# A Comparative Analysis of Collaborative and Content-Based Filtering using k-Nearest Neighbors for Anime Recommendation

Jay Mark V. Agsoy[1], Neil Robert J. Morales[2], Lex Jayshier Tesorero[3]

[123]*University of Mindanao, Matina, Davao City, Philippines*
*j.agsoy.545281@umindanao.edu.ph,*
*n.morales.544543@umindanao.edu.ph, l.tesorero.546228@umindanao.edu.ph*

**Abstract - The global expansion of anime and the proliferation of digital streaming platforms have resulted in a vast content ecosystem, often leading to choice overload for viewers.1 To address this, recommender systems utilizing Collaborative Filtering (CF) and Content-Based Filtering (CB) are employed; however, their comparative effectiveness remains under-researched within the specific context of anime's unique metadata and community-driven trends. This study provides a comparative analysis of CF and CB filtering by holding the algorithmic architecture constant through the use of k-Nearest Neighbors (k-NN). Using a 2023 MyAnimeList dataset, we implemented an item-based CF approach using rating patterns and a CB approach utilizing metadata features such as genres, studios, and TF-IDF vectorized synopses. Both models were evaluated using Top-N ranking metrics, including Precision@k, Recall@k, Mean Average Precision (MAP@k), and R-Precision. Empirical results indicate that the Content-Based model significantly outperformed the Collaborative model in ranking quality and precision, achieving a Precision@10 of 0.1820 compared to CF's 0.0556. While CF exhibited higher growth in Recall at larger recommendation windows (k=100), it suffered from data sparsity that hindered ranking stability. Qualitative analysis confirmed that both models successfully identified semantically relevant content, such as sequels and genre-peers. The study concludes that while metadata provides a more resilient signal for immediate relevance in anime, the discovery potential of collaborative data suggests that a hybrid approach is necessary to optimize the balance between precision and content variety.**

**Keywords - Recommender Systems, Collaborative Filtering, Content-Based Filtering, k-Nearest Neighbors, Anime, Machine Learning, Comparative Analysis.**

## 1. Introduction

Anime is a form of media as well as a type of popular entertainment from Japanese popular culture [1]. It is perceived as an international phenomenon and has become mainstream, emerging as a new and important market reaching a record 3.34 trillion yen in 2023. This explosion, driven by rapid global expansion and the growth of digital streaming platforms, has made larger content libraries available to viewers worldwide [2][3]. As these services expand their libraries and reach more audiences, the volume of accessible anime content has expanded to a point where audiences face increasingly complex content ecosystems [4], which contribute to choice overload, a condition in which an abundance of options leads to increased decision difficulty and reduced satisfaction [5].

To combat this, services use recommenders, which offer relevant suggestions for when users need to choose an item from extensive collections [6][7]. When these recommenders are well-designed, they can reduce user effort, improve decision confidence, and enhance overall satisfaction when navigating large content collections [8]. Recommender systems primarily use two paradigms, Collaborative Filtering (CF) and Content-Based Filtering (CB). CF recommends items to target users with what other users with similar preferences liked in the past, whereas CB focuses on the items themselves, through attributes such as title, genre, studio, or synopsis, and recommends items that are similar to what the user likes [7]. While both paradigms are well-established, their comparative effectiveness within the specific domain of anime remains unclear due to the medium's nature. Anime is characterized by rich metadata, which favors CB approaches; however, it is also driven by community engagement and seasonal "hype" cycles, which heavily favor CF approaches [1].

CF suffers from the "cold start" problem, where new users or new items have insufficient interaction data, making it difficult for the system to predict preferences accurately [9][10][11]. This limitation often results in generic or irrelevant recommendations for first-time users, reducing engagement and satisfaction. Meanwhile, CB also has a "cold start" problem for new users; it suffers more from "over-specialization", where the system repeatedly recommends items very similar to a user's existing preferences, limiting diversity and hindering discovery of novel content [12]. A hybrid of the two is the common way recommender systems solve this; however, it is unclear which limitation (cold start vs over-specialization) is less detrimental in the context of anime recommendation [12]. As such, this study aims to provide a comparative analysis of Collaborative and Content-Based Filtering using k-Nearest Neighbours (k-NN), an algorithm used in both paradigms to evaluate them for anime recommendation.

1.2. Related Work

Recommendation systems are widely studied in information retrieval and machine learning, with Collaborative Filtering (CF) and Content-Based Filtering (CB) as the main paradigms. CF predicts user preferences based on interactions and similarities between users or items, while CB recommends items using metadata and past user preferences [13]. Each method has strengths and limitations that affect recommendation quality.

1.2.1 Collaborative Filtering

CF has been effectively applied in anime and media recommendation, leveraging user similarity to provide personalized suggestions [13]. However, it faces challenges such as cold-start and data sparsity, which reduce performance for new users or items [14]. Variants include user-based, item-based, and model-based approaches, such as matrix factorization, often using similarity measures like cosine similarity or Pearson correlation [15].

1.2.2 Content-Based Filtering

CB approaches focus on item features such as genre, synopsis, or other descriptive metadata to recommend similar items [16]. While effective in exploiting rich metadata, CB can suffer from over-specialization, limiting diversity and reducing opportunities for discovering novel content [17].

1.2.3 Comparative and Hybrid Studies

Hybrid methods combine CF and CB to mitigate individual limitations, sometimes integrating deep learning for improved personalization [18]. Despite these advances, few studies systematically compare CF and CB using a consistent algorithmic framework such as k-Nearest Neighbors (k-NN) for anime recommendation [19]. The present study addresses this gap, applying k-NN for both paradigms and evaluating their performance using uniform preprocessing, feature engineering, and evaluation metrics.

1.3. Objectives of the Study

1.3.1. To apply appropriate preprocessing for both Collaborative Filtering and Content-Based Filtering.

1.3.2. To use k-Nearest Neighbors with each respective preprocessed data.

1.3.3. To evaluate and compare the performance of the two recommendation algorithms.

**2. Methodology**

2.1. Data Gathering

Our data is sourced from Kaggle. Specifically, we used the “anime-dataset-2023.csv” and “users-score-2023.csv” datasets available at https://www.kaggle.com/datasets/dbdmobile/myanimelist-dataset/.

**2.2. Data Analysis**

The anime dataset used in this study consists of two main components: a user–anime interaction dataset and an anime metadata dataset, both of which are commonly utilized in recommendation system research. The user–anime interaction dataset records explicit feedback in the form of ratings, where each row represents a rating assigned by a user to a specific anime. Ratings range from 1 to 10, and, as shown in Table 1, the dataset contains no missing values for user IDs, anime IDs, or ratings, ensuring data completeness for interaction-based analysis.

In Table 1, user IDs range from 1 to 1,291,097, while anime IDs range from 1 to 56,085. The rating values have a mean of 7.62 and a median of 8, suggesting that users generally provide favorable ratings.

Despite the richness of the metadata, several attributes contain missing values, as shown in Table 2. Notably, english_name (58.53%) has a high proportion of missing entries, while others like genres (19.79%) and synopsis (18.21%) are partially incomplete. In contrast, core numerical attributes such as episodes exhibit relatively low missing rates (below 3%). This imbalance suggests that while metadata incompleteness may limit certain analyses, the remaining structured features still provide sufficient information to support content-based and hybrid recommendation models.

*Table 1. Statistical Analysis of the Dataset (users-score-2023)*

| Column | Missing Value | Min | Max | Median | Mean | Std Dev |
|---|---|---|---|---|---|---|
| User ID | 0 (0%) | 1.00 | 1291097.00 | 387978.00 | 440384.273792 | 366946.885298 |
| Anime ID | 0 (0%) | 1.00 | 56085.00 | 4726.00 | 9754.686305 | 12061.964645 |
| Rating | 0 (0%) | 1.00 | 10.00 | 8.00 | 7.622930 | 1.661510 |

*Table 2. Statistical Analysis of the Dataset (anime-dataset-2023)*

| Column | Min | Max | Median | Mean | Std Dev |
|---|---|---|---|---|---|
| anime_id | 1.0 | 56085.0 | 34628.0 | 29776.709014 | 17976.07629 |

*Table 3. Missing Values of the Dataset (anime-dataset-2023)*

| Column | Missing Values | Percentage % |
|---|---|---|
| anime_id | 0 | 0.00% |
| name | 0 | 0.00% |
| english_name | 14577 | 58.53% |
| other_name | 128 | 0.51% |
| genres | 4929 | 19.79% |
| synopsis | 4535 | 18.21% |
| type | 74 | 0.30% |
| episodes | 611 | 2.45% |
| rating | 669 | 2.69% |

2.3.1. Handling Missing Values

For Collaborative Filtering (CF), the users-score-2023 dataset, and for Content-Based (CB) filtering, the anime-dataset-2023 dataset, both share a similar cleaning framework to handle duplicates, invalid entries, and indicators of missing or unknown values. While the user–anime interaction dataset contains no missing metadata and is largely complete, the anime metadata dataset (anime-dataset-2023) includes a substantial number of missing or unknown values, represented by a wide variety of placeholder strings such as “Unknown”, “N/A”, “?”, or “No description available”. Depending on the column, the row was either dropped or filled with a marker.

2.3.2. Handling Outliers

For CF, we focused on removing users who provided no discriminative information, such as bots or users who rated every show identically. Users with a standard deviation of zero (0 variance) were excluded from the dataset, as their lack of preference variation contributes nothing to the collaborative filtering similarities.

For CB, several columns not essential for similarity computation were removed, leaving only a small set of relevant features. No outlier treatment was applied to the other retained features.

2.3.3 Feature Engineering

For CF, to reduce sparsity and noise, we retained only anime that had received at least 50 ratings (removing niche or obscure titles) and users who had rated at least 10 shows (removing inactive users). Finally, we dropped descriptive columns such as anime_title and username, keeping only user_id, anime_id, and rating to construct the interaction matrix.

During feature engineering, a large number of metadata columns were removed to reduce noise and retain only attributes that contribute meaningfully to content-based similarity. The remaining features were selected to capture semantic content, format, pacing, and production-related characteristics of anime titles. The final set of retained columns includes identifiers, textual descriptions, categorical attributes, and a small number of numeric features. Although anime_id was retained for reference and indexing purposes, it was excluded from feature vectors used in similarity computation.

For content-based filtering (CB), multi-label categorical features such as genres and ratings were encoded using One-Hot Encoding via Multi-Label Binarization, creasssting binary vectors indicating the presence or absence of each unique category. Concurrently, the textual synopsis was vectorized using Term Frequency–Inverse Document Frequency (TF-IDF) to quantify narrative content and reduce the impact of common words [20][21]. Single-label categorical attributes, including anime type and age rating, were encoded using one-hot or ordinal encoding depending on their inherent ordering. Numeric features such as episode count were scaled to represent structural and pacing similarities and to prevent magnitude differences from biasing similarity calculations.

2.3.4. Data Splitting

To objectively evaluate the generalization capability of the proposed models, the dataset was partitioned into training and testing subsets using an 80:20 ratio. For collaborative filtering (CF), we utilized the train_test_split function from the Scikit-Learn library with stratified sampling [21] based on user IDs. This method ensures that 80% of each user's ratings are assigned to the training set to build the User-Item matrix, while the remaining 20% are withheld for testing. Such

stratification is crucial to avoid the "cold-start" problem during evaluation, ensuring that every user in the test set has a known history in the training set for generating recommendations.

For content-based filtering (CB), the dataset was similarly split into training and testing subsets using the same 80:20 ratio, with 80% of the anime metadata used for training and 20% reserved for testing. This separation ensures that the content representations learned during training are evaluated on unseen anime entries, allowing an objective assessment of the model's generalization capability. The split was performed after data cleaning to preserve consistency across both subsets.

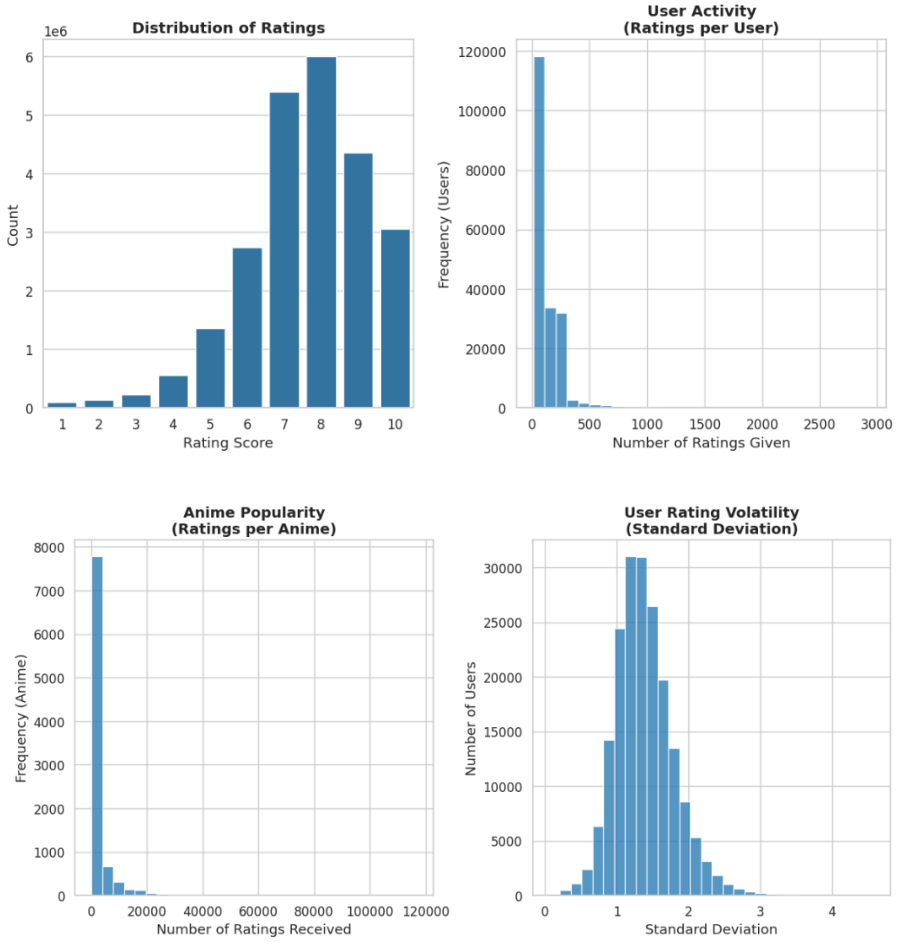


*Figure 1: Dataset distribution after handling missing data and outliers.*

**2.4. Algorithms**

2.4.1 k-Nearest-Neighbors

We used k-NN, which uses proximity to classify or predict a grouping of an individual data point [22]. We selected k-NN as the common baseline for both the CF and CB models, allowing us to isolate and evaluate the performance of each filtering approach without introducing confounding variables from differing model architectures.

2.4.1.1. Collaborative Filtering Approach

In this study, we employed an Item-Based Collaborative Filtering approach. We constructed a sparse interaction matrix R of dimensions $(items \times users)$, where each row vector represents a specific anime's ratings across the entire user base. The k-NN algorithm identifies the k most similar items based on rating patterns using Cosine Similarity [23]. This metric is preferred for high-dimensional sparse data as it measures vector orientation rather than magnitude:

$$similarity(A, B) = \frac{A \cdot B}{||A||\,||B||}[24]$$

where A and B are the rating vectors of two different anime. Predictions for a target user are generated by computing the weighted average of that user's ratings for the k nearest neighbors.

2.4.1.2. Content-Based Filtering Approach

In parallel to the collaborative model, we implemented a Content-Based Filtering approach that leverages item metadata rather than user interactions. We constructed a feature matrix F of dimensions $(items \times features)$, where each row vector represents an anime's attribute profile. The features were derived from multiple metadata columns and processed as follows: genres and producers were encoded using multi-label one-hot encoding, studios and type were one-hot encoded, episodes and duration_min were scaled to standardize their ranges, rating was ordinal-encoded to capture age constraints, and synopsis text was converted

into numerical vectors using TF-IDF. Using this feature matrix, we applied the k-Nearest Neighbors (k-NN) algorithm to identify the k most similar anime based on their content attributes. Cosine similarity was used as the distance metric, as it effectively measures similarity between high-dimensional vectors [23]. For a given anime, recommendations are generated by retrieving its k nearest neighbors in the feature space, representing the most similar titles based on genres, semantic content, format, and other metadata attributes.

To evaluate and compare the performance of both models, we employed Top-N ranking metrics, specifically Precision@k, Recall@k, Mean Average Precision@k (MAP@k), F1-Score@k, and R-Precision rather than error-based metrics like RMSE. Since the primary goal of a recommender system is to present a relevant list of items rather than to predict exact rating values, ranking metrics provide a more practical measure of utility.

**3. Results**

We evaluated the CF and CB k-NN models by using Precision@k, Recall@k, (MAP@k), F1-Score@k, and R-Precision. Unlike standard classification metrics such as global accuracy, which can be misleading in recommender systems due to class imbalance, these metrics specifically evaluate the quality of the top-ranked suggestions presented to the user [25].

Precision@k measures the proportion of relevant items found within the top-k recommendations [26].

Recall@k assesses the model's coverage, defined as the ratio of relevant items successfully recommended out of the total relevant items in the user's test set[26].

Mean Average Precision (MAP@k) evaluates the ranking quality, penalizing relevant items that appear lower in the recommendation list [27].

F1-Score@k provides a harmonic mean of Precision and Recall, offering a single metric to balance the trade-off between recommendation quality and coverage [28].

R-Precision is a metric that shows the precision at rank R, where R is the number of documents relevant to the query [29].

For all experiments, k-NN was configured with k=10, k=50, and k=100 neighbors for the similarity search. The evaluation was conducted on a stratified random sample of 1,000 users, with the system generating a Top-10, 50, and 100 recommendation list for each user.

*Table 4.1: Machine learning algorithm evaluation matrix results at k = 10*

| Algorithm | Precision [k=10] | Recall [k=10] | MAP [k=10] | F1-Score [k=10] | R-Precision [k=10] |
|---|---|---|---|---|---|
| Collaborative Filtering k-NN | 0.0556 | 0.0492 | 0.0199 | 0.0522 | 0.0509 |
| Content-Based Filtering k-NN | 0.1820 | 0.0370 | 0.1256 | 0.0614 | 0.0370 |

*Table 4.2: Machine learning algorithm evaluation matrix results at k = 50*

| Algorithm | Precision [k=50] | Recall [k=50] | MAP [k=50] | F1-Score [k=50] | R-Precision [k=50] |
|---|---|---|---|---|---|
| Collaborative Filtering k-NN | 0.0436 | 0.1718 | 0.0234 | 0.0696 | 0.0478 |
| Content-Based Filtering k-NN | 0.1072 | 0.1072 | 0.0449 | 0.1072 | 0.1072 |

*Table 4.3: Machine learning algorithm evaluation matrix results at k = 100*

| Algorithm | Precision [k=100] | Recall [k=100] | MAP [k=100] | F1-Score [k=100] | R-Precision [k=100] |
|---|---|---|---|---|---|
| Collaborative Filtering k-NN | 0.0364 | 0.2712 | 0.0271 | 0.0642 | 0.0492 |
| Content-Based Filtering k-NN | 0.0857 | 0.1752 | 0.0587 | 0.1145 | 0.1098 |

**4. Discussion**

This research employs the k-Nearest Neighbors (k-NN) algorithm to evaluate and compare the effectiveness of Collaborative Filtering and Content-Based approaches for anime recommendation. By holding the algorithmic architecture constant, the results gained offer valuable insights into the intrinsic quality and predictive power of the underlying data representations without the confounding influence of model complexity.

Precision@k is a fundamental metric that reveals the proportion of relevant recommendations found within the top-k list generated for each user [26]. As k increases from 10 to 100, precision for both models naturally declines. This follows the expected trend where increasing the recommendation window introduces more noise, making it harder to maintain a high "hit rate" per item recommended.

Recall@k signifies the model's capability to retrieve the full set of relevant items available in the test data [26]. Both models show substantial gains as k increases; CF k-NN improves from 0.0492 at k=10 to 0.2712 at k=100. Notably, while CB is more precise at lower k, CF begins to close the gap in Recall at higher k, suggesting that CF is better at discovering a broader variety of relevant items once the "narrow" top-tier constraints are relaxed.

MAP@k evaluates the quality of the ranking order, penalizing relevant items that appear lower down the recommendation list [27]. The MAP scores for CF remain consistently low (0.0199 to 0.0271), indicating that even when the model finds relevant items, they are often buried lower in the list. CBF maintains a superior MAP, though it also sees a sharp decline as k increases, likely because the specific metadata matches that drive high-rank hits are exhausted quickly.

F1@10 is a metric that balances the trade-off between precision and recall [26][28]. The Content-Based (CBF) approach consistently outperformed CF across nearly all metrics. At k=10, CBF achieved a precision of 0.1820, more than triple the 0.0556 achieved by CF. This indicates that anime metadata provides a much denser signal for immediate relevance than sparse user-rating matrices [25][7].

R-Precision, a metric that shows the precision at rank R, where R is the number of documents relevant to the query [29]. The CB model shows significantly higher stability here (0.1098 at k=100) compared to CF (0.0492), reinforcing the conclusion that content-based features are more resilient to the data sparsity that plagues collaborative methods in this dataset.

To address the discrepancy between the low metric scores and actual recommendation quality, we performed a qualitative case study. As shown in Tables 5.1 and 5.2, both models consistently retrieve direct sequels and genre-aligned peers, demonstrating that both models can generate semantically valid recommendations.

*Table 5.1: Qualitative assessment of CF k-NN*

| Query Anime | Rank | Recommended Show | Distance | Rationale |
|---|---|---|---|---|
| Naruto | 1 | Naruto: Shippuden | 0.2366 | Sequel of the query show |
| | 2 | Bleach | 0.3554 | One of the original “Big 3” alongside Naruto |
| | 3 | Death Note | 0.3965 | High audience overlap, shared status as “gateway anime” |
| Bocchi the Rock! | 1 | Chainsaw Man | 0.5582 | Aired in the same season as Bocchi the Rock!, hype cycle. |
| | 2 | "Oshi no Ko" | 0.5595 | Similar entertainment industry theme, viral hits one season apart |
| | 3 | Do It Yourself!! | 0.6181 | Aired in the same season as Bocchi the Rock!, hype cycle. |
| Natsume Yuujinchou | 1 | Zoku Natsume Yuujinchou | 0.2382 | Sequel of the query show |
| | 2 | Natsume Yuujinchou San | 0.2827 | Sequel of the query show |
| | 3 | Natsume Yuujinchou Shi | 0.3550 | Sequel of the query show |

| | | | | |
|---|---|---|---|---|
| Kimetsu no Yaiba | 1 | Kimetsu no Yaiba Movie: Mugen Ressha-hen | 0.3566 | Movie sequel of the query show |
| | 2 | Kimetsu no Yaiba: Yuukaku-hen | 0.3625 | Sequel of the query show |
| | 3 | Jujutsu Kaisen | 0.3710 | One of the modern “Big 3” alongside Kimetsu no Yaiba |
| K | 1 | K: Missing Kings | 0.5488 | Sequel of the query show |
| | 2 | K: Return of Kings | 0.5589 | Sequel of the query show |
| | 3 | Magi: The Labyrinth of Magic | 0.5624 | Aired in the same season, high audience overlap |
| Death Note | 1 | Code Geass: Hangyaku no Lelouch | 0.3243 | Overlap in the context of ‘strategic, morally complex narratives. |
| | 2 | Fullmetal Alchemist | 0.3592 | Co-consumption and co-rating patterns in ‘serious storytelling, ethical dilemmas, and long-form narrative depth. |
| | 3 | Code Geass: Hangyaku no Lelouch R2 | 0.3659 | User behavior alignment with Death Note. |
| One Piece | 1 | Bleach | 0.4785 | One of the original “Big 3” alongside One Piece |
| | 2 | Naruto | 0.4972 | One of the original “Big 3” alongside One Piece |
| | 3 | Naruto: Shippuuden | 0.5026 | One of the original “Big 3” alongside One Piece |
| Kimi no na wa | 1 | Koe no Katachi | 0.3561 | Shared emotional storytelling and themes of personal growth, though the focus on drama. |
| | 2 | Boku dake ga Inai Machi | 0.4761 | Shared appeal in narrative-driven plots, emotional depth, and character-centered tension. |
| | 3 | One Punch Man | 0.5083 | General user behavior patterns or interest in popular, well-rated anime, even though the genres differ. |

*Table 5.2: Qualitative assessment of CB k-NN*

| **Query Anime** | **Rank** | **Recommended Show** | **Distance** | **Rationale** |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Naruto | 1 | Bleach: Sennen Kessen-hen | 0.2081 | Shares action–adventure–fantasy genres, shounen structure, overlapping producers (TV Tokyo, Aniplex, Shueisha), and the same studio (Pierrot) |
| | 2 | Bleach: Sennen Kessen-hen – Ketsubetsu-tan | 0.2697 | Continuation with identical genres, Pierrot animation, and shared production partners, reinforcing strong stylistic similarity. |
| | 3 | Nanatsu no Taizai: Funnu no Shinpan | 0.2750 | Action–fantasy shounen with PG-13 tone, large-scale battles, and long-form adventure storytelling similar to *Naruto*. |
| Bocchi the Rock! | 1 | Akuei to Gacchinpo: Tenkomori | 0.3239 | Shares pure comedy focus, PG-13 rating, and Aniplex involvement, matching the lighthearted tone of *Bocchi the Rock!*. |
| | 2 | Kotowaza Gundam-san" | 0.3453 | Comedy-centric format with short, gag-driven humor similar to *Bocchi the Rock!*'s comedic style. |
| | 3 | Yule Zong Dong Yuan | 0.3453 | Matches the comedy genre and PG-13 tone, emphasizing humor over plot, aligning with *Bocchi the Rock!*'s style. |
| Natsume Yuujinchou | 1 | Vampire Knight | 0.4545 | Shares supernatural elements and emotional drama, focusing on relationships between humans and otherworldly beings. |
| | 2 | Tonari no Kaibutsu-kun | 0.4846 | Emphasizes character-driven drama and everyday interactions, similar to the slice-of-life aspects of *Natsume Yuujinchou*. |
| | 3 | Mushishi Zoku Shou | 0.5006 | Calm, episodic storytelling centered on supernatural encounters and quiet reflection, closely matching *Natsume Yuujinchou*'s tone. |
| Kimetsu no Yaiba | 1 | D.Gray-man Hallow | 0.3704 | Dark action–fantasy about humans fighting supernatural enemies, with intense battles and serious themes. |
| | 2 | Bleach: Sennen Kessen-hen | 0.3926 | High-stakes supernatural combat, strong shounen action, and mature violence similar to *Kimetsu no Yaiba*. |

| | | | | |
|---|---|---|---|---|
| | 3 | Rurouni Kenshin (2023) | 0.4027 | Focuses on sword-based combat, emotional struggles, and a serious tone despite a non-fantasy setting. |
| K | 1 | Coppelion | 0.3828 | Shares action-driven storytelling with mystery elements and a serious tone centered on dangerous missions and survival. |
| | 2 | K: Missing Kings | 0.3869 | Sequel of the query show |
| | 3 | K: Return of Kings | 0.3909 | Sequel of the query show |
| Death Note | 1 | Munou na Nana | 0.1974 | Strong supernatural suspense and R-17+ themes; shares tense, morally complex storytelling with Death Note. |
| | 2 | Rainbow: Nisha Rokubou no Shichinin | 0.3815 | Dark, mature suspense with overlapping producers (VAP, Nippon TV) and Madhouse studio, similar tone and tension |
| | 3 | Yuukoku no Moriarty Part 2 | 0.3884 | Suspense and crime-focused narrative with R-17+ rating; similar strategic and psychological elements despite different studios (Production I.G). |
| One Piece | 1 | Boruto: Naruto Next Generations Part 2 | 0.0871 | It shares action–adventure fantasy genres, PG-13 rating, and long-running shounen-style storytelling similar to One Piece. |
| | 2 | King's Raid: Ishi wo Tsugumono-tachi | 0.1496 | Action–adventure fantasy with PG-13 rating; similar epic quest and world-building themes, though different studios (OLM, Sunrise Beyond). |
| | 3 | Chain Chronicle: Haecceitas no Hikari | 0.1518 | Matches action–adventure fantasy genres and PG-13 rating; features large-scale adventure and ensemble casts, similar in tone to One Piece. |
| Kimi no na wa | 1 | Sarusuberi: Miss Hokusai | 0.1530 | Shares award-winning, drama, and supernatural genres with PG-13 rating; similar artistic and emotional storytelling despite different studios (Production I.G). |
| | 2 | Shisha no Sho | 0.1667 | Matches drama and supernatural themes with PG-13 rating; stylistically similar in tone and narrative focus on human and mystical elements. |

| | | | |
|---|---|---|---|
| 3 | Colorful (Movie) | 0.2077 | Combines drama, slice-of-life, and supernatural elements; PG-13 rated with emotional, reflective storytelling akin to Kimi no Na wa. |

## 5. Conclusion and Recommendation

This study evaluated the comparative effectiveness of Collaborative Filtering (CF) and Content-Based (CB) approaches for anime recommendation by employing a constant k-Nearest Neighbors (k-NN) architecture. By controlling for model complexity, the research isolated the intrinsic predictive value of user-interaction data versus metadata representations.

The empirical results indicate that the Content-Based model generally outperforms the Collaborative Filtering approach in terms of ranking quality and stability. The CB model maintained superior scores in MAP@k and R-Precision, demonstrating a resilience to data sparsity that significantly hampered the CF model. This suggests that in the context of anime recommendation, metadata (genres, studios, etc.) provides a more reliable foundation for generating immediate, high-relevance matches than user rating correlations alone.

However, Collaborative Filtering exhibited distinct utility in terms of Recall at higher k values. While CF struggled to rank relevant items at the top of the list, its improving performance as the recommendation window widened suggests it is capable of discovering a broader variety of relevant items once strict ranking constraints are relaxed.

Qualitatively, the case study confirmed that despite low raw metric scores, both models successfully generated semantically valid recommendations, such as direct sequels and genre peers. This discrepancy highlights a potential disconnect between rigid statistical metrics and perceived user utility. Ultimately, the trade-off observed, CB's superior precision versus CF's broader discovery potential. This shows that neither model is sufficient on their own. To achieve an optimal balance between relevance and coverage, future work should focus on Hybrid approaches that fuse the semantic density of content features with the serendipitous discovery mechanisms of collaborative filtering, and use more powerful algorithms such as Neural Networks.

## References


[1] Michal Daliot-Bul and Nissim Otmazgin. 2020. *The Anime Boom in the United States: Lessons for Global Creative Industries* (Vol. 406). Brill, Leiden, The Netherlands.

[2] Manuel Hernández-Pérez. 2019. Looking into the 'Anime Global Popular' and the 'Manga Media': Reflections on the Scholarship of a Transnational and Transmedia Industry. *Arts* 8, 2 (April 2019), 55.

[3] Association of Japanese Animations. 2024. *Anime Industry Report 2024*. Retrieved from https://aja.gr.jp/english/japan-anime-data

[4] Álvaro David Hernández Hernández. 2019. The Anime Industry, Networks of Participation, and Environments for the Management of Content in Japan. *Arts* 8, 3 (2019), 1–23. https://doi.org/10.3390/arts8030089

[5] Alexander Chernev, Ulf Böckenholt, and Joseph Goodman. 2015. Choice overload: A conceptual review and meta-analysis. *Journal of Consumer Psychology* 25, 2 (2015), 333–358. https://doi.org/10.1016/j.jcps.2014.08.002

[6] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. 2013. WTF: The Who to Follow service at Twitter. In *Proceedings of the 22nd International Conference on World Wide Web (WWW '13)*. 505–514. https://doi.org/10.1145/2488388.2488433

[7] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2022. Recommender Systems: Techniques, Applications, and Challenges. In *Recommender Systems Handbook* (3rd ed.), F. Ricci, L. Rokach, and B. Shapira (Eds.). Springer, New York, NY, USA, 1–35. https://doi.org/10.1007/978-1-0716-2197-4_1

[8] Bart P. Knijnenburg, Martijn C. Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction* 22, 4–5 (2012), 441–504. https://doi.org/10.1007/s11257-011-9118-4

[9] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal. 2012. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems* 26 (Feb. 2012), 225–238. https://doi.org/10.1016/j.knosys.2011.07.021

[10] A. Panteli and B. Boutsinas. 2023. Addressing the Cold-Start Problem in Recommender Systems Based on Frequent Patterns. *Algorithms* 16, 4 (March 2023), 182.

[11] A. Hussein, S. K. Bhatti, and F. Karray. 2013. A Personalized Content-Based Recommender System. *Journal of Universal Computer Science* 19, 15 (2013), 2224–2240.

[12] H. Sobhanam and A. K. Mariappan. 2013. A Hybrid Approach to Solve Cold Start Problem in Recommender Systems using Association Rules and Clustering Technique. *Int. J. Computer Applications* 74, 4 (July 2013), 17–23. https://doi.org/10.5120/12873-9697

[13] H. D. Putri and M. Faisal. 2023. Analyzing the Effectiveness of Collaborative Filtering and Content-Based Filtering Methods in Anime Recommendation Systems. *Komputasi dan Informatika (Komtika)* 7, 2 (2023). Retrieved from https://journal.unimma.ac.id/index.php/komtika/article/view/9219

[14] J. Murel and E. Kavlakoglu. 2025. What is collaborative filtering? *IBM Research*. Retrieved from https://www.ibm.com/think/topics/collaborative-filtering

[15] M. Hu, W. Xiao, and Y. Li, "Exploring the relationship between users' characteristics and movie recommendations using a KNN-based recommender system," 2025. [Online]. Available: https://ace.ewapub.com/article/view/10619

[16] B. S. Pratama, E. N. Furqon, and C. Natalia. 2025. Machine Learning for Anime Recommendation System. *International Journal in Engineering & Emerging Technology* (2025). Retrieved from https://ojs.uajy.ac.id/index.php/IJIEEM/article/view/9402

[17] M. R. Fauzi, I. Sanjaya, and I. L. Kharisma, "Genre-based anime recommendation system using KNN with fanbase bias detection," *BT Journal*, 2025. [Online]. Available: https://jurnal.kdi.or.id/index.php/bt/article/download/2917/1564/19657

[18] GeeksforGeeks. 2025. Content-Based vs Collaborative Filtering: Difference. *GeeksforGeeks*. Retrieved from https://www.geeksforgeeks.org/content-based-vs-collaborative-filtering-difference/

[19] V. Subburaj, Deep anime recommendation system: Recommending anime using hybrid filtering, M.Sc. research project, Data Analytics, 2024. [Online]. Available: https://norma.ncirl.ie/7762/1/varshinisubburaj.pdf

[20] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24, 5 (1988), 513–523.

[21] Fabian Pedregosa et al. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[22] Eda Kavlakoglu. n.d. What is the k-nearest neighbors (KNN) algorithm? *IBM*. Retrieved from https://www.ibm.com/think/topics/knn

[23] A. Wong and M. Raheem. 2020. An appraisal on the methods and techniques of recommender models for personalised marketing campaigns. *Journal of Physics: Conference Series* 1712, 1 (Dec. 2020), 012043. https://doi.org/10.1088/1742-6596/1712/1/012043

[24] GeeksforGeeks. 2025. Cosine Similarity. Retrieved July 15, 2025 from https://www.geeksforgeeks.org/dbms/cosine-similarity/

[25] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on 'top-n' recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*.

[26] Jaime Arguello. 2013. *Evaluation metrics*. INLS 509: Information Retrieval, lecture notes. Univ. of North Carolina at Chapel Hill. Retrieved from https://ils.unc.edu/courses/2013_spring/inls509_001/lectures/10-EvaluationMetrics.pdf

[27] K. Rink. 2023. Mean Average Precision at K (MAP@K) clearly explained. *Towards Data Science*. Retrieved January 18, 2023 from https://towardsdatascience.com/mean-average-precision-at-k-map-k-clearly-explained-538d8e032d2/

[28] Krishna Pullakandam. 2024. Understanding Precision, Recall, and F-Score at K in Recommender Systems. *Medium*. Retrieved June 20, 2024 from https://krishnapullak.medium.com/understanding-precision-recall-and-f-score-at-k-in-recommender-systems-7146a0dce68e

[29] Javed A. Aslam and Emine Yilmaz. 2005. A geometric interpretation and analysis of R-precision. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management* (CIKM '05). ACM, New York, NY, USA, 664–671.