Computer Systems Science & Engineering
DOI:10.32604/csse.2022.020670
Article

Tech Science Press

Profile and Rating Similarity Analysis for Recommendation Systems Using Deep Learning

Lakshmi Palaniappan[1,*] and K. Selvaraj[2]

[1]Department of Computer Science and Engineering, Veerammal Engineering College, K.Singarakottai, Dindigul, 624708, India
[2]Department of Information Technology, PSNA College of Engineering and Technology, Dindigul, 624622, India
*Corresponding Author: Lakshmi Palaniappan. Email: lakshmipalaniappanre@gmail.com

Received: 02 June 2021; Accepted: 03 July 2021

Abstract: Recommendation systems are going to be an integral part of any E-Business in near future. As in any other E-business, recommendation systems also play a key role in the travel business where the user has to be recommended with a restaurant that best suits him. In general, the recommendations to a user are made based on similarity that exists between the intended user and the other users. This similarity can be calculated either based on the similarity between the user profiles or the similarity between the ratings made by the users. First phase of this work concentrates on experimentally analyzing both these models and get a deep insight of these models. With the lessons learned from the insights, second phase of the work concentrates on developing a deep learning model. The model does not depend on the other user's profile or rating made by them. The model is tested with a small restaurant dataset and the model can predict whether a user likes the restaurant or not. The model is trained with different users and their rating. The system learns from it and in order to predict whether a new user likes or not a restaurant that he/she has not visited earlier, all the data the trained model needed is the rating made by the same user for different restaurants. The model is deployed in a cloud environment in order to extend it to be more realistic product in future. Result evaluated with dataset, it achieves 74.6% is accurate prediction of results, where as existing techniques achieves only 64%.

Keywords: Deep learning; restricted boltzman machine; profile based similarity; rating based similarity; item based similarity

1 Introduction

E-commerce has extended its horizon to provide different types of services and products to the end customers. Recommendation systems act as an integral part of the modern e-commerce sites and applications. Whatever the products may be, the sellers use recommendation systems for making their products reach their customer. The products range from music [1], movies [2] and even job portals are recommending jobs to their registered users [3].

The objective of this work is to analyses the recommendation systems based on user similarity both profile based and rating based. In addition to this a deep neural network model based on restricted boltzman machine is also developed and the performance is measured. The experimentation is done with the small dataset which could be helpful for the restaurants with small number of customer ratings.

The models implemented in this work, are as follows

- Recommendation of restaurants to the user based on similarity between the users in the context of profile-based similarity.
- Recommendation of restaurants to the user based on similarity between the users in the context of rating-based similarity.
- Recommendation of restaurants to the user based on the rating made by the same user with deep neural network.

In order to provide an impact of real time implementation, the details of the restaurants and the corresponding rating are deployed as the web services in the Google cloud environment. When a particular user is needed to be recommended with a set of restaurants, the ratings and the details of the restaurants are obtained from the web services and further processed. For ease of implementation, it has been considered the profiles of the users are stored in a separate web service. The model is depicted in the following Fig. 1.

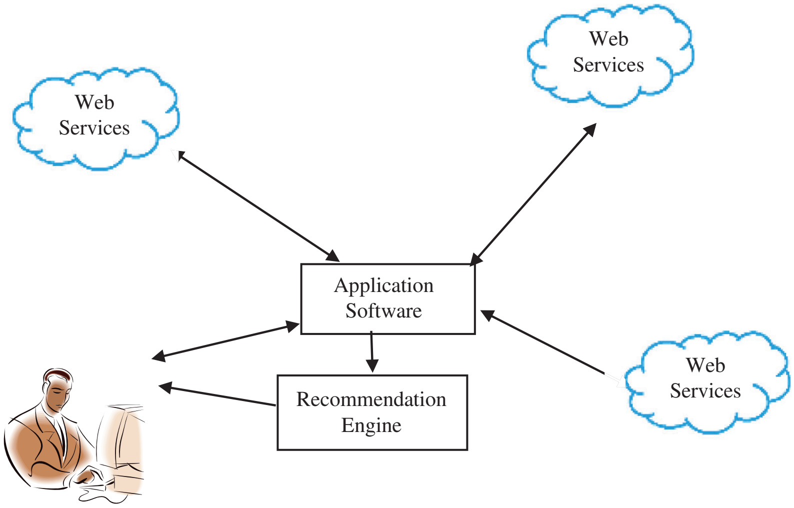

Figure 1: System model

The paper is organized as follows. The second section explains the state of art model that are in practice, the third section discusses in brief the various models implemented, the fourth section gives the results and the final section provides conclusion and the future work.

2 Related Work

The various approaches for recommendation system are depicted in the following Fig. 2. Content based recommendation models are the one that recommends items to the users based on the profile of the user's interest and the features that describes the item. This kind of model requires both user profile and the item profile in one or other form. Content based similarity model uses the similarity between the users and the similarity between the items. Knowledge based recommender systems are the one that makes some queries to the users and based on the replies made by the user, the products are recommended.

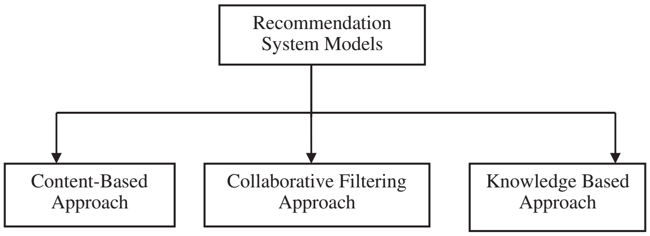
Recommendation System Models
Content-Based Approach
Collaborative Filtering Approach
Knowledge Based Approach

Figure 2: Classification of recommendation system model

Collaborative filtering is the commonly followed approach in recommendation systems. State of art models employ this kind of approaches for recommending items to the users. Collaborative filtering can be classified in to two types as shown in the below Fig. 3.

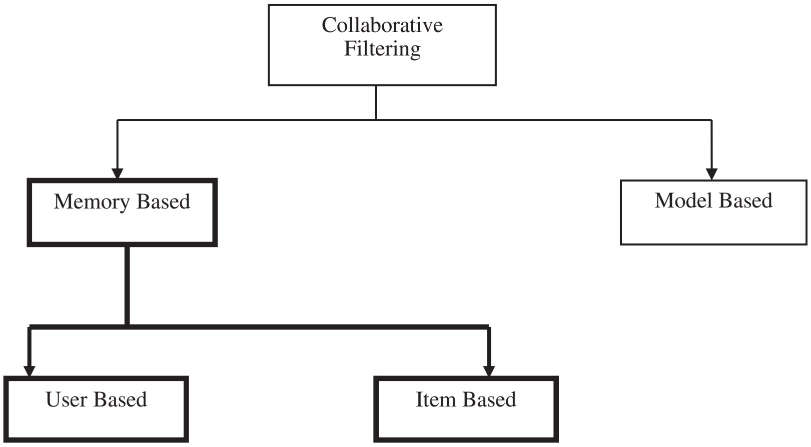

Figure 3: Classification of collaborative filtering based recommendation systems

The study made in this includes the various collaborative filtering approaches and their variants in terms of the way similarity measures are calculated and other such criteria.

Achrafand Lotfi et al. [4] discuss the various similarity measures that are commonly used for finding the similarity between the users and also between the items. They have listed the drawbacks that occur with those similarity models. They have proposed a new similarity model. The similarity measure is defined in such a way that it satisfies the specified qualitative conditions and institutive conditions. the similarity model is formulated in to mathematical equations and resolved to frame a kernel function. It has been observed from the results that the employment of this similarity model produces comparatively higher accuracy with the benchmark datasets. The dataset employed in this is the Movie-lens dataset.

The concept of multi criteria rating model is combined with deep neural networks in [5]. The dataset employed here is Trip-advisor's dataset which contains seven different criteria and their corresponding rating. First, the deep neural network designed for predicting the criteria rating are fed with the user Id and the Id of the item. The model would predict the rating of the different criteria. These criteria rating is normalized and given as input the deep neural network model for predicting the overall rating which in turn gives the recommendation. Mean absolute error is identified for the rating made for the individual criteria.

Valdiviezo-Diaz et al. [6] have built a recommendation engine based on naive bayes classifier. The computation is made by predicting the probability that a particular item will be given a particular rating by a specific user. This probability values are calculated with the ratings that already exist in the considered dataset. The process of calculating this probability includes the naive bayes classification model..

A new similarity measurement technique is proposed in [7]. Commonly used Pearson correlation and cosine similarity is replaced with intuitionistic fuzzy reasoning. The process of recommendation is done with finite prior ordering method. In addition to the user-based similarity and the item-based similarity that are followed in the traditional collaborative filtering-based recommendation models, Jawarneh et al. [8] have used the contextual information. The model proposed is called as the context aware neural collaborative filtering. The main step of the work proposed by the authors is the incorporation of item splitting which is used for including the context information in to the items.

Event recommendations are made in [9] with the help of the context aware hybrid collaborative filtering model which also includes semantic content analysis. For every user, with the help of his/her history of event registration, two interest models are created by the authors, short term and long-term models. In order to include the various impacts of the events on users, weight is also assigned called as influence weight. For selecting the similar neighbors long term influence model is used and for making user event matrix, short term influence model is used. From the similar user ratings, event ratings are predicted.

The number of web services pertaining to a particular application is increasing day by day; Botangen et al. [10] have designed a recommendation system that considers geographical location in recommending web services. The view of the authors is that the geographical location feature would have a great impact in applications related to Internet of Things. A probabilistic matrix factorization is also employed by the authors for making preferences taking into consideration the geographical location. The precision of the recommendations made increases with the inclusion of this feature as per the results obtained by the authors.

While most of the recommendation systems concentrates on recommending products to individuals, Pujahari and Sisodia et al. [11] have concentrated on recommending items to a group of users. In order to address the challenges created by the different preferences, and make suitable recommendations, matrix factorization with respect to preference relation is employed. It is followed by graph aggregation for aggregating the users based on preferences. The result indicates that the model performs well for group recommendations.

Marchand et al. [12] have put effort to give the reason behind the recommendations made. They believe that more precise recommendations with the reasons would help in building the trust of the customers by the online retailers. The collaborative filtering model and the content-based model are combined together to achieve this objective. The key part of the work is to consider the attribute-based preferences of the users and the attributes of the items and finding the interaction between them for modelling the rating.

According to Afoudi et al. [13] collaborative filtering models does not perform well when the contextual data is included. Sparsity problem also degrades the performance of collaborative filtering models. So, authors have incorporated content-based models. The recommendation system is built with two parameters; first, the contextual information extracted from the profile of the users and seconds the features of the restaurant.

The complexity of the problem increases in this kind of models. Finding neighborhood is also a tedious task in scenarios like this. Cai et al. [14] deals with this kind of model and the main finding of the authors are that in the traditional recommendation models, the nature of the neighborhood is asymmetric in nature which leads to problems in neighborhood. This problem is overcome with the employment of reciprocal neighborhood. Based on this, a new recommendation algorithm is proposed called as the K-Reciprocal nearest neighborhood algorithm. The results obtained are better than the related algorithms.

In [15], user based collaborative filtering model is enhanced with neural networks for differentiating the user's importance in the learning process. Though there exist various methods for addressing the sparsity problem in collaborative filtering, the complication increases with the increase in the size of the sparsity data. This problem is addressed in [16] where the linear relations and the nonlinear relations between the users are extracted, with the help of which multi factor similarity measure is calculated.

Recommendation of lists has become popular with applications such as generation of play lists. When the traditional collaborative filtering model considers only individual items when predicting the preferences, a mechanism is needed for considering a list of items instead of a single item. Such a model is proposed in [17]. The foremost process in this model is the learning made by a neural network enabled with attention mechanism and gate mechanism on list embedding, item embedding and the user embedding. This is followed by learning of three types of interactions by an interaction network; this includes interaction between the user and the item, interaction between the user and the list and finally interaction between the user and the item. The accuracy of the model in terms of recommending items as well as recommending lists are high than the other models.

3 System Design

The dataset employed in the work is obtained from UCI repository and it contains 138 users and 130 restaurants. It also contains the details of the individual users and the details of the individual restaurants. The user details are latitude, longitude, drink level, dress preference, ambience, transport, marital status, birth year, interest personality, religion, activity, color, weight, budget and height. In case of the restaurants, the details are latitude, longitude, name, address, city, state, country, fax, zip, alcohol, Smoking area, dress code, accessibility, price, URL, ambience, franchise, area and other services.

Similarly, the dataset also contains the rating made by the users for different restaurants. The ratings include overall rating, rating made for food and rating made for service.

Three models are designed,

- ▪ The first one employs the features of the users and to find the similarity and followed by the collaborative filtering way of recommending the items.
- ▪ The second model uses the user's rating for finding the similarity and make recommendation.
- ▪ The third model is based on restricted boltzman machine. The third model is also based on the rating and hence it can be applied in the various applications such as E-commerce, live streaming product recommendations.

The pre-processing steps included in the first model is as follows

- Handling missing/null values
- Handling categorical data
- Splitting the dataset into train set and test set

The dataset is split into training set and test set. 90% of the data is allocated for training set and 10% is used for test set. The models built are explained in the following sub sections.

3.1 Recommendation with User Profile-Based Similarity

The Fig. 4 represents the architecture of this model. The process of building the model is explained with a help of a single user, the first process is to find the similarity between the users. The percentage of similarity between the taken users and the other users are identified by constructing the KDTree provided in the scikit-learn library [18], KDTree employs Euclidean distance given in Eq. (1) to find the distance between the users from which the percentage of similarity is calculated.

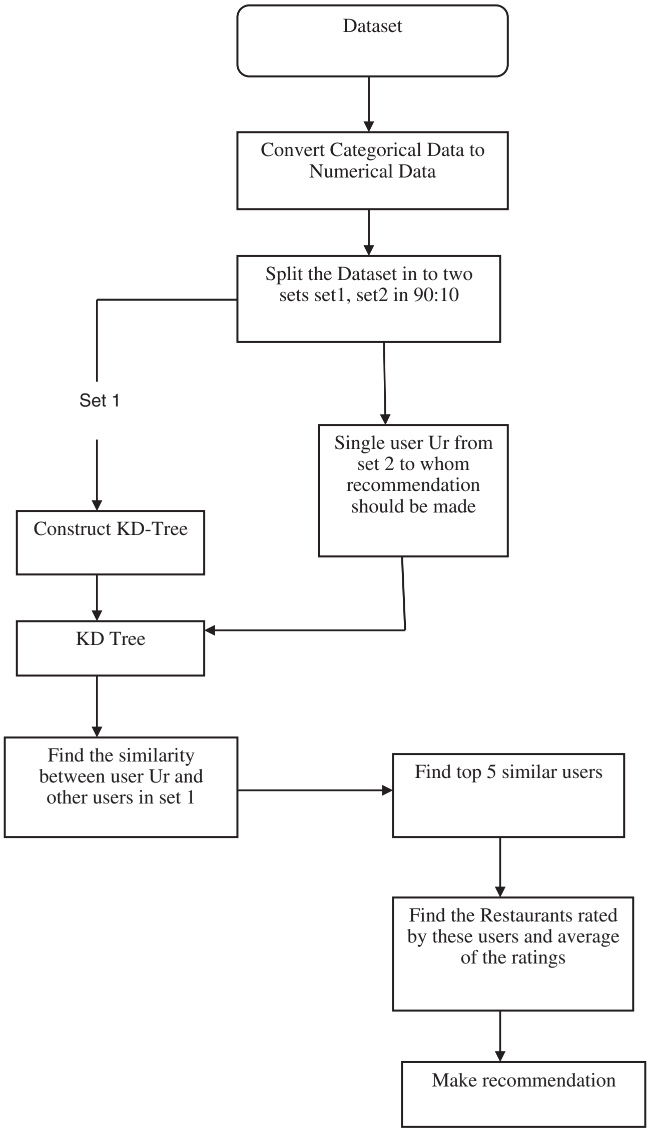
Dataset
Convert Categorical Data to Numerical Data
Split the Dataset in to two sets set1, set2 in 90:10
Set 1
Single user Ur from set 2 to whom recommendation should be made
Construct KD-Tree
KD Tree
Find the similarity between user Ur and other users in set 1
Find top 5 similar users
Find the Restaurants rated by these users and average of the ratings
Make recommendation

Figure 4: Architecture of user profile-based similarity model

$$d(X^{(i)}, X^{(j)}) = \sqrt{\sum_f (x_f^{(i)} - x_f^{(j)})^2} \tag{1}$$

For instance, for a particular user with user ID **U1029** the top 5 similar users are **U1021, U1053, U1063, U1116, U1131.** The percentage of similarity of the particular user to the top 5 similar users is given in the following Tab. 1.

The restaurants that are rated by these top users are consolidated and the ratings are averaged and restaurants with top average ratings are recommended to the user U1029. It can be observed the list is not in the right order as shown in Tab. 2. For instance the overall rating of the restaurant 135059 made by the user U1029 is 2 which is higher than the other restaurants. But in the list the particular restaurant occupies position behind all other restaurants rated lower by the same user is listed in Tab. 2.

Table 1: Top 5 similar users and percentage similarity

Si.No	User ID	Percentage of similarity
1	U1021	99.39867136999888
2	U1053	98.9496556930428
3	U1063	98.88673572911277
4	U1116	98.9496556930428
5	U1131	100.0

Table 2: Ranking of restaurants based on the user profile similarity

S. No	PlaceID	Name	Latitude	Longitude	Recommendation percentage
1	132866	Chaires	22.14122	−100.931	100
2	132870	Tortas y hamburguesas el gordo	22.14308	−100.935	100
3	132869	Dominos Pizza	22.14124	−100.924	100
4	132851	KFC	22.13687	−100.935	100
5	135054	Restaurante y Pescaderia Tampico	22.14063	−100.916	100
6	135082	la Estrella de Dimas	22.15145	−100.915	100
7	132668	TACOS EL GUERO	23.73821	−99.152	99.39867137
8	132715	tacos de la estacion	23.73242	−99.1587	99.39867137
9	132740	Carreton de Flautas y Migadas	23.7522	−99.1666	99.39867137
10	135032	Cafeteria y Restaurant El Pacifico	22.15248	−100.973	98.94965569
11	**132754**	**Cabana Huasteca**	**22.14774**	**−100.991**	**98.94965569**
12	135081	El Club	22.16484	−100.96	98.94965569
13	135062	Restaurante El Cielo Potosino	22.1537	−100.979	98.94965569
14	135027	Restaurant Orizatlan	22.14715	−100.974	98.94965569
15	**135085**	**Tortas Locas Hipocampo**	**22.1508**	**−100.983**	**98.94965569**
16	**132834**	**Gorditas Doa Gloria**	**22.15647**	**−100.986**	**98.94965569**
17	**135047**	**Restaurante Casa de las Flores**	**22.15092**	**−100.994**	**98.94965569**
18	135052	La Cantina Restaurante	22.15098	−100.977	98.94965569
19	**132825**	**puesto de tacos**	**22.14739**	**−100.983**	**98.94965569**
20	135038	Restaurant la Chalita	22.15565	−100.978	98.94965569
21	135063	Restaurante Alhondiga	22.15672	−100.976	98.94965569
22	**135059**	**Restaurant Bar Hacienda los Martinez**	**22.14511**	**−100.99**	**98.94965569**

The restaurants that are rated by U1029 are listed in the Tab. 3 given below and the ratings provided by them are also given. The Tab. 3. also includes rating provided by the similar users for the same set of restaurants if they have rated it.

Table 3: Rating made by target user and similar users

Restaurant	U1029			U1053			U1116		
	Overall rating	Food rating	Service rating	OR	FR	SR	OR	FR	SR
135047	1	1	1				2	2	2
135059	2	1	1				0	2	2
132937	1	1	1						
135085	1	1	1				2	2	2
132834	0	1	0				2	2	2
132754	0	0	0	1	2	1			
132825	1	1	0	1	2	0	2	2	2
132921	1	1	1						
132862	1	1	1						
132922	1	1	1						

The average rating of the other users and the average rating of the intended user is compared in the following Fig. 5.

Though it is known that comparison of rating made by two different users is not a good idea, it is made to get an idea of how efficient the recommendations are. It has also been observed from the results when ordered according to the average rating of the similar users also does not show better results. This is not the case with a single user, the model has been tested with 18 users and the average rating between the intended user and the similar users are compared and the root mean square is calculated, the value of which is 0.7912287009094507.

It has been inferred from this user similarity based recommendation model is that it could be used in cases where the users have not made any previous ratings. This can even helps in addressing the cold start problem that prevails in the recommendation systems. But, since the similar users are identified with the profile of the user rather than the rating made by the users, there is a scope for enhancing it which forms the second model.

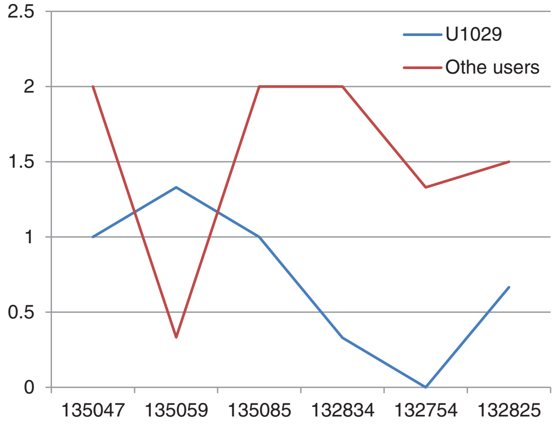

Figure 5: Comparison of rating made by the target user and similar user

3.2 Recommendation with User Rating-Based Similarity

In the second proposed model, the similarity between the users is calculated on the basis of the rating made by them on different restaurants. Given a user, the objective of the model is to recommend a set of restaurants are shown in Fig. 6. The steps of the model are given in Tab. 4.

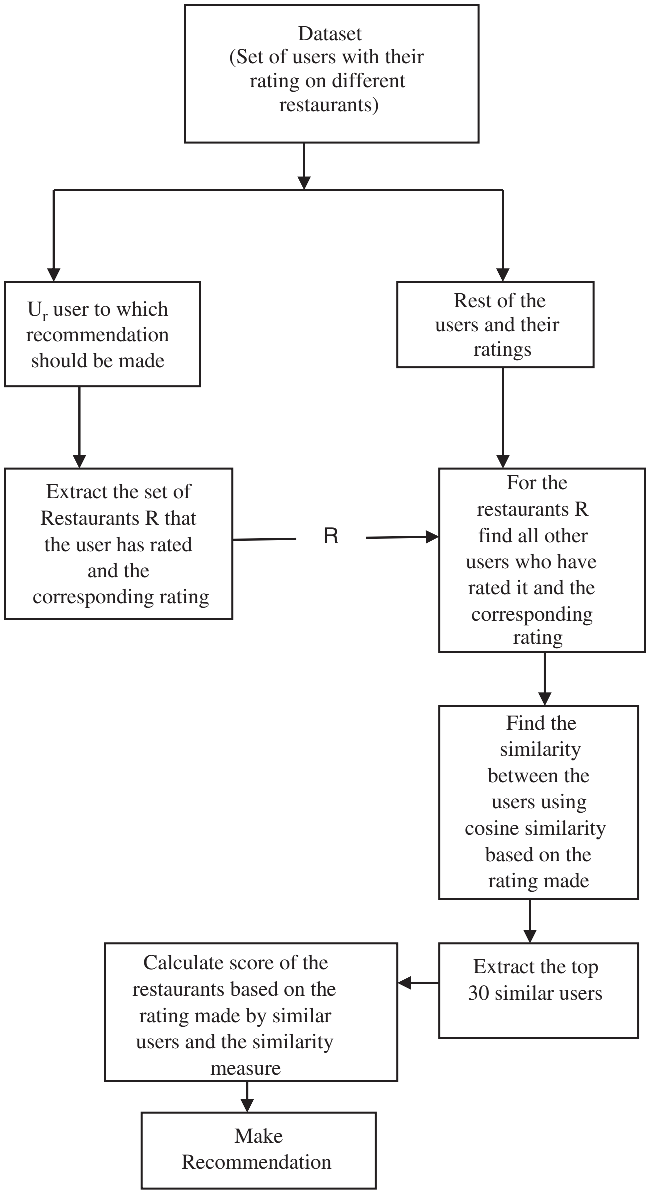
Dataset
(Set of users with their rating on different restaurants)
U_r user to which recommendation should be made
Rest of the users and their ratings
Extract the set of Restaurants R that the user has rated and the corresponding rating
R
For the restaurants R find all other users who have rated it and the corresponding rating
Find the similarity between the users using cosine similarity based on the rating made
Extract the top 30 similar users
Calculate score of the restaurants based on the rating made by similar users and the similarity measure
Make Recommendation

Figure 6: Architecture of user similarity-based model

The model is based on the similar between the ratings made by different users. In general, various normalization methods are followed to normalize the ratings of the user. This is done in order to address the variations in the rating made by the users with respect to their tolerance and conservative nature. But in the considered dataset the rating is made only with three values 0, 1, 2. So it is believed that normalizing the value is not required as there cannot be much variation. As the first step, Cosine similarity is calculated between the given user and all other users, cosine similarity given in Eq. (2) .

$$\text{Similarity (A, B)} = \cos(\theta) = \frac{A.B}{\|A\|\|B\|} \tag{2}$$

where A and B are vectors and in the considered scenario, they are the ratings made by the user. With the help of the cosine similarity, the top 30 users who are similar to the intended user are found. The rating made by the similar users for a particular restaurant is extracted. With these values, a score is calculated based on which the restaurants are recommended to the user. The score is calculated as per (3).

Table 4: Procedure for recommendation system using user rating

1. Consider a set of users U = (u1, u2,….un)
2. Let R={R1, R2, R3,….Rn} be the set of restaurants
3. Let RR be the rated restaurants , RR = {RR1, RR2,…RRm} and RR ⊆ R
4. Each Ui ∈ U has rated RRi Where RRi = { R1, R2, R3,…Rm} and RRi⊆ R
5. Let ur be the user to whom the recommendation should be made and ur ∈ U
6. Consider Uo, where Uo = U – {ur}
7. Find RRj rated by Ur
8. Find UR, where UR = {u1, u2,…um} , UR ⊆ U and rated one or more Ri in RRj
9. For each e in UR

Find similarity between u_r and e

10. Find top 30 similar users
11. Calculate score for each restaurant with rating made by individual users and the similar users.
12. Make recommendations

$$\text{Score} = \frac{\sum_{i=1}^{30} r_i sm_i}{sm_i} \tag{3}$$

where, r_i is the rating made by the similar users, sm_i is the similarity measure between the users

Finding the score enables us to getter a better idea on the results unlike the earlier model.

The important thing to note in this model is only the overall rating is considered rather than the considering all other ratings. The following Tab. 5. depicts it.

The comparison of the trend of change in the score with respect to the rating can be inferred from the below Fig. 7. With this method of calculating the score, the scores of all the restaurants rated by the similar users are calculated and the five restaurants with high score are recommended for the users. In both the models, there are no direct model of evaluation is available. Next a deep learning model is developed for recommending restaurants.

3.3 Item Similarity Based Model

The deep learning model employed in this work is restricted boltzman machine. The steps involved are described below

Table 5: Score of individual restaurants

Restaurant	U1029	
	Overall rating	Score
132825	1	1.2812499999999998
132937	1	1.4999999999999996
132862	1	1.3888888888888888
135059	2	1.6666666666666665
135047	1	1.0999999999999999
132834	0	0.9666878227378957
132922	1	1.838888183281475
132921	1	1.2607810762326521
132754	0	1.461538461538461
135085	1	1.3111250295293724

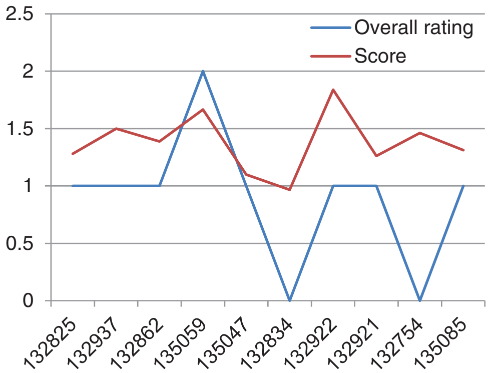

Figure 7: Comparison of the trend of change in the score with respect to the rating

3.3.1 Preparing the Dataset

The training set and the test set are in the separate files and they are loaded separately. Both the training set and test set contains the same data as of the ratings file. The number of observations in training data is 636 and number of observations in test data is 508. It follows the classical train and test split.

The training set and the test set are converted into an array such that the rows represent the users and the columns represent the restaurants. It has also been observed that the user rating is not common. The users are common in training and test set but the restaurants and the corresponding rating are different. The training and test set are converted into pytorch tensors and given as inputs to the boltzman machine. The train and test contain ratings in the range of 0 to 2 it has to be converted in to binary classification 1 or 0.

3.3.2 Restricted Boltzman Machine

Restricted boltzman machines unlike other models contains only two layers, the input layer and the hidden layer as shown in the following Fig. 8.

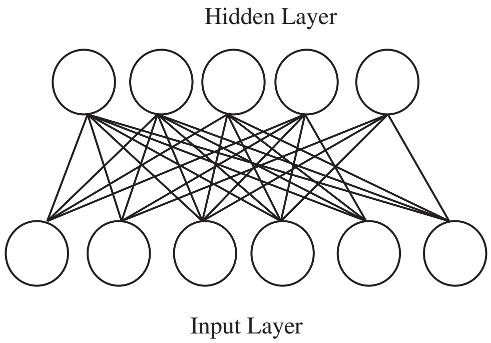

Figure 8: Restricted boltzman machine

The input layer is the one that has the value either 0 or 1. 1 represents that the particular user likes the restaurant and 0 represents that the user dislikes it. Hidden layer represents the factors such as the Food, Service etc. It is called as the latent factors which are used for describing the restaurant choices. The system finds the latent factors hidden, based on the preferences of the user when the training set of that user is given as input. Bernoulli distribution is used in the Restricted boltzman machine for identifying the neurons in the hidden layer that would be activated. In the hidden layer the value from the input layer is multiplied with a weight which is updated with contrastive divergence. The procedure of the model is given in Tab. 6.

Table 6: Procedure for restricted boltzmann machine

```
Let
W= Weight Matrix , a = bias vector of visible units , b= bias vector of hidden units , v= momentum,
$v_1$= visible units, Training vector of the model = $v_{1.}$
For given number of epochs:
For each j in the hidden units:
Calculate $P(h_{1j} = 1|v) = \frac{1}{1+e^{(-(b_i+w_iv_i))}}$
Sampling $h_{1j} \in \{0, 1\}$ from P ($h_{1j}$ $V_1$)
For each i in the visible units:
Calculate $P(v_{2i} = 1|h) = \frac{1}{1+e^{(-(a_i+w_ih_i))}}$
Sampling $v_{2i} \in \{0, 1\}$ from P($v_{2i}V_1$)
For each j in hidden units:
Calculate $P(h_{1j} = 1|v) = \frac{1}{1+e^{(-(b_i+w_iv_i))}}$
d =P ($h_1$=1$V_1$)
d1= d * $V_1^T$
e =P ($h_2$=1$V_2$)
e1= e * $V_2^T$
w = w + ∈ (d1−e1)
b = b + ∈ (d−e)
```

The restricted boltzman machine works in this fashion and the rating of the restaurant that has not been yet rated by the target user is identified. Since the probabilistic values are used in the models it paves us a way to measure the performance of the model. The values that are obtained with this model are given in the results section.

4 Experimental Setup and Results

The implementation of the web services in the cloud context is made in the cloud environment. The implementation of the interface between the recommendation system and the recommendation engine are done in python. The result obtained with the designed restricted boltzman machine is given below. The performance parameter used here is average distance which depicts the difference between the predicted values and the true values. The Fig. 9 represents the loss occurred during the training phase and the Fig. 10 represents the loss that occurs in the test phase.

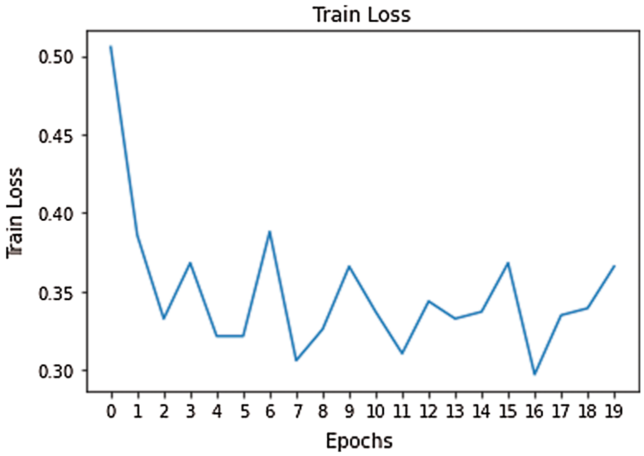

Figure 9: Training loss in terms of average distance between the predicted and the true values

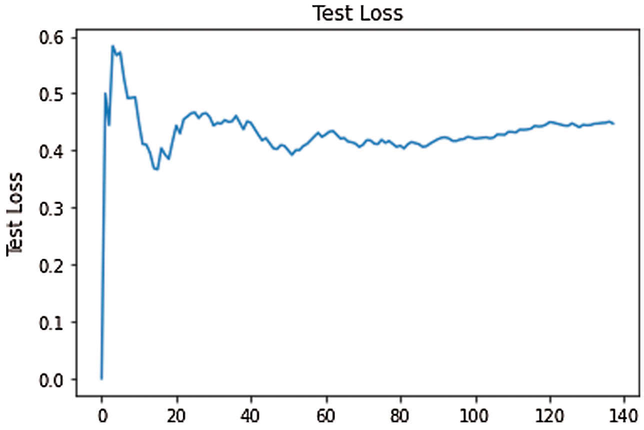

Figure 10: Test loss in terms of average distance between the predicted and the true values

From the average distance we can identify the percentage value of correct prediction. From the results obtained the percentage of correct predictions made during training is 64% and the percentage of correct prediction during test is 56%. It can be observed that the results obtained are low. But in general, the property of the deep learning model is that the performance of the model increases with the increase in the size of the dataset. The dataset employed here is a small dataset. The same model is tested with the movie lens dataset and the percentage of correct prediction is 74.6%.

5 Conclusion

Experimental analysis of the two recommendation models, one based on the similarity of the users based on their profile relevant to the context of choosing a restaurant and another based on the similarity of the users based on the ratings made by them is done. The advantages and the disadvantages of the models are analyzed. With the inferences made with these two models, a deep learning model with restricted boltzman machine is designed. This is designed to predict the rating of the restaurants that the user has not made earlier. The predictions are made only with the ratings made by the user for other restaurants. The implementation is made with the details of the restaurants, users and the corresponding rating as web services. This paves way for the implementation of the model in real time in future. Future work would also include the improvement of the percentage of correct predictions.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Mao, G. Chen, Y. Hu and L. Zhang, “Music recommendation using graph-based quality model,” *Signal Process*, vol. 120, pp. 806–813, 2016.

[2] C. Yang, L. Miao, B. Jiang, D. Li and D. Cao, “Gated and attentive neural collaborative filtering for user generated list recommendation,” *Knowledge-Based Systems*, vol. 187, 104839, 2020.

[3] M. Reusens, W. Lemahieu, B. Baesens and L. Sels, “A note on explicit versus implicit information for job recommendation,” *Decision Support Systems*, vol. 98, pp. 26–35, 2017.

[4] A. Gazdar and L. Hidri, “A new similarity measure for collaborative filtering-based recommender systems,” *Knowledge-Based Systems*, vol. 188, 105058, 2020.

[5] N. Nassar, A. Jafar and Y. Rahhal, “A novel deep multi-criteria collaborative filtering model for recommendation system,” *Knowledge-Based Systems*, vol. 187, 104811, 2020.

[6] P. Valdiviezo Diaz, F. Ortega, E. Cobos and R. LaraCabrera, “A collaborative filtering approach based on naïve Bayes classifier,” *IEEE Access*, vol. 7, pp. 108581–108592, 2019.

[7] Y. Zhang, Y. Wang and S. Wang, “Improvement of collaborative filtering recommendation algorithm based on intuitionistic fuzzy reasoning under missing,” *IEEE Access*, vol. 8, pp. 51324–51332, 2020.

[8] I. M. Jawarneh,P. Bellavista, C. Corradi, L. Foschini and R. Montanari, “A pre-filtering approach for incorporating contextual information into deep learning based recommender systems,” *IEEE Access*, vol. 8, pp. 40485–40498, 2020.

[9] M. Xu and S. Liu, “Semanticenhanced and context-aware hybrid collaborative filtering for event recommendation in event-based social networks,” *IEEE Access*, vol. 7, pp. 17493–17502, 2019.

[10] K. A. Botangen, J. Yu, Q. Z. Sheng, Y. Han and S. Yongchareon, “Geographic-aware collaborative filtering for web service recommendation,” *Expert Systems with Applications*, vol. 151, 113347, 2020.

[11] A. Pujahari and D. S. Sisodia, “Aggregation of preference relations to enhance the ranking quality of collaborative filtering-based group recommender system,” *Expert Systems with Applications*, vol. 156, 113476, 2020.

[12] A. Marchand and P. Marx, “Automated product recommendations with preference-based explanations,” *Journal of Retailing*,” vol. 96, no. 6, pp. 328–343, 2020.

[13] Y. Afoudi, M. Lazaar and M. A. Achhab, “Impact off feature selection on content-based recommendation system,” in *Proc.WITS*, Fez, Morocco, pp. 1–6, 2019.

[14] W. Cai, W. Pan, J. Liu, Z. Chen and Z. Ming, “K-reciprocal nearest neighboralgorithm for one-class collaborative filtering,” *Neurocomputing*, vol. 381, pp. 207–216, 2020.

[15] J. Chen, X. Wang, S. Zhao, F. Qian and Y. Zhang, “Deep attention user-based collaborative filtering for recommendation,” *Neurocomputing*, vol. 383, pp. 57–68, 2020.

[16] C. Feng, J. Liang, P. Song and Z. Wang, “A fusion collaborative filtering method for sparse data in recommender systems,” *Information Sciences*, vol. 521, pp. 365–379, 2020.

[17] S. M. Choi, S. K. Ko and Y. S. Han, “A movie recommendation algorithm based on genre correlations,” *Expert System and Applications*, vol. 39, no. 9, pp. 8079–8085, 2012.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V.Michel and B.Thirion, “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.