Karachi AQI Prediction – Project Report

1. Project Overview

The objective of this project was to develop an **Air Quality Index (AQI) prediction system for Karachi** using real-time and historical environmental data. The system followed an end-to-end machine learning pipeline including **data collection, exploratory data analysis (EDA), feature storage, model training, and deployment**.

AQI data was obtained from the **OpenWeather API**, which provides AQI values on a **1–5 scale**.

2. Data Collection (OpenWeather API)

Approach

The OpenWeather API was used to fetch:

- AQI (1–5 scale)
- PM2.5 concentration
- Temperature
- Humidity

The data primarily consisted of **current and limited historical observations**.

Issues Faced

- **Limited Data Availability**
- The API does not provide extensive historical AQI data.
- Only a short time window was accessible.
- This resulted in:
- Small dataset size
- Weak temporal patterns for modeling
- **Repeated Data from API**
- Multiple API calls returned **identical AQI and feature values**.
- This repetition initially went unnoticed and later caused issues during training.

3. Exploratory Data Analysis (EDA)

Actions Taken

- Identified duplicated records based on:
- Timestamp
- AQI value
- Pollutant readings
- Removed repeated rows
- Analyzed feature distributions and variability

Key Finding

EDA revealed that without cleaning:

- The model would learn AQI as a **constant value**
- Predictions would be meaningless

This confirmed that the problem was **data-related rather than model-related**.

4. Data Validation and Consistency Checks

Validation Steps

- Verified AQI values remained within the 1–5 range
- Checked for missing or null values
- Ensured correct mapping between AQI and pollutant features

Issues Encountered

- In several cases, AQI remained unchanged while pollutant values varied.
- This indicated incomplete or inconsistent data being fetched from the API.

5. Feature Store Integration

Tool Used

- **Hopsworks**

Purpose

Store cleaned and validated features centrally
Ensure consistent features for training and inference

Issues Faced

Unstable Feature Store Connectivity
Features were written but not always synced properly.
Online feature store access was unreliable.
Impact
Models trained on outdated or incomplete feature data.
Prediction accuracy was negatively affected.

6. Model Training Challenges

Observed Behavior

The model repeatedly predicted the **same AQI value**.
Learning collapsed to predicting the average AQI.

Root Causes

Limited historical data
Low variation in AQI values
Feature store inconsistencies
Coarse AQI scale (1–5)

Conclusion

The issue was not with the algorithm, but with:

Data quantity
Data quality
Feature consistency

7. CI/CD Pipeline

Tool Used

GitHub Actions

Status

Pipelines executed successfully
No major build or deployment failures
CI/CD was not a contributing factor to model issues

8. Streamlit Deployment

Tool Used

Streamlit

Issues Faced

“No trained models found” Error
The Streamlit application failed to locate trained model artifacts.
Likely causes:
Incorrect model paths
Models not saved or registered properly
Feature store not accessible at runtime
Limited Forecast Horizon
The app predicted AQI for only **three days** based on current AQI.
Proper time-series forecasting was not possible due to:
Lack of long-term historical data
Missing lag-based features