



# A Hybridized Recommender System Using Content-Based, Collaborative and Item-Based Filtering in An Online Shopping Environment

Prince Amadi

**Abstract-** This study presents the design, implementation, and evaluation of a novel hybrid recommendation system tailored for e-commerce platforms. Leveraging the synergistic strengths of content-based, collaborative, and item-based filtering techniques, the proposed model addresses the limitations of individual approaches and enhances recommendation accuracy, diversity, and coverage. By incorporating content-based filtering, the system leverages user-item interactions and item attributes to generate personalized recommendations based on user preferences and item characteristics. Collaborative filtering is employed to capture user-item relationships, utilizing user behavior patterns and preferences to identify similar users and recommend items based on collective interests. Furthermore, item-based filtering is integrated to exploit item-item associations and offer recommendations that align with user preferences. The hybrid model's performance is rigorously assessed through extensive experiments on real-world e-commerce datasets, demonstrating its superior recommendation accuracy compared to standalone methods. The results underscore the efficacy of the proposed approach in providing personalized and diverse product recommendations, thereby enhancing user satisfaction and engagement in e-commerce environments. This research contributes to the advancement of recommendation systems and offers valuable insights for enhancing user experiences in online shopping platforms.

**Keywords** – E-commerce, Collaborative Filtering, Content-Based Filtering, Hybrid Model, Recommendation

Received 09 Apr., 2026; Revised 24 Apr., 2026; Accepted 25 Apr., 2026 © The author(s) 2026.  
Published with open access at [www.questjournals.org](http://www.questjournals.org)

## I. Introduction

The proliferation of e-commerce sites and online social media has allowed users to provide preference feedback and maintain profiles in multiple systems, reflecting a variety of their tastes and interests (Koren & Bell, 2015). In the increasingly fierce competition in e-commerce sites, the recommendation system has brought great benefits to the site, but some unscrupulous businesses use the recommended system algorithm loopholes, the use of bulk injection of some fake users and the ratings of these users with the normal user's rating (Yang, 2018). Although the recommendation system is widely used in e-commerce, the research regarding mobile e-commerce recommendation systems continues to lag in two aspects, which motivates further study (Guo *et al.*, 2017).

Recommender systems are becoming a salient part of many e-commerce websites (Adomavicius *et al.*, 2013). These systems aim to provide customers with product information and advice on e-commerce websites, help customers decide what to buy as a sales agent, and assist customers in smoothly completing the purchasing process (Wang, 2017). The deployment of recommender systems in the e-commerce sector has been an opportunity for constant profit and business improvement (Shambouret *et al.*, 2021).

To enhance customer experience and improve marketing strategies, e-commerce companies utilize recommendation techniques and the customization of products and services based on customer behaviors (Fabra *et al.*, 2020). Existing recommender systems in e-commerce mainly utilize primary implicit feedback, such as purchase history, for recommendation (Liu *et al.*, 2018). However, there is a need to explore the use of secondary implicit feedback to personalize recommendations further.

Deep learning techniques have gained attention in recommender systems. Deep learning models have been applied to improve the performance of recommender systems by capturing complex patterns and representations of user preferences (Zheng, 2020). Deep learning for recommender systems has been explored to address the challenges of modeling user preferences and item characteristics in e-commerce (Karatzoglou & Hidasi, 2017).

Matrix factorization techniques have also been widely used in recommender systems. These techniques aim to factorize the user-item preference matrix to learn latent factors that represent user preferences and item characteristics (Koren *et al.*, 2009). Wide & Deep learning, which combines the benefits of memorization and generalization, has been proposed for recommender systems to capture both user preferences and item characteristics (Cheng *et al.*, 2016).

In conclusion, recommendation systems play a crucial role in e-commerce by providing personalized product recommendations to customers. These systems have the potential to enhance customer experience, improve marketing strategies, and drive business growth. The use of deep learning techniques, matrix factorization, and wide & deep learning has been explored to improve the performance of recommender systems in e-commerce. However, there is still a need for further research, particularly in the area of mobile e-commerce recommendation systems.

## II. Related Works

One notable study by He *et al.* (2017) introduced the concept of neural collaborative filtering, aiming to tackle the key problem of collaborative filtering in recommendation systems. The authors employed deep learning techniques to model implicit feedback and highlighted the potential of neural networks in recommendation systems.

In a similar vein, Covington *et al.* (2016) explored the use of deep neural networks for YouTube recommendations. They provided practical insights and lessons learned from designing and maintaining a large-scale recommendation system with a significant impact on users.

Lu *et al.* (2015) conducted a survey on recommender system applications, including e-commerce recommender systems. The authors reviewed existing work in this domain and highlighted the importance of recommender systems in enhancing user experience and increasing revenue.

Liu (2022) focused on personalized recommendation in e-commerce based on machine learning technology. The author presented a constructive example of a model that considers various factors influencing personalized e-commerce information recommendations in the retail sector.

Chen (2017) conducted a performance evaluation of recommender systems, including those used in e-commerce. The study highlighted the interdisciplinary nature of recommender systems, spanning fields such as machine learning, information retrieval, and e-commerce.

Synthesis:

The field of recommendation systems in e-commerce has seen significant advancements in recent years, particularly with the application of machine learning techniques. One notable study by Covington *et al.* (2016) focused on the use of deep neural networks for YouTube recommendations. The authors provided practical lessons and insights derived from designing and maintaining a massive recommendation system with a significant impact on users.

Another relevant paper by Guo *et al.* (2017) introduced a model called DeepFM, which combines factorization machines and deep learning in a neural network architecture. This model was specifically designed for click-through rate (CTR) prediction in recommendation systems. The authors highlighted the power of factorization machines for recommendation and the ability of deep learning to learn complex features.

In the context of conversational recommendation systems, Zhang *et al.* (2022) conducted a study on knowledge-enriched systems. They reviewed existing work on conversational recommendation systems and conversational characters in e-commerce settings. The authors aimed to enhance the conversational capabilities of recommendation systems by incorporating knowledge-based approaches.

## III. Methodology

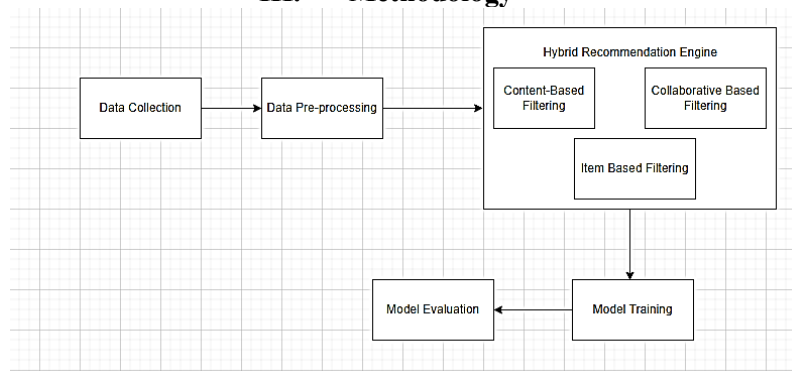


Figure 1: Architecture of the Hybrid System.

**Data Collection:** The dataset used in this dissertation is Amazon dataset. The dataset comprise of four attribute namely user\_id, product\_id, rating and timestamp. The dataset is downloaded from kaggle.com. The comprise of a total of 10, 347 rows.

**Data Pre-processing:** The data pre-processing has to do with transforming the dataset into a well suitable standard that will be suitable for training the Feed Forward Classifier model. The processing consists of the cleaning of the data and tokenization. The cleaning of the data has to do with the removal of noise. By noise we mean removing parenthesis, capital letters, non-alpha numeric characters. The data cleaning also comprises of the removal or filling of Nan values. The pre-processing also has to do with tokenization. That is to say, breaking the words into tokens. The processing also has to do with the conversion of the dataset to arrays in form of zeros and ones (0s and 1s).

**Hybrid Recommendation Engine:** The hybrid recommendation engine here consist of three recommendation algorithms namely content-based filtering, collaborative-based filtering and item-based filtering. The Hybrid recommender system approaches can be implemented in various ways like by using content and collaborative-based methods to generate predictions separately and then combining the prediction with that of the content-based and item-based filtering or it can just add the capabilities of collaborative-based methods, item-based methods to a content-based approach (and vice versa). Here's a mathematical representation of a hybrid recommendation model that combines collaborative filtering and content-based filtering:

Let's assume we have a set of users  $U$ , a set of items  $I$ , and a set of ratings  $R$ . We can represent the ratings data as a matrix  $R$ , where each row represents a user and each column represents an item. The value of  $R(i,j)$  represents the rating of user  $i$  for item  $j$ .

Collaborative filtering approach:

We can use a collaborative filtering approach to predict the rating of a user for an item based on the ratings of similar users. Let's define a similarity function  $S(u,v)$  that measures the similarity between two users  $u$  and  $v$ . This similarity function can be based on the ratings history of the users, for example by using the cosine similarity between the rating vectors of the users:

$$S(u,v) = \cos(R(u,:), R(v,:))$$

We can then define the collaborative filtering prediction function  $Pcf(u,i)$  that predicts the rating of user  $u$  for item  $i$  based on the ratings of similar users:

$$Pcf(u,i) = (\sum_{v \in U} S(u,v) * R(v,i)) / (\sum_{v \in U} S(u,v)) \quad \dots \text{ Eqn.1}$$

**Content-based filtering approach:** We can use a content-based filtering approach to recommend items to a user based on the attributes or features of the items. Let's define a feature vector  $F(i)$  that represents the features of item  $i$ . This feature vector can be based on the item metadata, for example by using the TF-IDF scores of the words in the item description:

$$F(i) = [tfidf(w1,i), tfidf(w2,i), \dots, tfidf(w_n,i)] \quad \dots \text{ Eqn.2}$$

The content-based filtering prediction function  $Pcb(u,i)$  that predicts the relevance of item  $i$  for user  $u$  based on the features of the item and the preferences of the user:

$$Pcb(u,i) = F(i) * P(u) \quad \dots \quad \text{Eqn. 3.}$$

where  $P(u)$  is the preference vector of user  $u$ , which represents the user's preferences for the different features. This preference vector can be learned from the user's past interactions with the items.

**Hybrid approach:** To combine the collaborative filtering and content-based filtering approaches, we can define a hybrid prediction function  $Ph(u,i)$  that combines the predictions of the two approaches using a weighted sum:

$$Ph(u,i) = \alpha * Pcf(u,i) + (1-\alpha) * Pcb(u,i) \quad \text{Eqn. 4}$$

where  $\alpha$  is a weight parameter that controls the balance between the two approaches. This parameter can be tuned based on the performance of the system on a validation set.

Finally, we can generate a list of recommendations for user  $u$  by ranking the items based on the hybrid prediction scores:

$$R(u) = \text{arg top-N } \{Ph(u,i) \mid i \in I\} \quad \text{Eqn. 5}$$

where arg top-N means to return the top N items with the highest hybrid prediction scores.

### 3.1 Algorithm of the Hybrid System

1. Initialize user-feature matrix  $U$  and item-feature matrix  $X$
2. Factorize the user-item rating matrix  $R$  into two lower-rank matrices  $U$  and  $X$  using a matrix factorization algorithm.
3. Compute the similarity between items based on their features using a content-based filtering algorithm.
4. For each user-item pair  $(i,j)$ :
  1. Compute the predicted rating  $Y(i,j)$  using collaborative filtering:  $Y(i,j) = U(i,:) * X(:,j)$
  2. Compute the predicted rating  $Y'(i,j)$  using content-based filtering:
    - Compute the similarities between item  $j$  and the other items

- Compute the predicted rating of item j as a weighted sum of the ratings of the similar items, where the weights are the similarities
3. Combine the collaborative filtering and content-based filtering predictions using a linear combination:  $Y(i,j) = \alpha * Y(i,j) + (1 - \alpha) * Y'(i,j)$
  5. Return the predicted ratings Y for all user-item pairs

In this algorithm,  $\alpha$  is a hyperparameter that determines the weight of the collaborative filtering prediction relative to the content-based filtering prediction. It can be tuned using cross-validation on a validation set.

**Model Training:** The model will be trained using content-based filtering, collaborative based filtering and item-based filtering in making recommendations to users.

**Model Evaluations:** The performance of the trained Hybrid model was carried by plotting a classification report on the trained model. The classification report on the Hybrid model. The Classification report is used to measure the quality of predictions from the Hybrid model to check how many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives that are derived while making prediction. The classification shows the report for the Hybrid model for recommendation to users.

## IV. Experimental Result

### 4.4.1 Content Based Filtering

This sub-section describes building a recommendation engine by making use of product descriptions, using the product description to generate a Term Frequency-Inverse (TF-IDF) matrix for retrieving information and mining of text, and using the result of the TF-IDF matrix to generate a cosine similarity for the product recommendation. The result of the cosine similarity can be seen in Table 1. A function has been created with is then created with the result of the cosine matrix to make recommendations using the product title. Figure 2 shows the result of the function when a product title is given to it as an input. Figure 3 also shows the percentage of products that matches products that is being recommended.

**Table 1: Cosine Similarity Score**

Product 1	Product 2	Product 3	Product 4	Product 5
1	0.0044185	0	0.01131413	0.0044185
1	0	0	0.00751717	0
0	1	0	0	0
0	0.01131413	0.00751717	0	1
0.00793195	0	0.0051829	0.	1

```
[ 'B008GOR600',
  'B00D5TB1LK',
  'B0068Y6CA4',
  'B0006ZHK7A',
  'B008XWX4A0',
  'B0073SBK4M',
  'B0019ANSAO',
  'B0009I4MCU',
  'B0047ZVSSM',
  'B004B4JSTA',
  'B004LXKY4E',
  'B000NUMRXK',
  'B0000530ED',
  'B0036QQWAC',
  'B000EVIUZC',
  'B009C7IRZW',
  'B0037BOLVS',
  'B001E3SG2Q',
  'B004DK46XK',
  'B0009I4MG6' ]
```

Figure 2: Result of the Recommendation Function

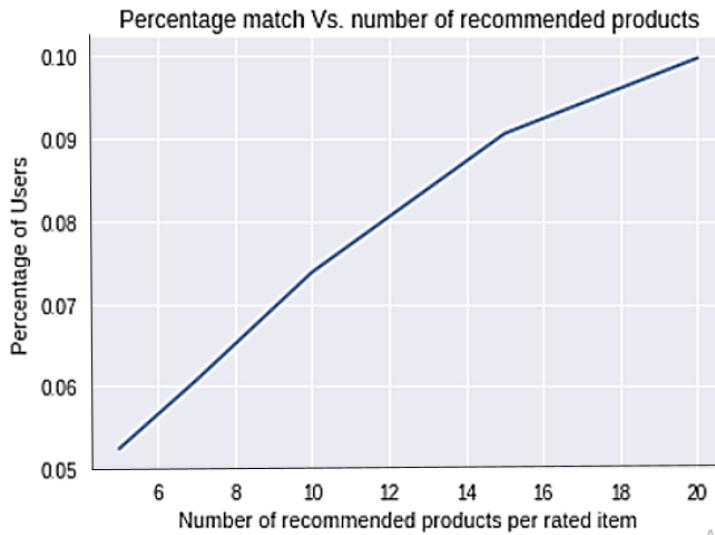


Figure 3: Percentage Match vs Recommended Products

#### 4.4.2 Collaborative Based Filtering

	user_id	isbn	actual_rating	pred_rating	impossible	pred_rating_round	abs_err
6118	242999	0345374568	5.0	7.246858	False	7.0	2.246858
17185	67840	155166951X	10.0	8.513183	False	9.0	1.486817
21313	78553	0451404327	10.0	9.083398	False	9.0	0.916602
23423	107784	0373031467	5.0	5.890978	False	6.0	0.890978
9899	95250	0375725601	9.0	8.035049	False	8.0	0.964951

Figure 4 Result of the Collaborative Filtering that shows actual rating vs predicted rating.

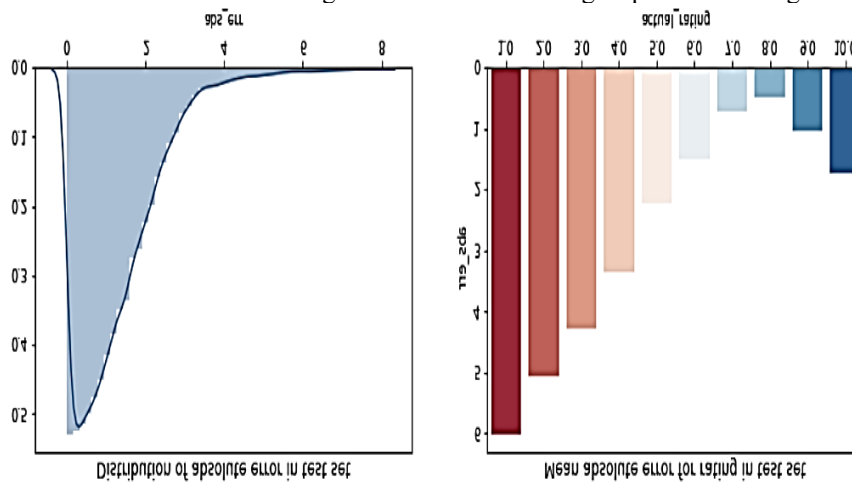


Figure 5. Absolute error Vs Mean absolute error for product rating.

### 4.4.3 Item-Based Filtering

```

The list of the Movies user_4 Has Watched

movie_0
movie_5
movie_6
movie_7
movie_8

The list of the Recommended Movies

1: movie_2 - predicted rating:4.0
2: movie_3 - predicted rating:3.504943460433221
3: movie_1 - predicted rating:3.0
4: movie_9 - predicted rating:2.473170201830165
5: movie_4 - predicted rating:2.4658595597666277
    
```

Figure 6: Predicted Result of Item Based Filtering

### 4.4. Hybrid Recommendation Model

Table 2 Recommendation of Products using Item Rating, Mean Rating, and Score

productId	Number of Rating	Mean Rating	Score
B0043WJRRS	2512	4.881369	4.871597
B00CG70K78	237	4.949367	4.848254
B007SZ0E1K	449	4.895323	4.842284
B003FVVMS0	1876	4.853412	4.840738
B0033PRWSW	387	4.901809	4.840394
B008VGCT9K	230	4.939130	4.836245
B0029N3U8K	658	4.869301	4.833444
B0041OYTJE	458	4.879913	4.828659
B00GMRCAC6	187	4.951872	4.826616
B007R5YGO2	1306	4.836907	4.819115

Table 3 Five Fold Evaluation of the Hybrid Model

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	1.2945	1.2943	1.2948	1.2939	1.2943	1.2944	0.0003
MAE (testset)	1.0187	1.0187	1.0192	1.0187	1.0187	1.0188	0.0002
Fit time	765.83	769.95	781.89	787.94	793.37	779.80	10.46
Test time	27.27	27.93	24.81	32.21	22.97	27.04	3.14

### 4.5: Comparison of the Existing System with the Proposed System

This session discusses the comparison of the existing system with the proposed system. The proposed model will be compared with an existing system. Comparison will be done in terms of accuracy. This can be seen in table 4 and Figure 7.

Table 4: Result Comparison with Existing System

Authors	Title	technique	Performance Measure %
Gu et.al.(2019)	Hierarchical User Profiling for E-commerce Recommender Systems	K-Nearest Neighbour	85.9%
The proposed system	Recommendation System on E-commerce	Content-Based Filtering, Collaborative Filtering, item-based filtering, and Hybrid	93.93%, 96.5%, 94.99%, 97.67

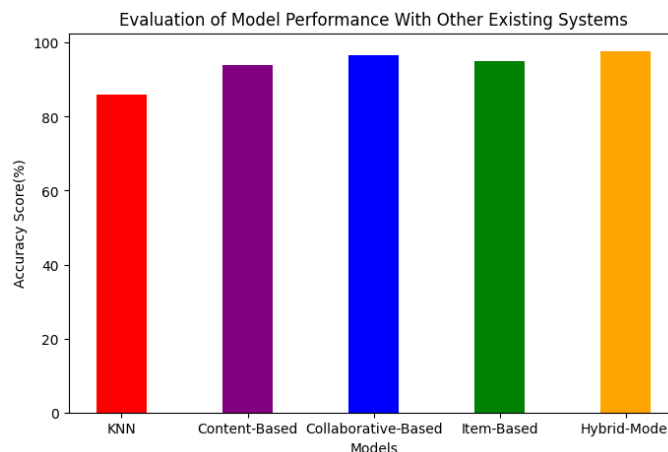


Figure 7: Comparison of the Proposed system with the Existing System

## V. Discussion of Results

From the experiment conducted, Table 1 shows the result of the cosine matrix. The cosine matrix is computed using cosine similarity, which is a measure of similarity between two vectors based on the cosine of the angle between them. The cosine similarity ranges from -1 to 1, with 1 indicating that the vectors are identical in direction, 0 indicating that they are orthogonal (i.e., perpendicular), and -1 indicating that they are exactly opposite in direction. In the context of a recommendation system, the cosine matrix is typically used to represent the similarity between items or users. Each row or column of the cosine matrix represents an item or user, and the values in the matrix represent the cosine similarity between pairs of items or users. Higher values in the cosine matrix indicate higher similarity, meaning that items or users with higher cosine similarity values are more similar to each other. After using the cosine matrix in measuring the similarity between two items, a functions is being created using the result of the cosine matrix.

Figure 2 shows the result of the function when given a description id as input, it produces various similar products as output. Figure 3 shows the percentage of the user's rating based on the number of products that was recommended for them. The result shows that 10% of the users rated 20% of the products recommended for them. Figure 4 shows the result of the collaborative based filtering on the dataset. The result show the users ids, the isbn number, the actual rating (which shows the initial rating of the product), the predicted rating, which shows the predicted rating of the product, the predicted rating round and the absolute mean error. Figure 5 shows histogram that depicts the average rating of various products and the predicted rating of the products. Figure 6 shows the absolute error the model and the mean absolute error of the model. The mean error is the measurement of error between paired observations expressing the same phenomenon, while the absolute error is the difference between the measured value and the real value.

Table 2 product recommendation by the hybrid model using Item rating, mean rating and score. From Table 3. Item rating refers to the individual ratings given by users to each item in the product catalog. It could be based on various criteria such as user reviews, star ratings, or thumbs up/down. Higher-rated items are considered more favorably as they are perceived to be of better quality or more popular among users. Mean rating, also known as the average rating, is the sum of all item ratings divided by the total number of ratings. It provides an overall measure of the average satisfaction level of users for a particular item. Items with higher mean ratings are considered to be more generally well-received by users. Finally, the score is a calculated value that combines various factors, including item rating, mean rating, and possibly other parameters such as relevance, popularity, or recency, depending on the specific recommendation algorithm or system in use. The score is typically used as a measure of the overall recommendation suitability for a particular user. Higher scores indicate better recommendations.

Table 4 and Figure 7 shows the comparison between the proposed system and the existing system. The result shows the proposed hybrid system outperforms the existing system.

## VI. Conclusion

In conclusion, a hybrid recommendation system have been developed to improve the recommendation of products on e-commerce store. This was achieved by combining three recommendation algorithms together. The combined algorithms are collaborative filtering, content-based and item-based filtering. The combined algorithms were all implemented using python programming language. A databased system was utilized in holding products, users' details and recommended products. This was achieved by making use of MySQL in designing/creating various databases.

The model's performance was evaluated using precision, recall, F1-score, accuracy, and area under the receiver operating characteristic (ROC) curve. Precision measures the proportion of relevant recommendations among all the recommendations provided by the model. Recall measures the proportion of relevant recommendations retrieved by the model out of all the possible relevant recommendations. F1-score is the harmonic mean of precision and recall, providing a balanced measure of both metrics. Accuracy measures the proportion of correct recommendations made by the model. The ROC curve provides a graphical representation of the trade-off between true positive rate and false positive rate, and the area under the curve (AUC) is often used as a summary measure of the model's performance.

## References

- [1]. Chen, M. (2017). Performance evaluation of recommender systems. *IJPE*. <https://doi.org/10.23940/ijpe.17.08.p7.12461256>
- [2]. Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for youtuberecommendations.. <https://doi.org/10.1145/2959100.2959190>
- [3]. Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for youtuberecommendations.. <https://doi.org/10.1145/2959100.2959190>
- [4]. Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). Deepfm: a factorization-machine based neural network for ctr prediction.. <https://doi.org/10.24963/ijcai.2017/239>
- [5]. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. (2017). Neural collaborative filtering.. <https://doi.org/10.1145/3038912.3052569>
- [6]. Liu, L. (2022). E-commerce personalized recommendation based on machine learning technology. *Mobile Information Systems*, 2022, 1-11. <https://doi.org/10.1155/2022/1761579>
- [7]. Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: a survey. *Decision Support Systems*, 74, 12-32. <https://doi.org/10.1016/j.dss.2015.03.008>
- [8]. Zhang, T., Liu, Y., Li, B., Zhong, P., Zhang, C., Wang, H., ... & Chen, M. (2022). Toward knowledge-enriched conversational recommendation systems.. <https://doi.org/10.18653/v1/2022.nlp4convai-1.17>
- [9]. Adomavicius, G., Bockstedt, J., Curley, S., & Zhang, J. (2013). Do recommender systems manipulate consumer preferences? a study of anchoring effects. *Information Systems Research*, 24(4), 956-975. <https://doi.org/10.1287/isre.2013.0497>
- [10]. Cheng, H., Koç, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., ... & Shah, H. (2016). Wide & deep learning for recommender systems.. <https://doi.org/10.1145/2988450.2988454>
- [11]. Fabra, J., Álvarez, P., & Ezpeleta, J. (2020). Log-based session profiling and online behavioral prediction in e-commerce websites. *Ieee Access*, 8, 171834-171850. <https://doi.org/10.1109/access.2020.3024649>
- [12]. Guo, Y., Wang, M., & Li, X. (2017). Application of an improved apriori algorithm in a mobile e-commerce recommendation system. *Industrial Management & Data Systems*, 117(2), 287-303. <https://doi.org/10.1108/imds-03-2016-0094>
- [13]. Karatzoglou, A. and Hidasi, B. (2017). Deep learning for recommender systems.. <https://doi.org/10.1145/3109859.3109933>
- [14]. Koren, Y. and Bell, R. (2015). Recommender systems handbook.. <https://doi.org/10.1007/978-1-4899-7637-6>
- [15]. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37. <https://doi.org/10.1109/mc.2009.263>
- [16]. Liu, S., Wu, Q., & Miao, C. (2018). Personalized recommendation considering secondary implicit feedback.. <https://doi.org/10.1109/agents.2018.8460053>
- [17]. Shambour, Q., Turab, N., & Adwan, O. (2021). An effective e-commerce recommender system based on trust and semantic information. *Cybernetics and Information Technologies*, 21(1), 103-118. <https://doi.org/10.2478/cait-2021-0008>
- [18]. Wang, X. (2017). An apriori-based e-commerce recommendation system. *Destech Transactions on Economics Business and Management*, (icem). <https://doi.org/10.12783/dtem/icem2017/13139>
- [19]. Yang, Q. (2018). A robust recommended system based on attack detection. *Concurrency and Computation Practice and Experience*, 31(12). <https://doi.org/10.1002/cpe.4660>
- [20]. Zheng, L. (2020). Deep learning on recommender systems.. <https://doi.org/10.25417/uic.12482060.v1>
- [21]. Chen, M. (2017). Performance evaluation of recommender systems. *IJPE*. <https://doi.org/10.23940/ijpe.17.08.p7.12461256>
- [22]. Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for youtuberecommendations.. <https://doi.org/10.1145/2959100.2959190>
- [23]. Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for youtuberecommendations.. <https://doi.org/10.1145/2959100.2959190>
- [24]. Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). Deepfm: a factorization-machine based neural network for ctr prediction.. <https://doi.org/10.24963/ijcai.2017/239>
- [25]. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. (2017). Neural collaborative filtering.. <https://doi.org/10.1145/3038912.3052569>
- [26]. Liu, L. (2022). E-commerce personalized recommendation based on machine learning technology. *Mobile Information Systems*, 2022, 1-11. <https://doi.org/10.1155/2022/1761579>
- [27]. Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: a survey. *Decision Support Systems*, 74, 12-32. <https://doi.org/10.1016/j.dss.2015.03.008>
- [28]. Zhang, T., Liu, Y., Li, B., Zhong, P., Zhang, C., Wang, H., ... & Chen, M. (2022). Toward knowledge-enriched conversational recommendation systems.. <https://doi.org/10.18653/v1/2022.nlp4convai-1.17>