ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

MÔN: PHÂN TÍCH DỮ LIỆU

ĐỀ TÀI

XÂY DỰNG MÔ HÌNH DỰ ĐOÁN CHỈ SỐ CHẤT LƯỢNG KHÔNG KHÍ

Nhóm sinh viên thực hiện

3121410296 - Nguyễn Hoàng Long

3121410482 - Nguyễn Minh Thuận

GVHD: Phan Thành Huấn

TP. HỒ CHÍ MINH, THÁNG 12 NĂM 2024

MỤC LỤC

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến thầy Phan Thành Huấn về sự hướng dẫn và định hình trong quá trình hoàn thành báo cáo đồ án môn học "Phân tích dữ liệu".

Thầy đã dành thời gian và công sức để chia sẻ kiến thức sâu rộng của môn học và hỗ trợ chúng em vượt qua những thách thức trong quá trình nghiên cứu và phân tích. Bằng sự tận tâm và kiên nhẫn của mình, thầy đã giúp chúng em hiểu sâu hơn về vấn đề, từ đó xây dựng nên báo cáo một cách tổng thể và logic.

Chúng em rất trân trọng những lời khuyên, nhận xét và sự chỉ bảo tận tình từ thầy trong suốt quá trình làm việc. Điều này đã giúp chúng em không chỉ hoàn thành báo cáo một cách thành công mà còn nâng cao kỹ năng và kiến thức của mình trong lĩnh vực phân tích dữ liệu này.

Một lần nữa, chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy Phan Thành Huấn về sự đóng góp không ngừng nghỉ và sự hỗ trợ quý báu của thầy. Mong rằng chúng em sẽ tiếp tục nhận được sự hướng dẫn và động viên từ thầy trong những chặng đường tiếp theo của cuộc hành trình học tập và nghiên cứu của mình.

Trân trọng.

NHẬN XÉT CỦA GIẢNG VIÊN

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

TP. Hồ Chí Minh, ngày 10 tháng 12 năm 2024

Giảng viên hướng dẫn

Phan Thành Huấn

CHƯƠNG 1: TỔNG QUAN VỀ ĐỒ ÁN

1.1 Lý do chọn đề tài

Hiện nay, vấn đề ô nhiễm không khí đang trở thành một trong những thách thức nghiêm trọng nhất đối với sức khỏe con người và sự phát triển bền vững trên toàn thế giới. Theo các báo cáo của tổ chức môi trường quốc tế, chất lượng không khí tại nhiều khu vực đô thị đang có xu hướng suy giảm, gây ra nhiều hệ lụy như bệnh hô hấp, tim mạch và làm giảm chất lượng cuộc sống. Vì vậy, việc hiểu rõ và phân tích các thông số liên quan đến chất lượng không khí là bước quan trọng để đề xuất các biện pháp cải thiện hiệu quả.

Trong bối cảnh đó, dữ liệu về chất lượng không khí là một nguồn tài nguyên quý giá, giúp chúng ta hiểu rõ hơn về mức độ ô nhiễm, xu hướng biến đổi qua thời gian và sự khác biệt giữa các khu vực. Đề tài này tập trung vào việc sử dụng dữ liệu quan trắc môi trường từ các trạm đo để phân tích, đánh giá và trực quan hóa chất lượng không khí. Thông qua đó, sinh viên có thể:

- Áp dụng kiến thức đã học về phân tích dữ liệu vào một vấn đề thực tiễn và mang tính cấp thiết.
- Hiểu sâu hơn về các chỉ số quan trọng như PM2.5, PM10,..., AQI và các chất gây ô nhiễm khác.
- Đề xuất các giải pháp hoặc biện pháp cải thiện dựa trên kết quả phân tích.

Ngoài ra, với sự phát triển của công nghệ và dữ liệu lớn, các công cụ phân tích hiện đại giúp việc xử lý và trực quan hóa dữ liệu trở nên dễ dàng và hiệu quả hơn. Thông qua đề tài này, sinh viên không chỉ rèn luyện kỹ năng làm việc với dữ liệu thực tế mà còn xây dựng nền tảng quan trọng cho việc nghiên cứu chuyên sâu hoặc ứng dụng vào các lĩnh vực liên quan sau này.

Với những lý do trên, đề tài này không chỉ mang tính thực tiễn cao mà còn góp phần nâng cao nhận thức về vấn đề ô nhiễm không khí và giá trị của dữ liệu trong việc giải quyết các thách thức của xã hội.

1.2 Mục tiêu nghiên cứu

Mục tiêu của nghiên cứu này là xây dựng một hệ thống dự đoán AQI chính xác dựa trên các yếu tố ô nhiễm và đặc điểm mùa vụ. Nghiên cứu bao gồm các mục tiêu cụ thể sau:

- **Phân tích và lựa chọn các yếu tố quan trọng nhất ảnh hưởng đến AQI**:
  - Xác định các chỉ số ô nhiễm chính (như PM2.5, PM10, NOx, NO2, CO, v.v.) có mối quan hệ mạnh mẽ nhất với AQI thông qua các phương pháp thống kê và ma trận tương quan.
  - Đánh giá sự ảnh hưởng của đặc trưng mùa vụ (Season) đến AQI.
- **Xây dựng và so sánh hiệu suất của các mô hình dự đoán AQI**:

Sử dụng các thuật toán học máy hiện đại như:

  - **Multiple Linear Regression:** Để kiểm tra mối quan hệ tuyến tính giữa AQI và các yếu tố đầu vào.
  - **Random Forest:** Để mô hình hóa các mối quan hệ phi tuyến mạnh mẽ và giảm nhiễu.
  - **Gradient Boosting Regressor và XGBoost Regressor:** Để cải thiện khả năng dự đoán bằng cách tăng cường hiệu suất dần dần.
- **Đánh giá và cải thiện mô hình thông qua đặc trưng mùa vụ**:
  - Tích hợp thêm thông tin về mùa (Season) dựa trên dữ liệu "Tháng" (Month) để xem xét tác động của thời điểm trong năm đối với AQI.
- **Cung cấp cái nhìn trực quan hóa**:
  - Sử dụng các biểu đồ trực quan hóa (biểu đồ boxplot, heatmap, scatterplot) để minh họa mối quan hệ giữa các yếu tố đầu vào và AQI.

1.3 Tầm quan trọng của nghiên cứu

Nghiên cứu không chỉ cung cấp một cách tiếp cận hiệu quả để dự đoán AQI mà còn giúp nhận diện rõ hơn về các yếu tố gây ô nhiễm chính, từ đó hỗ trợ chính quyền và các nhà hoạch định chính sách trong việc xây dựng các biện pháp kiểm soát ô nhiễm không khí. Việc tích hợp đặc trưng mùa vụ cũng mang lại giá trị thực tiễn, giúp dự đoán tốt hơn trong các giai đoạn thời gian cụ thể, từ đó cảnh báo sớm cho người dân và cơ quan chức năng về các điều kiện không khí nguy hiểm.

1.4 Mô tả tập dữ liệu

Tập dữ liệu sử dụng trong đồ án được lấy từ nền tảng Kaggle, cụ thể từ bộ dữ liệu **"Station_day.csv"** do Dhanajani Jayarukshi công bố. Bộ dữ liệu này cung cấp thông tin về chất lượng không khí được ghi nhận từ các trạm quan trắc, bao gồm các chỉ số

quan trọng như PM2.5, PM10, NO2, CO, SO2, O3, và chỉ số chất lượng không khí tổng hợp AQI.

Link download: **https://www.kaggle.com/datasets/station_day.csv**

- **Quy mô tập dữ liệu**

Số dòng dữ liệu: 108,035 bản ghi.

Số cột: 16 cột, bao gồm các thông tin như mã trạm quan trắc, ngày đo, nồng độ các chất ô nhiễm và chỉ số chất lượng không khí (AQI).

- Station Id: Mã nhận diện của trạm quan trắc.
- Date: Ngày quan sát.
- PM2.5: Nồng độ bụi mịn PM2.5 (µg/m³)..
- PM10: Nồng độ bụi thô PM10 (µg/m³).
- NO : Nồng độ Nitric oxide (NO) (µg/m³)..
- NO2 : Nồng độ Nitrogen dioxide (NO_2) (µg/m³).
- NOx: Nồng độ các oxit Nitơ (µg/m³)..
- NH3: Nồng độ Ammonia (µg/m³).
- CO: Nồng độ Carbon monoxide (mg/m³).
- SO2: Nồng độ Sulfur dioxide (µg/m³).
- O3: Nồng độ Ozone (µg/m³).
- Benzene: Nồng độ Benzen (µg/m³).
- Toluene : Nồng độ Toluen (µg/m³).
- Xylene: Nồng độ Xylen (µg/m³).
- AQI: Chỉ số chất lượng không khí.
- AQI Bucket: Mức phân loại chất lượng không khí (Tốt, Trung bình, Kém, v.v.).

- **Bảng mô tả dữ liệu**

STT	Tên cột	Ý nghĩa	Kiểu dữ liệu	Ví dụ
1	StationId	Mã nhận diện của trạm quan trắc.	object	"AP001"
2	Date	Ngày quan trắc	object	"2017-11-24"
3	PM2.5	Nồng độ bụi mịn PM2.5 (μg/m³).	float64	71.36
4	PM10	Nồng độ bụi thô PM10 (μg/m³).	float64	115.75
5	NO	Nồng độ Nitric Oxide (NO) (μg/m³).	float64	1.75
6	NO2	Nồng độ Nitrogen Dioxide (NO_2) (μg/m³).	float64	20.65
7	NOx	Nồng độ các Oxit Nitơ (μg/m³).	float64	12.40
8	NH3	Nồng độ Ammonia (NH_3) (μg/m³).	float64	12.19
9	CO	Nồng độ Carbon Monoxide (CO) (mg/m³).	float64	0.10
10	SO2	Nồng độ Sulfur Dioxide (SO_2) (μg/m³).	float64	10.76

11	O3	Nồng độ Ozone (O_3) ($\mu g/m^3$)	float64	109.26
12	Benzene	Nồng độ Benzen ($\mu g/m^3$).	float64	0.17
13	Toluene	Nồng độ Toluen ($\mu g/m^3$).	float64	5.92
14	Xylene	Nồng độ Xylen ($\mu g/m^3$).	float64	0.10
15	AQI	Chỉ số chất lượng không khí tổng hợp (**Air Quality Index**).	float64	184.0
16	AQI_Bucket	Mức phân loại chất lượng không khí (Tốt, Trung bình, Kém, v.v.).	object	"Moderate"

- **Mục đích dữ liệu**

Dữ liệu này được thiết kế để phục vụ nghiên cứu và đánh giá chất lượng không khí tại nhiều khu vực. Một số ứng dụng cụ thể bao gồm:

- **Theo dõi ô nhiễm không khí:** Đánh giá mức độ ô nhiễm của các chất ô nhiễm chính (PM2.5, PM10, NO2, CO, SO2).
- **Dự đoán chỉ số AQI:** Sử dụng các mô hình học máy để dự đoán AQI dựa trên nồng độ các chất ô nhiễm.
- **Xác định nguồn gốc ô nhiễm:** Phân tích mối liên hệ giữa các yếu tố ô nhiễm để đề xuất biện pháp giảm thiểu.
- **Nghiên cứu xu hướng:** Phân tích sự thay đổi của AQI theo thời gian hoặc giữa các khu vực.

- **Tính đầy đủ và sạch sẽ của dữ liệu**

- **Dữ liệu thiếu:** Một số cột có dữ liệu bị thiếu (NaN), đặc biệt ở các cột như Xylene, NH3, PM10.
- **Độ sạch dữ liệu:** Trước khi phân tích, cần thực hiện xử lý dữ liệu bị thiếu, lọc các giá trị ngoại lệ

- **Các điểm mạnh của dữ liệu**

- **Quy mô lớn:** Với hơn 100.000 bản ghi, tập dữ liệu cung cấp thông tin đầy đủ để phân tích và dự đoán chất lượng không khí.
- **Đa dạng yếu tố:** Bao gồm nhiều chất ô nhiễm, giúp phân tích mối quan hệ đa chiều.
- **Khả năng mở rộng:** Có thể áp dụng để nghiên cứu các khu vực khác hoặc so sánh giữa các mùa trong năm.

- **Các điểm hạn chế**

- **Dữ liệu thiếu:** Một số cột như NH3 và Xylene có nhiều giá trị bị thiếu, cần xử lý trước khi sử dụng.
- **Không đầy đủ thông tin vị trí:** Mặc dù có mã trạm StationId, nhưng tập dữ liệu không cung cấp vị trí địa lý cụ thể của từng trạm.
- **Không rõ nguồn thời gian thực:** Dữ liệu được tổng hợp nhưng không có thông tin chi tiết về tần suất và cách thức đo lường.

CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU

Tiền xử lý dữ liệu là một bước quan trọng trong quy trình phân tích dữ liệu và xây dựng mô hình dự đoán. Nó đảm bảo rằng dữ liệu đầu vào được làm sạch, đầy đủ và phù hợp để đạt hiệu quả cao nhất khi áp dụng các phương pháp phân tích và thuật toán học máy. Trong chương này, chúng em trình bày chi tiết các bước tiền xử lý được thực hiện trên tập dữ liệu **Station_day.csv.**

2.1 Tìm hiểu và phân tích dữ liệu ban đầu

Mục tiêu: Xác định cấu trúc dữ liệu, các giá trị bị thiếu, các đặc điểm thống kê cơ bản, và phát hiện các bất thường trong dữ liệu.

- **Hiển thị thông tin dữ liệu**

```
print("Thông tin dữ liệu ban đầu:")
print(data.info())
```

Hình 2.1 Đoạn code hiển thị thông tin tập dữ liệu

```
Thông tin dữ liệu ban đầu:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 108035 entries, 0 to 108034
Data columns (total 16 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   StationId   108035 non-null  object
 1   Date        108035 non-null  object
 2   PM2.5       86410 non-null   float64
 3   PM10        65329 non-null   float64
 4   NO          90929 non-null   float64
 5   NO2         91488 non-null   float64
 6   NOx         92535 non-null   float64
 7   NH3         59930 non-null   float64
 8   CO          95037 non-null   float64
 9   SO2         82831 non-null   float64
 10  O3          82467 non-null   float64
 11  Benzene     76580 non-null   float64
 12  Toluene     69333 non-null   float64
 13  Xylene      22898 non-null   float64
 14  AQI         87025 non-null   float64
 15  AQI_Bucket  87025 non-null   object
dtypes: float64(13), object(3)
memory usage: 13.2+ MB
None
```

Hình 2.2 Thông tin tập dữ liệu ban đầu

- **Nhận xét:**
  - Bộ dữ liệu gồm 108,035 dòng và 16 cột, bao gồm các cột số liệu và cột phân loại.
  - Một số cột chứa nhiều giá trị bị thiếu, ví dụ: Xylene chỉ có 22,898 giá trị.
- **Kiểm tra số lượng giá trị bị thiếu của mỗi cột**

```
print("\nTổng số giá trị thiếu:")
print(data.isnull().sum())
```

Hình 2.3 Đoạn code kiểm tra số giá trị thiếu mỗi cột

```
Tổng số giá trị thiếu:
StationId        0
Date             0
PM2.5        21625
PM10         42706
NO           17106
NO2          16547
NOx          15500
NH3          48105
CO           12998
SO2          25204
O3           25568
Benzene      31455
Toluene      38702
Xylene       85137
AQI          21010
AQI_Bucket   21010
dtype: int64
```

Hình 2.4 Thông tin về dữ liệu thiếu mỗi cột

- **Nhận xét:**
  - Một số cột như PM10, NH3, và Xylene chứa rất nhiều giá trị bị thiếu.
- **Thống kê mô tả dữ liệu**

```
print("\nThống kê dữ liệu ban đầu:")
print(data.describe())
```

Hình 2.5 Thống kê dữ liệu ban đầu

Thống kê dữ liệu ban đầu:

	PM2.5	PM10	NO	...	Toluene	Xylene	AQI
count	86410.000000	65329.000000	90929.000000	...	69333.000000	22898.000000	87025.000000
mean	80.272571	157.968427	23.123424	...	15.345394	2.423446	179.749290
std	76.526403	123.418672	34.491019	...	29.348587	6.472409	131.324339
min	0.020000	0.010000	0.010000	...	0.000000	0.000000	8.000000
count	86410.000000	65329.000000	90929.000000	...	69333.000000	22898.000000	87025.000000
mean	80.272571	157.968427	23.123424	...	15.345394	2.423446	179.749290
std	76.526403	123.418672	34.491019	...	29.348587	6.472409	131.324339
min	0.020000	0.010000	0.010000	...	0.000000	0.000000	8.000000
mean	80.272571	157.968427	23.123424	...	15.345394	2.423446	179.749290
std	76.526403	123.418672	34.491019	...	29.348587	6.472409	131.324339
min	0.020000	0.010000	0.010000	...	0.000000	0.000000	8.000000
25%	31.880000	70.150000	4.840000	...	0.690000	0.000000	86.000000
50%	55.950000	122.090000	10.290000	...	4.330000	0.400000	132.000000
std	76.526403	123.418672	34.491019	...	29.348587	6.472409	131.324339
min	0.020000	0.010000	0.010000	...	0.000000	0.000000	8.000000
25%	31.880000	70.150000	4.840000	...	0.690000	0.000000	86.000000
50%	55.950000	122.090000	10.290000	...	4.330000	0.400000	132.000000
25%	31.880000	70.150000	4.840000	...	0.690000	0.000000	86.000000
50%	55.950000	122.090000	10.290000	...	4.330000	0.400000	132.000000
50%	55.950000	122.090000	10.290000	...	4.330000	0.400000	132.000000
75%	99.920000	208.670000	24.980000	...	17.510000	2.110000	254.000000
max	1000.000000	1000.000000	470.000000	...	454.850000	170.370000	2049.000000

Hình 2.6 Kết quả Thống kê dữ liệu ban đầu

- **Nhận xét:**
  - Một số cột có giá trị rất cao, ví dụ: PM2.5 và PM10 có giá trị lớn nhất lên tới 1000.
  - Chỉ số AQI có giá trị cao nhất là 2049, cho thấy có các ngoại lệ cần xử lý.

2.2 Làm sạch và tiền xử lý dữ liệu

Mục tiêu: Xử lý giá trị bị thiếu, loại bỏ cột không sử dụng, xử lý ngoại lệ và thêm các đặc trưng cần thiết để chuẩn bị dữ liệu cho phân tích.

2.2.1 Chuyển đổi định dạng ngày tháng

```
data['Date'] = pd.to_datetime(data['Date'], errors='coerce')
```

Hình 2.7 Đoạn code chuyển định dạng ngày tháng

- **Nhận xét:**
  - Chuyển cột Date sang định dạng datetime để sử dụng các tính năng liên quan đến ngày tháng, như trích xuất tháng.

2.2.2 Loại bỏ cột không cần thiết

```
# Loại bỏ cột không sử dụng
if 'AQI_Bucket' in data.columns:
    data = data.drop(columns=['AQI_Bucket'])
```

Hình 2.8 Đoạn code loại bỏ cột không cần thiết

- **Nhận xét:**

  Cột **AQI_Bucket** được loại bỏ vì không được sử dụng trong phân tích và xây dựng mô hình.

2.2.3 Loại bỏ các dòng chứa giá trị bị thiếu trong các cột chỉ số ô nhiễm

```
# Loại bỏ các dòng mà tất cả các chỉ số ô nhiễm đều trống
pollutant_columns = ['PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2', 'O3', 'Benzene', 'Toluene', 'Xylene', 'AQI']
data = data.dropna(subset=pollutant_columns, how='all')
```

Hình 2.9 Đoạn code xóa các dòng có toàn bộ chỉ số ô nhiễm trống

Kết quả sau xử lý:

```
Dữ liệu sau khi loại bỏ các dòng trống tất cả các chỉ số ô nhiễm:
<class 'pandas.core.frame.DataFrame'>
Index: 101148 entries, 0 to 108034
Data columns (total 15 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   StationId  101148 non-null  object
 1   Date       101148 non-null  datetime64[ns]
 2   PM2.5      86410 non-null   float64
 3   PM10       65329 non-null   float64
 4   NO         90929 non-null   float64
 5   NO2        91488 non-null   float64
 6   NOx        92535 non-null   float64
 7   NH3        59930 non-null   float64
 8   CO         95037 non-null   float64
 9   SO2        82831 non-null   float64
 10  O3         82467 non-null   float64
 11  Benzene    76580 non-null   float64
 12  Toluene    69333 non-null   float64
 13  Xylene     22898 non-null   float64
 14  AQI        87025 non-null   float64
```

- **Nhận xét:**
  - Các dòng mà tất cả các chỉ số ô nhiễm đều bị thiếu được loại bỏ để tập trung vào dữ liệu có giá trị.

2.2.4 Loại bỏ các cột có hơn 50% giá trị bị thiếu

```
missing_threshold = 0.5
col_threshold = missing_threshold * len(data)
data = data.dropna(thresh=col_threshold, axis=1)
```

Hình 2.9 Đoạn code loại bỏ cột có hơn 50% giá trị thiếu

Kết quả sau xử lý:

```
Dữ liệu sau khi loại bỏ các cột có hơn 50% giá trị bị thiếu:
<class 'pandas.core.frame.DataFrame'>
Index: 101148 entries, 0 to 108034
Data columns (total 14 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   StationId  101148 non-null  object
 1   Date       101148 non-null  datetime64[ns]
 2   PM2.5      86410 non-null   float64
 3   PM10       65329 non-null   float64
 4   NO         90929 non-null   float64
 5   NO2        91488 non-null   float64
 6   NOx        92535 non-null   float64
 7   NH3        59930 non-null   float64
 8   CO         95037 non-null   float64
 9   SO2        82831 non-null   float64
 10  O3         82467 non-null   float64
 11  Benzene    76580 non-null   float64
 12  Toluene    69333 non-null   float64
 13  AQI        87025 non-null   float64
dtypes: datetime64[ns](1), float64(12), object(1)
memory usage: 11.6+ MB
None
```

- **Nhận xét:**
    - Các cột có hơn 50% giá trị bị thiếu như Xylene được loại bỏ.

2.2.4 Điền giá trị bị thiếu bằng giá trị trung bình

```
num_cols = data.select_dtypes(include=["float64"]).columns
num_imputer = SimpleImputer(strategy="mean")
data[num_cols] = num_imputer.fit_transform(data[num_cols])
```

Hình 2.10 Đoạn code điền giá trị thiếu

Kết quả sau xử lý:

```
Dữ liệu sau khi điền giá trị thiếu (trung bình):
<class 'pandas.core.frame.DataFrame'>
Index: 101148 entries, 0 to 108034
Data columns (total 14 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   StationId   101148 non-null  object
 1   Date        101148 non-null  datetime64[ns]
 2   PM2.5       101148 non-null  float64
 3   PM10        101148 non-null  float64
 4   NO          101148 non-null  float64
 5   NO2         101148 non-null  float64
 6   NOx         101148 non-null  float64
 7   NH3         101148 non-null  float64
 8   CO          101148 non-null  float64
 9   SO2         101148 non-null  float64
 10  O3          101148 non-null  float64
 11  Benzene     101148 non-null  float64
 12  Toluene     101148 non-null  float64
 13  AQI         101148 non-null  float64
```

- **Nhận xét:**
  - Giá trị bị thiếu trong các cột số liệu được điền bằng trung bình của cột.

2.2.5 Xử lý ngoại lệ bằng clip giá trị

Trong bất kỳ bài toán học máy nào, việc xử lý ngoại lệ (outliers) là rất quan trọng vì chúng có thể làm sai lệch kết quả của mô hình. **Ngoại lệ** là những giá trị không điển hình, không đại diện cho phần lớn các quan sát trong tập dữ liệu. Những giá trị này có

thể xuất hiện do các lỗi đo đạc, sai sót trong quá trình thu thập dữ liệu hoặc các sự kiện bất thường.

Trong bài toán dự đoán **AQI** (Chỉ số chất lượng không khí), các ngoại lệ có thể xuất hiện do:

- **Lỗi ghi nhận**: Đo đạc sai hoặc trục trặc kỹ thuật của cảm biến.
- **Sự kiện bất thường**: Một sự kiện môi trường đột biến như đám cháy rừng hoặc ô nhiễm không khí nghiêm trọng.

Các ngoại lệ này có thể ảnh hưởng đến việc huấn luyện mô hình, khiến mô hình không học được những mẫu dữ liệu đúng, hoặc làm giảm độ chính xác của mô hình. Do đó, **loại bỏ ngoại lệ** giúp mô hình học tốt hơn từ những dữ liệu có tính đại diện cao hơn.

- **Quy trình xử lý ngoại lệ**

- Xác định ngưỡng ngoại lệ

  - Đối với mỗi cột số liệu trong các cột chỉ số ô nhiễm (pollutant_columns), tính bách phân vị thứ 1 (1st percentile) và bách phân vị thứ 99 (99th percentile).
  - Các giá trị nhỏ hơn bách phân vị thứ 1 và lớn hơn bách phân vị thứ 99 sẽ được coi là ngoại lệ.

- Cắt giá trị ngoài ngưỡng (Clipping)

  - Giá trị nhỏ hơn ngưỡng 1% được thay bằng giá trị ngưỡng 1%.
  - Giá trị lớn hơn ngưỡng 99% được thay bằng giá trị ngưỡng 99%.

- Code xử lý ngoại lệ

```
for col in pollutant_columns:
    upper_limit = data[col].quantile(0.99)  # Ngưỡng 99%
    lower_limit = data[col].quantile(0.01)  # Ngưỡng 1%
    data[col] = data[col].clip(lower=lower_limit, upper=upper_limit)
```

- Minh họa với cột PM2.5

Trước khi xử lý

```
print("Trước khi xử lý ngoại lệ:")
print(data['PM2.5'].describe())
```

```
count    86410.000000
mean        80.272571
std         76.526403
min          0.020000
25%         31.880000
50%         55.950000
75%         99.920000
max       1000.000000
```

- **Tính toán ngưỡng ngoại lệ**

```
upper_limit = data['PM2.5'].quantile(0.99)
lower_limit = data['PM2.5'].quantile(0.01)
print(f"Ngưỡng 1%: {lower_limit}, Ngưỡng 99%: {upper_limit}")
```

- **Kết quả**

```
Ngưỡng 1%: 7.45, Ngưỡng 99%: 359.36
```

- **Sau khi xử lý**

```
data['PM2.5'] = data['PM2.5'].clip(lower=lower_limit, upper=upper_limit)
print("Sau khi xử lý ngoại lệ:")
print(data['PM2.5'].describe())
```

```
count    86410.000000
mean        78.418634
std         65.785344
min          7.450000
25%         35.650000
50%         66.520000
75%         88.550000
max        359.362400
```

- **Nhận xét chung**

- Ưu điểm của phương pháp:

- Bảo toàn được nhiều dữ liệu quan trọng bằng cách không loại bỏ toàn bộ dòng có ngoại lệ.
- Dữ liệu được chuẩn hóa về mức độ hợp lý, giảm ảnh hưởng của các giá trị bất thường đến mô hình dự đoán.

Các giá trị cực đại và cực tiểu trong dữ liệu đều nằm trong khoảng 1% và 99%, phù hợp với phân phối thực tế của các chỉ số ô nhiễm.

2.2.6 Thêm đặc trưng "Mùa" (Season)

```
data['Month'] = data['Date'].dt.month
data['Season'] = data['Month'].apply(lambda x: 'Spring' if x in [3, 4, 5] else
                                     'Summer' if x in [6, 7, 8] else
                                     'Autumn' if x in [9, 10, 11] else
                                     'Winter')
data['Season'] = data['Season'].map({'Spring': 1, 'Summer': 2, 'Autumn': 3, 'Winter': 4})
```

- **Nhận xét:**

- Mùa (Season) được mã hóa thành dạng số để sử dụng trong mô hình dự đoán.

2.3 Kết luận về Tiền xử lý Dữ liệu

2.3.1 Phần tìm hiểu và phân tích dữ liệu ban đầu

Dữ liệu ban đầu có tổng cộng **108,035 mẫu** và **16 cột**, trong đó bao gồm các cột chứa các chỉ số ô nhiễm như PM2.5, PM10, NO, NO2, và AQI, cùng với các cột khác như Date, StationId, và AQI_Bucket. Một số kết quả cụ thể từ việc phân tích dữ liệu ban đầu:

- **Thông tin dữ liệu:**
  - Tổng cộng có 13 cột số liệu dạng float64, và 3 cột khác dạng object (bao gồm cột Date và StationId).
  - Cột AQI_Bucket chứa thông tin phân loại chất lượng không khí (categorical), nhưng không được sử dụng trong phân tích.
- **Tổng số giá trị bị thiếu:**
  - Một số cột có lượng giá trị bị thiếu lớn, ví dụ:
    - PM2.5: Thiếu 21,625 giá trị.
    - PM10: Thiếu 42,706 giá trị.
    - Xylene: Thiếu 85,137 giá trị (gần 80% tổng số dữ liệu).
  - Điều này dẫn đến việc xem xét loại bỏ các cột hoặc hàng có lượng giá trị bị thiếu lớn.
- **Thống kê cơ bản:**

- Chỉ số PM2.5 có giá trị trung bình là 80.27, độ lệch chuẩn là 76.53, và một số giá trị ngoại lệ rất cao (lên tới 1,000).
- Chỉ số AQI có giá trị trung bình là 179.75, với giá trị cao nhất lên đến 2,049.

2.3.2 Phần làm sạch và tiền xử lý dữ liệu

- **Chuyển đổi dữ liệu thời gian:**
  - Cột Date được chuyển đổi sang định dạng thời gian (datetime) để hỗ trợ các phân tích thời gian sau này.
- **Loại bỏ cột không cần thiết:**
  - Cột AQI_Bucket được loại bỏ do không sử dụng trong phân tích dự đoán.
- **Loại bỏ các dòng dữ liệu thiếu toàn bộ:**
  - Các dòng mà toàn bộ các chỉ số ô nhiễm (ví dụ: PM2.5, PM10,...) đều bị thiếu đã được loại bỏ. Kết quả:
  - Số lượng mẫu giảm từ 108,035 xuống còn 101,148.
- **Loại bỏ các cột có hơn 50% giá trị bị thiếu:**
  - Cột Xylene có hơn 80% giá trị bị thiếu đã được loại bỏ.
- **Điền giá trị thiếu:**
  - Sử dụng trung bình để điền các giá trị bị thiếu trong các cột số liệu (PM2.5, PM10, NO,...).
  - Kết quả: Không còn giá trị bị thiếu trong tập dữ liệu.
- **Loại bỏ ngoại lệ:**
  - Dựa trên phân vị 1% và 99%, các giá trị ngoại lệ của mỗi cột được loại bỏ bằng cách sử dụng phương pháp clip. Ví dụ:

    - PM2.5 được giới hạn từ 7.45 đến 359.36.

    - AQI được giới hạn từ 32 đến 571.

- **Thêm cột đặc trưng Season:**
  - Dựa trên cột Month, một cột mới Season được tạo ra để phân loại dữ liệu theo mùa (Xuân, Hè, Thu, Đông).
  - Cột này được mã hóa thành số nguyên: Xuân = 1, Hè = 2, Thu = 3, Đông = 4.

2.3.3 Kết quả đạt được sau tiền xử lý

- Số lượng mẫu: 101,148.
- Số lượng cột: 16, bao gồm các chỉ số ô nhiễm, cột thời gian, và cột Season.
- Không còn giá trị bị thiếu hoặc ngoại lệ trong dữ liệu.
- Tập dữ liệu đã sẵn sàng để tiến hành phân tích tương quan và xây dựng mô hình.

```
Thông tin dữ liệu cuối cùng sau tiền xử lý:
<class 'pandas.core.frame.DataFrame'>
Index: 101148 entries, 0 to 108034
Data columns (total 16 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   StationId  101148 non-null  object
 1   Date       101148 non-null  datetime64[ns]
 2   PM2.5      101148 non-null  float64
 3   PM10       101148 non-null  float64
 4   NO         101148 non-null  float64
 5   NO2        101148 non-null  float64
 6   NOx        101148 non-null  float64
 7   NH3        101148 non-null  float64
 8   CO         101148 non-null  float64
 9   SO2        101148 non-null  float64
 10  O3         101148 non-null  float64
 11  Benzene    101148 non-null  float64
 12  Toluene    101148 non-null  float64
 13  AQI        101148 non-null  float64
 14  Month      101148 non-null  int32
 15  Season     101148 non-null  int64
dtypes: datetime64[ns](1), float64(12), int32(1), int64(1), object(1)
```

CHƯƠNG 3: PHÂN TÍCH MỐI QUAN HỆ VÀ LỰA CHỌN ĐẶC TRƯNG

3.1 Ma trận tương quan

3.1.1 Khái niệm

Ma trận tương quan là một công cụ trong thống kê, dùng để đo lường mức độ liên hệ hoặc mối quan hệ giữa hai hoặc nhiều biến số. Cụ thể, tương quan thể hiện mức độ mà sự thay đổi của một biến có thể dự đoán được sự thay đổi của một biến khác. Nếu các biến có mối quan hệ mạnh mẽ, sự thay đổi của một biến có thể làm thay đổi biến kia.

- **Hệ số tương quan (Correlation Coefficient)** là một chỉ số thể hiện mối quan hệ giữa các biến. Nó có giá trị nằm trong khoảng từ -1 đến +1:
    - +1: Mối quan hệ hoàn toàn thuận (khi một biến tăng, biến kia cũng tăng theo tỉ lệ).
    - -1: Mối quan hệ hoàn toàn nghịch (khi một biến tăng, biến kia giảm).
    - 0: Không có mối quan hệ tuyến tính giữa các biến.

3.1.2 Mục đích sử dụng Ma Trận Tương Quan

- **Xác định mối quan hệ giữa các biến:** Phân tích xem các chỉ số ô nhiễm như PM2.5, PM10, NO2,... có mối quan hệ như thế nào với AQI. Điều này giúp hiểu rõ hơn về những yếu tố nào ảnh hưởng mạnh nhất đến chất lượng không khí.
- **Giảm thiểu đa cộng tuyến (Multicollinearity)**: Đảm bảo rằng không có các biến độc lập có mối quan hệ quá mạnh, điều này có thể ảnh hưởng đến độ chính xác của các mô hình dự đoán.
- **Lựa chọn biến cho mô hình học máy:** Từ ma trận tương quan, ta có thể lựa chọn các biến quan trọng nhất để đưa vào mô hình học máy, từ đó tối ưu hóa hiệu suất của mô hình.

3.1.3 Phương pháp tính toán ma trận tương quan

Ma trận tương quan có thể được tính bằng nhiều phương pháp khác nhau. Trong trường hợp dữ liệu có quan hệ tuyến tính, phương pháp phổ biến nhất là Hệ số tương quan Pearson.

- **Hệ số tương quan Pearson:** Đây là phương pháp phổ biến nhất để tính toán tương quan giữa các biến số. Nó đo lường mối quan hệ tuyến tính giữa hai biến và được tính theo công thức:

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

- r là hệ số tương quan Pearson.
- X_i và Y_i là giá trị của các biến X và Y tại điểm dữ liệu thứ i.
- $\bar{X}$ và $\bar{Y}$ là trung bình của các giá trị của biến X và Y.
    - |r| < 0.1: mối tương quan rất yếu
    - |r| < 0.3: mối tương quan yếu
    - |r| < 0.5: mối tương quan trung bình
    - |r| ≥ 0.5: mối tương quan mạnh

Hệ số tương quan này có thể được tính cho tất cả các cặp biến trong tập dữ liệu, từ đó xây dựng ma trận tương quan.

3.2 Áp dụng Ma Trận Tương Quan trong Dữ liệu Ô nhiễm Không khí

Việc lựa chọn đặc trưng quan trọng là một bước quan trọng trong tiền xử lý dữ liệu để giảm chiều dữ liệu, loại bỏ nhiễu, và tập trung vào các biến có tác động lớn nhất đến biến mục tiêu (trong trường hợp này là AQI). Chúng ta sử dụng ma trận tương quan để xác định các mối quan hệ chặt chẽ giữa các biến số.

- **Mục tiêu**
    - Tìm các cột chỉ số ô nhiễm có mối quan hệ chặt chẽ nhất với chỉ số AQI.
    - Loại bỏ các biến có tương quan thấp hoặc không liên quan, nhằm giảm tải tính toán và tăng hiệu quả của mô hình dự đoán.
- **Quy trình**

3.2.1 Tính toán ma trận tương quan:

Sử dụng phương pháp corr() của pandas để tính toán ma trận tương quan cho tất cả các cột số liệu.

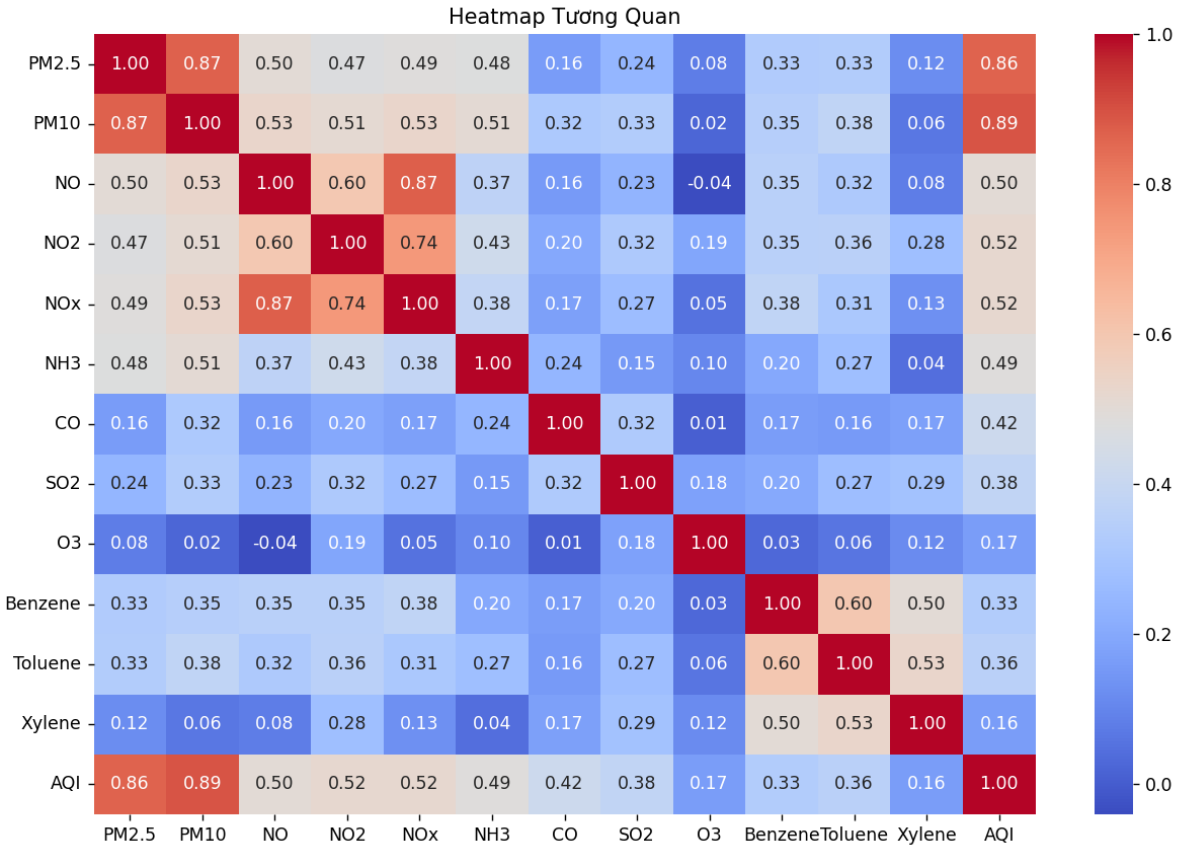
- Ma trận này cho biết mức độ tương quan (từ -1 đến 1) giữa các biến số:
  Giá trị gần 1: Quan hệ đồng biến mạnh.
  Giá trị gần -1: Quan hệ nghịch biến mạnh.

Giá trị gần 0: Ít hoặc không có quan hệ

Đoạn mã sử dụng:

```
plt.figure(figsize=(12, 8))
corr_matrix = data[pollutant_columns].corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Heatmap Tương Quan')
plt.show()
```

3.2.2 Hình ảnh ma trận tương quan



Dựa vào ma trận tương quan, chúng ta có thể thấy mối quan hệ giữa các biến với AQI (cột cuối cùng), từ đó xác định các biến có ảnh hưởng mạnh nhất đến AQI:

- **PM10 (Tương quan: 0.89):**
  - Đây là đặc trưng có mức độ tương quan cao nhất với AQI.
  - PM10 biểu thị nồng độ hạt bụi lớn hơn 2.5 micromet trong không khí. Các hạt này thường là sản phẩm của giao thông, công nghiệp, và đốt cháy nhiên liệu.
  - Giá trị tương quan 0.89 cho thấy PM10 là yếu tố quan trọng nhất ảnh hưởng đến AQI.
- **PM2.5 (Tương quan: 0.86):**

- Đây là đặc trưng quan trọng thứ hai. PM2.5 biểu thị các hạt bụi mịn (dưới 2.5 micromet), thường được tạo ra từ khí thải công nghiệp và giao thông.
- Với giá trị tương quan cao, PM2.5 đóng vai trò chính trong đánh giá mức độ ô nhiễm không khí.

- **NO2 (Tương quan: 0.52):**
  - Nitrogen dioxide (NO2) là một trong những khí gây ô nhiễm chính, thường được phát sinh từ đốt nhiên liệu hóa thạch, chẳng hạn như xe hơi và nhà máy công nghiệp.
  - Với tương quan 0.52, NO2 có mối quan hệ khá chặt chẽ với AQI, thể hiện vai trò quan trọng trong dự đoán.
- **NOx (Tương quan: 0.52):**
  - NOx là hỗn hợp của nitrogen monoxide (NO) và nitrogen dioxide (NO2). NOx là yếu tố quan trọng trong việc hình thành ozone tầng thấp và mưa axit.
  - Giá trị tương quan cao (0.52) chứng minh rằng NOx có ảnh hưởng đáng kể đến AQI.
- **CO (Tương quan: 0.42):**
  - Carbon monoxide (CO) là một loại khí không màu, không mùi, phát sinh từ đốt nhiên liệu hóa thạch. Tuy nhiên, mức tương quan 0.42 thấp hơn các biến khác, cho thấy CO ít ảnh hưởng đến AQI hơn so với PM và NOx.
- **Các biến có tương quan thấp:**
  - Ozone (O3): Tương quan rất thấp (0.17), cho thấy không có mối quan hệ mạnh giữa nồng độ ozone và AQI.
  - Benzene (0.33) và Toluene (0.36): Tương quan trung bình, không đóng vai trò lớn trong dự đoán AQI.

3.3 Lựa chọn các biến quan trọng có mối tương quan mạnh đến AQI

Dựa vào phân tích ở trên, các biến có tương quan mạnh nhất với AQI được lựa chọn bao gồm:

1. **PM10** (Tương quan: 0.89)
2. **PM2.5** (Tương quan: 0.86)
3. **NO2** (Tương quan: 0.52)
4. **NOx** (Tương quan: 0.52)
5. **NO** (Tương quan: 0.50)

Lý do lựa chọn:

- Các biến này có giá trị tương quan cao với AQI, cho thấy chúng là các yếu tố chính quyết định mức độ ô nhiễm không khí.
- Những đặc trưng này không chỉ có tương quan cao với AQI mà còn mang ý nghĩa thực tế trong việc đánh giá và quản lý chất lượng không khí.

3.4 Bổ sung biến "Season"

Ngoài 5 thuộc tính có tương quan mạnh nhất được lựa chọn từ ma trận tương quan, chúng ta bổ sung thêm biến Season (mùa) vào tập đặc trưng. Việc bổ sung biến này được thực hiện dựa trên các yếu tố khí hậu và điều kiện môi trường, vốn có tác động đáng kể đến chất lượng không khí (AQI).

3.4.1 Lý do bổ sung biến Season

- **Sự thay đổi theo mùa**

  Mùa đông:
  - Thường có AQI cao hơn do sự xuất hiện của hiện tượng nghịch nhiệt (temperature inversion). Hiện tượng này xảy ra khi không khí ở tầng thấp bị giữ lại bởi lớp không khí lạnh hơn ở phía trên, khiến cho các chất ô nhiễm không thể khuếch tán và bị giữ lại gần mặt đất.
  - Độ ẩm cao và sự hình thành sương mù cũng làm giảm khả năng khuếch tán các chất ô nhiễm**.**

  Mùa hè:
  - Trong mùa hè, tốc độ khuếch tán các chất ô nhiễm trong không khí cao hơn nhờ tốc độ gió lớn hơn và sự gia tăng của tầng đối lưu.
  - Cường độ ánh sáng mặt trời cũng ảnh hưởng đến sự hình thành ozone tầng mặt đất, có thể làm tăng hoặc giảm AQI tùy thuộc vào vị trí địa lý và nguồn phát thải.

  Mùa xuân và mùa thu:
  - Thường có sự chuyển đổi giữa các điều kiện thời tiết của mùa hè và mùa đông, với ảnh hưởng ít cực đoan hơn. Tuy nhiên, những ngày đầu xuân hoặc cuối thu có thể chịu ảnh hưởng của hiện tượng nghịch nhiệt, gây ra AQI cao hơn.

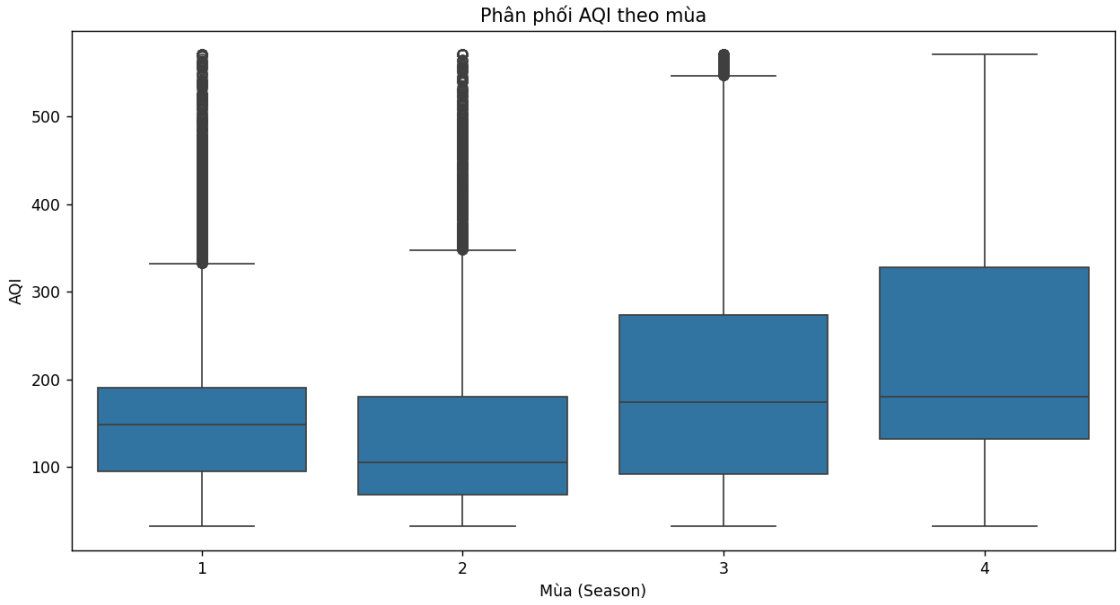
3.4.2 Phân tích phân phối AQI theo Season(mùa)

Dữ liệu cho thấy AQI thay đổi đáng kể giữa các mùa. Biểu đồ boxplot dưới đây minh họa rõ sự khác biệt trong phân phối AQI theo mùa

- **Trực quan hóa phân phối AQI theo mùa**

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='Season', y='AQI', data=data)
plt.title('Phân phối AQI theo mùa')
plt.xlabel('Mùa (Season)')
plt.ylabel('AQI')
plt.show()
```

- **Biểu đồ minh họa:**



Kết quả phân tích:

- Mùa đông có trung vị AQI cao nhất, với phạm vi biến động lớn, cho thấy mức độ ô nhiễm cao và không ổn định.
- Mùa hè có AQI thấp hơn và ổn định hơn, với ít điểm ngoại lệ hơn so với mùa đông.
- Mùa xuân và mùa thu có mức AQI nằm ở giữa, với phạm vi biến động trung bình.

3.4.3 Chi tiết về biến Season

Đoạn code thêm biến season

```
data['Month'] = data['Date'].dt.month
data['Season'] = data['Month'].apply(lambda x: 'Spring' if x in [3, 4, 5] else
                                     'Summer' if x in [6, 7, 8] else
                                     'Autumn' if x in [9, 10, 11] else
                                     'Winter')
data['Season'] = data['Season'].map({'Spring': 1, 'Summer': 2, 'Autumn': 3, 'Winter': 4})
```

- **Nhận xét:**

- Mùa (Season) được mã hóa thành dạng số để sử dụng trong mô hình dự đoán.

- **Tạo biến Season**: Dữ liệu ngày tháng từ cột **Date** được sử dụng để tạo thêm một cột mới mang tên **Season**. Cụ thể:
    - Tháng 3, 4, 5: **Mùa xuân (Spring)** - Mã hóa là **1**.
    - Tháng 6, 7, 8: **Mùa hè (Summer)** - Mã hóa là **2**.
    - Tháng 9, 10, 11: **Mùa thu (Autumn)** - Mã hóa là **3**.
    - Tháng 12, 1, 2: **Mùa đông (Winter)** - Mã hóa là **4**.
- **Lý do mã hóa số**: Mã hóa dạng số cho biến **Season** giúp sử dụng dễ dàng hơn trong các thuật toán mô hình hóa. Ngoài ra, biến này cũng giúp phản ánh sự thay đổi theo mùa khi kết hợp với các đặc trưng khác như PM2.5, PM10.

Kết luận về việc bổ sung Season

- Sự hiện diện của biến **Season** giúp mô hình dự đoán nắm bắt được ảnh hưởng của thời tiết và khí hậu theo thời gian.
- Giảm thiểu sai số của mô hình, đặc biệt trong các khoảng thời gian có mức ô nhiễm biến động lớn (như mùa đông).
- Tăng khả năng tổng quát hóa của mô hình, đặc biệt đối với các khu vực có đặc điểm khí hậu tương tự.

3.5 Kết luận của bước lựa chọn đặc trưng

Sau quá trình phân tích và lựa chọn, tập đặc trưng cuối cùng bao gồm 6 thuộc tính:

1. PM2.5
2. PM10
3. NO
4. NO2
5. NOx
6. Season

- **Cơ sở lựa chọn:**
    - **Dựa trên ma trận tương quan:**
        - PM2.5, PM10, NO, NO2, và NOx là 5 thuộc tính có hệ số tương quan cao nhất với chỉ số AQI. Những đặc trưng này thể hiện mối quan hệ mạnh mẽ và trực tiếp với mức độ ô nhiễm không khí.

- Các chất ô nhiễm như PM2.5 và PM10 thường là yếu tố chính trong việc tính toán AQI, do đó, mối liên hệ chặt chẽ giữa chúng và AQI là điều dễ hiểu.
- NO, NO2, và NOx là những chất khí độc hại từ các nguồn phát thải giao thông và công nghiệp, góp phần lớn vào sự suy giảm chất lượng không khí.

o **Bổ sung biến Season:**

- Biến Season được bổ sung để phản ánh ảnh hưởng của yếu tố thời gian và điều kiện khí hậu. Phân tích cho thấy AQI có sự thay đổi đáng kể giữa các mùa, do đó biến này giúp mô hình dự đoán nắm bắt được các yếu tố liên quan đến sự biến động khí hậu.

- **Đảm bảo tính toàn diện và đại diện:**
  - **Toàn diện**
    - Tập đặc trưng không chỉ tập trung vào các yếu tố có liên quan trực tiếp (các chất ô nhiễm như PM2.5, PM10) mà còn xem xét các yếu tố gián tiếp (Season) để bao quát các yếu tố có khả năng ảnh hưởng đến AQI.
  - **Đại diện:**
    - Các thuộc tính được lựa chọn từ các nguồn khác nhau như bụi mịn (PM2.5, PM10), khí độc (NO, NO2, NOx), và yếu tố thời gian (Season), đảm bảo mô hình có thể dự đoán chính xác trong nhiều bối cảnh khác nhau.
- **Kỳ vọng từ tập đặc trưng:**
  - Giúp các mô hình học máy nắm bắt được các yếu tố chính ảnh hưởng đến AQI, từ đó đưa ra dự đoán chính xác hơn.
  - Tăng cường khả năng giải thích của mô hình, khi các thuộc tính được chọn có mối liên hệ rõ ràng với AQI.
  - Giảm thiểu sự phức tạp không cần thiết, bằng cách loại bỏ các đặc trưng ít liên quan hoặc dư thừa.
- **Kết luận:**

Tập đặc trưng gồm 6 thuộc tính này là một lựa chọn hợp lý và cân đối. Nó đảm bảo tính đại diện của dữ liệu, giảm thiểu sự dư thừa và tập trung vào các yếu tố có ảnh hưởng lớn đến AQI. Với tập đặc trưng này, mô hình dự đoán sẽ đạt hiệu suất cao hơn và có khả năng giải thích tốt hơn trong việc dự đoán chất lượng không khí.

CHƯƠNG 4: XÂY DỰNG MÔ HÌNH DỰ ĐOÁN

4.1 Mục tiêu

4.1.1 Tổng quan

Bước này tập trung vào việc xây dựng các mô hình học máy để dự đoán chỉ số chất lượng không khí (AQI) dựa trên tập đặc trưng đã được lựa chọn từ bước trước. Mục tiêu không chỉ là tìm ra mô hình dự đoán hiệu quả nhất mà còn hiểu rõ cách các đặc trưng

ảnh hưởng đến kết quả dự đoán. Điều này hỗ trợ trong việc phát triển các chiến lược giảm thiểu ô nhiễm không khí.

4.1.2. Mục tiêu cụ thể

- **Xây dựng các mô hình học máy phổ biến**
  - Sử dụng 4 mô hình học máy: Multiple Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, và XGBoost Regressor.
  - Mỗi mô hình được cấu hình với các tham số mặc định và huấn luyện trên dữ liệu đã tiền xử lý.
- **Đánh giá hiệu suất mô hình**
  - **Đánh giá các mô hình dựa trên các chỉ số:**
    - **Mean Absolute Error (MAE):** Sai số tuyệt đối trung bình, biểu thị độ chênh lệch trung bình giữa giá trị dự đoán và giá trị thực.
    - **Root Mean Squared Error (RMSE):** Căn bậc hai của sai số bình phương trung bình, nhấn mạnh hơn vào các giá trị dự đoán sai lớn.
    - **R^2 (Hệ số xác định):** Đo lường mức độ mô hình giải thích được phương sai của dữ liệu.
  - Trực quan hóa hiệu suất qua biểu đồ "Actual vs Predicted AQI"
- **Lựa chọn mô hình tốt nhất:**
    - Chọn mô hình có hiệu suất cao nhất dựa trên chỉ số R^2, đồng thời xem xét MAE và RMSE.
    - Đảm bảo mô hình không chỉ chính xác mà còn phù hợp với các yếu tố đặc thù của dữ liệu AQI.

4.1.3 Lý do lựa chọn 4 mô hình

- **Multiple Linear Regression**:
  - Là mô hình cơ bản nhất, được sử dụng để kiểm tra xem mối quan hệ tuyến tính giữa đặc trưng và AQI có đủ mạnh để dự đoán không.
  - Làm cơ sở so sánh với các mô hình phi tuyến phức tạp hơn.
- **Random Forest Regressor**:
  - Là một mô hình phi tuyến dựa trên cây quyết định, hoạt động tốt với các đặc trưng phi tuyến và dữ liệu không đồng nhất.
  - Được biết đến với khả năng giảm thiểu overfitting nhờ kỹ thuật bagging.
- **Gradient Boosting Regressor**:
  - Là mô hình dựa trên thuật toán boosting, tối ưu hóa dần qua việc sửa lỗi của các cây trước đó.
  - Hiệu quả với dữ liệu phức tạp và khả năng giải thích các mối quan hệ phi tuyến giữa đặc trưng và đích.

- **XGBoost Regressor**:
  - Là phiên bản cải tiến của Gradient Boosting với tốc độ nhanh hơn và hiệu suất cao hơn nhờ tối ưu hóa các phép tính.
  - Được lựa chọn vì tính phổ biến và khả năng xử lý dữ liệu lớn.

4.2 Giới thiệu các mô hình sử dụng

4.2.1 Hồi quy tuyến tính bội (Multiple Linear Regression)

Hồi quy tuyến tính bội là một trong những phương pháp phân tích dữ liệu cơ bản nhưng hiệu quả, được sử dụng để mô hình hóa mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập. Đây là một phương pháp phổ biến trong lĩnh vực khoa học dữ liệu và học máy, bởi tính dễ hiểu, tính toán đơn giản, và khả năng giải thích trực quan.

- **Mục tiêu của hồi quy tuyến tính bội**

Mục tiêu chính của hồi quy tuyến tính bội là dự đoán giá trị của biến phụ thuộc (y) dựa trên các giá trị của các biến độc lập (x1,x2,...,xn) Điều này có ý nghĩa đặc biệt quan trọng trong các bài toán dự đoán như:

- Dự đoán giá nhà dựa trên diện tích, số phòng, và vị trí.
- Phân tích chỉ số chất lượng không khí (AQI) dựa trên các yếu tố gây ô nhiễm không khí.
- Ước lượng hiệu suất của một hệ thống dựa trên các thông số kỹ thuật.

Phương trình của hồi quy tuyến tính bội có dạng:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

Trong đó:

- **y:** Biến phụ thuộc (trong bài toán này là chỉ số AQI).
- **xi:** Các biến độc lập (các đặc trưng liên quan đến ô nhiễm không khí như PM2.5, PM10, NO, NO2, Nox,Season).
- **βi:** Hệ số hồi quy, đại diện cho mức độ ảnh hưởng của từng biến xi đến y. Các hệ số này được ước lượng từ dữ liệu.
- **B0:** Hằng số (intercept), đại diện cho giá trị dự đoán khi tất cả các xi=0
- **ϵ:** Sai số ngẫu nhiên, biểu thị các yếu tố không được mô hình hóa hoặc ảnh hưởng không đo lường được.

- **Giả định của hồi quy tuyến tính bội**

Để mô hình hoạt động hiệu quả, một số giả định cần được đáp ứng:

- **Tính tuyến tính:** Mối quan hệ giữa biến mục tiêu (y) và các biến độc lập (xi) là tuyến tính.
- **Phân phối chuẩn của sai số (ϵ):** Sai số giữa giá trị thực và giá trị dự đoán được giả định có phân phối chuẩn, với trung bình bằng 0.
- **Độc lập giữa các biến độc lập:** Các biến độc lập không nên có mối tương quan mạnh (tránh hiện tượng đa cộng tuyến).
- **Phương sai đồng nhất:** Phương sai của sai số không phụ thuộc vào giá trị của các biến độc lập.

- **Ứng dụng của hồi quy tuyến tính bội trong bài toán AQI**

  Trong bài toán dự đoán chỉ số chất lượng không khí (AQI), các biến độc lập như PM2.5, PM10, NO, NO2, Nox, Season đều có mối quan hệ trực tiếp hoặc gián tiếp với chỉ số AQI. Dựa vào phân tích ma trận tương quan, các biến này có mức độ tương quan cao với AQI, làm cho hồi quy tuyến tính bội trở thành một lựa chọn phù hợp để bắt đầu mô hình hóa.

  Ví dụ:

  - **PM2.5 và PM10:** Các hạt bụi mịn là nguyên nhân chính ảnh hưởng trực tiếp đến AQI.
  - **NOx, NO2:** Các khí thải từ phương tiện giao thông và công nghiệp thường đóng góp vào ô nhiễm không khí.
  - **Season (Mùa):** Yếu tố mùa cũng ảnh hưởng đến AQI do các điều kiện khí hậu như nhiệt độ, tốc độ gió và độ ẩm.

- **Ưu và nhược điểm của hồi quy tuyến tính bội**

  Ưu điểm:

  - **Dễ hiểu và giải thích:** Là mô hình trực quan nhất để phân tích tác động của từng biến độc lập đến biến phụ thuộc.
  - **Dễ triển khai:** Chỉ cần các biến đặc trưng được chuẩn hóa và không có hiện tượng đa cộng tuyến nghiêm trọng, mô hình dễ dàng được triển khai.
  - **Thích hợp cho dữ liệu tuyến tính:** Khi dữ liệu có mối quan hệ tuyến tính rõ ràng, hồi quy tuyến tính bội cho kết quả dự đoán chính xác.

- **Nhược điểm:**

- **Nhạy cảm với ngoại lệ:** Các giá trị ngoại lệ trong dữ liệu có thể làm sai lệch ước lượng của βi\beta_iβi.
- **Hạn chế khi dữ liệu phi tuyến:** Nếu mối quan hệ giữa y và xi không tuyến tính, mô hình có thể không hiệu quả.
- **Dễ bị ảnh hưởng bởi đa cộng tuyến:** Khi các biến độc lập có mối quan hệ mạnh, mô hình có thể bị mất ổn định.

- **Kỳ vọng từ hồi quy tuyến tính bội**

  Hồi quy tuyến tính bội cung cấp một nền tảng cơ bản để hiểu mối quan hệ giữa các yếu tố gây ô nhiễm không khí và chỉ số AQI. Dù không phải mô hình phức tạp nhất, nó cho phép chúng ta thiết lập một tiêu chuẩn để so sánh với các mô hình phi tuyến hoặc mạnh mẽ hơn như Random Forest, Gradient Boosting, hay XGBoost.

4.2.2 Random Forest Regressor

- **Giới thiệu Random Forest Regressor**
  - Random Forest là một thuật toán học máy thuộc nhóm học máy tổ hợp (ensemble learning), được phát triển bởi Leo Breiman. Thuật toán này hoạt động dựa trên việc kết hợp nhiều mô hình yếu (weak learners), cụ thể là các cây quyết định (Decision Trees), để tạo ra một mô hình mạnh hơn và đáng tin cậy hơn. Random Forest không chỉ được sử dụng cho bài toán hồi quy mà còn rất phổ biến trong các bài toán phân loại. Trong bài toán hồi quy, Random Forest Regressor có khả năng dự đoán giá trị liên tục (như AQI) bằng cách trung bình hóa kết quả của nhiều cây quyết định.
  - Mô hình Random Forest khắc phục được nhiều nhược điểm của cây quyết định đơn lẻ, chẳng hạn như dễ bị overfitting và không ổn định khi dữ liệu thay đổi nhỏ. Nó đạt được điều này thông qua việc xây dựng các cây quyết định độc lập trên các tập dữ liệu ngẫu nhiên và thực hiện dự đoán bằng cách tổng hợp kết quả từ các cây.
- **Cơ chế hoạt động của Random Forest Regressor**

  Quy trình hoạt động của Random Forest Regressor bao gồm các bước sau:
  - **Kỹ thuật Bagging (Bootstrap Aggregating):**
    - Một tập con của dữ liệu gốc được tạo ra bằng cách chọn mẫu ngẫu nhiên có hoàn lại (sampling with replacement). Điều này có nghĩa là một số mẫu dữ liệu có thể xuất hiện nhiều lần trong tập con, trong khi một số mẫu khác có thể bị bỏ qua.
    - Mỗi tập con này được sử dụng để huấn luyện một cây quyết định riêng biệt.

- **Xây dựng cây quyết định:**
  - Ở mỗi nút của cây, chỉ một tập con ngẫu nhiên của các đặc trưng được xem xét để tách nhánh (split). Điều này giúp giảm sự tương quan giữa các cây, tăng tính đa dạng và hiệu quả của mô hình tổ hợp.
  - Cây được xây dựng đến độ sâu tối đa hoặc đến khi đạt điều kiện dừng (như số lượng mẫu trong nút nhỏ hơn một ngưỡng nhất định).
- **Dự đoán:**
  - Mỗi cây trong rừng sẽ thực hiện dự đoán độc lập.
  - Đối với bài toán hồi quy, kết quả cuối cùng được tính bằng cách lấy trung bình dự đoán của tất cả các cây.

- **Ưu điểm của Random Forest**
  - **Khả năng giảm thiểu overfitting**: Random Forest giảm thiểu overfitting nhờ vào việc tổng hợp kết quả từ nhiều cây. Một cây quyết định đơn lẻ thường có xu hướng overfit dữ liệu huấn luyện, nhưng khi kết hợp nhiều cây với kết quả trung bình, Random Forest trở nên ổn định hơn và ít bị ảnh hưởng bởi dữ liệu nhiễu.
  - **Hiệu quả với dữ liệu phi tuyến**: Random Forest có khả năng mô hình hóa các mối quan hệ phi tuyến giữa các đặc trưng và biến mục tiêu. Điều này làm cho nó vượt trội hơn các mô hình tuyến tính như Linear Regression trong nhiều trường hợp.
  - **Khả năng đánh giá tầm quan trọng của đặc trưng:** Một trong những tính năng quan trọng của Random Forest là khả năng cung cấp thông tin về mức độ quan trọng của các đặc trưng. Điều này được thực hiện bằng cách đo lường mức độ giảm phương sai (variance reduction) của mỗi đặc trưng trên tất cả các cây.
  - **Khả năng xử lý dữ liệu không đồng nhất**: Random Forest hoạt động tốt trên dữ liệu hỗn hợp (cả dữ liệu số và dữ liệu phân loại) và không yêu cầu các đặc trưng phải chuẩn hóa hoặc chuẩn hóa trước khi huấn luyện.
- **Nhược điểm của Random Forest**
  - **Tốn tài nguyên tính toán:** Việc xây dựng nhiều cây quyết định và thực hiện dự đoán từ tất cả các cây làm cho Random Forest trở nên chậm hơn, đặc biệt khi số lượng cây hoặc kích thước dữ liệu lớn.
  - **Khó giải thích:** Mặc dù Random Forest cung cấp kết quả chính xác hơn so với một cây quyết định đơn lẻ, nhưng mô hình này được xem là một "hộp đen" (black box) vì khó giải thích cơ chế hoạt động chi tiết.
- **Kỳ vọng từ Random Forest Regressor**
  - Với dữ liệu AQI, nơi có thể tồn tại các mối quan hệ phi tuyến hoặc phức tạp giữa các đặc trưng (như PM2.5, NO2) và biến mục tiêu, Random Forest là một lựa chọn lý tưởng. Chúng ta kỳ vọng Random Forest sẽ vượt trội hơn so với Linear

Regression trong việc nắm bắt các mối quan hệ phức tạp, đồng thời duy trì tính ổn định và độ chính xác cao.

4.2.3. Gradient Boosting Regressor

- **Giới thiệu Gradient Boosting Regressor**

Gradient Boosting Regressor (GBR) là một thuật toán học máy mạnh mẽ và linh hoạt, thuộc nhóm boosting trong học máy tổ hợp (ensemble learning). Thuật toán này được thiết kế để giải quyết các bài toán hồi quy và phân loại bằng cách xây dựng tuần tự nhiều mô hình yếu (weak learners), thường là các cây quyết định nông.

Điểm khác biệt chính giữa Gradient Boosting và các phương pháp ensemble khác (như Random Forest) nằm ở cơ chế xây dựng các cây. Thay vì xây dựng các cây quyết định độc lập và tổng hợp kết quả, Gradient Boosting xây dựng các cây tuần tự, mỗi cây mới được tối ưu hóa để sửa lỗi của cây trước đó. Phương pháp này giúp Gradient Boosting trở thành một công cụ mạnh mẽ trong các bài toán phức tạp và phi tuyến.

- **Cơ chế hoạt động của Gradient Boosting Regressor**
  Gradient Boosting hoạt động thông qua ba bước chính:
  - **Khởi tạo mô hình ban đầu**
    - Mô hình đầu tiên chỉ đơn giản là dự đoán giá trị trung bình của biến mục tiêu (ví dụ: giá trị trung bình của AQI trong tập huấn luyện).
    - Giá trị trung bình này được sử dụng làm dự đoán ban đầu cho tất cả các mẫu.
  - **Xây dựng các cây tuần tự**
    - Sai số (residuals) giữa giá trị thực tế và giá trị dự đoán hiện tại được tính toán.
    - Một cây quyết định mới được huấn luyện để dự đoán sai số này. Mục tiêu của cây mới là giảm thiểu phần sai số còn lại từ các bước trước.
  - **Cập nhật dự đoán**
    - Dự đoán tổng thể được cập nhật bằng cách cộng thêm dự đoán từ cây mới với một trọng số được điều chỉnh bởi tham số học suất (learning rate).
    - Quá trình này được lặp lại nhiều lần, mỗi lần thêm một cây mới vào mô hình.
- **Tối ưu hóa bằng Gradient Descent**:
  - Gradient Boosting tối ưu hóa một hàm tổn thất (loss function), chẳng hạn như lỗi bình phương (MSE). Thuật toán sử dụng Gradient Descent để điều chỉnh mô hình, giúp tối ưu hóa dự đoán.

- **Ưu điểm của Gradient Boosting Regressor**
    - **Hiệu suất cao**: Gradient Boosting nổi bật với khả năng xử lý các bài toán phi tuyến và phức tạp. Nhờ cơ chế tối ưu hóa tuần tự, nó có thể nắm bắt các mối quan hệ tinh tế giữa các biến đặc trưng và biến mục tiêu.
    - **Tùy chỉnh linh hoạt**: Thuật toán cung cấp nhiều tham số có thể điều chỉnh, như:
        - **Learning rate (tốc độ học)**: Quyết định mức độ điều chỉnh mô hình trong mỗi bước.
        - **Số lượng cây**: Kiểm soát độ phức tạp và khả năng khái quát hóa của mô hình.
        - **Độ sâu cây**: Giới hạn số cấp của cây quyết định, giúp kiểm soát overfitting.
    - **Xử lý tốt dữ liệu mất cân bằng**: Nhờ khả năng tập trung vào các mẫu khó dự đoán, Gradient Boosting thường hoạt động tốt trên các tập dữ liệu có sự mất cân bằng.
- **Nhược điểm của Gradient Boosting Regressor**
    - **Thời gian huấn luyện dài**: Gradient Boosting đòi hỏi thời gian huấn luyện lâu hơn so với các thuật toán ensemble khác, như Random Forest, do các cây được xây dựng tuần tự.
    - **Dễ bị overfitting**: Gradient Boosting có thể overfit dữ liệu huấn luyện nếu không được điều chỉnh tham số cẩn thận (đặc biệt là số lượng cây và tốc độ học).
    - **Cần điều chỉnh tham số cẩn thận**: Hiệu suất của Gradient Boosting phụ thuộc nhiều vào việc lựa chọn các tham số phù hợp, điều này có thể đòi hỏi nhiều thời gian và công sức để thử nghiệm.
- **Kỳ vọng từ Gradient Boosting Regressor**
    - Gradient Boosting được kỳ vọng mang lại hiệu suất cao trong việc dự đoán AQI nhờ khả năng tối ưu hóa dự đoán từng bước và tập trung vào các mẫu khó dự đoán. So với các mô hình khác như Linear Regression và Random Forest, Gradient Boosting có tiềm năng vượt trội hơn trong việc xử lý các mối quan hệ phi tuyến và phức tạp giữa các đặc trưng.

4.2.4. XGBoost Regressor

- **Giới thiệu XGBoost Regressor**
    - **XGBoost (Extreme Gradient Boosting)** là một cải tiến nổi bật của Gradient Boosting, được thiết kế để tối ưu hóa tốc độ và hiệu suất trong các bài toán hồi

quy và phân loại. Thuật toán này ra đời nhằm khắc phục những hạn chế của Gradient Boosting truyền thống, như thời gian huấn luyện dài và nguy cơ overfitting.

- **XGBoost** được biết đến như một trong những mô hình mạnh mẽ nhất, thường được các nhà khoa học dữ liệu sử dụng trong các cuộc thi trên nền tảng Kaggle và các bài toán thực tế. Điểm mạnh của XGBoost nằm ở khả năng tích hợp các công nghệ tối ưu hóa, từ xử lý song song đến chuẩn hóa (regularization).

- **Cơ chế hoạt động của XGBoost Regressor**

  XGBoost Regressor hoạt động tương tự Gradient Boosting, nhưng với những cải tiến sau:

  - **Tối ưu hóa hàm tổn thất (Loss Function):**
    - Sử dụng thuật toán Gradient Descent để giảm thiểu sai số giữa giá trị thực tế và giá trị dự đoán.
    - Hỗ trợ nhiều hàm tổn thất khác nhau (như MSE, MAE), giúp mô hình phù hợp với nhiều loại bài toán.
  - **Regularization (chuẩn hóa):**
    - XGBoost bổ sung hai hình thức regularization: L1 (Lasso) và L2 (Ridge).
    - Regularization giúp giảm thiểu overfitting, làm cho mô hình hoạt động tốt hơn trên dữ liệu kiểm tra.
  - **Xử lý giá trị bị thiếu tự động**:
    - XGBoost tự động phát hiện và xử lý các giá trị bị thiếu trong tập dữ liệu, tránh việc phải xử lý trước đó.
  - **Tính toán song song**:
    - XGBoost tận dụng tính toán song song để tăng tốc độ huấn luyện, đặc biệt trên các tập dữ liệu lớn.
  - **Tối ưu hóa cây quyết định**:
    - Các cây được xây dựng dựa trên thuật toán tối ưu hóa "approximation algorithms", giúp cải thiện hiệu suất mà không làm mất đi tính chính xác.
  - **Shrinking (Thu nhỏ cây)**:
    - Sau mỗi vòng lặp, các trọng số của cây trước đó được giảm dần theo một hệ số học suất (learning rate) để tập trung vào các mẫu khó.

- **Ưu điểm của XGBoost Regressor**
  - **Tốc độ nhanh:**
    - XGBoost vượt trội về tốc độ huấn luyện nhờ tính toán song song và tối ưu hóa thuật toán.
  - **Hiệu quả cao trên dữ liệu lớn:**

- Mô hình hoạt động tốt với các tập dữ liệu lớn và phức tạp, nhờ vào sự kết hợp của regularization và các thuật toán tối ưu hóa.

  - **Tích hợp xử lý giá trị thiếu:**
    - Không cần xử lý thủ công các giá trị bị thiếu, XGBoost tự động xác định và xử lý chúng trong quá trình xây dựng cây.
  - **Khả năng khái quát hóa tốt:**
    - Nhờ regularization, XGBoost có thể khái quát hóa tốt, giảm nguy cơ overfitting ngay cả khi số lượng cây lớn.

- **Nhược điểm của XGBoost Regressor**
  - **Tốn tài nguyên tính toán**:
    - XGBoost yêu cầu nhiều tài nguyên tính toán, đặc biệt khi huấn luyện trên các tập dữ liệu lớn.
  - **Khó tinh chỉnh tham số**:
    - Mặc dù cung cấp nhiều tham số linh hoạt, việc điều chỉnh các tham số như learning rate, max depth, và số lượng cây đòi hỏi nhiều kinh nghiệm và thử nghiệm.
  - **Độ phức tạp thuật toán cao**:
    - So với Gradient Boosting thông thường, XGBoost phức tạp hơn về mặt lý thuyết và triển khai.

- **Kỳ vọng từ XGBoost Regressor**

XGBoost được kỳ vọng là một trong những mô hình hiệu quả nhất trong bài toán dự đoán AQI, nhờ vào khả năng tối ưu hóa tốc độ, hiệu suất và tính khái quát. Với sự hỗ trợ của các công nghệ tiên tiến như regularization và tính toán song song, XGBoost hứa hẹn mang lại dự đoán chính xác và hiệu quả ngay cả với các mối quan hệ phi tuyến phức tạp giữa các đặc trưng.

4.2.5 So sánh giữa các mô hình

Mô hình	Loại mô hình	Phức tạp	Khả năng xử lý dữ liệu phi tuyến	Dễ giải thích
Linear Regression	Hồi quy tuyến tính	Thấp	Không	Có
Random Forest Regressor	Ensemble	Trung bình	Có	Trung bình
Gradient Boosting Regressor	Boosting	Cao	Có	Khó
XGBoost Regressor	Boosting nâng cao	Cao	Có	Khó

4.3. Quy trình huấn luyện và đánh giá

4.3.1. Chuẩn bị dữ liệu cho quá trình huấn luyện

```
features = data[["PM2.5", "PM10", "NO", "NO2", "NOx", "Season"]]
target = data['AQI']
```

- **features**: Tập đặc trưng, bao gồm 6 thuộc tính đã chọn ở **chương 3**.
- **target**: Biến mục tiêu, chính là chỉ số AQI.

Sau khi lựa chọn được tập đặc trưng quan trọng gồm 6 thuộc tính: **PM2.5**, **PM10**, **NO**, **NO2**, **NOx**, và **Season**, chúng ta tách dữ liệu thành hai phần:

- **Tập huấn luyện (Training set)**: Sử dụng để huấn luyện mô hình, chiếm 80% dữ liệu.
- **Tập kiểm tra (Test set)**: Sử dụng để đánh giá hiệu quả của mô hình trên dữ liệu chưa từng được nhìn thấy, chiếm 20% dữ liệu.

```
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)
```

Quá trình tách dữ liệu được thực hiện bằng hàm train_test_split từ thư viện **scikit-learn**, với tham số random_state=42 để đảm bảo kết quả có thể tái lập

4.3.2. Mô hình hóa và huấn luyện

Các mô hình được huấn luyện trên tập huấn luyện và sau đó được đánh giá trên tập kiểm tra. Quy trình huấn luyện được thực hiện như sau:

- **Khởi tạo mô hình**: Các mô hình được định nghĩa bao gồm
  - Linear Regression
  - Random Forest Regressor
  - Gradient Boosting Regressor

- XGBoost Regressor

```
models = {
    "Linear Regression": LinearRegression(),
    "Random Forest": RandomForestRegressor(random_state=42, n_estimators=100),
    "Gradient Boosting Regressor": GradientBoostingRegressor(random_state=42, n_estimators=100),
    "XGBoost Regressor": XGBRegressor(random_state=42, n_estimators=100)
}
```

- **Huấn luyện mô hình trên tập huấn luyện**

```
results = {}
for name, model in models.items():
    print(f"\nĐang huấn luyện mô hình: {name}...")
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    rmse = mse ** 0.5
    r2 = r2_score(y_test, y_pred)
    results[name] = {
        "MAE": mae,
        "RMSE": rmse,
        "R2": r2
    }
    print(f"Kết quả mô hình {name}:")
    print(f"MAE: {mae}, RMSE: {rmse}, R2 Score: {r2}")
```

- **results:** Lưu kết quả của mỗi mô hình.
- **for name, model in models.items():** Lặp qua các mô hình trong từ điển models
- **Sử dụng tập huấn luyện qua phương thức fit**

```
model.fit(X_train, y_train)
```

Huấn luyện mô hình với dữ liệu huấn luyện (X_train là đặc trưng và y_train là giá trị thực tế).

- **Dự đoán trên tập kiểm tra:** Dự đoán AQI trên tập kiểm tra với phương thức predict

```
y_pred = model.predict(X_test)
```

Dự đoán giá trị AQI cho tập kiểm tra (X_test).

4.3.3. Đánh giá hiệu suất mô hình

Hiệu suất của từng mô hình được đánh giá thông qua các chỉ số sau:

- **Mean Absolute Error (MAE):** Trung bình giá trị tuyệt đối của sai số dự đoán

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- **Mean Squared Error (MSE):** Trung bình bình phương của sai số dự đoán

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **Root Mean Squared Error (RMSE):** Căn bậc hai của MSE, biểu thị sai số dự đoán trong cùng đơn vị với biến mục tiêu.

$$RMSE = \sqrt{MSE}$$

- **R^2 Score (Coefficient of Determination)**: Đo lường mức độ phù hợp của mô hình với dữ liệu. Giá trị R2R^2R2 càng gần 1 cho thấy mô hình càng tốt.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Cách tính:

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5
r2 = r2_score(y_test, y_pred)
```

- **Kết quả của các mô hình được lưu trữ và so sánh:**

```
results[name] = {
    "MAE": mae,
    "RMSE": rmse,
    "R2": r2
}
print(f"Kết quả mô hình {name}:")
print(f"MAE: {mae}, RMSE: {rmse}, R2 Score: {r2}")
```

Kết quả của các mô hình :

- **Linear Regression:**
  - MAE: 36.90
  - RMSE: 56.43
  - R^2: 0.75
- **Random Forest Regressor:**
  - MAE: 27.18
  - RMSE: 47.04
  - R^2: 0.83
- **Gradient Boosting Regressor:**
  - MAE: 29.53
  - RMSE: 48.95
  - R^2: 0.81
- **XGBoost Regressor**:
  - MAE: 27.24
  - RMSE: 46.83
  - R^2: 0.83

4.3.4. Trực quan hóa kết quả dự đoán

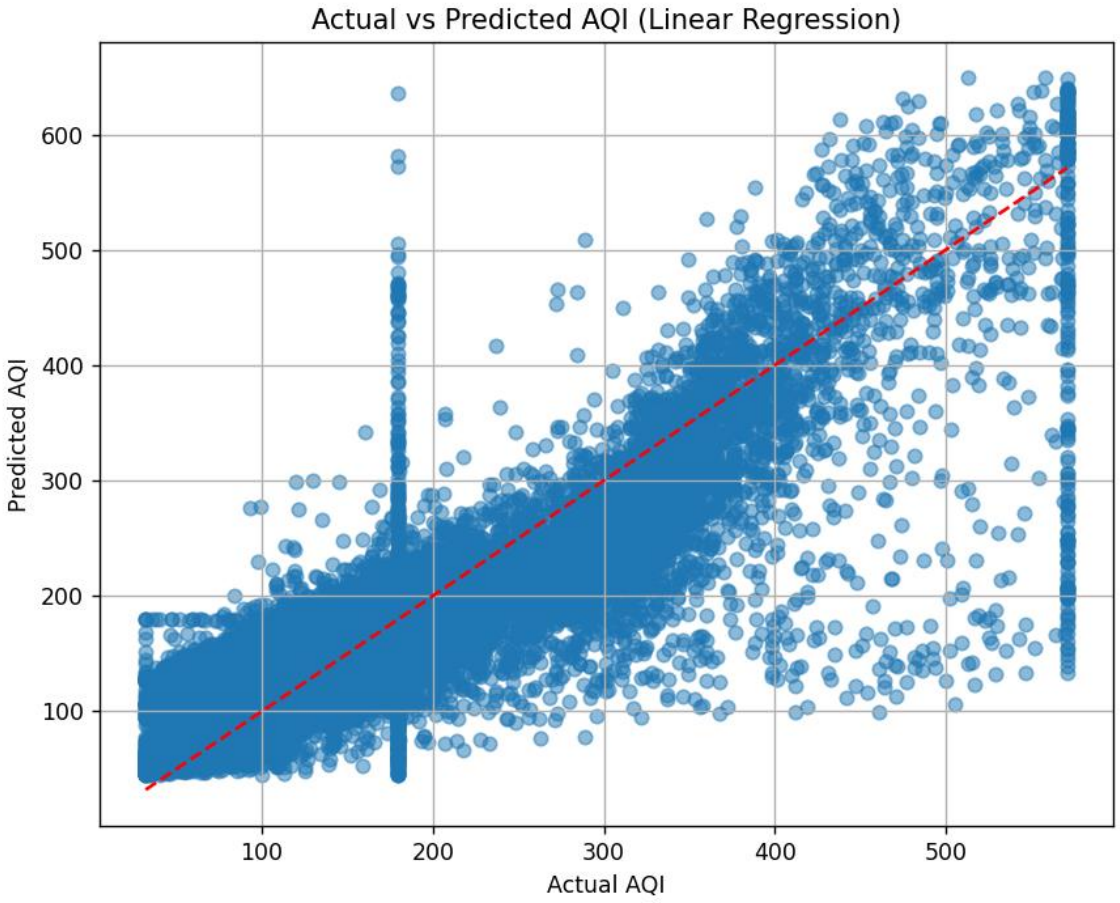
Để đánh giá trực quan hiệu suất mô hình, biểu đồ Actual vs Predicted (Thực tế so với Dự đoán) được vẽ cho từng mô hình. Biểu đồ này cho thấy mối tương quan giữa giá trị AQI thực tế và giá trị AQI dự đoán:

- Đường màu đỏ y=x biểu diễn mối quan hệ hoàn hảo (dự đoán chính xác hoàn toàn).
- Điểm dữ liệu càng gần đường đỏ, mô hình càng chính xác.

```
# Trực quan hóa "Actual vs Predicted"
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.title(f'Actual vs Predicted AQI ({name})')
plt.xlabel('Actual AQI')
plt.ylabel('Predicted AQI')
plt.grid(True)
plt.show()
```

- Dự đoán của mô hình được so sánh với giá trị thực tế (y_test vs y_pred) qua biểu đồ phân tán (scatter plot).
- Dòng thẳng đỏ r-- là đường chéo, đại diện cho việc dự đoán hoàn toàn chính xác (nếu các điểm nằm trên đường này thì dự đoán hoàn toàn chính xác).
- Mỗi mô hình sẽ có biểu đồ của riêng mình để giúp hình dung độ chính xác của dự đoán.

- **Linear Regression**



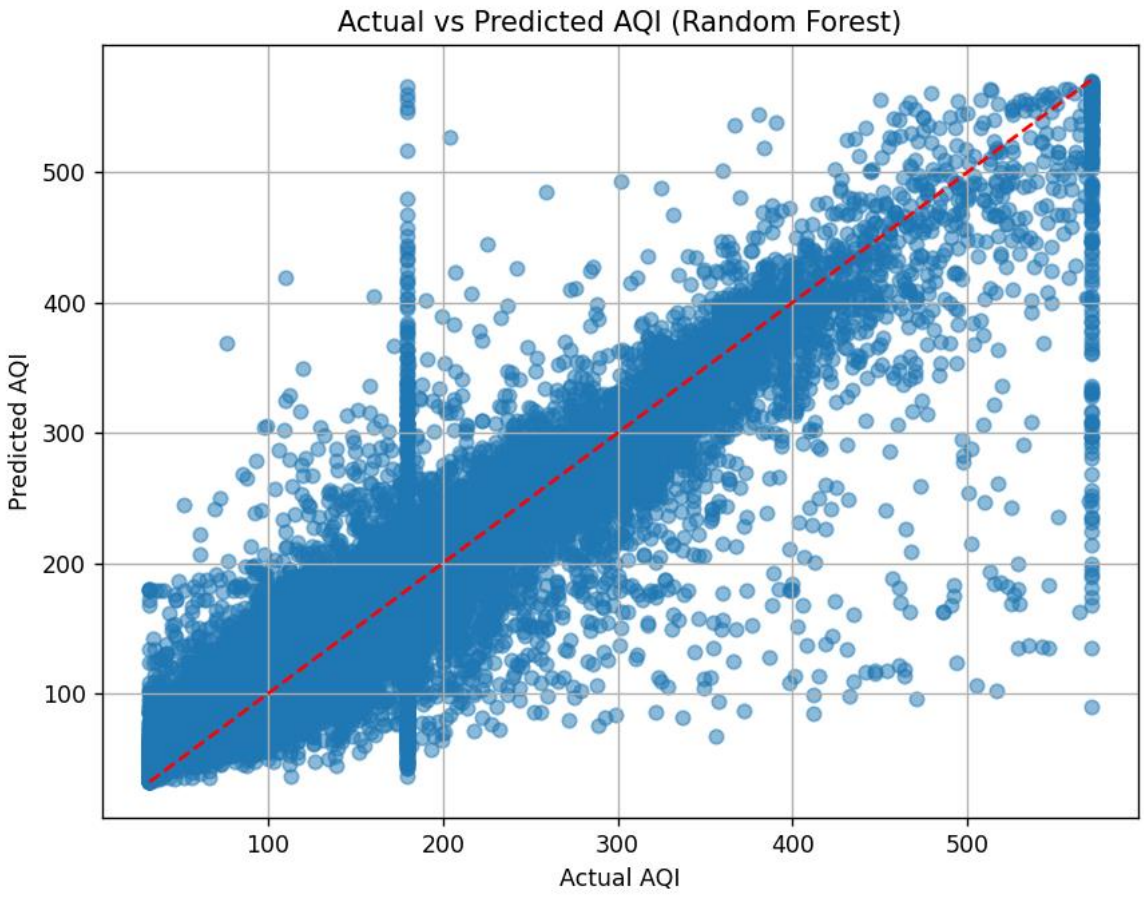
- **Nhận xét:**

- **Dự đoán tương đối tốt ở mức thấp và trung bình:** Biểu đồ cho thấy rằng, với các giá trị AQI thấp và trung bình, mô hình dự đoán khá chính xác. Các điểm dữ liệu gần đường chéo đỏ, cho thấy dự đoán gần đúng với giá trị thực tế.
- **Sự sai lệch lớn ở các giá trị AQI cao:** Tuy nhiên, khi giá trị AQI tăng lên (đặc biệt trên 400), các điểm dữ liệu bắt đầu bị phân tán rộng và không còn gần với đường chéo nữa. Điều này cho thấy mô hình hồi quy tuyến tính gặp khó khăn trong việc dự đoán chính xác các giá trị AQI cao, có thể do mô hình không đủ mạnh để nắm bắt sự thay đổi phi tuyến tính trong dữ liệu.
- **Phân bố không đồng đều của dữ liệu:** Dữ liệu bị tập trung nhiều ở một số khu vực, đặc biệt là gần trục hoành (AQI thực tế thấp) và một số khu vực có các điểm dồn lại (như các giá trị AQI xung quanh 200 và 400). Điều này có thể cho thấy sự phân bố không đồng đều của các giá trị AQI trong dữ liệu.

Kết luận:

Mô hình hồi quy tuyến tính cho thấy khả năng dự đoán tốt đối với các giá trị AQI thấp và trung bình nhưng gặp khó khăn trong việc dự đoán chính xác các giá trị AQI cao. Để cải thiện kết quả, có thể thử các mô hình phức tạp hơn, như Random Forest, Gradient Boosting hoặc XGBoost, để bắt kịp sự phi tuyến tính trong dữ liệu.

- **Random Forest**

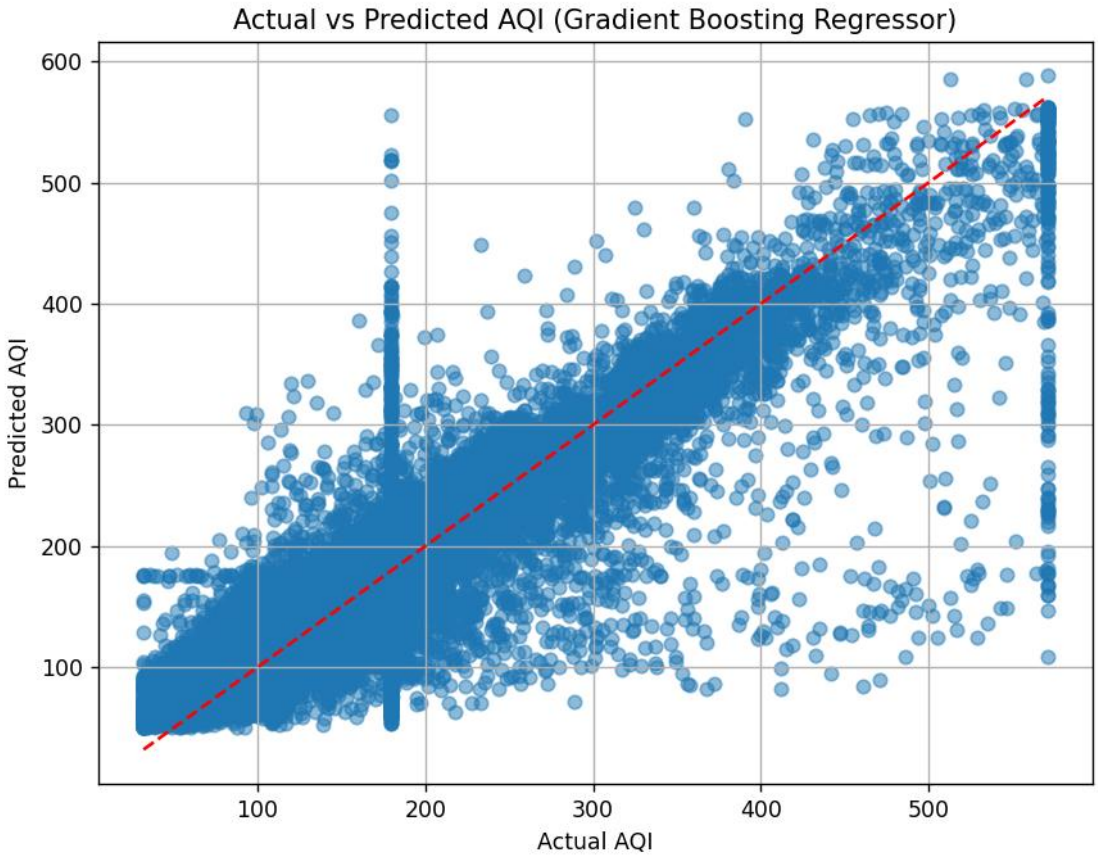


- **Nhận xét:**
  - **Dự đoán chính xác hơn ở hầu hết các giá trị:** Biểu đồ cho thấy rằng mô hình Random Forest có hiệu quả dự đoán AQI tốt hơn so với mô hình hồi quy tuyến tính. Các điểm dữ liệu nằm gần đường chéo đỏ hơn, cho thấy dự đoán gần đúng với giá trị thực tế, đặc biệt là ở các giá trị AQI trung bình.
  - **Mô hình xử lý tốt các giá trị cao:** So với mô hình hồi quy tuyến tính, Random Forest có khả năng dự đoán tốt hơn với các giá trị AQI cao (trên 400). Các điểm dữ liệu ở phần này của đồ thị không bị phân tán quá nhiều, mà gần đường chéo đỏ, cho thấy mô hình xử lý tốt các điểm ngoại lệ và giá trị lớn.
  - **Sự sai lệch nhỏ ở các giá trị thấp:** Dù mô hình Random Forest hoạt động tốt hơn, nhưng ở các giá trị AQI rất thấp, vẫn có một số điểm dự đoán bị lệch ra khỏi đường chéo. Tuy nhiên, sai lệch này không quá lớn, và mô hình vẫn thể hiện được sự chính xác cao.

Kết luận:

Mô hình Random Forest có sự cải thiện đáng kể so với mô hình hồi quy tuyến tính, đặc biệt là ở các giá trị AQI cao. Mô hình này có khả năng xử lý sự phi tuyến tính trong dữ liệu tốt hơn, làm cho dự đoán trở nên chính xác và ổn định hơn.

- **Gradient Boosting Regressor**

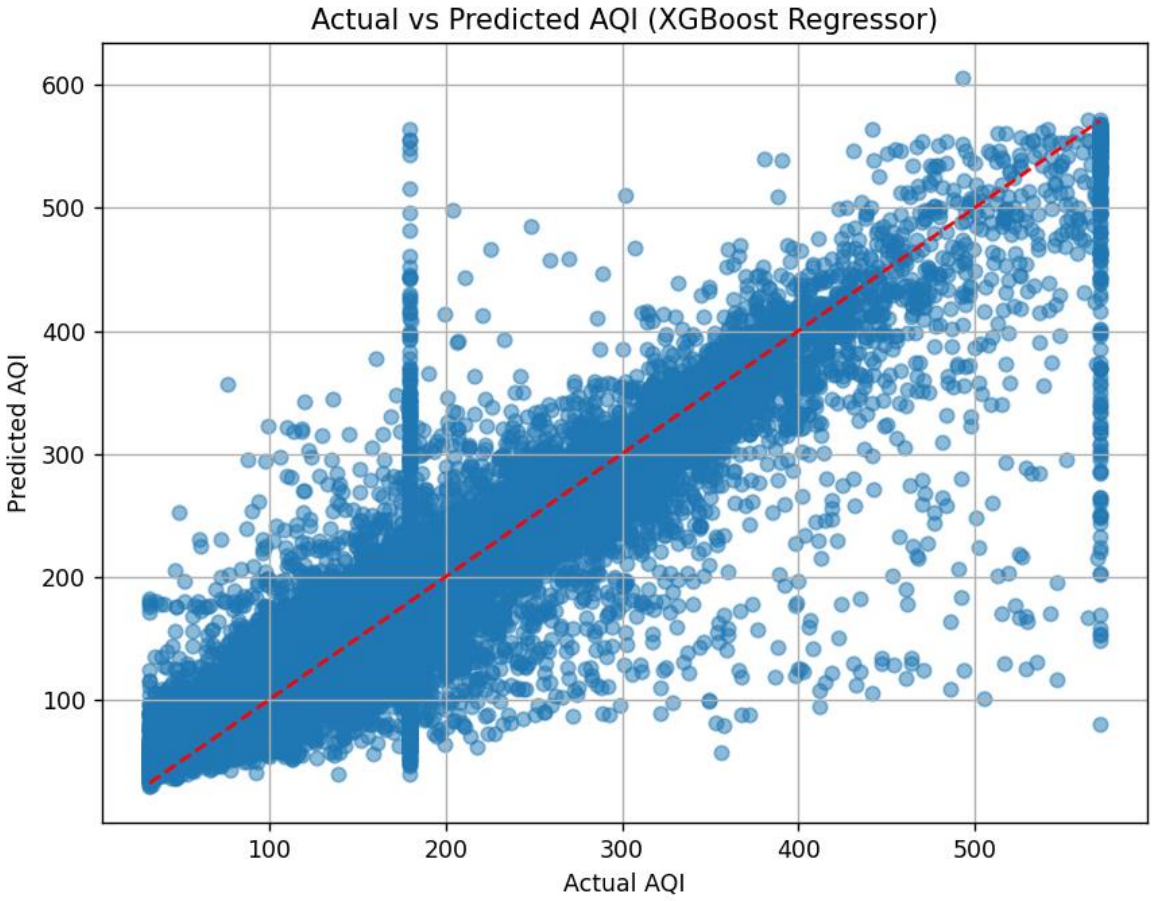


- **Nhận xét:**
  - **Dự đoán chính xác ở phần lớn các giá trị:** Biểu đồ cho thấy mô hình Gradient Boosting có khả năng dự đoán rất tốt, với các điểm dữ liệu gần đường chéo đỏ, cho thấy giá trị dự đoán gần giống với giá trị thực tế. Đặc biệt là ở các giá trị AQI trung bình và cao, mô hình này thực hiện rất tốt trong việc dự đoán chính xác.
  - **Mô hình xử lý tốt các giá trị AQI cao**: Tương tự như mô hình Random Forest, Gradient Boosting xử lý rất tốt các giá trị AQI cao (trên 400), khi các điểm dữ liệu không bị phân tán quá nhiều mà nằm gần đường chéo đỏ, cho thấy mô hình có khả năng dự đoán chính xác ngay cả với các giá trị lớn.
  - **Sự sai lệch nhỏ ở các giá trị thấp và trung bình:** So với mô hình Random Forest, mô hình Gradient Boosting có xu hướng dự đoán chính xác hơn một chút ở các giá trị thấp và trung bình. Các điểm ở vùng này cũng nằm gần đường chéo đỏ hơn, cho thấy dự đoán của mô hình này có xu hướng chính xác hơn.

Kết luận:

Mô hình Gradient Boosting cho thấy kết quả vượt trội, với khả năng dự đoán chính xác hơn, đặc biệt là ở các giá trị AQI cao. Mô hình này có thể xử lý tốt sự phi tuyến tính trong dữ liệu, mang lại độ chính xác cao hơn so với các mô hình như Linear Regression và Random Forest.

- **XGBoost Regressor**



- **Nhận xét**
  - **Dự đoán chính xác ở các giá trị thấp và trung bình:** Biểu đồ cho thấy rằng mô hình XGBoost có khả năng dự đoán chính xác ở các giá trị AQI thấp và trung bình, với các điểm dữ liệu gần đường chéo đỏ. Điều này cho thấy mô hình này đang hoạt động tốt đối với các giá trị trong phạm vi này.
  - **Dự đoán chính xác với các giá trị AQI cao:** Mô hình XGBoost có hiệu quả cao trong việc dự đoán các giá trị AQI cao (trên 400). Các điểm dữ liệu ở phần trên của đồ thị (AQI cao) gần đường chéo đỏ hơn so với các mô hình khác. Điều này cho thấy XGBoost xử lý tốt các giá trị cực trị và không bị phân tán nhiều như các mô hình khác.
  - **Sự phân tán không đáng kể ở hầu hết các giá trị:** Cũng giống như mô hình Gradient Boosting, mô hình XGBoost thể hiện sự phân tán điểm dữ liệu nhỏ và ổn định, đặc biệt là ở các giá trị cao. Điều này chỉ ra rằng mô hình này không chỉ học tốt mối quan hệ tuyến tính mà còn có khả năng xử lý các sự thay đổi phi tuyến tính trong dữ liệu.

Kết luận:

Mô hình XGBoost là một trong những mô hình mạnh mẽ nhất ở đây, với khả năng dự đoán chính xác các giá trị AQI, đặc biệt là ở các giá trị cao. XGBoost cho thấy sự cải

thiện rõ rệt so với các mô hình như Linear Regression và có hiệu suất gần bằng hoặc tốt hơn Random Forest và Gradient Boosting trong việc xử lý các giá trị phi tuyến tính và ngoại lệ.

4.4 Nhận xét và đánh giá

- **Bảng Tổng hợp Kết quả:**

Mô Hình	MAE	RMSE	R^2
Linear Regression	36.90	56.43	0.75
Random Forest Regressor	27.18	47.04	0.83
Gradient Boosting Regressor	29.53	48.95	0.81
XGBoost Regressor	27.24	46.83	0.83

- **Nhận xét:**
  - **Linear Regression**
    - **Ưu điểm:**
      - Đơn giản, dễ triển khai và có thể sử dụng nhanh chóng trong các bài toán tuyến tính.
      - Cung cấp mô hình dễ hiểu và giải thích được mối quan hệ giữa các biến.
    - **Hạn chế:**
      - Hiệu quả thấp nhất trong các mô hình đã thử ($R^2 = 0.75$), cho thấy không phù hợp với dữ liệu phi tuyến tính.
      - Không thể xử lý tốt các mối quan hệ phức tạp giữa các yếu tố trong dữ liệu.
  - **Random Forest**
    - **Ưu điểm:**
      - Hiệu suất tốt nhất ($R^2 = 0.83$), có khả năng xử lý dữ liệu phi tuyến tính và mối quan hệ phức tạp giữa các đặc trưng
      - Được đánh giá cao về khả năng giải thích kết quả và ít bị overfitting..
    - **Hạn chế:**
      - Tốn nhiều tài nguyên tính toán và thời gian huấn luyện, đặc biệt khi dữ liệu lớn.
      - Mô hình có thể trở nên khó giải thích khi số lượng cây tăng lên, làm giảm tính minh bạch.
  - **Gradient Boosting**
    - **Ưu điểm:**

        - Hiệu suất tốt ($R^2 = 0.81$), có khả năng xử lý các mối quan hệ phi tuyến tính phức tạp.
        - Hỗ trợ việc giảm thiểu sai số qua quá trình học liên tục, giúp cải thiện hiệu quả so với các mô hình đơn giản.
    - **Hạn chế:**
        - Thời gian huấn luyện lâu hơn so với các mô hình khác, điều này có thể làm giảm tính khả thi khi cần huấn luyện nhiều mô hình hoặc trong môi trường sản xuất.
    - **XGBoost Regressor:**
        - **Ưu điểm:**
            - Hiệu suất dự đoán xuất sắc ($R^2 = 0.83$), tương đương Random Forest, nhưng với thời gian huấn luyện nhanh hơn và khả năng xử lý dữ liệu phi tuyến tính rất tốt.
            - Có khả năng chống overfitting tốt nhờ các kỹ thuật như regularization, giúp cải thiện độ chính xác khi làm việc với dữ liệu phức tạp.
        - **Hạn chế:**
            - Mặc dù nhanh trong việc huấn luyện, nhưng vẫn có thể yêu cầu nhiều tài nguyên tính toán khi làm việc với dữ liệu lớn.
            - Việc tinh chỉnh tham số (hyperparameter tuning) có thể phức tạp và cần thời gian để tối ưu hóa hoàn toàn.

- **Kết Luận**

Mô hình XGBoost Regressor được đề xuất sử dụng vì những lý do sau:

- **Hiệu suất dự đoán xuất sắc ($R^2 = 0.83$):** XGBoost có khả năng dự đoán tốt với các dữ liệu phi tuyến tính và có sự ổn định trong kết quả, mang lại một mô hình mạnh mẽ và hiệu quả.
- **Thời gian huấn luyện nhanh và khả năng xử lý dữ liệu phi tuyến tính tốt:** XGBoost vượt trội hơn các mô hình như Gradient Boosting trong việc huấn luyện nhanh và chính xác. Nó có khả năng cải thiện hiệu suất dự đoán mà không yêu cầu quá nhiều tài nguyên tính toán, đặc biệt khi số lượng dữ liệu lớn hoặc yêu cầu tốc độ xử lý nhanh.
- **Khả năng tối ưu hóa:** Mặc dù việc tinh chỉnh tham số có thể yêu cầu thêm thời gian và công sức, nhưng XGBoost mang lại kết quả tuyệt vời sau khi tối ưu, đặc biệt trong các bài toán phức tạp.

CHƯƠNG 5: TỔNG KẾT

5.1 Tóm tắt nội dung thực hiện

Trong nghiên cứu này, chúng ta đã thực hiện các bước sau đây một cách có hệ thống, nhằm xây dựng một mô hình dự đoán chất lượng không khí (AQI) hiệu quả:

- **Tìm hiểu dữ liệu ban đầu**:
  - Dữ liệu được thu thập từ tập **tin station_day.csv**, chứa 108,035 bản ghi và 16 cột, bao gồm các thông tin về chỉ số ô nhiễm (PM2.5, PM10, NO, NO2, v.v.) và chất lượng không khí (AQI).
  - Phân tích sơ bộ cho thấy dữ liệu có nhiều giá trị bị thiếu và chứa các giá trị ngoại lệ.
- **Làm sạch và tiền xử lý dữ liệu**
  - Các giá trị bị thiếu được xử lý bằng phương pháp điền trung bình, đồng thời loại bỏ các cột và dòng không đạt yêu cầu về độ đầy đủ dữ liệu.
  - Loại bỏ các ngoại lệ bằng cách sử dụng ngưỡng phần trăm (percentile) để giữ lại các giá trị nằm trong khoảng 1%–99%.
  - Bổ sung cột "Season" dựa trên cột "Month", giúp phản ánh các biến động theo mùa.
- **Phân tích ma trận tương quan và chọn đặc trưng**
  - Sử dụng ma trận tương quan để xác định các đặc trưng có mối quan hệ mạnh nhất với AQI.
  - Lựa chọn 5 đặc trưng quan trọng nhất (PM2.5, PM10, NO, NO2, NOx) và bổ sung thêm biến Season vào tập đặc trưng cuối cùng.
- **Xây dựng và đánh giá mô hình dự đoán**
  Huấn luyện và đánh giá 4 mô hình hồi quy, bao gồm:
  - Hồi quy tuyến tính bội (Multiple Linear Regression)
  - Rừng ngẫu nhiên (Random Forest Regressor)
  - Gradient Boosting Regressor
  - XGBoost Regressor

  Sử dụng các chỉ số MAE, RMSE và R^2 để đánh giá hiệu suất của từng mô hình và chọn ra mô hình tốt nhất.
- **Trực quan hóa kết quả**
  - Các biểu đồ "Actual vs Predicted" minh họa hiệu suất của từng mô hình, từ đó so sánh và lựa chọn mô hình tốt nhất cho bài toán.

Toàn bộ quy trình đã được triển khai một cách logic và khoa học, đảm bảo chất lượng của dữ liệu, tính hợp lý trong việc chọn đặc trưng, và hiệu quả trong đánh giá mô hình dự đoán. Kết quả đạt được không chỉ là các chỉ số hiệu suất cao từ mô hình, mà còn

cung cấp cái nhìn sâu sắc về mối quan hệ giữa các yếu tố ô nhiễm và chất lượng không khí.

5.2 Kết quả đạt được

Trong nghiên cứu này, chúng tai đã đạt được các kết quả nổi bật sau:

- **Hoàn thành xử lý và làm sạch dữ liệu**
  - Dữ liệu gốc chứa 108,035 bản ghi với nhiều giá trị bị thiếu và ngoại lệ.
  - Sau khi xử lý, tập dữ liệu được giảm xuống còn 101,148 bản ghi với tất cả các giá trị bị thiếu được điền và các ngoại lệ được loại bỏ.
  - Thêm biến Season giúp phản ánh ảnh hưởng của thời gian đến chất lượng không khí, mang lại giá trị bổ sung trong phân tích và dự đoán.
- **Chọn lựa tập đặc trưng quan trọng:**
  - Dựa vào ma trận tương quan và kiến thức thực tiễn, 5 đặc trưng quan trọng nhất (PM2.5, PM10, NO, NO2, NOx) đã được lựa chọn.
  - Bổ sung biến Season để tăng cường khả năng dự đoán.
- **Triển khai và đánh giá 4 mô hình dự đoán:**
  - Linear Regression đạt hiệu suất cơ bản, phù hợp với dữ liệu có quan hệ tuyến tính.
  - Random Forest và Gradient Boosting thể hiện hiệu suất vượt trội nhờ khả năng xử lý phi tuyến.
  - XGBoost đạt kết quả tốt nhất với R^2 cao nhất và lỗi dự đoán nhỏ nhất.
- **Hiệu suất của các mô hình:**
  - **Linear Regression:**
    - MAE: 36.89
    - RMSE: 56.42
    - R^2: 0.75
  - **Random Forest:**
    - MAE: 26.77
    - RMSE: 46.12
    - R^2: 0.83.
  - **Gradient Boosting:**
    - MAE: 29.49
    - RMSE: 48.78
    - R^2: 0.81.
  - **XGBoost:**
    - MAE: 26.94

- RMSE: 46.18
- R^2: 0.83.

5.3 Hạn chế của đề tài

- **Phạm vi dữ liệu:**
  - Dữ liệu chỉ tập trung tại một khu vực hoặc một số trạm đo, không bao quát toàn bộ các khu vực địa lý khác nhau.
  - Một số biến tiềm năng như tốc độ gió, nhiệt độ hoặc mật độ giao thông không có sẵn trong dữ liệu, làm hạn chế khả năng dự đoán chính xác hơn.
- **Tính thời điểm:**
  - Dữ liệu chỉ mang tính lịch sử và không bao gồm các yếu tố thay đổi theo thời gian, như chính sách quản lý môi trường hoặc các sự kiện bất thường (cháy rừng, bão cát).
- **Hạn chế mô hình:**
  - Mặc dù XGBoost đạt hiệu suất cao, việc tinh chỉnh tham số chưa được thực hiện toàn diện, có thể cải thiện kết quả nếu tối ưu hóa kỹ hơn.
  - Linear Regression không phản ánh tốt quan hệ phi tuyến trong dữ liệu.
- **Chi phí tính toán:**
  - Các mô hình phức tạp như Random Forest và XGBoost yêu cầu nhiều tài nguyên tính toán, làm tăng thời gian huấn luyện.

5.4 Định hướng phát triển trong tương lai

- **Mở rộng dữ liệu**
  - Thu thập thêm dữ liệu từ nhiều khu vực khác nhau để tăng tính đại diện.
  - Bổ sung các đặc trưng như tốc độ gió, độ ẩm, mật độ giao thông, và dữ liệu vệ tinh về môi trường.
- **Tích hợp yếu tố thời gian**
  - Xây dựng mô hình theo chuỗi thời gian (time-series) để dự đoán AQI trong tương lai, không chỉ dựa trên dữ liệu tĩnh.
- **Tối ưu hóa mô hình**
  - Sử dụng các thuật toán tìm kiếm siêu tham số như Grid Search hoặc Bayesian Optimization để cải thiện hiệu suất của các mô hình phức tạp như Gradient Boosting hoặc XGBoost.
- **Ứng dụng thực tiễn**
  - Phát triển một hệ thống dự báo AQI thời gian thực dựa trên các mô hình đã xây dựng.
  - Tích hợp mô hình vào các ứng dụng hoặc nền tảng web để cung cấp dự báo chất lượng không khí cho người dân.

- **Khám phá các mô hình mới**
  - Thử nghiệm các mô hình học sâu (Deep Learning) như mạng nơ-ron nhân tạo (ANN) hoặc mạng nơ-ron hồi quy (RNN) để cải thiện hiệu suất.

5.5 Kết luận

Đề tài này đã hoàn thành một cách hệ thống và toàn diện, từ làm sạch dữ liệu, phân tích đặc trưng, xây dựng mô hình dự đoán đến đánh giá hiệu suất. Việc sử dụng XGBoost và các mô hình phức tạp khác đã minh chứng hiệu quả trong dự đoán chỉ số AQI, với khả năng áp dụng vào thực tế để hỗ trợ quản lý môi trường và bảo vệ sức khỏe cộng đồng.

Tuy nhiên, nghiên cứu vẫn còn những hạn chế nhất định, đặc biệt trong phạm vi dữ liệu và mô hình hóa yếu tố thời gian. Với những định hướng phát triển trong tương lai, nghiên cứu này có tiềm năng trở thành cơ sở cho các ứng dụng dự báo chất lượng không khí và cung cấp giải pháp hữu ích trong việc cải thiện môi trường sống.

HẾT