A Novel Hybrid Deep Multi-Criteria Model for Recommender System

Nour Salim Nassar (nournassar123@gmail.com)

Higher Institute for Applied Sciences and Technology https://orcid.org/0000-0003-0449-6835

Research Article

Keywords: Deep learning, hybrid recommender system, multi-criteria, recommender system

Posted Date: October 11th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-836949/v1

A Novel Hybrid Deep Multi-Criteria Model for Recommender System

Nour Nassar[1]

Higher Institute for Applied Sciences and Technology, Damascus, Syria.

Abstract

Recommender systems are everywhere books, products, movies, and more. Traditional recommender systems typically use a single criterion in the recommendation, while studies have shown that multi-criteria recommending is more accurate.

Novel deep learning techniques have produced remarkable achievements in many fields. The use of such techniques in recommendation systems has started to get attention recently, and several models of recommendation have been proposed based on deep learning. However, there is still no work for using deep learning in hybrid multi-criteria recommender systems. In this work, a model for a hybrid deep multi-criteria recommender system was presented. The model mainly includes two major parts: In the first one, the model obtains the user ID, item ID, and the item metadata to be used as input to a deep neural network in order to predict the criteria ratings. In the second part, the obtained ratings act as an input to another deep neural network, where the overall rating is predicted.

Our experiments were conducted on a real-world dataset. They demonstrated the superiority of the proposed novel model over the other models in all measures used to evaluate performance. This indicates the successful use of hybrid deep multi-criteria in the recommendation systems.

Keywords: Deep learning, hybrid recommender system, multi-criteria, recommender system

1. Introduction

Recommender systems (RSs) are widely used on a large scale, they surround people in their daily life, as online shopping, social networks, and entertainment are among the leading areas of RSs. RSs help in solving the problem of the large data volumes, as they filter and distill the data into the topics that may be interesting, and thus facilitate the decision-making process [1]. These systems provide recommendations and suggestions for the options that are most appropriate for each person. Most RSs attract users' opinions about an item through a single criterion, which leads to performance deficiencies because when the user expresses his opinion on an item [2], he may enjoy some of the element's properties (such as Food) and do not enjoy other properties, such as restaurant service. As a result, RSs

[1] Corresponding author.
E-mail addresses: nournassar123@gmail.com (N. Nassar).

based on several criteria were created, in order to evaluate the various properties of an item; these systems are called Multi-Criteria Recommender Systems (MCRSs) [3].

The early tries showed that the use of multi-criteria can increase the level of accuracy of RSs [4], as it will make RSs able to represent more preferences and tendencies for every user.

Deep Learning (DL) is currently making tremendous contributions in various fields such as text processing [5] and image visualization [6]. DL algorithms are being developed due to their ability to solve complex artificial intelligence tasks such as speech recognition or object detection and recognition, while traditional algorithms of machine learning have failed or showed average performance [7]. These accomplishments and others helped in raising the expectations for implementing deep learning in RSs. DL in RSs is an emerging area of research that shows promising results [8]. The strength of DL lies in its ability to represent complex models that describe user behavior for the recommendation process, and its ability to deal with sparse data in big data systems. Microsoft, Spotify, YouTube, and recently revealed significant improvements through the use of deep learning in the recommendation engines used within their systems [9]. We can thus enhance the RSs through the use of DL, in order to ensure that we obtain better results in line with our needs and interests.

Most of the recent research in RSs, which depend on DL, deal with models of traditional systems that use the one criterion in the evaluation, and according to the study that was conducted in this work, there is no research for a hybrid MCRS that adopts DL.

In this paper, we propose a novel model for a hybrid MCRS based on deep learning, using one of the conventional recommendation techniques, the hybrid technique. This technique combines collaborative filtering that uses only user ratings of items and content-based which uses item metadata. The model composes of two main stages: the first one uses a Deep Neural Network (DNN) to predict the criteria ratings, while the second part estimates the function expressing the relation between criteria ratings and the overall rating using another deep neural network.

By evaluating the model on a real-world dataset, it can be demonstrated the efficiency of using DL and multi-criteria in improving the performance of traditional hybrid RSs.

We divided the content of the paper into five sections. The second section presents the related works, while the third section describes the model in detail. Section four introduces the evaluation and results of our model. Finally, Section five provides the conclusion and sheds light on the prospects.

2. Related Work

[10] suggested a wide and deep learning structure for apps recommendation on the Google Play Store. The authors reveal that over-generalization when using sparse data leads to the recommendation with less relevant items. A wide structure was added to the model to avoid the over-generalization problem found in deep (feed-forward) neural networks and to recommend diverse and relevant items, TensorFlow-based implementation shows significant improvements in online and offline evaluation compared to wide or deep only structures.

[11] used two deep neural networks to recommend videos on YouTube, the first network generates the candidates while the second network ranks them. In this model, the candidate videos are selected using

a network based on a collaborative filtering technique (activity log as input) and then these videos are ordered according to their properties using another neural network. The network architecture follows a tower pattern where the number of neurons is halved in each successive layer, and ReLU activation function is used.

[12] presented a deep multi-criteria collaborative filtering model for RS that has two networks one for predicting the criteria ratings while the other for predicting the overall rating, then in [2], they fused matrix factorization with deep neural network in order to predict the criteria ratings.

3. Method

The proposed model is an enhancement of the deep multi-criteria collaborative filtering model, [12] have introduced, by using a hybrid technique that combines collaborative filtering and content-based to use the benefits of both techniques. In the model, as shown in Figure 1, a deep neural network was used to predict the multi-criteria in the first step, and in the second step, we learned the aggregation function f by using another deep neural network, where we provide a detailed description of these steps below.

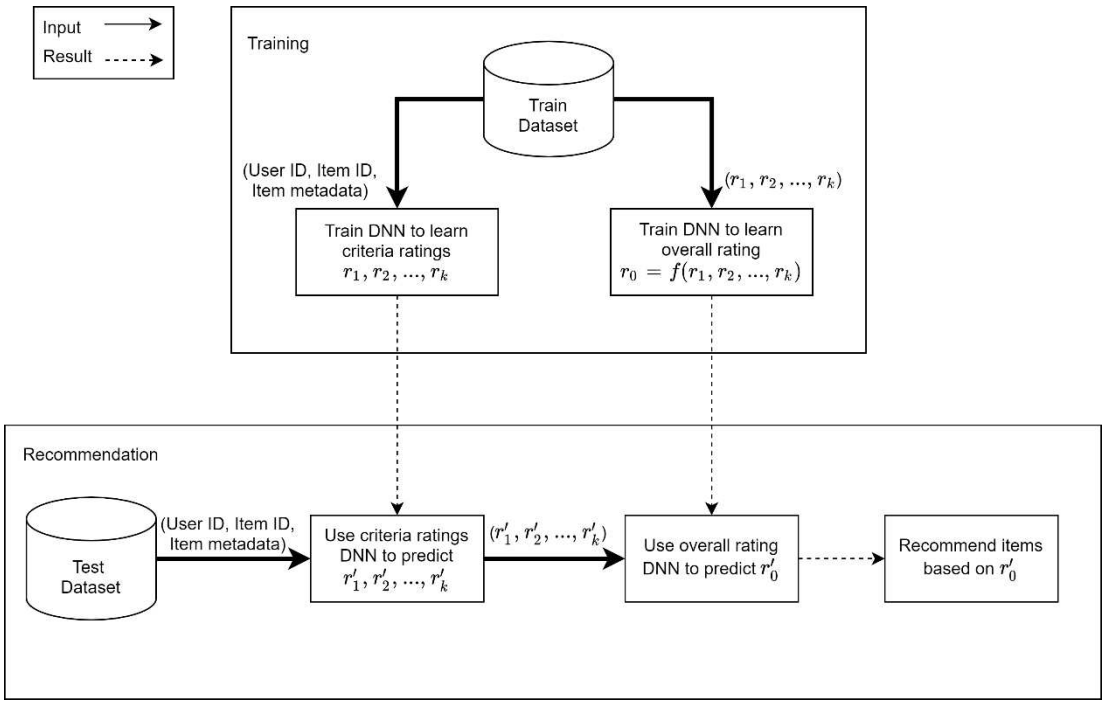

Figure 1: The proposed model overview.

3.1 Criteria Ratings DNN

At this stage, we will predict the criteria ratings using a DNN as shown in Figure 2. The continuous real-valued features are normalized because the DNNs are sensitive for the input normalization and distribution [13]. For categorical features, that do not have a logical order (e.g., User ID), we converted them to embeddings, dense real-valued and low dimensional vectors that are initialized randomly, and then the values are trained in order to minimize the value of the loss function during model training.

The input x is the concatenation of the embeddings and the continuous features and then followed by a number of hidden layers. The hidden layer output:

$$h^{(l)} = ReLU\left(W^{(l)}h^{(l-1)} + b^{(l)}\right) \tag{1}$$

where $W^{(l)}$ and $b^{(l)}$ the weights and bias of layer l and $h^{(0)} = x$.

The user u criteria ratings $r_1, r_2, \dots, r_k$ for item i at the output layer, as follows:

$$y_{ui} = ReLU\left(W^{(L)}h^{(L-1)} + b^{(L)}\right) \tag{2}$$

$$[r_1, r_2, \dots, r_k]^T = y_{ui} \tag{3}$$

where L is the network depth.

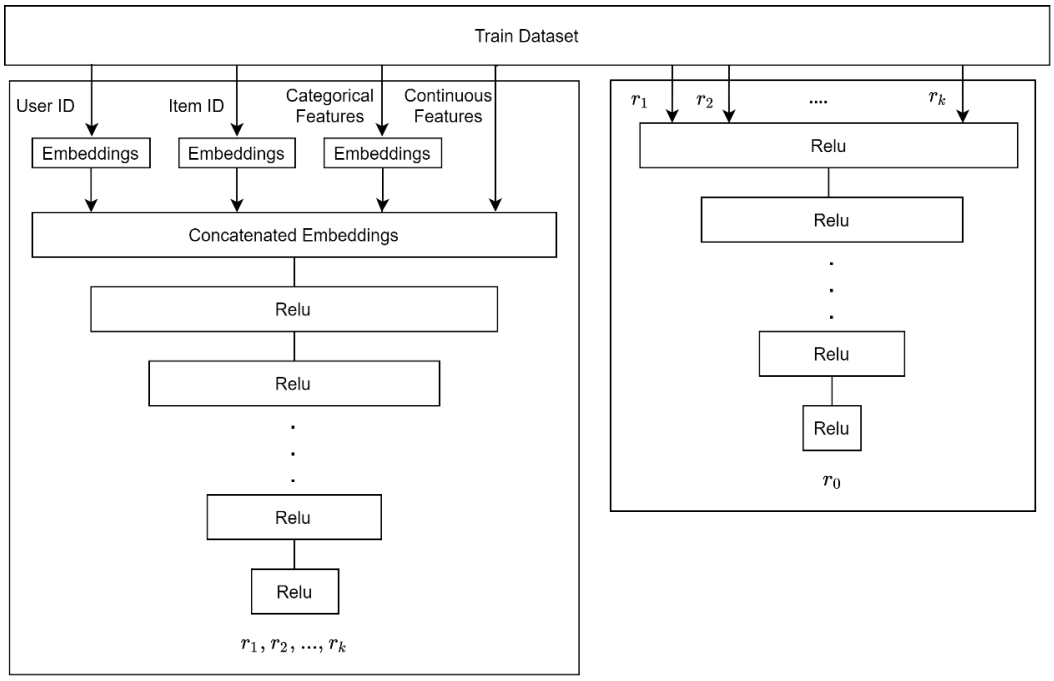

Figure 2: The model DNNs.

3.2 Overall Rating DNN

The aggregation-function-based approach considers that the overall rating can be estimated from the criteria ratings using an aggregate function. In this stage, we will learn the relation f between the criteria $r_1, r_2, \dots, r_k$ and the overall rating r_0:

$$r_0 = f(r_1, \dots, r_k) \tag{4}$$

The input of the DNN is the criteria ratings, continuous real-valued, so we normalize them first.

The input x is the normalized criteria ratings, and then followed by a number of hidden layers as in equation (1), and the output layer, using equation (2), as follows:

$$r_0 = y_{ui} \tag{5}$$

In designing both DNNs, we followed the tower pattern as shown in Figure 2, with the first layer being the widest, and each successive layer having a smaller number of neurons. We used the MAE loss function and Adam's optimization algorithm.

3.3 Providing Recommendations

After completing the model training process, each part was trained separately and independently from the other part. Now we can use the model to predict a user's overall rating on items that they did not rate before. The recommendation stage happens as shown in Figure 1:

- Predict criteria ratings. First, we obtain Item ID, User ID, and item metadata; input them into the criteria ratings DNN after normalizing the continuous features, to predict multi-criteria ratings $r_1', r_2', \dots, r_k'$.
- Predict the overall rating. In this step, we normalize the criteria ratings $r_1', r_2', \dots, r_k'$ that we predicted in the previous step and input them into the overall rating DNN to predict the overall rating r_0'.
- Make recommendations. In the last step, we provide recommendations to the user with the items based on the overall rating r_0', as in traditional single criterion RSs.

4. Results and Discussion

4.1 Dataset

This dataset includes real-world beer ratings from the BeerAdvocate2 website. The data span over more than ten years, with ~1.5 million ratings up to November 2011. Each review consists of ratings in terms of four criteria appearance, aroma, palate, taste, and an overall rating. Ratings vary on a scale from 5 to 1. Reviews include item metadata brewery, style, and alcohol by volume (abv). Table 1 shows the statistics of the dataset.

Table 1: BeerAdvocat Dataset Statistics

Dataset Statistics	Definition
Number of ratings	1,586,259
Number of users	33,387
Number of items	66,051
Timespan	Jan 1998 - Nov 2011

[2] https://www.beeradvocate.com/

4.2 Model Settings

We will tune the model parameters for both the criteria ratings DNN and the overall rating DNN in order to obtain the best results.

4.2.1 Settings of Criteria Ratings DNN

We set the parameters of the network, where we begin by initializing the weights of the network using the normal distribution with 0 mean and 0.05 standard deviation. We tested several optimizers to choose the algorithm with the best results. The results of the tests are shown in Table 2, and accordingly, we chose Adam's optimizer.

We tested several embedding sizes [4, 8, 16, 32, 64]. We plotted the MAE metric vs. the embedding sizes and obtained the results shown in Figure 3. As a result, we chose the embedding size equals to 8; we also set batch size and epochs equal to 256, and 16, respectively.

We tried many combinations of hidden layers as shown in Table 3. We followed the pattern of the tower, meaning reducing the neurons number by half in each successive layer. We selected the hidden layers according to the following order [64 → 32 → 16 → 8]. Finally, we placed 4 neurons in the output layer corresponding to the number of criteria ratings.

Table 2: Optimizers

Optimizer	MAE
SGD	0.4795 ± 0.0001
RMSprop	0.4646 ± 0.0015
Adamax	0.4666 ± 0.0031
Adam	**0.4641 ± 0.0016**
Adagrad	0.4921 ± 0.0022
Adadelta	0.5019 ± 0.0052

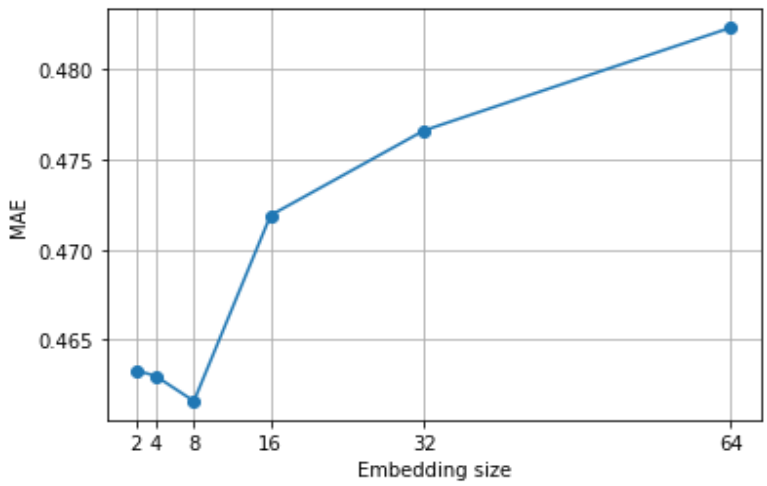

Figure 3: Embedding size results.

Table 3: Hidden Layers

Hidden Layers	MAE
[8]	0.4750 ± 0.0023
[16 → 8]	0.4701 ± 0.0028
[32 →16 → 8]	0.4646 ± 0.0033
[64 →32 →16 → 8]	**0.4616 ± 0.0017**
[128 →64 →32 →16 → 8]	0.4624 ± 0.0023

4.2.2 Settings of Overall Rating DNN

We made several tries to tune the deep neural network parameters to obtain the best estimation for the overall rating, and the parameters' values for the network are in Table 4.

Table 4: Overall Rating DNN Parameters

Parameter	Value
Batch size	1024
Epochs	64
Hidden layers	[32 → 16 → 8→ 4]
Output layer	1

4.3 Results

We used 5-fold cross-validation for the testing as mentioned earlier, where the data were divided into 20% for testing and 80% for training containing 10% for validation, and we repeated this process 5 times. We used the metrics MAE, Recall, Precision, and F1, where we chose the number of recommended items k = 10. For each metric, we computed the mean value on the overall rating. We compared our model to collaborative deep MCRS model, the one proposed by [12], we also compared it to a collaborative deep traditional model and the results are shown in Table 5. We notice from the results that our model

surpasses the models in all metrics, for example in the MAE metric, the improvement is at least 6%. Our model also surpassed the rest of the models on the recall metric and outperformed them on the precision and F1 metrics. Figure 4 and Figure 6 show the superiority of our hybrid two DNN model over the collaborative deep traditional model, because the overall rating gives information about how much the user prefers the item, while criteria ratings help understand why he prefers it, hence multi-criteria ratings estimate the similarity between users more precisely. Also, Figure 5 and Figure 6 demonstrate that our model provided a more accurate prediction for the ratings than the collaborative deep MCRS model because the item metadata helped in improving estimating the similarity between the items. Finally, the results demonstrate the benefits of using hybrid multi-criteria with deep learning for RS.

Table 5: Evaluation Results

Model	MAE	Precision	Recall	F_1
Collaborative Deep Traditional Model	0.4968 ± 0.0052	0.8452 ± 0.0258	0.4971 ± 0.0857	0.6201 ± 0.059
Collaborative Deep MCRS Model	0.4937 ± 0.0058	0.8484 ± 0.0016	0.5219 ± 0.0012	0.6462 ± 0.0014
Hybrid Deep MCRS Model (Ours)	**0.4616 ± 0.0017**	**0.8559 ± 0.0117**	**0.5284 ± 0.0475**	**0.6517 ± 0.0340**

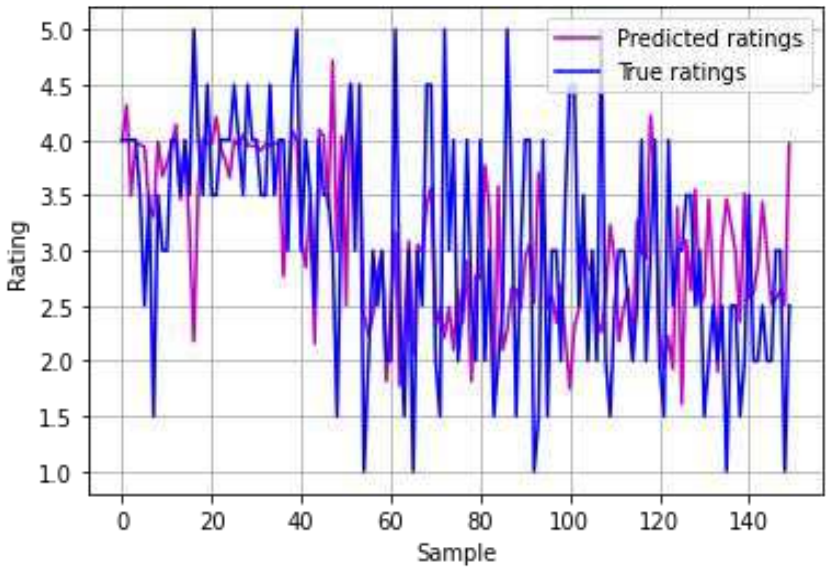

Figure 4: Collaborative Deep Traditional Model.

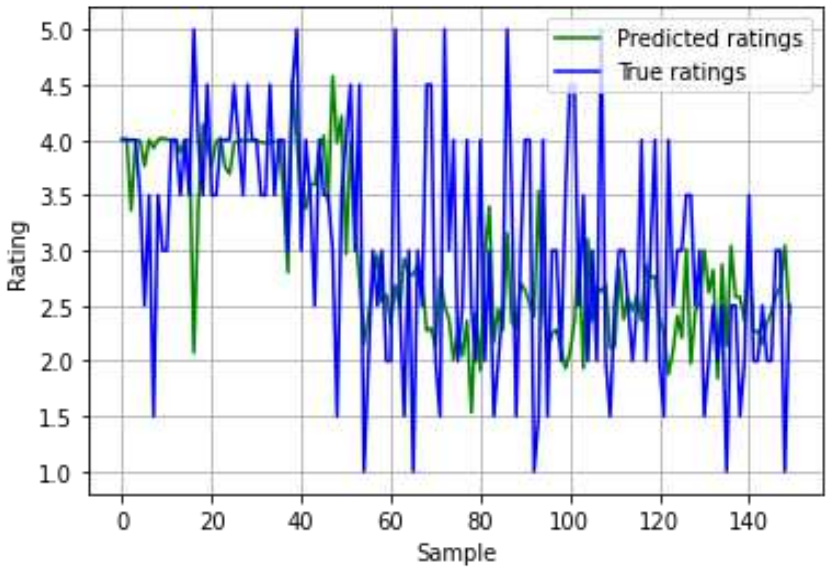

Figure 5: Collaborative Deep MCRS Model.

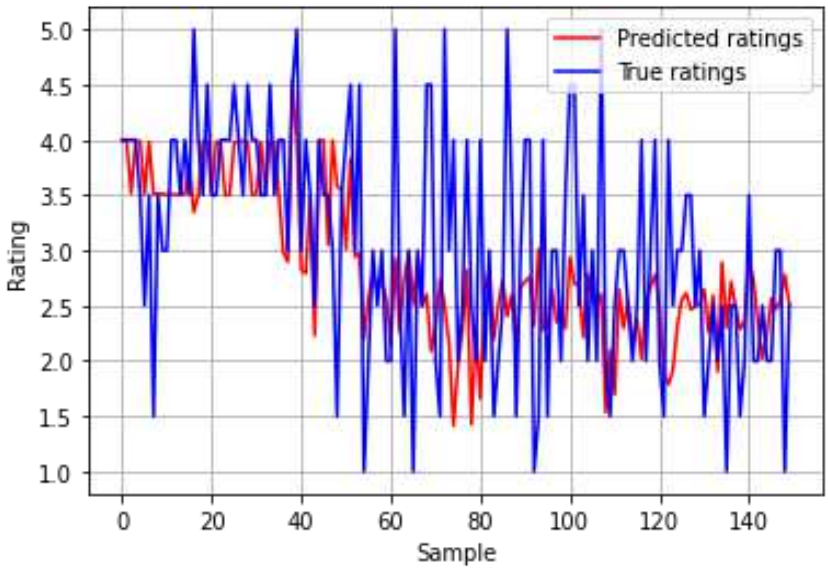

Figure 6: Hybrid Deep MCRS Model.

5. Conclusion and Future Work

In this paper, we proposed a model for a hybrid deep MCRS, in the first part, the model obtains the user ID, the item ID, and the item metadata to be used as input to a DNN that predicts the criteria ratings. These criteria ratings form the input to the second part, which is a DNN used to predict the overall rating. The model proved its effectiveness, and the experimental results showed that the proposed model performed well and it outperformed the rest of the models in all evaluation metrics, thus it demonstrated that the use of hybrid multi-criteria and deep learning is a successful method to improve the performance

of recommendation systems. Finally, the proposed model is generic, easy to implement, and independent of deep learning architectures, so we can improve the model using a different architecture.

As a follow up in the future, we want to enhance the performance of our model, to obtain better performance and thus provide more accurate recommendations, by studying other deep learning architectures such as the CNN, RNN, and autoencoder, integrating DNN with other models, or by using other feature representation methods.

Abbreviations

DL: Deep Learning; DNN: Deep Neural Network; MAE: Mean Absolute Error; MCRS: Multi-Criteria Recommender System; ReLU: Rectified Linear Unit; RS: Recommender system.

Acknowledgments

The authors would like to thank Simon Tarboush for the continued support.

Authors' contributions

All the corresponding authors contributed equally to the conduct of the present study. All authors read and approved the final manuscript.

Funding

Not applicable.

Ethics approval

Not applicable.

Consent to participate

Not applicable.

Availability of data and materials

The dataset used during the current study is available in the Datasets repository, https://github.com/nournassar/Datasets .

Competing interests

The authors declare that they have no competing interests.

References

[1] M. Shojaei, H. Saneifar, MFSR: A Novel Multi-Level Fuzzy Similarity Measure for Recommender Systems, Expert Syst. Appl. (2021) 114969. doi:https://doi.org/10.1016/j.eswa.2021.114969.

[2] N. Nassar, A. Jafar, Y. Rahhal, Multi-criteria collaborative filtering recommender by fusing deep neural network and matrix factorization, J. Big Data. 7 (2020) 34. doi:10.1186/s40537-020-00309-6.

[3] S. Gupta, V. Kant, An aggregation approach to multi-criteria recommender system using genetic programming, Evol. Syst. 11 (2020) 29–44. doi:10.1007/s12530-019-09296-3.

[4] C. Musto, M. De Gemmis, G. Semeraro, A Multi-criteria Recommender System Exploiting Aspect-based Sentiment Analysis of Users ' Reviews, (2017) 321–325.

[5] A.B. Nassif, A. Elnagar, I. Shahin, S. Henno, Deep learning for Arabic subjective sentiment analysis: Challenges and research opportunities, Appl. Soft Comput. 98 (2021) 106836. doi:https://doi.org/10.1016/j.asoc.2020.106836.

[6] M. Coccia, Deep learning technology for improving cancer care in society: New directions in cancer imaging driven by artificial intelligence, Technol. Soc. 60 (2020) 101198. doi:https://doi.org/10.1016/j.techsoc.2019.101198.

[7] G. Jeon, M. Anisetti, E. Damiani, B. Kantarci, Artificial intelligence in deep learning algorithms for multimedia analysis, Multimed. Tools Appl. 79 (2020) 34129–34139. doi:10.1007/s11042-020-09232-7.

[8] M.F. Aljunid, M. Dh, An Efficient Deep Learning Approach for Collaborative Filtering Recommender System, Procedia Comput. Sci. 171 (2020) 829–836. doi:https://doi.org/10.1016/j.procs.2020.04.090.

[9] A. Berdeddouch, A. Yahyaouy, Y. Benanni, R. Verde, Deep Based Recommender System For Relevant K Pick-up Points, in: 2020 Int. Conf. Intell. Syst. Comput. Vis., 2020: pp. 1–7. doi:10.1109/ISCV49265.2020.9204065.

[10] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, H. Shah, Wide & Deep Learning for Recommender Systems, (2016) 1–4. doi:10.1145/3109859.3109933.

[11] P. Covington, J. Adams, E. Sargin, Deep Neural Networks for YouTube Recommendations, Proc. 10th ACM Conf. Recomm. Syst. - RecSys '16. (2016) 191–198. doi:10.1145/2959100.2959190.

[12] N. Nassar, A. Jafar, Y. Rahhal, A novel deep multi-criteria collaborative filtering model for recommendation system, Knowledge-Based Syst. 187 (2020) 104811. doi:https://doi.org/10.1016/j.knosys.2019.06.019.

[13] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, (2015). doi:10.1007/s13398-014-0173-7.2.