22 FEB 2024

Pollution Prediction

Using Online Learning

MIA 2023-24

Brian García Machado
Fernando Núnez Sánchez
Marcin Jedrzejowski
Santiago Suárez Carrera

Pollution Predictor

1. Introduction

Project Overview

- **Predicting pollution levels** is crucial for public health and environmental policy-making.
- Our objective is to forecast PM2.5 pollution levels **24 hours in advance** using **weather forecasts** and **current pollution data**.
- We'll utilize **batch learning** and **online learning** techniques to model those pollution levels.

Problem description

- **Problem:** Time series forecasting of pollution levels 24 hours ahead.
- **Type of problem: Regression**.
- **Data balance**: The distribution of pollution levels is skewed towards lower values, with fewer high-peak events.
- **Potential for Concept Drift:** Seasonal changes, urban development, and policy shifts may impact pollution levels over time.
- **Evaluation Metrics:** Considering **Mean Average Error (MAE)** for comprehensive model performance evaluation, reflecting the precision of our predictions.
- **Assumptions:** weather data is considered a 24h forecast, treating historical weather data as a proxy for future conditions.

2. Dataset selection

Dataset

This Kaggle dataset is key for predicting PM2.5 levels, **offering hourly atmospheric and pollution measurements**. Its comprehensive coverage and granularity facilitate an in-depth analysis of air quality trends, essential for our predictive modelling efforts.

kaggle

PM2.5	**Pollution Target**
5 years	**Time Coverage**
US embassy in Beijing	**Location**

3. Data preparation

Data Selection

Features:

- Timestamp of the observation
- Current pollution
- Dew point (+24h)
- Temperature (+24h)
- Pressure (+24h)
- Wind direction (+24h)
- Wind speed (+24h)
- Snowfall (+24h)
- Rainfall (+24h)

Target:
24h-ahead pollution

	date	pred_pollution	dew	temp	press	wnd_dir	wnd_spd	snow	rain	current_pollution
0	2010-01-03 00:00:00	90.0	-7	-6.0	1027.0	SE	58.56	4	0	129.0
1	2010-01-03 01:00:00	63.0	-8	-6.0	1026.0	SE	61.69	5	0	148.0
2	2010-01-03 02:00:00	65.0	-8	-7.0	1026.0	SE	65.71	6	0	159.0
3	2010-01-03 03:00:00	55.0	-8	-7.0	1025.0	SE	68.84	7	0	181.0
4	2010-01-03 04:00:00	65.0	-8	-7.0	1024.0	SE	72.86	8	0	138.0

Data Distribution

Pollution

Exhibits an exponential distribution.

Decreasing frequency towards higher values.

Dew and Temperature

Follow a binomial distribution.
Expected for weather data across seasons.
Sporadic spikes suggest localized increases.

Air Pressure

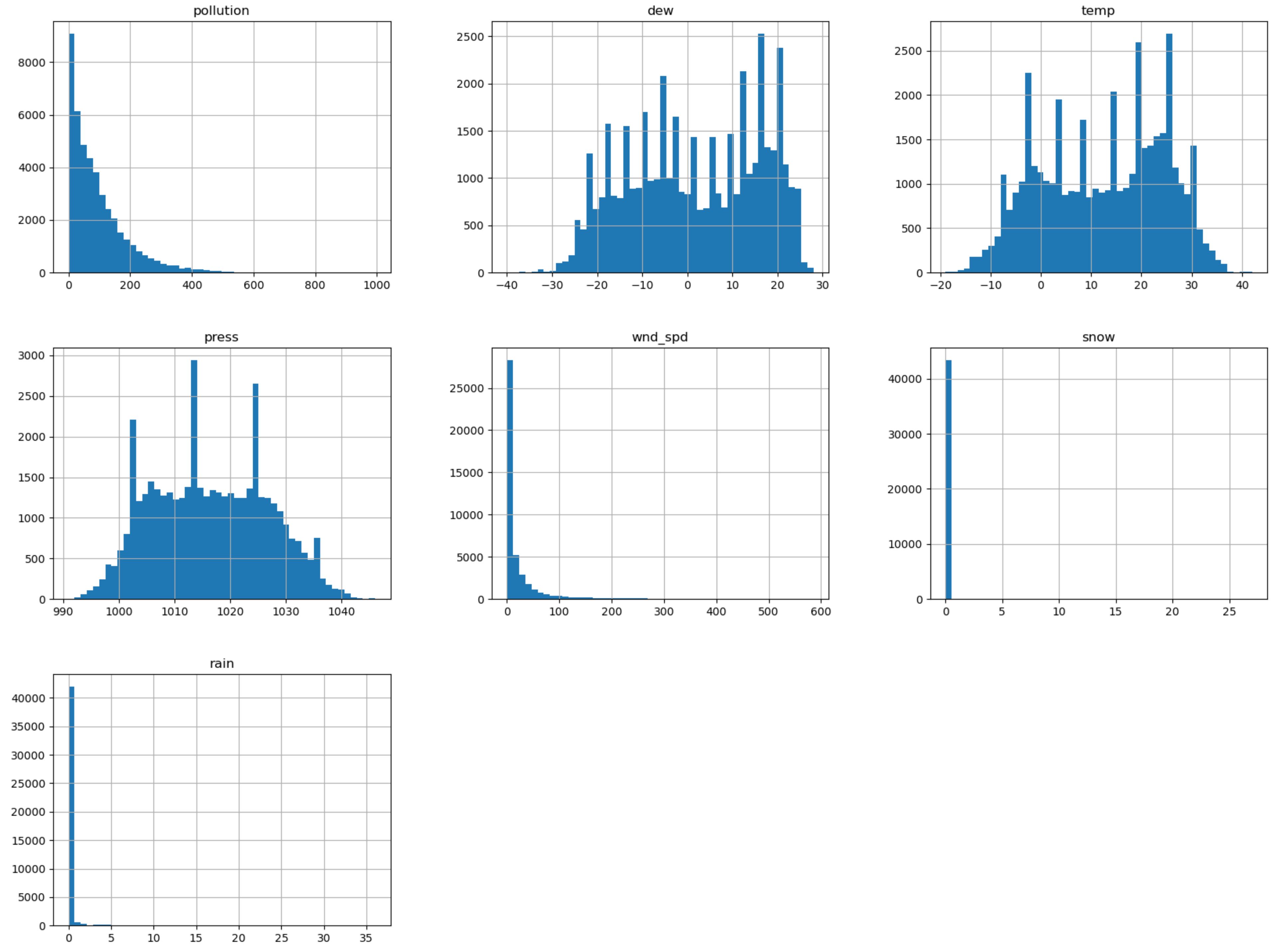
Adheres to a normal distribution.
Spans values around 990 to 1040 hPa.
Average pressure close to 1013 hPa.

Wind Speed

Follows an exponential distribution.
More common occurrence of weaker winds.

Snowfall and Rainfall

Predominantly cluster around 0.
Reflects the warm temperate zone climate.



Data preparation

1. Shifting the *pollution* column by 24 hours.
2. Forecast weather features.
3. Encode the *wind direction* categorical variable.
4. Standarization of numerical features.

🕐 **Shift** the hours to create *current pollution*.

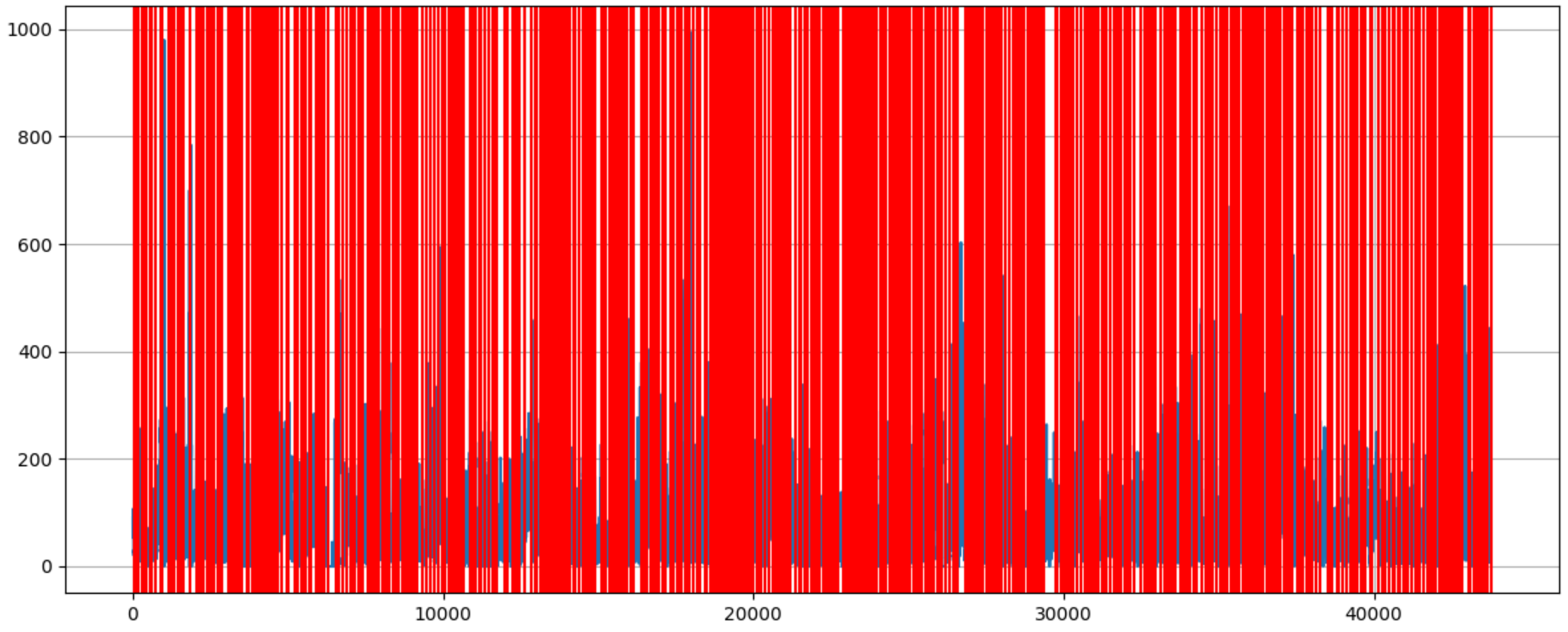
📐 **Standarize** and **encode** categorical values.

🗜️ Accommodate **River's requirements**

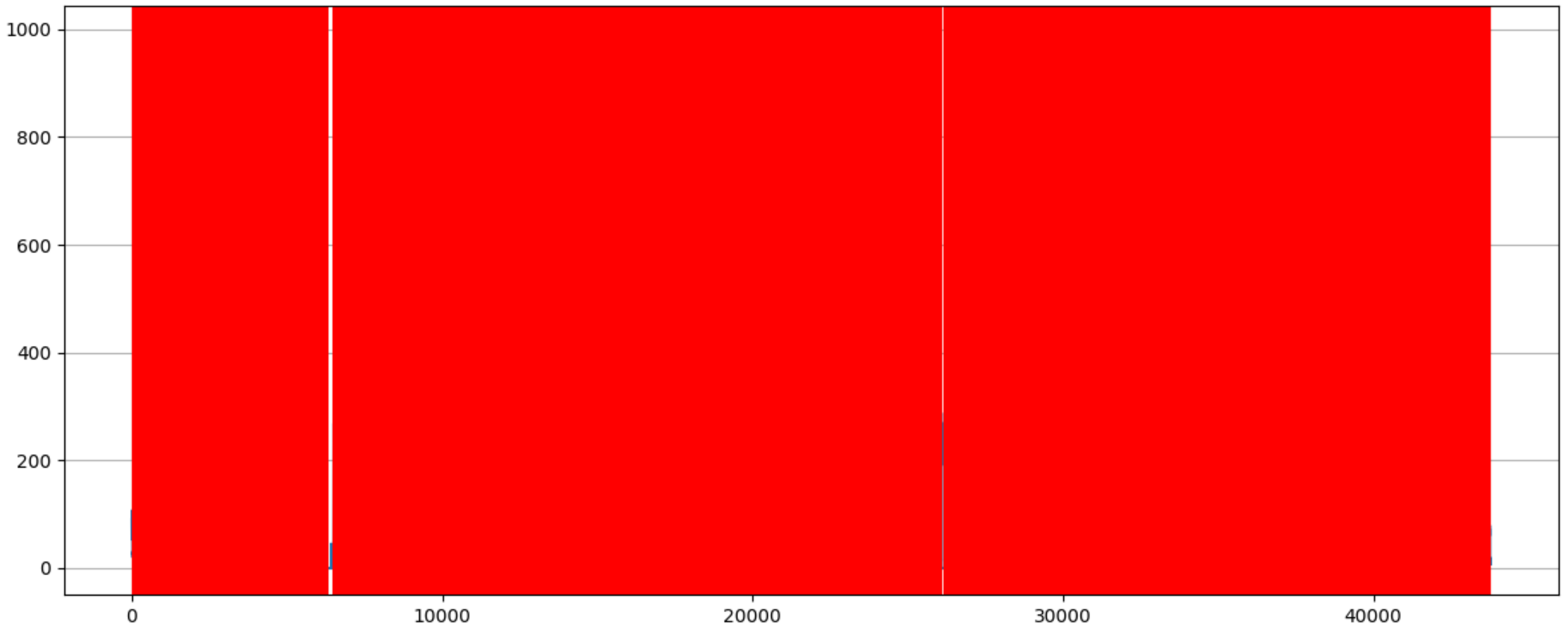
4. Concept drift

Concept drifts

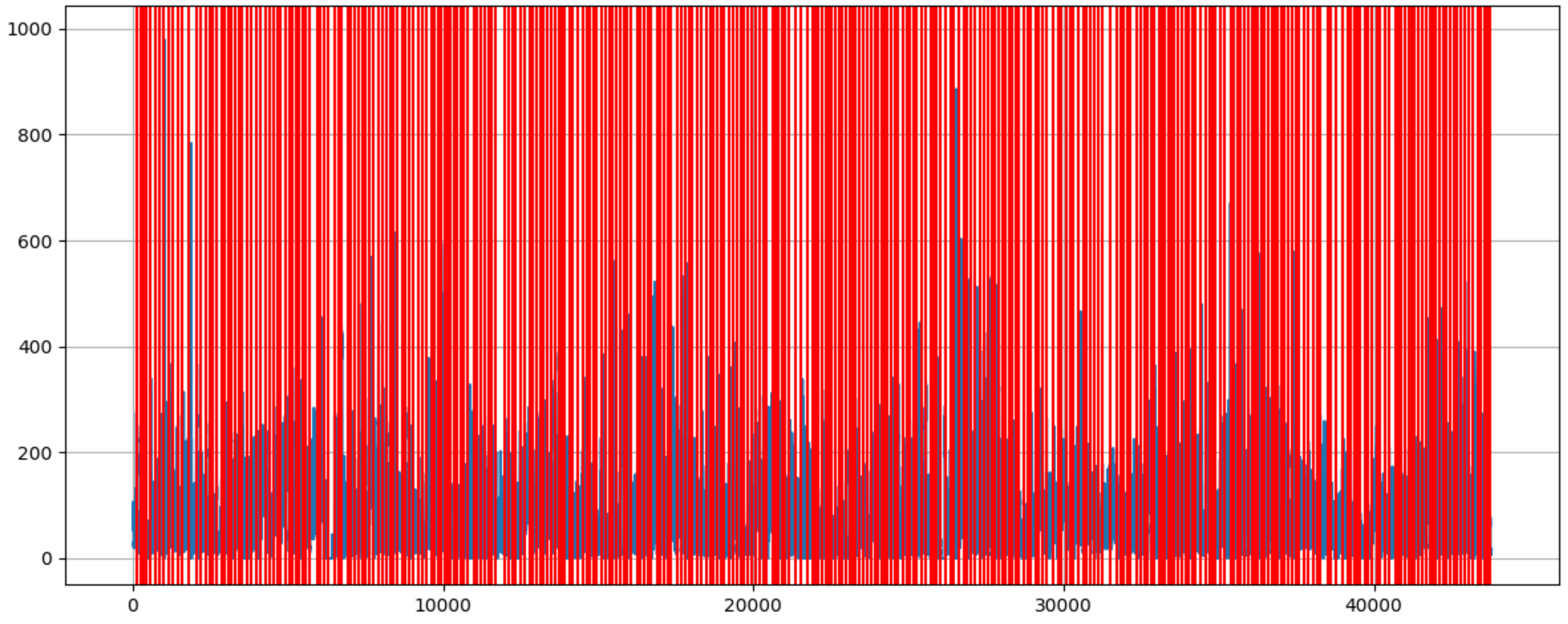
ADWIN (510 drifts detected)



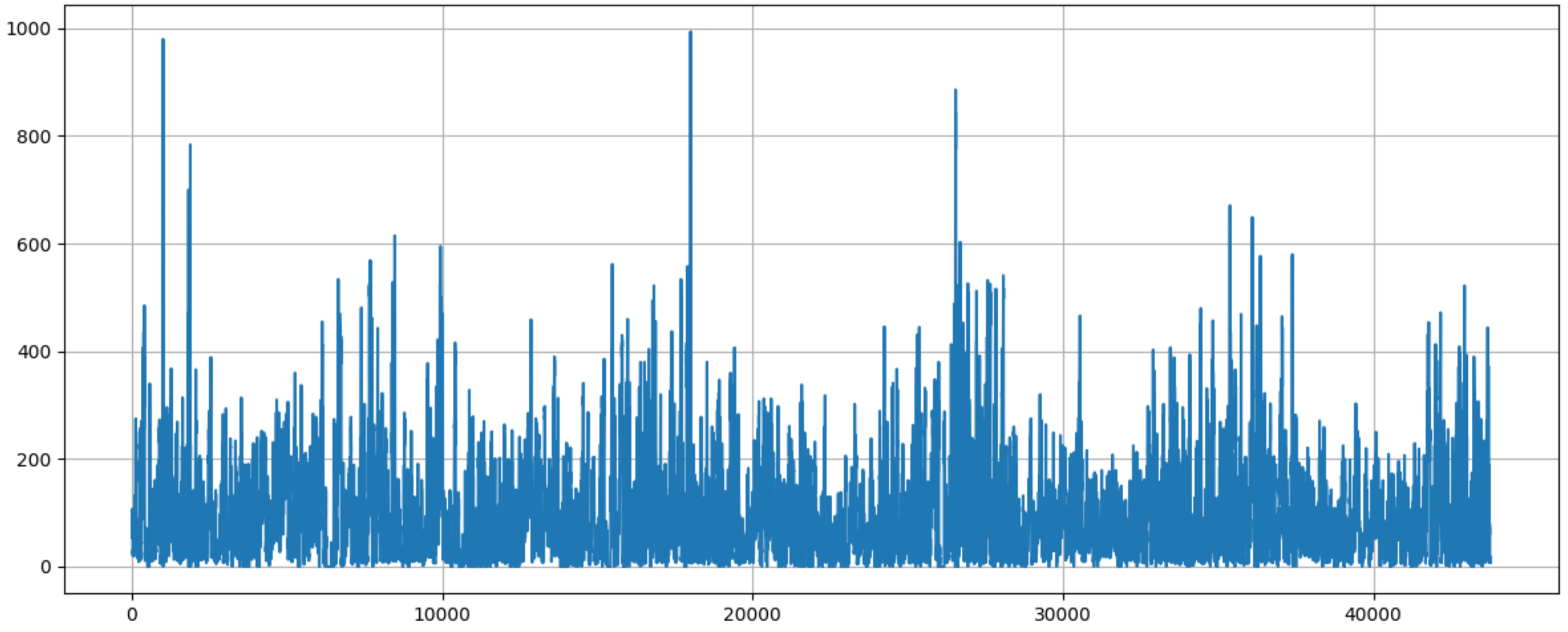
PageHinkley (1432 drifts detected)



KSWIN (345 drifts detected)



ADWIN (0 drifts detected)



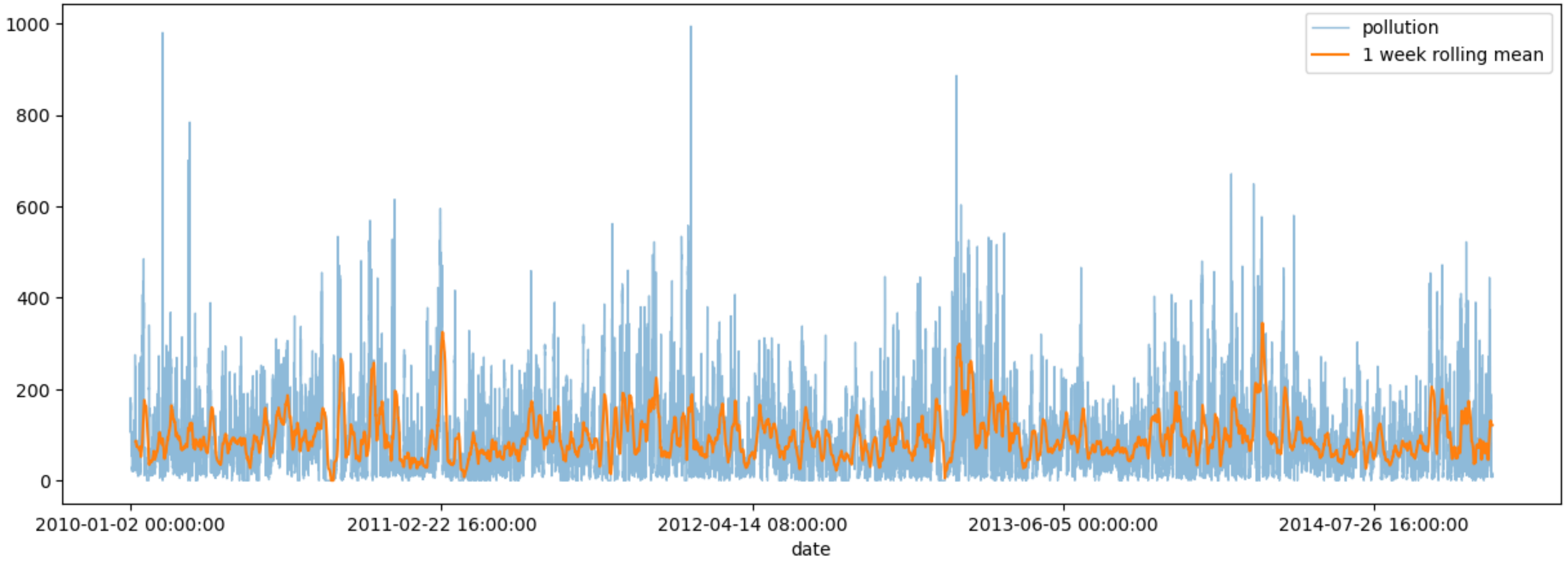
Concept drifts

Few drifts or none:

- Stable data distribution
- Infrequent changes
- No need to update the model as often
- Not our case

High number of drifts:

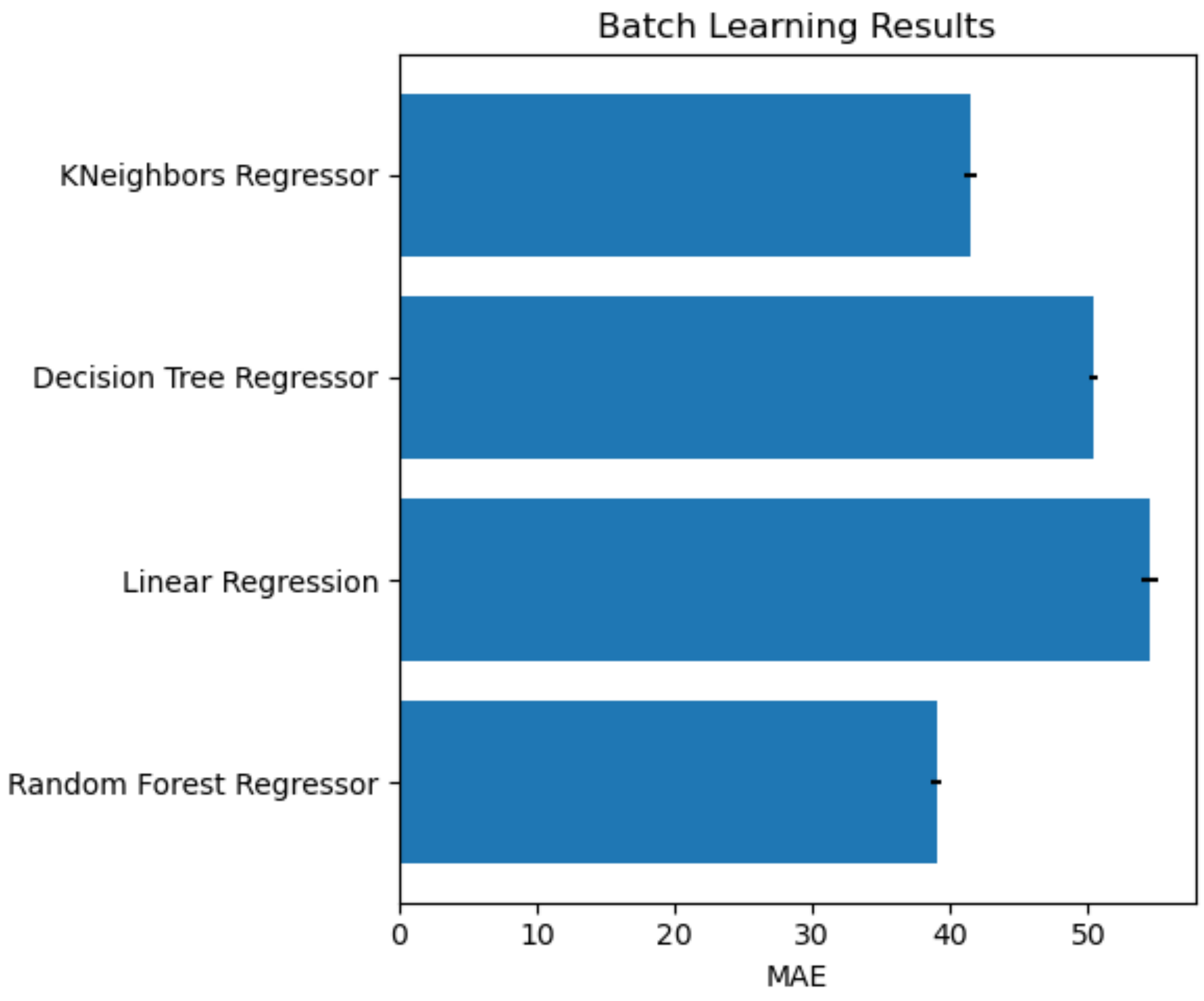
- **Unstable** data distribution
- Quickly **adaptation** to changes is needed
- **Our case** (time series, seasonal effect, dynamic phenomenas, …)



5. Batch learning

Batch learning

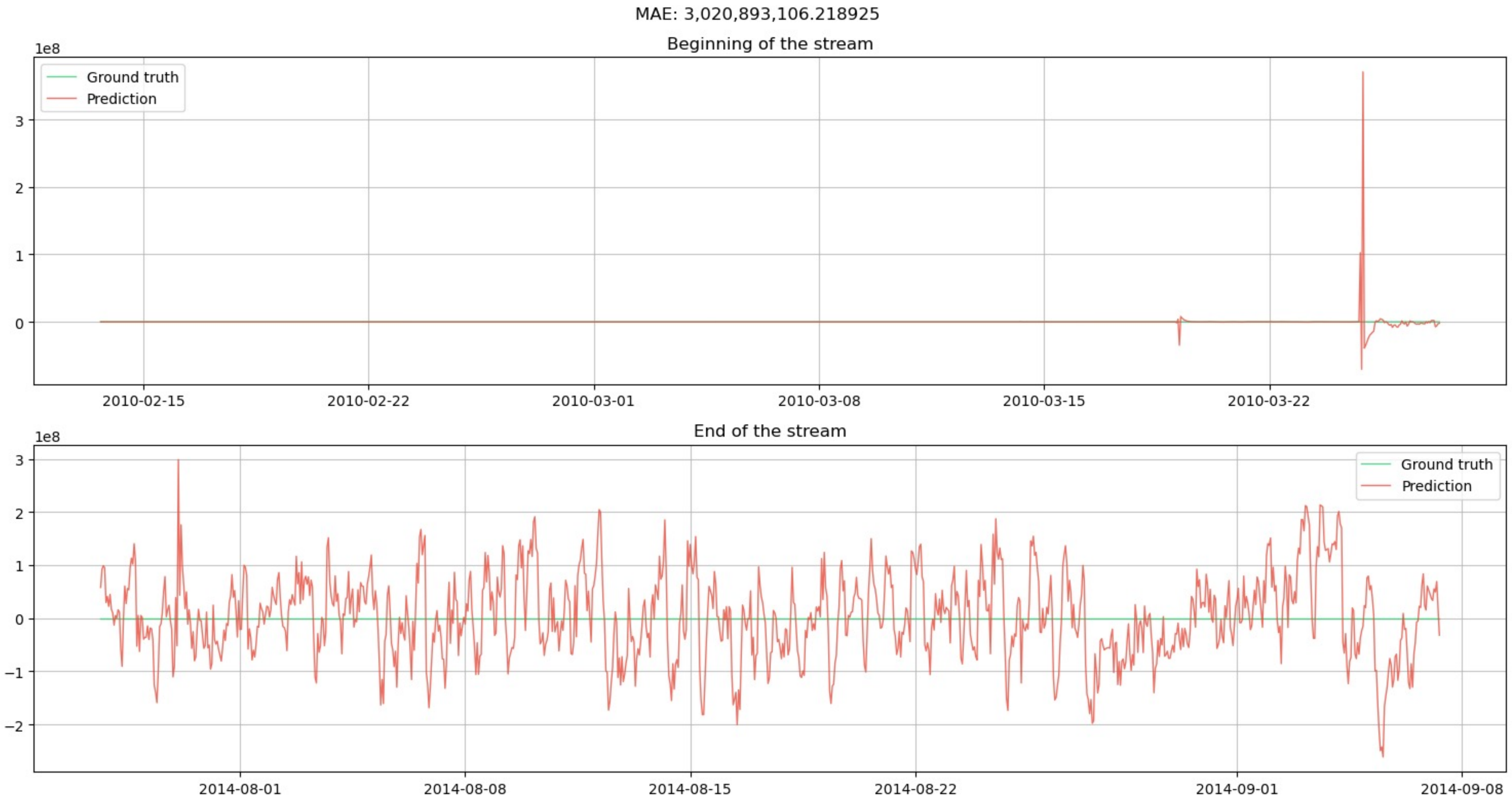
```
# Define transformer for scaling
columns_to_scale = ['current_pollution', 'dew',
       'temp', 'press', 'wnd_spd', 'snow', 'rain']

preprocessor = ColumnTransformer(
transformers=[('num', StandardScaler(),
       columns_to_scale)],
       remainder='passthrough')

# Initialize the RandomForestRegressor model
model = RandomForestRegressor(n_estimators=100,
       random_state=42)

# Define the pipeline
pipe = Pipeline([
       ('preprocessor', preprocessor),
       ('regressor', model)])

# Perform cross-validation and calculate scores
scores = cross_val_score(pipe, X, y,
       scoring=mae_scorer, cv=cv)
```



6. Stream learning

Online Linear Regressor

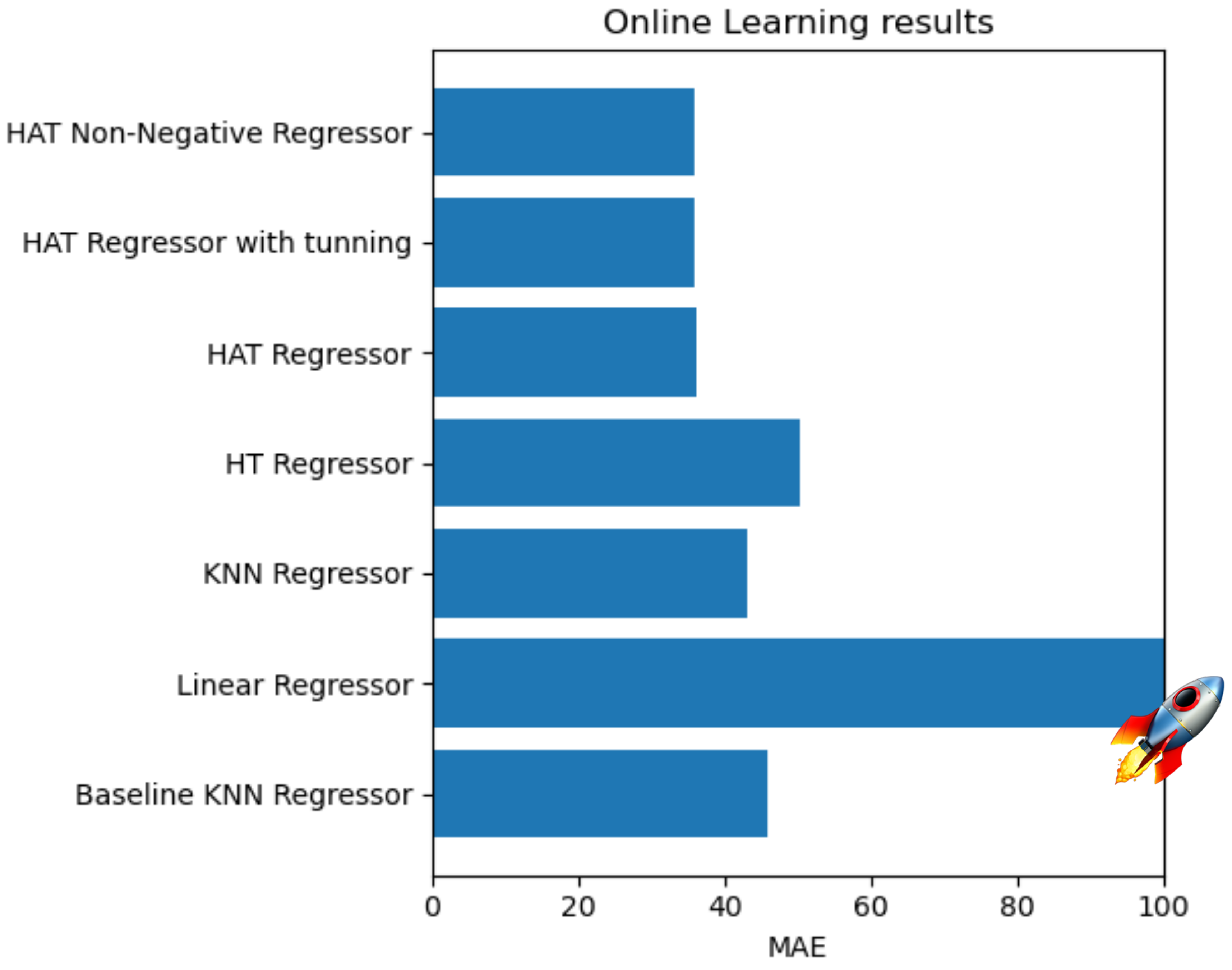
💩



Hoeffding Adaptive Tree Regressor with non-negative restriction

Stream learning

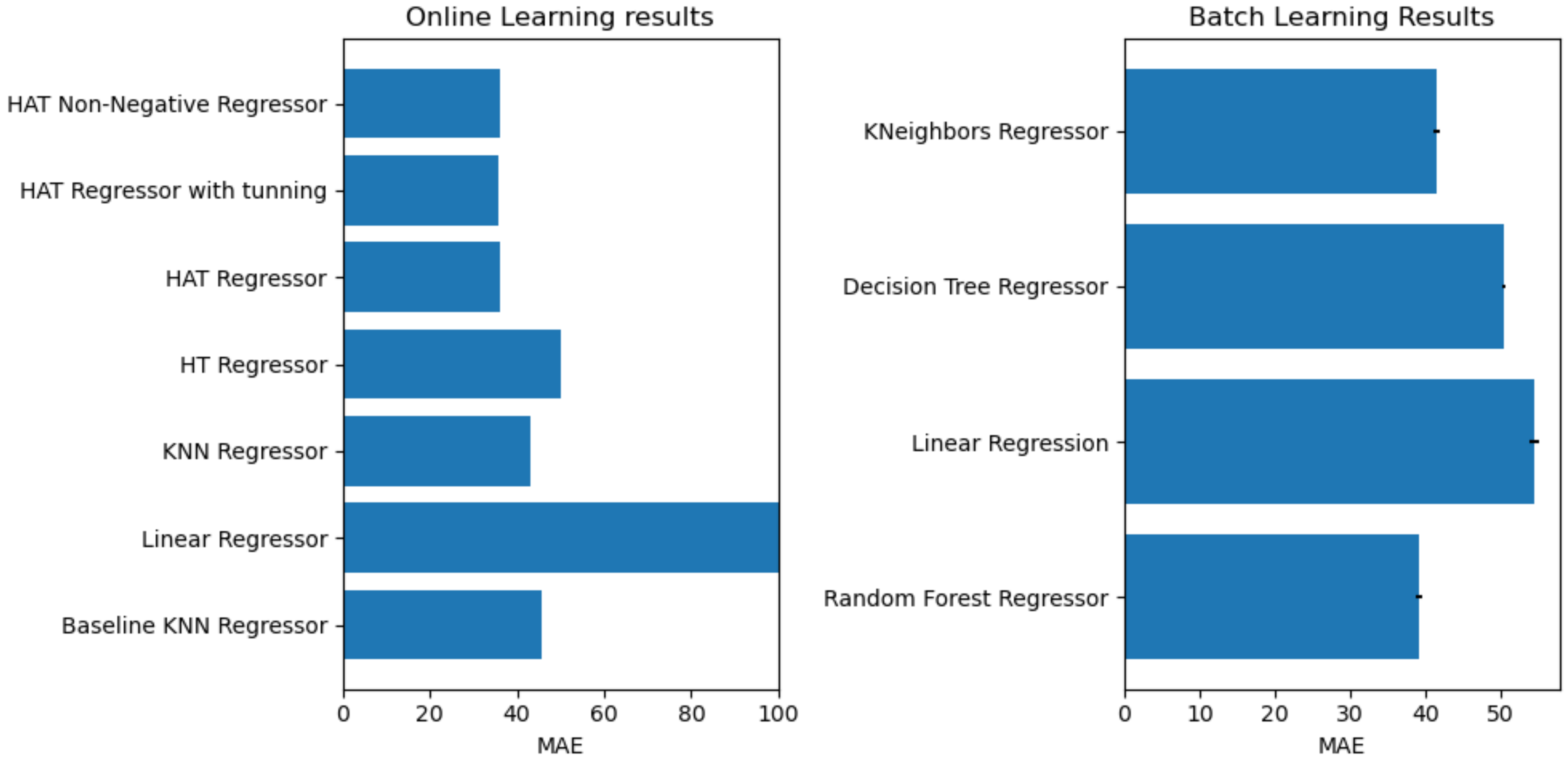
```
# Define the data stream
data_stream =
stream.iter_csv('data/air_pollution_dataset_modified.csv',
        target='pred_pollution',
        converters={'current_pollution': float, 'dew':
                float, 'temp': float, 'press': float,
                'wnd_spd': float, 'snow': float, 'rain':
                float, 'pred_pollution': float, 'date':
                pd.to_datetime})

# Define the pipeline
model = (compose.Select('wnd_dir') |
        preprocessing.OneHotEncoder())

model += (compose.Select('current_pollution', 'dew',
        'temp', 'press', 'wnd_spd', 'snow',
        'rain')|preprocessing.StandardScaler())

model |= HoeffdingAdaptiveTreeRegressor(grace_period =
        250, drift_detector=drift.ADWIN())

# Evaluate
evaluate.progressive_val_score(dataset=data_stream,
        model=model, metric=MAE(), print_every=2500)
```



7. Results

Metric comparison

- **Online learning models match batch models' MAE**, except for Linear Regressor.
- **HAT Regressors** outperform all, even the best batch model.



8. Conclusions

Given their competitive performance, immediate prediction capabilities, and resilience to Concept Drift, OL Models prove to be more suitable and efficient for forecasting predictions than traditional ML approaches.

💪 OL Models hold their ground with competitive MAE scores.

🏃‍♀️ OL Models have the advantage of making predictions **from the get-go**, without the need for substantial historical data.

⏳ Batch learning models are likely to **struggle in the long term** when Concept Drift events take place.

🚀 Exploring advanced online learning algorithms could further enhance accuracy and adaptability in **future work**.

UNIVERSIDADE DA CORUÑA

Universidade de Vigo

22 FEB 2024

Thank you for your time!

Any questions?