Project Report: Karachi AQI Prediction Bot

1. Overview

This project **forecasts Karachi's Air Quality Index (AQI) using real-time pollutant and weather data via a fully automated ML pipeline**.
It handles **live data fetching, preprocessing, feature engineering, cloud storage (Hopsworks), and automated retraining (GitHub Actions CI/CD), visualized through a Streamlit dashboard**.

AQI is computed per **U.S. EPA standards (May 2016)** using truncation, breakpoint mapping, and linear interpolation for $PM_{2.5}$, PM_{10}, CO, NO_2, SO_2, and O_3.
Open-Meteo APIs (Air Quality, Weather Forecast, and Historical Archive) were chosen for reliability, pollutant coverage, and hourly forecasts.

---

2. Data Collection & Preprocessing

Scripts: fetch_data.py, process_data.py, merge_features.py, backfill_data.py.
• Merged pollutant and weather datasets (Jan 2024–Nov 2025) into a unified hourly DataFrame with standardized units (µg/m³).
• Outliers capped (1st–99th percentile) for $PM_{2.5}$, PM_{10}, CO; NO_2 retained for genuine peaks.
• Correlation (r=0.53) showed $PM_{2.5}$ & PM_{10} as key AQI drivers.
• Observed winter buildup, dusty summer peaks, and spring ozone rise.
→ Clean dataset ready for AQI computation.

---

3. AQI Computation & Feature Engineering

AQI (aqi_utils.py): Converted pollutant concentrations to EPA units and calculated sub-indexes; final AQI = max(sub-indexes).
Features (process_features.py):
• Time: hour, weekday, month, cyclic encodings.
• Trends: 24h rolling mean, 1h lag.
• Derived: $PM_{2.5}/PM_{10}$, temp-humidity ratio, wind dispersion, high-pollution flag.
• Missing values interpolated.
→ Final dataset: 23 engineered features.

---

4. Feature Refinement

Analyzed 15,744 records, 40 variables.
• Removed redundant/highly correlated (>0.9) features.
• Retained $PM_{2.5}$, PM_{10}, SO_2, O_3 as core predictors.
• Found AQI highest in Dec–Jan mornings due to temperature inversion.
→ Exported as final_selected_features.csv.

---

5. Cloud Integration (Hopsworks)

Scripts: upload_to_hopswork.py, run_feature_pipeline.py.
• Secure upload via API key to versioned Feature Group aqi_features_v1.
• Daily updates, no duplication (timestamp-based).
• Local CSV saving disabled for CI/CD.
→ Chronologically validated (Jan 2024–Nov 2025).

---

6. Model Training & Evaluation

Script: train_model.py.
• Data fetched from Hopsworks; dropped leakage-prone features; added ±5% Gaussian noise.
• Time-based 80/20 split.
Models: Ridge, Random Forest, XGBoost.

Model	Train RMSE	Test RMSE	R^2	Remarks
Ridge	12.48	13.21	0.91	Underfit
XGBoost	5.73	8.21	0.97	High variance
Random Forest	**3.91**	**6.59**	**0.99**	✓ Best model

→ Saved as best_model_random_forest.pkl.

---

7. CI/CD Automation

Implemented via GitHub Actions:

Workflow	File	Time (PKT)	Function
Feature Pipeline	feature_pipeline.yml	8:10 AM	Fetch, process, upload to Hopsworks
Training Pipeline	training_pipeline.yml	8:30 AM	Retrain ML model

→ Ensures daily updates and retraining automatically.

---

8. Streamlit Dashboard (Locally for now)

Files: app.py, utils.py.
• Loads latest model; predicts AQI for current + next 3 days.
• Displays 3-day bar chart, hourly line chart, and AQI category cards (EPA breakpoints).
→ Real-time, interactive visualization of Karachi's air quality.

---

9. Challenges & Fixes

Issue	Resolution
Future-day overlap	Adjusted UTC window
Duplicate entries	Enforced datetime_str primary key
Workflow timing mismatch	Updated cron jobs
API/dependency errors	Added confluent-kafka, version-pinned hopsworks[python]
Multi-day fetch	Created aqi_features_v2 and reuploaded clean dataset

---

10. Conclusion

The Karachi AQI Prediction Bot delivers a production-grade, automated MLOps pipeline that:

- Fetches & cleans live data per EPA AQI standards.
- Generates engineered features and uploads them to Hopsworks.
- Trains and retrains models via CI/CD.
- Visualizes predictions through Streamlit.

The system is stable, scalable, and extendable to other cities. SHAP-based interpretability is planned for future work. (couldn't do cause of time-constraints and exam clash)

Deliverables

- Automated data ingestion & cloud integration
- Random Forest AQI model (R^2=0.99)
- Daily CI/CD retraining
- Interactive Streamlit dashboard
- Scalable production-ready MLOps framework
