FMCG Retail Sales Forecasting Using Machine Learning on Walmart M5 Dataset

Sri Ram Sathiya Narayanan[1] and Dr.Tala Talaei Khoei[2]

[1]Khoury College of Computer Sciences, Northeastern University, Boston
[2]Khoury College of Computer Sciences, Northeastern University, Portland

***Abstract*—Predicting retail sales involve difficulties that hugely are caused by seasonality, product mix heterogeneity, and promotional events that further contribute to demand. The present work, through a very selective machine learner pipeline applied to the M5 Walmart data, attempts to solve the above-mentioned challenges. The M5 dataset, along with the information about the calendar, price, and promotion, has been significantly extended and integrated. At the first stage of our approach, the research team performs in detail the data preparation process which is followed by feature engineering aimed at handling the trends of the data sets, the moving average, and the datable-based indicators of the holidays and SNAP days. We have built several supervised regression models - Lasso, Ridge, Decision Tree, and LightGBM - that have been optimized through the randomized hyperparameter search in the local time-series validation approach. The results of using metrics like RMSE, R^2, and RMSSE show the effectiveness of the LightGBM model which is the most efficient in handling the complex interaction of the variables. Generally, through multiple metrics, the assessment of our pipeline performing to capture seasonal and promotional changes has been a success, resulting in precise and dependable forecasts. This research provides a balancing forecasting toolkit that is not only powerful in terms of prediction but also business-relevant, thus it can be readily used for FMCG retail inventory and supply chain management to generate efficient and actionable solutions.**

Keywords: Time series forecasting, feature engineering, gradient boosting, LightGBM, Lasso regression, Ridge regression, hyperparameter tuning, cross-validation, lag features, rolling mean, expanding window.

I. Introduction

Foreseeing demand with precision is the core of successful business in the retail sector. This, in turn, greatly influences the way store shelves are refilled, the staff hours are scheduled, and, finally, the profits are optimized. The forecast of coming sales is a tool that grants firms the ability to regulate the supply chain matching the market requirements, become free from the threat of lacking or overstocking, and effectively organizing the shipping and marketing plans. Nevertheless, retail demand forecasting is still very tricky because of multiple temporal factors such as yearly cycles, occasional discounts, festive effects, and data sparsity of product-store combinations.

Conventional Statistical methods like ARIMA and exponential smoothing (ETS) have been generally used for time series forecasting and are not suitable to model non-linearities and high-dimensional data found in large and complex realtime datasets. The great machine learning potential is very much exposed via such techniques as LightGBM. This gradient boosting algorithm gives detailed information for interaction detection between time and categorical features that significantly increase predictive power and model resistance.

This paper is an in-depth Walmart M5 dataset study employing various modeling strategies that rely on/transition from feature engineering (lagged sales, rolling aggregates, and event encoding) to multiple regression models. The article is written as if it were a part of a large research program with strict cross-validation and RMSSE-based evaluation designed explicitly to prevent data leakage in time series. Our work features include:

- Advanced feature engineering that mirrors the nature of retail domain dynamics,
- RMSSE implementation for insightful accuracy measurement in time series context,
- Time being cross-validation designs that pledge fair evaluation results,
- Four-regression-model comparison with the emphasis on interpretability-accuracy trade-off relationships.

II. Problem Description

This work's main objective is to build an effective prediction system that can foresee product sales over many stores and categories of the Walmart M5 retail dataset. In essence, the challenge is the univariate and multivariate time series regression problem, with the goal of estimating sales $\hat{y}_t$ for each future time t, basing on historical observations y_t and features.

Formalizing this higher sales forecast problem, the product is expressed as a function of lagged sales and explanatory variables: where y_t are the real unit sales at time t, p is the number of lag periods, and X_t is the feature vector that consists of exogenous variables such

as production material prices, special event, weekday, holiday, and SNAP discount flags.

Mathematically, the forecasting function can be expressed as:

$$\hat{y}_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-p}, X_t)$$

The modeling horizon is N days ahead to predict sales, whereas data from the past five years will be used for model training. In an attempt to simulate a realistic forecasting scenario, the dataset has been split in a time-ordered manner, i.e., early periods are for training and later periods for validation and testing, thus completely eliminating the chance of temporal leakage.

Among essential limitations are the imposition confining features to be causal (no future-looking variables) and the handling of non-stationarity arising from seasonal sales, sales promotions, and very few transactions at the product level. Such a configuration requires models to trade-off bias and variance efficiently and at the same time be able to generalize over the different product-store combinations.

The present formulation covers the statistical and practical aspects of retail forecasting and, therefore, can be used to compare machine learning methods such as regularized linear models and gradient-boosting algorithms in realistic business scenarios.

III. Methodology

The methodology was a structured plan that combined thorough data preprocessing, elaborate feature engineering, and controlled model evaluation targeted at temporal consistency. The dataset was the Walmart M5 Forecasting dataset that contained the data on more than 3000 products in 10 stores located in 3 states in the U.S. spanning over 1941 days. The dataset has three main files: `sales_train_evaluation.csv` (historical daily sales per product-store), `sell_prices.csv` (weekly item prices), and `calendar.csv` (date events, SNAP flags, and weekday info).

A. Preprocessing

Data preprocessing was comprised of:

- Missing value handling: Prices that were missing were imputed via store-item median values and empty event data were filled with "No_event".
- Outlier treatment: The interquartile range (IQR) based capping was used for numeric features to limit the effect of extreme values.
- Memory optimization: Numeric data types were downcasted in order to reduce RAM usage without losing precision.

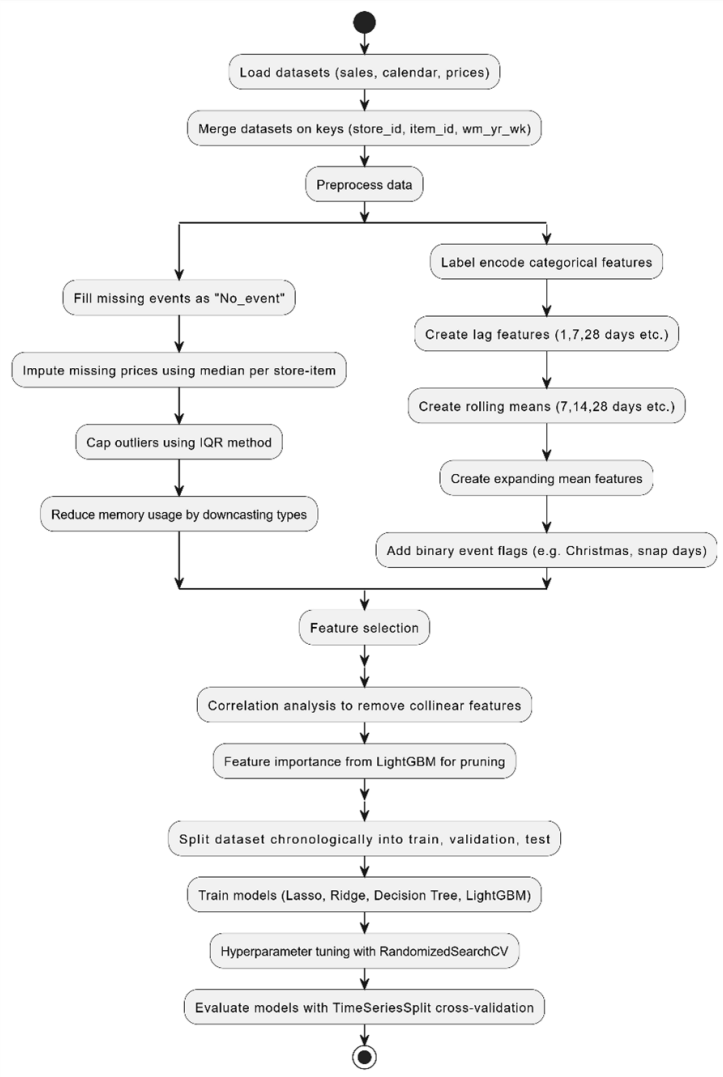

Fig. 1: Workflow

B. Feature Engineering

Feature creation revolved around augmenting both temporal and categorical signals. To capture autocorrelation, lag features (y_{t-1}, y_{t-7}, y_{t-28}) were created, followed by the calculation of rolling statistics (mean, standard deviation) as well as cumulative expanding averages. The use of calendar encoding such as `is_weekend`, `is_high_sale_months`, and `is_christmas` helped to model seasonality as well as event-driven volatility. Categorical variables such as `store_id`, `event_name_1`, and `state_id` were label-encoded in order to make discrete identifiers numerical, which is the form suitable for model ingestion.

C. Feature Selection

In order to tackle the problem of multicollinearity and that of high dimensionality, we:

- Investigated feature correlations to identify clusters of highly correlated variables.
- Employed LightGBM feature importance scores to select only the most predictive feature per correlated group.
- Removed redundant features including that of highly correlated rolling means and categorical duplicates.

D. Data Splitting

The data was divided in a time ordered manner into training, validation, and testing sets so that the process of forecasting would be close to that of the real world and future data leakage would be avoided:

- Training: First $\sim 1,678$ days
- Validation (cross-validation folds): Days $\sim 1,679$ to $1,913$
- Testing: Last 28 days onward

This time-aware split keeps the temporal aspect intact and thus ensures that the evaluation is not biased.

E. Models

1) Lasso Regression (L1 Regularization): This model uses a penalty which is proportional to the absolute value of the coefficients, thus reducing the density by forcing some coefficients to be exactly zero. By this method, the model implicitly performs feature selection, which helps to understand the model better and helps to lower the risk of overfitting. Formally, Lasso minimizes the objective:

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \beta)^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right\}$$

where α determines the level of regularization. In the case of sales time series forecasting, this feature helps to put the focus on those features that are most helpful for the prediction task, while at the same time the noise is controlled.

2) Ridge Regression (L2 Regularization): Ridge, on the other hand, penalizes the square of the coefficients, thereby forcing them to be close to zero, but it does not remove the variables. This helps to reduce the variance of the model in the case where there is multicollinearity among the features which, in this case, are lagged and rolling sales indicators. The objective of Ridge is:

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \beta)^2 + \alpha \sum_{j=1}^{p} \beta_j^2 \right\}$$

Ridge frequently generalizes better to complex datasets where numerous correlated predictors are present, which matches the nature of retail sales data.

3) Decision Tree Regression: Decision tree regressors split the feature space through recursive binary partitions, thereby generating piecewise constant estimations of the target variable. Trees gain the ability to model nonlinear interactions and threshold effects inherently but at the risk of overfitting if left unbounded. Therefore, controlling depth or pruning is essential. Decision trees yield interpretable segmentations that can help in discovering product-store specific patterns.

4) LightGBM (Gradient Boosted Decision Trees): LightGBM creates a set of models by consecutively fitting new trees to the gradient (residual) of previous models' errors, thus directly reducing loss. Several parameters are provided to limit learning rate, tree depth, and leaf counts to regulate bias-variance tradeoff. The model is capable of dealing with heterogeneous features, missing values, and intricate nonlinearities, which is in line with the better forecasting results.

F. Training/Test Split Logic

The data were divided in time order with the earliest periods being utilized for training, the next time window for validation and the last segment for testing. Such a temporal split safeguards against leakage of future information, thereby confirming that our model's performance measures are a close match to the real-world forecasting scenarios in which future data are not accessible at the time of prediction.

G. Hyperparameter Tuning and Cross-Validation

We resorted to RandomizedSearchCV as our main hyperparameter tuning method because of its capability to efficiently explore large and complicated hyperparameter spaces without requiring the exhaustive computations of GridSearchCV. Contrary to grid search that deterministically evaluates every point in the hyperparameter grid, RandomizedSearchCV picks random combinations from the predefined distributions thus allowing a broader search with fewer iterations.

1) Parameter Grids: We delineated the key parameter ranges for each of the models that were informed by both domain expertise and preliminary exploratory runs, comparisons made based on their performance and time taken with standard metrics along with graph. Parameters such as Alpha for Lasso and Ridge regression and `num_leaves`, `max_depth`, `learning_rate`, `n_estimators` for LGBM were utilized.

The LightGBM's `num_leaves` and `max_depth` hyperparameters, for instance, were very well balanced: while increasing the former allowed the model to capture more feature interactions thus leading to better generalization and thus a higher R^2, the latter helped control overfitting hence the final R^2 improvement is due to both factors.

2) Cross-Validation Strategy: TimeSeriesSplit with three folds was our choice to represent forecasting scenarios accurately. This is because it does not shuffle data randomly and thus keeps temporal causality intact. Every fold was a chronological division having train-validation windows of adequate length for proper model training and testing.

The reason why no shuffling was done is that it preserves the order of the data which is of great importance for time series so as not to have cases where information from the future is leaked into training.

3) Tuning Outcomes: During tuning, models' RMSE and R^2 scores were computed for each fold, resulting in mean $\pm$ standard deviation summaries that point to consistent performance enhancement after parameter adjustment. RandomizedSearchCV curtailed the overfitting problem by setting the right regularization and the tree complexity parameters that mirror the temporal dynamics in the dataset.

IV. Results and Evaluation

The different measures were used to quantitatively evaluate the model's performance to reflect the real picture of the assessment. Those measures were: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Root Mean Squared Scaled Error (RMSSE). These measures collectively serve the purpose of measuring the deviation magnitude, the relative fit, and temporal scaling performance.

In general, LightGBM was able to maintain the best balance between bias and variance and thus, it was superior to both linear and tree-based baselines in terms of all the metrics considered. Regularized linear models (Lasso and Ridge) yielded smooth forecasts with interpretable trends, whereas the decision tree model overfitted due to the fact that its depth was not limited and it was not regularized.

A. Interpretation of Performance

One of the main reasons of LightGBM performance being the best is that it can hierarchically model the interactions between price, promotion, and seasonality by doing boosting over residuals. The regularization parameters such as shallow depth and learning rate are there to help prevent overfitting. On the other hand, decision trees, which do not have ensemble averaging, are not able to generalize across temporal discontinuities.

Regularized regressions keep up with the competition and can be used for interpretable forecasting, feature engineering being the main factor and correlation-based selection for noise reduction. Altogether, the models demonstrate the advantages of temporal features engineering and the use of non-linear machine learning methods for retail forecasting which is robust.

Before Tuning

Model	MSE	RMSE	MAE	R^2	RMSSE
Lasso	0.45	0.6708	0.5211	0.5016	0.5469
Ridge	0.4273	0.6537	0.4754	0.5266	0.5356
Decision Tree	0.9552	0.9773	0.6096	-0.0581	0.7969
LightGBM	0.4273	0.6537	0.4754	0.5266	0.5332

Fig. 2: Before Tuning

After Tuning

Model	MSE	RMSE	MAE	R^2	RMSSE
Lasso (α=0.001)	0.4316	0.657	0.4835	0.5219	0.5356
Ridge (α=0.1)	0.4315	0.6568	0.4831	0.5221	0.5356
LightGBM (leaves=100, depth=1, est=300, lr=0.1)	0.4321	0.6573	0.4859	0.5214	0.5361

Fig. 3: After Tuning

Cross validation

Model	Average RMSE ± SD	Average R^2 ± SD
Lasso	0.5962 ± 0.0305	0.5694 ± 0.0359
Ridge	0.5831 ± 0.0301	0.5882 ± 0.0347
Decision Tree	0.8671 ± 0.0518	0.0887 ± 0.0914
LightGBM	0.5780 ± 0.0290	0.5954 ± 0.0330

Fig. 4: Cross validation

V. Reflections and Discussion

The comparative study of the models that we performed reveals that the selected models have different strengths and weaknesses. The tree-based LightGBM was able to capture complex non-linear interactions and hierarchical dependencies inherent in multi-store, multi-product sales data and thus it consistently outperformed other models. Its gradient boosting approach was able to make a great use of the hyperparameter tuning, most notably the changes to the number of leaves and tree depth which had a significant impact on the coefficient of determination. On the other hand, simpler interpretable models like Lasso and Ridge were able to demonstrate quite good performance due to regularization, thus noise and multicollinearity influence were decreased, although these models could hardly handle nonlinearities.

Without boosting, the decision tree model may have overfitted the training data, as evidenced by the poorer cross-validation scores. This fact points to the need for ensemble methods that can be used for retail forecasting and are able to generalize.

Feature engineering was crucial in facilitating model accuracy. Lag features, rolling averages, and expanding window statistics all served as effective means for capturing temporal autocorrelation, while event flags -

like holiday indicators and SNAP days - were valuable domain-specific signals that accounted for sales declines during national holidays and spikes during promotional events. This domain awareness was critical in reducing forecast error and improving business relevance.

Besides that, strict temporal cross-validation through TimeSeriesSplit was free of future leakage and thus it allowed to get quite realistic assessments of model performance and it also prevented over-optimistic generalization claims.

Overall, the selection of models should be a trade-off between interpretability and predictive capabilities where LightGBM could be considered as a potent instrument if used together with thorough feature engineering and tuning. Features constructed with the help of domain knowledge and proper validation strategies are indispensable if one wants to operationalize sales forecasts that will be instrumental in managing inventory in an optimized way and strategic planning.

VI. Conclusion

The research features a powerful machine learning system for retail sales prediction, which has been tested on the difficult-to-handle Walmart M5 dataset. The major outcomes first of all point to the fact that by merging the domain knowledge-driven feature engineering with the advanced regularized linear models and the boosted decision trees one can improve the forecasting accuracy considerably, which is evidenced by the lower RMSE and higher R^2 and RMSSE scores.

The advantage of LightGBM over other methods indicates that it is very important to model complex nonlinear interactions as well as hierarchical sales dynamics in retail data. By means of Lasso and Ridge regularization the model overfitting is avoided thus the obtained forecasts are stable and interpretable. The application of time-series cross-validation during hyperparameter tuning in our experiments provides a fair and realistic evaluation that closely resembles the actual forecasting scenarios.

In fact, the suggested method allows retailers to recognize demand changes caused by seasonality, promotions, and events like SNAP days and thus to avoid both stockouts and the costs of excess inventory. Deep learning architecture that could capture long-range temporal dependencies more effectively is one of the limitations of this work along with only using structured retail data without the external economic indicators.

Work to be done in the future might consider hybrid models that combine LSTM networks and attention mechanisms with gradient boosting to achieve interpretability and sequence learning at the same time. Moreover, macroeconomic and competitor data can be included to make the forecast even more reliable and have more significant business impact. The present research is a step forward in the application of temporal machine learning in retail and a great help in inventory and supply chain management optimization that is based on data.

References

[1] Kaggle. (2020). M5 Forecasting - Accuracy Competition. Kaggle Competitions. [Online]. Available: https://www.kaggle.com/competitions/m5-forecasting-accuracy

[2] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Proc. of the 31st International Conf. on Neural Information Processing Systems (NeurIPS)*, 2017.

[3] V. Assimakopoulos and K. Nikolopoulos, "The theta model: a decomposition approach to forecasting," 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207021001874bb2

[4] C.S. Bojer and J.P. Meldgaard, "Kaggle forecasting competitions: An overlooked learning opportunity," 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207021001874bb4

[5] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pp. 785–794, 2016.

[6] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M5 Accuracy Competition: Results, Findings, and Conclusions," *International Journal of Forecasting*, Elsevier, 2020.

[7] C. Nicault, "M5 Forecasting Accuracy Competition Overview," 2021. [Online]. Available: https://www.christophenicault.com/post/m5 forecasting accuracy

[8] E. Spiliotis, R.J. Hyndman, and S. Makridakis, "Weighted and Root Mean Squared Scaled Error (RMSSE) in Hierarchical Forecasting Evaluation," *International Journal of Forecasting*, 2021.