Data Streams Quest

Team RHYMe
Fundamentals of Operationalizing AI

Hemant Sathish Meghna Nair Reuben Mathew Yash Thakkar

Contents

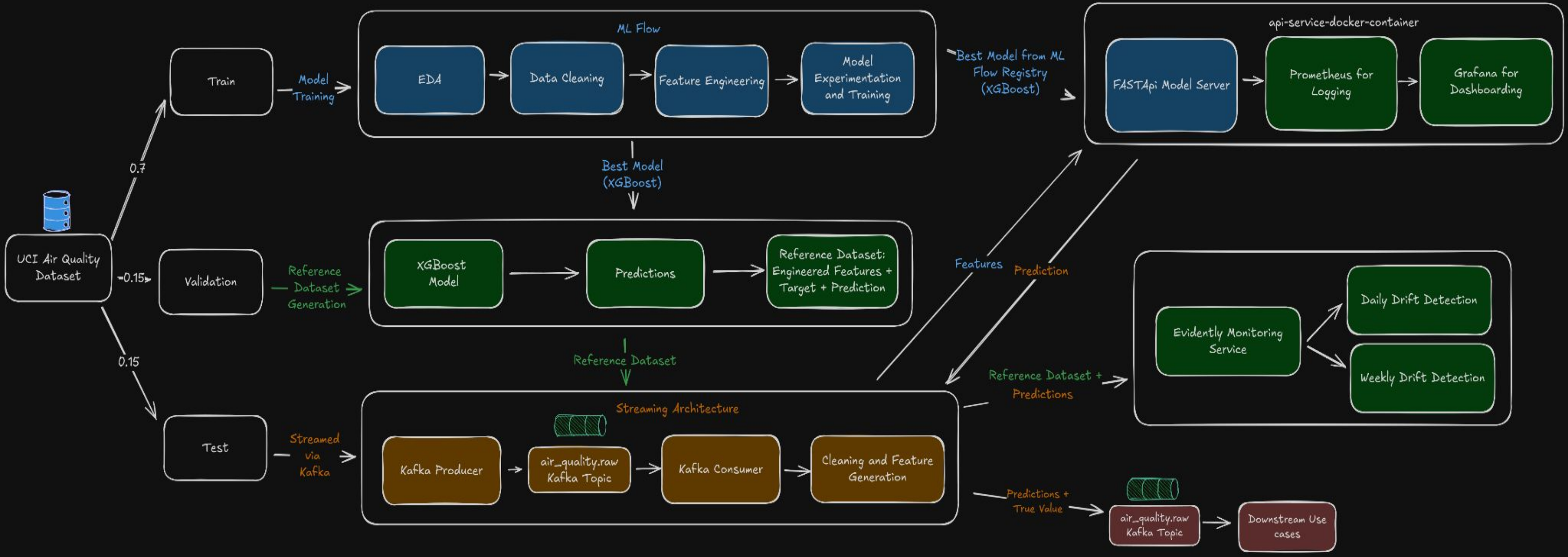
Introduction

Context:
Air pollution is one of the top global health risks which requires immediate anomaly detection

Dataset:
UCI Air Quality with hourly sensor readings of pollutants from a monitoring station

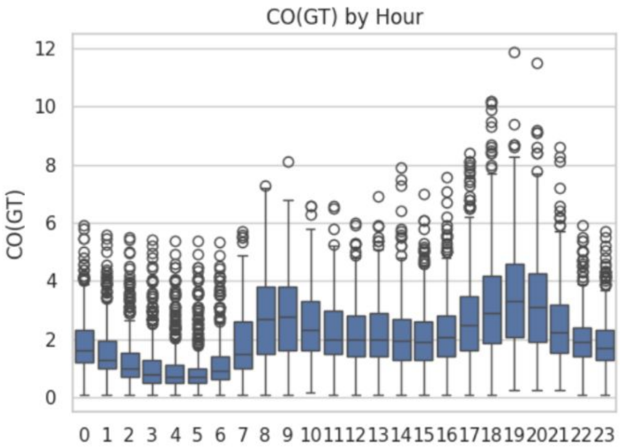
Goal:
Develop an end to end data streaming pipeline for environmental time series forecasting that predicts the level of Carbon Monoxide in real time

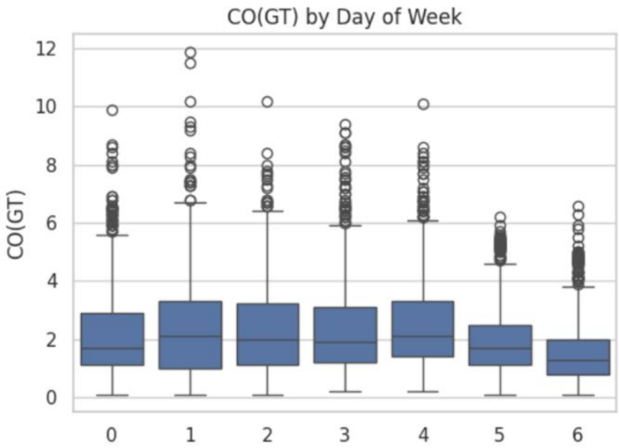
Project Infrastructure

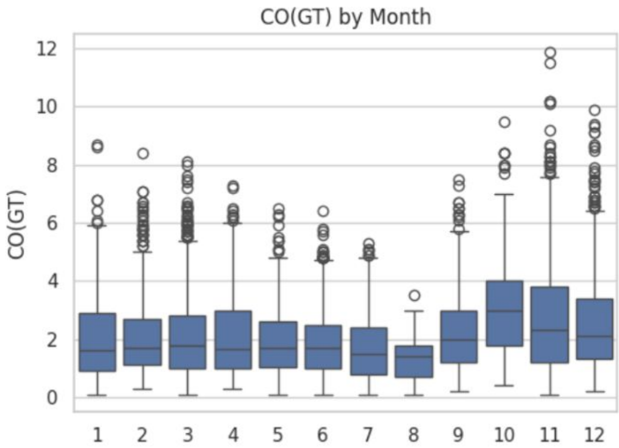


Data Overview and Feature Engineering

UCI Machine Learning Repository Air Quality Data Set	**9357** Hourly Observations March 2004 to April 2005	**15 Parameters** Gaseous Pollutants, MO Sensors, Meteorological	**CO** Target Variable







Feature Engineering

Category	**Examples**	**Purpose**
Lag Features	CO(GT)_lag_{1,2,3,6,12,24,48,72}	Model recent history and autoregressive effects
Rolling Statistics	Mean / Std / Min / Max over {3, 6, 12, 24, 48, 168 h}	Capture short and long-term trends
Rate-of-changes	1 h, 3 h, 24 h differences of pollutant sensors	Detect sudden spikes or decays
Cross-Pollutant interactions	Pairwise products & ratios	Encode chemical/physical coupling
Environmental effects	$T \times AH$, T^2, AH^2, $T \times RH$	Capture nonlinear weather influences
Temporal context	Hour, Day, Month, Week, sin/cos encodings	Preserve cyclical daily / seasonal patterns
Binary Flags	Weekend, Rush hour, Night, Winter, Summer	Represent categorical time patterns

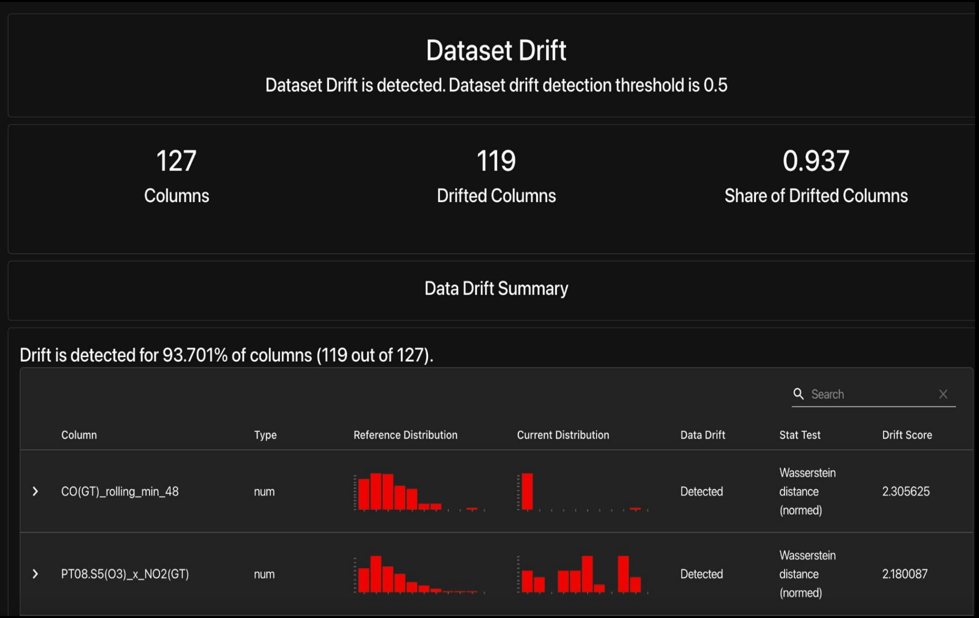
Model Training

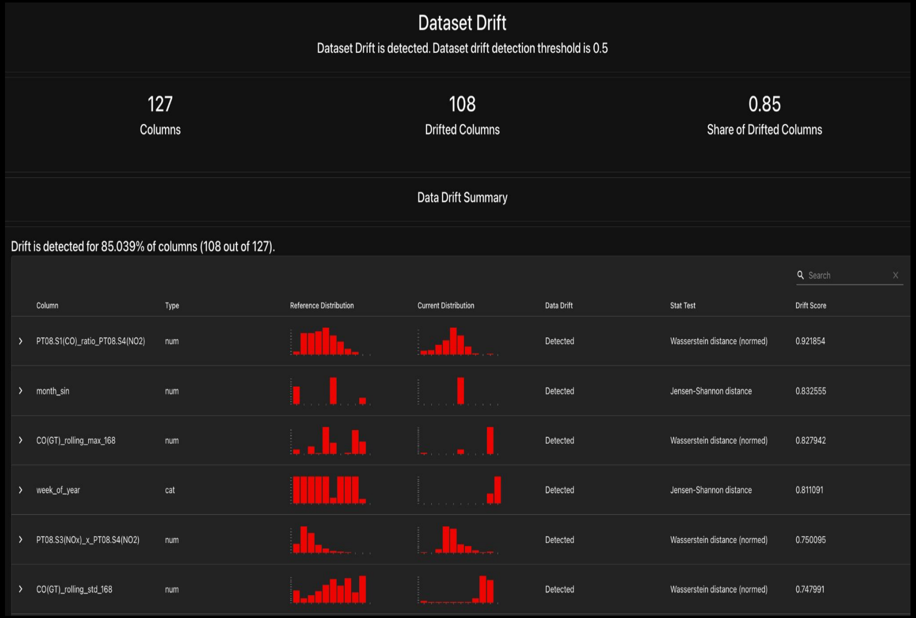
Model	RMSE	MAE	R^2	SMAPE
Random Forests	0.3618	0.2166	0.9249	11.978%
Extra Trees	0.4099	0.2743	0.9036	15.636%
Gradient Boosting	0.3142	0.2165	0.9434	13.561%
XGBoost (XGB)	0.2686	0.1715	0.9586	10.498%
LightGBM (LGB)	0.3115	0.2006	0.9443	11.923%
CatBoost (CAT)	0.3111	0.2074	0.9445	12.848%
StackEnsemble	0.2889	0.1956	0.9521	12.454%
WeightedEnsemble	0.2907	0.1859	0.9515	11.185%

XGBoost Training
Bayesian optimization tuned XGBoost hyperparameters.
Early stopping to find the optimal number of trees.
Retrained model using using the optimal parameters.

Evidently

Continuously tracks **data quality**, **feature drift**, and **model stability** in the real-time air-quality pipeline.Compare **current feature distributions** with a **reference dataset.**





Evidently - Model Performance

Regression Model Performance. Target: 'CO(GT)'

Current: Model Quality (+/- std)

0.03 (0.24)	0.17 (0.17)	9.51 (0.09)
ME	MAE	MAPE

Reference: Model Quality (+/- std)

-0.01 (0.23)	0.15 (0.17)	9.94 (0.2)
ME	MAE	MAPE

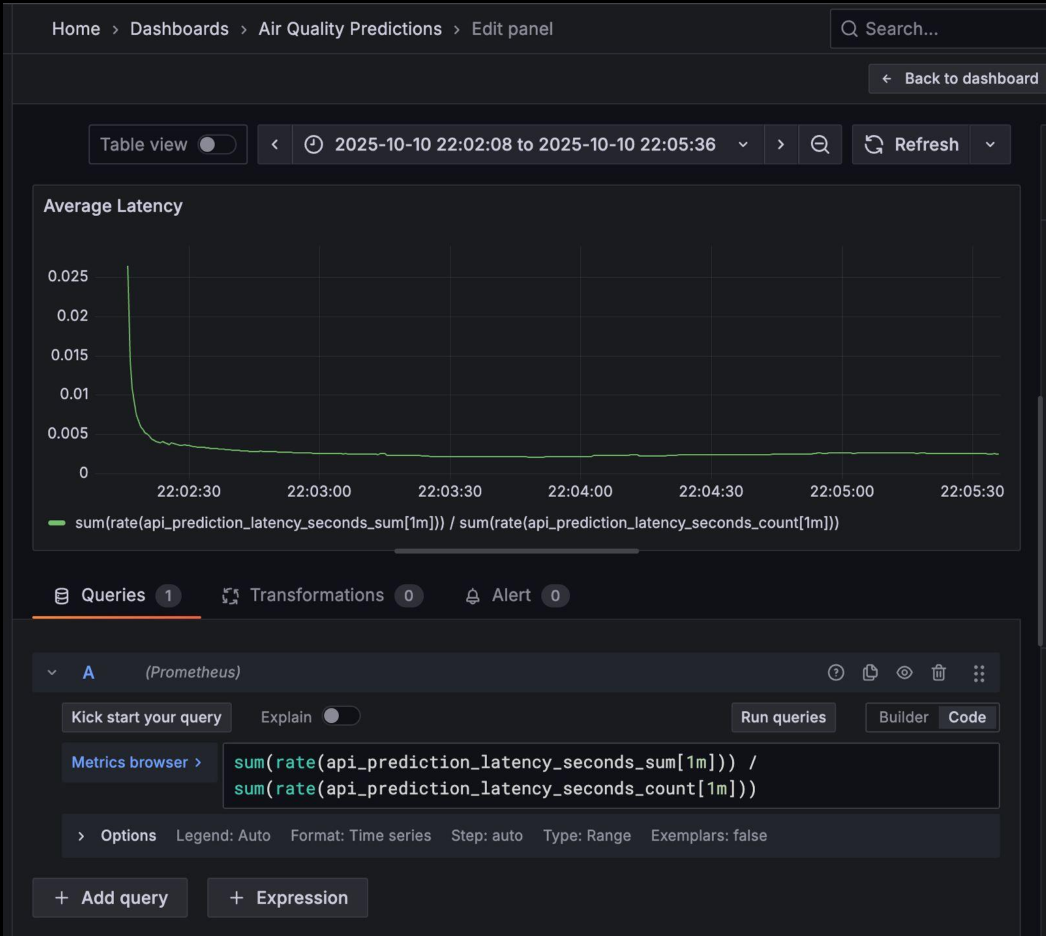
Predicted vs Actual in Time

Bonus: Grafana & Prometheus

Grafana pulls data from Prometheus which comes from tracking API service predictions.

Avg. prediction latency says how long each API request takes to return a result.

Early spike due to cold start, stabilizes as requests flow.



Learnings

- Feature engineering strongly improved predictive power
- Bayesian search, took a long time but helped finding optimal configurations
- Databricks with MLFlow streamlined training and comparison

Future Enhancements

- Investigate
  - Dimensionality Reduction
  - Multi-source Forecasting
- Potential LLM integration for automated business decision making for stakeholders and leadership members