**Internship Project Report**

**Submitted by (Intern): Areeba Ahmed** **Submitted to: 10Pearls**

## Topic: AQI Predictor

### 1. Data Collection & Preprocessing

The pipeline begins with collecting hourly AQI and weather data for Karachi using the OpenWeather API. The data includes temperature, humidity, wind speed, and AQI values. All records are stored in a Hopsworks feature group called karachi_weather_hourly. For model training, we extracted the last 30 days of hourly data (approx. 720 rows) using a script (last_30_days.py).

Once the data was loaded, we performed null checks to ensure data integrity. Any rows with missing AQI or weather values were dropped, since incomplete lag sequences would reduce prediction accuracy. We also verified that timestamps were continuous and hourly spaced, which is critical for time-series modeling. This preprocessing ensured that the dataset was clean, reliable, and ready for feature engineering.

### 2. Exploratory Data Analysis (EDA)

EDA was conducted to understand AQI distribution and its relationship with weather variables. We observed that AQI values from OpenWeather are provided on a **scaled 1–5 range** (not the standard 0–500 scale).

Correlation checks revealed that while weather features (temperature, humidity, wind speed) had some influence, the **strongest predictor of AQI was its own past values**. This insight guided our feature selection strategy.

### 3. Feature Selection, Lags, and Target

Based on EDA, we selected **lag features** as the primary predictors. Specifically, we created 72 lag features, representing the AQI values from the past 72 hours.

- **Why lags?** AQI is highly autocorrelated — recent values strongly influence near-future conditions due to pollutant accumulation and slow environmental changes. By including the last 72 hours, the model captures short-term cycles and trends.

- **Target variable:** The AQI value one hour ahead. This was chosen because the forecasting goal is to predict the next 72 hours step by step. Aligning lag features with the next hour's AQI allows the model to learn temporal dependencies.

**4. Model Training & Evaluation**

We experimented with multiple models: **Random Forest, XGBoost, and Linear Regression**. Evaluation metrics included RMSE, MAPE, accuracy, and F1 score (for AQI categories).

- **Random Forest Regressor (version 13)** was selected as the final model.
- Performance:

  Accuracy: **0.96**

  R^2: **0.743**

  RMSE: **0.31**

Random Forest was chosen because it provided the best balance of accuracy, robustness, and generalization. The model was registered in Hopsworks Model Registry.

**5. Deployment with Flask**

For deployment, we built a **Flask web app** that integrates the trained model with a user interface. The app workflow is:

1. Load the latest 72 AQI values from Hopsworks.
2. Run the Random Forest model to forecast the next 72 hours.
3. Display results in a clean UI with:
   - Hourly forecast table
   - Daily averages with AQI categories
   - A chart showing past and predicted AQI values

**GitHub repo link:**

https://github.com/areeba61/karachi-aqi-pipeline