Quest Journals
Journal of Electronics and Communication Engineering Research
Volume 9 ~ Issue 5 (2023) pp: 40-47
ISSN(Online) : 2321-5941
www.questjournals.org

Research Paper

Recommendation System in Online Sales Using Cooperative Filtering Algorithm

Tran Thi Thanh
Department of Computer engineering, Faculty of Electronic Engineering, Thai Nguyen University of Technology, Thai Nguyen, 250000, Vietnam
Corresponding author: Tran Thi Thanh

Abstract
The growth of online shopping has been a great driving force for the development of e-commerce platforms. The integration of recommendation systems aims to provide information to help users decide which products to choose based on the number of products sold, the user's personal information, and the analysis of the previous purchase behavior of the user to make predictions about the future purchase behavior of the customer. In this article, a product recommendation system using a collaborative filtering algorithm is built on the Asp.net MVC platform and posted on Lazada online sales website. The obtained results show that the system has met the suggestion requirements with quite high accuracy for different user objects. In addition, the system is deployed on the host to collect more data with a user-friendly interface, and product data is updated regularly, fully and displayed suitable for from desktop to mobile devices.
Keywords: *Recommender systems, algorithms, collaborative filtering, online sales*

R*eceived 17 May, 2023;* ***R****evised 28 May, 2023;* ***A****ccepted 31 May, 2023 © The author(s) 2023.*
P*ublished with open access at* ***www.questjournals.org***

I. INTRODUCTION

Recommender Systems or Recommendation Systems (RS) is a form of decision support system that provides a personalized solution without having to go through a complicated search process. The recommender system learns from the users and suggests the best products among the suitable products [1].

The recommender system uses the knowledge of products, experts, or users' behavior to make recommendations about products they like in thousands of products in the system. E-commerce websites of books, movies, music, newspapers, etc., use a recommendation system to provide information to help users decide which product to choose [2]. Suggested products are based on the number of products that have been sold, the personal information of the user, and the analysis of the previous purchase behavior of the user to make predictions about the customer's future purchase behavior. Suggestion types include: recommending products to consumers, personalized product information, summarizing community opinions, and providing sharing, criticism, and community evaluation related to the requirements and purposes of that user.

In recent years, with the rapid growth of e-commerce platforms, especially the impact of the Covid 19 epidemic, now online shopping has become a popular trend. Therefore, the demand for recommender systems is increasing day by day and is widely applied. All recommender systems require a user model. A common approach to building a user model is through user feedback. Therefore, user feedback to the system becomes more and more important. With product introduction or review websites, users often have problems finding the right product set for their needs due to the large number of products and limited time. A normal recommendation system can give you suggestions that interest you, but after looking at the product information, you notice that a certain attribute does not match your requirements for that product [3 ,4]. It is worth mentioning here that you have to fumble and take a lot of time. And your patience may be limited and you get nothing. The recommender system can be found to be quite effective, but it can do even better. The suggestion system in this article will solve the above problem, and at the same time increase the effectiveness of suggestions and user satisfaction [5].

When users use the recommendation system, can they actually tell the system what they need, how to search for products? From there, the system will give a list of suggestions that best match that request of the user. This will help increase system efficiency, save time for users and increase satisfaction when using the system. The system's suggested product list will be based on what the user has responded to, what the user is viewing. In other words, the system will stick to user preferences to make suggestions.

In RS, the feedback value r_{ui} of each user on the item will be recorded as a basis for predicting the next values. Depending on the system, this value will have different meanings. For example, it can be used to measure "fitness" or "likeness" (usually reviews on products) in brand systems. e-commerce or the “competence/performance” of users in e-learning systems.

The r_{ui} value can be specified explicitly (explicit feedbacks) such as through rating/rating (for example, rating from ★ to ★★★★★; or like (1) and dislike (0)) that *u* voted for *i*; or r_{ui} can be determined implicitly (implicit feedbacks) through the number of mouse clicks, the time that *u* browsed/viewed *i*, etc.

There are two main types of problems in RS: rating prediction of systems with explicit feedback as described above, and item prediction/recommendation, which is the determination of probabilities. that the user likes the corresponding item [6].

Currently, in RS, there are many proposed algorithms, but there are some main groups as follows [7-10]:

a) Group of collaborative filtering algorithms. The main algorithms are used:

● Neighborhood-based approach, also known as Memory-based, which is either based on past data of users "similarity" (user-based approach), or is based on data past of "similar" items (item-based approach)

● Model-based approach. This group is concerned with building predictive models based on data collected in the past such as Bayesian models, latent factor models), in which matrix factorization is a typical example.

b) Content-based Filtering: Recommending items based on user profiles or based on the content/attributes of items similar to those of the user selected in the past.

c) Hybrid approach. The combination of both methods above.

d) Non-personalization technical group.

One of the disadvantages of the content-based filtering method is the difficulty in collecting information, while most models based on collaborative filtering only need three information types (user id, item id, feedback) to function well. Therefore, the current trend of studies favors collaborative filtering groups [2]. The main content of this paper is to study and build a product recommendation system in online sales using collaborative filtering algorithm.

II. MATERIAL AND METHOD

The sales website will be built on the Asp.net MVC platform so that it can take advantage of the support library system and outstanding features of management and statistics. For the sales website model, the items correspond to the products, the users are divided into two types: users with registered accounts, distinguished by user_id and having related information stored in the database. and the user does not register an account. We call the first type of user a member and the other a visitor. After selecting any product, only registered members can comment. The rating level varies from 1 to 5 stars, with 1 star being the least favorite and 5 stars being the very favorite, product view data and customer purchases are also saved in the database. database, serving for suggestions.

In the database, the factors will be interested in the following tables (Figure 1):

• Productrating: stores the reviews, the number of stars for each product, and the ids of the users who have commented on that product.

• Productviewbehavior: store the number of product views.

• order_user: store the product name and number of products that the user has purchased.

From the three tables above, the following necessary information is queried:

• productId: The product's ID.

• rating: Rating of the product.

• Countbuy: Purchase quantity of each product.

• Countview: The number of views of each product.

• userId: The user's ID.

For the Matrix Factorization method, the model was built from all the above data.

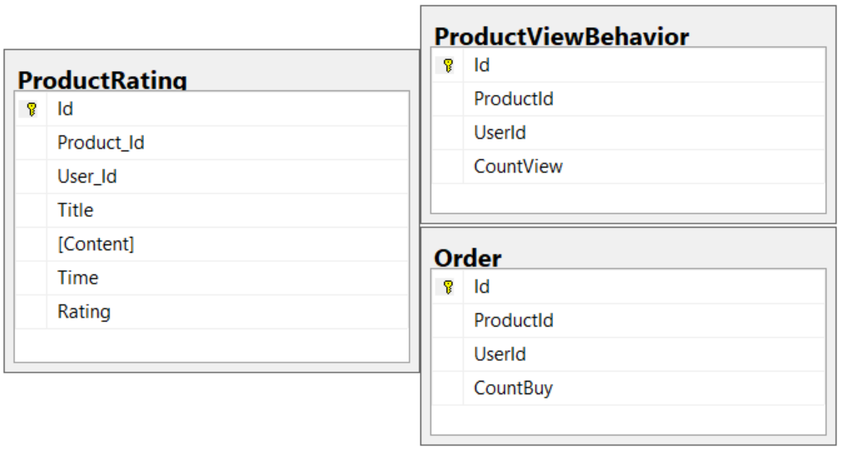


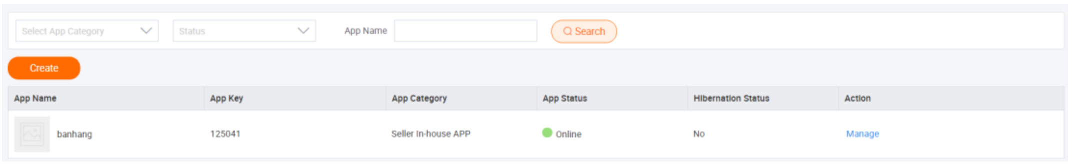
Figure 1. Structure of tables containing input data

For the Neighborhood-based Collaborative Filtering (NBCF) method, the Utility matrix was built and the data used includes:
• productId: The product's ID.
• rating: Rating of the product.
• userId: The user's ID.
After having the Utility matrix, two algorithms will be used to find the ids of the products that have the highest similarity with the products that users have rated by each user. We will recommend these products to the respective users.
The movie recommendation system uses Collaborative Filtering based on the user's past data, the disadvantage is that it only applies to users who already have data, so there are two proposed options:

- In case of the first time of visiting, the system will suggest to the visitor the products with the highest weight.
- If the data of the visitors is available, the system will suggest them the products with the highest similarity to that product.

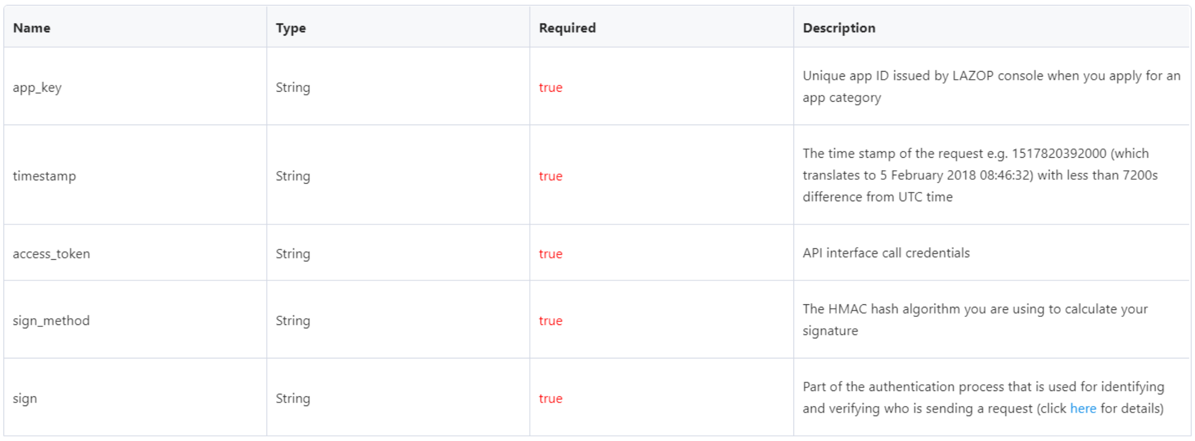
III. RESULTS AND DISCUSSION

Lazada Open Platform is an integrated platform that provides end-to-end API development, provisioning, upgrading and operations. It allows you to programmatically exchange Lazada seller data about products, orders, package shipping and finance, and provides a development environment for you to integrate your application with the Lazada system through via API, improve your application development efficiency and quality, and share your application to Lazada sellers. The steps are used to put the products on Lazada as follows: (1) Initialize the Api connection application (Figure 2); (2) Get Access Token, to grant access to the seller; (3) Create a Post request with the request parameters sent to the server (Figure 3); (4) The server returns a responsive response containing success or failure results (Figures 4, 5).



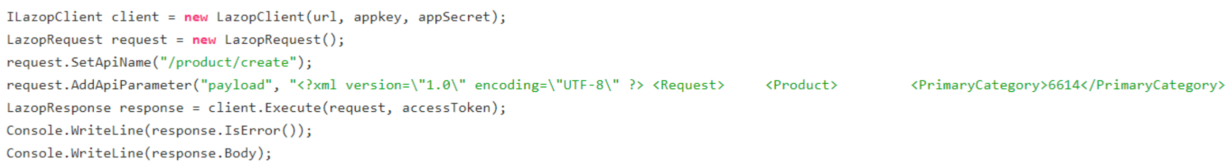
App Name	App Key	App Category	App Status	Hibernation Status	Action
banhang	125041	Seller In-house APP	● Online	No	Manage

Figure 2. Api connection application initialization



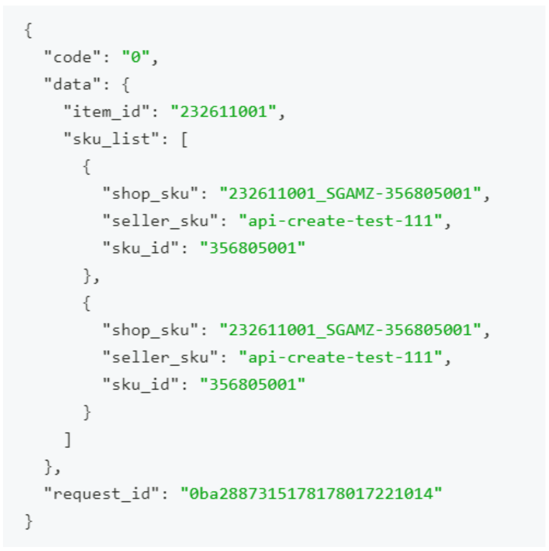
Name	Type	Required	Description
app_key	String	true	Unique app ID issued by LAZOP console when you apply for an app category
timestamp	String	true	The time stamp of the request e.g. 1517820392000 (which translates to 5 February 2018 08:46:32) with less than 7200s difference from UTC time
access_token	String	true	API interface call credentials
sign_method	String	true	The HMAC hash algorithm you are using to calculate your signature
sign	String	true	Part of the authentication process that is used for identifying and verifying who is sending a request (click here for details)

Figure 3. The request parameter to the server



```
ILazopClient client = new LazopClient(url, appkey, appSecret);
LazopRequest request = new LazopRequest();
request.SetApiName("/product/create");
request.AddApiParameter("payload", "<?xml version=\"1.0\" encoding=\"UTF-8\" ?> <Request>     <Product>         <PrimaryCategory>6614</PrimaryCategory>
LazopResponse response = client.Execute(request, accessToken);
Console.WriteLine(response.IsError());
Console.WriteLine(response.Body);
```

Figure 4. Code to call api for the product creation



```
{
  "code": "0",
  "data": {
    "item_id": "232611001",
    "sku_list": [
      {
        "shop_sku": "232611001_SGAMZ-356805001",
        "seller_sku": "api-create-test-111",
        "sku_id": "356805001"
      },
      {
        "shop_sku": "232611001_SGAMZ-356805001",
        "seller_sku": "api-create-test-111",
        "sku_id": "356805001"
      }
    ]
  },
  "request_id": "0ba2887315178178017221014"
}
```

Figure 5. Server content returned to client

A test run was carriedout for the system, and Figure 6 illustrates the home page interface of the system. Figure 7 illustrates the product list interface. When using the Neighborhood-based Collaborative Filtering filter, the results are illustrated in Figure 8. In case of using the Matrix Factorization method, the results are shown in Figure 9. An illustration of the product interface with the most views and purchases is shown in Figures 10, 11. The interface of product suggestions of the similar type is illustrated in Figure 12.

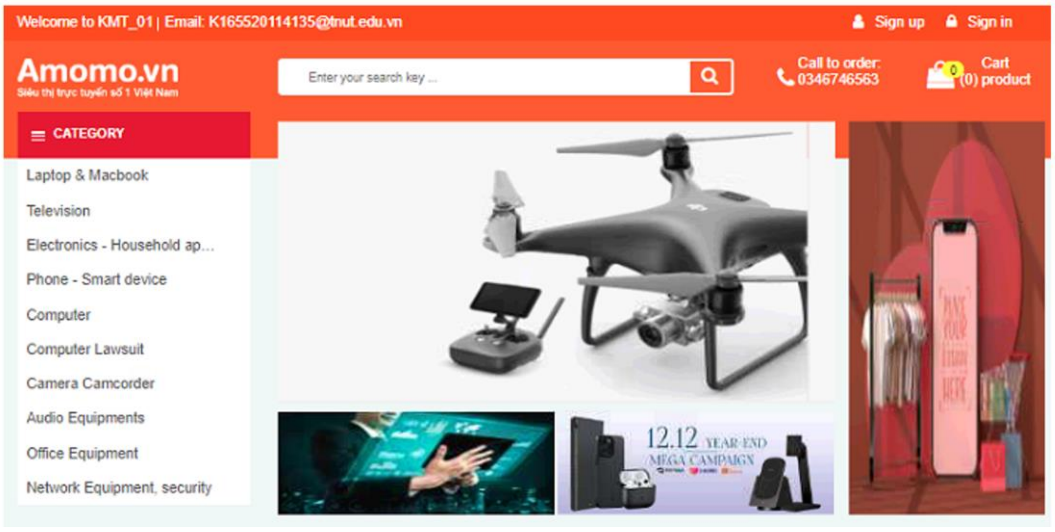

Welcome to KMT_01 | Email: K165520114135@tnut.edu.vn
Sign up
Sign in
Amomo.vn
Siêu thị trực tuyến số 1 Việt Nam
Enter your search key ...
Call to order: 0346746563
Cart (0) product
CATEGORY
Laptop & Macbook
Television
Electronics - Household ap...
Phone - Smart device
Computer
Computer Lawsuit
Camera Camcorder
Audio Equipments
Office Equipment
Network Equipment, security
12.12 YEAR-END MEGA CAMPAIGN

Figure 6. Home page interface of the system

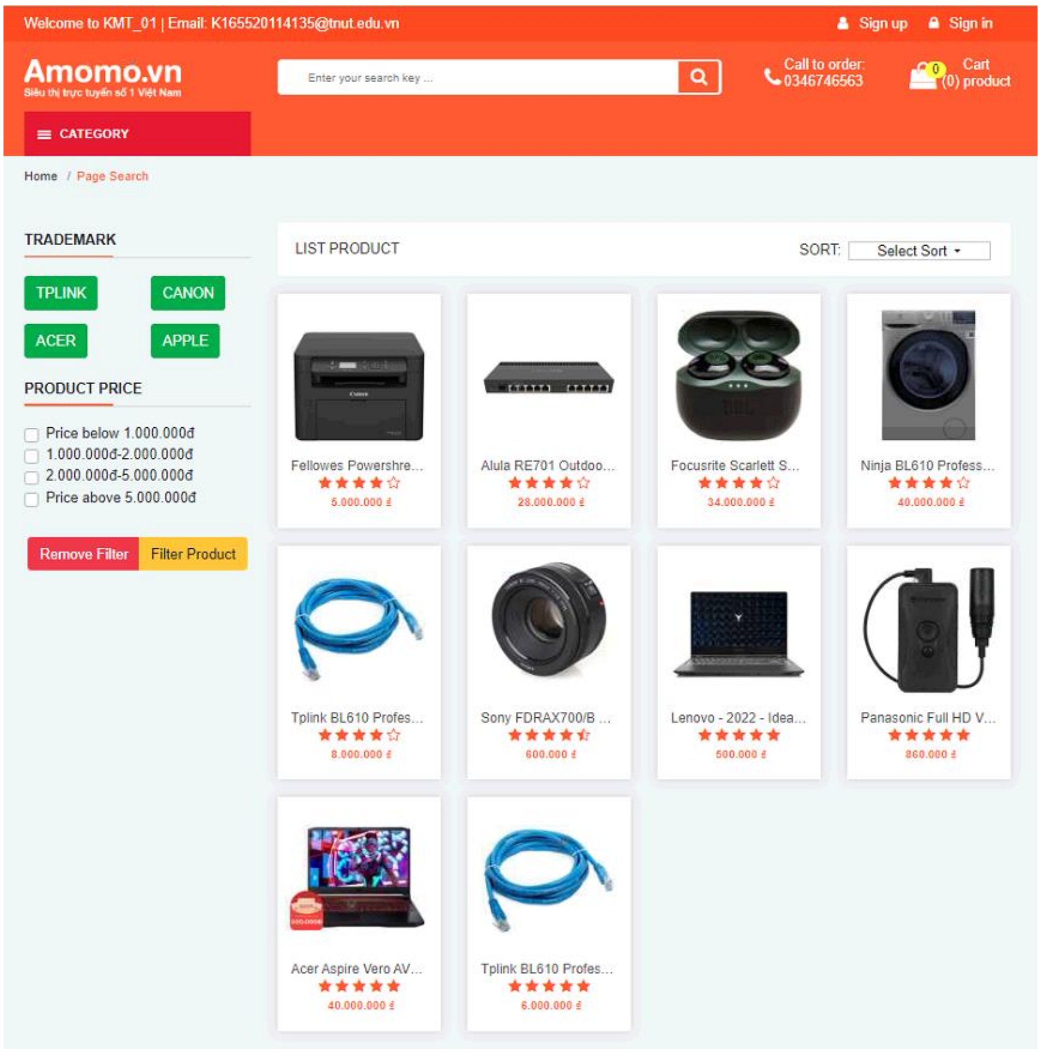
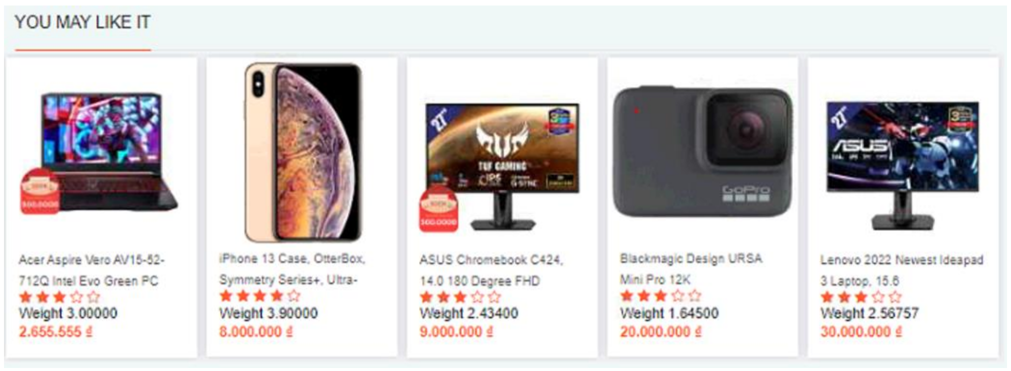

Welcome to KMT_01 | Email: K165520114135@tnut.edu.vn
Sign up
Sign in
Amomo.vn
Siêu thị trực tuyến số 1 Việt Nam
Enter your search key ...
Call to order: 0346746563
Cart (0) product
CATEGORY
Home / Page Search
TRADEMARK
TPLINK
CANON
ACER
APPLE
PRODUCT PRICE
Price below 1.000.000đ
1.000.000đ-2.000.000đ
2.000.000đ-5.000.000đ
Price above 5.000.000đ
Remove Filter
Filter Product
LIST PRODUCT
SORT: Select Sort
Fellowes Powershre... 5.000.000 ₫
Alula RE701 Outdoo... 28.000.000 ₫
Focusrite Scarlett S... 34.000.000 ₫
Ninja BL610 Profess... 40.000.000 ₫
Tplink BL610 Profes... 8.000.000 ₫
Sony FDRAX700/B ... 600.000 ₫
Lenovo - 2022 - Idea... 500.000 ₫
Panasonic Full HD V... 860.000 ₫
Acer Aspire Vero AV... 40.000.000 ₫
Tplink BL610 Profes... 6.000.000 ₫

Figure 7. The product list interface



YOU MAY LIKE IT

Acer Aspire Vero AV15-52-712Q Intel Evo Green PC
★★★☆☆
Weight 3.00000
2.655.555 ₫

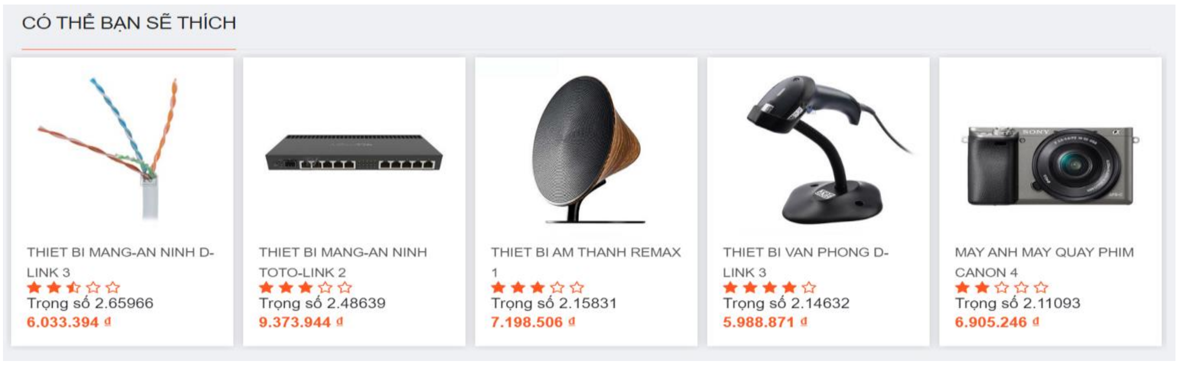
iPhone 13 Case, OtterBox, Symmetry Series+, Ultra-
★★★★☆
Weight 3.90000
8.000.000 ₫

ASUS Chromebook C424, 14.0 180 Degree FHD
★★★☆☆
Weight 2.43400
9.000.000 ₫

Blackmagic Design URSA Mini Pro 12K
★★★☆☆
Weight 1.64500
20.000.000 ₫

Lenovo 2022 Newest Ideapad 3 Laptop, 15.6
★★★☆☆
Weight 2.56757
30.000.000 ₫

Figure 8. Interface results by using Neighborhood-based Collaborative Filtering (NBCF) method



CÓ THỂ BẠN SẼ THÍCH

THIET BI MANG-AN NINH D-LINK 3
★★⯪☆☆
Trọng số 2.65966
6.033.394 ₫

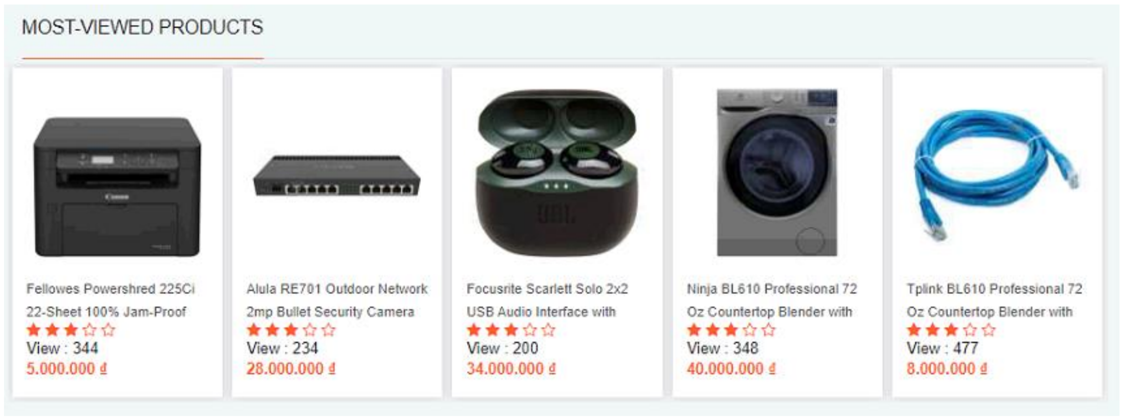
THIET BI MANG-AN NINH TOTO-LINK 2
★★★☆☆
Trọng số 2.48639
9.373.944 ₫

THIET BI AM THANH REMAX 1
★★★☆☆
Trọng số 2.15831
7.198.506 ₫

THIET BI VAN PHONG D-LINK 3
★★★★☆
Trọng số 2.14632
5.988.871 ₫

MAY ANH MAY QUAY PHIM CANON 4
★★☆☆☆
Trọng số 2.11093
6.905.246 ₫

Figure 9. Interface results by using Matrix Factorization method



MOST-VIEWED PRODUCTS

Fellowes Powershred 225Ci 22-Sheet 100% Jam-Proof
★★★☆☆
View : 344
5.000.000 ₫

Alula RE701 Outdoor Network 2mp Bullet Security Camera
★★★☆☆
View : 234
28.000.000 ₫

Focusrite Scarlett Solo 2x2 USB Audio Interface with
★★★☆☆
View : 200
34.000.000 ₫

Ninja BL610 Professional 72 Oz Countertop Blender with
★★★☆☆
View : 348
40.000.000 ₫

Tplink BL610 Professional 72 Oz Countertop Blender with
★★★☆☆
View : 477
8.000.000 ₫

Figure 10. Suggestion interface for most-viewed products

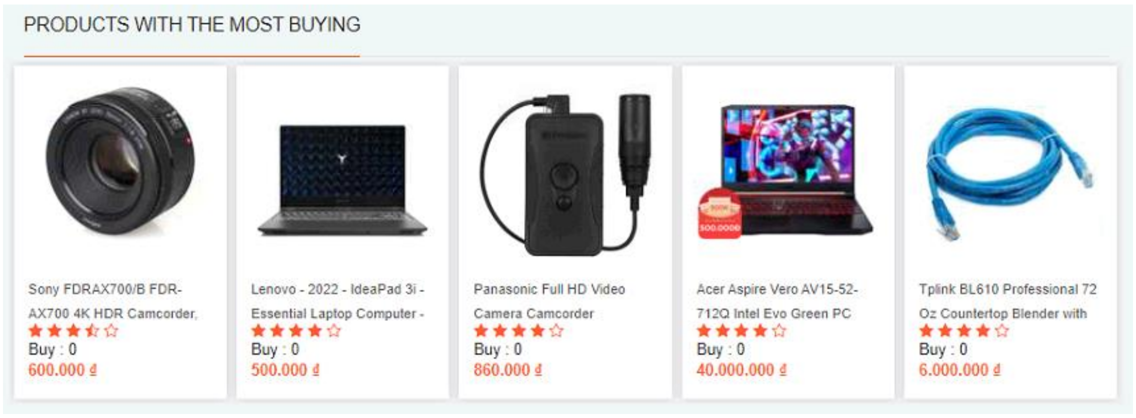

PRODUCTS WITH THE MOST BUYING
Sony FDRAX700/B FDR-AX700 4K HDR Camcorder,
Buy : 0
600.000 ₫
Lenovo - 2022 - IdeaPad 3i - Essential Laptop Computer -
Buy : 0
500.000 ₫
Panasonic Full HD Video Camera Camcorder
Buy : 0
860.000 ₫
Acer Aspire Vero AV15-52-712Q Intel Evo Green PC
Buy : 0
40.000.000 ₫
Tplink BL610 Professional 72 Oz Countertop Blender with
Buy : 0
6.000.000 ₫

Figure 11. Suggestion interface for products with the most buying

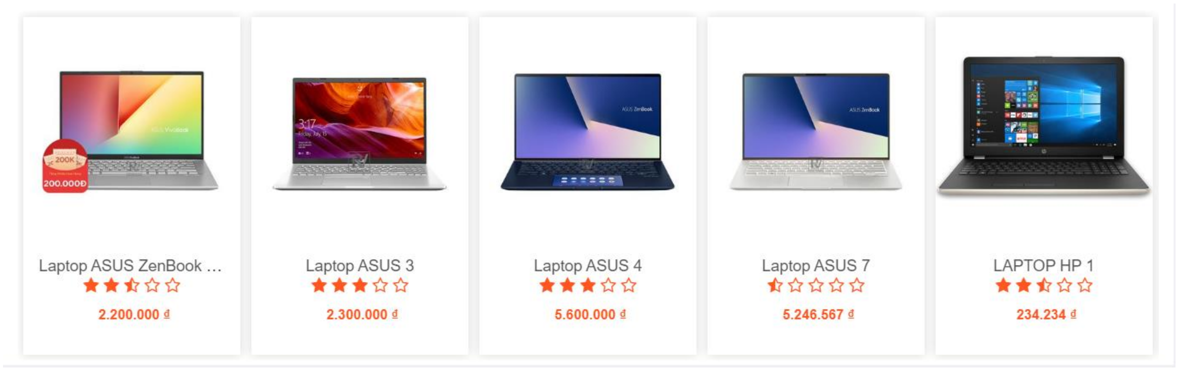


Figure 12. Suggestion interface for similar products

IV. CONCLUSION

In this paper, an integrated recommendation system has been studied and built on an online sales website using a collaborative filtering algorithm. The system has responded to the suggestion request for different user objects, deployed on the host to collect more data. Some of the new contributions of the built-in suggestion system:

- User-friendly interface with the display suitable for different devices from desktop to mobile devices.
- Product data is updated regularly and fully.
- Applying a suggestion algorithm to give highly accurate suggestions to different users.

Some of the shortcomings of the system:

- The Utility matrix must be recalculated every time a new review data is updated, which increases the suggested loading time for each product in the future.
- The item-based Collaborative Filtering recommendation algorithm often encounters a "cold-start" problem: the system cannot find suggestions for products that have never been evaluated by any user.

In further studies, the following issues will be further investigated and developed:

- Optimize the algorithm to reduce the time to calculate the Utility matrix when the amount of input data increases.
- Combine Content - based Filtering and Collaborative Filtering to overcome the "cold - start" situation.
- Integrate many payment methods.
- Add some product management functions in the process of putting products on Lazada.

Acknowledgments

The work presented in this paper is supported by Thai Nguyen University of Technology, Thai Nguyen University, Vietnam.

Reference

[1]. Katarya, R., & Verma, O. P. (2017). An effective collaborative movie recommender system with cuckoo search. Egyptian Informatics Journal, 18(2), 105–112. doi:10.1016/j.eij.2016.10.002
[2]. Inan, E., Tekbacak, F., & Ozturk, C. (2018). Moreopt: A goal programming based movie recommender system. Journal of Computational Science, 28, 43–50. doi:10.1016/j.jocs.2018.08.004
[3]. Zhang, S., Jin, Z., & Zhang, J. (2016). The dynamical modeling and simulation analysis of the recommendation on the user–movie network. Physica A: Statistical Mechanics and Its Applications, 463, 310–319. doi:10.1016/j.physa.2016.07.049

[4]. Alhijawi, B., & Kilani, Y. (2020). A collaborative filtering recommender system using genetic algorithm. Information Processing & Management, 57(6), 102310. doi:10.1016/j.ipm.2020.102310

[5]. Renjith, S., Sreekumar, A., & Jathavedan, M. (2019). An extensive study on the evolution of context-aware personalized travel recommender systems. Information Processing & Management, 102078. doi:10.1016/j.ipm.2019.102078.

[6]. Pilászy, I., & Tikk, D. (2009). Recommending new movies. Proceedings of the Third ACM Conference on Recommender Systems - RecSys '09. doi:10.1145/1639714.1639731

[7]. Basile, P., Greco, C., Suglia, A., & Semeraro, G. (2019). Bridging the gap between linked open data-based recommender systems and distributed representations. Information Systems, 86, 1–8. doi:10.1016/j.is.2019.07.001

[8]. L. Chen, G. Chen, and F. Wang, "Recommender systems based on user reviews: the state of the art," User Modeling and User-Adapted Interaction, vol. 25, pp. 99-154, 2015.

[9]. Herce-Zelaya, J., Porcel, C., Bernabé-Moreno, J., Tejeda-Lorente, A., & Herrera-Viedma, E. (2020). New technique to alleviate the cold start problem in recommender systems using information from social media and random decision forests. Information Sciences. doi:10.1016/j.ins.2020.05.071

[10]. Ojagh, S., Malek, M. R., Saeedi, S., & Liang, S. (2020). A location-based orientation-aware recommender system using IoT smart devices and Social Networks. Future Generation Computer Systems, 108, 97–118. doi:10.1016/j.future.2020.02.041