BÁO CÁO ĐỒ ÁN

Phân tích chất lượng không khí

Giới thiệu đề tài

- Trong bối cảnh hiện đại, ô nhiễm không khí đã trở thành một trong những thách thức môi trường nghiêm trọng nhất, ảnh hưởng trực tiếp đến sức khỏe con người và hệ sinh thái. Các chất ô nhiễm trong không khí như bụi mịn (PM2.5, PM10), các loại khí như O_3, NO_2, SO_2, và CO không chỉ góp phần gây ra các bệnh về hô hấp mà còn làm gia tăng đáng kể tỷ lệ tử vong sớm trên toàn cầu. Theo các báo cáo quốc tế, ô nhiễm không khí là một trong những nguyên nhân hàng đầu gây ra các vấn đề sức khỏe nghiêm trọng, đặc biệt là ở các khu vực đô thị đông dân cư.

Giới thiệu đề tài

- Dự án này sử dụng các chỉ số dữ liệu chính, bao gồm nồng độ các chất ô nhiễm và Chỉ số Chất lượng Không khí (AQI), được thu thập từ Thành phố Hồ Chí Minh—một khu vực đang chịu áp lực lớn từ ô nhiễm không khí tại Việt Nam. Trọng tâm chính của dự án là phân tích chất lượng không khí ở các thành phố lớn, nhằm đánh giá mức độ ô nhiễm, xác định các yếu tố đóng góp chính, và cung cấp các thông tin khoa học về xu hướng biến động chất lượng không khí.

Workflow

01
Thu thập dữ liệu

02
Khám phá dữ liệu

03
Đặt câu hỏi

04
Mô hình hóa

01

Thu thập dữ liệu

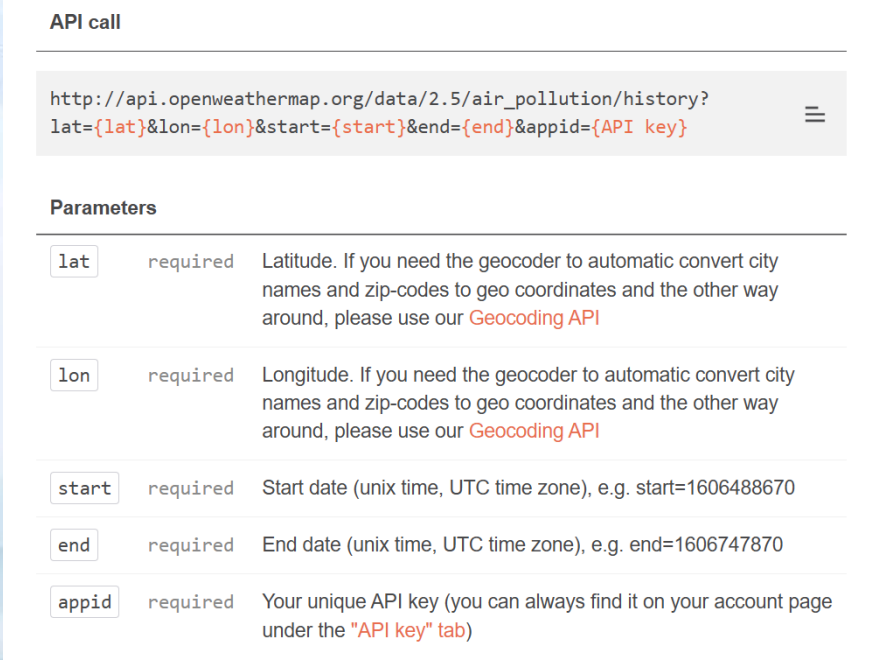
Thu thập dữ liệu

Chủ đề: Chất lượng không khí

Nguồn thu thập dữ liệu: API từ trang web

openweathermap.org

Thư viện hỗ trợ: request và json



API call

```
http://api.openweathermap.org/data/2.5/air_pollution/history?
lat={lat}&lon={lon}&start={start}&end={end}&appid={API key}
```

Parameters

`lat`	required	Latitude. If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
`lon`	required	Longitude. If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API
`start`	required	Start date (unix time, UTC time zone), e.g. start=1606488670
`end`	required	End date (unix time, UTC time zone), e.g. end=1606747870
`appid`	required	Your unique API key (you can always find it on your account page under the "API key" tab)

Thu thập dữ liệu

Các bước thu thập dữ liệu:

- Xây dựng các tham số phù hợp với yêu cầu của API.
- Thiết kế hàm với các tham số phù hợp để lấy dữ liệu từ API.
- Lưu kết quả vào file csv.

	dt	aqi	co	no	no2	o3	so2	pm2_5	pm10	nh3
0	1609459200	3	700.95	0.44	35.99	17.35	32.90	20.33	26.64	8.99
1	1609462800	3	847.82	2.46	38.04	18.06	36.24	23.32	30.54	9.37
2	1609466400	3	894.55	5.25	38.39	23.25	41.01	24.16	31.93	9.25
3	1609470000	3	827.79	6.20	36.33	33.98	43.39	23.20	30.91	8.61
4	1609473600	2	660.90	3.69	29.13	54.36	35.76	19.50	25.60	6.21
...	...	...	...	...	...	...	...	...	...	...
33812	1732910400	2	600.81	1.30	37.70	5.99	23.13	21.54	27.61	9.25
33813	1732914000	2	554.08	0.75	35.99	8.85	23.13	20.50	26.39	8.36
33814	1732917600	2	567.44	0.64	36.67	10.19	24.80	22.20	28.90	8.04
33815	1732921200	2	600.81	0.76	37.36	9.66	26.23	24.03	32.19	8.61
33816	1732924800	3	747.68	2.54	38.04	6.79	27.42	29.40	40.27	10.26

02

Khám phá dữ liệu

Khám phá dữ liệu

Dữ liệu được thu thập về chất lượng không khí từ 01/01/2021 đến 30/11/2024

Dữ liệu bao gồm 33817 dòng và 10 cột

Mỗi dòng là dữ liệu theo giờ của từng ngày

Mỗi cột là dữ liệu về thời gian ghi nhận, chỉ số chất lượng không khí (AQI) và nồng độ các chất gây ô nhiễm

Khám phá dữ liệu

Mỗi cột là dữ liệu về thời gian ghi nhận, chỉ số chất lượng không khí (AQI) và nồng độ các chất gây ô nhiễm

Qualitative name	Index	Pollutant concentration in µg/m^3					
		SO_2	NO_2	PM_{10}	$PM_{2.5}$	O_3	CO
Good	1	[0; 20)	[0; 40)	[0; 20)	[0; 10)	[0; 60)	[0; 4400)
Fair	2	[20; 80)	[40; 70)	[20; 50)	[10; 25)	[60; 100)	[4400; 9400)
Moderate	3	[80; 250)	[70; 150)	[50; 100)	[25; 50)	[100; 140)	[9400-12400)
Poor	4	[250; 350)	[150; 200)	[100; 200)	[50; 75)	[140; 180)	[12400; 15400)
Very Poor	5	⩾350	⩾200	⩾200	⩾75	⩾180	⩾15400

Kiểu dữ liệu

Cột ‘dt’ là thời gian ghi nhận số liệu, vì vậy ta có thể chuyển đổi cột ‘dt’ 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

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

Thu thập dữ liệu

Dữ liệu có bị thiếu?

Không có dữ liệu thiếu

Dữ liệu có hợp lệ không?

```
There are no negative values in the column 'co'
There are no negative values in the column 'no'
The column 'no2' contains negative values:
The column 'o3' contains negative values:
There are no negative values in the column 'so2'
There are no negative values in the column 'pm2_5'
The column 'pm10' contains negative values:
There are no negative values in the column 'nh3'
```

Sự phân bố của dữ liệu dạng số

- Missing ratio: tỉ lệ phần trăm giá trị thiếu
- Min: giá trị nhỏ nhất
- Lower quartile: tứ phân vị dưới
- Median: Trung vị
- Upper quartile: tứ phân vị trên
- Max: giá trị nhỏ nhất

	co	no	no2	o3	so2	pm2_5	pm10	nh3
missing_ratio	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
min	317.10	0.00	6.34	0.00	5.84	3.45	4.41	1.52
lower_quartile	687.60	1.65	24.33	0.02	26.70	21.43	28.80	6.02
median	1028.06	9.28	33.24	4.34	38.15	40.09	51.15	8.61
upper_quartile	1762.39	32.63	45.93	31.83	56.74	79.43	97.37	12.92
max	18585.21	393.39	213.86	446.32	270.84	936.13	1034.27	186.44

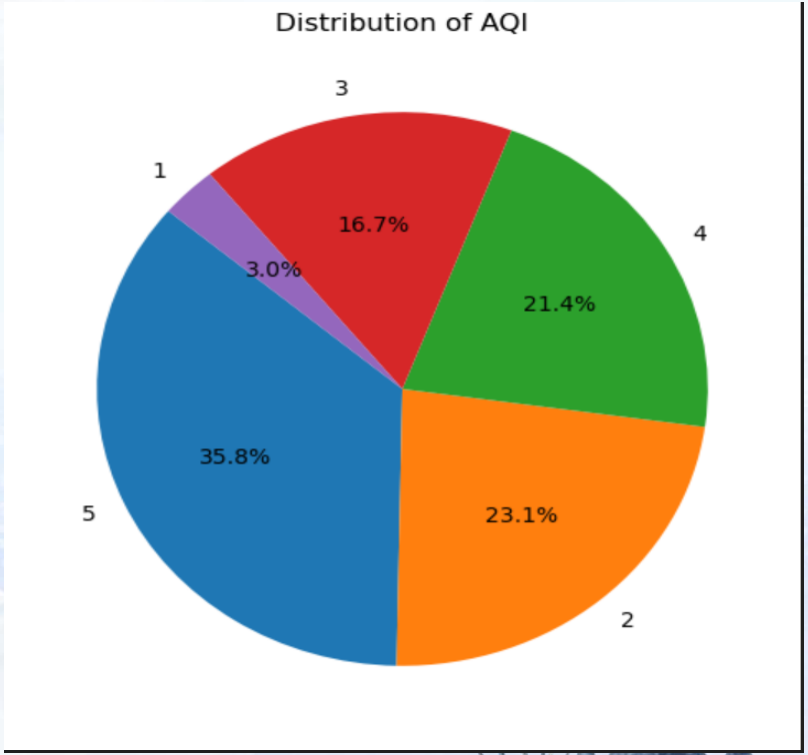
Sự phân bố của dữ liệu dạng category

- Phần trăm dữ liệu thiếu
- Số lượng giá trị (không tính giá trị thiếu)
- Phần trăm của mỗi giá trị (sắp xếp giảm dần)

	aqi
missing_ratio	0.0
num_diff_vals	5
value_ratios	{5: 12108, 2: 7815, 4: 7228, 3: 5637, 1: 1025}

Sự phân bố của dữ liệu dạng category

- Nhìn vào biểu đồ, ta có thể thấy được chất lượng không khí rất kém chiếm tỉ lệ cao nhất (35,8%),
- Tỉ lệ chất lượng không khí ở mức khá cao hơn mức trung bình và kém
- Tỉ lệ chất lượng không khí ở mức tốt rất thấp (16,7%)



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

Câu 1: Nồng độ các chất ô nhiễm thay đổi như thế nào qua các năm?

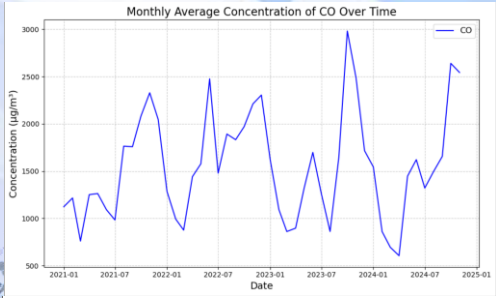
Ý Nghĩa: Mang lại những hiểu biết quan trọng về tình hình ô nhiễm biến đổi trong các khoảng thời trong vòng ba năm qua. Nó giúp ta xác định những thời điểm mức độ ô nhiễm cao có thể gây hại hơn cho sức khỏe.

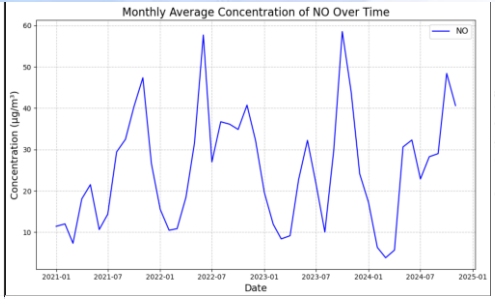
	year	month	aqi	co	no	no2	o3	so2	pm2_5	pm10	nh3	date
0	2021	1	4.280556	1122.597097	11.433903	44.051014	45.902986	50.500069	59.328667	75.461500	11.303486	2021-01-01
1	2021	2	4.419643	1215.552619	12.021652	48.128125	60.039702	56.046964	65.230685	85.290045	11.919211	2021-02-01
2	2021	3	3.178763	759.386210	7.332124	32.517231	48.422527	40.621694	29.149046	40.530349	7.951586	2021-03-01
3	2021	4	3.965278	1250.529625	18.077972	40.794903	29.289625	52.287833	51.501944	64.054792	9.540597	2021-04-01
4	2021	5	3.961022	1262.420269	21.518589	34.826022	20.992124	45.914449	50.046599	61.866183	9.284261	2021-05-01
5	2021	6	3.790278	1091.096056	10.646722	28.021750	19.260000	27.931236	43.780333	52.852778	7.975917	2021-06-01
6	2021	7	3.413978	982.802849	14.389113	25.890847	16.743145	30.560255	38.855054	46.780054	7.161344	2021-07-01
7	2021	8	3.588710	1763.566075	29.429543	34.926384	12.460860	41.675390	85.535134	97.035457	10.793938	2021-08-01
8	2021	9	3.951389	1757.768403	32.486431	34.015042	11.295069	44.731125	78.854333	91.447194	10.395375	2021-09-01
9	2021	10	4.069892	2082.089395	40.431438	39.610833	9.457715	50.863185	89.465981	104.233454	11.662191	2021-10-01
10	2021	11	4.547222	2328.519097	47.380181	49.478569	11.037236	63.460361	102.558014	120.623639	12.825750	2021-11-01
11	2021	12	4.391129	2046.063347	26.581868	48.635524	29.664556	50.329530	93.125887	115.153091	14.275027	2021-12-01
12	2022	1	4.384722	1286.133306	15.512389	46.403486	48.799125	54.165611	62.935236	81.082597	11.328500	2022-01-01
13	2022	2	3.598765	992.106682	10.474815	41.863858	42.751667	50.746404	39.685602	52.230201	10.147994	2022-02-01
14	2022	3	3.301075	875.304140	10.866465	36.418723	43.575645	44.917755	37.572191	49.904341	8.083737	2022-03-01

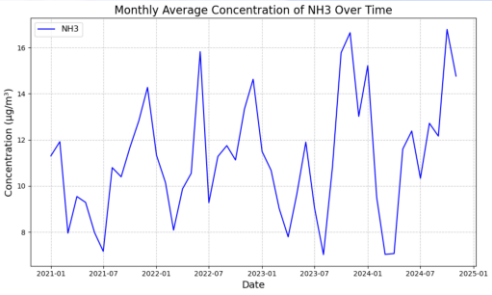
Các bước trả lời câu hỏi:

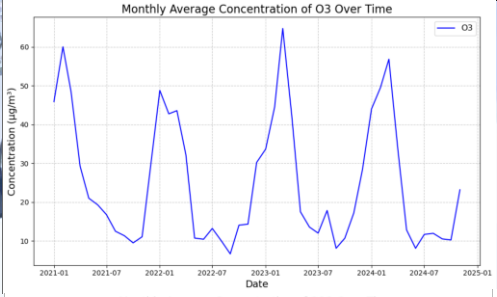
1. Tạo một cột ‘month’ để ghi lại các tháng của mỗi năm
2. Gom dữ liệu theo tháng và tính hàm lượng trung bình
3. Vẽ biểu đồ đường

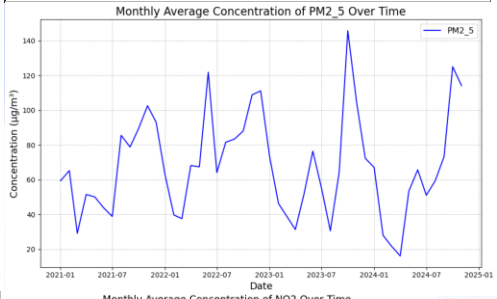
Câu 1: Nồng độ các chất ô nhiễm thay đổi như thế nào qua các năm?

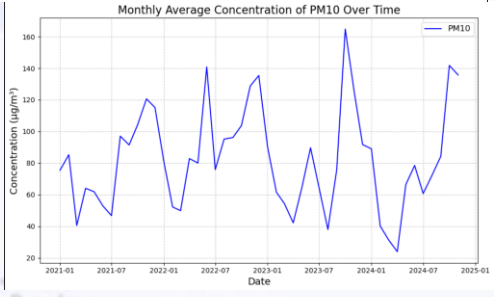


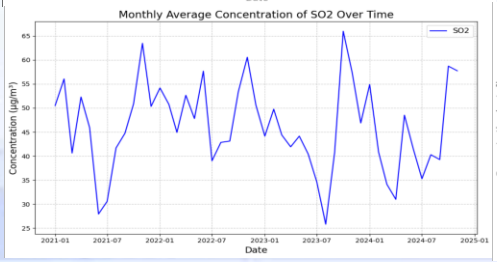


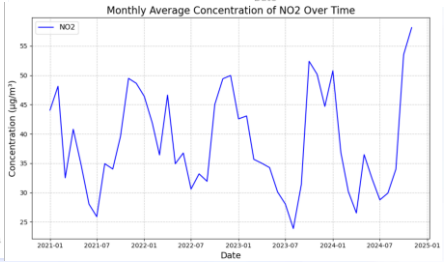












Nhìn chung, nồng độ các chất ô nhiễm có xu hướng giảm qua các năm. Cụ thể, nồng độ PM2.5, PM10, CO, NO, và SO2 đều có xu hướng giảm đáng kể từ năm 2021 đến 2024. Tuy nhiên, nồng độ O3 và NH3 có sự biến động và không có xu hướng giảm rõ ràng.

Câu 2: Mức độ tương quan của các yếu tố ảnh hưởng đến AQI là gì?

Ý Nghĩa: Xác định những yếu tố nào có tác động đáng kể nhất đến AQI, mối quan hệ này là tích cực hay tiêu cực, và mức độ mạnh yếu của các mối quan hệ đó. Thông tin này rất quan trọng để hiểu động lực chất lượng không khí, hỗ trợ quyết định chính sách và định hướng các nỗ lực giảm ô nhiễm, cải thiện sức khỏe cộng đồng.

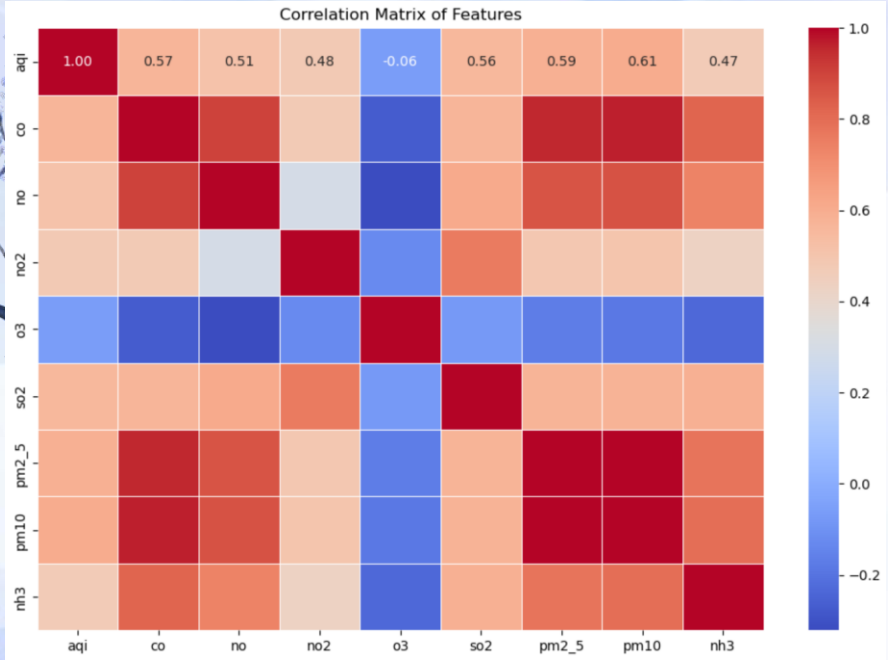
pm10	0.605171
pm2_5	0.590828
co	0.567394
so2	0.558435
no	0.513296
no2	0.482355
nh3	0.472291
o3	-0.061197

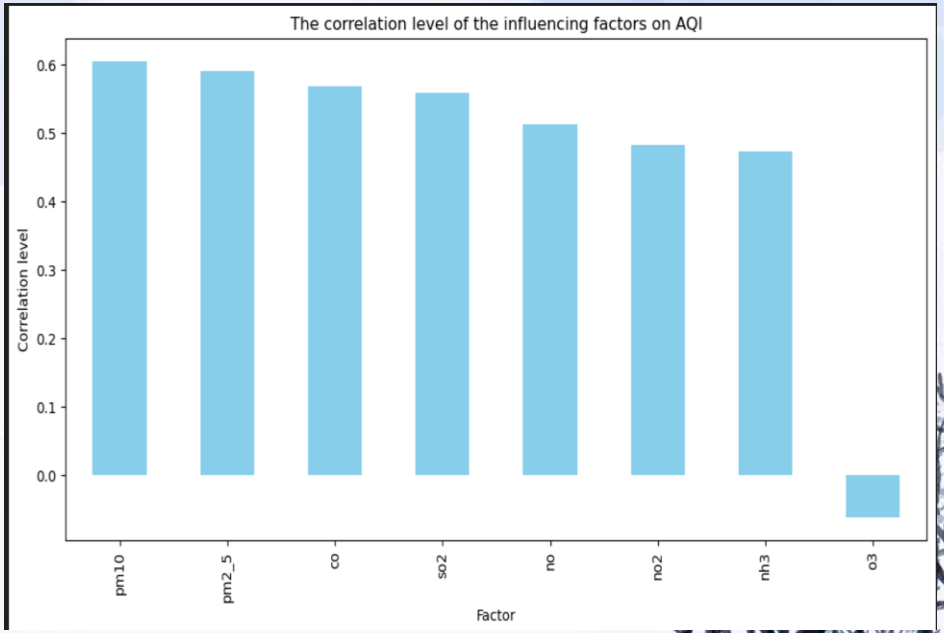
Các bước trả lời câu hỏi:

1. Tạo một dataframe mới và loại bỏ các cột không liên quan đến aqi.
2. Sử dụng hàm corr() để tính ma trận tương quan
3. Trực quan hóa kết quả

Câu 2: Mức độ tương quan của các yếu tố ảnh hưởng đến AQI là gì?

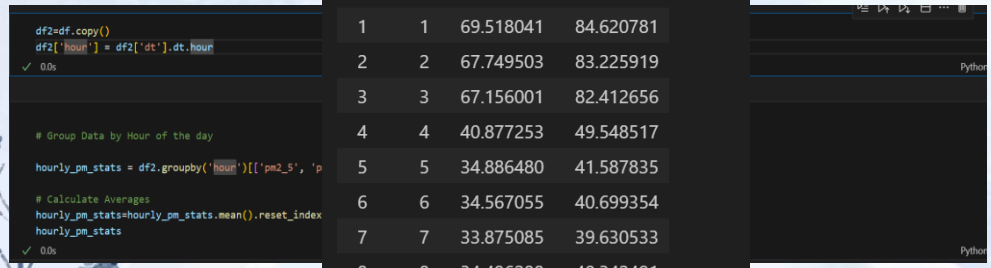
AQI cho thấy tương quan dương mạnh với PM2.5 (0.59), PM10 (0.61) và CO (0.57), nhấn mạnh rằng các chất ô nhiễm này là các yếu tố chính gây ra chất lượng không khí kém





Câu 3: Nồng độ trung bình của PM2.5 và PM10 theo từng giờ trong ngày là bao nhiêu?

Ý nghĩa: Xác định những thời điểm trong ngày có mức độ ô nhiễm cao. Phân tích này có thể chỉ ra những khoảng thời gian có không khí tương đối sạch hơn, cung cấp thông tin giá trị cho các khuyến cáo về sức khỏe cộng đồng, kế hoạch hoạt động ngoài trời.



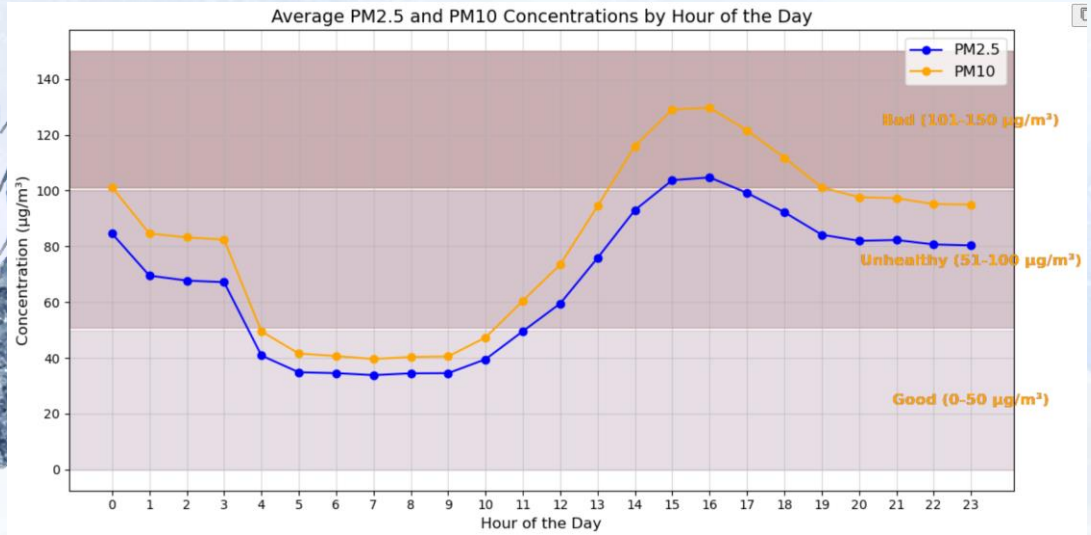
	hour	pm2_5	pm10
0	0	84.596820	101.138935
1	1	69.518041	84.620781
2	2	67.749503	83.225919
3	3	67.156001	82.412656
4	4	40.877253	49.548517
5	5	34.886480	41.587835
6	6	34.567055	40.699354
7	7	33.875085	39.630533
8	8	34.496288	40.342491
9	9	34.567630	40.575302
10	10	39.532988	47.340802
11	11	49.585415	60.509255
12	12	59.518971	73.497871
13	13	75.853400	94.436267
14	14	93.006402	115.990128
15	15	103.719716	129.012697

Để trả lời câu hỏi này, ta sẽ:

1. Gom nhóm dữ liệu theo giờ trong ngày
2. Tính toán các giá trị trung bình
3. Trực quan hóa kết quả

Câu 3: Nồng độ trung bình của PM2.5 và PM10 theo từng giờ trong ngày là bao nhiêu?

Trong những giờ đêm và sáng sớm (0:00–8:00), cả nồng độ PM2.5 và PM10 đều ở mức thấp, với PM2.5 dao động từ 33 đến 85 µg/m³ và PM10 từ 39 đến 101 µg/m³, đạt mức thấp nhất vào khoảng 4:00 đến 8:00. Khi các hoạt động ban ngày tăng lên, nồng độ PM2.5 và PM10 tăng đáng kể vào buổi sáng và đầu buổi chiều (9:00–13:00), với PM2.5 trung bình từ 34 đến 76 µg/m³ và PM10 từ 40 đến 94 µg/m³. Đỉnh điểm xảy ra vào cuối buổi chiều (14:00–17:00), khi PM2.5 đạt tới 105 µg/m³ và PM10 gần 130 µg/m³



Nguyên nhân: sự gia tăng khí thải từ phương tiện và công nghiệp kết hợp với các điều kiện khí quyển giữ lại các chất ô nhiễm. Vào buổi tối và đêm (18:00–23:00), nồng độ dần giảm xuống còn 80–92 µg/m³ đối với PM2.5 và 95–111 µg/m³ đối với PM10 khi các hoạt động giảm và nhiệt độ mát mẻ giúp phân tán các chất ô nhiễm

Câu 4: Chỉ số chất lượng không khí (AQI) thay đổi như thế nào theo tuần và theo giờ?

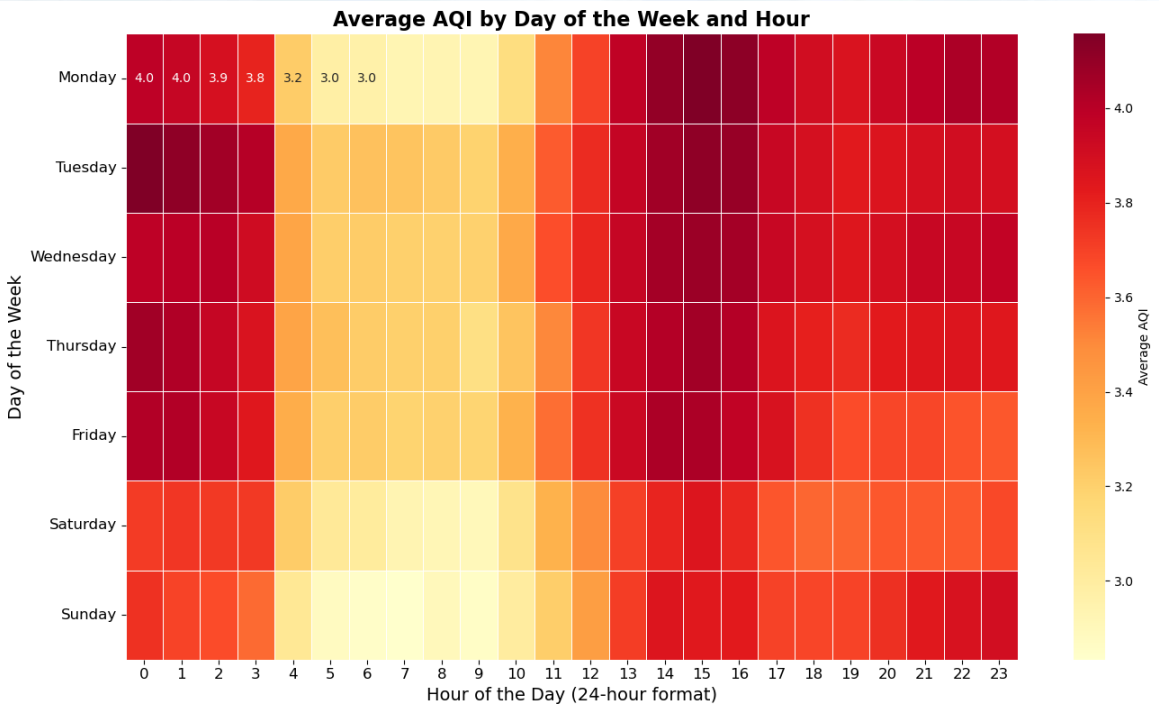
Ý nghĩa: Giúp ta xác định tình trạng chất lượng không khí theo từng khung giờ của các ngày trong tuần. Liệu các ngày cuối tuần có chất lượng không khí tốt hơn so với các ngày trong tuần hay không, hoặc liệu những giờ nhất định, như sáng sớm hoặc tối muộn, có mức AQI cao hay thấp ổn định.

hour	0	1	2	3	4	5	6	7	8	9	...	14	15	
day_of_week														
Monday	3.984848	3.950739	3.886700	3.797030	3.216749	2.980296	2.965517	2.931034	2.935961	2.931034	...	4.103448	4.157635	4
Tuesday	4.152709	4.113861	4.064356	4.009901	3.371287	3.222772	3.267327	3.252475	3.227723	3.193069	...	4.069307	4.113861	4
Wednesday	3.985149	3.995025	4.000000	3.915423	3.388060	3.213930	3.218905	3.205000	3.199005	3.199005	...	4.054726	4.084577	4
Thursday	4.064677	4.024752	3.955446	3.871287	3.396040	3.277228	3.217822	3.202970	3.202970	3.108911	...	4.014851	4.064356	4
Friday	4.019704	4.019704	3.945813	3.842365	3.354680	3.206897	3.216749	3.187192	3.197044	3.182266	...	4.034483	4.034483	3
Saturday	3.719212	3.735000	3.725000	3.725000	3.220000	3.030000	3.010000	2.935000	2.920000	2.905000	...	3.795000	3.855000	3
Sunday	3.753769	3.696970	3.666667	3.585859	3.040404	2.878788	2.853535	2.833333	2.898990	2.858586	...	3.853535	3.838384	3

Để trả lời câu hỏi này, ta sẽ:

1. Trích xuất ngày trong tuần và giờ từ cột datetime
2. Tạo một dataframe avg_aqi_by_day_hour để tính toán AQI trung bình
3. Trực quan hóa kết quả bằng heatmap

Câu 4: Chỉ số chất lượng không khí (AQI) thay đổi như thế nào theo tuần và theo giờ?



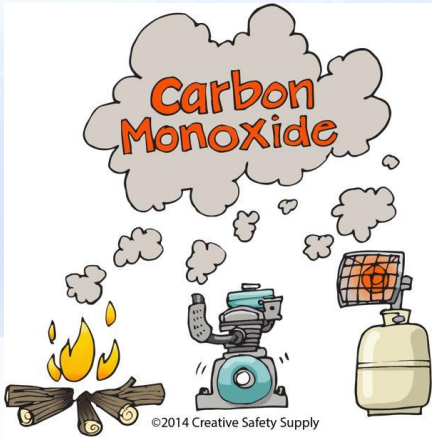
Chất lượng không khí trong suốt tuần, dựa trên giá trị AQI, chủ yếu dao động từ 3.2 đến 4.2, với mức trung bình và kém chiếm ưu thế. Trong các ngày trong tuần, đặc biệt từ sáng đến chiều, AQI có xu hướng cao hơn, dao động từ 3.8 đến 4.2, phản ánh mức độ ô nhiễm không khí ở mức trung bình đến kém, chủ yếu do ảnh hưởng của giao thông và các hoạt động công nghiệp. Tuy nhiên, vào cuối tuần, đặc biệt là vào Chủ nhật, chất lượng không khí có thể cải thiện một chút, với AQI dao động từ 2.87 đến 3.9, có thể là do giao thông và các hoạt động công nghiệp giảm.

Câu 5: Nồng độ khí CO thay đổi như thế nào theo mùa?

Ý nghĩa: Giúp ta nắm được tình trạng khí CO thay đổi như thế nào qua các tháng và mùa. Từ đó, ta có thể đưa ra các biện pháp kiểm soát ô nhiễm, xác định các nguồn ô nhiễm tiềm ẩn, và lập kế hoạch hành động ngay lập tức để giảm thiểu rủi ro đối với sức khỏe cộng đồng.

Vậy khí CO là gì? Ảnh hưởng của nó ra sao?

Carbon monoxide (CO) là một khí không màu, không mùi, được sinh ra do sự cháy không hoàn toàn của các vật liệu chứa carbon. Trong bối cảnh chất lượng không khí và giám sát môi trường, CO được sử dụng như một chỉ số để theo dõi mức độ ô nhiễm không khí. Mức độ CO cao có thể chỉ ra chất lượng không khí kém và gây ra các rủi ro về sức khỏe đối với con người, đặc biệt ảnh hưởng đến tim và phổi.



Câu 5: Nồng độ khí CO thay đổi như thế nào theo mùa?

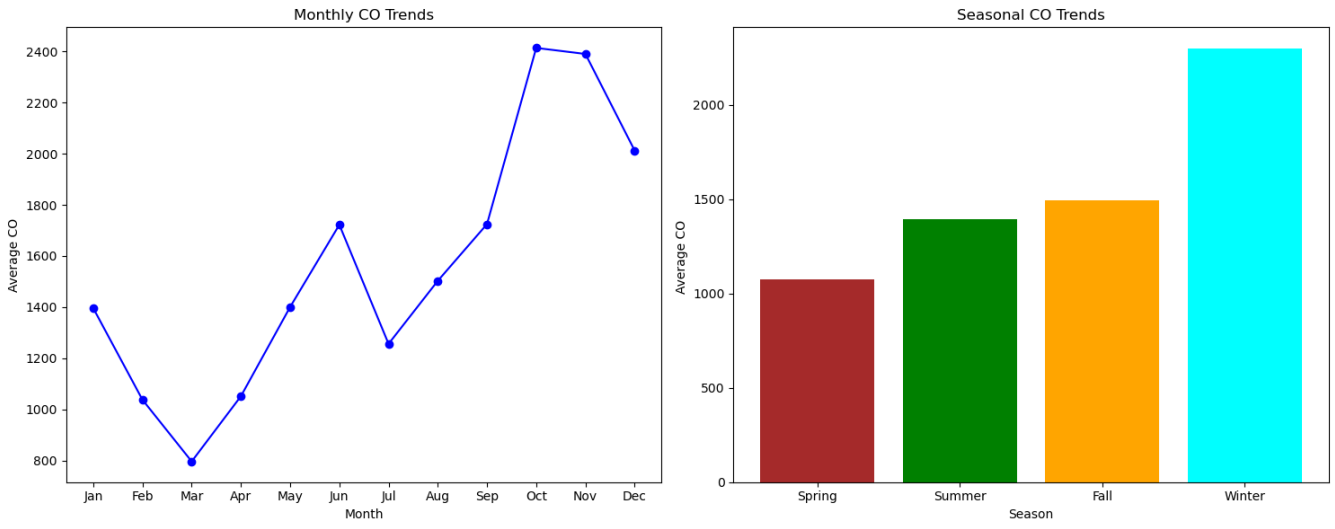
Các bước trả lời câu hỏi:

1. Tạo cột Tháng và Mùa trong dataframe
2. Tạo một dataframe mới nhóm theo tháng và mùa, sau đó tính toán trung bình CO
3. Trực quan hóa kết quả

```
1     1395.973881
2     1037.843005
3      796.590654
4     1051.677174
5     1400.642853
6     1722.237707
7     1256.502105
8     1502.841856
9     1724.675653
10    2413.710708
11    2390.121064
12    2012.554991
Name: co, dtype: float64
```

Câu 5: Nồng độ khí CO thay đổi như thế nào theo mùa?

- Vào mùa xuân, nồng độ CO thấp nhất ở mức 1076.43, có thể do thời tiết ấm hơn và giảm các hoạt động công nghiệp hoặc giao thông.
- Trong mùa hè, nồng độ CO tăng lên 1391.65, phản ánh sự gia tăng nhẹ trong các hoạt động giao thông và công nghiệp
- Mùa thu thấy nồng độ CO cao hơn nữa, đạt 1492.90, có thể do các hoạt động nông nghiệp như đốt đồng và các điều kiện khí quyển hạn chế sự phân tán chất ô nhiễm.
- Mùa đông chứng kiến nồng độ CO cao nhất, đạt 2296.51, có thể do sự gia tăng sản xuất và sự tích tụ chất ô nhiễm trong không khí lạnh, đặc và dày.



04
Mô hình hóa dữ liệu

Bài toán: phân loại chất lượng không khí dựa vào nồng độ các chất ô nhiễm

- **Mục tiêu và lựa chọn đặc trưng:**

Phân loại chất lượng không khí (aqi) dựa trên các chỉ số CO, NO, NO2, NH3, PM10, PM2.5

```
#  features (X)
X = data[['no', 'co', 'so2', 'no2', 'pm2_5', 'pm10', 'nh3']]
#target variable (y)
y = data['aqi']
```

- **Chuẩn hóa đặc trưng**

```
# Feature Scaling (Standardisation)
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

- Khi khoảng giá trị giữa hai 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

➢ Phân chia tập dữ liệu:

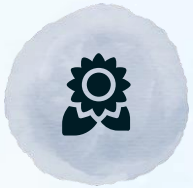
- Chia thành hai tập: train và test
- Mục đích: Nếu không chia dữ liệu mà sử dụng toàn bộ dữ liệu để huấn luyện, mô hình có thể học "quá khớp" (overfitting), không tổng quát hóa được cho dữ liệu mới.
- Tập kiểm tra giúp đánh giá xem mô hình có đang học từ dữ liệu một cách tổng quát hay chỉ học "nhớ" 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.

```
# divide the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Huấn luyện mô hình

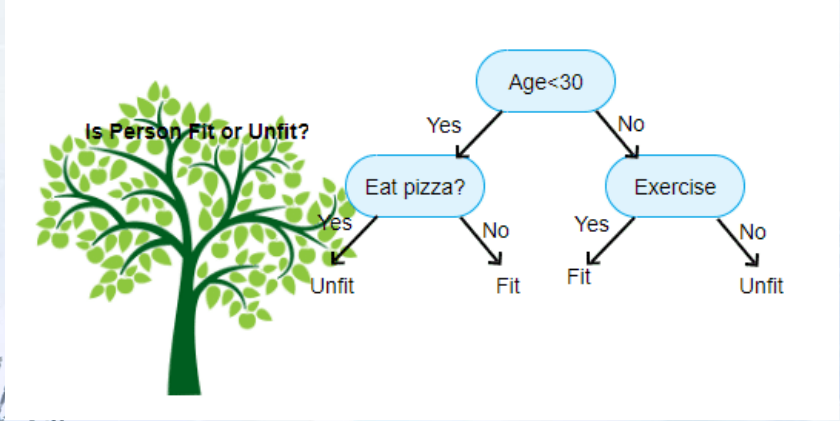
Decision Tree

Random Forest



SVM

1. Cây quyết định



- Cây quyết định là mô hình học máy được sử dụng cho các tác vụ phân loại.
- Nó hoạt động bằng cách chia dữ liệu thành các tập hợp con dựa trên các giá trị đặc điểm, tạo ra cấu trúc cây phân cấp. Mỗi nút bên trong biểu diễn một đặc điểm hoặc thuộc tính, mỗi nhánh tương ứng với một quy tắc quyết định và mỗi nút lá biểu diễn một nhãn lớp.
- Ở mỗi bước, mô hình chọn đặc điểm và tiêu chí chia tách tốt nhất để phân tách các lớp, thường sử dụng các số liệu như Gini hoặc Entropy (Information Gain) để đánh giá chất lượng của các lần chia tách.

1. Cây quyết định

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

```
# Init Decision Tree
decision_tree_model = DecisionTreeClassifier(criterion='entropy', max_depth=5, min_samples_split=2, min_samples_leaf=1)

# train Decision Tree model
decision_tree_model.fit(X_train, y_train)
```

- Kết quả :

```
Accuracy ( using CV ): [0.78262477 0.79057301 0.78151571 0.78262477 0.78539741]
```

```
Accuracy (on test set): 0.7826408398639657
```

- 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 78%, mức này là khá tốt của 1 mô hình. Kết quả cho ra ở tập test cũng khác tương đồng (xấp xỉ 78%)

1. Cây quyết định

- Để đánh giá tốt hơn mô hình, ta tiến hành phân tích thông qua classification report

	precision	recall	f1-score	support
1	0.88	0.94	0.91	193
2	0.84	0.94	0.89	1545
3	0.56	0.36	0.44	1151
4	0.60	0.65	0.63	1446
5	0.91	0.95	0.93	2428
accuracy			0.78	6763
macro avg	0.76	0.77	0.76	6763
weighted avg	0.77	0.78	0.77	6763

- Lớp 1,2,5 cho ra kết quả khá tốt.
- Lớp 3 và 4: kết quả của các lớp này khá thấp. Nguyên nhân là có thể là do sự phân bố không đồng dều dữ liệu giữa các lớp.

1. Cây quyết định

- Để mô hình hoạt động hiệu quả hơn, ta sẽ sử dụng GridSearchCV để tìm ra các siêu tham số phù hợp.

```
# Init Decision Tree
DT_model = DecisionTreeClassifier()

# Define the grid of hyperparameters to search
param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [ 5, 10, 15, 20,25],
    'min_samples_split': [1, 2, 5, 15, 25],
    'min_samples_leaf': [1, 2, 3, 4, 5]
}

#  Perform grid search with cross-validation
gridcv_decision_tree = GridSearchCV(DT_model, param_grid, cv=5, scoring='accuracy')
gridcv_decision_tree.fit(X_train, y_train)

print("Best Parameters:", gridcv_decision_tree.best_params_)
print("Best Accuracy:", gridcv_decision_tree.best_score_)
```

✓ 1m 5.7s

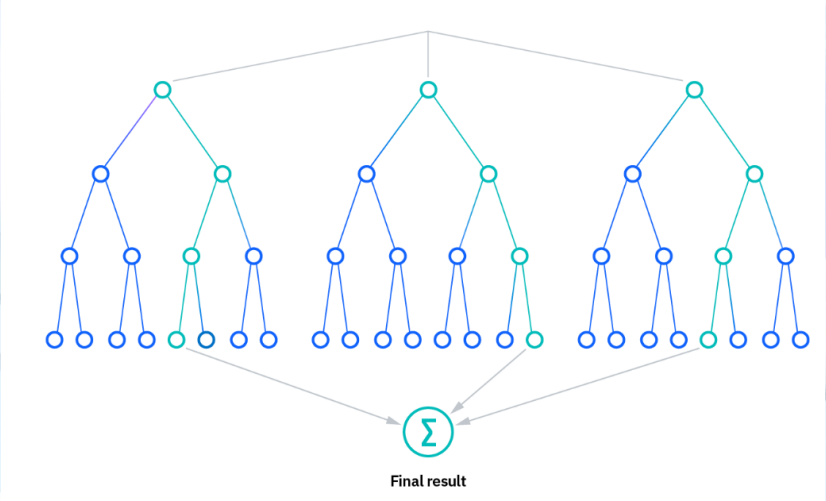
```
Best Parameters: {'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 2}
Best Accuracy: 0.7943438077634011
```

- Với các tham số như gini, độ sâu tối đa: 10, số lượng mẫu tối thiểu để chia 1 lá là 2 và số lượng mẫu tối thiêu của 1 lá là 2 thì ta có thể thu được kết quả tốt hơn so với những lần trước (79%).

2. Random Forest

- Là một thuật toán ensemble, tức là nó kết hợp nhiều mô hình đơn giản (trong trường hợp này là các cây quyết định) để tạo ra một mô hình mạnh mẽ hơn.
- Cách hoạt động:

1. Tạo ra nhiều cây quyết định, mỗi cây được xây dựng trên một tập dữ liệu con ngẫu nhiên được rút ra từ tập dữ liệu gốc (bootstrap sampling).
2. Tại mỗi nút, chỉ chọn một tập hợp con ngẫu nhiên các thuộc tính để tìm thuộc tính tốt nhất để phân chia dữ liệu.
3. Để đưa ra dự đoán, mỗi cây trong rừng sẽ đưa ra một dự đoán và dự đoán cuối cùng sẽ được quyết định bằng cách bỏ phiếu đa số.



2. Random Forest

- Xây dựng một mô hình đơn giản và sử dụng Cross-Validation để chia dữ liệu thành năm phần, sử dụng 4 phần để huấn luyện và phần còn lại để đánh giá

```python
# Init
rf_model = RandomForestClassifier(random_state=42)
# train Decision Tree model
rf_model.fit(X_train, y_train)

# Cross-validation on train set
cv_scores = cross_val_score(rf_model, X_train, y_train, cv=5)
print("Cross-Validation Scores:", cv_scores)

#  Predict on the test set
test_accuracy = rf_model.score(X_test, y_test)
print(f'Test Accuracy: {test_accuracy}')
```

✓ 16.1s

```
Cross-Validation Scores: [0.80924214 0.82107209 0.81774492 0.81441774 0.8168207 ]
Test Accuracy: 0.8181280496820937
```

- Mô hình này cho ra kết quả khá ổn.

2. Random Forest

- Để đánh giá tốt hơn mô hình, ta tiến hành phân tích thông qua classification report

	precision	recall	f1-score	support
1	0.95	0.92	0.93	193
2	0.89	0.94	0.92	1545
3	0.65	0.53	0.59	1151
4	0.66	0.68	0.67	1446
5	0.91	0.95	0.93	2428
accuracy			0.82	6763
macro avg	0.81	0.80	0.81	6763
weighted avg	0.81	0.82	0.81	6763

- Lớp 1,2,5 cho ra kết quả khá tốt.
- Lớp 3 và 4: các chỉ số ở lớp này không được cao. Điều này chứng tỏ mô hình khó phân lớp trên các lớp này.

2. Random Forest

- Để mô hình hoạt động hiệu quả hơn, ta sẽ sử dụng GridSearchCV để tìm ra các siêu tham số phù hợp.

```
# Init RandomForest model
RF_model = RandomForestClassifier(random_state=42)

# Define the grid of hyperparameters to search
param_grid = {
    'n_estimators': [20, 40, 50],
    'max_depth': [2,5,7,9],
    'min_samples_split': [2,5,7],
    'min_samples_leaf': [1, 2, 3, 4],
    'max_features': ['sqrt', 'log2'],
    'bootstrap': [True, False]
}

#  Perform grid search with cross-validation
gridcv_RF = GridSearchCV(RF_model, param_grid, cv=5, scoring='accuracy')
gridcv_RF.fit(X_train, y_train)

print("Best Parameters:", gridcv_RF.best_params_)
print("Best Accuracy:", gridcv_RF.best_score_)
```

✓ 23m 42.6s

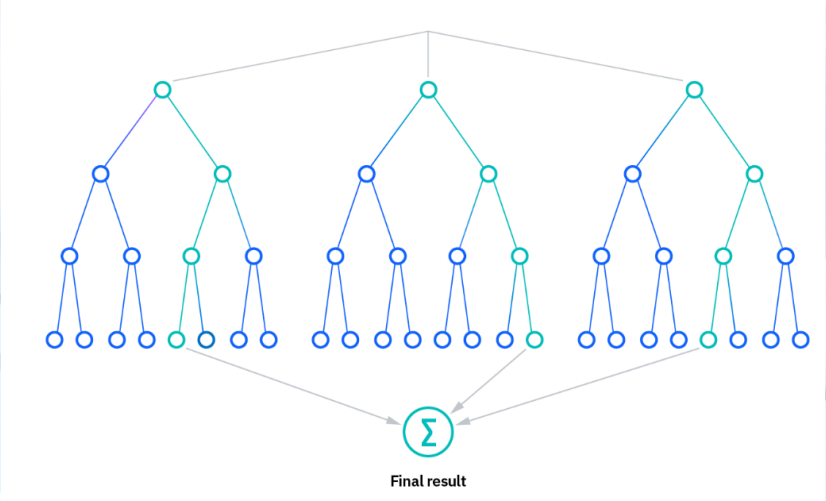
```
Best Parameters: {'bootstrap': False, 'max_depth': 9, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 40}
Best Accuracy: 0.8082809611829944
```

- Với các tham số, mô hình cho ra accuracy xấp xỉ 81%

3. Support Vector Machine (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:

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



3. Support Vector Machine (SVM)

- Khởi tạo một mô hình SVM với C = 5,

- Sử dụng Cross-Validation để chia dữ liệu thành năm phần, sử dụng 4 phần để huấn luyện và phần còn lại để đánh giá

```
#init SVM model
svm_model = SVC(kernel='linear', C=5, random_state=42)
svm_model.fit(X_train, y_train)
# cross validation on train set
cv_scores = cross_val_score(svm_model, X_train, y_train, cv=5)
print("Cross-Validation Scores:", cv_scores)

test_accuracy = svm_model.score(X_test, y_test)
print(f'Test Accuracy: {test_accuracy}')
```

✓ 40.0s

```
Cross-Validation Scores: [0.75656192 0.76303142 0.75951941 0.76118299 0.76469501]
Test Accuracy: 0.7530681650155256
```

3. Support Vector Machine (SVM)

- Lớp 1,2,5 cho ra kết quả rất tốt đặc biệt là lớp 2. Điều chứng tỏ mô hình chạy rất hiệu quả trên lớp 2.
- Mô hình chạy không cho ra hiệu suất cao trên lớp 3 và lớp 4.

```
classification_report:
              precision    recall  f1-score   support

           1       0.83      0.76      0.79       193
           2       0.84      0.89      0.86      1545
           3       0.49      0.48      0.49      1151
           4       0.57      0.56      0.56      1446
           5       0.92      0.91      0.91      2428

    accuracy                           0.75      6763
   macro avg       0.73      0.72      0.72      6763
weighted avg       0.75      0.75      0.75      6763
```

3. Support Vector Machine (SVM)

Mô hình có tốt hay không phụ thuộc rất nhiều vào tham số C:

- Giá trị C lớn thì đường biên chặt chẽ hơn nhưng có thể dẫn đến overfitting và thời gian chạy lâu.
- Giá trị C nhỏ thì mô hình có thể tổng quát hóa tốt hơn trên dữ liệu mới nhưng có nguy cơ trở nên underfitting.

```
classification_report:
              precision    recall  f1-score   support

           1       0.83      0.76      0.79       193
           2       0.84      0.89      0.86      1545
           3       0.49      0.48      0.49      1151
           4       0.57      0.56      0.56      1446
           5       0.92      0.91      0.91      2428

    accuracy                           0.75      6763
   macro avg       0.73      0.72      0.72      6763
weighted avg       0.75      0.75      0.75      6763
```

3. Support Vector Machine (SVM)

- Để mô hình hoạt động hiệu quả hơn, ta sẽ sử dụng GridSearchCV để tìm ra các siêu tham số phù hợp.

```
# Init SVM modelmodel
svm = SVC(kernel='linear', random_state=42)

# Define CC
param_grid = {'C': [0.001, 0.01, 0.1, 1, 10, 100]}

# GridSearchCV
svm_grid_search = GridSearchCV(svm, param_grid, cv=5, scoring='accuracy')

# Train GridSearchCV
svm_grid_search.fit(X_train, y_train)

print("Best C:", svm_grid_search.best_params_['C'])

# AccuracyAccuracy
print("Best Accuracy:", svm_grid_search.best_score_)
```

6] ✓ 4m 28.0s

```
Best C: 10
Best Accuracy: 0.7619223659889094
```

- Với các tham số mới, mô hình cho ra accuracy xấp xỉ 76%

TỔNG KẾT

- Tạo mẫu để phân loại chất lượng không khí
- Kết quả thu được là AQI = 2

```
Predicted AQI: [2]
```

```python
# create sample
sample = {
    'no': 0.09,
    'co': 300.76,
    'so2': 16,
    'no2': 14.43,
    'pm2_5': 22.5,
    'pm10': 35.2,
    'nh3': 12,
}
```

✓ 0.0s

TỔNG KẾT

- Mô hình cây quyết định:
  - Ưu điểm:
    + Dễ hiểu và diễn giải. Có thể trực quan hóa cây quyết định
    + Có khả năng xử lí cả dữ liệu số và số liệu phân loại.
    + Không cần nhiều tiền xử lí.
  - Nhược điểm:
    + Dễ bị overfitting nếu cây quá sâu.
    + Có thể không hiệu quả khi có quá nhiều biến và mối quan hệ phức tạp
- Mô hình Random Forest:
  - Ưu điểm:
    + Giảm thiểu tình trạng overfitting.
    + Khả năng thích ứng với các tập dữ liệu lớn.
  - Nhược điểm:
    + Tốn nhiều thời gian để huấn luyện hơn.
    + Tốn nhiều bộ nhớ.
- Mô hình SVM:
  - Ưu điểm:
    + Hiệu quả trong không gian chiều kích 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
  - Nhược điểm:
    + 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.