UDC 004.8
DOI: https://doi.org/10.17721/AIT.2025.1.07

Maksym SHEVCHENKO, PhD Student
ORCID ID: 0009-0008-5104-9767
e-mail: maksym_shevchenko@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

AN ANALYTICAL REVIEW OF CONTENT-BASED AND COLLABORATIVE FILTERING IN RECOMMENDER SYSTEMS

Background. ***With the rapid growth of digital content, recommender systems are becoming a key tool for providing personalized offers. They contribute to the discovery of new movies, music, and products, maintaining user interest in using platforms. The relevance of researching recommender system algorithms is due to the need to improve their work to satisfy individual user preferences. This paper presents a review and analytical study of recommender system algorithms. The purpose of this paper is to systematize, classify, and critically analyze two main approaches in recommender systems: content-based filtering and collaborative filtering.***

Methods. ***A review of existing recommender system methods, a comparative and analytical assessment.***

Results. ***The work analyzes recommender system algorithms. A formal definition of the recommendation problem is given, where user preferences are modeled as a functional dependence on object properties. Within the framework of content-based filtering, the use of classification algorithms, such as a Naive Bayes classifier and decision trees, as well as the Rocchio algorithm, which uses relevant feedback to update the user profile, is considered. The strengths and weaknesses of different similarity measures between vectors are analyzed. In collaborative filtering, the memory-based approach (user-based and item-based methods) and model-based techniques with an emphasis on the k-NN algorithm are investigated. To overcome the shortcomings of individual methods, a hybrid approach is proposed that combines their advantages. Methods for integrating systems into a hybrid model are presented, which allows improving the accuracy of recommendations.***

Conclusions. ***The results of the work highlight the features of the specified filtering methods, demonstrate the impact of the implementation of algorithms and input data on the accuracy of recommendations and response time. The analysis of shortcomings emphasizes the importance of the combined use of filtering algorithms to improve the efficiency of recommendation systems, which makes the hybrid approach a promising direction for further research and implementation.***

Keywords: ***collaborative filtering, content-based filtering, hybrid filtering, recommender systems, Rocchio algorithm, vector space model.***

Background

The modern digital world is filled with an excess amount of data and information. Recommender systems have become important tools, helping users find what they're looking for among all the options. At their core, these systems aim to predict user preferences and offer personalized recommendations tailored to individual tastes.

Given the increasing reliance on recommender systems in various industries, optimizing their performance and improving their recommendation accuracy is a critical research area. The study of underlying principles, different similarity measures, and hybrid models contributes to enhancing the efficiency and adaptability of these systems. The selection of appropriate models and recommendation algorithms remains an open question in optimizing recommender system performance. As a result, numerous studies have been devoted to the development and evaluation of recommendation algorithms, which makes systematic review and analysis of existing approaches particularly valuable.

Two main methods used in recommender systems are content-based filtering and collaborative filtering. Content-based filtering uses the intrinsic properties of items and user profiles to generate recommendations, while collaborative filtering relies on the collective behavior of users to provide personalized suggestions. Each of these methods has its own advantages and drawbacks.

This paper is an analytical review of recommender system techniques, with a particular focus on content-based and collaborative filtering methods. The primary object of this research is recommender systems, focusing on content-based and collaborative filtering techniques. The aim of the research is to analyze and compare different algorithms used in recommender systems. The objectives include investigating the principles and algorithms behind content-based and collaborative filtering, and evaluating the effectiveness of different similarity measures, such as Euclidean distance, cosine similarity, Pearson correlation etc. The methodology includes an analytical review of existing recommender system techniques, their classification, systematization, and a comparison of their performance based on the related studies, highlighting the advantages and limitations of each approach.

The comparative analysis presented in this study is based on a set of qualitative and performance-oriented criteria. These include the type and characteristics of input data (explicit and implicit feedback, availability of item metadata, data sparsity, and vector dimensionality), recommendation accuracy reported in related studies, and computational aspects such as time performance and algorithmic complexity. Additional comparison dimensions involve scalability with respect to dataset size, interpretability of recommendation results, adaptability to evolving user preferences, and robustness to the cold-start problem for new users and items. Furthermore, the analysis considers the sensitivity of different methods to data quality and feature representation, highlighting how these factors influence both effectiveness and efficiency across content-based, collaborative, and hybrid recommendation approaches.

Methods

General model of recommender systems. A recommender system consists of essential components that facilitate personalized recommendations (Fig. 1). It is built around a set of source data files that can have different structures and be curated from various sources (Xia et al., 2024). The item catalog serves as the repository from which recommendations are drawn, ensuring a diverse selection for users.

User feedback is central to improving recommendation accuracy. Explicit feedback, derived from direct user input such as ratings, likes/dislikes, or comments, provides clear indications of user preferences. However, collecting explicit feedback can be challenging due to user participation barriers and potential biases (Kumar, 2022). Implicit feedback is inferred from user behavior, such as browsing history, clicks, and purchases, offering an alternative method that does not require direct user engagement. However, it may not fully capture user preferences as it relies on observable actions (Kumar, 2022). Combination of direct user input and observed behavior can provide a more comprehensive view of user interests and preferences (Mandal, & Maiti, 2018).

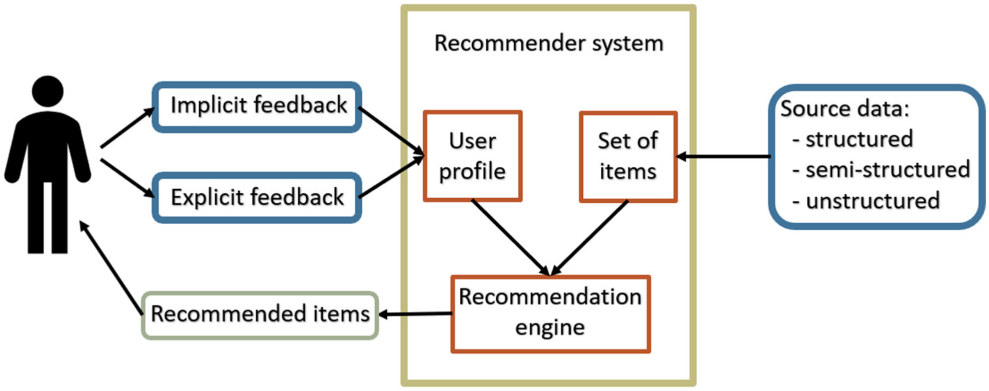


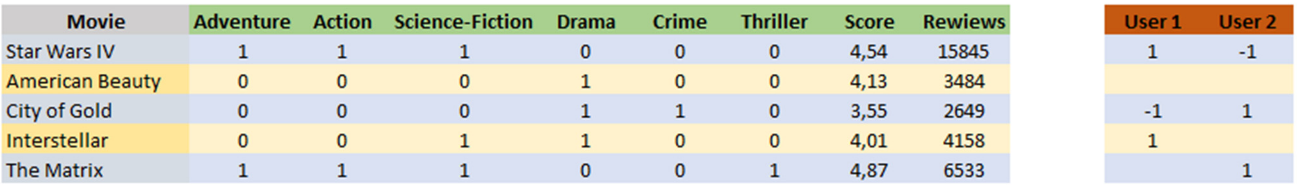
Fig. 1. General design of recommender systems

Leveraging the collected user feedback, the recommender system constructs individual user profiles that encapsulate their preferences. These profiles are dynamic representations that evolve over time based on the user's interactions. User profiles serve as the basis for generating personalized recommendations, allowing the system to match users with items that align closely with their preferences and interests.

In a formal context, recommender systems tackle the following problem (Zisopoulos et al., 2008): given a set of items S, a subset E (estimates) of S, with $|E|\ll|S|$ and the values f(e) for each $e\in E$ of a function $f(x):S\rightarrow\{0,1\}$, where the exact nature of f(x) is unknown, the aim is to find an estimate $g(x):S\rightarrow\{0,1\}$, and a subset R (recommendations) of S, with $|R|\ll|S|$ and $E\cap R=\emptyset$, such that the probability $P([g(r)=f(r)]=1)$, for every r in R is maximized. Both f(x) and g(x) can also target numerical sets, employing a threshold mechanism to discern between different classes. In other words, it is assumed that user preferences can be modeled by an unknown target function, and the goal of the recommender system is to approximate this function as accurately as possible.

Content-based filtering. Content-based filtering suggests items to users based on their features and the user's past interactions. The main idea is to find similarities between items and match them with user preferences.

This approach works by creating user profiles and item profiles (Fig. 2). A user profile stores information about a person's interests, preferences, and past interactions. An item profile contains details about an item, such as its description, metadata, or specific features. By comparing these profiles, content-based filtering algorithms find and recommend items that best match a user's interests (Herimanto, Samosir, & Ginting, 2024).



Movie	Adventure	Action	Science-Fiction	Drama	Crime	Thriller	Score	Rewiews
Star Wars IV	1	1	1	0	0	0	4,54	15845
American Beauty	0	0	0	1	0	0	4,13	3484
City of Gold	0	0	0	1	1	0	3,55	2649
Interstellar	0	0	1	1	0	0	4,01	4158
The Matrix	1	1	1	0	0	1	4,87	6533

User 1	User 2
1	-1
-1	1
1	
	1

Fig. 2. An example of item and user profiles

Before feeding the data into recommendation algorithms, preprocessing steps are typically applied to clean, normalize, and transform the data. This may involve handling missing values, encoding categorical variables, and scaling numerical features. Additionally, feature extraction techniques may be employed to derive meaningful representations of items and users, such as content features, word embeddings or contextual information, in high-dimensional vectors.

The primary objective of data preprocessing is to transform raw item features into a structured, numerical format suitable for input into mathematical models and machine learning algorithms. These numerical representations form the basis for measuring similarity, learning user preferences, and generating accurate, personalized recommendations.

Since recommender systems aim to estimate the likelihood of user satisfaction with specific items (often categorizing this satisfaction into discrete outcomes such as "like", "dislike", or "neutral"), the recommendation task can naturally be framed as a classification problem. In this formulation, the system predicts the class label that best represents a user's potential reaction to an item, based on historical interactions and item attributes. By training on labeled data, where the labels represent user preferences, classification algorithms build predictive models that can then be used to classify new items for recommendation.

Different classification algorithms can be utilized in content-based filtering. Naive Bayes is a powerful algorithm for content-based filtering in recommender systems due to its simplicity, speed, and effectiveness in handling text data (Ricci, 2002). It operates on the principle of Bayes' theorem, assuming that the presence of a particular feature in a class is unrelated to the presence of any other feature. This "naivety" simplifies the calculation of probabilities, allowing effectively model the content of items and recommend similar items based on the features. However, it's important to note that while Naive Bayes is fast and easy to implement, it makes strong assumptions about feature independence, which might not hold true in all datasets. Therefore, it's crucial to evaluate its performance on your specific dataset and consider other algorithms if necessary.

Decision trees are powerful and interpretable models used in machine learning to solve classification problems by hierarchically dividing the feature space. In content-based filtering, they provide a structured approach for predicting user preferences based on item attributes (Gershman et al., 2010). A decision tree consists of internal nodes that evaluate feature conditions and branches that represent outcomes, ultimately leading to leaf nodes associated with predicted classes (Fig. 3). During training, the tree splits data recursively, aiming to reduce impurity or maximize information gain, resulting in a set of decision rules that are easy to follow and explain.

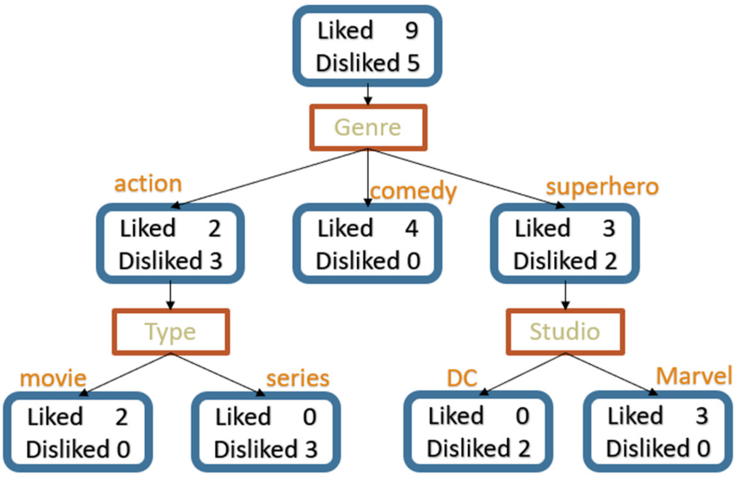


Fig. 3. An example of a decision tree

In recommendation systems, decision trees apply these rules to evaluate item relevance for each user, enabling personalized suggestions. Their visual structure enhances transparency, making them accessible to stakeholders without deep technical expertise. Decision trees support both categorical and numerical features and offer efficient performance suitable for large-scale systems.

However, the effectiveness of decision trees relies heavily on the quality of the input features. Proper feature engineering and preprocessing are critical to accurately model user preferences. Additionally, decision trees can overfit the training data if allowed to grow too deep, capturing noise rather than general trends. To mitigate overfitting, it's crucial to tune the hyperparameters of the decision tree, such as the maximum depth and the minimum number of samples required to split a node, ensuring better generalization to unseen data.

Classic classification algorithms often face challenges in adapting to new data. Traditional models are typically static and require retraining to incorporate additional user interactions. This can be inefficient and difficult to scale, especially with large datasets. Although some modifications to standard algorithms support incremental learning, allowing the model to update with new data without full retraining, implementing such methods can be complex and resource-intensive in practice. As user preferences evolve over time, maintaining model relevance becomes critical. Relying on outdated models risks degrading recommendation quality, while frequent retraining demands substantial computational resources.

To address this challenge, the technique of relevance feedback has emerged as a promising solution. Unlike traditional approaches, relevance feedback empowers systems to dynamically adjust existing models based on user feedback. At the forefront of relevance feedback algorithms stands the Rocchio algorithm, renowned for its effectiveness in refining recommendations in response to user interactions (Meteren, & Someren, 2000). The Rocchio algorithm operates within the vector-space model, necessitating the transformation of unstructured data into a structured format. In this model, each item is represented as a vector in an n-dimensional space (Fig. 4), where n corresponds to the number of characteristics or features.

Central to the Rocchio algorithm is the concept of user profiling, where each user is characterized by a vector that reflects their preferences. This vector is engineered to exhibit high correlation with positively rated items while displaying lower correlation with negatively rated ones. The essence of the Rocchio algorithm can be succinctly captured by the following formula:

$$P' = a \cdot P + b \cdot ST(rel) - c \cdot ST(non - rel). \quad (1)$$

Here, P' represents the updated user profile, P denotes the original user profile vector, $ST(rel)$ signifies the vector sum of positively rated items, and $ST(non - rel)$ represents the vector sum of negatively rated items. Hence, when incorporating a newly rated item into the user's profile, a simple operation suffices. If the item T is deemed relevant, the updated user profile P' is calculated as $P' = a \cdot P + b \cdot T$. Conversely, if the item T is assessed as non-relevant, the updated user profile is determined as $P' = a \cdot P - c \cdot T$. This streamlined process enables the continual refinement of the user's profile, ensuring it remains highly correlated with relevant yet unrated items.

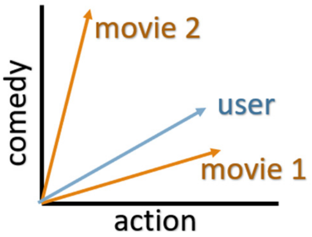


Fig. 4. An example of the user and items profile representation in a vector space model

The coefficients a, b and c modulate the influence of each component on the updated user profile, allowing for fine-grained adjustments based on user feedback. Parameter a manages the balance between past preferences and new interests, ensuring the profile retains useful historical data while adapting to changes. Parameter b determines how quickly the profile updates when encountering relevant items, helping it learn new preferences efficiently. Conversely, parameter c regulates the impact of non-relevant items, preventing incorrect recommendations from distorting the user profile. By carefully tuning these values, recommender systems can better adapt to individual user behavior, improving recommendation accuracy and overall user satisfaction. Continuous updates based on feedback allow the profile to evolve dynamically, making recommendations more responsive and relevant over time.

In content-based recommender systems, various similarity measures, including Euclidean distance, Pearson correlation, and Jaccard similarity, as well as other methods, are used to evaluate the similarity between user profiles and items based on their features (Deutschman, 2023). Each of these methods serves different purposes and is suitable for different types of data and requirements.

Euclidean distance measures the straight-line distance between two points in a multi-dimensional space (2), which can be visualized as the length of the line segment connecting them. It is particularly useful in cases where the magnitude of the vectors is meaningful and should be taken into account. For instance, when the features of the vectors represent absolute values, such as the number of times a user has interacted with certain items or the ratings a user has given, Euclidean distance can effectively capture the overall difference between two users or items:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}. \tag{2}$$

Cosine similarity quantifies the similarity between two vectors by measuring the cosine of the angle between them in a multi-dimensional space (3). It is particularly effective in scenarios where the magnitude of the vectors is less important than their direction. This characteristic makes cosine similarity ideal for identifying items that are similarly oriented in a high-dimensional space, regardless of their length. For instance, when comparing articles of varying lengths, cosine similarity can effectively find articles that are conceptually similar, even if one article is much longer than another:

$$S_c(x,y) = \cos(\theta) = \frac{x \cdot y}{||x|| \cdot ||y||} = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i{}^2} \cdot \sqrt{\sum_{i=1}^{n} y_i{}^2}}. \tag{3}$$

Pearson correlation measures the linear correlation between two variables (4). It is commonly used in recommender systems to assess the strength and direction of the relationship between two variables. Pearson correlation indicates the degree of similarity between users' preferences. A high positive correlation suggests similar preferences, while a negative correlation suggests divergent preferences. However, Pearson correlation can be problematic with sparse data, as it may not accurately reflect the similarity between items with few common features. In such cases, cosine similarity might be a better choice (Gaurav, 2023):

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}. \tag{4}$$

Jaccard similarity measures the similarity between two sets by dividing the size of the intersection by the size of the union of the two sets (5). It is particularly useful in recommender systems for binary data, such as tag-based recommendations. For instance, in a movie recommender system, Jaccard similarity can be used to determine how similar two movies are based on their top tags:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}. \tag{5}$$

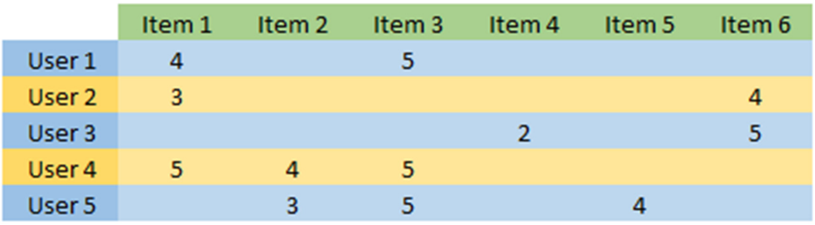
Semantic similarity and relatedness between words is an important aspect of natural language processing tasks, which can enhance the performance of various applications within recommender systems. By going beyond surface-level comparisons, semantic similarity measures assess the meaning and contextual relationships between words. These methods enable the construction of algorithmic models that enhance context-linguistic analysis, addressing challenges such as word sense disambiguation, named entity recognition, and text analysis. A particularly effective approach involves estimating the semantic distance between words through a weighted modification of the Lesk algorithm (Anisimov, Marchenko, & Kysenko, 2011). By incorporating these refined semantic similarity measures, recommender systems can better understand and utilize the nuanced relationships between words, leading to more precise and contextually relevant recommendations.

These similarity measures have been compared on a real learning platform dataset (Joy, & Renumol, 2020). The study evaluated Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics. It was found that Jaccard similarity and Euclidean distance perform similarly well (MAE 0.66, RMSE 0.91) and better than Pearson correlation (MAE 0.72, RMSE 0.94), while cosine similarity outperforms them all (MAE 0.6, RMSE 0.87). Another study (Wijewickrema, Petras, & Dias, 2019) compared normalized discounted cumulative gain (NDCG). This study showed that BM25 measure can outperform cosine similarity in a journal recommender system – BM25 achieved an average NDCG score of 0.626 for the social science domain and 0.615 for the medicine domain, compared to 0.436 and 0.469 for cosine similarity respectively.

A comparative study has been performed to evaluate the time performance of various distance measures using different sizes of datasets (Marchenko, & Shevchenko, 2024). The study revealed that cosine similarity, inner product, and Euclidean distance produced comparable results in content-based filtering, showing no significant differences in retrieval performance. Also, since feature vectors in content-based filtering have a predefined shape that does not depend on dataset size, the response time for all three measures exhibited a linear growth pattern as the dataset size increased.

These studies are showing that there is no one-size-fits-all answer to which similarity measure performs best in content-based filtering. The optimal choice depends on the characteristics of the data and the specific needs of the recommender system. Experimentation and evaluation using real-world data are crucial to determining the most effective similarity measure for a given application.

Collaborative filtering. Rather than relying on item attributes or content descriptors, collaborative filtering algorithms analyze user-item interactions, such as ratings, purchases, or clicks, represented as a utility matrix (Fig. 5). By identifying users with similar preferences and recommending items favored by their peers, collaborative filtering endeavors to deliver tailored recommendations that resonate with individual tastes. This approach is widely used in various domains, including product recommendations on e-commerce sites, movie and TV show recommendations on streaming platforms and article recommendations on news sites.



	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	4		5			
User 2	3					4
User 3				2		5
User 4	5	4	5			
User 5		3	5		4	

Fig. 5. An example of a utility matrix

Model-based approaches represent a paradigm in collaborative filtering, where intricate models are crafted to capture underlying patterns and relationships inherent in user-item interactions, and are encompassing various techniques (Grover, 2017). Matrix factorization techniques, such as Singular Value Decomposition (Nguyen, 2016), Principal Component Analysis (Yadav et al., 2021), Nonnegative Matrix Factorization (Anisimov, Marchenko, & Vozniuk, 2014) and Alternating Least Squares (Gosh et al., 2021), as well as the tensor approach (Marchenko, 2016), decompose the user-item interaction matrix into lower-dimensional representations. These latent factors capture user preferences and item characteristics, enabling accurate prediction of user-item interactions. Probabilistic graphical models, such as Bayesian networks (Qazi et al., 2017) or Markov random fields (Steck, 2019), represent the dependencies between users and items probabilistically. Deep learning architectures, such as neural networks or autoencoders (Liu et al., 2018), learn hierarchical representations of user-item interactions. By leveraging multiple layers of abstraction, deep learning models capture intricate patterns and nuances in the interaction data, enabling the generation of highly accurate recommendations.

Unlike model-based approaches that rely on learned parameters, memory-based methods directly compute similarities between users or items to infer preferences and foster recommendation generation. A widely used technique within memory-based collaborative filtering is the k-Nearest Neighbors (k-NN) algorithm (Analytics Vidhya, 2024). The k-NN identifies similar users or items by computing distances in the feature space, based on the assumption that the target user shares preferences with their nearest neighbors. Memory-based collaborative filtering encompasses user-based and item-based approaches (Belhaouari et al., 2023).

User-based approach identifies k users who are similar to the target user based on their past behavior or ratings. The similarity between users is typically measured by the similarity of their ratings or interactions with items. These similar users serve as a reference group whose preferences are aggregated to generate recommendations for the target user. It recommends items that these similar users have liked but the target user has not yet interacted with (Fig. 6). This facilitates the discovery of relevant items that align with the preferences of the target user (Vijay, 2020).

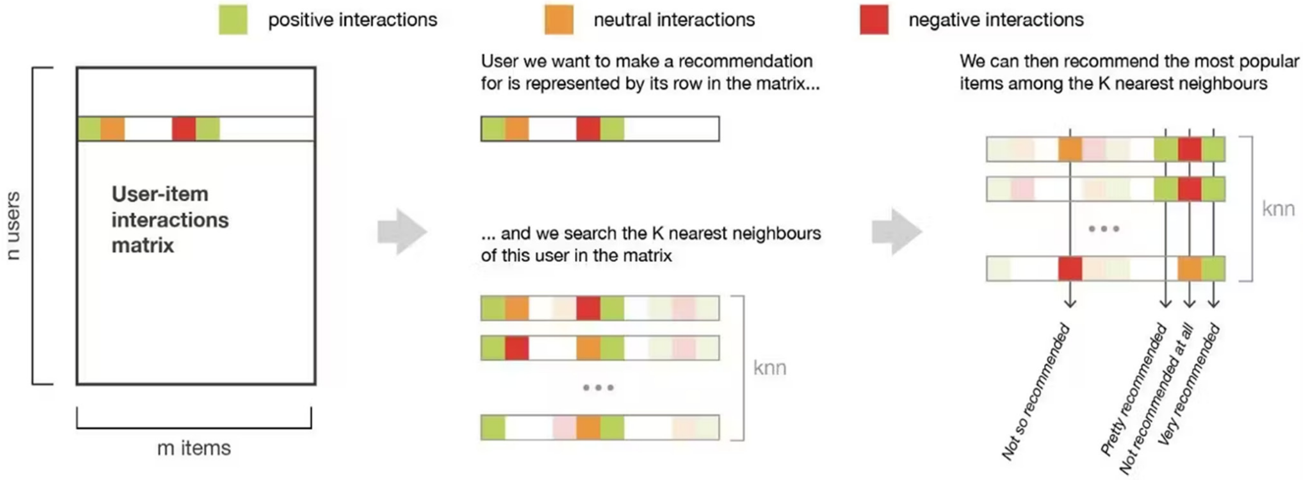

🟩 positive interactions
🟧 neutral interactions
🟥 negative interactions
n users
User-item interactions matrix
m items
User we want to make a recommendation for is represented by its row in the matrix...
... and we search the K nearest neighbours of this user in the matrix
knn
We can then recommend the most popular items among the K nearest neighbours
knn
Not so recommended
Pretty recommended
Not recommended at all
Very recommended

Fig. 6. An illustration of the user-based collaborative filtering algorithm (Vijay, 2020)

Item-based method focuses on the items themselves rather than the users. K-NN operates by identifying items that are similar to those a user has interacted with or rated highly (Fig. 7). The similarity between items is determined by the similarity of the ratings of those items by the users who have rated both items. The algorithm finds the k items that are most similar to the items the user has interacted with and recommends these items to the user.

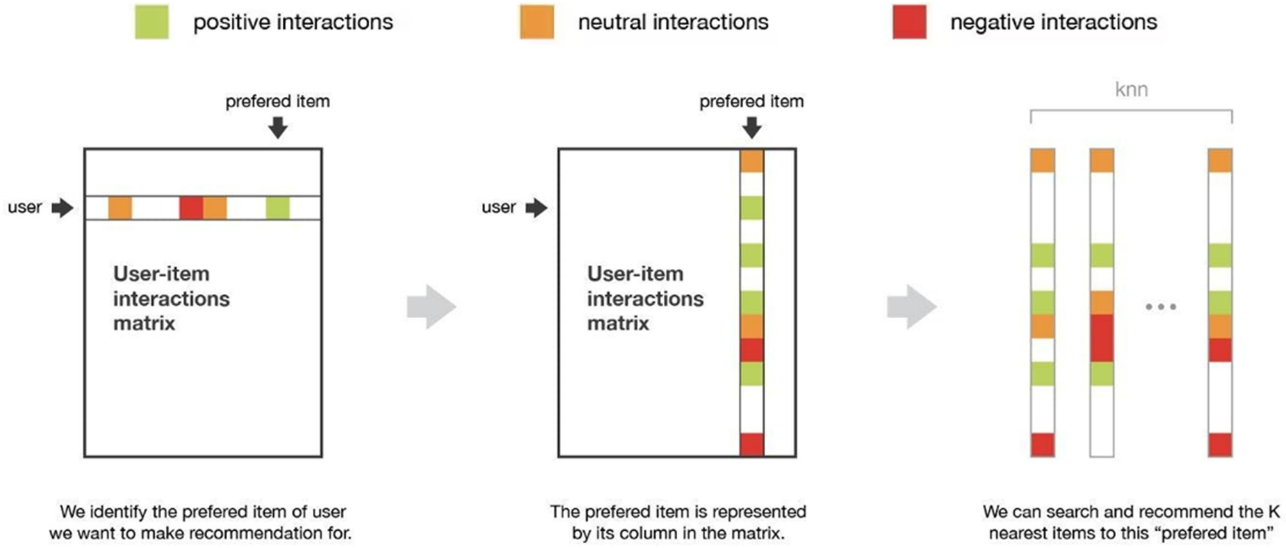

positive interactions
neutral interactions
negative interactions
prefered item
user
User-item interactions matrix
prefered item
user
User-item interactions matrix
knn
We identify the prefered item of user we want to make recommendation for.
The prefered item is represented by its column in the matrix.
We can search and recommend the K nearest items to this "prefered item"

Fig. 7. An illustration of the item-based collaborative filtering algorithm (Vijay, 2020)

Collaborative filtering and content-based filtering may differ in their underlying principles, but they often share common ground when it comes to similarity measures. The same similarity measures utilized in content-based filtering can also be seamlessly integrated into collaborative filtering frameworks, enriching the recommendation process with insights derived from user-item interactions (Kumar, 2022).

The experimental study (Fkih, 2022) demonstrates that no single similarity measure is universally effective for both user-based and item-based approaches, with some measures performing better in one approach than the other. Additionally, the study highlights that the effectiveness of a similarity measure can vary based on dataset density, even within the same filtering approach. The study (Saranya, Sadasivam, & Chandralekha, 2016) proposed a new similarity measure that incorporates Pearson Correlation Coefficient and Jaccard Coefficient. The proposed combined similarity measure outperformed the Bhattacharyya Coefficient and the Jaccard Uniform Operator Distance, and achieved an average F-score measure of 0.77 compared to 0.72 and 0.75 in the competitors respectively. Another study (Sun et al., 2017) proposed integrating Triangle and Jaccard similarities for recommendation to enhance recommender system performance. This combination allows the system to leverage the strengths of different measures, addressing their individual limitations and providing a more robust and adaptable recommendation process. The proposed measure achieved MAE of 0.614 and RSME of 0.816 for the FilmTrust dataset, compared to MAE of 0.617–0.852 and RSME of 0.817–1.043 for other similarity measures.

Another research investigated the impact of similarity measures on recommendation time (Baxla, 2014). The findings revealed that correlation-based measures require significantly more time for execution. One more comparative study (Marchenko, & Shevchenko, 2024) demonstrated that cosine similarity has significantly worse time performance on high-dimensional datasets compared to inner product and Euclidean distance. Additionally, the experiment revealed that due to data sparsity, the response time on a high-dimensional collaborative filtering dataset can be lower than on a lower-dimensional content-based filtering dataset. Therefore, when selecting an appropriate similarity measure, it is crucial to consider not only the accuracy but also the computational efficiency.

Results

Pros and cons. Comparative analysis. Content-based filtering relies on structured representations of item attributes and user profiles derived from explicit or implicit individual feedback. Because recommendations are generated by matching item features to learned user preferences, the underlying reasoning is typically transparent and easy to interpret, especially in rule-based models, decision trees, or relevance feedback approaches such as the Rocchio algorithm (Ricci, 2002). From a data perspective, these methods are particularly well suited to environments with rich and expressive metadata, where item characteristics can be encoded as fixed-dimensional vectors obtained from metadata, textual descriptions, or semantic representations. Under such conditions, comparative studies report stable and reliable recommendation accuracy, provided that similarity measures are appropriately selected and features adequately capture user interests (Joy, & Renumol, 2020).

From a computational standpoint, most content-based algorithms are efficient and scalable. Similarity calculations between user profiles and items typically exhibit linear time complexity with respect to the number of items, while fixed-dimensional feature spaces enable predictable performance even in large and frequently updated catalogs (Marchenko, & Shevchenko, 2024). Another notable advantage is robustness to the item cold-start problem: new items can be recommended immediately based solely on their attributes, without requiring historical user-item interaction data or information about other users' preferences (Herimanto, Samosir, & Ginting, 2024). This self-sufficiency also allows content-based systems to effectively serve users with niche or specialized interests, including items that may be unpopular in the broader population.

Despite these strengths, content-based filtering suffers from several limitations. Its effectiveness is highly sensitive to feature engineering quality: incomplete, noisy, or poorly designed item representations can substantially degrade recommendation performance and narrow the recommendation space (Zisopoulos et al., 2008). Typical implementation pitfalls include inadequate preprocessing, inappropriate similarity measure selection for the data type, and lack of proper feature normalization, all of which may bias similarity computations. Moreover, content-based filtering systems are prone to over-specialization, repeatedly recommending items that are very similar to those previously consumed and thereby limiting diversity and serendipity (Ricci, 2002).

Finally, content-based filtering may exhibit unstable or unsatisfactory behavior in dynamic or subjective domains. Because recommendations are constrained to known item characteristics, these methods struggle to introduce entirely novel or conceptually different items and often fail to capture rapidly evolving user interests without frequent retraining. As a result, delayed adaptation and increased computational overhead can occur when user preferences change quickly. Such weaknesses are especially evident in scenarios with sparse or poorly defined metadata, context-dependent or subjective tastes, and highly dynamic user behavior, where static content representations are insufficient to model the true drivers of user satisfaction (Herimanto, Samosir, & Ginting, 2024).

Unlike content-based approaches, collaborative filtering does not require explicit item features or rich metadata, allowing it to recommend items even when descriptive attributes are incomplete or unavailable (Vijay, 2020). By exploiting patterns in collective user behavior, collaborative filtering can uncover latent preferences and recommend items that users might not have discovered on their own, thereby introducing serendipity and promoting exposure to diverse content. When sufficient interaction data are available, both memory-based and model-based methods demonstrate high recommendation accuracy by capturing shared interests and implicit behavioral signals (Fkih, 2022).

From an accuracy and adaptability perspective, collaborative filtering is particularly effective in domains where subjective preferences dominate and are difficult to encode through explicit features. Model-based approaches, such as matrix factorization and probabilistic models, are especially successful at learning latent structures that generalize beyond observed interactions, often outperforming simpler methods in dense data settings (Nguyen, 2016; Gosh et al., 2021). However, this advantage diminishes in sparse environments, where unreliable similarity estimates and limited historical data reduce recommendation quality (Belhaouari et al., 2023). Adaptability to evolving user interests varies by method: memory-based approaches naturally incorporate recent interactions, whereas model-based techniques require periodic retraining or incremental updates to reflect preference shifts.

Computational complexity and scalability represent central challenges for collaborative filtering. Memory-based methods, including user-based and item-based k-NN, incur substantial computational costs due to pairwise similarity calculations over high-dimensional and sparse matrices, which can lead to unstable response times without proper optimization (Marchenko, & Shevchenko, 2024). Model-based methods mitigate online computation by shifting complexity to an offline training phase, enabling faster recommendation at runtime but at the cost of increased training overhead and resource consumption. As a result, large-scale collaborative filtering-based systems often rely on dimensionality reduction, approximation techniques, or distributed computing frameworks to achieve acceptable scalability (Nguyen, 2016).

In terms of interpretability, collaborative filtering generally offers limited transparency compared to content-based methods. Recommendations are derived from aggregated user behavior or latent factors rather than explicit item characteristics, making it difficult to provide clear explanations to end users. While this opacity may be acceptable in some applications, it can reduce user trust and hinder system debugging or evaluation, particularly in high-stakes domains (Fkih, 2022).

Despite its strengths, collaborative filtering is subject to several fundamental limitations and common pitfalls. The cold-start problem remains a major challenge, as new users and new items lack sufficient interaction data to generate reliable recommendations (Belhaouari et al., 2023). Data sparsity further exacerbates this issue, leading to noisy or biased similarity estimates and degraded accuracy, especially in user-based approaches. Popularity bias is another frequent concern, whereby frequently interacted items are over-recommended, limiting diversity and marginalizing niche content (Vijay, 2020). Moreover, inappropriate choices of similarity measures, neighborhood size, or update strategies can significantly harm both accuracy and efficiency (Baxla, 2014).

Hybrid approach. Hybrid recommender systems combine the strengths of both content-based and collaborative filtering approaches to enhance the effectiveness of recommendations (Wayesa et al., 2023). This approach is particularly useful when there is insufficient data to effectively apply either method alone. By leveraging both types of data, hybrid systems can offer more personalized and accurate recommendations.

Hybrid recommender systems have been extensively studied as a means to improve recommendation quality by combining multiple filtering techniques. The foundational work (Burke, 2007) introduced a systematic framework for hybridization, focusing on integrating collaborative filtering and content-based methods. Based on the interaction patterns between constituent models, hybrid recommender systems can be classified into several major types listed below, each addressing specific challenges of recommendation tasks to a different extent.

The weighted hybrid approach (Fig. 8) combines the outputs of multiple recommendation models using predefined or learned weights. Typically, predictions generated by content-based and collaborative filtering models are linearly aggregated to produce the final recommendation score. This approach effectively improves recommendation accuracy by balancing complementary information sources and partially alleviates data sparsity issues. However, weighted hybrids do not fully resolve the cold-start problem for new users, as collaborative components still require sufficient interaction history. Additionally, static weighting schemes may limit adaptability to dynamic user behavior and may lead to suboptimal performance if weights are not carefully tuned.

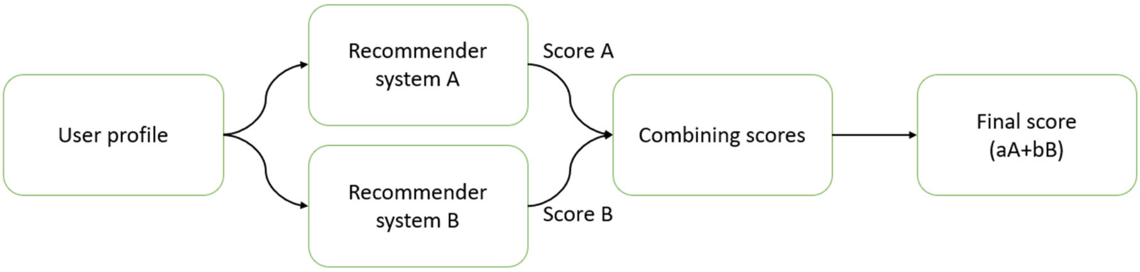


Fig. 8. Weighted hybrid recommender system

In switching hybrid systems (Fig. 9), the recommendation process dynamically selects one of several models based on contextual conditions, such as user profile completeness, interaction history, or item availability. This approach explicitly addresses the cold-start problem by activating content-based methods for new users or items and switching to collaborative filtering when sufficient data becomes available. Switching hybrids offer improved robustness across heterogeneous usage scenarios but do not directly combine model outputs, which may limit accuracy gains compared to weighted or mixed hybrids. Furthermore, incorrect switching criteria can lead to unstable or inconsistent recommendations.

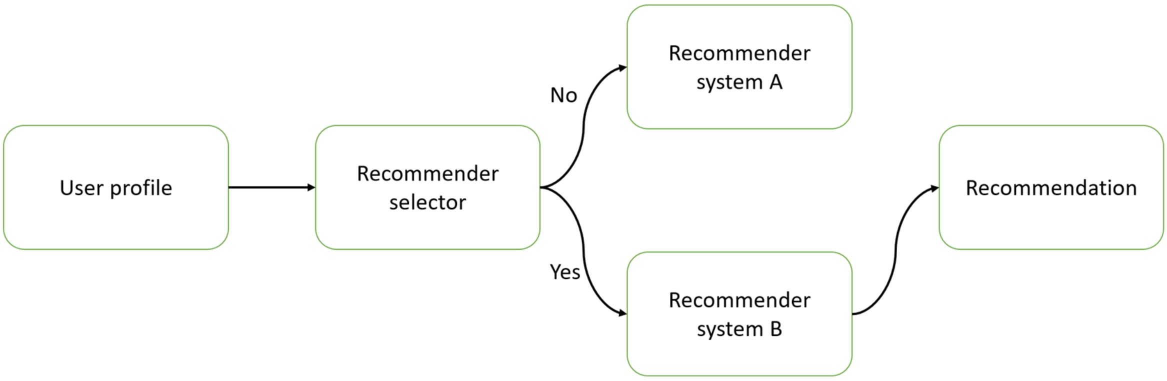


Fig. 9. Switching hybrid recommender system

The mixed hybrid (Fig. 10) approach generates recommendations from multiple models simultaneously and merges the resulting candidate lists into a unified output. This strategy enables the system to exploit diverse recommendation signals in parallel and is particularly effective for increasing recommendation coverage and diversity. Mixed hybrids can handle sparse and partial datasets more effectively by matching subsets of users or items to appropriate models. However, this approach increases computational complexity and may introduce redundancy or conflicting recommendations if result aggregation is not properly managed.

Cascaded hybrid systems (Fig. 11) employ a hierarchical structure in which a primary recommendation model generates an initial ranked list, while secondary models refine the output by resolving ambiguities, such as tie-breaking or missing values. This approach is well suited for improving precision and ranking quality without significantly increasing computational overhead. Cascading hybrids, however, depend heavily on the quality of the primary model and do not inherently address cold-start or data sparsity problems unless explicitly incorporated into the cascade structure.

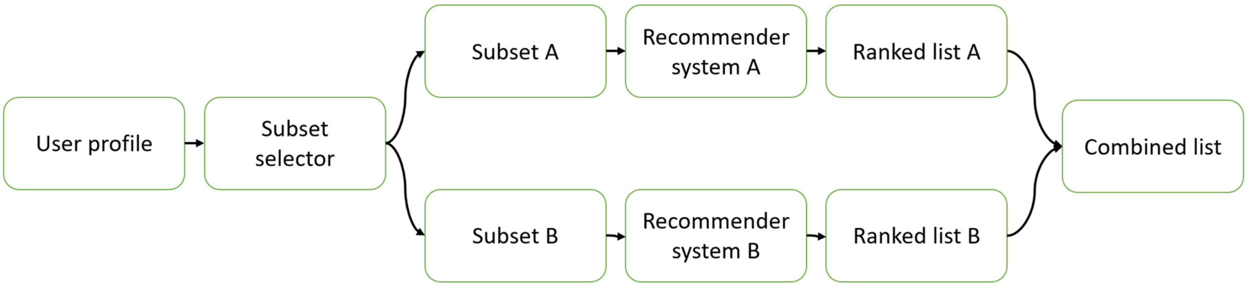


Fig. 10. Mixed hybrid recommender system

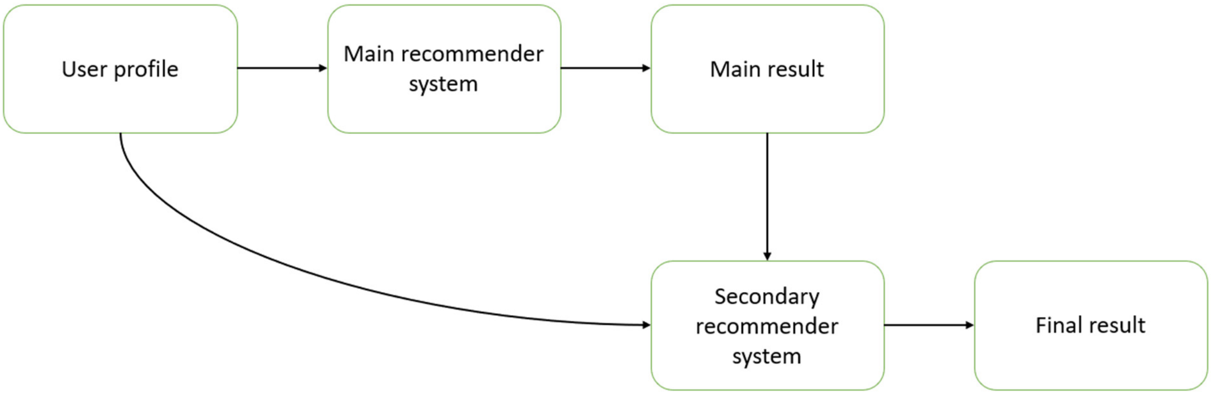


Fig. 11. Cascaded hybrid recommender system

Feature augmentation (Fig. 12) integrates a contributing recommendation model by using its output, such as predicted ratings or class labels, as input features for the main recommendation system. This strategy enhances the representational power of the core model and improves recommendation accuracy without altering its fundamental structure. Feature augmentation is effective for incorporating collaborative signals into content-based models and partially mitigating data sparsity. Nevertheless, it does not fully eliminate cold-start limitations and may propagate errors from the auxiliary model into the main system.

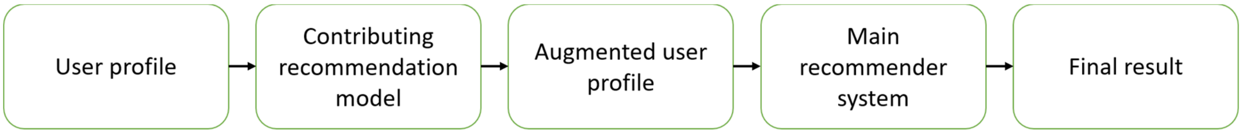


Fig. 12. Feature augmentation hybrid recommender system

The feature combination approach (Fig. 13) merges features derived from different recommendation paradigms into a unified feature space. For example, collaborative latent factors may be injected into content-based user profiles, enabling the model to consider both item attributes and user interaction patterns. This method provides strong flexibility and improves personalization by jointly modeling heterogeneous data. However, feature combination increases model complexity and may suffer from scalability issues in high-dimensional feature spaces. Additionally, interpretability can be reduced due to the integration of latent and explicit features.

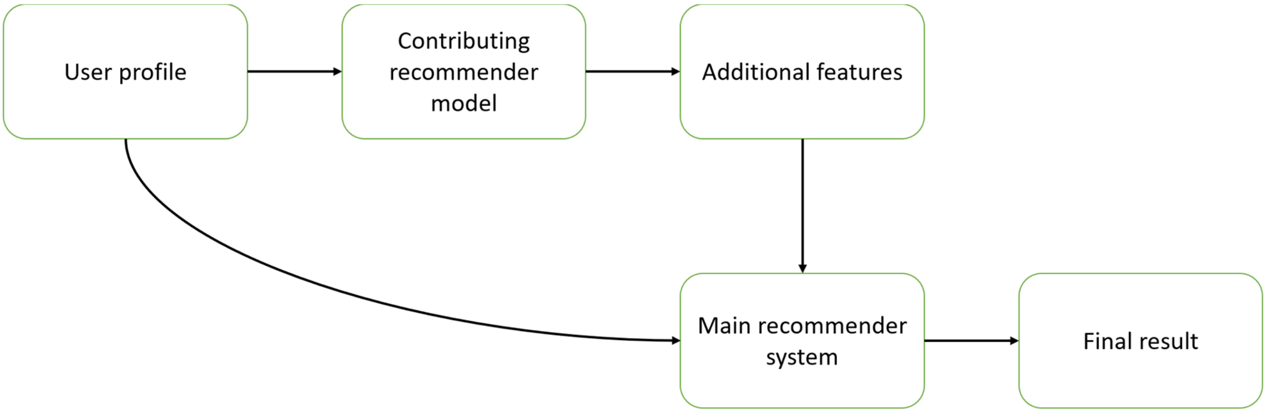


Fig. 13. Feature combination hybrid recommender system

The study (Burke, 2007) highlighted the trade-offs of hybrid systems, including increased computational complexity and the necessity for mindful parameter tuning. Recent systematic reviews on hybrid recommender systems highlight their growing significance in addressing key challenges such as cold-start problems, data sparsity, and recommendation accuracy. The study (Çano, 2017) found that most hybrid models combine collaborative filtering with other strategies, often using weighted hybridization due to its simplicity and flexibility. The review emphasized that cold-start and data sparsity remain the most frequently addressed issues. Moreover, accuracy metrics dominate the assessment of hybrid recommenders, but user satisfaction evaluations remain a challenge. Another study (Sabiri et al, 2025) provided a broader perspective by exploring hybridization techniques within big data environments. This review underscored the increasing adoption of machine learning-based hybridization methods and the necessity for scalability improvements. The review also highlighted the importance of balancing accuracy with computational efficiency, suggesting the use of hybrid parallel algorithms and large-scale datasets to address scalability concerns. Ethical considerations and user engagement metrics were also noted as emerging research priorities.

These studies emphasize that while hybrid recommender systems provide substantial benefits, their successful implementation requires significant computational resources and careful design considerations. In many cases it also requires distributed or parallel processing to maintain scalability and acceptable response times (Çano, 2017). Interpretability varies widely across hybridization strategies: weighted and cascaded models tend to preserve a moderate level of transparency, whereas deeply integrated or machine learning-based hybrids are less interpretable due to latent representations and multi-stage transformations (Liu et al., 2018).

From a performance and robustness perspective, hybrid recommender systems demonstrate higher accuracy and greater adaptability to evolving user preferences. The content-based component enables recommendations for new users or items with limited interaction history, while the collaborative component contributes serendipity and discovery of novel items beyond a user's immediate profile. This balance makes hybrid systems especially suitable for large-scale, real-world applications characterized by heterogeneous data and dynamic user behavior (Sabiri et al., 2025).

Despite their overall robustness, hybrid recommender systems are not immune to instability or suboptimal performance. Poorly chosen integration strategies, imbalanced weighting between components, or misalignment between algorithmic assumptions and data characteristics can lead to degraded accuracy or inconsistent behavior. Additional risks include overfitting due to model complexity, inefficient handling of high-dimensional data, and unstable recommendations in extremely sparse, noisy, or rapidly changing environments. These limitations highlight the importance of context-aware hybridization.

The continued evolution of hybridization methods, particularly through machine learning and contextual adaptation, remains a key focus in recommender system research. Despite the fact that designing and implementing hybrid recommender systems can be complex, they can offer more comprehensive and effective recommendations, addressing the limitations of using content-based or collaborative filtering alone.

Modern challenges in recommender systems. Several fundamental challenges of recommender system such as the cold start problem, data sparsity, and scalability have been discussed in earlier sections of this paper. While these issues are central to recommendation accuracy and system performance, they do not fully capture the complexity of deploying recommender systems in real-world applications. In practice, a number of additional challenges must be considered to ensure robustness, efficiency, fairness, and user trust in operational systems.

From a system-level perspective, latency remains a critical concern, especially in interactive and time-sensitive applications such as e-commerce and online advertising. Even when scalability challenges are addressed through distributed architectures, complex recommendation models may still introduce unacceptable response delays. High latency negatively affects user experience and can lead to reduced engagement or abandonment. Common mitigation approaches include offline precomputation of recommendations, model simplification for real-time inference, and approximate similarity search techniques that trade minimal accuracy loss for substantial speed improvements (Roy, & Dutta, 2022).

Beyond technical performance, bias and fairness have emerged as key ethical challenges in modern recommender systems. Recommendation algorithms may unintentionally reinforce existing biases present in training data, resulting in popularity bias, exposure bias, or demographic bias. Such effects can limit content diversity, disadvantage minority groups, and raise concerns about transparency and accountability. Addressing these issues requires fairness-aware modeling, diversity-promoting re-ranking strategies, and regular auditing of recommendation outputs to balance accuracy with equitable exposure (Mehrabi et al., 2021).

One important yet often overlooked challenge is the vulnerability of recommender systems to shilling attacks, also referred to as profile injection attacks. In such attacks, malicious users or automated agents intentionally introduce biased or fraudulent interactions to manipulate recommendation outcomes, for example by artificially promoting specific items or suppressing competitors. These attacks can significantly distort ranking mechanisms and undermine the credibility of recommendation platforms. Mitigation strategies include anomaly detection techniques, trust-aware recommendation models, and machine learning approaches that identify abnormal rating patterns and suspicious user behavior (Gunes et al., 2014).

Another challenge arises from semantic ambiguity and synonymy in item descriptions and user queries. In content-based and knowledge-aware recommender systems, different terms may refer to the same concept, leading to fragmented representations and reduced retrieval effectiveness. This issue is particularly prevalent in systems that rely on textual metadata or user-generated content. Advances in natural language processing, such as latent semantic analysis and neural word embedding models, have been shown to alleviate synonymy by capturing semantic similarity in latent representation spaces (Roy, & Dutta, 2022).

Privacy concerns are increasingly important as recommender systems rely on extensive collection and processing of personal user data. The use of behavioral signals and detailed user profiles raises risks related to data leakage, unauthorized inference, and regulatory non-compliance. Privacy-preserving techniques, including differential privacy, federated learning, and secure computation, have been proposed to reduce these risks while maintaining recommendation effectiveness. Moreover, transparent data usage policies and user-controlled privacy settings play a crucial role in fostering trust and long-term adoption of recommendation technologies (Yang et al., 2019).

Modern recommender systems must operate under complex and often conflicting constraints, balancing accuracy, efficiency, robustness, fairness, and privacy. Addressing these challenges requires integrated solutions that combine advances in machine learning, distributed systems, and ethical AI, moving beyond purely performance-driven optimization toward socially responsible recommendation technologies.

Discussion and conclusions

The analytical comparison conducted in this study demonstrates that the effectiveness of recommender systems is determined not by the choice of a single algorithmic paradigm, but by the alignment between model assumptions, data characteristics, and operational constraints. As summarized in Table 1, content-based and collaborative filtering exhibit complementary strengths that manifest across multiple evaluation dimensions, including data requirements, interpretability, scalability, robustness, and adaptability.

The analysis confirms that content-based filtering is most reliable in environments with well-structured and expressive item metadata, where interpretability and predictable time performance are critical. Its stable computational behavior and robustness to item cold-start scenarios make it particularly suitable for domains with rapidly evolving catalogs or limited interaction data. However, empirical evidence reviewed in this work indicates that recommendation quality in content-based systems is highly sensitive to feature engineering choices and similarity measures, and that over-specialization remains a persistent limitation in practice.

Collaborative filtering, in contrast, demonstrates superior capability in uncovering latent user preferences and promoting serendipity when interaction data are sufficiently dense. Both memory-based and model-based approaches achieve high accuracy under favorable data conditions, yet the analysis highlights substantial trade-offs in terms of scalability, interpretability, and robustness to sparsity. Experimental findings reviewed in the paper further indicate that similarity measure selection and computational cost play a critical role in large-scale deployments, often requiring approximation techniques or offline training pipelines to maintain acceptable response times.

Hybrid recommender systems emerge from the analysis as a practically validated strategy for mitigating the structural limitations of single-method approaches. Different hybridization schemes address different problem dimensions: switching hybrids effectively handle cold-start scenarios, weighted and feature-level hybrids improve accuracy and robustness, while cascading approaches refine ranking quality with limited overhead. At the same time, the review makes clear that hybrid systems introduce increased model complexity, higher maintenance costs, and sensitivity to integration design choices, which can lead to unstable behavior if not carefully managed.

Several open challenges remain evident from the analytical perspective of this study. First, there is no universally optimal hybridization strategy. Performance strongly depends on domain characteristics, data quality, and system constraints. Second, evaluation practices remain fragmented, with accuracy metrics dominating experimental studies, while user-centric measures such as satisfaction, trust, and long-term engagement are less consistently addressed. Finally, scalability, fairness, and privacy considerations increasingly shape real-world recommender system design but are not yet systematically integrated into comparative evaluation frameworks.

Table 1

Comparative analysis of recommendation approaches, summary

Criterion	Content-based filtering	Collaborative filtering	Hybrid filtering
Core principle	Recommends items similar to those a user liked in the past based on item features	Recommends items based on preferences of similar users or items	Combines content-based and collaborative signals using different hybridization strategies
Required data	Item metadata and user profiles	User-item interaction matrix	Item metadata, interaction data, and possibly contextual features
Personalization level	High (user-specific models)	High (community-driven patterns)	High, with increased robustness due to multi-source signals
Recommendation accuracy	Moderate; limited by feature quality	High when sufficient interaction data is available	Generally higher than single methods if properly tuned
Ability to recommend novel items	Strong (does not rely on popularity)	Weak (new items lack interactions)	Improved compared to collaborative filtering; depends on hybrid type
Cold-start (user)	Handles reasonably well	Performs poorly	Typically mitigated, but not fully eliminated
Cold-start (item)	Handles well if metadata is available	Performs poorly	Largely alleviated through content signals
Diversity and serendipity	Low to moderate; tends to over-specialization	Higher diversity due to social effects	Higher than content-based filtering; controllable via hybrid design
Explainability	High (feature-based explanations)	Low to moderate	Medium to high, depending on dominant component
Scalability	Scales well with users; depends on feature dimensionality	Scalability issues for large user-item matrices	Often more computationally expensive than single methods
Robustness to noise	Sensitive to poor or sparse metadata	Sensitive to noisy or biased interaction data	More robust due to redundancy of information sources
Adaptability to context	Limited unless explicitly modeled	Limited	High when contextual or feature-level hybrids are used
Model complexity	Low to moderate	Moderate to high	High, varies by hybridization strategy
Implementation complexity	Relatively simple	More complex (similarity learning, factorization)	Most complex; requires integration and tuning
Maintenance cost	Low	Moderate	High (multiple models, parameters, and pipelines)
Typical failure cases	Overspecialization, feature misrepresentation	Data sparsity, popularity bias	Error propagation, overfitting, misbalanced model dominance
Best-suited scenarios	Domains with rich item descriptions and stable user interests	Large-scale platforms with dense interaction data	Production systems requiring robustness and higher accuracy

Future research should therefore focus on developing standardized, multi-criteria evaluation methodologies that jointly assess accuracy, efficiency, interpretability, and ethical considerations. Promising directions include context-aware and adaptive hybridization strategies, integration of lightweight machine learning models to balance performance and complexity, and exploration of privacy-preserving and fairness-aware recommendation techniques. From a practical standpoint, the findings of this study suggest that robust recommender systems should be designed as adaptive architectures rather than static algorithmic choices, with hybrid models serving as a flexible foundation for addressing evolving data and application requirements.

Sources of funding. This study did not receive any grant from a funding institution in the public, commercial, or non-commercial sectors. The funding is covered by the authors' own expense.

References

Anisimov, A. V., Marchenko, O. O., & Kysenko, V. K. (2011). A method for the computation of the semantic similarity and relatedness between natural language words. *Cybernetics and Systems Analysis*, 47, 515–522. https://doi.org/10.1007/s10559-011-9334-2

Anisimov, A. V., Marchenko, O. O., & Vozniuk, T. G. (2014). Determining Semantic Valences of Ontology Concepts by Means of Nonnegative Factorization of Tensors of Large Text Corpora. *Cybernetics and Systems Analysis*, 50, 327–337. https://doi.org/10.1007/s10559-014-9621-9

Baxla, M. A. (2014). *Comparative study of similarity measures for item based top n recommendation* [Unpublished thesis, National Institute of Technology Rourkela]. CORE. https://files.core.ac.uk/download/53190130.pdf

Belhaouari, S. B., Fareed, A., Hassan, S., & Halim, Z. (2023). A collaborative filtering recommendation framework utilizing social networks. *Machine Learning with Applications*, 14, 1–20. https://doi.org/10.1016/j.mlwa.2023.100495

Burke, R. (2007). Hybrid Web Recommender Systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *Lecture Notes in Computer Science,* 4321. *The Adaptive Web* (pp. 377–408). Springer. https://doi.org/10.1007/978-3-540-72079-9_12

Çano, E. (2017). Hybrid Recommender Systems: A Systematic Literature Review. *Intelligent Data Analysis*, 21, 1487–1524. https://doi.org/10.3233/IDA-163209

Deutschman, Z. (2023, August 7). *Recommender Systems: Machine Learning Metrics and Business Metrics.* Neptune AI. https://neptune.ai/blog/recommender-systems-metrics

Fkih, F. (2022). Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. *Journal of King Saud University - Computer and Information Sciences*, 34(9), 7645–7669. https://doi.org/10.1016/j.jksuci.2021.09.014

Gaurav, P. (2023, February 14). *Step by Step Content-Based Recommender system.* Medium. https://medium.com/@prateekgaurav/step-by-step-content-/based-recommendation-system-823bbfd0541c

Gershman, A., Meisels, A., Luke, K.-H., Rokach, L., Schclar, A., & Sturm, A. (2010). A Decision Tree Based Recommender System. In G. Eichler, P. Kropf, U. Lechner, P. Meesad, & H. Unger (Eds.), *10th International Conference on Innovative Internet Community Services: Vol. 165. Lecture Notes in Informatics* (pp. 170–179). Gesellschaft für Informatik. https://dl.gi.de/server/api/core/bitstreams/ca0e5035-3a82-48a1-8eb8-8f49ee374161/content

Gosh, S., Nahar, N., Wahab, M. A., Biswas, M., Hossain, M. S., & Andersson, K. (2021). Recommender system for E-commerce Using Alternating Least Squares (ALS) on Apache Spark. In P. Vasant, I. Zelinka, & G. W. Weber (Eds.), *Intelligent Computing and Optimization: Vol. 1324. Advances in Intelligent Systems and Computing* (pp. 880–893). Springer. https://doi.org/10.1007/978-3-030-68154-8_75

Grover, P. (2017, December 28). *Various Implementations of Collaborative Filtering.* Medium. https://towardsdatascience.com/various-implementations-of-/collaborative-filtering-100385c6dfe0

Gunes, I., Kaleli, C., Bilge, A., & Polat, H. (2014). Shilling attacks against recommender systems: a comprehensive survey. *Artificial Intelligence Review*, 42, 767–799. https://doi.org/10.1007/s10462-012-9364-9

Herimanto, H, Samosir, K., & Ginting, F. (2024). A Comparative Analysis of Content-Based Filtering and TF-IDF Approaches for Enhancing Sports Recommendation Systems. *Innovation in research of informatics*, 6(2), 90–97. https://doi.org/10.37058/innovatics.v6i2.12404

Joy, J., & Renumol, V. G. (2020). Comparison of Generic Similarity Measures in E-learning Content Recommender System in Cold-Start Condition. In *IEEE Bombay Section Signature Conference* (pp. 175–179). Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/IBSSC51096.2020.9332162

Kumar, S. (2022, September 25). *Collaborative Filtering based Recommender Systems for Implicit Feedback Data.* Sumit's Diary. https://blog.reachsumit.com/posts/2022/09/explicit-implicit-cf/

Liu, Y., Wang, S., Khan, M. S., & He, J. (2018). A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering. *Big Data Mining and Analytics*, 1(3), 211–221. https://doi.org/10.26599/BDMA.2018.9020019

Mandal, S., & Maiti, A. (2018). Explicit Feedbacks Meet with Implicit Feedbacks: A Combined Approach for Recommender system. In L.M. Aiello, H. Cherifi, P. Lió, L.M. Rocha, C. Cherifi, R. Lambiotte (Eds.), *7th International Conference on Complex Networks and their Applications: Vol. 813. Studies in Computational Intelligence* (pp. 169–181). Springer. https://doi.org/10.1007/978-3-030-05414-4_14

Marchenko, O. O. (2016). A Method for Automatic Construction of Ontological Knowledge Bases. Development of a Semantic-Syntactic Model of Natural Language. *Cybernetics and Systems Analysis*, 52, 20–29. https://doi.org/10.1007/s10559-016-9795-4

Marchenko, O., & Shevchenko, M. (2024). Influence of distance measures and data characteristics on time performance in content-based and collaborative filtering datasets. In A. Anisimov, V. Snytyuk, A. Chris, A. Pester, F. Mallet, I. Krak, N. Cogan, O. Chertov, O. Marchenko, S. Bozóki, T. Needham, V. Tsyganok, & V. Vovk (Eds.), *Information Technology and Implementation: Vol. 3909. Central Europe University Repository Workshop Proceedings* (pp. 99–108). CEUR-WS. https://ceur-ws.org/Vol-3909/Paper_8.pdf

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, *54*(6), 115. https://doi.org/10.1145/3457607

Meteren, R., & Someren, M. (2000). Using Content-Based Filtering for Recommendation. In *Proceedings of the machine learning in the new information age: Vol. 30. MLnet/ECML2000 workshop* (pp. 47–56). ICS. https://users.ics.forth.gr/~potamias/mlnia/paper_6.pdf

Analytics Vidhya. (2024, October 14). *Movie Recommendation and Rating Prediction using K-Nearest Neighbors*. https://www.analyticsvidhya.com/blog/2020/08/recommendation-system-k-nearest-neighbors/

Nguyen, A. (2016). *Singular Value Decomposition in Recommender Systems* [Honors project, Texas Christian University]. TCU Digital Repository. https://repository.tcu.edu/server/api/core/bitstreams/7483e691-6fc0-4a82-9185-3adeb00cde44/content

Qazi, M., Fung, G. M., Meissner, K. J., & Fontes, E. R. (2017). An insurance recommendation system using Bayesian networks. In *11th ACM Conference on Recommender Systems* (pp. 274–278). Association for Computing Machinery. https://doi.org/10.1145/3109859.3109907

Ricci, F. (2002). *Content-Based Filtering and Hybrid Methods*. EIA. http://eia.udg.es/arl/Agentsoftware/3-ContentBasedHybrid.pdf

Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, 9, 59. https://doi.org/10.1186/s40537-022-00592-5

Sabiri, B., Khtira, A., El Asri, B., & Rhanoui, M. (2025). Hybrid Quality-Based Recommender Systems: A Systematic Literature Review. *Journal of Imaging*, 11(1), 12. https://doi.org/10.3390/jimaging11010012

Saranya, K. G., Sadasivam, G. S., & Chandralekha, M. (2016). Performance Comparison of Different Similarity Measures for Collaborative Filtering Technique. *Indian Journal of Science and Technology*, 9(29), 1–8. https://doi.org/10.17485/ijst/2016/v9i29/91060

Steck, H. (2019). Markov Random Fields for Collaborative Filtering. *Advances in Neural Information Processing Systems*, 32, 5473–5484. https://doi.org/10.48550/arXiv.1910.09645

Sun, S.-B., Zhang, Z.-H., Dong, X.-L., Zhang, H.-R., Li, T.-J., Zhang, L., & Min, F. (2017). Integrating Triangle and Jaccard similarities for recommendation. *PLoS ONE*, 12(8), e0183570. https://doi.org/10.1371/journal.pone.0183570

Vijay, H. (2020, April 11). *Recommendation System using kNN.* Auriga. https://aurigait.com/blog/recommendation-system-using-knn/

Wayesa, F., Betalo, M. L., Asefa, G. & Kedir, A. (2023). Pattern-based hybrid book recommender system using semantic relationships. *Scientific Report*, 13, 3693. https://doi.org/10.1038/s41598-023-30987-0

Wijewickrema, M., Petras, V., & Dias, N. (2019). Selecting a text similarity measure for a content-based recommender system: A comparison in two corpora. *The Electronic Library*, 37(3), 506–527. https://doi.org/10.1108/EL-08-2018-0165

Xia, Z., Sun, A., Xu, J., Peng, Y., Ma, R., & Cheng, M. (2024). Contemporary Recommendation Systems on Big Data and Their Applications: A Survey. *IEEE Access*, 12, 196914–196928. https://doi.org/10.1109/ACCESS.2024.3517492

Yadav, V., Shukla, R., Tripathi, A., & Maurya, A. (2021). A New Approach for Movie Recommender System using K-means Clustering and PCA. *Journal of Scientific & Industrial Research*, 80(2), 159–165. https://doi.org/10.56042/JSIR.V80I02.40102

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2), 12. https://doi.org/10.48550/arXiv.1902.04885

Zisopoulos, Z., Karagiannidis, S., Demirtsoglou, G., & Antaris, S. (2008, October). *Content-Based Recommender systems.* ResearchGate. https://www.researchgate.net/publication/236895069_Content-Based_Recommendation_Systems

Отримано редакцією журналу / Received: 21.07.25
Прорецензовано / Revised: 20.09.25
Схвалено до друку / Accepted: 20.09.25

Максим ШЕВЧЕНКО, асп.
ORCID ID: 0009-0008-5104-9767
e-mail: maksym_shevchenko@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

АНАЛІТИЧНИЙ ОГЛЯД КОНТЕНТНОЇ ТА КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЙ У РЕКОМЕНДАЦІЙНИХ СИСТЕМАХ

Вступ. ***За стрімкого зростання обсягу цифрового контенту рекомендаційні системи стають ключовим інструментом для надання персоналізованих пропозицій. Вони сприяють відкриттю нових фільмів, музики та товарів, підтримуючи зацікавленість користувачів у використанні платформ. Актуальність дослідження алгоритмів рекомендаційних систем зумовлена необхідністю вдосконалення їхньої роботи для задоволення індивідуальних уподобань користувачів. Ця робота являє собою огляд та аналітичне дослідження алгоритмів рекомендаційних систем. Метою цієї роботи є систематизація, класифікація та критичний аналіз двох основних підходів у рекомендаційних системах: фільтрації на основі вмісту (контентної) та колаборативної фільтрації.***

Методи. ***Огляд існуючих методів рекомендаційних систем, порівняльне й аналітичне дослідження.***

Результати. ***Проаналізовано алгоритми рекомендаційних систем. Дано формальне визначення задачі рекомендацій, де вподобання користувачів моделюються як функціональна залежність від властивостей об'єктів. У межах фільтрації на основі вмісту розглянуто використання класифікаційних алгоритмів, таких як наївний баєсів класифікатор, і дерев рішень, а також алгоритму Роккіо, який застосовує релевантний зворотний зв'язок для оновлення профілю користувача. Проведено аналіз сильних і слабких сторін різних мір подібності між векторами. У колаборативній фільтрації досліджено memory-based підхід (user-based та item-based методи) і model-based техніки з акцентом на алгоритмі k-NN. Для подолання недоліків окремих методів запропоновано гібридний підхід, який об'єднує їхні переваги. Представлено способи інтеграції систем у гібридну модель, що дає змогу покращити точність рекомендацій.***

Висновки. ***Результати роботи виокремлюють особливості зазначених методів фільтрації, демонструють вплив реалізації алгоритмів і вхідних даних на точність рекомендацій і час відповіді. Аналіз недоліків підкреслює значення комбінованого використання алгоритмів фільтрації для підвищення ефективності рекомендаційних систем, що робить гібридний підхід перспективним напрямом для подальших досліджень і впровадження.***

Ключові слова: ***алгоритм Роккіо, векторна модель, гібридна фільтрація, колаборативна фільтрація, контентна фільтрація, рекомендаційні системи.***

Автор заявляє про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The author declares no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.