node):** Những node khác node lá.

1.7.2 Thuật toán

Có một vài thuật toán để tạo một cây quyết định, chúng ta sẽ nói về 2 trong số chúng:

1. ID3 (Iterative Dichotomiser 3)

ID3 là một thuật toán xây dựng cây quyết định dựa trên việc chọn thuộc tính sao cho thu hoạch thông tin (Information Gain) là lớn nhất. Với mỗi điều kiện để tách thì sẽ có chỉ số information gain tương ứng, chỉ số information gain càng cao thì việc tách càng tốt. Do đó mình sẽ duyệt qua hết các thuộc tính của dữ liệu, mỗi thuộc tính thử các giá trị để tách khác nhau, rồi chọn điều kiện có chỉ số information gain cao nhất để tách, và tiếp tục như thế cho tới node lá, chỉ gồm dữ liệu 1 lớp duy nhất. Các bước chính của ID3 bao gồm:

- **Entropy (Entropy function):** Đo lường mức độ không chắc chắn (uncertainty) trong tập dữ liệu. Entropy của một tập dữ liệu S được tính như sau:

$$\text{Entropy}(S) = - \sum_{i=1}^c p_i \log_2 p_i$$

Trong đó c là số lớp, p_i là tỷ lệ các mẫu thuộc lớp i trong tập S .

- **Information Gain (IG):** Đo lường lượng thông tin mới mà thuộc tính A mang lại cho việc phân chia dữ liệu. Information Gain được tính như sau:

$$\text{IG}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

Trong đó A là thuộc tính cần phân chia, $\text{Values}(A)$ là các giá trị có thể của thuộc tính A , S_v là tập con của các mẫu trong S mà thuộc tính A có giá trị v , và $|S|$ là tổng số mẫu trong S .

- **Ưu điểm của ID3:**
 - Hiệu quả đối với dữ liệu rời rạc.
 - Dễ hiểu và giải thích.
- **Nhược điểm của ID3:**
 - Không xử lý được dữ liệu liên tục mà cần phải rời rạc hóa trước.
 - Dễ bị ảnh hưởng bởi nhiễu trong dữ liệu.

2. CART (Classification and Regression Trees)

- **Sử dụng Chỉ số Gini (Classification)**

CART sử dụng Chỉ số Gini để đo lường độ đồng nhất của một tập dữ liệu tại mỗi nút của cây quyết định. Gini index tương tự như information

gain, dùng để đánh giá xem việc phân chia ở node điều kiện có tốt hay không. Chỉ số Gini được tính như sau cho một nút t :

$$Gini(t) = 1 - \sum_{i=1}^K (p_i)^2$$

Trong đó:

- K là số lớp (classes).
- p_i là tỷ lệ của lớp i trong tập dữ liệu tại nút t .

Chỉ số Gini thường được sử dụng để lựa chọn thuộc tính và ngưỡng phân chia sao cho khi phân chia dữ liệu, độ đồng nhất của các tập con (dựa trên tỷ lệ các lớp) là nhỏ nhất.

Vì khi tách mình muốn chỉ số gini ở các node con nhỏ, nên gini index mình mong muốn càng lớn càng tốt. Thuật toán CART khá giống ID3, chỉ thay gini index bằng information gain. Để tìm điều kiện tách, mình thử ở tất các thuộc tính, mỗi thuộc tính thử một số giá trị chia, rồi so sánh xem điều kiện nào chỉ số gini index giảm nhiều nhất thì sẽ chọn để chia.

– **Ưu điểm của CART:**

- * Phù hợp với cả bài toán phân loại và hồi quy.
- * Tính linh hoạt và dễ hiểu.

– **Nhược điểm của CART:**

- * Dễ bị overfitting nếu không kiểm soát.
- * Không cân bằng dữ liệu có thể ảnh hưởng đến hiệu suất.

Cách giải quyết khi model Decision Tree bị overfitting

1. Dừng việc thêm các node điều kiện vào cây dựa vào các điều kiện:

- Giới hạn độ sâu của cây.
- Chỉ định số phần tử tối thiểu (n) trong node lá, nếu 1 node có số phần tử ít hơn n thì sẽ không tách nữa.

2. Pruning.

1.8 Mô hình Random Forest Regression

1.8.1 Khái niệm

Random là ngẫu nhiên, Forest là rừng, nên ở thuật toán Random Forest mình sẽ xây dựng nhiều cây quyết định bằng thuật toán Decision Tree, tuy nhiên mỗi cây quyết định sẽ khác nhau (có yếu tố random). Sau đó kết quả dự đoán được tổng hợp từ các cây quyết định.

1.8.2 Thuật toán

– **Bootstrapping (Lấy mẫu tái chọn lọc):**

- Tập dữ liệu gốc có n mẫu dữ liệu.
- Sử dụng kỹ thuật Bootstrapping để lấy ngẫu nhiên n mẫu từ tập dữ liệu gốc. Quá trình này cho phép một số mẫu bị lấy nhiều lần (trùng lặp) trong tập dữ liệu mới.

– **Lấy mẫu thuộc tính:**

- Sau khi có tập dữ liệu mới từ Bootstrapping, chọn ngẫu nhiên k thuộc tính từ tổng số d thuộc tính ban đầu (với $k < d$).
- Điều này giúp mỗi cây quyết định trong Random Forest được xây dựng trên một tập dữ liệu con khác nhau và chỉ sử dụng một số thuộc tính để giảm sự phụ thuộc giữa các cây.

– **Xây dựng cây quyết định (Decision Tree):**

- Mỗi cây quyết định được xây dựng trên tập dữ liệu con và sử dụng chỉ số phân chia như Gini Index (trong trường hợp phân loại) hoặc MSE (trong trường hợp hồi quy) để tối ưu hóa quá trình tách các node.
- Các thuộc tính của cây quyết định như độ sâu tối đa (max depth), số mẫu tối thiểu trong một node để có thể tách (min samples split) cũng cần được xác định để tránh overfitting và đảm bảo tính tổng quát của mô hình.

– **Tích hợp dự đoán từ nhiều cây quyết định:**

- Kết quả dự đoán cuối cùng của Random Forest được tính bằng cách lấy trung bình (trong trường hợp hồi quy) hoặc phương pháp đa số (trong trường hợp phân loại) của các dự đoán từ các cây quyết định thành viên.

– **Tối ưu hóa và cấu hình:**

- Khi áp dụng Random Forest, các tham số quan trọng như số lượng cây quyết định (`n_estimators`), số lượng thuộc tính dùng để xây dựng mỗi cây (`max_features`) thường cần được điều chỉnh và tối ưu hóa để đạt hiệu suất tốt nhất trên tập kiểm tra.
- Điều này thường được thực hiện thông qua các kỹ thuật như **Grid Search** (lặp lại huấn luyện và đánh giá mô hình với các giá trị tham số khác nhau trên một lưới (grid) giá trị) và **Random Search** (lựa chọn ngẫu nhiên các giá trị trong khoảng giá trị được định trước cho mỗi tham số) để đánh giá và chọn lựa các giá trị tối ưu cho các tham số này.

1.9 Mô hình XGBoost

1.9.1 Khái niệm

XGBoost là một thuật toán học máy mạnh mẽ được sử dụng rộng rãi cho các bài toán hồi quy. Trong hồi quy, XGBoost xây dựng một tập hợp các cây quyết định (CART - Classification and Regression Trees) để dự đoán biến đầu ra dựa trên các biến đầu vào.

Hàm dự đoán

Hàm dự đoán cho một mẫu i được tính như sau:

$$\hat{y}_i = F(X_i) = \sum_{d=1}^D f_d(X_i),$$

trong đó:

- \hat{y}_i là giá trị dự đoán của mẫu i .
- $F(X_i)$ là hàm dự đoán cuối cùng, là tổng của dự đoán từ tất cả D cây.
- $f_d(X_i)$ là dự đoán từ cây thứ d cho mẫu i .

Hàm mục tiêu

Hàm mục tiêu là sự kết hợp giữa hàm mất mát và một phần thưởng chính quy. Nó được định nghĩa tại lần lặp d như sau:

$$\mathcal{L}^{(d)} = \sum_{i=1}^n \ell(y_i, \hat{y}_i^{(d-1)} + f_d(X_i)) + \sum_{j=1}^T \Omega(f_j),$$

trong đó:

- $\ell(y_i, \hat{y}_i)$ là hàm mất mát đo lường sự khác biệt giữa giá trị thực y_i và giá trị dự đoán \hat{y}_i .
- $\Omega(f_d)$ là phần thưởng chính quy cho cây thứ d .

Hàm mất mát

Trong bài toán hồi quy, hàm mất mát thường được sử dụng là hàm mất mát bình phương:

$$\ell(y_i, \hat{y}_i) = \frac{1}{2}(y_i - \hat{y}_i)^2$$

Phần thưởng chính quy

Phần thưởng chính quy giúp ngăn chặn mô hình overfitting bằng cách phạt các mô hình quá phức tạp. Nó được định nghĩa như sau:

$$\Omega(f_d) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^T w_j^2,$$

trong đó:

- γ là tham số phạt số lượng lá T .
- λ là hệ số phạt cho chuẩn ℓ_2 của trọng số lá w_j .

Cấu trúc cây

Cấu trúc q của cây có thể được định nghĩa như sau:

$$q : \mathbb{R}^p \rightarrow \{1, 2, \dots, T\},$$

trong đó q chỉ định một quan sát vào một lá cụ thể j trong cây. Mỗi lá j có liên kết với một trọng số w_j , và dự đoán cho một quan sát X_i trong cây d được đưa ra bởi:

$$f_d(X_i) = w_{q(X_i)}$$

Gradient và Hessian

Gradient và Hessian của hàm mất mát được tính như sau:

$$g_i = \frac{\partial \ell(y_i, \hat{y}_i)}{\partial \hat{y}_i}$$

$$h_i = \frac{\partial^2 \ell(y_i, \hat{y}_i)}{\partial \hat{y}_i^2}$$

Hàm mục tiêu được xấp xỉ bằng khai triển Taylor bậc hai:

$$\mathcal{L}^{(d)} \approx \sum_{i=1}^n \left[\ell(y_i, \hat{y}_i^{(d-1)}) + g_i f_d(X_i) + \frac{1}{2} h_i f_d(X_i)^2 \right] + \Omega(f_d)$$

Trọng số lá tối ưu

Trọng số lá tối ưu w_j được tính như sau:

$$w_j = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

trong đó I_j là tập hợp các chỉ số cho các quan sát rơi vào lá j .

Cắt tỉa

XGBoost sử dụng kỹ thuật cắt tỉa cây để kiểm soát độ phức tạp của cây và tránh overfitting. Sau khi các cây được xây dựng, các nút không cung cấp lợi ích đáng kể sẽ được cắt tỉa bằng cách sử dụng tham số ngưỡng.

1.9.2 Thuật toán

XGBoost là một dạng cụ thể của gradient boosting, một kỹ thuật học máy dựa trên ý tưởng kết hợp nhiều mô hình yếu để tạo ra một mô hình mạnh hơn. Dưới đây là các bước cơ bản của thuật toán XGBoost:

1. Gradient Boosting

Gradient boosting là một phương pháp iterative, trong đó mỗi mô hình mới được thêm vào để giảm thiểu lỗi của mô hình hiện tại. Mô hình hiện tại được cập nhật bằng cách thêm vào một mô hình mới, được huấn luyện để dự đoán các phần dư (residuals) của mô hình hiện tại.

2. Regularization

XGBoost bao gồm các thuật toán điều chuẩn (regularization) để kiểm soát độ phức tạp của mô hình và giảm thiểu hiện tượng overfitting. Hai dạng điều chuẩn chính được sử dụng là L1 (Lasso) và L2 (Ridge).

3. Tree Pruning

XGBoost thực hiện pruning trên cây quyết định để giảm thiểu các nhánh không cần thiết, giúp cải thiện hiệu suất và độ chính xác của mô hình. Cây được xây dựng dựa trên một metric cụ thể (như gain) và các nhánh có gain thấp được cắt bỏ.

4. Shrinkage

Shrinkage là một kỹ thuật giảm tốc độ học (learning rate) của mô hình. Sau mỗi lần cập nhật, các dự đoán mới được nhân với một hệ số nhỏ hơn 1 (learning rate), giúp mô hình học dần dần và tránh việc học quá mức từ dữ liệu.

5. Handling Missing Values

XGBoost có cơ chế xử lý các giá trị thiếu (missing values) tự động. Khi gặp giá trị thiếu, cây quyết định có thể tự động quyết định hướng đi tốt nhất dựa trên việc tối ưu hóa độ giảm lỗi.

6. Parallel Processing

XGBoost được thiết kế để tận dụng khả năng xử lý song song (parallel processing) của các phần cứng hiện đại. Điều này giúp tăng tốc độ huấn luyện mô hình đáng kể.

1.10 Mô hình CatBoost

1.10.1 Khái niệm

CatBoost (Categorical Boosting) là một thuật toán tăng cường độ chính xác (gradient boosting) được phát triển bởi Yandex, nhằm cải thiện hiệu suất và độ chính xác trong các bài toán phân loại và hồi quy, đặc biệt là khi làm việc với các dữ liệu có nhiều đặc điểm phân loại (categorical features).

CatBoost được thiết kế để giải quyết một số vấn đề phổ biến trong các mô hình gradient boosting truyền thống:

- Xử lý đặc điểm phân loại: CatBoost cung cấp một cách tiếp cận tự động và hiệu quả để xử lý các đặc điểm phân loại, giúp giảm thiểu việc cần phải biến đổi và tiền xử lý dữ liệu.
- Giảm hiện tượng overfitting: Thuật toán CatBoost được thiết kế để giảm hiện tượng overfitting thông qua cơ chế thứ tự ngẫu nhiên (ordered boosting).
- Hiệu suất cao: CatBoost tối ưu hóa việc tính toán gradient boosting để đạt hiệu suất cao cả về thời gian huấn luyện và khả năng suy luận.

1.10.2 Thuật toán

1. Gradient Boosting

CatBoost dựa trên ý tưởng của gradient boosting, trong đó mô hình cuối cùng được xây dựng thông qua việc cộng dồn nhiều cây quyết định nhỏ (decision trees). Mỗi cây quyết định mới được thêm vào để giảm thiểu lỗi của mô hình hiện tại.

2. Ordered Boosting

Ordered boosting là một trong những điểm khác biệt chính của CatBoost so với các thuật toán boosting truyền thống. Thay vì sử dụng toàn bộ dữ liệu huấn luyện để xây dựng mỗi cây quyết định mới, CatBoost chia dữ liệu thành hai phần theo thứ tự ngẫu nhiên. Một phần dữ liệu được sử dụng để xây dựng cây quyết định, và phần còn lại để dự đoán. Điều này giúp giảm thiểu hiện tượng overfitting.

3. Handling Categorical Features

CatBoost có cách tiếp cận đặc biệt để xử lý các đặc điểm phân loại:

- Mean Encoding: CatBoost sử dụng mean encoding, trong đó các giá trị phân loại được thay thế bằng giá trị trung bình của mục tiêu dự đoán cho các nhóm phân loại.
- Target Statistics: CatBoost sử dụng các thống kê mục tiêu để mã hóa các đặc điểm phân loại một cách hiệu quả, giúp mô hình học được các mối quan hệ phức tạp trong dữ liệu.

4. Quy trình của CatBoost:

- (a) Chuẩn bị dữ liệu: Bao gồm việc xác định các đặc điểm phân loại và đặc điểm số.
- (b) Phân chia dữ liệu: Chia dữ liệu huấn luyện thành các tập con ngẫu nhiên.
- (c) Xây dựng cây quyết định: Sử dụng ordered boosting để xây dựng các cây quyết định.
- (d) Kết hợp mô hình: Kết hợp các cây quyết định để tạo thành mô hình cuối cùng.

1.11 Mô hình LightGBM

1.11.1 Khái niệm

LightGBM (Light Gradient Boosting Machine) là một thuật toán học máy dựa trên kỹ thuật Gradient Boosting Framework, được phát triển bởi Microsoft

Research. Được thiết kế để cải thiện hiệu suất và tốc độ so với các thuật toán gradient boosting truyền thống như XGBoost.

1.11.2 Thuật toán

Thuật toán Gradient Boosting: LightGBM là một thuật toán học máy thuộc dạng gradient boosting, có nghĩa là nó xây dựng một chuỗi các cây quyết định (decision trees) để tối thiểu hóa hàm mất mát (loss function).

1. Tối ưu hóa cho tốc độ và hiệu suất: LightGBM được thiết kế đặc biệt để cải thiện tốc độ huấn luyện và tiêu tốn bộ nhớ so với các thuật toán gradient boosting truyền thống như XGBoost.
2. Histogram-based Gradient Boosting: Thay vì phân tách dữ liệu theo giá trị chính xác của đặc trưng, LightGBM sử dụng histogram để phân chia dữ liệu. Điều này giúp giảm đáng kể thời gian tính toán cho các bài toán có tập dữ liệu lớn.
3. Leaf-wise Growth (Lá lá cây quyết định): LightGBM sử dụng cách tiếp cận cây quyết định theo hướng lá lá (leaf-wise), nghĩa là nó sẽ phát triển cây theo chiều sâu, chọn lá cây có giảm mất mát lớn nhất để phân chia tiếp theo.
4. Xử lý dữ liệu lớn và tập trung: LightGBM được tối ưu hóa cho các bài toán với tập dữ liệu lớn và có khả năng xử lý các bài toán học máy có dữ liệu phân loại nhiều.
5. Tùy biến linh hoạt: LightGBM hỗ trợ nhiều tùy chọn tùy chỉnh, bao gồm các hàm mất mát (loss functions), các tham số học (learning parameters), và các phương pháp tối ưu (optimization methods).

1.12 Mô hình MLP

1.12.1 Khái niệm

Mô hình MLP (Multi-Layer Perceptron) là một loại mạng nơ-ron nhân tạo có cấu trúc đa tầng, nổi bật trong lĩnh vực học sâu (deep learning). Đây là một trong những kiến trúc mạng nơ-ron phổ biến và mạnh mẽ nhất hiện nay, được sử dụng rộng rãi cho các bài toán học máy phức tạp như phân loại và dự đoán.

MLP là một công cụ mạnh mẽ trong học sâu và đã được áp dụng thành công trong nhiều lĩnh vực, từ nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên cho đến

dự đoán thời gian thực và các bài toán phân loại phức tạp. Tuy nhiên, để huấn luyện hiệu quả, MLP đòi hỏi một lượng lớn dữ liệu và thời gian tính toán.

1.12.2 Thuật toán

Thuật toán MLP (Multi-Layer Perceptron) là một trong những thuật toán quan trọng trong lĩnh vực học sâu (deep learning), được sử dụng rộng rãi trong các bài toán phân loại và dự đoán. Dưới đây là các bước cơ bản của thuật toán MLP:

1. Khởi tạo mạng nơ-ron: MLP bao gồm một số lượng lớn các nơ-ron được tổ chức thành các tầng (layers). Các tầng bao gồm tầng đầu vào (input layer), các tầng ẩn (hidden layers), và tầng đầu ra (output layer).
2. Học và lan truyền ngược (Backpropagation):
 - Lan truyền ngược: Thuật toán này được sử dụng để tính toán độ lỗi giữa đầu ra dự đoán và đầu ra thực tế, sau đó điều chỉnh trọng số của các kết nối nơ-ron theo hướng giảm thiểu độ lỗi này.
 - Thuật toán lan truyền ngược sử dụng gradient descent để cập nhật trọng số của mô hình mạng nơ-ron, tối ưu hóa hàm mất mát.
3. Hàm kích hoạt (Activation function): Các nơ-ron trong các tầng ẩn của MLP thường có hàm kích hoạt phi tuyến tính như ReLU (Rectified Linear Unit), Sigmoid, hay Tanh. Hàm này quyết định giá trị đầu ra của mỗi nơ-ron dựa trên tổng trọng số của các kết nối đầu vào.
4. Đào tạo mô hình: Quá trình huấn luyện mô hình MLP bao gồm lặp lại các bước lan truyền tiến (forward propagation) để tính toán đầu ra của mạng, sau đó sử dụng lan truyền ngược để điều chỉnh trọng số. Quá trình này tiếp tục cho đến khi mô hình hội tụ đủ độ chính xác mong muốn.
5. Độ phức tạp: MLP có thể học được các mô hình phức tạp nhờ vào cấu trúc đa tầng và khả năng học các đặc trưng phi tuyến tính.

Chương 2

Bài toán Dự đoán hành vi mua hàng của khách hàng

2.1 Phân tích bài toán

2.1.1 Bài toán nghiệp vụ

Mô tả bài toán

Bài toán này nhằm mục đích dự báo hành vi mua hàng của khách hàng trong tương lai dựa trên dữ liệu lịch sử. Việc dự báo chính xác hành vi mua hàng sẽ giúp các doanh nghiệp đưa ra quyết định chiến lược, tối ưu hóa các chiến dịch tiếp thị, quản lý tồn kho hiệu quả, và tăng cường trải nghiệm khách hàng. Qua đó, doanh nghiệp có thể tối ưu hóa lợi nhuận và giảm thiểu rủi ro.

Yêu cầu nghiệp vụ

- Thu thập dữ liệu
 - Thu thập thông tin khảo sát hành vi người tiêu dùng.
 - Bao gồm các thông tin về dữ liệu hành vi trực tuyến, dữ liệu giao dịch, dữ liệu khách hàng, dữ liệu từ bên ngoài.
- Phân tích Dữ liệu
 - Sử dụng các phương pháp thống kê để phân tích dữ liệu.
 - Xác định các yếu tố ảnh hưởng đến hành vi mua hàng của khách hàng.
- Dự đoán hành vi mua hàng

- Áp dụng các mô hình thống kê, học sâu để dự đoán hành vi khách hàng.
- Đánh giá:
 - Sử dụng các chỉ số đánh giá: Precision, Recall, F1-score, Support.
 - Đánh giá tổng quan: Accuracy, Macro avg, Weighted avg.

2.1.2 Bài toán kỹ thuật

Mô tả bài toán

Xây dựng và triển khai các mô hình học máy, học sâu, thống kê để dự đoán hành vi mua hàng của khách hàng. Bài toán này sẽ được giải quyết bằng cách sử dụng các kỹ thuật xử lý dữ liệu, thiết kế mô hình, huấn luyện mô hình và đánh giá mô hình.

Yêu cầu kỹ thuật

- Chuẩn bị dữ liệu
 - Thu thập dữ liệu hành vi tiêu dùng của khách hàng.
 - Tiền xử lý dữ liệu bao gồm: loại bỏ các giá trị null, loại bỏ các trường không cần thiết, chuẩn hóa dữ liệu, trích xuất đặc trưng.
- Thiết kế và xây dựng mô hình
 - Sử dụng các thư viện để xây dựng các mô hình deep learning: TensorFlow, Keras, Pytorch.
- Huấn luyện mô hình
 - Chia dữ liệu thành tập huấn luyện và tập kiểm tra.
 - Sử dụng các kỹ thuật tối ưu hóa để huấn luyện mô hình.
- Đánh giá và tinh chỉnh mô hình
 - Đánh giá mô hình trên tập kiểm tra bằng các chỉ số: Precision, Recall, F1-score, Support; Đánh giá chỉ số tổng quan: Accuracy, Macro avg, Weighted avg.
 - Tinh chỉnh các tham số hyperparameters của mô hình để cải thiện độ chính xác.
- Triển khai mô hình
 - Đóng gói các mô hình đã huấn luyện.
 - Kịch bản demo ứng dụng.

2.2 Mô hình đề xuất

- Mô hình hồi quy logistic.
- Mô hình SVC.
- Mô hình KNN.
- Mô hình Gaussian NB.
- Mô hình cây quyết định.
- Mô hình rừng ngẫu nhiên.
- Mô hình Gradient Boosting.
- Mô hình XGBoost.
- Mô hình CatBoost.
- Mô hình Light Gradient Boosting.
- Bộ phân loại Perceptron đa tầng.

Chương 3

Thực nghiệm và kết quả

3.1 Dữ liệu

3.1.1 Mô tả bộ dữ liệu

Báo cáo sử dụng hai bộ dữ liệu khác nhau để triển khai trên các mô hình thực nghiệm. Thông tin chi tiết hai bộ dữ liệu như sau:

- Bộ dữ liệu 1: hành vi người tiêu dùng Amazon:

Bộ dữ liệu gồm 22 thuộc tính liên quan đến nhân khẩu học của khách hàng, tương tác của người dùng và đánh giá. Bộ dữ liệu này nhằm mục đích cung cấp thông tin chuyên sâu về sở thích, thói quen mua sắm và quy trình ra quyết định của khách hàng trên nền tảng Amazon. Chi tiết các thuộc tính như sau:

1. Timestamp: Dấu thời gian dữ liệu được ghi lại
2. age: Tuổi của khách hàng
3. Gender: Giới tính của khách hàng
4. Purchase_Frequency: Tần suất mua hàng của khách hàng
5. Purchase_Categories: Loại sản phẩm khách hàng thường mua
6. Personalized_Recommendation_Frequency: Khách hàng đã bao giờ mua hàng dựa trên đề xuất sản phẩm được cá nhân hóa từ Amazon chưa?
7. Browsing_Frequency: Tần suất khách hàng vào web hoặc ứng dụng Amazon
8. Product_Search_Method: Phương thức tìm kiếm của khách hàng
9. Search_Result_Exploration: Khách hàng có xu hướng khám phá nhiều trang kết quả tìm kiếm hay tập trung vào trang đầu tiên?

10. Customer_Reviews_Importance: Đánh giá của khách hàng khác quan trọng như thế nào trong quá trình ra quyết định của khách hàng?
11. Add_to_Cart_Browsing: Khách hàng có thêm sản phẩm vào giỏ hàng của mình khi duyệt trên Amazon không?
12. Cart_Completion_Frequency: Khách hàng có thường xuyên hoàn tất việc mua hàng sau khi thêm sản phẩm vào giỏ hàng của mình không?
13. Saveforlater_Frequency: Khách hàng có sử dụng tính năng "Save for Later" của Amazon không và nếu có thì tần suất như thế nào?
14. Review_Left: Khách hàng đã bao giờ để lại đánh giá sản phẩm trên Amazon chưa?
15. Review_Reliability: Khách hàng dựa vào đánh giá sản phẩm đến mức nào khi mua hàng?
16. Review_Helpfulness: Khách hàng có tìm thấy thông tin hữu ích từ đánh giá của khách hàng khác không?
17. Personalized_recommendation_Frequency: Khách hàng có thường xuyên nhận được đề xuất sản phẩm được cá nhân hóa từ Amazon không?
18. Recommendation_Helpfulness: Khách hàng có thấy những đề xuất này hữu ích không?
19. Rating_Accuracy: Khách hàng đánh giá mức độ liên quan và chính xác của các đề xuất khách hàng nhận được như thế nào?
20. Shopping_Satisfaction: Khách hàng hài lòng đến mức nào với trải nghiệm mua sắm tổng thể của mình trên Amazon?
21. Service_Appregation: Khách hàng đánh giá cao khía cạnh nào của dịch vụ Amazon nhất?
22. Improvement_Areas: Có lĩnh vực nào Khách hàng nghĩ Amazon có thể cải thiện không?

Dựa vào hành vi tiêu dùng của khách hàng, đưa ra kết quả khách hàng có mua sản phẩm hay không với đầu ra là 0, 1, 2 (trong đó, 0 là không mua, 1 là cân nhắc, 2 là có mua).

- Bộ dữ liệu 2: hành vi người mua hàng trực tuyến:

Bộ dữ liệu gồm 12330 hàng tương ứng với 12330 phiên khách hàng, 18 cột. Bài toán sẽ sử dụng thuộc tính "Revenue" (True hay False) làm biến mục tiêu, 17 thuộc tính còn lại sẽ là biến dự đoán, thuộc tính hoặc đặc điểm của khách hàng. Tập dữ liệu này được hình thành trong khoảng thời gian 1

năm không bao gồm các ngày lễ, ngày đặc biệt hoặc các chiến dịch cụ thể. Kích thước của tập dữ liệu đủ để xây dựng các mô hình. Dưới đây là chi tiết các thuộc tính:

1. Administrative: Số trang được khách truy cập truy cập về quản lý tài khoản.
2. Administrative_Duration: Tổng lượng thời gian (tính bằng giây) mà khách truy cập dành cho các trang liên quan đến quản lý tài khoản.
3. Informational: Số trang được khách truy cập truy cập về trang Web, thông tin liên lạc và địa chỉ thông tin website mua sắm.
4. Informational_Duration: Tổng lượng thời gian (tính bằng giây) mà khách truy cập dành cho các trang thông tin.
5. ProductRelated: Số trang được khách truy cập truy cập về các trang liên quan đến sản phẩm.
6. ProductRelated_Duration: Tổng lượng thời gian (tính bằng giây) mà khách truy cập dành cho các trang liên quan đến sản phẩm.
7. BounceRates: Giá trị tỷ lệ thoát trung bình của các trang được khách truy cập truy cập.
8. ExitRates: Giá trị tỷ lệ thoát trung bình của các trang được khách truy cập.
9. PageValues: Giá trị trang trung bình của các trang được khách truy cập.
10. SpecialDay: Thời điểm tham quan địa điểm sắp đến một ngày đặc biệt.
11. Month: tháng truy cập.
12. OperatingSystems: hệ điều hành của khách truy cập.
13. Browser: trình duyệt khách dùng để truy cập.
14. Region: khu vực địa lý mà khách truy cập đến trang Web.
15. TrafficType: nguồn lưu lượng truy cập mà khách truy cập đến trang Web.
16. VisitorType: loại khách truy cập là "khách truy cập mới", "khách truy cập quay lại" và "khác".
17. Weekend: giá trị Boolean cho biết ngày truy cập có phải là cuối tuần hay không.
18. Revenue: nhãn lớp cho biết liệu lượt truy cập đã được hoàn tất bằng một giao dịch hay chưa.

3.1.2 Trực quan hóa và khám phá dữ liệu

Khám phá dữ liệu là một giai đoạn quan trọng trong quy trình xây dựng mô hình dữ liệu nhằm giúp hiểu rõ hơn về bộ dữ liệu thực nghiệm, qua đó xác định các vấn đề và các biện pháp cần thiết để làm sạch và tiền xử lý dữ liệu. Dưới đây là kết quả trực quan hóa và khám phá dữ liệu của hai bộ dữ liệu:

Bộ dữ liệu 1:

1. Hiển thị thông tin tổng quan của bộ dữ liệu:

```
RangeIndex: 602 entries, 0 to 601
Data columns (total 24 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Timestamp                             602 non-null    object
1   age                                    602 non-null    int64
2   Gender                                602 non-null    object
3   Purchase_Frequency                    602 non-null    object
4   Purchase_Categories                    602 non-null    object
5   Personalized_Recommendation_Frequency 602 non-null    object
6   Browsing_Frequency                    602 non-null    object
7   Product_Search_Method                  600 non-null    object
8   Search_Result_Exploration              602 non-null    object
9   Customer_Reviews_Importance            602 non-null    int64
...
22  Improvement_Areas                      602 non-null    object
23  Timestamp                             602 non-null    datetime64[ns, UTC-05:30]
dtypes: datetime64[ns, UTC-05:30](1), int64(5), object(18)
memory usage: 113.0+ KB
```

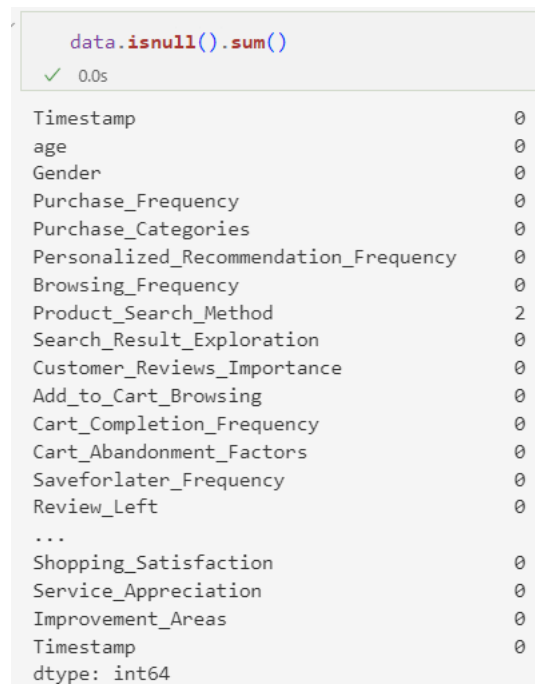
Hình 3.1: Tổng quan bộ dữ liệu

2. Thông tin thống kê của một số thuộc tính có kiểu dữ liệu số trong bộ dữ liệu được thể hiện trong hình dưới đây:

	age	Customer_Reviews_Importance	Personalized_Recommendation_Frequency	Rating_Accuracy	Shopping_Satisfaction
count	602.000000	602.000000	602.000000	602.000000	602.000000
mean	30.790698	2.480066	2.699336	2.672757	2.463455
std	10.193276	1.185226	1.042028	0.899744	1.012152
min	3.000000	1.000000	1.000000	1.000000	1.000000
25%	23.000000	1.000000	2.000000	2.000000	2.000000
50%	26.000000	3.000000	3.000000	3.000000	2.000000
75%	36.000000	3.000000	3.000000	3.000000	3.000000
max	67.000000	5.000000	5.000000	5.000000	5.000000

Hình 3.2: Thông tin thống kê

3. Số giá trị null của từng cột:



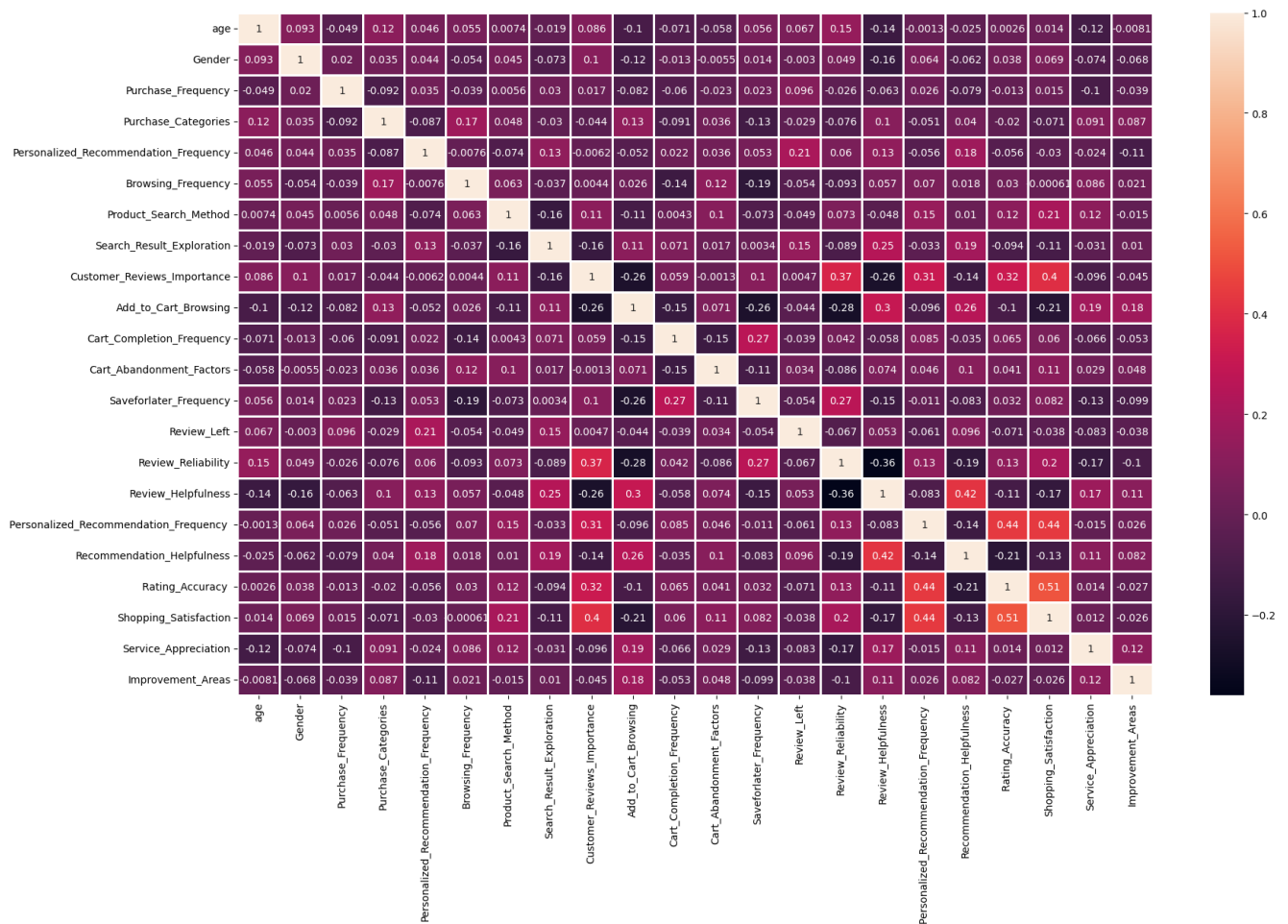
```
data.isnull().sum()
```

Timestamp	0
age	0
Gender	0
Purchase_Frequency	0
Purchase_Categories	0
Personalized_Recommendation_Frequency	0
Browsing_Frequency	0
Product_Search_Method	2
Search_Result_Exploration	0
Customer_Reviews_Importance	0
Add_to_Cart_Browsing	0
Cart_Completion_Frequency	0
Cart_Abandonment_Factors	0
Saveforlater_Frequency	0
Review_Left	0
...	
Shopping_Satisfaction	0
Service_Appreciation	0
Improvement_Areas	0
Timestamp	0

dtype: int64

Hình 3.3: Số giá trị null từng cột

4. Để dễ dàng cho việc khám phá dữ liệu và xử lý dữ liệu, ta thực hiện chuyển đổi các biến phân loại thành các giá trị số và phân loại tuổi thành các nhóm tuổi
5. Vẽ biểu đồ heatmap:

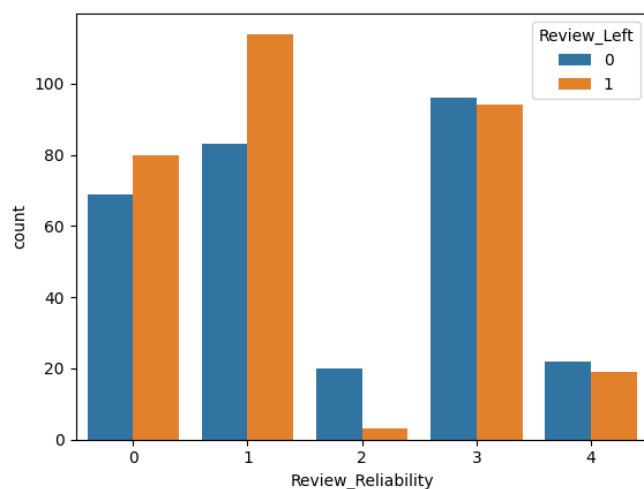


Hình 3.4: Biểu đồ heatmap thể hiện độ tương quan giữa các biến

Từ đây ta thấy:

- Browsing_Frequency và Shopping_Satisfaction có tương quan thấp
- Rating_Accuracy và Shopping_Satisfaction có tương quan cao

6. Biểu đồ thống kê Review:



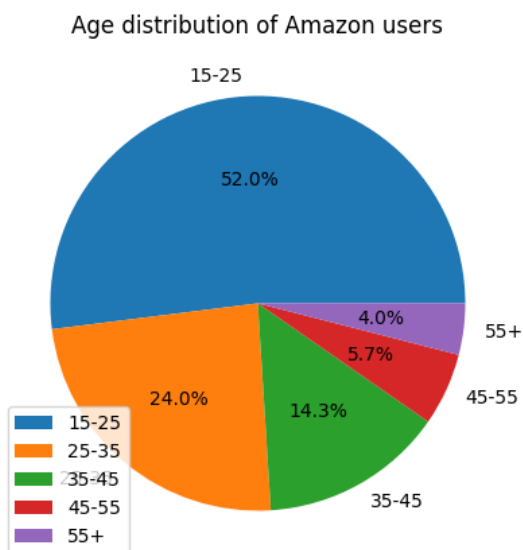
Hình 3.5: Biểu đồ thống kê Review

0 đại diện cho những người chưa để lại đánh giá nào và 1 đại diện cho những người đã để lại đánh giá

những người tiếp tục đánh giá có phạm vi "Rất hài lòng", "Hài lòng", "Trung bình", "Không hài lòng" và "Rất không hài lòng"

Dựa trên biểu đồ, có vẻ như những cá nhân hài lòng với sản phẩm hoặc dịch vụ có nhiều khả năng để lại đánh giá hơn. Biểu đồ cho thấy những người đánh giá trải nghiệm của họ là "Rất hài lòng" hoặc "Hài lòng" có tần suất để lại đánh giá cao hơn. Mặt khác, những cá nhân không hài lòng hoặc có ý kiến trung bình sẽ ít để lại đánh giá hơn.

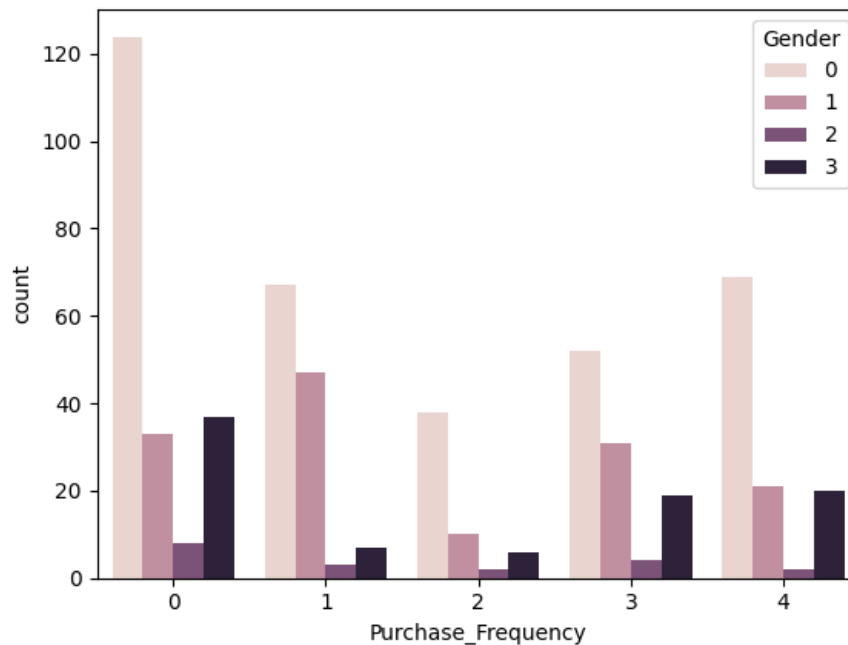
7. Biểu đồ phân loại khách hàng theo nhóm tuổi:



Hình 3.6: Biểu đồ thống kê tần suất mua hàng theo từng giới tính

Dựa trên biểu đồ, có thể suy ra rằng các cá nhân trong độ tuổi 0-20 có mức độ tương tác người dùng cao hơn so với các nhóm tuổi khác. Điều này ngụ ý rằng các cá nhân trong độ tuổi này tham gia nhiều hơn

8. Biểu đồ thống kê tần suất mua hàng theo từng giới tính:



Hình 3.7: Biểu đồ thống kê tần suất mua hàng theo từng giới tính

0 đại diện cho những người xác định là nữ, 1 đại diện cho những người xác định là nam, 2 đại diện cho những người không muốn nói về giới tính của mình, 3 đại diện cho những người xác định thuộc giới tính khác. Tần suất duyệt web được phân loại là "Nhiều lần trong ngày", "Nhiều lần một tuần", "Vài lần một tháng", "Mỗi tháng một lần" và "Ít hơn một lần một tháng". Dân số nữ ở đây tích cực mua hàng hơn các giới tính khác

Bộ dữ liệu 2:

1. Tổng quan bộ dữ liệu

```

RangeIndex: 12330 entries, 0 to 12329
Data columns (total 18 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Administrative                        12330 non-null  int64
1   Administrative_Duration              12330 non-null  float64
2   Informational                        12330 non-null  int64
3   Informational_Duration              12330 non-null  float64
4   ProductRelated                      12330 non-null  int64
5   ProductRelated_Duration            12330 non-null  float64
6   BounceRates                         12330 non-null  float64
7   ExitRates                           12330 non-null  float64
8   PageValues                          12330 non-null  float64
9   SpecialDay                          12330 non-null  float64
10  Month                               12330 non-null  object
11  OperatingSystems                    12330 non-null  int64
12  Browser                             12330 non-null  int64
13  Region                             12330 non-null  int64
14  TrafficType                         12330 non-null  int64
15  VisitorType                         12330 non-null  object
16  Weekend                             12330 non-null  bool
17  Revenue                             12330 non-null  bool
dtypes: bool(2), float64(7), int64(7), object(2)
memory usage: 1.5+ MB

```

Hình 3.8: Tổng quan bộ dữ liệu

Bộ dữ liệu 2 gồm 12330 hàng, 18 cột.

```

Out[66]:
Administrative      0
Administrative_Duration  0
Informational       0
Informational_Duration  0
ProductRelated     0
ProductRelated_Duration  0
BounceRates        0
ExitRates          0
PageValues         0
SpecialDay         0
Month              0
OperatingSystems   0
Browser            0
Region            0
TrafficType       0
VisitorType       0
Weekend           0
Revenue           0
dtype: int64

```

Hình 3.9: Hiển thị giá trị null

Bộ dữ liệu không có các giá trị null.

	count	unique	top	freq	mean	std	min	25%	50%
Administrative	12330.0	NaN	NaN	NaN	2.315166	3.321784	0.0	0.0	1.0
Active_Duration	12330.0	NaN	NaN	NaN	80.818611	176.779107	0.0	0.0	7.5
Informational	12330.0	NaN	NaN	NaN	0.503569	1.270156	0.0	0.0	0.0
Informational_Duration	12330.0	NaN	NaN	NaN	34.472398	140.749294	0.0	0.0	0.0
ProductRelated	12330.0	NaN	NaN	NaN	31.731468	44.475503	0.0	7.0	18.0
ProductRelated_Duration	12330.0	NaN	NaN	NaN	1194.74622	1913.669288	0.0	184.1375	598.936905
BounceRates	12330.0	NaN	NaN	NaN	0.022191	0.048488	0.0	0.0	0.003112
ExitRates	12330.0	NaN	NaN	NaN	0.043073	0.048597	0.0	0.014286	0.025156
PageValues	12330.0	NaN	NaN	NaN	5.889258	18.568437	0.0	0.0	0.0
SpecialDay	12330.0	NaN	NaN	NaN	0.061427	0.198917	0.0	0.0	0.0
Month	12330	10	May	3364	NaN	NaN	NaN	NaN	NaN
OperatingSystems	12330.0	NaN	NaN	NaN	2.124006	0.911325	1.0	2.0	2.0
Browser	12330.0	NaN	NaN	NaN	2.357097	1.717277	1.0	2.0	2.0
Region	12330.0	NaN	NaN	NaN	3.147364	2.401591	1.0	1.0	3.0
TrafficType	12330.0	NaN	NaN	NaN	4.069586	4.025169	1.0	2.0	2.0
VisitorType	12330	3	Returning_Visitor	10551	NaN	NaN	NaN	NaN	NaN
Weekend	12330	2	False	9462	NaN	NaN	NaN	NaN	NaN
Revenue	12330	2	False	10422	NaN	NaN	NaN	NaN	NaN

Hình 3.10: Số liệu thống kê mô tả

```

Out[69]:
Administrative          int64
Administrative_Duration float64
Informational           int64
Informational_Duration  float64
ProductRelated          int64
ProductRelated_Duration float64
BounceRates             float64
ExitRates               float64
PageValues              float64
SpecialDay              float64
Month                   object
OperatingSystems        object
Browser                 object
Region                  object
TrafficType             object
VisitorType             object
Weekend                 object
Revenue                 object
dtype: object

```

Hình 3.11: Chuyển đổi dtype từ boolean thành chuỗi.

2. Phân tích biến mục tiêu

```

Revenue
False    10422
True      1908
Name: count, dtype: int64

```

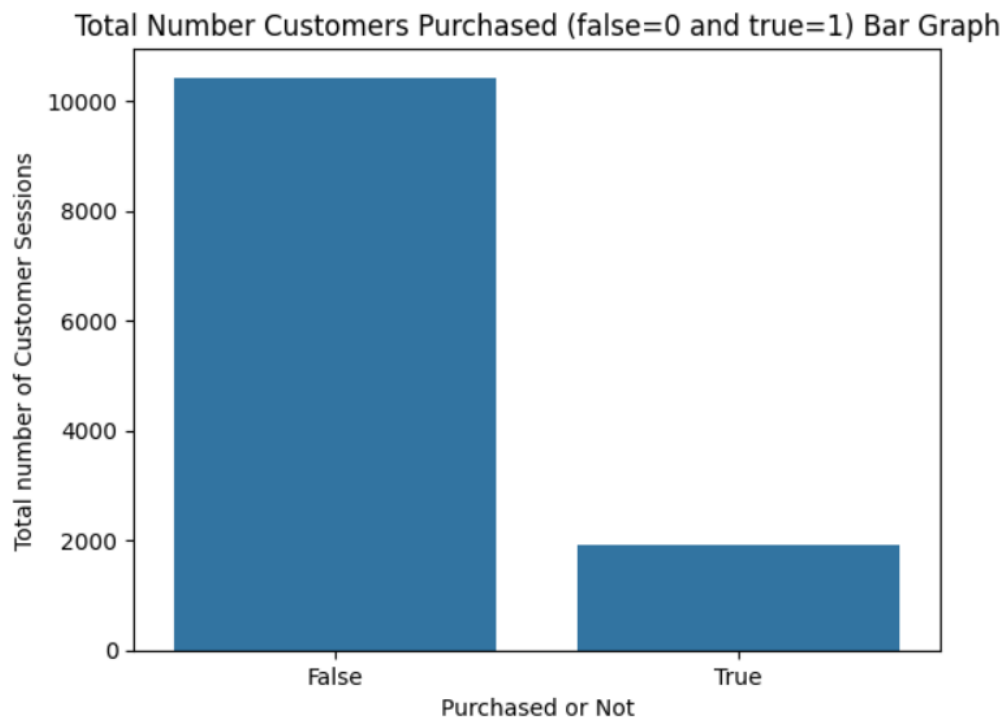
Hình 3.12: Số giá trị của biến mục tiêu

```

Revenue
False    0.845255
True     0.154745
Name: count, dtype: float64

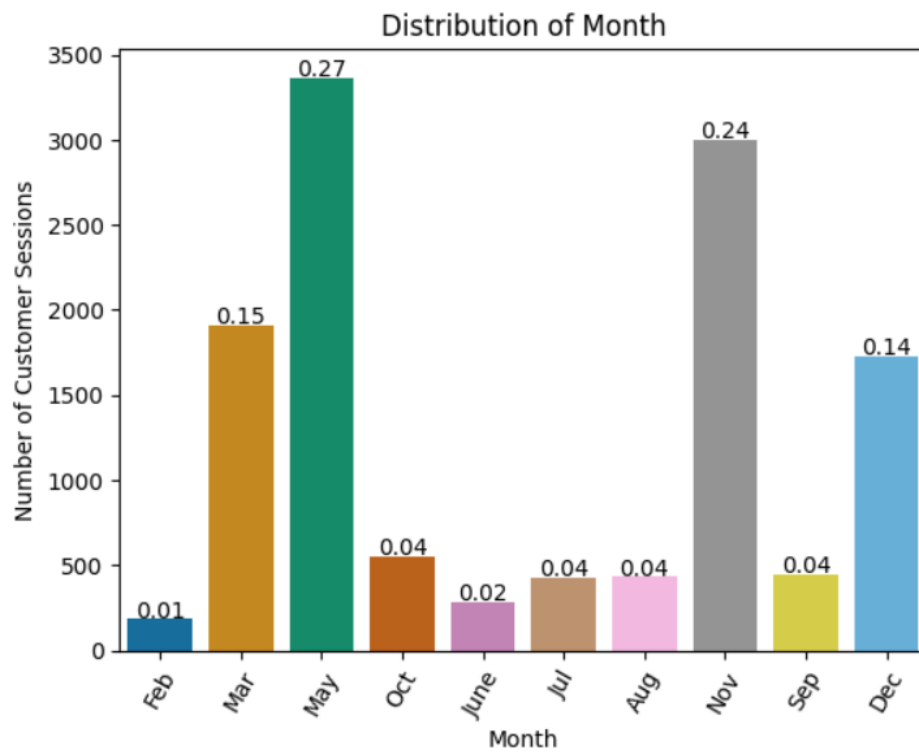
```

Hình 3.13: Phân tích phần trăm của tỷ lệ Revenue.

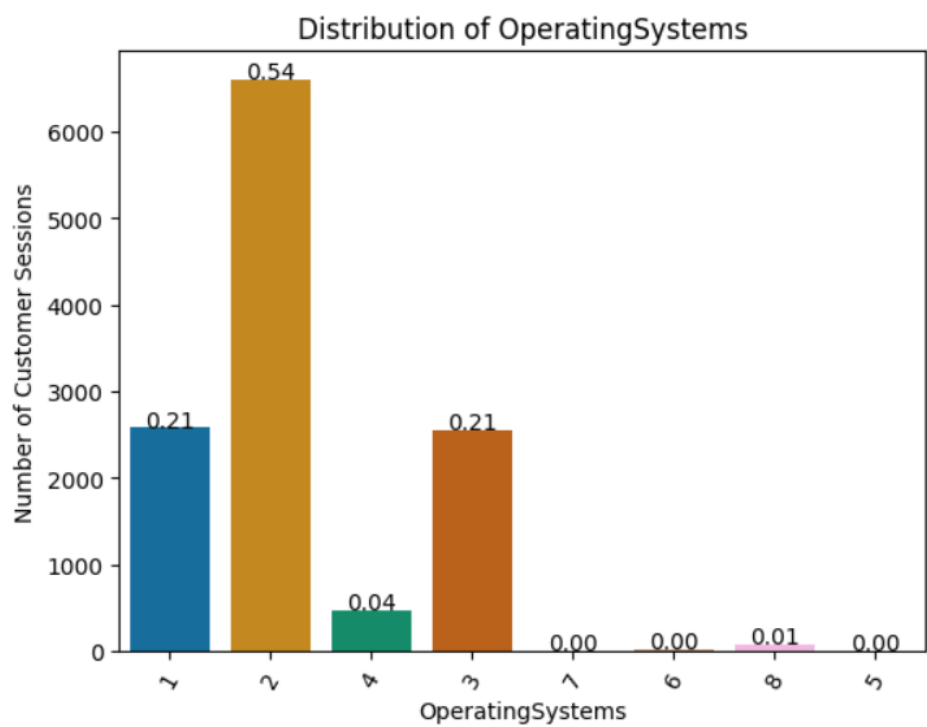


Hình 3.14: Tổng số khách hàng đã không mua và mua.

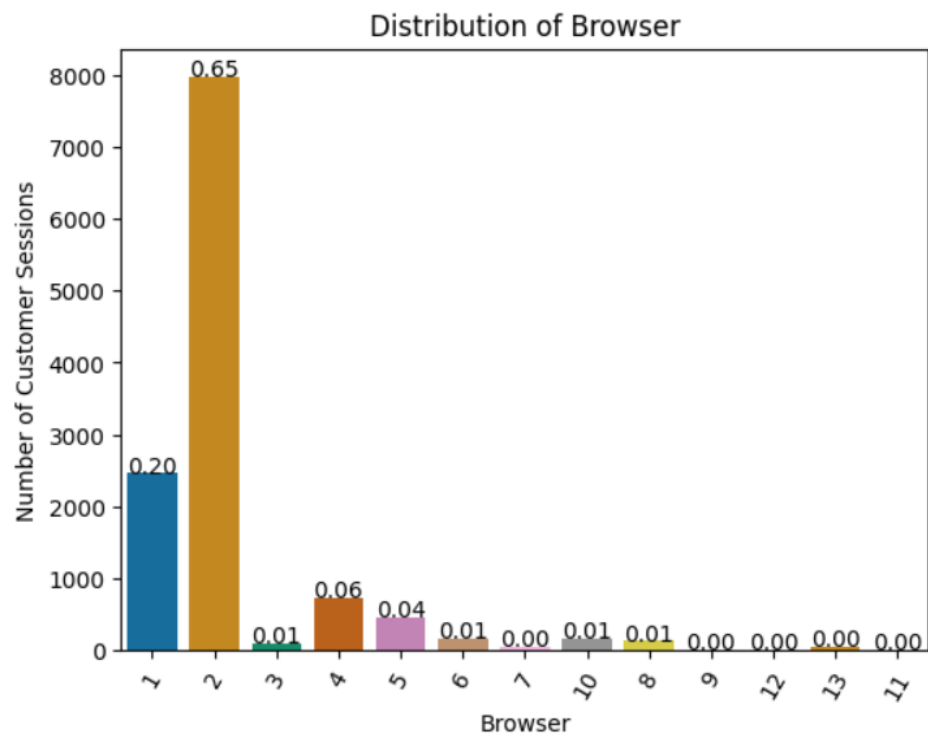
3. Sơ đồ phân phối của các biến phân loại.



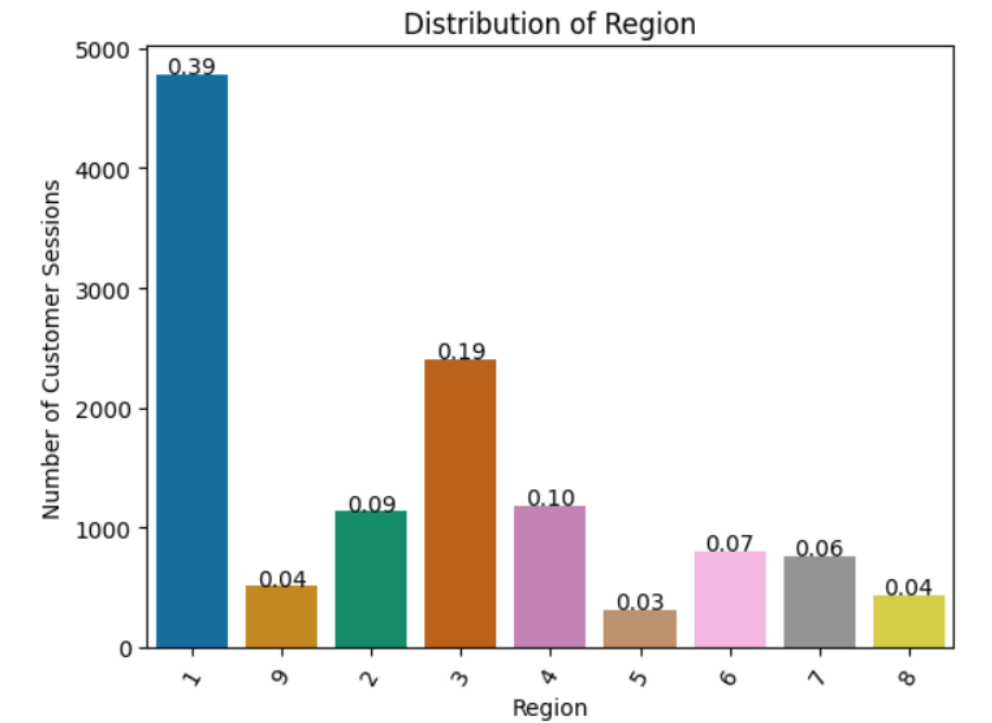
Hình 3.15: Phân phối theo tháng



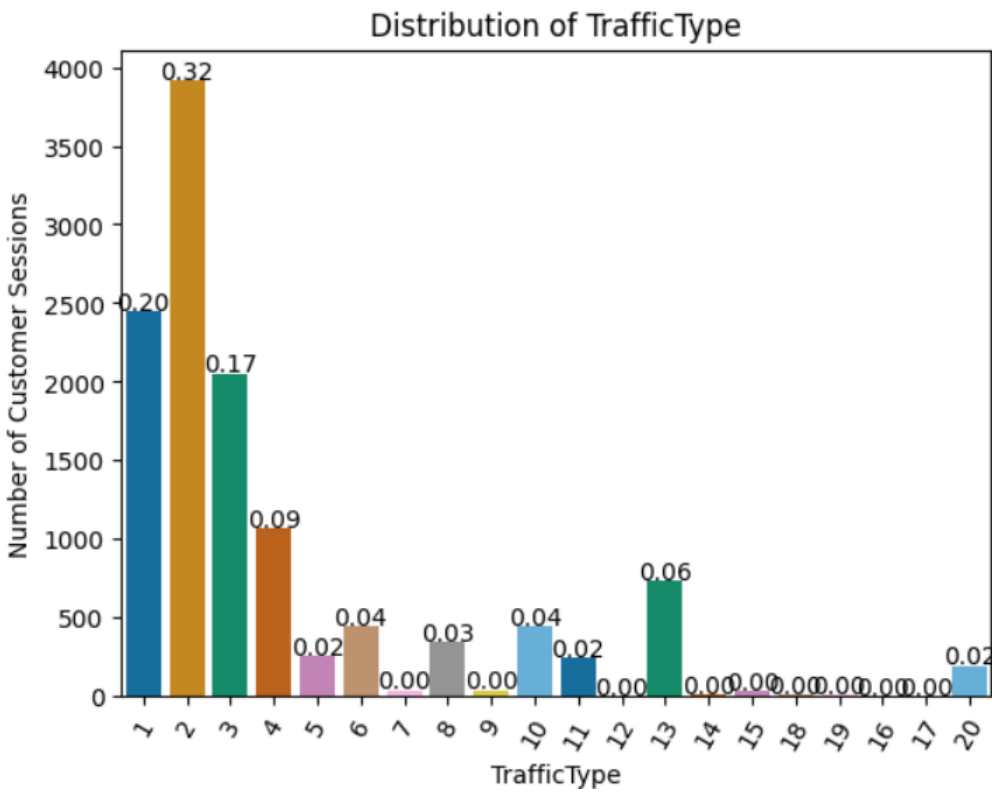
Hình 3.16: Phân phối theo Operating Systems



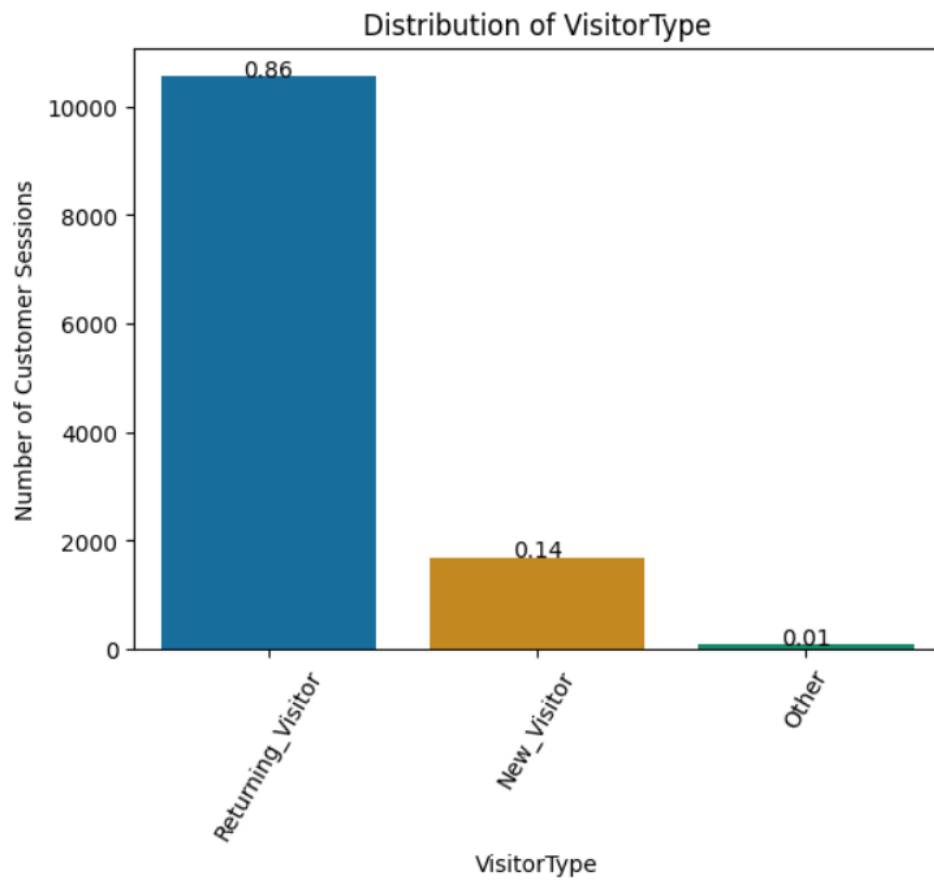
Hình 3.17: Phân phối theo Browser



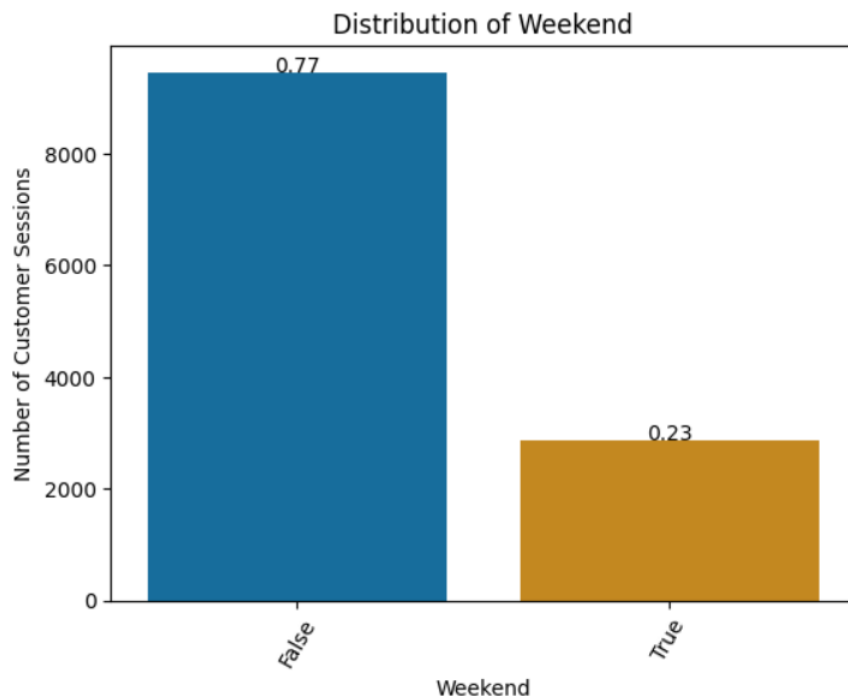
Hình 3.18: Phân phối theo Region



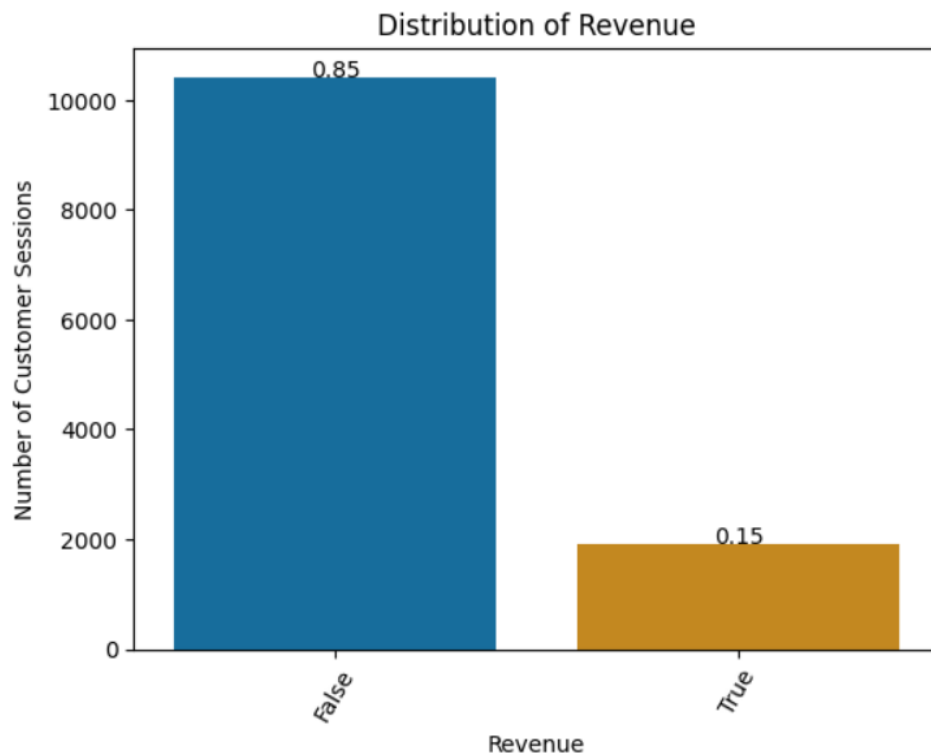
Hình 3.19: Phân phối theo Traffic Type



Hình 3.20: Phân phối theo Visitor Type



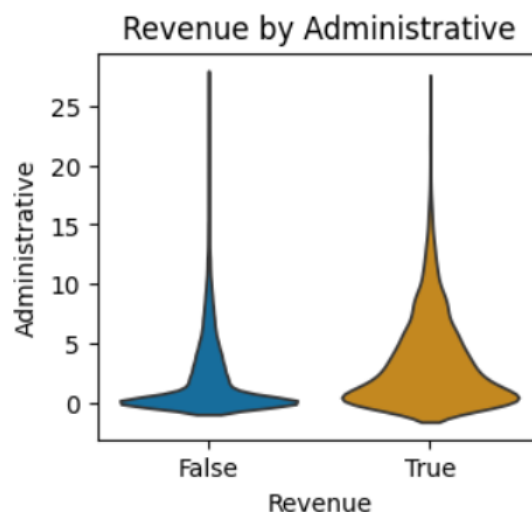
Hình 3.21: Phân phối theo tuần



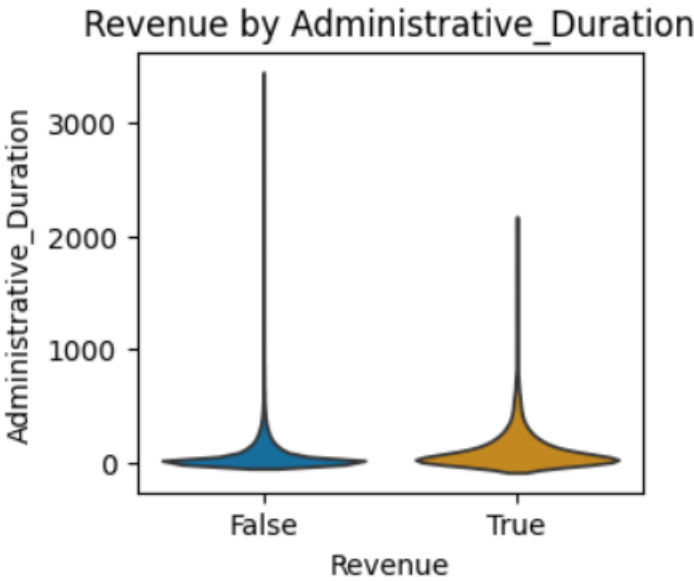
Hình 3.22: Phân phối theo Revenue

Hầu hết khách hàng mua sắm vào tháng 5, tháng 11 và thực hiện hầu hết việc mua sắm trong tuần. Họ sử dụng hệ điều hành 2, trình duyệt 2 và sử dụng loại lưu lượng truy cập 2. Họ sống ở khu vực 1 và là khách hàng cũ. Hầu hết họ không mua bất cứ thứ gì và vào trang web để mua sắm.

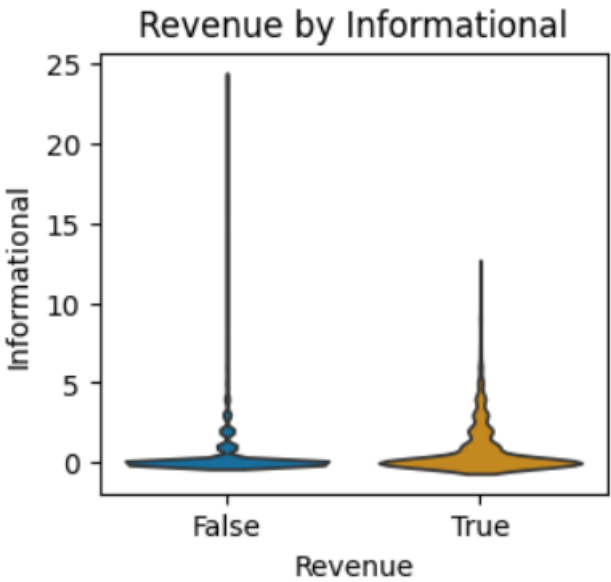
4. Kiểm tra sự phân bố của các biến số bằng cách sử dụng biểu đồ violin.



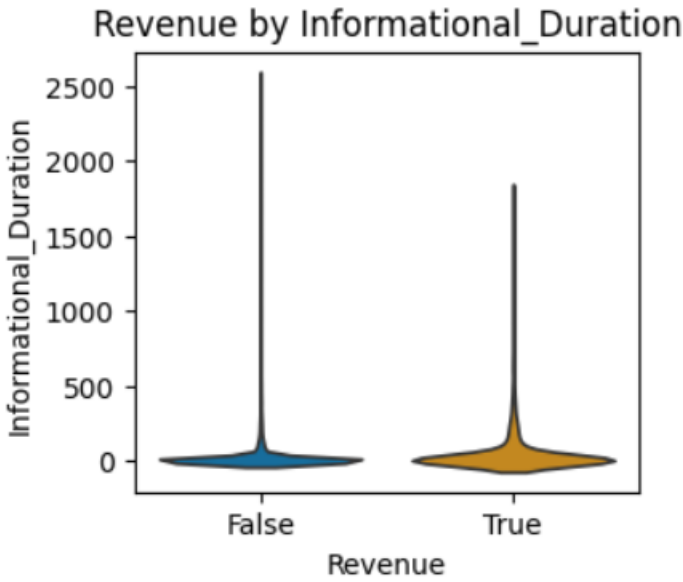
Hình 3.23: Revenue theo Administrative



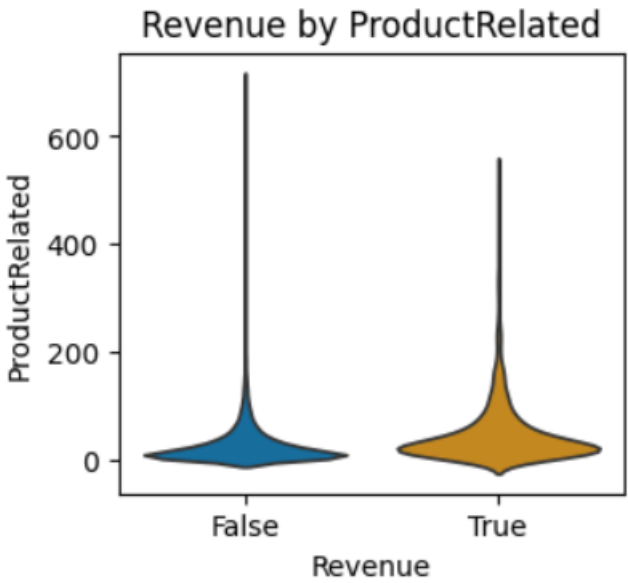
Hình 3.24: Revenue theo Administrative_Duration



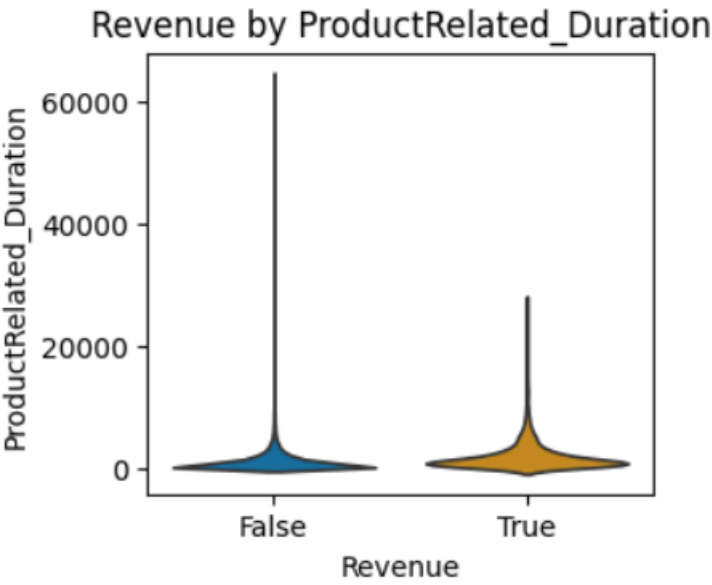
Hình 3.25: Revenue theo Informational



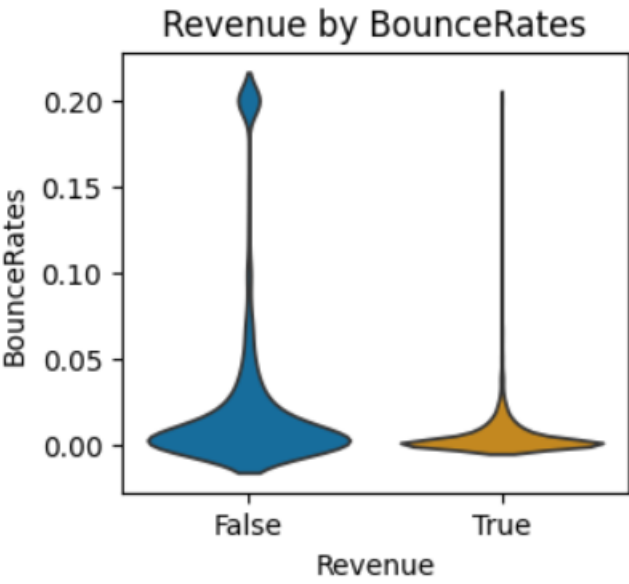
Hình 3.26: Revenue theo Informational_Duration



Hình 3.27: Revenue theo ProductRelated



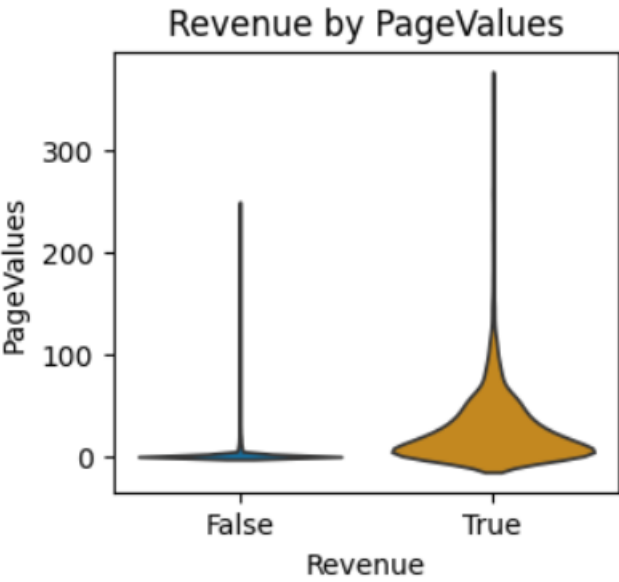
Hình 3.28: Revenue theo ProductRelated_Duration



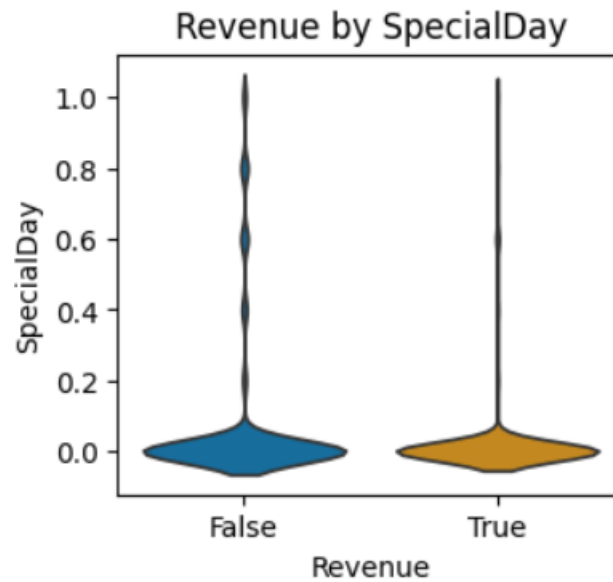
Hình 3.29: Revenue theo BounceRates



Hình 3.30: Revenue theo ExitRates



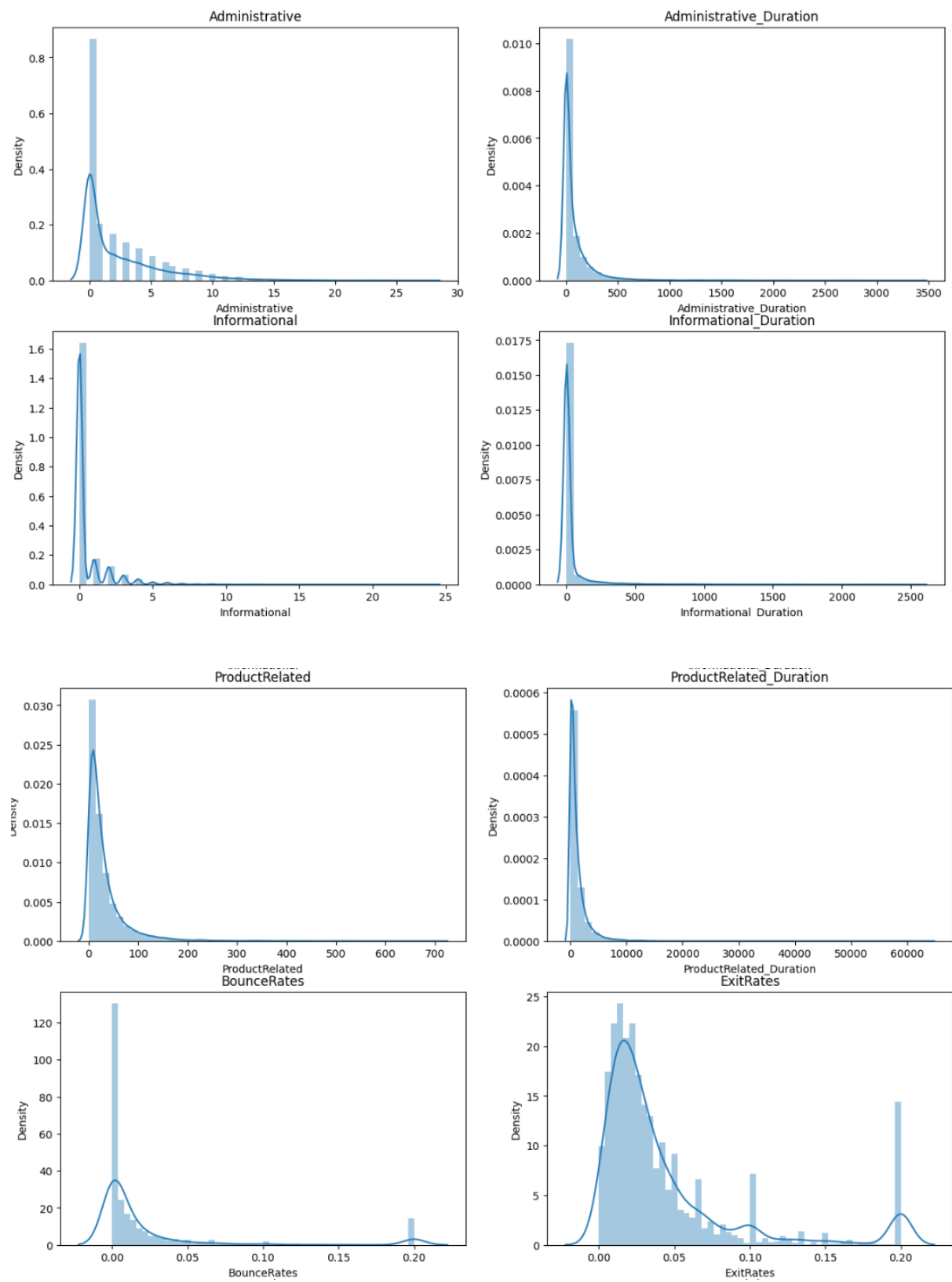
Hình 3.31: Revenue theo PageValues

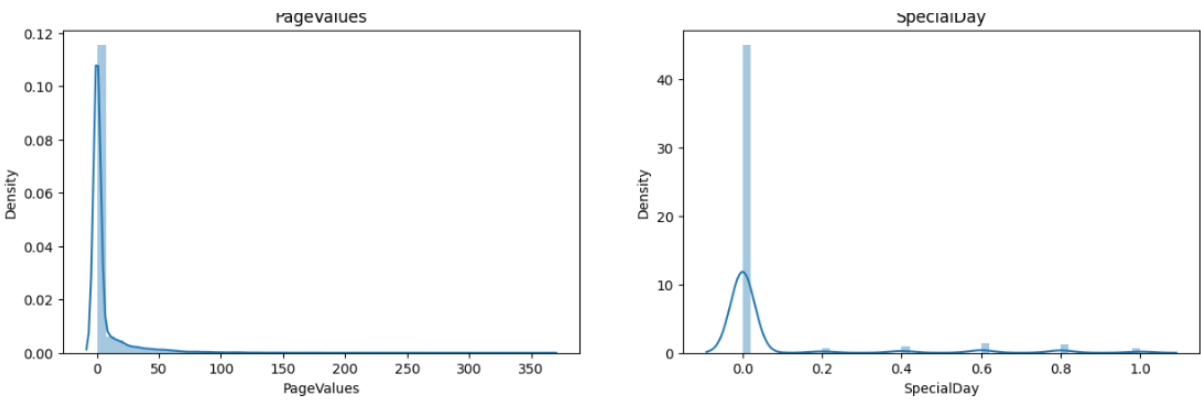


Hình 3.32: Revenue theo SpecialDay

Thông tin chi tiết thu được từ các biểu đồ violin cho thấy số trang được truy cập càng cao thì khách hàng sẽ mua thứ gì đó càng nhiều. Một thông tin chi tiết khác là những khách hàng đã mua một mặt hàng có tỷ lệ thoát và tỷ lệ thoát ngắn hơn so với những khách hàng không mua mặt hàng đó. Một chiến lược để giữ chân khách hàng trên trang web lâu hơn là cung cấp các mặt hàng mà họ muốn mua thông qua hệ thống giới thiệu.

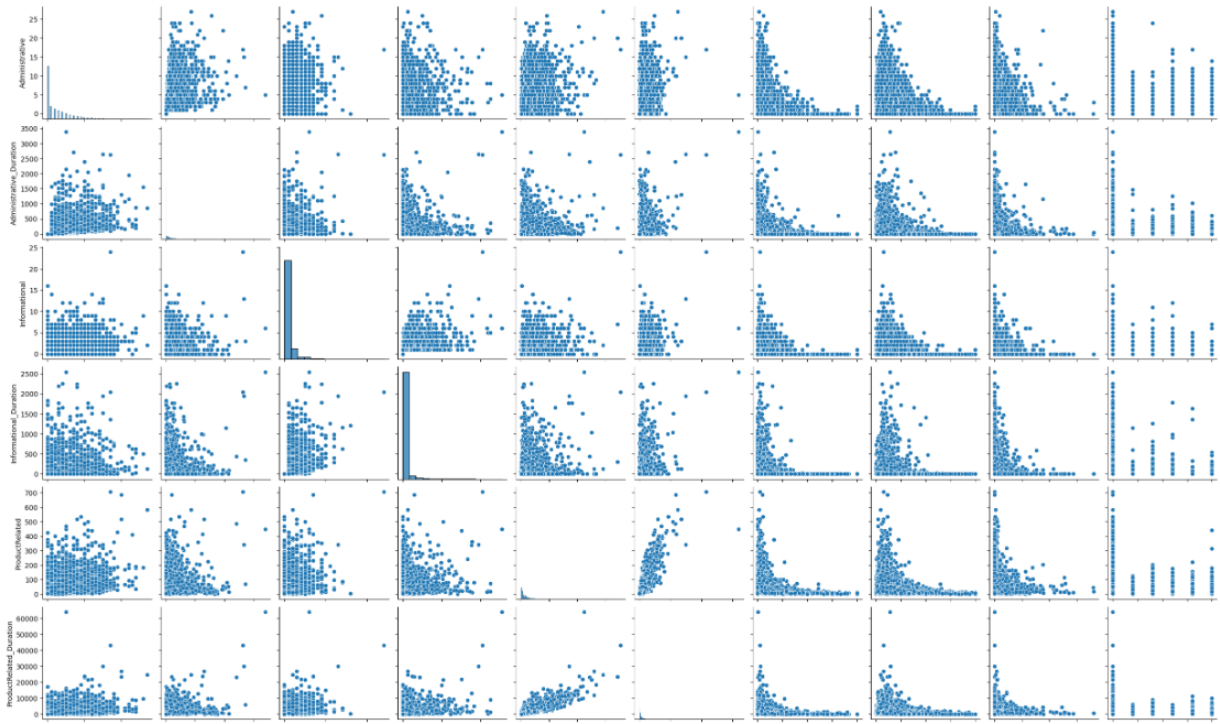
5. Kiểm tra sự phân bố của các biến số.

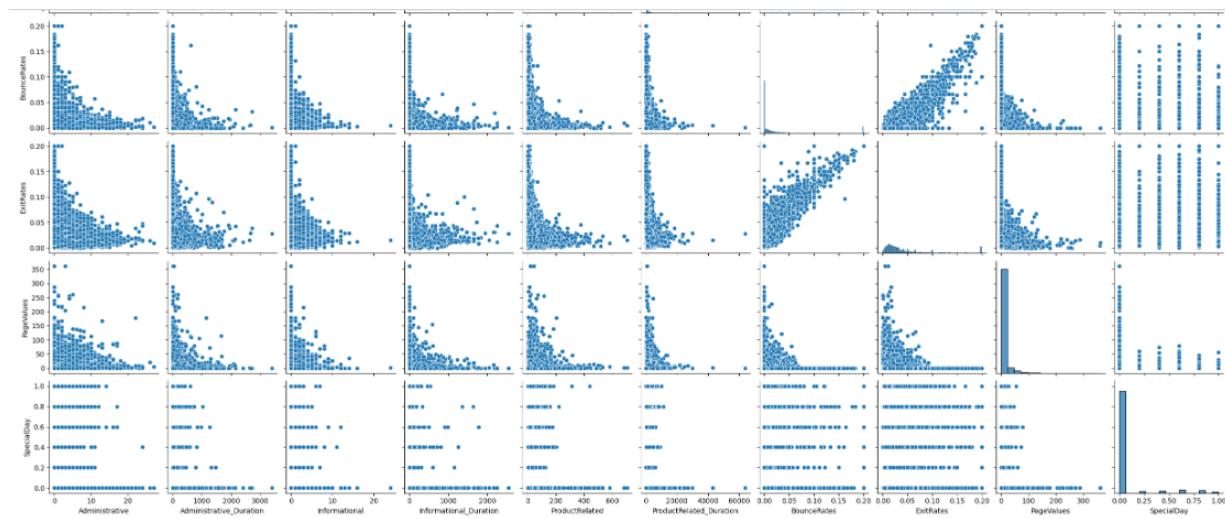




Hình 3.33: Sự phân bố của các biến số

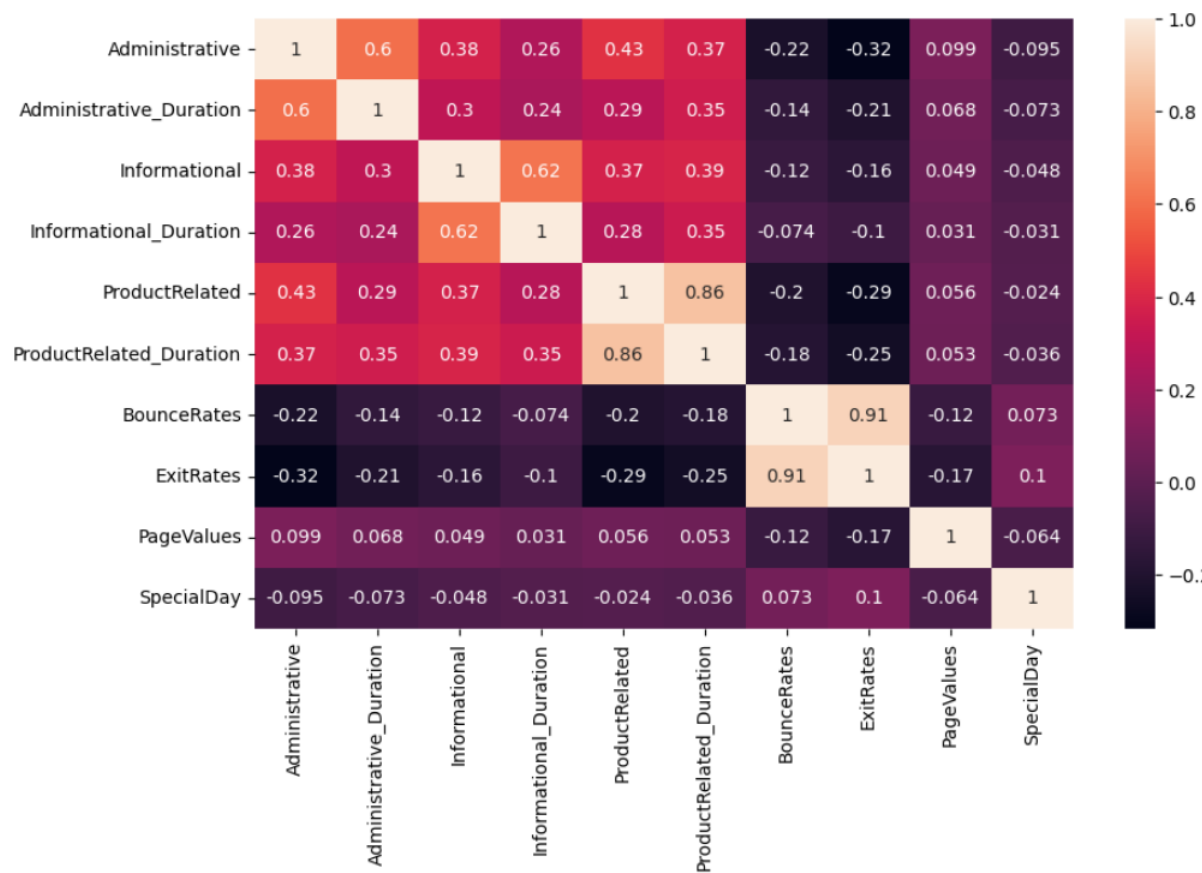
6. Kiểm tra mối quan hệ giữa các biến số.



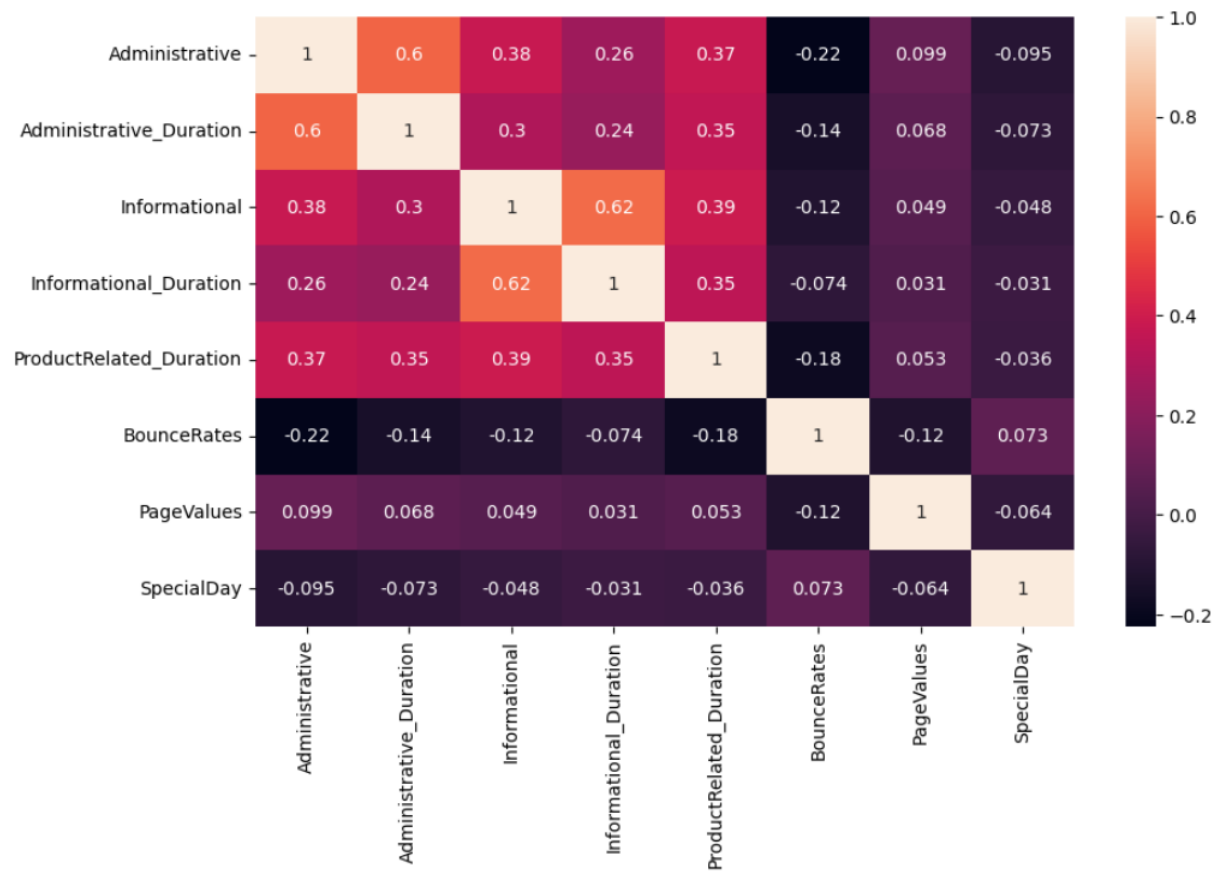


Hình 3.34: Kiểm tra mối quan hệ giữa các biến số.

7. Kiểm tra đa cộng tuyến bằng heatmap.

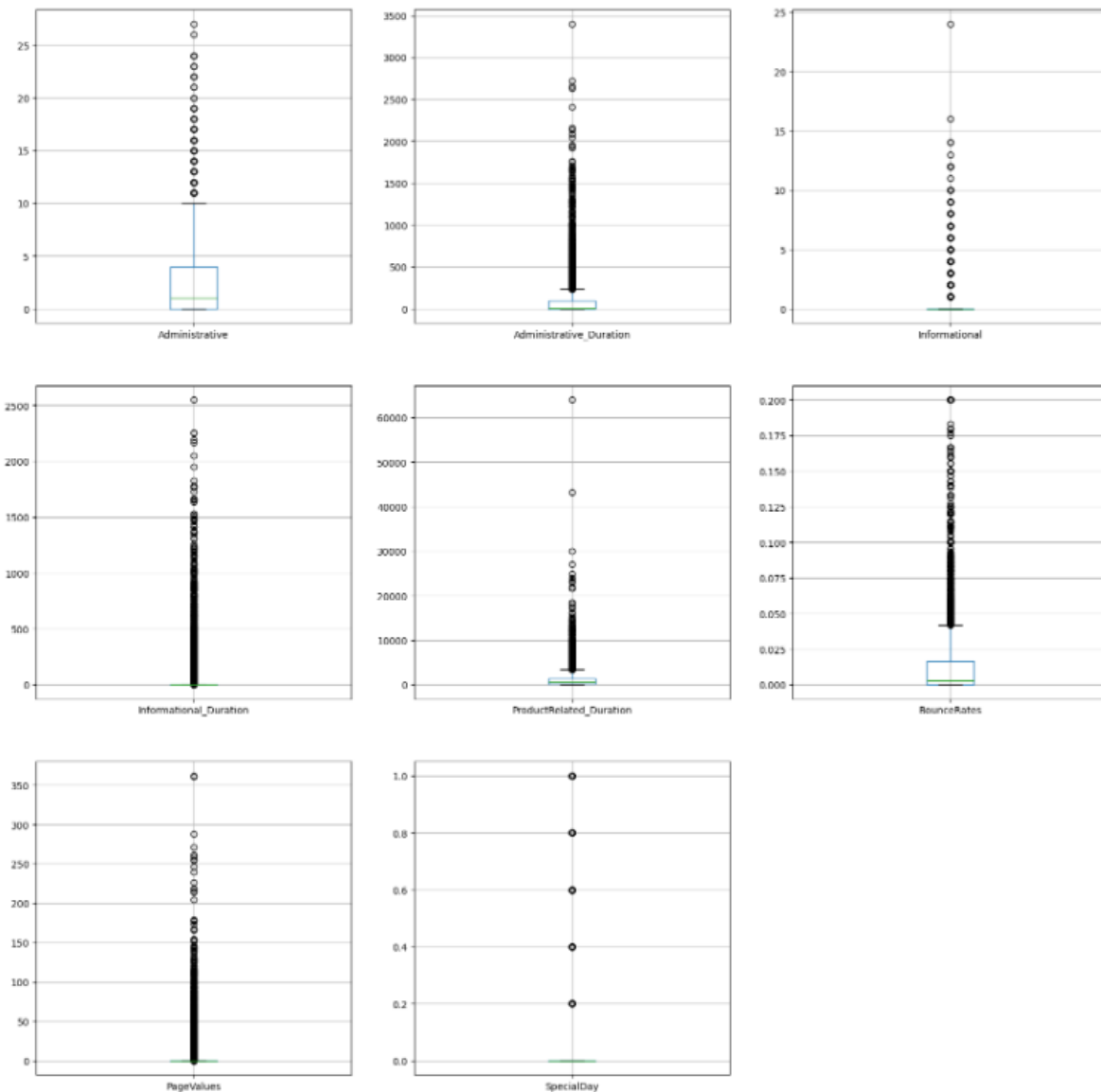


Hình 3.35: Kiểm tra đa cộng tuyến.



Hình 3.36: Kiểm tra tính đa cộng tuyến sau khi loại bỏ ProductRelation và ExitRates

8. Phát hiện các ngoại lệ thông qua vẽ đồ thị hộp.



Hình 3.37: Phát hiện các ngoại lệ.

3.1.3 Tiền xử lý dữ liệu

Trước khi chia bộ dữ liệu thành tập train và tập test, nhóm em xin thực hiện tiền xử lý dữ liệu:

- Xóa cột "Timestamp" Kết quả:

	age	Gender	Purchase_Frequency	Purchase_Categories	Personalized_Recommendation_Frequency	Browsing_Frequency	Product_Search_Method	Search_Result_Exploration	Customer_Review
0	23	Female	Few times a month	Beauty and Personal Care	Yes	Few times a week	Keyword	Multiple pages	
1	23	Female	Once a month	Clothing and Fashion	Yes	Few times a month	Keyword	Multiple pages	
2	24	Prefer not to say	Few times a month	Groceries and Gourmet Food;Clothing and Fashion	No	Few times a month	Keyword	Multiple pages	
3	24	Female	Once a month	Beauty and Personal Care;Clothing and Fashion;...	Sometimes	Few times a month	Keyword	First page	
4	22	Female	Less than once a month	Beauty and Personal Care;Clothing and Fashion	Yes	Few times a month	Filter	Multiple pages	
...
597	23	Female	Once a week	Beauty and Personal Care	Sometimes	Few times a week	categories	Multiple pages	
598	23	Female	Once a week	Clothing and Fashion	Sometimes	Few times a week	Filter	Multiple pages	
599	23	Female	Once a month	Beauty and Personal Care	Sometimes	Few times a week	categories	Multiple pages	
600	23	Female	Few times a month	Beauty and Personal Care;Clothing and Fashion;...	Yes	Few times a month	Keyword	Multiple pages	
601	23	Female	Once a week	Clothing and Fashion	Sometimes	Multiple times a day	Keyword	Multiple pages	

602 rows x 22 columns

Hình 3.38: Kết quả sau khi xóa cột "Timestamp"

• Gán nhãn tuổi

- Tuổi trong khoảng [0-20) được gán nhãn '15-25'
- Tuổi trong khoảng [20-30) được gán nhãn '25-35'
- Tuổi trong khoảng [30-40) được gán nhãn '35-45'
- Tuổi trong khoảng [40-50) được gán nhãn '45-55'
- Tuổi trong 50 trở đi được gán nhãn '55+'

Kết quả:

```

0      25-35
1      25-35
2      25-35
3      25-35
4      25-35
...
597    25-35
598    25-35
599    25-35
600    25-35
601    25-35
Name: age, Length: 600, dtype: category
Categories (5, object): ['15-25' < '25-35' < '35-45' < '45-55' < '55+']

```

Hình 3.39: Kết quả sau khi phân nhóm tuổi

- Thực hiện chuyển đổi các biến phân loại thành các giá trị số Kết quả:

```

age                                int32
Gender                             int32
Purchase_Frequency                 int32
Purchase_Categories                 int32
Personalized_Recommendation_Frequency int32
Browsing_Frequency                 int32
Product_Search_Method              int32
Search_Result_Exploration          int32
Customer_Reviews_Importance        int64
Add_to_Cart_Browsing               int32
Cart_Completion_Frequency          int32
Cart_Abandonment_Factors           int32
Saveforlater_Frequency             int32
Review_Left                        int32
Review_Reliability                 int32
Review_Helpfulness                 int32
Personalized_Recommendation_Frequency int64
Recommendation_Helpfulness         int32
Rating_Accuracy                   int64
Shopping_Satisfaction              int64
Service_Appreciation               int32
Improvement_Areas                  int32
dtype: object

```

Hình 3.40: Kết quả sau khi phân nhóm tuổi

• Chọn các đặc điểm tốt nhất để phân loại bằng ANOVA

- Thực hiện tính điểm dựa trên phân tích phương sai (ANOVA).
- Lựa chọn ra 10 đặc trưng tốt nhất bao gồm:
 - * Personalized_Recommendation_Frequency
 - * Search_Result_Exploration
 - * Add_to_Cart_Browsing
 - * Cart_Completion_Frequency
 - * Saveforlater_Frequency
 - * Review_Reliability
 - * Review_Helpfulness
 - * Personalized_Recommendation_Frequency
 - * Rating_Accuracy
 - * Shopping_Satisfaction
- Xóa các đặc trưng còn lại

Kết quả:

	Personalized_Recommendation_Frequency	Search_Result_Exploration	Add_to_Cart_Browsing	Cart_Completion_Frequency	Review_Left	Review_Reliability	Review_Helpfulness	Personalized_Recommendation_Frequency	Rating_Accuracy	Shopping_Satisfaction
0	2	1	2	4	1	3	2	2	1	1
1	2	1	2	2	0	0	2	2	3	2
2	0	1	2	4	0	3	0	4	3	3
3	1	0	0	4	1	0	2	3	3	4
4	2	1	2	4	0	0	2	4	2	2
...
587	1	1	0	4	1	1	1	3	3	4
588	1	1	0	4	1	0	1	3	3	3
589	1	1	0	4	1	3	1	3	2	3
600	2	1	2	2	0	0	2	2	2	2
601	1	1	0	2	1	1	1	3	3	3

602 rows x 10 columns

Hình 3.41: Kết quả chọn ra 10 đặc trưng tốt nhất

- Thực hiện tăng cường mẫu cho lớp thiểu số của bộ dữ liệu
- Chuẩn hóa dữ liệu về khoảng giá trị từ 0 đến 1 Kết quả:

```
array([[1. , 1. , 1. , ..., 0.25, 0. , 0. ],
       [1. , 1. , 1. , ..., 0.25, 0.5 , 0.25],
       [0. , 1. , 1. , ..., 0.75, 0.5 , 0.5 ],
       ...,
       [1. , 1. , 1. , ..., 0.25, 0.5 , 0.25],
       [0. , 1. , 1. , ..., 1. , 0.5 , 0.25],
       [1. , 0. , 1. , ..., 0.25, 0.25, 0. ]])
```

Hình 3.42: Kết quả sau khi chuẩn hóa

- Chia nhỏ bộ dữ liệu để kiểm tra và huấn luyện dữ liệu

3.2 Các phương pháp đánh giá mô hình

Số liệu phân loại

Giả sử ta có một ma trận nhầm lẫn như sau:

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

Trong đó:

- **TN (True Negative)**: Số lượng các trường hợp mà mô hình đã dự đoán đúng là tiêu cực và thực tế là tiêu cực.
- **FP (False Positive)**: Số lượng các trường hợp mà mô hình đã dự đoán là tích cực nhưng thực tế là tiêu cực.
- **FN (False Negative)**: Số lượng các trường hợp mà mô hình đã dự đoán là tiêu cực nhưng thực tế là tích cực.
- **TP (True Positive)**: Số lượng các trường hợp mà mô hình đã dự đoán đúng là tích cực và thực tế là tích cực.

Ma trận này giúp chúng ta đánh giá hiệu quả của mô hình phân loại dựa trên số lượng các dự đoán đúng và sai trong từng trường hợp. Nó là một công cụ quan trọng để tính toán các chỉ số đánh giá như accuracy, precision, recall và F1-score.

1. Accuracy

Accuracy là một thước đo để đánh giá các mô hình phân loại. Về mặt hình thức, accuracy có thể được định nghĩa là số lần dự đoán đúng trên tổng số lần dự đoán.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Viết điều này theo True Positive và True Negative: Tỷ lệ phần trăm trường hợp tích cực trong tổng số trường hợp tích cực thực tế. Do đó, mẫu số ($TP + FN$) ở đây là số lượng thực tế các trường hợp dương có trong tập dữ liệu.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Precision and Recall

Precision (còn được gọi là giá trị dự đoán dương) là phần nhỏ của các trường hợp có liên quan trong số các trường hợp được truy xuất, trong khi Recall (còn được gọi là độ nhạy) là phần của các trường hợp có liên quan đã được truy xuất. Do đó, cả độ chính xác và thu hồi đều dựa trên mức độ liên quan.

Trong phân loại, Precision là số lượng trường hợp tích cực thực sự (TP) trong tổng số trường hợp được gắn nhãn là thuộc lớp tích cực.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall là số lượng các trường hợp dương thực sự trên tổng số các trường hợp thực sự thuộc về lớp tích cực.

$$\text{Recall} = \frac{TP}{TP + FN}$$

3. F1 - score

Tùy thuộc vào ứng dụng, bạn có thể muốn ưu tiên Recall hoặc Precision cao hơn. Nhưng có nhiều ứng dụng trong đó cả thu hồi và độ chính xác đều quan trọng. Do đó, việc nghĩ ra cách kết hợp cả hai thành một chỉ số duy nhất là điều hoàn toàn tự nhiên.

Cuối cùng, thật tuyệt khi có một con số để đánh giá một mô hình học máy giống như bạn đạt được một điểm duy nhất trong một bài kiểm tra ở trường. Do đó, việc kết hợp các chỉ số đo độ chính xác và độ nhớ lại sẽ rất hợp lý; cách tiếp cận phổ biến để kết hợp các số liệu này được gọi là điểm f.

$$F_{\beta} = (1 + \beta^2) \times \frac{\text{Precision} \times \text{Recall}}{(\beta^2 \times \text{Precision}) + \text{Recall}}$$

Các tham số β cho phép điều chỉnh sự đánh đổi giữa độ chính xác (Precision) và thu hồi (Recall). Khi $\beta < 1$, sự quan tâm tăng đối với Precision hơn, trong khi $\beta > 1$, sự quan tâm tăng đối với Recall hơn.

Một số số liệu phổ biến kết hợp độ chính xác và thu hồi được gọi là điểm F1, là trung bình hài hòa của độ chính xác và thu hồi được định nghĩa là:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Vì giá trị trung bình hài hòa của một danh sách các số nghiêng nhiều về các phần tử nhỏ nhất của danh sách, nên nó có xu hướng (so với giá trị trung bình cộng) để giảm thiểu tác động của các giá trị ngoại lệ lớn và làm trầm trọng thêm tác động của các giá trị nhỏ.

Một nhược điểm là cả độ chính xác và thu hồi đều có tầm quan trọng như nhau do đó theo ứng dụng của chúng tôi, chúng tôi có thể cần một điểm cao hơn điểm kia và điểm F1 có thể không phải là số liệu chính xác cho nó. Do đó, điểm F1 có trọng số hoặc nhìn thấy đường cong PR hoặc ROC có thể hữu ích.

4. Support:

- Support là số lượng các điểm dữ liệu thực sự thuộc vào từng lớp nhãn trong tập dữ liệu kiểm tra.
- Support giúp bạn hiểu được phân phối của các lớp nhãn trong tập dữ liệu và có thể đánh giá được sự cân bằng giữa các lớp.

5. Macro avg:

$$\text{Macro avg} = \frac{1}{N} \sum_{i=1}^N \frac{P_i + R_i + F1_i}{3}$$

Trong đó N là số lượng lớp, P_i , R_i và $F1_i$ lần lượt là precision, recall và f1-score của lớp thứ i .

- Macro avg là trung bình không trọng số của precision, recall và f1-score qua tất cả các lớp.

- Nó tính toán bằng cách lấy trung bình của các chỉ số precision, recall và f1-score cho từng lớp một và sau đó lấy trung bình của các giá trị này.
- Macro avg giúp cân bằng các chỉ số đánh giá qua các lớp nhãn mà không phụ thuộc vào kích thước của từng lớp.

6. Weighted avg

$$\text{Weighted avg} = \frac{1}{N} \sum_{i=1}^N \left(\frac{P_i \cdot n_i + R_i \cdot n_i + F1_i \cdot n_i}{3 \cdot n} \right)$$

- Weighted avg là trung bình có trọng số của precision, recall và f1-score qua tất cả các lớp.
- Nó tính toán bằng cách lấy trung bình có trọng số của các chỉ số precision, recall và f1-score, trong đó trọng số là số lượng mẫu thực tế của từng lớp.
- Weighted avg hữu ích khi mô hình của bạn có các lớp không cân bằng về kích thước.

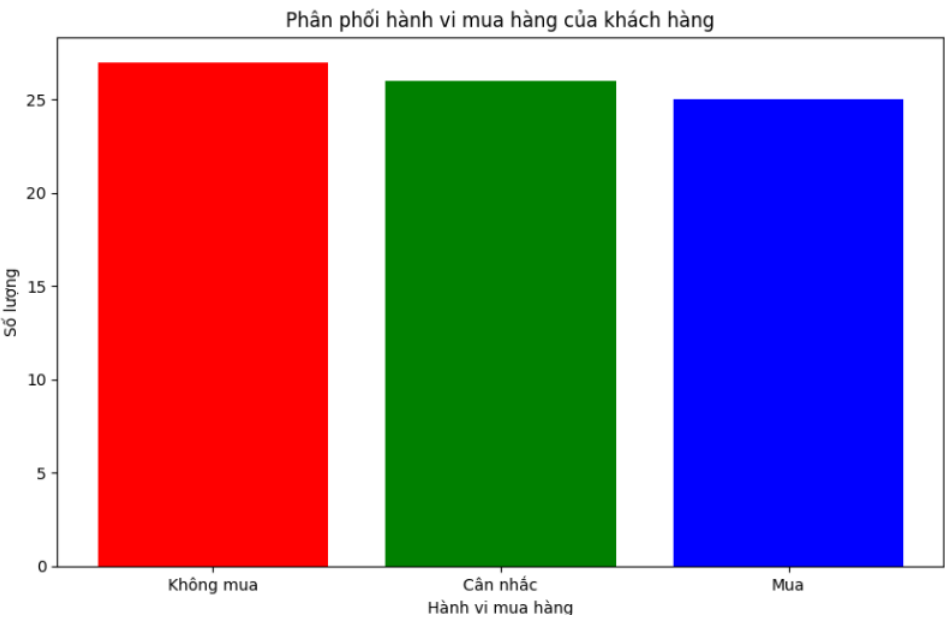
3.3 Kết quả và chỉ số đánh giá

Bộ dữ liệu được chia theo tỷ lệ tập train và test là 80:20. Sau quá trình huấn luyện, các mô hình đưa ra kết quả được đánh giá so với tập test như sau:

3.3.1 Bộ dữ liệu 1

```
array([2, 1, 0, 0, 2, 1, 0, 0, 2, 2, 2, 0, 1, 2, 0, 0, 0, 2, 0, 0, 0, 1,
       0, 2, 0, 2, 0, 0, 1, 2, 1, 2, 2, 0, 2, 1, 0, 1, 1, 2, 1, 0, 1, 2,
       1, 1, 2, 2, 1, 2, 2, 2, 1, 1, 1, 1, 2, 1, 0, 1, 2, 2, 0, 1, 2, 1,
       0, 0, 1, 2, 2, 0, 1, 1, 0, 1, 1, 0])
```

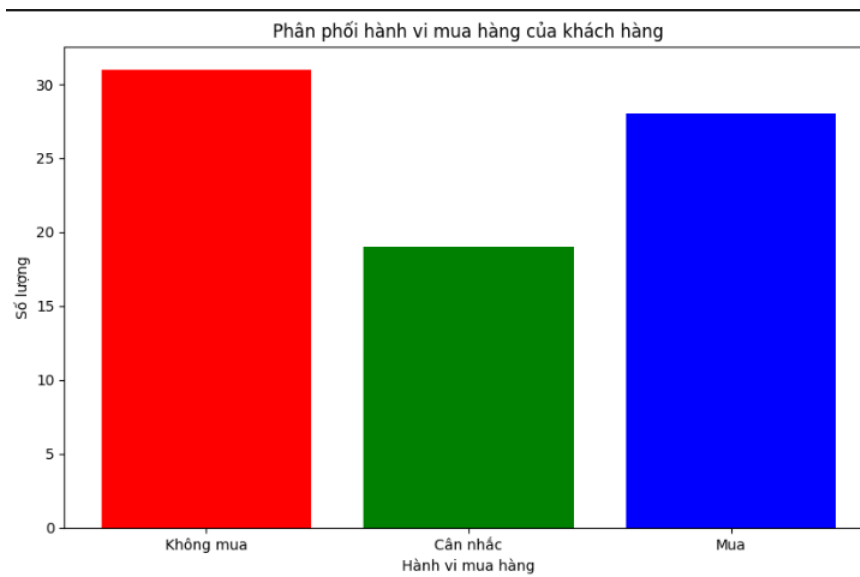
	precision	recall	f1-score	support
0	0.52	0.50	0.51	26
1	0.63	0.61	0.62	28
2	0.54	0.58	0.56	24
accuracy			0.56	78
macro avg	0.56	0.56	0.56	78
weighted avg	0.57	0.56	0.56	78



Hình 3.43: Mô hình hồi quy logistic

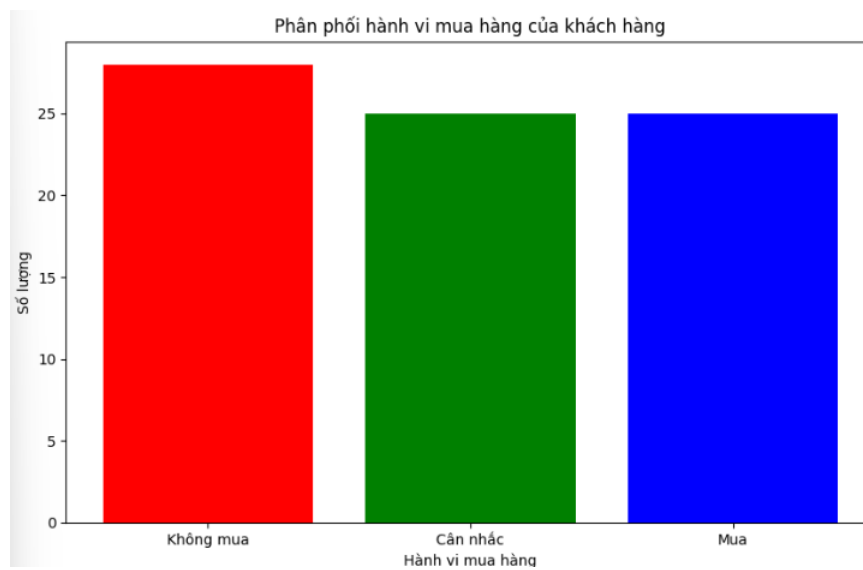
```
array([0, 2, 0, 0, 2, 0, 0, 0, 1, 2, 2, 0, 2, 2, 2, 2, 0, 2, 0, 1, 0, 1,
       0, 2, 1, 2, 1, 0, 2, 2, 1, 2, 2, 0, 1, 1, 0, 1, 0, 2, 1, 2, 0, 2,
       1, 1, 2, 2, 2, 0, 2, 0, 2, 2, 0, 1, 0, 1, 0, 1, 0, 2, 1, 0, 2, 2,
       0, 0, 2, 0, 2, 1, 0, 0, 1, 1, 0, 1])
```

	precision	recall	f1-score	support
0	0.59	0.65	0.62	26
1	0.65	0.46	0.54	28
2	0.66	0.79	0.72	24
accuracy			0.63	78
macro avg	0.63	0.64	0.63	78
weighted avg	0.63	0.63	0.62	78



Hình 3.44: Mô hình SVC

	precision	recall	f1-score	support
0	0.60	0.69	0.64	26
1	0.68	0.46	0.55	28
2	0.76	0.92	0.83	24
accuracy			0.68	78
macro avg	0.68	0.69	0.68	78
weighted avg	0.68	0.68	0.67	78

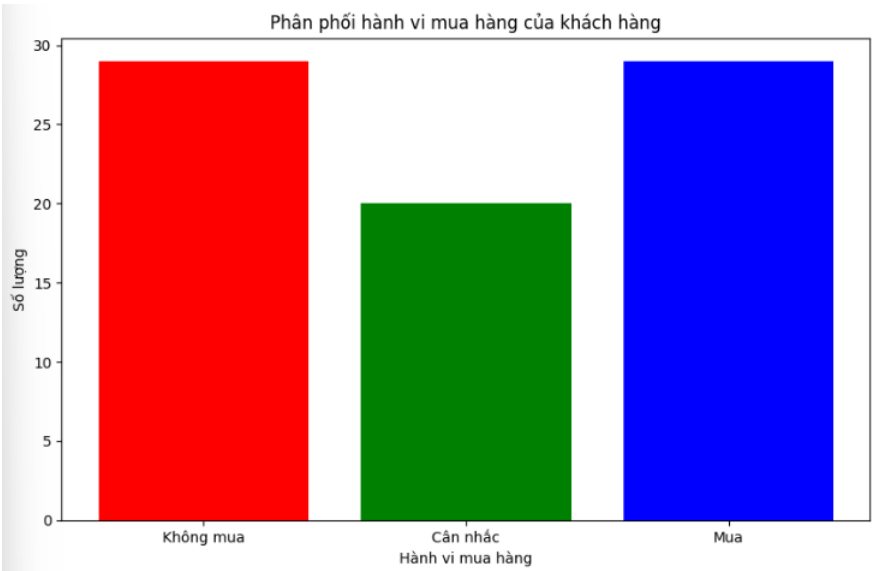


Hình 3.45: Mô hình KNN

```
array([2, 1, 0, 0, 2, 1, 0, 0, 2, 2, 1, 0, 1, 2, 0, 0, 0, 2, 0, 0, 0, 0,
       0, 2, 0, 2, 0, 0, 1, 2, 0, 2, 2, 0, 1, 1, 0, 1, 1, 2, 0, 0, 1, 2,
       0, 0, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 1, 1, 0, 1, 0, 2, 1, 1, 2, 2,
       1, 1, 2, 2, 2, 0, 2, 1, 0, 1, 2, 0])

precision recall f1-score support
0 0.50 0.54 0.52 26
1 0.64 0.50 0.56 28
2 0.57 0.67 0.62 24

accuracy 0.56 78
macro avg 0.57 0.57 0.56 78
weighted avg 0.57 0.56 0.56 78
```

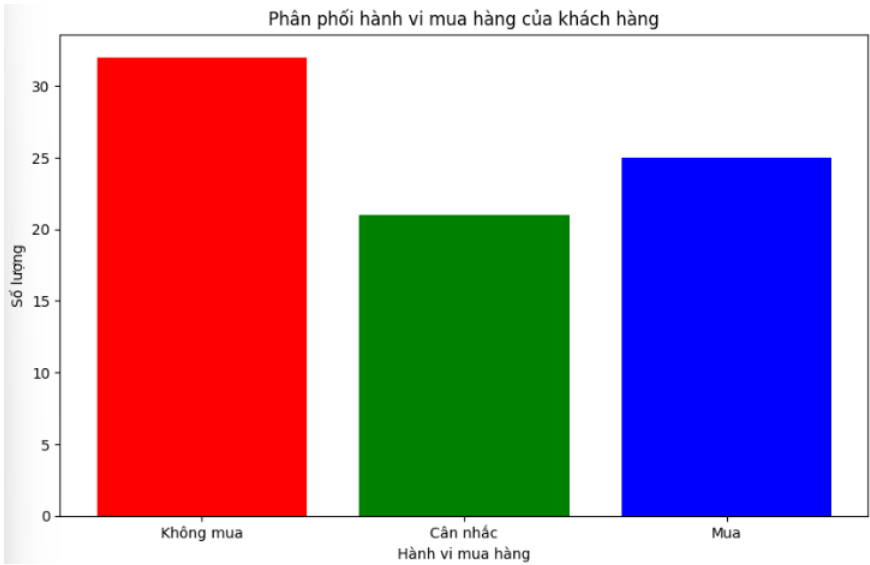


Hình 3.46: Mô hình GNB

```
array([2, 1, 2, 2, 2, 1, 0, 0, 2, 1, 2, 0, 1, 2, 0, 1, 0, 1, 0, 0, 0, 1,
       0, 2, 2, 2, 0, 2, 2, 2, 1, 2, 2, 0, 1, 1, 1, 1, 0, 2, 2, 2, 0, 2,
       2, 2, 0, 2, 2, 2, 2, 0, 0, 2, 0, 1, 0, 2, 0, 1, 0, 1, 0, 1, 2, 2,
       0, 1, 2, 1, 2, 2, 0, 0, 0, 1, 0, 2])

precision recall f1-score support
0 0.54 0.54 0.54 26
1 0.74 0.50 0.60 28
2 0.55 0.75 0.63 24

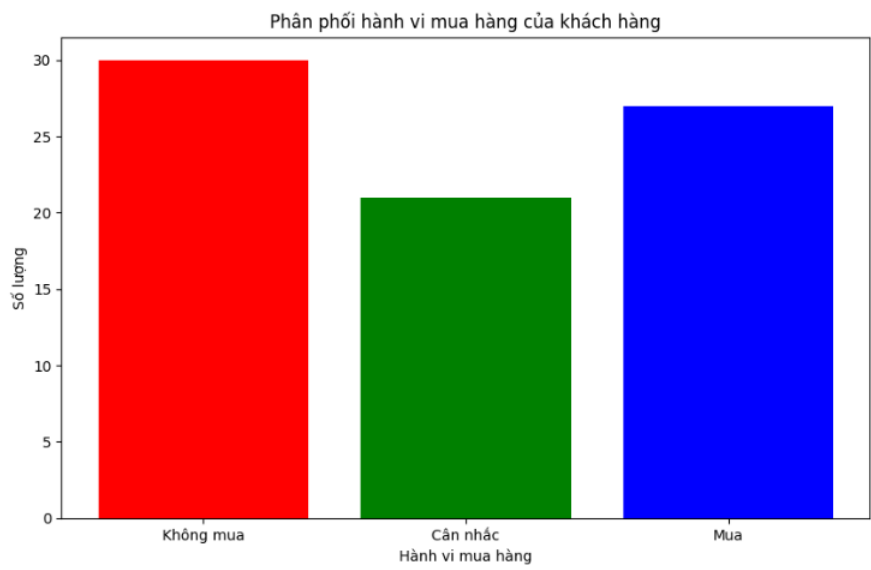
accuracy 0.59 78
macro avg 0.61 0.60 0.59 78
weighted avg 0.61 0.59 0.59 78
```



Hình 3.47: Mô hình cây quyết định

```
array([0, 1, 0, 2, 2, 1, 0, 0, 2, 2, 2, 2, 1, 2, 2, 0, 0, 2, 0, 0, 0, 1,
       2, 2, 0, 2, 1, 0, 2, 2, 1, 2, 1, 0, 1, 1, 0, 1, 0, 2, 1, 2, 0, 2,
       1, 1, 2, 2, 2, 0, 2, 0, 2, 2, 0, 1, 0, 1, 0, 1, 0, 2, 0, 0, 2, 2,
       0, 0, 2, 2, 2, 0, 1, 0, 0, 1, 0, 0])
```

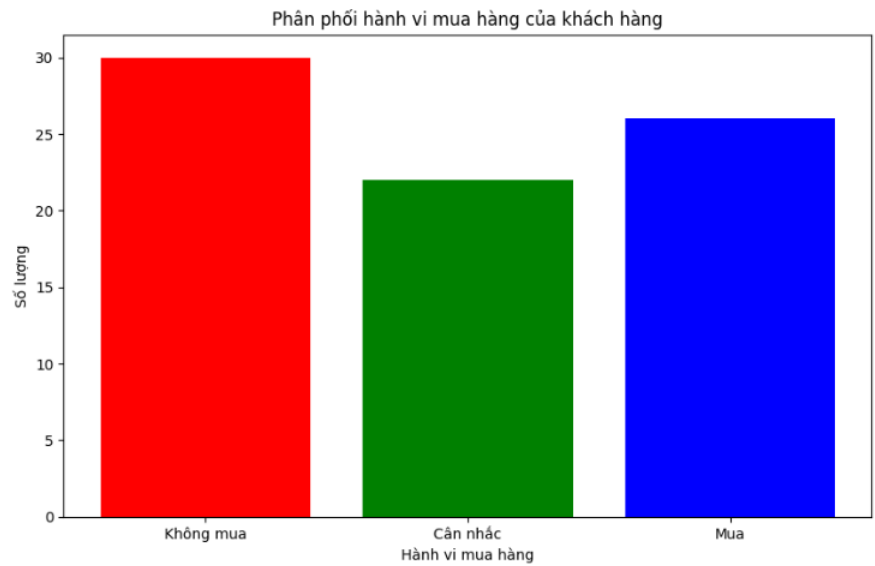
	precision	recall	f1-score	support
0	0.67	0.77	0.71	26
1	0.83	0.54	0.65	28
2	0.73	0.92	0.81	24
accuracy			0.73	78
macro avg	0.74	0.74	0.73	78
weighted avg	0.75	0.73	0.72	78



Hình 3.48: Mô hình rừng ngẫu nhiên

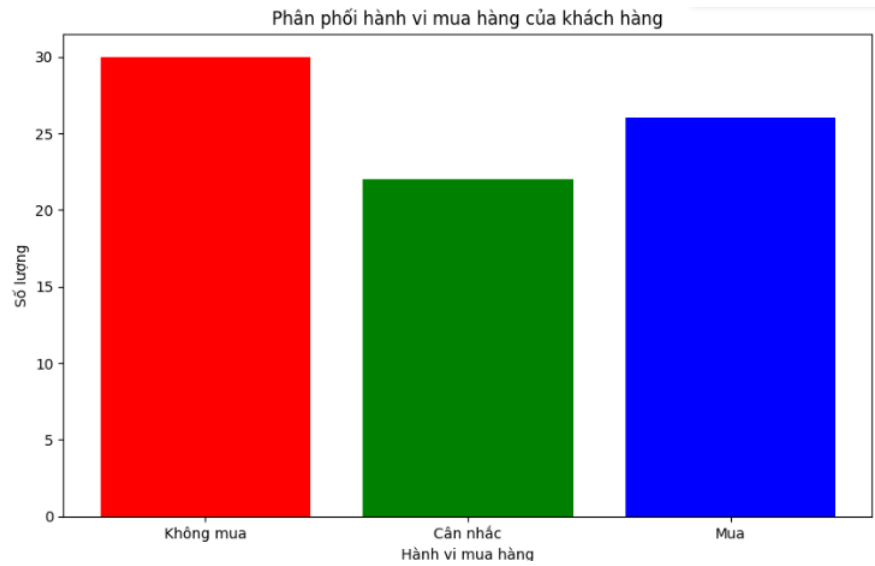
```
array([0, 1, 0, 2, 2, 1, 0, 0, 2, 2, 2, 2, 1, 2, 2, 0, 0, 2, 0, 0, 0, 1,
      2, 2, 2, 2, 1, 0, 2, 2, 1, 2, 1, 0, 1, 1, 0, 1, 0, 2, 1, 2, 0, 2,
      1, 1, 2, 2, 2, 0, 2, 0, 0, 2, 0, 1, 0, 1, 0, 1, 0, 2, 0, 0, 2, 2,
      0, 0, 2, 1, 2, 1, 0, 0, 0, 1, 0, 2], dtype=int64)
```

	precision	recall	f1-score	support
0	0.66	0.73	0.69	26
1	0.79	0.54	0.64	28
2	0.73	0.92	0.81	24
accuracy			0.72	78
macro avg	0.73	0.73	0.71	78
weighted avg	0.73	0.72	0.71	78



Hình 3.49: Mô hình XGBoost

	precision	recall	f1-score	support
0	0.67	0.77	0.71	26
1	0.75	0.54	0.62	28
2	0.79	0.92	0.85	24
accuracy			0.73	78
macro avg	0.73	0.74	0.73	78
weighted avg	0.73	0.73	0.72	78



Hình 3.50: Mô hình CatBoost

3.3.2 Bộ dữ liệu 2

```

LogisticRegression()
Model score: 0.879
['False' 'False' 'False' ... 'False' 'False' 'False']
Confusion Matrix:
[[2032  52]
 [ 247 135]]
Classification Report:

```

	precision	recall	f1-score	support
False	0.89	0.98	0.93	2084
True	0.72	0.35	0.47	382
accuracy			0.88	2466
macro avg	0.81	0.66	0.70	2466
weighted avg	0.87	0.88	0.86	2466

Hình 3.51: Mô hình hồi quy logistic

```

SVC(C=0.025, probability=True)
Model score: 0.870
['False' 'False' 'False' ... 'False' 'False' 'False']
Confusion Matrix:
[[2051  33]
 [ 288  94]]
Classification Report:

```

	precision	recall	f1-score	support
False	0.88	0.98	0.93	2084
True	0.74	0.25	0.37	382
accuracy			0.87	2466
macro avg	0.81	0.62	0.65	2466
weighted avg	0.86	0.87	0.84	2466

Hình 3.52: Mô hình SVC


```

DecisionTreeClassifier()
Model score: 0.861
['False' 'False' 'False' ... 'False' 'False' 'False']
Confusion Matrix:
[[1919  165]
 [ 179  203]]
Classification Report:

```

	precision	recall	f1-score	support
False	0.91	0.92	0.92	2084
True	0.55	0.53	0.54	382
accuracy			0.86	2466
macro avg	0.73	0.73	0.73	2466
weighted avg	0.86	0.86	0.86	2466

Hình 3.53: Mô hình cây quyết định

```

RandomForestClassifier()
Model score: 0.895
['False' 'False' 'False' ... 'False' 'False' 'False']
Confusion Matrix:
[[2023  61]
 [ 197  185]]
Classification Report:

```

	precision	recall	f1-score	support
False	0.91	0.97	0.94	2084
True	0.75	0.48	0.59	382
accuracy			0.90	2466
macro avg	0.83	0.73	0.76	2466
weighted avg	0.89	0.90	0.89	2466

Hình 3.54: Mô hình rừng ngẫu nhiên

```

GradientBoostingClassifier()
Model score: 0.903
['False' 'False' 'False' ... 'False' 'False' 'False']
Confusion Matrix:
[[2006  78]
 [ 160 222]]
Classification Report:

```

	precision	recall	f1-score	support
False	0.93	0.96	0.94	2084
True	0.74	0.58	0.65	382
accuracy			0.90	2466
macro avg	0.83	0.77	0.80	2466
weighted avg	0.90	0.90	0.90	2466

Hình 3.55: Mô hình Gradient Boosting

```

LGBMClassifier()
Model score: 0.898
['False' 'False' 'False' ... 'False' 'False' 'False']
Confusion Matrix:
[[1999  85]
 [ 167 215]]
Classification Report:

```

	precision	recall	f1-score	support
False	0.92	0.96	0.94	2084
True	0.72	0.56	0.63	382
accuracy			0.90	2466
macro avg	0.82	0.76	0.79	2466
weighted avg	0.89	0.90	0.89	2466

Hình 3.56: Mô hình LGBM

```

MLPClassifier()
Model score: 0.883
['False' 'False' 'False' ... 'False' 'False' 'False']
Confusion Matrix:
[[1987  97]
 [ 192 190]]
Classification Report:

```

	precision	recall	f1-score	support
False	0.91	0.95	0.93	2084
True	0.66	0.50	0.57	382
accuracy			0.88	2466
macro avg	0.79	0.73	0.75	2466
weighted avg	0.87	0.88	0.88	2466

Hình 3.57: Mô hình MLP

3.4 Thống kê và phân tích lỗi

Qua các chỉ số đánh giá, ta nhận xét các mô hình như sau:

1. Mô hình KNN:

- Loại lỗi phổ biến:
 - False Positives (FP): Mô hình dự đoán sai các điểm của lớp âm là lớp dương.
 - False Negatives (FN): Mô hình dự đoán sai các điểm của lớp dương là lớp âm.
- Nguyên nhân tiềm ẩn của lỗi:
 - K nhỏ: K nhỏ có thể dẫn đến mô hình quá nhạy cảm với nhiễu trong dữ liệu huấn luyện.
 - K lớn: K lớn có thể làm giảm độ chính xác do việc xem xét quá nhiều điểm láng giềng.
 - Phân phối không đồng đều: Dữ liệu không đồng đều hoặc chồng chéo giữa các lớp có thể gây nhầm lẫn.
- Cách khắc phục:
 - Điều chỉnh K: Tìm giá trị K tối ưu bằng cách sử dụng cross-validation.
 - Chuẩn hóa dữ liệu: Đảm bảo tất cả các đặc trưng có cùng thang đo.
 - Sử dụng trọng số: Trọng số các láng giềng dựa trên khoảng cách của chúng đến điểm dự đoán.

2. Mô hình Gaussian Naive Bayes:

- Loại lỗi phổ biến:
 - False Positives (FP): Mô hình dự đoán sai các điểm của lớp âm là lớp dương.
 - False Negatives (FN): Mô hình dự đoán sai các điểm của lớp dương là lớp âm.
- Nguyên nhân tiềm ẩn của lỗi
 - Giả định phân phối chuẩn: Nếu các đặc trưng không tuân theo phân phối chuẩn, mô hình có thể hoạt động kém.
 - Đặc trưng không độc lập: GNB giả định các đặc trưng độc lập, điều này không phải lúc nào cũng đúng trong thực tế.
 - Mất cân bằng lớp: Sự mất cân bằng giữa các lớp có thể ảnh hưởng đến hiệu suất của GNB.
- Cách khắc phục:
 - Biến đổi đặc trưng: Sử dụng các biến đổi như Box-Cox hoặc log để biến đổi dữ liệu về phân phối chuẩn.
 - Chọn lọc đặc trưng: Loại bỏ các đặc trưng không độc lập hoặc sử dụng các phương pháp như PCA để giảm số chiều của dữ liệu.
 - Cân bằng dữ liệu: Sử dụng các kỹ thuật như oversampling hoặc undersampling để cân bằng dữ liệu.

3. Mô hình cây quyết định:

- Loại lỗi phổ biến:
 - False Positives (FP): Mô hình dự đoán sai các điểm của lớp âm là lớp dương.
 - False Negatives (FN): Mô hình dự đoán sai các điểm của lớp dương là lớp âm.
- Nguyên nhân tiềm ẩn của lỗi:
 - Overfitting: Mô hình học quá mức vào dữ liệu huấn luyện, gây giảm hiệu quả trên dữ liệu kiểm tra.
 - Quá sâu: Cây quyết định có thể trở nên quá phức tạp với quá nhiều cấp độ.
 - Dữ liệu nhiễu: Dữ liệu huấn luyện chứa nhiễu.
- Cách khắc phục:
 - Pruning: Sử dụng kỹ thuật cắt tỉa cây để loại bỏ các nhánh không cần thiết.

- Giới hạn độ sâu cây: Đặt giới hạn cho độ sâu của cây. Tăng cường dữ liệu: Sử dụng thêm dữ liệu hoặc kỹ thuật làm mịn dữ liệu.

4. Mô hình rừng ngẫu nhiên

- Loại lỗi phổ biến:
 - False Positives (FP): Mô hình dự đoán sai các điểm của lớp âm là lớp dương.
 - False Negatives (FN): Mô hình dự đoán sai các điểm của lớp dương là lớp âm.
- Nguyên nhân tiềm ẩn của lỗi:
 - Cấu hình chưa tối ưu: Số lượng cây hoặc các tham số khác của mô hình chưa được tối ưu hóa.
 - Mất cân bằng dữ liệu: Sự mất cân bằng giữa các lớp có thể ảnh hưởng đến hiệu suất của Random Forest.
 - Quá phức tạp: Mô hình có thể trở nên quá phức tạp với quá nhiều cây.
- Cách khắc phục:
 - Tăng số lượng cây: Tăng số lượng cây để cải thiện độ chính xác.
 - Điều chỉnh tham số: Tìm các giá trị tối ưu cho các tham số như số lượng cây, độ sâu cây, số lượng đặc trưng sử dụng tại mỗi nút chia.
 - Cân bằng dữ liệu: Sử dụng các kỹ thuật như oversampling hoặc undersampling để cân bằng dữ liệu.

5. Mô hình XGBoost

- Loại lỗi phổ biến:
 - False Positives (FP): Mô hình dự đoán sai các điểm của lớp âm là lớp dương.
 - False Negatives (FN): Mô hình dự đoán sai các điểm của lớp dương là lớp âm.
- Nguyên nhân tiềm ẩn của lỗi:
 - Overfitting: Mặc dù XGBoost có cơ chế giảm thiểu overfitting, việc cấu hình sai tham số có thể vẫn gây ra overfitting.
 - Underfitting: Khi mô hình quá đơn giản hoặc không đủ mạnh để nắm bắt được cấu trúc của dữ liệu.
 - Mất cân bằng dữ liệu: Các lớp không cân bằng có thể gây ra các vấn đề về hiệu suất.

- Cách khắc phục:
 - Điều chỉnh tham số: Tối ưu hóa các tham số như learning rate, max_depth, và n_estimators.
 - Cân bằng dữ liệu: Sử dụng các kỹ thuật như oversampling hoặc undersampling để cân bằng dữ liệu.
 - Regularization: Sử dụng regularization mạnh mẽ hơn (L1, L2) để giảm thiểu overfitting.
 - Tăng cường dữ liệu: Sử dụng thêm dữ liệu hoặc các kỹ thuật làm mịn dữ liệu.

6. Mô hình CatBoost

- Loại lỗi phổ biến:
 - False Positives (FP): Mô hình dự đoán sai các điểm của lớp âm là lớp dương.
 - False Negatives (FN): Mô hình dự đoán sai các điểm của lớp dương là lớp âm.
- Nguyên nhân tiềm ẩn của lỗi:
 - Overfitting: Mặc dù CatBoost có cơ chế giảm thiểu overfitting, cấu hình sai tham số vẫn có thể gây ra overfitting.
 - Underfitting: Khi mô hình quá đơn giản hoặc không đủ mạnh để nắm bắt được cấu trúc của dữ liệu.
 - Mất cân bằng dữ liệu: Các lớp không cân bằng có thể gây ra các vấn đề về hiệu suất.
- Cách khắc phục:
 - Điều chỉnh tham số: Tối ưu hóa các tham số như learning rate, depth, và iterations.
 - Cân bằng dữ liệu: Sử dụng các kỹ thuật như oversampling hoặc undersampling để cân bằng dữ liệu.
 - Regularization: Sử dụng regularization mạnh mẽ hơn (L2) để giảm thiểu overfitting.
 - Tăng cường dữ liệu: Sử dụng thêm dữ liệu hoặc các kỹ thuật làm mịn dữ liệu.

7. Mô hình MLP

- Loại lỗi phổ biến:

- False Positives (FP): Mô hình dự đoán sai các điểm của lớp âm là lớp dương.
- False Negatives (FN): Mô hình dự đoán sai các điểm của lớp dương là lớp âm.
- Nguyên nhân tiềm ẩn của lỗi:
 - Overfitting: Khi mạng quá phức tạp, mô hình có thể học quá kỹ các chi tiết của dữ liệu huấn luyện và không tổng quát hóa được cho dữ liệu mới.
 - Underfitting: Khi mạng quá đơn giản hoặc không đủ sâu để học các mối quan hệ phức tạp trong dữ liệu.
 - Lựa chọn tham số sai: Các tham số như số lượng lớp, số lượng nơ-ron mỗi lớp, learning rate, và batch size có thể ảnh hưởng lớn đến hiệu suất của mô hình.
- Cách khắc phục:
 - Điều chỉnh tham số: Tối ưu hóa các tham số của mạng nơ-ron như số lượng lớp, số lượng nơ-ron mỗi lớp, learning rate, và batch size.
 - Regularization: Sử dụng các kỹ thuật như dropout hoặc L2 regularization để giảm thiểu overfitting.
 - Early Stopping: Sử dụng kỹ thuật early stopping để dừng huấn luyện khi mô hình bắt đầu overfit dữ liệu huấn luyện.
 - Cân bằng dữ liệu: Sử dụng các kỹ thuật như oversampling hoặc undersampling để cân bằng dữ liệu.
 - Tăng cường dữ liệu: Sử dụng các kỹ thuật tăng cường dữ liệu (data augmentation) để tạo ra các biến thể của dữ liệu huấn luyện và cải thiện khả năng tổng quát của mô hình.

Chương 4

Đóng gói mô hình

4.1 Giao diện chương trình

4.2 Kịch bản chương trình

B là một quản lý marketing và muốn dự đoán hành vi mua hàng của khách hàng dựa trên các đặc điểm và hành vi lịch sử của họ. B có một công cụ trực tuyến có thể dự đoán hành vi mua hàng dựa trên các mô hình thống kê và học sâu với các bộ dữ liệu đa dạng. B tiến hành sử dụng như sau:

1. Truy cập ứng dụng web: Chạy lệnh **flask run** trên terminal, sau đó truy cập vào địa chỉ **http://127.0.0.1:5000**.
2. Chọn khách hàng: Chọn một khách hàng cụ thể hoặc một nhóm khách hàng để dự đoán hành vi mua hàng.
3. Thử qua các mô hình dự đoán:
 - Logistic Regression
 - Decision Tree
 - Random Forest
 - XGBoost
 - CatBoost
 - LGBosting
 - MLP (Multi-Layer Perceptron)
4. Xem kết quả dự đoán: Đối với mỗi mô hình, B sẽ xem kết quả dự đoán về khả năng mua hàng của khách hàng.

5. Đưa ra quyết định marketing:

- Nếu khả năng mua hàng cao: B có thể đưa ra các chiến lược marketing tập trung, như gửi email quảng cáo, ưu đãi đặc biệt, hoặc gọi điện trực tiếp để khuyến khích khách hàng mua hàng.
- Nếu khả năng mua hàng thấp: B có thể xem xét các chiến lược khác để thu hút sự quan tâm của khách hàng, như cải thiện nội dung website, cung cấp thông tin hữu ích, hoặc chạy các chiến dịch quảng cáo nhằm tăng cường nhận thức về thương hiệu.

Tài liệu tham khảo

- [1] C. Kingsford and S. L. Salzberg, "What are decision trees?", *Nature Biotechnology*, 2008. Available: <https://www.nature.com/articles/nbt1384>
- [2] "XGBoost", *Wikipedia*. Available: <https://en.wikipedia.org/wiki/XGBoost>
- [3] "Gradient Boosting", *Wikipedia*. Available: https://en.wikipedia.org/wiki/Gradient_boosting
- [4] "Gradient Boosting Regression Example", *Scikit-learn*. Available: https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_regression.html
- [5] R. H. Shumway, et al., "ARIMA models", *Time Series Analysis and Its Applications: With R Examples*, 2017, pp. 75-163.
- [6] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network", *Physica D: Nonlinear Phenomena*, vol. 404, 2020, Art. no. 132306.
- [7] Duong, *Recurrent Neural Network (Phần 1): Tổng quan và ứng dụng*, <https://viblo.asia/p/recurrent-neural-networkphan-1-tong-quan-va-ung-dung-jvElaB4m5kw>, Viblo, 2024
- [8] Sakar, C.O., Polat, S.O., Katircioglu, M. et al., *Neural Comput & Applic*, (2019) 31: 6893.<https://doi.org/10.1007/s00521-018-3523-0>

Checklist

STT	Loại yêu cầu	Yêu cầu	Điểm chữ	Điểm số	Check	Minh chứng	Người thực hiện	
							Vang	Châu
1	Xử lý dữ liệu (2 điểm)	Thu thập dữ liệu từ nhiều nguồn	A	1		Trang 30 - 32	X	
2		Đánh nhãn dữ liệu	A	1		Trang 56	X	
3		Tiền xử lý dữ liệu	A	1		Trang 55 - 58		X
4		Thống kê dữ liệu mẫu	A	1		Trang 33 - 54	X	
5	Đánh giá mô hình (1 điểm)	Đề xuất và lựa chọn các tiêu chí đánh giá (về độ chính xác, tốc độ, khả năng ứng dụng,...)	A	1		Trang 58 - 60		X
6		Thống kê và phân tích lỗi	A	1		Trang 71 -75	X	
7	Cải tiến mô hình (5 điểm)	Mô hình hồi quy logistic với bộ dữ liệu 1	A	1		Trang 9, 61, 62		X
8		Mô hình hồi quy logistic với bộ dữ liệu 2	A	1		Trang 9,		X
9		Mô hình SVC với bộ dữ liệu 1	A	1		Trang 12, 63	X	
10		Mô hình SVC với bộ dữ liệu 2	A	1		Trang 12, 68		X
11		Mô hình KNN với bộ dữ liệu 1	A	1		Trang 10, 11, 63	X	
12		Mô hình Gaussian Naive Bayes với bộ dữ liệu 1	A	1		Trang 13, 14, 64		X
13		Mô hình cây quyết định với bộ dữ liệu 1	A	1		Trang 15, 16, 64, 65		X
14		Mô hình cây quyết định với bộ dữ liệu 2	A	1		Trang 15, 16, 69	X	
15		Mô hình rừng ngẫu nhiên với bộ dữ liệu 1	A	1		Trang 19, 66	X	
16		Mô hình rừng ngẫu nhiên với bộ dữ liệu 2	A	1		Trang 19, 69		X
17		Mô hình XGBoost với bộ dữ liệu 1	A	1		Trang 20 - 22, 67		X
18		Mô hình XGBoost với bộ dữ liệu 2	A	1		Trang 20 - 22, 70	X	
19		Mô hình CatBoost với bộ dữ liệu 1	A	1		Trang 23, 67		X
20		Mô hình LGBosting với bộ dữ liệu 2	A	1		Trang 24, 70	X	
21		Mô hình MLP	A	1		Trang 25, 26, 80	X	X
22	Đóng gói mô hình (2 điểm)	Có giao diện chương trình	F	0				
23		Có khả năng ứng dụng vào một ngữ cảnh cụ thể (trình bày kịch bản demo ứng dụng)	A	1		Trang 76, 77	X	
24		Có sử dụng mô hình tiên tiến trong 5 năm trở lại đây (chỉ ra paper liên quan) (paper 8)	A	1		Trang 80	X	
25		Cải tiến mô hình so sánh với các mô hình tiên tiến trong 5 năm gần đây	F	0		Trang		
		Tổng điểm/20		18/20				
		Tổng điểm/10		10-Sep				