Hindawi
Mobile Information Systems
Volume 2022, Article ID 2854741, 18 pages
https://doi.org/10.1155/2022/2854741

Research Article

Rating-Based Recommender System Based on Textual Reviews Using IoT Smart Devices

Muqeem Ahmed,[1] Mohd Dilshad Ansari ,[2] Ninni Singh ,[3] Vinit Kumar Gunjan ,[3,4] Santhosh Krishna B. V. ,[5] and Mudassir Khan[6]

[1]*Department of CS & IT, MANUU (Central) University, Hyderabad, India*
[2]*Department of Computer Science & Engineering, CMR College of Engineering & Technology, Hyderabad, India*
[3]*Department of Computer Science & Engineering, CMR Institute of Technology, Hyderabad, India*
[4]*Department of Computer Science and Engineering, University of Science and Technology Chattogram, Chattogram, Bangladesh*
[5]*Department of Electronics and Communication Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India*
[6]*Department of Computer Science, College of Science & Arts, King Khalid University, Tanumah, Abha, Saudi Arabia*

Correspondence should be addressed to Vinit Kumar Gunjan; vinit.gunjan@ustc.ac.bd

Received 11 March 2022; Revised 28 May 2022; Accepted 7 June 2022; Published 11 July 2022

Academic Editor: M. Praveen Kumar Reddy

Recommender system (RS) is a unique type of information clarification system that anticipates the user's evaluation of items from a large pool based on the expectations of a single stakeholder. The proposed system is highly useful for getting expected meaning suggestions and guidance for choosing the proper product using artificial intelligence and IoT (Internet of Things) such as chatbot. The current proposed technique makes it easier for stakeholders to make context-based decisions that are optimal rather than reactive, such as which product to buy, news classification based on high filtering views, highly recommended wanted music to choose, and desired product to choose. Recommendation systems are a critical tool for obtaining verified information and making accurate decisions. As a result, operational efficiency would skyrocket, and the risk to the company that uses a recommender system would plummet. This proposed solution can be used in a variety of applications such as commercial hotels OYO and other hotels, hospitals (GYAN), public administrative applications banks HDFC, and ICICI to address potential questions on the spot using intelligence computing as a recommendation system. The existing RS is considering a few factors such as buying records, classification or clustering items, and user's geographic location. Collaborative filtering algorithms (CFAs) are much more common approaches for cooperating to mesh the respective documents they retrieved from the historical data. CFAs are distinguished in plenty of features that are uncommon from other algorithms. In this existing system classification, precision and efficiency and error rate are statistical measurements that need to be enhanced according to the current need to fit for global requirements. The proposed work deals with enhancing accuracy levels of text reviews with the recommender system while interacting by the numerous users for their domains. The authors implemented the recommender system using a user-based CF method and presented the significance of collaborative filtering on the movie domain with a recommender system. This whole experiment has been implanted using the RapidMiner Java-based tool. Results have been compared with existing algorithms to differentiate the efficiency of the current proposed approach.

1. Introduction

RS is the most significant technique that processes the resources of online businesses to generate useful suggestions for items that match users' interests and needs. Items/products can be any of the resources of businesses that are consumed by users online, such as books, CDs, or news. RSs produce certain decisions to make suggestions of which items to purchase or what news to read online. Generally speaking, the mechanism of a recommender system is based on the investigation of historical data about each user's feedback about items used previously to recommend a list of estimated preferred items that the user may like in his next purchasing operations. Users are overwhelmed by a huge number of choices of items that websites give. Therefore, RS serves to filter alternatives in a personalized manner and recommend the items that will most likely attract the user's consideration. The RS is considered an active mechanism that provides personalized recommendations of information. RS allows the end users for the online trades in opinion sharing about any product endorsements over e-commerce applications such as Paytm, Flipchart, Alibaba, Amazon Prime, and eBay [1–3]. Traditionally, RSs have two approaches: collaborative filtering (CF) RSs and content-cased (CB) RSs [4–6]. CF-related systems make use of historical rating information to assess the end stakeholders and their neighbors. In this system [7, 8], the future interests of users are determined based on high similarity history profiles. The CB system [9, 10] examines features that appear in entity related content that users experienced in prior time; subsequently, it advises recommended items based on applicable features to the users. Earlier authors or those who are researching on recommendation systems have accommodated reliable connections [11–13] based on end-user reliable relations to improve conventional RSs by advising further products. In the recent past, various authors have employed distinct techniques towards RS over online reviews about products and all techniques used NLP to determine sentiment emotions in text reviews that are shared by end-users about products [14–16]. According to the RS, user profiles lack in sparsity as very few users would pose reviews towards entities, eventually affecting the computational efficiency of the CF and CB approaches. Online relations in RSs are not precisely likeness of social associations [17]. The authors have employed many techniques for machine learning applications for various domains to resolve current problem-related aspects. This paper is an attempt to enlighten the importance of textual reviews for the recommender system. 78% of data are in an unstructured way, so there is much need to pay attention towards the textual reviews to utilize that resource for the recommender system and also implement the user-based collaborative filtering (CF) which is a widely used method for RS and assess the performance of the movie rating with the help of RapidMiner tool [18–20].

2. Review of Literature

Recommender systems have appeared as an independent and significant research domain right from the beginning on recommendation methods introduced as part of research. In the past decade, retailers have integrated the service of providing personalized recommendations as an important part of their systems. These services include the popular e-commerce retailers, Amazon.com and eBay (eBay.com), as well as the movie and entertainment industry, MovieLens (movielens.org), Netflix (netflix.com), and Film Trust (trust.mindswap.org/Film Trust), and the music industry, such as CDNOW (cdnow.com), Ringo (ringo.com) [20–22], LastFm (last.fm), and Pandora (pandora.com), in addition to pictures and photo-sharing systems, such as flickr.com, expertise finder systems, such as LinkedIn, and news recommendation sites, such as Google News [20]. Nowadays, modern e-commerce systems highly require recommender systems as a useful tool to increase their profit, similar to the model followed by Amazon.com. The next section outlines business motivations that encourage commercial platforms to exploit recommender systems. Reference [23] emphasized a collaborative RS for filtering for user profiles and characteristics of products along with classification techniques. Reference [24] focused on current progress of enhanced recommendation approaches and a subset of techniques for making collaborating filtering as part of CFs. Reference [25] has done the state of art survey to decide about the best way to choose different factors of algorithms in recommendation systems. Reference [26] diagnosed about various algorithms towards hybrid approaches in recommended systems and identified research gaps to enhance features of current RSs in different domains and various languages. References [27–30] have focused on solutions with the help of machine learning techniques and natural language processing techniques and cold start recommendation systems functionalities in terms of scalability factors.

Shani et al. [30] developed MDP (Markov decision processes) based recommender system; a predictive model as well as the solution and update algorithm was used. The actual performance was tested on a commercial site. Similarly, Walek et al. [31] proposed a monolithic hybrid recommender system. This recommender system uses collaborative filtering (based on the SVD algorithm), content-based system and also uses a fuzzy expert system. The developed RS is appropriate to recommend movies. The performance of the work is up to 80%, which is better compared to the other RSs.

Recommender system usually has three input elements: (1) users, (2) items, and (3) feedback, where users provide their feedback about items. Feedback would be expressed in terms of ratings, which is the most common way in which users describe their opinions about entities. Let us assume that users of a system are represented as U, user $u \varepsilon U$, and N indicates a total number of end-users in set U. Consider the set of items I, where item $i \varepsilon I$ and M indicates the total number of items in set I. The authors considered the item that is rated by one end-user using the notation iu and the user who rated an item as ui. Every user and item are represented in the systems by a unique index to identify them clearly in the system. The system's input elements are then transferred to a user-item rating matrix, which has the space of $N \times M$. We assume that g is the utility function that computes the level of interest of user set U in item set I:

$$g\colon U * I \longrightarrow \widehat{R}. \tag{1}$$

Here $\widehat{R}$ indicates a recommended set of items in a specified numerical range. More specifically, we move now to find the set of items $i \varepsilon I$ which maximizes the user utility for each user $u \varepsilon U$ [5] as follows:

$$i_u = \arg Maxi \in Ig(I, u). \tag{2}$$

Most of recommender systems' core engines try to extrapolate the utility g to build a model that can estimate a user's judgment towards unknown items using already known feedback on items used before according to certain criteria to test the performance of RS, such as measuring the error in the estimated ratings. After the model has been built, unknown ratings can be estimated and provided to users in an ordered set. Researchers use machine learning algorithms, approximation theory, and some heuristics for the prediction step. Next, we discuss how users' profiles and items' profiles can be built and we also describe the types of feedback that can be considered within recommender systems.

2.1. Classification Recommendation Methods

2.1.1. Content-Type Recommendation. Content-type recommendation method is also known as content-based (CB) recommendation method which is used for the explanation of items to shape item representations and end-user outlines. CB methods analyze the content/description in user profiles based on the items they liked before; then they match up this content with similar content of unseen items. In other words, these methods generate recommendations by examining the features of items, such as the item textual descriptive information, and testing regularities that may occur in this content to recommend similar items in the future. Many current CB recommenders build recommendations on items that contain textual information, including websites (URLs), books, documents, and news. For instance, for book recommendation related applications, to suggest appropriate books to a particular user, this kind of recommender would analyze the consistencies between the books that the active user has valued extremely in the earlier (specific authors, types, subject matters, etc.). Afterwards, records which are great degree of likeness to which the user preferred would be suggested. The CB method has its own roots from IRS [32]. One step which has been enhanced over conventional information recovery methods is the best use of end-user profiles that consist of information regarding items a user has favored and liked. The user's profile can be built by observing the users' transaction behavior related to certain items [33–35]. Usually, pri is an item towards profile, a collection of characteristics of an item i described in pri. This set of features is extracted to be incorporated into the recommendations process as appropriate. As specified before, CB schemes are planned to ultimately recommendation-based text; the features in this systems are labelled in keywords. If a system indorses web pages to end-users, it may represent the web pages content with the most important keywords. The significance of word k in document d is defined as a weight $w_{d,k}$ that can be measured by several methods.

One which is frequently used statistical measures for assigning keyword weights about Term Frequency or TF-IDF [5]. Assuming *freqk*, d is the number of times keyword k_i occurs in document d_j; then, the frequency term is TFk; d is the term frequency or the normalized frequency of keyword ki in document djwill be as follows:

$$TF_k = freqk, \frac{d}{\max freqk}, d, \tag{3}$$

where the max freqK, d is calculated over all the frequencies in d of all keywordsK. In fact, term frequency (TF) in many documents tends not to be a good indication of relevant documents and nonrelevant ones as it treats all words equally in terms of importance. It is obvious that some keywords appear more commonly in certain topics; for example, the word program is included in almost every document about programming, so this word has a little effect when determining the relevance of certain documents. Hence, a statistical measurement of IDF is used together with TF to reduce the effect of terms that have less importance when computing the weights. The inverse frequency of keyword k_i in a document will be

$$IDF_k = \frac{\log N}{n_k}. \tag{4}$$

Here, N indicates total number of documents which can be provided to an end-user as a recommendation while n_k is a subset of N, where the keyword k appears. Hence, to compute the weight for keyword k_i in document d_j, the calculation should combine both $TF_{k,d}$ and IDF_k as follows:

$$W_{k,d} = TF_{k,d} * IDF_k. \tag{5}$$

Then, the profile in document d_j is defined as

$$\mathrm{Pr}_d = (w1, w2, \ldots\ldots, wK). \tag{6}$$

As mentioned earlier, recommenders that use the CB approach suggest items similar to those that a user liked previously. Hence, the level of similarity is computed between two types of profiles; the first type is item profiles which contain item features as weights of keywords, and the second type is user profiles which contain weights of keywords of items seen or rated by the user in the past. For instance, if a user has read online documents on a specified topic of networks, subsequently, the CB methods would provide other articles related to networks to that user. This is the case because these articles will contain terms related to networks (e.g., router, protocol, and "wireless") as opposed to articles on other topics. Therefore, recommenders use similarity measures to identify higher similarity values between a user's profile and those articles profiles that have network terms with higher weights. To assign similarity weights between users' profiles and items' profiles, CB recommenders apply similarity techniques, such as a vector cosine similarity measure, defined as follows:

$$Sim(u, d) = \cos(wu, wd) = \frac{wu * wd}{\|wu\|_2 \, wd\|_2}. \tag{7}$$

In vector cosine similarity algorithm, the profile of user u and document d will be treated as two vectors wu and $w\ d$.To calculate the resemblance between the two vectors, the method will measure the cosine of the angle between them. Various techniques are based on a model learned from data related to users and items rather than the use of heuristic approaches. Various machine learning algorithms are used for CB recommenders, such as classification and clustering algorithms presented in [36–38]. For example, the authors of [9] rated a set of web pages in two categories: relevant and irrelevant by the user. They used users' profiles to learn their interest in different pages by determining 128 informative words as features. They then used a Naive Bayesian (NB) supervised classifier to categorize unrated web pages by estimating the probability that a page p_j belongs to a particular class C_i (relevant or irrelevant), given the set of keywords $k1$, j,:::, kn, j, using the probability function $P(Cijk1, j\ \&\ldots\&kn, j)$. To overawe the disadvantage of poor prediction in the case when some users are unwilling to rate many pages, some initial knowledge regarding the user's benefits can be used.

(1) Challenges in CB Recommendation. CB techniques have many advantages: First, these methods are based only on users' preferences presented in ratings to build users' profiles and do not require input from other users' ratings to build these profiles. Second, the explanation advantage allows the system to exploit the list of keywords to justify why a specific item has been recommended. Also, systems based on CB can recommend novel items that have not received any ratings so far, a feature that does not apply to other techniques. Despite all these advantages, CB techniques still exhibit several limitations [5, 26]:

(i) Limited content analysis: Automatic extraction of features to describe items in a system will be much harder to implement when the data are graphical imageries, as well as acoustic and audiovisual streams. Information retrieval techniques achieve good results in mining features from script documents; however, when describing documents by their most important keywords, CB techniques cannot differentiate between high-quality articles and badly written ones when the same keywords are used [19].

(ii) Overspecialization: The recommended items will be limited to what the end-user has valued in the past. In other terms, only items matching the user's built profile of preferences with a high score of similarity will be recommended. Moreover, this overspecialization drawback is not only seen with CB recommenders in cases where they could not suggest items which are dissimilar from what end-user has experienced before, but, in convinced cases, very similar substances should not be suggested by the system.

2.1.2. Collaborative Filtering Recommendation. CF is an initial recommendation scheme, including such tools as the Tapestry system [38], GroupLens system [39] and Video Recommender [40]. CF techniques work by building profiles of customers' preferences [41]. The discussion on CF recommender system is as follows.

The following discussion is related to Table 1, which shows five users' profiles represented as rows in a user-item rating matrix $U1$,$U2$,$U3$,$U4$,$U5$. The columns represent items $I1$, $I2$, $I3$, and $I4$. Ratings profiles of users are

Table 1: An example of a user-item matrix.

	$I1$	$I2$	$I3$	$I4$
$U1$	4	φ	5	5
$U2$	4	2	1	φ
$U3$	5	φ	φ	4
$U4$	5	3	φ	φ
$U5$	φ	3	2	1

$$pr1 = [4, \varpi, 5, 5],\ pr2 = [4, 2, 1, \varpi],\ pr3 = [5, \varpi, \varpi, 4],$$
$$pr4 = [5, 3, \varpi, \varpi] \text{ and } pr5 = [\varpi; 3, 2, 1]. \quad (8)$$

For example, let us consider $U1$ as an active user and, by looking at his/her profile, we can see that $U1$ did not rate $I2$ and the system's objective is to decide whether $I2$ is a convenient suggestion to $U1$. Only three users $U2$, $U4$, and $U5$ have used $I2$ and they have certain opinions about it. However, $U2$ and $U4$ have close tastes in some items to $U1$. Therefore, it is better to consult their tastes to estimate whether item $I2$ is a good or bad suggestion for $U1$. By contrast, $U5$ has dissimilar tastes to $U1$ and, hence, the rating by $U5$ will have a very negligible effect on the recommendation decision, or, rather, the opinion of $U5$ will be ignored. Also, $U3$ will be excluded from the recommendation process as this user has not used and rated $I2$ although he/she has similar tastes to $U1$. Collaborative filtering algorithms can be classified into two main subcategories, namely, memory-based and model-based algorithms.

(1) Challenges in CF Recommendation. CF recommenders have several useful points when they are applied. One of the most important points is that they can provide recommendations using only the direct data source which is the rating profiles. Furthermore, the more the users respond to the recommendations, the better the systems can adapt to users' tastes over time; that is, rich users' profiles imply better quality recommendations. The most challenging aspects of using CF approaches, which the literature pointed out and efforts have been made to try and overcome, include the following: First is the level of prediction accuracy: most of the recommender models in the literature seek high levels of prediction accuracy. Researchers use the accuracy aspect to provide proof of how successful their proposed recommendation approaches are in providing predictions that are close to users' tastes, as stated above in [42].

Second is cold start problem: both CB and CF RSs lack from the cold-start difficulty. To provide accurate recommendations, users' preferences should be analyzed by collecting ratings that are not the case for new users who have recently entered the system; hence, this may lead to poor recommendation results. Similarly, the problem of cold start also exists in the case of new items. Recommendation systems are updated regularly with new items. To incorporate these new items in the recommendation process, these items should be valued first by a substantial number of users.

Third is the problem of sparsity: a common observation is that the number of items that people rate in slight RSs is minor in contrast to the total number of items that need rating prediction. Ultimately, the maximum number of end-users' ratings affects the success of collaborative recommenders. An example, over movie recommendation collaborative system, when a film has been valued by specific people, even if rated highly, this movie would not be recommended extensively. Furthermore, a common issue with users who have uncommon taste equated to the application community is that the schemes might not be allocate similar users [43–45].

2.1.3. Memory-Based Collaborative Filtering. Memory-based or neighborhood techniques [28, 31] utilize end-user item related medium to calculate recommendation predictions for unseen items. These systems identify like-minded users or so-called neighbors or nearest-neighbors. Generally, memory-based recommenders work according to the following steps:

(i) First, memory-based recommenders ask users to rate some items to be able to recognize their taste.

(ii) Second, statistical techniques are applied to define the user's neighbors who shared similar prior preferences.

(iii) Third, after defining like-minded users, the systems provide predictions of ratings to those items that have not been seen by the user and then provide recommendations accordingly.

In step two, a measurement is made use of computing the resemblance between two profiles. The close neighbors can be allocated by (1) defining a certain number of neighbors N; (2) predefining a suitable threshold, and hence only neighbors whose similarity level exceeds the threshold are incorporated in the recommendation encounter; and (3) excluding neighbors who have highly dissimilar tastes reflected by a negative similarity degree. In step three, the systems can choose to provide the recommendations as a score of rating and the rating values will be expressed using the same scale used to express opinions, for example, a scale from 1 to 5. Another different way of recommendation would be to yield a list of Top N recommendation items. In memory-based techniques, prediction aimed at an energetic u about an item i denoted as $r(u,i)$ is computed as aggregation of ratings of neighbors who have previously rated i.

$$R_{u,i} = aggra \in Neig\left(r_a, i\right). \tag{9}$$

Here, a is the neighbor of u who rated i and $Neig$ refers to the set of neighbors, where $Neig\ C\ U$ and $|Neig| = N$. Examples of some aggregation functions are given below:

$$ru, i = \frac{1}{N} \sum a \in Neig(ra, i). \tag{10}$$

This simple average is divided by the number of neighbors N or $Neig$. The most shared accumulation method is to calculate the anticipation as a weighted sum as shown below:

$$r\,u, i = \frac{\sum a\varepsilon Neig\, sim(u,a)(r\,a,i)}{\sum a\varepsilon Neig|sim(u,a)|}. \tag{11}$$

sim here refers to the similarity degree between an active user u who needs recommendations and another user a. Therefore, a large weight of sim indicates that u and a are very similar to each other and consequently the rating (ra,i) will participate more in the prediction of ru,i. From Equation (11), the problem is that this computation does not consider the fact that users may express their ratings using the score gage contrarily. The accustomed biased sum, discussed in Chapter 6, is extensively used to discourse this limitation. In the current method, rather than using complete principles of ratings, the prejudiced sum uses their deviances from a middling rating of the equivalent user.

Similarity computation is a primary element in memory-based methods to compute different levels of resemblance between particular users; a heuristic utility of similarity should be defined as sim. To measure this resemblance among two users, an active user u and user a denoted as the utility sim(u,a), this function illustrates the distance between the two users. The closer the users a and u are, the more weight will be given to the predicted ratings for user u in the prediction process. One popular approach to calculate the resemblance weight is the Pearson correlation coefficient (PCC) [46–50]:

$$sim(u,a) = \frac{\sum i\varepsilon Ia,u(r\,a,i - *r)}{\sqrt{\sum i\varepsilon Ia,u(r\,a,i - *r)2 \sum i\varepsilon Ia,u\,(r\,u,i - *u)2}}, \tag{12}$$

where ru,i is the rating worth obtained from an active end-user u to respective item i, while $^*ru$ indicates the average of all ratings obtained by user u. Similarly, the rating obtained by neighbor a to the same item i is ra,i and the regular of all scores given by a is $^*ra$. sim will be computed over the set of the corated items represented by Iu,a. Furthermore, certain heuristic similarity measures to overcome the cold start problems are proposed in [29], while [42] presented a similarity metric using a prior stage, in which a genetic algorithm generated weights which are dependent mainly on the nature of the dataset provided from each recommender system.

2.1.4. Model-Based Collaborative Filtering. CF makes use of ML techniques to make intelligent predictions. Building models algorithms includes three main steps:

(i) The designed models first learn the pattern of collected users' ratings in a training dataset.

(ii) The designed models are tested and all the needed parameters are tuned to satisfy the problem requirements, for example, minimizing the absolute squared error.

(iii) After building reliable and tested models, ratings can be predicted and recommendations can be provided to the desired users using these models.

Researchers in the recommendation area have introduced different learning models based on machine learning algorithms.

Classification. The authors in [51–53] used simple Bayesian CF algorithms in a collaborative filtering prediction process. Using a Naive Bayesian (NB) strategy, they assumed that the features are self-governing, for considering class, to compute the probability of a given class with the rest of the mechanisms. The anticipated class will be classified when the class has a high probability as follows:

$$r(u,i) = E(r(u,i)) = \sum c\varepsilon Classset\, P(r(u,i), \\ = c|r(u,j), j|Iu), \tag{13}$$

where c represents the rating class from the scale used Classset and j is an item that has been seen and rated by u and belongs to the set of all items Iu which are rated by an individual. In the case of a class that has missing features, the model computes the probability and classification over observed data. In the rating matrix, they converted multiclass data to binary class data as a Boolean feature vector for simplicity. However, this method caused limited scalability and loss of multiclass data. Real-world problems include multiclass data; therefore, researchers tried to improve the Bayesian CF algorithm to adapt to multiclass data. The authors in [54–56] conducted empirical experiments that showed that Bayesian CF can be better scalable and less time-consuming in the process of prediction; however, these can have worse predictive precision from the Pearson correlation CF algorithm.

Clustering. Clustering-based CF algorithms have also been used to improve prediction quality. The basic idea behind clustering techniques is to assign a cluster to every similar group of data, a group of users that have partly similar tastes. Cluster members would be seen as like-minded neighbors. Sarwar et al. [81] discussed the limitations of CF techniques and presented clustering-based algorithms to enhance speed performance. They applied two-phase algorithms: (a) clustering the user-item rating database to N partitions and (b) using memory-based CF algorithms to estimate recommendations for every user within the clusters based on only preferences from cluster members. An experimental study showed that making recommendations predictions within smaller clusters, using k-means algorithms, improves scalability in clustering techniques when compared with classical CF techniques; they reduce the number of neighborhoods to be tested due to the static precomputed clusters, and, as a result, the online prediction process becomes much faster [57]. The experiment in Sarwar et al.'s clustering methods presented two observations. First, clustering algorithms showed lower prediction quality in comparison with the basic CF approach. Furthermore, it was evident that as the number of clusters increased, the prediction error also increased. An explanation of this can be that the increased number of clusters may lead to smaller cluster sizes, therefore resulting in an insufficient number of neighbors to create a representative opinion about a particular item. Also, clustering techniques have the limitation that users clustered to a single group may not receive recommendations outside the cluster's taste trend, and this often causes less-personal recommendations and most often worse accuracy than memory-based algorithms [58–70].

Regressions. Regression-based CF algorithms are also presented in the literature to approximate users' ratings. Due to the numerical nature of the rating data in the real world, regressions models can contribute to predicting numerical values. Assume that $X = (x1, x2, \ldots, xm)$ is a set of random ratings where $x\ C\ Iu$.

$$Y = AX + E, \tag{14}$$

where A is $m \times k$ matrix, k is the k-dimensional rating space, and E signifies the noise in user preferences. Predicted matrix Y is $m \times n$, where Yu,i is the rating of end-user u to item i. The study reported in [38] proposed a collection of linear models to search for similarities between items. This regression approach succeeded in combining linear models to compute score predictions for a specific active end-user. To estimate the factors of the linear regression purpose, the authors used a least-squares metric. They showed that their approach offers good performance in addressing sparsity, a common problem in CF methods.

Matrix Factorization Models. Recently, several matrix factorization (MF) approaches have also been proposed [70–79]. The idea of implementing MF models has widely attracted researchers because of two properties: (1) an attractive level of accuracy and (2) sufficient scalability over large datasets. The general idea behind MF is modeling both users and items as an inner product and producing a joint space of latent features space of a specific level of dimensionality. MF models infer hidden structures causing the ratings interaction among users and items in the user-item rating matrix. These features explain how one user may rate an item and the model uses these features to approximate the rating matrix into a low-rank one. In the movie recommender, features may measure how much a given user is interested in a given movie. More formally, the rating is computed by the subsequent perdiction formulation:

$$\widehat{r_{u,i}} = \mu + bi + bu + qiT\,pu. \tag{15}$$

The model parameters bi and bu are cultured by minimizing squared error. μ is the average overall ratings. For example, the approach in [39] achieved the goal of approximation of the original user-item rating matrix using maximum margin MF (MMMF) that minimizes the sum of the squared errors between actual ratings and predicted ratings. Bell and Koren [76] presented a solution for the Netflix Prize [51, 52] by combining several linear combinations of prediction models and managed to win the prize; the challenge goal for them was to achieve a reduction in the Root Mean Squared Error (RMSE). Several approaches have appeared as extensions to the MMMF model; for example, the study in [44] proposed avoiding overfitting of the regularization parameters, a completely Bayesian action of probabilistic matrix factorization (PMF).The authors in [70–76] proposed several matrix factorization (MF) approaches such as an incremental variation of MF which competently handles novel end users' ratings. Next, we discuss the advantages and challenges related to CF techniques.

2.2. Other Types of Recommendation Methods. In this section, hybrid recommendation solutions are briefly introduced along with other recommendations approaches. Hybrid approaches combine two different approaches to integrate the advantages of different approaches and to overcome some of their challenges. To address the limitations in CB and CF techniques, several studies compared the performances of both CB and CF recommenders and demonstrated that hybrid approaches can generate recommendations with higher accuracy compared to using either CB or CF methods. Combining CB and CF methods can be proposed in different ways [5]:

(1) Combining the outcome from the prediction process after applying CF and CB distinctly.

(2) Totaling some CB features to improve collaborative recommenders.

(3) Totaling some CF features to improve the content-based system.

(4) Defining an overall model that includes both CB and CF features.

Recently, different recommender system approaches have appeared to propose solutions to the inherent limitations mentioned earlier. Due to the growth of Web 2.0 applications, new areas are emerging and need to be explored; for instance, the social tagging system (STS). Social tagging facilities appear to permit ordinary end-users to publish and edit contents and share free keywords. Thus, RSs are implemented to support end-users as result relevant topic and few marketable STS have been initialized to generate recommendations such as Deliciuis.com. This new direction of RS requires examining novel facets, approaches, and algorithms. STS deals with a third dimension, apart from the user, item dimensions, which may affect the complexity of the algorithms being used [77–80].

2.3. Standard Review-Based Recommenders. The enormous survey has been done to deed opinion in textual reviews to supplement justification in collaborative recommenders [80–85]. Reference [72] enhanced recommendation systems with working subject and sentimentality information at the sentence granularity level. The authors assessed ratings from review text remarks posted by end-users regarding various domain datasets for determining emotions classes. Also, Leung et al. [16] obtained a structure based on probabilistic sentiment inference, which they named Probabilistic Rating infErence Framework (PREF). The authors employed NLP techniques to compute opinion views in reviews. In other models, Naive Bayesian supervised technique was employed to predict ratings; this technique collaborated inference ratings from reviews with a collaborative filtering algorithm to improve the accuracy of suggested items to customers. Peleja et al. [65] inferred ratings from reviews by applying sentiment information to expand the complete superiority of RSs. The authors applied a multiple Bernoulli cataloging algorithm to calculate probabilistic rankings from the text analyses.

2.4. Microblog-Based Recommenders. As there is exponential progress of info on the Internet, end-users can effortlessly browse websites and post their opinions on online microblogs. A huge number of substitutes have become obtainable to users; hence, it is a novel test that entices investigators in the area of RS to investigate the potentials of initialing recommendations with the help of OSNs features and surroundings to improve RSs. Garcia Esparza et al. [74, 75] provided an example of incorporating OSNs in the foundation of recommendations. The authors examined how to get recommendations from virtual microblog services. They obtained a solution to focus on the content of small posts transcribed by end-users as artifact reviews to overcome the insufficient metadata about items in CB recommenders and ratings in CF recommenders. These posts are used to index the most frequently used and the most important terms to create end-user and item profiles using the TF-IDF technique described. Eventually, the query search algorithm was employed to retrieve pertinent item profiles with the help of data from a Twitter-like review service called Blipper.

This service gave users the ability to write short reviews and to rate movies at the same time. The learning is alike to work in terms of utilizing microblog's short posts to generate recommendations; however, people using microblogs do not usually rate items; they mainly express opinions. Also, items profiles are built based on a global point of view, not local and personalized, as all reviews written about an item are combined to construct its profile, and hence the profile is fixed over all the community. Another weakness in this study is that they assume that users have rich profiles and they introduce short reviews accompanied with ratings. Hence, the usual cold start problem, the case of new end users, was not addressed in these papers. Interestingly, in [4], the authors proposed a solution to the cold start tricky mobile software applications for smartphones (Apps). They used SNS like Twitter to recognize signals about the new release of these Apps by using Apps' accounts on Twitter. In the example in Figure 1, the Angry Birds Star Wars App has an official Twitter account with the handle @angrybirds5. They explored the followers list of these accounts. Their solution was an averaging technique in which the likelihood of how likely a given end-user would like a particular application is computed by observing how the Twitter users like this App. As per the assumptions, from a collection of Twitter users, the probability of users who like App is computed using the following formula:

$$\begin{aligned} p(+|a,u) &= \sum t\varepsilon T(a) p(+,|a,u) \\ &= \sum t\varepsilon T(a) p(+,|t,u) P(t,a). \end{aligned} \quad (16)$$

The sign + shows positive interest in App a, and $T(a)$ represents the set of people who follow the account of App a. However, the major drawback of the current work is that users need to follow item Apps on social media, which is not always the case. Another problem with this approach is that the products require official accounts on Twitter so that people can follow these accounts. However, no attempt was made to quantify the trust association among users based on their communications. In comaprison to this study, the

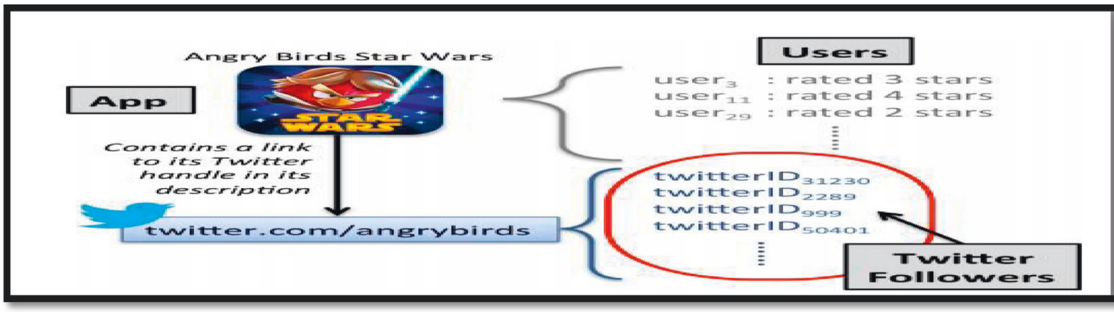

FIGURE 1: Using official accounts of apps on twitter in recommendation [4].

scenario in this thesis challenges the more general case for any item, especially when no product account exists on Twitter in a very personalized and subjective manner.

2.5. Challenges in Recommender System

(i) The current effort mostly emphasizes on categorizing end-clients into binary opinions sentiments such as positive or negative

(ii) The current methods primarily influence product category info to learn the interpersonal impact

(iii) These approaches are all limited to the structured data, which is not continuously obtainable on some websites

3. Techniques and Algorithms of Recommender System

3.1. Matrix Factorization. Matrix factorization (MF) is a method that calculates a hidden feature model of a system with the help of user item communication. Moreover, end-user item relations may be embodied as a matrix and one axis and substances on the other. Additionally, in maximum movie recommendation systems communications ratings are given by users for movies but they can be diverse data as well, like implied feedback and temporal properties. Furthermore, rating matrix is represented as sparse matrix real-time applications and end-users valued a fraction of all the movies in the organization. Likewise, MF is to construct expected forecasts on sparse matrices. Also, the matrix factorization technique decreases the dimensions of the rating matrix r by factorizing it into a product of two latent factor matrices, p for the users and q for movies [7].

$$\begin{bmatrix} r11 & \cdots & r1i \\ \vdots & \ddots & \vdots \\ ru1 & \cdots & rui \end{bmatrix} = \begin{bmatrix} p1 \\ p2 \\ p3 \\ \vdots \\ pu \end{bmatrix} \begin{bmatrix} q1 & q2 & \ldots & qi \end{bmatrix}, \quad r = PQ^T. \tag{17}$$

We have that f is the number of features extricated, u is the number of users, and I is the number of items (in our case, movies). Every row pu indicates vector of features for a user u and every row qi is a vector of features for an item i. Additionally, the product of vectors styles an approximation of the original rating.

$$r\,ui = P_u Q_i^{\,T}. \tag{18}$$

There are numerous ways of factorizing a matrix through several parts, utilized in various fileds of machine learning as well as statistics; nevertheless, maximum approaches do not work in case of any misplaced values in the matrix. Moreover, if it can be completed, Moreover, if it is completed, not only will observed values be estimated, but all missing values will also be forecasted. Additionally, one technique is to impute the lost values; however, doing so can distort the discovered data due to the sparseness of the original matrix. One more is to factorize by only utilizing the detected ratings and try to decrease the squared error.

$$Min \sum (r\,ui - puqiT). \tag{19}$$

However, there is a consequence in overfitting the preparation data. In order to avoid overfitting, a regularization term is presented to the squared error. Influence of regularization is measured by constant β [68].

$$\min \sum (rui - puqiT)2 + \beta (\|pu\|2 + \|qi\|2), \tag{20}$$

where $||.||$ denotes the Frobenius norm. This method has been displayed to be extremely fruitful and at the similar time scalable on especially large datasets, for example, the Netflix Prize competition where it was utilized in the two utmost accomplishing solutions [51]. Furthermore, the factorization could be completed and less memory is required when constructing a forecast (two vectors of size nf) compared to neighborhood methods where it is essential to keep the whole rating matrix or a subset of it in memory.

3.2. User k-NN Algorithm. The algorithm focused on in this experiment is *k-NN*, which measures the distance between nodes in a graph with respect to user or item similarities. The similarities for the neighborhood-based implementations can be calculated with different metrics; this study will use the Pearson correlation coefficient. This is a widely and commonly used metric for similarity measurements, used in earlier studies of k-NN benchmark analysis for movie datasets [30, 31].

This similarity formula calculates the similarity among user u and user v:

$$pearson\_sim(u,v) = \frac{\sum(rui - \mu v).(rvi - \mu v)}{\sqrt{\sum(rui - \mu u)\hat{2} \cdot \sqrt{\sum\ (rvi - \mu v)\hat{2}}}}, \tag{21}$$

where Iuv are the items rated by both end-user u and user v. rui is the rating produced by end-user u to item i. rvi is the rating produced by user v to item i. μu is the mean rating produced by user u. μv is the mean rating produced by user v.

This similarity formula calculates the similarity among item i and item j:

$$Pearson_{sim(i,j)} = \frac{\sum_{u\in U_{ij}}(r_{ui} - u_i).(r_{uj} - u_j)}{\sqrt{\sum_{u\in U_{ij}}(r_{ui} - u_i)^2}.\sqrt{\sum_{u\in U_{ij}}(r_{uj} - u_j)^2}}, \tag{22}$$

where U_{ij} are the users who have rated both items i and j. r_{ui} is the rating given by user u to item i. u_i indicates mean rating for item i. u_j indicates mean rating for item j.

3.2.1. User-Related k-Nearest Neighbors. The memory-based algorithm k-nearest neighbor is the most prevalent CF method in RSSEs. The basic strategy used by this algorithm is to measure weight for the user's score by looking at votes by other k similar users. In this study, this algorithm will be referred to as User KNN.

The prediction r_{ui} for an item rating for an end-user is computed with the following formula:

$$\widehat{r_{ui}} = \frac{\sum_{v\in N_i^k(u)} sim(u,v) \cdot r_{vi}}{\sum_{v\in N_i^k(u)} sim(u,v)}, \tag{23}$$

where $N_i^k(u)$ is the k-neighborhood of users concerning user u. sim(u, v) represents the similarity function explained above among end-user u and user v. r_{vi} indicates rating for item i given by user v.

3.2.2. Item k-Nearest Neighbors. There is another memory-based neighbor algorithm, calculating resemblance among items instead of end-users in the graph. In this study, this algorithm will be referred to as Item KNN. The rating prediction for a user of an item is computed with the help of the following formula:

$$\widehat{r_{ui}} = \frac{\sum_{j\in N_u^k(i)} sim(i,j) \cdot r_{uj}}{\sum_{j\in N_u^k(i)} sim(i,j)}, \tag{24}$$

where $N_i^k(u)$ is the k-neighborhood of users concerning user u. sim(i, j) indicates similarity function explained above among item i and item j. r_{uj} is the rating for item j given by user u.

3.3. User-Based Collaborative Filtering. The user-related CF procedure yields a reference list for the thing end users concerning the view of other end users. Ratings of various items valued by limited end-users are alike; the rankings of additional items valued by end-users will also be similar; CF RSs use statistical methods to hunt the adjacent neighbors of the item user, subsequently basing indicates on the object score rated by the adjacent neighbors to forecast the thing rating rated by the object user and give an equivalent endorsement list. CF component makes use of strategy as follows. According to this algorithm, a subgroup of users are selected based on their resemblance to the active end-user; a weighted combination of their ratings is to yield forecasts for the active end-user.

The algorithm would follow the below steps:

Step 1. Each user is weighed concerning resemblance with the active end-user. Similarity among users is computed as the correlation among their rating vectors.

Step 2. Active users having the highest similarity are chosen.

Step 3. Calculate a prediction, $P_{a,u}$, from a weighted blend. Similarity among two users is computed with the help of correlation coefficient.

$$P_{a,u} = \frac{\sum_{i=1}^{m}\left(r_{a,i} - \overline{r_a}\right) * \left(r_{u,i} - \overline{r_u}\right)}{\sqrt{\sum_{i=1}^{m}\left(r_{a,i} - \overline{r_a}\right)^2 * \sum_{i=1}^{m}\left(r_{u,i} - \overline{r_u}\right)|^3}}, \tag{25}$$

where $r_{a,i}$ indicates the rating valued to an item i by user a and r_a indicates mean rating given by user a.

Step 4. Forecasts are calculated as the weighted average of deviations from the neighbor's mean:

$$P_{a,i} = \overline{r}_a + \frac{\sum_{u=1}^{n}\left(r_{u.i} - \overline{r}_u\right) * P_{a,u}}{\sum_{u=1}^{n} P_{a,u}}, \tag{26}$$

where $P_{a,i}$ are the factors to forecast for the active end-user a for item i. $P_{a,u}$ is the similarity between users a and u. n is the number of users in the neighborhood.

4. Proposed Methodology

In this experiment, we implement the recommender system using user-related CF (collaborative filtering) and prediction of the movie ratings of MovieLens dataset using a Rapid-Miner tool.

4.1. Dataset. For our dataset, we are using MovieLens, a set of movie ratings structured and made available by Group-Lens. We use two sizes of it, MovieLens 100K containing 100.000 ratings and MovieLens 1M containing 1.000.000 ratings. The ratings are explicit containing user ID, movie ID, rating [1, 5], and a time tag. Moreover, the diverse datasets on the website are free for research and education (Table 2).

4.2. Method. The collaborative filtering (CF) scheme is to forecast user preferences for the unrated items and, after that, it recommends the most preferred items out of the list to the users. Nowadays, it is the mostly used recommender system method. Moreover, it delivers the best preferences to

Table 2: MovieLens dataset.

Datasets	Users	Movies	Ratings
MovieLens 100K	943	1682	105
MovieLens 1M	6040	3706	106
MovieLens 10M	69878	10677	107

the user. Many websites use this method like Amazon, Twitter, and so forth. As we are aware that 33% of the sales of Amazon are just because of the recommender system which uses the collaborative method, the recommender system using user-based CF is implemented.

Movie	Movie-id, title, year, description
Rating	User-id, movie-id, rating

Schema of Database

```
CREATE TABLE `movie` (
  `movieId` varchar(3) NOT NULL,
  `title` varchar(150) NOT NULL,
  `year` varchar(4) NOT NULL,
  `description` varchar(250) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `rating` (
  `userId` int(11) NOT NULL,
  `movieId` int(11) NOT NULL,
  `rating` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The k-nearest neighbor computation based on comparison with an unclear example and the k making examples are the neighboring elements of the obscure sample. The main goal is to use a database wherein the information emphases are inaccessible into a few classes to antedate the order of another sample point. KNN can be utilized for classification — the yield is class participation (predicts a class — a discrete worth). An item is arranged by a dominant part vote of its neighbors, with the article being relegated to the class generally regular among its k closest neighbors. It can likewise be utilized for regression — output is the incentive for the article (predicts constant qualities). This worth is the normal (or middle) of the estimations of its k closest neighbors.

4.3. Tool

4.3.1. RapidMiner Extension for Collaborative Recommender Engine. RS in RapidMiner Extension lead is allowed to set up with the help of e-LICO (e-Laboratory Interdisciplinary Collaborative Research) in Data-Science. RS is three types of administrators as object Recommendation, object rating prediction, and recommender performance. The author identifies with collaborative filtering in "thing rating prediction." Shared grounded administrators return a model set as having to prepare information, repeating a prepared model, and unaffected preparing information. The current applied technique rating forecast administrator recovers a prepared model and a test as information. The result of the currently employed model is a practice to figure execution utilizing the performance administrator. The performance operator will evaluate the error methods and error rates using the following methods. (i) Root Mean Square Error [RMSE], Mean Absolute Error [MAE], and Normalized Mean Absolute Error [NMAE]. Eventually, error measures quantities values are represented as vector of performance example set. In this experiment, the authors used a dataset called MovieLens which is freely available (see Figure 2). Here rating interval is considered as 1 to 5. In this process three files are considered: ratings.dat, users.dat, and movie.dat. In order to predict end-user rating with the help of CF method, three attributes are considered: (i) user ID, (ii) movie ID, and (iii) rating.

4.4. Create a Workflow of Item Recommendation in RapidMiner

4.4.1. Use of Single Model. The authors used AML operators to retrieve sparse dataset. This operator generates as result two files, training.aml and testing.aml. Collaborative recommender is an operator that creates recommendation model with the help of user-item matrix. The operator example set consists of data through which user rating is predicted. Item recommendation operator is used to identify dissimilar items rankigs for every individual user stored as part of database. Item related recommendation system gives matrix related to the resemblances among the rest of the items that are from different movies. KNN would identify unranked items with existing short list with the help of similar items. At last KNN predicts items that are weighted average of the ratings brought by the user. Afterwards, the below workflow is built as shown in Figure 3 according to the RapidMiner tool.

An item recommendation follows in Figure 3; it reads AML operator twice according to what is produced in training and testing sample data. Subsequently, roles are assigned for the below attributes:

User ID indicates end-user identification.

Item ID indicates item identification.

Rating indicates target label.

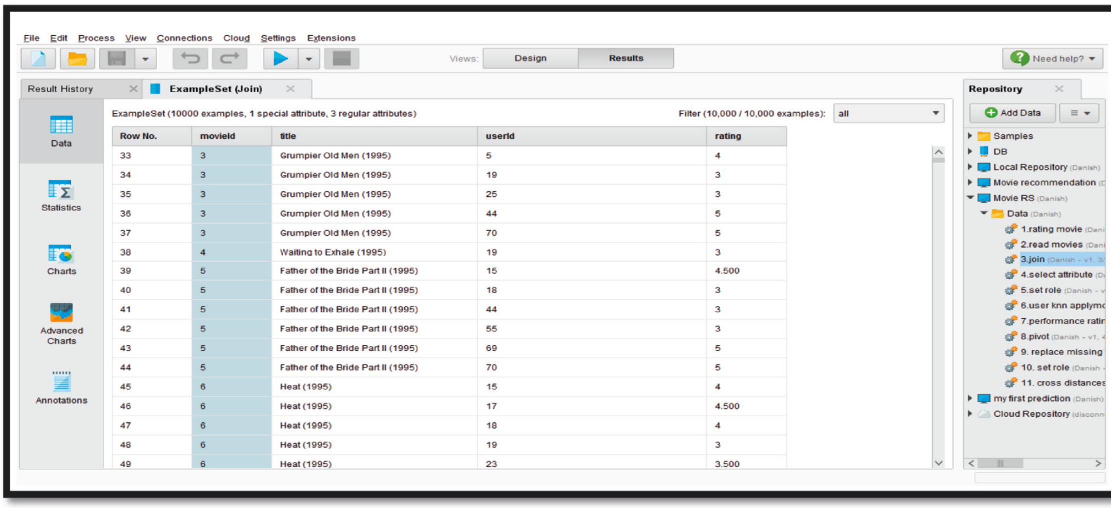

Figure 2: Initial dataset from MovieLens.

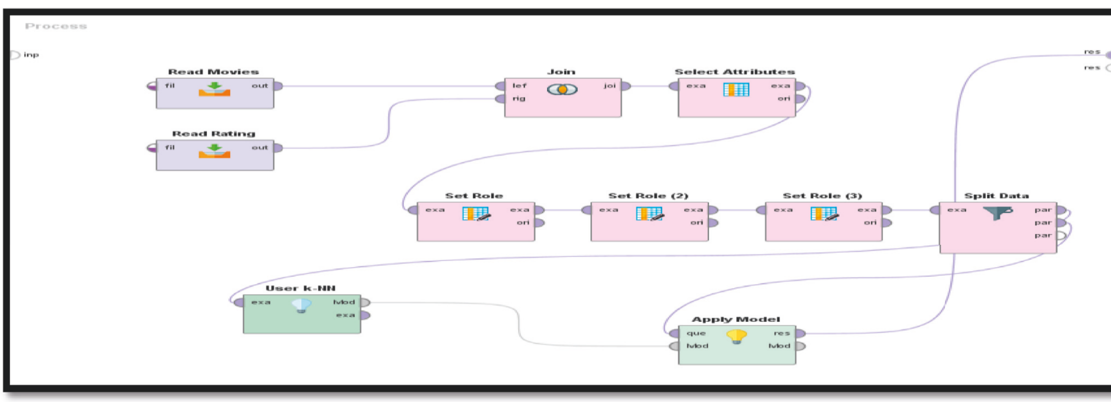

Figure 3: Workflow to find users' ratings for movies.

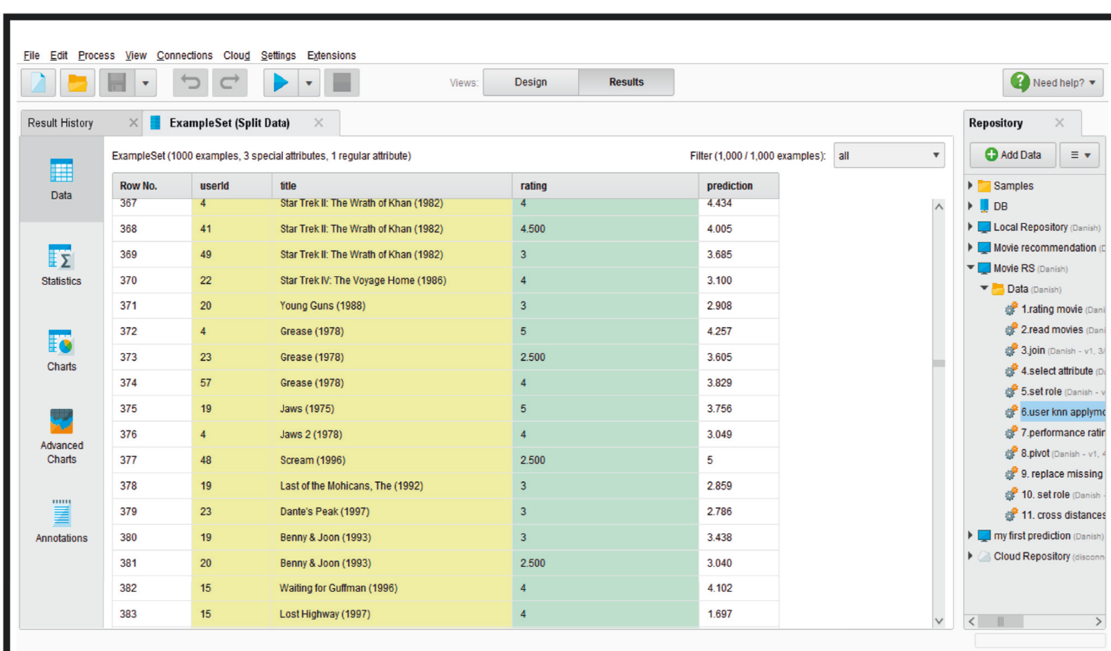

Figure 4: ratings for movies.

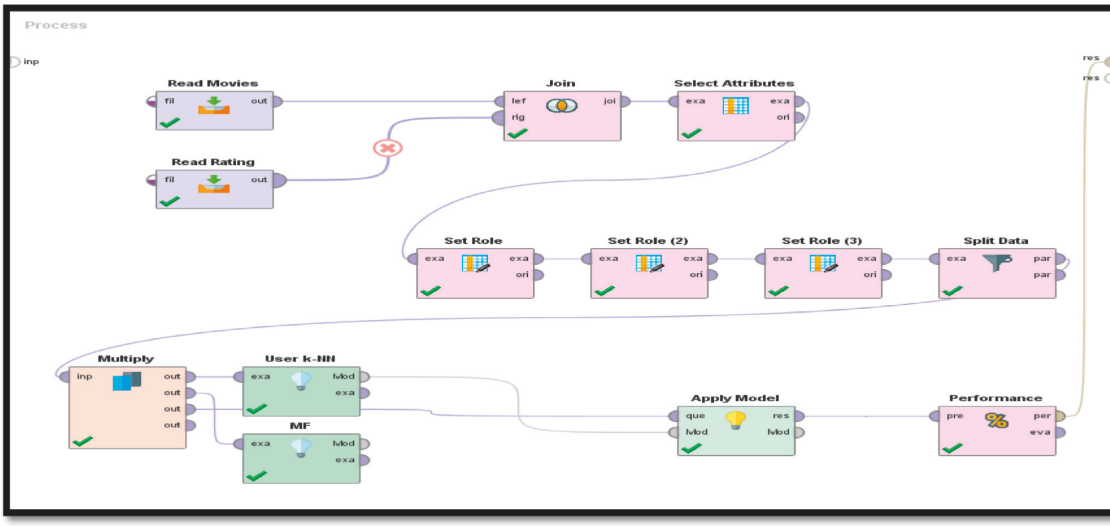

Figure 5: Using Multiply operator, Model Combiner (rating prediction), to put on multiple models to a set of data.

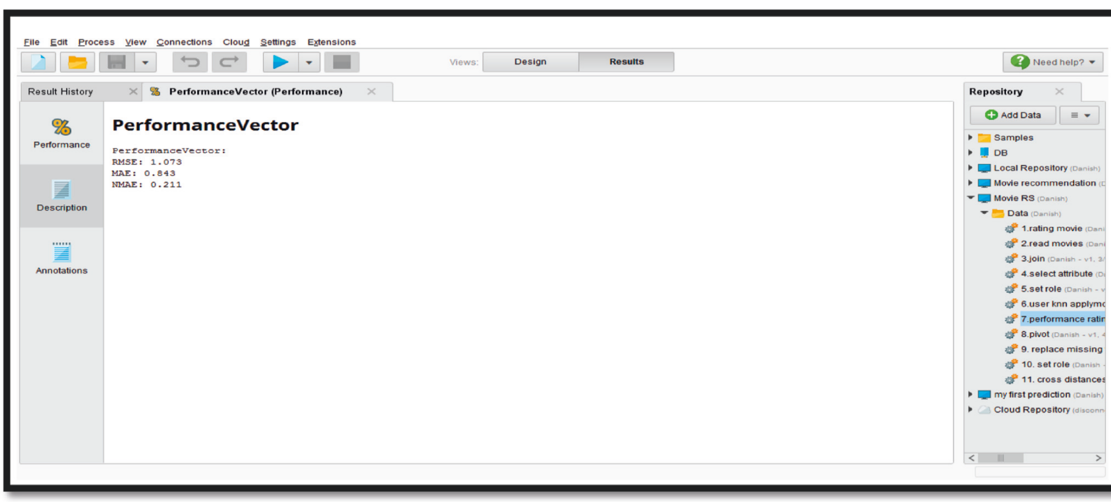

Figure 6: Performance of the recommender system using a single model.

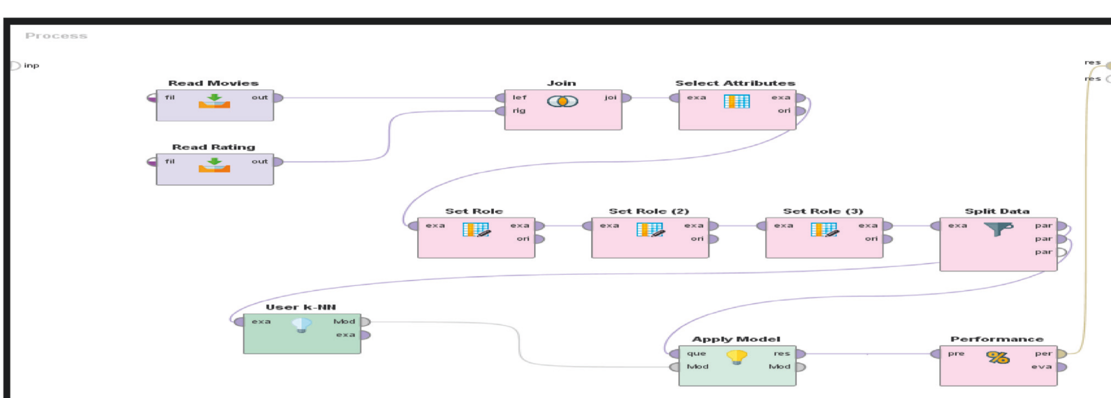

Figure 7: The complete model with user *k-NN* as model builder.

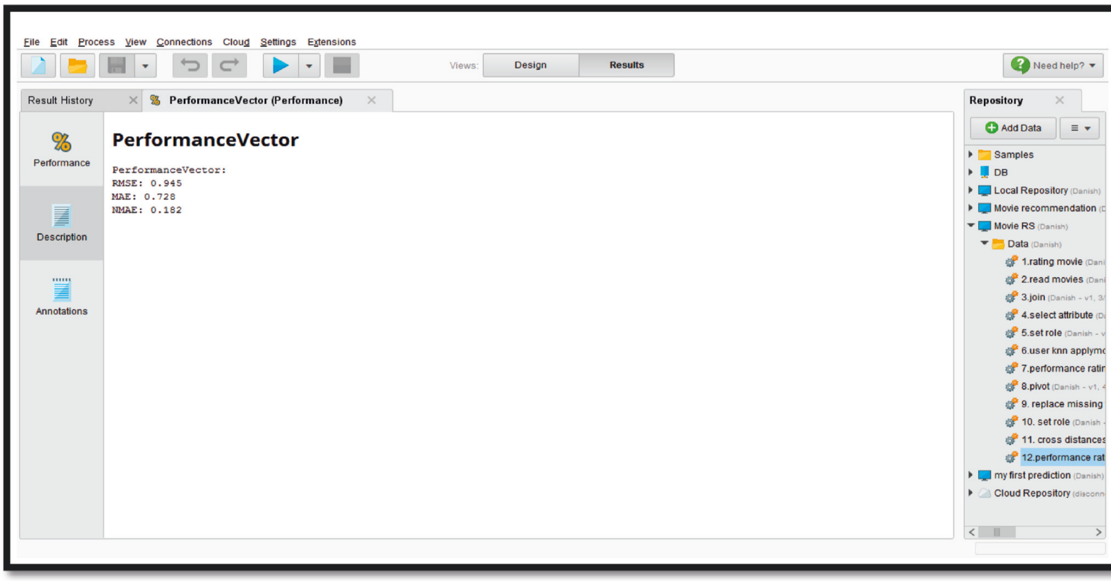

FIGURE 8: Performance of the recommender system using multiple models.

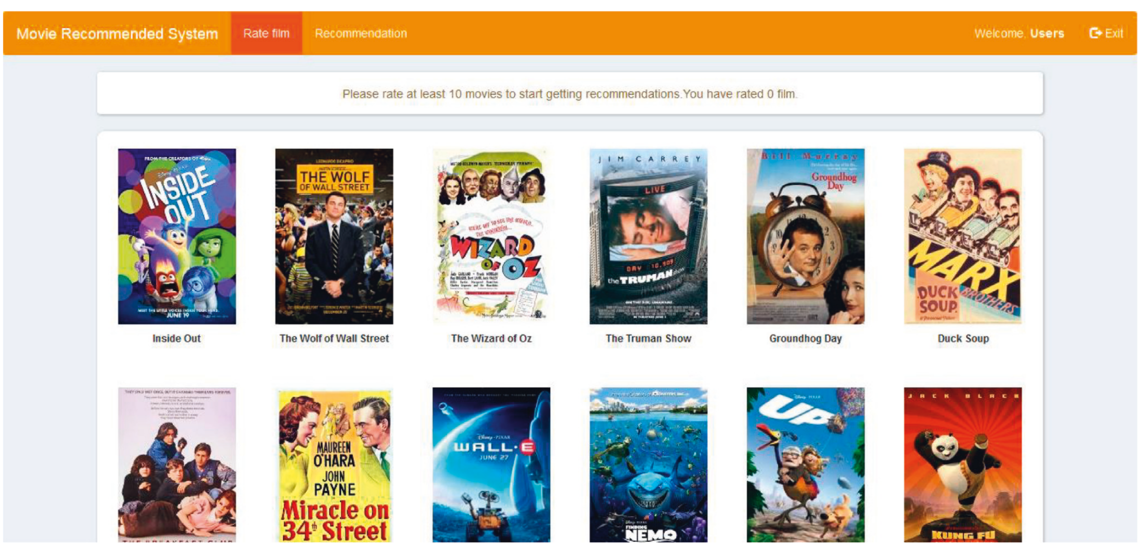

FIGURE 9: Rating at least 10 movies for recommendation.

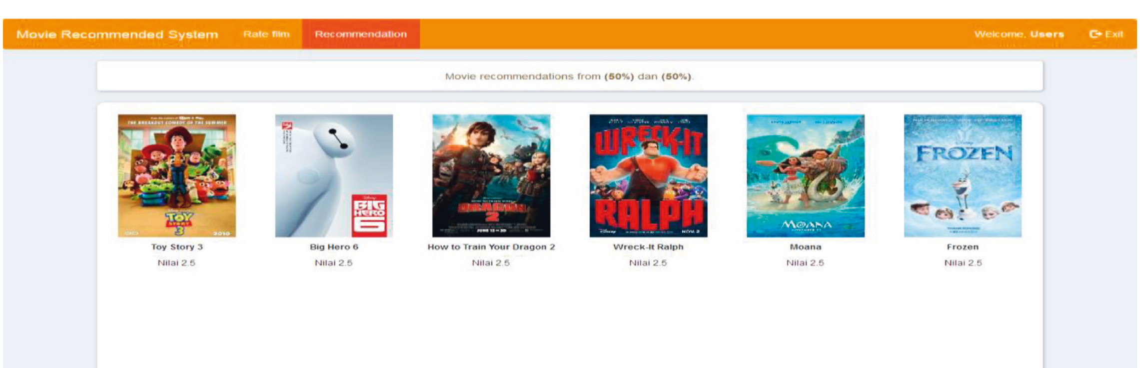

FIGURE 10: Recommended movies.

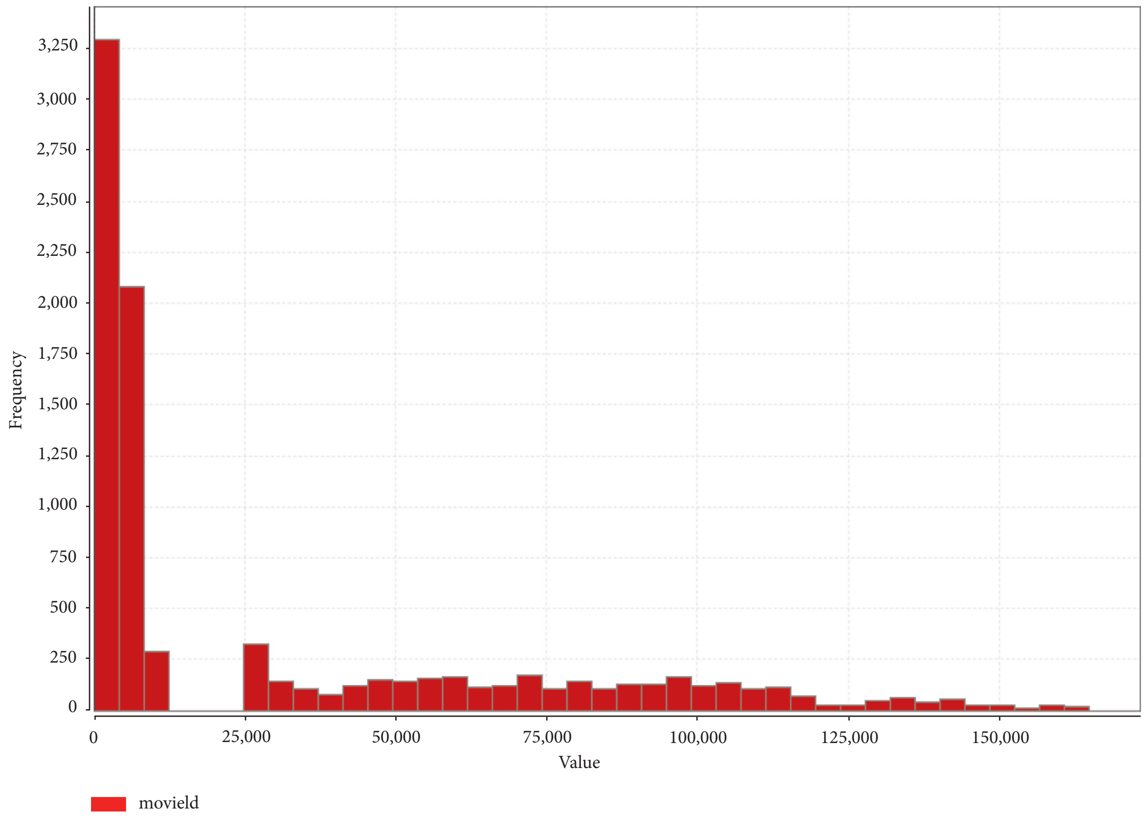

Figure 11: Movie dataset.

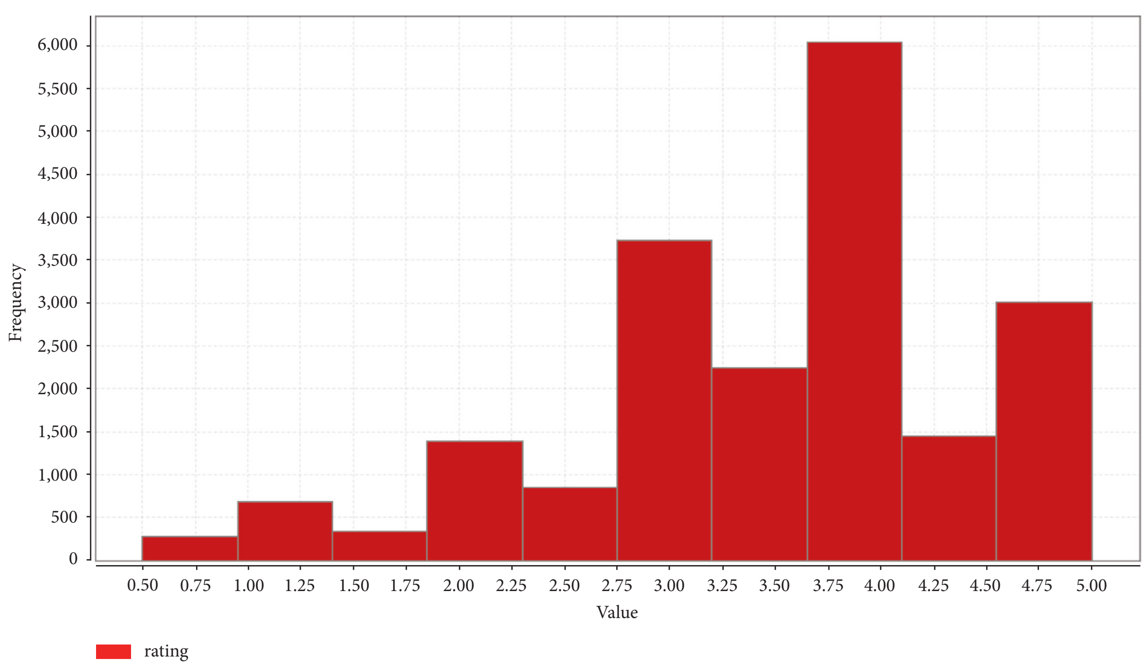

Figure 12: Distribution of user ratings.

The KNN operator component rating anticipation and apply model are used for forecasting rating. This operator yields a sample set possessing forecast ratings towards movies about a significant user in a testing example set. Figure 4 presents forecast output in a given dataset of movies of a specific user. Figure 5 demonstrates steps to measure the efficiency of a recommendation model. Working process to assess the recommendation system model efficiency in Figure 6 is analogous to Figure 7. The AML operator twice stores the input data or information and eventually sets roles to user ID and item ID. Subsequently, KNN operator used to predict item rating and employed model performance is also

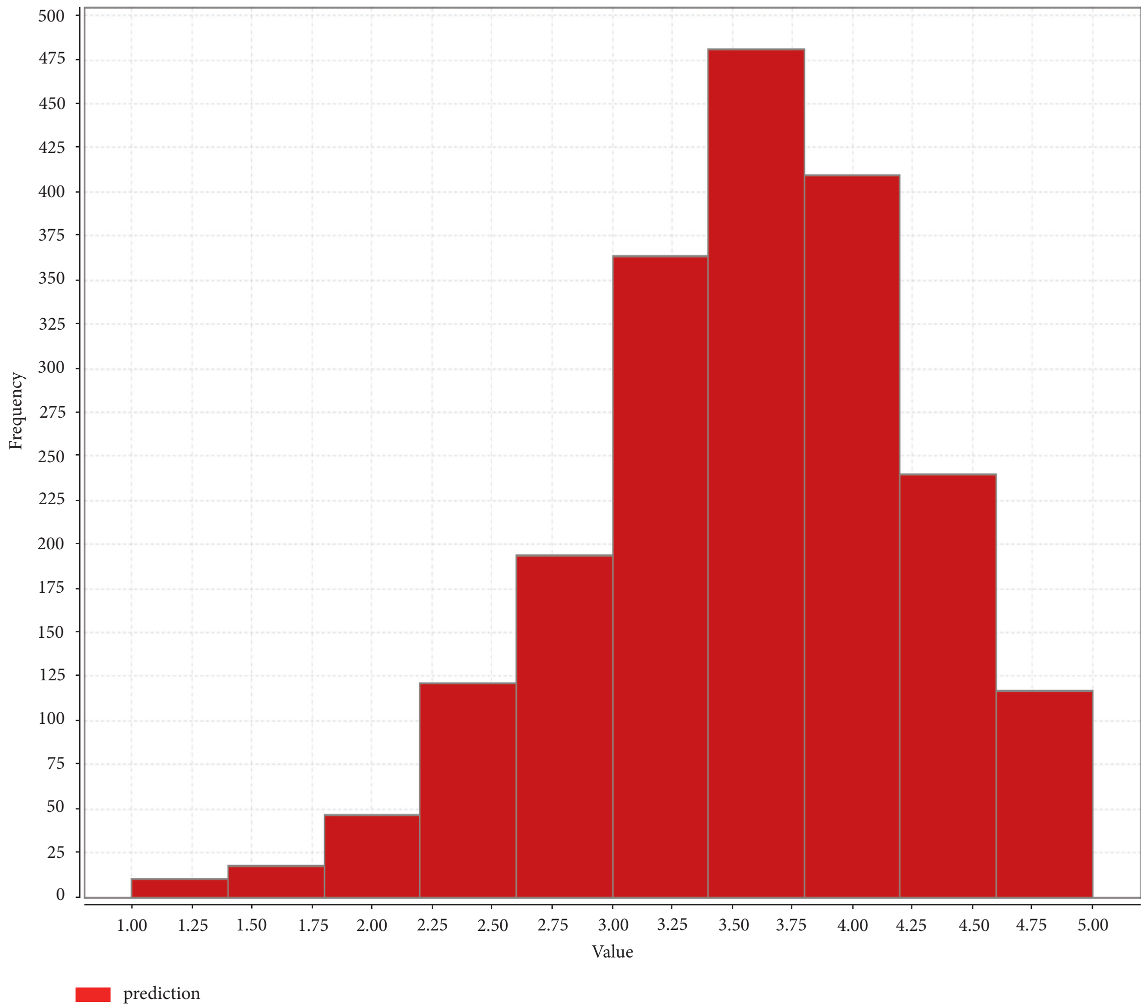

Figure 13: Distribution of prediction ratings.

evaluated and then it is run. Afterwards, results are computed by applying RMSE, MAE, and NMAE as demonstrated in Figure 6. Less values of RMSE and MAE indicate better accuracy.

4.4.2. Ensemble of Models. There are various procedures existing for item rating and they would be related in hybrid; also their result will be mutual with respect to weighted method in order to shape a model for the recommendation system. In order to apply the ensemble model, the authors used twofold operators where they multiply Model Combiner for the Rating Forecast. In this circumstance, the authors utilized an item KNN and matrix factorization algorithms in an analogous manner. Below steps involved in following process are presented in Figure 5.

After executing the above model, with predefined parameters, performance of recommendation system is drastically increased. The result is presented in Figure 8.

5. Result Analysis

Instruction for the user test was integrated into the applications menu. The users were asked to first rate 10 movies, making them eligible as a recommended item to friends. After rating the ten movies, six movies would be recommended to the user (see Figures 9 and 10).

After rating, the ten movies based on star top movies are recommended to the user.

The system movie database consists of 200 movies. The goal was to create a list of well-known movies that could generate many ratings from the user.

A total of 1000 ratings were registered in the system.

5.1. Dataset. The dataset used for prediction is in the form of Excel. There are two files: one contained the movie's information and the other contained the rating information. A total of 10,000 movies were taken into account.

A total of 10,000 ratings were registered by 671 users. It was shown that end-users were likely to rate movies higher rather than less. An end-user's average rating was computed to 11–13 (see Figures 11–13).

6. Conclusions

Recommender systems are a popular method for anticipating user behavior in systems that have a lot of data. The systems give the applications in which they are utilized with important client suggestions, allowing the business to make more money. They assist the customer by providing appropriate products while also generating revenue for the specific organization. The system must also provide adequate recommendations to newly recruited users so that they are satisfied with the suggestions. One technique to improve the precision of the predictions is to ask the end-user to rate numerous movies, which will gradually improve the precision of the suggestions. Currently, a community-based separation calculation is being carried out and evaluated. This paper proposes recommender frameworks and the use of RRE [RapidMiner Recommender Engine] to generate community-oriented film choices. RapidMiner, on the other hand, helps researchers to unleash their creativity by providing easy-to-use built-in models for distinguishing experiments and datasets. This article expected and supplied a number of machine learning tailored operators that could be used to solve various data science and machine learning problems. Deep learning models have also been introduced for various experiments for ensemble purposes. Nonetheless, the efficient use of recommender frameworks for tailored recommendation systems greatly contributes to present world expectations. This existing paradigm, which is based on environmental and business concerns, will alter dynamically in the future as technology improves. The authors looked into current literature for future models that could be improved based on the provided results and discussions. For example, when displaying a user's preferences, combining multiple types of evaluation elements may be more engaging than assuming a single kind. The authors have cited previous authors' implementations, research, and conclusions in fields such as multicriteria recommender systems, context-based recommender systems, and emotion-consideration recommender systems.

Data Availability

The processed data are available upon request from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Alag, *Collective Intelligence in Action*, Manning New York, 2009.

[2] P. Victor, M. De Cock, and C. Cornelis, "Trust and recommendations," in *Recommender Systems Handbook*, pp. 645–675, Springer, 2011.

[3] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd International ACM SIGIR Conference onResearch and Development in Information Retrieval*, pp. 203–210, ACM, 2009.

[4] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua, *"Addressing Cold-Start in Apprecommendation: Latent User Models Constructed from Twitter Followers," inProceedings of the 36th International ACM SIGIR Conference on Research Anddevelopment in Information Retrieval*, , pp. 283–292, ACM, 2013.

[5] G. Tuzhilin and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[6] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," *Recommender Systems Handbook*, pp. 1–35, 2011.

[7] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: anopen architecture for collaborative filtering of netnews," in *Proceedings of The1994 ACM Conference on Computer Supported Cooperative Work*, pp. 175–186, ACM, 1994.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filteringrecommendation algorithms," in *Proceedings of the 10th Internationalconference on World Wide Web*, pp. 285–295, ACM, 2001.

[9] M. Billsus and D. Billsus, "Learning and revising user profiles: the identificationof interesting web sites," *Machine Learning*, vol. 27, no. 3, pp. 313–331, 1997.

[10] L. Si and R. Jin, "Flexible mixture model for collaborative filtering," *Machine Learning-International Workshop Then Conference-*, vol. 20, p. 704, 2003.

[11] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems withsocial regularization," in *Proceedings of the Fourth ACM International Conferenceon Web Search and Data Mining*, pp. 287–296, ACM, 2011.

[12] Y. Kim and R. Phalak, *A Trust Prediction Framework in Rating-Based Experiencesharing Social Networks without a Web of Trust*, Information Sciences, 2012.

[13] C. Ronca and D. Ronca, "Advances in Clustering Collaborative Filtering by means of Fuzzy C-means and trust," *Expert Systems with Applications*, vol. 40, no. 17, pp. 6997–7009, 2013.

[14] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classificationusing machine learning techniques," in *Proceedings of the ACL-02 conferenceon Empirical methods in natural language processing*vol. 10, pp. 79–86, Association for Computational Linguistics, 2002.

[15] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: iamproving rating predictionsusing review text content," in *12th International Workshop on the Weband Databases*, 2009.

[16] C. W.-K. Leung, S. C.-F. Chan, F.-L. Chung, and G. Ngai, "A probabilistic rating inference framework for mining user preferences from reviews," *World Wide Web*, vol. 14, no. 2, pp. 187–215, 2011.

[17] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, "The state-of-the-art in personalized recommender systems for social networking," *Artificial Intelligence Review*, vol. 37, no. 2, pp. 119–132, 2012.

[18] S. Rosenthal, P. Nakov, S. Kiritchenko, S. M. Mohammad, A. Ritter, and andV. Stoyanov, "Semeval-2015 task 10: sentiment analysis in twitter," in *Proceedingsof the 9th International Workshop on Semantic Evaluation*SemEval, 2015.

[19] U. Shardanand and P. Maes, "Social information filtering: algorithms for automatingword of mouth," in *Proceedings of the SIGCHI Conference on Humanfactors in Computing Systems*, pp. 210–217, ACM Press/Addison-Wesley PublishingCo, 1995.

[20] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization:scalable online collaborative filtering," in *Proceedings of the 16th Internationalconference on World Wide Web*, pp. 271–280, ACM, 2007.

[21] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[22] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.

[23] S. Jayasri, J. Pryadharshini, P. Subathra, and P. N. Kumar, "Analysis and performance of collaborative filterinng and classification algorithms," *International Journal of Applied Engineering Research*, vol. 10, pp. 24529–24540, 2015.

[24] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Foundations and Trends in Human-Computer Interaction*, vol. 4, no. 2, pp. 81–173, 2011.

[25] A. Gunawardana and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *Journal of Machine Learning Research*, pp. 2935–2962, 2009.

[26] M. Poussevin, V. Guigue, and P. Gallinari, "Extracting a vocabulary of surprise by collaborative filtering mixture and analysis of feelings," in *Proceedings of the CORIA 2015—Conference in Search Infomations and Applications—12th French Information Retrieval Conference*France, Paris, March 2015.

[27] M. Z. Kurdi, "Lexical and syntactic features selection for an adaptive reading recommendation system based on text complexity," in *Proceedings of the 2017 International Conference on Information System and Data Mining*pp. 66–69, Charleston, SC, USA, April 2017.

[28] M. A. Ghazanfar and A. Pr¨ugel-Bennett, "An improved switching hybrid recommender system using naive Bayes classifier and collaborative filtering," in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 (IMECS)*Hong Kong, China, 2010.

[29] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval—SIGIR '02*New York, NY, USA, 2002.

[30] G. Shani, D. Heckerman, R. I. Brafman, and C. Boutilier, "An MDP-based recommender system," *Journal of Machine Learning Research*, vol. 6, no. 9, pp. 1265–1295, 2005.

[31] B. Walek and V. Fojtik, "A hybrid recommender system for recommending relevant movies using an expert system," *Expert Systems with Applications*, vol. 158, Article ID 113452, 2020.

[32] R. Baeza-YatesB. Ribeiro-Neto et al., *Modern information retrieval*, Vol. 463, ACM press, , New York, 1999.

[33] R. J. Mooney and L. Roy, "Content-based book recommending using learningfor text categorization," in *Proceedings of the Fifth ACM Conference on Digitallibraries*, pp. 195–204, ACM, 2000.

[34] M. Steinbach, G. Karypis, V. Kumar et al., "A comparison of document clusteringtechniques," *KDD workshop on text mining*, vol. 400pp. 525-526, Boston, 2000.

[35] M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro, "Semantics-Aware Content-Based Recommender Systems," in *Recommender Systems Handbook*, pp. 119–159, Springer, 2015.

[36] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[37] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[38] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluatingchoices in a virtual community of use," in *Proceedings of the SIGCHIconference on Human Factors in Computing Systems*, pp. 194–201, ACMPress/Addison-Wesley Publishing Co, 1995.

[39] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, *"Recommender systems forlarge-scale e-commerce: Scalable neighborhood formation using clustering," inProceedings of the fifth international conference on computer and informationtechnology*, vol. 1, 2002.

[40] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictivealgorithms for collaborative filtering," in *Proceedings of the Fourteenth Conferenceon Uncertainty in Artificial Intelligence*, pp. 43–52, Morgan KaufmannPublishers Inc, 1998.

[41] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, p. 4, 2009.

[42] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008.

[43] J. Bobadilla, F. Ortega, A. Hernando, and J. Alcalá, "Improving collaborative filtering recommender system results and performance using genetic algorithms," *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1310–1316, 2011.

[44] K. Miyahara and M. J. Pazzani, "Improvement of collaborative filtering with thesimple bayesian classifier," *IPSJ Journal*, vol. 43, no. 11, pp. 3429–3437, 2002.

[45] X. Su and T. M. Khoshgoftaar, "Collaborative filtering for multi-class data usingbelief nets algorithms," in *Tools with Artificial Intelligence, 2006. ICTAI'06.18th IEEE International Conference on*, pp. 497–504, IEEE, 2006.

[46] G.-R. Xue, C. Lin, Q. Yang et al., "Scalablecollaborative filtering using cluster-based smoothing," in *Proceedings Ofthe 28th Annual International ACM SIGIR Conference on Research and Developmentin Information Retrieval*, pp. 114–121, ACM, 2005.

[47] S. Obradovic and Z. Obradovic, "Collaborative Filtering Using a Regression-Based Approach," *Knowledge and Information Systems*, vol. 7, no. 1, pp. 1–22, 2005.

[48] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Advances in Neural Information Processing Systems*, pp. 1329–1336, 2004.

[49] J. D. Rennie and N. Srebro, "Fast maximum margin matrix factorization forcollaborative prediction," in *Proceedings of the 22nd International Conferenceon Machine Learning*, pp. 713–719, ACM, 2005.

[50] G. Tak´acs, I. Pil´aszy, B. N´emeth, and D. Tikk, "Investigation of various matrixfactorization methods for large recommender systems," in *Data MiningWorkshops, 2008. ICDMW'08. IEEE International Conference on*, pp. 553–562, IEEE, 2008.

[51] N. Singh and N. J. Ahuja, "Bug model based intelligent recommender system with exclusive curriculum sequencing for learner-centric tutoring," *International Journal of Web-Based Learning and Teaching Technologies*, vol. 14, no. 4, pp. 1–25, 2019.

[52] J. Bennett and S. Lanning, "The netflix prize," *Proceedings of KDD cup andworkshop*, vol. 2007, p. 35, 2007.

[53] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization usingmarkov chain Monte Carlo," in *Proceedings of the 25th International Conferenceon Machine Learning*, pp. 880–887, ACM, Helsinki Finland, July 2008.

[54] N. Singh, A. Kumar, and N. J. Ahuja, "Implementation and Evaluation of Personalized Intelligent Tutoring System Int," *J. Innov. Technol. Explor. Eng. (IJITEE)*, vol. 8, pp. 46–55, 2019.

[55] T. George and S. Merugu, "A scalable collaborative filtering framework basedon co-clustering," in *Data Mining, Fifth IEEE International Conference on*, p. 4, IEEE, 2005.

[56] A. J. Zargar, N. Singh, G. Rathee, and A. K. Singh, "Image data-deduplication using the block truncation coding technique," in *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, pp. 154–158, IEEE, 2015.

[57] G. SuryaNarayana, K. Kolli, M. D. Ansari, and V. K. Gunjan, "A traditional analysis for efficient data mining with integrated association mining into regression techniques," in *Lecture Notes in Electrical Engineering*, pp. 1393–1404, Springer, 2021.

[58] H.-W. Tung and V.-W. Soo, "A personalized restaurant recommender agent formobile e-service," in *Proceedings of the E-Technology, E-Commerce and E-Service, 2004. EEE'04.2004 IEEE International Conference on*, pp. 259–262, IEEE, Taipei, Taiwan, March 2004.

[59] O. Averjanova, F. Ricci, and Q. N. Nguyen, "Map-based interaction with a conversationalmobile recommender system," in *Mobile Ubiquitous ComputingSystems, Services and Technologies, 2008. UBICOMM'08. The Second InternationalConference on*, pp. 212–218, IEEE, 2008.

[60] M. J. Pazzani, "A framework for collaborative, content-based and demographicfiltering," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[61] N. Singh and N. J. Ahuja, "Empirical analysis of explicating the tacit knowledge background, challenges and experimental findings," *International Journal of Innovative Technology and Exploring Engineering*, pp. 4559–4568, 2019.

[62] L. B. Marinho, A. Nanopoulos, L. Schmidt-Thieme et al., *Social tagging recommender systems*, pp. 615–644, Springer, Berlin, Germany, 2011.

[63] C.-L. Huang, P.-H. Yeh, C.-W. Lin, and D.-C. Wu, "Utilizing user tag-based interests in recommender systems for social resource sharing websites," *Knowledge-Based Systems*, vol. 56, pp. 86–96, 2014.

[64] B. Mishra, N. Singh, and R. Singh, "Master-slave group based model for co-ordinator selection, an improvement of bully algorithm," in *Proceedings of the 2014 International Conference on Parallel, Distributed and Grid Computing*, pp. 457–460, IEEE, Solan, India, December 2014.

[65] F. Peleja, P. Dias, and J. Magalh̃aes, "A regularized recommendation algorithmwith probabilistic sentiment-ratings," in *Proceedings of the Data Mining Workshops (ICDMW),2012 IEEE 12th International Conference on*, pp. 701–708, IEEE, Brussels, Belgium, December 2012.

[66] S. R. Chen and M. Y. Chen, "Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web," *Management Science*, vol. 53, no. 9, pp. 1375–1388, 2007.

[67] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: opinionextraction and semantic classification of product reviews," in *Proceedings of The12th International Conference on World Wide Web*, pp. 519–528, ACM, New Yark, NY, USA, May 2003.

[68] P. A. Dimoka and A. Dimoka, "The Nature and Role of Feedback Text Comments in Online Marketplaces: Implications for Trust Building, Price Premiums, and Seller Differentiation," *Information Systems Research*, vol. 17, no. 4, pp. 392–414, 2006.

[69] E. Rashid, M. D. Ansari, V. K. Gunjan, and M. Khan, *Enhancement in teaching quality methodology by predicting attendance using machine learning technique*, pp. 227–235, Studies in Computational Intelligence, Berlin, Germany, 2020.

[70] S. Horváth and T. Horv´ath, "Opinion-driven matrix factorization for rating prediction," in *Proceedings of the User Modeling, Adaptation, and Personalization*, pp. 1–13, Springer, Rome, Italy, June 2013.

[71] N. Singh, N. J. Ahuja, and A. Kumar, "A novel architecture for learner-centric curriculum sequencing in adaptive intelligent tutoring system," *Journal of Cases on Information Technology*, vol. 20, no. 3, pp. 1–20, 2018.

[72] F. Lee and H. J. Lee, "Use of social network information to enhance collaborative filtering performance," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4772–4778, 2010.

[73] F. Shaik, A. K. Sharma, S. M. Ahmed, V. K. Gunjan, and C. Naik, "An improved model for analysis of Diabetic Retinopathy related imagery," *Indian J Sci Technol*, vol. 9, p. 44, 2016.

[74] S. Garcia Esparza, M. P. O'Mahony, and B. Smyth, "On the real-time web asa source of recommendation knowledge," in *Proceedings of the Fourth ACMconference on Recommender Systems*, pp. 305–308, ACM, New Yark, NY, USA, September 2010.

[75] S. Garcia Esparza, M. P. O'Mahony, and B. Smyth, "Mining the real-time web: A novel approach to product recommendation," *Knowledge-Based Systems*, vol. 29, pp. 3–11, 2012.

[76] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[77] M. D. Ansari, V. K. Gunjan, and E. Rashid, "On security and data integrity framework for cloud computing using tamper-proofing," in *Lecture Notes in Electrical Engineering*, pp. 1419–1427, Springer, Singapore, 2021.

[78] D. Maynard, K. Bontcheva, and D. Rout, "Challenges in Developing Opinionmining Tools for Social media," *Proceedings of@ NLP can u tag# usergeneratedcontent*, 2012.

[79] V. K. Gunjan, P. S. Prasad, and S. Mukherjee, "Biometric template protection scheme-cancelable biometrics," *Lecture Notes in Electrical Engineering. In ICCCE 2019*, Springer, vol. 570, pp. 405–411, Singapore, 2020.

[80] [Internet]Surprise,201,basicknN, 2017, http://surprise.readthedocs.io/en/stable/knn_inspired.html#surprise.prediction_algorithms.knns.KNNBasic.

[81] B. Sarwar, K. George, K. Joseph, and J. Riedl, *HongKongItem-Based Collaborative Filtering Recommendation Algorithms*, 2001.

[82] Y. Koren, *Las Vegas, Nevada. USA.Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model*, 2008.

[83] M. Papagelis and D. Plexousakis, "Qualitative analysis of userbasedand item-based prediction algorithms for recommendation agents," *Engineering Applications of Artificial Intelligence*, vol. 18, pp. 781–789, 2005.

[84] Q. Wang, X. Yuan, and M. Sun, "Collaborative Filtering Recommendation Algorithm Based on Hybrid User Model," in *Proceedings of the 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, Yantai, China, August 2010.

[85] A. S. Lampropoulos and G. A. Tsihrintzis, *Machine Learning Paradigms: Applications inRecommender Systems [Internet]*, Springer International Publishing, New York, USA, 2015.