

A Machine Learning-based Decision Framework for Demand Forecasting and Price Optimization in Competitive E-Commerce Environments

Aera K. Leboulluec
University of Texas at Arlington
701 S Nedderman Drive
Arlington, TX 76019

Janhavi Datye
University of Texas at Arlington
701 S Nedderman Drive
Arlington, TX 76019

ABSTRACT

The growth of e-commerce has made accurate demand forecasting and well-judged discount decisions central to pricing strategy. This study develops and evaluates a machine learning framework that forecasts order-level demand and uses the forecast to identify revenue-maximizing discount levels. The analysis uses a publicly available dataset of 30,600 Indian e-commerce orders spanning 36 months (January 2023 – December 2025), covering eight product categories and five geographic zones. Four regression models (Linear Regression, Ridge Regression, Random Forest, and XGBoost) were tuned with time-series cross-validation on the first 30 months, evaluated on the final six months, and compared with two simple baselines. XGBoost produced the most accurate demand forecast (MAE = 12.83 units, $R^2 = 0.413$), reducing error by 21.4% compared with a naive baseline, although its advantage over Linear Regression was small (1.7%). A revenue benchmark showed that the near-perfect accuracy obtained when units sold is used as an input ($R^2 = 0.999$) reflects the identity revenue = price $\times$ units rather than forecasting skill; without units sold, a two-stage demand $\times$ price forecast achieved $R^2 = 0.663$. SHAP analysis identified sales events, customer age, geographic zone, and discount depth as the main drivers of demand, while product price level, category, and calendar month had little effect. A discount simulation showed that revenue peaks at moderate discounts of roughly 20–45% and that deeper discounts consistently reduce revenue.

General Terms

Machine Learning, Forecasting, Data Mining, Algorithms.

Keywords

Demand Forecasting, Discount Optimization, Price Elasticity, Revenue Management, XGBoost, Random Forest, SHAP, Time-Series Cross-Validation, E-Commerce.

1. INTRODUCTION

The rapid growth of e-commerce has transformed retail, generating large volumes of transaction data and intensifying competition between sellers. Online retailers can change prices and discounts quickly and at low cost, which makes pricing one of their most powerful levers, but also one of the easiest to misuse. Using that lever well requires two capabilities: forecasting how much customers will buy, and understanding how that demand responds to price. Ferreira et al. [1] showed that combining machine learning demand forecasts with price optimization can raise revenue for an online retailer, demonstrating the practical value of linking the two.

Demand forecasting supports inventory planning, supply chain coordination, and promotion design. Traditional statistical

approaches often struggle with the nonlinear relationships between demand and its drivers, such as price, promotions, customer characteristics, and market conditions. Machine learning methods can model these relationships flexibly from data [2], and large-scale evaluations such as the M5 forecasting competition, which used Walmart retail sales data, found that tree-based machine learning methods were among the most accurate approaches for retail demand [3]. Ensemble methods such as Random Forest [4] and gradient-boosted trees, including XGBoost [5], are now widely used for this purpose, and explanation methods such as SHAP [6] make it possible to see which factors drive their predictions.

Pricing decisions depend on price elasticity, the sensitivity of demand to changes in price. A lower price usually increases the number of units sold but reduces the revenue earned on each unit, so revenue is maximized at the price where these two effects balance [7, 8]. In online retail, the price a customer pays is often set through a discount on a list price, especially during promotional sales events. The discount level is therefore the practical price lever studied in this paper.

Two methodological issues can make machine learning results in this area look stronger than they are. First, when revenue is the prediction target and the quantity sold is among the inputs, a model can reach near-perfect accuracy simply by learning that revenue equals price multiplied by quantity. Because the quantity sold is only known after a sale occurs, such a model does not forecast anything useful in advance. Second, when observations are ordered in time, randomly shuffled training and test sets allow a model to learn from the future, which produces optimistic accuracy estimates; evaluation should instead respect the time order of the data [9]. A third, related issue is that model accuracy is rarely compared with simple baselines, which makes it hard to judge how much value a complex model adds.

This study addresses these issues using a 36-month dataset of Indian e-commerce orders [10]. Indian e-commerce provides a useful setting because demand varies across geographic zones, is strongly affected by festival sales events, and differs between mass-market and premium brands. The dataset also records the level of competition and inventory pressure for each order, which allows the study to test whether competitive conditions affect demand once discounts are taken into account.

The study develops a framework that forecasts order-level demand with four regression models (Linear Regression, Ridge Regression, Random Forest, and XGBoost), explains the drivers of demand, and uses the demand forecast to identify revenue-maximizing discount levels. The main contributions are threefold:

- (i) **A leakage-aware evaluation of demand forecasting models.** Models are trained on the first 30 months and tested on the final six, tuned with time-series cross-validation, and compared with two simple baselines, so that the reported accuracy reflects realistic forecasting conditions.
- (ii) **A revenue forecasting benchmark.** The study measures how much of the apparent accuracy of revenue models comes from using units sold as an input, and shows that a two-stage forecast (predicted demand multiplied by price) is a more accurate and more useful alternative when units sold is unknown.
- (iii) **A link from demand forecasts to discount decisions.** Using SHAP analysis and a discount simulation checked against the raw data, the study identifies the main drivers of demand, estimates price elasticity at different discount levels, and identifies the discount ranges that maximize revenue for different sales events and brand types.

The remainder of this paper is organized as follows. Section 2 describes the dataset and the methodology. Section 3 presents the regression models. Section 4 reports and discusses the results. Section 5 concludes the paper and outlines limitations and directions for future research.

2. DATA AND METHODOLOGY

2.1 Dataset

The study uses the Indian E-commerce Pricing, Revenue Growth Dataset, publicly available on Kaggle [10]. It contains 30,600 order records covering 36 months, from January 2023 to December 2025, with 16 attributes describing each order's pricing, customer, product, and market conditions (Table 1). Orders come from 14 Indian states grouped into five geographic zones and cover eight product categories. The dataset contains no missing values.

Table 1. Description of dataset variables

Variable	Description
order_id	Unique order identifier
order_date	Month of the order (recorded as the last day of each month, Jan 2023 – Dec 2025)
state	Indian state of the customer (14 states)
zone	Geographic zone (North, South, East, West, Central)
category	Product category (Beauty & Personal Care, Electronics, Fashion, Footwear, Grocery Essentials, Home & Living, Premium Lifestyle, Sports & Fitness)
brand_type	Brand tier (Mass or Premium)
customer_gender	Gender of the customer (Male or Female)
customer_age	Age of the customer in years (18 – 65)
base_price	List price before discount (INR)
discount_percent	Discount applied to the list price (%)
final_price	Price paid per unit after discount (INR)
units_sold	Number of units in the order — primary target (demand)
revenue	Order revenue in INR — secondary target (benchmark)
sales_event	Type of sales event (Normal or Festival)
competition_intensity	Level of market competition (Low, Medium, High)
inventory_pressure	Level of inventory pressure (Low or High)

Table 2 summarizes the main numeric variables. Units sold per order range from 2 to 161, with a mean of 31.7 and a right-skewed distribution (median = 27). Discounts range from almost

0% to 65%, with a mean of 38.1%. Festival sales events account for 25.2% of orders and mass-market brands for 75.1%.

Table 2. Descriptive statistics of the numeric variables (n = 30,600)

Variable	Mean	Std. dev.	Min	Median	Max
units_sold	31.71	21.38	2	27	161
revenue (INR)	71,062	83,627	248	42,139	906,141
base_price (INR)	3,898.64	3,371.29	100.03	2,975.02	14,999.17
discount_percent (%)	38.11	16.61	0.01	37.29	65.00
final_price (INR)	2,415.02	2,264.89	35.01	1,728.60	14,682.28
customer_age (years)	35.57	12.87	18	33	65

Three structural properties of the dataset shape the analysis. First, revenue is exactly equal to final price multiplied by units sold for every order, and final price is exactly equal to base price multiplied by $(1 - \text{discount})$. Revenue is therefore fully determined once price and units sold are known, which is why units sold, rather than revenue, is used as the primary forecasting target. Second, every month contains exactly 850 orders, all dated on the last day of the month, so the month is the finest time resolution available. Third, these regular patterns suggest that the dataset is simulated or heavily processed rather than raw platform data. The findings should therefore be read as a demonstration of the framework and its evaluation, and they need confirmation on operational data before being applied to real pricing decisions.

2.2 Feature Engineering and Encoding

Two calendar features were extracted from the order date: month (1 – 12) and year. A season feature was derived using an India-specific mapping: Winter (December – February), Summer (March – May), Monsoon (June – September), and Festive (October – November). Four further features were constructed: discount amount (base price – final price), price ratio (final price ÷ base price), brand_event (brand type combined with sales event), and category_zone (category combined with zone). A revenue-per-unit feature was not used, because it is identical to final price in this dataset and is computed from the revenue target.

The demand models use 19 input features: the 11 categorical variables (state, zone, category, brand_type, customer_gender, sales_event, competition_intensity, inventory_pressure, season, brand_event, and category_zone) and 8 numeric variables (customer_age, base_price, discount_percent, final_price, month, year, discount_amount, and price_ratio). Units sold and revenue are excluded from the inputs.

Categorical variables were encoded differently for the two model families. For Linear and Ridge Regression, they were one-hot encoded (one binary column per category, with the first category dropped) and numeric variables were standardized to zero mean and unit variance, because label codes such as 0, 1, 2 for states have no numeric meaning in a linear model. For Random Forest and XGBoost, categorical variables were label encoded, which tree-based models can split on directly, and no scaling was applied. All encoders and scalers were fitted on the training data only.

2.3 Train–Test Split and Cross-Validation

To reflect real forecasting conditions, the data were split by calendar month rather than at random. The training set contains the first 30 months (January 2023 – June 2025, 25,500 orders) and the test set contains the final six months (July – December 2025, 5,100 orders), so no month appears in both sets and every model is evaluated only on months it has never seen.

Hyperparameters were tuned on the training set only, using five-fold time-series cross-validation [9]. In each fold, the model is

trained on earlier orders and validated on the orders that follow, so the model never learns from the future. The same procedure was used to report the cross-validated R^2 of each final model. The test set was used once, for the final evaluation.

2.4 Evaluation Metrics

Model performance was measured with five metrics, where y_i is the actual value, $\hat{y}_i$ the predicted value, $\bar{y}$ the mean of the actual values, n the number of test observations, and p the number of input features:

$$MAE = (1/n) \sum |y_i - \hat{y}_i| \quad \dots \text{eq (1)}$$

$$RMSE = \sqrt{(1/n) \sum (y_i - \hat{y}_i)^2} \quad \dots \text{eq (2)}$$

$$R^2 = 1 - \sum (y_i - \hat{y}_i)^2 / \sum (y_i - \bar{y})^2 \quad \dots \text{eq (3)}$$

$$\text{Adjusted } R^2 = 1 - (1 - R^2)(n - 1) / (n - p - 1) \quad \dots \text{eq (4)}$$

$$MAPE = (100/n) \sum |(y_i - \hat{y}_i) / y_i| \quad \dots \text{eq (5)}$$

MAE and RMSE are reported in the units of the target (units sold for demand, INR for revenue). MAE is used as the main accuracy measure, because MAPE becomes very large for small orders, where a small absolute error turns into a large percentage error.

2.5 Baselines

Two simple baselines were used to judge whether the models add value. The **naive baseline** predicts the training-set average of units sold (31.8 units) for every test order. The **segment baseline** predicts the training-set average for the order's group, defined by sales event $\times$ brand type $\times$ zone (20 groups). Both baselines use only information available before an order is placed.

2.6 Experimental Design

The analysis was carried out in four parts.

Demand forecasting (primary task). The four regression models described in Section 3 were trained to predict units sold from the 19 input features and were compared with the two baselines on the test set.

Revenue forecasting benchmark. Revenue was forecast under three scenarios. In scenario B1, units sold was added as an input (20 features). In scenario B2, revenue was predicted directly from the 19 demand features, without units sold. In scenario B3, a two-stage forecast was formed by multiplying each model's demand forecast by the order's final price. Only B2 and B3 use information that is available before an order is placed.

Explaining demand. SHAP values [6] were computed for the XGBoost demand model on the test set using the TreeExplainer method, which measures how much each feature moves an individual prediction away from the average prediction. Features were ranked by their mean absolute SHAP value.

Discount simulation. To link the demand forecast to pricing decisions, the discount of every test order was set in turn to each value from 0% to 65% in steps of 2.5 percentage points. Final price, discount amount, and price ratio were recomputed accordingly, all other attributes were held fixed, and demand was predicted with the XGBoost model. For each discount level d , a revenue index was calculated as

$$\text{Revenue index}(d) = \text{predicted units}(d) \times (1 - d/100) \quad \dots \text{eq (6)}$$

which equals expected revenue per INR 1 of list price and removes the effect of differences in product price. Because tree-based models cannot extrapolate beyond the range of their training data, the revenue-maximizing discount for each group was searched only between the 5th and 95th percentiles of the discounts observed for that group in the training data. Discounts

with a revenue index within 2% of the maximum were reported as a near-optimal range. The price elasticity of demand at discount level d was estimated as an arc (midpoint) elasticity between $d - 5$ and $d + 5$ percentage points:

$$E(d) = [(Q_2 - Q_1) / ((Q_2 + Q_1)/2)] / [(P_2 - P_1) / ((P_2 + P_1)/2)] \quad \dots \text{eq (7)}$$

where subscripts 1 and 2 refer to discount levels $d - 5$ and $d + 5$, Q is the average predicted demand, and $P = 1 - d/100$ is the price relative to list price. Demand is elastic when $E < -1$, meaning a deeper discount increases revenue, and inelastic when $-1 < E < 0$, meaning a deeper discount reduces revenue. To check that the simulated response reflects the data rather than the model, it was compared with raw training-set averages of units sold and of the revenue index in 5-percentage-point discount bands containing at least 50 orders. The simulation estimates associations learned from historical data, not causal effects, and maximizes revenue rather than profit, because the dataset contains no cost information.

All analyses were carried out in Python 3.13 using scikit-learn 1.9 [11], XGBoost 3.4 [5], and SHAP 0.54 [6], with a fixed random seed (42) for reproducibility.

3. MODELS AND METHODS

Four regression models were compared, ranging from a simple linear model to a gradient-boosted ensemble. All models were trained on the same training set and tuned with the five-fold time-series cross-validation described in Section 2.3, using R^2 as the selection criterion. The search spaces and the selected hyperparameters are listed in Table 3.

3.1 Linear Regression

Linear Regression models the target as a weighted sum of the input features plus a constant [12]:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p \quad \dots \text{eq (8)}$$

where β_0 is the intercept and β_1 to β_p are coefficients estimated by ordinary least squares, which minimizes the sum of squared errors on the training data. The model is fast and easy to interpret but cannot capture nonlinear relationships or interactions unless they are added as features. It was used without regularization as the reference model, with the one-hot encoded and standardized inputs described in Section 2.2.

3.2 Ridge Regression

Ridge Regression extends Linear Regression by adding an L2 penalty on the size of the coefficients, which reduces overfitting and stabilizes the estimates when features are correlated [13]. The coefficients minimize

$$\|y - X\beta\|^2 + \alpha \|\beta\|^2 \quad \dots \text{eq (9)}$$

where $\alpha \geq 0$ controls the strength of the penalty; $\alpha = 0$ gives ordinary least squares. Because the penalty depends on the scale of the coefficients, the numeric inputs were standardized before fitting. The value of α was selected from {0.01, 0.1, 1, 10, 50, 100, 200, 500, 1000} by grid search with time-series cross-validation.

3.3 Random Forest Regression

Random Forest is an ensemble of decision trees, each trained on a bootstrap sample of the training data, with only a random subset of features considered at each split [4]. This randomness makes the trees different from one another, and averaging their predictions reduces variance:

$$\hat{y} = (1/N) \sum_{k=1}^N T_k(x) \quad \dots \text{eq (10)}$$

where N is the number of trees and $T_k(x)$ is the prediction of the k -th tree. Random Forest captures nonlinear effects and interactions without requiring them to be specified in advance. Hyperparameters were selected by randomized search with 15 candidate settings over the number of trees, maximum tree depth, minimum number of samples required to split a node and to form a leaf, and the share of features considered at each split.

3.4 XGBoost Regression

XGBoost (Extreme Gradient Boosting) builds an ensemble of decision trees sequentially, with each new tree fitted to correct the errors of the trees before it [5]. After t rounds, the prediction is

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta f_t(x) \quad \dots \text{eq (11)}$$

where f_t is the tree added in round t and η is the learning rate, which scales the contribution of each tree. Each tree is chosen to minimize a regularized objective,

$$L = \sum_i l(y_i, \hat{y}_i) + \sum_k [\gamma T_k + \frac{1}{2} \lambda \|w_k\|^2 + \alpha \|w_k\|_1] \quad \dots \text{eq (12)}$$

where l is the squared-error loss, T_k is the number of leaves in tree k , w_k are its leaf weights, γ penalizes adding leaves, and λ and α are the L2 and L1 penalties on the leaf weights. These penalties, together with row and column subsampling, limit overfitting.

Hyperparameters were selected by randomized search with 30 candidate settings. The number of boosting rounds was then set by early stopping: the model was trained on January 2023 – December 2024 and validated on the last six training months (January – June 2025), adding up to 3,000 rounds and stopping when the validation RMSE did not improve for 50 consecutive rounds. The final model was refitted on the full training set with the selected number of rounds. The test set was not used at any point in this process.

Table 3. Hyperparameter search spaces and selected values for the demand and revenue (B2) models

Model	Parameter	Search space	Demand	Revenue (B2)
Ridge	α	0.01, 0.1, 1, 10, 50, 100, 200, 500, 1000	100	0.01
Random Forest	Number of trees	200, 300	200	300
	Maximum depth	8, 12, 16, 20	8	16
	Min. samples to split	2, 5, 10	5	5
	Min. samples per leaf	1, 2, 4, 8	2	8
	Features per split	$\sqrt{p}$, log-p, 50%, 100%	50%	50%
XGBoost	Boosting rounds	300, 500, 800 during search; final value set by early stopping (max. 3,000)	355	226
	Maximum depth	3, 4, 5, 6, 8	3	4
	Learning rate (η)	0.01, 0.03, 0.05, 0.1	0.03	0.03
	Row subsample	0.6, 0.8, 1.0	0.8	0.8
	Column subsample	0.6, 0.8, 1.0	0.6	0.6
	Min. child weight	1, 5, 10	10	5
	γ (min. split loss)	0, 0.1, 0.5	0.1	0.5
	λ (L2 penalty)	1, 5, 10	1	10
	α (L1 penalty)	0, 0.1, 1	0	1

Linear Regression has no hyperparameters. p = number of input features. Models for revenue scenario B1 were tuned with the same procedure.

The selected settings are consistent with the results in Section 4. For demand, both tree-based models chose shallow trees (maximum depth 8 for Random Forest and 3 for XGBoost) and Ridge chose a moderate penalty, which fits a demand

relationship that is close to additive and contains few strong interactions.

4. RESULTS AND DISCUSSION

All results in this section come from a single run of the analysis pipeline described in Section 2. Every model was tuned and cross-validated on the training period only (January 2023 – June 2025, 25,500 orders) and was evaluated once on the held-out test period (July – December 2025, 5,100 orders). Demand forecasting is reported first, followed by the revenue benchmark and the discount analysis. Because revenue in this dataset is exactly the product of final price and units sold, the two tasks are reported separately, and every model is compared with the two baselines defined in Section 2.5.

4.1 Demand Forecasting Performance

Table 4 reports test-set performance for the two baselines and the four models, together with five-fold time-series cross-validation (CV) R^2 computed on the training period. XGBoost achieved the lowest error on every metric, with a mean absolute error (MAE) of 12.83 units, a root mean squared error (RMSE) of 16.08 units, and an R^2 of 0.413. Random Forest was a close second (MAE = 12.88, R^2 = 0.405), while Linear and Ridge Regression produced almost identical results (MAE = 13.05 and 13.04, R^2 = 0.389 for both).

Table 4. Demand forecasting performance on the test period (target = units sold per order)

Model	MAE	RMSE	R^2	Adj. R^2	MAPE (%)	CV R^2 (mean $\pm$ SD)
Naive mean	16.32	21.00	-0.001	-0.001	98.53	—
Segment mean	14.24	18.17	0.251	0.251	82.84	—
Linear	13.05	16.41	0.389	0.387	71.80	0.379 $\pm$ 0.012
Ridge	13.04	16.41	0.389	0.387	71.85	0.381 $\pm$ 0.009
Random Forest	12.88	16.19	0.405	0.403	72.04	0.390 $\pm$ 0.014
XGBoost	12.83	16.08	0.413	0.411	71.07	0.404 $\pm$ 0.011

MAE and RMSE in units. CV R^2 = five-fold time-series cross-validation on the training period; baselines have no tuned parameters. Best values in bold.

All four models clearly outperform both baselines (Figure 1). Compared with the naive baseline (MAE = 16.32 units), XGBoost reduces MAE by 21.4% and RMSE by 23.5%. Compared with the stronger segment baseline (MAE = 14.24 units, R^2 = 0.251), it still reduces MAE by 9.9% and raises R^2 by 0.16. The models therefore learn useful information beyond group averages, mainly the effects of discount depth and customer age within each group (Section 4.3).

The gap between the models, however, is small. The MAE of XGBoost is only 1.7% lower than that of Linear Regression. This indicates that the relationship between the available features and demand in this dataset is close to additive: units sold rise steadily with discount and shift up or down with sales event, zone, and customer age, with limited interaction for the tree-based models to exploit. Ridge Regression selected a fairly strong penalty (α = 100) yet matched ordinary least squares, which shows that overfitting is not a concern with 25,500 training orders and 19 features.

The models are also stable over time. Cross-validated R^2 on the training period (0.379 – 0.404, standard deviation $\leq$ 0.014) is close to the test R^2 (0.389 – 0.413), so performance did not degrade on the six unseen months. At the same time, an R^2 of about 0.41 means that roughly 59% of the order-to-order variation in units sold is not explained by the recorded features. This is a moderate level of accuracy and should be read as such. The mean absolute percentage error (MAPE) of about 71% overstates the error, because small orders (as few as 2 units) turn

small absolute errors into large percentages; MAE is therefore used as the main accuracy measure in this section.

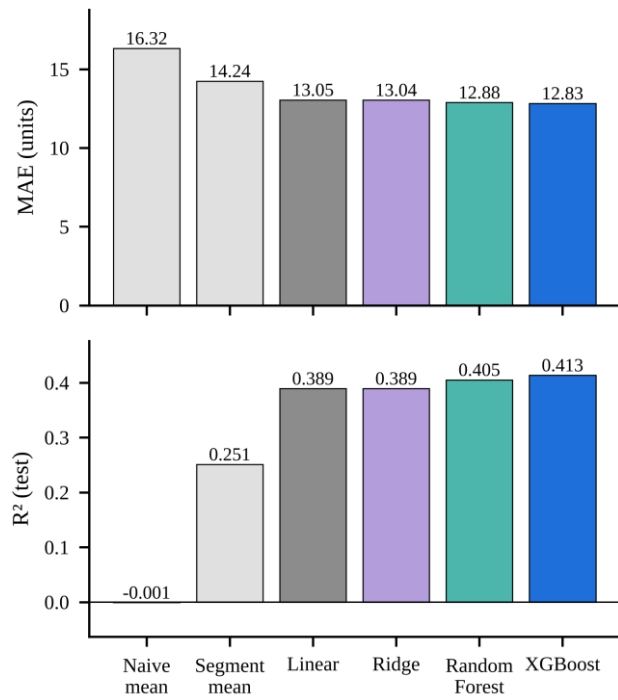


Figure 1. Test-set performance of the demand forecasting models compared with two baselines. Top: MAE in units per order (lower is better). Bottom: R² (higher is better). Grey bars are baselines.

Figure 2 plots actual against predicted units for each model. All four models predict a narrower range than the actual values: they over-predict small orders (orders below 10 units are over-predicted by 16.3 units on average by XGBoost) and under-predict large orders (orders above 60 units are under-predicted by 25.6 units on average). This pull toward the average is expected when a large share of the variation is unexplained. Random Forest and XGBoost separate high-demand orders slightly better than the linear models, which is visible as a denser band of points along the diagonal at higher volumes. The XGBoost residuals are centred at zero (mean = -0.20 units, standard deviation = 16.08 units) and roughly symmetric, but their spread widens as predicted demand increases, so absolute errors are largest for high-volume orders.

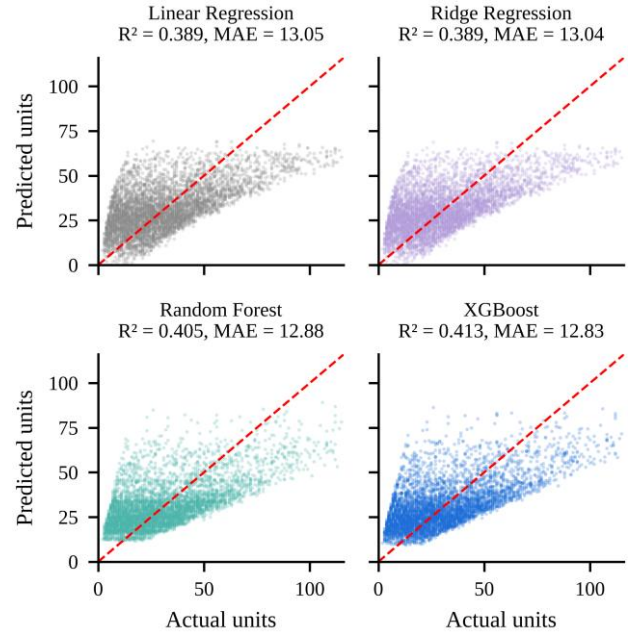


Figure 2. Actual versus predicted units sold on the test period for the four models. The red dashed line marks a perfect prediction. Test R² and MAE are shown above each panel.

4.2 Error Analysis by Segment

To check whether the overall error hides weak spots, Table 5 breaks down the XGBoost test error by product category, zone, sales event, and brand type.

Table 5. XGBoost demand forecast error by segment (test period)

Segment	Level	Orders	Mean units	MAE	MAPE (%)	R ²
Category	Beauty & Personal Care	686	29.6	12.87	76.7	0.368
	Electronics	662	32.4	12.91	63.4	0.409
	Fashion	623	32.5	13.00	67.4	0.394
	Footwear	599	29.6	12.33	72.0	0.361
	Grocery Essentials	607	31.1	12.61	68.6	0.439
	Home & Living	607	31.3	12.80	72.7	0.424
Zone	Premium Lifestyle	658	31.5	13.63	78.1	0.442
	Sports & Fitness	658	31.1	12.41	69.3	0.443
	Central	704	25.4	10.37	66.8	0.353
	East	700	24.1	10.32	73.5	0.231
	North	1,483	32.1	13.27	71.9	0.414
	South	1,454	33.9	14.06	73.1	0.376
Sales event	West	759	35.8	14.19	67.3	0.446
	Festival	1,248	47.2	19.32	72.4	0.271
Brand type	Normal	3,852	25.9	10.72	70.6	0.282
	Mass	3,811	32.6	13.22	68.6	0.410
	Premium	1,289	26.9	11.66	78.5	0.386

MAE in units. R² within a segment is lower than the overall R² because part of the overall explained variation comes from separating one segment from another.

Accuracy is consistent across the eight product categories: MAE ranges only from 12.33 to 13.63 units and R² from 0.36 to 0.44, so no category is systematically harder to forecast. Differences across zones and sales events follow the level of demand rather than model quality. Festival orders have almost twice the MAE of normal orders (19.32 vs 10.72 units), but they are also almost twice as large on average (47.2 vs 25.9 units), and the percentage error is nearly the same (72.4% vs 70.6%). Likewise, the Central and East zones have the lowest MAE because they have the lowest demand (about 24 – 25 units per order). The East zone has the lowest R² (0.231), meaning that order-to-order variation there is the least well explained. For practical use, the forecast error band should scale with the expected demand level, and festival

periods need wider buffers in absolute units even though the forecast is not relatively worse.

4.3 Drivers of Demand

Figure 3 shows the mean absolute SHAP value [6] of each feature for the XGBoost demand model, which measures how many units, on average, each feature moves a prediction away from the average prediction of 31.8 units. Sales event is the strongest single driver (5.58 units). Festival orders are pushed up by 11.2 units on average and normal orders pulled down by 3.8 units. Customer age is second (3.53 units) and has a clear negative effect: orders from customers aged 25 or younger are raised by about 5.9 units, while orders from customers over 55 are lowered by about 5.7 units. Zone is third (3.22 units), with the South and West zones above average (+3.1 and +3.5 units) and the Central and East zones below (−5.9 and −5.3 units).

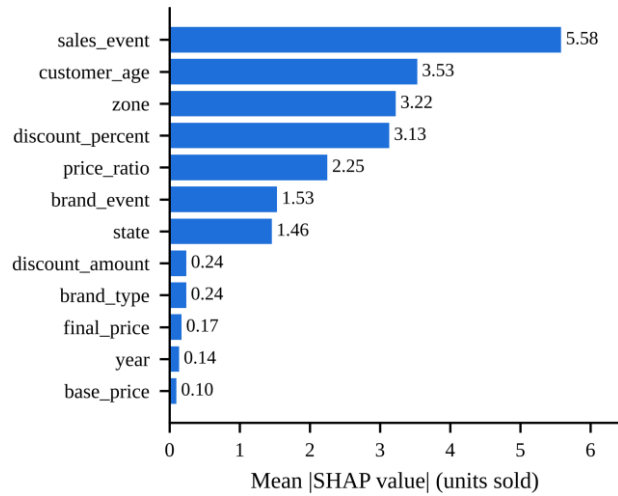


Figure 3. Mean absolute SHAP value of the twelve most important features in the XGBoost demand model (test period), in units sold per order.

Discount depth appears as three separate features (discount percent, price ratio, and discount amount) because they carry the same information. Their individual SHAP values (3.13, 2.25, and 0.24 units) therefore understate the role of discount. When the three are combined, discount moves a prediction by 5.51 units on average, which is as large as the effect of sales event. Orders with discounts below 20% are lowered by 8.8 units and orders with discounts above 50% are raised by 8.2 units.

Several features that might be expected to matter have almost no effect once the others are known. Base price, final price, and product category each contribute less than 0.25 units, which means that, in this dataset, how much a customer buys depends on the discount offered rather than on the price level of the product. Month and season also contribute almost nothing (0.04 and 0.02 units). This is because festival sales events are spread evenly across the year, making up between 23.6% and 26.7% of orders in every calendar month; the demand lift comes from the sales event itself, not from the time of year. Finally, competition intensity and inventory pressure show differences in raw averages (for example, 34.0 units under high competition vs 29.8 under low), but their SHAP values are near zero (0.04 and 0.02 units). Orders under high competition also carry deeper discounts (42.9% vs 33.8% on average), so once discount is in the model these two features add no further information.

4.4 Revenue Forecasting Benchmark

Table 6 and Figure 4 compare revenue forecasts under the three scenarios defined in Section 2.6: B1, with units sold as an input;

B2, direct prediction without units sold; and B3, a two-stage forecast of predicted demand $\times$ final price.

Table 6. Revenue forecasting performance on the test period under three scenarios

Scenario	Model	MAE (INR)	RMSE (INR)	R ²	MAPE (%)
B1: units known	Linear	24,066	38,980	0.785	167.88
	Ridge	24,053	38,981	0.785	167.77
	Random Forest	950	3,778	0.998	1.26
	XGBoost	1,282	2,122	0.999	4.46
B2: direct	Linear	34,091	52,820	0.605	123.03
	Ridge	34,089	52,819	0.605	122.89
	Random Forest	30,570	50,040	0.645	73.11
	XGBoost	30,378	49,204	0.657	76.91
B3: two-stage	Linear	30,692	50,230	0.642	71.80
	Ridge	30,673	50,174	0.643	71.85
	Random Forest	30,120	48,873	0.661	72.04
	XGBoost	29,995	48,773	0.663	71.07

B1 uses units sold, which is only known after an order is placed, so it is not a usable forecast. B2 and B3 use only information available before the order.

When units sold is known (B1), the tree-based models reach near-perfect accuracy ($R^2 = 0.999$ for XGBoost and 0.998 for Random Forest), while the linear models reach only 0.785. This pattern is not evidence of forecasting skill. Since revenue equals final price $\times$ units sold exactly, a tree-based model given both quantities simply learns to approximate their product, which a linear model cannot represent. Because units sold is only known after an order occurs, scenario B1 cannot be used to plan prices or inventory in advance.

When units sold is not available, accuracy falls sharply. Predicting revenue directly (B2), XGBoost achieves $R^2 = 0.657$, and the linear models reach 0.605. The two-stage approach (B3) performs better than the direct approach for every model: XGBoost reaches $R^2 = 0.663$ with an MAE 1.3% lower than in B2, and Linear Regression improves the most (from 0.605 to 0.642), because multiplying the demand forecast by the known price supplies the multiplicative structure that a linear model cannot learn by itself. The drop from $R^2 = 0.999$ in B1 to 0.663 in B3 shows how much of the apparent accuracy of revenue models comes from knowing units sold. The two-stage approach is also more useful in practice: the demand model can be reused to test different discount levels before they are applied, which is done in the next section.

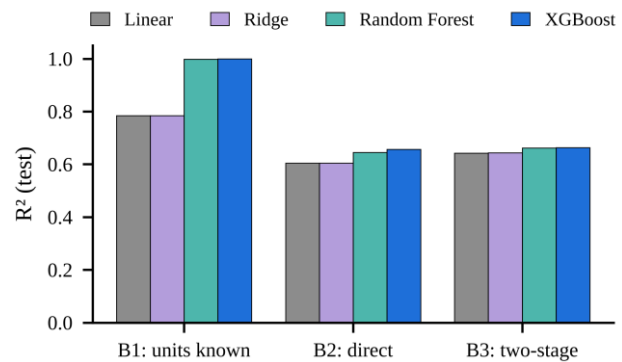


Figure 4. Test R^2 of the four models for revenue forecasting under scenarios B1 (units sold known; not a usable forecast), B2 (direct prediction without units sold), and B3 (two-stage forecast of predicted demand $\times$ final price).

4.5 Discount Response and Revenue-Maximizing Discount

The discount simulation described in Section 2.6 was applied to the XGBoost demand model. Figure 5 shows the resulting demand and revenue-index curves for normal and festival sales

events, together with the raw training-data averages used as a check.

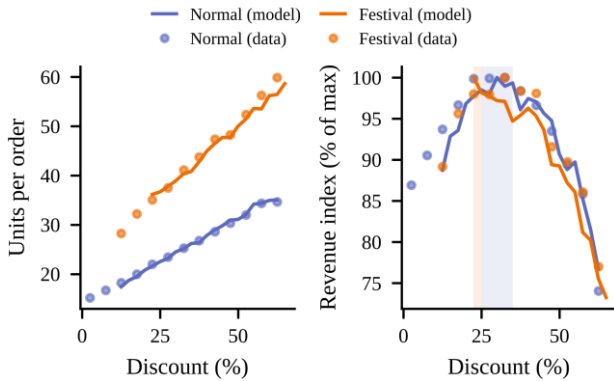


Figure 5. Response of demand and revenue to discount. Left: units sold per order. Right: revenue index as a percentage of its maximum. Lines show the XGBoost simulation within the observed discount range; dots show raw training-data averages per 5-point discount band; shaded areas show the model's near-optimal range (within 2% of the maximum).

Predicted demand rises almost linearly with discount (Figure 5, left). Under normal events, predicted demand increases from 17.4 units per order at a 12.5% discount to 35.0 units at 62.5%; under festival events, it increases from 36.2 units at 22.5% to 58.6 units at 65%. The model curves closely follow the raw data averages, confirming that the learned response reflects the data rather than an artifact of the model.

Revenue, however, follows an inverted-U shape (Figure 5, right), because each extra point of discount lowers the price of every unit while adding fewer and fewer extra units. The arc price elasticity of demand (Equation 7) makes this explicit. For normal events, elasticity is -1.65 at a 15% discount and -1.24 at 25% (demand is elastic, so a deeper discount raises revenue), but it becomes -0.83 at 35%, -0.60 at 45%, and -0.53 at 55% (demand is inelastic, so a deeper discount lowers revenue). For festival events, elasticity is already below 1 in absolute value across the observed range (-0.79 at 25%, -0.94 at 35%, -0.59 at 45%, and -0.52 at 55%). Revenue is therefore highest where elasticity is close to -1 , at moderate discounts, and falls quickly at deep discounts: at a 62.5% discount, the normal-event revenue index is 23.3% below its peak, and at 65% the festival index is 26.8% below its peak.

Table 7. Revenue-maximizing discount by segment compared with current discount levels (all values in %)

Segment	Range	Current	Best	Near-opt. (model)	Near-opt. (data)	Gain
Normal (all)	10.6 – 64.4	35.9	30.0	25 – 35	20 – 40	0.7
Festival (all)	20.7 – 65.0	44.9	22.5	22.5 – 25	30 – 45	6.8
Normal, Mass	13.2 – 65.0	39.5	30.0	25 – 35	—	2.7
Normal, Premium	6.5 – 43.1	24.9	30.0	25 – 35	—	2.0
Festival, Mass	23.5 – 65.0	48.3	25.0	25 – 32.5	—	9.9
Festival, Premium	16.4 – 52.7	34.6	22.5	17.5 – 32.5	—	3.2

Range = 5th to 95th percentile of training-data discounts. Current = current average discount. Best = model revenue-maximizing discount. Near-opt. = discounts with a revenue index within 2% of the maximum. Gain = predicted change in the revenue index from the current average discount to the best discount. Revenue, not profit, is maximized.

Table 7 compares the revenue-maximizing discount with current practice. For normal events, the model optimum is a 30% discount with a near-optimal range of 25 – 35%, which agrees

with the raw data (20 – 40%). The current average discount of 35.9% already lies inside this range, and moving to the optimum would raise the revenue index by only 0.7%. For festival events, the evidence is less clear. The model places the optimum at 22.5%, at the low end of the discounts observed for festival orders, and predicts a 6.8% gain over the current average of 44.9%. The raw data, however, show an almost flat revenue curve between 30% and 45%, which includes the current average. The festival gain should therefore be treated as uncertain. The result that holds in both the model and the data is that discounts above about 45% reduce revenue for both event types. This matters most for mass-market brands during festivals, where the current average discount (48.3%) is the deepest of any segment and the model estimates the largest potential gain (9.9%). These estimates describe associations in historical data rather than causal effects and maximize revenue rather than profit; these limits are discussed in Section 5.1.

4.6 Summary of Findings

Four main findings emerge. First, order-level demand can be forecast with moderate accuracy (XGBoost $R^2 = 0.413$, MAE = 12.83 units), clearly better than simple baselines, with stable performance over time and across product categories. Second, the advantage of ensemble models over linear models is small for demand (1.7% lower MAE), because the demand relationship in this data is close to additive. Third, the near-perfect accuracy obtained for revenue prediction when units sold is an input ($R^2 = 0.999$) does not reflect forecasting skill; without units sold, the best achievable R^2 is 0.663, obtained with a two-stage demand $\times$ price forecast. Fourth, sales events, customer age, zone, and discount depth drive demand, while product price level, category, and calendar month do not, and revenue is highest at moderate discounts of roughly 20 – 45%, with discounts above about 45% consistently reducing revenue.

5. CONCLUSION AND FUTURE RESEARCH

This study developed and evaluated a machine learning framework that forecasts order-level demand in e-commerce and uses the forecast to support discount decisions. Using 36 months of Indian e-commerce orders, four regression models (Linear Regression, Ridge Regression, Random Forest, and XGBoost) were trained on the first 30 months, tuned with time-series cross-validation, and evaluated on the final six months against two simple baselines.

XGBoost produced the most accurate demand forecast (MAE = 12.83 units, $R^2 = 0.413$), reducing error by 21.4% compared with the naive baseline and by 9.9% compared with the segment baseline. Its advantage over Linear Regression, however, was small (1.7% lower MAE), and the selected models used shallow trees, indicating that the demand relationship in this dataset is close to additive. Accuracy was stable over time and consistent across product categories, with absolute errors rising in proportion to the level of demand.

The revenue benchmark showed why the choice of forecasting target matters. When units sold was used as an input, tree-based models predicted revenue almost perfectly ($R^2 = 0.999$), but this reflects the identity revenue = price $\times$ units rather than forecasting skill, because units sold is only known after an order is placed. Without units sold, the best result was obtained with a two-stage forecast of predicted demand multiplied by price ($R^2 = 0.663$), which outperformed direct revenue prediction for all four models.

SHAP analysis showed that sales events, discount depth, customer age, and geographic zone were the main drivers of

demand. Product price level, product category, and calendar month had little effect, and competition intensity and inventory pressure added no information once discount was taken into account. The discount simulation showed that demand rises steadily with discount but revenue follows an inverted-U shape: demand is price-elastic at shallow discounts and becomes inelastic at deep ones. Revenue was highest at moderate discounts of roughly 20–45%, and discounts above about 45% reduced revenue for both normal and festival sales events.

These results have three practical implications. First, revenue forecasts should be built from a demand forecast rather than from inputs that already contain the quantity sold; otherwise reported accuracy will be misleading. Second, new forecasting models should be compared with simple baselines and with linear models, because a complex model may add little. Third, deep discounts deserve particular scrutiny: in this data, discounts above about 45%, which were applied to 59% of mass-market orders during festival events, reduced revenue rather than increasing it.

5.1 Limitations

The study has several limitations. First, the dataset shows regular patterns, including exactly 850 orders per month and revenue that exactly equals price multiplied by units, which suggest that it is simulated or heavily processed; the findings therefore need to be confirmed on operational data. Second, the data are recorded at monthly resolution, so weekly or daily patterns and the effect of recent sales history could not be studied. Third, about 59% of the variation in order-level demand remains unexplained by the available features. Fourth, the discount simulation estimates associations learned from historical data, not causal effects, and it is limited to the range of discounts observed in the data. Finally, the analysis maximizes revenue rather than profit, because the dataset contains no cost or margin information.

5.2 Future Research

Future work could extend this study in four directions. First, the framework should be applied to operational e-commerce data that include product costs, so that discounts can be optimized for profit rather than revenue. Second, data with daily or weekly timestamps would allow time-series features, such as recent sales and promotion history, to be added to the demand models. Third, the effect of discounts could be estimated causally through controlled pricing experiments or causal machine learning methods, which would make the recommended discount ranges more reliable. Fourth, probabilistic forecasts that provide prediction intervals, rather than single values, would support inventory and safety-stock decisions, particularly during festival events, when demand and forecast errors are both highest.

6. ACKNOWLEDGMENTS

The authors express sincere thanks to the experts and specialists who contributed to the development of this research. Acknowledgment is also extended to the contributors of the

Indian E-commerce Pricing, Revenue Growth Dataset [10] used in this study.

7. REFERENCES

- [1] K. J. Ferreira, B. H. A. Lee, and D. Simchi-Levi, “Analytics for an online retailer: Demand forecasting and price optimization,” *Manufacturing & Service Operations Management*, vol. 18, no. 1, pp. 69–88, 2016.
- [2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.
- [3] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “M5 accuracy competition: Results, findings, and conclusions,” *International Journal of Forecasting*, vol. 38, no. 4, pp. 1346–1364, 2022.
- [4] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [5] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794.
- [6] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30*, Long Beach, CA, USA, 2017, pp. 4765–4774.
- [7] R. L. Phillips, *Pricing and Revenue Optimization*. Stanford, CA, USA: Stanford University Press, 2005.
- [8] K. T. Talluri and G. J. van Ryzin, *The Theory and Practice of Revenue Management*. New York, NY, USA: Springer, 2004.
- [9] C. Bergmeir and J. M. Benítez, “On the use of cross-validation for time series predictor evaluation,” *Information Sciences*, vol. 191, pp. 192–213, 2012.
- [10] S. Shukla, “Indian E-commerce Pricing, Revenue Growth Dataset,” Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/shukla922/indian-e-commerce-pricing-revenue-growth>. [Accessed: May 2026].
- [11] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [12] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, 6th ed. Hoboken, NJ, USA: Wiley, 2021.
- [13] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.