

Karachi — Air Quality Monitoring & Forecast Dashboard

Author: Afifa Siddique

Internship Project: 10Pearls Data Science – 2026

1. Introduction

Air pollution has become a major health concern in urban areas. Real-time monitoring and forecasting of Air Quality Index (AQI) can enable individuals and policymakers to take proactive measures. The **Pearls AQI Predictor** is an end-to-end serverless machine learning solution designed to **predict the AQI in Karachi for the next 3 days**. The system integrates automated data collection, feature engineering, model training, real-time prediction, and a web-based dashboard.

Key objectives:

- Provide **real-time AQI readings** and forecasts.
- Display AQI trends and top contributing pollutants.
- Issue alerts and health recommendations based on predicted AQI.
- Implement a **serverless, automated pipeline** for continuous feature and model updates.

2. Data Collection

2.1 Data Source

- **OpenWeather Air Pollution API:**
 - Provides hourly AQI and pollutant levels (PM2.5, PM10, CO, NO2, SO2, O3).
 - AQI values are **discretized on a scale of 1–5**:
 - 1 → Good (Green)
 - 2 → Fair (Yellow)
 - 3 → Moderate (Orange)
 - 4 → Poor (Red)
 - 5 → Hazardous (Purple)
- **Historical Data:** 4+ months of hourly AQI and pollutants were fetched for model training.

2.2 Data Storage

- **Feature Store:** Hopsworks
 - Stores processed historical and real-time features.

- Maintains a **versioned feature group** (`karachi_aqi_features`) with time-based indexing (`timestamp`).
- Ensures consistent, high-quality input for training and prediction.
- **Model Registry:** Dagshub MLflow
 - Stores and registers the best-performing model after each training run.
 - Enables reproducibility and version control for deployed models.

3. Feature Engineering

Features were derived from raw pollutant and timestamp data:

1. **Time-based features:**
 - Hour, Day, Month, Weekday
2. **Pollutant measurements:**
 - PM2.5, PM10, CO, NO2, SO2, O3
3. **Derived features:**
 - Pollutant change rates over past 7 days
 - Historical trends used for generating short-term forecasts

Processing Pipeline:

- Raw data from OpenWeather is processed using `src/features/historical_feature_pipeline.py` and `src/features/feature_pipeline.py`.
- Missing pollutant values are filled with -1.
- Features are inserted into Hopsworks for both historical backfill and real-time ingestion.

4. Exploratory Data Analysis (EDA)

Performed in `notebooks/eda.ipynb`. Key insights:

- Daily AQI exhibits **hourly and weekly seasonality**.
- PM2.5 and PM10 are the most influential pollutants.
- Minimal missing data due to API reliability and preprocessing.

5. Model Training Pipeline

5.1 Architecture

- Pipeline (`pipelines/training_pipeline.py`) connects to Hopsworks to fetch features and target.
- **Training split:** 80% train / 20% test.
- Models trained daily:
 1. Random Forest Regressor

2. Ridge Regression
3. Neural Network (TensorFlow Keras)

5.2 Evaluation Metrics

- **RMSE** (Root Mean Squared Error)
- **MAE** (Mean Absolute Error)
- **R² Score** (Coefficient of Determination)

5.3 Best Model Selection

- Random Forest achieved **best performance**:
 - MAE: 0.00964
 - R²: 0.9913
 - RMSE: 0.0861

Reasons for Random Forest superiority:

- Ensemble method reduces variance and overfitting.
- Handles **non-linear interactions** between pollutants and time features.
- Discrete AQI scale (1–5) makes regression stable.
- Consistency in features from Hopsworks ensures reliable daily predictions.

5.4 Model Registration

- Best model automatically registered in Dagshub MLflow as **AQI_Predictor_Best**.
- Deployed for inference in the dashboard pipeline.

6. Automated CI/CD Pipeline

Implemented via **GitHub Actions**:

1. **Feature Pipeline**
 - Runs hourly (`.github/workflows/feature_pipeline.yml`)
 - Fetches latest AQI and pollutant data
 - Updates Hopsworks feature store
2. **Training Pipeline**
 - Runs daily at 00:30 UTC
(`.github/workflows/training_pipeline.yml`)
 - Retrains models on latest data
 - Registers best model

Benefits:

- Fully automated serverless architecture
- Ensures **up-to-date features and models**
- Eliminates manual intervention

7. Inference & Prediction

7.1 Feature Forecasting

- `src/inference/predict_aqi.py` generates **future features** using:
 - Last 7 days of pollutant data from Hopsworks
 - Linear extrapolation + random perturbations (demo logic for 3-day forecast)

7.2 AQI Prediction

- Best model loaded from Dagshub MLflow.
- Generates 3-day AQI forecast along with optional SHAP explanations.

7.3 SHAP Analysis

- Identifies **top features driving predictions** for transparency and interpretability.

8. Web Dashboard

Implemented using Streamlit (`app/app.py`):

Features

1. **Real-time AQI display** (1–5 scale, color-coded)
2. **3-Day AQI Forecast Table**
3. **3-Day Trend Visualization** (Altair line chart, y-axis 1–5)
4. **30-Day Demo Forecast** for long-term visualization
5. **Top 5 SHAP Feature Contributions**
6. **Live Pollutant Composition – Last 7 Days**
7. **Interactive Map of Karachi AQI** (PyDeck)
8. **Health Recommendations & Alerts** based on AQI category

UX Highlights:

- Color-coded AQI categories for immediate understanding
- Clear visualizations for pollutants and trends
- Compact SHAP bar charts for feature importance

9. Technology Stack

Component	Technology / Tool
Data Collection	OpenWeather API
Feature Store	Hopworks
Model Registry	Dagshub MLflow
Machine Learning Models	Random Forest, Ridge, Neural Network
ML Libraries	scikit-learn, xgboost, TensorFlow
Model Explainability	SHAP
Dashboard / Frontend	Streamlit, Altair, PyDeck
CI/CD	GitHub Actions
Programming Language	Python 3.11
Dependency Management	requirements.txt, requirements_pipeline.txt

10. Results & Observations

- **Random Forest consistently provides best daily performance** due to ensemble averaging and robustness to noise.
- Predictions **align closely with historical trends** (high R²).
- SHAP analysis shows PM2.5 and PM10 dominate AQI forecasting, matching real-world pollution patterns.
- Daily automated retraining ensures **dynamic adaptation to new data**, improving forecast reliability.

11. Conclusion

The Pearls AQI Predictor demonstrates a full **serverless ML solution** for urban air quality forecasting. Key achievements:

- Automated **data ingestion, feature engineering, and model training**.
- Integration of **Hopworks Feature Store** and **Dagshub MLflow** for reproducibility.
- Deployment of a **user-friendly Streamlit dashboard** with real-time predictions, SHAP explanations, and health recommendations.

- Stable, highly accurate predictions using **Random Forest**, with MAE < 0.01 and R² > 0.99.

This project provides a **scalable, maintainable, and interpretable AQI prediction framework**, suitable for real-world deployment in smart city applications.

12. Future Enhancements

- Integrate **more granular AQI sensors** across Karachi for higher spatial resolution.
 - Implement **LSTM or Transformer models** for long-term AQI forecasting.
 - Include **weather forecasts** as features to improve 3-day and 30-day predictions.
 - Add **push notifications / email alerts** for hazardous AQI levels.
 - Expand to **multi-city AQI predictions** with the same serverless architecture.
-

Project Repository: <https://github.com/AfifaSiddiquee/pearls-aqi-predictor>

Live App: [Pearls AQI Predictor · Streamlit](#)