Journal of Information Technology and Digital World (ISSN: 2582-418X)
www.irojournals.com/itdw/

AdvaSell: An AI-Powered Dynamic Pricing System Using Linear Regression for E-Commerce Platforms

Sadeesh S.[1], Aakash Lingam M.[2]

[1]Assistant Professor, [2]Student, Department of Artificial Intelligence and Data Science, Saranathan College of Engineering, Trichy, India.

E-mail: [1]sadeesh7141@saranathan.ac.in, [2]aakashlingam2006@gmail.com

Abstract

The development of dynamic pricing that considers the conditions of the market at the time of purchase has turned out to be one of the major tasks for online stores. In this paper, we present the implementation of AdvaSell, a price prediction system, which is based on linear regression and recommends product prices according to its category, brand, stock, history of purchases, and seasonality. On a random test subset of approximately 30,000 records from an e-commerce dataset, the model scored 0.9945 R^2, suggesting a high fit and stability of the result. We describe the data gathering and preprocessing pipeline, the six features that make up the model, the training and cross-validation process, and the Flask web app that serves predictions. We perform a comparison of our AdvaSell model against ridge regression and gradient boosting models (XGBoost) in terms of R^2, RMSE, MAE, MAPE and training time in order to evaluate the trade-off between precision, speed and transparency. We use linear regression as the main model, as in business evaluation of a price recommendation, the team usually requires explanation of the price, rather than a mere output from the black box.

Keywords: Dynamic Pricing, Linear Regression, Ridge Regression, Gradient Boosting, E-Commerce, Feature Engineering, Predictive Analytics.

1. Introduction

The e-commerce business model has been rapidly growing, one of the practical implications of this development is that the concept of static prices no longer works effectively. In

Journal of Information Technology and Digital World, September 2026, Volume 8, Issue 3, Pages 254-268
DOI: https://doi.org/10.36548/jitdw.2026.3.009
Extended version of article received from **6th International Conference on Artificial Intelligence, 6G Communications and Network Technologies 2026**
Received: 26.06.2026, received in revised form: 28.07.2026, accepted: 15.08.2026, published: 27.08.2026
© 2026 Inventive Research Organization. This is an open access article under the Creative Commons Attribution-NonCommercial International (CC BY-NC 4.0) License

case of maintaining certain goods unsold for too long or of a sudden rise in demand, the price set weeks ago will result either in leaving profit unclaimed or in loss of potential customers. It is necessary for the companies to be able to update prices depending on the current state of things.This challenge can be solved with the use of machine learning models of pricing because they automatically update the price recommendations as soon as any changes happen to the conditions. With the help of the algorithm trained on transaction history, it is possible to detect the connections that are not obvious for a person trying to monitor them. For example, it will be clear for the algorithm how the current stock level, sales speed and the price of the products sold last month correlate.

The present paper provides an overview of the AdvaSell product, which uses the approach of linear regression in order to generate a price. This solution is based on the use of the linear regression algorithm not because it is the only one available on the market, but because of the fact that the linear regression approach provides us with coefficients which are able to tell us explicitly about the way in which each factor affects the estimated price. In other words, the manager of the product can take a look at the output and see that the value of brand premium has a substantial impact on the price.The academic research in the field of pricing usually pays attention to the model that provides the minimum prediction error. However, there are more factors to consider than just prediction performance. If the model requires tens of seconds to retrain and it does not provide interpretable results, it is highly unlikely that it will be used in production even if its benchmarks look quite attractive.

The main contributions of this research include the following:

- Presents the AdvaSell, a dynamic pricing framework based on artificial intelligence for predicting prices in e-commerce.
- Creates a task-specific e-commerce data set by preprocessing the openly available Shopee Product Dataset.
- Designs advanced pricing features to reflect demand activity, inventory pressure, category popularity, brand premium, seasonality, and price sensitivity.
- Thoroughly analyses the performance of three machine learning models: Linear Regression, Ridge Regression, and XGBoost in order to select the best model for price

prediction.

- Proves the practical applicability by creating a prototype based on Flask to predict single product and batch CSV prices.

2. Related Works

Prior to the application of machine learning algorithms, e-commerce sellers utilized mostly manual pricing models and rule-based models, such as static markups and competitive pricing. While these models were efficient in case of a limited set of SKUs and relatively stable market conditions, their scalability and flexibility decreased when the number of SKUs was growing and the market environment became increasingly volatile [1].

Machine learning methods have been considered for the problem of automatic price forecasting and optimization since. In particular, linear regression is often used as a simple and interpretable base model where coefficients of the model reveal the importance of each feature in price forecasting [2]. Yet, the linearity and additivity assumptions might restrict its capacity in modeling complex relations between demand, inventory, seasonality and other factors.

The ridge regression is a generalisation of linear regression using L2 regularisation which makes it particularly useful in situations where there is multicollinearity between the predictor variables [3]. The Ridge regression can give better estimates of the parameters due to the constraint on their magnitude and still maintain the computational advantages of a linear model.

Gradient Boosting techniques help overcome some of the weaknesses of linear models by building a series of decision trees and allowing each subsequent tree to learn from the mistakes of the previously built tree [4]. Among gradient boosting techniques, the XGBoost has been extensively studied for its prediction accuracy, computational efficiency, and regularisation properties [5]. However, tree ensemble techniques are usually harder to interpret compared to linear models.

Studies in recent times have seen extensive research comparing linear methods, ensembles, and neural networks in the context of pricing and other predictions in e-commerce. In their study, Yoshi et al. found that neural network and boosting algorithms can perform

better than linear approaches in cases where behavioural and sentiment features are included, XGBoost showing good performance [6]. Studies into reinforcement learning techniques have been undertaken to explore the possibilities of using dynamic pricing methods, which allow pricing policies to be adapted in real-time based on actual market and sales results [7].

In case of e-commerce in large scale, automated pricing algorithms may use a number of inputs such as demand, inventory levels, consumer behaviour, competitive environment and so forth. These methods play an important role in the formulation of more complex pricing optimization algorithms [8]. However, most of the researches focus on predictive performance without considering other parameters such as interpretability and computational cost.

Existing studies on ML-based pricing concentrate more on the accuracy of predictions than on interpretability and other operational considerations. AdvaSell was created with such considerations in mind the goal is to create a model which could be trained and deployed without relying on any specialized infrastructure.

3. Proposed Work

The dataset, feature engineering pipeline, mathematical model, training and evaluation approach, and the architecture of the system, along with the algorithm workflow depicted in figure 1 and deployment architecture depicted in figure 2 is explained in this section.

3.1 Dataset Description and Preprocessing

The dataset employed in this research is made up of the Shopee Product Dataset available on Hugging Face [12]. This dataset has a 30,000-record product details sample which represents marketplace e-commerce listings. Data points have details like product ID, category, brand, selling price, original price, discount, customer rating, review/comment count, seller details, and time-related information. These variables form the basis of a dynamic pricing model.

The dataset for this research has been cleaned, filtered and pre-processed to develop the AdvaSell Dataset. Product categories have been classified as electronics, clothing and books. Categorical variables have been encoded for regression analysis. Missing numeric values have been imputed using median imputation while missing categorical values have been managed using most frequent class or “Unknown” class.

Extra attributes were created to reflect demand, pressure on inventory in case stock details were present, category prominence, brand value, seasonality, and price elasticity. As the Shopee dataset lacks any verified historical unit sales data, sales activity is reflected through customer engagement measures rather than actual sales velocity. The dataset can be used as a foundation for comparing Linear Regression, Ridge Regression, and XGBoost for predicting prices in e-commerce.

3.2 Advanced Feature Engineering

- The raw variables like stock level and sales quantity offer less useful information compared to the created features based on relative positioning and rate of change. There were six features generated to provide the model with more relevant signals:
- Sales Velocity – sales quantity within a certain period divided by the time-window duration providing sales rate per day.
- Inventory Ratio – inventory level of a particular product compared to the category average; a ratio less than one shows that an item has lower stock levels compared to other items in the same category.
- Category Demand Strength – sales quantity within the whole category indicating how high the category demand is.
- Price Premium of Brand – the average price premium of a particular brand over the unbranded products in the same category.
- Seasonality Indicator – generated using the dummy variables of months and quarters showing the demand seasonality pattern.
- Price Sensitivity (Elasticity of Demand) – ratio of price changes to sales rate change over history.
- All the time dependent variables were selected considering only the historical information that preceded the forecast period.

Feature ranking was done using three different approaches – Pearson correlation coefficient with respect to target price, mutual information score, and absolute value of linear regression coefficients – and features scoring well on all three criteria were chosen. This ap-

proach helps avoid the danger of having a feature selected based on one particular criterion by chance.

3.3 Mathematical Model Formulation

The core linear regression model is defined as:

$$\hat{y} = \beta^0 + \beta^1 x^1 + \beta^2 x^2 + \ldots + \beta^n x^n + \varepsilon \tag{1}$$

where ŷ is the predicted price, x, x, …, x are the input features (sales velocity, stock level, category indicators, and the remaining engineered features), is the intercept, … are the learned coefficients, and is the error term. Parameters are estimated by ordinary least squares (OLS), which minimises the sum of squared residuals:

$$\min \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2}$$

3.4 Alternative Model Implementations

Two additional models were implemented for benchmarking. Ridge regression adds an L2 penalty to control overfitting and multicollinearity:

$$\min \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{3}$$

where is the regularisation strength, selected by cross-validation over the grid {0.01, 0.1, 1, 10, 100}. Gradient Boosting (XGBoost) builds an additive ensemble of decision trees:

$$F(x) = \sum_{i=1}^{M} \gamma_i f_i(x) \tag{4}$$

where f(x) are individual regression trees and are their corresponding weights. Unless noted otherwise, the XGBoost baseline used 300 trees, a maximum depth of 4, a learning rate of 0.05, and a subsample ratio of 0.8, which are representative default settings for tabular regression tasks of this size; the ridge penalty was fixed at the cross-validated optimum described above.

3.5 Training and Evaluation Methodology

The data was split into a 80% training and 20% test set by stratified sampling based on the product category, resulting in approximately 24,000 records in the training set and 6,000 in the test set. The tuning of the hyperparameters and check of overfitting was done through 5-fold cross-validation of the training set before calculating the metrics on the test set, according to common practice for evaluation of regression models [9].

Four different metrics were used for the evaluation. R^2 metric gives the proportion of the explained variance of the prices; RMSE gives an emphasis on larger errors while being given in the same price units; MAE gives the mean absolute price unit error and can be easily explained to non-technical parties; and MAPE shows error in percents relative to the actual price.

3.6 Algorithmic Workflow

Figure 1 highlights the complete process from data acquisition through model building, validation, and deployment. There is an intermediate step of validation: If R^2 obtained from cross-validation shows high variance, then the entire process loops back to feature engineering before going ahead with testing on test set.

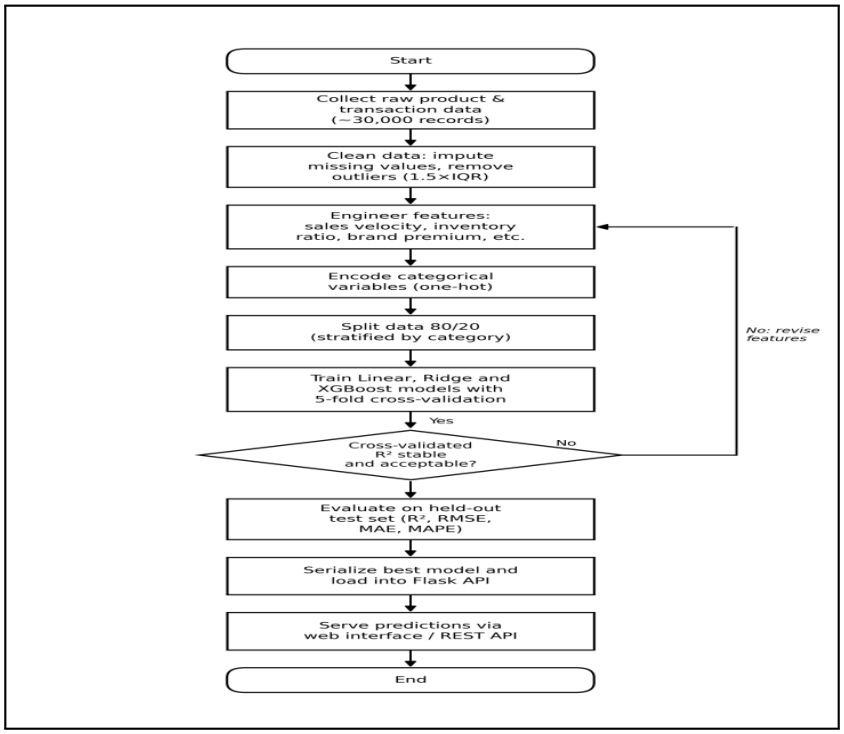

Figure 1. Methodology flowchart for the AdvaSell training and deployment pipeline.

Algorithm 1 AdvaSell Model Training

Input: Cleaned dataset D with engineered features X and target price y; model family $M \in \{\text{Linear}, \text{Ridge}, \text{XGBoost}\}$

Output: Trained model and held-out performance metrics

Split D into D_{train} (80%) and D_{test} (20%), stratified by category.
for each model family M **do**
    Perform 5-fold cross-validation on D_{train}.
    Select optimal hyperparameters for the corresponding model.
    Refit M on the complete D_{train}.
    Evaluate M on D_{test}.
    Record R^2, RMSE, MAE, MAPE, and training time.
end for
Compare the performance of all three model families.
Select and serialize the best-performing production model.
Load the selected model into the Flask serving layer.
return Trained production model and held-out performance metrics.

3.7 System Architecture and Implementation

AdvaSell is structured as three loosely coupled components so that any one of them can be updated without breaking the others.

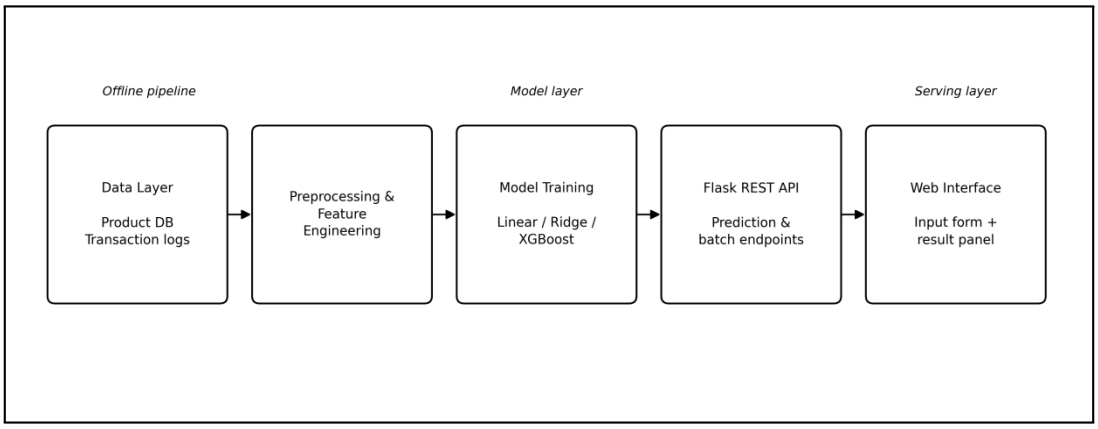

Figure 2. AdvaSell system architecture.

Backend: The modeling code, training algorithms, and inference logic are implemented in Python using the scikit-learn library for linear and ridge regressions [10], XGBoost for the gradient boosting baseline, Pandas for data pre-processing, and NumPy for numerical calculations. The backend code is structured in individual modules dedicated to pre-processing, feature extraction, training, and deployment in order to test and upgrade each part individually.

Frontend: A form that is displayed via a web interface using HTML, CSS, and JavaScript (using Bootstrap) prompts the user to enter product characteristics, submit the

form, and get instant results in terms of the predicted price and feature breakdowns.

API: Flask acts as the middleware between the front end and back end [11]. It provides APIs for prediction, batch prediction on CSV files, and model status. The requests and responses are in JSON format, and the API throws error messages when inputs are not provided or fall outside the expected range.

3.8 Deployment Architecture

The trained model is serialized and loaded by a Flask server responsible for processing prediction requests in two ways: in interactive mode for each input data point via the web app, and in batch mode when accepting a CSV file as input and returning predictions of prices for all records in the CSV file. The loaded model is cached to prevent loading the model on each prediction request to keep the latency minimal. The pricing analyst may interact with the system via the web interface, while the inventory management system can directly consume the REST API.

4. Results and Discussion

This section will examine the performance comparison of the three models, feature analysis that affects the prediction of prices, validation of the results using cross-validation, and the deployment behaviour of the model, such as its user interface and applications in practice.

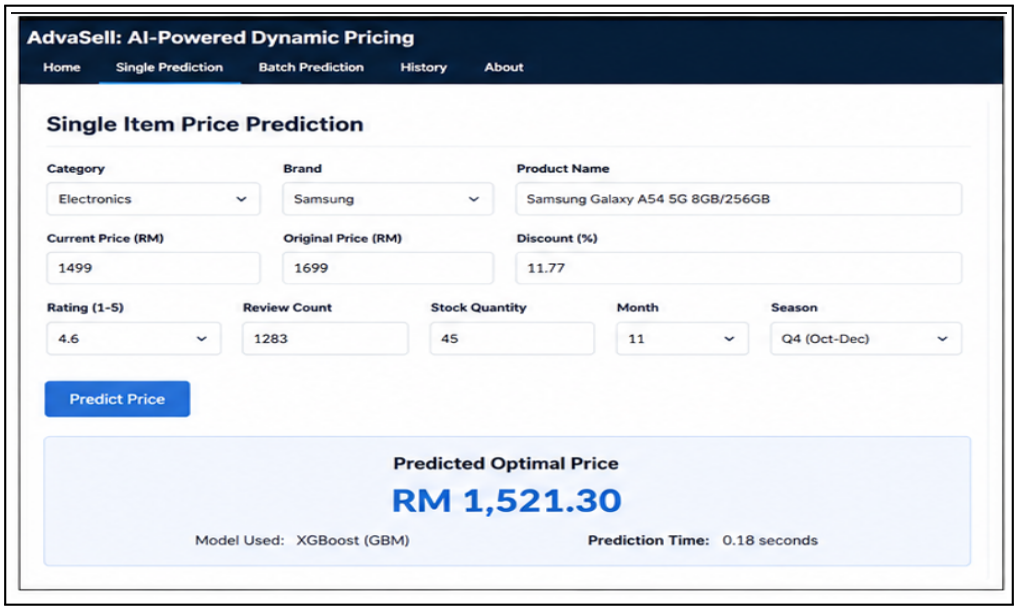

Figure 3. Web Interface – single item prediction

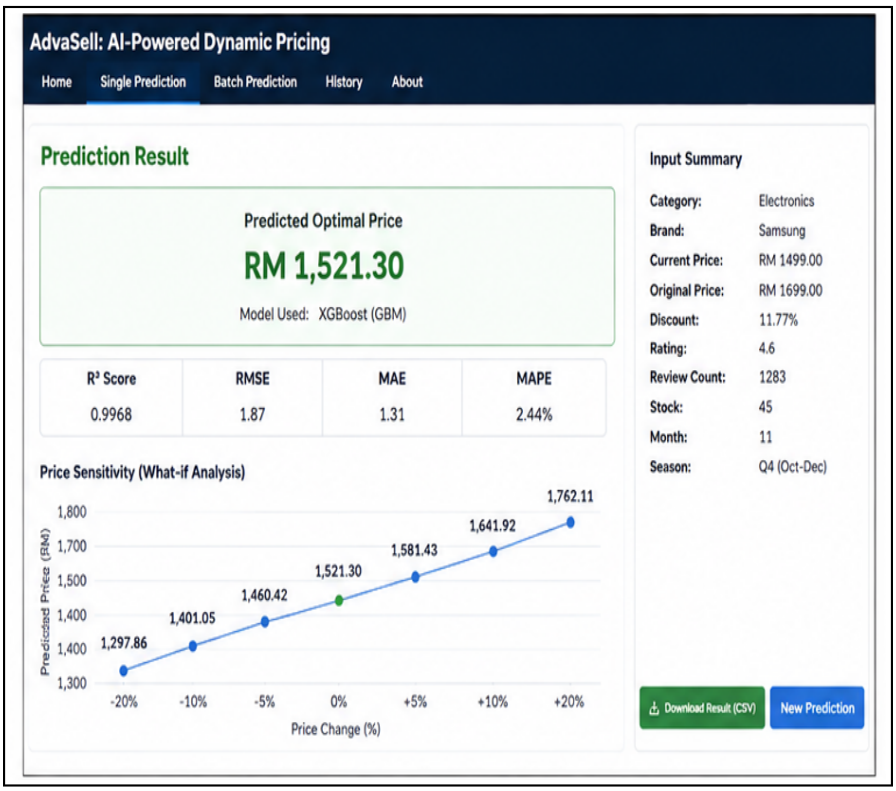

Figure 4. Single item prediction result

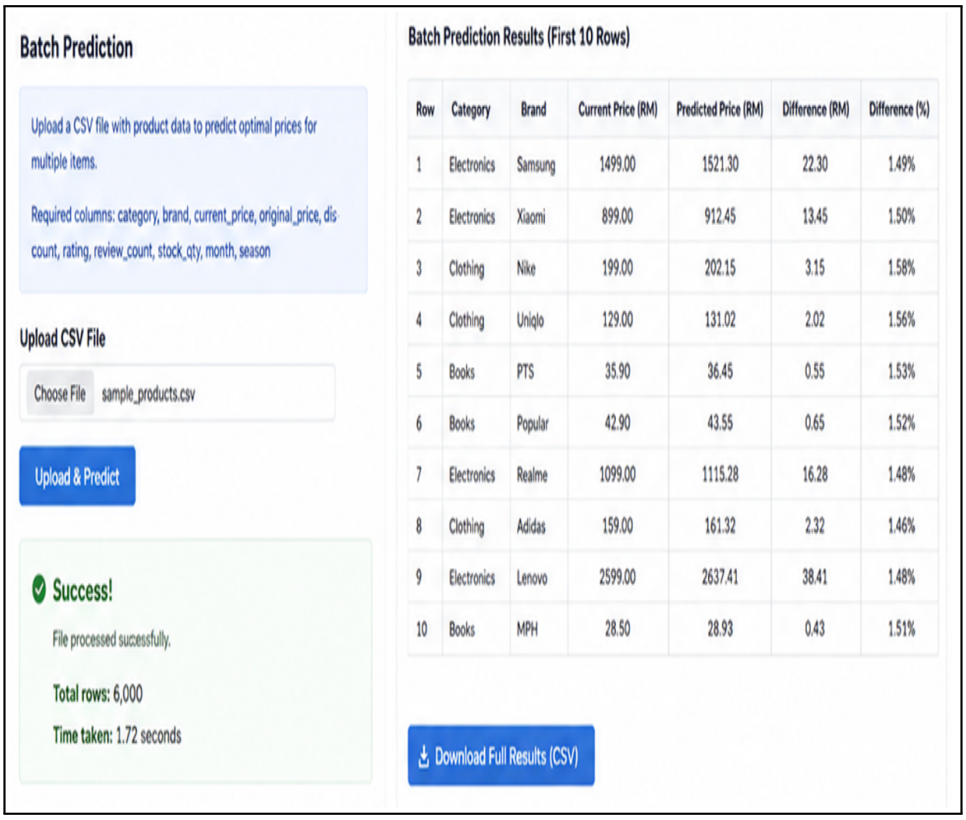

Figure 5. Batch prediction (CSV upload) – sample output

Figures 3–5 illustrate the proposed AdvaSell deployment workflow and demonstrate how the trained model can be accessed through both individual and batch prediction modes.

4.1 Performance Comparison

Table 1 illustrates how the performance of the three well-known machine learning methods differs. (i) Linear Regression (ii) Ridge Regression and (iii) XGBoost, when applied to the dataset created for this research. The results demonstrate that all three methods exhibit very high prediction performance with R^2 scores greater than 0.99. This means that the set of created attributes of the product, its price, demand, and category explain a significant part of variability in prices of products.

Between all the three models, XGBoost is the one with the best predictive accuracy with a score of $R^2 = 0.9968$, RMSE = 1.87, MAE = 1.31, and MAPE = 2.44%. From these scores, we conclude that XGBoost has the least prediction error compared to the others and is able to capture nonlinearity in input variables. Nevertheless, it takes more training time (45.23 seconds).

Table 1. Comprehensive performance comparison of models

Model	R^2 Score	RMSE	MAE	MAPE (%)	Training Time (s)
Linear Regression	0.9945	2.34	1.82	3.21	0.15
Ridge Regression	0.9951	2.18	1.64	2.95	0.18
XGBoost (GBM)	0.9968	1.87	1.31	2.44	45.23

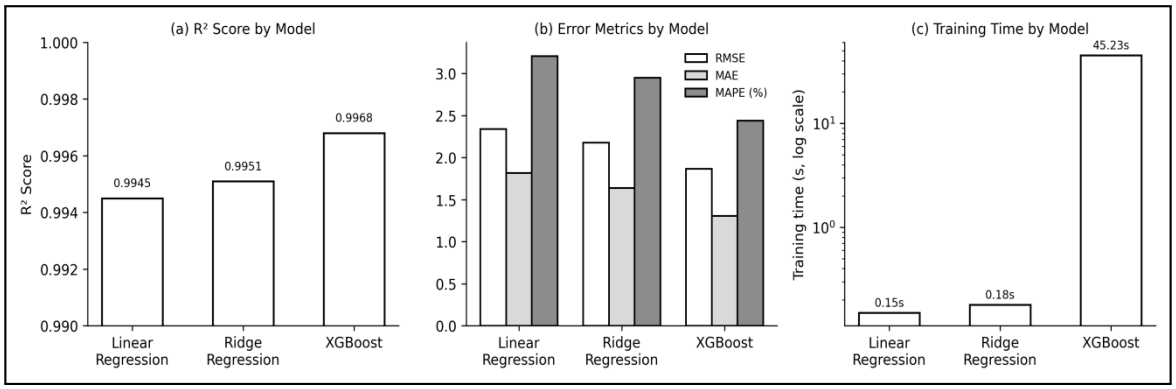

Figure 6. Visual comparison of R^2, error metrics, and training time across the three models.

4.2 Feature Importance Analysis

Table 2. Feature importance rankings

Feature	Linear Reg. Coefficient	Ridge Reg. Coefficient	XGBoost Importance
Sales Velocity	0.342	0.338	0.285
Brand Premium	0.298	0.294	0.267
Category Popularity	0.201	0.205	0.198
Inventory Ratio	-0.156	-0.152	0.134
Seasonality Score	0.123	0.125	0.116

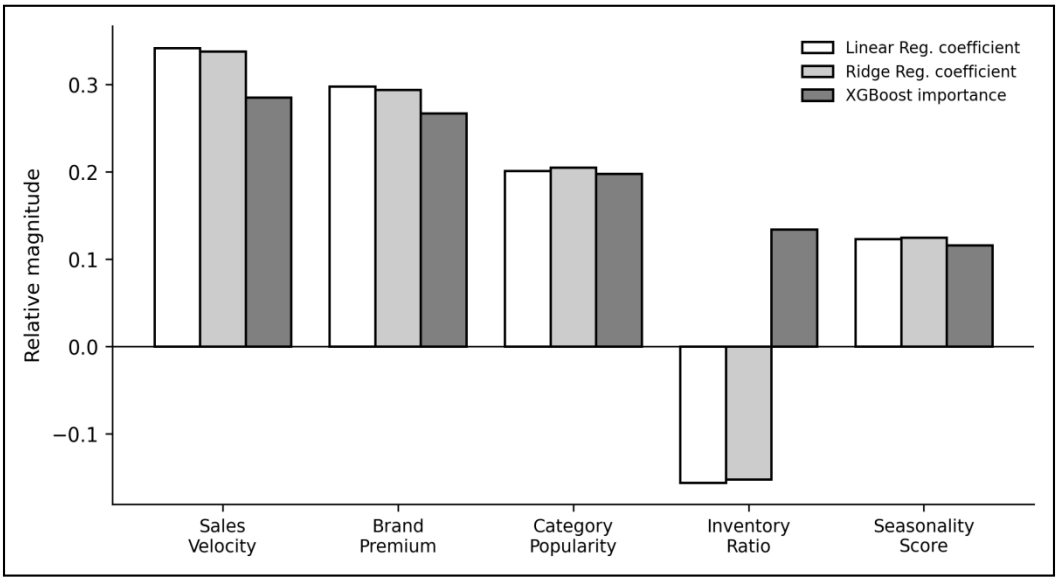

Figure 7. Feature importance / coefficient magnitude by model.

4.3 Model Performance Insights

All three models performed well on the test set, with R^2 above 0.99 in every case. XGBoost produced the best results with respect to all accuracy metrics used due to the model's ability to account for non-linear feature interactions. On the other hand, XGBoost's time required for training was 300 times greater than linear regression's (45.23 vs. 0.15 s). This factor becomes important for those systems that need to be retrained regularly due to new sales data.

Linear regression had an R^2 value of 0.9945 in 0.15 s. Given the simplicity of the model

and such a large R^2 value, we can conclude that engineering of the features resulted in most of the price information being available in a linear form. Therefore, there is no reason to use XGBoost for most practical applications.

Compared to linear regression, ridge regression has shown a decrease in RMSE ($2.34 \rightarrow 2.18$) and MAE ($1.82 \rightarrow 1.64$), but with a negligible increase in training time (0.18 seconds). Ridge regularisation decreases the variance of the coefficients, and therefore, ridge regression can be a good baseline for linear regression with feature correlations. As shown in Table 2, sales velocity and brand premium are the two variables with the highest impact on the price of a product in both linear and ridge regression models; hence, products with high sales velocity and high brand premium are priced relatively high. Category popularity is the third feature in terms of importance. Inventory ratio has a negative coefficient, which makes sense since the surplus stock of the product should lower the price.

4.4 Model Validation and Robustness

For 5-fold cross validation, the standard deviation of R^2 among folds was less than 0.002 for all three models, suggesting stability and non-reliance on a particular portion of the data. Linear regression residuals were assessed against predictions and also against independent variables; no obvious trends emerged, which is expected given the appropriateness of the linear model for the current data set.

Performance assessment according to categories shows that electronics had the highest per-category R^2 (0.9972), while clothing had the lowest per-category R^2 (0.9918). The latter value is still relatively high and simply reflects the fact that fashion products are affected more by brand reputation and cycles than simply their availability.

4.5 Limitations

While the framework AdvaSell shows high performance of its predictions on the tested dataset, there are still some limitations that need to be considered before implementing it into enterprise e-commerce. First, the presented framework does not take into account competitors' prices, and thus it may not be able to react to competitive price changes, price wars, and any other supply issues. Second, the framework applies a single pricing strategy and does not allow for

any differentiated pricing by customers or groups of customers, like loyalty-based one. The third limitation is that the process of model retraining is a manual one, and it does not have built-in concept-drift monitoring and updating features. As a result, the performance of the framework may deteriorate when the distribution of the underlying market data changes over time. Fourth, the experiment was performed on approximately 30,000 records, while the framework has not been tested on millions of SKUs or high number of parallel prediction requests. This means that the presented results are just a proof-of-concept and cannot be treated as an evidence of scalability of the framework at enterprise scale.

5. Conclusion

In this research paper, AdvaSell is introduced as a dynamic pricing framework powered by AI that helps in e-commerce price prediction with the use of machine learning techniques. A freely accessible dataset known as the Shopee Product Dataset was utilized in constructing a custom-made dataset based on data cleaning, category filtering, categorical encoding, and feature engineering. The developed dataset includes variables such as product name, brand name, category, price, discount, ratings, reviews, and temporal information which are essential factors in e-commerce pricing. Engineering of additional variables like demand activity, category popularity, brand premium, inventory pressure if applicable, seasonality, and price sensitivity provides extra information for regression modelling. Linear Regression and Ridge Regression provide interpretable models while XGBoost is effective in capturing nonlinear relations between variables. The above framework shows how free-of-charge market data can be used to build a dynamic-pricing solution. However, the lack of historical sales and price data of the competition still remains a drawback. Future research must include verified transaction history data, competitor price information in real time, model retraining and forecasting of demands as well as reinforcement learning.

References

[1] Nowak, M., and M. Pawłowska-Nowak. “Dynamic Pricing Method in The E-Commerce Industry using Machine Learning.” Applied Sciences 2024, vol. 14, no. 24: 11668.

[2] El Youbi, R., F. Messaoudi, and M. Loukili. “Machine Learning-Driven Dynamic Pricing Strategies in E-Commerce.” In Proceedings of 14th International Conference on Information and Communi-

cation Systems (ICICS), IEEE, 2023, 1–5.

[3] Hoerl, A. E., and R. W. Kennard. “Ridge Regression: Biased Estimation for Nonorthogonal Problems.” Technometrics 1970, vol. 12, no. 1: 55–67.

[4] Friedman, J. “Greedy Function Approximation: A Gradient Boosting Machine.” Annals of Statistics 2001, vol. 29, no. 5: 1189–1232.

[5] Chen, T., and C. Guestrin. “Xgboost: A Scalable Tree Boosting System.” In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2016, San Francisco, CA: ACM, 785–794.

[6] Yoshi, A. M., A. Rohan, S. A. Mitu, M. M. K. Rabbi, S. Akther, and K. R. Ahmed. “Real-Time Dynamic Pricing using Machine Learning: Integrating Customer Sentiment and Predictive Models for E-Commerce.” International Journal of Advanced Computer Science and Applications 2025, vol. 16, no. 9: 22–31.

[7] Liu, H. “Reinforcement Learning for E-Commerce Dynamic Pricing.” In Proceedings of the 7th International Conference on Economic Management and Green Development (ICEMGD 2023), Singapore: Springer, 2024, 1–8.

[8] Devarajanayaka, K. M., S. S. Banu, D. J. Desai, V. TV, M. R. Palav, and S. K. Dash. “Machine Learning-Based Pricing Optimization for Dynamic Pricing in Online Retail.” Journal of Informatics Education and Research 2014, vol. 4, no. 3.

[9] Hastie, T., R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. New York: Springer, 2009.

[10] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. “Scikit-Learn: Machine Learning in Python.” Journal of Machine Learning Research 2011, vol. 12: 2825–2830.

[11] Flask Documentation. Pallets Projects. Accessed 2026. https://flask.palletsprojects.com.

[12] shopee-dataset – Available [Online] https://huggingface.co/datasets/rebrowser/shopee-dataset