TRƯỜNG ĐẠI HỌC KINH TẾ - ĐẠI HỌC ĐÀ NẴNG

Môn học: Kho và khai phá dữ liệu

DỰ ĐOÁN VIỆC HỦY PHÒNG CỦA KHÁCH SẠN

GVHD: Nguyễn Văn Chức

Nhóm: 1

Thành viên:

Nguyễn Thị Quý

Nguyễn Thị Mỹ Thúy

Huỳnh Nhật Linh

Phan Thị Thảo Ngân

Lê Thị Phương Thanh

Lớp học phần: ELC3009_46K29.2

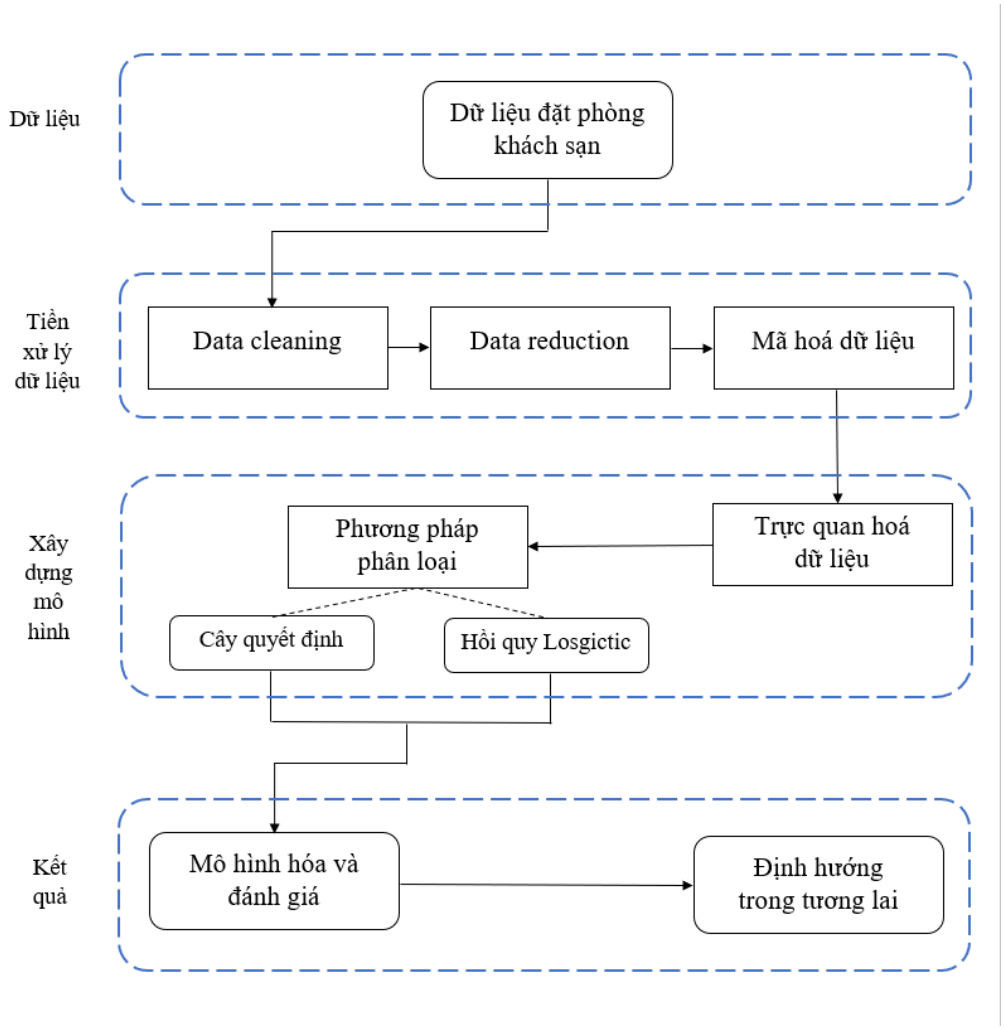
Bảng đánh giá hoàn thành nhiệm vụ của các thành viên:

Họ và tên	Phần trăm đóng góp
Nguyễn Thị Quý	20%
Nguyễn Thị Mỹ Thúy	20%
Huỳnh Nhật Linh	20%
Phan Thị Thảo Ngân	20%
Lê Thị Phương Thanh	20%

Dataset nhóm sử dụng được lấy từ:
https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand

Dữ liệu ban đầu từ bài báo Bộ dữ liệu nhu cầu đặt phòng khách sạn, được viết bởi Nuno Antonio, Ana Almeida và Luis Nunes cho Dữ liệu trong Tóm tắt, Tập 22, tháng 2 năm 2019. Dữ liệu đã được Thomas Mock và Antoine Bichat tải xuống và làm sạch cho #TidyTuesday trong tuần của ngày 11 tháng 2 năm 2020.

Tiền trình làm bài tập báo cáo nhóm:



MỤC LỤC

I. ĐẶT VẤN ĐỀ

Việc hủy đặt phòng có tác động đáng kể đến các quyết định quản lý nhu cầu trong ngành khách sạn. Việc hủy bỏ phòng dẫn đến hạn chế việc đưa ra các dự báo chính xác về hiệu suất quản lý doanh thu.

Để tránh những rắc rối do việc hủy đặt phòng gây ra, các khách sạn thực hiện các chính sách hủy đặt phòng cứng nhắc và chiến lược đặt trước quá nhiều, điều này cũng có thể có ảnh hưởng tiêu cực đến doanh thu và danh tiếng.

II. MỤC ĐÍCH CỦA BÀI TOÁN

Chúng ta cần khám phá phân tích các tác nhân, động lực dẫn đến hủy phòng của các khách hàng, từ đó có thể hạn chế tình trạng hủy phòng và đưa ra chiến lược tốt nhất để làm giảm rủi ro cho khách sạn và mục đích cuối cùng là tăng doanh thu.

Kết quả cho phép các nhà quản lý khách sạn dự đoán chính xác nhu cầu ròng và xây dựng dự báo tốt hơn, cải thiện chính sách hủy đặt phòng, xác định chiến thuật đặt trước tốt hơn và do đó sử dụng các chiến lược định giá và phân bổ phòng, cũng như phân bổ nhân sự cho phù hợp và thực phẩm tốt hơn.

III. THU THẬP VÀ TIỀN XỬ LÍ DỮ LIỆU

1. Tổng quan về dữ liệu

Index	Columns	Dtype	Description
1	hotel	Object	Loại khách sạn (Gồm City Hotel và Resort Hotel)
2	is_canceled	Integer	Giá trị cho biết: Hủy phòng(1), Không hủy(0)
3	lead_time	Integer	Số ngày đặt phòng trước khi đến ở
4	arrival_date_year	Integer	Năm đặt phòng
5	arrival_date_month	Object	Tháng đặt phòng
6	arrival_date_week_number	Integer	Số tuần trong năm
7	arrival_date_day_of_month	Integer	Ngày trong tháng
8	stays_in_weekend_nights	Integer	Số đêm khách đặt vào cuối tuần(T7 hoặc CN)
9	stays_in_week_nights	Integer	Số đêm khách đặt vào trong tuần(từ T2 đến T6)
10	adults	Integer	Số người lớn
11	children	Float	Số trẻ em
12	babies	Integer	Số trẻ sơ sinh
13	meal	Object	ID bữa ăn(SC: không có bữa ăn,BB: Bữa sáng,HB: Bữa sáng và tối,FB: 3 bữa)
14	country	Object	Quốc gia
15	market_segment	Object	Phân khúc thị trường
16	distribution_channel	Object	Kênh phân phối
17	is_repeated_guest	Integer	(1) là khách quen, (0) là khách mới
18	previous_cancellations	Integer	Số lần khách hủy phòng trước khi đặt lại tại thời điểm hiện tại
19	previous_bookings_not_canceled	Integer	Số lần không bị khách hủy khi đặt trước
20	reserved_room_type	Object	Mã loại phòng đặt trước
21	assigned_room_type	Object	Mã cho loại phòng được chỉ định cho đặt phòng
22	booking_changes	Integer	Số lượng thay đổi ý định đặt hay hủy phòng của khách

23	deposit_type	Object	Loại tiền gửi gồm 3 loại:(không hoàn lại,có thể hoàn lại, không có tiền đặt cọc)
24	agent	Float	Giấy tờ tùy thân của khách hàng
25	company	Float	ID công ty của khách hàng
26	days_in_waiting_list	Integer	Số ngày đặt chỗ trong danh sách chờ trước khi nó được xác nhận cho khách hàng
27	customer_type	Object	Loại khách hàng có 4 nhóm (nhóm, hợp đồng, tạm thời hoặc bên tạm thời)
28	adr	Float	Tỷ lệ trung bình đặt phòng bằng cách chia tổng của tất cả các giao dịch lưu trú cho số đêm lưu trú
29	required_car_parking_spaces	Integer	Số lượng chỗ đậu xe ô tô theo yêu cầu của khách hàng
30	total_of_special_requests	Integer	Số lượng yêu cầu đặt biệt của khách hàng (ví dụ:giường đôi hoặc tầng cao)
31	reservation_status	Object	Trạng thái đặt chỗ cuối cùng
32	reservation_status_date	Object	Ngày trả phòng

2. Làm sạch dữ liệu (Data cleaning)

Khai báo thư viện:

```
import pandas as pd

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('hotel_bookings.csv')
data.head()
```

```
data.shape
```

```
(119390, 32)
```

→Dữ liệu gồm 119390 dòng và 32 cột.

```
data.isnull().sum().sum() #Kiểm tra có bao nhiêu giá trị Null
```

Dữ liệu có 129425 giá trị Null cần được xử lí.

```
# Xem các cột có giá trị null
null_columns = data.columns[data.isnull().any()]
data[null_columns].isnull().sum()
```

Các cột có giá trị null gồm: children có 4 giá trị, country có 488 giá trị, agent có 16340 giá trị, company có 112593 giá trị.

```
# Tạo giá trị mới cho các giá trị bị null, xóa bỏ cột có quá nhiều giá trị null
data['children'] =  data['children'].fillna('0')
data['country'] = data['country'].fillna(data['country'].mode().index[0])
data['agent']= data['agent'].fillna('0')
data=data.drop(['company'], axis =1)
```

```
# Thay đổi kiểu dữ liệu
data['children']= data['children'].astype(int, errors='ignore')
data['agent']= data['agent'].astype(int)
data['country']= data['country'].astype(str)
```

```
#Kiểm tra số lượng giá trị trong các thuộc tính định tính
data.describe(include = "object").T
```

	count	unique	top	freq
hotel	119390	2	City Hotel	79330
arrival_date_month	119390	12	August	13877
meal	119390	5	BB	92310
country	119390	178	PRT	48590
market_segment	119390	8	Online TA	56477
distribution_channel	119390	5	TA/TO	97870
reserved_room_type	119390	10	A	85994
assigned_room_type	119390	12	A	74053
deposit_type	119390	3	No Deposit	104641
customer_type	119390	4	Transient	89613
reservation_status	119390	3	Check-Out	75166
reservation_status_date	119390	926	2015-10-21	1461

Xuất file dữ liệu để trực quan hóa:

```
data.to_csv('hotel_bookings_clean.csv',index=0) #Tạo file để trực quan hóa
```

Dữ liệu lúc này gồm:

```
data_clean = pd.read_csv('hotel_bookings_clean.csv')
data.shape
```

```
(119390, 31)
```

→Dữ liệu gồm 119390 dòng và 31 cột.

3. Thu giảm dữ liệu (Data reduction)

Tạo cột "change_in_room": xem khách hàng nào thay đổi phòng khi đặt phòng.

```
def roomChange(row):
    if row['assigned_room_type'] == row['reserved_room_type']:
        return False
    else:
        return True

data['change_in_room'] = data.apply(roomChange, axis=1)
```

Xác định các thuộc tính quan trọng: Thuộc tính có mối quan hệ tương quan cao với biến "is_canceled":

```
# Xem thuộc tính nào quan trọng (bao gồm cả tương quan thuận và nghịch)
cancel_corr = data.corr()["is_canceled"]
cancel_corr.abs().sort_values(ascending=False)[1:]
```

- Đối với biến định lượng: Quan sát bảng ta có thể thấy rằng các cột về thời gian không có nhiều ý nghĩa trong bài toán này nên chúng ta sẽ xóa hết nó. Đồng thời 5 tính năng quan trọng là "lead_time", "total_of_special_requests", "Requi_car_parking_spaces", "booking_changes" và "before_cancellations".

```
lead_time                       0.293123
total_of_special_requests       0.234658
required_car_parking_spaces     0.195498
booking_changes                 0.144381
previous_cancellations          0.110133
is_repeated_guest               0.084793
adults                          0.060017
previous_bookings_not_canceled  0.057358
days_in_waiting_list            0.054186
adr                             0.047557
agent                           0.046529
babies                          0.032491
stays_in_week_nights            0.024765
arrival_date_year               0.016660
arrival_date_week_number        0.008148
arrival_date_day_of_month       0.006130
children                        0.006070
stays_in_weekend_nights         0.001791
Name: is_canceled, dtype: float64
```

- Đối với biến định tính: Ta có thể thấy "reservation_status" có ảnh hưởng nhất đến "is_canceled", tuy nhiên cột này chỉ có ý nghĩa đối với các khách đã đặt phòng, không có ý nghĩa đối với bài toán hủy phòng của chúng ta nên chúng ta sẽ xóa cột này. Ngoài ra ta sẽ xóa các cột có mối quan hệ tương quan yếu với "is_canceled". Đồng thời xóa luôn 2 cột "assigned_room_type" và "reserved_room_type" vì đã có cột "change_in_room" thay thế.

```
#Loại bỏ các thuộc tính không cần thiết cho mô hình
data.drop(['reservation_status_date' ,'arrival_date_week_number',
'stays_in_week_nights', 'arrival_date_month','arrival_date_year','arrival_date_day_of_month',
'stays_in_weekend_nights','adr','agent' ], axis = 1, inplace = True)
data.shape
```

```
(119390, 22)
```

```
data.drop(['reservation_status','meal','assigned_room_type','reserved_room_type'], axis = 1, inplace = True)
data.shape
```

```
(119390, 19)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 19 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119390 non-null  object
 1   is_canceled                     119390 non-null  int64
 2   lead_time                       119390 non-null  int64
 3   adults                          119390 non-null  int64
 4   children                        119390 non-null  int32
 5   babies                          119390 non-null  int64
 6   country                         119390 non-null  object
 7   market_segment                  119390 non-null  object
 8   distribution_channel            119390 non-null  object
 9   is_repeated_guest               119390 non-null  int64
 10  previous_cancellations          119390 non-null  int64
 11  previous_bookings_not_canceled  119390 non-null  int64
 12  booking_changes                 119390 non-null  int64
 13  deposit_type                    119390 non-null  object
 14  days_in_waiting_list            119390 non-null  int64
 15  customer_type                   119390 non-null  object
 16  required_car_parking_spaces     119390 non-null  int64
 17  total_of_special_requests       119390 non-null  int64
 18  change_in_room                  119390 non-null  bool
dtypes: bool(1), int32(1), int64(11), object(6)
memory usage: 16.1+ MB
```

→Dữ liệu gồm 119390 dòng và 19 cột.

4. Mã hóa dữ liệu

- Mã hóa dữ liệu để thực thiện mô hình Logistic Regression:

```
data_model_logistic = data.copy()
```

Sử dụng hàm ***get_dummies*** để mã hóa dữ liệu từ dạng chữ sang dạng số:

```
from sklearn.preprocessing import LabelEncoder, StandardScaler, MinMaxScaler
lb = LabelEncoder()
var = ['hotel', 'customer_type', 'deposit_type', 'change_in_room', 'market_segment', 'distribution_channel', 'country']
for item in var:
    data_model_logistic[item] = lb.fit_transform(data_model_logistic[item])
data_model_logistic = pd.get_dummies(data_model_logistic, columns=['hotel', 'customer_type', 'deposit_type',
                                    'change_in_room', 'market_segment', 'distribution_channel', 'country'])
```

Xuất file .csv:

```
data_model_logistic.to_csv('hotel_bookings_model_logistic.csv',index=0) #Tạo file để tạo mô hình dự báo, phân loại
```

- Mã hóa dữ liệu để thực hiện mô hình Decision Tree:

```
data_model_decision_tree = data.copy()
```

Sử dụng hàm ***replace*** để chuyển dữ liệu cột "is_canceled" từ dạng số sang dạng chữ để tạo class cho dữ liệu:

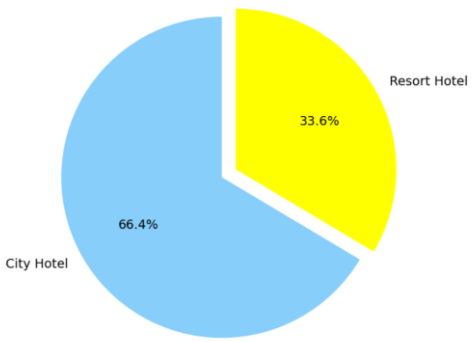
```
data_model_decision_tree['cancelled'] = data_model_decision_tree['is_canceled'].replace({0: 'No', 1: 'Yes'})

data_model_decision_tree.drop(['is_canceled'],axis = 1, inplace = True)
```

Xuất file .csv:

```
data_model_decision_tree.to_csv('hotel_bookings_decision_tree.csv',index=0) #Tạo file để tạo mô hình dự báo, phân loại
```

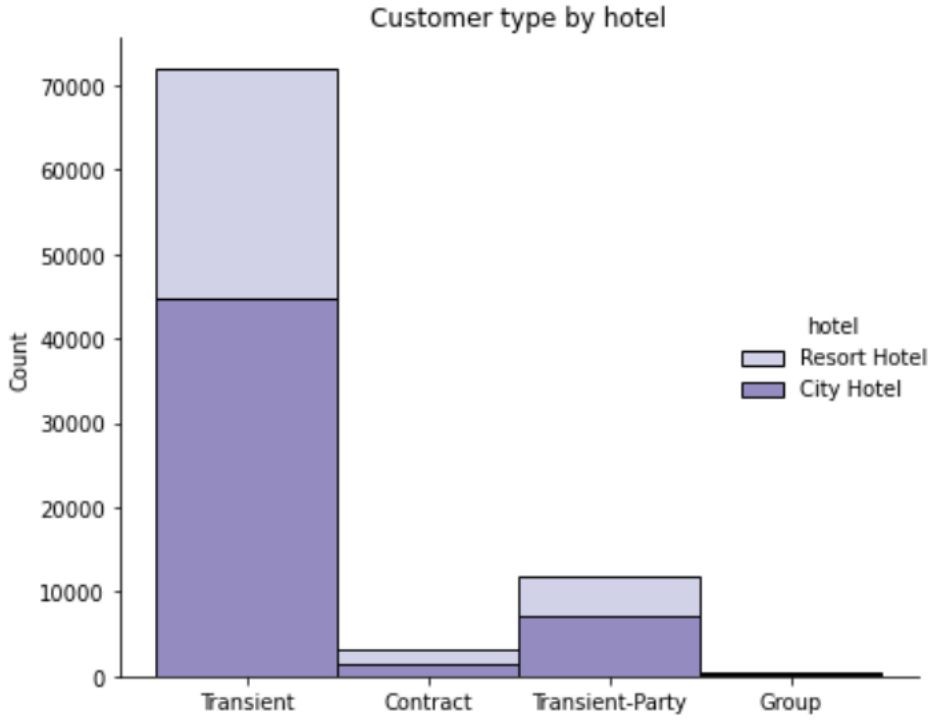
IV. TRỰC QUAN HÓA DỮ LIỆU

1. Một vài biểu đồ trực quan

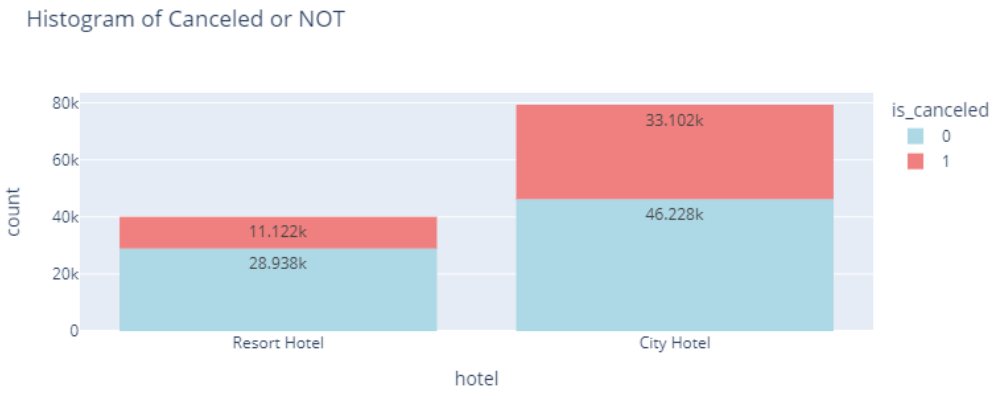


Như biểu đồ trên cho ta thấy số lượng khách đặt phòng ở 2 loại khách sạn: City Hotel chiếm 66,4% và Resort Hotel chiếm 33,6%.

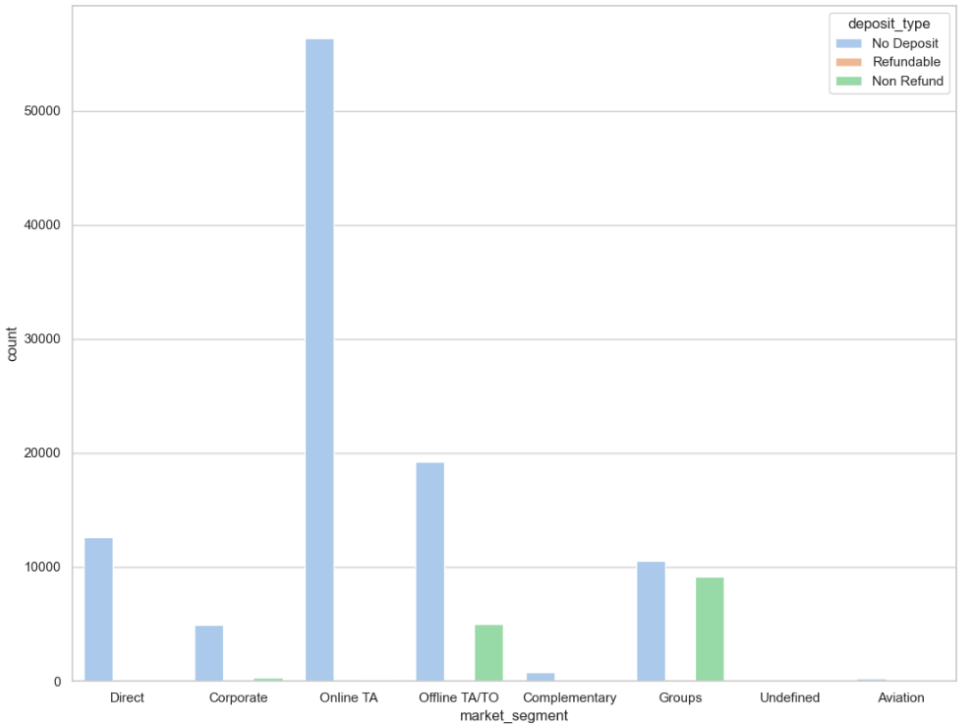
- Loại khách hàng nào phổ biến ở mỗi khách sạn?

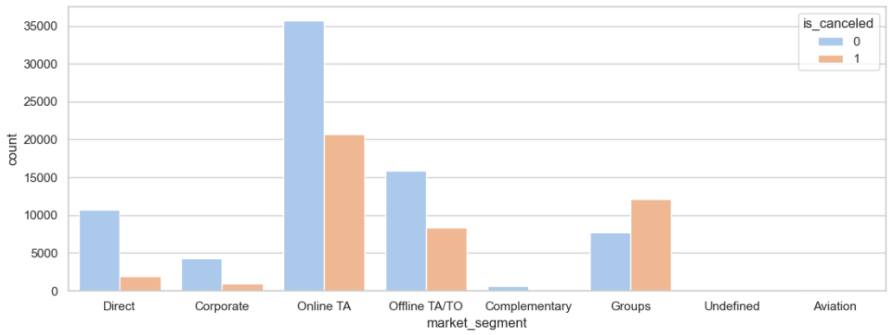


→ Khách hàng thuộc nhóm Transient có số lượng đặt phòng cao nhất.



→ City Hotel có lượng đặt phòng nhiều hơn gần gấp đôi so với Resort Hotel, và có lẽ vì thế mà lượt hủy phòng cũng cao hơn nhiều.

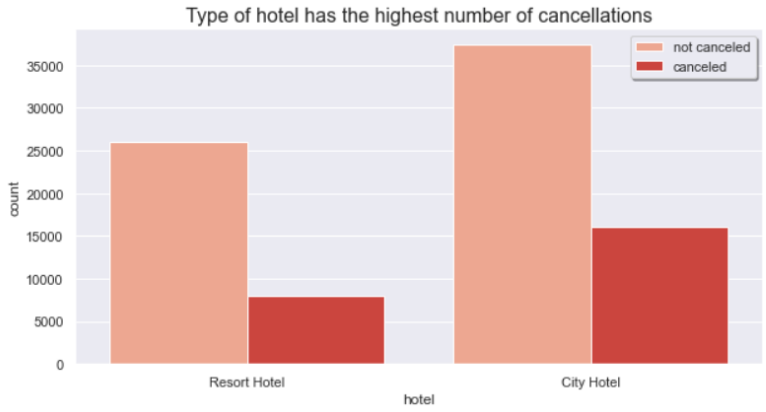




→ Hầu hết các kênh trên thị trường là Không cần đặt cọc trước, số ít là không hoàn tiền và hầu như không có loại khách hàng là có thể hoàn trả.

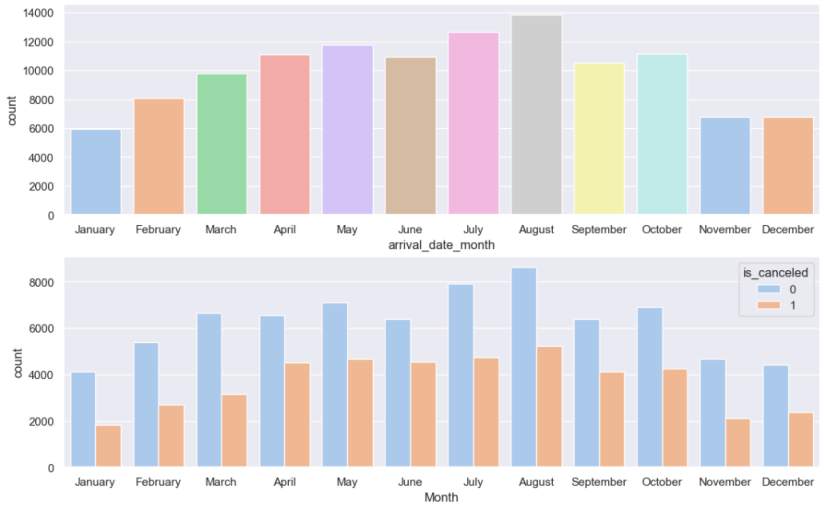
- Group segment có tỉ lệ trả phòng rất cao, hơn 50% trả phòng.
- Ở Online TA và Offline TA/TO tỉ lệ trả phòng cũng khá cao khoảng 33% mặc dù đã áp dụng chính sách không cần đặt cọc trước.

- Tình trạng hủy phòng như thế nào?

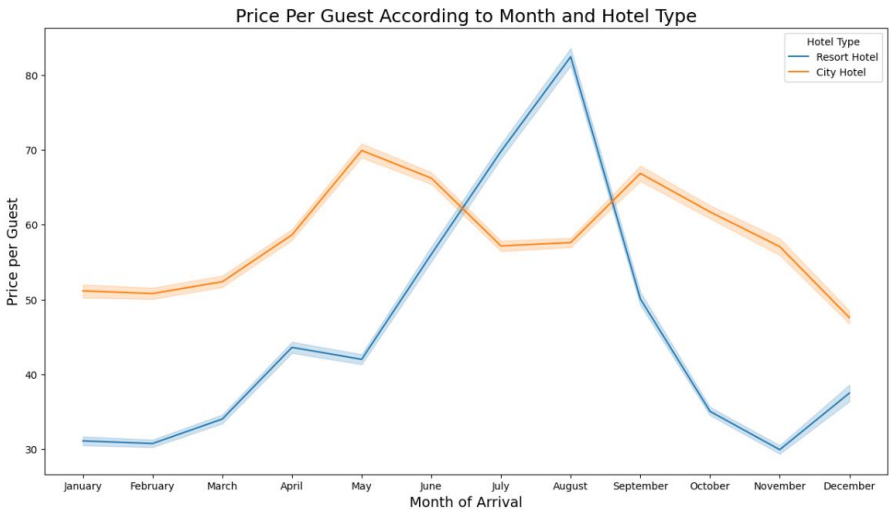


→ Loại hình khách sạn quyết định tỷ lệ hủy đặt phòng với số lượng hủy phòng cao hơn ở các khách sạn thành phố so với các khách sạn nghỉ dưỡng.

→ Khi thời gian chờ càng tăng thì khả năng bị hủy càng tăng.



→ Tỉ lệ hủy phòng từ tháng 4 đến tháng 8 khá là cao. Vì đây là khoảng thời gian vào mùa hè – mùa cao điểm nên nhu cầu du lịch của khách nội địa và quốc tế đi tham quan và du lịch tăng cao.



→ Đồ thị này cho thấy rõ rằng giá trong Resort Hotel cao hơn nhiều trong mùa hè và giá của City Hotel thay đổi ít hơn và đắt nhất trong mùa hè và mùa thu. Đây là mùa cao điểm → lượng khách tăng cao → giá khách sạn tăng.

2. Đưa ra nhận xét về tình hình đặt phòng và hủy phòng của khách sạn

Khách hàng chủ yếu đặt phòng nhiều nhất vào thời điểm tiết trời chuyển sang hè và thu, bởi đây là lúc học sinh sinh viên được nghỉ hè, là thời điểm hoàn hảo để mọi người được nghỉ xả hơi, vui chơi, du lịch một cách thoải mái.

Nhu cầu đặt phòng cao cũng sẽ đi đôi với những rủi ro huỷ đặt phòng, lượng khách đặt phòng chủ yếu ở các khách sạn thành phố → chính vì thế nên lượng huỷ phòng xuất hiện nhiều tại các khách sạn thành phố.

3. Giảm thiểu vấn đề hủy phòng của khách hàng

Hủy đặt phòng là điều mà không có một khách sạn nào mong muốn, ảnh hưởng đến hiệu suất sử dụng phòng và doanh thu của khách sạn. Và khách hàng có những lý do chính đáng như bận việc đột xuất hoặc thay đổi kế hoạch nhưng cũng có những trường hợp khách đã tìm được nhà cung cấp dịch vụ tốt hơn.

Để giảm thiểu tình trạng huỷ đặt phòng hiện nay, khách sạn nên thiết lập các chính sách huỷ phòng với những quy định và điều khoản rõ ràng để giảm thiểu tình trạng khách đặt phòng rồi không đến.

Tạo ưu đãi đặc biệt cho các đặt phòng trực tiếp.

Cung cấp giá cả rõ ràng, nên công bố giá của từng phòng và giá đó đã bao gồm thuế và phí dịch vụ. Bằng cách đó, khách hàng sẽ biết những gì họ mong đợi.

Kết hợp trò chuyện trực tiếp, bất cứ khi nào khách gặp sự cố khi đặt phòng trên trang web của khách sạn, họ cần một người có thể giúp họ giải quyết ngay lập tức. Do đó, sẽ rất hữu ích nếu bạn có thể kết hợp trò chuyện trực tiếp trên trang

web khách sạn của mình. Làm như vậy có thể giúp khách của bạn đặt phòng thành công. Nó cũng ngăn việc phát sinh thêm một lần đặt phòng khách sạn bị hủy bỏ khác.

V. PHƯƠNG PHÁP PHÂN LOẠI

1. Giới thiệu phân loại

Data classification hay phân loại dữ liệu là công việc sắp xếp các dữ liệu dựa theo những tiêu chí khác nhau được đặt ra, hay mức độ tần suất truy cập sử dụng data. Dựa vào những yếu tố trên để tiến hành phân loại theo nhiều tầng lớp và mức độ khác nhau cho từng loại dữ liệu.

Quá trình gồm hai bước:

- Bước học (giai đoạn huấn luyện): xây dựng bộ phân loại (classifier) bằng việc phân tích/học tập huấn luyện.
- Bước phân loại (classification): phân loại dữ liệu/đối tượng mới, nếu độ chính xác của bộ phân loại được đánh giá là có thể chấp nhận được (acceptable).

Các giải thuật phân loại dữ liệu:

- Logistic Regression
- Cây quyết định (Decision tree)
- Mạng Bayesian
- Mạng Neural

2. Phân loại dữ liệu với mô hình hình Logistic Regression

a. Sơ lược về mô hình hồi quy Logistic

Hồi quy Logistic là kỹ thuật để phân tích mối liên hệ giữa biến độc lập với biến phụ thuộc là biến nhị phân (có 2 trạng thái/giá trị).

b. Khai phá dữ liệu với mô hình

Ma trận nhầm lẫn (Confusion matrix) là một bảng thường được sử dụng để mô tả hiệu suất của một mô hình phân loại trên một tập dữ liệu thử nghiệm mà các giá trị thực được biết đến.

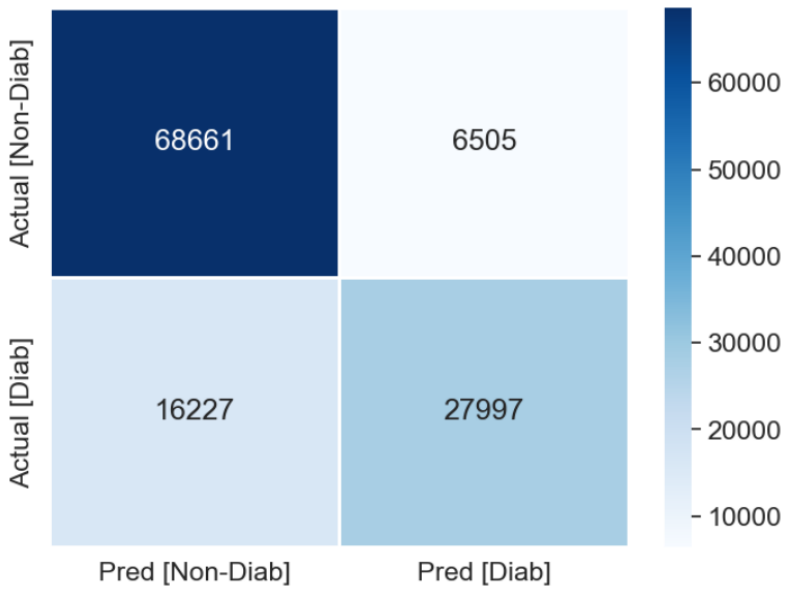
Từ mô hình:

1. TN = 68661

2. TP = 27997

3. FN = 16227

4. FP = 6505



Với Diab = Hủy phòng và Non-Diab = Không hủy phòng, ta có thể giải thích các con số:

1. TP: Mô hình dự đoán 27997 phòng bị hủy và trên thực tế phòng bị hủy (Mô hình đúng ở đây)

2. TN: Mô hình dự đoán 68661 phòng không bị hủy và trên thực tế phòng không bị hủy (Mô hình đúng ở đây)

3. FP: Mô hình dự đoán 6505 phòng bị hủy và trên thực tế phòng không bị hủy (Mô hình sai ở đây - "Sai lầm loại 1")

4. FN: Mô hình dự đoán 16227 phòng không bị hủy và trên thực tế phòng bị hủy (Mô hình sai ở đây - "Sai lầm loại 2")

Các chỉ số đánh giá mô hình

- Độ chính xác (Accuracy): Mô hình dự đoán đúng đến 80,69% trong tổng thể.

	precision	recall	f1-score	support
0	0.81	0.91	0.86	75166
1	0.81	0.63	0.71	44224
accuracy			0.81	119390
macro avg	0.81	0.77	0.78	119390
weighted avg	0.81	0.81	0.80	119390

- Độ tập trung (Precision): Khi mô hình dự đoán là Hủy phòng và thực tế là Hủy phòng thì độ chính xác của nó là 81%. Mô hình dự đoán đạt hiệu quả cao.
- Độ nhạy (Recall): Tỷ lệ phần trăm số phòng hủy được xác định chính xác là có hủy phòng.

→ Recall thấp, Precision cao: Cho biết rằng chúng ta bỏ lỡ rất nhiều ví dụ tích cực (FN cao) nhưng những ví dụ mà chúng ta dự đoán là tích cực thực sự là tích cực (FP thấp).

→ Nói cách khác, ta bỏ lỡ rất nhiều ví dụ về việc hủy phòng khách sạn nhưng những trường hợp mà dự đoán là hủy phòng thực sự là hủy phòng.

- Độ đặc hiệu (Specificity): Tỷ lệ phần trăm số phòng không hủy được xác định chính xác là không hủy phòng.

- F1-Score: được tính bằng 2*(Precision*Recall)/(Precision+Recall)

Rất khó để so sánh hai mô hình có Precision thấp và Recall cao hoặc ngược lại. Vì vậy, để so sánh chúng, chúng tôi sử dụng F-Score. Điểm F giúp đo lường Precision và Recall cùng một lúc. Điểm F1 truyền tải sự cân bằng giữa Precision và Recall.

c. Một số vấn đề cần xử lí

Có 75166 trường hợp khách không hủy phòng và 44224 trường hợp khách hủy phòng. Chúng ta có thể thấy rõ rằng, tập dữ liệu này không cân bằng.

Vì một số lý do:

- Các thuật toán như Logistic Regression giả định rằng dữ liệu là cân bằng. Nhưng nếu dữ liệu không cân bằng, họ sẽ đặt nặng hơn lên tầng lớp đa số. Trong trường hợp này, nhóm không hủy phòng (75166).

→ Độ chính xác không phải là thước đo hữu ích.

Do đó, chúng ta sẽ thấy các chỉ số khác như Độ nhạy, Độ đặc hiệu và Roc_Auc.

→ Sử dụng điểm Roc_Auc làm thước đo.

AUC cho biết mô hình có bao nhiêu khả năng phân biệt giữa các lớp:

- AUC là Khu vực dưới Đường cong. AUC cao hơn, mô hình dự đoán 0s là 0s và 1s là 1s tốt hơn.
- Bằng cách tương tự, AUC cao hơn, mô hình tốt hơn trong việc phân biệt giữa khách có hủy phòng và không hủy phòng.

- Đường cong ROC được vẽ với Sensitivity (Độ nhạy): TPR so với Tỷ lệ ô sai (1-Độ đặc hiệu): FPR, trong đó TPR nằm trên trục y và FPR trên trục x.

```
Roc_Auc = metrics.roc_auc_score(y, y_pred)
print ('Roc Auc Score: ', Roc_Auc)
```

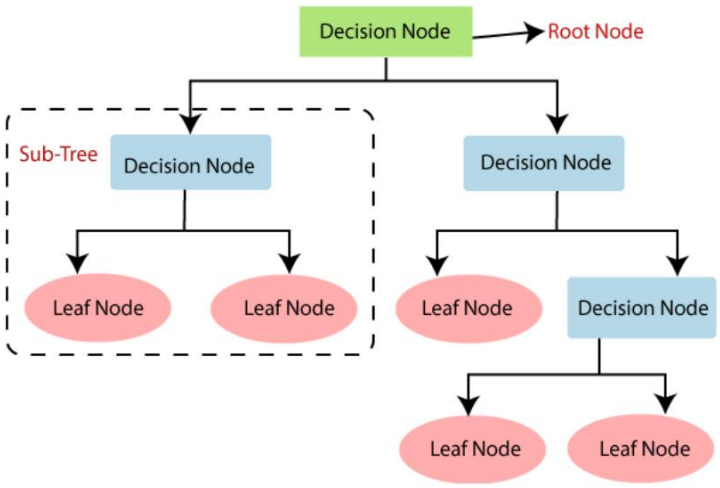
```
Roc Auc Score:  0.7732653761435423
```

→ Vì vậy, nó có điểm ROC AUC tốt. Có 77% cơ hội rằng mô hình sẽ có thể phân biệt giữa phân loại HỦY PHÒNG và KHÔNG HỦY PHÒNG.

3. Phân loại dữ liệu với cây quyết định

a. Sơ lược về mô hình Decision Tree

- Cây quyết định là một kỹ thuật học có giám sát có thể được sử dụng cho cả bài toán phân loại và hồi quy. Nó là một bộ phân loại có cấu trúc cây, trong đó các nút bên trong biểu thị các tính năng của tập dữ liệu, các nhánh biểu thị các quy tắc quyết định và mỗi nút lá biểu thị kết quả.
- Trong cây Quyết định, có hai nút, đó là Nút Quyết định và Nút Lá. Các nút quyết định được sử dụng để đưa ra bất kỳ quyết định nào và có nhiều nhánh, trong khi các nút Lá là đầu ra của các quyết định đó và không chứa bất kỳ nhánh nào nữa.



Giải thuật xây dựng cây quyết định như ID3, J48, C4.5, CART,….

Giới thiệu một số độ đo:

- Entropy: là phép đo giá trị biến ngẫu hay độ thay đổi của dữ liệu được cho.
- Gain Ratio (được dùng trong C4.5).
- Gini Index (được dùng trong CART).
- Information Gain (độ lợi thông tin): giá trị này đại diện cho độ biến thiên dữ liệu đầu vào và đầu ra. Nó chính là giá trị thay đổi giữa các giai đoạn của entropy.

$$\text{Info}(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Trong đó:

- Info(D): Lượng thông tin cần để phân loại một phần tử D.
- p_i: xác suất để một phần tử bất kỳ trong D thuộc về lớp C_i, với i = 1, …, m

$$\text{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \text{Info}(D_j)$$

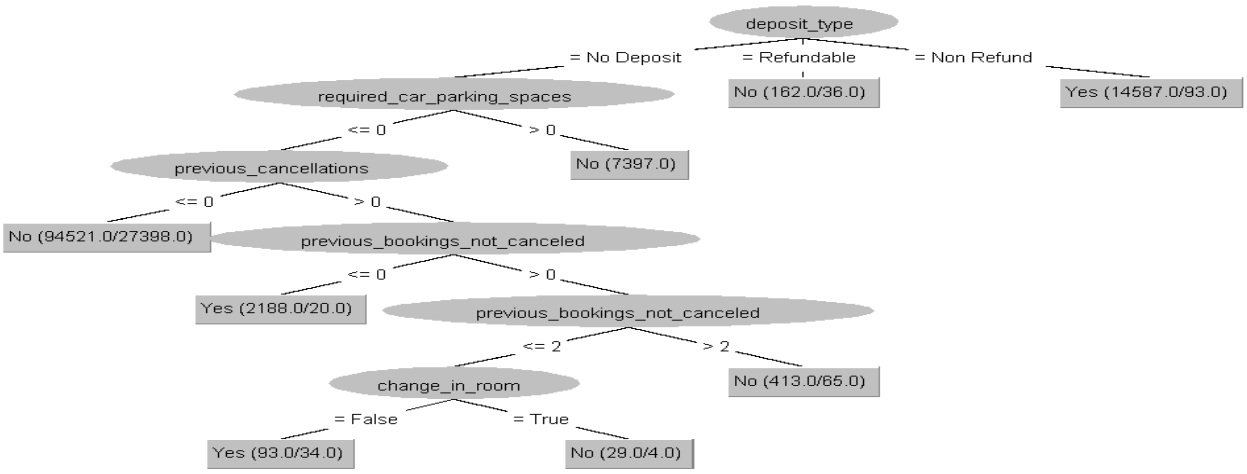
$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

b. Khai phá dữ liệu với mô hình

Đánh giá Độ lợi thông tin (Information Gain):

```
Attribute Evaluator (supervised, Class (nominal): 19 cancelled):
        Gain Ratio feature evaluator

Ranked attributes:
 0.34336  13 deposit_type
 0.18155  10 previous_cancellations
 0.12864  16 required_car_parking_spaces
 0.10349  18 change_in_room
 0.0636   11 previous_bookings_not_canceled
 0.0586   14 days_in_waiting_list
 0.04701  17 total_of_special_requests
 0.03825  12 booking_changes
 0.02922   9 is_repeated_guest
 0.02888   8 distribution_channel
 0.02744   6 country
 0.02629   7 market_segment
 0.02371   2 lead_time
 0.01495   1 hotel
 0.01454   5 babies
 0.01431  15 customer_type
 0.00562   3 adults
 0.00145   4 children

Selected attributes: 13,10,16,18,11,14,17,12,9,8,6,7,2,1,5,15,3,4 : 18
```

Từ bảng trên ta lựa ra các thuộc tính tốt như: 13, 10, 16, 18, 11



Trường hợp 1: Với khách không đặt cọc trước thì:

- Nếu khách có sự yêu cầu chỗ để xe thì khả năng cao khách hàng sẽ không hủy phòng.
- Nếu khách không yêu cầu chỗ để xe trước thì ta sẽ xem xét thêm điều kiện:
  - ➢ Nếu đó là khách chưa từng hủy phòng ở khách sạn thì có khả năng sẽ không hủy phòng.
  - ➢ Nếu đó là khách có đã từng hủy phòng ở khách sạn, ta sẽ xem xét thêm trường hợp khách hàng đó đã từng đặt phòng mà không hủy là bao nhiêu lần :
    - ✓ Nếu khách hàng đó chưa từng hủy phòng đã đặt thì khả năng cao sẽ hủy phòng luôn.
    - ✓ Nếu khách hàng đó có hủy phòng đã đặt thì:
      - o Nếu khách hàng đặt phòng mà không hủy trên 2 lần thì sẽ không có khả năng hủy phòng.
      - o Nếu khách hàng đặt phòng mà không hủy dưới 2 thì ta xét thêm điều kiện khách có thay đổi phòng trong quá trình đặt phòng không:
        - ⇒ Nếu có thay đổi thì không có khả năng hủy phòng.
        - ⇒ Nếu không thay đổi thì có khả năng hủy phòng.

Trường hợp 2: Với số tiền đặt cọc không hoàn lại thì có khả năng hủy phòng.

Trường hợp 3: Với số tiền đặt cọc có hoàn lại thì không có khả năng hủy phòng.

```
Correctly Classified Instances       91738               76.8389 %
Incorrectly Classified Instances     27652               23.1611 %
Kappa statistic                          0.431
Mean absolute error                      0.3296
Root mean squared error                  0.406
Relative absolute error                 70.6637 %
Root relative squared error             84.0664 %
Total Number of Instances           119390

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.622    0.732      0.998    0.844      0.521    0.719     0.784     No
                 0.378    0.002    0.991      0.378    0.547      0.521    0.719     0.685     Yes
Weighted Avg.    0.768    0.392    0.828      0.768    0.734      0.521    0.719     0.747

=== Confusion Matrix ===

     a     b   <-- classified as
 75021   145 |     a = No
 27507 16717 |     b = Yes
```

→ Mô hình có độ chính xác 76,8389%.

→ Dự đoán đúng 91738 dữ liệu [Đường chéo].

+ Mô hình dự đoán có 75021 phòng không bị hủy và trên thực tế nó không bị hủy.
+ Mô hình dự đoán có 16717 phòng bị hủy và thực tế là nó bị hủy.

→ Dự đoán sai 27652 điểm dữ liệu gồm:

+ 145 quyết định không hủy nhưng bị dự đoán sai là hủy.
+ 27507 quyết định hủy nhưng bị dự đoán sai là không hủy.

- Precision: Mô hình dự đoán Không hủy và thực sự Không hủy – mô hình dự đoán đúng khoảng 82,8%.
- Recall: Mô hình dự đoán Hủy và thực sự Hủy – mô hình dự đoán đúng khoảng 76,8%.
- Vì với mục đích của mô hình thì hủy phòng hay không hủy đều quan trọng như nhau vì thế nên chúng ta phải sử dụng F1- score là một chỉ số để trung hòa giữa Precision và Recall.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = 0.734$$

- ROC: là một đường cong biểu diễn xác suất. Đường cong này dựa trên hai chỉ số TP race và FP race

$$FPR = \frac{FP}{FP + TN} = 0.719$$

4. Đánh giá hai mô hình Logistic Regresstion và Decision Tree

- Có thể thấy sử dụng Hồi quy Logistic mang lại kết quả tốt hơn.
- Tuy nhiên, mô hình Cây quyết định dễ diễn giải hơn so với hồi quy logistic.

- Hệ số mô hình hồi quy logistic thể hiện độ quan trọng của biến số, tuy nhiên không đưa ra được lý giải về cách dự báo.
- Cây quyết định có thể đưa thêm sự tương tác phi tuyến tính giữa các biến số mà hồi quy logistic không thể hiện được.

VI. MÔ HÌNH HÓA VÀ ĐÁNH GIÁ, NHẬN XÉT

Ta sử dụng PMS, CRS, các kênh trong phân khúc thị trường, các kênh phân phối để thực hiện mô hình khách sạn

Một số định nghĩa cơ bản:

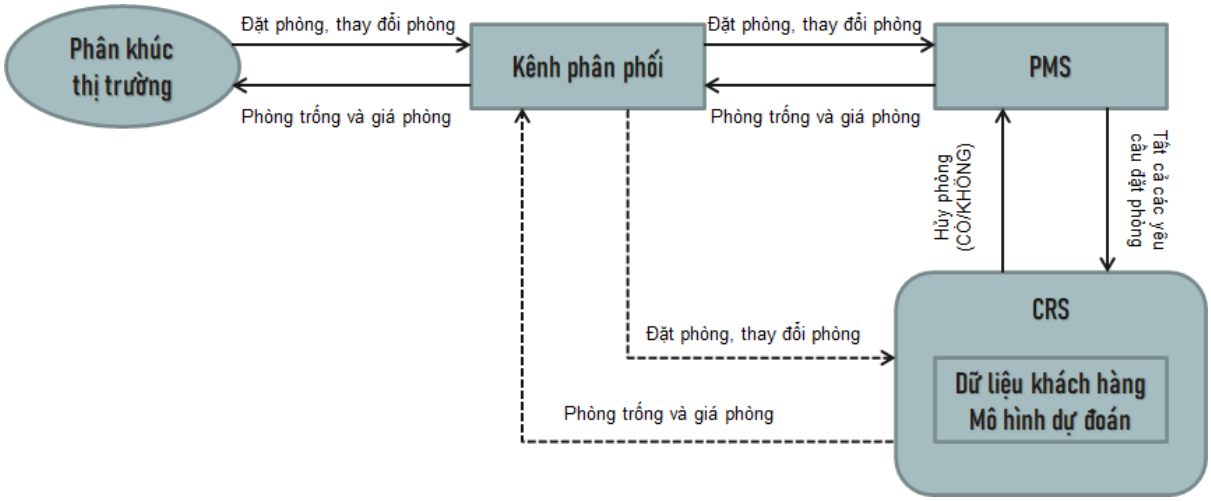
- PMS (property management system) là hệ thống giúp quản lý mọi vận hành của khách sạn từ lễ tân đến buồng phòng. PMS quản lý các hoạt động nhận và trả phòng, lưu trữ hồ sơ khách hàng, quản lý thông tin khách sạn, phòng ốc và quản lý công việc của nhân viên.
- CRS là hệ thống đặt phòng trung tâm được hiểu là một hệ thống giúp quản lý toàn bộ thông tin đặt phòng của khách sạn từ tất cả các nguồn, kênh phân phối bán phòng.
    - Sự khác biệt chính giữa hệ thống CRS và PMS nằm ở cách phân bổ, lưu trữ dữ liệu.

Triển khai mô hình:

Mô hình dự đoán hủy đặt phòng không nên được thực hiện riêng lẻ, vì không chắc rằng kết quả thực tế sẽ tác động tốt tới doanh thu.

Bộ phận đặt phòng khách sạn theo cách truyền thống sẽ không tận dụng được ưu thế của mô hình vì khối lượng dữ liệu khá lớn, phức tạp và luôn cập nhật liên tục nên chúng ta tích hợp mô hình dự đoán vào CRS của khách sạn.

→ CRS dự báo nhu cầu ròng chính xác hơn và đưa ra dự báo tổng thể tốt hơn.



Cách thức vận hành mô hình:

Khi khách hàng đặt phòng qua các kênh như Online TA, Offline TA/TO,… nằm trong Phân khúc thị trường, các thông tin khách hàng sẽ được tự động cập nhật và đồng bộ các thông tin này cho toàn bộ kênh phân phối.

Kênh phân phối sẽ tích hợp 2 chiều với hệ thống quản lý khách sạn PMS đồng nghĩa với việc thông tin thay đổi phòng, số lượng phòng trống, giá phòng sẽ được cập nhật dựa trên thông tin từ PMS theo thời gian thực. Điều này sẽ giúp tiết kiệm được rất nhiều thời gian vận hành của bạn

Kênh phân phối hoạt động độc lập hoặc tích hợp 2 chiều với hệ thống PMS

Kênh phân phối sẽ tích hợp qua một hệ thống trung gian trước khi qua PMS đó là CRS – hệ thống đặt phòng trung tâm. CRS đã được tích hợp mô hình dự đoán sẽ tổng hợp toàn bộ các yêu cầu đặt phòng, thay đổi phòng đến từ các kênh và đưa ra dự đoán rằng khách hàng đó có khả năng hủy phòng hay không, sau đó CRS sẽ phân bổ số lượng phòng trống và giá phòng về cho từng khách sạn trong chuỗi.CRS sẽ cung cấp kết nối thông tin đặt phòng từ các kênh phân phối đến PMS và đưa ra báo cáo chi tiết về tình trạng đặt phòng, giá phòng. CRS và PMS hỗ trợ cho nhau để tạo nên sự tối ưu hoàn toàn vận hành cho khách sạn từ lúc nhận đặt phòng đến lúc khách trả phòng.

Nhận xét:

- Mô hình hoạt động tốt sẽ cho được kết quả tốt về việc hủy đặt phòng của khách hàng. Điều này giúp người quản lý khách sạn có thể thực hiện các biện pháp để tránh những trường hợp hủy bỏ tiềm ẩn này, chẳng hạn như cung cấp dịch vụ, giảm giá, được tham gia các hoạt động giải trí miễn phí trong khung viên khách sạn và một số đặc quyền khác.
- Chạy mô hình mỗi ngày nhất giúp cho các nhà quản lý khách sạn có thể phát triển các chính sách hủy đặt phòng, điều này sẽ giúp giảm rủi ro và ít phát sinh chi phí hơn.

VII. KẾT LUẬN VÀ ĐỊNH HƯỚNG TRONG TƯƠNG LAI

Bằng cách sử dụng kết hợp khoa học dữ liệu như trực quan hóa dữ liệu, học máy cùng với các hệ thống như CRS, PMS đã giúp cho việc dự đoán hủy đặt phòng khách sạn.

Các tính năng khác nhau được đưa vào mô hình sẽ có kết quả khác nhau, tầm quan trọng sẽ khác nhau tương ứng với khách sạn, có nghĩa là một mô hình không thể phù hợp với tất cả các khách sạn và do đó, mỗi khách sạn nên có mô hình riêng cho mình.

Mô hình dự đoán này cho phép các nhà quản lý khách sạn giảm thiểu tổn thất doanh thu do việc hủy đặt phòng và giảm thiểu rủi ro liên quan đến việc đặt trước quá nhiều.

Các mô hình hủy đặt phòng cũng cho phép các nhà quản lý khách sạn thực hiện các chính sách hủy đặt phòng ít cứng nhắc hơn, mà không làm tăng sự không chắc chắn. Từ đó có khả năng làm tăng doanh thu hơn.

Định hướng trong tương lai:

- Nghiên cứu sâu hơn về những thông tin có thể có liên quan như thông tin thời tiết, đối thủ cạnh tranh, … để cải thiện hiệu quả mô hình và đo lường ảnh hưởng của các tính năng này trong việc hủy đặt phòng.