



A Multi-Domain Recommendation System Using Domain-Specific Recommendation Strategies for Heterogeneous Information Domains

Yugant Shahare¹, Joseph Telang², Ayush Bhagwat³, Payal Khade⁴, Yash Nagrale⁵,
Khushi Yesur⁶, Sushama V. Telrandhe⁷

^{1,2,3,4,5}UG Student, Department of Artificial Intelligence, Guru Nanak Institute of Engineering and Technology (GNIET), Nagpur, Maharashtra, India

⁶Assistant Professor, Department of Artificial Intelligence, GNIET, Nagpur, Maharashtra, India

⁷Associate Professor, Department of Artificial Intelligence, GNIET, Nagpur, Maharashtra, India

Abstract

How to Cite this Article:

Shahare, Y., Telang, J., Bhagwat, A., Khade, P., Nagrale, Y. & Yesur, K. (2026). A Multi-Domain Recommendation System Using Domain-Specific Recommendation Strategies for Heterogeneous Information Domains. International Journal of Creative and Open Research in Engineering and Management, 2(10), 1-9.
<https://doi.org/10.55041/ijcope.v2i10.124>

License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.



<https://doi.org/10.55041/ijcope.v2i10.124>

Recommender systems are usually designed around a single domain and a single interaction model, yet practical applications often require recommendations across heterogeneous domains whose data differ sharply in structure and signal strength. This paper presents a multi-domain recommendation system implemented as a unified Streamlit application covering movies, hospitals, books, places, careers, and learning roadmaps. Rather than forcing one algorithm onto every domain, the system selects a method according to the available signal: truncated singular value decomposition (SVD) for personalized movie recommendation; TF-IDF with cosine similarity for content-based movie and book recommendation; deterministic disease-to-specialty matching for hospitals; location and category filtering for places; stream-to-exam rules for career guidance; and a prerequisite- and progress-aware rule engine for learning roadmaps. Optional live retrieval supplies movie, book, place, and learning resources, while SQLite persists roadmap progress and administrator-managed content. On the MovieLens ml-latest-small dataset, under a chronological leave-one-rating-per-user protocol, the SVD model attains an RMSE of 1.012 and an MAE of 0.780, against 1.117 and 0.918 for a global-mean baseline—error reductions of approximately 9.35% and 14.99%. The paper distinguishes the real MovieLens benchmark from the reproducibly generated demonstration datasets used for hospitals, books, and places. The result is a transparent, modular prototype showing that heterogeneous recommendation problems can be served by domain-appropriate techniques rather than a one-model-fits-all architecture.

Keywords— recommender systems, multi-domain recommendation, collaborative filtering, singular value decomposition, TF-IDF, content-based filtering, learning roadmap, Streamlit.



INTRODUCTION

Recommender systems help users discover items, services, information, and actions in a large search space [1], [2]. Classical research often assumes a homogeneous item space such as movies, products, music, or news. In a practical educational or consumer application, however, a user may need several different recommendation tasks at once: discovering movies, locating a hospital specialty, finding related books, exploring destinations, selecting an entrance examination, or following a personalized learning roadmap. These tasks differ in their available signals, semantics, constraints, and acceptable explanations.

This work addresses the problem with a single application shell that hosts multiple domain-specific recommendation modules. The guiding principle is algorithmic fit: collaborative filtering is used where repeated user–item ratings form a useful interaction matrix; content-based similarity is used where item metadata is the strongest signal; and deterministic or rule-based ranking is used where the relationship is semantic or constraint-driven rather than learned from ratings. This is consistent with established practice, in which matrix factorization is a strong collaborative technique [3] and content representations are effective when item attributes are available [4].

The contributions of this paper are:

- a unified multi-domain architecture in which each domain uses its own recommendation strategy;
- a movie component combining personalized SVD ranking with TF-IDF similarity, evaluated on the MovieLens benchmark [5];
- lightweight modules for domains that lack a dense user–item feedback matrix;
- persistent, prerequisite-aware learning-roadmap recommendation;
- optional live retrieval from external services with offline fallbacks; and
- an explicit data-provenance policy separating benchmark data from synthetic demonstration data.

RELATED WORK

Collaborative filtering. Matrix factorization represents users and items in a shared latent space and has become a central collaborative approach; Koren, Bell, and Volinsky describe it as an effective family of techniques for recommender systems [3]. Here, mean-centered user–item ratings are factorized with truncated SVD and the latent factors are used to predict unseen movie ratings.

Content-based recommendation. Content-based methods recommend items from descriptive attributes rather than other users' interactions. TF-IDF is a classical term-weighting scheme that converts text into numerical vectors [4]. This work combines movie genres with community tags and applies cosine similarity; books are handled analogously using category and author.

Multi-domain recommendation. Multi-domain systems must account for differences between domains rather than assuming one feature representation is meaningful everywhere. The architecture here is deliberately modular: each domain owns its data preparation, scoring logic, and constraints while sharing a common user-facing shell.

Learning-resource recommendation. Learning recommendations need additional structure such as prerequisites, progress, resource quality, and completion state. Because no historical learner–interaction training set exists, a transparent rule-based personalization engine is used instead of a learned model.

SYSTEM OBJECTIVES

- Serve heterogeneous domains through one application.
- Select a suitable technique per domain instead of one algorithm for all data.
- Support personalized movie recommendation from ratings and similarity-based recommendation from metadata.
- Provide transparent rule-based recommendation where a deterministic relationship is more appropriate than collaborative filtering.
- Persist roadmap state, prerequisites, resources, and progress in SQLite.
- Support live discovery while retaining offline fallbacks.
- Maintain data provenance and clearly label generated demonstration values.

SYSTEM ARCHITECTURE

The application is layered. A Streamlit presentation layer collects user preferences and displays ranked results. Domain modules perform recommendation logic over local CSV/SQLite data or optional live retrieval services. Separating the interface from recommendation logic and persistence makes each module independently testable and replaceable (Fig. 1).

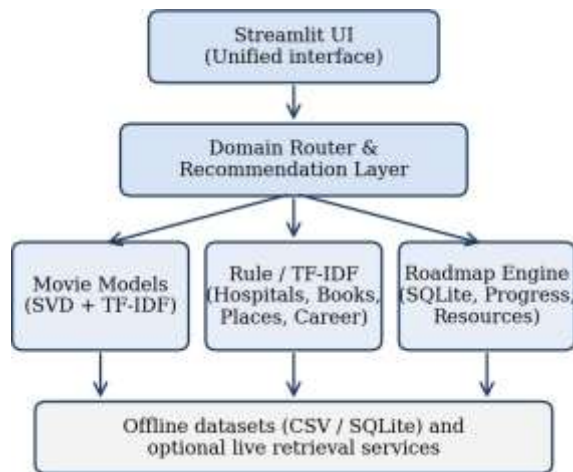


Fig. 1: High-level architecture of the multi-domain system.

Domain-Specific Strategies

Table I maps each domain to its technique and primary signal.

Table I: Domains, Methods, and Available Signals

Domain	Technique	Primary signal
Movies (personalized)	SVD collaborative filtering	User–movie ratings
Movies (similar)	TF-IDF + cosine similarity	Genres + tags
Hospitals	Disease-to-specialty match; TF-IDF fallback	Disease/specialty semantics
Books	TF-IDF + cosine similarity	Category + author
Places	Filter + rating sort	Country/city/category
Career	Stream-to-curated-exam mapping	Academic stream
Learning roadmaps	Rules + prerequisites + progress	Topics, resources, history

Movie Collaborative Filtering

The MovieLens ml-latest-small ratings matrix R is loaded and, for each user u , the mean of observed ratings μ_u is subtracted from the observed entries:

$$\tilde{R}_{ui} = r_{ui} - \mu_u \quad (1)$$

A truncated SVD with up to $k = 50$ latent factors is learned from the centered matrix, $\tilde{R} \approx U_k \Sigma_k V_k^T$. The predicted rating is the user mean plus the reconstructed latent preference, clipped to the original rating range:

$$\hat{r}_{ui} = \text{clip}_{[1,5]}(\mu_u + p_u^T q_i) \quad (2)$$

where p_u and q_i are the user and item latent vectors. Recommendations exclude movies the user has already rated, rank the rest by predicted rating, and return the top- N candidates.

Content-Based Filtering

Movie metadata is converted to a text document of genres and community tags, with genres repeated to increase their influence. TF-IDF vectors are computed with English stop-word removal and a vocabulary capped at 5,000 features. Items similar to a selected movie are found by cosine similarity,

$$\text{sim}(a, b) = a \cdot b / (|a| |b|) \quad (3)$$

A Bayesian-style weighted rating is additionally used for display ranking so that an item with very few ratings does not dominate one supported by many. The book module applies the same pattern to category and author attributes, reusing the algorithm without assuming that movie-style interaction data exist for books.

Rule-Based and Constraint-Aware Modules

Hospitals. A disease or health-related query is mapped to relevant specialties, optionally filtered by city and country. This module is an information-retrieval demonstration only; it does not diagnose disease, select treatment, or replace professional medical advice.

Places and careers. The place module filters candidates by country, city, and optional category and sorts by rating. The career module maps an academic stream to a manually curated set of entrance examinations ordered by a manually assigned difficulty rank. These modules are deliberately transparent because there are insufficient user-choice data to justify collaborative learning.

Learning roadmaps. The engine uses prerequisite relationships, resource verification, progress status, homework completion, and available time to recommend the next topic or resource or to construct a plan. It can combine persistent verified resources with live YouTube search results. It is a rule-based personalization engine, not a supervised learning model.

DATASETS AND PROVENANCE

The primary benchmark is MovieLens ml-latest-small [5], containing 610 users, 9,742 movies, 100,836 ratings, and 1,572 movie-tag records. The remaining domain files are demonstration datasets generated or curated for this work. The hospital, book, and place tables combine real entity names with reproducible, seed-based simulated ratings and review counts rather than scraped real-world ratings. The exam table is



manually curated, and its difficulty ordering is a demonstration ranking rather than a learned measure. Keeping these distinctions explicit supports reproducibility and avoids presenting synthetic values as public sentiment (Table II).

Table II: Dataset Inventory and Provenance

Dataset	Size	Provenance
MovieLens movies	9,742 movies	Real benchmark
MovieLens ratings	100,836 ratings; 610 users	Real benchmark
Movie tags	1,572 rows	Derived from MovieLens tags
Hospitals	197 rows	Generated demonstration data
Books	120 rows	Generated demonstration data
Places	158 rows	Generated demonstration data
Entrance exams	43 rows	Curated project data
Roadmaps	SQLite curriculum/resources	Seeded + admin/live resources

EXPERIMENTAL METHODOLOGY

To obtain a reproducible quantitative check of the principal learned component, ratings were ordered chronologically within each user. Each user's final rating was held out as a test observation and all earlier ratings were used for training, yielding 610 test observations and avoiding use of a later interaction to predict an earlier one. This follows standard practice for offline evaluation of collaborative filtering [6].

The SVD model uses 50 latent factors where matrix dimensions permit, a fixed random seed of 42, user-mean centering, and prediction clipping to [1, 5]. Accuracy is reported as root mean squared error and mean absolute error over the test set T :

$$\text{RMSE} = \sqrt{(1/|T|) \sum (r_{ui} - \hat{r}_{ui})^2} \quad (4)$$

$$\text{MAE} = (1/|T|) \sum |r_{ui} - \hat{r}_{ui}| \quad (5)$$

A global-mean predictor trained only on the training ratings serves as the baseline.

IMPLEMENTATION

- **Framework:** Python with Streamlit for the interactive interface.
- **Data and ML:** pandas and NumPy for processing; scikit-learn [7] for TF-IDF, cosine similarity, and truncated SVD.

- **Persistence:** SQLite for roadmaps, topics, prerequisites, resources, concepts, learning history, sessions, and progress.
- **Live retrieval:** optional TMDb, Google Books, Google Places, and YouTube modules, with offline fallbacks where implemented.
- **Administration:** an admin passphrase gate and CRUD operations for curriculum and resource management.
- **Modularity:** one Python module per domain, so domain logic can change without redesigning the interface.

RESULTS AND DISCUSSION

Table III and Fig. 2 report the chronological leave-one-out results. The SVD model reduces RMSE by approximately 9.35% and MAE by approximately 14.99% relative to the global-mean baseline, indicating that it learns useful structure from the rating matrix. These results support latent collaborative modeling for the movie domain of this prototype; they should not be generalized to the other domains, which use different data and objectives.

Table III: Chronological Leave-One-Out Results on MovieLens ml-latest-small (610 Test Users)

Model	RMS E	MAE	Reduction vs. baseline
Global mean	1.117	0.918	—
SVD (50 factors)	1.012	0.780	9.35% RMSE; 14.99% MAE

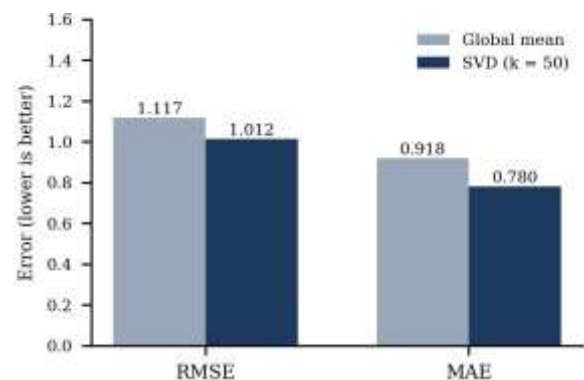


Fig. 2: Error of the global-mean baseline and the SVD model (lower is better).

No accuracy metric is reported for the generated hospital, book, and place ratings, as they are not externally validated; the career ranking and roadmap rules are likewise treated as system logic rather than learned performance claims.

The remaining domains illustrate an engineering point. A hospital request is naturally constrained by specialty and location; a book request can be answered from



category and author metadata; a place request can be narrowed by location and category; and a career request can be mapped from academic stream to examinations. Applying SVD to these domains would not make the system more rigorous, because the necessary interaction data are absent.

The roadmap module extends the idea from item recommendation to sequential learning assistance. Prerequisites prevent premature topic recommendations, progress status prevents repeating completed resources, and resource verification adds a curation layer over live retrieval. The result is an interpretable cycle: select an eligible topic, assess completion state, select a suitable resource, connect it to concepts and practice problems, and update progress.

A major strength is transparency: every recommendation can be explained by predicted rating, content similarity, specialty match, location filter, stream mapping, prerequisite state, or resource score. A major limitation is that the system has not yet been evaluated in a cross-domain user study.

SECURITY AND RESPONSIBLE USE

The roadmap administrator gate is lightweight, demonstration-level protection rather than a full identity-management system. A production deployment should replace the default passphrase, store secrets outside source control, use a proper authentication provider, enforce authorization on all write operations, protect session state, and use HTTPS.

The hospital component is an information-retrieval feature only and must not be described as a clinical recommendation engine. Synthetic ratings must remain labeled as synthetic. Career information such as examination names, eligibility, conducting bodies, and admission pathways can change, so production use should verify it against official sources.

LIMITATIONS

- Only the MovieLens component has a real public rating benchmark suitable for the reported experiment.
- Hospital, book, and place ratings and review counts are simulated and are not externally validated.
- The SVD component is a prototype truncated-SVD approach, not a production bias-aware or implicit-feedback matrix-factorization framework.
- Personalization transfer across domains has not been evaluated.

- Live APIs introduce availability, rate-limit, licensing, and data-freshness dependencies.
- The admin authentication suits a college prototype but needs hardening for public deployment.
- Cold-start users and items remain difficult for collaborative filtering; content-based and rule-based modules mitigate but do not eliminate the problem.

FUTURE WORK

- Collect consented real interaction data across domains and study cross-domain latent representations.
- Add stronger baselines: popularity ranking, item-based collaborative filtering [8], Bayesian personalized ranking [9], and regularized matrix factorization.
- Evaluate Precision@K, Recall@K, NDCG@K, coverage, diversity, novelty, calibration, and latency in addition to rating error.
- Develop a learned hybrid recommender that combines collaborative and content signals while preserving domain constraints.
- Add explanations and user-controllable diversity and novelty settings.
- Replace demonstration authentication with a production identity provider and role-based access control.
- Replace simulated ratings with licensed or openly licensed real datasets, documenting provenance, collection dates, and preprocessing.
- Conduct a user study measuring usefulness, trust, satisfaction, and perceived quality.

CONCLUSION

This paper presented a modular multi-domain recommendation system that treats recommendation as a set of domain-specific problems rather than a single modeling task. It combines SVD collaborative filtering and TF-IDF similarity for movies, content-based recommendation for books, transparent semantic and constraint-based ranking for hospitals, places, and careers, and a prerequisite- and progress-aware engine for learning roadmaps. The MovieLens experiment showed a measurable improvement over a global-mean baseline, and the architecture offers a practical path to domains where collaborative training data are unavailable.

Importantly, the system separates benchmark evidence from demonstration data: the reported evaluation is tied to the real MovieLens benchmark, while synthetic and curated datasets are identified as prototype



resources. Future work should validate the remaining domains with real, licensed datasets and user studies.

REFERENCES

- [1] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, 1997.
- [2] F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*, 3rd ed. New York, NY, USA: Springer, 2022.
- [3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
- [4] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.
- [5] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, Art. no. 19, 2015, doi: 10.1145/2827872.
- [6] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.
- [7] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [8] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web (WWW)*, 2001, pp. 285–295.
- [9] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell. (UAI)*, 2009, pp. 452–461.