Demand Forecasting and Stock Optimisation for a Sports Retailer

A machine learning pipeline on Sprinter's sales and stock history

Paula García

MSc in Economics with Data Science

Universidad de Alicante

Internship: Demand Planning, Merchandising — Sprinter (JD Sports Group)

September 20, 2026

Abstract

Demand Planning teams in fashion and sports retail forecast at the level of product typologies, not individual references, and the forecast they produce drives how much stock is bought and how much of it sits unsold. This work builds an end-to-end forecasting pipeline on Sprinter's historical sales and stock records and measures, at every step, what it delivers. The raw data are two anonymised extracts covering 463,818 product × market × month records between January 2022 and June 2026. After cleaning, market classification and business exclusions, the series are aggregated to a *dictionary* × month panel of 13,805 observations over 306 typologies: a change of grain that reduces the problem by a factor of 34, lowers the share of zero-sales months to 10.2 %, and raises the coverage of the twelve-month lag from 42 % to 74 % of rows, which is what makes a seasonal forecast estimable at all. A single global gradient boosting model with a Tweedie loss, trained across all series with the typology as a categorical feature, reaches a test WMAPE of 17.97 % against 33.53 % for the best naive baseline, 36.87 % for a per-series AutoETS and 38.96 % for an AutoARIMA: a relative improvement of 46.4 % over the baseline, significant at the dictionary level under sign and Wilcoxon tests, and stable across four walk-forward windows (mean 20.3 %, s.d. 2.8 pp). An ablation model restricted to calendar and autoregressive features scores 21.90 %, placing the marginal contribution of price, stock and assortment information at 3.9 percentage points of WMAPE. When the machine learning model is stripped of its horizon advantage and forecast recursively over the same six months the classical models face, its WMAPE rises to 36.1 %, and both figures are reported side by side rather than compared across horizons. Translating the accuracy gain into an inventory policy under a periodic review with lead time L and cycle service level α, the reduction in required safety stock is 50.1 % of a seasonal naive incumbent's, a percentage that is invariant to both policy parameters under the Gaussian formulation. That figure is then tested against the stock the company actually held: over the six test months, replenishment sized with the model's forecast would have reached the observed 96.5 % coverage of demand with 31.3 % fewer units, or

raised coverage to 98.5 % with the same units, cutting unserved demand by 57 %, and the gap widens rather than narrows when the censoring of recorded sales is corrected. What remains open is not the outcome but the comparison: Sprinter's own forecast, the true incumbent, was not available, so the exercise establishes what those units would have bought under a model-driven allocation and not that the model improves on the planning process that produced them.

Keywords: demand forecasting; retail; gradient boosting; global forecasting models; hierarchical aggregation; censored demand; safety stock; service level; WMAPE.

Contents

1 Introduction

Sprinter is a Spanish sports retailer belonging to the JD Sports group. Its Merchandising department contains a Demand Planning area whose job is to decide, for each product typology and each month ahead, how many units to buy and how to distribute them across the sales channels. Two failures cost money in opposite directions. A *stockout* is a sale that could have happened and did not: the customer either buys nothing or buys elsewhere, and the loss never appears in any system, because a sale that does not occur leaves no record. *Dead stock* is the mirror image: units sitting in stores or in the warehouse, tying up working capital and eventually clearing at a markdown that consumes the margin the purchase was made for. The exploratory analysis in this work puts numbers on both: on average 18.6 % of the SKUs in a typology are out of stock in a month in which that typology has demand, while 60.5 % hold stock and sell nothing.

Between those two failures sits the forecast. Everything Demand Planning decides rests on an expectation of how much will sell, and the quality of that expectation sets a floor on how much safety stock has to be carried to hit a given service level. This is the practical reason a forecasting project in this setting is not an academic exercise: an improvement in forecast accuracy is not valuable in itself, it is valuable because it buys the same service level with less inventory.

Three features of Sprinter's data make this a harder problem than a textbook time-series exercise, and each of them shapes the methodology that follows.

First, **volume and format**. The history arrives as Excel workbooks at product × market × month grain, with the records not on the first sheet, the header not on the first row, the same field spelled differently across files, and a price column to which an anonymisation factor had been applied in one file but not the other. Before anything can be modelled, the extraction has to be made reproducible rather than manual, and the working tables have to be small and fast enough that an analytical question can be asked and answered in seconds instead of minutes.

Second, **granularity**. At the individual reference level the data are sparse: thousands of short series, many of them intermittent, in which a twelve-month lag exists for only a minority of rows because most references do not stay on sale for a full year. Forecasting at that level would mean forecasting mostly zeros. It would also be the wrong level for the decision: Demand Planning does not buy reference by reference, it reasons in terms of typologies, which Sprinter encodes in a field internally called the *dictionary*.

A dictionary is not an arbitrary grouping. It gathers the references that fulfil the same commercial function: the same kind of product, aimed at the same customer and playing the same role in the assortment, regardless of the code each of them happens to carry in a given season. This matters because the catalogue underneath is constantly renewed. A typology contains *seasonal* references, which live for a single campaign and are replaced the

following year by an almost identical product under a new code, and *carry-over* references, which stay in the assortment across seasons. Taken one by one, a seasonal reference is a short, truncated series that says little about future demand, and its disappearance from the records is a catalogue decision rather than a demand signal. Taken together, those same references form a series that is continuous, has several complete annual cycles behind it, and behaves in a way that can be measured, analysed and forecast. The dictionary is therefore the finest level at which demand is stable enough to model and, at the same time, the level at which the purchasing decision is actually taken, which is why it is the unit of analysis throughout this work.

Third, **censoring**. The recorded variable is units sold, not demand. In a month in which a large share of a typology's references were out of stock, the recorded figure is what could physically be sold given the available stock, not what customers wanted to buy. A model trained naively on that figure is being asked to predict a number that understates true demand in precisely the months where an accurate forecast matters most for replenishment.

The work is organised around the following research questions:

RQ1. To what extent does changing the forecasting grain from SKU to dictionary level improve the statistical properties of the series (sparsity, seasonal lag coverage) and make a seasonal forecast feasible?

RQ2. Can a single global machine learning model, trained across all dictionaries, outperform naive heuristics and classical local time-series models (AutoETS, AutoARIMA) in out-of-sample accuracy?

RQ3. How much of this improvement is due to Sprinter's proprietary business information (prices, stock, assortment size) rather than to calendar and autoregressive structure alone?

RQ4. Does the comparison with classical models remain favourable once forecast horizons are aligned?

The work is organised around the three objectives agreed with the company, which this document addresses in order and evidences separately.

1. **Data optimisation.** Extract, clean and unify the historical records of sales, stock levels and stockouts, and reduce the volume and the query cost to the point where the analysis becomes iterative. This objective is met when the gain can be *measured*, not merely described: memory footprint before and after type assignment, storage format and read latency, row reduction from detail to panel, and the modelling gain in lag coverage that the change of grain produces.

2. **Machine learning modelling.** Apply the methods studied during the master's degree to the forecasting of demand and stock for sporting goods, and establish how much they add over what is already available. "Adding value" is defined here against three successively

harder reference points: the naive baselines a planner could apply without any model, well-fitted classical univariate models (AutoETS and AutoARIMA) that see nothing but each series' own history, and an ablation of the machine learning model itself restricted to calendar and autoregressive features. The last of the three is what isolates the contribution of the business information, which is the question the company actually cares about.

3. **Business impact.** Evaluate the accuracy of the models in terms a Demand Planning team can act on, by converting the forecast error into the safety stock required to sustain a given cycle service level, and quantifying how much of that stock the better forecast makes unnecessary.

Beyond applying standard tools to a company dataset, this work makes four choices that are worth stating explicitly, because each of them is defended with evidence produced inside the project rather than asserted.

The change of grain is treated as a result, not a convenience. Aggregating from product × market to dictionary level is justified by measuring what it buys (10.2 % of months at zero instead of a third; the twelve-month lag available for 74 % of rows instead of 42 %) and by stating what it costs, namely the loss of visibility over divergent behaviour *inside* a typology, which is why the cleaned detail panel is retained for auditing.

Every comparison is made at the same horizon. The headline one-step-ahead figure of the machine learning model describes a monthly rolling planning process in which last month's sales are known before the next month is forecast. The classical models produce a six-month-ahead forecast from a single origin. Comparing the two directly would flatter the machine learning model, so the same model is also run recursively from the same origin, and both numbers are reported with their horizon attached.

The censoring problem is addressed and then tested rather than assumed away. A separate panel is built in which months with a high current-month stockout rate have their target reconstituted from seasonal analogues in clean years, with declared caps on how much demand the correction may invent; the entire lag and rolling-feature block is rebuilt on the corrected series so that inputs and labels are internally consistent; and the resulting model is evaluated, like every other model here, against *observed* test sales, because reconstituting the evaluation target too would make the exercise circular.

The inventory conclusion is derived per series, not from a pooled error. The pooled test RMSE mixes typologies selling a few hundred units a month with typologies selling hundreds of thousands, and is not the error dispersion of any real series; safety stock is therefore sized from per-dictionary error dispersion, under three alternative formulations whose disagreement is itself reported.

The remainder of the document is organised as follows. Section 2 reviews the literature on retail demand forecasting, global models, intermittent and censored demand, accuracy measurement and inventory theory. Section 3 describes the raw data, the ETL pipeline, the

market taxonomy and the aggregation to dictionary level. Section 4 presents the exploratory analysis and the modelling guardrails derived from it. Section 5 sets out the forecasting problem, the feature set, the validation scheme and the models. Section 6 reports the results and robustness checks. Section 7 translates forecast accuracy into safety stock. Section 8 discusses the findings, limitations and future work, and Section 9 concludes.

2 Background and Literature Review

Demand forecasting in the retail sector, particularly for sporting goods and fashion, presents unique structural challenges. The demand is heavily influenced by strong seasonal patterns, such as the dichotomy between summer and winter collections, as well as exogenous calendar events like Back to School campaigns and Black Friday promotions. Furthermore, individual stock keeping units (SKUs) often exhibit exceptionally short life cycles. Many products are introduced for a single season and replaced the following year, leading to sparse historical data at the granular level. Consequently, accurate forecasting necessitates aggregating demand to higher product hierarchies, such as typologies or families, to capture underlying consumer trends without being hindered by the short lifespan of individual references (Hyndman and Athanasopoulos 2021).

Aggregation also has a statistical justification. If the demand of a typology is the sum of the demand of its references, idiosyncratic noise at reference level partly cancels out, and the coefficient of variation of the aggregate is generally lower than that of its components. The cost is a loss of detail: a forecast at dictionary level must later be disaggregated to references, sizes and stores. In the case of Sprinter this cost is limited, because the purchasing decisions studied here are taken at dictionary level.

Traditionally, time-series forecasting has relied on local models, such as Exponential Smoothing (ETS) or Autoregressive Integrated Moving Average (ARIMA). These models are fitted individually to each series, relying exclusively on the specific historical pattern of that item (Hyndman and Athanasopoulos 2021). While effective for stable, long-history series, local models struggle with intermittency and cold-start problems. In contrast, global forecasting models train a single algorithm across an entire panel of time series simultaneously (Januschowski et al. 2020). By incorporating the series identifier (e.g., the product dictionary) as a categorical variable, a global model can learn shared dynamics, cross-series seasonality, and complex non-linear relationships across the entire dataset. This cross-learning paradigm is particularly advantageous in retail, where many typologies share similar seasonal peaks and promotional impacts (Makridakis et al. 2022).

Among global models, Gradient Boosting Decision Trees (GBDT) have proven to be state-of-the-art for tabular demand forecasting. Frameworks like LightGBM (Ke et al. 2017) offer exceptional computational efficiency and the ability to natively handle categorical features without requiring one-hot encoding. Crucially, GBDT models can be optimized using objective functions specifically tailored to the distribution of retail demand. Standard loss

functions like Mean Squared Error (MSE) often perform poorly on demand data, which is non-negative, right-skewed, and frequently contains zero values. Instead, employing a Tweedie or Poisson loss function correctly models the probability mass at zero and the variance-to-mean relationship characteristic of retail sales, yielding more robust and business-aligned predictions.

Not all demand series are equally forecastable, and the literature has proposed classification schemes to decide which methods are appropriate. The most widely used is the scheme of Syntetos et al. (2005), which classifies series according to two statistics: the Average Demand Interval (ADI), the mean number of periods between two non-zero demands, and the squared coefficient of variation of non-zero demand sizes (CV^2). With the cut-off values ADI = 1.32 and $CV^2 = 0.49$, series are classified as *smooth*, *erratic*, *intermittent* or *lumpy*. Intermittent and lumpy series are traditionally forecast with specialised methods such as Croston's method (Croston 1972), whereas smooth and erratic series can be handled by standard regression and time-series methods.

A fundamental challenge in retail analytics is that historical records reflect *sales*, not true *demand*. In periods where a product is out of stock, the recorded sales volume is artificially capped by the available inventory. This phenomenon is known as right-censored demand. Training a model naively on sales data systematically biases the forecast downwards, particularly during high-demand periods when inventory depletion is most likely. While formal econometric approaches to censoring, such as Tobit models or Expectation-Maximization (EM) algorithms, offer rigorous solutions, they are often computationally prohibitive for large-scale, high-dimensional machine learning pipelines. Alternatively, operational heuristics based on fill-rate corrections can be employed to reconstitute unconstrained demand (Nahmias 1994). This study addresses the censoring problem by implementing a seasonal-analogue fill-rate heuristic, isolating and correcting periods of severe stockouts before feature engineering.

The choice of accuracy measure is not neutral. Hyndman and Koehler (2006) review the main families of measures and show that percentage errors such as MAPE are undefined when actual values are zero and are dominated by low-volume observations. Scale-dependent measures such as MAE or RMSE cannot be compared across series of different size. Volume-weighted measures such as WMAPE solve both problems at portfolio level, since they divide total absolute error by total actual demand. For this reason WMAPE is widely used in retail practice.

Beyond point estimates of accuracy, it is important to test whether differences between models are statistically meaningful. The test of Diebold and Mariano (1995) compares the expected loss of two forecasts for a single series. When many series are available, simple non-parametric tests on the per-series differences in error, such as the sign test and the Wilcoxon signed-rank test, provide a robust alternative that does not require assumptions about the distribution of errors.

The ultimate value of a demand forecast in a business context is its ability to optimize inventory levels. Under a standard periodic review system, inventory is replenished at regular intervals to cover demand during the lead time (L) plus the review period. The uncertainty of the forecast dictates the required safety stock to achieve a target cycle service level (α), the probability of not facing a stockout during the replenishment cycle (Silver et al. 2016). By reducing the forecast error dispersion (e.g., the standard deviation of the error, σ_1), a more accurate model directly translates into a proportional reduction in safety stock. This mathematically links the performance of the machine learning pipeline to tangible financial savings and working capital optimization.

3 Data Extraction, Preprocessing and Aggregation

The raw dataset provided by Sprinter consists of two Excel workbooks of historical records, with a combined size of 40.90 MB, and a third workbook holding the mapping from product references to typologies, which assigns 7,745 references to 323 dictionaries and covers 99.99 % of the units in the history. The records are at a highly granular level: product reference (SKU) × market × month. In total, the extraction contains 463,818 raw rows spanning from January 2022 to May 2027 (including system-generated future rows). For confidentiality purposes, the company applied a constant anonymisation factor to the sales volumes, which preserves the relative proportions and variance needed for modelling, although absolute levels are masked.

During the initial data audit, an inconsistency was detected in the `PVP_Act` (current price) column, where an anonymisation factor of 3 had been applied in only one of the source files. This was corrected programmatically to ensure consistency of the price features across periods.

A programmatic ETL (Extract, Transform, Load) pipeline was developed to bypass the inconsistencies of manual Excel reporting. The pipeline incorporates automatic detection of data sheets and header rows, which varied across the provided workbooks, alongside rigorous column name normalization. Identical duplicate rows were removed. A critical optimization step was the reassignment of data types (e.g., downcasting floats and integers, converting strings to categoricals), which yielded a substantial 75.9 % reduction in RAM memory footprint. Validation checks confirmed that the merged dataset represented a continuous temporal history of 65 months with no gaps. Furthermore, 1,844 rows reporting negative sales units were identified as legitimate customer returns and were preserved to accurately reflect net demand.

The steps of that pipeline are the following:

1. **Extraction.** The two record workbooks are read and concatenated into a single table, harmonising column names and types across files, and the reference-to-dictionary mapping is loaded from the third.

2. **Audit and correction.** Duplicate rows are removed, and known anomalies, such as the inconsistent anonymisation factor in `PVP_Act`, are corrected.

3. **Type optimisation.** 64-bit floating-point columns are downcast to 32-bit, and repeated strings (such as identifiers and descriptions) are converted to categorical types.

4. **Market classification.** Market identifiers are mapped to a business taxonomy.

5. **Business exclusions.** Typologies that do not represent plannable demand are removed.

6. **Aggregation.** Records are aggregated to dictionary $\times$ month level.

7. **Storage.** The resulting panel is saved in a columnar format so that later stages can load it directly without repeating the previous steps.

The raw data categorizes transactions through the `ID_MERCADO` variable, which contained 9 distinct identifiers. Through exploratory data analysis and business logic, these were reclassified into an actionable taxonomy, summarised in Table 1. Seven of these identifiers represent active sales channels. Identifier 99 was isolated as the central warehouse; its stock levels denote replenishment capacity rather than consumer-facing availability, and thus it was separated to prevent double-counting. Identifier 98, which accounted for 47.2 % of all future-dated rows, was identified as a pending purchase/order channel. As this identifier does not reflect actual historical customer demand, it was completely excluded from the forecasting universe.

Table 1: Reclassification of the `ID_MERCADO` identifiers.

Identifier(s)	Business meaning	Sales as target	Stock as feature
7 identifiers	Customer-facing channels	Yes	Yes
99	Central warehouse	No	Yes (separate)
98	Pending purchases / orders	No	No (excluded)

Forecasting at the individual SKU level in retail is notoriously difficult due to extreme sparsity and short product life cycles. To formulate a robust forecasting problem, the data grain was elevated by aggregating individual references into broader product typologies, encoded by the business as "dictionaries". This operation collapsed the dataset from 463,818 detail rows down to a manageable panel of 13,805 rows distributed across 306 unique dictionaries.

Formally, let s index SKUs, m customer-facing markets and t months, and let $\mathcal{S}(d)$ denote the set of SKUs belonging to dictionary d. The target variable of the aggregated panel is

$$y_{dt} = \sum_{s \in \mathcal{S}(d)} \sum_{m \in \mathcal{M}} q_{smt}, \tag{1}$$

where q_{smt} are the units sold and $\mathcal{M}$ is the set of customer-facing markets. Stock and price variables are aggregated analogously, using sums for quantities and volume-weighted

averages for prices.

This change of grain, which reduced the data volume by a factor of 34, drastically improved the time-series properties required for autoregressive modelling. As shown in Table 2, the aggregation reduced the proportion of zero-sales months from approximately one-third of the dataset to just 10.2 %. More importantly, it elevated the coverage of the crucial twelve-month lag (lag_12) from 42.0 % to 74.0 % of the rows, making the estimation of yearly seasonality mathematically viable.

Table 2: Impact of hierarchical aggregation on sparsity and seasonal lag coverage.

Metric	**SKU Level (Raw)**	**Dictionary Level (Aggregated)**
Data points (rows)	463,818	13,805
Zero-sales months	$\sim$ 33.3 %	10.2 %
12-month lag coverage	42.0 %	74.0 %

Not all products within the company's catalog behave according to standard retail demand dynamics. To prevent the model from learning distorted patterns, specific typologies were excluded based on business rules. Reusable shopping bags, while technically products, are driven purely by checkout transactions rather than independent consumer demand. Despite representing massive volume shares (28.9 % and 8.2 % of total units for the two main bag typologies), they were removed. Additionally, poorly classified catch-all dictionaries were excluded as they group disparate items lacking a coherent underlying seasonality. Cumulatively, these business exclusions removed 39.3 % of the raw unit volume, ensuring the model trains strictly on planable sporting goods demand.

This figure deserves a brief comment. A large share of the removed volume corresponds to low-value items whose "demand" is a by-product of other purchases. Keeping them would have inflated the denominator of WMAPE with easily predictable volume and would have made accuracy figures look better than they are for the products that actually matter for purchasing decisions. The exclusions therefore make the evaluation more demanding, not less.

The culmination of the data processing phase resulted in a highly optimized analytical pipeline, directly fulfilling the first objective of this project. By transitioning from Excel workbooks to the Parquet format, storage size and read latency were drastically minimized, enabling iterative experimentation. The memory tuning per column and the strategic aggregation to the dictionary level not only solved the computational bottlenecks but fundamentally cured the sparsity of the dataset. This robust, automated data foundation paves the way for the complex feature engineering and machine learning models implemented in the subsequent sections.

4 Exploratory Data Analysis

Before defining the model architecture or engineering features, an exploratory data analysis (EDA) was carried out to understand the structure of the catalogue. The analysis addresses four questions: how total demand evolves over time, how concentrated it is across dictionaries, how strong seasonality is, and how healthy stock levels are.

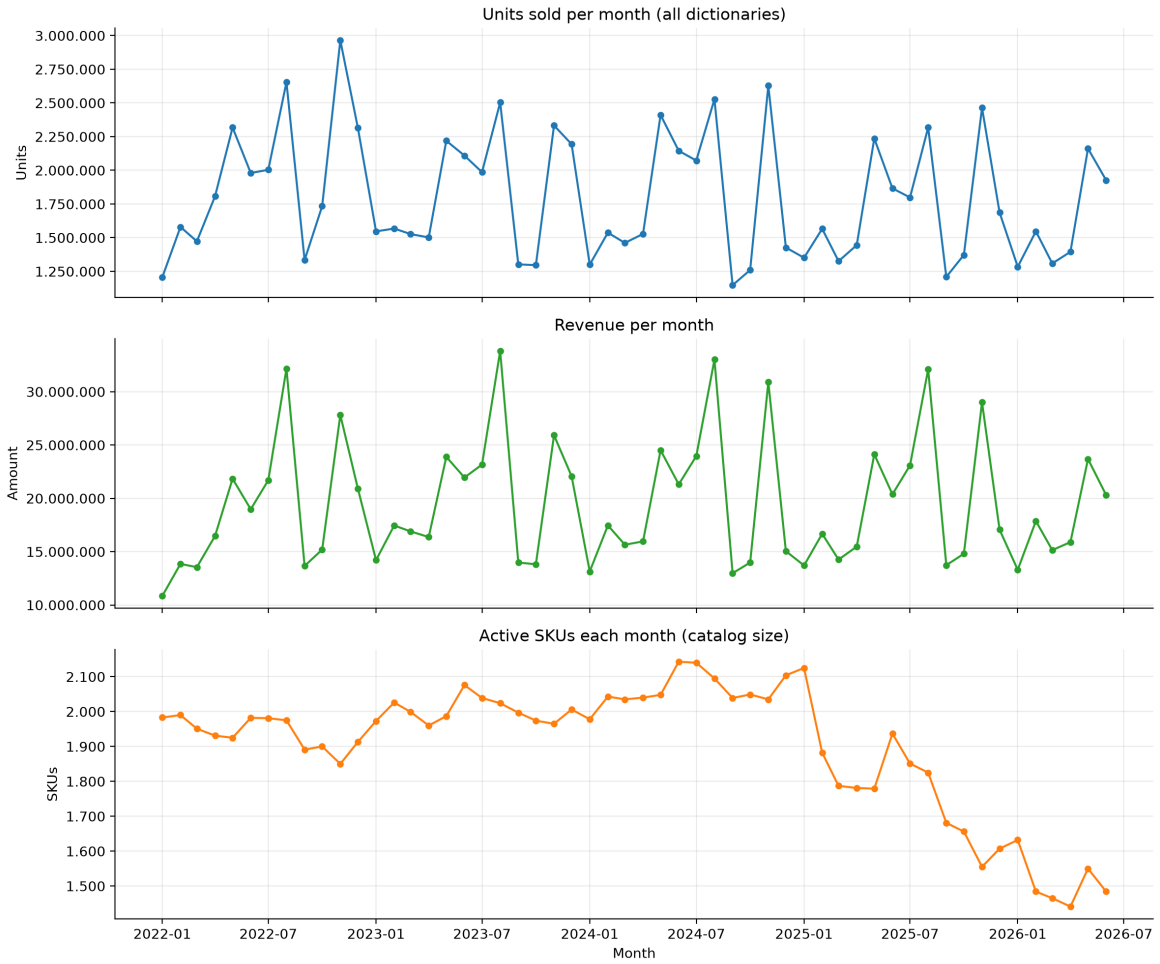

Figure 1: Monthly evolution of sales volume across the historical period.

The aggregate demand curve (Figure 1) shows a dynamic environment with recurrent peaks and troughs, which makes simple moving-average approaches insufficient for planning. A moving average reacts to peaks only after they have occurred, and therefore systematically lags behind seasonal changes.

A fundamental characteristic of retail environments is the unequal contribution of different products to overall sales volume, typically modeled through an ABC inventory classification. Applying a Pareto analysis to the aggregated dataset reveals a high degree of concentration within Sprinter's catalog. Specifically, 23.2 % of the dictionaries generate 79.7 % of the total units sold (Figure 2).

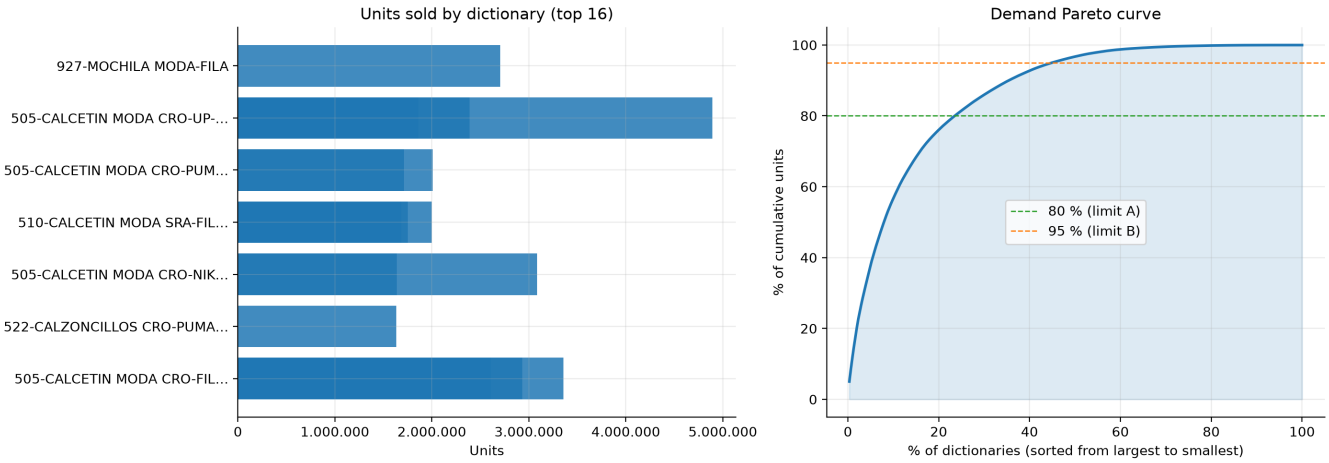

Figure 2: Cumulative distribution of sales volume by dictionary (ABC segmentation).

Following standard practice, dictionaries are classified into class A (the top group accounting for about 80 % of volume), class B and class C (the long tail). This imbalance implies that a forecasting model must perform well on class A to deliver business value: errors in these high-volume categories have a large effect on revenue and supply chain stability, whereas errors in class C carry a lower financial risk. The ABC classification is used throughout the rest of the work to break down results.

Sporting goods demand is heavily dictated by seasonal shifts and commercial campaigns. A monthly seasonality index was calculated across the historical data to isolate these patterns. For each calendar month k, the index is defined as

$$I_k \;=\; 100 \times \frac{\bar{y}_k}{\bar{y}}, \tag{2}$$

where $\bar{y}_k$ is the average total sales in calendar month k and $\bar{y}$ is the average over all months, so that an index of 100 corresponds to an average month.

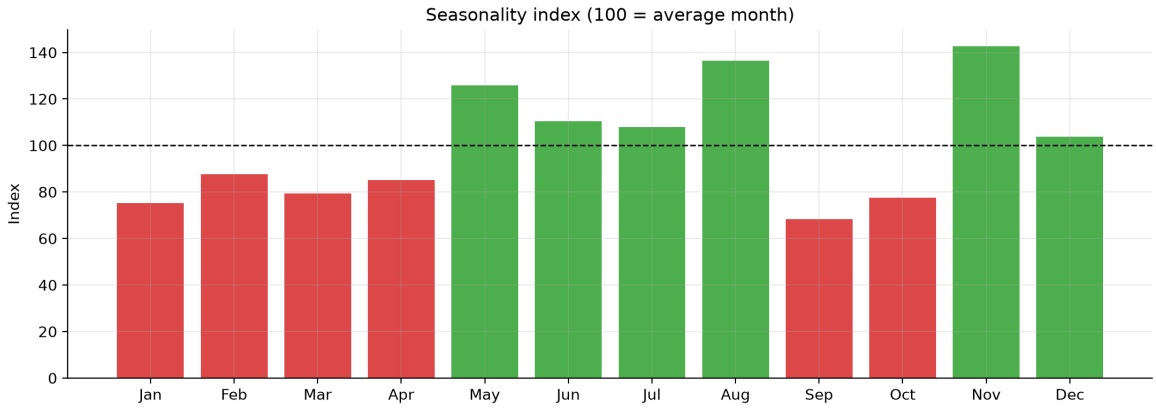

Figure 3: Monthly seasonality index showing relative demand fluctuations across the calendar year.

The analysis (Figure 3) highlights severe demand peaks in November (index 142.6), driven

by Black Friday and early holiday shopping, and August (index 136.4), corresponding to the summer clearance and Back to School campaigns. Conversely, September represents the deepest trough with an index of 68.3. Based on these structural peaks, the high season months for feature engineering purposes are formally defined as May, June, July, August, November, and December.

To select the appropriate forecasting algorithms, the time series were classified according to the Syntetos–Boylan framework (Syntetos et al. 2005), which categorizes demand based on intermittent intervals and variation in size. For each dictionary d with n_d observed months and n_d^+ months with positive sales, the two statistics are

$$\mathrm{ADI}_d = \frac{n_d}{n_d^+}, \qquad CV_d^2 = \left(\frac{\sigma(y_{dt} \mid y_{dt} > 0)}{\mu(y_{dt} \mid y_{dt} > 0)} \right)^2, \tag{3}$$

and each series is assigned to one of four classes according to the thresholds shown in Table 3. Evaluated at the aggregated dictionary level, the distribution of demand patterns is as follows: 43.8 % erratic, 37.6 % smooth, 17.0 % lumpy and 1.6 % intermittent.

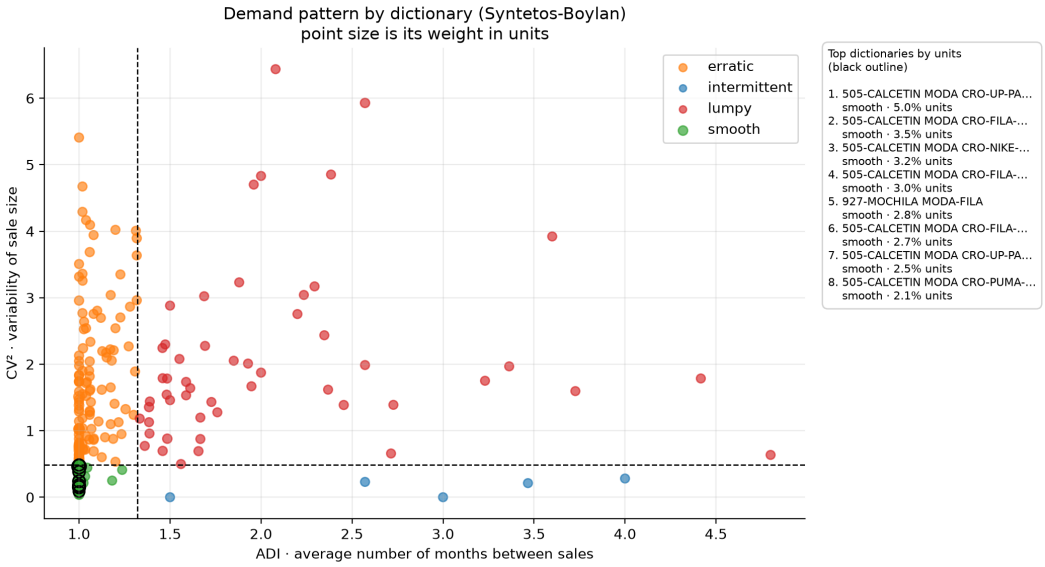

Figure 4: Syntetos–Boylan demand classification for the aggregated dictionary series.

Table 3: Syntetos–Boylan classification of the dictionary series.

Class	Definition	Share of series
Smooth	ADI < 1.32, $CV^2 < 0.49$	37.6 %
Erratic	ADI < 1.32, $CV^2 \geq 0.49$	43.8 %
Lumpy	ADI ≥ 1.32, $CV^2 \geq 0.49$	17.0 %
Intermittent	ADI ≥ 1.32, $CV^2 < 0.49$	1.6 %

At dictionary level (Figure 4 and Table 3), the distribution is favourable for standard re-

gression approaches: 81.4 % of the series are smooth or erratic, and only 1.6 % are strictly intermittent. Thanks to the change of grain, zero-sales periods account for only 10.24 % of the panel, so a GBDT model can be trained without being dominated by zeros. The high share of erratic series, however, indicates that the size of demand is volatile even when demand occurs regularly, which supports the use of a loss function that allows variance to grow with the mean.

The exploratory analysis also quantified the operational extremes of inventory management: stockouts and dead stock. For a dictionary d in month t, the stockout rate is defined as the share of its active SKUs with zero stock in a month in which the dictionary registers demand, and the dead stock rate as the share of SKU instances with positive stock and zero sales. Inventory coverage is measured as stock divided by average monthly sales, expressed in months.

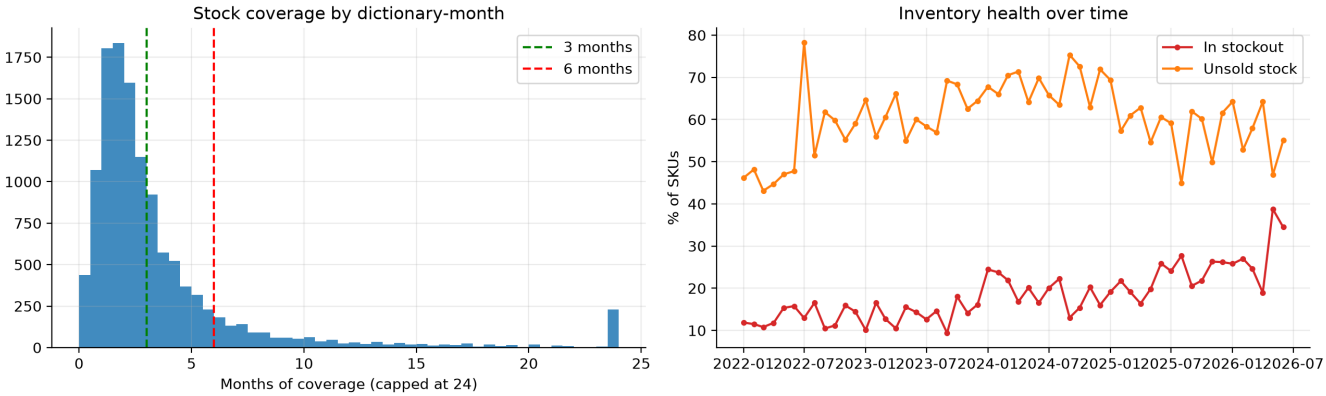

Figure 5: Distribution of stockouts and dead stock instances across active typologies.

In any given month where a typology has registered demand, an average of 18.6 % of its SKUs are out of stock, which confirms that right-censoring is widespread and suppresses the true demand signal (Figure 5). At the other end, 60.5 % of inventory instances correspond to stock that sells nothing in the month, tying up capital, and the median inventory coverage is 2.3 months, which indicates a generally adequate buffer with clear room for optimisation. The simultaneous presence of stockouts and dead stock suggests that the problem is not only the *amount* of inventory but also its *allocation*: stock is often available in the wrong products or places.

Stock and price conditions also change over time. Figure 6 shows the monthly median, across dictionaries, of four variables used later as features: stock coverage at the start of the month, the share of SKUs out of stock in the previous month, the catalogue price (PVP) and the actual discount applied.

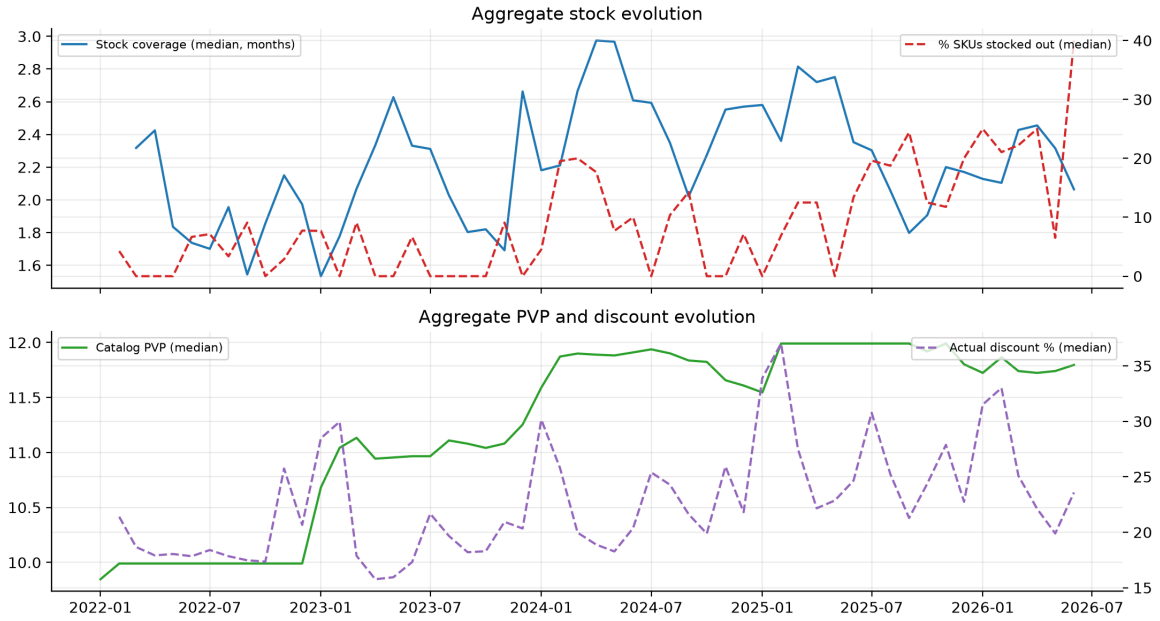

Figure 6: Aggregate evolution of stock and price conditions (median across dictionaries). Top: stock coverage in months and share of SKUs out of stock in the previous month. Bottom: catalogue price (PVP, anonymised) and actual discount.

Median stock coverage moves between roughly 1.5 and 3 months without a clear trend, but the median share of SKUs out of stock is higher in the second half of the sample than in 2022–2023, which means that censoring becomes more relevant precisely in the most recent data used for validation and testing. The median catalogue price increases in steps at the beginning of 2023 and 2024 and remains stable afterwards, while the actual discount oscillates between about 15 % and 35 %, with peaks that coincide with the winter and summer sales periods. Two consequences follow for modelling. First, stock and discount variables must enter the model lagged, since both react to demand within the month. Second, because discounts are highest when collections are being cleared, a high discount is not necessarily a signal of higher demand; it can also mark the end of a product's life cycle, a point illustrated with individual cases in Section 6.

The insights derived from this exploratory analysis directly inform several critical modelling decisions implemented in the subsequent sections:

- **Evaluation metric:** The presence of zero-sales months (10.24 % of the panel) makes MAPE undefined for part of the sample, so the volume-weighted WMAPE is used as the main metric.
- **Validation strategy:** The extreme seasonality (e.g., the November peak) makes random k-fold cross-validation inappropriate. A strict temporal block validation (walk-forward) is required to prevent data leakage from the future.
- **Model selection:** Because strictly intermittent series account for only 1.6 % of the catalogue, specialised intermittent-demand methods such as Croston's are not required.

- **Global architecture:** The shared seasonal peaks and ABC concentration justify treating the product dictionary as a categorical variable within a single global model, rather than fitting 306 isolated local models.
- **Baselines:** The strong annual seasonality dictates that the naive baselines must include a seasonal component (same month, previous year) rather than relying solely on recent moving averages.
- **Loss function:** Non-negative, right-skewed and erratic demand supports a Tweedie loss rather than MSE.
- **Censoring:** The high stockout rate requires either an explicit correction of the target or stockout features that allow the model to account for censoring.
- **Reporting:** Given the strong concentration of demand, results should be broken down by ABC class and not only reported in aggregate.

5 Methodology and Modelling Framework

5.1 Features and Validation Design

Let y_{dt} be the (anonymised) units sold of dictionary d in month t, and let $\mathcal{I}_{t-1}$ be the information available at the end of month $t-1$. The main forecasting task is to produce a one-step-ahead forecast

$$\hat{y}_{dt} \;=\; f(\mathbf{x}_{dt})\,, \qquad \mathbf{x}_{dt} \in \mathcal{I}_{t-1}, \tag{4}$$

where $\mathbf{x}_{dt}$ is a vector of features built only from information available before month t starts, and f is a single function shared by all dictionaries. In the global model, the dictionary identifier is itself one of the components of $\mathbf{x}_{dt}$, which allows f to learn series-specific levels while sharing the rest of its structure.

To capture the non-linear dynamics of retail demand, the panel was transformed into a feature matrix of 53 predictive variables (3 categorical and 50 numeric), grouped as shown in Table 4.

Table 4: Groups of features used in the global model.

Group	Description
Identifier	Dictionary identifier, treated as a categorical feature.
Calendar	Month, year and high-season indicator (May–August, November, December).
Autoregressive	Lags of sales at 1, 2, 3, 6 and 12 months.
Rolling statistics	Rolling mean and standard deviation of sales over 3, 6 and 12 months.
Price	Current price, discount and margin.
Stock	Lagged stock in customer-facing channels and in the central warehouse.
Assortment	Lagged number of active SKUs per dictionary.
Censoring	Share of SKUs out of stock in the previous month (`pct_stockout_prev_month`).
Market context	Aggregate indicators of the market in which the dictionary is sold.

Calendar features give the tree-based model a direct reference to annual seasonality. Autoregressive features and rolling statistics capture recent momentum and volatility. Business features aim to reproduce the context a human planner uses: prices and discounts, inventory split between stores and warehouse, and assortment size. Assortment size is especially relevant, because the demand of a dictionary often grows with the number of references offered to the customer. It is important to note that trees cannot extrapolate beyond the range of values seen in training; for this reason, level information is provided through lags and rolling means rather than through a time trend.

A strict non-anticipation rule was applied so that the model only uses information that would actually be available at forecast time. The rule is that no variable describing month t can include data that are only known once month t has ended. All endogenous operational variables, such as closing stock, realised active SKUs or monthly stockout rate, were therefore replaced by their value in the previous month (`_lag1` versions). Rolling statistics are computed on windows that end at $t-1$. Prices are the only variables that may refer to month t, since planned prices are known in advance.

Only dictionaries with at least 24 months of history were modelled, since without a full year of history there is no twelve-month lag and no reliable way to validate the forecast. This criterion keeps 274 of the 306 dictionaries; the 32 excluded ones represent only 0.4 % of historical units. The dataset was then split chronologically, as summarised in Table 5. The test set contains the last 6 months of observed data (January to June 2026), with 1,428 out-of-sample rows used only for final evaluation. The validation set contains the preceding 6 months (1,520 rows) and is used to choose the number of trees by early stopping. The

training set contains all earlier data (10,530 rows). Once the number of trees is fixed, the final model is re-trained on training and validation data together (12,050 rows) and evaluated once on the test set.

Table 5: Chronological partition of the dictionary panel.

Set	Period	Use
Training	January 2022 – June 2025	Model fitting (10,530 rows)
Validation	July 2025 – December 2025	Early stopping (1,520 rows)
Test	January 2026 – June 2026	Final evaluation (1,428 rows)

Once hyperparameters and the number of trees are fixed, the test set is never used to make modelling decisions. This guarantees that the reported test error is a fair estimate of out-of-sample performance.

The main metric is the Weighted Mean Absolute Percentage Error (WMAPE):

$$\text{WMAPE} \;=\; \frac{\sum_d \sum_t |y_{dt} - \hat{y}_{dt}|}{\sum_d \sum_t y_{dt}}. \tag{5}$$

By weighting absolute errors by actual volume, WMAPE avoids the division-by-zero problem of MAPE and prevents low-volume series from dominating the aggregate evaluation. It can be read as the share of total volume that is "misallocated" by the forecast.

The relative improvement of a model M over a benchmark B is reported as

$$\Delta_{M,B} \;=\; \frac{\text{WMAPE}_B - \text{WMAPE}_M}{\text{WMAPE}_B}. \tag{6}$$

For the inventory analysis, the per-dictionary root mean squared error is used:

$$\text{RMSE}_d \;=\; \sqrt{\frac{1}{T_d} \sum_t (y_{dt} - \hat{y}_{dt})^2}, \tag{7}$$

where T_d is the number of test months for dictionary d. Unlike WMAPE, RMSE is expressed in units and penalises large errors more heavily, which is the relevant property for sizing safety stock.

5.2 Models

Three naive benchmarks were implemented, representing simple operational heuristics:

- **Seasonal naive** ($\hat{y}_{dt} = y_{d,t-12}$): repeats the sales of the same month of the previous year.
- **Moving average** ($\hat{y}_{dt} = \frac{1}{3}\sum_{j=1}^{3} y_{d,t-j}$): the average of the last three months.
- **Always zero** ($\hat{y}_{dt} = 0$): used only as a diagnostic, since its WMAPE is by construction 100 %.

The seasonal naive benchmark is particularly relevant because it is close to how many planners build a first estimate: "what we sold last year in this month".

To compare against traditional methods, univariate AutoETS and AutoARIMA models were fitted with the `statsforecast` library. Both were estimated as local models, one per series, for the same 274 dictionaries with at least 24 months of history. AutoETS selects among exponential smoothing specifications (error, trend and seasonal components) by the corrected Akaike information criterion (AICc), and AutoARIMA selects the orders of a seasonal ARIMA model in a similar way. Since model selection happens within the fitting window, both models were fitted directly on training and validation data and produce a 6-month forecast from the end of December 2025. For 110 of the 274 series with at least one zero-sales month, AutoETS was restricted to an additive specification (AZA), because multiplicative components are not defined for non-positive values; the remaining 164 series went through the full search. AutoARIMA needs no such restriction, since its seasonal structure is linear.

The core of the pipeline is a global GBDT model implemented with LightGBM. Instead of fitting 306 separate models, a single model is trained on the whole panel. The model minimises the Tweedie deviance. For a power parameter $1 < p < 2$, the loss for an observation with actual value y and predicted mean μ is

$$\ell_p(y, \mu) \;=\; -\frac{y\,\mu^{1-p}}{1-p} \;+\; \frac{\mu^{2-p}}{2-p}, \tag{8}$$

which corresponds to a compound Poisson–Gamma distribution with $\mathrm{Var}(Y) = \phi\,\mu^p$. Values of p close to 1 behave like a Poisson model, while values close to 2 behave like a Gamma model. The model predicts on the log scale, which guarantees non-negative forecasts.

The dictionary identifier is handled as a native categorical variable. Hyperparameters (number of leaves, learning rate, minimum observations per leaf, feature and row subsampling, and the Tweedie power p) were tuned on the validation set. The number of trees was selected by early stopping on the validation set, which stopped at 347 trees. As a check against overfitting the validation period, the same procedure was repeated on training and validation data, holding out the last three months, which suggested 501 trees; the more conservative value of 347 was kept for the final re-training.

A point forecast estimates the conditional mean of demand, which is not sufficient for inventory decisions under uncertainty. To size safety stock directly, additional LightGBM models were trained with the quantile (pinball) loss

$$\rho_\tau(u) \;=\; u\,(\tau - \mathbf{1}\{u < 0\})\,, \qquad u = y - \hat{q}_\tau, \tag{9}$$

for $\tau \in \{0.50,\, 0.80,\, 0.90,\, 0.95\}$. The minimiser of the expected pinball loss is the conditional τ-quantile of demand. The forecast $\hat{q}_{0.95}$, for example, is the level of stock that would cover demand in 95 % of months, and the difference $\hat{q}_{0.95} - \hat{q}_{0.50}$ gives a direct, distribution-free estimate of the buffer required for that service level.

5.3 Forecast Horizons, Censoring and Robustness Design

The global model produces one-step-ahead forecasts, which corresponds to a rolling monthly planning process in which the latest observed sales are incorporated each month. Classical models, in contrast, are estimated once and produce a h-step forecast from a single origin, $h = 1, \ldots, 6$. Comparing the two directly would favour the machine learning model, which sees more recent information.

To align horizons, the global model was also evaluated recursively. Starting from the last observed month before the test period, the forecast for month $T + 1$ is computed and then used in place of the actual value when building the lag and rolling features for month $T + 2$, and so on:

$$\hat{y}_{d,T+h} \;=\; f(\tilde{\mathbf{x}}_{d,T+h})\,, \qquad \tilde{y}_{d,T+j} = \begin{cases} y_{d,T+j} & j \le 0, \\ \hat{y}_{d,T+j} & j > 0, \end{cases} \tag{10}$$

where $\tilde{\mathbf{x}}_{d,T+h}$ is built from $\tilde{y}$. In this setting the model faces the same information constraints as AutoETS and AutoARIMA, and errors accumulate along the horizon.

As a robustness check, the model was also trained on a reconstructed demand series. The idea of the fill-rate heuristic is that, when a share of the SKUs in a dictionary is out of stock, recorded sales underestimate demand, and the missing part can be approximated from a comparable period without stockouts. For each dictionary-month with a stockout rate above a threshold, the unconstrained demand is estimated by scaling observed sales with the fill rate of a seasonal analogue (the same month of the previous year), and the correction is capped to avoid extreme values.

A single test window may be favourable or unfavourable by chance. To assess stability, a walk-forward backtest was carried out over four consecutive 6-month windows. In each window the model is re-trained with all data available before the window start, using the same feature set and configuration, and evaluated on the following 6 months. The last window coincides with the main test period.

To test whether the improvement over the seasonal naive benchmark is systematic across dictionaries, the per-dictionary absolute error of both models was compared with two non-parametric paired tests. The sign test counts the number of dictionaries in which the global model has a lower error and tests whether this proportion differs from 0.5. The Wilcoxon signed-rank test also takes into account the size of the differences. Both tests use a scaled error for each dictionary, so that series of different size are comparable, and are robust to non-normal errors. The same tests were applied against AutoETS and AutoARIMA. The classical Diebold–Mariano test (Diebold and Mariano 1995) was also computed on the monthly aggregate loss differential, but with only six test months it has very little power, and it is reported only for completeness.

Finally, two diagnostics were run on the residuals of the global model. The Ljung–Box test checks whether residuals still contain autocorrelation that the model has not captured,

both on the monthly aggregate residual and separately for each dictionary. Permutation importance measures, on the test period, how much WMAPE increases when the values of a feature are randomly shuffled; unlike the gain-based importance reported by LightGBM, it is measured out of sample and on the same metric used to evaluate the model.

6 Results and Robustness Checks

6.1 Forecast Accuracy

Table 6 summarises the out-of-sample performance of all models over the test window (January–June 2026). The global Gradient Boosting model achieved a WMAPE of 17.97 %. The seasonal naive benchmark scored 33.53 %, AutoETS 36.87 %, AutoARIMA 38.96 %, and the 3-month moving average was the weakest at 41.92 %. This corresponds to a relative improvement of 46.4 % over the best naive benchmark.

Table 6: Test-set accuracy (January–June 2026, 1,428 rows). MAE and RMSE in anonymised units. Δ is the relative WMAPE improvement against the seasonal naive benchmark (Equation (6)).

Model	Type	MAE	RMSE	WMAPE (%)	Δ
LightGBM (full features)	Global, one-step	1,183	3,205	17.97	46.4 %
LightGBM (calendar + lags only)	Global, one-step	—	—	21.90	34.7 %
Seasonal naive (*lag_12*)	Heuristic	2,207	4,767	33.53	—
LightGBM (recursive, 6 months)	Global, multi-step	2,374	5,204	36.10	−7.7 %
AutoETS	Local, multi-step	2,427	4,902	36.87	−10.0 %
AutoARIMA	Local, multi-step	2,565	5,426	38.96	−16.2 %
Moving average (3 months)	Heuristic	2,760	7,126	41.92	−25.0 %

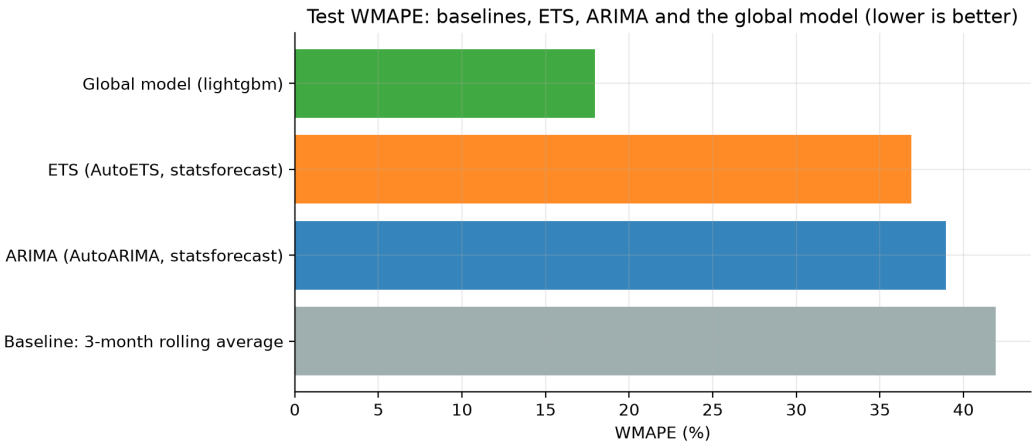

Figure 7: Test WMAPE of the naive benchmarks, AutoETS, AutoARIMA and the global LightGBM model (one-step).

Figure 7 compares the global model with the two classical local models and the moving-average benchmark; the seasonal naive benchmark and the always-zero diagnostic are reported in Table 6 but left out of the chart, the first because it is discussed separately below

and the second because its 100 % WMAPE is true by construction and only compresses the scale of the rest. Two points stand out. First, the moving average is the weakest of the four, as expected for a method that ignores seasonality in a business where November sells more than twice as much as September, and AutoETS and AutoARIMA improve on it only marginally. Second, and more telling, Table 6 shows that neither classical model beats the seasonal naive benchmark (33.53 % against 36.87 % and 38.96 %): repeating the same month of the previous year is more accurate than fitting an exponential smoothing or an ARIMA model to each series. This is consistent with the EDA. Annual seasonality is the dominant pattern, and with only a few years of history per series, ETS and ARIMA have little data to estimate seasonal components reliably. In addition, the structural break discussed below, caused by a market that only appears in 2024, is read by local models as a trend, which distorts their extrapolation.

The aggregate figures hide how the classical models behave on the series they were designed for. Figure 8 shows the six highest-volume dictionaries of the test window, with actual demand, the AutoETS and AutoARIMA forecasts and the 80 % prediction interval of the ETS model; the vertical line marks the start of the test period.

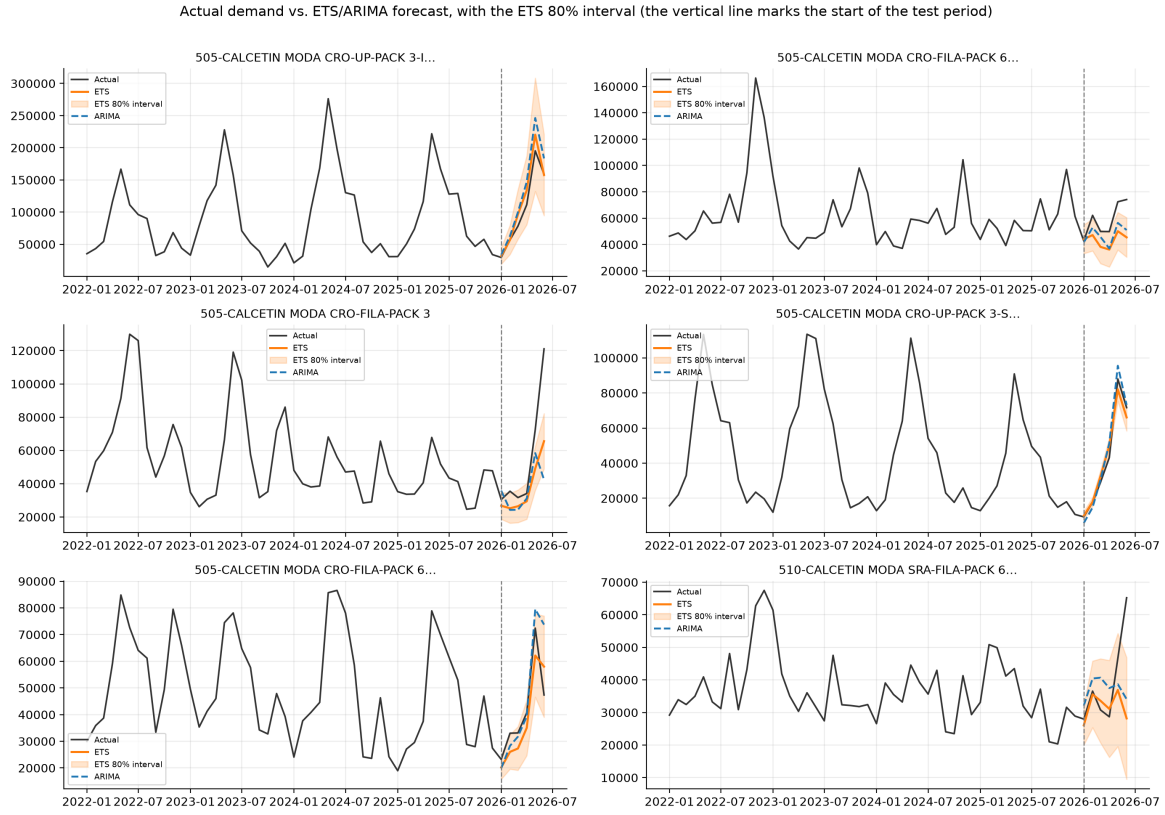

Figure 8: Actual demand against the AutoETS and AutoARIMA forecasts for the six highest-volume dictionaries, with the 80 % prediction interval of the ETS model. The vertical line marks the start of the test window.

On these six series the local models are competitive. Their test WMAPE ranges from 8.8 % to 31.6 % for AutoETS and from 9.8 % to 36.8 % for AutoARIMA, far below the portfolio-

wide 36.87 % and 38.96 % of Table 6. The weak aggregate result of the classical models is therefore not a failure on the large, smooth typologies they are built for, but on the tail of the catalogue, which is precisely what the ABC breakdown of Table 7 shows: AutoETS moves from 29.51 % in class A to 82.62 % in class C.

What the panels do expose is a systematic under-reaction to the June peak. In `505-CALCETIN MODA CRO-FILA-PACK 3` demand reaches 121,047 units in June 2026 while AutoETS forecasts 65,552 and AutoARIMA 42,369; in `510-CALCETIN MODA SRA-FILA-PACK 6-INVISIBLE` the actual figure is 65,232 units against 28,178 and 34,072 respectively. Both models reproduce the shape of the annual cycle, but they estimate its amplitude from each series' own history alone, with no access to the price, stock or assortment conditions of that particular month. The 80 % interval is wide, and in several panels the June observation still falls above its upper bound, which motivates the directly fitted quantile models described in Section 5 rather than relying on the parametric interval of a local model.

Table 7 breaks down the test WMAPE by ABC class.

Table 7: Test-set WMAPE (%) by ABC class and model.

Class	LightGBM	Seas. naive	AutoETS	AutoARIMA
A	16.93	27.31	29.51	31.55
B	20.47	44.92	53.10	53.71
C	23.79	78.13	82.62	89.12
Total	17.97	33.53	36.87	38.96

Error increases from class A to class C for all models, as expected, since high-volume series are less noisy in relative terms. The difference between models, however, grows sharply along the tail: in class A the global model reduces WMAPE by about 10 points with respect to the seasonal naive benchmark, while in class C the reduction exceeds 50 points. Local models and the seasonal naive benchmark copy the noise of small series into the forecast, whereas the global model can borrow seasonal structure from larger dictionaries.

The headline figure of 17.97 % corresponds to a one-step-ahead forecast, which simulates a rolling monthly planning process in which actual sales are updated each month. Classical models, in contrast, produce a multi-step forecast from a single origin. When the global model is evaluated recursively over the same 6-month horizon, without access to actual values after the forecast origin, its WMAPE rises to 36.1 %. Figure 9 shows how the error evolves with the horizon.

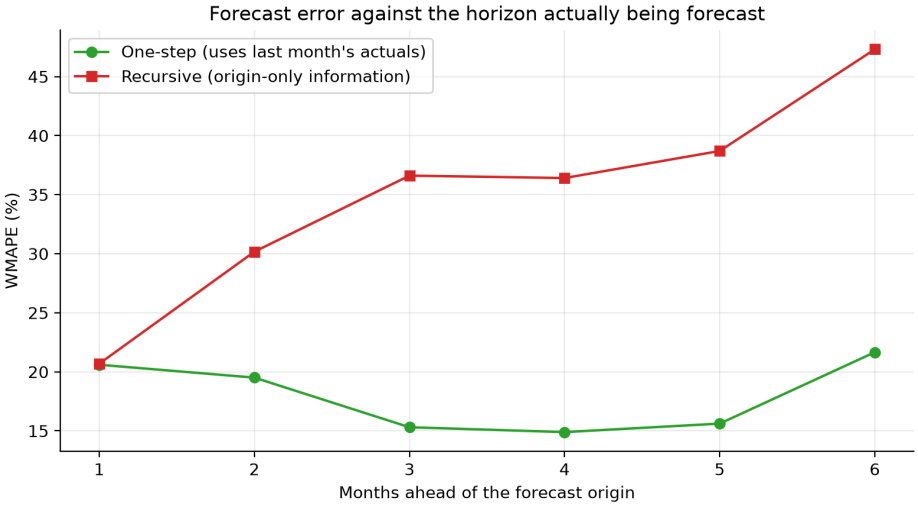

Figure 9: Test WMAPE of the global model by months ahead of the forecast origin: one-step forecast (using last month's actuals) and recursive forecast (origin-only information).

At one month ahead both versions coincide (about 20.7 %), as they should. From there the recursive error grows steadily, reaching 30.2 % at two months, around 36–39 % at three to five months and 47.3 % at six months, while the one-step error stays between 15 % and 22 %. By ABC class, the recursive WMAPE is 29.2 % for class A, 53.7 % for class B and 72.5 % for class C.

For a like-for-like comparison with AutoETS and AutoARIMA, the 36.1 % recursive figure is therefore the appropriate benchmark. In this setting the global model still outperforms AutoETS (36.87 %) and AutoARIMA (38.96 %), although by a much smaller margin, and it no longer beats the seasonal naive benchmark (33.53 %). This result is informative in itself: much of the advantage of the global model comes from incorporating the most recent month of information, and the value of the pipeline in practice depends on being re-run each month. Recursive errors also accumulate because lag features at later horizons are built from forecasts rather than observations. A direct multi-horizon strategy, with a separate model for each horizon h, could reduce this accumulation and is left for future work.

An ablation study was carried out to isolate the value of Sprinter's business data. A restricted LightGBM model, trained only on calendar and autoregressive features, scored 21.90 %. The difference with the full model (17.97 %) is 3.9 percentage points of WMAPE, which measures the marginal predictive value of adding price, stock and assortment information.

The ablation also allows the total improvement over the seasonal naive benchmark to be decomposed. Of the 15.6 percentage points separating the seasonal naive benchmark (33.53 %) from the full model, 11.6 points are obtained by the restricted model, that is, by a global model that only uses the history of sales and the calendar. The remaining 3.9 points come from business features. The largest share of the gain is therefore methodological (the global architecture and its use of recent lags and cross-series learning), while business data provide an additional and non-negligible improvement. This distinction is useful for the company:

part of the benefit could be obtained with sales data alone, while the rest requires integrating information from pricing, stock and assortment systems.

6.2 Performance under Stock and Price Anomalies

The figures reported so far describe an average month. They say nothing about the months a planner actually worries about: those in which the shelf was empty, or in which the typology was being cleared at a discount deep enough that price, rather than demand, explains most of what was sold. This subsection isolates those months and asks a narrower question: does the model hold up where the seasonal naive benchmark breaks down?

Two flags are defined at dictionary × month level. Let c_{dt} denote the stock coverage at the start of the month — the channel stock available divided by the average sales of the three preceding months, so that it is read directly in months of typical sales — let x_{dt} be that channel stock itself, and let δ_{dt} be the discount actually charged, obtained from the realised average price against the full catalogue price. A *stockout episode* and an *excessive markdown episode* are then defined as

$$R_{dt} \;=\; \mathbf{1}\Big[\, c_{dt} < 0.15\,\bar{c}_d^{\,\text{train}} \Big] \cdot \mathbf{1}\Big[\, x_{dt} < 0.50\,\bar{x}_d^{\,\text{train}} \Big], \qquad \bar{c}_d^{\,\text{train}} \;=\; \frac{1}{|T_d^{\text{train}}|} \sum_{t \in T_d^{\text{train}}} c_{dt}, \tag{11}$$

$$D_{dt} \;=\; \mathbf{1}[\,\delta_{dt} \;>\; 0.40\,], \qquad \delta_{dt} \;=\; 1 - \frac{\text{realised average price}_{dt}}{\text{full catalogue price}_{dt}}, \tag{12}$$

where T_d^{train} are the training months of dictionary d and $\bar{x}_d^{\,\text{train}}$ is defined analogously.

Four properties of this construction matter. The stockout threshold is relative to each typology's own history, because a level of cover that signals a rupture for a backpack is ordinary for a three-pack of socks, and it is estimated on the training window only, so the criterion never sees the evaluation period. The second condition of Equation (11) is not decorative: coverage also collapses when sales surge — a launch, a seasonal campaign — which is the opposite of a rupture, and requiring the stock itself to have fallen below half of its usual level makes the flag read "there is little stock for what is being sold" rather than "a lot was sold". The cut-offs (15 % and 40 %) are business criteria agreed with Demand Planning, not quantiles fitted to the data. And neither flag is a model feature: the model only sees their lagged counterparts, in accordance with the non-anticipation rule of Section 5. The flags are diagnostic labels used to read the results, and they are computed with current-month information precisely because they describe the month whose sales were distorted.

Measuring the rupture in months of cover rather than in units per store is a deliberate choice. A ratio of stock to store count is self-stabilising: when a typology is withdrawn, the number of stores carrying it falls together with its stock, so the quotient barely moves and the episode goes undetected. Coverage does not have that problem, it is invariant to the anonymisation factor because numerator and denominator carry it alike, and it is read in a unit a planner already uses.

Table 8 reports how common these episodes are. Stockout episodes are infrequent in rows and almost negligible in units: the 81 affected test months carry 0.20 % of test demand, which is itself a consequence of the phenomenon, since an empty shelf sells nothing. Markdown episodes are a different matter. A quarter of the panel sits above a 40 % discount, and those months account for 11.2 % of the units sold in the test window. This is not an anomaly in the statistical sense: clearing stock at more than 40 % off is established commercial practice at Sprinter. What the flag marks is not an error but a regime in which the price, and not the underlying demand, drives a large part of the observed figure.

Table 8: Incidence of the two episode types. Panel restricted to the 274 modelled dictionaries.

Metric	Stockout	Excessive markdown
Affected dictionary × month rows	730	3,066
as % of the panel	6.0 %	25.3 %
Dictionaries with at least one episode	148	217
Affected months inside the test window	81	341
as % of test demand (units)	0.20 %	11.2 %

Of the 274 dictionaries, 132 experience both types of episode at some point, so the two categories are not disjoint and the rows of Table 9 should not be added together. That table partitions the test window by the state of each month and reports WMAPE for the global model and for the seasonal naive benchmark within each partition.

Table 9: Test WMAPE (%) by state of the month. Same rows and same horizon as Table 6; only the partition changes.

State of the month	Months	LightGBM	Seas. naive	Advantage (pp)
Whole test window	1,428	17.97	33.53	15.56
Normal months	1,063	16.26	28.09	11.82
Stockout months	81	74.46	1,075.46	1,001.00
Excessive markdown months	341	31.15	74.98	43.83

Two readings follow, and they are not equally strong. The result on markdown months is the one with material weight: those months carry more than a tenth of the units in the test window, and the model's error there (31.15 %) is roughly two and a half times lower than the benchmark's (74.98 %), against an advantage of 11.82 points in ordinary months. The mechanism is transparent. The seasonal naive benchmark replays the same month of the previous year, promotion included, whereas the global model has the previous month's realised discount and relative price position among its features and can separate a typology whose volume comes from a clearance from one whose volume is structural.

The result on stockout months is visually the most striking and analytically the most fragile.

A benchmark WMAPE above 1,000 % is not a meaningful quantity: WMAPE divides by realised demand, and in a month in which the shelf was empty that denominator collapses towards zero, so any absolute error inflates without bound. The figure should be read as a statement of direction, not of magnitude, and the defensible evidence is the per-month absolute error shown in Figure 10, not the ratio. This is also why the stockout comparison, despite its size, cannot carry the inventory conclusions of this work: it concerns 0.20 % of test demand.

It should equally be stated that the model itself degrades inside both types of episode (74.46 % and 31.15 %, against 16.26 % in ordinary months). This is expected rather than disappointing. A stockout month does not measure demand, it measures availability (Nahmias 1994), so no model trained on realised sales can recover what customers would have bought; that is the problem the reconstituted panel of the next subsection attacks directly. The claim supported by Table 9 is therefore comparative, not absolute: the model degrades far less than the seasonal forecast does, and it does so because it observes the conditions under which the month took place.

Figures 10 and 11 illustrate each mechanism on a single typology. The two cases were selected under an explicit rule: the episode must fall inside the test window, last at least two months, and the model must beat the benchmark in the majority of the individual months of the episode, not merely on aggregate WMAPE. The last condition is a safeguard against a case whose apparent advantage comes from a single month in which the benchmark happened to fail badly.

In the stockout case, coverage falls from 1.49 months in January 2026 to 0.19 in March, crossing the typology's own threshold of 0.39 months, while channel stock drops from 31,029 units to 4,266, well below the 29,854 that the second condition requires. Realised demand collapses from 23,274 units in January to 549 in June. The model follows the collapse (19,510 $\rightarrow$ 283 units forecast), whereas the seasonal naive benchmark keeps forecasting between 13,000 and 23,000 units throughout, because a year earlier those were ordinary months. Within the three months of the episode the model wins every month, with a WMAPE of 137.3 % against 1,802.7 %. Two details belong in the reading of this case. June is not flagged, even though stock keeps falling, because demand had fallen further and coverage recovered to 1.19 months: the criterion measures scarcity relative to what is being sold, not stock in absolute terms. And the same typology is under a 50–62 % discount across the whole test window, so the two regimes overlap; what identifies stock as the operative factor is that the discount is roughly constant over the six months while demand only collapses once the coverage threshold is crossed.

In the markdown case, the typology crosses the 40 % discount threshold in January (82.4 %), May (46.0 %) and June (68.2 %). In June the discount coincides with a demand peak of 12,375 units, more than double the level of the same month a year earlier: the benchmark forecasts 5,280 units and the model 7,361. Across the three months of the episode the model

wins all three, with a WMAPE of 30.3 % against 67.7 %.

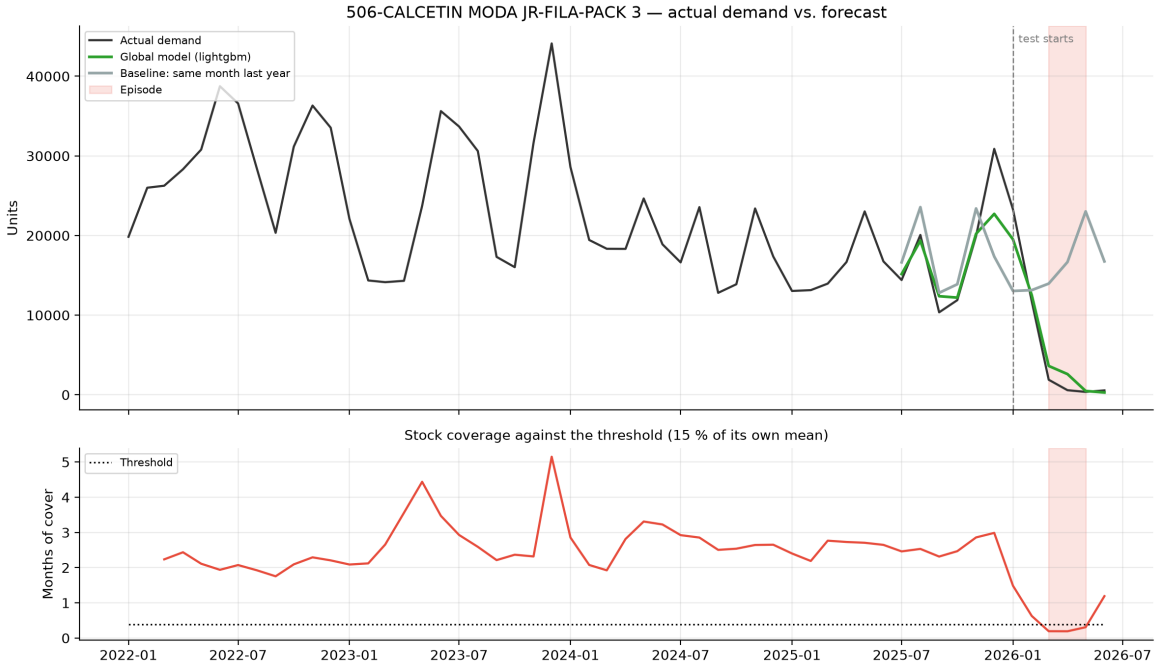

Figure 10: Stockout case. Top: actual demand against the global model and the seasonal naive benchmark; the shaded band marks the months flagged by Equation (11) and the dashed line the start of the test window. Bottom: stock coverage, in months of typical sales, against the typology's own threshold.

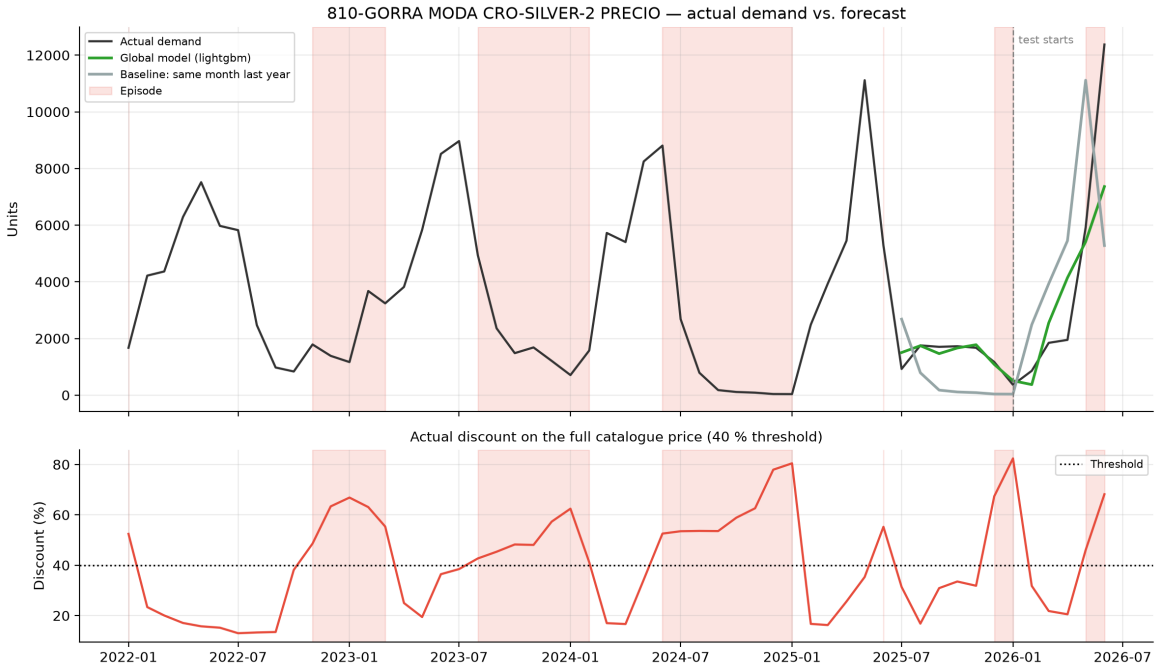

Figure 11: Excessive markdown case. Same layout; the bottom panel shows the discount actually charged against the 40 % threshold of Equation (12).

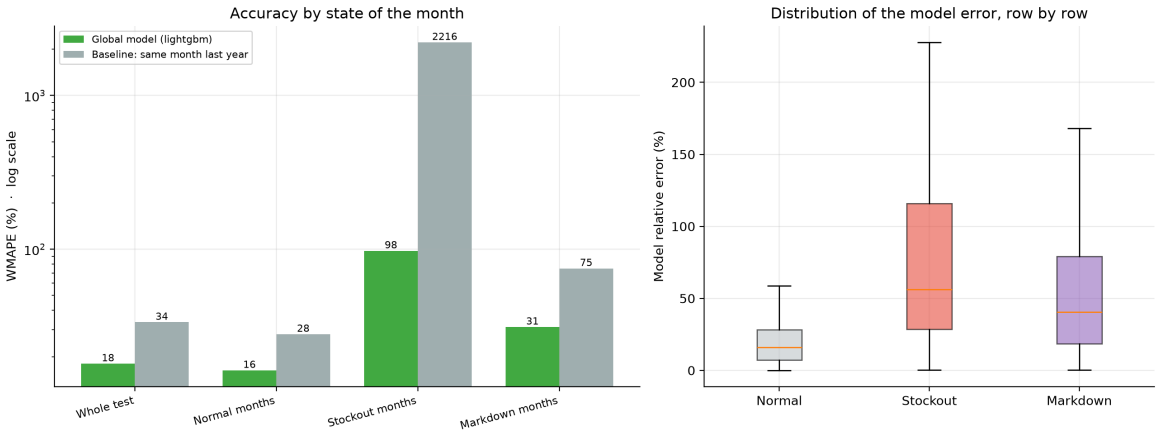

Figure 12: Left: test WMAPE by state of the month, on a logarithmic scale, for the global model and the seasonal naive benchmark. Right: distribution of the model's relative error per row in each state. The widening of the box under both types of episode is the cost the model itself pays; the distance to the benchmark in the left panel is what the comparison establishes.

Stated conservatively, the advantage of the global model over the seasonal naive benchmark is not uniform across the test window, and it is largest precisely in the months in which the observed figure is distorted by an operational condition, namely an empty shelf or a clearance price. The model does not need those months to be removed from its training data in order to handle them: it has the conditions themselves among its features and reacts to them, whereas a calendar-based forecast has no channel through which to learn that this year is not last year. The complementary exercise, correcting the training target in stockout months rather than merely observing the effect, is reported below.

6.3 Robustness and Statistical Validity

Statistical tests at dictionary level confirm that the improvement is not due to chance. Over the 234 dictionaries with forecasts from all models, the global model has a lower scaled error than the seasonal naive benchmark in 85.5 % of them, than AutoARIMA in 85.0 % and than AutoETS in 82.5 %. In all three comparisons, both the sign test and the Wilcoxon signed-rank test give p-values below 0.001. The improvement is therefore not driven by a small number of large series but is spread across most of the catalogue. The Diebold–Mariano test on the six monthly aggregate losses does not reject equal accuracy, but with six observations this result is not informative, and the conclusion rests on the dictionary-level tests.

Table 10 reports the results of the walk-forward backtest over four consecutive 6-month windows.

Table 10: Walk-forward WMAPE of the global model over four consecutive 6-month windows. Window 4 is the main test period.

Window	**WMAPE (%)**
Window 1 (July–December 2024)	24.41
Window 2 (January–June 2025)	19.29
Window 3 (July–December 2025)	19.57
Window 4 (January–June 2026, main test)	17.97
Mean	20.31
Standard deviation	2.82

The error is higher in the first window, which is expected because the model was trained on a shorter history, and decreases as more data become available. The mean WMAPE of 20.3 % is somewhat higher than the headline test figure, which suggests that the most recent window was slightly favourable; the mean should be considered the more conservative estimate of expected performance. In all windows, however, the error remains well below that of the naive and classical benchmarks on the main test period.

Two further robustness checks were carried out, summarised in Table 11. First, a structural break was identified in the data: one sales market has no records before January 2024. Since the panel sums over markets, its appearance raises the level of the affected series at a fixed date. Average monthly units before 2024 were 1,894,000, compared with 1,710,217 in the period in which all markets are present, a level shift of −9.7 %. When the model was re-trained only on the homogeneous window from January 2024 onwards (6,324 training rows instead of 12,050), it scored 20.1 %.

The loss of accuracy with respect to the full-history model indicates that older data, despite the break, still contain useful information (in particular for the twelve-month lag and seasonal patterns), and that the tree-based model is able to handle the level shift without manual truncation.

Second, training the model on reconstructed demand using the fill-rate heuristic yielded a test WMAPE of 17.6 %. The small improvement indicates that the lagged stockout feature (`pct_stockout_prev_month`) already allows the base model to account for most of the effect of censoring. It should be noted that the evaluation is still carried out against recorded sales, which are themselves censored; the true benefit of the correction on unconstrained demand cannot be observed directly.

Table 11: Robustness checks on the test set.

Specification	WMAPE (%)
Base model (full history, recorded sales)	17.97
Trained only on data from 2024 onwards	20.1
Trained on reconstructed (uncensored) demand	17.6

Finally, the residuals of the global model show no remaining autocorrelation. On the monthly aggregate residual, the Ljung–Box test does not reject the null hypothesis at any lag between 1 and 6 (all p-values above 0.78), although with twelve monthly observations this test has low power. The per-dictionary test is more informative: of 238 dictionaries tested, only 7 (2.9 %) reject the null at the 5 % level, close to the 5 % expected by chance, and none survives the Benjamini–Hochberg correction for multiple testing. These results indicate that the model has captured the temporal structure available in the series.

An analysis of where the model fails complements these results. A regression of the scaled absolute error on characteristics of each dictionary-month (with standard errors clustered by dictionary) shows that the only clearly significant driver is volume: errors are larger in months with low sales (coefficient on log demand -0.158, $p < 0.001$). The share of dead stock in the previous month has a positive coefficient close to significance ($p = 0.053$), while stockouts, series age, stock coverage, recent volatility and high season are not significant. The explanatory power is low ($R^2 = 0.16$), which means that most of the remaining error is not linked to any single observable condition. By segment, the largest errors and the strongest tendency to over-forecast are found in lumpy class C dictionaries.

7 From Forecast Accuracy to Inventory

A forecast is not valuable because it is accurate, but because a more accurate forecast lets the business hold less stock for the same service level. This section performs that translation. The incumbent against which the saving is measured is the seasonal naive benchmark, not Sprinter's own forecast, which was not available for this work; Section 7.6 complements that theoretical comparison with one against the stock the company actually held, and Section 8 discusses what neither of the two establishes.

7.1 Forecast-error dispersion is a property of the series, not of the portfolio

The natural temptation is to size stock with the pooled error of the model. That number is 2,519 units of RMSE across all dictionary-months, and it is the wrong input: it mixes typologies selling a few hundred units a month with typologies selling hundreds of thousands, and it corresponds to no real series. The median per-dictionary error dispersion is 659 units, roughly a quarter of the pooled figure, and the pooled RMSE sits close to the 85th percentile

of the per-series distribution (Figure 13). Sizing every typology with it would over-provision most of the catalogue and still under-provision the largest series.

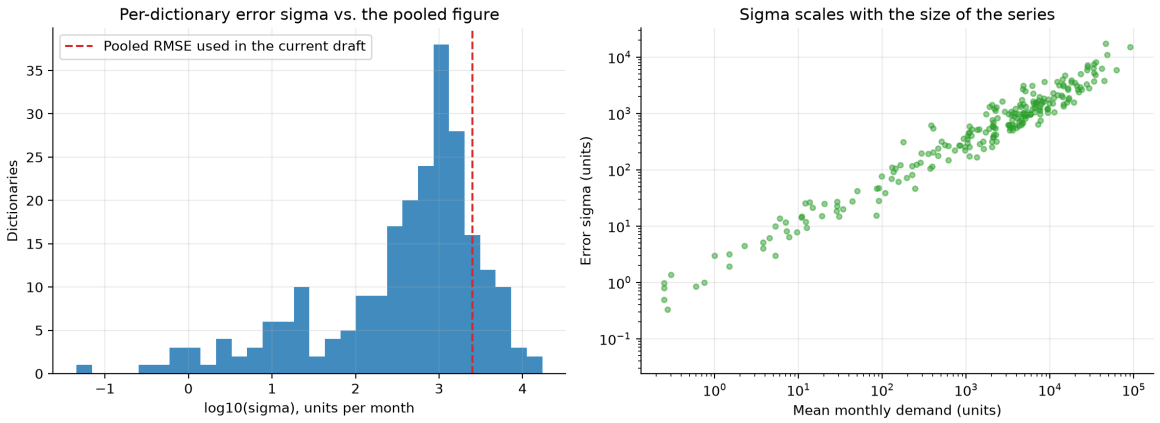

Figure 13: Left: distribution of the per-dictionary forecast-error dispersion, on a logarithmic scale, against the pooled RMSE. Right: error dispersion against mean monthly demand, which is close to proportional.

The right-hand panel explains why: error dispersion scales almost proportionally with the size of the series. Safety stock is therefore computed dictionary by dictionary and aggregated afterwards, never from a portfolio-level error.

7.2 Sizing safety stock

Under a periodic review policy with lead time L and target cycle service level α, the safety stock of a series whose one-period forecast error has standard deviation σ_1 is

$$SS \;=\; z_\alpha \, \sigma_1 \sqrt{L}, \tag{13}$$

where z_α is the corresponding quantile of the standard normal distribution (Silver et al. 2016). The total requirement of the catalogue is the sum of Equation (13) over the 240 dictionaries with an evaluable error, computed once with the error dispersion of the seasonal naive benchmark and once with that of the global model.

Two properties of this construction matter for what follows. First, the comparison between the two forecasts is a ratio of dispersions: since z_α and $\sqrt{L}$ multiply both sides, they cancel, and the *percentage* of avoidable safety stock does not depend on either policy parameter. Second, that cancellation says nothing about the distributional assumption, which affects both sides but not necessarily in the same proportion; this is examined below.

7.3 How much safety stock the better forecast makes avoidable

In the base scenario, a lead time of two months and a 95 % cycle service level, the seasonal naive forecast requires 1,502,999 units of safety stock and the global model 749,957, so 753,042 units become unnecessary. That is 50.1 % of the incumbent's requirement.

Table 12 repeats the calculation across a grid of policy assumptions. The absolute figure moves substantially with both parameters, from 414,871 units at $L = 1$ and $\alpha = 90\,\%$ to 1,151,555 at $L = 3$ and $\alpha = 98\,\%$; the percentage does not move at all. The company can therefore act on the relative result before agreeing on its own lead times and service levels, which are the two inputs this work did not have.

Table 12: Avoidable safety stock (anonymised units) under the Gaussian formulation, and the same quantity as a percentage of the incumbent's requirement.

	Avoidable units			**As % of the incumbent**		
Lead time	$\alpha = 90\,\%$	$\alpha = 95\,\%$	$\alpha = 98\,\%$	90 %	95 %	98 %
1 month	414,871	532,481	664,851	50.1	50.1	50.1
2 months	586,716	753,042	940,241	50.1	50.1	50.1
3 months	718,577	922,284	1,151,555	50.1	50.1	50.1

7.4 Results by ABC class

The saving is not uniform across the catalogue, and it is not largest where the money is. Table 13 breaks the base scenario down by ABC class.

Table 13: Safety stock in the base scenario ($L = 2$, $\alpha = 95\,\%$) by ABC class, in anonymised units.

Class	**Dictionaries**	**Demand**	**SS incumbent**	**SS model**	**Reduction**	**SS / demand**
A	71	15,585,639	988,074	528,021	46.6 %	3.4 %
B	66	3,414,783	330,431	151,414	54.2 %	4.4 %
C	103	1,083,024	184,494	70,522	61.8 %	6.5 %

The relative reduction grows along the tail, from 46.6 % in class A to 61.8 % in class C, which mirrors the accuracy breakdown of Table 7: the model's advantage over the naive forecast is largest precisely where the naive forecast is worst. In absolute terms the ranking reverses, since class A concentrates 77 % of the demand and contributes 460,053 of the 753,042 avoidable units. The last column gives the operational reading: after the change, safety stock represents 3.4 % of monthly demand in class A and 6.5 % in class C, so the long tail remains proportionally the most expensive part of the catalogue to serve.

7.5 How far the Gaussian assumption can be trusted

Equation (13) assumes normally distributed forecast errors, and the errors here are not normal. Table 14 reports the shape of the scaled error by demand pattern.

Table 14: Distribution of the scaled forecast error by demand pattern. The last column compares the empirical 95th percentile with the Gaussian one: a negative figure means the normal formula provisions more than the observed errors require.

Pattern	Obs.	Skew	Excess kurtosis	Jarque–Bera p	Empirical vs Gaussian q_{95}
Smooth	1,236	0.163	4.27	< 0.001	$-4.3\,\%$
Erratic	1,359	0.717	15.20	< 0.001	$-28.0\,\%$
Lumpy	284	-0.947	0.93	< 0.001	$-52.5\,\%$
Intermittent	22	-1.083	0.97	0.076	$-92.3\,\%$

Normality is rejected for every pattern except the intermittent one, where only 22 observations are available and the test has no power. The direction of the deviation is worth stating because it is counter-intuitive: at the single-period level the normal formula is *conservative*, not optimistic. The empirical 95th percentile of the scaled error is 28 % below the Gaussian one for erratic series and 52 % below for lumpy ones (Figure 14), because those distributions are heavy-tailed but concentrated around zero: a few large errors inflate σ far more than they move the 95th percentile.

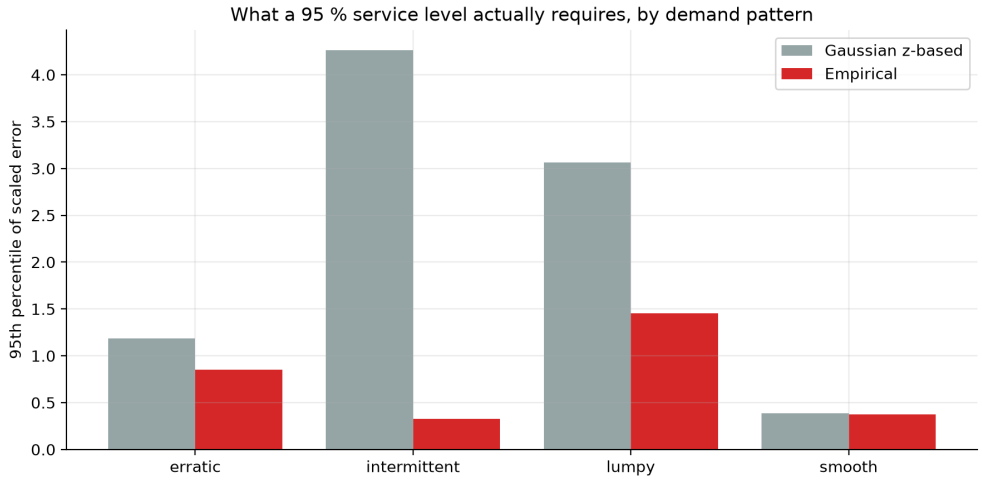

Figure 14: What a 95 % service level requires by demand pattern: the Gaussian z-based quantile of the scaled error against the empirical one.

The picture reverses once the lead time is taken into account. Comparing the empirical quantile of the *cumulative* error over a two-month lead time with the $\sqrt{L}$ extrapolation of the single-period Gaussian figure, the empirical requirement is 126 % higher for erratic series, 161 % higher for lumpy ones and 36 % higher for smooth ones. The $\sqrt{L}$ rule assumes independent errors across periods, and consecutive monthly errors of the same typology are not independent, so aggregating them understates the cumulative risk. The two deviations therefore push in opposite directions: the single-period quantile is generous and the lead-time scaling is optimistic.

The practical conclusion is that the *level* of safety stock reported above should be read as an order of magnitude and re-estimated with empirical quantiles once Sprinter fixes its

lead times, while the 50.1 % *ratio* is the robust part of the result, since both sides of the comparison are computed under the same assumption on the same series. The quantile models of Section 5 provide a third, distribution-free route to the same figure, which sizes the buffer directly from $\hat{q}_{0.95} - \hat{q}_{0.50}$ without assuming any shape for the error.

7.6 A counterfactual against the stock actually held

Everything above compares the model with a hypothetical incumbent. The panel, however, records the stock Sprinter actually held in each dictionary and month, which allows a different and more demanding question: what would have happened, over the same six test months, if replenishment had been sized with the model's forecast instead?

The counterfactual policy asks for $S_{dt}(k) = \max(0,\ \hat{y}_{dt} + k\,\sigma_d)$ units at the start of each month, where σ_d is the standard deviation of the model's forecast error estimated *on the validation window only* — using the test window to size the buffer would be sizing stock with knowledge of what happened. That quantity is compared against `stock_start_of_month`, the channel stock actually available, and both are measured against the demand actually registered. The buffer is parameterised by the multiple k rather than by a service level, because the values involved fall in the tail where the normal approximation is least credible and calling them a "99.9 % service level" would suggest a precision the distribution does not support.

Three limits of the exercise should be stated before the figures. It is a single-period comparison: both quantities are start-of-month positions covering one month of demand, and replenishment within the month is ignored on both sides. It is at dictionary level, so it says nothing about allocation across stores, which is a separate problem. And the demand it is evaluated against is registered sales, which are censored in exactly the months where stock was binding; the sensitivity below addresses that.

Over the test window Sprinter had 18,128,790 units of channel stock available, covered 96.5 % of registered demand, left 331,311 units unserved and ended the month with 9,058,152 units unsold. Table 15 reports what the model-based policy would have done with those same months.

Table 15: Counterfactual over the test window, in anonymised units. The two readings fix one magnitude and let the other move.

	Stock required	**Demand covered**	**Demand unserved**
Stock actually held	18,128,790	96.5 %	331,311
Model policy, same service level	12,462,081	96.5 %	331,311
Model policy, same stock	18,128,790	98.5 %	143,223

The two readings are the same result seen from either end. Holding service constant, the same 96.5 % coverage is reached with 5,666,709 fewer units, a reduction of 31.3 % of the stock actually carried. Holding stock constant, the same units raise coverage to 98.5 % and cut

unserved demand by 57 %. Figure 15 places both on the same axes: the curve is what the model-based policy buys as the buffer multiple grows, and the observed position sits well to the right of it at the same height.

Figure 15: Stock–service frontier over the test window. Each point of the curve is a buffer multiple of the model-based policy. The red marker is the stock actually held and the coverage it achieved.

Two properties of this result deserve to be stated rather than left for the reader to find.

The first is that the gain is concentrated in time. Figure 16 breaks the comparison down by month: four of the six months are indistinguishable between the two policies, and almost the entire difference comes from May 2026, where the stock actually held covered 91.1 % of demand against the counterfactual's 98.3 %. May is a high-season month in which demand rose sharply; the observed position was sized for an ordinary month and the model's was not. The result is therefore driven by one episode rather than by a uniform improvement, which is consistent with what Section 6.2 found about where the model's advantage lives.

The second is that the counterfactual is not better everywhere. By ABC class it gains 2.3 points in class A and 3.0 in class B but loses 3.8 in class C, where the observed stock covers 97.9 % of demand and the policy only 94.1 %. In the tail the model's relative error is larger, so a buffer proportional to it is thin, while the stock actually held in those typologies is inflated for reasons the forecast does not capture. Any deployment would need a floor for class C rather than a single rule applied to the whole catalogue.

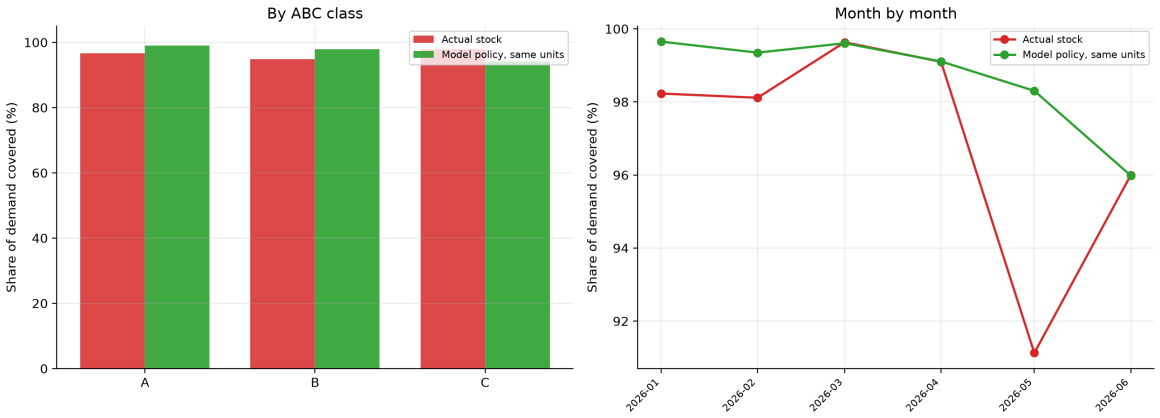

Figure 16: Share of demand covered by the stock actually held and by the model-based policy at the same total stock, broken down by ABC class and by month.

Because registered sales understate demand precisely in the months where stock was binding, the whole comparison was repeated on the reconstituted panel of Section 5, whose correction adds 1.6 % to total demand over this window. The gap does not narrow: coverage falls to 95.8 % for the observed stock and to 98.3 % for the counterfactual, so the difference widens from 2.0 to 2.5 points. The result is not an artefact of censoring.

One objection remains, and it can be quantified. If the stock held above the level the forecast justifies existed to keep every reference on display in every store, the surplus would be structural and none of it could be released. Reserving one unit for every SKU–store combination in which the typology was present over the window — the sum, across detail records, of the number of stores each reference reached — gives a presentation floor of 2,038,371 units, 11.2 % of the stock actually held; the observed position amounts to 8.9 units per SKU and store. Since the policy of Table 15 asks for 12,462,081 units, six times that floor, the surplus is not explained by presentation minimums.

What this exercise does not establish is that the model beats Sprinter's planning process. The comparison is against the stock that process produced, not against the forecast behind it, and the distance between the two is the buying and allocation policy, which is not observable in these data. The defensible statement is narrower and still useful: over the test window, the same units allocated according to the model's forecast would have covered two points more of demand, and the coverage actually achieved would have been reachable with roughly a third less stock.

7.7 From units to a currency figure

Every figure in this section is expressed in anonymised units, because the anonymisation factor applied by the company to sales volumes is preserved through the whole pipeline. Converting the saving into money is a multiplication, not a further modelling step: at a unit holding cost h, covering warehousing, capital and obsolescence, the avoidable safety stock of the base scenario is worth $753{,}042 \times h$ in the same anonymised units, and applying the

factor to Sprinter's own figures yields the currency amount.

Three inputs are therefore open, and all three belong to the company rather than to the model: the operational lead time per dictionary or per class, the target service level per class, and the unit holding cost. The grid of Table 12 is the deliverable while they are missing, because it shows how the answer moves with each assumption, which is more useful than a single number resting on parameters that were guessed.

8 Discussion

8.1 What the results support

Of the three research questions that can be answered affirmatively, the change of grain (RQ1) is the least contestable. Aggregating to dictionary level takes the share of zero-sales months from roughly a third of the records to 10.2 % and the availability of the twelve-month lag from 42.0 % to 74.0 % of rows. Without an annual anchor on three quarters of the panel, seasonality — the dominant pattern in this business — cannot be estimated for most of the catalogue, so the aggregation is a precondition rather than a convenience, and it happens to be the level at which Merchandising buys.

The advantage of the global model (RQ2) survives every check applied to it. It is not concentrated in a few large series, since the model has a lower scaled error than the seasonal naive benchmark in 85.5 % of the 234 dictionaries evaluated, with sign and Wilcoxon tests rejecting equal accuracy at $p < 0.001$; it is not an artefact of a favourable window, since four walk-forward windows give a mean WMAPE of 20.31 % with a standard deviation of 2.82 points; and it does not leave structure unexploited, since only 2.9 % of dictionaries show residual autocorrelation and none survives a correction for multiple testing. Two breakdowns matter more than the headline: the gain is largest where the naive forecast is weakest, from 78.13 % to 23.79 % in class C against 27.31 % to 16.93 % in class A, and largest again in the months distorted by an operational condition, where a markdown month costs the model 31.15 % against the benchmark's 74.98 %.

On the contribution of the business data (RQ3) the answer is positive but bounded, and the bound is the useful part. Of the 15.6 percentage points separating the seasonal naive benchmark from the full model, 11.6 are obtained by a model that sees only the calendar and the series' own history and 3.9 come from price, stock, assortment and market context. Most of the improvement is architectural, and only a quarter of it requires integrating systems outside the sales history — which means a large part of the benefit is available before any integration work is done.

Finally, the inventory translation holds under both of its formulations. The theoretical comparison gives a 50.1 % reduction in required safety stock, invariant to the lead time and the service level; the counterfactual against the stock actually held gives 31.3 % fewer units for the same coverage, or two additional points of coverage for the same units, and that gap

widens when the censoring of recorded sales is corrected.

8.2 What the results do not support

The comparison at an aligned horizon (RQ4) is the clearest limit of the exercise. The headline 17.97 % presumes a process re-run every month with last month's sales already known. Forced to forecast six months from a single origin, the same model scores 36.10 % and no longer beats the seasonal naive benchmark's 33.53 %. At the horizon the classical models face, the global model beats those classical models and not the naive one: the advantage over the naive benchmark is an advantage of the *process*, and every operational claim here is conditional on the pipeline being executed monthly.

Neither inventory result establishes that the model improves on Sprinter's planning. The theoretical figure is measured against a seasonal naive proxy, and the counterfactual against the stock that the current process produced, not against the forecast behind it; the distance between those two is the buying and allocation policy, which is not observable in these data. Nor is either figure a realised saving: no order was placed and no service level was measured during this work.

Two specific results should also not be over-read. The stockout comparison of Section 6.2 is directional rather than quantitative, since WMAPE divides by a demand that has collapsed, and it concerns 0.20 % of test demand; the markdown result, covering 11.2 % of test units, is the one that can bear weight. And the counterfactual is not better everywhere: it loses 3.8 points of coverage in class C, so a single rule applied to the whole catalogue would degrade the tail.

8.3 Limitations

On the data side, the unit levels are masked by a constant anonymisation factor, so every figure expressed in units is an anonymised quantity and only the scale-free conclusions — WMAPE, relative improvements, percentage reductions — can be read directly. The forecasting universe is also a deliberate subset: business exclusions removed 39.3 % of the raw unit volume, and the 24-month history requirement left out 32 of the 306 dictionaries. That second exclusion is small in volume (0.4 % of units) but not in meaning, because those are the newest typologies and the cold-start case is precisely where a planner has least to go on. The panel is not a single regime either, since one market has no records before January 2024 and its appearance shifts the level by −9.7 %. And the test window, six months from January to June 2026, contains no November, the highest month of the year with a seasonality index of 142.6: the month in which a forecast error is most expensive is not represented in the headline evaluation, and only the walk-forward backtest speaks to it.

On the methodological side, the treatment of censoring is a heuristic rather than an estimator — seasonal analogues with declared caps, not a Tobit or EM specification (Nahmias 1994) — its effect was small (17.6 % against 17.97 %), and the evaluation remains against observed

sales because reconstituting the target would make the comparison circular. The multi-horizon forecast is recursive, so errors compound and the 36.10 % figure is the cost of this implementation rather than the ceiling of the approach. Hyperparameters were selected once, on a single validation window. And the inventory translation rests on a normality assumption the data reject: the Gaussian formula is conservative at the single-period level and optimistic once the lead time is taken into account, so the ratios survive but the absolute unit levels should be re-estimated from empirical quantiles once the policy parameters are fixed.

Two limits cut across both. A regression of the scaled error on the observable conditions of each dictionary-month reaches an R^2 of 0.16, with volume as the only clearly significant driver, so most of what the model gets wrong is not associated with anything recorded in the data — an argument for looking outside them, at promotional calendars, competitor activity or weather, before looking for a better algorithm. And all of the above describes one company, one catalogue and one period: the methods are standard and transferable, the magnitudes are not.

9 Conclusions

This thesis develops an end-to-end machine learning pipeline that transforms raw, censored and sparse retail records into forecasts and inventory recommendations. The results support the main hypothesis: raising the forecasting grain to dictionary level solves the structural sparsity of SKU data and provides the continuous history needed for a global gradient boosting model to learn seasonal patterns. Trained with a Tweedie loss and enriched with business features, the LightGBM model reaches a test WMAPE of 17.97 %, a 46.4 % improvement over the best naive benchmark, and the improvement is statistically significant and stable over time.

The work also links statistical metrics with supply chain outcomes, and does so twice. Against a seasonal naive incumbent, the reduction in forecast-error dispersion corresponds to a 50.1 % reduction in the theoretical safety stock required for a 95 % cycle service level, a percentage invariant to the lead time and the service level assumed. Against the stock the company actually held over the test window, a replenishment sized with the model's forecast would have matched the observed 96.5 % coverage of demand with 31.3 % fewer units, or raised that coverage to 98.5 % with the same units. The second figure is the one that matters for the business case, because its denominator is inventory that exists rather than a formula.

Both readings come with the same caveat, and it is worth repeating in closing. The horizon-aligned comparison shows that the accuracy benefit depends on running the model in a rolling monthly process, and the absence of Sprinter's own forecast means the counterfactual compares against the stock that the current planning process produced, not against the forecast behind it. Obtaining that forecast, extending the evaluation to a window that includes November, and testing the quantile models in a shadow deployment are the natural

next steps to move the project from a historical backtest to a decision-support tool for the Demand Planning team.

References

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Operational Research Quarterly*, 23(3):289–303.

Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3):253–263.

Hyndman, R. J. and Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*. OTexts, Melbourne, 3rd edition.

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688.

Januschowski, T., Gasthaus, J., Wang, Y., Salinas, D., Flunkert, V., Bohlke-Schneider, M., and Callot, L. (2020). Criteria for classifying forecasting methods. *International Journal of Forecasting*, 36(1):167–177.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30*, pages 3146–3154.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings and conclusions. *International Journal of Forecasting*, 38(4):1346–1364.

Nahmias, S. (1994). Demand estimation in lost sales inventory systems. *Naval Research Logistics*, 41(6):739–757.

Silver, E. A., Pyke, D. F., and Thomas, D. J. (2016). *Inventory and Production Management in Supply Chains*. CRC Press, 4th edition.

Syntetos, A. A., Boylan, J. E., and Croston, J. D. (2005). On the categorization of demand patterns. *Journal of the Operational Research Society*, 56(5):495–503.