Pearls AQI Predictor: System Briefing

Executive Summary

The Pearls AQI Predictor is a fully automated, serverless machine learning system designed to provide a 3-day Air Quality Index (AQI) forecast for Karachi, Pakistan. The system exemplifies modern MLOps practices, integrating a suite of tools to create an end-to-end pipeline that requires no manual intervention. Key components include live data ingestion from Open-Meteo APIs, automated feature engineering, and storage in a Hopsworks Feature Store. A Random Forest Regressor model is retrained daily on 90 days of historical data, with performance evaluated using Mean Absolute Error (MAE). The entire process is orchestrated via two distinct CI/CD workflows in GitHub Actions, one running hourly for data ingestion and another daily for model training. The project culminates in an interactive web application built with Streamlit, which displays forecasts, provides model explanations using SHAP, and issues alerts for hazardous air quality levels (AQI > 150). The system is described as a production-ready solution that demonstrates a mastery of applied machine learning and MLOps principles.

----

1. Project Overview

The Pearls AQI Predictor is an end-to-end system for forecasting air quality in Karachi. It leverages real-time weather and pollution data to generate predictions, making it a practical tool for monitoring environmental conditions. The project's foundation is a complete MLOps pipeline designed for automation, scalability, and reliability.

Core Project Objectives & Features:

- **Forecasting:** Predicts the Air Quality Index (AQI) for Karachi over the next 3 days.
- **Automation:** Employs 100% CI/CD automation for both data pipelines and model training, eliminating the need for manual intervention.
- **Real-Time Data:** Ingests live, hourly weather and pollution data from Open-Meteo APIs.
- **Advanced Feature Engineering:** Automatically creates time-based and lag-based features to improve model accuracy.
- **Daily Retraining:** The machine learning model is retrained daily to adapt to new data patterns.
- **Interpretability:** Integrates SHAP (SHapley Additive exPlanations) to provide clear explanations for the model's predictions.
- **User Interface:** A real-time dashboard provides forecasts and alerts for hazardous AQI levels.

2. System Architecture and MLOps Pipeline

The system is architected around two primary automated pipelines, supported by a robust MLOps infrastructure. Both pipelines are orchestrated using GitHub Actions and have a reported status of "GREEN."

2.1 Hourly Feature Pipeline

This pipeline is responsible for the continuous ingestion and processing of raw data into machine learning-ready features.

- **Script:** `feature_pipeline.py`
- **Automation:** Runs every hour via a GitHub Actions workflow (`features.yml`) triggered by a cron job (`'0 * * * *'`).
- **Data Source:** Fetches real-time weather and air quality data from Open-Meteo APIs.
- **Feature Engineering:** Engineers over 12 distinct features, including:
    - **Pollutants:** PM2.5, PM10
    - **Meteorological Data:** Temperature, humidity
    - **Time-Based Features:** Hour, day, month, dayofweek
    - **Lag-Based Features:** AQI change rate
- **Storage:** Processed features are stored in the offline mode of the Hopsworks Feature Store to ensure data consistency.
- **Status:** Reported as actively running for over two days without issues.

2.2 Daily Training Pipeline

This pipeline manages the daily retraining, evaluation, and versioning of the predictive model.

- **Script:** `training_pipeline.py`
- **Automation:** Runs daily at midnight via a GitHub Actions workflow (`train.yml`).
- **Training Data:** Utilizes 90 days of backfilled historical data for robust model training.
- **Model:** Employs a **Random Forest Regressor** to predict AQI values.
- **Evaluation:** Model performance is assessed using **Mean Absolute Error (MAE)** and feature importance analysis.
- **Artifacts and Versioning:** The trained model and a corresponding SHAP feature importance plot are saved to the Hopsworks Model Registry for version control and retrieval.

2.3 Core MLOps Infrastructure

The project implements several MLOps best practices using a modern technology stack.

Component	Technology	Purpose & Implementation
CI/CD Automation	GitHub Actions	Orchestrates two workflows: `features.yml` for hourly data ingestion and `train.yml` for daily model retraining.
Feature Store	Hopsworks	Provides a centralized repository for versioned features, ensuring consistency between training and inference.
Model Registry	Hopsworks	Manages model versions, storing trained models and associated artifacts like SHAP plots.
Explainable AI (XAI)	SHAP	Integrated to explain model predictions, enhancing trust and interpretability.
Monitoring	GitHub Actions Logs	Used for monitoring the status and execution of the automated data and training pipelines.

3. Web Application and User Interface

A user-facing dashboard provides actionable insights from the system's predictions.

- **Framework:** Built with Streamlit.
- **Functionality:**
  - **Data Retrieval:** Loads the latest trained model and features directly from the Hopsworks Feature Store and Model Registry.
  - **AQI Forecast:** Displays a 3-day AQI forecast for Karachi.
  - **Model Interpretability:** Shows a SHAP feature importance plot, illustrating which factors most influence the predictions.
  - **Alerting System:** Triggers an alert when the predicted AQI exceeds 150, which is classified as "Hazardous."

4. Project Achievements and Deliverables

The project is presented as a complete and production-ready system that successfully demonstrates the implementation of advanced MLOps concepts.

Key Achievements

- **End-to-End Automation:** The system operates without any manual intervention for data processing or model updates.
- **Production-Ready System:** A fully automated and scalable pipeline suitable for real-world application.
- **Implementation of MLOps Best Practices:** Successfully integrated a Feature Store, Model Registry, automated retraining, and Explainable AI.