

Hybrid Filtering Implementation with Personal Threshold Mechanism to Improve Movie Recommendation Accuracy

¹Debora Patricia Sharon Rembet, ^{2*}Antonius Rachmat Chrismanto, ³Lucia Dwi Krisnawati

¹Informatics Department, Universitas Kristen Duta Wacana, Yogyakarta, INDONESIA

²Informatics Department, Universitas Kristen Duta Wacana, Yogyakarta, INDONESIA

³Informatics Department, Universitas Kristen Duta Wacana, Yogyakarta, INDONESIA

e-mail : ¹debora.patricia@ti.ukdw.ac.id, ^{2*}anton@ti.ukdw.ac.id, ³krisna@staff.ukdw.ac.id

Publisher's Note: JPPM stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Corresponding Author: Antonius Rachmat Chrismanto

Abstract

The growing volume of content on film streaming platforms makes it harder for users to find films that suit their tastes, highlighting the need for a personalized recommendation system. This research introduces a Hybrid Filtering architecture that unifies Collaborative Filtering using Singular Value Decomposition with Content-Based Filtering employing TF-IDF weighted vectors and cosine similarity measurement. The defining innovation is a dual-threshold mechanism grounded in each user's personal average rating (user_avg_rating), which individually filters candidate films before their hybrid scores are combined. New users are handled through a genre-based onboarding flow supported by a weighted popularity score. Experiments on the MovieLens ml-latest-small dataset (100,836 ratings, 610 users, 9,742 films) show that Hybrid Filtering achieves the best Precision@10 of 0.27 and NDCG@10 of 0.32. Switching from a fixed global threshold to the personal threshold raised Genre Precision from 0.40 to 0.90 (+125%). The SVD component attained RMSE of 0.88 and MAE of 0.67, and a user study with 10 participants yielded a mean relevance score of 8.2 out of 10.

Keywords—Hybrid Filtering, Collaborative Filtering, Content-Based Filtering, Personal Threshold

1 Introduction

The exponential growth of digital streaming services including platforms such as Netflix, Disney+, and HBO Max has produced content libraries of unprecedented scale. While this abundance benefits viewers in principle, it simultaneously creates what researchers term the information overload or paradox of choice [1] [2] [3] [13]. Two approaches that are commonly used to address this challenge are Collaborative Filtering (CF), which draws on shared behavioral patterns among users to predict individual preferences [2] [4], and Content-Based Filtering (CBF), which matches film attributes such as genre and synopsis to a user's viewing history [5]. Both methods come with well-known weaknesses: CF cannot generate recommendations for users who have not yet interacted with the system (the cold-start problem) [6], while CBF tends to over-specialize by only recommending content very similar to what the user has already watched [3] [7]. Addressing both limitations requires a combined approach that uses the strengths of each method [12].

Hybrid Filtering has become a common solution to these limitations by combining CF and CBF so that each method compensates for the weaknesses of the other [8] [9] [16]. This research implements a Hybrid Filtering system that uses SVD as the CF component and TF-IDF with cosine similarity as the CBF component, enriched by film content data retrieved from The Movie Database API (TMDb API). The main contribution of this research is a dual-threshold mechanism based on each user's personal average rating (user_avg_rating), applied before hybrid score combination to ensure that every candidate film meets that user's personal quality standard unlike previous studies that used a single fixed threshold for all users [4] [6] [10].

A genre-based onboarding flow and an explicit feedback mechanism are also built in to handle new users, allowing the system to produce personalized hybrid recommendations within the very first session. Performance is measured through four evaluation groups: Top-N relevance (Precision@10, Recall@10, F1@10, NDCG@10) under three ground truth scenarios, rating prediction accuracy (RMSE and MAE), Genre Precision as a supplementary metric, and a direct user study. The goals of this research are: (1) to implement and evaluate Hybrid Filtering with a personal threshold mechanism, (2) to compare its performance against standalone CF and CBF, and (3) to measure the effectiveness of the personal threshold compared to a global threshold.

©2026 Rembet et al.

2 Research Methods

This research followed seven stages carried out in sequence, building from data collection up to evaluation. Figure 1 shows the overall system flowchart. The system is delivered as a web application built on the Flask micro-framework, with SQLite handling user authentication, and a Python back-end leveraging the Surprise library (SVD), Scikit-learn (TF-IDF and cosine similarity), Pandas (data wrangling), and NumPy (numerical computations).

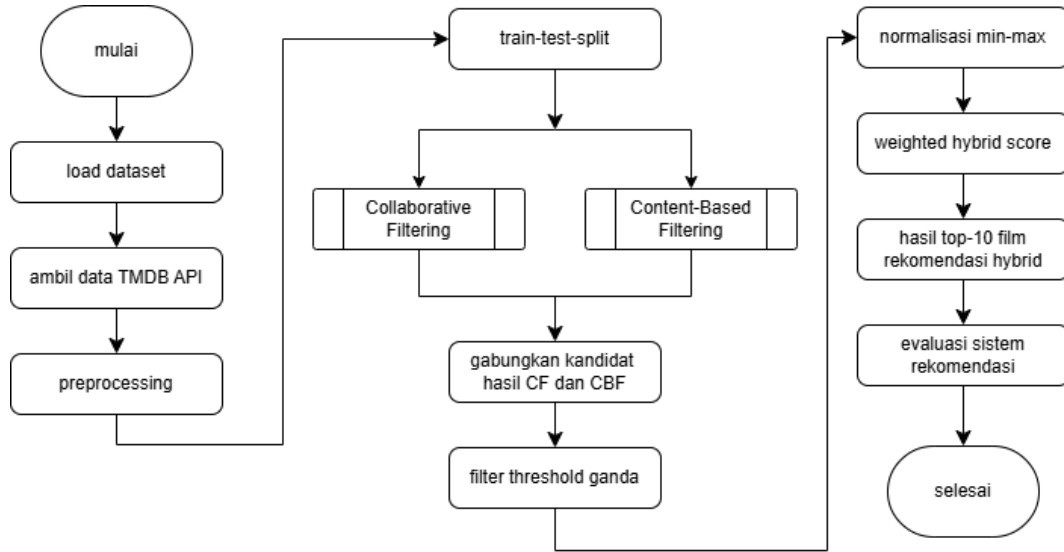


Figure 1. System Flowchart

2.1 Dataset

This research employs the MovieLens ml-latest-small dataset by GroupLens. This dataset has become a widely used reference in recommendation system research, owing to its combination of rich metadata, computationally manageable size, and extensive track record of adoption across numerous prior studies [2]. Three CSV files make up the dataset, ratings.csv holds 100,836 rating records, each containing a userId, a movieId, a score on a 0.5–5.0 scale with 0.5 increments, and timestamp. movies.csv lists 9,742 films, storing each film's title (with the release year inside parentheses) and its genres written as pipe-separated tokens. tags.csv logs 3,683 free-text labels that users attached to particular films, organized by userId, movieId, the tag itself, and a timestamp. Table 1 provides a summary of the dataset's core characteristics.

Table 1. Profile of MovieLens ml-latest-small Dataset

Component	Description	Value
Dataset Version	MovieLens ml-latest-small	–
Total Ratings	User rating entries against films	100,836
Total Users	Unique users who provided ratings	610
Total Films	Unique films that received ratings	9,742
Rating Scale	Range of user rating values	0.5 – 5.0
Matrix Sparsity	Proportion of unobserved interactions	98.3%
Most Frequent Rating	Rating value with highest occurrence	4.0 (26.60%)
File ratings.csv	userId, movieId, rating, timestamp	–
File movies.csv	movieId, title, genres	–
File tags.csv	userId, movieId, tag, timestamp	–

Rating values in the dataset follow a positively skewed distribution: the score of 4.0 appears most frequently at 26.60%, with 3.0 at 19.88% and 5.0 at 13.10%. Scores at or above 3.0 collectively represent 81.09% of all entries, leaving only 18.92% below that threshold. This asymmetry suggests a selective rating behavior in which users predominantly record scores for films they enjoyed, intensifying the sparsity challenge. At 98.3% vacancy, the user-item interaction matrix demands a dimensionality-reduction technique such as SVD that can extract meaningful signals from heavily incomplete data.

In addition to the base MovieLens dataset, supplementary content data was retrieved from The Movie Database API (TMDb API). Data retrieved includes synopsis (overview), cast list (up to 8 main actors), poster path, and release year. All data obtained from TMDb API is stored locally in a cache file (tmdb_cache.csv) with a request delay of 0.27 seconds per request to comply

with API rate limits. This caching strategy ensures that data retrieval from the API is only performed once, so subsequent system startups can load data directly from the cache without requiring repeated internet connections.

2.2 Data Preprocessing

Preprocessing proceeded through a defined series of steps aimed at achieving data quality and representational consistency prior to model training. First, user-provided tags per film are aggregated using groupby operations, combining all tags from different users for the same film into a single text representation per film. Second, the genre format is normalized from pipe-separated (e.g., Adventure|Animation|Children) to lowercase space-separated text (e.g., adventure animation children), so that each genre is treated as an independent token by TfidfVectorizer. Third, all film titles are standardized to lowercase, and article positions are corrected (e.g., "Incredibles, The" becomes "The Incredibles") to improve text processing consistency. Fourth, supplementary content data from TMDb API synopsis and cast list is converted to lowercase. Fifth, five film information components are combined into a single combined_features representation: combined_features = title_clean + " " + genres_clean + " " + tag + " " + overview_lower + " " + cast_lower. This combined representation serves as the basis for TF-IDF vectorization in the CBF model. Table 2 illustrates the preprocessing result for the film Toy Story (1995) as a concrete example.

Table 2. Comparison of Data Before and After Preprocessing (Toy Story, 1995)

Attribute	Before Preprocessing	After Preprocessing
genres	Adventure Animation Children Comedy Fantasy	adventure animation children comedy fantasy
title	Toy Story (1995)	toy story
tag	pixar, fun	pixar fun
overview	Led by Woody, Andy's toys live happily...	led by woody, andy's toys live happily...
cast	Tom Hanks, Tim Allen, Don Rickles...	tom hanks, tim allen, don rickles...
combined_features	–	toy story adventure animation... pixar fun led by woody... tom hanks...

2.3 Train-Test-Split

Dataset partitioning followed a per-user stratified approach at an 80:20 ratio. A purely random split risks excluding certain users from the training partition entirely, which would prevent the model from learning their preferences. Keeping each user present in both subsets yields a more equitable evaluation setting where every test-set user has contributed ratings during training. Users whose total rating count fell below five were assigned exclusively to the training partition, as too few test observations would make their evaluation statistically unreliable. From the 100,836 total ratings, 80,668 (80%) were allocated to the training set and 20,168 (20%) to the test set, with all 610 users present in both partitions. The training set serves three purposes: fitting the SVD model, constructing the TF-IDF similarity matrix, and computing each user's user_avg_rating. The test set is reserved solely for evaluation and remains completely isolated from all training and recommendation generation steps.

2.4 Collaborative Filtering (SVD)

CF component was built using the SVD implementation provided by the Surprise library. SVD operates as a matrix factorization method that breaks down the user-item rating matrix R into three constituent matrices, as expressed in Equation (1). This decomposition uncovers latent preference dimensions that are not directly observable from raw rating data and would be inaccessible to neighborhood-based techniques [4].

$$R = U \Sigma V^T \quad (1)$$

In Equation (1), R denotes the $m \times n$ rating matrix spanning m users and n films, U serves as the $m \times k$ matrix that encodes latent user representations, Σ is the $k \times k$ diagonal matrix whose diagonal entries measure the relative importance of each latent dimension, and V^T is the $k \times n$ matrix encoding latent film representations. By jointly optimizing these factor matrices, SVD produces compact encodings that simultaneously capture the underlying characteristics of both users and films. Training configuration: 150 latent factors, 30 epochs, a learning rate of 0.005, a regularization coefficient of 0.02, and a random seed of 42 to ensure reproducibility. Upon completion of training, the model scores every unwatched film for a given user, and the top 200 highest-scoring candidates are forwarded to the hybrid stage.

2.5 Content-Based Filtering (TF-IDF and Cosine Similarity)

The content-based component adopts a weighted multi-feature TF-IDF strategy. Rather than encoding all textual attributes into a single undifferentiated representation, individual TF-IDF matrices are constructed for each of the five content dimensions and then horizontally stacked via scipy's hstack function, with feature-specific multipliers that reflect their relative informational value. The TF-IDF calculation follows Equations (2) and (3) [8]:

$$TF - IDF(t, d) = TF(t, d) \times IDF(t) \quad (2)$$

$$TF(t, d) = 1 + \log(f_{td}), IDF(t) = \log\left(\frac{N}{df_t}\right) \quad (3)$$

Where f_{td} denotes how frequently term t appears in document d , N is the total number of films in the collection, and df_t refers to the count of films in which term t occurs. The weight assigned to each field is: title and genre each receive 0.35 because they are the most stable and informative descriptors, synopsis receives 0.15 for narrative-level semantics, cast receives 0.10 for actor information, and user tags receive 0.05 because they are the sparsest field. The resulting concatenated matrix, of dimensions $9,742 \times 5,500$, is submitted to cosine similarity computation as formalized in Equation (4) [5], yielding a square $9,742 \times 9,742$ matrix in which each cell quantifies the pairwise content affinity between two films.:

$$\text{Cos}(\theta) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}} \quad (4)$$

A user's CBF score profile is formed by averaging the cosine similarity vectors of all films the user rated at or above their `user_avg_rating`, evaluated against every unrated film in the catalogue. The 200 films with the highest mean similarity values are then forwarded as CBF candidates to the hybrid stage.

2.6 Hybrid Score Combination with Personal Threshold Mechanism

The primary innovation of this research is the dual-threshold mechanism based on the user's personal average rating (`user_avg_rating`), calculated from training data for each user. Before combining CF and CBF scores, two-stage candidate filtering is applied. Stage 1 eliminates all films whose CF prediction score falls below `user_avg_rating`; only films with `cf_score` $\geq$ `user_avg_rating` are retained. Stage 2 eliminates films whose global average rating across the entire dataset is also below `user_avg_rating`, as well as films with fewer than 50 ratings (to ensure minimum data reliability). After filtering, The remaining films then have their CF and CBF scores rescaled to $[0,1]$ using min-max normalization, putting both scores on an equal footing before combination. The final hybrid score is computed as shown in Equation (5):

$$\text{Score}_{\text{Hybrid}} = \alpha \times \text{Score}_{\text{CF}} + (1 - \alpha) \times \text{Score}_{\text{CBF}} \quad (5)$$

The value $\alpha = 0.6$ gives greater weight to the CF component, as SVD has been shown to be more effective at capturing comprehensive user preferences based on collective behavior patterns. This weight assignment is consistent with findings by [6] which showed that CF-dominant hybrid configurations produced more personal and relevant recommendations. The top-10 films with the highest final `ScoreHybrid` are then returned as the final recommendation output.

2.7 Cold-Start Handling

New users without any rating history cannot be served by either CF (no user latent factors) or CBF (no liked-film profile) methods, a well-documented challenge in recommendation system research [18]. This cold-start problem is addressed through two complementary and sequential layers that enable the system to gradually build user profile data within the first session. Layer 1 is genre-based onboarding. Upon first login, new users are presented with a genre selection interface where they choose their favorite genres. Based on these selections, the system recommends films using a weighted popularity score calculated by Equation (6), filtered to include only films with release year ≥ 2000 , rating_count ≥ 50 , and complete TMDb content data (non-empty synopsis and cast):

$$\text{final}_{\text{score}_{\text{popular}}} = \text{avg_rating} \times \log_{1p}(\text{rating_count}) \quad (6)$$

The logarithmic transformation $\log_{1p}(\text{rating_count})$ is applied to prevent films with extremely large numbers of ratings from dominating unfairly, allowing high-quality films with moderate rating counts to also appear in recommendations. The combination of `avg_rating` and $\log_{1p}(\text{rating_count})$ balances quality (average rating) and popularity (number of ratings) to produce recommendations that are both well-regarded and sufficiently popular. Layer 2 is explicit feedback collection. From the displayed initial recommendation list, new users actively mark films they have previously watched. Each selected film is stored as a new rating entry in the dataset with a default value of 3.5 (midpoint of the 0.5–5.0 scale). The data is immediately added to the global DataFrame in-memory and written permanently to `ratings.csv`. The system then automatically re-generates hybrid recommendations based on this newly provided rating history, allowing the user to transition from popularity-based cold-start recommendations to personalized hybrid recommendations within a single first session.

2.8 Web Application Implementation

The recommendation system is deployed as a web-based application using the Flask micro-framework with Python as the primary back-end language. Flask was selected for its minimal footprint, design flexibility, and mature ecosystem for embedding machine learning models within web services [2] [19]. SQLite is used as the database for user authentication and genre preference management, selected for its simplicity and zero-configuration deployment.

The application provides nine main routes that handle the complete user flow: `/login` (GET, POST) for user authentication with password hash matching, `/register` (GET, POST) for new account creation with Werkzeug password encryption, `/pilih_genre<user_id>` (GET) for genre selection interface for new users, `/select_genres` (POST) for saving genre selections to database and generating cold-start recommendations, `/existing_user` (GET) for returning user profile page, `/recommend` (POST) for triggering hybrid recommendation generation, `/result` (GET) for displaying top-10 recommendation results, `/watched` (GET) for viewing rating history with interactive genre filter and sorting, and `/logout` (GET) for terminating the active session.

The user interface is constructed with HTML, CSS, and the Jinja2 template engine coordinated by Flask, spanning seven main pages with an intuitive and responsive visual design. New user accounts are assigned `userId` values above 610 (the dataset

maximum) to avoid conflicts with existing dataset users. After new users provide their rating feedback via the /save_watched route, the SVD model is automatically retrained on the updated dataset to incorporate the new interaction data and generate improved personalized recommendations.

2.9 Evaluation Methods

Four evaluation groups were used to assess the system from both quantitative and qualitative perspectives. The first group measures Top-N Recommendation Relevance by comparing CF, CBF, and Hybrid Filtering on a random sample of 50 test users (random_state=42). Four metrics are reported: Precision@10 reflects the proportion of top-10 recommended films that are genuinely relevant to the user, Recall@10 captures how much of the total relevant films in the test set were successfully retrieved within the top-10 list, F1@10 consolidates precision and recall into a single harmonic mean, and NDCG@10 evaluates ranking quality by attributing greater weight to relevant films positioned higher in the ranked list. NDCG is computed via Equations (7) and (8):

$$DCG@K = \sum_{i=1}^K \frac{1}{\log_2(i+2)} \times rel_i \quad (7)$$

$$NDCG@K = \frac{DCG@K}{IDCG@K} \quad (8)$$

Three different ground truth scenarios are applied. In the Genre scenario, relevant films are defined as those in the test data that share at least one genre with the user's top-5 favorite genres identified from training data films rated $\geq$ user_avg_rating. In the Rating scenario, relevant films are those in the test data to which the user assigned a rating $\geq$ user_avg_rating. In the Combined (AND) scenario, a film is considered relevant only when it simultaneously satisfies both the genre and rating criteria described above. The second group quantifies rating prediction accuracy using RMSE and MAE, both calculated across the complete set of 20,168 test observations as defined in Equations (9) and (10):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{r}_i - r_i)^2} \quad (9)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{r}_i - r_i| \quad (10)$$

The third group reports Genre Precision, a supplementary metric sharing the same precision formula as the first group but with relevance grounded solely in genre alignment between recommended films and each user's training-derived genre profile, measured over 50 randomly selected training users. The fourth group conducts User Testing, enlisting ten respondents with varied film-watching backgrounds to rate the perceived relevance of the system's output for a designated reference user on a scale from 1 to 10.

3 Results and Discussion

3.1 Functional Testing Results

Black box testing was performed on all 13 main system functions. All functions produced results consistent with specified functional requirements without any failures detected. Table 3 presents the complete functional testing results.

Table 3. Black Box Functional Testing Results

No.	Function Tested	Expected Result	Status
1	Load MovieLens dataset	100,836 ratings, 9,742 films loaded successfully	Passed
2	TMDb API data caching	Synopsis, cast, poster saved to tmdb_cache.csv	Passed
3	Preprocessing & combined_features	Each film has complete combined_features representation	Passed
4	Per-user train-test split (80:20)	All 610 users represented in both subsets	Passed
5	SVD model training (n_factors=150)	Model trained can predict ratings for unseen films	Passed
6	TF-IDF matrix & cosine similarity	9,742×9,742 similarity matrix built successfully	Passed
7	Dual-threshold mechanism	Films below user_avg_rating correctly eliminated	Passed
8	Weighted hybrid combination ($\alpha=0.6$)	Top-10 recommendations generated with final score	Passed
9	New user registration	Account created with encrypted password, userId > 610	Passed
10	Login & authentication	Session started redirect based on user status	Passed
11	Cold-start genre onboarding	Modern films recommended by selected genres	Passed

No.	Function Tested	Expected Result	Status
12	Recommendation page information	Poster, title, genre, year, score all displayed	Passed
13	Viewing history with filter & sort	Genre filter and rating sort function correctly	Passed

The successful completion of all 13 functional tests confirms that the system architecture built using Flask, SQLite, Surprise library, and Scikit-learn has been well-integrated. Key aspects highlighted by functional testing include: (1) TMDb API data caching proved effective in maintaining system efficiency, with data loaded from the local cache on subsequent starts, (2) per-user stratified splitting guarantees all users contribute to the training data, enabling the SVD model to learn every user's preferences, and (3) the dual-threshold mechanism based on `user_avg_rating` operates as designed, correctly filtering candidates before hybrid score combination.

3.2 Recommendation Performance Evaluation

Performance testing compared three approaches simultaneously against a random sample of 50 users from test data (`random_state=42`). Tables 4, 5, and 6 present evaluation results for each ground truth scenario, measuring the performance of Collaborative Filtering (CF), Content-Based Filtering (CBF), and Hybrid Filtering (CF-CBF).

Table 4. Evaluation Results – Genre Scenario

Method	Precision@10	Recall@10	F1@10	NDCG@10
Collaborative Filtering (SVD)	0.17	0.07	0.10	0.19
Content-Based Filtering (TF-IDF)	0.15	0.12	0.13	0.21
Hybrid Filtering (CF-CBF)	0.27	0.09	0.13	0.32

Table 5. Evaluation Results – Rating Scenario

Method	Precision@10	Recall@10	F1@10	NDCG@10
Collaborative Filtering (SVD)	0.20	0.09	0.12	0.23
Content-Based Filtering (TF-IDF)	0.14	0.15	0.15	0.23
Hybrid Filtering (CF-CBF)	0.24	0.13	0.17	0.29

Table 6. Evaluation Results – Combined Scenario (Genre+Rating AND)

Method	Precision@10	Recall@10	F1@10	NDCG@10
Collaborative Filtering (SVD)	0.19	0.10	0.13	0.23
Content-Based Filtering (TF-IDF)	0.10	0.12	0.11	0.13
Hybrid Filtering (CF-CBF)	0.23	0.14	0.17	0.30

Hybrid Filtering achieves the leading Precision@10 and NDCG@10 values across all three evaluation scenarios without exception. In the Genre scenario, Hybrid attains Precision@10=0.27 and NDCG@10=0.32, outperforming CF (0.17, 0.19) and CBF (0.15, 0.21). Under the Rating scenario, it posts Precision@10=0.24 and NDCG@10=0.29. In the most rigorous Combined scenario, it records Precision@10=0.23 and NDCG@10=0.30. The F1@10 of 0.17 achieved in the Rating and Combined scenarios reflects the strongest precision-recall equilibrium among all approaches.

The slightly lower Recall@10 of Hybrid Filtering compared to CBF in some scenarios is an expected and acceptable trade-off from the selective personal threshold mechanism. This mechanism prioritizes recommendation precision over coverage filtering out films that, while potentially genre-adjacent, do not meet the user's personal quality standard. This is reflected in consistently superior Precision and NDCG values, which are more critical metrics for user experience since users typically focus on the top few recommendations. These findings are consistent with those of [8] and [3], which similarly documented hybrid precision superiority over individual methods.

3.3 Genre Precision Evaluation

Genre Precision is a complementary metric that measures the proportion of recommended films whose genres align with the user's genre preference profile built from training data. This metric provides an additional perspective on recommendation quality independent of the test set ground truth. Table 7 presents Genre Precision results for all three methods, evaluated against 50 randomly sampled training users.

Table 7. Genre Precision Evaluation Results

Method	Genre Precision
Collaborative Filtering (SVD)	0.87
Content-Based Filtering (TF-IDF and Cosine Similarity)	0.95
Hybrid Filtering (CF-CBF)	0.90

Content-Based Filtering achieves the highest Genre Precision (0.95), as it directly compares content feature vectors where genre is the most heavily weighted component ($W=0.35$) in the combined features representation [17]. This explains its natural tendency to recommend films with similar genre profiles. All three methods produce Genre Precision above 0.85, indicating that recommendations from all approaches are generally aligned with user genre preferences. Hybrid Filtering (0.90) surpasses CF (0.87) while remaining near the CBF level, demonstrating that incorporating the CBF component into the hybrid system successfully maintains high genre alignment without sacrificing the personalization benefits of the CF component.

3.4 Global vs. Personal Threshold Comparison

To empirically demonstrate the contribution of the personal threshold mechanism, additional testing was conducted comparing performance under two conditions: (1) global threshold with a fixed value of 4.0, representing the configuration used in most previous studies, and (2) personal threshold based on `user_avg_rating` computed adaptively per user from training data. Both conditions were evaluated using the same sample (50 users, `random_state=42`), the same three ground truth scenarios, and all metrics.

Table 8. Evaluation Comparison: Global Threshold (4.0) vs. Personal Threshold

Metric / Scenario	CF Global	CBF Global	Hybrid Global	CF Personal	CBF Personal	Hybrid Personal
Precision@10 – Genre	0.20	0.14	0.16	0.17	0.15	0.27
Recall@10 – Genre	0.06	0.04	0.07	0.07	0.12	0.09
F1@10 – Genre	0.10	0.07	0.10	0.10	0.13	0.13
NDCG@10 – Genre	0.22	0.15	0.25	0.19	0.21	0.32
Precision@10 – Rating	0.19	0.14	0.16	0.20	0.14	0.24
F1@10 – Rating	0.10	0.15	0.10	0.12	0.15	0.17
NDCG@10 – Rating	0.20	0.23	0.23	0.23	0.23	0.29
Precision@10 – Combined	0.22	0.12	0.20	0.19	0.10	0.23
NDCG@10 – Combined	0.26	0.23	0.29	0.23	0.13	0.30

Table 9. Genre Precision: Global Threshold vs. Personal Threshold

Method	Genre Precision (Global Threshold = 4.0)	Genre Precision (Personal Threshold)
Collaborative Filtering (SVD)	0.87	0.87
Content-Based Filtering (TF-IDF)	0.95	0.95
Hybrid Filtering (CF-CBF)	0.40	0.90

The most striking finding is the dramatic difference in Hybrid Filtering Genre Precision: only 0.40 under the global threshold (4.0), compared to 0.90 under the personal threshold. Under the global threshold, the CF component which dominates the hybrid score with $\alpha=0.6$ pulls recommendations toward films with high global predicted ratings that may not necessarily align with individual genre preferences. The fixed threshold (4.0) is above the average `user_avg_rating` for many users, meaning fewer films pass the filter, and those that do tend to be globally popular films regardless of genre alignment.

Conversely, with the personal threshold, the filtering process is calibrated to each user's actual rating standard. Users who tend to give lower ratings (e.g., `user_avg_rating=3.2`) will have a wider candidate pool including more genre-diverse films, while users who give higher ratings (e.g., `user_avg_rating=4.2`) will receive a more selective pool of high-quality films. This structural difference reshapes the composition of the candidate pool before score combination, ensuring that the resulting recommendations better align with each user's individual genre profile and quality expectations.

For CF and CBF, Genre Precision is identical (0.87 and 0.95) under both threshold conditions. This is expected because the threshold mechanism does not modify how CF and CBF generate their individual scores it only filters the candidates

presented to the hybrid combination step. This further confirms that the personal threshold's primary effect is on the hybrid combination process, specifically in aligning the candidate pool with individual user preferences before the weighted score aggregation.

3.5 SVD Rating Prediction Accuracy

Rating prediction accuracy was evaluated using RMSE and MAE metrics against all 20,168 test data entries. Table 10 presents the results.

Table 10. RMSE and MAE Evaluation Results of SVD Model

Metric	Value
RMSE (Root Mean Squared Error)	0.88
MAE (Mean Absolute Error)	0.67

With hyperparameters 150 latent factors, 30 epochs, a learning rate of 0.005, a regularization coefficient of 0.02, SVD model achieves RMSE=0.88 and MAE=0.67 on the complete test partition. Both values are consistent with the dataset's 98.3% sparsity level and align well with figures reported in comparable prior work.[4] reported RMSE=0.877 on a MovieLens dataset with similar characteristics, and [3] reported RMSE=1.0802 on an anime dataset. The lower RMSE value in this research compared to [3] is partly attributable to the use of a denser and more consistent MovieLens dataset.

RMSE applies greater penalty to substantial individual errors by squaring each deviation before the averaging process, which increases its sensitivity toward outlier predictions, MAE assigns equal weight to all errors, offering a more direct indication of average prediction deviation. The gap between the two (0.88 vs. 0.67) signals the presence of some outlier predictions, which is expected with such a sparse matrix. Even so, both values are adequate for the CF component's role in the hybrid system, where what matters most is whether the model ranks unseen films correctly relative to one another, not whether each absolute predicted score is exact.

3.6 User Testing Results

User testing was conducted to complement the quantitative evaluation with a subjective assessment of recommendation relevance from the direct user perspective. The reference profile selected was userId 447 from the MovieLens dataset. This profile was chosen because it has a representative and diverse rating history: 79 films watched, personal average rating of 3.88, and dominant genres of Drama (34 films), Action (33 films), and Thriller (31 films). The active threshold for this user is: cf_score ≥ 3.88 , avg_rating_global ≥ 3.88 , and rating_count ≥ 50 . Table 11 presents the complete list of 10 recommended films generated by the Hybrid Filtering system for this user profile.

Table 11. Top-10 Hybrid Filtering Recommendations for userId 447

Rank	Film Title	Genre	Year	Final Score
1	Life Is Beautiful (La Vita è bella)	Comedy, Drama, Romance, War	1997	0.93
2	Star Wars: Episode IV – A New Hope	Action, Adventure, Sci-Fi	1977	0.84
3	Indiana Jones and the Last Crusade	Action, Adventure	1989	0.83
4	The Silence of the Lambs	Crime, Drama, Thriller	1991	0.83
5	Goodfellas	Crime, Drama	1990	0.79
6	Heat	Action, Crime, Drama, Thriller	1995	0.78
7	The Usual Suspects	Crime, Mystery, Thriller	1995	0.77
8	Apocalypse Now	Drama, War	1979	0.75
9	A Beautiful Mind	Drama	2001	0.74
10	The Deer Hunter	Drama, War	1978	0.72

All 10 recommended films align well with the user's dominant genres (Drama, Action, Thriller). Films such as Life Is Beautiful, The Silence of the Lambs, and Goodfellas are classic films with high global average ratings that also match the user's genre profile and exceed the personal threshold of 3.88. This demonstrates that the personal threshold mechanism effectively combines personalization (user_avg_rating filter) with quality assurance (rating_count ≥ 50 and global avg_rating filter). Ten respondents with diverse viewing backgrounds were asked to rate the overall relevance of the 10 recommendations against the specified user profile (userId 447) on a scale of 1–10. Table 12 presents the detailed results from all respondents.

Table 12. User Testing Assessment Results

Respondent	Age (Year)	Score
Res1	22	9
Res2	21	10
Res3	22	8
Res4	21	8
Res5	21	6
Res6	23	8
Res7	22	10
Res8	22	7
Res9	22	7
Res10	23	9
Average Score		8.2

The assessment from all 10 respondents produces an average score of 8.2 out of 10. This value indicates that the recommendation system generates results considered highly relevant subjectively, confirming the effectiveness of the Hybrid Filtering approach with the personal threshold mechanism. Respondent feedback reveals several interesting patterns. First, most respondents with frequent viewing habits gave scores of 8–10, reflecting strong alignment between recommendations and genre expectations. Second, some respondents noted that the ranking order could be further optimized, particularly regarding the placement of Drama-genre films. Third, one respondent (R5) gave a lower score of 6, noting that 4 of the 10 films felt less fitting. These observations provide valuable input for future system improvements, particularly in refining the ordering algorithm and expanding genre diversity within recommendations.

3.7 Discussion

The overall evaluation results confirm that Hybrid Filtering with a personal threshold mechanism consistently outperforms both standalone methods across all evaluation dimensions [14]. The superiority is most pronounced in Precision@10 and NDCG@10, where Hybrid Filtering achieves the highest values in all three ground truth scenarios. These findings align with the broader literature: [8] reported Hybrid CF-CBF achieving hit rate 0.8042 and MAP@10 0.3159 versus CF alone (0.7904, 0.3054), and [3] reported Precision@10 of 82.77% for hybrid versus individual methods.

The most critical finding of this research is the empirical evidence for the personal threshold mechanism's contribution. The improvement in Hybrid Filtering Genre Precision (from 0.40 to 0.90) when switching from a global threshold to a personal threshold demonstrates that threshold personalization fundamentally changes the composition of the candidate pool. Under the global threshold, the CF component's tendency to predict high ratings for globally popular films dominates the hybrid output, pulling recommendations away from individual genre preferences. The personal threshold corrects this by anchoring the candidate selection to each user's actual rating standard, ensuring that genre diversity from the CBF component can meaningfully influence the final recommendations.

This finding directly addresses the gap identified by [4], whose static-weight hybrid produced only Precision@10 = 0.004 demonstrating that the combination mechanism itself is insufficient without personalized candidate selection. The current research shows that by implementing adaptive threshold filtering before score combination, the same weighted hybrid formula ($\alpha=0.6$) can produce dramatically better results. This confirms the hypothesis stated in the Introduction: that an adaptive candidate selection mechanism is the key factor enabling CF-CBF combination to achieve true synergy.

The Genre Precision results provide an additional layer of validation beyond the test-set-based metrics. While Precision@10 and NDCG@10 measure relevance based on actual user behavior (ratings in the test set), Genre Precision measures content alignment directly from the recommendation output against the user's genre profile built from training data. The consistency between these two perspectives Hybrid Filtering performing best on test-set metrics while maintaining near CBF level genre alignment confirms that the system successfully balances personalization depth (CF component) with content relevance (CBF component).

4 Conclusions

1. Hybrid Filtering system combining SVD-based CF and TF-IDF with cosine similarity-based CBF was successfully constructed and deployed as a Flask web application on the MovieLens ml-latest-small dataset (100,836 ratings, 610 users, 9,742 films). Black box testing across all 13 core functions produced no failures, verifying that every component from dataset loading and model training to the threshold filter and the user interface operates as intended.
2. Hybrid Filtering consistently produces the highest Precision@10 and NDCG@10 across all three evaluation scenarios: Genre (P@10=0.27, NDCG=0.32), Rating (P@10=0.24, NDCG=0.29), and Combined (P@10=0.23, NDCG=0.30).

The F1@10 of 0.17 in the Rating and Combined scenarios shows the best precision-recall balance. The SVD model yields RMSE=0.88 and MAE=0.67, confirming adequate rating prediction accuracy as the CF component basis. User testing with 10 respondents produced an average score of 8.2/10, validating subjective relevance.

3. The dual-threshold mechanism based on user_avg_rating (personal threshold) makes a real and significant contribution, raising Hybrid Filtering Genre Precision from 0.40 (global threshold) to 0.90. These improvements quantitatively confirm that adaptive candidate selection is the key enabling factor for CF-CBF combination to achieve genuine synergy, directly addressing the weakness of static global thresholds identified in prior studies.
4. The cold-start problem for new users was effectively resolved through two sequential mechanisms: genre-based onboarding supported by a weighted popularity score ($\text{avg_rating} \times \log_{1p}(\text{rating_count})$) with quality filters (year $\geq$ 2000, rating_count $\geq$ 50, complete TMDb metadata), and explicit feedback collection that enables a smooth progression from popularity-driven recommendations to fully personalized hybrid recommendations within a single initial session.

5 Suggestions

Based on the findings and limitations of this research, two suggestions are proposed for future work. First, the dataset can be scaled to a larger version such as MovieLens 25M to improve system coverage and recommendation diversity. Second, the current static hybrid weight ($\alpha = 0.6$ for CF, 0.4 for CBF) can be replaced with an adaptive weighting mechanism that automatically adjusts the alpha value based on the volume of user interaction data assigning greater weight to the CF component for established users with sufficient rating history, and greater weight to CBF for users with limited data.

6 Acknowledgments

The author expresses gratitude to the Informatics Study Program, Faculty of Information Technology, Universitas Kristen Duta Wacana, for the opportunities, facilities, and academic support provided throughout the study period.

BIBLIOGRAPHY

- [1] E. W. Nugroho, A. N. Rohman, and D. Prabowo, "SISTEM REKOMENDASI FILM INDONESIA MENGGUNAKAN COLLABORATIVE FILTERING," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 5, pp. 9070–9076, Oct. 2025.
- [2] S. Agni, P. Raswidhyantoro, M. Idhom, and A. N. Sihananto, "Implementasi Sistem Rekomendasi Film Menggunakan Algoritma Matrix Factorization Berbasis Flask untuk Personalisasi Konten di Era Disrupsi Digital," *Seminar Nasional Informatika Bela Negara (SANTIKA)*, vol. 5, no. 2, pp. 227–231, 2025.
- [3] S. Hendrawan and Yakub, "OPTIMASI SISTEM REKOMENDASI FILM MENGGUNAKAN METODE HYBRID FILTERING," *POTERS: Proceedings of Technology, Engineering and Computers*, vol. 1, no. 2, pp. 316–323, Jul. 2025, [Online]. Available: <https://jurnal.ubd.ac.id/index.php/poters/index>
- [4] A. P. Ariani, D. A. Handayani, P. M. Setyowati, D. K. Nugroho, and M. N. Fauzy, "Evaluasi Komparatif Hybrid Filtering dan Model-Based SVD pada Sistem Rekomendasi Film Menggunakan Dataset MovieLens," *Jurnal Pustaka Data*, vol. 6, no. 1, pp. 22–31, Feb. 2026, doi: 10.55382/jurnalpustakadata.v6i1.1723.
- [5] D. Velamentosa and E. Zuliarso, "SISTEM REKOMENDASI FILM MENGGUNAKAN METODE CONTENT-BASED FILTERING," *Jurnal Mahasiswa Teknik Informatika*, vol. 9, no. 2, pp. 2918–2922, Apr. 2025.
- [6] M. F. Hartono, Opitasari, and I. Firdaus, "Pengembangan Hybrid Recommender System Menggunakan Scikit-Learn dan Pandas untuk Rekomendasi Film," *Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI)*, vol. 07, no. 2, pp. 408–416, 2026.
- [7] M. Fajriansyah, P. P. Adikara, and A. W. Widodo, "Sistem Rekomendasi Film Menggunakan Content Based Filtering," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 6, pp. 2188–2199, May 2021, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [8] H. H. Arfisko and A. T. Wibowo, "Sistem Rekomendasi Film Menggunakan Metode Hybrid Collaborative Filtering Dan Content-Based Filtering," *e-Proceeding of Engineering*, vol. 9, no. 3, pp. 2149–2159, Jun. 2022.
- [9] R. Widayanti, M. H. R. Chakim, C. Lukita, U. Rahardja, and N. Lutfiani, "Improving Recommender Systems using Hybrid Techniques of Collaborative Filtering and Content-Based Filtering," *Journal of Applied Data Sciences*, vol. 4, no. 3, pp. 289–302, Sep. 2023.
- [10] C.-Y. Lai, N. Hashim, and L. D. Krisnawati, "Hybrid-Based Movie Recommender System: Techniques, Case Studies, Evaluation Metrics, and Future Trends," *Journal of Informatics and Web Engineering*, vol. 4, no. 3, pp. 90–110, Oct. 2025, doi: 10.33093/jiwe.2025.4.3.5.
- [11] I. K. G. A. Pramarta and Z. K. A. Baizal, "Hybrid Recommender System Using Singular Value Decomposition and Support Vector Machine in Bali Tourism," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 7, no. 2, pp. 408–418, 2022.
- [12] K. R. Putra and M. A. Rachman, "Perbandingan Metode Content-based, Collaborative dan Hybrid Filtering pada Sistem Rekomendasi Lagu," *MIND Journal*, vol. 9, no. 2, pp. 179–193, 2024, doi: 10.26760/mindjournal.v9i2.179-193.
- [13] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," *Journal of Big Data*, vol. 9, no. 1, pp. 1–36, 2022, doi: 10.1186/s40537-022-00592-5.
- [14] U. Muntaha, J. K. Silitonga, E. C. Simanjuntak, M. S. Wisnubroto, F. Farid, and M. C. Untoro, "A Hybrid Content-Based and Collaborative Filtering Approach for Recommending Optimal Movie Release Months," *Journal of Applied Informatics and Computing (JAIC)*, vol. 10, no. 2, pp. 2548–6861, 2026.

- [15] T. W. I. Permadani, D. A. Prasetya, and E. Daniati, "Sistem Rekomendasi Film Berdasarkan Genre Menggunakan Metode Content-Based Filtering dengan Algoritma Cosine Similarity," *INOTEK*, vol. 9, 2025.
- [16] K. S. Y. Putri, I. M. A. D. Suarjaya, and W. O. Vihikan, "Sistem Rekomendasi Skincare Menggunakan Metode Content Based Filtering dan Collaborative Filtering," *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 4, no. 3, pp. 764–774, 2024, doi: 10.51454/decode.v4i3.601.
- [17] R. A. Sari, S. F. Isnaini, and E. Seniwati, "Sistem Rekomendasi Film Menggunakan Content Based Filtering," *IJCSR: The Indonesian Journal of Computer Science Research*, vol. 4, no. 1, pp. 19–27, 2025.
- [18] Y. I. Lubis, D. J. Napitupulu, and A. S. Dharma, "Implementasi Metode Hybrid Filtering (Collaborative dan Content-based) untuk Sistem Rekomendasi Pariwisata," 2020.
- [19] K. A. M. Gala and A. Prapanca, "Pengembangan Sistem Rekomendasi Film Berbasis Web Menggunakan Algoritma Collaborative Filtering," *Journal of Informatics and Computer Science*, vol. 7, no. 1, pp. 50–56, 2025.