**SHADOW FOX INTERNSHIP**

**TASK-2 DOCUMENTATION**

**Air Quality Index (AQI) Analysis and Visualization in Delhi**

**CONTEXT**

|  |
| --- |
| * **Overview** * **Loading and Preprocessing Data** * **Calculating AQI** * **Visualizing Pollutant Concentrations Over Time** * **Visualizing AQI Categories** * **Analyzing AQI Trends Over Time** * **Correlation Analysis Between Pollutants** * **Visualizing Hourly Average Rate of Pollutants** * **Visualizing Hourly Average AQI** * **Visualizing Average AQI by Day of the Week** * **Visualizing Average AQI on Weekends** * **Visualizing Pollutants' Average AQI Contribution** |

**Overview:**

This project aims to analyze and visualize the Air Quality Index (AQI) in Delhi using data collected from various monitoring stations. The analysis includes calculating the AQI for different pollutants, visualizing their contributions, and exploring trends over time.

**Data Source:**

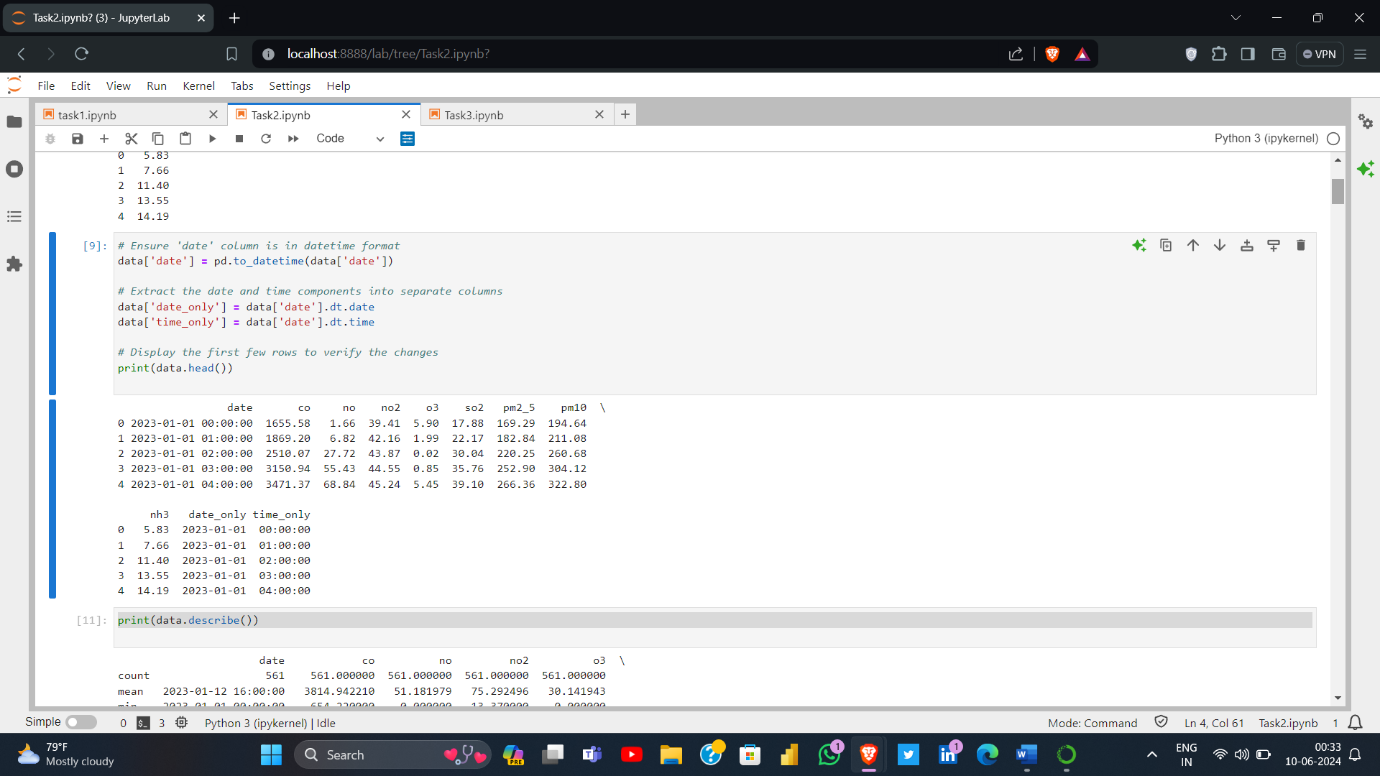
The data used in this project is stored in a CSV file named delhiaqi.csv,downloaded from Kaggle. The dataset contains air quality readings for various pollutants along with their respective timestamps.

**1. Loading and Preprocessing Data**

### Loading Data

The data is loaded using Pandas:

|  |
| --- |
| import pandas as pd  # Load the data  data = pd.read\_csv("Downloads/delhiaqi.csv")  print(data.head()) |

 Converting to Datetime

**2. Calculating AQI**

AQI Breakpoints

* Define AQI breakpoints and corresponding AQI values:

|  |
| --- |
| # 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(concentration, breakpoints):  for low, high, aqi in breakpoints:  if low <= concentration <= high:  return aqi  return None  def calculate\_overall\_aqi(row, pollutants, breakpoints):  aqi\_values = []  for pollutant in pollutants:  concentration = row[pollutant]  aqi = calculate\_aqi(concentration, breakpoints)  if aqi is not None:  aqi\_values.append(aqi)  return max(aqi\_values) if aqi\_values else None  # List of pollutants  pollutants = ['co', 'no', 'no2', 'o3', 'so2', 'pm2\_5', 'pm10', 'nh3']  # Calculate AQI for each row  data['AQI'] = data.apply(lambda row: calculate\_overall\_aqi(row, pollutants, aqi\_breakpoints), axis=1) |

Categorize AQI

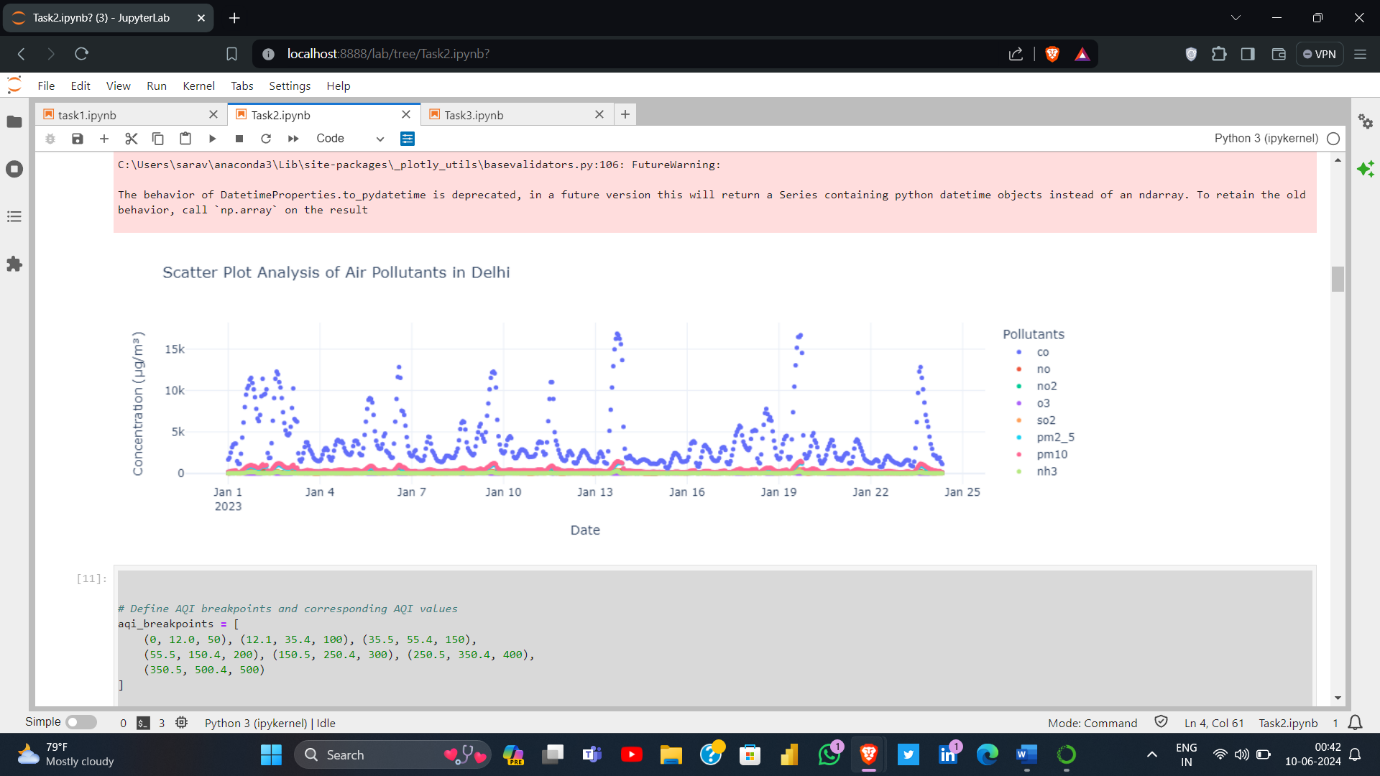
Define AQI categories and categorize each AQI value:

|  |
| --- |
| # Define AQI categories  aqi\_categories = [  (0, 50, 'Good'), (51, 100, 'Moderate'), (101, 150, 'Unhealthy for Sensitive Groups'),  (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) |

**3. Visualizing Pollutant Concentrations Over Time**

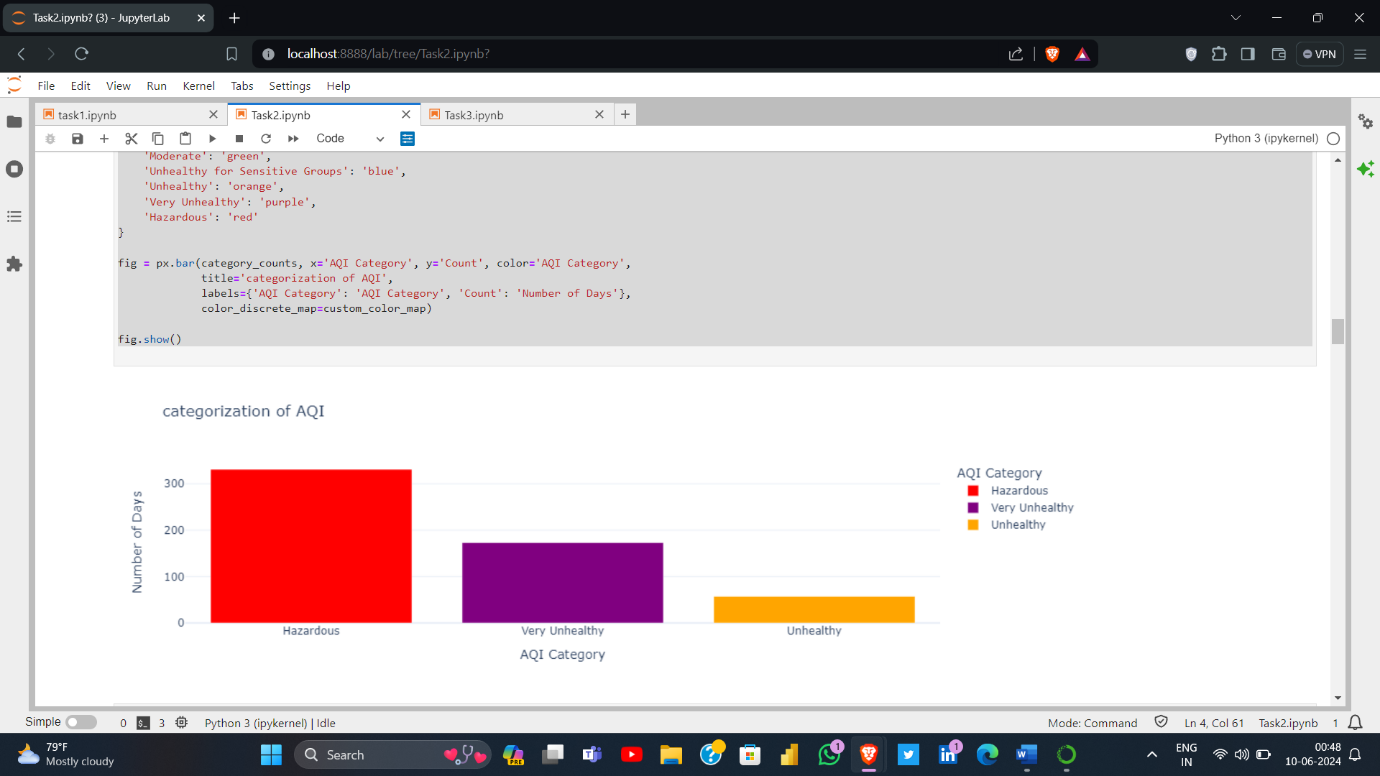
Scatter Plot of Pollutants

* Visualize the concentration of pollutants over time using scatter plots:



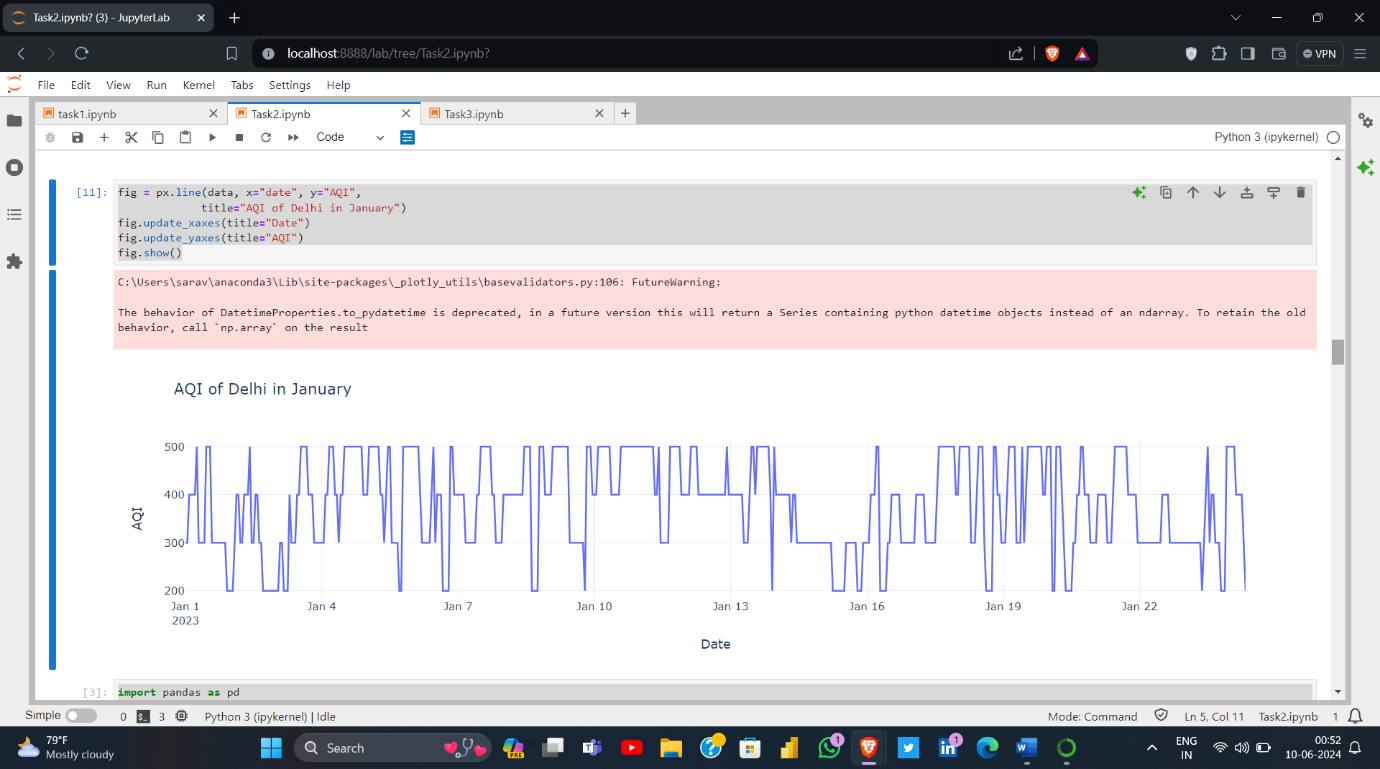
4. Visualizing AQI Categories

Bar Plot of AQI Categories

* To Visualize the distribution of AQI using barplot with different categories and finding out them:

**5. Analyzing AQI Trends Over Time**

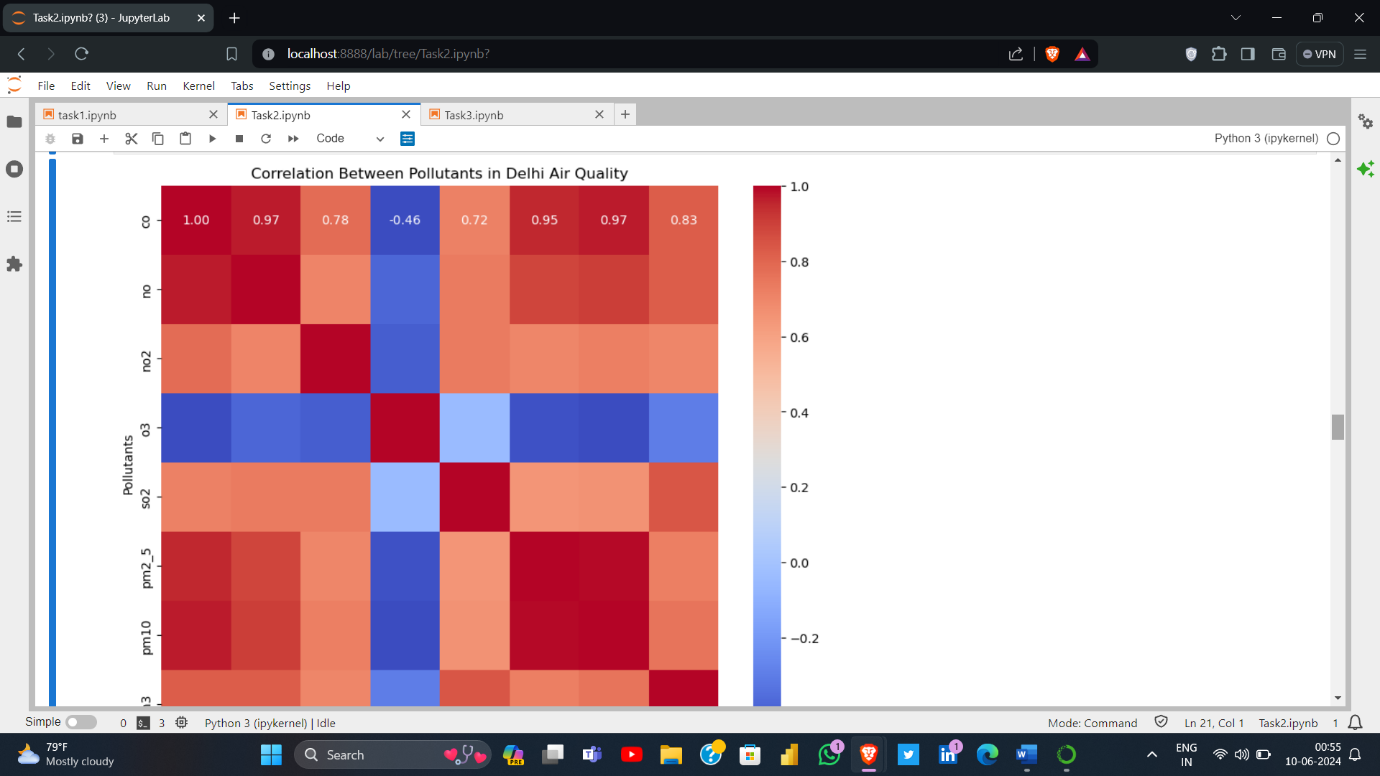
Line Graph of AQI Over Time

* Visualize the AQI trends over time using a line graph:

6. Correlation Analysis Between Pollutants

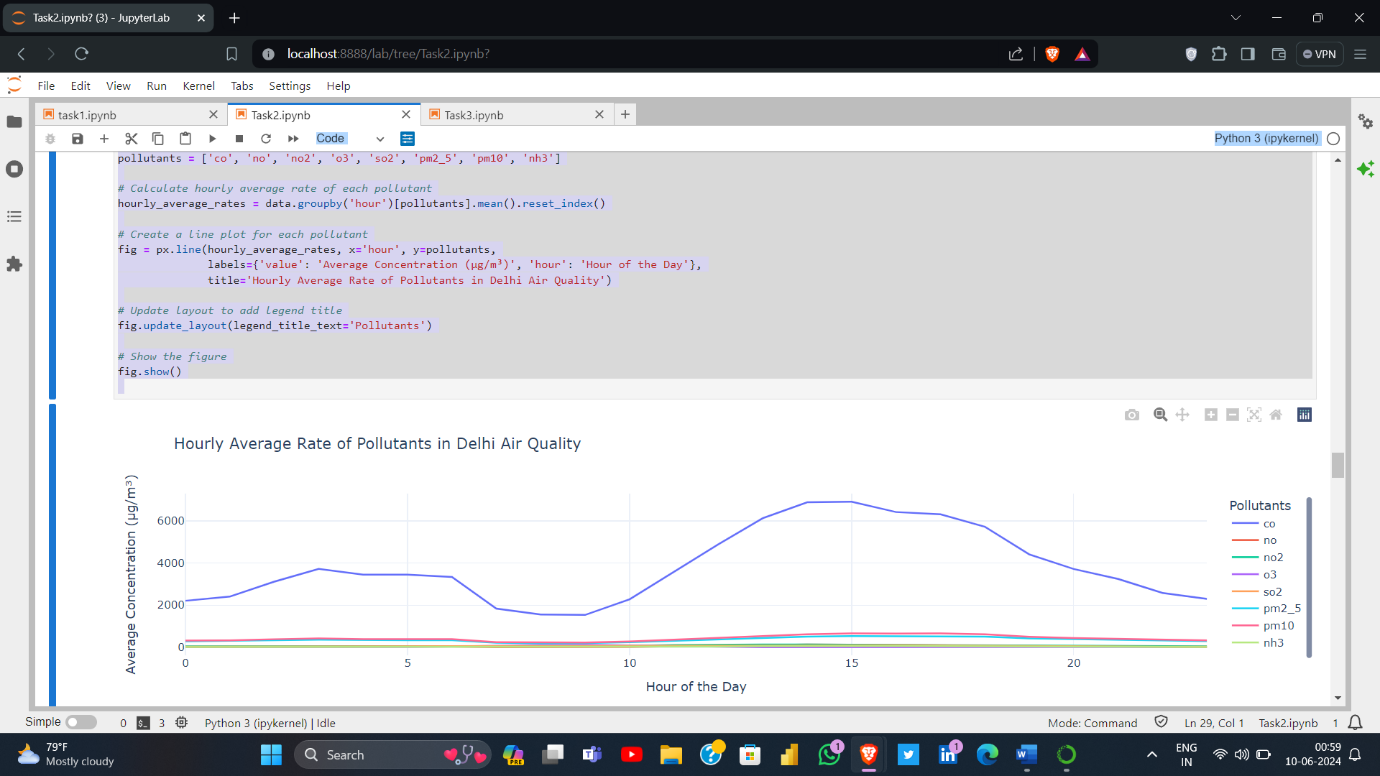
Heatmap of Pollutant Correlations

Visualize the correlation between different pollutants using a heatmap:

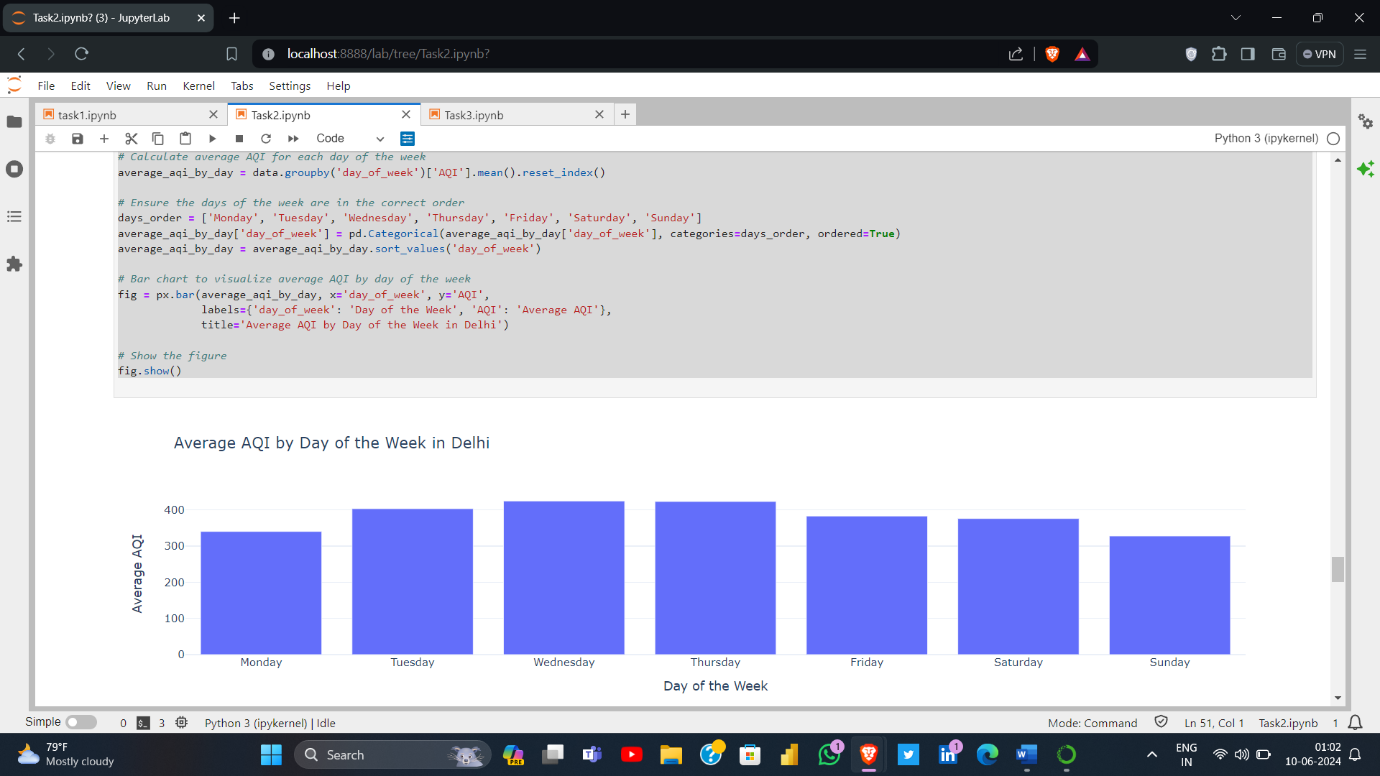


**7. Visualizing Hourly Average Rate of Pollutants**

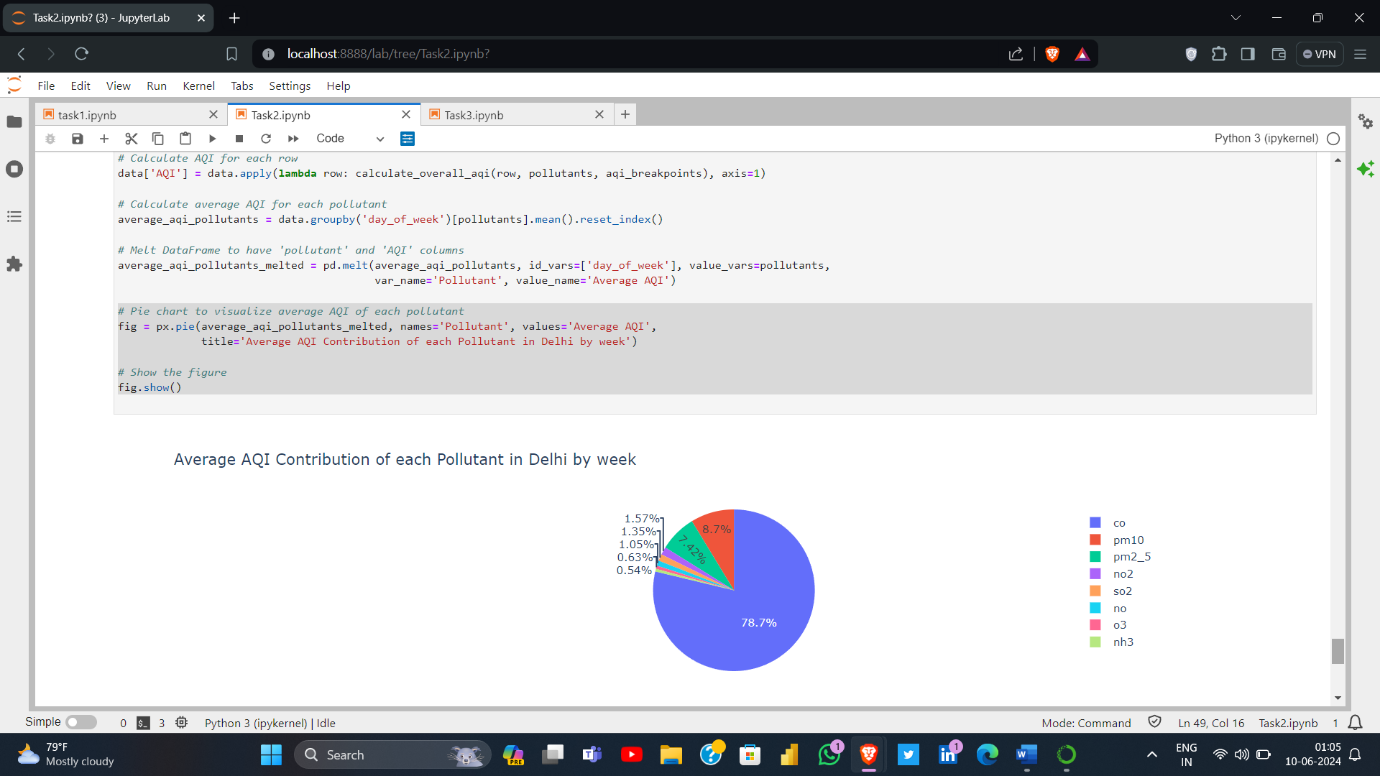
Line Graph of Hourly Average Pollutant Concentrations

 Calculate and visualize the hourly average rate of pollutants:

8. visualizing average AQI by day of the week using bar chart:

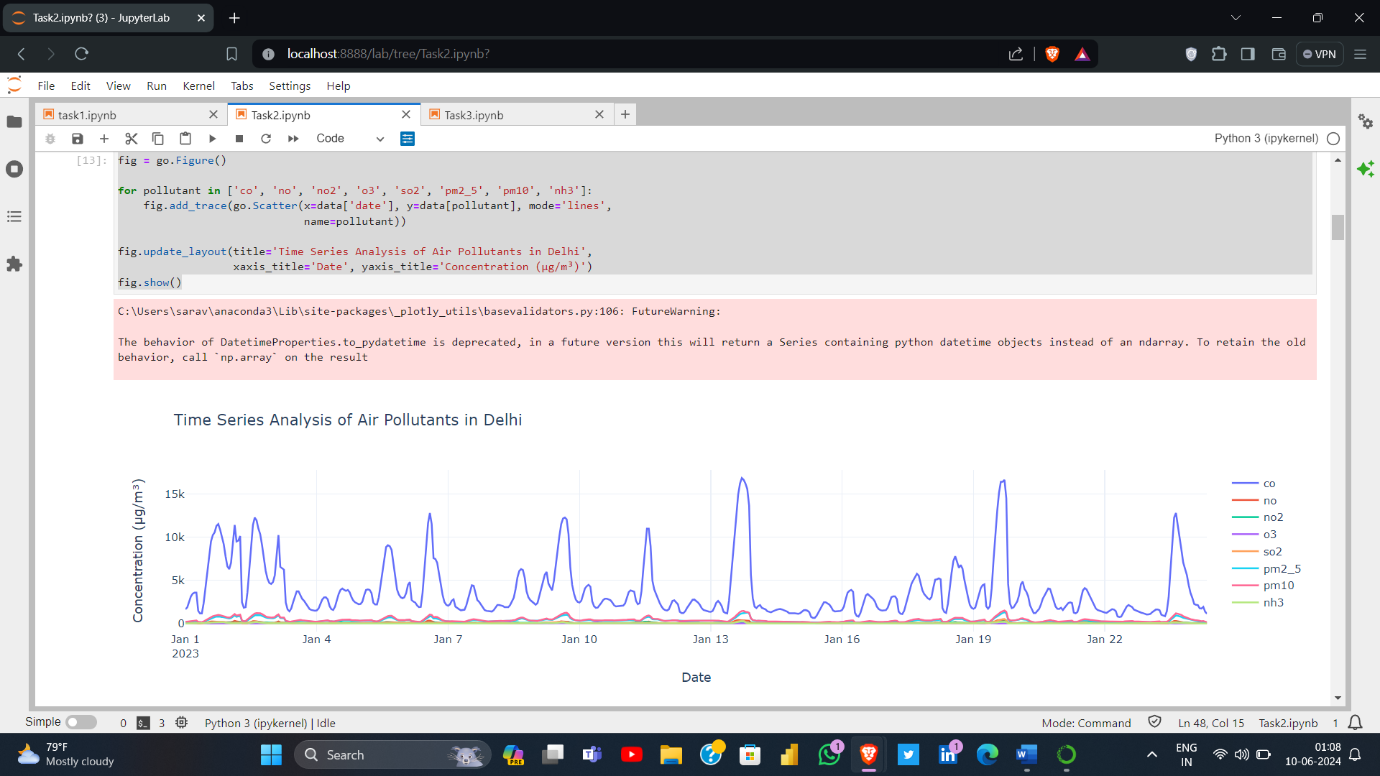


9.visualize average AQI Contribution of each pollutant using pie chart



10. Time Series Analysis of Air Pollutants in Delhi:

By using statistical factor the analysis is made to find series of air pollutants in delhi and to over certain period.



**Conclusion:**

Summary of Findings

This analysis provides a comprehensive analysis of air quality in Delhi through various visualizations and statistical analyses. Key findings from the analysis include:

Pollutant Concentrations Over Time:

The scatter plot analysis shows the variation in concentrations of major pollutants (CO, NO, NO2, O3, SO2, PM2.5, PM10, NH3) over time. These visualizations reveal significant fluctuations, with certain pollutants exhibiting higher variability.

AQI Calculation and Categorization:

The Air Quality Index (AQI) was calculated for each pollutant and categorized into different levels such as 'Good', 'Moderate', 'Unhealthy for Sensitive Groups', 'Unhealthy', 'Very Unhealthy', and 'Hazardous'. The bar plot illustrates that a substantial number of days fall into the 'Unhealthy' and 'Very Unhealthy' categories, indicating poor air quality on many days.

AQI Trends Over Time:

The line graph showing the AQI over time highlights periods of elevated AQI levels, particularly in winter months, which aligns with seasonal pollution trends often observed in Delhi.

Correlation Between Pollutants:

The heatmap of the correlation matrix indicates strong correlations between several pollutants. For instance, PM2.5 and PM10 show a high positive correlation, suggesting they often increase and decrease together, likely due to common sources.

Hourly Patterns of Pollutants:

The hourly average concentration plots reveal diurnal patterns in pollutant levels, with peaks often occurring during morning and evening rush hours due to increased vehicular emissions.

Hourly and Daily AQI Analysis:

The hourly average AQI analysis further supports the presence of daily cycles in air quality, with poorer air quality during traffic peaks. Additionally, the average AQI by day of the week shows variations, with weekends typically having different patterns compared to weekdays.

Weekend Air Quality:

The analysis of weekend air quality reveals that AQI tends to be lower on weekends, likely due to reduced industrial activity and vehicular traffic.

Implications and Recommendations

The findings of this project have several implications for policymakers, environmental agencies, and the public:

Traffic Management:

Implementing stricter traffic regulations during peak hours could help mitigate the spikes in pollution levels. Encouraging the use of public transportation and carpooling can also reduce vehicular emissions.

Industrial Regulations:

Enforcing stricter emission controls on industries, especially those contributing to PM2.5 and PM10, can help reduce the overall pollutant levels in the city.

Public Awareness:

Increasing public awareness about the health impacts of air pollution and promoting behavioral changes such as using air purifiers, wearing masks, and reducing outdoor activities during high AQI days.

Seasonal Interventions:

Implementing seasonal interventions, such as banning the use of fireworks during festivals and controlling crop residue burning in surrounding regions, can help reduce the seasonal spikes in pollution levels.

Real-time Monitoring:

Enhancing real-time air quality monitoring and making this data accessible to the public can empower citizens to make informed decisions regarding their outdoor activities.

|  |
| --- |
| Note:   1. <https://github.com/Saravana999/Shadow-Fox> 2. <http://localhost:8888/lab/tree/delhiaqi.ipynb> |