



# Enhanced Books Recommendation Using Clustering Techniques and Knowledge Graphs

Priscila Valdiviezo-Diaz<sup>(✉)</sup>  and Janneth Chicaiza 

Departamento de Ciencias de la Computación y Electrónica,  
Universidad Técnica Particular de Loja, Ecuador  
{pmvaldiviezo,jachicaiza}@utpl.edu.ec

**Abstract.** Nowadays, there is an increased interest in recommender systems in different fields, since they allow personalizing the delivery of content according to the needs and preferences of each user. This paper proposes a recommendation approach that suggests books according to their metadata and explicit feedback given by users to them. First, we reuse a public dataset of books and use k-means to identify different groups of books based on their information. Second, we analyze the variation between these groups to predict the rating that a specific user would give a book. Third, we use the collaborative filtering technique to represent books, using the information of the user's rating and the group of the book to be recommended. Finally, users will receive an enhanced explainable response, i.e. a list of books with relevant metadata, i.e., the output could be helpful for users to understand the delivered recommendations. The original book's metadata was enriched using information available on DBPedia, a well-known RDF-based Knowledge Graph fed from Wikipedia. Results include experiments of clustering and prediction techniques because our approach considers the information of the user's ratings and the available information of the books to the recommendation, facilitating the understanding of the outputs. This approach achieves a good performance using Precision, Recall, and F-measure to measure the quality of the recommendations, and MAE to measure the prediction's accuracy.

**Keywords:** Books · Collaborative filtering · K-means · Knowledge graph · Recommender system

## 1 Introduction

Every day, users employ search engines to find resources such as books, articles, products, services, software, and people. Many of these resources also are accessible in specialized data sources such as institutional web pages, document repositories, and databases. However, it is not easy for users with specific needs to search and find relevant resources due to the large number of online resources

and data sources distributed all over the web. To alleviate this situation, recommender systems are one of the most popular applications provided by the industry.

In many contexts, the Recommender Systems (RS) are helpful to address the problem of delivering relevant material to users. RS can leverage the information available on resources and users to find similar resources according to the topic of interest and the user's profile. For example, integrating RS in e-learning environments enables students to enrich their learning process with online resources, reducing the time for selecting the right resources. In e-commerce, RS help customers find products to purchase.

RS are tools capable of filtering relevant information according to the needs and data of a specific user. For a target user, these systems identify the preferences, interests, and opinions of other users with similar interests or preferences as him/her. Different information filtering techniques are used for recommendation, including collaborative filtering, content-based, demographic, knowledge-based, and hybrid approaches. In this article, we focus on studying the first filtering strategy, which is, Collaborative Filtering (CF).

CF is the most popular technique for recommending appropriate items to a user. It uses a rating matrix that represents the user's tastes for an item in the past; that is, it uses the available ratings given by similar users. To discover suitable information for a target user, CF-based recommender systems can be designed by following two approaches [1]: (a) Memory-Based, which generally uses the K-Nearest Neighbor (KNN) algorithm to predict the rating of users using similarity measures (for example, Cosine, Pearson, Jaccard); these models are simple to implement, and the results are easy to understand [2]. (b) Model-Based, which uses a model to predict the rating of the registered users. Some examples of these models are those based on matrix factorization methods [3], bayesian models [4], clustering [5], and knowledge graphs [6]. The matrix factorization methods present good performance but the main problem with these methods is that may be very difficult to understand and explain the recommendations [7].

In addition to the lack of explanation of traditional RS, there are other associated problems such as scalability, data sparsity, and cold-start [2]. To address these issues, mainly the cold-start (new item), that is when no users of the system have rated the new item yet, this paper proposes a recommendation approach that suggests books according to explicit feedback (also called ratings) given by users and metadata of the books.

## 1.1 Related Work

Recommendation systems have been successfully implemented in domains such as e-commerce [8], e-learning [9], tourism [10], social networks [11] and health [12]. Likewise, interest in RS has increased due to information overload since they are at the forefront of personalizing content, adapting its delivery to the needs and preferences of each user.

In this section, we review proposals related to the use of CF and the k-means algorithm in recommendation systems. Later, we introduce the use of knowledge graphs in this kind of application.

**CF-Based Recommender Systems Using K-means Clustering Algorithm.** In the educational field, there are some researches where k-means had been used to cluster learners or items. According to [13], RS could have a key role to build virtual communities of similar interest. In this context, we highlight three proposals: (1) [14] propose a system that recommends suitable courses to learners based on their learning history and past performance. (2) [15] face the problem of low user satisfaction and long response time in traditional online English education platforms. Here, clustering is used to cluster the learners and a CF algorithm selects a set of relevant course materials for learners according to their English level. (3) [13] carried out a comparative study between K-Nearest Neighbors and K-means; the comparison is made to identify the most efficient algorithm in terms of prediction and accuracy.

In another field, e-commerce, for the recommendation of movies, we found three proposals: (1) [16] designed a recommender system that uses the concept of typicality to improve accuracy and predict ratings of the user based on ratings of neighbours. (2) Another similar proposal is presented by [17] that is based on user and item clustering to recommend movies for an active user. (3) [18] analyze two similarity measures: KL Divergence and Euclidean distance; the results show that CF using Euclidean distance metrics for similarity measure performs well than the KL divergence.

To alleviate the sparsity and the new user problem, in [19] presents a CF-based system in which users are clustered based on their *personality traits*. In the proposed method, users with similar personalities are placed in the same cluster using the k-means algorithm.

According to the analysis of the presented works, we can conclude that the k-means clustering algorithm has been used to group users, and thus deal with the cold-start problem of the new user. In the case of [13], K-means turns out to be better than KNN, and in [16] it turns out to be better than Topic Modelling. Furthermore, according to [15] user satisfaction is significant and the response times of the proposed system are acceptable. Unlike these works, our proposal uses K-means to deal with the cold-start problem of the new item, which means that the recommendation system will be prepared to offer recommendations when new books are added to the base.

**Use of Knowledge Graphs in Recommendation.** A Knowledge Graph (KG) contains pieces of knowledge that describe the entities of a domain, their attributes, and semantic relations between them [20].

According to [21], Artificial Intelligence will drive the next industrial technology revolution, and KGs comprise the main foundation of this revolution. Last years, knowledge graphs have been increasingly applied not only in academia but also in several industries [22].

The growing interest in KGs is because data structures organized as graphs facilitate the integration and querying of data from different sources. In addition, a graph allows describing complex relationships between data and reduces the chances of information-meaning problems occurring. These features facilitate the implementation of applications such as search engines, recommenders, question/answering systems, chatbots, and others.

Regarding recommendation, in a previous work [6], we found that features of a KG can be leveraged by a recommender system to overcome the problems faced by collaborative and content-based filtering approaches. Mainly, when there are no data about users or items, a knowledge-based RS can take advantage of domain knowledge and data available in open RDF-based KGs.

In this work, we use data available in DBpedia for the final stage of the recommendation, that is, to generate sets of items including key information that help the user to make a better decision.

## 1.2 Main Contributions

We reuse an online dataset of books, and to understand its structure, we carry out a dataset analysis to identify relationships between features and detect anomalies in the data. Next, we use unsupervised learning to analyze the related metadata to the books so that the preferences of a target user can be predicted according to books that other similar users have rated. Then, a model-based CF approach is used through a method of matrix factorization based on clustering for the prediction of books.

Also, to help users understand the output generated by a recommender system, the latest generations of RS include the use of Knowledge Graphs. Using knowledge of a given domain or cross-domain, and using structures of related entities as a graph, the system could make recommendations with some domain knowledge and tell the user why such items are recommended [6]. In this paper, in addition to proposing a CF-based recommendation approach, we try to take advantage of the data from the RDF-based graph named DBpedia which is fed from Wikipedia content. Specifically, in our proposal, we use the graph data to enrich the books' metadata to be recommended and generate outputs with the necessary information so that the user can take action as soon as the recommendations are received.

The paper is structured as follows: Sect. 2 includes materials and methods; Sect. 3 encloses the results of the experiments; and Sect. 4 shows the conclusions of this paper.

## 2 Material and Methods

In this section, we present two subsections: the data preparation realized for the experiments and the recommendation proposed approach. Programming and experiments were performed using the R language.

## 2.1 Data Preparation

For the experiments, we use two sources of data: (1) A dataset<sup>1</sup> available in the Kaggle repository that contains users' ratings for books and books' metadata; and (2) DBPedia<sup>2</sup> which is a repository that contains structured and machine-readable descriptions of millions of entities such as people, organizations, books, etc. The first dataset named *goodbooks-10k* was used for generating the recommendation model, and the second dataset was used to enrich the first one and explain the recommendation outputs.

**Data for Recommendations.** The dataset *goodbooks-10k* is made up of five CSV files with 981,756 ratings provided by 53,424 users on 10,000 books. From the original dataset, we use three files: ratings.csv, books.csv, tags.csv. The first file contains all users' ratings of the books, the second file contains 23 variables with information on the books such as the book identifier, best book identifier, the number of editions for a given work, author, publication year, ISBN, language, title, image URL, and details about the rating. Finally, tags.csv contains the tags users have assigned to that books.

From dataset analysis, we find that users tend to give positive ratings to books. Most of the ratings are in the 3–5 range, while very few ratings are in the 1–2 range. Furthermore, the largest number of books corresponds to the years of publication from 2000–2016. Next, we proceeded to perform a processing of our data set in which:

- Some variables that do not provide relevant information about books were removed from the file books.csv
- Duplicate ratings were removed.
- Users who rated fewer than 5 books were removed.
- 25% users were selected from the original dataset for experiments setting.
- Genres attribute is calculated by grouping similar tags.

Once the data processing has been carried out, a final data set is obtained. Table 1 shows the information of the final data set used for generating the recommendations.

**Table 1.** Descriptive information of the final data set

Entity	Count
#Users	8,902
#Books	9,497
#Ratings	231,119
Scale	5-stars
#Book attributes	13

<sup>1</sup> <https://www.kaggle.com/zygmunt/goodbooks-10k>.

<sup>2</sup> <https://www.dbpedia.org>.

**Data for Improving Explanations.** To enrich descriptions of books, we annotate the original data using the Tagme API<sup>3</sup>. From the books' titles, the API finds candidates between the Wikipedia pages that match the title. After a semi-supervised process of checking, we find the equivalent Wikipedia pages for 4,548 books which represents one in two resources that make up the population. Next, from this corpus, we select 3,129 books that are part of the books set that was used for generating the recommendations. The metadata of these books was used to enrich the explanation of the recommendations given to the user.

From the set of verified pages, we implement queries using the SPARQL language and the DBpedia Endpoint to extract additional data from each book such as: abstract, type of work (book, novel, series, collection, etc.), publisher, author's page, country and external links where the user can find detailed information about books and its authors.

Data collected from DBpedia is used at the final stage of the recommender system to provide users with organized and dynamic information about the books that can be interesting for them.

## 2.2 Recommendation Approach Based on Books Clustering

In the proposed approach, the neighbour of an item is found based on the application of K-means. Unlike traditional approaches which use similarity measures based on memory for computing the neighbours of an item, in this case, we compute with a clustering algorithm, which makes it more efficient to generate a more similar neighbourhood set to the target item.

Three phases are part of this approach: (1) book clustering (2) rating prediction, and (3) book recommendation and visualization.

**Books Clustering.** To generate the books clusters according to their characteristics we used the K-means algorithm. The first step is to find the optimal number of clusters. We determine the optimal number of clusters using the elbow method [23], obtaining as result 4 clusters as the most optimal. The next step is to cluster each book based on its similarity using the K-means algorithm. Books with the most common characteristics are grouped into the same cluster. We evaluated the grouping using silhouette score [24], with which we obtained a value of 0.613.

From the cluster analysis, we can highlight the following:

- 23% of the books are in Cluster 1, whose year of publication is mostly in the range 2006–2009. Most of the books correspond to the genre: fantasy, historical-fiction and science-fiction. This group contains books with few qualifications followed by cluster 4. This means that the number of users' ratings on those books is small.

<sup>3</sup> <https://sobigdata.d4science.org/web/tagme/tagme-help>.

- In Cluster 2, about 5% of the books are grouped, most of which correspond to the year 2006. The genre of book that stands out the most in this group is fantasy followed by historical-fiction. These books have the highest ratings compared to other groups. However, although it is the smallest cluster, it is the group that has the books that have received the most ratings from users, that is, this group focuses on the most rated books.
- 61% of the books are in cluster 3, the largest number of books of this group correspond to the years 2003–2016, this group also contains the older texts. A considerable amount of books is related to genres: fantasy, mystery, horror, and memoir. In this group are the less frequently rated books.
- 11% of books belong to cluster 4, most of which correspond to the year 1999–2002, the genre of book that stands out the most in this group is fantasy and science-fiction, Moreover is the group with the lowest ratings.

**Rating Prediction.** We proceed to make the predictions of the ratings of the books that have not been seen by users using the CF technique, for we generate a predicted rating matrix using the Probabilistic Matrix Factorization (PMF) [25] method. We selected this method because it provides good prediction and recommendation results, as well as a higher scalability [26]. PMF factorizes the rating matrix into two matrices that represent the users and items in a latent factor space of dimensionality  $k$  [27], in our case,  $k$  value will be the optimal number of clusters.

The prediction of rating is computed by the dot product between user feature vectors and book feature vectors that belong to a specific cluster. Therefore, for the predictions, all books of the dataset are not considered but only the books of the cluster where the target book is located.

The predicted rating that the user  $u$  will give to the book belonging to a given cluster  $ic$  is computed as shown in the following equation:

$$\hat{r}(u, ic) = p_u \cdot q_{ic}^T \quad (1)$$

where:  $p_u$  is the latent user vector, and  $q_{ic}$  is the latent items vector of a given cluster.

Table 2 shows an example of how the CF matrix is represented with the group obtained for each book.

**Table 2.** Example collaborative filtering matrix

	<b>Book1</b>	<b>Book2</b>	<b>Book3</b>	<b>Book4</b>	<b>Book5</b>
User1	4	0	0	0	2
User2	0	3	0	0	0
User3	0	1	3	0	5
	Clus 1	Clus 2	Clus 1	Clus3	Clus4

**Books Recommendation and Visualization.** We generate a recommendation list made up of books that a user has not read and rated, and which belong to the same cluster, the books with a high rating prediction are used to compile the recommendation list, and then those are recommended to the target user. The idea is to optimize the recommendation algorithm such that it generates faster recommendations without affecting the recommendation quality.

For a given user, the recommender generates a list of books made up of the identifiers of the books that may interest him/her the most. With this base information, we query the system’s database in order to retrieve relevant attributes of each book.

The system database contains the book’s data, obtained from the Kaggle dataset, and other metadata that was obtained from DBPedia. Thanks to this additional data, we were able to create a result preparation and visualization service. The service returns the user with basic information on each book, but also it allows the user to refine the results, due to the links associated with key metadata such as country, type of work, genre, author, and country. In the next section, we illustrate the type of output that the user receives from our system.

### 3 Results

In this section, we describe the experimental settings to evaluate the effectiveness of the approach. We have tested our recommendation algorithm on the book dataset. We used cross-validation to obtain improved results.

#### 3.1 Experimental Results

To analyze the behaviour of the system and the quality of the predictions, we use Mean Absolute Error (MAE) [28], and to measure the quality of recommendations, we use Precision, Recall, and F-measure.

The results were compared with two different algorithms: traditional CF based on items (IBCF), and the Bayesian non-negative matrix factorization (BNMF) algorithm, using the same sample of users. IBCF estimates the user’s rating for the target item based on ratings given by the user to other similar items [29]. BNMF is based on factorizing the rating matrix into two non-negative matrices whose components provide an understandable probabilistic meaning [26].

During the experiments, we set the following hyperparameters for each CF algorithm.

- IBCF was used with the cosine similarity to determine the nearest neighbours to the target book and predict the target user’s rating. The number of neighbours of a book was  $nn = 30$ .
- With BNMF a vector of  $K$  components to each user and each item is necessary, therefore we have the following hyperparameters:  $k = 8$ ,  $\alpha = 0.8$  and  $\beta = 3$ ,  $number\_of\_iterations = 120$ .



- Cluster-based CF (the proposed approach) was started with: number of latent factors (optimal number of clusters)  $k = 4$ ,  $\alpha = 0.02$  and  $\beta = 0.02$ ,  $number\_of\_iterations = 80$ .

These hyperparameters were selected to maximize the accuracy of the CF algorithms for the quality measures used.

Table 3 contains the prediction quality values for the dataset used in the experiments with each CF approach.

**Table 3.** MAE Results

Algorithm	MAE
IBCF	0.6804
BNMF	0.5932
Cluster-based CF	0.5684

According to results shown in Table 3, our approach based on clustering provides more accurate predictions, since the MAE value is lower compared with IBCF and BNMF. We believe that this improvement is because the proposed approach, unlike baselines used, predicts based on those items most similar to the target item that belong to the same cluster.

In addition to obtaining acceptable predictions, our approach has the advantage of providing understandable item vectors since they are clustered according to their similar features. This allows us to easily explain the recommendations.

Figure 1 contains the quality values for Precision and Recall. We can observe that Cluster-based CF and IBCF provide better precision than the BNMF algorithm. On the other hand, when the number of recommendations increases, the precision is better with Cluster-based CF compared to other algorithms. This figure shows also that when the number of recommendations increases, BNMF exhibits better performance in recall compared to IBCF and Cluster-based CF algorithm.

Figure 2 shows the F-Measure value obtained for each algorithm. F-Measure combines precision and recall to measure the performance of the recommender system. The F-measure value of Cluster-based CF is the highest among the three algorithms when the number of recommendations is greater than 5. Moreover, Fig. 2 shows that when the number of recommendations increases, the F-measure value of both approaches (IBCF and Cluster-based CF) increases.

From the results shown in Figs. 1 and 2, it seems that Cluster-based CF is better than the IBCF and BNMF in terms of MAE, precision and F-measure. Moreover, cluster-based CF shows a behaviour similar to BNMF in recall when the number of recommendations increases.

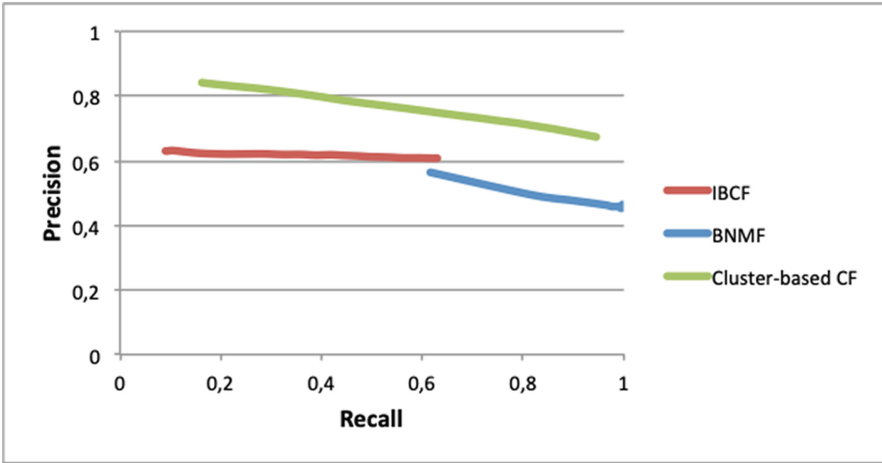


Fig. 1. Precision and recall results of each CF algorithm

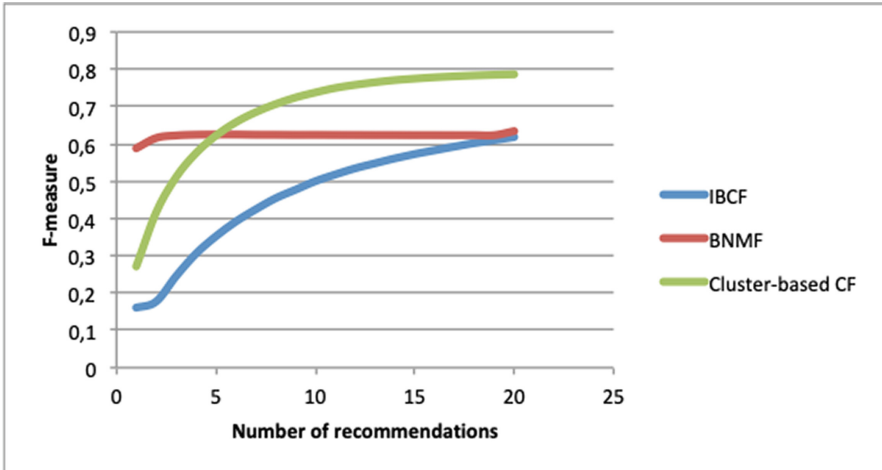


Fig. 2. F-measure results of each CF algorithm

### 3.2 Output Presentation

Figure 3 illustrates how recommendations are displayed to a target user. Here our recommendation proposal generates the top-3 list of books that could interest him/her the most. If the predicted rating for a given recommended book meets the relevance threshold of 4, then it will be displayed to the target user.

By providing extra and relevant information about each top book, users could be able to understand and refine results using facets like author, country, type of work, and genre. Likewise, if the user wishes, he/she could go to Wikipedia or DBpedia to review additional information on a specific book. Finally, a rating



**Fig. 3.** Top-3 of the recommended books for a target user. The screenshot is what the target user receives when interacting with the system.

heatmap of each work is generated automatically, so the user can know which is the book with the highest proportion of ratings by each scale value [1 : 5].

We believe that this type of output will allow users to understand the recommendations and support the suggestions provided by the algorithm.

### 3.3 Discussion

The cluster-based CF approach mainly allows dealing with the coldstart (new-item) problem. Every time the system receives a new book, it will be classified using the clusters; thus the system could recommend to the user a set of books based on the prediction of ratings for similar books belonging to the same cluster. However, as occurs in most recommendation systems based on CF, one of the limitations of our approach occurs when there are unregistered users, who have not yet rated any book (new user coldstart problem). To alleviate this problem, in the future we will try to complete the proposed approach by clustering user information too, in such a way that we obtain a hybrid approach.

The experimental results also show that our approach can achieve good results when the user-item matrix is sparse. On the other hand, the scalability issue has been addressed by grouping books sharing common features so that only a smaller subset of books needs to be considered during the recommendation.

## 4 Conclusions

This paper presented a CF-based recommendation approach designed to alleviate the cold-start problem of new items. The proposal employs a clustering algorithm to compute several features of books and build an item list that provides recommendations close to the users' preferences. In addition, the approach allows the user to understand the recommendation output which is enhanced with the data collected from a knowledge graph. In this way, our approach allows users to get easily interpretable results and provides them with a better understanding of the recommendations.

We tested the cluster-based CF approach using the book dataset and two CF baselines: the IBCF memory-based algorithm and BNMF model-based algorithm. The results showed an improvement in the proposed approach with respect to other CF algorithms in MAE. Likewise, it gets better precision results, especially when the number of recommendations is low. Hence, the results obtained with cluster-based CF are considered acceptable, since the approach allows predicting the rating a book could have considering the explicit feedback given by users and books metadata.

As future work, we plan: a) to extend the proposal to face the cold-star problem of new-user by using domain knowledge and connecting open or queryable data on the web, b) to make more experiments applying other model-based algorithms by using large-scale datasets, and c) to extend the proposed approach to large-scale cases like distributed ones to determine if the quality of the recommendations improves and the user experience.

## References

1. Hernando, A., Bobadilla, J., Ortega, F., Gutiérrez, A.: A probabilistic model for recommending to new cold-start non-registered users. *Inf. Sci.* **376**, 216–232 (2017). <https://doi.org/10.1016/j.ins.2016.10.009>
2. Chen, R., Hua, Q., Chang, Y.S., Wang, B., Zhang, L., Kong, X.: A survey of collaborative filtering-based recommender systems: from traditional methods to hybrid methods based on social networks. *IEEE Access* **6**(October), 64301–64320 (2018). <https://doi.org/10.1109/ACCESS.2018.2877208>
3. Shi, Y., Lin, H., Li, Y.: IU-PMF: probabilistic matrix factorization model fused with item similarity and user similarity. In: *International Conference on Cloud Computing and Security*, pp. 747–758 (2017)
4. Valdiviezo-Diaz, P., Ortega, F., Cobos, E., Lara-Cabrera, R.: A collaborative filtering approach based on naïve bayes classifier. *IEEE Access* **7**, 108581–108592 (2019). <https://doi.org/10.1109/access.2019.2933048>
5. Mohammadpour, T., Bidgoli, A.M., Enayatifar, R., Javadi, H.H.S.: Efficient clustering in collaborative filtering recommender system: hybrid method based on genetic algorithm and gravitational emulation local search algorithm. *Genomics* **111**(6), 1902–1912 (2019). <https://doi.org/10.1016/j.ygeno.2019.01.001>
6. Chicaiza, J., Valdiviezo-Diaz, P.: A comprehensive survey of knowledge graph-based recommender systems: technologies, development, and contributions. *Information* **12**(6), 232 (2021). <https://doi.org/10.3390/info12060232>

7. Rastegarpanah, B., Crovella, M., Gummadi, K.P.: Exploring explanations for matrix factorization recommender systems. In: FATREC 2017. No. 1 (2017). <https://doi.org/10.18122/B2R717>
8. Adak, F., Ucar, M.: A book recommendation system using decision tree-based fuzzy logic for e-commerce sites. In: HORA 2021–3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings (2021)
9. Gulzar, Z., Leema, A.A., Deepak, G.: PCRS: personalized course recommender system based on hybrid approach. *Procedia Comput. Sci.* **125**, 518–524 (2018). <https://doi.org/10.1016/j.procs.2017.12.067>
10. Orama, J.A., Borràs, J., Moreno, A.: Combining cluster-based profiling based on social media features and association rule mining for personalised recommendations of touristic activities. *Appl. Sci.(Switzerland)* **11**(14) (2021). <https://doi.org/10.3390/app11146512>
11. Tey, F.J., Wu, T.-Y., Lin, C.-L., Chen, J.-L.: Accuracy improvements for cold-start recommendation problem using indirect relations in social networks. *J. Big Data* **8**(1), 1–18 (2021). <https://doi.org/10.1186/s40537-021-00484-0>
12. Yang, H., Gao, H.: User recommendation in online health communities using adapted matrix factorization. *Internet Res.* **31**(6), 2190–2218 (2021). <https://doi.org/10.1108/INTR-09-2020-0501>
13. Zriaa, R., Amali, S.: A comparative study between k-nearest neighbors and k-means clustering techniques of collaborative filtering in e-learning environment. *Innovations in Smart Cities Applications Volume 4*, pp. 268–282 (2021). [https://doi.org/10.1007/978-3-030-66840-2\\_21](https://doi.org/10.1007/978-3-030-66840-2_21)
14. Mondal, B., Patra, O., Mishra, S., Patra, P.: A course recommendation system based on grades. In: 2020 International Conference on Computer Science, Engineering and Applications, ICCSEA (2020). <https://doi.org/10.1109/ICCSEA49143.2020.9132845>
15. Huang, Y.: Research on design and application of online english education platform based on web. *Int. J. Antennas Propag.* **2021** (2021). <https://doi.org/10.1155/2021/7648856>
16. Kaur, S., Challa, R., Kumar, N., Solanki, S., Sharma, S., Kaur, K.: Recommendation generation using typicality based collaborative filtering. In: Proceedings of the 7th International Conference Confluence 2017 on Cloud Computing, Data Science and Engineering, pp. 210–215 (2017). <https://doi.org/10.1109/CONFLUENCE.2017.7943151>
17. Alok, S., Jagadev, K., Mohanty, S.: A collaborative filtering approach for movies recommendation based on user clustering and item clustering, *Communications in Computer and Information Science*, 906 (2018)
18. Jeyasekar, A., Akshay, K., Karan: Collaborative filtering using Euclidean distance in recommendation engine. *Indian J. Sci. Technol.* **9**(37), 1–5 (2016)
19. Yusefi Hafshejani, Z., Kaedi, M., Fatemi, A.: Improving sparsity and new user problems in collaborative filtering by clustering the personality factors. *Electron. Commer. Res.* **18**(4), 813–836 (2018). <https://doi.org/10.1007/s10660-018-9287-x>
20. Liu, Z., Han, X.: Deep learning in knowledge graph. In: Deng, L., Liu, Y. (eds.) *Deep Learning in Natural Language Processing*, pp. 117–145. Springer, Singapore (2018). [https://doi.org/10.1007/978-981-10-5209-5\\_5](https://doi.org/10.1007/978-981-10-5209-5_5)
21. Zhao, M., et al.: Construction of an industrial knowledge graph for unstructured chinese text learning. *Appl. Sci.* **9**(13), 2720 (2019). <https://doi.org/10.3390/app9132720>

22. Bader, S.R., Grangel-Gonzalez, I., Nanjappa, P., Vidal, M.-E., Maleshkova, M.: A knowledge graph for industry 4.0. In: Harth, A., Kirrane, S., Ngonga Ngomo, A.-C., Paulheim, H., Rula, A., Gentile, A.L., Haase, P., Cochez, M. (eds.) ESWC 2020. LNCS, vol. 12123, pp. 465–480. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-49461-2\\_27](https://doi.org/10.1007/978-3-030-49461-2_27)
23. Muhammad-Ali, S., Bain-Khusnul, K., Eka-Mala, S., Satoto, B.D.: Integration k-means clustering method and elbow method for identification of the best customer profile cluster. *IoP Conf. Ser.: Mater. Sci. Eng.* **336**, 012017 (2018). <https://doi.org/10.1088/1757-899x/336/1/012017>
24. Shahapure, K.R., Nicholas, C.: Cluster quality analysis using silhouette score. In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), pp. 747–748 (2020). <https://doi.org/10.1109/DSAA49011.2020.00096>
25. Huang, L., Tan, W., Sun, Y.: Collaborative recommendation algorithm based on probabilistic matrix factorization in probabilistic latent semantic analysis. *Multi-media Tools Appl.* **78**(7), 8711–8722 (2018). <https://doi.org/10.1007/s11042-018-6232-x>
26. Bobadilla, J., Bojorque, R., Hernando, A., Hurtado, R.: Recommender systems clustering using bayesian non negative matrix factorization. *IEEE Access* **3536**, 1–1 (2018). <https://doi.org/10.1109/ACCESS.2017.2788138>
27. Pujahari, A., Sisodia, D.S.: Pair-wise preference relation based probabilistic matrix factorization for collaborative filtering in recommender system. *Knowl.-Based Syst.* **196**(3), 105798 (2020), cited By :22
28. Wang, W., Lu, Y.: Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. *IoP Conf. Ser.: Mater. Sci. Eng.* **324**(1), 012049 (2018). <https://doi.org/10.1088/1757-899x/324/1/012049>
29. Thakkar, P., Varma, K., Ukani, V.: Outcome fusion-based approaches for user-based and item-based collaborative filtering. In: Satapathy, S.C., Joshi, A. (eds.) ICTIS 2017. SIST, vol. 84, pp. 127–135. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-63645-0\\_14](https://doi.org/10.1007/978-3-319-63645-0_14)