

Date Based Inventory Monitoring and Discount Rules with Diagnostic Evaluation of KNN

Syahprian

Politeknik Ganesha Medan, Medan

syahprian29@gmail.com

ABSTRACT

Recorded expiration status and future inventory loss are different decision targets. This study audits a date-based monitoring architecture, a historical K-nearest neighbor (KNN) experiment, and independent markdown rules. Source-document recovery established that the 1,000 simulated labels were deterministic date bands: High at seven days or fewer, Medium at 8–30 days, and Low beyond 30 days. The revised architecture assigns date status directly and retains KNN only as an offline diagnostic. The historical K = 9 confusion matrix gives accuracy 0.735, macro F1 0.502, balanced accuracy 0.488, High recall 0.441, and Medium recall 0.063; 32 of 59 High records were classified Low. These results are not an independently selected-model estimate. A new sensitivity grid evaluated 135 hypothetical combinations of demand response, sell-through, cost, and discount. Discounting improved accounting profit in 60 of 108 discounted scenarios and worsened it in the remainder; these counts describe a designed grid, not real-world success probabilities. Forty black-box cases passed for a new Python reference decision core, without validating the original PHP application. The findings support direct treatment of known dates and conditional economic review of markdowns. Original-data ablation, matched classifier comparisons, renewed validation, and application-level testing remain dependent on the original CSV and source code.

Keywords: expiration monitoring; date-based rules; K-nearest neighbor; economic sensitivity; software verification

INTRODUCTION

Perishable inventory creates a time-dependent decision problem: stock that remains available today may lose its saleable value before demand can absorb it. Effective monitoring therefore requires more than recording quantities. Product age, storage conditions, demand, and the timing of intervention jointly shape the decisions available to a retailer. Kayikci et al. (2022) connect sensor-informed pricing to perishable food waste reduction, while Syed et al. (2024) examine the organizational and data requirements of dynamic pricing. These studies motivate linking inventory information to actions, but they do not imply that every monitoring application must optimize prices.

Expiration-date-based discounting provides a practical intervention before the end of a product's saleable period (Haijema et al., 2026). More advanced approaches jointly learn pricing and ordering policies under uncertain and changing demand (Keskin et al., 2022), or explicitly balance profit and waste in inventory control (Hasiloglu-Ciftciler & Kaya, 2025). Inventory information itself can constrain how frequently retailers update prices (Chehrazhi et al., 2025). Models incorporating preservation investment (Modak et al., 2024) and freshness-keeping effort (Shi & You, 2023) further show that economic outcomes depend on variables beyond calendar proximity to expiration. Consequently, a fixed markdown schedule should be evaluated as a policy implementation, rather than described as an optimal pricing strategy.

A complementary research stream concerns monitoring infrastructure and inventory classification. Nguru and Iriane (2026) describe an IoT system for household stock and expiration-date monitoring. Dewi et al. (2022) and Syahab et al. (2026) apply KNN to sales-related product classification. Such studies establish relevant implementation precedents, but sales popularity and expiration risk are different targets. Classification accuracy obtained for one target cannot be transferred to the other, and monitoring notifications alone do not establish an economically effective intervention policy.



A category defined by a recorded expiration date is an observable state, not a forecast of future spoilage or unsold inventory. If the date is valid and available at review time, the category should be assigned directly. A learned proxy can be studied diagnostically but must demonstrate added value before influencing inspection priorities. Aggregate accuracy can also conceal poor minority-class recognition (Chicco & Jurman, 2020), so the reported metrics must reflect the intended decision and expose class-level errors (Naser & Alavi, 2023).

The motivating context is Koperasi Sinergi Kreasi Mandiri, for which a PHP/MySQL monitoring prototype was described in the source thesis (Syahprian, 2026). Operational historical records were unavailable for the classification experiment, which used 1,000 simulated records. The revision distinguishes previously reported classification summaries, methodological details recovered from the thesis, and newly executed economic and reference-code analyses. Newly written reference code is not represented as the original application.

The revised study asks whether learning is necessary for the documented target, how the historical errors constrain its defensible use, and under which stated economic assumptions markdowns improve or worsen accounting outcomes. Its contribution is a decision audit supported by explicit date semantics, class-level error analysis, reproducible economic scenarios, and executable reference-policy checks. Improved KNN accuracy, user benefit, algorithmic novelty, and actual waste reduction are not claimed.

METHOD

Research Design and System Architecture

The study combines a documented web-prototype design with a secondary audit of its classification summaries and new computational analyses. The original application uses native PHP and MySQL/MariaDB and iterative prototyping, an approach also used in web sales systems (Rinanda et al., 2023). Its executable source and database were unavailable during revision. A Python reference implementation was therefore written to make the recovered policy testable, with an evidence scope separate from the PHP application.

Figure 1 shows the revised operational boundary. Valid dates determine date status, discounts, withdrawal, and date notifications. Invalid dates require correction rather than an inferred safe status. KNN remains in an offline diagnostic branch and cannot suppress date warnings, override withdrawal, or set prices. Merely displaying KNN labels beside discounts is no longer claimed as useful operational integration.

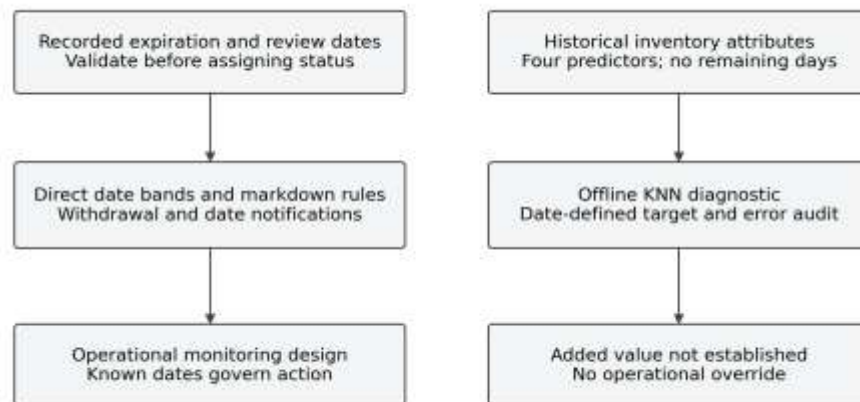


Figure 1. Date-based operational decisions and a separate offline classification audit.

Dataset and Target Definition

The historical experiment contains 1,000 simulated records with 13 attributes. Its four predictors are shelf life, stock, daily sales, and storage condition (Table 1). Conceptually, shelf life describes nominal product duration, stock the quantity exposed to loss, daily sales a demand-velocity proxy, and storage condition an environmental-quality proxy. These motivations are not measured contributions to a date-defined target. Their inclusion does not demonstrate that every feature is necessary or causally relevant.

Table 1. Variables used by the classification and recommendation modules

Variable	Representation	Role
Shelf life	Days	KNN predictor
Stock	Product units	KNN predictor
Daily sales	Units per day	KNN predictor
Storage condition	Good = 1; Fair = 2; Poor = 3	KNN predictor
Remaining days to expiration	Integer day difference	Date-rule input; excluded from KNN
Date-defined status	Low; Medium; High	Classification target

The source thesis defines High when remaining days $r \leq 7$, Medium when $8 \leq r \leq 30$, and Low when $r > 30$ (Syahprian, 2026, Table 3.11). The direct date reference $g(r)$ is this same mapping. Its agreement with labels is therefore exact by construction for valid records following that policy. This is a logical identity, not a newly executed 200-record test. The target is interpreted as date-defined status, not a probability of future spoilage or unsold stock.

The thesis reports zero missing, duplicate, and invalid records, and training extrema of 3–720 days for shelf life, 0–171 for stock, 0–32 for daily sales, and 1–3 for storage codes (Syahprian, 2026, Tables 4.2–4.3). These are recovered documentary observations, not independent checks of the CSV. Generation code, distributions, seed, and predictor dependencies remain unavailable. Recovering ranges and thresholds does not recreate the dataset. Synthetic-data evaluation still requires resemblance and utility evidence (Espinosa & Figueira, 2023; Fonseca & Bacao, 2023; Hernandez et al., 2023).

Table 2 retains the documented class counts; training counts follow by subtraction. The 293 High records include 244 records with $r \leq 0$ and only 49 still-saleable records with 1–7 days remaining. Thus, 83.3% of the full High class was already expired. Active inventory contains 756 records, but active-only test performance cannot be recovered from the confusion matrix. Expired products are withdrawal cases and should bypass learned prioritization.

Table 2. Risk-class distribution across the reported partition

Risk class	All records	Share (%)	Training	Test
High	293	29.3	234	59
Medium	83	8.3	67	16
Low	624	62.4	499	125
Total	1000	100.0	800	200

Preprocessing and Information Boundaries

The thesis reports that missing-value, duplicate, and invalid-value checks found no records requiring removal. Its stratified 80:20 partition contains 800 training and 200 test records. Min–max parameters were reportedly fitted on training records and reused for test transformation in Equation (1). The original split indices and seed are unavailable. Training-only preprocessing is required to prevent evaluation information entering fitting (Apicella et al., 2025; Kapoor & Narayanan, 2023).

$$z_{ij} = \frac{x_{ij} - a_j}{b_j - a_j} \quad (1)$$

Here, x_{ij} is feature j of record i , and a_j and b_j are its training extrema. The recovered ranges are nonconstant. Storage codes Good = 1, Fair = 2, and Poor = 3 become 0, 0.5, and 1 after scaling. Adjacent categories therefore contribute 0.25 to squared Euclidean distance, while the extreme-category contribution is 1.0. This geometry is an assumption, not an empirical finding. Feature importance cannot be inferred from the ranges.

Historical KNN excluded remaining days and tried to reconstruct date status from inventory proxies. That design is retained only to interpret the historical scores. A recorded date available at decision time is legitimate information; excluding it does not create a useful future-risk forecast. The revised operational branch uses $g(r)$. Predicting unsold stock or spoilage would require separately observed future outcomes, a horizon, and earlier predictors; those observations are unavailable.

Historical KNN Comparison and Selection Bias

KNN assigns a class through the labels of nearby training observations, following the nearest-neighbor classification principle (Cover & Hart, 1967). For the four scaled predictors, Euclidean distance is calculated using Equation (2), and the predicted class follows majority voting as expressed in Equation (3).

$$d(z_i, z_\ell) = \sqrt{\sum_{j=1}^4 (z_{ij} - z_{\ell j})^2} \quad (2)$$

$$\hat{y}_i = \arg \max_c \sum_{\ell \in N_K(z_i)} \mathbb{1}(y_\ell = c) \quad (3)$$

$N_K(z_i)$ denotes the K nearest training records and $\mathbb{1}$ the indicator function. The historical experiment compared $K \in \{3, 5, 7, 9\}$ on one 200-record partition. Its voting tie rule and exact implementation remain unspecified; odd K does not prevent three-class ties. The documented results contain no feature ablation, alternative encoding, mixed-data distance, or imbalance-treatment comparison.

The held-out outcomes informed both the historical choice of K and its reported score. The revised paper withdraws the claim that $K = 9$ is an independently validated best configuration. Its score is only the largest observed among inspected candidates, and its confusion matrix is retained for error accounting. Selection bias is not repaired by renaming the partition or recalculating summaries (Cawley & Talbot, 2010). Section 2.9 specifies the required renewed analysis, which has not been executed without the CSV.

Independent Discount and Withdrawal Rules

Let r denote remaining calendar days, calculated as the recorded expiration date minus the review date. The recommendation function $A(r)$ follows Table 3. Products with $r \leq 0$ are assigned withdrawal, including products reaching the date on the review day. This is the prototype’s conservative business rule, rather than a general statement about the legal or safety meaning of all date labels. The policy does not distinguish product categories or date-label types.

Table 3. Deterministic recommendation policy

Remaining days r	Action $A(r)$
$r > 30$	No discount
$15 \leq r \leq 30$	10% discount
$8 \leq r \leq 14$	20% discount
$4 \leq r \leq 7$	40% discount
$1 \leq r \leq 3$	50% discount
$r \leq 0$	Withdraw from sale

The percentages are the original fixed policy, not demand-optimal estimates. Increasing markdowns as expiration approaches does not justify their magnitudes economically. Sections 2.7 and 3.4 add a sensitivity analysis of those magnitudes. The intervals remain mutually exclusive and exhaustive for integer r . Neither KNN output nor a storage proxy can override withdrawal.

The source describes boundary checks but supplies no original per-case execution log. The revision adds a separate reference decision core and executes black-box contract cases against its documented rules. Section 3.5 reports those results with their implementation scope. They must not be attributed retrospectively to the PHP application.

Evaluation Measures and Analytical Reference

Classification is evaluated using accuracy and per-class precision, recall, and F1-score. Macro averages give each risk class equal weight. With $C = 3$ classes, the measures are defined by Equations (4)–(6), using true positives TP_c , false positives FP_c , and false negatives FN_c for class c .

$$\text{Accuracy} = \frac{1}{n} \sum_{i=1}^n 1(y_i = \hat{y}_i) \quad (4)$$

$$P_c = \frac{TP_c}{TP_c + FP_c}; \quad R_c = \frac{TP_c}{TP_c + FN_c} \quad (5)$$

$$F1_c = \frac{2P_c R_c}{P_c + R_c}; \quad F1_{macro} = \frac{1}{3} \sum_{c=1}^3 F1_c \quad (6)$$

Macro precision and macro recall are calculated analogously; macro recall equals balanced accuracy in this multiclass setting. Per-class metrics and the confusion matrix are used to identify overlooked risk classes. Although work advocating Matthews correlation emphasizes the limits of individual summary metrics (Chicco & Jurman, 2023), the present analysis retains the reported multiclass measures and does not treat any single score as sufficient evidence of operational suitability.

The majority reference always predicts Low, the most frequent training class. Its accuracy, balanced accuracy, and macro F1 are derived from the test supports, with zero F1 for unpredicted classes. A second reference, $g(r)$, agrees exactly with the date-defined target by definition and is reported as a semantic identity rather than an executed benchmark. Discount counts cover all 1,000 records, separate from the 200-record classification audit.

Hypothetical Economic Sensitivity Analysis

The new deterministic analysis considers one markdown held constant over a fixed comparison horizon. It does not simulate sequential deployment of every date tier, reconstruct transactions, or estimate consumer response. Assumed opening quantity Q is 100 units, list price p is 100 monetary units, and disposal cost w is 2 per unsold unit. Unit procurement cost c/p is 0.50, 0.70, or 0.90; baseline horizon sell-through q is 0.20,

0.50, or 0.80; and constant price elasticity ϵ is 0.5, 1, or 2. Discount d is 0, 0.10, 0.20, 0.40, or 0.50. The grid contains 135 scenarios, including 27 no-discount references.

Discounted demand is capped by stock, and accounting profit charges procurement on the entire opening quantity, as in Equations (7)–(8). Fractional units represent expected demand. The scenarios omit replenishment, substitution, demand uncertainty, and changing elasticity. These are illustrative assumptions, not estimates from the cooperative.

$$s(d) = \min(Q, Qq(1 - d)^{-\epsilon}) \quad (7)$$

$$\pi(d) = p(1 - d)s(d) - cQ - w[Q - s(d)] \quad (8)$$

Incremental profit is $\Delta\pi(d) = \pi(d) - \pi(0)$, comparing the same horizon and opening stock. Unit gross margin is $p(1 - d) - c$, so nonnegative unit margin requires $d \leq 1 - c/p$. That screen is not a complete clearance decision: procurement is already committed, and selling below its unit cost can still reduce an existing total loss. Absolute profit and incremental change are therefore distinguished. No scenario-grid optimum is substituted automatically for the fixed schedule.

Reference-Core Checks and Application Evidence Boundary

A Python 3.12 reference core implements strict YYYY-MM-DD parsing, calendar subtraction, date bands, fixed discounts, withdrawal, a date-notification function, and a below-cost flag. Forty black-box cases call public functions and compare outputs or exceptions with independently enumerated policy expectations. They cover both sides of every markdown transition, expired and same-day products, invalid dates, leap years, year transitions, invalid input types, and notification boundaries. The case log supplies inputs through named cases, expected outputs, actual outputs, and pass/fail flags.

The reference notification threshold follows the source dashboard attention boundary: $r \leq 0$ produces withdrawal notice, 1–14 days produces attention, and $r > 14$ produces no date notice. This tests reference-policy behavior only. Original authentication, database persistence, imports, dashboard aggregation, and notification delivery were not executed. Supplementary local calculation timing excludes HTTP and database work and is not web response time.

Required Empirical Validation Protocol

The supplementary data-dependent script requires the original CSV. It specifies a renewed stratified 80:20 internal split with seed 20261019, five stratified development folds, and selection by development macro F1 before holdout evaluation. Fitting and scaling remain inside each training fold. KNN, Decision Tree, and Logistic Regression share the same folds and final partition. This would neither reconstruct the undocumented original split nor provide external validation. No study-data result from this protocol is reported here.

KNN comparisons include ordinal Euclidean distance, Euclidean distance with a one-hot-equivalent storage block, and bounded mixed dissimilarity based on Gower's principle (Gower, 1971). The latter averages training-range-normalized numerical differences and categorical mismatch, capping held-out numerical contributions at one. One-hot block scaling makes any storage-category mismatch contribute one to squared Euclidean distance. Single-feature and leave-one-feature-out conditions evaluate predictive relevance, not causation. Ordinary versus inverse-frequency class voting tests imbalance treatment. Each KNN condition searches $K = 3, 5, 7$, or 9 using development folds only.

Decision Tree searches depths 3, 5, or unrestricted, minimum leaf sizes 1, 5, or 10, and ordinary or balanced class weights. Logistic Regression searches $C = 0.1, 1$, or 10 with ordinary or balanced weights. Reports include macro F1, balanced accuracy, High/Medium recall, High-to-Low errors, confusion matrices, predictions, and split indices. The script also offers offline inspection-budget retrieval; that is not a user acceptance study. A future-risk model requires a separately measured future outcome before model selection. These analyses remain pending until the original data are supplied.

RESULTS

Historical Neighborhood Sensitivity

Table 4 retains the historical scores and analytical majority reference. Accuracy rises from 69.0% at $K = 3$ to 73.5% at $K = 9$, or nine additional correct records in the inspected partition. Macro F1 rises from 0.475 to 0.502. This is retrospective neighborhood sensitivity, not evidence that $K = 9$ generalizes best. None of the candidates was selected by training-only cross-validation.

Table 4. Historical KNN sensitivity and analytical majority reference

Configuration	Accuracy (%)	Macro precision	Macro recall	Macro F1
$K = 3$	69.0	0.539	0.467	0.475
$K = 5$	70.5	0.546	0.472	0.482
$K = 7$	72.5	0.661	0.482	0.496
$K = 9$	73.5	0.678	0.488	0.502
Always Low	62.5	—	0.333	0.256

$K = 9$ exceeds majority prediction by 11.0 accuracy percentage points and approximately 0.246 macro F1 in the historical partition. In contrast, $g(r)$ reproduces the documented target exactly because the labels were defined from those dates. A fitted method cannot improve upon that definitional agreement. This semantic reference has not been executed against individual records because the CSV is unavailable. Exceeding majority prediction therefore does not establish value beyond known dates.

Class-Specific Errors

Table 5 contains the $K = 9$ confusion matrix. The diagonal sums to 147 correct predictions, consistent with 73.5% accuracy. The model predicts Low risk for 166 of 200 observations, compared with 125 actual Low-risk records. Medium risk receives only two predictions. This strong concentration of predictions in Low risk explains why aggregate accuracy and balanced accuracy lead to different assessments.

Table 5. Confusion matrix for $K = 9$

Actual class	Predicted High	Predicted Medium	Predicted Low	Total
High	26	1	32	59
Medium	1	1	14	16
Low	5	0	120	125
Total	32	2	166	200

Table 6 presents metrics recalculated directly from the integer confusion matrix. All displayed values use conventional rounding to three decimal places; consequently, $1/16$ is reported as 0.063 and $26/32$ as 0.813. Macro values remain consistent with the original aggregate results. The Low-risk F1-score is 0.825, whereas the Medium-risk F1-score is 0.111. High-risk precision of 0.813 indicates that High predictions are relatively selective, but recall of 0.441 indicates that most actual High-risk records are missed.

Table 6. Class-level evaluation recalculated from the confusion matrix

Risk class	Precision	Recall	F1-score	Support
High	0.813	0.441	0.571	59
Medium	0.500	0.063	0.111	16
Low	0.723	0.960	0.825	125
Macro average	0.678	0.488	0.502	200

The principal directional error is High-to-Low classification: 32 of 59 High records (54.2%) receive Low predictions. Including the High-to-Medium error, 33 of 59 are missed as High. Medium recall rests on just one correct record out of 16. These outcomes do not support using KNN as a gatekeeper for date alerts. The unavailable record-level data prevent separating expired from still-active products within these errors. No improved minority recall is claimed in this revision.

Distribution of Date-Based Recommendations

Table 7 reports the action distribution obtained by applying the date policy to all 1,000 simulated records. The 132 discounted records represent 13.2% of the dataset; 24.4% receive withdrawal. Together, discounting or withdrawal applies to 376 records, or 37.6%. These frequencies describe the simulated inventory and the chosen thresholds. They do not measure conversion, avoided waste, or realized revenue.

Table 7. Recommendation distribution across all simulated records

Action	Remaining days	Records	Share (%)
No discount	> 30	624	62.4
10% discount	15–30	54	5.4
20% discount	8–14	29	2.9
40% discount	4–7	29	2.9
50% discount	1–3	20	2.0
Withdraw	≤ 0	244	24.4
Total	All ranges	1000	100.0

The action counts sum to 1,000. Of the 293 High-labeled records, 244 are withdrawal cases, while the 49 still-saleable High records occupy the 40% and 50% tiers. This agreement follows from shared date boundaries and checks aggregate consistency, not row-level execution or realized sales. The policy count is not 1,000 independent classification predictions.

Economic Sensitivity of Fixed Markdown Levels

Table 8 reports a slice of the new grid with baseline sell-through $q = 0.50$ and procurement cost $c/p = 0.70$. At elasticity 0.5, every positive markdown reduces profit despite higher sales volume. At elasticity 1, improvements arise from assumed disposal savings until inventory saturates. At elasticity 2, a 20% discount improves profit by 1,306.25 monetary units, whereas 50% improves it by only 100.00. These are conditional calculations, not measured customer responses.

Table 8. Hypothetical profit change from no discount at $q = 0.50$ and $c/p = 0.70$

Discount	$\Delta\pi$ at $\varepsilon = 0.5$	$\Delta\pi$ at $\varepsilon = 1$	$\Delta\pi$ at $\varepsilon = 2$	Unit gross margin
0%	0.00	0.00	0.00	30
10%	-251.17	11.11	579.01	20
20%	-516.06	25.00	1,306.25	10
40%	-1,097.92	66.67	1,100.00	-10
50%	-1,423.04	100.00	100.00	-20

Note. Profit changes are in monetary units; the final column is monetary units per sold unit. Negative values indicate lower profit or a below-cost unit margin, respectively.

For this slice, no-discount profit is $-2,100.00$ monetary units. Positive $\Delta\pi$ does not imply positive absolute profit: at $\varepsilon = 2$ and $d = 0.20$, profit is still -793.75 . For cost ratios 0.50, 0.70, and 0.90, the nonnegative unit-margin discount limits are 50%, 30%, and 10%, respectively. Thus, the original 40% and 50% tiers sell below unit procurement cost at $c/p = 0.70$. At $c/p = 0.90$, every scenario in the grid retains negative absolute profit.

Across 108 discounted scenarios, 60 have positive $\Delta\pi$ and 48 negative $\Delta\pi$. Procurement cost shifts absolute profit but cancels from the incremental comparison at a fixed opening inventory, so each comparison repeats at three cost ratios. Equivalently, 20 of 36 unique elasticity–sell-through–discount combinations improve profit. These designed-grid counts are not probabilities of policy success. Every scenario is reproduced in the supplementary CSV. No optimal sequence of time-dependent markdowns is inferred.

Executed Reference-Core Contract Results

All 40 cases passed in the newly supplied reference core. Table 9 summarizes categories, and the supplementary log provides individual expected and observed outputs. Withdrawal is tested at -1 and 0 days; positive-day boundaries include 1, 3, 4, 7, 8, 14, 15, 30, and 31. Invalid or absent dates raise errors rather than producing a status. These results show agreement of this reference code with the specified policy, not reliability of the original web deployment.

Table 9. Black-box cases executed on the new Python reference decision core

Case category	Executed	Passed
Calendar arithmetic	4	4
Date validation	7	7
Discount boundary	11	11
Margin flag	3	3
Notification	5	5
Risk boundary	6	6
Type validation	4	4
Total	40	40

Remaining Web-Application Evidence

The source describes login, dashboard summaries, product CRUD and CSV import, classification, discounts, and notifications. Interface documentation does not establish end-to-end correctness. These

original-source functions were not tested during revision. Their evaluation requires executable PHP source, database schema, launch instructions, and test accounts, with cases for access control, persistence, failed imports, boundary dates, and notification display or delivery. User acceptance and HTTP/database response-time results remain unavailable.

DISCUSSION

Date Status Is Not a Future-Risk Outcome

The recovered labeling rule resolves the central semantic issue: the target is a deterministic date category. Known dates determine it exactly without learning. None of the four inventory proxies has demonstrated incremental value for reproducing it. Their possible relevance to future unsold stock or spoilage is a different hypothesis requiring future outcomes. The revision does not replace the old target with invented spoilage or unsold-stock labels.

Date monitoring remains useful as record management and policy execution, but the previous claim for useful KNN integration is withdrawn. Operational status and withdrawal should be calculated directly. A learned classifier should be introduced only after defining a distinct decision target and establishing incremental user benefit. This boundary prevents poor historical High recall from suppressing date-based attention.

Interpreting Imbalance and Minority-Class Failure

The majority comparison explains how 73.5% accuracy coexists with weak High and Medium detection. The direct date reference shows why exceeding majority is insufficient for the current target. High recall, Medium recall, and High-to-Low errors remain central diagnostics. The four features, ordinal encoding, Euclidean distance, and $K = 9$ have not been empirically justified through ablation or renewed validation.

The Medium support of 16 also limits the stability of its observed recall. Imbalance may contribute, but feature–target mismatch and overlap remain competing explanations. Neighborhood undersampling addresses such interactions (Kumar et al., 2024); SMOTE alters the minority distribution (Elreedy et al., 2024), and hybrid methods add filtering (Sun et al., 2024). These are not reported as completed improvements. The supplied data-dependent script instead tests balanced voting and matched alternatives once the actual CSV is supplied.

Economic and Operational Implications

The new scenarios supply an economic basis for questioning the markdown magnitudes. Greater sell-through can coexist with lower profit, and reducing an accounting loss does not imply positive total profit or positive unit margin. Dynamic pricing research models demand and inventory interactions explicitly (Keskin et al., 2022; Hasiloglu-Cifteler & Kaya, 2025). The present grid is a narrower sensitivity calculation with assumed response values.

A practical review requires product costs, margins, demand response, and the meaning of each date label. The reference below-cost flag prompts review; it does not imply that below-cost clearance is always irrational when procurement is already committed. Likewise, passing reference tests shows policy agreement only within their scope, not original-application reliability, user acceptance, or business effectiveness.

Validity Limits and Further Evaluation

The original CSV and generator remain unavailable. Recovered thresholds, ranges, and reported quality counts correct documentation gaps but do not recover sampling distributions, seed, or real-world resemblance. Already-expired products in High also prevent interpreting the historical results as active-stock prioritization. Original-data ablation, encoding and distance comparisons, matched classifier

benchmarks, and renewed validation have not been executed. These remain substantive requirements, not problems solved by the new economic or code checks.

The original test partition informed model choice and cannot serve as an untouched final evaluation. The supplied development protocol would restrict selection to training folds and then evaluate a renewed internal holdout, while external or prospective validation would still be preferable. Algorithms must share partitions, with any group or temporal dependence reflected in the design. A new seed alone does not create external evidence.

Economic assumptions are hypothetical and uncalibrated from transactions. They quantify neither actual waste reduction nor an optimal markdown sequence. Original-application black-box testing and a measured workflow comparison with and without a learned prioritizer also remain outstanding. Staff time, overlooked active-stock cases, and action quality should be compared against date-only monitoring before claiming benefit. Known dates remain authoritative while those claims are unverified.

CONCLUSION

The revised analysis identifies the target as date-defined status and places direct rules at the operational boundary. Historical KNN errors do not support using that classifier to control warnings or withdrawal. Newly executed scenarios show that 10%–50% discounts can improve or worsen accounting outcomes under explicit response assumptions, while 40 passing reference-core cases supply reproducible policy checks. These additions do not establish optimal prices or validate the original PHP application. Completing the empirical revision requires the actual dataset and generator for matched ablation and classifier evaluation, and the executable application for end-to-end and user-workflow testing.

DATA AND CODE AVAILABILITY

The supplementary package contains the reference decision core, 40 executed test cases and log, 135 hypothetical economic scenarios, analysis scripts, and a data-dependent benchmarking script. The original 1,000-row CSV, generator, original split indices, and PHP/MySQL source are not included because they were unavailable. The package reproduces the added analyses but does not reproduce or independently validate the historical classification experiment.

REFERENCES

- Apicella, A., Isgrò, F., & Prevete, R. (2025). Don't push the button! Exploring data leakage risks in machine learning and transfer learning. *Artificial Intelligence Review*, 58, Article 339. <https://doi.org/10.1007/s10462-025-11326-3>
- Cawley, G. C., & Talbot, N. L. C. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11, 2079–2107. <https://www.jmlr.org/papers/v11/cawley10a.html>
- Chehraz, N., Sanders, R. E., & Stamatopoulos, I. (2025). Inventory information frictions explain price rigidity in perishable groceries. *Marketing Science*, 44(2), 411–436. <https://doi.org/10.1287/mksc.2023.0473>
- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21, Article 6. <https://doi.org/10.1186/s12864-019-6413-7>
- Chicco, D., & Jurman, G. (2023). The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification. *BioData Mining*, 16, Article 4. <https://doi.org/10.1186/s13040-023-00322-4>
- Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27. <https://doi.org/10.1109/TIT.1967.1053964>



- Dewi, S. P., Nurwati, & Rahayu, E. (2022). Penerapan data mining untuk prediksi penjualan produk terlaris menggunakan metode K-nearest neighbor [Applying data mining to predict best-selling products using the K-nearest neighbor method]. *Building of Informatics, Technology and Science (BITS)*, 3(4), 639–648. <https://doi.org/10.47065/bits.v3i4.1408>
- Elreedy, D., Atiya, A. F., & Kamalov, F. (2024). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. *Machine Learning*, 113(7), 4903–4923. <https://doi.org/10.1007/s10994-022-06296-4>
- Espinosa, E., & Figueira, A. (2023). On the quality of synthetic generated tabular data. *Mathematics*, 11(15), Article 3278. <https://doi.org/10.3390/math11153278>
- Fonseca, J., & Bacao, F. (2023). Tabular and latent space synthetic data generation: A literature review. *Journal of Big Data*, 10, Article 115. <https://doi.org/10.1186/s40537-023-00792-7>
- Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, 27(4), 857–871. <https://doi.org/10.2307/2528823>
- Hajjema, R., Duijvestijn, L., Akkerman, R., & Crujssen, F. (2026). Dynamic expiration date-based discounting of fresh food products. *International Journal of Production Economics*, 292, Article 109824. <https://doi.org/10.1016/j.ijpe.2025.109824>
- Hasiloglu-Ciftciler, M., & Kaya, O. (2025). Dynamic inventory control and pricing strategies for perishable products considering both profit and waste. *Computers & Operations Research*, 181, Article 107103. <https://doi.org/10.1016/j.cor.2025.107103>
- Hernandez, M., Epelde, G., Alberdi, A., Cilla, R., & Rankin, D. (2023). Synthetic tabular data evaluation in the health domain covering resemblance, utility, and privacy dimensions. *Methods of Information in Medicine*, 62(Suppl. 1), e19–e38. <https://doi.org/10.1055/s-0042-1760247>
- Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), Article 100804. <https://doi.org/10.1016/j.patter.2023.100804>
- Kayikci, Y., Demir, S., Mangla, S. K., Subramanian, N., & Koc, B. (2022). Data-driven optimal dynamic pricing strategy for reducing perishable food waste at retailers. *Journal of Cleaner Production*, 344, Article 131068. <https://doi.org/10.1016/j.jclepro.2022.131068>
- Keskin, N. B., Li, Y., & Song, J.-S. (2022). Data-driven dynamic pricing and ordering with perishable inventory in a changing environment. *Management Science*, 68(3), 1938–1958. <https://doi.org/10.1287/mnsc.2021.4011>
- Kumar, A., Singh, D., & Yadav, R. S. (2024). Entropy and improved k-nearest neighbor search based under-sampling (ENU) method to handle class overlap in imbalanced datasets. *Concurrency and Computation: Practice and Experience*, 36(2), Article e7894. <https://doi.org/10.1002/cpe.7894>
- Modak, I., Bardhan, S., & Giri, B. C. (2024). Dynamic pricing and preservation investment strategy for perishable products under quality, price and greenness dependent demand. *Journal of Industrial and Management Optimization*, 20(7), 2260–2281. <https://doi.org/10.3934/jimo.2024001>
- Naser, M. Z., & Alavi, A. H. (2023). Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences. *Architecture, Structures and Construction*, 3, 499–517. <https://doi.org/10.1007/s44150-021-00015-8>
- Nguru, F. M., & Iriane, G. R. (2026). Perancangan sistem monitoring stok dan tanggal kadaluarsa berbasis IoT pada kulkas rumah tangga [Design of an IoT-based stock and expiration-date monitoring system for household refrigerators]. *Jurnal Publikasi Manajemen Informatika*, 5(1), 239–246. <https://doi.org/10.55606/jupumi.v5i1.4696>
- Rinanda, A. R., Ondri, O. R., & Sulistiyono. (2023). Sistem informasi penjualan berbasis web dengan metode prototype [A web-based sales information system using the prototype method]. *BINER: Jurnal Ilmu Komputer, Teknik dan Multimedia*, 1(2), 290–299. <https://www.journal.mediapublikasi.id/index.php/Biner/article/view/2957>

- Shi, R., & You, C. (2023). Joint dynamic pricing and freshness-keeping effort strategy for perishable products with price-, freshness-, and stock-dependent demand. *Journal of Industrial and Management Optimization*, 19(9), 6572–6592. <https://doi.org/10.3934/jimo.2022228>
- Sun, P., Wang, Z., Jia, L., & Xu, Z. (2024). SMOTE-kTLNN: A hybrid re-sampling method based on SMOTE and a two-layer nearest neighbor classifier. *Expert Systems with Applications*, 238, Article 121848. <https://doi.org/10.1016/j.eswa.2023.121848>
- Syahab, M. A., Rambe, A., Harahap, B., & Syahputra, D. (2026). Penerapan algoritma K-nearest neighbor untuk klasifikasi produk terlaris pada usaha mikro [Applying the K-nearest neighbor algorithm to classify best-selling products in microenterprises]. *Jurnal Minfo Polgan*, 15(2), 1705–1712. <https://doi.org/10.33395/jmp.v15i2.16298>
- Syahrman. (2026). *Sistem monitoring kedaluwarsa produk menggunakan algoritma K-nearest neighbor untuk rekomendasi diskon pada Koperasi Sinergi Kreasi Mandiri* [Unpublished diploma thesis]. Politeknik Ganesha Medan.
- Syed, T. A., Aslam, H., Bhatti, Z. A., Mehmood, F., & Pahuja, A. (2024). Dynamic pricing for perishable goods: A data-driven digital transformation approach. *International Journal of Production Economics*, 277, Article 109405. <https://doi.org/10.1016/j.ijpe.2024.109405>