TREE-BASED MODELS FOR MULTI-STEP TIME SERIES FORECASTING:

A COMPARATIVE STUDY OF RECURSIVE AND DIRECT APPROACHES WITH SLIDING AND EXPANDING WINDOWS

SPYRIDON TEFOS

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

314506

COMMITTEE

Dr. B. Čule
Dr. Mirella De Sisto

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

December 5th, 2024

WORD COUNT

8783

ACKNOWLEDGMENTS

TREE-BASED MODELS FOR MULTI-STEP TIME SERIES FORECASTING:

A COMPARATIVE STUDY OF RECURSIVE AND DIRECT APPROACHES WITH SLIDING AND EXPANDING WINDOWS

SPYRIDON TEFOS

Abstract

In the dynamic landscape of retail demand forecasting, traditional methods face challenges, necessitating a comprehensive consideration of influential factors like price changes, population shifts, and logistics complexities. This study focuses on enhancing predictive capabilities in retail sales forecasting, utilizing machine learning models—LightGBM (LGBM) and Random Forest (RF)—on weekly sales data from Walmart. The central problem involves adapting forecasting models to evolving sales patterns.

Distinguishing itself from prior approaches, the research employs a comprehensive evaluation of direct and recursive forecasting methods, integrating sliding and expanding window techniques. The primary dataset comprises weekly sales data from Walmart, transformed from daily to accommodate computational constraints.

Key findings reveal that RF models consistently outperform LGBM ones in the direct approach, indicating lower Root Mean Squared Error (RMSE) and Weighted Root Mean Squared Scaled Error (WRMSSE) metrics. In contrast, LGBM exhibits better performance in the recursive approach, highlighting adaptability to evolving patterns over successive forecasting periods.

This study contributes by assessing model performance and exploring various forecasting approaches, recommending the consideration of longer observation windows and the incorporation of external factors like weather. Despite limitations, such as the transformation of daily sales data into weekly data, this research provides valuable insights into the strengths and weaknesses of machine learning models in retail sales forecasting, guiding future research for more effective predictive modeling in the evolving retail landscape.

1 DATA SOURCE, ETHICS, CODE, AND TECHNOLOGY STATEMENT

The dataset used in this thesis, publicly accessible due to its association with the M5 competition organized by Markidakhs was obtained through an online request and is anonymized [1]. No data collection involving human participants or animals was conducted. All libraries and frameworks used, including version numbers, are listed. All figures in the thesis, were created by the author. ChatGPT has been used as a debugging tool to resolve coding errors [2]. For assistance in academic writing and grammar, the author used Thesaurus [3] and Grammarly was employed for additional spelling and grammar checks [4].The Overleaf LaTeX template provided by Tilburg University was used for typesetting, and no other typesetting tools or services were employed.

2 INTRODUCTION

The retail industry, particularly giants like Walmart, serves as a linchpin in society's socioeconomic fabric, having a profound impact on many aspects of daily life. These retail behemoths are more than just commercial entities; they shape the economic landscape and influence societal stability. The National Retail Federation and PwC report on the retail industry's economic impact highlights the industry's significant role as a vital pillar of the US economy.

In 2018, the retail business supported 52 million jobs, accounting for 25.8% of total US employment. This impact stretched to a total labor income of $2.3 trillion, accounting for 18.7% of national labor income, and a total GDP impact of $3.9 trillion, accounting for 18.7% of US GDP. These data highlight the retail sector's enormous economic importance, putting it as a primary driver of employment, income, and GDP at both the national and state levels. The broad network of outlets throughout many states, together with a diverse range of product categories, reinforces the industry's societal significance (Federation, 2020).

The retail industry, on the other hand, is subject to a plethora of external factors that can introduce volatility and uncertainty into their operations. These factors include anything from technological advancements to economic conditions to social trends. The retail trade volume index, for example, fell sharply in April 2020 as a result of the Covid-19 crisis [5] .

[1] Kaggle:https://www.kaggle.com/competitions/m5-forecasting-accuracy/data
[2] Chat GPT 3.5https://chat.openai.com
[3] Thesaurus:https://www.thesaurus.com/
[4] Grammarly:https://www.grammarly.com/
[5] eurostat: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Impact_of_Covid-19_crisis_on_retail_trade

As a result, the ability to forecast future sales becomes critical for these companies to navigate the volatile retail environment and secure their future.

To some extent, the retailers' future strategic planning is dependent on demand predictions made possible by methodologies and procedures integrated into a forecasting support system. The accuracy of demand forecasting has a significant impact on organizational performance by improving a variety of processes throughout the retail supply chain. There is a significant and direct increase in profitability, particularly in low-margin, high-volume retail scenarios (Ben Taieb & Hyndman, 2012).

Unfortunately, a few large corporations, such as Walmart and Amazon, are still laggards in data analytics (Tan, 2020). Furthermore, most retailers use basic analytic tools despite the fact that advanced tactics can increase their margins by 60% (DeHoratius et al., 2023; Lekhwar et al., 2019). This is a disadvantage for the current industry because small and medium-sized businesses account for 99 percent of the retail ecosystem [6]. As a result, advanced sales prediction models are in high demand in the industry.

Retail behemoths like Walmart can make informed decisions about resource allocation, strategic planning, and risk management by accurately forecasting sales. This not only ensures the long-term viability of their operations, but also contributes to the overall stability of the retail industry and, by extension of the society. As a result, this study has important implications for both academia and business, providing valuable insights into the use of machine learning techniques in sales forecasting for large-scale retail operations.

Accurately forecasting sales for large-scale retail operations like Walmart is a challenging task due to the sheer volume and complexity of the data. This research proposes a novel approach that utilizes tree-based algorithms and transforms daily sales data into weekly aggregates. This transformation not only reduces computational costs but also enables efficient data processing across all stores, states, and product categories. Additionally, this study comprehensively evaluates direct and recursive prediction methods, as well as sliding and expanding tuning techniques, providing insights into the effectiveness of various forecasting methods. This holistic approach effectively captures the complexities of Walmart's operations and sheds light on the most effective forecasting methods and techniques for large retailers.

Based on the aforementioned challenges, an overarching research question was developed:

[6] retail_eu: https://single-market-economy.ec.europa.eu/single-market/services/retail_en

To what extent do different Tree based machine learning models perform in forecasting Walmart's retail weekly sales?

As such, the sub-questions can be listed separately:

RQ1 *How do direct and recursive forecasting strategies affect the predictive performance of LGBM and Random Forest models predictive performance?*

RQ2 *To what extent do sliding and expanding window techniques affect the predictive performance of LGBM and Random Forest?*

3 RELATED WORK

Time series forecasting is a difficult analysis, especially when dealing with large hierarchical datasets, where the complexity and computational cost rise. Large retail chains sell a diverse range of products in a variety of locations, making high-quality predictions difficult and frequently inaccurate.

Several studies have used machine learning techniques for time series forecasting, with the goal of improving the prediction models' accuracy and efficiency. For instance, Effrosynidis et al. (2023) stated that machine learning models outperformed statistical models for forecasting in large-scale data, with tree-based algorithms accounting for three out of seven machine learning models. Another study that used historical data to estimate the number of confirmed cases of COVID-19 in the next two weeks found that the Extreme Gradient Boosting Machine (XGBM) and Light Gradient-Boosting Machine (LGBM) models outperformed other models, including statistical models (Radwan, 2021). Finally, the LGBM outperformed other models in the M5 Competition, which contained 42,840 hierarchical sales data from 10 Walmart stores (Makridakis et al., 2022a). Chakraborty et al. (2020) show that LGBM, NGBM, and XGBM are comparable and top performers when comparing six different machine learning algorithms. They also claim that those models are not affected by under- or over-fitting. LGBM is a gradient boosting algorithm that includes objective function regularization, reducing the likelihood of overfitting and increasing process speed. In the Makridakis competition, which had 7092 participants, the top 20% used machine learning models rather than statistical ones, and the top 50 competitors used LGBM, a decision tree-based ML approach that reportedly outperformed all other alternatives (Makridakis et al., 2022a).

Furthermore, another study comparing the Long Short-Term Memory (LSTM) model to the LGBM, focusing on Demand Forecasting of a Multinational Retail Company, concluded that the LGBM outperformed the

LSTM model (Saha et al., 2022). A recent study comparing XGBM and deep learning algorithms for tabular datasets found that XGBM and LGBM outperformed Tabular Neural Network (TabNet) on multiple datasets (Shwartz-Ziv & Armon, 2021), supporting another study's conclusion that deep learning algorithms for tabular data are still understudied (Arik & Pfister, 2021).

Demand forecasting techniques, notably recursive and direct strategies, play a pivotal role in enhancing predictive accuracy. Recursive strategies continuously refine forecasts as new data arrive, while direct strategies generate a final forecast using all historical data upfront. As exemplified in the study by Xue et al. (2019) about predicting next-day heat load curves, recursive strategies have demonstrated superiority over direct strategies in terms of accuracy, stability, and the overall modeling process.

In time-series forecasting, the selection between sliding and expanding windows is crucial. Sliding windows, employed in finance research (Bollerslev et al., 2018; Degiannakis & Filis, 2017; Ma et al., 2019), adapt to structural data changes by using recent observations for parameter estimation. Conversely, expanding windows, often used in macroeconomics (Gillitzer & McCarthy, 2019), incorporate all data for comprehensive analysis. A study on ARIMA models for short-term export forecasts across European countries demonstrates that sliding windows outperform expanding windows for immediate forecasts, indicating their relative efficacy in certain predictive scenarios (Lehmann, 2021).

Despite the widespread use of sliding and expanding windows alongside direct and recursive prediction methods in various forecasting applications, research on their applications in the retail industry remains limited. More studies are needed to investigate the effectiveness of these techniques in the specific context of retail sales forecasting.

This thesis fills a significant gap in the existing literature on time series forecasting in the broad area of Walmart sales. In addition, we plan to utilize the best-performing algorithms identified through related work in this domain, which are tree-based algorithms. The transformation of daily sales data into a weekly format is a unique feature of this study, as no study of the related dataset was conducted on weekly sales. This change speeds up the training process across the entire dataset, which includes sales data from multiple stores and states. This study distinguishes itself in the field of forecasting techniques by systematically investigating the performance of tree-based models using direct and recursive prediction methodologies. Furthermore, the study incorporates expanding and sliding window techniques into the tuning process, providing novel insights into the optimal combination of these techniques within tree-based algorithms. This research aims to fill a critical gap in understanding the most effective

forecasting methods tailored for large-scale retail datasets, by leveraging findings from the M5 Competition, including insights from esteemed participants contacted by Makridakis.

4 METHOD

4.1 *Experimental Setup*

The experimental procedure entails a methodically structured series of tasks for extracting insights and improving the accuracy of time series forecasting. These foundational datasets, include: sales.csv,' , 'calendar.csv,' and 'sell_price.csv,' lay the groundwork for subsequent analyses and model development. Following exploratory data analysis on daily sales patterns, data frames are preprocessed to handle missing values and ensure data integrity. Data frames are combined to form a comprehensive dataset, which allows for exploratory analysis of weekly sales patterns, revealing trends and variations. Feature engineering is a critical stage in which new variables are created to capture critical information, such as pricing, calendar features, and object features, using encoding techniques.

Furthermore, strategic dataset splitting ensures temporal integrity, allowing for more effective model training and evaluation. For subsequent model development, the dataset is divided into training, validation and test sets. LGBM and RF, two well-known machine learning algorithms, are implemented for adaptive learning using sliding and expanding window methodologies. The identification of the best parameters through rigorous tuning processes is critical for optimizing model performance. Model performance is assessed using key metrics such as RMSE and WRMSSE in both direct and recursive forecasting. The experimental steps are encapsulated in the Figure 1 concise flowchart, ensuring a systematic approach to time series forecasting. Each phase contributes to the development of robust models capable of accurately predicting complex sales dynamics.

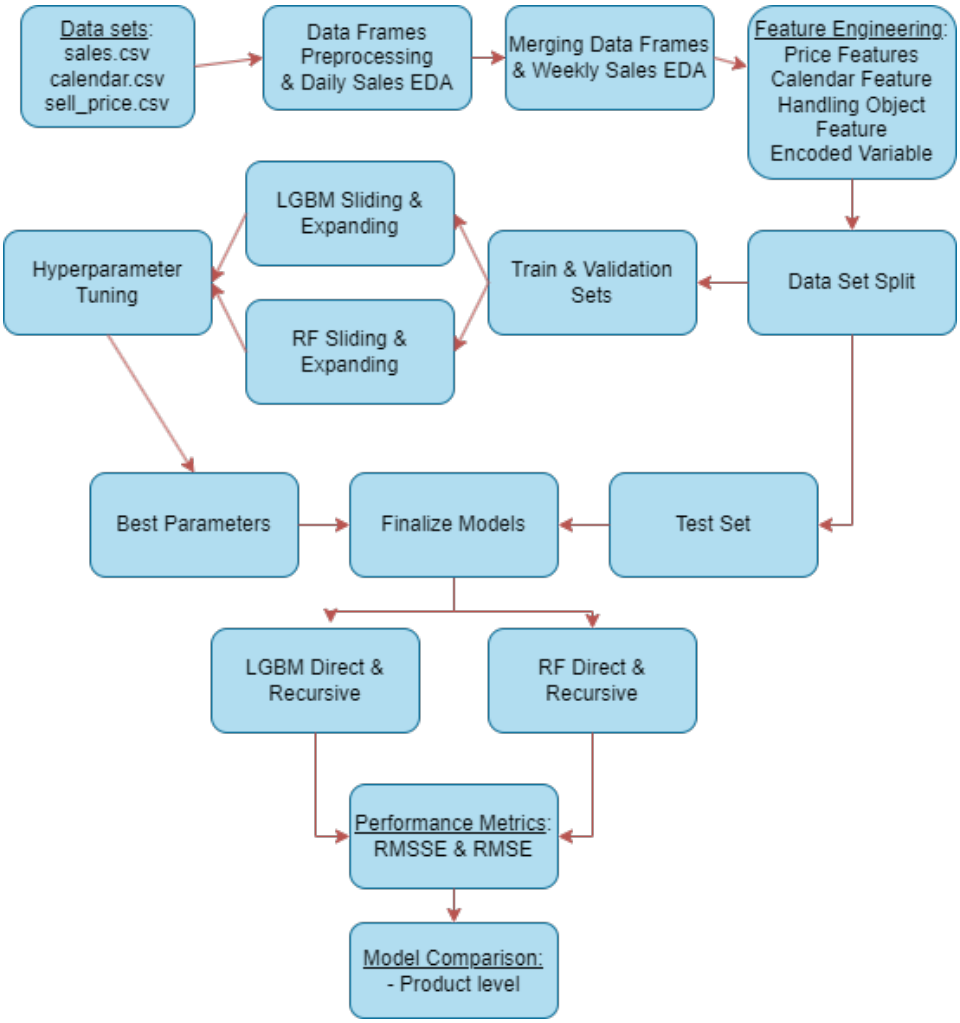

Figure 1: Flowchart of experimental procedure

4.2 *Dataset Description*

The M5 dataset is a rich collection of sales data from Walmart, the world's largest company by revenue [7]. It was used in the M5 Forecasting Competition conducted by the Makridakis Open Forecasting Center (MOFC) at the University of Nicosia. The main objective of the competition was to estimate or predict the unit sales of Walmart retail goods at stores in various locations for the next 28 days.

The data-set is organized in the form of grouped time series and involves the unit sales of various products sold in the USA. It includes several CSV files:

[7] Kaggle:https://www.kaggle.com/competitions/m5-forecasting-accuracy/data

- `calendar.csv`
- `sales.csv`
- `sell_prices.csv`

4.2.1 *Sales Data*

The Sales data set includes the historical unit sales of 3049 products, categorized into 3 product categories and 7 departments, sold in 10 stores in 3 states . The dataset has one column for each of the 1941 days from 2011-01-29 to 2016-05-22 (d_1 to d_1941.). The first 5 columns of the data set represent the 'item_id' (3049 products), 'state_id' (CA, TX, WI), cat_id' (Foods, Household, Hobbies),'store_id' (10 stores), and 'dept_id' (7 departments). The Sales data is available in Table 13, in Appendix A. For a more detailed description of the unique values of each variable, please refer to Table 14, in Appendix A.

4.2.2 *Calendar Data*

The Calendar data set has dates and promotion features. It also contains a primary key feature 'd' (Day number, d_1 to d_1941), which matches the Sales columns, and 'wm_yr_wk' (week number), which is the same in the Price data. Furthermore, the Calendar data set includes date related features and binary features indicating whether a specific day is applicable to promotion or not. The 'event_name_1' feature indicates if the specific day is Valentines day, Super Bowl day, Presidents day, etc. The 'event_name_2' feature indicates if the specific day is Father's day, Easter, Cinco De Mayo, etc. The 'event_type_1' and 'event_type_2' features indicate if those events are Religious, National or Cultural. The binary features 'snap_CA', 'snap_TX' and 'snap_WI' correspond to the Supplemental Nutrition Assistance Program (SNAP), a federal assistance program providing funds for eligible individuals and families to purchase food. The Calendar data is available in Table 15, in Appendix A.

4.2.3 *Price Data*

The Price data set has 'store_id', 'item_id', 'wm_yr_wk', and 'sell_price' features, which contain the weekly price of each product and store, and the first three serve as primary key for the future merging with the other data sets. The Price data set doesn't include product prices for every week, meaning that in weeks where there is no price, the specific product did not have any sales. The Price data is available in the Table 16, in Appendix A.

4.3 *Data Preprocessing*

The preprocessing steps performed on the M5 Walmart Sales Forecasting data set aim to transform raw data into a structured format suitable for model training. The list of the software, and packages used, is provided in Table 17 Appendix A. The following description outlines the key preprocessing steps:

4.3.1 *Melting and Aggregation*

The first stage of preprocessing entails melting the sales training data to convert it from a wide to a long format. This step is accomplished by melting the data using the identifier variables 'id', 'item_id', 'dept_id', 'cat_id','store_id', 'state_id', and the remaining columns as melted values representing the unit sales ('sold') for each day ('d'); the result is 58.327.370 rows in total. After adding the 'wm_yr_wk' from the calendar, the melted data are aggregated on a weekly basis.csv based on 'id', totaling unit sales for each unique combination of 'id', 'item_id', 'dept_id', 'cat_id','store_id','state_id', and 'wm_yr_wk' features. After aggregation, the total number of rows is 6.688.671, allowing the models to be trained at every level of hierarchy.

A critical transformation in the preprocessing phase is aggregating the initial daily sales data into weekly sales. This strategic conversion improves computational efficiency while retaining the essence of temporal patterns. The data set's granularity is effectively reduced by adding the weekly unit sales for each unique combination of product, store, and state. This not only makes the data set more manageable, but it also allows for a comprehensive model to capture sales patterns for each store, state, and product category. Because there are seven days in a week, the computational load is significantly reduced, potentially reducing the required computational power by a factor of seven. This transformation optimally aligns the data set with the forecasting task's periodicity, ensuring efficient resource utilization in the subsequent modeling process.

4.3.2 *Processing Null Values*

The Prices data set, as mentioned in the methodology, is missing pricing information for a few weeks. To address this, rows in the sales data set were removed that did have corresponding null values in the price data set. This step was critical because the models used in this study cannot handle missing values. Those were unofficially sold products that contributed insignificantly to the data set. This arrangement recognizes

the ever-changing retail landscape in which products may be introduced or discontinued.

The Prices data set, as outlined in the methodology, exhibited a notable proportion of null values before the aggregation of weekly sales, amounting to 58%. Post-aggregation, a substantial reduction to 23% in the prevalence of null values was observed. Additionally, it is noteworthy that the aggregation process effectively addressed the challenge of numerous continuous days with zero sales for various products. This refinement is reflected in Figures 2 and 3, which visually demonstrate the minimization of this issue by showing the overall demand of a single product. It is imperative to highlight that the removal of rows without corresponding non-null values in the price data set was a crucial preprocessing step, considering the incompatibility of the models used with missing values.

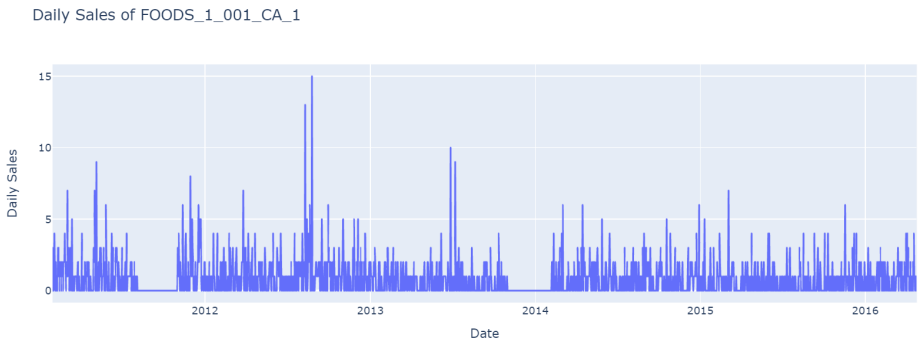

Figure 2: Daily Unit Sales of FOODS_1_001_CA_1

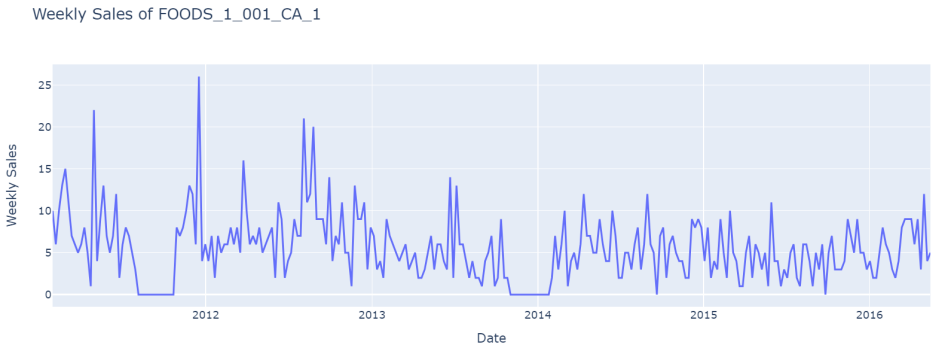

Figure 3: Weekly Unit Sales of FOODS_1_001_CA_1

The remainder of Calendar's data are incorporated into the data set to provide a temporal context. On the 'wm_yr_wk' column, the weekly aggregated sales data is combined with the calendar information. This integration contains information such as the date and the following:

4.3.3 *Calendar Features Extraction*

To enrich the data set, additional features are extracted from the calendar data. Event types (event_type_1 and event_type_2) are used to generate new columns that represent the number of occurrences for each event type per week. This step refines the data set by capturing the temporal patterns associated with events.

4.3.4 *Snap Counts Calculation*

The calendar 's' snap_CA', 'snap_TX', and 'snap_WI' columns are converted to binary format, and the number of occurrences of '1' is calculated per week. These totals represent the number of days in each week with specific snap conditions.

4.4 *Feature Engineering*

4.4.1 *Generation of Time Related Features*

The introduction of innovative features aims to enrich the model's ability to discern patterns within the dataset. Among these features are indicators for the first or last week of the month, along with extracted date components such as week, month, quarter, year and season, based on specific seasonal trends.

Date-time decomposition is applied to derive 'week,' 'month,' and 'year' features, enabling the model to capture weekly, monthly, and yearly seasonality and trends. Binary features 'is_month_end' and 'is_month_start' are incorporated. An analysis of monthly average sales reveals increased demand during specific months, such as November, December, and summer, as shown in Figure 4. This increase in demand is likely attributed to a surge in people shopping for gifts during the winter holiday weeks. To address these seasonality effects, a 'season' feature is introduced. Moreover, Figure 5 encapsulates the aggregate weekly demand spanning the years 2011 to 2016, unveiling significant and notable seasonal patterns within each annual cycle.

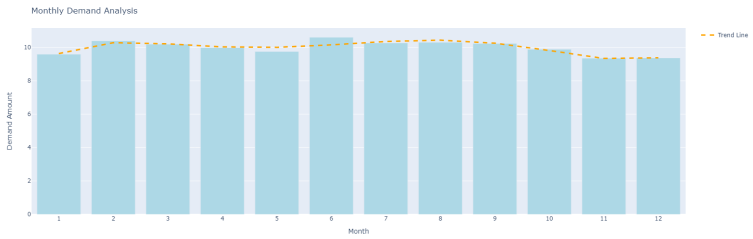

Figure 4: Monthly Overall Demand

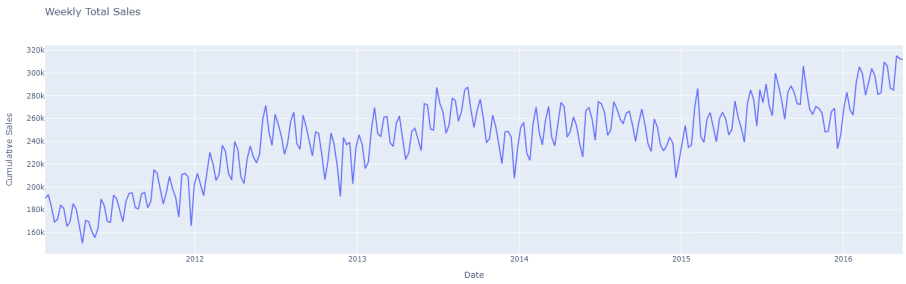

Figure 5: Total Weekly Demand

The months are organized into distinct seasons. Spring (1) encompasses March, April, and May. Summer (2) spans June, July, and August. Autumn (3) includes September, October, and November. Winter (4) covers December, January, and February. Despite the absence of significant trends between quarters in aggregate sales, a 'quarter' feature is added to account for potential meaningful trends in (Product-Store) time series (see Figure 18 in Appendix A).

4.4.2 *Demand Related Lag & Rolling Features*

This study focuses on identifying principal features for forecasting weekly unit sales, employing statistical methods to enhance predictive accuracy. The Augmented Dickey-Fuller (ADF) test assesses sales data stationarity, a prerequisite for Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) application. Statistical analysis reveals 99% stationarity, indicating consistent behavioral patterns.

Having established the stationarity of the data, the ACF was then utilized to evaluate the correlation between the time series and its past values at varying time lags. The generated correlation coefficients, within the range of -1 to 1, elucidated the strength and orientation of the relationship. The ACF diagram provided a comprehensive view of how correlation strength evolves over different lags, offering insights into the enduring

patterns within the time series. PACF extends the analysis, measuring correlation with lagged versions, eliminating intermediate lag influences to discern immediate impacts (Ensafi et al., 2022).

ACF and PACF identify significant lag values (1, 2, 3, 4) capturing relevant patterns as shown in Figure 6. These values enhance subsequent time series forecasting models, based in sales data structure.

Recognizing the lag features' fundamental role, we extend our methodology to include rolling mean, standard deviation, maximum, and minimum at different time spans (2, 3, 4, 5). These rolling statistics capture evolving trends, providing a smoothed representation to discern patterns and outliers impacting accuracy.

The choice of rolling statistics stems from their ability to encapsulate different aspects of time series dynamics. The rolling mean indicates the general trend, standard deviation quantifies variability, and maximum/minimum values highlight extremes and anomalies.

By integrating lag features and rolling statistics, we have created a nuanced feature set contributing to a comprehensive understanding of temporal dependencies within weekly unit sales data. This approach strengthens predictive models, providing a robust feature set capable of capturing intricate patterns in dynamic sales environments. Lag and rolling features are presented in Table 18, Appendix B.

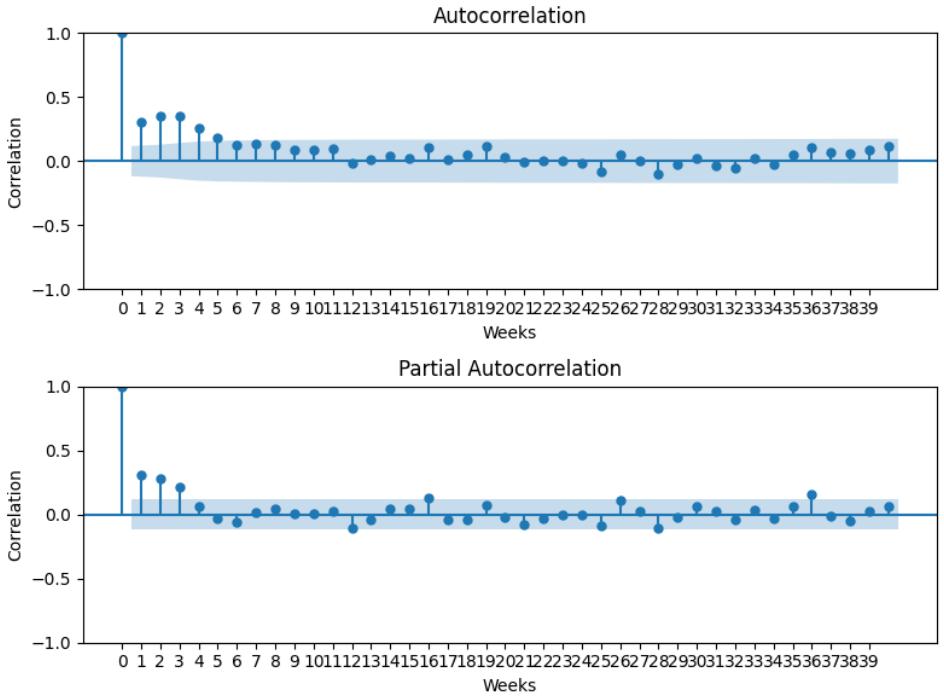

Figure 6: ACF & PACF Diagrams of weekly demand

4.4.3 *Price-related Features*

In order to improve our time series forecasting models, we introduced a set of carefully chosen price-related features, each selected for its unique contribution to capturing distinct aspects of pricing dynamics. Among the features are:

- Maximum Store and Daily Prices (price_store_max, price_day_max): capture the highest observed prices at the store and daily levels, revealing potential outliers and extreme fluctuations influencing consumer behavior.
- Overall Maximum and Minimum Prices (price_max, price_min): represent global price extremes across all stores and days, assisting in the identification of broader pricing trends and periods of exceptional pricing.
- Price Variability and Central Tendency (price_std, price_mean): this metric measures price variability and average pricing, providing valuable insights into price distribution and volatility.
- Normalized Price (price_norm): enabling comparison by normalizing prices relative to their mean or another reference point, ensuring that relative changes rather than absolute levels are prioritized.
- Price Momentum (price_momentum, price_momentum_m): identify trends and directional changes in pricing by capturing the rate of change in prices over time.

These characteristics, taken together, improve our model's ability to discern intricate relationships between pricing dynamics and sales patterns, resulting in a more robust forecasting framework. Table 19, in Appendix B shows the price features created.

4.4.4 *Feature Transformation and Encoding*

The inherent ability of decision tree-based algorithms, such as LGBM to recursively split data based on feature sets and threshold values eliminates the need for scaling features. In essence, the algorithm adjusts its thresholds during the splitting process, making the original feature scale irrelevant. As part of the preprocessing steps, the categorical variables 'event_type_1_count', 'event_type_2_count','snap_CA','snap_TX', and 'snap_WI' were transformed into numerical representations. This conversion is required for model compatibility, as it ensures that the categorical

nature of these features is translated into an algorithmic-processable format.

In the case of the Random Forest algorithm, a distinct challenge arises due to its requirement for categorical features to be encoded into numerical values. Given the computational complexities associated with alternative encoding methods, such as one-hot encoding, a pragmatic approach was adopted. Specifically, 'id,' 'item_id,' 'store_id,' and 'state_id' features underwent label encoding. Notably, the 'id' variable alone encompasses 3049 unique values, emphasizing the computational burden that alternatives like one-hot encoding could impose, particularly in scenarios with a considerable number of categories. On the other hand, mean encoding was avoided due to its overfitting risk and potential for data leakage, particularly with categorical variables. Label encoding emerged as a practical solution that facilitated model compatibility, while mitigating the computational burden inherent in handling a large number of unique categories.

Finally, the feature transformation methods were carefully chosen to align with the inherent requirements of the specific algorithms used. The encoding decision was made with the characteristics of each algorithm in mind, ensuring optimal model compatibility and performance.

4.5 *Data Types Optimization*

Finally, a memory reduction function is provided to optimize the memory usage of a Pandas DataFrame, which is especially useful for large datasets such as the Walmart sales dataset. Based on observed data ranges, the function dynamically adjusts numeric column types to the smallest suitable types (for example, from int64 to int16). This results in a more memory-efficient representation of the data, which contributes to lower computational overhead and better performance in subsequent analyses or machine learning tasks. This optimization is consistent with best practices for managing large datasets, such as those used in Walmart sales forecasting.

4.6 *Custom Split for Hyperparameter Tuning: Facilitating Sliding and Expanding Windows*

The specific requirement to implement sliding and expanding window methodologies necessitated a departure from the traditional time series split during the hyperparameter tuning phase. The traditional time series split follows a strict chronological order, with the validation set coming after the training set. However, the need to incorporate sliding and expanding

windows for tuning purposes necessitated a customized approach to better simulate real-world forecasting scenarios.

The primary reason for the custom split was to make it easier to implement sliding and expanding window methodologies. Sliding windows enable a dynamic assessment of model performance by sequentially moving through the dataset, whereas expanding windows include more data for training progressively, allowing for a comprehensive evaluation. Designed for sliding and expanding windows to closely align with the forecasted period, the Custom Split is presented in Table 1. This configuration includes distinct validation sets (v1, v2, v3, v4), contributing to an enhanced evaluation of the model's performance. . During the model evaluation phase, this deliberate design aimed to create a more realistic simulation of the forecasting scenario.

Set	**Week (wm_yr_wk)**
Training Set	11106–11608
v1	11609
v2	11610
v3	11611
v4	11612
p1	11613
p2	11614
p3	11615
p4	11616

Table 1: Custom Data Split

Traditional time series splits may not capture the dynamics of the forecasting task effectively, especially when sliding and expanding windows are used in the evaluation. In scenarios where the goal is to predict future weeks, the custom split strategy allowed for a more relevant assessment of model performance. The custom split attempted to bridge the gap between the rigidity of chronological splits and the dynamic requirements posed by the task's forecasting nature, by embracing sliding and expanding windows. This method attempted to align model evaluation with evolving data patterns.

From the creation of the final training set (final training) to the start of the validation periods, the models were exposed to a wide range of temporal patterns. This exposure aided in the generalization and robust learning required for forecasting tasks. The advantages of the custom split stemmed from its ability to accommodate the complexities of sliding and expanding window methodologies, thereby increasing the relevance of model evaluations in the context of real-world forecasting scenarios.

4.6.1 *Sliding Window Approach*

The sliding window methodology takes a dynamic approach, training the forecasting model iteratively with a rolling window of historical data. This enables the model to adapt to changing patterns considering a subset of the data at each iteration. The algorithm starts with a training set and gradually slides the window forward, updating the model parameters and fine-tuning hyper parameters as it goes. Each prediction task entails training the model with the appropriate sliding window and adjusting parameters to achieve the best forecasting accuracy. The sliding window approach is shown in the Figure 7.

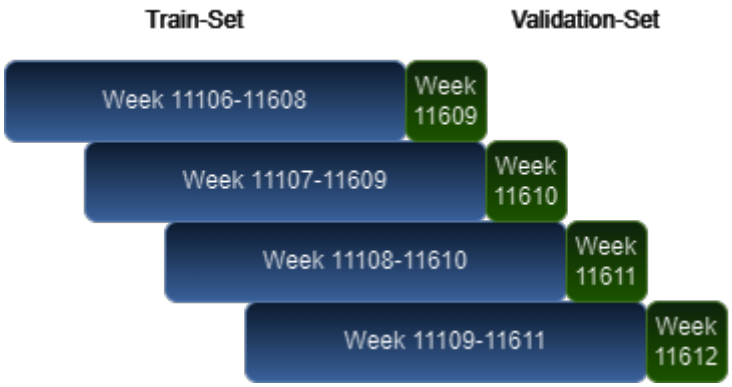

Figure 7: Sliding Window Method

4.6.2 *Expanding Window Approach*

The expanding window method takes a cumulative approach, gradually expanding the training set to include all historical data up to the current time point. This captures the evolving nature of time series data considering all previous observations. The algorithm starts with a small training set and expands iteratively to include the next data point. The updated set is then used to train the model, and the hyper parameters are fine-tuned accordingly. The model thus trained on the entire expanding training set is used for prediction tasks and hyper parameters can be adjusted for each prediction task. The expanding window approach is show in the Figure 8.

Figure 8: Expanding Window Method

4.7 *Algorithms and Software*

This section elucidates the machine learning models and forecasting techniques employed in the study. Initially, recursive and direct techniques were introduced, paving the way for a detailed exploration of the Random Forest (RF) and LGBM regression algorithms, which constitute integral components of the analytical framework applied in this research effort. Lastly, Optuna is described as the tool of hyperparametring tunning of both models. The list of the software and packages used, is provided in Table 17, Appendix A.

4.7.1 *Recursive Prediction*

The forecasting process in the recursive prediction strategy involves predicting one-time step at a time, while incorporating the predicted values into the input features for subsequent predictions. The prediction iteration then proceeds by predicting the next time step, updating input features with the predicted value and repeating the circle until the entire prediction horizon has been covered. A more detailed description can be found in Figure 9.

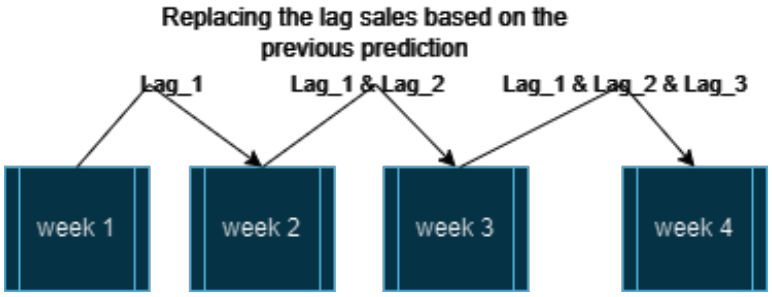

Figure 9: Detailed recursive methodology

4.7.2 *Direct Prediction*

Direct prediction, in contrast to the recursive strategy, entails training a single model to predict all time steps of the desired horizon at the same time. The algorithm begins with a complete training set that does not include the prediction horizon. The prediction phase is straightforward, with the trained model predicting all time steps at once. The evaluation process assesses forecasting accuracy across the entire prediction horizon, providing insights into the model's ability to capture dependencies across multiple time steps at the same time.

4.7.3 *Moving Average*

The moving average model, widely adopted for time series forecasting, predicts future values by averaging past values within a specified window, effectively smoothing short-term fluctuations and emphasizing longer-term trends. This choice aligns with a study by Cerqueira et al. (2020), comparing performance estimation methods for time series forecasting. The authors used a simple moving average model with a four-week window as a baseline to assess advanced models like ARIMA, ETS, and TBATS. Their findings indicated that while the moving average model is often outperformed by sophisticated models, especially in non-stationary time series, it serves as a valuable reference, highlighting both the strengths and limitations of the moving average model in forecasting scenarios.

In this study, we treated the window size as a hyperparameter, systematically exploring various sizes to determine the optimal configuration. Our analysis revealed that the moving average model achieved the lowest WRMSSE and RMSE when employing a four-week window. This observation led us to select the four-week window, as it demonstrated superior performance in capturing patterns within the time series data. Our findings, summarized in Table 2, indicated that the lowest WRMSSE and RMSE were achieved with a four-week window for the moving average. This dis-

covery suggests that a four-week window size effectively captures patterns in the time series data.

Window Size	Average RMSE	Average WRMSSE
3	7.815 10	0.255 23
4	7.796 74	0.256 60
5	7.902 99	0.256 59
6	8.092 09	0.259 66
7	8.280 32	0.262 68

Table 2: Average RMSE and RMSSE for Different Window Sizes

4.7.4 *Light Gradient Boosting Machine (LGBM)*

LGBM is a gradient boosting framework for distributed and efficient decision tree model training. It was created by Microsoft in 2017 and has gained popularity due to its speed and scalability, especially on large data sets. LGBM uses a leaf-wise tree growth strategy rather than the traditional level-wise strategy, which allows for faster training times. This method reduces the number of nodes in the tree, resulting in less memory usage and greater efficiency. LGBM is well-suited for tasks like regression, classification and ranking, making it a versatile machine learning tool.

Distinguishing itself from the XGBoost model which was invented by Chen and Guestrin (2016), LGBM employs histogram-based algorithms to expedite the training process, diminish memory usage, and adopt a leaf-wise growth strategy within specified depth constraints. The fundamental concept behind the histogram algorithm involves discretizing continuous floating-point eigenvalues into k bins and constructing a histogram with a width of k. Unlike methods requiring additional storage for pre-sorted results, the histogram algorithm eliminates this need. Furthermore, it preserves the value post-discretization of features, typically stored with an 8-bit integer, thereby reducing memory consumption to 1/8 of the original. Despite this coarse partitioning, there is no compromise in the model's accuracy (Ke et al., 2017a).

The data preprocessing phase preceding the application of LGBM entails converting object type features to categorical. The primary machine learning model used in this research is LGBM, which was chosen for its efficiency, speed, low memory usage and top performance in the the M5 competition (Makridakis et al., 2022b). The model will be used in a multi-step direct and recursive forecast and will be tuned based on the expanding and the sliding techniques.

4.7.5 *Random Forest Regression*

Random Forest(RF), initially proposed by Breiman (Breiman, 2001), is a powerful ensemble-learning algorithm with applications in both classification and regression tasks. This method entails assembling a collection of distinct classification and regression decision trees to form a comprehensive decision forest ensemble. The algorithm applies hierarchically organized conditions from root to the leaf, representing the regression function, in the context of regression.

Importantly, the Random Forest method generates trees at each node separation using bootstrap samples and randomly chosen *m features*. The *m features* ensure diversity within the ensemble by being significantly smaller than the total number of features. During the regression process, decision trees created with the Random Forest remain unpruned and continually divide until only a few units remain in the leaf node.

Random Forest distinguishes itself among machine learning techniques by combining validity estimation and model interpretation. The use of random sampling, combined with improved ensemble method properties, results in better generalizations and valid estimates. Karasu and Altan (2019) provide additional details on the Random Forest algorithm for a comprehensive understanding.

4.7.6 *Hyperparameter Tuning of LGBM and RF Using Optuna*

Optuna is a Python hyper parameter optimization library that uses a versatile and efficient Bayesian optimization algorithm. This library employs a probabilistic model to forecast the behavior of the objective function in hyper parameter space, allowing it to make informed decisions about where to look for the next set of hyper parameters. The Tree-structured Parzen Estimator (TPE) algorithm is used by Optuna, a Bayesian optimization technique that efficiently balances exploration and exploitation. Optuna, in particular, provides a user-friendly and extensible interface for defining hyper parameter search spaces and running optimization experiments. Its use in hyper parameter tuning ensures that machine learning models such as LightGBM and Random Forest are fine-tuned for optimal performance. The systematic exploration of hyper parameter space is critical for balancing model complexity and predictive accuracy, which contributes to the overall effectiveness of the forecasting methodology (Akiba et al., 2019b).

For the hyperparameter tuning process, we utilize an expanding window and a sliding window split in every iteration. This approach involves manually performing the data split, ensuring that the model is trained and validated on distinct subsets of the dataset in each iteration. This nuanced strategy contributes to a robust evaluation of the model's perfor-

mance under varying conditions, thereby enhancing the reliability of the hyperparameter optimization process.

4.7.7 *Hyperparameters for LGBM*

LGBM, as a gradient boosting framework, demands meticulous tuning of hyperparameters to attain optimal performance. The chosen hyperparameters and their rationale combine to create a model that balances accuracy and convergence speed. The learning rate strikes a balance between model accuracy and convergence speed. A moderate rate ensures efficient convergence while minimizing the risk of overshooting. The number of estimators is chosen to avoid overfitting and underfitting while capturing patterns without adding unnecessary complexity. The tree learner's 'data' setting corresponds to data set characteristics, balancing computational efficiency and model performance (Ding et al., 2021). Considerations for modeling specific target variable characteristics, such as financial data or insurance claims, are reflected in Tweedie variance power (He Zhou & Yang, 2022). A balanced number of leaves prevents over fitting by capturing enough information without going into unnecessary detail. The max bin parameter strikes a balance between granularity and computational efficiency, which is critical for dealing with outliers. The addition of bagging fraction and frequency introduces randomness, which improves model robustness without sacrificing stability. The feature fraction balances the model's feature diversity without overly constraining it. Min data in leaf prevents overspecification, by balancing granularity and generalization. Regularization is provided by the min sum hessian in the leaf, which ensures model flexibility without sacrificing stability. This methodology takes a comprehensive approach, considering into account factors such as convergence speed, model complexity and data set characteristics (Ke et al., 2017a). In Tables 3, 4 and 5 the selected parameter ranges are specified, accompanied by their optimal values following the tuning procedure, using the sliding and expanding techniques.

Table 3: LightGBM Hyperparameter Tuning

Hyperparameter	**Search Space**
"objective"	["tweedie", 'regression']
"random_state"	42
"learning_rate"	[0.01, 0.3]
"n_estimators"	[3, 10]
"metric"	"rmse"
"boosting_type"	"gbdt"
"tree_learner"	["feature", "data"]
"tweedie_variance_power"	[1.1, 1.2, 1.4, 1.7, 1.8]
"num_leaves"	[5, 20]
"max_bin"	[50, 100]
"bagging_fraction"	[0.4, 0.9]
"bagging_freq"	[1, 10]
"feature_fraction"	[0.4, 0.9]
"min_data_in_leaf"	[2, 16]
"min_sum_hessian_in_leaf"	[1, 10]

Table 4: LGBM Hyper Parameters - Sliding Window

Hyper Parameters	**Value**
Learning Rate	0.272528
N Estimators	9
Tree Learner	'data'
Tweedie Variance Power	1.2
Num Leaves	14
Max Bin	70
Bagging Fraction	0.753449
Bagging Freq	2
Feature Fraction	0.669199
Min Data in Leaf	6
Min Sum Hessian in Leaf	9

Table 5: LGBM Hyper Parameters - Expanding Window

Hyper Parameters	**Value**
Learning Rate	0.28097707291851126
N Estimators	9
Tree Learner	'data'
Tweedie Variance Power	1.2
Num Leaves	18
Max Bin	95
Bagging Fraction	0.5439292388995909
Bagging Freq	8
Feature Fraction	0.6407865116499937
Min Data in Leaf	7
Min Sum Hessian in Leaf	6

4.7.8 *Hyperparameters for Random Forest*

The hyper parameters used to train the Random Forest model were carefully chosen to strike a balance between model complexity and generalization performance. The reasoning behind each parameter choice is outlined below.

To balance computational efficiency and model robustness, the number of estimators, which represents the number of decision trees in the forest, was chosen moderately. This decision is consistent with the principle that a large number of estimators may result in diminishing returns, without significantly improving predictive accuracy. Setting a relatively larger maximum depth for each decision tree aims to allow the trees to capture intricate patterns in the data, potentially improving the model's ability to represent complex relationships. Higher values were assigned to the parameters controlling the minimum number of samples required to split an internal node (min_samples_split) and the minimum number of samples required to be at a leaf node (min_samples_leaf). This option is intended to prevent overfitting to noise in the training data by requiring a higher degree of generalization, thereby promoting model robustness. The max_features parameter 'sqrt' setting encourages diversity among individual trees, by limiting the maximum number of features considered for splitting a node. This constraint prevents the model from being overly influenced by specific features, resulting in better generalization (Liaw & Wiener, 2001). Because bootstrap sampling is disabled (boot-strap=False), each tree is trained on the entire data set. This option increases tree diversity. while lowering the risk of overfitting, by preventing the model from relying too heavily on specific subsets of data (Bilolikar et al., 2023).

Tables 6, 7 and 8 specify the selected parameter ranges, accompanied by their optimal values following the tuning procedure using the sliding and expanding techniques.

Table 6: Random Forest Hyperparameter Tuning

Hyperparameter	**Search Space**
`n_estimators`	[3, 10]
`max_depth`	[5, 20]
`min_samples_split`	[2, 16]
`min_samples_leaf`	[1, 10]
`max_features`	["sqrt", "log2"]
`bootstrap`	[True, False]
`random_state`	42

Table 7: Random Forest Hyper Parameters - Sliding Window

Hyper Parameters	**Value**
`n_estimators`	7
`max_depth`	19
`min_samples_split`	4
`min_samples_leaf`	8
`max_features`	'sqrt'
`bootstrap`	False

Table 8: Random Forest Hyper Parameters - Expanding Window

Hyper parameters	**Value**
`n_estimators`	8
`max_depth`	15
`min_samples_split`	12
`min_samples_leaf`	9
`max_features`	`'sqrt'`
`bootstrap`	`False`

4.8 *Evaluation Method*

Various measures were employed in previous studies to evaluate point forecast accuracy, with the M5 "Accuracy" competition primarily utilizing the root mean squared scaled error (RMSSE). As a variant of the mean absolute

scaled error (MASE), RMSSE is independent of data scale, has predictable behavior and symmetrically penalizes both positive and negative forecast errors (Makridakis et al., 2022a). It overcomes challenges associated with sporadic unit sales and zero values, thus providing a robust evaluation for series with intermittent demand patterns (Davydenko & Fildes, 2013; Prestwich et al., 2014).

The weighted RMSSE (WRMSSE), calculated by averaging RMSSE scores across all series with appropriate weights based on cumulative actual unit sales, serves as a measure of overall forecast accuracy. It considers unit sales, selling volumes and prices hierarchically, aiming to identify forecasting methods suitable for accurately predicting series with higher revenues (Makridakis et al., 2022a). The equal weighting of all aggregation levels in WRMSSE aligns with the competition's emphasis on comprehensive evaluation, rather than addressing specific decision-making problems.

RMSE (Root Mean Squared Error) measures a model's accuracy in comparison to a naive forecast. It was chosen for its ability to reduce the impact of zero sales days, which is especially important for time series forecasting in retail settings (Makridakis et al., 2022a). While RMSSE provides a comprehensive evaluation of forecast accuracy, RMSE adds value by addressing specific challenges associated with zero sales days. These metrics, when combined, provide a solid assessment framework for forecasting methods in the context of the M5 competition.

5 RESULTS

The time series forecasting experiments on historical sales data from Walmart utilized two prominent machine learning models: LGBM and RF. Predictions were generated using both direct and recursive approaches, after being tuned with sliding and expanding window techniques and the forecasting performance was assessed through RMSE and WRMSSE metrics. Detailed results can be found in Tables 9 & 10.

5.1 *Root Mean Squared Error (RMSE) Analysis*

In the comprehensive RMSE analysis, both RF and LGBM models consistently outperformed the baseline moving average model. Specifically, RF models, including RF Sliding Direct and RF Expanding Direct, exhibited notable superiority over LGBM models. Across all forecasting weeks, RF Sliding Direct and RF Expanding Direct consistently demonstrated lower RMSE values, surpassing their LGBM counterparts, highlighting the effec-

Table 9: RMSE

Model	Week 1	Week 2	Week 3	Week 4	Average
Base Line MA	7.49882	8.31027	8.22504	7.15285	7.79674
LGBM Sliding Direct	6.71488	7.81394	7.65157	7.12367	7.32602
LGBM Sliding Recursive	6.71488	7.83352	7.96409	7.29677	7.45232
LGBM Expanding Direct	6.64904	7.91746	7.54046	7.17658	7.32089
LGBM Expanding Recursive	6.64904	7.87266	7.71498	7.21667	7.36334
RF Sliding Direct	6.46578	7.23178	7.04479	6.84074	6.89577
RF Sliding Recursive	7.49907	8.31027	8.22504	7.152285	7.79681
RF Expanding Direct	6.36794	7.38401	7.00152	6.80354	6.88925
RF Expanding Recursive	6.34998	8.21201	8.04721	7.20548	7.79681

Table 10: WRMSSE

Model	Week 1	Week 2	Week 3	Week 4	Average
Base Line MA	0.25874	0.26253	0.25484	0.25049	0.25660
LGBM Sliding Direct	0.23169	0.24685	0.23708	0.24947	0.24133
LGBM Sliding Recursive	0.39245	0.42377	0.43973	0.41504	0.41779
LGBM Expanding Direct	0.22942	0.25012	0.23363	0.25132	0.24119
LGBM Expanding Recursive	0.38860	0.42589	0.42598	0.41049	0.41277
RF Sliding Direct	0.22310	0.22846	0.21827	0.23956	0.22742
RF Sliding Recursive	0.37146	0.45578	0.47841	0.43162	0.43443
RF Expanding Direct	0.21972	0.23327	0.21693	0.23826	0.22711
RF Expanding Recursive	0.37112	0.44424	0.44432	0.40985	0.41742

tiveness of the direct approach within RF models for superior predictive performance.

The percentage differences in average RMSE values between RF and LGBM models further highlighted this superiority. Specifically, RF Sliding Direct exhibited a 5.87% lower average RMSE (6.8925), when compared to LGBM Sliding Direct (7.32601) and RF Expanding Direct showed a 5.89% lower average RMSE compared to LGBM Expanding Direct. Additionally, it is noteworthy that both models demonstrated the smoothest and lowest distribution of errors in the direct method when compared to the recursive approach. However, RF's performance in the direct method stood out as superior, underscoring its efficacy in minimizing errors in direct forecasting scenarios.

Examining the week-to-week variation, it is noteworthy that, in the recursive approach, as shown in Table 11, both the LGBM and RF models demonstrated significant increases in RMSE values, suggesting potential challenges in adapting to evolving patterns. Specifically, in the case of LGBM Sliding Recursive, the RMSE increased by 16.65% after the first week, followed by additional increases of 1.66% in the third week and a decrease of 8.37% in the last week. Similarly, the RF Sliding Recursive exhibited a 10.81% increase after the first week, followed by additional decreases of 1.02% and 13.03%. Furthermore, in the second week of our analysis, a significant uptrend in actual sales occurred, with an increase of 10.53%. Notably, during this period, RF demonstrated a superior ability to capture and adapt to this uptrend as compared to LGBM. RF exhibited a more accurate prediction that aligned closely with the observed increase in actual sales, thus emphasizing its robust performance in responding to immediate changes in sales patterns. Subsequently, Figures 10 and 11 illustrate the observed trends in the models' performance over consecutive weeks.

Models	**Week 1**	**Week 1-2**	**Week 2-3**	**Week 3-4**
Base Line MA	7.49881	10.82%	-1.02%	-13.03%
LGBM Sliding Direct	6.71488	16.36%	-2.07%	-6.89%
LGBM Sliding Recursive	6.71488	16.65%	1.66%	-8.37%
LGBM Expanding Direct	6.64904	19.07%	-4.76%	-4.82%
LGBM Expanding Recursive	6.64904	18.40%	-2.00%	-6.45%
RF Sliding Direct	6.46577	11.84%	-2.58%	-2.89%
RF Sliding Recursive	7.49906	10.81%	-1.02%	13.03%
RF Expanding Direct	6.36794	15.95%	-5.17%	-2.82%
RF Expanding Recursive	6.34998	29.32%	-2.00%	-10.45%

Table 11: Weekly Percentage Change in RMSE

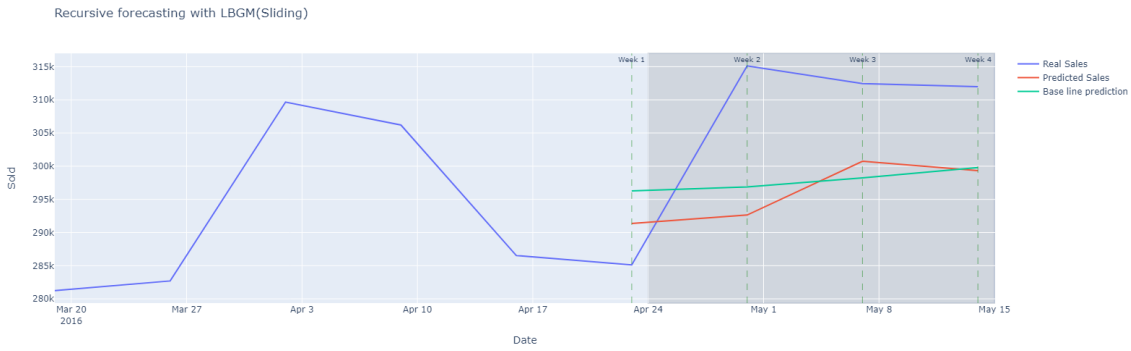

Figure 10: Recursive forecasting with LGBM(Sliding)

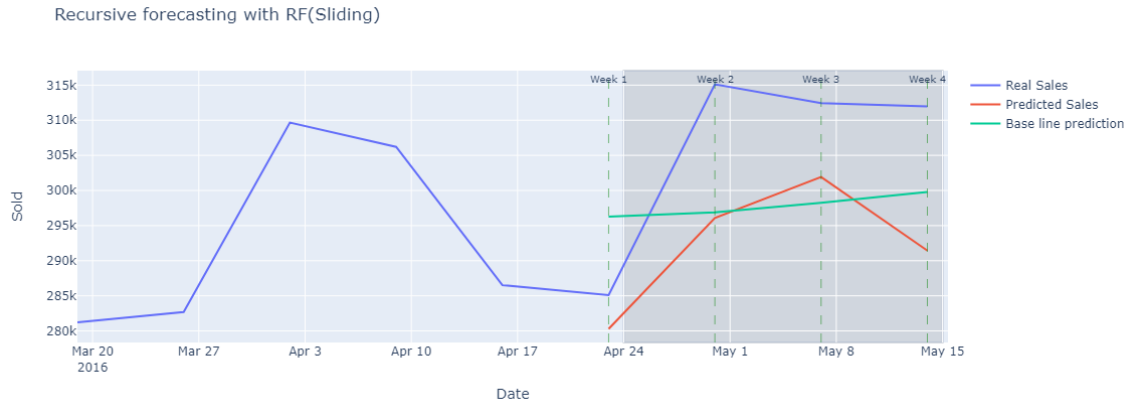

Figure 11: Recursive forecasting with RF(Sliding)

For the expanding window approach, LGBM Expanding Recursive exhibited an initial increase of 18.40% in the first week, followed by more modest decreases of 2.0% and 6.45%. In contrast, RF Expanding Recursive demonstrated the largest increase of 29.32% after the first week, making it not only the most substantial increase but also the highest absolute error observed, reaching levels comparable to the Baseline Moving Average. Subsequent decreases of 2.00% and 10.45% followed. This nuanced analysis sheds light on the challenges associated with the RF recursive approach, particularly in capturing evolving patterns over successive forecasting periods. Figures 12 and 13 visually present the observed trends in the models' performance over consecutive weeks.

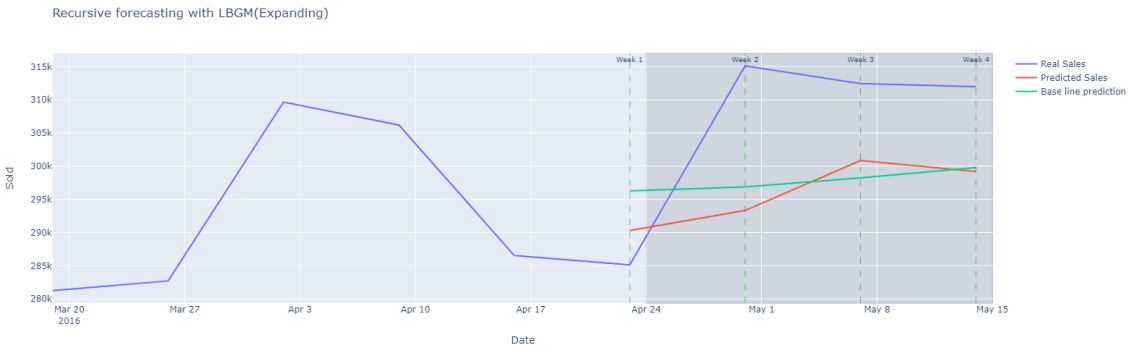

Figure 12: Recursive forecasting with LGBM(Expanding)

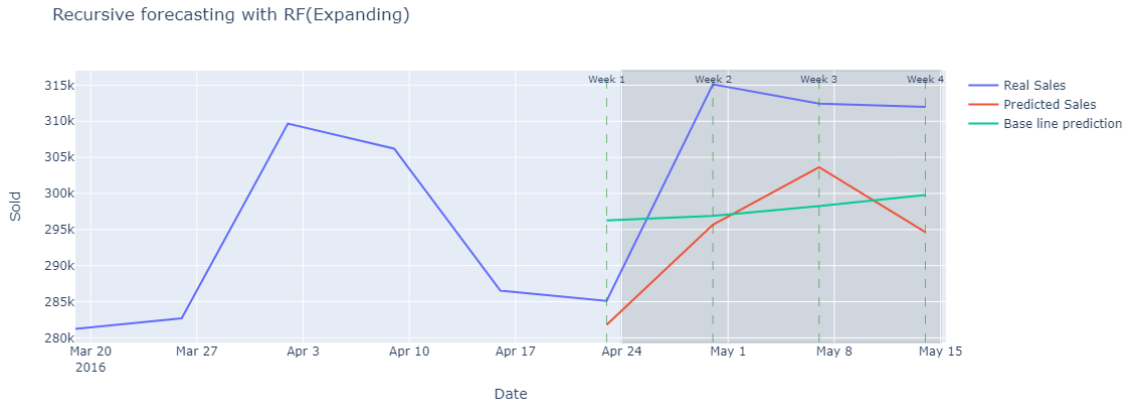

Figure 13: Recursive forecasting with RF(Expanding)

Conversely, LGBM models revealed a consistent advantage in the recursive approach over both sliding and expanding windows. LGBM Sliding Recursive and LGBM Expanding Recursive consistently achieved lower average RMSE values compared to their RF counterparts, by 4.62% and 5.89%, respectively. It is worth noting that, overall, the expanding window method performed slightly better in both recursive and direct approaches in both models, exhibiting around a 1% difference in performance as compared to the sliding window method.

5.2 *Weighted Root Mean Squared Scaled Error (WRMSSE) Analysis*

The WRMSSE metric, considering both sales volume and the impact of zero sales over the previous four weeks, provides a nuanced perspective. The observed week-to-week variations in WRMSSE align with the RMSE trends,

but with more moderate changes, highlighting the metric's sensitivity to sales volume dynamics and its ability to capture subtler variations in model performance.

In the thorough examination of WRMSSE, the continued excellence of RF Sliding Direct and RF Expanding Direct models was evident, surpassing various setups, including those involving LGBM models. The effectiveness of the direct approach within RF models was re-emphasized through the scaled error analysis, mirroring the trends observed in the RMSE results.

The percentage differences in average WRMSSE values between RF and LGBM models remained noticeable. RF Sliding Direct demonstrated a 5.76% lower average WRMSSE compared to LGBM Sliding Direct, and RF Expanding Direct exhibited a 5.85% lower average WRMSSE as compared to LGBM Expanding Direct. Figures 14 & 15 demonstrate how the models performed in each week.

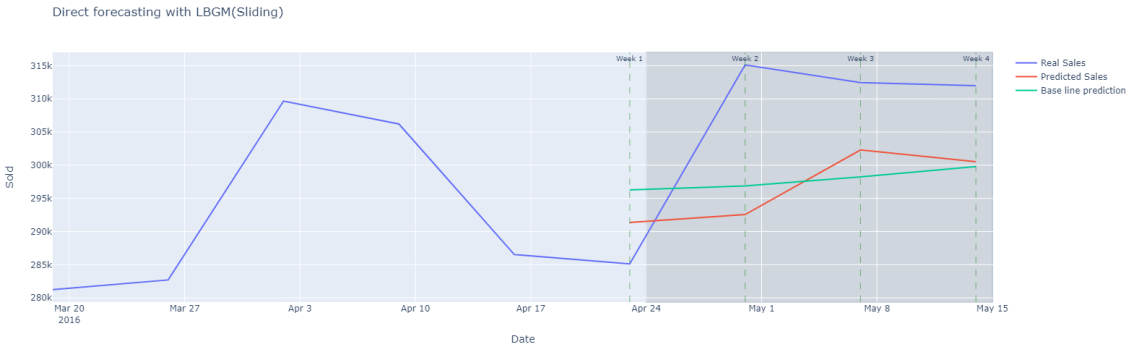

Figure 14: Direct forecasting with LGBM(Sliding)

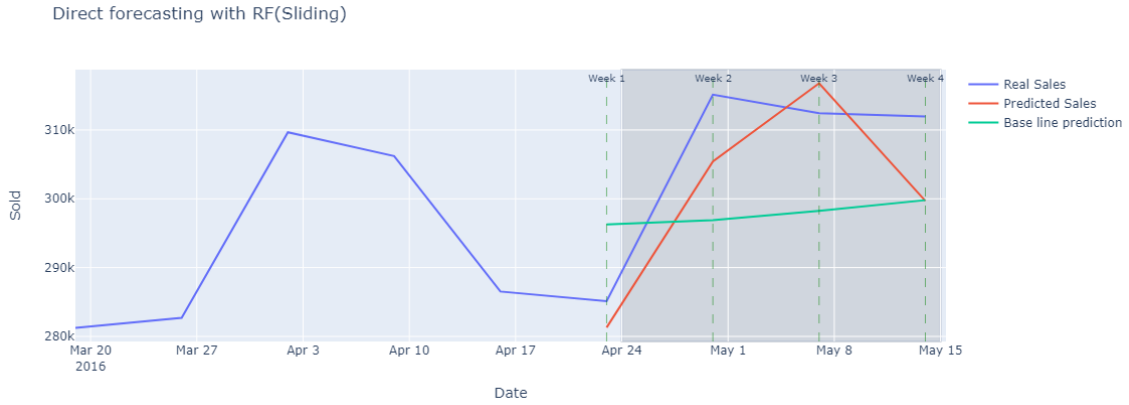

Figure 15: Direct forecasting with RF(Sliding)

Analyzing week-to-week variations, a similar trend to the RMSE analysis is observed in the WRMSSE results. Under the recursive approach, as shown in Table 12, both LGBM and RF models show increases, indicating potential challenges in adapting to evolving patterns. Following the first week, LGBM Sliding Recursive experienced a WRMSSE increase of 7.98%, with additional increases of 3.76% and a subsequent 5.61% decrease. In comparison, RF Sliding Recursive demonstrated a WRMSSE increase of 22.69% after the first week, indicating that the change in error is higher than the RMSE (10.81%), followed by a further increase of 4.97% and a subsequent 9.78% decrease. The big difference between the two errors is due to the sharp increase in sales in the second week, as WRMSSE is impacted more significantly by changes in sales scale than RMSE. This occurs because RMSSE, and consequently WRMSSE, scales RMSE relative to historical data trends, typically using the historical mean. Hence, substantial week-to-week changes in sales, such as the notable increase observed impacted disproportionately RMSSE. This scaling amplifies RMSSE's response to significant sales fluctuations, especially when these deviate from historical patterns.

Models	**Week 1**	**Week 1-2**	**Week 2-3**	**Week 3-4**
Base Line MA	0.25874	1.46%	-2.92%	-1.70%
LGBM Sliding Direct	0.23169	6.54%	-3.95%	5.22%
LGBM Sliding Recursive	0.39245	7.98%	3.76%	-5.61%
LGBM Expanding Direct	0.22942	9.02%	-6.59%	-7.57%
LGBM Expanding Recursive	0.38860	9.59%	0.02%	-3.63%
RF Sliding Direct	0.22309	2.40%	-4.45%	9.75%
RF Sliding Recursive	0.37145	22.69%	4.96%	-9.77%
RF Expanding Direct	0.21972	6.16%	-7.00%	9.82%
RF Expanding Recursive	0.37112	19.70%	0.01%	-7.75%

Table 12: Weekly Percentage Change in WRMSSE

For the expanding window approach, LGBM Expanding Recursive showed an increase of 9.59% after the first week, followed by more moderate increases of 0.02% and a decrease of 3.63%. RF Expanding Recursive exhibited a WRMSSE increase of 19.7% after the first week, remaining the same in the third week, followed by a 7.75% decrease. This granular examination underscores the challenges associated with the recursive approach, particularly in the context of scaled error metrics where capturing evolving patterns is crucial. Figures 16 & 17 demonstrate how the models performed in each week.

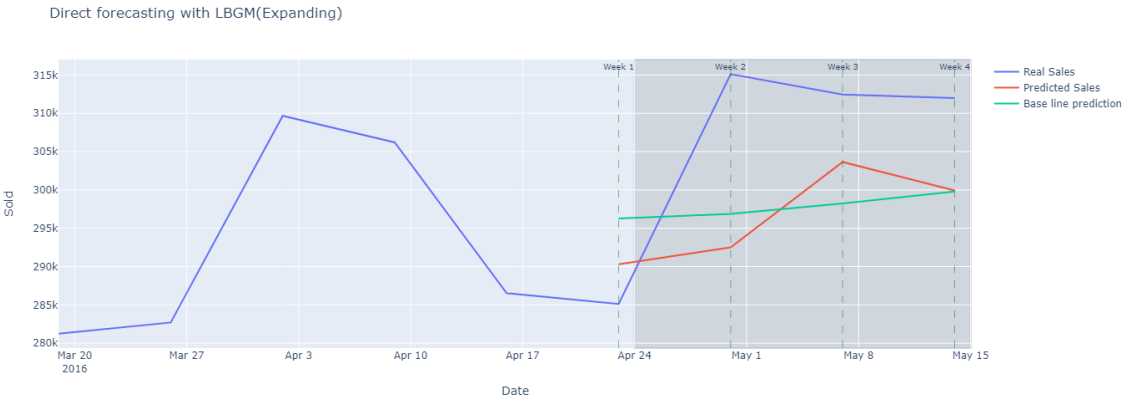

Figure 16: Direct forecasting with LBGM(Expanding)

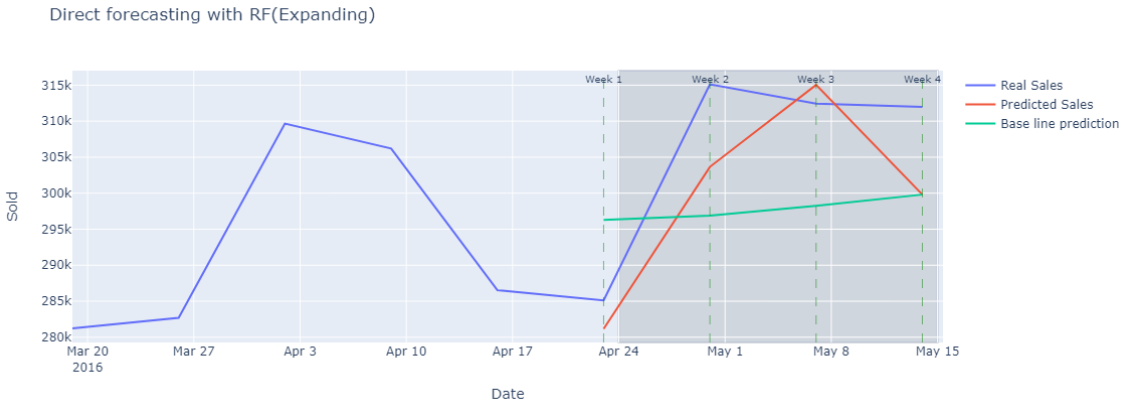

Figure 17: Direct forecasting with RF(Expanding)

Reiterating the trend observed in the RMSE analysis, LGBM models consistently demonstrated a marginal, yet discernible, superiority in performance in the recursive approach over both sliding and expanding windows. LGBM Sliding Recursive and LGBM Expanding Recursive consistently achieved lower average WRMSSE values when compared to their RF counterparts by 8.33% and 8.45%, respectively.

These detailed findings provide a nuanced understanding of the comparative performance of LGBM and RF models under various configurations, emphasizing the complexities associated with the recursive and sliding approaches and their impact on capturing evolving patterns in the context of Walmart sales forecasting. The ensuing discussion will delve into the implications of these results, offering insights into potential contributing factors and paths for further exploration.

6 DISCUSSION

6.1 *Results Discussion*

Our exploration into the performance of tree-based algorithms, specifically LGBM and RF, against the Moving Average baseline uncovered a substantial enhancement in predictive capabilities. Both LGBM and RF consistently surpassed the baseline, demonstrating the proficiency of machine learning models in capturing intricate patterns within Walmart's sales data.

In the comparative analysis of recursive and direct approaches in forecasting models, a notable performance discrepancy was observed between the LGBM and RF algorithms. Specifically, the LGBM consistently demonstrated superior performance over RF when utilizing the recursive strategy. In contrast, the direct approach revealed an opposite trend, with RF models outperforming LGBM. This variation in performance invites further investigation into each algorithm's adaptability to changing patterns and the impact of input sequences on forecasting accuracy. Additionally, in both models, the direct approach exhibited overall better performance than the recursive method. However, this finding contrasts with the study by Xue et al. (2019), which reported a superior performance of recursive prediction over direct prediction. This contradiction underscores the importance of data characteristics and domain-specific dynamics in determining the most effective prediction technique.

Notably, LGBM's boosting framework plays a pivotal role in its observed superiority in the recursive approach. Boosting algorithms, such as LGBM, are designed to iteratively correct errors in previous predictions, allowing the model to adapt and learn from the evolving nature of the data. This characteristic proves particularly advantageous in recursive predictions, where the model needs to continuously refine its forecasts as new information becomes accessible.

Our investigation into the sliding and expanding window methodologies, despite their relatively short four-week window, provided intriguing insights into their impact on predictive performance. The expanding window consistently beat the sliding window, underscoring the significance of progressively incorporating more historical data in the domain of retail forecast. This contradict the findings of Lehmann (2021), were the rolling window was superior in performance. However, the overall impact was somewhat constrained, possibly due to the short window duration.

In contrasting our findings with the broader landscape of time series analysis, we compare them to the M5 forecasting competition by Makridakis et al. (2022b). While our study focused on weekly data with a four-week forecast horizon, differing from the M5 competition's daily data

and 28-day forecast horizon, these disparities highlight the potential influence on the optimal algorithm choice. Unlike the M5 competition, where LGBM was identified as the best-performing model, our study highlights RF as the superior algorithm. This deviation underscores the sensitivity of model performance to factors like temporal granularity and forecast horizon, suggesting that the effectiveness of machine learning algorithms in time series forecasting is highly contingent on dataset specifics and the methodology employed.

Detailed analysis exposes significant differences between recursive and direct approaches, notably in RF. Recursive models face challenges capturing substantial sales fluctuations, underlining immediate pattern recognition issues. A marked decrease in errors in the last week of the recursive approach suggests potential parity with the direct approach's performance over a larger predictive horizon. Investigation into handling extreme values highlights RF's effectiveness in capturing sudden sales increases, attributed to its ensemble nature. Aligning algorithmic strengths with the forecasting task is crucial, emphasizing the significance of handling extreme values for accurate sales predictions.

This discussion provides a comprehensive exploration of our results, offering insights into the intricate dynamics of algorithmic performance, temporal considerations, and the broader implications for time series forecasting in the retail domain.

6.2 *Limitations*

Despite the comprehensive analysis and valuable insights gained from our study, several limitations merit consideration. Firstly, due to computational constraints, we transformed the original daily sales data into a weekly format to enable training on the entire data set. This transformation might have resulted in the loss of subtle daily patterns, potentially impacting the models' ability to capture short-term dependencies.

The relatively short four-week window used in sliding and expanding methodologies may have constrained the models' ability to capture long-term dependencies. A more extended window could offer a more robust evaluation of their impact.

Additionally, the data set's temporal granularity and forecast horizon, differing from the M5 competition, highlight the need for caution in generalizing our findings. The specific characteristics of weekly data and a four-week forecast horizon may influence algorithm performance and our results might not directly extrapolate to daily data and longer forecast periods.interpretation and generalisation.

6.3 *Recommendations*

Our study offers key recommendations for advancing retail sales forecasting. Researchers should explore the impact of longer observation windows, specifically delving into the effects of sliding and expanding window methodologies to understand the models' adaptability to longer-term sales dependencies. Sensitivity analyses on temporal granularity are essential to comprehend the models' performance under varying time intervals. Investigating different temporal resolutions provides insights into generalizability and adaptability to diverse datasets.

Expanding the scope to include diverse retail datasets beyond Walmart's enriches our understanding of algorithmic performance in various retail contexts, identifying models with more universal effectiveness. Integration of external factors, like weather conditions and inflation rates, into forecasting models is crucial for enhancing contextual awareness and improving predictions in dynamic retail environments.

Despite potential computational expenses, researchers are encouraged to explore training and tuning models separately for different stores or product categories. This tailored approach acknowledges potential heterogeneity in sales patterns across segments, potentially leading to more accurate and context-specific predictions. In conclusion, addressing these recommendations will refine existing methodologies and contribute to the development of more effective predictive models in the complex and dynamic landscape of retail sales forecasting.

7 CONCLUSION

In conclusion, this study systematically evaluated the predictive performance of LGBM and RF models using weekly sales data from Walmart. Through an exhaustive exploration of direct and recursive approaches, incorporating sliding and expanding window techniques, the research uncovered valuable insights into the intricacies of retail sales forecasting.

Distinct patterns emerged from the comparative analysis, stressing out the consistent superiority of RF models in the direct approach, whether employing sliding or expanding windows. This superiority was evident in lower RMSE and WRMSSE metrics. Conversely, LGBM models demonstrated a slightly better performance in the recursive approach, highlighting their adaptability to evolving patterns over successive forecasting periods, as compared to RF models.

In addition, it is noteworthy that the expanding window technique consistently outperformed the sliding window in both LGBM and RF models, across direct and recursive predictions. This observation emphasizes

the significance of progressively incorporating more historical data for improved predictions, suggesting that the expanding window approach provides a valuable advantage in the context of retail sales forecasting.

The study significantly contributes to existing literature by not only assessing model performance, but also delving into the nuances of various forecasting approaches. Notably, the observed trends underscore that the expanding window consistently outperformed the sliding window one, thus emphasizing the significance of progressively incorporating more historical data for improved predictions.

The findings suggest exploring longer observation windows, conducting sensitivity analyses on temporal granularity and incorporating external factors like weather and inflation for more accurate predictions in dynamic retail settings. However, limitations such as transforming daily sales into weekly data and a short observation window do exist. Future research can focus on diversifying algorithms, extending observation windows and tailoring models for specific stores or product categories. Overall, this research enhances understanding of machine learning in retail sales forecasting, providing insights into model performance and guiding future research for more effective predictive modeling in the evolving retail landscape.

REFERENCES

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019a). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019b). Optuna: A next-generation hyperparameter optimization framework.

Arik, S. O., & Pfister, T. (2021). Tabnet: Attentive interpretable tabular learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(8), 6679–6687.

Ben Taieb, S., & Hyndman, R. (2012). *Recursive and direct multi-step forecasting: The best of both worlds* (Monash Econometrics and Business Statistics Working Papers No. 19/12). Monash University, Department of Econometrics and Business Statistics. https://EconPapers.repec.org/RePEc:msh:ebswps:2012-19

Bilolikar, D. K., More, A., Gong, A., & Janssen, J. (2023). How to outperform default random forest regression: Choosing hyperparameters for applications in large-sample hydrology.

Bollerslev, T., Hood, B., Huss, J., & Pedersen, L. H. (2018). Risk Everywhere: Modeling and Managing Volatility. *The Review of Financial Studies*, *31*(7), 2729–2773. https://doi.org/10.1093/rfs/hhy041

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating time series forecasting models: An empirical study on performance estimation methods. *Machine Learning*, *109*(11), 1997–2028. https://doi.org/10.1007/s10994-020-05910-7

Chakraborty, D., Elhegazy, H., Elzarka, H., & Gutierrez, L. (2020). A novel construction cost prediction model using hybrid natural and light gradient boosting. *Advanced Engineering Informatics*, *46*, 101201. https://doi.org/https://doi.org/10.1016/j.aei.2020.101201

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785

Davydenko, A., & Fildes, R. (2013). Measuring forecasting accuracy: The case of judgmental adjustments to sku-level demand forecasts. *International Journal of Forecasting*, *29*, 510–522. https://doi.org/10.1016/j.ijforecast.2012.09.002

Degiannakis, S., & Filis, G. (2017). Forecasting oil price realized volatility using information channels from other asset classes. *Journal of International Money and Finance*, *76*, 28–49. https://doi.org/https://doi.org/10.1016/j.jimonfin.2017.05.006

DeHoratius, N., Musalem, A., & Rooderkerk, R. (2023). Why retailers fail to adopt advanced data analytics. *Harvard Business Review*, *101*(2), 84–91.

Ding, Y., Fan, L., & Liu, X. (2021). Analysis of feature matrix in machine learning algorithms to predict energy consumption of public buildings. *Energy and Buildings*, *249*, 111208. https://doi.org/https://doi.org/10.1016/j.enbuild.2021.111208

Effrosynidis, D., Spiliotis, E., Sylaios, G., & Arampatzis, A. (2023). Time series and regression methods for univariate environmental forecasting: An empirical evaluation. *Science of The Total Environment*, *875*, 162580. https://doi.org/https://doi.org/10.1016/j.scitotenv.2023.162580

Ensafi, Y., Amin, S. H., Zhang, G., & Shah, B. (2022). Time-series forecasting of seasonal items sales using machine learning – a comparative analysis. *International Journal of Information Management Data Insights*, *2*(1), 100058. https://doi.org/https://doi.org/10.1016/j.jjimei.2022.100058

Federation, N. R. (2020). *Retail's impact on the economy*. https://nrf.com/research-insights/retails-impact

Gillitzer, C., & McCarthy, M. (2019). Does global inflation help forecast inflation in industrialized countries? *Journal of Applied Econometrics*, *34*(5), 850–857. https://doi.org/https://doi.org/10.1002/jae.2704

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

He Zhou, W. Q., & Yang, Y. (2022). Tweedie gradient boosting for extremely unbalanced zero-inflated data. *Communications in Statistics - Simulation and Computation*, *51*(9), 5507–5529. https://doi.org/10.1080/03610918.2020.1772302

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Karasu, S., & Altan, A. (2019). Recognition model for solar radiation time series based on random forest with feature selection approach. *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, 8–11. https://doi.org/10.23919/ELECO47770.2019.8990664

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017a). Lightgbm: A highly efficient gradient boosting decision tree. *Microsoft Research*.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017b). *Lightgbm: A highly efficient gradient boosting decision tree* [URL https://github.com/microsoft/LightGBM].

Lehmann, R. (2021). Forecasting exports across europe: What are the superior survey indicators? *Empirical Economics*, *60*. https://doi.org/10.1007/s00181-020-01838-y

Lekhwar, S., Yadav, S., & Singh, A. (2019). Big data analytics in retail. In S. C. Satapathy & A. Joshi (Eds.), *Information and communication technology for intelligent systems* (pp. 469–477). Springer Singapore.

Liaw, A., & Wiener, M. (2001). Classification and regression by randomforest. *Forest*, *23*.

Ma, F., Wahab, M., & Zhang, Y. (2019). Forecasting the u.s. stock volatility: An aligned jump index from g7 stock markets. *Pacific-Basin Finance Journal*, *54*, 132–146. https://doi.org/https://doi.org/10.1016/j.pacfin.2019.02.006

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022a). M5 accuracy competition: Results, findings, and conclusions [Special Issue: M5 competition]. *International Journal of Forecasting*, *38*(4), 1346–1364. https://doi.org/https://doi.org/10.1016/j.ijforecast.2021.11.013

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022b). The m5 competition: Background, organization, and implementation. *International Journal of Forecasting*, *38*, 1325–1336. https://doi.org/10.1016/J.IJFORECAST.2021.07.007

Pandas Development Team. (2020). *Pandas* (Version latest). Zenodo. https://doi.org/10.5281/zenodo.3509134

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn (version 1.2.1). https://scikit-learn.org

Prestwich, S., Rossi, R., Armagan Tarim, S., & Hnich, B. (2014). Mean-based error measures for intermittent demand forecasting. *International Journal of Production Research*, *52*(22), 6782–6791. https://doi.org/10.1080/00207543.2014.917771

Python Software Foundation. (2021). *Python language reference, version 3.9* [Available online at https://docs.python.org/3/library/datetime.html].

Radwan, A. M. (2021). Forecasting of covid-19 using time series regression models. *2021 Palestinian International Conference on Information and Communication Technology (PICICT)*, 7–12. https://doi.org/10.1109/PICICT53635.2021.00014

Saha, P., Gudheniya, N., Mitra, R., Das, D., Narayana, S., & Tiwari, M. K. (2022). Demand forecasting of a multinational retail company using deep learning frameworks [10th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2022]. *IFAC-PapersOnLine*, *55*(10), 395–399. https://doi.org/https://doi.org/10.1016/j.ifacol.2022.09.425

Shwartz-Ziv, R., & Armon, A. (2021). Tabular data: Deep learning is not all you need. *arXiv preprint arXiv:2106.03253*. https://doi.org/10.48550/arXiv.2106.03253

Statsmodels Developers. (2021). *Statsmodels: Statistical modeling and econometrics in python* [URL https://www.statsmodels.org].

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Xue, P., Jiang, Y., Zhou, Z., Chen, X., Fang, X., & Liu, J. (2019). Multi-step ahead forecasting of heat load in district heating systems using machine learning algorithms. *Energy*, *188*, 116085. https://doi.org/https://doi.org/10.1016/j.energy.2019.116085

APPENDIX A

	id	item_id	dept_id	cat_id	store_id	state_id	d_1	d_2	d_1940	d_1941
1	HOBBIES_1_001_CA_1	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	1
2	HOBBIES_1_002_CA_1	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0
3	HOBBIES_1_003_CA_1	HOBBIES__003	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	1
30488	FOODS_3_826_WI_3	FOODS_3_826	FOODS_3	FOODS	WI_3	WI	0	0	1	0
30489	FOODS_3_827_WI_3	FOODS_3_827	FOODS_3	FOODS	WI_3	WI	0	0	5	1

Table 13: Sales.csv

Variable	**Unique Counts**
item_id	3049
dept_id	7
cat_id	3
store_id	10
state_id	3
week	52
month	12
quarter	4
year	6
season	4

Table 14: Unique Counts for Different Variables

date	wm_yr_wk	weekday	wday	month	year	d	event_name_1	event_type_1	snap_CA
2011-01-29	11101	Saturday	1	1	2011	d_1	No event	No event	0
2011-01-30	11101	Sunday	2	1	2011	d_2	No event	No event	0
2011-01-31	11101	Monday	3	1	2011	d_3	No event	No event	0
2011-02-01	11101	Tuesday	4	2	2011	d_4	No event	No event	1
2011-02-02	11101	Wednesday	5	2	2011	d_5	No event	No event	1

Table 15: Calendar.csv (excluding some columns due to size)

store_id	item_id	wm_yr_wk	sell_price
CA_1	HOBBIES_1_001	11325	9.58
CA_1	HOBBIES_1_001	11326	9.58
CA_1	HOBBIES_1_001	11327	8.26
CA_1	HOBBIES_1_001	11328	8.26
CA_1	HOBBIES_1_001	11329	8.26

Table 16: Price.csv

Package	Source
python 3.12.1	(Python Software Foundation, 2021)
pandas	(Pandas Development Team, 2020)
numpy	(Harris et al., 2020)
Matplotlib	(Hunter, 2007)
seaborn	(Waskom, 2021)
Scikit-learn, RandomForestRegressor, MeanSquareError	(Pedregosa et al., 2011)
optuna	(Akiba et al., 2019a)
datetime	(Python Software Foundation, 2021)
LGBMRegressor	(Ke et al., 2017b)
statsmodels	(Statsmodels Developers, 2021)

Table 17: Packages and Software

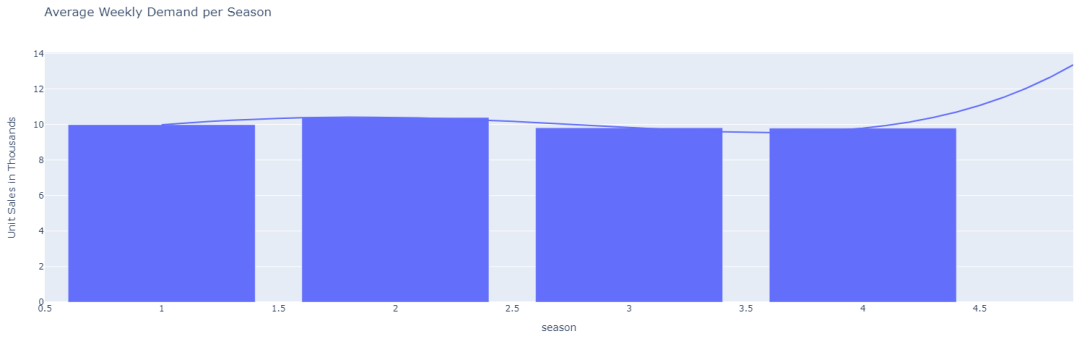

Figure 18: Unit sales per Season

APPENDIX B

Lag Features	Rolling Features
'shift_1_sold'	'shift_1_Rolling_mean_2'
'shift_2_sold'	'shift_1_Rolling_std_2'
'shift_3_sold'	'shift_1_Rolling_max_2'
'shift_4_sold'	'shift_1_Rolling_min_2'
	'shift_1_Rolling_mean_3'
	'shift_1_Rolling_std_3'
	'shift_1_Rolling_max_3'
	'shift_1_Rolling_min_3'
	'shift_1_Rolling_mean_4'
	'shift_1_Rolling_std_4'
	'shift_1_Rolling_max_4'
	'shift_1_Rolling_min_4'
	'shift_1_Rolling_mean_5'
	'shift_1_Rolling_std_5'
	'shift_1_Rolling_max_5'
	'shift_1_Rolling_min_5'

Table 18: Lag and Rolling Features

Price-Related Features	Description
'price_store_max'	Maximum price observed at the store level
'price_day_max'	Maximum price observed on a particular day
'price_max'	Overall maximum price observed
'price_min'	Overall minimum price observed
'price_std'	Standard deviation of prices
'price_mean'	Mean of prices
'price_norm'	Normalized prices relative to a reference point
'price_momentum'	Price momentum in the most recent period
'price_momentum_m'	Price momentum over an extended period

Table 19: Price-Related Features