Xirui Huang

Nov 12th, 2024

Wat.AI Scrum

Q1: River Library is incremental, which means it handles data as a singular data point, allowing us to process online streamed data as it flows down a river (as it loads live) without having to recompute the whole dataset. This can useful when the patterns change over time in our dataset, like when we are importing live data from a TTC webpage.

River handles incremental data by (.learn\_one())

ARF (Adaptive Random Forest) handles incremental updates by updating each tree using .learn\_one() method without retraining from scratch. ARF in River can also drop or replace a tree if said tree’s accuracy falls under a certain threshold. Lastly, ARF trees can also be weighted based on their performance. Trees that do not perform well could be reset to avoid outdated data patterns, thus improving incremental live online learning.

Q2: Drift Detection in River ARF – Drift is when the pattern changes as data is being passed live into our model. **Recurring drift is a type of drift where the pattern changes at reoccurring rates (ex: season changes, times of day).**

DDM – Drift Detection Method: For abrupt sudden changes in data patterns. Tracks model’s error rate then flags model if the error goes past a statistical threshold. Better for per-instance data (one at a time to detect sudden shift in pattern). DDM will not detect gradual changes in data pattern.

ADWIN – Adaptive Windowing: Has a sliding window of recent data points in memory that is divided into sub-windows. ADWIN flags model if there is a notable difference between two sub-windows. Better for frequent small amounts of data, batches of around 10 data points.

Scaler: .StandardScaler(), .MinMaxScaler(), .MaxAbsScaler(), .AdaptiveStandardScaler() all possible, could use to scale data

River Encoder: Only One Hot Encoding and Ordinal Encoding available

What categorial data would be encoding (station names)?

Q3: Necessary data preprocessing requirements to feed data into a River ARF

Same preprocessing steps as any other ML model: Scaling, Encoding, Imputing

Pipeline:

scaling + encoding + imputing -> ARF + preprocessed data -> incremental training + ADWIN

Q4: Metrics to use to track model performance

1. Metrics sub-library has metrics such as Accuracy, Confusion Matrix, Precision, Recall, etc. We should use Accuracy and Rolling ROCAUC (for ADWIN rolling data)
2. To track error rate, we can use LogLoss() for probabilities, MSE() mean squared error for regression models, MAE() for average error. Error is when predictions do not match result
3. Tracking system: Custom Error/Accuracy Log with Python? External Libraries – MLflow, Prometheus + Grafana for real time monitoring (river data streaming)