การแนะนำสินค้าด้วยวิธีการอิงเนื้อหาและการกรองข้อมูลแบบพึ่งพาผู้ใช้ร่วม

A Content-Based and Collaborative Filtering Approach for Product Recommendation

Pitchayakorn Lake[1] Parin Kittisophontham[2] Saranthon Maungmee[3] Dechanuchit Katanyutaveetip[4] and Phoosis Wongjetjun[5]

Department of Information Technology, Siam University[1, 2, 3, 4]

Department of Computer Engineering, Siam University[5]

E-mail: pitchayakorn@siam.edu[1], prin.kit@siam.edu[2], saranthon.mau@siam.edu[3], dechanuchit@siam.edu[4], phoosis.won@siam.edu[5]

บทคัดย่อ

บริษัทยูไนเต็ดฟูดส์ จำกัด เป็นบริษัทที่มีชื่อเสียงทางธุรกิจในภาคอุตสาหกรรมผลิตอาหารว่างในประเทศไทย โดยมีผลิตภัณฑ์ที่หลากหลาย รวมถึงวาเฟอร์, เซียงไฮอัลมอนด์ช็อกโกแลต, โยโย เจลลี่, โตโรป๊อปคอร์น และผลิตภัณฑ์อื่น ๆ บริษัทวางยุทธศาสตร์เพื่อเพิ่มยอดขายและปรับปรุงกลุ่มผลิตภัณฑ์ ให้เข้ากับความพอใจของผู้บริโภคในปัจจุบัน เพื่อผลกำไรทางธุรกิจและยกระดับความพึงพอใจของลูกค้า วัตถุประสงค์ของการวิจัย คือการพัฒนาผลิตภัณฑ์โดยใช้โมเดลแนะนำสินค้าการกรองแบบอิงเนื้อหา โดยการนำความคล้ายคลึงของเนื้อหาเพื่อกรองข้อมูลเชิงลึกของผู้ใช้ การศึกษานี้เกี่ยวข้องกับการใช้ไลบรารี Non-negative Matrix Factorization (NMF) ในการประยุกต์ใช้ข้อมูล

คำสำคัญ: ระบบแนะนำ, การกรองแบบอิงเนื้อหา, การกรองข้อมูลแบบพึ่งพาผู้ใช้ร่วม, เหมืองข้อมูล

Abstract

United Food Public Co., Ltd. emerges as a prominent entity within the snack manufacturing sector in Thailand, presenting a diverse array of snack products, encompassing wafers, Sanghai, chocolate almonds, Yoyo jelly, Toropopcorn, and other offerings. The company strategically concentrates on augmenting sales and aligning its product portfolio with prevailing consumer preferences, thereby fostering business profitability and elevating customer satisfaction. The objective of this research is the formulation of a product recommendation model founded on content-based filtering methodologies grounded in principles such as content similarity and collaborative filtering, drawing insights from user-generated data. The study entails the application of the Non-negative Matrix Factorization (NMF) library.

Keywords: Recommendation System, Content-Based, Collaborative Filtering, Data Mining

1. Introduction

United Food Public Co., Ltd., a subsidiary company entrusted with product sales through cash vans or point-of-sale methods on behalf of United Food, is engaged in a customer outreach process. Within the organizational framework, a multitude of employees and customer data are in play. A collaborative effort with the IT department brought forth a prominent challenge: the imperative to boost sales for business proprietors, offer existing products tailored to customer preferences and facilitate customers' purchasing decisions. In this context, the implementation of a product recommendation system emerges as a viable solution, capable of enhancing revenue generation for business owners and aiding consumers in the selection and procurement of products aligning with their preferences and interests.

Consequently, the researchers have undertaken the development of a Recommendation System aimed at suggesting to users, thereby streamlining product selection and acquisition, and ultimately, elevating the company's sales performance. Recommendation systems are instrumental in propelling sales and enjoy widespread application across websites and applications. To address the aforementioned challenge, the researchers have seamlessly integrated a recommendation system with the company's existing customer and product data.

2. Related Literature

2.1 Recommendation System

A recommendation system is a system designed to suggest products or services to users, which helps users in choosing and purchasing various products and services. The system learns from the user's past behavior to provide product recommendations tailored to the user's preferences. [1]

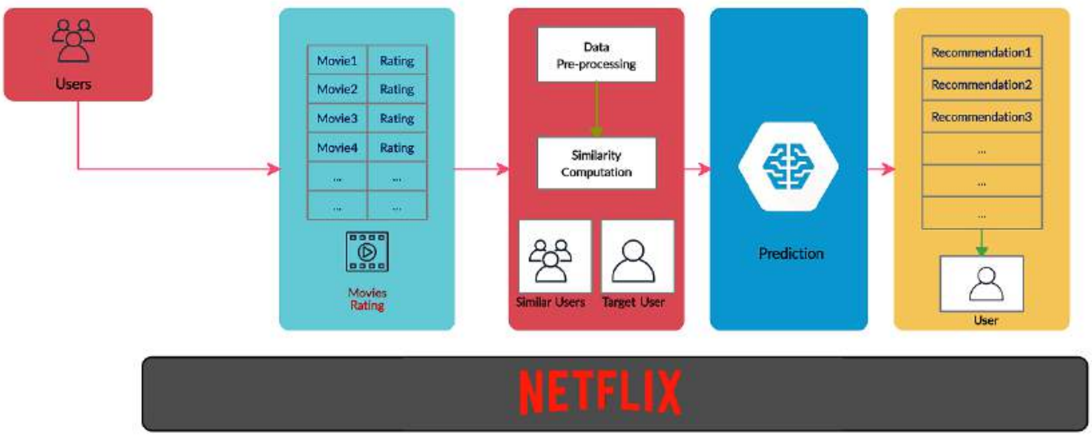

Figure 1 Netflix's recommendation system [2].

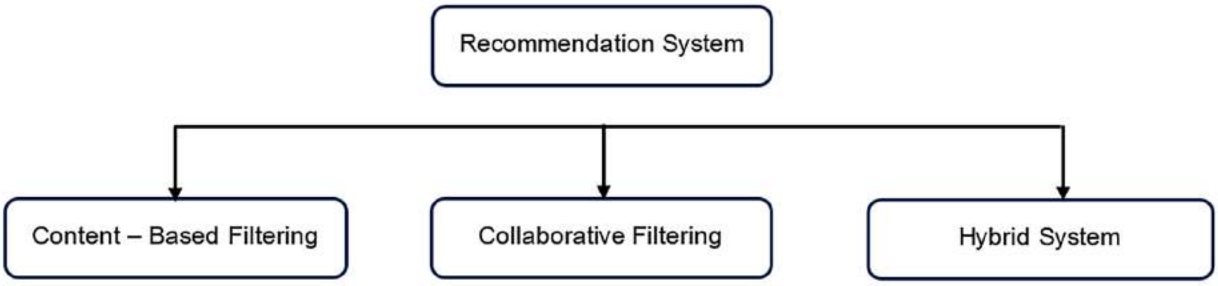

Figure 2 The methodology for a product recommendation system [3].

2.2 Content-Based Filtering

The technique of content-based filtering is a recommendation system that is derived from the consideration of the similarity in user behavior history. It assesses the characteristics of products to recommend and suggests products with features similar to those previously used or favored by the user. The system recommends content for movies that bear resemblance to the ones the user has previously watched, based on the illustrations below. One of the advantages of content-based filtering is that it does not require a large amount of data processing to make recommendations. It can also recommend new products without data because it considers similarities in product characteristics from the same type. On the downside, content-based filtering cannot recommend products that are substantially different from the ones users have purchased or used, which limits the diversity of recommendations for users, as shown in Figure 3 [3].

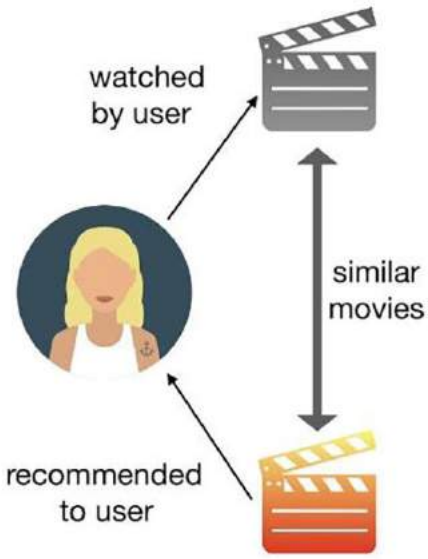

Figure 3 An example of content-based filtering technique [2].

Movies	Reviews Given	Rating
Mission Imposible	✓	Good
James Bond	✓	Good
Toy Story	✓	Bad

Figure 4 An example of movie recommendations for Netflix users [4].

Figure 4, is an example of movie recommendations for Netflix users using the profile name 'Nikhil.' In the case where Nikhil rates 'Mission Impossible' and 'James Bond,' both categorized as action movies, highly and rates 'Toy Story,' a children's movie, low, we can create a user vector for Nikhil based on the top 3 rated genres, as shown in Figure 5, using a rating scale from -10 to 10. Since Nikhil enjoys action movies, he assigns a value of 9 to the action genre. Nikhil has not watched any animation movies, so a value of 0 is assigned to the animation genre. Given that Nikhil has given poor ratings to children's movies, a value of -6 is assigned to the children's genre.

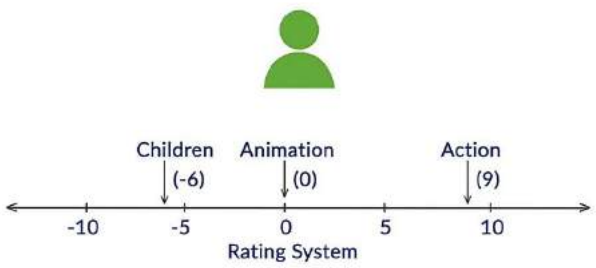

Figure 5 Creating a user vector [4].

2.3 Collaborative Filtering

The collaborative filtering technique involves recommending products or services by considering similarities with other users and incorporating their data into the recommendation. This approach utilizes 'The Wisdom of the Clouds,' which relies on having user ratings for products or services in the database to make predictions. User ratings data can be used for two main approaches in making recommendations: 1. User-Based and 2. Item-Based. User-Based: This method groups users who have similar preferences into the same category. For example, User 1 and User 3 prefer products 3 and 4, when User 1 adds product 1 to cart, the recommendation system will suggest the same product to User 3 based on their similar preferences. Item-Based: This approach, focuses on the similarity between items (products) rather than users. If products 1 and 3 share certain similarities, when User 1 decides to purchase product 1, the system will recommend product 3 to them since they are related, as shown in Figure 6 and Figure 7 [5].

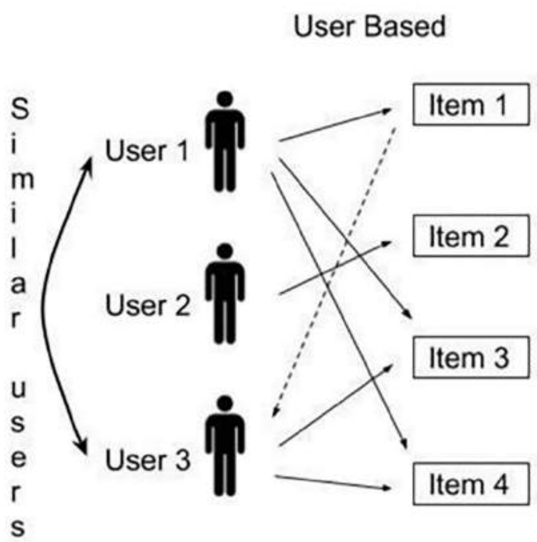

Figure 6 User-Based recommendation [5].

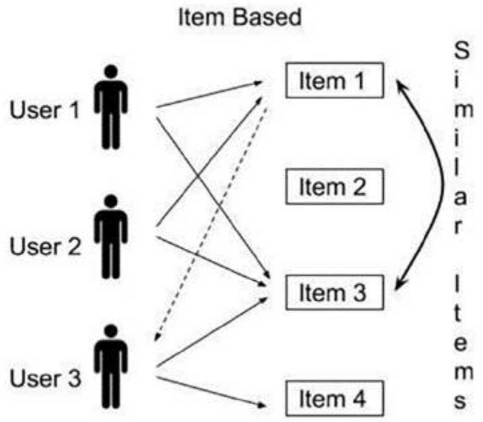

Figure 7 Item-Based recommendation [5].

2.4 Matrix Factorization

A simple embedding model. Given the feedback matrix A ∈ R m × n, where is the number of users (or queries) and is the number of items, the model learns: A user embedding matrix U ∈ R m × d, where row i is the embedding for user I [4][6].

User 1 has given item 2 a rating of 3. Matrix factorization creates two smaller matrices, one for each user and object, which when multiplied together yield this approximate rating matrix, omitting the elements having a value of 0, as shown in Figure 8

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	0
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	0

Figure 8 A matrix of user/item ratings [6].

The number of users is represented by "m" in the matrix m x n, and the number of items by "n." The factors we need are an m x d matrix and a d x n matrix, where "d" is chosen to be large enough to represent the number of dimensions along which interactions between users and items are likely to vary significantly and small enough for the computation to be efficient. In the image above, the value of "d" is set to 2. This means that the dot product of the vectors representing the user and the item is the expected rating that a certain user will provide for a particular item, as shown in Figure 9.

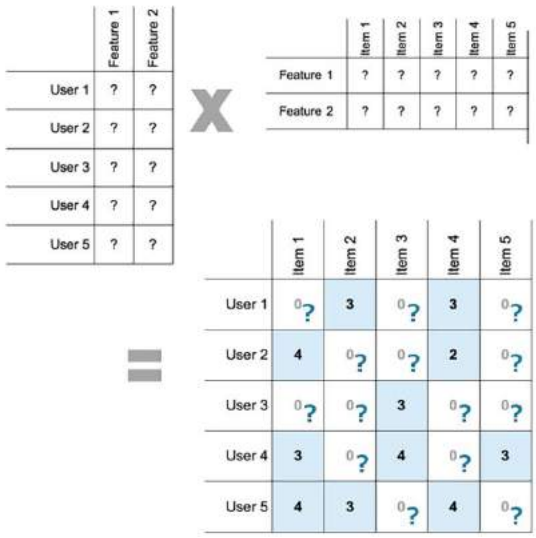

Figure 9 The matrix factorization separated into two parts [6].

Hameed et al. proposed an overview of collaborative filtering recommendation systems. It outlines the different aspects of collaborative filtering, including the categorization of recommendation systems, computation of similarity, and the evaluation of system performance. They discuss various measures used for evaluating collaborative filtering algorithms, such as predictive accuracy metrics, classification accuracy metrics, precision and recall, and ROC curves. user evaluation of recommendation systems [7]. Z. Shahbazi and Y.-C. Byun focus on the development and evaluation of a content-based filtering recommendation system using the XGBoost machine learning algorithm to suggest items to users based on their click information and user profile. The study specifically uses the Jeju online shopping mall dataset to recommend items based on user click information. They discuss the impact of technology on recommendation systems, emphasizing the growth of online shopping and the benefits for both users and sellers using content-based filtering and collaborative filtering [8]. Fu et al. proposed a two-stage process for recommendation systems, involving the learning of low-dimensional embeddings for users and items, followed by the generation of predicted ratings using multiview feed-forward neural networks. The research included extensive experiments on benchmark datasets (MovieLens 1M and MovieLens 10M). Results demonstrated that their proposed approach consistently outperformed or achieved performance close to state-of-the-art when compared to existing collaborative filtering (CF)-based methods. The research involved detailed analyses and visualizations of learned representations, exploring different types of embeddings, transformation layers, and views in the recommendation system. Overall, the findings suggested that the deep learning-based CF framework. presented has the potential to significantly enhance recommendation systems [9]. B. Abdollahi and O. Nasraoui explore the ethical and transparency implications of machine learning models, focusing on the design of explainable intelligent systems in recommender systems. They propose an explainability-constrained Matrix Factorization (MF) technique that generates accurate and explainable recommendations [10].

3. Methodology

3.1 Data Analysis

The researchers have studied the Python language and learned how to use various functions in Google Colab to increase sales volume and provide product recommendations to both new and existing customers based on data relationships. They examined the dataset, which consists of sales data imported from a database, consists of sales data imported from a database and converted it into a CSV file for modeling in Google Colab, as shown in Figure 10 and Figure 11.

```
[ ]  1 datarecom = pd.read_csv('/data_recom.csv') #นำเข้าไฟล์ csv
     2 datarecom.head() #แสดงข้อมูล
```

	CustomerID	ID	TotalOrderQty
0	103110301682	1	1
1	206101102034	1	2
2	1031101120477	1	1
3	206104602037	1	4
4	206104201277	1	3

```
1 itemrecom = pd.read_csv('/data_item.csv') #นำเข้าไฟล์ csv
2 itemrecom.head()
```

	ProductID	ID	ProductRefCode	ProductName
0	30000048	1	30000048	ลูกอมรสบ๊วย 100 เม็ด 1x24x100
1	30000815	2	30000815	ลูกอม 123 ซ่าส์ รวมรส 1x24x100
2	30001003	3	30002926	โยโย โคลา 20g. 1x12x12
3	30001021	4	30001021	แก๊ป ชอกโกแลต 40g 1x6x12
4	30001080	5	30001080	แก๊ป สตรอเบอร์รี่ 40g 1x6x12

Figure 10 Customer data and product listings are filtered based on user preferences.

```
1 df.head()
```

	CustomerID	ID	TotalOrderQty
0	01111111111	1	1
1	01111111111	2	1
2	01111111111	3	2
3	01111111111	6	3
4	01111111111	7	4

Figure 11 Customer data filtered by content-based filtering.

3.2 Collaborative Filtering Technique

For collaborative filtering data techniques, researchers use Pandas and Numpy libraries to assist in creating a product recommendation model, which utilizes the NMF algorithm, considering similarity based on cosine similarity values, as shown in Figure 12 and Figure 13 [11].

```
[ ]  1 from google.colab import drive #นำเข้าไฟล์จาก drive
     2 drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[ ]  1 cmat = pd.crosstab(datarecom['CustomerID'],datarecom['ID'],datarecom['TotalOrderQty'],aggfunc=sum) #แปลงข้อมูล

     1 cmat
```

ID CustomerID	1	2	3	6	7	8	9	14	15	16	...	288	290	291	292	293	295	296	297	299	304
103100905001	2.0	NaN	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
103100905004	NaN	NaN	1.0	NaN	NaN	NaN	NaN	2.0	1.0	1.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
103100905008	NaN	NaN	5.0	3.0	NaN	NaN	NaN	NaN	2.0	1.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
103100905009	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
103100905018	NaN	NaN	1.0	NaN	NaN	NaN	NaN	1.0	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2061412050007	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2061412050008	NaN	NaN	2.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2061412050009	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.0	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2061412130001	NaN	NaN	3.0	4.0	NaN	NaN	NaN	6.0	6.0	8.0	...	NaN	NaN	3.0	NaN	3.0	NaN	NaN	3.0	3.0	NaN
2061414010001	NaN	NaN	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

15539 rows × 224 columns

Figure 12 Preparing the data for the NMF algorithm

.

```
[ ]  1 from sklearn.decomposition import NMF #นำเข้า libraries NMF
     2 nmf = NMF(50) #แบ่ง 50 กลุ่มที่คล้ายคลึงกัน
     3 nmf.fit(cmat) #คำนวณจาก cmat ตัวแปรที่ประกาศ

NMF
NMF(n_components=50)

     1 cmat
```

ID CustomerID	1	2	3	6	7	8	9	14	15	16	...	288	290	291	292	293	295	296	297	299	304
103100905001	2.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
103100905004	0.0	0.0	1.0	0.0	0.0	0.0	0.0	2.0	1.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
103100905008	0.0	0.0	5.0	3.0	0.0	0.0	0.0	0.0	2.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
103100905009	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
103100905018	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2061412050007	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2061412050008	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2061412050009	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2061412130001	0.0	0.0	3.0	4.0	0.0	0.0	0.0	6.0	6.0	8.0	...	0.0	0.0	3.0	0.0	3.0	0.0	0.0	3.0	3.0	0.0
2061414010001	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

15539 rows × 224 columns

Figure 13 Dividing customers into 50 groups with similar purchasing patterns.

```
1 H = pd.DataFrame(np.round(nmf.components_,2), columns=cmat.columns) #คำนวณหาค่าเมทริก H
2 H
```

0	0.03	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.00	...	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.01	0.00	0.00
1	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.15	0.00	0.00	0.00	0.00	0.00	0.02	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	224.47	49.56	0.00	0.00	0.00	0.78	0.00	22.18	...	1.97	0.00	0.00	0.16	0.00	0.74	0.00	0.00	7.23	1.09
4	0.05	0.04	0.00	0.14	0.00	0.00	0.00	0.03	0.00	0.08	...	0.00	0.00	0.01	0.00	0.08	75.45	9.08	0.07	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.15	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.01	0.07	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.38	0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.65	0.00	0.03	0.00	0.00	0.35	0.00	0.00
8	0.04	1.44	0.00	0.00	1.55	0.00	0.00	0.00	0.00	0.00	...	0.05	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.23	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
10	0.00	0.00	5.19	1.18	0.00	0.00	0.00	0.72	0.00	0.00	...	0.00	0.00	0.70	0.00	0.00	0.00	0.00	0.00	0.93	0.47
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01	0.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.21	0.13	0.00	0.00	0.13	0.00	0.00	0.00	0.38	0.00	...	0.00	0.00	0.99	0.00	0.88	0.00	0.24	0.16	0.00	0.00
13	0.59	0.00	1.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.10	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.19	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	6.71	0.00	0.00	0.00	0.00	0.00	8.75	0.00	...	0.00	0.00	2.72	0.00	2.83	0.24	0.00	8.11	0.00	0.00
17	0.00	3.58	0.00	0.00	0.00	0.00	0.00	0.00	1.53	0.00	...	0.66	0.00	0.00	0.02	0.00	0.00	0.00	0.00	1.31	1.63
18	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.23	0.00	0.09
20	0.09	0.00	0.00	0.27	0.00	0.00	0.00	0.00	0.39	0.00	...	0.00	0.00	0.00	0.00	0.28	0.00	0.86	0.00	0.00	0.00
21	0.39	1.42	0.67	0.00	1.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.68	0.00	0.00	0.00	0.00	0.14	0.00	0.00

Figure 14 Calculate the matrix 1.

```
1 W = pd.DataFrame(np.round(nmf.transform(cmat),2), columns=H.index) #คำนวณหาค่าเมทริก W
2 W
```

	0	1	2	3	4	5	6	7	8	9	...	40	41	42	43	44	45	46	47	48	49
0	0.00	0.0	0.00	0.01	0.0	0.00	0.0	0.0	0.0	0.0	...	0.0	0.02	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
1	0.19	0.0	0.01	0.00	0.0	0.00	0.0	0.0	0.0	0.0	...	0.0	0.00	0.01	0.00	0.0	0.06	0.00	0.0	0.01	0.04
2	0.00	0.0	0.00	0.01	0.0	0.00	0.0	0.0	0.0	0.0	...	0.0	0.00	0.02	0.05	0.0	0.00	0.00	0.0	0.01	0.06
3	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	0.0	0.0	...	0.0	0.00	0.00	0.03	0.0	0.00	0.00	0.0	0.00	0.00
4	0.04	0.0	0.06	0.00	0.0	0.00	0.0	0.0	0.0	0.0	...	0.0	0.00	0.01	0.02	0.0	0.02	0.00	0.0	0.01	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15534	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	0.0	0.0	...	0.0	0.00	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
15535	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	0.0	0.0	...	0.0	0.00	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
15536	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	0.0	0.0	...	0.0	0.00	0.00	0.00	0.0	0.00	0.00	0.0	0.00	0.00
15537	0.00	0.0	0.03	0.01	0.0	0.00	0.0	0.0	0.0	0.0	...	0.0	0.09	0.07	0.08	0.0	0.06	0.01	0.0	0.11	0.24
15538	0.00	0.0	0.00	0.00	0.0	0.02	0.0	0.0	0.0	0.0	...	0.0	0.00	0.01	0.00	0.0	0.00	0.00	0.0	0.00	0.00

15539 rows × 50 columns

Figure 15 Calculate the matrix 2.

Following in Figure 14 and Figure 15, the procedural methodology encompasses the computation of Matrix 1 and Matrix 2, with the explicit objective of discerning analogous products in the context of providing recommendations to customers.

4. Results

The system will select customer code 2061414010001 as a random sample for testing, recommending 10 items. This includes products the customer has previously purchased and products the customer has not purchased before, as shown in Figure 16 and Figure 17.

```
reconstructed = pd.DataFrame(np.round(np.dot(W,H),2), columns=cmat.columns) #คำนวณหาค่าเมทริก W และ H นำมาคูณกัน
reconstructed.index = cmat.index #เติมค่าลงใน  cmat
reconstructed
```

ID	1	2	3	6	7	8	9	14	15	16	...	288	290	291	292	293	295	296	297	299	304
CustomerID																					
103100905001	0.61	0.40	3.11	0.56	0.43	0.00	0.00	0.09	0.08	0.22	...	0.02	0.00	0.01	0.00	0.04	0.01	0.00	0.01	0.09	0.02
103100905004	0.01	0.09	0.24	0.00	0.03	0.00	0.00	3.08	1.25	0.56	...	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.02	0.01	0.01
103100905008	1.21	0.65	2.90	0.70	0.63	0.00	0.00	0.02	2.00	1.12	...	0.08	0.00	1.83	0.00	1.34	0.03	0.00	0.81	1.00	0.16
103100905009	0.00	0.04	0.11	0.03	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.04	0.00	0.01	0.00	0.00	0.01	0.06	0.01
103100905018	0.30	0.20	0.12	0.01	0.21	0.00	0.00	0.00	0.04	0.00	...	0.00	0.00	0.03	0.00	0.01	0.00	0.00	0.00	0.03	0.02
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2061412050007	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2061412050008	0.00	0.00	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2061412050009	0.00	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2061412130001	0.64	0.48	3.34	1.78	0.27	0.54	0.34	6.67	7.84	3.86	...	0.39	0.01	10.77	0.01	8.03	0.12	0.24	4.42	4.81	0.79
2061414010001	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

15539 rows × 224 columns

Figure 16 Results from Calculate the matrix 1 and matrix 2.

```
import re

def recomendation(uid,topk=5): #หา Top 5 index
  res = list(reconstructed.T[uid].sort_values(ascending=False)[0:topk].index)
  res = itemrecom[itemrecom['ID'].isin(res)]
  mList = datarecom[datarecom['CustomerID']==uid][['ID','TotalOrderQty']] #หาข้อมูลการซื้อสินค้าของลูกค้าตามสินค้าที่หาได้
  # #return mList, res
  res = res.merge(mList,on='ID',how='left') #จะมีสินค้าที่เขาเคยซื้อกับไม่เคยซื้อ
  # #return res
  res = res[['ID','ProductID','ProductName','TotalOrderQty']]
  return res
```

```
res = recomendation(2061414010001,topk=10)
res
```

	ID	ProductID	ProductName	TotalOrderQty
0	24	30001781	โยโย องุ่น 20g 1x12x12 Pro	1.0
1	25	30001782	โยโย รวมรส 20g 1x12x12 Pro	NaN
2	26	30001825	เซียงไฮ้ เคลือบช็อกโกแลต 6g 1x36x15	1.0
3	27	30001827	เซียงไฮ้ เคลือบวานิลา 6g 1x36x15	1.0
4	28	30001829	เซียงไฮ้ เคลือบสตรอเบอร์รี่ 6g 1x36x15	1.0
5	30	30001831	เซียงไฮ้ เคลือบนม 6g 1x36x15	1.0
6	45	30001975	ไทม ช็อกโกแลต 30g 1x12x12	1.0
7	47	30001978	ไทม นม 30 g. 1x12x12	NaN
8	48	30001979	ไทม สตรอเบอร์รี่ 30g 1x12x12	NaN
9	50	30002455	เซียงไฮ้ ซุปเปอร์จัมโบ้ รสช็อกโกแลต 34 g. 1x12x12	1.0

Figure 17 Results from collaborative filtering techniques.

For content-based filtering techniques, researchers use the Pandas and Numpy libraries to assist in creating a product recommendation model for new customers, which utilizes the Cosine Math algorithm. Researchers simulate customer data and compare it with the purchase history of other customers, considering cosine similarity values, as shown in Figure 18.

```
[ ]
data = {'CustomerID': ['01111111111'],'ID': 1, 'TotalOrderQty': 1}# ข้อมูลในแถวที่จะเพิ่ม
df = pd.DataFrame(data)  # สร้าง DataFrame
print(df)
# เพิ่มแถวใน DataFrame
new_row = {'CustomerID': '01111111111','ID': 2, 'TotalOrderQty': 1}
df = df.append(new_row, ignore_index=True)
new_row = {'CustomerID': '01111111111','ID': 3, 'TotalOrderQty': 2}
df = df.append(new_row, ignore_index=True)
new_row = {'CustomerID': '01111111111','ID': 6, 'TotalOrderQty': 3}
df = df.append(new_row, ignore_index=True)
new_row = {'CustomerID': '01111111111','ID': 7, 'TotalOrderQty': 4}
df = df.append(new_row, ignore_index=True)
print(df)
```

```
     CustomerID  ID  TotalOrderQty
0  01111111111   1              1
     CustomerID  ID  TotalOrderQty
0  01111111111   1              1
1  01111111111   2              1
2  01111111111   3              2
3  01111111111   6              3
4  01111111111   7              4
<ipython-input-118-51036aa85777>:6: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(new_row, ignore_index=True)
<ipython-input-118-51036aa85777>:8: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(new_row, ignore_index=True)
<ipython-input-118-51036aa85777>:10: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(new_row, ignore_index=True)
<ipython-input-118-51036aa85777>:12: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  df = df.append(new_row, ignore_index=True)
```

```
[ ]
df.head()
```

	CustomerID	ID	TotalOrderQty
0	01111111111	1	1
1	01111111111	2	1
2	01111111111	3	2
3	01111111111	6	3
4	01111111111	7	4

Figure 18 Add customer data for testing.

Researchers will select customer code 01111111111 for testing and recommend 5 items. The recommendations will include products that the customer has previously purchased and products they have not purchased. To determine similarity, researchers will compare their purchase history with other customers. In this specific instance, at position 6417, the customer code is 206130506002, so researchers will recommend products that customer code 01111111111 has not purchased but are similar to those purchased by customer code 206130506002, as shown in Figure 19.

Create Application

```
from scipy.spatial.distance import cosine #ดึง libraries cosinemath

similarity = []
for idx in range(len(reconstructed)): #วนลูปตัวดาต้าที่สร้างขึ้นมาสำหรับลูกค้าใหม่ ออกมาเทียบกับข้อมูลลูกค้าทีละคน
    similarity.append(cosine(test.values.reshape(224),reconstructed.iloc[idx].values))
similarity = pd.Series(similarity).fillna(0).tolist()
close_to = np.argsort(similarity)[-1]
close_to #เทียบแล้วเลือกคนที่ 6417 ใน index
```

```
/usr/local/lib/python3.10/dist-packages/scipy/spatial/distance.py:622: RuntimeWarning: invalid value encountered in double_scalars
  dist = 1.0 - uv / np.sqrt(uu * vv)
6417
```

```
test.values.reshape(224)
```

```
(224,)
```

```
reconstructed.iloc[idx].values.shape
```

```
(224,)
```

```
recomendation(reconstructed.index[close_to],5) # เทียบตำแหน่งใน index ท็อป 5
```

	ID	ProductID	ProductName	TotalOrderQty
0	46	30001976	ไทม ส้ม 30 g. 1x12x12	2.0
1	47	30001978	ไทม นม 30 g. 1x12x12	NaN
2	48	30001979	ไทม สตรอเบอร์รี่ 30g 1x12x12	NaN
3	50	30002455	เซียงไฮ ซุปเปอร์จัมโบ้ รสช็อกโกแลต 34 g. 1x12x12	2.0
4	63	30002519	ไทมมะพร้าว 30 g 1x12x12	NaN

```
reconstructed.index[close_to] #ตำแหน่ง 6417 คือรหัสลูกค้า 206130506002
```

```
206130506002
```

Figure 19 The results obtained from content-based filtering techniques.

5. Conclusion and Recommendation

The company has undertaken the development of a product recommendation system, which can provide product recommendations to both new and existing customers with the overarching goal of enhancing sales performance and operational efficiency. For example, sales representatives can utilize the system to recommend products to new customers based on their historical order data, specifically suggesting items that have not been previously purchased.

The drawback of this method lies in the absence of content for specific items, as content-based systems primarily function as document classifiers and frequently do not extend their applicability to various other product categories, such as movies or restaurants. Additionally, there exists the potential for an unwarranted degree of restrictiveness, as these systems are fundamentally tailored to suggest items akin to those previously evaluated by the user. Consequently, users may be prone to overlooking items of potential interest that fall beyond the purview of documents they have previously assessed.

Furthermore, for repeat customers with established purchase histories, the system employs a comparative analysis approach, aligning their purchasing patterns with those of similar customers to propose products they have not yet acquired. Additionally, the limitation intrinsic to this study is discerned in the acknowledgment that a substantial volume of data contributes to the refinement of precision in the recommendation process implemented by the system.

References

[1] C. C. Aggarwal, Recommender Systems: The Textbook, 1st ed. 2016 edition. New York, NY: Springer, 2016.

[2] M. S. Reddy, P. T. R. Kumar, L. M. Siddarth, and R. Mothukuri, “Designing Recommendation System for Hotels Using Cosine Similarity Function,” in Soft Computing for Security Applications, G. Ranganathan, Y. EL Allioui, and S. Piramuthu, Eds., in Advances in Intelligent Systems and Computing. Singapore: Springer Nature, 2023, pp. 1–15. doi: 10.1007/978-981-99-3608-3_1.

[3] U. Javed, K. Shaukat Dar, I. Hameed, F. Iqbal, T. Mahboob Alam, and S. Luo, “A Review of Content-Based and Context-Based Recommendation Systems,” Int. J. Emerg. Technol. Learn. IJET, vol. 16, Feb. 2021, doi: 10.3991/ijet.v16i03.18851.

[4] B. Balu, “Content-Based Recommender System,” Medium. Accessed: Nov. 15, 2023. [Online]. Available: https://medium.com /@bindhubalu/content-based-recommender-system-4db1b3de03e7

[5] L. Jiang, Y. Cheng, L. Yang, J. Li, H. Yan, and X. Wang, “A trust-based collaborative filtering algorithm for E-commerce recommendation system,” J. Ambient Intell. Humaniz. Comput., vol. 10, no. 8, pp. 3023–3034, Aug. 2019, doi: 10.1007/s12652-018-0928-7.

[6] N. Gillis, Nonnegative Matrix Factorization. Philadelphia: SIAM - Society for Industrial and Applied Mathematics, 2020.

[7] M. A. Hameed, O. Al Jadaan, and S. Ramachandram, “Collaborative filtering based recommendation system: A survey,” Int. J. Computer Sci. Eng., vol. 4, no. 5, p. 859, 2012.

[8] Z. Shahbazi and Y.-C. Byun, “Product recommendation based on content-based filtering using XGBoost classifier,” Int J Adv Sci Technology, vol. 29, pp. 6979–6988, 2019.

[9] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, “A novel deep learning-based collaborative filtering model for recommendation system,” IEEE Trans. Cybern., vol. 49, no. 3, pp. 1084–1096, 2018.

[10] B. Abdollahi and O. Nasraoui, Using Explainability for Constrained Matrix Factorization. 2017, p. 83. doi: 10.1145/3109859.3109913.

[11] P. K. Halder, S. Kundu, and S. Majumder, Online Book Recommendation System Using Collaborative Filtering: Application of Machine Learning using Python. LAP LAMBERT Academic Publishing, 2022.