10Pearls Internship Task: Data Sciences

Real-Time Air Quality Index (AQI) Prediction System

Submitted By: Savera Rizwan

Project Overview:

Air pollution poses a major health and environmental challenge, particularly in rapidly urbanizing cities like Karachi. This project presents a Real-time Air Quality Index (AQI) prediction system**.**
It integrates live pollutant data from OpenWeather's Air Pollution API, applies feature engineering and EPA-based AQI calculation, and uses machine learning models deployed through Hopsworks to predict AQI and compare it with the actual AQI value. Furthermore, the project uses automated scheduling and pipeline orchestration through GitHub Actions. The frontend is made on Streamlit, providing an interactive dashboard that visualizes predictions and pollutant details.

Objectives

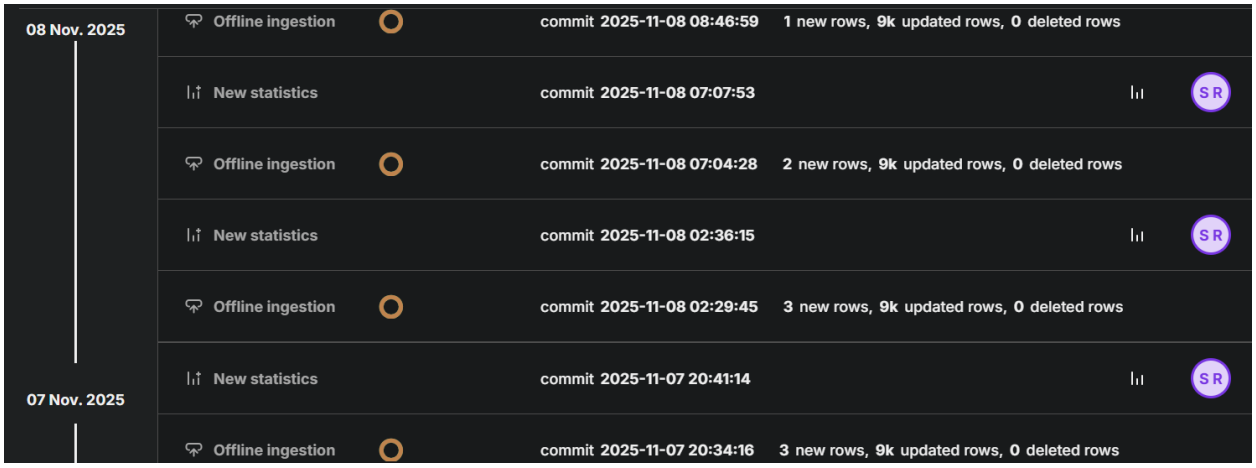
- Fetch live air pollution data (e.g., $PM_{2.5}$, PM_{10}, CO, NO_2, SO_2, O_3) from OpenWeather API**.**
- Compute the AQI using EPA standards and train the ML models (Gradient Boosting, Random Forest, and XGBoost) with features and AQI.
- Predict AQI using trained ML models.
- Automate the fetching of data, applying EDA on updated data and retraining on updated data
- Compare predicted vs actual AQI in real-time.
- Display results in a clean, interactive Streamlit dashboard.

Workflow

1- Fetching of Data
    - We fetched last two years data (2 years to current time) using OpenWeather API. The dataset was divided into 4 parts, each representing 6-month segments and resampled into 3-hour intervals to match OpenWeatherMap API frequency. This data is saved in a CSV file.
    - Every 3 hours the Fetch Data Workflow (fetch_data.yml) is run automatically which updates the CSV file with new data. The updated data is automatically committed and pushed.

2- Data Cleaning and Preprocessing, Exploratory Data Analysis (EDA) and Feature Engineering

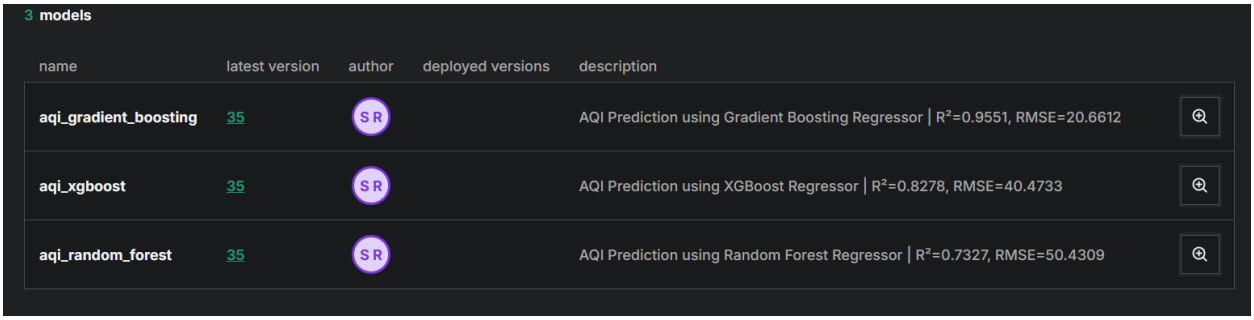
- Data cleaning and preprocessing is done for example handling missing values, removing outliers and handling negative values (if any).
- OpenWeather provides simplified 1-5 AQI scale which not suitable for precise modeling so we calculate AQI using the pollutants data. Overall AQI = Maximum of all individual pollutant AQIs. Dominant pollutant is identified (the one causing highest AQI)
- Carry out feature engineering like temporal features, pollutant interactions, rolling averages. Around 26 features were engineered. Using F-Statistic Method and Mutal Information Method we determined which features to choose.
- Carry out data validation for Hopsworks to comply with Hopsworks Feature Store schema requirements.
- This EDA occurs every 15 mins after 3 hours i.e after every 15 mins of fetching data.



The data in the Feature store is being updated. Due to Github Actions delays in free tier, there are some irregularities in timings so whenever the workflow executes, the number of new rows added will be added as new rows and the overall rows will be updated.

3- Model Development

- Data is split in training (70%), validation (10%) and test (20%)
- Three models have been trained: Random Forest Regressor, Gradient Boosting Regressor, and XGBoost Regressor
- The models are retrained every 6 hours with latest data.

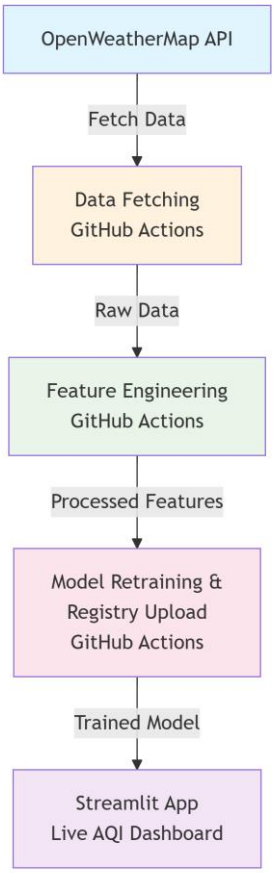


The models in model registry. The version number corresponds to the number of times the models have been retrained. So this is the 35th version which has been retrained on updated data.

4- Frontend

- We can select any of the three models.
- Real-time data is being fetched from OpenWeather API. AQI will be calculated using EPA formula. The actual AQI and predicted AQI will be displayed.

System Architecture Overview



Results

Based on the comprehensive evaluation of three machine learning models for AQI prediction, Gradient Boosting emerged as the clear winner, demonstrating exceptional performance with the highest R-squared score and lowest error metrics. XGBoost secured the second position, delivering good overall performance though falling short of Gradient Boosting's excellence. Random Forest ranked third among the evaluated models, showing acceptable but comparatively modest performance. This model demonstrated the lowest variance explanation capability and exhibited the highest error rates among the three.