HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

BÁO CÁO BÀI TẬP LỚN
MÔN: KHO DỮ LIỆU VÀ KHAI PHÁ DỮ LIỆU
Đề tài: Dự đoán khả năng hủy phòng của khách hàng dựa trên lịch sử đặt phòng khách sạn

Nhóm: 01
Giảng viên: Nguyễn Quỳnh Chi
Thành viên: Nguyễn Trọng An – B16DCCN003
Lê Thị Ly – B16DCCN221
Trần Ngọc Nam – B16DCCN245
Trần Văn Tâm – B16DCCN308

Hà Nội, tháng 7 năm 2020

Mục Lục

1. Lời giới thiệu

Vấn đề bùng nổ về dữ liệu: khi các công cụ thu thập dữ liệu tự động và công nghệ về cơ sở dữ liệu đã trở nên hoàn thiện, một lượng lớn dữ liệu được thu thập và lưu trữ trong những các cơ sở dữ liệu, kho dữ liệu và các kho lưu trữ thông tin khác.

Lúc này, chúng ta đang có quá nhiều dữ liệu, chưa mang tính phục vụ có mục đích cho người sử dụng. Chúng ta đang thiếu tri thức, dữ liệu đã qua xử lý và phục vụ riêng cho mục đích của người sử dụng. Vấn đề là làm thế nào để khai thác tri thức từ đống dữ liệu khổng lồ hiện đang có trong tay.

Giải pháp cho việc khai phá ra tri thức chính là sự ra đời của công nghệ kho dữ liệu và các phương pháp khai phá dữ liệu. Giải pháp này liên quan tới những khía cạnh sau đây:

- Công nghệ để xây dựng một kho dữ liệu lớn và các phương thức để xử lý phân tích trực tuyến

- Trích lọc ra tri thức có ích cho con người bao gồm các luật, thể chế, mẫu, và các ràng buộc từ khối lượng lớn dữ liệu của một hay nhiều cơ sở dữ liệu có kích cỡ lớn.

Các lý do cần khai phá dữ liệu trên quan điểm thương mại trong thế giới thực.

- Rất nhiều dữ liệu đã được thu thập trong thế giới thực và được lưu trữ một cách hệ thống trong các kho dữ liệu bao gồm:

  - Các dữ liệu trên web, các dữ liệu thương mại điện tử
  - Các dữ liệu mua bán tại các cửa hàng, gian hàng trong siêu thị
  - Các dữ liệu của giao dịch ngân hàng, thẻ tín dụng

- Máy tính trở nên rẻ hơn và có sức mạnh xử lý dữ liệu hơn

- Sức ép cạnh tranh mạnh mẽ hơn: cần cung cấp các dịch vụ tốt hơn và tùy biến với khách hàng hơn (nhất là trong quan hệ với khách hàng)

Các lý do cần khai phá dữ liệu trên quan điểm khoa học

- Các dữ liệu được thu thập và lưu trữ với tốc độ rất nhanh (GB/h) thông qua

  - Bộ cảm biến (sensor) điều khiển từ xa trên các trạm vệ tinh
  - Kính viễn vọng quan sát bầu trời
  - Dùng công cụ microarray để sinh ra dữ liệu thể hiện đặc tính của gene (gene expression data)
  - Dùng các bộ mô phỏng khoa học để tạo ra hàng tera byte dữ liệu

- Các kỹ thuật truyền thống không còn khả thi cho lượng lớn các dữ liệu thô

- Các kỹ thuật khai phá dữ liệu có thể sẽ giúp ích được các nhà khoa học hơn trong các công việc

  - Phân loại và phân mảnh dữ liệu
  - Hình thành các giả thuyết trong nghiên cứu khoa học

Khai phá dữ liệu (phát hiện tri thức trong cơ sở dữ liệu sẵn có) là việc trích lọc ra những thông tin có ích (không hiển nhiên, không tường minh, không biết trước, và có ích một cách tiềm năng), những mẫu dữ liệu trong cơ sở dữ liệu lớn.
Khai phá dữ liệu có một số tên gọi khác khi được sử dụng khi được đề cập đến trong cuộc sống cũng như trong sách và tạp chí khoa học như:

- Khám phá tri thức (knowledge discovery) trong cơ sở dữ liệu (thường được viết tắt theo tiếng anh là KDD).
- Trích lọc tri thức
- Phân tích mẫu/dữ liệu
- Khảo cổ dữ liệu
- Tri thức kinh doanh (business intelligence) và còn nhiều tên khác nữa ít dùng.

Khi thực hiện một công việc khai phá dữ liệu, để đưa ra các quyết định cần thiết cho công việc khai phá, chúng ta cần xác định những yếu tố sau:

- Loại cơ sở dữ liệu cần khai phá: Các loại cơ sở dữ liệu có thể dùng cho khai phá bao gồm cơ sở dữ liệu quan hệ, cơ sở dữ liệu giao dịch, hướng đối tượng, cơ sở dữ liệu quan hệ - đối tượng, không gian, cơ sở dữ liệu văn bản, chuỗi thời gian, đa phương tiện, cơ sở dữ liệu hỗn tạp, cơ sở dữ liệu luật, cơ sở dữ liệu Web, và các loại cơ sở dữ liệu khác nữa.

- Loại tri thức cần phát hiện ra: Bao gồm tri thức miêu tả đặc điểm của các cá thể trong tập cá thể đang xét, phân biệt cá thể này với cá thể khác, luật kết hợp, tìm xu hướng, phân loại cá thể trong một tập hợp, phân cụm gộp nhóm các cá thể giống nhau, phân tích tìm ra cá thể ngoại lai và sự khác biệt đối với phần đông các cá thể khác, v.v... Ngoài ra, tri thức còn là các chức năng tích hợp, đa chức năng và khai phá ở nhiều mức độ khác nhau.

- Loại kỹ thuật cần được sử dụng để giải quyết vấn đề: Bao gồm kỹ thuật theo hướng cơ sở dữ liệu, kỹ thuật kho dữ liệu (xử lý phân tích trực tuyến), các phương pháp học máy, các phương pháp thống kê, biểu diễn trực quan, mạng nơron nhân tạo, và các phương pháp khác.

- Loại ứng dụng cần được xây dựng, áp dụng cho vấn đề khai phá: Bao gồm các ứng dụng trong lĩnh vực bán lẻ, truyền thông, ngân hàng, phân tích lỗi, khai phá dữ liệu gen, phân tích thị trường chứng khoán, khai phá dữ liệu Web, phân tích Weblog

Một công việc nữa cần được xác định là nhận thức rõ nhiệm vụ của bài toán khai phá dữ liệu là thuộc loại nào trong hai loại sau đây:

- Bài toán khai phá dữ liệu dạng mô tả: Nhiệm vụ của bài toán dạng này là tìm ra các mẫu mô tả dữ liệu mà con người có thể hiểu được.

- Bài toán khai phá dữ liệu dạng tiên đoán: Sử dụng một vài biến để tiên đoán các giá trị chưa biết hoặc trong tương lai của các biến khác.

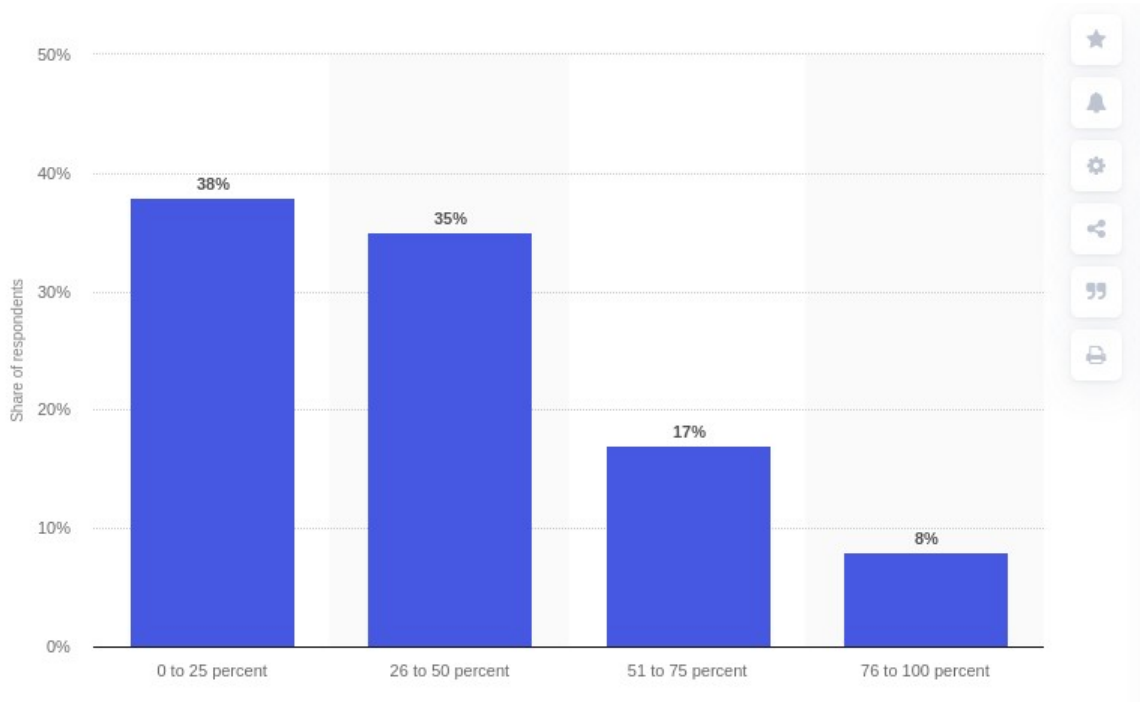
Các nhiệm vụ thường gặp của việc khai phá dữ liệu

- Phân loại: thuộc loại bài toán tiên đoán
- Phân cụm: thuộc loại bài toán mô tả
- Phát hiện luật kết hợp: thuộc loại bài toán mô tả
- Phát hiện mẫu dạng liên tục: thuộc loại bài toán mô tả
- Bài toán hồi quy: thuộc loại bài toán tiên đoán
- Phát hiện sự khác biệt: thuộc loại bài toán tiên đoán

2. Lí Do Chọn Đề Tài.

Ngày nay, dịch vụ Khách sạn – Nhà hàng đã và đang phát triển vô cùng mạnh mẽ nhằm đáp ứng được nhu cầu ngày càng cao của con người. Vậy bạn có biết ngành nghề Khách sạn xuất hiện từ khi nào và phát triển ra sao không?Từ đầu thế kỷ 16 trước Công nguyên, con người đã bắt đầu biết đến trao đổi ngoại thương và du lịch. Các vùng du lịch bắt đầu được mở rộng, nhu cầu chỗ ở và ăn uống ngày càng cao, tuy nhiên nhà hàng khách sạn thì còn rất sơ khai, yếu kém về dịch vụ và lòng hiếu khách, chỉ được điều hành bởi những người không chuyên và chủ nhà lạc hậu.Cho đến khi cuộc Cách mạng kỹ nghệ ở Anh năm 1790 mới có những dấu hiệu của sự tiến bộ và những ý tưởng mới về kinh doanh nhà trọ.

Cùng với sự phát triển thì việc đảm chất lượng dịch vụ cho khách hàng là một trong các yếu tố hết sức quan trọng để giữ khách cũng như tăng tỉ lệ cạnh tranh đối với các khách sạn khác.



Hình 2.1 Guests returning to hotels worldwide as of July 2014 , Published by Statista Research Department [1]

Hình 2.1 là biểu đồ thể hiện thống kê cho thấy sự quay trở lại khách sạn trên toàn thế giới kể từ tháng 7 năm 2014. Trong cuộc khảo sát:

- 38 % khách sạn được hỏi thì có đến 25 % khách của họ quay trở lại khách sạn hoặc quay lại tại 1 cơ sở khác của khách sạn.
- 35 % khách sạn được hỏi thì có đến 50 % khách của họ quay trở lại khách sạn hoặc quay lại tại 1 cơ sở khác của khách sạn.

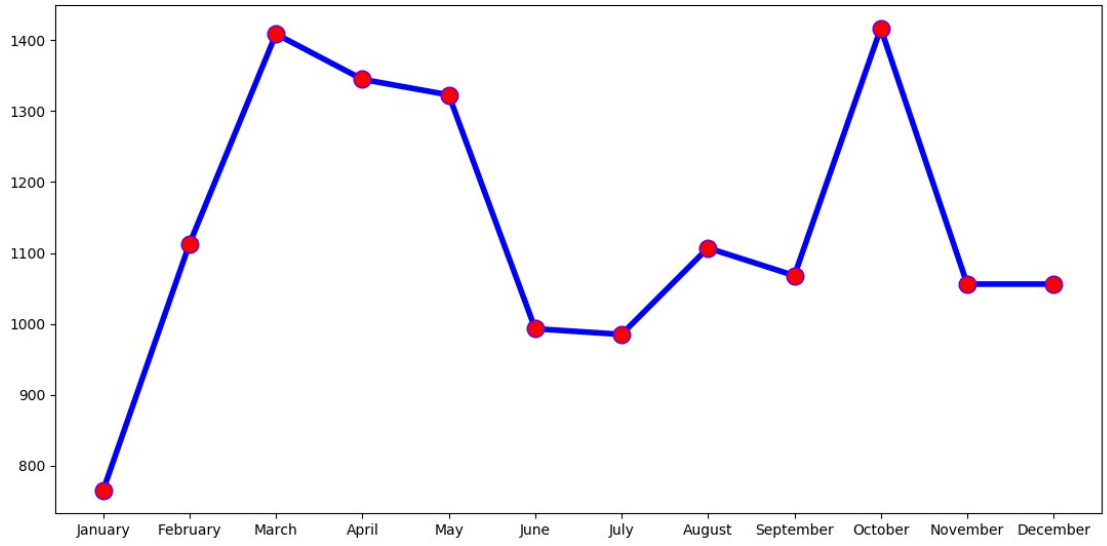
17 % khách sạn được hỏi thì có đến 75 % khách của họ quay trở lại khách sạn hoặc quay lại tại 1 cơ sở khác của khách sạn.

- 8 % khách sạn được hỏi thì có đến 100 % khách của họ quay trở lại khách sạn hoặc quay lại tại 1 cơ sở khác của khách sạn.

Và một trong những tiện nghi được đa số khách hàng yêu cầu bắt buộc phải có đối với phòng của họ book là phải có điều hòa không khí (2015). Cũng trong năm 2015 thì có khoảng 53%

khách của họ sẽ phàn nàn hoặc không ở nếu họ phát hiện phòng có mùi thảm khó chịu. (theo Statista Research Department)

Mặt khác, thì thực tế khách du lịch thường sẽ đi du lịch theo từng mùa và từng tháng. Sau cứ 4 năm thì sẽ có 1 lần mùa World cup được tổ chức tại 1 quốc gia hoặc 2 quốc gia nếu là đồng tổ chức, Vậy việc lượng khách đi du lịch và tham gia xem mùa hội bóng đá lớn nhất thế giới sẽ rất lớn. Cũng tương tự như vậy hàng năm thì các nước sẽ thường tổ chức festivals thu hút rất nhiều khách tham quan và du lịch.



Hình 2.2 : Thể hiện lượt khách qua từng tháng của khách sạn Resort Hotel ở Bồ Đào Nha năm 2016 [2]

dựa trên biểu đồ thì hoàn toàn trả lời được câu hỏi tháng nào là tháng bận nhất trong năm .

***) Nhận Xét:**

- **Vậy nếu chúng ta lưu lại các dữ liệu thông tin của khách hàng chúng ta sẽ trả lời được các câu hỏi giúp chúng ta có thể ra quyết định trong việc quản lí khách sạn cũng như ra quyết định quản lí khinh doanh của nhà hàng:**
  - ✓ **Khách Thường được đến từ đâu ?**
  - ✓ **Số tiền mà Khách Hàng trả tiền sử dụng dịch vụ trong 1 đêm là bao nhiêu ?**
  - ✓ **Tháng nào là Khách Sạn là bận nhất ?**
  - ✓ **Mọi người sẽ ở lại khách sạn trong bao lâu ?**
  - ✓ **v.v**
- **Nếu chúng ta có thể dự đoán được Hàng Khách có thật muốn nhận phòng sau khi book phòng dựa trên thông tin khách hàng trên Booking hay không? điều này giúp chúng ta thống kê được lượng khách của khách sạn trong tương lai. Từ giúp khách sạn chuẩn bị được kịch bản và những dịch vụ, thực phẩm, nhân viên , v.v … để tăng chất lượng dịch vụ khách sạn đưa uy tín khách sạn tăng lên và góp phần giúp khách sạn giữ được khách vào những dịp họ quay lại.**

Phát Biểu Bài Toán: dự đoán Khách Hàng sẽ ckeckin hay cancel dựa trên thông tin của khách hàng trên booking .

- đầu vào: thông tin khi khách hàng booking khác hàng
- đầu ra : yes or no

Dữ Liệu bài toán được lấy từ Hotel Booking Demand Datasets

3. Phân tích dữ liệu và lựa chọn feature.

3.1. Mô tả dữ liệu

Datasets bao gồm hai tập dữ liệu liên quan đến nhu cầu đặt phòng khách sạn. Khách sạn 1 (H1) là một khách sạn ở một khu nghỉ dưỡng và cái còn lại là một khách sạn ở trong thành phố (H2). Cả hai bộ dữ liệu đều có chung cấu trúc, với 31 biến mô tả 40060 quan sát của H1 và 79330 quan sát H2. Mỗi quan sát đại diện cho một đặt phòng khách sạn. Cả hai bộ dữ liệu đều bao gồm các yêu cầu đặt phòng đến từ ngày 1 tháng 7 năm 2015 đến ngày 31 tháng 8 năm 2017, bao gồm các đặt phòng đã được nhận và cả các đặt phòng đã bị hủy. Vì đây là dữ liệu thực của khách sạn, tất cả các yếu tố dữ liệu liên quan đến nhận dạng khách sạn hoặc chi phí đã bị xóa. Do sự khan hiếm dữ liệu kinh doanh thực sự cho mục đích khoa học và giáo dục, các bộ dữ liệu này có thể có vai trò quan trọng đối với nghiên cứu và giáo dục trong quản lý doanh thu, học máy hoặc khai thác dữ liệu, cũng như trong các lĩnh vực khác.

Bảng mô tả:

Subject area	Quản lý khách sạn
More specific subject area	Quản lý doanh thu
Type of data	Tệp văn bản và đối tượng R
How data was acquired	Khai thác từ hệ thống quản lý khách sạn
Data format	Hỗn hợp
Experimental factors	Một số biến được thiết kế từ các biến khác từ các bảng cơ sở dữ liệu khác nhau. Thời gian cho mỗi lần quan sát được xác định là ngày trước ngày đến.
Experimental features	Data was extracted via TSQL queries executed directly in the hotels' PMS databases and R was employed to perform data analysis Data was extracted via TSQL queries executed directly in the hotels' PMS databases and R was employed to perform data analysis Dữ liệu được trích xuất thông qua các truy vấn T SQL được thực hiện trực tiếp trong cơ sở dữ liệu PMS của khách sạn và R được sử dụng để thực hiện phân tích dữ liệu
Data source location	Cả hai khách sạn đều nằm ở Bồ Đào Nha: H1 tại khu nghỉ mát Algarve và H2 tại thành phố Lisbon
Data accessibility	Dữ liệu được cung cấp cùng tài liệu

Dữ liệu:

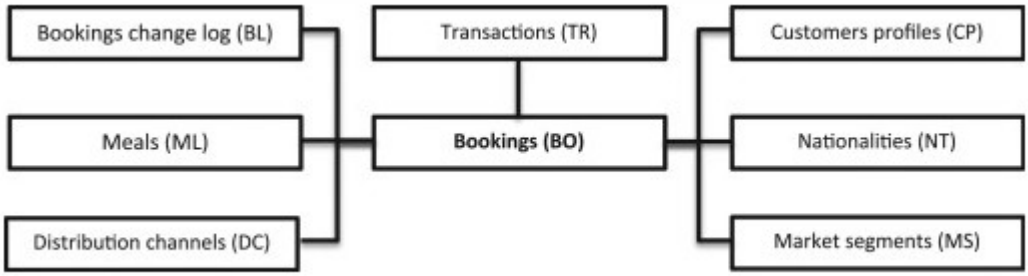
Trong các ngành liên quan đến du lịch, hầu hết các nghiên cứu về các vấn đề dự báo nhu cầu quản lý doanh thu sử dụng dữ liệu từ ngành hàng không, ở định dạng được gọi là Bản ghi tên hành khách (PNR). Đây là một định dạng được phát triển bởi ngành hàng không. Tuy nhiên, các ngành du lịch và du lịch còn lại như khách sạn, du lịch trên biển, công viên giải trí, v.v., có những yêu cầu và đặc thù khác nhau không thể khám phá đầy đủ nếu không có dữ liệu cụ thể của ngành. Do đó, hai bộ dữ liệu khách sạn với dữ liệu nhu cầu được chia sẻ để giúp khắc phục hạn chế này.

Các bộ dữ liệu hiện có sẵn đã được thu thập nhằm mục đích phát triển các mô hình dự đoán để phân loại đặt phòng khách sạn có khả năng bị hủy bỏ. Tuy nhiên, do đặc điểm của các biến có trong các bộ dữ liệu này, việc sử dụng chúng vượt xa vấn đề dự đoán hủy bỏ này. Một trong những tính chất quan trọng nhất trong dữ liệu cho các mô hình dự đoán là không thúc đẩy rò rỉ thông tin trong tương lai. Để ngăn điều này xảy ra, dấu thời gian của biến mục tiêu phải xảy ra sau các biến đầu vào Dấu thời gian. Do đó, thay vì trích xuất trực tiếp các biến từ bảng cơ sở

dữ liệu đặt chỗ, khi có sẵn, các giá trị của biến được trích xuất từ nhật ký thay đổi đặt chỗ, với dấu thời gian liên quan đến ngày trước ngày đến (đối với tất cả các đặt chỗ được tạo trước ngày đến) .

Không phải tất cả các biến trong các bộ dữ liệu này đều đến từ các bảng đặt chỗ hoặc thay đổi bảng cơ sở dữ liệu nhật ký. Một số đến từ các bảng khác và một số được thiết kế từ các biến khác nhau từ các bảng khác nhau. Một sơ đồ trình bày các bảng cơ sở dữ liệu PMS từ đó các biến được trích xuất được trình bày trong. Một mô tả chi tiết của từng biến được cung cấp trong phần sau.

Hình 1. Sơ đồ các bảng cơ sở dữ liệu PMS nơi các biến được trích xuất từ đó.



Thiết kế, tài nguyên và phương pháp

Dữ liệu được lấy trực tiếp từ các cơ sở dữ liệu của PMS tại các máy chủ của Microsoft bằng cách thực hiện truy vấn TSQL trên SQL Server Studio Manager, công cụ môi trường tích hợp để quản lý cơ sở dữ liệu Microsoft SQL. Truy vấn này trước tiên thu thập giá trị hoặc ID (trong trường hợp khóa ngoại) của từng biến trong bảng BO. Bảng BL sau đó đã được kiểm tra xem có sự thay đổi nào liên quan đến ngày trước khi đến không. Nếu một thay đổi được tìm thấy, giá trị được sử dụng là giá trị hiện diện trong bảng BL. Đối với tất cả các biến giữ giá trị trong các bảng liên quan (như bữa ăn, kênh phân phối, quốc tịch hoặc phân khúc thị trường), các giá trị liên quan của chúng đã được truy xuất. Một mô tả chi tiết về các biến được trích xuất, nguồn gốc của chúng và các quy trình kỹ thuật được sử dụng trong việc tạo ra nó được thể hiện trong Bảng 1.

Bảng 1. Mô tả biến.

Biến	**Kiểu**	**Mô tả**	**Nguồn / Kỹ thuật**
ADR	Numeric	Tỷ lệ trung bình hàng ngày	BO, BL và TR / Tính bằng cách chia tổng của tất cả các giao dịch lưu trú cho tổng số đêm lưu trú
Adults	Integer	Số lượng người lớn	BO và BL
Agent	Categorical	ID của công ty du lịch đã đặt phòng	BO và BL
ArrivalDateDayOfMonth	Integer	Ngày đến trong tháng	BO và BL
ArrivalDateMonth	Categorical	Tháng đến với 12 tháng: “Tháng 1” đến “Tháng 12”	BO và BL
ArrivalDateWeekNumber	Integer	Tuần của ngày đến	BO và BL
ArrivalDateYear	Integer	Năm của ngày đến	BO và BL
AssignedRoomType	Categorical	Mã cho loại phòng	BO và BL

		được chỉ định để đặt phòng. Đôi khi loại phòng được chỉ định khác với loại phòng dành riêng do lý do vận hành khách sạn (ví dụ: đặt trước quá nhiều) hoặc theo yêu cầu của khách hàng. Mã được trình bày thay vì chỉ định vì lý do ẩn danh	
Babies	Integer	Số em bé	BO và BL
BookingChanges	Integer	Số lượng thay đổi / sửa đổi được thực hiện đối với đặt phòng kể từ thời điểm đặt phòng được nhập trên PMS cho đến thời điểm nhận phòng hoặc hủy bỏ	BO và BL / Được tính bằng cách thêm số lần lặp duy nhất thay đổi một số thuộc tính đặt phòng, cụ thể là: người, ngày đến, đêm, loại phòng dành riêng hoặc bữa ăn
Children	Integer	Số trẻ em	BO và BL / Tổng của cả trẻ em phải trả và không phải trả
Company	Categorical	ID của công ty / đơn vị thực hiện đặt phòng hoặc chịu trách nhiệm thanh toán đặt phòng. ID được trình bày thay vì chỉ định vì lý do ẩn danh	BO và BL
Country	Categorical	Nước xuất xứ	BO, BL và NT
CustomerType	Categorical	Loại đặt phòng, giả sử một trong bốn loại: Hợp đồng - khi đặt phòng có một khoản giao hoặc loại hợp đồng khác liên quan đến nó; Nhóm - khi đặt phòng được liên kết với một nhóm; Tạm thời - khi đặt phòng không phải là	BO và BL

		một phần của một nhóm hoặc hợp đồng, và không liên quan đến đặt phòng tạm thời khác; Bên tạm thời - khi đặt phòng tạm thời, nhưng được liên kết với ít nhất các đặt phòng tạm thời khác	
DaysInWaitingList	Integer	Số ngày đặt phòng nằm trong danh sách chờ trước khi được xác nhận với khách hàng	BO / Tính bằng cách trừ ngày đặt phòng được xác nhận cho khách hàng kể từ ngày đặt phòng được nhập trên PMS
DepositType	Categorical	Chỉ định nếu khách hàng đã đặt cọc để đảm bảo đặt phòng. Biến này có thể giả sử ba loại: Không có tiền gửi - không có khoản tiền gửi nào được thực hiện; Không hoàn lại tiền - một khoản đặt cọc đã được thực hiện theo giá trị của tổng chi phí lưu trú; Hoàn lại tiền - một khoản đặt cọc đã được thực hiện với một giá trị dưới tổng chi phí lưu trú.	BO và TR / Giá trị được tính dựa trên các khoản thanh toán được xác định cho đặt phòng trong bảng giao dịch (TR) trước ngày đến hoặc ngày hủy của đặt phòng. Trong trường hợp không tìm thấy khoản thanh toán nào, giá trị là không có tiền gửi. Nếu khoản thanh toán bằng hoặc vượt quá tổng chi phí lưu trú, giá trị được đặt là Không Non Hoàn lại tiền. Nếu không, giá trị được đặt là" hoàn tiền"
DistributionChannel	Categorical	Kênh phân phối đặt chỗ	BO, BL và DC
IsCanceled	Categorical	Giá trị cho biết nếu đặt phòng đã bị hủy (1) hay không (0)	BO
IsRepeatedGuest	Categorical	Giá trị cho biết tên đặt phòng có phải từ một khách lặp lại (1) hay không (0)	BO, BL và C / Biến được tạo bằng cách xác minh nếu một hồ sơ được liên kết với khách

			hàng đặt phòng. Nếu vậy, và nếu ngày tạo hồ sơ khách hàng là trước ngày tạo cho đặt phòng trên cơ sở dữ liệu PMS, thì giả sử đặt phòng là từ một khách lặp đi lặp lại
LeadTime	Integer	Số ngày trôi qua giữa ngày nhập của đặt phòng vào PMS và ngày đến	BO và BL / Phép trừ của ngày nhập từ ngày đến
MarketSegment	Categorical	Chỉ định phân khúc thị trường.	BO, BL và MS
Meal	Categorical	Loại bữa ăn đặt. Các hạng mục được thể hiện trong các gói bữa ăn khách sạn tiêu chuẩn: Không xác định / SC - không có gói bữa ăn; BB - Giường & Bữa sáng; HB - Half board (bữa sáng và một bữa ăn khác - thường là bữa tối); FB - Hội đồng quản trị đầy đủ (bữa sáng, bữa trưa và bữa tối)	BO, BL và ML
PreviousBookingsNotCanceled	Integer	Số lượng đặt phòng trước đó không bị hủy bởi khách hàng trước khi đặt phòng hiện tại	BO và BL / Trong trường hợp không có hồ sơ khách hàng liên quan đến đặt phòng, giá trị được đặt thành 0. Nếu không, giá trị là số lượng đặt phòng có cùng hồ sơ khách hàng được tạo trước khi đặt phòng hiện tại và không bị hủy.
PreviousCancellations	Integer	Số lượng đặt phòng trước đó đã bị hủy bởi khách hàng trước khi đặt phòng hiện tại	BO và BL / Trong trường hợp không có hồ sơ khách hàng liên quan đến đặt phòng, giá trị

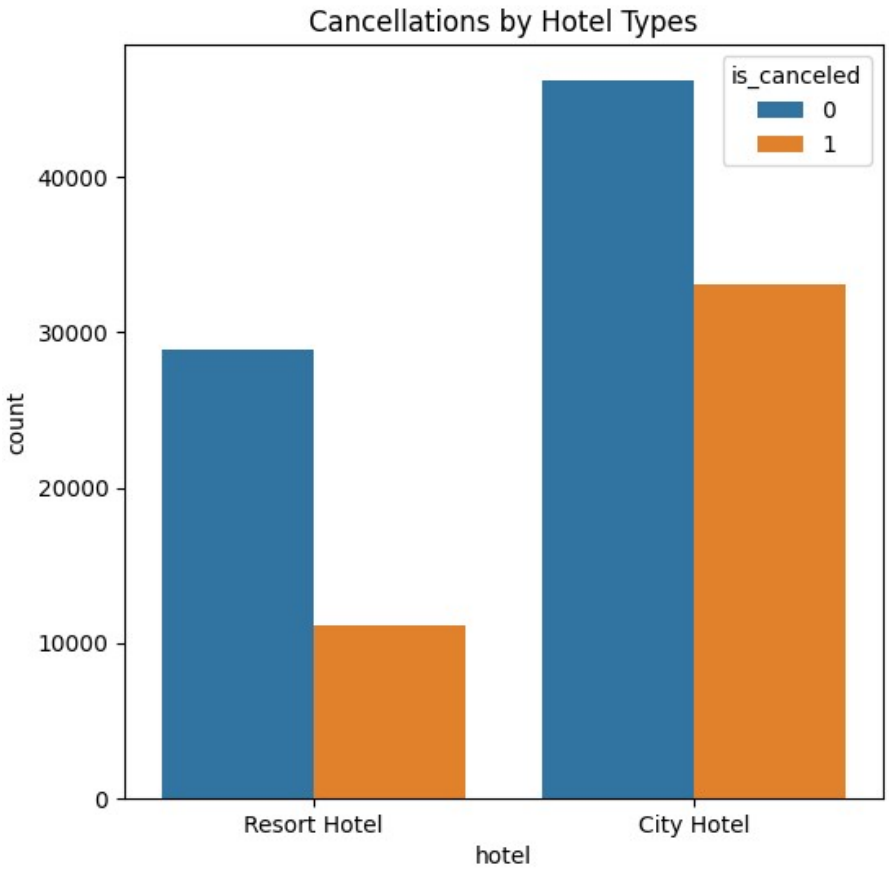
			được đặt thành 0. Nếu không, giá trị là số lượng đặt phòng có cùng hồ sơ khách hàng được tạo trước khi đặt phòng hiện tại và bị hủy.
RequiredCardParkingSpaces	Integer	Số lượng chỗ đậu xe ô tô theo yêu cầu của khách hàng	BO và BL
ReservationStatus	Categorical	Trạng thái đặt phòng cuối cùng, giả sử một trong ba loại: Hủy bỏ - đặt phòng đã bị hủy bởi khách hàng; Trả phòng - khách hàng đã đăng ký nhưng đã khởi hành; No-Show - khách hàng không nhận phòng và đã thông báo cho khách sạn về lý do tại sao	BO
ReservationStatusDate	Date	Ngày mà trạng thái cuối cùng được đặt. Biến này có thể được sử dụng cùng với DepositStatus để hiểu khi nào đặt phòng bị hủy hoặc khi nào khách hàng trả phòng khách sạn	BO
ReservedRoomType	Categorical	Mã loại phòng dành riêng. Mã được trình bày thay vì chỉ định vì lý do ẩn danh	BO và BL
StaysInWeekendNights	Integer	Số đêm cuối tuần (thứ bảy hoặc chủ nhật) khách lưu trú hoặc đặt phòng tại khách sạn	BO và BL / Tính bằng cách đếm số đêm cuối tuần từ tổng số đêm
StaysInWeekNights	Integer	Số đêm trong tuần (Thứ Hai đến Thứ Sáu) khách lưu trú hoặc đặt phòng để ở tại khách sạn	BO và BL / Tính bằng cách đếm số đêm trong tuần từ tổng số đêm
TotalOfSpecialRequests	Integer	Số lượng yêu cầu đặc biệt được thực hiện bởi khách hàng (ví	BO và BL / Tổng của tất cả các yêu cầu đặc biệt

		dụ: giường đôi hoặc tầng cao)	

PMS đảm bảo không có dữ liệu bị thiếu trong các bảng cơ sở dữ liệu của nó. Tuy nhiên, trong một số biến phân loại như Đại lý hoặc Công ty, thì NULL 'được trình bày như một trong những danh mục. Điều này không nên được coi là một giá trị còn thiếu, mà là vì không áp dụng được. Ví dụ: nếu đặt phòng thì Đặc vụ Đặc quyền là Định nghĩa là NU NU, có nghĩa là việc đặt phòng không đến từ đại lý du lịch.

3.2. Phân tích dữ liệu.

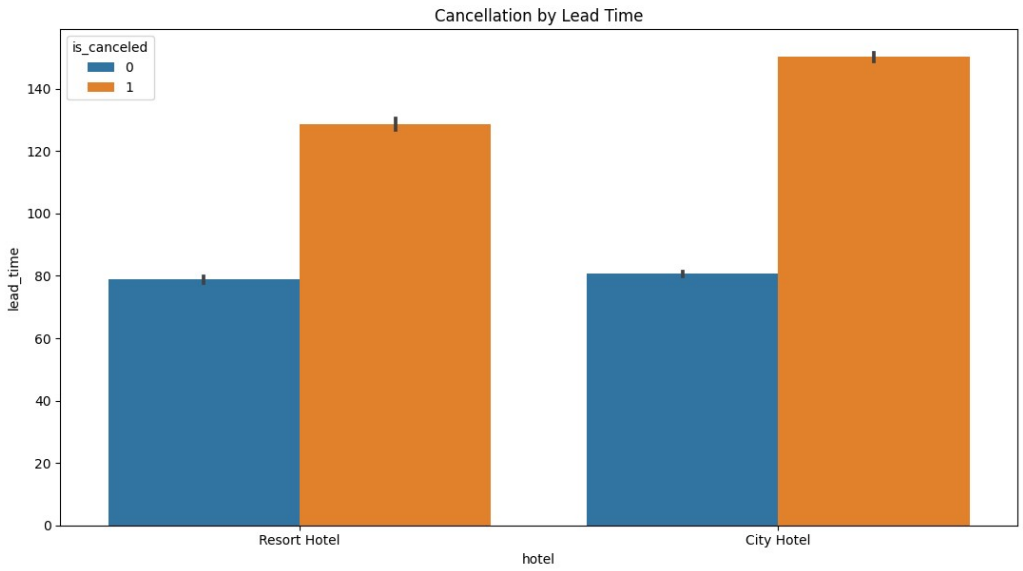
Dữ Liệu gồm có tất cả là 32 cột và 119390 bản ghi từ 2 khách sạn Resort Hotel và City Hotel. Cả 2 khách sạn đều nằm tại Bồ Đào Nha .



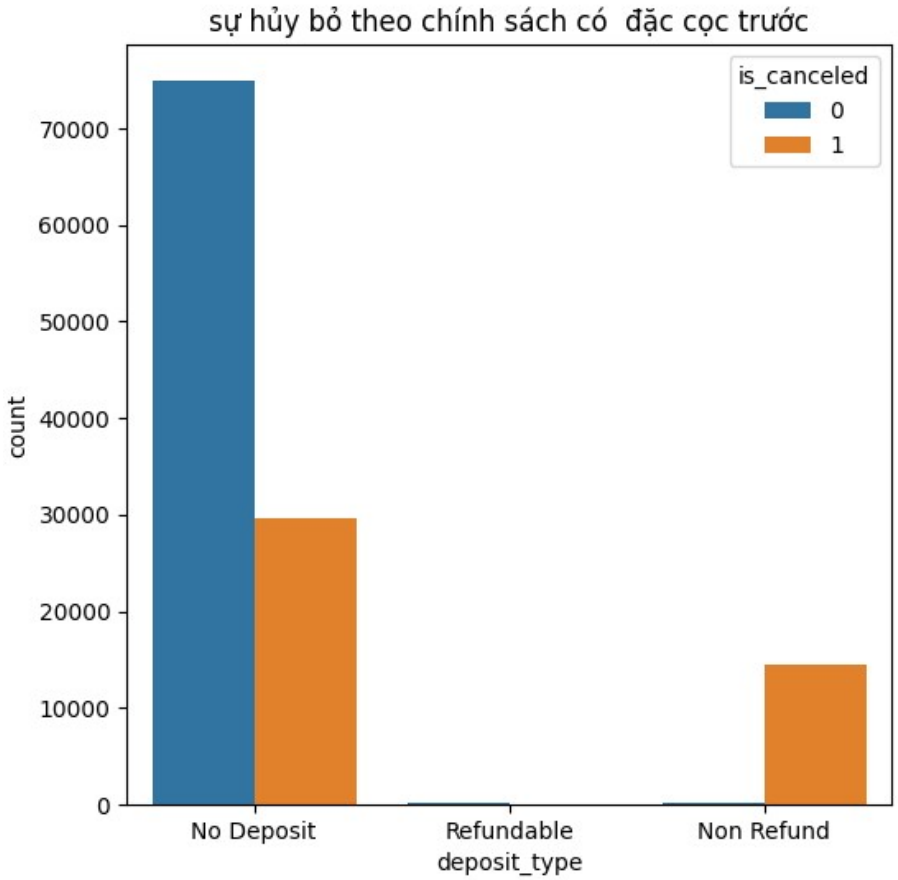
Hình 3.2.1 thể hiện lượng khách nhận phòng và hủy phòng của 2 khách sạn.

Dựa vào hình 3.2.1 có thể thấy rằng chỉ số hủy phòng của khách sạn Resort Hotel là 27,98 %. Trong khi đó chỉ số hủy phỏng của City Hotel là 41,77 % .

Độ phụ thuộc của nhận phòng và trả phòng theo Số ngày trôi qua giữa ngày nhập của đặt phòng vào PMS và ngày đến. Theo như hình 3.2.2 thì thời gian chờ nhận phòng càng dài thì khả năng cao là phòng bị hủy càng cao .



Hình 3.2.2 . Thống kê booking theo Lead Time



Hình 3.2.3 Thể hiện sự hủy phòng theo kiểu chính sách phòng loại đặt cọc trước.

Các thuộc tính hay các cột đều mang một giá trị riêng ta lần lượt vẽ những biểu đồ thể hiện mối quan hệ gữa sự hủy bỏ và các giá trị trong thuộc tính đó. Từ đó sẽ giúp các chuyên gia

nhận định 1 cách khách quan nhất về tính độc lập và phụ thuộc của nhãn vào thuộc tính đó. Ví dụ nếu sự phân bố của của các nhãn đối với các giá trị trong thuộc tính đó thì ta có thể suy ra được nhãn độc lập tuyến tính đối với tập dữ liệu đã cho với thuộc tính trên. Như hình 3.2.3 thì có thể thấy sự hủy bỏ phụ thuộc vào chính sách đặt cọc trước đặc biệt là Non Refund tỉ lệ hủy bỏ phòng cao hẳn.

3.3. Lựa chọn feature

Đầu Tiên chúng ta sẽ phải loại bỏ những cuộc mà khi khách hàng sử dụng phòng hoặc nhận phòng xong chúng ta mới có dữ liệu là:

- is_canceled
- meal
- *StaysInWeekNights*
- *StaysInWeekendNights*
- ADR
- *ReservationStatus*

Tiếp Theo ta loại bỏ những cột mang thông tin cá nhân và không liên quan gì đến chất lượng dịch vụ hay lịch sử dụng dịch vụ:

- *ReservedRoomType*.

Tiếp tục ta loại bỏ những cột mà quá nhiều dữ liệu null :

- Company

cuối cùng ta loại bỏ những dữ liệu không cần thiết cho bài toán phân loại theo ý kiến của các ý kiến chuyên gia. Nhưng ở đây

4. Cơ sở lí thuyết.

4.1. Các kĩ thuật transform dữ liệu.

4.1.1. Chuyển đổi attributes mà thuộc dạng là object.

Các cột thuộc tính mà dữ liệu cung cấp có cột đã được biểu diễn dưới dạng số nhưng có cột thì lại được được biểu diễn dưới dạng object tức dạng đối tượng . vậy chúng ta phải chuyển đổi sang dạng số để máy tính hiểu được và thực hiện các thuật toán. Một attriubute gồm có 4 loại chính đó là:

- ✓ Nominal
- ✓ Ordinal
- ✓ Interval
- ✓ ratio

Đối với dạng dữ liệu thuộc Nominal ví dụ như cột loại phòng thì tiểu luận xin đề xuất sử dụng one-hot coding. Đối với dạng dữ liệu dạng Ordinal thì tiểu luận sử dụng đánh số từ 0 cho đến rank cao nhất đối với thuộc tính đó . dữ liệu thuộc dạng interval và ratio tiểu luận không chuyển đổi để tránh mất thông tin.

4.1.2. Loại bỏ những dữ liệu miss value.

Trong lúc thu thập thông dữ liệu 1 việc khó tránh khỏi đó việc thu thập thiếu hoặc giá trị thu được quá lớn so với giá trị mà máy tính thu được người ta gọi tắt đó là miss value. Trong đề tài lần này tiểu luận in được đề xuất thay thế tất cả những dữ liệu miss value bằng giá trị trung bình của các thuộc tính mà không bị miss value.

4.1.3. Normalize dữ liệu.

Sau khi chuyển đổi dữ liệu sang hết dạng số thì việc không thể tránh khỏi đó là việc các thuộc tính các khoảng thuộc tính sẽ bị chênh nhau quá lớn . Ví dụ như 1 thuộc tính có range là (0,10) nhưng có thuộc tính có range là (-100,1000000) vì vậy nó ảnh hưởng đến rất nhiều đến phần trainning model cũng như khả năng dự đoán của model đặc biệt các model sử dụng điểm dữ liệu để cập nhật paramater .

Phương pháp normalize dữ liệu mà tiểu luận sử dụng đó là Min Max Scaler. Min Max Scaler nó được mô tả theo công thức sau:

$$atribute\_normalizer = \frac{atribute - Min(atribute)}{Max(atribute) - Min(atribute)}$$

4.2. Thuật toán naive bayes.

Phân loại Bayes đơn giản được dùng trong trường hợp mỗi ví dụ được cho bằng tập các thuộc tính $(x_1, x_2, \ldots x_n)$ và cần xác định nhãn phân loại y trong một tập nhãn hữu hạn C.

Trong giai đoạn huấn luyện, dữ liệu huấn luyện được cung cấp dưới dạng các mẫu (x_i, y_i). Sau khi huấn luyện xong, bộ phân loại cần dự đoán nhãn cho mẫu mới x.

Nhãn phân loại được xác định bằng cách tính xác xuất điều kiện của nhãn khi quan sát tổ hợp giá trị thuộc tính$(x_1, x_2, \ldots x_n)$. Thuộc tính được chọn, ký hiệu là C_{MAP} là thuộc tính có xác xuất điều kiện cao nhất.

$$y = C_{MAP} = \underset{c_i \in C}{\text{argmax}} = P(c_i \mid x_1, x_2, ..., x_n)$$

Sử dụng quy tắc Bayes công thức được viết lại như sau:

$$C_{MAP} = \underset{c_j \in C}{argmax} \frac{P(x_1, x_2, \ldots x_n | c_j) P(c_j)}{P(x_1, x_2, \ldots x_n)}$$

Mà P($x_{1,} x_2, \ldots, x_n$) không phụ thuộc vào c_j nên không ảnh hưởng tới giá trị của C_{MAP} vì vậy ta có thể bỏ mẫu số và viết lại như sau:

$$C_{MAP} = \underset{c_j \in C}{argmax}\ P¿¿$$

Hai thành phần của biểu thức trên được tính từ dữ liệu huấn luyện. Giá trị $P(c_j)$ được tính bằng tần suất quan sát thấy nhãn c_j trên tập huấn luyện, tức là bằng số mẫu có nhãn là c_j chia cho tổng số mẫu. Việc tính $P(x_1, x_2, \ldots x_n | c_j)$ khó khăn hơn nhiều. Vấn đề là số tổ hợp giái trị của n thuộc tính cùng với nhãn phân loại là rất lớn khi n lớn. Để tính xác suất này chính xác, mỗi tổ hợp giá trị của thuộc tính phải xuất hiện cùng nhãn phân loại đủ nhiều, trong khi số mẫu huấn luyện thường khoong đủ lớn.

Để giải quyết vấn đề trên, ta giả sử các thuộc tính là độc lập xác suất với nhau khi biết nhãn phân loại c_j. Trên thực tế, các thuộc tính thường không độc lập với nhau như vậy. Chính vì dựa trên giả thiết độc lập xác suất đơn giản như vậy nên phương pháp có tên gọi Bayes đơn giản. Trên thực tế, phân loại Bayes đơn giản có độ chính xác tốt trong rất nhiều ứng dụng.

Với giả thiết tính độc lập xác suất có điều kiện ta có thể viết như sau:

$$P(x_1, x_2, \ldots x_n | c_j) = \text{P}(x_1 \vee c_j)\text{P}(x_2 \vee c_j)\ldots\text{P}(x_n \vee c_j ¿$$

Tức là xác suất đồng thời quan sát thấy các thuộc tính bằng tích xác suất điều kiện của từng thuộc tính riêng lẻ. Thay vào biểu thức ở trên, ta được bộ phân loại Bayes đơn giản (có đầu ra ký hiệu là C_{BN}) như sau.

$$C_{BN} = \underset{c_j \in C}{argmax}\ P(¿ c_j) \prod_i P(¿ x_i \vee c_j) ¿¿$$

Trong đó P($x_i \vee c_j$) được tính từ dữ liệu huấn luyện bằng tần số x_i xuất hiện cùng với c_j chia cho số lần c_j xuất hiện. Việc tính xác suất này đòi hỏi ít dữ liệu hơn nhiều so với tính $P(x_1, x_2, \ldots x_n | c_j)$

Huấn luyện

Quá trình huấn luyện hay học Bayes đơn giản là quá trình tính các xác suất P(c_j) và các xác suất điều kiện P($x_i \vee c_j$) bằng cách đếm trên tập dữ liệu huấn luyện. Các xác suất P(c_j) và các xác suất điều kiện P($x_i \vee c_j$) được tính trên tập dữ liệu huấn luyện theo công thức sau:

$$\text{P}(c_j) = \frac{số\ mẫu\ có\ nhãn\ là\ c_j}{Tổng\ số\ mẫu\ trong\ tập\ huấn\ luyện}$$

$$P(x_i \vee c_j) = \frac{Số\ mẫu\ có\ giá\ trị\ thuộc\ tính\ X_i = x_i\ và\ nhãn\ là\ c_j}{số\ mẫu\ có\ nhãn\ là\ c_j}$$

Trong trình bày trên, ta mới đề cập tới tính độc lập xác suất giữa các đặc trưng mà chưa xét tới dạng phân bố xác suất cụ thể của từng đặc trưng, tức là dạng phân bố của $P(x_i \vee c_j)$, ngoài ra ta cũng chưa xét trường hợp đặc trưng nhận giá trị liên tục. Một dạng phân bố xác suất thường dùng với Bayes đơn giản, bao gồm các phân bố liên tục là phân bố Gauss.

Trong trường hợp thuộc tính nhận giá trị liên tục, người ta thường giả sử giá trị đặc trưng liên quan tới mỗi nhãn phân loại tuân theo phân bố Gauss và sử dụng phân bố này để biểu diễn. Mô hình này được gọi là Bayes đơn giản Gauss (Gaussian naive Bayes). Cụ thể với mỗi thuộc tính liên tục x_i, trước tiên ta phân chia dữ liệu theo các phần theo giá trị của nhãn phân loại. Tiếp theo ta tính giá trị trung bình μ_y và phương sai σ_y^2 cho các giá trị của thuộc tính x_i gắn với nhãn phân loại y. Xác suất thuộc tính x_i nhận giá trị v được tính bằng cách thay v vào biểu thức phân bố Gauss với giá trị trung bình và độ lệch chuẩn tính được ở trên:

$$P(x_i = v \mid y ¿ = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(\frac{-(v-\mu_y)^2}{2\sigma_y^2}\right)$$

4.3. Thuật toán logistic regression.

4.3.1. Linear regression.

4.3.1.1. Giới thiệu bài toán.

Một trong những model (mô hình) đơn giản nhất của bài toán hồi quy (regression) là hồi quy tuyến tính (linear regression). Mô hình hồi quy tuyến tính là một mô hình liên quan đến sự kết hợp tuyến các đầu vào sao cho:

$$y \approx f(x, \mathrm{w}) = w^t x = w_0 + w_1 x_1 + w_2 x_2 + .. + w_D x_D$$

Trong đó:

- $\mathrm{w} = (w_0, w_1, w_2, ..., w_D)^T$ là vector weight (hay thường được gọi là paramaters)
- $x = (1, \mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_D)^T$ là vector đầu vào.

Trong một số sách thì được viết: $y \approx f(x, \mathrm{w}) = w^T \phi(x)$ trong đó $\phi(x)$ là 1 hàm transform đầu vào của x mục đích để biến tập dữ liệu không dự đoán bằng linear thành tập dữ liệu có thể dự đoán được bằng hàm linear.

4.3.1.2. Xây dựng bài toán:

Cũng giống như các bài toán supervide linear regression cũng đi tìm những tham số cho model chi tiết là vector weight sao cho

$$y \approx \hat{y} = f(x, \mathrm{w}) = w^T x = w_0 + w_1 x_1 + w_2 x_2 + .. + w_D x_D$$

Để tìm ra paramaters ta cần 1 phép đánh giá tính phù hợp của tham số đối với bài toán. Với bài toán regression nói chung thì chúng ta đang muốn sự sai khác e giữa giá trị thực y và giá trị dự đoán $\hat{y}$ là nhỏ nhất. Nói cách khác, chúng ta đang muốn giá trị sau đây càng nhỏ càng tốt. Ta gọi đó là hàm Lossfuntion:

$$L(\mathrm{w}) = \frac{1}{2N} \sum_{i=1}^{N} (y_i - w^T x_i)^2 = \frac{1}{2N} \left\| y - (w^T X)^T \right\|_2^2 = = \frac{1}{2N} \left\| y - X^T w \right\|_2^2$$

Với $y = [\mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, ..., \mathrm{y}_N]$.

$$w = \text{argmin}_w L(w)$$

Để hàm lossfuntion nhỏ nhất theo w thì nghiệm ta cần tìm là . Chúng ta có thể sử dụng phương pháp đạo hàm và giải nghiệm bằng cách cho đạo hàm bằng 0. Nhưng tiểu luận xin phép trình bày về phương pháp Stochastic Gradient Descent (SGD).

thuật toán SGD với linear regression:

Khởi Tạo:

X - ma trận đầu vào

y - mục vector mục tiêu Datas = (X,y)

lr - learning rate

Lặp:

For epoch from 0 to epochs:

Datas = shuffle(Datas)

for x,y in Datas:

$$w \leftarrow w - lr * \nabla_w L(w, x, y)$$

endfor

endfor

return:

return w

với : $\Delta_w L(w, x, y) = x(x^T w - y)$ và shuffle(Datas) là hàm xáo trộn lại cặp (input,output) để đảm bảo tính ngẫu nhiên. Việc này cũng ảnh hưởng tới hiệu năng của SGD. Đây cũng chính là lý do thuật toán này có chứa từ stochastic (ngẫu nhiên).

4.3.2. Logistic regression

thuật toán loggistic regression là thuật toán thuộc nhóm binary classification. trong bài toán phân loại nhị phân mô hình xác xuất là được đánh giá cao nhất. với output sẽ được biểu diễn là y = 1 với nhãn thuộc positive và y = 0 đối với nhãn negative. Do đó mô hình sẽ đi tính $p(y=1/x,\theta)$ vậy nhiệm vụ của logistic regression là đi tìm bộ tham số θ dựa trên tập training sao cho :

$y \approx p(y/x,\theta)$

a) tìm $p(y/x,\theta)$

Theo bayes ta có :

$$p(y=1/x,\theta)=\frac{p(x,\theta/y=1)\,p(y=1)}{p(x,\theta)}$$

$$p(y=1/x,\theta)=\frac{p(x,\theta/y=1)*p(y=1)}{p(x,\theta/y=1)\,p(y=1)+p(x,\theta/y=0)\,p(y=0)}$$

$$p(y=1/x,\theta)=\frac{p(x,\theta/y=1)}{p(x,\theta/y=1)+p(x,\theta/y=0)\frac{N_0}{N_1}}$$

$$p(y=1/x,\theta)=\frac{1}{1+\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1}}$$

Nhận xét : $\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1}\in(0,+\infty)$ vậy nên ta biến đổi căn bản:

$$\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1}=\exp\left(\ln\left(\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1}\right)\right)$$

ta đặt : $\ln\left(\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1}\right)=z$ vậy ta có: $p(y=1/x,\theta)=\frac{1}{1+e^z}$

b) xây dựng hàm lossfuntion.

nhận xét : $z=\ln\left(\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1}\right)\in(-\infty,+\infty)$ vậy nhiệm vụ mới chúng ta là từ đầu vào x làm sao dự đoán được $z=\ln\left(\frac{p(x,\theta/y=0)}{p(x,\theta/y=1)}\frac{N_0}{N_1}\right)$ bằng bao nhiêu ? vậy kết luận lại là bài toán quay lại là bài toán regression vậy nên thuật toán có tên logistic regression. Để đơn giản hóa bài thuật toán ta sử dụng linear regression để dự đóan z hay

$$z \approx w^T x+b \Rightarrow p(y=1/x,\theta)\approx\frac{1}{1+e^{w^T x+b}}$$

- Độ bất định: trong lí thuyết thông tin thì độ bất định là độ đo tính mập mờ của 1 sự kiện. độ bất định càng cao thì độ mập mờ của thông tin càng cao hay độ bất định

càng thấp thì thông tin càng có ý nghĩa . độ bất định có quan hệ mật thiết với xác suất.

- các tính chất của độ bất định:

  - Nếu sự kiện X chỉ có 1 trường hợp sảy ra là x suy ra p(x) = 1 kéo theo độ bất định d(x) = 0
  - Xác suất sảy ra của 1 trường hợp càng lớn thì độ bất định càng nhỏ .
  - Và số trường hợp sảy ra càng lớn thì trung bình độ bất định cả sự kiện càng lớn.

Công thức tính độ bất định là : $d(x) = -\ln(p(x))$ trong đó p(x) là xác suất để trường hợp x sảy ra trong tập trường hợp X.

- như phần a đã trình bày thì chúng đang cố gắng tạo ra 1 model sao cho $y \approx p(y/x, \theta)$ ta sử hàm lossfunction là tổng độ bất định của tất cả các phép thử :

$$L(x,w) = \sum_{x \in C_1} \ln(p(y=1/x,\theta)) + \sum_{x \in C_2} \ln(1 - p(y=1/x,\theta))$$

hay lossfunction thường được viết :

$$L(x,w,y) = \sum_{i=0}^{N} y_i \ln(p(x_i) + (1-y_i)\ln(1-p(x_i)))$$

4.4. Thuật toán cây quyết định.

4.4.1. Định nghĩa

Decision Tree là một thuật toán thuộc loại Supervised Learning, phương pháp học có giám sát, kết quả biến mục tiêu của Decision Tree chủ yế là biến phân loại. Các thuật toán được xây dựng giống hình dạng một các cây có ngọn cây, thân cây, lá cây kết nối bằng các cành cây, và mỗi thành phần đều có ý nghĩa riêng của nó, như các yếu tốc tác động lên quyết định sau cùng.

Một cây quyết định bao gồm:

- 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 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).

4.4.2. Thuật toán ID3

ID3 (J. R. Quinlan 1993) sử dụng phương pháp tham lam tìm kiếm từ trên xuống thông qua không gian của các nhánh có thể không có backtracking. ID3 sử dụng Entropy và Information Gain để xây dựng một cây quyết định.

Entropy trong cây quyết định(Decision Tree)

Emtropy là thuật ngữ thuộc Nhiệt động lực học, là thước đo của sự biến đổi, hỗn loạn hoặc ngẫu nhiên. Năm 1984, Shannon đã mở rộng khái niệm Emtropy sang lĩnh vực nghiên cứu, thống kê với công thức sau:

Với một phân phối xác xuất của một biến rời rạc x có thể nhận n giá trị khác nhau x1,x2,...,xn.

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

Ký hiệu phân phối này là $p=(p_1 ,p_2 ,\ldots,p_n)$. Entrpy của phân phối này được định nghĩa là:

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

Information Gain trong cây quyết định(Decision Tree)

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 Infomation 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 N_c phần tử thuộc lớp c cho trước:

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

•**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 $S_1, S_2, \ldots, S_K$ với số điểm trong mỗi child node lần lượt là $m_1, m_2 ,\ldots, m_K$, ta có:

$$H(x, S) = \sum_{k=1}^{K} (m_k / N) * H(S_k)$$

Bước 3: Chỉ số Gain Information được tính bằng:

$G(x, S) = H(S) – H(x,S)$

4.4.3 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):

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

Trong đó: Split Info được tính như sau:

$$-\sum_{i=1}^{n} Di\ log_2 Di$$

Giả sử chúng ta phân chia biến thành n nút cón và Di 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.

4.4.4 Ưu/nhược điểm của thuật toán cây quyết định

Ưu điểm

Cây quyết định là một thuật toán đơn giản và phổ biến. Thuật toán này được sử dụng rộng rãi bởi những lợi ích của nó:

- 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 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 làm việc với dữ liệu lớn
- Cây quyết định ít ảnh hưởng với dữ liệu ngoại lệ (outliers).
- Dễ dàng chuyển sang luật ra quyết định (Decesion rule)

Nhược điểm

Kèm với đó, cây quyết định cũng có những nhược điểm cụ thể:

- 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(Là hiện tượng mô hình ghi nhớ quá tốt dữ liệu huấn luyện và phụ thuộc vào nó, việc này khiến cho mô hình không thể tổng quát hóa các quy luật để hoạt động với dữ liệu chưa từng được chứng kiến.)

Cách làm giảm Overfit của cây quyết định:

Dùng phương pháp “Stopping Criteria” yếu tố ngừng phân nhánh với Pruning method, phương pháp” ngắt cành sao sao thuật toán Decision trees mang lại kết quả phân loại tối ưu

Lưu ý:

- Mục đích của quá trình phân tích dữ liệu hay huấn luyện mô hình phân tích là để làm sao khi áp dụng cho bộ dữ liệu thực tế chhusng đem lại kết quả chính xác nhất chứ không phải tập trung vào dữ liệu training.
- Không phải phương pháp Stopping criteria hay Pruning lúc nào cũng đem lại hiệu quả, do đó bất kể mô hình nào thì chúng ta cũng phải sử dụng các phương pháp đánh giá(Classification evaluation method) để kiểm tra và đưa ra những điều chỉnh kịp thời

Phương pháp Stopping criteria có thể kể đơn giản như các phương pháp hạn chế kích thước hay chiều sâu của cây quyết định bao gồm giới hạn, hay cung cấp số lượng tập con, hay số lượng mẫu(sample) tối thiểu cho một lần phân nhánh từ một node, nhưng node không có phân nhánh tiếp theo(termial node) hay giới hạn tối đa số thuộc tính được dùng để phân nhánh.

4.5 Thuật toán KNN

Định nghĩa KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới. Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

Các bước trong KNN

1. Ta có D là tập các điểm dữ liệu đã được gắn nhãn và A là dữ liệu chưa được phân loại.
2. Đo khoảng cách (Euclidian, Manhattan, Minkowski, Minkowski hoặc Trọng số) từ dữ liệu mới A đến tất cả các dữ liệu khác đã được phân loại trong D.
3. Chọn K (K là tham số mà bạn định nghĩa) khoảng cách nhỏ nhất. 4.Kiểm tra danh sách các lớp có khoảng cách ngắn nhất và đếm số lượng của mỗi lớp xuất hiện.
5. Lấy đúng lớp (lớp xuất hiện nhiều lần nhất).
6. Lớp của dữ liệu mới là lớp mà bạn đã nhận được ở bước 5.

Ưu điểm

1. Thuật toán đơn giản, dễ dàng triển khai.
2. Độ phức tạp tính toán nhỏ.
3. Xử lý tốt với tập dữ liệu nhiễu đối với K đủ lớn

Nhược điểm

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

4.6 F1-score

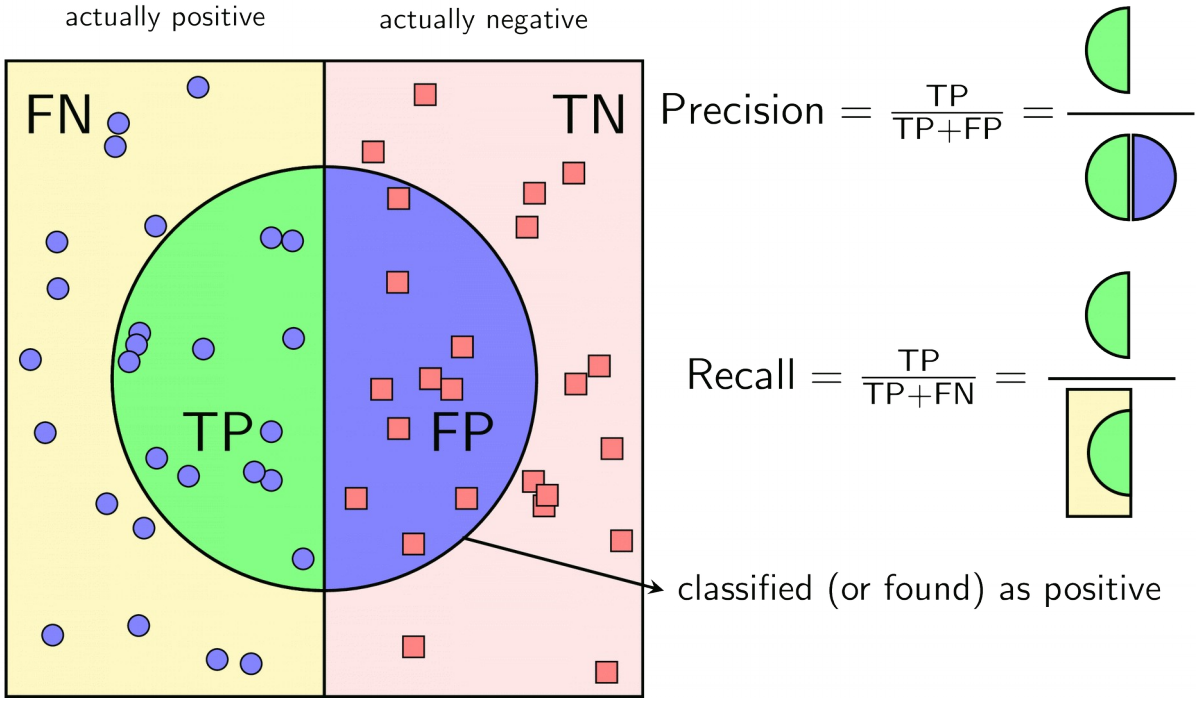
4.6.1 Precision và Recall

Hay còn gọi là **Độ chính xác** và **Độ bao phủ,** được sử dụng trong các loại bài toán phân loại.Trước hết xét bài toán phân loại nhị phân. Ta cũng coi một trong hai lớp là *positive*, lớp còn lại là *negative*.

Ta có ma trận nhầm lẫn(*confusion matrix*)

Lớp c_i		Được phân lớp bởi hệ thống	
		Thuộc	Ko thuộc
Phân lớp thực sự (đúng)	Thuộc	TP_i	FN_i
	Ko thuộc	FP_i	TN_i

Xét Hình 3 dưới đây:



Hình 3: Cách tính Precision và Recall.

Với một cách xác định một lớp là *positive*, **Precision** được định nghĩa là tỉ lệ số điểm **true positive** trong số những điểm **được phân loại là *positive*** (TP + FP).

Recall được định nghĩa là tỉ lệ số điểm **true positive** trong số những điểm **thực sự là *positive*** (TP + FN).

Một cách toán học, Precison và Recall là hai phân số có tử số bằng nhau nhưng mẫu số khác nhau:

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

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

Bạn đọc có thể nhận thấy rằng TPR và Recall là hai đại lượng bằng nhau. Ngoài ra, cả Precision và Recall đều là các số không âm nhỏ hơn hoặc bằng một.

Ý nghĩa của percision và recall:

Precision: *trong tập tìm được thì bao nhiêu cái (phân loại) đúng.*

Recall*: trong số các tồn tại, tìm ra được bao nhiêu cái (phân loại).*

4.6.2 F1-score:

F1-score là 1 phương pháp để đánh giá tính hiệu quả của 1 hệ thống phân lớp

F_1-score, là *harmonic mean* của precision và recall (giả sử rằng hai đại lượng này khác không)

$$F_1=\frac{2}{precision^{-1}+recall^{-1}}=2\frac{precion\cdot recall}{precision+recall}$$

F_1-score có giá trị nằm trong nửa khoảng (0,1]. F_1 càng cao, bộ phân lớp càng tốt. Khi cả recall và precision đều bằng 1 (tốt nhất có thể), F_1=1. Khi cả recall và precision đều thấp

4.7 Những vấn đề hay gặp khi training.

Overfitting

Là hiện tượng mô hình trở lên quá khớp với dữ liệu huấn luyện, mô hình nhớ những dữ liệu huấn luyện, dẫn tới độ chính xác trên tập huấn luyện cao, nhưng độ chính xác trên tập kiểm tra thấp. Nguyên nhân do mô hình không đủ tính tổng quát:có nhiều tập dữ liệu nhưng có quá nhiều nhiễm , Ít dữ liệu huấn luyện, học quá lâu, …

Một trong những cách nhận biết mô hình bị *overfitting*: mất mát trên tập đánh giá đang giảm thì tăng trong khi mất mất trên tập huấn luyện vẫn giảm

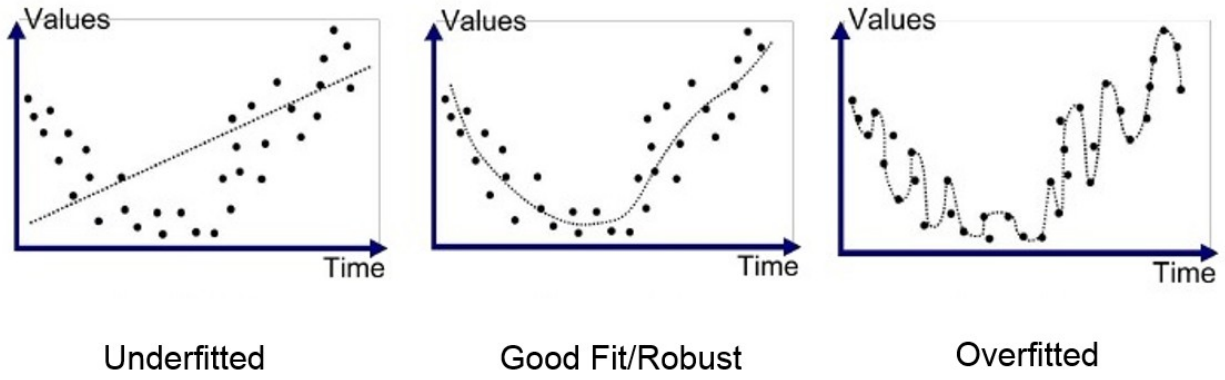
Một số giải pháp: mỗi thuật toán đều có những phương pháp giải quyết riêng. Đối với cây quyết định chúng ta có thể ngừng phân nhánh khi số lượng mẫu trong 1 nhánh bằng 1 số luật mà chúng ta đặtr a từ trước (Stopping). Đối với KNN thì việc chọn lựa K đủ lớn. Hoặc đối với thuật toán logistic regression chúng ta có thể thêm lượng phạt theo chuẩn l1 hoặc là l2…

Underfitting

Ngược lại với overfitting, mô hình dự đoán kém trên cả tập huấn luyện và kiểm tra. Nguyên nhân do mô hình quá đơn giản.

Một trong những cách nhận biết: độ chính xác trên cả tập huấn luyện và kiểm tra đều thấp.

Một số giải pháp: Tăng độ phức tạp của mô hình lên.



Hình 4.7.1 : Minh họa đồ thị của hàm số trong 3 trường hợp: underfitting, good fit, overfitting[1]

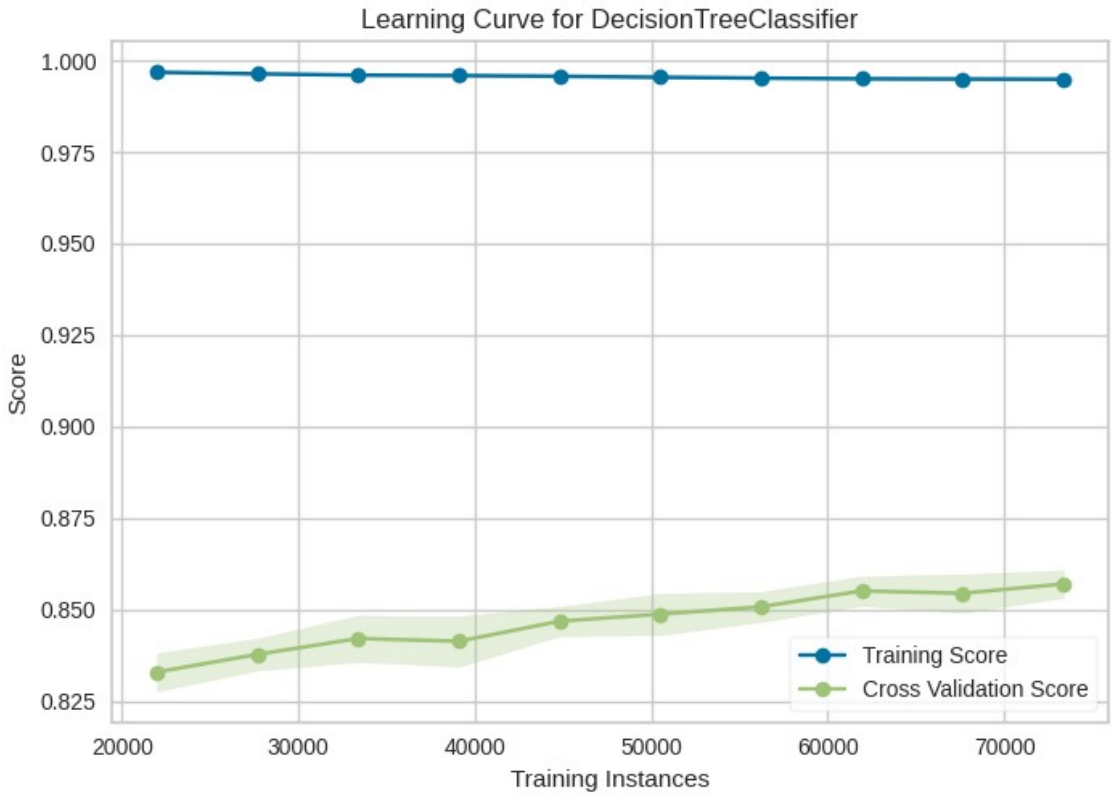
5. Thực nghiệm.

Dữ liệu bao gồm là 119390 bản ghi sau khi lựa feature và tranform dữ liệu thì mỗi bản ghi sẽ có 236 chiều dữ liệu. ta sẽ chia ra 39399 bản ghi làm test để đánh giá mô hình và 79991 bản ghi làm tập tranning.

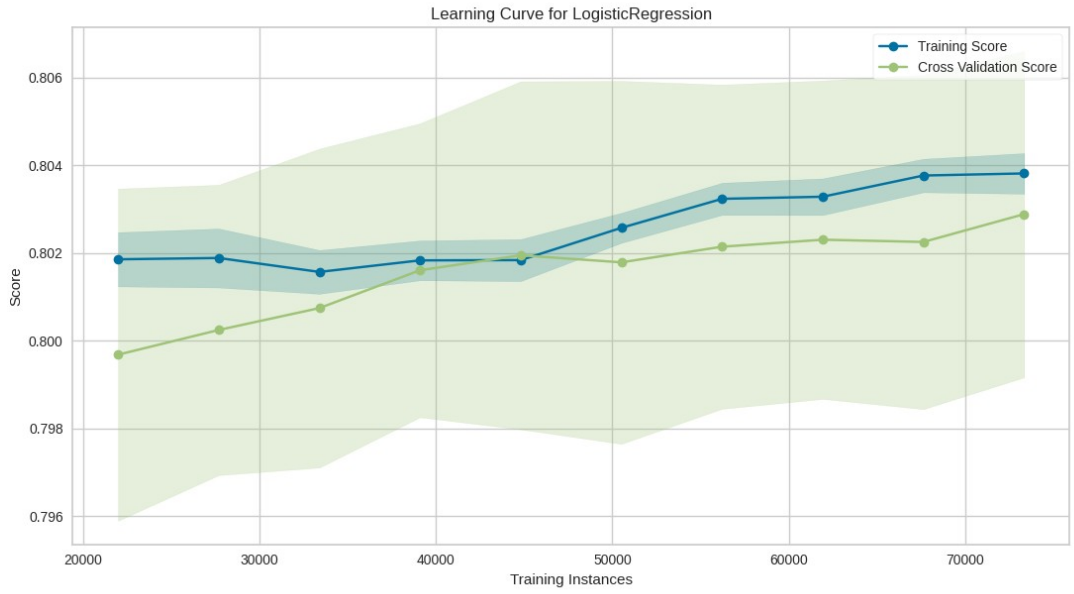
Bảng kết quả tranning.

Thuật toán	Nhãn	precision	recall	F1-score	Average F1
Naïve bayes	**0**	**0.76**	**0.92**	**0.83**	**0.73**
	1	**0.80**	**0.51**	**0.62**	
Logistic regression	**0**	**0.81**	**0.91**	**0.86**	**0.79**
	1	**0.8**	**0.64**	**0.72**	
Decision Tree	**0**	**0.88**	**0.91**	**0.89**	**0.85**
	1	**0.84**	**0.79**	**0.81**	
KNN	**0**	**0.79**	**0.91**	**0.85**	**0.77**
	1	**0.80**	**0.59**	**0.68**	

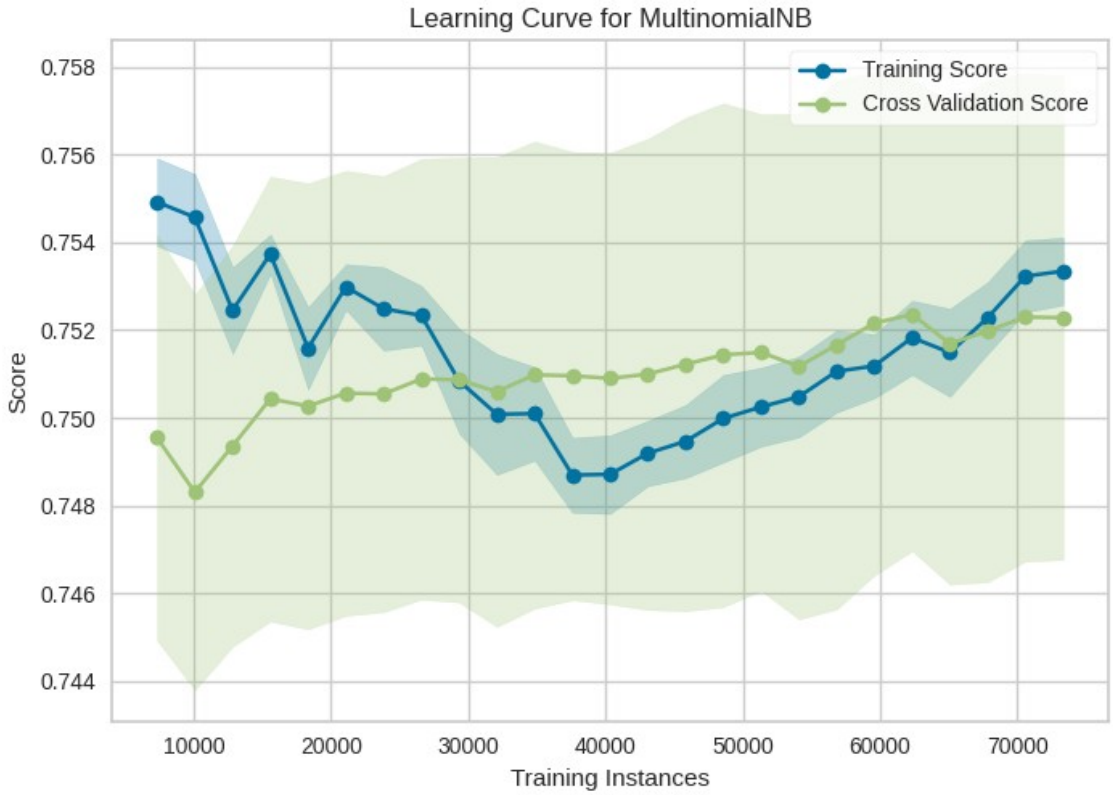
[1] Nguồn: https://medium.com/analytics-vidhya/understanding-overfitting-and-underfitting-in-machine-learning-2a2f3577fb27



Hình 4.1 Learning Curve của mô hình cây quyết định



Hình 4.2 Learning Curve của mô hình Logistic regression



Hình 4.3 Learning Curve của mô hình Naïve Bayes

Nhận Xét: Với dữ liệu Booking Hotel demand thì mô hình cây quyết định cho kết quả cao nhất với F-Measure đạt 0.85. Mô hình Naïve Bayes cho kết quả thấp nhất với F-Measure đạt 0.73. Mô hình cây quyết định có kết quả cao như vậy là do cây quyết định nó có những tính ưu việt như phương pháp không sử dụng tham số hoàn toàn tuân theo luật thống kê , Và nhờ đó kết quả phân tích dữ liệu trở nên khách quan nhất (tự nhiên nhất) Và đặc biệt rằng Mô hình cây quyết định xử lí rất tốt những dữ liệu mà những thuộc tính nó độc lập tuyến tính với nhau hay cũng xử lí rất tốt những bản ghi có dữ liệu bị miss value. Bên cạnh đó thì mô hình Naïve Bayes cũng không dùng các siêu tham số và học hoàn toàn dựa trên thống kê nhưng mô hình Naïve Bayes rất nhạy cảm với dữ liệu ngoại lệ (outlier).

Tài liệu tham khảo.

[1] https://www.statista.com/statistics/324901/share-of-guests-returning-to-hotels-worldwide/
[2] Dữ liệu được lấy từ Hotel booking demand datasets-Nuno Antnio, Ana de Almeida, Luis Nunes
[3] https://machinelearningcoban.com/2017/08/31/evaluation/
[4] Giáo trình Nhập môn trí tuệ nhân tạo – Thầy Từ Minh Phương
[5] Multiclass classification - wiki
[6] theo wikipedia: https://vi.wikipedia.org/wiki/Khai_ph%C3%A1_d%E1%BB%AF_li%E1%BB%87u
[7] Bài giảng kho dữ liệu và kỹ thuật khai phá – Nguyễn Quỳnh Chi
[8] Pattern Recognition and Machine Learning- Christopher M. Bishop