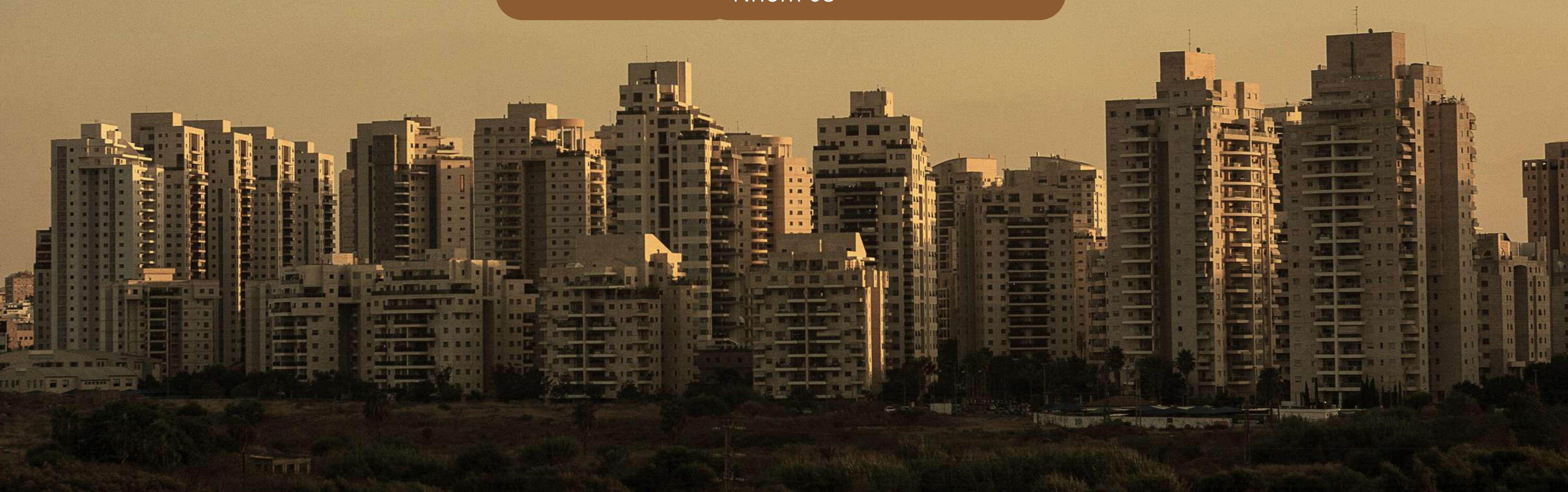


# Báo cáo đồ án

## Phân tích và phân loại chất lượng không khí

Nhóm 08





# Thông tin thành viên

1

**21120091\_Hồ Sỹ Kiên**

Nhóm trưởng

2

**21120099\_Hoàng Thành  
Nam**

Thành viên

3

**21120035\_Nguyễn Hoài An**

Thành viên

4

**21120176\_Đinh Thị Thúy  
Hường**

Thành viên

# Work flow



## Pha 01

Thu thập dữ liệu



## Pha 02

Khám phá dữ liệu



## Pha 03

Đặt câu hỏi và  
trả lời









































## Pha 04

Mô hình hóa dữ  
liệu

# Phân công nhiệm vụ

## Tasks

 All tasks  Board +

	Aa Task name	 Assignee	 Due
	 Data Collecting	 Kiên Hồ Sỹ	December 10, 2023 → December 13, 2023
	 First Explore (Meaning, Shape)	 Kiên Hồ Sỹ	December 18, 2023
	 Explore Cont.(Handle Data type, Missing)	 Hoài An Nguyễn	December 18, 2023
	Explore Cont.(Numeric Distribution)	 Nam	December 18, 2023
	Explore Cont.(Non-numeric Distribution)	 Hường Thúy	December 18, 2023
	Ask Question (2)	 Kiên Hồ Sỹ	December 26, 2023
	Ask Question (1)	 Hoài An Nguyễn	December 26, 2023
	 Ask Question (1)	 Nam	December 26, 2023
	Ask Question (1)	 Hường Thúy	December 26, 2023
	Model Training (Decision Tree)	 Kiên Hồ Sỹ	January 2, 2024
	Model Training (SVM)	 Hường Thúy	January 2, 2024
	Model Training (KNN)	 Nam	January 2, 2024
	Model Training (MLP)	 Hoài An Nguyễn	January 2, 2024
	Edit report slide	 Kiên Hồ Sỹ  Hường Thúy  Hoài An Nguyễn  Nam	January 3, 2024



# Giới thiệu đề tài và lý do chọn đề tài



- Trong những năm gần đây, ô nhiễm không khí đang trở thành một vấn đề môi trường nghiêm trọng trên toàn cầu. Theo báo cáo của WHO năm 2022, ô nhiễm không khí là nguyên nhân gây tử vong cao thứ tư trên thế giới, chỉ đứng sau cao huyết áp, suy dinh dưỡng và hút thuốc lá-> gây ra khoảng 7 triệu ca tử vong sớm mỗi năm, trong đó 6,6 triệu ca tử vong xảy ra ở các nước đang phát triển.
- Tình trạng ô nhiễm không khí ở Việt Nam cũng đang ở mức báo động. Theo báo cáo của Bộ Tài nguyên và Môi trường Việt Nam năm 2022, 80% các thành phố lớn ở Việt Nam có chất lượng không khí không đáp ứng các tiêu chuẩn của WHO.



- ➔ Vì vậy phân tích chất lượng không khí là một công cụ quan trọng để theo dõi xu hướng ô nhiễm không khí, xác định các nguồn gây ô nhiễm và đánh giá tác động của ô nhiễm không khí đối với sức khỏe và môi trường.  
=> là lí do nhóm quyết định thu thập dữ liệu về chất lượng không khí ở các thành phố lớn, cụ thể là thành phố Hồ Chí Minh, để có những cái nhìn về chất lượng không khí nơi đó cũng như tại các thành phố lớn trên thế giới.

# MỤC TIÊU

Có nhiều thang đo AQI tùy theo quốc gia và vùng lãnh thổ. Ví dụ:

- Mỹ thang đo 6 bậc.
- Anh thang đo 4 bậc.
- Trong đó, nguồn dữ liệu nhóm sử dụng OpenWeatherMap sử dụng thang đo 5 *bậc* của EU.



Từ các mẫu không khí đã được chấm điểm mà nhóm thu thập được, tìm ra hàm chấm điểm cho các mẫu không khí để phục vụ cho các mục đích sau này (VD: Xây dựng một ứng dụng chấm điểm không khí, đánh giá chất lượng không khí khi có thêm mẫu mới được thu thập).

# Pha 01.

## Thu thập dữ liệu

**Chủ đề:** Nồng độ các chất gây ô nhiễm trong không khí

**Nguồn:** API từ trang web  
<https://openweathermap.org>

**Thư viện hỗ trợ:** request và json

- Mẫu API hỗ trợ:

[http://api.openweathermap.org/data/2.5/air\\_pollution/history?lat={lat}&lon={lon}&start={start}&end={end}&appid={API key}](http://api.openweathermap.org/data/2.5/air_pollution/history?lat={lat}&lon={lon}&start={start}&end={end}&appid={API key})

- Trong đó:
  - **{lat}**: Lattitude, vĩ độ điểm lấy mẫu.
  - **{lon}**: Longitude, kinh độ điểm lấy mẫu.
  - **{start}** : Thời điểm bắt đầu lấy mẫu ở dạng timestamp.
  - **{end}** : Thời điểm bắt đầu lấy mẫu ở dạng timestamp.
  - **{API key}** : key được cấp cho mỗi tài khoản khi đăng ký, thể hiện quyền được request tới API.



```
{
  "main": {
    "aqi": 2
  },
  "components": {
    "co": 460.63,
    "no": 0,
    "no2": 15.08,
    "o3": 25.03,
    "so2": 7.87,
    "pm2_5": 11.18,
    "pm10": 15.02,
    "nh3": 8.74
  },
  "dt": 1610128800
},
```



# Quy trình thu thập dữ liệu

1. Thiết kế các params theo yêu cầu của API
2. Sử dụng hàm `load_data()` với tham số là các params đã thiết kế để lấy dữ liệu từ API và lưu dưới dạng dataframe
3. Lưu dataframe dưới dạng file csv

## Hàm `load_data()`

- Sử dụng thư viện để load data do API cung cấp từ link theo các params trên
- Làm phẳng dữ liệu và lưu dưới dạng data frame

	B	C	D	E	F	G	H	I
	aqi	co	no	no2	o3	so2	pm2_5	pm10
09	3	700.95	0.44	35.99	17.35	32.9	20.33	26.6
09	3	847.82	2.46	38.04	18.06	36.24	23.32	30.5
09	3	894.55	5.25	38.39	23.25	41.01	24.16	31.9
09	3	827.79	6.2	36.33	33.98	43.39	23.2	30.9
09	2	660.9	3.69	29.13	54.36	35.76	19.5	25.
09	3	614.17	2.77	27.76	67.95	34.81	20.08	25.6
		100.82	2.38	26.05	80.82	34.33	21.1	26.2
		4.09	1.9	22.28	91.55	28.85	20.34	25.0
		4.03	1.17	20.74	85.83	24.32	17.08	21.1
		0.68	0.66	19.19	75.1	20.98	14.04	17.4
		0.71	0.14	19.88	64.37	20.27	12.91	15.9
		0.76	0.01	21.25	55.79	20.98	13.44	16.6
		4.09	0	20.74	50.78	20.74	14.14	17.7
		0.76	0	21.25	46.49	21.7	15.2	19.4



# Pha 02. Khám phá dữ liệu

Có những cái nhìn tổng quan nhất về dữ liệu  
thu thập được và tiền xử lý dữ liệu

Tiếp tục



# Nội dung thực hiện

01

Đọc dữ liệu sau khi thu thập được và tính số dòng và cột

02

Dữ liệu các dòng có bị lặp không?

03

Khám phá ý nghĩa của mỗi dòng và cột của dữ liệu

04

Khám phá và chuyển đổi kiểu dữ liệu của các cột

05

Xử lý dữ liệu thiếu (nếu có)  
Kiểm tra tính hợp lệ của dữ liệu

06

Xem xét sự phân bố giá trị của các cột dữ liệu dạng số

07

Xem xét sự phân bố giá trị của các cột dữ liệu không phải dạng số

# Rows

**25177**

**Dòng dữ liệu**

Dữ liệu được thu thập là dữ liệu về chất lượng không khí từ 01/01/2021 tới 01/12/2023

Mỗi dòng là dữ liệu theo giờ của từng ngày  
Không có dòng nào bị lặp

## Tổng quan về dữ liệu

# Columns

**10**

**Cột dữ liệu**

Bao gồm các cột dữ liệu về thời gian ghi nhận, chỉ số aqi (đo chất lượng không khí) và nồng độ các chất gây ô nhiễm trong không khí

Các cột là các chất gây ô nhiễm bao gồm:

- CO, O3, NO, NO2, SO2, NH3, PM2.5, PM10

Chỉ số chất lượng không khí là AQI, trong đó:

- 1 = Tốt
- 2 = Khá
- 3 = Trung bình
- 4 = Kém
- 5 = Rất kém

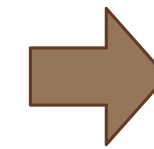


# Chuyển đổi kiểu dữ liệu

- Cột 'dt' là thời gian ghi nhận số liệu, vì vậy ta có thể chuyển đổi cột 'dt' từ kiểu *int64* sang kiểu *datetime*.
- Cột 'aqi' là chỉ số chất lượng không khí có giá trị từ 1-5 nên ta có thể chuyển cột 'aqi' từ kiểu *int64* sang kiểu *category*.

## Kiểu dữ liệu ban đầu của từng cột

dt	int64
aqi	int64
co	float64
no	float64
no2	float64
o3	float64
so2	float64
pm2_5	float64
pm10	float64
nh3	float64
dtype:	object



## Kiểu dữ liệu sau khi chuyển đổi

dt	datetime64[ns]
aqi	category
co	float64
no	float64
no2	float64
o3	float64
so2	float64
pm2_5	float64
pm10	float64
nh3	float64
dtype:	object

# Dữ liệu có bị thiếu ?

Không có dữ liệu bị thiếu phải xử lý



## Dữ liệu có hợp lệ ?

Với những cột chỉ số các chất gây ô nhiễm (CO, O3, NO, NO2, SO2, NH3, PM2.5, PM10) có kiểu dữ liệu numeric, giá trị của chúng phải lớn hơn hoặc bằng 0.

```
Không có giá trị bé hơn 0 trong cột 'co'  
Không có giá trị bé hơn 0 trong cột 'no'  
Không có giá trị bé hơn 0 trong cột 'no2'  
Có giá trị bé hơn 0 trong cột 'o3'  
Không có giá trị bé hơn 0 trong cột 'so2'  
Không có giá trị bé hơn 0 trong cột 'pm2_5'  
Không có giá trị bé hơn 0 trong cột 'pm10'  
Không có giá trị bé hơn 0 trong cột 'nh3'
```

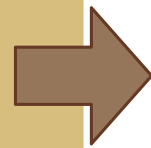
Có giá trị bé hơn 0 ở cột 'o3', có thể xóa dòng đó ra khỏi data.

**Sau khi xử lý dữ liệu, lưu data sau xử lý thành cleaned\_df và lưu vào thư mục Data để sử dụng**

# Xem xét sự phân bố giá trị của các cột dữ liệu dạng số

Đối với các cột có kiểu dữ liệu số, nhóm em sẽ tính:

- missing\_ratio : Tỷ lệ phần trăm (từ 0 đến 100) giá trị bị thiếu
- min : Giá trị tối thiểu
- lower\_quartile : Tứ phân vị dưới (phân vị 25)
- median : Trung vị (phân vị 50)
- upper\_quartile : Tứ phân vị trên (phân vị 75)
- max : Giá trị tối đa

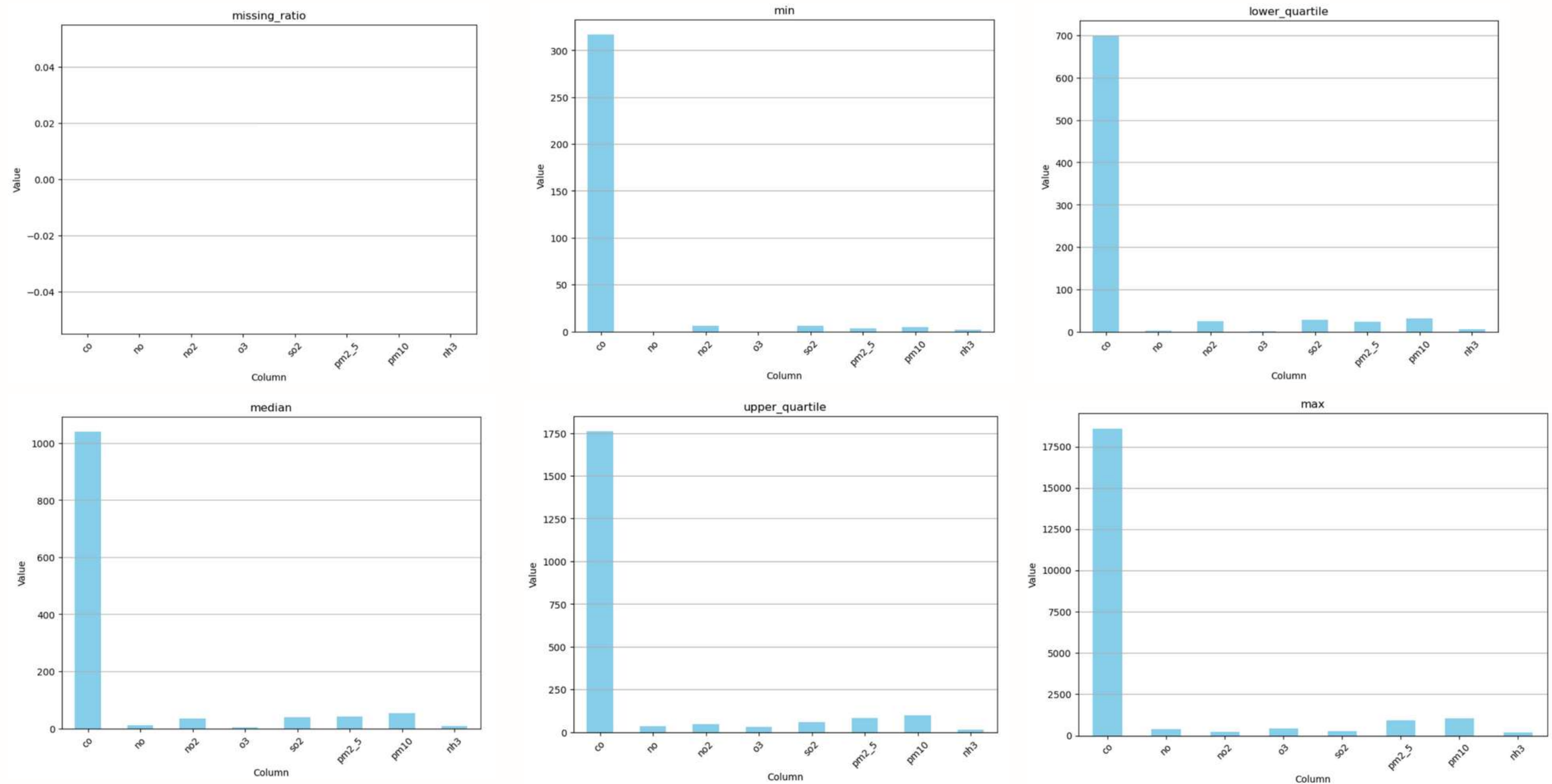


	co	no	no2	o3	so2	pm2_5	pm10	nh3
<b>missing_ratio</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
<b>min</b>	317.1	0.0	6.3	0.0	5.8	3.4	4.7	1.5
<b>lower_quartile</b>	701.0	1.8	24.3	0.0	27.2	23.3	31.0	6.0
<b>median</b>	1041.4	9.8	33.2	3.9	38.6	42.2	53.3	8.5
<b>upper_quartile</b>	1762.4	32.6	45.9	31.1	57.7	80.8	98.6	12.4
<b>max</b>	18585.2	393.4	213.9	446.3	270.8	936.1	1034.3	186.4



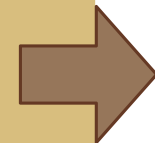
# Xem xét sự phân bố giá trị của các cột dữ liệu dạng số

Biểu đồ phân bố cho từng thông số **"missing\_ratio"**, **"min"**, **"lower\_quartile"**, **"median"**, **"upper\_quartile"**, **"max"**

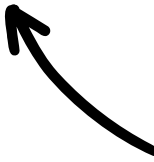


# Xem xét sự phân bố giá trị của các cột dữ liệu **không** phải dạng số

- Phần trăm giá trị thiếu sót.
- Số lượng giá trị (các giá trị ở đây là các giá trị khác nhau và không xét giá trị thiếu).
- Phần trăm của mỗi giá trị được sắp xếp theo tỉ lệ phần trăm giảm dần.
- Lưu kết quả vào **DataFrame** **cat\_col\_info\_df**, trong đó:
  - Tên của các cột là tên của các cột không phải là số trong `air_quality_df`.
  - Tên của các dòng là: **"missing\_ratio"**, **"num\_values"**, **"value\_ratios"**.



là cột dữ liệu  
"category"



**aqi**

**missing\_ratio**

0.0

**num\_values**

5

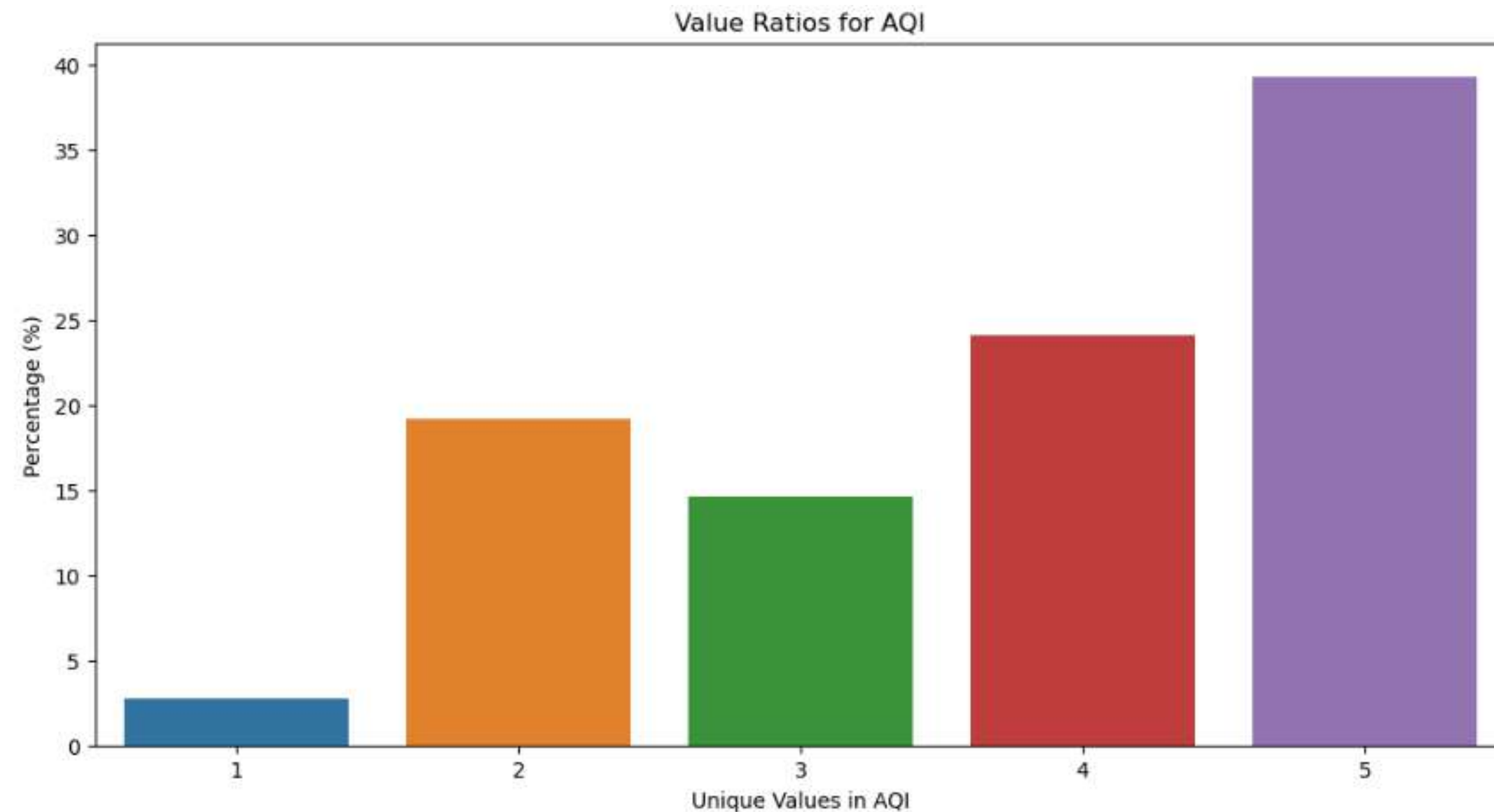
**value\_ratios** {5: 39.3, 4: 24.1, 2: 19.2, 3: 14.6, 1: 2.7}

# Xem xét sự phân bố giá trị của các cột dữ liệu **không** phải dạng số

## Nhận xét:

- Nhìn vào biểu đồ, ta có thể thấy được chất lượng không khí **rất kém** chiếm tỉ lệ cao nhất (39,3%), tiếp đến là chất lượng không khí kém (24,1%).
- Tỉ lệ chất lượng không khí ở mức khá cao hơn mức trung bình.
- Tỉ lệ chất lượng không khí tốt rất ít (chỉ 2,7%).

Vậy, không khí có chất lượng rất kém và kém rất phổ biến.



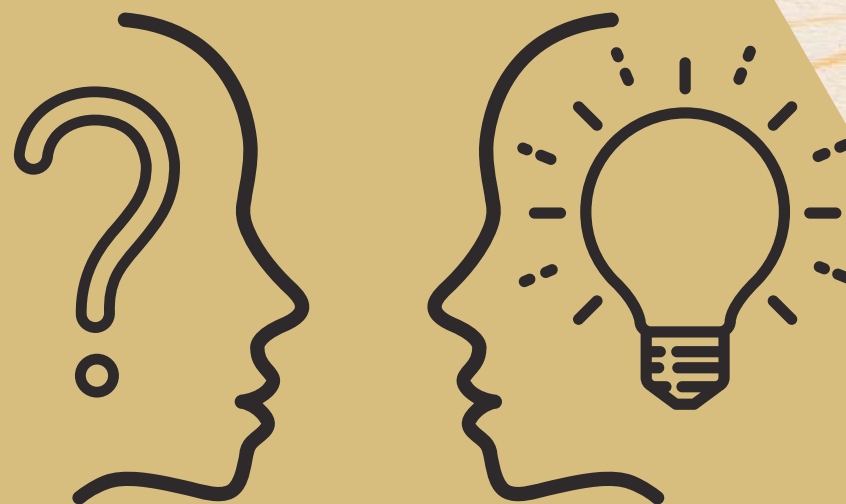


# Pha 03.

## Đặt câu hỏi và trả lời

Sau khi đã khám phá dữ liệu và có cái nhìn tổng quan về dữ liệu, tiến hành khai thác dữ liệu thông qua những câu hỏi giúp nhắm rõ hơn về mối quan hệ của các dữ liệu.

Tiếp tục

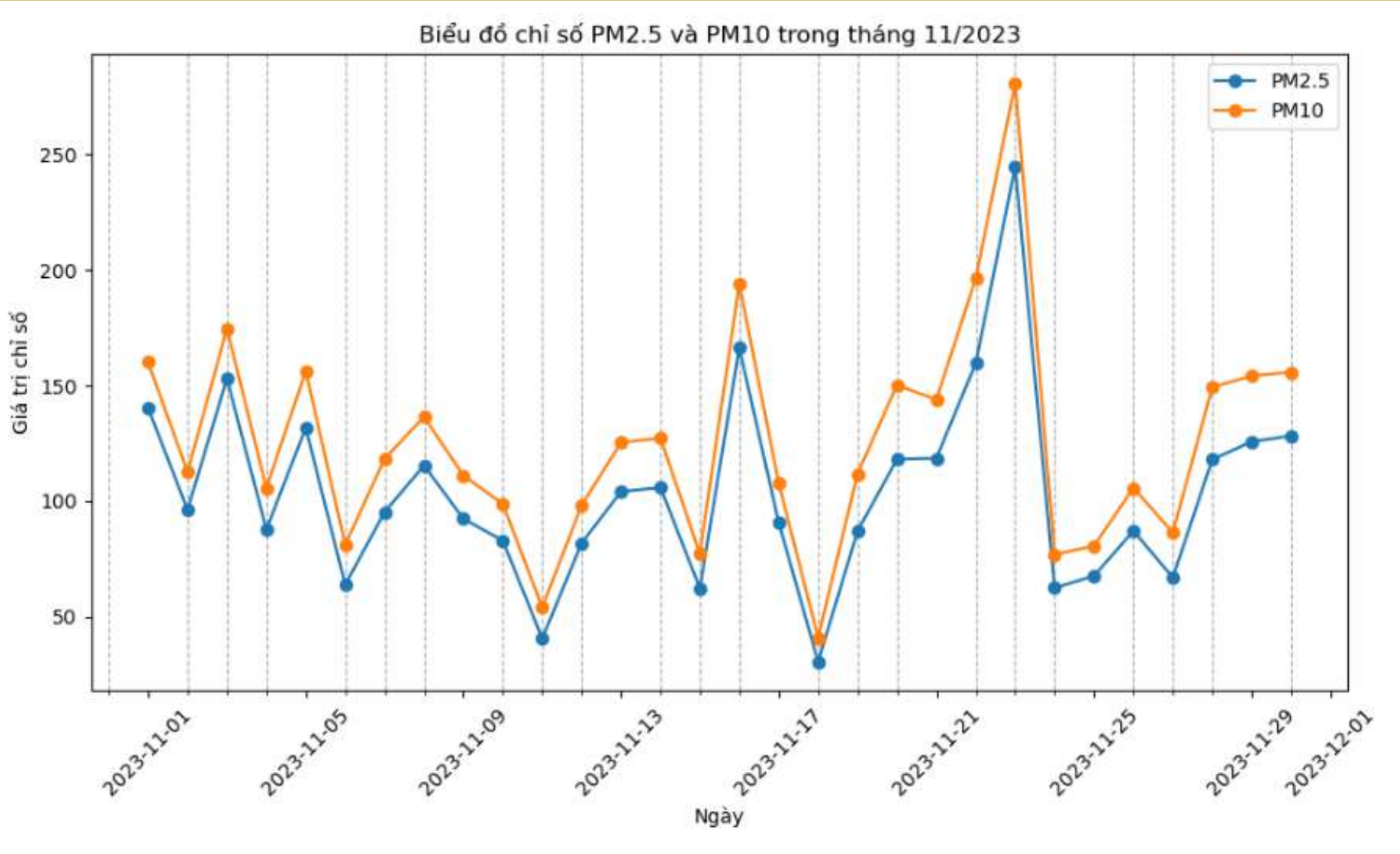


# Câu hỏi 1: Tình hình biến đổi của chỉ số bụi (PM2.5, PM10) trong khu vực nghiên cứu trong thời điểm gần nhất (tháng 11/2023) là như thế nào?

- **Ý nghĩa:** giúp ta hiểu rõ hơn về mức độ ô nhiễm không khí vì bụi trong thời kỳ gần đây và có thể đánh giá tình hình ô nhiễm ở giai đoạn cụ thể này.

	dt	pm2_5	pm10
1034	2023-11-01	140.233750	160.085000
1035	2023-11-02	96.292500	112.643333
1036	2023-11-03	152.972917	174.512083
1037	2023-11-04	87.808750	105.729583
1038	2023-11-05	131.744583	156.283333
1039	2023-11-06	63.880833	81.139167
1040	2023-11-07	95.296250	118.343333
1041	2023-11-08	115.453333	136.435833
1042	2023-11-09	92.357083	111.108333
1043	2023-11-10	82.953333	98.789167

- **Để phân tích dữ liệu trả lời cho câu hỏi này:**
  - Tạo một dataframe mới gồm các cột thời gian (dt) và 2 cột chỉ số cần xét (pm2\_5, pm10)
  - Xét cột dt làm index và sử dụng .resample('D') để chuyển đổi dữ liệu theo ngày và tính trung bình theo từng ngày
  - Reset index để có DataFrame với cột 'dt' trở thành một cột thông thường và lọc dữ liệu trong khoảng thời gian cần xét (tháng 11/2023)
  - Xem xét phân bố của 2 giá trị và trực quan hóa kết quả



Biểu đồ thể hiện sự thay đổi của chỉ số PM2.5 và PM10 trong tháng 11/2023

	dt	pm2_5	pm10
count	30	30.000000	30.000000
mean	2023-11-15 12:00:00	104.200125	125.734069
min	2023-11-01 00:00:00	30.561667	41.055417
25%	2023-11-08 06:00:00	82.157708	98.382292
50%	2023-11-15 12:00:00	95.794375	115.493333
75%	2023-11-22 18:00:00	123.954688	153.191875
max	2023-11-30 00:00:00	244.757500	280.703750
std	NaN	42.532152	48.173162

Bảng thể hiện phân bố giá trị của 2 giá trị PM2.5 và PM10 trong tháng 11/2023

- Có thể thấy sự thay đổi của 2 chỉ số này gần như là tương đương nhau nhưng chỉ số PM10 luôn lớn hơn.
- Kết quả thấp nhất ghi nhận được là vào 18/11/2023 , PM2.5 và PM10 lần lượt là 30,56 và 41,06
- Kết quả cao nhất ghi nhận được là vào 23/11/2023, PM2.5 và PM10 lần lượt là 244,75 và 280,7.
- Cả 2 chỉ số đều thường xuyên giao động ở mức cao (trung bình của PM2.5 là 104,2 và của PM10 là 125,7) cho thấy bụi trong không khí rất nhiều, điều này cho thấy không khí đang bị nhiễm bụi khá nặng. Nguyên nhân có thể đến từ mật độ giao thông lớn của thành phố và các hoạt động công nghiệp.



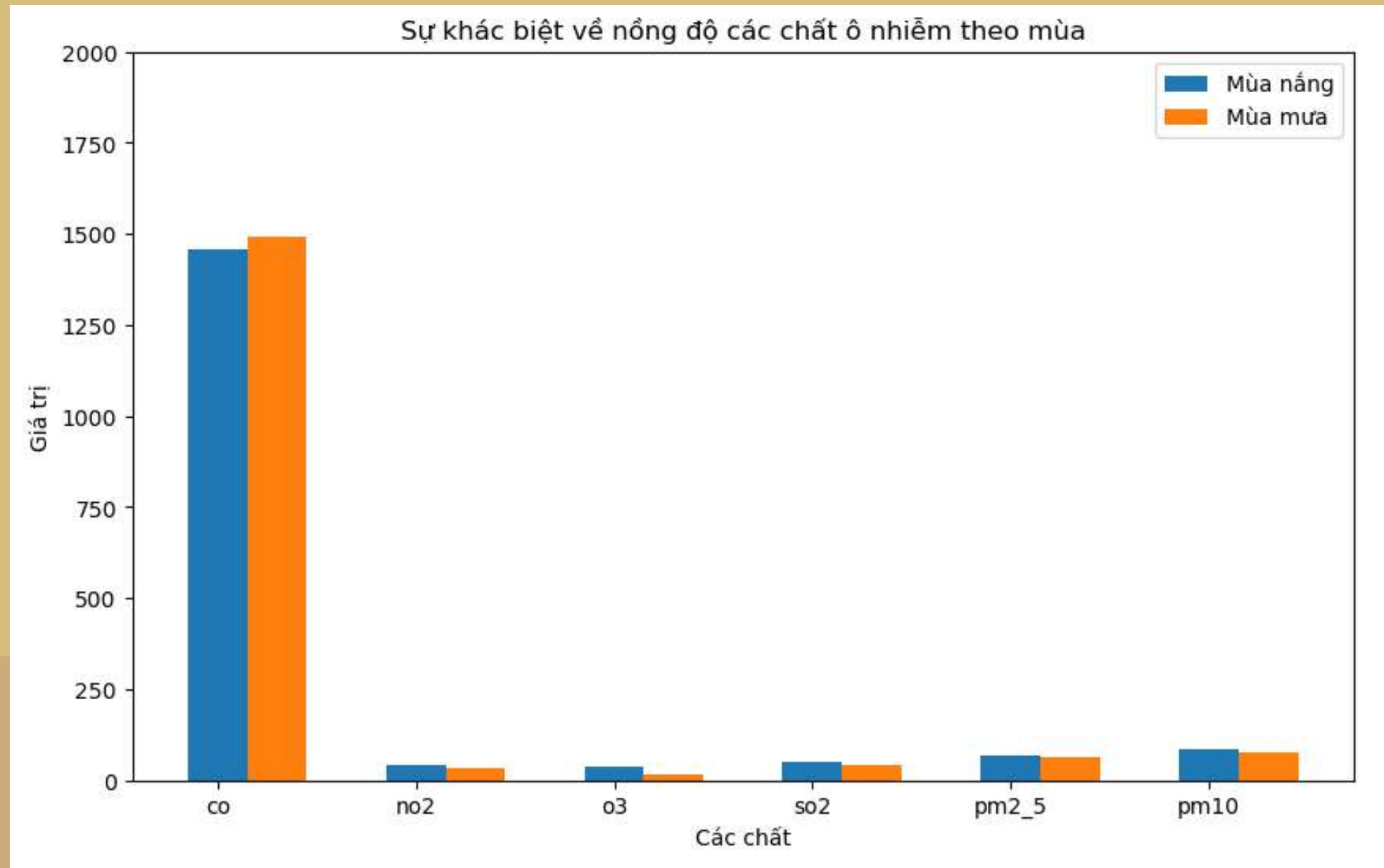
## Câu hỏi 2: Liệu có sự thay đổi trong mức độ ô nhiễm trong các mùa khác nhau (mùa nắng, mùa mưa)?

- **Ý nghĩa:** giúp ta phân tích sự thay đổi của ô nhiễm không khí qua các mùa khác nhau từ đó đánh giá tác động của điều kiện thời tiết đối với chất lượng không khí. Bên cạnh đó còn có thể nắm bắt được xu hướng tăng/giảm của ô nhiễm không khí giữa các mùa có thể giúp xác định các nguyên nhân có thể gây ra sự thay đổi này, ví dụ như hoạt động công nghiệp, giao thông, hay thậm chí là điều kiện thời tiết đặc biệt.

	co	no2	o3	so2	pm2_5	pm10
season						
Mùa mưa	1490.669402	32.902067	15.032181	40.322187	64.456662	75.741135
Mùa nắng	1455.854356	43.850319	37.159481	52.090282	66.770227	83.436144

- **Để phân tích dữ liệu trả lời cho câu hỏi này:**
  - Lọc dữ liệu trong năm 2021
  - Tạo cột 'season' và lưu mùa với mùa mưa là các tháng từ 5 tới 10 và mùa nắng là các tháng còn lại bằng cách xét tháng trong cột 'dt'
  - Gom nhóm theo cột 'season' và tính trung bình các chỉ số theo mùa
  - Xem xét kết quả và trực quan hóa kết quả

## Câu hỏi 2: Liệu có sự thay đổi trong mức độ ô nhiễm trong các mùa khác nhau (mùa nắng, mùa mưa)?



Biểu đồ thể hiện sự khác biệt về nồng độ các chất ô nhiễm theo mùa

Từ biểu đồ ta có thể nhận thấy: Hầu hết chỉ số các chất ô nhiễm đều cao hơn vào mùa nắng, chỉ CO là thấp hơn nhưng mức chênh lệch không lớn. Điều này cho thấy không khí có xu hướng ô nhiễm hơn vào mùa nắng. Nguyên nhân có thể do:

- Hoạt động giao thông diễn ra thường xuyên hơn vào mùa nắng.
- Các hoạt động công nghiệp sử dụng nhiên liệu đốt tăng mạnh khi nhu cầu sử dụng năng lượng lớn.
- Thời tiết khô ráo và có gió vào mùa nắng là điều kiện phát tán lí tưởng của các chất gây ô nhiễm



# Câu hỏi 3: Chỉ số chất lượng không khí (AQI) thay đổi như thế nào trong mỗi giờ (từ 0h đến 23h) của các ngày trong tuần (thứ 2 đến chủ nhật)?

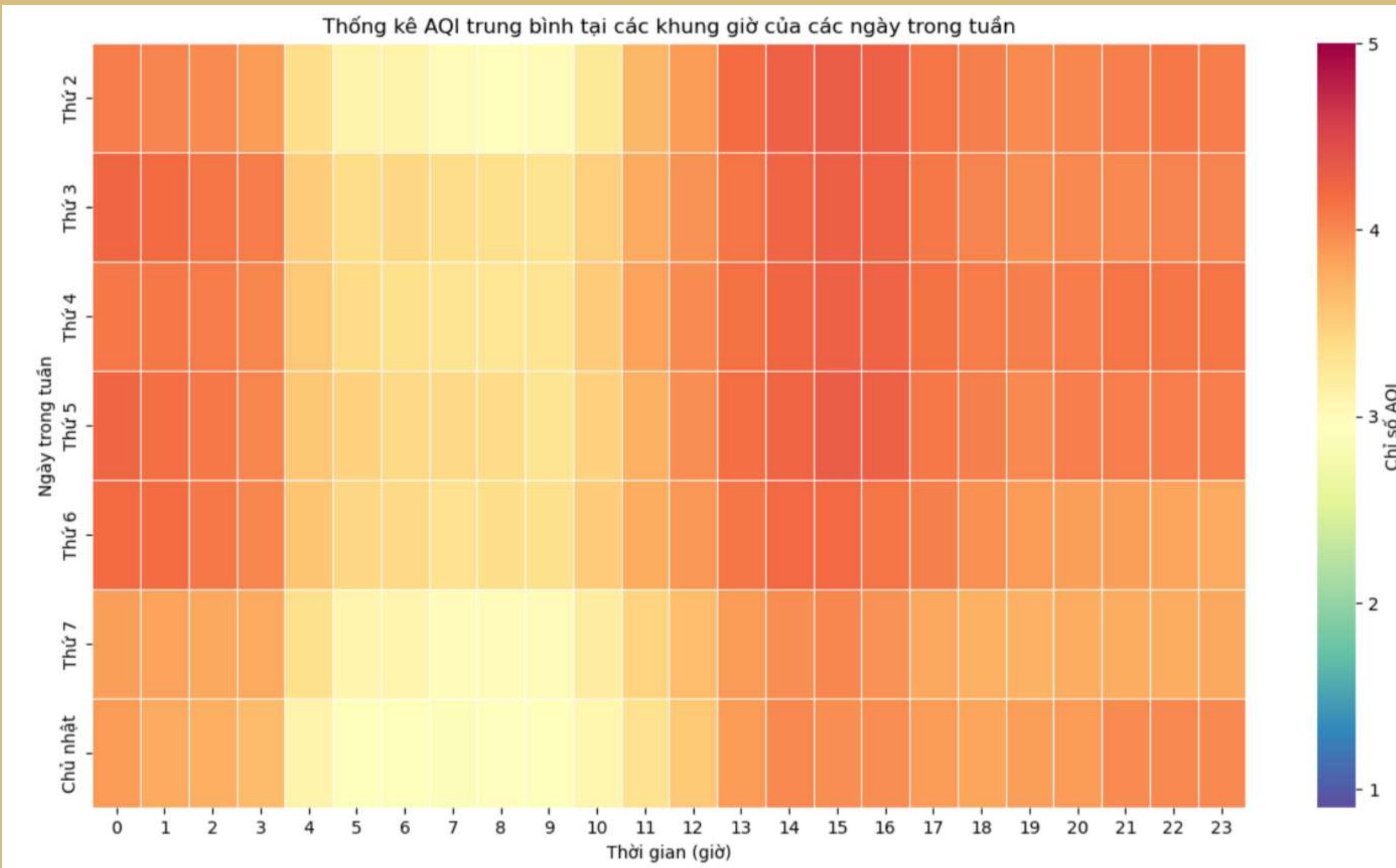
- **Ý nghĩa:** giúp ta đánh giá tình trạng chất lượng không khí theo từng khung giờ của các ngày trong tuần.

	dt	aqi	Day of week	Hour of day
0	2021-01-01 00:00:00	3	4	0
1	2021-01-01 01:00:00	3	4	1
2	2021-01-01 02:00:00	3	4	2
3	2021-01-01 03:00:00	3	4	3
4	2021-01-01 04:00:00	2	4	4
...	...	...	...	...
25172	2023-11-30 20:00:00	5	3	20
25173	2023-11-30 21:00:00	5	3	21
25174	2023-11-30 22:00:00	4	3	22
25175	2023-11-30 23:00:00	3	3	23
25176	2023-12-01 00:00:00	3	4	0

- **Phân tích dữ liệu để trả lời câu hỏi này**
  - Lấy dữ liệu từ 2 cột “dt” và “aqi”
  - Cột “Day of week”: trích xuất thứ trong tuần
  - Cột “Hour of day”: trích xuất giờ trong ngày
  - Tính chỉ số AQI trung bình rồi lưu vào một matrix có kích thước 7×24, trong đó mỗi phần tử biểu thị chỉ số AQI trung bình trong một giờ nhất định của một ngày trong tuần.



## Biểu đồ thống kê AQI trung bình tại các khung giờ của các ngày trong tuần



### Nhận xét

- Chỉ số AQI có sự thay đổi qua từng khung giờ và từng ngày trong tuần.
- Chỉ số AQI trung bình của TP.HCM dao động ở mức từ 2-4 (khá đến kém).
- Vào thứ 2, thứ 7 và chủ nhật, từ 4h-10h sáng chỉ số AQI ở mức 2-3. Còn thứ 3,4,5 và thứ 6, từ 4h-10h sáng chỉ số AQI ở mức 3-4. Ở các khung giờ còn lại, chỉ số AQI khoảng mức 4.

➔ Điều này có thể cho thấy mức độ các hoạt động của TP.HCM thường có xu hướng diễn ra mạnh mẽ vào buổi trưa chiều và tối ở các ngày giữa tuần.

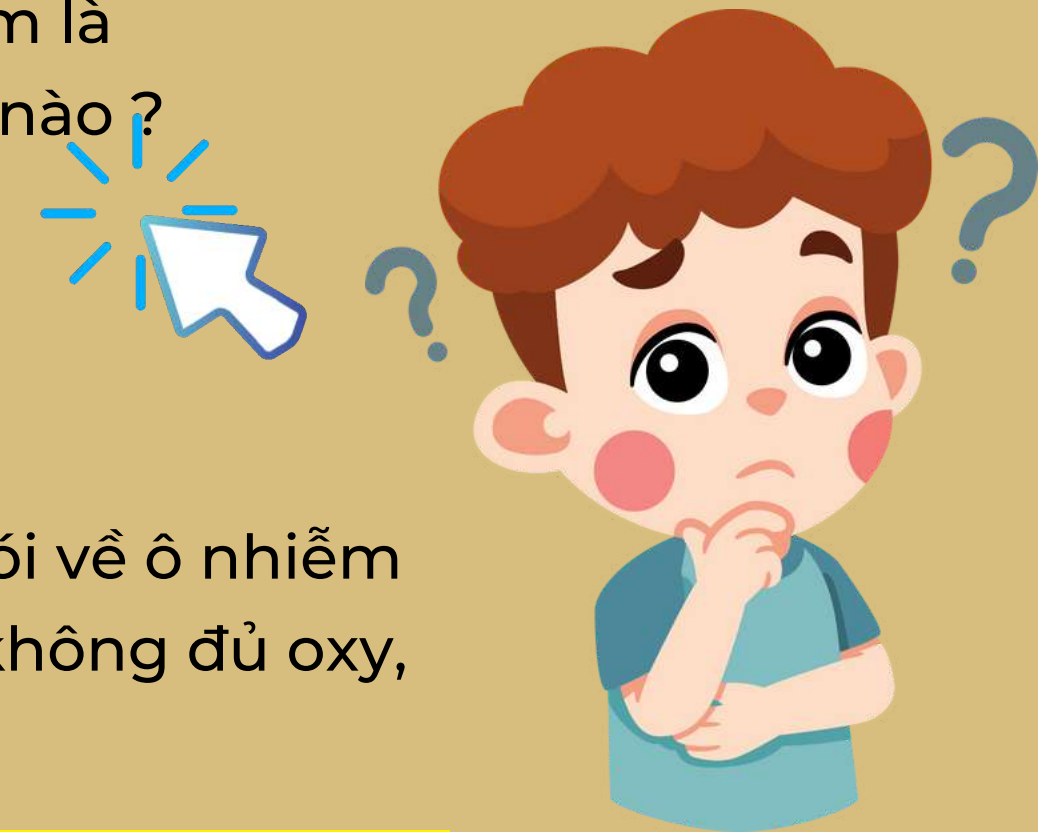
## Câu hỏi 4: Trong khoảng thời gian buổi đêm (0h-6h) so với buổi sáng (6h-12h) trong tháng gần nhất năm 2023, có sự biến đổi đáng kể nào về nồng độ CO trong không khí?

- Trả lời câu hỏi này sẽ giúp ta biết được, giữa 2 khoảng thời gian này, được xem là khoảng thời gian khi ta ngủ và khi ta thức dậy khí CO có sự thay đổi như thế nào?

- **Vậy khí CO là gì? Ảnh hưởng như nào đến con người?**

- Là một loại khí độc hại mà người ta thường xem xét khi nói về ô nhiễm không khí. CO được tạo ra chủ yếu từ quá trình đốt cháy không đủ oxy, thường là từ xe cộ, nhà máy, đốt cháy nhiên liệu.

- Ngộ độc khí CO từ nhẹ đến nặng



# Câu hỏi 4: Trong khoảng thời gian buổi đêm (0h-6h) so với buổi sáng (6h-12h ) trong tháng gần nhất năm 2023, có sự biến đổi đáng kể nào về nồng độ CO trong không khí?

- Ta lọc ra dữ liệu tháng gần nhất năm 2023 trong dữ liệu thu thập, tức tháng 11/2023.
- Tính trung bình lượng khí CO ( $\mu\text{g}/\text{m}^3$ ) trong buổi đêm và buổi sáng của từng ngày trong tháng.
- Kết quả thu được bảng sau:

part_of_day	co	
	night	morning
2023-11-01	1573.561429	2932.865000
2023-11-02	1007.080000	2692.540000
2023-11-03	1512.525714	3268.878333
2023-11-04	1637.458571	834.465000
2023-11-05	1152.990000	2265.293333
2023-11-06	1605.987143	788.848333
2023-11-07	903.130000	2067.248333
2023-11-08	1609.801429	1355.171667
2023-11-09	1576.424286	1268.386667
2023-11-10	2077.101429	1464.208333
2023-11-11	989.915714	895.658333

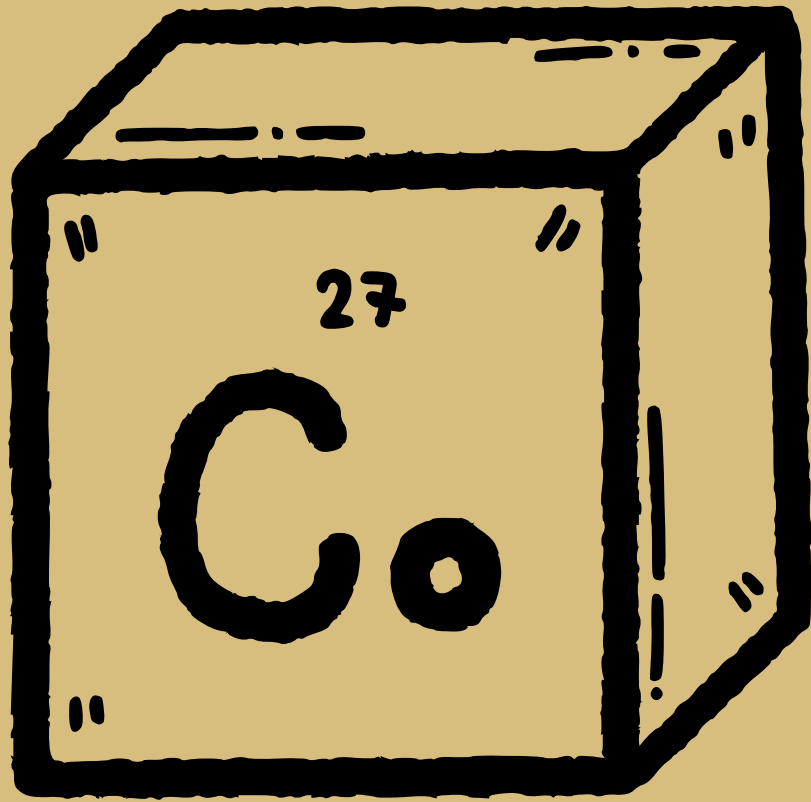
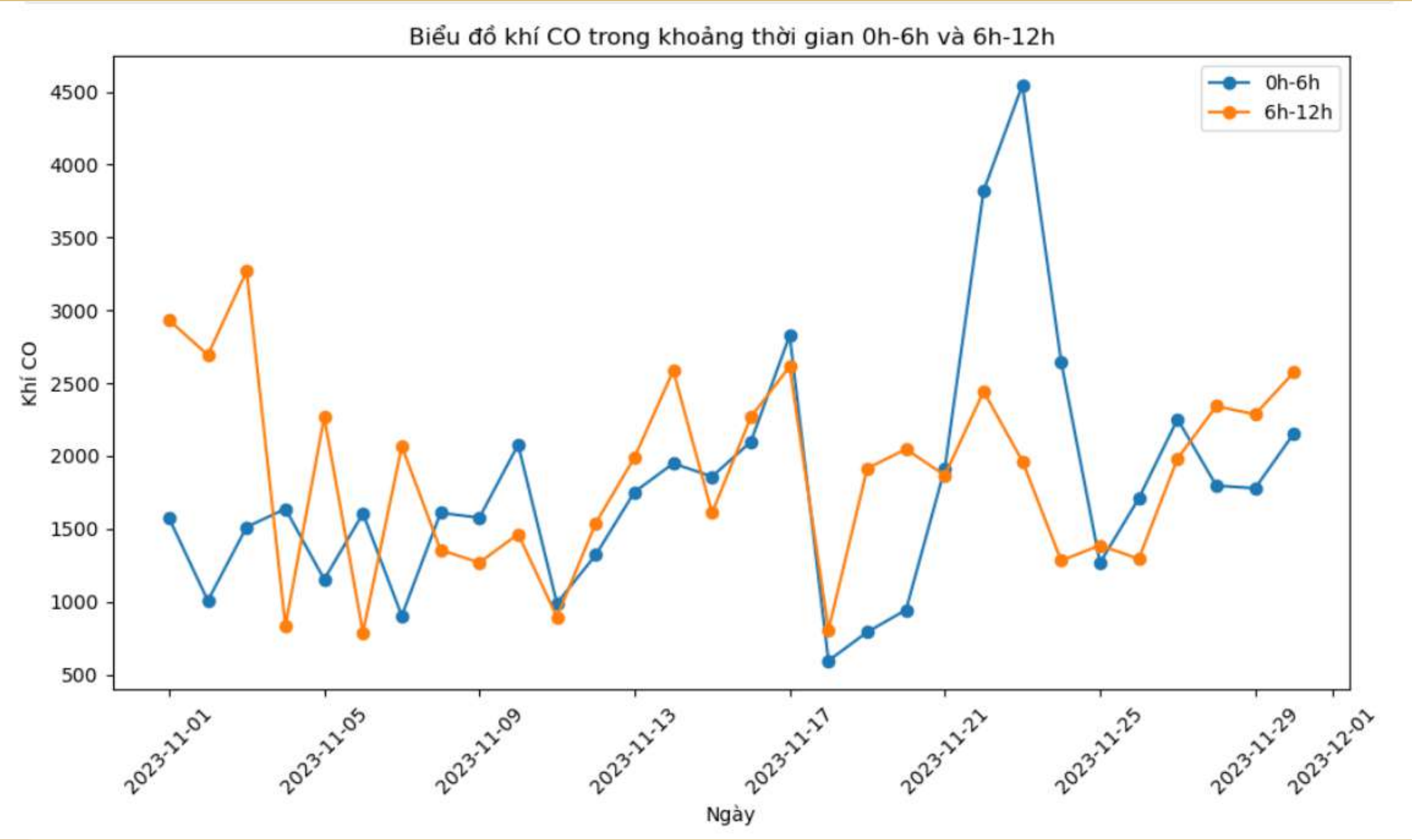


- Tạo ra 2 cột “night” và “morning” để dễ dàng vẽ biểu đồ so sánh



# Câu hỏi 4: Trong khoảng thời gian buổi đêm (0h-6h) so với buổi sáng (6h-12h ) trong tháng gần nhất năm 2023, có sự biến đổi đáng kể nào về nồng độ CO trong không khí?

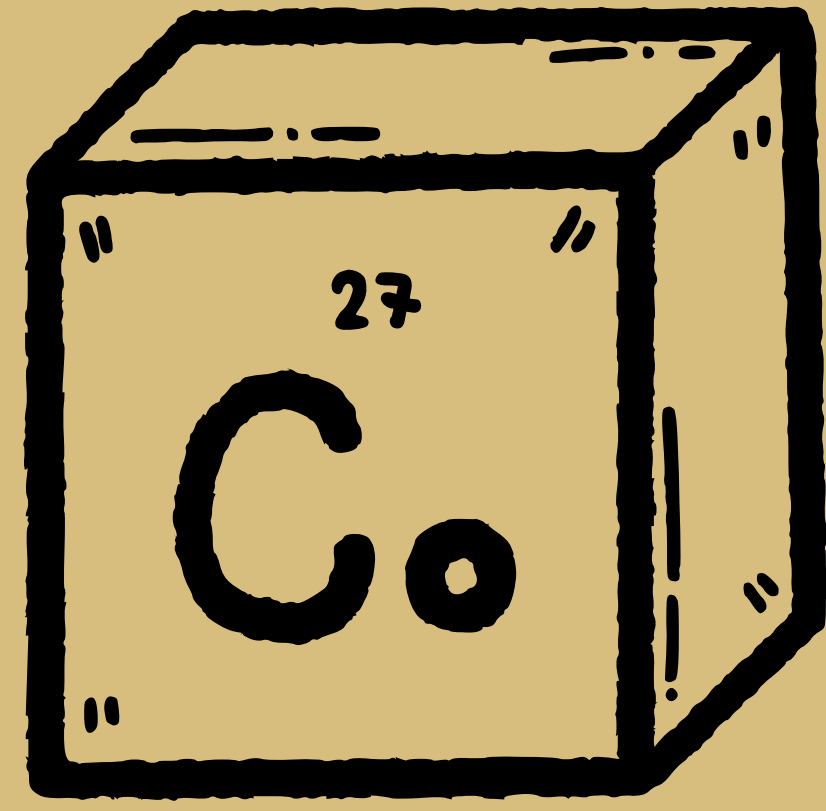
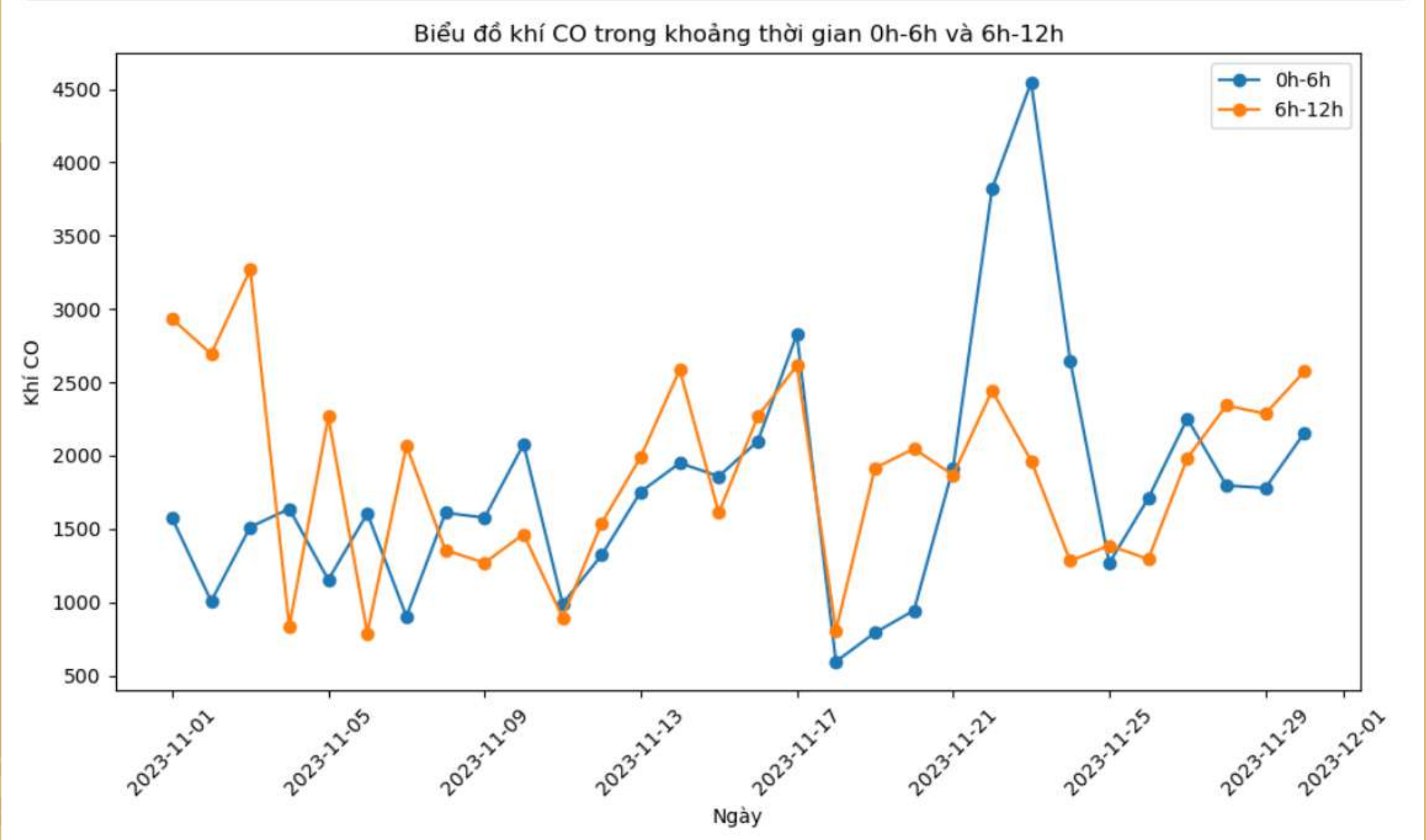
- Sử dụng thư viện Matplotlib để vẽ biểu đồ ta thu được biểu đồ như sau:





# Câu hỏi 4: Trong khoảng thời gian buổi đêm (0h-6h) so với buổi sáng (6h-12h ) trong tháng gần nhất năm 2023, có sự biến đổi đáng kể nào về nồng độ CO trong không khí?

- Nhận xét:
  - Giá trị trung bình lượng khí CO trong không khí đều ở ngưỡng an toàn.
  - Trong khoảng 6-12h đa số cao hơn so với 0-6h.
  - Tuy nhiên, trong tháng 11/2023 có 1 số ngày lượng khí CO buổi đêm rất cao so với buổi sáng. Ví dụ ngày 23/11/2023.



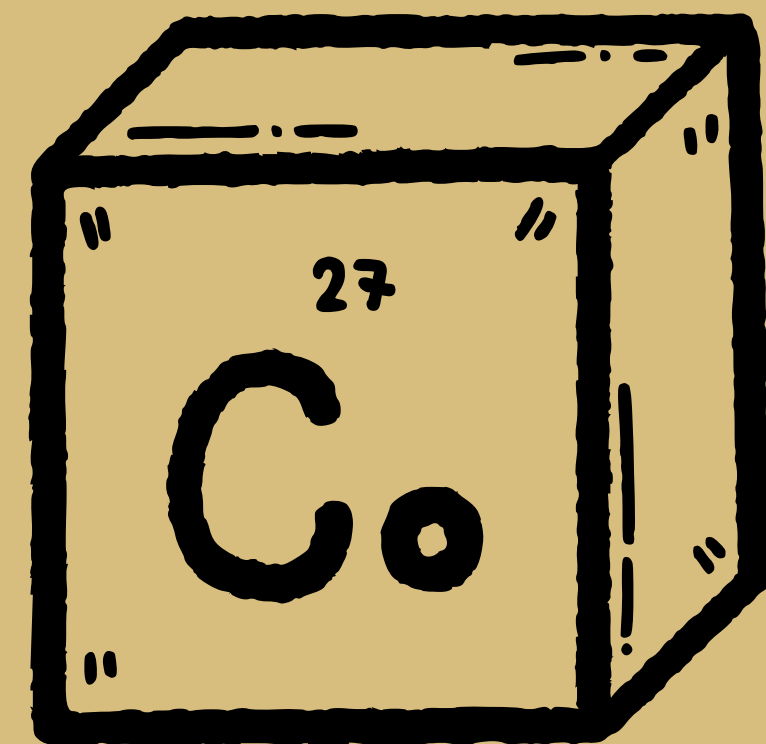
## Câu hỏi 4: Trong khoảng thời gian buổi đêm (0h-6h) so với buổi sáng (6h-12h) trong tháng gần nhất năm 2023, có sự biến đổi đáng kể nào về nồng độ CO trong không khí?

- **Giải thích :**

- Ban đêm dù hoạt động như xe cộ ít, nhưng 1 số khu công nghiệp nhà máy vẫn hoạt động.



- Hiện tượng nghịch nhiệt: nhiệt độ của không khí sát mặt đất giảm nhanh hơn các lớp không khí phía trên. Điều này khiến không khí sát mặt đất trở nên đặc hơn và khó lưu thông hơn. Các chất ô nhiễm, bao gồm cả khí CO, sẽ bị giữ lại trong không khí sát mặt đất, khiến nồng độ của chúng tăng cao.
- Sương mù khiến khí CO đọng lại trong sương, dẫn đến nồng độ khí tăng cao.



# Câu hỏi 5: Các chất ảnh hưởng đến chất lượng không khí có xu hướng tăng hay giảm qua các tháng trong năm?

**Ý nghĩa:** giúp ta đánh giá xu hướng các chất ảnh hưởng đến chất lượng không khí theo từng tháng -> đưa ra những biện pháp bảo vệ sức khỏe phù hợp, biết được chất nào sẽ tăng/giảm mạnh vào tháng nào.

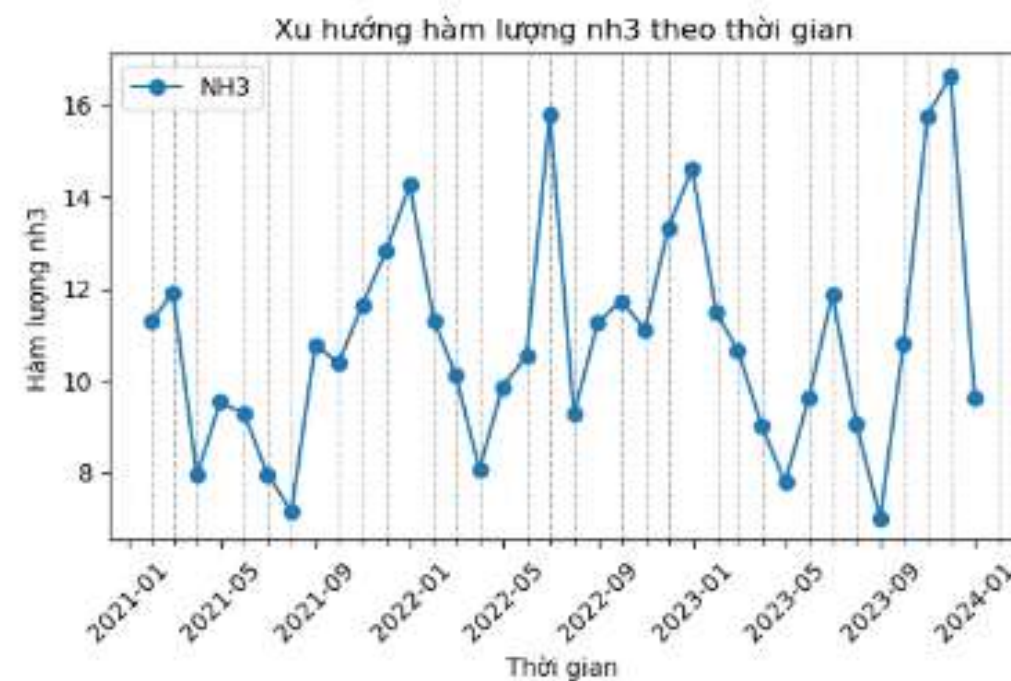
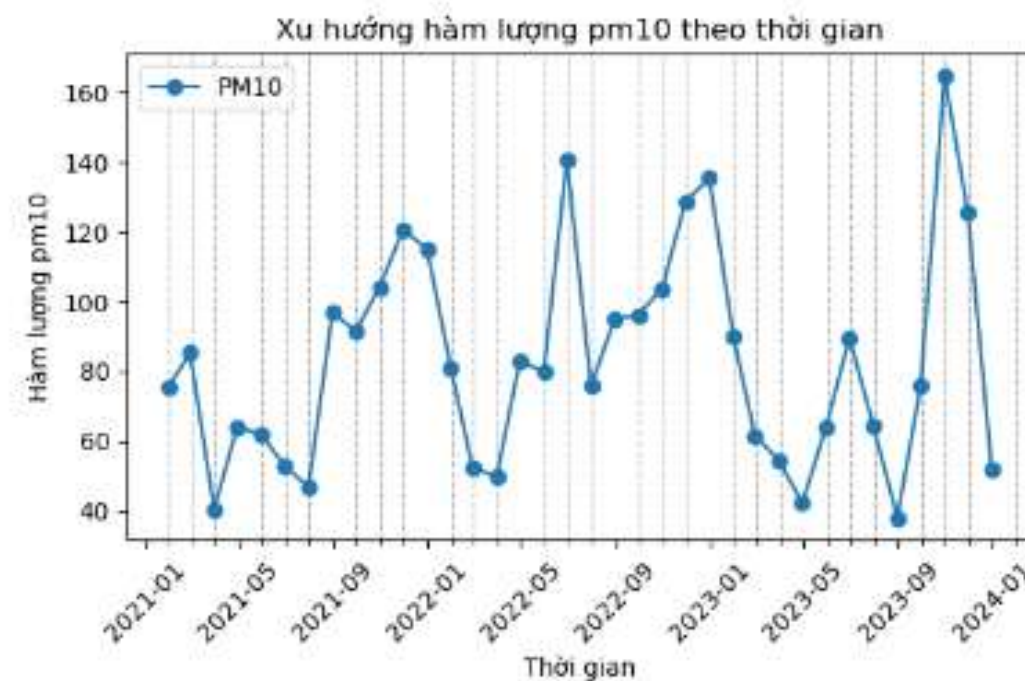
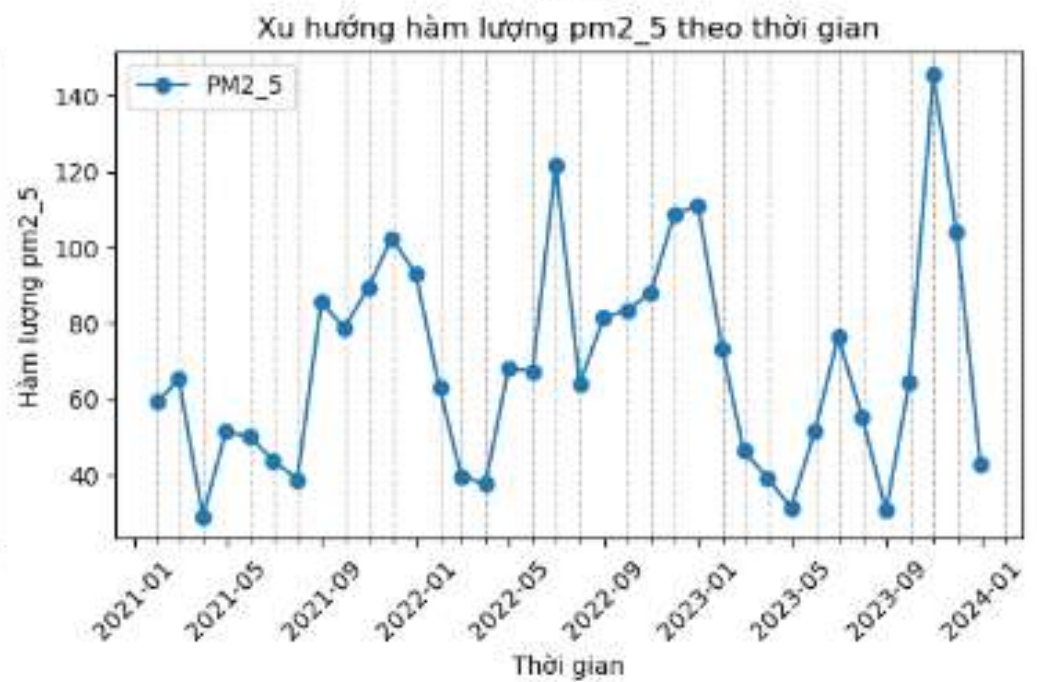
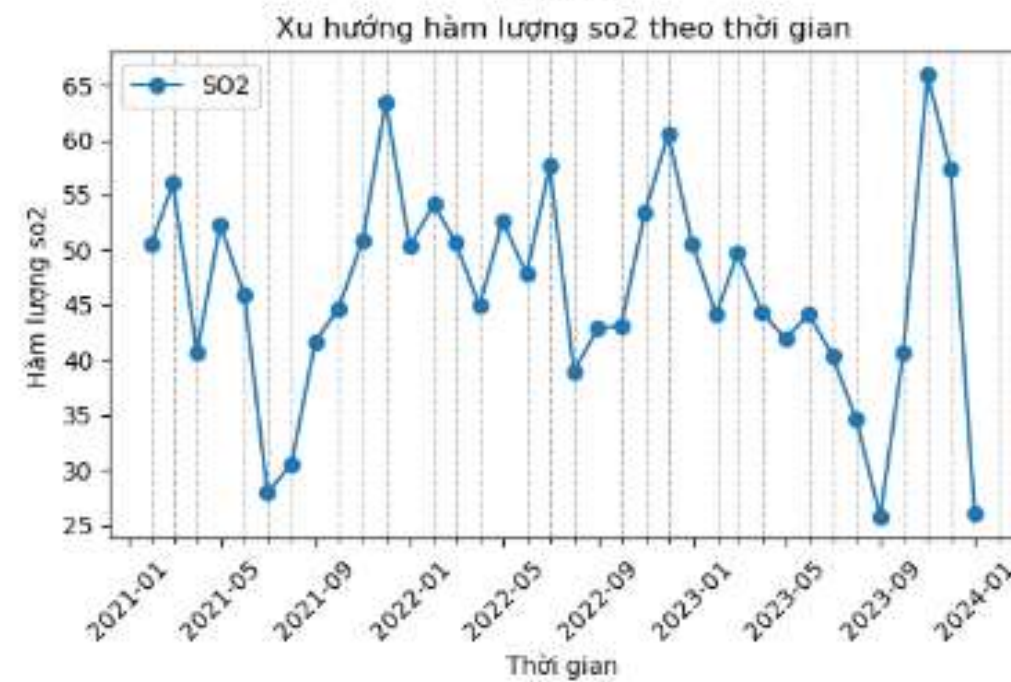
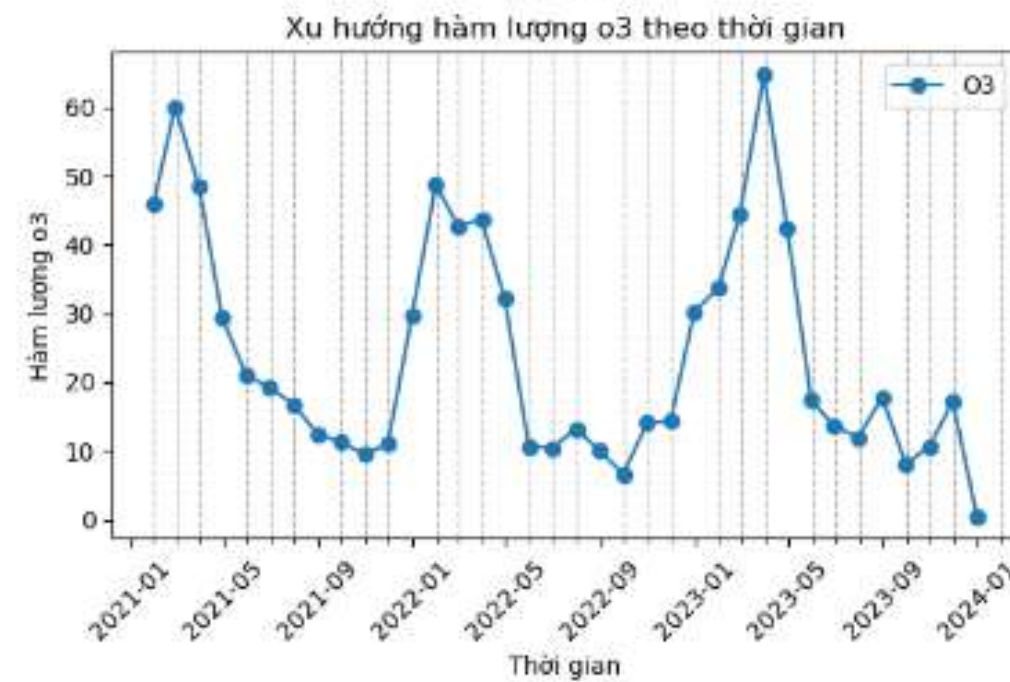
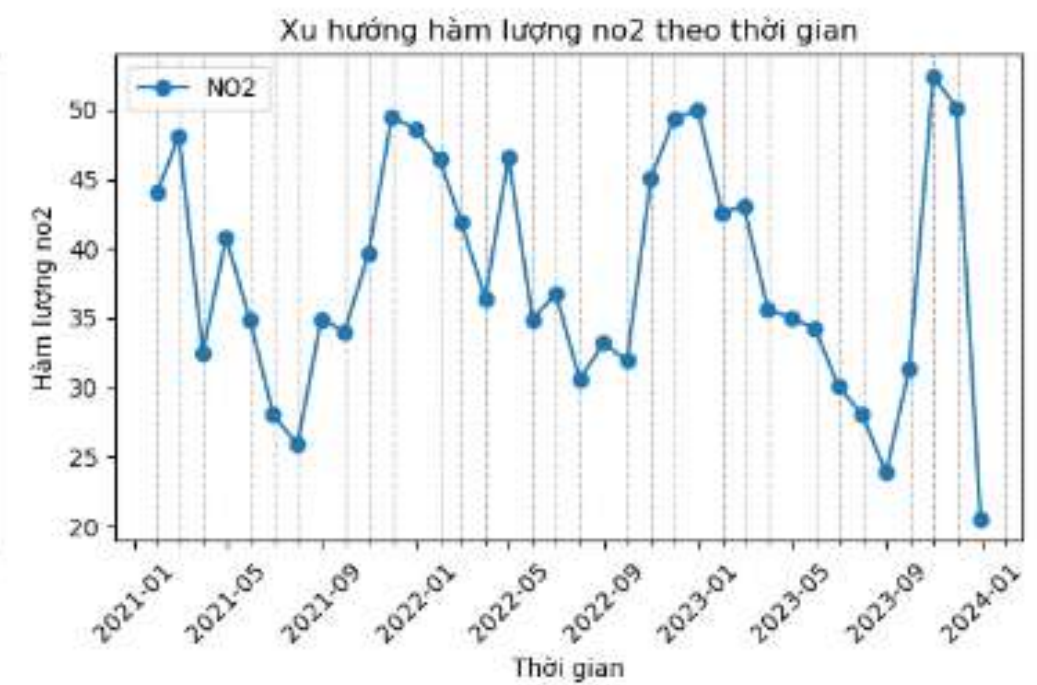
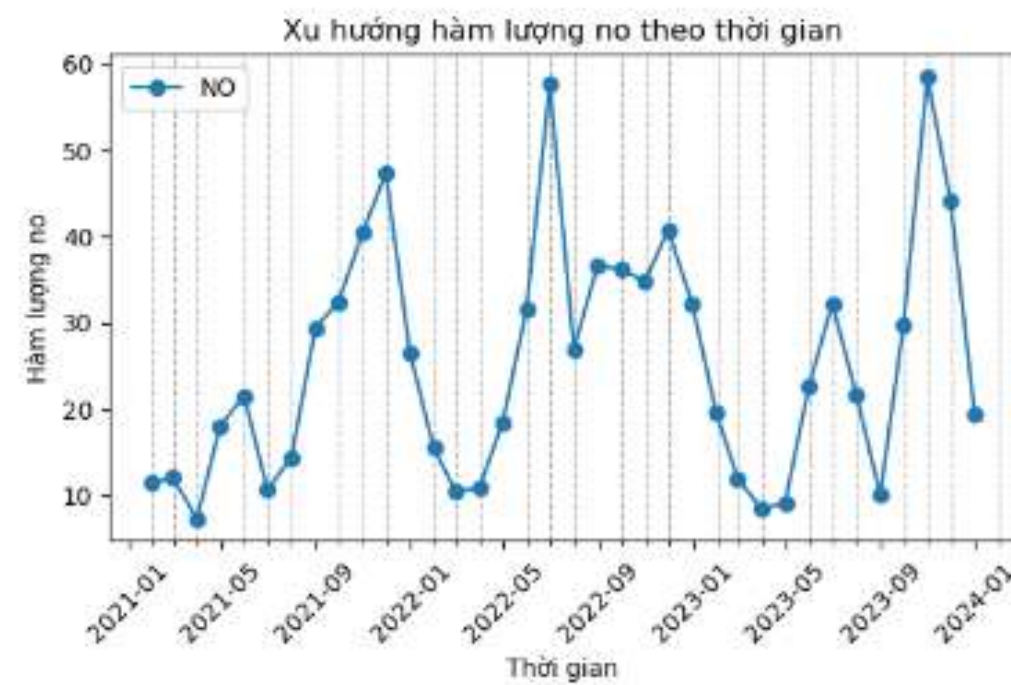
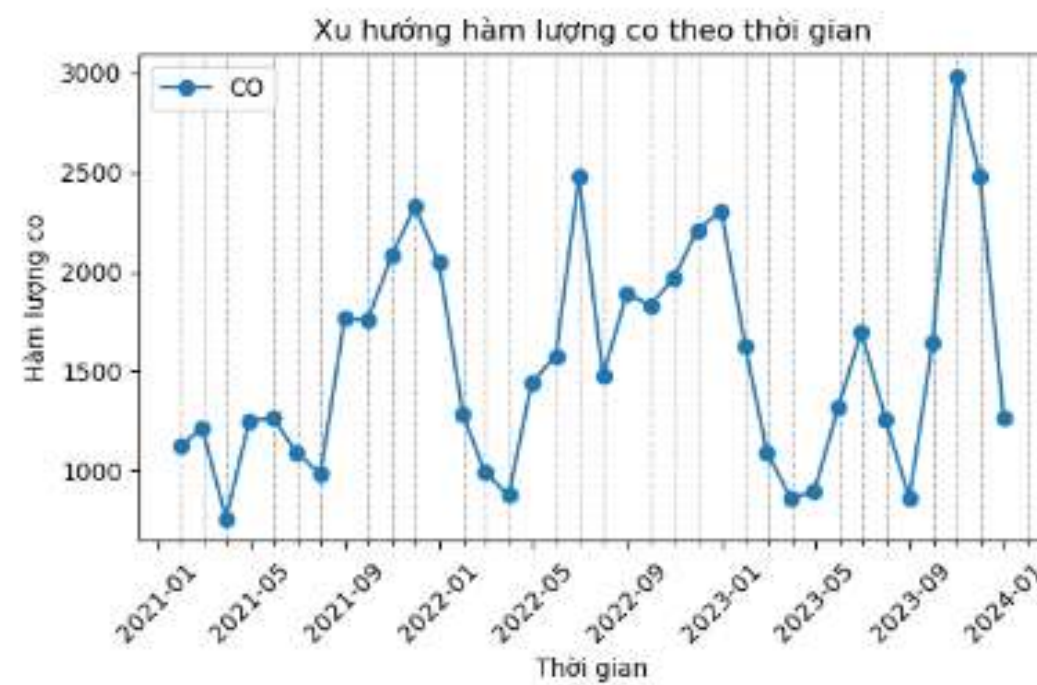
	dt	co	no	no2	o3	so2	pm2_5	pm10	nh3
0	2021-01-31	1122.597097	11.433903	44.051014	45.902986	50.500069	59.328667	75.461500	11.303486
1	2021-02-28	1215.552619	12.021652	48.128125	60.039702	56.046964	65.230685	85.290045	11.919211
2	2021-03-31	759.386210	7.332124	32.517231	48.422527	40.621694	29.149046	40.530349	7.951586
3	2021-04-30	1250.529625	18.077972	40.794903	29.289625	52.287833	51.501944	64.054792	9.540597
4	2021-05-31	1262.420269	21.518589	34.826022	20.992124	45.914449	50.046599	61.866183	9.284261
5	2021-06-30	1091.096056	10.646722	28.021750	19.260000	27.931236	43.780333	52.852778	7.975917
6	2021-07-31	982.802849	14.389113	25.890847	16.743145	30.560255	38.855054	46.780054	7.161344
7	2021-08-31	1763.566075	29.429543	34.926384	12.460860	41.675390	85.535134	97.035457	10.793938
8	2021-09-30	1757.768403	32.486431	34.015042	11.295069	44.731125	78.854333	91.447194	10.395375
9	2021-10-31	2082.089395	40.431438	39.610833	9.457715	50.863185	89.465981	104.233454	11.662191
10	2021-11-30	2328.519097	47.380181	49.478569	11.037236	63.460361	102.558014	120.623639	12.825750

**Để trả lời câu hỏi này,** chúng ta sẽ tính hàm lượng *trung bình của 1 tháng* rồi vẽ biểu đồ đường, mỗi đường là một chất và thời điểm sẽ là mỗi tháng trong năm.

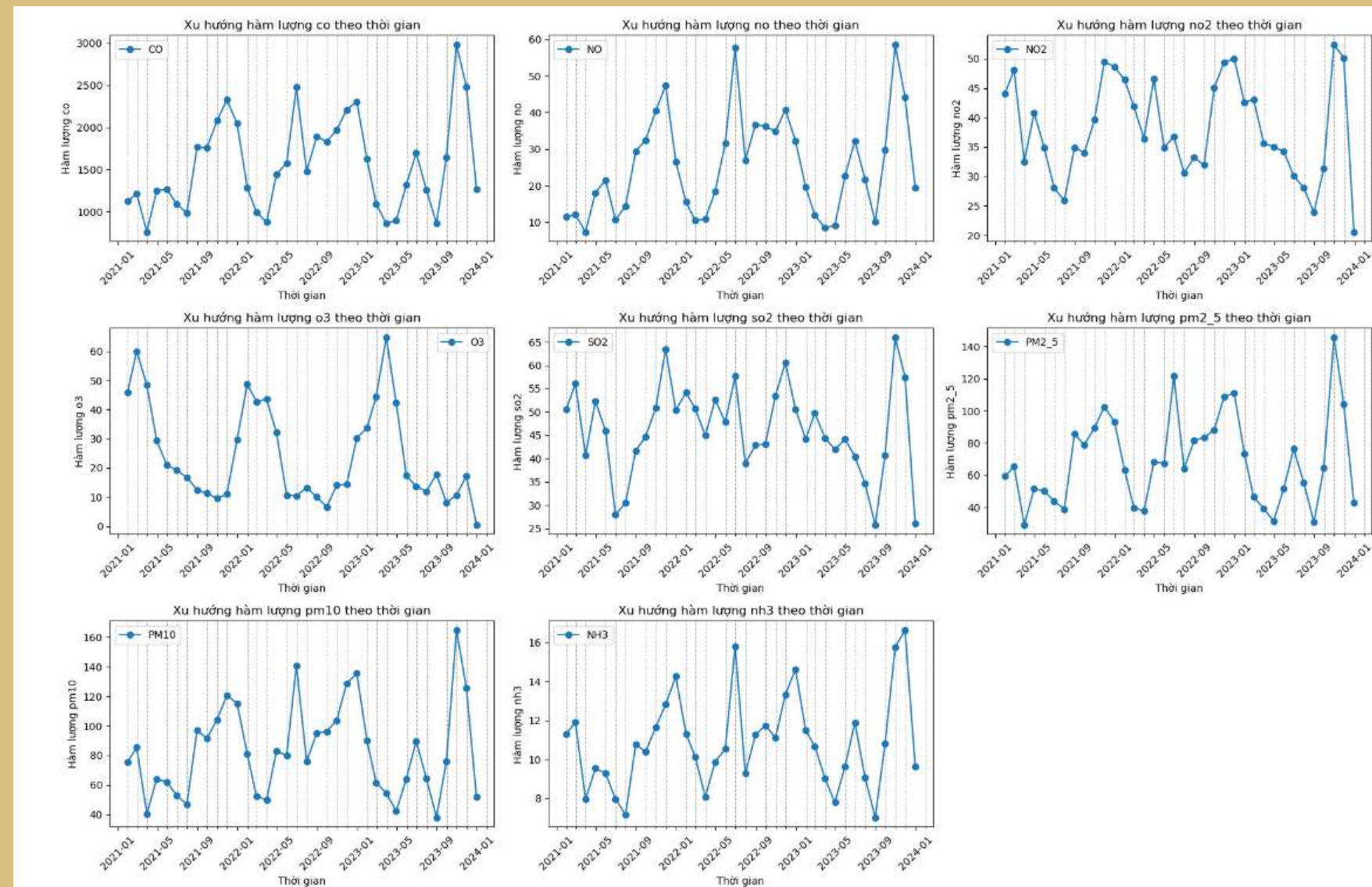


# Phân bố của các giá trị monthly_data.describe()									
	dt	co	no	no2	o3	so2	pm2_5	pm10	nh3
count	36	36.000000	36.000000	36.000000	36.000000	36.000000	36.000000	36.000000	36.000000
mean	2022-07-15 22:00:00	1555.130651	25.323017	38.085979	24.270385	46.323208	68.987209	83.322120	10.882145
min	2021-01-31 00:00:00	759.386210	7.332124	20.560000	0.510000	25.804637	29.149046	37.997581	7.020605
25%	2021-10-23 06:00:00	1114.971637	11.999937	32.368173	11.230611	41.445603	45.677860	59.761341	9.282330
50%	2022-07-15 12:00:00	1459.473552	22.225762	36.044285	17.328511	45.416102	64.806776	80.559276	10.735342
75%	2023-04-07 12:00:00	1911.500319	33.081036	45.374490	35.807767	52.375597	86.165312	98.708488	11.785993
max	2023-12-31 00:00:00	2981.855758	58.544562	52.387997	64.777667	65.974562	145.736857	164.747872	16.648403
std	NaN	552.935312	13.874441	8.377333	16.641215	9.730686	28.233170	31.353231	2.410833

Nhìn vào bảng phân bố giá trị, vì khí CO có hàm lượng rất cao so với các khí khác, nên nếu vẽ chung 1 biểu đồ thì sẽ rất khó đánh giá và quan sát, vì vậy em quyết định mỗi chất 1 biểu đồ để ta có thể phân tích chi tiết hơn.







Nhìn chung, hàm lượng các chất ô nhiễm không khí đều có xu hướng **tăng** dần trong giai đoạn từ năm 2021 đến năm 2023. Tháng 11, 12 hàm lượng các chất rất cao (trừ O3).

**Giải thích:** Nguyên nhân chính của tình trạng này là do các hoạt động sản xuất công nghiệp, sử dụng phương tiện giao thông và đốt nhiên liệu hóa thạch, các tháng mùa thu và mùa đông thường có nguồn nhiệt đốt nhiên liệu gia tăng, chẳng hạn như sưởi ấm bằng nhiên liệu fossile dẫn đến tăng cao của các khí CO, NO, NO2, và SO2 trong không khí..



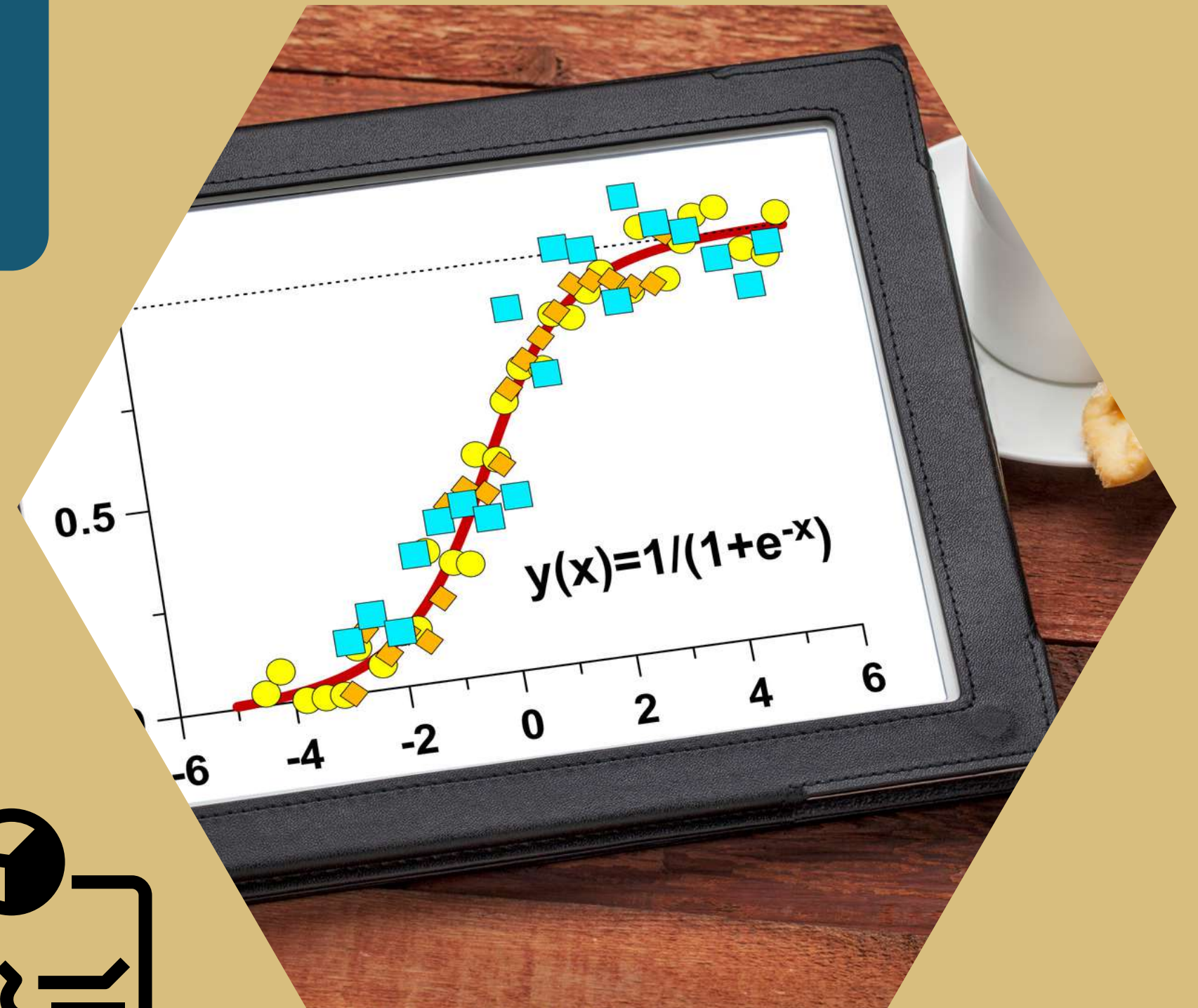
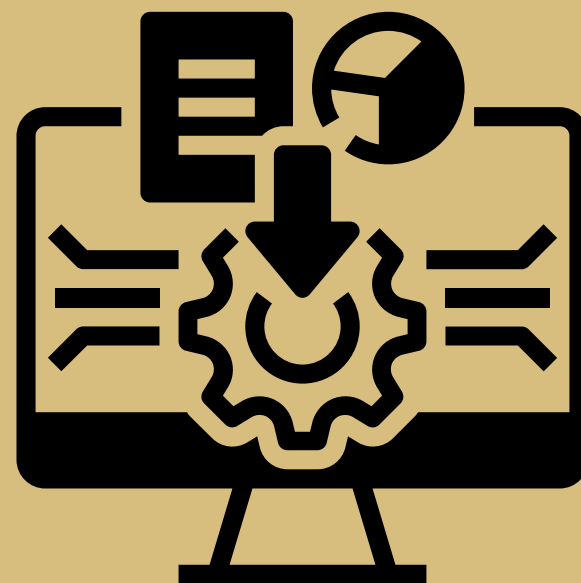


# Pha 04.

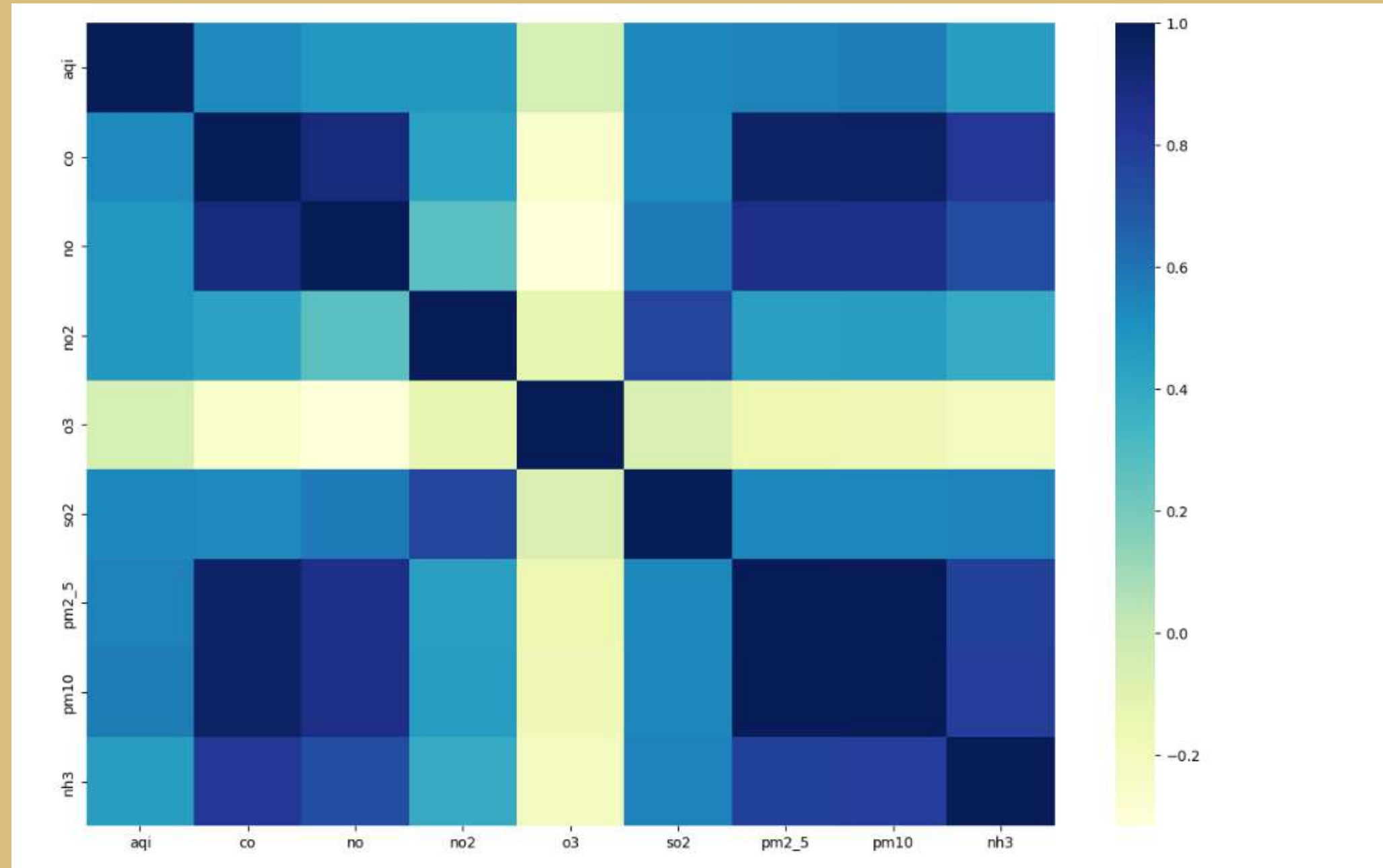
## Mô hình hóa dữ liệu

Bài toán đặt ra: Phân loại dữ liệu theo mức **aqi**

Tiếp tục



## Xem xét tương quan giữa các features



Dựa vào heatmap ta thấy chỉ số **o3** gần như không có sự tương quan với aqi còn mức tương quan của các chỉ số còn lại với **aqi** khá tương đồng nhau. Vì vậy tiến hành phân loại chất lượng không khí theo chỉ số **aqi** dựa trên các chỉ số **no, co, so2, no2, pm2\_5, pm10, nh3**

## Mục tiêu và lựa chọn thuộc tính

Tiến hành phân loại chất lượng không khí theo chỉ số **aqi** dựa trên các chỉ số **no**, **co**, **so2**, **no2**, **pm2\_5**, **pm10**, **nh3**

```
# Chia dữ liệu thành features (X) và target variable (y)
X = data[['no', 'co', 'so2', 'no2', 'pm2_5', 'pm10', 'nh3']] # Chọn các chỉ số khí quyển làm features
y = data['aqi'] # Chọn cột AQI làm target variable
```

## Feature Scaling

Khi khoảng giá trị giữa 2 thuộc tính quá cách xa nhau thì việc mô hình hóa cũng như trực quan mối quan hệ có thể gặp khó khăn, do đó phải thực hiện kĩ thuật 'Feature Scaling'

Trong bài này nhóm chọn phương pháp Standardisation để scaling khoảng giá trị của thuộc tính về khoảng gần hơn với giá trị của tập y là aqi.

```
# Chuẩn hóa dữ liệu
scaler = StandardScaler()
X = scaler.fit_transform(X)
```





## Phân tách bộ dữ liệu thành 3 tập training set, validation set và test set

- Mục đích:
  - Nếu không chia dữ liệu mà sử dụng toàn bộ dữ liệu để huấn luyện, mô hình có thể học "quá khớp" (overfitting), không tổng quát hóa được cho dữ liệu mới.
  - Tập kiểm tra giúp đánh giá xem mô hình có đang học từ dữ liệu một cách tổng quát hay chỉ học "nhớ" dữ liệu huấn luyện. Thông qua chúng ta có thể đánh giá hiệu suất của mô hình và cải thiện nó thông qua việc điều chỉnh các tham số hoặc phương pháp huấn luyện.
- Kích thước mỗi tập như sau:
  - Size of Training set = 80% \* (Size of Dataset)
  - Size of Test set = 20% \* (Size of Dataset).

```
# Chia dữ liệu thành tập train và tập test  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```





# Huấn luyện mô hình

---

## Lựa chọn mô hình máy học

Vì bài toán là phân loại chất lượng không khí theo chỉ số ***aqi*** - kiểu dữ liệu *category* nên nhóm sẽ sử dụng các thuật toán ***Classifier*** để mô hình hóa dữ liệu.

1

Mô hình Decision Tree

2

Mô hình SVM

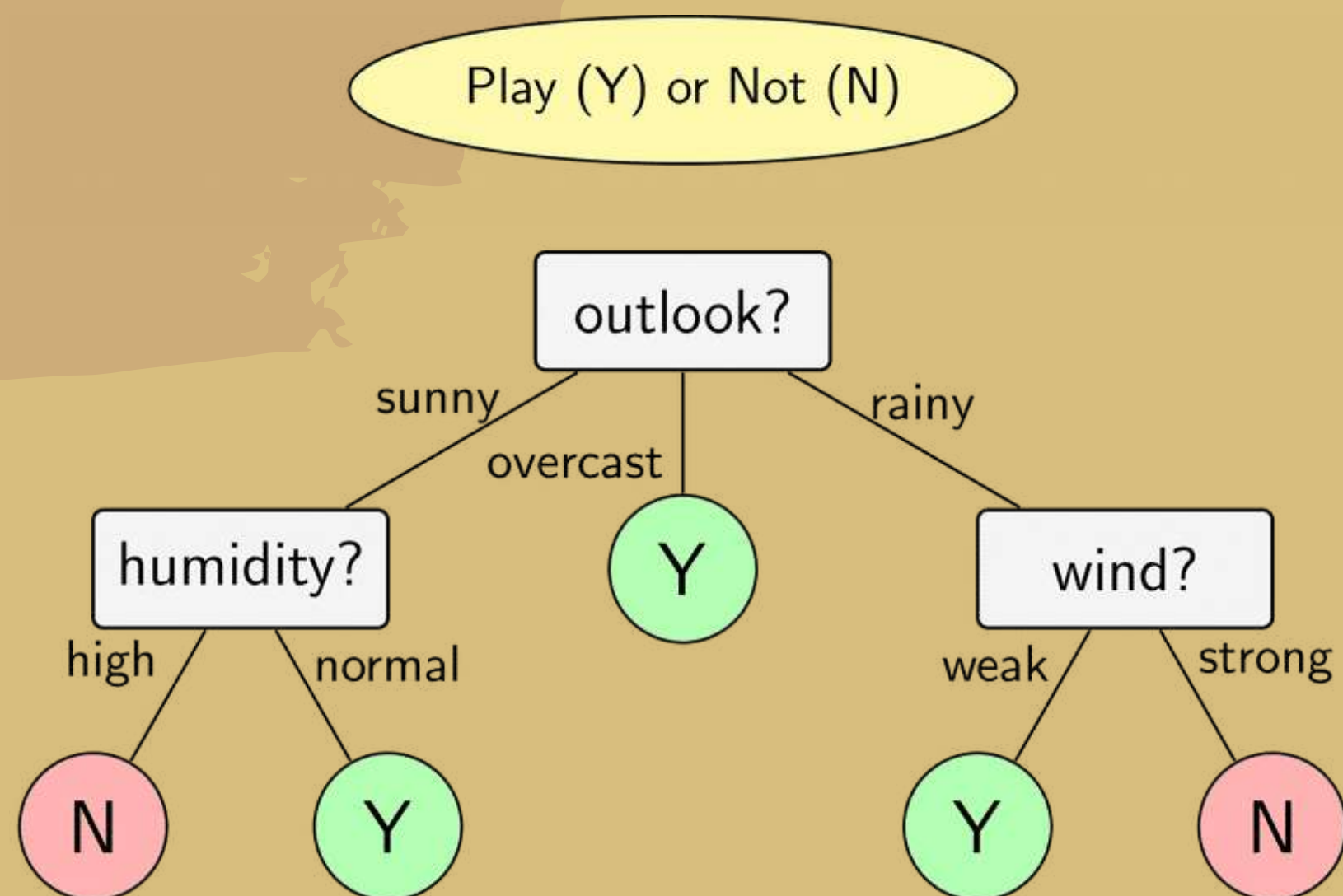
3

Mô hình KNN  
(K-Nearest Neighbors)

4

Mô hình Multi-layer  
Neural Network

# 1. Mô hình Decision Tree



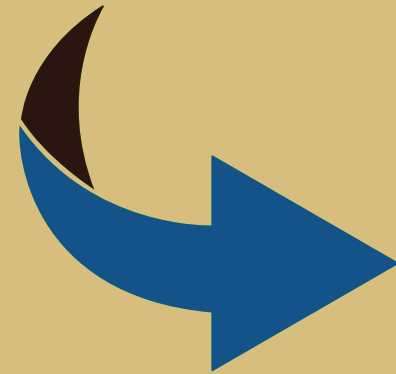
- Mô hình Decision Tree (cây quyết định) là một thuật toán học máy có cấu trúc dạng cây, chia dữ liệu dựa trên các quy tắc quyết định đơn giản. Mỗi nút lá của cây đại diện cho một nhãn hoặc một giá trị dự đoán, trong khi các nút gốc và nội bộ biểu diễn các quy tắc quyết định để chia tách dữ liệu.
- Cách hoạt động chính của Decision Tree:
  - Chọn thuộc tính quan trọng: Các thuộc tính được chọn dựa trên độ quan trọng của chúng trong việc chia dữ liệu.
  - Tách nút (node splitting): Mỗi nút trong cây đại diện cho một thuộc tính và một ngưỡng (threshold). Dữ liệu được chia thành các nhánh dựa trên giá trị của thuộc tính này.
  - Xây dựng cây: Quá trình tách nút được thực hiện đệ quy cho đến khi một điều kiện dừng được đáp ứng (như đạt đến độ sâu tối đa hoặc không thể chia tách thêm).

# 1. Mô hình Decision Tree

- Xây dựng một mô hình đơn giản với các siêu tham số như sau. Sử dụng Cross-Validation để chia dữ liệu thành 5 fold và đánh giá. Cuối cùng đánh giá mô hình trên tập test

```
# Khởi tạo mô hình Decision Tree
tree_model = DecisionTreeClassifier(criterion='gini', max_depth=3, min_samples_split=2, min_samples_leaf=1)

# Huấn luyện mô hình trên tập train
tree_model.fit(X_train, y_train)
```



```
# Đánh giá mô hình bằng cross-validation trên tập train
cv_scores = cross_val_score(tree_model, X_train, y_train, cv=5)
print("Cross-Validation Scores:", cv_scores)
print("Mean CV Accuracy:", cv_scores.mean())
```

```
# Đánh giá mô hình trên tập test (đánh giá cuối cùng)
test_accuracy = tree_model.score(X_test, y_test)
print(f'Test Accuracy: {test_accuracy}')
```

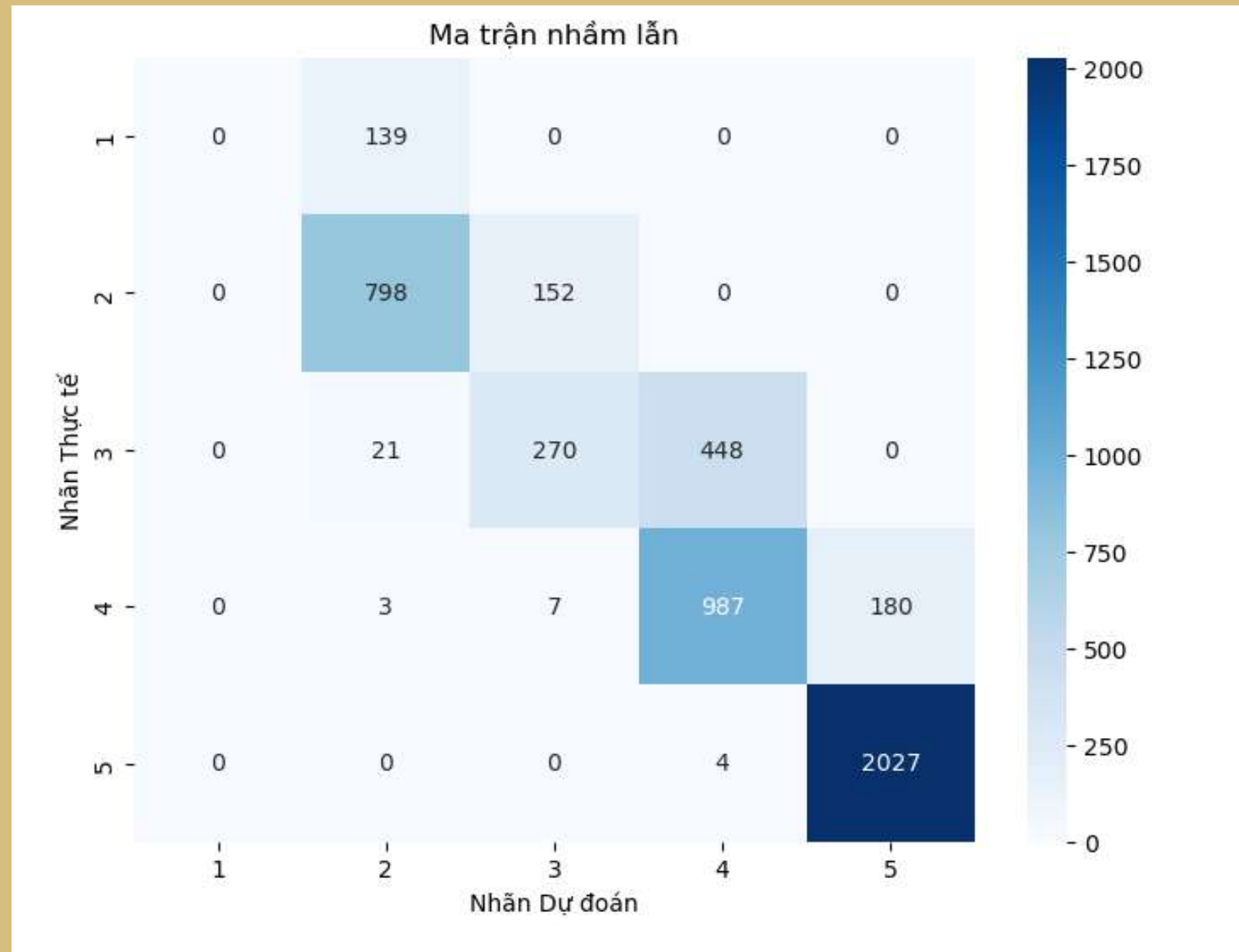
```
Cross-Validation Scores: [0.80635551 0.81330685 0.80685204 0.80983118 0.80362463]
Mean CV Accuracy: 0.8079940417080437
Test Accuracy: 0.8105639396346307
```



**Kết quả của mô hình sau khi chia 5 fold và sử dụng Cross-Validation cho ra kết quả khá tương đồng. Trung bình xấp xỉ 80%, mức này là khá tốt của 1 mô hình. Kết quả cho ra ở tập test cũng khác tương đồng (xấp xỉ 81%)**

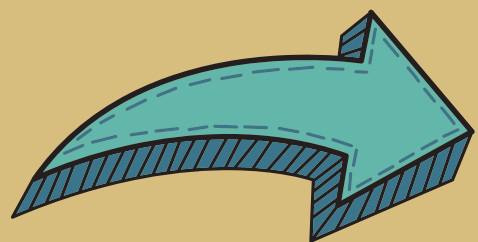
# 1. Mô hình Decision Tree

- Để có cái nhìn rõ hơn về kết quả của mô hình, nhóm tiến hành phân tích dữ liệu từ kết quả của mô hình được thực hiện thông qua ma trận nhầm lẫn (confusion matrix) và báo cáo phân loại (classification report)



## Báo cáo phân loại:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	139
2	0.83	0.84	0.84	950
3	0.63	0.37	0.46	739
4	0.69	0.84	0.75	1177
5	0.92	1.00	0.96	2031
accuracy			0.81	5036
macro avg	0.61	0.61	0.60	5036
weighted avg	0.78	0.81	0.79	5036

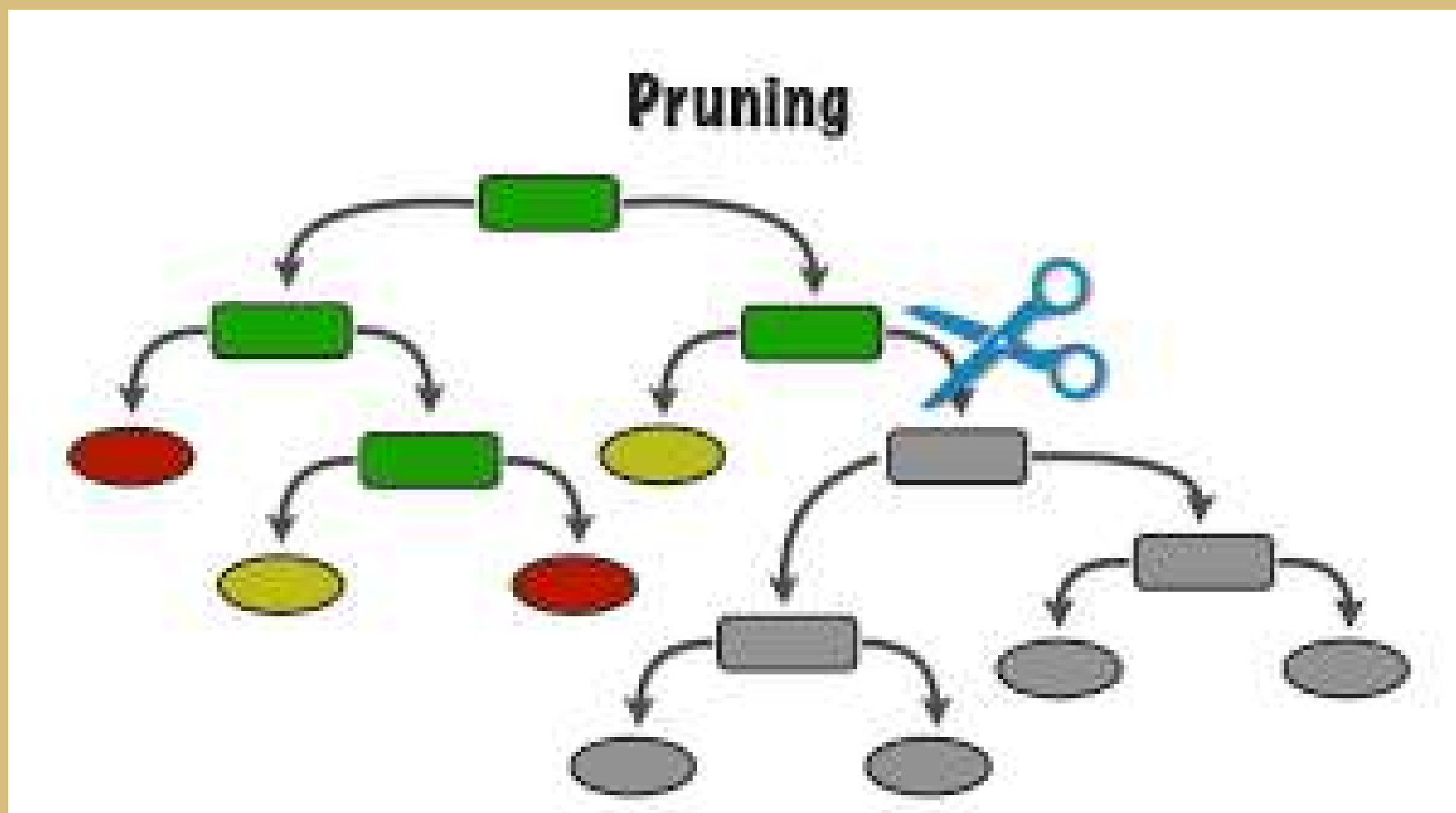


Giữa các lớp có sự chênh lệch về độ chính xác, điều này đặc biệt rõ với lớp 1 và lớp 5. Nguyên nhân có thể do sự phân bố không đều giữa số lượng dữ liệu của các lớp (dữ liệu thu thập được tập trung ở lớp 5 và ít có dữ liệu ở lớp 1)



# 1. Mô hình Decision Tree

- Mô hình có thể hoạt động tốt hơn bằng cách cắt tỉa (pruning). Quá trình cắt tỉa cây trong mô hình cây quyết định là quá trình loại bỏ một số nhánh hoặc lá của cây để cải thiện tính tổng quát của mô hình. Mục tiêu là loại bỏ các phần của cây mà không cung cấp nhiều thông tin hoặc góp phần vào overfitting.



- Cách thức hoạt động:
  - Tạo một cây quyết định ban đầu
  - Tính toán chi phí-phức tạp
  - Cắt tỉa cây
  - Huấn luyện lại mô hình
  - Đánh giá và kiểm tra hiệu suất



# 1. Mô hình Decision Tree

- Sử dụng method **cost\_complexity\_pruning path()** và **ccp.\_alphas** để tính toán alpha tốt nhất để cắt tỉa cây

```
# Tìm alpha tốt nhất để cắt tỉa cây
path = tree_model.cost_complexity_pruning_path(X_train, y_train)
ccp_alphas, impurities = path.ccp_alphas, path.impurities

# Tìm giá trị alpha tốt nhất
best_alpha = ccp_alphas[np.argmax(cv_scores)]

# Cắt tỉa cây theo alpha tốt nhất
pruned_tree_model = DecisionTreeClassifier(criterion='gini', ccp_alpha=best_alpha, random_state=42)
pruned_tree_model.fit(X_train, y_train)
```



- Đánh giá mô hình sau khi cắt tỉa trên tập test

```
# Đánh giá mô hình cắt tỉa trên tập test
test_accuracy_pruned = pruned_tree_model.score(X_test, y_test)
print(f'Test Accuracy of Pruned Tree: {test_accuracy_pruned}')
```

Test Accuracy of Pruned Tree: 0.835583796664019

# 1. Mô hình Decision Tree

- Mô hình có thể cho ra những kết quả tốt hơn với những siêu tham số khác, vì vậy ta có thể kiểm tra thông qua thư viện **GridSearchCV**

```
# Khởi tạo mô hình Decision Tree
dtree = DecisionTreeClassifier()

# Định nghĩa lưới các giá trị tham số cần tìm kiếm
param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [3, 5, 10, 15, 20],
    'min_samples_split': [2, 3, 5, 7, 10],
    'min_samples_leaf': [1, 2, 3, 4, 5]
}

# Tạo đối tượng GridSearchCV
tree_grid_search = GridSearchCV(dtree, param_grid, cv=5, scoring='accuracy')

# Thực hiện tìm kiếm trên lưới tham số
tree_grid_search.fit(X_train, y_train)

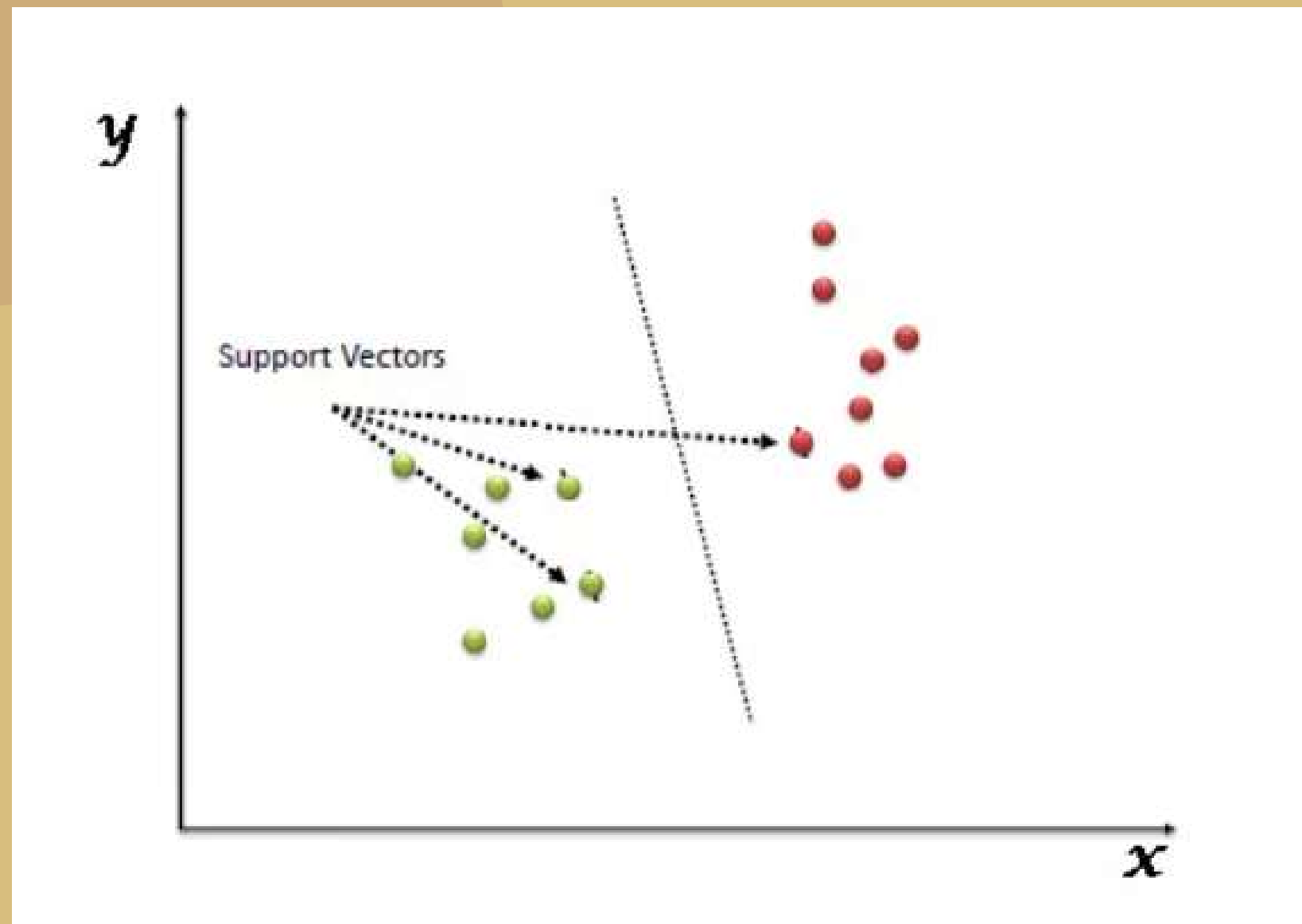
# In ra tham số tốt nhất
print("Best Parameters:", tree_grid_search.best_params_)

# In ra độ chính xác tốt nhất trên tập kiểm tra
print("Best Accuracy:", tree_grid_search.best_score_)

Best Parameters: {'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 2}
Best Accuracy: 0.8315292949354518
```

So sánh với kết quả từ việc cắt tỉa mô hình, kết quả khá tương đồng nhau (xấp xỉ 83%). Vì vậy ta có thể chọn model như từ những params vừa tìm được ở trên để dự đoán cho dữ liệu hiện có.

## 2. Mô hình SVM



Mô hình SVM (Support Vector Machine) là một thuật toán học có giám sát được sử dụng chủ yếu cho các vấn đề phân loại và hồi quy -> tập trung vào việc tìm một đường ranh giới phân chia tốt nhất giữa các lớp dữ liệu.

### Cách hoạt động của SVM:

1. Tìm đường ranh giới (Decision Boundary)
2. Tối ưu hóa ranh giới
3. Kernel Trick



## 2. Mô hình SVM

Tạo một mô hình SVM với  $C=1.0$ , là giá trị mặc định thường được sử dụng và  $\text{random\_state}=42$  để đảm bảo tính tái sử dụng của quá trình huấn luyện.

```
# Tạo một mô hình SVM
# mô hình SVM được tạo với C=1.0, là giá trị mặc định thường được sử dụng
# và random_state=42 để đảm bảo tính tái sử dụng của quá trình huấn luyện.
svm_model = SVC(kernel='linear', C=1.0, random_state=42)

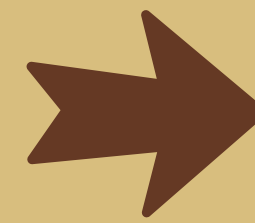
# Huấn luyện mô hình
svm_model.fit(X_train, y_train)
```

Sau đó đánh giá mô hình này dựa trên Cross-validation bằng cách chia tập train thành 5 phần bằng nhau và sử dụng 4 phần để huấn luyện và phần còn lại để đánh giá mô hình.

```
# Đánh giá mô hình bằng cross-validation trên tập train
cv_scores = cross_val_score(svm_model, X_train, y_train, cv=5)
print("Cross-Validation Scores:", cv_scores)
print("Mean CV Accuracy:", cv_scores.mean())

# Đánh giá mô hình trên tập test (đánh giá cuối cùng)
test_accuracy = svm_model.score(X_test, y_test)
print(f'Test Accuracy: {test_accuracy}')
```

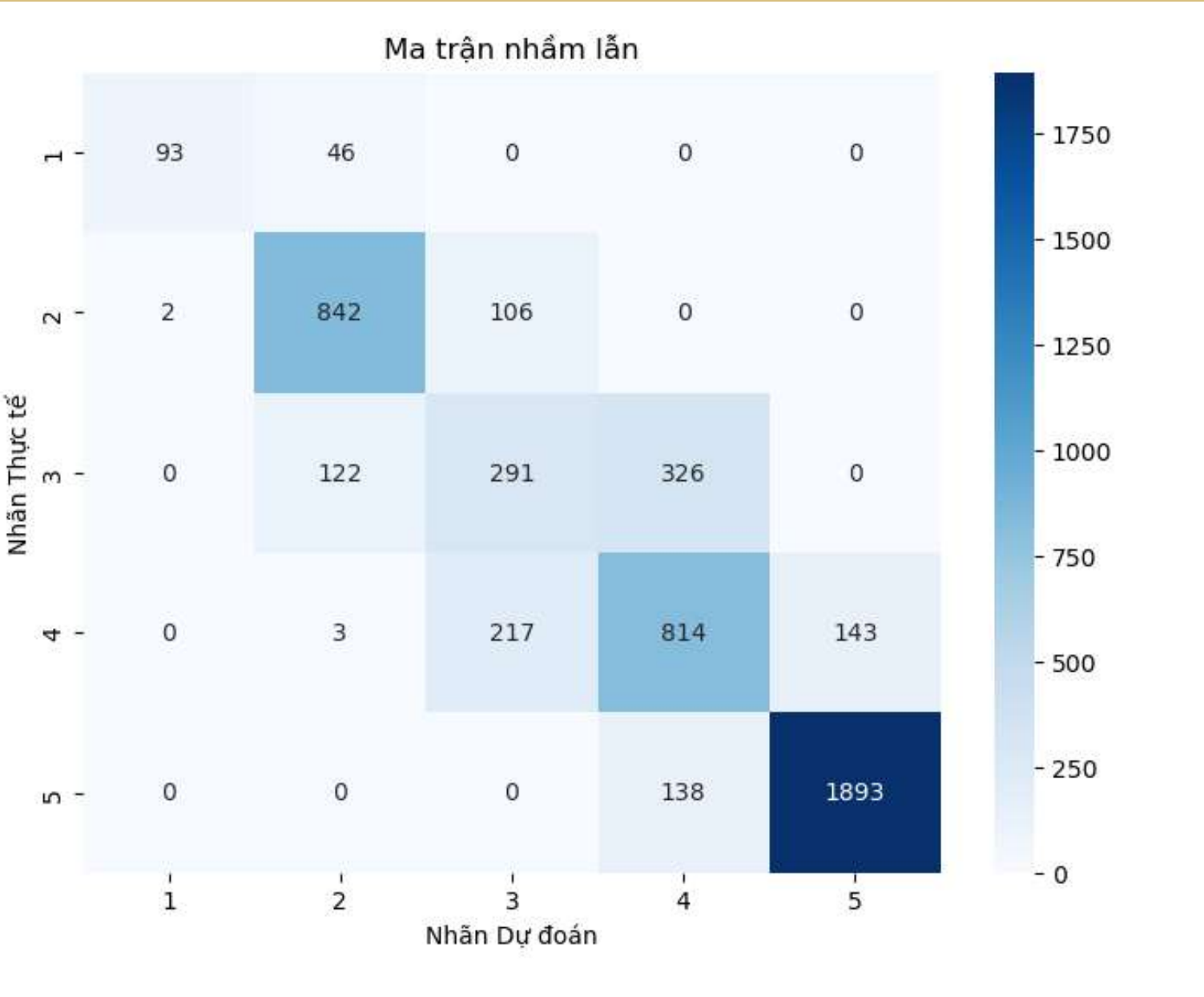
```
Cross-Validation Scores: [0.790715  0.78500497 0.78773585 0.79096326 0.78922542]
Mean CV Accuracy: 0.788728847715988
Test Accuracy: 0.7809769658459095
```



**khá tương đồng nhau  
mô hình này khá hiệu quả.**

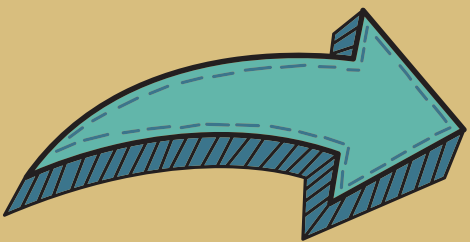
# 2. Mô hình SVM

- Để có cái nhìn rõ hơn về kết quả của mô hình, nhóm tiến hành phân tích dữ liệu từ kết quả của mô hình được thực hiện thông qua ma trận nhầm lẫn (confusion matrix) và báo cáo phân loại (classification report)



Báo cáo phân loại:


	precision	recall	f1-score	support
1	0.98	0.67	0.79	139
2	0.83	0.89	0.86	950
3	0.47	0.39	0.43	739
4	0.64	0.69	0.66	1177
5	0.93	0.93	0.93	2031
accuracy			0.78	5036
macro avg	0.77	0.71	0.74	5036
weighted avg	0.78	0.78	0.78	5036



Giữa các lớp có sự chênh lệch về độ chính xác, điều này đặc biệt rõ với lớp 1 và lớp 3.

## 2. Mô hình SVM

Mô hình có tốt hay không phụ thuộc vào việc lựa chọn C:

- Giá trị C lớn:
    - Đường biên chặt chẽ hơn
    - Mô hình có thể quá mức nhận thức
    - Có nguy cơ mô hình trở nên quá mức đồng cấu hóa và không tổng quát hóa tốt trên dữ liệu mới (overfitting).
  - Giá trị C nhỏ:
    - Tạo ra một đường biên linh hoạt hơn và chấp nhận một số lỗi đào tạo.
    - Mô hình có thể tổng quát hóa tốt hơn trên dữ liệu mới
    - Có nguy cơ mô hình trở nên quá mức "mềm dẻo" và không thể đạt được độ chính xác cao trên tập đào tạo.
- 

Để cải thiện mô hình và tinh chỉnh siêu tham số C, ta có thể sử dụng GridSearchCV từ thư viện scikit-learn.

```
# Tạo một mô hình SVM
svm = SVC(kernel='linear', random_state=42)

# Định nghĩa các giá trị C để thử nghiệm
param_grid = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}
```



**Tìm ra C phù hợp nhất**



## 2. Mô hình SVM

- Nhóm sử dụng **GridSearchCV** để tìm ra C tốt nhất và so sánh với kết quả trên.

```
# In ra giá trị C tốt nhất được chọn
print("Best C:", svm_grid_search.best_params_['C'])

# In ra độ chính xác tốt nhất trên tập kiểm tra
print("Best Accuracy:", svm_grid_search.best_score_)
```

Best C: 100

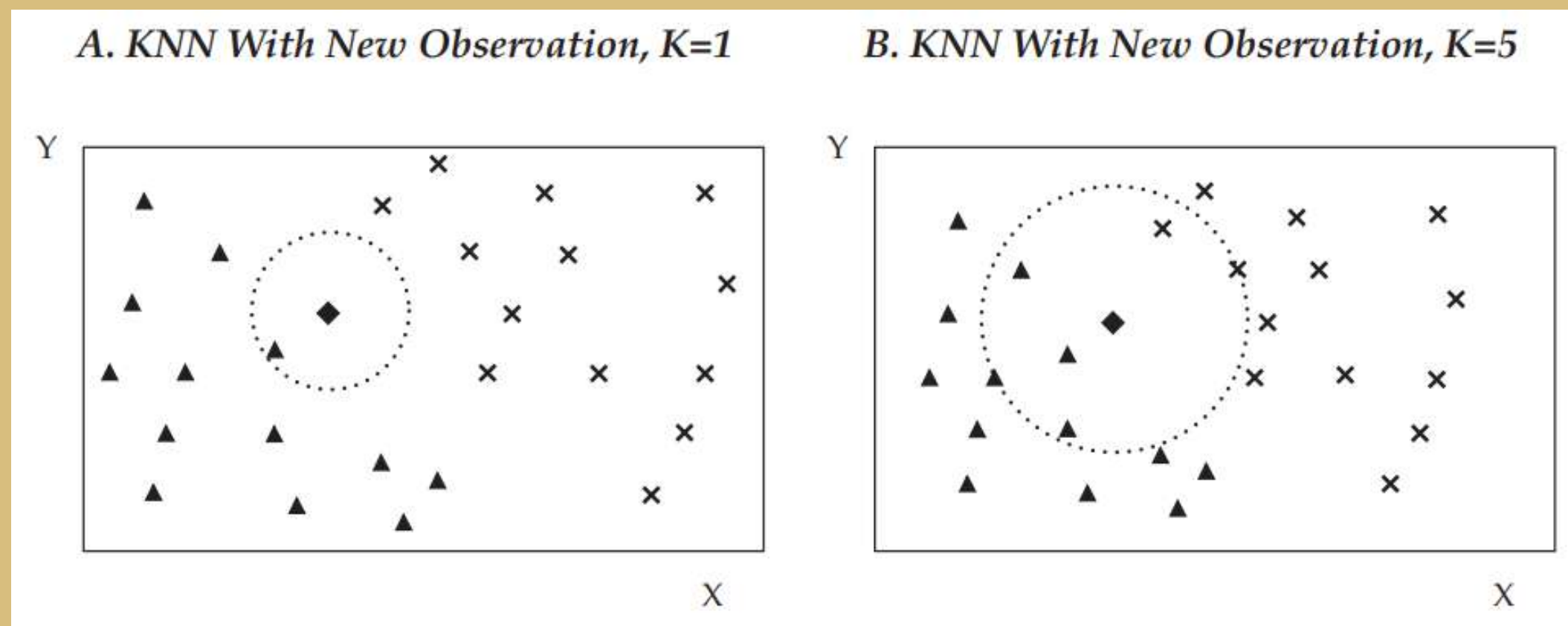
Best Accuracy: 0.7937437934458788



Dựa vào kết quả trên, C=100 và độ chính xác là 79,37% là lựa chọn tốt nhất cho mô hình SVM với dữ liệu hiện tại.

### 3. Mô hình KNN (K-Nearest Neighbors)

- Mô hình K-Nearest Neighbors (KNN) hoạt động dựa trên nguyên lý đơn giản: "Nhóm các điểm dữ liệu gần nhau trong không gian đặc trưng sẽ có cùng nhãn hoặc giá trị dự đoán". KNN không học được một mô hình tường minh mà chỉ lưu trữ dữ liệu huấn luyện.



- Cách thức hoạt động sẽ là dựa vào các giá trị điểm trong dữ liệu huấn luyện để tính toán khoảng cách điểm hiện tại có cùng nhãn với giá trị nào nhiều nhất thì đó là giá trị dự đoán của điểm hiện tại.
- Thuật toán tính khoảng cách Euclidean

$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

■ Trong công thức này:

- $p$  và  $q$  là hai điểm trong không gian  $n$  chiều.
- $p_i$  và  $q_i$  là các thành phần tương ứng của hai điểm  $p$  và  $q$ .
- $n$  là số chiều của không gian đặc trưng.

### 3. Mô hình KNN (K-Nearest Neighbors)

- Đầu tiên, nhóm em sẽ tiếp cận mô hình với  $K=5$ , sử dụng thư viện **KNeighborsClassifier**.

```
# Tạo mô hình KNN Classifier (ví dụ với k=5)
knn_model = KNeighborsClassifier(n_neighbors=5)

# Huấn luyện mô hình với dữ liệu huấn luyện
knn_model.fit(X_train, y_train)
```

- Sau đó đánh giá mô hình này dựa trên Cross-validation bằng cách chia tập train thành 5 phần bằng nhau và sử dụng 4 phần để huấn luyện và phần còn lại để đánh giá mô hình.

```
# Đánh giá mô hình bằng cross-validation trên tập train
cv_scores = cross_val_score(knn_model, X_train, y_train, cv=5)
print("Cross-Validation Scores:", cv_scores)
print("Mean CV Accuracy:", cv_scores.mean())
```

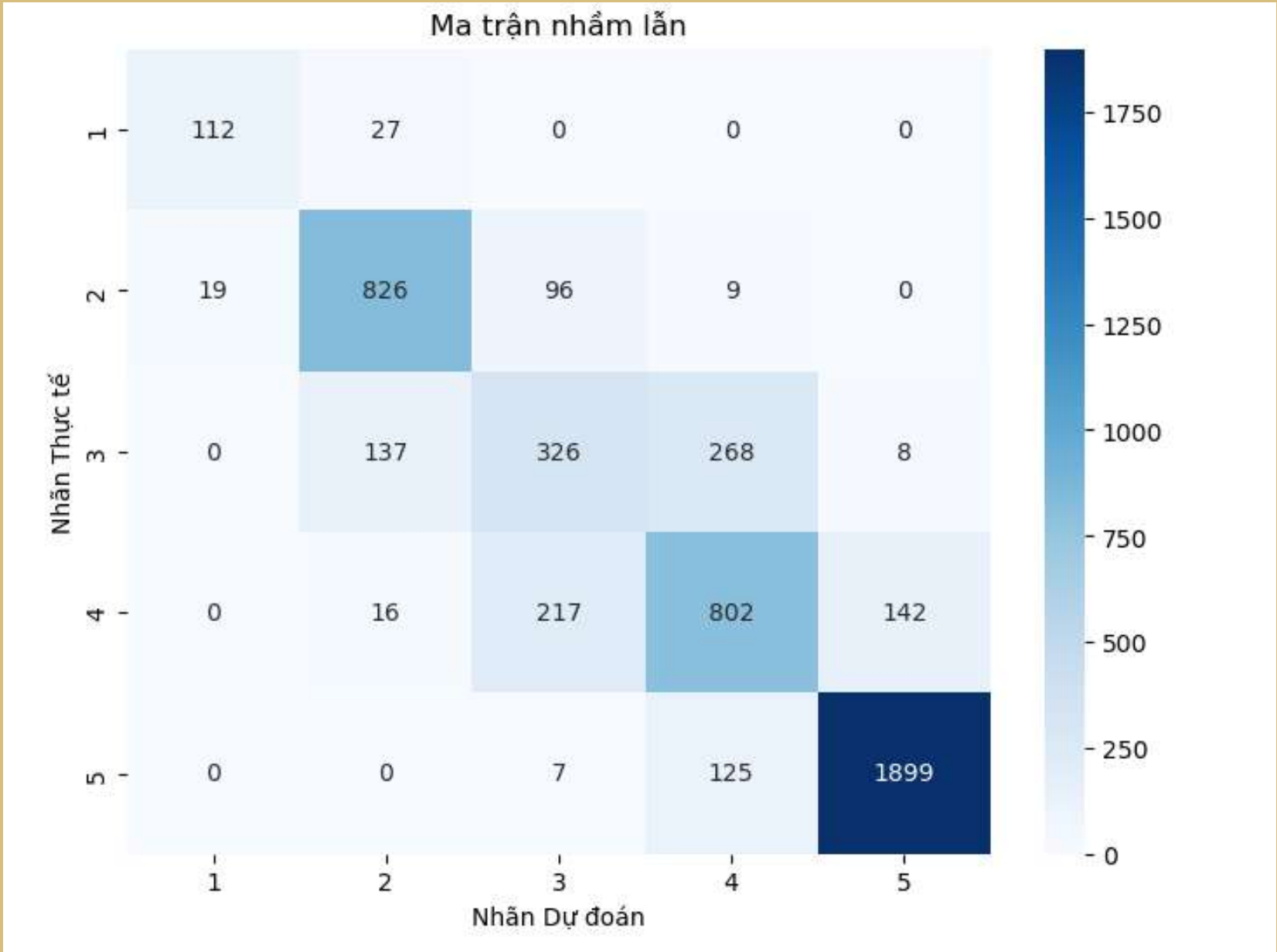
```
Cross-Validation Scores: [0.78252234 0.78003972 0.78550149 0.78699106 0.7775571 ]
Mean CV Accuracy: 0.7825223435948361
```



Kết quả của mô hình sau khi chia 5 fold và sử dụng Cross-Validation cho ra kết quả khá tương đồng. Trung bình ở 78,25%, mức này là khá tốt của 1 mô hình.

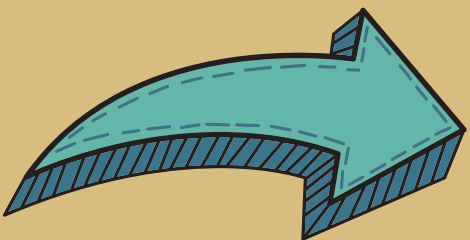
# 3. Mô hình (K-Nearest Neighbors)

- Để có cái nhìn rõ hơn về kết quả của mô hình, nhóm tiến hành phân tích dữ liệu từ kết quả của mô hình được thực hiện thông qua ma trận nhầm lẫn (confusion matrix) và báo cáo phân loại (classification report)



Báo cáo phân loại:

	precision	recall	f1-score	support
1	0.85	0.81	0.83	139
2	0.82	0.87	0.84	950
3	0.50	0.44	0.47	739
4	0.67	0.68	0.67	1177
5	0.93	0.94	0.93	2031
accuracy			0.79	5036
macro avg	0.75	0.75	0.75	5036
weighted avg	0.78	0.79	0.78	5036

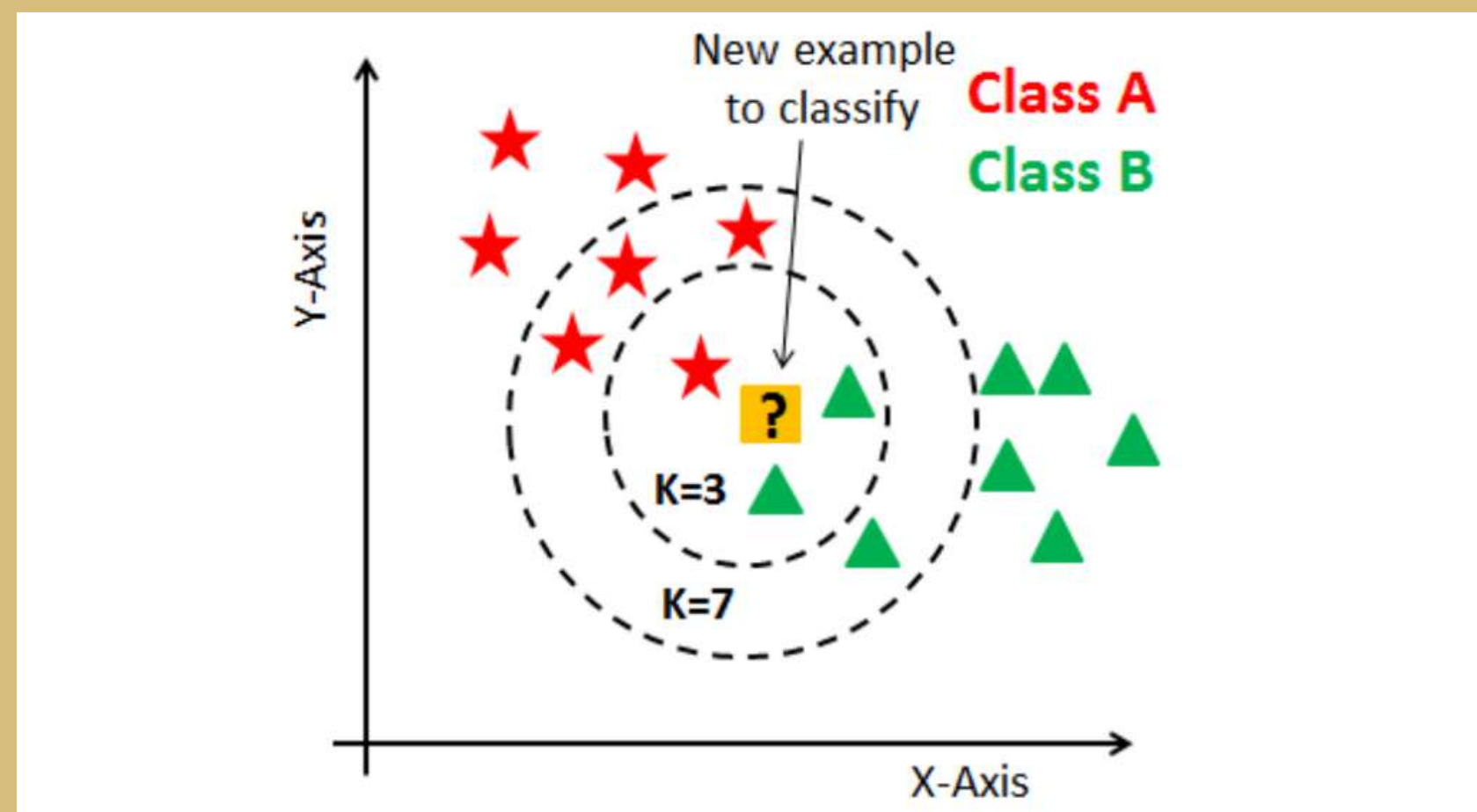


Giữa các lớp có sự chênh lệch về độ chính xác, điều này đặc biệt rõ với lớp 1 và lớp 3.



### 3. Mô hình KNN (K-Nearest Neighbors)

- Mô hình KNN có tốt hay không phụ thuộc vào việc chọn K ( n\_neighbors) phù hợp.
  - K có thể ảnh hưởng đến hiện tượng overfitting hoặc underfitting.
  - K càng lớn thì mô hình càng đơn giản nhưng có thể dẫn đến việc bỏ qua các chi tiết nhỏ.
  - K càng nhỏ thì mô hình có thể trở nên quá phức tạp và dễ bị nhiễu.
- Ví dụ K=3 thì không gian điểm lấy để tìm nhãn sẽ nhỏ hơn không gian điểm của K=7



Vì vậy, nhóm em sẽ test với  $K = [3:101]$  để tìm ra K tốt nhất

### 3. Mô hình KNN (K-Nearest Neighbors)

- Nhóm sẽ sử dụng **K-Fold Cross-validation** để chia dữ liệu thành các tập con và đánh giá hiệu suất của mô hình với mỗi giá trị K. Sau đó vẽ biểu đồ để tìm ra K tốt nhất



- Với  $K > 9$  và ta thấy độ chính xác ngày càng giảm. Vì vậy ta không cần xét với  $K > 101$

- Bên cạnh đó, nhóm sử dụng **GridSearchCV** để tìm ra K tốt nhất và so sánh với kết quả trên.

```
# Tạo mô hình KNN
knn = KNeighborsClassifier()
n_neighbors = [num for num in range(3, 101) if num % 2 != 0]
# Thiết lập các tham số cần tìm kiếm
param_grid = {'n_neighbors': n_neighbors} # Các giá trị K cần thử

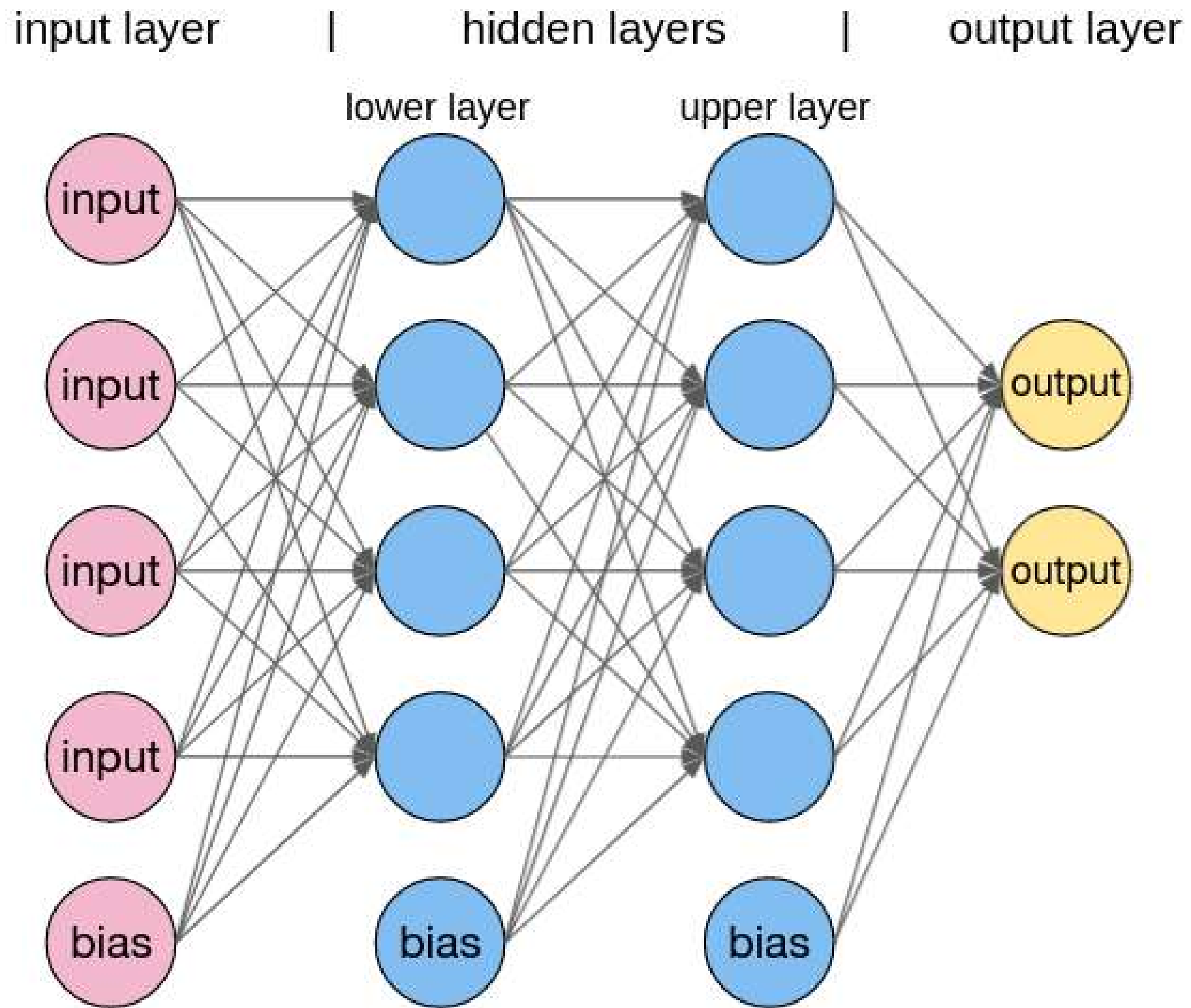
# Sử dụng GridSearchCV để tìm giá trị K tốt nhất
knn_grid_search = GridSearchCV(knn, param_grid, cv=5, scoring='accuracy')
knn_grid_search.fit(X_train, y_train)

# In ra giá trị K tốt nhất và độ chính xác tương ứng
print("Giá trị K tốt nhất:", knn_grid_search.best_params_)
print("Độ chính xác tốt nhất:", knn_grid_search.best_score_)
```

Giá trị K tốt nhất: {'n\_neighbors': 9}  
Độ chính xác tốt nhất: 0.7925024826216485

Dựa vào kết quả trên,  $K=9$  và độ chính xác là 79,25% là lựa chọn tốt nhất cho mô hình KNN với dữ liệu hiện tại.

# 4. Mô hình Multi-layer Neural Network



Mô hình Multi-layer neural network-MLP (mạng nơ-ron nhiều lớp) là tập hợp của các perceptron chia làm nhiều nhóm, mỗi nhóm tương ứng với một layer.

MLP có ba lớp chính:

- Lớp đầu vào (Input): Nhận dữ liệu đầu vào cho mạng.
- Lớp ẩn (Hidden): Có thể có một hoặc nhiều lớp ẩn. Các nơ-ron trong lớp ẩn nhận tín hiệu từ các nơ-ron ở lớp trước, thực hiện một phép tính bằng một activation function, và gửi kết quả cho các nơ-ron ở lớp tiếp theo.
- Lớp đầu ra (Output): Cung cấp kết quả của mạng.

## 4. Mô hình Multi-layer Neural Network

Đầu tiên, nhóm tiến hành xây dựng mô hình MLP với các tham số mặc định:

- hidden\_layer\_sizes=(100,)
- activation='relu'
- solver='adam'

```
# Khởi tạo mô hình MLP với các siêu tham số mặc định và max_iter = 50 (số lượng vòng lặp tối đa)
mlp_model = MLPClassifier(hidden_layer_sizes=(100,), activation='relu', solver='adam', max_iter = 50)

# Huấn luyện mô hình trên tập train
mlp_model.fit(X_train, y_train)
```

Sau đó đánh giá mô hình MLP dựa trên Cross-validation .

```
# Đánh giá mô hình bằng cross-validation trên tập train
cv_scores = cross_val_score(mlp_model, X_train, y_train, cv=5, scoring='accuracy')
print("Cross-Validation Scores:", cv_scores)
print("Mean CV Accuracy:", cv_scores.mean())

# Đánh giá mô hình trên tập test
test_accuracy = mlp_model.score(X_test, y_test)
print(f'Test Accuracy: {test_accuracy}')
```

```
Cross-Validation Scores: [0.80238332 0.79617676 0.80014896 0.80461768 0.80163853]
Mean CV Accuracy: 0.8009930486593844
Test Accuracy: 0.7998411437648928
```



- Việc chia tập train thành 4 fold và sử dụng Cross-Validation cho ra kết quả accuracy khá tương đồng nhau (xấp xỉ 80%).
- Kết quả trên tập test cũng có accuracy xấp xỉ 80%. Kết quả này là khá tốt.



# 4. Mô hình Multi-layer Neural Network

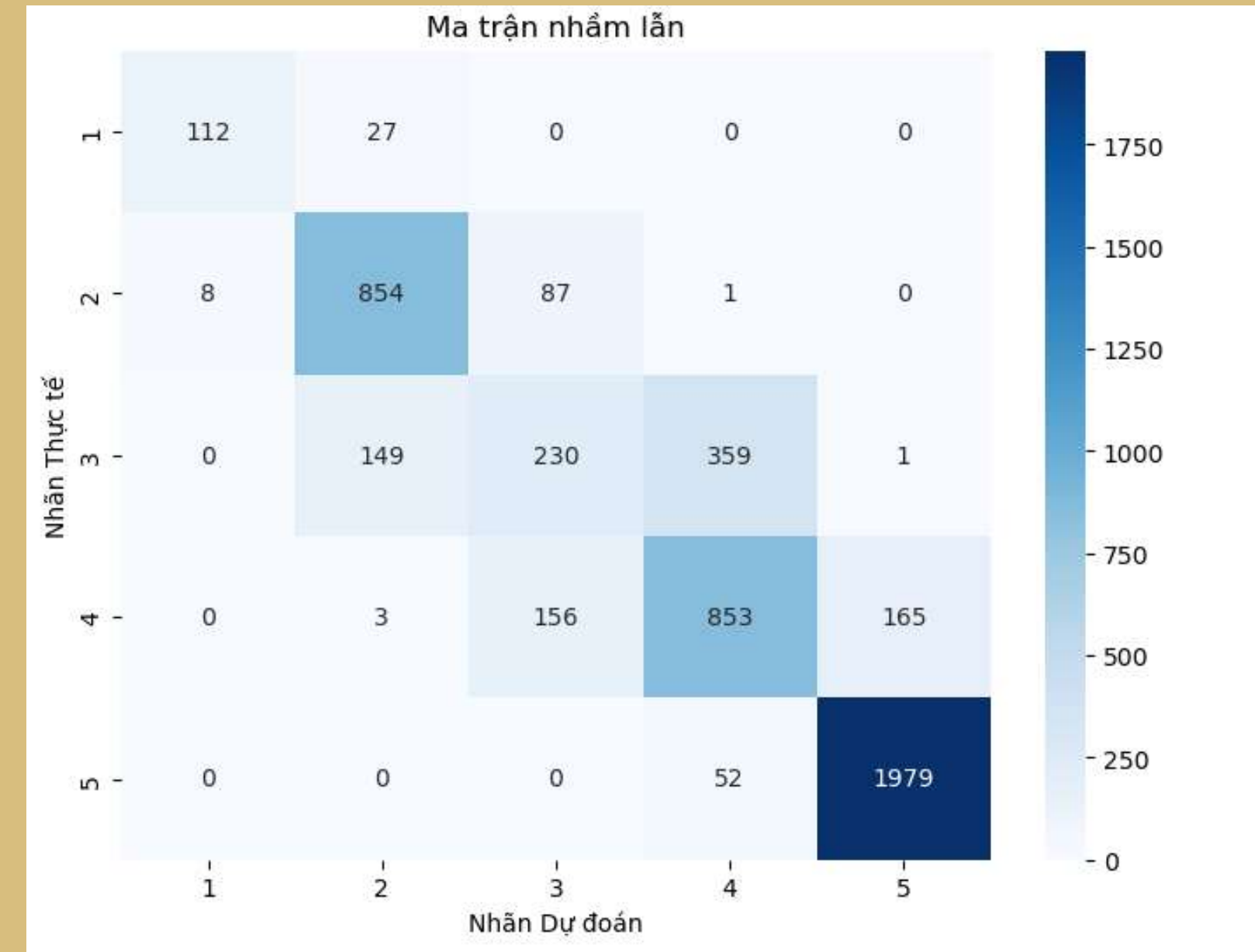
- Đánh giá bằng báo cáo phân loại

Báo cáo phân loại:				
	precision	recall	f1-score	support
1	0.93	0.81	0.86	139
2	0.83	0.90	0.86	950
3	0.49	0.31	0.38	739
4	0.67	0.72	0.70	1177
5	0.92	0.97	0.95	2031
accuracy			0.80	5036
macro avg	0.77	0.74	0.75	5036
weighted avg	0.78	0.80	0.79	5036

## Phân tích Kết Quả

- Lớp 1, Lớp 2, Lớp 4, Lớp 5: Các lớp này có Precision và Recall từ khá đến cao cho thấy mô hình có hiệu suất tốt trong việc dự đoán chất lượng không khí của các lớp trên.
- Lớp 3: Precision và Recall thấp, cho thấy khả năng dự đoán đúng trên lớp này không cao.

- Đánh giá bằng ma trận nhầm lẫn



## 4. Mô hình Multi-layer Neural Network

Các siêu tham có thể ảnh hưởng tới hiệu suất mô hình như sau:

- Hidden\_layer\_sizes: số lượng lớp ẩn và kích thước của lớp ẩn càng lớn thì mô hình càng có thể học được các mối quan hệ phức tạp hơn, nhưng cần nhiều dữ liệu hơn để huấn luyện và dễ bị overfitting và mất tính tổng quát.
- Activation: Hàm kích hoạt quyết định cách thức mà các nơ-ron trong mô hình xử lý thông tin.
- Solver: quyết định phương pháp tối ưu hóa được sử dụng để cập nhật trọng số và bias của mô hình.

Sử dụng **GridSearchCV** từ thư viện *scikit-learn* để cải thiện mô hình

Mô hình sau khi cải thiện có độ chính xác là 81,01%, ta có thể sử dụng các siêu tham số này cho mô hình MLP.

```
# Khởi tạo mô hình MLP với max_iter = 50 (số lượng vòng lặp tối đa)
mlp = MLPClassifier(max_iter = 50)

# Định nghĩa lưới các giá trị tham số cần tìm kiếm
mlp_param_grid = {
    'hidden_layer_sizes': [(100,), (5, 10, 20), (20, 20)],
    'activation': ['logistic', 'tanh', 'relu'],
    'solver': ['sgd', 'adam'],
}

# Tạo đối tượng GridSearchCV
mlp_grid_search = GridSearchCV(mlp, mlp_param_grid, cv=5, scoring='accuracy')

# Thực hiện tìm kiếm trên lưới tham số
mlp_grid_search.fit(X_train, y_train)

# In ra các siêu tham số tốt nhất
print("Best Parameters:", mlp_grid_search.best_params_)

# In ra độ chính xác tốt nhất trên tập kiểm tra
print("Best Accuracy:", mlp_grid_search.best_score_)

Best Parameters: {'activation': 'tanh', 'hidden_layer_sizes': (5, 10, 20), 'solver': 'adam'}
Best Accuracy: 0.8101290963257199
```

# Dự đoán

## Mô hình Decision Tree

```
# Dự đoán aqi sử dụng best_model vừa huấn luyện trên  
predicted_aqi = best_model_tree.predict(new_sample)  
  
print(f'Predicted AQI: {predicted_aqi}')
```

Predicted AQI: [2]

## Tạo mẫu để dự đoán

```
# Tạo sample data mới  
new_data = {  
    'no': 0.5,  
    'co': 700.3,  
    'so2': 35.1,  
    'no2': 37.3,  
    'o3': 52.7,  
    'pm2_5': 19.5,  
    'pm10': 20.2,  
    'nh3': 7.99  
}
```

```
# Chuyển sample data thành Dataframe  
new_df = pd.DataFrame([new_data])
```

```
# Chuẩn hóa sample data  
new_sample = scaler.transform(new_df)
```

## Mô hình SVM

```
# Dự đoán AQI cho mẫu dữ liệu mới  
predicted_aqi = best_svm_model.predict(new_data)  
  
print(f'Predicted AQI: {predicted_aqi}')
```

Predicted AQI: [2]

## Mô hình KNN

```
# Dự đoán aqi sử dụng best_model vừa huấn luyện trên  
predicted_aqi = best_model_knn.predict(new_sample)  
  
print(f'Predicted AQI: {predicted_aqi}')
```

Predicted AQI: [2]

## Mô hình MLP

```
# Dự đoán aqi sử dụng best_model vừa huấn luyện trên  
predicted_aqi = best_mlp_model.predict(new_sample)  
  
print(f'Predicted AQI: {predicted_aqi}')
```

Predicted AQI: [2]

**Các mô hình đều cho ra kết quả cho ra AQI = 2.**

# TỔNG KẾT

- Với bộ dữ liệu của nhóm, Mô hình cho ra kết quả tốt nhất là mô hình **Decision Tree** với accuracy từ mô hình tốt nhất tìm được là gần **83%**. Kết quả này là vì đây là bộ dữ liệu không quá phức tạp với không quá nhiều biến và mối quan hệ giữa các biến không quá phức tạp. Bên cạnh đó với các mô hình còn lại là SVM, KNN, MLP thì độ chính xác cũng ở mức khá cao và chấp nhận được (xấp xỉ 80%).
- Các mô hình có thể phân loại tốt với các mẫu không khí có chất lượng kém và rất kém ( $aqi = 4,5$ ), tuy nhiên với các mẫu không khí từ bình thường đến tốt thì các mô hình hoạt động vẫn chưa tốt lắm. Điều này có thể đến từ sự phân bố tập trung quá nhiều vào các mẫu có chất lượng kém và rất kém trong bộ dữ liệu thu thập được.
- Các yếu tố có thể ảnh hưởng đến mô hình phân loại:
  - Có nhiều yếu tố ngoại cảnh: thời tiết, nhiệt độ, độ ẩm,...
  - Bên cạnh đó, cách đo và thu thập các chỉ số cũng ảnh hưởng lớn đến kết quả của mô hình



# TỔNG KẾT

Qua quá trình tìm hiểu, nhóm đã có một sự so sánh tổng quan về Decision Tree, MLP (Multi-layer Perceptron), SVM (Support Vector Machine) và KNN (K-Nearest Neighbors) như sau:

- Decision Tree:
  - Ưu điểm:
    - Dễ hiểu và diễn giải. Có thể trực quan hóa cây quyết định.
    - Khả năng xử lý cả dữ liệu số học và phân loại.
    - Không cần nhiều tiền xử lý dữ liệu, có thể xử lý dữ liệu bị thiếu.
  - Hạn chế:
    - Dễ bị overfitting nếu cây quá sâu và không được cắt tỉa (pruning).
    - Khá nhạy cảm với việc nhận dạng và sửa chữa nhiễu trong dữ liệu.
    - Có thể không hiệu quả khi có quá nhiều biến và mối quan hệ phức tạp.
- MLP (Multi-layer Perceptron):
  - Ưu điểm:
    - Có khả năng học và ánh xạ các mô hình phức tạp.
    - Phù hợp cho việc học từ dữ liệu phi cấu trúc hoặc phi tuyến tính.
  - Hạn chế:
    - Đòi hỏi nhiều dữ liệu huấn luyện và thời gian huấn luyện lâu.
    - Dễ bị overfitting nếu không kiểm soát được các tham số (ví dụ: số lớp, số nút, hàm kích hoạt, v.v.).

# TỔNG KẾT

- SVM (Support Vector Machine):
  - Ưu điểm:
    - Hiệu quả trong không gian chiều cao.
    - Hỗ trợ phân loại tốt khi có ranh giới quyết định rõ ràng giữa các lớp.
    - Có thể sử dụng các hàm nhân (kernel) để ánh xạ dữ liệu vào không gian cao chiều.
  - Hạn chế:
    - Khó áp dụng và tinh chỉnh đối với dữ liệu lớn và không cân bằng.
    - Yêu cầu lựa chọn kernel phù hợp và tinh chỉnh siêu tham số một cách thích hợp.
- KNN (K-Nearest Neighbors):
  - Ưu điểm:
    - Dễ triển khai và không cần huấn luyện.
    - Hiệu quả với dữ liệu có cấu trúc đơn giản và không có phân cụm rõ ràng.
  - Hạn chế:
    - Độ phức tạp tính toán cao khi có nhiều điểm dữ liệu.
    - Độ chính xác thấp với dữ liệu có nhiều chiều hoặc có nhiễu.

THANK YOU  
SO MUCH!

