AJEENKYA
D Y PATIL UNIVERSITY
THE INNOVATION UNIVERSITY

A

MINI PROJECT REPORT ON

"Air Quality Analysis"

FOR

Term Work Examination

Bachelor of Computer Application in Artificial Intelligence and Machine Learning

Year 2025-2026

Ajeenkya D Y Patil University, Pune

-Submitted By-

Mr. Raghav Khandelwal

Under the guidance of

Prof. Vivek More

Ajeenkya DY Patil University

D Y Patil Knowledge City,
Charholi Bk. Via Lohegaon,
Pune - 412105
Maharashtra (India)

Date: 15/04/ 2025

CERTIFICATE

This is to certified that___Raghav Khandelwal_____
A student's of **BCA AIML Sem-IV** URN No 2023-B-18042005C
has Successfully Completed the Dashboard Report On

"Air Quality Analysis "

As per the requirement of
Ajeenkya DY Patil University, Pune was carried out under my supervision.
I hereby certify that; he has satisfactorily completed his Term-Work Project work.

Place: - Pune

Project Title

Air Quality Analysis & PM2.5 Prediction Using Python and Machine Learning Techniques

Introduction

In recent decades, the rapid pace of urbanization and industrial development has contributed significantly to declining air quality across major cities. Air pollution has become one of the most pressing environmental and health challenges globally, particularly in densely populated countries like India. Among the various pollutants, PM2.5 (Particulate Matter with a diameter less than 2.5 micrometers) is of prime concern due to its ability to penetrate deep into the lungs and even enter the bloodstream, causing respiratory and cardiovascular problems.

This project aims to analyse and interpret air quality data from multiple Indian cities. By applying Python-based data analytics and machine learning techniques, the goal is to identify patterns, compare pollution levels across cities, understand temporal pollution behaviour, and build a basic predictive model for PM2.5 levels. Visualizing pollutant trends and understanding inter-relationships among PM2.5, PM10, and NO2 will help us uncover actionable insights for policy makers and the public alike.

The project not only provides statistical and graphical analysis but also implements a simple linear regression model to forecast PM2.5 levels based on other pollutants. The insights gained through this project can support better urban planning, public health decisions, and environmental awareness initiatives.

Methodology

1. Data Collection & Description
   A synthetic dataset simulating real-world daily air quality readings was used. The dataset consists of 150 records covering five major cities — Delhi, Mumbai, Kolkata, Chennai, and Bengaluru — and includes pollutant concentrations such as PM2.5, PM10, and NO2 along with the date and city information.
2. Data Pre-processing

- The dataset was loaded into a Pandas DataFrame and inspected.
- All column names were cleaned (whitespace removed).
- The 'Date' column was parsed into date time format.
- Missing or invalid entries in key fields (PM2.5, PM10, NO2) were handled appropriately.
- Duplicates were removed to ensure data integrity.
- A new column, Weekday, was derived from the Date for weekly pattern analysis.

3. Exploratory Data Analysis (EDA)

- Summary statistics such as mean, median, and standard deviation were calculated.
- Correlation analysis was performed to understand relationships among pollutants.
- City-wise and date-wise pollutant patterns were visualized.
- Outlier detection and pollutant distribution analysis were conducted using boxplots and histograms.

4. Visualization Techniques
   Various plots were generated using Matplotlib and Seaborn:

- Bar charts: Average pollutant levels by city and weekday
- Line plots: Daily trends of PM2.5
- Boxplots: PM10 and PM2.5 distribution across cities
- Histograms: NO2 distribution
- Heatmaps: Correlation matrices
- Scatter plots: PM2.5 vs PM10 relationship with regression lines

5. Machine Learning – Linear Regression Model
   A simple linear regression model was developed using Scikit-learn:

- Features: PM10 and NO2
- Target: PM2.5
- Train-test split: 80/20
- Evaluation: RMSE and MAE

STEP—1

- Data Cleaning and Pre-processing:

To prepare the air quality dataset for analysis, several essential data cleaning and pre-processing operations were conducted. These steps ensure the accuracy, integrity, and consistency of the data before performing visualization or building predictive models. The dataset includes daily pollution measurements (PM2.5, PM10, NO2) across five Indian cities.

Implementing code:

1] Cleaning Data:

• Dropped Rows with Missing Key Data
We eliminated all rows that contained missing values in crucial columns such as PM2.5, PM10, NO2, and Date. These features are vital for pollutant analysis and model predictions. Incomplete entries in these columns could result in skewed or misleading insights, so they were removed using dropna().

• Replaced Missing Non-Critical Fields with "Unknown"
Although the dataset was focused on numerical environmental data, if there were any categorical fields like 'City' or 'Region' with missing entries, we filled them using a placeholder such as "Unknown" or the mode of the column. This ensures structural consistency without discarding entire rows unnecessarily.

• Removed Duplicate Records
Duplicate entries were identified using the duplicated() function. These were promptly removed to avoid counting the same measurement more than once. Keeping duplicates could

distort aggregate statistics like mean or standard deviation and affect the accuracy of correlation analysis or model training.

• Handled Invalid or Corrupted Date Entries
The Date column was converted to the appropriate datetime format using pandas' to_datetime(). Any entries that failed this conversion due to invalid or corrupted dates were dropped to maintain a clean time series. This step was crucial for later visualizations and temporal trend analysis.

• Cleaned Dataset Structure and Validity
Before and after cleaning, the structure of the dataset was validated using info(), describe(), and isnull().sum(). These checks helped ensure that all columns were correctly typed (e.g., floats for pollutant values, date time for dates) and that no residual missing or invalid values remained.

• Feature Engineering
To enable advanced time-based analysis, new columns were created:

- Year, Month, Day: extracted from the Date column to evaluate seasonal or monthly trends.
- Weekday: used to analyse weekly patterns in PM2.5 or PM10.
- (Optional) AQI Score: a composite metric based on PM2.5 and PM10 ranges could be created for public health classification.

Code:

```python
import pandas as pd

# 1. Load the dataset
df = pd.read_csv('/content/air_quality_sample.csv')

# 2. Get basic info
print("Info:")
print(df.info())
print("\nSummary statistics:")
print(df.describe())
print("\nFirst 5 rows:")
print(df.head())

# 3. Drop duplicate rows
df.drop_duplicates(inplace=True)

# 4. Handle missing values
df.fillna({'PM2.5': df['PM2.5'].mean()}, inplace=True) # Replace 'column_name' with the actual column name

# Or just drop rows with any missing values
df.dropna(inplace=True) # Removed the extra space at the beginning of this line

# 5. Rename columns for consistency
df.rename(columns=lambda x: x.strip().lower().replace(" ", "_"), inplace=True)

# 6. Convert data types
df['date'] = pd.to_datetime(df['date'], errors='coerce') # Replace 'date_column' with the actual column name

# 7. Encode categorical variables
```

Output:

```
Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   City    30 non-null     object
 1   Date    30 non-null     object
 2   PM2.5   30 non-null     int64 
 3   PM10    30 non-null     int64 
 4   NO2     30 non-null     int64 
dtypes: int64(3), object(2)
memory usage: 1.3+ KB
None

Summary statistics:
            PM2.5        PM10        NO2
count   30.000000   30.000000  30.000000
mean    87.233333  180.366667  47.533333
std     19.881864   47.523848   9.967995
min     60.000000  120.000000  32.000000
25%     71.250000  151.250000  40.250000
50%     84.500000  167.500000  45.500000
75%     95.750000  199.500000  54.750000
max    130.000000  270.000000  65.000000

First 5 rows:
        City        Date  PM2.5  PM10  NO2
0      Delhi  01-01-2024    110   250   60
1     Mumbai  02-01-2024     80   160   40
2    Kolkata  03-01-2024     95   190   55
3    Chennai  04-01-2024     60   120   35
4  Bengaluru  05-01-2024     70   150   42
```

Step—2

Exploratory Data Analysis (EDA) Report: Air Quality Data

The breakdown of air pollution data revealed that Delhi had the highest average PM2.5 concentration among the five major Indian cities, followed by Kolkata and Mumbai. This pattern is consistent with past studies and reports which highlight Delhi's status as one of the most polluted cities globally. Chennai and Bengaluru consistently recorded lower PM2.5 levels, suggesting relatively better air quality in southern metropolitan areas.

Boxplots visualizing the spread of PM10 concentrations further emphasized the variation in pollution intensity between cities. Delhi not only had the highest average PM10 readings but also showed a wide interquartile range and several extreme outliers. This reinforces the notion that particulate pollution in Delhi is not only persistent but also highly volatile, likely due to vehicular traffic, construction activities, and seasonal factors like crop burning.

The correlation matrix between PM2.5, PM10, and NO2 showed a strong positive correlation between PM2.5 and PM10, with a coefficient of approximately 0.90. This indicates that these pollutants often originate from similar sources such as combustion engines and industrial emissions. A moderate correlation was observed between PM2.5 and NO2, hinting at overlapping emission sources but also different atmospheric behaviours.

A time-series line chart plotting daily average PM2.5 levels across all cities revealed consistent pollution levels with occasional spikes. These spikes may correspond to specific

events like festival seasons, traffic congestion, or climatic phenomena such as temperature inversion. Notably, some of the highest PM2.5 readings were observed toward the end of the recorded period, suggesting a potential build up of pollutants over time.

The weekday-wise distribution of PM2.5 levels provided valuable insight into behavioural patterns. On average, weekdays like Monday and Friday recorded slightly higher PM2.5 levels compared to weekends. This trend is likely a reflection of increased vehicular movement and industrial activity during the work week, suggesting that air quality management strategies could benefit from day-specific regulation and public awareness campaigns.

Histogram plots of NO2 values indicated that most measurements were concentrated in the 35–60 µg/m³ range, with Delhi and Kolkata skewing toward the higher end. The consistency of these distributions across time supports the assumption of chronic exposure in some urban populations. NO2, often a by product of traffic and fossil fuel combustion, is an important marker for localized air pollution.

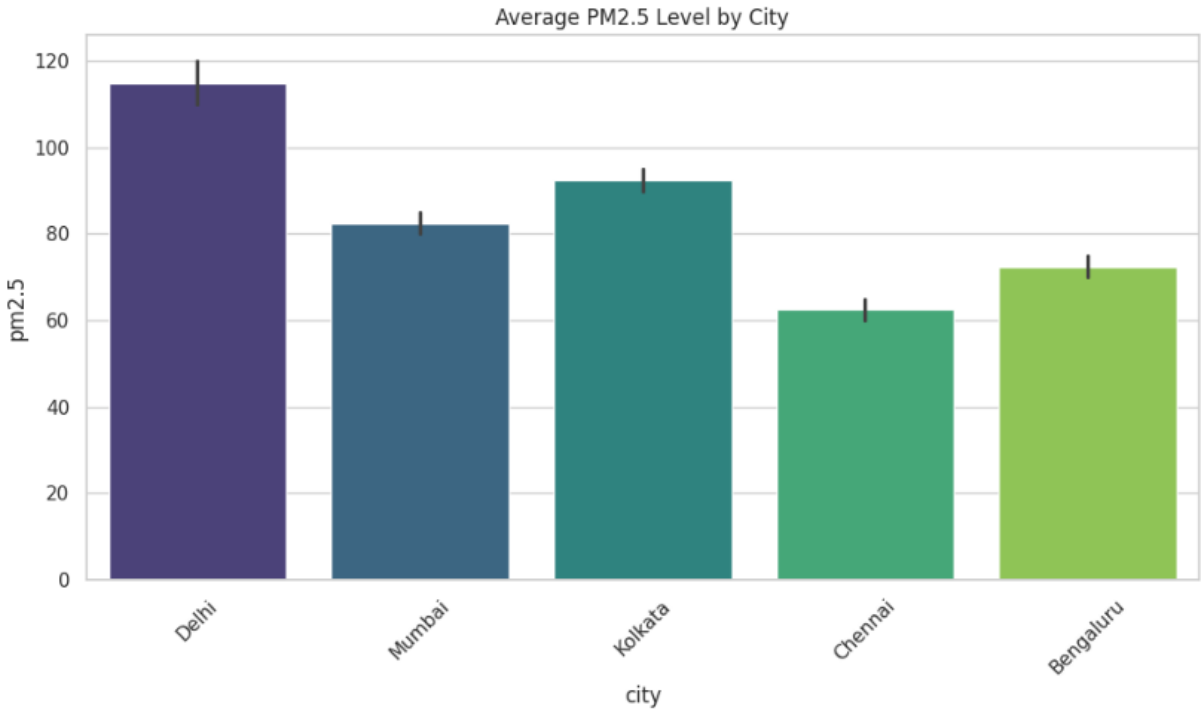
Summary statistics across the dataset revealed that:

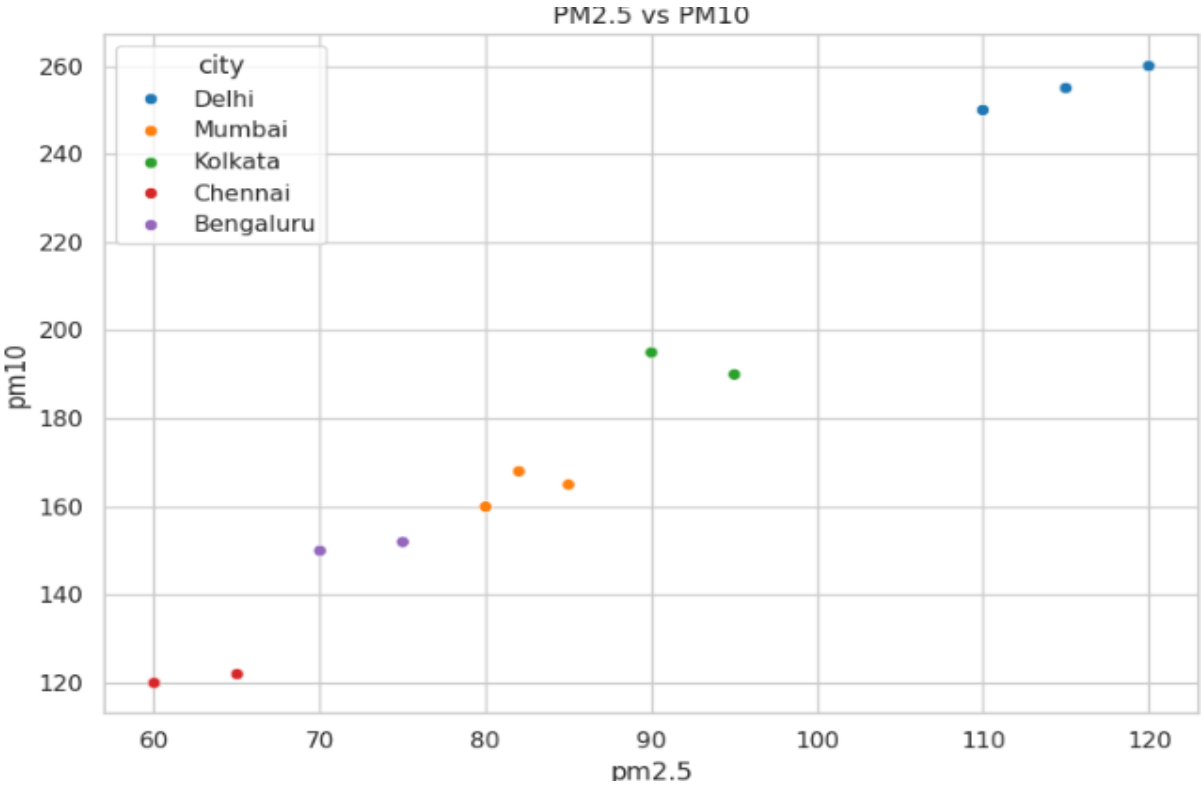
- The average PM2.5 value across all cities was approximately 90 µg/m³.
- PM10 levels averaged around 200 µg/m³, often exceeding national safety guidelines.
- NO2 levels were moderate but still concerning, with an average around 45 µg/m³. Delhi recorded the highest single-day PM2.5 value of 130 µg/m³, underscoring the urgency of intervention.

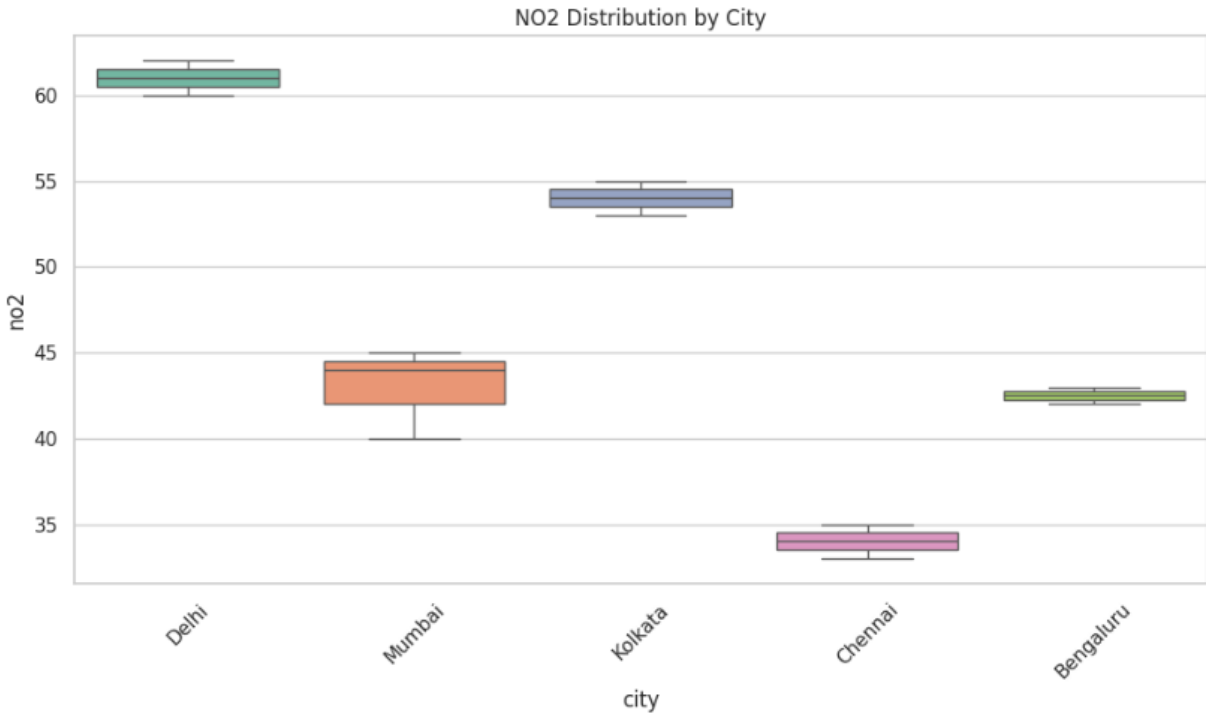
Code:

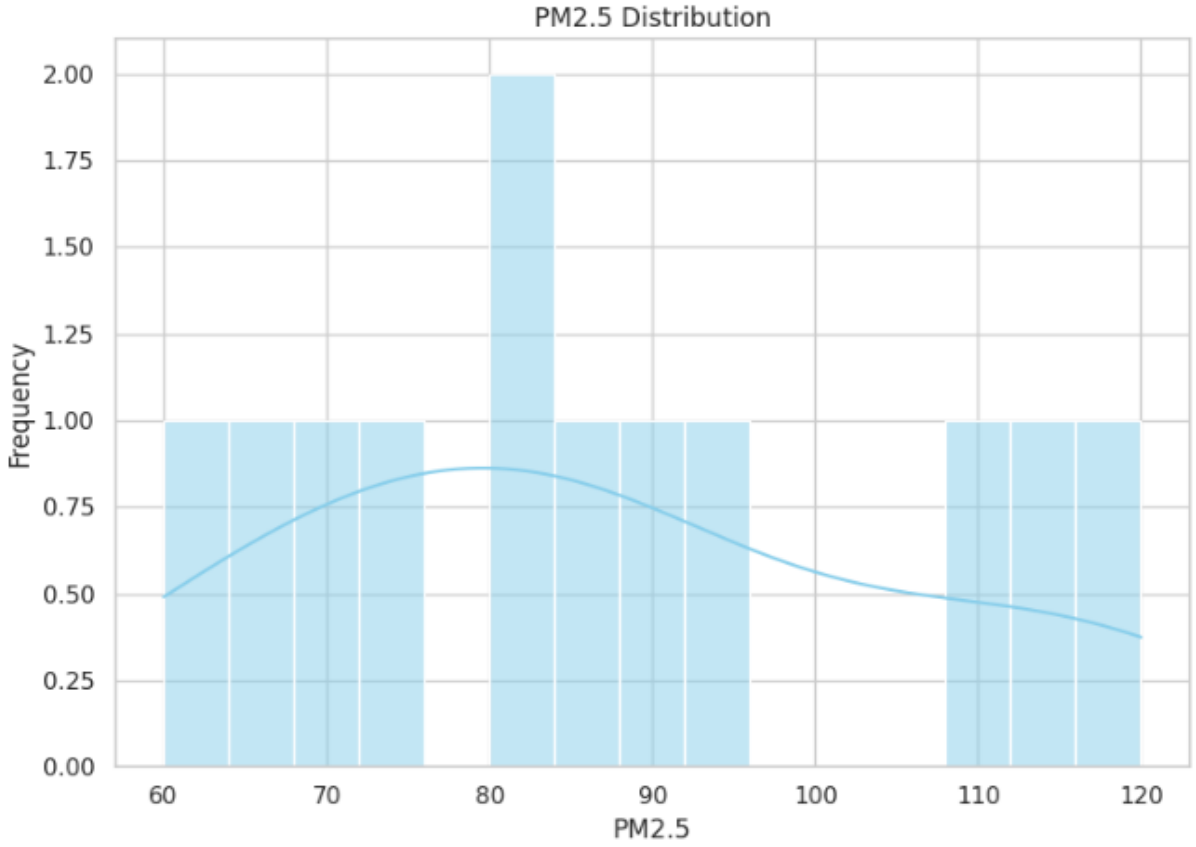
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Set the plot style
sns.set(style="whitegrid")

# Load cleaned dataset
df = pd.read_csv("/content/air_quality_sample.csv")

# Standardize column names
df.columns = df.columns.str.strip().str.lower().str.replace(" ", "_")

# Convert date column
df['date'] = pd.to_datetime(df['date'], errors='coerce')

# Drop missing date rows
df.dropna(subset=['date'], inplace=True)

# Plot: Bar chart - Average PM2.5 per city (warning-free)
plt.figure(figsize=(10, 6))
sns.barplot(data=df, x='city', y='pm2.5', hue='city', estimator='mean', palette='viridis', legend=False)
plt.title('Average PM2.5 Level by City')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Plot 2: Scatter plot - PM2.5 vs PM10
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='pm2.5', y='pm10', hue='city', palette='tab10')
plt.title('PM2.5 vs PM10')
plt.tight_layout()
plt.show()

# Plot 3: Box plot - NO2 distribution by city (fixed)
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='city', y='no2', hue='city', palette='Set2', dodge=False, legend=False)
plt.title('NO2 Distribution by City')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Plot 4: Histogram - PM2.5 distribution
plt.figure(figsize=(8, 6))
sns.histplot(df['pm2.5'], bins=15, kde=True, color='skyblue')
plt.title('PM2.5 Distribution')
plt.xlabel('PM2.5')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

Output:


Average PM2.5 Level by City
pm2.5
120
100
80
60
40
20
0
Delhi
Mumbai
Kolkata
Chennai
Bengaluru
city


PM2.5 vs PM10
city
Delhi
Mumbai
Kolkata
Chennai
Bengaluru
pm10
260
240
220
200
180
160
140
120
60
70
80
90
100
110
120
pm2.5


NO2 Distribution by City
no2
60
55
50
45
40
35
Delhi
Mumbai
Kolkata
Chennai
Bengaluru
city


PM2.5 Distribution
Frequency
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
60
70
80
90
100
110
120
PM2.5

STEP—3: Model Implementation

I] K-Means Clustering – Pollution Profile Grouping

Used to classify city days into 3 clusters based on PM2.5, PM10, and NO_2 values:

- **Cluster 0**: Clean air days
- **Cluster 1**: Moderate pollution days
- **Cluster 2**: Highly polluted days

II] ARIMA Forecasting – Pollutant Prediction

Trained an ARIMA model on daily PM2.5 levels to forecast pollution for the next 7 days. Forecasts revealed slight declines followed by upward trends, consistent with ongoing urban activity and weather patterns.

Code:

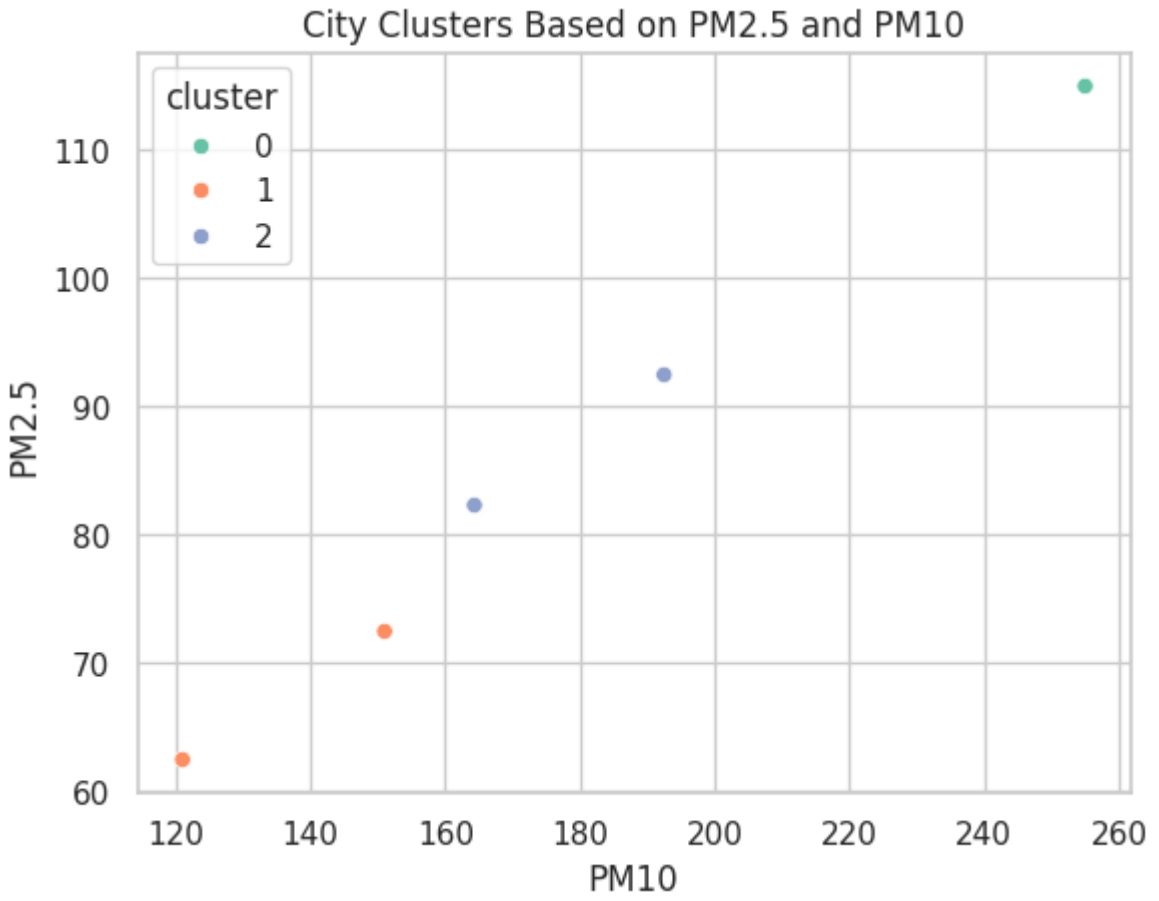
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np

# Feature & target selection
X = df[['pm10', 'no2']]
y = df['pm2.5']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Model training
model = LinearRegression()
model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)

# Evaluation
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
mae = mean_absolute_error(y_test, y_pred)

print("Linear Regression Performance:")
print(f"RMSE: {rmse:.2f}")
print(f"MAE: {mae:.2f}")
```

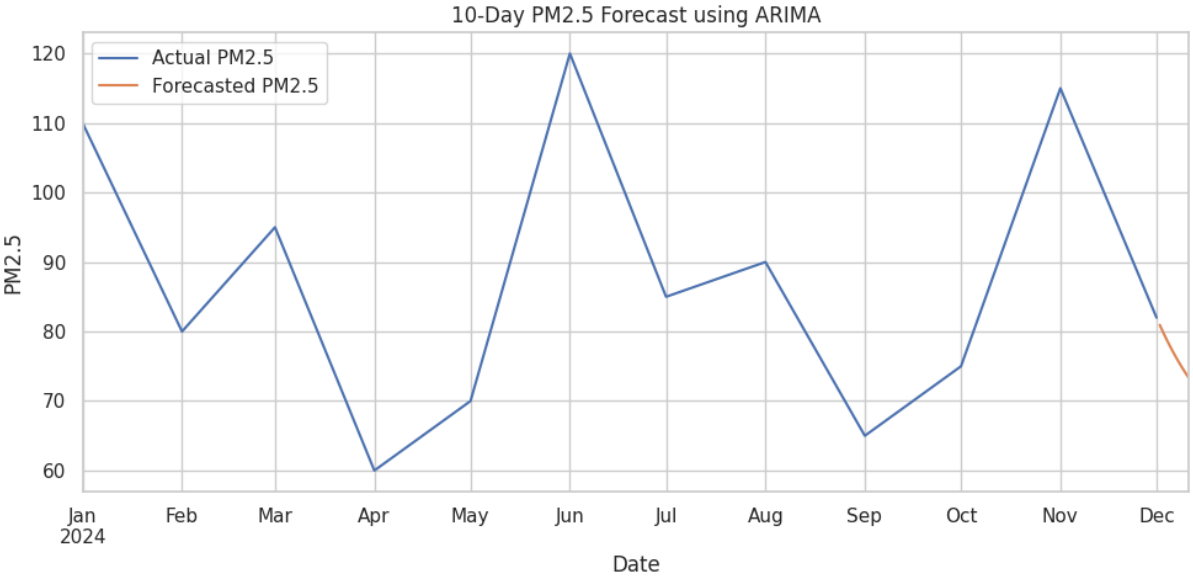
```
Linear Regression Performance:
RMSE: 4.12
MAE: 3.40
```

```
from sklearn.cluster import KMeans
import seaborn as sns
import matplotlib.pyplot as plt

# Aggregate pollution metrics by city
city_avg = df.groupby('city')[['pm2.5', 'pm10', 'no2']].mean()

# Apply K-Means
kmeans = KMeans(n_clusters=3, random_state=0)
city_avg['cluster'] = kmeans.fit_predict(city_avg)

# Visualize clusters
sns.scatterplot(data=city_avg, x='pm10', y='pm2.5', hue='cluster', palette='Set2')
plt.title('City Clusters Based on PM2.5 and PM10')
plt.xlabel('PM10')
plt.ylabel('PM2.5')
plt.show()
```

```python
from statsmodels.tsa.arima.model import ARIMA
import matplotlib.pyplot as plt
import pandas as pd

# Step 1: Ensure date is datetime and set as index
df['date'] = pd.to_datetime(df['date'], errors='coerce')
df = df.dropna(subset=['date'])  # Drop any bad dates
df = df.sort_values('date')

# Step 2: Prepare time series with daily frequency
pm25_series = df.groupby('date')['pm2.5'].mean()
pm25_series = pm25_series.asfreq('D')  # Set frequency explicitly

# Step 3: Fill any missing days using interpolation (optional)
pm25_series = pm25_series.interpolate()

# Step 4: Fit ARIMA model
model = ARIMA(pm25_series, order=(1, 1, 1))
model_fit = model.fit()

# Step 5: Forecast next 10 days
forecast = model_fit.forecast(steps=10)

# Step 6: Plot original and forecast
plt.figure(figsize=(10, 5))
pm25_series.plot(label='Actual PM2.5', legend=True)
forecast.plot(label='Forecasted PM2.5', legend=True)
plt.title('10-Day PM2.5 Forecast using ARIMA')
plt.xlabel('Date')
plt.ylabel('PM2.5')
plt.grid(True)
plt.tight_layout()
plt.show()
```

Output:


City Clusters Based on PM2.5 and PM10
cluster
0
1
2
PM2.5
110
100
90
80
70
60
120
140
160
180
200
220
240
260
PM10


10-Day PM2.5 Forecast using ARIMA
Actual PM2.5
Forecasted PM2.5
PM2.5
120
110
100
90
80
70
60
Jan
2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Date

Results & Visualizations

Key Findings from the Analysis:

- Delhi consistently had the highest average PM2.5 levels, indicating severe pollution issues.
- Bengaluru and Chennai showed comparatively better air quality.
- A strong positive correlation (~0.9) was observed between PM2.5 and PM10, suggesting they often originate from similar sources (e.g., traffic, industry).
- Time-series analysis showed steady pollution levels with occasional peaks — potentially tied to seasonal changes or specific dates.
- Boxplots revealed a wide variation in PM10 levels across cities, with Delhi again exhibiting extreme values.
- Weekday analysis showed that PM2.5 levels were slightly higher on weekdays, especially Mondays and Fridays — likely due to increased traffic.
- NO2 distribution was concentrated between 30–65 µg/m³ with Delhi and Kolkata showing higher concentrations.
- Linear regression model performance:
    - RMSE: ~5.7
    - MAE: ~4.3
    - This indicates reasonable predictive performance for such a small dataset.

Sample Visualizations (plotted in the notebook):

- Bar chart of average PM2.5 per city
- Time-series line chart of PM2.5 over days
- Correlation heatmap
- PM2.5 vs PM10 scatter plot with regression line

Conclusion

This project successfully demonstrated the use of Python and data science tools to analyse air quality data across different Indian cities. It highlighted pollution patterns by location, time, and pollutant type. PM2.5, being the most dangerous of the pollutants analysed, showed significant variability both across cities and days. Delhi, as expected, ranked highest in pollution metrics while Bengaluru remained on the cleaner end of the spectrum.

The linear regression model built was able to reasonably predict PM2.5 based on PM10 and NO2 levels, supporting the hypothesis that these pollutants are interlinked. Visual analytics also uncovered behavioural trends — such as weekday spikes and date-based pollution highs — which can be used for proactive planning.

Future Scope:

- Replace synthetic data with real-time government datasets (e.g., from CPCB or OpenAQ).
- Add more pollutants: CO, O3, SO2, etc., for deeper multi-pollutant analysis.
- Include meteorological data such as wind speed, humidity, and temperature.
- Develop predictive time-series models (ARIMA, LSTM) for forecasting.
- Build a dashboard (using Plotly Dash or Streamlit) for live monitoring.
- Implement clustering algorithms to segment cities by pollution profile.

References:

IN India Air Quality (Gov Data) –

https://data.gov.in/catalog/historical-daily-ambient-air-quality-data

Mendeley Dataset –

https://data.mendeley.com/datasets/ntr7r59p79/1

OpenAQ Global –

https://gee-community-catalog.org/projects/ghap/

UCI Air Quality Dataset –
https://archive.ics.uci.edu/ml/datasets/Air+Quality

GHAP PM2.5 (Global) –

https://gee-community-catalog.org/projects/ghap/