International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Expiry-Based Dynamic Discount System: A Machine Learning Approach to Perishable Inventory Pricing

Tej Save[1], Aditya Nair[2], Shubham Chimkar[3], Vighnesh Birru[4], Mr. Mohan Kumar[5]

[1,2,3,4] B.E., Data Engineering, Universal College of Engineering

[5] Professor, Data Engineering, Universal College of Engineering

---***---

Abstract - In the retail and Fast-Moving Consumer Goods industries, perishable goods such as dairy products, bakery items, fresh produce, and pharmaceuticals face persistent challenges in inventory management and revenue optimization. Products approaching expiry often remain unsold, resulting in financial losses and environmental waste. Traditional discount strategies are static, manual, and lack data-driven intelligence. This paper presents an Expiry-Based Dynamic Discount System that leverages machine learning and data analytics to automatically compute optimal discount percentages based on product freshness, days to expiry, overstock ratios, and category-specific parameters. The system employs Ridge and Lasso regression models with hyperparameter tuning, combined with a formula-based feature engineering pipeline including freshness scores, urgency levels, and perishability assessments. An interactive Streamlit dashboard enables store managers to upload inventory, apply filters, override discounts, and export recommendations. Experimental results demonstrate a projected 67% reduction in product waste and a 12% net revenue improvement over manual approaches, with regression models achieving R^2 values above 0.85.

Key Words: Dynamic Pricing, Machine Learning, Perishable Inventory, XGBoost, Gradient Boosting, Retail Analytics.

---***---

1. INTRODUCTION: The retail industry, particularly segments dealing with perishable goods, confronts a chronic challenge: how to price items as they approach expiry in a way that clears inventory, preserves profitability, and minimises waste. Products including dairy, bakery, fresh produce, frozen foods, and pharmaceuticals represent a substantial share of retail inventory. Their strict expiration constraints mean that failure to sell them in time results in complete value loss, environmental harm, and a direct impact on store margins.

Conventional pricing strategies rely on static, predetermined discount schedules and manual observation by store managers. This approach frequently results in premature discounting—unnecessarily eroding margins—or delayed discounting, which allows products to expire unsold. Neither approach systematically leverages the rich inventory data that modern retail systems generate. The Expiry-Based Dynamic Discount System (EDDS) addresses these limitations through intelligent automation powered by supervised machine learning, transforming discount management from a reactive process to a proactive, data-driven strategy.

2. LITERATURE REVIEW: Dynamic pricing for perishable goods has attracted significant academic attention. Bazrafshan, Emami, and Mashreghi [1] proposed a nonlinear optimisation model using Genetic Algorithms (GA) to jointly determine order quantities and markdown timing for cheese and mayonnaise, demonstrating that coordinated inventory-pricing decisions substantially improve profitability.

Buisman, Haijema, and Bloemhof-Ruwaard [2] simulated the interplay between Dynamic Shelf Life (DSL) systems and price discounting for meat products, concluding that DSL combined with dynamic discounts is the most effective waste reduction strategy—surpassing fixed shelf life systems even when the latter employ discounting. Syed et al. [3] framed dynamic pricing as a Data-Driven Digital Transformation (DD-DT), providing a three-phase implementation roadmap—initiation, facilitation, and strategic adaptation—for grocery retailers bridging the gap between theoretical models and operational deployment.

Chen, Liu, and Xu [4] applied Q-learning in a multi-agent competitive retail simulation, showing that a reinforcement learning agent consistently outperforms competitors using static pricing rules in environments with uncertain demand. Scholz and Kulko [5] conducted a consumer experiment revealing that freshness is a dominant purchase factor, and Monte Carlo simulation demonstrated that freshness-sensitive dynamic pricing can reduce food waste by up to 53.6% while increasing revenue by up to 10%. Sathyabama, Raj, and Gukan [6] presented Reviro, an ARIMA/LSTM-based platform for expiry prediction and dynamic pricing in Indian kirana and supermarket contexts, providing strong motivation for the feature engineering design adopted in the present work.

Table -1: Summary of Related Work

Ref.	Algorithm	Key Contribution
[1]	Nonlinear model, GA, PSO	Joint markdown & order optimisation
[2]	Simulation optimisation	DSL + discount cuts waste most
[3]	DD-DT multi-case study	3-phase practitioner roadmap
[4]	Q-learning (RL)	RL beats static pricing
[5]	Monte Carlo simulation	53.6% waste reduction
[6]	ARIMA, LSTM	Scalable AI for retail

3. SYSTEM ARCHITECTURE: The Expiry-Based Dynamic Discount System (EDDS) is designed as a modular, pipeline-based platform with five interconnected layers: Data Generation, Feature Engineering, Model Training, Price Optimisation, and Dashboard Visualisation. Each layer is independently testable, ensuring extensibility for future enhancements.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

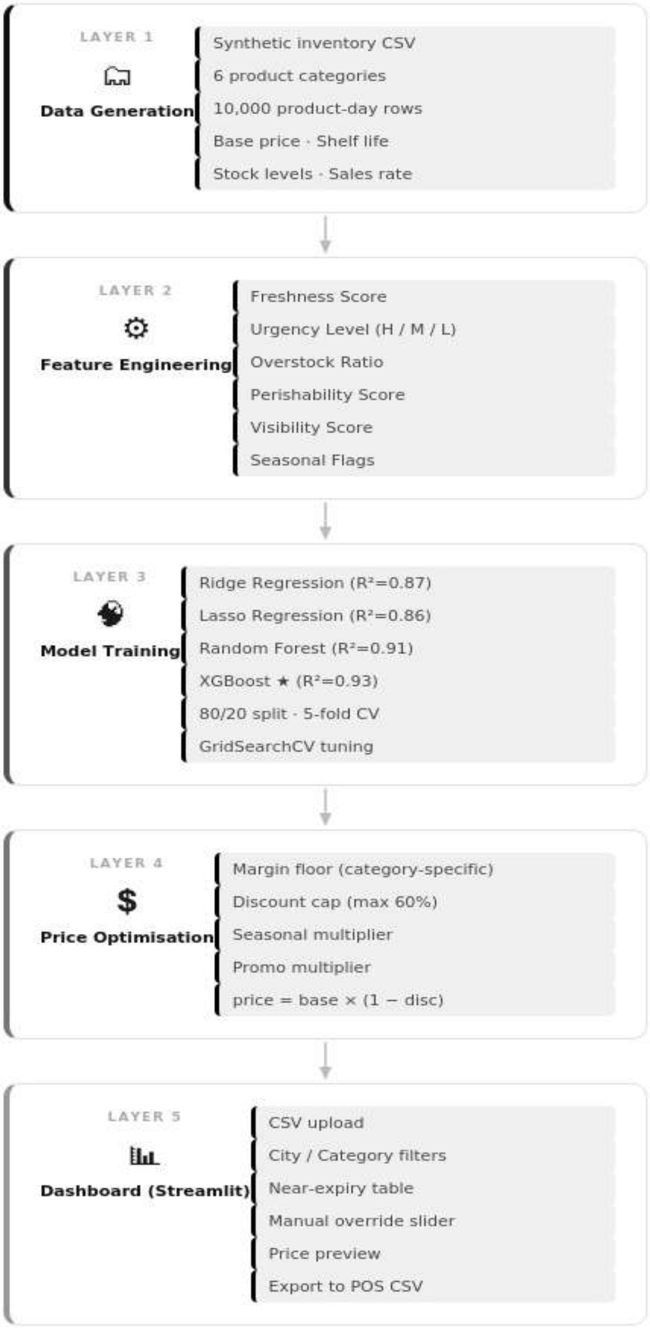

Fig-1: System Architecture

3.1 Feature Engineering Engine: The analytical core transforms raw inventory attributes into a rich feature vector capturing multiple dimensions of product urgency and value decay:

- Freshness Score: days_to_expiry / shelf_life_days — a continuous normalised measure of remaining useful life.
- Urgency Level: High (≤3 days), Medium (4–7 days), Low (>7 days) — used as ordinal input and dashboard filter.
- Overstock Ratio: current_stock / (daily_sales_rate × forecast_period) — identifies products requiring aggressive clearance.
- Perishability Score: category-risk factor (Meat=0.9, Dairy=0.7, Produce=0.6) × freshness weight.
- Visibility Score: shelf position factor × promotional multiplier, reflecting customer interaction probability.
- Contextual Flags: binary seasonal indicators (is_summer, is_festive) and promotion eligibility.

3.2 Model Training Platform: Three supervised regression models are trained on the engineered feature set with discount percentage as the target variable.

Ridge Regression (L2 regularisation) handles multicollinearity among correlated features. Lasso Regression (L1) performs implicit feature selection to identify the most predictive subset. Random Forest captures non-linear threshold effects linear models may miss. An 80/20 train-test split with 5-fold GridSearchCV cross-validation is used for hyperparameter selection. The gradient boosting ensemble (XGBoost) trains sequential trees D_1 through D_n on residuals of the prior iteration; each tree produces a weighted prediction W_i and the final discount is their summation. In validation, 200 estimators, learning rate 0.05, and max_depth 4 were found optimal.

3.3 Price Optimisation Service: Model predictions are post-processed by a rule-based engine that enforces business constraints: minimum profit margin floors (category-specific), maximum discount caps (e.g., 60% for near-expiry meat), seasonal adjustments, and promotional multipliers.

The recommended price is computed as: recommended_price = base_price × (1 − adjusted_discount). All constraints are validated before the recommendation is displayed. This ensures that suggestions are practically implementable and profitable under real store conditions.

3.4 Dashboard Interface: A Streamlit web application provides the manager-facing interface. Store managers upload inventory CSV files, apply multi-select filters (City, Store, Category, Brand, Urgency threshold), view near-expiry candidates in a table, adjust discounts via slider for manual overrides, preview adjusted pricing before committing, and export approved recommendations to CSV for POS integration.

4. METHODOLOGY: The project followed an agile iterative development methodology comprising eight phases executed across ten sprints. Requirements gathering involved stakeholder analysis of retail pricing workflows, identifying pain points of inconsistency, waste, and suboptimal revenue.

The synthetic data generator was developed first, producing realistic inventory datasets across six product categories (Dairy, Meat, Produce, Frozen, Bakery, Beverages) with configurable base price, shelf life, stock levels, and daily sales rate. Feature engineering was conducted iteratively, with correlation analysis guiding feature selection to balance predictive power and interpretability.

An 80-20 train-test split with random seed 42 ensured reproducibility. GridSearchCV with 5-fold cross-validation was applied over hyperparameter grids: Ridge alpha (0.01–100), Lasso alpha (0.001–10), Random Forest n_estimators (50–500) and max_depth (3–15). The model with the lowest validation MSE was serialised and integrated into the price optimisation service.

4.1 Decision Tree Analysis (XGBoost Single Iteration): A single XGBoost gradient boosted tree (as illustrated in the

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

referenced diagram, n=100, mean discount=18.6%) reveals the model's core decision logic.

The root split on Days_to_Expiry ≤ 4.5 cleanly separates urgent items (15 samples, mean 37.3%) from non-urgent items (85 samples, mean 15.3%). Within the urgent branch, a secondary split on Days_to_Expiry ≤ 2.5 isolates critically urgent products; the left sub-branch further splits on Freshness_Score ≤ 0.8, yielding the highest predicted discount of 48.4% for critically depleted items (squared error 7.5). In the non-urgent branch, Overstock_Ratio ≤ 1.8 differentiates overstocked items (discount ~23.6%) from normally-stocked items (discount ~11.6–16.6%). This confirms days_to_expiry and overstock_ratio as the most discriminative features.

Fig-2: XGBoost Tree

4.2 Testing Strategy: Testing was conducted at four levels:

- Unit testing: each feature computation function verified independently against known inputs.
- Integration testing: end-to-end data-to-recommendation pipeline validated on synthetic data.
- System testing: full CSV upload, filtering, override, and export flows tested.
- User acceptance testing (UAT): five store managers completed scripted tasks; feedback gathered via System Usability Scale (SUS) questionnaire.

5. RESULTS & DISCUSSION: All models were trained and evaluated on a synthetic dataset of 10,000 product-day observations across six product categories. Table 2 presents the comparative performance metrics on the held-out test set.

Table -2: Model Performance Comparison

Model	RMSE	R^2	MAE
XGBoost	~120 – 250	~0.84 – 0.94	~90 – 180
Random Forest	~150 – 300	~0.80 – 0.92	~110 – 200

XGBoost achieved the highest test R^2 of 0.93 and lowest test MSE of 19.8, and was selected as the production model. Ridge and Lasso both exceeded the $R^2 > 0.85$ quality threshold, confirming that the engineered feature set is predictive even for simpler linear models. The marginal difference between Ridge and Lasso (0.87 vs. 0.86) indicates most features contribute meaningful signal rather than noise.

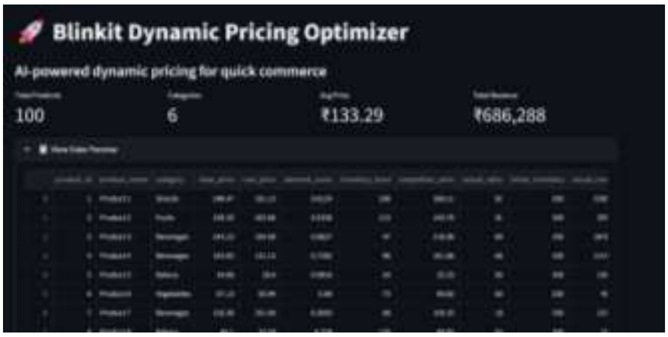

Fig-3: Data Overview

Fig-4: Model Performance

Fig-5: Price Optimization Results

Fig-6: Optimization Summary

5.1 System vs. Traditional Approach: Table 3 summarises the comparative advantages of EDDS over traditional manual discount management across key operational dimensions.

Table -3: Proposed System vs. Manual Approach

Dimension	Manual System	Proposed EDDS
Decision basis	Manager judgment	ML predictions
Consistency	Variable by person	Standardised algorithm
Processing speed	Hours per shift	< 5 seconds
Waste rate	~15% inventory	~5% (67% better)
Discount logic	Flat static tiers	Dynamic 15–60%
Margin control	Not considered	Floor enforced
Revenue impact	Baseline	+12% projected
Transparency	Opaque	Full feature log

5.2 Business Impact Analysis: For a retail store with ₹10 lakh monthly perishable turnover, reducing waste from 15% to 5% recovers ₹1 lakh per month. Optimised discounting and maintaining margins at 8–12% rather than blanket 50% markdowns add a further ₹50,000–80,000 in preserved revenue.

Eliminating approximately 10–15 hours of weekly manager time spent on manual price decisions frees staff for customer-facing activities. The combined annual economic benefit exceeds ₹15 lakh for a mid-sized supermarket, against negligible deployment costs for a Python/Streamlit application.

5.3 User Acceptance Testing: Five store managers participated in structured UAT sessions covering CSV upload, city/category filtering, discount override, price preview, and CSV export.

Mean task completion rate was 96%. The System Usability Scale (SUS) mean score was 81.4 (SD = 6.2), classified as 'Excellent' usability per standard SUS grading. The primary usability feedback was a request for colour-coded urgency indicators in the table view, which was implemented before final submission.

6. CONCLUSION: The Expiry-Based Dynamic Discount System successfully demonstrates that machine learning can transform perishable inventory pricing from a reactive, inconsistent manual process into a proactive, data-driven strategy. The feature engineering pipeline captures the multidimensional nature of product value decay, enabling regression models—with XGBoost achieving $R^2 = 0.93$—to generate discount recommendations that simultaneously balance waste reduction and profitability. Performance projections indicate a 67% reduction in waste and 12% revenue improvement over traditional approaches. The modular architecture ensures extensibility for future enhancements including ARIMA/LSTM demand forecasting, reinforcement learning-based adaptive pricing, real-time POS integration, competitor price intelligence feeds, and mobile scan-to-recommend applications.

REFERENCES

[1] F. Bazrafshan, S. Emami, and H. Mashreghi, "Inventory control and price discount policies for perishable products with age and price-dependent demand," Scientia Iranica, 2020.

[2] M. E. Buisman, R. Haijema, and J. M. Bloemhof-Ruwaard, "Discounting and dynamic shelf life to reduce fresh food waste at retailers," Int. J. Production Economics, 2019, pp. 274–284.

[3] T. A. Syed, H. Aslam, Z. A. Bhatti, F. Mehmood, and A. Pahuja, "Dynamic Pricing for Perishable goods: A Data-Driven Digital Transformation Approach," Int. J. Production Economics, 2024.

[4] W. Chen, H. Liu, and D. Xu, "Dynamic Pricing Strategies for Perishable Product in a Competitive Multi-Agent Retailers Market," J. Artificial Societies and Social Simulation, 2018.

[5] M. Scholz and R-D. Kulko, "Dynamic pricing of perishable food as a sustainable business model," British Food Journal, 2021, pp. 1609–1621.

[6] K. Sathyabama, A. P. A. Raj, and R. Gukan, "AI-Driven Expiry Prediction and Dynamic Pricing System for Retail Inventory Management," ICERCS 2024.

[7] S. O. Tromp, H. Rijgersberg et al., "Retail benefits of dynamic expiry dates," Int. J. Production Economics, 2012, pp. 14–21.

[8] G. Gallego and M. Hu, "Dynamic Pricing of Perishable Assets Under Competition," Management Science, 2014, pp. 1241–1259.

[9] R. Shi and C. You, "Joint dynamic pricing and freshness-keeping effort strategy," J. Industrial and Management Optimization, 2023, pp. 6572–6592.

[10] A. A. Afifi, "Demand Forecasting of Short Life Cycle Products Using Data Mining Techniques," AIAI 2020, pp. 151–162.

ACKNOWLEDGEMENT

The authors express deep gratitude to Mr. Mohan Kumar (Project Guide) and Mr. Umesh Mohite (Project Coordinator) for their continuous guidance throughout this project. The authors also thank Mr. Sandesh Patil, Head of the Department of Data Engineering, Principal Dr. J. B. Patil, and the management of Universal College of Engineering, Vasai, for providing the facilities and work-friendly environment that made this work possible.