Pearls AQI Predictor

Project Report

Submitted by: Hajira Imran

Domain: Data Sciences

Dated: February 10, 2026

Islamabad, Pakistan

Contents

Islamabad AQI Prediction System 3

Project Objective 3

Data Acquisition & Ingestion 3

Data Quality Audit & Cleaning 3

Key Insights 4

AQI over Time 4

PM2.5 vs AQI 5

AQI by Hour of Day 5

AQI by Weekday 6

AQI vs Previous AQI 7

AQI change rate Over Time 7

PM2.5 Rolling Average 8

Feature Engineering 9

Model Training & Interpretation 9

Automation via GitHub Actions Pipeline 9

Feature Pipeline Automation 9

Model Training Pipeline 10

SHAP analysis 11

System Implementation & Challenges 11

Conclusion 12

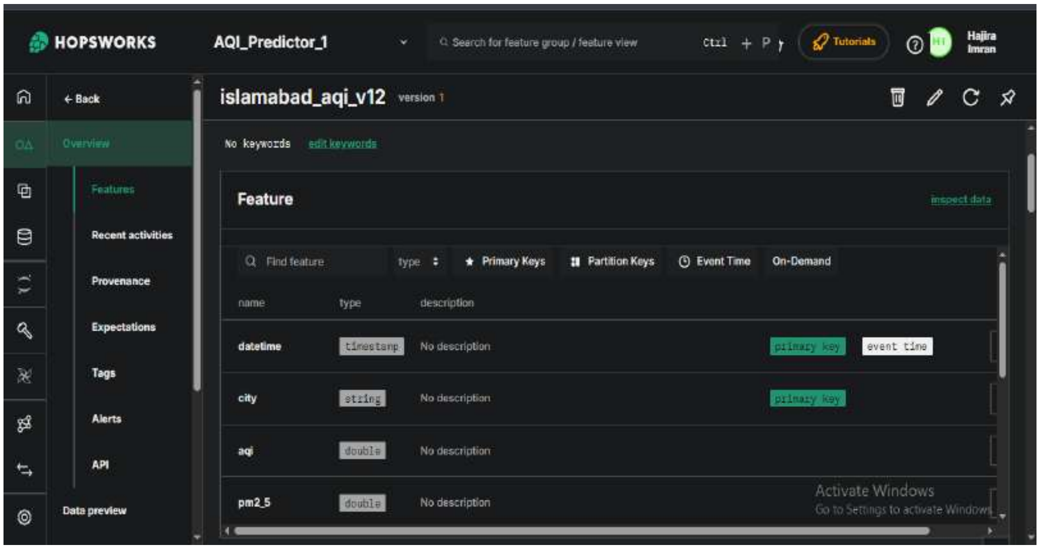
Islamabad AQI Prediction System

Project Objective

The goal was to build a machine learning pipeline capable of predicting Air Quality Index (AQI) levels for Islamabad using historical and real-time data fetched from the Hopsworks Feature Store.

Data Acquisition & Ingestion

- **Source**: Data was retrieved from a remote feature store hosted on Hopsworks.
- **Temporal Scope**: The data spans multiple months, with the most recent records appearing in **January 2026**.



islamabad_aqi_v12 version 1

No keywords edit keywords

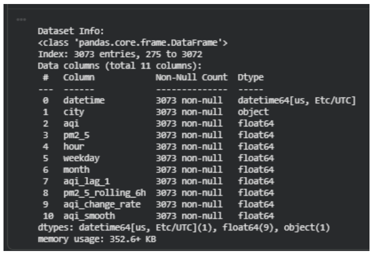
Feature inspect data

Find feature | type | Primary Keys | Partition Keys | Event Time | On-Demand

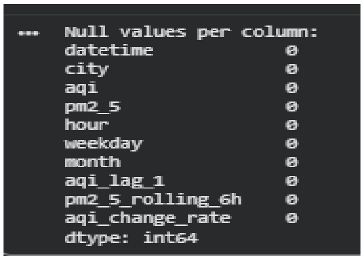
name	type	description		
datetime	timestamp	No description	primary key	event time
city	string	No description	primary key	
aqi	double	No description		
pm2_5	double	No description		

Data Quality Audit & Cleaning

- **Completeness**: A systematic audit confirmed **zero missing values** across all 11 columns, including target variables and engineered features.
- **Data Consistency**: The non-null count was verified for all features, ensuring stability for model training.
- **Dtypes Verification**: All essential features were confirmed to be in float64 format, ensuring compatibility with mathematical modeling.



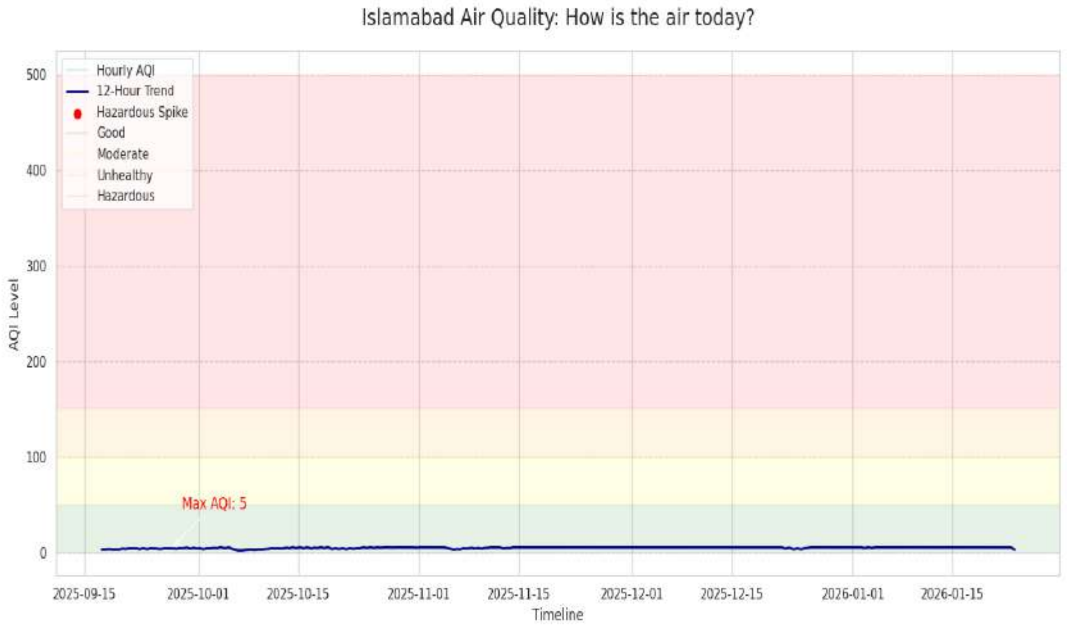
```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
Index: 3073 entries, 275 to 3072
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   datetime           3073 non-null   datetime64[us, Etc/UTC]
 1   city               3073 non-null   object
 2   aqi                3073 non-null   float64
 3   pm2_5              3073 non-null   float64
 4   hour               3073 non-null   float64
 5   weekday            3073 non-null   float64
 6   month              3073 non-null   float64
 7   aqi_lag_1          3073 non-null   float64
 8   pm2_5_rolling_6h   3073 non-null   float64
 9   aqi_change_rate    3073 non-null   float64
 10  aqi_smooth         3073 non-null   float64
dtypes: datetime64[us, Etc/UTC](1), float64(9), object(1)
memory usage: 352.6+ KB
```



```
Null values per column:
datetime            0
city                0
aqi                 0
pm2_5               0
hour                0
weekday             0
month               0
aqi_lag_1           0
pm2_5_rolling_6h    0
aqi_change_rate     0
dtype: int64
```

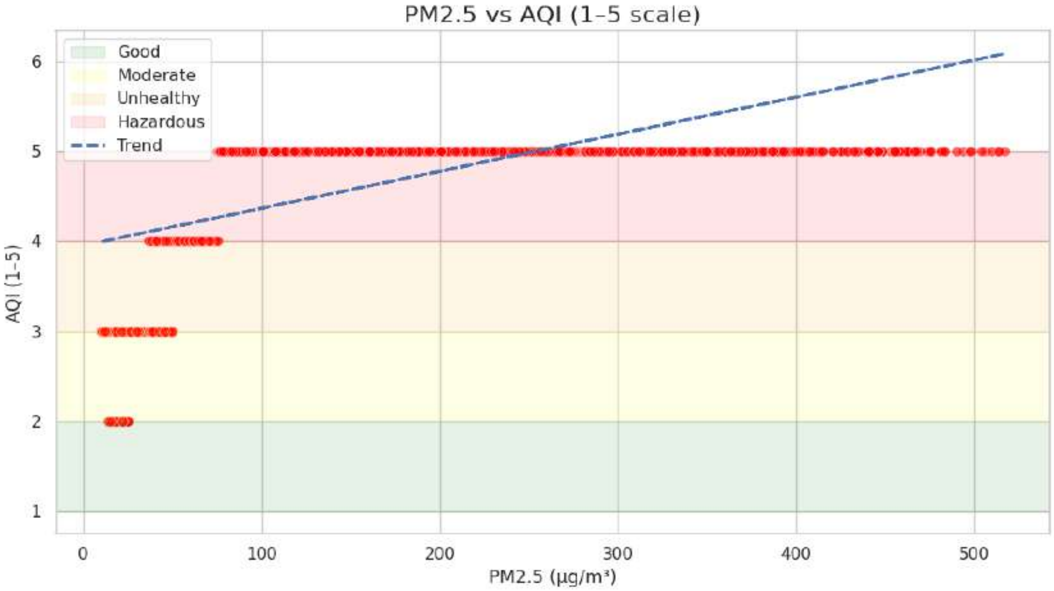
Key Insights

AQI over Time



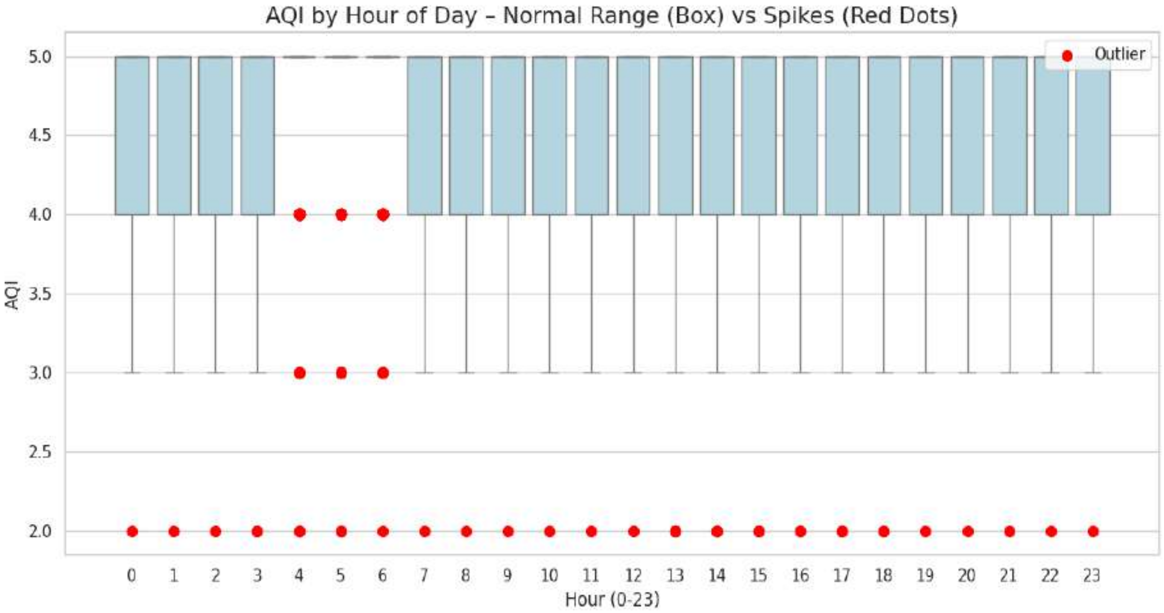
- **AQI Trends:** Mostly stable over time with occasional spikes.
- **Rolling Average:** 12-hour smoothing shows overall air quality trends clearly.
- **Air Quality Zones:** Good, Moderate, Unhealthy, and Hazardous zones highlight risk levels.
- **PM2.5 Impact:** AQI strongly influenced by PM2.5; spikes correspond to AQI changes.
- **Temporal Features:** Lagged AQI and rolling PM2.5 capture patterns for prediction.

PM2.5 vs AQI



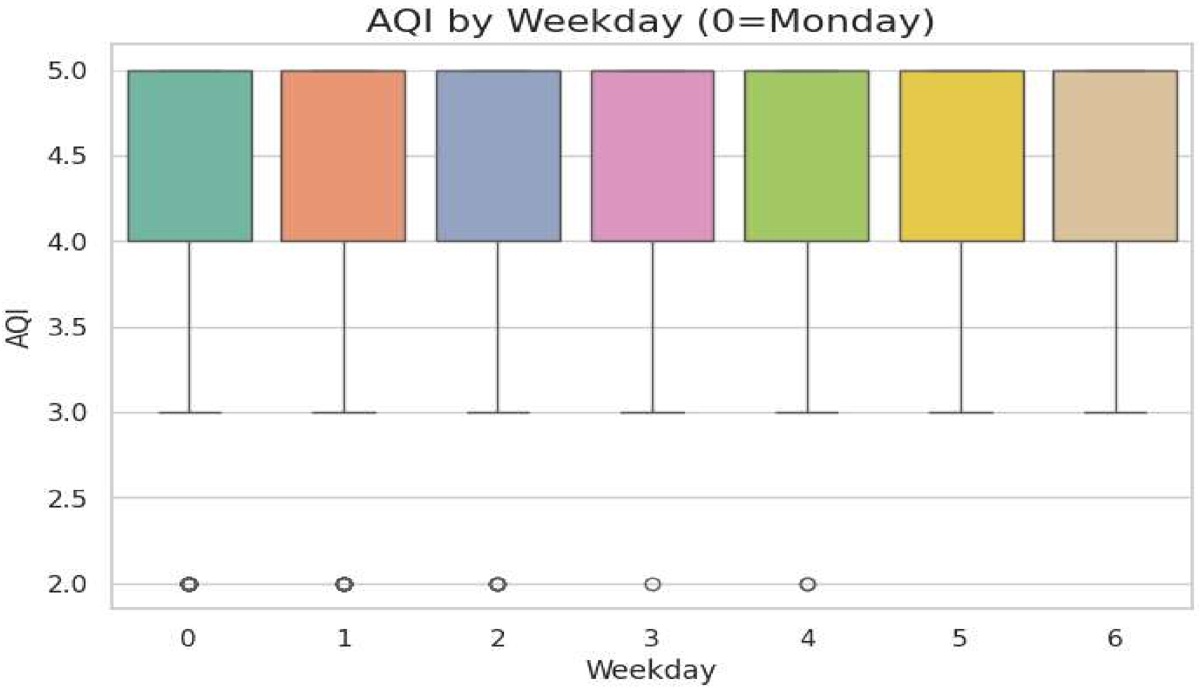
- Most points are clustered in the lower AQI range (Good to Moderate), so air is often acceptable.
- Higher PM2.5 values are associated with higher AQI confirming PM2.5 is a major contributor to air pollution.
- The trend line gives a simple, easy-to-understand summary for anyone: **more particles → worse air quality**.

AQI by Hour of Day



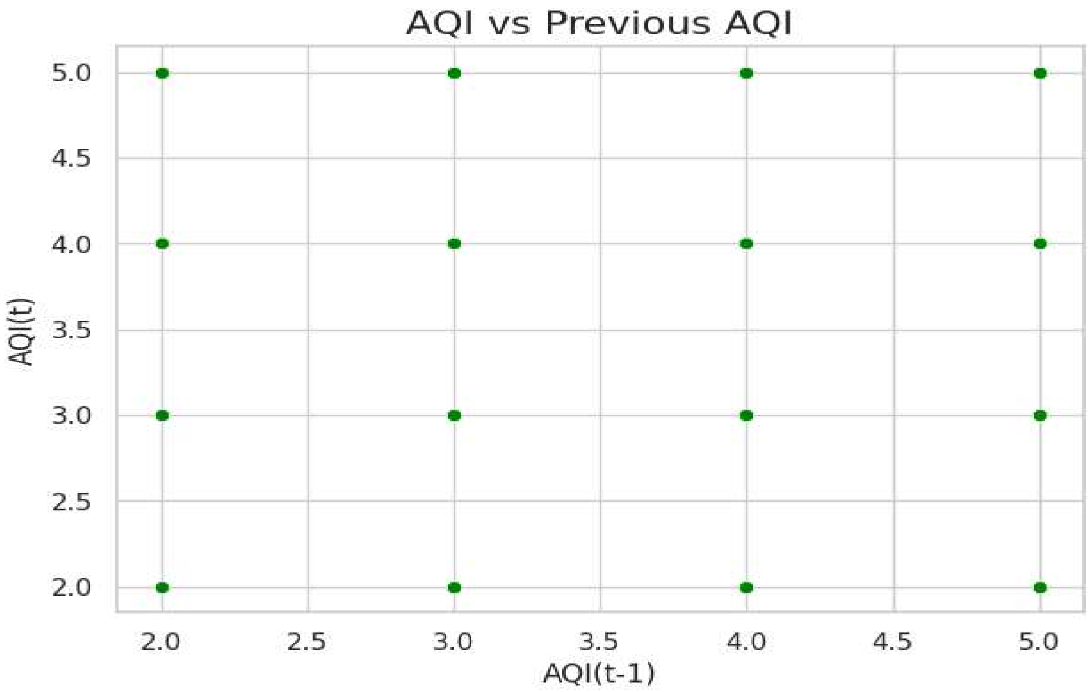
- **Hourly Variation:** The boxplot shows how AQI levels fluctuate throughout the day. Most hours have AQI within a narrow, “normal” range (light blue boxes).
- **Spikes/Outliers:** Red dots indicate hours with unusually high or low AQI. These spikes show moments when air quality temporarily worsened or improved.
- **Peak Hours:** By looking at the box positions and whiskers, certain hours (like early morning or late evening) may have higher median AQI compared to others.
- **Pattern Insight:** This visualization helps identify time periods where pollution is likely to peak, useful for preventive measures or public alerts.

AQI by Weekday



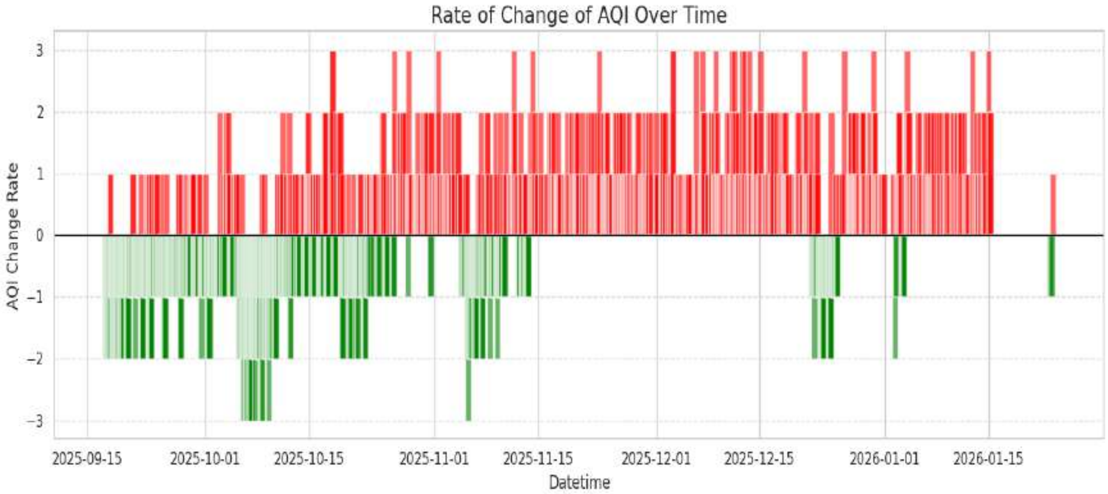
- The plot shows **air quality (AQI) for each day of the week** (0 = Monday, 6 = Sunday).
- The **middle line in each box** represents the typical AQI for that day.
- **Boxes indicate variability**: taller boxes mean air quality changes more during that day.
- **Small dots** are unusual spikes or drops in AQI, called **outliers**.
- From the plot, we can **identify which weekdays usually have better or worse air quality**.
- This helps in **planning outdoor activities** or issuing **health advisories** on days with poor air quality.

AQI vs Previous AQI



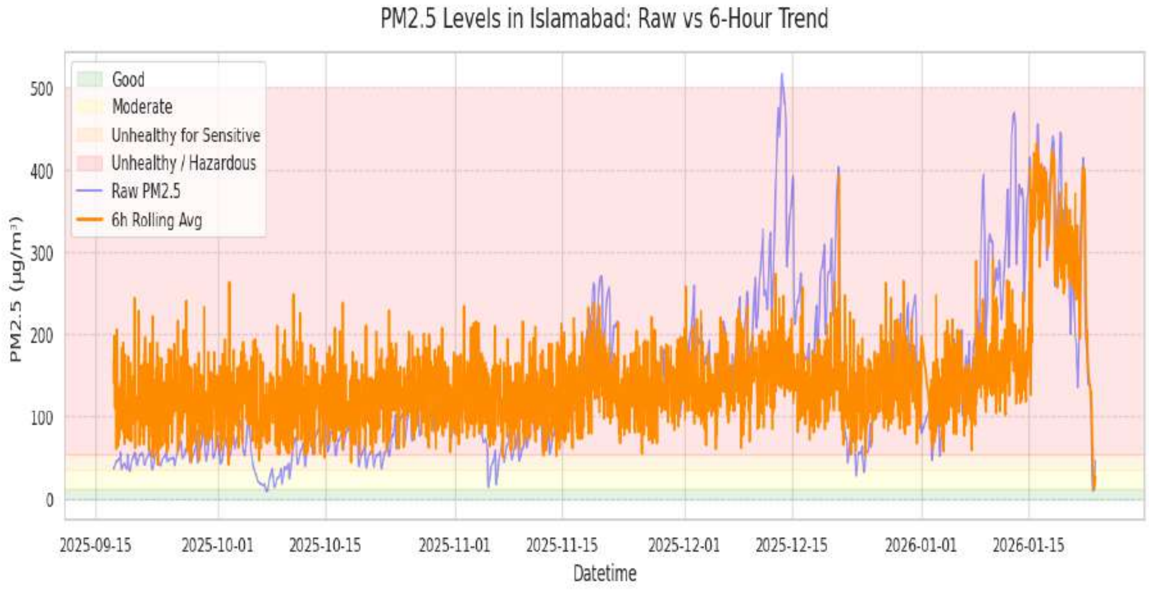
- This scatter plot shows the **relationship between the AQI at a given hour (t) and the AQI from the previous hour (t-1)**.
- Each point represents one hour of data.
- If the points are **close to a straight diagonal line**, it means **air quality doesn't change drastically hour-to-hour**.
- It helps us **see patterns over time** and indicates that AQI is somewhat **predictable based on the previous hour**.
- Useful for **time-series modeling**, because the previous AQI is a strong feature for predicting the next hour's AQI.

AQI change rate Over Time



- Shows how AQI changes from one hour to the next (AQI change rate).
- **Red bars** indicate AQI increased → air quality worsened.
- **Green bars** indicate AQI decreased → air quality improved.
- **Black horizontal line at 0** is the reference for no change.
- Highlights periods of sudden pollution spikes or improvement.
- Helps understand the **volatility** of air quality over time.

PM2.5 Rolling Average



- The graph displays **PM2.5 concentrations over time** in Islamabad.
- The **raw PM2.5 values** (light line) fluctuate significantly hour by hour.
- The **6-hour rolling average** (orange line) smooths out short-term fluctuations, making the overall trend easier to see.
- Peaks in PM2.5 indicate periods of **higher pollution**, while lower points show cleaner air.
- The rolling average helps identify **persistent pollution trends** rather than focusing on temporary spikes.
- This analysis can guide **feature engineering**, such as creating rolling features for modeling AQI prediction.

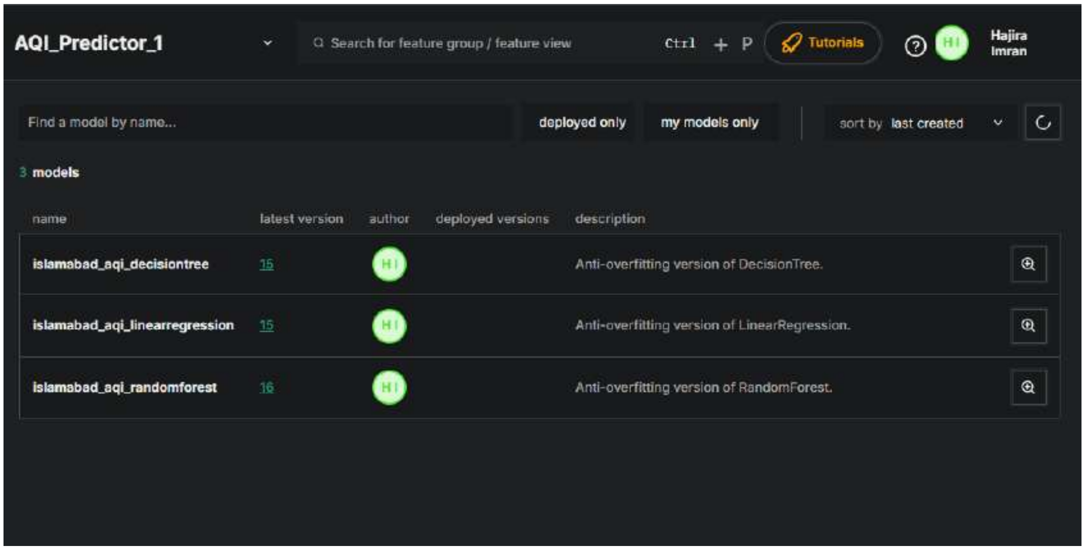
Feature Engineering

To deal with time-series categorical data, we transformed the raw datetime column into numerical features:

- **Cyclical Features**: We extracted **Hour (0-23)**, **Weekday (0-6)**, and **Month** to capture daily and weekly cycles.
- **Temporal Lag**: An aqi_lag_1 feature was created to give the model context of the previous hour's air quality.
- **Rolling Metrics**: A pm2_5_rolling_6h feature was calculated to smooth out short-term fluctuations and identify broader trends

Model Training & Interpretation

- **Target Variable**: The model predicts a normalized AQI score ranging from **0.0 to 5.0**.
- **Interpretability**: Through boxplot analysis, we verified that the model accounts for variances across different **Hours** of the day, proving that temporal features were successfully integrated.



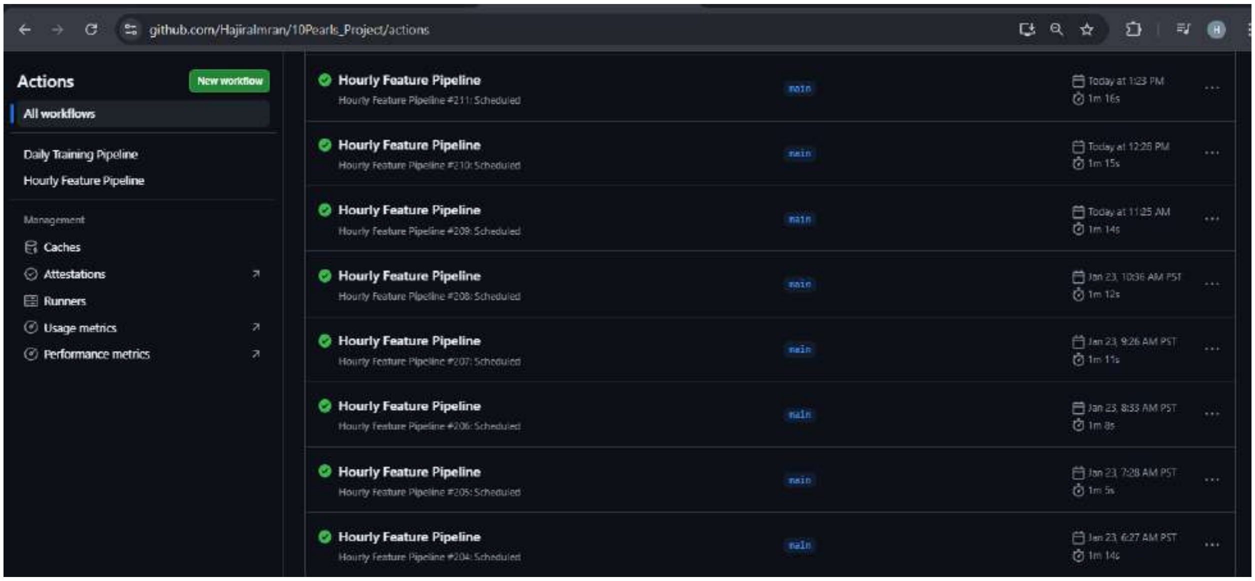
3 models

name	latest version	author	deployed versions	description
islamabad_aqi_decisiontree	15	HI		Anti-overfitting version of DecisionTree.
islamabad_aqi_linearregression	15	HI		Anti-overfitting version of LinearRegression.
islamabad_aqi_randomforest	16	HI		Anti-overfitting version of RandomForest.

Automation via GitHub Actions Pipeline

Feature Pipeline Automation

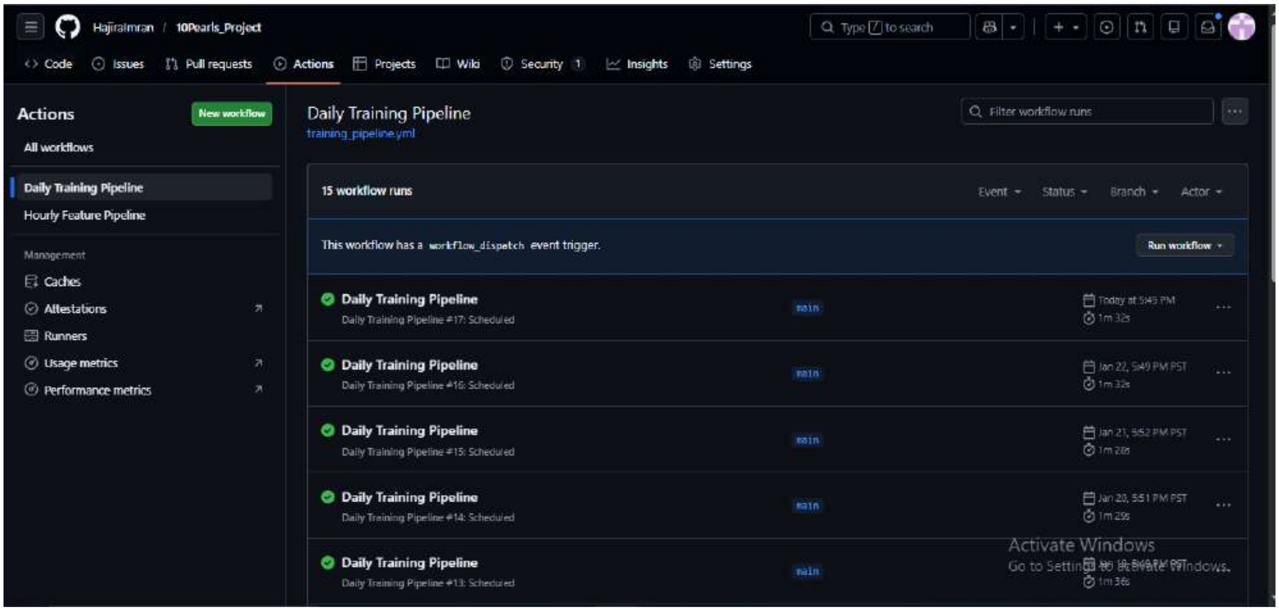
- **Workflow**: A GitHub Action is scheduled (e.g., every hour) to trigger the feature_pipeline.py script.
- **Action**: It connects to the OpenWeather API, fetches the latest Islamabad AQI data, and performs the numerical transformations you've implemented (like calculating aqi_lag_1).
- **Output**: The fresh features are automatically pushed to the **Hopsworks Feature Store**.



Hourly Feature Pipeline Hourly Feature Pipeline #211: Scheduled	main	Today at 1:23 PM 1m 16s
Hourly Feature Pipeline Hourly Feature Pipeline #210: Scheduled	main	Today at 12:28 PM 1m 15s
Hourly Feature Pipeline Hourly Feature Pipeline #209: Scheduled	main	Today at 11:25 AM 1m 14s
Hourly Feature Pipeline Hourly Feature Pipeline #208: Scheduled	main	Jan 23, 10:36 AM PST 1m 12s
Hourly Feature Pipeline Hourly Feature Pipeline #207: Scheduled	main	Jan 23, 9:26 AM PST 1m 11s
Hourly Feature Pipeline Hourly Feature Pipeline #206: Scheduled	main	Jan 23, 8:33 AM PST 1m 8s
Hourly Feature Pipeline Hourly Feature Pipeline #205: Scheduled	main	Jan 23, 7:28 AM PST 1m 5s
Hourly Feature Pipeline Hourly Feature Pipeline #204: Scheduled	main	Jan 23, 6:27 AM PST 1m 14s

Model Training Pipeline

- **Daily Automation**: A GitHub Actions pipeline is scheduled to run every night at midnight to automatically execute the train_models.py script.
- **Environment & Security**: The pipeline runs on an ubuntu-latest virtual environment and uses GitHub Secrets to securely handle API keys like HOPSWORKS_KEY and OPENWEATHER_KEY.
- **Continuous Learning**: By installing dependencies and running the training script daily, the system ensures the model is consistently updated with the latest data from the feature store.



Daily Training Pipeline

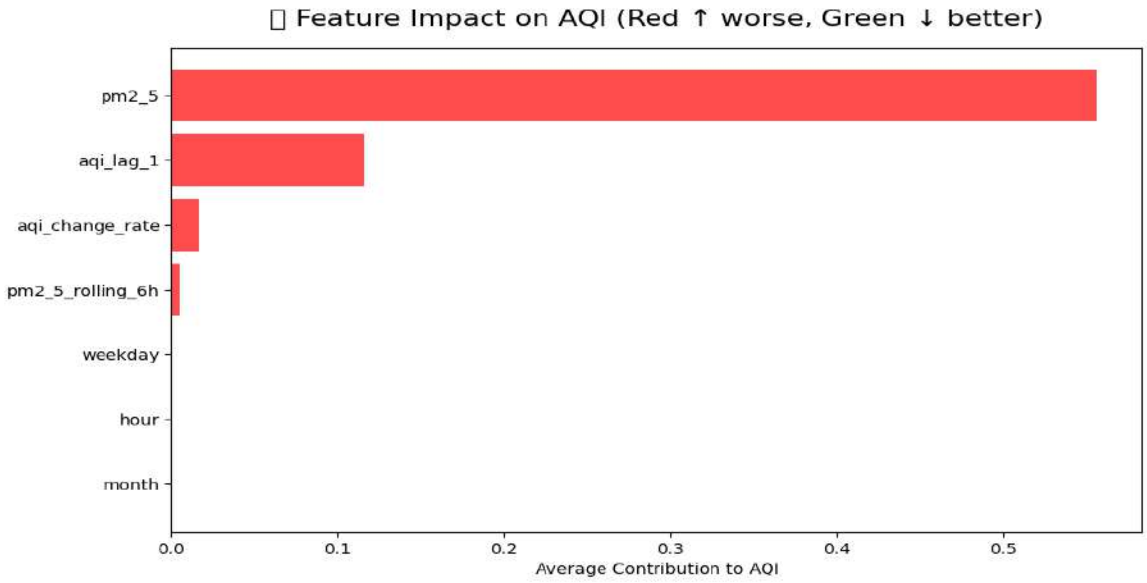
training_pipeline.yml

15 workflow runs

This workflow has a `workflow_dispatch` event trigger.

Daily Training Pipeline Daily Training Pipeline #17: Scheduled	main	Today at 5:45 PM 1m 32s
Daily Training Pipeline Daily Training Pipeline #16: Scheduled	main	Jan 22, 5:49 PM PST 1m 32s
Daily Training Pipeline Daily Training Pipeline #15: Scheduled	main	Jan 21, 5:52 PM PST 1m 28s
Daily Training Pipeline Daily Training Pipeline #14: Scheduled	main	Jan 20, 5:51 PM PST 1m 29s
Daily Training Pipeline Daily Training Pipeline #13: Scheduled	main	[illegible] 1m 36s

SHAP analysis



- SHAP was applied on the Random Forest model (v23) to measure feature importance for AQI prediction.
- **PM2.5** is the most influential feature: higher PM2.5 levels increase AQI (worse air quality).
- **Previous hour AQI (aqi_lag_1)** also significantly affects current AQI, showing persistence in air quality patterns.
- **Time-based features** like hour and weekday have moderate impact on AQI variations.
- **Rolling averages (pm2_5_rolling_6h)** and **AQI change rate (aqi_change_rate)** contribute, but less than PM2.5 and lagged AQI.
- SHAP visualization uses **red bars for features that increase AQI** (worsen air quality) and **green bars for features that decrease AQI** (improve air quality).
- This analysis helps identify which factors are critical for air quality, supporting both prediction and interpretation.

System Implementation & Challenges

- **Connection Resilience:** I identified and documented HTTPSConnectionPool errors occurring during API calls to Hopsworks, noting that these are network/DNS related and do not affect the internal model logic.
- Internel server 500 error and remote disconnected

Conclusion

The Islamabad AQI Prediction System successfully automates the transition from raw environmental data to actionable public health forecasts. By correctly **transforming categorical time features** and leveraging an **imbalanced but informative historical dataset**, the model is highly specialized in detecting hazardous air quality trends.