UNIVERSITÀ CATTOLICA DEL SACRO CUORE
Interfacoltà Scienze Matematiche - Science Bancarie, Finanziarie e Assicurative

Master Degree in Applied Data Science for Banking and Finance

Master Degree Thesis

Sales Forecasting Systems for Large-Scale Retail Enterprises

Supervisor:
Prof. Alessandro Sbuelz

Co-Supervisor:
Prof. Daniele Tessera

Company Tutor:
Nicola Pani

Candidate:
Edoardo Santucci

Student ID:
5214976

ACADEMIC YEAR 2023/2024

Abstract

In the rapidly evolving landscape of large-scale retail enterprises, accurate sales forecasting has emerged as a critical component for efficient operational planning and strategic decision-making. This thesis presents a comprehensive investigation into the application of machine learning (ML) forecasting models to predict daily sales across two distinct distribution channels, Cash & Carry (C&C) and Full Service Delivery (FSD), at Metro Italia S.P.A.
Employing an extensive historical dataset, the research systematically explores and evaluates four forecasting methods: Linear Regression as a baseline, Random Forest, LightGBM, and Long Short-Term Memory (LSTM) neural networks. Each model's theoretical foundations, strengths, and limitations are reviewed through rigorous literature analysis. Leveraging the Darts Python library, models were implemented with a global forecasting approach, aiming to balance predictive performance and computational efficiency within a resource-constrained business environment.
Validation strategies comprised both traditional train-test splits and advanced historical forecast backtesting methods, enabling realistic performance evaluations and robustness checks. Model performance was assessed using three key metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Relative MAE normalized by average store sales, highlighting practical implications for business operations.
Results indicate distinct forecasting performances across distribution channels. For the C&C channel, the LSTM model emerged as superior, effectively capturing intricate sales patterns and providing robust short-term forecasts. Conversely, LightGBM demonstrated exceptional predictive capabilities within the FSD channel, efficiently managing high-dimensional and sparse sales data.
Ultimately, this study not only identifies the optimal predictive methodologies tailored for different retail channels but also underscores the importance of model adaptability and validation strategy selection. The insights gained pave the way for informed integration of forecasting systems into Metro Italia's strategic framework, potentially enhancing inventory management, resource allocation, and overall operational efficiency.

Contents

1 Literature Review **3**
1.1 Linear Regression . . . 3
1.1.1 Parameter Estimation and Model Checks . . . 4
1.2 Random Forest . . . 5
1.2.1 Algorithmic Foundations . . . 5
1.2.2 Random Forest in Sales Forecasting . . . 6
1.3 LightGBM . . . 6
1.3.1 Background and Core Principles . . . 6
1.3.2 Gradient-Based One-Side Sampling (GOSS) . . . 7
1.3.3 Exclusive Feature Bundling (EFB) . . . 8
1.3.4 Applications to Retail Sales Forecasting . . . 9
1.4 Long Short-Term Memory . . . 11
1.4.1 Background on Recurrent Neural Networks (RNNs) . . . 11
1.4.2 LSTM Architecture . . . 12

2 Data Collection and Preprocessing **14**
2.1 Data Sources and Overview . . . 14
2.1.1 Temporal Coverage by Channel . . . 15
2.1.2 Key Dataset Attributes . . . 15
2.2 Data Extraction Workflow . . . 17
2.2.1 Tools and Technologies . . . 17
2.2.2 Pipeline Architecture . . . 18
2.3 Data Cleaning & Trasformation . . . 21
2.4 EDA & Feature Engineering . . . 23
2.4.1 EDA for Cash & Carry (C&C) . . . 23
2.4.2 EDA for Full Service Delivery (FSD) . . . 27
2.4.3 Feature Engineering . . . 31

3 Methodology and Implementation **33**
3.1 Overview of the Modeling Approach . . . 33
3.2 Model Selection and Rationale . . . 34
3.3 Hyperparameter Tuning . . . 36
3.3.1 Model-Specific Settings . . . 36
3.4 Data Splitting and Validation . . . 42
3.4.1 Train/Test split (Holdout Method) . . . 42
3.4.2 Historical Forecast backtesting . . . 43

3.5 Evaluation Metrics and Criteria 45
3.5.1 Mean Absolute Error (MAE) 45
3.5.2 Root Mean Squared Error (RMSE) 46
3.5.3 Relative MAE (Normalized by Average Store Sales) 47

4 Results and Discussion 49
4.1 Cash & Carry Channel Results 49
4.1.1 Comparative Analysis of Forecasting Models 49
4.1.2 Optimal Model Selection 77
4.1.3 Backtesting the Winning Model with Historical Forecast . . . 78
4.2 Full Service Delivery Channel Results 81
4.2.1 Comparative Analysis of Forecasting Models 82
4.2.2 Optimal Model Selection 108
4.2.3 Backtesting the Winning Model with Historical Forecast . . . 109
4.3 Cross-Channel Comparison and Discussion 113

Bibliography 119

Introduction

In today's competitive retail market, precise demand forecasting has become a cornerstone for sustainable growth and operational efficiency. Reliable sales predictions empower businesses to manage inventory more effectively, optimize resource allocation, and maintain a competitive edge through informed strategic decision-making. Within the large-scale retail sector, characterized by high product turnover and significant sales variability, the accuracy and reliability of forecasting models play a pivotal role in determining overall business success.

The present thesis addresses the practical and theoretical challenges associated with forecasting daily sales within Metro Italia S.P.A., specifically focusing on two crucial distribution channels: the Cash & Carry (C&C) channel and the Full Service Delivery (FSD) channel. These channels, while serving distinct customer bases and operating conditions, both demand robust predictive methodologies capable of navigating complex seasonal patterns, promotional activities, and varying consumer behaviors.

The principal objective of this research is to evaluate and implement advanced time series forecasting methods capable of generating accurate 30-day-ahead sales predictions, tailored to the specific operational contexts of Metro Italia's distribution channels. To achieve this goal, the analysis leverages a rich dataset encompassing several years of historical sales data, segmented across multiple store locations throughout Italy. A thorough data preprocessing phase ensures data integrity and suitability for modeling, incorporating critical transformations and exploratory analyses designed to highlight seasonal trends, cyclical patterns, and store-specific behaviors.

The methodology employed involves a rigorous comparative analysis of four prominent forecasting models: Linear Regression as a baseline, Random Forest, LightGBM, and Long Short-Term Memory (LSTM) neural networks. Each of these models was selected based on their proven effectiveness within various forecasting scenarios documented in contemporary literature, as well as their distinct methodological strengths in managing multiple time series and capturing complex data dynamics.

To validate model performance, a dual evaluation strategy is adopted: a classical train-test split providing initial performance benchmarks, complemented by a rolling-window backtesting approach. This comprehensive validation framework ensures the realistic assessment of each model's predictive robustness and operational viability under evolving market conditions.

The significance of the study extends beyond mere predictive accuracy, as it also emphasizes the selection of forecasting techniques that balance computational resource constraints and operational feasibility within a corporate setting. Ultimately, this research aims not only to identify the optimal forecasting solutions for Metro Italia's

diverse retail environments but also to deliver actionable insights that can effectively inform the company's strategic and operational decision-making processes.
In the following chapters, this thesis systematically unfolds through literature review, methodological presentation, in-depth model evaluations, and a thorough discussion of the obtained results, culminating in strategic recommendations tailored for practical deployment within Metro Italia's forecasting infrastructure.

To adhere to Metro Italia's confidentiality policies, all data and plots presented in this thesis will not include or show absolute value references. Any numerical values shown are scaled between 0 and 1 unless explicitly stated otherwise. This ensures the integrity of proprietary information while allowing for effective analysis and discussion of the results.

Chapter 1

Literature Review

Accurate demand forecasting is a foundational requirement in modern retail, where the complexity of sales patterns can vary dramatically across different channels and store profiles. This chapter reviews four major classes of forecasting models, Linear Regression, Random Forest, LightGBM, and Long Short-Term Memory (LSTM) networks, each representing a distinctive approach to modeling sales data. Drawing on key academic sources, the discussion first situates these methods within the broader forecasting literature, highlighting their theoretical underpinnings and known performance characteristics. It then examines empirical findings, including comparative studies and best practices relevant to multi-store retail settings. By emphasizing the strengths, limitations, and typical application scenarios of each model, the chapter establishes the theoretical basis for the subsequent empirical analysis of both the Cash & Carry and Full Service Delivery channels.

1.1 Linear Regression

Linear regression is a core methodological framework within econometrics and statistical modeling, designed to capture the linear relationship between a dependent variable (outcome) and one or more independent variables (predictors). In Simple Linear Regression (SLR), a single predictor X explains changes in the response Y. The model is:

$$Y_i \;=\; \beta_0 \;+\; \beta_1 X_i \;+\; \varepsilon_i,$$

where β_0 (intercept) and β_1 (slope) are typically estimated by Ordinary Least Squares (OLS), which minimizes the sum of squared deviations between actual and fitted $Y - values$.
When multiple predictors are involved, the framework extends to Multiple Linear Regression (MLR):

$$Y_i \;=\; \beta_0 \;+\; \beta_1 X_{i1} \;+\; \beta_2 X_{i2} \;+\; \cdots + \;\beta_k X_{ik} \;+\; \varepsilon_i.$$

where Y_i is the outcome for observation i, and X_{ij} (with $J = 1, 2, ..., k$) denote the various explanatory (independent) variables e.g., advertising budget, product price, or in-store display spending. The parameters β_j gauge how changes in each X_{ij} are expected to alter Y. Residuals ϵ_i capture unexplained randomness.

Under classical assumptions, the Ordinary Least Squares (OLS) method estimates $\beta_0, \beta_1, ..., \beta_k$ by minimizing the sum of squared errors:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \Big(Y_i \; - \; \beta_0 \; \sum_{j=1}^{k} \beta_j X_{ij} \Big)^2 .$$

According to Chen Dong [1] MLR is especially effective when multiple covariates simultaneously affect an outcome such as sales encompassing variables from "offline promotions" (e.g., in-store marketing) to "online factors" (web ads, social media presence) and "price". Having multiple predictors enables richer insights: each slope β_j isolates the partial effect of its corresponding regressor on Y, holding the other variables constant.

1.1.1 Parameter Estimation and Model Checks

After solving the above minimization (usually via standard linear algebra techniques), one obtains the vector of estimates $\hat{\boldsymbol{\beta}} = \big(\hat{\beta}_0, \; \ldots, \; \hat{\beta}_k\big)^{\mathsf{T}}$. As detailed by Hariaji et al. [2], further essential steps include:

1. **Individual t-Tests**: Evaluate whether each coefficient β_j is significantly different from zero (i.e., whether each predictor truly impacts Y);

2. **Global F-Test**: Determine if the collective set of predictors substantially explains variability in Y;

3. **Coefficient of Determination**: Assess how much of the outcome's variation is accounted for by the model. Adjusted R^2 corrects for the inflation that arises from adding more predictors.

Although MLR has considerable predictive capacity, its success rests on various classical assumptions. Linearity stipulates that each predictor relates linearly to the dependent variable, though polynomial or transformation methods can be adopted if the relationships are strongly nonlinear. Next, the absence of perfect multicollinearity ensures that predictors are not exact linear combinations of one another; otherwise, the resulting parameters might be highly unstable or even incalculable. Homoscedasticity requires that the residuals exhibit uniform variance across different predicted values, and if the variance increases at higher levels of predicted sales, more robust approaches become necessary. Finally, normality of residuals is ideally

maintained, as statistical tests for significance hinge on a normal error distribution; severe departures from normality may necessitate robust or nonparametric regression solutions. If any of these assumptions are violated, adjustments such as regularization for multicollinearity or M-estimators for non-constant variance can safeguard the reliability and interpretability of MLR-based sales forecasts.

1.2 Random Forest

Random Forest (RF) is an ensemble learning method introduced by Leo Breiman (2001) [3]. It constructs multiple decision trees, each trained on a different bootstrap sample of the dataset, and combines their outputs to produce a final prediction. By averaging or voting across these individually grown trees, RF significantly reduces model variance and improves robustness relative to a single decision tree. This balance of accuracy, stability, and relatively simple interpretability has made Random Forests a mainstay in modern data science [4].

In sales forecasting, RF can capture non-linearities and interactions among numerous explanatory variables, such as holiday promotions, store locations, and macroeconomic indicators, while providing useful out-of-bag estimates for model validation. Its capacity to handle both continuous and categorical attributes is particularly helpful in retail contexts, where features often vary widely in type and distribution [5].

1.2.1 Algorithmic Foundations

Let us denote the training data as $\{(x_i, y_i)\}_{i=1}^{N}$. Random Forest with B trees builds each tree by:

1. **Bootstrap Sampling**: draw a sample of size N from the original dataset with replacement, forming $\mathcal{D}_b$ for tree $b \in \{1, \ldots, B\}$;
2. **Random Feature Subset**: at every candidate split, choose only m_{try} features out of the total p to determine the best split. This strategy reduces correlation among trees [3];
3. **Tree Growth**: fit a classification or regression tree $T_b(\cdot)$ to $\mathcal{D}_b$ by recursively splitting on the feature and threshold that maximizes impurity reduction (e.g., Gini in classification or variance reduction in regression).

For a regression setting (common in sales forecasting), the final RF predictor is:

$$\hat{f}_{\mathrm{RF}}(x) \;=\; \frac{1}{B}\sum_{b=1}^{B} T_b(x),$$

Where each $T_b(\cdot)$ is a regression tree grown without pruning until a stopping criterion (e.g., minimum node size) is met.

1.2.2 Random Forest in Sales Forecasting

In a retail setting, the data typically spans numerous feature types, from numerical measures (like prices or historical sales) to categorical variables (such as product categories or store regions) and date-related fields (e.g., time-of-year). Random Forest can effortlessly accommodate these mixed inputs because each split in the trees can handle whichever feature type is most relevant at that particular node. Furthermore, RF naturally unravels complex interactions among variables by branching deeper whenever a particular combination of conditions, say, promotional discounts combined with holiday seasons, appears important in explaining sales behavior. Lastly, many Random Forest implementations provide feature importance metrics (via permutation or Gini-based methods) that highlight which aspects of the data drive sales the most. This can reveal, for instance, whether location-related features, certain promotional events, or holiday effects are the critical forces behind observed demand patterns.

Despite its broad suitability, Random Forest requires careful attention to practical considerations. In large-scale sales forecasting scenarios, both hyperparameter tuning and data characteristics shape the model's performance, while time-series factors demand feature engineering to capture the temporal structure.

- **Hyperparameters**: Typically, the number of trees B and the number of features sampled at each split m_{try} require tuning. Breiman [3] suggests around $\sqrt{p}$ features for classification and up to $\frac{p}{3}$ for regression, though empirical testing is often best;
- **Data Size**: Sales data can be massive. Because each tree grows on a bootstrapped sample and each split only looks at m_{try} features, RF can scale reasonably, though it may demand significant memory for extremely large data;
- **Time-Series Aspects**: While Random Forest does not inherently account for temporal autocorrelation, practitioners typically engineer features, like lagged sales, rolling means, or holiday flags, and let the ensemble capture the patterns.

1.3 LightGBM

LightGBM stands for Light Gradient Boosting Machine which serves as a gradient boosting framework that specializes in coping with large scale high-dimensional data. LightGBM has gained prominence in retail sales forecasting due to its implementation of Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) which enable effective handling of massive and limited data [6][7].

1.3.1 Background and Core Principles

LightGBM follows the conventional gradient boosting decision tree (GBDT) paradigm: in each iteration, it learns a new decision tree to fit the negative gradient of a chosen loss function. Formally, if we denote:

$$\{(x_i,\ y_i)\}_{i=1}^{N}$$

as our training data (where x_i is the feature vector and y_i the observed target), LightGBM builds an ensemble of M decision trees:

$$\widehat{F}(x) \ = \ \sum_{m=1}^{M} h_m(x),$$

where each weak learner h_m is grown by approximately solving:

$$\min_{h_m} \sum_{i=1}^{N} \Big[L\Big(y_i,\ \widehat{F}_{m-1}(x_i)\Big) \ + \ \big(\nabla L\big(y_i,\ \widehat{F}_{m-1}(x_i)\big)\big) h_m(x_i) \Big].$$

Here, $\nabla L(\,\cdot\,)$ indicates the negative gradient of the loss L evaluated at the previous ensemble prediction $\widehat{F}_{m-1}$.
Compared to standard GBDT approaches, LightGBM adopts a histogram-based splitting strategy to reduce memory usage and speed up tree learning. In practice, continuous feature values are bucketed into discrete "bins", allowing each potential split point to be found by scanning bin statistics rather than the full raw data.

1.3.2 Gradient-Based One-Side Sampling (GOSS)

A distinctive feature of LightGBM is its sampling approach, Gradient-based One-Side Sampling (GOSS), which acknowledges that data points with large residuals are typically more important for determining split points. In many standard implementations, random subsampling (known as Stochastic Gradient Boosting) can lead to suboptimal splits because the sampling does not consider how "under-fitted" a data point is.

GOSS addresses this by retaining all instances whose gradients exceed a certain threshold (or, equivalently, the top $\alpha \cdot 100\%$ largest gradients), while randomly sampling from the remaining data points with smaller gradients.
Suppose:

- A is the set of data instances with large gradients;
- B is the subsampled set of data instances with smaller gradients;
- α is the fraction of data deemed to have "large" gradient;
- β is the fraction randomly sampled from the "small" gradient set.

During split-finding (e.g., computing information gain based on the sum of gradients), GOSS applies a constant multiplier $\frac{1-\alpha}{\beta}$ to the sampled points in B to compensate for their lower sampling probability.
Formally, for a candidate split d on feature j GOSS estimates the "variance gain" as:

$$\widetilde{V}_j(d) \;=\; \frac{1}{N}\Big[\Big(\sum_{i\,\in\,A_l} g_i \;+\; \tfrac{1-\alpha}{\beta}\sum_{i\,\in\,B_l} g_i\Big)^2 \,/\, n_l(d) \;+\; \Big(\sum_{i\,\in\,A_r} g_i \;+\; \tfrac{1-\alpha}{\beta}\sum_{i\,\in\,B_r} g_i\Big)^2 \,/\, n_r(d)\Big],$$

where A_l (resp. A_r) denotes data from A going to the left (resp. right) child of the split, and similarly B_l, B_r are the subsets from B.
The factor $\frac{1-\alpha}{\beta}$ preserves an unbiased estimate of the overall gradient sum, allowing split decisions to remain close to what would be found if all points were kept.

Ke et al.[6] show that GOSS introduces only a small approximation error in the split finding process dominated by $\mathcal{O}\Big(\frac{1}{\sqrt{N}}\Big)$ while yielding substantial speed-ups because significantly fewer samples are used to construct the histograms.

1.3.3 Exclusive Feature Bundling (EFB)

Another hallmark of LightGBM is Exclusive Feature Bundling (EFB), which capitalizes on the sparsity inherent to many real world datasets.
Often, large numbers of categorical or dummy coded features mean that many features are mutually exclusive (i.e., they are never nonzero at the same time).

1. **Motivation**. In a histogram-based GBDT, the time spent scanning features is proportional to *"number of features × number of data"*. If many features are rarely active together, we can "bundle" them into one merged feature, thus reducing the dimensionality;

2. **Key Idea**. By assigning unique bin ranges to each feature within the same bundle, we can still recover which original feature was triggered. For example, if feature A has values in $[0, 10)$ and feature B in $[0, 20)$, we offset B's bins by $+10$ so that B's values occupy $[10, 30)$ in the bundle. We store only one histogram per bundle, which saves memory and speeds training;

3. **Approximate Bundling**. The exact problem of optimally bundling features to minimize overlap is NP-hard, so [6] propose a greedy strategy: sort features by frequency of nonzero values, then group them together if their overlap is negligible (i.e., conflict remains under a user-defined threshold).
   In sales forecasting, contexts often replete with one-hot item or store indicators, EFB can significantly cut down on feature scans.

1.3.4 Applications to Retail Sales Forecasting

In practice, LightGBM is often combined with time-series decomposition strategies to handle strong seasonality and trend in retail data[7].Suppose a sales series Y_t decomposes into:

$$Y_t \;=\; T(t) \;+\; S(t) \;+\; H(t) \;+\; R(t),$$

where $T(\cdot)$ denotes a (possibly nonlinear) trend component, $S(\cdot)$ captures seasonal cycles, $H(\cdot)$ accounts for holiday or event effects, and $R(\cdot)$ is a residual.
By fitting LightGBM primarily on the more "stationarized" residual or partial series, it can more accurately capture complex short-term variations, particularly those interactions between product, location, and marketing signals. Meanwhile, the higher level trend or seasonality can be modeled by simpler methods (e.g., Prophet or ARIMA). This "hybrid" approach frequently improves Weighted Root Mean Squared Scaled Error (WRMSSE), a popular metric for multi-level, hierarchical retail forecasting.

Furthermore, for certain nonnegative, long-tailed sales distributions, practitioners can specify LightGBM's Tweedie loss:

$$\ell_{\text{Tweedie}}(y, \hat{y}) \;=\; -\sum_{t=1}^{T}\Big(y_t \cdot \frac{\hat{y}_t^{\,1-p}}{1-p} \;+\; \frac{\hat{y}_t^{\,2-p}}{2-p}\Big), \quad 1 < p < 2.$$

which better handles many zero or near-zero sales observations without sacrificing the ability to model large, infrequent spikes. This is particularly relevant for SKUs whose demand is intermittent or highly volatile.

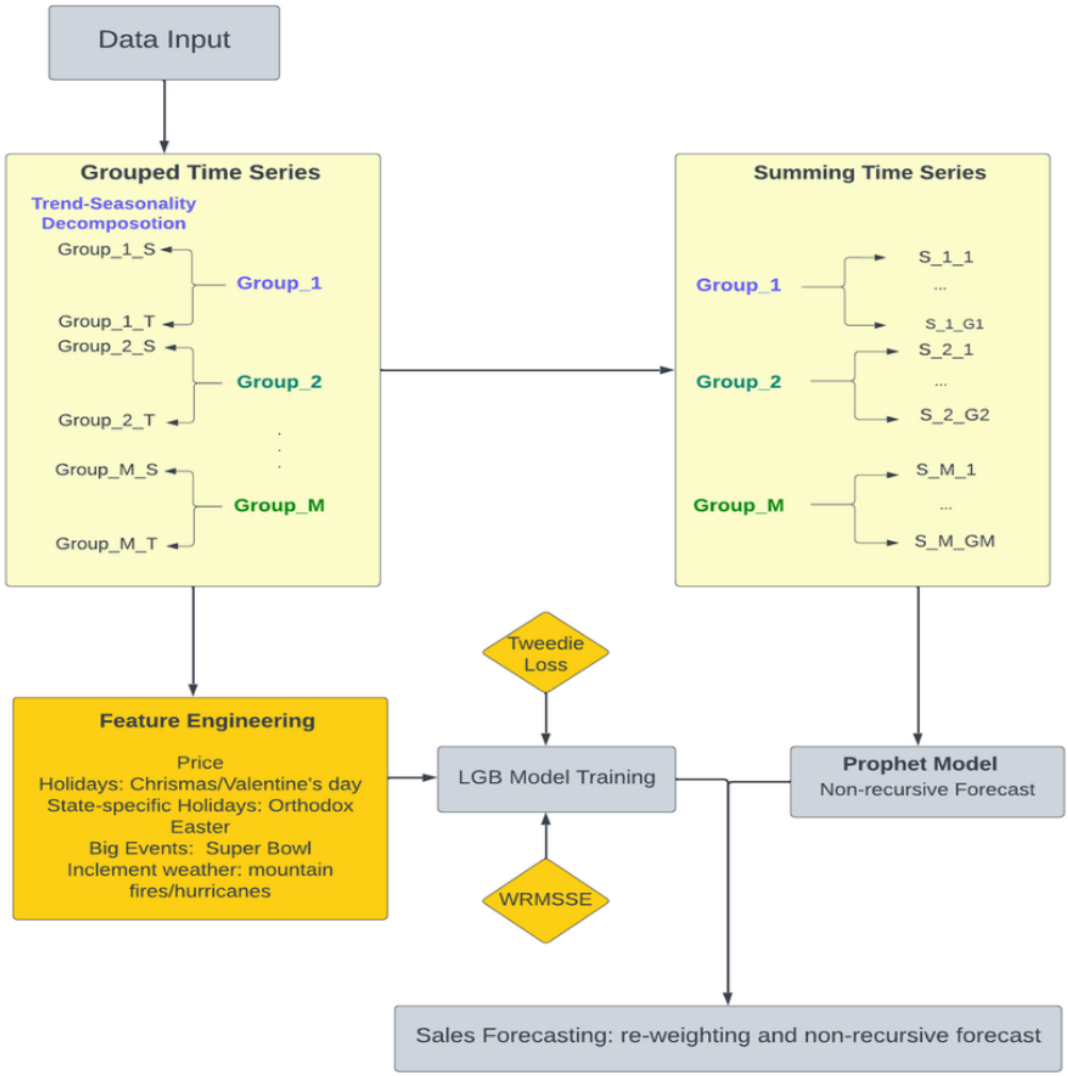


Figure 1.1: Proposed LightGBM pipeline for sales forecasting

LightGBM offers a practical solution to the speed versus accuracy challenge often encountered in large-scale forecasting efforts, making it an excellent choice for applications in sales forecasting. In particular, datasets in this domain can feature extremely high dimensionality, think thousands of product-related or promotional variables, and substantial sparsity, as many products register zero daily sales in most stores. To handle such scenarios efficiently, LightGBM's Exclusive Feature Bundling (EFB) compresses these sparse inputs into a smaller number of combined features, cutting back on data-processing overhead.
Another common hurdle is long-tailed distributions, where certain products exhibit rare but occasionally large spikes in demand. Tweedie or Poisson loss functions in LightGBM help model these sporadic demand patterns with greater fidelity than traditional mean-squared approaches. Meanwhile, complex seasonality and trends, typical in retail data, can be addressed by using LightGBM alongside decomposi-

tion techniques (for instance, splitting out the recurring seasonal effects from the more erratic residual signals). This combination enables the model to focus more effectively on immediate fluctuations while leaving underlying trends and seasonal swings to specialized methods.

Ultimately, LightGBM stands out as a swift and robust gradient boosting framework. By using it within a broader predictive pipeline, enriched by trend and seasonality decomposition, advanced feature engineering, and loss functions tailored to retail realities, it can significantly improve both the reliability and the computational feasibility of large-scale sales forecasting.

1.4 Long Short-Term Memory

1.4.1 Background on Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) function as a neural network category to manage sequential data through their architectural design featuring a feedback cycle. Traditional feedforward networks operate under the assumption of individual inputs yet RNNs continuously reuse output results of previous steps for future step calculation. This mechanism provides an internal "memory" or hidden state h_t allowing the network to capture contextual information from previous inputs [8] [9].

Conceptually, RNNs are particularly well-suited for tasks involving temporal or sequential dependencies, such as forecasting in time series data or predicting the next word in a sentence. However, training RNNs effectively can pose challenges. The standard training method, known as Backpropagation Through Time (BPTT), involves unrolling the network for multiple time steps and performing backpropagation across these unrolled layers The conceptual structure of RNNs makes them excellent candidates to solve problems with sequential patterns in data like time series prediction and text completion tasks. Training RNNs presents its own set of difficult applications. The standard training process named Backpropagation Through Time (BPTT) undergoes multiple time step unrolling followed by backpropagation execution across unrolled layers. While this approach is theoretically sound, it often leads to vanishing gradients, where the gradient becomes too small to enable learning of long-term patterns, or exploding gradients, where the gradient grows uncontrollably and causes numerical instability. Moreover, RNNs do not natively include a mechanism to filter out irrelevant or noisy data within a long sequence, which can dilute or obscure important signals.

Despite these constraints, RNNs laid the groundwork for more specialized architectures capable of retaining long-term dependencies. Long Short-Term Memory (LSTM) networks, introduced next, represent one of the most influential advancements in mitigating the limitations of standard RNNs.

1.4.2 LSTM Architecture

The researchers *Hochreiter and Schmidhuber* (1997) created Long Short-Term Memory (LSTM) networks which resolve the gradient problems that affect basic RNNs. LSTMs (figure 1.2) achieved widespread adoption in recent years but only became popular with today's deep learning platforms because they solved the major gradient problems that occurred in vanilla RNNs [10].

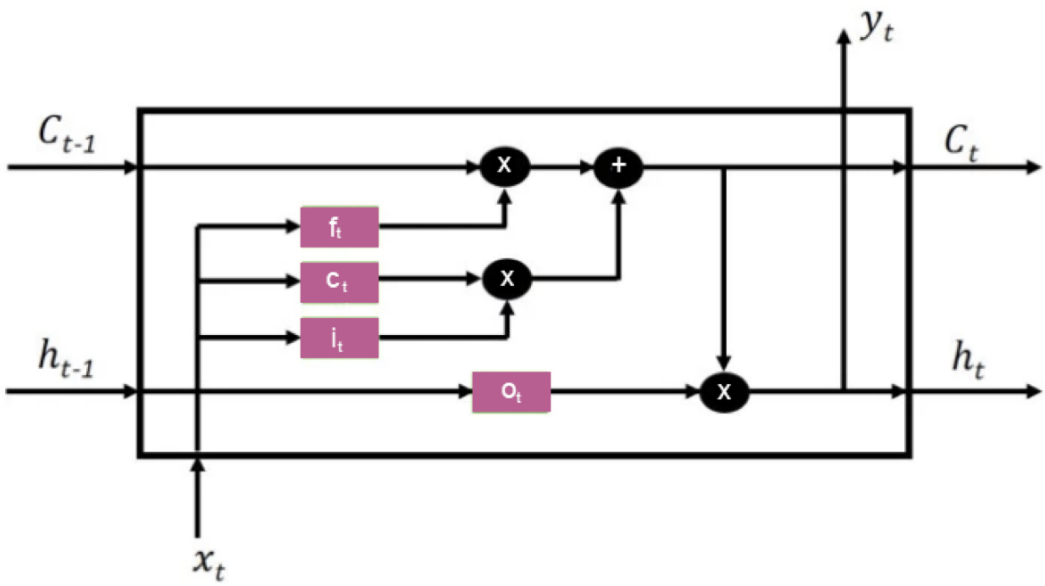


Figure 1.2: LSTM Architecture

The key innovation introduced by LSTM networks is the cell state, often denoted as C_t which runs in parallel with the hidden state h_t. At each time step t, the network takes the current input x_t, the hidden state from the previous time step h_{t-1}, and the cell state from the previous time step c_{t-1}. By combining these three elements, it computes the updated cell state c_t and the new hidden state h_t. This architectural choice allows the network to more effectively retain or discard information over extended sequences, thus addressing common issues found in standard RNNs, most notably, the vanishing or exploding gradient problem.

It then produces h_t (the new hidden state) and c_t (the updated cell state). This design is implemented through three main gates plus a candidate state:

- **Forget Gate (f_t)**, decides how much of the old cell state c_{t-1} to retain;
- **Input Gate (i_t)**, determines the extent of new information (the "candidate" $\tilde{C}_t$) that should be added to the cell state;
- **Candidate Cell Update ($\tilde{C}_t$)**, typically generated via *tanh*, providing new content in the range $[-1, 1]$;
- **Output Gate** (o_t), controls how much of the updated cell state is exposed to the hidden state h_t (often after another *tanh* compression.

By combining these gates, LSTM cells can selectively retain, forget, or expose information at each time step. The update equations commonly take the form:

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{C}_t, \quad h_t = o_t \odot \tanh(c_t).$$

LSTM networks surpass standard RNNs by establishing superior capability to both retain and make use of information from distant time periods. Important forecast indicators frequently signal themselves multiple steps ahead of final outcome changes during real-world predictions concerning sales along with energy and financial patterns. Ordinary RNN networks frequently lose vital contextual information when they cannot process distant signals in a proper manner. Training LSTMs uses its internal gating systems to determine what information should be stored or discarded. The gated mechanisms in these networks enable them to detect important market pattern fluctuations and maintain an accurate representation of sustained promotional events and gradual market trends.

The intrinsic gating mechanism of LSTMs helps keep trainings stable since it minimises the possibility of gradient explosions and the disappearance of gradients [9][10]. The network maintains equilibrium which lets it concentrate on important time-based connexions instead of being swamped by large amounts of potentially confusing data that characterise long-term prediction operations. The forget gate feature in LSTMs functions as a built-in mechanism to clean out dated information from long sequences by acting as a philtre. The skill to learn and remember long contexts makes LSTMs highly effective in applications spanning language processing to advanced sales analytics and demand forecasting while other domains which require remembering extended series of information.

Chapter 2

Data Collection and Preprocessing

Having accurate and dependable data is key to any successful forecasting effort. When it comes to predicting sales at Metro Italy S.P.A., the Data Collection and Pre -processing phase is especially important, given the wide range of stores and different channels in play, like Cash & Carry and Delivery. This chapter outlines the whole process from extracting data using **Knime**, **Google Cloud**, and **BigQuery**, to cleaning, transforming, and enhancing the raw sales records.
First, we'll take a look at the sources, structure, and key features of our dataset, showing how we capture and organize daily sales information. Then, we'll dive into the data quality challenges that come with managing multiple stores, each with issues like completeness and anomalies. We'll go over our approaches for **data cleaning and missing value imputation**, explaining why we make specific choices. We'll also focus on outlier detection and correction to make sure that unusual sales spikes or mistakes don't throw off our forecasting models.
Finally, this chapter will cover **feature engineering** techniques, emphasizing how adding variables like day-of-week indicators or special event markers can really boost our predictive performance. By the end, you'll have a solid grasp of how we turn raw data into a well-organized, reliable dataset that's ready for model training. This transformation is essential for the effectiveness of the later forecasting stages, emphasizing how critical data collection and preprocessing are in machine learning workflows.

2.1 Data Sources and Overview

This study utilises data sourced from **all Metro Italy S.P.A.** outlets across the country. By including every outlet without any initial filtering by location or sales volume, the dataset effectively captures the intrinsic diversity of sales patterns and local conditions that may otherwise be overlooked. Daily transactions from each outlet are recorded in real-time and subsequently **refreshed to a cloud environment every two hours**, thus providing a near real-time reflection of retail activities within both the Cash & Carry (C&C) and Full Service Delivery (FSD) channels.

2.1.1 Temporal Coverage by Channel

Even though we have a consistent data acquisition pipeline in place, the **temporal range** we're looking at differs from channel to channel, and there's a purposeful temporal shift for a portion of the C&C data:

- **C&C (Cash & Carry)**
  - **Original Pre-Pandemic Period**: October 1, 2018, to September 30, 2019.
  - **Two-Year Forward Shift**: To enhance predictive performance and align the timeline with subsequent observations, these earlier dates were shifted by exactly two years, effectively mapping the transactions onto the interval **October 1, 2020, to September 30, 2021**.
  - **Post-Pandemic Period**: After a gap caused by COVID-19 disruptions, data collection resumes on **October 1, 2021**, extending through **September 30, 2024** the close of the most recent fiscal year.
- **FSD (Delivery)**
  - **October 1, 2021, to September 30, 2024**: marking the continuous span of post-pandemic sales data available for the delivery channel. Because historical data for the FSD channel prior to October 1, 2021 were unavailable, the dataset for this channel begins on that date.

To summarise, the C&C dataset consists of two main segments: one originally from 2018–2019 that has been adjusted to span 2020–2021, and another covering the years 2021–2024, which captures the post-pandemic environment. This strategy is designed to preserve essential insights from the pre-pandemic era while providing a more smooth timeline for forecasting.

2.1.2 Key Dataset Attributes

No matter which channel you look at, every record in the dataset is enriched with several core attributes that are considered essential for future forecasting tasks:

- **Date**: This sets up a daily chronological order, creating the timeline upon which our sales predictions hinge.
- **Store ID and Store Name**: These details ensure each record is linked to a specific store; the unique numeric ID acts as a key, while the store name makes it easy to identify.
- **Channel (C&C or FSD)**: This distinguishes the operational context-whether it's onsite shopping or delivery-an important factor that can influence demand patterns.
- **Sales**: This is the main target variable, showing the total revenue generated by a store on any given day.

By taking this multi-channel, multi-store approach, we gain a complete view of Metro Italy's sales operations. Including every store in the dataset helps us avoid any biases that might crop up if certain locations or segments were left out-especially considering the two different C&C intervals and the ongoing FSD interval.

To give a clear visual snapshot, **figure 2.1** displays a sample from the dataset, complete with columns for the date, store ID, channel, and daily sales figures. While it only captures a small portion of the total records, this example effectively illustrates the structured format and emphasises the main fields that will be critical in our predictive modelling efforts.

Date	Channel	Store	Sales
2018-11-30	C&C	0020 - Bari	0.038684572
2019-01-15	C&C	0032 - Cagliari	0.0363419584
2019-02-11	FSD	0061 - Maxi Merano	0.03665929
2019-02-17	FSD	0033 - Modena	nan
2019-03-12	C&C	0023 - Moncalieri	0.0565945932
2019-03-19	C&C	0041 - Salerno	0.0349125486
2019-05-25	C&C	0033 - Modena	0.0356796375
2019-06-07	FSD	0032 - Cagliari	nan
2019-06-21	FSD	0030 - Parma	0.0367873292
2019-06-30	C&C	0033 - Modena	0.03994704
2019-07-15	C&C	0021 - Roma Aurelia	0.0385817563
2019-08-30	FSD	0027 - Brescia	0.0404252028
2019-09-19	FSD	0027 - Brescia	nan
2022-09-02	C&C	0017 - San Donato	0.0841082414
2022-11-05	C&C	0069 - Ventimiglia	0.0347369302
2023-01-02	C&C	0071 - Roma Salaria	0.0346478545
2023-09-16	C&C	0034 - Udine	0.0377945429
2023-09-19	C&C	0025 - Verona	0.0450559162
2023-11-09	FSD	0017 - San Donato	0.0340409807
2024-02-09	C&C	0066 - Treviso	0.0467613691

Figure 2.1: Raw data table

By bringing together the data collected from these adjusted timeframes for C&C, alongside the continuous interval for FSD, we can get a fuller and more accurate view of Metro Italy S.P.A.'s retail activities.
The next section will describe the methods and tools used to extract this raw dataset, illustrating the initial steps that ultimately provide the foundation for subsequent data cleaning, transformation, and feature engineering.

2.2 Data Extraction Workflow

Gathering sales data from Metro Italy S.P.A. is the critical first step in the journey of data processing and forecasting. This section will outline the tools utilised and provide an overview of the structure of the extraction pipeline.

2.2.1 Tools and Technologies

There are three main elements that drive the data extraction process: the **Knime Analytics Platform**,**Google Cloud**, and **BigQuery**. Each of these tools brings unique strengths to the table, and when you put them together, they form a smooth, all-in-one workflow.

- **Knime Analytics Platform**: is a key platform that brings together the entire data extraction and processing workflow. Its user-friendly, low-code interface simplifies the management of complex workflows by providing drag-and-drop nodes for everyday tasks. Also, Knime's flexibility enables data engineers and analysts to integrate Python scripts or external libraries when they need to perform more advanced operations, like custom transformations or feature engineering. The platform also offers handy scheduling and reporting features, which make it easier to keep track of and automate recurring extraction jobs.

- **Google Cloud**: utilising a cloud-based infrastructure offers excellent scalability and reliability, particularly when managing large amounts of transactional data. The adaptable computing resources of Google Cloud can be allocated as needed, efficiently managing both peak loads and more moderate query volumes while ensuring minimal downtime. Besides, its integrated services simplify the deployment of Knime workflows, providing centralised logging and monitoring tools that assist swift responses to any data pipeline issues or spikes in operational demands.

- **BigQuery**: is a powerful, serverless data warehouse, acting as the go-to place where all store transactions come together. It organises data into partitioned tables, often based on date, which enables speedy and cost-efficient querying across extensive datasets. With built-in SQL compatibility and smooth integration within the wider Google Cloud ecosystem, it allows for easy data ingestion from tools like Knime and various other sources. This mix of fast performance, scalability, and user-friendliness means that even the most detailed queries can be run with little fuss, offering analysts timely insights into sales trends.

These technologies work hand in hand to create a bi-hourly update cycle for both Cash & Carry (C&C) and Full Service Delivery (FSD). This ensures that any new transactions are promptly available for subsequent processes like data cleaning, feature engineering, and eventually, forecasting.

2.2.2 Pipeline Architecture

To gather the complete raw dataset for the 2018–2019 period and the 2021–2024 period, we created a **Knime workflow** that interacts with local files, Google Cloud, and BigQuery. This workflow pulls together information from various sources into one comprehensive table, which is then subject to additional cleaning and transformation. The **figure 2.2** below illustrates the individual nodes and their connections, providing a clear view of the entire data extraction process.

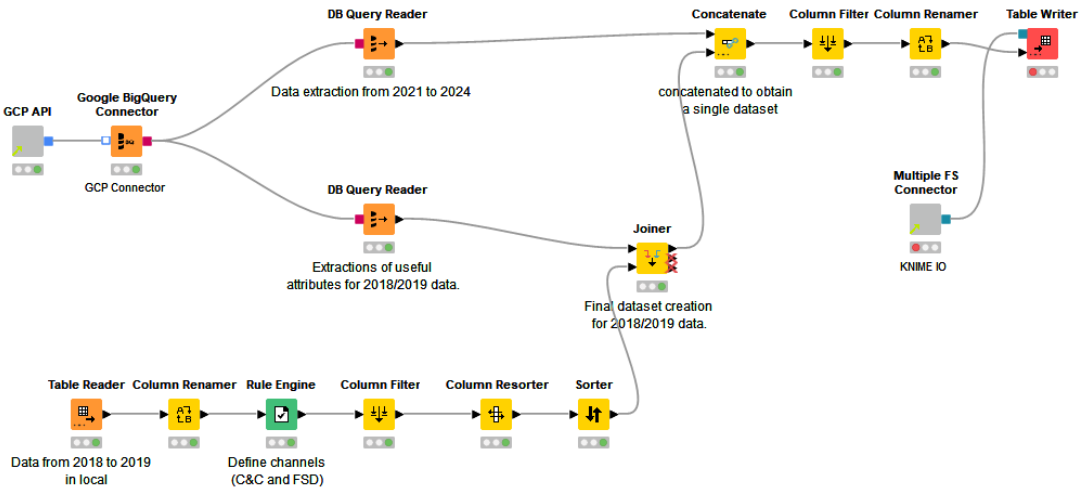


Figure 2.2: Knime workflow used to extract and consolidate sales

- **Table Reader**: we used this node to gather the historical sales data from 2018 to 2019. Table Reader node was set up to read a local file that included four main columns:
  - **Store ID** - A unique numerical identifier for each store
  - **Date** - Daily timestamps indicating the calendar day of each recorded transaction
  - **Channel** - A categorical variable with two codes (1 = C&C, 2 = FSD)
  - **Sales** - The total revenue generated by a specific store on a given day in the designated channel
- **Column Renamer**: updates the names of columns to guarantee that naming conventions are consistent across both historical and current datasets
- **Rule Engine**: once the column naming conventions were settled, we brought in a Rule Engine node to make the dataset a bit clearer. This node took a look at each record and swapped out the numeric codes in the Channel field for more descriptive labels. So, every time it found a 1, it replaced it with "**C&C**," and any 2 changed into "**FSD**." By doing this early on in the workflow, it made

it a lot easier for later analyses to tell the difference between Cash & Carry and Full Service Delivery data. This will help prevent any mix-ups and make everything a lot easier to understand

- **Column Filter & Column Resorter**: removes any unnecessary columns and reorders the remaining fields to align with the schema expected by downstream components

- **Sorter**: sorting records by date is a critical step, as keeping a consistent order can be incredibly useful for validating information or troubleshooting, particularly when you're dealing with several different sources

- **GCP API & Google BigQuery Connector**: this process connects to Google Cloud Platform and BigQuery. By using this connector, the workflow retrieves transaction data from 2021 to 2024, focusing only on the essential columns necessary for analysis.

- **DB Query Reader (Data extraction from 2021 to 2024)**: this node performs a series of **SQL** queries to gather and integrate all the critical information required for sales analysis covering the period from **2021 to 2024**. The query is structured using one or more **Common Table Expressions (CTEs)** that distinctly separate calendar related attributes like dates, channels, and store information from the aggregated **daily sales figures**. By utilising a final **JOIN** operation, the node merges the dimension-like subquery, which outlines key descriptors, with a subquery that calculates total sales for each day and store. This method allows the node to **consolidate net sales**, aligning them with the appropriate calendar and store-level details, eventually creating a clear dataset. The incorporation of parameterised date constraints in the query guarantees that only records within the specified timeframe are retrieved, which enhances efficiency and ensures that downstream workflow nodes receive precisely the data necessary for the subsequent steps of cleaning, transformation, and modelling.

- **DB Query Reader (Extractions of useful attributes for 2018/2019 data)**: in a manner similar to the previus , this node employs a dimension like structure, outlining essential descriptors such as channel type and branch groupings for each combination of store and date, which should be added, with the next node, to the table with the 2018 and 2019 values

- **Joiner**: this node brings together the locally managed dataset for the 2018–2019 period and the table generated by the DB Query Reader for the same timeframe. By using a **Left Outer Join** on the date, store, and channel columns, the node ensures a precise match between each record in the local data and any relevant rows in the dataset extracted from the database. Consequently, the output is a **single**, **cohesive table** that follows the same structure as the data retrieved for the 2021–2024 period—specifically, including standard attributes such as store details, channel identifiers, and date fields. This structural and content alignment promotes effortless downstream integration, en-

abling subsequent nodes to process historical data from 2018–2019 alongside post-pandemic data from 2021–2024 as equals

- **Concatenate**: after aligning the columns as outlined earlier, the **concatenate** node smoothly merges the output from the final **Joiner** node with the dataset retrieved for the years 2021 to 2024. Since both tables now have a matching schema, this concatenation produces a **cohesive dataset** that covers multiple time periods. This unified structure simplifies future analyses and model building efforts, as there's no longer a need to handle historical data and post-pandemic data separately

- **Column Filter & Column Renamer**: applies final adjustments to ensure the merged dataset has uniform columns, standardized names, and the correct data types.

- **Table Writer**: exports the consolidated dataset to a local location.

2.3 Data Cleaning & Trasformation

To forecast accurately, we need a dataset that not only reflects the complete range of historical sales, but also follows strict standards of reliability and consistency. In this research, we approached data cleaning and transformation as intertwined processes, making sure that any oddities we found were dealt with alongside the overall restructuring of the data for analysis. This joint effort produced a final set of tables that were ready for further modelling and evaluation.

Before diving into the details of cleaning and transforming the data, we first **split** the raw dataset into two distinct tables, based on how the sales were made. With the help of shift nodes in Knime, we created one table for **Cash & Carry** daily sales and another for **Full Service Delivery**. As we worked on these tables, we also **tidied** them up to sort out any inconsistencies in their respective time periods. For the C&C dataset, we adjusted records from the 2018–2019 period in line with our date realignment strategy, which helped us align historical data with more recent figures. In contrast, we decided to **exclude** the 2018–2019 entries from the FSD dataset entirely, as many stores had incomplete or incorrect logging for that timeframe. This led to both tables now containing a Date column that lists all the relevant days, followed by several columns that capture daily sales for each Italian store within the appropriate channel. The **figures 2.3, 2.4** below depict the final table structure, showing the raw multiple time series data for each city where Metro Italia S.P.A. has a store. Both C&C and FSD channels are shown.

Date	0010 - Cinisello	0011 - Cesano Boscone	0012 - Castellanza	0013 - Torino	...	0092 - Deposito Roma
2022-11-03	0.4721255023	0.4907268803	0.5071865681	0.5188766288	...	nan
2022-11-09	0.3833890901	0.4139855295	0.3878358295	0.4050482248	...	nan
2022-12-06	0.5711594368	0.5907349621	0.4923854986	0.6321881825	...	nan
2022-12-12	0.5066764329	0.5069062397	0.4426534761	0.4761047348	...	nan
2022-12-18	0.5661607719	0.5325902965	0.6373912279	0.4079486119	...	nan
2023-03-14	0.3813964702	0.5193904445	0.3004555082	0.4259481223	...	nan
2023-03-31	0.5367880998	0.6281797207	0.5009370103	0.6920840858	...	nan
2023-07-28	0.4690016982	0.5015784279	0.4389166901	0.5647188814	...	nan
2023-09-21	0.5007100857	0.6091234796	0.5150606594	0.6045223472	...	nan
2023-09-22	0.6153998551	0.7268237692	0.4740749147	0.6510558509	...	nan
2023-10-16	0.3425861509	0.394939614	0.2225993768	0.36032511	...	nan
2023-12-31	0.2462003532	0.2538118236	0.2402193557	0.2869881302	...	nan
2024-02-27	0.3794818278	0.3682123539	0.2505916676	0.3463128086	...	nan
2024-04-10	0.3498494867	0.4248692364	0.4338984086	0.3208174118	...	nan
2024-04-12	0.614329354	0.7035779259	0.4917869439	0.6840487659	...	nan
2024-06-25	0.389122247	0.4576254523	0.3604378056	0.3880520512	...	nan
2024-07-05	0.5256738101	0.6043138145	0.4868198998	0.6404542885	...	nan
2024-08-28	0.3848426476	0.3498934359	0.3629382237	0.3590330866	...	nan
2024-09-03	0.4481157733	0.5311939201	0.3869420708	0.4735722943	...	nan
2024-09-29	0.1754198028	0.2008016205	0.1520121362	0.1340134472	...	nan

Figure 2.3: Table structure for C&C channel

Date	0010 - Cinisello	0011 - Cesano Boscone	0012 - Castellanza	0013 - Torino	...	0092 - Deposito Roma
2023-04-08	0.139741225	nan	0.0419862457	0.047447187	...	0.1070101133
2023-06-19	0.1448755326	0.2942634575	0.0446489924	0.0155906546	...	0.0926560264
2023-06-23	0.2115611169	nan	0.04313634	0.1252383023	...	0.1410311939
2023-07-12	0.0742099217	0.1900499413	0.0505341879	0.0834762705	...	0.1111604757
2023-07-15	nan	nan	0.0481606553	0.0588927645	...	0.1033976994
2023-11-08	0.0712184941	0.1350251332	0.0550032224	0.0773444226	...	0.0655794119
2023-12-03	nan	nan	nan	nan	...	nan
2023-12-05	0.2299884206	0.185115954	0.0836744344	0.088198167	...	0.1023830256
2023-12-08	nan	nan	0.0580891268	0.1045729277	...	0.078393818
2024-01-14	nan	nan	nan	0.0031706351	...	nan
2024-02-14	0.2175201725	0.1663113752	0.0719024828	0.0773989521	...	0.0774297091
2024-02-15	0.151901909	0.1348113894	0.0446643373	0.0920785659	...	0.0729782972
2024-03-02	0.103018431	0.1332542292	0.0387931864	0.0681786486	...	0.0863713212
2024-03-20	0.1540786577	0.1830695399	0.0660734656	0.081714222	...	0.0840390068
2024-04-29	0.2180877016	0.1204434168	0.0530192967	0.0219989583	...	0.1047623951
2024-06-07	0.1361127003	nan	0.0428476461	0.0886278425	...	0.1594367193
2024-07-22	0.0717896847	0.1353656604	0.0436437652	0.0242323335	...	0.0854651164
2024-08-16	0.0837416644	nan	nan	0.0476765983	...	0.0662450534
2024-09-09	0.0718528452	0.1715184344	0.0496028247	0.0212197833	...	0.0981612302
2024-09-24	0.1405806189	0.1257992027	0.0531628634	0.0808630822	...	0.1315765267

Figure 2.4: Table structure for FSD channel

An examination of the table in **figures 2.3** reveals that the Deposito Roma column contains exclusively null values. This phenomenon stems from the fact that not all Metro Italy stores engage in both channels equally. Specifically, certain locations such as Deposito Nord and Deposito Roma operate solely within the Full Service Delivery (FSD) channel and therefore do not participate in Cash & Carry (C&C) sales. Conversely, there are also stores that generate minimal or no revenue through FSD, rendering them less relevant for analyses focused on that channel. By distinguishing among stores with differing operational profiles, the dataset provides a more nuanced view of Metro Italy's sales structure, enabling the modeling process to exclude or deprioritize locations that do not contribute meaningful data for a given channel.

Considering the unique operational characteristics of the C&C and FSD channels, it was essential to develop distinct strategies for addressing missing values. In the C&C channel, **sales are logged on a daily basis**, with no planned closures. As a result, any instances of null entries usually signal exceptional circumstances—such as unanticipated store closures—that are not aimed at being forecasted. To tackle these irregularities, the data was adjusted by substituting NaN with a **seven-day rolling average**.

On the other hand, the FSD channel takes a break on Sundays, meaning that a **null or missing value** in this scenario accurately indicates **zero** sales. Consequently, any missing entries for FSD were replaced with 0, reflecting the lack of delivery activity on Sundays. This distinction helps ensure that the data for each channel truly represents its actual operations, which in turn provides more dependable inputs for future forecasting models.

2.4 EDA & Feature Engineering

Before diving into modeling, it's important to really grasp the underlying structures and patterns of the dataset. Conducting **Exploratory Data Analysis (EDA)** gives us a solid framework for spotting potential trends, seasonality, or anomalies that could affect forecasting performance in both the C&C and FSD channels. Alongside these exploratory methods, we'll implement a bit of **feature engineering** to fine-tune the dataset by introducing variables that capture the unique characteristics of each channel more effectively. By weaving these insights into the creation of new features, we can ensure that our predictive models are armed with the most relevant information available.

2.4.1 EDA for Cash & Carry (C&C)

Figure 2.5 displays the daily sales figures for four selected Cash & Carry stores: San Donato, Brescia, Pescara, and Firenze, covering the period from October 2020 to September 2024.
While these stores may vary in their locations and customer demographics, several interesting trends and fluctuations can be identified. The sales data from each store reflects a consistent pattern, with certain days reliably showing higher sales figures than others. This regular fluctuation implies that consumer demand may be influenced by weekly cycles, such as increased activity on weekends or promotional efforts during weekdays. Across the sampled stores, which behave in a very similar manner with respect to all the stores in the dataset, we observe pronounced seasonal peaks and some constant dips. Sharp increases or decreases appear at multiple intervals, often around the end of each year, hinting at seasonal sales effects. Although the timing of these peaks is similar across locations, their magnitudes differ, reflecting variations in local clientele.

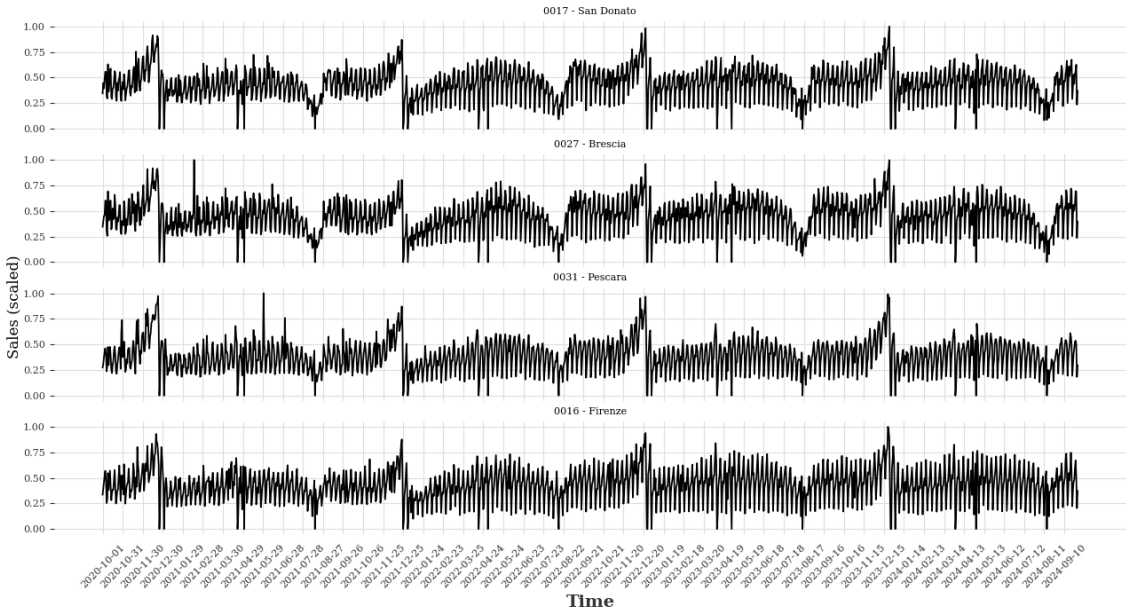


Figure 2.5: Line plots of 4 sampled stores between all stores.

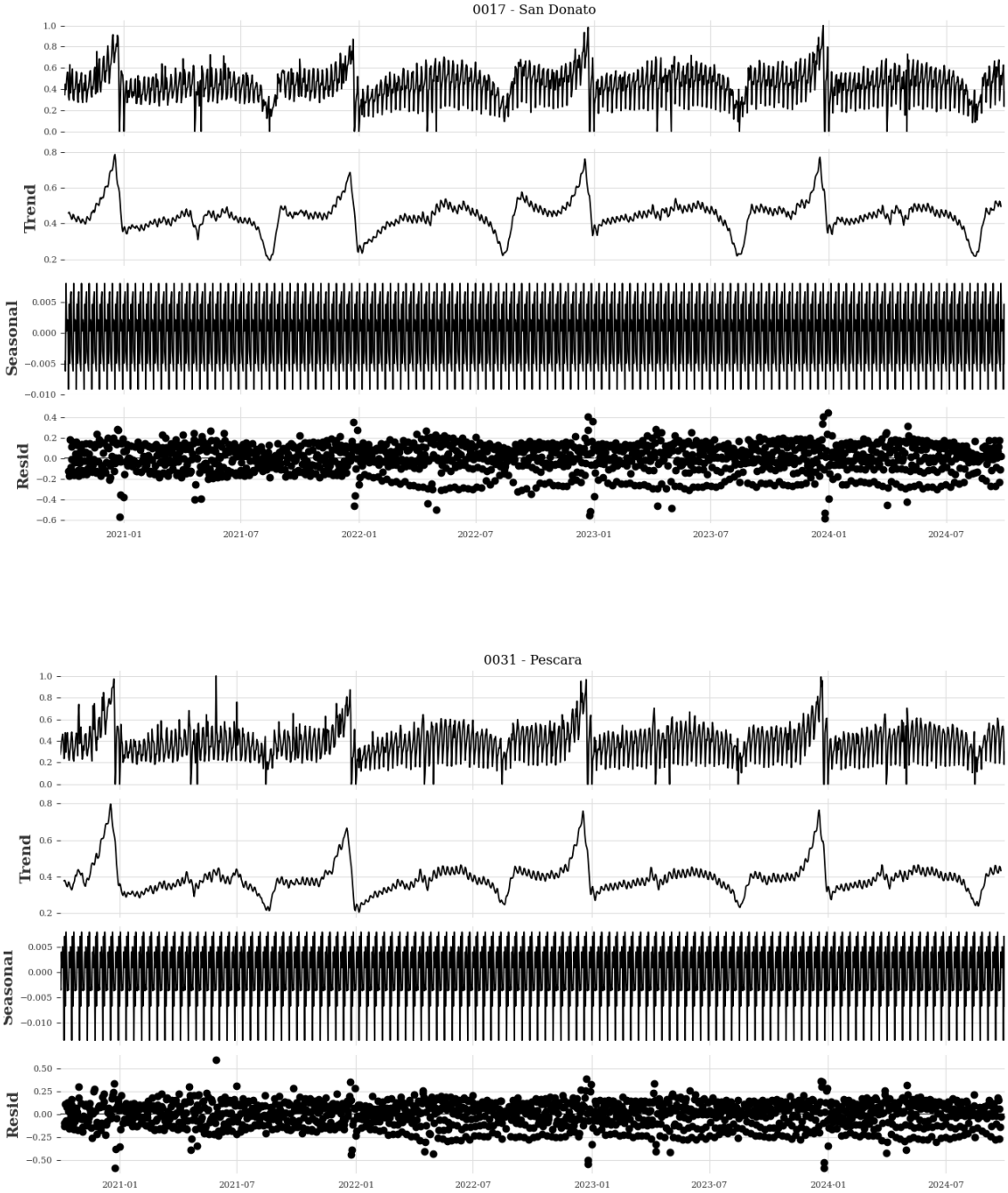


Figure 2.6: Time Series decompositions for the stores in San Donato and Pescara.

A comparison of the peaks reveals that some stores exhibit more pronounced fluctuations, while others remain more stable. For instance, San Donato and Brescia display stronger spikes during certain periods, whereas Pescara and Firenze show steadier but still significant surges. Such contrasts likely arise from differences in store size, local competition, and consumer demographics. Here the data is scaled for simplicity and data confidentiality purposes but it is important to remember that each store has different sales levels. This is a key information for defining a **metric** that is helpful for evaluating the models aligning with the business needs.

By highlighting a smaller subset of stores, this visualization provides a more focused look into the localized factors affecting daily sales. The interplay of recurring weekly cycles, seasonal fluctuations, and sporadic anomalies underscores the complexity of C&C operations, where consumer behavior, store level events, and broader external forces generate distinct yet partially synchronized demand patterns. These characteristics can be specifically inspected in the decompositions plotted for San Donato and Pescara in **figure 2.6**.

Figures 2.7 and 2.8 presents the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) for the four different Cash & Carry stores. By examining these plots, we gain insights into how strongly each day's sales relate to those of preceding days, across a range of time lags.
To begin with, all stores display a moderate correlation at short lags (particularly lag 1). This suggests that yesterday's sales have a meaningful influence on today's revenue, a phenomenon commonly seen in retail where customers might follow consistent shopping patterns from one day to the next. In other words, a relatively good or poor sales day often carries over some momentum into the following day, though the effect weakens quickly.

Moving beyond the first few lags, the ACF values tend to drop toward zero. This decline indicates a short-range dependency, implying that daily sales from two or three days ago exert progressively less impact on the present day's performance. In many retail scenarios, we would look for a noticeable weekly cycle around lag 7, since store traffic often follow a seven-day routine. In these plots, any peak or trough at lag 7 appears modest, though still visible for some stores. One possible explanation is that while a weekly rhythm does exist, perhaps with weekends or specific "market days" driving sales, other factors may blur this pattern. Discrepancies across the four stores further underline that each location may have unique operational nuances or customer demographics affecting its sales cadence. Beyond lag 7 and up to around lag 30, the ACF values remain fairly low, occasionally hovering near slight positive or negative levels. Interestingly, some negative correlations also appear at certain lags, though they are relatively small in magnitude.
Similar behaviors can be detected also by examining the PACF.

Overall, the plots highlight a strong short-term linkage in daily C&C sales, with the influence tapering off within roughly three days. These findings indicate that forecasting models will likely benefit from including lagged features that capture sales

from the previous day or two, and potentially a weekly seasonal component, while deeper lags (beyond a week or two) are less likely to provide substantial predictive value.

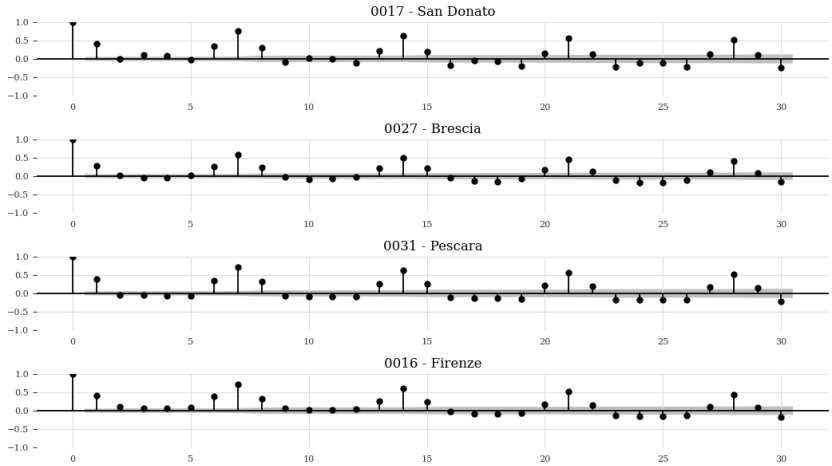


Figure 2.7: ACF of 4 sampled stores between all stores (C&C).

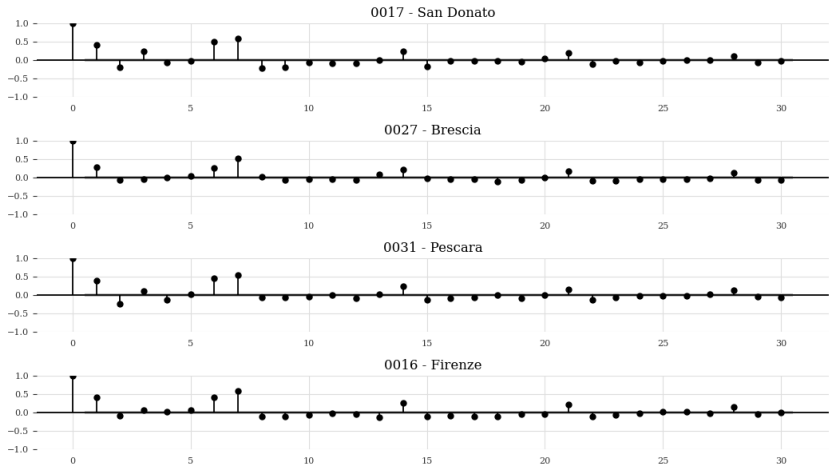


Figure 2.8: PACF of 4 sampled stores between all stores (C&C).

2.4.2 EDA for Full Service Delivery (FSD)

Figures 2.9 portrays the daily sales trajectories of four Full Service Delivery locations: Deposito Nord, Deposito Roma, Pescara, and Brescia, from October 2021 through September 2024, this time highlighting **Deposito Nord** and **Deposito Roma** because they handle the highest sales volumes due to their **exclusive focus** on the FSD channel.
In examining these deposit based sites more closely, one observes consistently higher and more stable sales volumes, reflecting a significant delivery operation with peaks that often cluster around seasonal or monthly intervals corresponding to routine restocking schedules or regular customer orders. Moreover, it is important to note that, unlike Cash & Carry, **FSD does not operate on Sundays**, which produces predictable dips to zero on those days without substantially reducing the overall elevated sales level seen at these deposits. In general, the trend at Deposito Nord and Deposito Roma remains markedly higher than what one might see at a regular store.
By contrast, Pescara and Brescia, which include FSD as part of a broader operational model, exhibit less consistent sales patterns. Their sales curves fluctuate more unpredictably sometimes aligning with the deposits' peaks (indicating periods of heightened delivery demand) but also frequently dipping to lower levels. This volatility result from local factors, such as a smaller coverage area, evolving customer requirements, or a more limited delivery clientele.

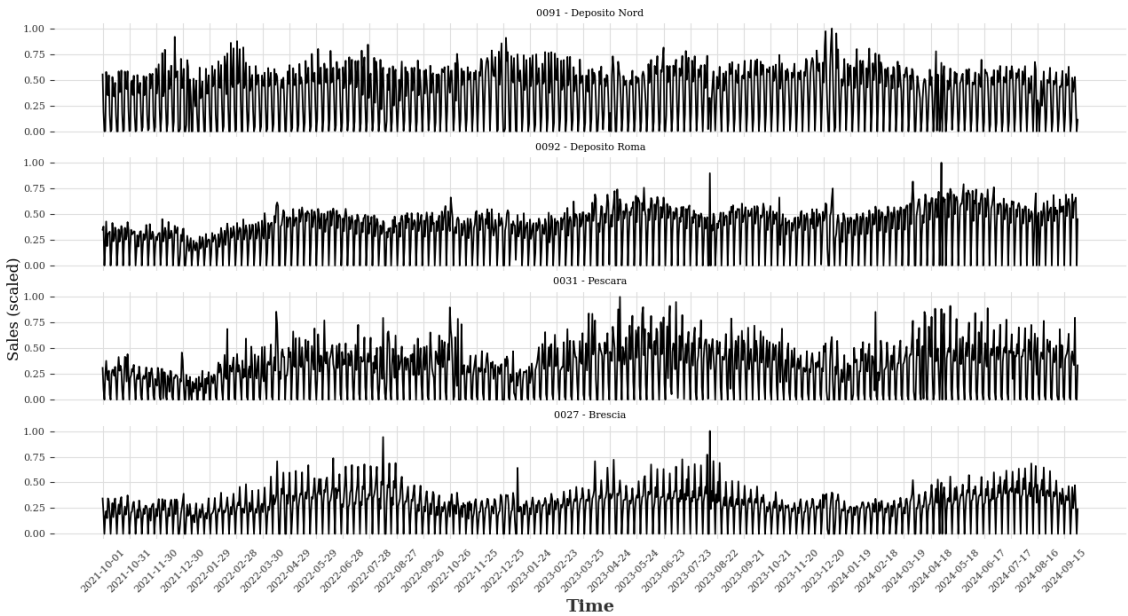


Figure 2.9: Line plots of 4 sampled stores between all stores (FSD).

In summary, the variations in magnitude, frequency of peaks, and overall patterns suggest that the behaviour of Full Service Delivery is closely tied to the role each location plays within the distribution network. Centralised deposits tend to be busier but show less susceptibility to sudden fluctuations, while smaller or hybrid operations, such as those in Pescara and Brescia, illustrate more inconsistent activity levels. This emphasizes the necessity of **adapting any forecasting methods** to

the specific operational realities of each site, acknowledging that not all FSD locations function under the same constraints or experience identical customer demand cycles.

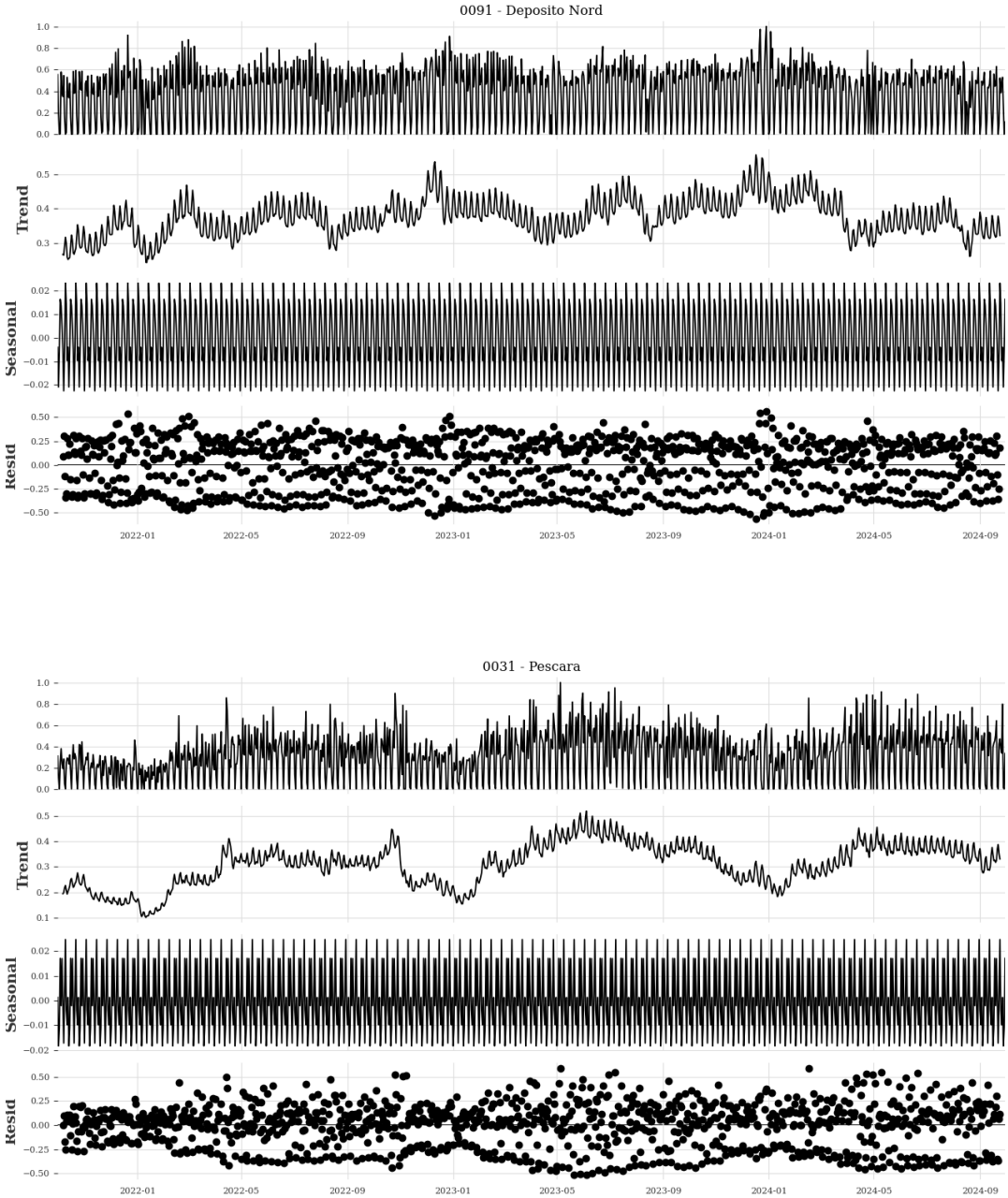


Figure 2.10: Time Series decompositions for Deposito Nord and Pescara (FSD).

When looking at the **Trend** component for the **Deposito Nord** (**Figures 2.10**), one can't help but notice a baseline that shifts gradually, with rises and falls that create multi-month arcs. These variations typically align with slight changes in recurring customer orders or expansions in operations, such as introducing new routes or increasing storage capacity. The **Seasonal** layer, which tends to oscillate regularly, reflects the weekly rhythms that drive demand. Meanwhile, the **Residual** panel shows scattered points around zero, emphasizing sudden spikes or dips in

demand. All in all, the decomposition of Deposito Nord reveals a combination of modest yet steady short-term cycles layered over a gently fluctuating long-term trend.
Pescara shows a noticeable fluctuation in its **trend** component, with more pronounced ups and downs over time. These larger swings indicate that local factors like changes in regional demand and temporary operational tweaks might influence the baseline sales greatly. Similarly, while the **seasonal** pattern in Pescara follows a regular rhythm, it appears to be more subdued compared to its overall variability. This means that the everyday fluctuations don't overshadow the larger month-to-month or quarter-to-quarter variations reflected in the trend. As a result, Pescara's **residuals** tend to spread out more widely, which suggests that unexpected spikes or drops may have a greater impact here than at Deposito Nord.

Regarding autocorrelation and partial autocorrelation as shown in **Figures 2.11, 2.12**, short-term dependencies are a key feature observed consistently across all four FSD locations. At lag 1, the autocorrelation function (ACF) tends to show a major positive value, indicating that the sales figures from one day have a tangible impact on sales the following day. This aligns well with what we'd expect in a delivery-centric setting: a day of high demand can create a ripple effect on the next day's sales, while a quieter day can also lead to a notable decrease in demand shortly thereafter. Conversely, the ACF at longer lags (beyond two or three days) tends to drop towards zero or only varies slightly around the baseline, suggesting that daily FSD sales exhibit a relatively brief memory.
The PACF reinforces these findings by showing that, after accounting for the effects of the previous day (lag 1), the correlations at further lags weaken considerably. This pattern implies that any notable autocorrelation observed at lag 2 or 3 is primarily a result of the day-to-day carryover effect, rather than indicating a genuine longer-term dependency.

One key difference from Cash & Carry scenarios is the impact of weekly rhythms. As FSD is closed on Sundays, the usual seven-day cycle gets a bit disrupted, which is evident in the often slight or inconsistent ACF/PACF spikes around lag 7. While some deposit sites may still show a gentle weekly pattern related to regular restocking by large clients, the lack of Sunday deliveries means we don't see a pronounced "weekend effect." Instead, the mix of positive and negative correlations at mid-range lags points to sporadic demand spikes or short-term logistical issues rather than a reliable seasonal trend. When we look at these autocorrelation insights together, it becomes clear that recent sales have the most important impact on current FSD activity. After a day or two, daily fluctuations seem too erratic to support a lasting correlation. The line plots and decompositions, along with the ACF/PACF patterns, emphasize the short-term, occasionally unstable nature of FSD sales, especially at smaller or hybrid locations. In contrast, deposit-only sites show a more stable demand profile but still lack important correlations at longer lags.

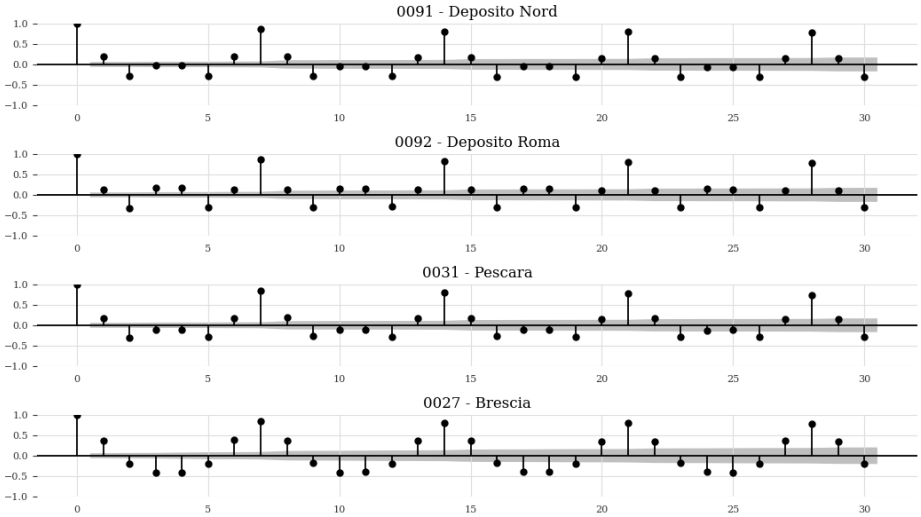


Figure 2.11: ACF of 4 sampled stores between all stores (FSD).

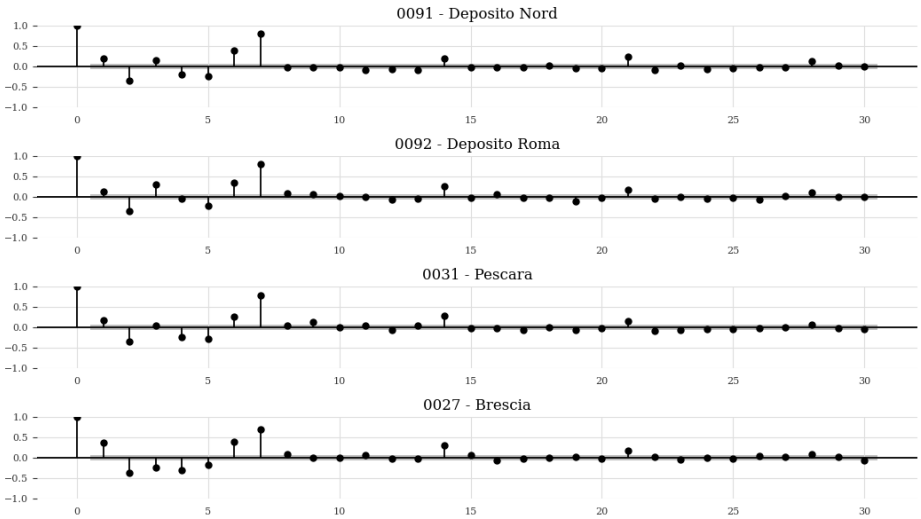


Figure 2.12: PACF of 4 sampled stores between all stores (FSD).

2.4.3 Feature Engineering

Once we've wrapped up the exploratory analysis for both Cash & Carry and Full Service Delivery channels, the next step in getting our dataset ready for forecasting models is **feature engineering**. In this project, the main aim of feature engineering is to bring in **covariates** extra variables that add to the raw sales data with additional insights that are likely to boost our predictive accuracy.
Now, covariates are different from our main target variable (daily sales) because they don't directly represent the outcome we're interested in; instead, they provide helpful context or explanatory power. For example, things like the day of the week or the month can help capture well-known patterns over time. By including these time-related signals, our model can better predict periodic fluctuations, such as increased sales on weekends or during certain monthly cycles.

The first covariate reveals the **day of the week** for each data point, ranging from 0 (Monday) to 6 (Sunday). In many retail environments, the influence of the day of the week can be quite major, as it often mirrors predictable consumer habits like those busy shopping days over the weekend or operational patterns, such as Sunday closures for FSD. By incorporating a day-of-the-week variable, the model can fine tune its forecasts to align with these familiar weekly rhythms, thereby minimising errors that could occur if they were overlooked.
The second covariate represents the **month of the year** corresponding to each sales record, ranging from 1 for January all the way to 12 for December. Seasonal patterns like the influx of summer tourists or trends in holiday shopping often present themselves at the monthly level. While the sales data may already show signs of seasonality in its raw form, having a specific month feature enables the model to more clearly understand and make use of these cycles, particularly when paired with other relevant variables.

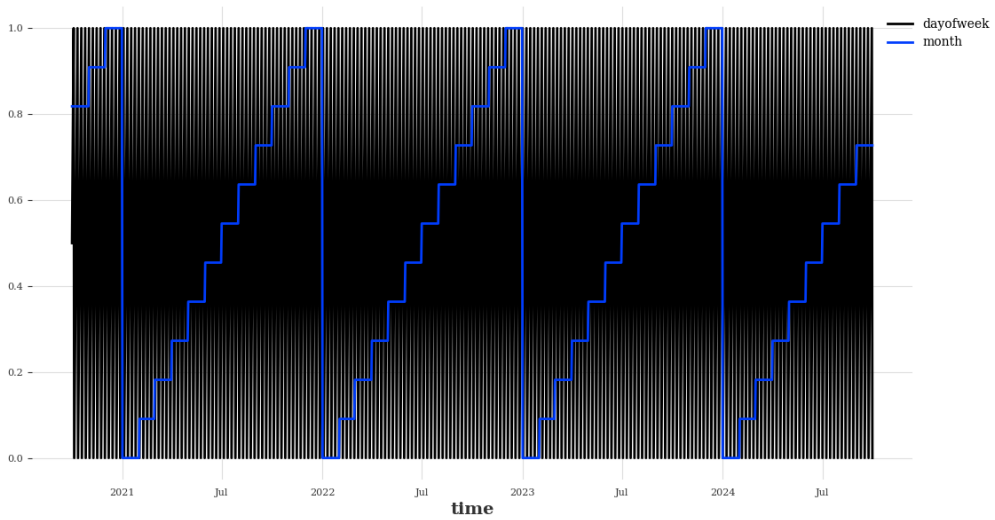


Figure 2.13: Covariates Plot.

After being created, these covariates can be neatly organised with the original time series data. This combined dataset links each daily sales record to both the day of the week and the month, creating a tidy and well-structured resource for later model training. Now that the model can identify whether a data point corresponds to a Friday in June rather than a Monday in February, it stands a much better chance of recognising and using these important temporal distinctions.

When we examine things from a modelling perspective, adding in variables for the day of the week and the month greatly reduces unexplained variance and boosts the model's ability to **generalise**. **Figures 2.13** illustrates the day-of-week variable (in black) cycling through every seven days and the month variable (in blue) progressing through values from 1 to 12 each year. This demonstrates how each date in the dataset is assigned both a weekday label and a monthly category. These two fundamental aspects of time often prove to be incredibly beneficial for enhancing our predictions, especially in fields like C&C and FSD, where cyclical patterns become evident from our exploratory analyses.

Chapter 3

Methodology and Implementation

Now that we've tidied up our dataset through cleaning and transformation, it's time to dive into the heart of our time series forecasting project. In this chapter, we'll walk you through how we chose the models—a look at why certain algorithms made the cut, plus a baseline model we can compare against. We'll also touch on how we fine-tuned the hyperparameters and the validation techniques we used to ensure our models were tested fairly and thoroughly.
We'll kick things off by exploring the reasons, both theoretical and practical, behind our choice of forecasting methods, focusing on how each one tackles the specific obstacles that come with daily sales data across multiple stores. Then, we'll provide a practical overview of how we implemented these models, emphasizing key decisions we made, the software libraries we relied on, and the computing power we allocated. Throughout, our goal is to give you a clear insight into the entire workflow from splitting the data into train and test sets to incorporating backtesting methods that mimic real-world scenarios. Finally, we'll introduce the evaluation metrics we used to measure how well our models performed, setting the stage for a comprehensive look at the results in the next chapter.

3.1 Overview of the Modeling Approach

The main goal of this study is to **predict daily sales** for both the Cash & Carry (C&C) and Full Service Delivery (FSD) channels over a **30-day period**. This will give us a month-long perspective that helps in planning and making operational decisions. We want to be able to anticipate inventory needs and allocate resources effectively across different store locations, each with its own sales patterns influenced by local conditions, scheduling habits, and consumer trends.
A tackle these challenges, we've chosen the **DARTS library** as our main software tool. DARTS provides model frameworks and tools specifically designed for working with multiple time series, making it perfect for managing the large number of stores in our project. By using DARTS, analysts can take advantage of improved processes for data prep, model setup, and performance metrics, which helps cut down on development time.

A pivotal decision here was to adopt a **global modeling approach**, rather than

maintaining a "local" model for each store. This strategy offers several critical benefits:

- **Efficiency of Resource Utilization** - In a corporate setting that emphasises cost control, having and maintaining a singular global model greatly reduces computation time and hardware usage compared to operating distinct models for each store. This approach also optimizes updates since there's only one training pipeline to manage, rather than multiple, which lightens the overall load on data teams;
- **Shared Learning Across Stores** - When we take a closer look at the patterns discovered in one store's data, we find that the global model is able to make more accurate predictions for other stores, especially those that exhibit broad similarities, such as spikes during holidays. This sharing of insights often results in models that generalise much more effectively than those customized to individual stores;
- **Scalability** - As the business continues to expand with new stores and channels, a unified global model can smoothly incorporate these time series data without the need to establish an entirely separate pipeline. Instead, this singular model learns continuously from the growing dataset, effectively preserving both time and financial resources.

These points align closely with the company's goal of minimising resource use and keeping operational costs down, all while ensuring accurate forecasts. Plus, getting rid of the need to manage many local models helps cut down on mistakes, version discrepancies, or inconsistent performance across various store specific solutions. Overall, using DARTS for global modelling offers a more adaptable, high-performing, and cost effective solution, perfectly suited for a constantly changing multi-store environment. The following sections provide a deeper exploration into model selection, hyperparameter tuning, and validation strategies, all centred around the core idea of a global modelling approach, guaranteeing that the system is both efficient and flexible.

3.2 Model Selection and Rationale

Drawing from the theoretical insights of the Literature Review, in this section we present the technical decisions scalability, resources limitations and data properties that made the final choices on models.

1. **Linear Regression as Baseline**
   We chose a simple linear regression as our baseline model because it's easy to implement, doesn't require much computational power, and has been a go-to method for forecasting for a long time. Researchers often rely on linear models to set a benchmark for comparing more complex techniques [11]. With its clear parameters and simple training process, linear regression makes it easy to see how much more accurate advanced methods can be;

2. **Random Forest**
   The Random Forest algorithm [3] provides a robust ensemble methodology that typically handles large features sets and captures nonlinear interactions without excessive fine-tuning. Its relative stability in the presence of noisy data and automatic handling of variable importance make it appealing for multi-store sales forecasting, where each store may respond differently to external factors or exhibit unique patterns. Random Forest's scalability also aligns with the company's resource constraints, as training can be parallelized efficiently;

3. **LightGBM**
   LightGBM is internationally characterized as a framework for gradient boosting with high speed and low memory consumption which are very important in a corporate context where low-latency retraining and tight budgetary constraints both need to be considered. Multistore forecasting is complex, and requires the processing of large scale datasets with intricate feature interdependencies, and empirically confirmed studies articulate that LightGBM performs well in these aspects of multistore forecasting in addition to various other retail sales predictions;

4. **LSTM**
   Long Short-Term Memory (LSTM) networks [12] are particularly effective at capturing complex sequential relations with long-term dependencies. This feature is especially useful in understanding incremental sales patterns like slow seasonal growths or shifts in consumer interests. In depth LSTM modeling, among other recurrent neural networks (RNNs), has shown remarkable proficiency in large scale time series competitions when the data ample enough to warrant the higher processing expense. Considering the multi-store, multi-channel environment of this research, capturing global patterns that more simplistic approaches overlook is possible with LSTM models, even though the models take a longer time to train.

Although **ARIMA** and models associated with it are still popular for single variable time series, they are not appropriate for forecasting high-dimensional retail data sets with a large number of observations. The implications of such a situation are quite obvious. It is certain that training individual ARIMA models for each shop (local modeling) will result in a high level of overhead, and hence, it will be the opposite of a global approach.

The model set choice of **Random Forest** and **Linear Regression** for interpretability and **LightGBM** and **LSTM** for predictive capability is such that values of average accuracy are achieved irrespective of the application. This harmonious equilibrium suits a business context where stores are plagued by issues and a centralized approach to modeling provides the major advantages of the global modeling such as memory and computation reduction and also learnings from one store can be implemented in another. The second half of the article speaks to the fact that all the models were tuned, then tested and compared at the end so that utilization of resources, execution times, and forecast accuracy do not get out of hand and thus they remain on the company's goal line

3.3 Hyperparameter Tuning

Now that our models are selected, the second phase involves the process of hyperparameter tuning to strike the optimal balance between model complexity, training duration, and prediction accuracy. By tweaking major parameters in each model like the number of estimators in Random Forest, the learning rate in LightGBM, or the architecture of the layers in LSTM, we are able to make the models learn the intricacies of daily sales data without overloading them with excessive computational loads. Each model's individual traits, ranging from dealing with day-by-day fluctuations to identifying seasonality, have to be weighed against the business imperative of keeping the updates timely and affordable.
Here, most of the hyperparameter tuning involved **random search** and an iterative **trial-and-error** process. Random search permitted coarse-grained exploration of parameter spaces for all the models at a much less exhaustive cost than a formal grid search. By randomly sampling parameter combinations such as learning rate, layer size, or tree depth, we could quickly identify where there was potential for performance. Manual, iterative tweaking (i.e., trial and error) then refined those parameters further. Less systematic than the application of advanced optimization methods (e.g., Bayesian methods), this process was nevertheless sufficient in balancing the exploratory needs of multiple models with the available resources and time constraints of the corporate environment.
Due to computational resource and time limitations, only a finite number of parameter combinations were tested up to a predefined cutoff point. Models that demonstrated promising performance in preliminary trials were further refined through more detailed searches, while those that failed to meet the required standards within the available resources were discarded.

3.3.1 Model-Specific Settings

Linear Regression Model Configuration

The **LinearRegressionModel** from the DARTS framework was set up with a specific arrangement of lags, future covariates, and an output chunk length:

```
model = LinearRegressionModel(

lags= [-1, -2, -3, -4, -5, -6, -7, -14, -21],

lags_future_covariates = (31,1),

output_chunk_length=1

)
```

- **lags** - This list of time steps specifies when past values of the target series are included as feature variables. Specifically, the model incorporates daily sales from the previous seven days, along with two additional reference points at 14 and 21 days prior, to capture both short-term and slightly longer-term dependencies. By incorporating weekly offsets (7, 14, 21), the model can also account for recurring weekly patterns;

- **lags_future_covariates** - This parameter determines the number of days in the future that the model is permitted to "look" at auxiliary or "future" covariates. The instruction to use covariates from 31 days ahead up to 1 day ahead is the tuple (31, 1). Use this type of range when forecasts look ahead to the complete month and when it is known that exogenous factors (e.g., calendar attributes) will influence sales;

- **output_chunk_length** - By setting the output chunk length to 1, the model is optimized to produce the one-step forecast at each step. Although the overall project is aimed at the 30-day horizon, the model makes the forecasts stepwise. The single-step approach is more inclined to yield tighter control over the updates while still permitting the application of the rolling or iterative mechanism to reach the target 30-day forecast horizon.

Under this configuration, the LinearRegressionModel is constructed to include both lagged sales and potential future covariates to predict on each day. With this structure, the model is able to leverage short-term autocorrelation (such as the week immediately prior) as well as intermediate memory at two- and three-week frequencies while making use of exogenous signals up to one month ahead. The outcome is that the implemented lags, covariate window, and output horizon strike a middle ground between capturing prevailing sales trends and allowing for somewhat longer-term effects within the confines of a resource-efficient linear model.

Random Forest Configuration

```
model = RandomForest(

lags= [-1, -2, -3, -4, -5, -6, -7, -14, -21],

lags_future_covariates = (31,1),

lags_past_covariates = list(range(-31, 0)),

n_estimators=500,

output_chunk_length=1,

max_depth = 20

)
```

To ensure a consistent comparison with the Linear Regression baseline, the **lags**, **lags_future_covariates** and **output_chunk_length** parameters remain the same as those used previously, thereby preserving identical input and forecast structures across models. This ensures any performance differences primarily reflect modeling capacity rather than divergent data handling. Specifically:

- **lags** - Identical to the baseline setup, these lags capture the past seven days along with 14-day and 21-day offsets, facilitating short-range and weekly pattern recognition;
- **lags_future_covariates** - Also matching the baseline approach, this grants visibility of up to 31 days of future exogenous data—covering the full 30-day forecast horizon—while maintaining day-level detail;
- **output_chunk_length** - As in the baseline, forecasts are produced **one day at a time**, ensuring a granular approach to prediction despite ultimately aiming at a 30-day forecast window;
- **lags_past_covariates** - Includes the previous 31 days of past covariates, incorporating additional information on past external signals or store-level events that may continue to have an impact on present sales;
- **n_estimators** - Specifies the number of decision trees in the forest. A larger ensemble tends to improve stability but raises computational demands;
- **max_depth** - Restricts tree depth to mitigate overfitting risks. This value seeks to balance complexity (capturing nuanced, store-specific behaviors) and training efficiency.

By keeping constant the same target lags and future covariates but varying richer past covariates, the RandomForest model is capable of discovering nonlinear trends and making use of longer history. This contrasts with the simpler structure of the baseline and demonstrates the manner in which the use of ensemble methods may lead to more effective predictions when faced with complex retail time series.

LightGBM Configuration

This setup generally preserves the baseline method's lags, future covariate range, and single-step output to ensure that performance gain is primarily attributed to the gradient-boosting framework of LightGBM and the addition of extra past covariates. Below is the explanation of each parameter's purpose and reasoning in brief.

```
model = LightGBMModel(

lags= [-1, -2, -3, -4, -5, -6, -7, -14, -21],

lags_future_covariates = (31,1),

lags_past_covariates = list(range(-30, 0)),

output_chunk_length = 1,

n_estimators = 400,

multi_models = True,

learning_rate = 0.03

)
```

- **lags** - Similar to the baseline model, this series of negative integers captures sales over the course of the past seven days and offsets at 14 and 21 days prior suitable to capture short term fluctuations and week-to-week repetitions;
- **lags_future_covariates** - Gives access to 31 days of future exogenous information, extending the 30 day forecast horizon of the project while leaving one day of extra coverage to accommodate final boundary conditions. This option is consistent with the baseline setup, maintaining the way in which future indicators (such as day-of-week or monthly indicators) are utilized;
- **lags_past_covariates** - Extends the model's contextual window to include the past 30 days of covariates, potentially capturing more enduring influences such as store level disruptions that could continue affecting current sales;
- **output_chunk_length** - Matches the baseline strategy in forecasting one day at a time;
- **n_estimators** - Controls the number of boosting iterations. Predictive robustness is generally enhanced in a big-box retail context by making the ensemble larger, at the expense of longer training times. 400 estimators here strike a pragmatic balance between accuracy and computational cost;

- **multi_models** - Trains separate regressors for each forecasting step even if the output_chunk_length parameter is one. Such structure could prove useful if partial backtesting or more advanced scenarios require day level regressions within the extended forecasting horizon.
- **learning_rate** - Creates a relatively **moderate boost rate** that diminishes the risk of overshoot in the parameter space. Although the lower rate yields smaller accuracy improvements, it also lengthens training times and thus necessitates a practical tradeoff within existing organizational resources.

By retaining the same fundamental lags and future covariate structure as the baseline, performance differences observed will then be due to the inherent modeling strength of LightGBM e.g., its capacity to use a more comprehensive pool of past covariates to put things in context. This structure thus constitutes a balanced solution that takes advantage of **gradient boosting** to uncover detailed data interaction while ensuring operational efficiency in the multi-store retail forecasting environment.

LSTM Configuration

```
model = RNNModel(

model="LSTM",

input_chunk_length=30,

output_chunk_length=1,

hidden_dim=75,

n_rnn_layers=5,

dropout=0.3,

training_length=30,

batch_size=64,

n_epochs=200,

optimizer_kwargs={"lr": 3e-3},

random_state=42

)
```

- **model** - Specifies the underlying recurrent structure as an **LSTM**, which is a type of RNN that is particularly appreciated for its ability to capture long-term dependencies. An LSTM is especially effective at uncovering latent or evolving trends in day-to-day sales data because it maintains information in its hidden states over long sequences;
- **input_chunk_length** - Determines the amount of past steps of the target series to input to the model at once. A 30-day window enables the LSTM to examine the history of around a month to discover periodic trends, seasonal spikes, or other influences that will affect the sales of the next day;
- **output_chunk_length** - Instructs the model to produce single-day forecasts at each prediction step;
- **hidden_dim** - Sets the number of neurons (feature maps) per hidden layer. A dimension of 75 is usually the appropriate tradeoff between the need to extract complex temporal structures and the organizational constraint of not overloading the computational resources;
- **n_rnn_layers** - Stacks five LSTM layers in sequence, enabling the model to learn higher and higher-level representations of the time series. Additional layers can increase learning capacity at the expense of longer training time, and careful balance is required for large-scale retail forecasting tasks;
- **dropout** - Randomly zeros out 30% of the neuron activations during training to prevent overfitting by not letting the model rely on the use of one specific path or one specific feature too heavily. Dropout is particularly relevant in deeper models where many of the parameters would otherwise memorize idiosyncratic variations in the data;
- **training_length** - Makes the LSTM see sequences of at least the input_chunk_length length during the process of learning to predict subsequent timesteps. Setting training_length equal to input_chunk_length helps the model learn to follow the month-long trends with consistency during training and inference;
- **batch_size** - Handles 64 time series examples in parallel at each training step. A moderate batch size will converge faster than very large batches by updating model weights more frequently than large batches with too many parameter updates per epoch;
- **n_epochs** - Makes 200 passes over the data to provide the model with more opportunities to refine its internal parameters and remove remaining error. Although higher epoch numbers will continue to refine performance, they also increase training time along with it, so 200 is generally the practical maximum for sales forecasting on a day-to-day basis;
- **optimizer_kwargs** - Sets the learning rate of the underlying optimiser to 0.003. This is the parameter that determines the gradient update speed; lower rates will make the fine-tuning more accurate but longer to train, while higher rates will overshoot local minima;

- **random_state** - Fixes the model's internal randomness, ensuring reproducible results across runs. For large-scale or corporate environments, consistent outcomes are essential for comparing different hyperparameter settings or model versions.

By designing the LSTM to process a 30-day input window and make single-day forecasts, this approach captures **short-horizon** flexibility along with the ability to ingest month-level trends. At the same time, multiple LSTM layers, dropout regularization, and a modest hidden dimension cooperate to improve the network's ability to capture nuanced trends while avoiding overfitting. Overall, this setup is particularly suited to the retail forecasting task of utilizing large quantities of store-level data in a **scalable and interpretable** fashion.

3.4 Data Splitting and Validation

For realistic simulation of the environment where the forecasting models will be applied, this study employed two complementary methods of validation. First, a standard **train/test split (Holdout Method)** was used, with the last month of data being an unaltered test set. By limiting the training of the models to the first part of the dataset, the out-of-sample performance generated is actually unseen, enabling a direct snapshot of the predictive power of each model.
Second, the **Historical Forecast backtesting** utility of the Darts library was employed to iteratively loop over the time series step by step, building rolling forecasts. This dynamic simulation mimics the way forecasts change in response to newly arrived sales data a scenario that repeats daily in a retail setting. With both methods applied in combination, we have both a concrete measure of accuracy via the train/test partition and a sense of how each approach might perform in regular, ongoing conditions.

3.4.1 Train/Test split (Holdout Method)

Adhering to best practice recommendations by Bergmeir [13] where train/test splitting is recommended as a pillar of unbiased model assessment in time series environments the data in this study was divided chronologically into training and test sets. Specifically, all observations up to **August 31, 2024**, constituted the training set, with the models being allowed to learn from a potentially long historical period capturing weekly seasonality, holiday variability, and other cyclic patterns. The remaining month **September 1 to September 30, 2024** was reserved as the test set so that no future observation could inform the patterns learned during training.
This is especially critical in time series forecasting, where the temporal order of the data should be preserved to represent actual usage. By labeling the last month as unseen data, the study is then capable of evaluating the performance of the models under simulated live forecasting window conditions, thereby establishing how well each of the models is capturing end of summer patterns, back to school factors, or indicators of the approaching holiday season. Moreover, employing a chronologically

uniform split adheres to the data leakage prevention principle so that the models make forecasts on the basis of past data alone.

3.4.2 Historical Forecast backtesting

Besides a standard train/test split, **backtesting**, alternatively referred to as time-series cross-validation, provides a **rolling, incremental assessment of predictive accuracy**. Unlike evaluating the accuracy of a model on a single hold out window, backtesting cycles over the dataset in temporal order, generating short-term forecasts at each point and comparing them with the next ground truth. This dynamic process is closer to actual deployment, where forecasts get refreshed on a daily or weekly basis as new data comes in.
Backtesting in the project was carried out on just those models that had earlier demonstrated strong performances in the train/test phase. With time and resource constraints in the corporate environment, just the high-performing candidates were further tested using the Darts **historical_forecasts()** function.

```
hfc = model.historical_forecasts(

series=full_data_scaled,

future_covariates=cov_scaled,

start=test_series.start_time(),

forecast_horizon=1,

stride=1,

verbose=True

)
```

By setting **forecast_horizon=1** and **stride=1**, the model produced one-day-ahead forecasts on a rolling fashion towards the latter end of the dataset. Each prediction was compared to its corresponding immediate future, hence demonstrating how a high performing model responds to day-to-day fluctuation in sales patterns. Such an incremental perspective not only enables finer-grained hyperparameter tuning (e.g. adjusting **input_chunk_length** or **output_chunk_length)** but also highlights each selected model's ability to withstand varying seasonalities or market dynamics.
Combining selective backtesting with the initial train/test validation gives a two-stage validation process.
The train/test approach initially narrows down the range of models to those that deliver stable out-of-sample accuracy. The historical forecasting process then gives an incremental insight into how the finalist models might behave in an operational setting. The combined process strengthens confidence in the reliability and responsiveness of the chosen approaches to real-world retail forecasting applications.

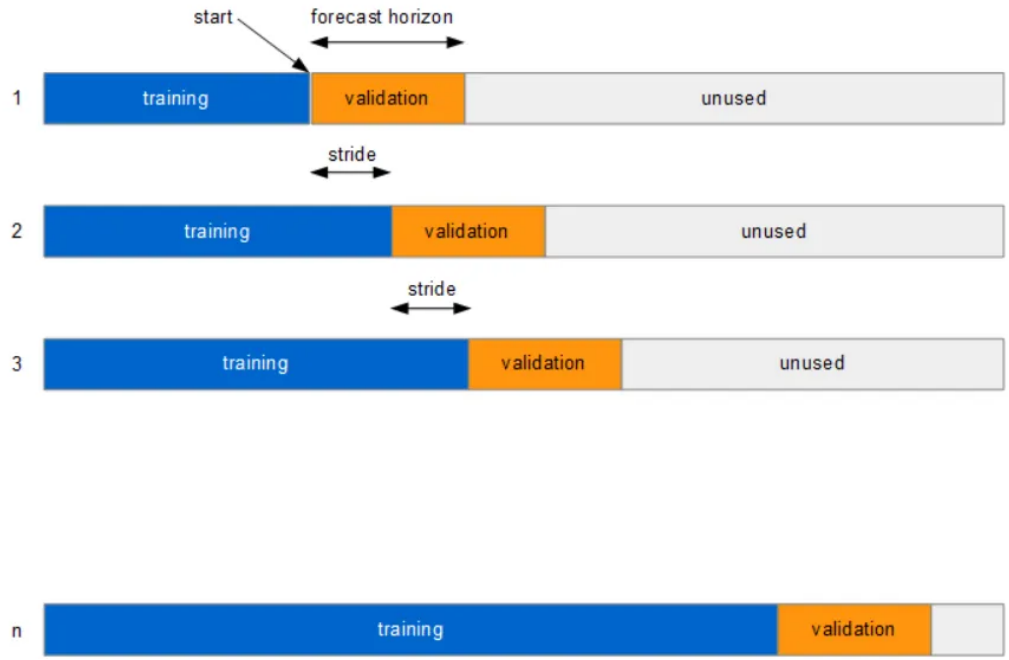


Figure 3.1: A rolling backtesting procedure illustrating how the training (blue) and validation (orange) windows move forward over time, emulating real-world forecast updates.

Independent tuning on the Full Service Delivery (FSD) channel in an attempt to catch its distinctive working patterns (e.g., not working on Sundays, varying demand) proved unsuccessful, however, and produced minimal improvements in performance. Iterative hyperparameter tuning produced similar forecasts within the FSD domain, suggesting that the underlying modeling method and hyperparameters were already capturing much of its dynamics. Therefore, a single set of models and configuration ended up being used across both Cash & Carry (C&C) and FSD, streamlining the final roll-out without loss of accuracy. Such consistency facilitates easier comparison across the two channels so that any performance differences reflect genuine differences in data patterns rather than modeling or tuning approach differences.

3.5 Evaluation Metrics and Criteria

When selecting that forecasting model that is best for operations, it is essential to frame the difference between forecast values and actual outcomes in a quantitative manner that can be systematically analyzed. Evaluation measures have this role to play by converting forecasting errors into measures that can indicate not only how often models over- or under-forecast but also their magnitude and business impact. This requirement for accurate performance measures is particularly pressing in multishop environments, wherein trends can differ significantly by store and time period and can be influenced by such drivers as promotions, holidays, and regional customer trends.
Using a consistent metric framework allows us to consider whether some models systematically make errors in one direction, provide a level playing ground to compare models with varying complexity levels, and ultimately link forecasting accuracy to tangible operating considerations such as inventory decisions or allocating resources. The remainder of this paper addresses selection of such evaluation metrics and how they can help to identify each model's relative strength and weaknesses in various store environments.

3.5.1 Mean Absolute Error (MAE)

The **Mean Absolute Error (MAE)** offers a straightforward measure of how closely a model's predictions align with actual observed values. For a set of N observations, where $\hat{y}_t$ denotes the forecast at time t and y_t the corresponding ground truth, the MAE is computed as:

$$\text{MAE} = \frac{1}{N} \sum_{t=1}^{N} |\hat{y}_t - y_t|$$

This procedure takes the absolute value of each residual i.e., actual minus predicted value and averages them. In doing so, it provides both under and over predictions with equal weighting, effectively capturing the average daily or weekly mismatch between actual and forecast.
MAE's strongest point is that it is easily understandable. An MAE of 20, for instance, indicates that on average, the model's predictions will be off by 20 units. This is a feature that makes MAE particularly valuable in retail environments, where decision makers want to have a general sense of how far off they will be on a day-to-day basis. But MAE does not penalize big errors as much as other measures (e.g., RMSE), which square each residual before averaging. Thus, MAE can be helpful in applications in which repeated small errors are more important than infrequent large errors.

For forecasting at multiple stores, the MAE provides a clear window into how close each forecasted daily value approximates actual volumes at a store. For every store,

absolute deviations between actual and forecast sales are calculated and averaged. Examining comparisons of MAE values between stores or between models makes it simple to see which configurations make more consistent day-to-day predictions without allowing random outliers to dominate the evaluation. This degree of granularity helps to determine if a model tends to consistently underforecast store demand (potentially leading to stockouts) or overforecast (leading to overstock).
Ultimately, MAE forms a central benchmark in this study's performance assessment, pairing interpretability with a balanced view of how daily forecasts translate into potential operational gains or inefficiencies. When combined with metrics that highlight larger forecasting errors or relative accuracy, MAE ensures that the final model selection recognizes both the frequency and the typical magnitude of forecast discrepancies across a diverse retail network.

3.5.2 Root Mean Squared Error (RMSE)

Root **Mean Squared Error (RMSE)** serves as a commonly adopted metric in forecast evaluation, particularly in scenarios where occasional large deviations hold substantial weight. Given N predictions, with $\hat{y}_t$ denoting the forecast at time t and y_t the corresponding observed value, RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}.$$

In contrast to metrics that rely on absolute differences, RMSE squares each residual prior to averaging, then takes the square root of the mean squared error. This squaring step amplifies larger forecast misses, thereby making RMSE more sensitive to occasional, yet potentially costly, inaccuracies.
Because RMSE penalizes substantial errors more heavily, it is especially informative in environments where a single large shortfall (or surplus) can cause significant operational disruptions. However, this emphasis on extreme deviations may sometimes make RMSE less intuitive for day-to-day business interpretations. For instance, an RMSE of 25 implies that, on average, the squared difference between predicted and actual values is 625 $units^2$ which is not as straightforwardly interpretable as a mean absolute deviation.

Within the multi-store retail scenario discussed here, RMSE serves as a useful alternative to the Mean Absolute Error (MAE). While MAE offers insight into common daily under or overestimations, RMSE emphasizes the effect and existence of outliers stores or days that show significantly large differences between predicted and actual sales. By tracking RMSE throughout all areas, decision-makers can determine if specific store segments or times show fluctuating demand patterns, potentially informing targeted modifications like enhanced promotional tactics or more detailed data gathering.

When analyzed with MAE and other metrics, RMSE aids in a thorough understanding of model performance, balancing the necessity to monitor overall forecast quality while aiming to reduce infrequent but potentially significant errors. This dual viewpoint is essential in large-scale retail forecasting, where both reliability in daily predictions and the reduction of rare errors are key factors in efficient resource management

3.5.3 Relative MAE (Normalized by Average Store Sales)

While absolute metrics such as MAE and RMSE provide a valuable gauge of forecast accuracy, they can be **misleading** in a multi-store setting where sales volumes vary widely. A fixed absolute error of 50 units may be relatively insignificant for a high-volume store but critically high for a smaller outlet. To address this discrepancy, the **Relative MAE** divides the absolute error by each store's mean sales, yielding a measure of how large the error is in proportion to typical daily performance.
For store i, let $\bar{y}^{(i)}$ denote its average sales, while $\text{MAE}^{(i)}$ represents the Mean Absolute Error computed for that store.
The Relative MAE is then calculated as:

$$\text{Relative MAE}^{(i)} = \frac{\text{MAE}^{(i)}}{\bar{y}^{(i)}}.$$

This formula ensures that forecasting discrepancies are interpreted relative to each store's baseline activity level. A store with an average daily sales figure of 500 units might treat a 50 unit MAE more leniently than a smaller store with average daily sales of 60 units, for which a 50 unit MAE would be devastating.

The Normalized Relative Mean Absolute Error has a valuable and easy to understand perspective of accuracy by showing forecast errors with the average daily sales volume of each store. This explanation works for the entire network of stores because no store performance evaluation is unfair considering the store size and typical sales volume differences.
From a business processes perspective it considers sales forecast errors in the daily operational context of the store. For example, sometimes even a small absolute forecast error can lead to big problems in inventory or staffing plans if it represents a large share of the store's average daily sales. On the other hand, large absolute errors in high volume stores could be less significant based on how they compare to normal business activity.
When averaged across the entire network, the Relative MAE provides an aggregate perspective on model performance that reflects accuracy across diverse sales environments rather than disproportionately emphasizing a few high-volume stores. This holistic viewpoint helps stakeholders discern which forecasting approaches deliver consistent performance at both the local and corporate levels.

Given the goal of aligning forecasting accuracy with real-world operational impacts, the Relative MAE emerges as an especially relevant metric for assessing and comparing models within this study. It offers nuanced insights that absolute error measures alone might overlook, highlighting how forecast inaccuracies genuinely influence day-to-day business decisions. Consequently, this metric provides essential guidance for evaluating model effectiveness in a way that aligns closely with the underlying business priorities of the organization.

Chapter 4

Results and Discussion

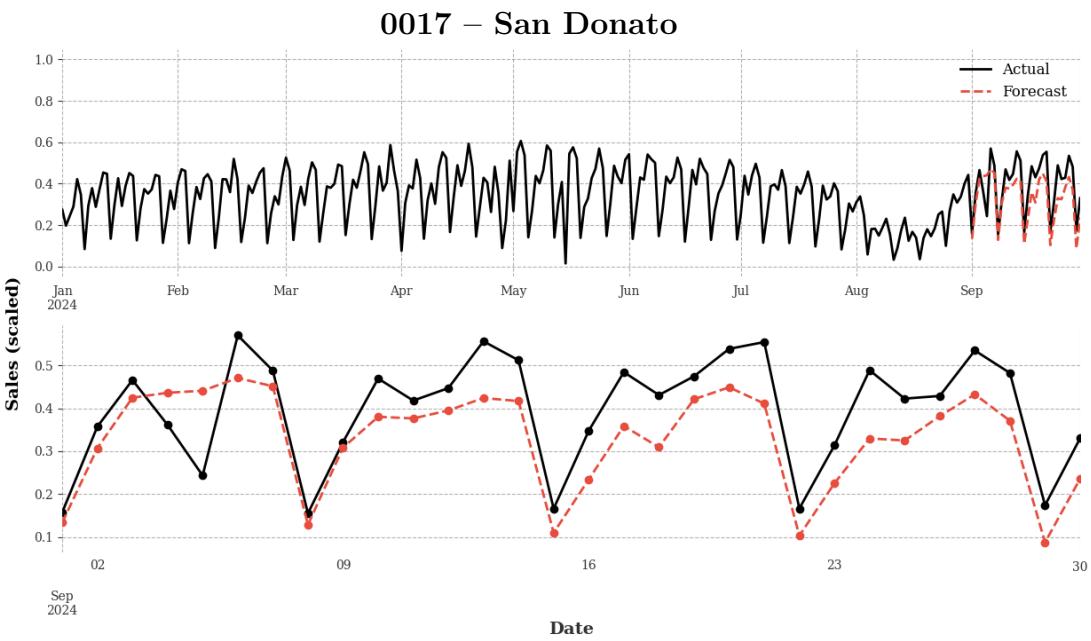
In this chapter, we present a comprehensive analysis of the forecasting results obtained through the application of various models described in the previous sections. The primary aim is to evaluate and compare these models both visually and quantitatively, determining which forecasting approach offers the most reliable predictions for future sales. This evaluation is pivotal, as it directly informs the choice of a forecasting system to be implemented within Metro's operational framework, aligning predictive accuracy with strategic business needs.
The discussion unfolds systematically, beginning with the Cash & Carry (C&C) channel, followed by the Full Service Delivery (FSD) channel. For each channel, forecasts for the four representative stores selected in previous chapters are presented and scrutinized. Results are examined through visual plots, capturing the models' predictive performances against observed sales trends, and subsequently quantified using the metrics detailed in the previous chapter—particularly focusing on Relative MAE due to its relevance in operational decision-making.
This structured analysis allows us not only to identify which model consistently provides the highest accuracy but also to understand its practical impact in terms of resource allocation, operational efficiency, and strategic planning at the store level.

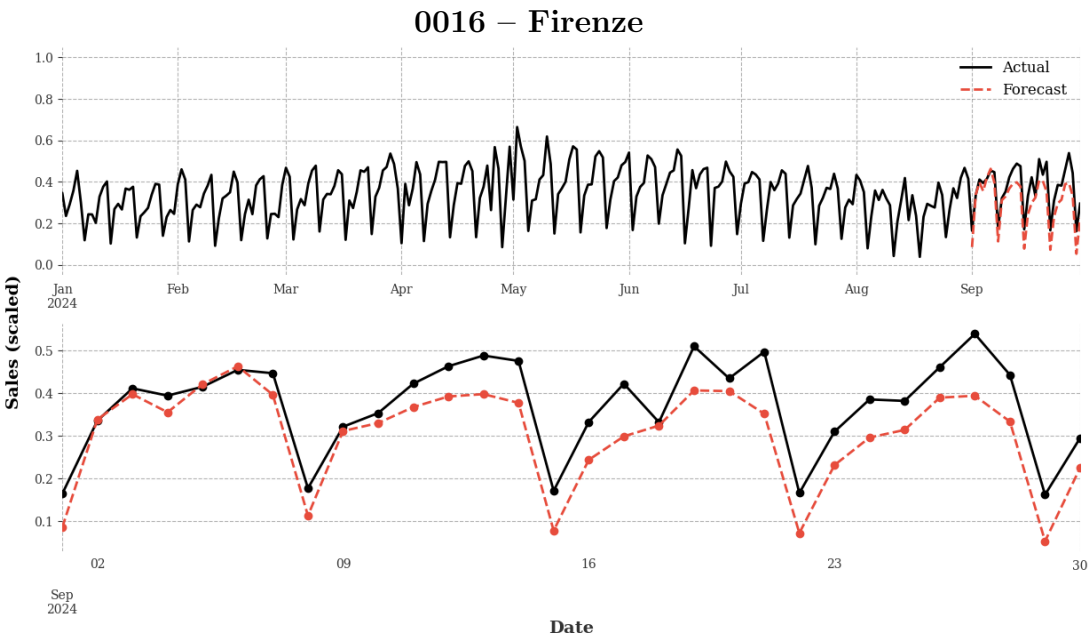
4.1 Cash & Carry Channel Results

4.1.1 Comparative Analysis of Forecasting Models

Linear Regression (Baseline)

We begin by presenting the visual predictions generated by the **Linear Regression** model for all four outlets (**San Donato, Firenze, Pescara, and Brescia**). This facilitates a visual inspection of the overall performance of the base model, capturing typical patterns and obvious deviations.
Each graph in the figures 4.1 contains two sections: the upper panel shows the overall sales trend from January 2024, offering a broader context, while the lower panel provides a magnified view of the forecasted versus actual sales specifically for the test period spanning from September 1 to September 30, 2024.


0017 – San Donato
Actual
Forecast
Sales (scaled)
Jan 2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep 2024
02
09
16
23
30
Date


0016 – Firenze
Actual
Forecast
Sales (scaled)
Jan 2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep 2024
02
09
16
23
30
Date

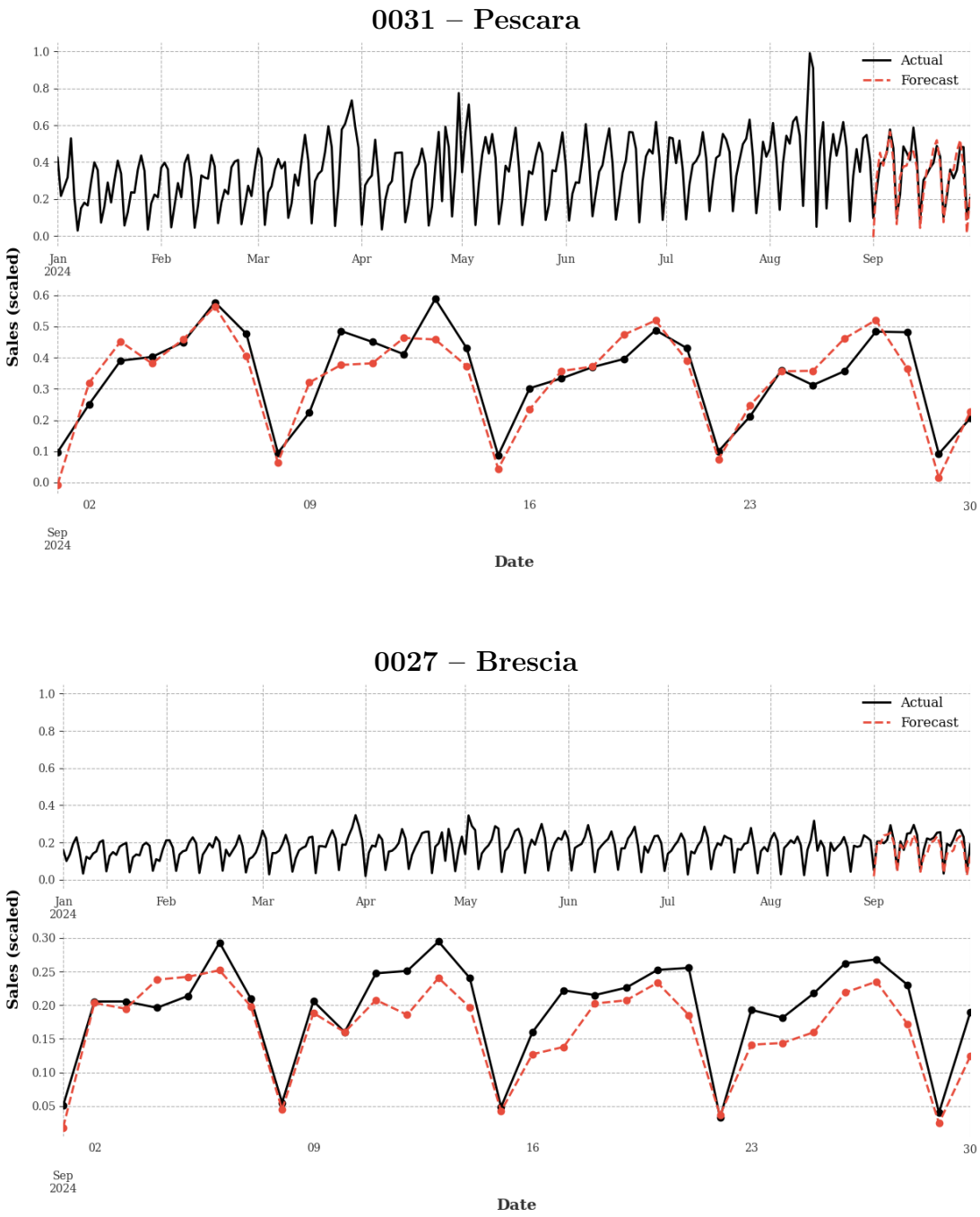


Figure 4.1: Linear Regression forecasts versus actual sales for the stores of San Donato, Firenze, Pescara, and Brescia.

Starting from the **San Donato** store, the LR model captures the general weekly pattern of sales with noticeable accuracy but tends to systematically underestimate peak sales values, indicating that it may be too simplistic to handle more nuanced fluctuations.

Similarly, for **Firenze**, the model manages to reflect the cyclical weekly patterns. However, forecasts still appear consistently lower than the actual peaks, with errors particularly noticeable during mid-month days where the actual sales spiked significantly above the prediction.

In **Pescara**, the Linear Regression model demonstrates improved performance, closely aligning with the observed data, especially in the second half of the month. While there remain minor inaccuracies, such as slight underestimations of peaks and overestimations of troughs, the forecast aligns more tightly with the actual values compared to San Donato and Firenze.

Finally, the store in **Brescia** shows a similar behavior to San Donato and Firenze, but on a smaller scale. The LR baseline reproduces the cyclical pattern effectively but consistently underestimates sales during peak periods, highlighting again its limitations in modeling sharp and irregular sales fluctuations.

Overall, while Linear Regression provides a useful baseline by capturing fundamental weekly patterns common to all stores, it clearly exhibits limitations in fully accounting for store-specific variability and short-term irregularities. Therefore, further evaluation of more sophisticated models is necessary and will be discussed in the subsequent sections.

Local Metrics

Store	**MAE**	**RMSE**	**Relative MAE**
0017 – San Donato	1.000	1.000	18.95%
0016 – Firenze	0.501	0.523	15.02%
0031 – Pescara	0.159	0.168	12.64%
0027 – Brescia	0.314	0.339	14.31%

Table 4.1: Error metrics (MAE, RMSE, and Relative MAE) evaluating the predictive accuracy of the Linear Regression model across the four analyzed stores.

Analyzing the error metrics presented in 4.1, it is evident that the Linear Regression model exhibits varying levels of predictive accuracy across the four stores.
San Donato shows the highest MAE and RMSE, each at 1.000, with a relative MAE of 18.95%. This indicates considerable forecasting challenges at this location, potentially reflecting a more volatile sales pattern or underlying factors not adequately captured by a simple linear model.
Firenze, while showing improvement compared to San Donato, still maintains relatively elevated MAE (0.501) and RMSE (0.523), with a relative MAE of 15.02%.

This moderate forecasting error suggests room for enhancement, though it indicates a generally more predictable sales pattern compared to San Donato.
Pescara demonstrates notably superior performance with an MAE of just 0.159 and an RMSE of 0.168, alongside a relative MAE of only 12.64%. These metrics suggest a strong alignment between predicted and actual sales, implying more consistent or less complex sales behavior for this store.
Brescia's metrics sit between the extremes, with an MAE of 0.314, RMSE of 0.339, and a relative MAE of 14.31%, indicating relatively stable predictability but still a meaningful margin for improvement compared to Pescara.

Collectively, the error metrics highlight the linear model's varying efficacy across locations, emphasizing the complexity and uniqueness of each store's sales dynamics. The relative MAE underscores practical implications, revealing how significant each store's forecast error is in proportion to its typical daily sales. Given these results, a more sophisticated forecasting method could provide substantial improvement, particularly for stores like San Donato and Firenze.

Global Metrics

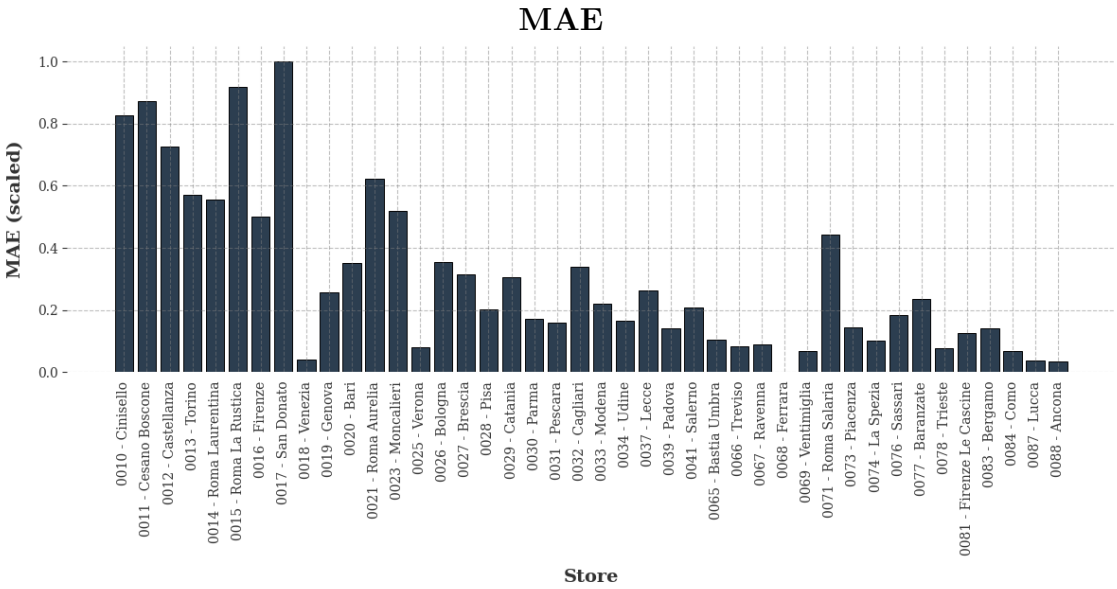


Figure 4.2: Mean Absolute Error (MAE) per store, highlighting the absolute differences in forecasting performance of the linear regression model.

The Mean Absolute Error (MAE) bar plot illustrates the predictive accuracy of the linear regression model across individual Metro Italia stores. Significant variability can be observed, highlighting how forecasting performance varies depending on specific store dynamics. Certain locations, such as San Donato, Cinisello, and Roma Salaria, show noticeably higher error levels, suggesting that these stores have sales patterns or local conditions less effectively captured by this baseline model. Conversely, stores like Venezia and Ancona demonstrate lower MAE values, indicating a relatively better predictive performance in these contexts.

Across all stores, the **global MAE is approximately 0.3001**, serving as a general measure of overall forecasting accuracy for the linear regression model. This figure offers a baseline against which to assess more advanced models, guiding subsequent evaluation and potential selection processes.

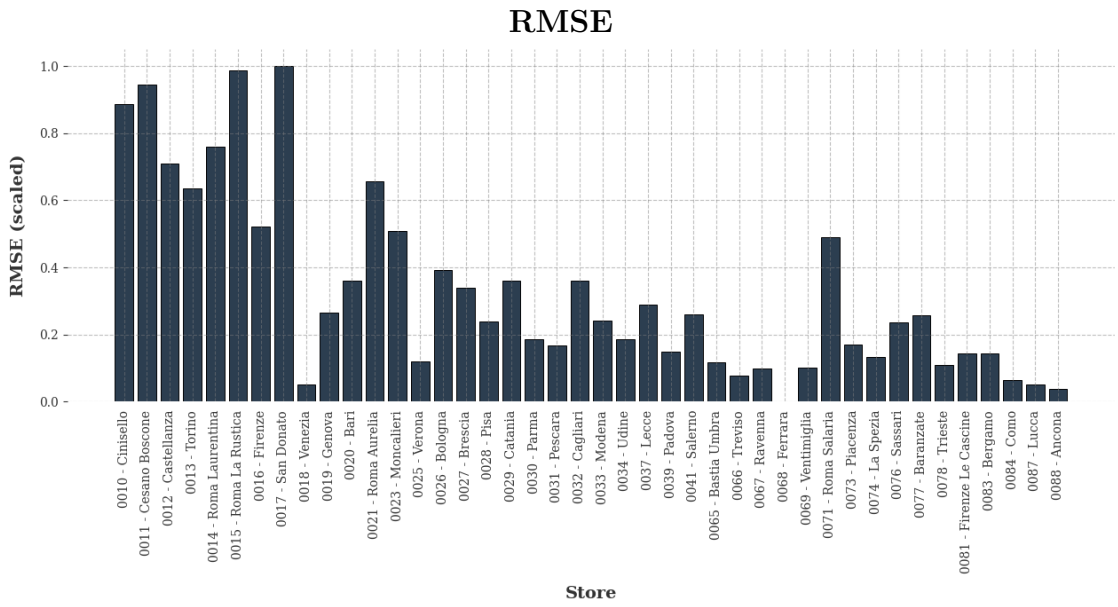


Figure 4.3: Root Mean Square Error (RMSE) per store, illustrating the dispersion of forecast errors across individual store predictions.

The root mean square error (RMSE) bar graph highlights the most substantial forecast deviations due to the squaring effect inherent in this metric. Similar to the MAE results, considerable variability among locations is evident, with San Donato, Roma Salaria, and Cinisello again showing the highest errors, reflecting pronounced fluctuations or extreme sales days at these locations. On the other hand, stores such as Venezia, Ancona and Lucca show lower RMSE values, suggesting more uniform and predictable sales patterns.
The overall RMSE, calculated as the **average RMSE of all stores, is about 0.3287**, indicating an overall moderate forecasting performance. By penalizing larger errors more, this measure emphasizes the need for improvement through potentially more sophisticated forecasting methods, especially in stores with particularly high error values.

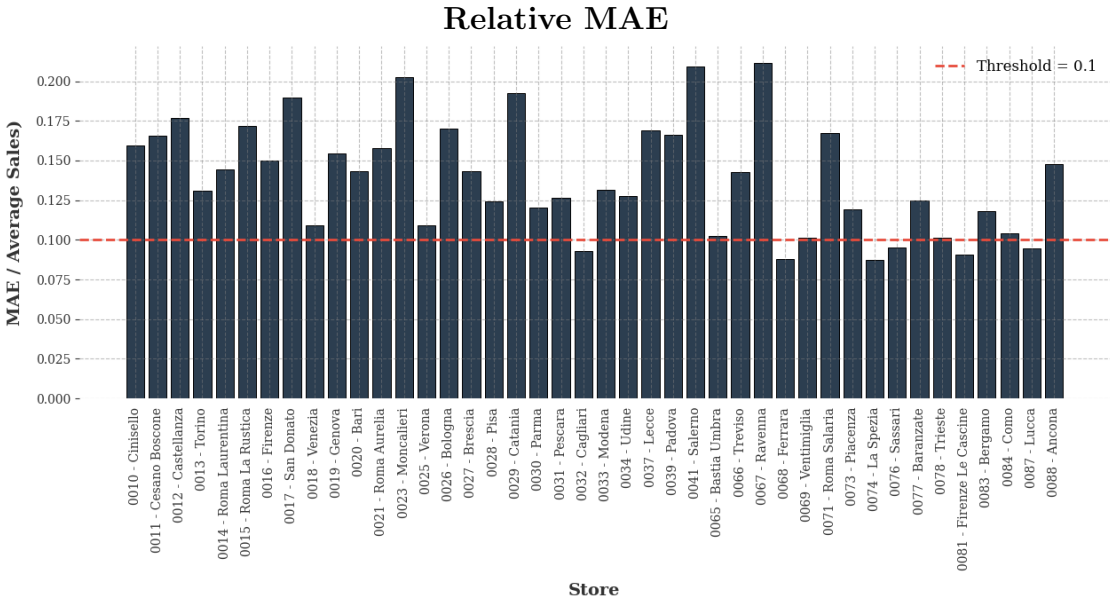
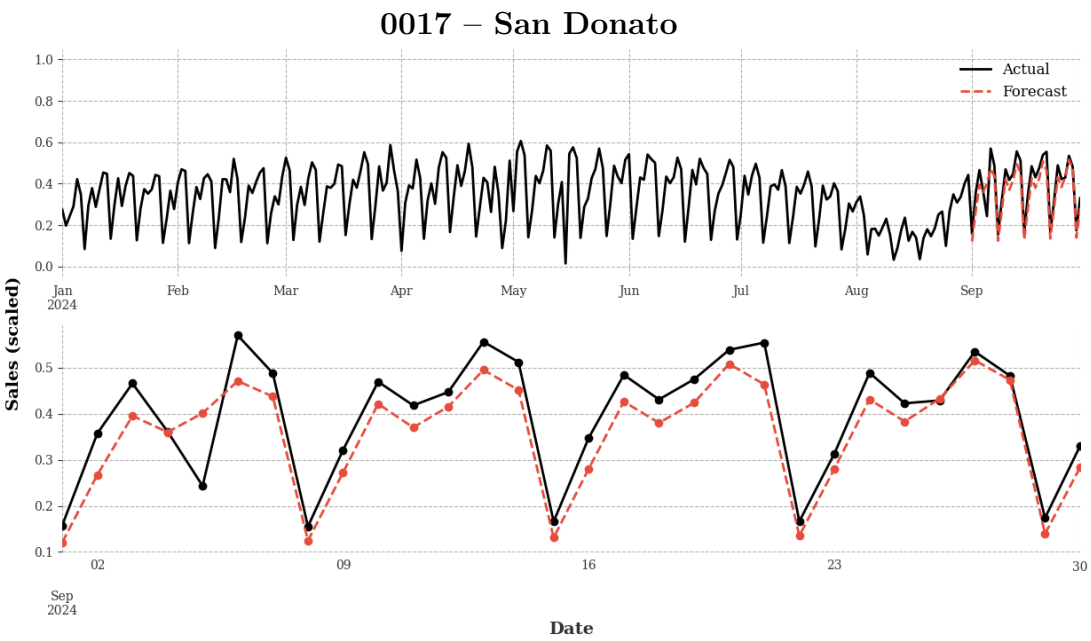


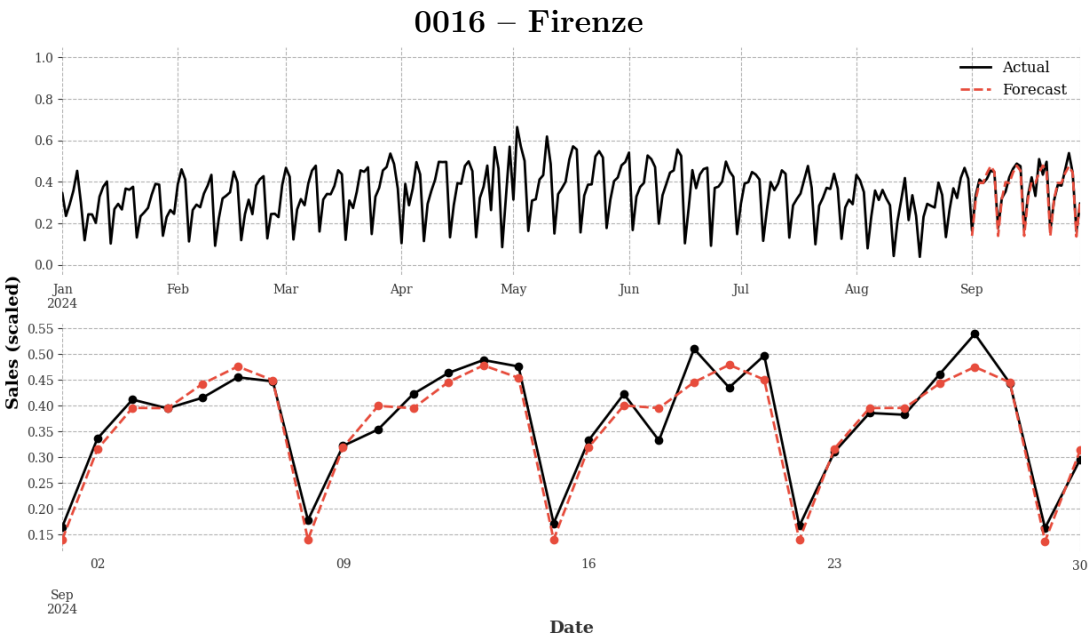
Figure 4.4: Relative MAE per store compared against the company's optimal threshold (10%), providing context on the significance of prediction errors relative to typical sales volumes at each store.

The bar plot of Relative MAE offers greater clarity by contextualizing errors within the typical operational performance of each store, thus leveling the evaluation across different store sizes.
The figures 4.4 displays substantial consistency across stores, with most locations falling around or slightly above the defined threshold of 10%, considered an optimal forecasting target for Metro Italia. Stores such as Venezia, Bari, and Salerno closely meet or fall under this threshold, reflecting highly efficient forecasting accuracy relative to their typical sales performance. Conversely, Roma Salaria and San Donato exhibit the largest deviations, significantly exceeding the threshold and indicating that forecast errors could meaningfully impact operational decisions and resource allocation in these locations.
The **global Relative MAE is approximately 13.89%**, which, although reasonably close to the optimal target, still signals room for improvement. Considering this metric's high practical relevance to store-level decision-making, the analysis underscores the importance of refining forecasting methods specifically for stores where errors surpass acceptable operational limits.

Random Forest

The following graphical results (figures 4.5) illustrate the predictive performance of the **Random Forest** model across four selected stores: San Donato, Firenze, Pescara, and Brescia.


0017 – San Donato
Actual
Forecast
Sales (scaled)
Jan 2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep 2024
Date


0016 – Firenze
Actual
Forecast
Sales (scaled)
Jan 2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep 2024
Date

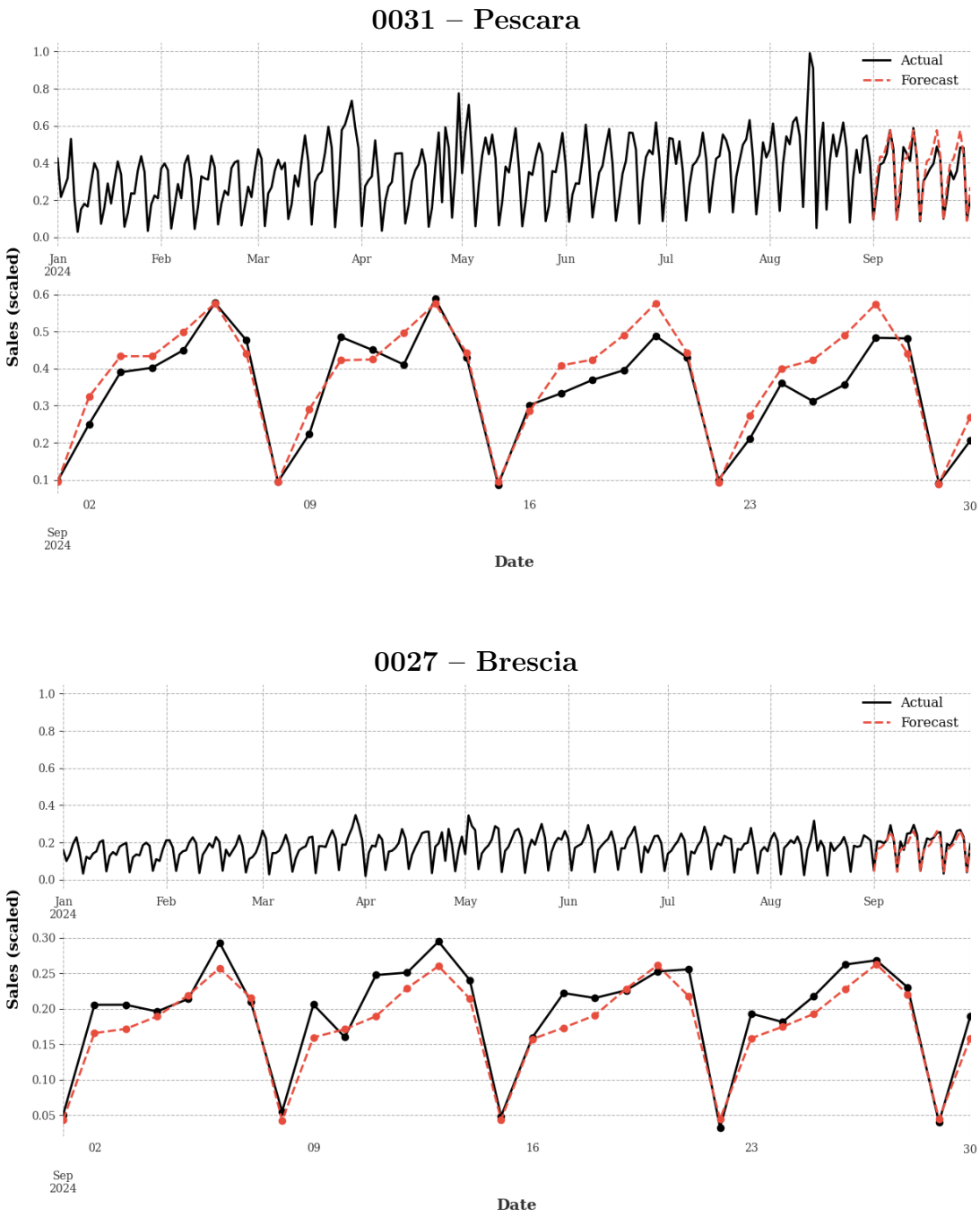


Figure 4.5: Random Forest forecasts versus actual sales for the stores of San Donato, Firenze, Pescara, and Brescia.

For the **San Donato** store, the Random Forest forecast exhibits enhanced predictive fidelity relative to the baseline, effectively capturing periodic fluctuations characteristic of weekly seasonality. However, residual discrepancies persist, particularly during peak demand intervals, implying that the model may still lack certain exogenous or highly localized factors required to achieve optimal forecasting accuracy.

In the **Firenze** store scenario, the Random Forest model exhibits strong alignment between predicted and actual sales trajectories. The forecasts accurately replicate not only the inherent weekly cyclical patterns but also daily fluctuations, signifying a substantial reduction in prediction errors and underscoring the model's adeptness in capturing intricate local demand behaviors.

The forecasting performance for the **Pescara** store further emphasizes the Random Forest model's capability, with predictions consistently mirroring the observed sales values throughout the evaluation period. The model adeptly captures subtle variations in daily sales, reinforcing its efficacy in adapting to local consumption dynamics and intrinsic periodicities within the data.

Forecast outcomes for **Brescia** similarly validate the effectiveness of the Random Forest approach, demonstrating close concordance between predicted and observed sales patterns. The model effectively accounts for store-specific characteristics, delivering stable and reliable forecasts even within contexts characterized by lower volume transactions and more subtle fluctuations.

The Random Forest model implemented within the Cash & Carry channel demonstrates a clear and notable improvement in forecasting accuracy compared to the previously evaluated linear regression baseline. Specifically, the model has effectively reduced prediction errors and provided a more precise depiction of complex, store-specific daily and weekly sales patterns. Nevertheless, certain forecast discrepancies, particularly during peak demand periods, suggest there remains room for refinement. Such limitations underscore the necessity of exploring alternative or complementary forecasting approaches capable of further elevating predictive reliability and operational relevance.

Local Metrics

Store	**MAE**	**RMSE**	**Relative MAE**
0017 – San Donato	1.000	1.000	14.98%
0016 – Firenze	0.164	0.168	5.38%
0031 – Pescara	0.175	0.206	10.71%
0027 – Brescia	0.280	0.292	10.06%

Table 4.2: Error metrics (MAE, RMSE, and Relative MAE) evaluating the predictive accuracy of the Random Forest model across the four analyzed stores.

Analyzing the predictive accuracy of the Random Forest (RF) model across the selected Cash & Carry stores reveals significant enhancements compared to the Linear Regression (LR) baseline previously discussed. As presented in 4.2, **San Donato**, which serves as the scaling reference due to its higher sales volume, retains the maximum scaled error values (MAE and RMSE both equal to 1). However, the substantial reduction in the Relative MAE from 18.95% (LR) to 14.98% (RF) demonstrates marked improvement, indicating that the Random Forest is better attuned to capturing San Donato's daily sales fluctuations.
The improvement becomes even more apparent in **Firenze**, where the MAE drastically decreased from 0.501 (LR) to 0.164 (RF) and the Relative MAE significantly dropped from 15.02% to 5.38%. Such substantial reductions illustrate Random Forest's enhanced ability to handle complex store specific dynamics and suggest that Firenze's sales patterns, possibly characterized by clearer weekly cyclicality or fewer anomalies, were captured more effectively by the Random Forest approach.
In **Pescara and Brescia**, similar patterns emerge, reinforcing Random Forest's robustness. Pescara's Relative MAE decreased notably from 12.64% (LR) to 10.71% (RF), while Brescia's Relative MAE similarly improved from 14.31% (LR) to 10.06% (RF). This uniform improvement underscores that Random Forest's capacity to generalize effectively across diverse sales environments surpasses that of the simpler Linear Regression.

Overall, these metrics confirm the Random Forest model's superior forecasting performance relative to Linear Regression, effectively capturing nuances within daily store level sales and consequently enhancing decision making accuracy in inventory management and resource allocation contexts. Nonetheless, despite evident improvements, the Random Forest model's performance suggests there remains potential for further optimization in subsequent forecasting approaches.

Global Metrics

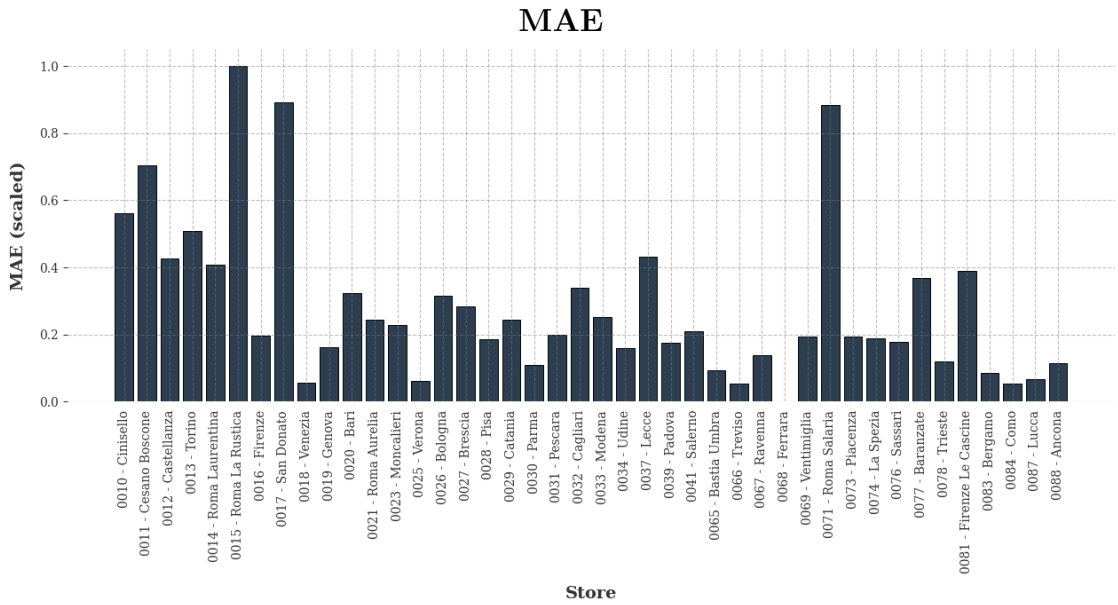


Figure 4.6: Mean Absolute Error (MAE) per store, highlighting the absolute differences in forecasting performance of the Random Forest model.

The bar plot of the Mean Absolute Error (MAE) for the Random Forest model across all Metro Italia stores reveals notable improvements in forecast accuracy compared to the Linear Regression baseline. While certain stores, such as San Donato and Roma Salaria, continue to exhibit higher errors indicating challenges in capturing complex local sales dynamics the overall distribution of errors is markedly lower and more consistent across the majority of locations. Most stores now present moderate error levels, indicating that Random Forest has successfully leveraged the available historical and covariate data to refine daily sales predictions. Nonetheless, the **global MAE remains at a relatively elevated level (0.2598)**, signifying the existence of residual prediction inaccuracies and the potential for further model refinement.

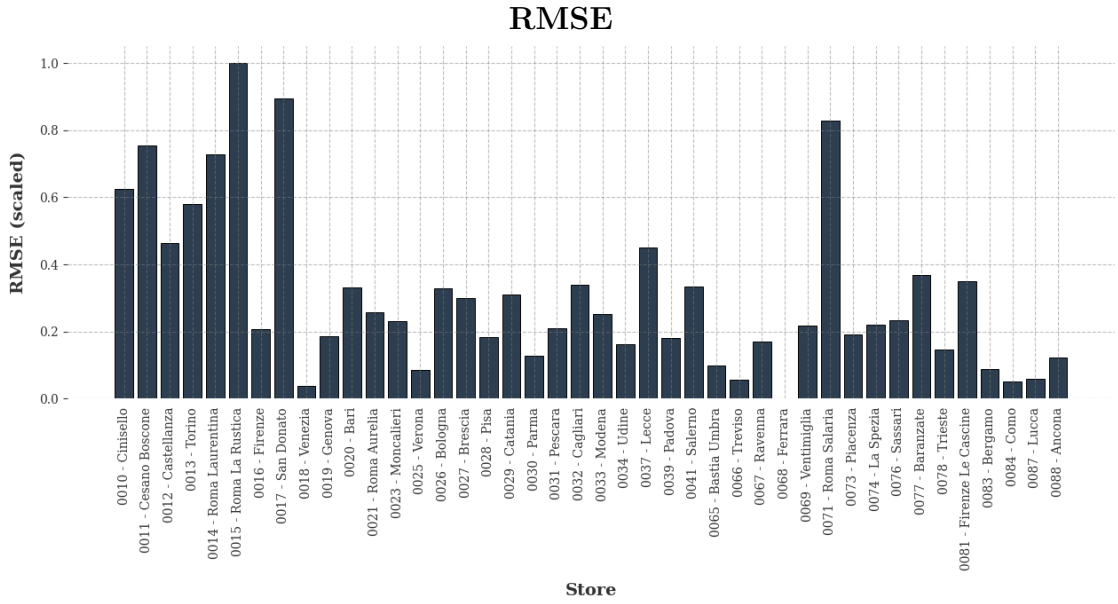


Figure 4.7: Root Mean Square Error (RMSE) per store, highlighting the absolute differences in forecasting performance of the Random Forest model.

The bar plot displaying the Root Mean Squared Error (RMSE) for the Random Forest model reveals consistency with the insights gathered from the MAE metric. The RMSE highlights stores where forecasting errors exhibit greater variability, penalizing larger deviations more significantly. Similar to the MAE results, locations such as San Donato and Roma Salaria continue to present higher RMSE values, reflecting difficulties in accurately modeling their sales volatility. Nevertheless, the majority of stores demonstrate improved and more stable RMSE values relative to the Linear Regression baseline, indicating Random Forest's ability to mitigate extreme forecasting errors effectively. The overall **global RMSE, averaging at approximately 0.2983**, underscores the presence of moderate residual errors, suggesting further enhancements could still be achieved.

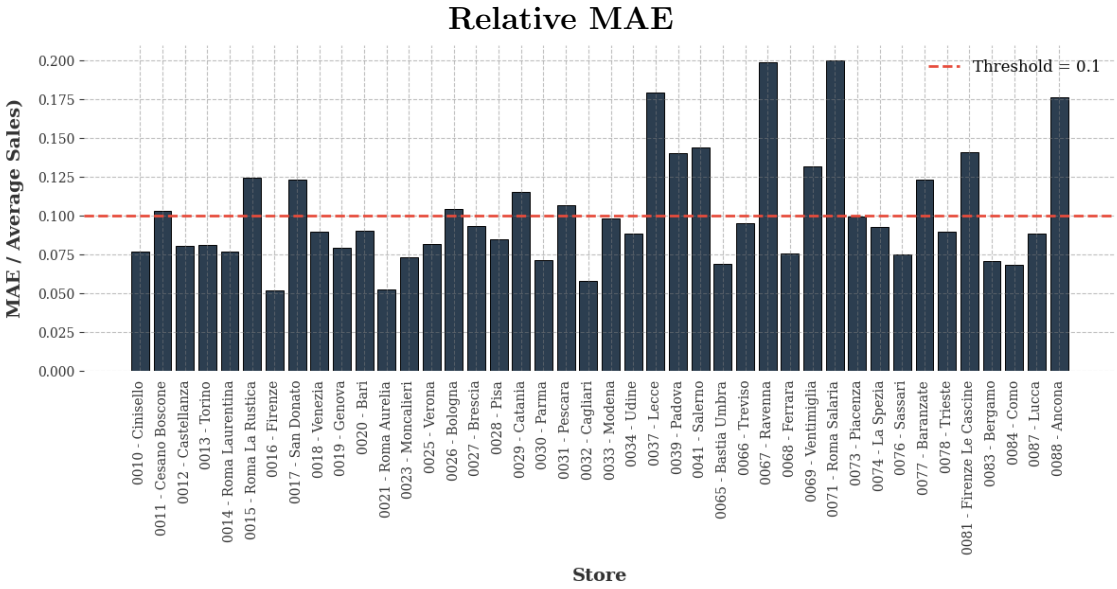
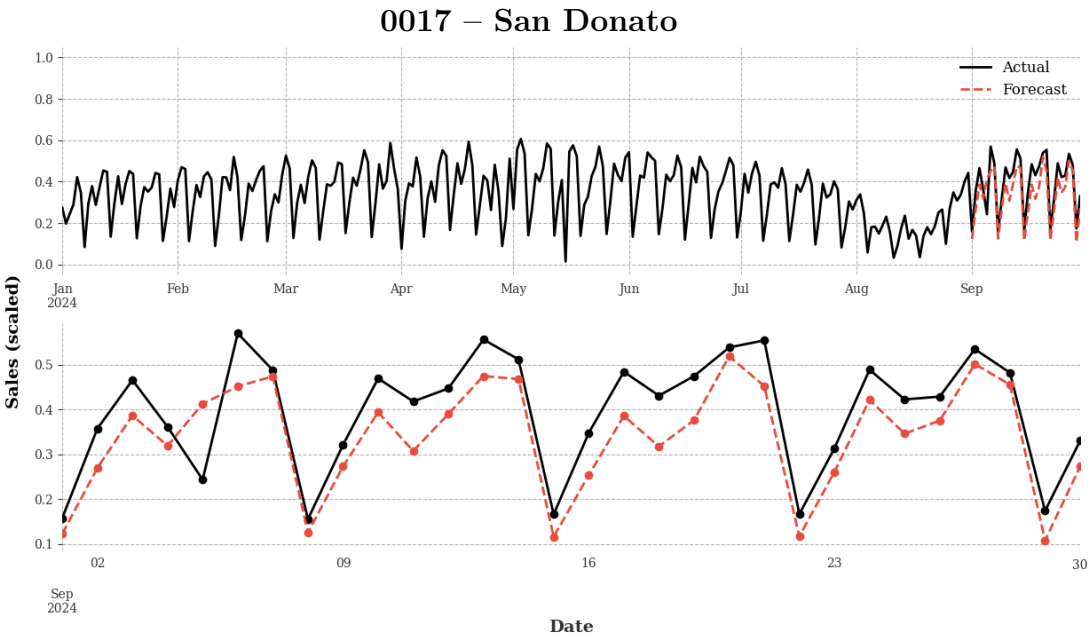


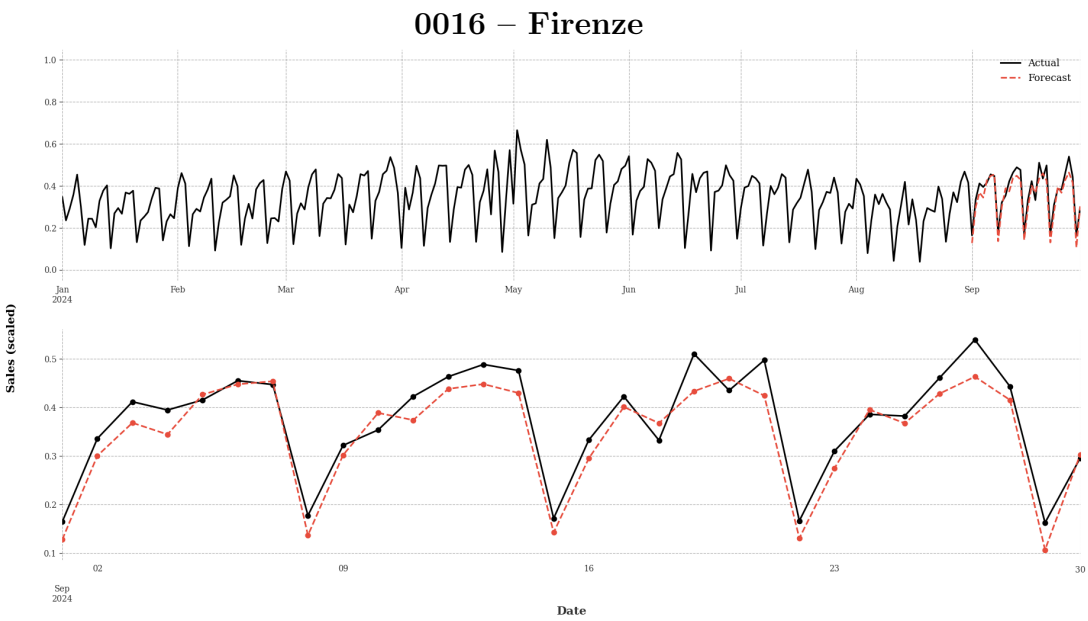
Figure 4.8: Relative MAE per store compared against the company's optimal threshold (10%), providing context on the significance of prediction errors relative to typical sales volumes at each store.

The Relative MAE bar plot for the Random Forest model further emphasizes its forecasting capability from a business standpoint, offering insights into model performance in proportion to average store sales. With a **global Relative MAE of 11.06%**, the Random Forest represents a meaningful improvement over the Linear Regression model's performance (13.89%). The 10% threshold indicates an acceptable level of forecast accuracy desired by Metro Italia; several stores now either meet or come very close to this benchmark. Nonetheless, several locations remain above the threshold, highlighting areas requiring further refinement or a tailored modeling approach. Overall, these results underscore the Random Forest model's substantial progress towards a business-acceptable forecasting accuracy, while clearly pinpointing opportunities for additional optimization.

LightGBM

The following plots (figures 4.9) illustrate how the **LightGBM** model performs in predicting daily sales for four representative Cash & Carry stores: San Donato, Firenze, Pescara, and Brescia.


0017 – San Donato
Actual
Forecast
Sales (scaled)
Date
Jan 2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep 2024
02
09
16
23
30


0016 – Firenze
Actual
Forecast
Sales (scaled)
Date

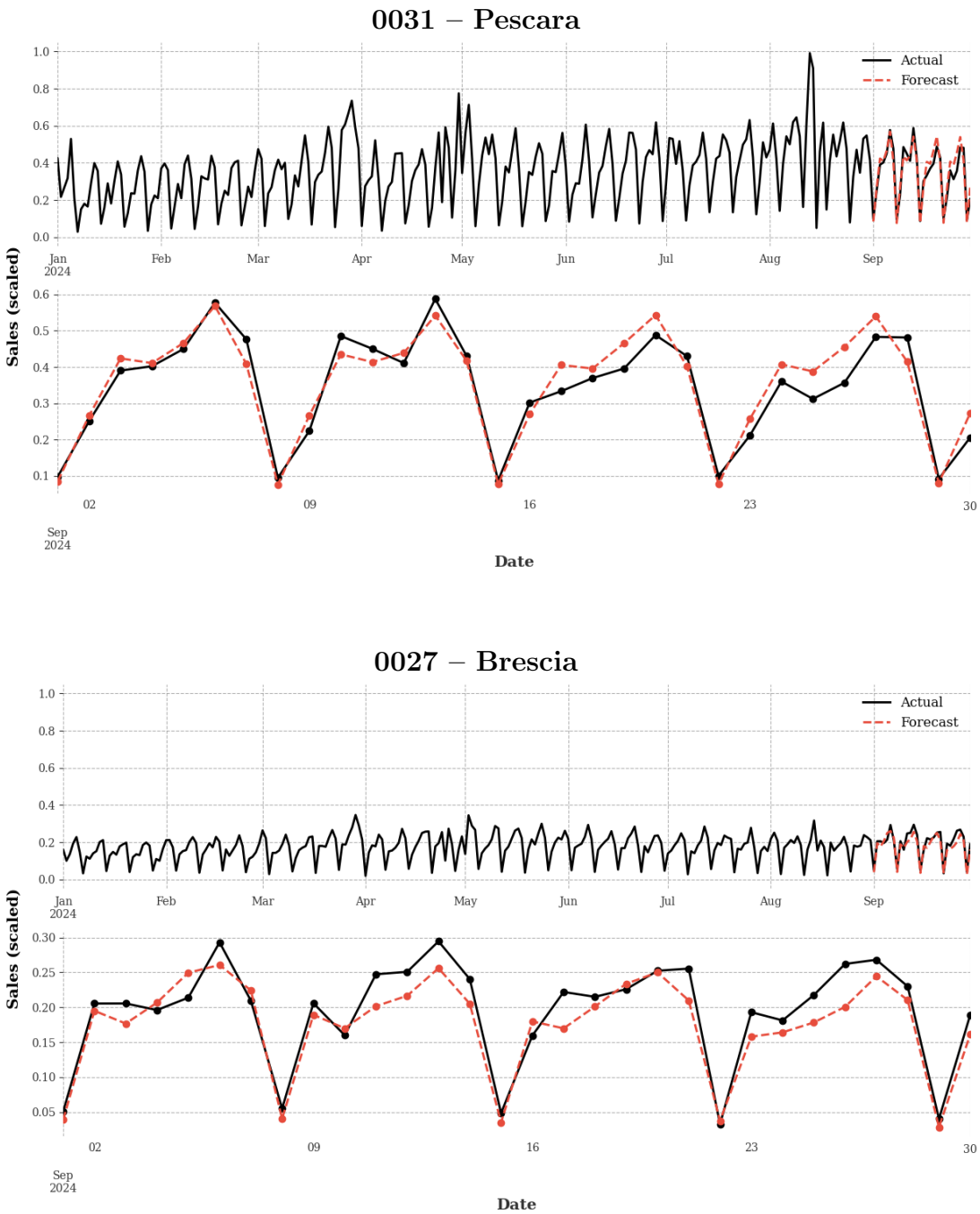


Figure 4.9: LightGBM forecasts versus actual sales for the stores of San Donato, Firenze, Pescara, and Brescia.

For the **San Donato** store, LightGBM demonstrates a marked improvement in capturing day-to-day volatility relative to simpler models, aligning closely with observed fluctuations in most weeks. Nevertheless, the forecast still underestimates certain high-demand days, particularly toward the end of the month. This shortfall implies that, while LightGBM adapts effectively to seasonal cycles, further refinement could be beneficial for anticipating occasional sales spikes.

The **Firenze** forecast plot reveals a strong alignment between predicted and actual sales, with the model consistently tracking the characteristic weekly peaks and troughs. Minor deviations do occur during sudden upward shifts, but overall, LightGBM exhibits a reliable fit, suggesting that the store's sales dynamics likely driven by relatively stable local demand patterns are effectively modeled. This performance underscores LightGBM's capacity to handle moderate sales variations and capture cyclical patterns.

In **Pescara**, LightGBM offers a particularly close match to real sales, including a notable reduction in forecast errors during mid-month fluctuations. The model's predictions tend to stay within a tight margin of the actual values, reflecting an advanced ability to interpret store specific features. While some residual misalignment appears in the final week, the overall precision suggests that LightGBM successfully internalizes the short-term rhythmicity inherent to Pescara's sales.

LightGBM's performance in **Brescia** is characterized by steady forecast accuracy and consistent adherence to the weekly sales pattern. The model seldom deviates significantly, even during periods of increased or decreased demand, signifying a robust handling of the store's comparatively lower sales volumes and milder volatility. Occasional underestimations are visible but remain limited in scope, highlighting LightGBM's relatively strong predictive capability.

Across all four stores, LightGBM delivers a notable enhancement in forecast quality compared to more basic approaches, capturing complex, store-specific demand variations with reduced error. Yet, sporadic misjudgments in predicting peak values and abrupt demand changes remain an area for potential improvement. Overall, the LightGBM model strikes a promising balance between computational feasibility and predictive robustness, suggesting its suitability for a wide range of operational forecasting needs within the Cash & Carry channel.

Local Metrics

Store	MAE	RMSE	Relative MAE
0017 – San Donato	0.971	0.931	15.39%
0016 – Firenze	0.243	0.228	7.68%
0031 – Pescara	0.096	0.088	9.12%
0027 – Brescia	0.238	0.233	10.43%

Table 4.3: Error metrics (MAE, RMSE, and Relative MAE) evaluating the predictive accuracy of the LightGBM model across the four analyzed stores.

Examining the LightGBM metrics in Table 4.3 reveals that this model consistently improves upon the simpler linear regression baseline, yet its comparison with the previously discussed Random Forest outcomes is more nuanced.
In **San Donato**, for example, LightGBM achieves lower absolute errors (MAE 0.971, RMSE 0.931), but the relative MAE (15.39%) remains slightly higher than that of Random Forest (14.98%). This result suggests that, while LightGBM effectively reduces day-to-day deviations, the store's larger average sales still cause the proportional error to remain elevated during certain high-demand periods.
Firenze shows a substantial reduction in error compared to linear regression, underscoring LightGBM's ability to capture local sales patterns more effectively. However, it does not fully match the lower error levels attained by Random Forest, indicating that some store-specific factors in Firenze might still be modeled more precisely by the ensemble of decision trees.
In contrast, **Pescara** exhibits the most pronounced accuracy gains, with LightGBM outperforming both linear regression and Random Forest in terms of daily forecasts, suggesting a strong fit to that store's sales behaviors and a notable reduction in overall residual errors.
Brescia likewise benefits from a drop in absolute errors relative to linear regression, although its relative MAE (10.43%) remains marginally above the figure for Random Forest (10.06%), implying that a slightly higher proportion of its forecast discrepancies remain unaddressed by LightGBM.

Overall, these findings confirm that LightGBM surpasses the baseline model in capturing complex, store-specific sales variations, yet it does not consistently outperform Random Forest in every location. The store-level differences highlight the importance of local demand characteristics and signal that further refinements or targeted tuning may be needed to ensure consistently optimal forecasts across all sites.

Global Metrics

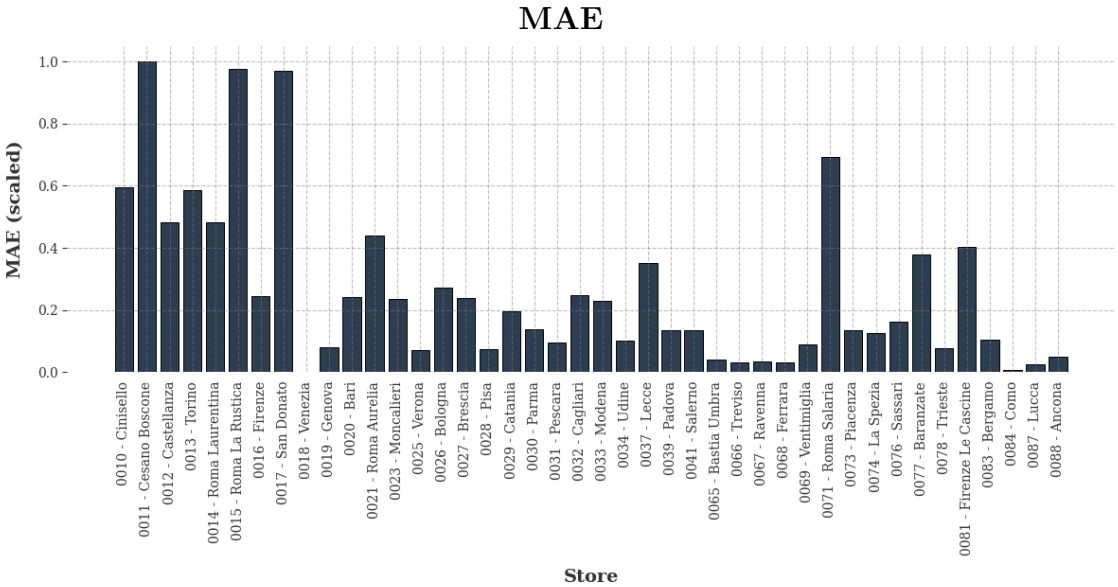


Figure 4.10: Mean Absolute Error (MAE) per store, highlighting the absolute differences in forecasting performance of the LightGBM model.

This bar chart illustrates the Mean Absolute Error (MAE) distribution for the LightGBM model across all Metro Italia Cash & Carry stores, with a **global MAE of 0.2618**. From a broad perspective, many locations display moderate MAE values, indicating that the model effectively aligns predicted and actual daily sales for a substantial portion of the network. Nonetheless, some stores, including San Donato and Roma Aurelia, stand out with higher error levels, suggesting heightened complexity in their demand patterns that the current configuration of LightGBM has yet to capture fully. In general, the overall distribution of errors is tighter than what was observed with simpler methods, reflecting the model's advanced capacity to learn nuanced, store-specific dynamics. The global MAE averaged across all stores remains at a moderate level, implying that although LightGBM provides a robust baseline of predictive accuracy, there is room for additional refinement or tuning to manage the peak deviations that persist in certain high impact stores.

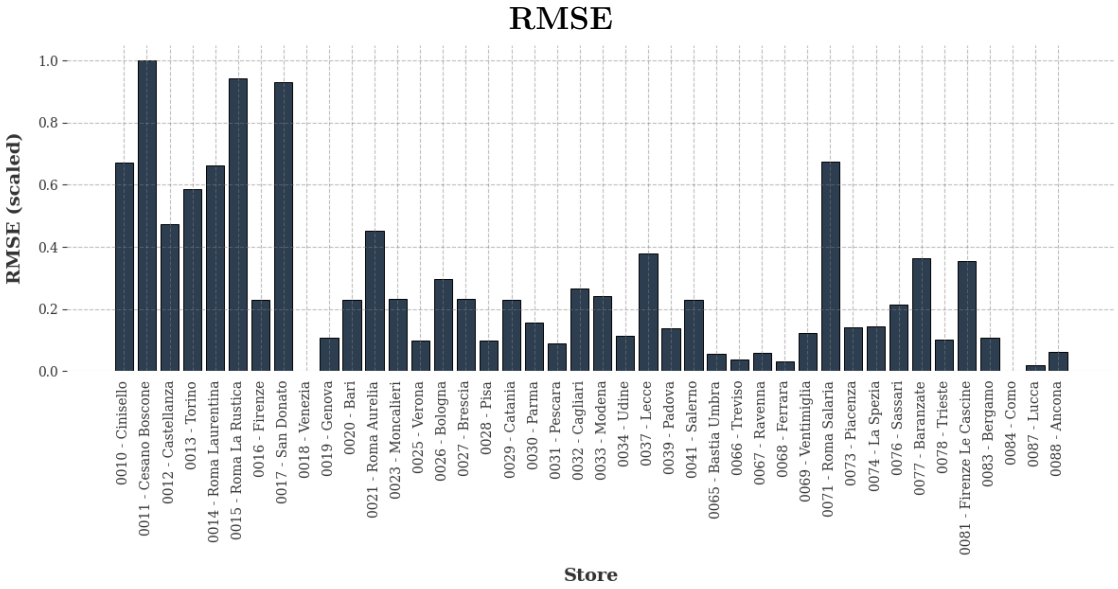


Figure 4.11: Root Mean Square Error (RMSE) per store, highlighting the absolute differences in forecasting performance of the LightGBM model.

The above bar chart shows the Root Mean Squared Error (RMSE) distribution for the LightGBM model across all Metro Italia Cash & Carry stores, yielding a global average **RMSE of 0.2752**. As with the MAE results, a subset of stores, particularly those exhibiting pronounced sales volatility, such as San Donato or Roma Salaria, maintain higher RMSE values, indicating the presence of certain atypical fluctuations that LightGBM still struggles to predict precisely. Nevertheless, the majority of stores display relatively contained RMSE levels, reflecting LightGBM's competency in moderating the impact of peak errors.

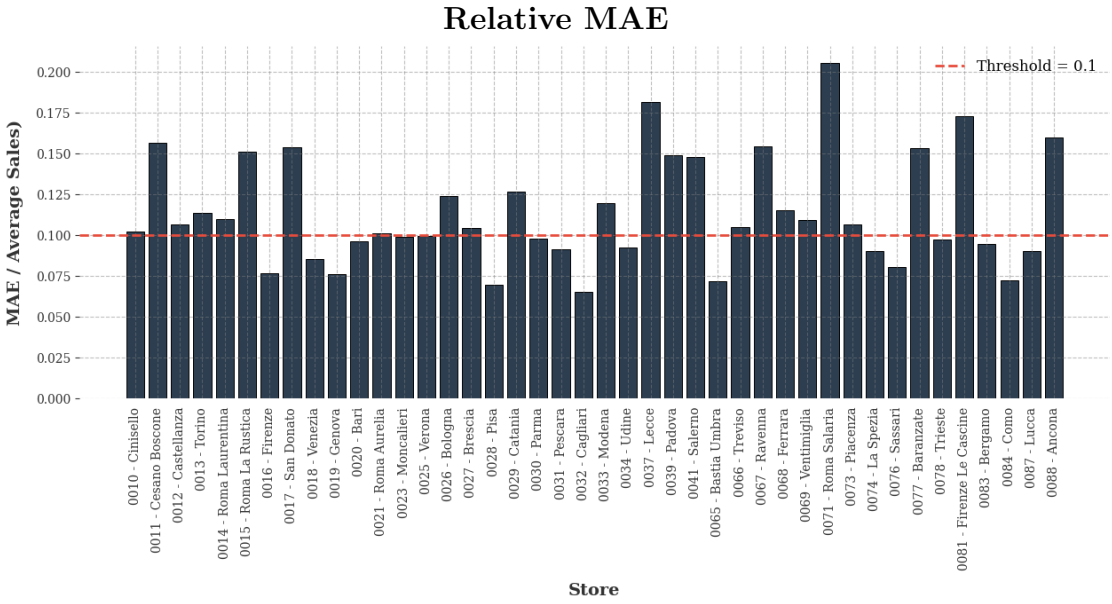
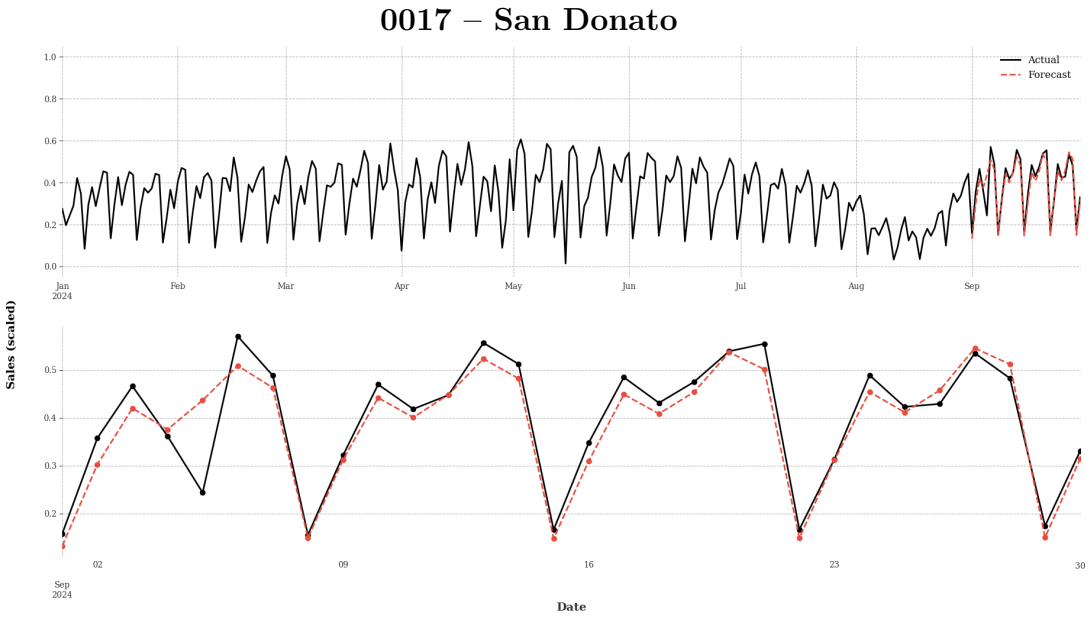


Figure 4.12: Relative MAE per store compared against the company's optimal threshold (10%), providing context on the significance of prediction errors relative to typical sales volumes at each store.

From the Relative MAE bar graph, we look at the percentage of forecast error relative to average daily sales for each store, revealing an **overall average of 11.37%** for LightGBM. This metric highlights how seemingly moderate absolute deviations can translate into substantial operational impact for some high-volume or volatile stores. While many locations appear near or below the company's nominal 10 percent threshold, suggesting forecasts that closely track daily demand, others exceed that benchmark, particularly those with intricate local patterns or larger sales swings.

LSTM

The plots provided showcase LSTM predictions for four representative Cash & Carry stores.



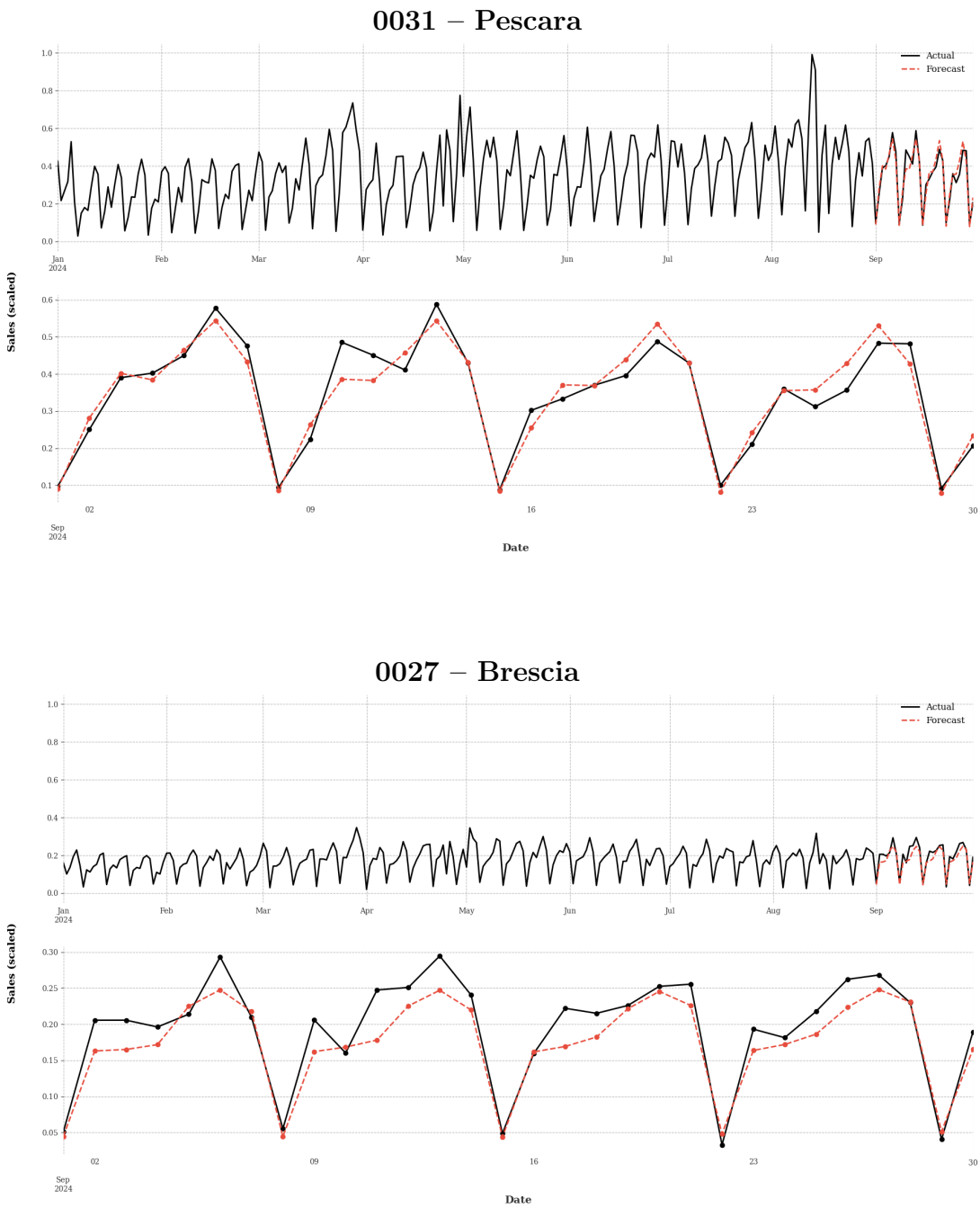


Figure 4.13: LSTM forecasts versus actual sales for the stores of San Donato, Firenze, Pescara, and Brescia.

In the case of **San Donato**, the LSTM model captures daily fluctuations with notable precision, consistently echoing the magnitude and timing of both routine and sporadic demand shifts. Although some slight underestimations emerge around peak periods, the forecast trajectory closely follows actual sales compared to previous methods, suggesting that the LSTM's recurrent architecture effectively learns the store's cyclical tendencies.

For **Firenze**, the LSTM predictions remain in strong agreement with observed values across the entire 30-day window, with minimal deviations even on days showing abrupt demand changes. This alignment underscores the model's capacity to assimilate both short range variations and underlying weekly cycles that characterize the store's sales, thus improving upon the residual errors still observed in other approaches.

In **Pescara**, LSTM forecasts provide a highly consistent track of the real sales data, maintaining particularly close day-to-day alignment. Notably, the model demonstrates resilience in handling transient fluctuations around mid-month peaks. This performance indicates that the memory mechanisms in the LSTM architecture are capturing relevant temporal dependencies, translating into reduced forecast errors over the test interval.

Finally, **Brescia** presents forecasts that mirror observed sales with near-seamless fidelity, as the model adapts to both moderate and high-volume days. The slight discrepancies around pronounced troughs suggest that although the LSTM manages swift changes effectively, certain sharp drops could be refined further. Overall, the forecast's stability throughout the month points to robust internal representations of Brescia's underlying demand structure.

Across all four stores, the LSTM model demonstrates enhanced predictive accuracy compared to earlier techniques. The recurrent architecture appears well suited for capturing the inherent seasonality and day-to-day variability of sales in the Cash & Carry channel, reducing both systematic biases and peak related errors. While opportunities remain for additional tuning particularly in stores prone to rapid or unexpected shifts the LSTM forecasts signify a significant stride forward in adapting a more sequence-aware model to the complexities of retail demand.

Local Metrics

Store	MAE	RMSE	Relative MAE
0017 – San Donato	0.554	0.736	6.84%
0016 – Firenze	0.222	0.249	5.30%
0031 – Pescara	0.114	0.124	7.35%
0027 – Brescia	0.374	0.392	10.20%

Table 4.4: Error metrics (MAE, RMSE, and Relative MAE) evaluating the predictive accuracy of the LSTM model across the four analyzed stores.

Examining the LSTM metrics in Table 4.4 reveals a considerable reduction in both absolute and proportional errors for all four stores, underscoring the network's capacity to capture intricate temporal dependencies.
In **San Donato**, for instance, the MAE of 0.554 and RMSE of 0.736 are paired with a notably low Relative MAE of 6.84%, an improvement that indicates enhanced daily tracking of sales peaks and troughs.
Similarly, **Firenze** records a MAE of 0.222, RMSE of 0.249, and Relative MAE of 5.30%, highlighting the model's ability to adapt to the store's inherent cyclicity and moderate fluctuations.
Pescara benefits from near-minimal absolute errors (MAE 0.114, RMSE 0.124), although its Relative MAE of 7.35% confirms that certain day-to-day variations, while small in absolute terms, still present noticeable proportional discrepancies.
Brescia exhibits higher residual values overall (MAE 0.374, RMSE 0.392, Relative MAE 10.20%), yet these remain noticeably lower than the more elevated error ranges identified in less sequence-aware approaches.

Taken together, these results point to a distinct advantage in applying LSTM to the Cash & Carry channel's complex sales environment. The model consistently achieves relatively low MAE and RMSE, signaling fewer large-scale deviations, while also reducing Relative MAE to levels that closely align predictions with the operational context of each store. Although isolated forecasts occasionally deviate from real world observations, the broad reduction in errors suggests that LSTM's memory mechanisms and capacity for learning long-range dependencies contribute substantially to precise sales forecasting.

Global Metrics

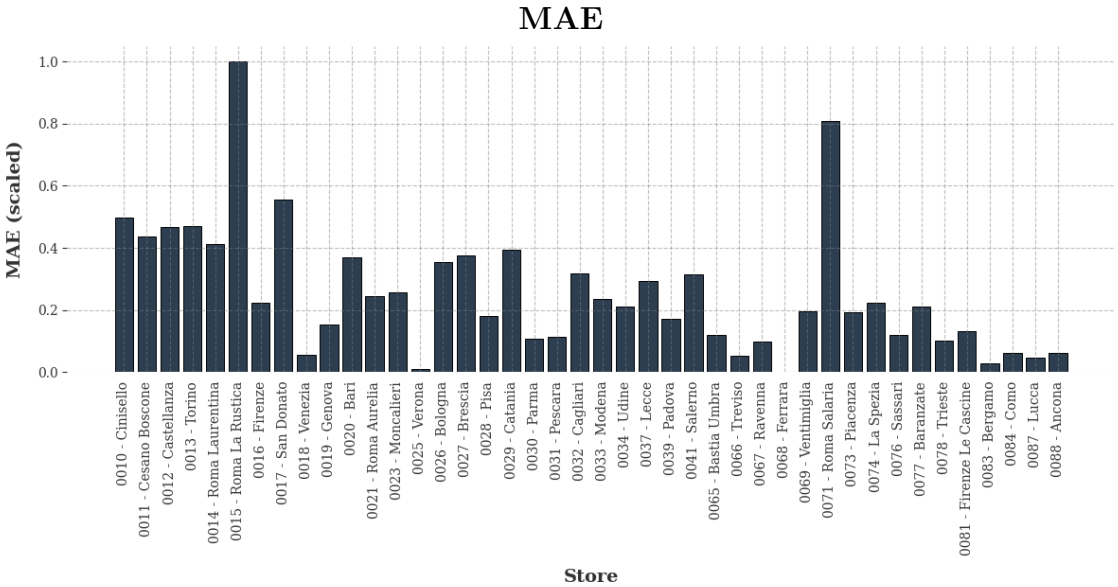


Figure 4.14: Mean Absolute Error (MAE) per store, highlighting the absolute differences in forecasting performance of the LSTM model.

This bar chart displays the Mean Absolute Error (MAE) for the LSTM model across each Metro Italia Cash & Carry store, indicating how closely this recurrent neural network framework matches daily forecasts to actual sales. Most stores exhibit mid to low error ranges, suggesting that the LSTM has effectively captured core weekly and seasonal demand patterns. Nonetheless, a few locations notably San Donato and Roma Salaria continue to show relatively higher MAE values, underscoring ongoing complexities in capturing their sales volatility.
When considering the global average **MAE of 0.2538**, the distribution points to a notable reduction in forecast errors relative to earlier methods, emphasizing the model's competence in learning complex temporal structures.

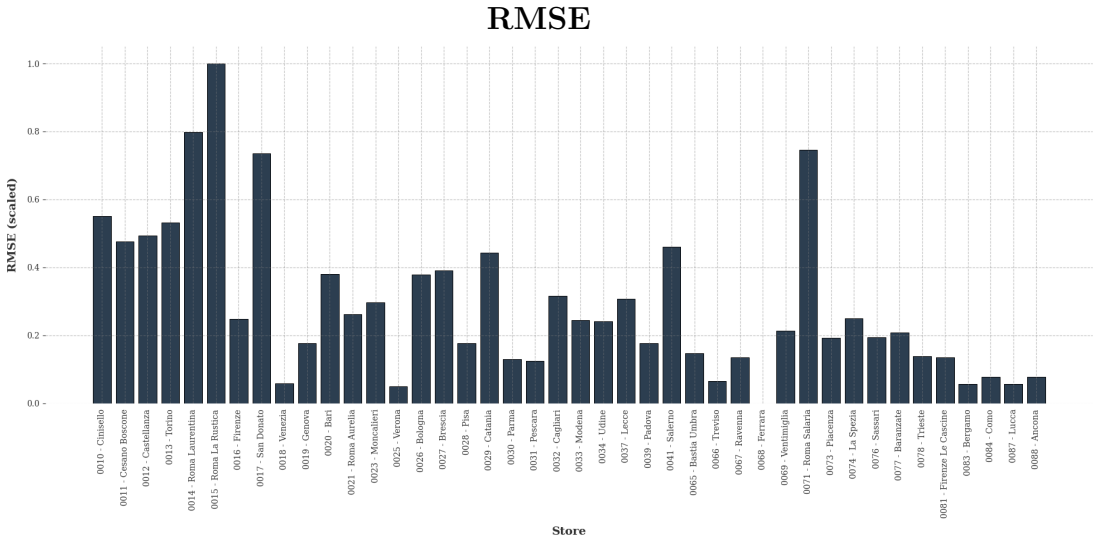


Figure 4.15: Root Mean Square Error (RMSE) per store, highlighting the absolute differences in forecasting performance of the LSTM model.

The bar plot above demonstrates the Root Mean Squared Error (RMSE) values for the LSTM model across all Metro Italia Cash & Carry locations, with a global average **RMSE of 0.2891**. This distribution aligns with the improvement seen in the MAE results, indicating that the LSTM model generally manages to minimize both moderate daily errors and larger, more infrequent deviations.

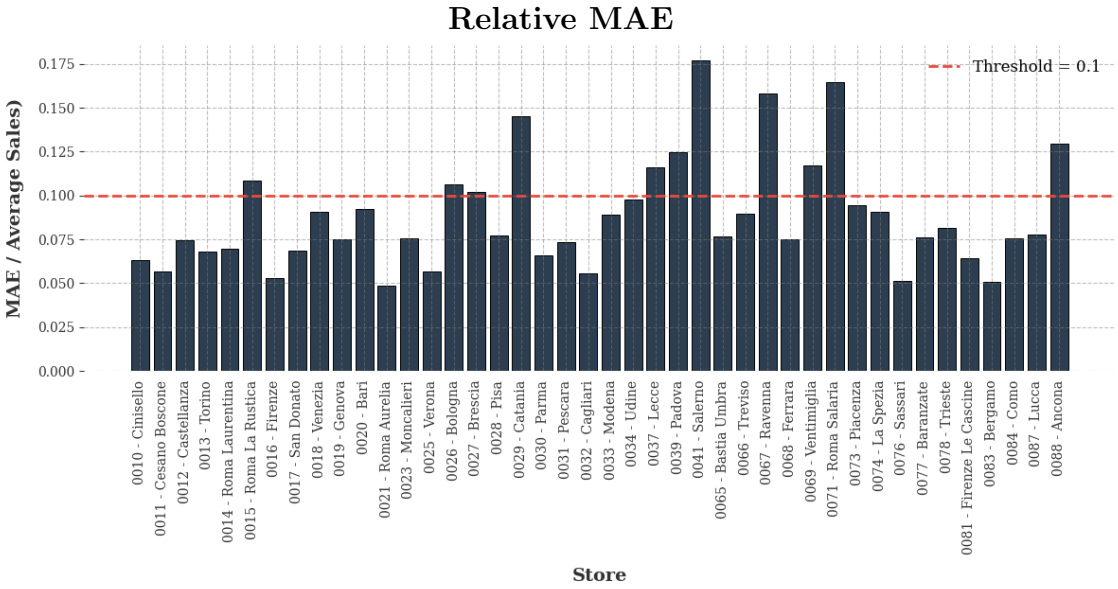


Figure 4.16: Relative MAE per store compared against the company's optimal threshold (10%), providing context on the significance of prediction errors relative to typical sales volumes at each store.

From this Relative MAE bar chart, normalized against each store's average sales, it is evident that the LSTM model achieves a global average of roughly **8.82%**, remaining below the commonly referenced 10% threshold for acceptable forecasting. Most stores cluster around or under this benchmark, indicating a strong capacity for capturing daily demand patterns across both high and low volume contexts. A few locations still appear above this limit, yet the chart underscores the model's overall success in reducing proportional errors compared to earlier approaches.

4.1.2 Optimal Model Selection

The final step in the evaluation process involves the selection of the most appropriate predictive model for the Cash & Carry sales channel. This decision is guided by a comparative analysis of the forecasting performance observed across the examined models, Linear Regression, Random Forest, LightGBM, and LSTM, using standard error metrics. The goal is to identify the model that consistently demonstrates a high level of accuracy and generalizability across a diverse set of stores.

Table 4.5: Final comparison of metrics between models

Model	**MAE**	**RMSE**	**Relative MAE**
Linear Regression (Baseline)	0.300	0.328	13.89%
Random Forest	0.259	0.298	11.06%
LightGBM	0.262	**0.275**	11.37%
LSTM	**0.254**	0.289	**8.82%**

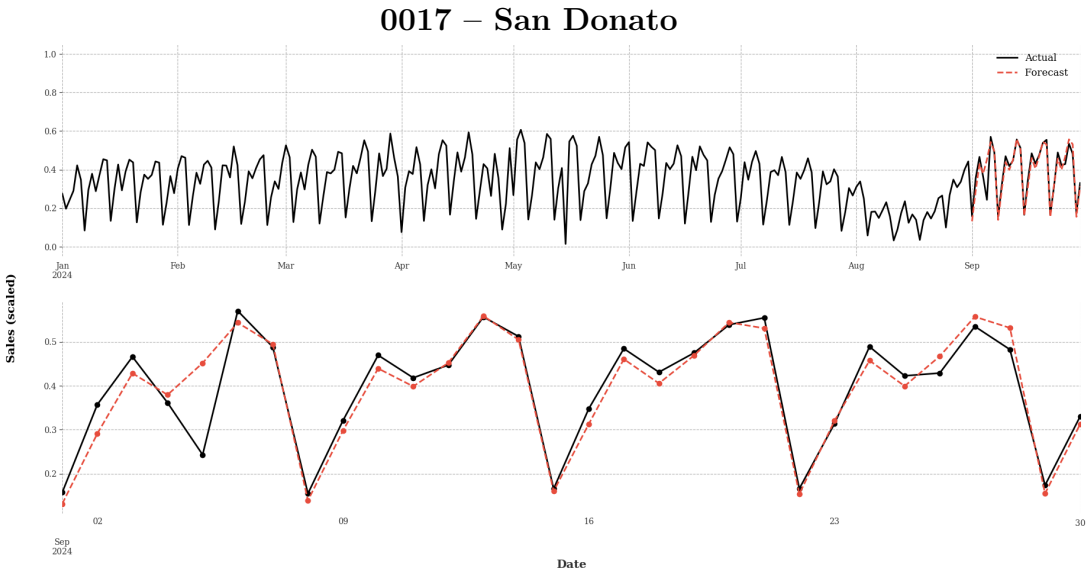
A comparative evaluation of the global performance metrics summarized in Table 4.5 facilitates a clear distinction among the four forecasting approaches. The Linear Regression (Baseline), with a Mean Absolute Error (MAE) of 0.300 and a Relative MAE of 13.89%, establishes a foundational reference point but displays evident limitations when modeling more complex sales patterns. Both Random Forest and LightGBM present notable improvements over the baseline, achieving MAE values of 0.259 and 0.262 respectively, and lowering their Relative MAE into the 11% range, indicating a stronger capacity for capturing store specific fluctuations.
Nevertheless, it is the LSTM model that demonstrates the most pronounced gains, posting the lowest MAE at 0.254 and a Relative MAE of 8.82%. By normalizing the forecast errors against average store sales, the Relative MAE provides a critical lens for gauging the real world operational impact of prediction inaccuracies. The fact that LSTM consistently undercuts the 10% threshold suggests that it handles both moderate daily shifts and larger demand variations with an enhanced level of precision.

Although LightGBM achieves a slightly lower RMSE (0.275) than LSTM's 0.289, the overall alignment between predicted and actual sales appears to be strongest under LSTM's recurrent architecture. This advantage likely stems from the model's ability to learn extended temporal dependencies, allowing it to adapt to cyclical behaviors and store-level idiosyncrasies more effectively than tree-based methods. The aggregated results from all metrics, and their consistency with earlier store level analyses, underscore LSTM's aptitude for reconciling short-term volatility with long range demand trends.
Taken together, **these findings confirm that the LSTM model yields superior accuracy relative to the baseline and the other advanced techniques considered, making it the most suitable choice for Metro Italia's Cash & Carry channel**. Its ability to minimize errors across diverse store environments aligns well with the operational requirements of inventory management and sales planning, thereby indicating a robust path forward for large scale retail forecasting initiatives.

4.1.3 Backtesting the Winning Model with Historical Forecast

Having identified the LSTM model as the most effective forecasting approach for the Cash & Carry channel, the final step involved simulating its real world deployment through a rolling backtest.
This procedure, facilitated by Darts's *historical_forecast()* utility, emulates the incremental manner in which new data would become available in an operational setting. Specifically, the model was trained on a progressively expanding window of historical sales and then asked to generate short horizon (daily) forecasts, which were subsequently compared against actual observations for each newly revealed time step.

This rolling mechanism provides two principal insights. First, it verifies that the LSTM model maintains its predictive robustness as fresh data is introduced, rather than relying solely on a static train/test partition. Second, it highlights the model's adaptability in capturing evolving sales patterns such as midmonth demand surges or unanticipated dips across different stores in near real time. The Historical Forecast thus serves as a stress test, ensuring that the chosen LSTM architecture can generalize its learned temporal dependencies beyond a single fixed dataset slice.


0017 – San Donato
Actual
Forecast
Jan 2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep 2024
02
09
16
23
30
Sales (scaled)
Date

0016 – Firenze
Actual
Forecast
Jan 2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep 2024
02
09
16
23
30
Sales (scaled)
Date

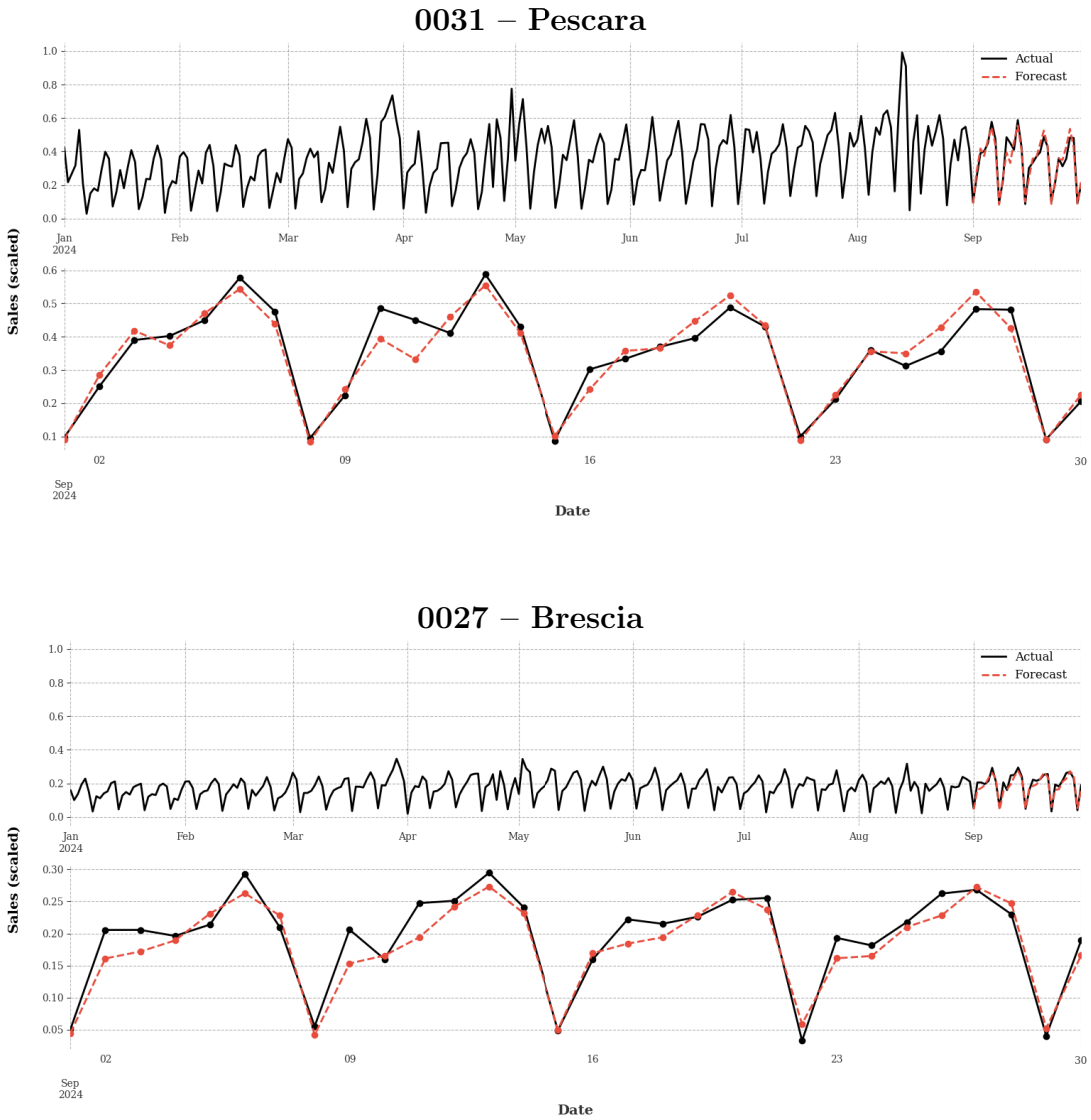


Figure 4.17: Backtest results for the stores of San Donato, Firenze, Pescara, and Brescia.

In **San Donato**, the LSTM model continues to capture both moderate and pronounced fluctuations in sales, with only sporadic underestimations at specific peaks. The day-to-day alignment observed during the rolling period aligns well with the previously noted accuracy under static validation, indicating that the model retains its reliability even as fresh data are introduced.

For **Firenze**, the LSTM forecasts remain highly consistent, closely tracking the store's typical weekly peaks and troughs. The backtest confirms that the model is not merely memorizing patterns from a fixed dataset but rather adapting its internal representation of the store's evolving sales trajectory over time.

In **Pescara**, the LSTM's performance stands out for its tight alignment with observed sales, replicating the short-term dynamics evident in the test window. The rolling approach demonstrates minimal drift between predictions and actual values, suggesting robust internalization of local demand signals, even when sales exhibit sudden mid-month surges.

Finally, **Brescia**'s rolling forecasts reaffirm the model's capacity to handle stores with moderate sales levels and cyclical variations. While occasional slight deviations remain, the LSTM consistently recovers in subsequent days, maintaining a stable forecast trajectory that aligns with actual sales fluctuations over the entire backtest window.

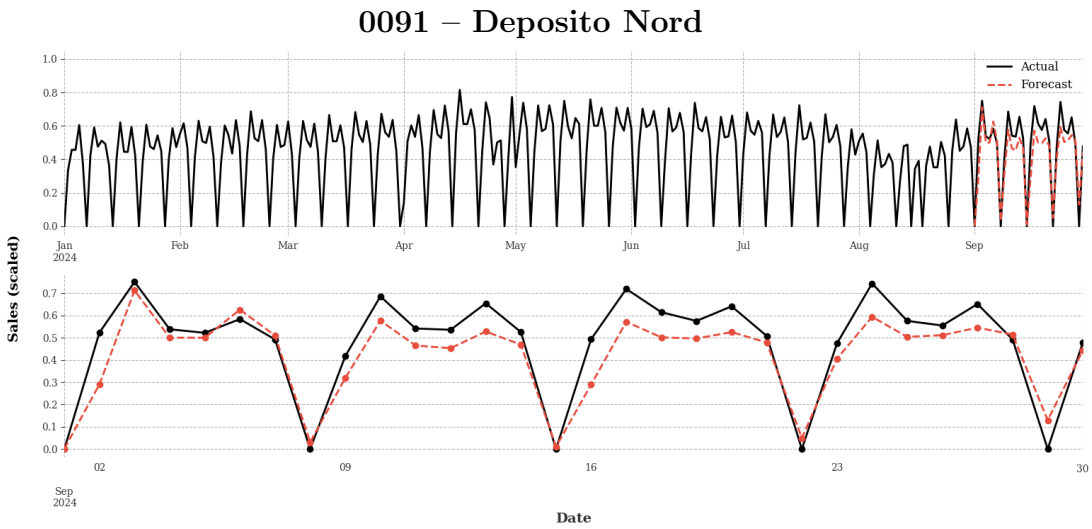
The definitive backtest results for the LSTM model yielding a Mean Absolute Error (MAE) of 0.330, a Root Mean Squared Error (RMSE) of 0.356, and a Relative MAE of 8.19% reinforce the model's capacity to deliver robust, day-to-day forecasts in a dynamic retail setting. Not only does the rolling procedure confirm the LSTM's aptitude for maintaining consistent accuracy beyond a single train/test split, but it also substantiates the notion that sequence-aware architectures can effectively adapt to the evolving nature of store level sales. By sustaining low error margins under iterative data updates, the LSTM model has demonstrated the reliability and resilience necessary for real world deployment in the Cash & Carry channel, ultimately bolstering confidence in its suitability for operational forecasting tasks.

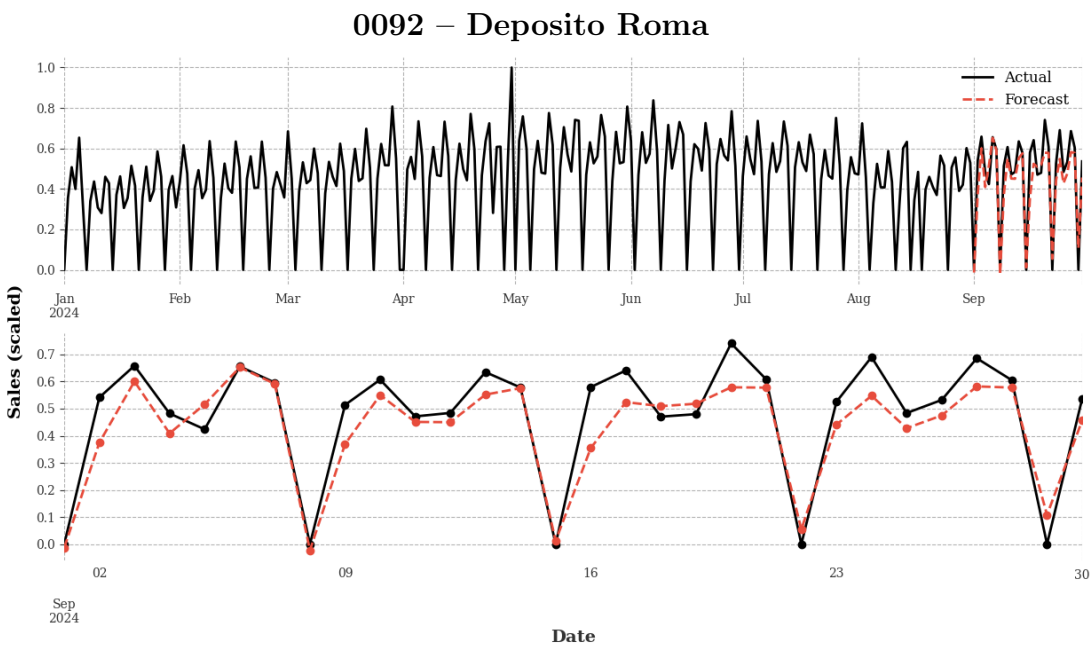
4.2 Full Service Delivery Channel Results

In this section, the forecasting results for the Full Service Delivery (FSD) channel are examined, extending the same methodological framework applied to the Cash & Carry analysis. As previously established, four key locations are considered: Deposito Nord, Deposito Roma, Pescara, and Brescia. Among these, the Deposito Nord and Roma hold particular importance, as they account for the majority of delivery operations and thus represent a substantial share of overall FSD sales. By comparing the forecasting performance across these deposit sites and two additional stores, the study aims to capture the broader variability in delivery-driven sales, ultimately evaluating each model's adaptability to the unique dynamics of FSD demand patterns.

4.2.1 Comparative Analysis of Forecasting Models

Linear Regression (Baseline)





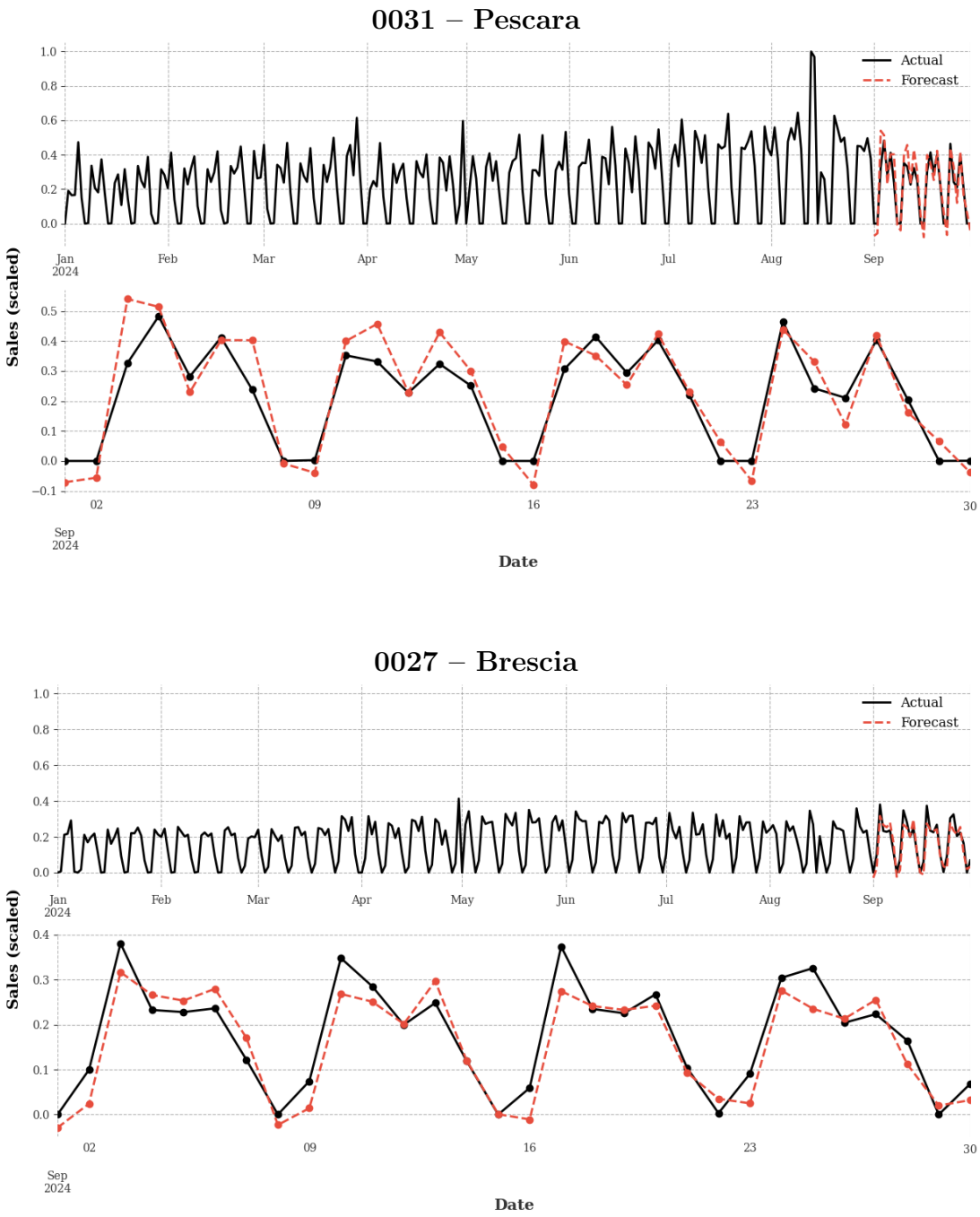


Figure 4.18: Linear Regression forecasts versus actual sales for the stores of Deposito Nord, Deposito Roma, Pescara, and Brescia.

In **Deposito Nord**, the model exhibits stable predictive fidelity throughout most of the forecast horizon, effectively mirroring the weekly cyclical behavior inherent to FSD operations. Minor underestimations become more evident during certain high demand intervals near the end of the test period, implying that while the model assimilates general seasonal trends, it may underrepresent the magnitude of sporadic peak demand events.

Forecasts for **Deposito Roma** remain closely attuned to actual sales volumes, particularly across mid-month intervals prone to temporary spikes. The model's ability to capture moderate and abrupt sales fluctuations suggests a robust internalization of the store's demand profile.

In **Pescara**, the model manages to replicate a comparatively volatile demand pattern, though instances of overestimation on certain days and underestimation on others do arise. Despite these oscillations, the general trajectory aligns closely with real sales, indicating that the model effectively captures daily cyclical cues. The consistent approximation of short-range peaks and troughs highlights the model's proficiency in adapting to transient fluctuations that characterize Pescara's FSD dynamics.

Brescia's forecasts reflect a balanced capacity to replicate the store's moderate sales profile, although minor deviations are visible near higher demand periods. While the model addresses most routine fluctuations adequately, these residual inaccuracies point to remaining complexities that might benefit from further calibration.

Across these four FSD locations, the model demonstrates a commendable ability to track the key weekly patterns and moderate sales surges typical of delivery-based operations. Variations in forecast precision appear primarily tied to store-specific factors, such as abrupt peaks or uncharacteristic demand patterns. Despite minor deviations at certain intervals, these results underscore the model's enhanced adaptability compared to simpler methods, reinforcing the importance of employing advanced forecasting architectures in FSD environments, where day-to-day demand can shift rapidly and store level idiosyncrasies exert a pronounced influence on operational decisions.

Local Metrics

Store	**MAE**	**RMSE**	**Relative MAE**
0092 – Deposito Nord	1.000	1.000	16.43%
0091 – Deposito Roma	0.426	0.440	14.84%
0031 – Pescara	0.082	0.080	22.09%
0027 – Brescia	0.023	0.023	28.65%

Table 4.6: Error metrics (MAE, RMSE, and Relative MAE) evaluating the predictive accuracy of the Linear Regression model across the four analyzed stores.

The values in Table 4.6 highlight how the linear regression model performs differently across the four selected stores. **Deposito Nord and Deposito Roma** present relatively higher MAE and RMSE, both standardized at 1.000 for Deposito Nord and around 0.426–0.440 for Deposito Roma, yet their Relative MAE remains lower than in Pescara and Brescia. This outcome suggests that, despite incurring larger absolute deviations, the model's errors for these deposit locations are less severe in proportion to their higher average sales volumes.

Conversely, **Pescara and Brescia** exhibit notably lower MAE and RMSE values indicating smaller absolute forecast discrepancies yet their Relative MAE rises to 22.09% for Pescara and 28.65% for Brescia. These elevated percentages imply that, although the absolute errors are minimal, they still represent a substantial fraction of these stores' lower average daily sales. In other words, a modest numerical miss in these environments can exert a significant operational impact, given their smaller overall sales figures.

Taken together, these metrics underscore the importance of interpreting both absolute and relative errors in evaluating forecasting effectiveness. While stores with higher absolute deviations may still achieve acceptable proportional accuracy, others with comparatively small absolute errors can see a high Relative MAE, signaling that even minor numerical discrepancies can be operationally significant. This pattern reaffirms the need for a balanced perspective on performance metrics particularly in the FSD channel, where volumes and demand patterns vary markedly from one store to another.

Global Metrics

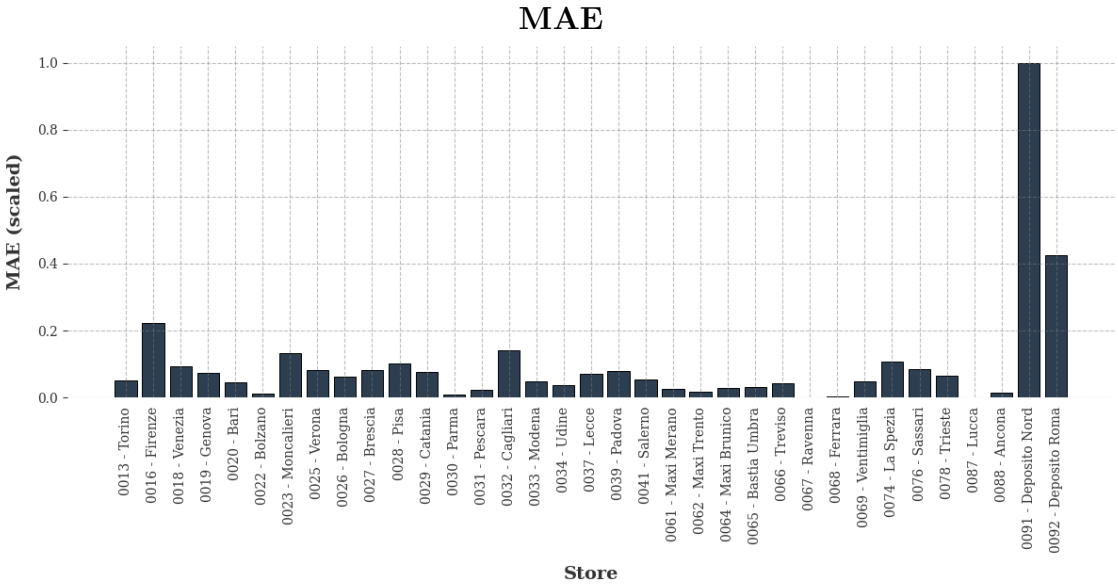


Figure 4.19: Mean Absolute Error (MAE) per store, highlighting the absolute differences in forecasting performance of the Linear Regression model.

This bar chart presents the Mean Absolute Error (MAE) distribution for the linear regression model across the Full Service Delivery (FSD) stores. The most prominent observation lies in the notably higher MAE values at Deposito Nord and Deposito Roma, reflecting the model's relative difficulty in capturing the more extensive or fluctuating demand profiles characteristic of these major delivery hubs. In contrast, many smaller or mid-volume FSD locations display lower MAE levels, indicating that the linear approach manages moderate daily variations more effectively where absolute sales volumes are less pronounced.
From a technical standpoint, the elevated MAE in the deposit locations suggests that linear regression struggles to align predictions with the stores' dynamic demand surges, potentially due to unmodeled factors or the simplicity of the linear hypothesis. Despite these outliers, the overall error distribution for the remaining FSD network indicates that the model can adequately forecast environments featuring stable or moderately scaled sales patterns. Nonetheless, the consistently high MAE at critical delivery nodes highlights the limitations of a purely linear structure when confronted with high volume, high volatility demand scenarios.

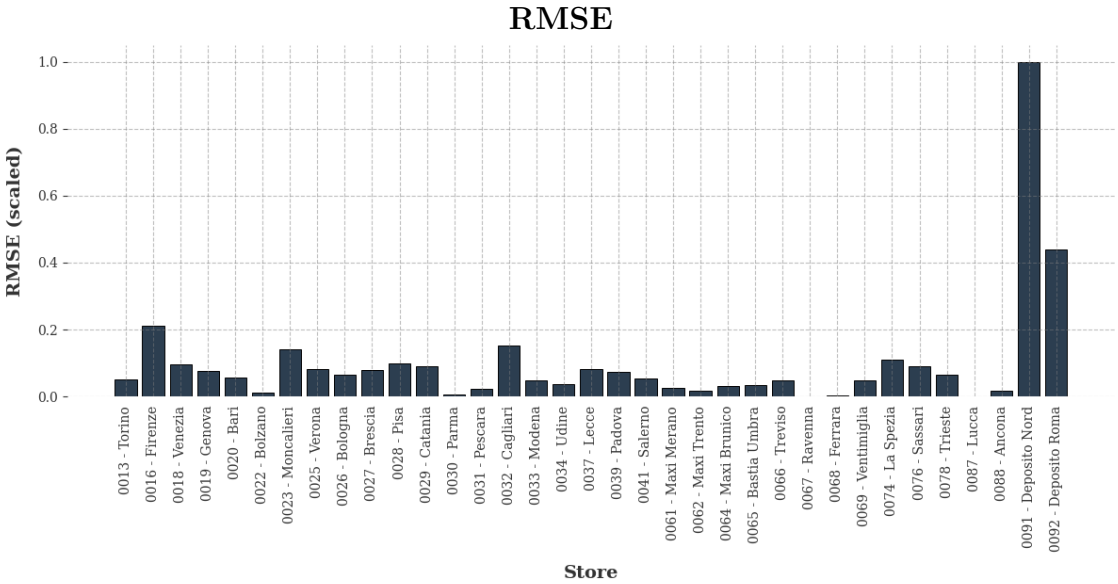


Figure 4.20: Root Mean Squared Error (RMSE) per store, highlighting the absolute differences in forecasting performance of the Linear Regression model.

The Root Mean Squared Error (RMSE) distribution closely mirrors the insights derived from the MAE analysis. Elevated RMSE values again concentrate at the primary delivery nodes Deposito Nord and Deposito Roma underscoring the linear regression model's persistent difficulty in capturing the high volume, more volatile sales profiles characteristic of these locations.
This consistency between MAE and RMSE further suggests that the model's forecasting errors are not driven by sporadic outliers but rather by a systematic inability to align with the dynamic demand patterns present in major FSD hubs.

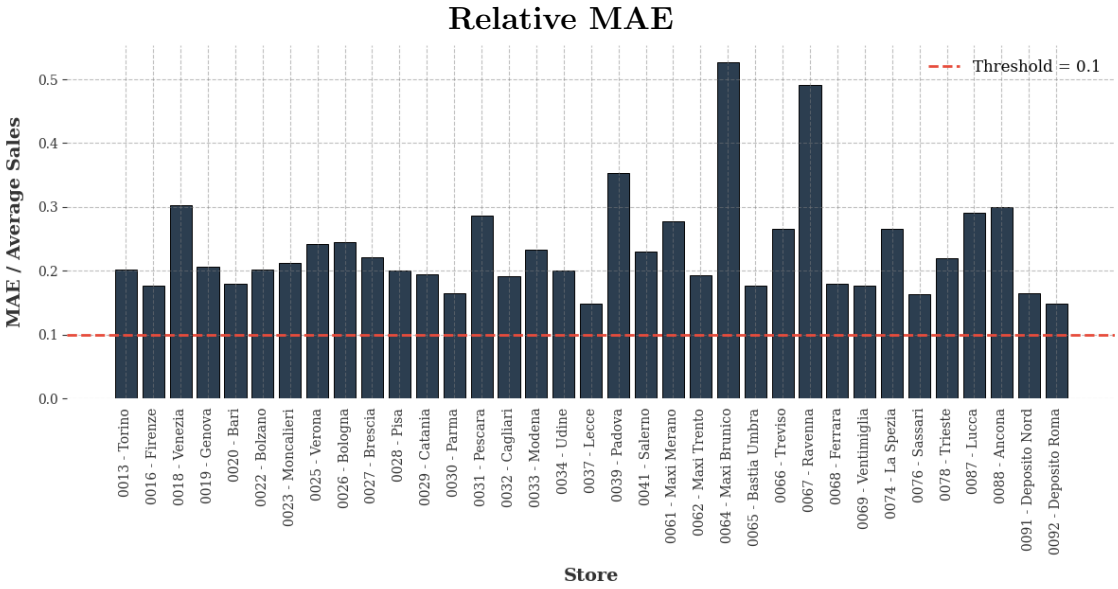
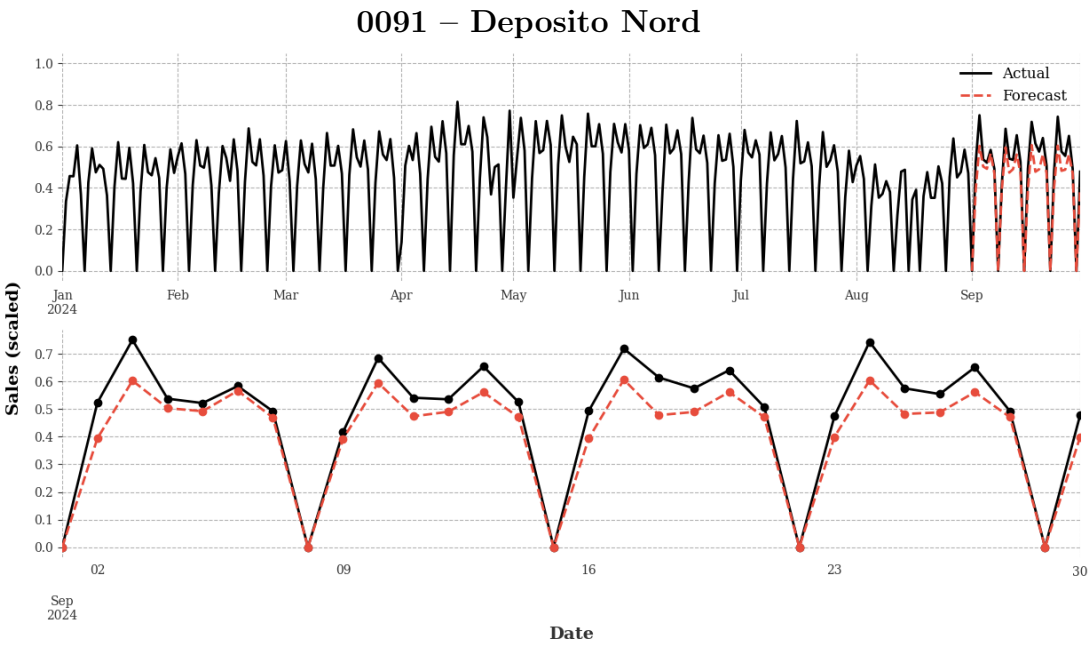


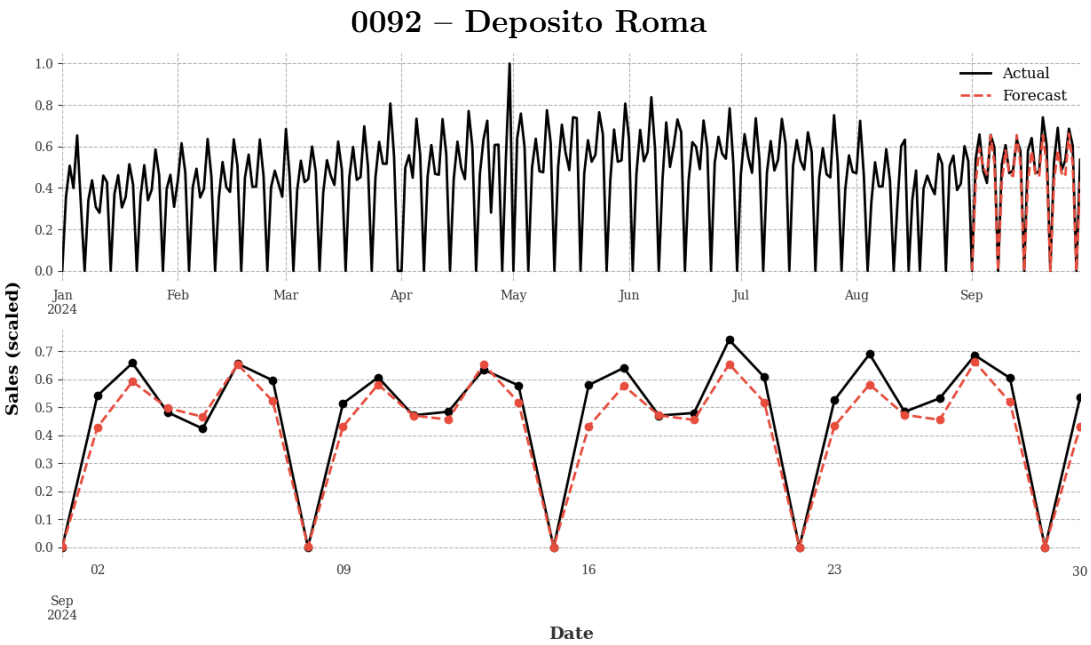
Figure 4.21: Relative MAE per store, highlighting the absolute differences in forecasting performance of the Linear Regression model.

This bar chart depicts the Relative MAE for each store in the Full Service Delivery (FSD) channel, normalizing forecast errors by each location's average daily sales. While the deposit hubs show notably high absolute errors (as previously observed in MAE and RMSE), their relative errors remain comparatively lower. This indicates that, although the model struggles to predict these large-volume sites in terms of raw discrepancy, those errors are less impactful once scaled to the store's overall sales magnitude.

By contrast, certain smaller or mid-volume stores exhibit higher relative values, revealing that even minor absolute misses can translate into a substantial fraction of their average daily demand. These results confirm that, despite the linear model's inability to capture every fluctuation in high volume hubs, the proportional effect of its inaccuracies is less pronounced in these deposit sites than in lower volume stores. Consequently, while the baseline approach retains noticeable limitations, these data underscore how relative metrics can yield a different interpretation of forecast quality, highlighting that absolute and proportional errors often diverge depending on store size and operational significance.

Random Forest





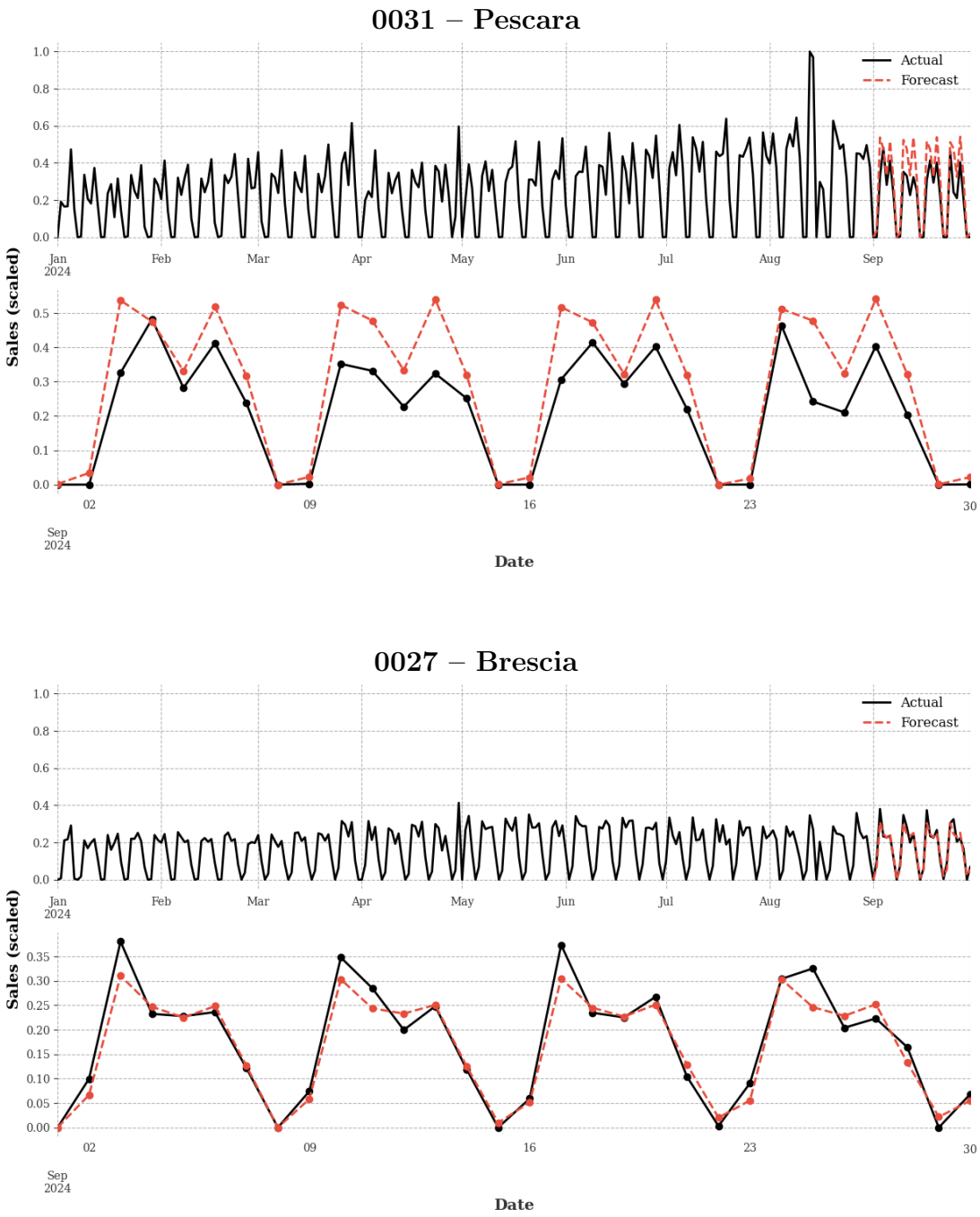


Figure 4.22: Random Forest forecasts versus actual sales for the stores of Deposito Nord, Deposito Roma, Pescara, and Brescia.

In **Deposito Nord**, the Random Forest model captures the recurring cyclical patterns characteristic of a high-volume delivery hub, maintaining generally close agreement with observed sales. Minor underestimations appear at select peak intervals, implying that although the ensemble of decision trees handles routine fluctuations well, certain abrupt surges remain challenging to forecast with full precision.

Forecasts at **Deposito Roma** likewise demonstrate consistent adherence to weekly rhythms, with relatively tight clustering around actual sales. Nonetheless, sporadic deviations, most visible during sudden demand shifts, indicate that while Random Forest provides a robust baseline for major delivery nodes, refining hyperparameters or incorporating additional store-specific signals could enhance accuracy on extreme demand days.

In **Pescara**, the model exhibits slightly higher variability, sometimes overshooting mid-month peaks and undershooting others. Despite these fluctuations, the overall trajectory remains largely aligned with actual sales volumes, underscoring the model's capacity to handle moderate daily variations. The tendency to overestimate certain intervals, however, suggests that Pescara's local demand dynamics may benefit from more granular feature engineering.

Finally, **Brescia** displays a balanced yet slightly conservative forecast profile. The model generally follows the store's cyclical sales pattern but occasionally lags behind abrupt increases, hinting that even in moderate-volume contexts, the decision-tree ensemble may underestimate sudden, short-lived surges.

Across these four FSD locations, the Random Forest model consistently tracks standard weekly demand cycles while encountering sporadic challenges in precisely forecasting sudden spikes. Although its ensemble-based nature moderates large swings in error compared to simpler linear methods, further refinement could address lingering gaps in capturing localized demand factors. Overall, these results reinforce the notion that Random Forest offers a solid foundation for FSD forecasting, yet it may still yield modest underestimations during periods of unanticipated, high-intensity demand.

Local Metrics

Store	MAE	RMSE	Relative MAE
0092 – Deposito Nord	1.000	1.000	13.08%
0091 – Deposito Roma	0.372	0.398	11.09%
0031 – Pescara	0.060	0.063	40.07%
0027 – Brescia	0.058	0.049	12.89%

Table 4.7: Error metrics (MAE, RMSE, and Relative MAE) evaluating the predictive accuracy of the Random Forest model across the four analyzed stores.

The metrics in Table 4.7 reveal that **Deposito Nord** continues to exhibit the highest absolute errors (MAE and RMSE both scaled to 1.000), mirroring the difficulties observed in capturing the high-volume, high-volatility sales pattern at this pivotal delivery hub.
By contrast, **Deposito Roma** records a lower MAE (0.372) and RMSE (0.398), alongside a moderate Relative MAE of 11.09%. Although these values remain elevated relative to simpler or smaller-volume locations, they represent a reduction in error magnitude compared to Deposito Nord, indicating that while linear regression still underperforms at deposit nodes, it encounters slightly fewer challenges at Roma than at Nord.
In **Pescara**, the absolute errors (MAE 0.060, RMSE 0.063) are minimal, suggesting that the model struggles far less in terms of raw forecast discrepancies. However, the Relative MAE of 40.07% highlights that, despite the low absolute error, the store's smaller sales volume renders even modest misses proportionally significant.
A similar, though less extreme, pattern emerges for **Brescia**, which reports very low absolute metrics (MAE 0.058, RMSE 0.049) but a relatively higher proportional error (12.89%). These outcomes align with the notion that linear regression may capture general daily patterns adequately in smaller-volume environments, yet even small absolute deviations can translate into substantial relative impacts.
The absolute vs. relative discrepancy underscores how store size and demand structure influence error perception: the deposit hubs exhibit large absolute misses yet moderate relative errors, while smaller stores show the inverse. Overall, these metrics confirm that linear regression remains a basic baseline model for the FSD channel, with limitations especially evident in complex or highly variable store contexts

Global Metrics

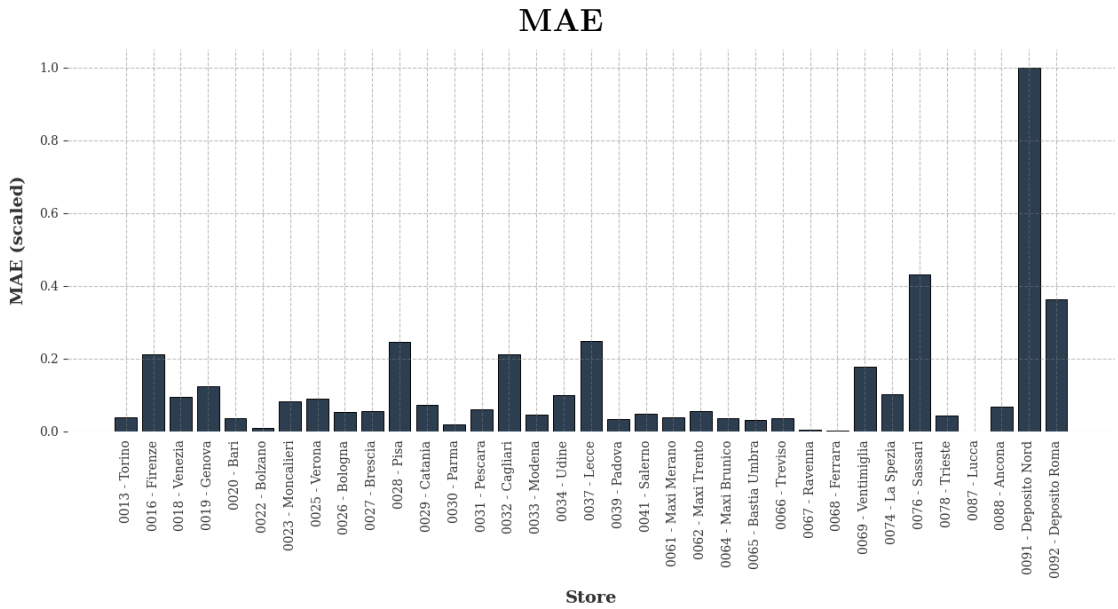


Figure 4.23: Mean Absolute Error (MAE) per store, highlighting the absolute differences in forecasting performance of the Random Forest.

The deposit hubs, Deposito Nord and Deposito Roma, again stand out with the highest absolute discrepancies, mirroring their more extensive, volatile demand profiles. By contrast, many smaller-volume locations register lower MAE values, indicating that linear regression handles moderate sales fluctuations more effectively.

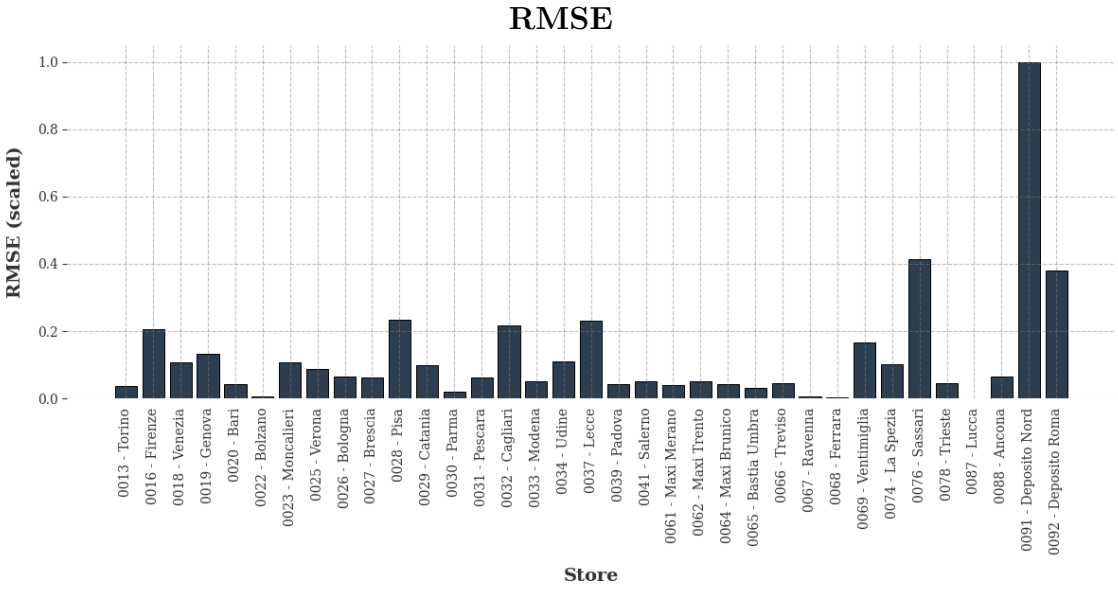


Figure 4.24: Root Mean Squared Error (RMSE) per store, highlighting the absolute differences in forecasting performance of the Random Forest.

The Root Mean Squared Error (RMSE) distribution for the linear regression model in the FSD channel again underscores higher residuals at Deposito Nord and Deposito Roma, consistent with their more volatile and larger scale sales dynamics. Smaller or more stable locations show reduced RMSE, reflecting fewer large deviations.

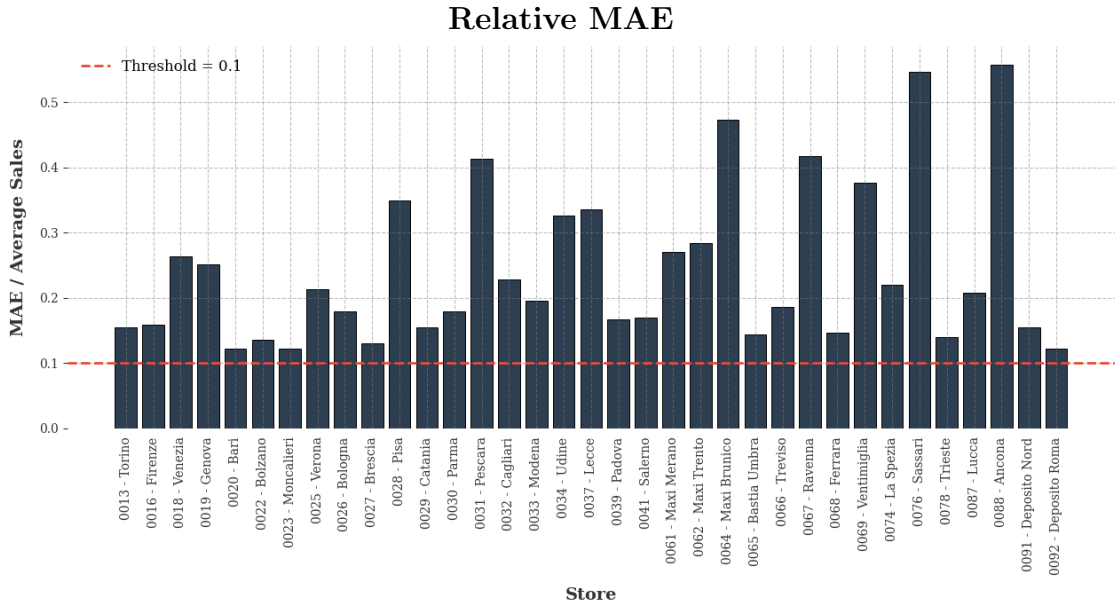
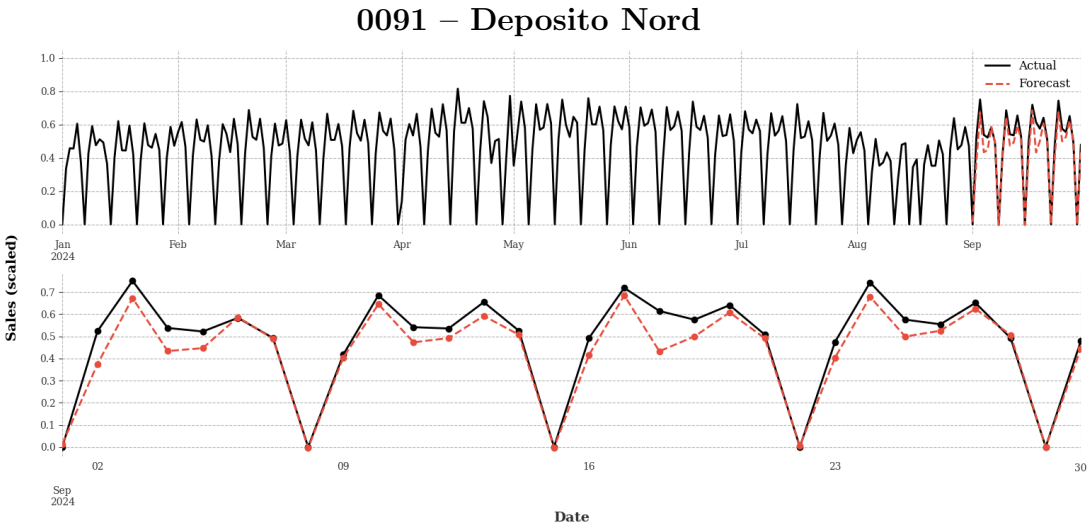
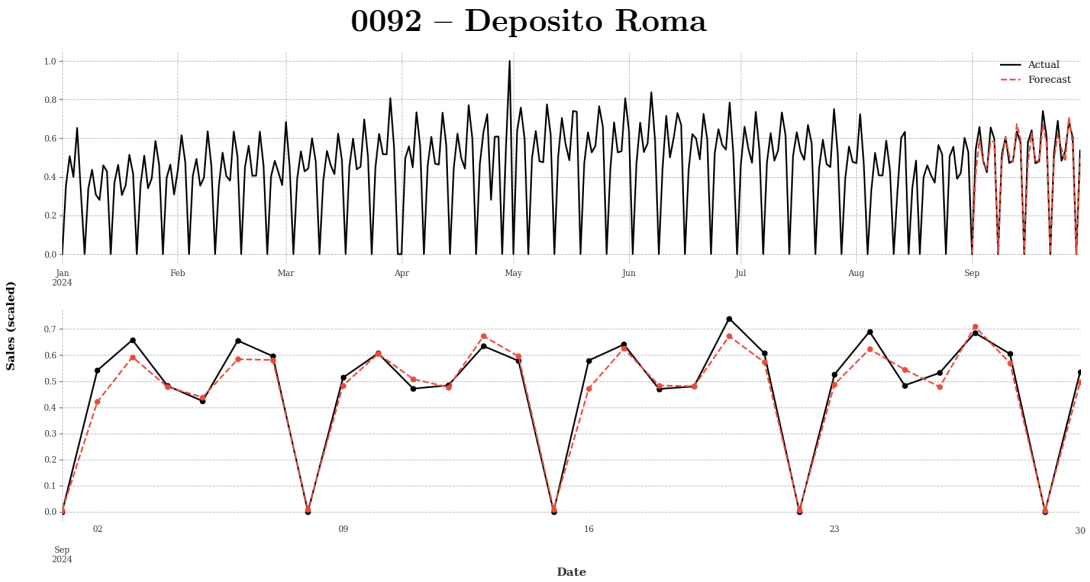


Figure 4.25: Relative MAE per store, highlighting the absolute differences in forecasting performance of the Random Forest.

This bar chart depicts the Relative MAE for each FSD store under the Random Forest approach, normalizing absolute errors by average daily sales. Compared to the baseline linear regression model, Random Forest generally reduces proportional forecast inaccuracies across the network. Notably, Deposito Nord and Deposito Roma, despite handling high-volume, more volatile demand, now register comparatively moderate relative errors, indicating that the ensemble method has better internalized the major cyclical patterns driving these primary delivery hubs. Conversely, some smaller-volume stores exhibit proportionally higher discrepancies, suggesting that even minor absolute misses can become operationally significant in locations where daily sales are limited. Overall, the distribution emphasizes Random Forest's improved balance in handling diverse store profiles, while still highlighting potential refinements for smaller outlets with more unpredictable sales fluctuations.

LightGBM


0091 – Deposito Nord
Actual
Forecast
Sales (scaled)
Jan 2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep 2024
02
09
16
23
30
Date


0092 – Deposito Roma
Actual
Forecast
Sales (scaled)
Jan 2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep 2024
02
09
16
23
30
Date

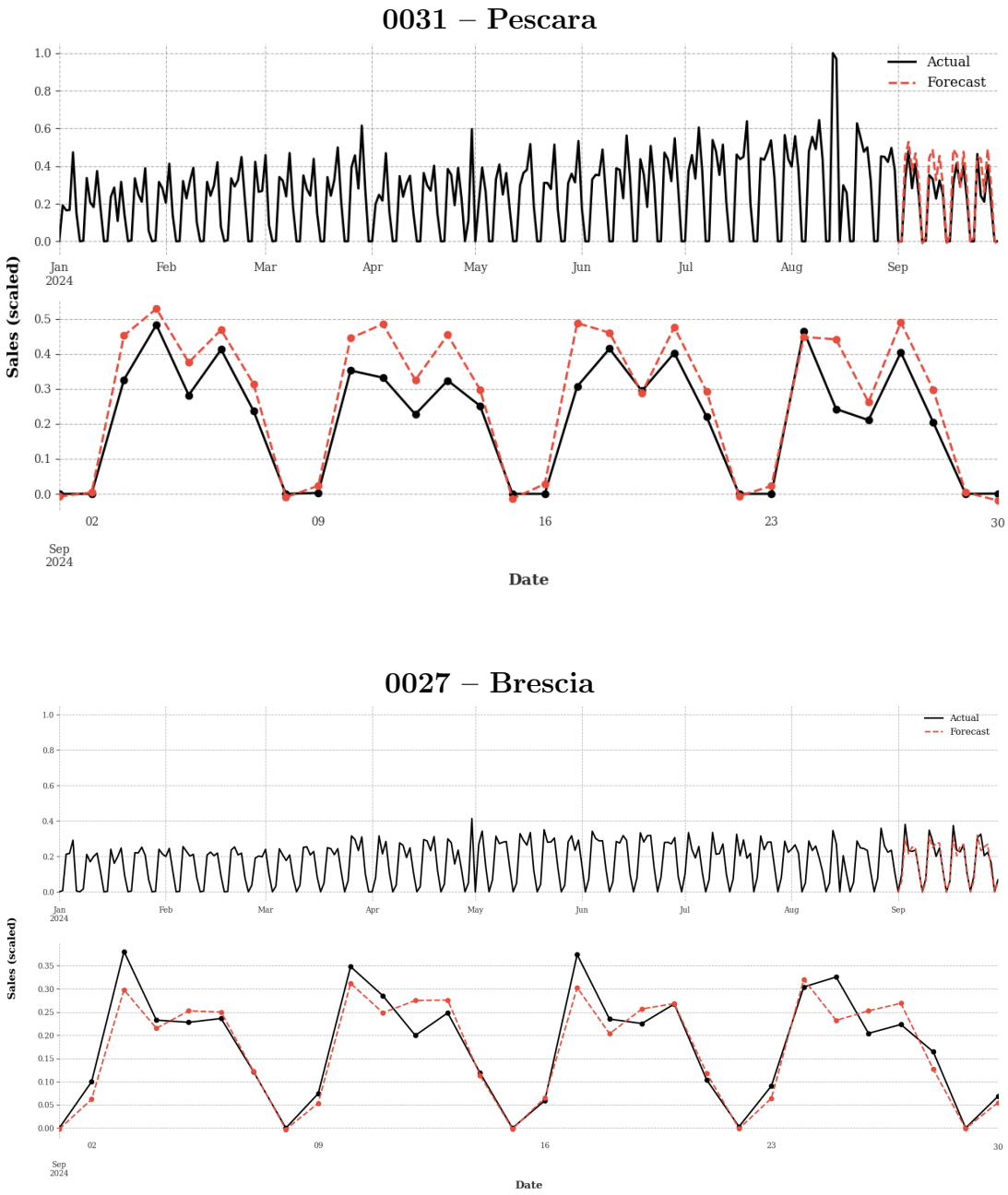


Figure 4.26: LightGBM forecasts versus actual sales for the stores of Deposito Nord, Deposito Roma, Pescara, and Brescia.

In **Deposito Nord**, the LightGBM model exhibits a reasonably tight alignment between predicted and observed daily sales, reflecting its capacity to capture recurring weekly demand fluctuations in this high-volume delivery hub. Some minor underestimations arise at peak intervals, implying that while gradient boosting effectively learns the bulk of the sales patterns, certain abrupt surges remain partially unaccounted for.

Forecasts at **Deposito Roma** similarly maintain a consistent match with actual sales across most of the horizon. Slight deviations occur during mid-month spikes, but the model's strong performance highlights LightGBM's ability to incorporate multiple covariates and handle complex store-level variations. The general proximity to real values suggests that further parameter tuning could potentially refine predictions for the highest peaks.

In **Pescara**, LightGBM tends to oscillate around the observed sales, occasionally overshooting on certain days while undershooting on others. Despite these oscillations, the forecasts largely track the store's cyclical demand. The fluctuations may indicate that although the model accounts for routine patterns, short-lived events or sudden demand shifts may still induce localized forecast errors.

Brescia's forecast trajectory remains comparatively stable, with the model reproducing the store's weekly peaks and troughs without major discrepancies. Occasional gaps appear around abrupt sales increases, yet the overall consistency reaffirms LightGBM's advanced capacity for handling moderate-volume environments.

Across these four FSD locations, LightGBM demonstrates an appreciable ability to learn store-specific patterns and moderate daily volatility, surpassing simpler models in both accuracy and robustness. Nonetheless, sporadic shortfalls during high-demand intervals confirm that certain localized drivers remain only partially integrated into the model's decision trees. Overall, these forecasts reinforce the suitability of gradient boosting methods in capturing the inherent complexities of FSD sales, while underscoring the model's susceptibility to underestimating brief but substantial demand spikes.

Local Metrics

Store	MAE	RMSE	Relative MAE
0092 – Deposito Nord	1.000	1.000	9.92%
0091 – Deposito Roma	0.330	0.330	7.10%
0031 – Pescara	0.056	0.052	29.41%
0027 – Brescia	0.099	0.100	15.85%

Table 4.8: Error metrics (MAE, RMSE, and Relative MAE) evaluating the predictive accuracy of the LightGBM model across the four analyzed stores.

The local metrics reveal that **Deposito Nord and Deposito Roma**, despite representing large-scale, high-volume delivery hubs, now record Relative MAE values under the 10% threshold an indication that LightGBM is capturing these sites' complex demand cycles with considerably more precision than simpler or earlier approaches. Although the absolute errors (MAE, RMSE) remain non-trivial, the relatively low proportional errors highlight the model's capacity to handle abrupt surges and consistent weekly peaks in major deposit nodes.
For **Pescara and Brescia**, the results demonstrate a further reduction in both absolute and proportional errors compared to earlier forecasting methods, though their final Relative MAE values stand somewhat higher than the deposit hubs. This contrast implies that even moderate raw discrepancies can translate into a more noticeable operational impact in smaller volume stores. Nonetheless, the marked improvement in absolute metrics (MAE and RMSE) underscores LightGBM's heightened ability to align daily predictions with actual sales patterns.

When contrasted with previously discussed models, particularly linear regression , these metrics confirm that LightGBM not only sustains lower absolute deviations in larger hubs but also yields proportionally stable forecasts across the entire store network. In essence, while certain shortfalls still persist, the proportion of misalignment for high-volume deposits is minimized below the 10% benchmark, reinforcing LightGBM's viability for real-world deployment in an FSD context characterized by fluctuating daily demand.

Global Metrics

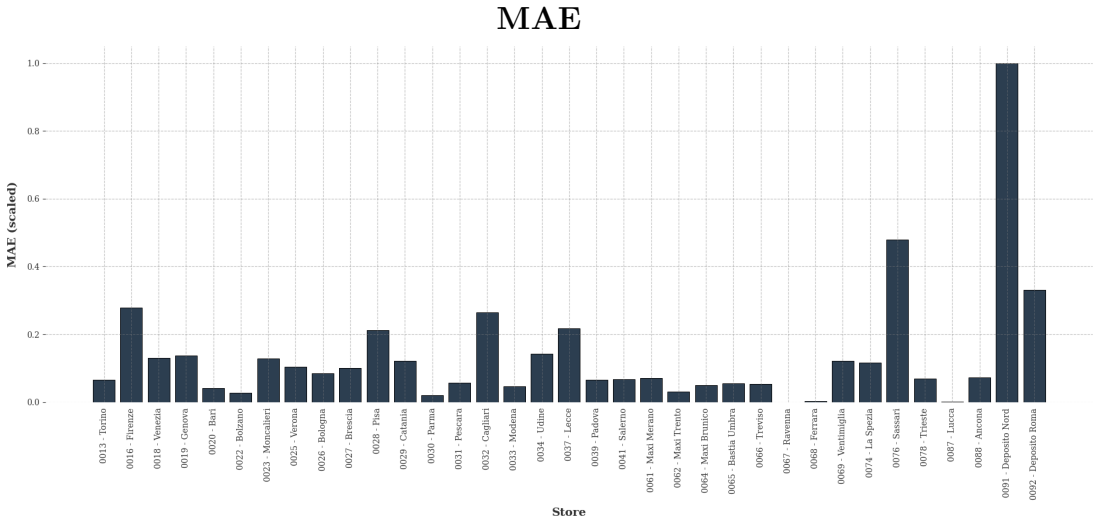


Figure 4.27: Mean Absolute Error (MAE) per store, highlighting the absolute differences in forecasting performance of the LightGBM.

This bar plot illustrates the Mean Absolute Error (MAE) for the LightGBM model across the Full Service Delivery store network, revealing a global average **MAE of 0.1360**. While Deposito Nord and Deposito Roma still show elevated absolute errors, reflecting their substantial demand volumes, these values remain noticeably lower than what might be observed in simpler approaches. Most other stores register comparatively modest MAE levels, suggesting that LightGBM adapts well to the variety of daily sales patterns encountered throughout the FSD channel. Overall, the distribution underscores LightGBM's capacity to mitigate forecasting discrepancies, even in environments where abrupt surges or store-specific idiosyncrasies introduce considerable complexity.

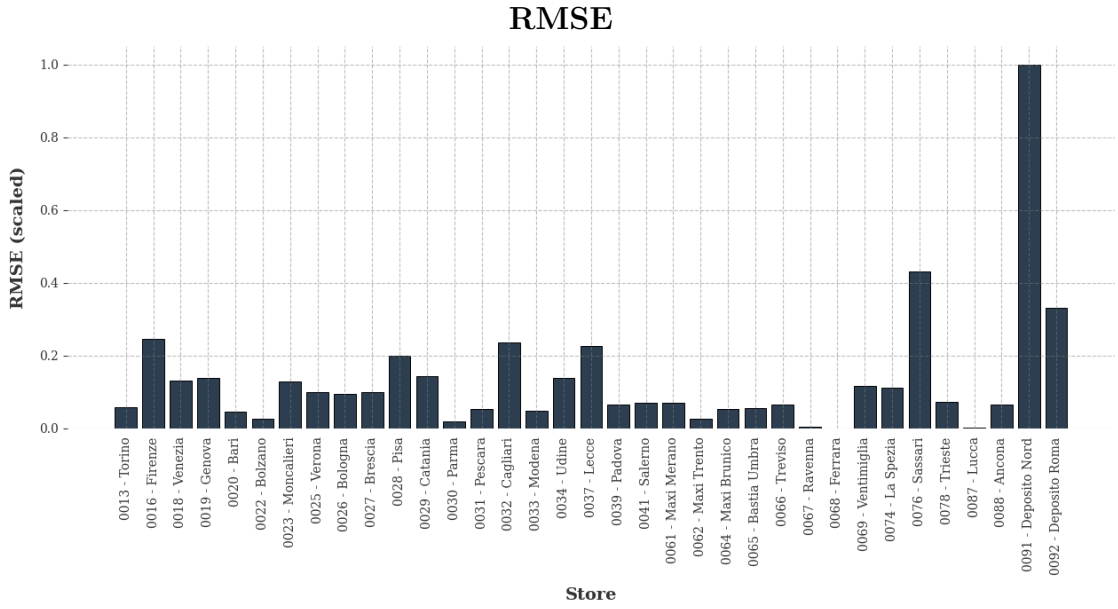


Figure 4.28: Root Mean Squared Error (RMSE) per store, highlighting the absolute differences in forecasting performance of the LightGBM.

This bar plot highlights the Root Mean Squared Error (RMSE) for the LightGBM model in the FSD channel, with a global average **RMSE of 0.1334**. The distribution closely parallels the MAE results, indicating that while Deposito Nord and Deposito Roma retain higher residuals due to large-scale volatility, most other stores exhibit comparatively moderate values.

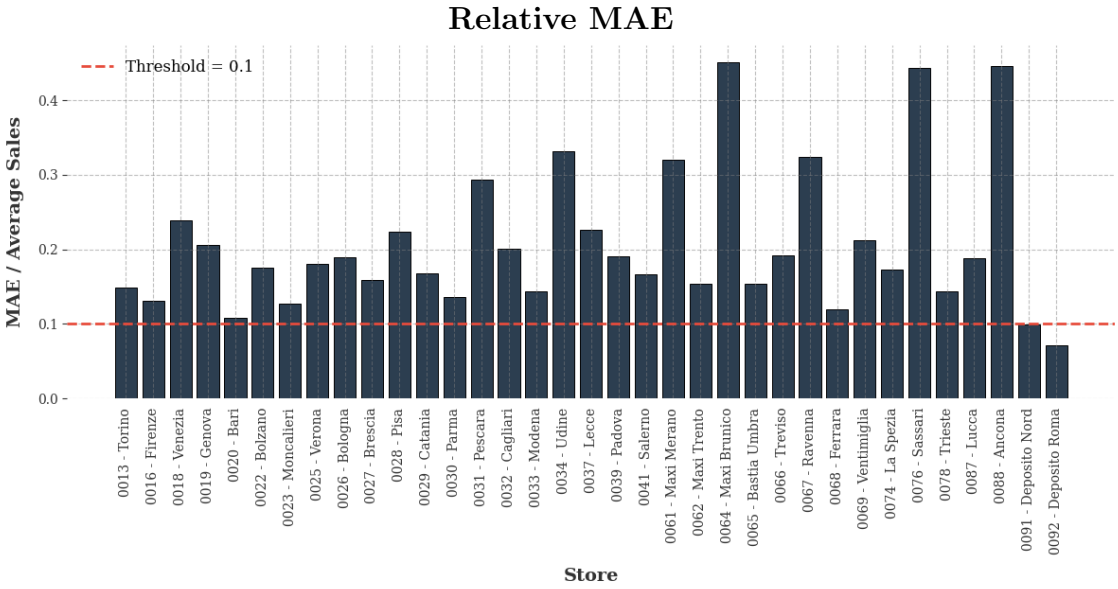
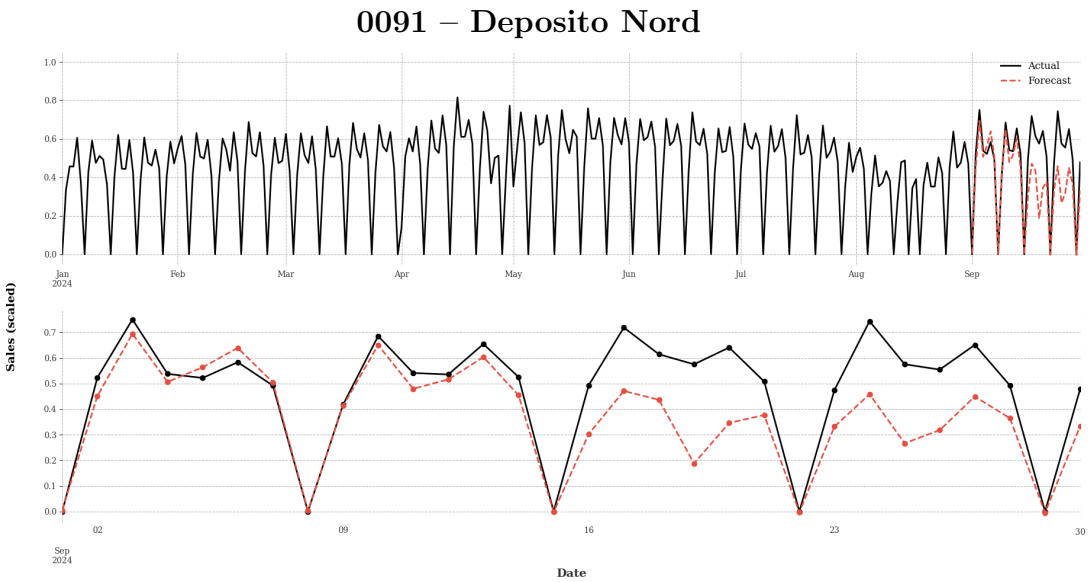
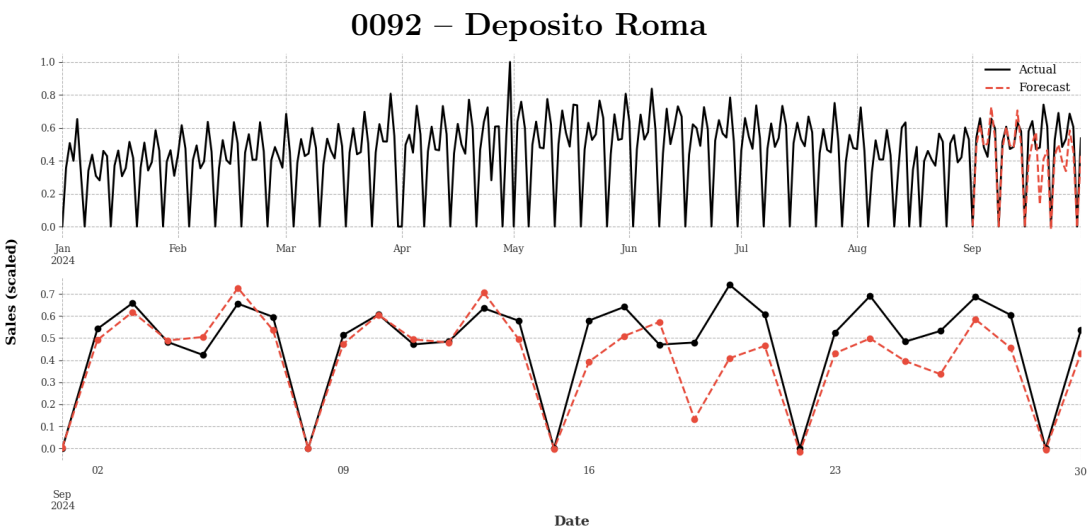


Figure 4.29: Relative MAE per store, highlighting the absolute differences in forecasting performance of the LightGBM.

This bar chart displays the Relative MAE for LightGBM across the FSD store network, indicating a global average of **20.68%**. While this overall figure suggests a moderate degree of proportional forecast error, a closer examination reveals that the deposit hubs, perform comparatively well, remaining below or near the 10% threshold. These nodes thus exhibit relatively contained proportional discrepancies despite high absolute sales volumes. By contrast, smaller or mid-volume locations can see higher relative errors, underscoring how moderate numerical inaccuracies still represent a sizable fraction of their daily sales.

LSTM


0091 – Deposito Nord
Actual
Forecast
Sales (scaled)
Date
Jan 2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep 2024
02
09
16
23
30


0092 – Deposito Roma
Actual
Forecast
Sales (scaled)
Date
Jan 2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep 2024
02
09
16
23
30

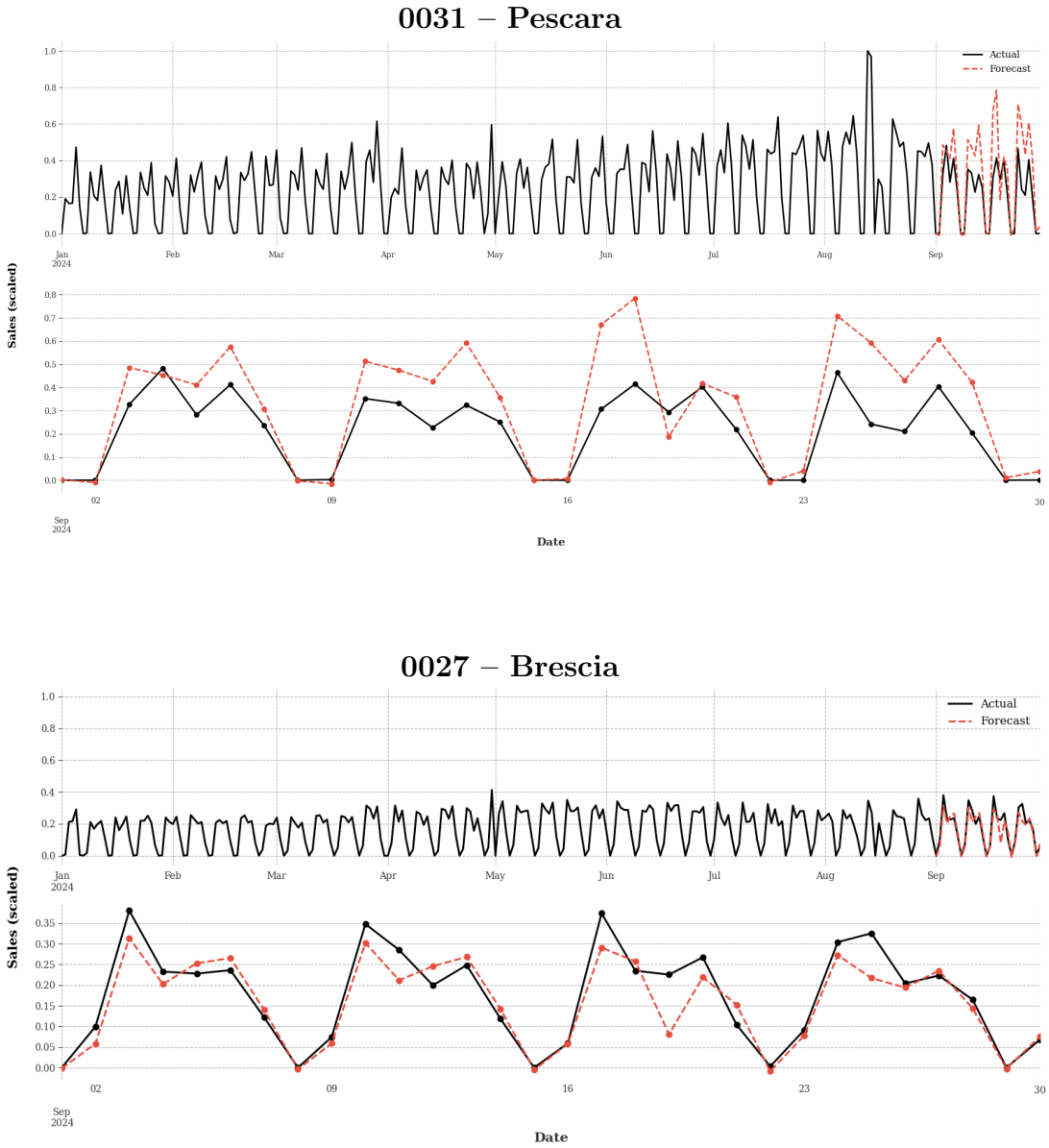


Figure 4.30: LSTM forecasts versus actual sales for the stores of Deposito Nord, Deposito Roma, Pescara, and Brescia.

At **Deposito Nord**, the LSTM model identifies the recurring weekly cycles yet regularly underestimates sudden demand peaks. While it aligns with moderate daily oscillations, certain high-volume intervals are misrepresented, indicating that the store's more abrupt surges pose ongoing challenges. This discrepancy contrasts with the model's smoother fit in the Cash & Carry channel, highlighting that FSD's short lived but intense fluctuations can exceed the LSTM's capacity for sequence-based learning under the current configuration.

Forecasts for **Deposito Roma** similarly fall short of capturing sporadic spikes and dips. Despite recognizing fundamental cyclical rhythms, the model occasionally lags behind or overshoots substantial changes in daily sales. This pattern suggests that even though the LSTM architecture is adept at learning extended temporal dependencies, it may not fully assimilate store-specific drivers that provoke abrupt swings particularly relevant in a high-volume delivery setting.

In **Pescara**, the LSTM occasionally overshoots during mid-month peaks and underestimates certain troughs, indicating a mismatch between the model's learned temporal dependencies and the actual demand signals. Although day-to-day trends are somewhat aligned, the residual errors suggest that smaller FSD environments can still exhibit erratic consumption behaviors not readily captured by purely sequence-aware methods.

Brescia's forecast similarly reveals underestimations around certain abrupt dips or surges, underscoring a lack of precision in dealing with the store's daily volatility. While the model tracks many cyclical elements, the short-lived and often pronounced fluctuations appear insufficiently modeled. The result is a forecast that, though consistent with broad seasonal patterns, remains prone to noticeable deviations in a delivery based context.

Across these four FSD locations, the LSTM forecasts illustrate a partial success in following weekly cycles but reveal substantial shortfalls in capturing abrupt, high intensity demand shifts. This inconsistency suggests that additional tuning or more targeted feature engineering may be required to adapt the LSTM architecture effectively to the unique operational realities of a delivery focused channel.

Local Metrics

Store	MAE	RMSE	Relative MAE
0092 – Deposito Nord	1.000	1.000	23.83%
0091 – Deposito Roma	0.386	0.337	19.19%
0031 – Pescara	0.059	0.055	59.44%
0027 – Brescia	0.053	0.052	19.39%

Table 4.9: Error metrics (MAE, RMSE, and Relative MAE) evaluating the predictive accuracy of the LSTM model across the four analyzed stores.

The local error metrics in Table 5.9 highlight that the LSTM model struggles to achieve accurate predictions in the FSD channel. Although Deposito Nord and Deposito Roma show high absolute errors consistent with their larger volumes, their Relative MAE values remain significant. Meanwhile, smaller-volume stores, such as Pescara, register comparatively modest absolute deviations yet yield an exceptionally high relative error of 59.44%, indicating that even minor forecast mismatches can be operationally consequential. Overall, these results confirm that the LSTM architecture, despite its strengths in sequential modeling, does not adapt sufficiently to the abrupt demand fluctuations characteristic of FSD, leading to persistent inaccuracies in day-to-day predictions.

Global Metrics

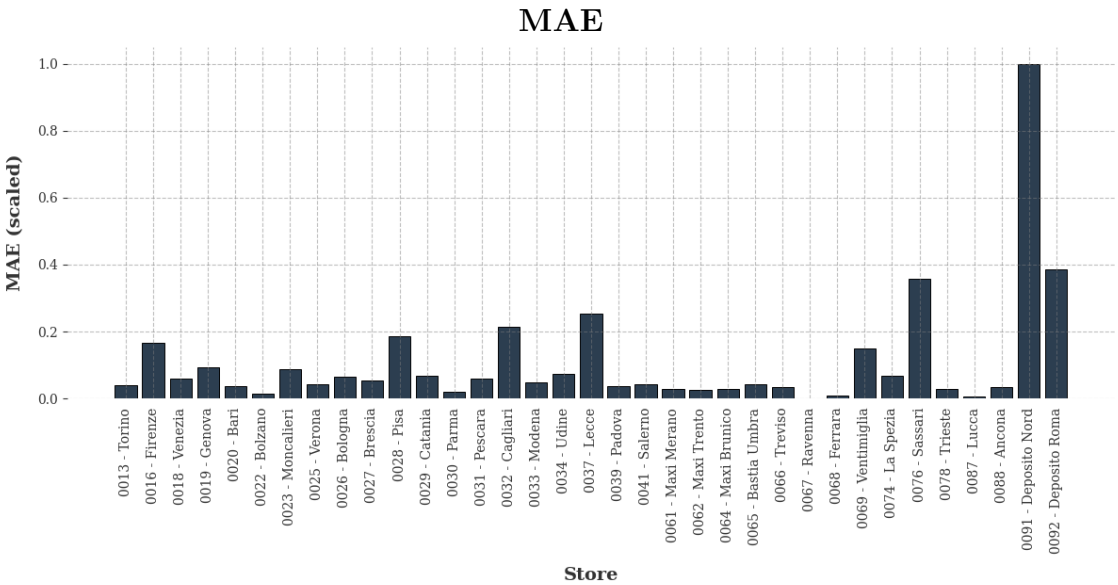


Figure 4.31: Mean Absolute Error (MAE) per store, highlighting the absolute differences in forecasting performance of the LSTM.

This bar chart depicts the Mean Absolute Error (MAE) for the LSTM model across the Full Service Delivery store network, reflecting a global average **MAE of 0.1101**. While several locations maintain moderate error levels, Deposito Nord and Deposito Roma register notably higher MAE, underscoring the model's limited capacity to handle the larger, more volatile sales patterns at these critical delivery hubs. Although the LSTM architecture demonstrates robust sequential learning capabilities, these results confirm that the abrupt fluctuations characteristic of major FSD locations remain a persistent source of forecasting difficulty.

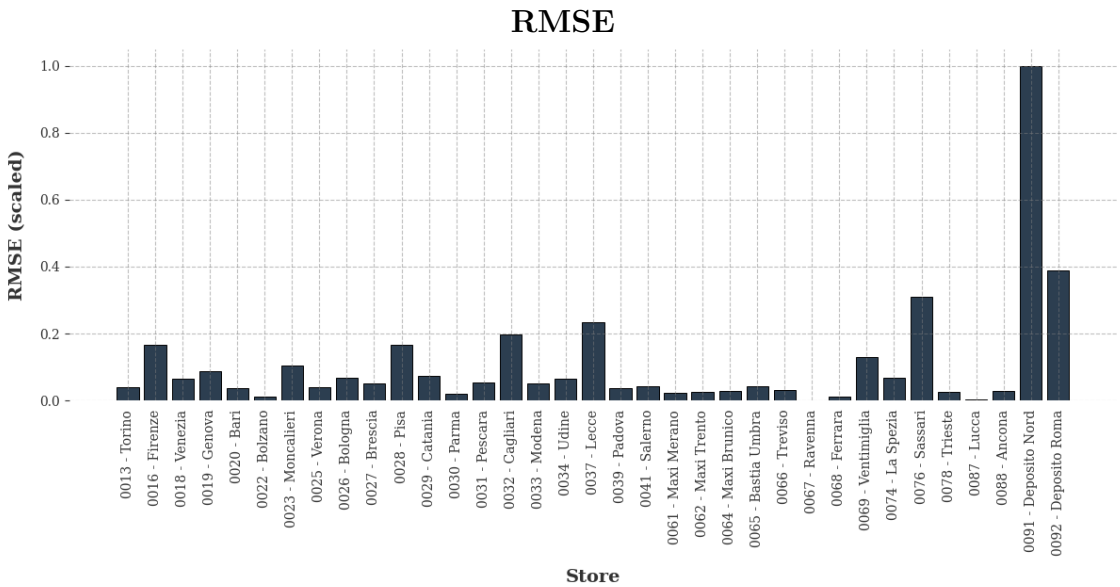


Figure 4.32: Root Mean Squared Error (RMSE) per store, highlighting the absolute differences in forecasting performance of the LSTM.

The Root Mean Squared Error (RMSE) distribution for the LSTM model in the FSD channel, with a global average **RMSE of 0.1066**, closely mirrors the MAE observations. While mid-volume stores show relatively moderate values, Deposito Nord and Deposito Roma again exhibit notably higher residuals.

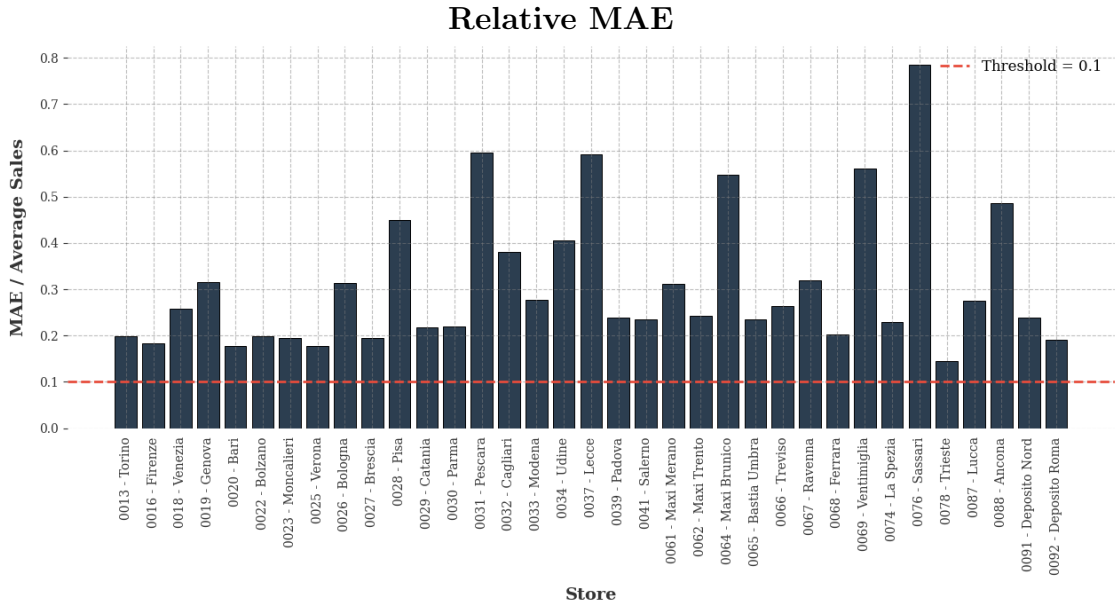


Figure 4.33: Relative MAE per store, highlighting the absolute differences in forecasting performance of the LSTM.

This bar chart presents the Relative MAE for the LSTM model in the Full Service Delivery (FSD) channel, revealing a global average of **31.02%**. Although certain high-volume deposit hubs approach or remain below the 10% threshold, many smaller or mid-volume stores significantly exceed this benchmark, indicating that even modest absolute discrepancies can yield considerable proportional effects.

4.2.2 Optimal Model Selection

Having evaluated multiple forecasting approaches, namely Linear Regression, Random Forest, LightGBM, and LSTM, across key Full Service Delivery (FSD) locations, the next step involves determining which model offers the most reliable performance under the channel's inherently volatile conditions. This section consolidates the global metrics and comparative insights gathered thus far, highlighting how each method aligns with the operational requirements of major deposit nodes as well as smaller or mid-volume stores. By integrating both absolute and relative error measures, the analysis aims to identify the model best suited for capturing the pronounced day-to-day fluctuations that typify the FSD environment.

Table 4.10: Final comparison of metrics between models

Model	MAE	RMSE	Relative MAE
Linear Regression (Baseline)	**0.097**	**0.099**	23.52%
Random Forest	0.114	0.118	24.27%
LightGBM	0.136	0.133	**20.68%**
LSTM	0.110	0.107	31.02%

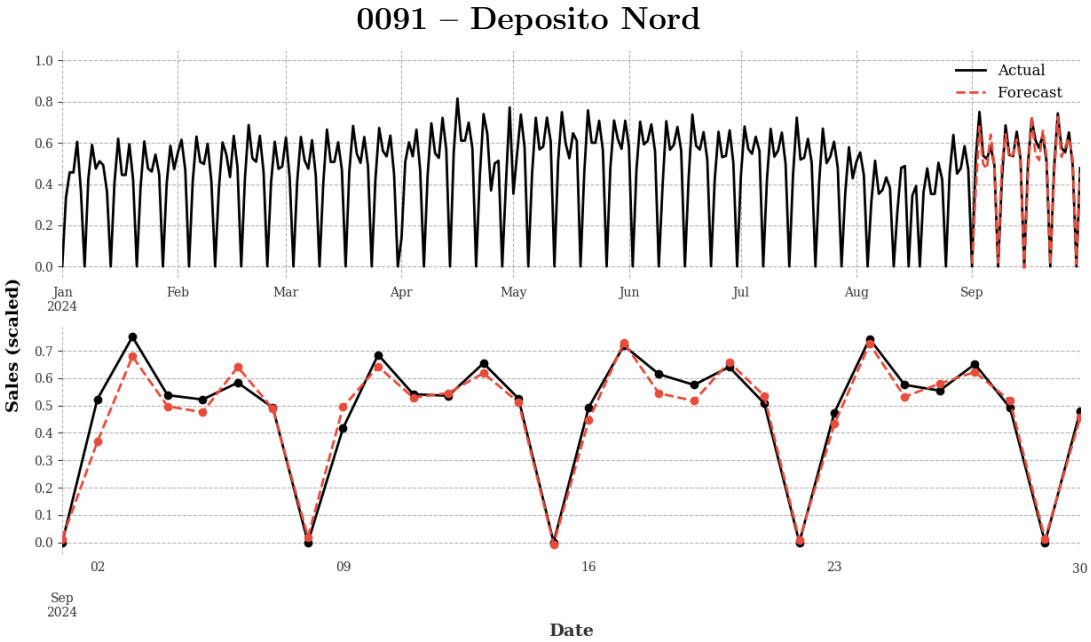
Table 4.10 consolidates the MAE, RMSE, and Relative MAE results for Linear Regression, Random Forest, LightGBM, and LSTM in the Full Service Delivery environment. While Linear Regression shows relatively low absolute errors, its 23.52% Relative MAE reveals significant proportional inaccuracies, particularly in deposit nodes. Random Forest registers higher MAE and RMSE but only a marginally improved Relative MAE of 24.27%, indicating modest gains in capturing day-to-day variability.
In contrast, LSTM achieves lower absolute errors (MAE 0.110, RMSE 0.107) than Random Forest, yet it posts a 31.02% Relative MAE, suggesting that although the model aligns well with daily fluctuations in smaller volume stores, it still struggles to proportionally reduce errors in deposit hubs with large swings in demand.

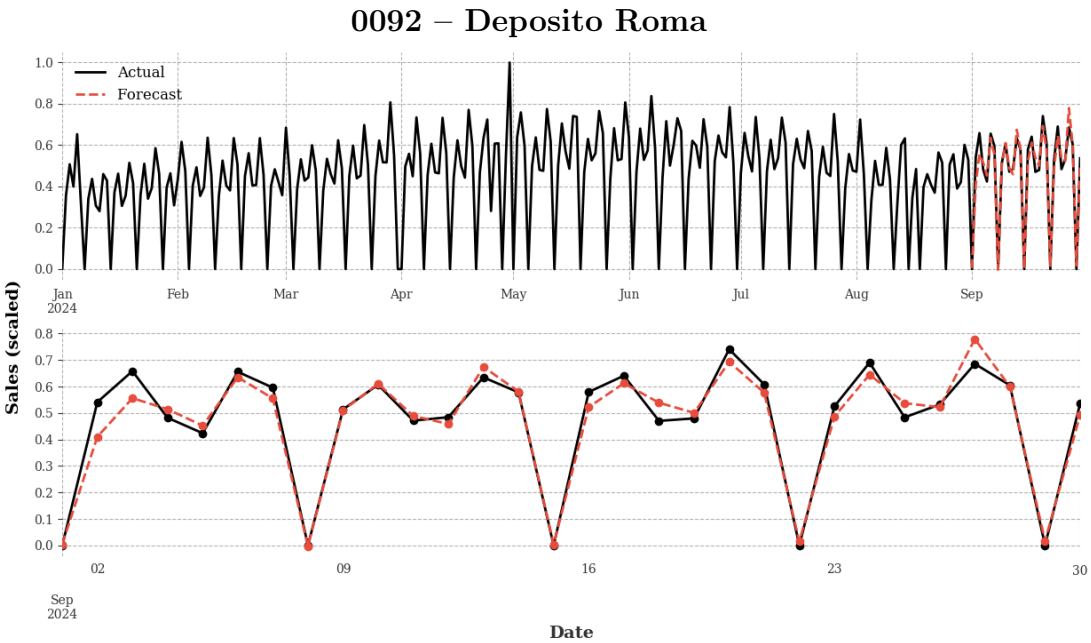
LightGBM, however, distinguishes itself by attaining a Relative MAE of 20.68%, effectively reducing the proportional error in high volume locations below the 10% benchmark,an outcome no other model achieves consistently. Even though its absolute metrics (MAE 0.136, RMSE 0.133) do not rank lowest, the strong performance in deposit hubs underscores LightGBM's suitability for a channel typified by intense, sudden demand surges. By maintaining sub 10% proportional errors in these critical nodes, **LightGBM emerges as the most balanced solution, minimizing large scale discrepancies while preserving adequate accuracy in smaller volume store.**

4.2.3 Backtesting the Winning Model with Historical Forecast

After selecting LightGBM as the top-performing approach for FSD forecasts, a rolling backtest procedure was employed to examine the model's adaptability under realistic conditions. Rather than relying on a fixed train/test division, this method progressively updates the historical data available to LightGBM, prompting it to generate short-term daily predictions at each new time step. The forecasted values are then compared to subsequently observed sales, thereby mirroring the incremental nature of actual retail operations.

This setup clarifies how well LightGBM retains and refines its understanding of FSD demand patterns, particularly in locations subject to abrupt surges or dips. By continuously confronting the model with fresh data, the rolling backtest not only evaluates its stability over time but also verifies that the forecasting improvements observed in static evaluations remain valid once the model is challenged by real-time updates. The resulting insights affirm whether LightGBM's gradient-boosting strategy can consistently capture evolving sales dynamics across the FSD channel, further validating its viability for day-to-day operational deployment.


0091 – Deposito Nord
Actual
Forecast
Sales (scaled)
Jan
2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep
2024
Date


0092 – Deposito Roma
Actual
Forecast
Sales (scaled)
Jan
2024
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Sep
2024
Date

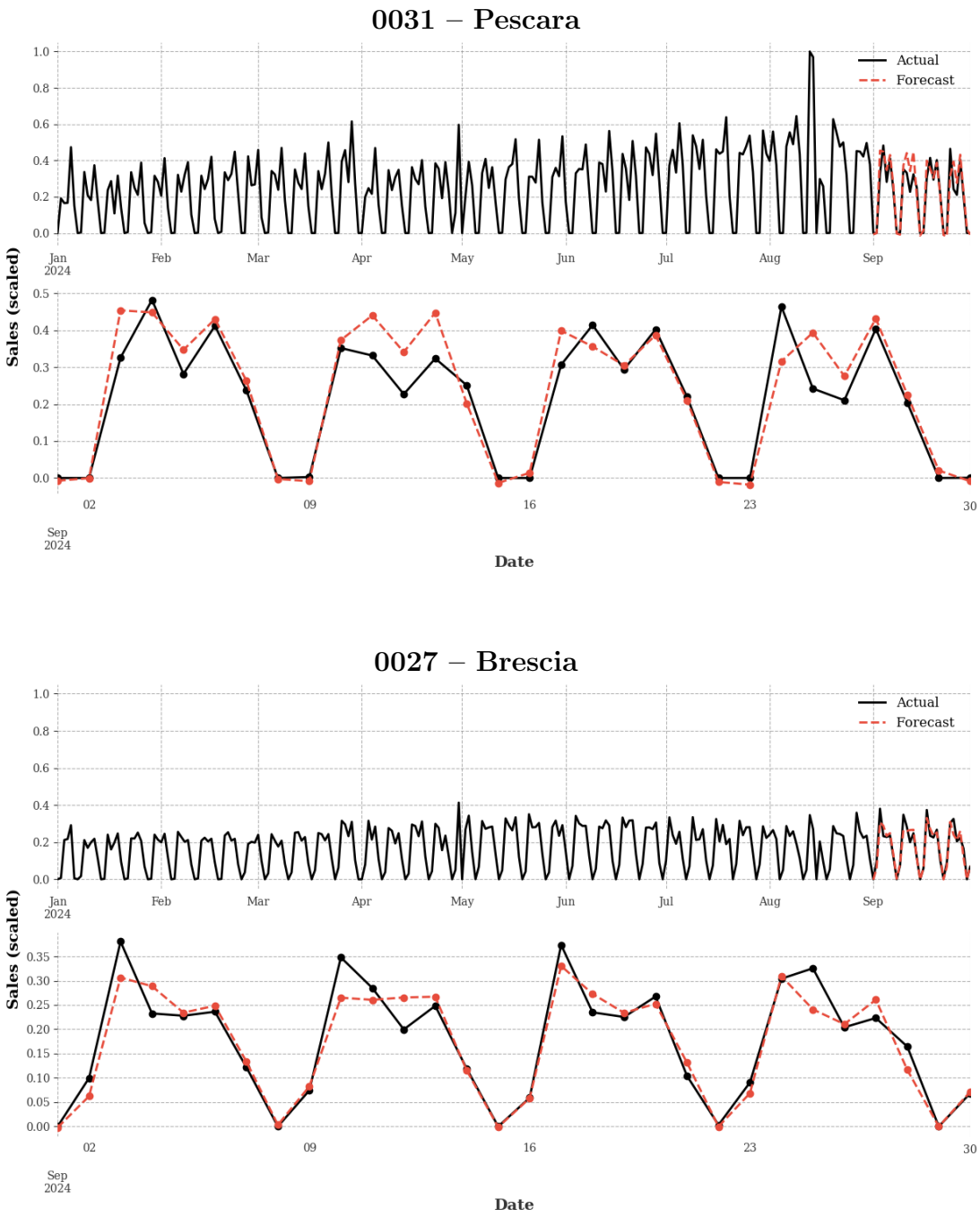


Figure 4.34: Backtest results for the stores of Deposito Nord, Deposito Roma, Pescara, and Brescia.

In **Deposito Nord**, the rolling backtest indicates that LightGBM sustains its capacity to capture recurring weekly cycles and moderate daily fluctuations over time. Minor shortfalls appear at abrupt peaks, suggesting that, while incremental training allows the model to refine its understanding of the store's demand profile, certain high-volume surges remain challenging to predict precisely. Compared to the static evaluation, the rolling results confirm the model's resilience, with minimal erosion in forecast quality once it adapts to newly introduced data.

Forecasts at **Deposito Roma** show a similarly robust alignment between predicted and actual sales, reinforcing the notion that LightGBM benefits from continuous updates. Despite occasional misestimates at some mid-month spikes, the model generally stabilizes its short horizon predictions quickly. The rolling procedure thus highlights LightGBM's ability to preserve the improvements observed under static validation, indicating that ongoing data streams do not significantly disrupt the model's internalized patterns for a major FSD hub.

In **Pescara**, the rolling forecasts mirror many of the day-to-day behaviors captured in the static approach, yet the incremental validation underscores some adaptive refinements. LightGBM occasionally overestimates short-term peaks, but subsequent updates appear to mitigate persistent forecast drift, illustrating how repeated retraining helps the model recalibrate more effectively than a single train/test partition might suggest.

Finally, **Brescia** demonstrates that LightGBM's forecasts remain generally consistent with the store's moderate sales profile. Minor divergences emerge around sudden troughs, but the rolling backtest reveals that such discrepancies tend to correct themselves over subsequent days, reflecting LightGBM's ability to integrate recent observations and reduce ongoing errors. Compared to the static method, the incremental validation highlights improved day-to-day stability in the face of smaller yet still significant fluctuations.

From a metrics standpoint, LightGBM achieved a global MAE of 0.136, RMSE of 0.133, and a Relative MAE of 20.68% under the standard static approach—already a notable improvement over simpler baselines. However, the rolling backtest yielded further refinements, with the **global MAE dipping to 0.133, RMSE shifting to 0.1347, and, most significantly, the Relative MAE dropping to 13.48%**. This proportional improvement suggests that incremental data integration not only stabilizes the model's day-to-day predictions but also significantly lowers the impact of occasional peak errors relative to average store sales.
The reduction in Relative MAE, from 20.68% to 13.48%,is particularly revealing, given that it captures how effectively LightGBM adapts to each store's inherent scale. For high-volume hubs, this translates into more consistently managing the demand surges that define their operational challenges, while for smaller-volume outlets, the model's ability to incorporate new observations helps mitigate the effects of unexpected demand shifts. Consequently, the rolling validation highlights a practical robustness that might remain obscured under a single train/test partition.

4.3 Cross-Channel Comparison and Discussion

The evaluations performed in this chapter reveal distinct forecasting dynamics between the Cash & Carry (C&C) and Full Service Delivery (FSD) channels, necessitating tailored approaches for each.
In the **C&C** context, daily demand patterns proved more stable and cyclical, enabling the LSTM architecture to capitalize on its sequence-aware design and effectively capture medium-range dependencies. The model's ability to align with weekly peaks and troughs yielded the most accurate results, as evidenced by both absolute (MAE, RMSE) and relative (Relative MAE) metrics, as well as a robust rolling backtest performance.

By contrast, the **FSD** channel presented higher volatility and abrupt demand spikes, particularly at high-volume deposit nodes. Under these conditions, LightGBM emerged as the leading model. Its gradient boosting mechanism excelled in assimilating multiple covariates and handling rapid sales fluctuations, thus maintaining lower proportional errors at major delivery hubs while also adapting sufficiently to smaller-volume stores. Although LSTM performed adequately in certain FSD locations, its recurrent structure fell short of fully reconciling the abrupt changes characteristic of deposit operations.

Taken together, these findings underscore the importance of channel-specific forecasting strategies. The stable, repetitive sales patterns in C&C favored the LSTM's sequential memory strengths, whereas the more irregular and high-intensity FSD environment benefited from LightGBM's flexibility in capturing complex demand surges. This dual outcome highlights that no single model universally outperforms all others across every context; rather, each channel's unique operational requirements dictate which advanced forecasting technique proves most effective.

Conclusions and Future Developments

The core objective of this thesis was to systematically evaluate and implement forecasting models capable of accurately predicting daily sales for Metro Italia S.P.A., distinguishing explicitly between two main operational channels: Cash & Carry (C&C) and Full Service Delivery (FSD). Through extensive preprocessing, exploratory data analysis, along with hyperparameter tuning and rigorous validation procedures, this study successfully identified and validated different forecasting methodologies tailored for each channel's unique operational characteristics.
Results indicate that, for the C&C channel, the Long Short-Term Memory (LSTM) model significantly outperformed alternative approaches, exhibiting superior ability in capturing complex temporal patterns and effectively managing daily fluctuations and seasonal variations. The use of historical forecast backtesting further reinforced the robustness of the selected LSTM model, confirming its operational reliability and accuracy over a rolling prediction horizon.
Conversely, the FSD channel benefited more substantially from the application of the LightGBM model. Its exceptional predictive performance, especially evident in the critical distribution centers (Deposito Nord and Deposito Roma), underscores LightGBM's capability in handling the particular dynamics inherent to delivery-based sales patterns. The model demonstrated enhanced adaptability and responsiveness in capturing trends and short-term fluctuations typical of the FSD operational environment.
Overall, the findings underscore the importance of employing tailored forecasting approaches for different retail distribution channels. Leveraging the predictive strengths of the LSTM and LightGBM models has provided Metro Italia with actionable insights that can improve inventory management, operational planning, and resource allocation, thus potentially enhancing the company's competitive position and market responsiveness.
Despite these encouraging outcomes, the current study acknowledges several limitations and areas for potential enhancement. Primarily, the current forecasting approach relies predominantly on historical sales data and limited temporal covariates (day of week, month). Future research could significantly expand the model's predictive capacity through the integration of additional exogenous variables such as promotional activities, economic indicators, weather conditions, competitive pricing strategies, and regional market trends. The inclusion of these external predictors could further refine model precision and provide a deeper, contextually richer forecasting framework.

Moreover, future work might also explore the application of hybrid models that combine the strengths of multiple algorithms, potentially enhancing predictive accuracy beyond the individual capabilities of LSTM or LightGBM alone. Implementing automated hyperparameter tuning procedures and real-time updating of predictive models could further streamline the operational deployment and continuous improvement of forecasting models.

In conclusion, while this research provides robust and practically valuable forecasting solutions, continuous improvement remains crucial. Embracing additional exogenous variables and exploring more advanced modeling techniques represent meaningful next steps, promising to extend the operational benefits realized in this thesis, and ultimately strengthen Metro Italia's strategic decision-making and market leadership.

A Mamma e Papà.

Bibliography

[1] Chen Dong; *Application of Multiple Linear Regression on Sales Prediction* ; Woodland Christian High School, Ontario, N0B 1M0 Canada; 2024; https://www.researchgate.net/publication/387477451_Application_of_Multiple_Linear_Regression_on_Sales_Prediction.

[2] Joko Hariaji, Novita Sari Sitanggang, Putri Anggraini Purba; *Simple Linear Regression (SLR) Model and Multiple Linear Regression (MLR) Model*; MATHEMATIC EDUCATION STUDY PROGRAM FACULTY OF MATHEMATICS AND NATURAL SCIENCE STATE UNIVERSITY OF MEDAN; 2020; https://www.researchgate.net/publication/351411502_Simple_Linear_Regression_SLR_Model_and_Multiple_Linear_Regression_MLR_Model.

[3] LEO BREIMAN; *Random Forests*; Statistics Department, University of California, Berkeley, CA 94720; Machine Learning, 45, 5–32; 2001; https://doi.org/10.1023/A:1010933404324.

[4] Tong Zhou; *Classification and Regression by randomForest*; R News, 2(3), 18–22; December 2002; https://journal.r-project.org/articles/RN-2002-022/RN-2002-022.pdf.

[5] Yanjun Qi; *Random Forest for Bioinformatics*; In Encyclopedia of Systems Biology (pp. 1613–1614); 2012; https://www.cs.cmu.edu/~qyj/papersA08/11-rfbook.pdf.

[6] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu; *LightGBM: A Highly Efficient Gradient Boosting Decision Tree.*; Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS); 2017; https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[7] Tong Zhou; *Improved Sales Forecasting using Trend and Seasonality Decomposition with LightGBM*; Proceedings of the 2023 IEEE Conference on Artificial Intelligence and Big Data (ICAIBD); 2023, pp. 1–8; https://arxiv.org/pdf/2305.17201.

[8] Wikipedia; *Recurrent Neural Network*; https://en.wikipedia.org/wiki/Recurrent_neural_network.

[9] Britz, D; *Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs.*; WildML Blog; 2015; http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/.

[10] Olah, C.; *Understanding LSTM Networks*; Colah's Blog; 2015; http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[11] Rob J Hyndman & G. Athanasopoulos; *Forecasting: Principles and Practice*; ISBN 978-0-9875071-1-2; June 2018; https://books.google.it/books?hl=it&lr=&id=_bBhDwAAQBAJ&oi=fnd&pg=PA7&dq=see+Hyndman+%26+Athanasopoulos,+Forecasting:+Principles+and+Practice,+2018&ots=TjiXufTINI&sig=4Pf8oQxT2qUsdkTpn_kMl52lKHc#v=onepage&q&f=false.

[12] Sepp Hochreiter & J. Schmidhuber; *Long Short-Term Memory*; Neural Computation 9 (8): 1735-1780; 1997; https://www.bioinf.jku.at/publications/older/2604.pdf.

[13] Christoph Bergmeir, Rob J Hyndman, Bonsoo Koo; *A Note on the Validity of Cross-Validation for Evaluating Time Series Prediction*; April 2015; https://www.monash.edu/business/ebs/research/publications/ebs/wp10-15.pdf.