**Electronic Integrated Computer Algorithm Journal**
Vol. 3, No. 2, April 2026, pp 41~48

ISSN: 3031-0350, DOI: 10.62123/enigma.v3i2.145

# Implementing Collaborative Filtering for E-Commerce Product Personalization Using a Rapid Application Development Approach

Mullah Cadre Giawa[1*], Egi Al Fansyah[2], Dwi Fahira Alsyah[2], Abdul Hakim Satria Nusantara[3], Daffa Shidqi Thamrin[2], Ratandi Ahmad Fauzan[2]

[1] *Information Technology Study Program, Faculty of Computing, President University, address, Jababeka Education Park, Jl. Ki Hajar Dewantara, Kota Jababeka, Cikarang Utara, Bekasi, 17550, Jawa Barat,* Indonesia.
[2] *Master of Information System Study Program, School of Industrial Engineering, Telkom University, Main Campus (Bandung Campus), Jl. Telekomunikasi no. 1, Bandung 40257, West Java*, Indonesia.
[3] *Information System Study Program, School of Industrial Engineering, Telkom University, Main Campus (Bandung Campus), Jl. Telekomunikasi no. 1, Bandung 40257, West Java*, Indonesia.

**DOI :** 10.62123/enigma.v3i2.145

**Received** : March 16, 2026
**Revised** : April 17, 2026
**Accepted** : April 22, 2026

***Keywords:***
*E-commerce, Recommender System, Cosine Similarity, RAD, MAE, RMSE*

**ABSTRACT**

The rapid expansion of e-commerce has increased the difficulty of guiding users toward relevant products, particularly as catalogs grow and user preferences become more diverse. This paper presents an end-to-end implementation of a personalized product recommendation feature using a memory-based collaborative filtering approach integrated into an e-commerce platform. Development followed a Rapid Application Development (RAD) workflow, enabling iterative prototyping, integration, and testing of the recommendation module within the operational system. Recommendations were generated using a K-Nearest Neighbors method with cosine-based similarity to identify related items from user interaction histories and to produce Top-N product suggestions in the storefront interface. Model evaluation employed a transactional dataset commonly used for recommender experiments, which was refined from 541,909 records (8 attributes) to 406,829 interaction-focused records (CustomerID, Description, Quantity). Performance was assessed using MAE, RMSE, and F1-score, yielding values of 0.6, 0.8, and 0.6, respectively. The results indicate that collaborative filtering can provide moderately accurate and relevant recommendations when interaction history is available, while also exposing practical limitations for users with limited transactions, reflecting a cold-start constraint. These findings suggest that RAD-supported integration of collaborative filtering is feasible for e-commerce personalization and provides a baseline for further enhancement.

## 1. INTRODUCTION

The expansion of e-commerce has made product catalogs larger and customer behavior more diverse. As platforms scale, users are often presented with thousands of alternatives, and finding products that match personal preferences can become time-consuming. This situation has encouraged many online stores to adopt recommendation systems that can narrow choices, surface relevant items, and support a smoother shopping experience that may translate into higher engagement and loyalty. E-commerce platforms increasingly operate under conditions of abundant choice, where users face large and continuously expanding catalogs. In such environments, personalization is often treated not merely as a convenience feature but as a mechanism that can influence engagement and commercial outcomes. Evidence from large-scale platform analyses suggests that stronger personalization capability tends to align with improvements in conversion-related indicators and longer-term customer value [1]. At the same time, consumer-facing outcomes appear to depend on more than algorithmic relevance. Studies focusing on user psychology indicate that trust, transparency, and enjoyment can meaningfully shape whether personalization translates into satisfaction, loyalty, or purchase intention [2][3].

Despite these benefits, recommender systems in e-commerce repeatedly encounter persistent technical constraints. Interaction data may be sparse for many users, preference signals can be unevenly distributed across the catalog, and new users or items often lack sufficient history to support reliable prediction. Survey and review-based syntheses consistently emphasize sparsity and cold-start as recurring bottlenecks that remain difficult to eliminate fully [4][5]. In parallel, preference is rarely static. Work on context-aware and sequential modeling suggests that users' interests shift over time, which can reduce the effectiveness of approaches that rely on a fixed snapshot of historical behavior [6][7]. These observations imply that a practical e-commerce recommender should be engineered with both data limitations and operational constraints in mind. Personalization, however, is not a straightforward feature to implement. User preferences can shift over time, interaction data tend to be sparse for many customers, and the platform must still deliver recommendations quickly and consistently inside the application. Collaborative Filtering (CF) is frequently used

*Corresponding Author Email: egialfansyah@student.telkomuniversity.ac.id

in this domain because it infers preferences from collective patterns in user behavior, such as purchases, ratings, or other interactions. Rather than relying solely on product attributes, CF uses similarities among users or items to estimate what a user is likely to find relevant, which makes it attractive for practical e-commerce settings.

Recent research has explored increasingly sophisticated solutions, including deep learning architectures and review-driven feature extraction, which can improve personalization by leveraging auxiliary signals from text or learned representations [8][9][10]. However, complex models may introduce higher computational cost and greater implementation overhead, which can be challenging when recommendation features must be delivered quickly and iterated within a live product. This motivates the focus of the present study: rather than optimizing a purely offline model, the work emphasizes an end-to-end implementation of collaborative filtering within an e-commerce system using a Rapid Application Development workflow. By positioning collaborative filtering as an interpretable baseline that can be deployed and refined quickly, the study aims to clarify how recommendation functionality can be integrated into production-facing workflows while still being evaluated rigorously through standard metrics [4][5].

Even when the algorithm is well understood, operationalizing CF within an e-commerce system raises additional design and engineering questions. The recommendation logic has to be integrated with the transaction and interaction data pipeline, connected to the storefront interface, and tested as a working feature, not only as an offline model. For that reason, this study adopts a Rapid Application Development (RAD) approach to support iterative prototyping and fast refinement, moving from requirements to implementation and evaluation in a way that aligns with the pace of typical e-commerce development.

This paper therefore focuses on implementing a collaborative filtering recommendation module within an e-commerce platform developed through RAD, then assessing its performance using standard evaluation metrics, including MAE and RMSE, with F1-score used to reflect recommendation relevance. The study is guided by two research questions: (1) How can collaborative filtering be implemented within an e-commerce platform to generate personalized product recommendations from user interaction history? (2) How effective is the implemented collaborative filtering feature in terms of recommendation accuracy and relevance based on evaluation metrics? By presenting an end-to-end implementation and a quantitative evaluation, the work offers practical evidence on how CF can be deployed for product personalization, while also indicating limitations that merit further development, particularly for users with limited interaction history and for improving adaptiveness as preferences evolve. The main contribution of this study lies in the end-to-end implementation of an item-based collaborative filtering recommendation module within a real e-commerce platform using a Rapid Application Development workflow. Unlike studies that focus mainly on offline model optimization, this work emphasizes practical system integration, user-facing deployment, and quantitative evaluation of recommendation performance in an operational setting.

## 2. LITERATURE REVIEW

### 2.1 Collaborative filtering as a practical baseline in e-commerce

Collaborative filtering remains widely discussed as a foundation for personalization because it infers relevance from collective behavioral patterns rather than requiring dense product attributes. Overviews of recommender-system research describe memory-based and model-based collaborative filtering as enduring baselines, partly because they are comparatively interpretable and easier to operationalize than many deep models [5]. In practice, similarity-based approaches often rely on measures such as cosine similarity, which can be applied directly to sparse interaction vectors. Empirical discussions comparing user-based and item-based configurations suggest that item-based strategies can be more stable under high sparsity, since item relationships may be estimated more reliably than user neighborhoods in large, uneven datasets [11].

### 2.2 Addressing sparsity and cold-start through auxiliary signals and hybridization

Although collaborative filtering is attractive, the literature repeatedly frames sparsity and cold-start as structural limitations, particularly for new users or low-frequency buyers [4][5]. A common response is to incorporate auxiliary information that can enrich the interaction signal. Review-driven approaches, for example, extract contextual entities or sentiments from text and then integrate them into prediction models to reduce reliance on ratings alone [8][11]. Other work highlights real-time feature adaptation based on incoming feedback, suggesting that personalization quality may improve when the system updates preference representations dynamically instead of relying on static history[10]. Beyond text, hybrid frameworks combining clustering and association rules have been used to group users or products more effectively and generate actionable bundles, particularly when interaction matrices are sparse [12][13]. Together, these studies imply that baseline collaborative filtering can be strengthened when combined with signals that increase matrix density or make user intent more explicit.

### 2.3 Personalization outcomes depend on user trust and recommendation presentation

Several papers indicate that accurate ranking alone may not guarantee impact if users do not perceive recommendations as credible or well-framed. Consumer research suggests that personalization can foster loyalty indirectly, often mediated by trust and hedonic enjoyment, implying that system designers may need to consider interface and communication strategies alongside algorithmic logic [1][14]. Related findings indicate that transparency can function as a prerequisite for trust, and trust may be a key mechanism through which personalization affects purchase intentions[2]. Experimental work further suggests that how recommendations are presented, such as message framing aligned with user regulatory focus or goal importance, can influence engagement and click-through behavior [15][16]. These perspectives support the view that a deployable recommender system should be evaluated not only on predictive accuracy but also on how recommendations are delivered in the user experience.

Evaluation in recommender-system research typically relies on accuracy-oriented and ranking-oriented metrics (for example, MAE, RMSE, precision, recall, and F1), which remain common across empirical implementations [11][12]. However, there is also an argument that offline accuracy can miss system value, especially when business objectives are central. Some work proposes business-aware evaluation approaches, such as profit-oriented metrics, to distinguish models that perform similarly on technical measures but differ in real impact [17]. Complementary perspectives also emphasize multi-criteria selection of algorithms that accounts for computational resources and business constraints, rather than choosing purely on a single metric [14][18]. This suggests that evaluation choices should align with the intended deployment context, especially for e-commerce systems operating under latency and scalability constraints.

## 3. RESEARCH METHODS

### 3.1 Research Design and Development Approach (RAD)

This study follows a software-oriented research design where the main output is a working recommendation feature embedded in an e-commerce application. The development process adopts Rapid Application Development (RAD) because it supports fast prototyping and iterative refinement, which fits the nature of e-commerce systems where features need to be integrated, tested, and adjusted in short cycles. In this work, RAD was operationalized through the following sequence of activities: (1) planning and requirements, (2) user-oriented design, (3) rapid prototyping, (4) feedback and refinement, (5) integration and testing, (6) evaluation and analysis, and (7) finalization and documentation. While RAD is often summarized into broader phases, these activities reflect how the project was executed in practice, from early requirement definition to model evaluation and reporting.

**Table 1.** RAD Activities Used in This Study

| RAD Activity | Main Output in This Study |
|---|---|
| Planning and Requirements | Feature scope, functional needs, data requirements for recommendations |
| User Design | Recommendation UI placement and interaction design (where recommendations appear and how users interact) |
| Rapid Prototyping | Initial CF model integration and early recommendation outputs |
| Feedback and Refinement | Iterative changes based on system behavior and usability considerations |
| Integration and Testing | Deployment into the e-commerce platform and functional validation |
| Evaluation and Analysis | Quantitative evaluation using MAE, RMSE, and F1-score |
| Finalization and Documentation | Final implementation and write-up for publication |

This study adopts a software-implementation research orientation where the primary artifact is a functional recommendation feature embedded within an operational e-commerce workflow. A Rapid Application Development approach is suitable for this objective because it supports iterative prototyping, short feedback cycles, and incremental refinement, which are commonly required when recommendation logic must be integrated into user-facing interfaces and data pipelines. The method aligns with the broader observation in the literature that system effectiveness is shaped not only by the learning algorithm but also by implementation constraints such as data availability, computational cost, and integration design [5][14].

From an algorithmic perspective, the work positions memory-based collaborative filtering as a deliberately pragmatic baseline. Prior studies that combine collaborative filtering with sentiment or review features still acknowledge that sparsity can make user-based neighborhoods unstable at scale, which motivates item-oriented similarity computation for more reliable retrieval under sparse matrices [11]. This baseline orientation also supports comparability with more complex approaches. Sequence and deep-learning recommenders may outperform classic collaborative filtering in dynamic settings, but they often introduce additional training complexity and reduced interpretability, which can be less compatible with rapid iteration in early-stage deployment [7][9]. By contrast, cosine-based similarity with a KNN neighbor search provides a transparent mechanism for producing Top-N recommendations that can be integrated quickly and tested end-to-end.

### 3.2 System Context and Functional Scope

The recommendation feature is integrated into an e-commerce system with standard shopping flows (product browsing, cart, checkout, ordering, payment), plus an administration area for managing platform content. The overall interaction scope is reflected through the system use-case model (Admin, Visitor, User, and the recommendation component as a system process).

**Table 2.** Functional Requirements for the Recommendation-Enabled E-commerce System

| ID | Requirement |
|---|---|
| FR1 | Collect and store customer interaction data (purchases, preferences, shopping behavior). |
| FR2 | Analyze interaction data to identify patterns among customers with similar behavior. |
| FR3 | Generate personalized product recommendations using Collaborative Filtering based on user preferences and buying history. |
| FR4 | Validate recommendation outputs by assessing whether predictions align with user preference signals. |
| FR5 | Present recommendations in a clear and visually usable format (for example: widget on the site, related items on product pages). |
| FR6 | Authenticate users at login and enforce access based on user roles/permissions. |

*Corresponding Author Email: egialfansyah@student.telkomuniversity.ac.id

### 3.3 Data Sources and Database Structure

The recommendation logic relies on behavioral data captured inside the e-commerce platform database. Interaction signals used by the model are derived from user activity recorded in transactional tables (for example, purchase-related records and review information). The database design organizes key entities such as user, product, review, and cart/order-related tables, which together provide the minimum structure required to connect users with products through observed interactions.

A simplified view of the data relationships is shown in the database design diagram (recommended to include in the paper as a figure). The schema indicates how user actions (such as cart items or reviews) can be mapped to user–product pairs that later become the input to collaborative filtering.

### 3.4 Data Preparation and Train–Test Split

Model building begins by extracting user and product records from the platform database, together with interaction logs that link users to products. These records are transformed into a user–item interaction matrix:

- Rows represent products.
- Columns represent users.
- Cell values represent the interaction strength for a given user–product pair (implemented as purchase frequency derived from cart/order item logs).
- If a user has no interaction history for a product, the value is set to 0.

For evaluation, the collected interaction data are divided into training and testing subsets using an 80:20 split. The training portion supports similarity learning and prediction generation, while the test portion is used to estimate how well predicted preferences align with held-out observations.

Data preparation is framed around constructing an interaction representation that reflects observable user behavior, since collaborative filtering effectiveness depends heavily on the quality and density of interaction signals [4][5]. Where interaction histories are limited, the literature suggests two broad enhancement directions: enriching the interaction matrix using auxiliary signals such as review-derived features, or combining collaborative filtering with clustering or association rules to reduce sparsity effects [8][19][20]. In this paper, these strategies can be positioned as extensions, while the implemented baseline focuses on interaction-driven similarity to maintain methodological clarity and RAD feasibility.

### 3.5 Collaborative Filtering Model and Recommendation Generation

This study implements memory-based, item-oriented Collaborative Filtering using a K-Nearest Neighbors (KNN) approach with cosine distance as the similarity basis. Practically, similarity is computed by fitting a nearest-neighbor model over product vectors derived from the interaction matrix. Cosine similarity formulation is
Cosine similarity between two vectors A and B is expressed as:

$$\text{cosine\_sim}(A,B) = \frac{A\,.\,B}{||A||\,||B||} \tag{1}$$

In the implementation, the KNN model uses cosine distance. Similarity is then interpreted as: sim=1−cosine_distance

The recommender uses a nearest-neighbor model configured with:

- similarity metric: cosine (via cosine distance)
- search strategy: brute force
- neighborhood size: $k$ = 3

For each target user, the system predicts preference scores for products the user has not interacted with. Prediction is computed as a weighted aggregation over neighbor products, where weights follow the similarity values. Items are then ranked, and the system returns the Top-N products with the highest predicted scores as personalized recommendations.

### 3.6 Tools, Platform, and Implementation Environment

The application and recommendation module were developed using Python and integrated into a Django-based e-commerce system. Supporting libraries used for data processing and modeling include pandas, NumPy, and scikit-learn.

**Table 3.** Software Stack

| Component | Used In This Study |
|---|---|
| Operating System | Windows 11 |
| Development Tools | Visual Studio Code |
| Backend Language | Python |
| Web Framework | Django |
| Payment Sandbox/Integration | PayPal Developer |
| Modeling Libraries | pandas, NumPy, scikit-learn |
| Diagram and Documentation Tools | Microsoft Visio (2019), Microsoft Word (2019) |
| Browser Testing | Google Chrome, Microsoft Edge |

### 3.7 Evaluation Metrics

The model is evaluated using a mix of prediction-accuracy metrics and a relevance-oriented metric:

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- F1-score summarizes precision and recall in a single value

To support objective testing, evaluation can be performed using publicly available “standard” e-commerce data containing identifiers and transactional attributes (such as customer identifiers, invoice numbers, and quantities), then comparing predicted preferences against observed signals in the test portion of the split. Evaluation follows a dual rationale. First, standard predictive metrics (MAE and RMSE) remain widely used to quantify error behavior in recommender systems and allow comparison with matrix-factorization and optimization-focused studies[21][22][23]. Second, relevance-oriented assessment through F1-style summarization is consistent with Top-N evaluation practices found in hybrid and sparsity-robust models [24][25][26]. At the same time, prior work argues that offline accuracy can diverge from business impact, motivating the discussion of evaluation limitations and the potential value of business-aware measures such as profit-oriented metrics or cost-sensitive selection frameworks in future iterations [27][28][29]. This framing keeps the current evaluation method defensible while acknowledging what a production-grade assessment could add.

## 4. RESULTS AND DISCUSSIONS

### 4.1 System Testing Outcome and Feature Verification

After the recommendation module was integrated into the e-commerce platform, system-level testing was conducted to verify that core user flows and recommendation-related behavior operated as expected. The application was tested in a standard client environment (Windows 11 64-bit, Microsoft Edge v113.0.1774.35, 16 GB RAM, laptop GF63 Thin 11SC). The test scope covered essential storefront features such as authentication and basic user interactions, then focused on the recommendation feature as the primary contribution of this work.

From an implementation perspective, the results indicate that the recommendation module behaves consistently with the logic of collaborative filtering. Because the model relies on historical user–item interactions, recommendation availability depends on whether a user has recorded transactions. This behavior is not simply a UI choice, it reflects a structural dependency on interaction history, which aligns with the theoretical limitations typically associated with collaborative filtering in cold-start conditions.The first step was to identify and model the main business processes in billet management. This process begins with the receipt of cast billets, quality inspection, transfer to the stacker system, stacking based on specific criteria, and transferring to the storage or further production area. To document this process systematically, a flowchart was created to illustrate all stages of activity in the Billet Stacker Rail system.

### 4.2 Recommendation Behavior in the E-Commerce Interface

The platform presents a dedicated “Recommended Products” section on the homepage. Two user states were observed during testing. First, when a user has not yet made any transaction, the recommendation area remains empty, even though featured products are still displayed normally. This outcome suggests that the system intentionally avoids producing recommendations when the interaction signal is insufficient. In practical terms, the system treats new or inactive users as having no reliable basis for similarity computation, which is consistent with collaborative filtering assumptions.

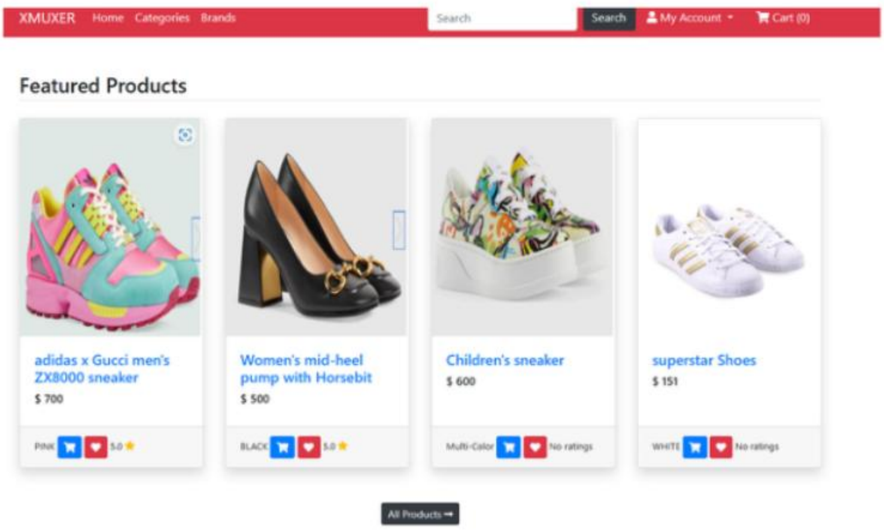


**Figure 1.** Home Page Products

Second, once a user has completed transactions, the homepage displays a populated recommendation list. The recommended items appear as a ranked set of products under the “Recommended Products” section, indicating that user history successfully triggers the model pipeline and returns candidate items for personalization. Together, these two states demonstrate that the system’s personalization output is driven by interaction history rather than static product attributes, which matches the collaborative filtering framing described earlier.

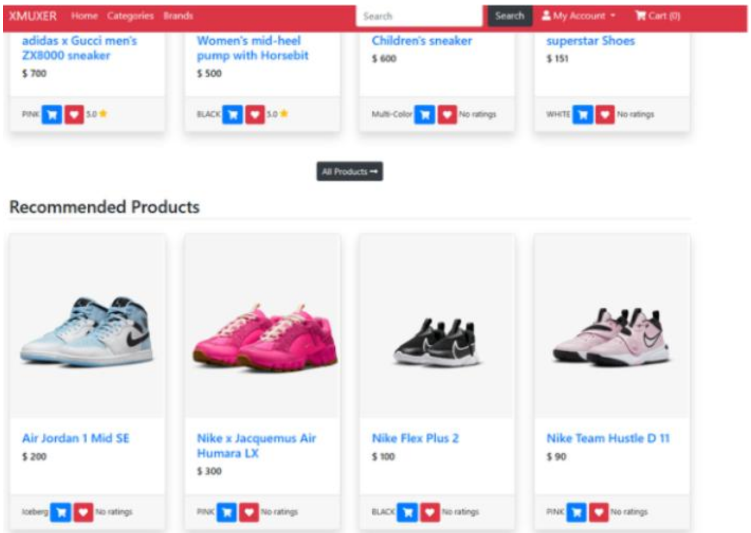


**Figure 2.** Recommendation Product

### 4.3 Evaluation Dataset and Preprocessing Results

To evaluate recommendation quality more objectively, the system was tested using a publicly available transactional dataset commonly used for recommendation experiments. The dataset contains customer identifiers and purchase-related attributes such as invoice number, product identifiers, quantity, time of purchase, unit price, and country. In its original form, the dataset includes 541,909 records across 8 columns.

Before modeling, the data were refined to focus on the interaction signals needed for collaborative filtering. After preprocessing, the dataset used for recommendation computation consisted of 406,829 records and was reduced to three primary fields: CustomerID, Description, and Quantity. This reduction supports the construction of a user–item interaction representation that is compatible with neighbor-based collaborative filtering while removing attributes that are not directly required for similarity computation in the implemented approach.

**Table 4.** Dataset Summary Before and After Preprocessing

| Dataset Stage | Records | Columns | Main Fields |
|---|---|---|---|
| Original dataset | 541,909 | 8 | InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country |
| After preprocessing | 406,829 | 3 | CustomerID, Description, Quantity |

This preprocessing outcome is relevant for interpretation of results: by focusing on purchase quantity and product descriptions as the item representation, the evaluation emphasizes behavior-based personalization, which fits the design goals of this study.

### 4.4 Quantitative Performance of the Collaborative Filtering Model

Model performance was assessed using a combination of prediction-error metrics and a relevance-oriented metric. The evaluation results reported for the implemented collaborative filtering model are:

- MAE: 0.6
- RMSE: 0.8
- F1-score: 0.6

**Table 5.** Recommendation Performance Metrics

| Metric | Value | Interpretation (brief) |
|---|---|---|
| **MAE** | 0.6 | Average absolute prediction error, smaller values indicate closer predictions |
| **RMSE** | 0.8 | Penalizes larger errors more strongly than MAE, useful for spotting larger deviations |
| **F1-score** | 0.6 | Balances precision and recall, higher values indicate more relevant Top-N recommendations |

Taken together, these values suggest that the model provides moderately accurate predictions while maintaining a reasonable balance between recommendation precision and coverage. The RMSE being higher than MAE is consistent with the expectation that some predictions deviate more substantially from observed behavior, because RMSE amplifies larger errors. The F1-score indicates that relevance is present but not yet strong enough to claim highly optimized retrieval quality, which leaves room for parameter tuning, alternative neighbor selection strategies, or hybridization.

### 4.5 Discussion: Implications and Practical Limitations

The reported MAE (0.6) and RMSE (0.8) indicate that the model achieves moderate predictive performance rather than high accuracy. The RMSE value being higher than MAE suggests that some recommendation errors are relatively large, meaning that the system still struggles in certain cases, particularly under sparse interaction conditions. Likewise, the F1-score of 0.6 indicates that the Top-N recommendations are reasonably relevant, but not yet strong enough to be considered highly optimized. Therefore, the current model should be interpreted as a practical baseline that is effective enough for initial deployment, while still leaving substantial room for improvement. Given the system behavior observed in the interface and the evaluation metrics, the findings tend to support two conclusions aligned with the research questions.

For the first research question, the results demonstrate that collaborative filtering can be operationalized as a real feature in an e-commerce platform developed through a RAD workflow. The recommendation section is integrated into the user journey and responds dynamically to user interaction history, which indicates that the model is not only implemented offline but also deployed as part of the transactional system. For the second research question, evaluation outcomes indicate a moderate level of effectiveness. The error metrics suggest the model is capable of producing reasonable preference estimates, while the F1-score shows that relevance remains imperfect. This pattern is consistent with known collaborative filtering constraints, especially sparsity and cold-start effects. The interface results illustrate this directly: users without transaction history do not receive recommendations, which improves safety against random suggestions, but also limits personalization for new users.

From a practical standpoint, these results suggest that collaborative filtering is a viable baseline for e-commerce personalization when interaction history is available. Future improvements could focus on alleviating cold-start cases and enhancing relevance, for example by introducing content-based signals, applying hybrid recommendation logic, or refining the interaction representation and evaluation strategy. In practical use, the recommendation module can help users find relevant products more efficiently by reducing the effort needed to browse large product catalogs. For the platform, this may support a more personalized shopping experience and potentially improve user engagement, especially for customers with established transaction histories.

## 5. CONCLUSSION

This study developed and implemented an item-based collaborative filtering recommendation module within an e-commerce platform using a Rapid Application Development approach. The main contribution of this work is not only the use of collaborative filtering itself, but its end-to-end integration into a functional e-commerce system, including recommendation generation, interface presentation, and system-level deployment. Regarding the first research question, the study shows that collaborative filtering can be implemented in an operational e-commerce environment by transforming transaction histories into a user–item interaction matrix, computing item similarity using cosine-based KNN, and presenting Top-N recommendations in a dedicated “Recommended Products” section. This confirms that collaborative filtering can function as a practical personalization feature when sufficient user interaction data are available.

Regarding the second research question, the implemented system demonstrates moderate effectiveness based on the reported evaluation results (MAE = 0.6, RMSE = 0.8, F1-score = 0.6). These values suggest that the model is able to generate reasonably relevant recommendations, although prediction errors and recommendation quality are not yet optimal. In practical terms, the system can help users discover relevant products more efficiently and support a more personalized shopping experience, especially for returning users with prior transaction histories.

However, the study also identifies important limitations. Recommendation quality depends heavily on the availability of interaction data, and the system is unable to provide personalized suggestions for users with little or no transaction history, reflecting the cold-start problem. In addition, the reported performance should be interpreted as a baseline rather than a highly optimized result, indicating the need for further refinement. Future work may improve coverage and effectiveness by incorporating additional behavioral signals, exploring hybrid recommendation approaches, and conducting broader evaluations under more diverse usage conditions. This study has several limitations. First, the recommendation mechanism relies mainly on transaction history, making it less effective for new users or users with very limited interactions. Second, the evaluation is based on offline metrics, which may not fully capture real business impact or user satisfaction in live environments. Third, the model uses a relatively simple memory-based collaborative filtering configuration, so the results should be interpreted as a deployable baseline rather than a fully optimized recommendation strategy.

## REFERENCES


[1] S. J. Khamdamov *et al.*, “The Impact of AI and Machine Learning on E commerce Personalization,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jul. 2025, pp. 115–128. doi: 10.1145/3726122.3726142.

[2] V. Chkoniya, “The Social Dynamics of Trust in the Digital Economy: A Quantitative Analysis of Personalization and Transparency in European E-Commerce,” *European Journal of Social Science Education and Research*, vol. 12, no. 3, pp. 81–96, Jul. 2025, doi: 10.26417/w73zrh06.

[3] K. D. Long, T. H. Yen, and D. T. Huong, "Driving Online Green Product Loyalty Through Website Personalization and Hedonic Quality Enjoyment: Evidence from Vietnamese E-Commerce Consumers," *International Journal of Sustainable Development and Planning*, vol. 20, no. 6, pp. 2575–2588, Jun. 2025, doi: 10.18280/ijsdp.200625.

[4] D. Wu and X. Li, "A Systematic Literature Review of Financial Product Recommendation Systems," Mar. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/info16030196.

[5] M. A. Hodovychenko and A. A. Gorbatenko, "Recommender systems: models, challenges and opportunities," 2023, *Odesa Polytechnic National University*. doi: 10.15276/hait.06.2023.20.

[6] S. Y. Jeong and Y. K. Kim, "Deep Learning-Based Context-Aware Recommender System Considering Change in Preference," *Electronics (Switzerland)*, vol. 12, no. 10, May 2023, doi: 10.3390/electronics12102337.

[7] L. Yu and L. Yan, "Tourism Product Recommendation Based on LSTM Model," in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 1228–1234. doi: 10.1016/j.procs.2025.05.164.

[8] R. Madani and A. Ez-Zahout, "A Review-based Context-Aware Recommender Systems: Using Custom NER and Factorization Machines." [Online]. Available: www.ijacsa.thesai.org

[9] B. R. Reddy and R. L. Kumar, "An E-Commerce Based Personalized Health Product Recommendation System Using CNN-Bi-LSTM Model," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 6, pp. 398–410, 2023, doi: 10.22266/ijies2023.1231.33.

[10] N. Zareena and T. R. Balaga, "Enhancing e-commerce personalization with review-based adaptive feature matching: a real-time approach," *IAES International Journal of Artificial Intelligence*, vol. 14, no. 3, pp. 2178–2184, Jun. 2025, doi: 10.11591/ijai.v14.i3.pp2178-2184.

[11] R. Thomas and J. R. Jeba, "A novel framework for an intelligent deep learning based product recommendation system using sentiment analysis (SA)," *Automatika*, vol. 65, no. 2, pp. 410–424, 2024, doi: 10.1080/00051144.2023.2295148.

[12] R. Renukadevi, M. Ramalingam, K. Sathishkumar, E. Boopathi Kumar, S. Janarthanam, and A. A. Kholmanovich, "An Improved Collaborative User Product Recommendation System Using Computational Intelligence with Association Rules," 2024.

[13] C. Udokwu, R. Zimmermann, F. Darbanian, T. Obinwanne, and P. Brandtner, "Design and Implementation of a Product Recommendation System with Association and Clustering Algorithms," in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 512–520. doi: 10.1016/j.procs.2023.01.319.

[14] E. Pawełek-Lubera, M. Przyborowski, D. Ślęzak, and A. Wasilewski, "Multi-criteria selection of data clustering methods for e-commerce personalization," *Appl. Soft Comput.*, vol. 182, Oct. 2025, doi: 10.1016/j.asoc.2025.113559.

[15] J. Zhang, M. S. Balaji, J. Luo, S. Jha, and Y. K. Dwivedi, "It is a match! The effect of regulatory fit on new products recommendations," *Psychol. Mark.*, vol. 41, no. 8, pp. 1882–1899, Aug. 2024, doi: 10.1002/mar.22016.

[16] L. Chen, S. Wang, K. Ma, and Y. Wu, "Effect of product recommendation type and focal goal importance on consumer purchase intention," *South African Journal of Business Management*, vol. 55, no. 1, 2024, doi: 10.4102/sajbm.v55i1.4263.

[17] R. Henriques and L. Pinto, "A novel evaluation framework for recommender systems in big data environments," *Expert Syst. Appl.*, vol. 231, Nov. 2023, doi: 10.1016/j.eswa.2023.120659.

[18] R. Mehta, S. Mishra, and S. Saha, "Onto Proximality in Non Negative Matrix Factorization for Recommender Systems," *IEEE Access*, vol. 13, pp. 67476–67487, 2025, doi: 10.1109/ACCESS.2025.3557945.

[19] E. Lex and M. Schedl, "Psychology-informed Recommender Systems: A Human-Centric Perspective on Recommender Systems," in *CHIIR 2022 - Proceedings of the 2022 Conference on Human Information Interaction and Retrieval*, Association for Computing Machinery, Inc, Mar. 2022, pp. 367–368. doi: 10.1145/3498366.3505841.

[20] A. D. Starke and M. Lee, "Unifying Recommender Systems and Conversational User Interfaces," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jul. 2022. doi: 10.1145/3543829.3544524.

[21] F. Maulana, H. Fajri, M. F. Safitra, and M. Lubis, "Unmasking log4j's Vulnerability: Protecting Systems against Exploitation through Ethical Hacking and Cyberlaw Perspectives," in *2023 9th International Conference on Computer and Communication Engineering (ICCCE)*, IEEE, Aug. 2023, pp. 311–316. doi: 10.1109/ICCCE58854.2023.10246082.

[22] T. Syafiera, M. Lubis, R. W. Witjaksono, and H. D. Anggana, "The Means of Engagement (MOE) Model of the Agreement towards the Enterprise Resource Planning (ERP) Implementation," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, IEEE, Oct. 2019, pp. 1–6. doi: 10.1109/ICIC47613.2019.8985900.

[23] M. Lubis, H. H. R. Ananza, and F. D. Suryoputro, "Analysis and Design of Policy and Standard Operating Procedure (SOP) for Information Technology in the Communication and Information Services Department," in *2020 6th International Conference on Interactive Digital Media (ICIDM)*, IEEE, Dec. 2020, pp. 1–7. doi: 10.1109/ICIDM51048.2020.9339659.

[24] S. Nurindasari *et al.*, *Keselamatan Kerja Dan Kesehatan Mental*. Yayasan Kita Menulis, 2025.

[25] M. F. Safitra, M. I. Alhari, D. P. Putri, M. Lubis, H. Fakhrurroja, and V. Satria Praditha, "Metaverse Trend: Definition, Application, Opportunities, Law, and Ethics," in *2023 IEEE International Conference on Computing (ICOCO)*, 2023, pp. 160–165. doi: 10.1109/ICOCO59262.2023.10397864.

[26] M. and L. A. R. and A. M. I. Safitra Muhammad Fakhrul and Lubis, "The Need for Energy-Efficient Networks: A Review of Green Communication Systems and Network Architectures," in *Intelligent Sustainable Systems*, D. S. and M. D. and J. A. Nagar Atulya K. and Jat, Ed., Singapore: Springer Nature Singapore, 2024, pp. 127–136.

[27] Y. A. Prasetyo and M. Lubis, "Smart City Architecture Development Methodology (SCADM): A Meta-Analysis Using SOA-EA and SoS Approach," Apr. 01, 2020, *SAGE Publications Inc.* doi: 10.1177/2158244020919528.

[28] A. R. Ahlan and M. Lubis, "Information security awareness in university: Maintaining learnability, performance and adaptability through roles of responsibility," in *2011 7th International Conference on Information Assurance and Security (IAS)*, IEEE, Dec. 2011, pp. 246–250. doi: 10.1109/ISIAS.2011.6122827.

[29] M. Lubis, M. Kartiwi, and S. Zulhuda, "Privacy and personal data protection in electronic voting: Factors and measures," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 15, no. 1, pp. 512–521, Mar. 2017, doi: 10.12928/TELKOMNIKA.v15i1.3804.