Hybridizing Forecasts: Blending Time Series & Machine Learning

This presentation introduces an advanced hybridization algorithm that merges Time Series (TS) and Machine Learning (ML) predictions into a unified, intelligent hybrid forecast, incorporating critical business rules.

Project: Demand Forecasting System

Component: Hybridization Module

Technologies: Python, Pandas, NumPy

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

Why Hybridization? Addressing the Forecasting Challenge

The Problem:

We leverage two distinct forecasting methods: Time Series (TS) and Machine Learning (ML). Each method possesses unique strengths across varying scenarios. The core challenge lies in intelligently selecting or combining these forecasts to achieve optimal accuracy.

Business Requirements:

- ML excels for promotions, new products, and short lifecycles due to its adaptability to dynamic patterns.
- TS proves more stable and reliable for low-demand products or those being phased out.
- For all other situations, an ensemble approach (combining both methods) offers the best of both worlds.

Decision Logic: Three Core Hybridization Rules

Rule 1: Utilize ML Forecast

When: Promotion (and segment is not "Retired") OR Short Lifecycle OR New Assortment.

Reason: ML models are superior at capturing marketing effects and emerging patterns.

Example: A promotional campaign for a newly launched product.

Rule 2: Utilize TS Forecast

When: Segment is "Retired" OR "Low Volume" AND TS forecast ≤ 0.01.

Reason: For minimal expected demand, TS offers greater stability and reliability.

Example: A retired product with a forecast of 0.005.

Rule 3: Utilize Ensemble (Average)

When: All other cases not covered by Rules 1 or 2.

Method: Simple arithmetic average of TS and ML forecasts.

Reason: Combines the strengths of both forecasting methods.

Example: A regular product with TS=75 and ML=85 results in a Hybrid=80.

Robustness: Data Preparation & Missing Value Handling

Step 1: Coalesce Logic for Missing Forecasts

- If TS is missing, ML is used as a substitute.
- If ML is missing, TS is used as a substitute.
- If both are missing, NaN is returned (handled later in ensemble).

Step 2: Handling Categorical Features

- If `SEGMENT_NAME`, `DEMAND_TYPE`, or `ASSORTMENT_TYPE` are missing, they are filled with NaN.
- This allows the algorithm to function effectively with incomplete input data.

Step 3: String Normalization

- All relevant string values are converted to lowercase using `.str.lower()`.
- **Purpose:** Ensures case-insensitive comparisons (e.g., "Promo" = "promo" = "PROMO").

Maximized Data Use

Leverages all available data points.

Input Robustness

Resilient to incomplete input datasets.

No Record Loss

Prevents data loss due to missing values.

Algorithm Core: The `calculate_hybrid_forecast` Function

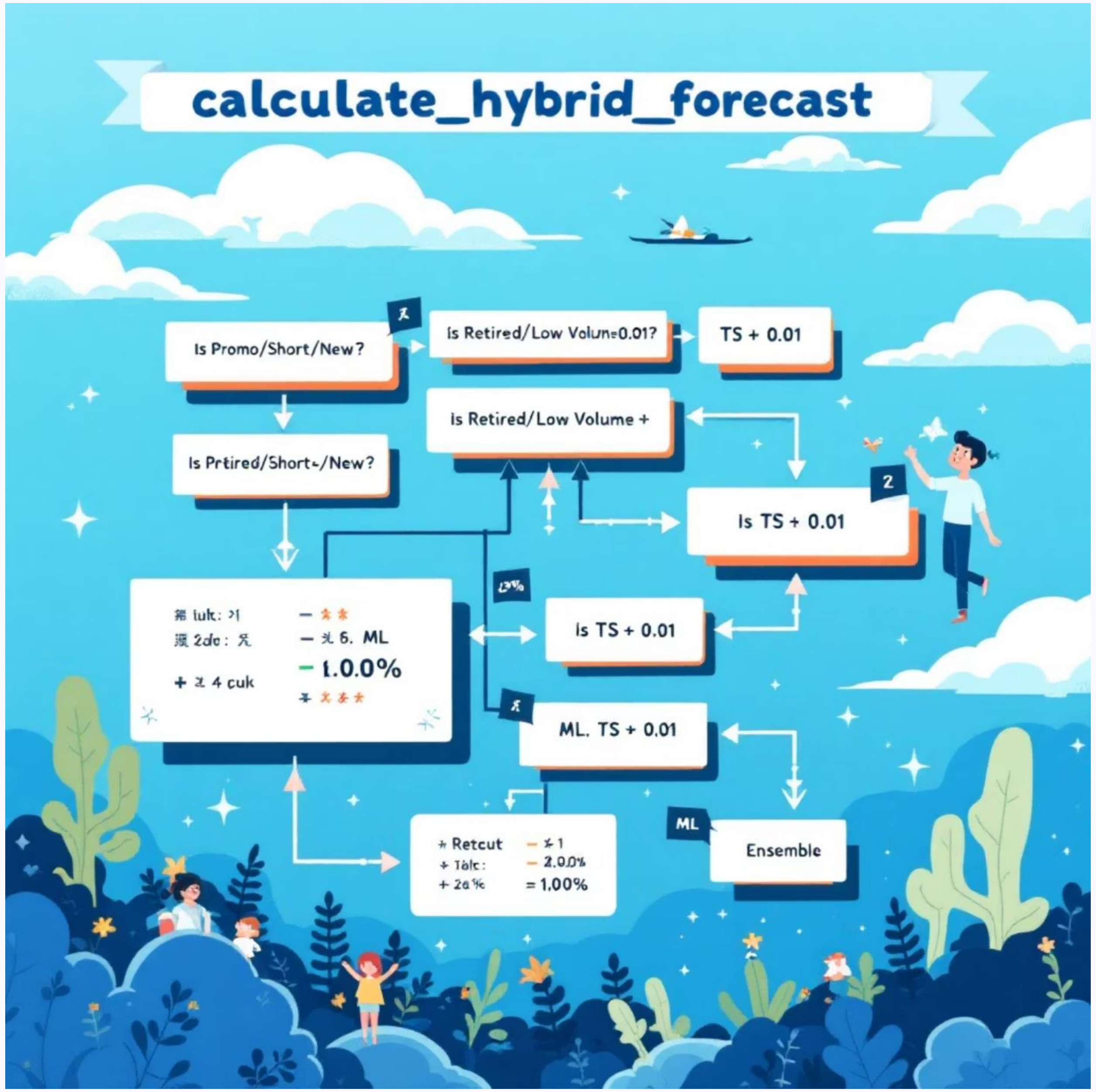
This function represents the heart of our hybridization algorithm, meticulously applying the defined business rules to determine the optimal forecast source.

Decision Logic (Pseudocode):

```
IF (DEMAND_TYPE == 'promo' AND SEGMENT != 'retired') OR
   SEGMENT == 'short' OR ASSORTMENT_TYPE == 'new':
    RETURN ML_FORECAST_VALUE_F
ELSE IF (SEGMENT == 'retired' OR 'low volume') AND
        TS_FORECAST <= 0.01:
    RETURN TS_FORECAST_VALUE_F
ELSE:
    IF TS_FORECAST <= 0.01:
        RETURN TS_FORECAST_VALUE_F
    ELSE:
        RETURN AVERAGE(TS_FORECAST, ML_FORECAST)
```

Implementation Details:

- Pre-calculation validation of values before averaging.
- Handles scenarios where both TS and ML forecasts are absent, returning NaN.
- A zero demand threshold (0.01) is applied even for regular segments within the ensemble logic.



Outputs & Transparency: Tracing the Forecast Source

New Output Columns:

Purpose: Ensures full transparency, auditability, and ease of debugging.

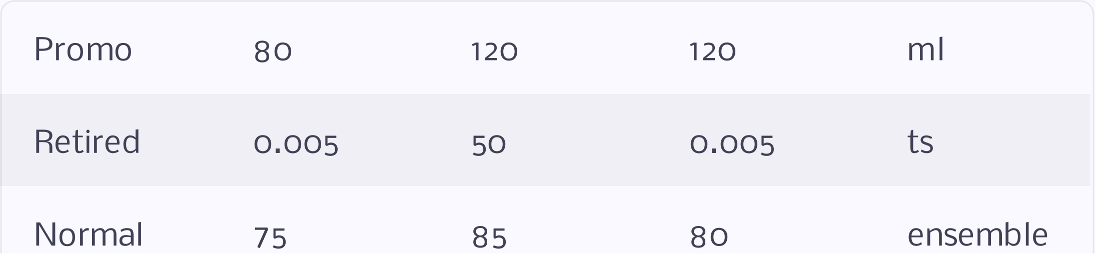
1. `HYBRID_FORECAST_VALUE`: The final hybrid forecast, used in downstream pipelines.
2. `FORECAST_SOURCE`: Indicates the origin of the selected forecast:
    - `'ml'` – Machine Learning forecast (Rule 1)
    - `'ts'` – Time Series forecast (Rule 2)
    - `'ensemble'` – Average of both (Rule 3)
3. `ENSEMBLE_FORECAST_VALUE`: The calculated average, even if not ultimately selected as the hybrid forecast.

Preserved Input Columns:

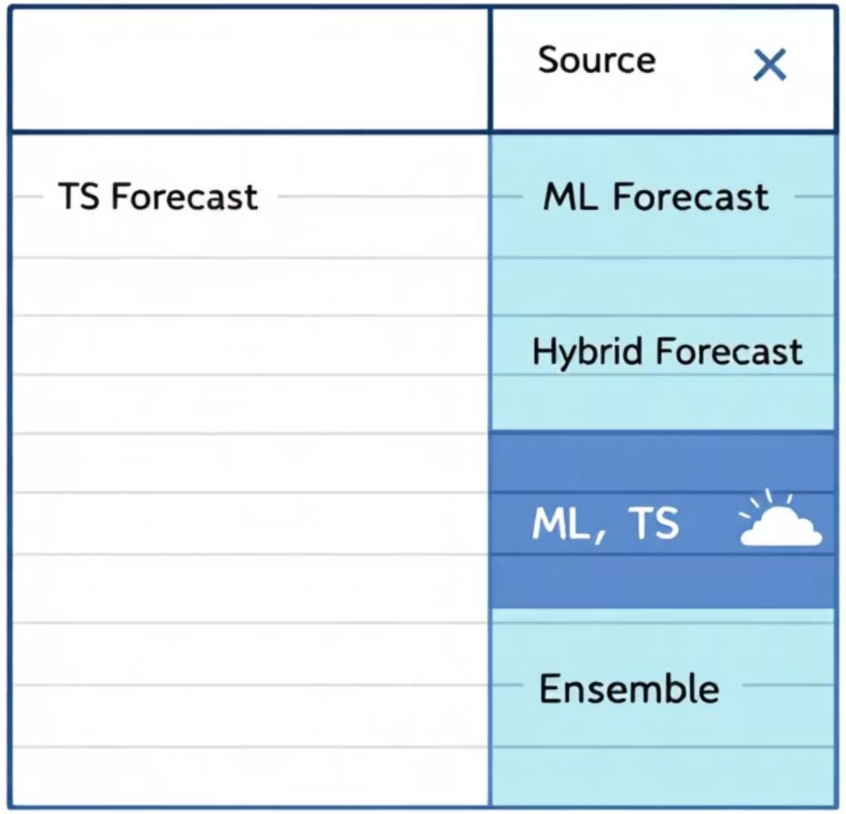
All original columns from `reconciled_forecast` are maintained, including `TS_FORECAST_VALUE_REC` (copied to `TS_FORECAST_VALUE`) and `ML_FORECAST_VALUE`.

Usage Example:

```
# Analyze forecast source distribution
forecast_sources = df['FORECAST_SOURCE'].value_counts()
# Example output: ml: 45%, ensemble: 40%, ts: 15%
```



Promo	80	120	120	ml
Retired	0.005	50	0.005	ts
Normal	75	85	80	ensemble



Resilience: Edge Cases & Configurable Parameters

Handling Edge Cases:

- **Missing Columns:** Automatically created with NaN or suitable alternatives are used.
- **Zero Demand in Normal Segments:** If TS is ≤ threshold, TS is used instead of the average for consistency.
- **Both Forecasts Missing:** Returns NaN, ensuring correct behavior and downstream handling.
- **Case Sensitivity:** All comparisons are case-insensitive, preventing discrepancies.

Configurable Parameter: `ib_zero_demand_threshold`

- **Default Value:** `0.01`
- **Purpose:** Defines the threshold for identifying "zero demand" scenarios.
- **Application:** Utilized in Rule 2 and within the ensemble logic.

How to Modify:

```
hybrid_forecast = hybridization(
    reconciled_forecast,
    ib_zero_demand_threshold=0.005 # Stricter threshold
)
```

Result: The algorithm operates stably and reliably, even when presented with incomplete or non-standard input data.

Practical Examples: Algorithm in Action

Promo Campaign	80	120	promo	Regular	old	120	ml
Short Cycle	60	95	regular	Short	old	95	ml
New Product	40	110	regular	Regular	new	110	ml
Retired Item	0.005	50	regular	Retired	old	0.005	ts
Low Volume	0.008	35	regular	Low Volume	old	0.008	ts
Normal Case	75	85	regular	Regular	old	80	ensemble
Low TS (Normal)	0.005	50	regular	Regular	old	0.005	ensemble

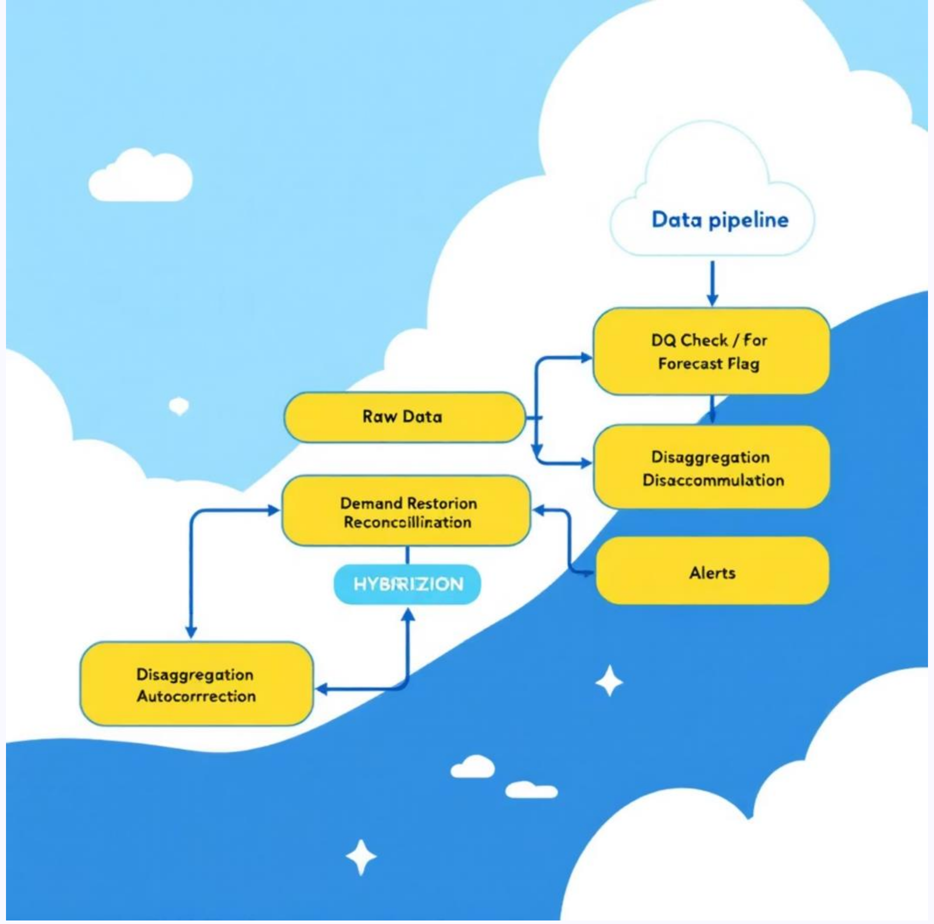
Key Takeaways:

- **Rule 1 (ML):** Applied for promotions, short cycles, and new product introductions.
- **Rule 2 (TS):** Chosen for retired or low-volume items with minimal forecasts.
- **Rule 3 (Ensemble):** Utilized in all other standard forecasting scenarios.

Integration & Key Achievements

Pipeline Integration:

The hybridization module fits seamlessly into the existing demand forecasting pipeline:



Data pipeline
DQ Check / For Forecast Flag
Raw Data
Disaggregation Disaccommulation
Demand Restorion Reconcoillination
Alerts
HYBRIZION
Disaggregation Autocorrection

- **Input:** `reconciled_forecast` (containing `TS_FORECAST_VALUE_REC`, `ML_FORECAST_VALUE`)
- **Output:** `hybrid_forecast` (with `HYBRID_FORECAST_VALUE`, `FORECAST_SOURCE`)

Intelligent Source Selection

Three business rules drive optimal forecast choice.

Robust Data Handling

Coalesce logic, normalization, and edge case management.

Full Transparency

`FORECAST_SOURCE` for audit and analysis.

Enhanced Flexibility

Configurable zero demand threshold.

Operational Stability

Consistent performance with imperfect data.

Business Value & Next Steps

Business Value:

- 🎯 Leverages the strengths of both TS and ML models for superior forecasts.
- 📊 Provides transparency in forecasting decisions, fostering trust.
- 🔧 Offers flexibility to adapt to evolving business requirements.
- 🛡️ Ensures reliable data processing and algorithm stability.

Next Steps:

- Continuous monitoring of forecast quality by source.
- In-depth analysis of `FORECAST_SOURCE` distribution.
- Ongoing optimization of the zero demand threshold for improved accuracy.

References and LLM documentation:

Main:

1. Bates & Granger (1969) - The combination of forecasts
2. Makridakid et al. (2020, 2022) - M4 and M5 Competitions

Additional:

1. Timmerman (2006) - Forecast combinations

LLM documentation:

1. Image creation for presentation - Flux 2 Pro model