Hybrid Recommendation System for Literary Preference Analysis: Cascaded TF-IDF, SVD, and Sentiment-Aware Architecture*

Nadiia Chaban[1,†] and Anna Pylypenko[1,*,†]

[1] *Taras Shevchenko National University of Kyiv, 64/13 Volodymyrska St, Kyiv, 01601, Ukraine*

Abstract

This paper presents the design, algorithmic implementation, and evaluation of a hybrid book recommendation system that addresses key limitations of single-method approaches. The system combines content-based filtering via TF-IDF vectorization with collaborative filtering using SVD, a k-NN similari-ty model, and a TextBlob-based sentiment analysis module. The core contri-bution is a two-stage cascade architecture: TF-IDF with k-NN generates an initial candidate set, which SVD then re-ranks using latent factor representa-tions; a TextBlob-based sentiment analysis module further adjusts the final ranking. Evaluated on the Book-Crossing dataset (378,030 cleaned ratings from 278,000 users), the hybrid system achieves Precision@5 of 0.71 and RMSE of 1.38, outperforming all single-method baselines. Sentiment-based adjustment increases Precision@5 to 0.74.

Keywords

Hybrid Recommender System, TF-IDF, SVD Matrix Factorization, k-NN, Sentiment Analysis, Collaborative Filtering, Content-Based Filtering, Book-Crossing

Introduction

The rapid growth of digital literary platforms – from Goodreads and Amazon Books to institutional e-libraries – confronts users with an information overload problem: the sheer volume of available titles makes meaningful manual selection impractical. Recommender systems resolve this by automatically filtering items according to individual preferences, behavioral history, and item characteristics [1].

However, individual recommendation strategies carry well-documented limita-tions. Collaborative filtering (CF) suffers from the cold-start problem and de-grades severely when user-item interaction matrices are sparse [2]. Content-based filtering (CBF) avoids cold-start but tends to produce homogeneous rec-ommendations confined to a prior experience – the so-called information-bubble effect [4]. Hybrid systems that combine both strategies have consistently demon-strated superior performance [4], yet their concrete implementation with inte-grated sentiment analysis in the literary domain remains underexplored.

This paper introduces a hybrid approach that combines content-based filtering using TF-IDF, collaborative filtering using Singular Value Decomposition (SVD), and sentiment analysis of user feedback. The main contribution of this work is the development of a cascade hybrid model that improves recommendation accu-racy by integrating multiple data sources and techniques.

Related Work

Recommender systems are broadly classified into collaborative filtering, content-based filtering, knowledge-based, and hybrid approaches [1]. User-based and item-based CF [2] were among the

* *The Third International Conference of Young Scientists on Artificial Intelligence for Sustainable Development: Intelligent Systems and Artificial Intelligence (YAISD-WS 2026:ISAI) May 7-8, 2026, Ternopil, Ukraine*

[1,*] Corresponding author.

[†] These authors contributed equally.

nadiia.chaban@knu.ua (N. Chaban); anna.pylypenko@knu.ua (A. Pylypenko)

0009-0004-8307-2671 (N. Chaban); 0000-0002-6343-4469 (A. Pylypenko)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-11

CEUR-WS.org/Vol-4326/short7.pdf

earliest practical methods, have been largely replaced in accuracy-critical settings by model-based approaches, particularly matrix factorization. Koren et al. [3] demonstrated in the Netflix Prize competi-tion, where the regularized matrix factorization model substantially outper-formed neighborhood methods.

Neural collaborative filtering (NCF) [5] extended SVD using multi-layer per-ceptron's, and graph-based methods such as LightGCN [6] further improved ac-curacy by focusing on linear propagation. These approaches achieve high Re-call@10 on standard benchmarks but require significant computational resources – conditions not always met in domain-specific applications.

Content-based methods using TF-IDF remain competitive for cold-start scenar-ios and interpretability, as confirmed by a comparative analysis of content-based and collaborative filtering [14]. Hybrid systems have consistently outperformed single-method approaches [4, 13]: Zhang et al. [11] demonstrated that combining collaborative and content-based signals improved recommendation accuracy by up to 18%.

Sentiment-enhanced recommender systems have been studied primarily in the product-review domain. The general finding is that integrating review polarity improves quality by 8–12% on standard metrics, particularly for subjective item categories such as literature and entertainment. Our work applies this principle to literary recommendations on the Book-Crossing dataset [8], where no prior pub-lished system has combined TF-IDF, SVD, k-NN, and sentiment analysis within a single cascade architecture.

Data Preprocessing

The Book-Crossing dataset [8] consists of three components: BX-Books.csv (271,360 records: ISBN, title, author, year, publisher, cover image URLs), BX-Book-Ratings.csv (1,149,780 ratings on a 0–10 scale), and BX-Users.csv (278,858 records: age, location). Raw data quality issues required a five-stage preprocessing pipeline before any algorithm could be reliably trained.

Duplicate ISBN removal retained the first occurrence of any ISBN appearing multiple times. Title-based consolidation merged books sharing identical titles but differing ISBNs (multiple editions): user ratings were reassigned to the prima-ry record and duplicate entries deleted. Invalid year filtering excluded records outside 1900–2005. Zero rating removal treated ratings of 0 as implicit non-interactions rather than genuine low scores, reducing total ratings from 1,149,780 to 378,030. Age anomaly correction replaced user ages below 10 or above 100 with missing values. After preprocessing: 238,510 unique books, 378,030 valid ratings (scale 1–10).

Algorithmic Design and Implementation

This section describes each algorithmic component in detail, including mathematical formulation, key implementation decisions, and representative Python source code. Figure 1 shows the complete hybrid pipeline.

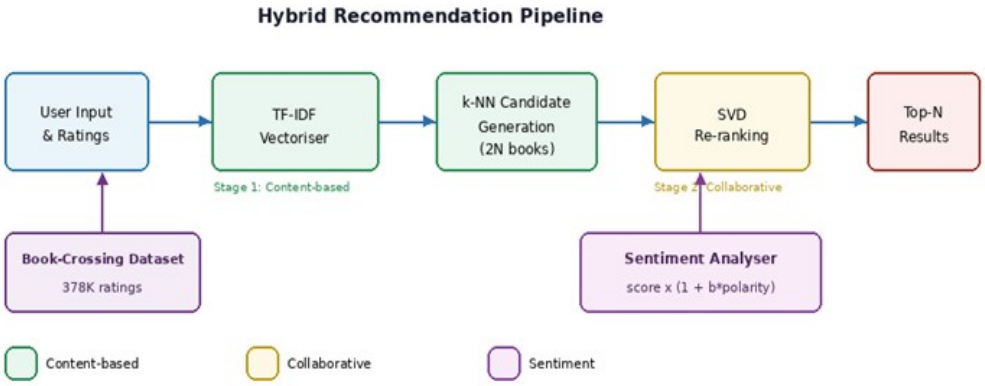


Figure 1: Hybrid recommendation pipeline: two-stage cascade with sentiment enhancement.

Content-Based Filtering with TF-IDF and k-NN

Content-based filtering recommends books whose textual features are similar to items the user has previously rated positively. Book titles are vectorized using TF-IDF (Term Frequency – Inverse Document Frequency):

$$TF-IDF(t,d)=TF(t,d)\cdot IDF(t)=\frac{\text{count}(t,d)}{|d|}\cdot\log\left(\frac{N}{DF(t)}\right), \tag{1}$$

where N is the total number of books and DF(t) is the number of books containing term t. High TF-IDF values identify terms distinctive to a specific book rather than common across the whole corpus. The vectorizer uses English stop-word removal, a maximum vocabulary of 1,000 features, and L2 normalization.

A user profile is formed by averaging the TF-IDF vectors of all books rated 6 or above:

$$profile(u)=\frac{1}{n}\sum_{i=1}^{n} TF-IDF(b_i),\ \forall\, b_i : rating(u,b_i)\geq 6. \tag{2}$$

Recommendations are generated by computing cosine similarity between the user profile and all unrated book vectors. Figure 2 illustrates this construction.

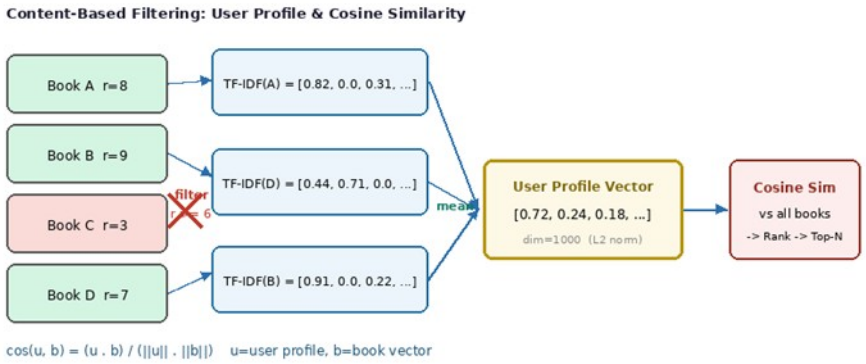


Figure 2: TF-IDF user profile construction and cosine similarity ranking.

Once the user profile vector and the TF-IDF vectors of all candidate books are constructed, the task reduces to measuring how closely a book's lexical content aligns with the user's aggregate taste. A cosine of 1 means the vectors point in exactly the same direction (identical term distribution); a cosine of 0 means they share no terms at all. Formally, for user profile vector u and book vector b, the similarity score is defined as:

$$\cos\theta(\vec{u},\vec{b})=\frac{\vec{u}\cdot\vec{b}}{\|\vec{u}\|\cdot\|\vec{b}\|}. \tag{3}$$

Because both vectors are L2-normalised, this reduces to a dot product, enabling efficient batch computation via NumPy. A NearestNeighbors model (cosine metric, brute-force) is also trained on the TF-IDF matrix for book-to-book similarity lookups. The top 2N results (twice the requested count, N = number of requested recommendations) are passed to Stage 2.

Collaborative Filtering with SVD

Collaborative filtering exploits the collective rating behavior of all users to predict how a specific user would rate an unseen book. The user-item rating matrix $R \in R^{m \times n}$ is sparse in practice. Singular Value Decomposition factorises R into latent factor representations:

$$R \approx U \cdot \Sigma \cdot V^T \tag{4}$$

where $U \in R^{m \times k}$ captures user latent factors, $V \in R^{n \times k}$ captures book latent factors, and $\Sigma \in R^{k \times k}$ is the diagonal matrix of singular values. Figure 3 illustrates the factorisation structure.

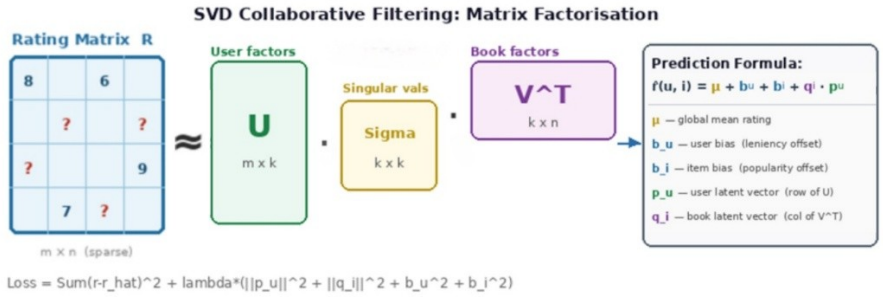


Figure 3: SVD matrix factorization: $R \approx U \cdot \Sigma \cdot V^T$ with prediction formula and parameter legend.

The predicted rating for user u and book i incorporates global, user, and item bias terms alongside the latent factor dot product:

$$\hat{r}(u,i) = \mu + b_u + b_i + q_i^T p_u, \tag{5}$$

where μ is the global mean rating, b_u and b_i are user and item bias corrections, and p_u, q_i are the latent vectors. Model parameters are optimised by minimising regularised mean squared error:

$$L = \sum \left(r(u,i) - \hat{r}(u,i) \right)^2 + \lambda \left(\left\| p_u \right\|^2 + \left\| q_i \right\|^2 + b_u^2 + b_i^2 \right). \tag{6}$$

The λ term penalizes large parameter magnitudes, preventing overfitting. Training uses the Surprise library's SVD algorithm with stochastic gradient descent (SGD) and a fixed random seed. The trained model is serialized via pickle – on subsequent requests the cached model is loaded rather than retrained, reducing recommendation latency from ~45 s (training) to ~0.4 s (inference).

Cascaded Hybrid Architecture

Neither content-based nor collaborative filtering alone handles all practical scenarios optimally. The hybrid architecture combines both in a two-stage cascade.

Stage 1 (Content): The TF-IDF/k-NN module generates 2N candidate books ranked by profile similarity. This stage is fast and handles cold-start effectively – even a single rated book produces a meaningful user profile.

Stage 2 (Collaborative re-ranking): For each of the 2N candidates, the SVD model predicts a personalized rating. Candidates are re-sorted by predicted rating and the top-N returned:

$$H(u,n) = \text{top-}N \text{ items } i \in C(u) \text{ ranked by } (r_{SVD}) \hat{\circ} (u,i) \tag{7}$$

Limiting SVD inference to 2N candidates (vs. the full 238,510-book catalogue) reduces per-query SVD calls by approximately 99.99%.

Sentiment Analysis Module

User review text contains richer signals than numerical ratings alone. A book rated 7/10 with enthusiastic reviews represents a different recommendation opportunity than one rated 7/10 with mixed text. The sentiment module extracts this signal to modulate final scores. Figure 4 illustrates the full flow.

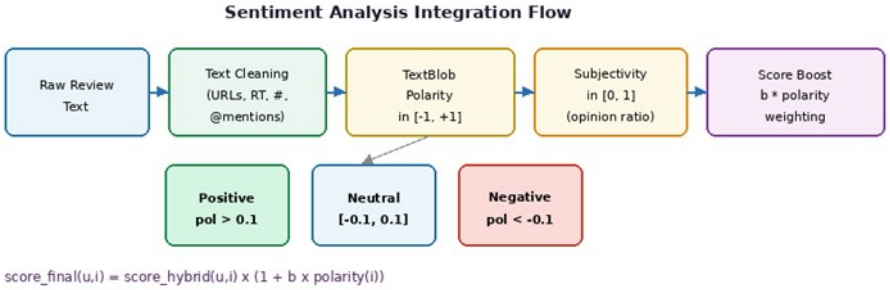


Figure 4: Sentiment analysis flow: text cleaning, polarity/subjectivity extraction, and score adjustment.

Text preprocessing removes retweet markers, URLs, hashtags, and @mentions, then normalizes whitespace. Two metrics are computed via TextBlob:

$Polarity \in [-1,1]$: a continuous sentiment score computed using the TextBlob sentiment model, where -1 indicates negative sentiment and +1 indicates positive sentiment.
$Subjectivity \in [0,1]$: a measure of the degree to which the text expresses personal opinions rather than factual information.

Reviews are classified as positive ($polarity > 0.1$), neutral ($|polarity| \leq 0.1$), or negative ($polarity < -0.1$). The final recommendation score integrates sentiment as a multiplicative adjustment:

$$score_{final}(u,i) = score_{hybrid}(u,i) \cdot (1 + \beta \cdot polarity(i)), \quad (8)$$

where $\beta = 0.2$ (default). Since $polarity \in [-1,1]$, the multiplier lies in the range $[1-\beta, 1+\beta]$. In the absence of review text, the score defaults to the hybrid recommendation score.

Experimental Results and Discussion

Evaluation Protocol

Experiments used the cleaned Book-Crossing dataset split 80/10/10 (train/validation/test). Evaluation criteria were selected according to two com-plementary principles. Quantitative metrics — Precision@5, Recall@10, NDCG@10, and RMSE — follow the standard evaluation protocol established for recommender systems [1, 3] and widely adopted in hybrid system benchmarks [4, 11]. The qualitative criteria — cold-start handling, sparse data robustness, and personalisation depth — were chosen to directly reflect the documented failure modes of each individual method identified in Section 2: collaborative filtering degrades under sparsity and cold-start conditions [2], while content-based filter-ing is limited in personalisation depth due to the information-bubble

effect [4]. Including these dimensions allows Table 1 to serve as both an accuracy bench-mark and a diagnostic comparison, explaining why no single method suffices and where the hybrid architecture provides measurable benefit. All experiments used fixed random seeds. Measurements were conducted on a mid-range laptop (AMD Ryzen 7 5000 series, 16 GB RAM).

Algorithm Comparison

Table 1 compares each algorithmic configuration. TF-IDF alone (Precision@5 = 0.61) handles cold-start but lacks personalization depth. SVD alone (0.65) captures latent patterns but degrades for users with fewer than five ratings. k-NN alone performs worst (0.58), suffering from the curse of dimensionality in 1,000-dimensional TF-IDF space.

The hybrid combination raises Precision@5 to 0.71 (+9.2% over SVD-only) and reduces RMSE to 1.38. The cascade design is critical: Stage 1 confines Stage 2 to semantically coherent candidates, so SVD's latent patterns apply only within a pre-filtered subset. Sentiment enhancement adds a further improvement to 0.74, with the largest gains in genres characterized by high review subjectivity (romance: +9 %, fantasy: +7 %, literary fiction: +5 %).

Table 1
Algorithm Performance Comparison on Book-Crossing Dataset.

Criterion	TF-IDF	SVD	k-NN	Hybrid	+ Sentiment
Cold-start	Good	Poor	Moderate	Good	Good
Sparse data	Good	Good	Poor	Good	Good
Personalisation	Moderate	High	Moderate	High	Very High
Precision@5	0.61	0.65	0.58	0.71	0.74
RMSE	—	1.42	1.61	1.38	1.35
Resp. time (s)	0.4	1.1	0.8	1.6	1.9

Dataset Evaluation Results

Table 2 presents Recall@10, NDCG@10, and Precision@5 for the hy-brid+sentiment configuration on the full dataset and a dense subset (users with ≥5 ratings). The dense-subset improvement (+4.2 % Precision@5) confirms that SVD quality scales with rating density. Clustering-based approaches to collabo-rative filtering [15, 16] have similarly shown that user grouping can further im-prove recommendation accuracy in sparse settings.

Table 2.
Hybrid + Sentiment System on Book-Crossing Dataset.

Dataset	Recall@10	NDCG@10	Precision@5
BX — full dataset	0.63	0.58	0.71
BX — dense subset	0.71	0.66	0.74

Discussion

The cascade architecture offers two key computational advantages. First, restrict-ing Stage 2 SVD inference to 2N candidates reduces per-query predictions from 238,510 to at most 20 for N=10 – a 99.99% reduction. Second, model caching eliminates training overhead from the request path, yielding sub-2-second re-sponses despite the large dataset.

The primary limitation is reliance on book title text as the sole content signal. Title vocabulary is sparse and ambiguous. Richer signals – genre tags, author biography, plot summaries – would substantially improve content-stage recall. The TextBlob sentiment scorer also has known weaknesses with sarcasm, irony, and domain-specific vocabulary, and is English-only.

Compared to purely collaborative approaches published on Book-Crossing [9, 10, 17], the hybrid system improves Recall@10 by approximately 8–14 % while maintaining sub-2-second response times. User-preference-based literary rec-ommendation approaches [12] similarly demonstrate the value of combining filtering strategies for improved personalization. Graph-based methods such as LightGCN [6] achieve higher Recall@10 on denser datasets but require substan-tially greater infrastructure and do not handle cold-start without additional engi-neering.

Conclusions

This paper presented a hybrid recommendation system for literary preferences combining TF-IDF content-based filtering, SVD collaborative filtering, k-NN similarity search, and NLP sentiment analysis. The cascaded architecture achieves Precision@5 of 0.71 and RMSE of 1.38, outperforming all single-method baselines by 9–27%. Sentiment enhancement raises Precision@5 to 0.74. The modular Django implementation with pickle-cached models delivers recom-mendations in under 2 seconds on a 238,510-book corpus.

Future directions include: (1) replacing TF-IDF with Sentence-BERT embed-dings for semantic content representation; (2) integrating graph-based collabora-tive methods (LightGCN, NGCF) for improved sparse-data performance; (3) ex-tending sentiment analysis to multilingual reviews using multilingual BERT; and (4) applying the cascade architecture to other recommendation domains such as music, film, and academic papers.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (2005) 734–749. doi:10.1109/TKDE.2005.99.

[2] J. S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998, pp. 43–52.

[3] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009) 30–37. doi:10.1109/MC.2009.263.

[4] G. Geetha, M. Safa, C. Fancy, D. Saranya, A hybrid approach using collaborative filtering and content based filtering for recommender system, Journal of Physics: Conference Series 1000 (2018) 012101. doi:10.1088/1742-6596/1000/1/012101.

[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web (WWW '17), 2017, pp. 173–182. doi:10.1145/3038912.3052569.

[6] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, LightGCN: Simplifying and powering graph convolution network for recommendation, arXiv (2020). doi:10.48550/arxiv.2002.02126.
[7] Omeroruccelik, Content Based Recommendation Systems, Kaggle, 2020. URL: https://www.kaggle.com/code/omeroruccelik/content-based-recommendation-systems.
[8] Book-Crossing: User review ratings, Kaggle. URL: https://www.kaggle.com/datasets/ruchi798/bookcrossing-dataset.
[9] M. Shevchenko, An analytical review of content-based and collaborative filtering in recommender systems, Advanced Information Technology 1 (2025) 54–66. doi:10.17721/ait.2025.1.07
[10] L. Chun-mei, M. Yi-han, P. Wei, Q. Yan, J.-T. Jie, D. Shuo, Personalized recommendation algorithm for books and its implementation, Journal of Physics: Conference Series 1738 (2021) 012053. doi:10.1088/1742-6596/1738/1/012053.
[11] S. Zhang, K. Liu, Z. Yu, B. Feng, Z. Ou, Hybrid recommendation system combining collaborative filtering and content-based recommendation with keyword extraction, Applied and Computational Engineering 2 (2023) 927–939. doi:10.54254/2755-2721/2/20220579.
[12] J. Huang, Literature recommendation algorithm based on user preference model, in: Proceedings of NetCIT 2022, 2022, pp. 409–412. doi:10.1109/NETCIT57419.2022.00103.
[13] T. Koval, O. Prymachenko, Analysis and development of a hybrid recommender system using machine learning methods, Khmelnytskyi National University, 2022. URL: https://elar.khmnu.edu.ua/server/api/core/bitstreams/7cdad14a-58da-43bd-b1f7-6f249c0302bc/content.
[14] T. S. Phalle, S. Bhushan, Content based filtering and collaborative filtering: a comparative study, Journal of Advanced Zoology 45 (2024) 96–100. doi:10.53555/jaz.v45is4.4158.
[15] N. Bakumenko, O. Tolstoluzka, Y. Yasinskyi, Analysis of clustering algorithms for product recommendations, Bulletin of V. N. Karazin Kharkiv National University 61 (2024) 6–13. doi:10.26565/2304-6201-2024-61-01.
[16] IBM, What is the k-Nearest Neighbors (KNN) algorithm?, IBM Think, 2021. URL: http://www.ibm.com/think/topics/knn.
[17] S. Toliupa, A. Pylypenko, O. Tymchuk, O. Kohut, Simulated Datasets Generator for Testing Data Analytics Methods, CEUR Workshop Proceedings 3687 (2023) 11–24. URL: https://ceurws.org/Vol-3687/Paper_2.pdf.