TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT

KHOA HỆ THỐNG THÔNG TIN

BÁO CÁO ĐỒ ÁN CUỐI KỲ

MÔN PHÂN TÍCH DỮ LIỆU VỚI R/ PYTHON

ĐỀ TÀI: XÂY DỰNG MÔ HÌNH PHÂN TÍCH CÁC YẾU TỐ ẢNH HƯỞNG VÀ DỰ ĐOÁN Ý ĐỊNH MUA HÀNG CỦA KHÁCH HÀNG TRÊN CÁC TRANG THƯƠNG MẠI ĐIỆN TỬ

Giảng viên: ThS. Nguyễn Phát Đạt

Trợ giảng: Anh Trần Lê Tấn Thịnh

Nhóm: Đậu

Thành phố Hồ Chí Minh, ngày 28 tháng 11 năm 2022

THÀNH VIÊN NHÓM

STT	Họ và tên	MSSV	Lớp	Vai trò	Đóng góp
1	Nguyễn Chí Bảo	K204060279	K20406	Nhóm trưởng	100%
2	Hồ Thị Minh Nguyên	K204110576	K20411	Thành viên	100%
3	Nguyễn Ngọc Yến Nhi	K204110578	K20411	Thành viên	100%
4	Trịnh Thị Minh Khai	K204111777	K20411	Thành viên	100%
5	Võ Ngọc Tường Vy	K204111793	K20411	Thành viên	100%

LỜI CẢM ƠN

Nhóm đã áp dụng những gì học được trong khóa học Phân tích dữ liệu với R/ Python thông qua quá trình học hỏi kiến thức lý thuyết cũng như kiến thức thực tế để xây dựng dự án.

Nhóm xin gửi lời cảm ơn chân thành đến tất cả những người đã hỗ trợ nhóm hoàn thành báo cáo đồ án cuối khóa học. Nhóm xin cảm ơn thầy Nguyễn Phát Đạt đã cung cấp kiến thức nền tảng vững chắc, đã có nhiều ý kiến đóng góp giúp nhóm hoàn thành tốt đồ án và đưa ra các giải pháp cho nhóm khi gặp khó khăn. Ngoài các giảng viên chính của bộ môn, nhóm xin cảm ơn trợ giảng Trần Lê Tấn Thịnh đã chia sẻ những thông tin vô cùng hữu ích.

Trong thời gian hạn hẹp của dự án, nhóm đã cố gắng lên ý tưởng và giải quyết các yêu cầu ban đầu một cách tốt nhất có thể. Tuy nhiên, vẫn còn nhiều trở ngại cần khắc phục, không tránh khỏi những sai sót. Mong thầy cô đọc và góp ý cho đề tài của nhóm để nhóm rút kinh nghiệm và hoàn thiện hơn.

Nhóm Đậu

LỜI CAM KẾT

Trong quá trình thực hiện đề tài, nhóm xin cam kết thực hiện đúng các quy định, các số liệu và kết quả trình bày trong báo cáo là chính xác. Tất cả các tài liệu tham khảo trên Internet, sách và giáo trình đều được trích dẫn cụ thể.

Nếu có sai phạm, nhóm xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật.

Nhóm Đậu

MỤC LỤC

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH

DANH MỤC TỪ VIẾT TẮT

STT	Ký hiệu chữ viết tắt	Chữ viết đầy đủ
1	ML	Machine Learning
2	AI	Artificial Intelligence
3	NBC	Naive Bayes Classification
4	KNN	K – Nearest Neighbor
5	AUC	Area Under the Curve
6	ROC Curve	The receiver operating characteristic curve
7	UI	User Interface
8	UX	User Experience
9	Pop-up	một hộp thoại nhỏ tự động bật lên khi người dùng mở trình duyệt hoặc truy cập vào một website
10	SEO	Search Engine Optimization

TỔNG QUAN VỀ ĐỒ ÁN

Trình bày tổng quan về nội dung của đề tài, bao gồm lý do chọn đề tài, mục tiêu, kết quả dự kiến, phương pháp nghiên cứu, công cụ nghiên cứu và quy trình thực hiện

1. LÝ DO CHỌN ĐỀ TÀI

Mua sắm trực tuyến là một hình thức mua hàng đang rất phát triển và chiếm một phần lớn doanh thu B2C (Doanh nghiệp với Khách hàng). Mua sắm trực tuyến là một dạng thương mại điện tử cho phép khách hàng trực tiếp mua hàng hóa hoặc dịch vụ từ người bán qua Internet sử dụng trình duyệt web. Người tiêu dùng tìm thấy một sản phẩm quan tâm bằng cách trực tiếp truy cập trang web của nhà bán lẻ hoặc tìm kiếm trong số các nhà cung cấp khác sử dụng công cụ tìm kiếm mua sắm, hiển thị sự sẵn có và giá của sản phẩm tương tự tại các nhà bán lẻ điện tử khác nhau. Kể từ năm 2016, khách hàng có thể mua sắm trực tuyến bằng nhiều loại máy tính và thiết bị khác nhau, bao gồm máy tính để bàn, máy tính xách tay, máy tính bảng và điện thoại thông minh.

Xu hướng mua sắm trực tuyến ở thế giới

Thị trường thương mại điện tử của Mỹ được dự báo sẽ đạt hơn 875 tỷ USD vào năm 2022, hơn một phần ba so với thị trường của Trung Quốc. Thị trường thương mại điện tử lớn thứ ba là Anh, chiếm 4,8% thị phần thương mại điện tử bán lẻ; tiếp đó là Hàn Quốc (2,5%).

Theo số liệu Statista, tỷ trọng thương mại điện tử xuyên biên giới trung bình của Đông Nam Á tăng từ 74 tỷ USD năm 2020 lên 120 tỷ USD năm 2021. Giai đoạn 2016-2020, tốc độ tăng trưởng đạt trung bình 37,7%/năm, cao hơn mức trung bình toàn cầu 27,4%/năm. Dự báo, doanh thu thương mại điện tử năm 2025 tại khu vực Đông Nam Á dự kiến đạt 234 tỷ USD. Trong hai năm qua, số lượng người mua sắm trực tuyến ở khu vực Đông Nam Á đã tăng đáng kể, đạt khoảng 70 triệu người tính đến thời điểm hiện tại. Trên quy mô khu vực, 70% tổng dân số ở Đông Nam Á đã bắt đầu mua sắm trực tuyến trước cả khi đại dịch Covid-19 bùng phát. Số lượng người mua sắm trực tuyến tại Đông Nam Á dự kiến tăng đến con số 380 triệu trước năm 2026.

Xu hướng mua sắm trực tuyến ở Việt Nam

Trong đó, ở Việt Nam, năm 2022, số lượng người Việt mua hàng trực tuyến lên đến hơn 51 triệu, tăng 13,5% so với năm trước, tổng chi tiêu cho việc mua sắm trực tuyến đạt 12,42 tỷ USD. Theo đó, Việt Nam là quốc gia đứng đầu với số lượng mua hàng trực tuyến trung bình lên đến 104 đơn hàng/năm, 73% đáp viên cho biết họ thường xuyên mua hàng trên các nền tảng mua sắm thương mại điện tử và 59% cho biết họ đã từng nhiều lần đặt hàng hoặc mua sắm trên các website quốc tế. Ngoài ra, Việt Nam hiện đang chiếm 15% tổng thị trường mua sắm trực tuyến tại Đông Nam Á, chỉ đứng sau Thái Lan với tỷ lệ 16% và ngang bằng với Philippines. Báo cáo cho thấy người Việt Nam yêu thích việc mua sắm online và đang dẫn đầu khu vực ở nhiều chỉ số. Theo một báo cáo khác từ Statista, Việt Nam dự kiến sẽ sở hữu thị trường thương mại điện tử lớn thứ 2 tại Đông Nam Á, chỉ sau Indonesia trước năm 2025.

Trước đây, chào hàng tiếp thị là một trong những chiến lược có giá trị nhất có thể được sử dụng, những ưu đãi này được đề xuất một cách không chọn lọc cho toàn bộ khách truy cập của một trang web thương mại điện tử nhất định. Tuy nhiên với tiềm năng phát triển vượt bật của thương mại điện tự đã được đề cập ở những báo cáo trên, thì việc tìm hiểu cách thức và thời điểm người dùng sẽ tìm kiếm và mua hàng trực tuyến là điều quan trọng đối với các doanh nghiệp vì họ có thể sử dụng thông tin chi tiết về hành vi của khách hàng để để thúc đẩy khách hàng tiềm năng hoàn thành mua hàng trực tuyến trong thời gian thực, tăng tỷ lệ chuyển đổi mua hàng tổng thể, đồng thời nhắm mục tiêu quảng cáo, tiếp thị và giao dịch cho khách hàng tiềm năng một cách phù hợp nhất nhằm tăng thêm doanh số và doanh thu của họ nói riêng và góp phần thúc đẩy thương mại điện tử phát triển nói chung.

Chính vì những lý do trên, nhóm chọn đã quyết định thực hiện đề tài: "***Nghiên cứu ý định mua hàng của người mua hàng trực tuyến***".

2. MỤC TIÊU THỰC HIỆN ĐỒ ÁN

Mục tiêu của đồ án chính là nghiên cứu và áp dụng Machine Learning vào trong việc dự đoán ý định mua hàng online của khách hàng dựa trên 18 features được thu thập thông qua trình

duyệt và thông tin của một website thương mại điện tử. Bộ dữ liệu lấy từ UCI Machine Learning Repository.

Để làm được điều này, nhóm đã thực hiện các nhiệm vụ sau:

- Nhóm huấn luyện mô hình và kiểm tra bài toán dựa trên 4 thuật toán Naive Bayes, K-nearest neighbor. Decision Tree, và Random Forest.
- Sau đó, nhóm đánh giá hiệu quả của từng thuật toán, tìm ra thuật toán tốt nhất để dự đoán ý định mua hàng.
- Dựa trên các dữ liệu đã phân tích, xác định các chỉ số chính đóng góp nhiều nhất vào việc dự đoán hành vi của người mua sắm
- Đề xuất một số cách có thể thu hút nhiều khách và cải thiện hiệu suất trong việc khách hàng đưa ra quyết định mua hàng.

Câu hỏi nghiên cứu

Để đạt được các mục tiêu đề ra, nhóm đặt ra các câu hỏi như sau:

1. Với dữ liệu phiên và clickstream của người dùng truy cập trang web thương mại điện tử, nhóm có thể dự đoán liệu khách truy cập đó có mua hàng hay không?
2. Giữa các thuật toán Decision Tree, Naive Bayes, KNN, Random Forest, thuật toán nào đem lại hiệu quả tốt hơn cho công ty?
3. Những thuộc tính tác động như thế nào tới việc thúc đẩy doanh thu của công ty? Thuộc tính nào có ảnh hưởng lớn tới kết quả của nghiên cứu?
4. Bounce Rate và Exit Rate tác động tới lợi nhuận của công ty như thế nào?
5. Thời lượng dành ra truy cập trang web dài hơn sẽ ảnh hưởng đến Bounce Rate (tỷ lệ thoát) như thế nào?

Với sự phát triển mạnh mẽ của thương mại điện tử, việc trả lời câu hỏi này là rất quan trọng đối với các công ty để đảm bảo rằng họ có thể duy trì lợi nhuận. Thông tin này có thể được sử dụng để thúc đẩy khách hàng tiềm năng hoàn thành mua hàng trực tuyến trong thời gian thực, tăng tỷ lệ chuyển đổi mua hàng tổng thể.

Kết quả mong muốn đạt được

- Hiểu được lý thuyết về cách dự đoán ý định mua hàng theo Decision Tree, Naive Bayes, K-Nearest Neighbors, Random Forest.
- Xây dựng mô hình hiệu quả để dự đoán ý định mua hàng của khách của các công ty thương mại điện tử.
- Đưa ra các đề xuất phù hợp để tăng lợi nhuận công ty thông qua phân tích dữ liệu.
- Áp dụng mô hình dự đoán ý định mua hàng của khách hàng để thấu hiểu các yếu tố quan trọng tác động đến việc khách hàng đưa ra quyết định mua hàng. Từ đó giúp công ty đưa ra các chiến lược marketing thúc đẩy các khách hàng tiềm năng mua hàng trực tuyến, tăng tỉ lệ chuyển đổi.

3. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

3.1. Đối tượng

Đối tượng nghiên cứu của đồ án là người dùng internet truy cập vào trang web thương mại điện tử trong khoảng thời gian 1 năm.

3.2. Phạm vi

Phạm vi nghiên cứu của đồ án là 12.330 phiên (sessions) trong tập dữ liệu được lấy từ UC Irvine Machine Learning Repository, đây là một trang web phổ biến với hàng trăm bộ dữ liệu có sẵn để phân tích. Mỗi hàng trong tập dữ liệu chứa một vectơ đặc trưng chứa dữ liệu tương ứng với một "phiên" (khoảng thời gian đã sử dụng) của một người dùng trên một trang web thương mại điện tử.

Lý do chọn tập dữ liệu này: Tập dữ liệu được tạo cụ thể để mỗi phiên sẽ thuộc về một người dùng duy nhất trong khoảng thời gian 1 năm để trách bất kỳ xu hướng tác động nào có liên quan đến chiến dịch cụ thể nào đó, ngày đặc biệt, hồ sơ người dùng hay một giai đoạn nào đó. Dataset này chứa rất ít missing values và tất cả các features đều liên quan đến ý định mua hàng dựa trên suy luận.

4. PHƯƠNG PHÁP NGHIÊN CỨU

Phương pháp nghiên cứu tài liệu:

- Nghiên cứu những tài liệu, sách và bài báo khoa học liên quan để khảo sát tình hình nghiên cứu nước ngoài, trong nước về dự đoán ý định mua hàng của khách hàng trên nền tảng website thương mại điện tử.
- Tìm hiểu các phương pháp khai thác dữ liệu, nghiên cứu và tiếp thu những mô hình, kiến thức, công nghệ mới liên quan đến phân tích dữ liệu.

Phương pháp thực nghiệm: Xây dựng các mô hình để phân tích và so sánh dữ liệu đã thu thập nhằm để tìm ra yếu tố nào tác động nhiều đến việc dự đoán ý định mua hàng của khách hàng. Từ đó đưa ra đề xuất, hướng phát triển.

5. CÔNG CỤ VÀ NGÔN NGỮ LẬP TRÌNH SỬ DỤNG

Nhóm sử dụng ngôn ngữ lập trình **Python**. Đây là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (Machine Learning - ML).

Lý do nhóm sử dụng Python là do:

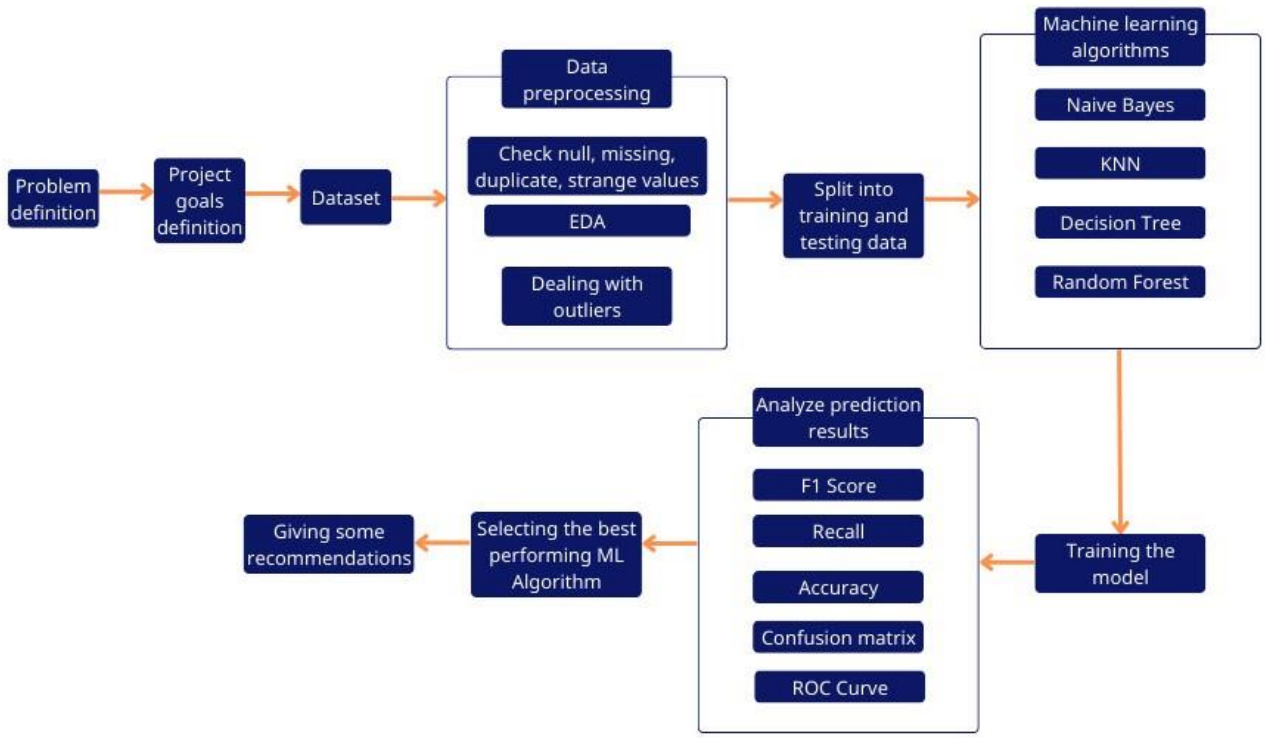
- Python có số lượng dòng mã ít hơn các ngôn ngữ lập trình khác để cho ra một kết quả tương tự, điều này giúp tiết kiệm thời gian hơn.
- Ngôn ngữ Python được xem là khá giống với ngôn ngữ tự nhiên của con người nên nó sẽ dễ viết và đọc hơn.
- Python có sẵn trên các hệ điều hành đa dạng khác nhau như Windows, Linux, Mac OS.
- Bên cạnh đó Python còn có một thư viện tiêu chuẩn lớn, với nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ như: Matplotlib, Pandas, Numpy,...

Hình 1. Ngôn ngữ Python (Nguồn: google.com)

Để sử dụng các thư viện này trên Python thì nhóm sẽ sử dụng công cụ **Google Colab**. Đây là sản phẩm đến từ Google cho phép thực thi Python trên nền tảng đám mây, nó không yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt. Vì lý do này, nhóm đã lựa chọn Google Colab để làm công cụ lập trình Python để thuận tiện cho làm việc nhóm.

Hình 2. Công cụ Google Colab (Nguồn: google.com)

6. QUY TRÌNH THỰC HIỆN ĐỒ ÁN



Hình 3. Quy trình thực hiện đồ án

7. CẤU TRÚC ĐỒ ÁN

Đồ án bao gồm … trang (bao gồm trang bìa). Ngoài phần lời cảm ơn, lời cam kết, các danh mục và phần tổng quan, nội dung đồ án được kết cấu thành 5 chương như sau:

Chương 1: Cơ sở lý thuyết

Chương 2: Tìm hiểu chung và khai phá dữ liệu (EDA)

Chương 3: Thực nghiệm và phân tích

Chương 4: Đánh giá và lựa chọn mô hình

Chương 5: Tổng kết

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

Trình bày khái quát tình hình nghiên cứu về ý định mua hàng trên thế giới; khái niệm, ý nghĩa, lợi ích của việc dự đoán ý định mua hàng. Trình bày khái niệm, cách tính công thức của các chỉ số đo lường trên website. Tổng quan về các thuật toán học máy có giám sát và các chỉ số đo lường kết quả thuật toán.

1. TỔNG QUAN VỀ VIỆC DỰ ĐOÁN Ý ĐỊNH MUA SẮM TRỰC TUYẾN CỦA KHÁCH HÀNG

Ý định mua hàng trực tuyến của khách hàng là một trong những lĩnh vực nghiên cứu chuyên sâu trong các tài liệu hiện có. Ý định mua hàng trực tuyến của khách hàng trong môi trường mua sắm trên web sẽ xác định sức mạnh ý định của người tiêu dùng để thực hiện một hành vi mua hàng cụ thể qua Internet (Salisbury, Pearson, Pearson and Miller, 2001). Hơn nữa, lý thuyết về hành động hợp lý (reasoned action) cho rằng hành vi của người tiêu dùng có thể được dự đoán từ những ý định tương ứng trực tiếp về hành động, mục tiêu và bối cảnh với hành vi của người tiêu dùng đó (Ajzen and Fishbein, 1980). Theo Day (1969), các biện pháp có chủ ý (intentional measures) có thể hiệu quả hơn các biện pháp hành vi (behaviour measures) khi nắm bắt tâm trí khách hàng vì khách hàng có thể mua hàng do bị ràng buộc thay vì sở thích thực.

Ý định mua có thể được phân loại là một trong những thành phần của hành vi nhận thức của người tiêu dùng về cách một cá nhân dự định mua một thương hiệu cụ thể. Laroche, Kim and Zhou (1996) khẳng định rằng các biến số như cân nhắc mua một thương hiệu và kỳ vọng mua một thương hiệu có thể được sử dụng để đo lường ý định mua của người tiêu dùng. Theo Pavlou (2003), ý định mua hàng trực tuyến là tình huống khi một khách hàng sẵn sàng và có ý định tham gia vào giao dịch trực tuyến. Có thể coi giao dịch trực tuyến là một hoạt động trong đó diễn ra quá trình truy xuất thông tin, chuyển thông tin và mua bán sản phẩm Pavlou (2003). Các bước truy xuất và trao đổi thông tin được coi là ý định sử dụng một trang web; tuy nhiên, việc mua sản phẩm được áp dụng nhiều hơn cho ý định xử lý một trang web (Pavlou, 2003). Do

đó, biết được khả năng khách hàng sẽ mua hàng thông qua các lần truy cập trang web là một yếu tố cần thiết để doanh nghiệp có thể tối ưu hóa website, tối ưu hóa quảng cáo và sản phẩm của mình để tăng trải nghiệm khách hàng, từ đó tăng tỷ lệ chuyển đổi và giúp đạt hiệu quả doanh thu cao.

Bên cạnh đó, còn có một số bài báo nhằm mục đích hiểu phân loại học máy cơ bản khác nhau và phân loại ý định của người dùng trong trường hợp mua hàng trực tuyến. Moe (2003) đã cố gắng phân loại khách hàng dựa trên hành vi của họ trên trang web cửa hàng trực tuyến. Luồng nhấp chuột của người dùng được thu thập và dựa vào đó ý định của họ được phân loại dựa trên niềm tin rằng các hoạt động của người dùng trong trang web cửa hàng trực tuyến phụ thuộc vào ý định của họ. Một tập hợp các tính năng của hoạt động người dùng đã được thu thập và đưa vào cụm Kmeans để phân loại người dùng. Một tập hợp các thuộc tính (features) của hoạt động người dùng đã được thu thập và đưa vào cụm Kmeans để phân loại người dùng. Các danh mục là “Direct buying”, ”Knowledge building”, ”Search”, ”Shallow” trong đó mua trực tiếp (Direct buying) là những người dùng trực tiếp truy cập trang và mua một mặt hàng. Mặt khác, người dùng rời khỏi trang web sau khi truy cập 2 trang được phân loại là người dùng nông (Shallow user). Với Mobasher (2002), hai cơ chế phân cụm khác nhau được sử dụng để xây dựng hồ sơ người dùng dựa trên lịch sử giao dịch và lượt xem trang của người dùng mà theo đó hành động thời gian thực có thể được thực hiện đối với người dùng đó để tăng khả năng nắm bắt được người dùng. Tương tự như Moe (2003), ở đây lượt kích chuột của người dùng cũng được thu thập để phân tích. Trong nghiên cứu của Mobasher (2002), khách hàng trên sàn thương mại điện tử được phân loại dựa trên dữ liệu lịch sử của họ trên máy chủ. Từ dữ liệu phiên, một số tính năng đã được thu thập để xác định các tính năng quan trọng nhất cho biết xác suất mua hàng cao. Trong một nghiên cứu khác, dự đoán về việc liệu khách hàng có mua bất kỳ sản phẩm nào vào cuối phiên hay không được dự đoán bằng cách sử dụng mức độ phổ biến của sản phẩm và dữ liệu tạm thời. Trong đó, người ta đã phát hiện ra rằng những khách truy cập có hồ sơ tại website, ý định của họ rất dễ dự đoán vì lịch sử trước đây của họ có sẵn. Nhưng thật khó để xác định ý định của bất kỳ khách hàng mới nào. Vì vậy, dữ liệu truy cập tạm thời được sử dụng cùng với mức độ phổ biến của sản phẩm để dự đoán ý định của họ. Theo Shekasta (2019), một hệ thống khuyến nghị đã được phát

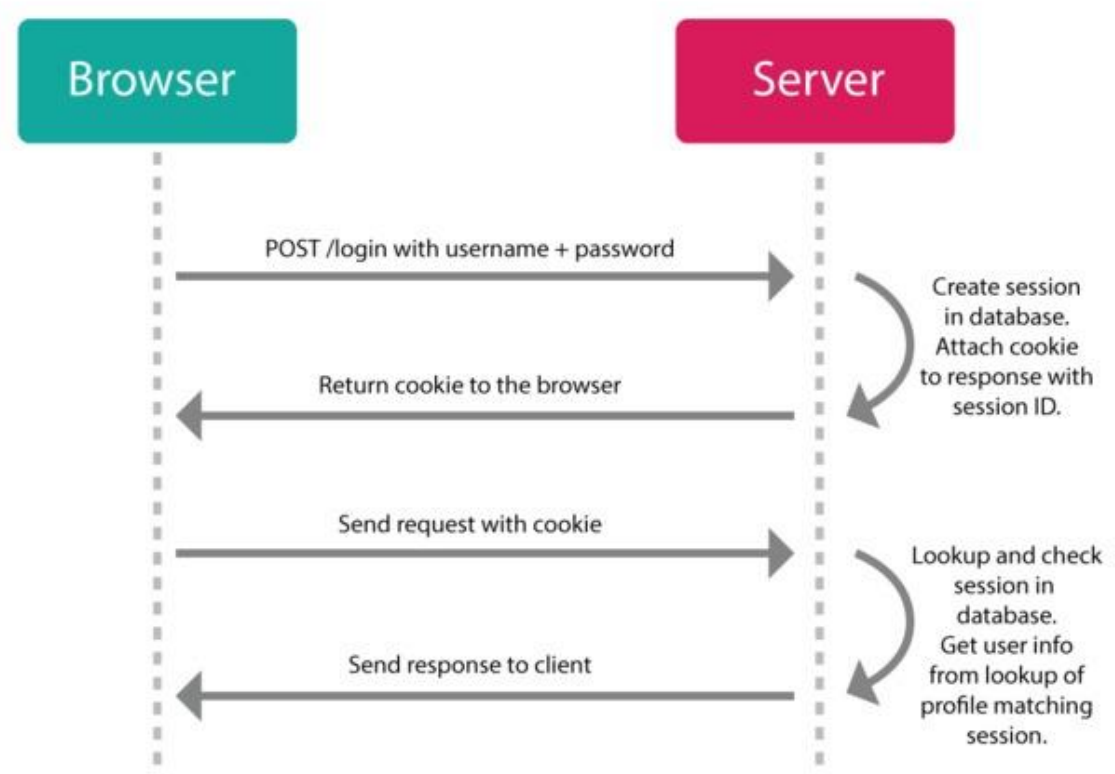
triển nhằm xác định ý định của người dùng mới và đưa ra các khuyến nghị về sản phẩm cho người dùng theo đó. Trong trường hợp này, một thuật toán dựa trên nội dung đã được sử dụng cho cả việc lập mô hình nội dung và dự đoán ý định của khách hàng. Họ đã sử dụng cơ sở học sâu trong thuật toán của mình.

Không giống như những phương pháp này, nhóm đã sử dụng phương pháp học máy có giám sát để xác định xem khách hàng mua một mặt hàng hay không.

2. MỘT SỐ THUẬT NGỮ LIÊN QUAN ĐẾN HÀNH VI TRUY CẬP TRÊN TRANG WEB CỦA KHÁCH HÀNG

2.1. Session

Session là một phiên làm việc, là một khái niệm phổ biến được dùng trong lập trình web có kết nối với database. Session bao gồm toàn bộ những dữ liệu xuyên suốt quá trình người dùng thao tác trên trang web hoặc web app. Mỗi khi có một người dùng truy cập vào website, họ sẽ được cấp 1 ID để bắt đầu một Session, các tài nguyên, dữ liệu này sẽ được lưu trữ lại trong ID Session được cấp.



Hình 4. Cơ chế hoạt động của Session

Tùy theo mỗi trang web, app, các nguồn tài nguyên/ dữ liệu có thể là:

- Những trang web người dùng đã xem
- Những thông tin mà người dùng đã nhập vào biểu mẫu
- Các mặt hàng người dùng vừa xem trên trang thương mại điện tử, giỏ hàng.

Một session được bắt đầu khi client gửi yêu cầu đến máy chủ (server), nó tồn tại xuyên suốt từ trang này đến trang khác trong ứng dụng web và chỉ dừng lại khi hết thời gian timeout hoặc khi người dùng đóng ứng dụng. Giá trị của các session sẽ được lưu trữ trong một file trên server.

Session là một chỉ số quan trọng trong Google Analytics. Những chỉ số mà Google Analytics phân tích phụ thuộc rất nhiều vào chỉ số của session.

2.2. Bounce Rate và Exit Rate

Thông thường, doanh nghiệp sẽ dựa vào các số liệu thu thập từ trang web để có thể dự đoán được tỉ lệ khách hàng sẽ mua món hàng đó sau vài lần xem. Các tỉ lệ thường dùng là Exit rate (Tỷ lệ thoát trang), Bounce rate (Tỷ lệ bỏ trang),… Thông qua các tỷ lệ đó, doanh nghiệp sẽ biết được khách hàng ghé xem món hàng nào nhiều nhất, xem chúng trong bao lâu và có hành động gì trên trang để thông qua đó dự đoán xem khách hàng này sẽ mua hay không.

Các thông tin về cách tính của các chỉ số liên quan:

$$Exite\ Rate\ (site) = \frac{Number\ of\ exits}{Total\ number\ of\ Pageviews}$$

$$Bounce\ Rate\ (site) = \frac{Numbe\ of\ bounces}{Total\ number\ of\ Visits}$$

Tỷ lệ bỏ trang (Bounce Rate) là tỷ lệ phần trăm số session (phiên truy cập) chỉ truy cập 1 trang duy nhất của người dùng và không có thêm tương tác nào khác trên trang (visitor engagement). Tỷ lệ bỏ trang cho thấy sự yếu kém về chất lượng nội dung hoặc hình thức của trang web hoặc khả năng điều hướng người dùng chưa tốt của trang web.

Tỷ lệ thoát (Exit Rate) là tỷ lệ phần trăm số người truy cập thoát khỏi website thông qua các webpages khác nhau. Nghĩa là họ đã truy cập vài trang khác nhau trên website rồi out ở một trang nào đó hay lượng Pageview lúc này đã lớn hơn 1. Tỷ lệ thoát trang cho thấy trang web đó đã đánh mất sự thu hút đối với người dùng tại đâu và chất lượng trang web đó với các trang web trước.

2.3. Page Value

Page Value hay còn gọi là Giá trị trang là một trong những chỉ số Google Analytic quan trọng trong SEO và đặc biệt quan trọng đối với các nhà đầu tư SEO. Bởi thông qua chỉ số này, chủ đầu tư SEO có thể tính toán hiệu quả của việc đầu tư SEO một cách chính xác, từ đó có thể đưa ra quyết định đầu tư cho các dự án SEO.

Chỉ số Page Value cho người xem thấy được giá trị trung bình mà trang đã đóng góp vào doanh thu cho website mỗi khi có người dùng thực hiện một chuyển đổi trên trang web. Từ đó

giúp nhà đầu tư SEO đánh giá được tầm quan trọng của từng trang trên site trong việc tạo ra chuyển đổi cho toàn site. Đây cũng là một chỉ số được Google Analytic chú trọng và đem đến cho người dùng.

Ngoài ra khi kết hợp hai chỉ số Traffic và Page Value, nhà đầu tư SEO có thể đánh giá chất lượng các trang trên site một cách hiệu quả hơn. Các trang có lượng traffic cao trên site thể hiện chủ đề mà trang đang nói đến đã được nhiều người quan tâm và tìm kiếm. Tuy nhiên nếu tỷ lệ chuyển đổi không cao lại cho thấy chất lượng của trang đó chưa được tốt.

Việc kết hợp các chỉ số time on page, bounce rate và exit rate sẽ giúp người đọc có một cái nhìn toàn cảnh hơn về trang web này. Ngược lại, chủ đầu tư SEO cũng cần tăng cường sự điều hướng người dùng đến những trang có lượng traffic thấp nhưng lại có Page Value cao.

2.4. Traffic

Traffic là số lượng người truy cập vào website thông qua nhiều kênh khác nhau. Có thể nói rằng, mức độ tương tác, khách hàng tiềm năng, chuyển đổi và lợi nhuận là hoàn toàn phụ thuộc vào số lượng người dùng truy cập vào website. Doanh thu sẽ càng nhiều nếu lưu lượng người truy cập vào website càng cao.

3. TỔNG QUAN VỀ MÁY HỌC

3.1. Định nghĩa

Machine learning (ML) hay máy học là một nhánh của trí tuệ nhân tạo (AI), nó là một lĩnh vực nghiên cứu cho phép máy tính có khả năng cải thiện chính bản thân chúng dựa trên dữ liệu mẫu (training data) hoặc dựa vào kinh nghiệm (những gì đã được học). Machine learning có thể tự dự đoán hoặc đưa ra quyết định mà không cần được lập trình cụ thể.

Bài toán machine learning thường được chia làm hai loại là dự đoán (prediction) và phân loại (classification). Các bài toán dự đoán như dự đoán giá nhà, giá xe... Các bài toán phân loại như nhận diện chữ viết tay, nhận diện đồ vật…

Trong bài này, nhóm cố gắng dự đoán hành vi mua hàng của khách hàng, mà dự đoán này không thể giải quyết được dễ dàng bằng lập trình truyền thống. Vì vậy, nhóm sẽ sử dụng các

thuật toán machine learning để xây dựng một mô hình hoặc một thuật toán, nhằm giúp kết quả dự đoán ý định mua hàng của khách hàng đạt một mức độ chính xác cao nhất có thể.

3.2. Các loại máy học

Có rất nhiều cách phân loại machine learning, thông thường thì machine learning sẽ được phân làm hai loại chính sau:

3.2.1. Máy học có giám sát (Supervised Machine Learning)

Đây là cách huấn luyện một mô hình trong đó dữ liệu học có đầu vào và đầu ra tương ứng đầu vào đó.Trong quá trình huấn luyện, thuật toán sẽ tìm kiếm các mẫu trong dữ liệu tương quan với kết quả đầu ra mong muốn. Sau khi được huấn luyện, supervised learning sẽ nhận các đầu vào chưa được biết trước và sẽ quyết định xem gán nhãn nào cho đầu vào dựa vào dữ liệu huấn luyện đã được học trước đó. Mục tiêu của mô hình học có giám sát là dự đoán nhãn chính xác cho dữ liệu đầu vào mới. Ở dạng cơ bản nhất, thuật toán học có giám sát có thể được viết đơn giản như sau:

$$y = f(x)$$

Trong đó y là đầu ra được dự đoán, được xác định bởi một hàm ánh xạ chỉ định một lớp cho giá trị đầu vào x. Hàm này được sử dụng để kết nối các features đầu vào với đầu ra được dự đoán. Học có giám sát có thể chia làm 2 loại:

- ***Phân loại (Classification):*** Trong quá trình đào tạo, một thuật toán phân loại sẽ được cung cấp các điểm dữ liệu với một danh mục (category). Công việc của thuật toán phân loại là lấy một giá trị đầu vào và gán cho nó một lớp, hoặc danh mục mà nó phù hợp dựa trên dữ liệu đào tạo được cung cấp. Mô hình sẽ tìm thấy mối tương quan giữa các đối tượng trong dữ liệu và lớp để tạo hàm ánh xạ đã đề cập trước đó: $y = f(x)$.
- ***Hồi quy (Regression):*** Hồi quy là một quá trình thống kê dự đoán, trong đó mô hình cố gắng tìm ra mối quan hệ quan trọng giữa các biến phụ thuộc và độc lập. Mục tiêu của một hồi quy thuật toán là dự đoán bằng cách xác định các yếu tố tạo ra tác động đến kết quả của dự đoán. Trong một mô hình hồi quy, chúng ta có các biến phụ thuộc và các biến độc lập. Biến phụ thuộc là các yếu tố chính có tác động đến kết quả trong khi các biến độc lập

là biến mà nghi ngờ có tác động đến các biến phụ thuộc. Phương trình tuyến tính cơ bản hồi quy có thể được viết như sau:

$$y = w[0] * x[0] + w[1] * x[1] + \cdots + w[i] * x[i] + b$$

Trong đó $x[i]$ là các cột dữ liệu, $w[i]$ và b là các tham số được phát triển trong quá trình huấn luyện. Đối với các mô hình hồi quy tuyến tính đơn giản chỉ có một cột dữ liệu, công thức sẽ là: $y = mx + c$

3.2.2. Máy học không có giám sát (Unsupervised Machine Learning)

Đây là cách huấn luyện một mô hình trong đó dữ liệu học chỉ bao gồm đầu vào mà không có đầu ra. Mô hình sẽ được huấn luyện cách để tìm cấu trúc hoặc mối quan hệ giữa các đầu vào, sử dụng các thuật toán học máy để phân tích và phân cụm các tập dữ liệu không được gắn nhãn. Các thuật toán này khám phá các mẫu ẩn hoặc phân nhóm dữ liệu mà không cần sự can thiệp của con người. Khả năng phát hiện ra những điểm tương đồng và sự khác biệt trong thông tin làm cho nó trở thành giải pháp lý tưởng để phân tích dữ liệu khám phá, chiến lược cross-selling, phân khúc khách hàng và nhận diện hình ảnh. Hai trong số các phương pháp chính được sử dụng trong học tập không giám sát là principal component và cluster analysis.

Phân tích cụm (cluster analysis) được sử dụng trong học tập không giám sát để nhóm hoặc phân đoạn, tập dữ liệu với các thuộc tính được chia sẻ để ngoại suy các mối quan hệ thuật toán. Nói cách khác, phân cụm liên quan đến việc tìm kiếm một cấu trúc trong một tập hợp dữ liệu không được gắn nhãn bằng cách tìm các nhóm riêng biệt trong tập dữ liệu. Để phân cụm, chúng ta cần xác định một thước đo khoảng cách cho hai điểm dữ liệu. Thuật toán clustering có thể phân loại như sau:

- ***Phân cụm độc quyền (Exclusive clustering):*** dữ liệu được nhóm theo một cách độc quyền, do đó một điểm chỉ thuộc về một cụm xác định. K-mean clustering là một trong những ví dụ về thuật toán exclusive clustering.
- ***Phân cụm chồng chéo (Overlapping clustering):*** sử dụng các tập mờ để phân cụm dữ liệu, vì vậy mỗi điểm có thể thuộc về hai hoặc nhiều cụm với các cấp độ thành viên khác nhau.

- ***Phân cụm theo thứ bậc (Hierarchical clustering):*** thuật toán hierarchical clustering có 2 loại:
    - *Phân cụm kết tụ (agglomerative clustering):* ban đầu coi mỗi điểm dữ liệu (data point) như một cụm (cluster) và sau đó kết hợp hai cụm gần nhất thành một cụm duy nhất cho đến khi tất cả các cụm đã được hợp nhất thành một cụm duy nhất chứa tất cả dữ liệu.
    - *Phân cụm phân chia (divisive clustering):* còn được gọi là cách tiếp cận từ trên xuống (top – down approach) là một cách tiếp cận của phân cụm phân cấp (hierarchical cluster). Phân cụm phân chia (divisive clustering) bắt đầu với một cụm (cluster) chứa toàn bộ dữ liệu và sau đó tách cụm được thực hiện đệ quy cho đến khi dữ liệu riêng lẻ được tách thành các cụm duy nhất.
- ***Phân cụm xác suất (Probabilistic clustering):*** Ví dụ: Mixture của Gaussian, sử dụng phương pháp tiếp cận theo xác suất.

Một số thuật toán phân cụm thông thường có thể kể đến: K – means, Fuzzy K-means, Mixture of Gaussians.

3.3. Lựa chọn giữa máy học có giám sát và máy học không giám sát

Ở đây, có thể thấy rằng vấn đề cần giải quyết là phải dự đoán được label cho class "Revenue", bằng cách sử dụng những feature đã được dán nhãn trước đó. Vì vậy có thể xem như đây là một bài toán học có giám sát. Áp dụng các thuật toán học có giám sát sẽ phù hợp hơn thay vì học không giám sát vì trong trường hợp của học không giám sát, dữ liệu không được dán nhãn, và mục đích là phải tìm ra cấu trúc của dữ liệu.

4. TỔNG QUAN VỀ THUẬT TOÁN NAIVE BAYES

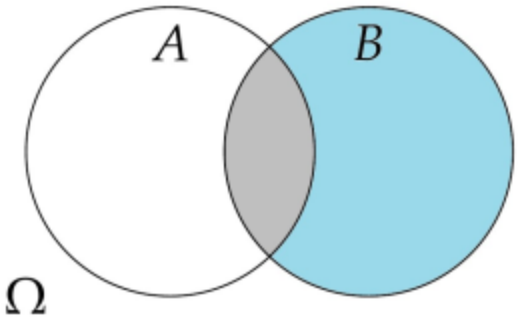
4.1. Khái niệm

Naive Bayes Classification (NBC) – thuật toán phân loại Naive Bayes - là một thuật toán dựa trên định lý Bayes về lý thuyết xác suất để đưa ra các phán đoán cũng như phân loại dữ liệu dựa trên các dữ liệu được quan sát và thống kê, được ứng dụng rất nhiều trong các lĩnh vực Machine learning dùng để đưa các dự đoán có độ chính xác cao, dựa trên một tập dữ liệu đã được thu thập. NBC thuộc vào nhóm học máy có giám sát.

4.2. Định lý Bayes

4.2.1. Xác suất có điều kiện

Ta có, vì biến cố B đã xảy ra, nên không gian mẫu của phép thử lúc này thu hẹp lại là tập các kết quả có thể xảy ra thuộc biến cố B. Do đó, để tính toán xác suất xảy ra biến cố A sau đó, ta cần xem xét các kết quả của biến cố A thuộc B. Đây chính là tập AB. Ta có thể minh họa điều này trong biểu đồ Venn ở hình sau.



Hình 5. Biểu đồ Venn của xác suất có điều kiện

Xác suất của biến cố A với điều kiện biến cố B đã xảy ra ($P(B) > 0$), ký hiệu $P(A|B)$ được tính theo công thức:

$$P(A|B) = \frac{P(AB)}{P(B)}$$

Tương tự, xác suất của biến cố B với điều kiện biến cố A đã xảy ra là:

$$P(B|A) = \frac{P(BA)}{P(A)}$$

Nếu A và B là hai biến cố độc lập thì:

$$P(A|B) = P(A) \; hoặc \; P(B|A) = P(B)$$
$$=> P(AB) = P(A).P(B)$$

4.2.2. Định lý Bayes

Cho $\{A_i, i = \overline{1,n}\}$ là một hệ đầy đủ, B là một biến cố tùy ý sao cho $P(B) > 0$. Khi đó ta có:

$$P(A_k|B) = \frac{P(BA_k)}{P(B)} = \frac{P(A_k)P(B|A_k)}{\sum_{i=1}^{n} P(A_i)P(B|A_i)}, k = \overline{1,n}$$

4.2. Naive Bayes

Áp dụng công thức Bayes ở trên ta có như sau:

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)}$$

Cho $\{A_i, i = \overline{1,n}\}$ là một hệ đầy đủ, thay thế vào công thức trên ta có:

$$P(B|A_i) = \frac{P(B)P(A_i|B)}{P(A_i)}$$

Dựa vào xác suất có điều kiện chúng ta có thể triển khai như sau:

$$P(A_1, \dots, A_n|B) = P(A_1|B, A_2, \dots, A_n)P(A_2, \dots, A_n|B)$$

$$P(A_1, \dots, A_n) = P(A_1|A_2, \dots, A_n)P(A_2, \dots, A_n)$$

Naive Bayes cho chúng ta một giả thuyết rằng tất cả các biến ngẫu nhiên A_i là độc lập với nhau. Vì vậy ta có thể suy ra như sau:

$$P(A_1|B, A_2, \dots, A_n) = P(A_1|B)$$

$$P(A_1|A_2, \dots, A_n) = P(A_1)$$

Công thức ban đầu sẽ trở thành:

$$P(B|A_1, \dots, A_n) = \frac{P(B)\prod_{i=1}^{n} P(A_i|B)}{P(A_1) \dots P(A_n)}$$

Để ước lượng xác suất của tất cả giá trị B với đầu vào A, ta có thể thấy rằng $P(A) = P(A_1) \dots P(A_n)$ là một hằng số. Vì vậy, để tìm được xác suất lớn nhất của giá trị B với đầu vào A ta có công thức sau:

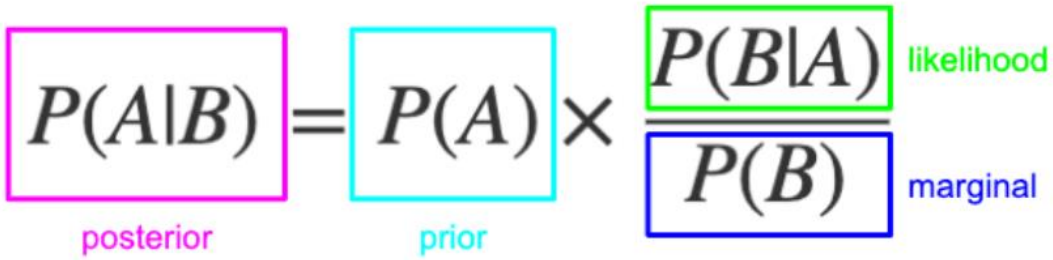
$$P(B|A_1, \dots, A_n) \propto P(B)\prod_{i=1}^{n} P(A_i|B)$$

$$\hat{B} = \underset{B}{argmax}\ (P(B)\prod_{i=1}^{n} P(A_i|B))$$

Với: $\propto$ là phép tính tỉ lệ (proportion)

Trên thực tế thì ít khi tìm được dữ liệu mà các thành phần là hoàn toàn độc lập với nhau. Tuy nhiên giả thiết này giúp cách tính toán trở nên đơn giản, training data nhanh, đem lại hiệu quả bất ngờ với các lớp bài toán nhất định.

4.3. Các bộ phận cấu thành



Hình 6. Định lý Bayes (Nguồn: canhminhdo.github.io)

Posterior probability (Xác suất hậu nghiệm): Xác suất được cập nhật sau đó khi có thêm thông tin từ mẫu. Hay là xác suất của A khi biết B đã xảy ra.

Prior probability (Xác suất tiên nghiệm): Xác suất ban đầu khi chưa có thêm thông tin từ mẫu. Nghĩa là xác suất của A khi chưa có điều kiện B đã xảy ra.

Likelihood hay Conditional probability (Xác suất có điều kiện): Là các hàm tham số của bất kỳ mô hình thống kê nào giúp mô tả xác suất chung của dữ liệu được quan sát. Nói ngắn gọn là xác suất của B khi biết rằng A đúng.

Marginal hay Predictor prior probability hay Evidence (Xác suất cận biên): Xác suất xảy ra thông tin được thêm từ mẫu. Tức là xác suất để xảy ra biến cố B.

4.4. Ưu nhược điểm của thuật toán Naive Bayes

Ưu điểm:

- Dễ dàng và nhanh chóng để dự đoán lớp của tập dữ liệu thử nghiệm. Nó cũng hoạt động tốt trong dự đoán nhiều lớp
- Khi giả định giữ độc lập, bộ phân loại Naive Bayes hoạt động tốt hơn so với các mô hình khác như hồi quy logistic và bạn cần ít dữ liệu đào tạo hơn.
- Nó hoạt động tốt trong trường hợp các biến đầu vào phân loại so với (các) biến số. Đối với biến số, phân phối chuẩn được giả định (đường cong hình chuông, một giả định mạnh).

Nhược điểm:

- Nếu biến phân loại có một danh mục (trong tập dữ liệu thử nghiệm), không được quan sát trong tập dữ liệu huấn luyện, thì mô hình sẽ chỉ định xác suất 0 (không) và sẽ không thể đưa ra dự đoán. Điều này thường được gọi là "Tần số không". Để giải quyết vấn đề này, chúng ta có thể sử dụng kỹ thuật làm mịn. Một trong những kỹ thuật làm mịn đơn giản nhất được gọi là ước lượng Laplace.
- Mặt khác, Naive Bayes cũng được biết đến như một công cụ ước lượng không tốt, vì vậy kết quả xác suất từ dự đoán không được quá coi trọng.
- Một hạn chế khác của Naive Bayes là giả định về các yếu tố dự đoán độc lập. Trong cuộc sống thực, hầu như không thể có được một tập hợp các yếu tố dự đoán hoàn toàn độc lập.

4.5. Ứng dụng của thuật toán Naive Bayes

Real time Prediction (Dự đoán thời gian thực): NBC chạy khá nhanh nên nó thích hợp áp dụng ứng dụng nhiều vào các ứng dụng chạy thời gian thực, như hệ thống cảnh báo phát hiện sự cố...

Multi class Prediction (Dự đoán đa lớp): Nhờ vào định lý Bayes mở rộng ta có thể ứng dụng vào các loại ứng dụng đa dự đoán, tức là ứng dụng có thể dự đoán nhiều giả thuyết mục tiêu.

Text classification/ Spam Filtering/ Sentiment Analysis (Phân loại văn bản / Lọc thư rác / Phân tích cảm xúc): NBC cũng rất thích hợp cho các hệ thống phân loại văn bản hay ngôn ngữ tự nhiên vì tính chính xác của nó lớn hơn các thuật toán khác. Ngoài ra các hệ thống chống thư rác cũng rất ưa chuộng thuật toán này. Và các hệ thống phân tích tâm lý thị trường cũng áp

dụng NBC để tiến hành phân tích tâm lý người dùng ưa chuộng hay không ưa chuộng các loại sản phẩm nào từ việc phân tích các thói quen và hành động của khách hàng.

Recommendation System (Hệ thống gợi ý): Naive Bayes Classifier được sử dụng rất nhiều để xây dựng hệ thống gợi ý.

5. TỔNG QUAN VỀ THUẬT TOÁN KNN (K – NEAREST NEIGHBORS)

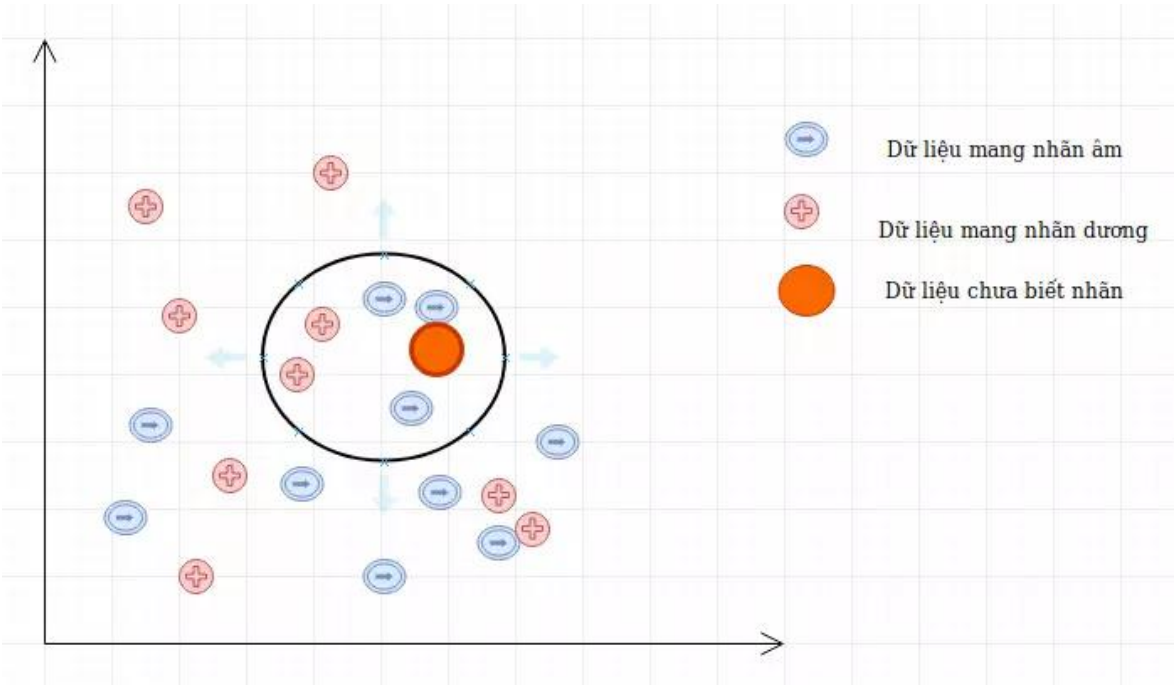
5.1. Khái niệm

Thuật toán KNN (K – Nearest Neighbor - K láng giềng gần nhất) là một kĩ thuật học có giám sát (supervised learning) dùng để phân loại quan sát mới bằng cách tìm điểm tương đồng giữa quan sát mới này với dữ liệu sẵn có.

Lớp (nhãn) của một đối tượng dữ liệu mới có thể được dự đoán từ các lớp (nhãn) của chỉ số k hàng xóm gần nó nhất.

Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. KNN có thể áp dụng được vào cả hai loại của bài toán Supervised learning là phân lớp và hồi quy.

5.2. Quy trình thực hiện



Hình 7. Ví dụ về KNN(K-Nearest Neighbor) (Nguồn: Slideshare.net)

Bước 1: Xác định tham số K= số láng giềng gần nhất.

Bước 2: Đo khoảng cách (Euclidean, Manhattan, Minkowski hoặc Trọng số - các công thức toán học) từ dữ liệu cần phân lớp đến tất cả các dữ liệu khác đã được phân loại.

$$Euclidean = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

$$Manhattan = \sum_{i=1}^{k}|x_i - y_i|$$

$$Minkowski = \left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{\frac{1}{q}}$$

Bước 3: Sắp xếp khoảng cách theo thứ tự tăng dần và xác định K đối tượng gần nhất với đối tượng cần phân lớp.

Bước 4: Lấy tất cả các lớp của K láng giềng gần nhất.

Bước 5: Dựa vào phần lớn lớp của K để xác định lớp cho đối tượng cần phân lớp.

5.3. Ưu, nhược điểm của thuật toán KNN

Ưu điểm:

- Thuật toán đơn giản, dễ dàng triển khai.
- Độ phức tạp tính toán nhỏ.
- Xử lý tốt với tập dữ liệu nhiễu

Nhược điểm:

- Với K nhỏ dễ gặp nhiễu dẫn tới kết quả đưa ra không chính xác
- Cần nhiều thời gian để thực hiện do phải tính toán khoảng cách với tất cả các đối tượng trong tập dữ liệu.
- Cần chuyển đổi kiểu dữ liệu thành các yếu tố định tính.

5.4. Ứng dụng của thuật toán KNN

KNN là một mô hình đơn giản và trực quan nhưng vẫn có hiệu quả cao vì nó không tham số; mô hình không đưa ra giả định nào về việc phân phối dữ liệu. Hơn nữa, nó có thể được sử dụng trực tiếp để phân loại đa lớp.

Thuật toán KNN có nhiều ứng dụng trong ngành đầu tư, bao gồm dự đoán phá sản, dự đoán giá cổ phiếu, phân bổ xếp hạng tín dụng trái phiếu doanh nghiệp, tạo ra chỉ số vốn và trái phiếu tùy chỉnh.

6. TỔNG QUAN VỀ THUẬT TOÁN DECISION TREE

6.1. Khái niệm

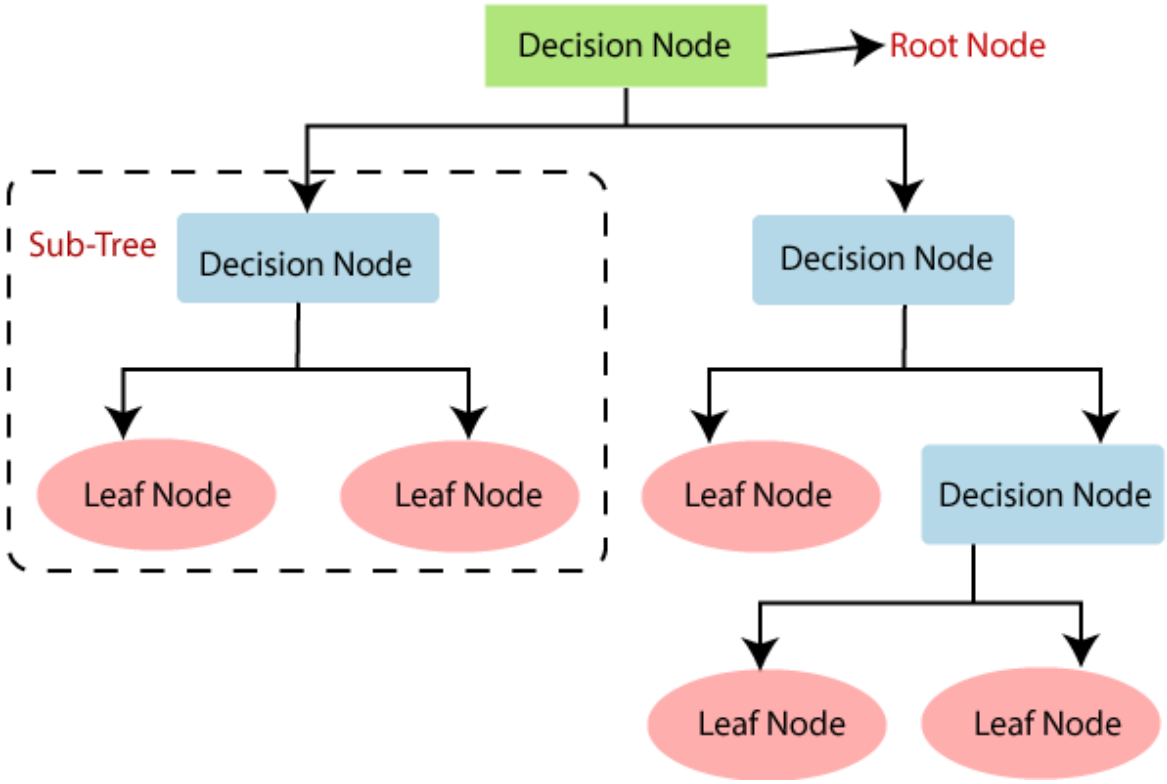
Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary), Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.

Tóm lại, cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết.

Decision Tree là thuật toán học có giám sát, có thể giải quyết cả bài toán hồi quy và phân loại.

Một cây quyết định bao gồm các thành phần sau:

- *Root node:* điểm ngọn chứa giá trị của biến đầu tiên được dùng để phân nhánh.
- Internal node: các điểm bên trong thân cây là các biến chứa các giá trị dữ liệu được dùng để xét cho các phân nhánh tiếp theo
- *Leaf node:* là các lá cây chứa giá trị của biến phân loại sau cùng.
- *Branch/sub tree*: là quy luật phân nhánh, nói đơn giản là mối quan hệ giữa giá trị của biến độc lập (Internal node) và giá trị của biến mục tiêu (Leaf node).



Hình 8. Cấu trúc của một cây quyết định (Nguồn: Javapoint.com)

6.2. Thuật toán Cây quyết định

6.2.1. Giải thuật ID3

Giải thuật ID3 (gọi tắt là ID3) Được phát triển đồng thời bởi Quinlan trong AI và Breiman, Friedman, Olsen và Stone trong thống kê. ID3 là một giải thuật học đơn giản nhưng tỏ ra thành công trong nhiều lĩnh vực. ID3 là một giải thuật hay vì cách biểu diễn tri thức học được của nó, tiếp cận của nó trong việc quản lý tính phức tạp, heuristic của nó dùng cho việc chọn lựa các khái niệm ứng viên, và tiềm năng của nó đối với việc xử lý dữ liệu nhiễu.

ID3 biểu diễn các khái niệm (concept) ở dạng các cây quyết định (decision tree). Biểu diễn này cho phép chúng ta xác định phân loại của một đối tượng bằng cách kiểm tra các giá trị của nó trên một số thuộc tính nào đó.

Như vậy, nhiệm vụ của giải thuật ID3 là học cây quyết định từ một tập các ví dụ rèn luyện (training example) hay còn gọi là dữ liệu rèn luyện (training data).

Input: Một tập hợp các ví dụ. Mỗi ví dụ bao gồm các thuộc tính mô tả một tình huống, hay một đối tượng nào đó, và một giá trị phân loại của nó.

Output: Cây quyết định có khả năng phân loại đúng đắn các ví dụ trong tập dữ liệu rèn luyện, và hy vọng là phân loại đúng cho cả các ví dụ chưa gặp trong tương lai.

6.2.2. Entropy

Entropy là thuật ngữ thuộc Nhiệt động lực học, là thước đo của sự biến đổi, hỗn loạn hoặc ngẫu nhiên. Năm 1948, Shannon đã mở rộng khái niệm Entropy sang lĩnh vực nghiên cứu, thống kê với công thức như sau:

Với một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau $x_1, x_2, x_3, \dots, x_n$.

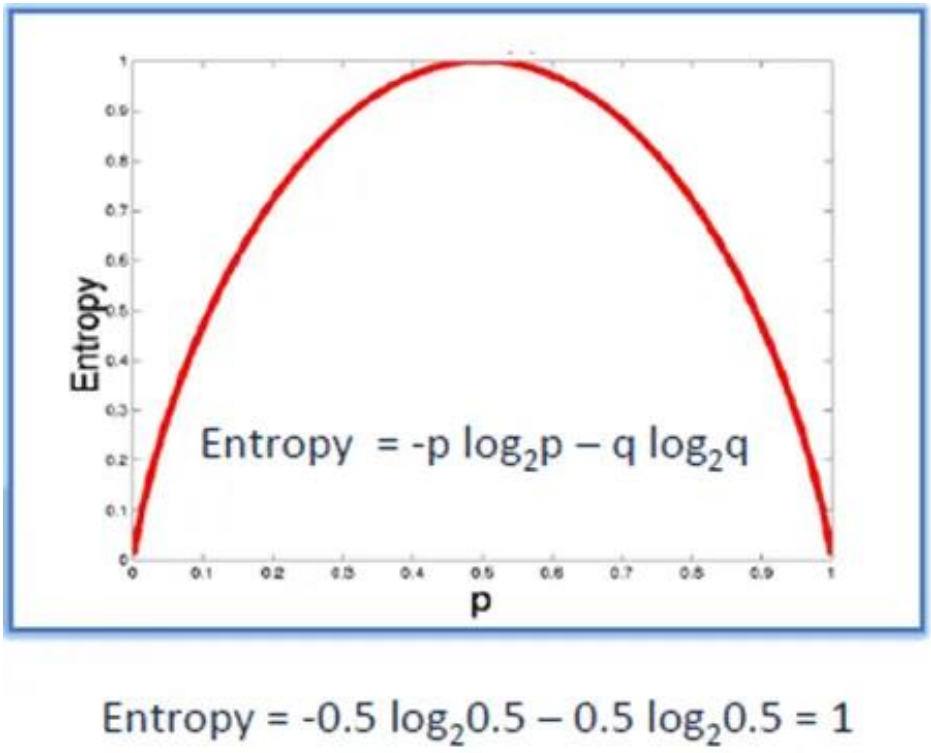
Giả sử rằng xác suất để x nhận các giá trị này là $p_i = p(x = x_i)$.

Ký hiệu phân phối này là $p = p_1, p_2, p_3, \dots, p_n$. Entropy của phân phối này được định nghĩa là:

$$H(p) = -\sum_{n=1}^{n} p_i \log(p_i)$$

Giả sử chúng ta tung một đồng xu, entropy sẽ được tính như sau:

$$H = -[0{,}5 * \ln(0{,}5) + 0{,}5 * \ln(0{,}5)]$$



Hình 9. Ví dụ minh họa về hàm Entropy

Hình vẽ trên biểu diễn sự thay đổi của hàm entropy. Ta có thể thấy rằng, entropy đạt tối đa khi xác suất xảy ra của hai lớp bằng nhau.

- P tinh khiết: $p_i = 0$ hoặc $p_i = 1$
- P vẩn đục: $p_i = 0{,}5$, khi đó hàm Entropy đạt đỉnh cao nhất

Kết luận giá trị entropy cực tiểu đạt được khi phân phối p là tinh khiết nhất, tức phân phối hoàn toàn thuộc về một nhóm. Trái lại, entropy cực đại đạt được khi toàn bộ xác suất thuộc về

các nhóm là bằng nhau. Một phân phối có entropy càng cao thì mức độ tinh khiết của phân phối đó sẽ càng thấp và ngược lại.

Như vậy về bản chất thì entropy là một thước đo về độ tinh khiết của phân phối xác suất. Dựa trên entropy chúng ta có thể đánh giá tính hiệu quả của câu hỏi ở mỗi node và quyết định xem đâu là câu hỏi hiệu quả hơn (có độ tinh khiết lớn hơn, entropy nhỏ hơn).

6.2.3. Chỉ số Ghini

Chỉ số Gini là một lựa chọn khác bên cạnh hàm entropy được sử dụng để đo lường mức độ bất bình đẳng trong phân phối của các lớp. Chỉ số này được tính bằng cách lấy 1 trừ đi tổng bình phương tỷ lệ phần trăm ở mỗi lớp.

$$Gini = 1 - \sum_{i=1}^{c} p_i^2$$

Gini thường được dùng đối với những biến rời rạc có số lượng các trường hợp là lớn vì nó có tốc độ tính toán nhanh hơn so với hàm entropy. Trong thuật toán CART của sklearn thì chỉ số gini được sử dụng thay cho hàm entropy.

6.2.4. Information Gain

Information Gain dựa trên sự giảm của hàm Entropy khi tập dữ liệu được phân chia trên một thuộc tính. Để xây dựng một cây quyết định, ta phải tìm tất cả thuộc tính trả về Information gain cao nhất.

Để xác định các nút trong mô hình cây quyết định, ta thực hiện tính Information Gain tại mỗi nút theo trình tự sau:

- ***Bước 1:*** Tính toán hệ số Entropy của biến mục tiêu S có N phần tử với Nc phần tử thuộc lớp c cho trước:

$$H(S) = -\sum_{c=1}^{c} \left(\frac{N_c}{N}\right) \log \left(\frac{N_c}{N}\right)$$

- ***Bước 2***: Tính hàm số Entropy tại mỗi thuộc tính: với thuộc tính x, các điểm dữ liệu trong S được chia ra K child node $S_1, S_2, \dots, S_K$ với số điểm trong mỗi child node lần lượt là $m_1, m_2, \dots, m_K$, ta có:

$$H(x,S) = \sum_{k=1}^{k} \left(\frac{m_K}{N}\right) * H(S_K)$$

- ***Bước 3:*** Chỉ số Information Gain được tính bằng:

$$G(x,S) = H(S) - H(x,S)$$

6.2.5. Thuật toán C4.5

Thuật toán C4.5 là thuật toán cải tiến của ID3.

Trong thuật toán ID3, Information Gain được sử dụng làm độ đo. Tuy nhiên, phương pháp này lại ưu tiên những thuộc tính có số lượng lớn các giá trị mà ít xét tới những giá trị nhỏ hơn. Do vậy, để khắc phục nhược điểm trên, ta sử dụng độ đo Gain Ratio (trong thuật toán C4.5) như sau:

- Đầu tiên, ta chuẩn hoá information gain với trị thông tin phân tách (split information):

$$Gain\ Ratio = \frac{Information\ Gain}{Slpit\ Info}$$

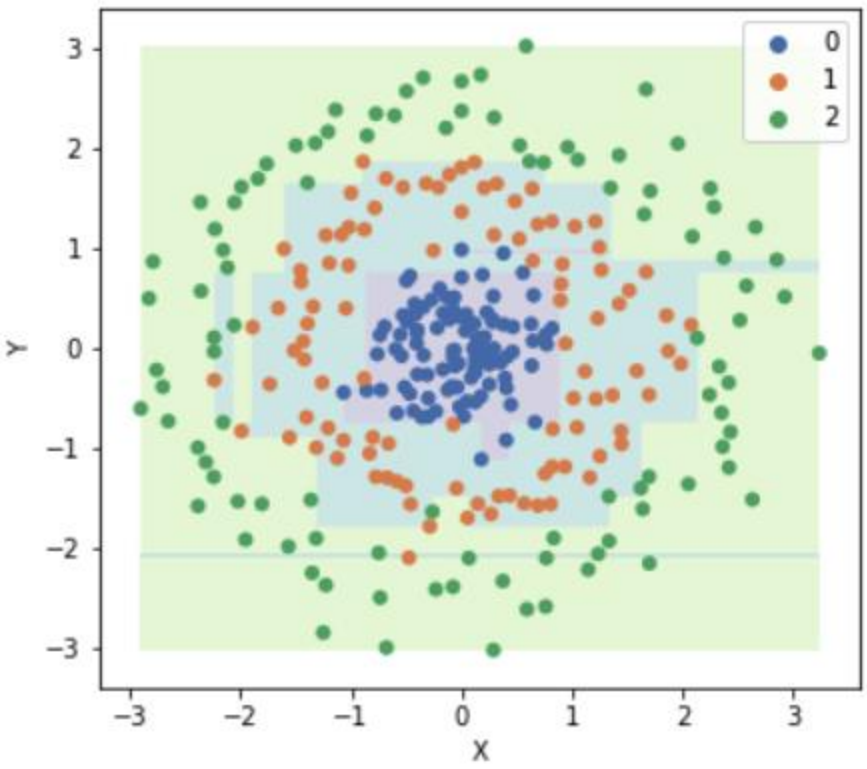
Trong đó: Split Info được tính như sau:

$$Slpit\ Info = -\sum_{i=1}^{n} D_i * \log_2(D_i)$$

- Giả sử chúng ta phân chia biến thành n nút con và D_i đại diện cho số lượng bản ghi thuộc nút đó. Do đó, hệ số Gain Ratio sẽ xem xét được xu hướng phân phối khi chia cây.

6.3. Overfitting

Các thuật toán Decision Tree nói chung nếu xây dựng cây quyết định đủ sâu thì sẽ tách được các node lá chỉ chứa dữ liệu một lớp nhất định, nên mô hình rất dễ bị overfitting.



Hình 10. Mô hình dữ liệu bị Overfitting (Nguồn: machinelearningcoban.com)

Mọi người thấy mô hình Decision Tree trên overfitting với dữ liệu, và tạo ra đường phân chia rất lạ. Thường có 2 cách giải quyết khi model Decision Tree bị overfitting:

- 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.
- Cắt tỉa (Prunning).

6.4. Tiêu chuẩn dừng

Trong các thuật toán Decision tree, với phương pháp chia trên, ta sẽ chia mãi các node nếu nó chưa tinh khiết. Như vậy, ta sẽ thu được một tree mà mọi điểm trong tập huấn luyện đều được dự đoán đúng (giả sử rằng không có hai input giống nhau nào cho output khác nhau). Khi đó, cây có thể sẽ rất phức tạp (nhiều node) với nhiều leaf node chỉ có một vài điểm dữ liệu. Như vậy, nhiều khả năng overfitting sẽ xảy ra.

Để tránh trường hợp này, ta có thể dừng cây theo một số phương pháp sau đây:

- Nếu node đó có entropy bằng 0, tức mọi điểm trong node đều thuộc một class.
- Nếu node đó có số phần tử nhỏ hơn một ngưỡng nào đó. Trong trường hợp này, ta chấp nhận có một số điểm bị phân lớp sai để tránh overfitting. Class cho leaf node này có thể được xác định dựa trên class chiếm đa số trong node.
- Nếu khoảng cách từ node đó đến root node đạt tới một giá trị nào đó. Việc hạn chế chiều sâu của tree này làm giảm độ phức tạp của tree và phần nào giúp tránh overfitting.
- Nếu tổng số leaf node vượt quá một ngưỡng nào đó.
- Nếu việc phân chia node đó không làm giảm entropy quá nhiều (information gain nhỏ hơn một ngưỡng nào đó).

6.5. Cắt tỉa

Đó là một phương pháp có thể giúp tránh overfitting. Nó giúp cải thiện hiệu suất của cây bằng cách cắt các nút hoặc nút con không quan trọng. Nó loại bỏ các nhánh có tầm quan trọng rất thấp.

Chủ yếu có 2 cách để cắt tỉa:

- Cắt tỉa trước (Pre – pruning) – ngừng phát triển cây sớm hơn, có nghĩa là ta có thể tỉa / loại bỏ / cắt một nút nếu nó có tầm quan trọng thấp trong khi phát triển cây.
- Cắt tỉa sau (Post – pruning) – khi cây đã được xây dựng đến độ sâu của nó, chúng ta có thể bắt đầu tỉa các nút dựa trên ý nghĩa của chúng.

6.6. Các bước xây dựng Cây quyết định

Bước 1: Bắt đầu cây với nút gốc, gọi là S, chứa toàn bộ tập dữ liệu.

Bước 2: Tìm thuộc tính tốt nhất trong tập dữ liệu bằng ASM (attribute selection measure). Có hai kỹ thuật phổ biến trong ASM, đó là: độ lợi thông tin (information gain) và chỉ số gini (gini index).

Bước 3: Chia S thành các tập con chứa các giá trị có thể có cho các thuộc tính tốt nhất.

Bước 4: Tạo ra các nút của cây quyết định chứa các thuộc tính tốt nhất.

Bước 5: Đệ quy tạo cây quyết định mới bằng cách sử dụng các tập hợp con của tập dữ liệu được tạo ở bước 3. Tiếp tục quá trình này cho đến khi đạt đến một giai đoạn mà không thể phân loại thêm các node và gọi node cuối cùng là nút lá.

6.7. Ưu, nhược điểm của thuật toán Cây quyết định

Ưu điểm

- Mô hình sinh ra các quy tắc dễ hiểu cho người đọc, tạo ra bộ luật với mỗi nhánh lá là một luật của cây.
- Dữ liệu đầu vào có thể là là dữ liệu missing, không cần chuẩn hóa hoặc tạo biến giả.
- Có thể làm việc với cả dữ liệu số và dữ liệu phân loại.
- Có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê.
- Có khả năng là việc với dữ liệu lớn.

Nhược điểm

- Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu của bạn. Thậm chí, với một sự thay đổi nhỏ trong bộ dữ liệu, cấu trúc mô hình cây quyết định có thể thay đổi hoàn toàn.
- Cây quyết định hay gặp vấn đề overfitting.

7. TỔNG QUAN VỀ THUẬT TOÁN RANDOM FOREST

7.1. Khái niệm

Random Forest là một Thuật toán học máy có giám sát, được sử dụng rộng rãi trong các bài toán Phân loại (Classification) và Hồi quy (Regression). Nó xây dựng cây quyết định trên các mẫu khác nhau và lấy đa số phiếu bầu để phân loại, trong trường hợp hồi quy là sẽ tính trung bình các phiếu bầu.

Một trong những tính năng quan trọng nhất của thuật toán Random Forest là nó có thể xử lý tập dữ liệu chứa các biến liên tục (continuous variables) trong trường hợp hồi quy và các biến phân loại (categorical variables) như trong trường hợp phân loại.

Ý tưởng phía sau Random Forest khá đơn giản. Thuật toán này sinh một số cây quyết định (thường là vài trăm) và sử dụng chúng.

7.2. Bài toán thực tế

Giả sử một người đàn ông tên X muốn tìm một địa điểm du lịch nhưng chưa biết nên đi đâu. Lúc này anh ta quyết định hỏi ý kiến của bạn bè về trải nghiệm du lịch trong quá khứ của họ đến những nơi khác nhau. Sau đó, anh ta sẽ nhận được một số câu trả lời của họ về địa điểm du lịch họ thích và cuối cùng là anh ta quyết định chọn ra một địa điểm du lịch được gợi ý bởi nhiều người nhất.

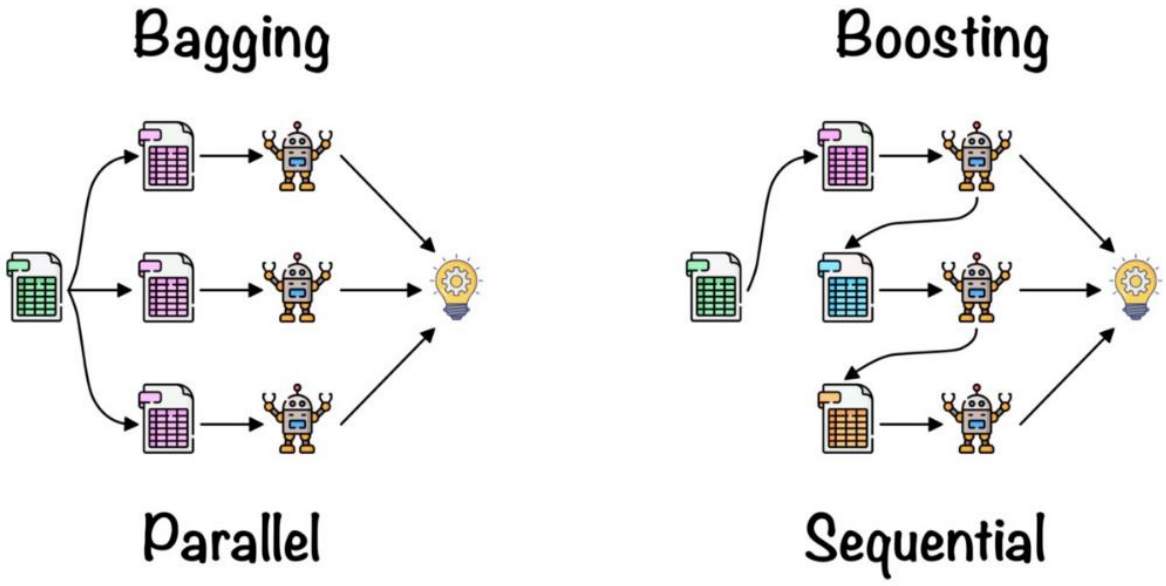
Hình 11. Bài toán thực tế - Random Forest

Trong quá trình quyết định ở trên, có hai phần. Trước tiên, hãy hỏi bạn bè về trải nghiệm du lịch cá nhân của họ và nhận được đề xuất từ nhiều nơi họ đã ghé thăm. Điều này cũng giống như sử dụng thuật toán cây quyết định. Ở đây, mỗi người trong số họ đã chọn những nơi du lịch tốt nhất. Phần thứ hai, sau khi thu thập tất cả các khuyến nghị, là thủ tục bỏ phiếu để chọn địa điểm tốt nhất trong danh sách các khuyến nghị. Toàn bộ quá trình nhận được khuyến nghị từ bạn bè và bỏ phiếu cho họ để tìm ra nơi tốt nhất được gọi là thuật toán Random Forest.

Kỹ thuật kết hợp (Ensemble technique): Ensemble đơn giản là kết hợp nhiều mô hình. Do đó, thay vì sử dụng một mô hình thì lúc này, một tập hợp các mô hình sẽ được sử dụng để đưa ra các dự đoán. Ensemble sử dụng 2 loại phương pháp:

- **Bagging:** Xây dựng một lượng lớn các mô hình (thường là cùng loại) trên những subsamples khác nhau từ tập training dataset (random sample trong 1 dataset để tạo 1 dataset mới). Những mô hình này sẽ được train độc lập và song song với nhau nhưng đầu ra của chúng sẽ được trung bình cộng để cho ra kết quả cuối cùng.

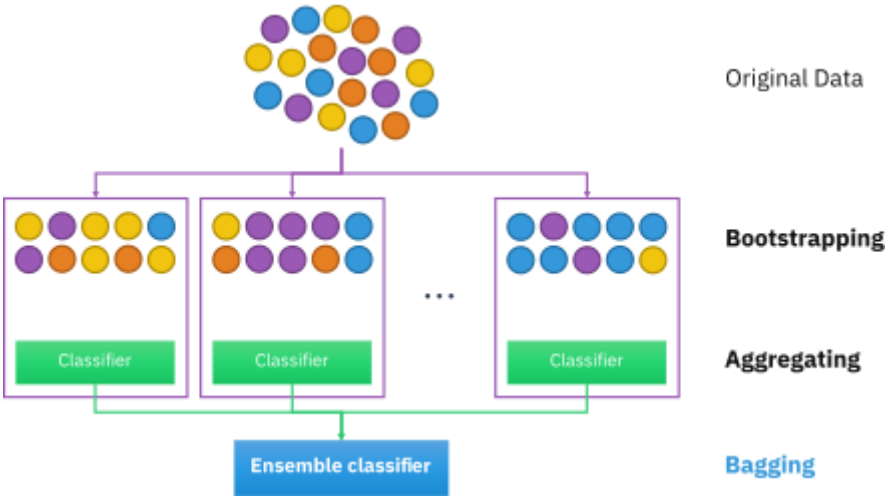
- **Boosting:** Xây dựng một lượng lớn các mô hình (thường là cùng loại). Mỗi model sau sẽ học cách sửa những lỗi của mô hình trước (dữ liệu mà mô hình trước dự đoán sai) để tạo thành một chuỗi các mô hình mà mô hình sau sẽ tốt hơn model trước bởi trọng số được cập nhật qua mỗi mô hình (cụ thể ở đây là trọng số của những dữ liệu dự đoán đúng sẽ không đổi, còn trọng số của những dữ liệu dự đoán sai sẽ được tăng thêm). Chúng ta sẽ lấy kết quả của mô hình cuối cùng trong chuỗi mô hình này làm kết quả trả về (vì mô hình sau sẽ tốt hơn mô hình trước nên tương tự kết quả sau cũng sẽ tốt hơn kết quả trước).



Hình 12. Kỹ thuật kết hợp (Bagging và Boostingg)

Và Random Forest sẽ hoạt dựa dựa trên nguyên tắc của Bagging.

Kỹ thuật Bootstrapping, hay còn gọi là random sampling with replacement, là phương pháp lấy mẫu có hoàn lại. Có nghĩa là khi ta lấy được 1 dữ liệu thì ta không bỏ dữ liệu đấy ra mà vẫn giữ lại trong tập dữ liệu ban đầu, rồi tiếp tục lấy cho tới khi lấy đủ n dữ liệu. Khi dùng kĩ thuật này thì tập n dữ liệu mới của mình có thể có những dữ liệu bị trùng nhau.



Hình 13. Kỹ thuật Bootstrapping

7.3. Các bước xây dựng thuật toán Random Forest

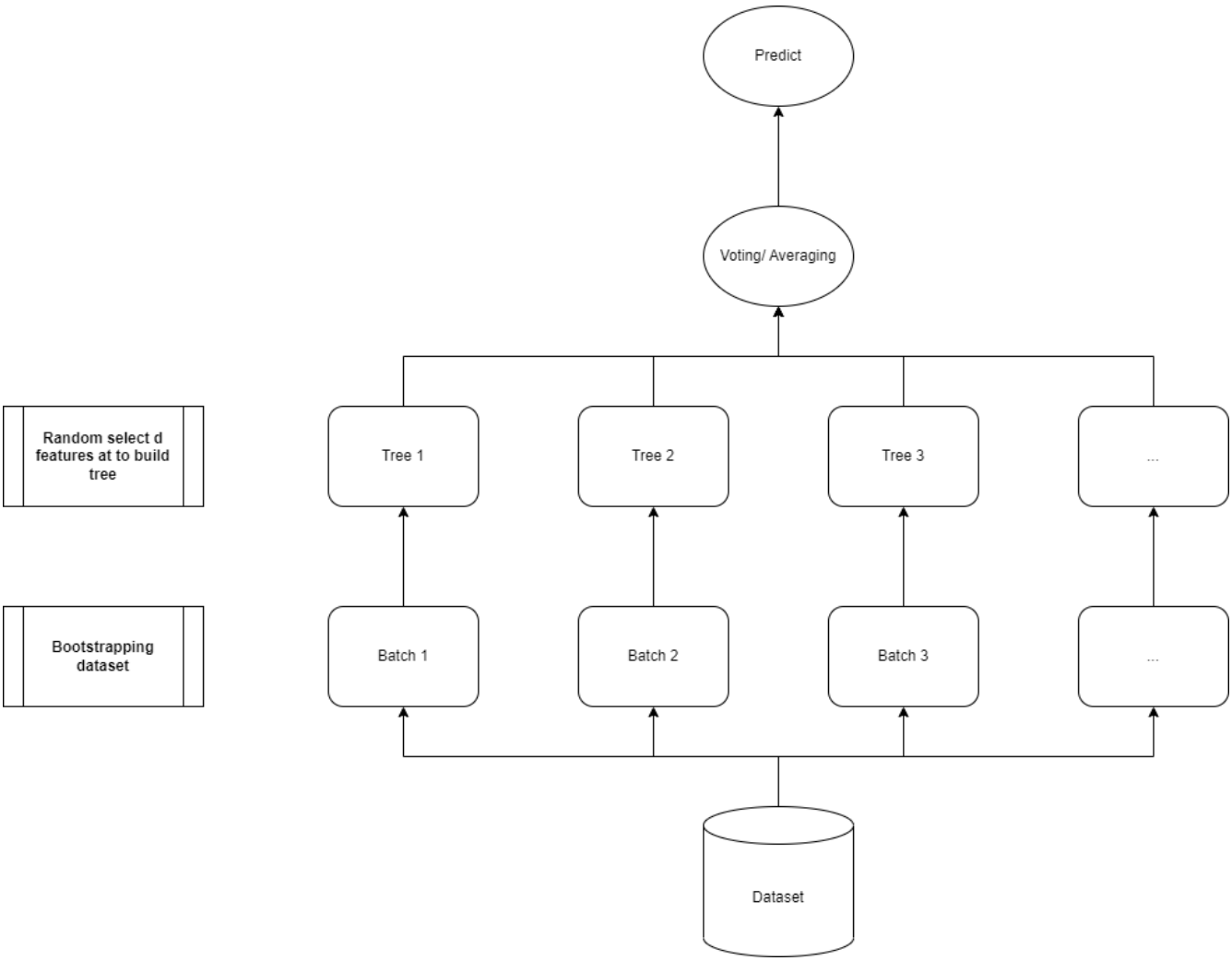
Mô hình rừng cây sẽ áp dụng cả hai phương pháp học kết hợp (ensemble learning) và lấy mẫu tái lập (bootstrapping). Thứ tự của quá trình tạo thành một mô hình rừng cây như sau:

Bước 1: Lấy ngẫu nhiên n dữ liệu từ bộ dữ liệu với kỹ thuật Bootstrapping để tạo thành một tập dữ liệu con. Chính xác hơn, Random Forest sẽ xoá một số quan sát và lặp lại một số khác một cách ngẫu nhiên. Xét toàn cục, những quan sát này vẫn rất gần với tập các quan sát ban đầu, nhưng những thay đổi nhỏ sẽ đảm bảo rằng mỗi cây quyết định sẽ có một chút khác biệt.

Bước 2: Lựa chọn ra ngẫu nhiên thuộc tính (feature) và xây dựng mô Decision Tree dựa trên những thuộc tính này và tập dữ liệu con ở bước 1. Chúng ta sẽ xây dựng nhiều cây quyết định nên bước 1 và 2 sẽ lặp lại nhiều lần.

Bước 3: Mỗi cây quyết định sẽ cho ra một kết quả.

Bước 4: Thực hiện bầu cử hoặc lấy trung bình giữa các cây quyết định để đưa ra kết quả dự báo cuối cùng.



Hình 14. Quy trình xây dựng thuật toán Random Forest

7.4. Ưu, nhược điểm của thuật toán Random Forest

Ưu điểm

- Random Forest có khả năng thực hiện cả hai nhiệm vụ Phân loại (Classification) và Hồi quy (Regression).
- Random Forest có khả năng tìm ra thuộc tính nào quan trọng hơn so với những thuộc tính khác. Trên thực tế, nó còn có thể chỉ ra rằng một số thuộc tính là hoàn toàn vô dụng.
- Nếu như mô hình cây quyết định thường bị nhạy cảm với dữ liệu ngoại lai (outlier) thì mô hình Random Forest được huấn luyện trên nhiều tập dữ liệu con khác nhau, trong đó có

những tập được loại bỏ dữ liệu ngoại lai, điều này giúp cho mô hình ít bị nhạy cảm với dữ liệu ngoại lai hơn.

- Cuối cùng các bộ dữ liệu được sử dụng từ những cây quyết định đều xuất phát từ dữ liệu huấn luyện nên quy luật học được giữa các cây quyết định sẽ gần tương tự như nhau và tổng hợp kết quả giữa chúng không có xu hướng bị chệch.
- Sự kết hợp giữa các cây quyết định giúp cho kết quả ít bị chệch và phương sai giảm. Như vậy chúng ta giảm thiểu được hiện tượng quá khớp (overfitting) ở mô hình Random Forest, một điều mà mô hình cây quyết định thường xuyên gặp phải.

Nhược điểm:

- Dù có độ chính xác khá cao nhưng cây quyết định tồn tại những hạn chế lớn đó là:
- Hạn chế chính của Random Forest là một số lượng lớn các cây có thể làm cho thuật toán trở nên chậm và không hiệu quả đối với các dự đoán thời gian thực (real-time). Nói chung, thuật toán này huấn luyện nhanh, nhưng khá chậm để tạo dự đoán sau khi đã được huấn luyện. Những kết luận dự báo từ chúng thường chỉ đúng trên tập huấn luyện mà không đúng trên tập kiểm tra.
- Nếu muốn dự đoán chính xác hơn đòi hỏi cần nhiều cây hơn, điều này dẫn đến làm mô hình chậm hơn.
- Trong tình huống bộ dữ liệu có số lượng biến lớn. Một cây quyết định có độ sâu giới hạn (để giảm thiểu quá khớp) thường bỏ sót những biến quan trọng.
- Cây quyết định chỉ tạo ra một kịch bản dự báo duy nhất cho mỗi một quan sát nên nếu model có hiệu suất kém thì kết quả sẽ bị chệch.

7.5. Ứng dụng của thuật toán Random Forest

Thuật toán Random Forest được sử dụng trong rất nhiều lĩnh vực, như tài chính, thị trường chứng khoán, y học và trong thương mại điện tử.

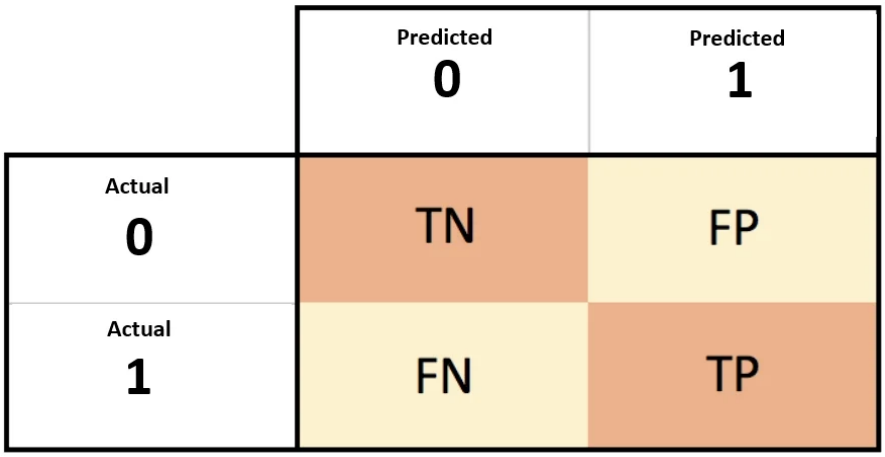
- Trong tài chính, được sử dụng để xác định khả năng khách hàng đó trả nợ có đúng hạn hay không. Bên cạnh đó, thuật toán còn được sử dụng để xác định lừa đảo trong ngân hàng.

- Trong thị trường chứng khoán, thuật toán được sử dụng để xác định giá cổ phiếu.
- Trong lĩnh vực y học, được dùng để xác định thành phần của thuốc và phân tích lịch sử bệnh của bệnh nhân để xác định bệnh của họ.
- Trong thương mại điện tử, Random Forest được dùng để quyết định xem liệu khách hàng có yêu thích sản phẩm hay không.

8. TỔNG QUAN VỀ CÁC TIÊU CHÍ ĐÁNH GIÁ MÔ HÌNH

8.1. Ma trận nhầm lẫn (Confusion Matrix)

Phương pháp Ma trận nhầm lẫn (Confusion matrix) thể hiện có bao nhiêu điểm dữ liệu thực sự thuộc vào một lớp, lớp nào được phân loại đúng nhiều nhất, dữ liệu của lớp nào bị phân loại nhầm vào lớp khác.



Hình 15. Ma trận nhầm lẫn (Nguồn: subscription.packtpub.com)

Đối với bài toán trong phần này:

- TP (True Positive): khi mô hình dự đoán đúng là mua.
- TN (True Negative): khi mô hình dự đoán đúng là không mua.
- FP (False Positive): khi mô hình dự đoán sai không mua thành mua.
- FN (False Negative): khi mô hình dự đoán sai mua thành không mua.

Accuracy (Độ chính xác) tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm tra. Phương pháp này chỉ cho chúng ta biết được bao nhiêu phần trăm

lượng dữ liệu được phân loại đúng mà không chỉ ra được cụ thể mỗi loại được phân loại như thế nào, lớp nào được phân loại đúng nhất, lớp nào bị phân loại nhầm.

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

Recall thể hiện khả năng mô hình dự đoán không bị sót nhãn, nó là tổng số ví dụ được phân loại chính xác của một lớp chia cho tổng số các ví dụ của lớp đó. Recall càng cao càng tốt, tức là FN trong mô hình này phân loại nhầm mua thành không mua càng nhỏ càng tốt.

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

Precision thể hiện khả năng mô hình dự đoán đúng nhãn, nó là tổng số ví dụ được phân loại chính xác của một lớp chia cho tổng số ví dụ được phân loại vào lớp đó. Precision càng cao càng tốt, tức là FP trong mô hình này phân loại nhầm không mua thành mua càng nhỏ càng tốt.

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

F1 – score là sự kết hợp giữa hai tiêu chí Precision và Recall, nó là trung bình điều hòa của Precision và Recall. Như vậy F1-score được dùng khi ta quan tâm đồng đều vai trò của Precision và Recall, nói cách khác muốn mô hình vừa nhạy, vừa chính xác.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

8.2. Diện tích dưới đường cong AUC

AUC (Area Under the Curve) là một phép đo tổng hợp về hiệu suất của phân loại nhị phân trên tất cả các giá trị ngưỡng có thể có. Để hiểu rõ hơn về metric này, chúng ta sẽ tìm hiểu về một khái niệm cơ sở trước, đó là ROC Curve

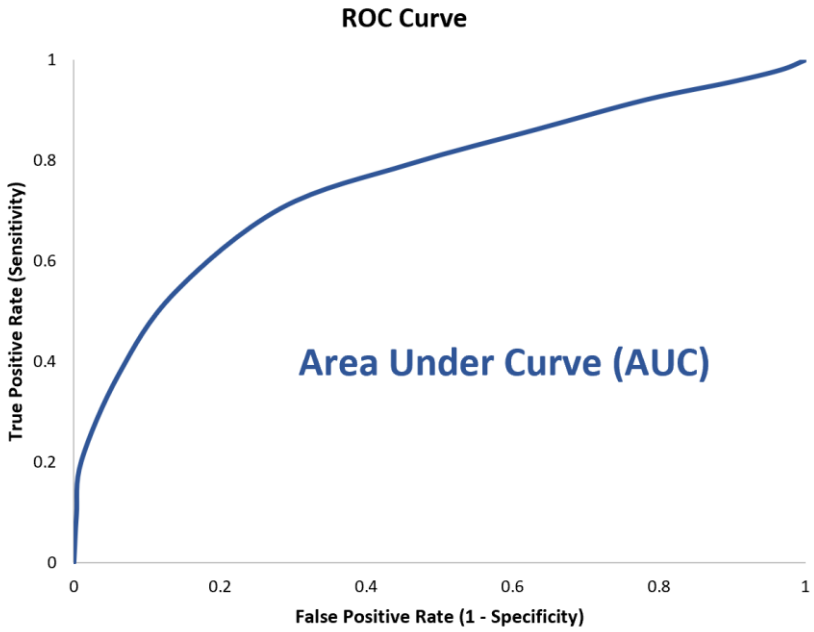
ROC Curve (The receiver operating characteristic curve) là một đường cong biểu diễn hiệu suất phân loại của một mô hình phân loại tại các ngưỡng threshold. Về cơ bản, nó hiển thị

True Positive Rate (TPR) so với False Positive Rate (FPR) đối với các giá trị ngưỡng khác nhau. Các giá trị TPR, FPR được tính như sau:

$$TPR = \frac{TP}{TP + FP}$$

$$FPR = \frac{FP}{TN + FN}$$

AUC là chỉ số được tính toán dựa trên đường cong ROC nhằm đánh giá khả năng phân loại của mô hình tốt như thế nào. Phần diện tích nằm dưới đường cong ROC và trên trục hoành chính là AUC, có giá trị nằm trong khoảng [0, 1].



Hình 16. Mô tả AUC (Nguồn: statology.org)

Khi diện tích này càng lớn, đường cong này sẽ dần tiệm cận với đường thẳng $y = 1$ tương đương với khả năng phân loại của mô hình càng tốt. Còn khi đường cong ROC nằm sát với đường chéo đi qua hai điểm (0, 0) và (1, 1), mô hình sẽ tương đương với một phân loại ngẫu nhiên.

CHƯƠNG 2: TÌM HIỂU CHUNG VÀ KHAI PHÁ DỮ LIỆU (EDA)

Trình bày tổng quan về công việc khai phá bộ dữ liệu, mô tả ý nghĩa của dữ liệu được hiển thị. Thực hiện dữ liệu EDA làm sạch dữ liệu. Hơn nữa, trực quan hóa dữ liệu và đưa ra các nhận xét, đồng thời đưa ra hướng giải quyết.

1. THÔNG TIN CHUNG

1.1. Mô tả dữ liệu

Bộ dữ liệu được sử dụng trong dự đoán ý định mua sắm trực tuyến của khách hàng được lấy từ Kho lưu trữ Máy học UCI, một trang web phổ biến với hàng trăm bộ dữ liệu có sẵn để phân tích. Tác giả của tập dữ liệu này là là C. Sakar và Yomi Kastro.

Mỗi hàng trong tập dữ liệu chứa một vectơ đặc trưng chứa dữ liệu tương ứng với một phiên (session) trên trang web thương mại điện tử. Tập dữ liệu được tạo cụ thể để mỗi phiên sẽ thuộc về một người dùng duy nhất trong khoảng thời gian 1 năm.

Tập dữ liệu này có 12330 dòng, được chia thành: 10422 dòng tương ứng người không mua hàng và 1908 dòng tương ứng người đã mua hàng.

Nhãn lớp trong tập dữ liệu được gọi là Revenue và chứa giá trị True hoặc False, tương ứng với việc người dùng có thực hiện mua hàng trên trang web trong lần truy cập của họ hay không. Có thể thấy tập dữ liệu không cân bằng, vì 85% phiên chứa nhãn False, 15% còn lại chứa nhãn True.

1.2. Mô tả thuộc tính

Tập dữ liệu có 18 cột và 12330 dòng bao gồm 10 thuộc tính số và 8 thuộc tính phân loại.

Thuộc tính “Revenue” được sử dụng làm nhãn lớp.

Thuộc tính	Mô tả	Ghi chú

Administrative	Số trang thuộc loại Administrative mà người dùng đã truy cập.	Giá trị của các features này được lấy từ thông tin URL của các trang mà người dùng đã truy cập và được cập nhật theo thời gian thực khi người dùng thực hiện một hành động (ví dụ: chuyển từ trang này sang trang khác).
Administrative_Duration	Tổng thời gian dành cho trang Administrative	
Informational	Số trang thuộc loại Informational mà người dùng đã truy cập.	
Informational_Duration	Tổng thời gian dành cho trang Informational	
ProductRelated	Số trang thuộc loại Product Related mà người dùng đã truy cập.	
ProductRelated_Duration	Tổng thời gian dành cho trang Product Related.	
BounceRates	Phần trăm khách truy cập vào trang web thông qua trang đó và thoát ra mà không kích hoạt bất kỳ tác vụ bổ sung nào.	Dữ liệu thu thập được từ Google Analytics
ExitRates	Phần trăm số lần xem trang trên trang web kết thúc tại trang cụ thể đó.	
PageValues	Giá trị trung bình cho một trang web mà người dùng đã truy cập trước khi hoàn tất giao dịch thương mại điện tử.	
SpecialDay	Giá trị này thể hiện mức độ gần của ngày duyệt với những ngày hoặc ngày lễ đặc biệt (ví dụ như	Giá trị của feature này được xác định bằng cách xem xét các động lực của thương mại

	Ngày của mẹ hoặc ngày lễ tình nhân) mà giao dịch có nhiều khả năng được hoàn tất hơn.	điện tử như khoảng thời gian giữa ngày đặt hàng và ngày giao hàng. Ví dụ: đối với ngày lễ Valentine, giá trị này nhận giá trị khác 0 trong khoảng thời gian từ ngày 2 tháng 2 đến ngày 12 tháng 2, nhận giá trị bằng 0 vào trước và sau ngày valentine và giá trị lớn nhất của nó là 1 vào ngày 8 tháng 2.
Month	Chứa tháng xảy ra lượt xem trang, ở dạng chuỗi.	
OperatingSystems	Một giá trị số nguyên đại diện cho hệ điều hành mà người dùng đã sử dụng khi xem trang.	
Browser	Một giá trị số nguyên đại diện cho trình duyệt mà người dùng đang sử dụng để xem trang.	
Region	Một giá trị số nguyên đại diện cho khu vực mà người dùng đang ở.	
Traffic Type	Một giá trị số nguyên đại diện cho loại lưu lượng truy cập mà người dùng được phân loại thành.	
VisitorType	Một chuỗi biểu thị liệu khách truy cập là Khách truy cập mới,	

	Khách truy cập quay lại hay Khác.	
Weekend	Một boolean đại diện cho việc phiên có vào cuối tuần hay không.	
Revenue	Một boolean đại diện cho việc người dùng đã hoàn tất giao dịch mua hay chưa.	

Bảng 1. Mô tả thuộc tính

2. THAO TÁC VÀ LÀM SẠCH DỮ LIỆU (MANIPULATION AND CLEANING DATA)

2.1. Kiểm tra thông tin và kiểm tra giá trị null của bộ dữ liệu

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
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 17. Thông tin của bộ dữ liệu

```
# Check for null values in data
nullcount = df.isnull().sum()
print('Total number of null values in dataset:',
nullcount.sum())
```

```
Total number of null values in dataset: 0
```

Hình 18. Tổng số các giá trị null của bộ dữ liệu

Ở đây chúng ta có thể thấy rằng không có giá trị null nào trong tập dữ liệu. Nhóm không cần sửa hoặc thay thế bất kỳ giá trị null nào trong tập dữ liệu.

2.2. Kiểm tra giá trị missing

```
# missing percentage of the data
missing_percentage = df.isnull().sum()/df.shape[0]
print(missing_percentage)
```

```
Administrative             0.0
Administrative_Duration    0.0
Informational              0.0
Informational_Duration     0.0
ProductRelated             0.0
ProductRelated_Duration    0.0
BounceRates                0.0
ExitRates                  0.0
PageValues                 0.0
SpecialDay                 0.0
Month                      0.0
OperatingSystems           0.0
Browser                    0.0
Region                     0.0
TrafficType                0.0
VisitorType                0.0
Weekend                    0.0
Revenue                    0.0
dtype: float64
```

Hình 19. Số các giá trị missing của bộ dữ liệu

Tương tự, phần trăm giá trị missing của các feature đều bằng 0

2.3. Kiểm tra giá trị duplicate và các giá trị unique

```
#duplicate values and drop it
df.duplicated().sum()
df=df.drop_duplicates()
```

```
#duplicate values and drop it
df.duplicated().sum()
```

```
125
```

Hình 20. Tổng số các giá trị duplicate của bộ dữ liệu

Trong dataset có chứa 125 duplicate rows. Vì vậy nhóm sẽ xóa các dòng bị duplicate trước khi đưa mô hình vào huấn luyện

```
# Checking for number of unique values for each feature
uniques = df.nunique(axis=0)
print(uniques)
```

```
Administrative             27
Administrative_Duration    3335
Informational              17
Informational_Duration     1258
ProductRelated             311
ProductRelated_Duration    9551
BounceRates                1872
ExitRates                  4777
PageValues                 2704
SpecialDay                 6
Month                      10
OperatingSystems           8
Browser                    13
Region                     9
TrafficType                20
VisitorType                3
Weekend                    2
Revenue                    2
dtype: int64
```

Hình 21. Số các giá trị unique của bộ dữ liệu

2.4. Kiểm tra giá trị bất thường và giải quyết các outliers

```
# Describe the dataset
df.describe(include='all')
```

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated
count	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000
unique	NaN	NaN	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN	NaN
mean	2.315166	80.818611	0.503569	34.472398	31.731468
std	3.321784	176.779107	1.270156	140.749294	44.475503
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	7.000000
50%	1.000000	7.500000	0.000000	0.000000	18.000000
75%	4.000000	93.256250	0.000000	0.000000	38.000000
max	27.000000	3398.750000	24.000000	2549.375000	705.000000

Hình 22. Kiểm tra các giá trị bất thường

ProductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay	Month	OperatingSystems
12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330	12330.000000
NaN	NaN	NaN	NaN	NaN	10	NaN
NaN	NaN	NaN	NaN	NaN	May	NaN
NaN	NaN	NaN	NaN	NaN	3364	NaN
1194.746220	0.022191	0.043073	5.889258	0.061427	NaN	2.124006
1913.669288	0.048488	0.048597	18.568437	0.198917	NaN	0.911325
0.000000	0.000000	0.000000	0.000000	0.000000	NaN	1.000000
184.137500	0.000000	0.014286	0.000000	0.000000	NaN	2.000000
598.936905	0.003112	0.025156	0.000000	0.000000	NaN	2.000000
1464.157214	0.016813	0.050000	0.000000	0.000000	NaN	3.000000
63973.522230	0.200000	0.200000	361.763742	1.000000	NaN	8.000000

Hình 23. Kiểm tra các giá trị bất thường

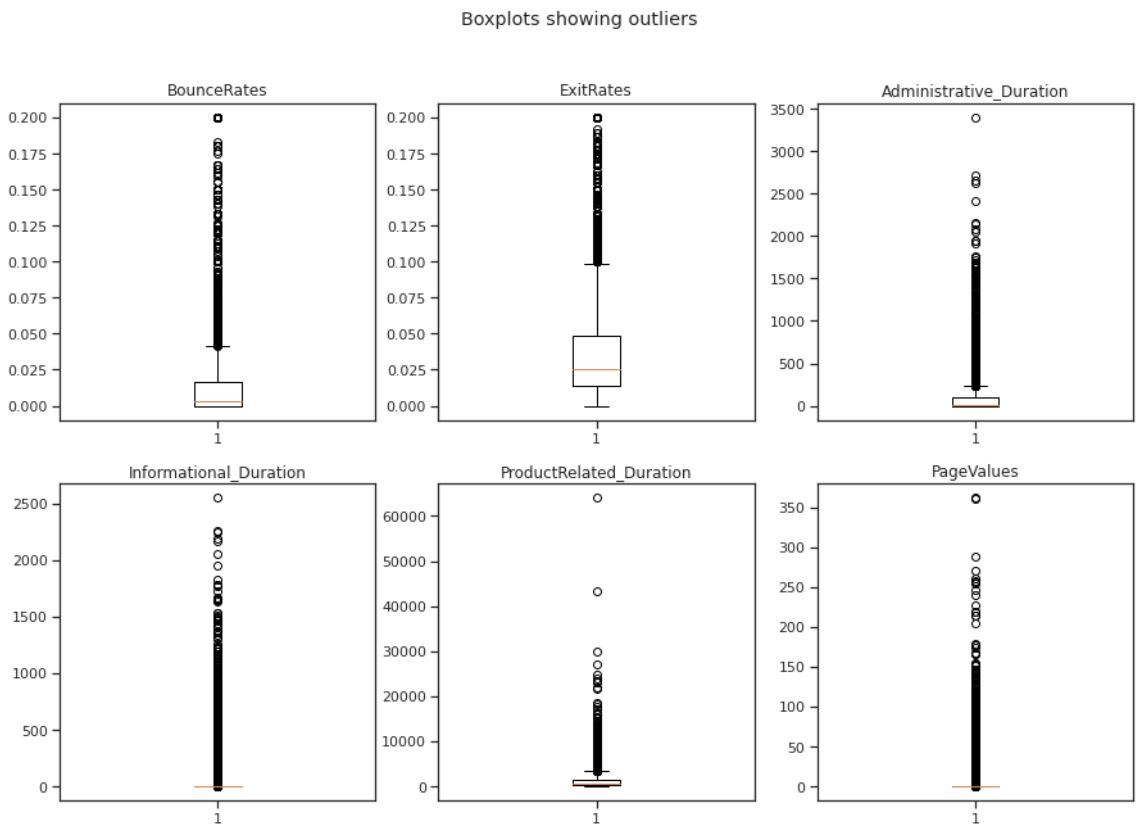
Browser	Region	TrafficType	VisitorType	Weekend
12330.000000	12330.000000	12330.000000	12330	12330
NaN	NaN	NaN	3	2
NaN	NaN	NaN	Returning_Visitor	False
NaN	NaN	NaN	10551	9462
2.357097	3.147364	4.069586	NaN	NaN
1.717277	2.401591	4.025169	NaN	NaN
1.000000	1.000000	1.000000	NaN	NaN
2.000000	1.000000	2.000000	NaN	NaN
2.000000	3.000000	2.000000	NaN	NaN
2.000000	4.000000	4.000000	NaN	NaN
13.000000	9.000000	20.000000	NaN	NaN

Hình 24. Kiểm tra các giá trị bất thường

Không có giá trị bất thường.

```
#check boxplots showing outlier
plt.figure(figsize = (15, 10))

ax=plt.subplot(231)
plt.boxplot(df['BounceRates'])
ax.set_title('BounceRates')

ax=plt.subplot(232)
plt.boxplot(df['ExitRates'])
ax.set_title('ExitRates')

ax=plt.subplot(233)
plt.boxplot(df['Administrative_Duration'])
ax.set_title('Administrative_Duration')
```

```
ax=plt.subplot(234)
plt.boxplot(df['Informational_Duration'])
ax.set_title('Informational_Duration')

ax=plt.subplot(235)
plt.boxplot(df['ProductRelated_Duration'])
ax.set_title('ProductRelated_Duration')

ax=plt.subplot(236)
plt.boxplot(df['PageValues'])
ax.set_title('PageValues')

plt.suptitle('Boxplots showing outliers')
```



Hình 25. Kiểm tra các giá trị outliers

Xóa các giá trị outlier: Sử dụng Z- Score, còn được gọi là điểm tiêu chuẩn. Giá trị này giúp hiểu rằng điểm dữ liệu cách giá trị trung bình bao xa. Và sau khi thiết lập giá trị ngưỡng, người ta có thể sử dụng giá trị điểm số z của các điểm dữ liệu để xác định các giá trị ngoại lệ.

Chọn giá trị ngưỡng ngoại lệ là 3. Vì 99,7% điểm dữ liệu nằm trong khoảng +/- 3 độ lệch chuẩn (sử dụng phương pháp Phân phối Gaussian).

```
num_col =
['Administrative_Duration','Informational_Duration','ProductRe
lated_Duration','BounceRates','ExitRates','PageValues']
data = df[num_col]
z=np.abs(stats.zscore(data))
filtered_entries = (z < 3).all(axis=1)
df = df[filtered_entries]
```

```
df
```

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	Spe
1	0	0.0	0	0.0	2	64.000000	0.000000	0.100000	0.000000	
3	0	0.0	0	0.0	2	2.666667	0.050000	0.140000	0.000000	
4	0	0.0	0	0.0	10	627.500000	0.020000	0.050000	0.000000	
5	0	0.0	0	0.0	19	154.216667	0.015789	0.024561	0.000000	
8	0	0.0	0	0.0	2	37.000000	0.000000	0.100000	0.000000	
...	...	...	...	...	...	...	...	...	...	
12325	3	145.0	0	0.0	53	1783.791667	0.007143	0.029031	12.241717	
12326	0	0.0	0	0.0	5	465.750000	0.000000	0.021333	0.000000	
12327	0	0.0	0	0.0	6	184.250000	0.083333	0.086667	0.000000	
12328	4	75.0	0	0.0	15	346.000000	0.000000	0.021053	0.000000	
12329	0	0.0	0	0.0	3	21.250000	0.000000	0.066667	0.000000	

10747 rows × 18 columns

Hình 26. Bộ dữ liệu sau khi đã xóa các giá trị outliers

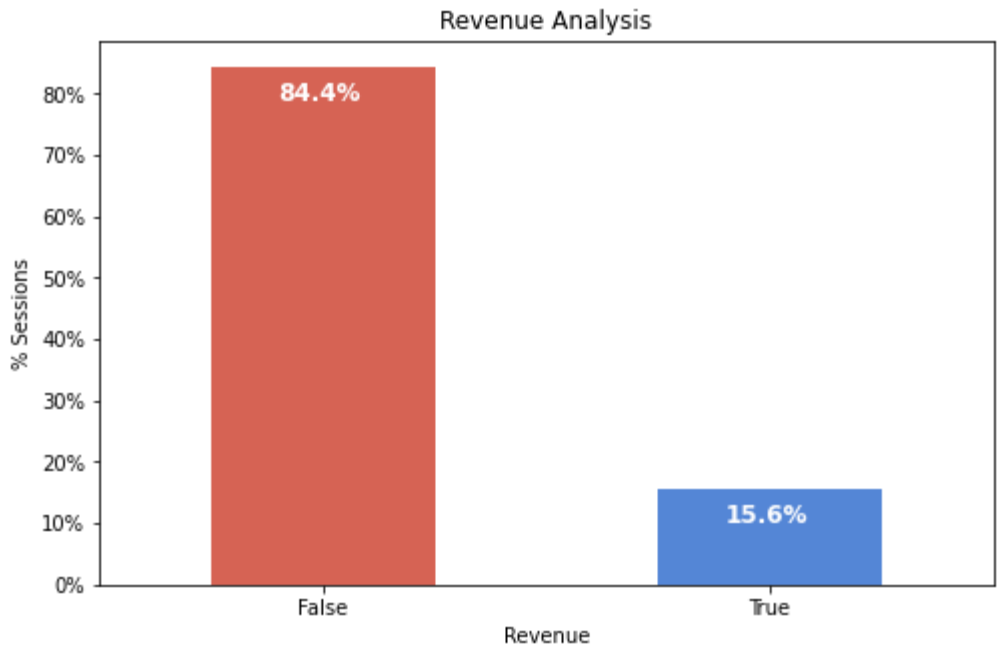
Dữ liệu còn 10747 dòng sau khi đã xóa các giá trị outlier.

3. TRỰC QUAN HÓA DỮ LIỆU, MỘT SỐ INSIGHTS VÀ HƯỚNG GIẢI QUYẾT

3.1. Phân tích thuộc tính nhãn lớp “Revenue”

```
import matplotlib.ticker as mtick
plt.figure(figsize = (8,5))
colors = ['#d66354','#5486d6']

ax1 = (df['Revenue'].value_counts()*100.0
/len(df)).plot(kind='bar', stacked = True,
                    rot = 0,color=colors)

ax1.yaxis.set_major_formatter(mtick.PercentFormatter())
ax1.set_ylabel('% Sessions')
ax1.set_xlabel('Revenue')
ax1.set_title('Revenue Analysis')
totals = []
```

```
for i in ax1.patches:
    totals.append(i.get_width())

total = sum(totals)

for i in ax1.patches:
    ax1.text(i.get_x()+.15, i.get_height()-5.5, \
             str(round((i.get_height()/total), 1))+'%',
             fontsize=12,
             color='white',
             weight = 'bold')
```



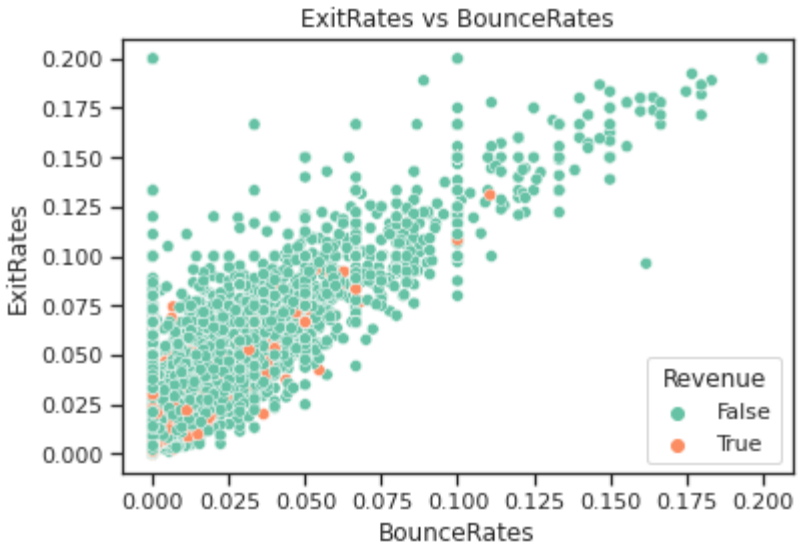
Hình 27. Phân tích thuộc tính "Revenue"

Biểu đồ cột ở trên cho thấy sự so sánh giữa 2 trạng thái của Revenue. Từ biểu đồ, có thể thấy rằng 84,4% số phiên không có kết quả doanh thu và 15,6% số phiên có doanh thu. Dù tập dữ liệu không cân bằng nhưng điều này cũng hợp lý vì phần lớn hoạt động mua sắm trực tuyến thông thường kết thúc mà không có sự mua hàng.

3.2. Tác động giữa các thuộc tính "BounceRates" và "ExitRates"

- Tương quan giữa BounceRates và ExitRates

```
ax = sns.scatterplot(x="BounceRates", y="ExitRates",hue =
'Revenue',palette = "Set2", data=df)
ax.set_title('ExitRates vs BounceRates')
```



Hình 28. Biểu đồ tương quan giữa BounceRates và ExitRates

BounceRates và ExitRates có mối tương quan dương. BounceRates và ExitRates cao dẫn đến không có doanh thu.

Revenue data bị mất cân đối nhiều.

- BounceRates và ExitRates sẽ ảnh hưởng đến Revenue như thế nào?

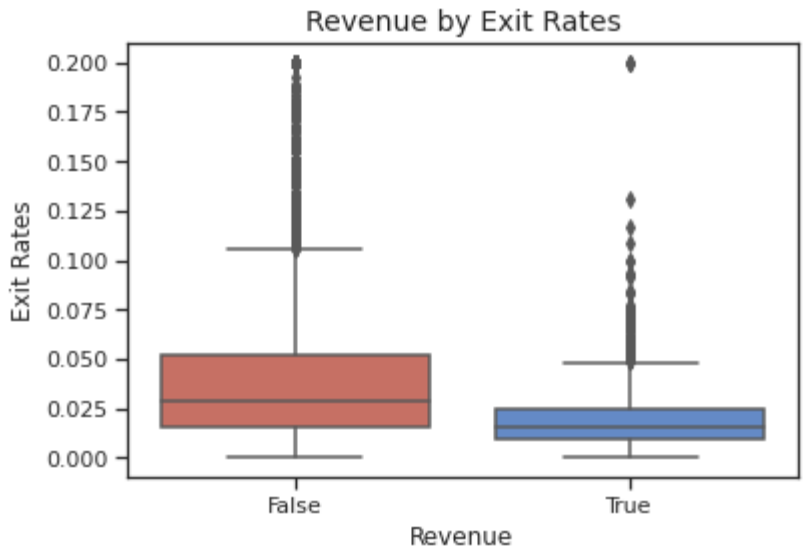
```
ax1 = sns.boxplot(x="Revenue",y="BounceRates",
data=df,palette=['#d66354','#5486d6'])
ax1.set_xlabel("Revenue", fontsize=12)
ax1.set_ylabel("Bounce Rates", fontsize=12)
ax1.set_title("Revenue by Bounce Rates", fontsize=14)
```

Hình 29. Boxplot của Revenue với BounceRates

Biểu đồ trên cho thấy boxplots của Revenue với BounceRates.

Tỷ lệ bỏ trang thấp hơn 0.01 dẫn đến không có doanh thu.

```
ax1 = sns.boxplot(x="Revenue",y="ExitRates",
data=df,palette=['#d66354','#5486d6'])
ax1.set_xlabel("Revenue", fontsize=12)
ax1.set_ylabel("Exit Rates", fontsize=12)
ax1.set_title("Revenue by Exit Rates", fontsize=14)
```



Hình 30. Boxplot của Revenue với ExitRates

Biểu đồ trên cho thấy boxplots của Revenue với ExitRates.

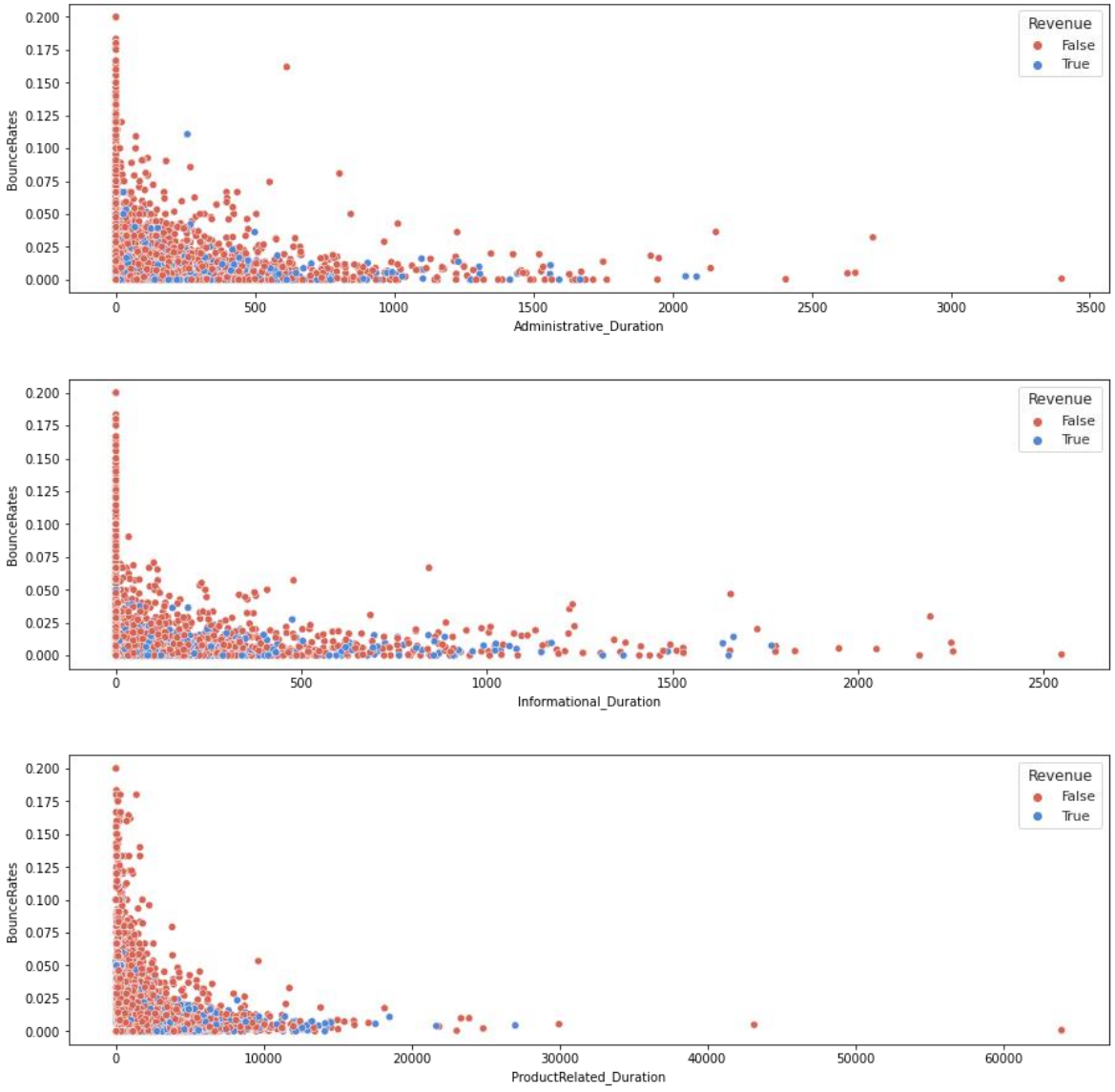
Khi ExitRates thấp hơn khoảng 0.02, thì khả năng cao là khách hàng sẽ mua 1 sản phẩm.

Tỷ lệ thoát cao hơn 0.03 dẫn đến không có doanh thu.

- Thời lượng khi truy cập trang web sẽ ảnh hưởng đến Bounce Rates như thế nào?

```
fig, ax = plt.subplots(3,figsize=(15,15))

sns.set(style="ticks")

ax1 = sns.scatterplot(x="Administrative_Duration",
y="BounceRates",hue="Revenue",palette =
['#d66354','#5486d6'],  data=df, ax=ax[0])
ax2 = sns.scatterplot(x="Informational_Duration",
y="BounceRates",hue="Revenue",palette = ['#d66354','#5486d6'],
data=df, ax=ax[1])
```

```
ax3 = sns.scatterplot(x="ProductRelated_Duration",
y="BounceRates",hue="Revenue",palette =
['#d66354','#5486d6'],  data=df, ax=ax[2])

plt.subplots_adjust(wspace = 0.2, hspace = 0.3, top = 0.9)
```



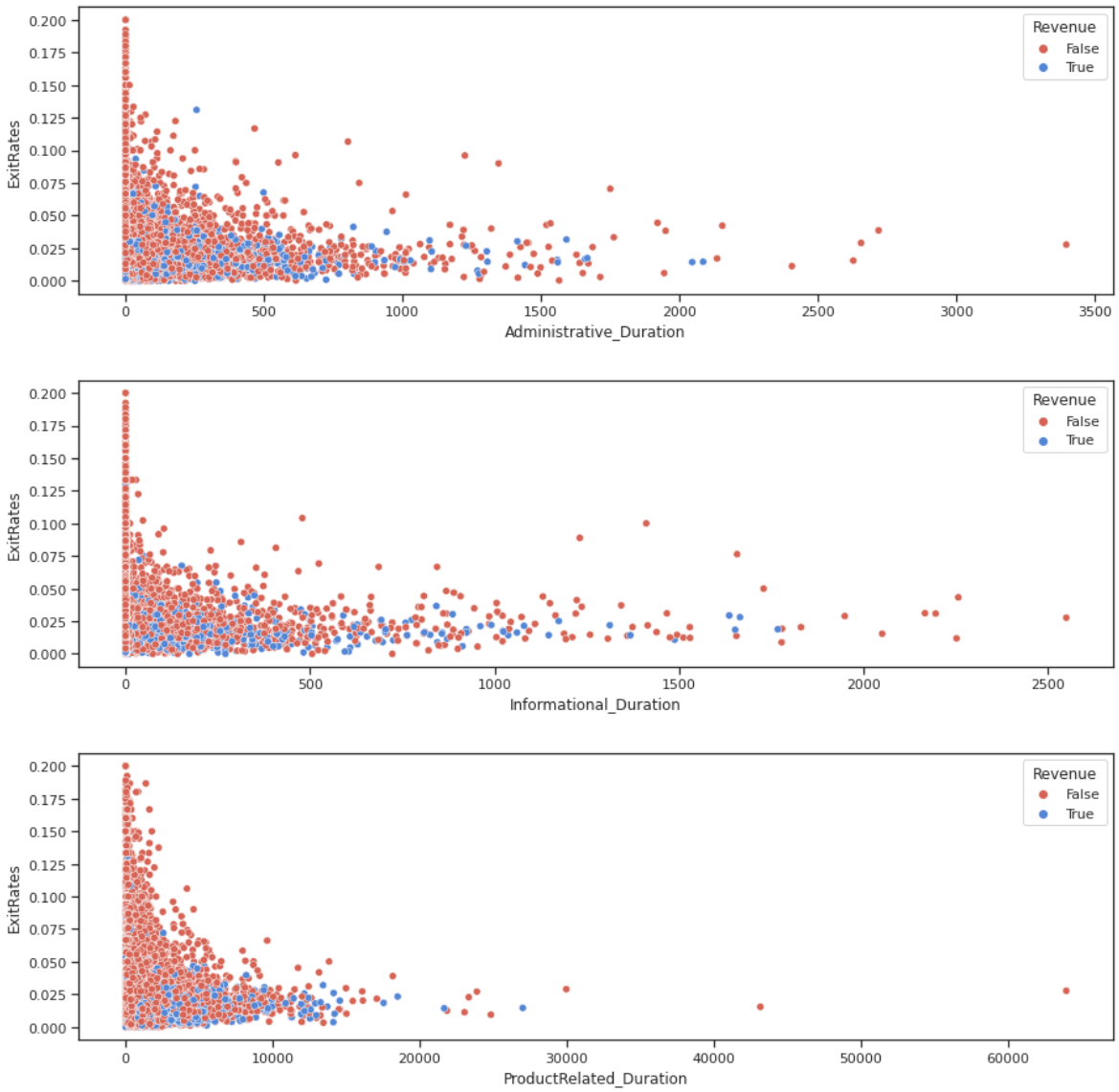
Hình 31. Ảnh hưởng của thời lượng truy cập trang web đến BounceRates

Biểu đồ trên cho thấy ảnh hưởng của thời lượng truy cập trang web đến tỷ lệ thoát (BounceRates). Có thể thấy rằng thời lượng trên trang web lâu hơn thì tỷ lệ thoát cũng giảm.

Đặc biệt là thời lượng khách hàng dành cho trang Product Related dài hơn đáng kể so với loại trang khác.

- Thời lượng khi truy cập trang web sẽ ảnh hưởng đến Exit Rates như thế nào?

```
fig, ax = plt.subplots(3,figsize=(15,15))

sns.set(style="ticks")
ax1 = sns.scatterplot(x="Administrative_Duration",
y="ExitRates",hue="Revenue",palette =
['#d66354','#5486d6'],  data=df, ax=ax[0])
ax2 = sns.scatterplot(x="Informational_Duration",
y="ExitRates",hue="Revenue",palette = ['#d66354','#5486d6'],
data=df, ax=ax[1])
ax3 = sns.scatterplot(x="ProductRelated_Duration",
y="ExitRates",hue="Revenue",palette =
['#d66354','#5486d6'],  data=df, ax=ax[2])
plt.subplots_adjust(wspace = 0.2, hspace = 0.3, top = 0.9)
```



Hình 32. Ảnh hưởng của thời lượng truy cập trang web đến ExitRates

Biểu đồ trên cho thấy ảnh hưởng của thời lượng truy cập trang web đến tỷ lệ thoát (ExitRates). Có thể thấy rằng thời lượng trên trang web lâu hơn thì tỷ lệ thoát cũng giảm.

Đặc biệt là thời lượng khách hàng dành cho trang Product Related dài hơn đáng kể so với loại trang khác.

Có thể thấy dù thời gian dành ra ở Product Related dài hơn nhưng tỷ lệ bỏ trang và tỷ lệ thoát trang lại cao và tỷ lệ chuyển đổi thấp. Nguyên nhân của việc này có thể là do:

- Vấn đề UX/UI khi khách hàng tiếp cận trang Product Related: giao diện không được bắt mắt, thông tin sản phẩm không rõ ràng khiến họ phải dành nhiều thời gian tìm kiếm.
- Vấn đề về phí giao hàng: khách hàng có thể sẽ sẵn sàng để mua sản phẩm với mức giá X. Nhưng khi đi vào trang giỏ hàng, thì phí giao hàng có thể sẽ cao hơn bất ngờ và khiến khách hàng không còn muốn mua sản phẩm đó nữa và họ sẽ thoát trang.
- Giá sản phẩm: đây cũng là một yếu tố quan trọng ảnh hưởng đến tỷ lệ bỏ trang và thoát trang nếu như giá sản phẩm quá cao.

Để cải thiện phần này, nhóm đề xuất một số hướng phát triển như sau:

Hướng phát triển 1: Vì thời lượng khách hàng dành ra trên loại trang Product Related nhiều hơn đáng kể, nên cần điều chỉnh các trang sản phẩm bằng cách thiết kế sao cho nút "thêm vào giỏ hàng" nổi bật hơn, giao diện người dùng (UI) thân thiện, bên cạnh đó nên cung cấp thêm một số miêu tả (description) ngắn và biểu tượng (icons) cho sản phẩm nếu cần thiết, màu sắc phải hiệu quả và đảm bảo rằng quá trình mua hàng của khách hàng càng trơn tru càng tốt.

Một khía cạnh quan trọng nữa là cần phải đảm bảo để phí giao hàng không tạo ra ảnh hưởng đáng kể đến ExitRates.

Hướng phát triển 2: Dựa vào thông tin đã thu thập được của từng khách hàng qua các form đăng ký thông tin, cần phải tối ưu hóa việc gửi email. Cụ thể là phải cá nhân hóa cho từng email gửi đến khách hàng. Cá nhân hóa sẽ mang lại sự trung thành trên quy mô lớn và từ đó giữ chân người dùng tốt hơn.

Ví dụ: Doanh nghiệp dựa vào dữ liệu được báo cáo sẽ biết được khách hàng đang dự định mua sản phẩm gì, nhưng họ vẫn còn do dự, vẫn còn sự so sánh giá cả giữa các nền tảng, thương hiệu khác nhau. Nếu doanh nghiệp nắm bắt được vấn đề này sẽ có hướng giải quyết phù hợp, ví như gửi mã voucher riêng biệt, chỉ những người có mã voucher đó mới được sử dụng. Điều này nhằm tạo cảm giác khách hàng được coi trọng và họ sẽ dành thời gian để nghiên cứu và mua hàng một cách nghiêm túc hơn.

Hướng phát triển 3: Tạo ra các pop-ups để cung cấp mã giảm giá (offer discounts) cho khách hàng hoặc những câu hỏi (query) được xây dựng cá nhân hóa mỗi khi khách hàng thoát trang hoặc là khi họ đang muốn rời website.

Xu hướng sản phẩm: có thể sản phẩm mà công ty này đang kinh doanh hiện đang không hợp xu hướng của người tiêu dùng.

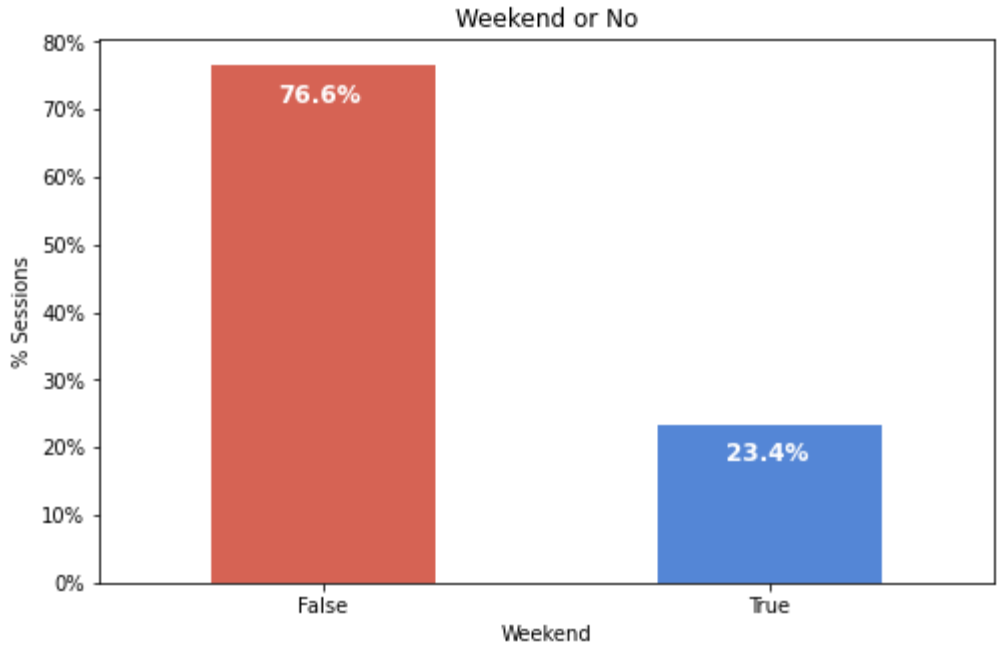
Hướng phát triển 4: Công ty cần nghiên cứu lại thị trường và nâng cấp (upgrade) lên xu hướng mới để các sản phẩm không bị lỗi thời.

3.3. Tác động của ngày cuối tuần đến việc mua sắm của khách hàng

- Phân tích thuộc tính "Weekend"

```
plt.figure(figsize = (8,5))
colors = ['#d66354','#5486d6']

ax1 = (df['Weekend'].value_counts()*100.0
/len(df)).plot(kind='bar', stacked = True,
                    rot = 0,color=colors)

ax1.yaxis.set_major_formatter(mtick.PercentFormatter())
ax1.set_ylabel('% Sessions')
ax1.set_xlabel('Weekend')
ax1.set_title('Weekend or No')
totals = []

for i in ax1.patches:
    totals.append(i.get_width())

total = sum(totals)

for i in ax1.patches:
```

```
    ax1.text(i.get_x()+.15, i.get_height()-5.5, \
            str(round((i.get_height()/total), 1))+'%',
            fontsize=12,
            color='white',
            weight = 'bold')
```



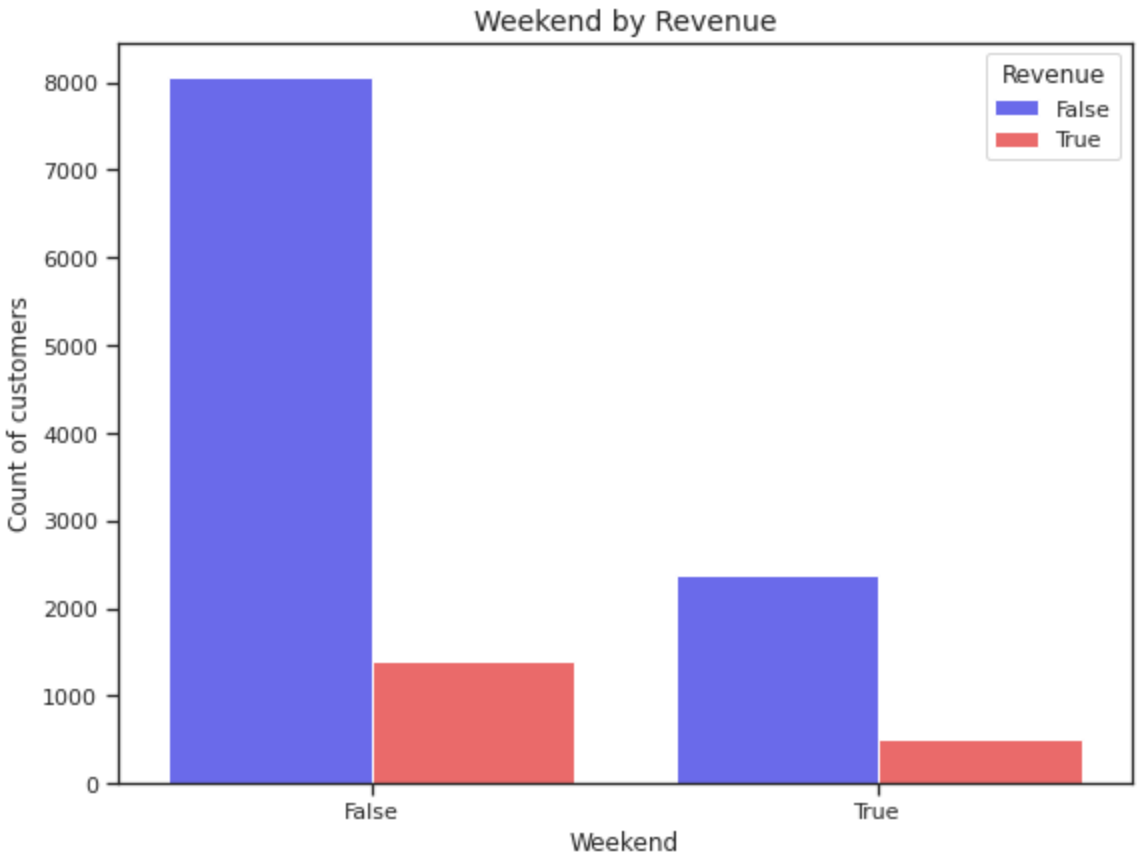
Hình 33. Phân tích thuộc tính "Weekend"

Biểu đồ cột ở trên thể hiện rằng liệu các phiên có diễn ra vào cuối tuần hay không. Từ biểu đồ, có thể thấy rằng 76,6% số phiên không xảy ra vào cuối tuần và 23,4% số phiên xảy ra vào cuối tuần.

- Tác động của ngày cuối tuần (thuộc tính "Weekend") đến doanh thu bán hàng (thuộc tính "Revenue")

```
plt.figure(figsize = (20,15))


ax = plt.subplot(221)
```

```
ax = sns.countplot(x="Weekend", data=df,
                   palette="seismic", hue="Revenue")
ax.set_xlabel("Weekend", fontsize=12)
ax.set_ylabel("Count of customers", fontsize=12)
ax.set_title("Weekend by Revenue", fontsize=14)
```



Hình 34. Tác động của Weekend đến Revenue

Phần trăm khách hàng mua hàng vào cuối tuần khá thấp. Hầu hết khách đến và mua hàng vào các ngày trong tuần.

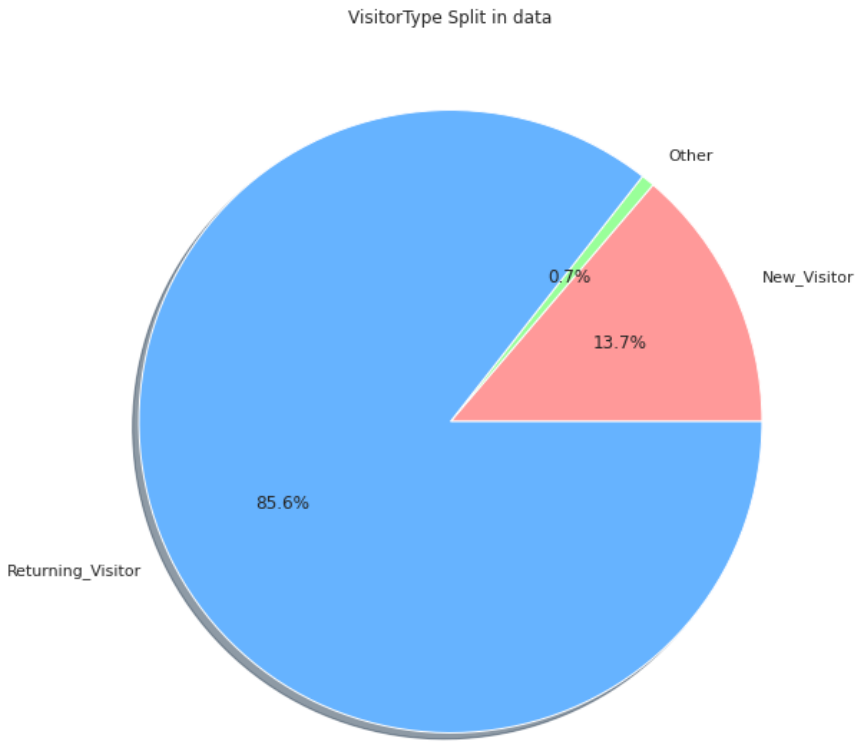
Đây là một "sự thật" (fact) về hành vi của khách hàng. Dù vậy, có thể cải thiện để tạo ra doanh thu nhiều hơn vào cuối tuần bằng hướng giải quyết sau:

Hướng phát triển 5: Ra mắt các chiến dịch marketing vào thời điểm cuối tuần hoặc tổ chức nhiều sự kiện khuyến mãi hơn vào cuối tuần để thúc đẩy khách hàng mua hàng nhiều hơn.

3.4. Tác động của loại khách hàng lên doanh thu

- Phân tích thuộc tính "VisitorType"

```
data_VisitorType =
df.groupby('VisitorType')['VisitorType'].count()
data_VisitorType =
pd.DataFrame({'VisitorType':data_VisitorType.index,
'Count':data_VisitorType.values})
plt.figure(figsize = (10,10))
colors = ['#ff9999','#99ff99','#66b3ff']
plt.pie(data_VisitorType['Count'],colors = colors, labels =
data_VisitorType['VisitorType'],autopct='%1.1f%%',shadow=True)
;
plt.title('VisitorType Split in data');
```

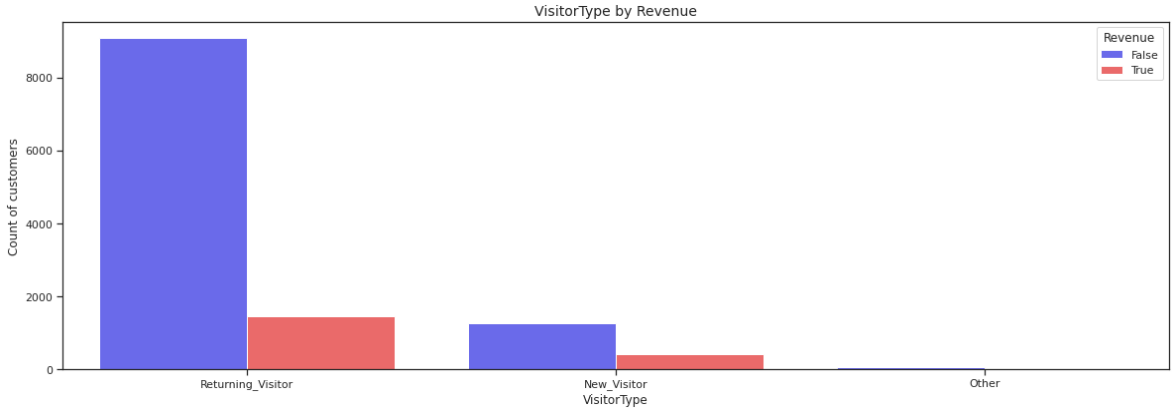


Hình 35. Biểu đồ tròn thể hiện các loại khách hàng truy cập vào trang web

Có 85.5% khách hàng cũ và 13.9% là khách hàng mới.

- Tác động của các loại khách hàng (thuộc tính "VisitorType") lên doanh thu bán hàng (thuộc tính "Revenue")

```
plt.figure(figsize = (20,15))
ax2 = plt.subplot(212)
ax2 = sns.countplot(x="VisitorType", data=df,
                    palette="seismic", hue="Revenue")
ax2.set_xlabel("VisitorType", fontsize=12)
ax2.set_ylabel("Count of customers", fontsize=12)
ax2.set_title("VisitorType by Revenue", fontsize=14)

plt.subplots_adjust(wspace = 0.6, hspace = 0.4, top = 0.9)
```



Hình 36. Tác động của VisitorType đến Revenue

Hình trên mô tả rằng hầu hết các khách hàng cho dù họ có mua hàng hay không, đều là khách hàng cũ, cho thấy rằng công ty đã xử lý tốt việc giữ chân khách hàng. Tuy nhiên, cần đẩy mạnh hơn nữa những hành động nhằm làm tăng tỷ lệ chuyển đổi. Mặc dù tỷ lệ giữ chân nói lên giá trị thương hiệu, nhưng nếu không có khách hàng mới, thì điều này có thể ảnh hưởng đáng kể đến doanh số bán hàng và tăng trưởng doanh thu.

Nguyên nhân của việc tỉ lệ chuyển đổi thấp có thể là do:

- Công ty chưa có chiến lược để tăng tỷ lệ chuyển đổi.

Hướng phát triển 6: Sử dụng khách hàng trung thành làm công cụ nhằm thu hút những khách hàng khác, bằng cách cung cấp chương trình ưu đãi (offer discounts) nếu họ mời bạn bè của họ tham gia mua hàng. Đối với những khách hàng mới, cũng sẽ áp dụng theo cách thức này để họ tiếp tục giới thiệu thêm nhiều người mới hơn nữa. Với phương pháp này, không chỉ khách hàng trung thành cảm thấy được quan tâm mà ngay cả khách hàng mới cũng thấy hứng thú hơn. Đồng thời giúp doanh nghiệp gia tăng tỷ lệ chuyển đổi cũng như thu hút thêm nhiều khách hàng mới.

Hướng phát triển 7: Xây dựng và phát triển trang mạng xã hội thông qua các nút chia sẻ mạng xã hội. Qua đó cho phép người truy cập tham gia và like các trang mạng như: Facebook, LinkedIn, Pinterest, Twitter,… Càng có được nhiều like, share và tweet, nhận dạng thương hiệu

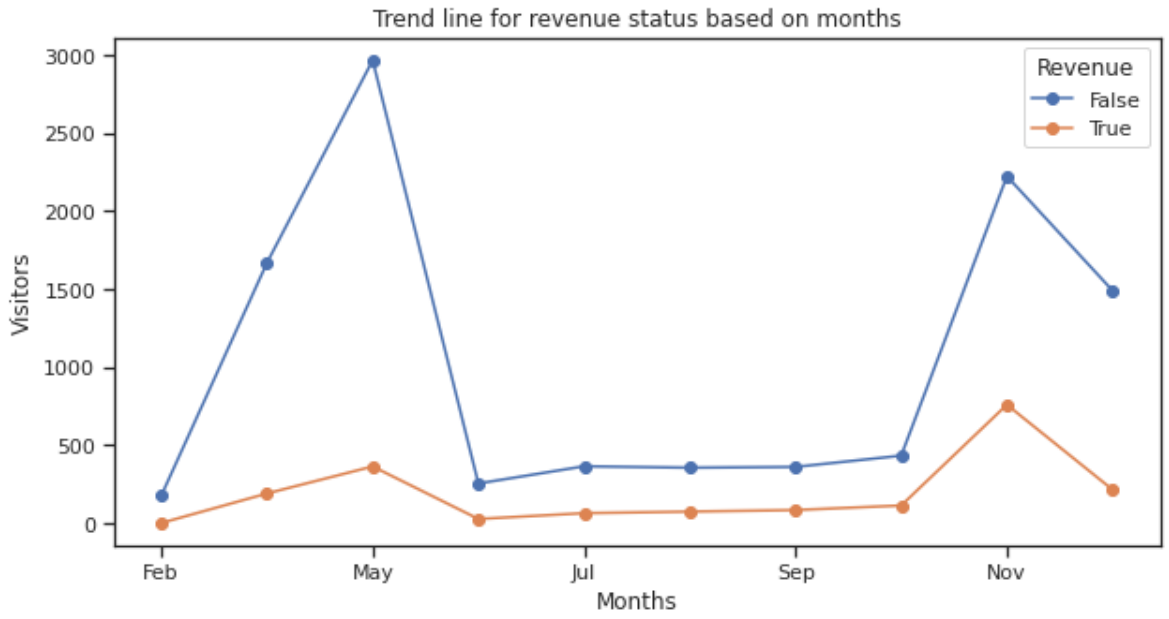
của doanh nghiệp sẽ càng mạnh mẽ. Khi người tiêu dùng đã trở nên quen thuộc hơn với một thương hiệu nhất định, xác suất để họ thường xuyên truy cập vào website càng cao.

- Các nguyên nhân về UX/UI, giá sản phẩm, loại sản phẩm. Hướng giải quyết cho vấn đề này sẽ tương tự hướng giải quyết 1, 2, 3, 4.

3.5. Tác động của ngày lễ lên tỷ lệ chuyển đổi từ khách hàng mới sang khách hàng cũ

- Khách hàng truy cập vào trang web khi nào?

```
#Trend line for revenue status based on months
df1 =
df.groupby(['Month','Revenue'])['Revenue'].count().unstack('Re
venue').fillna(0)
# arrange df1 by month
df1 =
df1.reindex(['Feb','Mar','May','June','Jul','Aug','Sep','Oct',
'Nov','Dec'])
# show all months in trend line chart
df1.plot(kind='line',figsize=(10,5),title='Trend line for
revenue status based on months', marker='o')
plt.xlabel('Months')
plt.ylabel('Visitors')
```



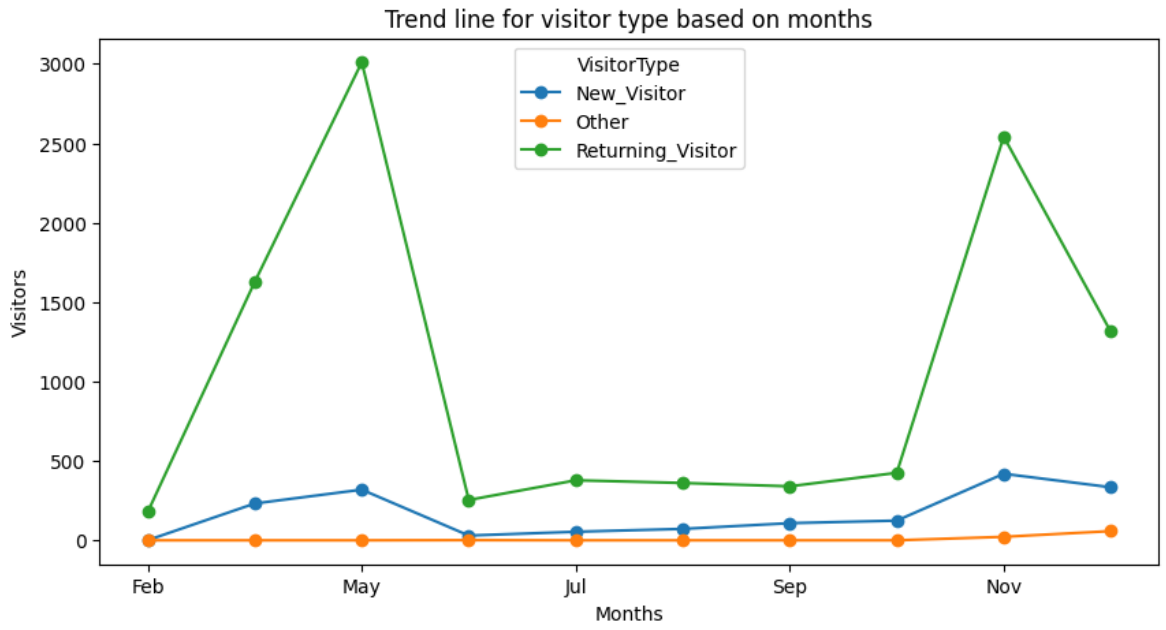
Hình 37. Mức tăng trưởng doanh thu theo từng tháng

Hình trên mô tả mức tăng trưởng doanh thu theo từng tháng. Có thể thấy mức độ tương tác của khách hàng với trang web cao trong các tháng 3, tháng 5 và tháng 11. Hơn nữa, trong khoảng thời gian từ tháng 6 đến tháng 10, lượt khách hàng truy cập vào trang web khá thấp cho đến khi Black Friday đến gần thì mức độ tương tác tăng cao dần (tháng 11).

- Từng loại khách hàng truy cập thường xuyên vào trang web khi nào?

```
# trend line for visitor type based on months
df2 =
df.groupby(['Month','VisitorType'])['VisitorType'].count().uns
tack('VisitorType').fillna(0)
# arrange df2 by month
df2 =
df2.reindex(['Feb','Mar','May','June','Jul','Aug','Sep','Oct',
'Nov','Dec'])
# show all months in trend line chart
```

```
df2.plot(kind='line',figsize=(10,5),title='Trend line for
visitor type based on months', marker='o')
plt.xlabel('Months')
plt.ylabel('Visitors')
```



Hình 38. Các loại khách hàng truy cập vào trang web thường xuyên khi nào

Vào những tháng có lượt truy cập trang web cao, nhìn có vẻ như sẽ có nhiều tương tác nhưng tỷ lệ chuyển đổi thấp hơn đáng kể do hầu hết các giao dịch mua này được thực hiện bởi khách hàng cũ. Mặc dù điều này cho thấy sự trung thành của tệp khách hàng cũ tốt, nhưng cần phải chú ý nhiều hơn đến tỷ lệ chuyển đổi vì các biểu đồ ở trên cho thấy rằng rất nhiều khách hàng đang xem sản phẩm của web nhưng không có ý định mua hàng.

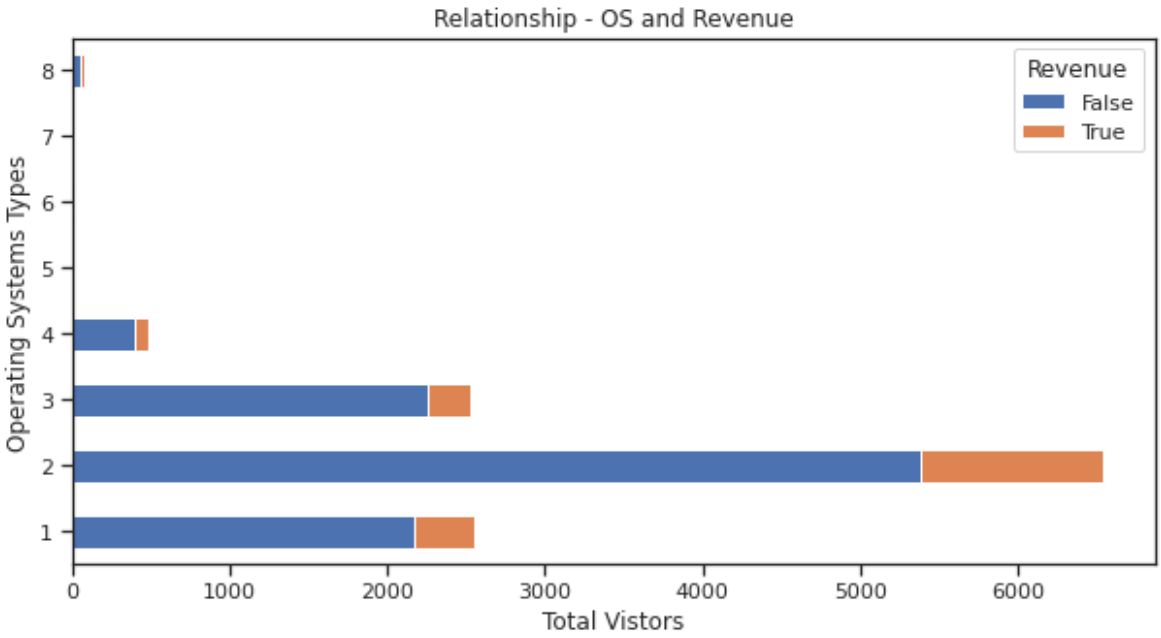
Để cải thiện phần này, nhóm đề xuất một số hướng phát triển như sau:

Hướng phát triển 8: Giới thiệu các chương trình khuyến mãi theo mùa với các ưu đãi và sự kiện hấp dẫn, thu hút nhiều lượt chuyển đổi hơn và đảm bảo khách hàng trung thành có một phần lợi ích trong việc mang lại các lượt chuyển đổi mới.

3.6. Tác động của các yếu tố khác lên doanh thu

- Mối quan hệ giữa doanh thu với hệ điều hành (OperatingSystem)

```
df3 =
df.groupby(['OperatingSystems','Revenue'])['Revenue'].count().
unstack('Revenue').fillna(0)
#Relationship between Operating Systems and Revenue
df3.plot(kind='barh',figsize=(10,5),title='Relationship - OS
and Revenue', stacked=True)
plt.xlabel('Total Vistors')
plt.ylabel('Operating Systems Types')
plt.show()
```



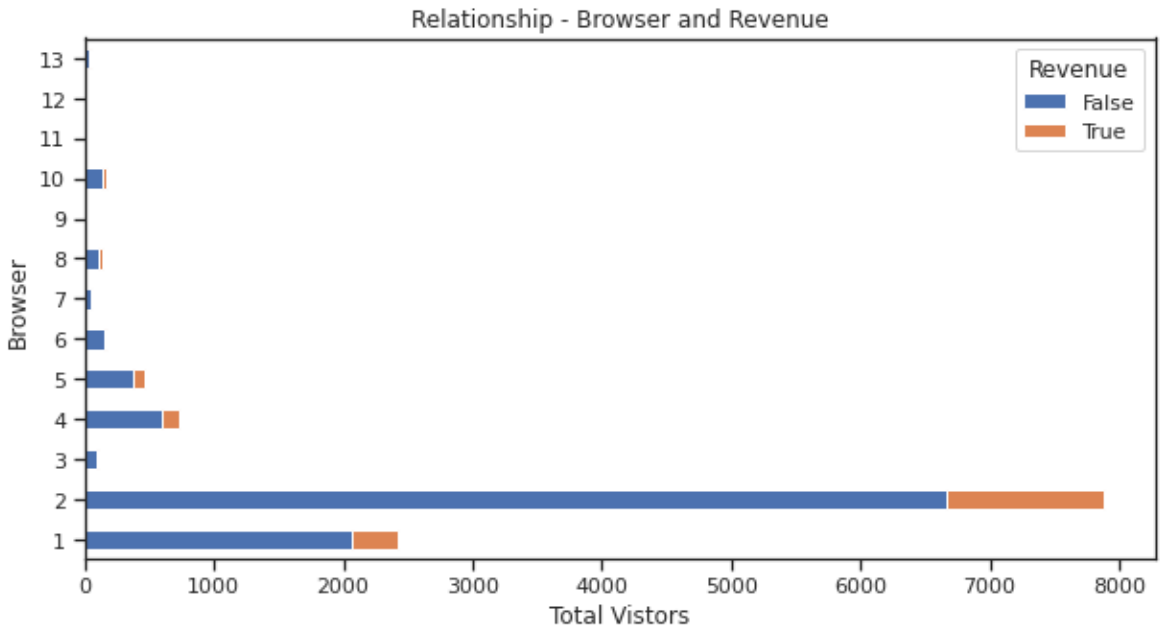
Hình 39. Mối quan hệ giữa OperatingSystems và Revenue

Hệ điều hành (operating system) được sử dụng nhiều nhất là hệ điều hành 2, được sử dụng bởi cả khách hàng có mua hàng lẫn không mua hàng. Tiếp theo là hệ điều hành 1 và 3, những hệ điều hành còn lại thì không có nhiều khách hàng sử dụng. Điều này có thể là do trang web không

dễ dùng trên các hệ điều hành đó hoặc do các hệ điều hành này vốn không có nhiều người sử dụng.

- Mối quan hệ giữa doanh thu với trình duyệt (Browser)

```
df4 =
df.groupby(['Browser','Revenue'])['Revenue'].count().unstack('
Revenue').fillna(0)
df4
#Relationship between Browser and Revenue
df4.plot(kind='barh',figsize=(10,5),title='Relationship -
Browser and Revenue', stacked=True)
plt.xlabel('Total Vistors')
plt.ylabel('Browser')
plt.show()
```



Hình 40. Mối quan hệ giữa Browser và Revenue

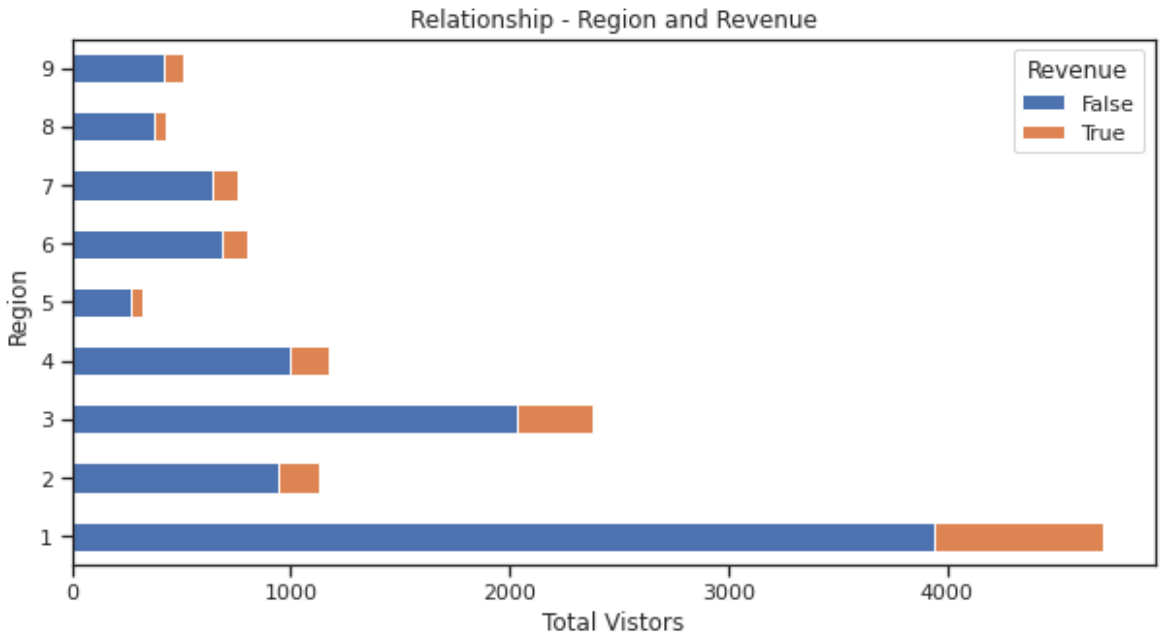
Trình duyệt (browser) 2 được sử dụng nhiều nhất, sau đó đến 1, 4 và 5, được dùng bởi cả khách hàng có mua hàng và không mua hàng. Nguyên nhân có thể là tương tự với nguyên nhân của hệ điều hành đã được trình bày ở trên.

Để cải thiện phần này, nhóm đề xuất một số hướng phát triển như sau:

Hướng phát triển 9: Đảm bảo vận hành kỹ thuật trơn tru với trải nghiệm giao diện người dùng nâng cao và cá nhân hóa, có thể hỗ trợ được tất cả các trình duyệt và hệ điều hành.

- Mối quan hệ giữa doanh thu với Region (Khu vực)

```
df5 =
df.groupby(['Region','Revenue'])['Revenue'].count().unstack('R
evenue').fillna(0)
#Relationship between Region and Revenue
df5.plot(kind='barh',figsize=(10,5),title='Relationship -
Region and Revenue', stacked=True)
plt.xlabel('Total Vistors')
plt.ylabel('Region')
plt.show()
```



Hình 41. Mối quan hệ giữa Region và Revenue

Đối với khu vực (region), biểu đồ trên cho thấy trang web được truy cập từ nhiều nơi khác nhau. Có thể thấy hầu hết khách truy cập các trang web thuộc khu vực 1, tiếp theo là khu vực 3, được ghi nhận từ cả khách có mua hàng và không mua hàng. Khách hàng ở khu vực 1 dẫn dầu trong việc truy cập vào website đã chỉ ra rằng phạm vi marketing đã chạm được đến khu vực này khá tốt. Bên cạnh đó, cũng nên cố gắng để mở rộng phạm vi marketing sang các khu vực khác.

Nguyên nhân của việc này có thể là do:

- Phạm vi marketing đã chạm được đến khu vực này khá tốt. Bên cạnh đó, cũng nên cố gắng để mở rộng phạm vi marketing sang các khu vực khác.

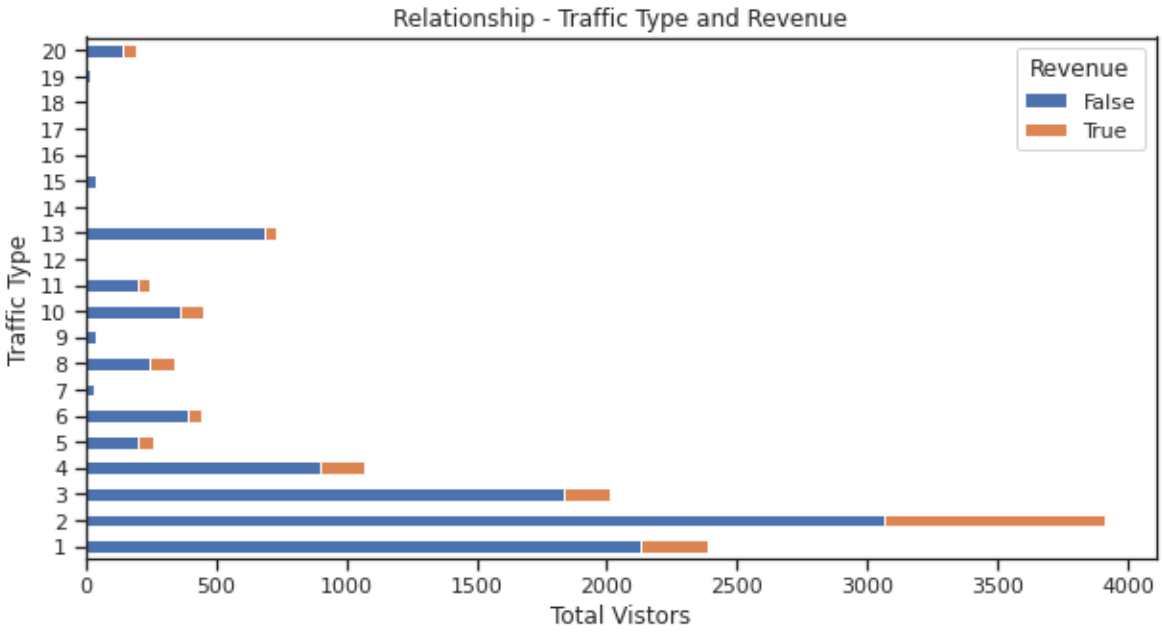
Hướng phát triển 10: Dựa vào phương pháp marketing cho khu vực 1, điều chỉnh để marketing cho những khu vực còn lại, tuy nhiên cần phải quan tâm đến các yếu tố văn hóa và yếu tố xã hội ở từng khu vực để có chiến dịch phù hợp hơn. Cá nhân hóa các quảng cáo và thực hiện A/B Testing nhằm đảm bảo tỷ lệ chuyển đổi và tỷ lệ giữ chân khách hàng đều phát triển ở hầu hết các khu vực.

- Phí giao hàng: Có thể thấy khách hàng ở khu vực 1 mua hàng nhiều hơn đáng kể so với các khu vực khác. Có thể do phí giao hàng ở khu vực 1 thấp hơn ở khu vực khác.

Hướng phát triển 11: Điều chỉnh và lựa chọn các gói phí giao hàng cho phù hợp ở các khu vực.

- Mối quan hệ giữa doanh thu với lưu lượng truy cập (Traffic Type)

```
df6 =
df.groupby(['TrafficType','Revenue'])['Revenue'].count().unsta
ck('Revenue').fillna(0)
#Relationship between traffic type and Revenue
df6.plot(kind='barh',figsize=(10,5),title='Relationship -
Traffic Type and Revenue', stacked=True)
plt.xlabel('Total Vistors')
plt.ylabel('Traffic Type')
plt.show()
```



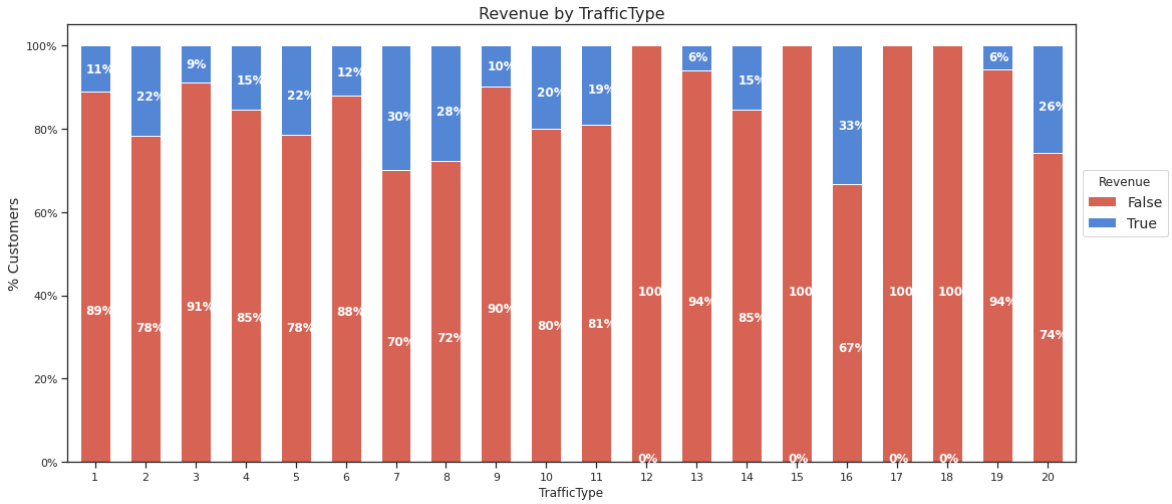
Hình 42. Mối quan hệ giữa TrafficType và Revenue

Lưu lượng truy cập số 2 dẫn đầu, theo sau là 1 và 3. Điều này chỉ ra tác động của việc tối ưu hóa Google SEO.

Tuy nhiên, trong tổng số lưu lượng truy cập thì phần trăm tạo ra doanh thu không quá cao. Vì vậy nên cần cải thiện SEO và đẩy mạnh quảng cáo trên Google và trên cả nền tảng mạng xã hội.

```
import matplotlib.ticker as mtick
traffic_revenue =
df.groupby(['TrafficType','Revenue']).size().unstack()
ax = (traffic_revenue.T*100.0 /
traffic_revenue.T.sum()).T.plot(kind='bar',
                                        width =
0.6,
stacked =
True,
    rot = 0,color = ['#d66354','#5486d6']
,                                                              fi
gsize = (18,8))
ax.yaxis.set_major_formatter(mtick.PercentFormatter())
ax.legend(bbox_to_anchor=(1, 0.5),loc='lower
left',prop={'size':14},title = 'Revenue')
ax.set_ylabel('% Customers',size = 14)
ax.set_title('Revenue by TrafficType',size = 16)
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.annotate('{:.0f}%'.format(height),
(p.get_x()+.20*width, p.get_y()+.4*height),
                color = 'white',
```

```
                    weight = 'bold',size =12)
```



Hình 43. Ảnh hưởng của TrafficType lên Revenue

Loại 12, 15, 17, 18 không có doanh thu.

Loại lưu lượng truy cập 16 cho thấy phần trăm khách hàng mua hàng cao nhất, chiếm 33%

Hầu như ít cơ hội đạt được doanh thu ở các loại lưu lượng truy cập.

Để cải thiện phần này, nhóm đề xuất một số hướng phát triển như sau:

Hướng phát triển 12: Đảm bảo tối ưu hóa SEO từ Google. Thực hiện A/B Testing cụ thể theo khu vực và độ tuổi trong Google Ads, Facebook Ads hoặc các nguồn khác.

CHƯƠNG 3: THỰC NGHIỆM VÀ PHÂN TÍCH

Thực nghiệm và phân tích thuật toán.

1. IMPORT CÁC THƯ VIỆN VÀ ĐỌC TẬP DỮ LIỆU

```
import pandas as pd
import numpy as np

from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier,
AdaBoostClassifier
from sklearn.utils import resample
from sklearn.model_selection import RandomizedSearchCV,
GridSearchCV, train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score
from scipy.special import boxcox1p
from sklearn.preprocessing import PowerTransformer
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from numpy import mean
from numpy import std
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
```

```
from sklearn.metrics import f1_score
import scipy
import scipy.stats as stats
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
import pandas as pd
import numpy as np

from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.utils import resample
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV, train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score
from scipy.special import boxcox1p
from sklearn.preprocessing import PowerTransformer
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from numpy import mean
from numpy import std
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
import scipy
import scipy.stats as stats
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

import seaborn as sea
import matplotlib.pyplot as plt
%matplotlib inline
```

Hình 44. Import các thư viện cần thiết

```
df= pd.read_csv('online_shoppers_intention.csv')
df.head()
```

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRate
0	0	0.0	0	0.0	1	0.000000	0.20	0.2
1	0	0.0	0	0.0	2	64.000000	0.00	0.1
2	0	0.0	0	0.0	1	0.000000	0.20	0.2
3	0	0.0	0	0.0	2	2.666667	0.05	0.1
4	0	0.0	0	0.0	10	627.500000	0.02	0.0

Hình 45. Đọc tập dữ liệu

2. ONE – HOT ENCODING VÀ LABEL ENCODING

- One – hot encoding

Cách biến đổi dummy hay mã hóa one-hot là một cách rất hiệu quả để biến đổi một biến thành một one-hot vector,cực kỳ hữu ích khi xây dựng mô hình đơn giản,các mô hình này bắt buộc giá trị đầu vào là ở dạng số.

Ta dùng hàm ‘get_dummies’ này để tìm toàn bộ các biến (có định dạng trường là object) và tự động trải phẳng chúng .Cấu trúc của các thuộc tính đã được thay đổi theo cơ sở phân loại và số.

- Label encoding

Cách biến đổi các giá trị sau khi mã hóa nhãn, giá trị số được gán cho từng giá trị phân loại. Thay thế giá trị phân loại bằng một giá trị số trong khoảng từ 0 đến số lớp trừ 1. Nếu giá trị biến phân loại chứa m lớp riêng biệt, sẽ sử dụng m-1 số gán.

Các cột Revenue,Weekend có giá trị là True,False được gán giá trị 1,0.

```
df1=pd.get_dummies(df)
df1.head()
le= LabelEncoder()
df1['Revenue']=le.fit_transform(df1['Revenue'])
df1['Weekend']=le.fit_transform(df1['Weekend'])
df1.head()
```

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay	...	Month_Ju
0	0	0.0	0	0.0	1	0.000000	0.20	0.20	0.0	0.0	...	
1	0	0.0	0	0.0	2	64.000000	0.00	0.10	0.0	0.0	...	
2	0	0.0	0	0.0	1	0.000000	0.20	0.20	0.0	0.0	...	
3	0	0.0	0	0.0	2	2.666667	0.05	0.14	0.0	0.0	...	
4	0	0.0	0	0.0	10	627.500000	0.02	0.05	0.0	0.0	...	

5 rows × 29 columns

Hình 46. One - hot encoding và Label encoding

3. CHIA FEATURES VÀ GÁN NHÃN CHO TẬP DỮ LIỆU

Sau đó ta chia tập dữ liệu đã được chuẩn hóa thành 2 phần theo tỉ lệ 8:2 bao gồm train (dùng để tìm các hệ số và xây dựng mô hình) và test (dùng để đưa ra các giá trị predict và đánh giá mô hình).

```
x1=df1.drop('Revenue', axis=1)
y1=df1['Revenue']
x1_train, x1_test, y1_train, y1_test= train_test_split(x1,y1,
test_size=0.20)
print(x1.shape)
print(y1.shape)
```

```
[23] x1=df1.drop('Revenue', axis=1)
     y1=df1['Revenue']
```

```
x1_train, x1_test, y1_train, y1_test= train_test_split(x1,y1, test_size=0.20)
print(x1.shape)
print(y1.shape)
```

```
(10747, 28)
(10747,)
```

Hình 47. Chia features và gán nhãn cho tập dữ liệu

4. XÂY DỰNG MÔ HÌNH

4.1. Naive Bayes

Sử dụng thuật toán Naive Bayes trên tập dữ liệu và tiến hành đánh giá score:

```
# Naive Bayes
nb = GaussianNB()
nb.fit(x1_train,y1_train)
nb.score(x1_train, y1_train), nb.score(x1_test, y1_test)
```

```
[33] # Naive Bayes
     nb = GaussianNB()
     nb.fit(x1_train,y1_train)

     GaussianNB()

[34] nb.score(x1_train, y1_train), nb.score(x1_test, y1_test)

     (0.7921367919041526, 0.7879069767441861)
```

Hình 48. Sử dụng thuật toán Naive Bayes

Kết quả confusion matrix:

```
# Predicted labels for X train
y_pred_train= nb.predict(x1_train)
# Predicted labels for X test
y_pred_test= nb.predict(x1_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y1_test,y_pred_test)
print(cm)
```

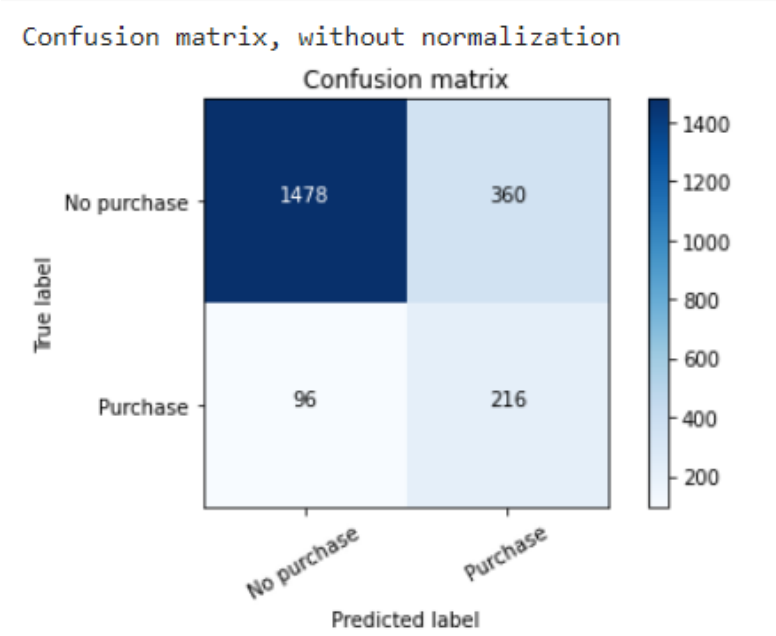
```
[42] # Predicted labels for X train
     y_pred_train= nb.predict(x1_train)
     # Predicted labels for X test
     y_pred_test= nb.predict(x1_test)

     from sklearn.metrics import confusion_matrix
     cm = confusion_matrix(y1_test,y_pred_test)
     print(cm)
```

```
[[1478  360]
 [  96  216]]
```

Hình 49. Kết quả confusion matrix - NBC

Trực quan hóa confusion matrix:

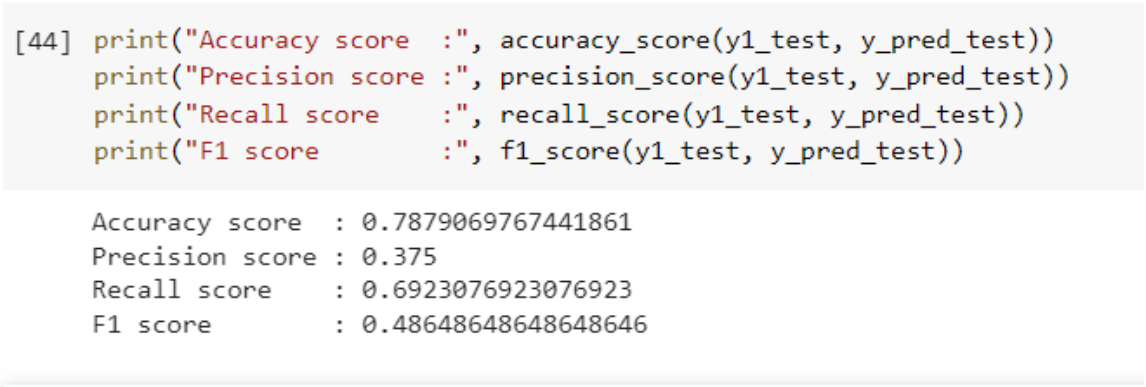
```
classes = ['No purchase', 'Purchase']
plot_confusion_matrix(cm, classes)
```



Hình 50. Trực quan hóa confusion matrix - NBC

Đánh giá các chỉ số:

```
print("Accuracy score  :", accuracy_score(y1_test,
y_pred_test))
print("Precision score :", precision_score(y1_test,
y_pred_test))
print("Recall score    :", recall_score(y1_test, y_pred_test))
print("F1 score        :", f1_score(y1_test, y_pred_test))
```



Hình 51. Đánh giá các chỉ số - NBC

4.2. Decision Tree

Sử dụng thuật toán Decision Tree trên tập dữ liệu và tiến hành đánh giá score:

```
# Decision Tree
dt= DecisionTreeClassifier()
dt.fit(x1_train,y1_train)
dt.score(x1_train, y1_train),dt.score(x1_test, y1_test)
```

```
[37] # Decision Tree
     dt= DecisionTreeClassifier()
     dt.fit(x1_train,y1_train)

     DecisionTreeClassifier()

[38] dt.score(x1_train, y1_train),dt.score(x1_test, y1_test)

     (1.0, 0.8488372093023255)
```

Hình 52. Sử dụng thuật toán Decision Tree

Kết quả confusion matrix:

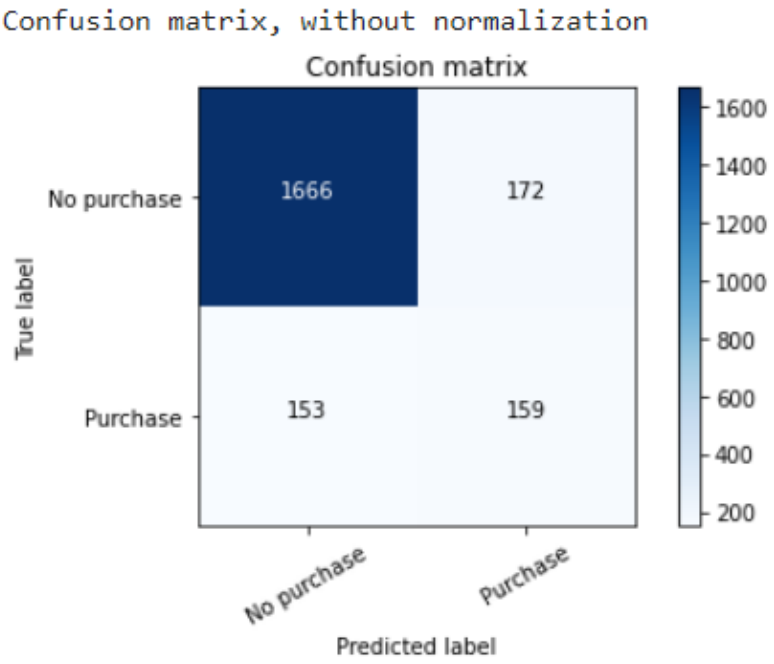
```
# Predicted labels for X train
y_pred_train= dt.predict(x1_train)
# Predicted labels for X test
y_pred_test= dt.predict(x1_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y1_test,y_pred_test)
print(cm)
```

```
# Predicted labels for X train
y_pred_train= dt.predict(x1_train)
# Predicted labels for X test
y_pred_test= dt.predict(x1_test)

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y1_test,y_pred_test)
print(cm)

[[1666  172]
 [ 153  159]]
```

Hình 53. Kết quả confusion matrix - Decision Tree

Trực quan hóa confusion matrix:

```
classes = ['No purchase', 'Purchase']
plot_confusion_matrix(cm, classes)
```



Hình 54. Trực quan hóa confusion matrix - Decision Tree

Đánh giá các chỉ số:

```
print("Accuracy score  :", accuracy_score(y1_test,
y_pred_test))
print("Precision score :", precision_score(y1_test,
y_pred_test))
print("Recall score    :", recall_score(y1_test, y_pred_test))
print("F1 score        :", f1_score(y1_test, y_pred_test))
```

```
[47] print("Accuracy score  :", accuracy_score(y1_test, y_pred_test))
     print("Precision score :", precision_score(y1_test, y_pred_test))
     print("Recall score    :", recall_score(y1_test, y_pred_test))
     print("F1 score        :", f1_score(y1_test, y_pred_test))

     Accuracy score  : 0.8488372093023255
     Precision score : 0.48036253776435045
     Recall score    : 0.5096153846153846
     F1 score        : 0.494556765163297
```

Hình 55. Đánh giá các chỉ số - Decision Tree

4.3. KNN

Sử dụng thuật toán KNN trên tập dữ liệu và tiến hành đánh giá score:

```
#knn
knn = KNeighborsClassifier()
knn.fit(x1_train,y1_train)
knn.score(x1_train, y1_train),knn.score(x1_test, y1_test)
```

```
[35] #knn
     knn = KNeighborsClassifier()
     knn.fit(x1_train,y1_train)

     KNeighborsClassifier()

[36] knn.score(x1_train, y1_train),knn.score(x1_test, y1_test)

     (0.8935675235547283, 0.867906976744186)
```

Hình 56. Sử dụng thuật toán KNN

Kết quả confusion matrix:

```
# Predicted labels for X train
y_pred_train= knn.predict(x1_train)
# Predicted labels for X test
y_pred_test= knn.predict(x1_test)
```

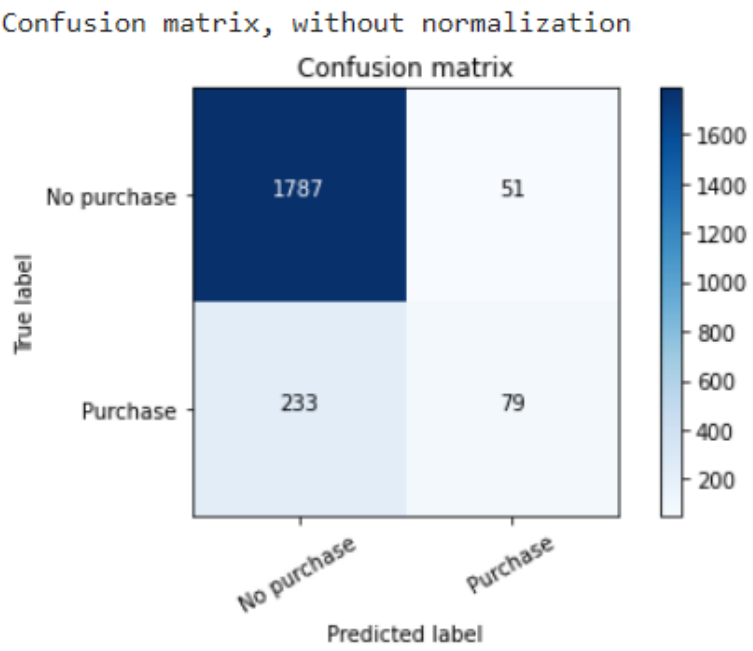
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y1_test,y_pred_test)
print(cm)
```

```
[48] # Predicted labels for X train
     y_pred_train= knn.predict(x1_train)
     # Predicted labels for X test
     y_pred_test= knn.predict(x1_test)

     from sklearn.metrics import confusion_matrix
     cm = confusion_matrix(y1_test,y_pred_test)
     print(cm)

     [[1787   51]
      [ 233   79]]
```

Hình 57. Kết quả confusion matrix - KNN

Trực quan hóa confusion matrix:

```
classes = ['No purchase', 'Purchase']
plot_confusion_matrix(cm, classes)
```



Hình 58. Trực quan hóa confusion matrix - KNN

Đánh giá các chỉ số:

```
print("Precision score :", precision_score(y1_test,
y_pred_test))
print("Recall score    :", recall_score(y1_test, y_pred_test))
print("F1 score        :", f1_score(y1_test, y_pred_test))
```

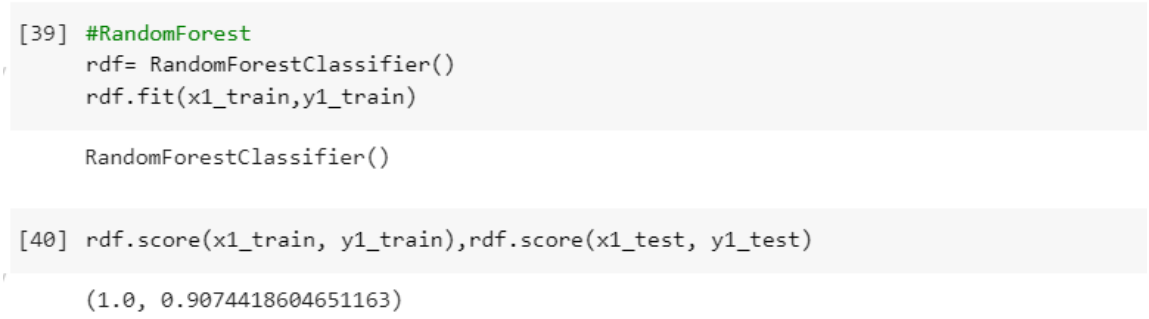
```
[50] print("Accuracy score  :", accuracy_score(y1_test, y_pred_test))
     print("Precision score :", precision_score(y1_test, y_pred_test))
     print("Recall score    :", recall_score(y1_test, y_pred_test))
     print("F1 score        :", f1_score(y1_test, y_pred_test))

     Accuracy score  : 0.867906976744186
     Precision score : 0.6076923076923076
     Recall score    : 0.2532051282051282
     F1 score        : 0.3574660633484163
```

Hình 59. Đánh giá các chỉ số - KNN

4.4. Random Forest

Sử dụng thuật toán Random Forest trên tập dữ liệu và tiến hành đánh giá score:

```
#RandomForest
rdf= RandomForestClassifier()
rdf.fit(x1_train,y1_train)
rdf.score(x1_train, y1_train),rdf.score(x1_test, y1_test)
```



Hình 60. Sử dụng thuật toán Random Forest

Kết quả confusion matrix:

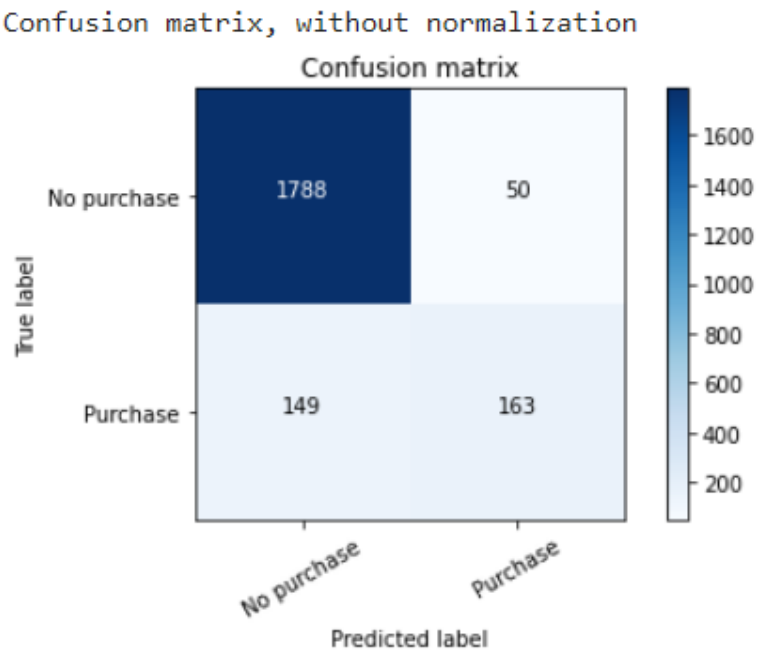
```
# Predicted labels for X train
y_pred_train= rdf.predict(x1_train)
# Predicted labels for X test
y_pred_test= rdf.predict(x1_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y1_test,y_pred_test)
print(cm)
```

```
[51] # Predicted labels for X train
     y_pred_train= rdf.predict(x1_train)
     # Predicted labels for X test
     y_pred_test= rdf.predict(x1_test)

     from sklearn.metrics import confusion_matrix
     cm = confusion_matrix(y1_test,y_pred_test)
     print(cm)

     [[1788   50]
      [ 149  163]]
```

Hình 61. Kết quả confusion matrix - Random Forest

Trực quan hóa confusion matrix:

```
classes = ['No purchase', 'Purchase']
plot_confusion_matrix(cm, classes)
```



Hình 62. Trực quan hóa confusion matrix - Random Forest

Đánh giá các chỉ số:

```
print("Accuracy score  :", accuracy_score(y1_test,
y_pred_test))
print("Precision score :", precision_score(y1_test,
y_pred_test))
print("Recall score    :", recall_score(y1_test, y_pred_test))
print("F1 score        :", f1_score(y1_test, y_pred_test))
```

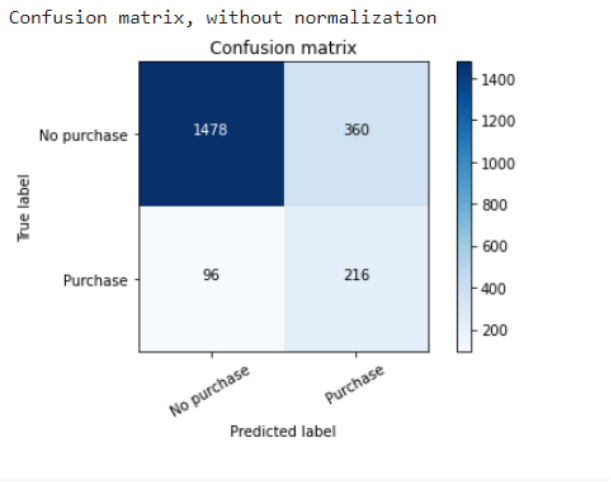
```
[53] print("Accuracy score  :", accuracy_score(y1_test, y_pred_test))
     print("Precision score :", precision_score(y1_test, y_pred_test))
     print("Recall score    :", recall_score(y1_test, y_pred_test))
     print("F1 score        :", f1_score(y1_test, y_pred_test))

     Accuracy score  : 0.9074418604651163
     Precision score : 0.7652582159624414
     Recall score    : 0.5224358974358975
     F1 score        : 0.620952380952381
```

Hình 63. Đánh giá các chỉ số - Random Forest

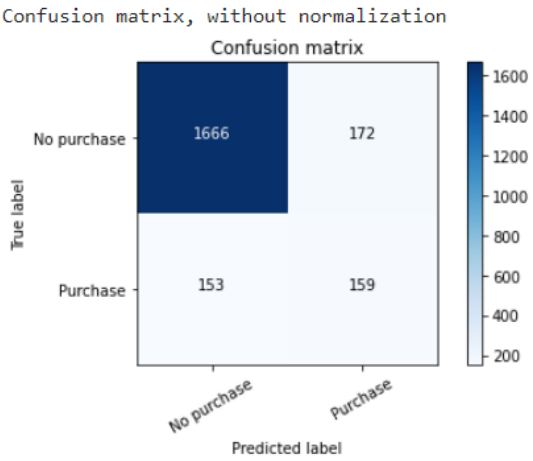
CHƯƠNG 4: ĐÁNH GIÁ VÀ LỰA CHỌN MÔ HÌNH

Đánh giá và lựa chọn mô hình tốt nhất. Bên cạnh đó, tìm ra yếu tổ ảnh hưởng nhiều nhất đến việc dự đoán ý định mua hàng.

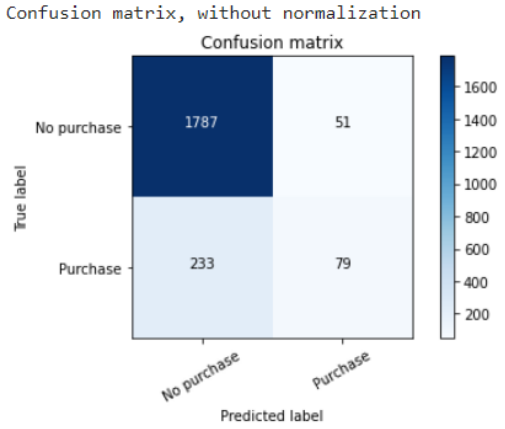
1. MA TRẬN NHẦM LẪN



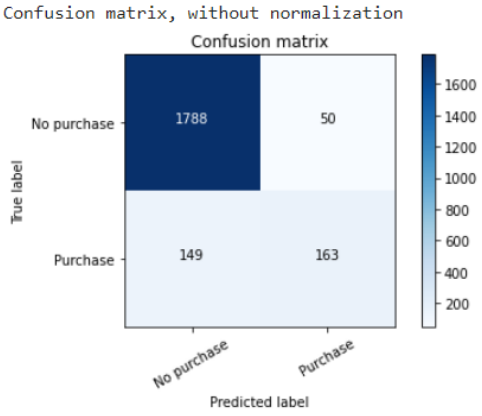
Hình 64. Naive Bayes



Hình 65. Decision Tree



Hình 66. KNN



Hình 67. Random Forest

Nhận xét:

Từ các ma trận nhầm lẫn ở trên, chúng ta có thể thấy rằng:

- Mô hình Naive Bayes có 1478 dự đoán đúng không mua, 216 dự đoán đúng mua. Có 96 dự đoán sai mua thành không mua, 360 dự đoán sai không mua thành mua.

- Mô hình Decision Tree có 1666 dự đoán đúng không mua, 159 dự đoán đúng mua. Có 153 dự đoán sai mua thành không mua, 172 dự đoán sai không mua thành mua.
- Mô hình KNN có 1787 dự đoán đúng không mua, 79 dự đoán đúng mua. Có 233 dự đoán sai mua thành không mua, 51 dự đoán sai không mua thành mua.
- Mô hình Random forest có 1788 dự đoán đúng không mua, 163 dự đoán đúng mua. Có 149 dự đoán sai mua thành không mua, 50 dự đoán sai không mua thành mua.

Suy ra:

- Mô hình Random forest có dự đoán True Negative cao nhất, Mô hình Naive Bayes có dự đoán True Positive cao nhất.
- Mô hình Naive Bayes có dự đoán False Negative thấp nhất, Mô hình Random forest có dự đoán False Positive thấp nhất.

2. CÁC CHỈ SỐ ĐÁNH GIÁ

Thuật toán	Train score	Test score	Accuracy score	Precision score	Recall score	F1 score
Naive Bayes	0.79	0.78	0.79	0.39	0.69	0.49
KNN	0.89	0.86	0.85	0.48	0.51	0.50
Decision Tree	1.00	0.84	0.87	0.61	0.25	0.36
Random Forest	1.00	0.90	0.91	0.77	0.52	0.62

Bảng 2. Các chỉ số đánh giá

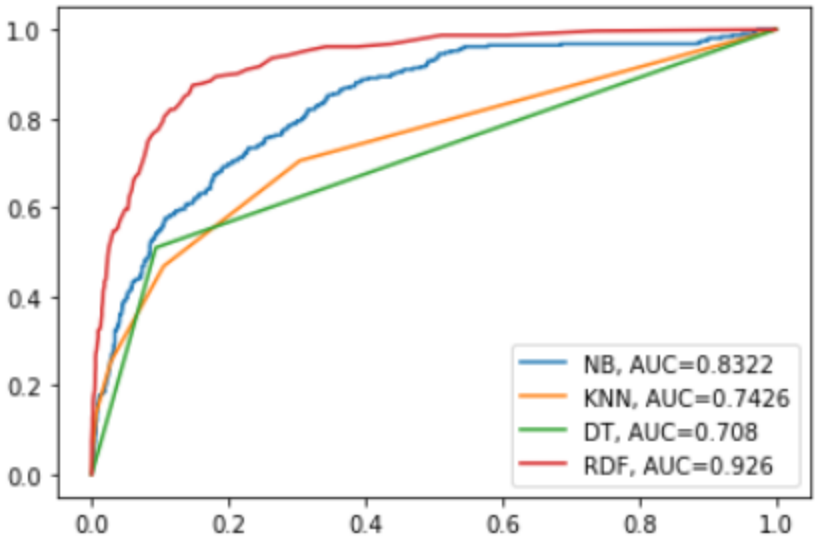
Nhận xét:

- Dựa vào các chỉ số cho thấy mô hình Decision Tree, Random forest có tỉ lệ đúng trên tập train chính xác 100%, mô hình Naive Bayes, KNN cũng khá cao.
- Chỉ số Accuracy score của các mô hình đều cao cho thấy tỷ lệ dự đoán đúng cao.
- Các chỉ số Precision score, Recall score, F1 score đều ở mức ổn.
- Mô hình Random forest có các chỉ số cao cho thấy mô hình hoạt động khá tốt.

3. CHỈ SỐ AUC

```
from sklearn import metrics
#set up plotting area
plt.figure(0).clf()

# model and plot ROC curve
#nb
y_pred = nb.predict_proba(x1_test)[:, 1]
fpr, tpr, _ = metrics.roc_curve(y1_test, y_pred)
auc = round(metrics.roc_auc_score(y1_test, y_pred), 4)
plt.plot(fpr,tpr,label="NB, AUC="+str(auc))

#knn
y_pred = knn.predict_proba(x1_test)[:, 1]
fpr, tpr, _ = metrics.roc_curve(y1_test, y_pred)
auc = round(metrics.roc_auc_score(y1_test, y_pred), 4)
plt.plot(fpr,tpr,label="KNN, AUC="+str(auc))

#dt
y_pred = dt.predict_proba(x1_test)[:, 1]
fpr, tpr, _ = metrics.roc_curve(y1_test, y_pred)
auc = round(metrics.roc_auc_score(y1_test, y_pred), 4)
plt.plot(fpr,tpr,label="DT, AUC="+str(auc))

#rdf
y_pred = rdf.predict_proba(x1_test)[:, 1]
fpr, tpr, _ = metrics.roc_curve(y1_test, y_pred)
auc = round(metrics.roc_auc_score(y1_test, y_pred), 4)
```

```
plt.plot(fpr,tpr,label="RDF, AUC="+str(auc))

#add legend
plt.legend();
```



Hình 68. Chỉ số AUC

Chỉ số AUC của mô hình Random Forest = 0.926 khá ổn, cao nhất trong các mô hình, có thể chọn mô hình này để đánh giá và đưa ra các quyết định.

- Tầm ảnh hưởng của các feature đối với kết quả dự đoán:

```
importances = rdf.feature_importances_
std = np.std([tree.feature_importances_ for tree in
rdf.estimators_],
             axis=0)
indices = np.argsort(importances)[::-1]

print("Feature ranking:")
```

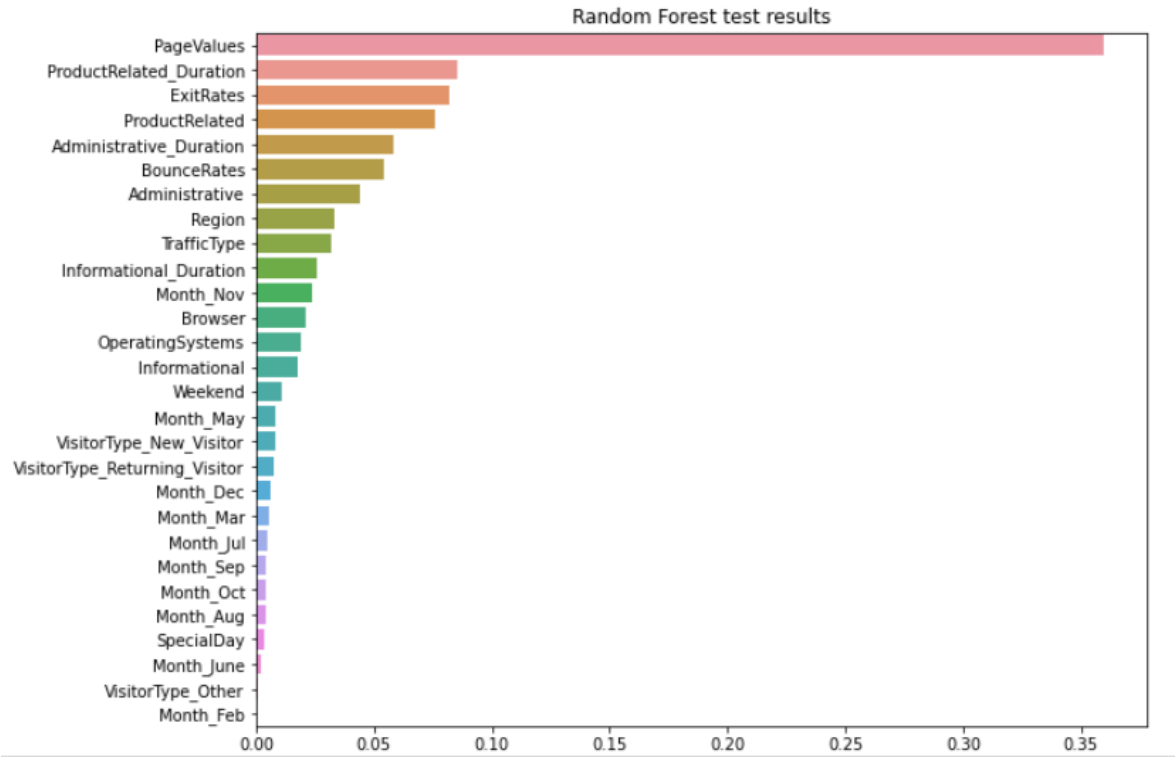
```
for f in range(x1_train.shape[1]):
    print("%d. Feature %d (%f)" % (f + 1, indices[f],
importances[indices[f]]))

col_names = pd.Series([col for col in x1_train.columns])

importance_df = pd.DataFrame(importances)
importance_df.rename(columns={0:'Importance'}, inplace=True)
importance_df.set_index(col_names,inplace=True)

imp_sorted = importance_df.sort_values(by='Importance',
ascending=False)
imp_sorted

feature_imp2 =
pd.Series(rdf.feature_importances_,index=x1_train.columns).sor
t_values(ascending=False)
fig, ax = plt.subplots(figsize=(10,8))
ax=sns.barplot(x=feature_imp2, y=feature_imp2.index)
ax.set_title('Random Forest test results')
ax.set(xlabel='Feature Importance')
```

```
Feature ranking:
1. Feature 8 (0.359822)
2. Feature 5 (0.085551)
3. Feature 7 (0.082050)
4. Feature 4 (0.076034)
5. Feature 1 (0.058350)
6. Feature 6 (0.053961)
7. Feature 0 (0.043758)
8. Feature 12 (0.033380)
9. Feature 13 (0.032195)
10. Feature 3 (0.025545)
11. Feature 22 (0.023551)
12. Feature 11 (0.020756)
13. Feature 10 (0.019076)
14. Feature 2 (0.017479)
15. Feature 14 (0.010824)
16. Feature 21 (0.007925)
```

Hình 69. Feature ranking

	Importance
PageValues	0.359822
ProductRelated_Duration	0.085551
ExitRates	0.082050
ProductRelated	0.076034
Administrative_Duration	0.058350
BounceRates	0.053961
Administrative	0.043758
Region	0.033380
TrafficType	0.032195
Informational_Duration	0.025545
Month_Nov	0.023551
Browser	0.020756
OperatingSystems	0.019076
Informational	0.017479
Weekend	0.010824
Month_May	0.007925

Hình 70. Important feature



Hình 71. Random Forest test result

Từ thuật toán Random Forest cho thấy feature PageValues có ảnh hưởng rất lớn đến kết quả dự đoán, tiếp theo có thể kể đến các features như ProductRelated_Duration, ExitRates, ProductRelated, Administrative_Duration,.... Mục đích là nắm bắt những thuộc tính đóng góp nhiều nhất vào tăng trưởng doanh thu để thực hiện các khuyến nghị nêu trên theo cách ưu tiên, nhằm cải thiện hơn nữa KPI (Các chỉ số hiệu suất chính) hơn nữa.

Để giải quyết vấn đề này, nhóm đưa ra hướng giải quyết dưới đây:

Hướng phát triển 13: Tác động đáng kể của PageValue đến kết quả dự đoán cho thấy rằng khách hàng khi vào trang web đã xem xét rất nhiều những sản phẩm khác nhau cũng như xem các trang sản phẩm liên quan. Do đó, cần phải cải tiến một cách đáng kể cho công cụ đề xuất sản phẩm cũng như đề xuất các gói combo sản phẩm, việc này sẽ mang lại tỷ lệ chuyển đổi cao hơn. Bên cạnh đó, việc cung cấp thêm nhiều sản phẩm giới hạn với số lượng ít sẽ giúp khai thác hiệu

ứng đuôi dài (Long tail effect) trong thương mại điện tử, đồng thời cũng sẽ mang lại nhiều động lực doanh thu hơn.

CHƯƠNG 5: TỔNG KẾT

Tìm hiểu đặc điểm nổi bật của khách hàng mua hàng và khách hàng không mua hàng, sắp xếp các hướng phát triển theo trình tự ưu tiên. Đồng thời trình bày những thuận lợi, khó khăn và cách khắc phục của đồ án.

1. KẾT LUẬN

1.1. Đặc điểm của từng loại khách hàng có quyết định mua hàng và không mua hàng trên các trang thương mại điện tử

Từ việc trực quan hóa dữ liệu cho thấy giá trị trung bình (mean) của mỗi thuộc tính có thể xác định đặc điểm của 2 loại khách hàng là sẽ mua hàng và không mua hàng:

- Khách hàng mua hàng (Revenue = True):
  - Số trang Administrative trong một phiên là khoảng 3.
  - Thời lượng dành cho những trang Administrative là khoảng 93.
  - Số trang Informational trong 1 phiên là khoảng từ 0 đến 1.
  - Thời lượng dành cho những trang Informational là khoảng 30.5.
  - Số trang Product Related trong 1 phiên là hơn 40.
  - Thời lượng dành cho những trang ProductRelated 1531.
  - Page value trung bình là 18.9.
  - Khoảng cách đến Special Day là 0.
- Khách hàng không mua hàng (Revenue = False):
  - Số trang Administrative trong một phiên là 2.
  - Thời lượng dành cho những trang Administrative là 58.4.
  - Số trang Informational trong 1 phiên là khoảng từ 0 đến 1.
  - Thời lượng dành cho những trang Informational là khoảng 15.9.
  - Số trang Product Related trong 1 phiên là gần 27.
  - Thời lượng dành cho những trang ProductRelated 960.9.

- Page value trung bình là 1.5.
- Khoảng cách đến Special Day là 0.1.

1.2. Đánh giá hướng phát triển

Sau khi thực nghiệm và phân tích trên 4 mô hình khác nhau (Naive Bayes, Decision Tree, KNN, Random Forest), nhóm đã tìm ra được mô hình phù hợp nhất trong việc dự đoán ý định mua sắm online của khách hàng đó là Random Forest.

Từ việc phân tích insight dữ liệu, mô hình cây quyết định, nhóm đề xuất thứ tự ưu tiên cho các hướng phát triển như sau:

- *Hướng phát triển 13* - Cải tiến công cụ đề xuất và cung cấp các sản phẩm giới hạn theo chiến lược "cái đuôi dài"
- *Hướng phát triển 1* - Tối ưu hóa trang sản phẩm và phí giao hàng
- *Hướng phát triển 2* - Cá nhân hóa email
- *Hướng phát triển 3* - Chiến lược tỷ lệ thoát với cửa sổ bật lên (pop-ups) được cá nhân hóa
- *Hướng phát triển 4* - Liên tục cập nhật loại sản phẩm bán ra để không bị lỗi thời
- *Hướng phát triển 10* - Thử nghiệm A/B (A/B Testing) dựa trên khu vực và tiếp cận thị trường
- *Hướng phát triển 11* - Lựa chọn các gói phí giao hàng từ các đối tác sao cho phù hợp với từng loại khu vực
- *Hướng phát triển 12* - Tối ưu hóa SEO và quảng cáo qua Mạng xã hội
- *Hướng phát triển 8* - Khuyến mãi và giảm giá theo mùa
- *Hướng phát triển 9* - Đảm bảo hệ thống website vận hành trơn tru và giao diện người dùng thân thiện
- *Hướng phát triển 5* - Khuyến mãi và giảm giá theo cuối tuần/thời gian
- *Hướng phát triển 6* - Giảm giá cho người mới và kết nối khách hàng trung thành
- *Hướng phát triển 7* - Xây dựng và phát triển doanh nghiệp qua các trang mạng xã hội

Qua quá trình thử nghiệm nhiều thuật toán khác nhau, nhóm nhận thấy mỗi loại thuật toán sẽ phù hợp với các bài toán khác nhau, vì vậy việc chọn ra thuật toán phù hợp với bài toán là rất cần thiết.

2. THUẬN LỢI

Việc tìm kiếm bộ dữ liệu phục vụ nghiên cứu trở nên dễ dàng hơn nhờ vào các thư viện data trên internet

Tài liệu kiến thức về các thuật toán, các câu lệnh Python...rất đa dạng giúp việc tìm kiếm trở nên dễ dàng.

Các thành viên trong nhóm có trách nhiệm hoàn thành công việc đúng thời hạn, cũng như luôn có mặt đầy đủ trong các buổi họp giúp cho việc triển khai được liên tục, đúng kế hoạch.

3. KHÓ KHĂN

Việc chọn lựa bộ dữ liệu phù hợp khiến nhóm tốn khá nhiều thời gian, khiến dự án trở nên gấp rút.

Bộ dữ liệu hiện đang phân tích chỉ cung cấp một số thông tin cơ bản, dẫn đến việc đưa ra insight mang tính "nhận định" chung hơn (assumption).

Kết quả của dự án vẫn chưa đạt được hết các mục tiêu đã đề ra trước đó.

Kiến thức phân tích thiên về tính toán khiến nhóm phải dành nhiều thời gian để tìm hiểu vì những kiến thức đó nhóm đã lâu chưa tiếp cận.

4. HƯỚNG PHÁT TRIỂN CHO ĐỒ ÁN

Thu thập thêm các thông tin có ích hơn để phục vụ cho việc phân tích dữ liệu: hiểu sâu được vấn đề hơn, tìm ra được nguyên nhân chính của vấn đề và đưa ra insight chính xác hơn.

Thu thập thêm các dữ liệu như lịch sử duyệt web của từng cá nhân trong một phiên để cải thiện hiệu suất của mô hình.

Có thể xem xét để ứng dụng thuật toán học không giám sát để phân cụm khách hàng.

TRÍCH DẪN TÀI LIỆU THAM KHẢO

[1] Trần Bình Trọng, Session là gì? Tìm hiểu Tổng quan về Session Khái quát Nhất, Tmarketing, < https://bom.so/Dux4UW>, Ngày truy cập: 28/11/2022.

[2] TopDev Blog, Session là gì? Hiểu rõ Session và Cookie, TopDev, <https://bom.so/Kyr2PFH6CkoUX>, Ngày truy cập: 28/11/2022.

[3] ITNavi (2020), Session là gì? Làm thế nào để sử dụng session hiệu quả?, ITNavi, <https://bom.so/bWyA1I>, Ngày truy cập: 28/11/2022.

[4] Tú DA (2015), Phân Biệt Exit Rate Và Bounce Rate, Đầu Tư SEO, < https://bom.so/dLwe6Y>

[5] Tối ưu Google Ads (2019), PHÂN BIỆT CƠ BẢN GIỮA BOUNCE RATE & EXIT RATE, NOVAON AUTOADS, < https://bom.so/k4Ztzf>, Ngày truy cập: 28/11/2022.

[6] Tú DA (2015), Page Value – Giá Trị Trang Web, Đầu Tư SEO, < https://bom.so/ec7rcr>, Ngày truy cập: 28/11/2022.

[7] Bá An (2021), Traffic là gì? 7 cách tăng lượt traffic đột phá cho website, TOPONSEEK, <https://bom.so/eee1ZL>, Ngày truy cập: 28/11/2022.

[8] Canh Minh Do (2020), Tìm hiểu về định lý Bayes và ứng dụng, Canh's Log Book, <https://bom.so/ncLpe7>, Ngày truy cập: 28/11/2022.

[9] Trí tuệ nhân tạo (2019), Phần 2: Phân loại Naive Bayes (Coding), Trí tuệ nhân tạo, <https://bom.so/TCRAsG>, Ngày truy cập: 28/11/2022.

[10] Phan Thị Phượng (2019), *Nghiên cứu thuật toán K – nearest neighbor và sử dụng iris flowers dataset đánh giá hiệu quả thuật toán,* < https://bom.so/wOE05U>, Ngày truy cập: 28/11/2022.

[11] Nguyen Thi Hop (2019), KNN (K-Nearest Neighbors) #1, Viblo, < https://bom.so/Z0VH4R>, Ngày truy cập: 28/11/2022.

[12] Trí tuệ nhân tạo (2019), Cây Quyết Định (Decision Tree), Trí tuệ nhân tạo, <https://bom.so/I4mn5u >, Ngày truy cập: 28/11/2022.

[13] JavaTpoint, Decision Tree Classification Algorithm, JavaTpoint, <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>, Ngày truy cập: 28/11/2022.

[14] Pham Dinh Khanh (2021), DecisionTree, Deep AI KhanhBlog, < https://bom.so/G5kfvW>, Ngày truy cập: 28/11/2022.
[15] Tuấn Nguyễn (2021), Decision Tree algorithm, Machine Learning Cơ bản, <https://bom.so/H1Jrdx>, Ngày truy cập: 28/11/2022.
[16] Websitehcm, Tìm hiểu về Decision Tree (cây quyết định), w3seo, < https://bom.so/Ew1k6x>, Ngày truy cập: 28/11/2022.
[17] Tuấn Nguyễn (2021), Random Forest algorithm, Machine Learning Cơ bản, <https://bom.so/HOkAxS >, Ngày truy cập: 28/11/2022.
[18] Pham Dinh Khanh (2021), Ý tưởng của mô hình rừng cây, Deep AI KhanhBlog, <https://bom.so/4nDJo2>, Ngày truy cập: 28/11/2022.
[19] Sruthi E R (2021), Understanding Random Forest, Analytics Vidhya, <https://bom.so/XcKLZk>, Ngày truy cập: 28/11/2022.
[20] Achoum's Blog, Machine Learning - Classification - phần 3, vnoi.info, < https://bom.so/noJXn3>, Ngày truy cập: 28/11/2022.
[21] Niklas Donges (2022), Random Forest Classifier: A Complete Guide to How It Works in Machine Learning, builtin, <https://bom.so/bmbVCB>, Ngày truy cập: 28/11/2022.

PHỤ LỤC

STT	Từ ngữ	Giải thích
1	Chiến lược “cái đuôi dài”	Chiến lược Cái đuôi dài là một chiến lược kinh doanh cho phép các công ty thu được lợi nhuận rất lớn bằng cách bán các mặt hàng hiếm có với số lượng ít cho nhiều khách hàng, thay vì chỉ bán các mặt hàng phổ biến với số lượng lớn.
2	Thử nghiệm A/B (A/B Testing)	Thử nghiệm A/B là một phương pháp nghiên cứu trải nghiệm người dùng.[1] Thử nghiệm A/B bao gồm một thử nghiệm ngẫu nhiên với hai biến thể, A và B.[2][3] Nó bao gồm việc áp dụng thử nghiệm giả thuyết thống kê hoặc "thử nghiệm giả thuyết hai mẫu" như được sử dụng trong lĩnh vực thống kê. Thử nghiệm A/B là một cách để so sánh hai phiên bản của một biến, thường bằng cách kiểm tra phản ứng của đối tượng đối với biến thể A so với biến thể B, và xác định xem biến thể nào có hiệu quả hơn.[4]