Air Quality Index (AQI) Prediction System
Technical Project Report

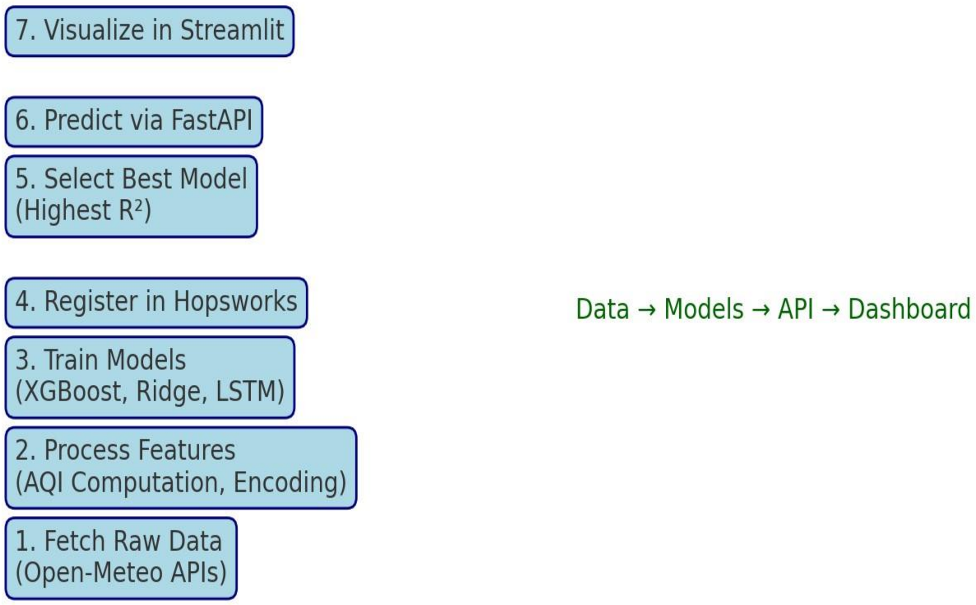
By: Muhammad Owais Iqbal

1. Introduction

The Air Quality Index (AQI) Prediction System aims to estimate real-time air quality based on weather and pollutant data. The system provides current and 3-day AQI forecasts. It uses machine learning and deep learning models such as XGBoost, Ridge Regression, and LSTM for training. The system automates data collection, feature processing, model training and deployment through Hopsworks, FastAPI, and Streamlit. The Automation is done through implementation of CI/CD pipelines.

2. Project Architecture

The project architecture is modular and follows a full MLOps workflow. It includes data collection, preprocessing, model training, model registry management, deployment, prediction and visualization. Hopsworks is used as the central platform for managing datasets, feature stores, and model registries.


7. Visualize in Streamlit
6. Predict via FastAPI
5. Select Best Model (Highest R²)
4. Register in Hopsworks
3. Train Models (XGBoost, Ridge, LSTM)
2. Process Features (AQI Computation, Encoding)
1. Fetch Raw Data (Open-Meteo APIs)
Data → Models → API → Dashboard

3. Data Fetching

The system fetches hourly air pollution and weather data for Lahore, Pakistan (Latitude 31.558, Longitude 74.3507) between May 2024 and November 2025 using Open-Meteo APIs. **Data Sources:** - Weather API → Relative Humidity - Air Quality API → PM10, PM2.5, Ozone (O3), Nitrogen Dioxide (NO2), Sulphur Dioxide (SO2), Carbon Monoxide (CO) The merged dataset is stored in the Hopsworks Feature Store as **'aqi_raw_weather_pollution (v1)'**.

4. Feature Engineering

This stage transforms raw data into model-ready inputs:

- Performed EDA steps (Analyzing Correlation, Temporal Feature Importance Analysis, feature importance, feature importance evolution over time for tree-based models etc.)
- Compute AQI subindices → final AQI.
- Checked for NaN, Missing values etc.
- Implement Early Stopping, Tuned hyper parameters to prevent Over fitting
- Did Scaling and Temporal Encoding. Applied RobustScaler to normalize numeric variables.

Temporal Feature Engineering:

Extracted **time-based features** from timestamp

- hour → cyclic encoding (hour_sin, hour_cos) for capturing periodic patterns.
- day_of_week → one-hot encoded (dow_0, dow_1, …).
- month → converted to **season** (winter, spring, summer, autumn) and one-hot encoded (dropping one dummy to avoid multicollinearity).

The final feature set includes timestamp, humidity, pollutants, encoded temporal features, and the computed AQI target variable. They are stored in feature group aqi_hourly_features (version 4) in Hopsworks.

5. Model Development

Three predictive models were trained using the processed features:

Model	Key Strength
XGBoost	Handles non-linear relationships
Ridge Regression	Interpretable and simple
LSTM	Captures temporal dependencies

Training involved time-series cross-validation with 5 folds, and models were evaluated on RMSE, MAE, and R^2 metrics. LSTM used a 24-hour sliding window for sequential learning.

6. Back-End API

A FastAPI backend is used to serve AQI predictions. It automatically connects to Hopsworks, retrieves all registered models, compares R^2 scores, and selects the best one dynamically.

Below are the Routes
Endpoints: - / → Root summary with current best model - /predict → Predict current AQI from latest feature store data - /forecast_3day → Autoregressive forecast for next 3 days

7. Training Methodology

Time-series cross-validation, preventing data leakage Temporal train-test split (80-20 ratio) Automated model selection based on R^2 thresholds

8. Visualization and Front-End

A Streamlit dashboard provides two major views: 1. **Current AQI** → Real-time AQI value with colored category box (Good, Moderate, Unhealthy, etc.) 2. **3-Day Forecast** → Interactive bar chart using Plotly It communicates with the FastAPI backend hosted on Render Cloud.

9. Evaluation

Model performance metrics stored in Hopsworks show XGBoost outperforming others in R^2 score.

Selected model with highest R^2 score.

Model	RMSE	MAE	R^2
XGBoost	Low=7.1839	Low=4.05791	Highest=0.99201
Ridge Regression	Moderate36.3860	Moderate=27.9578	Lower=0.795
LSTM	Comparable=0.04534	slightly higher=0.03134	High=0.9320

10.Deployment & Hosting

Backend: FastAPI deployed on Render Cloud.
Link: https://aqipredictor-gmnlqqv2zx3hsjqcamaq9l.streamlit.app/
Frontend: Streamlit dashboard.
Link: https://aqi-fastapi-backend.onrender.com/
Feature Store & Registry: Hopsworks (project, versioning, and model metrics). All components are connected through APIs for seamless automation.

11. Conclusion

The AQI Prediction System provides a comprehensive ML pipeline, spanning from data collection to deployment. By combining strong feature engineering with reliable modeling and automated workflows, it ensures accurate, stable predictions. Interactive visualizations make the platform user-friendly and suitable for ongoing air quality monitoring.