

ĐẠI HỌC UEH
TRƯỜNG KINH DOANH
KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH



ĐỒ ÁN
MÔN LẬP TRÌNH PHÂN TÍCH DỮ LIỆU
ĐỀ TÀI: Airlines Customer satisfaction

Giảng viên hướng dẫn: Nguyễn An Tế

Mã học phần: 22C1INF50907002

Nhóm sinh viên thực hiện: Nhóm 6

TP Hồ Chí Minh, ngày 4 tháng 12 năm 2022

MỤC LỤC NỘI DUNG

CHƯƠNG 1. TỔNG QUAN	1
1.1. Giới thiệu bài toán	1
1.2. Tổng quan về bộ dữ liệu	1
CHƯƠNG 2. TIỀN XỬ LÝ DỮ LIỆU.....	4
2.1. Xử lý dữ liệu bị thiếu.....	4
2.2. Rời rạc hóa các cột dữ liệu	5
2.2.1. Rời rạc hóa cột “Arrival Delay in Minutes”	5
2.2.2. Rời rạc hóa cột “Departure Delay in Minutes”	8
2.2.3. Rời rạc hóa cột “Age”	11
2.2.4. Rời rạc hóa cột “Flight Distance”	12
2.2. Xử lý thang đo Likert	14
CHƯƠNG 3. BIỂU ĐỒ PHÂN TÍCH DỮ LIỆU.....	15
3.1. Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng mức độ đánh giá ...	15
3.2. Biểu đồ thể hiện mối liên hệ giữa điểm tiêu chí chỗ ngồi thoải mái (Seat comfort) trong từng loại đường bay (Flight Distance) với sự hài lòng chung cho chuyến bay (satisfaction)	17

3.3. Biểu đồ thể hiện mối liên hệ giữa điểm tiêu chí chỗ để chân (Leg room service) trong từng loại đường bay (Flight Distance) với sự hài lòng chung cho chuyến bay (satisfaction).	18
3.4. Biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH satisfied.....	20
3.5. Biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH dissatisfied	22
3.6. Biểu đồ thể hiện sự hài lòng của hành khách đối với các loại hình du lịch	23
3.7. Biểu đồ thể hiện mức độ hài lòng ở các hạng vé.....	24
3.8. Biểu đồ biểu diễn mối tương quan giữa các tiêu chí đánh giá chuyến bay	26
CHƯƠNG 4. CHUYỂN ĐỔI DỮ LIỆU PHÂN LOẠI THÀNH DẠNG SỐ.....	28
CHƯƠNG 5. PHÂN LỚP	31
5.1. Train, test sets	31
5.2. Xây dựng mô hình	32
5.2.1. Phân lớp bằng phương pháp K-NN classification	32
5.2.2. Phân lớp bằng phương pháp Decision Tree	33
5.2.3. Phân lớp bằng phương pháp Support Vector Machine	34
5.2.4. Phân lớp bằng phương pháp Naive Bayes	35
5.3. Đánh giá mô hình	35

5.4. Áp dụng mô hình để dự báo sự đánh giá của khách hàng	36
---	----

CHƯƠNG 6. PHÂN CỤM	39
---------------------------------	-----------

6.1. Phương pháp K-Means	40
--------------------------------	----

6.2. Phương án 2	43
------------------------	----

6.3. Biểu diễn kết quả phân cụm	47
---------------------------------------	----

TÀI LIỆU THAM KHẢO	52
---------------------------------	-----------

BẢNG PHÂN CÔNG	53
-----------------------------	-----------

MỤC LỤC HÌNH ẢNH

Hình 1: Kết quả chạy code Python in ra số giá trị bị thiếu trong data	4
Hình 2: Chạy code Python xóa dữ liệu bị thiếu trong cột 'Arrival Delay in Minutes'	5
Hình 3: Kết quả chạy code Python in ra điểm dao động của thời gian delay khi máy bay hạ cánh đối với khách hàng hài lòng	5
Hình 4: Kết quả chạy code Python in ra điểm dao động của thời gian delay khi máy bay hạ cánh đối với khách hàng không hài lòng	5
Hình 5: Kết quả chạy code Python in ra số lượng đánh giá sự hài lòng của khách hàng khi thời gian hạ cánh của máy bay không bị delay.....	6
Hình 6: Code Python trực quan hóa sự hài lòng của khách hàng khi thời gian hạ cánh không bị delay.	6
Hình 7: Code Python in ra giá trị trung vị của 'Arrival Delay in Minutes'.....	7
Hình 8: Code Python tiến hành phân khoảng 'Arrival Delay in Minutes' và thay thế giá trị thành giá trị vừa phân khoảng	7
Hình 9: Dữ liệu sau khi rời rạc hóa cột 'Arrival Delay in Minutes'	8
Hình 10: Kết quả chạy code Python in ra điểm dao động của thời gian delay khi máy bay cất cánh đối với khách hàng hài lòng	8
Hình 11: Kết quả chạy code Python in ra điểm dao động của thời gian delay khi máy bay cất cánh đối với khách hàng không hài lòng	8
Hình 12: Kết quả chạy code Python in ra số lượng đánh giá sự hài lòng của khách hàng khi thời gian cất cánh của máy bay không bị delay.....	9
Hình 13: Code Python trực quan hóa sự hài lòng của khách hàng khi thời gian cất cánh không bị delay.	9
Hình 14: Code Python in ra giá trị trung vị của 'Departure Delay in Minutes'	10

Hình 15: Code Python tiến hành phân khoảng 'Departure Delay in Minutes' và thay thế giá trị thành giá trị vừa phân khoảng.....	10
Hình 16: Dữ liệu sau khi rời rạc hóa cột 'Departure Delay in Minutes'	11
Hình 17: Code Python chia đều các khoảng và thay thế cột 'Age' bằng cột 'Age Group'. ..	11
Hình 18: Dữ liệu sau khi rời rạc hóa cột 'Age'.	12
Hình 19: Code Python in ra tổng số giá trị duy nhất của biến 'Flight Distance'.	12
Hình 20: Code Python phân loại đường bay	13
Hình 21: Kết quả sau khi phân loại đường bay	13
Hình 22: Code Python tạo 1 danh sách gồm tên các cột dữ liệu	14
Hình 23: Code Python xử lý các giá trị 0 cho từng cột dữ liệu tiêu chí đánh giá	14
Hình 24: Kết quả bộ dữ liệu sau khi xử lý giá trị 0	15
Hình 25: In ra bảng 2 chiều thể hiện quan hệ số lượng giữa mức độ đánh giá và nhóm tuổi khách hàng.....	15
Hình 26: Code Python vẽ biểu đồ thể hiện quan hệ số lượng giữa mức độ đánh giá và nhóm tuổi khách hàng.....	16
Hình 27: Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng mức độ đánh giá.	16
Hình 28: Code Python vẽ biểu đồ thể hiện mối liên hệ giữa tiêu chí ‘Seat comfort’ trong từng loại ‘Flight Distance’ với sự hài lòng chung cho chuyến bay (satisfaction).....	17
Hình 29: Code Python vẽ biểu đồ thể hiện mối liên hệ giữa tiêu chí ‘Leg room service’ trong từng loại ‘Flight Distance’ với sự hài lòng chung cho chuyến bay (satisfaction). ..	18
Hình 30: Code Python in ra các cột có chứa thang đo từ 1 đến 5	20
Hình 31: Code Python đếm số lượng các đánh giá từ 1 đến 5 với từng tiêu chí.....	20
Hình 32: Kết quả sau khi	21

Hình 33: Code Python vẽ biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH satisfied	21
Hình 34: Code Python đếm số lượng khách đánh giá ở các dịch vụ khi mức độ hài lòng của khách hàng là ‘dissatisfied’	22
Hình 35: Kết quả số lượng khách đánh giá ở các dịch vụ khi mức độ hài lòng của khách hàng là ‘dissatisfied’	22
Hình 36: Code Python vẽ biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH dissatisfied	22
Hình 37: Code Python vẽ biểu đồ thể hiện sự hài lòng của hành khách đối với các loại hình du lịch.	23
Hình 38: Code Python vẽ biểu đồ thể hiện mức độ hài lòng ở các hạng vé.....	25
Hình 39: Code Python để vẽ biểu đồ biểu diễn mối tương quan giữa các tiêu chí đánh giá chuyến bay	26
Hình 40: Code Python chuyển đổi dữ liệu phân loại thành dạng số (1).....	29
Hình 41: Code Python chuyển đổi dữ liệu phân loại thành dạng số (2).....	29
Hình 42: Code Python đặt biến targets, features	31
Hình 43: Code Python tách dữ liệu thành tập train, tập test.....	31
Hình 44: In ra kết quả phân vùng khung dữ liệu ban đầu thành 4 tập dữ liệu khác nhau.	32
Hình 45: Code Python tạo ra bộ phân lớp kNN	32
Hình 46: Code Python để tính các điểm chỉ số (kNN)	32
Hình 47: Chỉ số thu được khi $k = 1$	33
Hình 48: Chỉ số thu được khi $k = 2$	33
Hình 49: Chỉ số thu được khi $k = 3$	33
Hình 50: Chỉ số thu được khi $k = 4$	33

Hình 51: Code Python tạo ra bộ phân lớp Decision Tree.....	34
Hình 52: Code Python để tính các điểm chỉ số (DecisionTree)	34
Hình 53: Chỉ số thu được (Decision Tree)	34
Hình 54: Code Python tạo ra bộ phân lớp SVM.....	34
Hình 55: Code Python để tính các điểm chỉ số (SVM)	34
Hình 56: Chỉ số thu được (SVM)	35
Hình 57: Code Python tạo ra bộ phân lớp Naive Bayes	35
Hình 58: Code Python để tính các điểm chỉ số (Naive Bayes)	35
Hình 59: Chỉ số thu được (Naive Bayes)	35
Hình 60: Code Python chạy mô hình dự báo với 2 phương pháp đã chọn	36
Hình 61: Chuyển đổi dạng số sang dạng chữ của mảng yhat.....	37
Hình 62: Code Python in ra bảng dự đoán theo mô hình Decision Tree	37
Hình 63: Kết quả dự báo với Decision Tree.....	38
Hình 64: Code Python in ra bảng dự đoán theo mô hình SVM	38
Hình 65: Kết quả dự báo với SVM.....	39
Hình 66: Code Python lập 1 DataFrame là bản sao của bộ dữ liệu gốc	40
Hình 67: Bộ dữ liệu mới.....	40
Hình 68: Code Python tạo bộ phân cụm kMeans	41
Hình 69: Code Python in ra số phần tử của mỗi cluster.....	41
Hình 70: Kết quả số phần tử của mỗi cluster	41
Hình 71: Code python in ra tọa độ của 2 trọng tâm	42
Hình 72: Kết quả tọa độ của 2 trọng tâm.....	42
Hình 73: Code Python đối chiếu kết quả phân cụm với bộ dữ liệu gốc.....	42

Hình 74: Silhouette score khi phân thành 2 cụm cho bộ dữ liệu 21 biến features	43
Hình 75: Tạo dataframe với 2 biến ‘Class’ và ‘Gate Location’	44
Hình 76: Dataframe thu được	44
Hình 77: Code Python biểu diễn kết quả Silhouette Score dưới dạng đồ thị	45
Hình 78: Từ trái qua phải, từ trên xuống dưới lần lượt là Silhouette score tương ứng với số cụm $k = 2$, $k = 3$, $k = 4$, $k = 5$	45
Hình 79: Với $k = 5$, Silhouette score = 0.602.....	46
Hình 80: Phân cụm bộ dữ liệu data_cluster2.....	46
Hình 81: Số phần tử của 5 cụm	47
Hình 82: Tọa độ của 5 trọng tâm của mỗi cụm).....	47
Hình 83: Code Python biểu diễn kết quả phân cụm	48
Hình 84: Kết quả phân cụm.....	48
Hình 85: Tần số của các điểm trên đồ thị	49

MỤC LỤC BẢNG VÀ BIỂU ĐỒ

Bảng 1: Bộ dữ liệu thông tin chi tiết về các chuyến bay	4
Bảng 2: Chuyển đổi dữ liệu phân loại thành dạng số	29
Biểu đồ 1: Đánh giá sự hài lòng của KH khi thời gian hạ cánh không bị delay	6
Biểu đồ 2: Biểu đồ đánh giá sự hài lòng của KH khi thời gian cất cánh không bị delay....	9
Biểu đồ 3: Biểu đồ thể hiện mối liên hệ giữa tiêu chí ‘Seat comfort’ trong từng loại ‘Flight Distance’ với sự hài lòng chung cho chuyến bay (satisfaction).	17
Biểu đồ 4: Biểu đồ thể hiện mối liên hệ giữa tiêu chí ‘Leg room service’ trong từng loại ‘Flight Distance’ với sự hài lòng chung cho chuyến bay (satisfaction)	19
Biểu đồ 5: Biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH satisfied.	21
Biểu đồ 6: Biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH dissatisfied ..	23
Biểu đồ 7: Biểu đồ thể hiện sự hài lòng của hành khách đối với các loại hình du lịch.....	24
Biểu đồ 8: Biểu đồ thể hiện mức độ hài lòng ở các hạng vé	25
Biểu đồ 9: Biểu đồ biểu diễn mối tương quan giữa các tiêu chí đánh giá chuyến bay	27
Biểu đồ 10: Biểu đồ Heatmap thể hiện hệ số tương quan giữa các biến.....	30

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu bài toán

Trong bài dự án này, nhóm chúng tôi chọn bộ dữ liệu có tên Invistico Airline với nội dung về việc khách hàng đánh giá chất lượng chuyến bay dựa trên nhiều tiêu chí khác nhau. Bộ dữ liệu này được đăng tải trên trang Kaggle bởi tác giả SAYANTAN JANA:

<https://www.kaggle.com/datasets/sjleshrrac/airlines-customer-satisfaction>

Đây là bộ dữ liệu đã được dán nhãn, với biến phân loại là biến Satisfaction (khách hàng hài lòng/không hài lòng với chuyến bay). Đối với bộ dữ liệu này, nhóm sẽ tiến hành các công việc sau:

- Kiểm tra bộ dữ liệu
- Tiền xử lý bộ dữ liệu
- Trực quan hóa dữ liệu để tìm ra xu hướng, điểm nổi bật trong bộ dữ liệu
- Phân lớp dữ liệu: chia bộ dữ liệu thành 2 tập con (tập huấn luyện và tập kiểm tra), áp dụng nhiều phương pháp phân lớp (Supervised Learning) để dự đoán các đối tượng trong tập kiểm tra hài lòng hay không hài lòng với chuyến bay, từ đó xác định đâu là phương pháp phân lớp phù hợp nhất với bộ dữ liệu này
- Phân cụm dữ liệu: bỏ qua biến phân loại Satisfaction, xem bộ dữ liệu là chưa được dán nhãn, áp dụng phương pháp phân cụm để dán nhãn các đối tượng, sau đó đối chiếu lại với kết quả dán nhãn gốc để đo độ chính xác mà phương pháp phân cụm đối với bộ dữ liệu.

1.2. Tổng quan về bộ dữ liệu

Bộ dữ liệu bao gồm thông tin chi tiết về thông tin chuyến bay cũng như đánh giá của khách hàng (trên thang điểm 1-5) về nhiều tiêu chí liên quan đến chuyến bay mà họ đã đi.

Bộ dữ liệu gồm 129880 đối tượng, với 23 biến, cụ thể:

Tên biến	Giá trị	Loại dữ liệu	Ý nghĩa
satisfaction	- satisfied - dissatisfied	Định danh	Mức độ hài lòng tổng quát (hài lòng/không hài lòng) của

			khách hàng đối với chuyến bay
Gender	- Male - Female	Định danh	Giới tính của khách hàng
Customer Type	- Loyal Customer - Disloyal Customer	Định danh	Phân loại độ trung thành của khách hàng (trung thành/ không trung thành)
Age	- Nhỏ nhất: 7 - Lớn nhất: 85	Định lượng	Độ tuổi của khách hàng
Type of Travel	- Personal Travel - Business Travel	Định danh	Mục đích di chuyển bằng máy bay của khách hàng (di chuyển với nhu cầu cá nhân / đi công tác)
Class	- Business - Eco - Eco Plus	Định danh	Hạng vé của khách hàng
Flight Distance	- nhỏ nhất: 50 - lớn nhất: 6951	Định lượng	Độ dài đường bay (km)
Seat comfort	Thang đo 1 - 5 1: vô cùng không hài lòng	Định tính	Sự thoải mái của chỗ ngồi
Departure/Arrival time convenient	5: rất hài lòng		Thời gian hạ cánh/ cất cánh có thuận tiện cho khách hàng không

Food and drink	0: không đánh giá (not rated)		Thức ăn, đồ uống trên chuyến bay
Gate location			Vị trí cổng ra vào, check in có thuận tiện di chuyển cho khách hàng không
Inflight wifi service			Chất lượng wifi trên máy bay
Inflight entertainment			Chất lượng các dịch vụ giải trí cung cấp trong chuyến bay
Online support			Hỗ trợ trực tuyến
Ease of Online booking			Sự dễ dàng khi đặt vé trực tuyến
On-board service			Dịch vụ trên máy bay
Leg room service			Chỗ để chân có thoải mái
Baggage handling			Nơi cất hành lý xách tay
Check-in service			Dịch vụ check-in
Cleanliness			Sự sạch sẽ
Online boarding			Thủ tục lên máy bay trực tuyến
Departure Delay in Minutes	- Nhỏ nhất: 0 - Lớn nhất: 1592	định lượng	Khoảng thời gian khởi hành bị dời (phút)

Arrival Delay in Minutes	- Nhỏ nhất: 0 - Lớn nhất: 1584	Định lượng	Khoảng thời gian hạ cánh bị dòi (phút)
--------------------------	-----------------------------------	------------	--

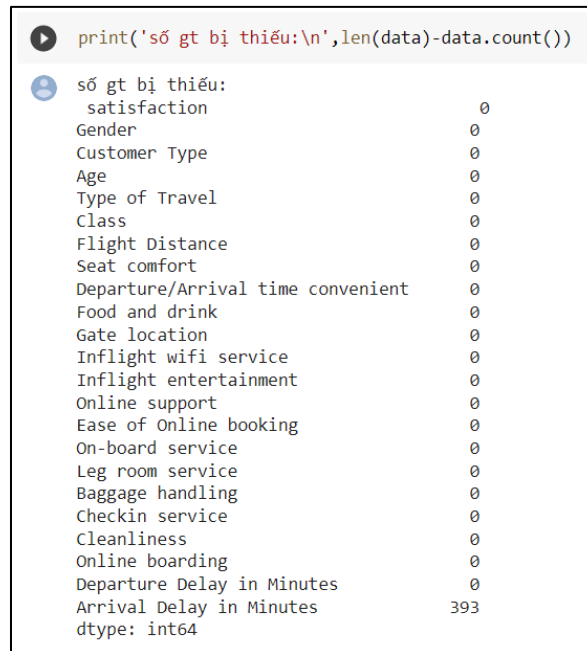
Bảng 1: Bộ dữ liệu thông tin chi tiết về các chuyến bay

CHƯƠNG 2. TIỀN XỬ LÝ DỮ LIỆU

2.1. Xử lý dữ liệu bị thiếu

Nhóm sẽ kiểm tra dữ liệu bị thiếu trên các cột bằng lệnh:

```
len(data)-data.count()
```



```
print('số gt bị thiếu:\n',len(data)-data.count())
```

```
số gt bị thiếu:
satisfaction                0
Gender                      0
Customer Type               0
Age                        0
Type of Travel              0
Class                      0
Flight Distance             0
Seat comfort                0
Departure/Arrival time convenient 0
Food and drink              0
Gate location               0
Inflight wifi service       0
Inflight entertainment      0
Online support              0
Ease of Online booking      0
On-board service            0
Leg room service            0
Baggage handling            0
Checkin service             0
Cleanliness                 0
Online boarding             0
Departure Delay in Minutes  0
Arrival Delay in Minutes    393
dtype: int64
```

Hình 1: Kết quả chạy code Python in ra số giá trị bị thiếu trong data

Từ kết quả trên, nhóm nhìn thấy chỉ có một cột ‘Arrival Delay in Minutes’ có 393 dòng dữ liệu bị khuyết thiếu, chỉ có 0.3% dữ liệu bị khuyết thiếu so với tổng các dòng dữ liệu được thu thập, nên nhóm sẽ xử lý bằng cách xóa những dòng bị thiếu.

```
data = data.dropna(inplace = False)
data = data.reset_index()
```

```
# Nhận xét: Cột 'Arrival Delay in Minutes' có 393 dòng dữ liệu bị thiếu
# Vì những dữ liệu bị thiếu ít nên ta sẽ xóa những dòng đó

data = data.dropna(inplace = False)
data = data.reset_index()
data = data.drop(columns = 'index')
display(data)
```

Hình 2: Chạy code Python xóa dữ liệu bị thiếu trong cột 'Arrival Delay in Minutes'

2.2. Rời rạc hóa các cột dữ liệu

2.2.1. Rời rạc hóa cột “Arrival Delay in Minutes”

Ở cột 'Arrival Delay in Minutes' và cột 'Departure Delay in Minutes', nhóm thực hiện kiểm tra xem thời gian delay có ảnh hưởng gì tới mức độ hài lòng 'satisfaction' hay không?

Nhóm tiến hành lọc ra dữ liệu có sự hài lòng của Khách hàng là 'satisfied' và thực hiện kiểm tra điểm dao động của thời gian delay khi máy bay hạ cánh, bằng lệnh:

```
# kiểm tra dao động của thời gian delay khi khách hàng đánh giá là 'satisfied'

df = data[data['satisfaction']=='satisfied']
print('Khoảng thời gian hạ cánh bị dời ít nhất đối với các khách hàng hài lòng:', df['Arrival Delay in Minutes'].min())
print('Khoảng thời gian hạ cánh bị dời nhiều nhất đối với các khách hàng hài lòng:', df['Arrival Delay in Minutes'].max())

Khoảng thời gian hạ cánh bị dời ít nhất đối với các khách hàng hài lòng: 0.0
Khoảng thời gian hạ cánh bị dời nhiều nhất đối với các khách hàng hài lòng: 1280.0
```

Hình 3: Kết quả chạy code Python in ra điểm dao động của thời gian delay khi máy bay hạ cánh đối với khách hàng hài lòng

Nhóm nhận thấy rằng khi khách hàng hài lòng với chuyến bay thì thời gian dao động delay khi hạ cánh từ 0 đến 1280 phút (tức là không bị delay cho đến khi delay gần khoảng 1 ngày). Còn khi khách hàng đánh giá sự hài lòng của chuyến bay là 'dissatisfied', nhóm cũng làm thao tác tương tự để kiểm tra:

```
# kiểm tra dao động của thời gian delay khi khách hàng đánh giá là 'dissatisfied'

df1 = data[data['satisfaction']=='dissatisfied']
print('Khoảng thời gian hạ cánh bị dời ít nhất đối với các khách hàng không hài lòng:', df1['Arrival Delay in Minutes'].min())
print('Khoảng thời gian hạ cánh bị dời nhiều nhất đối với các khách hàng không hài lòng:', df1['Arrival Delay in Minutes'].max())

Khoảng thời gian hạ cánh bị dời ít nhất đối với các khách hàng không hài lòng: 0.0
Khoảng thời gian hạ cánh bị dời nhiều nhất đối với các khách hàng không hài lòng: 1584.0
```

Hình 4: Kết quả chạy code Python in ra điểm dao động của thời gian delay khi máy bay hạ cánh đối với khách hàng không hài lòng

Khi này nhóm cũng nhìn thấy được, khi khách hàng đánh giá là không hài lòng với chuyến bay thì thời gian delay khi hạ cánh cũng dao động từ 0 đến 1584 phút (tức là không bị delay cho đến khi delay gần 1 ngày).

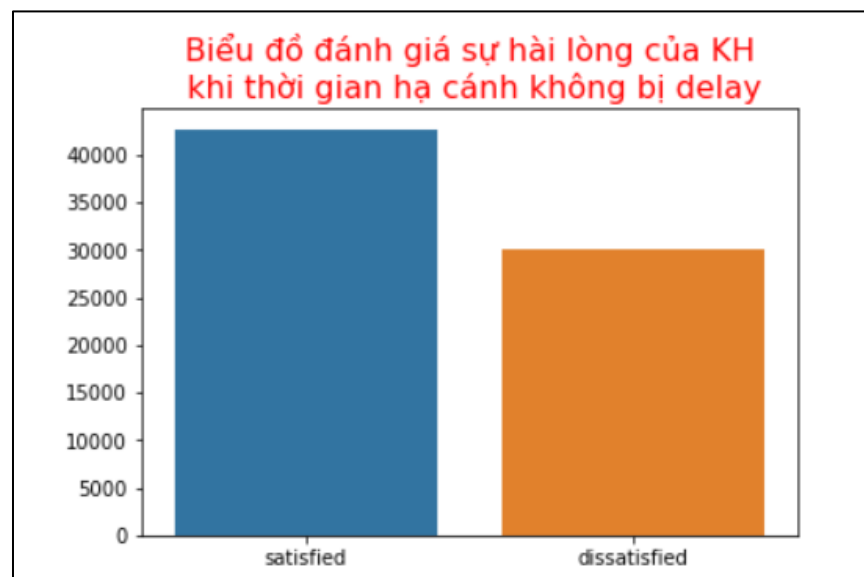
Để chắc chắn hơn về mức độ ảnh hưởng của biến ‘Arrival Delay in Minutes’ và biến ‘satisfaction’, nhóm sẽ thực hiện kiểm tra đếm sự đánh giá của Khách hàng về ‘satisfied’ và ‘dissatisfied’ khi thời gian hạ cánh của máy bay không bị delay.

```
print('Sự hài lòng của KH khi chuyến bay hạ cánh không bị delay:',  
      df[df['Arrival Delay in Minutes']==0].shape[0])  
Sự hài lòng của KH khi chuyến bay hạ cánh không bị delay: 42683  
  
print('Sự không hài lòng của KH khi chuyến bay hạ cánh không bị delay:',  
      df1[df1['Arrival Delay in Minutes']==0].shape[0])  
Sự không hài lòng của KH khi chuyến bay hạ cánh không bị delay: 30070
```

Hình 5: Kết quả chạy code Python in ra số lượng đánh giá sự hài lòng của khách hàng khi thời gian hạ cánh của máy bay không bị delay.

```
: x = ['satisfied', 'dissatisfied']  
y = [df[df['Arrival Delay in Minutes']==0].shape[0], df1[df1['Arrival Delay in Minutes']==0].shape[0]]  
sbn.barplot(x, y)  
plt.title('Biểu đồ đánh giá sự hài lòng của KH khi thời gian hạ cánh không bị delay', color = 'red', fontsize=16)  
plt.show()
```

Hình 6: Code Python trực quan hóa sự hài lòng của khách hàng khi thời gian hạ cánh không bị delay.



Biểu đồ 1: Đánh giá sự hài lòng của KH khi thời gian hạ cánh không bị delay

Từ tính toán và biểu đồ trên, nhóm có thể thấy được khoảng cách khi chuyển bay hạ cánh không bị trì hoãn giữa sự đánh giá của khách hàng là hài lòng và không hài lòng không quá cao, và khi số lượng khách hàng đánh giá không hài lòng cũng khá cao với chuyến bay mà thời gian không bị delay khi hạ cánh. Điều đó có thể thấy được, thời gian hạ cánh của chuyến bay có bị trì hoãn hay không hay bị trì hoãn nhiều hay ít cũng không ảnh hưởng tới sự đánh giá các dịch vụ của chuyến bay, từ đó nhóm thấy được Khách hàng có thể dựa trên nhiều tiêu chí để đánh giá chuyến bay.

Từ nhận xét trên của nhóm, sau khi thảo luận và thống nhất nhóm sẽ chia cột này thành hai khoảng từ [0 - 240] (tức là từ không bị delay đến 4 tiếng) và khoảng [240 - 1584] (tức là từ 4 tiếng trở lên đến giá trị delay cao nhất). Để chứng minh cho điều này, nhóm sẽ thực hiện thao tác tìm ra các giá trị duy nhất ở cột 'Arrival Delay in Minutes' và tìm ra giá trị trung vị.

```
# Từ biểu đồ trên, nhóm thấy được: nếu thời gian delay dưới 4 tiếng KH sẽ ít dissatisfied hơn
# Nếu nhóm chia thành 2 khoảng từ 0-240 (đơn vị phút) và lớn hơn 241
# Lọc xem trong cột 'Arrival Delay in Minutes' có bao nhiêu giá trị chính
unique_Data_Arri_Delay = data['Arrival Delay in Minutes'].unique().tolist()
unique_Data_Arri_Delay.sort()
print('Các giá trị có trong cột (Arrival Delay in Minutes):\n->', unique_Data_Arri_Delay, '\n-----\n')

# Tìm giá trị trung vị (median) để có thể chia đều số lượng các giá trị trong 2 khoảng bằng nhau
print('Giá trị trung vị của bộ dữ liệu:', np.median(unique_Data_Arri_Delay), '\n-----\n')
```

Các giá trị có trong cột (Arrival Delay in Minutes):
-> [0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0, 21.0, 22.0, 23.0, 24.0

Giá trị trung vị của bộ dữ liệu: 235.5

Hình 7: Code Python in ra giá trị trung vị của 'Arrival Delay in Minutes'

Ta có thể nhìn thấy giá trị trung vị gần bằng 4 tiếng (240 phút) và sau đó tiến hành rời rạc hóa cột dữ liệu này.

```
# Từ đó thì ta sẽ chia thành 2 khoảng như sau:
series_bins1 = pd.cut(data['Arrival Delay in Minutes'], [0,241,1585], labels = ['0-240', '241-1584'], right = False)

# Thay thế giá trị của cột 'Arrival Delay in Minutes' thành giá trị vừa phân khoảng
data['Arrival Delay in Minutes'] = series_bins1
print('Ta có được bảng dữ liệu mới:')
display(data)
```

Hình 8: Code Python tiến hành phân khoảng 'Arrival Delay in Minutes' và thay thế giá trị thành giá trị vừa phân khoảng

Nhóm thu được kết quả như sau:

Flight istance	Seat comfort	Departure/Arrival time convenient	Food and drink	...	Online support	Ease of Online booking	On- board service	Leg room service	Baggage handling	Checkin service	Cleanliness	Online boarding	Departure Delay in Minutes	Arrival Delay in Minutes
265	0	0	0	...	2	3	3	0	3	5	3	2	0	0-240
2464	0	0	0	...	2	3	4	4	4	2	3	2	310	241-1584
2138	0	0	0	...	2	2	3	3	4	4	4	2	0	0-240
623	0	0	0	...	3	1	1	0	1	4	1	3	0	0-240
354	0	0	0	...	4	2	2	0	2	4	2	5	0	0-240
...
1731	5	5	5	...	2	2	3	3	4	4	4	2	0	0-240
2087	2	3	2	...	1	3	2	3	3	1	2	1	174	0-240
2320	3	0	3	...	2	4	4	3	4	2	3	2	155	0-240
2450	3	2	3	...	2	3	3	2	3	2	1	2	193	0-240
4307	3	4	3	...	3	4	5	5	5	3	3	3	185	0-240

Hình 9: Dữ liệu sau khi rời rạc hóa cột 'Arrival Delay in Minutes'

2.2.2. Rời rạc hóa cột “Departure Delay in Minutes”

Sau khi rời rạc hóa cột ‘Arrival Delay in Minutes’, nhóm sẽ tiếp tục thực hiện các thao tác kiểm tra và rời rạc hóa cột ‘Departure Delay in Minutes’. Nhóm cũng thu được các kết quả tương tự như trên:

```
# kiểm tra dao động của thời gian delay khi khách hàng đánh giá là 'satisfied'

print('Khoảng thời gian khởi hành bị dờ ít nhất đối với các khách hàng hài lòng:', df['Departure Delay in Minutes'].min())
print('Khoảng thời gian khởi hành bị dờ nhiều nhất đối với các khách hàng hài lòng:', df['Departure Delay in Minutes'].max())

Khoảng thời gian khởi hành bị dờ ít nhất đối với các khách hàng hài lòng: 0
Khoảng thời gian khởi hành bị dờ nhiều nhất đối với các khách hàng hài lòng: 1305
```

Hình 10: Kết quả chạy code Python in ra điểm dao động của thời gian delay khi máy bay cất cánh đối với khách hàng hài lòng

```
# kiểm tra dao động của thời gian delay khi khách hàng đánh giá là 'dissatisfied'

print('Khoảng thời gian khởi hành bị dờ ít nhất đối với các khách hàng không hài lòng:', df1['Departure Delay in Minutes'].min())
print('Khoảng thời gian khởi hành bị dờ nhiều nhất đối với các khách hàng không hài lòng:', df1['Departure Delay in Minutes'].max())

Khoảng thời gian khởi hành bị dờ ít nhất đối với các khách hàng không hài lòng: 0
Khoảng thời gian khởi hành bị dờ nhiều nhất đối với các khách hàng không hài lòng: 1592
```

Hình 11: Kết quả chạy code Python in ra điểm dao động của thời gian delay khi máy bay cất cánh đối với khách hàng không hài lòng

Tương tự như khi kiểm tra thời gian delay khi hạ cánh, thì thời gian delay khi cất cánh so với mức độ hài lòng của Khách hàng cũng dao động từ không bị delay đến khoảng gần

1 ngày, và sự dao động giữa sự hài lòng và sự không hài lòng của Khách hàng là tương tự nhau. Sau đó nhóm cũng thực hiện tính toán xem sự khác biệt của sự đánh giá của khách hàng là ‘satisfied’ và ‘dissatisfied’ với xét trên thời gian không bị delay khi cất cánh để đánh giá mức độ ảnh hưởng của hai biến ‘satisfaction’ và ‘Departure Delay in Minutes’

```
# Để xem đánh giá của KH 'satisfied' khi không bị delay thời gian cất cánh
print('Sự hài lòng của KH khi chuyến bay cất cánh không bị delay:',
      df[df['Departure Delay in Minutes']==0].shape[0])

Sự hài lòng của KH khi chuyến bay cất cánh không bị delay: 41908

# Để xem đánh giá của KH 'dissatisfied' khi không bị delay thời gian cất cánh
print('Sự không hài lòng của KH khi chuyến bay cất cánh không bị delay:',
      df1[df1['Departure Delay in Minutes']==0].shape[0])

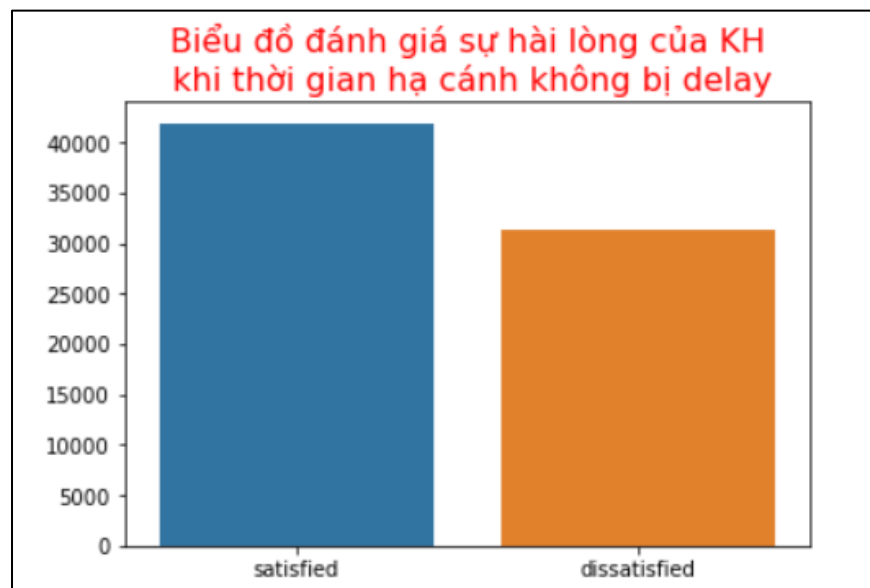
Sự không hài lòng của KH khi chuyến bay cất cánh không bị delay: 31301
```

Hình 12: Kết quả chạy code Python in ra số lượng đánh giá sự hài lòng của khách hàng khi thời gian cất cánh của máy bay không bị delay.

Và trực quan hóa nó để được có sự đánh giá rõ ràng:

```
# Biểu đồ thể hiện mức độ đánh giá của KH khi chuyến bay không bị delay
x = ['satisfied', 'dissatisfied']
y = [df[df['Departure Delay in Minutes']==0].shape[0], df1[df1['Departure Delay in Minutes']==0].shape[0]]
sbn.barplot(x, y)
plt.title('Biểu đồ đánh giá sự hài lòng của KH khi thời gian hạ cánh không bị delay', color = 'red', fontsize=16)
plt.show()
```

Hình 13: Code Python trực quan hóa sự hài lòng của khách hàng khi thời gian cất cánh không bị delay.



Biểu đồ 2: Biểu đồ đánh giá sự hài lòng của KH khi thời gian cất cánh không bị delay.

Từ đó, có thể thấy 2 cột ‘Arrival Delay in Minutes’ và cột ‘Departure Delay in Minutes’ so với cột ‘satisfaction’ không có sự khác biệt là mấy, nên mức độ ảnh hưởng của thời gian delay khi cất cánh so với mức độ hài lòng của khách hàng không ảnh hưởng nhiều. Và tương tự là nhóm có thể nhìn thấy sự đánh giá của khách hàng về chuyến bay có thể dựa trên nhiều yếu tố khác nhau và thời gian delay khi cất và hạ cánh đều không bị ảnh hưởng nhiều.

Khi đó cột ‘Departure Delay in Minutes’ nhóm cũng tiến hành rời rạc hóa nó để gom nhỏ các giá trị có trong cột này.

```
# Từ biểu đồ trên, nhóm thấy được: nếu thời gian delay dưới 4 tiếng KH sẽ ít dissatisfied hơn
# Nếu nhóm chia thành 2 khoảng từ 0-240 (đơn vị phút) và lớn hơn 241
# Học xem trong cột 'Arrival Delay in Minutes' có bao nhiêu giá trị chính
unique_Data_Departure_Delay = data['Departure Delay in Minutes'].unique().tolist()
unique_Data_Departure_Delay.sort()
print('Các giá trị có trong cột (Departure Delay in Minutes):\n->', unique_Data_Departure_Delay, '\n-----\n')

# Tìm giá trị trung vị (median) để có thể chia đều số lượng các giá trị trong 2 khoảng bằng nhau
print('Giá trị trung vị của bộ dữ liệu:', np.median(unique_Data_Departure_Delay), '\n-----\n')

Các giá trị có trong cột (Departure Delay in Minutes):
-> [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
-----
Giá trị trung vị của bộ dữ liệu: 231.5
-----
```

Hình 14: Code Python in ra giá trị trung vị của 'Departure Delay in Minutes'

Ở cột này, nhóm cũng thấy được giá trị trung vị của các giá trị có trong cột này cũng gần 4 tiếng (240 phút), nên nhóm cũng chia khoảng từ không bị delay đến 4 tiếng và từ lớn hơn 4 tiếng đến giá trị lớn nhất có trong bộ dữ liệu này. Sau đó nhóm tiến hành rời rạc hóa dữ liệu trong cột này.

```
# Từ đó thì ta sẽ chia thành 2 khoảng như sau:
series_bins2 = pd.cut(data['Departure Delay in Minutes'], [0,241,1593], labels = ['0-240', '241-1592'], right = False)

# Thay thế giá trị của cột 'Departure Delay in Minutes' thành giá trị vừa phân khoảng
data['Departure Delay in Minutes'] = series_bins2
print('Ta có được bảng dữ liệu mới:')
display(data)
```

Hình 15: Code Python tiến hành phân khoảng 'Departure Delay in Minutes' và thay thế giá trị thành giá trị vừa phân khoảng.

Và thu được kết quả bảng mới như sau:

Flight istance	Seat comfort	Departure/Arrival time convenient	Food and drink	...	Online support	Ease of Online booking	On- board service	Leg room service	Baggage handling	Checkin service	Cleanliness	Online boarding	Departure Delay in Minutes	Arrival Delay in Minutes
265	0	0	0	...	2	3	3	0	3	5	3	2	0-240	0-240
2464	0	0	0	...	2	3	4	4	4	2	3	2	241-1592	241-1584
2138	0	0	0	...	2	2	3	3	4	4	4	2	0-240	0-240
623	0	0	0	...	3	1	1	0	1	4	1	3	0-240	0-240
354	0	0	0	...	4	2	2	0	2	4	2	5	0-240	0-240
...
1731	5	5	5	...	2	2	3	3	4	4	4	2	0-240	0-240
2087	2	3	2	...	1	3	2	3	3	1	2	1	0-240	0-240
2320	3	0	3	...	2	4	4	3	4	2	3	2	0-240	0-240
2450	3	2	3	...	2	3	3	2	3	2	1	2	0-240	0-240

Hình 16: Dữ liệu sau khi rời rạc hóa cột 'Departure Delay in Minutes'

2.2.3. Rời rạc hóa cột "Age"

Sau khi hoàn thành rời rạc hóa dữ liệu 2 cột 'Arrival Delay in Minutes' và cột 'Departure Delay in Minutes', nhóm nhận thấy cột 'Age' cũng có nhiều giá trị và muốn phân nhóm ra để dễ dàng cho việc phân lớp / phân cụm sau này.

Nhóm sẽ sử dụng hàm `qcut` để việc chia nhóm được đều ra các nhóm tuổi và sau đó thay thế vào cột 'Age' bằng cột 'Age Group'

```
# Dùng hàm qcut để các khoảng sẽ tự chia đều
series_bins3 = pd.qcut(data['Age'], 4, labels = ["7-27", "28-40", "41-51", "52-85"])
data['Age'] = series_bins3
data.rename(columns = {'Age': 'Age Group'}, inplace = True)

print('Ta có được bảng dữ liệu mới:')
display(data)
```

Hình 17: Code Python chia đều các khoảng và thay thế cột 'Age' bằng cột 'Age Group'.

Và thu được kết quả như sau:

Ta có được bảng dữ liệu mới:

	satisfaction	Gender	Customer Type	Age Group	Type of Travel	Class	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	...	Online support	Ease of Online booking	On-board service
0	satisfied	Female	Loyal Customer	52-85	Personal Travel	Eco	265	0	0	0	...	2	3	3
1	satisfied	Male	Loyal Customer	41-51	Personal Travel	Business	2464	0	0	0	...	2	3	4
2	satisfied	Female	Loyal Customer	7-27	Personal Travel	Eco	2138	0	0	0	...	2	2	3
3	satisfied	Female	Loyal Customer	52-85	Personal Travel	Eco	623	0	0	0	...	3	1	1
4	satisfied	Female	Loyal Customer	52-85	Personal Travel	Eco	354	0	0	0	...	4	2	2
...
129482	satisfied	Female	disloyal Customer	28-40	Personal Travel	Eco	1731	5	5	5	...	2	2	3
129483	dissatisfied	Male	disloyal Customer	52-85	Personal Travel	Business	2087	2	3	2	...	1	3	2
129484	dissatisfied	Male	disloyal Customer	52-85	Personal Travel	Eco	2320	3	0	3	...	2	4	4
129485	dissatisfied	Male	disloyal Customer	52-85	Personal Travel	Eco	2450	3	2	3	...	2	3	3

Hình 18: Dữ liệu sau khi rời rạc hóa cột 'Age'.

2.2.4. Rời rạc hóa cột “Flight Distance”

```
1 len(data['Flight Distance'].unique())
```

5397

Hình 19: Code Python in ra tổng số giá trị duy nhất của biến 'Flight Distance'.

Biến Flight Distance thuộc kiểu dữ liệu liên tục, và trong bộ dữ liệu đang quan sát, biến Flight Distance có tổng cộng 5397 giá trị duy nhất. Vì thế, ta phải rời rạc hóa biến này để thuận lợi cho việc phân tích dữ liệu hơn. Sau khi tìm hiểu, trong lĩnh vực hàng không, độ dài đường bay được chia thành 3 loại: đường bay ngắn, đường bay trung bình và đường bay dài. Cụ thể:

- Đường bay ngắn có độ dài dưới 600-800 nmi (hoặc dưới 1,1100 - 1,500 km)
- Đường bay dài có độ dài từ 2,200 - 2,600 nmi (hoặc từ 4,100 - 4,800 km)
- Đường bay trung bình nằm giữa 800 - 2,200 nmi (từ 1,500 km đến 4,100 km)

Với thông tin trên, nhóm sẽ phân thành 3 khoảng cho biến Flight Distance:

- Short-haul: < 1500
- Medium-haul: $1500 \leq x \leq 4100$
- Long-haul: > 4100

```
bin_range_Flight_distance = [0,1500,4100,max(data['Flight Distance'])]
bin_names_Flight_distance = ['short-haul', 'medium-haul', 'long-haul']

data['Flight Distance'] = pd.cut(np.array(data['Flight Distance']),
                                bins=bin_range_Flight_distance,
                                labels=bin_names_Flight_distance)

data
```

Hình 20: Code Python phân loại đường bay

Flight Distance	Flight Distance
265	short-haul
2464	medium-haul
2138	medium-haul
623	short-haul
354	short-haul
...	...
1731	medium-haul
2087	medium-haul
2320	medium-haul
2450	medium-haul
4307	long-haul

Hình 21: Kết quả sau khi phân loại đường bay

2.2. Xử lý thang đo Likert

Bộ dữ liệu đang quan sát có tổng cộng 13 biến có kiểu dữ liệu định tính, sử dụng thang đo Likert với mức độ từ 1-5 (không hài lòng - rất hài lòng). Tuy nhiên, trong bộ dữ liệu có xuất hiện giá trị 0, đây không phải là giá trị thể hiện mức độ hài lòng của khách hàng về từng tiêu chí đánh giá mà là thể hiện khách hàng không có đánh giá tiêu chí đó (not rated). Nếu không xử lý giá trị 0 này thì khi tiến hành phân lớp, phân cụm sẽ gây ra hiểu lầm 0 cũng là 1 điểm trong thang điểm đánh giá, từ đó có thể làm sai lệch kết quả phân tích, khiến kết quả bị mâu thuẫn, phi logic.

Vì thế, ta sẽ xem 0 là dữ liệu bị thiếu. Nhóm quyết định sử dụng phương pháp thay thế dữ liệu bị thiếu này bằng 1 giá trị cụ thể, ở đây sẽ thay 0 thành 3. Nhóm chọn số 3 vì trong thang đo Likert, 3 nằm ở giữa, thể hiện thái độ bình thường, không yêu thích cũng không khó chịu, tương tự với việc khách hàng không quá coi trọng tiêu chí đánh giá đó và cảm thấy tiêu chí đó bình thường, không ảnh hưởng đến kết quả đánh giá chung về trải nghiệm bay của họ.

- Đầu tiên, ta tạo 1 danh sách gồm tên các cột dữ liệu:

```
l_columnsname = data.columns.tolist()
```

Hình 22: Code Python tạo 1 danh sách gồm tên các cột dữ liệu

- Ta tạo vòng lặp for để thay thế các giá trị 0 thành 3 cho từng cột dữ liệu sử dụng thang đo Likert:

```
1 for column in range (columnsname.index('Seat comfort'), len(columnsname)-2):  
2     data.loc[data[columnsname[column]] == 0, columnsname[column]] = 3  
3 display(data)
```

Hình 23: Code Python xử lý các giá trị 0 cho từng cột dữ liệu tiêu chí đánh giá

Seat comfort	Departure/Arrival time convenient	Food and drink	...	Online support	Ease of Online booking	On-board service	Leg room service	Baggage handling	Checkin service	Seat comfort	Departure/Arrival time convenient	Food and drink	...	Online support	Ease of Online booking	On-board service	Leg room service	Baggage handling
0	0	0	...	2	3	3	0	3	5	3	3	3	...	2	3	3	3	3
0	0	0	...	2	3	4	4	4	2	3	3	3	...	2	3	4	4	4
0	0	0	...	2	2	3	3	4	4	3	3	3	...	2	2	3	3	4
0	0	0	...	3	1	1	0	1	4	3	3	3	...	3	1	1	3	1
0	0	0	...	4	2	2	0	2	4	3	3	3	...	4	2	2	3	2
0	0	0	...	2	2	5	4	5	5
0	0	0	...	5	5	5	0	5	5	5	5	5	...	2	2	3	3	4
0	0	0	...	5	5	5	0	5	5	2	3	2	...	1	3	2	3	3
0	0	0	...	2	2	3	3	4	5	3	3	3	...	2	4	4	3	4
0	0	0	...	5	4	4	0	1	5	3	2	3	...	2	3	3	2	3
0	0	0	...	2	2	2	4	5	3	3	4	3	...	3	4	5	5	5

Hình 24: Kết quả bộ dữ liệu sau khi xử lý giá trị 0

CHƯƠNG 3. BIỂU ĐỒ PHÂN TÍCH DỮ LIỆU

3.1. Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng mức độ đánh giá

Mức độ hài lòng "satisfaction" được hiển thị trên trục hoành, đối với mỗi mức độ có 4 thanh tương ứng với nhóm tuổi (7-27, 28-40, 41-51, 52-85). Biến 'Age Group' (nhóm tuổi) mang tính liên tục nên màu sắc được sử dụng theo thang màu tuần tự.

Ở đây ta dùng hàm `pandas.crosstab` để tạo bảng chéo (một bảng 2 chiều) để tổng hợp số lượng khách hàng đối với từng mức độ đánh giá khác nhau. Cũng như phân tích mối quan hệ giữa mức độ đánh giá và nhóm tuổi khách hàng.

Age Group	7-27	28-40	41-51	52-85
satisfaction				
dissatisfied	18948	16366	10702	12589
satisfied	14723	17028	20825	18306

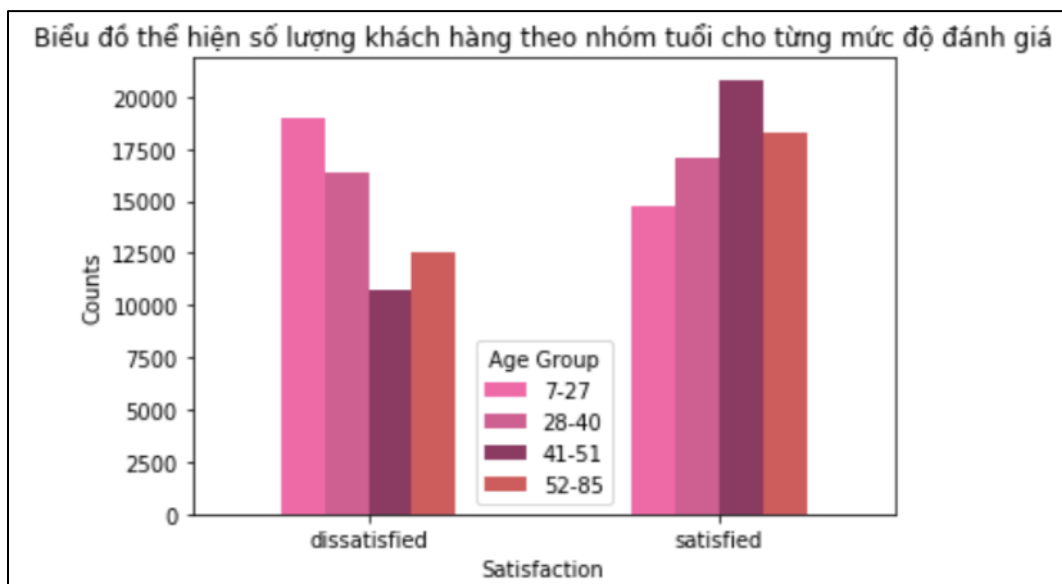
Hình 25: In ra bảng 2 chiều thể hiện quan hệ số lượng giữa mức độ đánh giá và nhóm tuổi khách hàng.

Sau đó tạo biểu đồ bằng hàm `DataFrame.plot.bar`

```
myfield1 = data['satisfaction']
myfield2 = data['Age Group']
cross = pd.crosstab(myfield1, myfield2)
barplot = cross.plot.bar(color=[(238/255, 106/255, 167/255), (205/255, 96/255, 144/255), (139/255, 58/255, 98/255), (205/255, 92/255, 92/255)], rot=0)
plt.title('Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng mức độ đánh giá')
plt.xlabel('Satisfaction')
plt.ylabel('Counts')
```

Hình 26: Code Python vẽ biểu đồ thể hiện quan hệ số lượng giữa mức độ đánh giá và nhóm tuổi khách hàng.

Ta thu được kết quả sau:



Hình 27: Biểu đồ thể hiện số lượng khách hàng theo nhóm tuổi cho từng mức độ đánh giá.

Biểu đồ trên cho ta thấy mối quan hệ giữa độ tuổi của khách hàng và đánh giá của họ với chuyến bay. Chiều cao của mỗi thanh tương ứng với số lượng của mỗi nhóm tuổi. Chiều rộng của các thanh không có ý nghĩa trong biểu đồ thanh cụm. Nhìn chung khách hàng có nhóm tuổi càng lớn sẽ có xu hướng đánh giá hài lòng đối với chuyến bay. Ngược lại khách hàng có nhóm tuổi càng trẻ có đánh giá không hài lòng nhiều hơn. Đặc biệt độ tuổi từ 41-51 có sự chênh lệch mức độ đánh giá khá lớn, cụ thể số lượng đánh giá hài lòng lớn gần gấp 2 lần đánh giá không hài lòng. Trong khi đó độ tuổi 28-40 không có sự chênh lệch nhiều về mức độ đánh giá.

3.2. Biểu đồ thể hiện mối liên hệ giữa điểm tiêu chí chỗ ngồi thoải mái (Seat comfort) trong từng loại đường bay (Flight Distance) với sự hài lòng chung cho chuyến bay (satisfaction)

Khi đi chuyến bằng máy bay, chỗ ngồi luôn là một trong những yếu tố ảnh hưởng đến tâm trạng của khách hàng. Tuy nhiên, tùy vào độ dài của đường bay mà khách hàng sẽ có mức đánh giá khác nhau về chỗ ngồi. Thông thường, chuyến bay càng dài thì chỗ ngồi thoải mái sẽ khiến khách hàng hài lòng về trải nghiệm bay hơn; ngược lại, khi phải ngồi lâu trên 1 chuyến bay nhưng chỗ ngồi quá chật, không thoải mái khiến khách hàng không vui vẻ và có xu hướng không muốn đi lại hãng bay đó nữa.

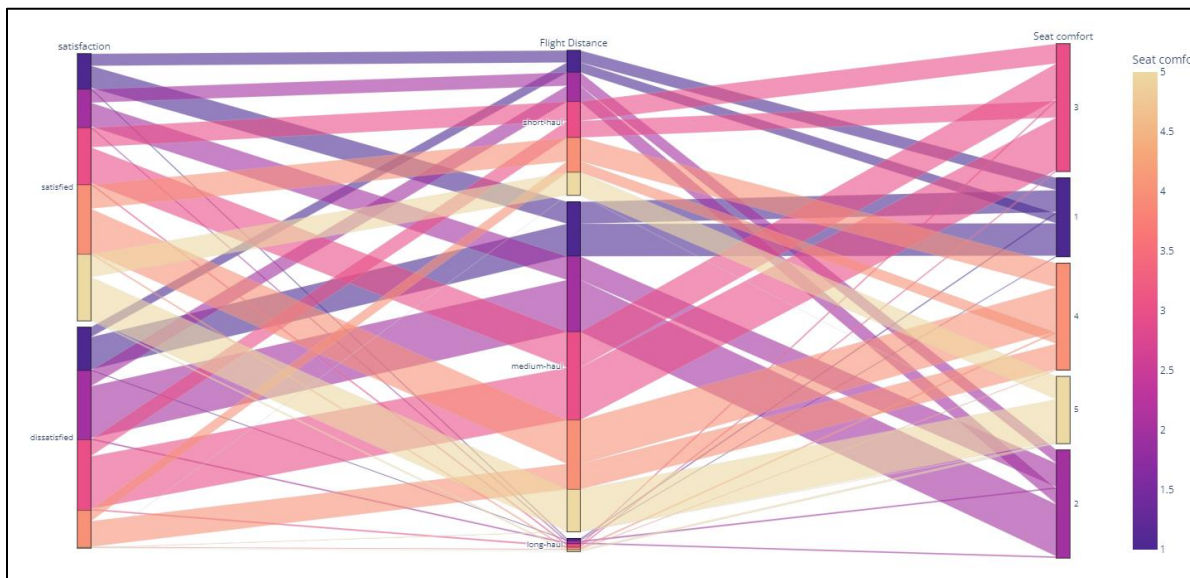
Ta sẽ xem xét 2 biến Seat comfort và Leg room service với bộ lọc là Flight Distance và satisfaction. Nhóm sử dụng thư viện Plotly để vẽ biểu đồ Parallel sets - 1 dạng biểu đồ mở rộng của biểu đồ tỷ lệ (pie chart).

```
import plotly.express as px

fig = px.parallel_categories(data,
                           dimensions=['satisfaction', 'Flight Distance', 'Seat comfort'],
                           color = 'Seat comfort', color_continuous_scale="agsunset",
                           title='Biểu đồ thể hiện mối liên hệ giữa điểm tiêu chí chỗ ngồi thoải mái (Seat comfort) trong từng loại đường bay (Flight Distance) với sự hài lòng chung cho chuyến bay (satisfaction)',
                           width=1500, height=800)

fig.show()
```

Hình 28: Code Python vẽ biểu đồ thể hiện mối liên hệ giữa tiêu chí 'Seat comfort' trong từng loại 'Flight Distance' với sự hài lòng chung cho chuyến bay (satisfaction).



Biểu đồ 3: Biểu đồ thể hiện mối liên hệ giữa tiêu chí 'Seat comfort' trong từng loại 'Flight Distance' với sự hài lòng chung cho chuyến bay (satisfaction).

Việc khách hàng cảm thấy có thoải mái với chỗ ngồi trong từng loại đường bay có ảnh hưởng đến kết quả đánh giá độ hài lòng chung của khách hàng đối với cả chuyến bay:

Đối với chuyến bay đường ngắn:

- Ở mức điểm 1-3, mức điểm thể hiện khách hàng không cảm thấy thoải mái với chỗ ngồi, sự thoải mái của chỗ ngồi không ảnh hưởng nhiều đến kết quả đánh giá tổng về chuyến bay. Cụ thể, tỉ lệ giữa việc khách hàng hài lòng với chuyến bay và không hài lòng với chuyến bay là xấp xỉ 50:50.

- Trong khi đó, khi khách hàng cảm thấy chỗ ngồi thoải mái thì tỷ lệ khách hàng hài lòng với chuyến bay áp đảo so với sự không hài lòng: Ở mức điểm 4, tỉ lệ hài lòng : không hài lòng là khoảng 3:1. Ở mức điểm 5, hầu như không có khách hàng nào cảm thấy không hài lòng với chuyến bay (6000 + hài lòng : 30 không hài lòng)

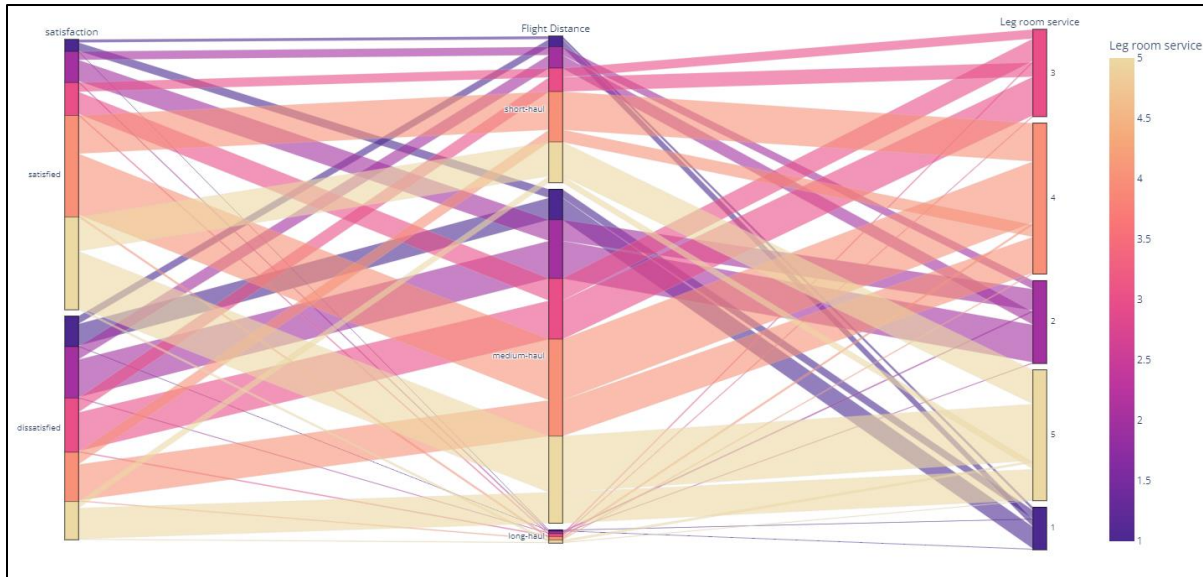
Đối với chuyến bay đường trung bình: Xu hướng cũng giống như chuyến bay đường ngắn, điểm số chỗ ngồi càng cao thì khách hàng thường sẽ hài lòng về chuyến bay

Đối với chuyến bay đường dài: việc đánh giá của khách hàng về sự hài lòng với chuyến bay phụ thuộc vào nhiều yếu tố khác, do vậy ở các điểm số 1-3 về chỗ ngồi, ta không nhìn thấy được sự liên hệ rõ ràng đến việc khách hàng hài lòng với chuyến bay; còn ở mức điểm 4-5, việc chỗ ngồi thoải mái một phần ảnh hưởng đến sự hài lòng của khách hàng với chuyến bay (hài lòng : không hài lòng \approx 2:1)

3.3. Biểu đồ thể hiện mối liên hệ giữa điểm tiêu chí chỗ để chân (Leg room service) trong từng loại đường bay (Flight Distance) với sự hài lòng chung cho chuyến bay (satisfaction).

```
1 import plotly.express as px
2
3 fig = px.parallel_categories(data,
4                             dimensions=['satisfaction', 'Flight Distance', 'Leg room service'],
5                             color = 'Leg room service',
6                             color_continuous_scale="agsunset",
7                             title='Biểu đồ thể hiện mối liên hệ giữa điểm tiêu chí chỗ để chân (Leg room serv
8                             width=1500, height=800
9                             )
10 fig.show()
```

Hình 29: Code Python vẽ biểu đồ thể hiện mối liên hệ giữa tiêu chí 'Leg room service' trong từng loại 'Flight Distance' với sự hài lòng chung cho chuyến bay (satisfaction).



Biểu đồ 4: Biểu đồ thể hiện mối liên hệ giữa tiêu chí ‘Leg room service’ trong từng loại ‘Flight Distance’ với sự hài lòng chung cho chuyến bay (satisfaction)

Việc khách hàng cảm thấy có thoải mái với chỗ để chân trong từng loại đường bay có ảnh hưởng đến kết quả đánh giá độ hài lòng chung của khách hàng đối với cả chuyến bay:

Đối với chuyến bay đường ngắn:

- Ở mức điểm 1-3, mức điểm thể hiện khách hàng không thích dịch vụ về chỗ để chân, yếu tố chỗ để chân không ảnh hưởng nhiều đến kết quả đánh giá tổng về chuyến bay. Cụ thể, tỉ lệ giữa việc khách hàng hài lòng với chuyến bay và không hài lòng với chuyến bay là xấp xỉ 50:50.

- Trong khi đó, khi khách hàng có thái độ tích cực với dịch vụ chỗ để chân thì tỷ lệ khách hàng hài lòng với chuyến bay áp đảo so với sự không hài lòng: Ở mức điểm 4, tỉ lệ hài lòng : không hài lòng là khoảng 3:1. Ở mức điểm 5, tỉ lệ hài lòng : không hài lòng là khoảng 8:1.

Đối với chuyến bay đường trung bình: Xu hướng cũng giống như chuyến bay đường ngắn, điểm số dịch vụ chỗ để chân càng cao thì khách hàng thường sẽ hài lòng về chuyến bay

Đối với chuyến bay đường dài: Việc đánh giá của khách hàng về sự hài lòng với chuyến bay phụ thuộc vào nhiều yếu tố khác, do vậy ở các điểm số 1-3 về chỗ để chân, ta

không nhìn thấy được sự liên hệ rõ ràng đến việc khách hàng hài lòng với chuyến bay; còn ở mức điểm 4-5, chất lượng dịch vụ chỗ để chân một phần ảnh hưởng đến sự hài lòng của khách hàng với chuyến bay (hài lòng : không hài lòng $\approx 2:1$)

3.4. Biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH satisfied

Để đánh giá xem xu hướng của khách hàng khi đánh giá toàn bộ chuyến bay là ‘satisfied’ hoặc đánh giá là ‘dissatisfied’ thì xét trên từng dịch vụ của chuyến bay thì xu hướng đánh giá trong khoảng từ 1 đến 5 của từng dịch vụ thì sẽ rơi vào thang điểm nào là nhiều nhất?

Dưới đây là danh sách các cột có chứa thang đo từ 1 đến 5 tức là chứa các đánh giá của khách hàng trên từng dịch vụ của chuyến bay.

```
# Chọn ra các cột có chứa thang đo từ 1 đến 5

lst_col = data.columns.tolist()
lst = lst_col[7:21]
lst

['Seat comfort',
 'Departure/Arrival time convenient',
 'Food and drink',
 'Gate location',
 'Inflight wifi service',
 'Inflight entertainment',
 'Online support',
 'Ease of Online booking',
 'On-board service',
 'Leg room service',
 'Baggage handling',
 'Checkin service',
 'Cleanliness',
 'Online boarding']
```

Hình 30: Code Python in ra các cột có chứa thang đo từ 1 đến 5

Sau đó, đếm số lượng các đánh giá từ 1 đến 5 mà khách hàng đánh giá khi mức độ hài lòng của toàn chuyến bay là ‘satisfied’.

```
df = pd.DataFrame()
df['x'] = ['1','2','3','4','5']
for temp in lst:
    d1 = data[data['satisfaction']=='satisfied'].groupby(temp).count()
    df[temp] = d1['satisfaction'].values
display(df)
```

Hình 31: Code Python đếm số lượng các đánh giá từ 1 đến 5 với từng tiêu chí

Ta thu được kết quả bảng như sau:

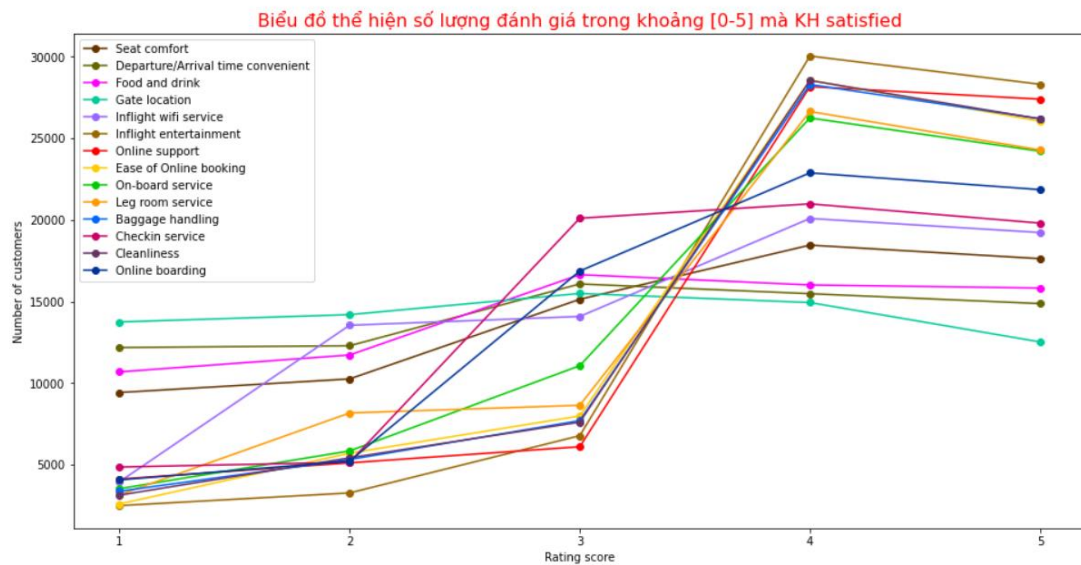
x	Seat comfort	Departure/Arrival time convenient	Food and drink	Gate location	Inflight wifi service	Inflight entertainment	Online support	Ease of Online booking	On-board service	Leg room service	Baggage handling	Checkin service	Cleanliness	Cabin
1	9416	12175	10682	13744	3939	2479	4104	2582	3515	3145	3362	4846	3122	
2	10249	12283	11717	14193	13542	3257	5103	5695	5834	8161	5305	5137	5408	
3	15133	16076	16646	15492	14080	6768	6090	7978	11061	8632	7693	20106	7600	
4	18457	15478	16013	14934	20092	30056	28176	28574	26257	26653	28310	20987	28555	
5	17627	14870	15824	12519	19229	28322	27409	26053	24215	24291	26212	19806	26197	

Hình 32: Kết quả sau khi

Và tiến hành biểu diễn nó dưới dạng biểu đồ để dễ dàng thấy được xu hướng của khách hàng

```
plt.figure(figsize=(16, 8))
colors = ['#663300', '#666600', '#FF00FF', '#00CC99', '#9966FF', '#996600',
          '#FF0000', '#FFCC00', '#00CC00', '#FF9900', '#0066FF', '#CC0066',
          '#663366', '#003399']
for index, temp in enumerate(lst):
    plt.plot(df['x'], df[temp], marker='o', label=temp, color=colors[index])
plt.title('Biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH satisfied', color='r', fontsize=16)
plt.xlabel('Rating score')
plt.ylabel('Number of customers')
plt.legend()
plt.show()
```

Hình 33: Code Python vẽ biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH satisfied



Biểu đồ 5: Biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH satisfied.

Từ biểu đồ trên, ta đưa ra một nhận xét về xu hướng của khách hàng khi đánh giá mức độ hài lòng của mình về chuyến bay là 'satisfied' thì các dịch vụ trên chuyến bay sẽ

được đánh giá nhiều nhất trong khoảng từ 4 đến 5. Khách hàng có thể đánh giá một số dịch vụ dưới 3 nhưng, những dịch vụ còn lại có thể sẽ đánh giá từ 4 trở lên. Và sẽ có một vài trường hợp khách hàng đánh giá nhiều dịch vụ trên 4 nhưng lại không ‘satisfied’ thì có thể do thời gian delay ảnh hưởng hoặc là do đánh giá khách quan.

3.5. Biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH dissatisfied

Tương tự như trên thì ta cũng đếm số lượng khách đánh giá ở các dịch vụ khi mức độ hài lòng của khách hàng là ‘dissatisfied’.

```
df_ = pd.DataFrame()
df_['x'] = ['1','2','3','4','5']
for temp in lst:
    d1 = data[data['satisfaction']=='dissatisfied'].groupby(temp).count()
    df_[temp] = d1['satisfaction'].values
df_
```

Hình 34: Code Python đếm số lượng khách đánh giá ở các dịch vụ khi mức độ hài lòng của khách hàng là ‘dissatisfied’

Và có bảng kết quả như sau:

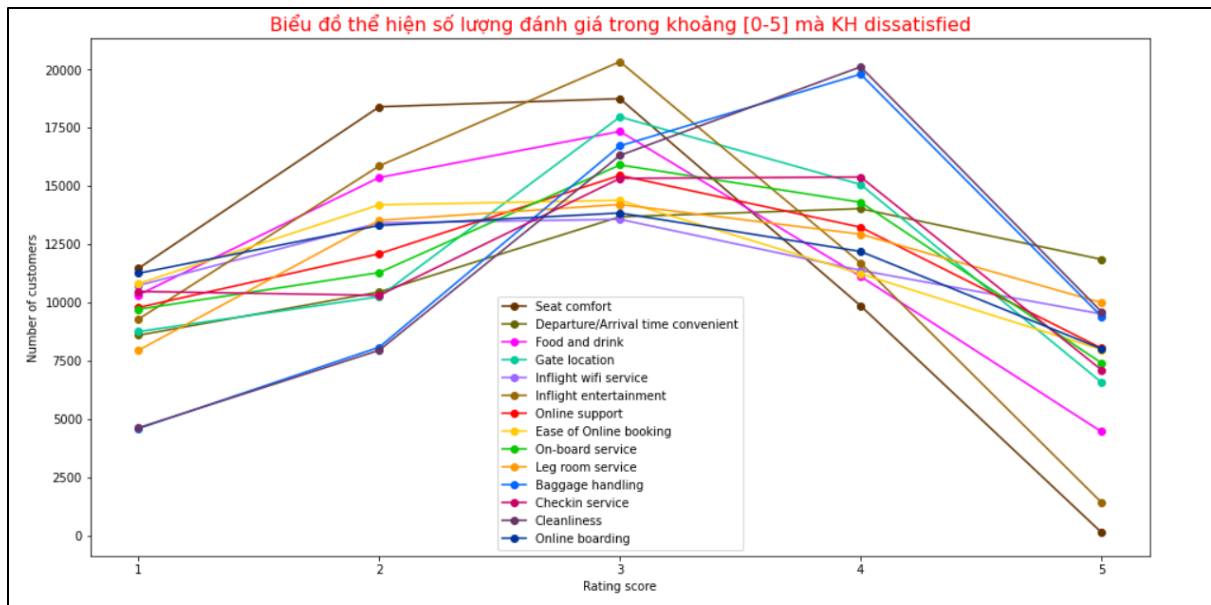
x	Seat comfort	Departure/Arrival time convenient	Food and drink	Gate location	Inflight wifi service	Inflight entertainment	Online support	Ease of Online booking	On-board service	Leg room service	Baggage handling	Checkin service	Cleanliness
1	11466	8596	10326	8753	10731	9289	9786	10815	9708	7953	4594	10476	4624
2	18396	10452	15361	10248	13415	15861	12093	14192	11283	13522	8083	10306	7953
3	18744	13678	17341	17961	13568	20333	15454	14384	15903	14207	16720	15325	16312
4	9858	14026	11116	15063	11382	11696	13230	11233	14301	12930	19797	15385	20110
5	141	11853	4461	6580	9509	1426	8042	7981	7410	9993	9411	7113	9606

Hình 35: Kết quả số lượng khách đánh giá ở các dịch vụ khi mức độ hài lòng của khách hàng là ‘dissatisfied’

Thu được biểu đồ:

```
plt.figure(figsize=(16, 8))
colors = ['#663300', '#666600', '#FF00FF', '#00CC99', '#9966FF', '#996600',
          '#FF0000', '#FFCC00', '#00CC00', '#FF9900', '#0066FF', '#CC0066',
          '#663366', '#003399']
for index, temp in enumerate(lst):
    plt.plot(df_['x'], df_[temp], marker='o', label=temp, color=colors[index])
plt.title('Biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH dissatisfied', color='r', fontsize=16)
plt.xlabel('Rating score')
plt.ylabel('Number of customers')
plt.legend()
plt.show()
```

Hình 36: Code Python vẽ biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH dissatisfied



Biểu đồ 6: Biểu đồ thể hiện số lượng đánh giá trong khoảng [0-5] mà KH dissatisfied

Ngược lại, khi khách hàng đánh giá là ‘dissatisfied’ thì các đánh giá của khách hàng ở các dịch vụ sẽ tập trung vào mức 3 là nhiều nhất. Và số lượng khách hàng đánh giá mức 1 cũng tăng cao so với khi khách hàng hài lòng với chuyến bay và ở mức 5 cũng giảm nhiều so với ở trên.

3.6. Biểu đồ thể hiện sự hài lòng của hành khách đối với các loại hình du lịch

Biểu đồ thể hiện sự hài lòng của hành khách đối với các loại hình du lịch với thang đo phía trong là tổng số lượng hành khách của mỗi loại hình du lịch thang đo phía ngoài là tổng số lượng hài lòng và không hài lòng của mỗi loại hình du lịch đó .

Ta sử dụng thư viện Plotly , biểu đồ sunburst với dữ liệu đầu vào data cho 2 cột Type of Travel và cột satisfaction.

```
fig = px.sunburst(data,
                  path=["Type of Travel","satisfaction"],
                  title='Biểu đồ thể hiện sự hài lòng của hành khách đối với các loại hình du lịch'
                  )
fig.show()
```

Hình 37: Code Python vẽ biểu đồ thể hiện sự hài lòng của hành khách đối với các loại hình du lịch.



Biểu đồ 7: Biểu đồ thể hiện sự hài lòng của hành khách đối với các loại hình du lịch

Biểu đồ cho thấy sự chênh lệch giữa 2 loại hình Business travel và Personal travel: Phần lớn là Business travel với số lượng 89,693; còn lại là 40,187 của Personal travel. Trong đó đánh giá không hài lòng của loại hình Personal lớn hơn số đánh giá hài lòng ($21,456 > 18,731$); mặt khác ở loại hình Business travel, số lượng đánh giá hài lòng lớn hơn số lượng đánh giá không hài lòng ($82,356 > 37,337$).

3.7. Biểu đồ thể hiện mức độ hài lòng ở các hạng vé

Việc lựa chọn hạng vé máy bay luôn là vấn đề cấp thiết để khách hàng có được chuyến bay thật sự thoải mái khi di chuyển. Theo bộ dữ liệu nhóm tìm hiểu, khách hàng có 3 sự lựa chọn về hạng vé là: vé hạng thương gia (Business Class), vé hạng phổ thông (Economy Class) và vé (Eco Plus). Mỗi loại vé có những mức giá và những dịch vụ tương ứng đi kèm.

Dựa vào cột dữ liệu satisfaction ta sẽ thấy rõ mức độ hài lòng của khách hàng đối với các dịch vụ và mức giá của hạng vé. Đầu tiên ta chọn biến satisfaction là biến target.

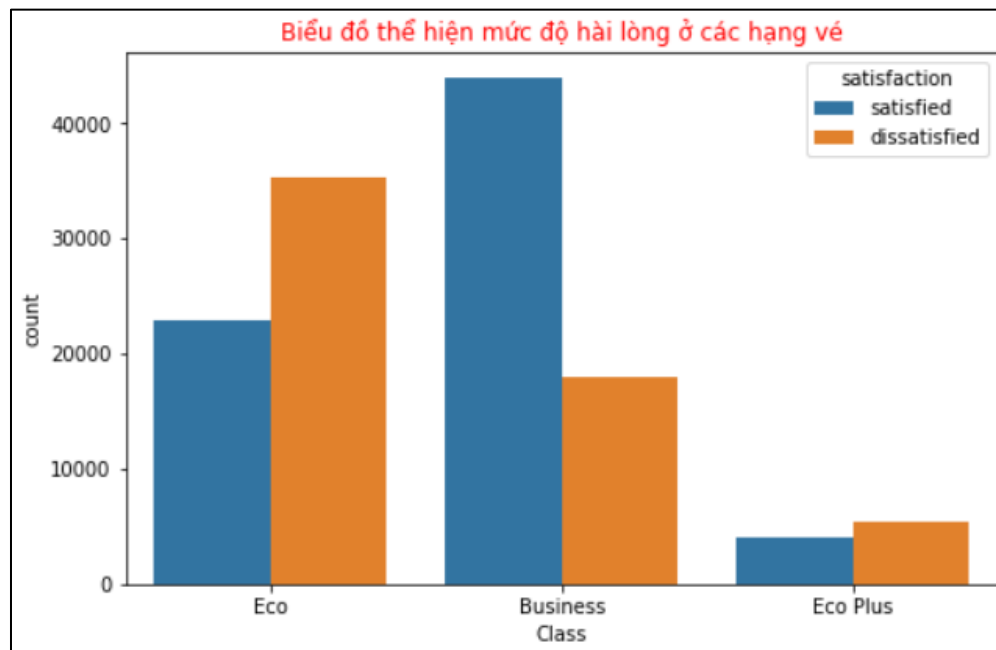
```
target = 'satisfaction'
```

Sử dụng thư viện **sbn** để vẽ biểu đồ count.plot với title “*Biểu đồ thể hiện mức độ hài lòng ở các hạng vé*”, biến x là dữ liệu cột Class và biến y là số lượng của cột satisfaction (= target)

```
plt.figure(figsize = (8, 5))
plt.title('Biểu đồ thể hiện mức độ hài lòng ở các hạng vé',color = 'r')
x = 'Class'
y = target
sbn.countplot(x, hue = y, data = data)
plt.show()
```

Hình 38: Code Python vẽ biểu đồ thể hiện mức độ hài lòng ở các hạng vé

Kết quả nhận được:



Biểu đồ 8: Biểu đồ thể hiện mức độ hài lòng ở các hạng vé

Biểu đồ cho ta thấy số lượng khách hàng đặt vé hạng Eco Plus vô cùng thấp so với 2 hạng vé còn lại (ước tính khoảng 10000 khách lựa chọn). Ngược lại với vé hạng thương gia (Business Class) và vé hạng phổ thông (Eco Class) được nhiều khách hàng lựa chọn (ước tính mỗi loại vé có khoảng 60000 khách). Tuy nhiên sự đánh giá từ khách hàng tham gia chuyến bay với 2 hạng vé trên có sự chênh lệch đáng kể. Vé hạng Business Class có lượng khách hàng đánh giá hài lòng về dịch vụ chuyến bay rất cao so với lượng khách đánh

giá không hài lòng (cao gấp 2 lần). Còn vé hạng Economy Class thì lại có lượng khách hàng không hài lòng về dịch vụ của chuyến bay cao hơn lượng khách đánh giá hài lòng.

Từ những số liệu trên ta thấy vé hạng thương gia là loại vé được ưa chuộng nhất và có những gói dịch vụ tốt nhất khiến cho khách hàng cảm thấy thoải mái khi tham gia chuyến bay dài.

3.8. Biểu đồ biểu diễn mối tương quan giữa các tiêu chí đánh giá chuyến bay

Việc biểu diễn mối tương quan giữa các tiêu chí đánh giá chuyến bay cho thấy được rằng những tiêu chí nào là có mối tương quan chặt chẽ với nhau nhất và tiêu chí nào là có tương quan kém nhất so với tất cả các tiêu chí đánh giá còn lại. Điều này có thể cung cấp cho chúng ta cái nhìn rõ ràng hơn về bộ dữ liệu.

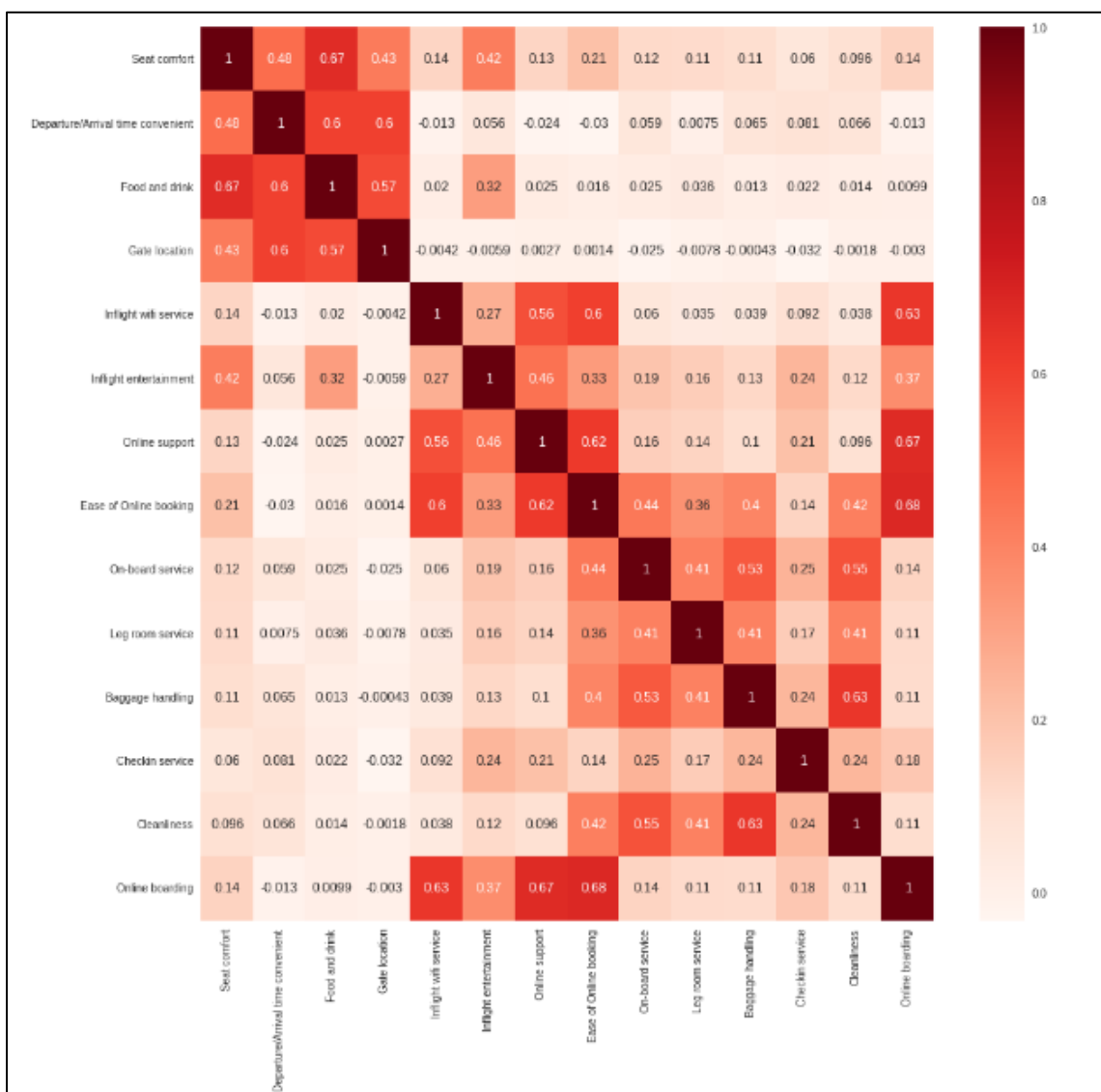
Đầu tiên, ta sử dụng hàm `list` để liệt kê ra những cột dữ liệu chứa tiêu chí đánh giá (thang đo từ 1-5). Sau đó ta sử dụng thư viện `seaborn` để vẽ biểu đồ heatmap dựa trên những cột dữ liệu vừa liệt kê.

```
[ ] list1 = list(data.columns)

plt.subplots(figsize=(15,15))
sbn.heatmap(data[list1].corr(), annot = True, cmap = plt.cm.Red)
plt.show()
```

Hình 39: Code Python để vẽ biểu đồ biểu diễn mối tương quan giữa các tiêu chí đánh giá chuyến bay

Kết quả thu được:



Biểu đồ 9: Biểu đồ biểu diễn mối tương quan giữa các tiêu chí đánh giá chuyến bay

Chẳng hạn như đối với các trường hợp có mối tương quan mạnh với nhau thì tất cả các tiêu chí đó cùng có đánh giá tốt đến từ khách hàng.

Tiêu chí 'Checkin service' (Dịch vụ checkin) có mối tương quan kém nhất với tất cả các tiêu chí khác. Đây là tiêu chí không được lòng khách hàng cho lắm, nhưng nó vẫn ở mức chấp nhận được và các tiêu chí khác sẽ bù qua đắp lại cho tiêu chí này.

Tiêu chí 'Food and Drink' (Đồ ăn và nước uống) có mối tương quan mạnh nhất với tiêu chí 'Seat Comfort' (Chỗ ngồi thoải mái) (0.67).

Tiêu chí 'Ease of Online booking' và 'Online support' có mối tương quan mạnh mẽ nhất với tiêu chí 'Online boarding' (0.68 và 0.67)

Tiêu chí 'Inflight wifi service', 'Online support', 'Ease of Online booking', 'Baggage Handling', 'Cleanliness' và 'Online boarding' có mối tương quan khá chặt chẽ với nhau (~0.6)

CHƯƠNG 4. CHUYỂN ĐỔI DỮ LIỆU PHÂN LOẠI THÀNH DẠNG SỐ

Trước khi tiến hành phân lớp, phân cụm, nhóm sẽ chuyển các cột có kiểu dữ liệu phân loại sang dạng số, để thỏa điều kiện dữ liệu đối với một số hàm xây dựng mô hình, phân cụm.

satisfaction	<ul style="list-style-type: none"> – satisfied → 1 – dissatisfied → 0
Gender	<ul style="list-style-type: none"> – Male → 1 – Female → 0
Customer Type	<ul style="list-style-type: none"> – Loyal Customer → 1 – disloyal Customer → 0
Type of Travel	<ul style="list-style-type: none"> – Business Travel → 1 – Personal Travel → 0
Class	<ul style="list-style-type: none"> – Business → 1 – Eco → 2 – Eco Plus → 3
Age Group	<ul style="list-style-type: none"> – 7-27 → 1 – 28-40 → 2 – 41-51 → 3 – 51-85 → 4
Flight Distance	<ul style="list-style-type: none"> – short-haul → 1 – medium-haul → 2

	– long-haul → 3
Departure Delay in Minutes	– 0-240 → 1 – 241-1592 → 2
Arrival Delay in Minutes	– 0-240 → 1 – 241-1584 → 2

Bảng 2: Chuyển đổi dữ liệu phân loại thành dạng số

```

mapping_satisfaction = {"satisfied": 1 , "dissatisfied": 0}
data['satisfaction'] = data['satisfaction'].map(mapping_satisfaction)

mapping_gender = {"Male": 1 , "Female": 0}
data['Gender'] = data['Gender'].map(mapping_gender)

mapping_customer_type = {"Loyal Customer": 1 , "disloyal Customer": 0}
data['Customer Type'] = data['Customer Type'].map(mapping_customer_type)

mapping_type_of_travel = {"Business travel": 1 , "Personal Travel": 0}
data['Type of Travel'] = data['Type of Travel'].map(mapping_type_of_travel)

mapping_class = {"Business": 1 , "Eco Plus": 2 , "Eco": 3}
data['Class'] = data['Class'].map(mapping_class)

mapping_age_group = {"7-27": 1 , "28-40": 2 , "41-51": 3 , "52-85": 4}
data['Age Group'] = data['Age Group'].map(mapping_age_group)

mapping_flight_distance = {"short-haul": 1 , "medium-haul": 2 , "long-haul": 3}
data['Flight Distance'] = data['Flight Distance'].map(mapping_flight_distance)

mapping_departure_delay_in_minutes = {"0-240": 1 , "241-1592": 2}
data['Departure Delay in Minutes'] = data['Departure Delay in Minutes'].map(mapping_departure_delay_in_minutes)

mapping_arrival_delay_in_minutes = {"0-240": 1 , "241-1584": 2}
data['Arrival Delay in Minutes'] = data['Arrival Delay in Minutes'].map(mapping_arrival_delay_in_minutes)

```

Hình 40: Code Python chuyển đổi dữ liệu phân loại thành dạng số (1)

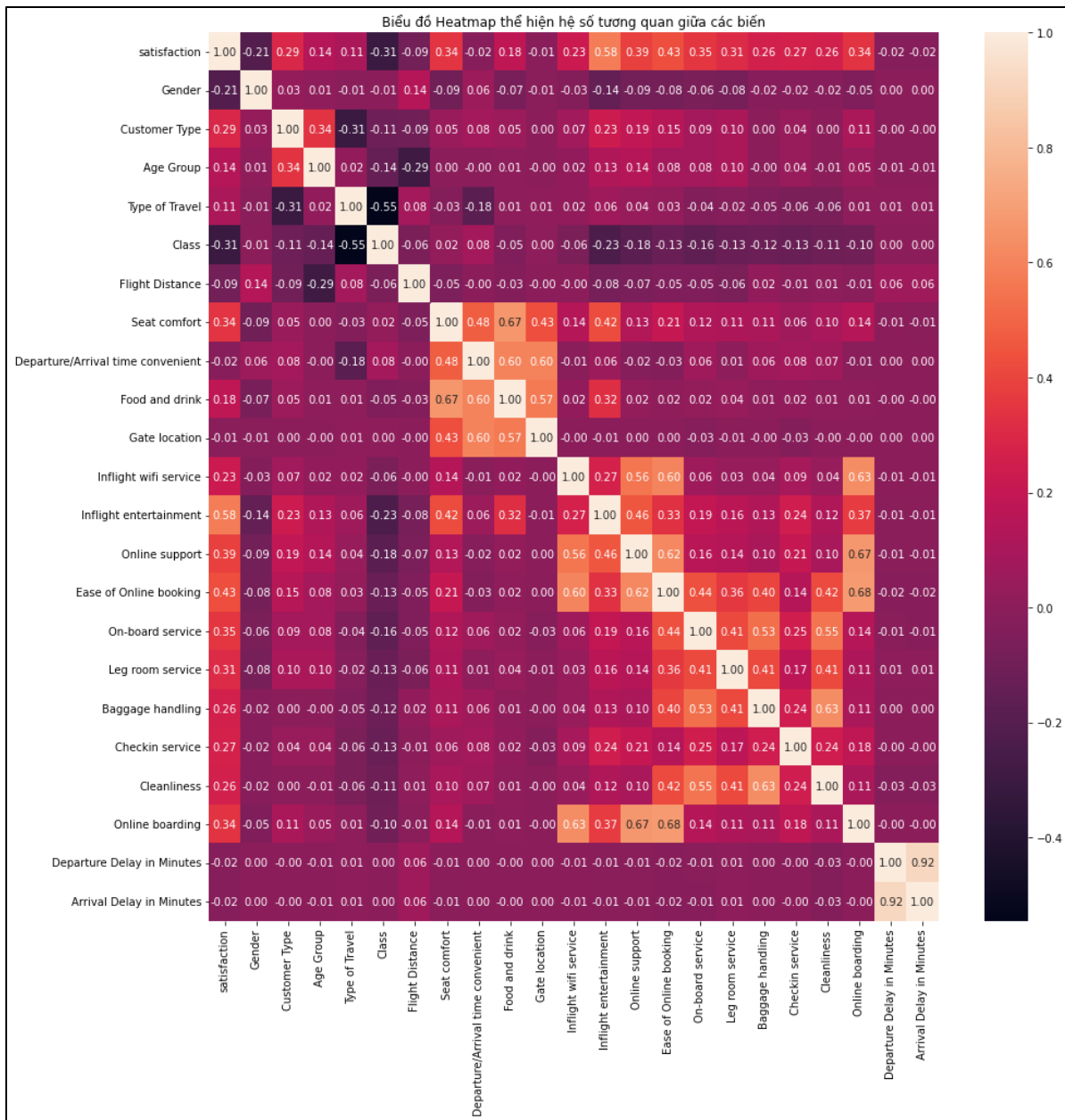
```

for i in columnsname:
    if type(data[i]) != 'int64':
        data[i] = data[i].astype('int64')

```

Hình 41: Code Python chuyển đổi dữ liệu phân loại thành dạng số (2)

Nhóm thực hiện vẽ biểu đồ Heatmap để đánh giá hệ số tương quan giữa các biến với nhau, đặc biệt giữa các biến độc lập với biến phụ thuộc ‘satisfaction’.



Biểu đồ 10: Biểu đồ Heatmap thể hiện hệ số tương quan giữa các biến

Qua biểu đồ ta có thể nhận thấy, hệ số tương quan giữa 2 biến thời gian delay rất cao, vì thế ta sẽ giảm chiều dữ liệu bằng cách bỏ đi 1 biến, đó là biến Departure Delay in Minutes.

CHƯƠNG 5. PHÂN LỚP

5.1. Train, test sets

Việc tạo các mẫu khác nhau để đào tạo và thử nghiệm giúp chúng ta đánh giá hiệu suất dự đoán của mô hình và xác nhận mô hình. Trước đó chúng ta cần xây dựng một mô hình máy học, đầu tiên nên cung cấp tập dữ liệu vào thuật toán máy học. Tập dữ liệu ban đầu để giúp thuật toán “học” gọi là tập dữ liệu huấn luyện (training dataset). Và để “kiểm tra” xem mô hình có hoạt động khi cần thiết hay không, ta dùng đến tập dữ liệu thử nghiệm (testing dataset) cho mô hình sau khi mô hình được xây dựng.

Tuy nhiên ta không thể đánh giá dựa vào dữ liệu đã sử dụng để đào tạo, ta cần tách các thuộc tính (features) cho mô hình khỏi biến mục tiêu (target). Gán cho biến X là mảng đầu vào chứa dữ liệu tất cả các cột ngoại trừ ‘satisfaction’ là thuộc tính mà ta muốn sử dụng để xây dựng mô hình. Biến y chứa các giá trị đích là ‘satisfaction’, vì mục tiêu của ta muốn nhận của dữ liệu thể hiện đánh giá hài lòng/ không hài lòng của khách hàng.

```
[ ] # Đặt biến targets, features
X = data.drop('satisfaction',axis=1).values
y = data['satisfaction'].values
```

Hình 42: Code Python đặt biến targets, features

Sử dụng hàm `train_test_split` của thư viện Scikit-learn (sklearn) để tách dữ liệu thành các tập con train và test với một lệnh gọi hàm duy nhất dưới đây. Tham số `test_size` cho phép chỉ định kích thước của bộ kiểm tra đầu ra. Ta đặt cho tỷ lệ 0.2 (20%) các quan sát sẽ có trong bộ kiểm tra, và 0.8 (80%) quan sát được chỉ định cho đào tạo.

```
# Tách dữ liệu thành tập train, tập test
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2, random_state = 100)
```

Hình 43: Code Python tách dữ liệu thành tập train, tập test

Từ đó ta phân vùng khung dữ liệu ban đầu thành 4 tập dữ liệu khác nhau, với:

```
print (X_train.shape, y_train.shape)
print (X_test.shape, y_test.shape)

(103589, 22) (103589,)
(25898, 22) (25898,)
```

Hình 44: In ra kết quả phân vùng khung dữ liệu ban đầu thành 4 tập dữ liệu khác nhau

- Tập X_train có 103589 dòng dữ liệu đào tạo, 22 cột ứng với 22 thuộc tính
- Tập y_train có 103589 dòng dữ liệu đào tạo, 1 cột ứng với 1 biến mục tiêu
- Tập X_test có 25898 dòng dữ liệu kiểm thử, 22 cột ứng với 22 thuộc tính
- Tập y_test có 25898 dòng dữ liệu kiểm thử, 1 cột ứng với 1 biến mục tiêu

5.2. Xây dựng mô hình

Với các tập X_train , y_train , X_test , y_test đã chia, ta phân lớp bằng các phương pháp và đánh giá mỗi phương pháp bằng các giá trị Accuracy, precision, recall, f1-score.

5.2.1. Phân lớp bằng phương pháp K-NN classification

```
k=2
model_knn=KNeighborsClassifier(n_neighbors=k)
model_knn.fit(X_train,y_train)
KNeighborsClassifier(n_neighbors=2)
```

Hình 45: Code Python tạo ra bộ phân lớp kNN

Tùy chọn số phân lớp $K = 1, 2, 3 \dots n$, tức là với mỗi điểm trong tập X_test, ta chỉ xét K điểm trong tập X_train gần nhất và lấy label của điểm đó để dự đoán cho điểm test này

Sử dụng các phương thức để tính các điểm chỉ số :

```
yhat = model_knn.predict(X_test)
print(f'Độ chính xác ={accuracy_score(y_test,yhat)*100:.2f}%')
print(f'precision={precision_score(y_test, yhat)*100:.2f}%')
print(f'recall={recall_score(y_test, yhat)*100:.2f}%')
print(f'f1-score={f1_score(y_test, yhat)*100:.2f}%')
```

Hình 46: Code Python để tính các điểm chỉ số (kNN)

Kết quả phân lớp của mô hình phân lớp này là:

Với $k = 1$: tại cùng 1 bộ dữ liệu train, test ta thu được chỉ số :

```
Độ chính xác =90.25%  
precision=91.60%  
recall=90.53%  
f1-score=91.06%
```

Hình 47: Chỉ số thu được khi $k = 1$

Với $k = 2$: tại cùng 1 bộ dữ liệu train, test ta thu được chỉ số :

```
Độ chính xác =89.76%  
precision=96.18%  
recall=84.70%  
f1-score=90.08%
```

Hình 48: Chỉ số thu được khi $k = 2$

Với $k = 3$: tại cùng 1 bộ dữ liệu train, test ta thu được chỉ số :

```
Độ chính xác =91.29%  
precision=93.65%  
recall=90.24%  
f1-score=91.92%
```

Hình 49: Chỉ số thu được khi $k = 3$

Với $k = 4$: tại cùng 1 bộ dữ liệu train, test ta thu được chỉ số :

```
Độ chính xác =90.77%  
precision=95.85%  
recall=86.94%  
f1-score=91.17%
```

Hình 50: Chỉ số thu được khi $k = 4$

5.2.2. Phân lớp bằng phương pháp Decision Tree

Khởi tạo mô hình phân lớp bằng phương pháp cây quyết định (decision tree) với thang đo entropy.

```
model_dtree=DecisionTreeClassifier(criterion='entropy')
model_dtree.fit(X_train,y_train)
DecisionTreeClassifier(criterion='entropy')
```

Hình 51: Code Python tạo ra bộ phân lớp Decision Tree

Sử dụng các phương thức để tính các điểm chỉ số :

```
yhat=model_dtree.predict(X_test)
print(f'Độ chính xác ={accuracy_score(y_test,yhat)*100:.2f}%')
print(f'precision={precision_score(y_test, yhat)*100:.2f}%')
print(f'recall={recall_score(y_test, yhat)*100:.2f}%')
print(f'f1-score={f1_score(y_test, yhat)*100:.2f}%')
```

Hình 52: Code Python để tính các điểm chỉ số (DecisionTree)

Kết quả phân lớp của mô hình phân lớp này là :

```
Độ chính xác =92.12%
precision=92.65%
recall=93.02%
f1-score=92.84%
```

Hình 53: Chỉ số thu được (Decision Tree)

5.2.3. Phân lớp bằng phương pháp Support Vector Machine

Khởi tạo mô hình phân lớp bằng phương pháp Support Vector Machine

```
model_svm=svm.SVC()
model_svm.fit(X_train,y_train)
```

Hình 54: Code Python tạo ra bộ phân lớp SVM

Sử dụng các phương thức để tính các điểm chỉ số :

```
yhat = model_svm.predict(X_test)
print(f'Độ chính xác ={accuracy_score(y_test,yhat)*100:.2f}%')
print(f'precision={precision_score(y_test, yhat)*100:.2f}%')
print(f'recall={recall_score(y_test, yhat)*100:.2f}%')
print(f'f1-score={f1_score(y_test, yhat)*100:.2f}%')
```

Hình 55: Code Python để tính các điểm chỉ số (SVM)

Kết quả phân lớp của mô hình phân lớp này là :

```
Độ chính xác =92.29%  
precision=93.64%  
recall=92.21%  
f1-score=92.92%
```

Hình 56: Chỉ số thu được (SVM)

5.2.4. Phân lớp bằng phương pháp Naive Bayes

Khởi tạo mô hình phân lớp bằng phương pháp Naive bayes với phân phối GaussianNB

```
model_nbayes= GaussianNB()  
model_nbayes.fit(X_train,y_train)
```

Hình 57: Code Python tạo ra bộ phân lớp Naive Bayes

Sử dụng các phương thức để tính các điểm chỉ số :

```
yhat = model_nbayes.predict(X_test)  
print(f'Độ chính xác ={accuracy_score(y_test,yhat)*100:.2f}%')  
print(f'precision={precision_score(y_test, yhat)*100:.2f}%')  
print(f'recall={recall_score(y_test, yhat)*100:.2f}%')  
print(f'f1-score={f1_score(y_test, yhat)*100:.2f}%')
```

Hình 58: Code Python để tính các điểm chỉ số (Naive Bayes)

Kết quả phân lớp của mô hình phân lớp này là :

```
Độ chính xác =81.85%  
precision=82.12%  
recall=85.54%  
f1-score=83.80%
```

Hình 59: Chỉ số thu được (Naive Bayes)

5.3. Đánh giá mô hình

Sau khi có kết quả từ các mô hình phân lớp, ta tiến hành đánh giá sự tối ưu của từng phương pháp thông qua các điểm chỉ số, tùy từng loại mô hình mà ta chọn các điểm chỉ số khác nhau , khi chỉ số accuracy khá tương đồng ta sẽ dùng các chỉ số khác như precision , recall, f1-score và kể cả ma trận nhầm lẫn.

Với phương pháp K-NN classification: Với giá trị $K = 1, 2, 3 \dots n$, ta thấy điểm số tại $K = 3$ là cao nhất với chỉ số Accuracy $\sim 91.29\%$ và chỉ số recall $\sim 86.94\%$.

→ Bộ chỉ số khá tốt

Với phương pháp Decision Tree: Theo thang đo entropy, ta thấy điểm số accuracy $\sim 92.25\%$ và chỉ số recall $\sim 93.08\%$

→ Bộ chỉ số tốt

Với phương pháp Support Vector Machine, ta có điểm số accuracy $\sim 92.29\%$ và recall $\sim 92.21\%$

→ Bộ chỉ số tốt

Với phương pháp Naive Bayes: Theo phân phối GaussianNB, ta có chỉ số accuracy $\sim 81.85\%$

→ Bộ chỉ số không tốt so với các phương pháp còn lại

Vì : Phương pháp K-NN classification mặc dù có điểm số accuracy khá cao với $k=3$ nhưng chỉ số recall lại thấp. Cả 2 phương pháp Decision Tree và Support Vector Machine có điểm số accuracy khá tương đồng với nhau và cao hơn các phương pháp còn lại, đồng thời các điểm chỉ số precision, recall sắp sĩ nhau và trong sai lầm loại 1 của cả hai phương pháp cũng chênh lệch không đáng kể.

Kết luận: Có 2 phương pháp phân lớp tốt là Decision Tree và Support Vector Machine

5.4. Áp dụng mô hình để dự báo sự đánh giá của khách hàng

Sau khi đã chọn được 2 phương pháp phân lớp tốt nhất cho bộ dữ liệu là Decision Tree và Support Vector Machine, nhóm sẽ tiến hành dự báo sự đánh giá của khách hàng với 2 phương pháp đó dựa trên tập X_{test} .

Đầu tiên, tiến hành chạy mô hình dự báo với hàm **predict**.

```
[ ] yhat_svm = model_svm.predict(X_test)
    yhat_dtree = model_dtree.predict(X_test)
```

Hình 60: Code Python chạy mô hình dự báo với 2 phương pháp đã chọn

Vì kết quả dự báo thu được sẽ ở dạng 0 và 1 (0 là không hài lòng và 1 là hài lòng), nên ở đây nhóm sẽ chuyển đổi dạng số sang dạng chữ của mảng yhat để cho quá trình quan sát được rõ ràng và cụ thể hơn.

```
[ ] def convert(a):  
    if(a==1) :  
        return 'Hài lòng'  
    else :  
        return 'Không hài lòng'
```

Hình 61: Chuyển đổi dạng số sang dạng chữ của mảng yhat

Tiến hành chạy vòng lặp for – in cho bảng dự đoán hài lòng / không hài lòng theo mô hình phân lớp Decision Tree

```
for i in range(len(X_test)):  
    print(X_test[i], ' -> ', convert(yhat_dtree[i]))
```

Hình 62: Code Python in ra bảng dự đoán theo mô hình Decision Tree

Thu được kết quả dự báo như sau:

Kết quả truyền trực tuyến bị cắt bớt đến 5000 dòng cuối.	
[0 1 2 1 1 2 3 2 2 2 2 3 3 3 3 3 1 3 3 1 1]	-> Không hài lòng
[0 1 4 1 3 1 3 1 1 1 5 3 4 3 3 3 2 3 4 1 1]	-> Không hài lòng
[0 1 1 1 1 1 3 3 3 3 3 3 3 4 1 3 2 4 3 1 1]	-> Hài lòng
[0 1 4 0 3 1 3 5 3 4 2 5 4 5 5 3 5 3 5 2 1 1]	-> Không hài lòng
[1 1 3 1 1 2 1 1 1 1 5 4 5 5 5 4 5 3 5 3 1 1]	-> Hài lòng
[1 1 1 0 3 2 3 5 3 5 5 3 1 5 3 4 4 4 5 5 1 1]	-> Hài lòng
[1 1 3 1 1 2 3 1 5 1 2 2 2 3 3 3 3 3 1 1 1 1]	-> Không hài lòng
[1 1 2 1 1 2 1 5 1 1 3 5 1 4 4 4 4 4 4 5 1 1]	-> Hài lòng
[1 0 2 1 1 2 3 3 3 3 2 3 2 2 5 3 5 3 5 2 1 1]	-> Không hài lòng
[0 1 1 0 3 2 4 5 4 5 2 4 2 2 3 1 4 2 3 2 1 1]	-> Không hài lòng
[0 1 4 1 1 1 2 2 2 2 5 5 5 4 4 4 4 4 4 4 1 1]	-> Hài lòng
[0 1 4 1 1 2 2 1 3 3 2 3 3 2 2 2 2 4 2 4 1 1]	-> Không hài lòng
[0 1 2 1 1 3 3 3 3 3 4 4 4 4 4 4 5 4 5 4 1 1]	-> Hài lòng
[0 1 1 0 3 2 4 4 4 4 5 2 1 5 5 5 5 1 5 5 1 1]	-> Hài lòng
[0 1 4 1 1 1 4 4 5 4 3 4 5 4 4 4 4 4 4 5 1 1]	-> Hài lòng
[1 1 2 0 3 2 3 4 3 3 2 3 2 2 3 2 5 5 4 2 1 1]	-> Không hài lòng
[0 1 1 0 3 2 4 4 4 1 2 4 2 2 3 3 3 5 3 2 1 1]	-> Không hài lòng
[0 0 1 1 1 2 3 3 3 4 5 3 5 5 4 2 4 3 4 5 1 1]	-> Hài lòng
[1 1 3 1 1 2 4 4 4 4 5 4 5 2 2 2 2 4 2 3 1 1]	-> Hài lòng
[0 1 2 1 3 2 3 3 3 1 5 3 5 5 4 3 5 3 5 5 1 1]	-> Hài lòng
[1 0 1 1 3 2 1 1 1 4 1 1 1 1 2 4 3 2 4 1 1 1]	-> Không hài lòng
[0 0 1 1 3 2 3 3 3 2 2 3 2 2 1 5 1 1 5 2 1 1]	-> Hài lòng
[1 1 2 0 1 1 4 4 4 4 5 5 4 4 4 4 4 3 4 3 1 1]	-> Không hài lòng
[0 1 4 1 1 2 1 5 5 5 2 4 3 1 1 1 1 3 1 1 1 1]	-> Không hài lòng
[1 1 4 1 1 2 1 1 2 1 2 5 3 4 4 4 4 2 4 1 1 1]	-> Hài lòng
[0 1 4 0 3 1 4 3 4 1 3 5 1 1 1 4 1 5 1 2 1 1]	-> Hài lòng
[1 0 1 1 3 2 5 3 5 2 2 5 2 2 5 5 3 5 2 2 1 1]	-> Hài lòng
[0 1 4 0 3 1 5 5 5 5 4 5 4 4 4 4 5 3 4 3 1 1]	-> Hài lòng
[0 1 4 1 1 1 3 3 3 3 3 5 5 4 4 4 4 5 4 3 1 1]	-> Hài lòng
[1 1 2 0 3 2 3 5 2 3 2 2 2 2 2 1 1 3 5 2 1 1]	-> Không hài lòng
[1 1 3 1 3 2 2 2 2 2 2 2 2 2 2 2 4 4 4 2 1 1]	-> Không hài lòng
[0 1 3 1 2 1 4 1 1 1 2 4 3 4 4 4 4 2 4 4 1 1]	-> Hài lòng
[0 1 4 1 1 1 5 5 5 5 5 5 4 5 5 5 5 5 5 3 1 1]	-> Hài lòng

Hình 63: Kết quả dự báo với Decision Tree

Chạy tiếp vòng lặp for – in cho bảng dự đoán hài lòng / không hài lòng theo mô hình phân lớp SVM.

```
for i in range(len(X_test)):
    print(X_test[i], ' -> ', convert(yhat_svm[i]))
```

Hình 64: Code Python in ra bảng dự đoán theo mô hình SVM

Thu được kết quả dự báo như sau:

Kết quả truyền trực tuyến bị cắt bớt đến 5000 dòng cuối.																									
[0	1	2	1	1	2	3	2	2	2	2	3	3	3	3	3	1	3	3	1	1]	->	Không	hài	lòng	
[0	1	4	1	3	1	3	1	1	1	5	3	4	3	3	3	2	3	4	1	1]	->	Không	hài	lòng	
[0	1	1	1	1	1	3	3	3	3	3	3	3	4	1	3	2	4	3	1	1]	->	Không	hài	lòng	
[0	1	4	0	3	1	3	5	3	4	2	5	4	5	5	3	5	3	5	2	1	1]	->	Không	hài	lòng
[1	1	3	1	1	2	1	1	1	1	5	4	5	5	5	4	5	3	5	3	1	1]	->	Hài	lòng	
[1	1	1	0	3	2	3	5	3	5	5	3	1	5	3	4	4	4	5	5	1	1]	->	Không	hài	lòng
[1	1	3	1	1	2	3	1	5	1	2	2	2	3	3	3	3	3	1	1	1]	->	Không	hài	lòng	
[1	1	2	1	1	2	1	5	1	1	3	5	1	4	4	4	4	4	5	1	1]	->	Hài	lòng		
[1	0	2	1	1	2	3	3	3	3	2	3	2	2	5	3	5	3	5	2	1	1]	->	Không	hài	lòng
[0	1	1	0	3	2	4	5	4	5	2	4	2	2	3	1	4	2	3	2	1	1]	->	Không	hài	lòng
[0	1	4	1	1	1	2	2	2	2	5	5	5	4	4	4	4	4	4	1	1]	->	Hài	lòng		
[0	1	4	1	1	2	2	1	3	3	2	3	3	2	2	2	2	4	2	4	1	1]	->	Không	hài	lòng
[0	1	2	1	1	3	3	3	3	3	4	4	4	4	4	4	5	4	5	4	1	1]	->	Hài	lòng	
[0	1	1	0	3	2	4	4	4	4	5	2	1	5	5	5	5	1	5	5	1	1]	->	Hài	lòng	
[0	1	4	1	1	1	4	4	5	4	3	4	5	4	4	4	4	4	5	1	1]	->	Hài	lòng		
[1	1	2	0	3	2	3	4	3	3	2	3	2	2	3	2	5	5	4	2	1	1]	->	Không	hài	lòng
[0	1	1	0	3	2	4	4	4	1	2	4	2	2	3	3	3	5	3	2	1	1]	->	Hài	lòng	
[0	0	1	1	1	2	3	3	3	4	5	3	5	5	4	2	4	3	4	5	1	1]	->	Không	hài	lòng
[1	1	3	1	1	2	4	4	4	4	5	4	5	2	2	2	2	4	2	3	1	1]	->	Hài	lòng	
[0	1	2	1	3	2	3	3	3	1	5	3	5	5	4	3	5	3	5	5	1	1]	->	Hài	lòng	
[1	0	1	1	3	2	1	1	1	4	1	1	1	1	1	2	4	3	2	4	1	1]	->	Không	hài	lòng
[0	0	1	1	3	2	3	3	3	2	2	3	2	2	1	5	1	1	5	2	1	1]	->	Không	hài	lòng
[1	1	2	0	1	1	4	4	4	4	5	5	4	4	4	4	4	3	4	3	1	1]	->	Hài	lòng	
[0	1	4	1	1	2	1	5	5	5	2	4	3	1	1	1	1	3	1	1	1	1]	->	Không	hài	lòng
[1	1	4	1	1	2	1	1	2	1	2	5	3	4	4	4	4	2	4	1	1	1]	->	Hài	lòng	
[0	1	4	0	3	1	4	3	4	1	3	5	1	1	1	4	1	5	1	2	1	1]	->	Hài	lòng	
[1	0	1	1	3	2	5	3	5	2	2	5	2	2	5	5	3	5	2	2	1	1]	->	Hài	lòng	
[0	1	4	0	3	1	5	5	5	5	4	5	4	4	4	4	5	3	4	3	1	1]	->	Hài	lòng	
[0	1	4	1	1	1	3	3	3	3	3	5	5	4	4	4	4	5	4	3	1	1]	->	Hài	lòng	
[1	1	2	0	3	2	3	5	2	3	2	2	2	2	2	1	1	3	5	2	1	1]	->	Không	hài	lòng
[1	1	3	1	3	2	2	2	2	2	2	2	2	2	2	2	4	4	4	2	1	1]	->	Không	hài	lòng
[0	1	3	1	2	1	4	1	1	1	2	4	3	4	4	4	4	2	4	4	1	1]	->	Không	hài	lòng
[0	1	4	1	1	1	5	5	5	5	5	4	5	5	5	5	5	5	5	3	1	1]	->	Hài	lòng	
[1	1	2	1	1	2	5	5	5	5	4	4	4	4	5	5	5	5	5	4	1	1]	->	Hài	lòng	

Hình 65: Kết quả dự báo với SVM

CHƯƠNG 6. PHÂN CỤM

Trong phân phân cụm dữ liệu, nhóm chọn biến satisfaction là biến phân loại. Nhiệm vụ của nhóm trong phần này là bỏ qua biến phân loại satisfaction, tiến hành phân cụm bộ dữ liệu, và so sánh kết quả phân cụm đó so sánh với giá trị của biến phân loại gốc để tính tỷ lệ phân cụm đúng của phương pháp phân cụm mà nhóm chọn.

2 phương pháp phân cụm phổ biến nhất đó là k-Means và Hierarchical Agglomerative Clustering - HAC. Tuy nhiên, vì kích thước bộ dữ liệu lớn (130000 quan

sát) nên không thể sử dụng phương pháp HAC để phân cụm. Do vậy, trong phần này, nhóm sẽ dùng 1 phương pháp duy nhất là k-Means để phân cụm.

6.1. Phương pháp K-Means

Nhóm lập 1 DataFrame `data_cluster1` là bản sao của bộ dữ liệu gốc, bỏ qua biến phân loại `satisfaction` và biến `Arrival Delay in Minutes` do biến này có tương quan cao với biến `Departure Delay in Minutes`.

```
data_cluster1 = data.drop(columns = ['satisfaction', 'Arrival Delay in Minutes'])
data_cluster1
```

Hình 66: Code Python lập 1 DataFrame là bản sao của bộ dữ liệu gốc

	Gender	Customer Type	Age Group	Type of Travel
0	0	1	4	0
1	1	1	3	0
2	0	1	1	0
3	0	1	4	0
4	0	1	4	0
...
129482	0	0	2	0
129483	1	0	4	0
129484	1	0	4	0
129485	1	0	4	0
129486	0	0	2	0
129487 rows × 21 columns				

Hình 67: Bộ dữ liệu mới

Nhóm chọn số cụm $k = 2$ vì theo như biến satisfaction, biến có 2 giá trị là satisfied và dissatisfied, nên nhóm sẽ chọn k giống với số giá trị duy nhất của biến phân loại gốc để phân cụm. Sau đó, nhóm chọn phương pháp K Means và tiến hành phân cụm cho data_cluster1.

```
k = 2
model_kMeans = KMeans(n_clusters = k)
model_kMeans.fit(data_cluster2)
```

Hình 68: Code Python tạo bộ phân cụm kMeans

Sau khi tiến hành phân cụm, nhóm sẽ biểu diễn 1 số kết quả. Đầu tiên, nhóm tạo 1 bộ dữ liệu mới bằng cách ghép bộ dữ liệu data_cluster1 và 1 cột mới là cột kết quả phân cụm 'cluster'. Tiếp theo, nhóm sẽ thể hiện mỗi cụm có bao nhiêu phần tử qua lệnh `.cluster.value_counts()`

```
## Các clusters
labels = model_kMeans.labels_
clustering = pd.concat([data_cluster2, pd.Series(labels, name = 'cluster')], axis = 1)
print('Số phần tử của mỗi cluster:')
print(clustering.cluster.value_counts(), '\n')
```

Hình 69: Code Python in ra số phần tử của mỗi cluster

```
Số phần tử của mỗi cluster:
0    77016
1    52471
Name: cluster, dtype: int64
```

Hình 70: Kết quả số phần tử của mỗi cluster

Trong thuật toán k-Means mỗi cụm dữ liệu được đặc trưng bởi một tâm (centroid). Tâm là điểm đại diện nhất cho một cụm và có giá trị bằng trung bình của toàn bộ các quan sát nằm trong cụm. Chúng ta sẽ dựa vào khoảng cách từ mỗi quan sát tới các tâm để xác định nhãn cho chúng trùng thuộc về tâm gần nhất.

```
## Các trọng tâm = các vectors trong không gian 4 chiều
print(f'Tọa độ của {k} trọng tâm:')
centroids = model_kMeans.cluster_centers_
print(centroids)
```

Hình 71: Code python in ra tọa độ của 2 trọng tâm

```
Tọa độ của 2 trọng tâm:
[[0.45359124 0.87205602 2.56291976 0.69470895 1.8213149 1.70776446
 3.23233609 3.19446862 3.10786059 3.01822575 3.84795853 3.94517985
 4.21820237 4.2987568 3.91880903 3.87734317 4.06544642 3.64598137
 4.07577391 4.06041908 1.00405305]
 [0.54976385 0.735945 2.31067647 0.68498134 2.18819989 1.76127447
 2.53448998 3.07052259 2.81530433 2.94966481 2.37872324 2.73023158
 2.4963815 2.26138874 2.80033519 2.937819 3.15304335 2.89327341
 3.16389883 2.31557096 1.00578959]]
```

Hình 72: Kết quả tọa độ của 2 trọng tâm

Tiếp theo, nhóm tiến hành so sánh kết quả phân cụm bằng phương pháp k-Means với giá trị của biến phân loại gốc 'satisfaction' để tính tỷ lệ phân cụm đúng. Tuy nhiên, ở đây nhóm không biết được 2 giá trị phân cụm [0;1] là biểu hiện cho giá trị nào (0 là satisfied hay dissatisfied?), vì thế nên nhóm sẽ thử 2 trường hợp: giá trị cột cluster giống giá trị cột satisfaction, và ngược lại. Sau khi thử 2 trường hợp thì nhóm rút ra được: trong cột cluster, 0 là dissatisfied, 1 là satisfied. Tỷ lệ phân cụm chính xác giữa kết quả sau khi phân cụm và biến target satisfaction ban đầu là 76,4%, một tỷ lệ có thể chấp nhận được.

```
1 count = 0
2 for i in range(0, len(data)-1):
3     if data_cluster1.satisfaction[i] != clustering.cluster[i]:
4         count += 1
5 print(f'Tỷ lệ dự đoán đúng khi đối chiếu với bộ dữ liệu gốc: {round(count/len(data_cluster1),3)*100}%')
```

Hình 73: Code Python đối chiếu kết quả phân cụm với bộ dữ liệu gốc

Silhouette đo lường khoảng cách của một điểm dữ liệu trong cụm đến Centroid, điểm trung tâm của cụm, và khoảng cách của chính điểm đó đến điểm trung tâm của cụm gần nhất (hoặc đến các điểm trung tâm của các cụm còn lại, và chọn ra khoảng cách ngắn nhất). Theo kinh nghiệm của các tác giả trong tài liệu “Data mining and Predictive analytics” của nhà xuất bản Wiley:

- Điểm trung bình Silhouette từ 0.5 trở lên, bằng chứng cho thấy có thể cluster này sát với thực tế
- Điểm trung bình Silhouette từ 0.25 đến 0.5, thì cần thêm kiến thức chuyên môn, kinh nghiệm để đánh giá thêm khả năng cluster có trong thực tế
- Điểm trung bình dưới 0.25, thì không nên tin tưởng cluster, và cần đi tìm nhiều bằng chứng khác.

Nhóm tiến hành tính silhouette score với số cụm hiện tại bằng 2 và bộ dữ liệu gồm 21 biến độc lập, để xem xét liệu việc phân cụm như vậy có hiệu quả không. Kết quả nhóm nhận được là 0.172, 1 điểm số quá thấp, khó chấp nhận được kết quả phân cụm này.

```

1 from sklearn.metrics import silhouette_score
2
3 # Fit the KMeans model
4 #
5 model_kMeans.fit_predict(data_cluster1)
6 #
7 # Calculate Silhouette Score
8 #
9 score = silhouette_score(data_cluster1, model_kMeans.labels_, metric='euclidean')
10 #
11 # Print the score
12 #
13 print('Silhouette Score: %.3f' % score)

```

➡ Silhouette Score: 0.172

Hình 74: Silhouette score khi phân thành 2 cụm cho bộ dữ liệu 21 biến features

6.2. Phương án 2

Vì việc phân cụm trên không hiệu quả nên nhóm sẽ tiến hành phân cụm với số lượng biến ít hơn, nhưng không theo mục tiêu đề ra ban đầu là phân cụm để đối chiếu với biến phân loại satisfaction, mà nhóm lập các cụm mới và nêu đặc điểm các cụm, từ đó có thể đưa ra các gợi ý giúp nâng cao trải nghiệm bay của khách hàng.

Trong lần phân cụm này, nhóm chọn 2 biến 'Class' và 'Gate location' vì trong biểu đồ Heatmap, hệ số tương quan giữa 2 biến rất thấp, chỉ có 0.00. Nhóm tạo 1 bảng dữ liệu mới là data_cluster2 chứa 2 biến trên.

```
data_cluster2 = data[['Class', 'Gate location']]
```

Hình 75: Tạo dataframe với 2 biến 'Class' và 'Gate Location'

	Class	Gate location
0	3	2
1	1	3
2	3	3
3	3	3
4	3	3
...
129482	3	3
129483	1	4
129484	3	3
129485	3	2
129486	3	3

129487 rows × 2 columns

Hình 76: Dataframe thu được

Nhóm tiến hành tính Silhouette score với số cụm k chạy từ 2 đến 5, và biểu diễn kết quả dưới dạng đồ thị. Số cụm nào có Silhouette score cao nhất sẽ được chọn.

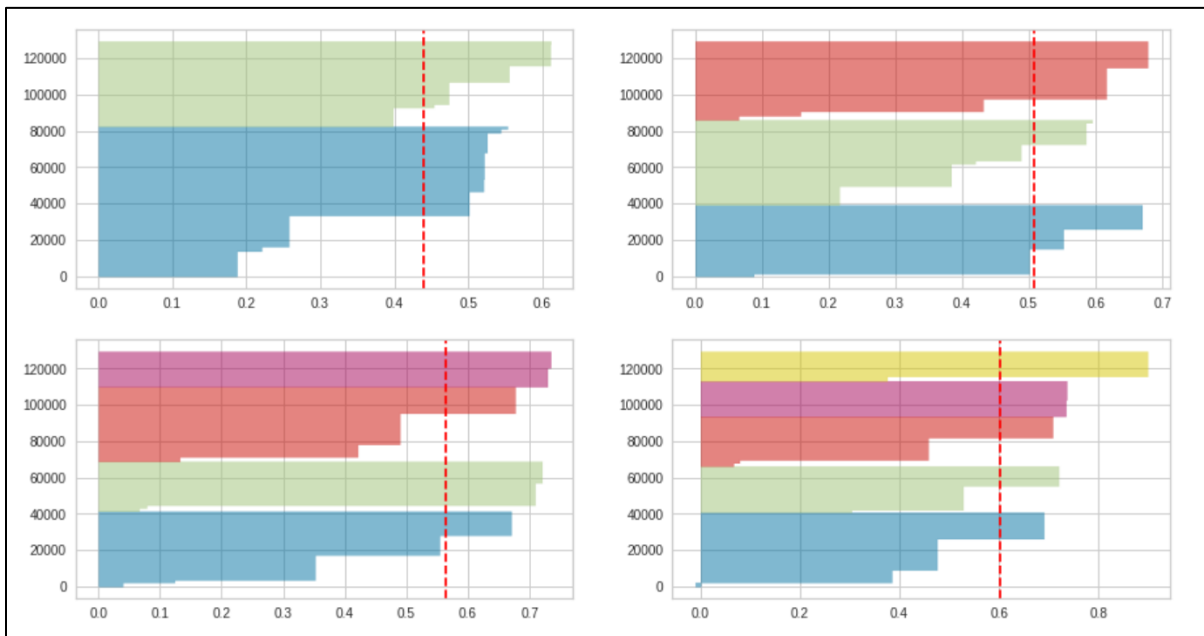
```

from yellowbrick.cluster import SilhouetteVisualizer

fig, ax = plt.subplots(2, 2, figsize=(15,8))
for i in [2, 3, 4, 5]:
    '''
    Create KMeans instance for different number of clusters
    '''
    model_kMeans = KMeans(n_clusters=i, init='k-means++', n_init=10, max_iter=100, random_state=42)
    q, mod = divmod(i, 2)
    '''
    Create SilhouetteVisualizer instance with KMeans instance
    Fit the visualizer
    '''
    visualizer = SilhouetteVisualizer(model_kMeans, colors='yellowbrick', ax=ax[q-1][mod])
    visualizer.fit(data_cluster2)

```

Hình 77: Code Python biểu diễn kết quả Sihouette Score dưới dạng đồ thị



Hình 78: Từ trái qua phải, từ trên xuống dưới lần lượt là Silhouette score tương ứng với số cụm $k = 2$, $k = 3$, $k = 4$, $k = 5$

Có thể thấy, tương ứng với số cụm $k = 5$, Silhouette score có giá trị cao nhất, đồng thời cũng vượt qua mức 0.5, cụ thể là 0.602 - có thể cluster này sát với thực tế. Vì thế, nhóm quyết định tiến hành phân cụm với số cụm là 5.

```

1 from sklearn.metrics import silhouette_score
2
3 # Fit the KMeans model
4 #
5 model_kMeans.fit_predict(data_cluster2)
6 #
7 # Calculate Silhoutte Score
8 #
9 score = silhouette_score(data_cluster2,model_kMeans.labels_, metric='euclidean')
10 #
11 # Print the score
12 #
13 print('Silhouetter Score: %.3f' % score)

```

Silhouetter Score: 0.602

Hình 79: Với $k = 5$, Silhouette score = 0.602

Nhóm tiến hành phân cụm bộ dữ liệu data_cluster2 bằng phương pháp kMeans với số cụm $k=5$. Các bước làm tương tự như khi phân cụm cho data_cluster1.

```

k = 5
model_kMeans = KMeans(n_clusters = k)
model_kMeans.fit(data_cluster2)

## Các clusters
labels      = model_kMeans.labels_
clustering1 = pd.concat([data_cluster2, pd.Series(labels, name = 'cluster')], axis = 1)
print('Số phần tử của mỗi cluster:')
print(clustering1.cluster.value_counts(), '\n')

## Các trọng tâm = các vectors trong không gian 4 chiều
print(f'Tọa độ của {k} trọng tâm:')
centroids  = model_kMeans.cluster_centers_
print(centroids)

```

Hình 80: Phân cụm bộ dữ liệu data_cluster2


```
Số phần tử của mỗi cluster:  
1      40847  
0      27468  
2      25439  
3      19470  
4      16263  
Name: cluster, dtype: int64
```

Hình 81: Số phần tử của 5 cụm

```
Tọa độ của 5 trọng tâm:  
[[1.12265181 1.51084899]  
 [2.94614048 3.74727642]  
 [1.05251779 4.48083651]  
 [3.          1.53461736]  
 [1.15218594 3.          ]]
```

Hình 82: Tọa độ của 5 trọng tâm của mỗi cụm

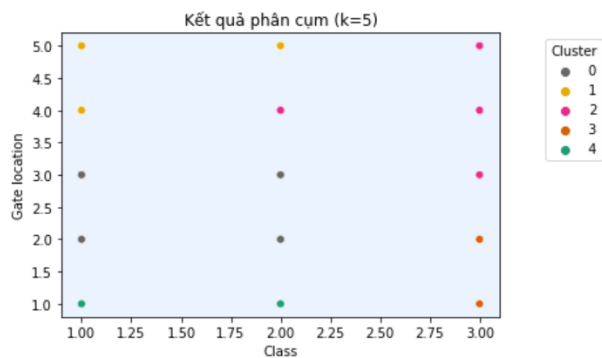
6.3. Biểu diễn kết quả phân cụm

Nhóm sẽ biểu diễn kết quả phân cụm bằng biểu đồ Scatter, với trục tung là số điểm đánh giá của tiêu chí ‘Gate location’, trục hoành là hạng vé của hành khách ‘Class’.

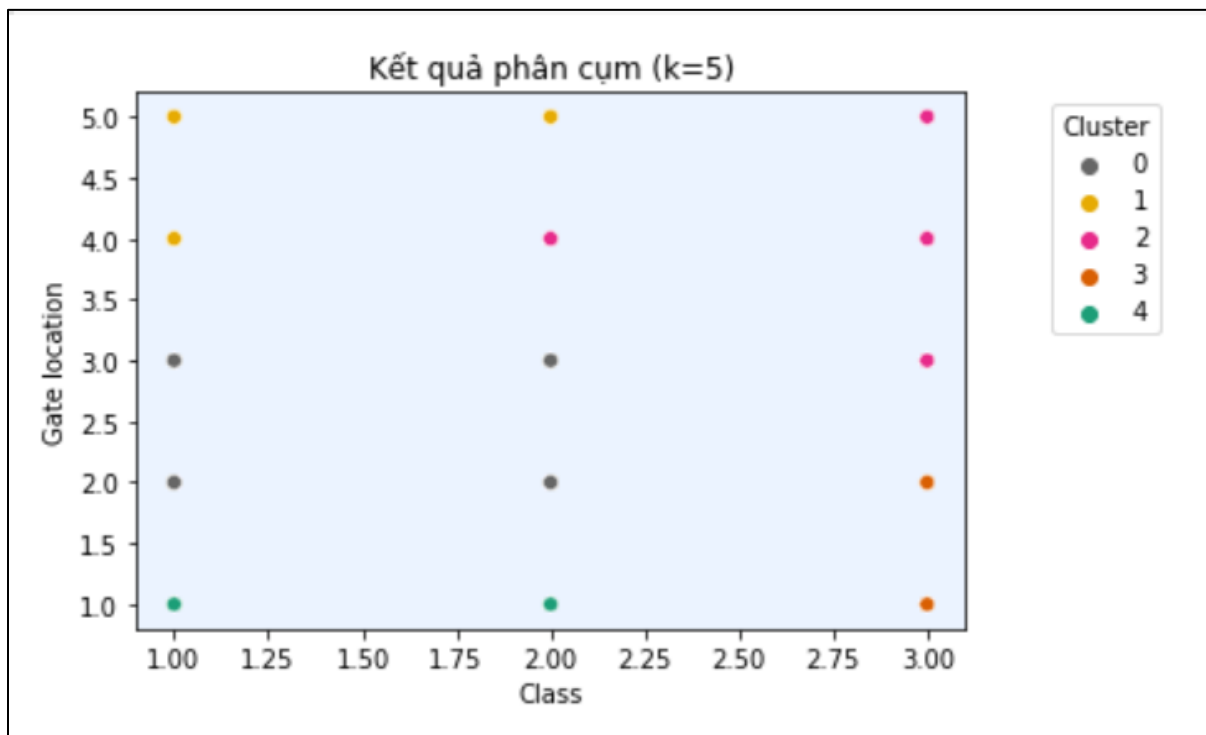
```

1 ax = plt.axes()
2 ax.set_facecolor('#EBF3FF')
3
4
5 scatter = sns.scatterplot(data = clustering1, x = 'Class', y = 'Gate location', hue = 'cluster', palette='Dark2_r')
6 plt.grid(False)
7 plt.legend(title="Cluster",
8           loc='upper right', bbox_to_anchor=(1.25, 1))
9 plt.title('Kết quả phân cụm (k=5)')
10 plt.show()

```



Hình 83: Code Python biểu diễn kết quả phân cụm



Hình 84: Kết quả phân cụm

Vì biến của nhóm là các biến rời rạc, ít giá trị duy nhất nên khi vẽ biểu đồ Scatter các phần tử trong cùng 1 cụm không tập trung lại gần nhau, nhưng trong biểu đồ này, các phần

tử trong cùng 1 cụm được tô màu giống nhau, ở vị trí rất gần nhau, không lẫn vào các cụm khác nên vẫn có thể phân biệt được các cụm trên biểu đồ.

```
1 from collections import Counter
2
3 # count the occurrences of each point
4 c = Counter(zip(clustering1['Class'], clustering1['Gate location'], clustering1['cluster']))
5 display(c)
```

```
Counter({(3, 2, 3): 10409,
          (1, 3, 0): 13788,
          (3, 3, 2): 17190,
          (3, 4, 2): 14590,
          (3, 1, 3): 9061,
          (1, 2, 0): 12284,
          (2, 2, 0): 1748,
          (2, 3, 0): 2475,
          (2, 4, 2): 2200,
          (1, 4, 1): 13207,
          (2, 1, 4): 1621,
          (1, 5, 1): 10896,
          (3, 5, 2): 6867,
          (2, 5, 1): 1336,
          (1, 1, 4): 11815})
```

Hình 85: Tần số của các điểm trên đồ thị

Đặc điểm của các cụm như sau:

Cụm	Đặc điểm
0	<ul style="list-style-type: none"> Gồm 2 nhóm hành khách đi hạng vé Business và Eco Plus, trong đó hành khách đi vé Business chiếm đa số (> 25000) so với Eco Plus (~ 4000) Mức đánh giá cho tiêu chí Gate location nằm trong khoảng 2-3 điểm Đây là nhóm khách hàng ở phân khúc vé cao cấp của hãng bay, và với điểm số đánh giá tiêu chí địa điểm cổng lên máy bay khá thấp, có thể thấy họ khá không hài lòng về vị trí cổng, nguyên nhân có thể vì di chuyển giữa các địa điểm tới cổng xa, hoặc cộng khó tìm, điều này cần dữ liệu thêm để hiểu rõ nguyên nhân hơn.
1	<ul style="list-style-type: none"> Gồm 2 nhóm hành khách đi hạng vé Business và Eco Plus, trong đó hành khách đi vé Business chiếm đa số (> 24000) so với Eco Plus (1363) Mức đánh giá cho tiêu chí Gate location nằm trong khoảng 4-5 điểm

	<ul style="list-style-type: none"> – Đây là nhóm khách hàng ở phân khúc vé cao cấp của hãng bay, và với điểm số đánh giá tiêu chí địa điểm cổng lên máy bay khá cao, thể hiện sự hài lòng của nhóm khách hàng này với vị trí thuận lợi của cổng lên máy bay.
2	<ul style="list-style-type: none"> – Gồm 2 nhóm hành khách đi hạng vé Eco Plus và Eco, trong đó hành khách đi vé Eco chiếm đa số (> 30000) so với Eco Plus (2200) – Mức đánh giá cho tiêu chí Gate location nằm trong khoảng 3-4 điểm – Đây là nhóm khách hàng ở phân khúc vé bình dân của hãng bay, và với điểm số đánh giá tiêu chí địa điểm cổng lên máy bay ở mức khá, nhóm khách hàng này không có yêu cầu cao về vị trí cổng.
3	<ul style="list-style-type: none"> – Gồm hành khách đi hạng vé Eco – Mức đánh giá cho tiêu chí Gate location nằm trong khoảng 1-2 điểm – Đây là nhóm khách hàng ở phân khúc vé bình dân của hãng bay, và với điểm số đánh giá tiêu chí địa điểm cổng lên máy bay ở mức thấp, có thể thấy họ không hài lòng về vị trí cổng
4	<ul style="list-style-type: none"> – Gồm 2 nhóm hành khách đi hạng vé Business và Eco Plus, trong đó hành khách đi vé Business chiếm đa số (> 11000) so với Eco Plus (1621) – Mức đánh giá cho tiêu chí Gate location là 1 điểm – Đây là nhóm khách hàng ở phân khúc vé cao cấp của hãng bay, và với điểm số đánh giá tiêu chí địa điểm cổng lên máy bay ở mức thấp nhất, có thể thấy họ đang không hài lòng về vị trí cổng, nguyên nhân có thể vì di chuyển giữa các địa điểm tới cổng xa, hoặc cộng khó tìm, điều này cần dữ liệu thêm để hiểu rõ nguyên nhân hơn.

Qua phân phân tích đặc điểm các cụm, có thể thấy nhóm ‘4’ và ‘0’ là nhóm đáng quan tâm, cần được chú ý. Việc làm hài lòng nhóm khách hàng ở phân khúc vé cao cấp nên được ưu tiên, vì nó thể hiện hình ảnh chuyên nghiệp của hãng bay, và chính những vị khách

ở phân khúc này, nếu khiến họ hài lòng, sẽ gia tăng sự trung thành nhãn hàng của họ và có thể tăng tỷ lệ chuyển đổi sang khách hàng trung thành của hãng. Yếu tố vị trí cổng lên máy bay có thể được cải thiện bằng cách thêm 1 số tiện ích cho hạng vé cao cấp, như có xe trung chuyển, sử dụng phòng chờ ngay cổng vào,..., khiến khách hàng có trải nghiệm bay tốt hơn.

TÀI LIỆU THAM KHẢO

1. Các file Colab của thầy Tế đăng tải trên trang LMS môn LTPTDL, DM, DV
2. *Airlines Customer satisfaction*. (n.d.). Kaggle. Retrieved November 30, 2022, from <https://www.kaggle.com/datasets/sjleshrac/airlines-customer-satisfaction>
3. *Flight length*. (n.d.). Wikipedia. Retrieved November 30, 2022, from https://en.wikipedia.org/wiki/Flight_length
4. *Parallel categories diagram in Python*. (n.d.). Plotly. Retrieved November 30, 2022, from <https://plotly.com/python/parallel-categories-diagram/>
5. Sunburst charts in Python. (n.d.). Plotly. Retrieved November 30, 2022, from <https://plotly.com/python/sunburst-charts/>
6. *Train Test Validation Split: How To & Best Practices [2022]*. (n.d.). v7labs. Retrieved November 28, 2022, from <https://www.v7labs.com/blog/train-validation-test-set>
7. <https://www.v7labs.com/blog/train-validation-test-set>
8. Training and Test Sets: Splitting Data. (n.d.). developers.google. Retrieved November 29, 2022, from <https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data>
9. <https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data>
10. *Matplotlib: Change Scatter Plot Marker Size*. (2021, May 14). Stack Abuse. Retrieved December 2, 2022, from <https://stackabuse.com/matplotlib-change-scatter-plot-marker-size/>
11. *How to have scatter points become larger for higher density using matplotlib?* (2017, October 12). Stack Overflow. Retrieved December 2, 2022, from <https://stackoverflow.com/questions/46700733/how-to-have-scatter-points-become-larger-for-higher-density-using-matplotlib>
12. *Silhouette score là gì*. (n.d.). Hàng Hiệu Giá Tốt. Retrieved December 2, 2022, from <https://hanghieugiatot.com/silhouette-score-la-gi>

BẢNG PHÂN CÔNG

STT	CÔNG VIỆC	NGƯỜI LÀM	CHI TIẾT CÔNG VIỆC	NHẬN XÉT
1	Tiền xử lý dữ liệu	Thanh, Tuyết Nhung		Hoàn thành đúng hạn
2	Biểu đồ trực quan phân tích dữ liệu	Cả nhóm	Mỗi người 1 biểu đồ biểu diễn trực quan 1 khía cạnh về bộ dữ liệu + nhận xét, giải thích về biểu đồ	Hoàn thành đúng hạn
3	Phân cụm	Thanh, Thy, Tuyết Nhung	Cho biến satisfaction là biến phân loại, lờ biến đó đi và tiến hành phân cụm (tùy team chọn 1 phương pháp hoặc thử cả 2), Đối chiếu lại với biến phân loại gốc để đo độ chính xác	Hoàn thành đúng hạn
4	Phân lớp	Thư, Nhơn, Cẩm Nhung	Chia tập dữ liệu thành training set và testing set Testing set thì bỏ biến satisfaction Áp dụng 1 / nhiều phương pháp phân lớp Tiến hành dự báo cho testing set	Hoàn thành đúng hạn

5	Bản báo cáo bản thô	Cả nhóm	<p>Bố cục:</p> <p>Giới thiệu bài toán, nội dung bộ dữ liệu, overall về bộ dữ liệu, đầu công việc của nhóm</p> <p>Tiền xử lý dữ liệu</p> <p>Phân tích dữ liệu</p> <p>Phân cụm</p> <p>Phân lớp</p> <p>Bảng phân công</p> <p>Tài liệu tham khảo</p>	Hoàn thành đúng hạn
6	Bản Colab hoàn chỉnh	Thanh	Sắp xếp lại phần trình bày	Hoàn thành đúng hạn
7	Bản báo cáo hoàn chỉnh	Thư + Cẩm Nhung		Hoàn thành đúng hạn
8	Nộp bài	Thanh		Hoàn thành đúng hạn