Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Sales Forecasting for Retail Business using XGBoost Algorithm

Prathana Dankorpho
Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok
Corresponding Author: Prathana Dankorpho, **E-mail**: prathana.dkp@gmail.com

| ABSTRACT

The retail industry is continuously evolving with the expansion of sales channels and the diversification of product assortments. However, current forecasting methods, relying on simplistic statistical models, frequently encounter difficulties in adjusting to the dynamic retail environment. This limitation leads to challenges in accurately predicting sales volumes and frequencies. Consequently, there is a critical need to improve the accuracy and frequency of sales predictions to enable timely decision-making for business strategies. Through a comprehensive analysis of datasets spanning from 2019 to 2023, this study illustrates the substantial advantages of integrating eXtreme Gradient Boosting (XGBoost) to gain deeper insights into sales patterns. Results demonstrate a significant enhancement in prediction accuracy, with an average reduction of 29.23% in Mean Absolute Error (MAE) and 34.54% in Root Mean Squared Error (RMSE) compared to conventional methods. Furthermore, the adoption of XGBoost facilitates the transition from monthly to daily forecasting, thereby optimizing the efficiency of the prediction process. Retailers can optimize inventory management, devise effective marketing strategies, and ultimately maximize revenue. The findings emphasize the importance of embracing innovative approaches to address the challenges of a rapidly evolving retail landscape and drive sustainable business growth.

| KEYWORDS

EXtreme Gradient Boosting, Machine learning, Nonlinear data, Retail business, Sales Forecasting, XGBoost.

| ARTICLE INFORMATION

ACCEPTED: 02 June 2024 **PUBLISHED:** 20 June 2024 **DOI:** 10.32996/jcsts.2024.6.2.15

1. Introduction

Retail businesses are growing incessantly, and there are various channels where customers can reach the products other than offline channels, as in the past, but expanded to online channels such as their own website, e-commerce platforms, or social media. Thus, forecasting sales becomes challenging because of diverse variables and continual changes in sales that are generated from all channels. The current processes have been done using basic statistics, and the straight-line average by humans is defined in this paper as the original method. Thus, it takes a longer time to make predictions than computing from machine learning. Moreover, the Original Method can provide only monthly forecasts, limiting the ability of retailers to respond promptly to changing market dynamics and consumer behavior. To address these challenges, numerous studies have explored sales forecasting methodologies, categorized into time series models and machine learning approaches. Time series models have been proven to be suited to handling linear data patterns, whereas machine learning performs better on nonlinear data. Given the nonlinear nature of our dataset, characterized by sales variations influenced by product discounts and customer behavior, using machine learning to address these complexities becomes imperative. The paper proposes the application of XGBoost to enhance prediction accuracy and frequency, enabling daily sales forecasts.

2. Literature Review

XGBoost was introduced as a powerful machine learning algorithm that excels in both speed and performance (Chen and Guestrin, 2016). It's built upon supervised machine learning, decision trees, and gradient boosting. Firstly, supervised machine learning employs algorithms to train a model, finding patterns in a dataset comprising features and corresponding labels. Subsequently,

the trained model applies these insights to predict labels for new datasets based on their features. Secondly, decision trees construct models that predict labels by evaluating a hierarchical structure of if-then-else feature questions and estimating the minimum number of questions needed to assess the probability of making a correct decision. Decision trees can be used for both regression and classification tasks, wherein they forecast either categorical or continuous numeric values, respectively. Their evaluation is based on Entropy and Information Gain. Entropy (E) quantifies the impurity of a node, reflecting its heterogeneity according to the formula :

$$\mathrm{E} = -\sum_{i=1}^{C} p_i \times \log_2(p_i)$$

where C is the number of classes and p_i represents the proportion of the i^{th} class within the set. Information Gain is then calculated to achieve a concise and effective tree with superior classification accuracy by comparing the entropy post and pre-split on an attribute. The attribute yielding the highest information gain is selected, and the process iterates across each branch. Lastly, Gradient boosting is an ensemble learning process that combines predictions from several models into one. Models are built sequentially to correct the errors of the previous ones, giving more weight to predictors that perform better. This process minimizes the loss function using a gradient descent algorithm. Beginning with the fitting of the first model using the original data, subsequent models are fitted using the residuals of the previous ones. The outcome is a strong predictive model created through this iterative process, as illustrated in Figure 1.

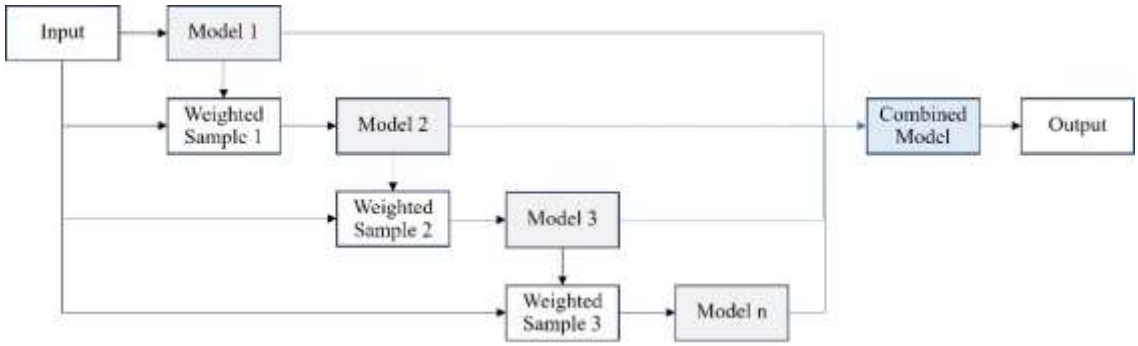

Figure 1: Gradient boosting ensemble learning process

Since XGBoost was introduced, it has been used to forecast various topics over time. (Ji et al., 2019) investigated sales forecasting using both Time Series Models (TSMs) and Machine Learning Algorithms (MLAs). Recognizing the limitations of TSMs, they proposed a combined approach using ARIMA and XGBoost models to capture both linear and nonlinear components of the data series, thereby overcoming the disadvantages of ARIMA because of their linear behavior. Similarly, (Wang and Guo, 2020) applied a mixed model of ARIMA and XGBoost for stock price forecasting. Their study showed improvement over a single model, especially in datasets with partial nonlinear components, which limit the performance of prediction in ARIMA. Integrating XGBoost helped explain nonlinear relationships and improve prediction accuracy by leveraging ensemble learning techniques. (Kalra et al., 2020) compared XGBoost with the tf-idf transform for predicting purchasing behavior on Black Friday. While XGBoost showed promising performance, the study highlighted the challenge of overfitting and recommended preprocessing steps to remove noise in the dataset before applying the model. In line with this, (Xia et al., 2020) demonstrated the advantage of XGBoost in predicting passenger car sales, enhancing accuracy through information gain and data correlation. They emphasized the importance of feature selection and proposed data filling algorithms to improve prediction accuracy, with mean filling outperforming other filling methods, and this approach will be adopted in this paper for the data cleansing process. Likewise, (Biswas et al., 2021) evaluated five algorithms, including LSTM, XGBoost, Linear Regression, Moving Average, and Last Value model for stock market price prediction, with LSTM performing best for their purpose. However, XGBoost ranked third in terms of performance, indicating its effectiveness and room for improvement in specific contexts. (Fildes et al.,2022) and (Pan, 2022) explored retail forecasting and price prediction for BMW, respectively, both concluding that machine learning algorithms, including XGBoost, provided more accurate forecasts for nonlinear data compared to traditional methods. Therefore, XGBoost has performed the best in Pan research. Guliyev and Mustafayev (2022) predicted changes in WTI crude oil price using machine learning models, with XGBoost consistently outperforming other metrics. They attributed XGBoost's success to its ability to select the most significant features and effectively improve prediction accuracy. (Ganapathy et al., 2022) Compared various algorithms for rainfall forecasting and found XGBoost to be the most accurate. According to enhancement from the conventional Gradient Boosting methodology, multiple decision trees have been assigned respective weights to explicitly give more importance to trees for determining the final output. Moreover, training time raking in the second, due to the parallel learning process, reduces the overall time for training by optimizing the

underlying hardware. (Mitra A., Jain A., Kishore A., et al., 2022) studied five regression techniques of machine learning, Random Forest (RF), XGBoost, Adaptive Boosting (AdaBoost), Artificial Neural Network (ANN), and hybrid (RF-XGBoost-LR) for demand forecasting in a multi-channel retail company. As a result, the hybrid model received the highest accuracy; both RF and XGBoost jointly overcame the problem of overfitting and training error in linear regression, concerned with determining the relationships between dependent and independent variables. Furthermore, XGBoost minimum number of resources with a parallel tree in ensemble method, rapid model exploration. Lastly, (Akanksha A., Yadav D., Jaiswal D., et al., 2022) concluded that XGBoost is best performed for store-sales forecasting. In conclusion XGBoost model has proven its ability on nonlinear data, outperforming in accuracy and efficiency across numerous research studies, making it a preferred choice for forecasting tasks in various domains.

3. Methodology

3.1 Data Preparation

The dataset in this study comprises daily sales transactions from a retail business, incorporating data from a total of 805 stores, including both online and offline channels. Gathered from a total of 3,988 products within five categories: watches, beauty, electronics, imported fashion, and owned brand fashion. This dataset spans over five years, extending from January 2019 to December 2023. Sales data within this research context are defined as net sales without VAT, encompassing sales after the deduction of discounts, trade GP, and Value Added Tax. The scope of sales considered in this analysis is limited to those generated from finished goods, thereby excluding premium items and testers. The Extract Transform Load (ETL) process is facilitated through three environments: SAP High-performance Analytic Appliance On-premises (SAP HANA), SAP Data Intelligence Cloud (SAP DI), and the Google Cloud Platform (GCP), as illustrated in Figure 2.

Figure 2: Extract Transform Load (ETL) process

Initially, sales data are gathered from SAP HANA and sourced through two distinct points contingent upon the distribution channels: SAP Customer Activity Repository (SAP CAR) for consignment and owned shops and SAP Enterprise Resource Planning (SAP ERP) for credit and online sales. Input sales from SAP CAR originate from scanning out at consignment locations via Mobile Inventory Management (MIM) or shops via Point of Sale (POS) systems, while credit and online sales are directly interfaced with SAP ERP via flat file. Data stored within SAP systems can be linked to SAP DI using Cloud Connector, establishing a direct connection to the respective tables if data conversion is unnecessary. However, for complex datasets, transformation, and modeling are necessitated through Core Data Service (CDS View) utilizing the Advanced Business Application Programming (ABAP) language. Subsequently, SAP DI serves as a facilitator between SAP HANA and GCP, akin to a bridge. Data pipelines are constructed within the modeler in SAP DI to effectuate data transformation utilizing available operators and Python scripts, subsequently exporting the transformed data to GCP in parquet format. This transformation process encompasses data cleansing, metadata-driven data type definition, and fundamental aggregation techniques aimed at ensuring data accuracy while minimizing data size. Finally, data are transmitted from SAP DI to GCP and stored within Cloud Storage. Herein, complex calculations are conducted within BigQuery to transform and store the finalized data for running prediction models leverage SQL, views, and stored procedures. During this transformation phase, the data is reconfigured into a time series dataset, with significant features derived from the structured grouping of product hierarchy and sales channels. Furthermore, mean filling techniques are applied to handle anomaly data to ensure data integrity and consistency while also mitigating the risk of overfitting.

3.2 Apply XGBoost Regression

The dataset has been partitioned into training and testing sets to evaluate model performance. Subsequently, the XGBRegressor was implemented within the Python environment, incorporating essential input features, target variables, and specified parameters for training. These parameters notably include n_estimators, maximum tree depth (max_depth), learning rate (eta), and subsample. To provide clarity and facilitate reference, detailed definitions of the parameters utilized in this study are presented in Table 1.

Table 1: Definitions for parameters defined in XGBoost model

Parameters	Definitions
n_estimators	No. of trees in ensemble model.
max_depth	Maximum depth of a tree refers to the upper limit of nodes from the root to the farthest leaf.
eta	The learning rate used to weight each model to prevents overfitting in each boosting step.
subsample	The number of samples training instances used in each boosting iteration to grow trees.

3.3 Optimization

Parameter tuning for the XGBoost model, which involves adjusting the number of trees, maximum tree depth, learning rate, and subsample ratios, plays a crucial role in optimizing model performance. As the values of these parameters increase, the complexity of the model also escalates. However, maintaining a delicate balance between prediction accuracy, model complexity, and computational efficiency is essential. The time consumed in running the model directly impacts the speed of business decision-making. Additionally, excessively high parameter values can precipitate overfitting, thereby compromising result accuracy. The analysis emphasizes that each product category has its own characteristics and patterns. This highlights the importance of customizing parameter settings to ensure the best performance of the model. All parameter specifics for each product category are summarized in Table 2.

Table 2: Parameters configured in the model by product category

Product Categories	n_estimators	max_depth	eta	subsample
Beauty	500	5	0.1	0.5
Electronic	700	9	0.1	0.5
Imported Fashion	250	9	0.1	0.3
Owned Brand Fashion	300	5	0.1	0.5
Watches	500	8	0.1	0.5

3.4 Evaluation

The results are evaluated by Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) using the formulas illustrated below.

$$\mathrm{MAE} = \left(\frac{1}{\mathrm{n}}\right) * \Sigma|y_i - x_i|$$

$$\mathrm{RMSE} = \sqrt{\left(\frac{1}{\mathrm{n}}\right) * \Sigma(y_i - x_i)^2}$$

4. Results and Discussion

The sales forecasts generated by the XGBoost algorithm are illustrated in Figure 3, expressed in Million Baht. The blue line represented the predicted values, while the orange line represented the actual sales numbers over time. The alignment between the predicted and actual figures within each product category is clearly illustrated, indicating the efficacy of the XGBoost algorithm in accurately forecasting sales across diverse product categories.

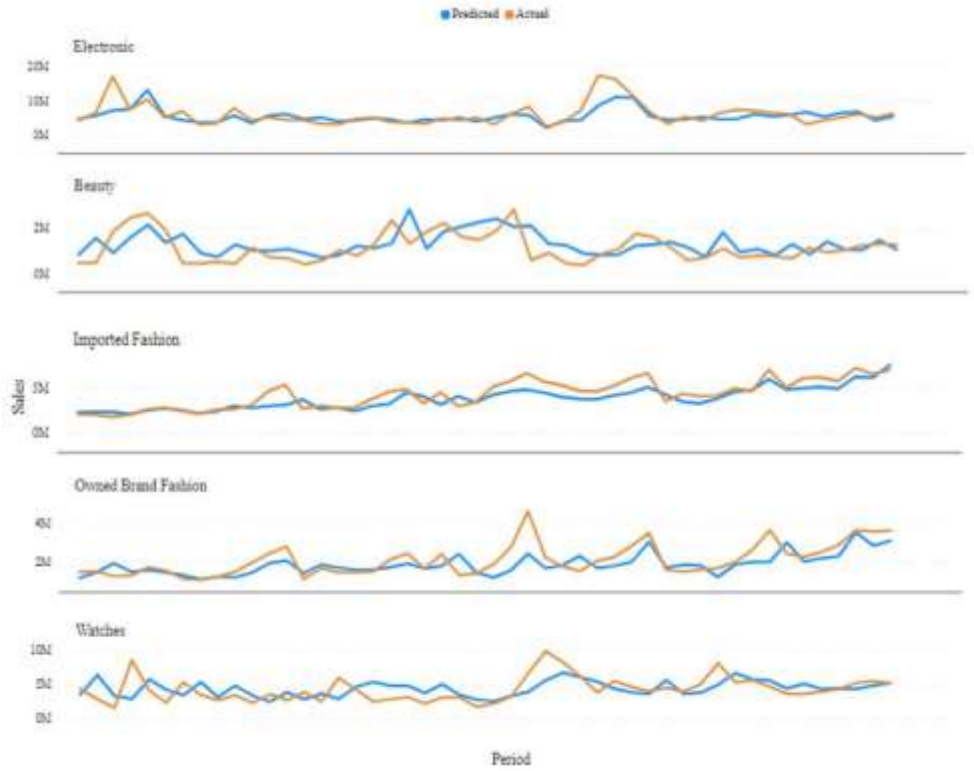

Figure 3: Comparison between sales prediction from XGBoost algorithm and actual value by product category

In order to compare the outcomes with the current prediction process, Table 3 presented a comparative analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) between the XGBoost model and the Original Method across various product categories.

Table 3: Comparison of the MAE and RMSE between XGBoost and Original Method

Unit: Million Baht	**XGBoost**		**Original Method**	
Product Categories	**MAE**	**RMSE**	**MAE**	**RMSE**
Beauty	0.49	0.61	1.15	1.91
Electronic	1.52	2.44	1.71	2.77
Imported Fashion	0.73	0.90	0.79	1.06
Owned Brand Fashion	0.46	0.63	0.86	1.43
Watches	1.39	1.79	2.01	2.54
Average	0.92	1.27	1.30	1.94

5. Conclusion

In this study, accuracy, measured through Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), exhibited improvement across all product categories. In summary, the average MAE of the Original Method was 1.30, whereas the XGBoost model achieved a substantially lower value of 0.92, indicating an enhancement in accuracy of 29.23%. Similarly, for RMSE, the Original Method yielded an average score of 1.94, whereas the XGBoost model demonstrated a lower value of 1.27, reflecting an increase in accuracy by 34.54%. Moreover, the adoption of the XGBoost model facilitated a transition from monthly to daily prediction cycles, thus escalating operational efficiency for businesses, enabling them to refine their planning and marketing strategies for superior performance. Future research can further enhance the accuracy and efficiency of the model by integrating additional forecasting algorithms such as ARIMA, LSTM, or other models and combining the strength of each model to overcome the predictive performance of the singular XGBoost model.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Chen T., and Guestrin C. (2016), XGBoost: A scalable tree boosting system, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794, doi: 10.1145/2939672.2939785
[2] Ji S., Wang X., and Zhao W., et al. (2019), An application of a three-stage XGboost-based model to sales forecasting of a cross-border e-commerce enterprise, *Mathematical Problems in Engineering*, 2019, doi: 10.1155/2019/8503252
[3] Wang Y. and Guo Y., (2020), Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost, *China Communications*, 17(3), 205-221.
[4] Kalra S., Perumal B., and Yadav S., et al. (2020), Analysing and Predicting the purchases done on the day of Black Friday, International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2020.
[5] Xia Z., Xue S and Wu L. (2020), ForeXGBoost: passenger car sales prediction based on XGBoost, *Distributed and Parallel Databases*, 38(3), 713-738, doi: 10.1007/s10619-020-07294-y
[6] Biswas M., Shome A., and Islam M., et al. (2021), Predicting stock market price: A logical strategy using deep learning, *ISCAIE 2021 - IEEE 11th Symposium on Computer Applications and Industrial Electronics*, 218-223, doi: 10.1109/iscaie51753.2021.9431817
[7] Fildes R., Ma S., and Kolassa S. (2022), Retail forecasting: Research and practice, *International Journal of Forecasting*, 38(4), 1283-1318, doi: 10.1016/j.iforecast.2019.06.004
[8] Pan L. (2022), Price Prediction for BMW Based on Multifactorial Linear and Machine Learning Model, *ACM International Conference Proceeding Series*, 521-525, doi: 10.1145/3514262.3514347
[9] Guliyev H., and Mustafayev E. (2022), Predicting the changes in the WTI crude oil price dynamics using machine learning models, *Resources Policy*, 77, doi: 10.1016/j.resourpol.2022.102664
[10] Ganapathy G., Srinivasan K., and Datta D., et al. (2022), Rainfall Forecasting Using Machine Learning Algorithms for Localized Events, *Computers, Materials and Continua*, 71(2), 6333-6350, doi: 10.32604/cmc.2022.023254
[11] Mitra A., Jain A. and Kishore A., et al. (2022), A Comparative Study of Demand Forecasting Models for a Multi-Channel Retail Company: A Novel Hybrid Machine Learning Approach, Operations Research Forum, 3(4), doi: 10.1007/s43069-022-00166-4
[12] Akanksha A., Yadav D., and Jaiswal D., et al. (2022), Store-sales Forecasting Model to Determine Inventory Stock Levels using Machine Learning, 5th International Conference on Inventive Computation Technologies, ICICT 2022 - Proceedings, 339-344, doi: 10.1109/icict54344.2022.9850468