Project: Pearls AQI Predictor
Domain: Data Sciences

Predict the Air Quality Index (AQI) in your city in the next 3 days, using a 100% serverless stack. This project involves building an end-to-end machine learning pipeline for AQI forecasting with automated data collection, feature engineering, model training, and real-time predictions through a web dashboard.

Internee: Anoosha Khalid
Computer Systems Engineering
NED UNIVERSITY OF ENG & TECH

Submitted to 10 Pearls

Abstract

Air pollution is a growing global concern, directly impacting public health and quality of life. This project, **Pearls AQI Predictor**, is a fully serverless, end-to-end machine learning system that predicts the **Air Quality Index (AQI) for the next three days** for Karachi.

The system automates **data ingestion, feature engineering, training, deployment, and real-time predictions** using modern MLOps practices. It integrates real-time weather and pollutant data, stores engineered features in a feature store (Hopsworks), retrains models automatically using GitHub Actions, and presents predictions through an interactive Streamlit web application.

1.Introduction

Air pollution causes severe health problems such as asthma, heart disease, and respiratory infections. Monitoring air quality allows people and authorities to take preventive actions.

Traditional AQI monitoring systems focus only on real-time reporting. However, **forecasting future AQI values** allows governments, healthcare systems, and citizens to prepare in advance. Pearls AQI Predictor uses **machine learning + MLOps** to predict AQI levels for the next 72 hours using real-time environmental data and historical patterns.

2.Objective

The major objective of this project was:

- Predict AQI for the next 3 days
- Automate data collection, training, and deployment
- Store reusable features in a feature store
- Provide a real-time prediction dashboard
- Follow MLOps best practices

3.System Overview

The system consists of five main components:

1. **Data Ingestion Layer** – Collects real-time weather and pollutant data from external APIs.
2. **Feature Pipeline** – Transforms raw data into meaningful features.
3. **Feature Store (Hopsworks)** – Stores and versions all feature.
4. **Training Pipeline** – Trains three models and stores the best one in a registry.
5. **Prediction Dashboard** – Displays real-time predictions using Streamlit.

4.Data Sources

- **AQICN**

This provides hourly readings, which are used both for training and real-time inference.

5.Feature Engineering Pipeline

Raw data is not directly suitable for machine learning. Therefore, a feature pipeline was developed to transform raw values into meaningful indicators.
This includes:

- Time-based features (hour, day, month)
- Rolling averages of AQI
- Lag features from previous hours
- Rate of change of AQI

These features are computed automatically and stored in the feature store.

```
df['aqi'] = df['pm25'].apply(calculate_aqi_pm25)
df = df.sort_values('date')

df['aqi_lag_1'] = df['aqi'].shift(1)
df['aqi_lag_3'] = df['aqi'].shift(3)
df['aqi_change_rate'] = df['aqi'].pct_change()

df['month'] = df['date'].dt.month
df['day_of_week'] = df['date'].dt.dayofweek
df['is_weekend'] = (df['day_of_week'] >= 5).astype(int)
```

```
# List of feature names
feature_names = [f.name for f in fg.features]
print(feature_names)
```

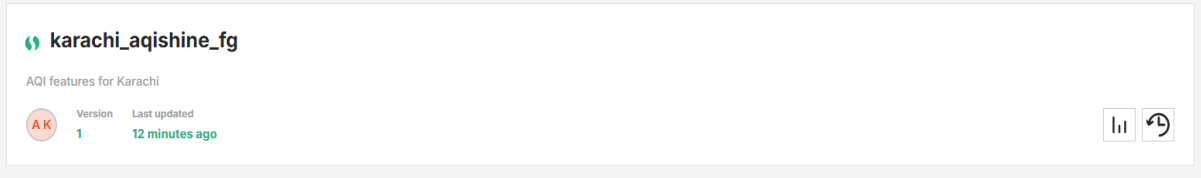
```
['date', 'pm25', 'aqi', 'aqi_lag_1', 'aqi_lag_3', 'aqi_change_rate', 'month', 'day_of_week', 'is_weekend']
```

Feature Store Using Hopsworks

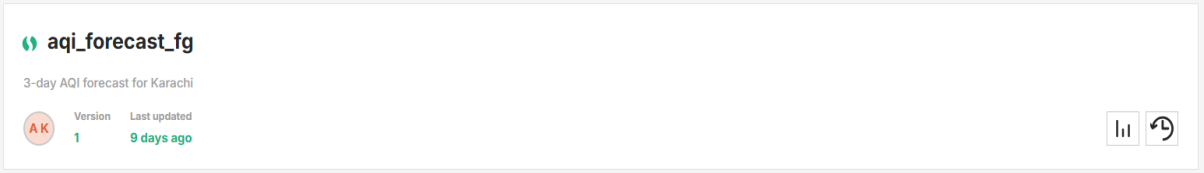
Hopsworks is used to store and manage all features. It ensures:

- Version control of features
- Consistency between training and inference
- Centralized access for pipelines

This eliminates data leakage and feature mismatches.



To store all the features



To store the prediction values, fetch by the training

Historical Data Backfilling

To train the models, historical data was generated by running the feature pipeline for previous timestamps. This created a large dataset that represents past air quality trends.

Before running the feature pipeline

After running the feature pipeline

6.Model Training Pipeline

The training pipeline fetches data from Hopsworks and trains multiple models, including:

- Random Forest
- Ridge Regression
- TensorFlow Neural Network

Model Selection and Registration Strategy

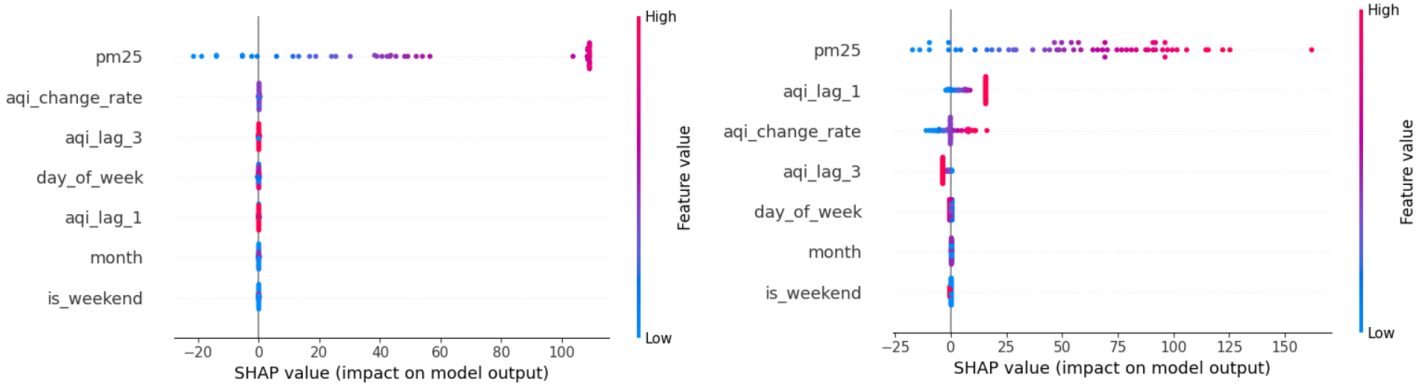
After training, each model was tested on the same unseen test dataset. Their predictions were evaluated using three standard regression metrics:

- Mean Squared Error **(MSE)** – measures the average squared difference between predicted and actual AQI values.
- Mean Absolute Error **(MAE)** – measures the average absolute prediction error.
- **R^2** Score – indicates how well the model explains the variance in AQI values.

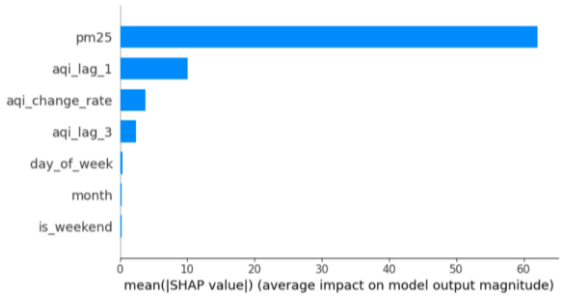
To select the best model, the system automatically identifies the model with the highest R^2 score, since a higher R^2 value means better predictive power and generalization.

Explainability with SHAP

After selecting the best-performing model, SHAP is used to analyze how each feature contributes to the predicted AQI.



Initial : Rainforest Model & Ridge



Initial training:

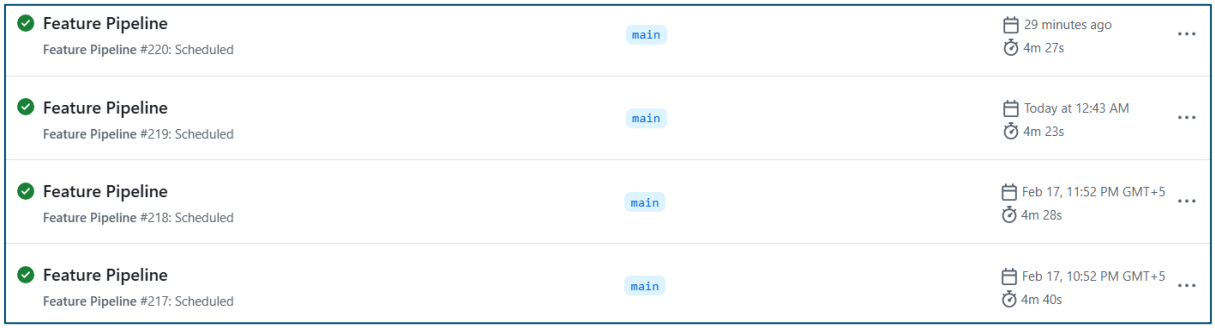
Model	MSE	R2	MAE
Random Forest	0.94	1.00	0.49
Ridge Regression	263.88	0.88	11.55
Neural Network	2331.39	-0.06	38.82

7.Automation and CI/CD

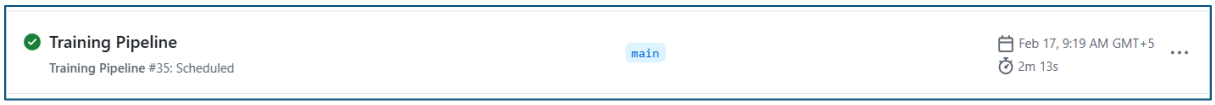
All pipelines are automated using **GitHub Actions**.

- Feature pipeline runs every hour
- Training pipeline runs daily

This ensures the model stays updated without manual intervention.



Feature Automation



Training Automation

8.Challenges Faced

During development, several challenges were encountered:

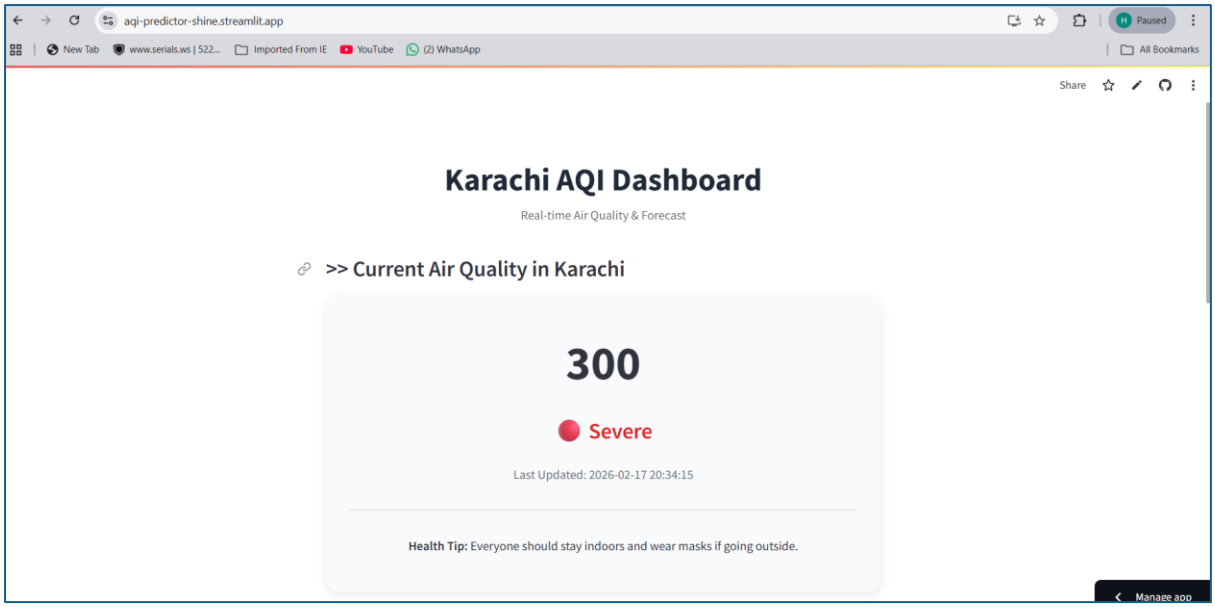
- GitHub Actions authentication failures
- Hopsworks is not compatible with VsCode
- Dependency version conflicts
- Feature store schema mismatches

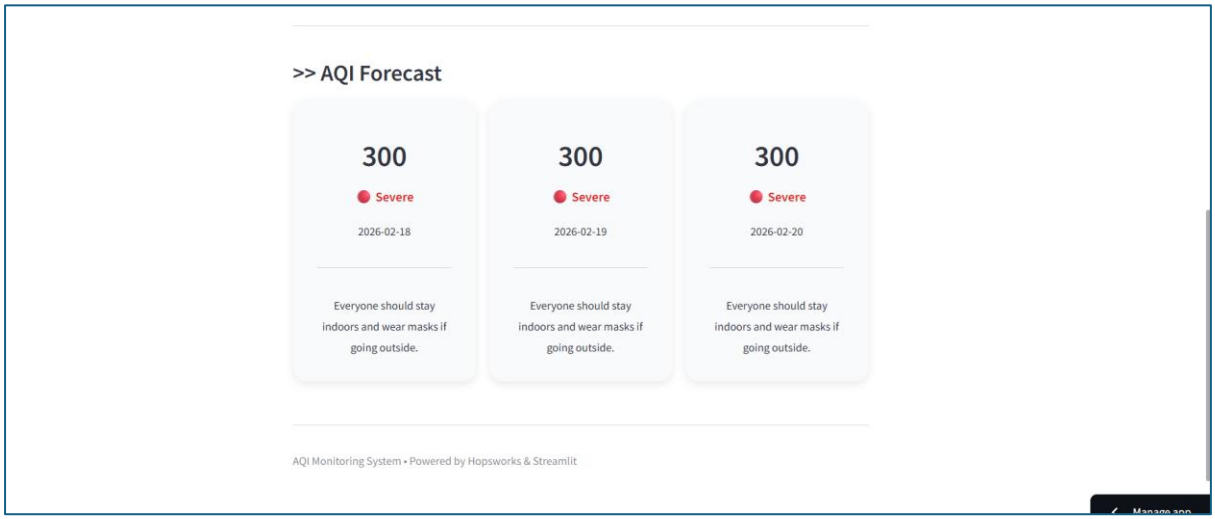
These issues required debugging YAML workflows, secrets, and service permissions.

9.Web Application

The Streamlit dashboard allows users to:

- View current AQI
- See predictions for the next three days





10.Results

The system successfully predicts AQI values and updates models automatically. The automated pipelines reduce manual effort and improve reliability.

11.Limitations

- Prediction accuracy depends on API data quality
- Sudden weather changes affect forecasts
- Limited to Karachi city

Pearls AQI Predictor demonstrates a real-world machine learning system that integrates automation, feature management, and cloud deployment. It shows how predictive models can be maintained in production environments.