BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÀ RỊA - VŨNG TÀU
KHOA KỸ THUẬT - CÔNG NGHỆ

BARIA VUNGTAU
UNIVERSITY
CAP SAINT JACQUES

BÁO CÁO MÔN HỌC
ỨNG DỤNG TRÍ TUỆ NHÂN TẠO

Tên đề tài:

SỬ DỤNG THUẬT TOÁN LOGISTIC REGRESSION PHÂN TÍCH DỮ LIỆU

GVHD: *TS. Bùi Thị Thu Trang*

Sinh viên: Lê Hồng Phong

MSV: 19034554

Lớp: DH19CT

Vũng Tàu, 01 tháng 01 năm 2023

LỜI CẢM ƠN

Em xin được gửi lời cảm ơn chân thành tới Cô *TS. Bùi Thị Thu Trang* - Giảng viên môn Ứng dụng trí tuệ nhân tạo trường Đại Bả Rịa Vũng Tàu đã tận tình hướng dẫn, giúp đỡ em tiếp cận với cách tư duy, giải quyết, trình bày biểu đồ qua ngon ngữ Python. Những điều này đã giúp em khắc phục được những hạn chế và tạo điều kiện tốt nhất để hoàn thành bài phân tích dữ liệu "in-vehicle-coupon-recommendation" để kết thúc học phần môn.

Mặc dù em đã cố gắng hoàn thành với sự nỗ lực và khả năng của mình, nhưng chắc chắn vẫn còn nhiều hạn chế và thiếu sót. Em mong nhận được sự cảm thông và góp ý từ cô và các bạn.

Vũng Tàu,02 tháng 1 năm 2023

Sinh Viên

Lê Hồng Phong

MỤC LỤC

DANH MỤC HÌNH ẢNH

LỜI NÓI ĐẦU

"Trí tuệ nhân tạo (Artificial Intelligence – AI) là gì? Từ xa xưa, loài người luôn mơ ước có những cỗ máy hỗ trợ con người làm tất cả mọi thứ theo ý mình. Để được như vậy, những cổ máy đó phải có được tri thức như tri thức của con người, từ những điều đơn giản nhất, đến những điều phức tạp nhất. Cuối thế kỷ 19, đầu thế kỷ 20, những nhà khoa học như Leonardo De Vinci, Blaise Pascal, Albert Einstein, Isaac Newton, Galileo Galilei, Alan Turing… đã bắt đầu nghiên cứu ra các phương pháp, cách thức, cỗ máy… giúp loài người thực hiện giấc mơ đó. Cho đến thời điểm hiện tại, những thành tựu nghiên cứu từ các nhà khoa học trong các lĩnh vực như toán học, khoa học máy tính, vật lý, hóa học, sinh học, … đã được ứng dụng rất nhiều trong xã hội loài người.

AI được ứng dụng trong rất nhiều hoạt động và lĩnh vực khác nhau. Đối với hoạt động nghiên cứu cơ bản trong các lĩnh vực toán học, vật lý lượng tử, sinh học di truyền, hóa học phân tích, AI giúp giải phương trình vi phân, đạo hàm riêng, tính toán mô phỏng quá trình tương tác ở mức lượng tử, mô phỏng tái tạo thành công lỗ hổng đen, tối ưu hóa Gen, xác định các marker cho điều chỉnh Gen, thiết kế thuốc trên Gen, xác định cấu trúc hóa học, đề xuất các kết hợp… Đối với hoạt động nghiên cứu ứng dụng, với các thành tựu trong các lĩnh vực như xã hội, quân sự, kinh tế, giao thông, y tế… AI đã hỗ trợ bác sỹ chẩn đoán bệnh, phân tích hình ảnh y khoa, dự báo dịch bệnh, xem xét tác động chính sách…

Hiện nay rất nhiều công ty, từ công ty nhỏ đến công ty hàng đầu trên thế giới đã áp dụng AI để xác định khách hàng tiềm năng, nhóm nhân viên rời bỏ công ty, phát triển sản phẩm, tối ưu vận chuyển, dự đoán xu thế nhu cầu khách hàng, đề xuất sản phẩm cần thiết cho người dùng… làm công cụ hữu dụng để tăng khả năng kinh doanh, cũng như quản lý và cạnh tranh cho doanh nghiệp của mình.

Vì vậy, bài báo cáo này xin được phân tích về dữ liệu "in-vehicle-coupon-recommendation". Đề tài cuối kì này là kết quả của quá trình tích lũy và vận dụng những kiến thức mà em đã tiếp thu và tìm hiểu được trong quá trình học tập. Trong quá trình thực hiện báo cáo em xin cám ơn cô ***TS. Bùi Thị Thu Trang*** đã định hướng cho em thực hiện và hoàn thành bài báo cáo này. Em xin gửi tới cô những lời cảm ơn chân thành nhất.

Em xin chân thành cảm ơn!

PHẦN I. MÁY HỌC VÀ ỨNG DỤNG

1.1. Công nghệ máy học là gì?

Máy học là môn khoa học nhằm phát triển những thuật toán và mô hình thống kê mà các hệ thống máy tính sử dụng để thực hiện các tác vụ dựa vào khuôn mẫu và suy luận mà không cần hướng dẫn cụ thể. Các hệ thống máy tính sử dụng thuật toán máy học để xử lý khối lượng lớn dữ liệu trong quá khứ và xác định các khuôn mẫu dữ liệu. Việc này cho phép chúng dự đoán kết quả chính xác hơn từ cùng một tập dữ liệu đầu vào cho trước. Ví dụ: các nhà khoa học dữ liệu có thể đào tạo một ứng dụng y tế chẩn đoán ung thư từ ảnh chụp X-quang bằng cách lưu trữ hàng triệu ảnh quét và chẩn đoán tương ứng.

1.2. Ứng dụng của máy học.

Công nghệ máy học được sử dụng trong lĩnh vực khác nhau chẳng hạn như:

1.2.1. Sản xuất

Máy học có thể hỗ trợ bảo trì dự đoán, kiểm soát chất lượng và nghiên cứu đổi mới trong lĩnh vực sản xuất. Công nghệ máy học cũng giúp các công ty cải thiện giải pháp hậu cần, bao gồm quản lý tài sản, chuỗi cung ứng và kho hàng. Ví dụ: gã khổng lồ 3M trong ngành sản xuất sử dụng AWS Machine Learning để cải tiến giấy nhám. Thuật toán máy học giúp các nhà nghiên cứu của 3M phân tích những thay đổi nhỏ về hình dạng, kích thước và định hướng có thể cải thiện khả năng mài mòn và độ bền ra sao. Những gợi ý này cung cấp thông tin cho quá trình sản xuất.

1.2.2. Chăm sóc sức khỏe và khoa học đời sống.

Sự phát triển như vũ bão của cảm biến và thiết bị có thể đeo được đã tạo ra một lượng lớn dữ liệu về sức khỏe. Các chương trình máy học có thể phân tích thông tin này và hỗ trợ bác sĩ chẩn đoán và điều trị trong thời gian thực. Các nhà nghiên cứu máy học đang phát triển giải pháp phát hiện khối u ung thư và chẩn đoán những bệnh về mắt, tác động đáng kể tới kết quả chăm sóc sức khỏe con người. Ví dụ: Cambia Health Solutions sử dụng AWS Machine Learning để hỗ trợ các công ty khởi nghiệp về chăm sóc sức khỏe, giúp họ tự động hóa và điều chỉnh phương pháp điều trị cho phụ nữ mang thai.

1.2.3. Dịch vụ tài chính

Các dự án máy học về tài chính giúp cải thiện khả năng phân tích rủi ro và quy định. Công nghệ máy học có thể giúp các nhà đầu tư xác định cơ hội mới bằng cách phân tích hoạt động của thị trường chứng khoán, đánh giá các quỹ phòng hộ hoặc hiệu chỉnh danh mục tài chính. Thêm vào đó, công nghệ máy học có thể giúp xác định các khách hàng vay nợ có rủi ro cao và giảm bớt dấu hiệu của hành vi lừa đảo. Công ty dẫn đầu trong lĩnh vực phần mềm tài chính Intuit sử dụng hệ thống AWS Machine Learning, Amazon Textract, để thực hiện hoạt động quản lý tài chính được cá nhân hóa tốt hơn và giúp người dùng cuối cải thiện tình hình tài chính của họ.

1.2.4. Bán lẻ

Nhà bán lẻ có thể sử dụng máy học để cải thiện dịch vụ khách hàng, quản lý hàng tồn kho, bán hàng gia tăng và tiếp thị đa kênh. Ví dụ: Amazon Fulfillment (AFT) giảm được 40% chi phí cơ sở hạ tầng bằng cách sử dụng mô hình máy học để xác định hàng tồn đặt sai chỗ. Việc này giúp họ thực hiện lời hứa của Amazon rằng một sản phẩm sẽ luôn được cung cấp cho khách hàng và được giao đúng hẹn, mặc dù công ty phải xử lý hàng triệu chuyến hàng trên toàn cầu mỗi năm.

1.2.5. Truyền thông và giải trí

Các công ty giải trí tìm đến máy học để hiểu rõ hơn đối tượng mục tiêu của họ đồng thời cung cấp nội dung chân thực, được cá nhân hóa và theo nhu cầu của khách hàng. Thuật toán máy học được triển khai để giúp thiết kế trailer và các dạng quảng cáo khác, từ đó đề xuất nội dung được cá nhân hóa cho người tiêu dùng và thậm chí là hợp lý hóa quy trình sản xuất.

Ví dụ: Disney đang sử dụng AWS Deep Learning để lưu trữ thư viện nội dung đa phương tiện của họ. Công cụ AWS Machine Learning tự động gắn thẻ, mô tả và sắp xếp nội dung đa phương tiện, cho phép biên kịch và họa sĩ hoạt hình nhanh chóng tìm kiếm và làm quen với các nhân vật của Disney.

PHẦN II. THUẬT TOÁN PHÂN LOẠI VÀ PHÂN TÍCH DỮ LIỆU

2.1. Thuật toán phân lớp.

Thuật toán phân lớp là quá trình phân lớp 1 đối tượng dữ liệu vào 1 hay nhiều lớp đã cho trước nhờ 1 mô hình phân lớp (model). Một ví dụ dễ hiểu đó là phân lớp email là "spam" hoặc "không phải spam". Có nhiều loại nhiệm vụ phân lớp khác nhau mà chúng ta có thể gặp phải trong học máy và các phương pháp tiếp cận chuyên biệt để lập mô hình có thể được sử dụng cho từng loại.

2.2. Thuật toán để phân tích dữ liệu.

Các thuật toán phổ biến có thể được sử dụng để phân tích dữ liệu được sử dụng phổ biến như:

- Logistic Regression (Hồi quy logistic):

Thuật toán được mượn từ lĩnh vực thống kê và cũng là phương thức tốt nhất dành cho các vấn đề phân loại nhị phân. Logistic Regression sử dụng một hàm không tuyến tính gọi là hàm Logistic. Hàm này giống như một lớp S lớn và có thể biến đổi bất cứ giá trị nào thành 0-1. Khi được loại bỏ thuộc tính không liên quan tới đầu ra hoặc tương tự nhau, hồi quy Logic hoạt động tốt hơn.

- k-Nearest Neighbors (KNN):

thuật toán đơn giản và hiệu quả với mô hình đại diện là toàn bộ dữ liệu tập huấn. Bạn có thể thực hiện dự đoán cho một điểm dữ liệu mới bằng cách tìm kiếm thông qua toàn bộ tập đào tạo. Nó được ứng dụng cho hầu hết các ví được K giống nhau và tóm tắt biến đầu ra cho các ví dụ K đó. Kỹ thuật đơn giản nhất để xác định sự giống nhau giữa các trường hợp dữ liệu là sử dụng Euclide (trong trường hợp thuộc tính cùng kích cỡ).

- Decision Tree (Cây quyết định):

thuật toán hỗ trợ đắc lực cho việc ra quyết định của các kỹ sư với mô hình dạng cây. Khi nhìn vào Decision Tree, người dùng có thể đưa ra những lựa chọn đúng đắn hơn. Mặc dù là một mô hình cũ nhưng Decision Tree vẫn là một sự lựa chọn tốt dành cho newbie. Dưới góc độ là một người làm chủ dự án, Decision Tree là danh sách tối ưu các phương án lựa chọn.

- Linear Regression (Hồi quy tuyến tính):

Là những thuật toán nổi tiếng nhất hiện nay và được dùng nhiều trong thống kê cũng như Machine Learning. Việc biểu diễn hồi quy tuyến tính là một phương trình mô tả đường thẳng phù hợp nhất với mối quan hệ giữa các biến đầu vào X và biến đầu ra Y. Trong đó có một số giải pháp như đại số tuyến tình dành cho Ordinary least square và tối ưu hoá Gradient descent. Quy tắc sử dụng kỹ thuật này là loại bỏ các biến tương tự nhau và các yếu tố xao lãng từ dữ liệu của người dùng.

- Support Vector Machine:

Mỗi hyperplane là một đường phân chia không gian biến đầu vào. Mỗi hyperplane được chọn sẽ phân tách tốt nhất các điểm ở trong không gian của các biến đầu vào hoặc lớp 0 và lớp 1. Support Vector Machines được coi là một trong những phương pháp phân loại hàng đầu để phân tích tập dữ liệu.

- Naive Bayes:

thuật toán đơn giản nhưng có mô hình tiên đoán cực mạnh mẽ. Nó bao gồm hai loại xác suất có thể được tính trực tiếp từ dữ liệu như xác suất của mỗi lớp và xác suất

có điều kiện cho mỗi lớp với mỗi giá trị X. Sau khi tính, mô hình có thể đưa ra dự đoán cho dữ liệu mới bằng định lý Bayes. Naive Bayes giả định mỗi biến đầu vào là độc lập và mạnh mẽ nhưng không thực tế với dữ liệu thực.

2.3. Dữ liệu.

Link dữ liệu:
https://archive.ics.uci.edu/ml/datasets/invehicle+coupon+recommendation

2.3.1. Mô tả dữ liệu và yêu cầu

Dữ liệu này được thu thập thông qua một cuộc khảo sát trên Amazon Mechanical Turk. Cuộc khảo sát mô tả các tình huống lái xe khác nhau bao gồm điểm đến, thời gian hiện tại, thời tiết, hành khách, v.v., sau đó hỏi người đó xem anh ta có chấp nhận phiếu giảm giá nếu anh ta là tài xế hay không. Để biết thêm thông tin về tập dữ liệu, vui lòng tham khảo bài báo: Wang, Tong, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl và Perry MacNeille. 'Một khuôn khổ bayes cho các bộ quy tắc học tập để phân loại có thể giải thích được.' Tạp chí Nghiên cứu Học máy 18, số 1 (2017): 2357-2393.

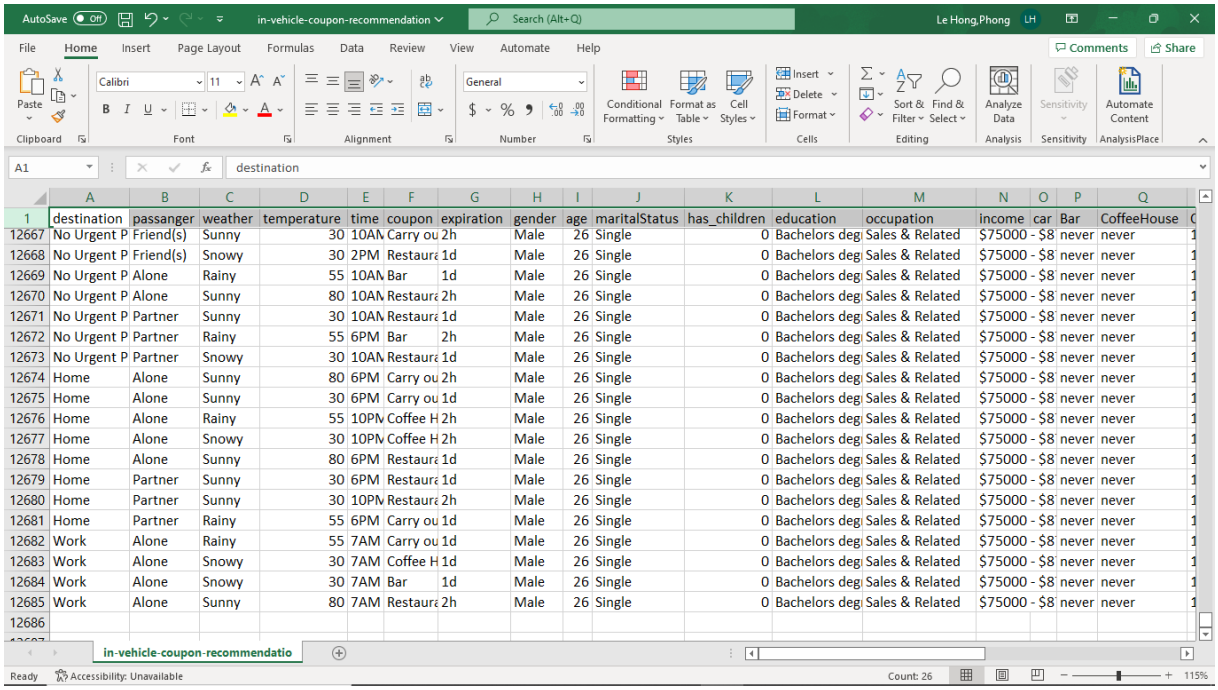
2.3.2. Thông tin các thuộc tính.

STT	Tên thuộc tính	Tên thuộc tính (vi)	Dữ liệu
1	destination	Điểm đến	No Urgent Place, Home, Work
2	passanger	Người đi cùng.	Alone, Friend(s), Kid(s), Partner (who are the passengers in the car)
3	weather	Thời tiết.	Sunny, Rainy, Snowy
4	temperature	Nhiệt độ.	55, 80, 30
5	time	Thời gian.	2PM, 10AM, 6PM, 7AM, 10PM
6	coupon	Phiếu mua hang.	Restaurant(<$20), Coffee House, Carry out & Take away, Bar, Restaurant($20-$50)
7	expiration	Hết hạn.	1d, 2h (the coupon expires in 1 day or in 2 hours)
8	gender	Giới tính.	Female, Male
9	age	Tuổi.	21, 46, 26, 31, 41, 50plus, 36, below21
10	maritalStatus	Tình trạng hôn nhân.	Unmarried partner, Single, Married partner, Divorced, Widowed
11	has_Children	Đã có con.	1, 0
12	education	Học vấn.	Some college - no degree, Bachelor’s degree, Associates

			degree, High School Graduate, Graduate degree (Masters or Doctorate), Some High School
13	occupation	Nghề nghiệp.	Unemployed, Architecture & Engineering, Student, Education&Training&Library, Healthcare Support, Healthcare Practitioners & Technical, Sales & Related, Management, Arts Design Entertainment Sports & Media, Computer & Mathematical, Life Physical Social Science, Personal Care & Service, Community & Social Services, Office & Administrative Support, Construction & Extraction, Legal, Retired, Installation Maintenance & Repair, Transportation & Material Moving, Business & Financial, Protective Service, Food Preparation & Serving Related, Production Occupations, Building & Grounds Cleaning & Maintenance, Farming Fishing & Forestry
14	income	Thu nhập.	$37500 - $49999, $62500 - $74999, $12500 - $24999, $75000 - $87499, $50000 - $62499, $25000 - $37499, $100000 or More, $87500 - $99999, Less than $12500
15	Car	Xe ô tô.	Mazda5, Scooter and motorcycle, crossover, do not drive

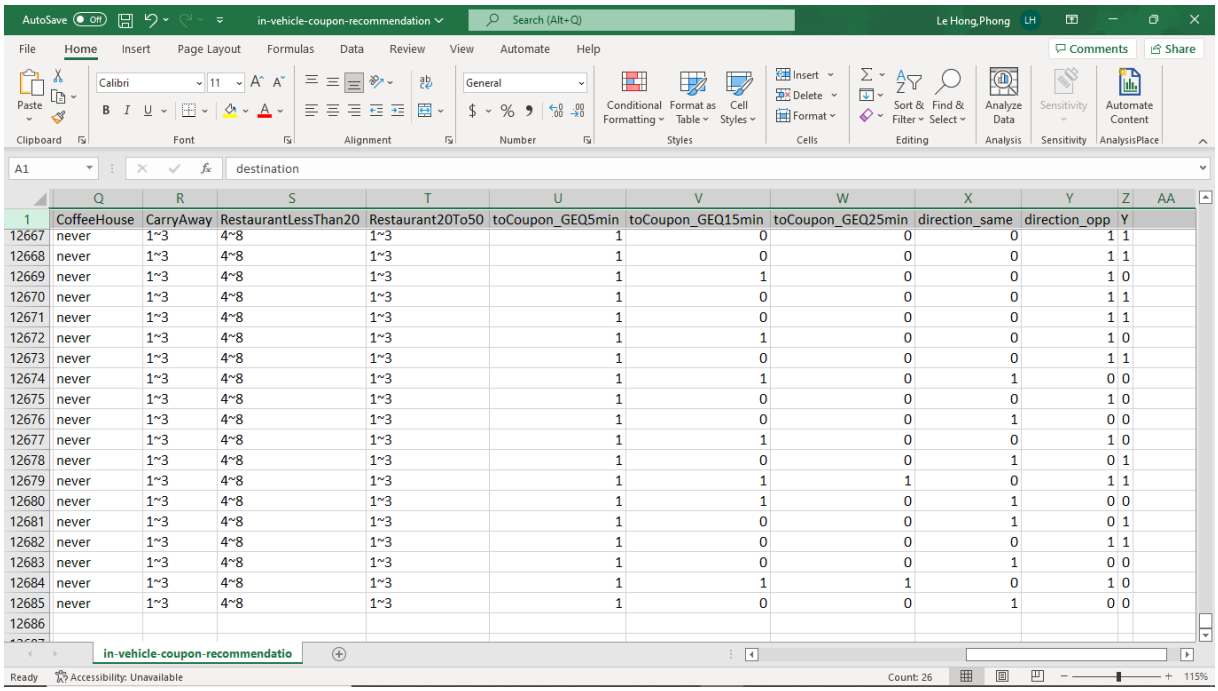
16	Bar	Số lần đến một quán bar bao nhiêu lần mỗi tháng.	never, less1, 1~3, gt8, nan4~8
17	CoffeeHouse	Số lần đến quán cà phê mỗi tháng.	never, less1, 4~8, 1~3, gt8, nan
18	CarryAway	Nhận được đồ ăn mang đi bao nhiêu lần mỗi tháng.	n4~8, 1~3, gt8, less1, never
19	RestaurantLessThan20	Đến nhà hàng bao nhiêu lần với chi phí trung bình cho mỗi người dưới 20 đô la mỗi tháng.	4~8, 1~3, less1, gt8, never
20	Restaurant20To50	Đến nhà hàng bao nhiêu lần với chi phí trung bình cho mỗi người là $20 - $50 mỗi tháng.	1~3, less1, never, gt8, 4~8, nan
21	toCoupon_GEQ15min	Khoảng cách lái xe đến nhà hàng/quán bar để sử dụng phiếu giảm giá lớn hơn 15 phút.	0,1
22	toCoupon_GEQ25min	Khoảng cách lái xe đến nhà hàng/quán bar để sử dụng phiếu giảm giá lớn hơn 25 phút.	0,1
23	direction_same	Nhà hàng/quán bar có cùng hướng với điểm đến hiện tại không.	0,1
24	direction_opp	Nhà hàng/quán bar có ngược hướng với điểm đến hiện tại không.	1,0
25	Y	Phiếu giảm giá có được chấp nhận hay không.	0,1

File dữ liệu CSV



	destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	has_children	education	occupation	income	car	Bar	CoffeeHouse
12667	No Urgent P	Friend(s)	Sunny	30	10AN	Carry ou	2h	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12668	No Urgent P	Friend(s)	Snowy	30	2PM	Restaura	1d	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12669	No Urgent P	Alone	Rainy	55	10AN	Bar	1d	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12670	No Urgent P	Alone	Sunny	80	10AN	Restaura	2h	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12671	No Urgent P	Partner	Sunny	30	10AN	Restaura	1d	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12672	No Urgent P	Partner	Rainy	55	6PM	Bar	2h	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12673	No Urgent P	Partner	Snowy	30	10AN	Restaura	1d	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12674	Home	Alone	Sunny	80	6PM	Carry ou	2h	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12675	Home	Alone	Sunny	30	6PM	Carry ou	1d	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12676	Home	Alone	Rainy	55	10PN	Coffee H	2h	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12677	Home	Alone	Snowy	30	10PN	Coffee H	2h	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12678	Home	Alone	Sunny	80	6PM	Restaura	1d	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12679	Home	Partner	Sunny	30	6PM	Restaura	1d	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12680	Home	Partner	Sunny	30	10PN	Restaura	2h	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12681	Home	Partner	Rainy	55	6PM	Carry ou	1d	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12682	Work	Alone	Rainy	55	7AM	Carry ou	1d	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12683	Work	Alone	Snowy	30	7AM	Coffee H	1d	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12684	Work	Alone	Snowy	30	7AM	Bar	1d	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12685	Work	Alone	Sunny	80	7AM	Restaura	2h	Male	26	Single	0	Bachelors deg	Sales & Related	$75000 - $8	never	never	
12686																	

Hình 1. File dữ liệu CSV



	CoffeeHouse	CarryAway	RestaurantLessThan20	Restaurant20To50	toCoupon_GEQ5min	toCoupon_GEQ15min	toCoupon_GEQ25min	direction_same	direction_opp	Y
12667	never	1~3	4~8	1~3	1	0	0	0	1	1
12668	never	1~3	4~8	1~3	1	0	0	0	1	1
12669	never	1~3	4~8	1~3	1	1	0	0	1	0
12670	never	1~3	4~8	1~3	1	0	0	0	1	1
12671	never	1~3	4~8	1~3	1	0	0	0	1	1
12672	never	1~3	4~8	1~3	1	1	0	0	1	0
12673	never	1~3	4~8	1~3	1	0	0	0	1	1
12674	never	1~3	4~8	1~3	1	1	0	1	0	0
12675	never	1~3	4~8	1~3	1	0	0	0	1	0
12676	never	1~3	4~8	1~3	1	0	0	1	0	0
12677	never	1~3	4~8	1~3	1	1	0	0	1	0
12678	never	1~3	4~8	1~3	1	0	0	1	0	1
12679	never	1~3	4~8	1~3	1	1	1	0	1	1
12680	never	1~3	4~8	1~3	1	1	0	1	0	0
12681	never	1~3	4~8	1~3	1	0	0	1	0	1
12682	never	1~3	4~8	1~3	1	0	0	0	1	1
12683	never	1~3	4~8	1~3	1	0	0	1	0	0
12684	never	1~3	4~8	1~3	1	1	1	0	1	0
12685	never	1~3	4~8	1~3	1	0	0	1	0	0
12686										

Hình 2. File dữ liệu CSV

Qua file CSV cho ta thấy file có “12684 dữ liệu đầu vào, và có 24 thuộc tính”

2.4. Phân tích dữ liệu

```
#import thư viện
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import seaborn as sns
```

✓ 0.4s

Hình 3. Import các thư viện cần dùng

Stt	Tên thư viện	Mục đích sử dụng
1	numpy	Xử lý dữ liệu dưới dạng mảng
2	pandas	Xử lý dữ liệu dưới dạng bảng
3	matplotlib	Vẽ biểu đồ phân tán
4	sklearn	Xây dựng thuật toán dự đoán qua các model ở phía trên
5	seaborn	Thư viện mở rộng của Matplotlib

Đọc file dữ liệu và hiển thị tên các cột.

```
#đọc dl từ file csv
data = pd.read_csv('in-vehicle-coupon-recommendation.csv')

#liệt kê danh sách các cột
print(list(data.columns))
```

✓ 0.2s

Hình 4.Đọc file dữ liệu

Kết quả in ra

```
['destination', 'passanger', 'weather', 'temperature', 'time', 'coupon', 'expiration',
'gender', 'age', 'maritalStatus', 'has_children', 'education', 'occupation', 'income',
'car', 'Bar', 'CoffeeHouse', 'CarryAway', 'RestaurantLessThan20', 'Restaurant20To50',
'toCoupon_GEQ5min', 'toCoupon_GEQ15min', 'toCoupon_GEQ25min', 'direction_same',
'direction_opp', 'Y']
```

Hiển thị dữ liệu 7 dòng đầu tiên ra màn hình

```
#Hiển thị dữ liệu
data.head(7)
```

✓ 0.1s Python

	destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	...	RestaurantLessThan20_1~3	RestaurantLessThan20_4~8
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Female	21	Unmarried partner	...	0	1
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Female	21	Unmarried partner	...	0	1
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Female	21	Unmarried partner	...	0	1
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Female	21	Unmarried partner	...	0	1
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Female	21	Unmarried partner	...	0	1
5	No Urgent Place	Friend(s)	Sunny	80	6PM	Restaurant(<20)	2h	Female	21	Unmarried partner	...	0	1
6	No Urgent Place	Friend(s)	Sunny	55	2PM	Carry out & Take away	1d	Female	21	Unmarried partner	...	0	1

7 rows × 133 columns

Hình 5. dữ liệu in ra màn hình

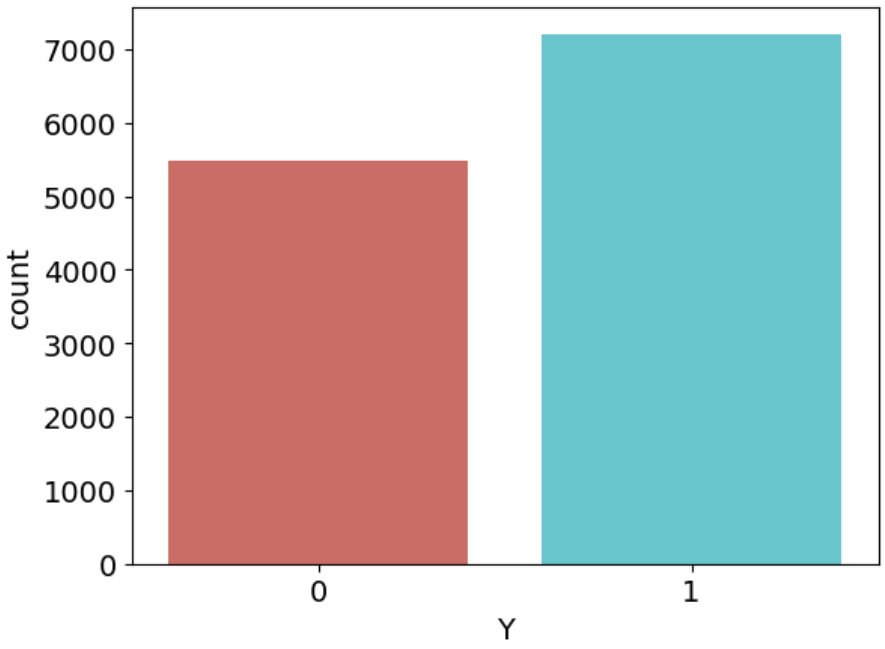
```
#hiển thị số lượng 0 là phiếu giảm giá không được chấp nhận, 1 là được chấp nhận
#kết quả: 7210 được chấp nhận phiếu giảm giá, 5474 là không được chấp nhận
data['Y'].value_counts()
```

✓ 0.1s

```
1    7210
0    5474
Name: Y, dtype: int64
```

```
#Biểu đồ cột sử dụng thư viện seaborn để hiển thị,
# cho ta thấy sự chênh lệch sự chênh lệch về việc liệu phiếu giảm giá có được chấp nhận hay không
sns.countplot(x='Y', data=data, palette='hls')
plt.show()
```

✓ 0.3s



Hình 6. Biểu đồ dạng cột thể hiện đồng ý và không đồng ý phiếu giảm giá

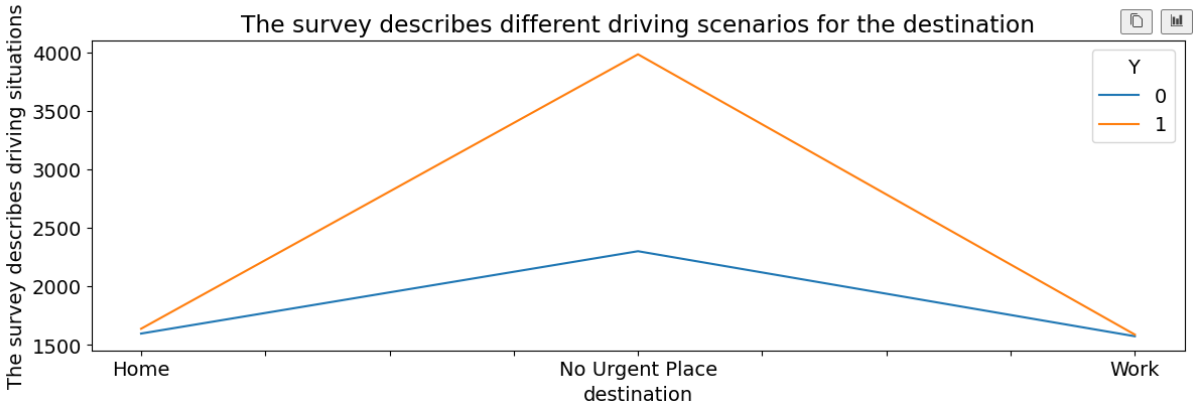
In biểu đồ để coi quan hệ hay sự ảnh hưởng của mỗi cột feature đến target

Vì đường line khi in ra sẽ không được rõ nét, nên sẽ tiến hành sử lý bài qua dạng biểu đồ cột để in ra chất lượng hơn

```
pd.crosstab(data['destination'], data['Y']).plot(kind='line', figsize=(14, 4))
plt.title('The survey describes different driving scenarios for the destination')
plt.xlabel('destination')
plt.ylabel('The survey describes driving situations')
print("##### driving situation to the destination and asking if he accepts the coupon #####")
```

✓ 1.8s

```
##### driving situation to the destination and asking if he accepts the coupon #####
```



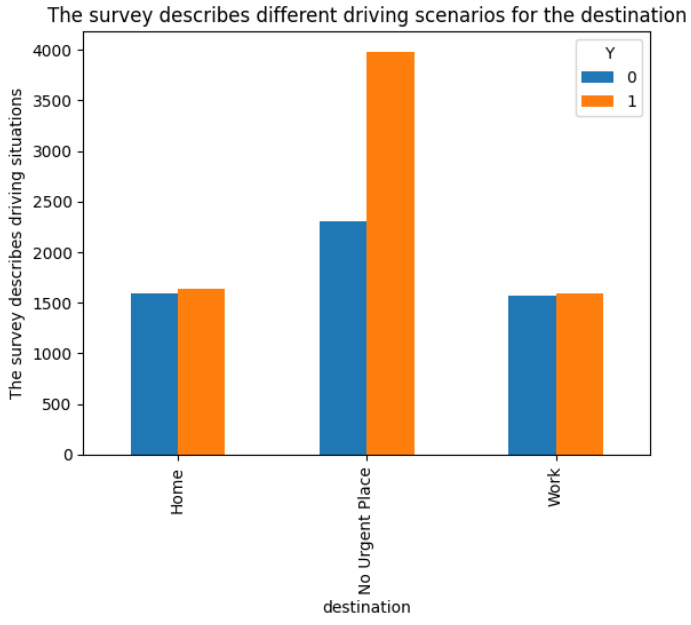
Hình 7. Biểu đồ dạng line giữa nơi đến và Y

Biểu đồ dạng cột giữa nơi đến với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['destination'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the destination')
plt.xlabel('destination')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=10)
print("##### driving situation to the destination and asking if he accepts the coupon #####")
```

✓ 1.3s

```
##### driving situation to the destination and asking if he accepts the coupon #####
```



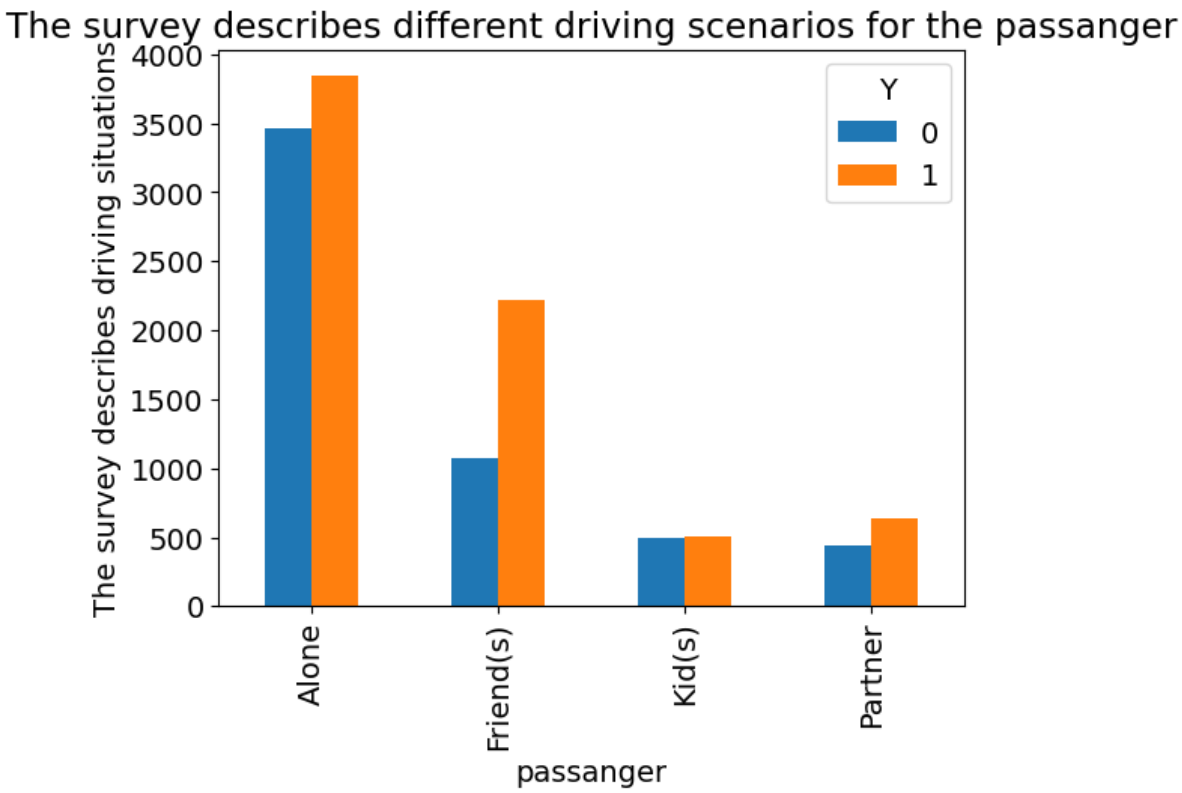
Hình 8. Biểu đồ cột giữa nơi đến và Y

giữa hành khách với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['passanger'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the passanger')
plt.xlabel('passanger')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the passanger and asking if he accepts the coupon #####")
```

✓ 1.1s

```
##### driving situation to the passanger and asking if he accepts the coupon #####
```



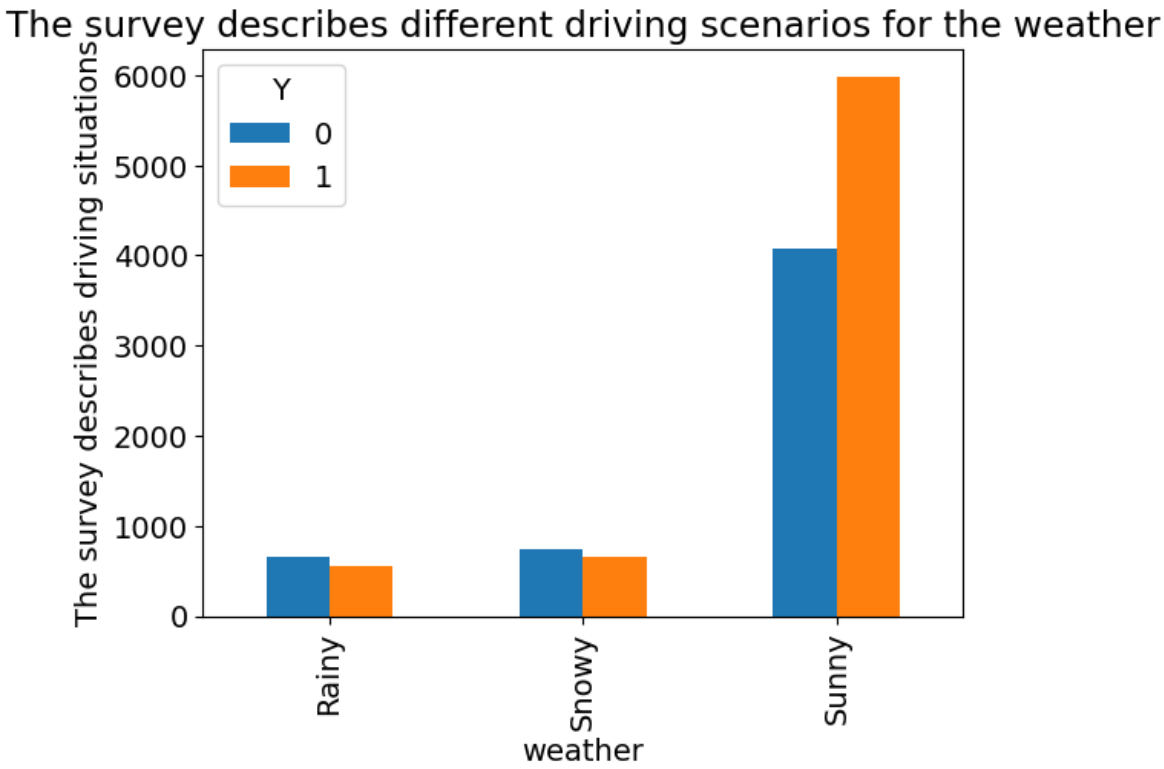
Hình 9. Biểu đồ cột giữa hành khách và Y

Giữa thời tiết với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['weather'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the weather')
plt.xlabel('weather')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the weather and asking if he accepts the coupon #####")
```

✓ 1.1s

```
##### driving situation to the weather and asking if he accepts the coupon #####
```



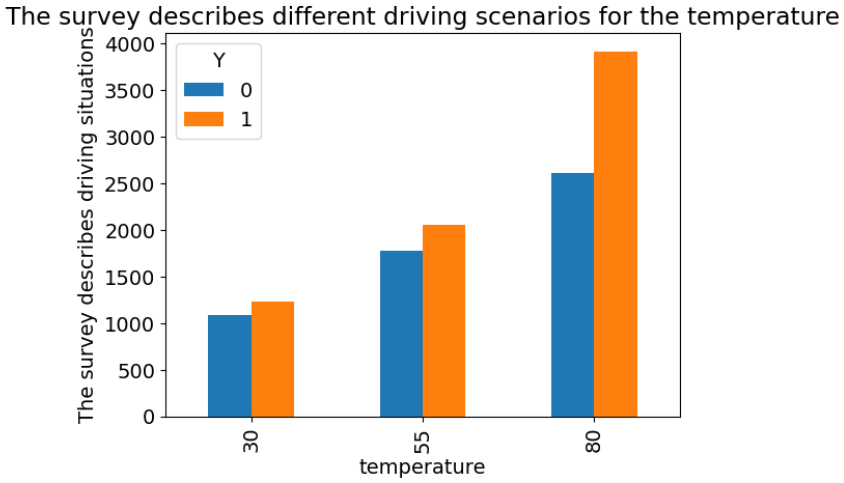
Hình 10. Biểu đồ cột giữa thời tiết và Y

Giữa Nhiệt độ với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['temperature'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the temperature')
plt.xlabel('temperature')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the temperature and asking if he accepts the coupon #####")
```

✓ 1.7s

```
##### driving situation to the temperature and asking if he accepts the coupon #####
```



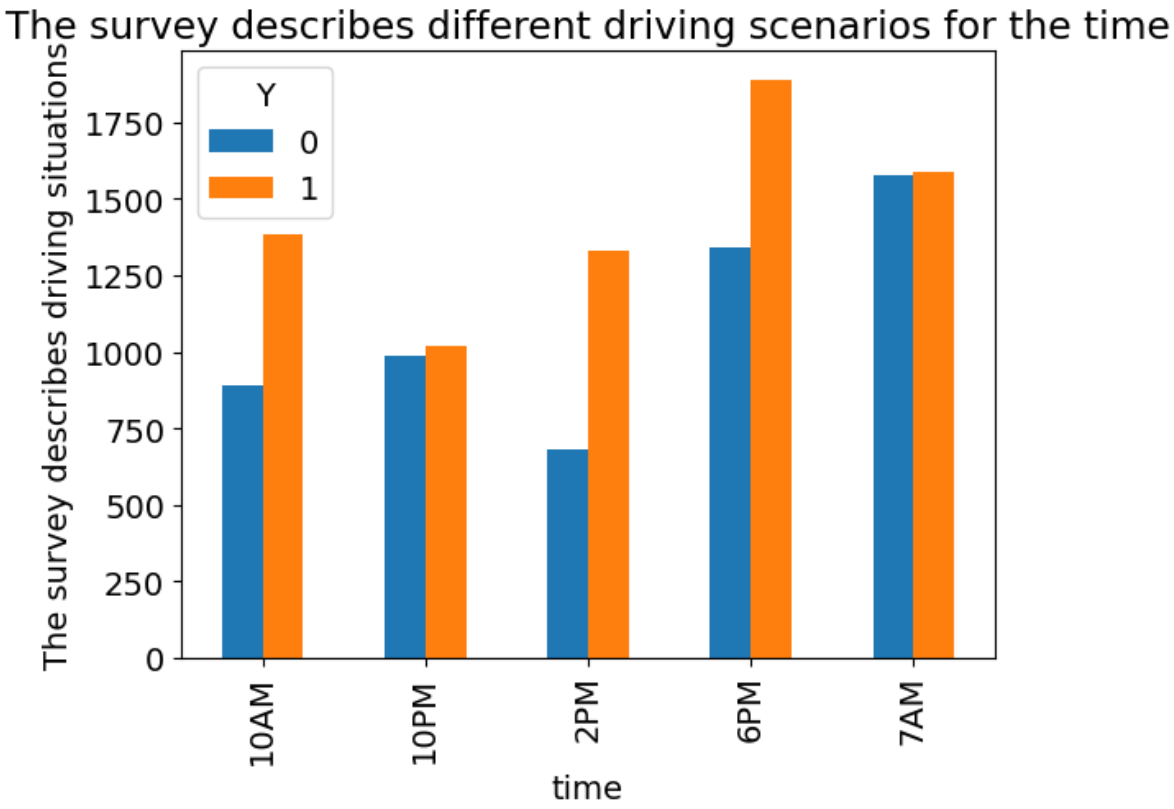
Hình 11. Biểu đồ cột giữa nhiệt độ và Y

Giữa thời gian với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['time'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the time')
plt.xlabel('time')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the time and asking if he accepts the coupon #####")
```

✓ 1.4s

```
##### driving situation to the time and asking if he accepts the coupon #####
```

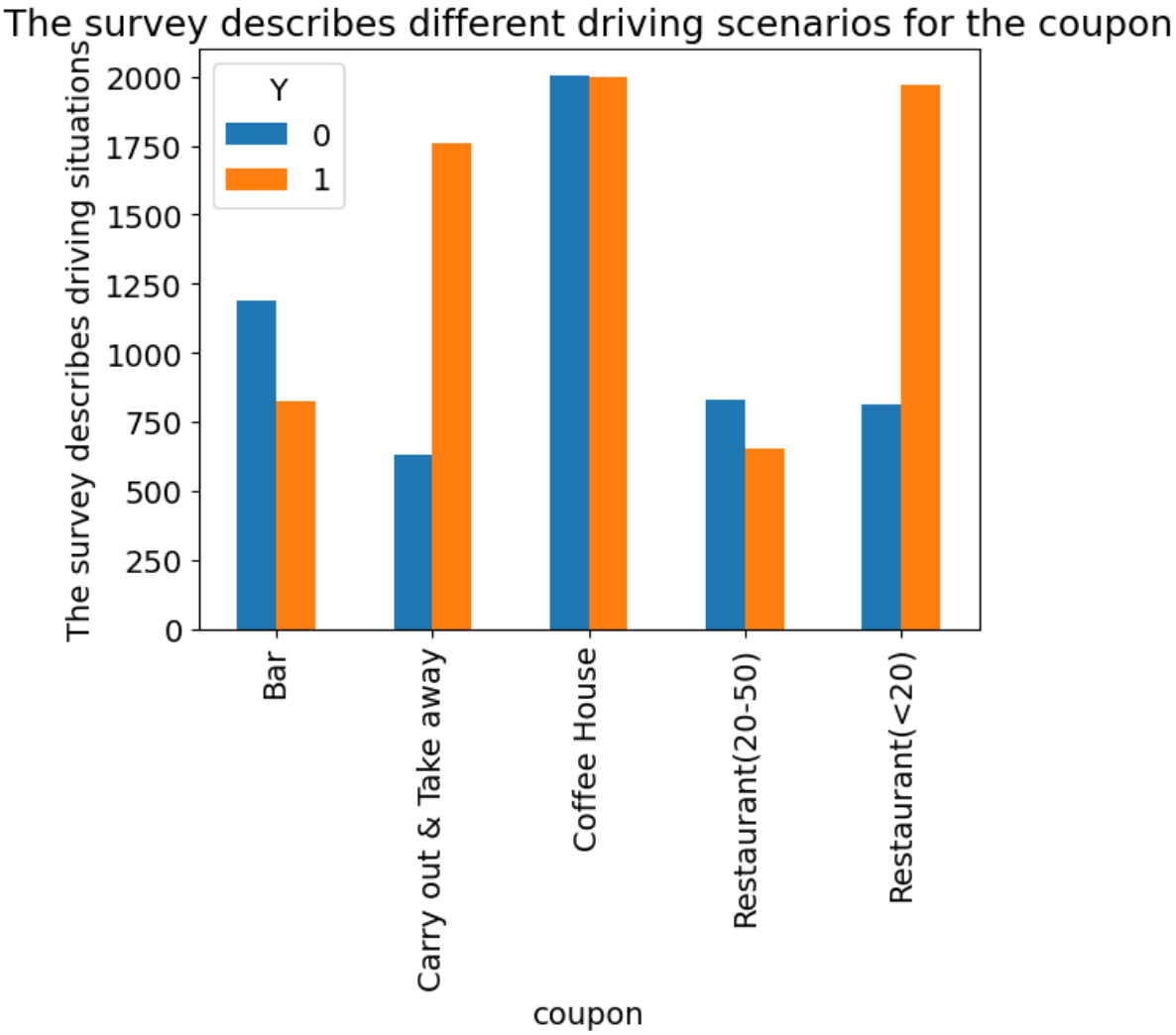


Hình 12. Biểu đồ cột giữa thời gian và Y

Giữa Phiếu mua hàng với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['coupon'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the coupon')
plt.xlabel('coupon')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=13)
print("##### driving situation to the coupon and asking if he accepts the coupon #####")
```

✓ 2.1s



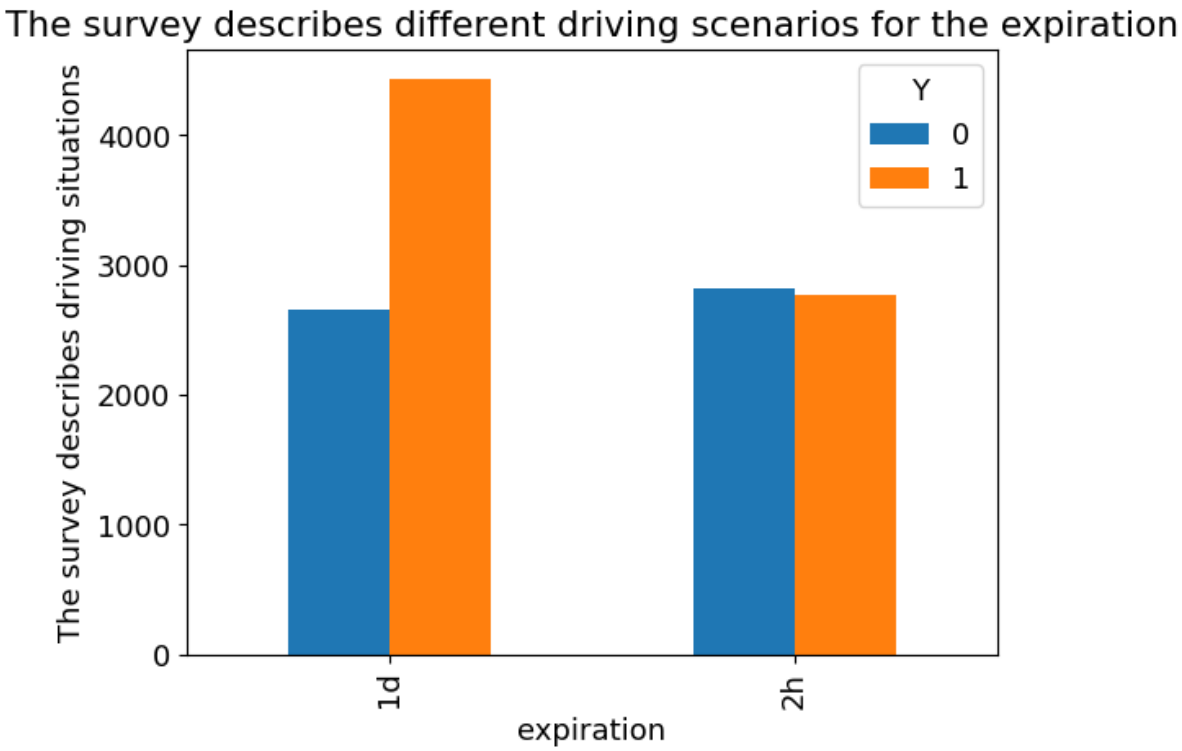
Hình 13. Biểu đồ cột giữa phiếu mua hàng và Y

Giữa Phiếu sắp hết hạn với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['expiration'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the expiration')
plt.xlabel('expiration')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the expiration and asking if he accepts the coupon #####")
```

✓ 1.4s

```
##### driving situation to the expiration and asking if he accepts the coupon #####
```



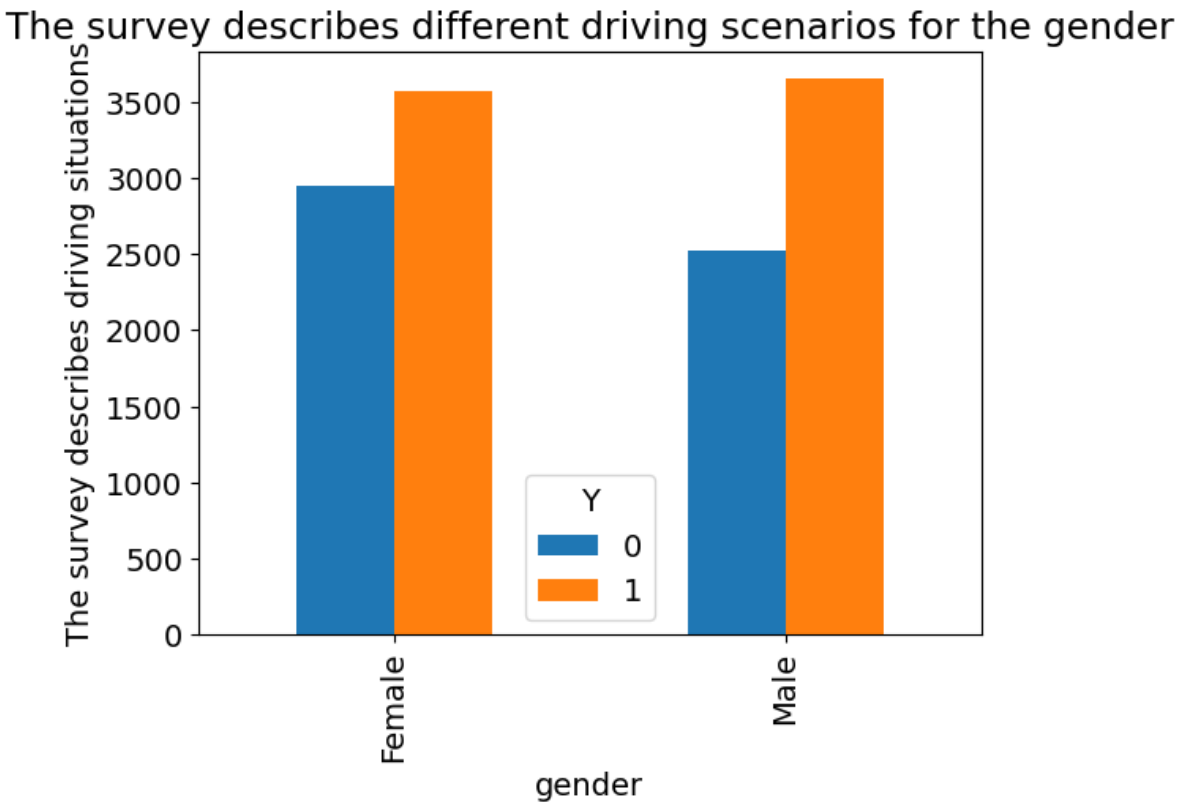
Hình 14. Biểu đồ cột giữa phiếu sắp hết hạn và Y

Giữa Giới tính với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['gender'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the gender')
plt.xlabel('gender')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the gender and asking if he accepts the coupon #####")
```

✓ 1.3s

```
##### driving situation to the gender and asking if he accepts the coupon #####
```



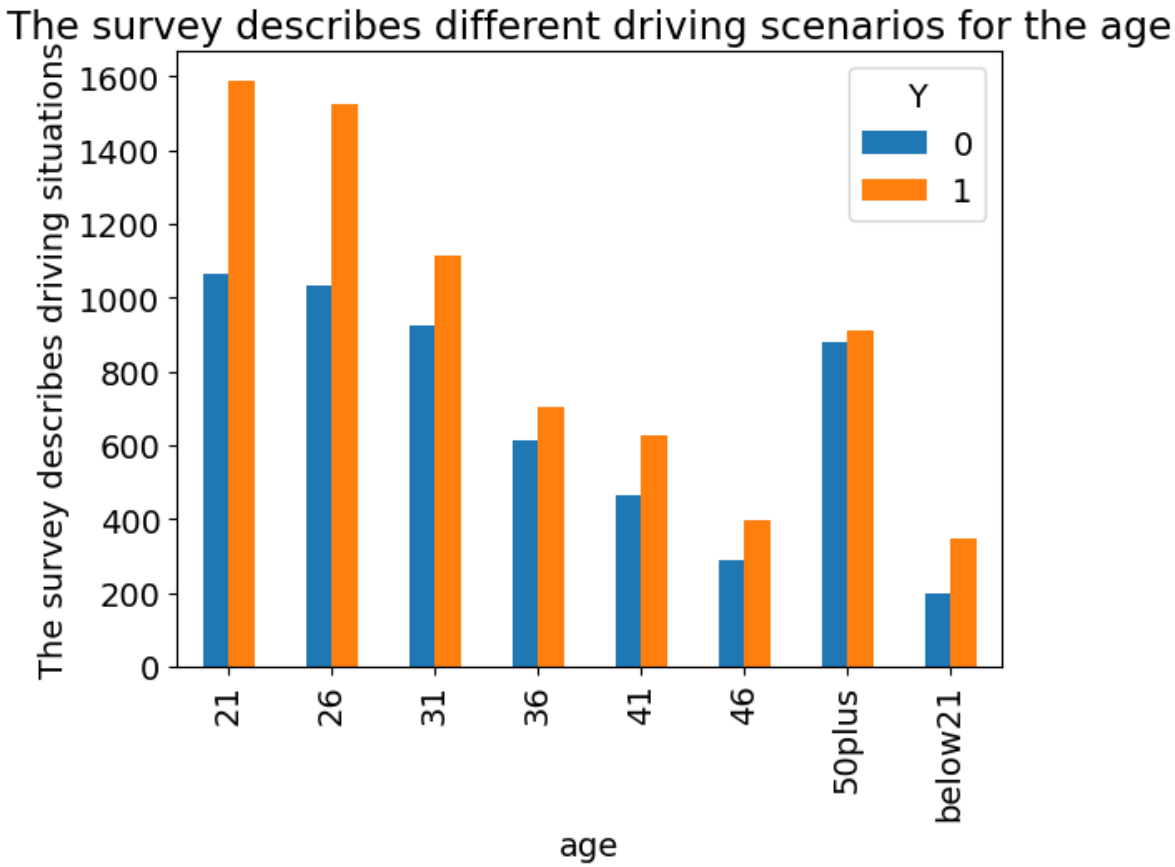
Hình 15. Biểu đồ cột giữa giới tính và Y

Giữa độ tuổi với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['age'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the age')
plt.xlabel('age')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the age and asking if he accepts the coupon #####")
```

✓ 1.2s

```
##### driving situation to the age and asking if he accepts the coupon #####
```

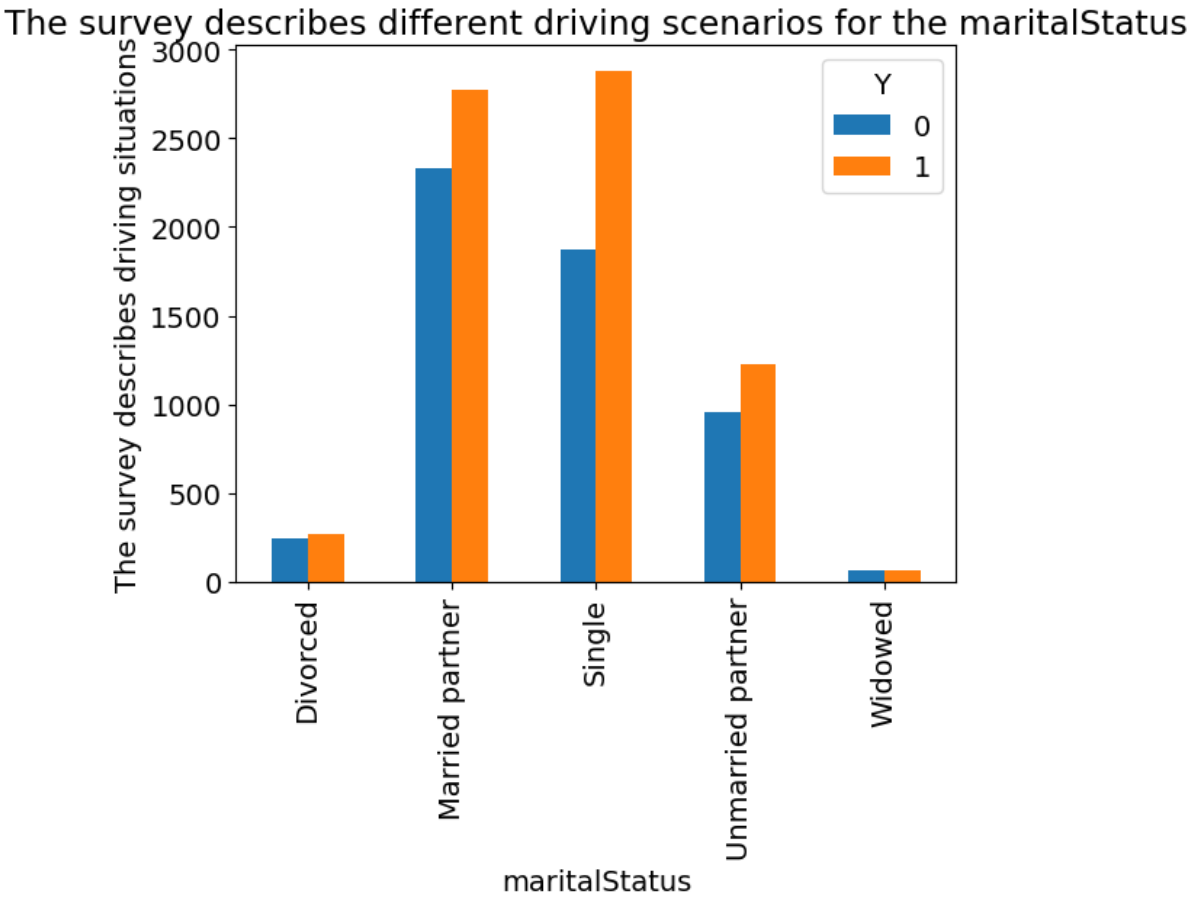


Hình 16. Biểu đồ cột giữa độ tuổi và Y

Giữa Tình trạng hôn nhân với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['maritalStatus'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the maritalStatus')
plt.xlabel('maritalStatus')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the maritalStatus and asking if he accepts the coupon #####")
```

✓ 1.1s



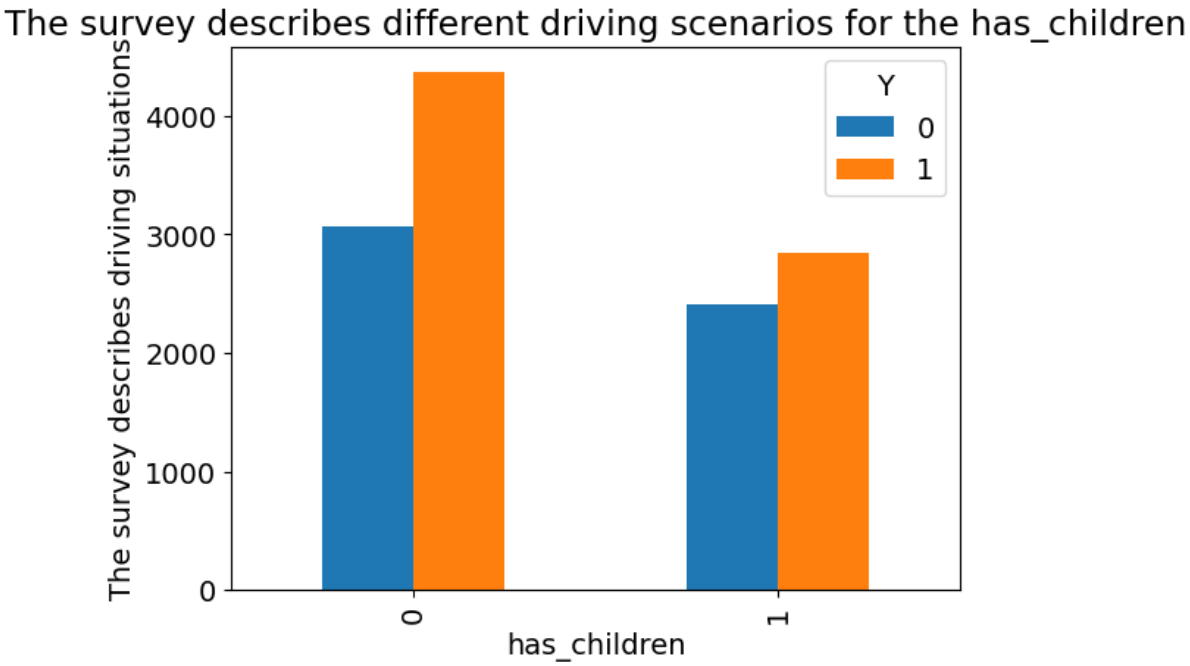
Hình 17. Biểu đồ cột giữa tình trạng hôn nhân và Y

Giữa có con với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['has_children'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the has_children')
plt.xlabel('has_children')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the has_children and asking if he accepts the coupon #####")
```

✓ 1.1s

```
##### driving situation to the has_children and asking if he accepts the coupon #####
```



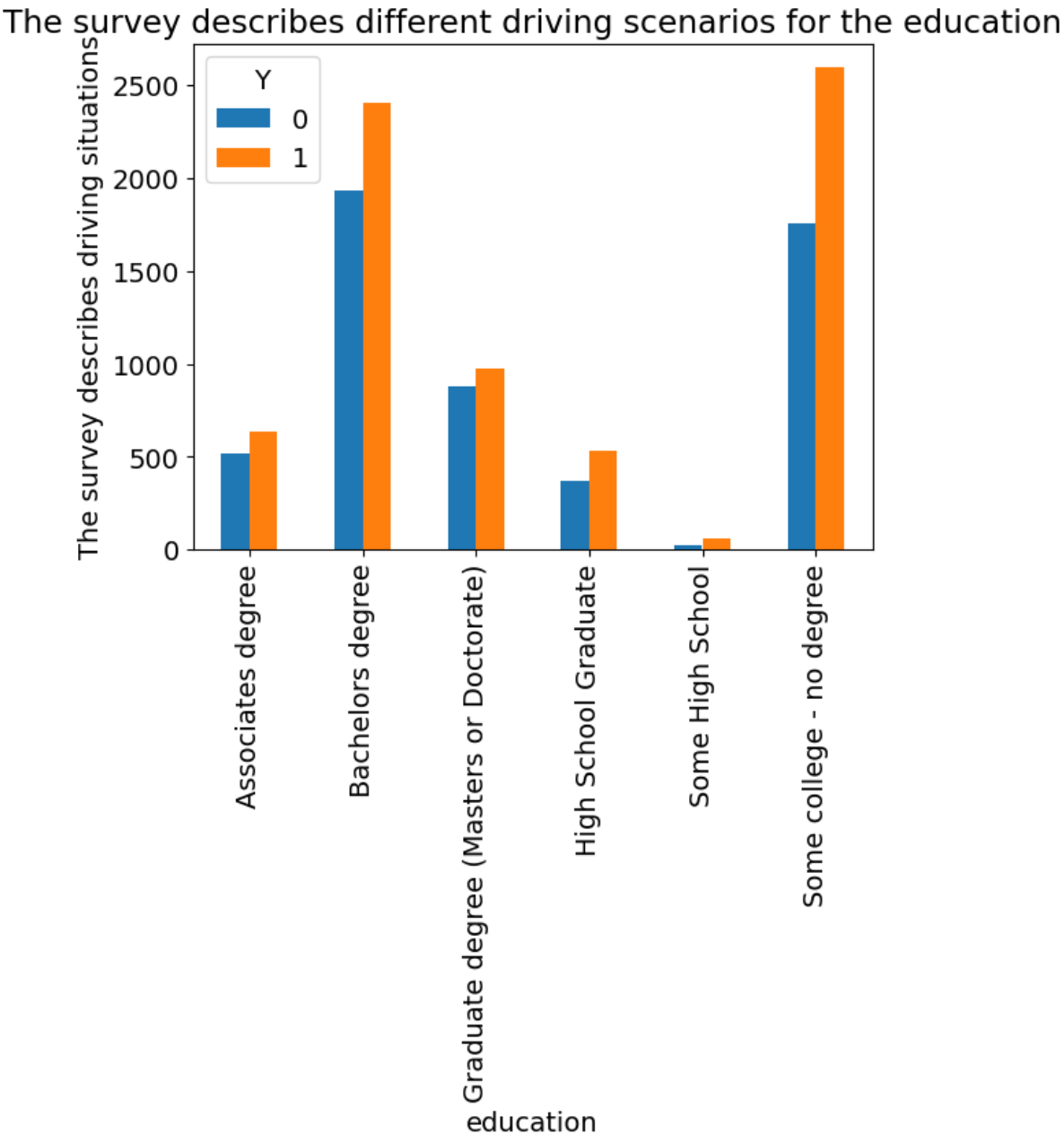
Hình 18. Biểu đồ cột giữa có con và Y

Giữa Học vấn với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['education'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the education')
plt.xlabel('education')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the education and asking if he accepts the coupon #####")
```

✓ 1.1s

```
##### driving situation to the education and asking if he accepts the coupon #####
```



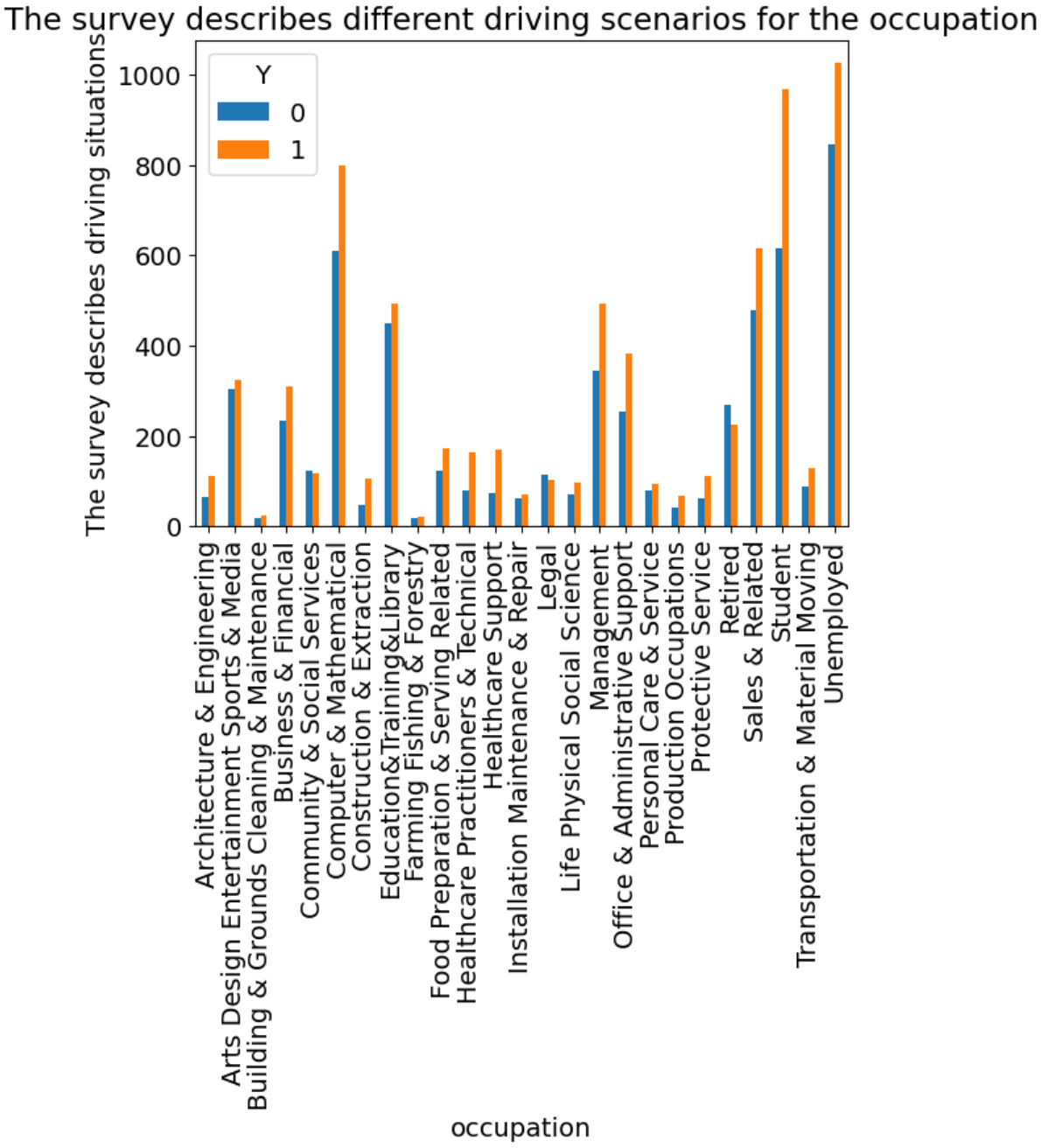
Hình 19. Biểu đồ cột giữa học vấn và Y

Giữa nghề nghiệp với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['occupation'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the occupation')
plt.xlabel('occupation')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the occupation and asking if he accepts the coupon #####")
```

✓ 3.4s

```
##### driving situation to the occupation and asking if he accepts the coupon #####
```



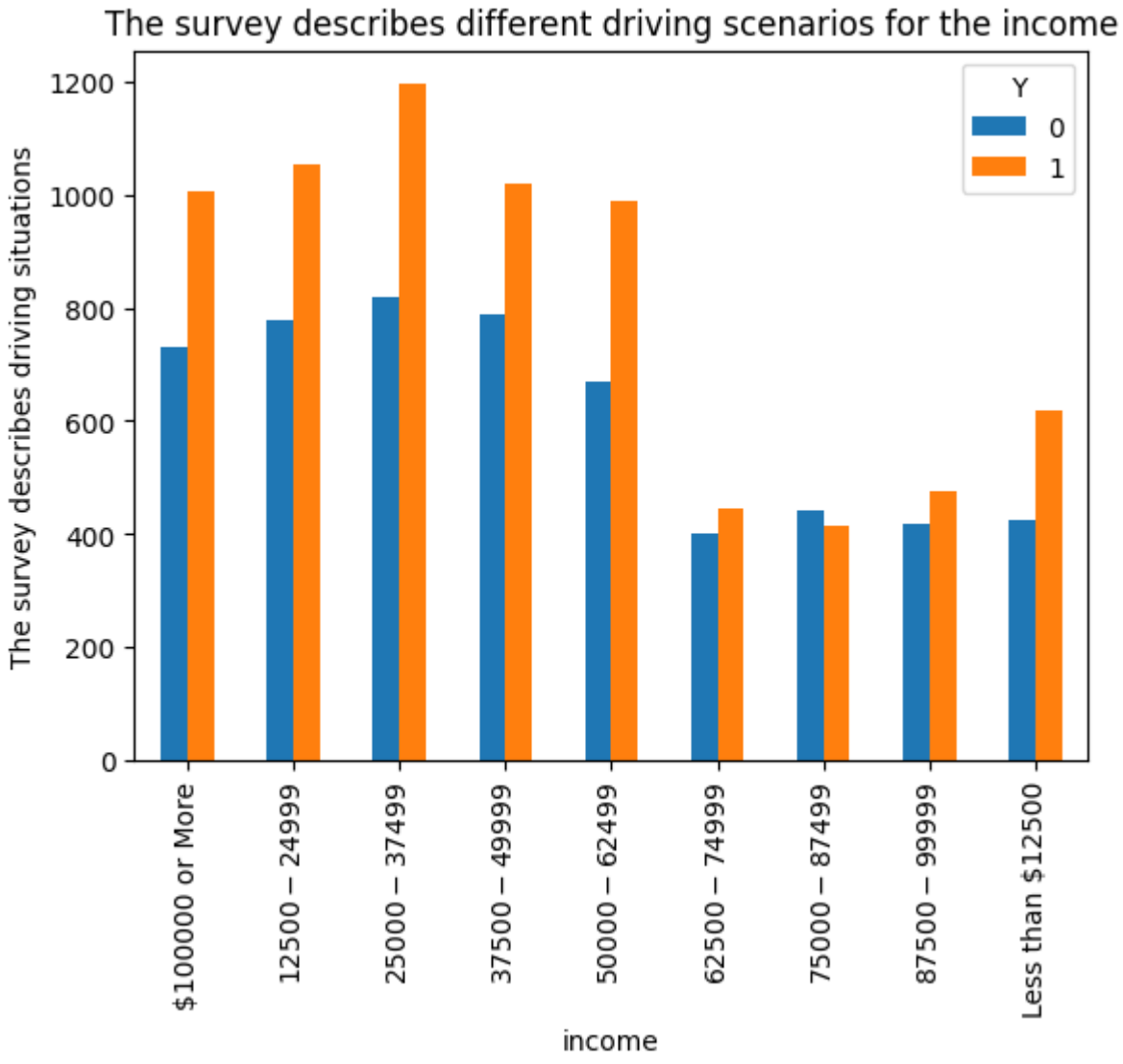
Hình 20. Biểu đồ cột giữa nghề nghiệp và Y

Giữa Thu nhập với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['income'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the income')
plt.xlabel('income')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the income and asking if he accepts the coupon #####")
```

✓ 2.1s

```
##### driving situation to the income and asking if he accepts the coupon #####
```



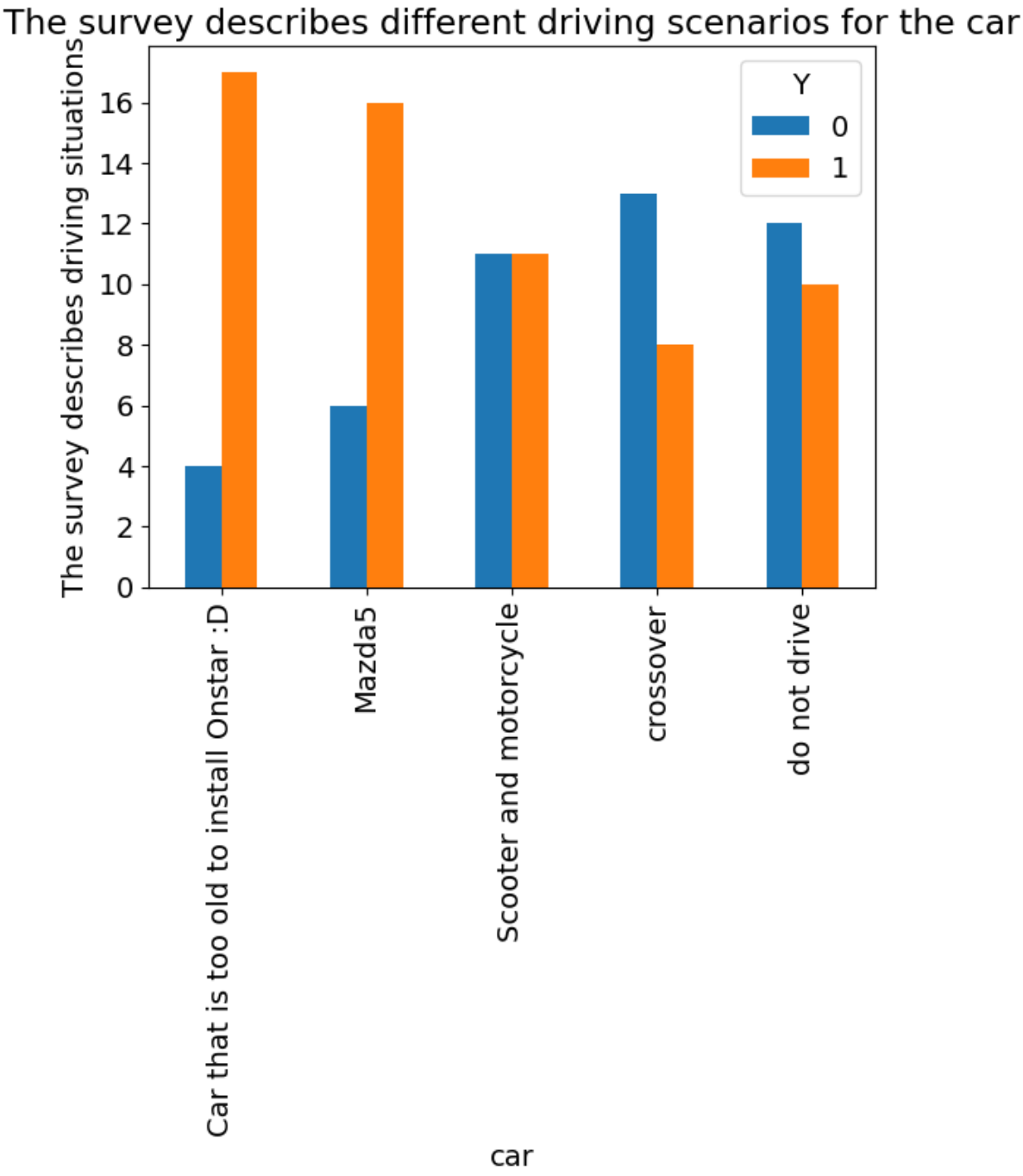
Hình 21. Biểu đồ cột giữa thu nhập và Y

Giữa Xe ô tô với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['car'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the car')
plt.xlabel('car')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the car and asking if he accepts the coupon #####")
```

✓ 1.1s

```
##### driving situation to the car and asking if he accepts the coupon #####
```



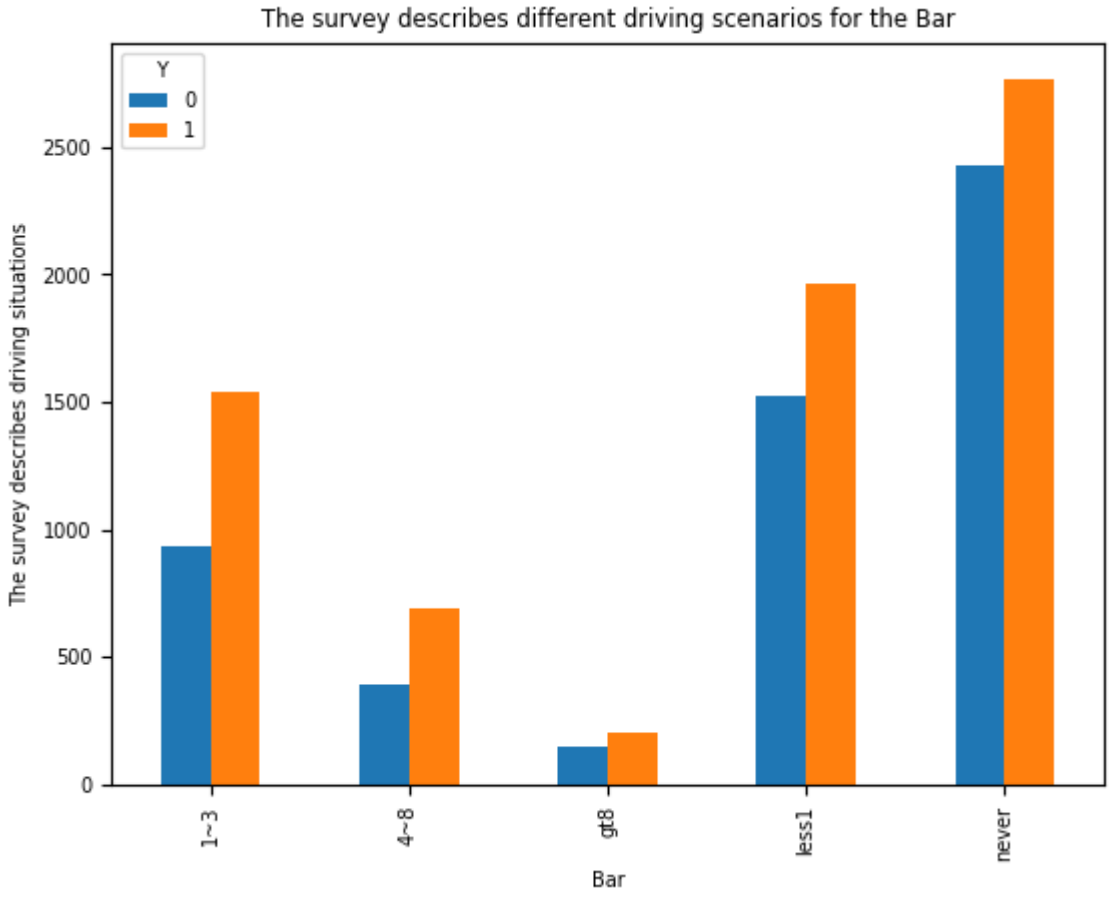
Hình 22. Biểu đồ cột giữa Xe Ô Tô và Y

Giữa Số lần đến một quán bar bao nhiêu lần mỗi tháng với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['Bar'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the Bar')
plt.xlabel('Bar')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the Bar and asking if he accepts the coupon #####")
```

✓ 1.1s

```
##### driving situation to the Bar and asking if he accepts the coupon #####
```



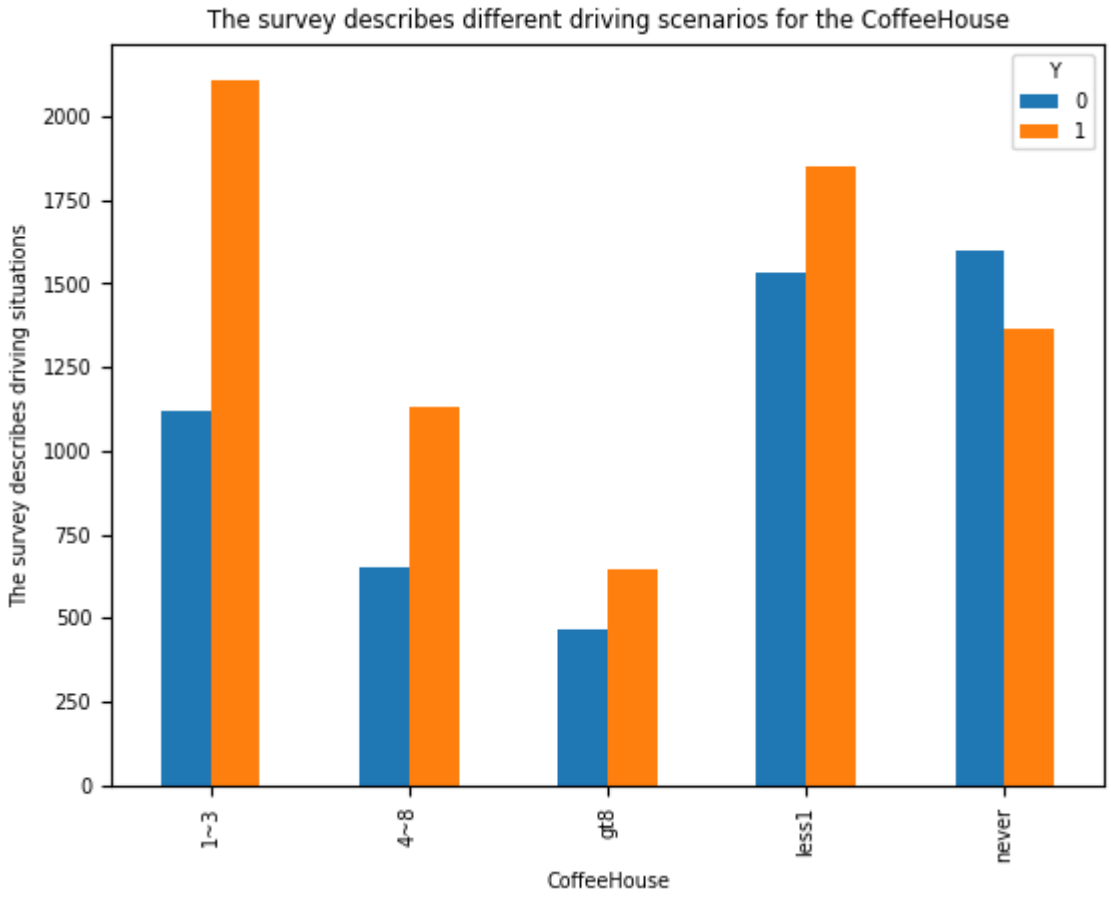
Hình 23. Biểu đồ cột giữa số lần đến một quán bar mỗi tháng và Y

Giữa Số lần đến quán cà phê mỗi thángvới Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['CoffeeHouse'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the CoffeeHouse')
plt.xlabel('CoffeeHouse')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=8)
print("##### driving situation to the CoffeeHouse and asking if he accepts the coupon #####")
```

✓ 1.6s

```
##### driving situation to the CoffeeHouse and asking if he accepts the coupon #####
```



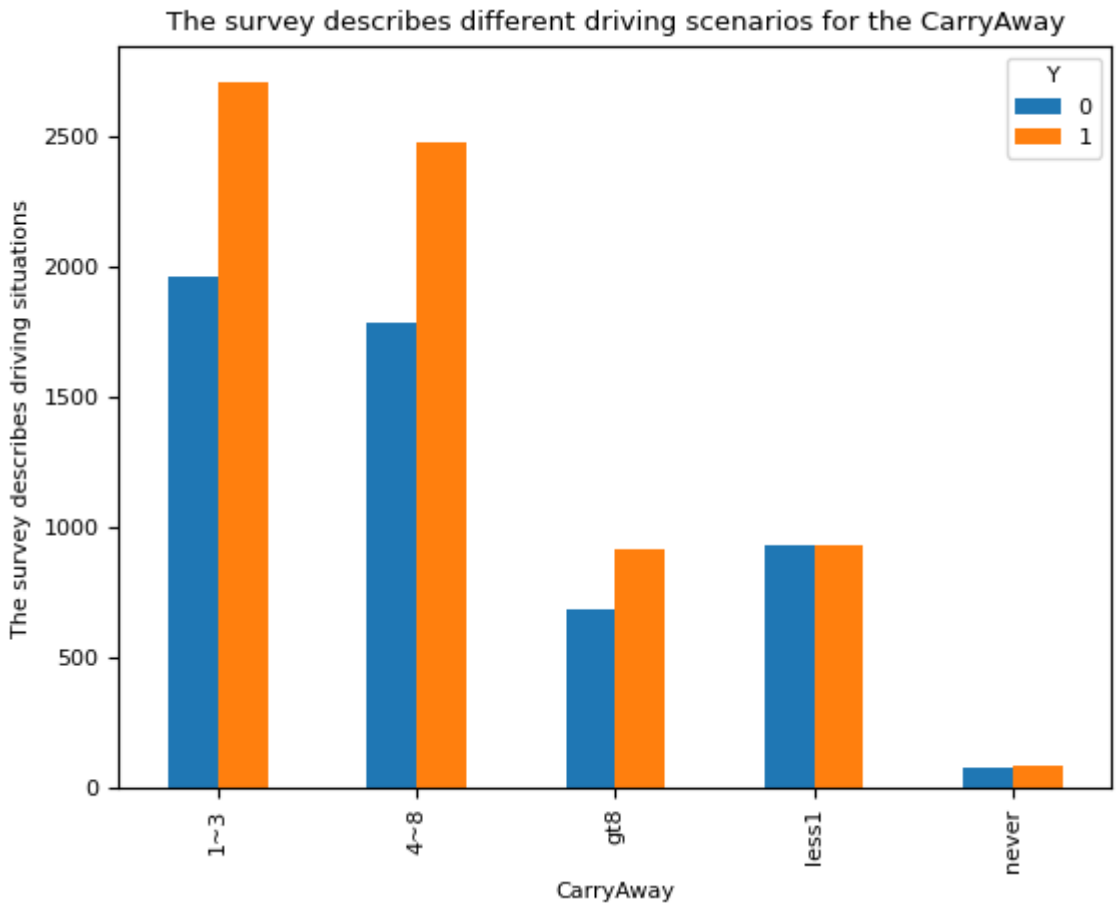
Hình 24. Biểu đồ cột giữa số lần đến cafe mỗi tháng và Y

Giữa Nhận được đồ ăn mang đi bao nhiêu lần mỗi tháng với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['CarryAway'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the CarryAway')
plt.xlabel('CarryAway')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=8)
print("##### driving situation to the CarryAway and asking if he accepts the coupon #####")
```

✓ 1.4s

```
##### driving situation to the CarryAway and asking if he accepts the coupon #####
```



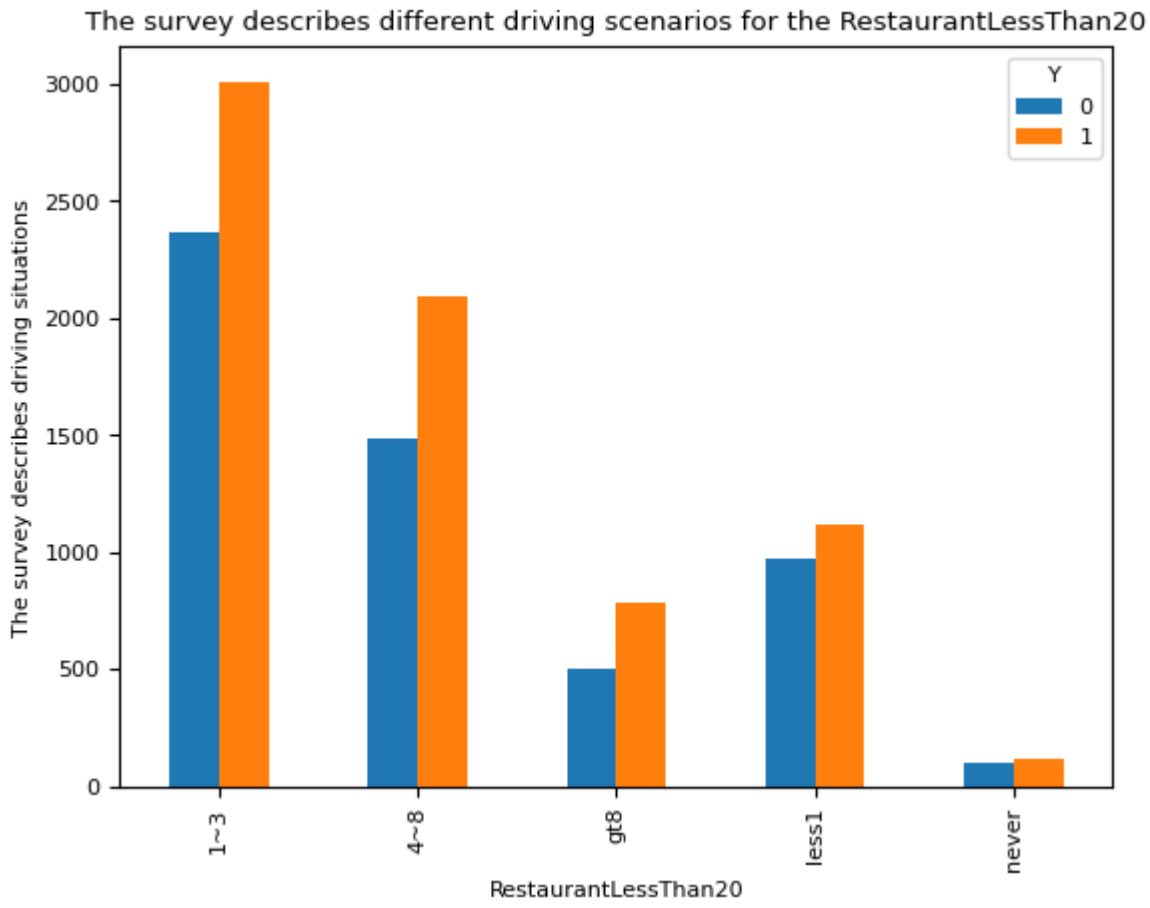
Hình 25. Biểu đồ cột giữa nhận đồ ăn mang đi và Y

Giữa Đến nhà hàng bao nhiêu lần với chi phí trung bình cho mỗi người dưới 20 đô la mỗi tháng với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['RestaurantLessThan20'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the RestaurantLessThan20')
plt.xlabel('RestaurantLessThan20')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=14)
print("##### driving situation to the RestaurantLessThan20 and asking if he accepts the coupon #####")
```

✓ 2.1s

```
##### driving situation to the RestaurantLessThan20 and asking if he accepts the coupon #####
```



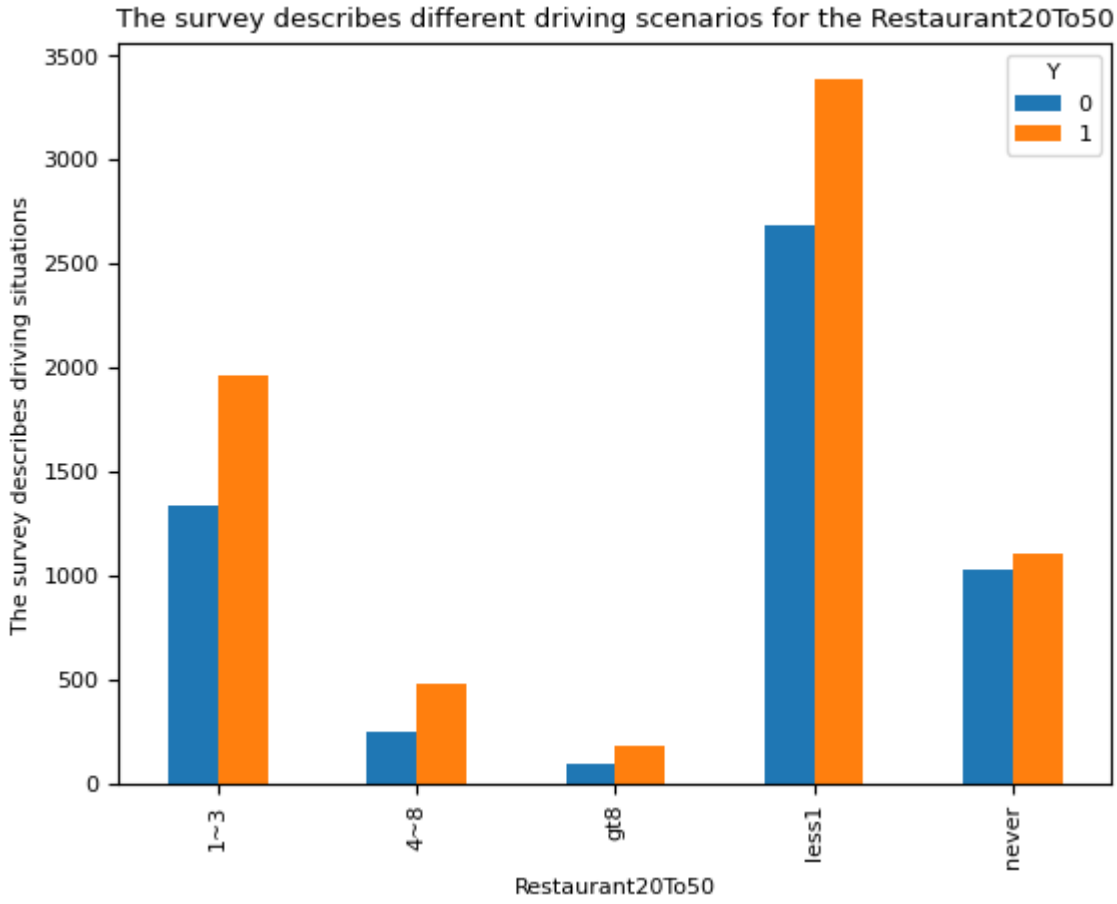
Hình 26. Biểu đồ cột giữa đến nhà hàng bao nhiêu lần và chi phí TB là 20$ 1 người và Y

Giữa Đến nhà hàng bao nhiêu lần với chi phí trung bình cho mỗi người là $20 - $50 mỗi tháng với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['Restaurant20To50'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the Restaurant20To50')
plt.xlabel('Restaurant20To50')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=8)
print("##### driving situation to the Restaurant20To50 and asking if he accepts the coupon #####")
```

✓ 1.4s

```
##### driving situation to the Restaurant20To50 and asking if he accepts the coupon #####
```



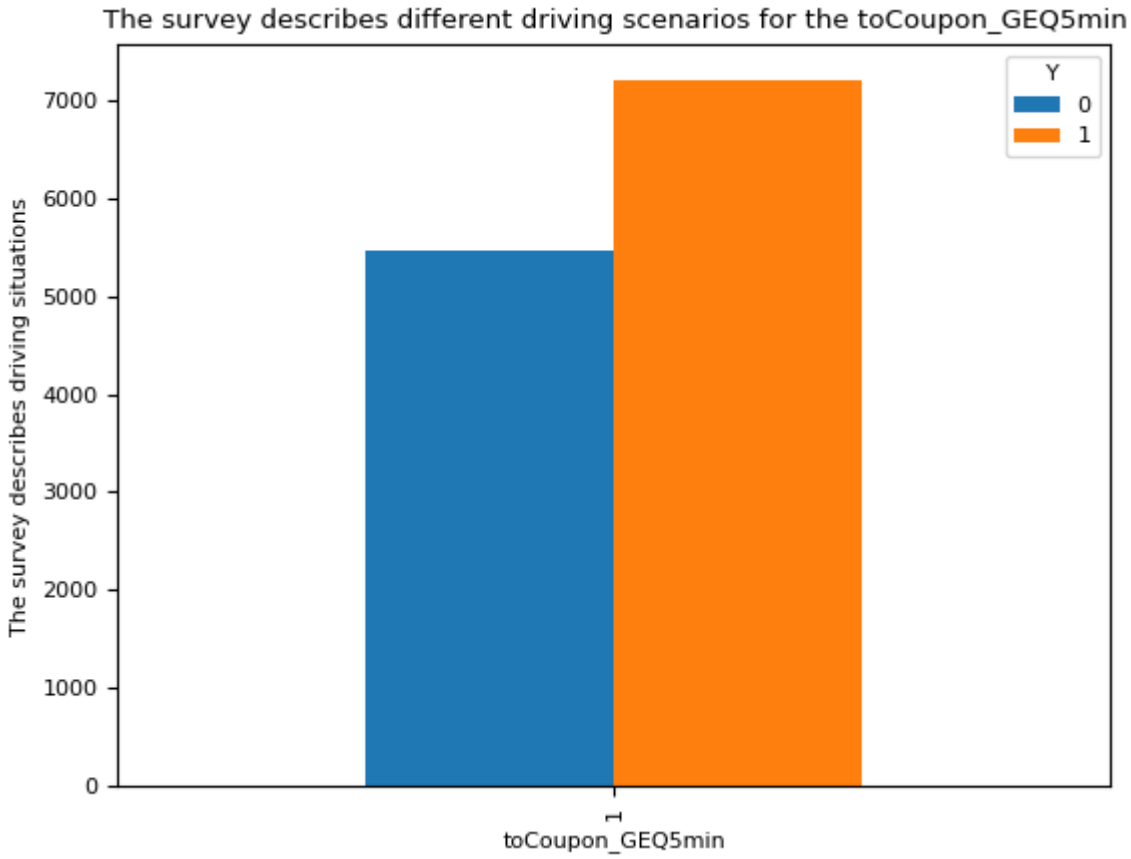
Hình 27. Biểu đồ cột giữa số lần đến nhà hàng với chi phí TB từ 20-50 một người và Y

Giữa Khoảng cách lái xe đến nhà hàng/quán bar để sử dụng phiếu giảm giá lớn hơn 15 phút với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['toCoupon_GEQ5min'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the toCoupon_GEQ5min')
plt.xlabel('toCoupon_GEQ5min')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=8)
print("##### driving situation to the toCoupon_GEQ5min and asking if he accepts the coupon #####")
```

✓ 1.2s

```
##### driving situation to the toCoupon_GEQ5min and asking if he accepts the coupon #####
```



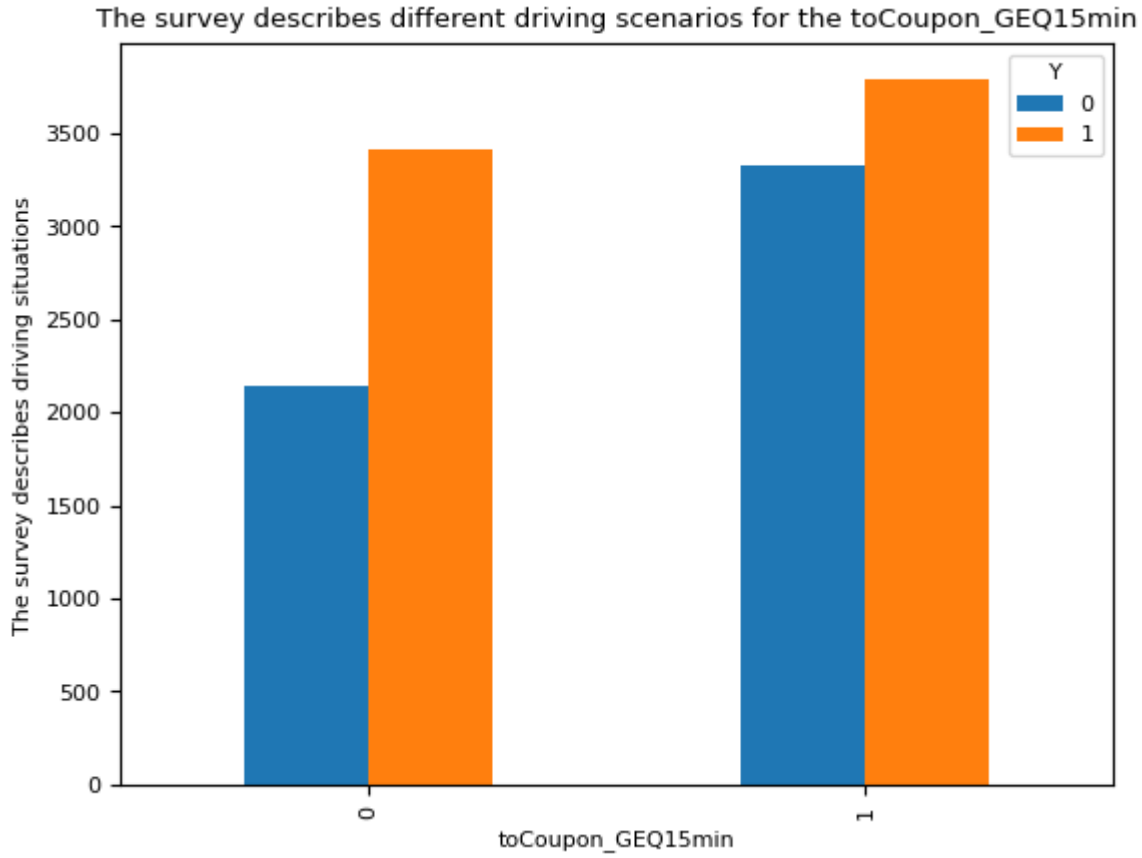
Hình 28. Biểu đồ cột giữa khoảng cách lái xe từ nhà hàng hoặc bar trên 15 phút và Y

Giữa Khoảng cách lái xe đến nhà hàng/quán bar để sử dụng phiếu giảm giá lớn hơn 25 phút với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['toCoupon_GEQ15min'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the toCoupon_GEQ15min')
plt.xlabel('toCoupon_GEQ15min')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=8)
print("##### driving situation to the toCoupon_GEQ15min and asking if he accepts the coupon #####")
```

✓ 1.3s

```
##### driving situation to the toCoupon_GEQ15min and asking if he accepts the coupon #####
```



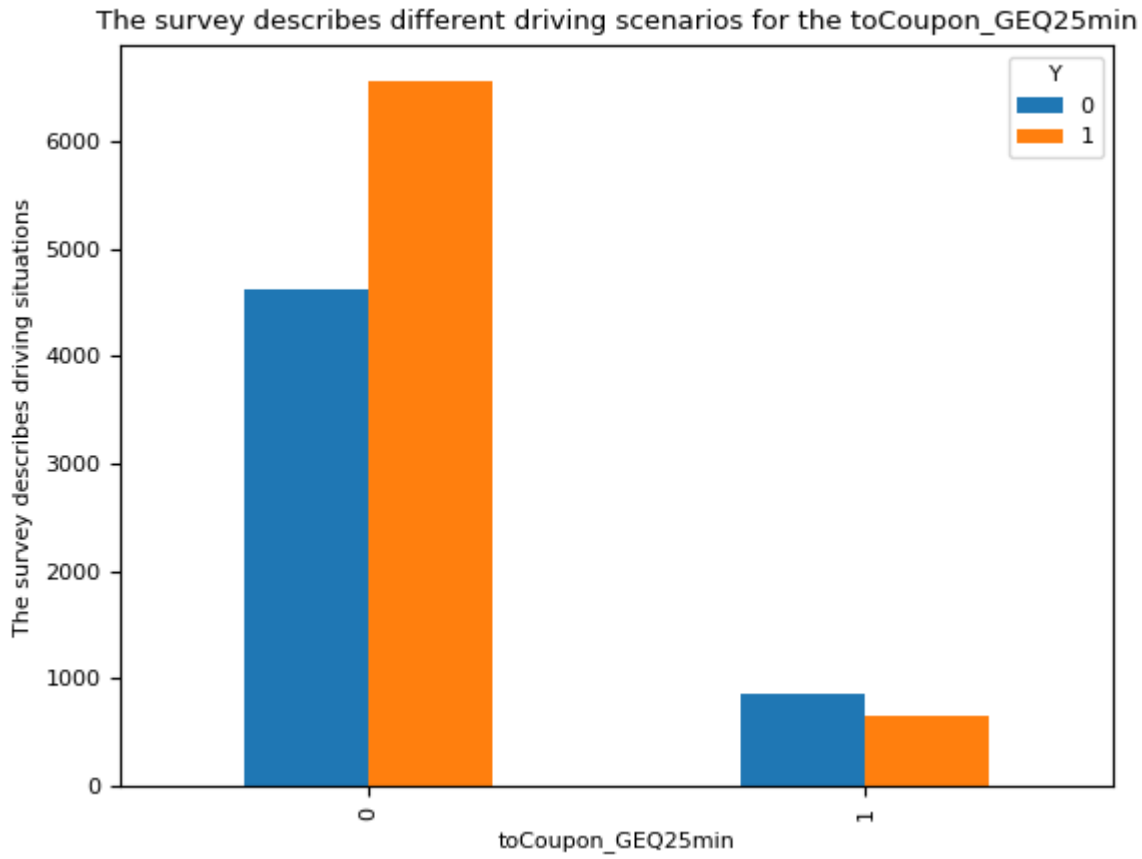
Hình 29. Biểu đồ cột giữa khoảng cách lái xe từ nhà hàng hoặc bar trên 25 phút và Y

Giữa Nhà hàng/quán bar có cùng hướng với điểm đến hiện tại không với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['toCoupon_GEQ25min'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the toCoupon_GEQ25min')
plt.xlabel('toCoupon_GEQ25min')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=8)
print("##### driving situation to the toCoupon_GEQ25min and asking if he accepts the coupon #####")
```

✓ 1.4s

```
##### driving situation to the toCoupon_GEQ25min and asking if he accepts the coupon #####
```



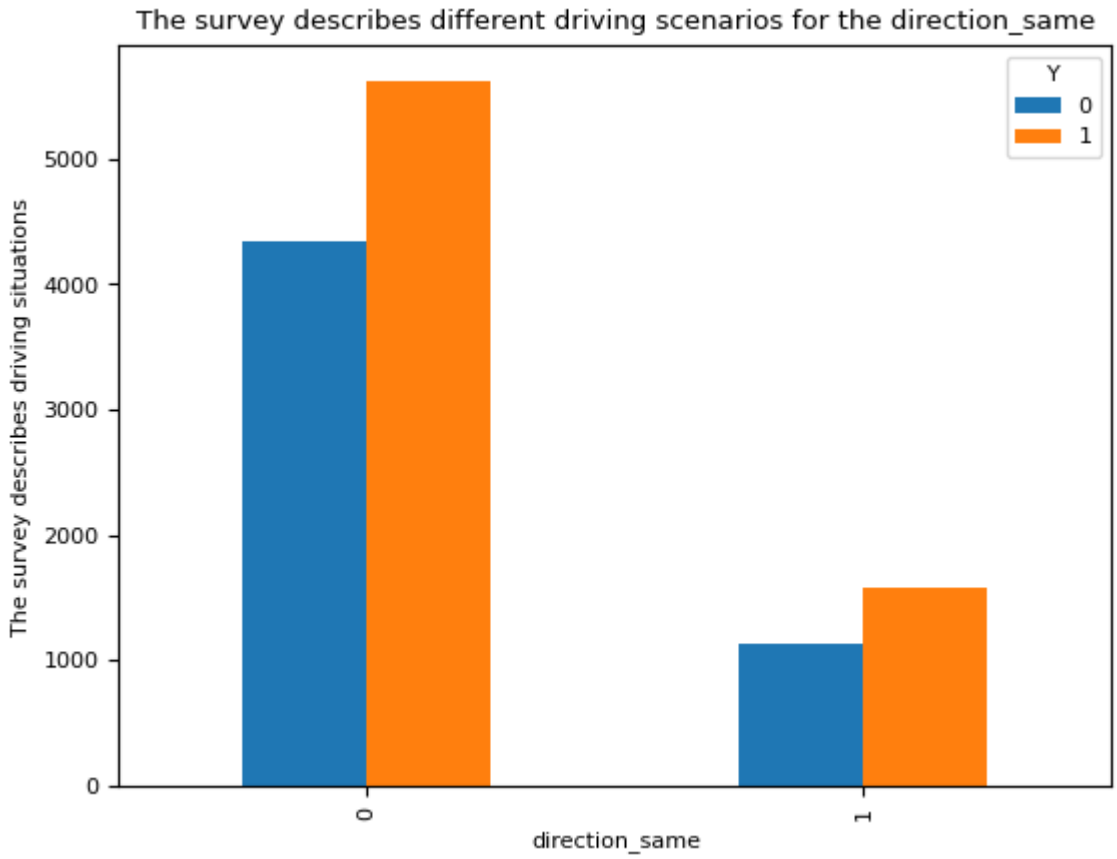
Hình 30. Biểu đồ cột giữa từ nhà hàng hoặc bar có cùng hướng với điểm đến hiện tại và Y

Giữa Nhà hàng/quán bar có ngược hướng với điểm đến hiện tại không với Y (có chấp nhận phiếu giảm giá hay không)

```
pd.crosstab(data['direction_same'], data['Y']).plot(kind='bar')
plt.title('The survey describes different driving scenarios for the direction_same')
plt.xlabel('direction_same')
plt.ylabel('The survey describes driving situations')
plt.rc('font', size=8)
print("##### driving situation to the direction_same and asking if he accepts the coupon #####")
```

✓ 1.2s

```
##### driving situation to the direction_same and asking if he accepts the coupon #####
```



Hình 31. Biểu đồ cột giữa từ nhà hàng hoặc bar có ngược hướng với điểm đến hiện tại và Y

Tiến hành phân tích thử xem những tên thuộc tính nào có dữ liệu dạng string và dạng int64

```
#load data
df = pd.read_csv('in-vehicle-coupon-recommendation.csv')

df.shape #lets check the dimensionality of the raw data
#Xem xét các loại dữ liệu
df.dtypes
```

✓ 0.1s

```
destination              object
passanger                object
weather                  object
temperature               int64
time                     object
coupon                   object
expiration               object
gender                   object
age                      object
maritalStatus            object
has_children              int64
education                object
occupation               object
income                   object
car                      object
Bar                      object
CoffeeHouse              object
CarryAway                object
RestaurantLessThan20     object
Restaurant20To50         object
toCoupon_GEQ5min          int64
toCoupon_GEQ15min         int64
toCoupon_GEQ25min         int64
direction_same            int64
direction_opp             int64
Y                         int64
dtype: object
```

Phân tích một chút về features ta thấy ('destination', 'passanger', 'weather', 'time', 'coupon', 'expiration', 'gender', 'age', 'maritalStatus', 'education', 'occupation', 'income', 'car', 'Bar', 'CoffeeHouse', 'CarryAway', 'RestaurantLessThan20', 'Restaurant20To50') là một bản ghi dạng text, không liên quan đến việc có được sử dụng mã giảm giá hay không

Cái thứ 2, để xây dựng model dự đoán ta cần chuyển data ở features tất cả sang dạng số mà tất cả những thuộc tính ghi trên lại dạng chữ.

Để làm được 2 việc trên ta cân thực hiện 2 bước:

Bước 1: Chuyển dữ liệu cột ('destination', 'passanger', 'weather', 'time', 'coupon', 'expiration', 'gender', 'age', 'maritalStatus', 'education', 'occupation', 'income', 'car', 'Bar', 'CoffeeHouse', 'CarryAway', 'RestaurantLessThan20', 'Restaurant20To50') thành các cột tương ứng dạng số có giá trị 0 hoặc 1

```
cat_features = [
    'destination', 'passanger', 'weather', 'time','coupon','expiration', 'gender', 'age','maritalStatus',
    'education', 'occupation', 'income', 'car', 'Bar', 'CoffeeHouse', 'CarryAway', 'RestaurantLessThan20', 'Restaurant20To50'
]

for feature in cat_features:
    print(feature)
    job_list = pd.get_dummies(data[feature], prefix = feature)
    new_data = data.join(job_list)
    data = new_data

data.head()
```

✓ 0.3s

```
destination
passanger
weather
time
coupon
expiration
gender
age
maritalStatus
education
education
occupation
income
car
Bar
CoffeeHouse
CarryAway
RestaurantLessThan20
```

	destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	...	RestaurantLessThan20_1~3
0	No Urgent Place	Alone	Sunny	55	2PM	Restaurant(<20)	1d	Female	21	Unmarried partner	...	0
1	No Urgent Place	Friend(s)	Sunny	80	10AM	Coffee House	2h	Female	21	Unmarried partner	...	0
2	No Urgent Place	Friend(s)	Sunny	80	10AM	Carry out & Take away	2h	Female	21	Unmarried partner	...	0
3	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	2h	Female	21	Unmarried partner	...	0
4	No Urgent Place	Friend(s)	Sunny	80	2PM	Coffee House	1d	Female	21	Unmarried partner	...	0

5 rows × 133 columns

Bước 2: Loại bỏ các cột không sử dụng để xây dựng model

```
data_features = data.columns.values.tolist()
print(data_features)
remove_features = [
    'destination', 'passanger', 'weather', 'time','coupon','expiration', 'gender', 'age','maritalStatus', 'education',
    'occupation', 'income', 'car', 'Bar', 'CoffeeHouse', 'CarryAway', 'RestaurantLessThan20', 'Restaurant20To50'
]

to_keep_features = [i for i in data_features if i not in remove_features]
print(to_keep_features)
```

✓ 0.7s Python

```
['destination', 'passanger', 'weather', 'temperature', 'time', 'coupon', 'expiration', 'gender', 'age', 'maritalStatus', 'has_children', 'education',
'occupation', 'income', 'car', 'Bar', 'CoffeeHouse', 'CarryAway', 'RestaurantLessThan20', 'Restaurant20To50', 'toCoupon_GEQ5min', 'toCoupon_GEQ15min',
'toCoupon_GEQ25min', 'direction_same', 'direction_opp', 'Y', 'destination_Home', 'destination_No Urgent Place', 'destination_Work', 'passanger_Alone',
'passanger_Friend(s)', 'passanger_Kid(s)', 'passanger_Partner', 'weather_Rainy', 'weather_Snowy', 'weather_Sunny', 'time_10AM', 'time_10PM', 'time_2PM',
'time_6PM', 'time_7AM', 'coupon_Bar', 'coupon_Carry out & Take away', 'coupon_Coffee House', 'coupon_Restaurant(20-50)', 'coupon_Restaurant(<20)',
'expiration_1d', 'expiration_2h', 'gender_Female', 'gender_Male', 'age_21', 'age_26', 'age_31', 'age_36', 'age_41', 'age_46', 'age_50plus', 'age_below21',
'maritalStatus_Divorced', 'maritalStatus_Married partner', 'maritalStatus_Single', 'maritalStatus_Unmarried partner', 'maritalStatus_Widowed',
'education_Associates degree', 'education_Bachelors degree', 'education_Graduate degree (Masters or Doctorate)', 'education_High School Graduate',
'education_Some High School', 'education_Some college - no degree', 'occupation_Architecture & Engineering', 'occupation_Arts Design Entertainment Sports &
Media', 'occupation_Building & Grounds Cleaning & Maintenance', 'occupation_Business & Financial', 'occupation_Community & Social Services', 'occupation_Computer
& Mathematical', 'occupation_Construction & Extraction', 'occupation_Education&Training&Library', 'occupation_Farming Fishing & Forestry', 'occupation_Food
Preparation & Serving Related', 'occupation_Healthcare Practitioners & Technical', 'occupation_Healthcare Support', 'occupation_Installation Maintenance &
Repair', 'occupation_Legal', 'occupation_Life Physical Social Science', 'occupation_Management', 'occupation_Office & Administrative Support',
'occupation_Personal Care & Service', 'occupation_Production Occupations', 'occupation_Protective Service', 'occupation_Retired', 'occupation_Sales & Related',
'occupation_Student', 'occupation_Transportation & Material Moving', 'occupation_Unemployed', 'income_$100000 or More', 'income_$12500 - $24999', 'income_$25000
- $37499', 'income_$37500 - $49999', 'income_$50000 - $62499', 'income_$62500 - $74999', 'income_$75000 - $87499', 'income_$87500 - $99999', 'income_Less than
$12500', 'car_Car that is too old to install Onstar :D', 'car_Mazda5', 'car_Scooter and motorcycle', 'car_crossover', 'car_do not drive', 'Bar_1~3', 'Bar_4~8',
'Bar_gt8', 'Bar_less1', 'Bar_never', 'CoffeeHouse_1~3', 'CoffeeHouse_4~8', 'CoffeeHouse_gt8', 'CoffeeHouse_less1', 'CoffeeHouse_never', 'CarryAway_1~3',
'CarryAway_4~8', 'CarryAway_gt8', 'CarryAway_less1', 'CarryAway_never', 'RestaurantLessThan20_1~3', 'RestaurantLessThan20_4~8', 'RestaurantLessThan20_gt8',
'RestaurantLessThan20_less1', 'RestaurantLessThan20_never', 'Restaurant20To50_1~3', 'Restaurant20To50_4~8', 'Restaurant20To50_gt8', 'Restaurant20To50_less1',
```

Hiển thị 5 dòng đầu tiên của dataset sau khi ta vừa chỉnh sửa

```
data_final = data[to_keep_features]
data_final.head()
```

✓ 0.2s Python

	temperature	has_children	toCoupon_GEQ5min	toCoupon_GEQ15min	toCoupon_GEQ25min	direction_same	direction_opp	Y	destination_Home	destination_No Urgent Place	...	Resta
0	55	1	1	0	0	0	1	1	0	1	...	
1	80	1	1	0	0	0	1	0	0	1	...	
2	80	1	1	1	0	0	1	1	0	1	...	
3	80	1	1	1	0	0	1	0	0	1	...	
4	80	1	1	1	0	0	1	0	0	1	...	

5 rows × 115 columns

Hiển thị data features

```
features = np.array(data_final.loc[:, data_final.columns != 'Y'])
print(features)
```

✓ 0.1s

```
[[55  1  1 ...  0  0  0]
 [80  1  1 ...  0  0  0]
 [80  1  1 ...  0  0  0]
 ...
 [30  0  1 ...  0  0  0]
 [30  0  1 ...  0  0  0]
 [80  0  1 ...  0  0  0]]
```

Hiển thị data targets

```
targets = np.array(data_final.loc[:, data_final.columns == 'Y'])
print(targets)
```

✓ 0.1s

```
[[1]
 [0]
 [1]
 ...
 [0]
 [0]
 [0]]
```

Dùng train_test_split để tách dự liệu ngẫu nhiên ra train và test

Ta lấy 30% để test

```
train_features, test_features, train_targets, test_targets = train_test_split(
    features, targets, test_size = 0.3, random_state=0
)
print("##### Training and test datasets #####")
print("Training size: ", len(train_targets))
print("Test size: ", len(test_targets))
print(test_targets)
```

✓ 0.1s

```
##### Training and test datasets #####
Training size:  8878
Test size:  3806
[[0]
 [1]
 [0]
 ...
 [0]
 [1]
 [0]]
```

Xây dựng model

```
classifier_logreg = LogisticRegression()
classifier_logreg.fit(train_features, train_targets)
```

✓ 0.6s

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

▾ LogisticRegression
LogisticRegression()

2.5. Xây dựng thuật toán dự đoán, tính toán độ chính xác và time

2.5.1. Thuật toán Logistic Regression

Sử dụng tập test để dự đoán kết quả chạy trong 0.6s

```
predictions = classifier_logreg.predict(test_features)
print("##### Prediction results of Logistic Regression #####")
print("Target labels: ", test_targets.T)
print("Prediction labels:  ", predictions)
```

✓ 0.6s

```
##### Prediction results of Logistic Regression #####
Target labels:  [[0 1 0 ... 0 1 0]]
Prediction labels:   [1 1 0 ... 0 1 0]
```

Tính độ chính xác của model dự đoán sau khi ta dùng tập test để dự đoán trong 0.1s

```
accuracy = 100 * accuracy_score(test_targets, predictions)
print("##### Prediction accuracy of Logistic Regression #####")
print("Accuracy: ", accuracy)
print(classification_report(test_targets, predictions))
```

✓ 0.1s

```
##### Prediction accuracy of Logistic Regression #####
Accuracy:  67.39358906988964
              precision    recall  f1-score   support

           0       0.66      0.56      0.60      1696
           1       0.68      0.77      0.72      2110

    accuracy                           0.67      3806
   macro avg       0.67      0.66      0.66      3806
weighted avg       0.67      0.67      0.67      3806
```

Hình 32. Kết quả thuật toán Logistic Regression với tập Test

Sử dụng tập test để dự đoán trong 0.8s và độ chính xác ~67,4%

```
predictions = classifier_logreg.predict(train_features)
print("##### Training - Prediction results of Logistic Regression #####")
print("Target labels: ", train_targets.T)
print("Prediction labels:  ", predictions)
```

✓ 0.8s

```
##### Training - Prediction results of Logistic Regression #####
Target labels:  [[0 1 0 ... 0 0 1]]
Prediction labels:   [1 1 0 ... 0 0 1]
```

Tính độ chính xác của mô hình dự đoán khi ta sử dụng tập train để dự đoán trong 0.1s

```
accuracy = 100 * accuracy_score(train_targets, predictions)
print("##### Prediction accuracy of Logistic Regression #####")
print("Accuracy: ", accuracy)
print(classification_report(train_targets, predictions))
```

✓ 0.1s

```
##### Prediction accuracy of Logistic Regression #####
Accuracy:  69.80175715251183
              precision    recall  f1-score   support

           0       0.67      0.58      0.62      3778
           1       0.72      0.79      0.75      5100

    accuracy                           0.70      8878
   macro avg       0.69      0.68      0.68      8878
weighted avg       0.70      0.70      0.69      8878
```

Hình 33. Kết quả thuật toán Logistic Regression với tập Train

Qua đó cho thấy Độ chính xác của mô hình logistic Regression trên tập test: ~67,4%.

Thời gian trả kết quả trên tập test là: 0.1s

Độ chính xác của mô hình logistic Regression trên tập train: ~69,8%.

Thời gian trả kết quả trên tập train là: 0.1s

2.5.2. Thuật toán K-nearest neighbor (KNN)

```
from sklearn import neighbors
classifier_KNN = neighbors.KNeighborsClassifier(n_neighbors = 1, p = 2)
classifier_KNN.fit(train_features, train_targets)

# K-nearest neighbor
predictions = classifier_KNN.predict(train_features)
print("##### Training - Prediction results of KNN #####")
print("Target labels: ", train_targets.T)
print("Prediction labels:  ", predictions)
accuracy = 100 * accuracy_score(train_targets, predictions)
print("##### Prediction accuracy of KNN  #####")
print("Accuracy: ", accuracy)
print(classification_report(train_targets, predictions))
```

✓ 3.4s

```
##### Training - Prediction results of KNN #####
Target labels:  [[0 1 0 ... 0 0 1]]
Prediction labels:   [0 1 0 ... 0 0 1]
##### Prediction accuracy of KNN  #####
Accuracy:  99.88736201847263
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3778
           1       1.00      1.00      1.00      5100

    accuracy                           1.00      8878
   macro avg       1.00      1.00      1.00      8878
weighted avg       1.00      1.00      1.00      8878
```

Hình 34. Kết quả thuật toán K-nearest neighbor với tập train

Sử dụng tập train để dự đoán trong 3,4s và độ chính xác ~99,9%

```python
# K-nearest neighbor
predictions = classifier_KNN.predict(test_features)
print("##### Training - Prediction results of KNN #####")
print("Target labels: ", test_targets.T)
print("Prediction labels:  ", predictions)
accuracy = 100 * accuracy_score(test_targets, predictions)
print("##### Prediction accuracy of KNN  #####")
print("Accuracy: ", accuracy)
print(classification_report(test_targets, predictions))
```

✓ 1.4s

```
##### Training - Prediction results of KNN #####
Target labels:  [[0 1 0 ... 0 1 0]]
Prediction labels:   [1 0 0 ... 0 1 1]
##### Prediction accuracy of KNN  #####
Accuracy:  64.76615869679453
              precision    recall  f1-score   support

           0       0.61      0.59      0.60      1696
           1       0.68      0.70      0.69      2110

    accuracy                           0.65      3806
   macro avg       0.64      0.64      0.64      3806
weighted avg       0.65      0.65      0.65      3806
```

Hình 35. Kết quả thuật toán K-nearest neighbor với tập test

Sử dụng tập test để dự đoán trong 1,4s và độ chính xác ~64,8%

2.5.3. Thuật toán Naive Bayes

```
#thuật toán Naive Bayes
from sklearn.naive_bayes import GaussianNB
classifier_NB = GaussianNB()
classifier_NB.fit(train_features, train_targets)

# K-nearest neighbor train
predictions = classifier_NB.predict(train_features)
print("##### Training - Prediction results of Naive Bayes #####")
print("Target labels: ", train_targets.T)
print("Prediction labels:  ", predictions)
accuracy = 100 * accuracy_score(train_targets, predictions)
print("##### Prediction accuracy of Naive Bayes  #####")
print("Accuracy: ", accuracy)
print(classification_report(train_targets, predictions))
```

✓ 0.1s

```
##### Training - Prediction results of Naive Bayes #####
Target labels:  [[0 1 0 ... 0 0 1]]
Prediction labels:   [0 0 0 ... 0 0 1]
##### Prediction accuracy of Naive Bayes  #####
Accuracy:  63.42644739806262
              precision    recall  f1-score   support

           0       0.56      0.69      0.62      3778
           1       0.72      0.59      0.65      5100

    accuracy                           0.63      8878
   macro avg       0.64      0.64      0.63      8878
weighted avg       0.65      0.63      0.64      8878
```

Hình 36. Kết quả thuật toán Naive Bayes với tập train

Sử dụng tập train để dự đoán trong 0.1s và độ chính xác ~63,4%

```
# K-nearest neighbor test
predictions = classifier_NB.predict(test_features)
print("##### Training - Prediction results of Naive Bayes #####")
print("Target labels: ", test_targets.T)
print("Prediction labels:  ", predictions)
accuracy = 100 * accuracy_score(test_targets, predictions)
print("##### Prediction accuracy of Naive Bayes  #####")
print("Accuracy: ", accuracy)
print(classification_report(test_targets, predictions))
```

✓ 0.7s

```
##### Training - Prediction results of Naive Bayes #####
Target labels:  [[0 1 0 ... 0 1 0]]
Prediction labels:   [1 0 0 ... 0 0 0]
##### Prediction accuracy of Naive Bayes  #####
Accuracy:  62.874408828166054
              precision    recall  f1-score   support

           0       0.57      0.68      0.62      1696
           1       0.70      0.59      0.64      2110

    accuracy                           0.63      3806
   macro avg       0.63      0.63      0.63      3806
weighted avg       0.64      0.63      0.63      3806
```

Hình 37. Kết quả thuật toán Naive Bayes với tập test

Sử dụng tập test để dự đoán trong 0.7s và độ chính xác ~62,9%

2.5.4. Thuật toán Tree Decision (Cây Quyết Định)

```
#train thuật toán Tree Decision (Cây Quyết Định)
from sklearn.tree import DecisionTreeClassifier
classifier_TD = DecisionTreeClassifier( criterion = "entropy", random_state = 100, max_depth = 3, min_samples_leaf = 5)
classifier_TD.fit(train_features, train_targets)
# K-nearest Tree Decision train
predictions = classifier_TD.predict(train_features)
print("##### Training - Prediction results of Tree Decision #####")
print("Target labels: ", train_targets.T)
print("Prediction labels:  ", predictions)
accuracy = 100 * accuracy_score(train_targets, predictions)
print("##### Prediction accuracy of Tree Decision  #####")
print("Accuracy: ", accuracy)
print(classification_report(train_targets, predictions))
```

✓ 0.2s

```
##### Training - Prediction results of Tree Decision #####
Target labels:  [[0 1 0 ... 0 0 1]]
Prediction labels:   [1 1 1 ... 1 1 1]
##### Prediction accuracy of Tree Decision  #####
Accuracy:  63.42644739806262
              precision    recall  f1-score   support

           0       0.72      0.23      0.35      3778
           1       0.62      0.94      0.75      5100

    accuracy                           0.63      8878
   macro avg       0.67      0.58      0.55      8878
weighted avg       0.66      0.63      0.58      8878
```

Hình 38. Kết quả thuật toán True Decision với tập train

Sử dụng tập Train để dự đoán trong 0.2s và độ chính xác ~63,4%

```
# K-nearest neighbor test
predictions = classifier_TD.predict(test_features)
print("##### Training - Prediction results of Tree Decision #####")
print("Target labels: ", test_targets.T)
print("Prediction labels:  ", predictions)
accuracy = 100 * accuracy_score(test_targets, predictions)
print("##### Prediction accuracy of Tree Decision  #####")
print("Accuracy: ", accuracy)
print(classification_report(test_targets, predictions))
```

✓ 0.8s

```
##### Training - Prediction results of Tree Decision #####
Target labels:  [[0 1 0 ... 0 1 0]]
Prediction labels:   [1 1 1 ... 1 1 1]
##### Prediction accuracy of Tree Decision  #####
Accuracy:  61.928533893851814
              precision    recall  f1-score   support

           0       0.72      0.23      0.35      1696
           1       0.60      0.93      0.73      2110

    accuracy                           0.62      3806
   macro avg       0.66      0.58      0.54      3806
weighted avg       0.66      0.62      0.56      3806
```

Hình 39. Kết quả thuật toán True Decision với tập test

Sử dụng tập test để dự đoán trong 0.8s và độ chính xác ~61,9%

2.6. So sánh, nhận xét và đưa ra kết luận giữa các thuật toán

STT	Thuật Toán	Độ chính xác Train	Độ chính xác Test
1	Logistic Regression	69.8%; 0.9s	67.4%; 0.6s
2	K-Nearest Neighbors	99.8%; 3.4s	64.8%; 1.4s
3	Naive Bayes	63.4%; 0.1s	62.9%; 0.7s
4	Tree Decision	63.4%; 0.2s	61.9%; 0.8s

Qua bảng so sánh sự khác nhau của các thuật toán "kết quả trên là kết quả trung bình của nhiều lần chạy huấn luyện)".

- Thuật toán có thời gian Test thấp nhất là thuật toán Naive Bayes
- Thuật toán có thời gian Test cao nhất là K-Nearest Neighbors
- Thuật toán có thời gian train thấp nhất là thuật toán Naive Bayes
- Thuật toán có thời gian train cao nhất là K-Nearest Neighbors
- Thuật toán có độ chính xác cao nhất trên tập train là thuật toán K-Nearest Neighbors, có độ chính xác ~99,8%.
- Thuật toán có độ chính xác thấp nhất trên tập train là thuật toán Tree Decision và Tree Decision, có độ chính xác ~63,4%.
- Thuật toán có độ chính xác cao nhất trên tập test là thuật toán Logistic Regression, có độ chính xác ~67,4%.
- Thuật toán có độ chính xác thấp nhất trên tập test là thuật toán Tree Decision, có độ chính xác ~61,9%.

Theo bảng trên và kết luận phía dưới có thể đưa ra kết luận rằng: Các thuật toán phân loại áp dụng trong mô hình này đều có độ chính xác trên 60%. Nên phải tùy vào trường hợp sử dụng để có thể dự đoán rằng coupon có được chấp nhận hay không.

TÀI LIỆU THAM KHẢO

1. https://aws.amazon.com/vi/what-is/machine-learning/
2. https://www.iostream.vn/ai-ml/bai-toan-phan-lop-trong-machine-learning-classification-in-machine-learning-515OIh

Đường link lưu trữ bài: https://github.com/lehongphongxm/Tri_Tue_Nhan_Tao