Air Quality Index (AQI) Prediction and Forecasting Dashboard

Introduction

The Air Quality Prediction System is a serverless machine learning application that predicts and visualizes real-time air quality trends in Karachi.
It leverages modern MLOps tools, automating the entire pipeline from data ingestion to model training and live visualization through a Streamlit-based dashboard.

Air quality monitoring is crucial for urban health planning, especially in high-density regions. This project provides actionable insights into short-term air pollution fluctuations, enabling better decision-making for both environmental analysts and the general public.

Objectives

The main objectives of this project are:

1. To automate **data collection**, **cleaning**, and **feature engineering** for AQI prediction.
2. To build a **machine learning pipeline** for 72-hour AQI forecasting.
3. To implement **model training and deployment** using a **serverless architecture**.
4. To provide an **interactive dashboard** that visualizes live and forecasted AQI values.

Tools and Technologies

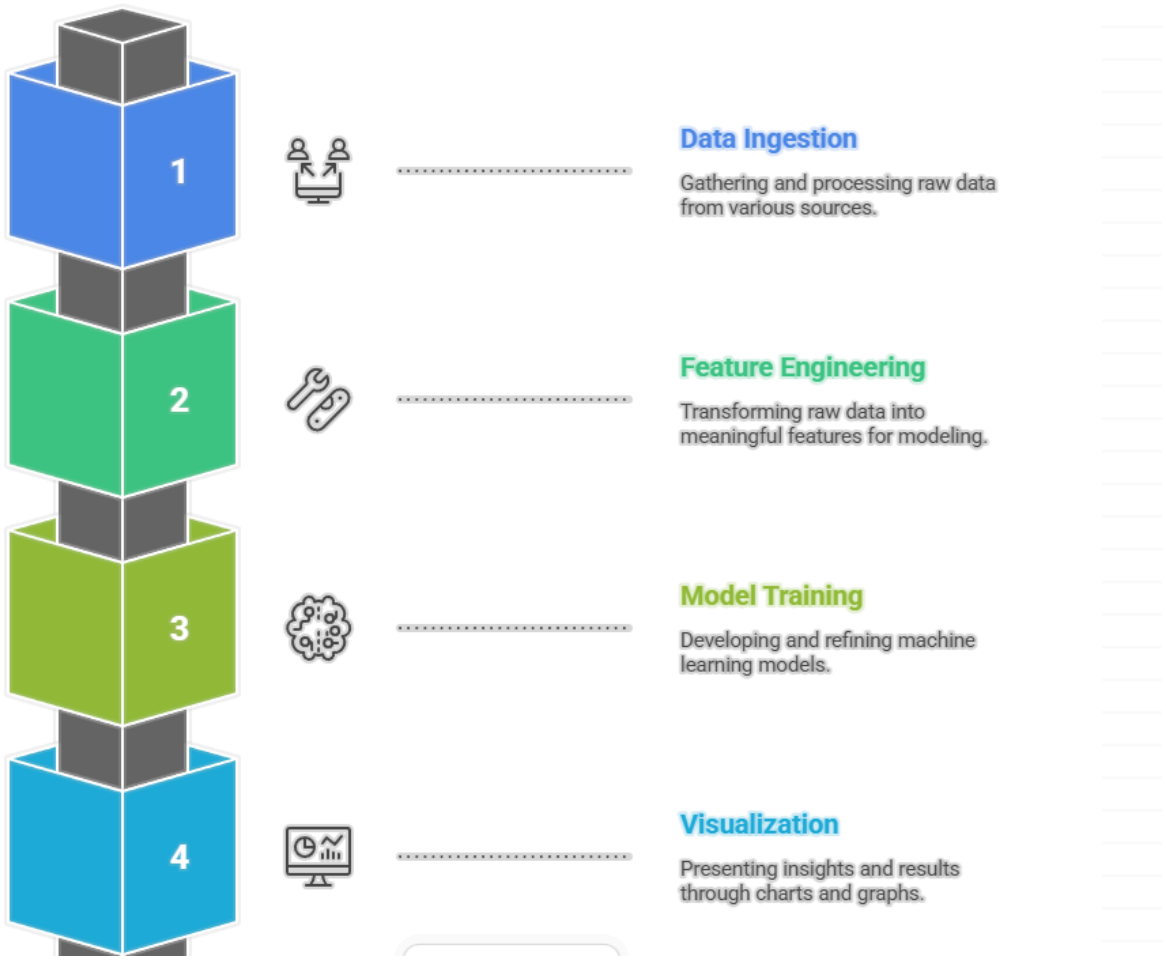
- **Language:** Python
- **Frameworks:** Streamlit, scikit-learn, Pandas, NumPy
- **Model Management:** Hopsworks Feature Store

- **CI/CD Integration:** GitHub Actions for automation
- **Visualization:** Matplotlib, Plotly, Streamlit Charts
- **Deployment:** Streamlit Cloud

Architecture Overview

The system follows a **serverless MLOps architecture** consisting of the following components:

1. **Data Ingestion Layer:**
    - Automatically fetches real-time meteorological and air quality data (PM2.5, PM10, NO_2, etc.) from open APIs.
    - Cleans and aggregates the data on an hourly basis.
2. **Feature Engineering & Storage:**
    - Stores processed features in **Hopsworks Feature Store**.
3. **Model Training:**
    - A regression model (Random Forest) is trained on historical AQI data.
    - GitHub Actions trigger model retraining automatically based on new data updates.
4. **Model Deployment:**
    - The trained model is deployed through **Streamlit**, integrated with Hopsworks for feature retrieval.
    - The inference endpoint generates predictions for the next 72 hours.
5. **Visualization Layer:**
    - The Streamlit dashboard displays real-time AQI values, predicted trends, and pollutant insights.
    - Users can download forecast data as CSV for analysis.



Implementation Details

1. **Data Processing:**
    - Raw AQI and weather data are ingested hourly from APIs.
    - Missing values are imputed using forward fill and median interpolation.
    - Normalization ensures the data is suitable for ML modeling.
2. **Model Training:**
    - The model predicts AQI using a supervised regression algorithm.

    - Cross-validation ensures the model generalizes well.
    - Trained models are stored and versioned in Hopsworks.
3. **Forecasting:**
    - The model predicts AQI values for the next 72 hours (3 days).
    - Predicted results are combined with timestamps and weather parameters.
4. **Automation:**
    - GitHub Actions automatically retrain the model when new data arrives.
    - Updated forecasts are pushed to the live Streamlit dashboard.

Output

The Streamlit dashboard provides a complete visual interface for monitoring and forecasting Air Quality Index (AQI) values for Karachi. The application is designed with a clean dark interface for readability and intuitive navigation, offering four major visualization and data components:

1. Current AQI Overview

Displays the **most recent AQI reading** fetched from the model's latest prediction cycle.

- Shows current AQI value and corresponding category (Good, Moderate, Unhealthy, etc.).
- Provides timestamp and location metadata.
- Color coding helps users assess current air pollution conditions at a glance.

Refer to Figure 1: Current AQI Section.

2. Forecasting Visualization

This section displays the **72-hour AQI forecast** as a dynamic line chart.

- Plots **Actual AQI** vs **Predicted AQI** to compare historical and forecasted performance.
- Users can visualize short-term pollution fluctuations over the next three days.
- Forecasts are automatically updated based on the most recent data pipeline run.

3. Daily Forecast Breakdown

The **Forecast Table** provides a detailed hourly prediction for Today, Tomorrow, and the Day After Tomorrow.

- Includes timestamp, predicted AQI value, and air quality category.
- Users can export the forecast as CSV for further data analysis.
- This section supports decision-making for outdoor planning and environmental assessment.

4. Weather and Pollutant Data Table

Displays the input weather and pollutant data used for prediction:

- Temperature, humidity, dew point, rainfall, wind speed, and pressure.
- Offers transparency about environmental conditions influencing AQI.
- Users can download the underlying dataset for verification and research purposes.

Findings and Model Performance

During experimentation, multiple models were evaluated for AQI forecasting, including **Random Forest Regressor** and **LSTM Neural Networks**.
The goal was to balance **accuracy, speed, and computational efficiency** for a serverless environment.

- **Random Forest Model:**
  Achieved strong predictive accuracy while remaining lightweight and fast during inference.
  It captured non-linear relationships between meteorological factors and pollutant concentrations effectively.
  Due to its tree-based ensemble structure, it also provided interpretability and stability against noisy input data.

- **LSTM Model:**
  Although it performed well on sequential dependencies, it required higher computational power and longer training times.
  Deploying LSTM in a serverless setup introduced latency and cost overhead due to GPU dependencies.

Conclusion

The AQI Prediction System demonstrates the power of serverless MLOps pipelines in environmental analytics. By integrating automated data ingestion, model retraining, and real-time visualization, it ensures an efficient, scalable, and accurate forecasting system.

Future work may involve:

- Expanding coverage to other cities.
- Integrating deep learning models (e.g., LSTMs or Transformers).
- Incorporating additional pollutant parameters such as CO, SO_2, and O_3.

This project provides a practical foundation for data-driven air quality management and sustainable urban planning.