Forecast Reconciliation: Aligning TS and ML to a Unified Granularity

Implementing an Algorithm for Forecast Alignment and Scaling

- **Project:** Demand Forecasting System
- **Component:** Reconciliation Module
- **Technologies:** Python, Pandas, NumPy

GitHub: https://github.com/dvank1mang1/reconciliation-and-hybridization

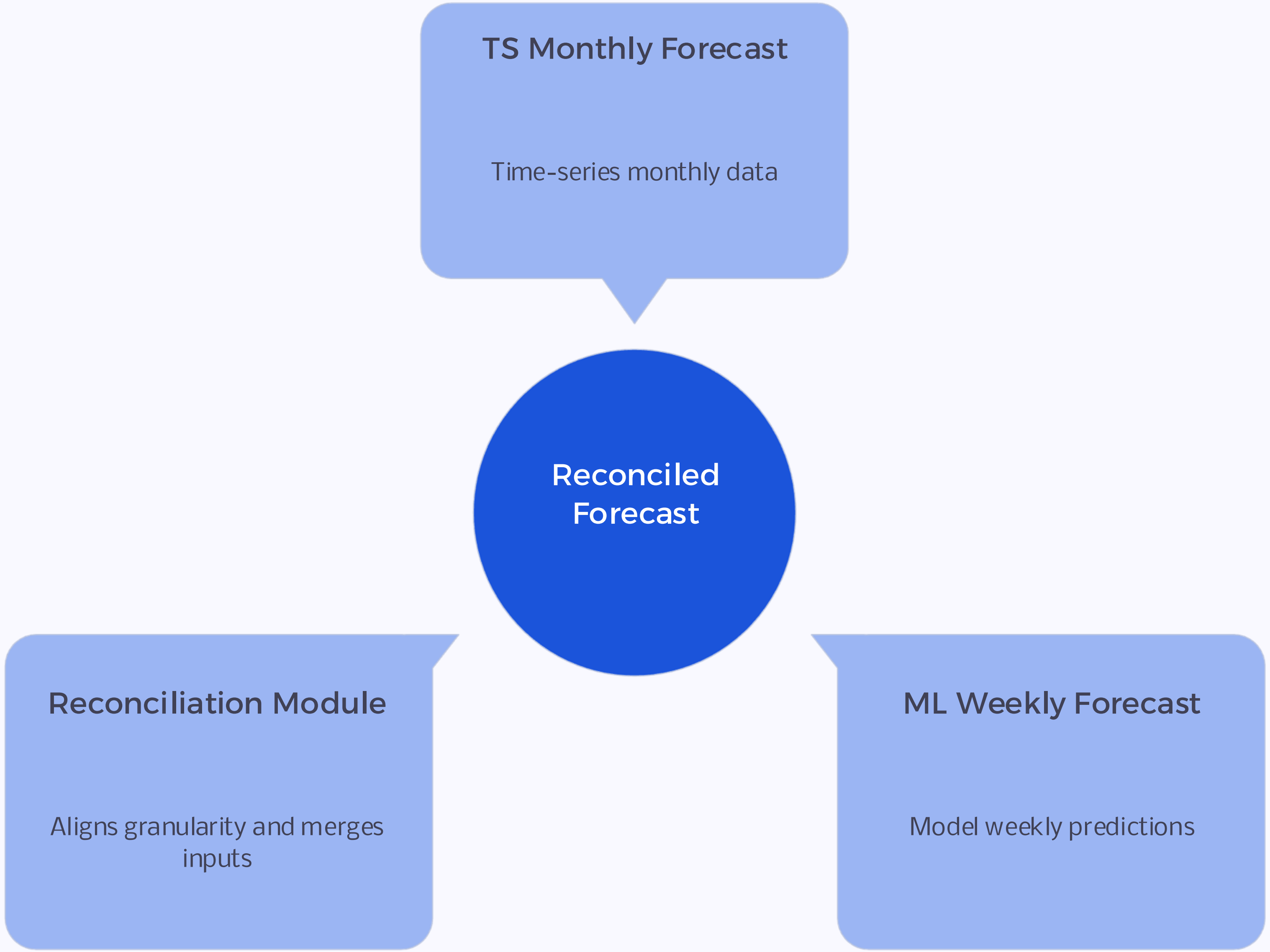
Why is Reconciliation Necessary?

The Problem:

- TS and ML forecasts have differing temporal granularities (e.g., months vs. weeks).
- Varied aggregation levels across products, locations, and customers.
- Forecasts are not directly comparable for further processing.
- A unified format is essential for hybridization.

Business Requirements:

- Align TS and ML forecast time periods.
- Achieve uniform granularity for comparison.
- Scale TS forecasts to match ML at the same aggregation level.
- Preserve all necessary attributes (segments, demand types).



Reconciliation Process: Key Steps

01

1. Data Preparation

Filter forecasts post-historical end date (IB_HIST_END_DT), calculate PERIOD_END_DT, and normalize column names (product_lvl_id, location_lvl_id, etc.).

02

2. Proportional Daily Distribution

Calculate days in period for TS and ML, then scale forecasts proportionally based on actual days in the period. `forecast_value * (actual_days / period_days)`.

03

3. Forecast Joining

Perform a Cartesian product of ML and TS, filtered by temporal intersection and matching IDs: `PERIOD_DT_ml <= PERIOD_END_DT_ts AND PERIOD_END_DT_ml >= PERIOD_DT_ts`.

04

4. Reconciliation Ratio Calculation

Group by key attributes (product, location, customer, channel, PERIOD_DT) and calculate `ratio = ML_total / TS_total`. If TS is zero or missing, ratio defaults to 1.0 or 0.0.

05

5. Apply Ratio to TS Forecasts

Scale TS forecasts: `TS_FORECAST_VALUE_REC = TS_FORECAST_VALUE * reconciliation_ratio`, aligning TS with ML at the same level.

Technical Implementation: Aligning Different Granularities

Problem: TS forecasts might be monthly, while ML are weekly.

Solution: Temporal Intersection

We use a precise joining method to find overlapping periods between different granularities.

- **Example:**
  - TS Period: 2024-02-01 to 2024-02-29 (month)
  - ML Period 1: 2024-02-05 to 2024-02-11 (week)
  - ML Period 2: 2024-02-12 to 2024-02-18 (week)
- **Join Result:**
  - Record 1: TS (February) × ML (Week 1) → intersection occurs → record created
  - Record 2: TS (February) × ML (Week 2) → intersection occurs → record created
- **New Periods:**
  - `PERIOD_DT` = max(`PERIOD_DT_ml`, `PERIOD_DT_ts`) – start of intersection
  - `PERIOD_END_DT` = min(`PERIOD_END_DT_ml`, `PERIOD_END_DT_ts`) – end of intersection
- **Grouping:** After joining, data is grouped by (product, location, customer, channel, PERIOD_DT).

Forecast Adjustment: Accounting for Actual Period Length

Problem: Forecasts may cover periods of varying lengths (e.g., 28, 30, or 31 days for months).

Solution: The `number_days` Function

- Logic:
  - DAY → 1 day
  - WEEK → 7 days
  - MONTH → actual number of days in the month (28-31)
- Scaling Forecasts:
  - For ML: `ML_FORECAST_VALUE = ML_FORECAST_VALUE * (actual_days / ml_period_days)`
  - For TS: `TS_FORECAST_VALUE = TS_FORECAST_VALUE * (actual_days / ts_period_days)`

This ensures accurate comparison of forecasts across different time intervals.

Reconciliation Ratio: Scaling TS Forecasts

Goal: Align TS forecasts with the level of ML forecasts at the same aggregation.

1. Grouping and Summation

Group by (product, location, customer, channel, PERIOD_DT). Calculate `ML_total` (sum of all ML forecasts in group) and `TS_total` (sum of all TS forecasts in group).

2. Ratio Calculation

`if TS_total > 0 and ML_total is not None: ratio = ML_total / TS_total`

`elif TS_total > 0: ratio = 1.0`

`else: ratio = 0.0`

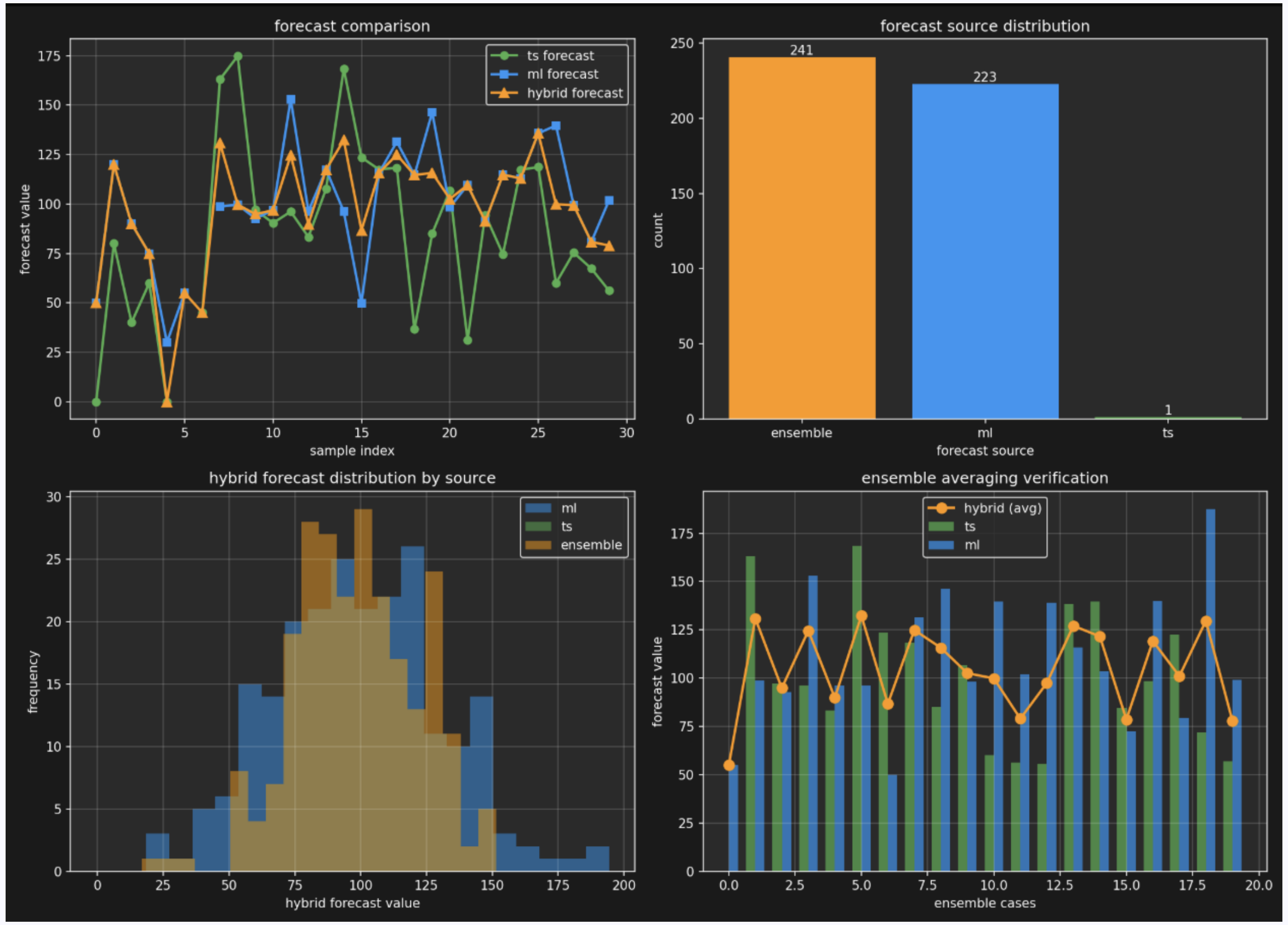
3. Applying the Ratio

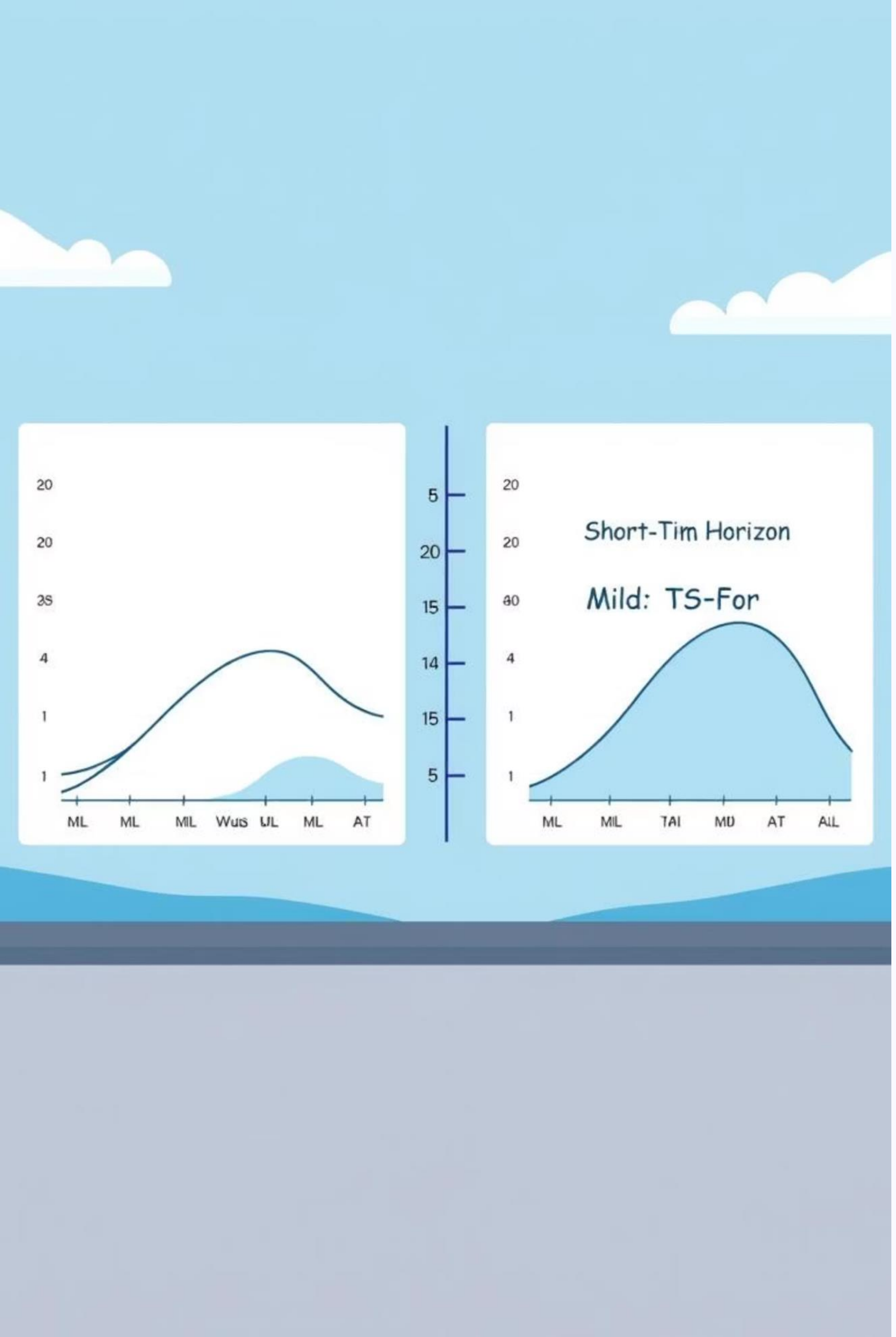
`TS_FORECAST_VALUE_REC = TS_FORECAST_VALUE * ratio`. Each TS record within the group is scaled uniformly by this ratio.

Example:

- Group: Product P001, Location L001, Period 2024-02-01
- `ML_total` = 500, `TS_total` = 400
- `ratio` = 500/400 = 1.25
- TS forecast 100 → `TS_FORECAST_VALUE_REC` = 100 × 1.25 = 125

Result: TS forecasts are scaled to match ML at the same aggregation level, ensuring consistency and comparability.





Special Case: Mid-Term Forecast Processing

Problem: ML forecasts may not be available for the distant horizon.

Solution: Separate Handling for Mid-Term Forecasts

- **Condition:** If `IB_FC_HORIZ > delays_config_length`, periods after `IB_HIST_END_DT + delays_config_length` are processed separately.
- **Logic:**
  - Extract TS forecasts for the mid-term horizon.
  - `ML_FORECAST_VALUE` = NaN (ML unavailable).
  - `DEMAND_TYPE` = 'regular'.
  - `ASSORTMENT_TYPE` = 'old'.
  - `TS_FORECAST_VALUE_REC` = `TS_FORECAST_VALUE` (no scaling applied).
- **Integration:** Mid-term forecasts are appended to the main result, providing a comprehensive final output.
- **When Used:** For forecasting horizons beyond ML model training data or when business rules dictate TS-only for mid-term periods.

Flexibility: Configurable Parameters

Our reconciliation module offers key parameters for tailored adjustments.

Parameter	Description	Default Value
`IB_HIST_END_DT`	End date for historical data, filters future forecasts.	Current Date
`IB_FC_HORIZ`	Total forecasting horizon in days.	90
`delays_config_length`	Boundary for short-term vs. mid-term forecasts.	0
`ts_time_lvl`	Time granularity level for TS forecasts.	'MONTH'
`ml_time_lvl`	Time granularity level for ML forecasts.	'WEEK.2'
Aggregation Levels	Optional: Define specific aggregation levels for TS and ML forecasts (e.g., product, location).	N/A

Example Configuration:

```
config = {
 'IB_HIST_END_DT': datetime(2024, 1, 31),
 'IB_FC_HORIZ': 90,
 'ts_time_lvl': 'MONTH',
 'ml_time_lvl': 'WEEK.2',
 'delays_config_length': 60
}
```

Output Structure and Segment Integration

Understanding the transformation from input to reconciled output.

Input Data

- **ts_forecast:** PRODUCT_LVL_ID, LOCATION_LVL_ID, CUSTOMER_LVL_ID, DISTR_CHANNEL_LVL_ID, PERIOD_DT, FORECAST_VALUE
- **ml_forecast:** Same as TS + DEMAND_TYPE, ASSORTMENT_TYPE
- **ts_segments (optional):** SEGMENT_NAME

Processing

Reconciliation module applies temporal alignment, proportional distribution, ratio scaling, and mid-term forecast handling.

Optional `ts_segments` are integrated via left join on ID columns, adding `SEGMENT_NAME` to the result.

Output Columns

- **Identifiers:** PRODUCT_LVL_ID, LOCATION_LVL_ID, etc. (and lowercase versions)
- **Time Periods:** PERIOD_DT (max of TS/ML), PERIOD_END_DT (min of TS/ML)
- **Forecasts:** TS_FORECAST_VALUE_REC (scaled), ML_FORECAST_VALUE (aligned)
- **Attributes:** DEMAND_TYPE, ASSORTMENT_TYPE, SEGMENT_NAME (if provided)

Practical Examples and Key Achievements

Understanding the algorithm's impact with a concrete example.

Algorithm in Action:

Input Data:

- **TS:** Product P001, Location L001, Period 2024-02-01 to 2024-02-29, Forecast = 1000
- **ML:** Product P001, Location L001, Period 2024-02-05 to 2024-02-11, Forecast = 300

After Proportional Distribution:

- **TS:** 1000 × (25 days / 29 days) = 862
- **ML:** 300 × (7 days / 7 days) = 300

After Grouping and Ratio Calculation:

- `ML_total` = 300, `TS_total` = 862
- `ratio` = 300 / 862 ≈ 0.348

Result:

- `TS_FORECAST_VALUE_REC` = 862 × 0.348 ≈ 300
- `ML_FORECAST_VALUE` = 300
- Forecasts are now aligned at the same level.

Key Achievements:

- **Granularity Alignment:** TS and ML brought to a unified temporal format.
- **Proportional Distribution:** Accounts for actual period lengths.
- **TS Scaling:** Reconciliation ratio for ML alignment.
- **Mid-Term Forecast Handling:** Supports horizons without ML.
- **Segment Integration:** Enriches forecasts with business attributes.

Business Value:

- Comparability: TS and ML forecasts are directly comparable.
- Unified Format: Seamless integration for subsequent processing.
- Flexible Configuration: Adaptable to diverse granularities.
- Robust Processing: Reliable handling of various time levels.

Next Steps:

- Transfer reconciled forecasts to the hybridization module.
- Monitor reconciliation ratio quality.
- Analyze forecast distribution across periods.

References and LLM documentation:

Main:

1. Hyndman & Athanasopoulos (2021) – Chapter 11: Forecasting hierarchical and grouped time series
2. Wickramasuriya et al. (2019) – Optimal forecast reconciliation

Additional:

1. Kourentzes & Athanasopoulos (2019) – Cross-temporal coherent forecasts

LLM documentation:

1. Image creation for presentation – Flux 2 Pro model