Karachi Air Quality Index (AQI) Prediction System

Muhammad Tahir

February 13, 2026

Abstract

This report details the development and deployment of an automated machine learning system for real-time Air Quality Index (AQI) prediction in Karachi, Pakistan. The system utilizes a serverless architecture featuring Hopsworks Feature Store, GitHub Actions for automation, and Streamlit Cloud for interactive visualization.

Contents

1 Introduction **3**

2 System Architecture **3**

3 MLOps and Automation **3**

4 Data Analysis and Feature Engineering **3**
4.1 Pollutant Distributions 4
4.2 Correlation Analysis 4

5 Model Development and Benchmarking **4**

6 Production Infrastructure and Dashboard **5**

7 Explainable AI (XAI) and Model Interpretability **8**

8 Challenges Faced and Solutions Implemented **8**
8.1 Feature Store and Data Engineering 8
8.2 Pipeline Development and Model Training 9
8.3 Cloud Deployment and Production 9

9 Conclusion **9**

1 Introduction

Air pollution is a critical health concern in Karachi. This project aims to provide accurate, real-time AQI forecasts using historical pollutant data and meteorological features. The objective is to build a self-improving pipeline that automates data ingestion, model training, and inference.

2 System Architecture

The system follows a modular MLOps architecture, ensuring that the model stays relevant and data stays fresh without manual intervention.

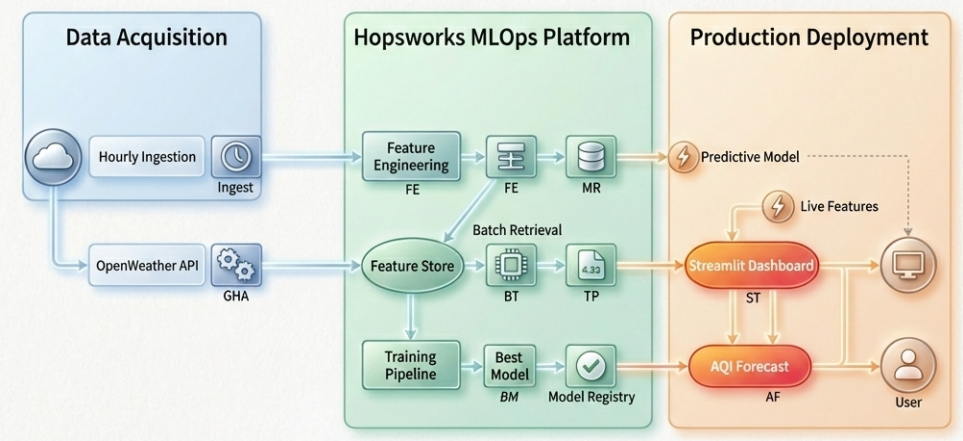

Figure 1: End-to-End System Architecture: Cloud Ingestion to Dashboard

3 MLOps and Automation

In industrial settings, the value of a machine learning model is tied to its reliability and freshness. This project implements two core MLOps pipelines:

- **Continuous Ingestion (CI):** Managed by GitHub Actions, this pipeline triggers every hour to sync new ground-truth data from the OpenWeather API. This represents the *Feature Pipeline* in a standard MLOps architecture.
- **Continuous Training (CT):** A daily scheduled workflow that re-validates the model performance against the most recent data stored in the Hopsworks Feature Store.

4 Data Analysis and Feature Engineering

Exploratory Data Analysis (EDA) revealed significant temporal patterns in Karachi's air quality.

4.1 Pollutant Distributions

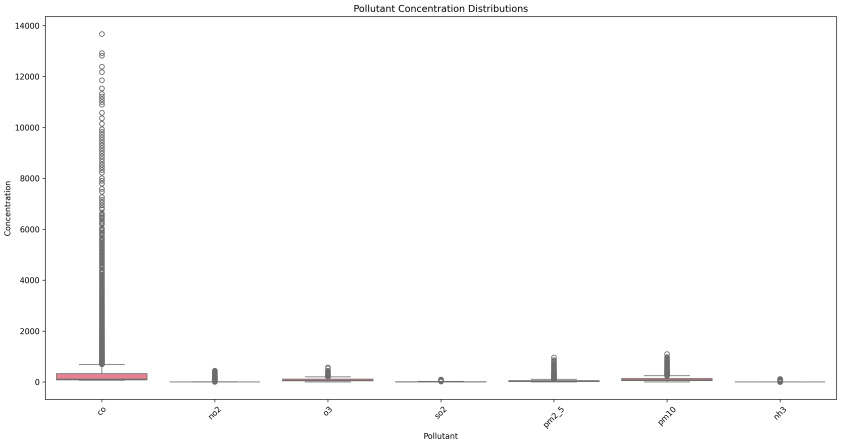

Figure 2: Distribution of various pollutants in Karachi

4.2 Correlation Analysis

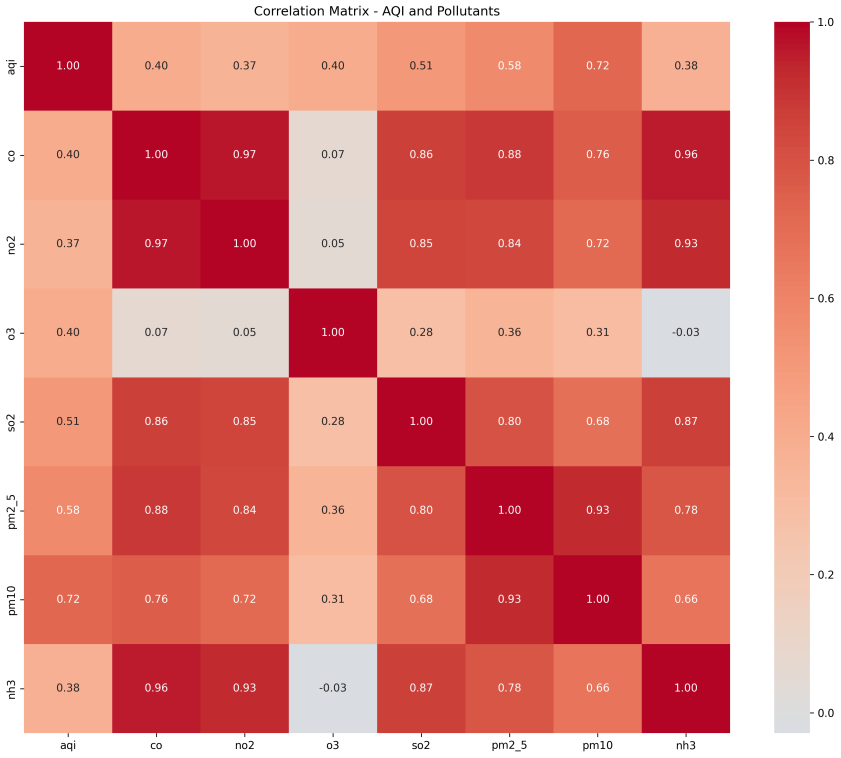

Figure 3: Correlation Heatmap of environmental variables

5 Model Development and Benchmarking

We performed rigorous benchmarking across multiple regression architectures to ensure the most robust deployment.

The **Gradient Boosting** regressor was selected as our production model due to its superior RMSE (0.0393) and R2 (0.9986), indicating it captures nearly all variance in the air quality distributions of Karachi.

Table 1: Model Performance Comparison

Model	Train MAE	Val MAE	Train RMSE	Val RMSE	Train R^2	Val R^2
Ridge	0.0332	0.0349	0.0763	0.1012	0.9949	0.9906
Random Forest	0.0014	0.0031	0.0176	0.0461	0.9997	0.9981
Gradient Boosting	0.0011	0.0027	0.0052	0.0393	1.0000	0.9986

6 Production Infrastructure and Dashboard

The system is deployed as a fully functional production application. This section showcases the interactive capabilities of the dashboard.

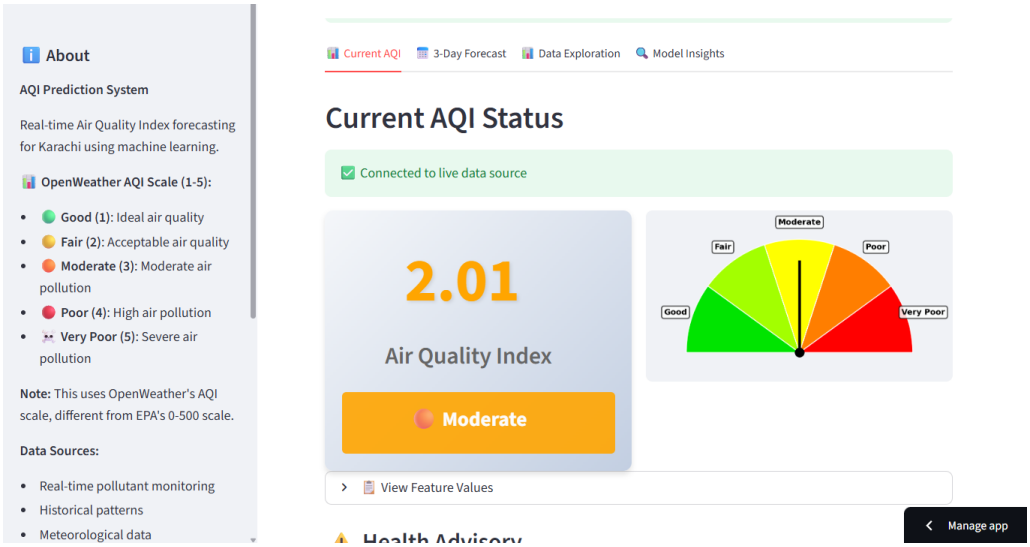

Figure 4: Production Environment: Real-time AQI Prediction and Live Status

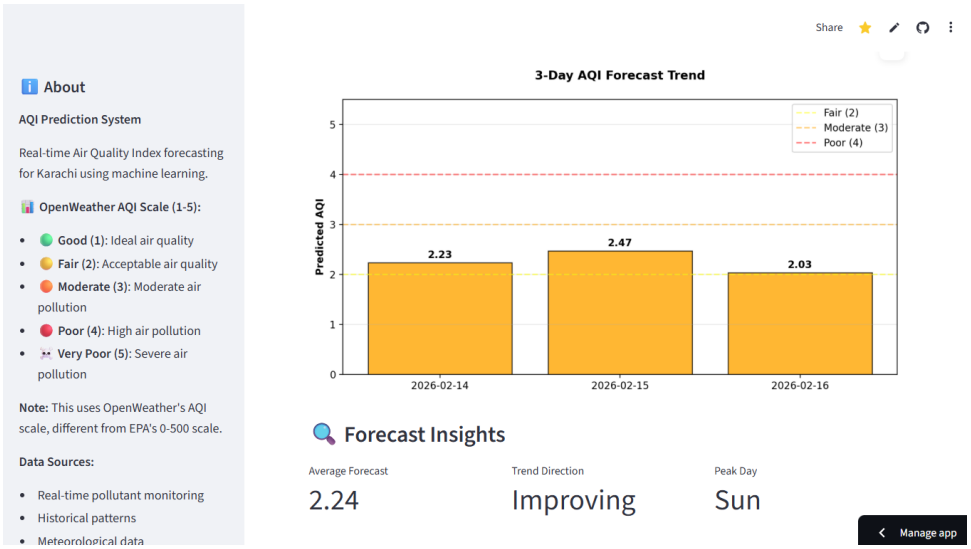

Figure 5: 3-Day Predictive Forecast

Figure 6: Pollutant Concentration Metrics

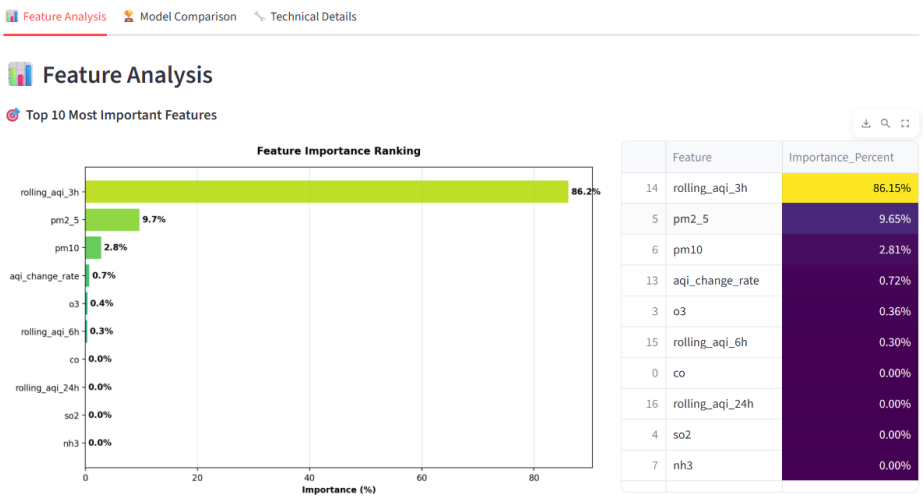

Figure 7: XAI and Model Interpretability

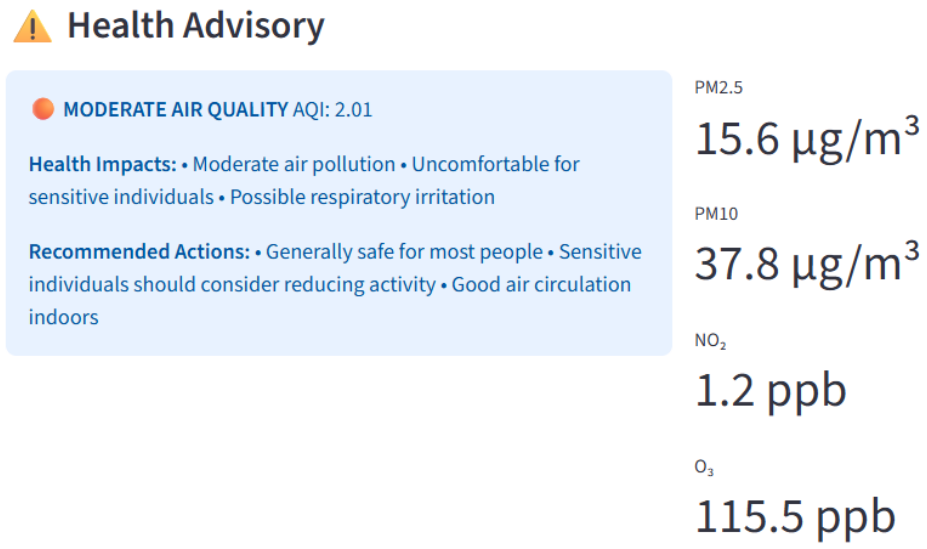

Figure 8: Smart Health Advisories

7 Explainable AI (XAI) and Model Interpretability

Industrial machine learning requires trust. We utilize SHAP values to provide global and local explanations for every prediction, ensuring health advisories are based on transparent pollutant drivers.

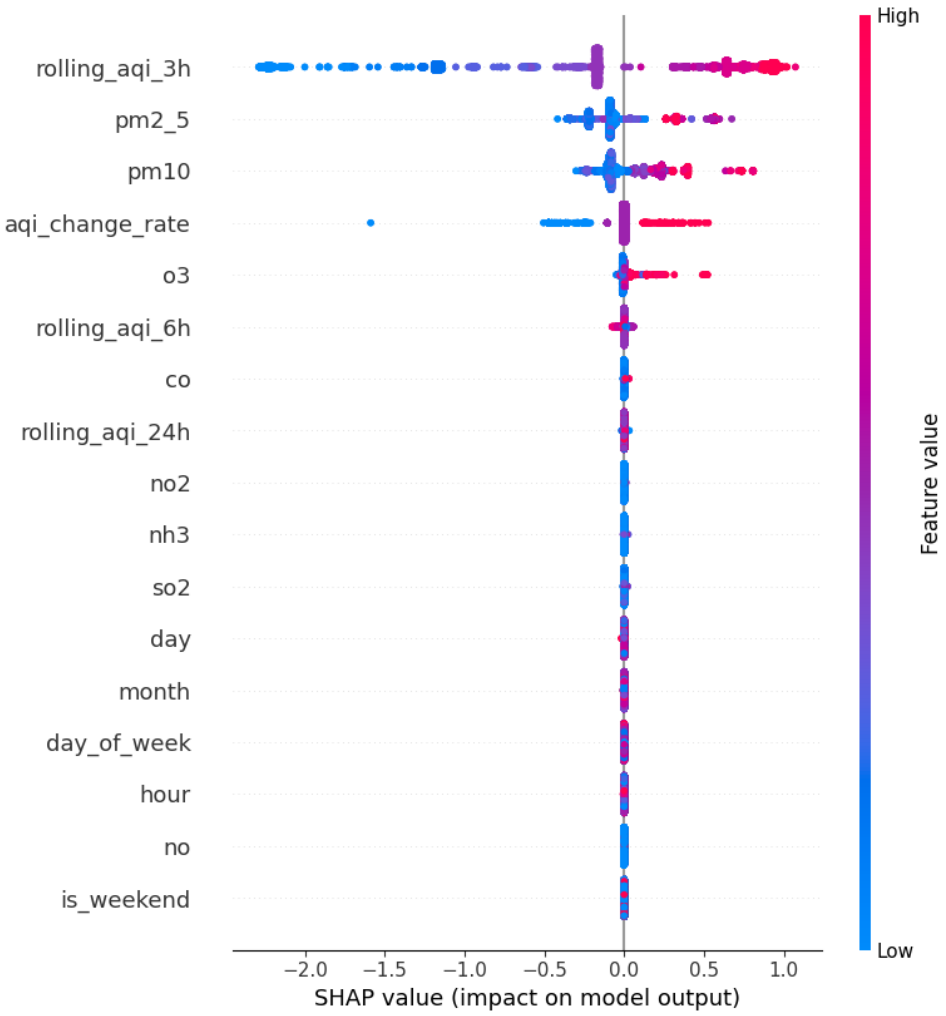

Figure 9: SHAP Summary Plot showing feature importance

8 Challenges Faced and Solutions Implemented

Building an end-to-end MLOps system involves navigating various technical hurdles. Below is a summary of the primary challenges encountered during development and the strategies used to resolve them.

8.1 Feature Store and Data Engineering

Hopsworks Schema Mismatches: Encountered data type conflicts (e.g., *bigint* vs *double*) between the OpenWeather API response and the Hopsworks Feature Group. Resolved via explicit type casting using Pandas `astype('int64')`.

Schema Corruption: A critical late-stage issue occurred where the `datetime` column name was stored incorrectly in the Feature Group metadata, breaking standard queries. A workaround was implemented to read data directly from the Feature Group and manually filter by `event_id`.

Validation Hurdles: Initial integration with Great Expectations lead to validation failures due to rigid regex patterns for IDs. This was solved by standardizing the `event_id` format to `dt.strftime("%Y%m%d%H")`.

8.2 Pipeline Development and Model Training

Temporal Leakage: The `datetime` and `event_id` columns were initially included in the feature set, causing potential leakage. These were explicitly dropped before training to ensure the model learns from pollutants and meteorological data only.

Dataset Missing Errors: Errors occurred when attempting to retrieve training data before the Feature View was fully instantiated. Re-implemented the logic to ensure `create_training_dataset=True` was correctly flagged during the first run.

8.3 Cloud Deployment and Production

SDK Versioning Conflict: A significant blocker appeared when Streamlit Cloud's default Hopsworks SDK (4.6.4) enforced stricter schema validation than the local version (4.2.10). Resolved by pinning the exact working version in `requirements.txt`.

UI and Interpretation: Discrepancies were found between the model's 1-5 AQI scale (OpenWeather) and standard US-EPA scales. The dashboard was updated with clear documentation and a "Dashboard Notice" to ensure user trust and clarity.

Visualization Errors: Advanced XAI libraries like SHAP encountered version compatibility issues in the cloud environment. Fallback mechanisms using native model `feature_importances_` were implemented to maintain observability.

9 Conclusion

The system successfully provides real-time AQI predictions with high accuracy. The automated pipelines ensure that the model adapts to seasonal variations without manual intervention.