CS361: Air Quality Alert System for Guwahati

Harsh Katara Himanshi Gautam Dhruv Patel Ridhiman Kaur Dhindsa

Abstract

Urban air pollution forecasting is crucial for mitigating its detrimental effects. This study employs feature engineering and various regression techniques to predict the Air Quality Index (AQI) for key pollutants including PM2.5, PM10, CO, NO2, SO2, and O3. Methods such as linear regression, lasso regression, ridge regression, decision tree, random forest, and gradient boosting are implemented both from scratch and utilizing library functions. Evaluation metrics including Mean Square Error and a self-defined metric, Downisde Error are employed to assess model accuracy. Through comprehensive comparison, this research aims to identify the most effective approach for AQI prediction among the ones we've chosen.

1. Introduction

In the pursuit of advancing public health and environmental awareness, our project focuses on the development of an alert system for the Air Quality Index (AQI). The AQI serves as a critical metric for assessing pollution levels and environmental health, guiding individuals and authorities towards informed decisions and proactive measures. By employing meticulous preprocessing, data cleaning, and exploratory data analysis (EDA) techniques, our objective is to uncover underlying patterns within the air quality dataset. This foundational step is integral to the accurate forecasting of AQI, enabling timely alerts that empower communities to take preemptive actions for pollution control and public health management. Through our efforts, we aim to contribute to a safer and healthier environment for all.

1.1. Motivation

At the heart of our mission lies a shared concern – the urgent issue of air pollution and its profound impacts on public health and the environment. Our endeavor to forecast the Air Quality Index (AQI) is driven by the necessity to empower stakeholders with actionable insights for effective pollution mitigation strategies. Through rigorous data analysis involving preprocessing, cleaning, and exploratory techniques, we aim to unveil the complex dynamics of air quality. Our ultimate goal is to furnish decision-makers with evidence-based strategies and interventions, facilitating the creation of cleaner, healthier environments for all. With our efforts focused on an alert system for AQI, we seek to provide timely information that enables proactive measures and safeguards public health.

Recent Reports: Guwahati was reported to be the 2nd most polluted city in the year of 2023 by IQAir, a Swiss Air Quality technology company with average PM2.5 concentration of 105.04 micrograms per cubic meter, which was 21 times the WHO guidelines.
Guwahati with an estimated population of 13 lakh was far cleaner in 2022 when its average annual PM2.5 concentration 51 (μg/m³). The report says, "Causing an estimated one in every nine deaths worldwide, air pollution is the greatest environmental threat to human health. According to the World Health Organization (WHO), air pollution is responsible for an estimated seven million premature deaths worldwide every year.

1.2. Major Challenges

- *Data Availability and Quality:*Obtaining reliable and comprehensive air quality data for India might be challenging. Data may be sparse, inconsistent, or contain missing values, requiring careful preprocessing.
- *Feature Engineering:* Identifying relevant features such as meteorological data (temperature, humidity, wind speed), geographical factors, and historical AQI values and engineering them effectively for model input.
- *Model Selection:* Choosing an appropriate machine learning model that can effectively capture the complex relationships between various factors affecting air quality and provide accurate forecasts values and engineering them effectively for model input.
- *Interpretability:* Ensuring the interpretability of the model outputs so that stakeholders can understand the factors influencing air quality predictions and trust the forecasts.

2. Data Pre-Processing

2.1. Data Preparation

The data is acquired from the Central Pollution Control Board (CPCB) pertaining to pollutant concentrations. The primary objective of this analysis was calculation of the Air Quality Index (AQI) based on the acquired data.The resultant AQI values were categorized into distinct air quality levels, ranging from 'Good' to 'Severe'.

- The AQI calculation uses 7 measures: PM2.5, PM10, SO2, NOx, NH3, CO and O3.
- For PM2.5, PM10, SO2, NOx and NH3 the average value in last 24-hrs is used with the condition of having at least 16 values.
- For CO and O3 the maximum value in last 8-hrs is used.
- Each measure is converted into a Sub-Index based on pre-defined groups.
- Sometimes measures are not available due to lack of measuring or lack of required data points.
- Final AQI is the maximum Sub-Index with the condition that at least one of PM2.5 and PM10 should be available and at least three out of the seven should be available.

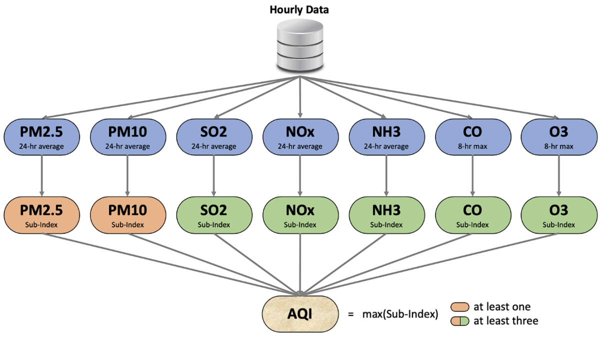
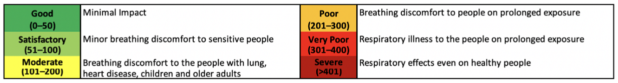


Figure 1. How AQI is calculated

- There is no theoretical upper value of AQI but it is rare to find values over 1000.
- The AQI values are divided into buckets.
- The pre-defined buckets of AQI are as follows:



Good (0–50)	Minimal Impact	Poor (201–300)	Breathing discomfort to people on prolonged exposure
Satisfactory (51–100)	Minor breathing discomfort to sensitive people	Very Poor (301–400)	Respiratory illness to the people on prolonged exposure
Moderate (101–200)	Breathing discomfort to the people with lung, heart disease, children and older adults	Severe (>401)	Respiratory effects even on healthy people

Figure 2. AQI Buckets

2.2. Data Analysis

This section presents a comprehensive analysis of the air quality dataset using statistical methods and visualizations. Various analyses, including seasonality, correlation, weekly patterns, scatter plots, and trend assessments, are conducted using pandas, matplotlib, and seaborn.

2.2.1. Seasonal Analysis

We analysed the calculated AQI dataset for four different stations namely: IIT Guwahati,Pan bazaar, Guwahati airport and railway colony for each month seasonally.
Observation:

- There seems to be a peak in pollution during winter season in Dec-Jan. The main reason for this could be that cold air is denser and moves slower than warm air. This density means that cold air traps the pollution but also doesn't whisk it away.
- There is a peak in August at Airport in the graph but this is not a valid observation as it is generated because not enough data is available for that month.

2.2.2. Correlation Analysis of Air Quality Indicators

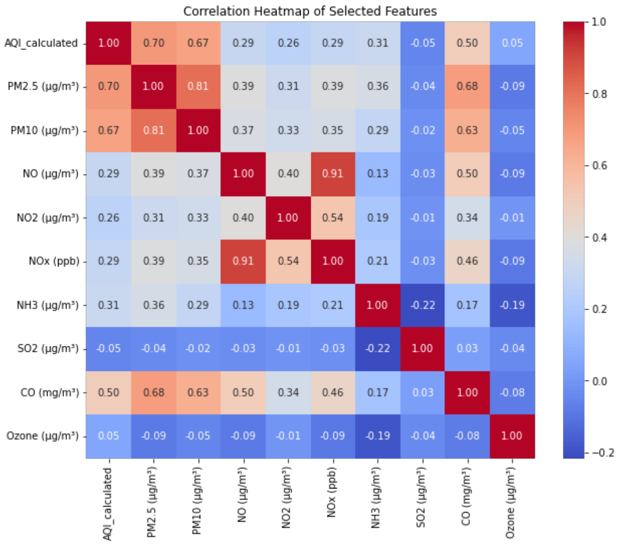


Figure 3. Weekly Analysis

The correlation heatmap (Figure 3) illustrates pairwise relationships between air quality indicators. Key observations include:

- **Positive Correlation:** Strong positive correlations are observed between PM2.5 and PM10, as well as between NO, NO2, and NOx, and between C0 and PM2.5 indicating shared emission sources.
- **Negative Correlation:** AQI shows a negative correlation with SO2 concentrations. It could suggest that factors such as particulate matter or ozone have a stronger influence on overall air quality, or that localized dispersion patterns play a role in lowering AQI values despite elevated SO2 levels.
- **Weak Correlation:** Some pollutants exhibit weak correlations, such as NH3 with other pollutants, suggesting independent sources or atmospheric behaviors.

2.2.3. Scatter plot Analysis

We have created scatter plots for concentration of each pollutant v/s AQI and analysed their relationship.
Observation:

- PM10 has a limit of 1000, sensor cannot measure more than 1000.This is evident from the scatter plot of PM10.
- PM10 and PM2.5 are the major Pollutants.

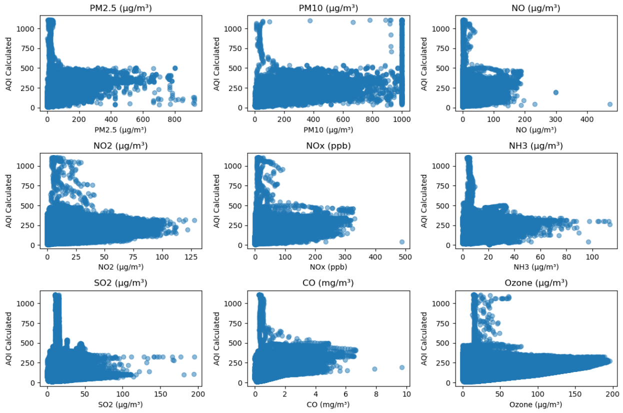


Figure 4. Scatter plots

2.2.4. Weekly Analysis

We analysed the calculated AQI dataset for four different stations namely: IIT Guwahati,Pan bazaar, Guwahati airport and railway colony for day of the week and compared them.
Observation

- At Airport,Railway colony and Pan Bazaar pollution peaks in weekends whereas in IITG pollution doesn't the reason behind this could be that in weekend the former three station are visited more frequently by people and are more crowed. Like due to more frequency of flights the pollution at airport increases. Frequency of vehicles on road increases which leads to more traffic which in turn increases AQI in pan Bazaar.

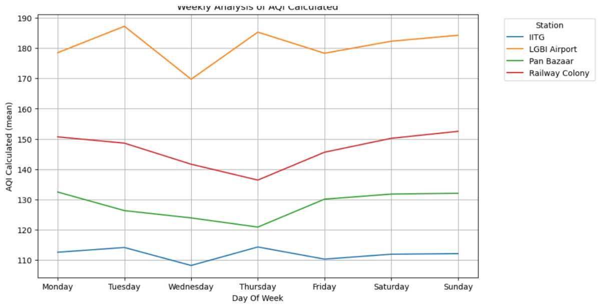


Figure 5. Weekly Analysis

2.2.5. Data availability and Trend Analysis

In this section, we evaluate data availability and trends in Air Quality Index (AQI) values across monitoring stations over time. We scrutinize data distribution to ensure reliability and pinpoint any gaps or irregularities.
Observation: As you can see sudden spikes are there in the graph which indicates that there are missing data around thoe values.

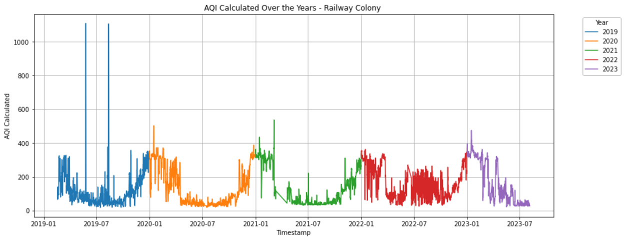


Figure 6.

2.3. Feature Engineering

In our investigation into AQI prediction across four distinct stations over multiple years, an exploratory data analysis (EDA) was conducted to discern prevailing patterns and trends. Notably, our analysis revealed pronounced seasonality trends in several features, namely the day of the week, month, year, and station. By integrating these insights, our proposed approach aims to capture the nuanced dynamics inherent in urban air quality fluctuations, thereby enhancing the predictive accuracy and robustness of our models. This strategic inclusion of seasonality features serves as a pivotal step towards achieving comprehensive and precise AQI forecasting.

3. Models Used

3.1. Linear Regression

Linear regression is represented mathematically as

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \text{error},$$

where y is the dependent variable, $x_1, x_2, \ldots, x_p$ are the independent variables, α is the intercept, and $\beta_1, \beta_2, \ldots, \beta_p$ are the coefficients. The loss function used for linear regression is the mean squared error (MSE), given by:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Here n is the number of observations, y_i are the observed dependent variable values, and $\hat{y}_i$ are the predicted values. The `predict` method calculates the predicted values $\hat{y}$ for a given model and input data X. The `fit` method determines the optimal model parameters that minimize the mean squared error (MSE) using the OLS formula: $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$, where $\mathbf{X}$ is the design matrix of independent variables and $\mathbf{y}$ is the vector of observed dependent variable values. Overall, this code provides a mathematical foundation for linear regression modeling, allowing for the prediction of dependent variable values based on independent variables.
It is widely used due to its simplicity and interpretability, making it a cornerstone in statistical modeling and a good starting point for regression tasks. However, it assumes a linear relationship between the features and the target variable, which may limit its effectiveness in capturing complex patterns in the data.

3.2. Lasso Regression

Lasso regression, an extension of linear regression, introduces L1 regularization to the standard least squares cost function. Let X be the design matrix and y the target vector. The objective is to minimize the following cost function:

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{n}|w_j|$$

Here, m denotes the number of samples, n represents the number of features, w is the vector of regression coefficients, $\hat{y}$ is the predicted output, and λ is the regularization parameter. The first term measures the squared difference between actual and predicted values, while the second term imposes a penalty on the absolute values of the regression coefficients. This penalty encourages sparsity in the coefficient estimates, effectively performing feature selection. The model is trained using gradient descent optimization to update the coefficients iteratively, balancing between fitting the training data and minimizing the regularization penalty. By tuning the regularization parameter λ, Lasso regression offers a flexible approach to linear regression with feature selection capabilities, particularly beneficial in high-dimensional datasets.
It extends linear regression by adding a penalty term to the loss function, promoting sparsity in the coefficient vector and providing automatic feature selection. This regularization technique helps prevent overfitting and can handle high-dimensional data efficiently.

3.3. Ridge Regression

Ridge regression, an extension of linear regression, introduces L2 regularization to the standard least squares cost function. Let X be the design matrix and y the target vector. The objective is to minimize the following cost function:

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{n}w_j^2$$

Here, m denotes the number of samples, n represents the number of features, w is the vector of regression coefficients, $\hat{y}$ is the predicted output, and λ is the regularization parameter. The first term measures the squared difference between actual and predicted values, while the second term imposes a penalty on the squared magnitudes of the regression coefficients. This penalty encourages smaller coefficient values, effectively reducing the impact of individual features and mitigating overfitting. The model is trained using gradient descent optimization to update the coefficients iteratively, balancing between fitting the training data and minimizing the regularization penalty. By tuning the regularization parameter λ, Ridge regression offers a flexible approach to linear regression with regularization capabilities, particularly beneficial in scenarios with multicollinearity among features.
Similar to Lasso, Ridge Regression adds a regularization term to the loss function, but it uses the L2 norm instead, resulting in smoother coefficient estimates. This regularization technique is particularly useful when dealing with multicollinearity among the features, as it stabilizes the model's parameter estimates.

3.4. Decision Tree

$$\text{RSS} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The decision tree methodology is a widely-used algorithm in machine learning and data mining, particularly for classification and regression tasks. Our implementation of it begins by initializing a root node containing the entire dataset and iteratively explores potential splits based on randomly selected predictors. The splitting criterion aims to maximize the reduction in the residual sum of squares (RSS) within resulting child nodes. For quantitative predictors, thresholds are determined by evenly dividing the range of values, while for categorical predictors, splits are based on randomly selected subsets of unique values. The process continues until a stopping criterion, such as maximum tree depth or minimum node size, is reached. Each node tracks information including its ID, parent ID, depth, and whether it is a leaf node. Predictions are made by traversing the tree based on feature values, ultimately arriving at leaf nodes where predictions are based on the mean of the target variable within that node's samples. Overall, this approach builds a flexible regression tree capable of capturing non-linear relationships between predictors and the target variable.

Decision trees are versatile models capable of capturing complex non-linear relationships in the data, making them suitable for both classification and regression tasks. They offer interpretability and can handle both numerical and categorical data without requiring feature scaling.

3.5. Random Forest

The *RandomForestRegressor* constructs an ensemble of decision trees, where each tree is trained independently on a random subset of the training data. Let T denote the total number of trees in the forest. During training, for each tree t in T, a bootstrap sample of the training data is drawn, and a decision tree of maximum depth `max_depth` is fitted to this sample. The model aggregates predictions from all trees by averaging them to produce the final output. Mathematically, the prediction $\hat{y}_i$ for each sample i in the test set is computed as the mean of predictions from all trees:

$$\hat{y}_i = \frac{1}{T}\sum_{t=1}^{T}\text{tree}_t(\mathbf{x}_i)$$

where $\text{tree}_t(\mathbf{x}_i)$ represents the prediction of the t-th tree for the input sample $\mathbf{x}_i$. This ensemble approach effectively reduces overfitting and enhances the model's predictive accuracy, making it suitable for regression tasks. Additionally, hyperparameters such as the number of trees T, maximum depth of each tree, and minimum samples required to split a node can be tuned to optimize model performance.

3.6. Gradient Boosting

Gradient Boosting is an ensemble learning technique that sequentially fits a series of weak learners, typically Decision Trees, to the negative gradient of a loss function with respect to the predicted values. Let $F_0(x)$ be the initial prediction, initialized as the mean of the target variable. At each iteration t, the negative gradient of the loss function $-\frac{\partial L(y,F(x))}{\partial F(x)}$ is computed, where L is the loss function. A new weak learner, denoted as $h_t(x)$, is trained to predict these negative gradients. The predictions of the weak learner are scaled by a learning rate η and added to the current prediction $F(x)$, updating it as $F(x) + \eta h_t(x)$. This process is repeated for a predefined number of iterations. The final prediction is obtained as the sum of all individual weak learner predictions $F(x) = F_0(x) + \sum_{t=1}^{T}\eta h_t(x)$. By iteratively minimizing the loss function, Gradient Boosting effectively learns a strong predictive model from a series of weak learners.

Gradient Boosting builds strong predictive models by sequentially adding weak learners, typically decision trees, and optimizing them to minimize the residual errors. This iterative approach results in highly accurate predictions and is particularly effective in handling heterogeneous data with varying feature importance.

4. Evaluation Metric (MSE)

4.1. Mean Square Error

We employed the Mean Squared Error (MSE) as one of our primary evaluation metrics for quantifying the error of our models and for comparative analysis with models from the scikit-learn library. MSE, a widely accepted measure in regression tasks, calculates the average squared differences between predicted values and actual values. By utilizing MSE, we were able to assess the performance of our models in terms of prediction accuracy and to make informed comparisons with industry-standard approaches implemented in scikit-learn

4.2. Down Size Error (DSE)

We introduce our proprietary evaluation metric, the Down Size Error (DSE), designed to quantify the average negative error present in our model predictions.

The rationale behind considering negative error stems from its potential impact on public health, particularly concerning sensitive individuals. A negative error in the Air Quality Index (AQI) prediction suggests that the actual AQI value exceeds the predicted value, potentially leading individuals to underestimate pollution levels and forego necessary precautions, such as wearing pollution masks. By computing the DSE, we aim to highlight instances where our model underestimates AQI values, providing valuable insights into potential risks and informing the development of more accurate forecasting models to safeguard public health.

5. Results

5.1. Without Feature Engineering

Table 1. Mean Square Error (MSE)

Model Name	Our Result	Sklearn result
Linear Regression	6661.4984	6661.4984
Lasso Regression	6674.4869	6677.4766
Ridge Regression	6661.2710	6660.3983
Decision Tree	2722.1665	2899.0544
Random Forest	1287.0119	1277.8464
Gradient Boosting	1174.7717	1344.9871

Table 2. Down side error (DSE)

Model Name	Our Result	Sklearn result
Linear Regression	4242.0040	4242.0040
Lasso Regression	4274.3934	4243.1611
Ridge Regression	4258.0875	4240.2167
Decision Tree	1473.2392	1544.4888
Random Forest	551.9467	547.3786
Gradient Boosting	709.4935	752.5551

5.2. With Feature Engineering

Table 3. Mean Square Error (MSE)

Model Name	Our Result	Sklearn result
Linear Regression	6530.9969	6530.9969
Lasso Regression	6548.6875	6554.4219
Ridge Regression	6529.6065	6528.7336
Decision Tree	587.8481	782.5233
Random Forest	343.5356	339.0998
Gradient Boosting	498.20000	404.4660

Table 4. Down side error (DSE).

Model Name	Our Result	Sklearn result
Linear Regression	4173.4740	4173.4740
Lasso Regression	4213.2328	4181.7961
Ridge Regression	4186.7543	4168.8758
Decision Tree	297.8543	345.6849
Random Forest	127.8588	124.0062
Gradient Boosting	318.2624	224.5465

6. Conclusion

In our study aimed at forecasting AQI for Guwahati across four distinct stations, extensive data preprocessing and exploratory data analysis (EDA) were conducted to unveil underlying trends and patterns. Through meticulous feature engineering guided by EDA insights, new features were incorporated to enhance predictive capabilities. We systematically implemented various regression models, including linear regression, lasso regression, ridge regression, decision tree, random forest, and gradient boosting, from scratch to evaluate their performance. Our findings revealed that random forest outperformed other models, emerging as the optimal choice for AQI prediction. Additionally, we explored the utility of Principal Component Analysis (PCA) for dimensionality reduction, but observed a degradation in results, indicating its limited effectiveness in this context. Ultimately, our study culminated in the successful development of an AQI forecasting system tailored to the specific needs of Guwahati.

7. Resources and Links

- Dataset Source
- Pollution Report- Guwahati, The Times of India

References

Bhattacharya, S. and Shahnawaz, S. Using machine learning to predict air quality index in new delhi. Technical report, Jadavpur University, 2021. URL https://arxiv.org/abs/2112.05753.

Kaur J., S. S. and S., P. K. Forecasting of aqi (pm2. 5) for the three most polluted cities in india during covid-19 by hybrid daubechies discrete wavelet decomposition and autoregressive (db-dwd-arima) model. Technical report, Environmental Science and Pollution Research, 2023. URL https://doi.org/10.1007/s11356-023-29501-w.

Kumar, A. and Goyal, P. Forecasting of daily air quality index in delhi. Technical report, Science of The Total Environment. URL https://www.sciencedirect.com/science/article/pii/S0048969711009661.