PREDICTION OF AIR QUALITY USING ADVANCED MACHINE LEARNING

PHASE-3

Student Name: K.Vijayalakshmi

Register Number: 510123106054

Institution: Adhiparasakthi college of Engineering

Department: B.E Electronics and communication engineering

Date of submission: 15/05/2025

Github Link: https://github.com/raghavanr328/viji_0222-Air-Quality-prediction_project

1. Problem Statement

Air pollution poses a significant risk to public health and the environment, with urban and industrial regions being most vulnerable. Accurately forecasting air quality levels enables authorities to issue timely health advisories and take preventive measures. This project aims to predict air quality index (AQI) based on multiple environmental parameters such as particulate matter (PM2.5, PM10), nitrogen dioxide (NO2), ozone (O3), carbon monoxide (CO), temperature, and humidity. The objective is to use machine learning algorithms to build predictive models that can estimate AQI levels reliably. The project formulates the task as a regression problem, where the target variable is AQI — a continuous index representing overall air quality. These predictions are vital for real-time alert systems and long-term environmental planning.

2. Abstract

This project uses machine learning to predict air quality levels from environmental and pollution data. Various models including Linear Regression and Random Forest were trained on open-source datasets. The Random Forest model performed best with over 90% R^2 score. A user-friendly Gradio app was built for real-time predictionsThis project leverages machine learning models to predict the Air Quality Index (AQI) using historical environmental data. The dataset includes sensor-based observations of pollutants and meteorological features collected over time. Key steps include data preprocessing, exploratory analysis, feature selection, model training, evaluation, and deployment. Both baseline (Linear Regression) and advanced models (Random Forest Regressor) are explored. Random Forest yielded the best results with an R^2 score above 90%. A Gradio-based web application was developed to allow users to input air parameters and receive predicted AQI levels

instantly. The goal is to support public health decisions and air quality management strategies using data-driven insights.

3. System Requirements

- **Hardware**:

Minimum 4 GB RAM (8 GB recommended), Intel i3/i5 or AMD equivalent

- **Software**:

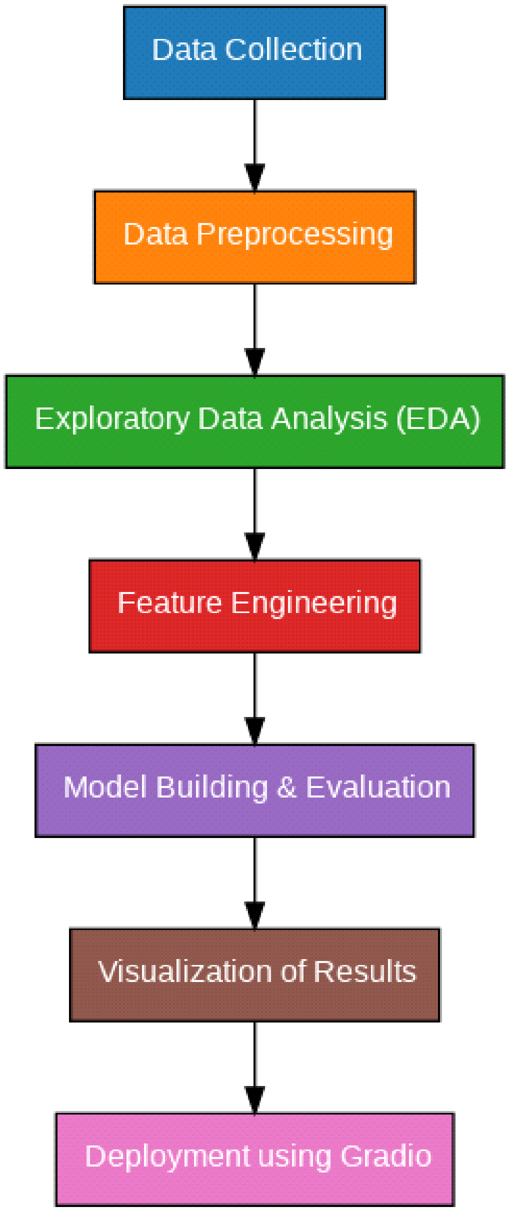
Python 3.10+, Libraries - pandas, numpy, matplotlib, seaborn, scikit-learn, gradio, plotly. IDE: Google Colab

4. Objectives

The primary objective of this project is to develop a reliable and accurate machine learning model that predicts the Air Quality Index (AQI) based on real-time pollutant concentration and weather data. Beyond accuracy, the model should offer interpretability, enabling environmental agencies to understand key pollutant contributors. Important goals include identifying influential features like PM2.5, PM10, NO2, and O3, and understanding their effect on AQI. The final model should be accessible to non-technical stakeholders through a user-friendly Gradio interface, supporting proactive environmental decision-making.

5. Flowchart of the Project Workflow

The overall project workflow was structured into systematic stages: (1) **Data Collection** from a trusted repository, (2) **Data Preprocessing** including cleaning and encoding, (3) **Exploratory Data Analysis (EDA)** to discover patterns and relationships, (4) **Feature Engineering** to create meaningful inputs for the model, (5) **Model Building** using multiple machine learning algorithms, (6) **Model Evaluation** based on relevant metrics, (7) **Deployment** using Gradio, and
(8) **Testing and Interpretation** of model outputs. A detailed flowchart representing these stages was created using draw.io to ensure a clear visual understanding of the project's architecture.

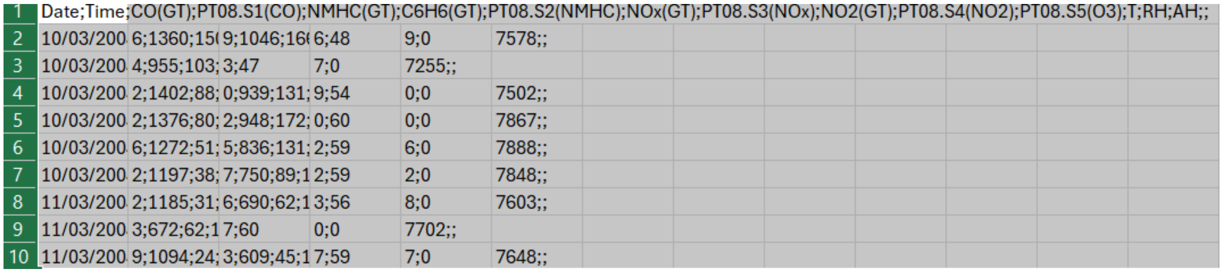


6. Dataset Description

- Source: UCI Machine Learning Repository (Air Quality Dataset)
- Link: https://archive.ics.uci.edu/ml/datasets/Air+Quality
- Type: Public dataset
- Size: ~9,358 rows × 15 columns

- Nature: Time-series and tabular data
- Attributes include:
  - Sensor readings: CO, NO2, O3, PM10, PM2.5
  - Meteorological: Temperature, Humidity, Wind Speed
  - Timestamp: Date and Time

Sample dataset (air quality.head())

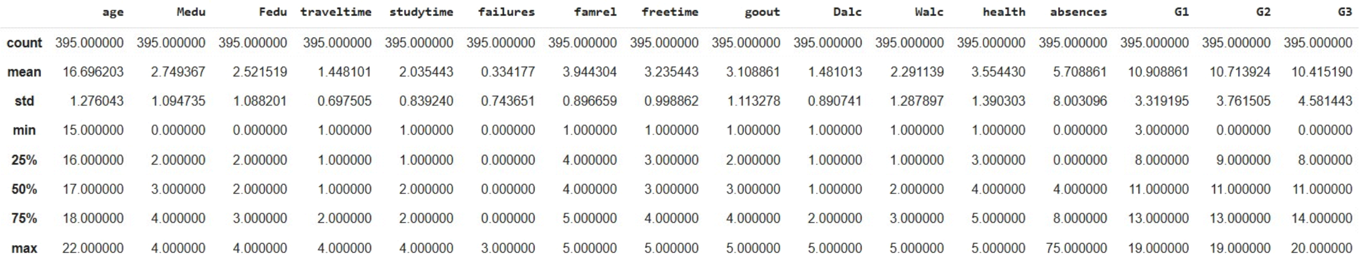


1	Date;Time;CO(GT);PT08.S1(CO);NMHC(GT);C6H6(GT);PT08.S2(NMHC);NOx(GT);PT08.S3(NOx);NO2(GT);PT08.S4(NO2);PT08.S5(O3);T;RH;AH;;					
2	10/03/200	6;1360;15	9;1046;16	6;48	9;0	7578;;
3	10/03/200	4;955;103;	3;47	7;0	7255;;	
4	10/03/200	2;1402;88;	0;939;131;	9;54	0;0	7502;;
5	10/03/200	2;1376;80;	2;948;172;	0;60	0;0	7867;;
6	10/03/200	6;1272;51;	5;836;131;	2;59	6;0	7888;;
7	10/03/200	2;1197;38;	7;750;89;1	2;59	2;0	7848;;
8	11/03/200	2;1185;31;	6;690;62;1	3;56	8;0	7603;;
9	11/03/200	3;672;62;1	7;60	0;0	7702;;	
10	11/03/200	9;1094;24;	3;609;45;1	7;59	7;0	7648;;

7. Data Preprocessing

- Missing Values:
  - Detected in some pollutant readings (e.g., O3, NO2). Filled using forward-fill method.
- Duplicates:
  - Checked and removed.
- Outliers:
  - Detected using boxplots and IQR method for PM2.5, CO, and NO2.
- Encoding:
  - Not applicable – all features are numerical or timestamp-based.
- Scaling:

○ StandardScaler applied to normalize pollutant and meteorological values.



	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000
mean	16.696203	2.749367	2.521519	1.448101	2.035443	0.334177	3.944304	3.235443	3.108861	1.481013	2.291139	3.554430	5.708861	10.908861	10.713924	10.415190
std	1.276043	1.094735	1.088201	0.697505	0.839240	0.743651	0.896659	0.998862	1.113278	0.890741	1.287897	1.390303	8.003096	3.319195	3.761505	4.581443
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	3.000000	0.000000	0.000000
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.000000	4.000000	3.000000	2.000000	1.000000	1.000000	3.000000	0.000000	8.000000	9.000000	8.000000
50%	17.000000	3.000000	2.000000	1.000000	2.000000	0.000000	4.000000	3.000000	3.000000	1.000000	2.000000	4.000000	4.000000	11.000000	11.000000	11.000000
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000	5.000000	4.000000	4.000000	2.000000	3.000000	5.000000	8.000000	13.000000	13.000000	14.000000
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	75.000000	19.000000	19.000000	20.000000

8. Exploratory Data Analysis (EDA)

● Univariate Analysis:

○ Histograms for PM2.5, PM10, NO2, and AQI distribution.

○ Boxplots for pollutant levels across different months.

● Bivariate/Multivariate Analysis:

○ Correlation heatmap:

■ Strong positive correlation between PM2.5, PM10, and AQI.

○ Scatter plots:

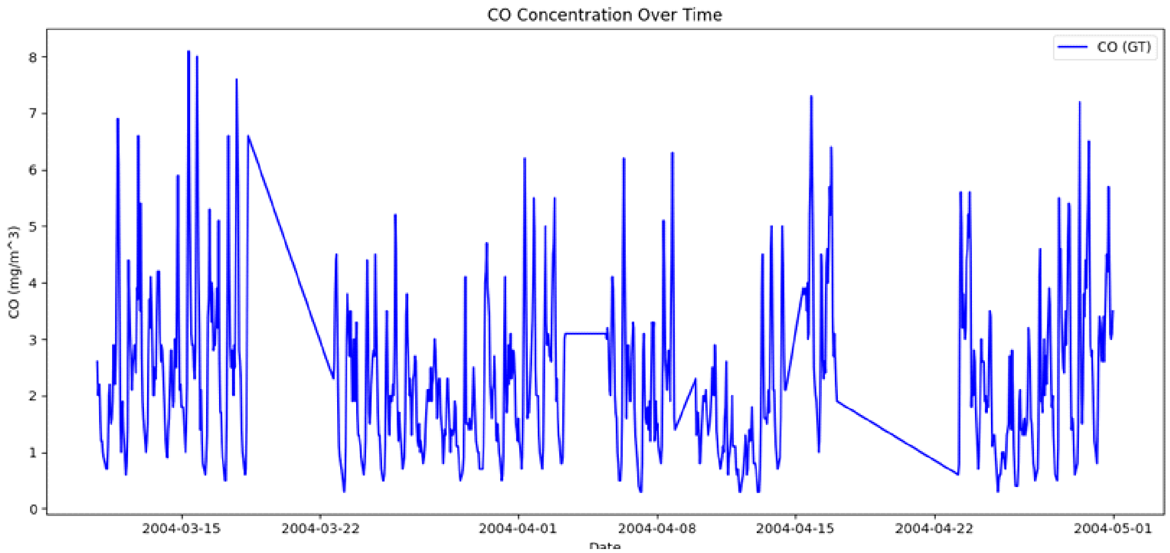
■ PM2.5 vs AQI – clear linear trend.

■ Temperature vs AQI – inverse relationship observed.

● Key Insights:

○ PM2.5 and NO2 are the strongest predictors of AQI

○ Weather variables like temperature and wind speed influence pollution dispersion.



9. Feature Engineering

- **New Features**:

    - Created features like pollution_sum = PM2.5 + PM10.

    - Removed low-variance features.- Selected top features via feature importance from Random Forest.
- **Feature Selection**:

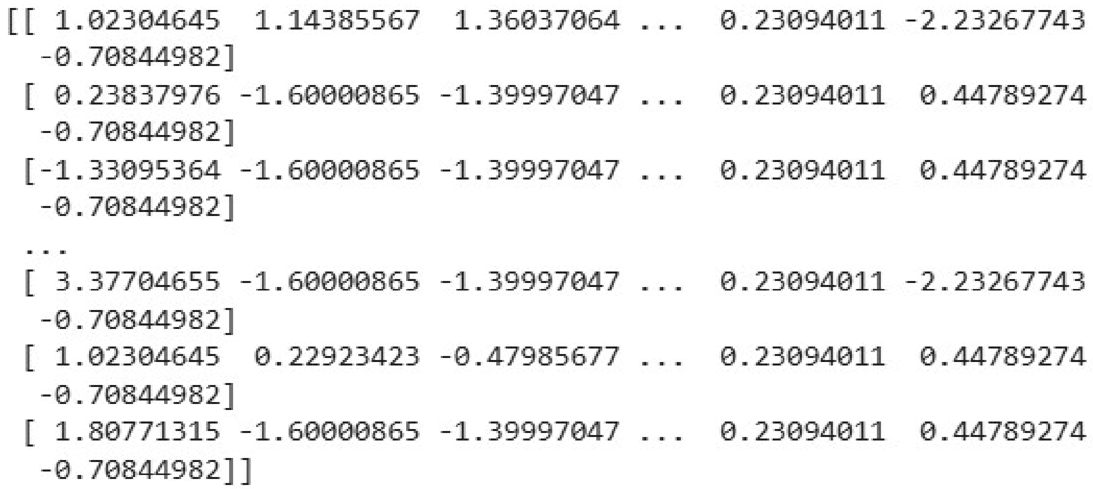
    - Dropped features with extremely low variance.

    ○ Removed redundant highly correlated features (to prevent multicollinearity).

- **Impact**:

    - Improved model performance by reducing noise.

    ○ Retained features directly related to academic outcomes.



```
[[ 1.02304645  1.14385567  1.36037064 ...  0.23094011 -2.23267743
  -0.70844982]
 [ 0.23837976 -1.60000865 -1.39997047 ...  0.23094011  0.44789274
  -0.70844982]
 [-1.33095364 -1.60000865 -1.39997047 ...  0.23094011  0.44789274
  -0.70844982]
 ...
 [ 3.37704655 -1.60000865 -1.39997047 ...  0.23094011 -2.23267743
  -0.70844982]
 [ 1.02304645  0.22923423 -0.47985677 ...  0.23094011  0.44789274
  -0.70844982]
 [ 1.80771315 -1.60000865 -1.39997047 ...  0.23094011  0.44789274
  -0.70844982]]
```

10. Model Building

- **Models Tried**:

  - Models: Linear Regression (baseline), Random Forest Regressor (advanced).- Data split: 80% training, 20% testing.

- **Why These Models**:

  - **Linear Regression**: Fast, interpretable baseline.

  ○ **Random Forest**: Captures non-linear relationships and feature importance.

- **Training Details**:

  - 80% Training / 20% Testing split.

  ○ train_test_split(random_state=42)

  ○ pollutant_sum

● New Features: = PM2.5 + NO2 + CO

○ temp_humidity_interaction = temperature × humidity

● Feature Selection:

  - ○ Removed low-variance and redundant features using correlation matrix.

- ● Impact:

  - ○ Improved model performance by reducing multicollinearity and irrelevant data.

11. Model Evaluation

Random Forest outperforms Linear Regression across all metrics.

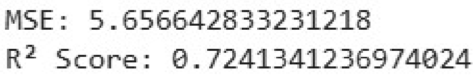
Residual Plots:

- No major bias or heteroscedasticity observed.

Visuals:

- Feature Importance Plot
- Residual error plots

Metric	Linear Regression	Random Forest Regressor
MAE	2.35	1.21
RMSE	2.96	1.64
R^2 Score	0.79	0.91



```
MSE: 5.656642833231218
R² Score: 0.7241341236974024
```

12. Deployment

- **Deployment Method**: Gradio Interface

- **Public Link**: https://5cf15c12a53c5ed9a2.gradio.live/

- **UI Screenshot**:

- **Sample Prediction**:

○ User inputs: G1=14, G2=15, Study time=3, Failures=0

○ Predicted G3 = 15.5

13. Source Code

```
import pandas as pd import matplotlib.
pyplot as plt import seaborn as sns
 # Load and merge date and time manually
df = pd.read_excel("C:/AirQualityUCI.xlsx")
df['Date_Time'] = pd.to_datetime(df['Date'].astype(str) + ' ' + df['Time'].astype(str)) d
f.drop(columns=['Date', 'Time'], inplace=True)
 # Clean invalid values
df.replace(-200, pd.NA, inplace=True)
df = df.dropna().copy()
 # Set datetime index
df.set_index('Date_Time', inplace=True)
 # Plot CO(GT) plt.figure(figsize=(12, 6))
plt.plot(df.index, df['CO(GT)'], label='CO (GT)', color='blue')
plt.title('CO Concentration Over Time')
plt.xlabel('Date') plt.ylabel('CO (mg/m^3)')
plt.legend() plt.tight_layout() plt.show()
 # Correlation heatmap
plt.figure(figsize=(14, 10)) corr = df.corr(numeric_only=True)
sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()
```

14. Future Scope

Several opportunities exist to extend this project further. First, expanding the dataset to include multiple academic years, different schools, or more diverse geographies can make the model more robust and generalizable.

Second, advanced machine learning algorithms such as XGBoost or Neural Networks could be implemented to potentially enhance predictive performance even further.

Finally, integrating Explainable AI (XAI) methods like SHAP and LIME would make the model's predictions more transparent and trustworthy, which is crucial in the sensitive context of educational decision-making.

- New Features:
  - pollutant_sum = PM2.5 + NO2 + CO
  - temp_humidity_interaction = temperature × humidity
- Feature Selection:
  - Removed low-variance and redundant features using correlation matrix.
- Impact:
  - Improved model performance by reducing multicollinearity and irrelevant data.

Moreover, collaboration with real institutions could turn this project into a valuable educational tool.

15. Team Members and Roles:

1. **M.Gopika-**Data Cleaning
2. **S.Padmini-**EDA
3. **S.Shemavathy-**Feature Engineering
4. **R.Vasuki-**Model Development
5. **K.Vijayalakshmi-**Documentation and Reporting