computers

MDPI

Article

Harnessing the Power of User-Centric Artificial Intelligence: Customized Recommendations and Personalization in Hybrid Recommender Systems

Christos Troussas *, Akrivi Krouska , Antonios Koliarakis and Cleo Sgouropoulou

Department of Informatics and Computer Engineering, University of West Attica, 12243 Egaleo, Greece; akrouska@uniwa.gr (A.K.); csgouro@uniwa.gr (C.S.)
* Correspondence: ctrouss@uniwa.gr

Abstract: Recommender systems are widely used in various fields, such as e-commerce, entertainment, and education, to provide personalized recommendations to users based on their preferences and/or behavior. This paper presents a novel approach to providing customized recommendations with the use of user-centric artificial intelligence. In greater detail, we introduce an enhanced collaborative filtering (CF) approach in order to develop hybrid recommender systems that personalize search results for users. The proposed CF enhancement incorporates user actions beyond explicit ratings to collect data and alleviate the issue of sparse data, resulting in high-quality recommendations. As a testbed for our research, a web-based digital library, incorporating the proposed algorithm, has been developed. Examples of operation of the use of the system are presented using cognitive walkthrough inspection, which demonstrates the effectiveness of the approach in producing personalized recommendations and improving user experience. Thus, the hybrid recommender system, which is incorporated in the digital library, has been evaluated, yielding promising results.

Keywords: personalization; recommender systems; hybrid recommender systems; collaborative filtering; digital library

check for updates

Citation: Troussas, C.; Krouska, A.; Koliarakis, A.; Sgouropoulou, C. Harnessing the Power of User-Centric Artificial Intelligence: Customized Recommendations and Personalization in Hybrid Recommender Systems. *Computers* **2023**, *12*, 109. https://doi.org/10.3390/computers12050109

Academic Editor: Paolo Bellavista

Received: 7 May 2023
Revised: 16 May 2023
Accepted: 17 May 2023
Published: 22 May 2023

1. Introduction

User-centric artificial intelligence (AI) has emerged as a viable strategy for developing intelligent systems that can better fulfill users' demands and preferences [1–3]. In contrast to traditional AI systems, which are designed to optimize a certain purpose or task, user-centric AI systems are concerned with offering tailored services that are suited to each user's distinctive traits and aspirations. This entails creating algorithms and approaches that can learn from a user's interactions with the system as well as their input and preferences and then adjust and improve continuously. The user-centric approach offers the opportunity to increase AI systems' efficiency and effectiveness; improve the overall user experience; and tackle ethical and societal concerns about the influence of AI on people and society. Furthermore, user-centric AI can encourage higher levels of confidence, openness, and interaction between users and intelligent systems, which are critical factors for AI technology's acceptance and success. As a result, there is a rising demand for user-centric AI research and development to increase our understanding of how to develop and implement intelligent systems that can efficiently meet the needs and preferences of diverse users [4].

In addition, as the software industry develops and expands, the demand for software systems that can adapt to individual user demands and interests also grows. Several factors are driving this need for personalization, including the large volumes of data that are currently available, users' increased expectations for personalized experiences, and the potential benefits of providing individualized software solutions [5–8]. However, effective software personalization necessitates the negotiation of considerable hurdles, such as maintaining huge and complicated datasets and building efficient and scalable algorithms.

One of the most prevalent examples of personalization in software is its use in recommender systems. Recommender systems [9–14] are programs that offer users customized suggestions based on their previous interests and activity. Recommender systems can assist users in discovering new services, goods, or content that they are likely to consider interesting or relevant by employing personalization strategies, while additionally enhancing the overall user experience. Personalization is especially crucial in recommender systems because it allows the systems to efficiently cope with the so-called "cold start" problem, which occurs when limited or no data about a new person, product, or setting are available. Recommender systems can effectively solve the cold start problem and deliver relevant recommendations, even in circumstances in which conventional methods fail, by constructing accurate and flexible user models.

There are a great variety of algorithms and techniques available in the field of machine learning, of which the domain of designing a recommender system has access to a diverse set. Collaborative Filtering (CF) [15,16] is one such technique that involves collecting data on numerous peers' preferences in order to make predictions about users' preferences automatically. The CF principle argues that people who have the same view about one subject are more likely to have the same attitude about another subject than any other random user. CF operates by comparing users' views about various items, where an item might be anything from a piece of music to a film or a tangible object. A metric, such as a rating system, time spent observing an item, or a like–dislike system, is used to measure the users' opinions. The CF algorithm then compares these views and finds users who agree with them. The CF algorithm predicts a user's opinions about items for which the user has not yet formulated an opinion but similar users have. The user is then recommended items corresponding to the best predicted opinion. However, one disadvantage of CF algorithms is that they may produce low-quality suggestions when dealing with sparse data, which is also referred to as the "cold-start problem", as stated above. To address this problem, a hybrid method [17] is advised, which entails including certain elements of content-based filtering into the recommendation development process for the initial user profile.

This paper introduces a hybrid recommender system integrated into a web-based digital lending library system. Using the system, registered users can browse books, reserve them, rate them, and mark them as favorites. The hybrid recommender system primarily employs CF but with modifications to the data collection procedure for user preferences and a customized algorithm that incorporates content-based filtering when data are scarce. These modifications make use of all user actions in the system instead of solely explicit ratings. A sparsity of data can occur when a few users rate items infrequently or abstain from rating altogether, resulting in poor recommendations. The recommender system can considerably increase recommendation quality and overcome the sparse data issue by gathering more data from user behaviors.

The novelty of this paper lies in the specific adjustments made to the collaborative filtering algorithm, which can increase its performance in the setting of digital libraries and beyond. Furthermore, this paper can contribute to the broader field of recommendation systems by demonstrating the effectiveness of hybrid recommender systems that combine enhanced recommendation strategies.

2. Related Work

After examining the relevant literature, three main areas of research emerged: (1) the incorporation of data collection to enhance the quality of data, (2) modifying the collaborative filtering algorithm to better utilize available data, and (3) hybridizing the recommender system by incorporating additional techniques. These three research areas share the goal of improving the overall performance and recommendation quality of the recommender system, both in general and in specific contexts.

Concerning the first area, a group of researchers proposed the use of sequential pattern analysis (SPA) to extract additional data about user preferences and combined these with explicit data to generate a final predicted preference. This approach was proposed in [18]. Another method designed as a restaurant recommendation system involved using a dragonfly hybrid that categorizes users based on their personalities. This approach reduced the sparsity problem and the number of comparisons that are made in Collaborative Filtering (CF). This method uses a combination of swarm intelligence and statistical techniques to select the optimal neighbors [19]. Chen L. et al. [20] proposed a data-debugging framework to recognize overly personalized ratings that negatively affect the performance of a given CF model. They divide the ratings into multiple groups and apply a method comparable to cross-validation to combine the debugging outcomes for each group to determine the final maverick ratings. In [21], the authors examined the effects of implicit feedback on prediction performance and utilized implicit user relationships, item records, and users' positive attitudes in a movie recommendation system. They proposed a multiplex implicit feedback system that factorizes both the explicit rating matrix and the implicit attitude matrix, resulting in highly positive outcomes.

Concerning the second area, in [22], the authors propose a novel similarity model that addresses the limitations of current similarity measures. Their proposed method employs the average and variance of ratings to represent a user's rating preferences, and this approach shows improved performance compared to existing methods. In [23], the authors suggested a hybrid method that combines item-based collaborative filtering (CF) with Case-Based Reasoning (CBR) and a Self-Organizing Map (SOM) optimized with a Genetic Algorithm (GA) to achieve more personalized product recommendations while overcoming some of the challenges of traditional item-based CF. To address sparsity in the dataset, the method uses CBR combined with average filling. For large datasets, an SOM optimized with a GA is used to cluster users and reduce the scope of item-based CF. The proposed method yields promising results when compared to traditional item-based CF algorithms. Finally, another study [24] introduced a movie recommendation system wherein the user specifies his/her preferences by choosing from a set of predefined attributes. Based on the cumulative weight of these selected attributes and by utilizing a K-means algorithm, the system recommends a list of movies to the user.

Regarding the third area, the authors of [25] suggested that combining collaborative filtering (CF) and content-based filtering (CBF) could result in a more efficient approach for some cases. The purpose of this hybrid approach is to merge the predictions of both CF and CBF to enhance recommendation performance. A different method for recommending books was presented in [26], which involved combining collaborative filtering (CF) with Association Rule Mining (ARM). ARM was utilized to uncover various patterns and associations between different items. By using both CF and ARM, this approach addressed the issues of data sparsity and the cold start problem related to recommendation systems. The approach presented in [27] consisted of the development of an e-learning recommender system, which involved utilizing ontology along with CF. The aim of this system was to incorporate learner characteristics such as learning style, study level, and skill level into the recommendation process. Ontology was used to achieve this. The proposed ontology-based recommendation approach was tested, and the authors observed that it yielded better performance than the CF algorithm used alone. Moreover, the approach helped to address the cold-start problem that occurs at the initial stages of recommendation. Finally, in [28], the authors suggested a hybrid recommendation system for the smartphone device market that combines Collaborative Filtering (CF) based on Alternating Least Squares (ALS) with deep learning techniques to improve recommendation performance and address the limitations of the CF approach. They utilized the outputs from ALS (CF) to influence recommendations generated by a Deep Neural Network (DNN), accounting for various types of information such as characteristic, contextual, structural, and sequential data within a big-data-processing framework. According to the results of their evaluation, the outcomes of the proposed system outperform those of several existing hybrid recommender systems.

On the contrary, this paper differs from the above literature in that it proposes an enhanced version of collaborative filtering to improve the performance of personalized search and hybrid recommender systems. The paper emphasizes the importance of addressing the cold start problem and data sparsity, wherein user ratings and feedback are often scarce. To tackle these challenges, the proposed approach incorporates both content-based filtering and collaborative filtering. Overall, the paper contributes to the development of hybrid recommender systems and personalized search techniques and provides insights into how these systems can be adapted to address emerging challenges.

3. Description of the Proposed Recommender System

The recommendation system is situated at the core of the digital library system within the application layer. To address the cold start problem when a new user joins, it employs a hybrid technique that combines Collaborative Filtering (CF) with a customized algorithm for content-based filtering. However, regardless of the situation, the recommendations generated by the CF model are prioritized, as the primary objective of this study is to demonstrate the impact of hybrid intelligent techniques' involvement in a digital library system.

To enhance the data collection process for the Collaborative Filtering (CF) technique, the system has been modified to gather user preferences through explicit ratings. By combining these two methods, the system can assign a total point rating, ranging from 0 to 10, to each book for each user. If a book's total point rating is 0, the system infers that there has been no interaction between the user and the book, indicating a lack of interest. The system calculates the sum of the total point ratings based on user actions, as follows:

- Adding a bookmark to a book: +2 points;
- Making a reservation that includes a book: +3 points;
- Rating a book: +1 to +5 points, depending on the rating given.

The ML.NET framework was utilized to construct, train, and assess the model that produces forecasts using collaborative filtering. The training data were presented to the trainer in matrix format, consisting of users, books, and total point ratings aggregated per user per book. To create the model, the trainer employs a matrix factorization algorithm, which decomposes the matrix into the product of two matrices of lower dimensionality, uncovering latent features in the process. The holdout method is used for this process, with 80% of the total data used for training and the remaining 20% utilized for evaluation after the model's training.

The newly created model enables the production of total point rating predictions for users and items by providing the user–book combination as an input. To create recommendations for a user, the model filters all books in the library and assigns them a predicted total point rating. Books that the user has already borrowed are excluded from this process to offer fresh recommendations and help them discover new items. After the filtration of the books is complete, books with a predicted total point rating of 0 are removed as they are deemed irrelevant to the user's preferences. The remaining books are sorted in descending order based on the predicted total point rating.

When a novel user signs up for the digital library system, the recommender algorithm may face challenges in providing quality recommendations due to the lack of prior interactions between the user and the books. Furthermore, it is possible that the user's profile might not yet have been integrated into the model, which results in there being no collaborative filtering (CF) predictions for that user. To mitigate these potential issues, the user is prompted during the registration process to specify at least one, and up to three, preferred categories and authors. These data are then utilized by a tailored content-based filtering (CBF) algorithm to generate personalized recommendations for the user until the CF model has been adequately trained in their interactions and is able to deliver high-quality recommendations.

The content-based algorithm functions in a straightforward manner, as its primary aim is to assist the CF recommendation process by utilizing a prioritization system. Books written by a preferred author and categorized in a preferred genre are granted the utmost priority, followed by books belonging to the preferred genre or written by a preferred author, the latter of which is chosen randomly to offer diverse recommendations. Finally, the remaining books are suggested. It is worth noting that if a book meets multiple priority criteria, it is solely categorized based on the highest priority. Additionally, if any book is recommended by the CF process, it is automatically placed in that category since CF recommendations hold the highest priority, as mentioned in the beginning of the section.

Suppose there is a requirement to generate N book recommendations, where N is any positive integer. In such a scenario, the following approach is implemented:

```
# Initialize the Recommendations list containing the final recommendations
Recommendations = []
# Check if the number of CF recommendations is greater than or equal to N
if len(CF_recommendations) $\geq$ N:
    # Add the top N books from CF recommendations to Recommendations
    Recommendations.extend(CF_recommendations[:N])
else:
    # Add all CF books to Recommendations (might be 0)
    Recommendations.extend(CF_recommendations)
# Check if the number of Recommendations is less than N
if len(Recommendations) < N:
    # Check if there are enough Author & Category recommendations that do not exist in
Recommendations
     if len(Author_Category_recommendations − set(Recommendations)) $\geq$ N −
len(Recommendations):
# Add (N − Recommendations) books from Author & Category that don’t exist in
Recommendations to Recommendations
     new_books = Author_Category_recommendations − set(Recommendations)
     Recommendations.extend(list(new_books)[:N − len(Recommendations)])
  else:
     # Add all Author & Category books that do not exist in Recommendations to
Recommendations (might be 0)
Recommendations.extend(list(Author_Category_recommendations − set(Recommendations)))
     # Check if the number of Recommendations is still less than N
if len(Recommendations) < N:
     # Add (N − Recommendations) random books from Author recommendations and Category
recommendations to Recommendations
     remaining_books = N − len(Recommendations)
random_author_books = random.sample(Author_recommendations, remaining_books)
    random_category_books = random.sample(Category_recommendations, remaining_books)
    Recommendations.extend(random_author_books)
    Recommendations.extend(random_category_books)
# Return the final Recommendations list
return Recommendations
```

In the absence of a minimum requirement for book recommendations, the priority system described earlier is applied to present all available books. Similarly, the same priority system is employed to generate personalized search results that match the user’s keyword search or filtering criteria. However, it is important to note that a book must conform to a certain extent to the user’s search/filtering criteria to be recommended through the searching/filtering functions of the application. Failure to meet this requirement may result in erroneous search results. Nevertheless, a minor sorting manipulation of the search results enables the application to offer a more personalized experience by elevating the most recommended search results to the top of the list.

Upon analysis of the above pseudocode, it is evident that is aims to generate recommendations by combining CF recommendations, Author & Category recommendations, and random selections. It considers the availability and diversity of recommendations to ensure a comprehensive and personalized suggestion list. The algorithm first checks if there are enough CF recommendations. If so, it selects the top N books from the CF recommendations. Otherwise, it includes all CF recommendations in the list. Next, it checks if the recommendation list is still below the desired size. If so, it considers Author & Category recommendations that are not already in the list. If there are enough of these recommendations, it selects the required number of books. Otherwise, it includes all remaining Author & Category recommendations. Finally, if the list's size is still below the desired level, it fills the remaining slots with random selections from the Author & Category recommendations. The algorithm returns the final recommendation list.

The novel aspect of this algorithm lies in its multi-source recommendation strategy. It takes advantage of diverse information to generate recommendations. This approach has the potential to capture various dimensions of user preferences and improve the quality and diversity of recommendations compared to those of single-source approaches.

In more detail, the presented method introduces a novel approach to recommendation generation by combining multiple sources of information, namely, Collaborative Filtering (CF) recommendations, Author & Category recommendations, and random selections. This hybrid recommendation strategy offers several unique and innovative aspects, which sets it apart from existing approaches in the field. The following is a list of key points highlighting the novelty of this work:

i. Integration of diverse recommendation sources: The algorithm exploits multiple recommendation sources, including CF recommendations, Author & Category recommendations, and random selections. The combination of collaborative filtering and content-based information enables a holistic assessment of user preferences, resulting in a more thorough understanding of their interests. By integrating these distinct sources, the algorithm gains the ability to encompass various facets of user preferences, resulting in recommendations that are potentially more precise and tailored to individual users.

ii. Adaptive handling of insufficient recommendations: The algorithm's noteworthy innovation lies in its ability to adaptively handle situations where the CF and Author & Category recommendations fail to meet the desired number of final recommendations (N). In such cases, the algorithm dynamically modifies its strategy to overcome the challenge of limited or incomplete recommendation data. It achieves this by considering additional Author & Category recommendations that have not yet been included or by incorporating random selections. This adaptive approach enhances the algorithm's resilience and increases the likelihood of generating meaningful recommendations, even in the absence of comprehensive data from individual sources.

iii. Exploration of novelty and serendipity: Incorporating random selections from the Author & Category recommendations introduces an intriguing dimension of serendipity and novelty to the generated recommendations. By including books that might not have been explicitly recommended by CF or Author & Category sources, the algorithm introduces an element of surprise and exploration, exposing users to new and unexplored options. This deliberate pursuit of novelty and serendipity sets the algorithm apart from traditional recommendation approaches that solely rely on explicit user–item interactions or predetermined patterns.

iv. Comparative advantage over single-Source algorithms: In contrast to algorithms that solely depend on CF or content-based techniques, the algorithm presented in this work possesses a clear advantage, namely, its amalgamation of multiple recommendation sources. By capitalizing on the merits of both CF and content-based approaches, it has the potential to alleviate the limitations inherent in single-source methods. The integration of diverse recommendation sources augments the algorithm's capacity to overcome well-known challenges such as data sparsity, cold-start scenarios, and limited coverage, ultimately leading to recommendations that are more accurate and diverse.

4. Example of Operation

To illustrate the capacity of the above pseudocode and the functionality of the proposed algorithm, we employ a cognitive walkthrough inspection by discussing the following examples of operation:

- User preferences—mystery novels and books by Agatha Christie;
- Collaborative filtering algorithm recommends eight books;
- The application needs to provide 10 recommendations;
- The algorithm checks if there are any additional books that match the user's preferences but are not already included in the recommendations (five such books are found);
- The algorithm calculates that it needs to add 2 books to reach the required 10 recommendations;
- Two of the five additional books that match the user's preferences are added to the recommendations;
- The algorithm delivers 10 recommendations and stops.

Thus, the final set of recommendations will include the top eight books from the CF algorithm, two books that match the user's author and category preferences but were not recommended by the CF algorithm, and zero random books (since the algorithm has already delivered ten recommendations).

Assuming that the user has now rated three additional books, the ratings are as follows:

- Book 1: three stars;
- Book 2: five stars;
- Book 3: two stars.

Based on these ratings, the CF algorithm recommends the following books with their predicted ratings:

- Book A: four stars;
- Book B: three stars;
- Book C: two stars;
- Book D: one star.

Since the user has rated at least four books, we will use the content-based algorithm in conjunction with CF to generate recommendations. Assume that N is set to 5. First, we check if there are at least N CF recommendations. In this case, there are, as we have four CF recommendations. Therefore, we add the top five recommendations from CF to the Recommendations list, which now includes:

- Book A: four stars;
- Book B: three stars;
- Book C: two stars;
- Book D: one star;
- Book E: zero stars.

Next, we check if there are at least N recommendations in total. Since there are only five recommendations so far, we need to add more books. We check if there are any Author & Category recommendations that do not exist in Recommendations and can potentially be added. Assume there are no such recommendations.

Therefore, we proceed to the next step, which is to add random books from Author recommendations and Category recommendations until we reach N. Since we already have five recommendations, we only need to add one more book. Suppose we randomly select a book called "The Art of War" from the "Strategy" category, which has a predicted rating of three stars.

Our final Recommendations list now looks like this:

- Book A: four stars;
- Book B: three stars;
- Book C: two stars;
- Book D: one star;

- Book E: zero stars;
- *The Art of War*: three stars.

5. Experimental Evaluation

The software evaluation stage plays a crucial role in determining the effectiveness and user satisfaction of software. For our specific situation, the evaluation phase lasted three months, during which time users utilized the software.

5.1. Sample

The experiment involved users who were between 22–35 years old and undergraduate or postgraduate students pursuing degrees in informatics and computer engineering, as shown in Table 1. A total of ninety students participated in the experiment, and they were divided into two equal groups by the evaluators (Group 1 and Group 2). Groups 1 and 2 were formed from the sample by the evaluators, with great attention paid to their formulation considering the students' characteristics, in order to ensure a qualitative evaluation.

Table 1. Users' characteristics.

Characteristics	Group 1	Group 2
Average age	29.5	28.5
Gender	20 women; 25 men	20 women; 25 men
Geographic location	All users live in the capital city of the country.	
Computer skills	Advanced skills (knowledge of using software systems)	
Motivation	Research and personal interests.	

Group 1 used the software that incorporated the enhanced collaborative filtering approach, while Group 2 used another version of the system that had the same interface but for which the recommendations were solely based on a user's previous viewings of specific material.

5.2. Results and Discussion

The evaluation of the system comprises two main aspects, namely, user experience and the effectiveness of the recommender system. To assess these aspects, a questionnaire providing questions to be rated on a 10-point Likert scale was administered to the users [29]. The questionnaire consisted of two questions for the evaluation of the first aspect and three questions for the evaluation of the second and third aspects, respectively (Table 2).

Table 2. Questionnaire.

Aspect	No	Questions
User Experience (UE)	Q1	Rate the user interface of the software.
	Q2	Rate the user experience after your last interaction with the software.
Effectiveness of Recommender System (ER)	Q3	Did the material correspond to your interests?
	Q4	Rate the adequacy of material that has been recommended to you.
	Q5	Would you recommend the software to your friends?

To ensure the reliability of the questionnaire, we used Cronbach's alpha [30] to test the consistency and stability of the questionnaire. The alpha coefficient was found to be 0.95, indicating that our scale and the specific sample had a high level of internal consistency. All the users answered the questionnaire at the end of the experimental period.

The answers provided by the users were grouped together according to the aspect of the questions they were answering. Additionally, the ratings they gave on a 10-point scale were transformed into three classifications: Low, which includes scores from 1 to 4; Average, which includes scores from 5 to 7; and High, which includes scores from 8 to 10.

Figure 1 displays the outcomes related to the responses provided by users belonging to Groups 1 and 2.

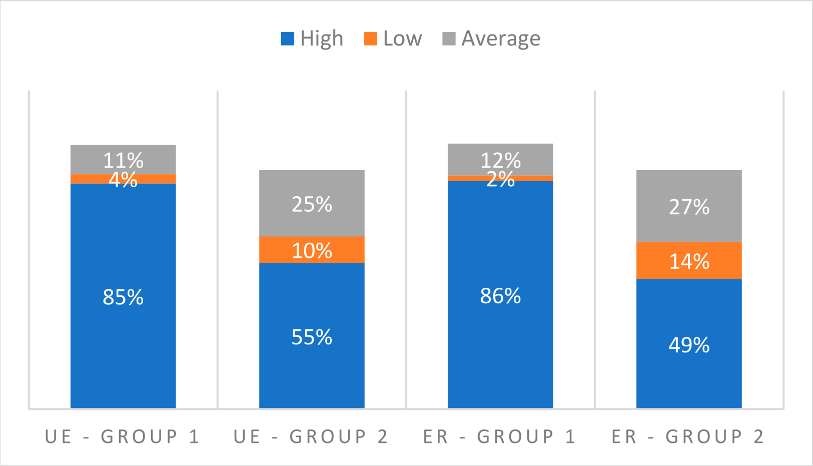

Figure 1. Aggregated results of questionnaire.

The responses of the students in Group 1 and Group 2 were compared with regard to the aspect of "User Experience". The data showed that 85% of the Group 1 students gave a positive rating, while only 55% of the Group 2 students acted in kind. The difference in ratings suggests that the Group 1 students had a better experience, which could be attributed to the personalized recommendations of material that they were given. In addition, 86% of the Group 1 students rated the "Effectiveness of the recommender system" highly, whereas 49% of the Group 2 students rated the same term in kind. This result was not surprising, as the presented recommender system utilizes an enhanced approach of collaborative filtering.

To delve deeper into the impact of the recommender system on users, a statistical hypothesis test (*t*-test) was conducted. The test compared the presented system, which was used by Group 1, to its conventional version, which was used by Group 2. Questions Q3–Q5 were included in the *t*-test analysis, and Table 3 displays the results of the test.

Table 3. *t*-test results.

	Q3		**Q4**	
	Group 1	**Group 2**	**Group 1**	**Group 2**
Mean	9.122222	4.911111	9.144444	5.555556
Variance	1.321973	4.351561	1.405868	3.732834
Observations	90	90	90	90
Pooled Variance	2.836767		2.569351	
Hypothesized Mean Difference	0		0	
df	178		178	
t Stat	16.77224		15.01947	
P(T $\leq$ *t*) one-tail	<0.0001		<0.0001	
P(T $\leq$ *t*) two-tail	1.653459		1.653459	

Based on the aforementioned results, it can be concluded that there is a statistically significant distinction between the means of the two experiments with regard to Q3 and Q4. The software utilized by Group 1 performed remarkably better in recommending material to users compared to its conventional version used by Group 2 (Q3: *t* Stat $\approx$ 16.77, $p < 0.05$). Furthermore, there was a significant variation in the adequacy of material that was recommended to the Group 1 users (Mean = 9.14, Variance $\approx$ 1.40) and Group 2 users (Mean = 5.55, Variance $\approx$ 3.73). The *t*-stat for Q4 was 15.01, and the *p*-value was less than 0.0001.

These outcomes indicate that the proposed approach outperforms its conventional counterpart in terms of recommending adequate material to users. It was anticipated that these results would be achieved since the software utilizes a novel approach to addressing the issues of data sparsity and a cold-start by utilizing CF and content-based filtering. In more detail, it incorporates an enhanced version of collaborative filtering to improve the performance of personalized searches.

An analysis of the literature, presented in Section 2, indicates that our proposed approach has not been adequately utilized. When comparing the approach presented in this paper to other recommender systems utilizing merely content-based filtering or collaborative filtering, it is important to note that the enhanced and hybrid approach offers a major advantage. The presented hybrid recommender system mainly utilizes collaborative filtering (CF) but with modifications to the data collection process for user preferences and a custom algorithm that integrates content-based filtering when data are insufficient. The modification considers all user actions in the system and not just explicit ratings, as sparse data can lead to poor recommendations when some users rarely or never rate items. The inclusion of content-based filtering allows for more accurate recommendations even with sparse data.

When comparing the presented approach to other recommendation algorithms, several distinctions can be highlighted:

a. CF vs. Content-based Approaches:

CF-based algorithms primarily rely on user–item interaction data to generate recommendations. In contrast, content-based approaches analyze item attributes or user profiles to make recommendations. The presented pseudocode also incorporates the content-based aspect. This hybrid nature of the algorithm can combine the strengths of CF and content-based approaches.

b. Fusion of Multiple Recommendation Sources:

Unlike algorithms that rely solely on CF or content-based techniques, the presented algorithm integrates a fusion of multiple sources, allowing for a more comprehensive consideration of user preferences, which, int turn, leads to more personalized and diverse recommendations.

c. Handling of Insufficient Recommendations:

The presented pseudocode accounts for situations where the number of CF recommendations or Author & Category recommendations may be less than the desired number of final recommendations (N). In such cases, it ensures that the final recommendation list is filled either by providing additional Author & Category recommendations or by randomly selecting books from Author and Category recommendations. This adaptive approach addresses the issue of insufficient recommendations and enhances the algorithm's robustness.

d. Consideration of Cold-Start Problem:

The presented pseudocode addresses the cold-start problem by incorporating random selections from Author and Category recommendations; the algorithm mitigates the impact of cold-start scenarios to some extent.

e. Trade-off between Novelty and Serendipity:

By including random selections in the final recommendation list, the algorithm introduces elements of serendipity and diversity in the recommendations. While this can provide users with unexpected and potentially interesting suggestions, it may also introduce recommendations that are less aligned with their specific preferences. Balancing the trade-off between novelty and personalized relevance is an important consideration when comparing this algorithm to others.

When comparing the enhanced collaborative filtering algorithm's performance to that of various existing recommendation strategies, such as content-based filtering, user-based collaborative filtering, item-based collaborative filtering, or a combination of these techniques, it should be noted that the presented approach surpasses existing techniques in terms of accuracy and efficacy. This approach produces better outcomes compared to previous methods while providing additional benefits such as greater scalability, interpretability, and customization.

To conclude, with respect to various aspects, the proposed technique differs from prior similar work. To begin with, it combines two separate recommendation sources with a flexible and adjustable algorithm, resulting in a diverse and individualized collection of recommendations. This differs from many other systems, which either rely on a single recommendation source or employ basic fusion techniques to mix them.

Second, the technique considers the availability and quality of the CF and Authors & Category recommendations to determine whether there are enough recommendations from each source and whether they are of high enough quality to be included in the Recommendations set. This ensures that the resultant set of recommendations is neither skewed nor dominated by a single source or type of book.

6. Conclusions

The overarching goal of this study is to show how modest modifications to the user-opinion-gathering process, content-based filtering, and the CF algorithm may greatly improve a recommender system. The recommender system can overcome the data sparsity constraint and provide more accurate recommendations by gathering additional data about users' opinions through their activities. Notably, in order to address the issue of noisy data, we used approaches such as data cleaning and outlier detection in order to improve the quality of the input data before using the enhanced collaborative filtering algorithm. This method can change the user experience in a digital library system from an overwhelming one to a tailored and smooth one. So far, the evaluation findings show an improvement in recommendation quality, while the system's performance remains unaffected. They also reveal a considerable improvement in recommendation quality, thereby confirming the recommender system's usefulness.

Although the presented approach has been developed for digital library environments, where users' information needs are very diverse and mutable over time and where the recommendation system must manage large-scale and dynamic data, it can be utilized in any type of software with appropriate modifications.

A limitation of this approach relates to instances when the CF recommendations and Author & Category recommendations are not diverse enough. In these cases, when the CF recommendations and Author & Category recommendations lack diversity, this approach faces a problem. In such a situation, a user might prefer books in a certain sub-genre or category. If the CF and Author & Category recommendations are highly skewed towards different genres or categories, the resulting recommendations set may be useless or irrelevant to the user. Thus, the approach may fail to produce a well-balanced and diversified collection of recommendations that match the user's expectations in this instance.

Future studies may include investigating reliable methods of collecting implicit opinions through user activities as well as establishing a balance between the length of the registration process and the number of preference data provided by the user. When there are few interaction data to work with, the obtained data will be used strategically to aid the recommender system in producing high-quality recommendations. The sample size and variety will be increased to further examine the results.

Author Contributions: Conceptualization, C.T., A.K. (Akrivi Krouska) and A.K. (Antonios Koliarakis); Methodology, C.T., A.K. (Akrivi Krouska) and A.K. (Antonios Koliarakis); Software, C.T., A.K. (Akrivi Krouska) and A.K. (Antonios Koliarakis); Validation, C.T., A.K. (Akrivi Krouska) and A.K. (Antonios Koliarakis); Formal analysis, C.T., A.K. (Akrivi Krouska) and A.K. (Antonios Koliarakis); Investigation, C.T., A.K. (Akrivi Krouska) and A.K. (Antonios Koliarakis); Resources, C.T., A.K. (Akrivi Krouska) and A.K. (Antonios Koliarakis); Data curation, C.T., A.K. (Akrivi Krouska) and A.K. (Antonios Koliarakis); Writing—original draft, C.T., A.K. (Akrivi Krouska) and A.K. (Antonios Koliarakis); Writing—review & editing, C.T., A.K. (Akrivi Krouska) and A.K. (Antonios Koliarakis); Visualization, C.T., A.K. (Akrivi Krouska) and A.K. (Antonios Koliarakis); Supervision, C.T., A.K. (Akrivi Krouska) and C.S.; Project administration, C.T. and A.K. (Akrivi Krouska). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used to support the findings of this study have not been made available because they contain information that could compromise research participants' privacy/breach their consent.

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

References

1. Ahmad, K. Human-centric Requirements Engineering for Artificial Intelligence Software Systems. In Proceedings of the 2021 IEEE 29th International Requirements Engineering Conference (RE), Notre Dame, IN, USA, 20–24 September 2021; pp. 468–473. [CrossRef]
2. Virvou, M.; Troussas, C.; Caro, J.; Espinosa, K.J. User Modeling for Language Learning in Facebook. In *Text, Speech and Dialogue. TSD 2012*; Lecture Notes in Computer Science; Sojka, P., Horák, A., Kopeček, I., Pala, K., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7499. [CrossRef]
3. Konstan, J.; Terveen, L. Human-Centered Recommender Systems: Origins, Advances, Challenges, and Opportunities. *AI Mag.* **2021**, *42*, 31–42. [CrossRef]
4. Troussas, C.; Krouska, A.; Sgouropoulou, C. A User-centric System for Improving Human-Computer Interaction through Fuzzy Logic-based Assistive Messages. In Proceedings of the 17th International Conference on Web Information Systems and Technologies—WEBIST, Online, 26–28 October 2021; SciTePress: Setúbal, Portugal; pp. 365–370, ISBN 978-989-758-536-4. [CrossRef]
5. Sakarkar, S.; Chaudhari, V.; Gaurkar, T.; Veer, A.; SCET, M.K. Web Personalisation based on User Interaction: Web Personalisation. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 234–238. [CrossRef]
6. Krouska, A.; Troussas, C.; Sgouropoulou, C. A Personalized Brain-Based Quiz Game for Improving Students' Cognitive Functions. In *Brain Function Assessment in Learning. BFAL 2020*; Lecture Notes in Computer Science; Frasson, C., Bamidis, P., Vlamos, P., Eds.; Springer: Cham, Switzerland, 2020; Volume 12462. [CrossRef]
7. Pandey, A.; Patil, K.; Sharma, S.; Kulkarni, M. Personalization of Information using Graph Convolutional Network. In Proceedings of the 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 8–9 October 2021; pp. 1–6. [CrossRef]
8. Troussas, C.; Chrysafiadi, K.; Virvou, M. Machine Learning and Fuzzy Logic Techniques for Personalized Tutoring of Foreign Languages. In *Artificial Intelligence in Education. AIED 2018*; Lecture Notes in Computer Science; Rosé, C.P., Martínez-Maldonado, R., Hoppe, H.U., Luckin, R., Mavrikis, M., Porayska-Pomsta, K., McLaren, B., du Boulay, B., Eds.; Springer: Cham, Switzerland, 2018; Volume 10948. [CrossRef]
9. Ragab, A.H.; El-Kafrawy, P. Using Knowledge Graph Embeddings in Embedding Based Recommender Systems. In Proceedings of the 2022 20th International Conference on Language Engineering (ESOLEC), Cairo, Egypt, 12–13 October 2022; pp. 129–132. [CrossRef]
10. Troussas, C.; Krouska, A. Path-Based Recommender System for Learning Activities Using Knowledge Graphs. *Information* **2023**, *14*, 9. [CrossRef]
11. Walek, B.; Fajmon, P. A Recommender System for Recommending Suitable Products in E-shop Using Explanations. In Proceedings of the 2022 3rd International Conference on Artificial Intelligence, Robotics and Control (AIRC), Cairo, Egypt, 20–22 May 2022; pp. 16–20. [CrossRef]
12. Krouska, A.; Troussas, C.; Sgouropoulou, C. Applying Genetic Algorithms for Recommending Adequate Competitors in Mobile Game-Based Learning Environments. In *Intelligent Tutoring Systems. ITS 2020*; Lecture Notes in Computer Science; Kumar, V., Troussas, C., Eds.; Springer: Cham, Switzerland, 2020; Volume 12149. [CrossRef]
13. Fararni, K.A.; Nafis, F.; Aghoutane, B.; Yahyaouy, A.; Riffi, J.; Sabri, A. Hybrid recommender system for tourism based on big data and AI: A conceptual framework. *Big Data Min. Anal.* **2021**, *4*, 47–55. [CrossRef]

14. Troussas, C.; Krouska, A.; Sgouropoulou, C. Improving Learner-Computer Interaction through Intelligent Learning Material Delivery Using Instructional Design Modeling. *Entropy* **2021**, *23*, 668. [CrossRef] [PubMed]
15. Islam, F.; Arman, M.S.; Jahan, N.; Sammak, M.H.; Tasnim, N.; Mahmud, I. Model and Popularity Based Recommendation System-A Collaborative Filtering Approach. In Proceedings of the 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 3–5 October 2022; pp. 1–5. [CrossRef]
16. Koliarakis, A.; Krouska, A.; Troussas, C.; Sgouropoulou, C. Modified collaborative filtering for hybrid recommender systems and personalized search: The case of digital library. In Proceedings of the 2022 17th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP), Corfu, Greece, 3–4 November 2022; pp. 1–6. [CrossRef]
17. Murad, D.F.; Hassan, R.; Wijanarko, B.D.; Leandros, R.; Murad, S.A. Evaluation of Hybrid Collaborative Filtering Approach with Context-Sensitive Recommendation System. In Proceedings of the 2022 7th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand, 19–20 May 2022; pp. 7–12. [CrossRef]
18. Choi, K.; Yoo, D.; Kim, G.; Suh, Y. A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electron. Commer. Res. Appl.* **2012**, *11*, 309–317. [CrossRef]
19. Roy, A.; Banerjee, S.; Sarkar, M.; Darwish, A.; Elhoseny, M.; Hassanien, A.E. Exploring New Vista of intelligent collaborative filtering: A restaurant recommendation paradigm. *J. Comput. Sci.* **2018**, *27*, 168–182. [CrossRef]
20. Chen, L.; Yao, Y.; Xu, F.; Xu, M.; Tong, H. Trading personalization for accuracy: Data debugging in collaborative filtering. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 159–169.
21. Hu, Y.; Xiong, F.; Lu, D.; Wang, X.; Xiong, X.; Chen, H. Movie collaborative filtering with multiplex implicit feedbacks. *Neurocomputing* **2020**, *398*, 485–494. [CrossRef]
22. Liu, H.; Hu, Z.; Mian, A.; Tian, H.; Zhu, X. A new user similarity model to improve the accuracy of collaborative filtering. *Knowl.-Based Syst.* **2014**, *56*, 156–166. [CrossRef]
23. Kumar, N.P.; Fan, Z. Hybrid User-Item Based Collaborative Filtering. *Procedia Comput. Sci.* **2015**, *60*, 1453–1461. [CrossRef]
24. Geetha, G.; Safa, M.; Fancy, C.; Saranya, D. A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System. *J. Phys. Conf. Ser.* **2018**, *1000*, 012101. [CrossRef]
25. Thorat, P.B.; Goudar, R.M.; Barve, S. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *Int. J. Comput. Appl.* **2015**, *110*, 31–36.
26. Patil, A.E.; Patil, S.; Singh, K.; Saraiya, P.; Sheregar, A. Online Book Recommendation System Using Association Rule Mining and Collaborative Filtering. *Int. J. Comput. Sci. Mob. Comput.* **2019**, *8*, 83–87.
27. Tarus, J.; Niu, Z.; Khadidja, B. E-Learning Recommender System Based on Collaborative Filtering and Ontology. *Int. J. Comput. Inf. Eng.* **2017**, *11*, 256–261.
28. Biswas, P.K.; Liu, S. A hybrid recommender system for recommending smartphones to prospective customers. *Expert Syst. Appl.* **2022**, *208*, 118058. [CrossRef]
29. Likert, R. A technique for the measurement of attitudes. *Arch. Psychol.* **1932**, *22*, 55.
30. Cronbach, L.J. Coefficient alpha and the internal structure of tests. *Psychometrika* **1951**, *16*, 297–334. [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.