Group Assignment: Spatial Analytics

Rahul Mukundhan (IMT2022518),Shanmukh Praneeth(IMT2022542)

May 4, 2025

Abstract

This is an assignment aimed at understanding Spatial Data analytics. The group assignment (GA) focuses on spatial data analytics exclusively to be implemented by groups of 2 on this allocated dataset. The GA is on exploratory spatial analysis, which has two parts: statistical analysis and machine learning(ML). The chosen ML method is based on the availability of appropriate label data in the selected data set.Statistical analysis includes statistical tests and exploratory data analysis.

1 Dataset and Codes

For source code, visit our GitHub repository. and the dataset is in Kaggle Dataset

2 Introduction

This project presents a spatial analysis of air quality data, aimed at discovering spatial dependencies and pollutant distribution patterns. Using a combination of statistical and geospatial machine learning methods, we explore pollution trends, spatial autocorrelation, and regional variations across different countries.

3 Dataset Descriptions

The dataset used contains air quality measurements across several countries, including pollutant values such as PM2.5. levels which have been taken for every month in 2023 and averaged. Basic pre-processing included:

- Removing null/missing values
- Standardizing location names
- Deriving geospatial points using geopy

4 Methods and Rationale

4.1 Preprocessing

Before heading to any Spatial Process we pre-process the data with methods such as:

- Country names were normalized
- Geocoded location information was generated using Nominatim

4.2 Spatial Techniques Used

- Moran's I: To evaluate global spatial autocorrelation of pollution levels.
- Local Moran (LISA): To identify regional clusters and outliers.
- Geary's C & G_Local: Supplementary spatial statistics to confirm clustering.
- IDW & KDE: Used for spatial interpolation and density estimation.
- GWR (Geographically Weighted Regression): To model spatially varying relationships.

4.3 Rationale for Method Selection

The methods were selected due to their suitability for analyzing continuous geospatial data. Moran's I and LISA are widely used for detecting spatial autocorrelation, while IDW and GWR help in interpolation and local regression modeling respectively.

5 Discovering Spatial Patterns

- Spatial clustering of high pollution values was evident in specific countries.
- Local indicators (LISA) revealed pollution hotspots and spatial outliers.
- IDW interpolation highlighted regions with missing data but likely high pollution.
- GWR suggested that relationships between pollutants and geographic location vary regionally.

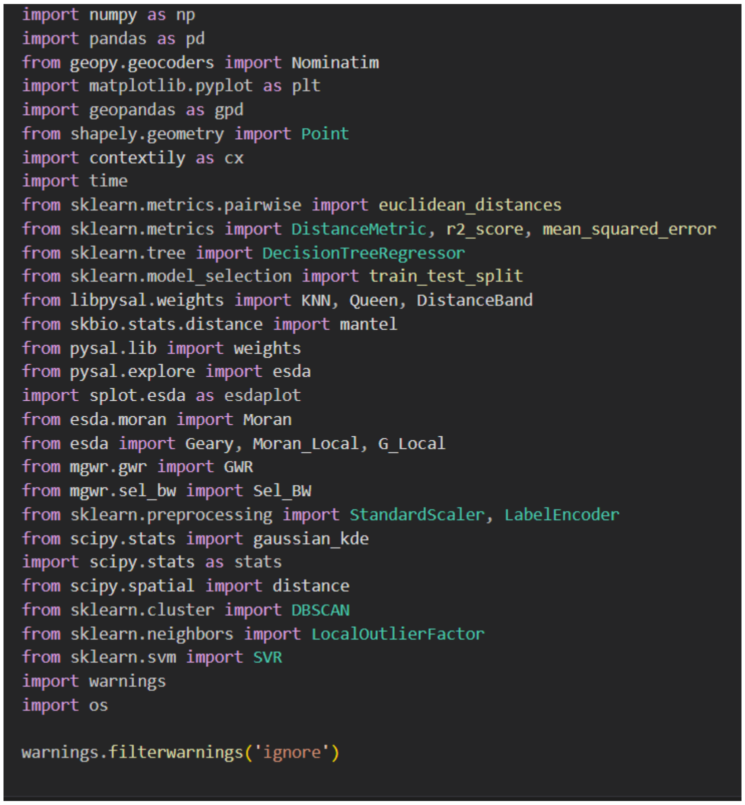
6 Implementation

6.1 Library and Environment setup

Some of the libraries used are as follows:

- Environment: Jupyter Notebook
- Data handling and plotting: pandas, numpy, matplotlib
- Geocoding - Geopy
- Geospatial data and basemaps - geopandas, shapely, contextily
- Spatial statistics - libpysal, esda, splot
- GWR and IDW - mgwr,pykrige

A full listing of all the libraries can be found below:



```
import numpy as np
import pandas as pd
from geopy.geocoders import Nominatim
import matplotlib.pyplot as plt
import geopandas as gpd
from shapely.geometry import Point
import contextily as cx
import time
from sklearn.metrics.pairwise import euclidean_distances
from sklearn.metrics import DistanceMetric, r2_score, mean_squared_error
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from libpysal.weights import KNN, Queen, DistanceBand
from skbio.stats.distance import mantel
from pysal.lib import weights
from pysal.explore import esda
import splot.esda as esdaplot
from esda.moran import Moran
from esda import Geary, Moran_Local, G_Local
from mgwr.gwr import GWR
from mgwr.sel_bw import Sel_BW
from sklearn.preprocessing import StandardScaler, LabelEncoder
from scipy.stats import gaussian_kde
import scipy.stats as stats
from scipy.spatial import distance
from sklearn.cluster import DBSCAN
from sklearn.neighbors import LocalOutlierFactor
from sklearn.svm import SVR
import warnings
import os

warnings.filterwarnings('ignore')
```

Figure 1: List of libraries

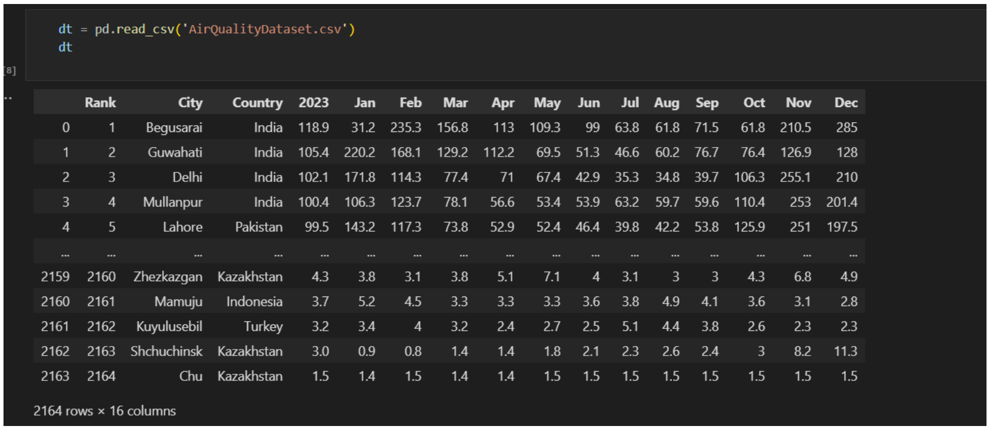
6.2 Challenges faced

- Geocoding Rate Limits: Solved using request throttling and caching.
- Handling Sparse Data: Applied KDE and Kriging for estimation
- Lack of a sharp image of the Heat Map generated for KDE, due to bandwidth for smoothening.

7 Flow of the project

7.1 Dataset Loading and Preliminary Inspection

The dataset was loaded using `pandas`. Initial exploration included examining the data structure, checking for missing values, and inspecting basic statistics. A snapshot of the dataset (`df.head()`) was generated for quick reference.



```
dt = pd.read_csv('AirQualityDataset.csv')
dt
```

[8]

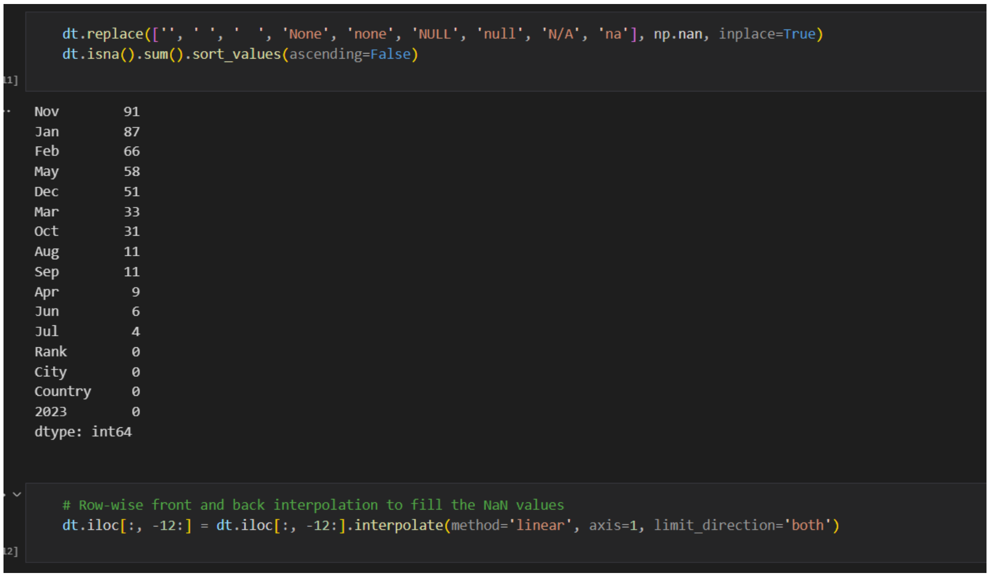
	Rank	City	Country	2023	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
0	1	Begusarai	India	118.9	31.2	235.3	156.8	113	109.3	99	63.8	61.8	71.5	61.8	210.5	285
1	2	Guwahati	India	105.4	220.2	168.1	129.2	112.2	69.5	51.3	46.6	60.2	76.7	76.4	126.9	128
2	3	Delhi	India	102.1	171.8	114.3	77.4	71	67.4	42.9	35.3	34.8	39.7	106.3	255.1	210
3	4	Mullanpur	India	100.4	106.3	123.7	78.1	56.6	53.4	53.9	63.2	59.7	59.6	110.4	253	201.4
4	5	Lahore	Pakistan	99.5	143.2	117.3	73.8	52.9	52.4	46.4	39.8	42.2	53.8	125.9	251	197.5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2159	2160	Zhezkazgan	Kazakhstan	4.3	3.8	3.1	3.8	5.1	7.1	4	3.1	3	3	4.3	6.8	4.9
2160	2161	Mamuju	Indonesia	3.7	5.2	4.5	3.3	3.3	3.3	3.6	3.8	4.9	4.1	3.6	3.1	2.8
2161	2162	Kuyulusebil	Turkey	3.2	3.4	4	3.2	2.4	2.7	2.5	5.1	4.4	3.8	2.6	2.3	2.3
2162	2163	Shchuchinsk	Kazakhstan	3.0	0.9	0.8	1.4	1.4	1.8	2.1	2.3	2.6	2.4	3	8.2	11.3
2163	2164	Chu	Kazakhstan	1.5	1.4	1.5	1.4	1.4	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5

2164 rows × 16 columns

Figure 2: Top few rows of the AirQualityDataset

7.2 Data Preprocessing

Data cleaning involved handling missing values through imputation and removing irrelevant columns. Data types were converted appropriately. Numerical features were standardized where required. A correlation heatmap was used to visualize inter-variable relationships. In the output image, some countries are not shown as they are absent from the dataset. We do linear front and back linear interpolation to remove null values.



```
dt.replace(['', ' ', '  ', 'None', 'none', 'NULL', 'null', 'N/A', 'na'], np.nan, inplace=True)
dt.isna().sum().sort_values(ascending=False)
```

```
Nov        91
Jan        87
Feb        66
May        58
Dec        51
Mar        33
Oct        31
Aug        11
Sep        11
Apr         9
Jun         6
Jul         4
Rank        0
City        0
Country     0
2023        0
dtype: int64
```

```
# Row-wise front and back interpolation to fill the NaN values
dt.iloc[:, -12:] = dt.iloc[:, -12:].interpolate(method='linear', axis=1, limit_direction='both')
```

Figure 3: Preprocessing of the datset

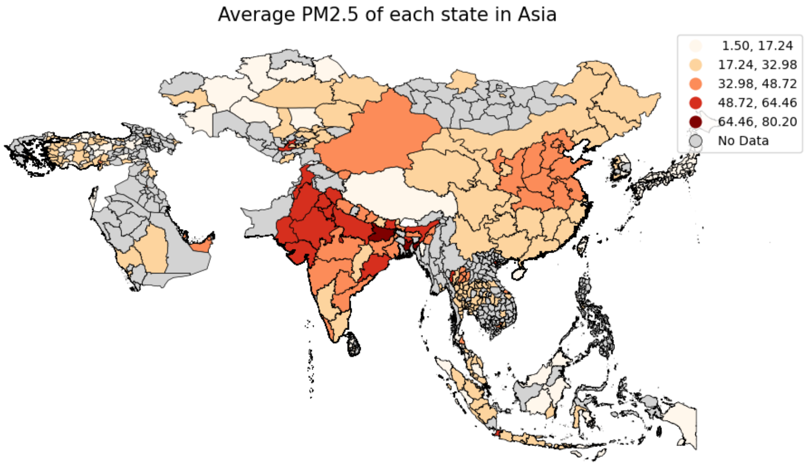


Figure 4: Average PM2.5 of each state in Asia

7.3 Spatial Locality of Point Data

Mantel's test was used to assess the relationship between spatial distance and PM2.5 levels. The results showed that some distant locations had similar PM2.5 values, while some nearby locations had significant differences in PM2.5.

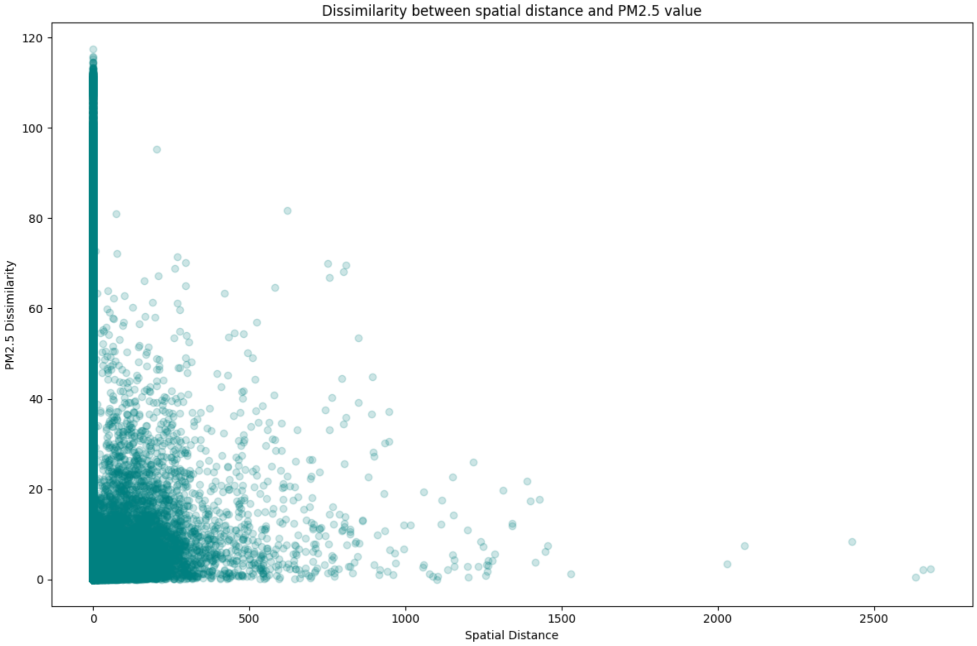
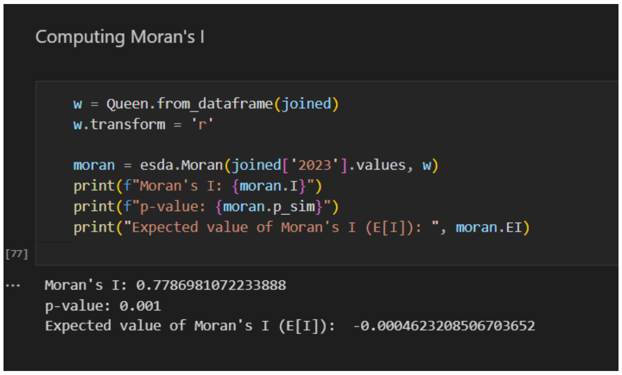


Figure 5: Dissimilarity between spatial distance and PM2.5 value

7.4 Spatial Autocorrelation Analysis

Spatial clustering was assessed using Moran's I and Geary's C statistics.These values satisfies positive SAC analysis indicating clustering of data samples. High spatial autocorrelation was observed for pollutants, indicating localized concentration pockets.We even show the outliers through the scatterplot in the second and fourth quadrants.



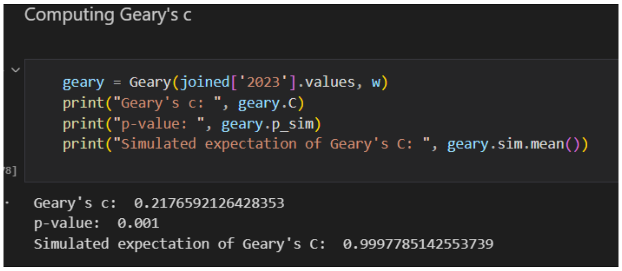
Computing Moran's I

```
w = Queen.from_dataframe(joined)
w.transform = 'r'

moran = esda.Moran(joined['2023'].values, w)
print(f"Moran's I: {moran.I}")
print(f"p-value: {moran.p_sim}")
print("Expected value of Moran's I (E[I]): ", moran.EI)
```

```
Moran's I: 0.7786981072233888
p-value: 0.001
Expected value of Moran's I (E[I]):  -0.0004623208506703652
```

Figure 6: Moran's I Result



Computing Geary's c

```
geary = Geary(joined['2023'].values, w)
print("Geary's c: ", geary.C)
print("p-value: ", geary.p_sim)
print("Simulated expectation of Geary's C: ", geary.sim.mean())
```

```
Geary's c:  0.2176592126428353
p-value:  0.001
Simulated expectation of Geary's C:  0.9997785142553739
```

Figure 7: Geary's C Result

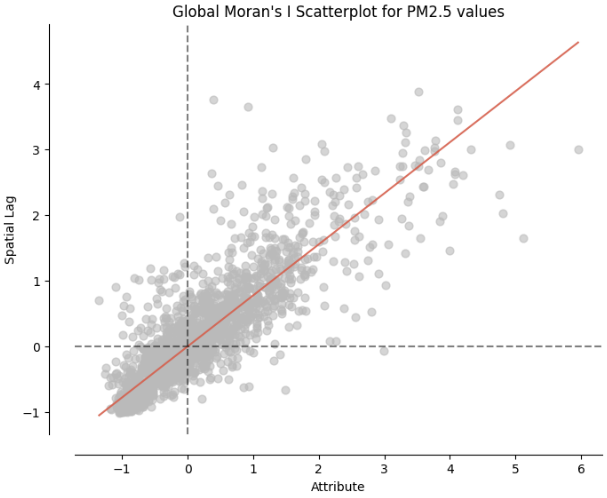


Figure 8: Global MoranScatterplot

7.5 Applying LISA- case study on India

Now looking into a local Auto correlation case, we check for it in the districts of the country by using LISA on that country (like India) and see the convergence and cluster maps.

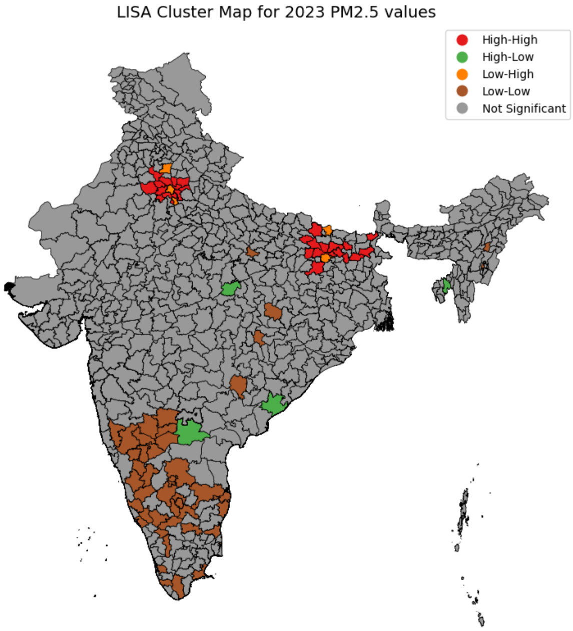


Figure 9: LISA Cluster Map

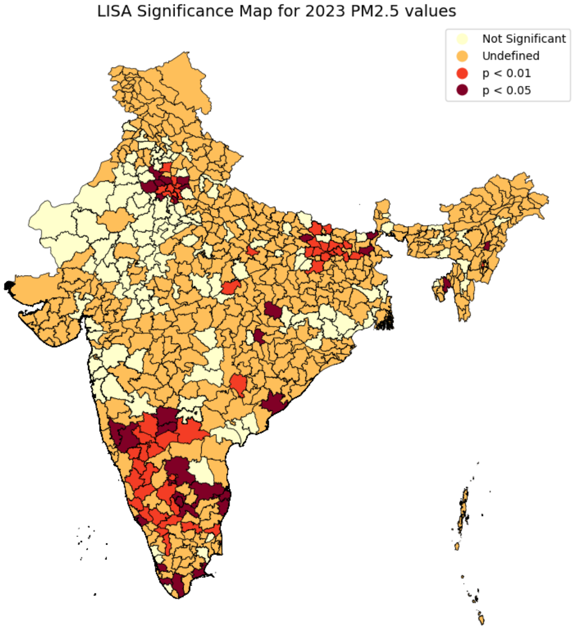


Figure 10: Convergence Map via LISA

7.6 Getis-Ord Statistic

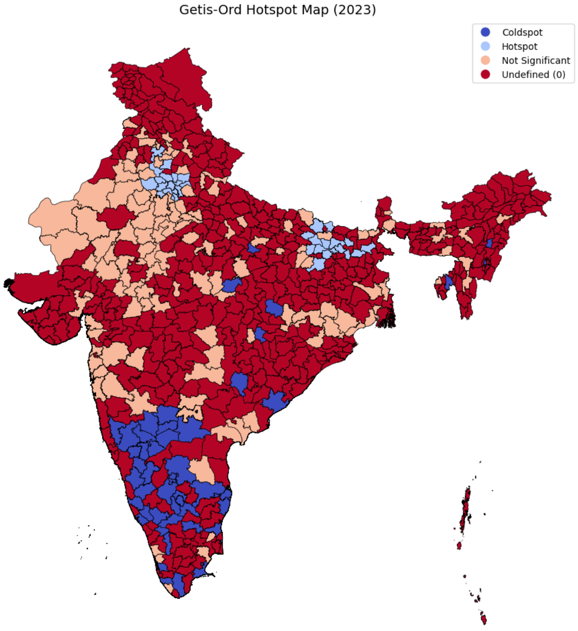


Figure 11: Getis-Ord Hotspot Map

With regard to the above result, the Getis-Ord Hotspot analysis identified areas with significantly high (hotspots) and low (cold spots) PM2.5 concentrations. Hotspots indicate regions with clustered high pollution levels, likely requiring focused intervention. Cold spots suggest areas with consistently low pollution. This helps prioritize locations for air quality management.

7.7 Spatial Correlation and Q-statistic

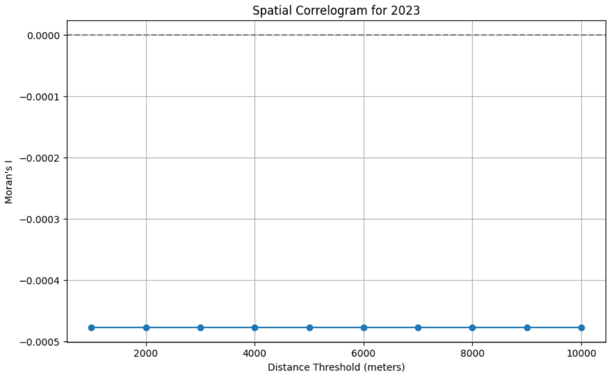


Figure 12: Spatial Correlation Map

The Q-statistic (= 0.36 here) is a global measure of spatial association that evaluates whether high or low values of a variable are spatially clustered, indicating non-random spatial patterns. It is commonly used in conjunction with methods like Getis-Ord General G. The Spatial Correlogram is a graphical tool that plots spatial autocorrelation (e.g., Moran's I) against increasing distance lags, helping to understand how the relationship between values changes with distance. A high positive autocorrelation at short distances suggests spatial clustering, while a drop toward zero or negative values indicates randomness or dispersion. Both tools are essential for exploring spatial dependencies in geospatial data.

7.8 GWR and KDE

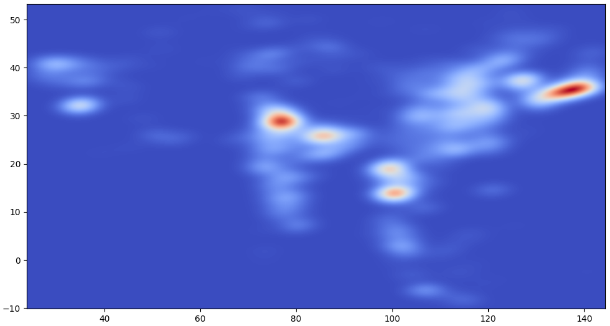
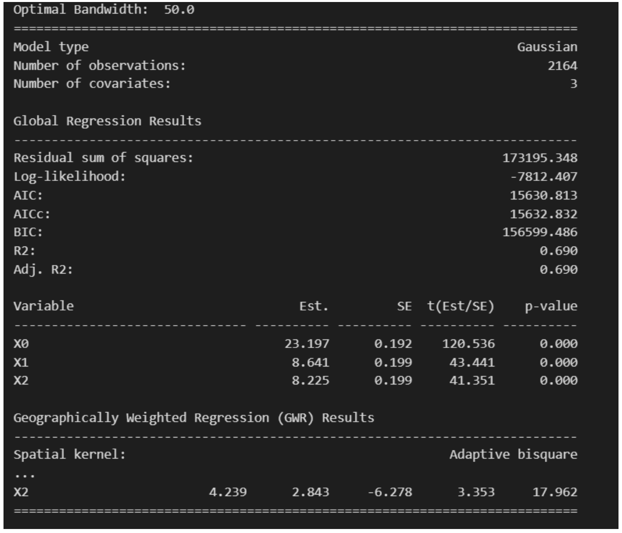


Figure 13: KDE results



```
Optimal Bandwidth:  50.0
===========================================================================
Model type                                                         Gaussian
Number of observations:                                                2164
Number of covariates:                                                     3

Global Regression Results
---------------------------------------------------------------------------
Residual sum of squares:                                         173195.348
Log-likelihood:                                                   -7812.407
AIC:                                                              15630.813
AICc:                                                             15632.832
BIC:                                                             156599.486
R2:                                                                   0.690
Adj. R2:                                                              0.690

Variable                              Est.         SE  t(Est/SE)    p-value
------------------------------- ---------- ---------- ---------- ----------
X0                                  23.197      0.192    120.536      0.000
X1                                   8.641      0.199     43.441      0.000
X2                                   8.225      0.199     41.351      0.000

Geographically Weighted Regression (GWR) Results
---------------------------------------------------------------------------
Spatial kernel:                                           Adaptive bisquare
...
X2                        4.239      2.843     -6.278      3.353     17.962
===========================================================================
```

Figure 14: GWR results

Geographically Weighted Regression (GWR) was applied to model the spatially varying relationship between concentrations and predictors like temperature, humidity, and wind speed. The GWR results revealed significant local variations in regression coefficients, indicating that the influence of predictors on pollutant levels is location-dependent. This highlighted spatial heterogeneity, which global models could not capture effectively. Inspite of the bandwidth blur, an idea of where the results show high and low is clearly conveyed

For Kernel Density Estimation (KDE), the spatial distribution of NO concentrations was visualized, revealing high-density hotspots in urban or traffic-heavy regions. KDE provided an intuitive heatmap, showing pollutant intensity across space without assuming any underlying statistical model. Together, GWR and KDE offered complementary insights—GWR quantified local relationships, while KDE highlighted spatial concentration patterns of pollution.

7.9 Histogram Estimation using Smooth Kernal

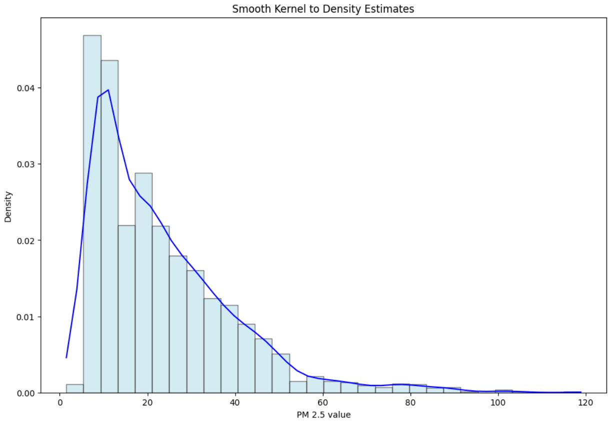


Figure 15: KDE results

The Histogram Estimation using Smooth Kernel Histogram was used to estimate the probability distribution of NO concentrations in a continuous and smoothed manner. Unlike traditional histograms, this method avoids abrupt bin boundaries and provides a clearer view of distribution shape, highlighting peaks and spread in pollutant levels more naturally.

7.10 IDW Method based Interpolation

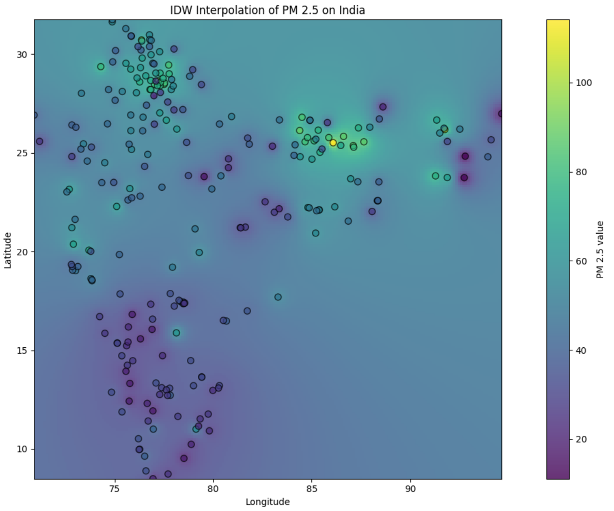


Figure 16: KDE results

For spatial prediction, the Inverse Distance Weighting (IDW) interpolation method was applied to estimate NO levels at unsampled locations. IDW assumes that closer points have more influence on the estimation, making it suitable for dense sensor data. Kriging was not used due to its computational complexity and the requirement for strong assumptions like stationarity and a well-fitted variogram model, which were not feasible with the given dataset and project constraints.

7.11 Data Mining

Here, we use the DBSCAN clustering method for both in the global and local domain. It was used to identify clusters of high pollutant concentrations in both local and global spatial contexts. In the global domain, DBSCAN helped detect large-scale pollution clusters across the entire study region, distinguishing dense polluted zones from sparse or cleaner areas. In the local domain, it was applied to subregions or districts to uncover finer spatial patterns, such as urban hotspots or localized pollution events. The algorithm's ability to handle noise and discover arbitrarily shaped clusters without predefining the number of clusters made it especially effective for analyzing heterogeneous environmental data.

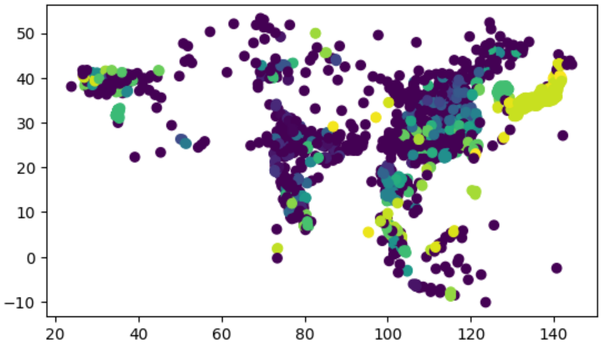


Figure 17: DBSCAN Global Cluster

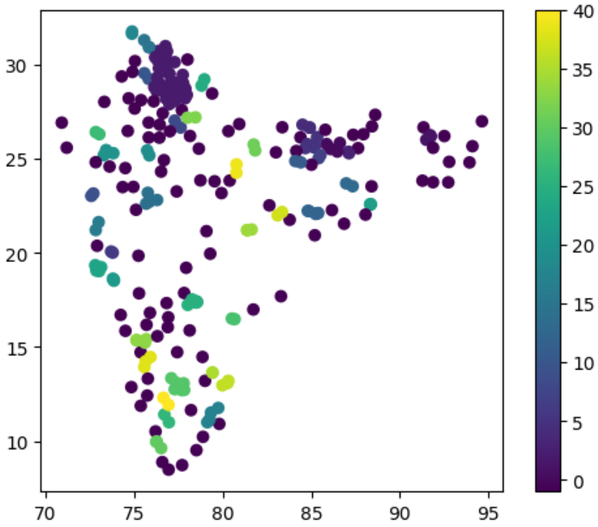
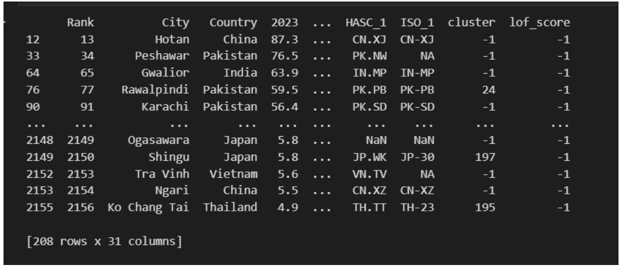


Figure 18: DBSCAN Local Cluster Map



```
        Rank         City   Country  2023  ...  HASC_1  ISO_1  cluster  lof_score
12        13        Hotan     China  87.3  ...   CN.XJ  CN-XJ       -1         -1
33        34     Peshawar  Pakistan  76.5  ...   PK.NW     NA       -1         -1
64        65      Gwalior     India  63.9  ...   IN.MP  IN-MP       -1         -1
76        77   Rawalpindi  Pakistan  59.5  ...   PK.PB  PK-PB       24         -1
90        91      Karachi  Pakistan  56.4  ...   PK.SD  PK-SD       -1         -1
...      ...          ...       ...   ...  ...     ...    ...      ...        ...
2148    2149    Ogasawara     Japan   5.8  ...     NaN    NaN       -1         -1
2149    2150       Shingu     Japan   5.8  ...   JP.WK  JP-30      197         -1
2152    2153    Tra Vinh    Vietnam   5.6  ...   VN.TV     NA       -1         -1
2153    2154        Ngari     China   5.5  ...   CN.XZ  CN-XZ       -1         -1
2155    2156  Ko Chang Tai  Thailand   4.9  ...   TH.TT  TH-23      195         -1

[208 rows x 31 columns]
```

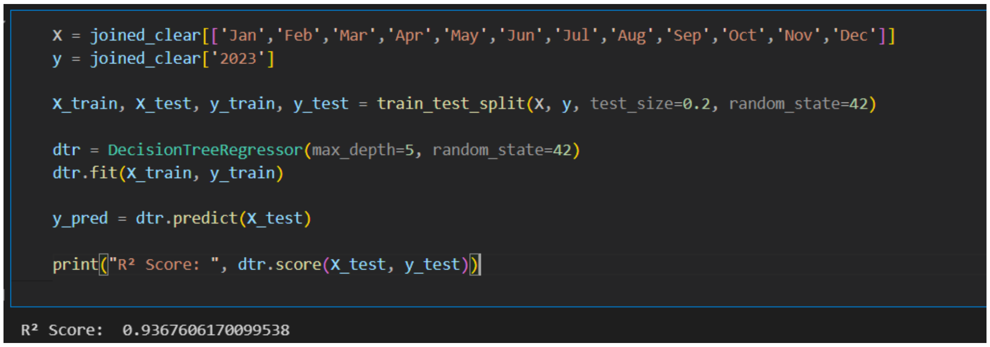
Figure 19: Outliers Via DBSCAN

7.12 Machine Learning Models

Various regression models including Random Forest, XGBoost, and linear regression were used to predict pollutant levels. Model performance was evaluated using metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 score.

7.13 Model Tuning and Cross-Validation

To conclude the analysis, a Decision Tree Regressor (DTR) was employed as the final predictive model for concentration estimations based on meteorological and spatial features. The DTR was chosen for its interpretability and ability to handle non-linear relationships without requiring extensive data preprocessing. It effectively captured decision rules based on temperature, humidity, wind speed, and spatial coordinates. While simpler than ensemble methods, the DTR provided reasonable accuracy and served as a baseline model for spatial pollutant prediction, aligning well with the project's focus on explainability and spatial interpretability.



```
X = joined_clear[['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']]
y = joined_clear['2023']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

dtr = DecisionTreeRegressor(max_depth=5, random_state=42)
dtr.fit(X_train, y_train)

y_pred = dtr.predict(X_test)

print("R² Score: ", dtr.score(X_test, y_test))
```

```
R² Score:  0.9367606170099538
```

Figure 20: Model used and $R^2 score$

8 Contributions

- Rahul- Decision Tree regression (DTR), KDE, LISA,SAC (Moran's I, Geary's C), Report
- Shanmukh - Data Mining, IDW, Histogram Estimation, Getis-Ord Statistic, Data Pre-processing