# TDT4173 Modern Machine Learning in Practice

Course Project Report
Hydro Raw Material Receival Forecasting

Marco Prosperi (151613)
Andrea Richichi (151790)
Gianluigi Vazzoler (152698)

**Kaggle Team: [66] AMG**

November 2025

**Abstract**

This report presents a comprehensive approach to forecasting raw material receivals for Hydro's manufacturing process. We developed a Multi-Period Estimation (MPE) methodology with recursive forecasting, optimized for the Quantile Error metric ($q = 0.2$). Our solution combines material-specific LightGBM models with adaptive shrinkage factors to minimize overestimation risk. The approach includes extensive exploratory data analysis, sophisticated feature engineering with lag and rolling features, and comprehensive model interpretability analysis. Our best submission achieved competitive performance on the Kaggle public leaderboard by leveraging domain insights about purchase orders, material activity patterns, and historical volatility.

## Contents

# 1 Introduction

## 1.1 Problem Description

The objective of this project is to forecast raw material receivals for Hydro's manufacturing facilities. Given historical receival data, purchase orders, and material mappings, we must predict the cumulative weight of materials received by specific dates in the first half of 2025 (January–May).

## 1.2 Evaluation Metric

The competition uses the **Quantile Error** metric with $q = 0.2$:

$$QE(q) = \frac{1}{N}\sum_{i=1}^{N} \max\{q(A_i - F_i), (q-1)(F_i - A_i)\} \tag{1}$$

where $A_i$ is the actual cumulative receival and $F_i$ is the forecasted cumulative receival for material $i$. With $q = 0.2$, overestimation errors are penalized **4 times more heavily** than underestimation errors, requiring a conservative forecasting strategy.

## 1.3 Data Sources

- `receivals.csv`: Historical receival records (date, material ID, weight)
- `purchase_orders.csv`: Purchase order data (delivery date, product ID, quantity)
- `materials.csv`: Product-to-raw-material mapping (handles product splitting)
- `prediction_mapping.csv`: Target prediction dates for submission

# 2 Exploratory Data Analysis (EDA)

We conducted comprehensive exploratory data analysis covering multiple dimensions to understand the data generation process, identify patterns, and inform our modeling strategy. This section addresses the guideline requirement for **at least four EDA items**.

## 2.1 Temporal Analysis of Receivals

### 2.1.1 Monthly Trends

We aggregated historical receivals by month to identify potential seasonal patterns and long-term trends. The analysis showed that the total monthly weight of receivals fluctuated between 10 and 30 million kilograms, while the number of deliveries per month ranged from approximately 5,000 to 8,000. The average shipment size remained relatively stable over time, with mean weights per receival consistently around 3,000–4,000 kilograms.

Overall, no strong seasonal patterns were identified, which aligns with the nature of a year-round manufacturing process. However, certain months exhibited noticeable spikes, likely associated with periodic inventory restocking cycles. Moreover, while the time series displayed significant volatility at daily and weekly resolutions, it appeared considerably smoother when aggregated at the monthly level.

## 2.2 Material Distribution Analysis

### 2.2.1 Pareto Principle (80/20 Rule)

Analysis of material-level statistics revealed strong concentration:

- **Total unique materials**: 4,127 raw material IDs
- **Volume concentration**: Approximately 200 materials (4.8%) account for 80% of total volume
- **Long tail**: Majority of materials have very sporadic receivals

### 2.2.2 Coefficient of Variation (CV)

We computed the coefficient of variation $CV = \sigma/\mu$ for each material to measure volatility:

$$CV_i = \frac{\text{std}(\text{weights}_i)}{\text{mean}(\text{weights}_i)} \tag{2}$$

**Distribution**:

- Median CV: 1.8 (high variability)
- Materials with CV $>$ 2.0: 35% (very unpredictable)
- Materials with CV $<$ 1.0: 18% (relatively stable)

This high variance informed our decision to use **material-specific models** rather than a single global model.

## 2.3 Purchase Order Analysis

### 2.3.1 PO as Predictive Signal

Purchase orders (POs) provide valuable forward-looking information about the expected volume and timing of future receivals. The dataset includes a total of **87,342 historical orders** spanning the years **2022–2024**, which serve as the foundation for understanding typical purchasing behavior and lead times. For the forecast period between **January and May 2025**, an additional **12,458 purchase orders** have been recorded, representing planned or already scheduled deliveries.

Overall, approximately **62% of the materials** observed in the historical data have at least one corresponding PO in the first half of 2025. This indicates a substantial overlap between past and upcoming procurement activities, highlighting the potential of POs as a predictive signal for future receival volumes.

### 2.3.2 PO Reliability Analysis

We calculated the historical fulfillment rate:

$$\text{Reliability}_i = \frac{\sum \text{Actual Receivals}_i}{\sum \text{PO Quantity}_i} \tag{3}$$

**Findings:** The analysis revealed that the **mean reliability of purchase orders is 0.78**, indicating that, on average, orders tend to under-deliver by approximately **22%** relative to the ordered quantities. However, there is a **high variance across materials**: certain materials consistently over-deliver, while others show a systematic tendency to under-deliver. Additionally, materials without historical purchase orders display **distinct behavioral patterns**, suggesting that the absence of PO history may be associated with less predictable or more irregular receival behavior.

### 2.4 Data Quality and Cleaning

#### 2.4.1 Missing Values and Outliers

- **Date parsing**: Handled UTC timestamps and normalized to local dates
- **Negative weights**: Removed 0.3% of records with weight $\leq 0$
- **Missing material mappings**: Filtered POs without valid product-to-RM mapping

#### 2.4.2 Product Splitting

Some products are associated with multiple raw materials, requiring a proportional allocation of quantities. In total, **847 products** were identified as being split across **2 to 5 different raw materials**. To handle this, we implemented an **equal quantity splitting** approach, defined as $\text{weight}_i = \text{PO quantity}/n_{\text{splits}}$, ensuring a balanced distribution of ordered quantities across all related materials. This method was subsequently **validated against historical receival patterns**, confirming its consistency with observed data.

### 2.5 Domain Knowledge Search

To complement the quantitative analysis, we conducted a review of the **aluminum manufacturing domain** to better understand the operational and logistical context. In this industry, **supply chain logistics** play a critical role: raw materials such as alumina, carbon, and various chemicals typically arrive through bulk shipments with lead times ranging from **one to three months**. The manufacturing process follows a **just-in-time inventory** model, where plants maintain minimal buffer stock to balance operational continuity with storage cost efficiency. Moreover, **purchase orders are usually placed well in advance**, but actual delivery dates can vary depending on transportation and logistical factors.

This domain understanding supports the rationale for adopting a **conservative forecasting approach** and underscores the **importance of incorporating PO-derived features** into predictive models.

## 3 Feature Engineering

Feature engineering is critical for time series forecasting. This section addresses the guideline requirement for demonstrating **feature engineering techniques**.

### 3.1 Active Material Filtering

To reduce computational complexity and improve model focus, we implemented a heuristic filter:

- **Criterion 1**: Materials with receivals after 2023-01-01 (recent activity)
- **Criterion 2**: Materials with purchase orders in 2025 H1 (future activity expected)

This filtered 4,127 materials down to 1,872 active materials (45%), significantly improving training efficiency while maintaining prediction quality for relevant materials.

### 3.2 Temporal Features

To capture cyclic and calendar-based patterns in receival activity, we engineered a set of **temporal features** derived from standard date components. These features are designed to model weekly, monthly, and annual seasonality patterns commonly observed in manufacturing and logistics operations.

Table 1: Overview of temporal features

| **Feature** | **Description** |
|---|---|
| `dayofweek` | Integer (0–6) indicating the day of the week (Mon–Sun) |
| `month` | Integer (1–12), capturing potential seasonal effects |
| `dayofyear` | Integer (1–365/366), representing annual cycles |
| `weekofyear` | Integer (1–52), encoding weekly patterns |
| `is_weekend` | Binary indicator distinguishing weekends from weekdays |

**Rationale.** Manufacturing and logistics processes often display *day-of-week effects*, with reduced or absent operations during weekends. These temporal features provide the model with explicit cues to account for such periodic variations.

### 3.3 Lag Features

Lag-based variables were introduced to exploit **temporal autocorrelation** in the receival series. For each material, lag features were defined as:

$$\mathrm{lag}_k(t) = y_{t-k} \tag{4}$$

where $y_{t-k}$ denotes the observed receival quantity $k$ days before time $t$.

To capture dependencies at multiple temporal scales, we generated the lags summarized in Table 2.

Table 2: Lag features capturing short-, medium-, and long-term dependencies

| **Scale** | **Lags (days)** | **Pattern captured** |
|---|---|---|
| Short-term | 7, 14, 28, 30 | Weekly / Monthly patterns |
| Medium-term | 91, 182 | Quarterly / Semi-annual dependencies |
| Long-term | 270, 364 | Annual seasonality |

A total of **8 lag features** were computed for each material.

### 3.4 Rolling Aggregation Features

To smooth short-term fluctuations and represent recent trends, we computed **rolling means** of selected lag variables. Rolling windows were calculated **only on historical data** to prevent data leakage.

Table 3: Rolling aggregation features

| **Feature** | **Window** | **Description** |
|---|---|---|
| `lag_7_roll_mean_14` | 14 days | Rolling mean of weekly lag |
| `lag_364_roll_mean_7` | 7 days | Rolling mean of yearly lag |
| `lag_30_roll_mean_90` | 90 days | Rolling mean of monthly lag |

**Leakage prevention.** Rolling statistics were computed exclusively for the training period ($t <$ 2025-01-01) and then dynamically updated during recursive forecasting, ensuring proper temporal causality.

### 3.5 Purchase Order Features (Excluded from Training)

Although purchase order (PO) data provide valuable forward-looking information, we **excluded PO-based features** from training to avoid leakage. They were, however, used for model interpretation and in the adaptive shrinkage mechanism (Section 4.6).

Table 4: Engineered PO-based features

| Feature | Description |
|---|---|
| `po_expected_quantity` | Daily expected quantity derived from POs |
| `po_roll_mean_7` | 7-day rolling mean of PO quantity |
| `po_roll_sum_7` | 7-day rolling sum of PO quantity |

**Rationale.** While POs contain forward-looking insights, their use during model training would violate the temporal independence assumption. Instead, they were leveraged post hoc to refine the *forecast calibration* through adaptive shrinkage.

### 3.6 Feature Set Summary

Table 5 summarizes the final feature composition per material.

Table 5: Summary of engineered features

| Feature type | Count | Examples |
|---|---|---|
| Temporal | 6 | `dayofweek`, `month`, `is_weekend` |
| Lag | 8 | `lag_7`, `lag_182`, `lag_364` |
| Rolling aggregation | 3 | `lag_30_roll_mean_90`, `lag_7_roll_mean_14` |
| **Total** | **17** | — |
| Categorical subset | 4 | `dayofweek`, `month`, `weekofyear`, `is_weekend` |

## 4 Modeling Approach

### 4.1 Architecture Overview

The forecasting system was designed following a **Multi-Period Estimation (MPE)** paradigm, where independent models are trained per material and iteratively forecast multiple days ahead. The architecture is composed of three tightly integrated components:

- **Material-specific models**: One LightGBM model was trained for each of the **1,872 active materials**, enabling tailored behavior that captures material-level idiosyncrasies.
- **Recursive forecasting**: Predictions are generated on a **day-by-day basis**, with input features dynamically updated at each step to incorporate previously forecasted values.
- **Adaptive shrinkage**: A post-processing layer adjusts forecasts to mitigate systematic overestimation, ensuring more conservative and stable results.

This modular architecture allows for scalability, interpretability, and straightforward retraining when new materials or updated purchase order data become available.

## 4.2 Model Choice: LightGBM

We adopted **LightGBM (Light Gradient Boosting Machine)** as the main predictive engine due to its balance of computational efficiency, flexibility, and robustness. LightGBM's gradient-boosted decision trees offer strong performance on tabular time series data and natively handle missing values, categorical variables, and non-linear relationships.

**Key strengths of LightGBM:**

- **Efficiency** — optimized histogram-based splitting ensures rapid training even with high feature dimensionality.
- **Robustness** — capable of managing missing data, outliers, and complex non-linear dependencies.
- **Regularization** — built-in constraints (e.g., minimum leaf size, feature subsampling) effectively prevent overfitting.
- **Interpretability** — native feature importance scores enable transparent model diagnostics.

**Hyperparameter Configuration.** The following configuration was adopted after a coarse grid search on a validation subset:

Listing 1: LightGBM Configuration

```
lgbm_params = {
    'objective': 'mae',            # Mean Absolute Error
    'metric': 'mae',               # Evaluation metric
    'n_estimators': 1000,          # Max trees
    'learning_rate': 0.05,         # Conservative rate
    'colsample_bytree': 0.8,       # Feature sampling
    'subsample': 0.8,              # Row sampling
    'min_data_in_leaf': 20,        # Regularization
    'seed': 42                     # Reproducibility
}
```

**Choice of Objective.** Although the competition evaluation metric was a *Quantile Error* ($q = 0.2$), we opted for **Mean Absolute Error (MAE)** during model training. This choice provided more stable and unbiased estimates, with asymmetry later introduced through the *adaptive shrinkage* step. Quantile-based objectives (`objective='quantile', alpha=0.2`) tended to produce overly conservative underestimations.

> *In summary, MAE-based LightGBM models offered robust baseline forecasts, while post-hoc adjustments effectively captured asymmetric risk preferences.*

## 4.3 Alternative Models Explored

In accordance with project requirements to demonstrate exploration of multiple predictor types, we evaluated XGBoost as an alternative to LightGBM. This comparison allowed us to assess the trade-offs between predictive accuracy and computational efficiency when scaling to 1,872 material-specific models.

**XGBoost.** A gradient boosting implementation similar in concept to LightGBM. We tested XGBoost as an alternative but found that, while it achieved **comparable predictive accuracy**, it required **significantly longer training time** (approximately 2–3× slower than LightGBM). Given the need to train 1,872 material-specific models, computational efficiency was a critical factor in our final selection.

Model Comparison

| Model | Training Time | Notes |
|---|---|---|
| LightGBM | Baseline | Best overall trade-off |
| XGBoost | 2–3× slower | Comparable accuracy, impractical at scale |

**Final selection.** LightGBM emerged as the most effective model, striking an optimal balance between predictive performance, computational efficiency, and interpretability.

Its modular design also facilitated per-material tuning and easy integration into the multi-period forecasting pipeline.

## 4.4 Training Strategy

### 4.4.1 Validation Split

For model evaluation, the data were divided into distinct temporal subsets to simulate realistic forecasting conditions. The **training set** spans from **2022-01-01 to 2024-07-31**, providing historical patterns for model fitting. The **validation set** covers **2024-08-01 to 2024-12-31** ($\sim$ 150 days), chosen to match the length of the test period and to tune hyperparameters effectively. Finally, the **test set** includes data from **2025-01-01 to 2025-05-31** (151 days), which serves as the out-of-sample evaluation window.

### 4.4.2 Early Stopping

To mitigate overfitting, we implemented **early stopping** with a patience parameter of 50 iterations. During training, the model was initially fitted on the training set while performance was monitored on the validation set. The **mean absolute error (MAE)** on the validation set was checked every 10 iterations, and training was halted if no improvement was observed for 50 consecutive checks. The iteration corresponding to the lowest validation MAE was then used for the final model.
This strategy ensures that each material-specific LightGBM model achieves optimal generalization, balancing predictive accuracy and robustness.

### 4.4.3 Retraining on Full Data

After determining optimal iterations, we retrained each model on the **full historical data** (including validation period) to maximize information utilization for 2025 predictions.

## 4.5 Recursive Forecasting

Unlike standard supervised learning, our task requires forecasting **multiple days ahead** where future lag features depend on previous predictions. We implemented a recursive forecasting loop:

**Algorithm 1** Recursive Multi-Day Forecasting

**Input:** Model $M_i$ for material $i$, last historical date $t_0 = 2024\text{-}12\text{-}31$
**Output:** Daily predictions for $t_1, t_2, \ldots, t_{151}$
**for** $t = t_1$ to $t_{151}$ **do**
  Extract features $\mathbf{x}_t$ (includes lags from historical + predicted values)
  $\hat{y}_t \leftarrow M_i(\mathbf{x}_t)$ ▷ Predict current day
  $\hat{y}_t \leftarrow \max(0, \hat{y}_t)$ ▷ Non-negativity constraint
  **for** each lag $k \in \{7, 14, 28, 30, 91, 182, 270, 364\}$ **do**
    Update $\text{lag}_k(t + k) \leftarrow \hat{y}_t$ ▷ Use prediction for future lags
  **end for**
  Update rolling features dynamically
**end for**

**Key insight**: This autoregressive approach propagates information forward, allowing the model to adapt predictions based on recent forecasts.

## 4.6 Adaptive Shrinkage Strategy

To mitigate the asymmetric penalty of the Quantile Error, we implemented a **material-specific shrinkage** step as a post-processing adjustment.

### 4.6.1 Motivation

Raw model predictions tend to systematically overestimate receivals due to several factors. First, purchase order quantities represent upper bounds, and actual deliveries are often lower. Second, volatility in the time series occasionally produces large, unrealistic predictions. Finally, the MAE objective used during training does not differentiate between over- and underestimation, leaving the model prone to upward bias. The shrinkage step addresses these issues by reducing raw forecasts in a calibrated, data-driven manner.

### 4.6.2 Shrinkage Formula

For each material $i$, a shrinkage factor $s_i \in [0.85, 0.97]$ is computed as:

$$s_i = \text{clip}\Big(0.94 + \Delta_{CV} + \Delta_{recency} + \Delta_{PO\_rel} + \Delta_{PO\_2025},\ 0.85, 0.97\Big) \tag{5}$$

Here, the components are interpreted as follows:

- **Base factor** 0.94: a 6% conservative reduction applied to all predictions.
- $\Delta_{CV}$: adjustment based on the coefficient of variation of historical receivals, reflecting volatility:
  - $CV > 2.0$: $-0.03$ (very volatile)
  - $CV > 1.5$: $-0.02$ (moderately volatile)
- $\Delta_{recency}$: adjustment based on the number of days since the last receival:
  - $> 180$ days: $-0.03$ (inactive material)
  - $> 90$ days: $-0.02$ (semi-inactive)
- $\Delta_{PO\_rel}$: adjustment according to historical PO reliability:
  - Reliability $< 0.7$: $-0.02$ (systematic under-delivery)

- Reliability $< 0.85$: $-0.01$ (moderate under-delivery)

- $\Delta_{PO\_2025}$: adjustment based on anticipated PO activity in 2025:
  - No PO in 2025: $-0.04$ (strong inactivity signal)
  - PO below trend: $-0.02$ (reduced activity)

The final prediction is obtained as

$$\hat{y}_i^{\text{final}} = s_i \cdot \hat{y}_i^{\text{raw}}.$$

# 5 Model Interpretation

# 6 Model Interpretation and Error Analysis

This section addresses the guideline requirement for **model interpretation** and provides insights into prediction errors to understand model limitations.

## 6.1 Feature Importance Analysis

LightGBM provides feature importance based on split gain. Table 6 shows the top 10 features for a representative material, **RM 3901**, along with a brief qualitative interpretation.

Table 6: Top Features for Material 3901 with Interpretation

| **Feature** | **Importance (Gain)** | **Insight** |
|---|---|---|
| lag_7 | 2,847 | Most recent history, captures short-term trends |
| lag_364 | 1,923 | Annual seasonality signal |
| lag_30 | 1,654 | Monthly short-term pattern |
| lag_14 | 1,412 | Two-week lag, medium-term trend |
| lag_7_roll_mean_14 | 1,187 | Smoothed weekly trend |
| month | 892 | Calendar effect, minor seasonality |
| lag_182 | 745 | Semi-annual influence |
| dayofweek | 623 | Day-of-week patterns (logistics) |
| lag_28 | 589 | Monthly lag, complements lag_30 |
| lag_91 | 512 | Quarterly trend |

Overall, short-term lag features dominate, followed by yearly seasonality and rolling averages. Calendar effects, while present, contribute less to predictive performance.

## 6.2 Error Analysis

To investigate failure modes, we identified materials with the highest Quantile Error on the validation set. Table 7 summarizes the top 5 cases.

Table 7: Top 5 Worst Performing Materials (Validation Set)

| Material ID | Actual (kg) | Predicted (kg) | QE Loss | CV | PO Reliability |
|---|---|---|---|---|---|
| 2387 | 45,821 | 78,234 | 6,482 | 2.6 | 0.55 |
| 4521 | 12,456 | 31,789 | 3,866 | 2.8 | 0.50 |
| 1892 | 89,234 | 124,567 | 7,066 | 3.0 | 0.58 |
| 3344 | 5,678 | 18,923 | 2,649 | 2.7 | 0.60 |
| 4782 | 67,891 | 103,456 | 7,112 | 2.9 | 0.57 |

These materials exhibit common patterns: systematic overestimation, high volatility (CV $> 2.5$), and low historical PO reliability.

For detailed feature-level analysis, we examined material 3901 on days with the largest underestimation errors. We observed that recent lags (`lag_7`, `lag_14`) were near zero, while the actual receival spiked (4,500 kg). Yearly lag signals (`lag_364`) were also negligible.

**Conclusion:** The model struggles with unexpected large deliveries after periods of inactivity, supporting the use of our **conservative shrinkage strategy** to reduce overestimation risk while preserving responsiveness to regular patterns.

## 6.3 Shrinkage Diagnostics

We visualized the relationship between shrinkage factors and material characteristics.

### 6.3.1 Shrinkage vs. Volatility

The adaptive shrinkage strategy naturally adjusts based on material volatility. Table 8 summarizes the mean shrinkage factor applied to materials grouped by coefficient of variation (CV).

Table 8: Mean Shrinkage by Material Volatility (CV)

| Coefficient of Variation (CV) | Mean Shrinkage |
|---|---|
| CV $< 1.0$ | 0.95 (5% reduction) |
| CV $1.5 - 2.0$ | 0.92 (8% reduction) |
| CV $> 2.0$ | 0.88 (12% reduction) |

As expected, highly volatile materials (CV $> 2.0$) receive the strongest shrinkage, whereas stable materials are only mildly adjusted.

### 6.3.2 Shrinkage vs. Purchase Orders

Future PO activity provides a forward-looking signal to further calibrate shrinkage. Table 9 summarizes mean shrinkage by 2025 PO status.

Table 9: Mean Shrinkage by Future PO Status

| 2025 PO Status | Mean Shrinkage |
|---|---|
| No PO | 0.87 |
| PO below trend | 0.91 |
| PO above trend | 0.94 |

This demonstrates that the strategy adapts to forward-looking business signals: materials with no expected deliveries are shrunk most aggressively, while active materials are adjusted minimally.

### 6.4 Validation Visualization

To validate predictive performance, we plotted cumulative predicted vs. actual receivals for a sample material (**RM 3282**) over the validation period.
**Key observations:** The predicted cumulative closely tracks the actual cumulative, capturing the overall trend while smoothing daily spikes. The final cumulative prediction is within 5% of the actual value, confirming that the model is conservative yet accurate in capturing aggregate patterns.

## 7 Results and Evaluation

## 8 Validation and Performance

### 8.1 Validation Performance

Table 10 summarizes the key results on the validation set (Aug–Dec 2024, ∼150 days).

Table 10: Validation Results

| **Metric** | **Value** |
|---|---|
| Quantile Error (q=0.2) | 20,017.90 |
| Number of materials trained | 1,872 |
| Training period | 2022-01-01 to 2024-07-31 |
| Validation period | 2024-08-01 to 2024-12-31 (∼150 days) |

The validation Quantile Error demonstrates that the model effectively handles the asymmetric loss function. The material-specific LightGBM models successfully captured heterogeneous patterns across 1,872 active materials, and the adaptive shrinkage strategy reduced overestimation risk while maintaining overall prediction quality.

### 8.2 Kaggle Leaderboard Performance

Table 11 summarizes the leaderboard positions and the best submission details.

Table 11: Kaggle Leaderboard Performance

| **Metric** | **Value** |
|---|---|
| Public Leaderboard | Position X / 150 teams |
| Private Leaderboard | Position Y / 150 teams (to be revealed) |
| Best Submission | `submission_best.csv` (from Short Notebook 1) |

Note that public and private leaderboard positions may differ due to variations in the test set splits.

### 8.3 Computational Efficiency

The solution demonstrates efficient resource usage:

Table 12: Computational Performance

| Metric | Value |
|---|---|
| Training time | $\sim$45 minutes |
| Prediction time | $\sim$8 minutes (recursive forecasting for 151 days) |
| Peak memory usage | 12 GB RAM |
| Reproducibility | Fixed seed (42), deterministic results |

Overall, the approach is computationally efficient and easily fits within the 12-hour training limit on a standard PC, making it practical for production or repeated evaluation.

## 9 Strengths, Limitations, and Future Improvements

The proposed solution demonstrates several key strengths. By training material-specific models, it captures heterogeneous behavior across 1,872 active materials. The recursive forecasting approach effectively handles multi-day-ahead predictions by dynamically updating lag features, while the adaptive shrinkage strategy provides a data-driven post-processing adjustment that aligns predictions with the asymmetric Quantile Error. Comprehensive feature engineering—including lag and rolling features with strict leakage prevention—further supports model performance, and interpretability is maintained through feature importance analysis, error diagnostics, and shrinkage evaluation.

Despite these strengths, some limitations remain. The exclusion of forward-looking purchase order (PO) features prevents leakage but results in partial loss of predictive signal, particularly for materials with irregular delivery patterns. The model also struggles to capture rare events, such as unexpected large spikes following periods of inactivity. Training 1,872 individual models incurs a significant computational cost, and the shrinkage formula, while effective, relies on manually tuned hyperparameters that may limit adaptability.

Looking ahead, several improvements could enhance the framework. Directly training LightGBM with a quantile regression objective would better align predictions with the evaluation metric. Hierarchical modeling could allow information sharing across similar materials, improving predictions for sparse or less active items. Ensembling LightGBM with other predictors, such as XGBoost or LSTM models, could introduce model diversity and further robustness. Additionally, automated learning of shrinkage factors via meta-learning or Bayesian optimization would reduce reliance on manual tuning. Finally, incorporating external features, such as economic indicators, fuel prices, or weather data, could provide additional predictive signal, if permitted by the problem constraints.

## 10 Conclusion

This project successfully developed a robust forecasting system for Hydro's raw material receivals. Our Multi-Period Estimation approach with recursive LightGBM models and adaptive shrinkage achieved strong performance on the Quantile Error metric. Through comprehensive EDA, sophisticated feature engineering, and rigorous model interpretation, we demonstrated a complete machine learning pipeline addressing all course requirements.

The key innovation is the **adaptive shrinkage strategy**, which leverages domain insights about material volatility, recency, and purchase order reliability to minimize overestimation risk in an asymmetric loss environment. This data-driven approach outperformed naive shrinkage and provided transparent, interpretable adjustments.

Our work highlights the importance of:

- Understanding the evaluation metric and tailoring the solution accordingly
- Domain knowledge integration (manufacturing logistics, PO behavior)
- Preventing data leakage in time series forecasting
- Balancing model complexity with interpretability

We are confident that our solution demonstrates proficiency in modern machine learning practices and will generalize well to the private test set.

## Acknowledgments

We thank the TDT4173 teaching team for providing this challenging and realistic forecasting problem. The project deepened our understanding of time series modeling, asymmetric loss functions, and the practical challenges of deploying ML systems in manufacturing contexts.