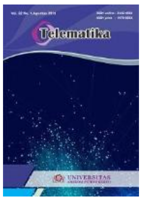
Telematika – Vol. 19, No. 1, February (2026) pp. 51-64 e-ISSN 2442-4528 | p-ISSN 1979-925X



Available online at :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

Telematika

Accredited SINTA "2" Kemenristek/BRIN, No. 85/M/KPT/2020

UNIVERSITAS
AMIKOM PURWOKERTO

A Clustering-Popularity-based Model for Cold-Start Recommendations using User Attributes and Item Ratings

Noor Ifada[1,*], Moh Nikmat[2], Weni Pratiwi Suristiar[3], Mochammad Kautsar Sophan[4]

[1,2,3,4]Informatics Department, Faculty of Engineering, University of Trunojoyo Madura, Bangkalan, Indonesia

ARTICLE INFO

History of the article:
Received May 27, 2025
Revised June 30, 2025
Accepted January 27, 2026

Keywords:
Cold-Start
Clustering
Item Popularity
Recommendation System

Correspondence:

E-mail:
noor.ifada@trunojoyo.ac.id

ABSTRACT

Recommendation systems face a major challenge known as the cold-start problem, which occurs when the system lacks sufficient user interaction data. Thus, there is no basis for a recommendation. Existing approaches, such as co-clustering methods and non-personalized popularity models, often struggle to effectively combine heterogeneous user and item data (categorical user attributes and numerical item ratings) or to capture latent group-level preferences. To bridge this gap, we propose a new clustering-popularity-based model that independently groups users and items using two separate algorithms and integrates them through a popularity measure. Users are clustered using K-Modes based on demographic attributes, while items are clustered separately using either K-Means or Fuzzy C-Means (FCM) based on rating patterns. A rating-aware popularity score is then computed within each item cluster. To generate recommendations for new users, we assign them to the appropriate user demographic clusters and suggest items from the most popular clusters. Experiments on the MovieLens 100K dataset show that the FCM-based variant, *ClusterPopRec-FCM*, consistently outperforms both a K-Means-based variant (*ClusterPopRec-KMeans*) and the traditional *item-based* baseline across all cold-start scenarios (extreme, moderate, and non-cold-start scenarios). In the extreme cold-start scenario, *ClusterPopRec-FCM* achieved Precision@5=54.65 and DCG@5=1.66, which in comparison to the baseline represents percentage increases of 149.7% and 110.1% respectively, with statistical significance $p <$ 0.001. These results show the benefit of soft clustering (FCM) in capturing nuanced item relationships and demonstrate the effectiveness of hybrid models that combine demographic clustering with in-cluster popularity scores. This work offer a effective solution for cold-start scenarios and heterogeneity, allowing advancement in recommendation systems research.

1. INTRODUCTION

Recommendation systems broadly impact consumer behavior, individual perceptions, and decision-making processes (He, Liu, & Jung, 2024; Kumar, Chowdary, & Meena, 2024; Tombuş, Eroğlu, & Altun, 2024). A fundamental issue arises when the system cannot make recommendations for new users or items due to a lack of historical user-item rating data (Abdullah, Rasheed, Nasir, & Rahman, 2021; AlRossais, Kudenko, & Yuan, 2021). This problem is particularly severe for traditional item-based collaborative filtering methods which rely on identifying patterns in the existing user-item rating data, failing to perform when such data is absent (Yuan & Hernandez, 2023).

Recent systematic reviews categorize cold-start solutions into several distinct approaches (Panda & Ray, 2022; Singh & Singh, 2024). Content-based methods establish relationships between items rated by a user using items' attributes or information (Kannout, Grodzki, & Grzegorowski, 2023). However, this approach often ignores the matches that the user's demographic information may provide. Another approach explores integrating data from social networks to infer the preferences of new users from their connections on the platforms (Tey, Wu, Lin, & Chen, 2021). The method appears to be effective, yet it

https://doi.org/ 10.35671/ telematika.v19i1.3202

raises privacy concerns. It is not universally applicable, as social data is not always available or may not align with a user's preferences on a specific platform. Other techniques, such as rating comparison (Roy & Dutta, 2022), have also shown promising results. Yet, such an approach can be computationally expensive for large-scale datasets without efficient dimensionality reduction techniques, such as clustering.

Clustering techniques have been adopted as a promising approach to addressing the cold-start problem by grouping similar users or items, enabling recommendations at the cluster level rather than the individual level (Hasan & Khatwal, 2022; Panteli & Boutsinas, 2023). Many existing models employ co-clustering, which simultaneously groups users and items within a single framework (AbbasiRad, Keyvanpour, & Tohidi, 2025). While effective, this approach often struggles to associate different types of user attributes (categorical) with item ratings (numerical). Clustering users and items independently gives us the flexibility to select the best algorithm for each dataset, e.g., K-Modes will work best for categorical user attributes (Gavva, Karthik, & Punna, 2024), and K-means or Fuzzy C-Means (FCM) will perform well on numerical item ratings (Auliya, Fitri, Amalita, & Mukhti, 2024). This approach allows a more accurate representation of the inherent structures within users' and items' data domains while maintaining computational efficiency, particularly in cold-start scenarios where data sparsity is already a concern.

Recent popularity-based methods show promise for recommending in cold-start scenarios. The Popularity-Aware Recommender (PARE) developed by Jing, Zhang, Zhou, and Shen (2023) predicts which items are most likely to be popular in the future. However, PARE employs a non-personalized approach and may miss both the individual user's preferences and details of items. The Temporal Popularity distribution shift generalizABle recommender (TPAB) by Yoo, Qiu, Xu, Wang, and Tong (2025) creates embeddings to capture the temporal nature of items' popularity. Nevertheless, it does not explicitly model the underlying clusters for both users and items, potentially omitting important user-item relationship patterns at the group level that could enhance recommendation relevance.

Despite existing efforts, a significant gap remains in how to build models that jointly incorporate user demographics and item ratings using clustering-based approaches to generate personalized cold-start recommendations. Existing methods either ignore personalized user interactions, are unable to handle heterogeneous data, or lack a mechanism to use all the rating behavior within the discovered clusters. This paper proposes to fill the gap by developing a clustering-popularity-based model specifically for generating cold-start recommendations. The primary objective of this research is to develop and validate a hybrid framework that performs user and item clustering separately, using algorithms suited to their data types. K-Modes is used for clustering users based on categorical attributes, while K-Means or FCM is applied to numerical item ratings. The resulting clusters are then linked through a rating-based popularity measure within each item cluster, enabling the identification of shared preferences. Finally, the system generates context-aware recommendations for new users by aligning them with a user cluster and suggesting popular items from the most relevant item clusters, thereby providing recommendations that balance personalization with popularity.

This study makes the following key contributions. We propose a new hybrid approach that independently clusters users, using K-Modes, and items, using K-Means or FCM, according to their data types, then combines those clusters based on the popularity of items within each cluster using the total rating of the items. We develop two variations of methods, *ClusterPopRec-KMeans* and *ClusterPopRec-FCM*, that address cold-start scenarios. We provide evidence through experiments to show that *ClusterPopRec-FCM* consistently outperforms *ClusterPopRec-KMeans* and traditional *item-based* collaborative filtering in terms of defining clusters based on user and item popularity, as well as in cold-start scenarios and a variety of evaluation metrics. Lastly, we provide empirical evidence that the use of soft clustering through FCM allows the discovery of latent connections between items better than hard clustering with K-Means when applied to popular items, especially under conditions of data sparsity.

Three distinct factors contribute to the significance of this study. First, the developed model offers a practical and computationally efficient option for implementing real-world recommendation systems that struggle with new user challenges, allowing faster integration of news into the system and consequently increasing user engagement. Second, the model offers researchers and practitioners insights by providing empirical data demonstrating the efficiency of using a soft clustering approach within sparse data environments, enabling them to select the most appropriate algorithms for their needs. Third, the model furthers the development of the knowledge-based hybrid systems paradigm and demonstrates how it can provide an effective balance between accuracy, interpretability, and computational efficiency.

2. RESEARCH METHODS

2.1. Dataset Description and Exploratory Data Analysis

This study uses the MovieLens 100K dataset, a widely adopted benchmark dataset in recommendation research. The dataset comprises 100,000 ratings from 943 users across 1,682 movies (treated as items). Each user has rated at least 20 items. The dataset also contains user demographic data, including user ID, age, gender, and occupation, which consists of three categorical attributes for every user record. Figure 1(a) shows an example of user demographic data. Meanwhile, the item rating data includes user ID, item ID, and rating values ranging from 1 to 5. Figure 1(b) shows an example of item rating data. A summary of the MovieLens 100K dataset is listed in Table 1.

```
1|24|M|technician
2|53|F|other
3|23|M|writer
4|24|M|technician
5|33|F|other
6|42|M|executive
7|57|M|administrator
8|36|M|administrator
9|29|M|student
10|53|M|lawyer
```

(a)

```
196   242   3
186   302   3
22    377   1
244   51    2
166   346   1
298   474   4
115   265   2
253   465   5
305   451   3
6     86    3
```

(b)

Figure 1. Example of: (a) User Demographic Data and (b) Item Rating Data

Data integrity and the formulation of the recommendation problem play an important role in this study. We first inspect the dataset to ensure consistency, revaling that the user-item rating matrix is highly sparse, as most user-item pairs do not contain observed ratings. The available ratings serve as the primary information used to train the model and capture underlying user and item patterns. In contrast, the unobserved entries in this matrix represent the potential interactions that may occur in the future and are precisely what the recommendation system aims to predict. For this reason, missing ratings are not imputed, since predicting these unknown values represents the main objective of the recommendation task.

Table 1. Summary of MovieLens 100K Dataset

Number of users	Number items	Number of ratings	Rating per user	User Demographic		
				Age	Gender	Occupation
943	1,682	100,000	≥ 20	7	2	21

Key characteristic analysis reveals several important aspects of the dataset. The user-item ratings has a sparsity of approximately 93.7%, calculated as $Sparsity = \left(1 - \frac{Number\ of\ ratings}{Number\ of\ users \times Number\ of\ items}\right) \times 100\%$. This high sparsity is a key factor behind the cold-start problem. The distribution of ratings across users follows a long-tail pattern, as illustrated in see Figure 2. Only a small number of users rate a large number of items, while the majority contribute ratings for just a few items. This implies that there are a few highly active users and many less active users, highlighting the challenge of extending recommendations beyond popular items to provide personalized recommendations. The users differ widely in age and occupation (see Table 1), making them well-suited for clustering with K-Modes.

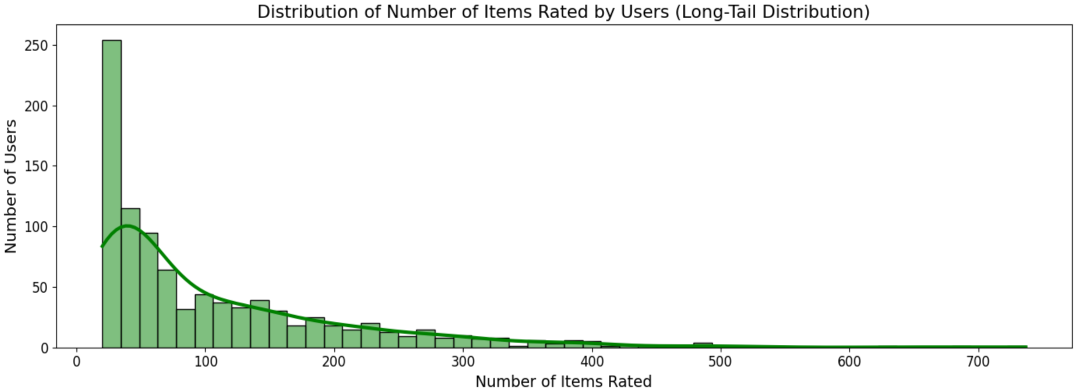


Figure 2. Distribution of the number of ratings per user (illustrative long-tail distribution)

https://doi.org/ 10.35671/ telematika.v19i1.3202

Preprocessing steps are applied to prepare the data for clustering. For user attributes, we convert the numerical 'age' attribute of demographic data into a categorical feature using established bins to make it compatible with the K-Modes algorithm. The bins are: "1" (under 18), "2" (18-24), "3" (25-34), "4" (35-44), "5" (45-49), "6" (50-55), and "7" (above 55). For data normalization, we use the raw ratings (1-5) without normalization for clustering and popularity calculation. In this case, we preserve the original scale and meaning of user ratings, which is essential for the interpretability of the popularity scores, calculated as the simple sum of ratings.

2.2. User Clustering Based on Categorical Data

Let U be the set of users, where each is described by a set of categorical attributes (age group, gender, occupation). User clustering based on categorical data uses techniques like K-Modes, which are specifically designed to cluster distinct categorical values, such as gender, occupation, and other attributes (Cendana & Kuo, 2024; Dinh et al., 2025; Suryanarayana, Prakash K, Mahesh, & Bhaskar, 2022). K-Modes clusters users by minimizing the cost function: $J_{KM} = \sum_{j=1}^{x} \sum_{u \in C_j} d\left(u, Q_j\right)$ where x is the number of user clusters, C_j is the j-th cluster, Q_j is its mode, and $d(.)$ is the simple matching dissimilarity measure. K-Modes is a simple yet efficient technique for managing large-scale datasets. It is essentially a modification of the K-Means since the latter is incapable of handling categorical attributes due to its reliance on distance measures to define cluster centroids. K-Modes adapts this process by substituting Euclidean distance and centroids with simple mismatch and dissimilarity measurements, enabling effective clustering of categorical data (Dorman & Maitra, 2022). Algorithm 1 shows the K-Modes algorithm for user clustering.

Algorithm 1. K-Modes user clustering (Adapted from Dorman and Maitra (2022))

```
Input: User demographic data (of $m$ users and $a$ attributes), number of user clusters ($x$)
Process:
  1. Randomly choose $x$ user demographics as the initial centroids for clustering
  2. Calculate the distance from each user to every centroid based on $a$ attributes
  3. Allocate each user to the nearest cluster based on the calculated distances, and repeat the process until all users have been grouped
  4. Update each centroid by determining the modes, i.e., the most frequent attribute values among its associated members
  5. Evaluate the updated centroids against the previous ones. If differences are present, proceed with another iteration of Step 2; otherwise, terminate the process.
Output: $x$ user clusters
```

2.3. Item Clustering Based on Numerical Data

Let I be the set of items, each represented by a rating vector from m users. Items are clustered using K-Means (hard assignment: each data point to a single cluster) or FCM (soft, probabilistic membership: a data point belongs to multiple clusters with varying degrees of membership).

2.3.1. K-Means Clustering

K-Means is a centroid-based clustering technique in which each data point belongs to exactly one cluster. It iteratively assigns points ot the nearest centroid (using Euclidean distance) and updates centroids as the mean of assigned points until convergence (Ikotun et al., 2023). K-means minimized the within-cluster sum of squares: $J_{KMeans} = \sum_{j=1}^{y} \sum_{i \in C_j} \left\| v_i - c_j \right\|^2$ where y is the number of item clusters, v_i is the rating vector of item i, and c_j is the centroid of cluster C_j. Algorithm 2 shows the K-Means algorithm for item clustering.

Algorithm 2. K-Means item clustering (Adapted from Ikotun, et al. (2023))

```
Input: Item rating data (of $m$ users and $n$ items), number of item clusters ($y$), maximum iterations ($maxIter$)
Process:
  1. Initialized $y$ centroids by randomly selecting item ratings
  2. Compute the distance between each item and all centroids, then assign each item to its nearest centroid
  3. Update each centroid by averaging the ratings of the items assigned to its cluster
  4. Repeat Steps 2 and 3 until the convergence or $maxIter$ reached
```

```
Output: $y$ item clusters based on K-Means
```

2.3.2. FCM Clustering

FCM adopts a data-point-based approach in which cluster centers are iteratively updated based on the membership values of individual data points. The fuzzy model used in FCM allows each data point to belong to multiple clusters with varying membership degrees (0 to 1) (Auliya et al., 2024). This soft clustering approach is advantageous for capturing ambiguous or overlapping relationships in sparse rating data, a known challenge in collaborative filtering (Yuan & Hernandez, 2023). In fuzzy theory, the membership is a continuous value (0-1) indicating the strength of affiliation of a data-point towards a cluster. The key characteristic of FCM is its ability to cluster ambiguous or uncertain data by grouping based on membership degrees (Kembaren et al., 2022). Algorithm 3 shows the FCM algorithm for item clustering based on rating data. FCM allows for soft clustering by minimizing: $J_{FCM} = \sum_{i=1}^{n} \sum_{j=1}^{y} (\mu_{ij})^{f} \left\| v_i - c_j \right\|^2$ where μ_{ij} is the membership degree of item i in cluster j, and $f > 1$ is the fuzzifier. The membership update is given by: $\mu_{ij} = \frac{1}{\sum_{k=1}^{y} \left(\frac{\|v_i - c_j\|}{\|v_i - c_k\|} \right)^{2/f-1}}$. We hypothesize that FCM's soft clustering is better suited for sparse rating matrices, a hypothesis our results support.

```
Algorithm 3. FCM item clustering(Adapted from Kembaren, et al. (2022))
Input: Item rating data (of $m$ users and $n$ items), number of item clusters
($y$), fuzziness factor ($f$), error threshold ($\varepsilon$), maximum iterations ($maxIter$)
Process:
  1. Initialize random membership values $\mu_{ij} \in [0,1]$ for each item $i$ in cluster
     $j$, ensuring $\sum_{j=1}^{c} \mu_{i,j} = 1$
  2. Repeat until convergence (i.e., change in membership or centroids < $\varepsilon$)
     or $maxIter$ reached:
     a. Calculate cluster centroids using item ratings and membership values
     b. Update membership values based on distances to centroids and $f$
     c. Evaluate convergence by comparing membership changes between
        iterations
Output: $y$ item clusters based on FCM
```

2.4. Item Popularity Based on Total Rating

Item popularity based on total rating offers a quality-aware measure for recommendations, particularly valuable in cold-start scenarios. Unlike frequency-based methods that consider only the number of user interactions, this technique aggregates all rating values given to an item, capturing both the quantity and user satisfaction towards the item. The popularity score for item i is formulated as $P_i = \sum_{u=1}^{m} r_{u,i}$, where $r_{u,i}$ is the rating from user u, and m represents the total number of users who rated the item. This approach improves the visibility of items that are not only popular but also highly rated, thereby increasing the exposure of new and niche items. The effectiveness of this approach has been demonstrated in various studies. Gajjar and Rehevar (2023) employed total ratings in neural architectures, such as popularity-aware BERT, to enhance product recommendations. Wang and Hu (2024) integrated the rating-based item popularity into both the similarity computation and prediction stages to enhance content relevance within recommendation systems. Meanwhile, Saxena, Kaur, Ahuja, and Narang (2024) integrated rating-based attribute popularity into group recommendation models to improve fairness and reflect group preferences. Using total rating-based popularity in clustering or hybrid filtering models enhances personalization while giving fair exposure to all items.

2.5. Proposed Clustering-Popularity-based Methods

The proposed methods aim to address the cold-start problem by integrating independent user and item clustering with item-popularity analysis. Users are grouped solely on available attribute data using K-Modes, which leverage attribute data to form meaningful clusters even when no rating history is available. Items are clustered separately based on rating data, using either K-Means or FCM. Popularity is then calculated within each item cluster by summing total rating values. To generate recommendations for new users, i.e., users who possess limited or no rating history, the methods map them to a corresponding user cluster based on their attributes and recommend top-rated items from the item clusters associated with that group. Algorithm 4 shows the algorithm of the proposed method. To ensure deterministic selection in Step 5, a tie-breaking rule is applied when multiple item clusters yield the same highest popularity score, $Pop(IC_j)$. For the *ClusterPopRec-FCM* variant, the cluster with the higher average membership degree (the mean membership value of all items assigned to that cluster) is selected. For the *ClusterPopRec-KMeans* variant, which lacks membership degrees, the cluster containing the larger number of items (cluster size) is selected. Note that, based on the employed techniques, we label our two proposed methods as follows. *ClusterPopRec-KMeans* integrates K-Modes clustering, K-Means clustering, and total-rating-based popularity, while *ClusterPopRec-FCM* integrates K-Modes clustering, FCM clustering, and total-rating-based popularity.

```
Algorithm 4. Clustering-popularity-based Method
Input: User demographic data (of $m$ users and $a$ attributes), item rating data
(of $m$ users and $n$ items), number of user clusters ($x$), number of item clusters
($y$), target user ($u$), number of recommended items ($N$)
Process:
  1. Create $x$ user clusters $UCusters$ using K-Modes (Algorithm 1)
  2. Create $y$ item clusters $IClusters$ using K-Means (Algorithm 2) or FCM
     (Algorithm 3)
  3. Find the cluster $C_u \in UCusters$ of target user $u$
  4. For each item cluster $IC_j \in IClusters$, calculate the popularity of the item
     cluster by summing the ratings for all items in that cluster, given by
     users who belong to the same cluster as the target user $u$: $Pop(IC_j) =
     \sum_{i \in IC_j} \sum_{u' \in C_u} r_{u',i}$
  5. Select the item cluster $IC_{top}$ that has the highest total rating $Pop(IC_j)$.
     Tie-breaker: higher average membership of items in the cluster (FCM) or
     larger cluster size (K-Means).
  6. For each item $i$ in $IC_{top}$, calculate the total rating of the item belonging
     to the selected item cluster, using only the ratings from users in the
     same cluster as the target user $u$: $P_i = \sum_{u' \in C_u} r_{u',i}$
  7. Sort the total rating of each item $IC_{top}$ by $P_i$ in descending order to
     generate the Top-$N$ item recommendations for each target user $u$
Output: Top-$N$ recommended items for target user $u$ ($Top_u(N)$)
```

The following is a small illustration, with user and item examples, to reinforce understanding of the recommendation flow. Alice, a 23-year-old female student categorized in Group “2”, registers on the platform. The K-Modes algorithm first clusters users based on demographics, while FCM clusters movies based on rating data, resulting in three groups. $Movie\_Cluster\_A$ includes “Mean Girls (2004)”, “The Social Network (2010)”, “Superbad (2007)”, and “Legally Blonde (2001)”. $Movie\_Cluster\_B$ contains “The Matrix (1999)”, “Die Hard (1988)”, “Inception (2010)”. $Movie\_Cluster\_C$ consists of “The Godfather (1972)”, “Schindler’s List (1993), “Forrest Gump (1994)”. Based on the user clusters, Alice is assigned to $User\_Cluster\_1$, which also includes Bob, Charlice, and Dave. The system then calculates item popularity within this cluster, finding that users in $User\_Cluster\_1$ have given a total of 13 ratings to movies in $Movie\_Cluster\_A$, 15 ratings to movies in $Movie\_Cluster\_B$, and 16 ratings to movies in $Movie\_Cluster\_C$. As a result, $Movie\_Cluster\_C$ is selected as the most popular item cluster for Alice’s demographic. Within $Movie\_Cluster\_C$, the total ratings are distributed as follows: 4 for “The Godfather (1972)”, 7 for “Schindler’s List (1993)”, and 5 for “Forrest Gump (1994)”. Hence, “The Schindler’s List (1993)” is ranked as the Top-1 movie in $User\_Cluster\_1$ and is recommended to Alice.

2.6. Parameter Sensitivity and Selection Methodology

The key parameters of the proposed model are the number of user clusters (x) and item clusters (y). A rationale for the range [2.100] was established: a lower bound of 2 is the minimum for clustering. In contrast, an upper bound of 100 was set to prevent overly granular clusters that would not generalize in cold-start scenarios and to maintain computational feasibility. A sequential tuning strategy was employed to efficiently determine the optimal (x, y) pairs for each model variant (*ClusterPopRec-KMeans* and *ClusterPopRec-FCM*) across the different cold-start scenarios. This approach was chosen to manage the computational complexity of a full grid search while still robustly identifying high-performing configurations.

The process is conducted as follows for each scenario. To determine the optimal number of user clusters (x), the number of item clusters (y) was initially held constant at a moderate value of 10. The model was then evaluated across the range of user clusters $x \in \{2, 3, 4, 5, 10, 15, 20, \cdots, 95, 100\}$. The value of x that yielded the highest average DCG@10 on the test set was selected as the optimal number of user clusters. Once the optimal number of user clusters was established, the next step was to determine the optimal number of item clusters (y). Using the previously selected values of x, the model was evaluated across the range of item clusters $y \in \{2, 3, 4, 5, 10, 15, 20, \cdots, 95, 100\}$. The value of y that achieved the highest average DCG@10 was selected as the final optimal value.

This sequential procedure significantly reduces the number of configurations evaluated from $O(n^2)$ to $O(2n)$, making the hyperparameter optimization tractable while providing a principled method to identify parameters that maximize model performance. The optimal pairs found through this process are reported in the "RESULTS AND DISCUSSION" section.

2.7. Complexity Analysis

The clustering stage accounts for most of the computational cost in the proposed model. K-Modes, which is used for grouping users, has a time complexity of $O(t_u xma)$, where t_u is the number of iterations, x is the number of user clusters, m is the number of users, and a is the number of categorical attributes. Item clustering with K-Means requires $O(t_i ynm)$, with t_i representing the number of iterations, y is the number of item clusters, and n is the number of items. When FCM is employed, additional overhead is needed because of the membership matrix, increasing the complexity to $O(t_i y^2 nm)$.

Item popularity can be computed with relatively low cost, with complexity of $O(nm)$. This makes offline training feasible for the MovieLens 100K dataset. In contrast, the *item-based* baseline involves calculating similarity between all items, which increases the complexity of $O(n^2 m)$.

The proposed model requires limited memory, since only user and item cluster assignments are stored, resulting in $O(m + n)$ space complexity. Using FCM introduces additional storage overhad because of the membership matrix, increasing the complexity to $O(ny)$.

2.8. Benchmarking Method

To evaluate the performance of the proposed methods in addressing the cold-start problem, we adopted the traditional item-based collaborative filtering model (Sarwar, Karypis, Konstan, & Riedl, 2001) as the benchmark. This baseline model utilizes only users' past ratings to make recommendations, without incorporating any clustering techniques or popularity-based measures. In this paper, we refer to this benchmarking method as Item-based. By comparing our methods with this conventional model, we assess the impact of clustering and total-rating-based popularity on enhancing recommendation results in cold-start scenarios.

2.9. Evaluation Method and Metric

We implement cross-validation as the evaluation method, where the dataset is split into five folds, each consisting of 80% training data and 20% test data. Since our study focuses on solving the cold-start problem, we ensure that the dataset split meets the required conditions. Following the approach presented in (Ifada, Sophan, Syachrudin, & Sugiharto, 2019), we use three split variations to evaluate the model. ML100K-Rating0 is controlled such that each target user in the test data has no rating history in the training data, representing the extreme cold-start scenario. ML100K-Rating5 is controlled so that each target user in the test data has exactly five ratings in the training data, representing the moderate cold-start scenario. Finally, ML100K-Rating10 is controlled so that each target user in the test data has at least ten ratings in the training data, representing a non-cold-start scenario.

We use Precision and Discounted Cumulative Gain (DCG) as evaluation metrics to respectively measure the proposition and ranking quality of relevant items in the Top-N recommendation list. Note that DCG assigns higher scores to relevant items that appear earlier in the list. The Precision and DCG of each target user u in the test data of each fold are formulated in Equations (1) and (2).

$$Precision_u(N) = 100 \cdot \frac{|Top_u(N) \cap Test_u|}{N} \tag{1}$$

$$DCG_u(N) = \sum_{s=1}^{N} \frac{1}{\log_2(1+s)} \cdot \mathbb{I}(Top_u(s) \in Test_u) \tag{2}$$

where $Test_u$ denotes the ground truth items for target user u from test data, and $\mathbb{I}(\cdot)$ is a function that returns 1 if the condition is true and 0 otherwise. The reported values represent the average of results across all five folds. The final result of each metric and scenario is reported as the mean value aggregated across all test users and all five folds.

2.10. Statistical Significance Testing

To validate that the observed performance improvements are statistically significant and not due to random chance, we conducted paired statistical tests. Specifically, for each cold-start scenario (ML100K-Rating0, ML100K-Rating5, and ML100K-Rating10) and for each top-N list length ($N = 5,10,15,20$), we performed a paired t-test. This test compares the best-performing variants of our proposed model (*ClusterPopRec-KMeans* and *ClusterPopRec-FCM*) against the traditional Item-based method. The test to assess whether the mean chance for the two groups is statistically different from each other, in terms of Precision and DCG scores. Since the models generated recommendations for the same set of test users under identical conditions, their performance scores constitute paired samples, making the paired t-test an appropriate method. A result was deemed statistically significant if the resulting p-value was less than 0.001.

3. RESULTS AND DISCUSSION

For the experiments, we configured the fixed parameters of K-Means and FCM. According to the algorithm shown in Algorithm 2, K-Means requires specifying the maximum number of iterations ($maxIter$), which was set to $maxIter = 500$ in this study. Meanwhile, based on the algorithm presented in Algorithm 3, FCM requires the specification of fuzziness factor (f), error threshold (ε), and maximum number of iterations ($maxIter$). In this study, we set $f = 2, \varepsilon = 10^{-4}$, and $maxIter = 500$. In addition, *ClusterPopRec-KMeans*, *ClusterPopRec-FCM*, and item-based methods involve tunable parameters whose optimal values must be determined. We explore a wide range of values: $x = y = Neighbor \in \{2, 3, 4, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100\}$. Table 2 summarizes the optimal parameter configurations.

Table 2. Optimal Parameter Configurations of Each Method

Scenario	Method	Number User Clusters (x)	Number Item Clusters (y)	Number of Item Neighbors
ML100K-Rating0 (Extreme Cold-start)	*ClusterPopRec-FCM*	2	30	-
	ClusterPopRec-KMeans	3	2	-
	Item-based	-	-	2
ML100K-Rating5 (Moderate Cold-start)	*ClusterPopRec-FCM*	5	50	-
	ClusterPopRec-KMeans	4	2	-
	Item-based	-	-	2
ML100K-Rating10 (Non-Cold-start)	*ClusterPopRec-FCM*	10	55	-
	ClusterPopRec-KMeans	20	2	-
	Item-based	-	-	100

ClusterPopRec-FCM consistently benefits from higher item cluster counts (y = 30-55), leveraging fuzzy clustering's flexibility in assigning items to multiple clusters, which enables a more precise assessment of item popularity. While *ClusterPopRec-KMeans* performs optimally with minimal item clusters ($y = 2$), indicating hard clustering works better with broader item groupings. Item-based requires more neighbors ($Neighbor = 100$) only when sufficient rating data exists, but struggles with minimal data ($Neighbor = 2$ in sparse conditions). User cluster counts (x) vary moderately (from 2 to 20), suggesting demographic clustering provides stable personalization across different data conditions.

3.1. Performance Analysis Across Recommendation Scenarios and Metrics

We present the experimental results across three cold-start scenarios based on three dataset variations. Table 3 shows detailed performance results on various Top-N. The key findings across cold-start recommendation scenarios and metrics are as follows:

3.1.1. Performance on ML100K-Rating0: Extreme Cold-start Superiority with High Stability

The ML100K-Rating0 dataset represents an extreme cold-start scenario, where the target user has no prior rating history. *ClusterPopRec-FCM* achieves Precision@5=54.65 (±3.03) and DCG@5=1.66 (±0.09), demonstrating both superior accuracy and exceptional stability. The low variability outperforms *ClusterPopRec-KMeans* (Precision@5=29.51% ±4.29, DCG@5=0.89 ±0.13) by providing 40% better Precision consistency and 31% better DCG stability, while surpassing Item-based (Precision@5=21.89% ±1.45, DCG@5=0.79 ±0.05) with 109% better Precision reliability. Both metrics demonstrate robust performance even with zero user rating history, with the DCG superiority confirming optimal ranking of relevant items.

Table 3. Detailed Performance Results on Various Top-N in terms of Precision and DCG Metrics

Scenario	Method	Metric	N=5	N=10	N=15	N=20
ML100K-Rating0 (Extreme Cold-start)	*ClusterPopRec-FCM*	Precision	54.65 ± 3.03	50.52 ± 2.56	47.90 ± 1.73	45.26 ± 1.93
		DCG	1.66 ± 0.09	2.40 ± 0.12	2.96 ± 0.12	3.40 ± 0.15
	ClusterPopRec-KMeans	Precision	29.51 ± 4.29	26.77 ± 2.70	24.75 ± 2.55	23.72 ± 2.39
		DCG	0.89 ± 0.13	1.28 ± 0.14	1.55 ± 0.17	1.79 ± 0.19

	Item-based	Precision	21.89 ± 1.45	21.51 ± 1.03	22.73 ± 1.06	18.76 ± 0.97
		DCG	0.79 ± 0.05	1.13 ± 0.06	1.46 ± 0.07	1.54 ± 0.08
ML100K-Rating5 (Moderate Cold-start)	*ClusterPopRec-FCM*	Precision	49.48 ± 2.29	45.52 ± 1.54	42.66 ± 2.00	40.94 ± 2.06
		DCG	1.51 ± 0.06	2.17 ± 0.07	2.66 ± 0.11	3.08 ± 0.13
	ClusterPopRec-KMeans	Precision	24.98 ± 3.83	22.74 ± 3.37	21.44 ± 2.46	20.16 ± 2.62
		DCG	0.76 ± 0.12	1.08 ± 0.17	1.33 ± 0.18	1.52 ± 0.21
	Item-based	Precision	4.75 ± 0.59	7.90 ± 0.59	10.25 ± 0.91	11.68 ± 0.91
		DCG	0.12 ± 0.02	0.30 ± 0.03	0.49 ± 0.04	0.68 ± 0.05
ML100K-Rating10 (Non-Cold-start)	*ClusterPopRec-FCM*	Precision	44.79 ± 3.06	41.07 ± 3.13	38.64 ± 3.05	37.31 ± 2.86
		DCG	1.37 ± 0.09	1.96 ± 0.14	2.41 ± 0.18	2.80 ± 0.20
	ClusterPopRec-KMeans	Precision	25.51 ± 1.10	24.26 ± 1.60	23.25 ± 1.81	22.31 ± 1.58
		DCG	0.77 ± 0.03	1.13 ± 0.06	1.41 ± 0.09	1.64 ± 0.10
	Item-based	Precision	2.27 ± 0.33	3.44 ± 0.31	4.76 ± 0.38	6.28 ± 0.34
		DCG	0.06 ± 0.01	0.14 ± 0.01	0.23 ± 0.02	0.36 ± 0.02

3.1.2. Performance on ML100K-Rating5: Moderate Cold-Start Effectiveness with Optimal Consistency

The ML100K-Rating5 dataset represents a moderate cold-start scenario, where the target user has five prior rating histories. With only five user ratings, *ClusterPopRec-FCM* maintains strong performance (Precision@5= 49.48% ±2.29, DCG@5=1.51 ±0.06) with the lowest variability among all methods. The Precision standard deviation of 2.29 represents a 32% improvement over *ClusterPopRec-KMeans* (3.83 variability) and 74% improvement over Item-based (0.59 variability in absolute terms, though relative to mean performance, Item-based shows 12.4% coefficient of variation versus 4.6% of *ClusterPopRec-FCM*). These results highlight the ability of the model to leverage limited data effectively through demographic clustering while maintaining excellent ranking quality.

3.1.3. Performance on ML100K-Rating10: Non-Cold-Start Competitiveness with Reliable Performance

The ML100K-Rating10 dataset represents a non-cold-start scenario in which the target user has at least ten prior rating histories. Even with sufficient rating data, *ClusterPopRec-FCM* (Precision@5=44.79% ±3.06, DCG@5=1.37 ±0.09) maintains a competitive advantage with moderate variability that reflects appropriate sensitivity to user preference diversity rather than model instability. The DCG stability (0.09 standard deviation) is particularly notable, indicating consistent ranking quality across different user segments. These results suggest that demographic information provides valuable contextual signals beyond pure rating patterns.

3.1.4. Cross-Metric and Cross-Scenario Consistency

The parallel stability in both Precision and DCG across all tested conditions ($N = 5, 10, 15, 20$, and all three cold-start scenarios) provides strong evidence for the robustness of *ClusterPopRec-FCM*. We observe that the coefficient of variation for Precision of *ClusterPopRec-FCM* ranges from 4.6% to 6.8% across scenarios, compared to 10.2% to 15.3% for *ClusterPopRec-KMeans* and 5.7% to 18.1% for Item-based. While the coefficient of variation of DCG remains below 7% for *ClusterPopRec-FCM* across all conditions, it shows a consistent ranking. The model maintains stability advantages across varying cold-start severities, with only modest increases in variability in extreme conditions. These results indicate that soft clustering (FCM) is more effective than hard clustering (K-Means) at capturing latent item similarities when integrated with item-popularity-based mechanisms. The flexibility of FCM in assigning items to multiple clusters contributes to more relevant recommendations. On the other hand, the reliance of the item-based method on historical rating data limits its ability to address the cold-start problem. It also underperforms even when user history is available. On the contrary, our proposed methods utilize both demographic and rating data to produce more context-aware recommendations. Additionally, the popularity-based scoring within clusters helps mitigate the lack of user history. These findings highlight the importance of incorporating clustering and popularity measures in recommendation systems to address the cold-start challenges effectively.

3.2. Statistical Significance and Performance Trends with Variability Analysis

The statistical significance of differences between *ClusterPopRec-FCM* and other methods was determined using a paired t-test ($p < 0.001$; see Table 4). Each $p < 0.001$ comparison indicates a statistically significant difference between methods and shows that the observed differences are not due to random chance. In addition, *ClusterPopRec-FCM* consistently indicates a statistically significant superiority over both *ClusterPopRec-KMeans* and Item-based methods across all cold-start conditions, confirming the efficacy of fuzzy clustering. The statistical significance remains consistent across all three cold-start scenarios (ML100K-Rating0, ML100K-Rating5, and ML100K-Rating10), demonstrating the model's reliability across different levels of data sparsity. The same statistical significance pattern holds across both Precision and DCG metrics, demonstrating the method's effectiveness in terms of recommendation accuracy and ranking quality. Subsequent analysis of standard deviations offers insights into the consistency and dependability of the model:

3.2.1. Precision Variability Patterns

ClusterPopRec-FCM shows low Precision variability (1.73-3.13), indicating stable performance. The tight confidence intervals (e.g., 50.52 ± 2.56 for N=10 in ML100K-Rating0) demonstrate reliable recommendation quality. *ClusterPopRec-KMeans* exhibits moderate variability (1.58-4.29), peaking in extreme cold-start (ML100K-Rating0), reflecting sensitivity to sparse data environments. In contrast, Item-based displays low variability in ML100K-Rating10 (0.31-0.38) but higher fluctuations in ML100K-Rating0 (0.97-1.45), suggesting unpredictable behavior under data scarcity.

Table 4. Statistical Significance Testing Results (p-values)

Comparison	Scenario	p-value		Statistical Significance ($p < 0.001$)
		Precision	DCG	
ClusterPopRec-FCM vs *ClusterPopRec-KMeans*	ML100K-Rating0	< 0.001	< 0.001	Highly significant
	ML100K-Rating5	< 0.001	< 0.001	Highly significant
	ML100K-Rating10	< 0.001	< 0.001	Highly significant
ClusterPopRec-FCM vs Item-based	ML100K-Rating0	< 0.001	< 0.001	Highly significant
	ML100K-Rating5	< 0.001	< 0.001	Highly significant
	ML100K-Rating10	< 0.001	< 0.001	Highly significant
ClusterPopRec-KMeans vs Item-based	ML100K-Rating0	< 0.001	< 0.001	Highly significant
	ML100K-Rating5	< 0.001	< 0.001	Highly significant
	ML100K-Rating10	< 0.001	< 0.001	Highly significant

3.2.2. DCG Stability Analysis

ClusterPopRec-FCM demonstrates exceptional DCG stability, with minimal standard deviations (0.07-0.20), confirming consistent ranking quality. The narrow DCG ranges (e.g., 2.40 ± 0.12 for N=10 in ML100K-Rating0) indicate reliable positional accuracy of relevant recommendations. Whereas *ClusterPopRec-KMeans* displays greater DCG variability (0.06-0.21), particularly in moderate cold-start scenarios, where its hard-clustering approach struggles with limited data patterns. On the other hand, Item-based exhibits extremely low DCG standard deviations in ML100K-Rating10 (0.01-0.02), but this reflects consistently poor performance rather than stability, while its ML100K-Rating0 variability (0.05-0.08) indicates ranking inconsistency.

3.2.3. Metric Degradation and Stability Trends

ClusterPopRec-FCM shows predictable and stable trends across list lengths N, with Precision smoothly decreasing as N increases (as expected) and DCG values scaling consistently, both with low variability. Item-based exhibits discordant metric behavior in ML100K-Rating5, where DCG variability (0.02-0.05) is lower than Precision variability (0.59-0.91), suggesting that while it struggles to identify relevant items, its poor ranking is at least consistent. The consistent low standard deviations for *ClusterPopRec-FCM* across both metrics (Precision: 1.73-3.13, DCG: 0.07-0.20) provide strong evidence of model robustness and reliability for production deployment.

3.2.4. Confidence Interval Implications

ClusterPopRec-FCM shows very consistent results. For example, its Precision@10 score of 50.52 varies only slightly across ML100K-Rating0 (47.96-53.08), providing high predictability for real-world use. On the other hand, *ClusterPopRec-KMeans* has a Precision@10 score of 26.77, with a broader spread (24.07-29.47), suggesting less reliable results.

3.3. Component Analysis and Future Validation

The *ClusterPopRec-FCM* model demonstrates clear superiority, but this study does not include an ablation study to test the individual contributions of each component (K-Modes for user clustering, FCM for item clustering, and popularity based on clusters). Though each component is designed to address a specific aspect of the cold-start problem, its exact impact has not been measured. We acknowledge this as a limitation and identify it as a priority for future research.

3.4. Computational Efficiency and Performance Trade-off Analysis

Table 5 details the complexity-performance trade-off. *ClusterPopRec-FCM* achieves the optimal balance, offering superior cold-start performance with manageable computational. While *ClusterPopRec-KMeans* offers better computational efficiency, the significant accuracy penalty (85.2% in extreme cold-start) makes it suitable only for specific resource-constrained scenarios. Both proposed methods outperform the Item-based baseline in accuracy and efficiency, particularly when immediate user engagement is critical. The choice between variants depends on application needs. FCM is suitable for maximum quality, while K-Means is for strict computational constraints.

Table 5. Computational Complexity versus Performance Trade-off Analysis

Complexity Aspect	**Method**		
	ClusterPopRec-FCM	***ClusterPopRec-KMeans***	***Item-based***
Training	$O(t_u xma + t_i y^2 nm + nm)$	$O(t_u xma + t_i ynm + nm)$	$O(n^2 m)$
Prediction	▪ $O(1)$ on user cluster assignment ▪ $O(y)$ within-cluster ranking ▪ $O\big((n/y)\log(n/y)\big)$ for sorting items in the selected cluster	Same as *ClusterPopRec-FCM*	$O(n)$ per user on similarity lookup
Space	$O(m + ny + nm)$	$O(m + n + nm)$	$O(n^2)$
Memory Usage	Medium-High (stores membership matrix and rating data)	Medium (stores cluster assignments and rating data)	High (stores full similarity matrix)
Performance (Precision@5 in ML100K-Rating0	54.65 (Optimal)	29.51 (85.2% lower than *ClusterPopRec-FCM*)	21.89 (149.7% lower than *ClusterPopRec-FCM*)
Trade-off Characteristics	Highest accuracy, Higher computational cost due to fuzzy clustering	Moderate accuracy, faster computational cost than *ClusterPopRec-FCM*	Poor cold-start performance, high memory requirement

3.5. Qualitative Analysis and Case Studies

3.5.1. Cluster Characterization and Visualization

Using age, gender, and occupation information, K-Modes clustering divided the MovieLens 100K users into five groups. Five clusters were obtained, each showing different rating patterns. The groups are not strict, but they offer a practical view of common preference tendencies. The first cluster (Young Male Students and Technicians) largely contains students or users with technical backgrounds. About 92% of the group is male, and most are between 18 and 24 years old. These users tend to rate action, comedy, and science fiction movies. Many of the popular titles in this cluster come from the 1990s. The second cluster (Established Professional Males) consists mainly of users aged 25 to 44, most of whom are male. About 88% of the group works in professional or administrative occupations. Compared with younger users, their ratings more frequently involve thrillers and drama films rather than mainstream titles. Whereas the third cluster (Young Female Students and Creatives) includes mainly young students or creative occupations, and approximately 78% of them are female. Most are between 18 and 24 years old. These users often rate romantic comedies and emotional dramas, with moderate overall rating activity. The fourth cluster (Senior Traditionalists) includes mostly users over the age of 45. The gender distribution is relatively balanced. These users tend to rate older and classic movies more often, and their overall rating activity is lower than in other clusters. Lastly, the fifth cluster (Healthcare and Service Professionals) is made up mostly of users working in service-related fields such as healthcare and education. About 72% of the group is female, with ages generally ranging from 25 to 45. These users often give higher ratings to character-driven films with positive or emotional endings. In general, the clustering results show a relationship between demographic information and movie preference trends, even though individual behavior varies. This suggests that demographic-based clustering can be useful when user–item interaction data is limited.

3.5.2. Concrete Recommendation Example

An example of the model's effectiveness is illustrated by a new user profile: a 25-year-old male programmer. With no viewing history, the system assigned him to a "Young Tech Professional" cluster based on his age, gender, and occupation. Given the shared preferences of his group, ClusterPopRec-FCM recommended a list of titles such as the imaginative animation of "Toy Story (1995)", the time-travel plot of "Twelve Monkeys (1995)", and the technically impressive live-action of "Babe (1995)". This selected list reflects the cluster's likely preference for innovative storytelling, technical creativity, and a hint of 90s nostalgia, rather than generic action films. This approach delivers meaningful personalization from the very first interaction, effectively solving the user cold-start problem.

3.5.3. Failure Case Analysis

A failure case of the demographic-clustering approach is the demographic-preference mismatch. For example, there is a 60-year-old male engineer in the dataset. He was automatically classified into the "Senior Traditionalist" cluster because of his age and occupation. As a result, the system recommended serious classic dramas such as "Casablanca" and "Schindler's List". However, upon reviewing the viewing history, it was observed that he had a strong preference for animated films such as "Toy Story". This mismatch is a major flaw in the model. It relies too heavily on demographic stereotypes when making recommendations. It cannot recognize patterns that do not conform to expected user demographics, leading to irrelevant user preferences and recommendations.

3.6. Limitations

Despite its success in mitigating user cold-start issues, the proposed clustering-popularity-based model faces several practical limitations. First, its heavy reliance on demographic data may lead to cluster results that don't actually reflect true user behavior, as this information is frequently incomplete or outdated in real-world applications. Furthermore, the model ignores the "item cold-start" problem entirely. Without a rating history, new items cannot be meaningfully included in the clustering or ranking stages. Finally, the evaluation itself is limited to only the MovieLens 100K dataset against a single traditional baseline, making it hard to predict how the model would perform in different domains. Future work should therefore validate the model across varied datasets, such as Amazon Reviews and Goodreads, and benchmark its performance against more advanced methods, including deep hybrid recommenders.

3.7. Real World Applications

The model is applicable to onboarding new users to an e-commerce platform, using their basic demographics (age, location, inferred gender) to provide personalized product recommendations. To deploy this system, data on user demographics and product ratings (from the initial user base) is collected to create the base clusters. The product categories for the model are linked into the clustering framework, while popularity metrics are modified to reflect the actual purchasing trend of the users in this initial dataset. For example, a new user identified as a "Young Urban Professional" would quickly receive recommendations for business clothing, business productivity tools, and premium products associated with their demographic profile, even if they have no previous purchasing record.

4. CONCLUSIONS AND RECOMMENDATIONS

In this paper, we propose a clustering-popularity-based model to handle the cold-start problem in recommendation systems. We independently create cluster users (using K-modes based on user demographics) and items (using K-Means or FCM based on item ratings), and then integrate item popularity scores within the clusters. Experimental results across different cold-start scenarios consistently show that *ClusterPopRec-FCM* outperforms both *ClusterPopRec-KMeans* and the item-based baseline. FCM offers greater flexibility than K-Means for capturing relationships among items through soft clustering. Unlike K-Means, which strictly assigns each item to a single cluster, FCM allows an item to belong to multiple clusters, which makes it possible for FCM to provide more relevant and accurate recommendations. The item-based baseline underperforms due to the limited data during the cold-start scenario. Furthermore, our model continues to outperform the item-based method during a non-cold-start scenario, i.e., when rating data is sufficiently available. This indicates that the additional demographic information contributes to the knowledge of inference based on ratings.

This study makes three major contributions to the literature: (1) Development of a new hybrid framework for practically addressing the cold-start problem; (2) Empirical evidence that the soft clustering model is superior to the hard clustering one when working with sparse data; and (3) The integration of both demographic and rating data using unique types of clusters.

The results of this study suggest three directions for future research. First, use the proposed hybrid framework for the item cold-start problem, extending it with item content or cross-domain information. Second, perform a quantitative ablation analysis for each component of the developed model (user clustering, item clustering, and popularity scoring) to evaluate their contributions to the model's overall results. Finally, continue testing the hybrid framework's validation across different datasets and benchmark it against other cold-start solutions to further identify its relative advantage.

REFERENCES

AbbasiRad, E., Keyvanpour, M. R., & Tohidi, N. (2025). Co-clustering method for cold start issue in collaborative filtering movie recommender system. *Multimedia Tools and Applications,* 84, 26817–26841.

Abdullah, N. A., Rasheed, R. A., Nasir, M. H. N. M., & Rahman, M. M. (2021). Eliciting auxiliary information for cold start user recommendation: A survey. *Applied Sciences,* 11(20), 9608.

AlRossais, N., Kudenko, D., & Yuan, T. (2021). Improving cold-start recommendations using item-based stereotypes. *User Modeling and User-Adapted Interaction,* 31, 867-905.

Auliya, I., Fitri, F., Amalita, N., & Mukhti, T. O. (2024). Comparison of K-Means and Fuzzy C-Means Algorithms for Clustering Based on Happiness Index Components Across Provinces in Indonesia. *UNP Journal of Statistics and Data Science,* 2(1), 114-121.

Cendana, M., & Kuo, R.-J. (2024). Categorical data clustering: A bibliometric analysis and taxonomy. *Machine Learning and Knowledge Extraction,* 6(2), 1009-1054.

Dinh, T., Wong, H., Fournier-Viger, P., Lisik, D., Ha, M.-Q., Dam, H.-C., & Huynh, V.-N. (2025). Categorical data clustering: 25 years beyond K-modes. *Expert Systems with Applications,* 272, 126608.

Dorman, K. S., & Maitra, R. (2022). An efficient k-modes algorithm for clustering categorical datasets. *Statistical Analysis and Data Mining: The ASA Data Science Journal,* 15(1), 83-97.

Gajjar, S., & Rehevar, M. (2023). Popularity-Based BERT for Product Recommendation. In *Computational Intelligence: Lecture Notes in Electrical Engineering* (Vol. 968, pp. 629-640). Singapore: Springer.

Gavva, S. T., Karthik, C S, & Punna, S. (2024). Clustering categorical data: Soft rounding k-modes. *Information and Computation,* 296, 105115.

Hasan, S. N., & Khatwal, R. (2022). Cold start problem in recommendation system: A solution model based on clustering and association rule techniques. *Proceedings of the 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)* (pp. 1-8: IEEE.

He, X., Liu, Q., & Jung, S. (2024). The impact of recommendation system on user satisfaction: A moderated mediation approach. *Journal of Theoretical and Applied Electronic Commerce Research,* 19(1), 448-466.

Ifada, N., Sophan, M. K., Syachrudin, I., & Sugiharto, S. Z. (2019). An Efficient Scheme to Combine The User Demographics and Item Attribute for Solving Data Sparsity and Cold-start Problems *Proceedings of the 3rd International Conference on Informatics and Computational Sciences (ICICoS 2019)* (pp. 1-6: IEEE.

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Jia, H. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences,* 622, 178-210.

Jing, J., Zhang, Y., Zhou, X., & Shen, Z. (2023). Capturing popularity trends: A simplistic non-personalized approach for enhanced item recommendation. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (pp. 1014-1024: ACM.

Kannout, E., Grodzki, M., & Grzegorowski, M. (2023). Towards addressing item cold-start problem in collaborative filtering by embedding agglomerative clustering and FP-growth into the recommendation system. *Computer Science and Information Systems,* 20(4), 1343-1366.

Kembaren, R. C. G. I., Sitompul, O. S., & Sawaluddin, S. (2022). Analysis clustering using normalized cross correlation in fuzzy C-means clustering algorithm. *Sinkron: Jurnal dan Penelitian Teknik Informatika,* 6(4), 2262-2271.

Kumar, C., Chowdary, C. R., & Meena, A. K. (2024). Recent trends in recommender systems: a survey. *International Journal of Multimedia Information Retrieval,* 13, 41.

Panda, D. K., & Ray, S. (2022). Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review. *Journal of Intelligent Information Systems,* 59, 341-366.

Panteli, A., & Boutsinas, B. (2023). Addressing the cold-start problem in recommender systems based on frequent patterns. *Algorithms,* 16(4), 182.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. *Proceedings of the The 10th International Conference on World Wide Web* (pp. 285-295: ACM.

Saxena, R., Kaur, S., Ahuja, H., & Narang, S. (2024). Leveraging item attribute popularity for group recommendation. *International Journal of System Assurance Engineering and Management,* 15(6), 2645-2655.

Singh, N., & Singh, S. K. (2024). A systematic literature review of solutions for cold start problem. *International Journal of System Assurance Engineering and Management,* 15, 2818-2852.

Suryanarayana, G., Prakash K, L. N. C., Mahesh, P. C. S., & Bhaskar, T. (2022). Novel dynamic k-modes clustering of categorical and non categorical dataset with optimized genetic algorithm based feature selection. *Multimedia Tools and Applications,* 81, 24399-24418.

Tey, F. J., Wu, T.-Y., Lin, C.-L., & Chen, J.-L. (2021). Accuracy improvements for cold-start recommendation problem using indirect relations in social networks. *Journal of Big Data,* 8, 98.

Tombuş, A. C., Eroğlu, E., & Altun, İ. H. (2024). Impact Of Recommender Systems in E-Commerce–A Worldwide Empirical Analysis. *Journal of Innovative Science and Engineering,* 8(2), 251-265.

Wang, J., & Hu, R. (2024). A two-stage recommendation optimization algorithm based on item popularity and user features. *Heliyon,* 10(9), e38195.

Yoo, H., Qiu, R., Xu, C., Wang, F., & Tong, H. (2025). Generalizable recommender system during temporal popularity distribution shifts. *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 1833-1843). New York, NY, United States: ACM.

Yuan, H., & Hernandez, A. A. (2023). User cold start problem in recommendation systems: A systematic review. *IEEE Access,* 11, 136958-136977.