



TRƯỜNG ĐẠI HỌC KINH TẾ - ĐẠI HỌC ĐÀ NẴNG  
KHOA THƯƠNG MẠI ĐIỆN TỬ  
NGÀNH KHOA HỌC DỮ LIỆU VÀ PHÂN TÍCH KINH DOANH

## BÁO CÁO DỰ ÁN

# ĐỀ TÀI: ỨNG DỤNG CÁC THUẬT TOÁN HỒI QUY ĐỂ DỰ ĐOÁN THỜI GIAN GIAO HÀNG VÀ THUẬT TOÁN PHÂN CỤM K-MEANS ĐỂ PHÂN KHÚC TÀI XẾ TRONG DỊCH VỤ GIAO ĐỒ ĂN

GV hướng dẫn:

Lớp học phần:

Nhóm:

Thành viên:

1. Trần Hoài Huệ
2. Phan Thị Ngọc Minh
3. Lê Thúy Quỳnh

TS. Lê Diên Tuấn

ELC3022\_48K29.1

04

# TABLE OF CONTENTS

**1. Giới thiệu dự án**

**2. Tiền xử lý dữ liệu**

**3. Trực quan hóa dữ liệu**

**4. Mô hình dự đoán thời gian giao hàng**

**5. Mô hình phân cụm tài xế**

**6. Triển khai ứng dụng và kết luận**

# **1. MỤC TIÊU DỰ ÁN**

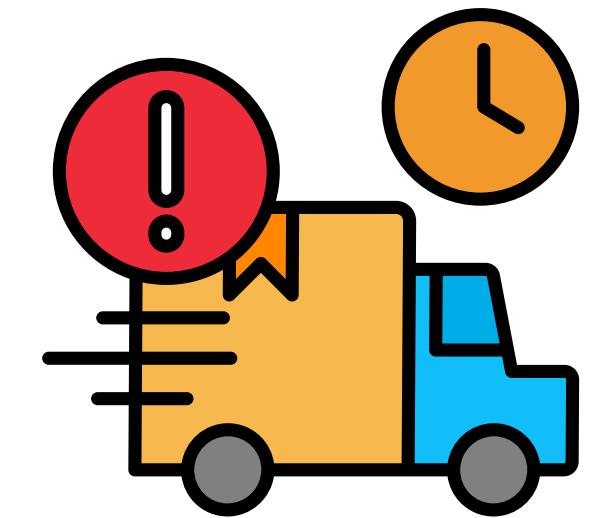
# 1. Mục tiêu dự án

## Tầm quan trọng:

- Sự phát triển vượt bậc của dịch vụ giao đồ ăn trực tuyến
- Giao hàng đúng giờ là thước đo uy tín và năng lực cạnh tranh

## Thách thức trong thực tế

- Kẹt xe, thời tiết, lỗi phương tiện, lịch làm việc tài xế đều gây biến động lớn
- Chỉ vài phút chậm trễ cũng làm giảm mức độ hài lòng của khách hàng



# 1. Mục tiêu dự án

## Giải pháp học máy

- Xây dựng **mô hình hồi quy (Linear Regression, Decision Tree, Random Forest, XGBoost)** để dự đoán thời gian giao hàng
- Sử dụng **mô hình phân cụm K-Means** để phân nhóm tài xế theo hành vi và hiệu suất làm việc
  - Nâng cao trải nghiệm khách hàng nhờ dự báo thời gian giao hàng đáng tin cậy
  - Tối ưu hóa nguồn lực vận hành thông qua việc phân loại và hỗ trợ tài xế theo nhóm
  - Tăng tính linh hoạt và khả năng ứng phó với các yếu tố bất định như thời tiết, tắc đường hay biến động lịch trình



## 1. Mục tiêu dự án

- Xây dựng các **mô hình hồi quy (Linear Regression, Decision Tree Regressor, Random Forest Regressor, XGBoost)** để dự đoán thời gian giao hàng dựa trên các yếu tố như khoảng cách, thời điểm trong ngày, mật độ giao thông, thời tiết,...  
=> Doanh nghiệp có thể chủ động trong việc cam kết thời gian, giảm thiểu rủi ro trễ hẹn và tăng mức độ hài lòng của khách hàng.
- Xây dựng **mô hình phân cụm K-Means** để phân nhóm tài xế có đặc điểm hành vi tương đồng với nhau. Việc phân khúc này giúp doanh nghiệp hiểu rõ hơn về từng nhóm tài xế  
=> Doanh nghiệp có thể đưa ra các chiến lược quản lý phù hợp như đào tạo, phân công đơn hàng thông minh và cải thiện hiệu suất tổng thể.



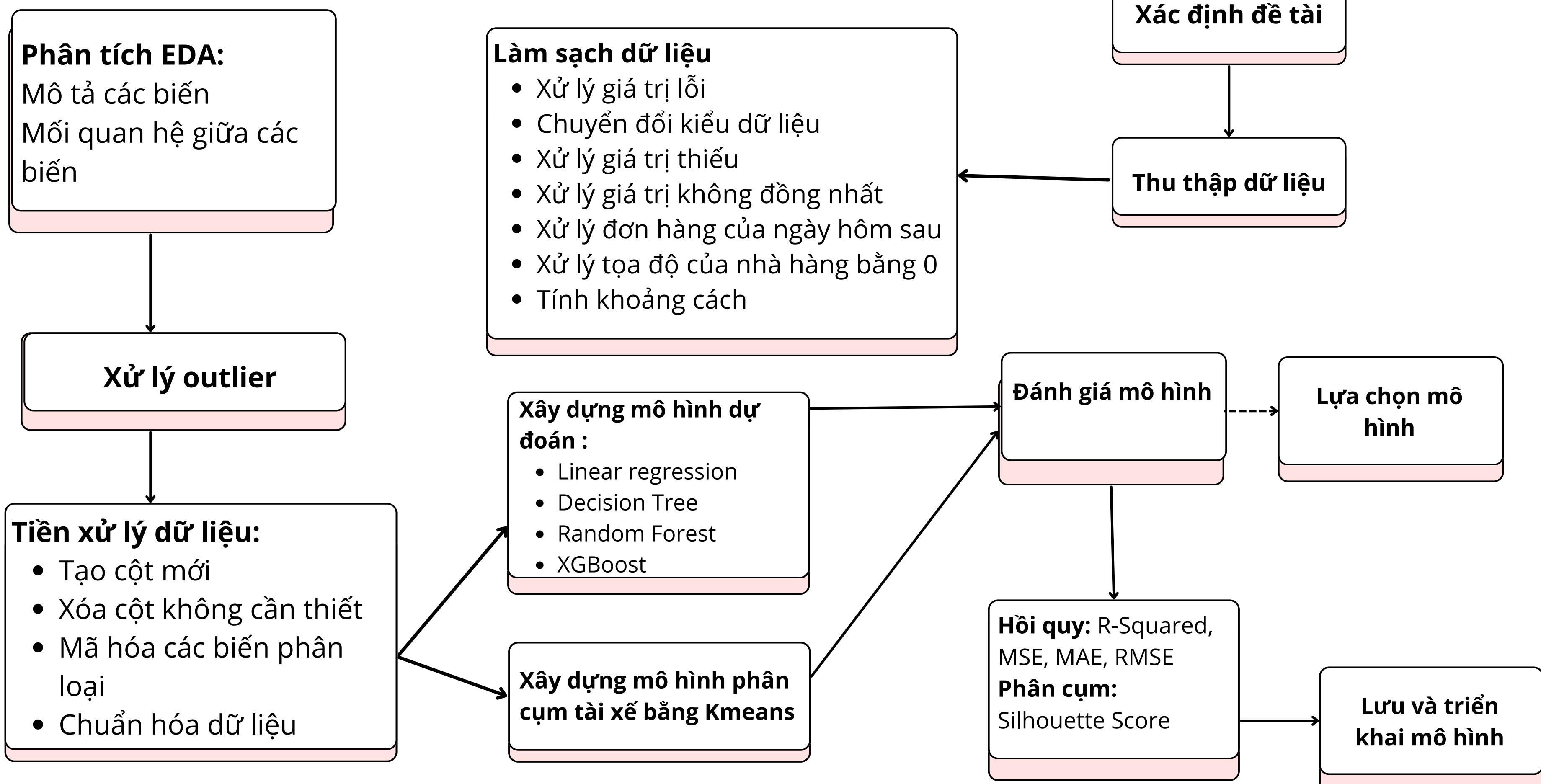
# Mô tả dữ liệu

- Bộ dữ liệu "Food Delivery Dataset" được lấy trên Kaggle thuộc lĩnh vực Logistics và Dịch vụ giao đồ ăn gồm 20 cột và 45.597 dòng

STT	Tên	Ý nghĩa	Kiểu dữ liệu
1	ID	Mã đơn hàng	String
2	Delivery_person_ID	Mã của người giao hàng	String
3	Delivery_person_Age	Tuổi của người giao hàng	Numeric
4	Delivery_person_Ratings	Xếp hạng của người giao hàng dựa trên các lần giao hàng trước	Numeric
5	Restaurant_latitude	Vĩ độ của nhà hàng	Numeric
6	Restaurant_longitude	Kinh độ của nhà hàng	Numeric
7	Delivery_location_latitude	Vĩ độ của địa điểm giao hàng	Numeric
8	Delivery_location_longitude	Kinh độ của địa điểm giao hàng	Numeric
9	Order_Date	Ngày đặt hàng	Date/time
10	Time_Orderd	Thời gian khách đặt hàng	Date/time

11	Time_Order_picked	Thời gian người giao hàng nhận đơn	Date/time
12	Weatherconditions	Điều kiện thời tiết	String
13	Road_traffic_density	Mật độ giao thông	String
14	Vehicle_condition	Tình trạng phương tiện của người giao hàng	Numeric
15	Type_of_order	Loại đơn hàng	String
16	Type_of_vehicle	Loại xe giao hàng	String
17	Multiple_deliveries	Số lượng đơn hàng giao cùng lúc	Numeric
18	Festival	Có phải ngày lễ không	Numeric
19	City	Loại thành phố	String
20	Time_taken(min)	Thời gian giao hàng	Numeric

# WORKFLOW



## 2. Tiền xử lý dữ liệu

# Làm sạch dữ liệu

# Làm sạch dữ liệu - Xử lý dữ liệu lỗi

- **Xử lý cột Weatherconditions:**

Tách chuỗi theo khoảng trắng. Giữ lại phần đầu tiên → Loại bỏ "conditions"

VD: "Cloudy conditions → "Cloudy"

- **Xử lý cột Time\_taken(min)**

Trích xuất phần số từ chuỗi, giúp xử lý dữ liệu số dễ dàng hơn.

- **Tách city\_code từ “Delivery\_person\_ID”**

Tách và lấy mã thành phố từ ID người giao hàng

- **Đổi tên cột “City” → “Area\_Type” và sửa lỗi chính tả** Metropolitan → Metropolitan

Weatherconditions	Time_taken(min)
conditions Sunny	(min) 24
conditions Stormy	(min) 33
conditions Sandstorms	(min) 26
conditions Sunny	(min) 21
conditions Cloudy	(min) 30
...	...
conditions Windy	(min) 32
conditions Windy	(min) 36
conditions Cloudy	(min) 16
conditions Cloudy	(min) 26
conditions Fog	(min) 36

Weatherconditions	Time_taken(min)
Sunny	24
Stormy	33
Sandstorms	26
Sunny	21
Cloudy	30
...	...
Windy	32
Windy	36
Cloudy	16
Cloudy	26
Fog	36

City	Time_taken(min)
Urban	(min) 24
Metropolitan	(min) 33
Urban	(min) 26
Metropolitan	(min) 21
Metropolitan	(min) 30
...	...
Metropolitan	(min) 32
Metropolitan	(min) 36
Metropolitan	(min) 16
Metropolitan	(min) 26
Metropolitan	(min) 36

Area_Type	Time_taken(min)	City_code
Urban	24	INDO
Metropolitan	33	BANG
Urban	26	BANG
Metropolitan	21	COIMB
Metropolitan	30	CHEN
...	...	...
Metropolitan	32	JAP
Metropolitan	36	AGR
Metropolitan	16	CHEN
Metropolitan	26	COIMB
Metropolitan	36	RANCHI

# Làm sạch dữ liệu - Chuyển đổi kiểu dữ liệu

- Chuyển đổi **các cột chứa số** sang kiểu **float64** để đảm bảo tính toán chính xác
- Chuyển đổi **Order\_Date** thành kiểu **datetime** để dễ dàng xử lý thời gian.

ID	object
Delivery_person_ID	object
Delivery_person_Age	object
Delivery_person_Ratings	object
Restaurant_latitude	float64
Restaurant_longitude	float64
Delivery_location_latitude	float64
Delivery_location_longitude	float64
Order_Date	object
Time_Orderd	object
Time_Order_picked	object
Weatherconditions	object
Road_traffic_density	object
Vehicle_condition	int64
Type_of_order	object
Type_of_vehicle	object
multiple_deliveries	object
Festival	object
Area_Type	object
Time_taken(min)	object
City_code	object
dtype: object	



ID	object
Delivery_person_ID	object
Delivery_person_Age	float64
Delivery_person_Ratings	float64
Restaurant_latitude	float64
Restaurant_longitude	float64
Delivery_location_latitude	float64
Delivery_location_longitude	float64
Order_Date	object
Time_Orderd	object
Time_Order_picked	object
Weatherconditions	object
Road_traffic_density	object
Vehicle_condition	float64
Type_of_order	object
Type_of_vehicle	object
multiple_deliveries	float64
Festival	object
Area_Type	object
Time_taken(min)	float64
City_code	object
dtype: object	

# Làm sạch dữ liệu - Xử lý dữ liệu không đồng nhất

- Xử lý trường hợp một tài xế có nhiều độ tuổi khác nhau trong 1 ngày**

Ta nhận thấy có **tất cả các dòng** trong bộ dữ liệu đều gặp lỗi này

→ Thay thế bằng cách đồng nhất độ tuổi bằng cách lấy **mode** theo từng tài xế

- Kiểm tra trùng lặp:** Không có giá trị trùng lặp

	Delivery_person_ID	Delivery_person_Age	Time_Ordered
92	AGRRES010DEL01	34.0	23:00:00
4558	AGRRES010DEL01	38.0	22:45:00
7750	AGRRES010DEL01	21.0	22:55:00
12437	AGRRES010DEL01	34.0	08:25:00
13244	AGRRES010DEL01	39.0	13:30:00
15371	AGRRES010DEL01	25.0	09:15:00
16556	AGRRES010DEL01	NaN	NaN
18240	AGRRES010DEL01	38.0	12:00:00
20236	AGRRES010DEL01	34.0	17:45:00
24595	AGRRES010DEL01	20.0	22:15:00
29279	AGRRES010DEL01	36.0	13:25:00
33710	AGRRES010DEL01	25.0	19:00:00
35450	AGRRES010DEL01	34.0	16:30:00
38976	AGRRES010DEL01	25.0	14:15:00

# Làm sạch dữ liệu - Xử lý giá trị thiếu

- **Kiểm tra giá trị thiếu:**

Thay thế tất cả giá trị '**NaN**' (ở dạng chuỗi) thành **np.nan** (giá trị NaN thực tế trong **NumPy**), giúp xử lý giá trị thiếu dễ dàng hơn.

- **Giải pháp**

- Điền giá trị thiếu trong '**Area\_Type**' bằng **mode**
- Điền giá trị thiếu trong '**Delivery\_person\_Ratings**' bằng **median**
- Điền giá trị thiếu trong các cột phân loại **Weatherconditions**', '**Festival**', '**multiple\_deliveries**', '**Road\_traffic\_density**' bằng **mode** theo '**City\_code**'
- Điền giá trị thiếu trong '**Time\_Orderd**" bằng **KNN Imputer**

Column	Null Count	Null Percentage
Delivery_person_Ratings	1908	4.184853
Time_Orderd	1731	3.796635
Area_Type	1200	2.631983
multiple_deliveries	993	2.177966
Weatherconditions	616	1.351085
Road_traffic_density	601	1.318185
Festival	228	0.500077

# Làm sạch dữ liệu - Xử lý TH đơn hàng bị lấy ngày hôm sau

## Vấn đề cần xử lý:

- Time\_Orderd và Time\_Order\_picked chỉ lưu thời gian trong ngày (HH:MM:SS), không có thông tin ngày tháng.
- Nếu đơn hàng được đặt vào 23:30 và được lấy vào 00:15, thì Time\_Order\_picked sẽ nhỏ hơn Time\_Orderd, gây sai lệch khi tính toán.

## Giải pháp:

- Nếu **Time\_Order\_picked < Time\_Orderd** (đơn hàng được lấy vào ngày hôm sau)  
• Cộng 1 ngày (**pd.DateOffset(1)**) vào **Order\_Date** trước khi cộng thêm **Time\_Order\_picked**.
- Ngược lại, chỉ cần cộng **Time\_Order\_picked** vào **Order\_Date**.

Time_Orderd_formatted	Time_Order_picked_formatted
2022-03-19 11:30:00	2022-03-19 11:45:00
2022-03-25 19:45:00	2022-03-25 19:50:00
2022-03-19 08:30:00	2022-03-19 08:45:00
2022-04-05 18:00:00	2022-04-05 18:10:00
2022-03-26 13:30:00	2022-03-26 13:45:00

# Làm sạch dữ liệu - Xử lý tọa độ nhà hàng bằng 0

## Vấn đề cần xử lý:

- Trong thực tế, không có nhà hàng nào có tọa độ (0,0), nên đây có thể là:
- Dữ liệu bị thiếu và được gán mặc định thành 0.0.
- Dữ liệu bị nhập sai do lỗi hệ thống hoặc sai sót khi thu thập dữ liệu

## Giải pháp

- Thay thế các giá trị tọa độ bằng giá trị NaN nếu bằng 0
- Thay thế giá trị NaN bằng **mode** theo từng nhóm

## City\_code

	Restaurant_latitude	Restaurant_longitude
33	0.0	0.0
52	0.0	0.0
57	0.0	0.0
59	0.0	0.0
67	0.0	0.0
...	...	...
45569	0.0	0.0
45576	0.0	0.0
45577	0.0	0.0
45579	0.0	0.0
45589	0.0	0.0

[3640 rows x 2 columns]

Empty DataFrame

Columns: [Restaurant\_latitude, Restaurant\_longitude]

Index: []



# Làm sạch dữ liệu - Tính khoảng cách

Distance_km
4.599023
19.995664
7.998623
3.401680
5.813103
...
1.513097
8816.011996
1.133420
2.199462
10.358311

## Tính khoảng cách địa lý giữa nhà hàng và vị trí giao hàng bằng công thức Haversine

**công thức Haversine** là một phương pháp trong hình học cầu để tính khoảng cách ngắn nhất giữa hai điểm trên bề mặt Trái Đất (hoặc một hình cầu bất kỳ), dựa trên vĩ độ và kinh độ.

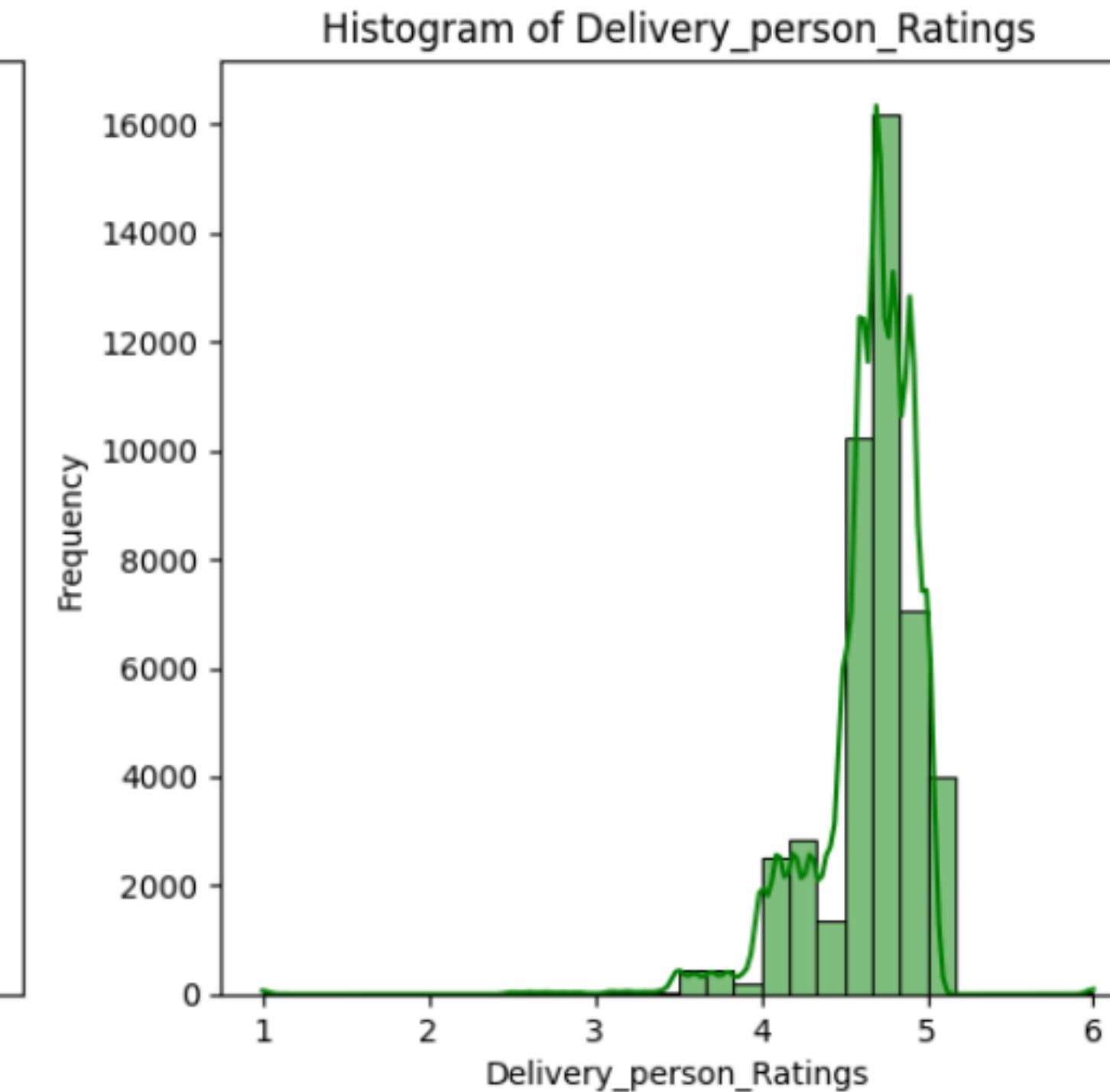
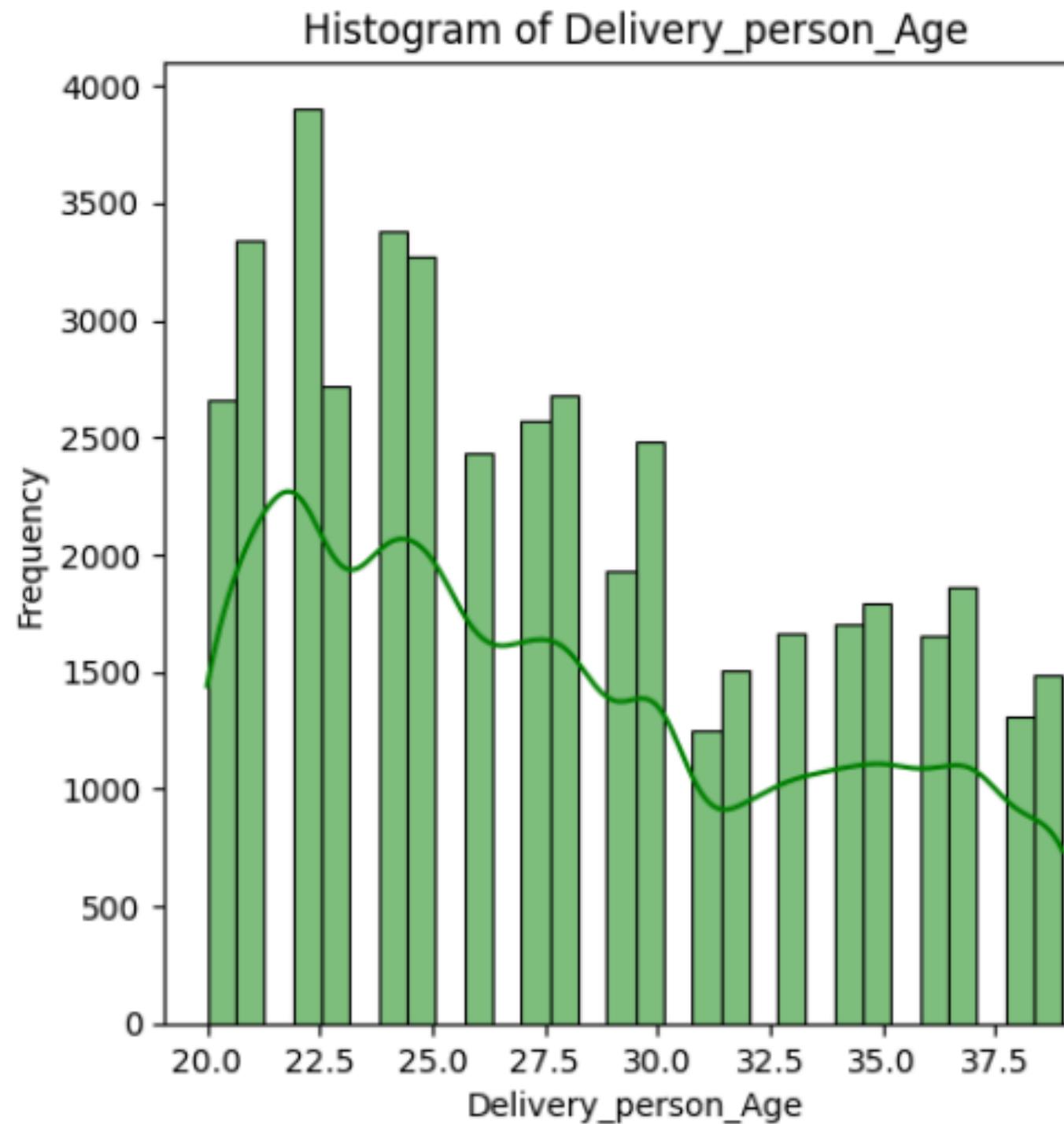
Công thức Haversine (biến thể dùng arccos):

Tính khoảng cách cung tròn giữa 2 điểm trên mặt cầu:

$$d = R \cdot \arccos (\sin(\text{lat}_1) \cdot \sin(\text{lat}_2) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \cos(\text{lon}_1 - \text{lon}_2))$$

# Phân tích khám phá EDA

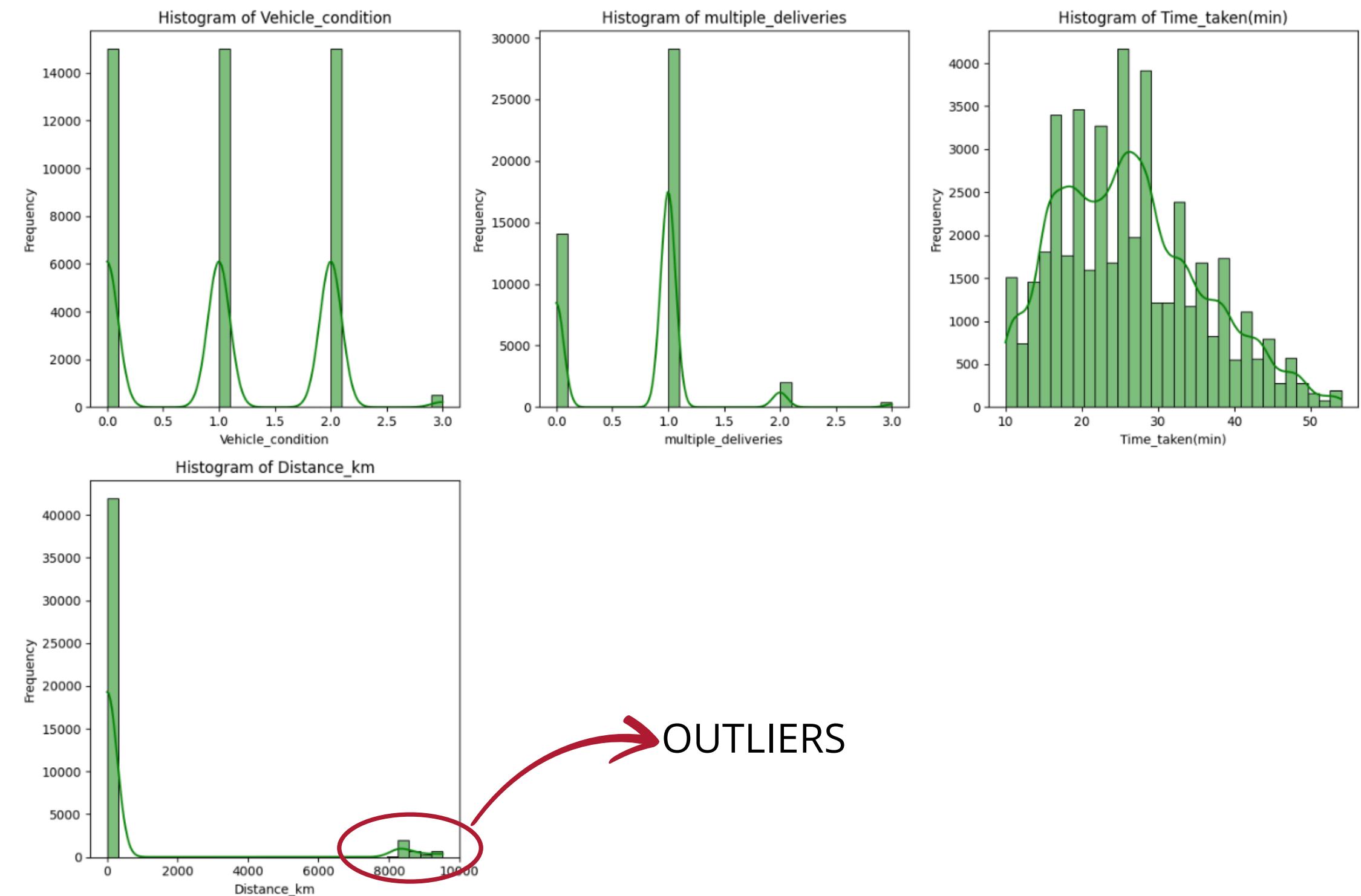
# 1. Các cột dữ liệu numeric



- Nhìn chung tuổi của tài xế **phân bố khá đồng đều** và có xu hướng **giảm dần về bên phải**
- Đánh giá tài xế: **Hầu hết** điểm số đánh giá nằm trong **khoảng 4 - 5**, có xu hướng **lệch trái** (phần lớn nhận điểm cao).

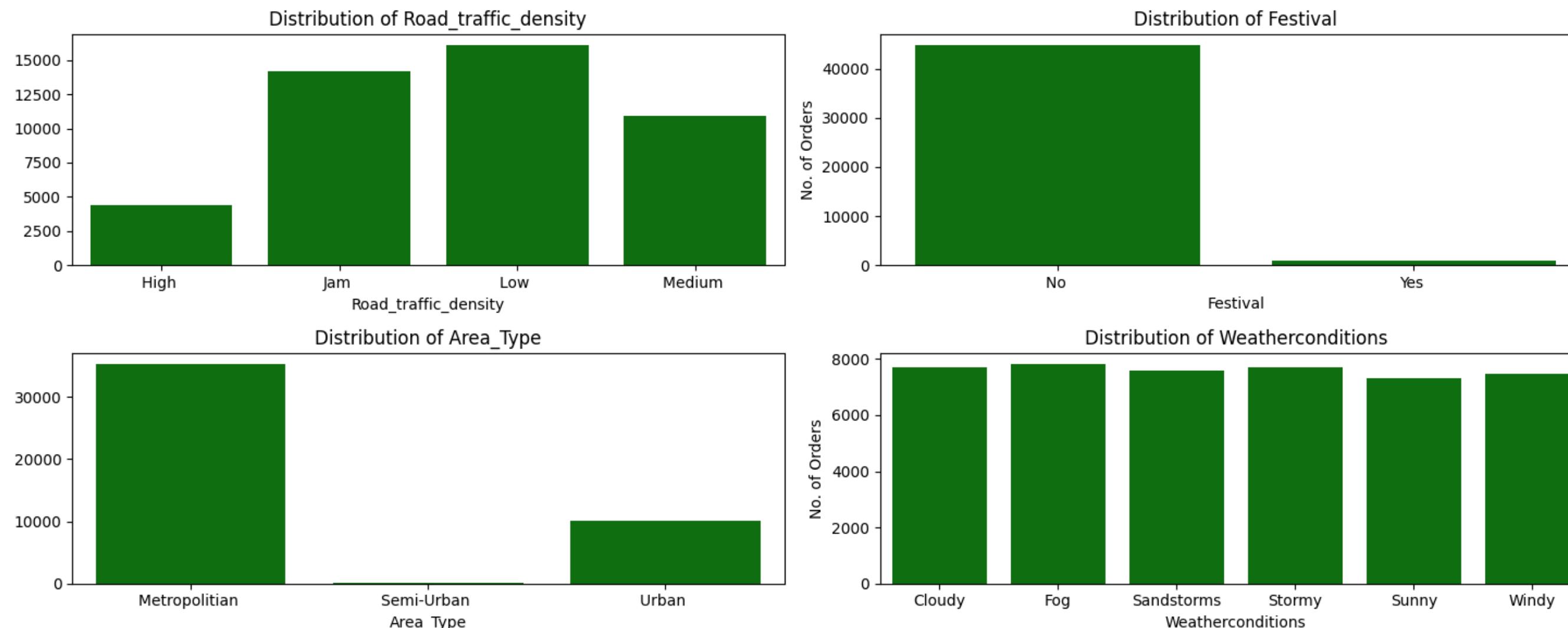
# 1. Các cột dữ liệu numeric

- Tình trạng phương tiện nằm ở ngưỡng **0-2** (Tức là xe hoạt động tốt, không hư hỏng nhiều) là **nhiều nhất**, mức 3 (tình trạng kém) chiếm khá ít
- Thời gian giao hàng phân phối **lệch phải**, nghĩa là phần lớn đơn hàng có thời gian giao hàng ngắn, nhưng vẫn có một số đơn hàng mất nhiều thời gian hơn.



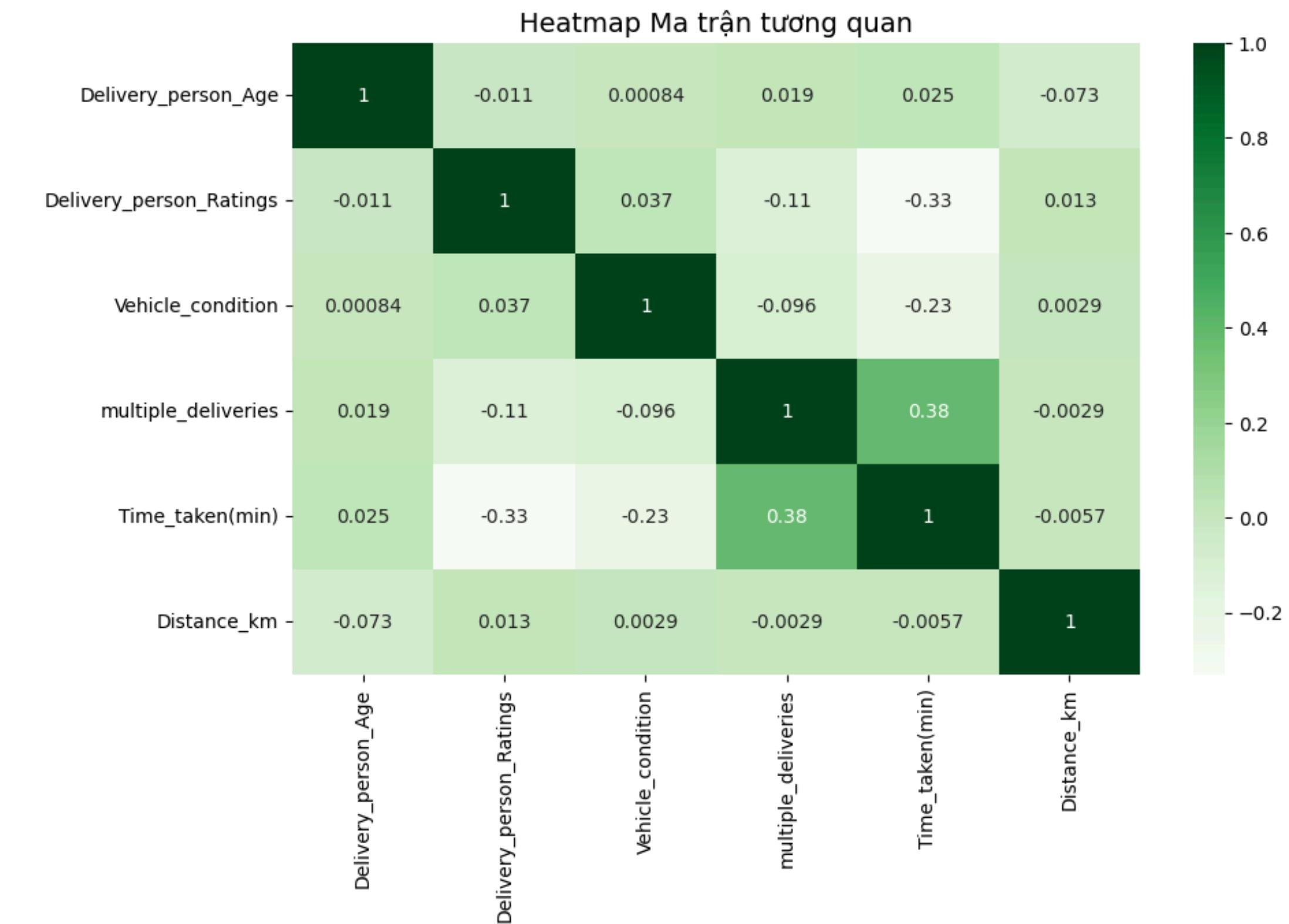
## 2. Các cột dữ liệu phân loại

- Mức "**High**" (**cao**) **có ít đơn hàng nhất** → Có thể do giao thông quá tệ nên cản trở việc giao hàng → Giao hàng thường được thực hiện khi giao thông không quá tệ (low) hoặc khi nhu cầu cao dù bị tắc đường (Jam).
- Hơn 40.000 đơn hàng không diễn ra trong dịp lễ hội. **Số đơn hàng trong lễ hội rất ít** → Hoạt động giao hàng giảm trong các ngày lễ.
- Khu vực **Semi-Urban (bán đô thị)** có số lượng đơn hàng thấp nhất → Có thể do dân cư thưa thớt hoặc mức độ tiếp cận thương mại điện tử thấp.



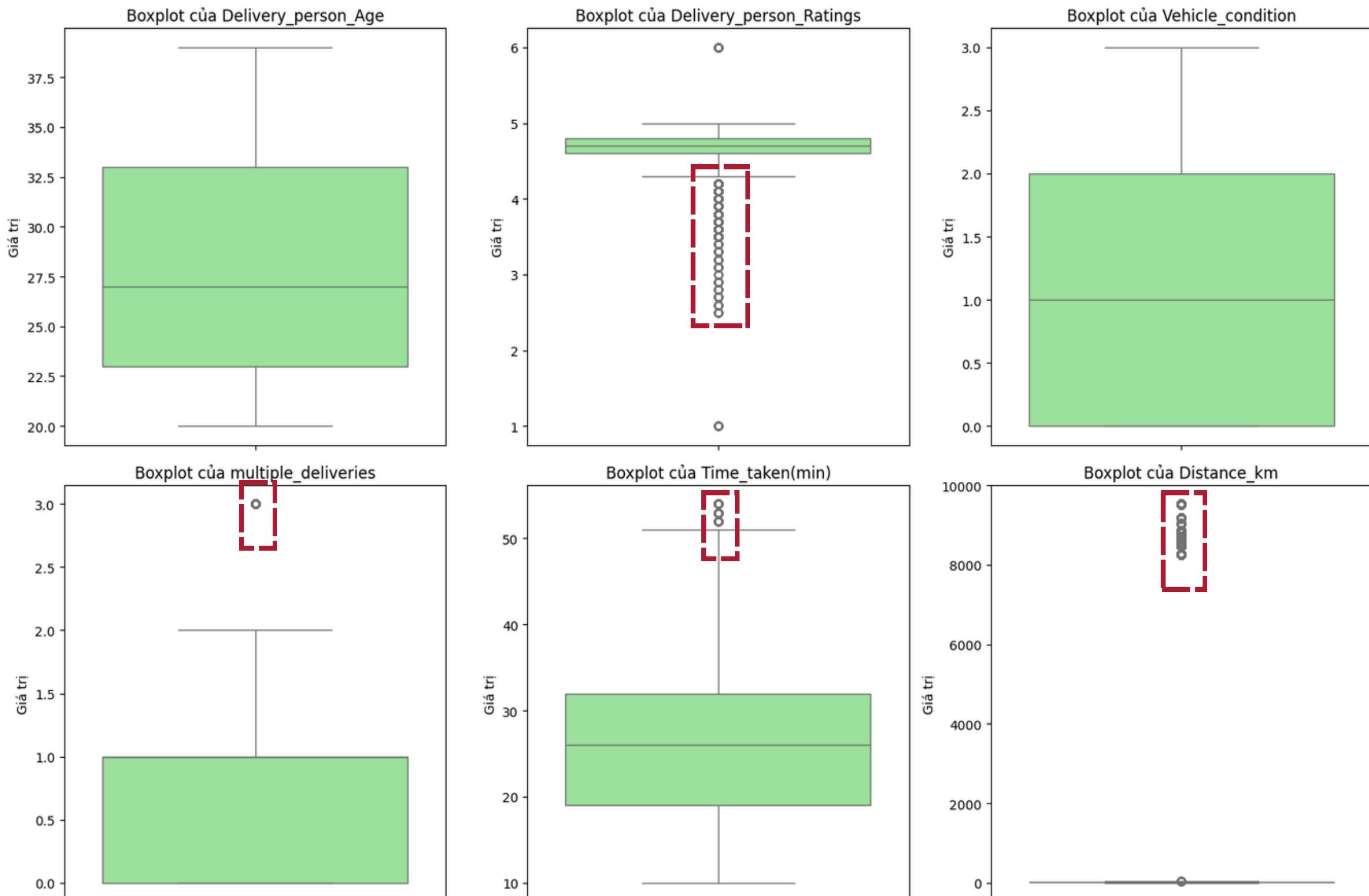
### 3. Kiểm định mối tương quan

- Cặp biến Time\_taken(min) & multiple\_deliveries có tương quan dương **0.38**: nhiều đơn giao cùng lúc → tăng thời gian giao hàng.
- Time\_taken(min) & Delivery\_person\_Ratings có tương quan âm **-0.33**: Xếp hạng cao thường đi kèm với thời gian giao hàng ngắn hơn.
- Time\_taken(min) & Vehicle\_condition có tương quan âm **-0.23**: Xe tốt → thời gian giao hàng có xu hướng giảm.



## 4. Kiểm tra giá trị ngoại lai

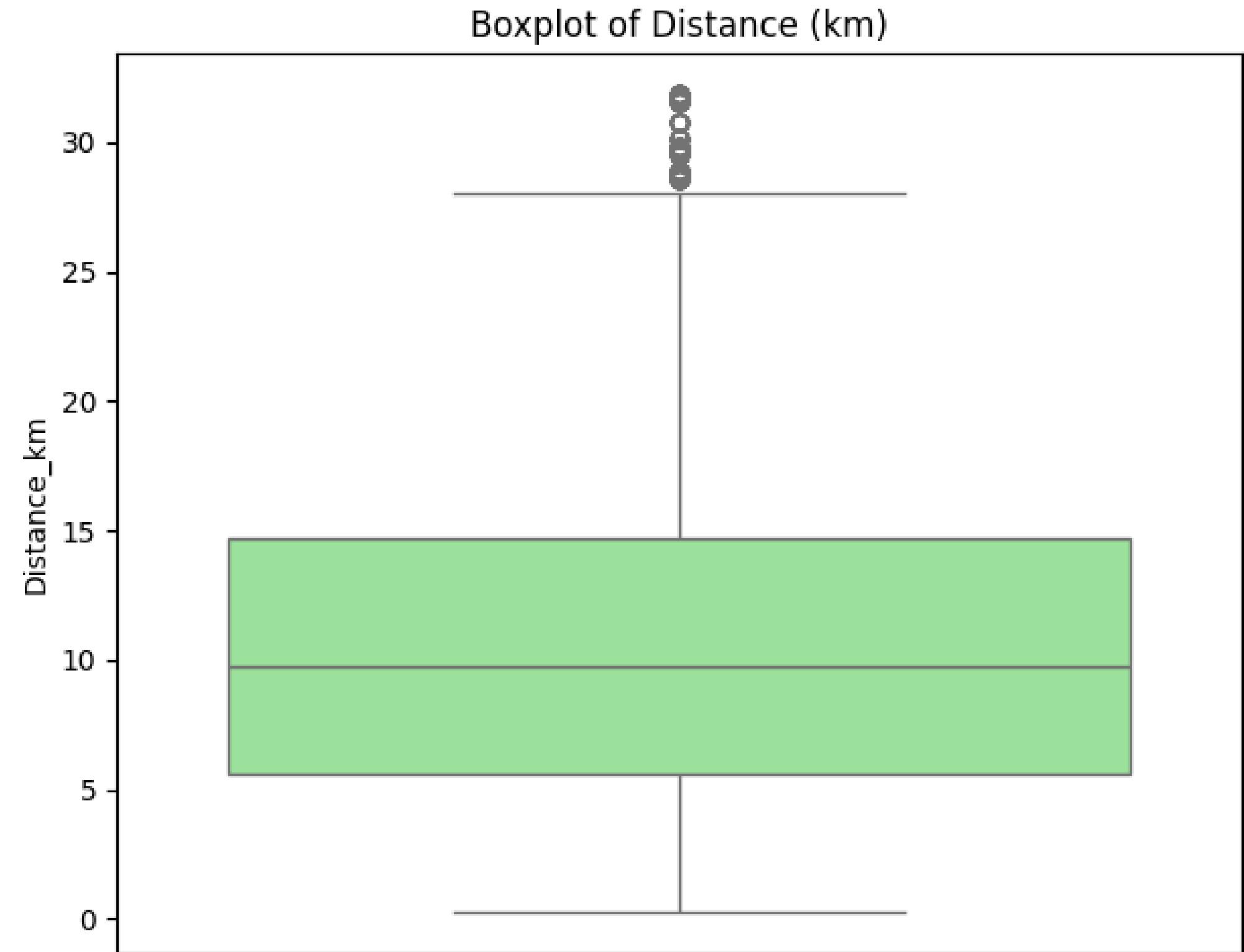
- Delivery\_person\_Ratings: Có nhiều ngoại lai ở phía dưới, có thể do một số tài xế bị đánh giá rất thấp.
- Multiple\_deliveries: Có một số ngoại lai ở phía trên, cho thấy một số đơn hàng có nhiều lần giao hơn mức bình thường.
- Time\_taken(min): Có một số giá trị ngoại lai lớn, cho thấy một số đơn hàng có thời gian giao hàng dài hơn đáng kể so với mức trung bình.
- Distance\_km: Xuất hiện nhiều ngoại lai phía trên, nghĩa là một số đơn hàng có khoảng cách di chuyển rất xa so với đa số dữ liệu.



## 5. Xử lý outlier của Distance\_km

- Đa số khoảng cách giao hàng nằm trong khoảng từ **~2 km đến ~20 km.**
- Có một số giá trị ngoại lai lớn ( $> 28$  km, lên đến hơn 30 km).

=> Sử dụng phương pháp IQR



### **3. Phân tích các yếu tố ảnh hưởng đến hiệu suất giao hàng**



Welcome to

## OVERVIEW DASHBOARD

Trang này cung cấp cái nhìn tổng quan về hiệu suất giao hàng, bao gồm tổng đơn hàng, phân bố theo tháng, ngày, khung giờ và các thành phố hàng đầu, giúp nắm bắt được kết quả kinh doanh của doanh nghiệp.

Month

All

Time\_Slot

All

Total orders

42K

City

22

Delivery persons

1167

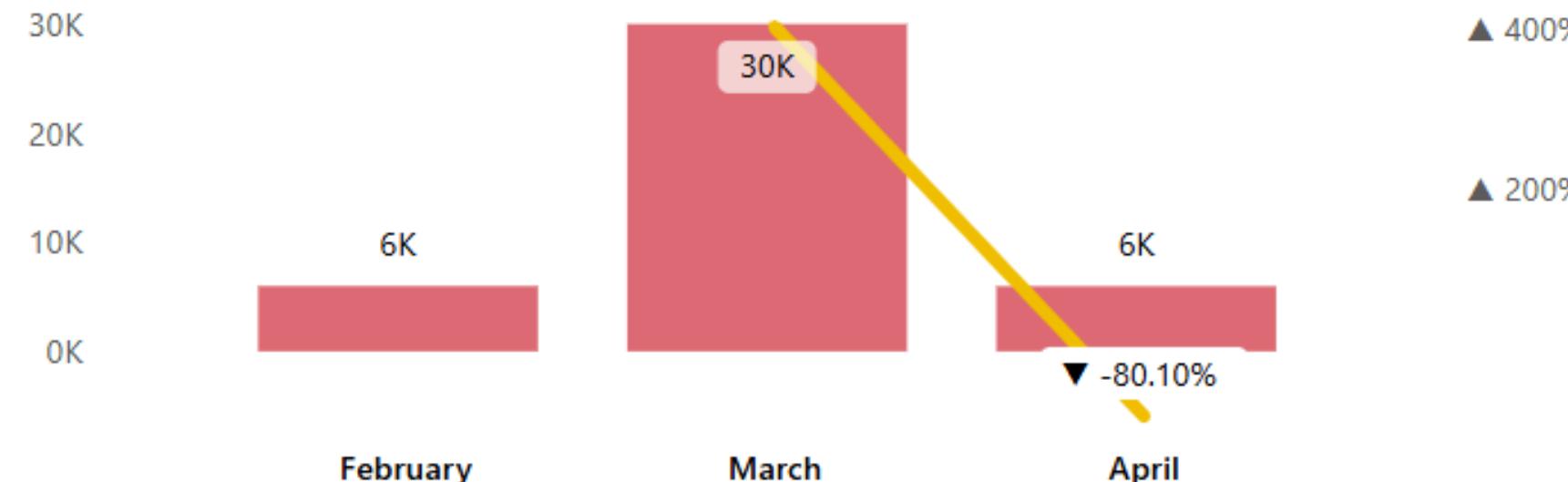
AVG Delivery Time

26.31 min

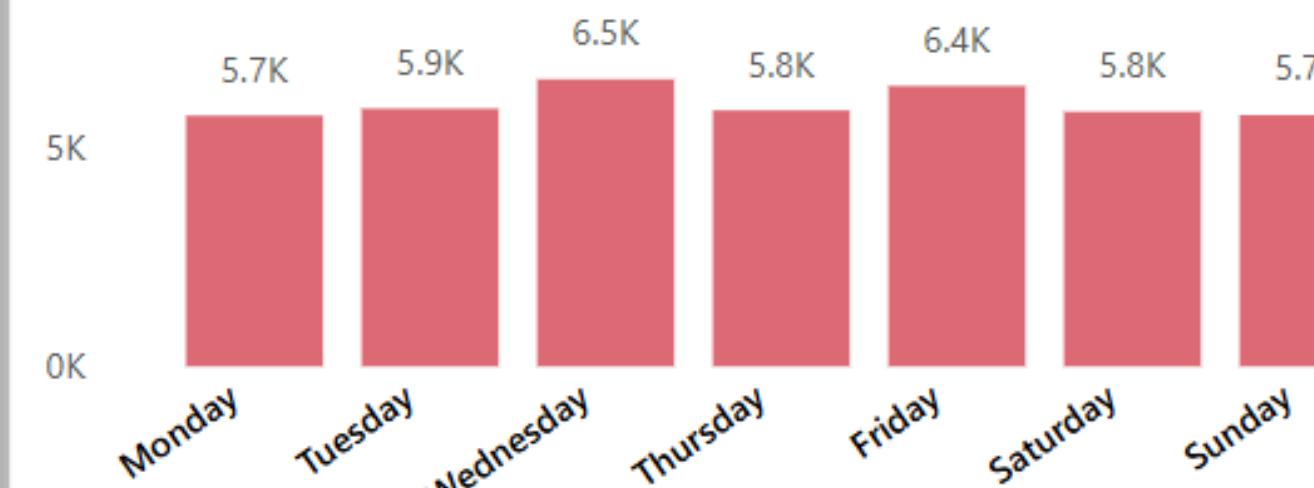
AVG Distance

10.55 Km

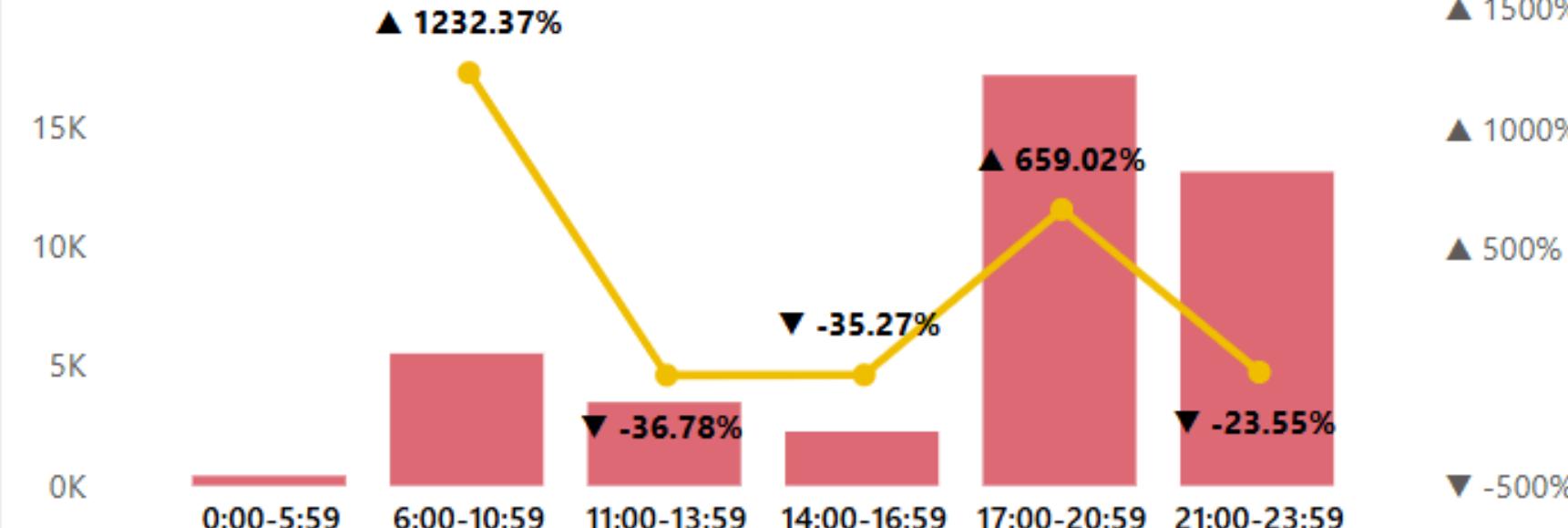
### Total Orders by Month



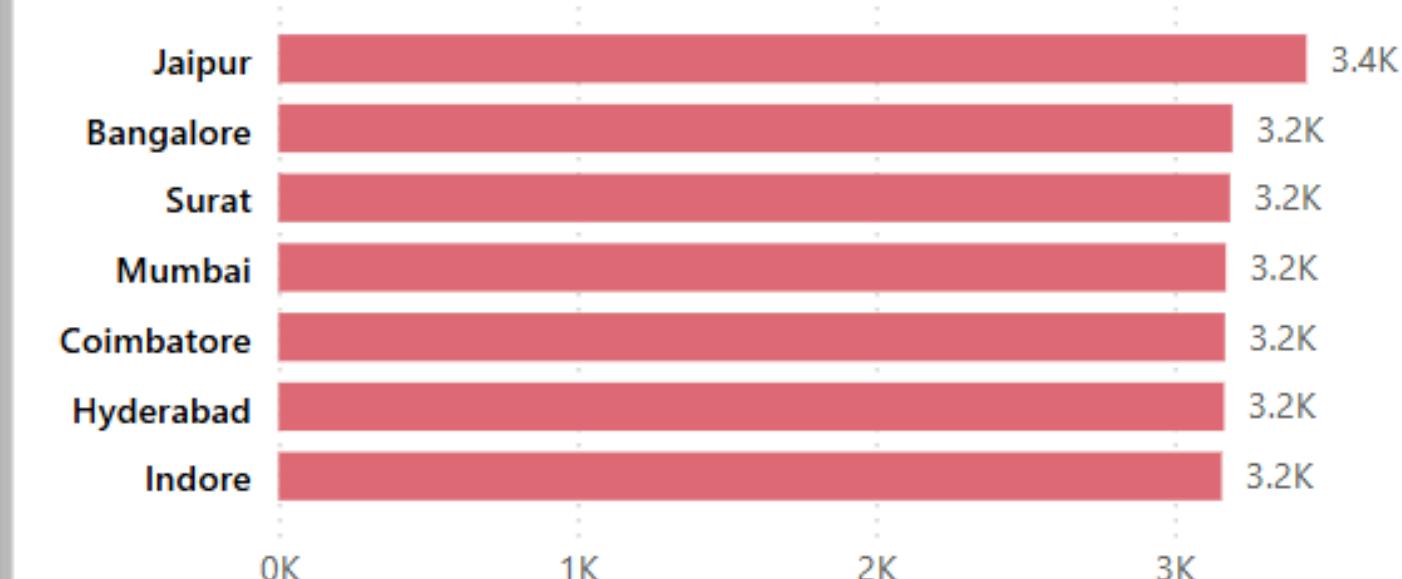
### Total Orders by Weekday



### Total orders by Time Slot



### Top Cities by Total Orders

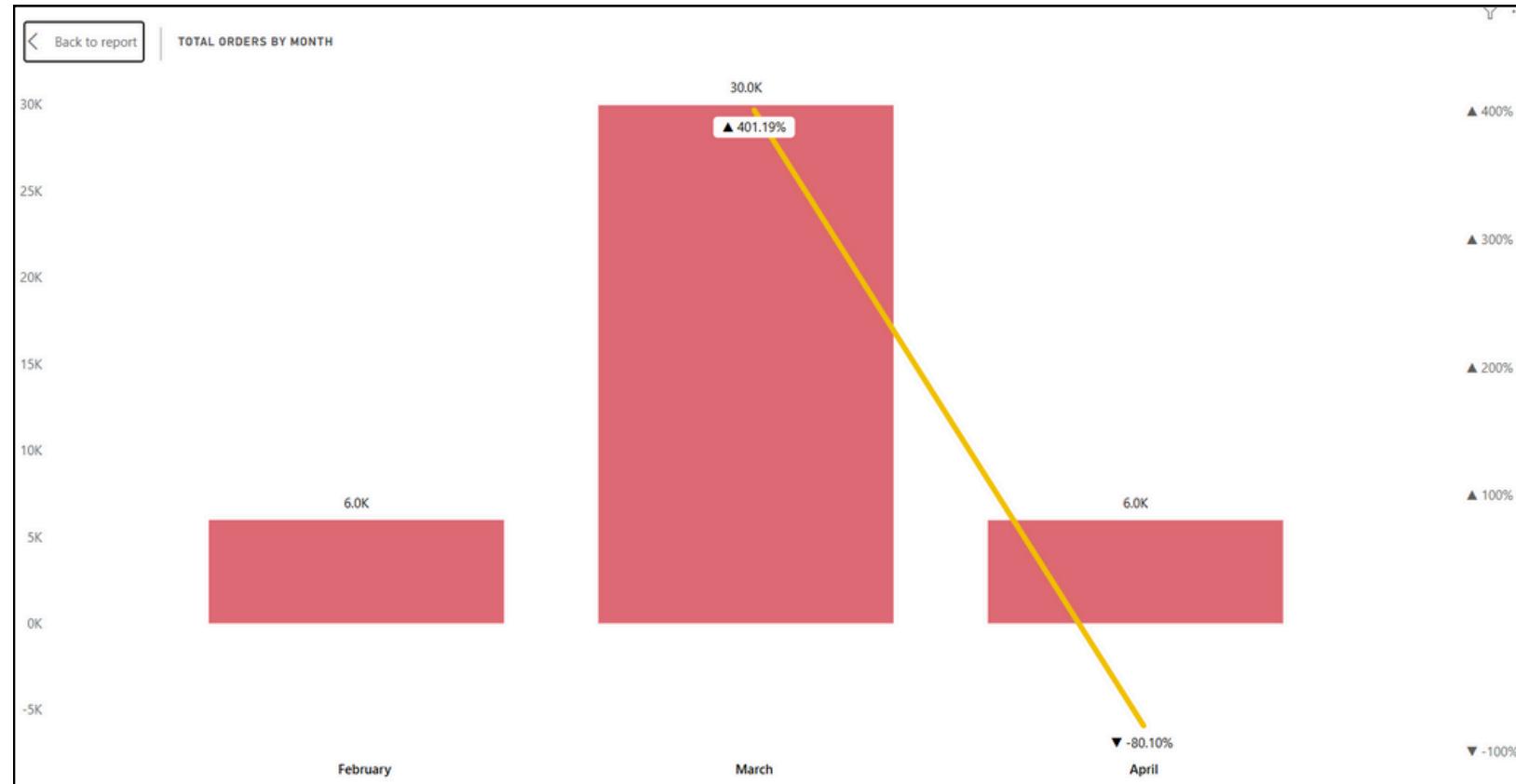


Introduction

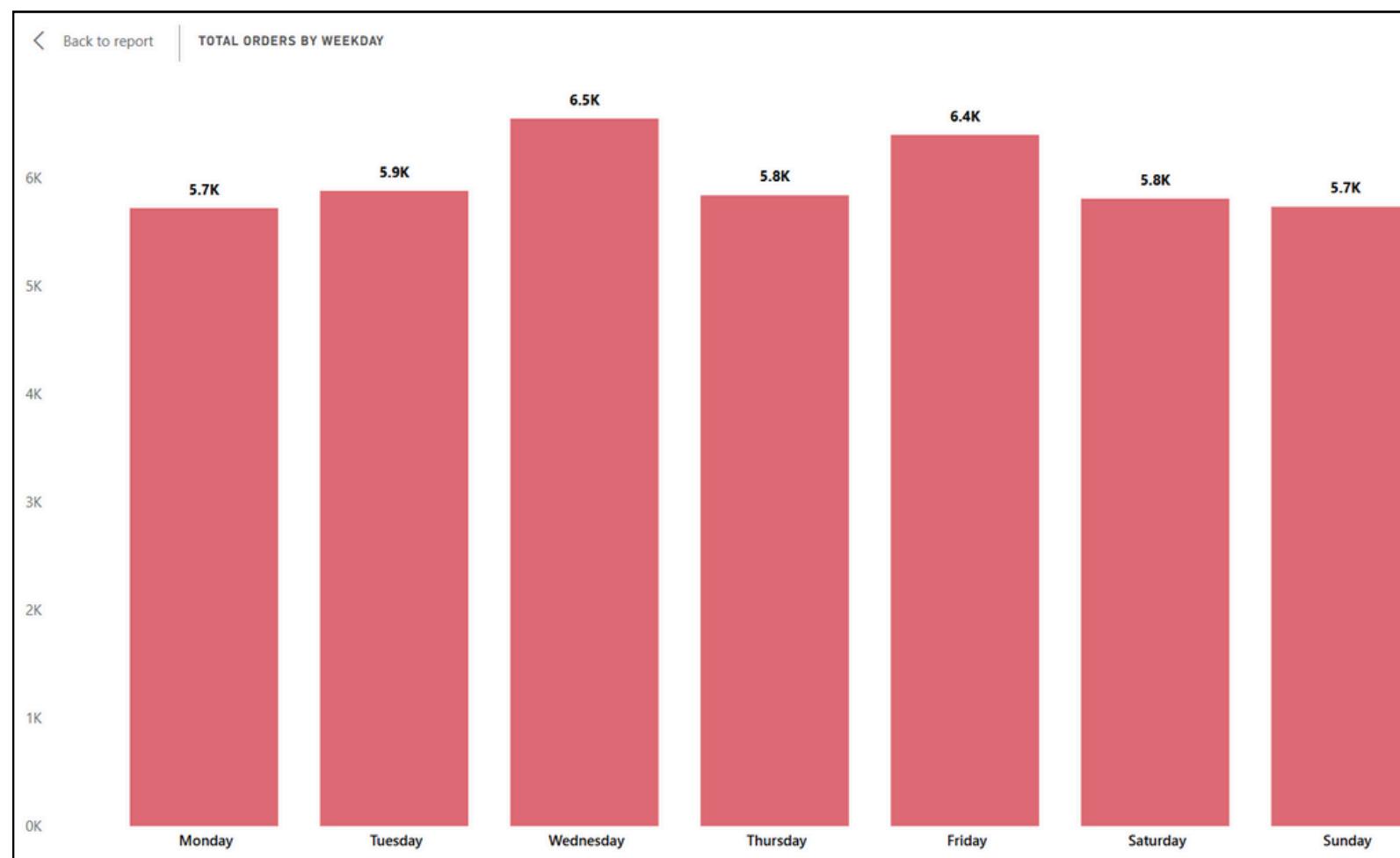
Overview

Delivery Analyst

# Tổng quan về tình hình kinh doanh

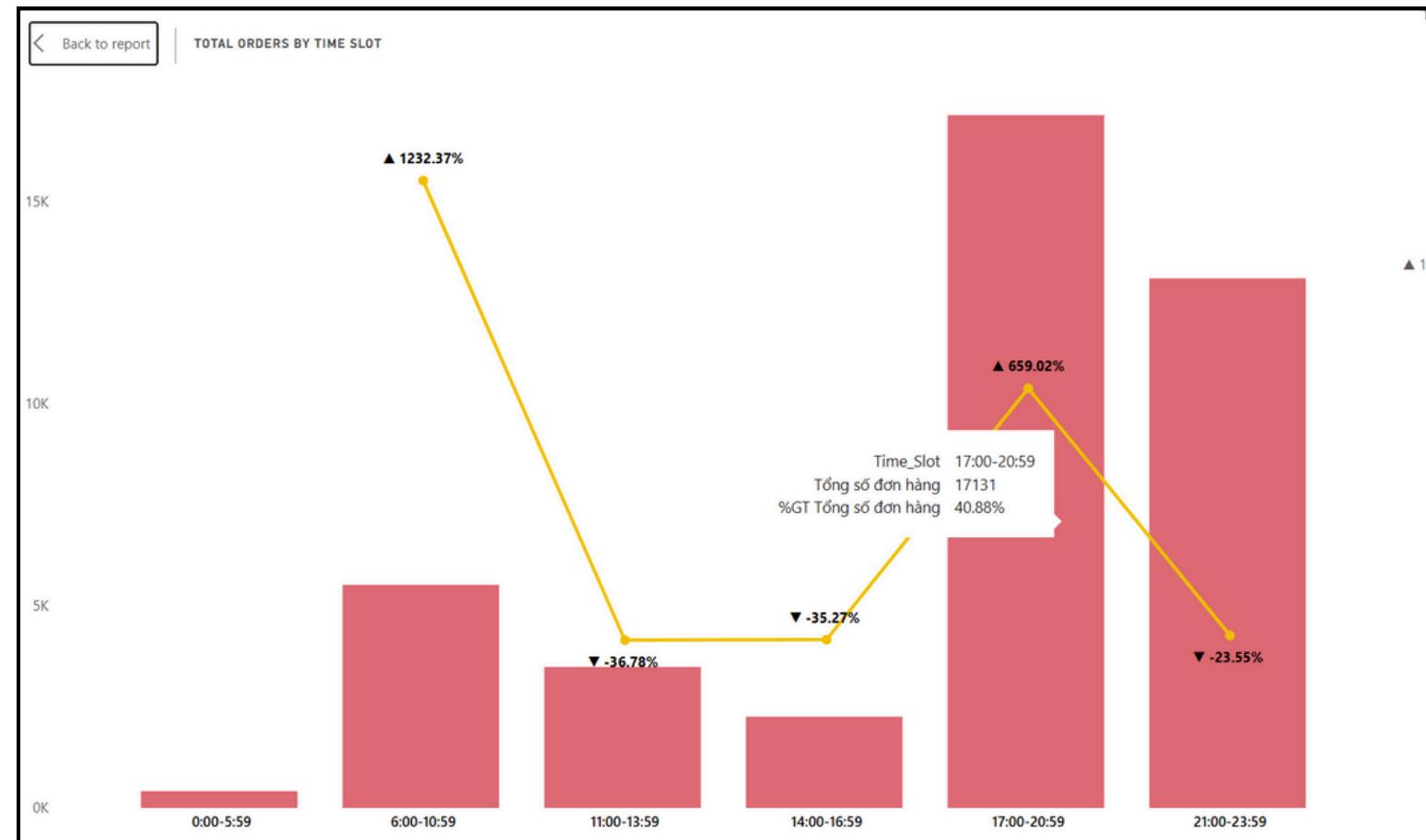


- Số đơn hàng biến động mạnh qua các tháng. Cụ thể, tháng 2/2022 có 5,978 đơn, chiếm 14.27% tổng số, tăng lên 29,961 đơn trong tháng 3/2022, tương ứng với mức tăng 401.19% so với tháng trước. Tuy nhiên, tháng 4/2022 lại giảm mạnh xuống 5,962 đơn (14.23%), giảm 80.1% so với tháng 3.

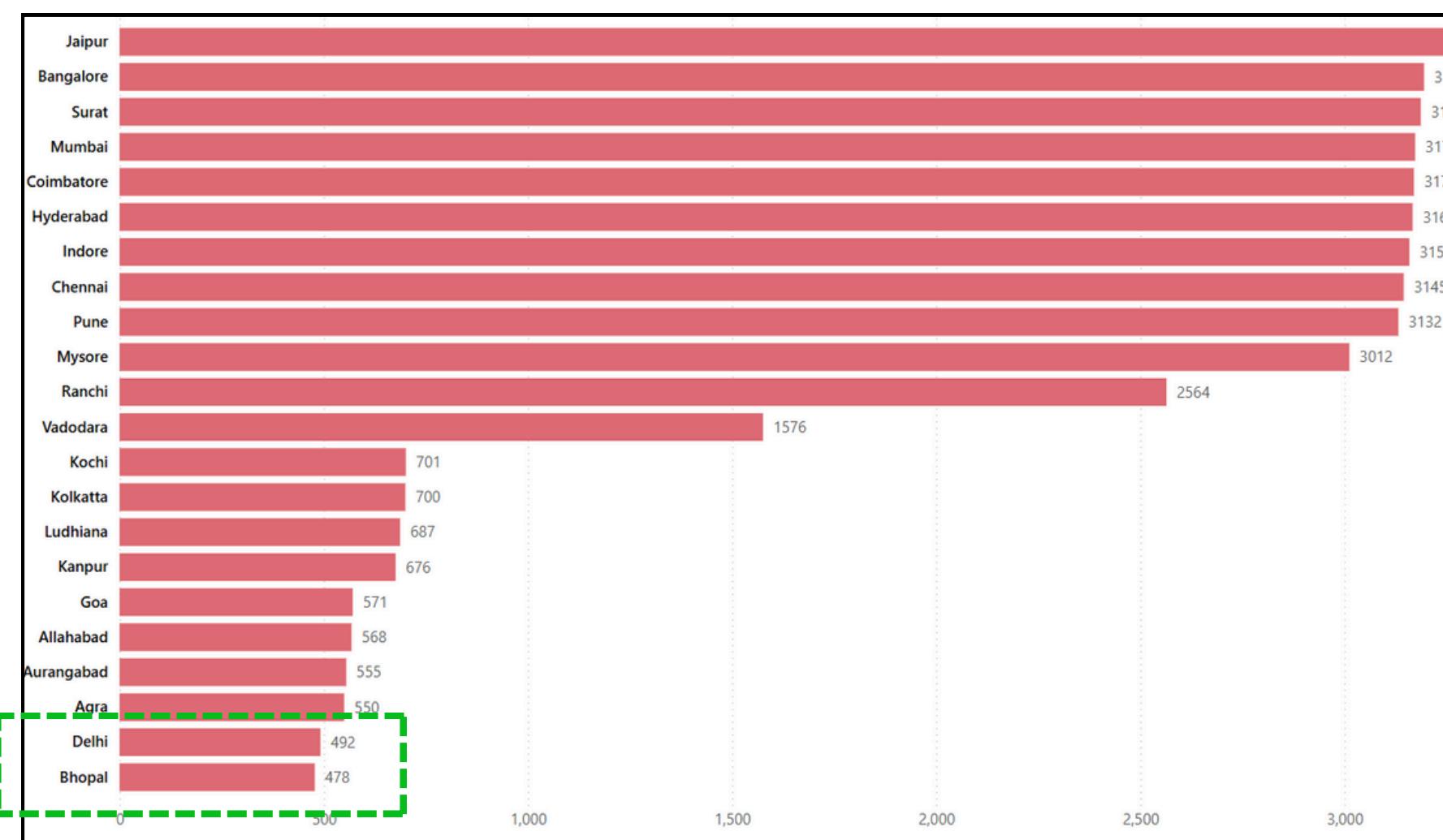


- Số đơn hàng phân bố tương đối đều qua các ngày trong tuần, nhưng Thứ Tư (6,545 đơn, 15.62%) và Thứ Sáu (6,394 đơn, 15.26%) nổi bật là các ngày cao điểm.

# Tổng quan về tình hình kinh doanh



- Khung giờ 17:00-20:59 ghi nhận 17,131 đơn, chiếm 40.9% tổng số, với mức tăng 659% so với khung 14:00-16:59 (2,257 đơn). Trong khi đó, khung 0:00-5:59 chỉ có 414 đơn, thấp nhất trong ngày.

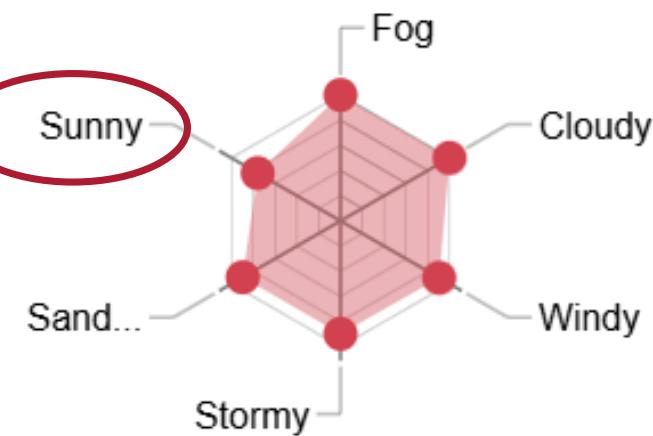


- Jaipur dẫn đầu với 3,443 đơn (8.22%), tiếp theo là Bangalore (3,195 đơn, 7.63%) và Surat (3,187 đơn, 7.61%)
- Thành phố lớn như Delhi (492 đơn, 1.17%) và Bhopal (478 đơn, 1.14%) có tỷ lệ đơn hàng thấp bất thường

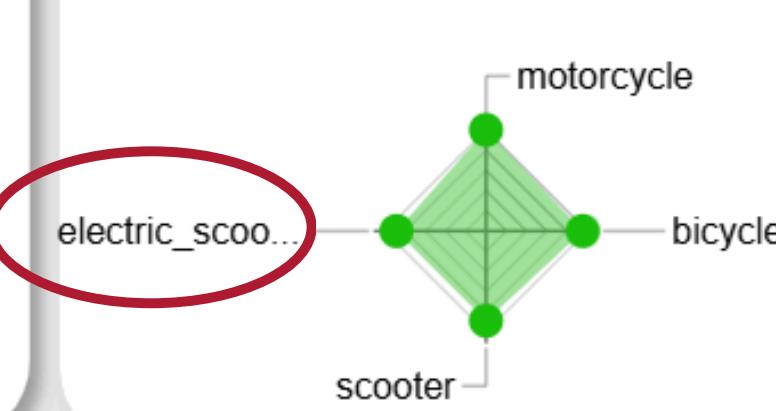
# Hiệu suất giao hàng

## AVG Delivery Time VS

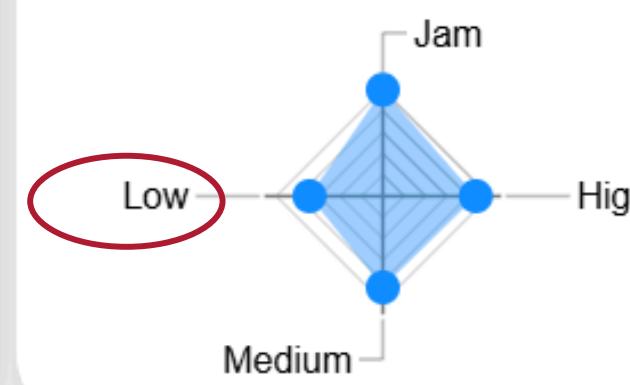
**Weatherconditions**



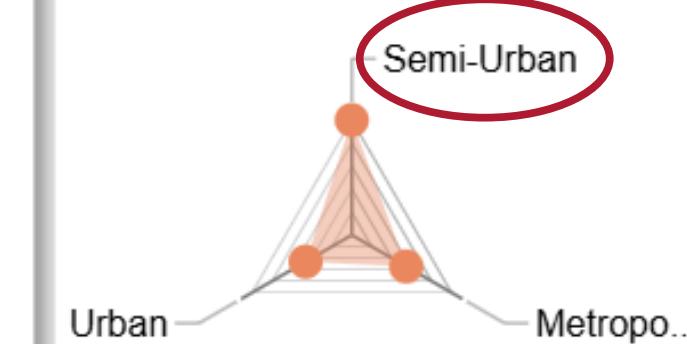
**Vehicle Type**



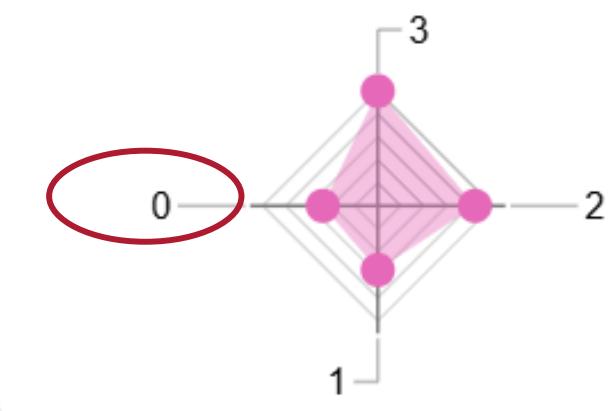
**Road traffic density**



**Area Type**



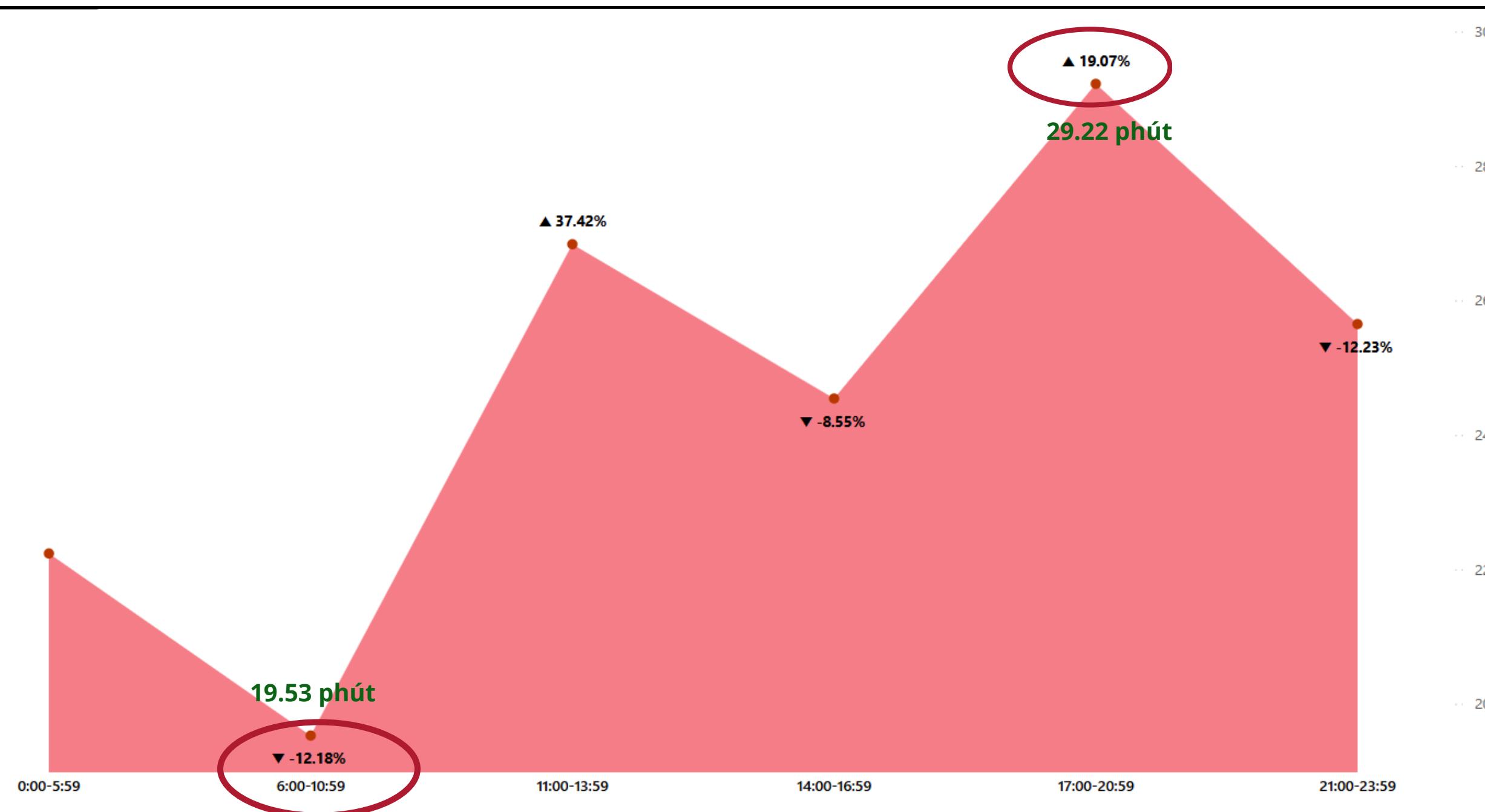
**Multiple deliveries**



- Thời gian giao hàng nhanh nhất trong điều kiện nắng (21.92 phút) và giao thông thấp (21.47 phút), nhưng chậm nhất trong sương mù (28.89 phút) và giao thông tắc nghẽn (31.17 phút).
- Xe điện (24.45 phút) và xe tay ga (24.51 phút) vượt trội so với xe máy (27.61 phút)
- Khu vực bán đô thị có thời gian giao hàng cao nhất (49.68 phút), gấp đôi khu vực đô thị (23.02 phút)
- Các đơn hàng không có đơn ghép thường được giao nhanh hơn

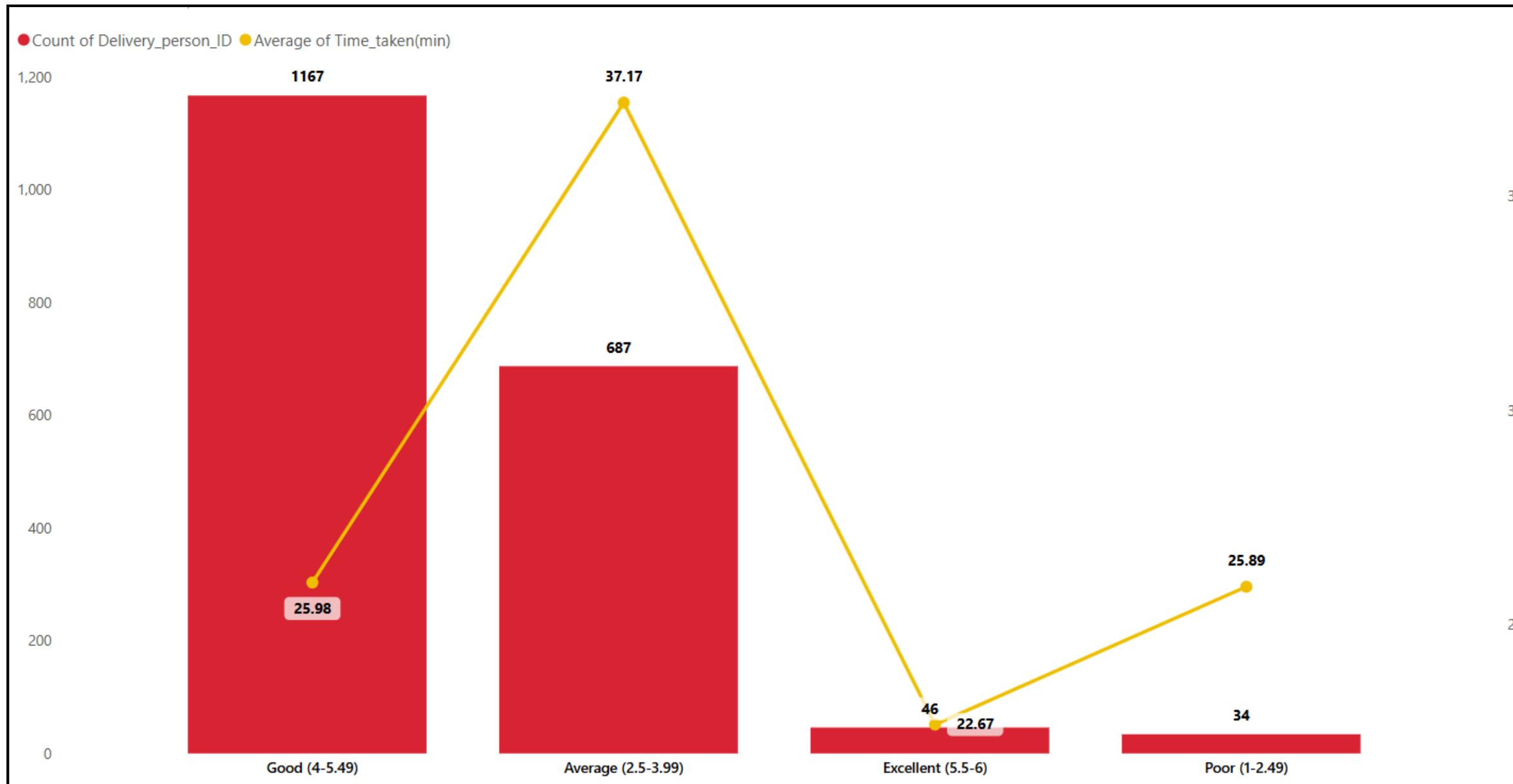
# Hiệu suất giao hàng

## AVG Delivery Time by Time Slot



- Khung 17:00-20:59 có thời gian trung bình cao nhất (29.22 phút), trong khi khung 6:00-10:59 nhanh nhất (19.53 phút).
- Các khung giờ khác như 11:00-13:59 (26.84 phút) và 21:00-23:59 (25.65 phút) nằm ở mức trung gian.
- Khung 17:00-20:59 có thời gian giao hàng chậm nhất do đây là khung giờ cao điểm với số lượng đơn hàng lớn (17,131 đơn)

## Driver Distribution by Rating Group vs AVG Delivery Time



- Nhóm tài xế Good (4-5.49) chiếm đa số (1,167 tài xế, xử lý 40,588 đơn chiếm 96.9% tổng số đơn hàng), với thời gian giao hàng trung bình 25.98 phút. Nhóm Excellent (5.5-6) chỉ có 46 tài xế (46 đơn), nhưng hiệu quả cao nhất với 22.67 phút/đơn
- Nhóm Average (2.5-3.99) có 687 tài xế (1,232 đơn) nhưng thời gian giao hàng chậm nhất (37.17 phút), và nhóm Poor (1-2.49) với 34 tài xế (35 đơn) có thời gian 25.89 phút.



Welcome to

# DELIVERY DASHBOARD

Trang này tập trung phân tích hiệu quả giao hàng qua các yếu tố như thời gian giao trung bình, điều kiện thời tiết, loại phương tiện và đánh giá tài xế, hỗ trợ nâng cao trải nghiệm khách hàng.

Delivery Persons

46

AVG Delivery Time

22.67 min

AVG Delivery Rating

6.00 /6

Rating Group

Average (2.5 - 3.99)

Excellent (5.5 - 6)

Good (4 - 5.49)

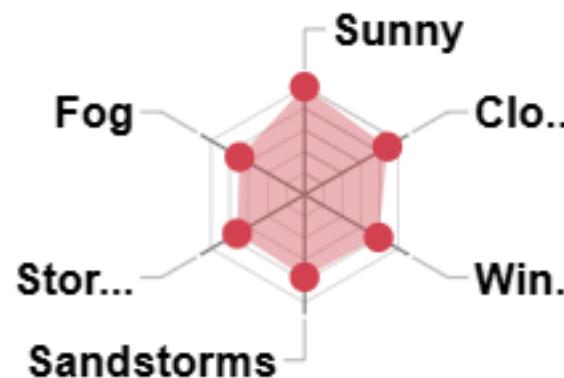
Poor (1 - 2.49)

Introduction

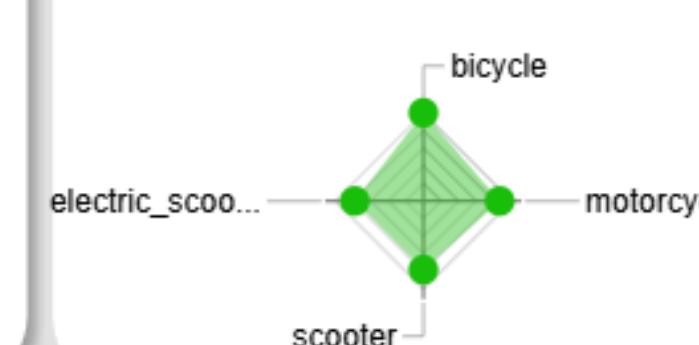
Overview

Delivery Analyst

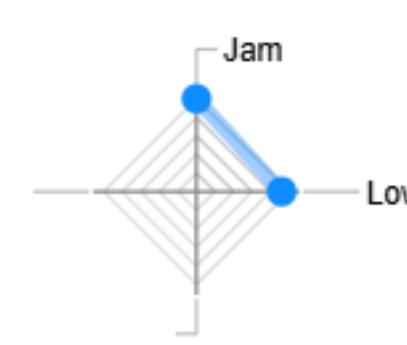
Weather conditions



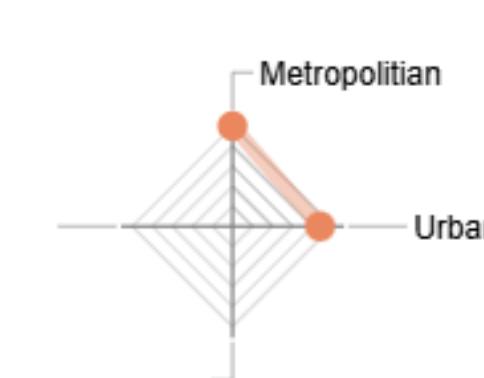
Vehicle Type



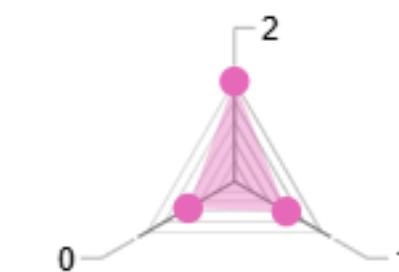
Road traffic density



Area Type



Multiple deliveries

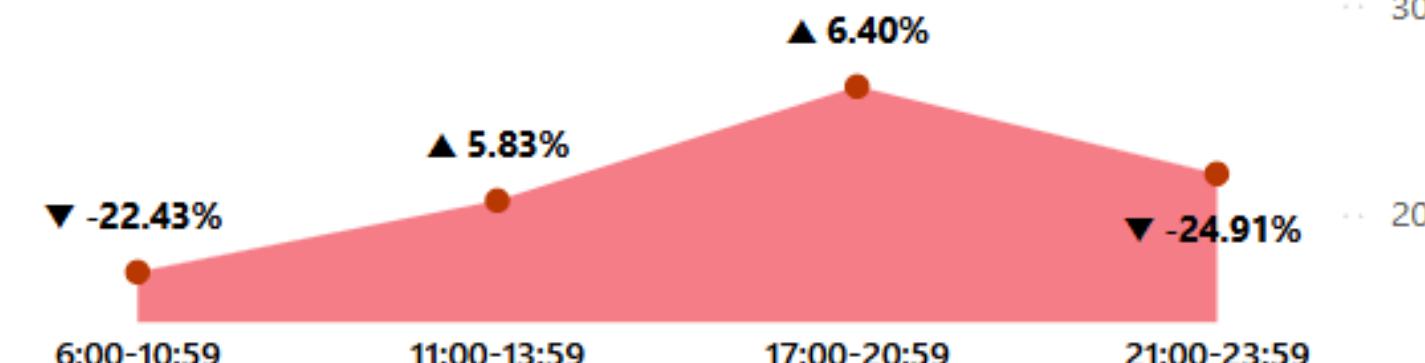


## Delivery Person Rating Group Details

Delivery_person_ID	Age	Orders Count	AVG Time Taken	Rating	Group
--------------------	-----	--------------	----------------	--------	-------

AGRRES13DEL02	29	1	20.00	6.00	Excellent (5.5-6)
AGRRES20DEL03	22	1	16.00	6.00	Excellent (5.5-6)
ALHRES01DEL02	30	1	33.00	6.00	Excellent (5.5-6)
ALHRES08DEL03	23	1	29.00	6.00	Excellent (5.5-6)
BANGRES010DEL01	39	1	17.00	6.00	Excellent (5.5-6)
BANGRES05DEL01	36	1	25.00	6.00	Excellent (5.5-6)
BANGRES05DEL03	31	1	19.00	6.00	Excellent (5.5-6)
BANGRES12DEL01	26	1	19.00	6.00	Excellent (5.5-6)
BANGRES13DEL01	26	1	28.00	6.00	Excellent (5.5-6)
BANGRES15DEL01	33	1	15.00	6.00	Excellent (5.5-6)
BANGRES19DEL01	23	1	18.00	6.00	Excellent (5.5-6)
BHPRES08DEL03	23	1	17.00	6.00	Excellent (5.5-6)
Total	27	46	22.67	6.00	

## AVG Delivery Time by Time Slot



## Driver Distribution by Rating Group vs AVG Delivery

● Count of Delivery\_person\_ID ● Average of Time\_taken(min)



## 6. Kết luận



**Nhu cầu thay đổi theo thời gian (tháng, ngày, khung giờ) như thế nào, và đâu là các thời điểm cao điểm? Làm sao để duy trì tăng trưởng ổn định và quản lý tải hiệu quả trong các thời điểm cao điểm?**

Nhu cầu không ổn định, với tháng 3/2022 là đỉnh cao, trong khi tháng 4 giảm mạnh do thiếu kế hoạch tiếp nối. Nhu cầu cao tập trung vào giữa và cuối tuần (Thứ Tư, Thứ Sáu) và khung giờ 17:00-20:59, phản ánh thói quen tiêu dùng (ăn tối, chuẩn bị cuối tuần), còn khung 0:00-5:59 có nhu cầu rất thấp.



**Thành phố nào có tiềm năng nhưng hiệu suất thấp, và làm thế nào để tăng thị phần tại đó?**

Delhi và Bhopal có tỷ lệ đơn hàng thấp bất thường, trong khi Jaipur và Bangalore cho thấy hiệu suất tốt.

=> Cần điều tra nguyên nhân, sau đó triển khai chiến dịch marketing nhắm mục tiêu

## 6. Kết luận



**Những yếu tố nào ảnh hưởng lớn đến thời gian giao hàng, và khung giờ nào có thời gian giao hàng chậm nhất? Làm sao để cải thiện và giảm thời gian giao hàng, đặc biệt trong khung giờ cao điểm?**

Thời tiết xấu, giao thông tắc nghẽn, khu vực bán đô thị và xe máy là các yếu tố chính gây chậm trễ. Khung 17:00-20:59 chậm nhất do áp lực từ số lượng đơn lớn và tắc nghẽn giao thông.



**Đánh giá và độ tuổi có ảnh hưởng đến hiệu quả giao hàng không? Làm sao để nâng cao chất lượng và quản lý đội ngũ tài xế dựa trên các yếu tố này?**

Đánh giá tài xế ảnh hưởng rõ rệt đến hiệu quả, nhóm Good và Excellent giao hàng nhanh hơn, trong khi nhóm Average làm tăng thời gian trung bình. Độ tuổi cũng có tác động, với nhóm Poor (lớn tuổi hơn, 30.3 tuổi) có hiệu suất thấp hơn, nhưng số lượng ít nên tác động tổng thể không lớn.

=> Cần đào tạo kỹ năng cho nhóm Average và Poor

# TRÍCH CHỌN ĐẶC TRƯNG

# Tạo các cột mới

**Mục đích:** Trích xuất thêm các đặc trưng từ dữ liệu thô nhằm mô tả rõ hơn hành vi giao hàng, cải thiện độ chính xác của mô hình học máy

- Trích xuất **giờ trong ngày** khi đơn hàng được đặt → **Order\_Hour**
- Trích xuất **giờ lấy hàng**, giúp biết thời điểm shipper bắt đầu đi giao → **Hour\_Pickup**
- Tính **thời gian chuẩn bị đơn hàng** (từ lúc đặt đến lúc lấy), đơn vị tính là phút → **Order\_Prepared\_Time**
- Trích xuất **thứ trong tuần** (0 = Thứ Hai, ..., 6 = Chủ Nhật) → **Day\_of\_Week**
- **Xác định đơn hàng được đặt vào cuối tuần** (Thứ 7, Chủ nhật) hay không → **weekend**
- **Phân loại ngày trong tháng** thành 3 giai đoạn: start\_month (từ ngày 1-10), middle\_month (từ ngày 11-20), end\_month (từ ngày 21 trở đi → **month\_intervals**

# Xóa các cột không cần thiết

Làm gọn DataFrame, loại bỏ dữ liệu không còn cần thiết để giảm kích thước và tập trung vào các đặc trưng quan trọng.

- - **ID, Delivery\_person\_ID**: Mã định danh không cần thiết cho phân tích.
- - **Order\_Date, Time\_Order\_picked\_formatted, Time\_Ordered\_formatted**: Các cột gốc đã được xử lý thành các cột mới.

# Mã hóa biến phân loại

Việc mã hóa đúng cách giúp mô hình học được các mối quan hệ thực sự giữa các đặc trưng và biến mục tiêu, từ đó nâng cao hiệu suất dự đoán.

**categorical\_columns** = "Weatherconditions", "Road\_traffic\_density", "Type\_of\_order",  
"Type\_of\_vehicle", "Festival", "Area\_Type", "City\_name", "month\_intervals"

Delivery_person_Age	Delivery_person_Ratings	Weatherconditions	Road_traffic_density	Vehicle_condition	Type_of_order	Type_of_vehicle	multiple_deliveries	Festival
0	37	4.9	4	0	2	3	2	0
1	21	4.5	3	1	2	3	3	1

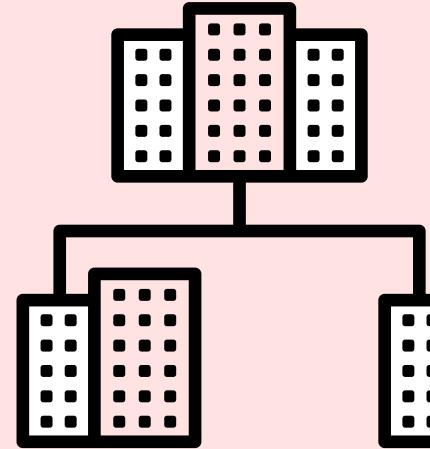
Area_Type	Time_taken(min)	City_name	Distance_km	Order_Hour	Hour_Pickup	Order_Prepares_Time	Day_of_Week	Weekend	month_intervals	Average_Delivery_Speed
2	24	10	4.599023	11	11	15.0	5	1	1	11.497558
0	33	3	19.995664	19	19	5.0	4	0	0	36.355753

# Chia tách dữ liệu



## Xác định các biến đầu vào và biến mục tiêu

- Tạo một biến X đại diện cho các biến đầu vào bằng cách loại bỏ cột **time\_taken (min)** khỏi bộ dữ liệu
- Tạo biến Y, chứa giá trị của biến mục tiêu là **time\_taken (min)**



## Chia dữ liệu thành tập huấn luyện và tập kiểm tra

- Tập huấn luyện (80% dữ liệu) để huấn luyện mô hình.
- Tập kiểm tra (20% dữ liệu) để đánh giá hiệu suất mô hình.

# Chuẩn hóa dữ liệu

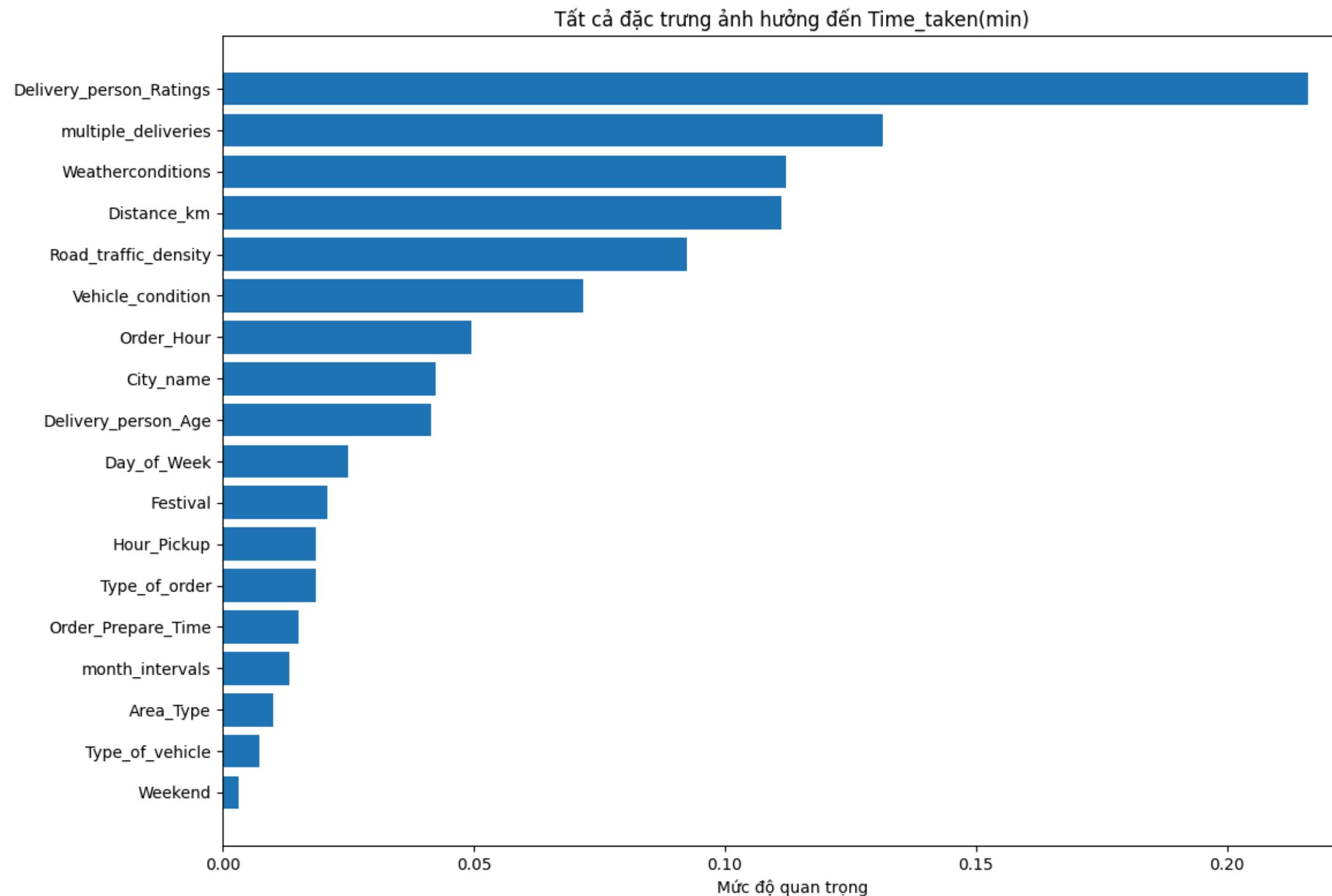
- Sử dụng phương pháp Standardization
- Chuẩn hóa đối với tất cả các cột trong tập X\_train

	Delivery_person_Age	Delivery_person_Ratings	Weatherconditions	Road_traffic_density	Vehicle_condition	Type_of_order	Type_of_vehicle	multiple_deliveries	Festival	Area_Type
0	37	4.9	4	0	2	3	2	0	0	2
1	21	4.5	3	1	2	3	3	1	0	0
2	23	4.4	2	2	0	1	2	1	0	2
3	24	4.7	4	3	0	0	2	1	0	0
4	20	4.6	0	0	1	3	3	1	0	0
...	...	...	...	...	...	...	...	...	...	...
45587	27	4.2	5	1	2	1	2	1	0	0
45588	27	4.8	5	0	1	2	2	0	0	0
45590	21	4.9	0	2	1	1	3	0	0	0
45591	21	4.7	0	0	0	3	2	1	0	0
45592	22	4.9	1	3	2	3	3	1	0	0

41901 rows × 20 columns

Time_taken(min)	City_name	Distance_km	Order_Hour	Hour_Pickup	Order_Prepares_Time	Day_of_Week	Weekend	month_intervals	Average_Delivery_Speed
24	10	4.599023	11	11	15.0	5	1	1	11.497558
33	3	19.995664	19	19	5.0	4	0	0	36.355753
26	3	7.998623	8	8	15.0	5	1	1	18.458362
21	6	3.401680	18	18	10.0	1	0	2	9.719086
30	5	5.813103	13	13	15.0	5	1	0	11.626207
...	...	...	...	...	...	...	...	...	...
33	19	16.600272	21	21	10.0	1	0	2	30.182313
32	11	1.513097	11	11	10.0	3	0	0	2.837057
16	5	1.133420	23	23	15.0	4	0	1	4.250324
26	6	2.199462	13	13	5.0	0	0	2	5.075682
36	19	10.358311	17	17	5.0	2	0	2	17.263851

# Feature selection



→ Ta nhận thấy **3 đặc trưng quan trọng nhất** ảnh hưởng đến thời gian giao hàng (time\_taken(min)) là **Delivery\_person\_Ratings, multiple\_deliveries** và **Weatherconditions**

# **HUẤN LUYỆN MÔ HÌNH**

# Phương trình hồi quy

$$Y = \beta_0 + \beta_1.X_1 + \beta_2.X_2 + \dots + \beta_{18}.X_{18} + U$$

- Y: Biến Time\_taken(min) cần dự báo
- $\beta_0$  Hệ số chẵn của mô hình
- $\beta_1, \beta_2, \beta_3, \dots, \beta_{18}$ : Các hệ số hồi quy, cho biết mức độ ảnh hưởng của từng biến độc lập lên thời gian giao hàng
- U: Sai số của mô hình - Đại diện cho các nhân tố tác động đến thời gian giao hàng nhưng không có trong mô hình

# Linear Regression

## Đánh giá mô hình Linear Regression

**MAE (Mean Absolute Error):** 5.71

Trung bình mỗi dự đoán của mô hình lệch khoảng 5.71 đơn vị so với giá trị thực tế.

**MSE (Mean Squared Error):** 50.53

Sai lệch bình phương trung bình giữa giá trị dự đoán và giá trị thực tế

**RMSE (Root Mean Squared Error):** 7.11

Mô hình có sai số trung bình khoảng 7.11 đơn vị so với giá trị thực tế.

**R<sup>2</sup> (R-Squared):** 0.4223

Mô hình chỉ giải thích được **42.23%** sự biến thiên của thời gian giao hàng

Kết quả đánh giá Linear Regression:

Mean Squared Error (MSE): 50.53

Root Mean Squared Error (RMSE): 7.11

Mean Absolute Error (MAE): 5.71

R<sup>2</sup> Score: 0.4223

# Decision Tree Regressor

## Khởi tạo mô hình Decision Tree Regressor

## Tìm kiếm siêu tham số tối ưu với GridSearchCV:

- Bộ siêu tham số tốt nhất tìm được như sau:
  - max\_depth = 10: Độ sâu tối đa của mỗi cây là 10
  - min\_samples\_leaf = 4 số mẫu tối thiểu ở một lá (leaf) là 4.
  - min\_samples\_split = 10: Số mẫu tối thiểu cần có để chia một nút thành các nút con là 10
  - max\_features = None: Sử dụng tất cả các đặc trưng khi tìm cách chia tại mỗi nút.
  - criterion = squared\_error: Hàm mất mát dùng để đánh giá độ chia là MSE - Bình phương trung bình cản sai số

## Huấn luyện mô hình Decision Tree với siêu tham số tối ưu:

- Sau khi tìm ra bộ siêu tham số tối ưu từ GridSearchCV, khởi tạo lại mô hình với các giá trị này và huấn luyện trên toàn bộ dữ liệu huấn luyện.

Kết quả đánh giá Decision Tree Regressor:

Mean Squared Error (MSE): 31.08

Root Mean Squared Error (RMSE): 5.57

Mean Absolute Error (MAE): 4.32

R<sup>2</sup> Score: 0.6447

# Decision Tree Regressor

## Đánh giá mô hình Decision Tree Regressor

**MAE (Mean Absolute Error):** 4.32

Trung bình mỗi dự đoán của mô hình lệch khoảng 4.32 đơn vị so với giá trị thực tế.

**MSE (Mean Squared Error):** 31.08

Sai lệch bình phương trung bình giữa giá trị dự đoán và giá trị thực tế

**RMSE (Root Mean Squared Error):** 5.57

Mô hình có sai số trung bình khoảng 5.57 đơn vị so với giá trị thực tế.

**R<sup>2</sup> (R-Squared):** 0.6447

Mô hình chỉ giải thích được **64.47 %** sự biến thiên của thời gian giao hàng

Kết quả đánh giá Decision Tree Regressor:

Mean Squared Error (MSE): 31.08

Root Mean Squared Error (RMSE): 5.57

Mean Absolute Error (MAE): 4.32

R<sup>2</sup> Score: 0.6447

# Random Forest Regressor

## Khởi tạo mô hình Random Forest Regressor

## Tìm kiếm siêu tham số tối ưu với GridSearchCV:

- Bộ siêu tham số tốt nhất tìm được như sau:
- max\_depth = 10: Độ sâu tối đa của mỗi cây là 10
- min\_samples\_leaf = 1 số mẫu tối thiểu ở một lá (leaf) là 1.
- min\_samples\_split = 10: Số mẫu tối thiểu cần có để chia một nút thành các nút con là 10
- n\_estimators = 300: mô hình sẽ tạo ra 300 cây quyết định.

## Huấn luyện mô hình Random Forest với siêu tham số tối ưu:

- Sau khi tìm ra bộ siêu tham số tối ưu từ GridSearchCV, khởi tạo lại mô hình với các giá trị này và huấn luyện trên toàn bộ dữ liệu huấn luyện.

Kết quả đánh giá Random Forest Regressor:

Mean Squared Error (MSE): 29.61

Root Mean Squared Error (RMSE): 5.44

Mean Absolute Error (MAE): 4.24

R<sup>2</sup> Score: 0.6614

# Random Forest Regressor

## Đánh giá mô hình Random Forest Regressor

**MAE (Mean Absolute Error):** 4.24

Trung bình mỗi dự đoán của mô hình lệch khoảng 4.24 đơn vị so với giá trị thực tế.

**MSE (Mean Squared Error):** 29.61

Sai lệch bình phương trung bình giữa giá trị dự đoán và giá trị thực tế

**RMSE (Root Mean Squared Error):** 5.44

Mô hình có sai số trung bình khoảng 1.04 đơn vị so với giá trị thực tế.

**R<sup>2</sup> (R-Squared):** 0.6614

Mô hình chỉ giải thích được **66.14%** sự biến thiên của thời gian giao hàng

**Kết quả đánh giá Random Forest Regressor:**

**Mean Squared Error (MSE):** 29.61

**Root Mean Squared Error (RMSE):** 5.44

**Mean Absolute Error (MAE):** 4.24

**R<sup>2</sup> Score:** 0.6614

# XGBoost

## Khởi tạo mô hình XGBoost

### Tìm kiếm siêu tham số tối ưu với GridSearchCV:

- Bộ siêu tham số tốt nhất tìm được như sau:
- colsample\_bytree = 1.0
- learning\_rate = 0.1
- max\_depth = 7
- n\_estimators = 100
- subsample = 1.0

### Huấn luyện mô hình XGBoost với siêu tham số tối ưu:

- Sau khi tìm ra bộ siêu tham số tối ưu từ GridSearchCV, khởi tạo lại mô hình với các giá trị này và huấn luyện trên toàn bộ dữ liệu huấn luyện.

Kết quả đánh giá XGBoost Regressor:

Mean Squared Error (MSE): 28.24

Root Mean Squared Error (RMSE): 5.31

Mean Absolute Error (MAE): 4.18

R<sup>2</sup> Score: 0.6771

# XGBoost

## Đánh giá mô hình XGBoost

**MAE (Mean Absolute Error):** 4.18

Trung bình mỗi dự đoán của mô hình lệch khoảng 4.18 đơn vị so với giá trị thực tế.

**MSE (Mean Squared Error):** 28.24

Sai lệch bình phương trung bình giữa giá trị dự đoán và giá trị thực tế

**RMSE (Root Mean Squared Error):** 5.31

Mô hình có sai số trung bình khoảng 5.31 đơn vị so với giá trị thực tế.

**R<sup>2</sup> (R-Squared):** 0.6671

Mô hình chỉ giải thích được **67.71%** sự biến thiên của thời gian giao hàng

Kết quả đánh giá XGboost Regressor:

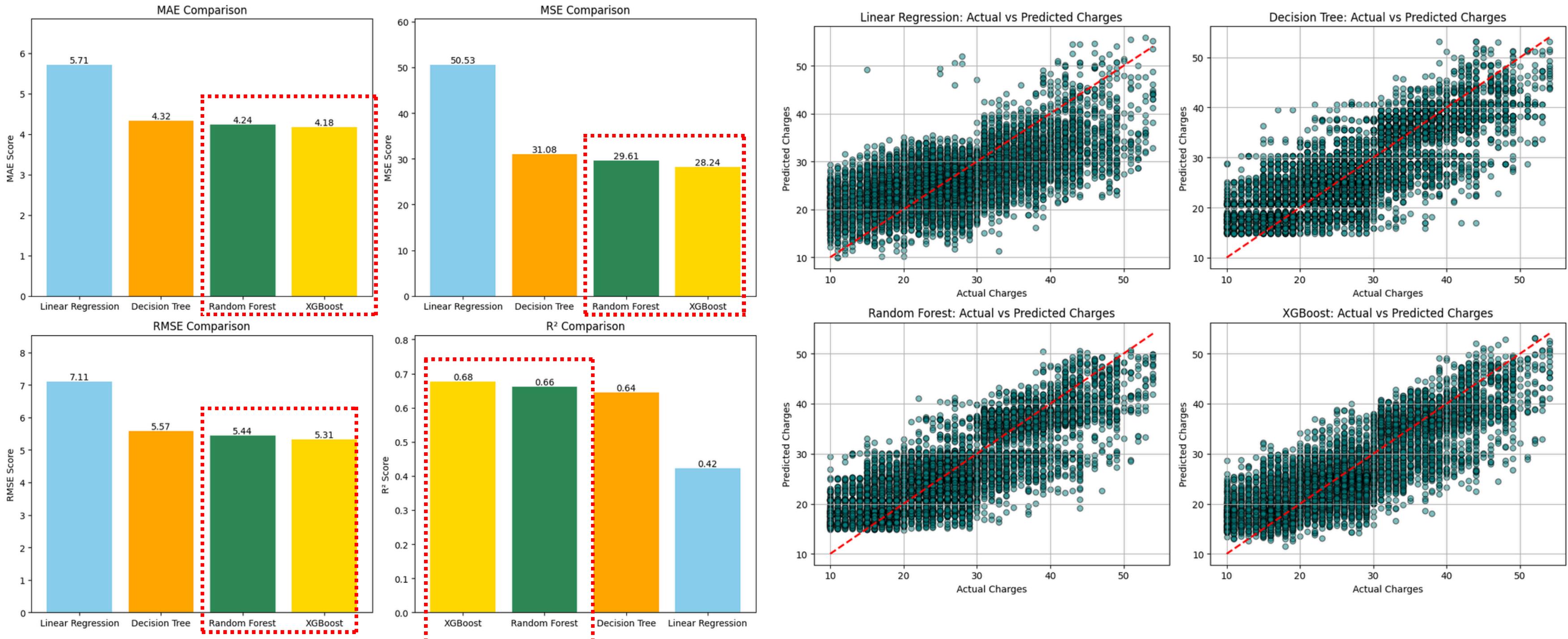
Mean Squared Error (MSE): 28.24

Root Mean Squared Error (RMSE): 5.31

Mean Absolute Error (MAE): 4.18

R<sup>2</sup> Score: 0.6771

# **SO SÁNH CÁC MÔ HÌNH**



- Mô hình XGBoost đạt hiệu suất tốt nhất trong việc dự đoán chi phí thời gian giao hàng, với MAE = 4.18, MSE = 28.24, RMSE = 5.31 và R<sup>2</sup> = 0.68. Các mô hình khác như Random Forest, Decision Tree và Linear Regression lần lượt giảm dần hiệu suất, với Linear Regression có kết quả kém nhất. Biểu đồ Actual vs Predicted cho thấy XGBoost có khả năng dự đoán chính xác nhất, trong khi Linear Regression và Decision Tree có xu hướng không bắt kịp các mẫu phức tạp trong dữ liệu.

# Kết luận

Dựa vào biểu đồ **feature selection** trước đó, Ta nhận thấy 3 đặc trưng quan trọng nhất ảnh hưởng đến thời gian giao hàng (time\_taken(min)) là **Delivery\_person\_Ratings**, **multiple\_deliveries** và **Weatherconditions**

- Delivery\_person\_Ratings (0.216071): Điểm đánh giá của tài xế là yếu tố có ảnh hưởng lớn nhất, chiếm 21.61% tầm quan trọng. Điều này cho thấy chất lượng và hiệu suất của tài xế (dựa trên đánh giá từ khách hàng) là yếu tố quyết định chính đến thời gian giao hàng, phản ánh mối quan hệ trực tiếp giữa kỹ năng tài xế và hiệu quả vận hành.
- Multiple\_deliveries (0.131402): Số lượng đơn giao cùng lúc ảnh hưởng 13.14%, cho thấy việc xử lý nhiều đơn hàng cùng một lúc làm tăng độ phức tạp và thời gian giao, đặc biệt trong các khung giờ cao điểm.
- Thời tiết ảnh hưởng 11.22% đến thời gian giao hàng, đứng thứ ba trong các đặc trưng.

# Giải pháp

- **Nâng cao Đánh giá Tài xế (Delivery\_person\_Ratings):**

- Khuyến khích hiệu suất: Thưởng cho tài xế có đánh giá cao hoặc cải thiện đánh giá theo thời gian, tạo động lực để họ giao hàng nhanh và chất lượng hơn.
- Hỗ trợ tài xế mới: Cung cấp hướng dẫn và giám sát cho các tài xế có đánh giá thấp để cải thiện kỹ năng, từ đó nâng cao tổng thể hiệu suất đội ngũ.
- Thu thập phản hồi: Khuyến khích khách hàng đánh giá thường xuyên để có dữ liệu chính xác, từ đó hỗ trợ tài xế cải thiện.

- **Đối với các đơn hàng ghép (multiple\_deliveries):**

- Giới hạn Số lượng Đơn Đồng thời trong Giờ Cao điểm: Trong khung 17:00-20:59, giới hạn số đơn giao cùng lúc (ví dụ: tối đa 2-3 đơn/tài xế) và tăng số lượng tài xế để tránh áp lực. Đồng thời, sử dụng công nghệ dự đoán tải để phân bổ đơn hàng hiệu quả hơn, đảm bảo không quá tải cho một tài xế.

- **Kế hoạch Dự phòng cho Thời tiết Xấu:**

- Tăng số lượng tài xế trong các ngày thời tiết xấu (sương mù, mây) để giảm tải cho mỗi tài xế, đồng thời điều chỉnh lịch trình giao hàng (ví dụ: ưu tiên đơn hàng gần trong điều kiện tầm nhìn thấp). Sử dụng dữ liệu dự báo thời tiết để chuẩn bị trước, đảm bảo không bị gián đoạn vận hành.

# DỰ ĐOÁN GIÁ TRỊ MỚI

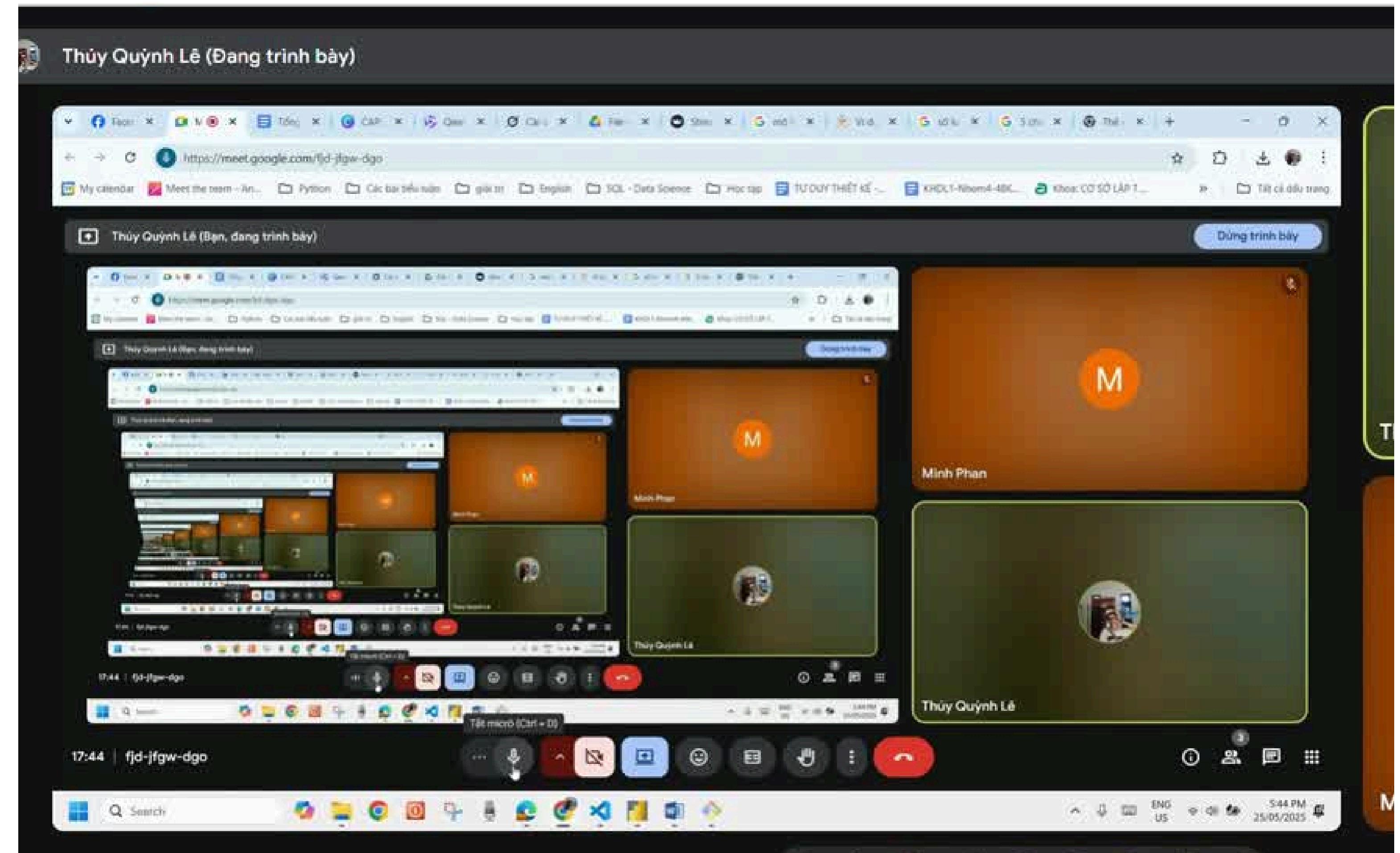
# Dự đoán

'Delivery_person_Age':	[37, 21],
'Delivery_person_Ratings'	[ <b>4.9</b> , 4.5],
'Weatherconditions'	[ <b>4</b> , 3],
'Road_traffic_density'	[0, 1],
'Vehicle_condition'	[2, 2],
'Type_of_order'	[3, 3],
'Type_of_vehicle'	[2, 3],
'multiple_deliveries'	[ <b>0</b> , 1],
'Festival'	[0, 0],
'Area_Type'	[2, 0],
'City_name'	[10, 3],
'Distance_km'	[4.599, 19.996],
'Order_Hour'	[11, 19],
'Hour_Pickup'	[11, 19],
'Order_Prepares_Time'	[15.0, 5.0],
'Day_of_Week'	[5, 4],
'Weekend'	[1, 0],
'month_intervals'	[1, 0],

Dự đoán thời gian giao hàng mẫu 1: 15.16 phút

Dự đoán thời gian giao hàng mẫu 2: 26.41 phút

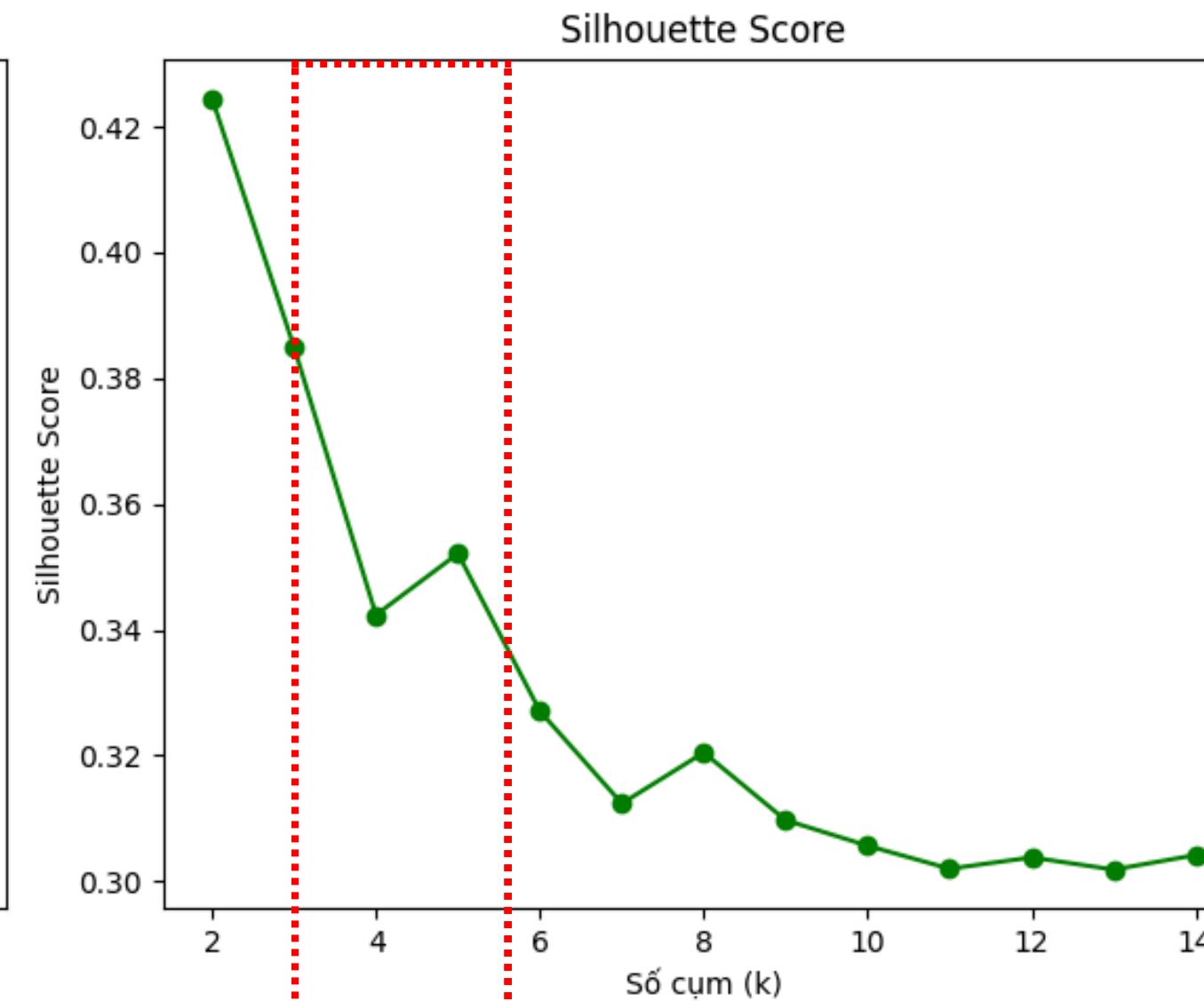
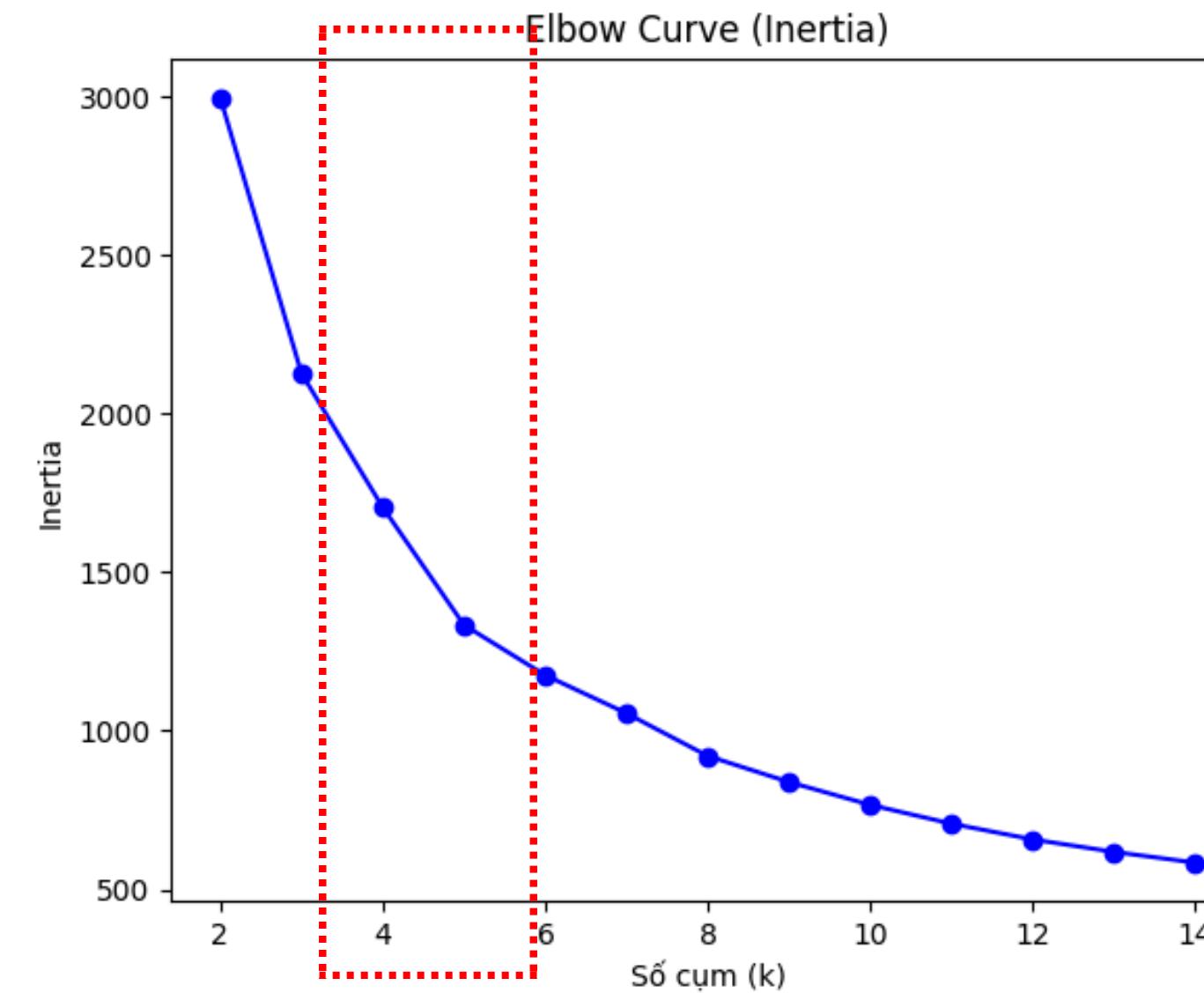
# LƯU VÀ TRIỂN KHAI MÔ HÌNH



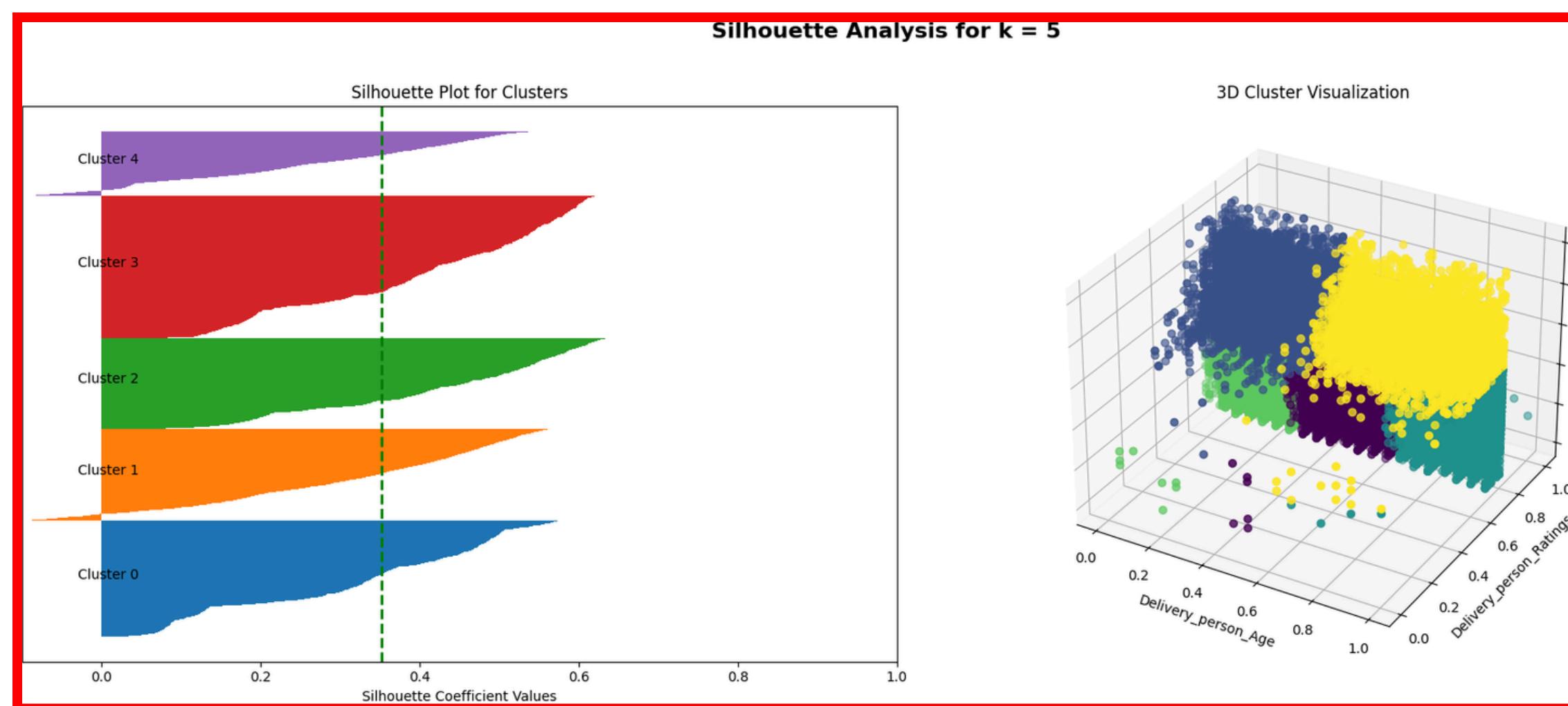
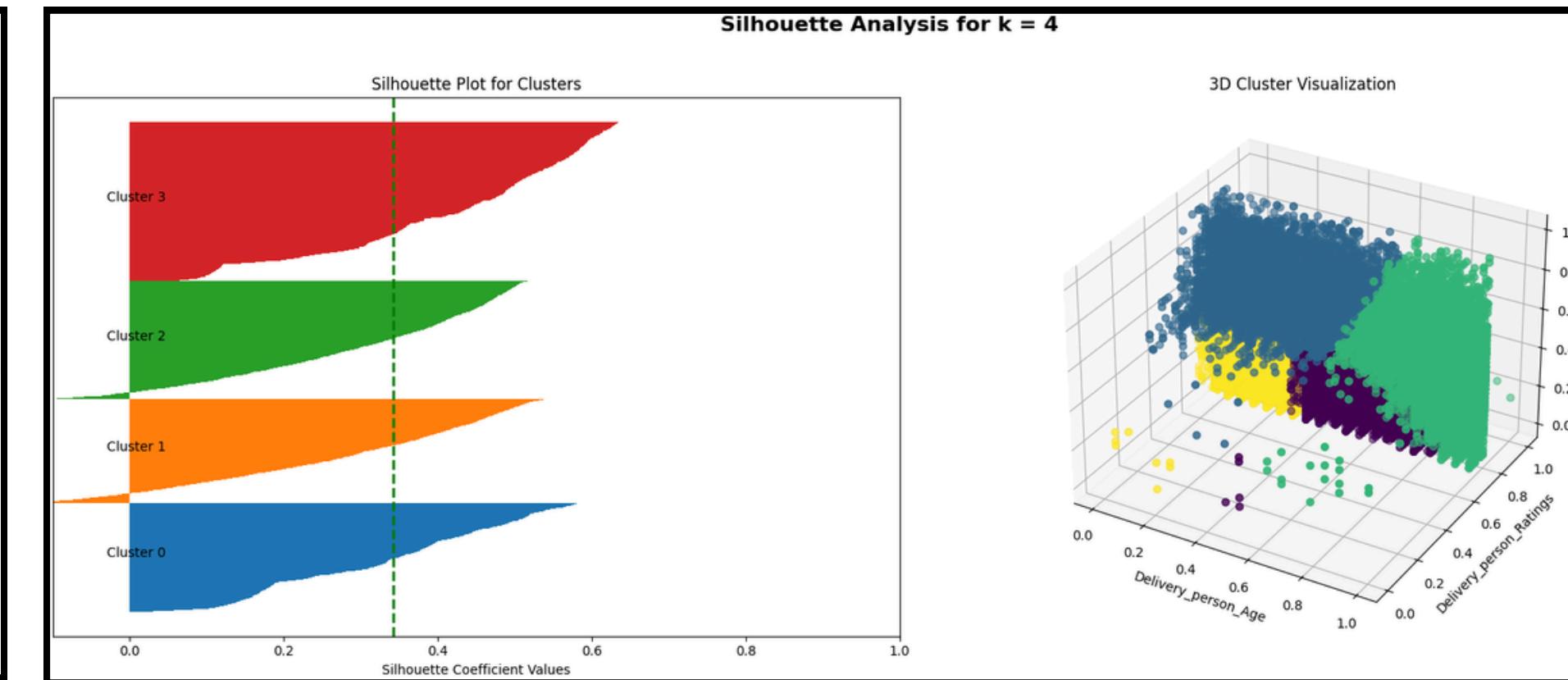
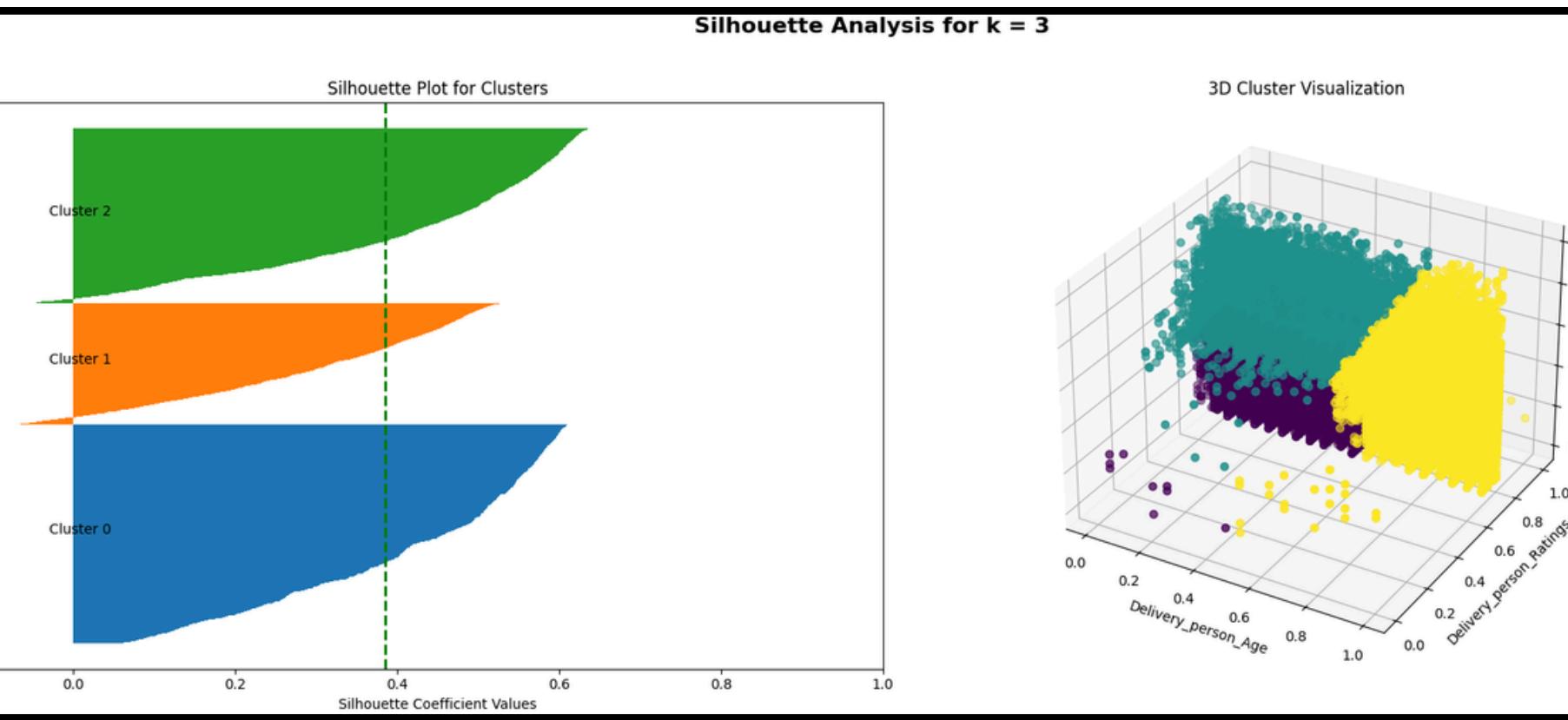
# **PHÂN CỤM TÀI XẾ**

# Chọn số cụm

Lựa chọn các đặc trưng Delivery\_person\_Age, Delivery\_person\_Ratings, và Time\_taken(min) để thực hiện phân cụm tài xế

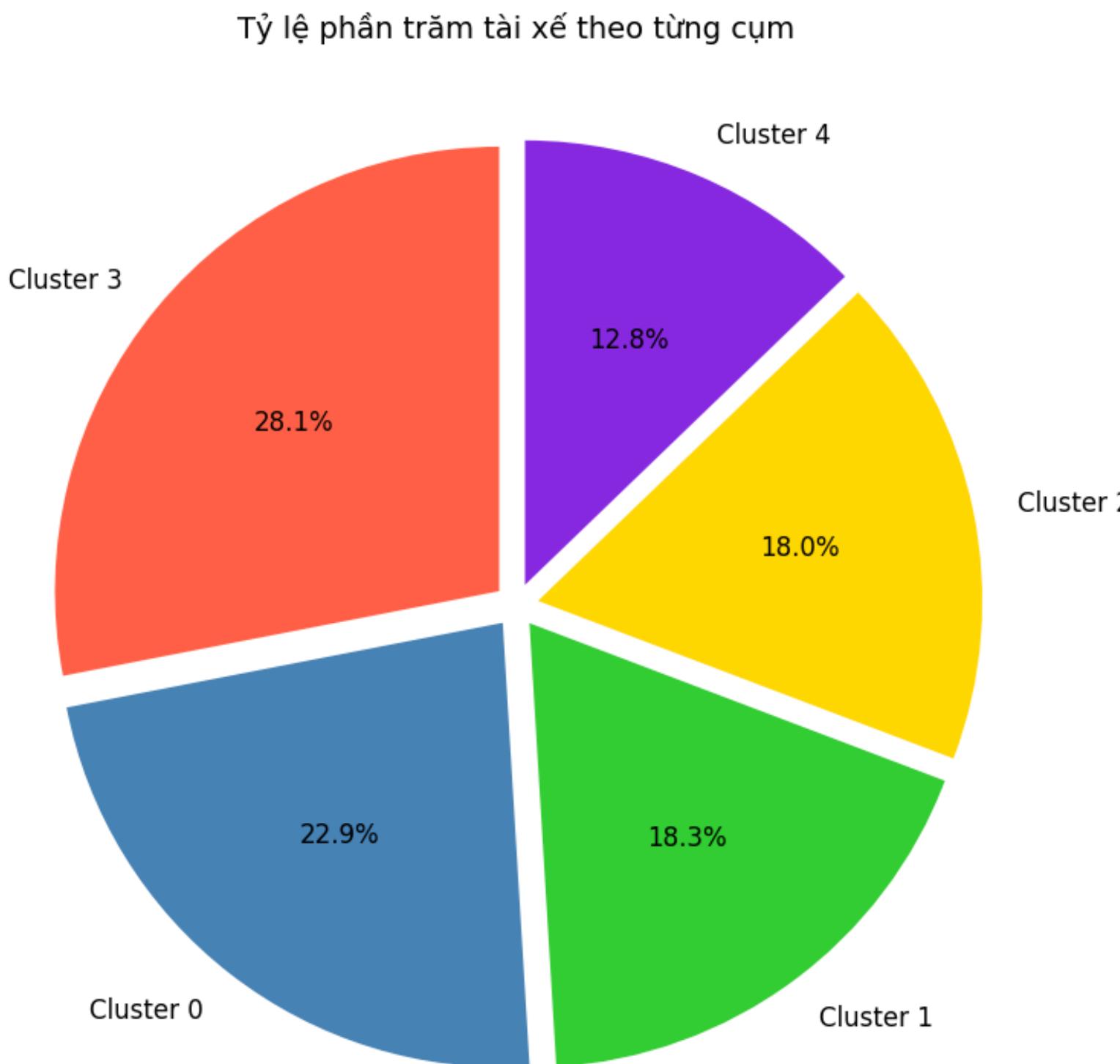


# Chọn số cụm



**WSS: 1333.852**  
**BSS: 7224.224**  
**Silhouette Score: 0.352**  
**Calinski Harabasz Score: 34834.378**

# Chọn số cụm

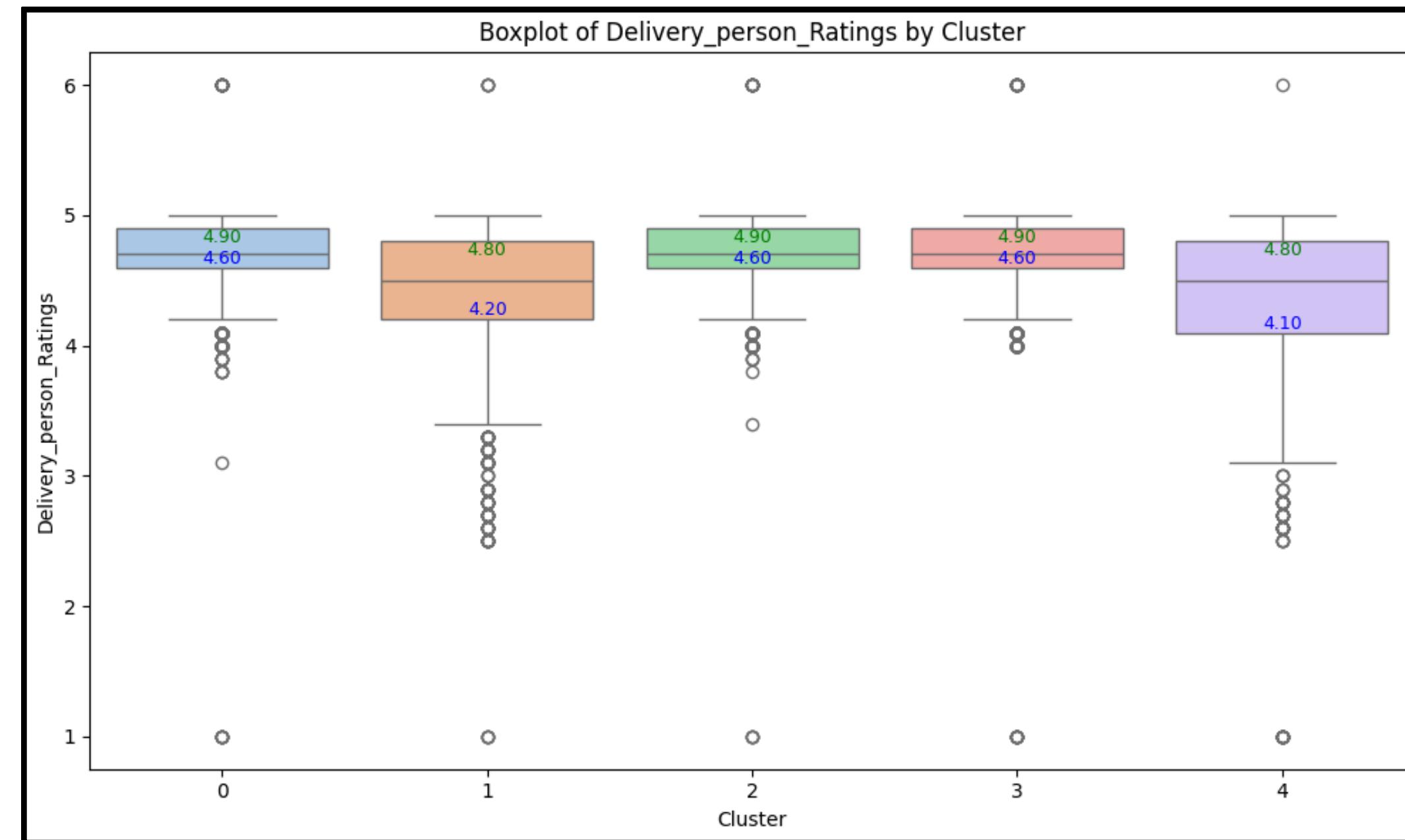


# Phân bố đặc trưng của các cụm



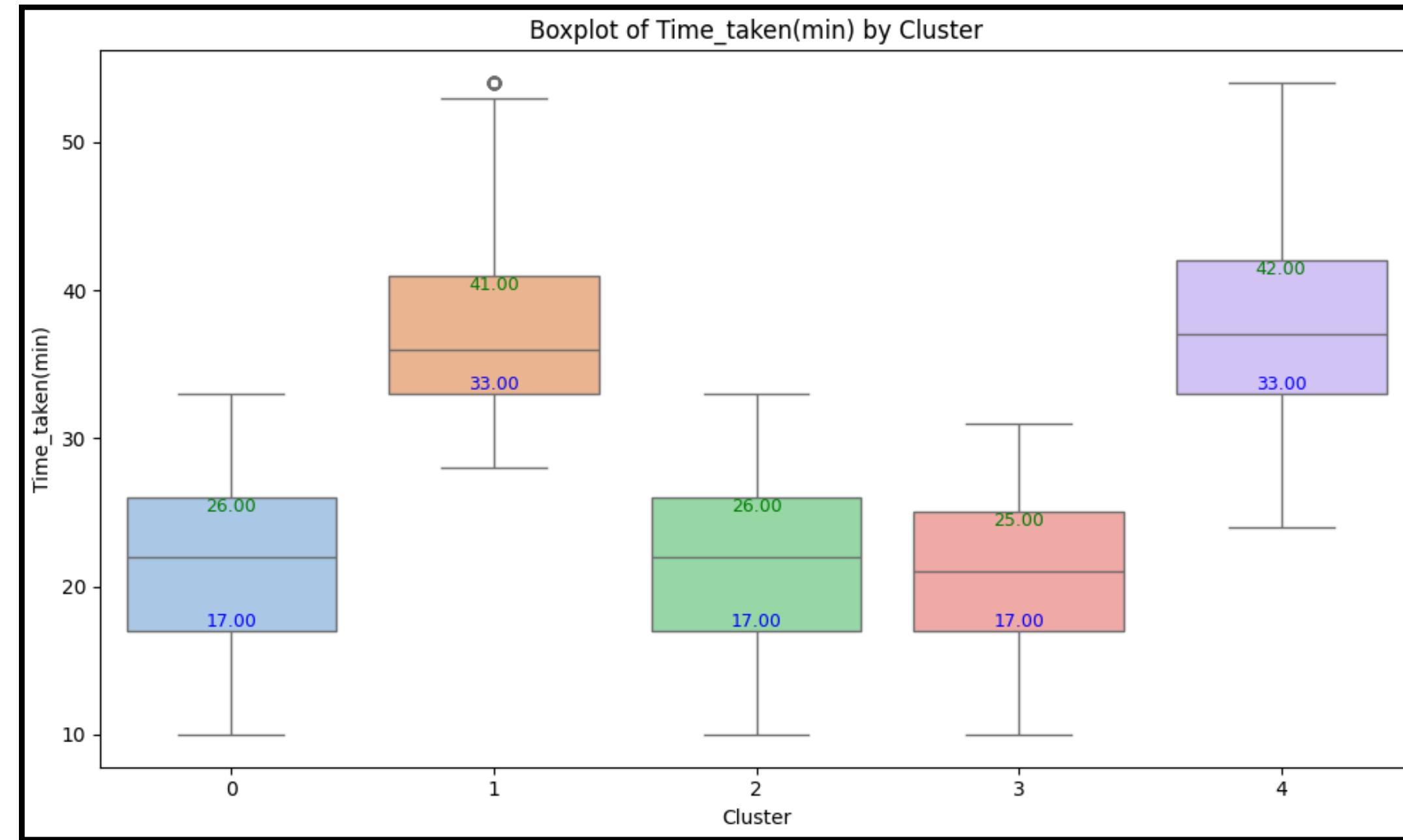
- 
- Cụm 2 là cụm có độ tuổi trung bình cao nhất, đại diện cho nhóm tài xế lớn tuổi hơn.
  - Cụm 1 và cụm 3 là hai cụm có độ tuổi trung vị thấp nhất, tập trung nhiều tài xế trẻ tuổi.
  - Cụm 0 và cụm 4 có độ tuổi trung vị tương đối cao, nhưng không cao bằng cụm 2.

# Phân bố đặc trưng của các cụm



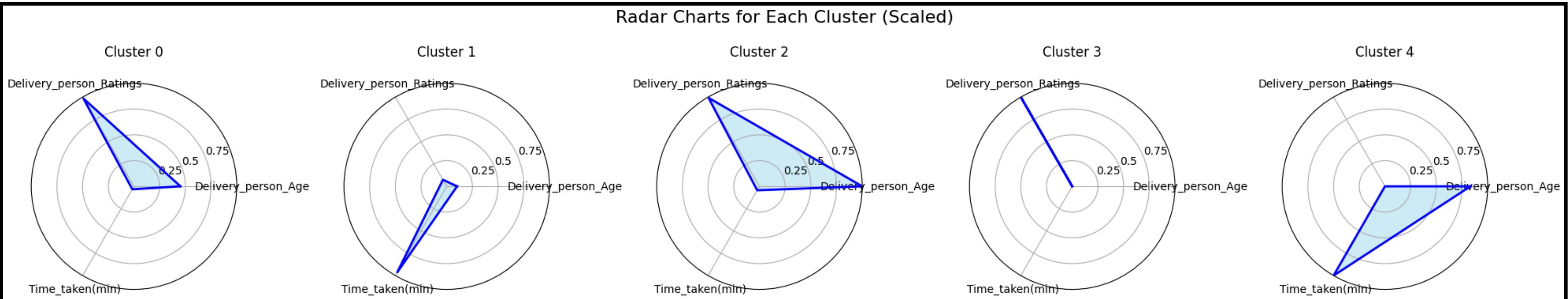
- 
- Cụm tốt (cao và ổn định): Cluster 0, 2, 3 => Tài xế cụm này được đánh giá khá cao và ổn định, dù vẫn có một số đánh giá rất thấp.
  - Cụm cần cải thiện: Cluster 1, 4 — do median thấp và nhiều đánh giá tệ (outliers) => Nhóm này cũng có vấn đề về chất lượng phục vụ. Có thể đây là nhóm tài xế mới hoặc thiếu kinh nghiệm.

## Phân bố đặc trưng của các cụm

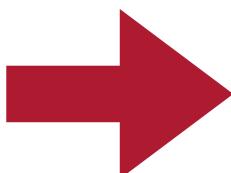


- 
- Cụm tốt (cao và ổn định): Cluster 0, 2, 3 => thời gian giao hàng nhanh và ổn định. Đây là những cụm hiệu quả nhất về thời gian.
  - Cụm cần cải thiện: Cluster 1, 4 => Thời gian giao hàng lâu nhất và biến động lớn. Đây là cụm kém hiệu quả nhất về tốc độ giao hàng.

# Kết luận và đề xuất



- **Cụm 2:** Đây là nhóm tài xế lớn tuổi và rất tốt – họ giao hàng nhanh và được đánh giá cao. Có thể là nhóm giàu kinh nghiệm, quen tuyến đường, cẩn thận và đáng tin cậy.
- Cụm 1 vs 3 (cùng là tài xế trẻ):
  - **Cụm 3** làm việc hiệu quả cao (giống như cụm 2).
  - **Cụm 1** làm việc kém hiệu quả nhất. Cần xem lại đào tạo hoặc các yếu tố khác như thái độ, thời gian làm việc.
- **Cụm 4:** Dù có độ tuổi khá cao, nhưng hiệu suất kém cả về thời gian lẫn rating. Có thể là nhóm tài xế lớn tuổi hơn nhưng chưa quen công nghệ/giao diện mới, hoặc làm việc ở khu vực khó giao hàng hơn.
- **Cụm 0:** Là cụm “cân bằng” – tuổi trung bình cao, giao hàng nhanh, rating tốt



## Hành động:

- Giữ lại và phát triển: Cluster 0, 2, 3
- Đào tạo lại hoặc giám sát kỹ: Cluster 1 (trẻ nhưng kém), Cluster 4 (lớn tuổi nhưng hiệu quả thấp)

# Thank You