Report On

Air Quality Index Analysis

Submitted in partial fulfillment of the requirements of the Course project in Semester VII of Fourth Year Artificial Intelligence and Data Science

by
Parth Puri (Roll No. 23)
Karan Patra (Roll No. 20)
Khushi Tiwari (Roll No. 28)
Reena Vaidya (Roll No. 31)

Supervisor
Prof. Bhavika Gharat

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science

(2023-24)

Vidyavardhini's College of Engineering & Technology
Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that the project entitled “Title of the project” is a bonafide work of "Parth Puri (Roll No. 23), Reena Vaidya (Roll No. 31), Khushi Tiwari (Roll No. 28), Karan Patra (Roll No. 20)" submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester VII of Second Year Artificial Intelligence and Data Science engineering.

Supervisor
Prof. Bhavika Gharat

Dr. Tatwadarshi P. N.
Head of Department

Table of Contents

Problem Statement

The report addresses the need for comprehensive Air Quality Index (AQI) analysis, focusing on the significance of monitoring and assessing air quality in specific regions. It emphasizes the importance of AQI in public health and environmental management and aims to provide an in-depth understanding of the AQI analysis process.

Air quality is not uniform worldwide. Different regions face distinct challenges, from industrial emissions in urban areas to natural events like forest fires. The problem statement acknowledges that AQI analysis should be region-specific and adaptable to the unique environmental conditions and pollutant sources in different areas.

AQI analysis is not a straightforward task. Air quality is affected by a multitude of factors, and AQI is influenced by the concentration of various pollutants. The problem statement recognizes the complexity of dealing with diverse data sources, handling missing or inconsistent data, and ensuring the accuracy and reliability of AQI calculations.

Description

- Data Collection: Involves gathering air quality data from various sources.
- Data Cleaning and Preprocessing: Focuses on data quality enhancement.
- AQI Calculation: Utilizes standardized formulas to compute AQI values.
- Data Visualization: Represents AQI data through various visualization techniques.
- Comparison with Standards: Assesses AQI against air quality standards.
- Practical Implementation: Demonstrates AQI analysis using Python and a specific dataset.
- Correlation Analysis: Explores relationships between air pollutants.

Module Description

The report discusses the primary modules used in AQI analysis, including data collection, data cleaning, AQI calculation, data visualization, and correlation analysis. Each module plays a crucial role in the overall analysis process.

Correlation analysis explores the relationships between different air pollutants within the dataset. The resulting correlation matrix provides insights into the strength and direction of the linear relationships between variables, helping identify shared sources or pollution patterns among specific pollutants.

Correlation analysis enhances our understanding of the interplay between various pollutants and can inform strategies for mitigating air quality issues more effectively. It provides a holistic view of air quality dynamics.
Each of these modules contributes to a comprehensive AQI analysis, ensuring that the assessment of air quality is thorough, accurate, and actionable for the betterment of public health and environmental management.
In this module, the calculated AQI metrics are compared with recommended air quality standards established by environmental agencies or regulatory bodies. These standards set thresholds for various pollutants, and the AQI is used to assess whether air quality in a particular location falls within acceptable limits.
This module illustrates the practical implementation of AQI analysis using Python and a well-suited dataset. It provides a step-by-step guide on how to calculate AQI values, visualize data, and compare them with established standards. The practical implementation demonstrates the real-world application of AQI analysis.

Brief Description of Software & Hardware Used and Its Programming

- Software: Python is the primary programming language used for AQI analysis, and various libraries, such as Pandas, Matplotlib, and NumPy, are employed for data manipulation, visualization, and analysis.
- Hardware: The analysis can be performed on standard computer hardware with adequate processing capabilities.

Code

```
import pandas as pd
import plotly.express as px
import plotly.io as pio
import plotly.graph_objects as go
pio.templates.default = "plotly_white"

data = pd.read_csv("/content/drive/MyDrive/BDA/delhiaqi.csv")
print(data.head())

"""I’ll convert the date column in the dataset into a datetime data type
and move forward:"""

data['date'] = pd.to_datetime(data['date'])

"""Now, let’s have a look at the descriptive statistics of the data:"""

print(data.describe())
```

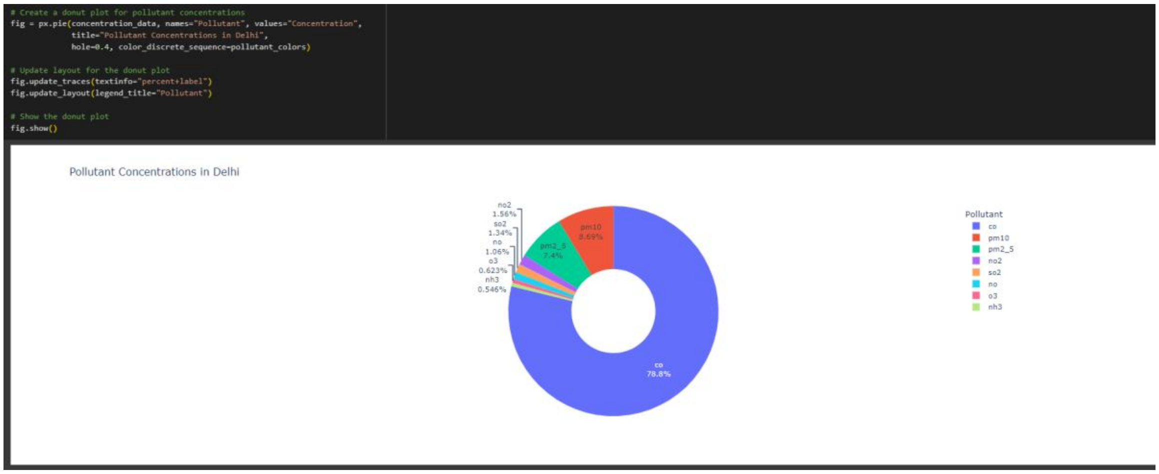
```python
"""Now let's have a look at the intensity of each pollutant over time in
the air quality:"""

# time series plot for each air pollutant
fig = go.Figure()

for pollutant in ['co', 'no', 'no2', 'o3', 'so2', 'pm2_5', 'pm10', 'nh3']:
    fig.add_trace(go.Scatter(x=data['date'], y=data[pollutant],
mode='lines',
                             name=pollutant))

fig.update_layout(title='Time Series Analysis of Air Pollutants in Delhi',
                  xaxis_title='Date', yaxis_title='Concentration (µg/m³)')
fig.show()

"""Now, before moving forward, we need to calculate the air quality index
and its category. AQI is typically computed based on the concentration of
various pollutants, and each pollutant has its sub-index. Here's how we
can calculate AQI:"""

# Define AQI breakpoints and corresponding AQI values
aqi_breakpoints = [
    (0, 12.0, 50), (12.1, 35.4, 100), (35.5, 55.4, 150),
    (55.5, 150.4, 200), (150.5, 250.4, 300), (250.5, 350.4, 400),
    (350.5, 500.4, 500)
]

def calculate_aqi(pollutant_name, concentration):
    for low, high, aqi in aqi_breakpoints:
        if low <= concentration <= high:
            return aqi
    return None

def calculate_overall_aqi(row):
    aqi_values = []
    pollutants = ['co', 'no', 'no2', 'o3', 'so2', 'pm2_5', 'pm10', 'nh3']
    for pollutant in pollutants:
        aqi = calculate_aqi(pollutant, row[pollutant])
        if aqi is not None:
            aqi_values.append(aqi)
    return max(aqi_values)

# Calculate AQI for each row
data['AQI'] = data.apply(calculate_overall_aqi, axis=1)

# Define AQI categories
aqi_categories = [
    (0, 50, 'Good'), (51, 100, 'Moderate'), (101, 150, 'Unhealthy for
Sensitive Groups'),
```

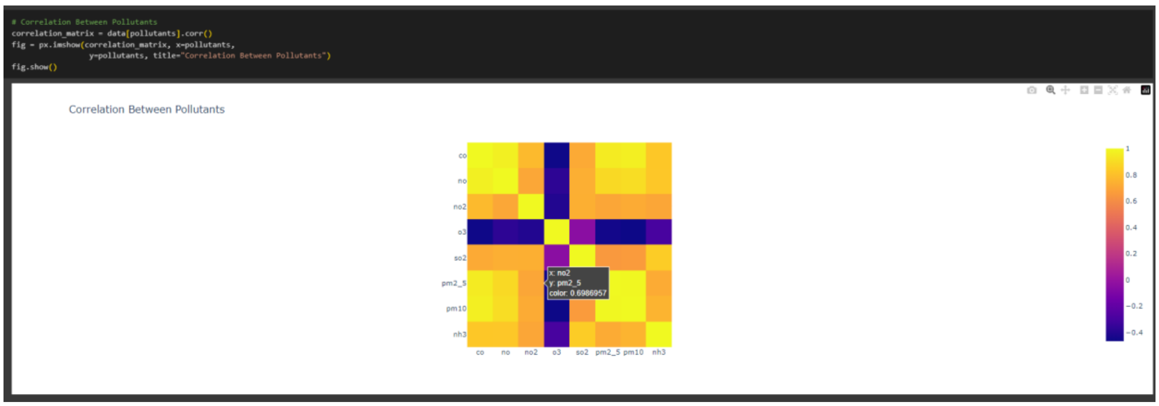
```python
    (151, 200, 'Unhealthy'), (201, 300, 'Very Unhealthy'), (301, 500,
'Hazardous')
]

def categorize_aqi(aqi_value):
    for low, high, category in aqi_categories:
        if low <= aqi_value <= high:
            return category
    return None

# Categorize AQI
data['AQI Category'] = data['AQI'].apply(categorize_aqi)
print(data.head())

"""In the above code, we are defining AQI breakpoints and corresponding
AQI values for various air pollutants according to the Air Quality Index
(AQI) standards. The aqi_breakpoints list defines the concentration ranges
and their corresponding AQI values for different pollutants. We then
define two functions:

calculate_aqi: to calculate the AQI for a specific pollutant and
concentration by finding the appropriate range in the aqi_breakpoints
calculate_overall_aqi: to calculate the overall AQI for a row in the
dataset by considering the maximum AQI value among all pollutants
The calculated AQI values are added as a new column in the dataset.
Additionally, we defined AQI categories in the aqi_categories list and
used the categorize_aqi function to assign an AQI category to each AQI
value. The resulting AQI categories are added as a new column as AQI
Category in the dataset.

Analyzing AQI of Delhi
Now, let’s have a look at the AQI of Delhi in January:
"""

# AQI over time
fig = px.bar(data, x="date", y="AQI",
             title="AQI of Delhi in January")
fig.update_xaxes(title="Date")
fig.update_yaxes(title="AQI")
fig.show()

"""Now, let’s have a look at the AQI category distribution:"""

fig = px.histogram(data, x="date",
                    color="AQI Category",
                    title="AQI Category Distribution Over Time")
fig.update_xaxes(title="Date")
fig.update_yaxes(title="Count")
fig.show()
```

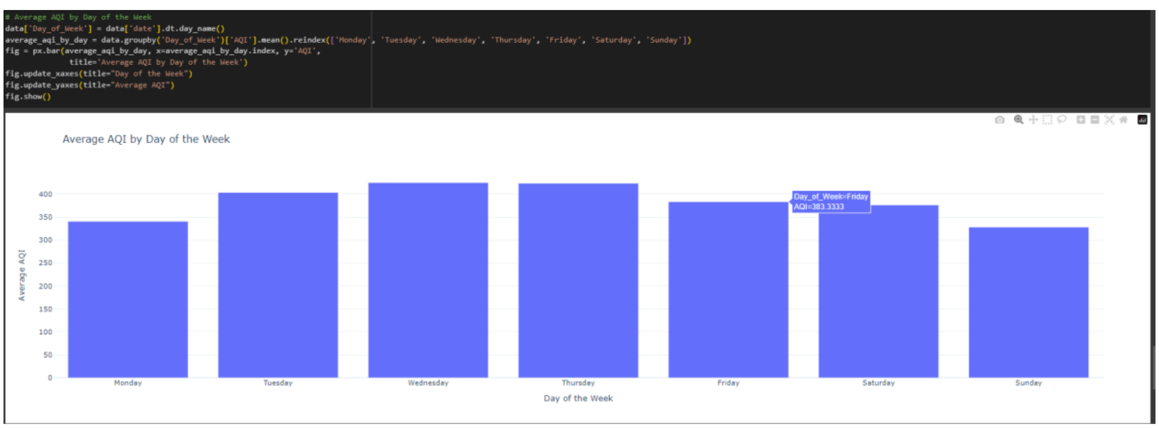
```python
"""Now, let's have a look at the distribution of pollutants in the air
quality of Delhi:"""

# Define pollutants and their colors
pollutants = ["co", "no", "no2", "o3", "so2", "pm2_5", "pm10", "nh3"]
pollutant_colors = px.colors.qualitative.Plotly

# Calculate the sum of pollutant concentrations
total_concentrations = data[pollutants].sum()

# Create a DataFrame for the concentrations
concentration_data = pd.DataFrame({
    "Pollutant": pollutants,
    "Concentration": total_concentrations
})

# Create a donut plot for pollutant concentrations
fig = px.pie(concentration_data, names="Pollutant",
values="Concentration",
             title="Pollutant Concentrations in Delhi",
             hole=0.4, color_discrete_sequence=pollutant_colors)

# Update layout for the donut plot
fig.update_traces(textinfo="percent+label")
fig.update_layout(legend_title="Pollutant")

# Show the donut plot
fig.show()

"""Now, let's have a look at the correlation between pollutants:"""

# Correlation Between Pollutants
correlation_matrix = data[pollutants].corr()
fig = px.imshow(correlation_matrix, x=pollutants,
                 y=pollutants, title="Correlation Between Pollutants")
fig.show()

"""The correlation matrix displayed here represents the correlation
coefficients between different air pollutants in the dataset. Correlation
coefficients measure the strength and direction of the linear relationship
between two variables, with values ranging from -1 to 1. Overall, the
positive correlations among CO, NO, NO2, SO2, PM2.5, PM10, and NH3 suggest
that they may share common sources or have similar pollution patterns,
while O3 exhibits an inverse relationship with the other pollutants, which
may be due to its role as both a pollutant and a natural atmospheric
oxidant.

Now, let's have a look at the hourly average trends of AQI in Delhi:
```

```
"""

# Extract the hour from the date
data['Hour'] = pd.to_datetime(data['date']).dt.hour

# Calculate hourly average AQI
hourly_avg_aqi = data.groupby('Hour')['AQI'].mean().reset_index()

# Create a line plot for hourly trends in AQI
fig = px.line(hourly_avg_aqi, x='Hour', y='AQI',
              title='Hourly Average AQI Trends in Delhi (Jan 2023)')
fig.update_xaxes(title="Hour of the Day")
fig.update_yaxes(title="Average AQI")
fig.show()

"""Now, let’s have a look at the average AQI by day of the week in
Delhi:"""

# Average AQI by Day of the Week
data['Day_of_Week'] = data['date'].dt.day_name()
average_aqi_by_day =
data.groupby('Day_of_Week')['AQI'].mean().reindex(['Monday', 'Tuesday',
'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'])
fig = px.bar(average_aqi_by_day, x=average_aqi_by_day.index, y='AQI',
              title='Average AQI by Day of the Week')
fig.update_xaxes(title="Day of the Week")
fig.update_yaxes(title="Average AQI")
fig.show()

"""It shows that the air quality in Delhi is worse on Wednesdays and
Thursdays.

Summary
Air quality index (AQI) analysis is a crucial aspect of environmental data
science that involves monitoring and analyzing air quality in a specific
location. It aims to provide a numerical value representative of overall
air quality, essential for public health and environmental management.
"""
```

Results and Conclusion







The report presents the results of AQI analysis using Python, the calculated AQI values for the Delhi air quality dataset, and correlation analysis findings. It concludes with the significance of AQI analysis in environmental data science and its implications for public health and environmental management.

References

Lave, L. B., & Seskin, E. P. (1977). Air pollution and human health. Science, 198(4327), 143-148.

U.S. Environmental Protection Agency. (2019). Air quality index (AQI) - A guide to air quality and your health. Retrieved from https://www.airnow.gov/aqi/aqi-basics/

Dominici, F., Peng, R. D., Bell, M. L., Pham, L., McDermott, A., Zeger, S. L., & Samet, J. M. (2006). Fine particulate air pollution and hospital admission for cardiovascular and respiratory diseases. JAMA, 295(10), 1127-1134.

World Health Organization. (2005). Air quality guidelines: Global update 2005. Retrieved from https://www.who.int/phe/health_topics/outdoorair/outdoorair_aqg/en/

Python Software Foundation. (2021). Python Language Reference, version 3.9.6. Retrieved from https://docs.python.org/3/reference/

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference, 51-56.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95.

Pandas Development Team. (2021). Pandas: Powerful data structures for data analysis. Retrieved from https://pandas.pydata.org/

Kim, K. H., Kabir, E., & Kabir, S. (2015). A review on the human health impact of airborne particulate matter. Environment International, 74, 136-143.

United States National Library of Medicine. (2020). MedlinePlus: Air pollution. Retrieved from https://medlineplus.gov/airpollution.html