*applied sciences*

MDPI

*Article*

# Retail Sales Forecasting Using Tree-Based Machine Learning Models: An Empirical Study of Preprocessing Configurations and Feature Ablation

**Sarah Albassam *, Atheer Alqahtani and Amal Alazba**

Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia; atheer.a.s.q@gmail.com (A.A.); aalazba@ksu.edu.sa (A.A.)
* Correspondence: salbassam@ksu.edu.sa

**Abstract**

Sales forecasting plays a critical role in business decision-making, including financial planning, resource allocation, and inventory management. Despite the growing adoption of machine learning in forecasting applications, controlled evaluations of how preprocessing decisions and contextual feature availability affect model performance remain limited. This study presents a controlled benchmarking evaluation that systematically examines preprocessing strategies and contextual feature availability across seven representative tree-based machine learning models, using two publicly available retail datasets: BigMart and Rossmann. Multiple preprocessing configurations are evaluated on both datasets, and ablation studies are conducted on the Rossmann dataset to quantify the independent contribution of seasonal and promotional features and the *Customers* variable to forecasting accuracy. Results show that GradientBoosting achieved the best performance on BigMart ($R^2$ = 0.611), while RandomForest and ExtraTrees led on Rossmann $R^2$ = 0.884 under realistic conditions excluding the *Customers* variable, and $R^2$ = 0.965 in an ablation scenario where *Customers*, typically unavailable at forecast time, is included. Ablation analysis confirmed that seasonal and promotional features are critical drivers of forecasting accuracy, and that the *Customers* variable is the single most influential predictor in the Rossmann dataset. These findings highlight the importance of preprocessing design and contextual feature availability in retail sales forecasting and provide practical insights for building effective forecasting pipelines.

**Keywords:** retail sales forecasting; tree-based machine learning; feature ablation; ensemble learning



Academic Editor: Christos Bouras

Received: 28 June 2026
Revised: 23 July 2026
Accepted: 27 July 2026
Published: 29 July 2026



## 1. Introduction

Accurate sales forecasting has become a critical requirement for modern organizations operating in increasingly dynamic and competitive markets. Forecasting enables businesses to anticipate future demand, optimize inventory levels, allocate resources efficiently, and support strategic decision-making processes. Inaccurate forecasts, on the other hand, may lead to inventory shortages, overstocking, inefficient resource utilization, and substantial financial losses. Consequently, sales forecasting is recognized as a fundamental component of organizational planning and business intelligence systems, particularly in data-driven enterprises that rely on timely and reliable predictive insights to maintain competitiveness [1].

The consequences of inaccurate forecasting can be substantial. A frequently cited example is the collapse of Target Canada, which has been partially attributed to poor demand forecasting and inventory planning, resulting in losses exceeding $2.1 billion and the closure of 133 retail stores [2]. Such cases emphasize the necessity of developing forecasting models capable of adapting to rapidly changing market conditions and incorporating external contextual factors.

The rapid advancement of artificial intelligence (AI) and machine learning (ML) technologies has significantly transformed forecasting practices across industries. Traditional forecasting approaches primarily relied on statistical techniques and historical sales patterns, which often struggled to capture nonlinear relationships and rapidly changing market conditions. In contrast, AI-driven forecasting models can analyze large-scale datasets, identify hidden patterns, and adapt to evolving trends with greater precision and flexibility [3]. Recent studies have reported that organizations adopting AI-based forecasting systems achieved reductions of approximately 15–20% in forecasting errors compared with conventional methods, leading to improved inventory management, reduced operational costs, and enhanced customer satisfaction [4].

Beyond improving predictive accuracy, modern forecasting systems provide organizations with deeper insights into customer behavior, purchasing patterns, and market dynamics. The integration of contextual information, including promotional campaigns, holidays, seasonal patterns, and customer traffic indicators, has enabled forecasting models to become more responsive and context-aware [4]. Additionally, automation capabilities offered by AI-based systems reduce repetitive manual tasks, minimize human error, and allow decision-makers to focus on strategic planning and business development activities. These capabilities highlight the growing importance of intelligent forecasting systems in supporting operational efficiency and organizational resilience.

Despite these advances, achieving highly accurate and adaptable sales forecasting remains a significant challenge. Many forecasting models rely primarily on historical sales information while giving limited attention to contextual variables such as seasonal effects, promotional activities, store characteristics, and customer traffic patterns [5,6]. Understanding the relative contribution of these factors remains an important challenge for developing robust forecasting systems. Furthermore, data preprocessing, one of the most critical stages in the machine learning pipeline, is frequently underemphasized in prior studies. Poor preprocessing practices can introduce inconsistencies, missing values, and noisy data, thereby negatively affecting the predictive performance of ML and DL models [7].

The significance of effective preprocessing is particularly important in real-world forecasting applications where datasets are often heterogeneous, incomplete, and highly complex. It has been estimated that data cleaning and preprocessing account for nearly 80% of a data scientist's effort, emphasizing their central role in the predictive analytics lifecycle [7]. Nevertheless, many forecasting studies place greater emphasis on model development than on evaluating the impact of data preprocessing and feature availability on forecasting performance. While preprocessing is a critical stage of the machine learning pipeline, the effects of common practices such as missing-value handling, categorical encoding, and feature selection are often reported only briefly or treated as secondary considerations. These limitations reduce understanding of the factors that most strongly influence forecasting performance.

Another major challenge lies in developing forecasting systems that are not only accurate but also scalable and accessible to organizations of different sizes. Small and medium-sized enterprises (SMEs), which contribute substantially to global economic growth and employment, often lack access to sophisticated forecasting technologies and advanced

analytical infrastructures [8]. Consequently, there is growing interest in forecasting approaches that balance predictive performance with implementation simplicity, interpretability, and computational efficiency. Although deep learning models have demonstrated strong forecasting capabilities in large-scale and highly complex prediction tasks, they typically require substantial computational resources, extensive hyperparameter tuning, and large volumes of training data to achieve optimal performance. In contrast, ensemble tree-based methods such as Random Forest, Extra Trees, Gradient Boosting, and XGBoost have consistently demonstrated competitive performance on structured tabular data while offering lower computational overhead and greater ease of deployment [9]. These characteristics make ensemble learning approaches particularly attractive for retail forecasting applications and for organizations operating in resource-constrained environments, where practical implementation considerations are often as important as predictive accuracy.

To address these challenges, this study presents a controlled benchmarking evaluation of seven tree-based machine learning models across two publicly available retail datasets: BigMart and Rossmann. Feature-ablation experiments are conducted on the Rossmann dataset to quantify the contribution of seasonal and promotional variables and the Customers feature to forecasting performance. The main contributions of this study are as follows:

- Providing a controlled evaluation of multiple preprocessing configurations, including missing-value imputation, categorical encoding, and zero-variance feature removal, to quantify their impact on retail sales forecasting performance under identical experimental conditions. This aspect often receives limited evaluation in prior retail forecasting studies.
- Quantifying, through controlled feature-ablation experiments on the Rossmann dataset, the impact of seasonal, promotional, and customer-related variables on forecasting accuracy. This provides an assessment of the contribution of contextual features that is seldom reported in a controlled manner in the retail forecasting literature.
- Establishing a controlled benchmarking framework based on seven representative tree-based machine learning algorithms evaluated under identical experimental conditions across two complementary retail forecasting datasets. This framework provides the experimental foundation for assessing the effects of preprocessing strategies and contextual feature availability.
- Providing empirical insights into how dataset characteristics and contextual feature availability influence the relative effectiveness of tree-based ensemble learning methods across different retail forecasting scenarios.

Collectively, these contributions address the research gaps identified in Section 2.6, particularly the limited systematic evaluation of preprocessing strategies and the lack of controlled feature-ablation analyses quantifying the contribution of contextual variables in retail sales forecasting.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on sales forecasting, machine learning, and deep learning approaches, while highlighting existing research gaps. Section 3 presents the proposed methodology, including dataset description, preprocessing procedures, model development, and evaluation metrics. Section 4 discusses the experimental results and practical implications. Finally, Section 5 concludes the study, outlines its limitations, and provides recommendations for future research directions.

## 2. Literature Review

Recent advances in machine learning (ML) and deep learning (DL) have significantly transformed sales forecasting research. Traditional statistical forecasting approaches often

struggle to capture nonlinear relationships, seasonality, and rapidly changing market conditions. Consequently, researchers have increasingly adopted ML and DL techniques because of their ability to process large-scale datasets, identify hidden patterns, and improve forecasting accuracy in dynamic business environments [10]. Existing studies have explored a wide range of forecasting models, including regression-based methods, ensemble learning approaches, tree-based algorithms, recurrent neural networks, and hybrid architectures. Despite these advancements, several limitations remain unresolved, particularly regarding the integration of external contextual variables and the effective use of advanced data preprocessing techniques.

### 2.1. Machine Learning-Based Sales Forecasting

Machine learning approaches have been extensively applied to sales forecasting problems due to their flexibility, scalability, and predictive capabilities. Early studies primarily focused on regression and ensemble learning techniques for forecasting retail sales. For example, Cheriyan et al. [11] employed Generalized Linear Models (GLM), Decision Trees (DT), and Gradient Boosting Techniques (GBT) for retail sales prediction and reported superior performance for GBT compared with other methods. Similarly, Krishna et al. [12] evaluated multiple regression-based approaches, including Lasso, Ridge, AdaBoost, and GradientBoost algorithms, emphasizing the importance of preprocessing techniques such as handling missing values during model development. However, both studies relied mainly on internal historical sales data while neglecting external influences such as market dynamics and consumer behavior.

Subsequent studies expanded toward time-series forecasting and hybrid machine learning architectures. Swami et al. [13] integrated XGBoost, Long Short-Term Memory (LSTM), and ARIMA models for forecasting retail product sales and found that XGBoost achieved superior performance because of efficient feature engineering and regularization mechanisms. Bajaj et al. [14] further compared Random Forest Regressor, XGBoost Regressor, K-Nearest Neighbor (KNN), and Linear Regression models, concluding that Random Forest achieved the highest prediction accuracy. Although these studies incorporated seasonality and temporal dependencies, they still primarily focused on internal sales patterns without adequately considering external contextual factors.

Recent research has increasingly emphasized the role of advanced preprocessing techniques and contextual information in improving forecasting accuracy. Comparative studies using Walmart and Big Mart datasets demonstrated that algorithms such as Extra Tree Regressor, Ridge Regression, and XGBoost can achieve high predictive performance when combined with systematic preprocessing techniques for handling outliers, inconsistencies, and feature transformations [15–17]. Martins and Galegale [18] conducted a meta-analysis of machine learning forecasting studies and highlighted that many forecasting models continue to overlook external variables, limiting practical applicability and prediction robustness. Similarly, Sim and Wei [19] showed that XGBoost-based forecasting systems achieved efficient prediction performance; however, external factors remained insufficiently explored. More recently, Hobor et al. [20] compared tree-based ensembles with deep learning architectures, including N-BEATS, N-HiTS, and the Temporal Fusion Transformer, on a large brick-and-mortar retail dataset with intermittent demand. Localized XGBoost and LightGBM models consistently outperformed all neural architectures while requiring substantially less training time.

Recent studies have begun integrating selected external variables into forecasting frameworks. Studies incorporating seasonality, market conditions, and temporal trends demonstrated noticeable improvements in prediction accuracy. Bijalwan et al. [21] showed that integrating seasonal effects improved forecasting performance by enabling models

to capture nonlinear time-series relationships more effectively. Likewise, Jewel et al. [22] demonstrated that incorporating external variables and dynamic preprocessing strategies significantly enhanced forecasting accuracy under changing market conditions. Nevertheless, the incorporation of external contextual information remains limited and inconsistent across the literature.

### 2.2. Tree-Based Ensemble Learning for Retail Forecasting

Tree-based ensemble learning methods have become increasingly popular in retail demand forecasting due to their ability to capture nonlinear relationships, handle mixed data types, and maintain strong predictive performance on structured tabular datasets. Unlike traditional statistical approaches that often rely on assumptions of linearity and stationarity, ensemble methods can effectively model complex interactions among product, store, promotional, seasonal, and customer-related variables. Furthermore, these methods generally require less extensive data preparation than many deep learning approaches, making them attractive for practical forecasting applications [9].

Ensemble learning techniques can generally be categorized into bagging and boosting approaches. Bagging-based methods, such as Random Forest (RF) and Extra Trees (ET), construct multiple decision trees using random subsets of observations and features, thereby reducing variance and improving model robustness. In contrast, boosting-based methods, including AdaBoost, Gradient Boosting (GB), Histogram Gradient Boosting (HGB), and Extreme Gradient Boosting (XGBoost), sequentially build learners that focus on correcting the errors of previous models, often resulting in improved predictive accuracy [23,24].

Recent studies have demonstrated the effectiveness of tree-based ensembles in retail forecasting applications. Seyedan et al. [25] proposed a cluster-based demand forecasting framework that combined multiple forecasting models through Bayesian Model Averaging (BMA). Using retail demand data from the sports retail industry, the authors reported that ensemble approaches consistently outperformed individual forecasting models and improved forecasting accuracy across different seasonal and demand scenarios.

Nasseri et al. [26] conducted a comprehensive comparison of tree-based ensemble methods and Long Short-Term Memory (LSTM) networks using more than 5.2 million demand observations collected from an Austrian supermarket. Their evaluation included Extra Trees, Random Forest, Gradient Boosting, and XGBoost models alongside LSTM networks. Using historical demand, pricing, promotional, weather, and calendar-related features, the study found that tree-based ensemble methods consistently achieved superior forecasting performance across multiple product categories while requiring lower computational complexity and less extensive model tuning than deep learning approaches. These findings highlight the effectiveness of ensemble methods for large-scale retail forecasting problems.

More recently, Mishra et al. [27] developed an explainable retail demand forecasting framework based on LightGBM and SHAP analysis. Using more than one million retail transactions and forty-one engineered features, the study demonstrated that gradient boosting models achieved superior forecasting performance compared with several baseline machine learning approaches. The results further showed that temporal, lag-based, and rolling statistical features were among the most influential predictors of retail demand, emphasizing the importance of feature availability alongside model selection. In a related large-scale evaluation, Gerg et al. [28] compared eight machine learning models, ranging from linear regression to tree-based ensembles, across three retail sectors and all 50 U.S. states, encompassing over 2400 model configurations. XGBoost and LightGBM consistently ranked among the top-performing models across this diverse experimental scope, reinforcing the robustness of tree-based ensembles across heterogeneous retail contexts and

geographies. Soleimani et al. [29] combined lag-based feature engineering with a bagged Random Forest–LightGBM ensemble for weekly Walmart sales forecasting. While an LSTM outperformed individual tree-based models, their final bagged and stacked ensemble ultimately surpassed the LSTM, achieving an $R^2$ of 0.981.

Although ensemble learning methods have consistently demonstrated strong performance on retail forecasting tasks, several limitations remain in the existing literature. First, most studies focus primarily on identifying the most accurate forecasting algorithm while providing limited analysis of how preprocessing decisions influence model performance. Second, although contextual variables such as promotions, seasonality, holidays, and customer-related information are frequently incorporated into forecasting models, their individual contribution is rarely quantified through controlled feature-ablation experiments. Finally, comparative evaluations of multiple ensemble methods across datasets with substantially different feature characteristics remain limited. These gaps motivate the need for controlled benchmarking studies that systematically examine the effects of preprocessing strategies and feature availability on retail forecasting performance.

### *2.3. Deep Learning-Based Sales Forecasting*

Deep learning has emerged as a promising direction for sales forecasting because of its ability to automatically learn nonlinear and hierarchical representations from large-scale data. Early DL-based forecasting studies focused primarily on Deep Neural Networks (DNNs). Chang et al. [30] utilized DNN architectures for forecasting pharmaceutical product sales and reported improvements in prediction performance for specific product categories. However, the study did not investigate external variables or advanced preprocessing strategies, limiting the model's adaptability and generalizability. Subsequent studies investigated more sophisticated DL architectures for forecasting applications. Dharshini and Vijila [10] compared ML and DL approaches for sales forecasting and highlighted the advantages of deep architectures in reducing forecasting error through multilayer feature extraction. However, the study was primarily theoretical and lacked empirical validation using real-world datasets. Similarly, Neelakandan et al. [31] proposed a Deep Learning Multilayer Neural Network (DLMNN) combined with Single Feature Selection (SFS) for e-commerce sales forecasting and demonstrated improvements in RMSE and prediction stability. Although extensive preprocessing was applied, the study still neglected external contextual variables such as economic trends and customer behavior.

Recent developments have increasingly focused on recurrent architectures such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRUs), which are particularly effective for modeling sequential and temporal data. Luyo Ballena et al. [32] compared RNN, GRU, and LSTM architectures and reported that LSTM achieved the best performance in terms of Mean Absolute Percentage Error (MAPE). Despite the strong predictive capability of recurrent architectures, most studies continued to rely primarily on internal sales data while neglecting external market conditions and comprehensive preprocessing procedures.

Table 1 summarizes the major studies discussed in the literature, including the methodologies used, consideration of external factors, preprocessing techniques, and identified limitations.

**Table 1.** Summary of Existing Sales Forecasting Studies.

| Study | Year | Methodology | External Factors | Pre-Processing | Main Findings/Limitations |
|---|---|---|---|---|---|
| Cheriyan et al. [11] | 2018 | GLM, DT, GBT | No | Limited | GBT achieved superior performance; however, the study ignored external variables and suffered from preprocessing limitations. |
| Krishna et al. [12] | 2018 | Regression Models, AdaBoost, GradientBoost | No | Yes | GradientBoost outperformed other models, highlighting the importance of preprocessing and handling missing values. |
| Swami et al. [13] | 2020 | XGBoost, LSTM, ARIMA | No | Yes | XGBoost demonstrated superior forecasting performance, although the study lacked external contextual integration. |
| Bajaj et al. [14] | 2020 | Random Forest, XGBoost, KNN | Limited | Yes | Random Forest achieved the highest prediction accuracy; however, external variables were not considered. |
| Raizada and Saini [17] | 2021 | Extra Tree, Random Forest, SVR | Yes | Yes | Extra Tree Regressor showed strong predictive performance, though model generalizability remained limited. |
| Huo [16] | 2021 | XGBoost, Linear Regression, Random Forest | No | Yes | Simpler regression models achieved competitive results despite limited contextual feature integration. |
| Ranjitha and Spandana [15] | 2021 | XGBoost, Ridge Regression | No | Yes | Ridge Regression and XGBoost outperformed linear and polynomial regression methods. |
| Martins and Galegale [18] | 2023 | Meta-analysis of ML studies | Limited | Partial | Highlighted the importance of preprocessing and identified gaps related to external variable integration. |
| Sim and Wei [19] | 2023 | XGBoost | No | Limited | XGBoost achieved efficient forecasting performance, but external factors were overlooked. |
| Bijalwan et al. [21] | 2023 | ML and Hybrid Models | Seasonality | Limited | Seasonal information improved forecasting accuracy, although the study favored linear regression models. |
| Jewel et al. [22] | 2024 | LSTM, LightGBM | Yes | Yes | LightGBM outperformed LSTM, and incorporating external variables improved prediction accuracy. |
| Chang et al. [30] | 2017 | Deep Neural Networks (DNN) | No | No | DNN improved prediction performance for specific products but lacked preprocessing and contextual integration. |
| Neelakandan et al. [31] | 2023 | DLMNN, SFS | No | Yes | Deep learning models improved RMSE and prediction stability but ignored external market conditions. |

**Table 1.** *Cont.*

| Study | Year | Methodology | External Factors | Pre-Processing | Main Findings/Limitations |
|---|---|---|---|---|---|
| Luyo Ballena et al. [32] | 2024 | RNN, GRU, LSTM | No | Limited | LSTM achieved the best MAPE performance among recurrent architectures. |
| Seyedan et al. [25] | 2022 | Bayesian Model Averaging (BMA), Prophet, LSTM | Yes | Yes | Cluster-based ensemble forecasting improved prediction accuracy compared with individual forecasting models and demonstrated the benefits of combining forecasts across heterogeneous demand patterns. |
| Nasseri et al. [26] | 2023 | Extra Trees, Random Forest, Gradient Boosting, XGBoost, LSTM | Yes | Yes | Tree-based ensemble methods consistently outperformed LSTM models across multiple retail product categories while requiring lower computational complexity and less extensive model tuning. |
| Mishra et al. [27] | 2026 | LightGBM, Random Forest, XGBoost, CatBoost | No | Yes | LightGBM achieved superior forecasting performance, while explainability analysis identified temporal, lag-based, and rolling statistical features as dominant predictors of retail demand. |
| Gerg et al. [28] | 2026 | Linear Regression, ElasticNet, Decision Tree, Random Forest, LightGBM, XGBoost, SGD, Iterative XGBoost | Yes | Yes | Weather-informed models reduced forecast error by 11.5% on average, with XGBoost performing best; gains varied by sector according to demand elasticity. |
| Soleimani et al. [29] | 2026 | Linear Regression, MLP, LSTM, Random Forest, XGBoost, LightGBM, Bagged/Stacked Ensemble | Limited | Yes | LSTM outperformed individual tree-based models, but a bagged Random Forest–LightGBM ensemble ultimately surpassed all models, including LSTM. |
| Hobor et al. [20] | 2026 | XGBoost, LightGBM, N-BEATS, N-HiTS, Temporal Fusion Transformer | Limited | Yes | Localized tree-based ensembles outperformed deep learning architectures on sparse, intermittent demand data while requiring far less training time. |

### 2.4. Contextual Variables in Retail Forecasting

Beyond model selection, forecasting performance is strongly influenced by the availability of contextual information. Retail demand is affected by numerous external and operational factors, including promotional campaigns, holidays, seasonality, pricing policies, store characteristics, and customer behavior. Incorporating such information enables forecasting models to better capture fluctuations in consumer demand and improve prediction accuracy.

Recent studies have demonstrated the importance of contextual variables in retail forecasting. Bijalwan et al. [21] reported that incorporating seasonal information significantly improved forecasting performance. Jewel et al. [22] further showed that the inclusion of promotional and contextual variables improved predictive accuracy for both LightGBM and LSTM models. Similarly, Nasseri et al. [26] incorporated pricing, promotional, weather, and calendar-related variables and found that contextual features contributed substantially to forecasting performance. Most recently, Gerg et al. [28] evaluated the value of weather data across eight machine learning models, three retail sectors, and all 50 U.S. states, finding that weather-informed models reduced forecast error by 11.5% on average, with gains varying systematically by sector according to demand elasticity. These findings reinforce the need to systematically quantify the contribution of contextual variables, which motivates the feature-ablation experiments conducted in this study.

Although contextual variables are frequently incorporated into forecasting frameworks, relatively few studies systematically quantify their individual contribution to forecasting performance. As a result, the relative importance of seasonal, promotional, and customer-related variables remains insufficiently understood.

### 2.5. Data Preprocessing in Forecasting Applications

Data preprocessing represents a critical stage in the machine learning workflow. Retail datasets frequently contain missing values, inconsistent records, categorical variables, and noisy observations that can adversely affect model performance if not appropriately addressed. Common preprocessing tasks include missing-value imputation, categorical encoding, feature scaling, feature selection, and outlier treatment.

Anwar et al. [7] estimated that data preparation activities account for a substantial proportion of the overall analytics process, highlighting the importance of preprocessing for predictive modeling. Previous forecasting studies have demonstrated that preprocessing can significantly influence model performance, particularly when datasets contain heterogeneous feature types and incomplete observations.

Despite its recognized importance, preprocessing is often treated as a secondary component of forecasting research. Many studies report preprocessing procedures only briefly and rarely investigate how individual preprocessing decisions influence forecasting outcomes. Consequently, the relative effectiveness of common preprocessing strategies remains insufficiently explored in retail forecasting applications.

### 2.6. Research Gaps and Study Motivation

The reviewed literature highlights several important research gaps. First, while numerous studies compare forecasting algorithms, relatively few systematically evaluate the influence of preprocessing strategies on forecasting performance. Second, although contextual variables such as promotions, seasonality, and customer-related information are widely recognized as important demand drivers, their individual contribution is rarely quantified through controlled feature-ablation experiments. Third, despite the widespread adoption of ensemble learning methods, comparative evaluations across datasets with substantially different feature characteristics remain limited.

Motivated by these limitations, this study presents a controlled benchmarking evaluation of seven tree-based machine learning models across two publicly available retail forecasting datasets. The study investigates the impact of preprocessing strategies and feature availability through multiple experimental configurations and controlled feature-ablation analyses. By isolating these factors, the study aims to provide a deeper understanding of the conditions under which retail forecasting models achieve their strongest performance and identify the variables that contribute most significantly to forecasting accuracy.

## 3. Materials and Methods

### *3.1. Proposed Approach*

This section describes the experimental framework adopted in this study, structured into the following stages: dataset preparation, experimental configuration, model training and cross-validation, and performance evaluation. Seven tree-based machine learning models are evaluated across multiple experimental configurations on two publicly available retail datasets: BigMart and Rossmann. For the Rossmann dataset, five experimental configurations were evaluated to assess the impact of preprocessing, zero-variance feature removal, and the inclusion or exclusion of seasonal and promotional features. Additionally, an ablation study is conducted to quantify the contribution of the *Customers* variable byrunning all configurations under two parallel settings, with and without the *Customers* feature. For the BigMart dataset, only the first three configurations were applicable, as the dataset does not contain seasonal or promotional features and configurations C4 and C5 are therefore not relevant. Model performance is assessed using MAE, RMSE, and $R^2$, computed on a held-out validation set, with 5-fold cross-validation used separately to assess model stability and generalization, as detailed in Section 3.5. Figure 1 presents an overview of the end-to-end pipeline.

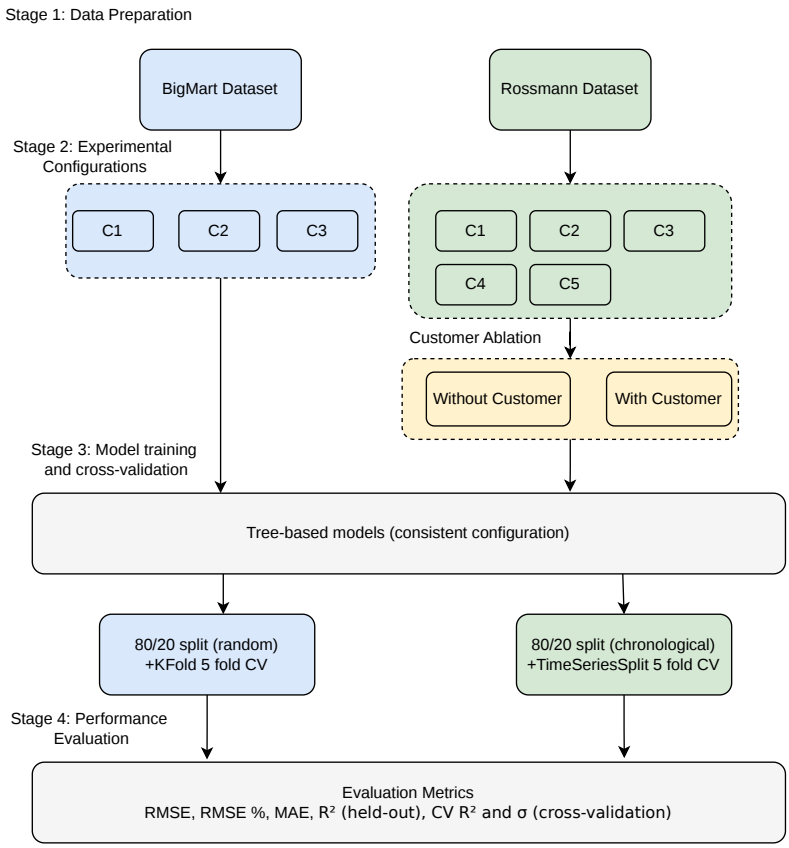



**Figure 1.** Overview of the experimental framework adopted in this study.

The following five configurations were defined for this study:

- C1—Baseline: Only the minimum preprocessing required for model execution is applied, consisting of one-hot encoding of categorical variables. No missing value imputation or zero-variance feature removal is performed.
- C2—Preprocessed: A full preprocessing pipeline is applied, including missing value imputation and one-hot encoding of categorical variables. No zero-variance feature removal is applied.
- C3—Preprocessed + Selected: Builds on C2 by additionally applying zero-variance feature removal (`VarianceThreshold(threshold=0.0)`) to eliminate features with identical values across all records.
- C4—Preprocessed, No Seasonality: Applies the same preprocessing as C2, but excludes seasonal and promotional features using a regex-based pattern matching column names containing the terms: holiday, weather, temp, season, weekend, quarter, weekday, and promo.
- C5—Preprocessed + Selected, No Seasonality: Combines C3 with the same seasonal and promotional feature exclusion applied in C4.

All seven models were trained and evaluated under consistent hyperparameter settings across all configurations, ensuring that observed performance differences are primarily associated with preprocessing and feature availability rather than differences in model-specific tuning. The models and their configurations are described in detail in Section 3.6.

### 3.2. Datasets

This study utilizes two publicly available retail sales datasets that represent complementary forecasting scenarios: the BigMart Sales dataset [33], which contains primarily internal product and outlet attributes, and the Rossmann Store Sales dataset [34], which includes seasonal, promotional, and customer-related variables alongside store characteristics. The contrast between the two datasets enables the evaluation of forecasting performance under different levels of feature availability and contextual information. Table 2 summarizes the key characteristics of both datasets.

**Table 2.** Dataset composition and details.

| Attribute | BigMart | Rossmann |
|---|---|---|
| Number of stores | 10 | 1115 |
| Number of products | 1559 | — |
| Number of features | 12 | 18 |
| Total records | 8523 | 1,017,209 |
| Missing values | Item_Weight, Outlet_Size | Competition and promotion fields |
| Seasonal and promotional features | No | Yes |
| Target variable | Item_Outlet_Sales | Sales |

#### 3.2.1. BigMart Sales Dataset

The BigMart Sales dataset contains sales records covering a wide range of product categories and outlet types. Its features include item type, visibility, outlet type, maximum retail price, outlet year of establishment, and store identifiers. Notably, two features, item weight and outlet size, contain missing values due to inconsistent reporting across stores, requiring appropriate imputation during preprocessing.

#### 3.2.2. Rossmann Store Sales Dataset

The Rossmann Store Sales dataset contains daily sales records from 1115 retail locations operated by Rossmann, a drugstore chain with over 3000 branches across seven European

countries. The dataset is structured across two files: a historical sales file capturing daily transactions, and a supplementary store information file describing store-level characteristics including store type, assortment type, and competition distance. The two files were merged on the store identifier, yielding 18 features prior to preprocessing. The dataset includes seasonal and promotional features, specifically school holidays, public holidays, and promotional indicators that are known to influence retail sales performance. In addition, the dataset contains a *Customers* variable representing the number of customers visiting each store on a given day. The presence of these features makes the Rossmann dataset particularly well-suited for evaluating the impact of seasonal, promotional, and customer-related information on forecasting accuracy, which is the focus of the ablation studies conducted in this study.

### *3.3. Data Preprocessing*

To ensure data quality and consistency across experimental configurations, a structured preprocessing pipeline was applied to both datasets. The pipeline was implemented in a fold-safe manner, meaning that all imputation statistics and encoding schemas were learned exclusively from the training folds and subsequently applied to the corresponding validation folds, thereby preventing data leakage.

#### 3.3.1. Missing Value Handling

Missing values were handled according to the nature of each feature. For numerical features, missing entries were replaced with the column median, as the median is more robust to skewed distributions and less sensitive to outliers, which are common in retail sales data. For categorical features, missing values were imputed using the mode, preserving the most representative category while minimizing the introduction of artificial bias into the distribution. This imputation strategy was applied to both datasets prior to any subsequent transformation steps.

#### 3.3.2. Categorical Variable Encoding

All categorical features in both datasets were transformed using one-hot encoding, which converts each categorical variable into a set of binary indicator columns. This approach is appropriate when categorical values carry no inherent ordinal relationship, as is the case with store types, item categories, and location identifiers present in the datasets. Encoding was performed after the imputation stage so that all categorical values were complete prior to transformation.

### *3.4. Zero-Variance Feature Removal*

Following preprocessing, a zero-variance feature removal procedure was applied using the scikit-learn `VarianceThreshold` function to eliminate features that carry no discriminative information. Specifically, features with zero variance (features with identical values across all records) were removed, as such features carry no predictive signal regardless of the model used. This conservative threshold was deliberately chosen to preserve as much information as possible while eliminating only features that are provably uninformative. The preprocessing configurations evaluated in this study (missing-value imputation, categorical encoding, and zero-variance feature removal) were selected because they address common preprocessing challenges encountered in tabular retail datasets while remaining broadly applicable across all evaluated tree-based models. Feature scaling and normalization were not considered because tree-based models base their split decisions on the ordering of feature values rather than their numerical scale, making these transformations unnecessary for the evaluated algorithms. Likewise, explicit outlier treatment was not included because tree-based models are generally less sensitive to outliers than

distance-based methods. Other preprocessing strategies, such as model-based imputation, alternative categorical encoding schemes, and automated feature engineering, were beyond the scope of this study.

### 3.5. Cross-Validation Strategy

Primary performance metrics were computed on a held-out validation set, while 5-fold cross-validation was used to assess model stability and generalization. Two complementary evaluation approaches were applied to each model. A fixed 80/20 split was used to compute primary performance metrics, RMSE, RMSE%, MAE, and $R^2$ on a held-out validation set. For the BigMart dataset, records were partitioned randomly, consistent with its static, non-temporal tabular structure. For the Rossmann dataset, a chronological split was applied to preserve temporal ordering. Additionally, five-fold cross-validation was applied independently to assess model stability: standard KFold (shuffle = True, random_state = 42) for BigMart and TimeSeriesSplit for Rossmann, ensuring that training data always preceded the corresponding validation data chronologically. The mean and standard deviation of $R^2$ across the five folds are reported as CV $R^2$ and $\sigma$ respectively.

In all cases, preprocessing transformations were fitted exclusively on the training folds and subsequently applied to the corresponding validation folds within a scikit-learn Pipeline, ensuring no information leakage between partitions. For the Rossmann dataset, where chronological splitting and TimeSeriesSplit were employed, all preprocessing parameters (e.g., median and mode values for imputation and encoding mappings) were learned exclusively from the historical training data and then applied to the corresponding future validation data, thereby preventing temporal leakage.

### 3.6. Predictive Models

This study evaluates seven tree-based machine learning models representing a diverse range of ensemble learning strategies, enabling a comparison of boosting-based, bagging-based, and baseline approaches for retail sales forecasting. All models were configured with a fixed random seed (`random_state = 42`) for reproducibility, 100 estimators for all ensemble models, and a maximum depth of 10 for the Decision Tree to prevent overfitting. Hyperparameter optimization was intentionally omitted because the objective of this study is to evaluate the impact of preprocessing configurations and feature availability under a controlled experimental setting. Using consistent model configurations allows performance differences to be attributed primarily to preprocessing and feature variations rather than model-specific tuning.

- Random Forest (RF) is a bagging ensemble that constructs multiple decision trees, each trained on a bootstrap sample of the training data. A random subset of features is evaluated at each node to promote tree diversity and reduce overfitting. The final prediction is the average of all individual tree outputs.
- Extra Trees (ET) or extremely randomized trees, extends the Random Forest approach by using the full training set for each tree and selecting split thresholds randomly rather than optimally. This additional randomness further reduces variance at the cost of a slight increase in bias.
- AdaBoost (Ada) is a sequential boosting algorithm that trains weak learners iteratively, assigning higher weights to poorly predicted instances so that subsequent learners focus on more difficult cases. The final prediction is a weighted combination of all learners.
- Gradient Boosting (GB) fits each new learner to the residual errors of the previous ensemble using gradient descent in function space. Early stopping was applied with a patience of 10 iterations to prevent overfitting on the training data.

- Histogram-Based Gradient Boosting (HGB) is an optimized variant of gradient boosting that discretizes continuous features into histograms before selecting splits, significantly reducing training time and memory consumption compared to standard GB.
- XGBoost (XGB) or extreme gradient boosting, extends the gradient boosting framework by incorporating second-order derivatives of the loss function and regularization terms to control model complexity.
- Decision Tree (DT) is a non-ensemble baseline model that recursively partitions the feature space based on the most informative splits. It serves as an interpretable reference for quantifying the added value of ensemble methods.

### 3.7. Classical Forecasting Baselines

To provide conventional performance references for the evaluated tree-based models, three baseline methods were included in the experimental evaluation.

- Ridge Regression is a linear model with L2 regularization ($\alpha = 1.0$), used as a representative linear baseline.
- Seasonal-Naive (lag-7) is a heuristic baseline that predicts each day's sales as the actual sales value from the same weekday in a prior week, using a backoff over recent weeks with a training-period store-mean fallback when no match is available.
- ETS (per-store) applies Holt–Winters exponential smoothing independently to each store's sales series, with an additive, damped trend component and additive weekly seasonality (seasonal_periods $= 7$).

Seasonal-Naive and ETS require a temporal ordering and weekly seasonal structure that is present in the Rossmann dataset but absent from BigMart, and were therefore applied to Rossmann only; Ridge Regression, requiring no such structure, was applied to both datasets. Together, these baselines provide complementary conventional forecasting references spanning linear regression, heuristic seasonal forecasting, and classical exponential smoothing, against which the tree-based ensemble models are compared. All baselines were evaluated using the best-performing preprocessing configuration (Configuration 3). For Rossmann, the comparison was conducted only under the without-*Customers* scenario, as real-time customer counts are generally unavailable at forecast time, whereas the with-*Customers* experiments are retained solely as a controlled feature-ablation analysis. The baselines were evaluated using the same held-out validation split and 5-fold cross-validation protocol (KFold for BigMart and TimeSeriesSplit for Rossmann) adopted for the tree-based models.

### 3.8. Evaluation Metrics

To evaluate the predictive performance of the evaluated models, three widely adopted regression metrics were employed: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the Coefficient of Determination ($R^2$).

#### 3.8.1. Mean Absolute Error (MAE)

MAE measures the average magnitude of errors between predicted and actual values, treating all deviations equally regardless of direction. It is defined as:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (1)$$

where $n$ is the number of observations, $y_i$ is the actual value, and $\hat{y}_i$ is the predicted value. MAE is straightforward to interpret and relatively robust to outliers; however, it does not penalize large errors proportionally and is not differentiable at zero, which limits its suitability for gradient-based optimization.

3.8.2. Root Mean Square Error (RMSE)

RMSE measures the square root of the average squared differences between predicted and actual values, and is expressed in the same units as the target variable, facilitating direct interpretation of error magnitude. It is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (2)$$

RMSE penalizes large errors more heavily than MAE due to the squaring operation, making it particularly sensitive to outliers. It is commonly used alongside MAE to provide a more comprehensive assessment of model accuracy.

3.8.3. RMSE Percentage (RMSE%)

To facilitate comparison across datasets with different target-value scales, RMSE is additionally reported as a percentage of the mean target value in the training data. It is defined as:

$$\text{RMSE\%} = \frac{\text{RMSE}}{\bar{y}_{\text{train}}} \times 100 \quad (3)$$

where $\bar{y}_{\text{train}}$ is the mean of the target values in the training set. Using the training-set mean, rather than statistics from the evaluation set, provides a fixed normalization factor for each dataset. The same training-set mean is used for all models evaluated on a given dataset, ensuring consistent normalization across models. Consequently, RMSE% enables direct comparison of prediction errors across datasets and experimental configurations with different target-value scales, which is not possible using RMSE alone.

3.8.4. Coefficient of Determination ($R^2$)

$R^2$ quantifies the proportion of variance in the dependent variable that is explained by the independent variables in the model. It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (4)$$

where $\bar{y}$ is the mean of the actual values. $R^2$ ranges from 0 to 1, where a value of 1 indicates a perfect fit. Higher values reflect stronger explanatory power of the model.

*3.9. Feature Importance and SHAP Analysis*

Feature importance and SHapley Additive exPlanations (SHAP) analyses were conducted for the best-performing model for each experimental scenario to identify the variables that contributed most strongly to forecasting performance and to provide insight into the predictive patterns learned by the models. Tree-based feature importance provides a global ranking of predictor importance, whereas SHAP explains both the magnitude and direction of each feature's contribution to individual predictions, providing complementary global and local interpretations of model behavior. Each model was refitted using the same preprocessing pipeline, train-validation split, hyperparameter configuration, and random seed (random_state = 42) as in the main experiments. TreeExplainer was used to compute SHAP values for the tree-based models. To maintain a consistent interpretability protocol and computational efficiency across all experimental scenarios, SHAP values were computed from a reproducible simple random sample of 500 validation observations (random_state = 42). SHAP explanations were generated exclusively from the validation data to interpret model behavior on previously unseen observations.

Because one-hot encoding expands categorical variables into numerous binary columns, the contribution of a single conceptual variable may be distributed across multi-

ple encoded features. Therefore, in addition to reporting mean absolute SHAP values for individual features, SHAP values were aggregated into predefined feature categories to facilitate higher-level interpretation. The categories comprised Pricing, Product, and Store for the BigMart dataset, and Promotional, Competition, Store, Calendar/Temporal, and, where applicable, Customer for the Rossmann dataset. Model interpretation was based on mean absolute SHAP values for individual features and predefined categories, SHAP summary plots, and dependence plots for the most influential features.

### 3.10. Experiment Environment

All experiments were implemented in Python 3.11.7 , utilizing a set of well-established libraries to support the machine learning workflow, namely Scikit-learn, NumPy 1.26.4, Pandas 2.1.4, Matplotlib 3.8.0, Joblib 1.5.3, and Seaborn 0.12.2. The hardware platform consisted of a MacBook Pro (Apple, Cupertino, CA, USA) equipped with an Intel Core i7 processor (Intel Corporation, Santa Clara, CA, USA) and 16 GB of RAM.

## 4. Results and Discussion

This section reports and discusses the performance of the seven evaluated models across all experimental configurations on the BigMart and Rossmann datasets. For Rossmann, two additional ablation studies are conducted to quantify the independent contributions of seasonal and promotional features and the *Customers* variable.

### 4.1. BigMart Dataset Results

Table 3 presents the performance of all seven evaluated models across the three experimental configurations in the BigMart dataset, evaluated using RMSE, RMSE %, MAE, $R^2$, and cross-validation $R^2$ (CV $R^2$).

The results reveal several consistent patterns across all configurations. GradientBoosting achieved the highest overall performance, attaining an $R^2$ of 0.611, RMSE of 1028.20, and MAE of 723.10 under C3, with a cross-validation $R^2$ of 0.597 ($\sigma$ = 0.012), indicating stable generalisation across folds. HistGradientBoosting performed comparably, achieving $R^2$ = 0.601 and RMSE% = 47.74% under C2 and C3, confirming the dominance of gradient-based boosting methods on this dataset. In contrast, AdaBoost was the weakest model across all configurations, with a best $R^2$ of only 0.440 and notably high CV $R^2$ variance ($\sigma$ = 0.149 in C1), suggesting instability and sensitivity to the feature space characteristics of BigMart.

Regarding the effect of preprocessing, the progression from C1 to C2 had a mixed effect across models. GradientBoosting showed the most meaningful improvement ($R^2$: 0.589 $\rightarrow$ 0.611), while RandomForest, ExtraTrees, XGBoost, and HistGradientBoosting remained largely stable (differences $\leq$ 0.01), and DecisionTree showed a modest improvement ($R^2$: 0.537 $\rightarrow$ 0.549). AdaBoost declined notably ($R^2$: 0.440 $\rightarrow$ 0.383), suggesting that its boosting mechanism is less compatible with the encoded categorical features introduced during preprocessing.

The subsequent application of zero variance feature removal in C3 produced negligible changes for GradientBoosting, HistGradientBoosting, RandomForest, and XGBoost (differences $\leq$ 0.001 $R^2$), while AdaBoost, DecisionTree, and ExtraTrees showed small but non-negligible gains (+0.011, +0.009, and +0.005 respectively), confirming that the selected feature subset retained most predictive information present in the preprocessed feature configuration. Across all evaluated models, predictive performance remained relatively stable across configurations, with the best result reaching $R^2 = 0.611$. This consistency suggests that the limited contextual information available in the BigMart dataset, which lacks temporal, seasonal, and promotional features, may constrain the predictive perfor-

mance achievable by the evaluated tree-based models. However, this observation should not be interpreted as an intrinsic upper bound of the dataset, since alternative feature engineering strategies, model architectures, or optimization approaches were beyond the scope of this study.

**Table 3.** Performance of the evaluated models on the BigMart dataset across three configurations. Bold values indicate the best result per metric. C1: Baseline; C2: Preprocessed; C3: Preprocessed + Selected Features.

| Configuration | Model | RMSE | RMSE% | MAE | $R^2$ | CV $R^2$ |
|---|---|---|---|---|---|---|
| C1: Baseline | AdaBoost | 1234.11 | 56.58 | 968.44 | 0.4396 | 0.3448 |
| | DecisionTree | 1121.73 | 51.43 | 765.66 | 0.5370 | 0.5405 |
| | ExtraTrees | 1141.44 | 52.33 | 796.57 | 0.5206 | 0.5073 |
| | GradientBoosting | 1056.43 | 48.43 | 761.72 | 0.5894 | 0.5791 |
| | HistGradientBoosting | 1034.69 | 47.43 | 728.93 | 0.6061 | 0.5904 |
| | RandomForest | 1097.34 | 50.31 | 763.14 | 0.5570 | 0.5519 |
| | XGBoost | 1081.00 | 49.56 | 747.86 | 0.5701 | 0.5673 |
| C2: Preprocessed | AdaBoost | 1294.52 | 59.35 | 1065.91 | 0.3834 | 0.4042 |
| | DecisionTree | 1106.72 | 50.74 | 761.38 | 0.5494 | 0.5454 |
| | ExtraTrees | 1143.07 | 52.40 | 798.50 | 0.5193 | 0.5155 |
| | GradientBoosting | 1028.45 | 47.15 | 723.22 | 0.6108 | 0.5969 |
| | HistGradientBoosting | 1041.45 | 47.74 | 725.75 | 0.6009 | 0.5887 |
| | RandomForest | 1085.62 | 49.77 | 755.70 | 0.5664 | 0.5558 |
| | XGBoost | 1080.94 | 49.55 | 745.47 | 0.5701 | 0.5715 |
| C3: Preprocessed + Selected | AdaBoost | 1282.91 | 58.81 | 1038.90 | 0.3945 | 0.4155 |
| | DecisionTree | 1095.31 | 50.21 | 758.28 | 0.5586 | 0.5495 |
| | ExtraTrees | 1137.40 | 52.14 | 794.79 | 0.5240 | 0.5150 |
| | **GradientBoosting** | **1028.20** | **47.14** | **723.10** | **0.6110** | **0.5970** |
| | HistGradientBoosting | 1041.45 | 47.74 | 725.75 | 0.6009 | 0.5887 |
| | RandomForest | 1085.62 | 49.77 | 757.11 | 0.5664 | 0.5556 |
| | XGBoost | 1080.94 | 49.55 | 745.47 | 0.5701 | 0.5715 |

Figure 2 illustrates the actual versus predicted sales for the best-performing model, GradientBoosting under C3. The predictions align well with actual values for low-to-mid sales ranges, closely following the perfect prediction line. However, increased scatter is observed at higher sales values, indicating that the model is less accurate for high-value transactions. This is expected in retail sales data, where large transactions are less frequent and therefore underrepresented during training.

These findings are consistent with the work of Swami et al. [13], who reported that XGBoost-based boosting methods achieved strong retail forecasting performance, and with Cheriyan et al. [11], who found Gradient Boosting techniques to outperform traditional regression models. However, unlike Bajaj et al. [14], who identified Random Forest as the best-performing tree-based model, our experiments indicate that GradientBoosting performs best on the BigMart dataset, suggesting that the relative superiority of boosting or bagging methods depends on the underlying dataset characteristics and available feature set.

### *4.2. Rossmann Dataset Results*

The Rossmann dataset introduces an additional experimental dimension absent in BigMart: the *Customers* variable, which records the number of customers visiting each store on a given day. Since customer footfall is a strong predictor of daily sales, yet is typically unavailable at the time of forecasting in real-world deployment scenarios, two parallel sets of experiments were conducted. The first excludes the *Customers* feature to reflect a realistic forecasting setting, while the second includes it to quantify its contribution to

predictive performance. This design enables a controlled ablation study of the *Customers* feature, and the results are reported in two separate tables accordingly.

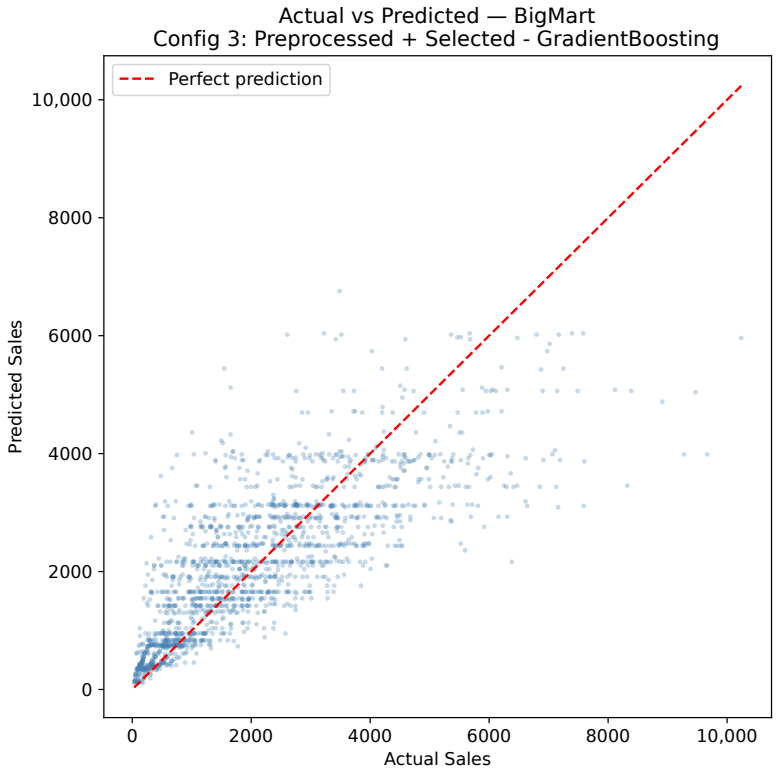



**Figure 2.** Actual vs. predicted sales for the best-performing model (GradientBoosting, C3: Preprocessed + Selected) on the BigMart dataset. The dashed red line represents perfect prediction.

#### 4.2.1. Results Without Customers

Table 4 presents the performance of all models across the five configurations when the *Customers* feature is excluded.

Without the *Customers* feature, RandomForest achieved the highest performance under C3, attaining an $R^2$ of 0.884, RMSE of 1044.25, and MAE of 701.01, with a CV $R^2$ of 0.832, confirming stable generalisation across temporal folds. ExtraTrees ranked second ($R^2$ = 0.872, CV $R^2$ = 0.822) and XGBoost third ($R^2$ = 0.827). The consistent superiority of RandomForest and ExtraTrees across configurations suggests that bagging-based ensemble methods are particularly effective for the Rossmann dataset. This finding is consistent with with the findings of Bajaj et al. [14], who reported Random Forest as the most accurate tree-based regressor for retail sales prediction.

AdaBoost was the weakest model, producing negative $R^2$ values across all configurations (as low as $-3.211$ in C1), indicating performance worse than a naive mean predictor. This finding is consistent with its performance on the BigMart dataset, where it also ranked last among the evaluated models. GradientBoosting likewise performed relatively poorly ($R^2 \approx 0.38$ across C1–C3), in contrast to its strong performance on the BigMart dataset. This difference highlights the extent to which model effectiveness depends on dataset characteristics and feature composition.

The progression from C1 through C2 to C3 had a mixed effect across models. At the preprocessing stage (C1 $\rightarrow$ C2), RandomForest showed the most meaningful improvement ($\Delta R^2 = +0.028$), followed by XGBoost ($+0.019$) and ExtraTrees ($+0.015$), while HistGradientBoosting, GradientBoosting, and DecisionTree showed small declines of $-0.004$, $-0.011$, and $-0.049$ respectively. The subsequent feature filtering step (C2 $\rightarrow$ C3) produced no meaningful change for any model (differences $\leq 0.001$ in $R^2$), suggesting that the removed

zero-variance features carried negligible predictive information, consistent with the BigMart findings.

**Table 4.** Performance of the evaluated models on the Rossmann dataset (excluding *Customers*) across five configurations. Bold values indicate the best result per metric. C1: Baseline; C2: Preprocessed; C3: Preprocessed + Selected; C4: Preprocessed, No Seasonality/promotional features; C5: Preprocessed + Selected, No Seasonality/promotional features.

| Config | Model | RMSE | RMSE% | MAE | $R^2$ | CV $R^2$ |
|---|---|---|---|---|---|---|
| C1 | AdaBoost | 6286.54 | 90.38 | 5581.30 | −3.2106 | −0.3705 |
| | DecisionTree | 2193.71 | 31.54 | 1585.27 | 0.4873 | 0.4650 |
| | ExtraTrees | 1161.15 | 16.69 | 781.30 | 0.8564 | 0.8143 |
| | GradientBoosting | 2400.32 | 34.51 | 1733.82 | 0.3862 | 0.3739 |
| | HistGradientBoosting | 1815.76 | 26.11 | 1316.79 | 0.6487 | 0.6361 |
| | RandomForest | 1161.65 | 16.70 | 781.61 | 0.8562 | 0.8173 |
| | XGBoost | 1341.28 | 19.28 | 937.16 | 0.8083 | 0.7746 |
| C2 | AdaBoost | 3899.48 | 56.06 | 3241.38 | −0.6201 | −1.0091 |
| | DecisionTree | 2295.76 | 33.01 | 1658.27 | 0.4385 | 0.4356 |
| | ExtraTrees | 1099.24 | 15.80 | 740.05 | 0.8713 | 0.8219 |
| | GradientBoosting | 2420.96 | 34.81 | 1736.54 | 0.3755 | 0.3773 |
| | HistGradientBoosting | 1825.77 | 26.25 | 1333.61 | 0.6448 | 0.6267 |
| | RandomForest | 1044.42 | 15.02 | 701.28 | 0.8838 | 0.8321 |
| | XGBoost | 1274.18 | 18.32 | 902.35 | 0.8270 | 0.7873 |
| C3 | AdaBoost | 3899.48 | 56.06 | 3241.38 | −0.6201 | −1.0091 |
| | DecisionTree | 2295.82 | 33.01 | 1658.25 | 0.4384 | 0.4389 |
| | ExtraTrees | 1095.94 | 15.76 | 737.62 | 0.8720 | 0.8217 |
| | GradientBoosting | 2420.96 | 34.81 | 1736.54 | 0.3755 | 0.3773 |
| | HistGradientBoosting | 1825.77 | 26.25 | 1333.61 | 0.6448 | 0.6267 |
| | **RandomForest** | **1044.25** | **15.01** | **701.01** | **0.8838** | **0.8319** |
| | XGBoost | 1274.18 | 18.32 | 902.35 | 0.8270 | 0.7873 |
| C4 | AdaBoost | 4489.09 | 64.54 | 3735.84 | −1.1470 | −0.5807 |
| | DecisionTree | 2489.65 | 35.79 | 1862.38 | 0.3396 | 0.3036 |
| | ExtraTrees | 1921.74 | 27.63 | 1435.64 | 0.6065 | 0.5800 |
| | GradientBoosting | 2702.62 | 38.86 | 2014.61 | 0.2218 | 0.2044 |
| | HistGradientBoosting | 2192.86 | 31.53 | 1646.26 | 0.4877 | 0.4696 |
| | RandomForest | 1921.67 | 27.63 | 1435.62 | 0.6066 | 0.5773 |
| | XGBoost | 1956.51 | 28.13 | 1457.53 | 0.5922 | 0.5691 |
| C5 | AdaBoost | 4489.09 | 64.54 | 3735.84 | −1.1470 | −0.5807 |
| | DecisionTree | 2489.65 | 35.79 | 1862.38 | 0.3396 | 0.3036 |
| | ExtraTrees | 1921.74 | 27.63 | 1435.64 | 0.6065 | 0.5800 |
| | GradientBoosting | 2702.62 | 38.86 | 2014.61 | 0.2218 | 0.2044 |
| | HistGradientBoosting | 2192.86 | 31.53 | 1646.26 | 0.4877 | 0.4696 |
| | RandomForest | 1921.67 | 27.63 | 1435.62 | 0.6066 | 0.5773 |
| | XGBoost | 1956.51 | 28.13 | 1457.53 | 0.5922 | 0.5691 |

The more central contribution of the Rossmann experimental design lies in the ablation of seasonal and promotional features, discussed in Section 4.2.4.

The ablation of seasonal and promotional features in C4 and C5 resulted in substantial performance degradation across all models. RandomForest dropped from $R^2$ = 0.884 (C3) to 0.607 (C4), a decline of 0.277 points, with RMSE% nearly doubling from 15.01% to 27.63%. The identical results between C4 and C5 further confirm that feature filtering cannot recover predictive information once seasonal and promotional features are removed.

#### 4.2.2. Results with Customers

Table 5 presents the performance of all models across the five configurations when the *Customers* feature is included. The inclusion of the *Customers* feature produced a substantial improvement in predictive performance across all models. Under C3, ExtraTrees and

RandomForest achieved virtually identical best overall results, with $R^2$ = 0.965 for both and RMSE% of 8.23% and 8.26% respectively. The difference between the two models is negligible (0.0003 in $R^2$), and both demonstrated exceptionally stable cross-validation $R^2$ values around 0.96 ($\sigma < 0.01$), confirming equally robust generalisation across temporal folds. Compared to the without-Customers scenario, the best $R^2$ improved from 0.884 to 0.965, an increase of 0.081, while RMSE% was nearly halved from 15.01% to 8.23%, underscoring the dominant predictive role of customer footfall in daily sales forecasting.

**Table 5.** Performance of the evaluated models on the Rossmann dataset (including *Customers*) across five configurations. Bold values indicate the best result per metric. Configuration labels as in Table 4.

| Config | Model | RMSE | RMSE% | MAE | $R^2$ | CV $R^2$ |
|---|---|---|---|---|---|---|
| C1 | AdaBoost | 1951.39 | 28.06 | 1553.70 | 0.5943 | 0.5446 |
| | DecisionTree | 1080.17 | 15.53 | 786.81 | 0.8757 | 0.8787 |
| | ExtraTrees | 627.18 | 9.02 | 440.93 | 0.9581 | 0.9560 |
| | GradientBoosting | 1108.14 | 15.93 | 795.34 | 0.8692 | 0.8776 |
| | HistGradientBoosting | 863.13 | 12.41 | 621.97 | 0.9206 | 0.9245 |
| | RandomForest | 598.75 | 8.61 | 417.30 | 0.9618 | 0.9597 |
| | XGBoost | 670.32 | 9.64 | 467.23 | 0.9521 | 0.9533 |
| C2 | AdaBoost | 2267.38 | 32.60 | 1904.98 | 0.4523 | 0.4208 |
| | DecisionTree | 1101.02 | 15.83 | 798.53 | 0.8708 | 0.8774 |
| | ExtraTrees | 572.14 | 8.23 | 389.37 | 0.9651 | 0.9598 |
| | GradientBoosting | 1116.97 | 16.06 | 800.09 | 0.8671 | 0.8783 |
| | HistGradientBoosting | 868.99 | 12.49 | 625.09 | 0.9195 | 0.9243 |
| | RandomForest | 574.68 | 8.26 | 391.28 | 0.9648 | 0.9609 |
| | XGBoost | 641.62 | 9.22 | 448.20 | 0.9561 | 0.9543 |
| C3 | AdaBoost | 2267.38 | 32.60 | 1904.98 | 0.4523 | 0.4208 |
| | DecisionTree | 1101.01 | 15.83 | 798.52 | 0.8708 | 0.8775 |
| | **ExtraTrees** | **572.18** | **8.23** | **389.13** | **0.9651** | **0.9598** |
| | GradientBoosting | 1116.97 | 16.06 | 800.09 | 0.8671 | 0.8783 |
| | HistGradientBoosting | 868.99 | 12.49 | 625.09 | 0.9195 | 0.9243 |
| | **RandomForest** | **574.62** | **8.26** | **391.28** | **0.9648** | **0.9609** |
| | XGBoost | 641.62 | 9.22 | 448.20 | 0.9561 | 0.9543 |
| C4 | AdaBoost | 1735.95 | 24.96 | 1334.89 | 0.6789 | 0.5804 |
| | DecisionTree | 1204.09 | 17.31 | 879.53 | 0.8455 | 0.8549 |
| | ExtraTrees | 812.44 | 11.68 | 572.84 | 0.9297 | 0.9336 |
| | GradientBoosting | 1276.72 | 18.36 | 929.68 | 0.8263 | 0.8402 |
| | HistGradientBoosting | 1040.08 | 14.95 | 756.57 | 0.8847 | 0.8937 |
| | RandomForest | 768.79 | 11.05 | 542.64 | 0.9370 | 0.9408 |
| | XGBoost | 829.73 | 11.93 | 583.86 | 0.9267 | 0.9331 |
| C5 | AdaBoost | 1735.95 | 24.96 | 1334.89 | 0.6789 | 0.5979 |
| | DecisionTree | 1204.09 | 17.31 | 879.53 | 0.8455 | 0.8544 |
| | ExtraTrees | 812.29 | 11.68 | 572.90 | 0.9297 | 0.9336 |
| | GradientBoosting | 1276.72 | 18.36 | 929.68 | 0.8263 | 0.8402 |
| | HistGradientBoosting | 1040.08 | 14.95 | 756.57 | 0.8847 | 0.8937 |
| | RandomForest | 768.77 | 11.05 | 542.63 | 0.9370 | 0.9408 |
| | XGBoost | 829.73 | 11.93 | 583.86 | 0.9267 | 0.9331 |

GradientBoosting recovered substantially with the inclusion of *Customers*, improving from $R^2$ = 0.376 (C3, without *Customers*) to 0.867 (C3, with *Customers*), suggesting that the Customers feature provides a key signal that compensates for its difficulty in modelling the dataset's temporal structure independently. In contrast, AdaBoost remained the weakest model despite the inclusion of *Customers*, achieving $R^2$ = 0.452 under C3, positive but far below all other models, consistent with its consistently poor performance across both datasets.

The progression from C1 through C2 to C3 was largely stable for most models, with the majority showing differences of less than 0.005 in $R^2$ across configurations. The main

exception was ExtraTrees which showed a marginally larger shift (+0.007), and AdaBoost, which declined notably from C1 to C2 ($R^2$: 0.594 → 0.452), suggesting sensitivity to the preprocessing transformations when *Customers* is present. Feature filtering in C3 produced no meaningful change for any model (differences $\leq 0.001$), suggesting that the removed zero-variance features carried negligible predictive information, consistent with findings on both BigMart and Rossmann without *Customers*.

#### 4.2.3. Impact of the Customers Feature

To quantify the contribution of the *Customers* variable to predictive performance, Table 6 presents a direct comparison of each model's $R^2$ and RMSE% under C3 across both scenarios, along with the absolute improvement (Δ) achieved by including *Customers*.

**Table 6.** Impact of the *Customers* feature on model performance under C3 (Preprocessed + Selected). $\Delta R^2$ and ΔRMSE% reflect the change when *Customers* is included. Negative ΔRMSE% indicates improvement.

| Model | $R^2$ (w/o) | $R^2$ (w/) | $\Delta R^2$ | RMSE% (w/o) | RMSE% (w/) | ΔRMSE% |
|---|---|---|---|---|---|---|
| AdaBoost | −0.6201 | 0.4523 | +1.0724 | 56.06 | 32.60 | −23.46 |
| DecisionTree | 0.4384 | 0.8708 | +0.4324 | 33.01 | 15.83 | −17.18 |
| ExtraTrees | 0.8720 | 0.9651 | +0.0931 | 15.76 | 8.23 | −7.53 |
| GradientBoosting | 0.3755 | 0.8671 | +0.4916 | 34.81 | 16.06 | −18.75 |
| HistGradientBoosting | 0.6448 | 0.9195 | +0.2747 | 26.25 | 12.49 | −13.76 |
| RandomForest | 0.8838 | 0.9648 | +0.0810 | 15.01 | 8.26 | −6.75 |
| XGBoost | 0.8270 | 0.9561 | +0.1291 | 18.32 | 9.22 | −9.10 |

The inclusion of the *Customers* feature consistently improved performance across all seven models, confirming its dominant role as a sales predictor in the Rossmann dataset. The magnitude of improvement, however, varied considerably depending on the model's baseline capability. Models that already performed well without *Customers*, namely RandomForest ($\Delta R^2 = +0.081$) and ExtraTrees ($\Delta R^2 = +0.093$) gained relatively modest improvements, as their tree-based structures were already able to extract strong predictive signal from the available temporal and promotional features. In contrast, models that struggled without *Customers* benefited dramatically: GradientBoosting improved by +0.492 in $R^2$ (from 0.376 to 0.867), DecisionTree by +0.432, and AdaBoost by +1.072, recovering from a negative $R^2$ of −0.620 to a positive 0.452. This pattern suggests that *Customers* acts as a key signal that compensates for the inability of weaker models to capture the dataset's temporal complexity independently.

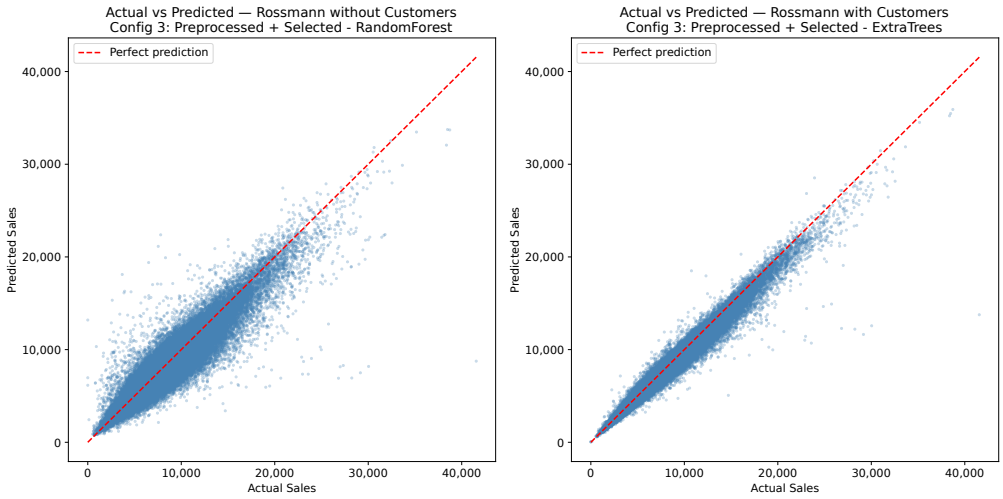
The performance improvement obtained after including customer footfall is consistent with previous studies emphasizing the importance of contextual variables in retail forecasting. For example, Jewel et al. [22] demonstrated that incorporating additional contextual information significantly improves predictive accuracy, while Nasseri et al. [26] reported similar benefits from integrating pricing, promotional, weather, and calendar-related variables. Our results extend these findings by quantifying, through a controlled feature-ablation experiment, the individual contribution of customer footfall to forecasting performance across multiple tree-based ensemble models.

It is worth noting that the model ranking shifts when *Customers* is included. Without *Customers*, RandomForest is the top performer ($R^2 = 0.884$); with *Customers*, ExtraTrees and RandomForest are effectively tied ($R^2 = 0.965$ for both). This consistency at the top of the ranking further reinforces the suitability of bagging-based ensemble methods for structured retail time-series forecasting.

It is important to note that the *Customers* variable, while highly predictive, is typically unavailable prior to the forecasting period in operational retail settings. The results in Table 6 therefore serve as an upper-bound reference, quantifying the maximum performance gain achievable under ideal feature availability. The without-Customers scenario remains

the more operationally relevant benchmark, with RandomForest achieving $R^2 = 0.884$ and RMSE% = 15.01% under C3.

Figure 3 further illustrates this contrast visually; the tighter alignment with the perfect prediction line when *Customers* is included confirms the substantial predictive gain quantified in Table 6.




(**a**) (**b**)

**Figure 3.** Actual vs. predicted sales for the best-performing models on the Rossmann dataset. (**a**) Without *Customers* (RandomForest, C3); (**b**) With *Customers* (ExtraTrees, C3).

#### 4.2.4. Impact of Seasonal and Promotional Features

To assess the contribution of seasonal and promotional features, comprising promotional indicators and holiday flags, Table 7 presents a direct comparison of model performance under C3 (full feature set) against C4 (seasonal and promotional features removed), across both experimental scenarios. Since C5 produced results identical to C4 across all models, the comparison focuses on C3 versus C4.

The ablation results reveal that seasonal and promotional features are critical for accurate sales forecasting on the Rossmann dataset, particularly in the absence of customer footfall data. Without *Customers*, removing these features caused severe degradation across all capable models. RandomForest suffered the largest absolute drop, declining from $R^2 = 0.884$ (C3) to 0.607 (C4), a reduction of 0.277, with RMSE% increasing by 12.62 percentage points. ExtraTrees showed a comparable decline ($\Delta R^2 = -0.266$, $\Delta$RMSE% = +11.87), while XGBoost dropped by 0.235 in $R^2$. These results confirm that promotional campaigns, public holidays, and competition proximity collectively serve as primary drivers of sales variation in the Rossmann dataset, and their removal renders even the strongest models unable to distinguish meaningful patterns from noise. These findings are consistent with Bijalwan et al. [21], who reported that seasonal information substantially improves forecasting accuracy. Unlike previous studies that simply incorporated seasonal variables, our controlled ablation experiments explicitly quantify their independent contribution to predictive performance.

When the *Customers* feature is present, the impact of removing seasonality and promotional features is substantially mitigated. The largest drop among capable models was only $\Delta R^2 = -0.035$ for ExtraTrees and $-0.028$ for RandomForest, far smaller than the corresponding drops in the without-Customers scenario. This indicates that *Customers* partially captures the predictive role of seasonality features, likely because customer footfall itself is influenced by promotions and holidays. Nevertheless, the consistent, if modest,

performance drops across all models confirm that seasonality and promotional features carry independent predictive value beyond what is already encoded in customer counts.

**Table 7.** Impact of removing seasonality and promotional features (C3 vs. C4) across both experimental scenarios. Δ values reflect the change from C3 to C4; negative $\Delta R^2$ and positive ΔRMSE% indicate performance degradation.

| Model | Without Customers | | | | With Customers | | | |
|---|---|---|---|---|---|---|---|---|
| | $R^2$ C3 | $R^2$ C4 | $\Delta R^2$ | ΔRMSE% | $R^2$ C3 | $R^2$ C4 | $\Delta R^2$ | ΔRMSE% |
| AdaBoost | −0.620 | −1.147 | −0.527 | +8.48 | 0.452 | 0.679 | +0.227 | −7.64 |
| DecisionTree | 0.438 | 0.340 | −0.099 | +2.78 | 0.871 | 0.846 | −0.025 | +1.48 |
| ExtraTrees | 0.872 | 0.607 | −0.266 | +11.87 | 0.965 | 0.930 | −0.035 | +3.45 |
| GradientBoosting | 0.376 | 0.222 | −0.154 | +4.05 | 0.867 | 0.826 | −0.041 | +2.30 |
| HistGradientBoosting | 0.645 | 0.488 | −0.157 | +5.28 | 0.920 | 0.885 | −0.035 | +2.46 |
| RandomForest | 0.884 | 0.607 | −0.277 | +12.62 | 0.965 | 0.937 | −0.028 | +2.79 |
| XGBoost | 0.827 | 0.592 | −0.235 | +9.81 | 0.956 | 0.927 | −0.029 | +2.71 |

A notable exception is AdaBoost in the with-Customers scenario, which unexpectedly improved when seasonality features were removed ($\Delta R^2 = +0.227$). One possible explanation is that this result reflects AdaBoost's known sensitivity to noisy or complex feature spaces: removing seasonality features may have simplified the feature space enough for AdaBoost's shallow base learners to perform better, though its absolute performance ($R^2 = 0.679$) remains far below that of the leading models.

Taken together, the two ablation studies, seasonality features and the *Customers* variable, provide a comprehensive picture of the relative contribution of each feature group to Rossmann sales predictability. Seasonality and promotional features are the dominant driver when customer data is unavailable, while *Customers* provides the strongest single-feature signal when available, with both contributing independently to overall model performance.

### 4.3. Comparison with Classical Forecasting Baselines

Table 8 compares the best-performing tree-based models against the classical baselines described in Section 3.7, evaluated under the identical experimental protocol described therein.

On the BigMart dataset, GradientBoosting substantially outperformed the linear baseline, improving the held-out $R^2$ from 0.406 to 0.611 while reducing RMSE by approximately 19%. Similar improvements were observed for MAE and RMSE%, indicating that the nonlinear tree-based ensemble captured predictive relationships beyond those represented by the linear baseline. Both models also exhibited stable cross-validation performance, with low variability across folds.

**Table 8.** Comparison between the best-performing tree-based models and conventional forecasting baselines under Configuration 3 (Preprocessed + Selected) and without the *Customers* feature for Rossmann. Bold values indicate the best result per metric within each dataset.

| Dataset | Model | RMSE | RMSE% | MAE | $R^2$ | CV $R^2$ |
|---|---|---|---|---|---|---|
| BigMart | Ridge Regression | 1270.90 | 58.26 | 941.86 | 0.4057 | 0.4188 |
| | **GradientBoosting** | **1028.20** | **47.14** | **723.10** | **0.6110** | **0.5970** |
| Rossmann (without *Customers*) | Ridge Regression | 2695.51 | 38.75 | 1958.88 | 0.2259 | 0.1891 |
| | Seasonal-Naive | 1956.36 | 28.13 | 1466.33 | 0.5922 | 0.4884 |
| | ETS | 1823.27 | 26.21 | 1347.27 | 0.6458 | 0.4887 |
| | **RandomForest** | **1044.25** | **15.01** | **701.01** | **0.8838** | **0.8319** |

On the Rossmann dataset, ETS and the Seasonal-Naive method showed broadly similar performance (held-out $R^2$ of 0.646 and 0.592, respectively) and both clearly outperformed Ridge Regression ($R^2 = 0.226$), which does not explicitly model temporal structure. ETS achieved a modestly lower RMSE and higher held-out $R^2$ than Seasonal-Naive, although its cross-validation performance was considerably more variable across folds (CV $R^2 = 0.489$, $\sigma = 0.122$, versus $\sigma = 0.042$ for Seasonal-Naive), suggesting less consistent generalization. Nevertheless, RandomForest consistently outperformed both statistical baselines, increasing the held-out $R^2$ to 0.884 (CV $R^2 = 0.832$, $\sigma = 0.052$) while reducing RMSE by approximately 43% relative to ETS, the strongest classical baseline by held-out $R^2$.

Overall, these results demonstrate that the evaluated tree-based models, particularly the best-performing ensemble methods, consistently outperform well-established conventional forecasting approaches. Together with the preprocessing and feature-ablation analyses, these findings provide further empirical evidence supporting the suitability of tree-based ensemble methods for retail sales forecasting under realistic feature availability.

### *4.4. Statistical Significance Analysis*

Tables 9 and 10 summarize the statistical significance analyses for the BigMart and Rossmann datasets, respectively. For BigMart, pairwise comparisons were performed using the Wilcoxon signed-rank test on absolute and squared prediction errors, with Holm correction applied to account for multiple comparisons. For Rossmann, where observations exhibit temporal dependence, the Diebold–Mariano (DM) test was applied to daily forecasting errors using both absolute- and squared-error loss functions, again with Holm correction for multiple comparisons.

For the BigMart dataset, the performance difference between the two highest-ranked models, GradientBoosting and HistGradientBoosting, was not statistically significant under either error metric after Holm correction ($p > 0.05$), indicating that both models achieved comparable predictive performance. In contrast, GradientBoosting significantly outperformed the Ridge Regression baseline ($p < 0.001$ for both metrics), demonstrating that the performance gains of the tree-based model were not attributable to random variation. The preprocessing analysis further showed that the transition from the raw data (Configuration 1) to the fully preprocessed data (Configuration 2) produced statistically significant improvements ($p < 0.01$), whereas the additional zero-variance feature removal step (Configuration 3) yielded only a small, although still statistically significant, improvement ($p < 0.05$).

For the Rossmann dataset without the *Customers* feature, RandomForest significantly outperformed ExtraTrees under both MAE- and MSE-based DM tests after Holm correction ($p < 0.05$), confirming that the observed performance advantage was statistically reliable despite the relatively small difference in predictive accuracy. RandomForest also significantly outperformed all conventional forecasting baselines, including Ridge Regression, Seasonal-Naive, and ETS (all Holm-corrected $p < 0.001$), providing statistical evidence that the superiority of the tree-based ensemble extends beyond comparisons with other machine-learning models. Similar to the BigMart results, the preprocessing pipeline produced significant improvements from Configuration 1 to Configuration 2 ($p < 0.001$), whereas the additional zero-variance feature removal step (Configuration 3) did not produce a statistically significant improvement ($p > 0.05$). In contrast, removing the seasonal and promotional variables in the transition from Configuration 3 to Configuration 4 produced a highly significant deterioration under both loss functions (Holm-adjusted $p < 0.001$), statistically confirming their substantial contribution to Rossmann forecasting performance. Finally, in the ablation experiment including the *Customers* feature, the performance difference between RandomForest and ExtraTrees remained statistically significant

after Holm correction ($p < 0.05$), indicating that RandomForest consistently retained a measurable performance advantage under both experimental scenarios.

**Table 9.** Paired Wilcoxon signed-rank tests for BigMart (Configuration 3 unless noted; $n = 1705$ held-out validation rows for all comparisons), with Holm–Bonferroni correction applied within the full family of $m = 8$ tests. Significance markers reflect the Holm-corrected $p$-value.

| Comparison | Metric | Mean Δ | $p$ (Raw) | $p$ (Holm) | Effect Size $r$ | Sig. |
|---|---|---|---|---|---|---|
| GB vs. HGB (top-2) | MAE | −2.65 | 0.431 | 0.863 | 0.022 | ns |
| | MSE | −27,411.74 | 0.569 | 0.863 | 0.016 | ns |
| GB (best tree) vs. Ridge (baseline) | MAE | −218.76 | <0.001 | <0.001 | −0.372 | *** |
| | MSE | −557,975.18 | <0.001 | <0.001 | −0.348 | *** |
| GB: C1 vs. C2 (preprocessing effect) | MAE | +38.50 | <0.001 | <0.001 | 0.115 | *** |
| | MSE | +58,350.82 | <0.001 | 0.001 | 0.102 | ** |
| GB: C2 vs. C3 (zero-variance effect) | MAE | +0.12 | 0.007 | 0.020 | −0.084 | * |
| | MSE | +503.16 | 0.003 | 0.011 | −0.094 | * |

ns: not significant ($p \geq 0.05$); * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$ (Holm-corrected).

**Table 10.** Diebold–Mariano tests for Rossmann (Configuration 3 unless noted; $T = 183$ daily loss observations for all comparisons; Newey–West HAC variance, lag = 7), with Holm–Bonferroni correction applied within the full family of $m = 16$ tests. Significance markers reflect the Holm-corrected $p$-value.

| Comparison | Loss | Mean Daily Δ | DM Stat. | $p$ (Raw) | $p$ (Holm) | Sig. |
|---|---|---|---|---|---|---|
| RF vs. ET, Without *Customers* (top-2) | MAE | −27.74 | −2.78 | 0.005 | 0.027 | * |
| | MSE | −84,527.41 | −2.64 | 0.008 | 0.027 | * |
| RF (best tree) vs. Ridge (baseline) | MAE | −1,613.89 | −45.05 | <0.001 | <0.001 | *** |
| | MSE | −10,660,679.79 | −15.90 | <0.001 | <0.001 | *** |
| RF (best tree) vs. Seasonal-Naive (baseline) | MAE | −1,037.98 | −16.09 | <0.001 | <0.001 | *** |
| | MSE | −4,825,343.87 | −9.57 | <0.001 | <0.001 | *** |
| RF (best tree) vs. ETS (baseline) | MAE | −731.69 | −12.85 | <0.001 | <0.001 | *** |
| | MSE | −3,133,328.22 | −7.98 | <0.001 | <0.001 | *** |
| RF: C1 vs. C2 (preprocessing effect) | MAE | +116.12 | 8.26 | <0.001 | <0.001 | *** |
| | MSE | +396,732.99 | 7.30 | <0.001 | <0.001 | *** |
| RF: C2 vs. C3 (zero-variance effect) | MAE | −0.26 | −0.47 | 0.640 | 1.000 | ns |
| | MSE | −1,542.04 | −0.53 | 0.595 | 1.000 | ns |
| RF: C3 vs. C4 (seasonal/promotional ablation) | MAE | −1,071.66 | −20.75 | <0.001 | <0.001 | *** |
| | MSE | −5,096,293.54 | −11.22 | <0.001 | <0.001 | *** |
| RF vs. ET, with *Customers* (ablation) | MAE | +10.94 | 3.11 | 0.002 | 0.011 | * |
| | MSE | +30,905.65 | 2.72 | 0.007 | 0.027 | * |

ns: not significant ($p \geq 0.05$); * $p < 0.05$; *** $p < 0.001$ (Holm-corrected).

Overall, the statistical analyses support the main experimental findings. The superiority of the best-performing tree-based models over the classical forecasting baselines is statistically robust, while the preprocessing analyses indicate that most of the predictive improvement arises from the primary preprocessing pipeline, with the subsequent zero-variance feature removal step having only a limited impact.

### *4.5. Sensitivity Analysis*

To evaluate whether the reported model rankings were influenced by the selected hyperparameter settings, a targeted one-factor-at-a-time sensitivity analysis was conducted on the models involved in the closest performance comparisons. Five hyperparameter settings were evaluated for GradientBoosting and HistGradientBoosting on the BigMart

dataset, and three hyperparameter settings were evaluated for RandomForest and ExtraTrees on the Rossmann dataset without the *Customers* feature. The tested hyperparameter settings modified ensemble size, learning rate (gradient-boosting models only), and model complexity, while all remaining hyperparameters, preprocessing steps, and evaluation procedures were kept identical to the main experiments. Table 11 summarizes the BigMart results and Table 12 summarizes the Rossmann results.

**Table 11.** Sensitivity analysis for BigMart, Configuration 3 (Preprocessed + Selected). Reference rows correspond to the default hyperparameter settings used throughout the main experiments.

| Model | Hyperparameter Setting | RMSE | RMSE% | MAE | $R^2$ | CV $R^2$ |
|---|---|---|---|---|---|---|
| GradientBoosting | Reference (main experiments) | 1028.20 | 47.14 | 723.10 | 0.6110 | 0.5970 |
| | n_estimators = 50 | 1028.39 | 47.15 | 724.02 | 0.6109 | 0.5970 |
| | n_estimators = 200 | 1028.20 | 47.14 | 723.10 | 0.6110 | 0.5970 |
| | learning_rate = 0.05 | 1026.87 | 47.08 | 723.37 | 0.6120 | 0.5972 |
| | learning_rate = 0.20 | 1033.23 | 47.37 | 725.74 | 0.6072 | 0.5942 |
| | max_depth = 2 (reduced complexity) | 1028.59 | 47.16 | 726.84 | 0.6107 | 0.5957 |
| HistGradientBoosting | Reference (main experiments) | 1041.45 | 47.74 | 725.75 | 0.6009 | 0.5887 |
| | max_iter = 50 | 1041.45 | 47.74 | 725.75 | 0.6009 | 0.5889 |
| | max_iter = 200 | 1041.45 | 47.74 | 725.75 | 0.6009 | 0.5887 |
| | learning_rate = 0.05 | 1032.32 | 47.33 | 730.83 | 0.6079 | 0.5918 |
| | learning_rate = 0.20 | 1041.76 | 47.76 | 725.27 | 0.6007 | 0.5808 |
| | max_leaf_nodes = 15 (reduced complexity) | 1025.54 | 47.02 | 720.30 | 0.6130 | 0.5966 |

For the BigMart dataset, varying the number of estimators and learning rate produced only minor changes in predictive performance, and GradientBoosting consistently remained the best-performing model under both the held-out and cross-validation evaluations. Only the reduced-complexity hyperparameter setting, in which the maximum tree depth (or the corresponding leaf-node constraint for HistGradientBoosting) was decreased, produced a small ranking reversal. Under this intentionally simplified setting, HistGradientBoosting achieved a slightly higher performance than GradientBoosting (held-out $R^2 = 0.613$ versus 0.611; CV $R^2 = 0.597$ versus 0.596), although the absolute difference remained very small.

A similar pattern was observed for the Rossmann dataset without the *Customers* feature. Changing the number of trees had a negligible effect on model ranking, with RandomForest consistently outperforming ExtraTrees under both evaluation protocols. When the maximum tree depth was restricted to 15, however, ExtraTrees marginally exceeded RandomForest (held-out $R^2 = 0.692$ versus 0.683; CV $R^2 = 0.683$ versus 0.668). Unlike the BigMart reversal, this ranking change was accompanied by a substantial reduction in predictive performance for both models relative to their default Configuration 3 settings, with held-out $R^2$ falling by approximately 0.18–0.20.

Overall, the sensitivity analysis indicates that the study's conclusions are robust to moderate variations in the evaluated hyperparameters, including ensemble size and learning rate. The only observed ranking reversals occurred when model complexity was intentionally restricted, but the consequence of this restriction differed markedly between datasets: on Rossmann, it substantially degraded performance for both RandomForest and ExtraTrees, whereas on BigMart it left GradientBoosting's performance essentially unchanged and modestly improved HistGradientBoosting's. These findings suggest that the reported superiority of the selected ensemble models is not an artifact of a single hyperparameter choice, while also highlighting that the sensitivity of model rankings to aggressive complexity reduction can itself vary by dataset, rather than following a single general pattern.

**Table 12.** Sensitivity analysis for Rossmann without *Customers*, Configuration 3 (Preprocessed + Selected). Reference rows correspond to the default hyperparameter settings used throughout the main experiments.

| Model | Hyperparameter Setting | RMSE | RMSE% | MAE | $R^2$ | CV $R^2$ |
|---|---|---|---|---|---|---|
| RandomForest | Reference (main experiments) | 1044.25 | 15.01 | 701.01 | 0.8838 | 0.8319 |
| | n_estimators = 50 | 1048.66 | 15.08 | 704.60 | 0.8828 | 0.8306 |
| | n_estimators = 200 | 1043.30 | 15.00 | 699.86 | 0.8840 | 0.8328 |
| | max_depth = 15 (reduced complexity) | 1725.12 | 24.80 | 1214.90 | 0.6829 | 0.6678 |
| ExtraTrees | Reference (main experiments) | 1095.94 | 15.76 | 737.62 | 0.8720 | 0.8217 |
| | n_estimators = 50 | 1099.04 | 15.80 | 739.96 | 0.8713 | 0.8208 |
| | n_estimators = 200 | 1095.50 | 15.75 | 736.97 | 0.8721 | 0.8226 |
| | max_depth = 15 (reduced complexity) | 1701.22 | 24.46 | 1170.44 | 0.6917 | 0.6834 |

### *4.6. Feature Importance and SHAP Results*

Feature importance and SHAP analyses were used jointly to interpret the best-performing model under each experimental scenario. Figures 4–6 present the conventional tree-based feature-importance rankings, whereas Figure 7 presents the corresponding SHAP summary plots and Figure 8 presents representative SHAP dependence plots for the most influential feature in each scenario. The two approaches are complementary because they quantify feature importance differently. Tree-based feature importance summarizes how much each variable contributes to the model during training, whereas mean absolute SHAP values summarize its average contribution to individual predictions. Consequently, some variation between their rankings is expected.

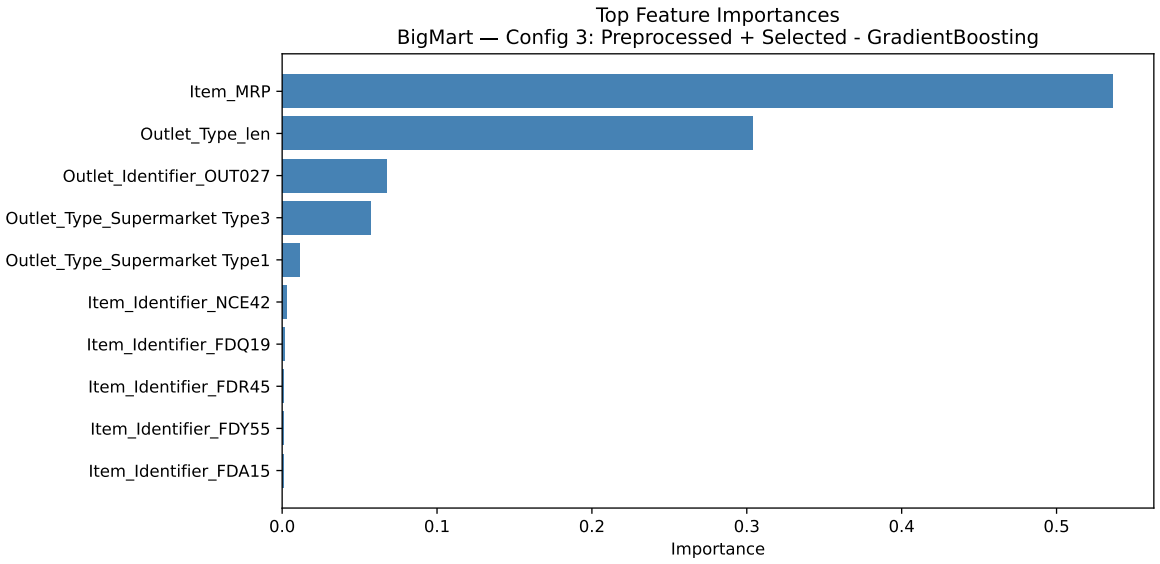



**Figure 4.** Top feature importances for the best-performing model on BigMart (GradientBoosting, C3: Preprocessed + Selected).

#### 4.6.1. BigMart

As shown in Figure 4, the GradientBoosting model exhibits a concentrated feature-importance distribution. `Item_MRP` is the dominant predictor, accounting for approximately 54% of the conventional importance, followed primarily by outlet-related variables. The SHAP ranking is largely consistent with this result. At the individual-feature level, `Item_MRP` accounts for approximately 50.6% of the total mean absolute SHAP attribution, followed by `Outlet_Type_len` (25.5%) (Figure 7a). Aggregating these contributions by variable category shows Pricing accounting for 50.6% of total attribution, Store-related

variables 47.9%, and Product-level variables only 1.5%. Thus, despite the large number of one-hot-encoded product columns, product identity contributes relatively little additional predictive information beyond that already captured by price and outlet characteristics.

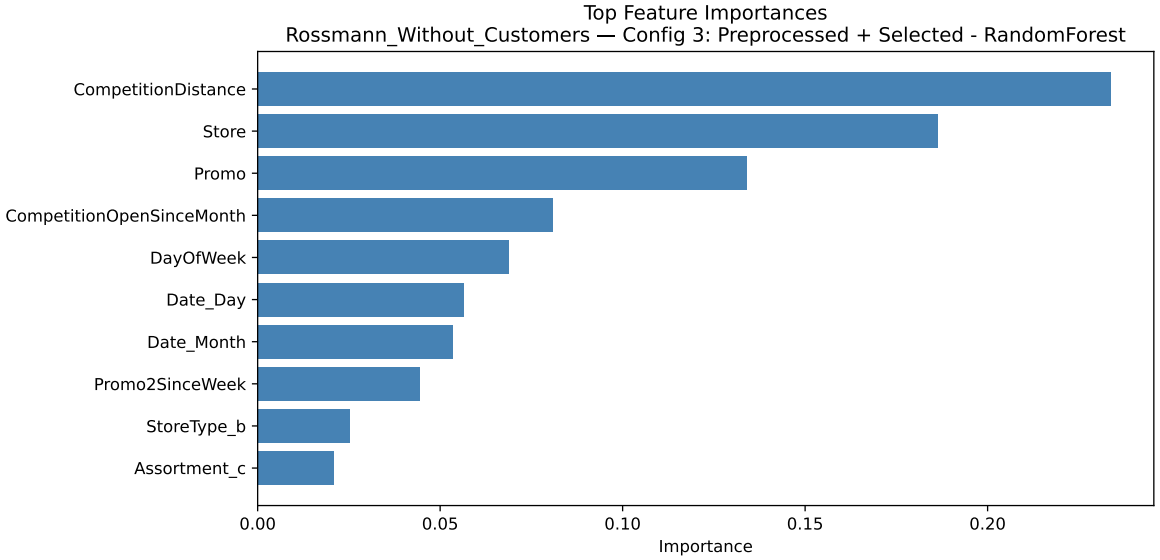



**Figure 5.** Top feature importances for the best-performing model on Rossmann without *Customers* (RandomForest, C3: Preprocessed + Selected).

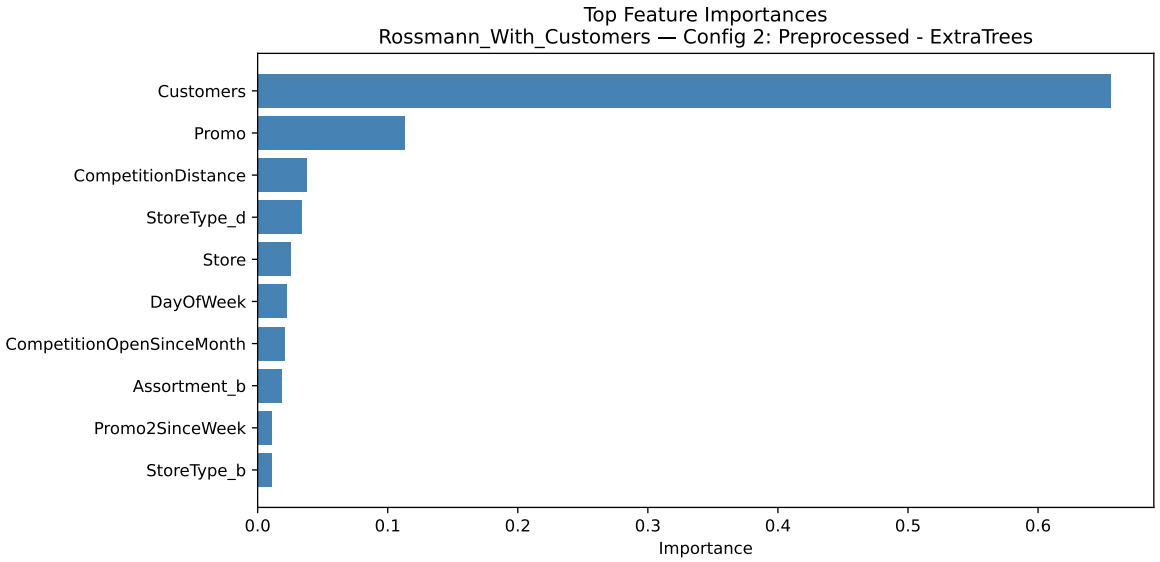
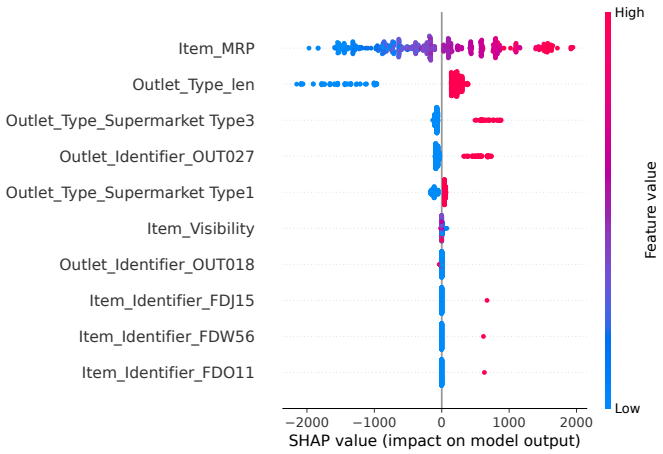


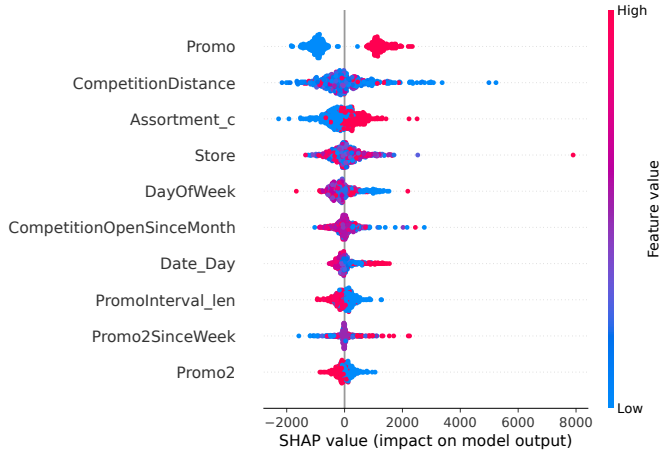

**Figure 6.** Top feature importances for the best-performing model on Rossmann with *Customers* (ExtraTrees, C3: Preprocessed + Selected).

The SHAP summary and dependence plots provide additional insight into the direction of these effects. Higher `Item_MRP` values are generally associated with positive SHAP values, whereas lower values are associated with negative contributions to the model prediction, revealing a clear and generally increasing relationship between maximum retail price and the predicted sales value (Figure 8a). This result should not be interpreted as evidence that increasing price causally increases unit demand. The target represents sales revenue, and higher-priced products may generate larger sales values even when sales volume is not observed separately. The dependence plot also suggests that the contribution of `Item_MRP` varies across outlet categories represented by the outlet-type indicator. However, because `Outlet_Type_len` is an engineered representation of the original outlet label rather

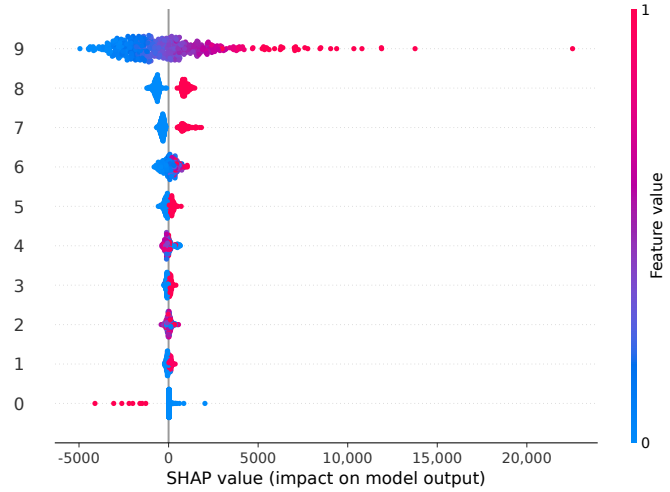
than an intrinsically ordered outlet characteristic, this pattern should be interpreted only as evidence of heterogeneity across outlet categories. Overall, the SHAP results indicate that the model relies predominantly on pricing and outlet characteristics, reflecting the comparatively limited set of predictive variables available in the BigMart dataset relative to the Rossmann dataset.




(**a**) BigMart (GradientBoosting, C3)




(**b**) Rossmann, without *Customers* (RandomForest, C3)



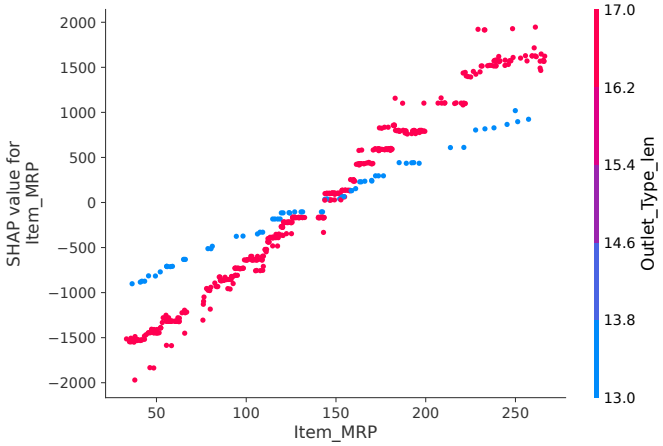

(**c**) Rossmann, with *Customers* (ExtraTrees, C3)

**Figure 7.** SHAP summary plots for the best-performing model on each dataset/scenario, all under Configuration 3 (Preprocessed + Selected). (**a**) BigMart (GradientBoosting); (**b**) Rossmann without *Customers* (RandomForest); (**c**) Rossmann with *Customers* (ExtraTrees). Point color indicates the relative value of the feature (red = high, blue = low), and horizontal position indicates the SHAP value (impact on model output) for each observation.

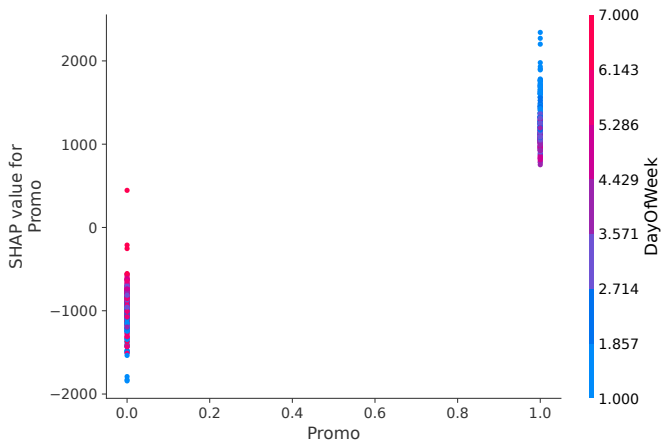
#### 4.6.2. Rossmann Without Customers

When customer traffic is excluded, the RandomForest feature-importance results in Figure 5 distribute predictive influence across a broader set of variables. `CompetitionDistance` ranks first in the conventional importance measure (approximately 0.23), followed by `Store` (approximately 0.19) and `Promo` (approximately 0.13). Temporal variables, including `CompetitionOpenSinceMonth`, `DayOfWeek`, `Date_Day`, and `Date_Month`, also contribute meaningfully. These results indicate that the model relies on promotional activity, competitive context, store heterogeneity, and calendar-related information, consistent with the seasonality

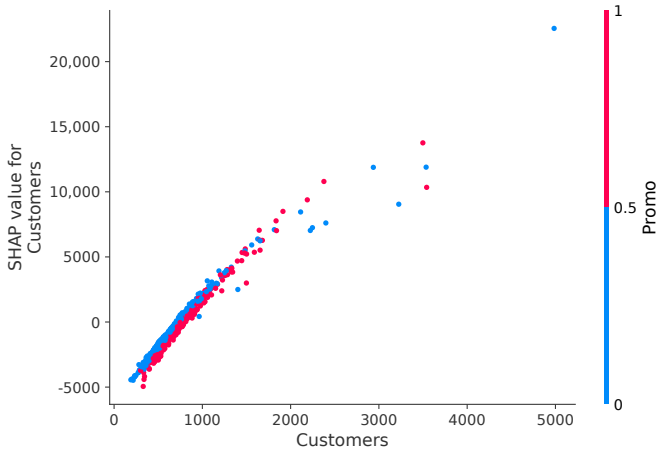
and promotion ablation results in which removing these variables substantially reduced forecasting accuracy.




(**a**) `Item_MRP` – BigMart (GradientBoosting, C3)




(**b**) `Promo` – Rossmann, without *Customers* (RandomForest, C3)




(**c**) `Customers` – Rossmann, with *Customers* (ExtraTrees, C3)

**Figure 8.** SHAP dependence plots for the most influential feature in each dataset/scenario, all under Configuration 3 (Preprocessed + Selected). (**a**) `Item_MRP` on BigMart (GradientBoosting); (**b**) `Promo` on Rossmann without *Customers* (RandomForest); (**c**) `Customers` on Rossmann with *Customers* (ExtraTrees). Each point represents one validation observation; the vertical axis shows the SHAP value (impact on model output) for the labeled feature, and point color indicates the value of a secondary interacting feature as noted in each panel.

SHAP provides a complementary ranking, identifying `Promo` as the largest average contributor to individual predictions (approximately 24.7% of total mean absolute SHAP), followed by `CompetitionDistance` (13.4%) and `Assortment_c` (9.2%) (Figure 7b). Aggregated by category, Promotional variables account for 39.7% of total SHAP attribution, Store-related variables 23.4%, Competition variables 18.7%, and Calendar/Temporal variables 18.2%. Promotional and Calendar/Temporal variables therefore jointly account for approximately 58% of total attribution, which is consistent with the substantial performance degradation observed when these variables were removed in the C3 → C4 ablation (Table 7).

The reordering of `Promo` and `CompetitionDistance` between the two methods reflects their different bases of measurement rather than a contradiction. The SHAP dependence plot for `Promo` shows a clear separation between promoted and non-promoted observations (Figure 8b): promotional periods generally increase predicted sales, while their absence contributes negatively relative to the model's baseline prediction. This strengthens the ablation

findings by showing not only that promotional variables improve predictive performance, but also that active promotions generally shift individual predictions upward. The dependence plot for `CompetitionDistance` shows a more heterogeneous and non-monotonic pattern, particularly among stores with nearby competitors, suggesting that the effect of competitive proximity varies across stores rather than following a simple monotonic relationship. These associations describe the fitted model and should not be interpreted causally.

#### 4.6.3. Rossmann with Customers

Including *Customers* substantially changes the interpretability profile (Figure 6). In the conventional ExtraTrees importance analysis, *Customers* accounts for approximately 65% of total importance, far exceeding `Promo` (approximately 0.11) and `StoreType_d` (approximately 0.03), consistent with the ablation results showing that including customer traffic markedly improved predictive accuracy.

The SHAP ranking confirms the dominant role of customer traffic, although its attribution share is lower under this distinct importance measure. At the individual-feature level, *Customers* accounts for approximately 43.8% of total mean absolute SHAP, followed by `Promo` (16.7%) and `StoreType_d` (10.9%) (Figure 7c). Aggregated by category, the Customer category accounts for 43.8% of total attribution, Promotional variables 22.6%, Store-related variables 19.3%, Competition variables 7.6%, and Calendar/Temporal variables 6.8%. The combined Promotional and Calendar/Temporal share therefore decreases from approximately 58% without *Customers* to approximately 29% when customer counts are included. This redistribution is consistent with the smaller C3 $\rightarrow$ C4 performance decline observed in the with-*Customers* scenario and suggests that customer footfall captures part of the predictive information otherwise provided by promotional and temporal variables.

The SHAP dependence plot for `Customers` shows a strong, generally increasing positive relationship: observations with larger customer counts receive increasingly positive contributions to predicted sales (Figure 8c). `Promo` remains the second-largest SHAP contributor, and the positive contributions associated with `Promo = 1` indicate that promotional activity retains predictive value even after customer traffic is included. The positive contribution associated with `StoreType_d` suggests systematic sales differences for this store format after accounting for customer traffic and the other predictors. The dependence plot further suggests that the contribution of store type may vary across customer volumes. However, because pairwise SHAP interaction values were not computed, this pattern should be interpreted as possible effect heterogeneity rather than conclusive evidence of interaction. Because `Customers` is generally unavailable at forecast time, this scenario should remain interpreted as a controlled ablation or upper-bound analysis rather than the primary operational forecasting setting.

Overall, the feature-importance and SHAP findings are broadly consistent, with SHAP additionally revealing the direction and distribution of each variable's contribution across individual predictions. The category-level SHAP results are qualitatively consistent with the feature-ablation findings, although the two analyses serve different purposes: ablation measures the change in predictive performance after features are removed and the model is reevaluated, whereas SHAP explains how the fitted full model distributes prediction contributions. As elsewhere in this section, all reported SHAP values describe predictive associations within the fitted models rather than causal relationships.

## 5. Conclusions and Future Directions

This study presented a controlled benchmarking evaluation of seven tree-based machine learning models for retail sales forecasting across two publicly available datasets: BigMart and Rossmann, with preprocessing configuration treated as the primary experi-

mental variable. On BigMart, GradientBoosting achieved the best performance ($R^2 = 0.611$), reflecting the limited feature availability in this dataset rather than an intrinsic dataset limitation, as discussed in Section 4.1. On Rossmann, ablation studies demonstrated that seasonal and promotional features are critical drivers of forecasting accuracy, with their removal causing severe degradation across all models. An ablation analysis further showed that the *Customers* variable, when available, substantially improved the best model's performance ($R^2$ from 0.884 to 0.965); however, as this variable is not typically available at forecast time in operational settings, $R^2 = 0.884$ represents the more realistic benchmark for practical deployment. Across both datasets, AdaBoost was consistently the weakest model, while gradient-based boosting methods led on BigMart and bagging-based methods dominated on Rossmann, suggesting that dataset characteristics play a significant role in determining the relative performance of ensemble strategies. From a practical perspective, the results are particularly relevant for SMEs, where computational resources and machine learning expertise may be limited. Although the best-performing model varied across datasets, tree-based ensemble methods consistently achieved strong forecasting performance while requiring relatively limited hyperparameter optimization, making them an attractive low-overhead solution for resource-constrained retail forecasting.

Despite these findings, several limitations remain. First, the evaluation is restricted to two publicly available retail datasets, which may limit the generalizability of the conclusions to other retail domains or data structures. Building on this, differences in product categories, geographic regions, customer behavior, promotional strategies, and data availability across retail environments may influence the relative effectiveness of preprocessing strategies and contextual features observed in this study. For instance, the dominant predictive role observed for the Customers variable is likely tied to Rossmann's physical, footfall-driven retail model and may be less pronounced in retail settings where customer traffic is not directly observed; similarly, the relative importance of promotional and seasonal features may vary across different retail sectors and product categories. The relative impact of preprocessing strategies may also differ for datasets with substantially different levels of missing data, noise, or feature composition than BigMart and Rossmann. Consequently, while the observed trends provide useful empirical evidence, their generalizability to other retail contexts should be validated using additional datasets with different characteristics. Nevertheless, the broader methodological framework of evaluating preprocessing strategies and contextual feature availability through controlled experiments is expected to remain applicable across a wide range of retail forecasting settings. Second, the preprocessing configurations and regex-based seasonal feature exclusion were designed specifically for the BigMart and Rossmann datasets, and may require adaptation before being applied to other retail contexts with different feature structures.

Looking ahead, several future research directions remain open. These include extending the benchmarking framework to additional retail datasets with richer and more diverse feature spaces, and exploring deep learning and hybrid forecasting architectures as complementary benchmarks. The preprocessing configurations could be expanded to include model-based imputation methods (e.g., KNN or iterative imputation), alternative categorical encoding schemes (e.g., target encoding), and automated feature engineering techniques, to assess whether these yield further performance improvements beyond the preprocessing steps evaluated in this study. Future work could also explore self-supervised and semi-supervised learning paradigms, which have demonstrated improved label efficiency in other time-series domains [35,36]. Such approaches could potentially be explored for retail sales forecasting, given the availability of large volumes of historical sales data, although their effectiveness for this application remains to be investigated.

**Author Contributions:** Conceptualization, S.A., A.A. (Amal Alazba) and A.A. (Atheer Alqahtani); methodology, S.A., A.A. (Amal Alazba) and A.A. (Atheer Alqahtani); software, S.A. and A.A. (Atheer Alqahtani); validation, S.A. and A.A. (Amal Alazba); formal analysis, S.A.; investigation, S.A., A.A. (Amal Alazba) and A.A. (Atheer Alqahtani); writing—original draft preparation, S.A. and A.A. (Atheer Alqahtani); writing—review and editing, S.A. and A.A. (Amal Alazba); supervision, S.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Ongoing Research Funding program (ORF- 2026-1313), King Saud University, Riyadh, Saudi Arabia.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data that support the findings of this study are openly available at https://www.kaggle.com/datasets/yasserh/bigmartsalesdataset (accessed on 5 June 2026) and https://www.kaggle.com/c/rossmann-store-sales (accessed on 5 June 2026).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Xactly. Sales Forecasting: What It Is and Why It's Important. 2024. Available online: https://www.xactlycorp.com/blog/forecasting/sales-forecasting-what-it-and-why-its-important (accessed on 1 May 2026).
2. Rostami-Tabar, B.; Greene, T.; Shmueli, G.; Hyndman, R.J. Responsible forecasting: identifying and typifying forecasting harms. *arXiv* **2024**, arXiv:2411.16531.
3. Badmus, O.; Rajput, S.A.; Arogundade, J.B.; Williams, M. AI-driven business analytics and decision making. *World J. Adv. Res. Rev.* **2024**, *24*, 616–633. [CrossRef]
4. Ganesan, P. AI-Powered Sales Forecasting: Transforming Accuracy and Efficiency in Predictive Analytics. *J. Artif. Intell. Mach. Learn. Data Sci.* **2024**, *2*, 1213–1216. [CrossRef]
5. Hasan, M.R. Addressing seasonality and trend detection in predictive sales forecasting: A machine learning perspective. *J. Bus. Manag. Stud.* **2024**, *6*, 100–109. [CrossRef]
6. Badorf, F.; Hoberg, K. The impact of daily weather on retail sales: An empirical study in brick-and-mortar stores. *J. Retail. Consum. Serv.* **2020**, *52*, 101921.
7. Anwar, M. The Importance of Data Preparation for Machine Learning. 2023. Available online: https://www.astera.com/type/blog/data-preparation-for-machine-learning (accessed on 1 May 2026).
8. Okeke, N.I.; Bakare, O.A.; Achumie, G.O. Forecasting financial stability in SMEs: A comprehensive analysis of strategic budgeting and revenue management. *Open Access Res. J. Multidiscip. Stud.* **2024**, *8*, 139–149. [CrossRef]
9. Grinsztajn, L.; Oyallon, E.; Varoquaux, G. Why do tree-based models still outperform deep learning on tabular data? *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 507–520. [CrossRef]
10. Dharshini, M.P.A.; Vijila, S.A. Survey of machine learning and deep learning approaches on sales forecasting. In *Proceedings of the 2021 4th International Conference on Computing and Communications Technologies (ICCCT)*; IEEE: Piscataway, NJ, USA, 2021; pp. 59–64.
11. Cheriyan, S.; Ibrahim, S.; Mohanan, S.; Treesa, S. Intelligent sales prediction using machine learning techniques. In *Proceedings of the 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*; IEEE: Piscataway, NJ, USA, 2018; pp. 53–58.
12. Krishna, A.; Akhilesh, V.; Aich, A.; Hegde, C. Sales-forecasting of retail stores using machine learning techniques. In *Proceedings of the 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*; IEEE: Piscataway, NJ, USA, 2018; pp. 160–166.
13. Swami, D.; Shah, A.D.; Ray, S.K. Predicting future sales of retail products using machine learning. *arXiv* **2020**, arXiv:2008.07779.
14. Bajaj, P.; Ray, R.; Shedge, S.; Vidhate, S.; Shardoor, N. Sales prediction using machine learning algorithms. *Int. Res. J. Eng. Technol.* **2020**, *7*, 3619–3625.
15. Ranjitha, P.; Spandana, M. Predictive analysis for big mart sales using machine learning algorithms. In *Proceedings of the 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*; IEEE: Piscataway, NJ, USA, 2021; pp. 1416–1421.
16. Huo, Z. Sales prediction based on machine learning. In *Proceedings of the 2021 2nd International Conference on E-Commerce and Internet Technology (ECIT)*; IEEE: Piscataway, NJ, USA, 2021; pp. 410–415.
17. Raizada, S.; Saini, J.R. Comparative analysis of supervised machine learning techniques for sales forecasting. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 102–110. [CrossRef]
18. Martins, E.; Galegale, N.V. Sales forecasting using machine learning algorithms. *Rev. Gest. Secr.* **2023**, *14*, 11294–11308. [CrossRef]

19. Sim, D.Y.Y.; Wei, Z. XGBoost Regression Algorithms for Efficient Predictions on Inventory Sales and Management. In *Proceedings of the 2023 6th International Conference on Electronics and Electrical Engineering Technology (EEET)*; IEEE: Piscataway, NJ, USA, 2023; pp. 66–71.
20. Hobor, L.; Brčić, M.; Polutnik, L.; Kapetanović, A. Comparative analysis of modern machine learning models for retail sales forecasting. *Croat. Oper. Res. Rev.* **2026**, *17*, 285–296. [CrossRef]
21. Bijalwan, O.; Arora, A.; Chauhan, R.; Bhatt, C.; Porkhariya, H. Impact of Machine Learning on Sales Forecasting Accuracy. In *Proceedings of the 2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*; IEEE: Piscataway, NJ, USA, 2023; pp. 472–476.
22. Jewel, R.M.; Linkon, A.A.; Shaima, M.; Sarker, M.S.U.; Shahid, R.; Nabi, N.; Hossain, M.J. Comparative Analysis of Machine Learning Models for Accurate Retail Sales Demand Forecasting. *J. Comput. Sci. Technol. Stud.* **2024**, *6*, 204–210. [CrossRef]
23. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
24. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 785–794.
25. Seyedan, M.; Mafakheri, F.; Wang, C. Cluster-based demand forecasting using Bayesian model averaging: An ensemble learning approach. *Decis. Anal. J.* **2022**, *3*, 100033. [CrossRef]
26. Nasseri, M.; Falatouri, T.; Brandtner, P.; Darbanian, F. Applying Machine Learning in Retail Demand Prediction—A Comparison of Tree-Based Ensembles and Long Short-Term Memory-Based Deep Learning. *Appl. Sci.* **2023**, *13*, 11112. [CrossRef]
27. Mishra, L.N.; Senapati, B.; Nayak, S.K.; Rangari, A.J. Explainable Retail Demand Forecasting: A Hybrid LightGBM-SHAP Framework for Inventory Optimization. *IEEE Access* **2026**, *14*, 43613–43640. [CrossRef]
28. Gerg, I.D.; Tashie, A.M.; Patanaik, A.; Koester, E.; Gupta, H.; Farnham, D.J. The substantial role of weather data in consumer spending prediction: A robust machine learning assessment. *J. Retail. Consum. Serv.* **2026**, *90*, 104649. [CrossRef]
29. Soleimani, A.; Amin, S.H.; Baki, F.; Shah, B. Forecasting weekly sales using lag-based feature engineering and a bagged tree ensemble. *Mach. Learn. Appl.* **2026**, *24*, 100922. [CrossRef]
30. Chang, O.; Naranjo, I.; Guerron, C.; Criollo, D.; Guerron, J.; Mosquera, G. A Deep Learning Algorithm to Forecast Sales of Pharmaceutical Products. 2017. Available online: https://www.academia.edu/35010555/A_Deep_Learning_Algorithm_to_Forecast_Sales_of_Pharmaceutical_Products (accessed on 5 June 2026).
31. Neelakandan, S.; Prakash, V.; PranavKumar, M.S.; Balasubramaniam, R. Forecasting of E-commerce system for sale prediction using deep learning modified neural networks. In *Proceedings of the 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5.
32. Luyo Ballena, J.P.; Ortiz Pallihuanca, C.P.; Carrera Salas, E.A. A Predictive Sales System Based on Deep Learning. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 179–186. [CrossRef]
33. Kaggle. BigMart Sales Dataset, Uploaded by Yasserh. 2022. Available online: https://www.kaggle.com/datasets/yasserh/bigmartsalesdataset (accessed on 5 June 2026).
34. Kaggle. Rossmann Store Sales. 2015. Available online: https://www.kaggle.com/c/rossmann-store-sales (accessed on 5 June 2026).
35. Wang, S. Temporal–Contextual Self-Supervised Time-Series Learning for Automated Fault Detection and Diagnosis of Air Handling Units in Buildings. *Build. Environ.* **2026**, *292*, 114300. [CrossRef]
36. Wang, S. Class-aware temporal and contextual contrastive framework for semi-supervised automated fault detection and diagnosis in air handling units. *Energy Build.* **2026**, *358*, 117233. [CrossRef]