AIR QUALITY PREDICTION SYSTEM

Real-Time PM2.5 Forecasting for Bahawalpur, Pakistan

A Comprehensive Machine Learning Approach to 3-Day Air Quality Index Prediction

Prepared by: Umyma Mohsin

Project Report

February 12, 2026

Summary

This project presents a comprehensive machine learning-based solution for predicting Air Quality Index (AQI) levels in Bahawalpur, Pakistan, up to 72 hours in advance. The system leverages real-time meteorological data, historical air quality measurements, and advanced ensemble modeling techniques to provide accurate PM2.5 concentration forecasts.

Key achievements include the development of a fully serverless MLOps pipeline, deployment of an interactive web dashboard for real-time predictions, and integration with the Hopsworks Feature Store for scalable data management. The solution employs multiple gradient boosting algorithms (XGBoost, LightGBM, CatBoost) to achieve optimal prediction accuracy across different forecast horizons (24h, 48h, 72h).

1. Introduction

1.1 Problem Statement

Air pollution is a critical environmental and public health challenge in Pakistan, particularly in urban areas. Bahawalpur, with its growing industrial activity and vehicular emissions, experiences fluctuating air quality levels that pose health risks to its residents. Real-time air quality monitoring exists, but the absence of reliable predictive systems limits the ability of citizens and authorities to take preventive measures.

The primary objective of this project is to develop an accurate, automated system that forecasts PM2.5 concentrations (the primary indicator of air quality) for the next 24, 48, and 72 hours, enabling proactive decision-making for health protection and policy interventions.

1.2 Project Objectives

- Develop a multi-horizon forecasting model for PM2.5 concentrations (24h, 48h, 72h)
- Create a fully automated, serverless MLOps pipeline for continuous model training and deployment
- Build an interactive web dashboard for real-time predictions and data visualization
- Implement feature engineering techniques to capture temporal and meteorological patterns
- Achieve production-grade model performance with automated monitoring and retraining

1.3 Scope and Deliverables

The project delivers a complete end-to-end solution encompassing data collection, feature engineering, model training, deployment, and user interface. The system operates entirely on cloud infrastructure with automated pipelines for data ingestion, model retraining, and inference serving.

2. Methodology and Approach

2.1 Alternative Approaches Considered

Several methodologies were evaluated for air quality prediction before selecting the final approach:

2.1.1 Time Series Forecasting Methods

- **ARIMA/SARIMA Models:** Traditional statistical approach for univariate time series. Not selected due to limited ability to incorporate meteorological features and non-linear patterns.
- **Prophet:** Facebook's additive regression model. Rejected due to lack of support for multi-horizon output and limited feature engineering capabilities.
- **VAR/VECM:** Vector autoregression for multivariate time series. Not chosen due to assumption of linear relationships and complexity in handling exogenous features.

2.1.2 Deep Learning Approaches

- **LSTM Networks:** Recurrent neural networks for sequence modeling. Considered but not implemented due to longer training times, higher computational requirements, and need for larger datasets to achieve comparable performance.
- **Transformer Models:** Attention-based architectures. Reserved for future work due to complexity and data volume requirements.
- **CNN-LSTM Hybrid:** Convolutional layers combined with LSTM. Not selected due to implementation complexity and marginal performance gains over gradient boosting for tabular data.

2.1.3 Ensemble Machine Learning (Selected Approach)

Gradient Boosting Ensemble: Selected as the primary approach combining multiple gradient boosting algorithms (XGBoost, LightGBM, CatBoost) with traditional regression baselines (Ridge, Lasso, Random Forest).

Rationale for selection:

- Superior performance on tabular data with mixed feature types
- Ability to capture complex non-linear relationships
- Built-in feature importance for interpretability
- Faster training and inference compared to deep learning
- Robust to missing data and outliers
- Lower computational resource requirements suitable for serverless deployment

3. Data Collection and Feature Engineering

3.1 Data Sources

- **Air Quality Data:** Hourly PM2.5 measurements from OpenMeteo Air Quality API for Bahawalpur (29.40°N, 71.68°E)
- **Meteorological Data:** Temperature, humidity, wind speed, atmospheric pressure from OpenMeteo Weather API
- **Temporal Coverage:** Historical data from August 2025 to present, updated hourly

3.2 Feature Engineering

The following engineered features were created to capture temporal patterns, trends, and meteorological influences:

Feature Category	Feature Name	Description
Lag Features	pm25_lag1, pm25_lag6, pm25_lag24	PM2.5 values from 1h, 6h, 24h ago
Rolling Statistics	pm25_ma6, pm25_ma24	6-hour and 24-hour moving averages
Change Features	pm25_change_1hr	Hour-over-hour rate of change
Meteorological	temperature_2m, relative_humidity_2m, wind_speed_10m, pressure_msl	Real-time weather conditions
Temporal	hour, day_of_week, day, month	Cyclical time patterns
Target	forecast_horizon	Prediction window (24/48/72 hours)

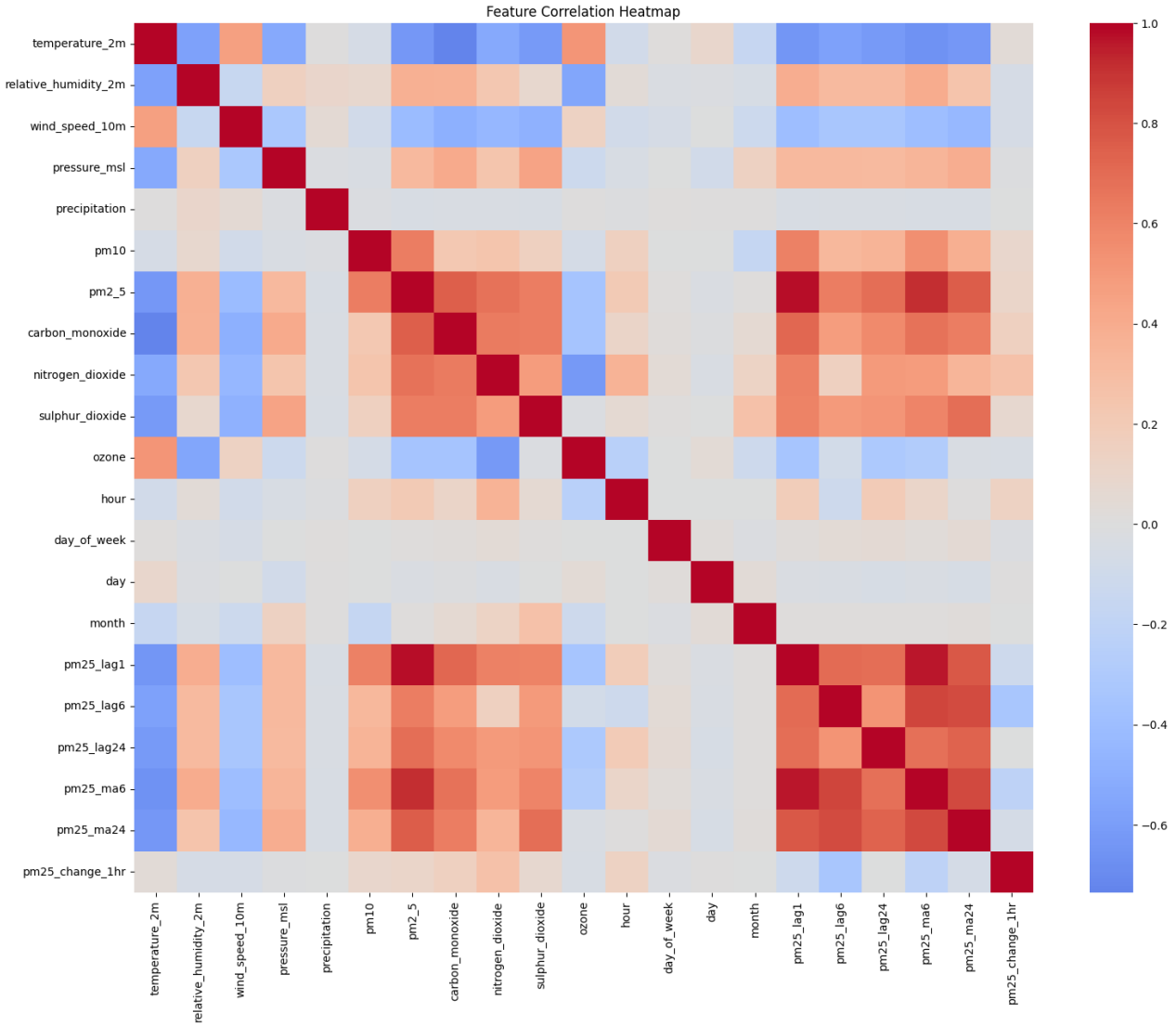
Total Features: 15 features per training instance (14 base features + 1 horizon indicator)

4. Exploratory Data Analysis

Comprehensive exploratory analysis was conducted to understand data patterns, identify relationships, and guide feature engineering decisions.

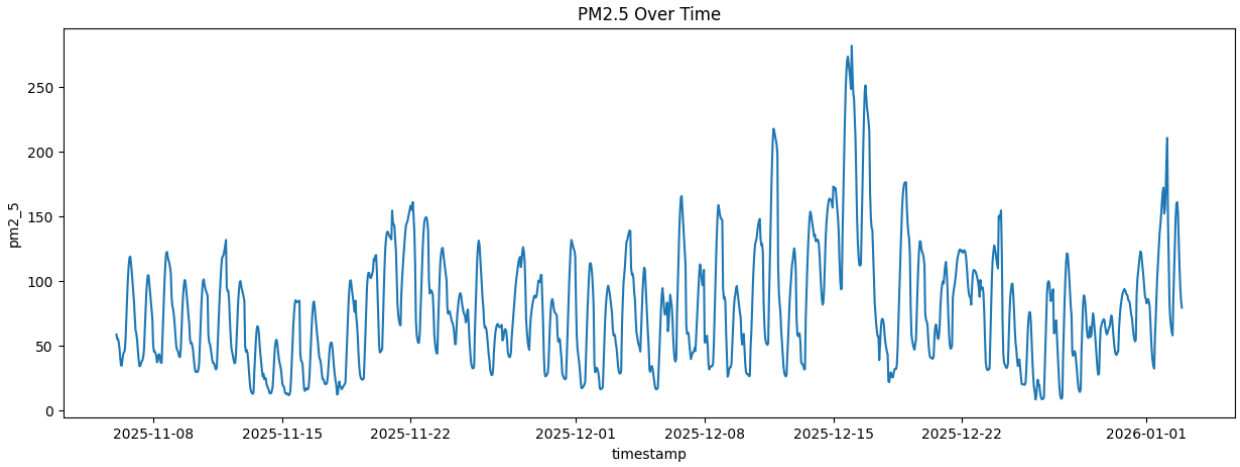
4.1 Feature Correlations

Correlation heatmap of all features



4.2 Temporal Patterns

Time series plot of PM2.5 levels over time



5. Model Training and Evaluation

5.1 Training Strategy

Multi-Horizon Unified Model: A single model was trained to predict all three forecast horizons (24h, 48h, 72h) simultaneously by incorporating the forecast horizon as an additional feature. This approach ensures consistency across predictions and reduces deployment complexity.

5.2 Evaluation Metrics

- **RMSE (Root Mean Squared Error):** Primary metric penalizing large errors
- **MAE (Mean Absolute Error):** Average magnitude of prediction errors
- **R^2 Score:** Proportion of variance explained by the model

5.3 Model Performance Comparison

Key Findings:

- Gradient boosting algorithms (XGBoost, LightGBM, CatBoost) consistently outperformed baseline models
- CatBoost achieved the best overall performance with lowest RMSE and highest R^2

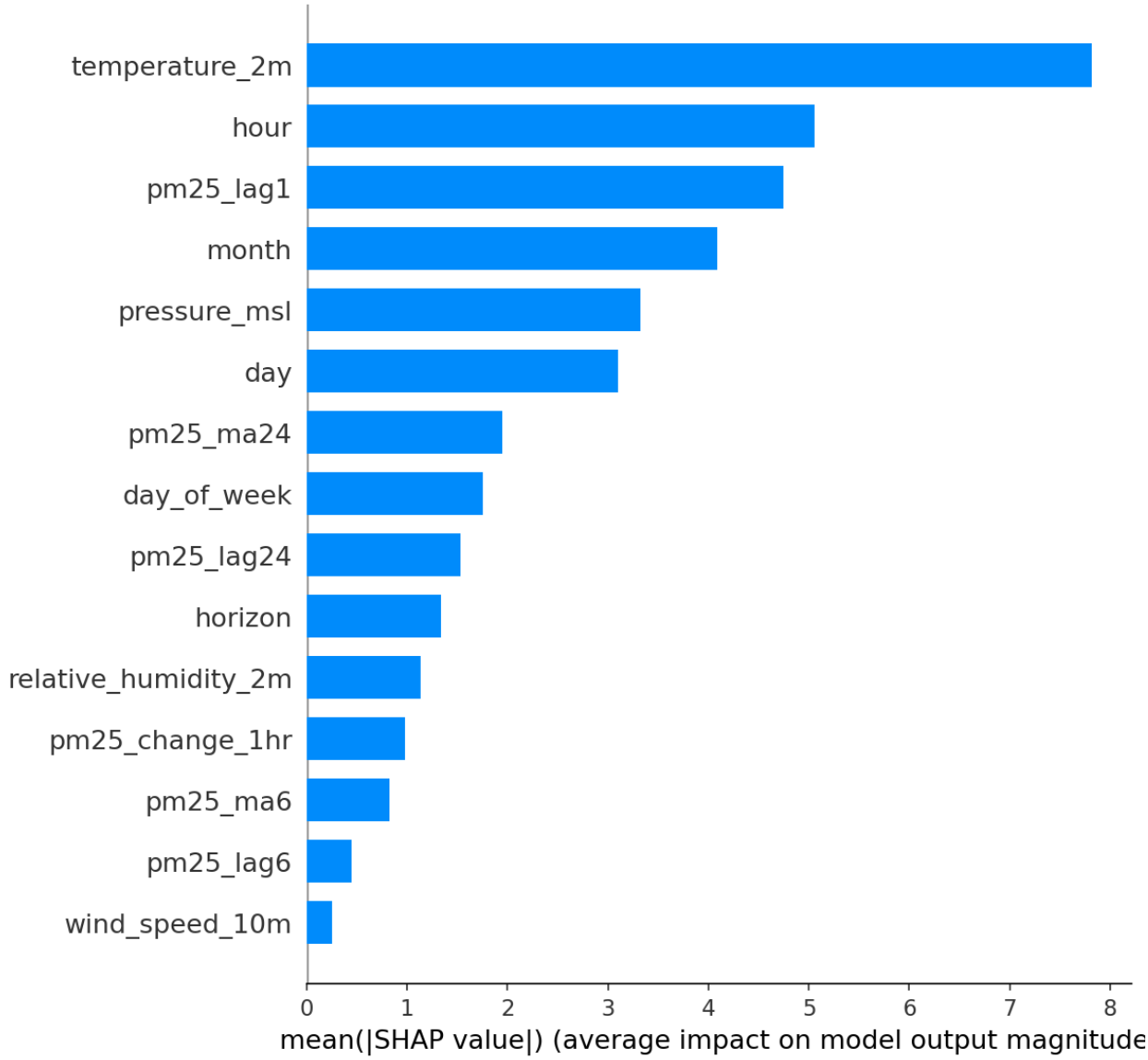
- Linear models (Ridge, Lasso) showed limited ability to capture non-linear patterns

6. Model Interpretability

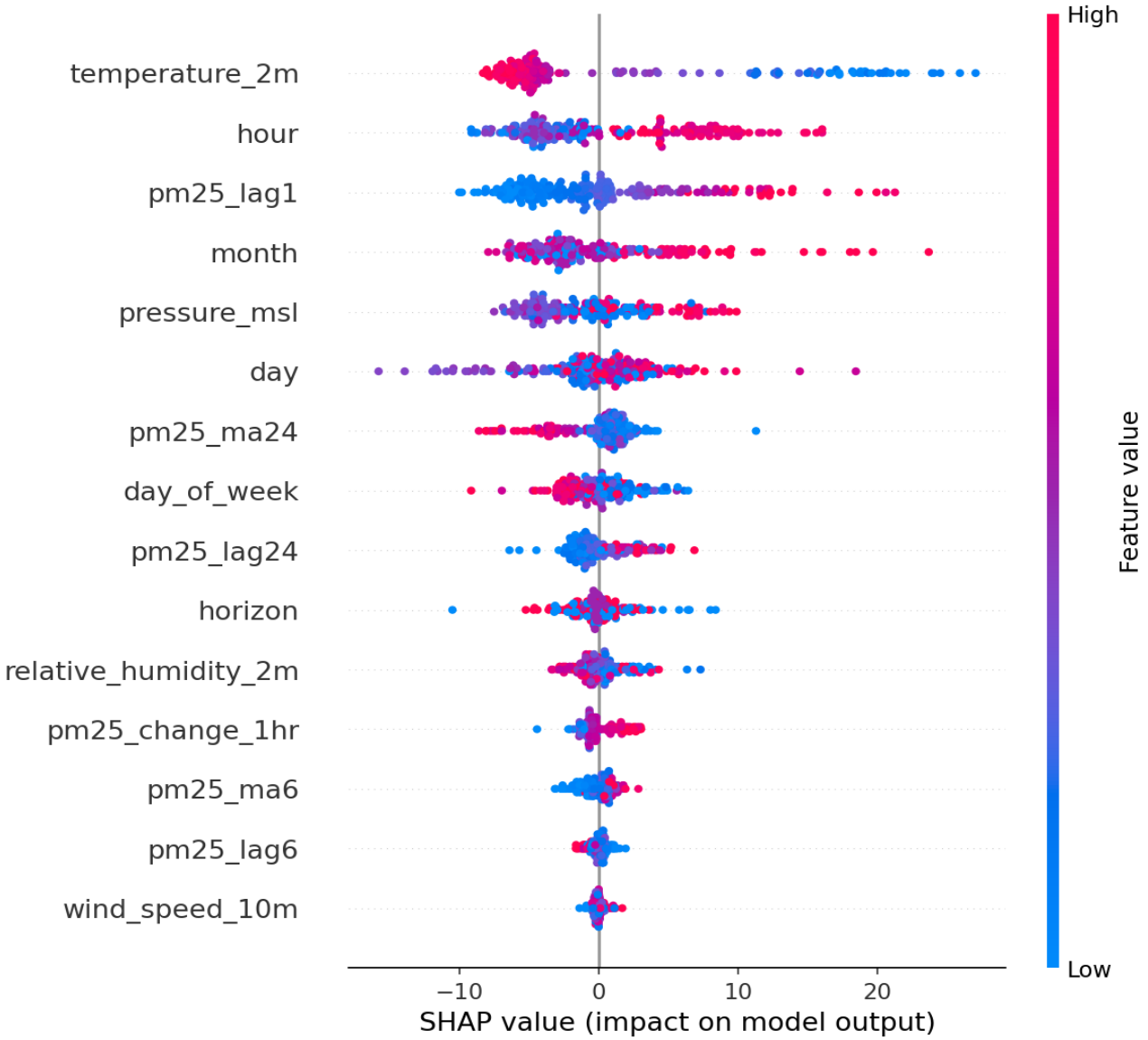
6.1 SHAP Analysis

SHapley Additive exPlanations (SHAP) values were computed to understand feature importance and individual prediction contributions.

6.1.1 Global Feature Importance



6.1.2 Feature Impact Distribution



6.2 Key Insights

The SHAP analysis indicates that temperature, time-of-day, and recent PM2.5 history are the strongest drivers of AQI predictions. Higher temperatures and specific hours consistently push AQI upward, reflecting meteorological and human activity patterns, while pm25_lag1 and the 24-hour moving average confirm strong pollution persistence. Seasonal effects captured by month and atmospheric pressure also contribute, whereas wind speed, short-term lags, and rapid PM2.5 changes have minimal influence.

7. MLOps Pipeline and Automation

7.1 Architecture Overview

The project implements a fully serverless MLOps pipeline with the following components:

1. **Feature Pipeline:** Automated data collection, preprocessing, and feature engineering (GitHub Actions scheduled hourly)
2. **Training Pipeline:** Model training and evaluation (scheduled once every day)
3. **Inference Pipeline:** Real-time predictions served via Streamlit dashboard
4. **Feature Store:** Hopsworks for centralized feature management and versioning
5. **Model Registry:** Hopsworks Model Registry for version control and deployment

7.2 Technology Stack

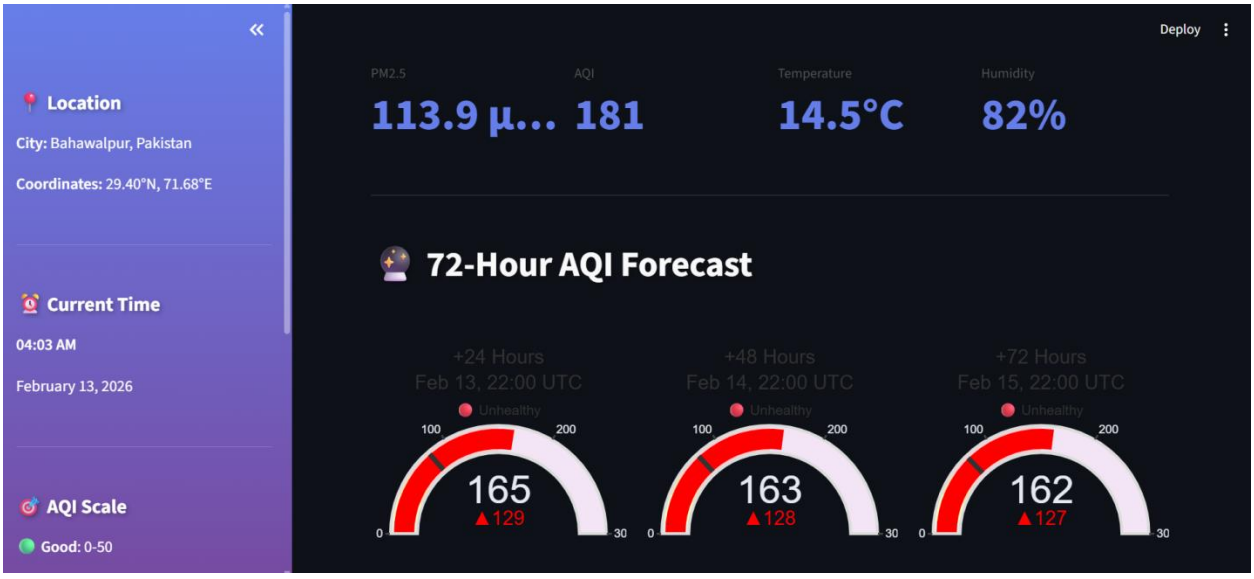
- **Cloud Platform:** GitHub Actions (CI/CD), Streamlit Cloud (deployment)
- **Feature Store:** Hopsworks Feature Store
- **Model Registry:** Hopsworks Model Registry
- **ML Frameworks:** scikit-learn, XGBoost, LightGBM, CatBoost, SHAP
- **Data Processing:** pandas, NumPy
- **Visualization:** Plotly, Matplotlib, Streamlit
- **APIs:** OpenMeteo Weather API, OpenMeteo Air Quality API

8. Dashboard Deployment

8.1 User Interface Features

- Real-time PM2.5 and AQI display with color-coded health categories
- Interactive gauge charts for 24h, 48h, and 72h forecasts
- Timeline visualization showing forecast trends
- Meteorological conditions display (temperature, humidity, wind, pressure)
- Model performance metrics and version information
- Responsive design for desktop and mobile devices

8.2 Dashboard Screenshots



9. Challenges

During the development of this project, I encountered several interconnected technical challenges. I faced version compatibility issues when Streamlit Cloud defaulted to Python 3.13, which was incompatible with the Hopsworks 4.2.x client, and resolved this by enforcing Python 3.12 and aligning dependent libraries. In the early stages, training models on only one month of historical data led to weak and unstable metrics, which significantly improved after expanding the dataset to approximately six months and using gradient boosting models. I initially trained separate models for each prediction horizon, but this resulted in suboptimal performance and increased inference latency, prompting a shift to a single unified model that improved generalization and reduced application load time. I also encountered negative R^2 scores due to insufficient feature engineering which was addressed through time-based splits, lag and rolling features, and feature refinement. I also ran into issues while uploading models to the Hopsworks Model Registry but i eventually resolved the issues and was able to export them to the registry. Finally, while the pipelines generally ran reliably, they occasionally failed due to transient Hopsworks connectivity issues, highlighting the need for stronger retry and monitoring mechanisms in future iterations.

10. Future Improvements

- **Additional Pollutants:** Incorporate PM10, CO, NO_2, SO_2, and O_3 measurements as features to capture multi-pollutant interactions
- **Feature Groups:** Organize features into logical groups (meteorological, temporal, pollutants) and create a unified Feature View in Hopsworks for better organization and reusability
- **Deep Learning Exploration:** Experiment with LSTM, GRU, and Transformer architectures for potential performance gains
- **Hyperparameter Optimization:** Automated hyperparameter tuning using Optuna or Ray Tune
- **R^2 Score Improvement:** Target $R^2 > 0.90$ through advanced feature engineering and model tuning

11. Conclusion

This project successfully demonstrates the application of modern machine learning and MLOps practices to solve a real-world environmental challenge. The developed system provides 3-day PM2.5 and AQI forecasts for Bahawalpur, Pakistan, using an ensemble of gradient boosting algorithms and comprehensive feature engineering. The system is currently operational and providing daily forecasts, with plans for continuous improvement.

References

1. OpenMeteo. (2024). Air Quality API Documentation. Retrieved from https://open-meteo.com/en/docs/air-quality-api
2. Hopsworks. (2024). Feature Store Documentation. Retrieved from https://docs.hopsworks.ai/
3. World Health Organization. (2021). WHO Global Air Quality Guidelines. Geneva: World Health Organization.
4. U.S. Environmental Protection Agency. (2024). Air Quality Index (AQI) Basics. Retrieved from https://www.airnow.gov/aqi/