Title: Delhi Air Quality Index (AQI) Analysis

Name: Prateek Sharma

Date: 04 October 2024

Objective: The objective of this project is to analyze air quality data to assess the levels and trends of various pollutants such as CO, NO, NO_2, O_3, SO_2, PM2.5, PM10, and NH_3 over time. By examining the data, we aim to uncover patterns, understand the relationship between different pollutants, and identify any anomalies or extreme values. This analysis will provide insights that could be valuable for evaluating air quality, understanding pollution dynamics, and supporting environmental decision-making.

Data Source: The dataset used for this analysis was obtained from Kaggle, which hosts various open-source datasets. This specific dataset contains information on air pollutant levels, including concentrations of CO, NO, NO_2, O_3, SO_2, PM2.5, PM10, and NH_3, recorded over multiple dates.

Data Preparation and Cleaning:

After downloading the raw dataset, I organized it in a well-structured folder system. Specifically, I created a folder named New Delhi AQI with subfolders for each phase of the project, such as Raw Data, Cleaned Data, Analysis, and Reports. The raw dataset was stored in the Raw Data subfolder.

Here are the detailed steps for the data cleaning process:

Loading Dataset: I used the pandas library to load the dataset from the Raw Data folder and display the first five rows.

```
import pandas as pd

import numpy as np

import os
```

```
# Checking the current working directory

print(os.getcwd())


# Loading the dataset

df = pd.read_csv('..\Raw Data\delhi_aqi - delhi_aqi.csv')

df.head()
```

The dataset contained air quality measurements for New Delhi, with columns such as CO, NO, NO2, O3, SO2, PM2.5, PM10, and NH3. Each column was in Âµmole units but unnecessarily repeated this unit in the data itself, so I proceeded to clean this.

Removing Unnecessary Text from Cells: I removed the ÂÂµmole unit from all the data entries and updated the column headers to reflect that the data is already in ÂÂµmole.

```
# Remove 'ÂÂµmole ' from the cells and update the column headers

df = df.replace({'Âµmole': ''}, regex=True)

df = df.rename(columns={
    'date': 'Date',
    'co': 'CO (ÂÂµmole)',
    'no': 'NO (ÂÂµmole)',
    'no2': 'NO2 (ÂÂµmole)',
    'o3': 'O3 (ÂÂµmole)',
    'so2': 'SO2 (ÂÂµmole)',
    'pm2_5': 'PM2.5 (ÂÂµmole)',
    'pm10': 'PM10 (ÂÂµmole)',
```

```
    'nh3': 'NH3 (ÂÂµmole)'
})
```

Data Inspection: I checked the dataset structure and data types to understand its current state.

```
# Checking the dataset shape
df.shape

# Displaying information about the dataset
df.info()
```

The Date column was stored as an object type, so it needed conversion to datetime.

Converting Date to datetime Format: I converted the Date column to the correct datetime format for proper time-series analysis.

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

Converting Object Types to Numeric: The other columns, which should be numeric, were in object format. I converted them to float.

```
# Converting numeric columns from object to float
```

```
columns_to_convert = ['CO (Âµmole)', 'NO (Âµmole)', 'NO2 (Âµmole)', 'O3 (Âµmole)', 'SO2 (Âµmole)', 'PM2.5 (Âµmole)', 'PM10 (Âµmole)', 'NH3 (Âµmole)']

df[columns_to_convert] = df[columns_to_convert].astype(float)
```

Handling Duplicate Rows: I checked for and identified duplicated rows, particularly duplicate dates, and handled them appropriately.

```
# Checking for duplicate rows

df.duplicated().sum()


# Checking for duplicates in Date

duplicate_dates = df[df.duplicated(subset='Date')]

if not duplicate_dates.empty:

   print("Duplicate Dates Found:")

   print(duplicate_dates)
```

Checking for Null Values: I ensured there were no missing values in the dataset.

```
# Checking for null values

df.isnull().sum()
```

Identifying Negative Values: I checked the dataset for any negative values, which would be unrealistic for air quality metrics, and identified columns containing such values if any existed.

```
# Dropping the 'Date' column for numeric value checks
numeric_df = df.drop(columns=['Date'])

# Check for negative values in the numeric columns
negative_values = (numeric_df < 0).any()

# Display the columns that contain negative values
negative_columns = negative_values[negative_values].index.tolist()

if negative_columns:
    print(f"Columns with negative values: {negative_columns}")
else:
    print("No negative values found.")
```

Final Cleaned Data: After completing these steps, the dataset was cleaned and prepared for analysis, ready for the next phase of the project.

The final shape of the cleaned data was:

```
df.shape
```

Output:

```
(18776, 9)
```

The dataset is now fully cleaned, with proper data types and no missing or invalid values. This cleaned data will be used for further analysis.

Data Analysis:

The data analysis process began with loading the cleaned dataset into a pandas DataFrame for further exploration and analysis. The dataset contains various pollutants such as CO, NO, NO2, O3, SO2, PM2.5, PM10, and NH3, along with the date of observation.

1. Importing Libraries and Loading Data

The required Python libraries for data manipulation and visualization were imported, including pandas, numpy, seaborn, and matplotlib. The dataset was loaded using the following command:

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt


cdf = pd.read_csv("../Clean Data/clean_data.csv")
```

2. Summary Statistics

To understand the distribution of the data, summary statistics were computed using the describe() method. This provided insights into key measures like the count, mean, standard deviation, and percentiles for each pollutant.

```
cdf.describe()
```

3. Distribution Analysis

For a visual understanding of the data distribution, histograms were plotted for each pollutant using Seaborn's histplot() function. This step helped in identifying the distribution patterns and the presence of any skewness or abnormalities in the data.

```
for col in cdf.columns[1:]:  # Skipping the date column
    plt.figure(figsize=(8, 3))
    sns.histplot(cdf[col], kde=True)
    plt.title(f"Distribution of {col}")
    plt.show()
```

4. Correlation Matrix

To explore the relationships between different pollutants, a correlation matrix was computed. This helped in identifying how strongly pollutants were related to each other. The correlation matrix was visualized using a heatmap.

```
corr_mat = cdf.drop('Date', axis=1).corr()


# Correlation Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(corr_mat, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Matrix of Pollutants')
plt.show()
```

5. Outlier Detection using Boxplots

To check for the presence of outliers and to better understand the distribution of the pollutant values, boxplots were generated for each pollutant. Boxplots are a useful tool for detecting extreme values that could potentially affect the analysis.

```
for col in cdf.columns[1:]:
    plt.figure(figsize=(8, 4))
    sns.boxplot(x=cdf[col])
    plt.title(f"Boxplot of {col}")
    plt.show()
```

6. Data Aggregation by Month

To simplify the analysis and focus on monthly trends, the dataset was aggregated by month. The resample() function was used to group the data based on the date column, calculating the mean for each pollutant over each month.

```
cdf['Date'] = pd.to_datetime(cdf['Date'])

# Aggregate by month
cdf_monthly = cdf.resample('M', on='Date').mean()
```

Key Findings:

Carbon Monoxide (CO)

- **Concentration Levels -** The histogram shows that most CO levels are concentrated between 0 and 5000 Âµmole, with a peak around 2000 Âµmole, indicating generally low pollution levels.
- **Correlation with Other Gases -** CO has a strong positive correlation with both NO (0.91) and NO2 (0.78), suggesting they likely share similar sources, particularly from combustion processes.
- **Outliers -** The box plot reveals significant outliers, indicating sporadic high pollution events that could be tied to specific industrial activities or traffic patterns.

Nitric Oxide (NO)

- **Frequency and Levels -** NO readings predominantly peak at 0 Âµmole, indicating low ambient levels, yet there are significant observations up to 10 Âµmole.
- **Strong Correlation -** A high correlation with CO (0.91) and PM2.5 (0.82) suggests that NO emissions are closely linked with carbon emissions and particulate matter, pointing towards combustion as a common source.
- **Seasonal Trends -** The line graph shows periodic peaks, especially in early 2022, indicating seasonal variations in NO emissions, potentially tied to increased vehicle usage or industrial activities during certain months.

Nitrogen Dioxide (NO2)

- **Concentration Range -** NO2 shows a concentration range primarily below 100 Âµmole, with a peak around 70 Âµmole, indicating moderate pollution levels.
- **Correlation Insights -** Similar to NO, NO2 is positively correlated with CO (0.78) and PM2.5 (0.75), reinforcing the idea of shared emission sources.
- **Outlier Events -** The box plot displays several outliers, indicating that high NO2 events can occur, which may be associated with urban pollution spikes.

Ozone (O3)

- **Behavior Patterns -** The histogram indicates a significant frequency of ozone levels at 0 Âµmole, with a gradual increase to 200 Âµmole. This behavior reflects good air quality concerning ozone.
- **Inverse Correlation -** Ozone shows negative correlations with CO (-0.4), NO (-0.35), and NO2 (-0.34). This suggests that while other pollutants may originate from direct emissions, ozone formation is more complex, potentially resulting from photochemical reactions in the atmosphere rather than direct emissions.
- **Seasonal Variation -** The time series line graph highlights peaks in ozone levels in mid-2021, suggesting that ozone formation is influenced by seasonal weather patterns, particularly increased sunlight in warmer months that promote photochemical reactions.

Sulfur Dioxide (SO2)

- **Concentration Levels -** SO2 levels primarily peak around 50 Âµmole, with minimal high concentrations, indicating effective control measures against sulfur emissions.
- **Correlation with Other Gases -** SO2 does not show strong correlation with CO or NO, indicating it may originate from different sources, such as industrial processes rather than vehicular emissions.
- **Controlled Emissions -** The box plot indicates few outliers, suggesting that overall sulfur dioxide emissions are well-regulated, reflecting effective policy measures.

Ammonia (NH3)

- **Concentration and Frequency -** NH3 concentrations peak at 5 µg/m³, indicating relatively low levels of ammonia in the air.
- **Limited Correlation -** The correlations with other pollutants are weaker, indicating that NH3 emissions may originate from agricultural practices rather than the combustion sources linked to CO, NO, and NO2.
- **Stable Levels -** The line graph shows stable low levels of ammonia over time, indicating effective control in managing ammonia emissions.