G-Tech : Jurnal Teknologi Terapan
Vol. 9, No. 3, July 2025, p. 1296-1305
E-ISSN: 2623-064X | P-ISSN: 2580-8737

Analysis of Selling Price Determination With Gradient Boosting Algorithm in Traditional Market Stores

Bagas Dwi Novianto[1]✉, **Kusrini**[2]

[1,2] Informatics, Computer Science Faculty, Universitas Amikom Yogyakarta, Indonesia

✉ **Corresponding Author :** Bagas Dwi Novianto (e-mail: bagasdnov@students.amikom.ac.id)

Article Information

Article History

Received : June 04, 2025
Revised : June 20, 2025
Published : July 10, 2025

***Keywords*:**

Gradient Boosting;
XGBoost;
Selling price;
SHAP;
Feature Importance.

ABSTRACT

Traditional market traders often face difficulties in determining optimal selling prices amid competition from modern retailers. This study aims to apply and compare Gradient Boosting and XGBoost algorithms to develop a selling price prediction model for traditional market stores. The research utilizes two datasets : a large-scale dataset from annual sales data and a small-scale dataset from one month of sales. Model training involves hyperparameter optimization using GridSearchCV and evaluation through metrics such as RMSE, MAE, R^2, and MAPE. Additionally, feature importance and SHAP analyses were conducted to interpret model behavior. The results demonstrate that both models performed well, with R^2 values nearing 1.0 and MAPE below 2%. Gradient Boosting outperformed XGBoost on the large dataset, while XGBoost showed better accuracy on the small dataset. These findings highlight the potential of machine learning in supporting data-driven pricing strategies for traditional markets.

INTRODUCTION

Traditional markets play a crucial role in the economy, particularly for local communities, by providing essential goods at affordable prices and enhancing community income and walfare through trading activities between merchants and buyers (Syukria, 2023). However, amid growing competition from modern retail stores and e-commerce platforms, traditional market vendors face significant challenges in determining appropriate selling prices to maintain business sustainability. The development of modern markets has a significant impact on traditional markets, especially for traders who face tight competition, declining turnover, and changes in consumer preferences. This condition encourages traders to adapt through updating business systems and increasing competitiveness(Wibowo et al., 2022).

Determining the selling price is very important in facing modern market competition and e-commerce (Sampe et al., 2023). Prices are influenced by various factors such as promotion, product and service quality, capital price, demand, availability of goods, season, and competitor prices. In traditional markets, service quality plays a major role in maintaining consumer loyalty (Permatasari et al., 2022). Due to the complexity of price determinants, analytical approaches such as the Gradient Boosting and XGBoost algorithms are needed to produce accurate predictions and identify the most influential factors in pricing.

Difficulty in setting the right selling price is a common problem in traditional markets in Indonesia. A study at Cakke Traditional Market shows that traders often use the going rate pricing method without considering operational costs and demand, which causes price instability

This is an open access article under the CC BY license

(Rusman Saleh et al., 2022). The lack of a data-based pricing system is an obstacle to efficiency. Therefore, the use of machine learning algorithms is a strategic solution to help traders set prices objectively, accurately, and responsively to market dynamics.

The Gradient Boosting model is an effective machine learning method in building predictive models through an additive process and is able to analyze large, complex, and unstructured data (Maulita et al., 2024) This method combines weak models gradually to improve prediction accuracy. Meanwhile, XGBoost is a development of Gradient Boosting which is equipped with two types of regularization, namely L1 (Lasso) and L2 (Ridge), which make the model more robust against overfitting and able to handle large numbers of variables efficiently (Razavizadeh et al., 2023).Research shows that XGBoost excels in terms of speed, accuracy, and robustness, and is often used in price, time, and cost estimation in various sectors (X. Chen et al., 2023). Therefore, this study uses both algorithms because of their ability to handle complex datasets and provide reliable prediction results in various analysis contexts.

The selection of the Gradient Boosting and XGBoost algorithms in this study is based on their ability to handle complex and non-linear data, such as the dynamics of selling prices in traditional markets. Both algorithms have proven to be superior to other methods such as linear regression which is less able to capture variable interactions, and SVM and KNN which are less efficient for large datasets (Anandhi & Nathiya, 2023). XGBoost, as a development of Gradient Boosting, has additional advantages in the form of double regularization (L1 and L2) to reduce overfitting, high computational efficiency, and feature importance features that can help traders understand the main factors in pricing (Kamalov & Leung, 2020). Although algorithms such as LightGBM and CatBoost offer high speed and efficiency on an industrial scale, their limitations such as the need for sensitive parameter tuning and preprocessing prerequisites make them less suitable for application in traditional markets with limited resources. In this context, selling price determination is a strategic decision that considers various factors such as capital prices, market demand, profit margins, and competitive conditions. With a data-driven approach such as Gradient Boosting, the pricing process can be done more objectively, adaptively, and in accordance with traditional market dynamics.

Gradient Boosting is a supervised learning technique that builds a model incrementally by combining predictions from a series of weak models, usually decision trees, through a process of minimizing a loss function (Elsa Suryana & Warsito, 2021). The process begins by initializing the model using a constant value, usually the average of the target. Next, the difference between the predicted value and the actual value (residual error) is calculated, then a new decision tree is built to predict the residual. The model is then updated by adding predictions from the new scaled tree. This process is repeated until the maximum number of models is reached or when the error reaches a minimum point. Gradient Boosting is very effective in handling complex data, but requires adjustments to parameters such as learning rate and tree depth for optimal performance(Jasman et al., 2022).

XGBoost is an optimized implementation of Gradient Boosting introduced by Tianqi Chen in 2016. Its main features include the use of L1/L2 regularization to reduce overfitting, parallel processing for fast training, and the ability to handle missing values and categorical variables without the need for preprocessing (T. Chen & Guestrin, 2016). In addition, XGBoost is designed for efficiency on large datasets (over 1 million rows) through technologies such as weighted quantile sketch and sparse-aware algorithms. XGBoost can achieve speeds up to 10 times faster than other boosting implementations with equivalent accuracy, as reflected in credit analysis with an AUC reaching 0.918 after SMOTE optimization (Saputra et al., 2024). Despite significant differences in the computational optimization aspects and regularization features, both algorithms remain popular for classification and regression tasks due to their flexibility in handling complex data patterns.

RESEARCH METHODS

This study uses an action research method that aims to evaluate the application of the Gradient Boosting and Extreme Gradient Boosting (XGBoost) algorithms to determine optimal selling prices in traditional markets. The study was conducted in three stores located in the market environment, with one store as the main focus and two other stores used as competitors.

Data And Sample

This study uses a dataset of goods prices in the last one year as a large dataset and one month as a small dataset that will be used for training and testing using the gradient boosting and XGBoost algorithms. This dataset includes market demand, availability of goods, and seasons. Market demand consists of three categories such as high, medium, and low. Availability of goods consists of two categories, namely easy and rare. Season consists of two categories, namely weekdays and holidays. Each data record is labeled with specific variables to support model training, including product (V1_encoded), date (V2), purchase price (V3), competitor 1 price (V4), competitor 2 (V5), market demand (V6), availability of goods (V7), season (V8), purchase volume (V9), selling price (V10), margin (V11), and margin percentage (V12). These labeled features serve as the foundation for building predictive models and identifying key factors influencing selling price determination in traditional market settings.

Data Collection Technique

This dataset was obtained by systematically interviewing shop owners, observing through direct observation of existing problems, and documenting by collecting historical sales data obtained directly from shop owners.

Data Analysis Method

The data analysis method in this study involves the application of conventional gradient boosting algorithms and XGBoost to determine the selling price in traditional markets. The analysis includes data pre-processing, application of both models with optimal parameters, and evaluation of model performance(Ardana, 2023). In addition, this study combines SHAP (SHapley Additive Explanations) analysis and feature importance evaluation to interpret the model. SHAP analysis helps explain how each feature affects the model's prediction by calculating the Shapley value, which measures the contribution of each input variable to the output, providing clear insight into the factors that are most influential in determining price. Feature importance further helps in identifying which variables have the greatest influence on the model's predictive performance, increasing interpretability and confidence in the results(Li, 2022).

Model Optimization

At this stage, the Gradient Boosting and XGBoost models are optimized through hyperparameter tuning using the Grid Search method. This process involves testing various parameter combinations periodically to find the best configuration that produces the highest model performance (Probst & Bischl, 2019). By selecting the most appropriate hyperparameters, the accuracy and generalization ability of the predictive model can be significantly improved.

Performance Evaluation

The performance of both models will be evaluated using several standard evaluation metrics, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Mean Absolute Percentage Error (MAPE). These metrics provide a comprehensive assessment of the model's predictive accuracy, consistency, and ability to generalize across different input data (Chicco et al., 2021). The use of multiple evaluation criteria ensures a more reliable comparison and helps identify the most effective model for estimating optimal selling prices.

RESULTS AND DISCUSSION

Model testing was conducted using the Python programming language with the pandas, numpy, matlotlib, seaborn, and shap libraries which are run using jupyter notebook in visual studio code. The dataset used consists of a large dataset and a small dataset used to test the gradient boosting and XGBoost models. The GridShearchCV method is used to find the best combination of parameters for both models (Jamiluddin et al., 2024). GridSearchCV uses a grid search method to evaluate various combinations of hyperparameters in gradient boosting and XGBoost to produce the best model (Nugraha & Sasongko, 2022). The combination of hyperparameters for both models and both datasets is shown in Table 1.

Tabel 1. Optimal hyperparameter combinations for both models and both dataset

No	Model	Hyperparameter Combination
1	Gradient Boosting	n_estimators = [50, 100, 200] learning_rate = [0.01, 0.05, 0.1, 0.2] max_depth = [3, 5, 7]
2	XGBoost	n_estimators = [100, 200] learning_rate = [0.05, 0.1] max_depth = [3, 5] reg_alpha = [0, 1] reg_lambda = [1, 2]

To obtain the best predictive performance of each algorithm, this study applies a hyperparameter search process using a grid search technique with three-fold cross validation. In the gradient boosting model, the combined parameters include n_estimators of 50, 100, and 200; learning_rate of 0.01, 0.05, 0.1, and 0.2; and max_depth at depths of 3.5, and 7. The best parameter search process is carried out with an evaluation score based on negative Root Mean Squared Error (RMSE), which reflects how close the model prediction is to the actual value in the original unit. Meanwhile, in the XGBoost model, grid search is performed on a wider combination of parameters. The parameters tested include n_estimators of 100 and 200, learning_rate of 0.05 and 0.1, max_depth at levels 3 and 5, and two regularization parameters, namely reg_alpha (0 and 1) and reg_lambda (1 and 2). The use of this regularization aims to avoid overfitting, especially when the model is applied to datasets with high complexity.

Tabel 2. Evaluation results of both algorithm using large datasets

No	Model	RMSE	MAE	R2	MAPE
1	Gradient Boosting	1222.6539	681.5817	0.9999	1.5869
2	XGBoost	1286.7560	779.9284	0.9999	1.9732

Tabel 3. Evaluation results of both algorithm using small datasets

No	Model	RMSE	MAE	R2	MAPE
1	Gradient Boosting	1008.5530	325.8905	0.9999	0.9844
2	XGBoost	673.4626	180.2140	0.9999	1.0038

The results of testing the gradient boosting and XGBoost models on two dataset scenarios, namely large datasets and small datasets, are presented in table 2 and table 3. The evaluation was carried out using four main metrics, namely Root Mean Squared Error (RMSE), Mean Absolute Error (Mae), Coefficient of Determination (R2), and Mean Absolute Percentage Error (MAPE). Based on table 2, the gradient boosting model shows better results compared to XGBoost in the

context of large datasets. Gradient Boosting recorded an RMSE value of 1222.65, MAE 681.58, and MAPE 1.5869%, with an R2 value of 0.9999 indicating a very high model fit to the actual data. Meanwhile, XGBoost produced a higher RMSE (1286.75) and a larger MAE (779.93), and a MAPE of 1.9732%. Although both models show very high R2 values, gradient boosting is proven to be superior in terms of precision and stability of predictions on large datasets. In contrast to the test with a small dataset in Table 3, the XGBoost model shows much better performance. XGBoost produces RMSE values of 673.46, MAE 180.21, and MAPE 1.0038, all of which are lower than gradient boosting which recorded RMSE of 1008.55, MAE 325.89, and MAPE 0.9844%. The R2 values for both models remain the same at 0.9999, but the lower prediction error of XGBoost shows its ability to handle small datasets more efficiently and accurately. This difference in results shows that the performance of the algorithm is highly dependent on the scale and complexity of the data. XGBoost excels on small datasets thanks to its ability to perform regularization through the reg_alpha and reg_lambda parameters, which are effective in preventing overfitting. On the other hand, gradient boosting is more effective on large datasets because it is able to capture complex patterns gradually through a more stable sequential boosting approach.

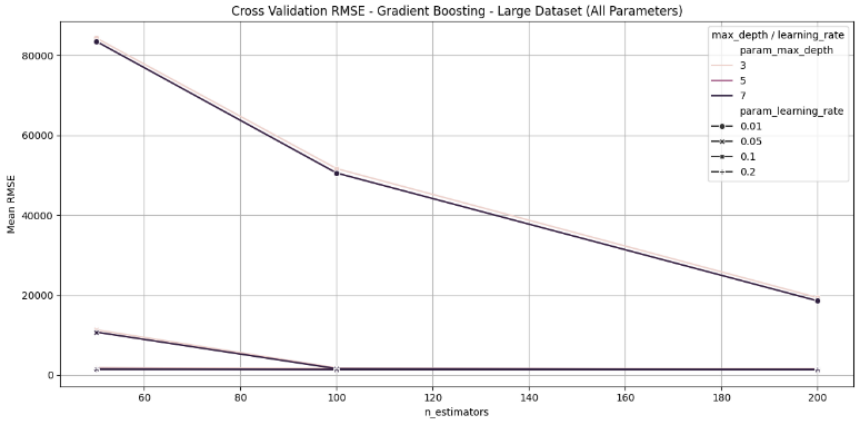

Figure 1. Cross validation RMSE gradient boosting of large dataset

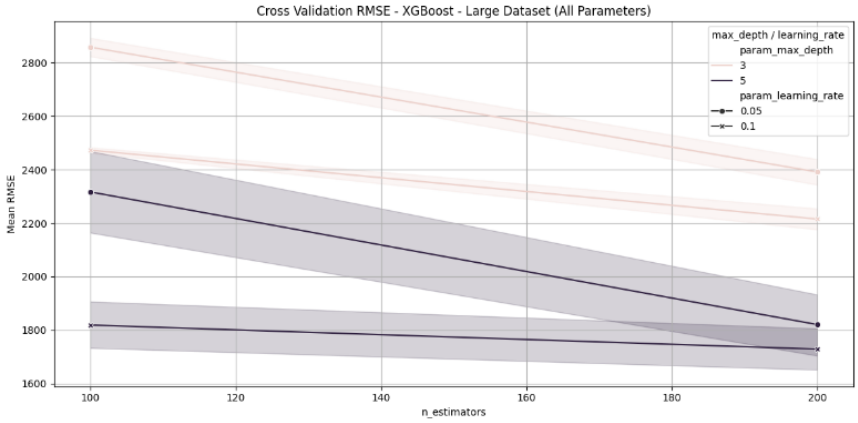

Figure 2. Cross validation RMSE XGBoost of large dataset

Model performance testing through cross-validation with a combination of parameters shows that both gradient boosting and XGBoost experience a decrease in the Root Mean Square Error (RMSE) value as the number of trees (n_estimators) increases. In the gradient boosting model, the graph shows that configurations with a larger max_depth and a smaller learning_rate tend to produce lower RMSE values, especially when n_estimators reaches 200. This indicates that the gradient boosting model requires a stepwise approach and sufficient tree depth to effectively handle the complexity of large datasets, and is more sensitive to variations in learning_rate. Meanwhile, the XGBoost model shows more consistent and stable

performance. Although variations in the learning_rate and max_depth parameters also affect the results, the RMSE values produced by XGBoost are generally lower than gradient boosting. The best combination is found at max_depth = 5, learning_rate = 0.1, and n_estimators = 200, which provides a balance between accuracy and efficiency. Compared to gradient boosting, XGBoost appears to be more robust to parameter variations and produces smoother graphs, indicating model stability.

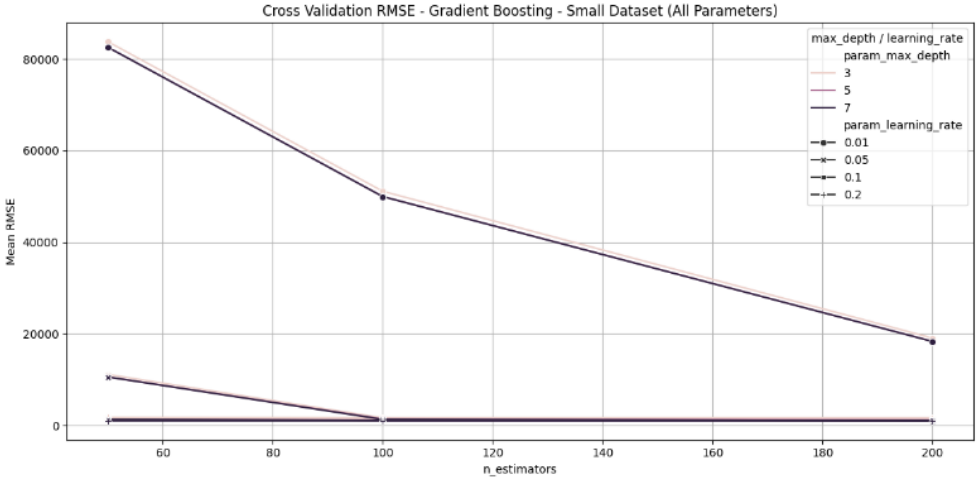

Figure 3. Cross validation RMSE gradient boosting of small dataset

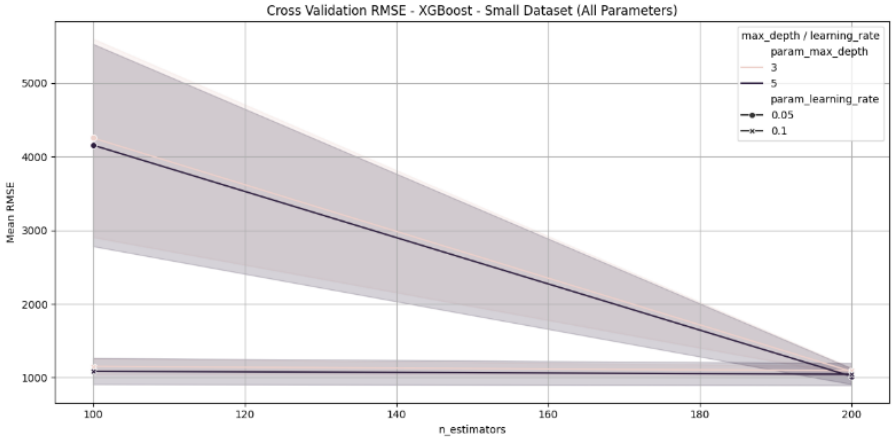

Figure 4. Cross validation RMSE XGBoost of small dataset

Model performance testing using cross-validation on a small dataset shows that increasing the number of n_estimators consistently decreases the Root Mean Squared Error (RMSE) value in both algorithms, both gradient boosting and XGBoost. The gradient boosting evaluation graph shows that all combinations of max_depth (3,5,7) and learning_rate (0.01, 0.05, 0.1, 0.2) parameters experience a decrease in RMSE when n_estimators increase from 50 to 200. This indicates that even though the amount of data is limited, the gradient boosting model still benefits from increasing the number of trees, especially when used with a small learning_rate. The best combination is found at n_estimators = 200, max_depth = 7, and learning_rate = 0.1, which shows more accurate prediction performance. Meanwhile, in the XGBoost model, the graph shows a sharper and more stable decrease in RMSE across all parameter configurations. The variation of learning_rate (0.05 and 0.1) and max_depth (3 and 5) shows that the combination of learning_rate = 0.1 and max_depth = 5 produces the lowest RMSE value overall. XGBoost performance is superior on small datasets compared to gradient boosting, both in terms of model stability and prediction error rate. This is supported by XGBoost ability to automatically adjust model complexity through regularization (reg_alpha, reg_lambda) which prevents overfitting.

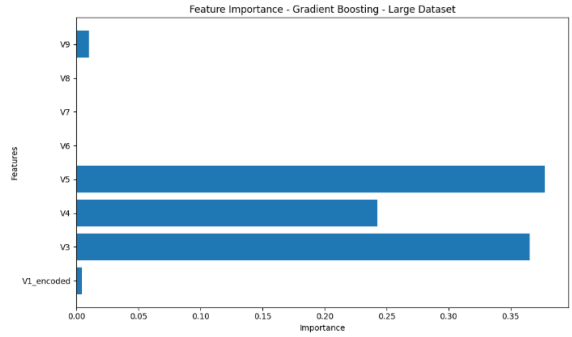

Figure 5. Feature Importance Gradient Boosting Large Dataset

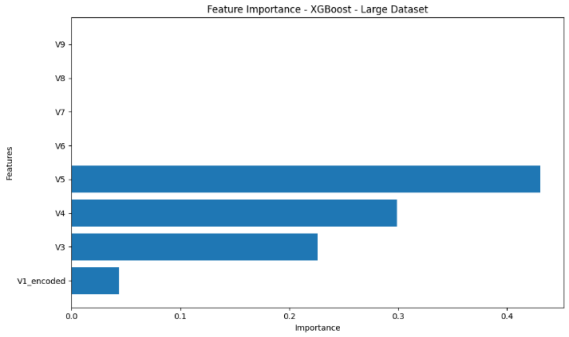

Figure 6. Feature Importance XGBoost Large Dataset

Feature importance analysis of gradient boosting and XGBoost on large datasets provides a deep understanding of the contribution of each variable to the prediction of selling prices. In the gradient boosting model, the most influential features are V5, V3, and V4 consisting of competitor 2, purchase price, competitor 1, each of which has the highest score compared to other features. The V5 or competitor 2 feature is recorded as having the largest contribution with a weight of , approaching 0.38, followed by V3 or purchase price and V4 or competitor 1, each of which has a score above 0.24. In contrast, other features such as V1_encoded, V6, V7, V8, and V9 show a very low level of influence, almost approaching zero. This shows that only a small part of the features are truly relevant in the process of forming a prediction model in the gradient boosting algorithm. Almost similar results were also found in the XGBoost model, where the V5 or competitor 2 feature also dominates with the highest score of 0.43, followed by V4 or competitor 1 and V3 or purchase price. However, a striking difference is seen from the sharper and more explicit weights in XGBoost compared to gradient boosting. XGBoost appears to provide a more focused portion of weights on the main features and more clearly distinguishes the importance between features.

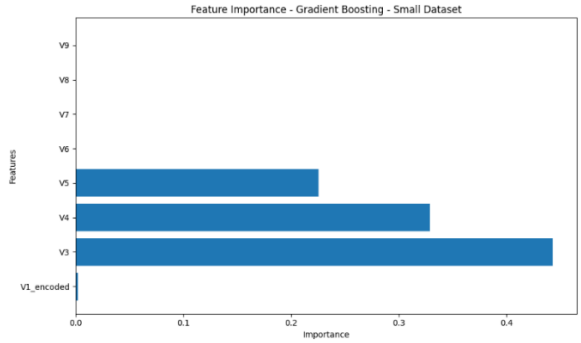

Figure 7. Feature Importance Gradient Boosting Small Dataset

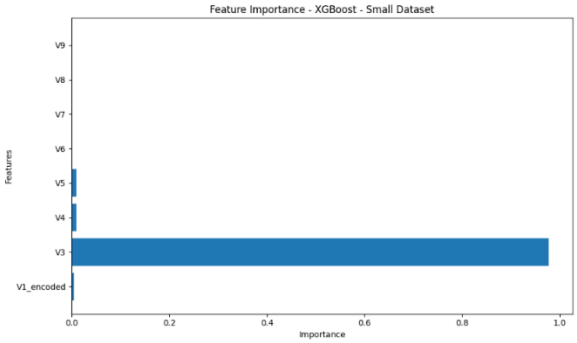

Figure 8. Feature Importance XGBoost Small Dataset

Feature importance analysis on small datasets shows quite striking differences between gradient boosting and XGBoost models in how they utilize features. In the gradient boosting model, features V3, V4, and V5 consisting of purchase price, competitor 1, competitor 2 have a significant contribution to the prediction of selling price, with V3 as the most dominant feature. The importance value for V3 or purchase price reaches 0.45, followed by V4 or competitor 1 around 0.33 and V5 or competitor 2 around 0.23, while other features such as V1_encoded, V6, to V9 have very small contributions. This shows that even though the amount of data is limited, the gradient boosting model is still able to distribute attention to several main features in a balanced manner. On the other hand, the feature importance results from the XGBoost model show a very large concentration of weight only on feature V3 or purchase price. The importance value of V3 or purchase price is close to 1.0, while other features have importance values that are almost zero. This pattern shows that the XGBoost model on small datasets performs feature selection very aggressively and tends to rely on only one main feature in forming decision trees. Although this strategy can improve accuracy on small data with low noise, it also indicates the risk of overfitting if the dominant feature is not representative enough.

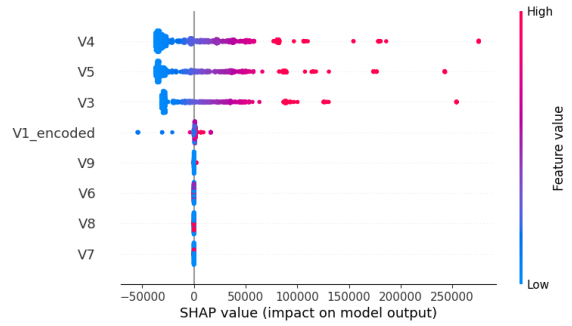

Figure 9. SHAP value large dataset

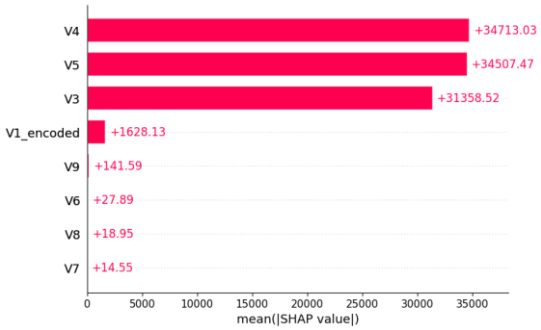

Figure 10. Mean SHAP value large dataset

The results of the SHAP analysis show that features V4, V5, and V3 have the greatest influence on the prediction of selling prices. These three features provide the greatest contribution to the model output, both on average (Mean SHAP value) and in the distribution of predicted values (beeswarm plot). High feature values tend to increase the prediction of selling prices, while low values decrease them. Meanwhile, other features such as V1_encoded, V6, to V9 have very little influence. This finding confirms that the model focuses its decisions on the main variables, and can be used as a basis for focusing pricing strategies in traditional markets.

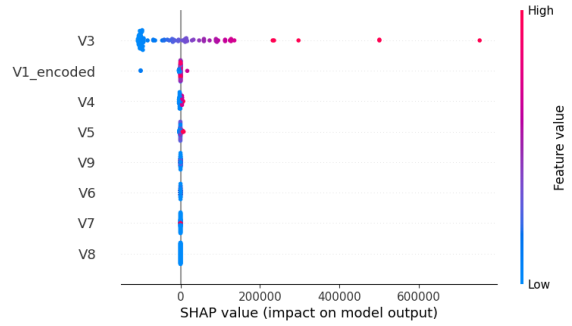

Figure 11. SHAP value small dataset

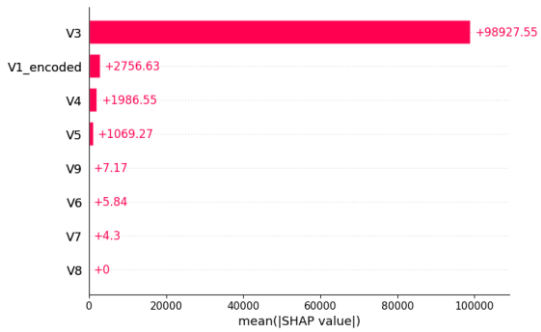

Figure 12. Mean SHAP value small dataset

SHAP analysis in Figure 11 and Figure 12 shows that feature V3 or purchase price has a dominant influence on the prediction of selling price, with an average SHAP value much higher than other features. followed by V1_encoded, V4, and V5, but their contributions are relatively small. Other features such as V6, V7, V8, and V9 contribute almost nothing. Beeswarm plot shows that high values in V3 significantly increase price prediction. This finding indicates that in small datasets, the model is highly dependent on one main feature, so interpretation and validation of this feature are very important.

The results of testing the gradient boosting and XGBoost models on two dataset scenarios, namely large datasets and small datasets, are presented in table 2 and table 3. The evaluation was carried out using four main matrics, namely Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), coefficient of Determination (R^2), and Mean Absolute Percentage Error (MAPE). Based on table 2, the gradient boosting model shows better results compared to XGBoost in the context of large datasets. Gradient Boosting recorded an RMSE value of 1222.65, MAE 681.58, and MAPE 1.5869%, with an R^2 value of 0.9999 indicating a very high model fit to the actual data. Meanwhile, XGBoost produced a higher RMSE 1286.75 and larger MAE 779.93, and MAPE of 1.9732%. Although both models show very high R^2 values, gradient boosting is proven to be superior in terms of precision and stability of predictions on large datasets. In contrast to the test with a small dataset in table 3, the XGBoost model shows much better performance. XGBoost produces RMSE values of 673.46, MAE 180.21, and MAPE 1.0038, all of which are lower than gradient boosting which recorded RMSE of 1008.55, MAE 325.89, and MAPE 0.9844%. The R^2 values for both models remain the same at 0.9999, but the lower predicition error of XGBoost shows its ability to handle small datasets more efficiently and accurately. This difference in results shows that the performance of the algorithm is highly dependent on the scale and complexity of the data. XGBoost excels on small datasets thanks to its ability to perform regularization through the reg_alpha abd reg_lambda parameters, which are effective in preventing overfitting. On the other hand, gradient boosting is more effective on large

datasets because it is able to capture complex patterns gradually through a more stable sequential boosting approach.

These findings are in line with previous studies. For instance, (Saputra et al., 2024) demonstrated that the XGBoost algorithm showed excellent predictive performance in credit risk analysis with high AUC scores, particularly when used on small and structured datasets. Similarly, (Razavizadeh et al., 2023) found that XGBoost outperformed traditional boosting algorithms in survival prediction of COVID-19 patients due to its robustness and regularization capability. On the other hand, (Maulita et al., 2024) showed that gradient boosting yielded more accurate results in flood prediction model using larger and more complex datasets, consistent with the results of this study on large datasets. Therefore, the consistency of these findings with existing literature reinforces the conclusion that algorithm selection must be aligned with dataset characteristics. This comparison also emphasizes the flexibility and reliability of both models, where XGBoost is preferable for small-scale applications, while gradient boosting offers stability and depth for larger datasets.

CONCLUSION

This study succesfully applied gradient boosting and XGBoost algorithms to develop predictive models for determining selling prices in traditional market stores. Both models demonstrated strong accuracy and adaptability to different data scales, with purchase price and competitor prices identified as the most influential factors based on feature importance and SHAP analyses. Beyond predictive performance, the practical implication of this research lies in its potential to help traditional traders implement data-driven pricing strategies. By integrating these models into a simple and accessible decision support system, traders can set prices more objectively, reduce reliance on intuition, and respond more effectively to market competition. It is recommended that traders begin adopting digital tools that incorporate these predictive models, supported by basic training on how to input data and interpret pricing suggestions. Such implementation could lead to more consistent pricing, improved profit margins, and increased competitiveness against modern retail formats. Additionally, forming small trader groups to share data and collectively monitor competitor prices could enhance the effectiveness of the system in daily practice.

ACKNOWLEDGEMENTS

The researcher would like to thank my academic supervisor, lecturer, and friends at Universitas Amikom Yogyakarta for their support and encouragement during research.

REFERENCES

Anandhi, P., & Nathiya, Dr. E. (2023). Application of linear regression with their advantages, disadvantages, assumption and limitations. *International Journal of Statistics and Applied Mathematics*, *8*(6), 133–137. https://doi.org/10.22271/maths.2023.v8.i6b.1463

Ardana, A. (2023). Performance Analysis of XGBoost Algorithm to Determine the Most Optimal Parameters and Features in Predicting Stock Price Movement. *Jurnal Informatika Dan Teknologi Informasi*, *20*(1), 91–102. https://doi.org/10.31515/telematika.v20i1.9329

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-August-2016*, 785–794. https://doi.org/10.1145/2939672.2939785

Chen, X., Hua, J., Yin, C., Chen, Z., Chen, Y., & Li, Z. (2023). A Precise Life Estimation Method for Retired Energy Storage Batteries Based on Energy Storage Batteries Attenuation Characteristics and XGBoost Algorithm. *IEEE Access*, *11*, 135968–135978. https://doi.org/10.1109/ACCESS.2023.3336813

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, 1–24. https://doi.org/10.7717/PEERJ-CS.623

Elsa Suryana, S., & Warsito, B. (2021). Penerapan Gradient Boosting dengan Hyperopt untuk Memprediksi Keberhasilan Telemarketing Bank. *JURNAL GAUSSIAN*, *10*, 617–623. https://doi.org/10.14710/j.gauss.10.4.617-623

Jamiluddin, F., Faisal, S., Lestari, S. A. P., & Fauzi, A. (2024). Implementasi Hyperparameter Tuning Grid Search CV Pada Prediksi Produksi Padi Menggunakan Algoritma Linear Regresi. *Journal of Information System Research (JOSH)*, *6*(1), 480–488. https://doi.org/10.47065/josh.v6i1.5930

Jasman, T. Z., Fadhlullah, M. A., Pratama, A. L., & Rismayani, R. (2022). Analisis Algoritma Gradient Boosting, Adaboost dan Catboost dalam Klasifikasi Kualitas Air. *Jurnal Teknik Informatika Dan Sistem Informasi*, *8*(2). https://doi.org/10.28932/jutisi.v8i2.4906

Kamalov, F., & Leung, H. H. (2020, November 3). Deep learning regularization in imbalanced data. *Proceedings of the 2020 IEEE International Conference on Communications, Computing, Cybersecurity, and Informatics, CCCI 2020*. https://doi.org/10.1109/CCCI49893.2020.9256674

Li, Z. (2022). Extracting spatial effects from machine learning model using local interpretation method: An example of SHAP and XGBoost. *Computers, Environment and Urban Systems*, *96*. https://doi.org/10.1016/j.compenvurbsys.2022.101845

Maulita, I., Widiawati, C. R. A., & Wahid, A. M. (2024). Analisis Komparatif Linear Regression, Random Forest, dan Gradient Boosting untuk Prediksi Banjir. *Jurnal Pendidikan Dan Teknologi Indonesia (JPTI)*. https://doi.org/DOI:10.52436/1.jpti.599

Nugraha, W., & Sasongko, A. (2022). Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search Hyperparameter Tuning on Classification Algorithm with Grid Search. *SISTEMASI: Jurnal Sistem Informasi*, *11*(2), 2540–9719. https://doi.org/10.32520/stmsi.v11i2.1750

Permatasari, E., Luthfiana, H., Pratama, N. A., & Ali, H. (2022). Faktor-faktor yang Mempengaruhi Pembelian Ulang: Promosi, Harga dan Produk. *Jurnal Ilmu Manajemen Terapan*, *3*(5). https://doi.org/10.31933/jimt.v3i5

Probst, P., & Bischl, B. (2019). Tunability: Importance of Hyperparameters of Machine Learning Algorithms. In *Journal of Machine Learning Research* (Vol. 20). https://doi.org/10.48550/arXiv.1802.09596

Razavizadeh, N. T., Salari, M., Jafari, M., Sabaghian, E., & Ghavami, V. (2023). Comparison of Two Methods, Gradient Boosting and Extreme Gradient Boosting to Pre-dict Survival in Covid-19 Data. In *J Biostat Epidemiol* (Vol. 9, Issue 3). https://doi.org/10.18502/jbe.v9i3.15450

Rusman Saleh, A., Hasan, M., & Supatminingsih, T. (2022). Analisis Faktor Penyebab Turunnya Harga Jual Beli Sayuran di Pasar Cakke Kecamatan Anggeraja Kabupaten Enrekang. *INTELEKTIVA : JURNAL EKONOMI, SOSIAL DAN HUMANIORA*. https://www.jurnalintelektiva.com/index.php/jurnal/article/view/741

Sampe, F., Ardianto, R., Yusuf, M., & Sultan Aji Muhammad Idris Samarinda, U. (2023). The Impact Of Brand Image And Prıce Onlıne Product Purchase Decısıons At Shopee. *Asian Journal of Management Entrepreneurship and Social Science*. https://ajmesc.com/index.php/ajmesc

Saputra, A. A., Sari, B. N., & Rozikin, C. (2024). Penerapan Algoritma Extreme Gradient Boosting (Xgboost) Untuk Analisis Risiko Kredit. *Jurnal Ilmiah Wahana Pendidikan*, *10*(7), 27–36. https://doi.org/10.5281/zenodo.10960080

Syukria, A. (2023). Potensi Pasar Tradisional dalam Peningkatan Ekonomi Masyarakat. *Jurnal Informatika Ekonomi Bisnis*, 306–311. https://doi.org/10.37034/infeb.v5i2.221

Wibowo, F., Khasanah, A. U., Fitroh, I., & Putra, S. (2022). Analisis Dampak Kehadiran Pasar Modern terhadap Kinerja Pemasaran Pasar Tradisional Berbasis Perspektif Pedagang dan Konsumen di Kabupaten Wonogiri. *Benefit: Jurnal Manajemen Dan Bisnis*, *7*(1), 53–65. https://doi.org/10.23917/benefit.v7i1.16057