Air Quality Index (AQI) Forecasting System

End-to-End Machine Learning Pipeline

Submitted by

Artasam Bin Rashid

Data Science Intern

Organization

10Pearls Pakistan

Date

January 22, 2025

1 Executive Summary

This report documents the design and implementation of an end-to-end **Air Quality Index (AQI) Forecasting System** developed during my internship at **10Pearls Pakistan**. The primary objective of the project was to build a production-ready machine learning pipeline capable of forecasting short-term air quality trends for Rawalpindi using real-time environmental data.

The system automates the complete workflow, including data ingestion, feature engineering, model training, evaluation, deployment, and visualization. The final model achieved an excellent performance with an R^2 score of **0.9993** and an RMSE of **0.91**. A Streamlit-based dashboard presents accurate, interpretable, and actionable AQI forecasts to end users.

2 Project Overview

2.1 Objective

The key objectives of this project were:

- Automate real-time AQI and weather data collection from public APIs.
- Engineer robust features and compute AQI values using EPA standards.
- Train, evaluate, and compare multiple machine learning models.
- Deploy the best-performing model via an interactive web dashboard.

2.2 Problem Statement

Air pollution poses a serious public health challenge in major cities such as Rawalpindi. While most existing platforms report current air quality conditions, they rarely provide reliable short-term forecasts. This project addresses that gap by enabling proactive AQI prediction, supporting early warnings and informed decision-making.

3 Methodology and Implementation

3.1 System Architecture

The system follows a modular and scalable architecture consisting of the following components:

1. **Data Ingestion**: Hourly pollutant and weather data collection using Open-Meteo APIs.

2. **Feature Engineering**: AQI computation, lag features, rolling statistics, time-based features, missing value handling, and normalization.

3. **Feature Store**: Centralized storage and retrieval using the Hopsworks Feature Store.

4. **Model Training**: Automated training and evaluation using multiple regression models.

5. **Deployment**: Real-time inference and visualization through a Streamlit dashboard.

4 Data and Feature Engineering

The dataset consists of hourly measurements of major pollutants including $\mathrm{PM}_{2.5}$, PM_{10}, CO, NO_2, SO_2, and O_3, along with meteorological variables such as temperature, humidity, wind speed, and atmospheric pressure.

4.1 AQI Computation

AQI values are calculated using the U.S. EPA standard formula:

$$\mathrm{AQI} = \frac{I_{\mathrm{high}} - I_{\mathrm{low}}}{C_{\mathrm{high}} - C_{\mathrm{low}}}(C - C_{\mathrm{low}}) + I_{\mathrm{low}}$$

where C represents the pollutant concentration. The final AQI for each timestamp is defined as the maximum AQI across all pollutants.

4.2 Derived Features

- **Time-based Features**: Timestamps are converted to Pakistan Standard Time (Asia/Karachi). Calendar attributes (hour, day of week, month) are encoded using sine and cosine transformations to capture cyclical patterns.

- **Lag Features**: Historical context is provided using shifted values such as 1-hour and 24-hour lags (e.g., $\mathrm{PM}_{2.5}^{t-1}$ and $\mathrm{PM}_{2.5}^{t-24}$).

- **Rolling Statistics**: For $\mathrm{PM}_{2.5}$, PM_{10}, and O_3, 24-hour rolling mean, standard deviation, and maximum values are computed after a one-step shift to prevent data leakage.

Missing values resulting from lagging and rolling operations are handled using forward-fill and backward-fill strategies.

5 Feature Store

The Hopsworks Feature Store serves as a centralized feature management layer, offering:

- Efficient time-based feature retrieval for training and inference.
- Feature reuse across models and retraining cycles.
- Versioning and schema enforcement to ensure data consistency.
- Lineage and provenance tracking to prevent train–serve skew.
- Scalable storage for production-grade ML pipelines.

6 Model Training

The following regression models were trained and evaluated:

- Random Forest Regressor
- LightGBM Regressor
- XGBoost Regressor

The dataset was split into 80% training and 20% testing sets. Model performance was evaluated using RMSE, MAE, and R^2 metrics. The best-performing model was automatically persisted for deployment.

7 Results and Achievements

7.1 Model Performance

LightGBM demonstrated the best overall balance between accuracy and stability and was selected for deployment.

Table 1: Model Performance Comparison

Metric	**LightGBM**	**Random Forest**	**XGBoost**
RMSE	0.91	1.25	1.08
MAE	0.50	0.32	0.56
R^2	0.9993	0.9998	0.9991

7.2 Key Achievements

- Designed and implemented a complete end-to-end ML pipeline.
- Automated real-time data ingestion and transformation.
- Built a scalable and reusable feature store.
- Deployed a production-ready AQI forecasting dashboard.

8 Technologies and Tools

- **Languages & Libraries**: Python, Pandas, NumPy, Scikit-learn, LightGBM, XGBoost, Streamlit, Joblib, SHAP, LIME.
- **Data Sources**: Open-Meteo Air Quality and Weather APIs.
- **Infrastructure**: Hopsworks Feature Store, Streamlit Cloud, Git, GitHub Actions.

9 Challenges and Solutions

- **Data Reliability**: Addressed missing and delayed sensor readings using validation rules, imputation, and ingestion retries.
- **Feature Consistency**: Prevented train–serve skew through centralized, versioned feature storage.
- **CI/CD Stability**: Resolved intermittent training failures using session reuse, controlled concurrency, and retry backoff.
- **Scalability**: Optimized feature retrieval and storage to reduce training and inference latency.

10 Conclusion and Future Work

This project successfully delivered a robust and accurate AQI forecasting system by integrating real-time data ingestion, advanced feature engineering, machine learning modeling, and MLOps best practices. The system enables reliable short-term AQI prediction and provides an interactive visualization interface for end users.

Future Enhancements:

- Expansion to multiple cities and regions.
- Integration of deep learning models such as LSTM and GRU.
- Automated monitoring, alerting, and retraining pipelines.

11 Acknowledgments

I sincerely thank **10Pearls Pakistan** for providing this valuable internship opportunity. I am especially grateful to my mentors and team members for their continuous guidance and support throughout the project.