Chhatrapati Sambhaji Nagar Air Quality Dashboard (Aug 2022 – Oct 2025)

– by Anand Saundarya

https://www.linkedin.com/in/anand-saundarya-728a00226/

Project Report

🎯 Table of Contents

1. Executive Summary

This report details the end-to-end data analytics project for analyzing the Air Quality Index (AQI) of Chhatrapati SambhajiNagar (Aurangabad) from August 2022 to October 2025. The primary objective was to source, process, and analyze complex atmospheric data to create a user-friendly, interactive Power BI dashboard.

Raw hourly data for six major pollutants was sourced from the **Open-Meteo API**, which provides data from the **Copernicus Atmosphere Monitoring Service (CAMS)**. This model-based data was processed using a **Python** script in a Jupyter Notebook. The script handled data cleaning, feature engineering (e.g., time_of_the_day), and the critical unit conversion of pollutants (e.g., CO from $\mu g/m^3$ to mg/m^3).

The core of the analysis involved implementing the official **Central Pollution Control Board (CPCB)** piecewise formulas to convert raw pollutant concentrations into individual AQI sub-indices. The final, clean dataset was then loaded into Power BI for visualization.

The key finding from the dashboard is that the city's **Median AQI is 82.83**, classifying its air quality as **"Satisfactory"**. The analysis reveals that **PM2.5** and **Ozone (O_3)** are the primary drivers of pollution events and that air quality is consistently worst during **Morning** hours.

2. Project Overview

Air pollution is a critical environmental and public health issue across India. The Air Quality Index (AQI) is a vital tool for communicating air quality status to the public in a simple, color-coded manner.

This project focuses on **Chhatrapati SambhajiNagar (Aurangabad)**, a major city in Maharashtra. The objective was to move beyond single-day averages and create a granular, hourly analysis of air pollution trends. By sourcing data from a scientific model and building an interactive dashboard, this project aims to answer key questions:

- What is the overall air quality status of the city?
- Which pollutants are the primary contributors?
- How does air quality fluctuate by season, year, and, most importantly, time of day?

3. Data Source and Provenance

The data for this project was not sourced from a single ground-level sensor but from a sophisticated global atmospheric model.

- **API: Open-Meteo Air Quality API**. This is a free, public gateway for accessing high-quality meteorological and atmospheric data.
- **Core Provider: CAMS (Copernicus Atmosphere Monitoring Service)**, operated by the **ECMWF (European Centre for Medium-Range Weather Forecasts)**. This is a globally recognized, research-grade institution providing validated atmospheric composition data.
- **Location & Grid:** The data was queried for the coordinates **19.8776° N, 75.3423° E**. The CAMS model provides data on a 0.25° grid, meaning our data represents a regional average for a ~25x25 km cell covering the greater Chhatrapati SambhajiNagar area.
- **Granularity:** The data was requested at an **hourly** interval, providing a far more detailed view than the daily averages published by many public sources.
- **Credibility:** The CAMS data is a legitimate, peer-reviewed source used by governments and scientific agencies worldwide. This project uses this model output directly.

4. CPCB Reference Standards

All AQI calculations are based on the **National Ambient Air Quality Standards (NAAQS)** set by the **CPCB, India**. These standards define the "breakpoints" for converting a pollutant's concentration into a sub-index.

Pollutant	**Safe Limit (24-hr)**	**Unit**	**Sources & Notes**
PM2.5	60	µg/m³	Fine particulate matter. From combustion, traffic. Major health concern.
PM10	100	µg/m³	Coarse dust particles. From construction, roads.
NO_2	80	µg/m³	Nitrogen Dioxide. From vehicle exhaust, industrial combustion.

SO_2	80	µg/m^3	Sulphur Dioxide. From burning coal, diesel, and industrial processes.
CO	2000 (1-hr)	µg/m^3	Carbon Monoxide. From incomplete combustion, vehicle emissions.
O_3	100 (8-hr)	µg/m^3	Ozone (Ground-level). Forms from other pollutants under sunlight.

5. Data Extraction Method

The data can be extracted via a API call specified in the user prompt:

https://open-meteo.com/en/docs/air-quality-api?latitude=19.8776&longitude=75.3423&current=pm2_5,carbon_monoxide,nitrogen_dioxide,sulphur_dioxide,ozone,pm10&hourly=pm10,pm2_5,carbon_monoxide,carbon_dioxide,nitrogen_dioxide,sulphur_dioxide,ozone&forecast_days=1&past_days=92
or through the website too

The Jupyter Notebook (Air Quality.ipynb) loaded a pre-downloaded CSV (Cleaning.csv) derived from a similar, longer-term API query spanning from 2022 to 2025.

6. Dataset Structure

The final cleaned dataset (Air Quality Cleaned.csv) used for the Power BI dashboard contains the following columns:

- time: (datetime) Timestamp for the hourly reading.
- pm2_5 (µg/m^3): (float) Concentration of PM2.5.
- pm10 (µg/m^3): (float) Concentration of PM10.
- carbon_monoxide (µg/m^3): (float) **Converted unit (mg/m^3) for AQI calculation.**
- nitrogen_dioxide (µg/m^3): (float) Concentration of NO_2.
- ozone (µg/m^3): (float) Concentration of O_3.

- sulphur_dioxide (µg/m^3): (float) Concentration of SO_2.
- date: (date) Derived from time.
- hour: (int) Derived from time.
- year: (int) Derived from time.
- time_of_the_day: (string) Categorical bin: 'Morning', 'Afternoon', 'Evening', 'Night'.
- AQI_PM2_5: (float) Calculated CPCB sub-index for PM2.5.
- AQI_PM10: (float) Calculated CPCB sub-index for PM10.
- AQI_NO2: (float) Calculated CPCB sub-index for NO_2.
- AQI_SO2: (float) Calculated CPCB sub-index for SO_2.
- AQI_CO: (float) Calculated CPCB sub-index for CO.
- AQI_O3: (float) Calculated CPCB sub-index for O_3.
- AQI: (float) The final overall AQI (the max of all sub-indices).

7. Data Cleaning and Processing

The core logic was implemented in the Python notebook.

1. **Load Data:** The raw CSV was loaded. It was discovered that the first 3 rows were metadata, so the file was re-loaded with header=3.
2. **Handle Missing Data:** The carbon_dioxide (ppm) column, which is not a CPCB pollutant, had a large number of nulls. This column was dropped. Then, dropna() was used to remove any remaining rows with missing pollutant data, ensuring calculation accuracy.
3. **Feature Engineering:** The time column was converted to datetime objects. From this, date, hour, and year were extracted. A custom function binned the hour into time_of_the_day.
4. **Critical Unit Conversion:** The CPCB standard for CO is in mg/m^3, but the data was in µg/m^3. The entire carbon_monoxide column was divided by 1000 to correct this unit *before* calculating the AQI.
5. **AQI Calculation:** A series of functions (e.g., aqi_pm25, aqi_co) were created to implement the CPCB's piecewise linear formulas.
6. **Final AQI:** The final AQI column was created by taking the max() value of all six pollutant sub-indices for each hour.
7. **Export:** The cleaned, processed DataFrame was saved to Air Quality Cleaned.csv.

8. Data Analysis and Python Workflow

The following is a step-by-step documentation of the Air Quality.ipynb notebook.

8.1. Importing Libraries

The script begins by importing the necessary libraries: numpy for numerical operations and pandas for data manipulation.

```
import numpy as np
import pandas as pd
```

8.2. Initial Data Loading & Inspection

The script first attempts a simple load, which reveals the messy header.

```
df = pd.read_csv("Cleaning.csv")
```

This shows the first few rows are metadata. The script then re-loads the data correctly, specifying header=3 (the 4th row) as the true header.

```
df = pd.read_csv("Cleaning.csv", header=3)
```

8.3. Data Cleaning and Preprocessing

The df.info() command shows that carbon_dioxide (ppm) is mostly null. This column is irrelevant for CPCB AQI and is dropped. Any other rows with nulls are also dropped to ensure data integrity, and the index is reset.

```
df = df.drop(columns=['carbon_dioxide (ppm)']) #droping this cause co2 is not a pollutant
df.dropna(inplace=True)
df.reset_index(drop=True,inplace = True)
```

8.4. Feature Engineering

The time column is converted to a datetime object. This allows for the extraction of new time-based features.

```
df['time'] = pd.to_datetime(df['time'])
df['date'] = df['time'].dt.date
df['hour'] = df['time'].dt.hour
df['year'] = df['time'].dt.year
```

A custom function is defined to map hours to four distinct periods.

```
def time_of_the_day(hour):
   if 5<= hour <12:
      return 'Morning'
   elif 12 <= hour < 17:
      return 'Afternoon'
   elif 17<= hour < 19:
      return 'Evening'
   else:
      return 'Night'
df['time_of_the_day'] = df['hour'].apply(time_of_the_day)
```

8.5. AQI Calculation Logic

This section contains the core business logic of the project.

Step 1: Unit Conversion (Critical)
The Carbon Monoxide column is converted from µg/m³ to mg/m³ to match CPCB standards.

```
df['carbon_monoxide (µg/m³)'] = df['carbon_monoxide (µg/m³)'] / 1000
```

Step 2: Defining CPCB Sub-Index Functions
A function is defined for each of the six pollutants, implementing the CPCB's piecewise formulas.

```
def aqi_pm25(c):
   if c <= 30:
      return c * (50/30)
   elif c <= 60:
      return 50 + (c - 30) * (50/30)
   elif c <= 90:
      return 100 + (c - 60) * (100/30)
   elif c <= 120:
      return 200 + (c - 90) * (100/30)
   elif c <= 250:
      return 300 + (c - 120) * (100/130)
   else:
      return 400 + (c - 250) * (100/250)

def aqi_pm10(c):
   # ... similar logic for PM10 ...
   pass

def aqi_no2(c):
   # ... similar logic for NO2 ...
   pass

def aqi_so2(c):
   # ... similar logic for SO2 ...
   pass

def aqi_co(c):
   # Note: This function now works on the converted (mg/m³) value
   if c <= 1:
        return c * (50 / 1)
   elif c <= 2:
      return 50 + (c - 1) * (50 / 1)
   # ... similar logic for CO ...
   pass

def aqi_o3(c):
   # ... similar logic for O3 ...
   pass
```

Step 3: Applying Functions

These functions are applied to their respective columns to create new sub-index columns.

```
df['AQI_PM2_5'] = df['pm2_5 (µg/m³)'].apply(aqi_pm25)
df['AQI_PM10'] = df['pm10 (µg/m³)'].apply(aqi_pm10)
df['AQI_NO2'] = df['nitrogen_dioxide (µg/m³)'].apply(aqi_no2)
df['AQI_SO2'] = df['sulphur_dioxide (µg/m³)'].apply(aqi_so2)
df['AQI_CO'] = df['carbon_monoxide (µg/m³)'].apply(aqi_co)
df['AQI_O3'] = df['ozone (µg/m³)'].apply(aqi_o3)
```

8.6. Final AQI Determination

Per the CPCB definition, the overall AQI is the **maximum** of the individual sub-indices.

```
df['AQI'] = df[['AQI_PM2_5', 'AQI_PM10', 'AQI_NO2', 'AQI_SO2',
'AQI_CO','AQI_O3']].max(axis=1)
```

8.7. Exporting Cleaned Data

The final, processed DataFrame is saved as a new CSV, ready for Power BI.

```
df.to_csv("Air Quality Cleaned.csv",index=False)
```

9. Power BI Dashboard Design

The Air Quality Cleaned.csv file was imported into Power BI. The dashboard was designed for clarity and quick insights.

- **Layout:** The dashboard uses a clean, professional layout.
  - **Top Banner:** Title and the overall date range.
  - **Row 1 (KPIs):** A status card ("Satisfactory"), the main AQI gauge, and six cards for the median concentration of each pollutant.
  - **Row 2 (Charts):** A large area chart showing pollutant trends over time, and a line chart showing the strong daily pattern.
  - **Row 3 (Daily Trend):** A bar chart showing the median AQI for each day.
  - **Slicers (Right):** Filters for Year, Month, Day, and Time of the Day, allowing for granular analysis.
- **Key Visuals Explained:**
  - **Median of AQI Gauge:** This is the main KPI. It uses a DAX measure Median_AQI = MEDIAN('Air Quality Cleaned'[AQI]) to get **82.83**. A color-coded scale (Green, Yellow, Red) provides instant context.
  - **Pollutant Cards:** These (PM2.5: 35.67, PM10: 34.80, etc.) are cards displaying the *median* of each pollutant's concentration, not their AQI.
  - **Median AQI by Pollutant (Area Chart):** This chart plots the median AQI of each of the six pollutants over the selected time. This is crucial for identifying *which* pollutant is driving "bad air" days.
  - **Time of the day (Line Chart):** This plots the Median_AQI against the categorical time_of_the_day column, revealing the powerful daily pattern.
- **Color Coding:** All colors are based on CPCB standards. The gauge and "Satisfactory" card (green) align with the 51-100 AQI range. Pollutant cards (e.g., NO_2 and SO_2) are in red to draw attention, though this is a static color choice.

10. Analysis and Insights

The dashboard reveals several key insights:

1. **Overall AQI Category:** The overall median AQI is **82.83 ("Satisfactory")**. This suggests that, on average, the air quality is acceptable but could pose minor issues for sensitive groups.
2. **Pollutant Dominance:** The "Median AQI by Pollutant" chart clearly shows the purple line (**Ozone - O_3**) and the light blue line (**PM2.5**) are consistently the highest, meaning they are the primary drivers of the overall AQI.
3. **Time-of-Day Variation:** This is the strongest insight. The "Time of the day" chart shows a clear, repeating pattern:
   - **Morning:** Air quality is worst (Median AQI ~120, "Moderate"). This is likely due to morning traffic and atmospheric inversion trapping pollutants.
   - **Afternoon/Evening:** Air quality improves.
   - **Night:** Air quality is best (Median AQI ~65, "Satisfactory").
4. **Year-wise Comparison:** Using the slicers, a year-over-year comparison shows that median AQI has

been relatively stable, with seasonal peaks and troughs. The worst pollution consistently occurs in the winter months (Oct-Feb).

11. Technical Difficulties & Workarounds

- **Data Pull Limits:** The Open-Meteo API has limits on past_days. For a multi-year analysis, data had to be pulled in chunks (e.g., 92 days at a time) and appended.
- **Messy CSV Header:** The metadata in the first 3 rows of the CSV was a common data cleaning challenge, solved by using the header=3 parameter in pd.read_csv.
- **Critical Unit Conversion:** The most significant technical challenge was identifying that the CO data (µg/m³) did not match the CPCB standard (mg/m³). Failure to convert this (by dividing by 1000) would have resulted in a massively inflated and incorrect CO sub-index.
- **Median in Power BI:** A simple AVERAGE DAX measure would be skewed by high-pollution spikes. The MEDIAN function was intentionally used for the main gauge and charts to provide a more robust and representative measure of central tendency.

12. Comparison with Google AQI Data

A key finding is the discrepancy between this dashboard and public sources like Google, which the user prompt correctly identified.

- **Why the Difference?** Google typically reports a **single daily value**, which is often a 24-hour average or the value from a specific time. Our dashboard uses **hourly median data**.
- **Example (13 Feb 2025):**
  - **Google's Value:** 116 ("Moderate") - a single average.
  - **Our Dashboard's Data:** This project's hourly data tells the *full story* of that day. The AQI was **163 ("Poor")** in the morning (when people are commuting) and dropped to **90 ("Satisfactory")** by night.
  - **Our Dashboard's Value:** The *median* of those 24-hour readings was **93 ("Satisfactory")**. This explains the difference. Our value is a more accurate representation of the day's *typical* experience, while also allowing us (via the charts) to see the *extremes* that a simple average hides.

13. Limitations

- **Model vs. Sensor:** This is CAMS grid-based model data (≈ 25 km resolution). It is a regional mean, not a hyper-local reading from a specific street corner. It may not capture localized pollution from a specific traffic junction or industrial site.
- **No Meteorological Data:** This analysis only covers pollutants. It would be significantly enhanced by correlating AQI with weather data (humidity, temperature, wind speed), which can dramatically affect pollutant concentration and dispersion.
- **No Local Calibration:** The CAMS model data is not calibrated against local, on-the-ground CPCB sensors in Chhatrapati SambhajiNagar.

14. Future Enhancements

- **Weather Correlation:** Integrate weather data (also available from Open-Meteo) to analyze the relationship between temperature, humidity, wind, and AQI.

- **Predictive Modeling:** Use this rich historical dataset to build a time-series forecasting model (e.g., SARIMA, LSTM) to predict AQI 24-48 hours in advance.
- **Automation:** Create an automated data pipeline (e.g., using Python scripts or Power Automate) to fetch new data daily and refresh the Power BI dataset automatically.
- **CPCB Sensor Integration:** If available, integrate data from actual CPCB ground sensors in the city to compare and calibrate the model's accuracy.

15. Conclusion

This project successfully demonstrates a complete data analytics workflow, from sourcing raw, complex scientific data to building an insightful and actionable Power BI dashboard. The analysis provides a clear, data-driven understanding of air quality in Chhatrapati SambhajiNagar, concluding that the city's air is generally **"Satisfactory" (Median AQI 82.83)**.

More importantly, it identifies a clear pattern of **"Moderate" to "Poor" air quality during morning hours**, driven primarily by **PM2.5 and Ozone**. This key insight, along with the entire interactive dashboard, provides a powerful tool for public awareness and policy-making.

16. References

- **Open-Meteo Air Quality API:** https://open-meteo.com/en/docs/air-quality-api
- **Copernicus Atmosphere Monitoring Service (CAMS):** https://atmosphere.copernicus.eu/
- **Central Pollution Control Board (CPCB), India:** https://cpcb.nic.in/ (for NAAQS and AQI calculation methods)