Pearls AQI Predictor

Developed by

Maria Abid

Table of Contents

Project Overview

The Pearls AQI Predictor is a data-driven system designed to forecast the Air Quality Index (AQI) specifically for Karachi over a three-day horizon. The project integrates advanced machine learning techniques, cloud-based feature management, and real-time visualization to provide accurate, interpretable, and actionable insights into urban air quality. The system collects environmental and pollutant data, including PM2.5, PM10, O_3 , NO_2 , CO, and SO_2 , along with meteorological parameters such as temperature, humidity, and wind speed, directly from the Open-Meteo API. These raw data are preprocessed, transformed, and stored in a centralized Hopsworks Feature Store, ensuring consistency and reliability between the model training and inference pipelines.

A modular machine learning workflow was implemented, training multiple forecasting models including Random Forest, Ridge Regression, and neural network-based architectures. Models were evaluated using standard regression metrics such as RMSE, MAE, and R^2, and the best-performing model was registered for deployment in production. The final deployment features a Streamlit-based dashboard, offering a visually intuitive interface that displays live AQI readings, 3-day forecasts, and detailed pollutant contributions. To enhance interpretability, SHAP-based explainability has been incorporated, allowing users to understand the influence of environmental factors on AQI predictions.

By combining data-driven forecasting and automated cloud workflows, the Pearls AQI Predictor supports urban sustainability initiatives and public health awareness, demonstrating the potential of intelligent systems in smart city applications.

Project Structure

```
aqi_forecaster/
├── .github/
│   └── workflows/
│       └── pipeline.yaml          # CI/CD workflow for GitHub Actions
│
├── app/
│   ├── dashboard.py               # Streamlit dashboard for visualization
│   └── style.css                  # Custom CSS for dashboard styling
│
├── data/
│   ├── raw_openmeteo/             # Raw weather data fetched from Open-Meteo API
│   ├── features/                  # Engineered features ready for training
│   └── predictions/               # Stored forecast results
│
├── features/
│   ├── backfill.py                # Historical data backfill and feature creation
│   ├── compute_aqi.py             # AQI computation logic (US AQI scale)
│   ├── live_aqi.py                # Fetches and updates live AQI readings
│   └── preprocess.py              # Cleans data, handles missing values, ensures consistency
│
├── trainings/
│   ├── train_sklearn.py           # Random Forest and Ridge Regression training
│   ├── train_tf.py                # TensorFlow Dense Neural Network training
│   └── predict.py                 # Forecasting next 3-day AQI using trained models
│
├── eda.ipynb                      # Exploratory Data Analysis notebook
├── requirements.txt               # Python dependencies
└── .env                           # Environment variables for API keys and config
```

3. Features

The Pearls AQI Predictor is a comprehensive system that integrates multiple pipelines to ensure accurate and real-time AQI forecasting. Its key functionalities include automated data collection, feature computation, model training, prediction generation, visualization, and explainability. The system is specifically tailored for Karachi, leveraging both pollutant and meteorological data to provide actionable insights for urban air quality monitoring.

3.1. Feature Pipeline

The feature pipeline is responsible for fetching raw air quality and weather data from the Open-Meteo API. It computes derived features such as AQI change rate and temperature-humidity index to enrich the dataset. All processed features are stored in the Hopsworks Feature Store, ensuring consistency between training and real-time inference. This modular pipeline allows for efficient, automated handling of environmental data. It supports scalability and ensures that features remain standardized across all downstream workflows.

3.2. Historical Data Backfill and Data Preprocessing

The backfill process generates comprehensive datasets for model training using a minimum of 90 days of hourly historical data. It ensures temporal consistency and completeness of features to improve forecasting accuracy. Historical backfill enables the system to learn long-term trends and seasonal patterns in air quality. Data preprocessing includes handling missing values, outliers, and time alignment. This foundation is critical for creating reliable machine learning models capable of predicting AQI under varying environmental conditions.

3.3. Training Pipeline

The training pipeline handles the development and evaluation of multiple machine learning models, including Random Forest, Ridge Regression, and TensorFlow Dense Neural Networks. Models are assessed using standard regression metrics such as RMSE, MAE, and R^2. The pipeline supports automated model selection and hyperparameter tuning to ensure optimal performance. Once trained, models are uploaded to the Hopsworks Model Registry for versioning and deployment. This setup ensures that production-ready models are consistently reproducible and auditable.

3.4. Prediction Pipeline

The prediction pipeline fetches the latest computed features to generate AQI forecasts for the next three days. Forecasts are calculated in real-time and uploaded to a dedicated Hopsworks Feature Group, karachi_aqi_predictions, for downstream use. This ensures seamless integration with the dashboard and other visualization tools. The pipeline supports automated scheduling, enabling continuous monitoring without manual intervention. It guarantees that predictions are always based on the most recent environmental data.

3.5. Dashboard

The interactive Streamlit dashboard provides a user-friendly interface to monitor Karachi's air quality. It displays the current AQI along with pollutant details, 3-day forecast cards, and actual vs predicted AQI charts. Trend charts and correlation heatmaps allow users to understand relationships between different pollutants and environmental factors. The dashboard is updated in real-time and visually highlights hazardous AQI levels using color-coded indicators. It provides actionable insights for both general users and city planners.

3.6. Explainability

Explainability is a core feature of the system, enabling users to understand how environmental factors influence AQI predictions. SHAP-based analysis highlights the

contribution of each pollutant and weather parameter. The system can also send alerts for hazardous AQI levels, either via dashboard notifications or color-coded UI cues. This ensures users are informed about potential health risks in real time. Explainability enhances trust in the model and provides transparency for data-driven decision-making.

4. Tech Stack

Karachi AQI Forecasting System leverages a modern, scalable technology stack combining machine learning, data engineering, and web visualization to provide automated and explainable AQI predictions in real time.

Programming & Development: Python 3.10 is used across the pipeline, with Jupyter and VS Code for experimentation, and Streamlit for the interactive dashboard. Git & GitHub manage version control and CI/CD workflows.

Machine Learning & Data Science: Models include Random Forest (scikit-learn) and optional TensorFlow networks. Data manipulation and computation are handled with pandas and NumPy, while matplotlib, seaborn, and plotly provide visualizations. joblib is used for efficient model serialization.

Data & MLOps Infrastructure: Features and models are stored in Hopsworks Feature Store and Model Registry, ensuring reproducibility. dotenv manages environment variables securely, while OpenWeatherMap and AQICN APIs supply live data.

Automation & CI/CD: GitHub Actions orchestrates scheduled pipeline execution, model retraining, and prediction uploads, with workflows defined in YAML for transparency and reproducibility.

Visualization & Front-End: The Streamlit dashboard displays live AQI, forecasts, pollutant breakdowns, and feature importances. Plotly charts and custom CSS styling provides a professional and user-friendly interface.

5. Setup and Installation

5.1. Create Virtual Environment

```
conda create aqi-py310
conda activate aqi-py310          # Windows
```

5.2. Install Dependencies

```
pip install --upgrade pip
pip install -r requirements.txt
```

6. Environment Setup

6.1. Configure .env file

```
HOPSWORKS_HOST=your host
AQICN_TOKEN=your_aqicn_token
HOPSWORKS_API_KEY=hopsworks_api_key
HOPSWORKS_PROJECT=name_of_the_project
HOPSWORKS_PROJECT_ID=project_id
```

6.2 . Hopsworks Authentication

```
python -m hopsworks.login
```

7. Project Workflow

The Pearls AQI Predictor operates as a fully automated, serverless, end-to-end machine learning system for forecasting Karachi's Air Quality Index (AQI). The system integrates data ingestion, feature engineering, model training, real-time prediction, visualization, and CI/CD automation to ensure accurate, interpretable, and continuously updated air quality insights.

7.1. Data Ingestion

Data is collected via the Open-Meteo API, including air pollutants (PM2.5, PM10, CO, NO_2 , SO_2 , O_3) and meteorological variables (temperature, humidity, wind speed). Data from backfill.py and live_aqi.py scripts are merged on timestamps to create a unified DataFrame. Processed features are uploaded to the Hopsworks Feature Store as versioned Feature Groups, ensuring consistency between training and inference. AQI is computed using compute_aqi.py, mapping pollutant concentrations to US EPA AQI values and assigning categorical labels (Good, Moderate, Unhealthy, Hazardous). Key Feature Groups include karachi_aqi_backfill and karachi_aqi_us.

7.2. Historical Data Backfill

Historical data for the past 90 days of hourly data is retrieved via backfill.py, ensuring temporal completeness and feature integrity. The collected data undergoes validation before being uploaded to Hopsworks asynchronously. Backfilled datasets provide the foundation for training, enabling the system to capture pollutant patterns, temporal trends, and meteorological dependencies for short-term forecasting.

7.3. Live Data Integration

The live_aqi.py module fetches hourly real-time data to update the Feature Store. It maintains feature consistency with the historical pipeline and ensures that the dashboard and prediction pipelines always have access to the latest environmental conditions. Live ingestion can be triggered automatically through CI/CD workflows for continuous updates.

7.5. Data Preprocessing

The preprocess.py script ensures data integrity before training and forecasting. It loads the karachi_aqi_us feature group from Hopsworks, checks for missing values and duplicates, and fills gaps in aqi_pm10 and aqi_o3 using linear interpolation and forward

filling. This process maintains consistency and reliability across all AQI and weather features used in model development.

7.4. Model Training

The system trains two types of models: Random Forest Regressor (Scikit-learn) and Dense Neural Network (TensorFlow/Keras).

- Random Forest: Uses pollutant and meteorological features to predict US AQI. Data is time-split (last 10 days for testing), missing values are removed, and feature importances are recorded. The trained model is stored in Hopsworks Model Registry with metadata and evaluation metrics (RMSE, MAE, R^2). Achieves ~87% R^2.
- Dense Neural Network: Captures nonlinear dependencies using two hidden layers (64, 32 units) with Dropout and ReLU activations. Input/output features are standardized, and early stopping and adaptive learning rate callbacks are applied. Achieves ~78% R^2, providing a secondary benchmark.

The Random Forest model was selected for production due to superior accuracy and interpretability.

7.5. Real-Time Predictions

The predict.py script uses the most recent 20 days of data to forecast AQI for the next 3 days. Predictions incorporate trend-based adjustments for temporal consistency and realism. Forecasts are automatically uploaded to the karachi_aqi_predictions Feature Group in Hopsworks, ensuring the dashboard always displays up-to-date values without manual intervention.

7.6. Prediction and Dashboard Workflow

The predict.py script generates 3-day AQI forecasts for Karachi by retrieving the latest features from the Hopsworks Feature Store and loading the most recent Random Forest model. Predictions combine model outputs with a 20-day AQI trend adjustment for

improved accuracy and realism, then are uploaded to the karachi_aqi_predictions feature group for real-time dashboard access.

The dashboard connects securely to Hopsworks and displays:

- **Live AQI Card:** Current AQI, health category, timestamp, and dominant pollutants.
- **3-Day Forecast Cards:** Predicted AQI values with color-coded categories.
- **AQI Trends & Actual vs Predicted**: Interactive line charts for recent AQI fluctuations and model performance.
- **Pollutant Insights:** Contribution over time and composition of key pollutants.
- Feature Importance & Correlation: Charts for model explainability and pollutant relationships.
- **Latest Data Sample**: Quick view of recent AQI records.

7.7. CI/CD Automation

GitHub Actions orchestrates the full pipeline:

- Hourly feature ingestion and backfilling
- Daily model retraining
- Automated 3-day AQI forecast generation and upload
- Streamlit dashboard deployment

All steps are defined in YAML workflows, ensuring serverless, reproducible automation. Logs, models, and predictions are stored as artifacts for traceability.

8. Key Highlights

- Retrieved 90 days of hourly air quality and weather data from Open-Meteo API, preprocessed it, and created a feature group in Hopsworks.
- Continuous ingestion of hourly air quality data via GitHub Actions, automatically appended to the existing feature group for up-to-date forecasting.

- Developed Random Forest and LSTM (Dense Neural Network) models. Random Forest outperformed due to its ability to handle smaller datasets efficiently, making it the primary production model. Trained models are stored in the Hopsworks Model Registry.
- Generated 3-day AQI forecasts using the trained Random Forest model. Predictions are uploaded to a new Hopsworks Feature Group for downstream access.
- The Streamlit dashboard dynamically retrieves real-time and predicted AQI from Hopsworks, displaying live AQI, 3-day forecasts, pollutant breakdowns, trends, and feature importance charts.
- The entire workflow from data ingestion and feature updating to model training, prediction, and dashboard updates is automated via GitHub Actions, ensuring a reliable, serverless, and continuous system.

9.Future Enhancements

- Integrate additional data sources such as satellite-based pollution data for improved AQI accuracy.
- Implement LSTM or hybrid deep learning models once sufficient historical data is available.
- Add mobile app integration for real-time notifications and personalized AQI alerts.
- Enhance dashboard with predictive analytics, trend forecasting, and scenario simulations.
- Incorporate automatic anomaly detection for sudden pollution spikes and alert systems.

Conclusion

The Pearls AQI Predictor is an automated system for forecasting Karachi's air quality using historical and live data, machine learning, and real-time visualization. The

Random Forest model provides accurate predictions even with limited data, while the dashboard offers actionable insights on AQI trends and pollutant contributions.

Fully automated pipelines via GitHub Actions ensure continuous data updates, model retraining, and real-time forecasting. The system demonstrates how AI, data engineering, and cloud automation can support environmental monitoring, public awareness, and smart-city applications.