informatics

MDPI

Article

An Explainable Collaborative Recommendation Framework Using K-Means Clustering and LLM-Based Explanations

Amjad Alaskar *, Eman Alasmari and Dimah Alahmadi

The Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; ealasmari0016@stu.kau.edu.sa (E.A.); dalahmadi@kau.edu.sa (D.A.)

* Correspondence: aaskar0002@stu.kau.edu.sa

Abstract

In social media contexts, recommendation systems are significant for enhancing the delivery of personalized content. More traditional recommendation systems aim primarily for recommendation accuracy, but do little on the topic related to the explainability and transparency of recommendation decisions. This paper presents an explainable user-based collaborative recommendation framework by leveraging K-Means clustering, similarity-based collaborative filtering, neighbor voting, and generative Large Language Model (LLM) explanations. The framework firstly utilizes K-Means clustering to cluster users with relatively similar behavioral traits. Pearson Similarity and Cosine Similarity are then utilized to select the Top-N similar users for each cluster. To generate personal recommendations for the content themes and content types, we employed similarity-based neighbor voting. Finally, Gemini 2.5 Flash is included as an explainability layer to create brief human-readable explanations of recommendation outputs. The findings presented a high behavioral similarity among adjacent users, which displayed Mean Pearson Similarity = 0.8044, Mean Cosine Similarity = 0.8113, and Mean Combined Similarity = 0.8078. It was found that a content-type recommendation performed better than a preferred-theme recommendation in social media platforms with large amounts of content. In general, this framework enhances interpretability, transparency, and the user understanding of recommendation while still providing efficient collaborative recommendation performance. It identified behaviorally similar users, and generated human-readable explanation suggestions through Gemini 2.5 Flash.

Keywords: collaborative filtering; explainable recommendation systems; generative AI; K-Means clustering; recommender systems

Check for updates

Academic Editor: Antony Bryant

Received: 19 June 2026
Revised: 8 September 2026
Accepted: 9 September 2026
Published: 15 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Social media platforms produce large volumes of user activity data, including demographic attributes, lifestyle behavior, engagement patterns, interaction behavior, and content interaction activities. Recommendation systems help convert this data into personalized content suggestions to enhance user experience and user engagement [1]. Traditional recommendation systems primarily utilize collaborative filtering and content-based filtering techniques to offer customized suggestions to users [2]. Nevertheless, current recommendation systems are still challenged by sparsity, scalability, behavioral diversity, and limited explainability [3,4]. The paper presents an explainable user-based collaborative recommendation framework for social media content recommendation. It integrates K-Means clustering, similarity-based collaborative filtering, neighbor voting, and generative

Large Language Model (LLM) explanations. The approach first applies K-Means clustering to group users with comparable behavioral characteristics and reduce the similarity search space. Behavioral similarity is then computed within each cluster using Pearson Similarity and Cosine Similarity [5]. The framework combines both similarity measures into a Combined Similarity score to enhance recommendation robustness, which is used in Top-N neighbor selection and collaborative recommendation generation.

Personalized recommendations for content themes and content types are subsequently generated using similarity-based neighbor voting among the most similar users. In addition, Top-K recommendation metrics, including Hit Rate@10, Precision@10, and Recall@10, are utilized to evaluate recommendation retrieval quality and recommendation effectiveness [6].

Finally, Gemini 2.5 Flash is integrated as an explainability layer to generate concise human-readable explanations for recommendation outputs. Recent studies highlighted that Large Language Models (LLMs) can improve recommendation transparency and user understanding by generating natural-language explanations for recommendation outputs [7,8].

Overall, the proposed framework aims to improve recommendation interpretability, transparency, and collaborative recommendation effectiveness in large-scale social media environments.

2. Problem Definition and Research Motivation

Many recommendation systems generate personalized recommendations without providing understandable explanations for why specific content themes or content types are recommended. This lack of explainability reduces recommendation transparency and makes it difficult for users to understand or trust recommendation decisions [4,9]. In social media environments, user behavior is highly multi-dimensional because it involves demographic attributes, lifestyle behavior, engagement patterns, and content interaction activities such as likes, comments, stories, and session behavior. Traditional recommendation approaches may struggle to effectively capture hidden behavioral similarities and complex interaction relationships between users [1].

In addition, users with similar engagement and interaction behavior may share similar content interests even when their profiles are not identical. Therefore, identifying behavioral similarity between users becomes an important challenge in personalized recommendation systems. Previous studies demonstrated that clustering techniques and similarity-based collaborative filtering can help identify users with related behavioral patterns more effectively [5,10].

The motivation behind this research arises from the growing need for recommendation systems that not only generate personalized recommendations, but also provide concise human-readable explanations for recommendation decisions. Improving recommendation interpretability and transparency may enhance user trust, recommendation acceptance, and overall user understanding in social media recommendation environments [9,11].

To address these challenges, this research proposes an explainable user-based collaborative recommendation framework that combines K-Means clustering, similarity computation, neighbor voting, and generative LLM explanations. First, demographic, lifestyle, and engagement-related features are preprocessed and normalized. K-Means clustering is then applied to group users with similar behavioral characteristics [12]. Within each cluster, Pearson Similarity and Cosine Similarity are computed to identify the Top-N most similar users based on behavioral engagement patterns [5]. To improve recommendation robustness, both similarity measures are integrated into a Combined Similarity score used during neighbor selection and recommendation generation.

Based on the preferences of neighboring users, the framework generates personalized recommendations for content themes and content types using similarity-based neighbor voting. Finally, Gemini 2.5 Flash is integrated as an explainability layer to generate concise natural-language explanations describing why recommendations were generated. Gemini 2.5 Flash was selected as the explainability layer due to its fast response time, efficient Python API integration, and low computational cost. It can quickly transform recommendation outputs into concise, human-readable natural-language explanations, improving recommendation transparency and user understanding. Recent explainable recommendation studies highlighted that LLM-generated explanations can improve recommendation transparency, interpretability, and user understanding [8,13].

The main objective of this research is to develop and evaluate an explainable personalized recommendation framework for social media content recommendation using clustering, similarity computation, and generative AI explanations. Specifically, this research aims to:

- Analyze demographic, lifestyle, and engagement-related features for behavioral user clustering and personalized recommendation generation.
- Design and implement a similarity-based collaborative recommendation framework using K-Means clustering, Pearson Similarity, Cosine Similarity, Combined Similarity, and Top-N neighbor voting.
- Evaluate recommendation retrieval performance using Top-K recommendation metrics including Hit Rate@10, Precision@10, and Recall@10.
- Integrate Gemini 2.5 Flash to generate concise human-readable explanations for recommendation outputs.

3. Literature Review

This section reviews previous studies related to recommendation systems, clustering techniques, similarity-based collaborative filtering, explainable recommendation systems, and generative LLM-based recommendation explanations. The review also identifies the main research gaps addressed in this study.

3.1. Traditional Recommendation Systems Overview

Traditional recommendation systems mainly rely on Collaborative Filtering (CF), Content-Based Filtering (CBF), and Hybrid Recommendation Systems to provide personalized recommendations to users [1]. Collaborative Filtering (CF) recommends items based on similarities between users or items using interaction history: ratings, likes, and clicks [2]. It is successful in discovering latent user preferences, but presents sparsity and cold-start problems when interaction data is sparse [3]. Content-Based Filtering (CBF) recommends items similar to those previously preferred by the user by analyzing item features like categories, keywords, and descriptions [2]. It offers personal recommendations that are more transparent, but at the same time could produce over-specialized results with less diversity [4]. Hybrid recommendation systems which also include content-based and collaborative recommendation methods are developed to enhance the accuracy of recommendations and to decrease the cold-start problem [2]. Such systems are currently widely utilized in e-commerce, streaming platforms, and applications in social media [1]. Although traditional recommendation systems exist, they suffer from sparsity, scalability, cold-start issues, and limited explainability. The majority of systems make recommendations without clearly explaining why specific items are suggested, which weakens the transparency and trust of users [4].

3.2. Clustering-Based Recommendation Systems

The clustering techniques have been employed in several studies to group users who share common behavior before recommending. When they provide recommendation systems based on clustering technology, this facilitates increasing scalability and simplifying the similarity computation at a large scale [14]. Among clustering algorithms, K-Means is a well-known and commonly used algorithm for recommendation systems since it is the simplest and most computationally efficient clustering algorithm [12]. K-Means clusters users based on similarities in preferences, behaviors, and patterns of interaction such that recommendation systems can give personalized recommendations in each cluster [10]. Clustering techniques have been used in e-commerce, education, and tourism recommendations, for instance, grouping users into clusters by purchasing behavior, learning behavior, tourism interest, etc., for enhancing the relevance and user engagement of recommendation-based algorithms [15,16]. As for user segmentation and behavioral clustering for filtering systems, clustering is a commonly used technique in collaborative filtering systems [10]. By doing so, recommendation efficiency can be increased while the computational complexity of searching all users from the dataset can be reduced as well [14]. Nevertheless, despite the success of clustering methods, much research remains limited due to cold-start problems, data sparsity, and clustering quality [10,12]. Moreover, based on the proposed K-Means algorithms, the classification process is limited and the classification efficiency is also disturbed for many algorithms, since it depends on the initial centroid, and it should be observed that the similarity or distance metric might be exclusive and this may lead to reduced clustering quality and recommendation performance [12]. Therefore, many clustering-based recommendation techniques are concentrated on recommendation accuracy and performance measures, while little concern is paid to the interpretability of recommendations.

3.3. Similarity-Based Collaborative Recommendation

Similarity-based collaborative recommendation methods recognize users or objects who prefer similar things, often using similarity metrics like Pearson Correlation and Cosine Similarity [5]. These techniques enable users to identify similar people with similar preferences within the collaborative filtering systems and therefore recommend them certain items [17]. In user–user collaborative filtering, the recommendation system computes similarity among users by their ratings and preferences [18]. The users with the greatest similarity scores are considered neighbors and their ratings are used to predict items for the target user [5]. Pearson Correlation is a widely used similarity measure among different similarity measures since it assesses the relationship among user rating patterns while considering rating differences between users [18]. Cosine Similarity quantifies how similar the users' or items' attributes are by vector similarity and is widely used in recommendation tasks [17]. Similarity-based recommendation systems also often employ neighbor-based recommendation protocols, in which the preferences of the closest neighbors are combined to make recommendation predictions [5]. This neighborhood selection criterion is crucial because recommendation quality relies on accurate recognition of users with similar interests and preferences [18]. The similarity measurement and cluster and k-nearest neighbor (KNN) methods were used in several studies for enhancement in the recommendation accuracy and similarity computation in a collaborative filtering system [19]. Since they can capture relationships between users and items better, these approaches are widely applied in movie recommendation settings and online environments [17]. Nonetheless, there is a risk of reduced recommendation robustness when a similarity measurement is preferred as different similarity measures correspond to different aspects of user preferences and rating behaviors [5]. Moreover, similarity-based collaborative filtering methods also suffer from

sparsity, cold-start, and limited rating data which might hamper recommendation quality in large-scale scenarios [18].

3.4. Neighbor Voting and Top-N Recommendation

Neighbor voting is a widely used mechanism in user-based collaborative filtering, where recommendations are generated based on the preferences of similar users. The recommendation process typically involves similarity computation, neighborhood selection, and rating prediction. After identifying the most similar users using methods such as the Top-K nearest neighbors, their ratings are aggregated to estimate the target user's preferences for unseen items [5].

Once neighborhood relationships are established, candidate items are ranked according to their predicted relevance, and the system generates a Top-N recommendation list containing the most relevant items. Neighborhood-based collaborative filtering models utilize learned user–user or item–item similarities to rank items and produce personalized recommendations [20]. The quality of Top-N recommendations is commonly evaluated using metrics such as Precision, Recall, and Hit Rate [6].

3.5. Explainable Recommendation Systems

Explainable Recommendation Systems (ERS) seek to generate transparency and user trust in the recommendation decisions by providing understandable explanations for recommendation outputs. Traditional recommender systems are typically black boxes, whereby users receive recommendations without understanding the reasons behind them. Explainable Artificial Intelligence (XAI) approaches aid this challenge by making recommendation decisions more understandable and interpretable to users and developers [21]. Explainability in recommender systems deals with explaining how input data and user preferences affect recommendation outputs [11]. Explainable AI techniques clarify how features contribute to recommendation decisions, resulting in improved transparency for users and improved understanding of overall recommendations. Two popular XAI techniques used in recommendation systems are Local Interpretable Model-Agnostic Explanations (LIMEs) and Shapley Additive Explanations (SHAPs). These methods reveal the rationale behind recommendations and determine the most influential features on recommendation outcomes. LIMEs and SHAPs improve transparency and explain recommendation outcomes and feature importance [11]. Natural-language explanations have also been used in explainable recommendation systems to provide human-readable recommendation justifications. Ref. [9] stated that explanations in conversational recommender systems may include recommendation reasons, item descriptions, and personal opinions to improve explanation quality and user understanding. The study also highlighted that natural-language explanations can improve transparency, persuasiveness, trustworthiness, and user satisfaction in recommendation systems.

User trust is considered one of the main objectives of explainable recommendation systems. Providing understandable explanations can increase users' confidence in recommendation systems and improve recommendation acceptance. Previous studies showed that transparency and explainability positively influence user trust, satisfaction, and transparency in recommendation systems [9].

However, many existing explainability approaches mainly focus on feature-level importance and technical interpretation methods. In addition, several explainable recommendation systems still provide limited human-readable natural-language explanations for recommendation decisions.

3.6. Explainable Generative LLMs in Recommendation Systems

Recent studies have explored the use of Large Language Models (LLMs) in recommendation systems to improve personalization, recommendation quality, and explanation generation. Advances in LLMs such as ChatGPT and Gemini introduced new capabilities including reasoning, natural-language understanding, in-context learning, and human-like text generation for recommendation applications [22]. LLMs have been increasingly applied in recommender systems because they can generate natural-language explanations that improve transparency and user understanding. Ref. [7] explained that LLMs can generate personalized and high-quality explanations for recommendation outputs using natural-language generation techniques. Their study also showed that LLM-generated explanations were positively perceived by users because they support recommendation understanding and decision-making. Recent research also highlighted that LLMs can extend existing recommender systems by generating post hoc explanations after recommendation items have already been selected [7]. This approach improves recommendation interpretability without replacing the original recommendation model. In addition, LLMs can transform recommendation outputs into human-readable explanations that improve transparency and user experience [8]. Several studies discussed the use of generative AI and LLMs for explanation generation in recommendation environments. Ref. [8] highlighted that LLM-generated explanations can improve transparency, trust, and user understanding by providing accessible natural-language recommendation justifications. Similarly, Ref. [23] demonstrated that factual contextual information, such as knowledge graphs, can improve the precision and comprehensibility of LLM-generated explanations while reducing hallucination risks. Recent explainable recommendation frameworks also demonstrated the use of LLMs as reasoning and explanation layers that transform structured recommendation evidence into transparent natural-language justifications [13], and proposed an explainable LLM-based recommendation framework that combines multimodal contextual information with structured prompting to generate human-readable recommendation reasoning and improve transparency and user trust.

However, existing LLM-based recommendation studies still face several limitations. Many approaches mainly focus on general recommendation capabilities and recommendation generation tasks rather than explainable recommendation generation [22]. In addition, LLM-generated explanations may suffer from hallucination, imprecise information, and high computational cost when explanations are generated without sufficient factual grounding [23].

In this paper, the Gemini 2.5 Flash layer is integrated as an explainability component to improve the interpretability of recommendation outputs. This layer does not replace the recommendation model. However, it takes in factual outputs from clustering, similarity calculation, and neighbor voting, and translates them into human-readable natural-language explanations for the ultimate recommendation results. This design is consistent with recent explainable LLM-powered recommendation frameworks, which use a structured prompting structure and transparent recommendation reasoning to increase interpretability and user trust [13].

3.7. Research Gap

Based on the reviewed literature, several research gaps can be identified. Conventional recommendation systems are predominantly based on accuracy of the recommendation with limitations in explainability and transparency [4]. Moreover, many clustering-based and similarity-based recommendation techniques focus mainly on single proximity measures and ignore the interpretability of recommendations [5,10]. However, previous research explored the use of LLMs for recommendation explanation generation; existing methods still offer limited human-readable explanations focusing on recommendation gen-

eration tasks only, leading to less explainable recommendation reasoning where clustering and collaborative filtering methods were integrated to generate a recommendation [22]. Therefore, this research proposes an explainable recommendation framework that combines K-Means clustering, similarity-based collaborative filtering, neighbor voting, and Gemini 2.5 Flash natural-language explanation generation to improve recommendation personalization, interpretability, transparency, and user understanding.

4. Methodology

4.1. Overall Framework Overview

This section presents the overall methodology of the proposed explainable recommendation framework for social media content recommendation. The methodology is based on a pipeline that generates recommendations from user-behavior data, searching for similar users, which is followed by recommendation generation and human text-based explanations for the recommendations. It builds upon these mechanisms by using user-based collaborative filtering, K-Means clustering, similarity calculation, Top-N neighbor selection, and generative AI explanation methods. The process starts by collecting demographics, lifestyle, and engagement features pertaining to social media users and preprocessing such features. Upon preprocessing, the users are further clustered into clusters according to behavioral similarity through K-Means clustering. The paper used Pearson Correlation and Cosine Similarity in each cluster to determine which users are most similar to one another. It produces users' personalized recommendations on content themes and content types from similarity-based neighbor voting. Finally, a generative explanation layer based on Gemini 2.5 Flash converts recommendation results into concise natural-language explanations to add interpretability and transparency for the recommendations. Figure 1 shows the overall methodology of the explainable recommendation framework for social media content recommendation. User clustering, similarity computation, collaborative recommendation generation, and generative AI explanations are integrated in the framework. These methods together are intended to produce personalized recommendations with human-readable interpretations.

4.2. Data Collection

The dataset used in this research was downloaded from the Social Media User Activity Dataset available on Kaggle and loaded as a CSV file within the Google Colab environment [24]. The dataset holds large-scale social media user activity data and behavioral profile variables recorded from Instagram-related usage patterns, for instance, demographic data, lifestyle features, users' engagement statistics, the engagement behavior of users with the content, session activity, likes, comments, stories, followers and other behavioral metrics regarding the social media usage. This dataset contains a total of 1,547,896 rows and 58 columns that allow for large and diverse data of user behavioral information, that can be used to perform clustering, similarity computation, and collaborative recommendation generation, as well as for the explainable recommendation analysis.

4.3. Data Preprocessing and Feature Selection

Several preprocessing operations are applied to prepare the dataset for clustering, similarity computation, and recommendation generation.

The preprocessing pipeline includes:

- Missing Value Handling: Numerical missing values are filled using the median, while categorical missing values are filled using the mode.
- Categorical Encoding: Categorical features are encoded using LabelEncoder to support clustering and similarity computation.

- Feature Selection: Demographic, lifestyle, and engagement-related features are selected for recommendation modeling.
- Feature Scaling: StandardScaler is applied to normalize numerical features before clustering and similarity analysis.
- Target Encoding: The target variables (preferred_theme and content_type) are encoded into numerical labels for recommendation evaluation.
- Dynamic Target Detection: The framework dynamically detects target columns using candidate matching and keyword-based identification to improve dataset flexibility.

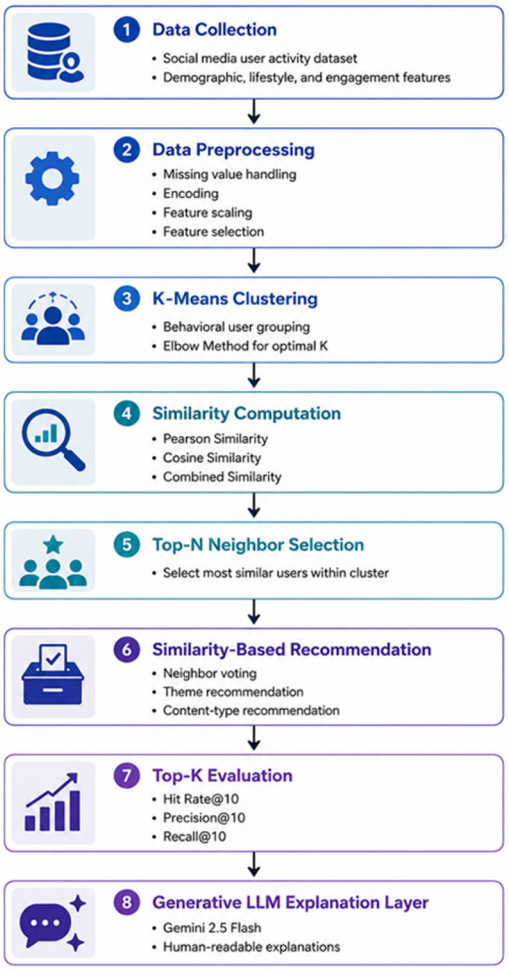


Figure 1. Overall methodology of the explainable recommendation framework. The authors used ChatGPT-5.5 (OpenAI) for diagram preparation.

In feature selection, 35 features were selected and grouped into three categories: demographic features (e.g., age, gender, country, education level, and income level), lifestyle features (e.g., exercise habits, sleep duration, stress level, happiness score, and social activities), and Instagram engagement-related features (e.g., daily active minutes, sessions per day, likes, comments, stories viewed, reels activity, followers, following behavior, and user engagement score). These features were selected because they capture users' personal characteristics, lifestyle preferences, and social media engagement patterns, which are important factors influencing content preferences and recommendation outcomes. To ensure consistency and improve model performance, categorical features were encoded using Label Encoding, while numerical features were standardized using StandardScaler before clustering and recommendation generation.

Only columns available in the dataset are selected for processing. StandardScaler is then applied to normalize numerical features using zero mean and unit variance normalization. This helps clustering and similarity-based methods treat features more consistently during distance and similarity computation.

4.4. K-Means Clustering

K-Means clustering is applied to group users with similar behavioral characteristics into clusters before recommendation generation. The clustering process minimizes the similarity search space by narrowing the number of neighbors to users who currently follow similar engagement and activity trends [25].

To determine the optimal number of clusters, the framework applies the Elbow Method by calculating K-Means inertia, also referred to as the within-cluster sum of squares (SSE), for multiple clustering configurations. For evaluating the consistency of clustering performance, the Elbow Method enables the comparison of the decrease in SSE based on the count of clusters and the calculation of the optimum count of clusters [25]. The elbow point is detected by KneeLocator and the preferred number of clusters is defined by the inertia curve. The Elbow Method analysis carried out to identify the optimal amount of clusters is shown in Figure 2. The solid blue line represents the inertia values for different values of K, and the blue dashed vertical line indicates the optimal number of clusters K = 5 identified by KneeLocator.

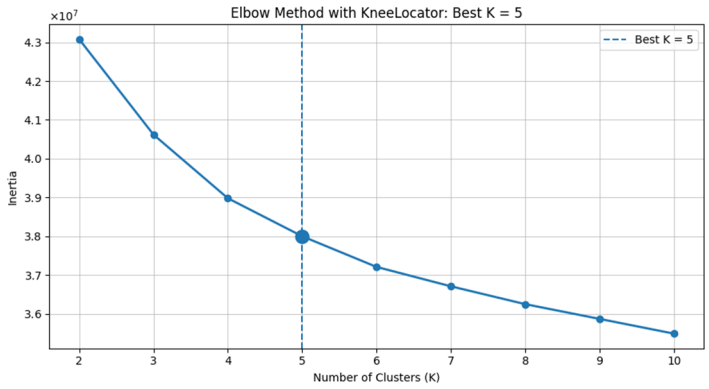


Figure 2. Elbow Method for optimal K-Means cluster selection.

Once the optimal K value is determined, the last K-Means model is trained on the entire training set. Each user is assigned a cluster label, and cluster sizes are analyzed to analyze how users are grouped among behavioral segments. Figure 3 illustrates the final training cluster distribution for the selected K-Means model with K = 5 clusters. The visualization shows how users are distributed across the behavioral clusters after the final clustering process.

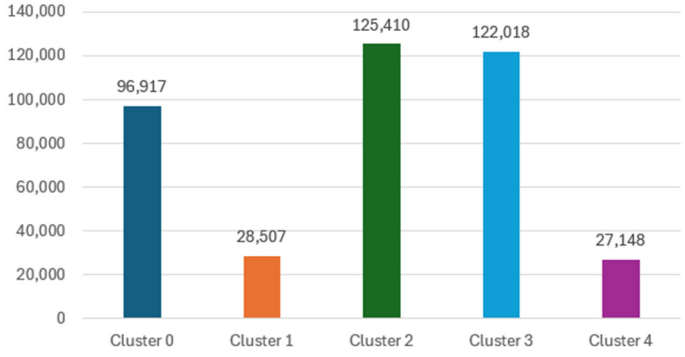


Figure 3. Training cluster distribution for the final K-Means model (K = 5).

4.5. Similarity-Based Recommendation

After clustering, similarity computation is performed within each cluster to identify users with similar behavioral patterns. The framework applies two similarity functions, namely Pearson Similarity and Cosine Similarity, to measure behavioral similarity between users based on their interaction patterns and rating behavior.

Pearson Similarity measures the linear relationship between users after mean-centering their rating vectors, helping to capture similarities in behavioral trends and interaction patterns [5].

$$PCC(u,v) = \frac{\sum_{f=1}^{m}\left(x_{uf} - \overline{x}_u\right)\left(x_{vf} - \overline{x}_v\right)}{\sqrt{\sum_{f=1}^{m}\left(x_{uf} - \overline{x}_u\right)^2}\sqrt{\sum_{f=1}^{m}\left(x_{vf} - \overline{x}_v\right)^2}} \quad (1)$$

As shown in Pearson Similarity (1), where m represents the total number of behavioral features, x_{uf} and x_{vf} denote the normalized values of feature f for users u and v, respectively, while $\overline{x}_u$ and $\overline{x}_v$ represent the mean values of their behavioral feature vectors [5].

Cosine Similarity measures the similarity between user vectors based on vector orientation and evaluates how closely users behave relative to each other regardless of magnitude differences [5].

$$\text{Cosine}(u,v) = \frac{\sum_{i \in I_{uv}} r_{ui}\, r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2}\sqrt{\sum_{i \in I_v} r_{vi}^2}} \quad (2)$$

In Equation (2), I_{uv} represents the set of commonly rated items between users u and v, while r_{ui} and r_{vi} denote the ratings assigned by users u and v to item i [5].

With these measures of similarity, the framework provides a unified similarity score to improve recommendation robustness. The final similarity value is computed by averaging Pearson Similarity and Cosine Similarity, resulting in a Combined Similarity score suitable for Top-N neighbor selection.

$$\text{Combined Similarity} = \frac{\text{Pearson} + \text{Cosine}}{2} \quad (3)$$

As shown in (3), the Combined Similarity score is the average of both Pearson Similarity and Cosine Similarity. This Combined measure allows us to balance behavioral relationship patterns and vector-based similarity of users. By integrating both similarity functions, the framework reduces the limitations of relying on a single similarity metric and improves the robustness of neighbor selection during recommendation generation. The Combined Similarity score is then used to identify the Top-N most similar users within each cluster for collaborative recommendation voting.

4.6. Top-K Recommendation Evaluation

The proposed framework analyzed the quality of recommendation retrieval using Top-K recommendation metrics such as Hit Rate@10, Precision@10, and Recall@10 [6]. These metrics are commonly used in recommender systems evaluation to analyze the effectiveness of recommendation ranking and retrieval performance.

Hit Rate@10 measures whether at least one relevant recommendation appears within the Top-10 recommended results [6].

$$HitRate@k(u) = I[\mid rel(u) \cap rec_k(u) \mid > 0] \quad (4)$$

In Equation (4), where $rel(u)$ represents the set of relevant items for user u, $rec_k(u)$ denotes the Top-K recommendation list generated for the user, and $I[\cdot]$ is an indicator

function that returns 1 if at least one relevant item exists within the recommendation list and 0 otherwise [6].

Precision@10 evaluates the proportion of relevant recommendations among the Top-10 retrieved recommendation results [6].

$$Precision@k(u) = \frac{| rel(u) \cap rec_k(u) |}{k} \quad (5)$$

As shown in (5), $rel(u)$ represents the set of relevant items for user u, while $rec_k(u)$ denotes the Top-K recommendation list generated for the user [6].

Recall@10 measures the proportion of relevant items successfully retrieved within the Top-K recommendation results [6].

$$Recall@k(u) = \frac{| rel(u) \cap rec_k(u) |}{| rel(u) |} \quad (6)$$

In Equation (6), $rel(u)$ represents the set of relevant items associated with user u, while $rec_k(u)$ denotes the generated Top-K recommendation list [6].

4.7. Recommendation Generation

For a selected query user, the framework identifies users within the same K-Means cluster and computes similarity scores using Pearson Similarity, Cosine Similarity, and the Combined Similarity measure.

Users are then ranked based on the Combined Similarity score, and the Top-N most similar users are selected for collaborative recommendation generation. If the cluster contains fewer than the required number of neighbors, the framework implements a fallback mechanism that searches across all remaining users to maintain recommendation stability. The final recommended content theme and content type are determined using majority voting among the Top-N similar users. This recommendation system suggests personalized recommendations for the query user based on the behavioral preferences of similar users. Table 1 presents an example of the Top-N similar users identified for a selected query user using Pearson Similarity, Cosine Similarity, and the Combined Similarity measure.

Table 1. Example of Top-N similar users and recommendation generation.

Rank	Cluster	Pearson Similarity	Cosine Similarity	Combined Similarity	Preferred Theme	Content Type
1	3	0.9228	0.9438	0.9333	Food	Videos
2	3	0.9190	0.9382	0.9286	Food	Videos
3	3	0.9186	0.9377	0.9281	Music	Photos
4	3	0.9226	0.9327	0.9277	Travel	Live
5	3	0.9148	0.9382	0.9265	Fitness	Videos

4.8. Generative LLM Explanation Layer

The Gemini 2.5 Flash layer is implemented in order to improve the interpretability of the recommendation outputs. This layer does not replace the recommendation model; instead, it receives factual outputs generated by the model, including the recommended theme, recommended content type, neighbor vote distribution, shared traits, and cluster signature. The explanation is split into two layers. In the first stage, a deterministic factual extractor extracts recommendation facts via clustering, similarity computation, and neighbor voting. In addition, a second stage, Gemini 2.5 Flash, transforms these facts into two simple sentences in plain English explaining why the recommendation was produced. A disk-based cache is used to avoid repeated API calls for the same cluster, theme, and content-type combination, improving efficiency during large-scale explanation generation.

The prompt provided to Gemini 2.5 Flash contains factual recommendation outputs generated by the recommendation framework, including:

- Recommended content theme;
- Recommended content type;
- Neighbor vote distribution;
- Shared behavioral traits;
- Cluster signature.

The model is then instructed to generate exactly two short natural-language sentences explaining why the recommendation was produced for the user. An example prompt structure is shown in Figure 4.

```
Recommended content theme: Food

Recommended content type: Videos

Among similar users, the theme votes are: {'Food': 3,
'Travel': 1}

You share these traits with your most similar users:
engagement_score, video_interaction

Your cluster signature is: high video engagement, active
interaction behavior

Now write exactly two short sentences explaining to the
user why this theme and content type are recommended.
```

Figure 4. Example prompt structure for Gemini 2.5 Flash.

5. Experimental Setup

Environment Setup

The proposed recommendation framework was implemented within the Google Colab environment using several Python libraries for data preprocessing, clustering, similarity analysis, visualization, and explainable recommendation generation. Pandas and NumPy were used for loading, processing, and manipulating large-scale social media user activity data. Matplotlib (version 3.10.0) was used to generate visualizations and similarity analysis figures throughout the experimental process. Scikit-learn utilities were applied for preprocessing, feature normalization, categorical encoding, and K-Means clustering. The framework used K-Means clustering to cluster users according to their behavioral engagement patterns. Behaviorally similar users were identified within each cluster through Pearson Similarity and Cosine Similarity. Subsequently, neighbor voting was applied to produce personalized content recommendations. For recommendation testing, Top-K recommendation metrics such as Hit Rate@K, Precision@K, and Recall@K were used by the framework to measure recommendation retrieval quality and similarity effectiveness. Furthermore, the explainability layer combined Gemini 2.5 Flash with the Google-GenAI package, creating concise natural-language explanations for the recommendation results. Environment configuration, API variable management, and caching mechanisms were also included for reproducibility and efficient explanation generation.

6. Results

This section presents the results of the proposed explainable recommendation framework. The evaluation focuses on similarity analysis, recommendation performance, and LLM-generated explanations. The results are organized into three parts: Similarity Results, Top-K Recommendation Results, and LLM Explanation Results to assess the effectiveness of the framework in generating personalized and explainable recommendations.

6.1. Similarity Results

The framework performed similarity analysis and Top-K recommendation metrics to appraise the quality of behavioral similarity and recommendation effectiveness of the proposed recommendation framework based on its collaborative recommendation process. The evaluation process measured:

- Mean Pearson Similarity;
- Mean Cosine Similarity;
- Mean Combined Similarity.

The evaluation results showed relatively high similarity values across neighboring users. The Mean Pearson Similarity score was 0.8044, the Mean Cosine Similarity 0.8113, and the Mean Combined Similarity 0.8078, representing strong behavioral similarity between neighboring users. These high similarity values indicate that the clustering and similarity computation stages successfully grouped users with similar engagement and interaction patterns.

Figure 5 presents the average similarity scores computed using Pearson Similarity, Cosine Similarity, and the Combined Similarity measure for Top-10 neighbor selection. Results indicate that the similarity measures yielded relatively high similarity scores, confirming the fact that users clustered together have strong behavioral relationships. The authors find an average similarity score that is best performing for Cosine Similarity, then Combined Similarity, and finally Pearson Similarity measures. The similarity analysis shows the same engagement and interaction patterns among the cluster members, which confirms the effectiveness of clustering prior to collaborative recommendation generation. Furthermore, the Combined Similarity method presents a balanced similarity representation based on Pearson and Cosine Similarity. This enhances the robustness of neighbor selection and fosters more stable personalized recommendation generation in the approach presented herein. The evaluation result suggests that the proposed framework successfully found users with related behavioral engagement patterns by clustering and similarity computations.

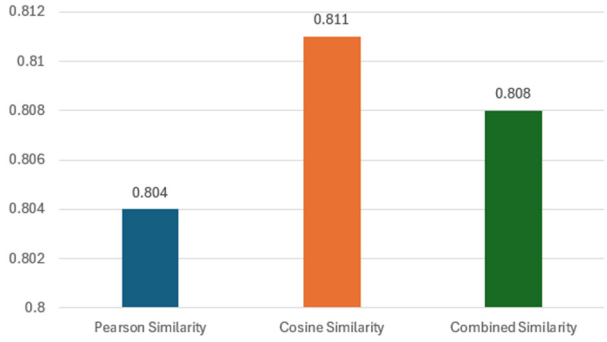


Figure 5. Average similarity scores for Top-10 neighbor selection.

6.2. Top-K Results

Top-K recommendation evaluation was performed using K = 10 to measure recommendation retrieval quality within the proposed collaborative recommendation framework. Results produced Hit Rate@10, Precision@10, and Recall@10 for both preferred-theme and content-type recommendation tasks. Hit Rate@10 determines whether relevant recommendations appear in the Top-10 recommended results. Precision@10 compares the proportion of relevant recommendations in the Top-10 identified recommendation candidates. Since every user has one target preference, Recall@10 is directly equivalent to Hit Rate@10. The results reveal better recommendation performance for content-type recommendation than preferred-theme recommendation, where user engagement behaviors are associated more with content consumption formats than broader thematic interests.

Figure 6 shows Top-K recommendation evaluation results for preferred-theme recommendation and content-type recommendation tasks, represented in terms of Hit Rate@10, Precision@10, and Recall@10. The results suggest that Top-K recommendation retrieval is relatively high, particularly for content-type recommendation. The high Hit Rate@10 and Recall@10 values indicate that relevant recommendations frequently appeared within the Top-10 recommendation results, verifying the effectiveness of clustering and similarity-based neighbor selection. However, Precision@10 is still low for both recommendation tasks. This result is expected in large-scale multi-class social media environments where users may interact with multiple content categories and behavioral preferences simultaneously. While the framework successfully retrieved behaviorally related recommendation candidates, not all retrieved recommendations exactly matched the user's actual preference. Furthermore, on all Top-K metrics, content-type recommendation was consistently superior to preferred-theme recommendation. This is further validated by the fact that user engagement behavior is more associated with content consumption formats, such as videos, reels, and stories, than with broader thematic interests. Overall, the Top-K evaluation results show that the presented similarity-based collaborative recommendation framework was effective in retrieving behaviorally relevant recommendation candidates via clustering and similarity-based neighbor selection. Consistently, this shows high similarity scores among users within the same behavioral cluster, reflecting behavioral consistency between engagement and interaction patterns. Majority voting among the Top-N similar users selected "Food" as the recommended content theme and "Videos" as the recommended content type. N = 5 was used for Top-N neighbor selection in the recommendation demonstration. Additional Top-10 recommendation metrics were later evaluated during recommendation performance analysis. Pearson Similarity, Cosine Similarity, and Combined Similarity scores for the Top-N similar users selected for recommendation generation are presented in Figure 7.

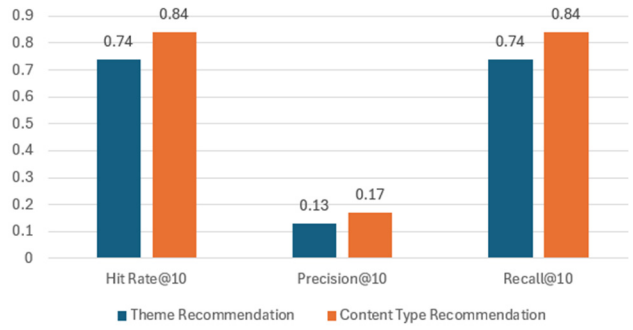


Figure 6. Top-K recommendation evaluation results.

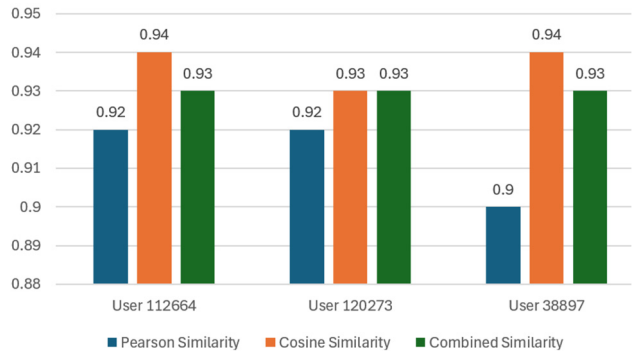


Figure 7. Similarity score comparison for Top-N similar users.

In Figure 7, the visualization shows that there are consistently high similarity ratings between all the selected neighbors, meaning that there is high behavioral similarity between

the query user and its neighboring users of the same cluster. Cosine Similarity produced slightly higher similarity scores compared to Pearson Similarity, while the Combined Similarity measure provided balanced similarity values by integrating both metrics during neighbor selection.

6.3. LLM Explanation Results

The framework internally provides structured behavioral recommendation facts to Gemini 2.5 Flash, including similarity patterns, neighbor voting, shared traits, and cluster characteristics. However, the final explanation presented to the user removes technical details and transforms the recommendation into short, human-readable natural-language explanations to improve interpretability and user understanding. Table 2 shows a sample of recommendation explanations generated by Gemini 2.5 Flash.

Table 2. Sample Gemini 2.5 Flash recommendation explanations.

Recommended Theme	Content Type	Gemini 2.5 Explanation
Art	Reels	Similar users also preferred Art reels and showed similar reel engagement behavior.
Tech	Stories	Similar users voted for Tech stories and shared similar messaging behavior.
Travel	Videos	Similar users frequently interacted with Travel videos and showed high engagement activity.

7. Discussion

This section discusses the effectiveness of the proposed explainable recommendation framework based on similarity analysis, recommendation retrieval performance, and recommendation interpretability. The discussion focuses on the behavioral consistency between clustered users, the effectiveness of similarity-based recommendation generation, and the role of generative LLM explanations in improving recommendation transparency and user understanding.

7.1. Similarity Analysis

High similarity scores indicate that clustering and similarity computation were effective in grouping users with similar engagement and interaction patterns. The consistently strong Pearson Similarity, Cosine Similarity, and Combined Similarity values suggest that neighboring users within the same cluster shared similar behavioral characteristics. Among the evaluated similarity measures, Cosine Similarity produced slightly higher similarity values, indicating strong consistency in user interaction patterns regardless of magnitude differences. Conversely, the Combined Similarity approach achieved a more balanced representation of user similarity by incorporating both Pearson Similarity and Cosine Similarity for neighbor selection. The results also indicate that clustering prior to similarity computation mitigated the behavioral variability between neighboring users and encouraged more uniformity in the collaborative recommendation generation. Therefore, the similarity analysis showed that this framework correctly identified users with similar behavioral engagement patterns.

7.2. Recommendation Performance Analysis

The evaluation results showed that content-type recommendation achieved better performance than preferred-theme recommendation across all Top-K evaluation metrics. This suggests that user engagement behavior is more strongly associated with content

consumption formats, such as videos, reels, stories, and photos, than with broader thematic interests such as food, travel, or fitness.

The relatively high Hit Rate@10 and Recall@10 values indicate that relevant recommendations frequently appeared within the Top-10 recommendation results. These findings demonstrate the effectiveness of clustering and similarity-based neighbor selection in retrieving behaviorally related recommendation candidates.

However, Precision@10 remained relatively low for both recommendation tasks. This result is expected in large-scale multi-class social media environments where users often interact with multiple content categories and behavioral preferences simultaneously. As a result, users with similar engagement behavior may still prefer different content themes despite consuming similar content formats.

The results also highlight the complexity of recommendation generation in dynamic social media environments, where behavioral diversity and continuously changing user interests increase recommendation difficulty. Overall, the evaluation results demonstrate that the framework was effective in retrieving behaviorally relevant recommendation candidates through clustering and similarity-based neighbor selection.

7.3. Explainability and Interpretability Analysis

The Gemini 2.5 Flash explanation layer improved the interpretability of the recommendation framework by transforming recommendation outputs into concise human-readable explanations. The generated explanations helped provide understandable recommendation reasoning based on clustering, similarity computation, and neighbor voting results.

Unlike traditional black-box recommendation systems, the framework provided transparent recommendation explanations that help users understand why specific content themes and content types were recommended. This improves recommendation transparency and supports better understanding of recommendation decisions.

In addition, concise natural-language explanations may improve user trust and recommendation acceptance by helping users understand the relationship between their behavioral patterns and the generated recommendations. The explainability layer therefore contributes to improving interpretability and usability in social media recommendation environments.

The proposed framework contributes an explainable recommendation pipeline that integrates clustering, similarity computation, Top-N recommendation generation, evaluation metrics, and generative LLM explanations. The Gemini 2.5 Flash explanation layer improved recommendation interpretability by generating concise human-readable explanations for recommendation outputs, supporting recommendation transparency and user understanding in social media recommendation environments.

8. Conclusions

This paper proposed an explainable user-based collaborative recommendation framework using K-Means clustering, similarity-based collaborative filtering, neighbor voting, and Gemini 2.5 Flash explanations.

The evaluation revealed high levels of behavioral similarity between neighboring users, Mean Pearson Similarity = 0.8044, Mean Cosine Similarity = 0.8113, and Mean Combined Similarity = 0.8078, indicating a good clustering and similarity calculation ability to find related user engagement behavior. Similarly, content-type recommendation performed even better than preferred-theme recommendation, indicating that user engagement behavior leans more towards content consumption formats rather than broader thematic interests. Despite the effectiveness of the proposed framework, several limitations remain. Recommendation quality is highly dependent on the diversity of data and variability in user behavior at scale across large-scale social media environments. In addition, K-Means

clustering primarily captures linear grouping behavior and may not fully represent complex nonlinear relationships between users. The current framework also does not support real-time recommendation updates for continuously changing user interaction behavior. Future work might employ explainable AI models, including SHAPs and LIMEs, for feature-level interpretation, deep learning and transformer-based recommendation algorithms, which may contribute to the robustness of recommendation and the personalization of recommendation quality. In addition, BLEU, ROUGE, and semantic similarity measures, as well as real user feedback loops, could also be integrated into future work to assess the explanation performance, increasing the adaptability of recommendation and personalization quality for the enhancement of the recommendation.

Author Contributions: Conceptualization, A.A., E.A. and D.A.; methodology, A.A., E.A. and D.A.; software, A.A., E.A. and D.A.; validation, A.A., E.A. and D.A.; formal analysis, A.A., E.A. and D.A.; investigation, A.A., E.A. and D.A.; resources, A.A., E.A. and D.A.; data curation, A.A., E.A. and D.A.; writing—original draft preparation, A.A., E.A. and D.A.; writing—review and editing, A.A., E.A. and D.A.; visualization, A.A., E.A. and D.A.; supervision, D.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are derived from public domain resources; the data presented in this study are publicly available from Kaggle. The dataset used is the Social Media User Activity Dataset, available at: https://www.kaggle.com/datasets/sadiajavedd/social-media-user-activity-dataset (accessed on 16 June 2026). These data were derived from public domain resources and were used for the analyses presented in this study.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT-5.5 (OpenAI) for language editing, manuscript organization, and diagram preparation. Additionally, the Google Gemini API (gemini-2.5-flash) was used within a two-stage framework to translate extracted data facts into concise, two-sentence recommendation explanations. The authors reviewed and edited all generated content and take full responsibility for the publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Raza, S.; Rahman, M.; Kamawal, S.; Toroghi, A.; Raval, A.; Navah, F.; Kazemeini, A. A comprehensive review of recommender systems: Transitioning from theory to practice. *Comput. Sci. Rev.* **2026**, *59*, 100849. [CrossRef]
2. Aggarwal, C.C. *Recommender Systems: The Textbook*, 1st ed.; Springer: Cham, Switzerland, 2016. [CrossRef]
3. Al-Hasan, T.M.; Sayed, A.N.; Bensaali, F.; Himeur, Y.; Varlamis, I.; Dimitrakopoulos, G. From traditional recommender systems to GPT-based chatbots: A survey of recent developments and future directions. *Big Data Cogn. Comput.* **2024**, *8*, 36. [CrossRef]
4. Hasan, E.; Rahman, M.; Ding, C.; Huang, J.X.; Raza, S. Review-based recommender systems: A survey of approaches, challenges and future perspectives. *ACM Comput. Surv.* **2025**, *58*, 3742421. [CrossRef]
5. Fkih, F. Similarity measures for collaborative filtering-based recommender systems: Review and experimental comparison. *J. King Saud Univ.–Comput. Inf. Sci.* **2022**, *34*, 7645–7669. [CrossRef]
6. Tamm, Y.-M.; Damdinov, R.; Vasilev, A. Quality metrics in recommender systems: Do we calculate metrics consistently? In *Proceedings of the 15th ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September–1 October 2021*; ACM: New York, NY, USA, 2021; pp. 708–713. [CrossRef]
7. Lubos, S.; Tran, T.N.T.; Felfernig, A.; Erdeniz, S.P.; Le, V.-M. LLM-generated explanations for recommender systems. In *Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization Adjunct Proceedings, Cagliari, Italy, 1–4 July 2024*; ACM: New York, NY, USA, 2024; pp. 276–285. [CrossRef]
8. Said, A. On explaining recommendations with Large Language Models: A review. *Front. Big Data* **2025**, *7*, 1505284. [CrossRef] [PubMed]

9. Guo, S.; Zhang, S.; Sun, W.; Ren, P.; Chen, Z.; Ren, Z. Towards explainable conversational recommender systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023*; ACM: New York, NY, USA, 2023; pp. 2786–2795. [CrossRef]
10. Gulzar, Y.; Alwan, A.A.; Abdullah, R.M.; Abualkishik, A.Z.; Oumrani, M. OCA: Ordered clustering-based algorithm for e-commerce recommendation system. *Sustainability* **2023**, *15*, 2947. [CrossRef]
11. Vultureanu-Albiși, A.; Murareț u, I.; Bădică, C. Explainable recommender systems through reinforcement learning and knowledge distillation on knowledge graphs. *Information* **2025**, *16*, 282. [CrossRef]
12. Zhang, J.; Zhang, L. Clustering and personalized course recommendation based on improved K-means algorithm for student learning behavior data. In *Proceedings of the ICEKIM, Cambridge, UK, 20 June 2025*; ACM: New York, NY, USA, 2025. [CrossRef]
13. Kim, T.; Seo, J.; Ko, H.; Han, K.; Lee, W. TipMate: An explainable fair tip recommender system based on LLM. *IEEE Access* **2026**, *14*, 66241–66254. [CrossRef]
14. Al-Asadi, A.; Jasim, M.N. Clustering-based collaborative recommender system using a nature-inspired algorithm. In *Proceedings of the IEEE, Padua, Italy, 18–23 July 2022*; IEEE: Piscataway, NJ, USA, 2022. [CrossRef]
15. Kumar, M.R.; Vishnu, S.; Roshen, G.; Kumar, D.N.; Revathi, P.; Baster, D.R.L. Product recommendation using collaborative filtering and K-means clustering. In *Proceedings of IEEE, Yokohama, Japan, 13–17 May 2024*; IEEE: Piscataway, NJ, USA, 2024. [CrossRef]
16. Zhang, X.; Wang, S. K-means clustering-based recommendation system for enhancing village folk tourism. In *Proceedings of the IEEE, Atlanta, GA, USA, 3–9 May 2025*; IEEE: Piscataway, NJ, USA, 2025. [CrossRef]
17. Raghavendra, C.K.; Srikantaiah, K.C. Similarity based collaborative filtering model for movie recommendation systems. In *Proceedings of the IEEE, Xi'an, China, 30 May–5 June 2021*; IEEE: Piscataway, NJ, USA, 2025. [CrossRef]
18. Al-bashiri, H.; Abdulgabber, M.A.; Romli, A.; Hujainah, F. Collaborative filtering similarity measures: Revisiting. In *Proceedings of the ACM, Mountain View, CA, USA, 23–27 October 2017*; ACM: New York, NY, USA, 2017. [CrossRef]
19. Wang, L.; Mistry, S.; Hasan, A.A.; Hassan, A.O.; Islam, Y.; Osei, F.A.J. Implementation of a collaborative recommendation system based on multi-clustering. *Mathematics* **2023**, *11*, 1346. [CrossRef]
20. Eleftherakis, S.; Koutrika, G.; Amer-Yahia, S. Optimizing Neighborhoods for Fair Top-N Recommendation. In Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization (UMAP 2024), Cagliari, Italy, 22 July 2024; pp. 57–66. [CrossRef]
21. Vultureanu-Albiși, A.; Bădică, C. *Recommender Systems: An Explainable AI Perspective*; IEEE: Piscataway, NJ, USA, 2021.
22. Deldjoo, Y.; He, Z.; McAuley, J.; Korikov, A.; Sanner, S.; Ramisa, A.; Vidal, R.; Sathiamoorthy, M.; Kasirzadeh, A.; Milano, S. A review of modern recommender systems using generative models (Gen-RecSys). In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Barcelona, Spain, 25–29 August 2024*; ACM: New York, NY, USA, 2024; pp. 6448–6458. [CrossRef]
23. Abu-Rasheed, H.; Weber, C.; Fathi, M. Knowledge graphs as context sources for LLM-based explanations of learning recommendations. In *Proceedings of the IEEE Global Engineering Education Conference (EDUCON), Kos Island, Greece, 8–11 May 2024*; IEEE: Piscataway, NJ, USA, 2024. [CrossRef]
24. Javed, S. Social Media User Activity Dataset. Kaggle. 2025. Available online: https://www.kaggle.com/datasets/sadiajavedd/social-media-user-activity-dataset/data (accessed on 16 June 2026).
25. Umargono, E.; Suseno, J.E.; Gunawan, V.S.K. K-Means clustering optimization using the Elbow Method and early centroid determination based on mean and median. In *Proceedings of the International Conference on Information System and Technology (CONRIST 2019), Yogyakarta, Indonesia, 5–6 December 2019*; SCITEPRESS: Setúbal, Portugal, 2020; pp. 234–240. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.