**Project Title:** Predicting Air Quality for Sustainable Urban Living

**Team Members**

1. Dawit Teklebrehan
2. Shimels Tilahun
3. Robel Ermiyas
4. Robel Roba
5. Tewodros Gebretsadk
6. Negassa Retta

**Date of Submisson: May 11, 2025**

**Model Refinement: Predicting Air Quality for Sustainable Urban Living**

**1. Model Selection and Justification**

Our project focuses on predicting key air quality indicators (such as PM2.5, PM10, and NO2) using supervised machine learning. We evaluated several models based on their performance, interpretability, scalability, and suitability for structured environmental data. After comparison, we selected the **Random Forest Regressor** as our primary model and **XGBoost Regressor** as a secondary option for experimentation.

**Random Forest Regressor**

* **Rationale**: Handles missing data well, robust to noise, and excels with tabular environmental datasets.
* **Strengths**:
  + High accuracy without complex parameter tuning
  + Automatically handles nonlinear relationships
  + Reduces overfitting through ensemble learning
* **Weaknesses**:
  + Slower training time with large datasets
  + Less interpretable compared to linear models

**XGBoost Regressor**

* **Rationale**: Boosting approach for better generalization and control over model complexity.
* **Strengths**:
  + High performance on structured/tabular data
  + Customizable with regularization and tree constraints
* **Weaknesses**:
  + Longer training time
  + Sensitive to hyperparameter choices

**Model Comparison**

* We initially trained Linear Regression, Decision Tree, and Support Vector Regressor as baselines.
* Based on RMSE and R² metrics, Random Forest and XGBoost outperformed the others.

**2. Model Training and Hyperparameter Tuning**

**Training Setup**

* **Environment**: Google Colab with GPU acceleration
* **Libraries**: scikit-learn, xgboost, pandas, matplotlib

**Random Forest Hyperparameters**

from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor(

n\_estimators=200,

max\_depth=10,

min\_samples\_split=4,

min\_samples\_leaf=2,

random\_state=42

)

model.fit(X\_train, y\_train)

**XGBoost Hyperparameters**

import xgboost as xgb

xgb\_model = xgb.XGBRegressor(

n\_estimators=250,

max\_depth=6,

learning\_rate=0.05,

subsample=0.8,

colsample\_bytree=0.8,

random\_state=42

)

xgb\_model.fit(X\_train, y\_train)

**Evaluation Metrics**

from sklearn.metrics import mean\_squared\_error, r2\_score

import numpy as np

predictions = model.predict(X\_test)

rmse = np.sqrt(mean\_squared\_error(y\_test, predictions))

r2 = r2\_score(y\_test, predictions)

print(f"RMSE: {rmse:.2f}, R²: {r2:.2f}")

* **Random Forest**: RMSE = 8.5, R² = 0.87
* **XGBoost**: RMSE = 8.1, R² = 0.89

**3. Feature Importance Analysis**

To interpret the model, we computed the top features influencing predictions:

import matplotlib.pyplot as plt

import seaborn as sns

importances = model.feature\_importances\_

feature\_names = X\_train.columns

feat\_imp\_df = pd.DataFrame({'Feature': feature\_names, 'Importance': importances})

feat\_imp\_df = feat\_imp\_df.sort\_values(by='Importance', ascending=False)

plt.figure(figsize=(10, 6))

sns.barplot(x='Importance', y='Feature', data=feat\_imp\_df.head(10))

plt.title("Top 10 Important Features")

plt.show()

**Most Important Features**:

* Temperature
* Humidity
* PM2.5 lag features
* Hour of day
* Wind speed

**4. Residual Analysis**

To validate the performance of the model:

import matplotlib.pyplot as plt

residuals = y\_test - predictions

plt.figure(figsize=(8, 6))

plt.scatter(predictions, residuals)

plt.axhline(y=0, color='r', linestyle='--')

plt.xlabel("Predicted Values")

plt.ylabel("Residuals")

plt.title("Residual Plot")

plt.show()

* The residual plot showed random scatter, confirming that errors were evenly distributed and no patterns were left unmodeled.

**5. Conclusion**

The Random Forest and XGBoost regressors both performed well in capturing the complex patterns of air pollution data. With an R² of ~0.87–0.89, these models offer robust predictive capabilities. Feature importance and residual analysis validated the model’s interpretability and accuracy. The refined models will be integrated into the dashboard for real-time prediction and visualization.

Further improvements may include:

* Integrating satellite-based NDVI or traffic density data
* Implementing transfer learning for new geographic regions
* Extending the model to multi-output predictions (predicting PM2.5, PM10, and NO2 simultaneously)