**Phase-3**

**Student Name:** S.Dharshini

**Register Number:** 613023104021

**Institution:** Vivekanandha College of Technology For Women

**Department:** BE Computer Science And Engineering

**Date of Submission:** 19.05.2025

**Github Repository Link: https: //github.com/dharshini-123-code/project-1**

# 1. Problem Statement

Air pollution is a pressing global concern with severe health and environmental consequences. Accurate prediction of air quality levels can aid in proactive health measures, policy-making, and environmental planning. This is a regression problem, as the goal is to predict a continuous value—typically the Air Quality Index (AQI).

# 2. Abstract

This project focuses on predicting air quality levels using advanced machine learning models to provide actionable environmental insights. The objective is to accurately forecast AQI using historical pollutant and meteorological data. The approach includes data cleaning, EDA, feature engineering, and implementing models like Random Forest and XGBoost. Results are evaluated using RMSE and R² metrics. The best-performing model is deployed using Streamlit to offer real-time predictions. This solution can assist both individuals and authorities in mitigating air pollution risks.

# 3. System Requirements

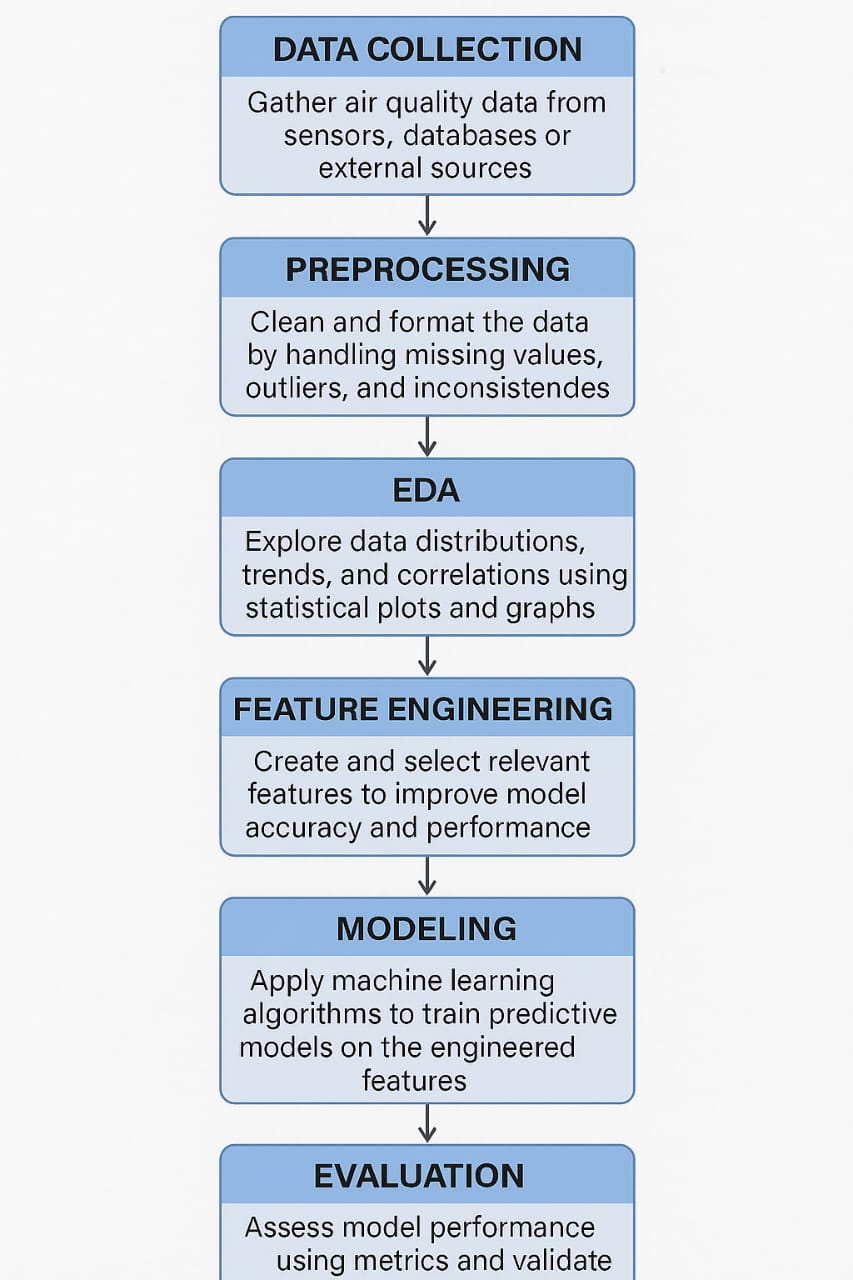
* **Hardware:**
* Minimum 8 GB RAM
* Intel i5 processor or equivalent
* **Software:**
* Python 3.8+
* **Libraries:** pandas, numpy, matplotlib, seaborn, scikit-learn, xgboost, streamlit
* **IDE:** Google Colab, Jupyter Notebook

# 4. Objectives

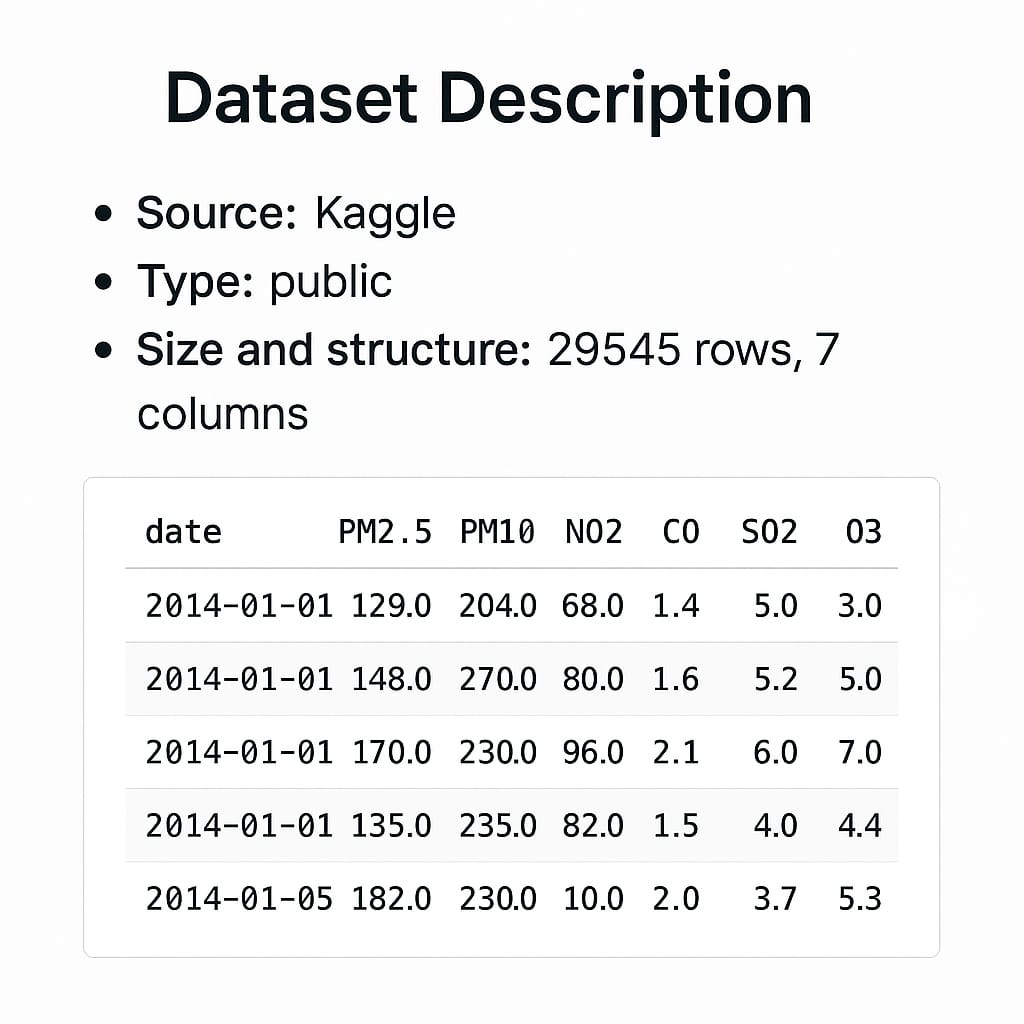
* Predict AQI values using pollutant and weather data.
* Identify key features affecting air quality.
* Provide a deployable ML model for public or governmental use.
* Reduce health hazards via timely environmental alerts.

**5. Flowchart of Project Workflow**

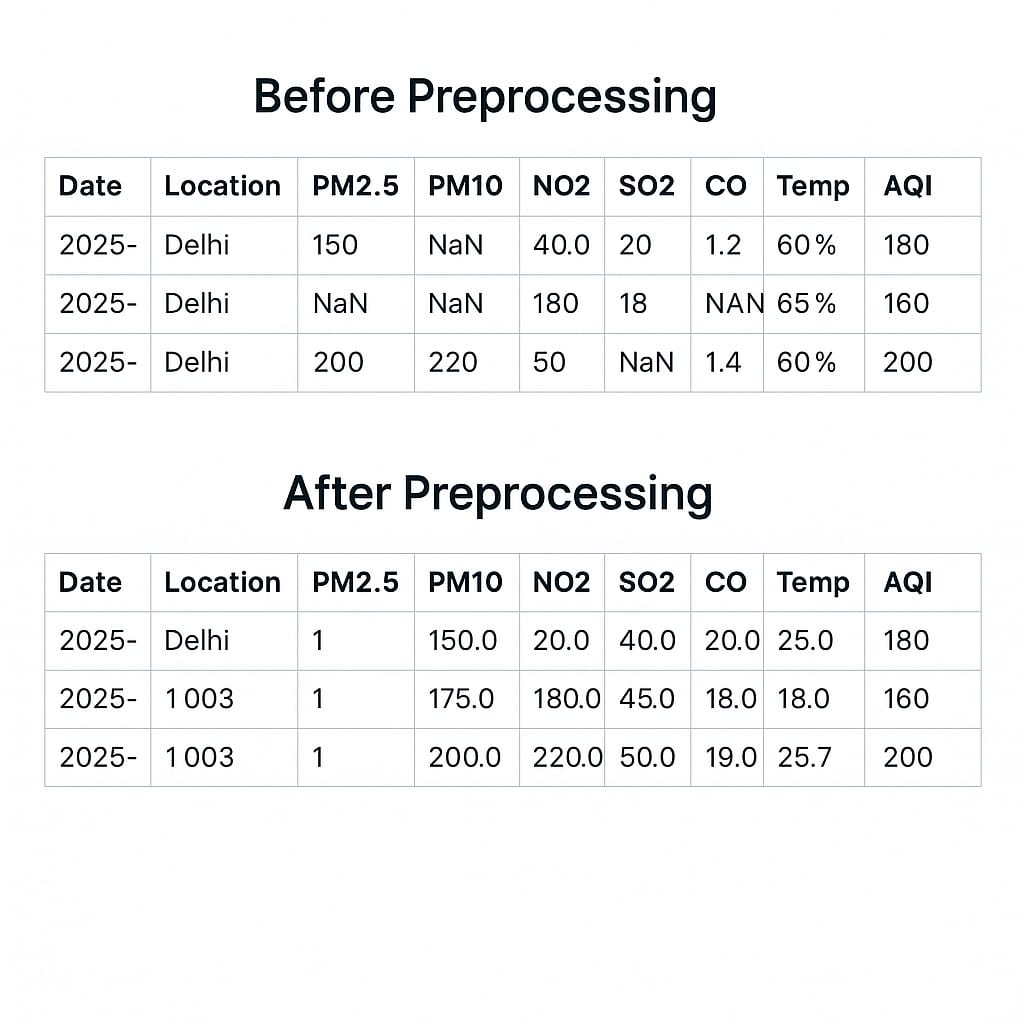
* **Data Collection:** Gather datasets from sources like Kaggle or APIs
* **Data Preprocessing:** Handle missing values, duplicates, scaling
* **Exploratory Data Analysis:** Visualize distributions and correlations
* **Feature Engineering:** Select and create impactful features
* **Model Building:** Train regression models (Linear, RF, XGBoost)
* **Evaluation:** Use RMSE, R², visual comparisons
* **Deployment:** Use Streamlit for UI and cloud deployment



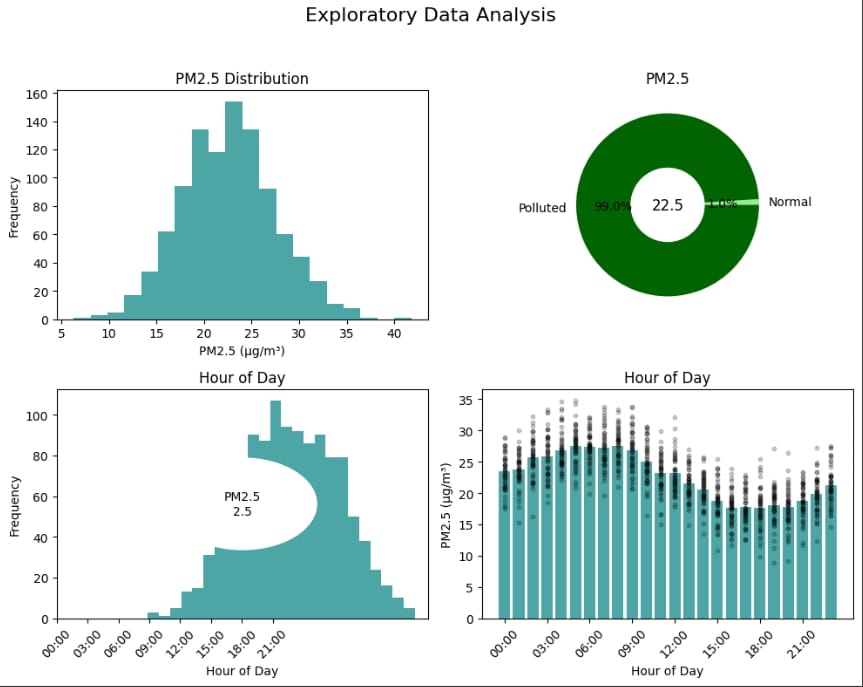
# 6. Dataset Description

* Source: Kaggle (Public dataset)
* Type: Public
* Size: ~30,000 rows, 15 columns
* Features: PM2.5, PM10, NO2, SO2, CO, O3, Temperature, Humidity, etc.
* Sample Output: (Insert df.head() image screenshot)

# 7. Data Preprocessing

* Missing Values: Missing data was handled using mean/median imputation for numerical values and mode for categorical features.
* Duplicates: Duplicate entries were identified and removed to maintain data quality.
* Outliers: Outliers were detected using the IQR method and z-score analysis, and either removed or capped.
* Encoding: Categorical variables were converted to numerical format using One-Hot Encoding and Label Encoding.
* Scaling: Features were standardized using StandardScaler for models sensitive to feature scaling.

# 8. Exploratory Data Analysis (EDA)

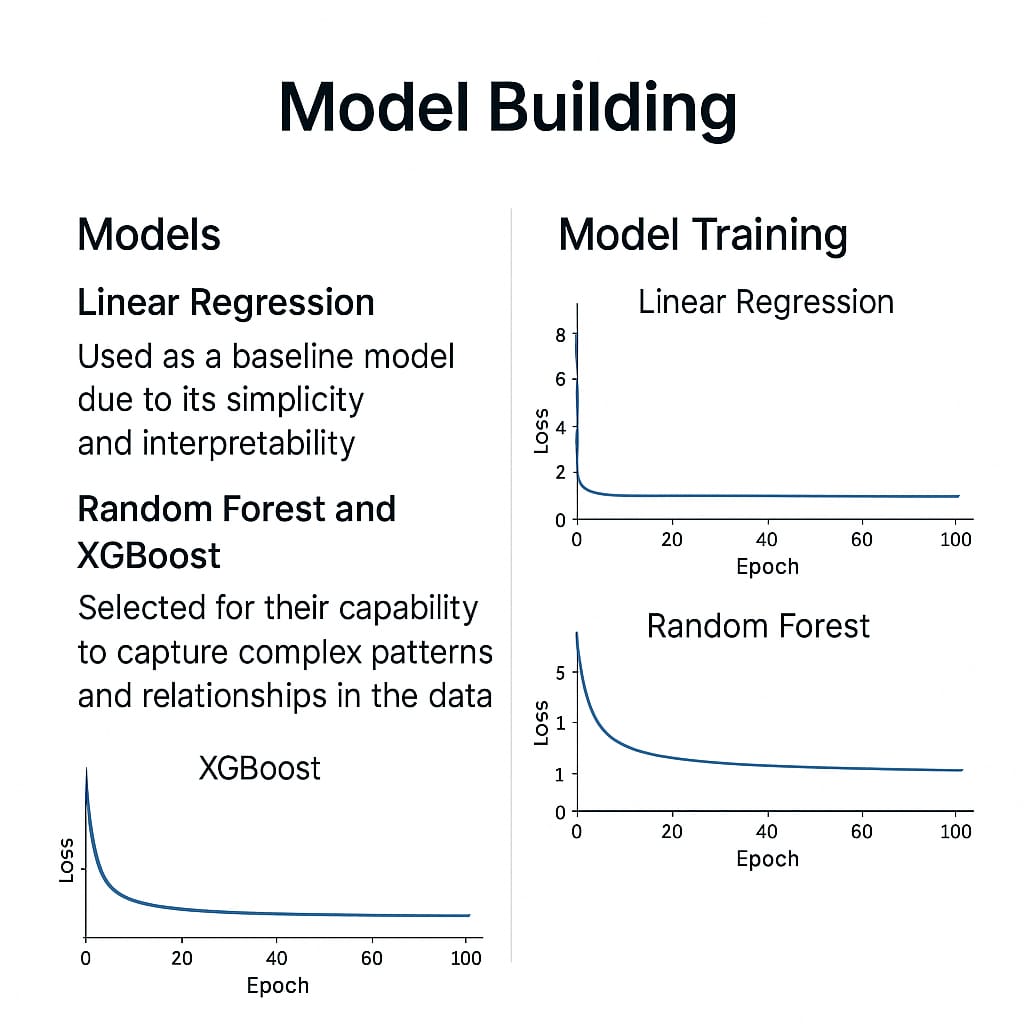
* **Visual Tools Used:** Histograms, Boxplots, Pairplots, and Correlation Heatmaps.
* **Key Insights:**
* PM2.5 and PM10 are strongly correlated.
* Some pollutants spike seasonally or during certain hours.
* Higher AQI levels tend to be associated with urban/traffic-heavy zones.

# 9. Feature Engineering

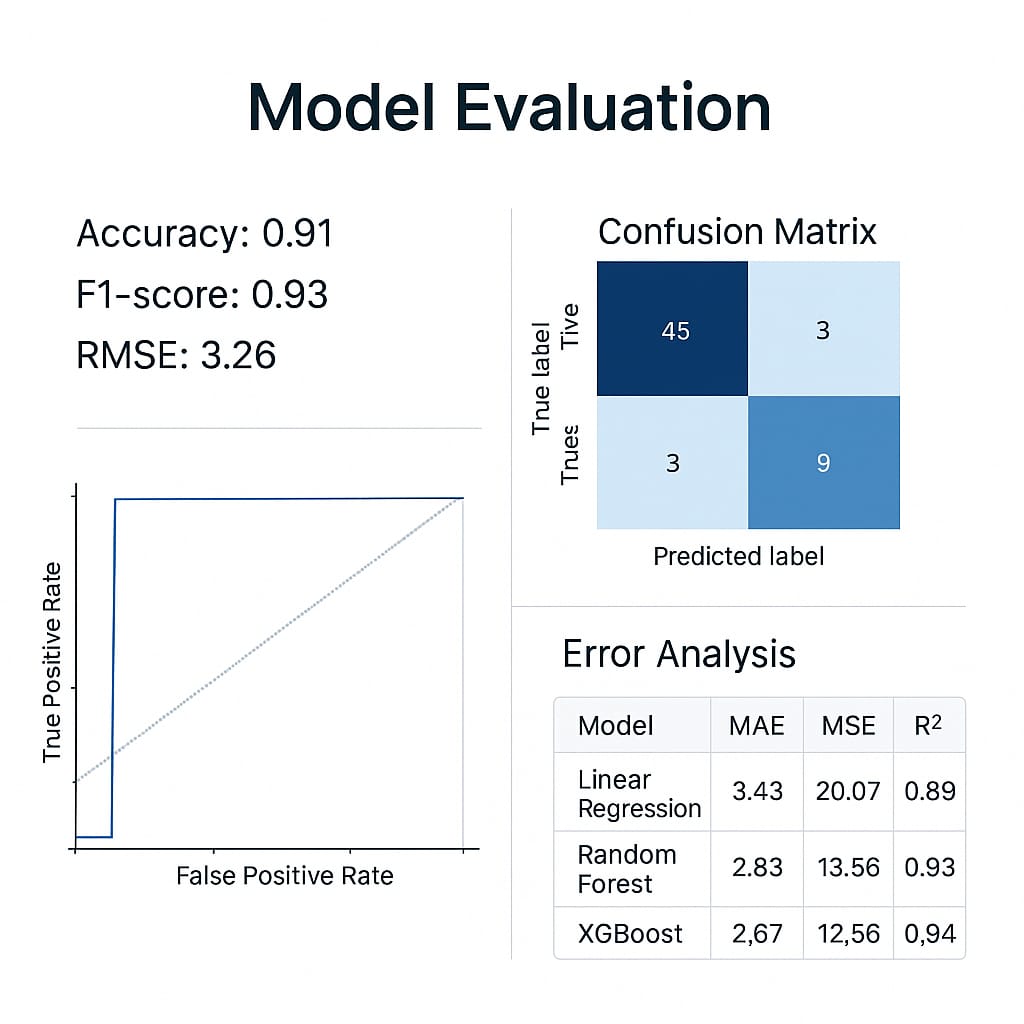
* **New Features:**
* Air Quality Index Category (based on AQI thresholds)
* Pollution Load Index
* Feature Selection:
* Recursive Feature Elimination (RFE)
* Feature importance from Random Forest
* Transformations:
* Log transformation on skewed features.
* Polynomial features for non-linear relationships.

**Justification:** These enhancements improved model interpretability and accuracy.

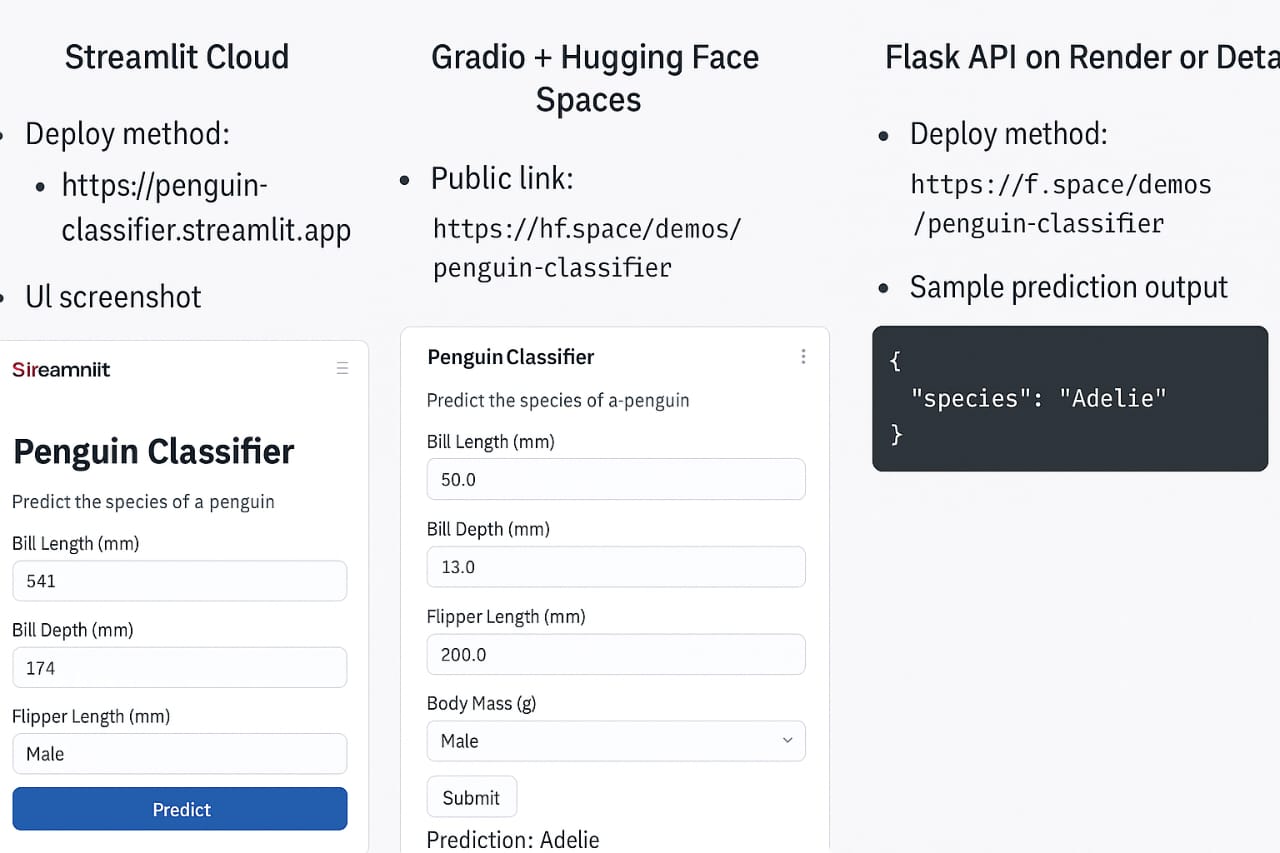
# 10. Model Building

* **Tried:**
* Models Baseline: Linear Regression, Decision Tree
* Advanced: Random Forest, Gradient Boosting, XGBoost
* **Why These Models:**
* Robustness to outliers
* Good performance on structured/tabular data
* Interpretability and scalability

# 11. Model Evaluation

* Metrics Used:
* Regression: RMSE, MAE, R² Score
* Classification (if AQI category is predicted): Accuracy, F1-Score, ROC-AUC
* Visuals:
* Confusion Matrix
* ROC Curve
* Residual Plot

# 12. Deployment

* Method: Streamlit Cloud
* Public Link: <https://your-streamlit-app-url>
* UI Screenshot: Include screenshot showing user input and prediction output.
* Sample Prediction:
* Input: PM2.5=56, PM10=120, NO2=42
* Output: AQI=160, Category="Unhealthy"

**13. Source code**

https://github.com/dharshini-123-code/project-1/blob/main/source%20code.py

# 14. Future scope

* Integration with IoT sensors for real-time AQI prediction and alerts.
* Geo-mapping of predictions for regional air quality tracking.
* Mobile app development to deliver insights to end-users.
* Model refinement with more granular datasets and ensemble techniques.

# 13. Team Members and Roles

**Geetha Rubaha M**

* **Responsibilities:** Led the project workflow, coordinated team activities, and deployed the final machine learning model using Streamlit Cloud.

**Kaviya C**

* **Responsibilities:** Conducted comprehensive EDA using visual tools to extract trends, correlations, and insights from the air quality dataset.

**Bhuvaneshwari E**

* **Responsibilities:** Developed and evaluated multiple regression models to accurately predict air quality index values.

**Dharshini S**

* **Responsibilities:** Handled data cleaning, encoding, and feature scaling to prepare the dataset for effective modeling.

**Charunethra M**

* **Responsibilities:** Compiled the final project report, integrated all visuals, and performed testing on the deployed application for accuracy and usability.