



ECOMMERCE PRODUCT RECOMMENDATION

Mrs. Vhandkar P. S¹, Rupesh Salve²

¹ Faculty Guide & Supervisor | ² Dattakala Group of Institutions, India

Corresponding author: rupeshsalve289@gmail.com

ARTICLE ID IJREK-2026-00165	ARTICLE TYPE Research Article	RESEARCH AREA Computer Science	PUBLICATION Volume 1, Issue Issue 1 (2026)
RECEIVED 25 October 2026	REVISED 27 October 2026	ACCEPTED 26 October 2026	PUBLISHED 28 October 2026

Abstract

E-commerce platforms present customers with large and frequently changing product catalogues. Finding relevant products can require substantial browsing, and generic rankings may not reflect an individual shopper's interests, current needs, or budget. This paper proposes an e-commerce product recommendation system that combines product-content similarity with behavioural signals to create personalized, context-aware product suggestions. The design includes catalogue preparation, interaction-event processing, candidate generation, ranking, availability filtering, and feedback collection. A hybrid approach is proposed because content-based recommendations can serve new or lesspopular products, while collaborative signals can identify patterns across users with similar preferences. The system also includes practical safeguards such as privacy-aware data handling, diversity controls, explanations, and monitoring for popularity bias. A prototype could be developed using a web interface, an API service, a product database, and a recommendation module. Evaluation is planned using ranking metrics and usability measures; no experimental performance is claimed in this conceptual paper. The proposed design aims to improve product discovery while keeping recommendations understandable and under customer control.

Keywords: E-commerce; Product Recommendation; Recommender System; Personalization; Collaborative Filtering; Content-Based Filtering; Hybrid Ranking; User Experience

2. PROBLEM STATEMENT AND OBJECTIVES

Many online stores use simple popularity lists or static category suggestions. These methods may repeatedly show the same products, overlook niche interests, or provide weak recommendations for new users and newly listed products. The problem addressed here is how to design a recommendation workflow that combines multiple signals while remaining responsive, explainable, and privacy-conscious. 2.1 Objectives • Design a modular recommendation pipeline suitable for an e-commerce application. • Combine content-based similarity with collaborative and session-level behavioural signals. • Apply business constraints such as stock availability, price range, and category eligibility before display. • Provide concise explanations for recommendations and allow users to adjust preferences. • Define a reproducible evaluation plan using offline ranking metrics and user-centred feedback. 2.2 Scope The proposed system focuses on product discovery rather than payment processing, order fulfilment, or inventory forecasting. It assumes the store can provide product metadata and, where permitted, interaction events such as product views, clicks, wish-list additions, cart additions, and purchases. Users should be able to browse without creating a detailed personal profile; anonymous session signals may be used only under the platform's applicable privacy and consent rules.

3. RELATED WORK

Recommender systems are commonly grouped into content-based, collaborative-filtering, and hybrid methods. Content-based methods compare product attributes with a user's interests, making them useful when item descriptions are available. Collaborative filtering uses interaction patterns across users or items and can uncover preferences not obvious from metadata, but it may struggle with sparse data and cold-start cases. Hybrid systems combine the strengths of these approaches and can switch or blend signals depending on available evidence. Classic recommender-system literature discusses the trade-offs among accuracy, coverage, diversity, novelty, and user trust. In commercial settings, recommendations must also account for catalogue changes, stock status, latency, and business rules. A system optimized solely for clicks may over-promote popular products or short-term curiosity.

Therefore, this proposal treats ranking quality and responsible presentation as joint design requirements. The intended contribution is a clearly documented prototype architecture and evaluation plan for a small-to-medium e-commerce catalogue. A final empirical paper should expand the literature review with verified, relevant studies and compare methods using a stated dataset and experimental protocol. E-commerce Product Recommendation System | 3

4. PROPOSED SYSTEM ARCHITECTURE

The system is organized into six logical modules: customer interface, event and profile processing, product catalogue, candidate generation, ranking and constraint checking, and presentation with feedback. Modules can initially run in a single application and later be separated into services if scale requires it.

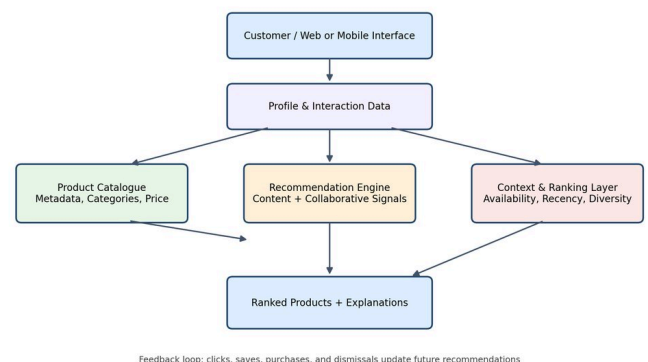


Figure 1. Proposed architecture of the hybrid e-commerce product recommendation system.

4.1 MODULE DESCRIPTION

TABLE I

Module	Responsibility	Example Data
Customer interface	Collects search, browsing and preference actions; displays ranked items.	Session ID, selected filters
Catalogue service	Stores product details and current availability.	Category, brand, price, stock
Event processor	Validates and aggregates consented interaction events.	View, click, cart, purchase
Candidate generator	Retrieves plausible items using content and collaborative signals.	Similar items, co-viewed items
Ranking and rules	Combines scores and applies constraints, diversity and freshness.	Score, eligibility, rank
Feedback and monitoring	Captures useful feedback and tracks quality indicators.	Dismissal, conversion, latency

4.2 DATA FLOW

A request begins when a customer opens a store page. The system reads only the context needed for that request, retrieves candidate products, removes unavailable or ineligible items, ranks the remaining candidates, and returns a compact list with optional explanations. Interaction feedback is stored according to retention and consent rules and can be used to refresh user or item signals. E-commerce Product Recommendation System | 4

5.1 PRODUCT AND INTERACTION FEATURES

Each product can be represented by structured fields such as category, brand, price band, product description, tags, rating summary, and availability. Text fields can be transformed into vectors using TF-IDF or text embeddings. Numeric features should be normalized where needed, and missing values should be handled explicitly. Behavioural events can be assigned different strengths; for example, a purchase may indicate stronger interest than a brief view, although event weights should be validated rather than assumed universally correct.

5.2 CANDIDATE GENERATION

Candidate generation creates a manageable set of products for ranking. Content-based retrieval can find items similar to products viewed or saved by the customer. Collaborative retrieval can find items frequently interacted with by users who share interaction patterns, or items commonly viewed together. Popularity-based or editorial candidates can provide a fallback for new visitors, but should not dominate every recommendation slot.

5.3 HYBRID SCORING

A simple hybrid score may be defined as: $\text{Score}(u,i) = \alpha \cdot \text{Content}(u,i) + \beta \cdot \text{Collaborative}(u,i) + \gamma \cdot \text{Context}(u,i)$ Here, u represents the current user or session, i represents a product, and α , β , and γ are tunable weights. Context may include the current category, selected price range, recency, and stock status. The scores should be calibrated to comparable ranges before blending. When collaborative evidence is unavailable, the system can increase reliance on content and context signals.

5.4 RANKING CONSTRAINTS AND EXPLANATIONS

TABLE II

Signal	Potential Benefit	Risk / Control
Product attributes	Works for new products with complete metadata.	Poor or misleading metadata; validate catalogue fields.
Collaborative patterns	Captures shared preferences and co-interest.	Sparse interactions; use fallbacks and monitor coverage.
Session context	Adapts to current shopping intent.	Overreacting to accidental clicks; limit short-term influence.
Business constraints	Avoids unavailable or ineligible items.	Commercial bias; disclose sponsored placement separately.

6. SYSTEM WORKFLOW AND IMPLEMENTATION

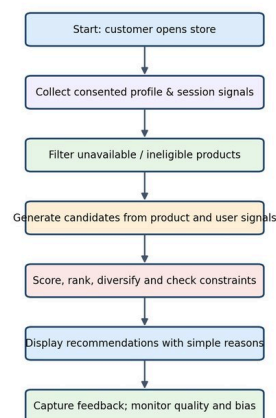


Figure 2. End-to-end recommendation workflow.

6.1 SUGGESTED TECHNOLOGY STACK

TABLE III

Layer	Suggested Option	Purpose
Frontend	React or a lightweight HTML/CSS/JavaScript interface	Product pages, filters, recommendation cards
API layer	Python FastAPI or Node.js/Express	Serves recommendation requests and catalogue APIs
Data store	PostgreSQL or MongoDB	Stores products, permitted events, and user preferences
Recommendation logic	Python, Pandas, scikit-learn	Feature preparation, similarity and ranking
Cache / analytics	Redis or scheduled batch jobs (optional)	Reduces repeated computation; aggregates events
Deployment	Container or managed application hosting	Configuration, monitoring, and versioned releases

6.2 IMPLEMENTATION STEPS

- Create a clean product catalogue schema and validate category, price, description, and stock fields.
- Define the allowed interaction events, consent rules, retention period, and access controls.
- Build a baseline recommendation such as category match or popularity with availability filtering.
- Add content-based similarity, then collaborative signals when sufficient interaction data exists.
- Implement ranking, diversity controls, explanations, and a fallback for new users or products.
- Expose an API endpoint and integrate recommendation cards into relevant pages.
- Log technical metrics and run offline tests before any limited usability pilot.

6.3 EXAMPLE API CONTRACT

A recommendation endpoint could accept a session or user identifier, page context, selected filters, and a requested result count. The response should return product identifiers, display-safe product details, ranking position, and a short explanation label. It should not expose internal model features or private interaction histories to the client. E-commerce Product Recommendation System | 6

7.1 SUGGESTED DATA ENTITIES

TABLE IV						
Entity	Key Fields				Purpose	
y Product	y product_id, title, category, brand, price, description, stock_status				p Catalogue and eligibility checks	
InteractionEvent	event_id, session/user event_type, timestamp	key,	product_id,	Permitted signals	behavioural	
Preference	yp p user/session key, category preferences, price range, updated_at				Explicit or preferences	summarized
RecommendationLog	request_id, product_id, rank, model_version, timestamp	Reproducibility diagnostics			and	
Feedback	p request_id, product_id, feedback_type, timestamp	Useful / not useful feedback				

7.2 ILLUSTRATIVE RECOMMENDATION SCENARIOS

TABLE V		
Scenario	System Response	Reason
New visitor	y p Show category trends or curated items, with clear filters.	No reliable personal history is available.
Repeated browsing	category Prioritize relevant items within that category.	Recent session signals suggest current intent.
p g y g Viewed a specific product	g y Show similar products and compatible alternatives.	g gg Content attributes provide an interpretable match.
Returning customer	Blend longer-term interests with recent activity.	Balances stable preference and current needs.
g Item out of stock	g y Exclude it and suggest available alternatives.	p Availability is a hard constraint.

7.3 PRIVACY AND SECURITY

The design should minimize collected data, use pseudonymous identifiers where practical, protect API endpoints, and restrict access to interaction logs. Personalization should not require sensitive personal attributes. Retention limits, deletion procedures, and an option to reduce or disable personalization should be defined. Security testing should cover injection, unauthorized access, insecure direct object references, and accidental leakage of customer histories.

7.4 FAIRNESS AND RESPONSIBLE RANKING

A recommender can reinforce popularity concentration, reduce product exposure for small sellers, or create repetitive results. Monitoring should include catalogue coverage and diversity in addition to clicks and purchases. Sponsored results, if present, should be labeled distinctly from organic recommendations. Ranking changes should be versioned so that unexpected shifts can be traced and rolled back. E-commerce Product Recommendation System | 7

8. EVALUATION PLAN AND DISCUSSION

Evaluation should compare the proposed hybrid system with a clearly defined baseline, such as popularity ranking or category-based recommendations. A time-aware train/test split is preferable when interaction timestamps exist, because it reduces the risk of using future behaviour to predict earlier events. Duplicate events and data leakage must be checked before metrics are calculated.

8.1 OFFLINE METRICS

TABLE VI		
Metric	What It Measures	Interpretation
Precision@K	Share of top-K recommendations that are relevant.	p Higher values indicate fewer irrelevant items in the displayed set.
Recall@K	Share of relevant items recovered in the top K.	Measures how much relevant content is surfaced.
NDCG@K	p Ranking quality with higher weight for relevant items near the top.	Useful when relevance has graded levels.
Coverage	Share of eligible catalogue items that can be recommended.	Helps reveal over-concentration on a small set of items.
Diversity	Difference or variety among items in a recommendation list.	Checks whether the list is overly repetitive.
Latency	Time required to produce a recommendation response.	a Important for a responsive shopping experience.

8.2 FUNCTIONAL TEST CASES

TABLE VII		
Test	Input / Condition	Expected Behaviour
TC-01	New session, no history	Return a valid fallback list without an error.
TC-02	Product has zero stock	Product is excluded from purchasable recommendations.
TC-03	User applies a price limit	Displayed products satisfy the price filter.
TC-04	Same request and same model version	Output is deterministic when the design expects it.
TC-05	Missing product description	System uses available metadata or a safe fallback.
TC-06	Recommendation service unavailable	UI shows a graceful fallback rather than breaking the page.

8.3 USER-CENTRED EVALUATION

A small usability study may ask participants to find a suitable product with and without recommendations, rate explanation clarity, and report whether the results are easy to control. Any study involving people should follow institutional approval and informed-consent requirements. The sample, tasks, questionnaire, analysis method, and limitations should be reported transparently. Click-through rate alone should not be treated as proof of better customer outcomes. 8.4 Discussion The hybrid approach is flexible and suitable for an incremental prototype. Content signals address cold-start products, while collaborative signals can improve discovery when interaction history is adequate. However, the benefit depends on data quality, evaluation design, and the catalogue. The system should be considered successful only after measured results show acceptable relevance, coverage, latency, and usability for the intended setting. E-commerce Product Recommendation System | 8

9. LIMITATIONS AND FUTURE SCOPE

The proposed design depends on accurate product metadata and sufficient, lawfully collected interaction data. New stores may have too little history for collaborative filtering, while large catalogues may require approximate retrieval or caching to meet latency targets. Customer intent can change rapidly, and a short session may not reflect a stable preference. Offline metrics also cannot fully represent trust, satisfaction, or long-term shopping outcomes. Future work can compare different content representations, matrix-factorization or nearest-neighbour approaches, and learning-to-rank methods. Context-aware ranking may incorporate device, season, or current page context only when justified and privacy-appropriate. Further work can examine calibrated explanations, seller exposure, robustness to manipulated interactions, and online controlled experiments. Any deployment should include monitoring, rollback, and a way for users to adjust personalization. 10. Conclusion This paper presents the design of a hybrid e-commerce product recommendation system that combines product content, behavioural patterns, contextual signals, and business constraints. Its modular architecture supports candidate generation, ranking, explanation, and feedback while accounting for cold-start conditions, availability,

privacy, and diversity. A prototype can be implemented with a web interface, API layer, product database, and Python-based recommendation logic. The proposed evaluation plan covers ranking quality, catalogue coverage, diversity, latency, functional correctness, and usability. Actual effectiveness must be established through reproducible testing and carefully reported evidence rather than assumed from the design alone.

REFERENCES

1. Ricci, F., Rokach, L., and Shapira, B. (eds.). *Recommender Systems Handbook*. 3rd ed., Springer, 2022.
2. Resnick, P., and Varian, H. R. "Recommender Systems." *Communications of the ACM*, 40(3), 1997, pp. 56–58.
3. Adomavicius, G., and Tuzhillin, A. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 2005, pp. 734–749.
4. Burke, R. "Hybrid Recommender Systems: Survey and Experiments." *User Modeling and User-Adapted Interaction*, 12, 2002, pp. 331–370.
5. Koren, Y., Bell, R., and Volinsky, C. "Matrix Factorization Techniques for Recommender Systems." *Computer*, 42(8), 2009, pp. 30–37.
6. Schafer, J. B., Konstan, J., and Riedl, J. "E-Commerce Recommendation Applications." *Data Mining and Knowledge Discovery*, 5, 2001, pp. 115–153.
7. Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. "Evaluating Collaborative Filtering Recommender Systems." *ACM Transactions on Information Systems*, 22(1), 2004, pp. 5–53.
8. Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. *Recommender Systems: An Introduction*. Cambridge University Press, 2010.
9. Gunawardana, A., and Shani, G. "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks." *Journal of Machine Learning Research*, 10, 2009, pp. 2935–2962.
10. Ekstrand, M. D., Kluver, D., Harper, F. M., and Konstan, J. A. "Letting Users Choose Recommender Algorithms: An Experimental Study." *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015.

Publication archival note: Official scholarly contribution published under CC BY 4.0 International License. Permanent record on <https://www.ijrek.org> | Article ID: IJREK-2026-00165