A Context-Aware Recommendation System with Effective Contextual Pre-Filtering Model

Duaa H. Hameed*, Rehab F. Hassan
Department of computer science, University of technology- Iraq, Baghdad, Iraq
E-mail: cs.21.22@grad.uotechnology.edu.iq, rehab.f.hassan@uotechnology.edu.iq
*Corresponding author

Keywords: collaborative filtering, contextual information, context, context-aware, pre-filtering

Received: July 31, 2023

Informational resources have significantly expanded as a result of the growth of the internet. Consequently, making personalized suggestions about different types of information, goods, and services is the best strategy to assist customers in solving the issue of information overload. As a result, recommendation systems are employed to aid clients in locating the products most appropriate to their interests. The majority of traditional recommender systems rely on a traditional model that just takes into account user-item-rating interactions without taking context into account. It has been demonstrated that context-aware recommender systems deliver improved predicted performance across a variety of areas by attempting to adapt to users' preferences across various settings. This study presents a proposed system to help the recommender system solve its difficulties in producing accurate predictions that are relevant to the user's preferences. The system is the Contextual Pre-filtering Based Collaborative Filtering (CPBCF) model, which is based on splitting items. To decrease the time and space needed for processing correlations, it depends on the recommended splitting approach utilizing the variance equation, which decreases the dataset depending on the most important attributes. In the proposed system experiments, the performance of CPBCF with and without contextual pre-filtering was enhanced by (5-7%) for the precision, (7-8%) for the recall, and (7-8%) for the f1-measuer. While the complexity time has enhanced by (3-4 sec). The effectiveness of the CPBCF model was evaluated using various numbers of neighbors. We can observe that neighborhood size does have an effect on forecast accuracy.

Povzetek: Kontekstno ozaveščeni priporočilni sistem z učinkovitim modelom kontekstnega predfiltriranja (CPBCF) izboljšuje kvaliteto priporočil z upoštevanjem kontekstnih informacij in uporabo metode razdelitve podatkov na podlagi variance.

1 Introduction

Finding what is needed among the vast amount of data in today's information-overloaded environment is difficult [1]. System designers suggested the Recommendation System (RS) as a solution for this issue [2]. By improving customers' ability to select the ideal products based on their preferences, RSs aim to improve the efficiency of E-Commerce (EC) systems [3]. Due to RS's importance in the field of EC, both the business community and people in general have shown an interest in it [4]. Recommendations are helpful in a wide range of circumstances, including recreation, online shopping, social networking, job portals, locating relevant web pages, and many more [5]. Several businesses, including Amazon.com, Netflix, Half.com, CDNOW, J.C. Penney, and Procter & Gamble, are examples of those that have successfully used commercial RSs. Additionally, these businesses reported higher client loyalty and better online and catalog sales [6].

In the past, user and item dimensions have been used as the basis for RSs, working under the assumption that a set of N things will interest M users [7]. The quality of these traditional RS suggestions has been reported to be somewhat low over time due to the uniformity of the information sources and a lack of user and item data [8]. It is usually affected by additional and varied criteria, sometimes referred to as "contextual factors," when determining how exactly an object is rated [9]. According to Dey et al., "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." Contextual information may therefore be crucial in RS [10]. The Context-Aware Recommendation System (CARS) has recently offered more precise and enjoyable suggestions for customers by incorporating contextual information into the RS process. Consequently, depending on the results of the scoring system, user, item, and context information are incorporated into the settings. When incorporating contextual information into CARS, there are three difficulties to be resolved: contextual pre-filtering, contextual post-filtering, and contextual modeling [11]. Use contexts as filters in contextual pre-filtering methods to eliminate pointless rating profiles. The suggestion list may then be created by applying any standard recommendation algorithms to the remaining ratings [12]. In the post-filtering stage, it initially ignores contextual information and then uses any standard Two Dimension (2D) RS on the entire dataset to forecast ratings. After that, each user's final list of suggestions is altered (contextualized) based on the contextual data [13]. In contextual modeling, the context information is immediately included in the suggested strategy in this recommendation technique, for example, as a step in the priority calculation process [11].

The remainder of the paper is divided into the following sections: There is a literature review in Section II. Section III explains the recommended method, which identifies the most important feature from the user's self-evaluation to produce contextualized suggestions. Section IV focuses on the experimental results of the proposed system. In Section V, the discussion is cleared. In Section VI, the conclusion is presented.

2 Literature review

Studies and research efforts indicate numerous approaches to creating customized CARS, as seen below.

In 2018, A. Bozanta and B. Kutlu suggested that in order to get around each method's limitations, a hybrid recommendation model be utilized to mix user- and item-based Collaborative Filtering (CF), Content-Based Filtering, and contextual data. Additionally, it is predicted for each user-venue conjunction under which specific circumstances the user would choose a certain venue. Additionally, each user has their own individually defined threshold values that signify their choice of a venue. In order to evaluate the results, experiments involving precision, recall, F1-measure, and user studies were employed. In both a user study and an experimental evaluation utilizing a real-world dataset, the RS outperformed the baseline system [14].

In 2019, M. Singh et al. proposed a novel approach based on splitting criteria for usage in apps for movie selection. This method creates a modified dataset that splits the original single item into two virtual items based on contextual value. The single user is then split into two virtual users based on contextual factors. Only when there is a significant difference between two virtual objects can a person or object be separated. Further, user-based CF is used to provide useful suggestions. The results show that the proposed technique is effective in light of several performance measurement criteria using the LDOSCOMODA dataset [15].

In 2019, Jesús Silva et al. might evaluate the effectiveness of collaborative systems for directing and assisting students in this decision-making process by looking at how these systems operate and how they impact the use of data that is different from the formal information that students frequently use. The research used the clustering-based Multi Dimension tensor factorization approach to develop an RS and demonstrate that the inclusion of tensors improves the suggestions' accuracy. This strategy enables the user to use contextual information to solve the sparsity issue and enhance recommendation accuracy [16].

In 2020, I. M. AL Jawarneh et al. created a hybrid algorithm that takes a contextual incorporation pre-filtering strategy and adapts and repurposes it, giving a new dimension to a DL-based neural CF method and recovering the advantages of both without their drawbacks. They also provided numerical results suggesting statistically significant margins of improvement for our method over the baselines [17].

In 2021, K. V. Rodpysh et al. introduced the innovative context-aware CSSVD recommendation method. The DPCC and IFPCC similarity criterion matrices, as well as the item's user property attribute matrices, were initially constructed in the CSSVD matrix before being used to produce the SSVD matrix for the Cold Start Problem (CSP). In the second phase, the context matrix is created utilizing the contextual data and the CWP similarity criteria. This matrix, which is based on the SSVD matrix created in the previous stage, creates a three-dimensional matrix based on tensor properties and solves the sparse data issue. They have used IMDB and STS data collection because they are evaluating the recommended approach by taking into account user characteristics, object properties, and contextual information. Experiential data demonstrate that the recommended algorithm CSSVD is equal to TF, HOSVD, BPR, and CTLSVD in terms of accuracy, recall, F1-measure, and NDCG measure. The results show how decreasing CSP and sparse data have enhanced suggestions to customers [11].

In 2021, Z. El Yebdri et al. trust statements have been used as a rich information source with context compensation in the development of a hybrid trust-based, context-aware post-filtering technique. This approach utilized the relative average difference between the context on the CF output that is trust conscious by adding explicit and implicit trust information. They used the idea of trust prior to generating predictions in order to exclude those with little trust from the confidence network. The results of the studies show that the recommended method outperforms suggestion techniques and standard RS on real-world datasets [18].

Table 1, which lists several methods used in various fields and displays the assessment criteria that authors employed to evaluate the efficiency and precision of the system.

3 Methodology

With the help of this suggested methodology, movies are recommended while taking numerous contextual factors into account. For Improve the accuracy of recommendations with contextual pre-filtering. Contextual factors have the ability to have an important impact in addition to explicit ratings in affecting user preferences. The dataset is reduced based on the contextual attribute, and a suggestion list is created using the smaller dataset.

Table 1: A comparative analysis of the CARS

Author, year	Dataset	Technique	Result
(A. Bozanta	Real-world dataset	A hybrid recommendati	Precisio n 0,282,

and B. Kutlu, 2018) [14]		on model can be utilized to mix user- and item-based collaborative filtering (CF), content-based filtering, and contextual data	Recall 0,276, F1-measure 0,279
(M. Singh et al., 2019) [15]	LDOSCOMODA dataset	Splitting criteria for usage in apps for movie selection	MAE 0,9024
J. Silva et al., 2019) [16]	Studied dataset	Clustering-based multi-dimensional tensor factorization	Recall 0,65, Precision 0,81, MAE 1,06
(, I. M. AL Jawarneh et al., 2020) [17]	Movielens 1M, DePaulMovie, and TripAdvisor datasets	Hybrid algorithm that takes a contextual incorporation pre-filtering strategy and adapts and repurposes it	MAE 2,35%, Accuracy 6.4%
(K. V. Rodpysh et al., 2021) [11]	STS and IMDB datasets	Context-aware CSSVD recommendation	RMSE 1,007, Precision 0,361, Recall 0.336, F1-measure 0,342 NDCG 0.5
(Z. El Yebdri et al., 2021) [18]	Real-world datasets	Development of a hybrid, trust-based, context-aware post-filtering	MAE 0,79, RMSE 1

The pre-filtering method builds an appropriate 2D ($item\times users$) dataset using the contextual attributes that are most pertinent to the scenario. To determine the variance for a contextual attribute, compute the percentage of values in the array that match the specified values for each contextual attribute. The most relevant attribute is the one with the lowest variance. Depending on the most significant attribute found, we use the chosen value for that attribute. Then reduce the dataset based on the specified value to only include items with the most significant attribute as the selected value. CF is now applied to this compressed dataset. To predict the products, CF looks at the opinions of others with similar interests as you have had in the past. For our work, the Pearson Correlation Coefficient (PCC) is used to determine how similar one item is to other items. In this manner, we obtain the contextualized suggestion based on the attribute determined to be most significant to the user. Figure 1 shows the proposed approach architecture.

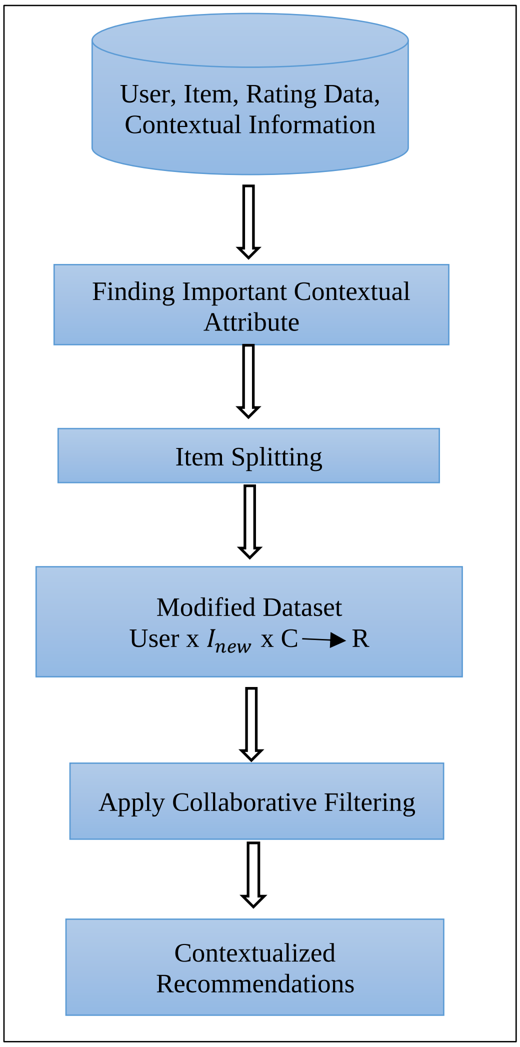


Figure 1: The proposed Contextual Pre-filtering.

3.1 Finding most important attribute and reduce the dataset

To identify the most important contextual attribute, we will compute the variance for each contextual attribute and then determine the most important attribute by taking the lowest variance for the contextual attributes.

The dispersion between the values in a dataset is measured as the variance. The dispersion, or variation, of data values increases with increasing variance. The following Equation (1) is used to compute the variance:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{N} \qquad (1)$$

Where x_i each item in the dataset, $\bar{x}$ is the mean of all the dataset's values, and N is the number of items in the dataset.

After computing the variance for each contextual attribute, determine the most important attribute, i.e., the contextual attribute with the lowest variance. Next, for the contextual feature that was determined to be the most significant, take the chosen value for that feature as ci, where i depend on feature. The dataset is reduced using a contextual pre-filter based on ci to only include records that have the value for ci, while all other values for user, product, and rating are left unchanged. The algorithm (1) shows contextual pre-filtering based on the most important attribute.

Algorithm (1): Contextual Pre-filtering Based on the Most Important Attribute
Input: Ratings for target user, LDOS-CoMoDa dataset. **Output:** Modified dataset.
Begin: **Step 1: Load** LDOS-CoMoDa dataset. **Step 2: Insert** user id. **Step 3: Get** contextual attribute for target user **Step 4: Calculate** the variances for each contextual attribute for target user. **Step 5: Sort** variances in descending order **Step 6: Choose low** variance as the most important attribute **Step 7: Calculate** the number of each value in most important attribute **Step 8: Sort** the number of each value in descending order **Step 9: Choose** the value with the most frequency **Step 10: Reduce** the dataset such that it only contains the most important components that have the value of the most important attribute **End**

Variance was chosen as a basic criterion for selecting attributes because it measures the extent of dispersion of values. Thus, we can know the user's preferences by measuring the variance for each contextual attribute within the user's preferences. Each contextual attribute has several values, which means that the contextual attribute with high variance has highly dispersed attribute values, while the contextual attribute with low variance has less dispersed attribute values. However, With the low variance, we can learn about user preferences by the most frequent value in this contextual attribute. Thus, we reduce the dataset to the basis of variance and the most frequently occurring value.

3.2 Collaborative filtering

Then, we utilize CF on the compressed dataset to predict the products that would interest customers currently based on customer product ratings. Equation (2), demonstrates how the item-based approach in the suggested system that uses PCC is required for calculating correlation between comparable goods. Utilizing the Equation (3), it is possible to determine a customer's predicted for a specific item i.

3.2.1 Building the rating matrix

After modifying the dataset, a data structure shown as a matrix was created to record the connections between various people and movies with their related ratings. From the modified dataset, extract UserID, ItemID, and Rating. The Rating Matrix (RM) can then be built. In RM, the columns correspond to the movie ID and the rows to the user ID. Each user's intersection with a movie had a cell value that held the corresponding rate that had been previously retrieved.

3.2.2 Measuring similarity

The next step after creating RM was to use the PCC to calculate the similarity between movies. The Items Similarity Matrix (ISM) was used in this model to store similarities between LDOS-CoMoDa movies. In this method, suggestions for the user are calculated by discovering objects that are related to other items. Equation (2) [19] has been used to show the PCC:

$$sim(k,l) = \frac{\sum_{u=1}^{m}(r_{u,k}-\bar{r}_k)(r_{u,l}-\bar{r}_l)}{\sqrt{\sum_{u=1}^{m}(r_{u,k}-\bar{r}_k)^2}\sqrt{\sum_{u=1}^{m}(r_{u,l}-\bar{r}_l)^2}} \quad \text{.........(2)}$$

Where m is the total number of people who rated both item k and item l, and $sim(k,l)$ denotes the degree of similarity between the two items. The average ratings for items k and l are $\bar{r}_k$ and $\bar{r}_l$, respectively; The terms $r_{u,k}$ and $r_{u,l}$respectively refer to the user u ratings of items k and l.

3.2.3 Prediction calculation

After the calculation of similarity, the prediction Equation (3) [20] was used to estimate the target user's prediction value and go on to the top N suggestions. A prediction on the product i of customer k is often made using the following formula:

$$P_{u,k} = \frac{\sum_{i=1}^{n}(r_{u,i}-\bar{r}_i)*sim(k,i)}{\sum_{i=1}^{n}|sim(k,i)|} \quad \text{........................(3)}$$

Where n is the total number of neighbors of item k, and $P_{u,k}$ is the prediction for user u on item k; By "$r_{u,i}$" we imply the user's evaluation of the item i, $\bar{r}_k$ is the average rating for the item k, $sim\ (k,i)$ represents the degree to which the item k and its neighbor i are similar; The term "$\bar{r}_i$" refers to the item i average rating.

In this way, we get the contextualized recommendation for the target user by implicitly extracting the preferences of the target user by applying the variance Equation (1) to contextual attributes, choosing the most important attribute, finding the most frequently used value in this attribute, and finally modifying the dataset based on this value, then applying CF.

The main steps of the suggested algorithm are listed below:

Step 1: Download the LDOS-CoMoDa from the website.
Step 2: Inter user ID.
Step 3: Finding the most important attribute and reducing the dataset.

a. Calculate the variance for each contextual attribute for the target user.
b. Take the lowest variance.
c. Calculate the most frequent value from the attribute with the lowest variance.
d. Reduce the dataset based on the most frequency value.

Step 4: Apply CF

a. Calculate the similarity between movies by using the PCC equation.
b. Calculate predictions to create a list of recommendations for the target user.

4 Experimental results

The major goal of the experiments is to evaluate the degree of accuracy of the suggestions provided by the proposed approaches. The CPBCF model was evaluated using different values of the nearest neighbors, which allowed us to choose the top-k neighbors among users who had given the same item a rating. It is possible to further narrow the neighborhood so that users have rated the item in the same contexts if contexts are taken into account. This provides a recommendation computation that is particular to the context. Subsection 4.1 describes the experimental setting; Subsection 4.2 describes the data; and Subsection 4.3 presents the metrics for system evaluation. The results are presented in subsection 4.4.

4.1 Experimental setup

Table 2 lists the particular parameters of the experimental setting used in this article.

Table 2: Experimental environmental factors.

Category	Parameter
The operating system	Windows 10
A programming language	Python
CPU	Core i7
Memory	8GB

4.2 Dataset

A context-aware dataset for movie recommendations is called LDOS-CoMoDa. It was published in 2011 on the www.lucami.org website. The dataset consists of 1232 items and 2296 rating records from 121 individuals, together with contextual information. Table 3 contains a description of the contextual information in the LDOS-CoMoDa dataset. Each record has a total of 30 variables, including 4 user attributes, 11 item attributes, 12 contextual attributes, a user, an item, and a rating (an integer value between 1 and 5).

Table 3: Description of LDOS-CoMoDa Dataset.

Contextual variable	No. of category	Description
Time	4	1-morning, 2-afternoon, 3-evening, 4-night
Day type	3	1-working day, 2-weekend, 3-holiday
Season	4	1-spring, 2-summer, 3-autumn, 4-winter
Location	3	1-home, 2-public place, 3-friend's house
Weather	5	1-sunny/clear, 2-rainy, 3-stormy, 4-snowy, 5-cloudy
Social	7	1-alone, 2-partner, 3-friends, 4-colleagues, 4-parents, 6-public, 7-family
EndEmo	7	1-sad, 2-happy, 3-scared, 4-surprised, 5-angry, 6-disgusted, 7-neutral
Dominant Emo	7	1-sad, 2-happy, 3-scared, 4-surprised, 5-angry, 6-disgusted, 7-neutral
Mood	3	1-positive, 2-neutral, 3-negative
Physical	2	1-healthy, 2-ill
Decision	2	1-user's choice, 2-given by other
Interaction	2	1-first, 2-n-th

4.3 Evaluation metrics

The quality metrics for recommendation have been chosen as precision, recall, F1-measure, and Normalized Discounted Cumulative Gain (NDCG), while the quality measures for prediction have been chosen as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). In addition, the computational complexity of the system was analyzed. Table 4 shows the confusion matrix that was used to calculate the value.

Table 4: Confusion matrix [19].

	Relevant	Irrelevant
Recommended	True Positive (TP)	False Positive (FP)
Not Recommended	False Negative (FN)	True Negative (TN)

1) Precision: the most important metric for determining how effectively an RS is performing [22]. The top k recommendations that are relevant to the consumer are listed [23].

$$Precision = \frac{(TP)}{(TP+FP)} \quad (4)$$

The greater value of this relationship indicates that the RS has made a more accurate suggestion [24]

2) Recall: presents how many of the customer's favorite items he is interested in, as well as how many of those items the system has actually suggested to him [22].

$$Recall = \frac{(TP)}{(TP+FN)} \quad (5)$$

The higher number indicates that the RS has the ability to arrange the items according to relevance and move them to the top of the final list [24].

3) F1-measure: which is the harmonic mean of precision and recall, is used to evaluate the accuracy and effectiveness of the results generated by the recommended RS.

$$F1 - measure = 2 * \frac{(Precision*Recall)}{(Precision+Recall)} \quad (6)$$

Due to accuracy and recall falling inside the [0, 1] range, an item with a high F1-measure value is considered to be more effective [24].

4) Normalized Discounted Cumulative Gain (NDCG): method to assess the grade of suggestions in order to calculate how much CARS learning has occurred. A list of suggested things with a greater NDCG value indicates that there are more related items placed higher in the list [12]:

$$NDCG = \frac{\sum_{k=1}^{n} G(u,n,k)D(k)}{\sum_{k=1}^{n} G*(u,n,k)D(k)} \quad (7)$$

where $G\ (u, n, k)$ is the gain associated with the k-th item in the ideal ranking of n size for u user, $G*\ (u, n, k)$ is the gain associated with the k-th item in the list L, and $D\ (k)$ is a discounting function [25].

5) Root Mean Square Error (RMSE): determines the probability of being incorrect when predicting a not-rated item for an active user [26]. This value must be lowered in order to more clearly illustrate the model's performance [27]. The equation is as follows:

$$RMSE = \sqrt{\frac{1}{N_P}\sum_{u,i}(P_{ui} - r_{ui})^2} \quad (8)$$

N_P refers to the total number of predictions, and P_{ui} and r_{ui} are the actual and expected ratings of how likely it is that the user would choose item I [25].

6) Mean Absolute Error (MAE): is a statistic that measures how effectively recommendation algorithms predict outcomes and is used in a similar way to RMSE [28]. It may be calculated using the formula provided in:

$$MAE = \frac{1}{|T|}\sum_{r_{u,i}\in T}\left|r_{u,i} - \hat{r}_{u,i}\right| \quad (9)$$

Where $r_{u,i}$ and $\hat{r}_{u,i}$ are, respectively, used to indicate the actual and predicted ratings of the item for the user u [29][30]. With a smaller MAE, the RS is better able to predict ratings [24].

7) Computational Complexity: refers to the amount of time an algorithm or system takes to complete the required tasks based on the size of the input data. In recommender systems, time complexity is measured to analyze the efficiency during different stages, such as calculating the similarity matrix, generating recommendations, and ranking the results [31]. For the system used in this research, the total time complexity can be represented by the relationship:

$$O(n^2.m + m.n.k + n.logn) \quad (10)$$

Where n is the number of items, m is the number of users, and k is the number of neighbors used in the recommendations.

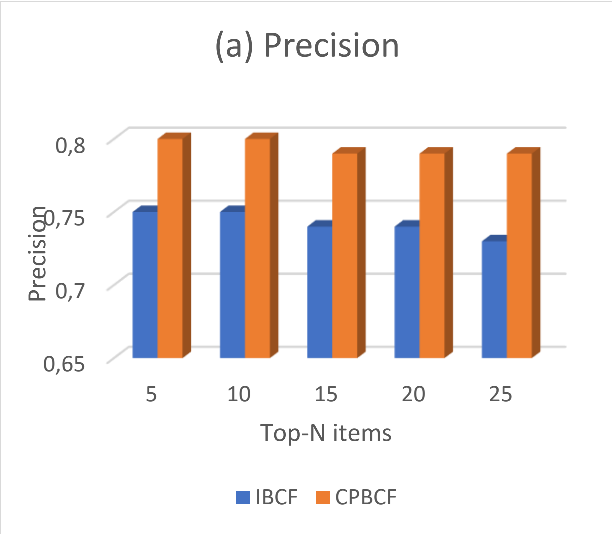
4.4 Results

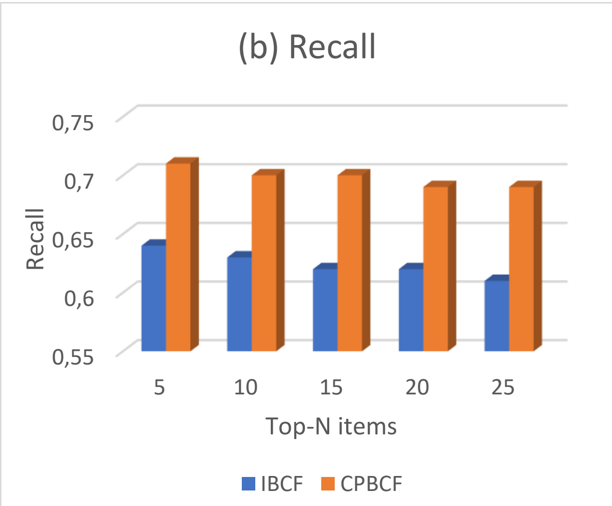
In the experiments, the performance of the suggested CPBCF with and without contextual pre-filtering was evaluated using the confusion matrix. The results showed that the suggested approach is capable of making product recommendations with higher precision, recall, and F1-measure than traditional IBCF. As for the NDCG scale, we notice a slight decrease in the results compared to the traditional method due to the reduction of data. For the suggested CPBCF model, Table 5 shows the precision, recall, F1-measure, and NDCG with different values of the nearest neighbors.

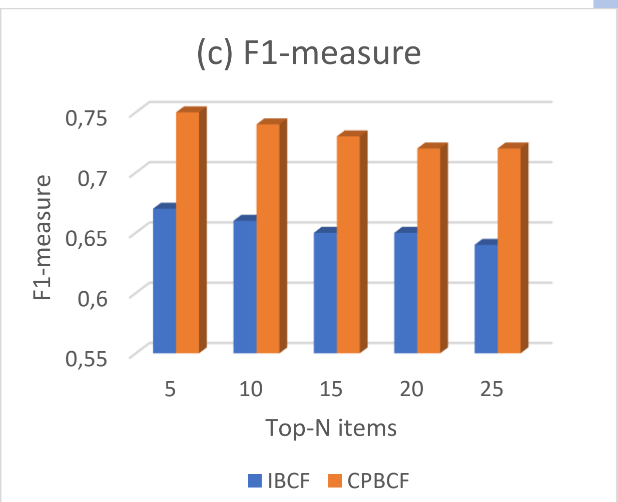
Table 5: Precision, Recall, F1-measure, and NDCG of RS with different number values.

Measure	Methods	Number of different nearest neighbor				
		5	10	15	20	25
Precision	IBCF	0.75	0.75	0.74	0.74	0.73
	CPBCF	0.80	0.80	0.79	0.79	0.79
Recall	IBCF	0.64	0.63	0.62	0.62	0.61
	CPBCF	0.71	0.70	0.70	0.69	0.69
F1-massure	IBCF	0.67	0.66	0.65	0.65	0.64
	CPBCF	0.75	0.74	0.73	0.72	0.72
NDCG	IBCF	0.49	0.48	0.47	0.47	0.47
	CPBCF	0.23	0.24	0.24	0.24	0.27

The algorithms' precision, recall, F1-measure, and NDCG are displayed in Figure 2. All methods' precision, recall, and F1-measure decrease while NDCG increases when the nearest neighbor grows.







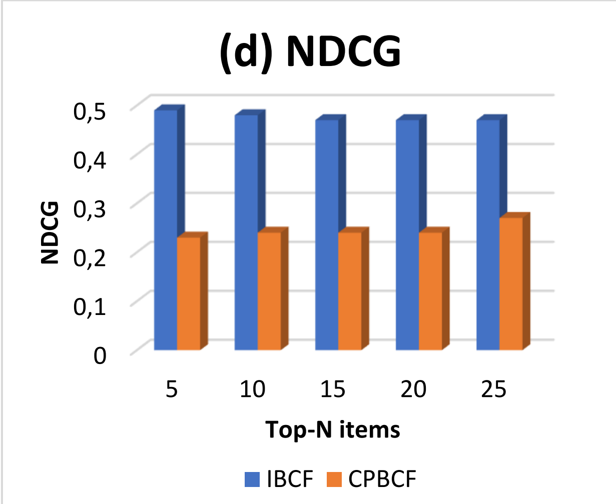


Figure 2: (a) Comparing Precision between IBCF with and without Contextual Pre-filtering, (b) Comparing Recall between IBCF with and without Contextual Pre-filtering, (c) Comparing F1-measure between IBCF with and without Contextual Pre-filtering, (d) Comparing NDCG between IBCF with and without Contextual Pre-filtering.

Some metrics are used to evaluate the suggested system. These measurements, known as MAE and RMSE, measure the variances between customer-suggested and actual preferences over time. According to the results of these measures, the CARS prediction results improve with decreasing error levels. Table 6 presents the MAE and RMSE results for the IBCF models with and without contextual pre-filtering prediction accuracy on the LDOS-CoMoDa dataset.

Table 6: MAE and RMSE of RS with different number values.

Measure	Methods	Number of different nearest neighbor				
		5	10	15	20	25
MAE	IBCF	0.034	0.035	0.035	0.037	0.040
	CPBCF	0.041	0.043	0.047	0.053	0.061
RMSE	IBCF	0.171	0.231	0.244	0.251	0.257
	CPBCF	0.232	0.261	0.319	0.357	0.393

The algorithms' MAE and RMSE are displayed in Figure 3. With k = 5, 10, 15, 20, and 25, different numbers of neighbors were used to evaluate the performance. We can see that the size of the neighborhood does have an impact on the accuracy of the forecast. The values of MAE and RMSE increased when the nearest neighbor increased.

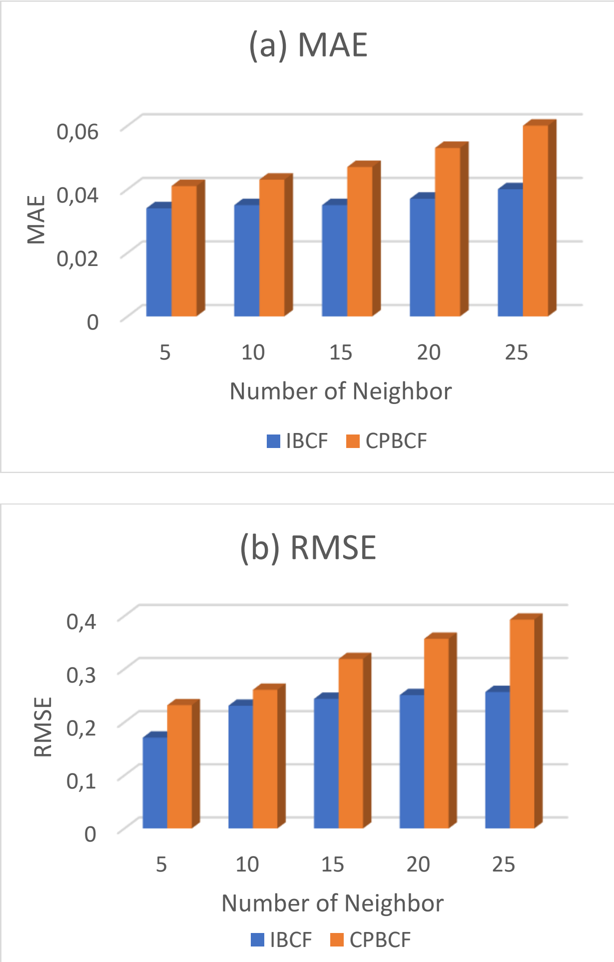


Figure 3: (a) Comparing MAE between IBCF with and without Contextual Pre-filtering, (b) Comparing RMSE between IBCF with and without Contextual Pre-filtering.

The complexity time of the efficiency analysis was measured during different stages, and the experimental results showed that the traditional method was more time-consuming compared to the CPBCF method, which benefited from data reduction before performing the calculations, which was positively reflected in the time performance of the system. Table 7 shows the complexity time with different values of nearest neighbor.

Table 7: Complexity time with different number values.

Measure	Methods	Number of different nearest neighbor				
		5	10	15	20	25
Complexity time	IBCF	4.925	5.057	5.12	5.716	5.868
	CPBCF	1.517	1.533	1.538	1.658	1.81

The complexity time of the methods is shown in Figure 4. Different numbers of neighbors were utilized for evaluating the performance with k = 5, 10, 15, 20, and 25. The use of CPBCF significantly improved the computational efficiency, making the method more scalable with big data.

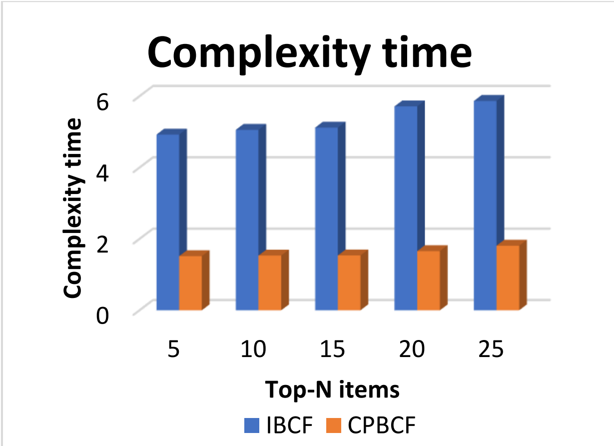


Figure 4: Comparing Complexity time between IBCF with and without Contextual Pre-filtering.

5 Discussion

In this work, the CPBCF model was created to incorporate contextual data into the recommendation process, hence addressing the issues associated with personalized suggestions. Compared to more conventional approaches, the results offer important new information about how well this approach enhances suggestion quality.

The results show that one of the most important steps in enhancing the recommendation process is pre-filtering the dataset using the variance of contextual factors. For the target user, the model successfully captures the most consistent relevant feature by determining which contextual characteristic has the lowest variance. The most frequent value of the attribute was used to filter the dataset, reducing noise and ensuring that only the most relevant data was used to provide recommendations.

The advantages of the suggested strategy were further demonstrated by the comparison between CPBCF and conventional IBCF. The enhanced precision, recall, and F1-measure of the CPBCF model demonstrate its ability to produce more accurate and pertinent recommendations. The pre-filtering procedure is responsible for this performance improvement since it ensures that the suggestions are based on an enhanced and contextually relevant dataset. These results support previous research, which highlights the importance of contextual information in recommendation systems.

A significant result from the tests was how neighborhood size affected the accuracy of recommendations. The F1-measure, precision, and recall were all adversely affected by the neighborhood size increase, even though it led to greater error rates (e.g., MAE and RMSE). Because they focus on individuals or objects that are more like the intended user, smaller, context-focused communities are consequently more successful. In addition, the NDCG results, although slightly lower, provide an interesting perspective. Reducing the dataset size due to the pre-filtering process resulted in a decrease in computational complexity, making the model more resource-efficient. This highlights the balance between improving recommendation accuracy and maintaining system efficiency.

The suggested CPBCF model suggests that contextual pre-filtering can improve recommendation systems in a number of domains, including movie suggestions and personalized services. The model improves the quality of the information by using contextual variables to produce high-quality suggestions in a scalable and effective manner.

Despite its positive results, the CPBCF model's emphasis on variance and most common values may make it unable to capture all contextual dynamics, especially in datasets with a high degree of variability. Future studies should look at sophisticated techniques for identifying important contextual elements, such as2 mutual information, association-based selection, or feature importance measures.

6 Conclusions

Everybody deals with RSs on a daily basis. The rapid expansion of knowledge and information available online has increased the importance of developing efficient and effective RSs. The suggested CPBCF model employs CARS to accomplish personalization, with each user having a variety of contexts that describe the situation and environment around them. Our model works better than the conventional RS since it performs better when compared on the basis of precision, recall, and F1-measure. The confusion matrix was used to evaluate the effectiveness of the proposed CPBCF with and without contextual pre-filtering. According to the results, the recommended method has more precision, recall, and F1-measure than standard IBCF in making product suggestions. In addition, the NDCG results showed a slight decrease, which is a result of reducing the data size through initial filtering, which reduced the complexity time and increased the system efficiency. This highlights the balance between improving recommendation performance and system efficiency at the same time. The effectiveness of the CPBCF model was evaluated using various numbers of neighbors. The size of the neighborhood has an effect on forecast accuracy. When the nearest neighbor grew, the values of the MAE and RMSE rose while the values of the F1-measure, recall, and accuracy decreased.

References

[1] N. F. AL-Bakri and S. H. Hashim, “Collaborative Filtering Recommendation Model Based on k-means Clustering,” Al-Nahrain J. Sci., vol. 22, no. 1, pp. 74–79, 2019, doi: 10.22401/anjs.22.1.10.

[2] A. H. N. Rafsanjani, N. Salim, A. R. Aghdam, and K. B. Fard, “Recommendation Systems: a review,” Int. J. Comput. Eng. Res., vol. 3, no. 5, pp. 47–52, 2013.

[3] F. T. A. Hussien, A. M. S. Rahma, and H. B. Abdulwahab, “An E-Commerce Recommendation System Based on Dynamic Analysis of Customer Behavior,” Sustain., vol. 13, no. 19, pp. 1–21, 2021, doi: 10.3390/su131910786.

[4] J. Anitha and M. Kalaiarasu, “Optimized machine learning based collaborative filtering (OMLCF) recommendation system in e-commerce,” J. Ambient Intell. Humaniz. Comput., vol. 12, pp. 6387–6398, 2021, doi: 10.1007/s12652-020-02234-1.

[5] R. Sharma, D. Gopalani, and Y. Meena, “Collaborative Filtering-Based Recommender System: Approaches and Research Challenges,” in 3rd International Conference on Computational Intelligence & Communication Technology (CICT), pp. 1–6, 2017.

[6] Z. Huang, D. Zeng, and H. Chen, “A Comparative Study of Recommendation Algorithms in E-Commerce Applications,” IEEE Intell. Syst., vol. 22, no. 5, pp. 68–78, 2007.

[7] S. Raza and C. Ding, “Progress in context-aware recommender systems - An overview,” Comput. Sci. Rev., vol. 31, pp. 84–97, 2019, doi: 10.1016/j.cosrev.2019.01.001.

[8] H. K. Omar, M. Frikha, A. K. Jumaa, “Improving Big Data Recommendation System Performance Using NLP Techniques With Multi-attributes,” Informatica, vol. 48, no. 5, 2024, doi: 10.31449/inf.v48i5.5255.

[9] L. Baltrunas and F. Ricci, “Experimental evaluation of context-dependent collaborative filtering using item splitting,” User Model. User-adapt. Interact., vol. 24, pp. 7–34, 2014, doi: 10.1007/s11257-012-9137-9.

[10] Y. Zhang and L. Wang, “Some Challenges for Context-aware Recommender Systems,” 5th Int. Conf. Comput. Sci. Educ., pp. 362–365, 2010.

[11] K. V. Rodpysh, S. J. Mirabedini, and T. Banirostam, “Employing singular value decomposition and similarity criteria for alleviating cold start and sparse data in context-aware recommender systems,” Electron. Commer. Res., pp. 1–27, 2021, doi: 10.1007/s10660-021-09488-7.

[12] Y. Zheng, “Context-Aware Collaborative Filtering Using Context Similarity: An Empirical Comparison,” Information, vol. 13, no. 1, pp. 1–18, 2022, doi: 10.3390/info13010042.

[13] G. Adomavicius and A. Tuzhilin, “Context-Aware Recommender Systems,” in Recommender Systems Handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, pp. 217–253, 2010, doi: 10.1007/978-0-387-85820-3_7.

[14] A. Bozanta and B. Kutlu, “Developing a Contextually Personalized Hybrid Recommender System,” Mob. Inf. Syst., pp. 1–13, 2018, doi: 10.1155/2018/3258916.

[15] M. Singh, H. Sahu, and N. Sharma, “A Personalized Context-Aware Recommender System Based on User-Item Preferences,” in InData Management, Analytics and Innovation: Proceedings of ICDMAI, vol. 2, pp. 357–374, 2019, doi: 10.1007/978-981-13-1274-8_28.

[16] J. Silva et al., “Multi-dimension Tensor Factorization Collaborative Filtering Recommendation for Academic Profile,” Adv. Neural Networks, pp. 200–209, 2019, doi: 10.1007/978-3-030-22808-8_21.

[17] I. M. Al Jawarneh et al., “A Pre-Filtering Approach for Incorporating Contextual Information Into Deep Learning Based Recommender Systems,” IEEE Access,

vol. 8, pp. 40485–40498, 2020, doi: 10.1109/ACCESS.2020.2975167.
[18] Z. El Yebdri, S. M. Benslimane, F. Lahfa, M. Barhamgi, and D. Benslimane, "Context-aware recommender system using trust network," Computing, vol. 103, no. 9, pp. 1919–1937, 2021, doi: 10.1007/s00607-020-00876-9.
[19] Q. Li and B. M. Kim, "Clustering Approach for Hybrid Recommender System," IEEE/WIC Int. Conf. Web Intell., pp. 33–38, 2003.
[20] Z. Zhu, "Personalized Recommendation Algorithm Based on Data Mining and Multi-objective Immune Optimization," Informatica, vol. 48, no. 19, 2024, doi: 10.31449/inf.v48i19.6546.
[21] R. G. Lumauag, A. M. Sison, and R. P. Medina, "An Enhanced Recommendation Algorithm Based on Modified User-Based Collaborative Filtering," in 4th International Conference on Computer and Communication Systems (ICCCS), pp. 198–202, 2019.
[22] T. Li, Y. Li, and C. Y. Phoebe, "Novel Multidimensional Collaborative Filtering Algorithm Based on Improved Item Rating Prediction," Sci. Program., pp. 1–14, 2021, doi: 10.1155/2021/2592604.
[23] S. G. K. Patro, B. K. Mishra, S. K. Panda, R. Kumar, and H. V. Long, "Knowledge-based preference learning model for recommender system using adaptive neuro-fuzzy inference system," J. Intell. Fuzzy Syst., vol. 39, no. 3, pp. 4651–4665, 2020, doi: 10.3233/JIFS-200595.
[24] X. Bai, M. Wang, I. Lee, Z. Yang, X. Kong, and F. Xia, "Scientific Paper Recommendation: A Survey," IEEE Access, vol. 7, pp. 9324–9339, 2019, doi: 10.1109/ACCESS.2018.2890388.
[25] S. Chakraborty, M. S. Hoque, N. R. Jeem, M. C. Biswas, D. Bardhan, and E. Lobaton, "Fashion Recommendation Systems, Models and Methods: A Review," Informatics, vol. 8, no. 3, pp. 1–34, 2021, doi: 10.3390/informatics8030049.
[26] C. K. Raghavendra and K. C. Srikantaiah, "Similarity Based Collaborative Filtering Model for Movie Recommendation Systems," 5th Int. Conf. Intell. Comput. Control Syst. ICICCS, pp. 1143–1147, 2021.
[27] Rohit, S. Sabitha, and T. Choudhury, "Proposed Approach for Book Recommendation Based on User k-NN," Ina. Comput. Comput. Sci. Proc. ICCCC, vol. 2, pp. 543–558, 2018, doi: 10.1007/978-981-10-3773-3_53.
[28] M. Schedl, H. Zamani, C. W. Chen, Y. Deldjoo, and M. Elahi, "Current challenges and visions in music recommender systems research," Int. J. Multimed. Inf. Retr., vol. 7, pp. 95–116, 2018, doi: 10.1007/s13735-018-0154-2.
[29] N. F. Al-Bakri and S. H. Hashim, "Reducing Data Sparsity in Recommender Systems," J. Al-Nahrain Univ. Sci., vol. 21, no. 2, pp. 138–147, 2018, doi:10.22401/jnus.21.2.20.
[30] D. H. Hameed, R. F. Hassan, "A review of recommendation system: Techniques, limitation and evaluation", 4th International Scientific Conference of Alkafeel University (ISCKU 2022) AIP Conf. Proc. 2977, 030002-1–030002-12.
[31] C. C. Aggarwal, "Recommender Systems, "cham springer International Publishing, 2016, doi:10.1007/978-3-319-29659-3.