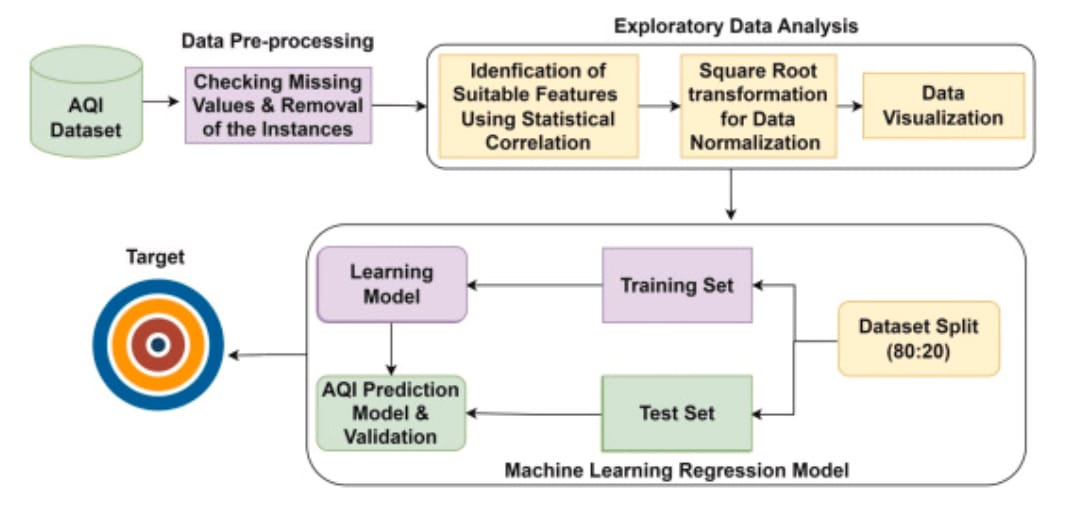
**Air Quality and Traffic Data Analysis**

**1. Introduction**

In recent years, the growing concern over urban air quality has led to increased attention on the relationship between air pollution and various contributing factors such as traffic density, meteorological conditions, and industrial activities. In our analysis, we examined a dataset that comprises air quality parameters—primarily focusing on PM2.5 concentrations—and supplementary variables such as meteorological measures (e.g., temperature, humidity, wind speed) and traffic metrics (e.g., current speed, traffic level). The dataset was collected from multiple stations located in various regions of Delhi over a specified period.

The objective of our analysis was to understand the statistical properties of the dataset, identify key patterns, trends, and potential outliers, and ultimately use this information to build predictive models for PM2.5 concentrations. By doing so, we aim to assist policymakers and urban planners in designing strategies to mitigate air pollution and improve public health outcomes. In this report, we present an overview of the data, a detailed statistical analysis with visualizations, a discussion of our key findings, and our conclusions with actionable insights.

**2. Data Collection**

The weather data was collected from the CPCB (Central Pollution Control Board) website, which provides comprehensive datasets across various monitoring stations. Initially, data from all available stations was downloaded; however, for the purposes of this analysis, only those stations with the most complete records (i.e., with the highest number of available values) were selected. This selection helped ensure that the dataset was robust and minimized the impact of missing data on subsequent analyses.

During the preprocessing phase, several columns that exhibited a high proportion of missing values were removed. In particular, variables such as wind speed, barometric pressure, ambient temperature, and rainfall were discarded because their large number of ‘NaN’ values would have compromised the integrity and reliability of the analysis. This pruning of variables was essential to focus on the most reliable and consistent data.

The time period for weather data collection spanned from 1st January 2024 00:00 to 31st July 2024 23:59, providing a detailed record over seven months. This interval was chosen to capture both diurnal and seasonal variations in air quality across the selected stations.

Subsequently, traffic data was collected using the TomTom APIs. These APIs were leveraged to gather real-time traffic parameters, such as current speed and traffic levels, which are critical for understanding the impact of vehicular flow on air quality. By integrating traffic data with the weather and pollutant measurements from CPCB, the analysis aimed to explore the intricate relationships between traffic patterns, meteorological conditions, and pollutant concentrations.

This systematic approach to data collection—prioritizing quality and completeness—ensured that the analysis could yield reliable insights into the factors influencing air quality in the region.

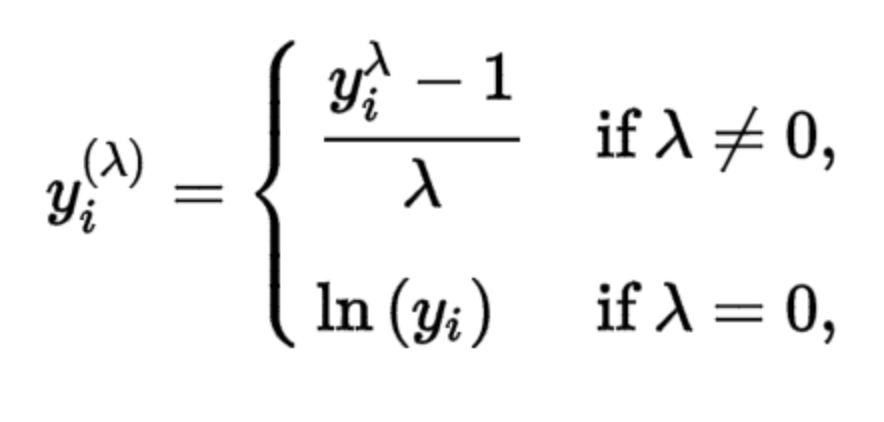
**3. Data Description**

* The dataset we’re analysing captures air quality observations recorded over several months from various monitoring stations across Delhi. With approximately 40,000 observations, it provides a detailed picture of pollution levels and influencing factors.
* Each record includes both time and location details, allowing us to track trends over different periods and areas. The key variables in the dataset include:
* **Timestamp**: The exact date and time of each measurement, which helps identify daily and seasonal pollution patterns.
* **Station Name** : The monitoring station where the reading was taken. The dataset covers multiple locations, including Alipur, Bawana, Dwarka, IHBAS, Jahangirpuri, Najafgarh, and Narela, among others.
* **PM2.5** : These are the fine particulate matter, refers to tiny air pollutants that are 2.5 micrometres or smaller in diameter—about 30 times thinner than a human hair..
* **Other Air Pollutants**: Parameters such as PM10, NO, NO2, NOx, NH3, SO2, CO, Ozone, Benzene, Toluene, and Eth-Benzene that offer a comprehensive view of the overall air quality. these particles can easily bypass our body’s lungs , travel deep into the lungs, and they are very harmful.
* **Meteorological Parameters**: Variables such as Relative Humidity (RH), Wind Direction (WD), Wind Speed , Temperature are crucial for understanding how weather affects pollutant dispersion for example : In particular area if the wind speed is high therefore it is possible that the air pollution in that area will be comparatively low because of transfer of the particulate matter from one area to another.
* **Traffic Data**: This includes parameters like Current Speed, Free Flow Speed, and Traffic Level, which are used to investigate the impact of traffic on air quality as the areas where the traffic congestion is high have very high possibility of having high air pollution compare to other areas where the traffic is low.

The dataset includes both continuous and categorical data, offering a wide view of air quality patterns. Continuous variables, such as PM2.5 levels and meteorological factors, are measured on an interval scale, allowing us to track trends over time (in our dataset it is 1 Hour interval). On the other hand, categorical variables, like Station Name, help provide location-specific insights into air pollution levels across different parts of the city. While the dataset is quite detailed, we did notice some missing values in important columns. To ensure the accuracy of our analysis, we used smart imputation techniques to fill in the gaps. This included time-based interpolation, which estimates missing values based on trends over time, and group-wise median filling, where missing values were replaced using the median of similar data points. This methods integrated the dataset and making sure that our finding is reliable.

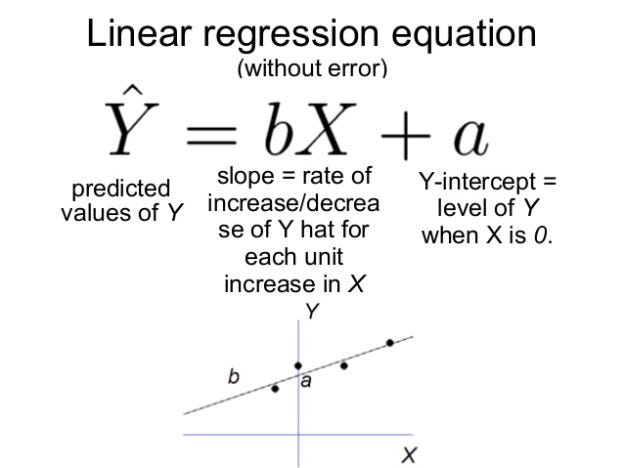
**Yeo-Johnson Transformation**

The Yeo-Johnson transformation is a useful method for handling skewed data distributions, making them more suitable for statistical modeling. Compared to methods such as log transformations that fail with zero or negative values, Yeo-Johnson works with all real numbers. It applies different transformations based on whether values are positive or negative, helps make the data more normal. For example, highly skewed pollution measurements (like PM2.5 concentrations) can be transformed to follow a more symmetrical, bell-shaped distribution. This adjustment helps machine learning algorithms perform better since many models assume normally distributed input data. The transformation is particularly useful in environmental datasets where extreme values are common. By reducing skewness, Yeo-Johnson improves model accuracy while maintaining the data's original meaning and relationships between variables.

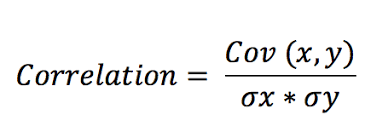


**Linear Regression**

Linear regression is a fundamental predictive modeling technique that establishes a linear relationship between input variables (features) and a continuous output (target). It works by fitting a straight line or a hyperplane that minimizes the difference between predicted and actual values. For instance, it could predict PM2.5 levels using traffic density and weather conditions as inputs. While intuitive and fast to run, linear regression has limitations: it assumes linear relationships between variables and is sensitive to outliers. Despite these constraints, it serves as an excellent baseline model and provides interpretable coefficients that quantify each feature's contribution. When relationships are truly linear, it can be surprisingly effective, but real-world data often requires more sophisticated approaches.

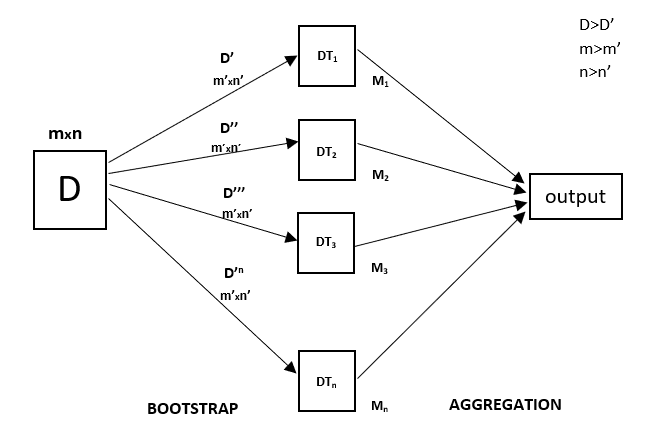


**Correlation Matrix**

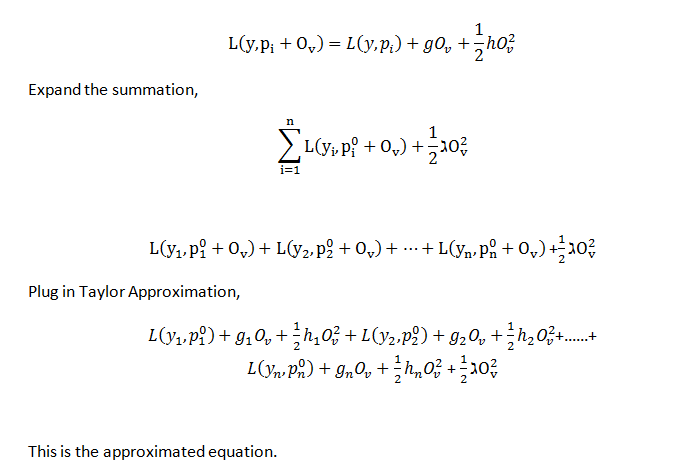
A correlation matrix is a comprehensive table that shows relationships between variables in a dataset, with values ranging from -1 (perfect inverse relationship) to +1 (perfect direct relationship). Zero indicates no linear correlation. This matrix is very helpful for exploratory data analysis, helping find useful patterns. For example, in pollution studies, it might show strong positive correlations between PM2.5 levels and traffic congestion, or negative correlations with wind speed. Beyond identifying key predictors, correlation matrices help detect multicollinearity (when predictors are too closely related), which can distort regression models. They also guide feature selection by highlighting redundant variables. Visualized as heatmaps, correlation matrices provide immediate, intuitive insights into complex datasets, making them essential for both preprocessing and model interpretation.

**Random Forest Regressor**

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs their average prediction. Each tree is built on a random subset of data (bagging) and considers only random feature subsets at each split, increasing diversity. This randomness makes the model robust against overfitting and noise. For predicting PM2.5, a Random Forest might combine hundreds of trees, each capturing different patterns in traffic, weather, and time variables. The model handles non-linear relationships and interactions automatically, requires minimal preprocessing (e.g., no need for feature scaling), and provides feature importance scores. While generally accurate, its "black-box" nature makes interpretation harder than linear models. However, for complex datasets with many variables, Random Forests often deliver superior performance with little tuning.

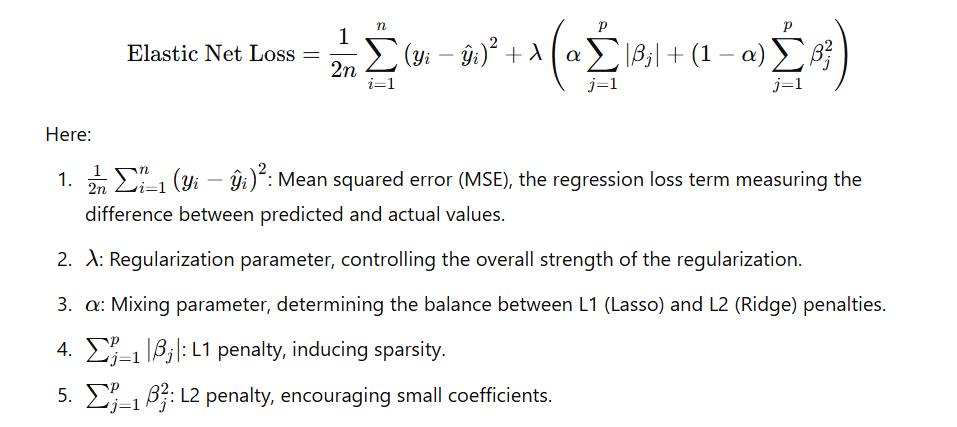


**XGBoost Regressor**

XGBoost (Extreme Gradient Boosting) is an advanced ensemble technique that sequentially builds decision trees, with each new tree correcting errors of its predecessors. It optimizes a loss function (e.g., RMSE) using gradient descent, adding trees until improvements plateau. Key features include regularization (to prevent overfitting), handling missing values, and parallel processing for speed. In PM2.5 prediction, XGBoost might outperform other models by capturing complex interactions between traffic patterns, weather, and time variables. Its flexibility allows tuning via parameters like learning rate, tree depth, and subsampling ratios. XGBoost often performs well in competitions due to its precision and efficiency, though it requires careful hyperparameter tuning. Interpretability tools can illuminate its decision-making, bridging the gap between performance and transparency.

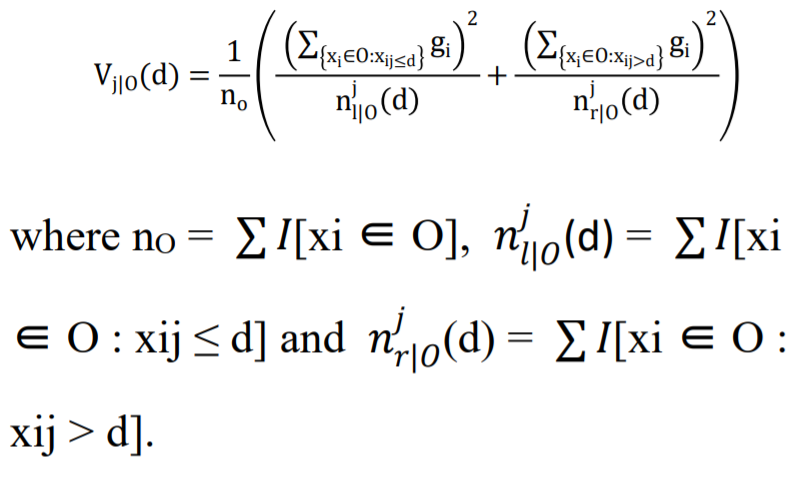
**Elastic Net Regression**

ElasticNet combines the strengths of Lasso (L1) and Ridge (L2) regression, balancing feature selection and regularization. Lasso shrinks some coefficients to zero (effective feature selection), while Ridge reduces coefficient magnitudes (handling multicollinearity). ElasticNet's hybrid approach is ideal when datasets have many correlated predictors (e.g., multiple pollution metrics). For PM2.5 modeling, it might retain only the most influential features (like traffic density and NOx levels) while damping less important ones. The model includes two tuning parameters: α (overall regularization strength) and λ (balance between L1/L2). Though less interpretable than plain linear regression, ElasticNet provides a good balance—simpler than tree-based models yet more adaptable to messy, high-dimensional data than basic linear techniques.



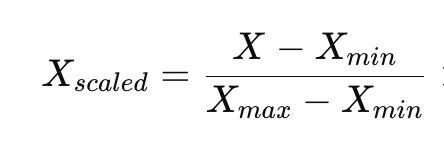
**Light GBM (Light Gradient Boosting Machine)**

LightGBM is a high-performance gradient boosting framework designed for efficiency and accuracy. Unlike traditional boosting algorithms that grow trees level-wise, LightGBM uses a leaf-wise growth strategy, which often leads to better accuracy with fewer trees. It's optimized for large datasets and can handle categorical features directly without needing one-hot encoding. For tasks like PM2.5 prediction, LightGBM excels at capturing complex patterns in traffic, weather, and pollution data while being computationally faster than XGBoost. Key features include:

* Handling missing values automatically.
* GPU support for accelerated training.
* Customizable loss functions for specific needs.  
  Though less interpretable than linear models, tools like SHAP (SHapley Additive exPlanations) can help explain its predictions.

**Min-Max Scaler (for Outliers)**

The Min-Max Scaler is a data normalization technique that rescales features to a fixed range, typically [0, 1]. It works by subtracting the minimum value and dividing by the range (max - min). While not specifically for outliers, it can mitigate their impact by compressing all values into a narrow range. For example, if PM2.5 values range from 10 to 500, Min-Max scaling would transform them proportionally between 0 and 1. However, extreme outliers can still distort the scaling. Alternatives like RobustScaler (using median/IQR) are better for outlier-heavy data. Use cases:

* Preparing data for algorithms sensitive to feature scales (e.g., neural networks).
* Ensuring equal weight for all features in distance-based models (e.g., k-NN).

**Data Augmentation**

Data augmentation artificially expands the training dataset by creating modified copies of existing data. For tabular data (like traffic/pollution datasets), techniques include:

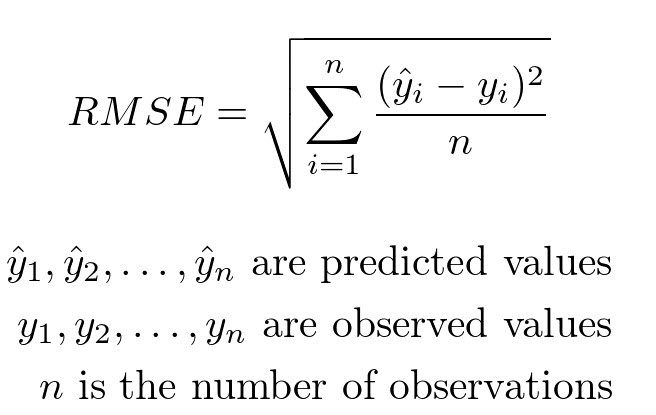
1. Noise Injection: Adding small random values (e.g., Gaussian noise) to numerical features (e.g., PM2.5, traffic speed).
2. Synthetic Sampling: Using methods like SMOTE (for classification) or bootstrapping to generate new samples.
3. Time-Series Perturbations: Shifting timestamps or interpolating missing periods in temporal data.

Why Use It?

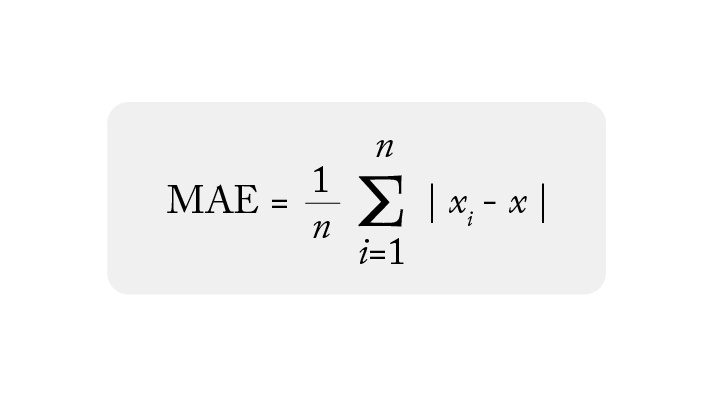
* Reduces overfitting when data is limited.
* Improves model generalization (e.g., for rare high-PM2.5 events).
* Balances class distributions in classification tasks.

**RMSE (Root Mean Squared Error)**

RMSE is a widely used metric for evaluating regression models, calculated as the square root of the average squared differences between predicted and actual values. It gives more weight to large errors (due to squaring) and is expressed in the same units as the target variable (e.g., μg/m³ for PM2.5). An RMSE of 3 means predictions typically deviate by 3 units from reality. Unlike metrics like R², RMSE gives a clear idea of the size of errors. However, it can be sensitive to outliers—a few large errors disproportionately increase RMSE. When comparing models, lower RMSE indicates better accuracy. For environmental monitoring, an RMSE of 2 μg/m³ for PM2.5 predictions might be acceptable, while 10 μg/m³ would be problematic. RMSE is especially useful when large errors are particularly undesirable.

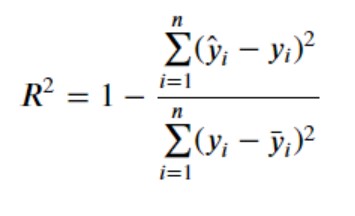


**MAE (Mean Absolute Error)**

MAE measures the average size of errors in model's predictions, giving a clear idea of how far off the estimates are in real-world terms. For example, if you're predicting PM2.5 levels (measured in µg/m³), an MAE of 2.5 means model's predictions are typically off by 2.5 units on average. Unlike RMSE (which squares errors and emphasizes large mistakes), MAE treats all errors equally, making it more intuitive for understanding typical prediction gaps. It’s especially useful when we want a straightforward, easy-to-explain metric—like telling policymakers, "Our air quality predictions are usually within 3 µg/m³ of the real values." However, because MAE doesn’t penalize big outliers heavily, it may hide occasional large errors that could be critical in applications like extreme pollution warnings.

**R² (R-Squared)**

R² tells how well model explains the ups and downs in your data. Think of PM2.5 levels changing due to traffic, weather, etc.—R² quantifies what fraction of those changes your model captures. An R² of 0.80 means 80% of PM2.5 variations are explained by your model, while the rest comes from unmeasured factors or randomness. Unlike MAE/RMSE, R² is has no units, so it’s great for comparing models across different datasets. But high R² doesn’t guarantee good predictions if the model is overfit, and it can be misleading with nonlinear patterns. For decision-makers, R² answers, "Does this model reveal meaningful patterns, or is it just noise?"



**4. Data Visualization**

In this analysis, we started by combining traffic data with weather and pollutant information for different regions and timestamps. One of the biggest challenges was dealing with missing data — especially for environmental factors like PM2.5 or NO2.

Instead of just dropping rows, we applied smart strategies:

• For pollutants, it used time-based interpolation per region, which made sense because pollution levels usually follow trends over time.

• For weather parameters like humidity (RH) and wind direction (WD), the values were filled using grouped medians by month and region — this ensured we captured seasonal and regional variations.

• And for traffic values like speed, forward and backward fill methods were used to preserve the time sequence.

Once the data was cleaned, we checked the region-wise data availability. One key thing I understood is that before doing any modelling, a huge amount of thoughtful preprocessing is needed — especially when dealing with real-world sensor data where missing or noisy values are common.

**Insights from the graphs and the plots:**

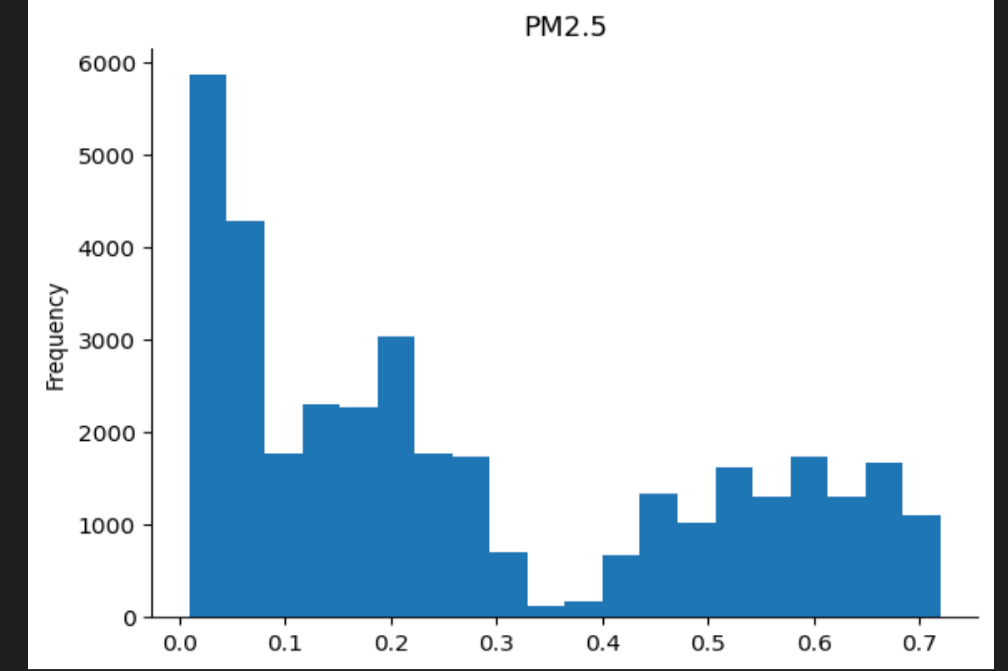
* 1. **Violin Plots for PM2.5 and PM10 by Region:**

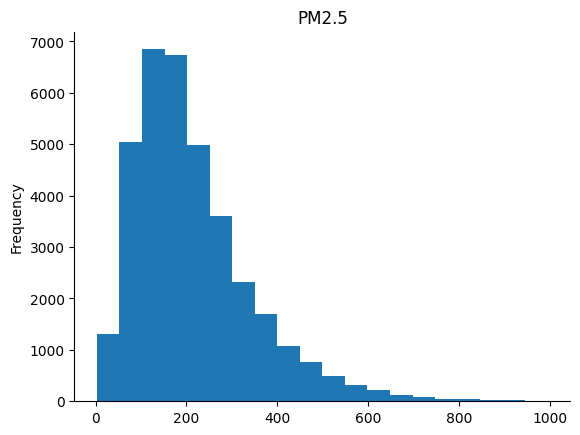
A graph of different colored shapes

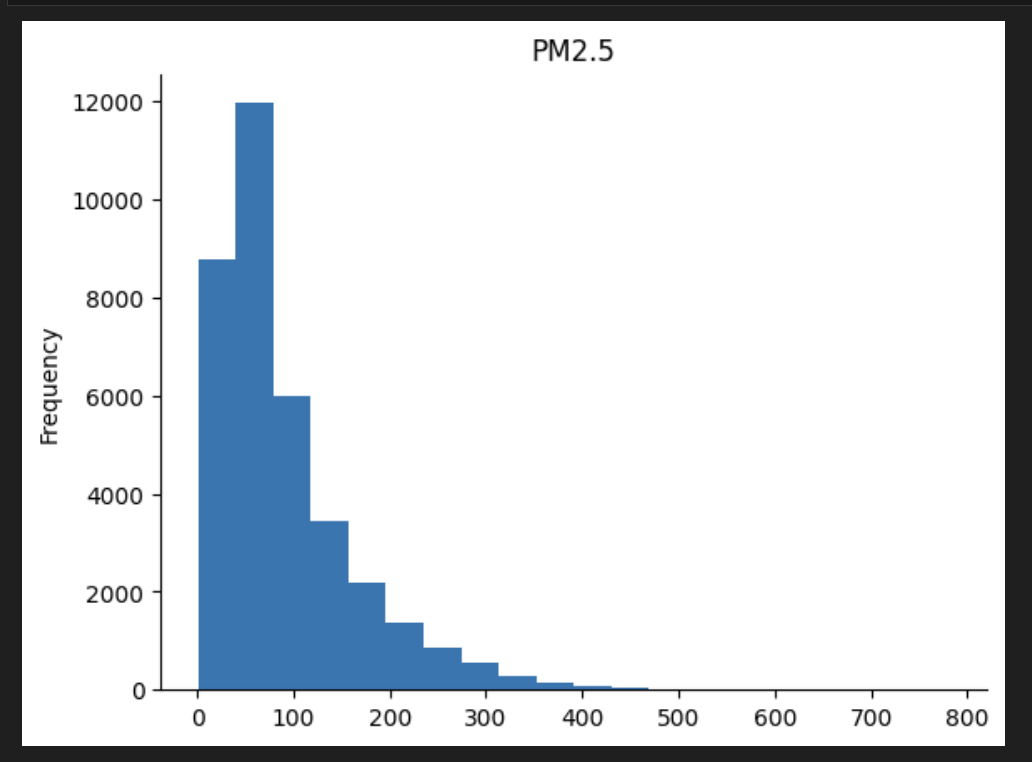
AI-generated content may be incorrect.A graph of different colored shapes

AI-generated content may be incorrect.These plots show how pollution levels vary by region. PM2.5 and PM10 concentrations vary a lot across regions. Some regions have a wider spread and higher concentration of fine particles, which likely points to higher pollution due to traffic or industrial activity .

**2. Histograms of PM2.5, PM10, and Traffic Level:**

These plots show the overall distribution of pollution and traffic data. The pollution data is right-skewed, which means most of the time pollution is low, but there are some bad pollution days with very high values. Similarly, traffic level also has a long tail, indicating that congestion is rare but severe when it happens.





The data was found to be **skewed**, especially pollution and traffic-related variables. To fix this:

They applied **Yeo-Johnson transformation**, which is a more flexible version of log transform. Then they scaled the data using MinMaxScaler (brings values to 0–1 range)

A graph with blue squares

AI-generated content may be incorrect.

Several new features were created to better understand traffic and environmental behaviour:

• **Congestion Factor**: Reflects traffic slowdown as a function of current vs free-flow speed.

• **PM2.5/PM10 Ratio**: Tells whether particles are fine (from combustion) or coarse (like dust).

• **RH × Wind Direction**: How humidity interacts with wind to affect pollution spread.

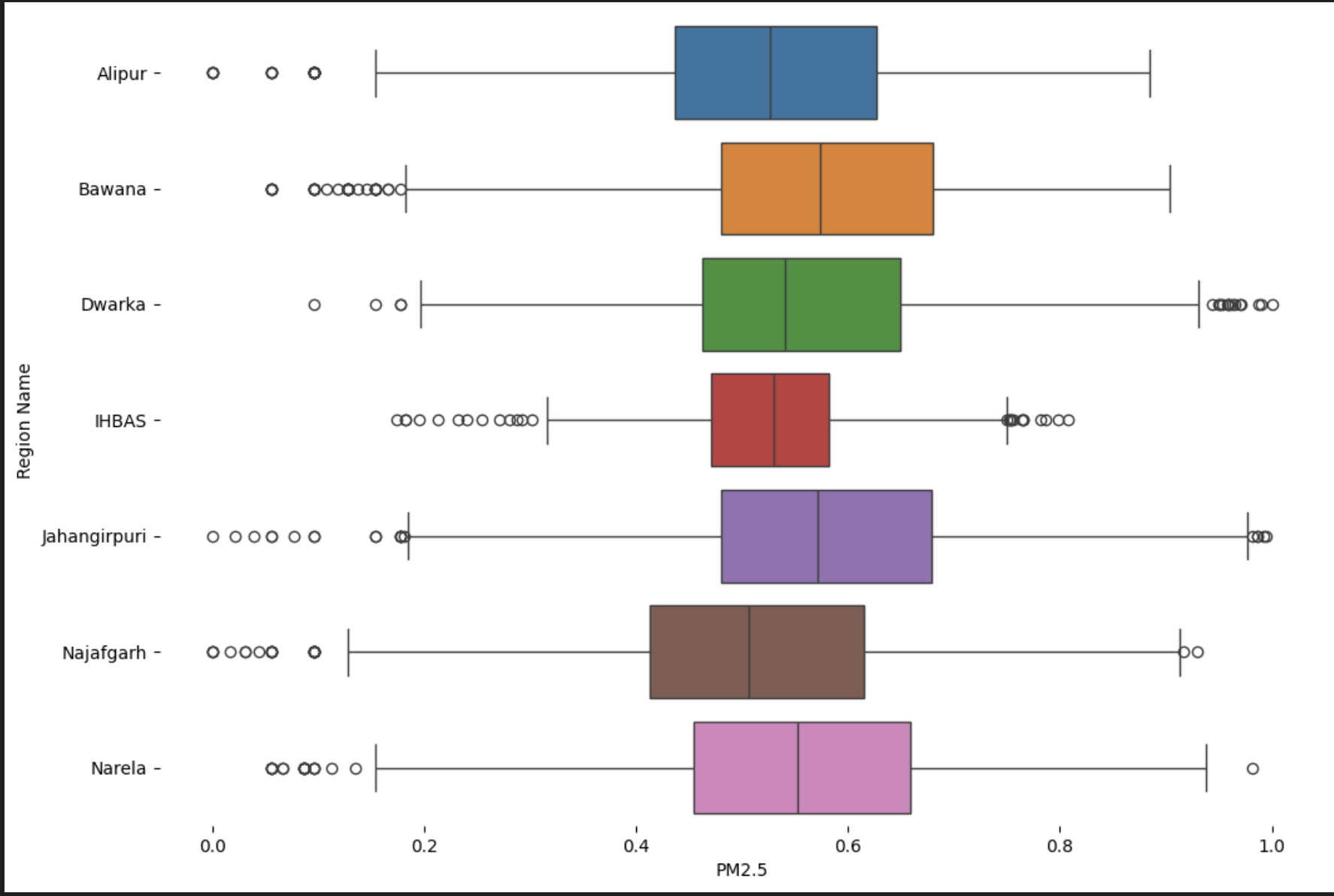
• **Inverse Wind Direction**: Inversely proportional to wind; when wind is slow, pollution tends to stay trapped.

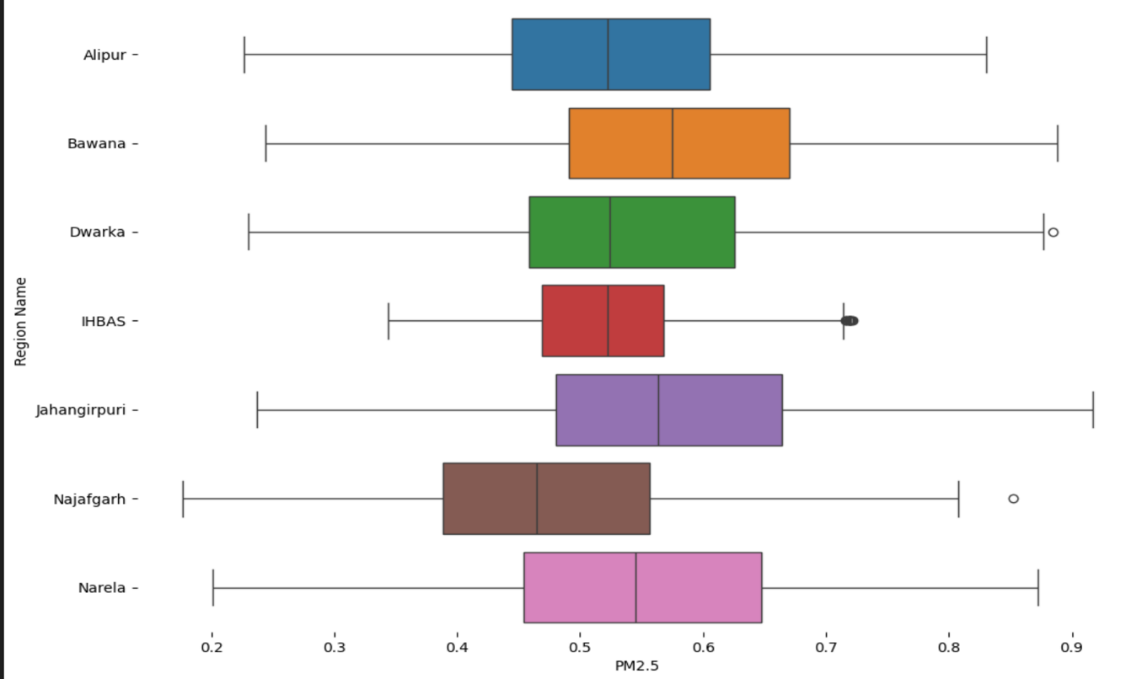
• **Mean of Gases**: A general air quality indicator from multiple gases.

• **Traffic Index**: Combines speed and traffic level to represent mobility.

**4. Boxplots for PM2.5 per Region**

This helped compare **regional air quality patterns** after transforming the data. Some regions consistently have **higher PM2.5 levels**, and there are clear **outliers** in several regions, possibly due to traffic spikes or pollution events. It shows that **location matters a lot** in air quality monitoring.



Box plot when the outliers was cleared.

While analysing the dataset, we found that pollution and traffic patterns vary significantly across different regions. Violin and boxplots clearly showed that PM2.5 and PM10 levels were not evenly distributed — some areas had sharp peaks, probably due to urban congestion or environmental factors.

Another interesting thing was how right-skewed the pollution data was. So we used transformations like **Yeo-Johnson** to normalize it before plotting or modelling and MIN-MAX to scale it . I also really liked how new features like the Congestion Factor and PM2.5/PM10 ratio gave more useful insights than raw data — they helped us understand where traffic slowdown and air quality problems are inter related.

Lastly, region-wise outlier cleaning helped ensure that one noisy location didn’t throw off our analysis. Overall, it gave a much clearer picture of how traffic and air quality interact on a regional level.

**Conclusion:**

Through this project, we got a deeper understanding of how traffic conditions and environmental factors interact across different regions and time periods. One of the first things I realized was that real-world datasets — especially those involving sensors and time-series data — are rarely clean or complete. So, a significant part of the work went into making the data reliable, by using thoughtful techniques like time-based interpolation, grouped median imputation, and intelligent outlier removal.

Once the data was cleaned, we observed the Patterns in PM2.5 and PM10 levels across regions, congestion levels throughout the day, and their relationship with weather features like humidity and wind direction revealed clear trends — for example, how pollution can vary when wind speeds are low, or how traffic congestion directly relates to a drop in average speed.

Creating new features like the Congestion Factor and PM2.5/PM10 ratio helped bring out hidden insights like turning raw numbers into meaningful indicators.