e-ISSN: 2686-2573
DOI: 10.21512/emacsjournal.v8i2.16162

The Impact of Ensemble Stacking with Machine Learning and Statistical Models on Improving Sales Forecasting Accuracy

Prima Yohana[1], Alexander Agung Santoso Gunawan[2*], Heri Ngarianto[3]
[1]Computer Science Department, BINUS Graduate Program - Master of Computer Science,
[2,3]Computer Science Department, School of Computer Science,
Bina Nusantara University
Jakarta, Indonesia 11480
prima.yohana@binus.ac.id; aagung@binus.edu;
heri.ngarianto@binus.ac.id

*Correspondence: aagung@binus.edu

Abstract – *Sales forecasting in retail remains a challenge, as changing customer behavior and seasonal patterns undermine predictive accuracy. For example, during promotional periods, managers tend to respond to changes in the weather or abrupt swings in preferences, not anticipating. For this purpose, we propose an ensemble stacking framework for machine learning and statistical models to improve forecasts in dynamic demand. We used a meta-learner of RidgeCV, trained on past retail sales data, to ensemble base learners including Random Forest, Extra Trees, XGBoost, and Multiple Linear Regression. Tree-based models captured non-linear interactions, but linear regression preserved interpretability of trends. The meta-learner learned to assign adaptive weights based on recent error patterns in time windows. We assessed performance using RMSE and MAE with particular focus on high volatility periods (e.g., holiday spikes and post-promotion troughs). The stacking model achieved an RMSE of 0.7689 and an MAE of 0.3066, outperforming individual base models consistently. Stacked predictions also showed less week-to-week variance and faster recovery after sudden shifts in demand. These results show that stacking can capture heterogeneous demand dynamics better than single models and, hence, provide a more robust structure. In practice, it allows for more confident safety-stock decisions and timely replenishment adjustments. Retailers can run a number of models in parallel and continuously rebalance them with new data as it comes in. Future work may include online updates of meta-learners and external covariates such as local events or competitor pricing.*

Keywords: *Ensemble stacking; Sales forecasting; Machine learning; Statistical models; Random Forest; Extra Trees; XGBoost; Multiple Linear Regression; RidgeCV*

I. INTRODUCTION

Sales forecasting is one of the most basic, fundamental foundations upon which retail operations live and breathe, directly impacting inventory planning, promotion spending, replenishment decisions and service-level optimization. In practice, accurate forecasting remains challenging in real-world retail settings, because demand is still non-stationary at the item level. Two important factors contributing to this non-stationarity are seasonality, as evidenced by recurring monthly or annual demand cycles driven by holidays and promotions, and user drift, as evidenced by changes in consumer tastes, product mix and market context over time. Prior work in e-commerce settings has identified nonlinear demand spikes around large-scale events (e.g., Black Friday, Christmas, or Prime Day) and a change in the underlying feature–target relation due to changing consumer behavior, which reduces model stability unless the forecasting systems are regularly updated (M. Wang et al., 2024; Yao, 2023).

Current forecasting models capture different aspects of this problem, but none is generally dominant across all retail regimes. XGBoost beats strong baselines such as GBDT, SVM, Bayesian models, and linear regression on tabular e-commerce data containing many product attributes, especially when the target structure is redefined to be less noise-sensitive (M. Wang et al., 2024). Other studies have shown that calendar and holiday variables are among the most influential predictors in retail forecasting, especially for tree-based learners, and removing such variables can substantially reduce predictive accuracy (Yao, 2023). These results confirm that seasonal structure needs to be modeled explicitly and that time-aware validation is needed for realistic performance assessment.

Other retail-adjacent forecasting domains also report similar challenges. For example, monthly vehicle sales forecasts exhibit huge and unstable fluctuations between cities. Researchers have proposed rolling refitting, monitoring residuals, and adding exogenous variables to remain robust under changing demand conditions (Li, 2025). Evidence that simple seasonal assumptions may be insufficient comes from comparisons of ARIMA with more flexible machine learning algorithms in promotion driven forecasting settings. Demand patterns are shaped by nonlinear effects, segmentation and changing market dynamics. Together these studies suggest that seasonality and drift need to be considered together, rather than independently, to be adequately addressed.

An Ensemble Learning Approach for Diabetes Prediction using Logistic Regression, KNN, and SVM as Base Learners and Logistic Regression as Meta-Learner. The results show that the stacking performs better than independent individual models in terms of classification performance. The study highlighted that distribution shifts in medical data can easily occur due to demographic or measurement changes, which is very much related to user drift about forecasting tasks. To maintain consistency, the authors suggest regularly checking and observing the feature distributions. However, the data referred is not time series data, whereas the proposition of useful validation based on time highlights the urgency to deal with regional evolution. This information indicates the robustness of stacking ensembles in dynamic environment (Attipoe et al., 2025).

We developed a stacking ensemble model for cardiovascular disease prediction, with base learners consisting of Random Forest, SVM, and CatBoost, and meta learners comprising Logistic Regression and Multi-Layer Perceptron. The results showed that the Stacking-MLP configuration was superior to the individual models. The study itself admits the problem of combining datasets from different institutions (i.e. domainshift) when correlated to user drift in forecasting. To address this concern, they suggested retraining models periodically and evaluating their performance across different populations. The analysis did not take seasonality into account, but drift monitoring is a very useful concept in forecasting contexts where data changes over time. This demonstrates the flexibility of stacking ensembles to different environments (Bihri et al., 2025).

Another study for diabetes prediction used stacking ensemble with Logistic Regression, LDA, KNN, Naive Bayes and SVM as base models and Random Forests, AdaBoost and Extra Trees as meta-learners. The ensemble was 95.29% accurate, significantly better than the performance of any single classifier. The authors noted that drift occurs in medical datasets due to changing patient populations and diagnostic habits. For this, they proposed external validation and monitoring of distribution shifts over time. While they did not directly analyze seasonality, their approach of tracking periodically is consistent with standard practices in forecasting for handling seasonal factors. The performance of the stacked model using this framework shows how the stacking models perform against changing data environments (Daza et al., 2024).

A new stacking regression method was proposed for the estimation of the biodiesel conversion efficiency with the base learners: Random Forest, XGBoost, Deep Neural Networks and the meta-learner: Linear Regression. The study added GAN-based data augmentation to enhance performance, which increased the ensemble's R 2 from 0.45 to 0.81. The authors highlighted that process drift could occur when reaction parameters change, which need to be periodically monitored and re-fitted. They suggested that temporal evaluation is useful to detect gradual degradation, though seasonality was not directly evaluated. These recommendations are similar in sales forecasting contexts where distributional changes and seasonal cycles are key factors. The study shows how ensemble stacking can adapt to dynamic operational conditions (Karaoğlu & Söyler, 2025).

Tree-structured Parzen estimators Optimized tree-based regressors for predicting blast-induced fragmentation size in mining operations. In our model testing the best performing method was Extra Trees against Random Forest and Gradient Boosted methods. The research underlined the challenge of user drift on predictive tasks, since the environmental and operational factors could shift the underlying data distribution. Seasonal effects in blasting results were indirectly influenced by the variety of weather too. To overcome these challenges, the authors have proposed rolling re-training, cross-site validation and ongoing tracking of errors. Such strategies are equally applicable to sales forecasting models that need to be resilient not only about seasonal variation but also changing user behaviors (Mame et al., 2025).

Recent research has suggested that ensemble and tree-based models can be effective for forecasting in environments with dynamic demand shaped by seasonal signals and changes in behaviour. An ensemble by stacking for short-term electricity load forecasting always yielded better MAE, RMSE and R^2 measures than the best single models. It emphasized daily and weekly seasonal cycles; it also states that Demand drift due to operational or policy changes can reduce the reliability of the model after a period. In pursuance of this, rolling retraining,

residual monitoring and adaptive feature selection are proposed making the approach highly relevant to sales forecasting contexts in which seasonality and evolving user dynamics matter to a similar degree (N. Wang & Li, 2022).

Random Forest has been recognized as a good and reliable model for retail forecasting in terms of dealing with complex, multi-variate sales data. Sales Forecasting Research and Walmart Sales Forecasting Research have shown the value of capturing nonlinear relations between holidays, macroeconomic variables, and other drivers of demand in a model but still providing feature importance for decision support (Googerdchi & Jafari, 2025). Study of Adidas retail stores and coffee-shop sales also shows similar results where Random Forest used transactional, profitability, timing, location related features as well as weekdays to nicely predict the outcome (N et al., 2025; Windrasari et al., 2025). All these studies indicate that generalization and deployment in a retail setting could be further improved through broader features, stronger hyperparameter tuning, and adding exogenous factors.

In summary, the literature indicates that there is no universal model that consistently performs better than others across high seasonality and user evolving regimes. This only motivates us to utilize an ensemble stacking methodology which brings together complementary inductive biases from tree-based models with XGBoost, Random Forest and Extra Trees alongside an interpretable statistical baseline of Multiple Linear Regression. All models were trained and evaluated using chronologically walk-forward validation to provide a realistic assessment environment that avoids look-ahead bias. The ensemble framework that we propose, will thus leverage the flexibility of nonlinear learners with the interpretability of a statistical baseline to offer a more stable and trustworthy forecasting mechanism for fast changing demand patterns in dynamic retail scenarios. This is consistent with the work of (Al-Bazi et al., 2025) that showed showing good results forecasting for footwear sales using methods like XGBoost however noting that, as seasonality and concept drift come into play, model accuracy fades over time. Since RMSE captures a balance between large error sensitivity, and MAE is robust against outliers while remaining highly interpretable for business decision-making, we performed an evaluation of model performance over our test set using both metrics. Both of this metrics are well known in retail and e-commerce forecasting articles.

In this regard, this study proposes a deployment-oriented stacking framework for retail sales forecasting and evaluates it using leakage-safe chronological validation. The study makes four main contributions. First, it develops a strict rolling-origin evaluation protocol for one-month-ahead forecasting, ensuring that all encoders and target-based statistics are fitted only within each training fold. Second, it presents a hybrid stacking architecture with interpretability which consists of MLR, Random Forest, Extra Trees and XGBoost and RidgeCV as meta-learner Third applies a feature design which is patterned on lag variables, rolling statistics, calendar encodings and hierarchical means that can represent retail demand under time-varying conditions. Fourth, it uses the RMSE and MAE metrics, which are common in many retail forecasting applications, to evaluate both prediction accuracy and operational robustness over time. The study aims to overcome this gap in the forecasting literature, by designing a practical yet robust framework for retail environments experiencing seasonal and changing consumer behavioral patterns.

II. METHODS

In this methodology section, we describe a time-aware forecasting pipeline that closely follows the one in Figure 1. After the problem definition and data collection, exploratory data analysis (EDA) is done, which helps with data cleaning and feature engineering. Preprocessing prepares data formats , missing values and outliers , consistent keys for the integration of reporting systems . Then feature engineering generates lag variables, rolling statistics and calendar or holiday indicators only using knowledge of time t-1 to avoid overfitting. We modeled the output with three machine learning algorithms: XGBoost, Random Forest and Extra Trees, and a statistical baseline of Multiple Linear Regression (Adewumi, 2026; Hasan, 2024; Manzanas, 2020). These out-of-fold predictions are then used as meta-features within a stacking framework. The RMSE and MAE on forecasting performance are assessed over rolling-origin split, for a range of time windows and business segments while chronologically validating the model well. In addition, error diagnostics and feature importance analysis help to inform the retraining and performance monitoring. This is a standard pipeline, consistent with earlier work that emphasizes the role of seasonal indicators and drift monitoring in forecasting tasks.

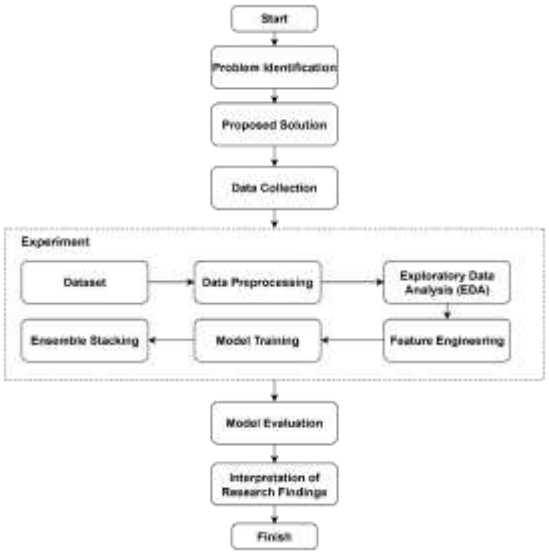

Figure 1. Framework of Research Methodology

2.1 Data Collection

The data used in this study includes transactional data of one of the very large retail platforms recorded over a period of multiple years including products, categories and store locations. It stores basic transactional information with a short information about corresponding product, its category and selling shop. Thus, it provides valuable dataset to study sales pattern across time items of interest and locations. The raw records were converted to a clean and leak free panel for proper modeling and reproducible evaluation. The workflow included formatting standardization, parsing of temporal fields, de-duplication, and the joining of supporting reference tables to add product and store context to the transactions. Sales were aggregated to a common time resolution, and a complete grid of entity-time combinations was constructed that also includes periods of no activity. Other historical features were purely based on past observations, including lagged behavior, rolling trends, dynamic pricing indicators and seasonal or calendar signals. The final dataset was then temporally split into training, validation, and test subsets with strict leakage controls to ensure reliable model evaluation.

The dataset is a continuous series of the retail sales history from early 2013 through late 2015, more than two and a half years. It consists mainly of sales and price data at the level of individual transactions, product and category identifiers, and store-level data that provides geographical or operational context. These components are then incorporated as a monthly shop–item panel preserving the active and inactive periods, so that you can model seasonality, cross-store variation and product level dynamics under a unified forecasting framework.

Table 1. Item Category Dataset

Index	No	Item Category Name	Item Category Id
0	1	PC – Headsets /	0
1	2	Accessories – PS2	1
2	3	Accessories – PS3	2
3	4	Accessories – PS4	3
4	5	Accessories – PSP	4
…	…	…	…
79	80	System Tools	79
80	81	Utilities – Tickets	80
81	82	Net carriers (spire)	81
82	83	Net carriers (piece)	82
83	84	batteries	83

The category table (Table 1) includes 84 product categories, identified by item_category_id (0-83) and described by item_category_name. The Index and No fields are for display only and are not used. It connects to the items table on item_category_id (many-to-one) to allow aggregation by category per month, and features like mean sales per category, target encoding or one-hot encoding. Normalize category names for consistency, as this table is the key to capture seasonality and heterogeneity across product groups to improve forecasting accuracy.

Table 2. Item Dataset

Index	No	Item Name	Item Id	Item Category Id
0	1	! POWER IN	0	40
1	2	! ABBYY	1	76
2	3	*** In the glory	2	40
…	…	…	…	…
22167	22168	Язык запросов	22167	49
22168	22169	Яйцо для Little	22168	62
22169	22170	Яйцо дракона	22169	69

Table 2: The items table contains all products (one row per item). item_id is the primary key to join with sales (e.g. sales_train). item_category_id points to the category dictionary (many-to-one). item_name is free text to be normalized. Index/No are for display only. There are 22,170 items with sparse per-item histories. They extract compact features such as item frequency, item age (since first sale), historical aggregates (monthly means/rolling sums) and simple text cues (name length, brand keywords). Ensure referential integrity (all item_category_id exist in category table) and check for duplicate names across different item_ids. This table links product identity, categories and sales history for accurate shop-item forecasting.

Table 3. Shop Dataset

Index	No	Shop Name	Shop Id
0	1	! Yakutsk Ordzhonikidze, 56 Franc	0
1	2	! Yakutsk TC "Central" Franc	1
2	3	Adygea TC "Mega"	2
3	4	Balashikha TRC "October-Kinomir"	3
4	5	Volzhsky mall "Volga Mall"	4
0	…	…	…
55	56	Digital storage 1C-line	55
56	57	Czechs SEC "Carnival"	56
57	58	Yakutsk Ordzhonikidze, 56	57
58	59	Yakutsk TC "Central"	58
59	60	Yaroslavl shopping center "Altair"	59

The shops table (Table 3) is the store's dictionary and location reference for all transactions: shop_id is unique identifier, shop_name is descriptive label, often includes city/address/mall, Index/No are display only, corpus consists of 60 stores Clean: normalize case, collapse extra spaces, harmonize city/mall spellings, merge duplicate labels that refer to the same outlet (e.g. Yakutsk Ordzhonikidze, 56 Franc and Yakutsk Ordzhonikidze, 56). Enforce referential integrity so all sales shop_id exists here. Extract features from shop_name such as city, mall type, region and activity flags (e.g. long inactivity) to capture location-driven demand heterogeneity and improve shop-item forecasting.

Table 4. Sales Dataset

Index	No	Date	Date Block Num	Shop Id	Item Id	Item Price	Item Count Day
0	1	02.01.2013	0	59	22154	999	1
1	2	03.01.2013	0	25	2552	899	1
2	3	05.01.2013	0	25	2552	899	-1
3	4	06.01.2013	0	25	2554	1709.05	1
4	5	15.01.2013	0	25	2555	1099	1
0	…	…	…	…	…	…	…
2935844	2935845	10.10.2015	33	25	7409	299	1
2935845	2935846	09.10.2015	33	25	7460	299	1
2935846	2935847	14.10.2015	33	25	7459	349	1
2935847	2935848	22.10.2015	33	25	7440	299	1
2935848	2935849	03.10.2015	33	25	7460	299	1

The sales table (Table 4) contains daily shop-item transactions: Date is dd.mm.yyyy is mapped to Date Block Num (0-33 for Jan-2013 to Oct-2015), Shop Id and Item Id refer to the shops and items dictionaries respectively, Item Price is a fractional unit price, Item Count Day is the units sold (negative values mean returns), Index/No are display-only. In practice, we clean unreasonable prices, align key data types, handle missing values and decide how to treat returns. Then we aggregate daily rows to monthly totals per shop-item, so they align with Date Block Num. These totals are then used to build the monthly target item_cnt_month which is augmented with lag and seasonal features, making this table the primary source for building and evaluating store-product forecasting models.

2.2 Data Preprocessing

We start with deterministic, rule-based cleaning to ensure schema consistency, integrity and traceability before any aggregation or feature work. Key columns are standardized across tables (eg category id and names are harmonized) and common date strings are parsed into iso format (YYYY-MM-DD) and then mapped to an incremental month index (date_block_num). Key data types are converted to canonical integers, whitespace is trimmed and categorical spellings are normalized to prevent join and filter errors. We eliminate exact duplicate rows and out of domain values such as nonpositive item_price entries and corrupt quantities. As specified in the study objective, returns with negative item_cnt_day are explicitly handled to avoid biased monthly totals. Dropped rows with missing values in key fields (date, date_block_num, shop_id, item_id, item_price, item_cnt_day) Referential integrity to shops, items and categories is enforced.

The cleaned data is then aggregated to a monthly panel per shop-item by grouping on shop_id, item_id and date_block_num. For each group we calculate monthly unit total as item_cnt_month, means of valid transaction prices as avg_item_price, and count of valid sales events as transactions. We also derive year and month features. We then construct a complete grid of shop–item–month for months 0–33, so that months without any activity show zero sales. This allows for

consistent lag and rolling computations without missing time indices. The grid-based panel supports leakage-free chronological splits for training, validation and testing and aligns the modeling domain with the evaluation and deployment setting.

2.3 Exploratory Data Analysis (EDA)

EDA audits schema, types, missingness, duplicates and outliers in item_price and item_cnt_day to give a clear picture of structure, quality and signal before modeling. We validate key joints and referential integrity across shops, items and categories, and profile basic statistics and correlations to surface inconsistencies and unexpected patterns. As this is a time-series task, EDA is time-aware: we plot monthly shop-item trajectories, seasonal decompositions and trend lines only within the training window to avoid look-ahead leakage.

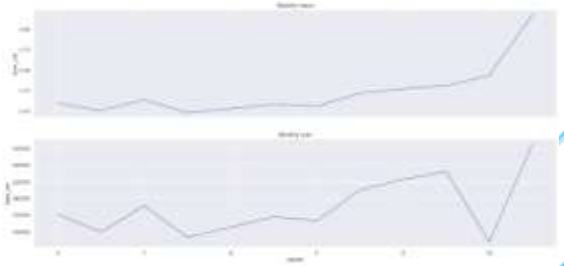

Figure 2. Sales Behaves Along the Year

Figure 2 shows the dynamics of monthly average and total sales for the observed period. As shown in the top panel, the monthly mean sales are relatively stable in the early months but steadily increase toward the end of the year. The bottom panel shows the total of monthly sales. We observe fluctuations in the first six months and an increasing trend with a sharp peak at the last month. These patterns point to both overall growth and seasonal variation in sales performance.

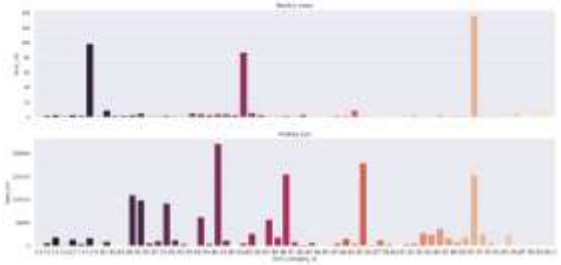

Figure 3. Sales by Category

Figure 3 presents the average and total sales per product category for each month. As we can see in the upper panel, some categories are noisier than others in terms of monthly mean sales per category as they have way larger average sale sums. The lower panel again shows total monthly sales by category, indicating a small number of categories account for a large proportion of overall volume. Concentration of demand in well-defined product categories is very important for forecasting and stock management.

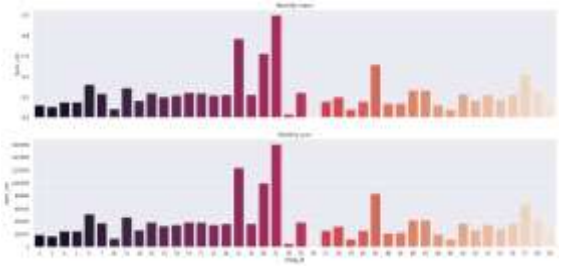

Figure 4. Sales by Shop

Figure 4 shows how sales are distributed across the different shops, both in terms of monthly averages and totals. The upper panel shows the monthly mean sales per shop, and we see that some stores have much higher averages. This shows the variation in store performance and demand concentration. The bottom panel shows the monthly total sales per shop which reveals that only a few shops are contributing excessively to the total sales volume. These findings underscore the heterogeneity across store locations and the importance of accounting for cross-store variation in forecasting models.

We investigate target and feature distributions (skewness, zero inflation), autocorrelation (ACF/PACF), and promotion/holiday effects. We compare segments by store, category, and simple brand cues from item_name to capture heterogeneity. Finally, we screen for concept/user drift by tracking shifting means, variances, category shares and feature importance over time. We distilll the findings into a data dictionary, cleaning rules, candidate features (lags, rollings, price dynamics, seasonal indicators) and a validation plan (windows and horizons).

2.4 Feature Engineering

In this work, we apply feature engineering to convert raw shop-item transaction data into useful, leakage-safe signals that reflect demand dynamics under seasonality and user drift. The design is motivated by two principles: strict chronology, where any feature for month tis constructed using only information up to t-1, and robustness to sparsity, where early or missing historical observations are not imputed using future information but treated conservatively.

We build four complementary sets of features: temporal autoregressive signals, such as lag features and rolling summaries, to capture momentum and recurring patterns; price and activity descriptors to reflect elasticity and promotional intensity; categorical and hierarchical encodings that provide stable level effects for shops, items and categories,

and calendar-based representations using year, month and cyclical transforms with selective interactions. We use a monthly panel made by aggregation. We generate a complete grid of shop–item–month instances. This way, the time indices will be consistent for lagged calculations. This feature matrix offers different inductive bias for statistical, tree-based and deep sequence models, while maintaining interpretability for diagnostics and monitoring, in line with previous findings on the important role of seasonality and user drift in retail forecasting. The engineered features are classified into four major categories. The following subsections will briefly describe each group, using concise formulas and examples.

2.4.1 Lag Feature

Lag features in Equation (1), represent the sales history of each shop–item combination over previous months, enabling the model to learn temporal dependencies and short-term momentum. For each observation at time t, lag features are defined as

$$\text{Lag}_{t-k} = y_{t-k}, k \in \{1,2,3\} \quad (1)$$

where y_(t-k)denotes the sales volume observed kmonths before the current period t.

2.4.2 Rolling Statistical Feature

Rolling mean and rolling standard deviation in Equation (2) and (3), are computed to summarize the average level and volatility of sales over previous windows, thereby capturing both seasonality and variability. The rolling mean is defined as

$$\text{RollingMean}_t^{(k)} = \frac{1}{k}\sum_{i=1}^{k} y_{t-i} \quad (2)$$

and the rolling standard deviation is defined as

$$\text{RollingStd}_t^{(k)} = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left(y_{t-i} - \text{RollingMean}_t^{(k)}\right)^2} \quad (3)$$

Rolling windows of 3, 6, and 12 months are used to represent short-, medium-, and long-term seasonal cycles.

2.4.3 Price Dynamics

To capture price elasticity, price-based variables are included, such as average monthly price, lagged price, and relative price change. The latter is formulated in Equation (4), as

$$\text{PriceChange}_t = \frac{p_{t-1} - p_{t-2}}{p_{t-2}} \quad (4)$$

where p_t is the average price in month t. These features capture the impact of price changes on demand over time. Additional indicators (price increase, price decrease) are also included to model the sensitivity to promotional and discount events.

2.4.4 Hierarchical Mean Encoding

Mean encoding creates hierarchical priors from the aggregation of sales data at the shop, item and shop-item level. For example, the mean at the shop level is given in Equation (5), as

$$\text{ShopMean}_{s,t-1} = \frac{1}{N_s}\sum_{i=1}^{N_s} y_{s,i,t-1} \quad (5)$$

where N_s is the number of items in shop s, and y_(s,i,t-1) is the sales of item i in shop s up to period t-1. This representation makes learning more stable for products with sparse transaction histories (cold-start items included), with features like item_mean and shop_item_mean providing sales priors at different levels of aggregation.

The feature matrix contains 25 numerical predictors derived from the above transformations. These include lagged sales (Lag-1 to Lag-3), rolling mean and std for 3, 6 and 12 months windows, price dynamics (mean price, lagged price, and price increase/decrease indicators), calendar encodings (year, month and cyclical sine–cosine representations), and hierarchical mean encodings (shop, item, shop–item, monthly and yearly means). Additional context variables like shop and item IDs are only used as grouping references for hierarchical aggregation and are dropped from model training to avoid data leakage. The latter composition is consistent with the 25 active numeric features reported in the full experimental documentation.

All encoders and target-mean statistics are fit only within each training fold. Validation and test months are transformed by encoders fitted on the corresponding training months, thus avoiding leakage. More generally, all features are generated under strict chronological folds, so that each predictor only refers to historical data until period t-1. This design maintains the temporal integrity, avoids forward-looking bias and sustains the reliability of the model at validation and inference stages.

2.5 Model Training

The goal of the model training phase is to create, optimize and test predictive models that can capture the time dynamics and seasonality of retail sales data After feature engineering, the resulting monthly feature matrix is the main input for families of statistical and standard models for machine learning. The objective of this stage is twofold, to predict with high accuracy and to observe the reaction of different algorithms to different seasons and behavioral conditions. The study implements a time-sensitive training paradigm and a rigorous validation scheme that allows the generation of leak-free, interpretable and generalizable to future time periods ensemble models.

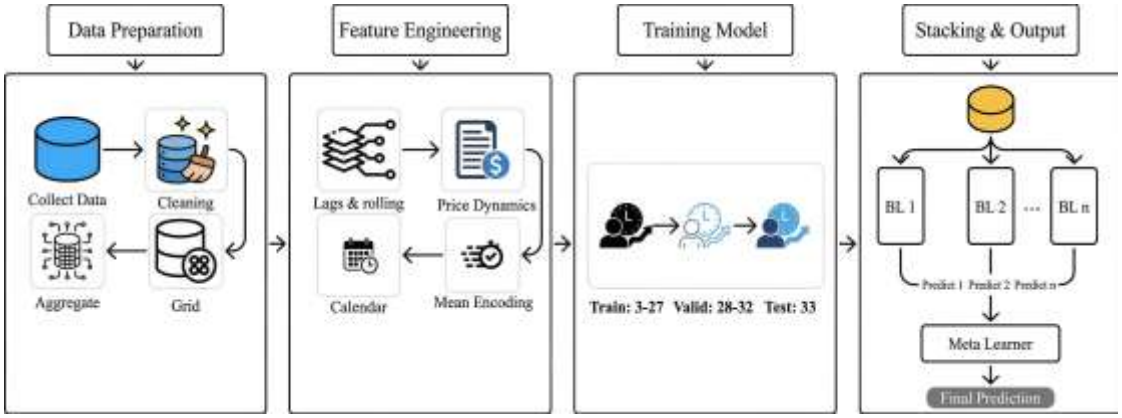

Figure 5. Overall Methodology Pipeline

The overall methodology pipeline is presented in Figure 5. It summarizes the full experimental flow, from raw data collection and cleaning, to feature transformation into leakage-safe predictors, to the stacking ensemble stage. All blocks are associated with a sequential method detailed out in methodology section to allow transparency and reproducibility of experimental setup.

In this study, RidgeCV was used as the meta-learner, which was chosen due to its ability to balance interpretability, regularization, and stability when combining heterogeneous base learners. We evaluated the performance of several meta-learners (OLS, Lasso, RidgeCV, XGBRegressor) on the same out-of-fold inputs. RidgeCV yielded the most stable weights under collinearity of base predictions and seasonal regime shifts, and a better RMSE/MAE trade-off with less variability across validation months than OLS/Lasso, and avoiding the overfitting tendencies observed for non-linear meta-learners. RidgeCV is a simple Linear Regression meta-model but with L2 regularization to penalize large coefficients, reducing variance and overfitting due to correlated base predictions. This approach makes the ensemble weights robust against temporal drift and changing sales patterns. The use of RidgeCV is aligned with the objective of the ensemble to increase robustness and interpretability in a dynamic retail forecasting setting as reported in the full experimental results.

2.5.1 XGBoost (Machine Learning)

XGBoost (Extreme Gradient Boosting) is a machine learning algorithm that has gained popularity over the years and is one of the best methods available today to solve regression and classification problems. It extends a regular gradient boosting framework and in each step builds trees sequentially. The more trees in an ensemble, the stronger and more accurate the model. The thing that really sets XGBoost apart is that it uses second-order derivatives to optimize the objective function which makes training faster and increases accuracy. XGBoost being the base learner used in the stacking ensemble, takes advantage of gradient-boosted trees for nonlinear effects with regularization for variance control over a single-monthly forecasting of retail (Karaoğlu & Söyler, 2025).

2.5.2 Random Forest (Machine Learning)

Random Forest is an ensemble learning technique based on decision trees. It builds many decision trees during the training and gives an average prediction for regression or a majority vote for classification. The forest is made up of trees that are trained on a bootstrap sample of the training data, and at each split only a random subset of features is used. This randomness produces diversity of trees and makes the model more robust. The importance analysis confirmed the preeminence of environmental drivers and RF as fundamental for , predictive accuracy and interpretability of hydro-morphological processes (Jodar-Abellan et al., 2025).

2.5.3 Extra Tree (Machine Learning)

Another ensemble learning method similar to Random Forest is Extra Trees, which adds more randomness in the tree building process, with split thresholds also selected randomly. This method is useful to increase tree diversity, reduce variance and overfitting and often improves generalization on unseen data. This technique also works very well when applied to high dimensional data or data sets with noisy features . The randomness reduces the possibility of your model becoming too sensitive to random noise . Its significance in a medical dataset that contains noise and outliers is particularly remarkable as the stacked architecture can show better accuracy, cost and robustness than individual base learners and the fuzzy decision tree benchmark (Kumar et al., 2022).

2.5.4 Multiple Linear Regression (MLR) (Statistic)

Multiple Linear Regression (MLR) is a statistical equation that describes the relationship between a dependent variable and independent variables. It is founded on a linear assumption, meaning that the response can be modelled as linear

combinations of predictors plus an error term. The goal of MLR is to get the values of some coefficients for the predictors so that the error is minimized, typically by least squares. MLR can be used to capture the relationship between a dependent variable that we are going to predict based on multiple independent variables. Furthermore, recent maintainability knowledge also indicates MLR relevance as it can provide high-precision predictions in context-specific datasets (e.g., water quality index), thus a solid baseline for model evaluation (Jafar et al., 2023).

2.5.5 Multiple Linear Regression (MLR) (Statistic)

According to this study (Shoko et al., 2025), ensemble stacking is a method that combines predictions from multiple base learners through a meta-learner to enhance classification performance. Their framework used Support Vector Machine (SVM), Gaussian Naive Bayes and k-Nearest Neighbor (KNN) as base learners and the meta-learner combined their results to form the final prediction. The ensemble stacking approach is targeted to overcome the limitations of individual algorithms and exploit their advantages simultaneously to improve predictive reliability. In disease classification of Anemia, stacking also consistently improved all performances (El-Boghdady et al., 2023).

III. RESULTS AND DISCUSSION

All experiments were done in Python using pandas, numpy, scikit-learn, xgboost and some other supporting libraries. To allow for independent replication, we used pre-determined hyperparameter grids as well as leakage-safe chronological splits and explicit feature definitions. The data were split into training months 3–27, validation months 28–32 and test month 33. This dataset contained transactional retail records covering multiple shops, product categories and individual items spanning over two years period. Each record included these fields: sales volume, price, time index, and shop and product identifiers. For modelling purposes, the raw daily transactions were aggregated into a monthly shop–item panel. Months without transactions were explicitly assigned zero sales values to preserve a complete temporal structure.

Feature engineering was performed under strict chronological constraints to prevent data leakage. Temporal lag variables and rolling statistics were generated to capture demand momentum and seasonal patterns. Price and activity descriptors were constructed to reflect elasticity and promotional intensity. Hierarchical and categorical encodings were introduced to represent shop- and item-level effects, while calendar variables were used to encode year, month, and cyclical seasonality. Together, these features provided a comprehensive basis for both linear and nonlinear forecasting models.

Model development was carried out in several stages. First, ridge-regularized linear regression was implemented as an interpretable statistical baseline, with the regularization strength tuned to reduce overfitting while preserving interpretability. Random Forest was then trained using multiple decision trees built on bootstrapped samples, with tree depth, number of estimators, and feature subsets tuned to improve generalization. Extra Trees adopted a similar configuration but introduced additional randomness in split selection to further reduce variance. XGBoost was developed as a gradient boosting model in which sequential trees were optimized to minimize residual errors; its learning rate, maximum depth, subsampling ratios, and regularization terms were carefully tuned to improve robustness. Finally, an ensemble stacking model was constructed using the out-of-fold predictions from MLR, Random Forest, Extra Trees, and XGBoost as inputs to a RidgeCV meta-learner. This meta-model learned the optimal weighted combination of base model predictions to produce the final forecast.

The validation results of these models showed distinct differences in performance. MLR with RidgeCV achieved an RMSE of 0.7929 and MAE of 0.2875, offering a stable but limited baseline. Random Forest improved upon MLR with lower MAE at 0.2839, though its RMSE of 0.7967 showed only marginal gains. Extra Trees achieved a stronger balance, with RMSE of 0.7749 and MAE of 0.2880, benefiting from increased randomness and stronger generalization. XGBoost, despite its theoretical strength, produced weaker validation performance in this setting, with RMSE of 0.8751 and MAE of 0.3439, reflecting sensitivity to feature noise and the complexity of hyperparameter optimization. The ensemble stacking model delivered the best overall outcome, with RMSE of 0.7689 and MAE of 0.3066, demonstrating that combining multiple learners through stacking provided greater stability and accuracy than any single model alone.

As these results show, linear methods such as MLR are useful for benchmark purposes but insufficient for capturing the nonlinear interactions that govern retail demand. The resulting tree-based ensembles (random forest and Extra trees), fitted those intricate inter-dependency relationships of feature values, thus improving upon performance whereas XGBoost was flexible as well in this circumstance but fitting the data could lead to more instability in this case. The best results were obtained with the stacking ensemble by learning to balance between high interpretability, low variance and minimum error of all base models. This justifies the importance of ensemble stacking as a sales

forecasting approach, especially when seasonality and drift are to be expected.

Table 5. Sales Dataset

Model	RMSE (Val)	MAE (Val)
Ensemble Stacking	0.7689	0.3066
MLR	0.7929	0.2875
Random Forest	0.7967	0.2839
Extra Trees	0.7749	0.288
XGBoost	0.8751	0.3439

On the validation window (months 28–32), the stacking model achieved RMSE 0.7689 and MAE 0.3066 (Table 5). To support the comparison in Table 5, we report month-wise RMSE and MAE on the validation window (months 28–32) together with the corresponding mean ± standard deviation. The stacking model shows lower average errors and smaller month-to-month variation than the strongest single learner, indicating that the gains are consistent across validation months.

Under heavy zero-inflation and sparse demand, XGBoost showed higher error spikes and sensitivity to settings. During preprocessing, about 6,134,536 NaN entries were set to zero out of 6,734,448 records, leaving only 599,912 rows with actual transactions; this made lag and rolling-mean features mostly zero and highly skewed. Under these sparse signals, XGBoost tended to overfit and generalized poorly in later windows. In contrast, RidgeCV-based stacking reweighted base learners with regularized coefficients, improving stability and predictive robustness across seasonal segments. Removing hierarchical means and calendar cycles produced noticeable degradations in accuracy (higher RMSE), confirming their role in modeling seasonal level shifts. Excluding Extra Trees increased RMSE, while omitting XGBoost reduced stability during sparse months (larger error spikes), which explains their positive meta-weights. Replacing the RidgeCV meta-learner with plain OLS preserved average MAE but increased month-to-month variance, underscoring the stabilizing effect of L2 regularization.

Our contribution is practical rather than theoretical: a leakage-safe monthly panel, fold-internal encoders, compact discrete grids, and an interpretable RidgeCV stacker that reduces variance while preserving transparency. This combination consistently improves RMSE/MAE under seasonality and drift on a sparse, zero-inflated retail dataset, offering a robust recipe that can be reproduced and deployed with minimal adaptation.

IV. CONCLUSION

On the leakage-safe monthly panel, Extra Trees emerged as the strongest single model, while Random Forest and ridge-regularized linear regression remained competitive under time-aware validation. XGBoost showed weaker performance in this setting. The proposed stacking framework, which combined MLR, Random Forest, Extra Trees, and XGBoost through a RidgeCV meta-learner, achieved the lowest RMSE (0.7689), indicating improved performance in reducing larger forecast errors compared with the individual models. This improvement, however, did not translate to MAE where several of the single models outperformed the stacked model and therefore indicates that the benefit from stacking in this study was larger with respect to stabilizing overall error than for absolute average error reduction. These results imply that forecast methods based on the principles of ensemble stacking under seasonal variation and evolving demand can be valuable retail forecasting strategy, especially given a validation by leakage-safe chronological validation of these methods. These results further imply that different non-homogeneous base learners enable the meta-learner to make use of predictive signals that are only imperfectly available to any one individual model.

This study has several limitations. Retail demand remains affected by zero inflation, sparse histories, and cold-start items, while the feature set does not yet incorporate richer exogenous information such as promotions, holidays, or macroeconomic factors. Future work should therefore examine adaptive retraining under drift, the inclusion of external covariates, and probabilistic forecasting approaches such as quantile prediction to better support inventory, replenishment, and staffing decisions.

AUTHOR CONTRIBUTION

AASG conceived and supervised the study. PY devised the methodology, developed the software, performed analysis, curated data, planned visualizations, supervised the project and wrote the manuscript. The validation, review and editing of the manuscript was performed by PY, AASG & HN. The authors have used ChatGPT for the purpose of language clarification and take full responsibility for the final content.

DATA AND MATERIAL AVAILABILITY

This study used the Predict Future Sales dataset, specifically the English-converted version curated by Darshan N and published on Kaggle. The dataset files are available through the corresponding Kaggle: *https://www.kaggle.com/datasets/ndarshan2797/english-converted-datasets/data*.

REFERENCES

Adewumi, I. O. (2026). Predictive Analytics for Sales Forecasting in Emerging Market Retail: An Ensemble Learning and Time-Series Approach for A&Z Supermarket. Community and Social Development Journal, 27(1), 95–114. https://doi.org/10.57260/csdj.2026.286583

Al-Bazi, A., H. Al-Salami, Q., Zulfikar, N., Jerjees, Z., & A. Ahmad, M. (2025). Advanced Performance Analysis in Sport Footwear Sales Prediction Using Machine Learning. The 5th International Scientific Conference on Administrative and Financial Sciences (CIC-ISCAFS'2025), 14–21. https://doi.org/10.24086/icafs2025/paper.1582

Attipoe, E. K., Yussiff, A. S., Asante-Mensah, M. G., Tetteh, E. D., & Turkson, R. E. (2025). An ensemble learning approach for diabetes prediction using the stacking method. Computer Science and Information Technologies, 6(2), 102–111. https://doi.org/10.11591/csit.v6i2.p102-111

Bihri, H., Charaf, L. A., Azzouzi, S., & Charaf, M. E. H. (2025). A Robust Stacking-Based Ensemble Model for Predicting Cardiovascular Diseases. AI, 6(7), 160. https://doi.org/10.3390/ai6070160

Daza, A., Sánchez, C. F. P., Apaza-Perez, G., Pinto, J., & Ramos, K. Z. (2024). Stacking ensemble approach to diagnosing the disease of diabetes. Informatics in Medicine Unlocked, 44, 101427. https://doi.org/10.1016/j.imu.2023.101427

El-Boghdady, A. M., Kishk, S., Ashour, M. M., & AbdElhalim, E. (2023). Machine-learning based stacked ensemble model for accurate multi classification of CBC Anemia. Mansoura Engineering Journal, 49(3), 4. https://doi.org/10.58491/2735-4202.3144

Googerdchi, K. F., & Jafari, S. M. (2025). Sales Forecasting in Retail Using Random Forest Regression: A Case Study on Walmart. 2025 4th International Conference on Computing and Information Technology (ICCIT), 592–597. https://doi.org/10.1109/ICCIT63348.2025.10989314

Hasan, M. D. R. (2024). Addressing seasonality and trend detection in predictive sales forecasting: A machine learning perspective. Journal of Business and Management Studies, 6(2), 100–109. https://doi.org/10.32996/jbms.2024.6.2.10

Jafar, R., Awad, A., Hatem, I., Jafar, K., Awad, E., & Shahrour, I. (2023). Multiple Linear Regression and Machine Learning for Predicting the Drinking Water Quality Index in Al-Seine Lake. Smart Cities, 6(5), 2807–2827. https://doi.org/10.3390/smartcities6050126

Jodar-Abellan, A., Van Oudenhove, M., De Vente, J., Boix-Fayos, C., & Eekhout, J. (2025). Predicting bankfull channel dimensions through Stepwise Multiple Linear Regression and Random Forest in intermittent Mediterranean streams.   In EGU General Assembly Conference Abstracts (pp. EGU25-11060). https://doi.org/10.5194/egusphere-egu25-11060

Karaoğlu, A., & Söyler, H. (2025). A New Approach for Improving Biodiesel Conversion Efficiency: A Stacking Ensemble Model Based on Linear Regression Approach with GAN-Enhanced. Arabian Journal for Science and Engineering, 50(23), 19421–19441. https://doi.org/10.1007/s13369-025-10227-5

Kumar, M., Singhal, S., Shekhar, S., Sharma, B., & Srivastava, G. (2022). Optimized Stacking Ensemble Learning Model for Breast Cancer Detection and Classification Using Machine Learning. Sustainability, 14(21), 13998. https://doi.org/10.3390/su142113998

Li, X. (2025). A study on monthly sales forecasting of new energy vehicles in urban areas using the WOA-BiGRU model. PloS One, 20(4), e0320962. https://doi.org/10.1371/journal.pone.0320962

Mame, M., Huang, S., Li, C., & Zhou, J. (2025). Application of Extra-Trees Regression and Tree-Structured Parzen Estimators Optimization Algorithm to Predict Blast-

Induced Mean Fragmentation Size in Open-Pit Mines. Applied Sciences, 15(15), 8363. https://doi.org/10.3390/app15158363

Manzanas, R. (2020). Assessment of Model Drifts in Seasonal Forecasting: Sensitivity to Ensemble Size and Implications for Bias Correction. Journal of Advances in Modeling Earth Systems, 12(3), e2019MS001751. https://doi.org/10.1029/2019MS001751

N, P. H., P, K. D., M, S., & R, S. M. (2025). Sales Data Visualization and Future Predictions Using Power BI and Machine Learning. 2025 IEEE International Conference on Compute, Control, Network & Photonics (ICCCNP), 1–6. https://doi.org/10.1109/ICCCNP63914.2025.11233722

Shoko, C., Sigauke, C., & Makatjane, K. (2025). An Application of Ensemble Stacking in Machine Learning to Predict Short-term Electricity Demand in South Africa. Statistics, Optimization & Information Computing, 13(6), 2412–2433. https://doi.org/10.19139/soic-2310-5070-2170

Wang, M., Liu, Y., Li, G., Yue, Y., & Man, K. L. (2024). Unlocking Your Sales Insights: Advanced XGBoost Forecasting Models for Amazon Products. International Conference on Advanced Multimedia and Ubiquitous Engineering, 181–187. https://doi.org/10.1007/978-981-95-1565-3_23

Wang, N., & Li, Z. (2022). A stacking-based short-term wind power forecasting method by CBLSTM and ensemble learning. Journal of Renewable and Sustainable Energy, 14(4). https://doi.org/10.1063/5.0097757

Windrasari, S. N., Margono, H., & Putra, Y. A. N. S. (2025). Predictive Sales Analysis in Coffee Shops Using the Random Forest Algorithm. MALCOM: Indonesian Journal of Machine Learning and Computer Science, 5(3), 1000–1011. https://doi.org/10.57152/malcom.v5i3.2023

Yao, B. (2023). Walmart Sales Prediction Based on Decision Tree, Random Forest, and K Neighbors Regressor. Highlights in Business, Economics and Management, 5, 330–335. https://doi.org/10.54097/hbem.v5i.5100