

AQI FORECASTER

Automated MLOps Pipeline for Air Quality Index

*A Deep Dive into Recursive Forecasting and Serverless
Architecture*

Shahzain Adil

Contents

1	Executive Summary	2
1.1	System Architecture	2
2	The Engineering Lifecycle	3
2.1	Feature Pipeline & Lag Engineering	3
2.2	Automated CI/CD Workflows	3
3	Challenges & Debugging	4
3.1	The Schema Desynchronization Crisis	4
3.2	Recursive Inference Logic	4
4	Performance Analysis	5
5	Conclusion	6

Executive Summary

Project Objective

To engineer a serverless, self-healing machine learning pipeline that predicts PM2.5 concentrations with high temporal accuracy, bridging the gap between raw environmental sensors and actionable public health data.

1.1 System Architecture

The system utilizes a decentralized architecture. By moving from a centralized feature store to a distributed **MongoDB Atlas** instance, the platform achieved a 75% reduction in cold-start latency for the inference engine.

The Engineering Lifecycle

2.1 Feature Pipeline & Lag Engineering

The core innovation of the model is the integration of **Temporal Autocorrelation**. Unlike standard regression models that treat each hour as an independent event, this pipeline treats air quality as a continuous physical process.

- **Lag-1 Feature:** Using $PM_{2.5_{t-1}}$ provides the model with the "environmental baseline."
- **Weather Modulation:** Wind speed and humidity are treated as "perturbation factors" that adjust the baseline.

2.2 Automated CI/CD Workflows

Automation is handled via GitHub Actions, which orchestrates two decoupled lifecycles:

1. **Hourly Ingestion:** Synchronizes real-time data from Open-Meteo and WAQI APIs.
2. **Nightly Training:** Re-evaluates model weights based on the last 24 hours of fresh data.

Challenges & Debugging

3.1 The Schema Desynchronization Crisis

A critical hurdle involved a naming mismatch where the backend provided `pm2.5` but the database expected `pm2_5`.

- **Symptom:** Empty dataframes during the "Recursive Loop" inference.
- **Resolution:** Implementation of a robust data-cleaning layer (Sanitization Layer) that standardizes all API payloads before they reach the Model Registry.

3.2 Recursive Inference Logic

To forecast 72 hours out, the system uses its own previous predictions as inputs for the next time step. This creates a "feedback loop" that mimics real-world atmospheric dispersion.

$$P_{t+n} = f(W_{t+n}, P_{t+n-1}) \quad (3.1)$$

Performance Analysis

The "Model Tournament" validates that a simpler **Linear Regression** model, when fueled by high-quality lag features, outperforms complex ensembles like Random Forest in terms of generalization on unseen data.

Model Strategy	RMSE	MAE	R ² Score
Linear Regression	4.85	3.22	85.01%
Random Forest	5.36	3.57	81.70%
Gradient Boosting	4.92	3.41	84.56%

Conclusion

This project successfully demonstrates that MLOps is not just about the "best" model, but about the reliability of the data pipeline. The resulting dashboard provides a transparent, scientifically grounded forecast of Karachi's air quality.