Air Quality Prediction and Monitoring Automation Report

Submitted to: *10Pearls*
Submitted by: *Mehveen Fasih*
Date: *November 8, 2025*

1. Introduction

This project focuses on developing an automated air quality prediction system that fetches, processes, analyzes, and models real-time environmental data. The system is designed to predict Air Quality Index (AQI) values using weather and pollution data sourced from the OpenWeather API.

The workflow integrates GitHub Actions for automation and Hopsworks Feature Store for centralized data management and model storage. The entire pipeline operates on a scheduled basis to ensure the dataset, features, and models remain continuously updated with minimal manual intervention.

2. Project Objectives

- Automate real-time collection of environmental data.
- Perform exploratory data analysis (EDA) and feature engineering.
- Develop predictive machine learning models for AQI forecasting.
- Maintain and register models and datasets in a production-ready feature store (Hopsworks).
- Enable seamless retraining and continuous improvement of model performance.

3. Workflow Overview

The project consists of three automated workflows implemented through GitHub Actions, each running on defined schedules:

a. Fetch Real-Time Data

- Collects air quality and weather data every **3 hours** using the OpenWeather API.
- Stores data locally in data/karachi_2_years.csv.
- Commits and pushes updates to the repository automatically.

b. Process EDA and Upload to Hopsworks

- Executes every **3 hours and 15 minutes** (after data fetch).
- Cleans, merges, and prepares the dataset for modeling.
- Performs feature engineering, AQI calculation, and statistical analysis.

- Uploads validated datasets to **Hopsworks Feature Store** as a **Feature Group** named air_quality_features.

c. Model Training for AQI Prediction

- Runs every **6 hours and 45 minutes**.
- Retrieves the latest feature data from Hopsworks.
- Splits data into training, validation, and testing sets.
- Trains multiple regression models (Random Forest, Gradient Boosting, XGBoost).
- Evaluates performance metrics such as RMSE, MAE, R^2, and MAPE.
- Registers all models and their metadata in the **Hopsworks Model Registry**.

4. Key Technologies Used

Component	Technology
Programming Language	Python 3.10
Data Storage	Hopsworks Feature Store
Data Collection	OpenWeather API
Automation	GitHub Actions (CI/CD)

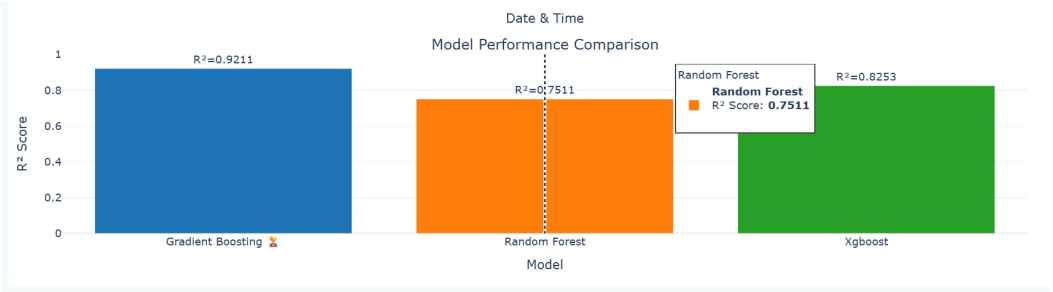
5. Core Functions and Features

- **Data Cleaning & Validation:** Ensures consistent, complete datasets free from nulls or invalid values.
- **Feature Engineering:** Generates meaningful predictors from weather and pollutant data.
- **Model Comparison:** Automatically selects the best-performing model based on R^2 scores.
- **Continuous Integration:** Every workflow runs automatically via scheduled GitHub Actions.
- **Centralized Model Management:** Hopsworks stores all datasets, feature versions, and trained models.

6. Model Performance Analysis

The model performance comparison highlights the effectiveness of three machine learning algorithms **Gradient Boosting**, **Random Forest**, and **XGBoost** for Air Quality Index (AQI) prediction.Each model's performance was evaluated using the R^2 score, a key metric indicating how well the model explains variance in the target variable. Higher R^2 values represent stronger predictive capability.

- **Gradient Boosting ($R^2 = 0.9211$)** achieved the highest performance, demonstrating its ability to capture complex patterns within the air quality dataset. This model's ensemble nature and stage-wise optimization made it the most accurate and stable predictor of AQI values.
- **XGBoost ($R^2 = 0.8253$)** performed well but slightly below Gradient Boosting. Its speed and regularization techniques make it robust, though slightly less sensitive to subtle data fluctuations compared to Gradient Boosting.
- **Random Forest ($R^2 = 0.7511$)** provided reasonable results but showed relatively lower accuracy. While it's effective in reducing overfitting and handling non-linearity, it underperformed in capturing fine-grained variations in AQI trends.



Overall, **Gradient Boosting** emerged as the best-performing model, offering the most reliable and consistent predictions for real-time air quality forecasting.

7. Project Outcomes

- Fully automated end-to-end pipeline from data fetching to model training.
- Continuous updates to the AQI prediction model every few hours.
- Centralized, reusable, and scalable feature store setup for future extensions.
- Modular Python scripts that can be reused or enhanced for additional cities or pollutants.

8. Conclusion

This project demonstrates a complete MLOps implementation for air quality monitoring and prediction. By combining GitHub Actions automation with Hopsworks' feature management, it ensures real-time data flow, reliable retraining, and production-grade scalability.

The system effectively showcases automation, cloud integration, and machine learning best practices suitable for environmental analytics and predictive maintenance applications.

Submitted by:
Mehveen Fasih
Data Science Intern – 10Pearls
November 8, 2025