Expert Systems With Applications 212 (2023) 118565

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A hybrid recommender system for an online store using a fuzzy expert system

Bogdan Walek [1], Petr Fajmon

Department of Informatics and Computers, University of Ostrava, 30. dubna 22, 701 03 Ostrava, Czech Republic

A R T I C L E I N F O

Keywords:
Recommender system
Hybrid recommender system
E-shopping recommender system
Expert system
Collaborative-filtering
Content-based filtering
Products

A B S T R A C T

Nowadays, various recommender systems are popular and their main aim is to recommend suitable content to the user based on various parameters. This article proposes a hybrid recommender system, Eshop recommender, which combines a recommender module composed of three subsystems (the subsystems use collaborative-filtering and content-based approaches) and a fuzzy expert system. It is an e-shopping recommender system for suggesting suitable products. The system works with different user preferences and their activity on the e-shop, and the resulting list of recommended products is created using the fuzzy expert system. The expert system works with several parameters - similarity level with already rated products, coefficient of purchased product, and an average rating of the product. Due to this, our proposed system achieves promising results based on standard metrics (Precision, Recall, F1-measure). The system achieves results above 90%. The system also achieves better results than traditional approaches. The main contribution is creating a comprehensive hybrid system in the area of product recommendation in an online store, which has been validated on a group of real users and compared with other traditional approaches and the recommendation module of another online store.

1. Introduction

Nowadays, various recommender systems are on the rise and their main goal is to recommend suitable content to the user based on various parameters.

A recommender system is an information system used to support user decision making and recommend suitable products, information or services in the environment of online stores, streaming services, online dating sites and many other industries (Falk, 2019).

In online stores, the main goal is to recommend relevant products that are interesting to the customer and have a high probability of being purchased by the customer. Representatives of the largest online stores that implement recommendation systems are Amazon.com, eBay.com or Taobao.com. Products are most often recommended to users based on the popularity of the product, customer demographics, and an analysis of the customer's previous shopping behaviour at the online store. (Schafer, Konstan, & Riedl, 1999) To evaluate the popularity of a product, the most common form of rating is using a popularity scale or star rating. For example, in the online store Amazon.com, customer feedback on a product is obtained using a rating scale from 1 to 5. These product ratings can be used to generate recommendations in the recommendation system of the online store. (Lu, Wu, Mao, Wang, & Zhang, 2015).

The main objective of this article is to design a hybrid recommender system for recommending suitable products in an online store. The proposed recommender system combines a recommender module composed of a collaborative-filtering system, a content-based system, and a fuzzy expert system. The proposed system works with the user's favourite product categories, viewed products, and purchase history. A comprehensive list of recommended products for the user is built using the fuzzy expert system to evaluate the importance of the products. The architecture and methodology of the proposed system are described in Section 3.

2. Related work and current state in recommender systems

Recommender systems have been in development for many years and applied in various problem domains: tourism (Logesh, Subramaniyaswamy, & Vijayakumar, 2018; Ravi & Vairavasundaram, 2016; Stanley, Lorenzi, Saldaña, & Licthnow, 2003), advertising (Cheung, Kwok, Law, & Tsui, 2003), e-commerce (Ghani & Fano, 2002), music (Rodríguez-García, Colombo-Mendoza, Valencia-García, Lopez-

E-mail addresses: bogdan.walek@osu.cz (B. Walek), P21163@student.osu.cz (P. Fajmon).
[1] ORCID ID: 0000-0003-1119-0420.

https://doi.org/10.1016/j.eswa.2022.118565
Received 8 October 2021; Received in revised form 8 August 2022; Accepted 12 August 2022
Available online 23 August 2022

Lorca, & Beydoun, 2015), and others. This article focuses on recommender systems for designing appropriate products in online stores, so we will now discuss recommender systems in this area.

2.1. E-commerce/e-shopping recommender systems

Recently, there has been an increasing number of online stores offering goods to their online customers. The global pandemic COVID-19 has significantly reinforced this trend, and a large number of new online shops have been established in the last year. Online shops are an essential area of the e-commerce sector.

In online stores, the most common element to determine the popularity of a product is its rating. The rating is often used in the form of stars or similar visual methods, and the user gives feedback to the internet on the specific product purchased. Typically, a scale of 1 to 5 is used. Product ratings can then be used to generate recommendations and are a standard input to recommendation systems.

Some of the largest online stores use recommender systems to recommend suitable products relevant for customers to purchase (Huang, Chung, & Chen, 2004; Schafer, Konstan, & Riedl, 2001). In these online stores, products can be recommended based on the product's overall popularity, the type of products that the customer is browsing, or by analysing previous purchases to suggest relevant products for future purchases. Different types and variations of recommender systems are proposed in research articles. These recommender systems are also validated for different types of online stores.

Wasabi Personal Shopper (WPS) is an example. It is a domain-independent tool for browsing electronic product catalogues (Burke, 1999). However, this system includes rather a basic way of recommending products since it is a general tool for browsing product catalogues.

Cao and Li (Cao & Li, 2007) developed a fuzzy recommender system for products made of different components. For example, when purchasing a laptop, customers often intuitively compare different laptops based on the performance of each component, such as CPU, motherboard, RAM, etc. The proposed recommender system recommends the best product candidates based on the weights of each component determined by the customer using a fuzzy similarity measure model. The advantage of this system is recommendation based on weights of individual components and the use of a fuzzy recommender system. However, a limitation, compared with our system and other systems, is the need to work with products containing different parameters (components). The system then recommends suitable items based on the combination of these components.

Mooney and Roy (Mooney & Roy, 2000) proposed a content-based book recommender system using information extraction and a machine learning algorithm for text categorisation. A naive Bayesian text classifier is used to train data extracted from the web to generate features of books and user profiles and find the best matching books for the target user. This system effectively works with content-based recommendation. A disadvantage is the absence of work with other users (readers of similar or other books) and the use of the collaborative-filtering approach.

Users often want to be informed about what suitable product is available for them or why the recommender system suggested that particular product. To provide relevant explanations for recommendations of why a given product is better than another, McCarthy, Reilly, McGinty, and Smyth (2004) developed a web-based shopping assistant called Qwikshop.com, on which compound critiques were used as explanations. Compound critiques are product feedback from users. This user feedback is used to generate a set of behavioural patterns, and then based on this feedback, recommendations of relevant products are made to the user. The advangate of this system is the use of explanations of why a given product has been recommended. A disadvantage is the absence of other standard approaches, e.g. CB, CBF or hybrid approach.

Another area is the sale of bundles of items or bundle promotions. Online stores often offer these multi-product bundles and promotional packages because they bring savings to the merchants and are convenient for customers. The product retrieval method was extended to recommend suitable product bundles and promotion packages in the system developed by Garfinkel, Gopal, Tripathi, and Yin (2006). With the advent of mobile phones and their use for online shopping, mobile-based recommender systems have also developed. Lawrence et al. (Lawrence, Almasi, Kotlyar, Viveros, & Duri, 2001) proposed a mobile recommender system for recommending new products to customers shopping in supermarkets and using PDA devices. The advantage was to order purchase and send it to the store where it was then ready for pick up by the customer. The association mining method was used to analyse the relationships between groups of products. The clustering method was then used to identify groups of customers with similar purchase histories. The generated groups of popular products were then input into the matching process of customers and products to generate recommendations. A disadvantage of this system is the need of a larger number of active customers so that the recommender system runs correctly.

Another area is the promotion of products on social media due to its great growth in recent years. The authors Li et al. (Li, Chou, & Lin, 2014) designed and developed a group-coupon recommender system for suggesting suitable location-sensitive products. The system was validated on Facebook on 20 products of different categories (catering, leisure and cosmetic/maintenance), and 726 users were involved in the validation. The results showed that the proposed system is able to accurately recommend products and also significantly increases the willingness to purchase the recommended products due to the rating of the product by other friends on Facebook. This system is interesting and beneficial despite its limitation of being generally used for the social network Facebook. The social network Facebook contains a recommender system that shows a user products rated by other friends on Facebook. With no connection to Facebook, the system would be less effective.

Cold-start problem and sparsity problem (Aggarwal, 2016; Falk, 2019) are common problems when using the collaborative-filtering method in recommender systems. Authors Huang et al. (Huang, Chen, & Chen, 2016) reduced the impact of these problems by creating and novel recommendation model using Google similarity. The Google similarity method is used by Google search to determine the correlation between two search phrases. The authors proposed a hybrid approach that combines standard item-based collaborative-filtering with Google Similarity-based collaborative-filtering. This made it possible to refine the item similarity calculation. It also increased the accuracy of the final prediction of recommended products. A limitation of this approach is its connection to the Goolge algorithm, which is a third-party system. In case of a modification or complete cancellation of Google algorithm, the recommender system would work differently or be unusable.

A recommender system working with product-seeded and basket-seeded scenarios has been proposed for the domain of small online stores (Kaminskas, Bridge, Foping, & Roche, 2017). The recommender system is designed for small-scale retail websites where a smaller group of returning customers makes standard user-centric techniques (e.g., collaborative-filtering) inapplicable. The authors applied an item-centric product recommendation strategy that combines two well-known methods-association rules, and text-based similarity-to generate recommendations based on a single 'seed' product. And then, they also adapted the proposed approach to recommendations based on a set of 'seed' products in a user's shopping basket. A limitation of this system is its focus on smaller e-shops, which prevents a wider use of this system.

When shopping in an online store, users often make decisions based on their basic needs and relative need to buy a product. Authors Tareq et al. (Tareq, Noor, & Bepery, 2020) proposed a model of dynamic recommendation system (DRS) for an online market. The proposed system provides an intelligent solution to overcome the problem of customer ratings and feedback by integrating market basket analysis, frequent item mining, bestselling items and customer personalisation.

The system is interesting despite being limited to dynamic recommendation, which is less effective when working with historical data, e.g. past purchases.

E-shopping recommender systems are therefore suitable for different types of products, whether they are digital products (music, movies, etc.) or physical products (electronics, books, food, etc.). Many different recommender systems have been developed in this area. They have been successfully validated in online stores and can be implemented by online store developers.

Thus, it can be concluded that important goals of e-shopping recommender systems include:

- Accurate recommendation of relevant products - brings increased sales on the e-shop side and satisfaction on the user side
- Explanation of why the product was suggested - increases the credibility of the e-shopping recommender system and provides the user with information about why the product was recommended.
- Connection to social networks - recommender systems that are connected to social networks and display ratings of the products by other "friends" increase the user's willingness to purchase the product

Our proposed system possesses several advantages over the analysed systems:

- Our system uses a hybrid recommender system and combines content-based and collaborative-filtering approaches.
- The use of the hybrid approach reduces the cold-start problem as new users are recommended products based on similarity with already viewed products.
- The system uses standard algorithms of recommender systems, which are fully integrated into the system without the need of any third-party algorithms.
- The proposed system is connected to a fuzzy expert module to ensure the final ranking of products based on various parameters. The module can be modified and extended, which increases its practical functionality and usability in other types of internet stores.

Let us present now several reasons which made us select a hybrid recommender system. The first reason is the fact that hybrid recommender systems combine basic approaches of recommender systems (content-based filtering, collaborative filtering, knowledge-based) and thus increase system performance and provide more accurate content recommendation. The main motivation for the creation of hybrid recommender system is the fact that current basic approaches have lots of weaknesses. Hybrid recommender systems focus on solving such drawbacks and increasing the efficiency of recommending suitable content (Aggarwal, 2016; Falk, 2019). The benefits of hybrid recommender systems and their evaluation is described in detail in a literature review (Véras, Prota, Bispo, Prudêncio, & Ferraz, 2015). Discussions appearing in literature focused on recommender systems also show that hybrid recommender systems are able of effective solutions to problems of traditional recommender systems (Aggarwal, 2016; Falk, 2019). Currently, a series of systems and approaches based on hybrid recommender system is being developed. They are also interconnected with methods of artificial intelligence (Biswas & Liu, 2022; Fan, Wu, Parvin, Beigi, & Pho, 2021; Kiran, Kumar, & Bhasker, 2020; Rebelo, Coelho, Pereira, & Fernandes, 2021; Vahidi Farashah, Etebarian, Azmi, & Ebrahimzadeh Dastjerdi, 2021). Of course, current literature reviews deal with hybrid recommender systems (Lavanya, Khokle, & Maity, 2021; Seth & Sharaff, 2022).

3. Recommender system

In this chapter, our proposed recommender system will be described in more detail. Our system is a hybrid system that combines a content-based approach (CB) and a collaborative-filtering approach (CBF). The system is also interfaced with an expert system for the final ranking of the displayed products.

The novelty of our system primarily lies in the interconnection of a hybrid recommender system with a fuzzy expert system, which can be modified and extended as needed. The expert system works with several parameters and serves for the final ranking of the displayed products. In addition, the input parameters into the expert system can be substituted by other or new parameters can be added, which increases its general usability for various types of e-shopping recommender systems.

Our proposed recommender is fully implemented as a web-based system called Eshop recommender. The architecture of the system is shown in Fig. 1:

The architecture of the described system consists of several main modules:

- User interface
- Recommender module
- Expert system

The method of architecture proposal stems from our experience, see the architecture proposed in our article for a different recommender area (Walek & Fojtik, 2020). The architecture was selected based on our experiments with various algorithms for recommender systems and their combination. A similar system architecture was also selected in other studies, for instance in this system (Carrer-Neto, Hernández-Alcaraz, Valencia-García, & García-Sánchez, 2012).

The Recommender module recommends products in several areas. The division of the recommender module and the principle of interconnection with other parts of the system is shown in Fig. 2.

The recommender module consists of the following submodules:

- Recommendation of relevant products to the viewed products (content-based approach)
- Recommendation of relevant products to the reviewed products (collaborative-filtering approach)
- Recommendation of relevant products to purchase history (content-based approach)

3.1. User interface and system database

The proposed recommender system is linked to the user interface and the database of products and users.

The user interface has been created in the form of a classic online shop, which contains the following functionalities:

- Display of product categories
- Display of product details
- Product ratings
- Adding products to the cart and purchasing them

Thanks to these functionalities, the user of the online store can browse individual products, view a list of products of a given category, rate products or purchase selected products.

The rating of the products is possible using a scale of 1–5 stars, where the value of 1 indicates the worst rating and the value of 5 indicates the best rating of the product.

In the case of products, the problem domain is the online grocery store. These are real food products that were imported from one of the biggest Czech online grocery stores Kosik.cz.

There are several characteristics of an online store which call for the suitability of implementing a hybrid recommender system with recommending suitable content:

- An online store enables user rating of products.

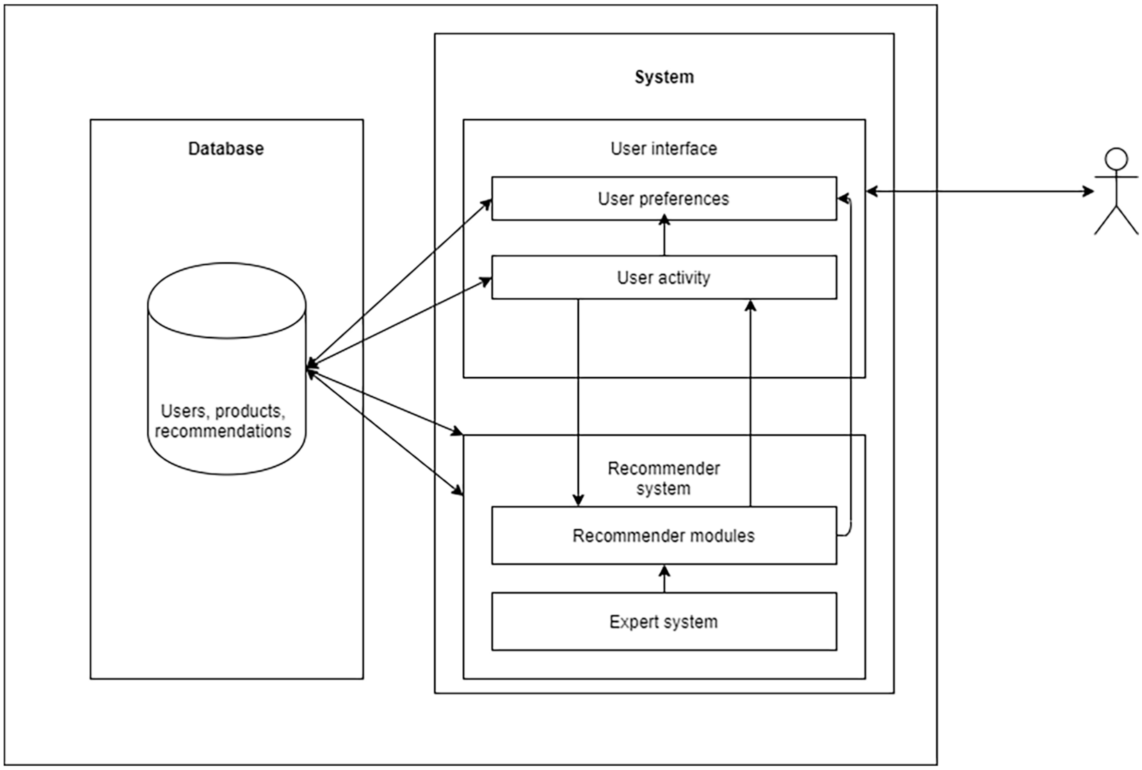

Fig. 1. Architecture of the proposed recommender system.

- An online store has a group of users that have already rated more than one product (rating of more products is necessary for correct functioning of the collaborative filtering approach).
- An online store offers types of goods that motivate the customer to shop regularly (electronics, consumer goods, food) and thus requires suitable content recommendation for this customer.

Our selected type of an online store meets all of these characteristics and thus is suitable for the implementation of a hybrid recommender system.

3.2. Recommender module

The recommender module is a monolithic hybrid recommender system. It consists of 3 subsystems. The first subsystem recommends relevant products relative to the products viewed and uses a content-based approach. The second subsystem recommends relevant products relative to the products reviewed and uses a collaborative-filtering approach. The third subsystem recommends relevant products given a user's purchase history and uses a content-based approach.

3.2.1. Content-based system – Viewed products

The first subsystem of the recommendation module is a content-based system applied to the products being viewed. The input to this subsystem is the products that the user views while browsing the online store. The output is then the most similar products to the viewed products.

Content-based filtering system generally consists of several components: a) preprocessing and feature extraction, b) content-based learning of user profiles, c) filtering and recommendation (Aggarwal, 2016; Falk, 2019). Within the preprocessing and feature extraction component, the most widely used algorithms are TF-IDF and LDA (Falk, 2019). The TF-IDF algorithm was chosen for our recommender system because of its widespread use, simpler implementation, and lower system resource requirements. Despite its simplicity, in most cases the algorithm gives results comparable to those of LDA (assuming the use of *n*-grams), thus its only disadvantage compared to LDA is that the whole training process has to be repeated after adding a new product, whereas with LDA an already created model can be reused repeatedly. Within the content-based learning component of user profiles and the nearest neighbor classification domain, we chose the cosine similarity function because it is one of the most widely used similarity functions. If we were working with structured data, it would be appropriate to use other similarity/distance functions such as Euclidean distance or Manhattan distance (Aggarwal, 2016; Pazzani & Billsus, 2007).

An active user of an online store gradually browses through different products of interest to him. The system stores these browsed products in a database. The working principle of the subsystem consists of the following steps:

1. Retrieving the list of products browsed by the user.
2. Applying the TF-IDF algorithm to each of the browsed products.
3. Ranking the most similar products by similarity.
4. Displaying similar products to the user.

For better illustration, we will demonstrate the next steps of our proposed recommender system and each subsystem on a real user ID 4. This user performed all the necessary actions in the online store linked to the proposed recommender system. He marked his favourite categories, viewed the selected products, rated the different products and made a purchase in the online store.

The user's marked favourite product categories are:

- Pastry
- Cereals
- Beverages
- Spices
- Alcoholic beverages

The list of products viewed by the user during the testing of the

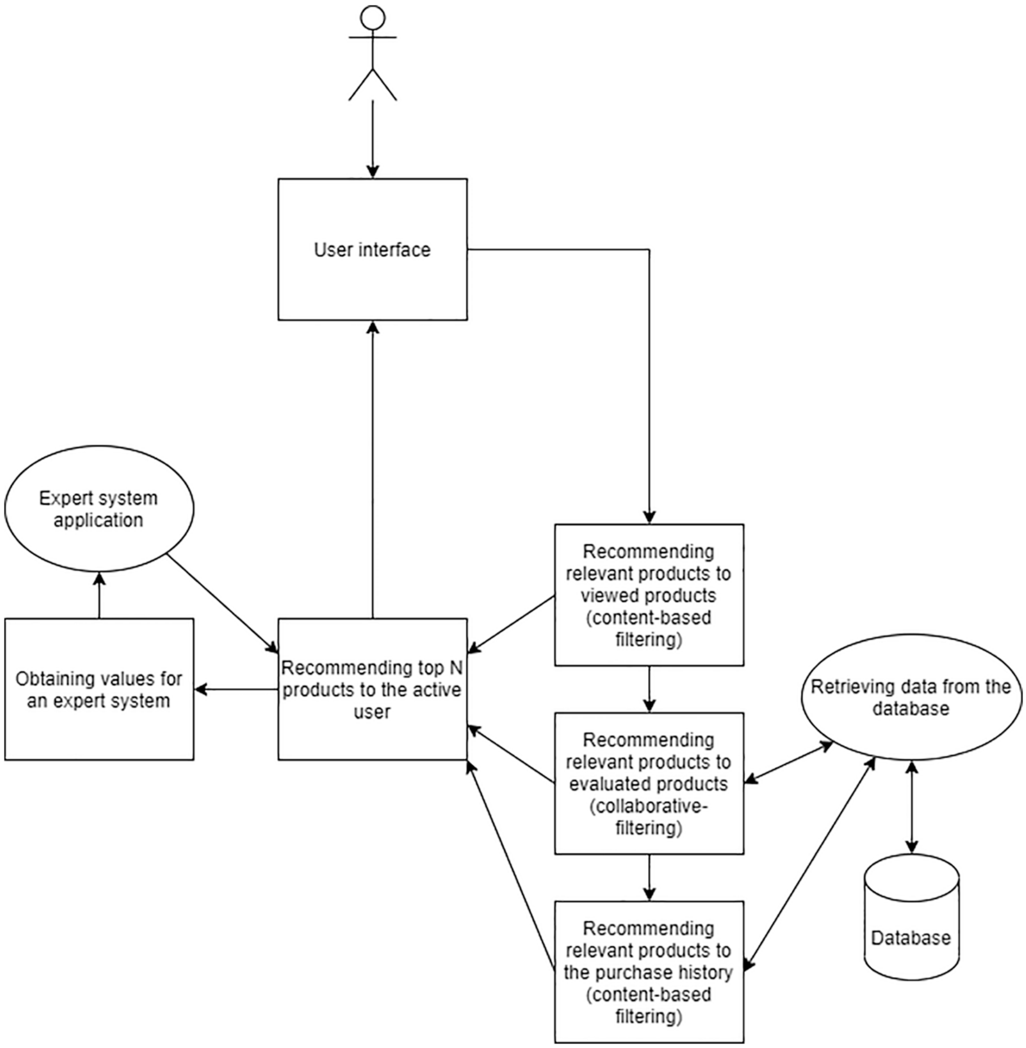

Fig. 2. Recommender module architecture.

recommender system is shown in Table 1.

Next, the calculation of the most similar products to the viewed products (see Table 1) was performed using the TF-IDF algorithm described above. The list of the most similar products calculated by the TF-IDF algorithm is shown in Table 2.

In Table 2, we see the most similar products to the viewed products of user ID 4 (most of these products were also rated by the user). The table includes products in the Fruit category because the user rated several fruits highly. There are also products from other categories that the user has rated, as well as products from the user's favourite categories.

Table 1
List of viewed products by user ID 4.

Product	Rating	Product category
Sprite	4	Beverages
Kofola	4	Beverages
Marjoram	2	Spices
Beetroot	2	Vegetables
Perch	2	Vegetables
Gyros	1	Spices
Plum	5	Fruit
Grapefruit	4	Fruit
7Up	not rated	Beverages
Pepsi cola	not rated	Beverages
Vodka	not rated	Alcoholic beverages
Jack Daniels Honey	not rated	Alcoholic beverages
Jagermeister	not rated	Alcoholic beverages
Orange	4	Fruit
Pineapple	5	Fruit
Rum	not rated	Alcoholic beverages

3.2.2. *Collaborative-filtering system – Rated products*

The second subsystem of the recommender module is a collaborative-filtering system applied to the viewed products. The input to this subsystem is the products that the user views while browsing the online store. The output is then the most similar products to the viewed products.

The collaborative-filtering system is an important part of the system and works with a user-item matrix, which in our case consists of two basic pieces of information:

- User - user rating of the product, a numerical rating of the product (value from the interval $< 0.5,5 >$) by one user.
- Item - rated product

This user-item matrix is then used to calculate the predicted rating, which will be used to rank the most suitable products for the user based on their product rating and product ratings by other users. Currently, there are two basic groups of collaborative-filtering systems (Falk, 2019):

Table 2
List of most similar products to the viewed products of user ID 4.

Product	Product similarity	Product rating	Product category
Kiwifruit	1.000	4.2	Fruit
Pear	0.899	4	Fruit
Grilled chicken	0.893	4	Spices
Fanta	0.876	4.2	Beverages
Oregano	0.810	4	Spices
Kofola	0.803	4.1	Beverages
Curry	0.766	3.8	Spices
Lemon	0.752	3.6	Fruit
Grid with pudding filling	0.713	4	Pastries
Coca-Cola	0.695	3.7	Beverages
Becherovka	0.683	4	Alcoholic beverages
Spaghetti	0.674	3.5	Spices
Energy drink Semtex	0.671	3.3	Beverages
Grapes	0.669	4.2	Fruit
Sunflower bun	0.651	3.9	Pastries
Zucchini	0.635	4.1	Vegetables
American potatoes	0.634	4.1	Spices
Braid	0.623	3.7	Pastries
Doughnut	0.622	3.9	Pastries
Lime	0.608	3.9	Fruits
Cabbage lettuce	0.607	3.8	Vegetables
Salad seasoning	0.604	3.8	Spices
Energy drink tiger	0.601	3.9	Beverages
Cereal bun	0.593	4.2	Pastries
Tangerine	0.589	4	Fruit
Strawberries	0.581	4.2	Fruit
Garlic ground	0.579	4	Spices
Goulash	0.571	3.7	Spices
Vanilla snail	0.555	3.8	Pastries
Breznak	0.550	3.9	Beer

- Neighborhood-based (memory-based) collaborative-filtering
- Model-based collaborative-filtering

In a model-based system, the main goal is to find latent factors in the data. This can be achieved using Matrix Decomposition, for example, using the Single Value Decomposition (SVD) method. We have used this method for collaborative-filtering system because it is one of the most widely used methods in collaborative-filtering approach (Abbasi, Khadivar, & Yazdinejad, 2019; Anwar & Uma, 2021; Pradeep & JayaBhaskar, 2018) and also this algorithm has the smallest average prediction error according to RMSE algorithm as per the calculations we have done in this work (Walek & Fojtik, 2020).

SVD is a method of decomposing the matrix M into individual components in order to simplify further calculations (Falk, 2019). The output of the SVD method is 3 matrices, U, Σ and V^T, where.

- M - a matrix we want to decompose, in our case rating matrix of all rating of products by users
- U - user feature matrix, user is a user which evaluates products
- Σ – weights diagonal matrix, give us information about how much we should reduce the dimensions
- V^T – item feature matrix, in our case item is a product and ^{T} represents the specific rated product

The process of decomposing matrix M is depicted in Fig. 3.

When using the SVD algorithm the Σ will always be a diagonal matrix (Falk, 2019).

The rating itself is then obtained by the scalar product of the U and Σ V^T matrices at a given position.

The principle of operation of the algorithm is shown in Fig. 4.

However, the problem with the SVD algorithm is that it cannot handle a matrix with missing values (it counts these values as 0), so in our system, we will add values to this matrix using the procedure described below.

The collaborative-filtering system first retrieves similar users to the active user from the database. Similar users are those who have similar favourite product categories as the active user. In our system, a similar user is one who has 50 % or more of the same favourite categories as the active user. In the next step, the U (users) field of these users is created along with the active user. Then, a field P (products) is created which contains all the products that the users from field U have rated. The system creates a user-item matrix where the rows represent the users from array U, and the columns represent the products from array P. It then inserts the product rating of that user into each user-product position. If the user has not rated the product, a value of 0 is added to that position. The SVD algorithm then decomposes this matrix into the matrices U, sigma, Vt. Based on these matrices, it calculates the similarity between the products for a given user and creates a group of recommended products.

The working principle of the collaborative-filtering subsystem consists of the following steps:

1. The user has rated several products.
2. The system creates a user-item matrix supplemented with the product ratings of each user. A value of 0 is added to the unfilled data.
3. The system applies the SVD algorithm and calculates the similarity between the active user's products.
4. The products already rated by the active user are removed from the products found.
5. The system delivers similar products to the active user

The group of similar users to user ID 4 contains 13 other users. Table 3 shows the rated products of user ID 4.

Table 4 shows a small part of the user-item matrix, which for the purpose of the SVD algorithm contains 98 rows (number of users) and 7169 columns (number of rated products). To demonstrate the algorithm's operation, we chose a portion of the user-item matrix that contains 10 rows (users) and 10 columns (products). The user-item matrix containing only product ratings by users is shown in Table 4.

For clarity of the table, we list the aliases for users and products:
U1 = user ID 4.
U2 = user ID 61.
U3 = user ID 68.
U4 = user ID 75.
U5 = user ID 83.

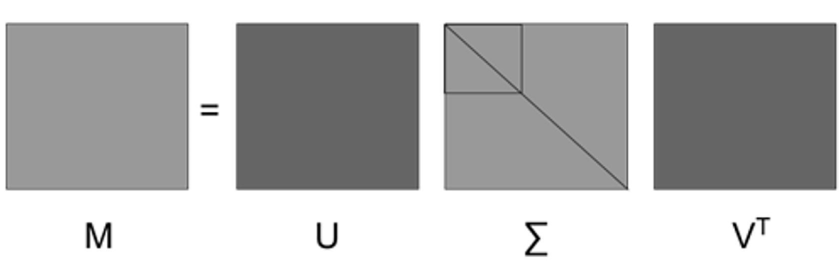

Fig. 3. Process of decomposing matrix M (Source: Falk, 2019).

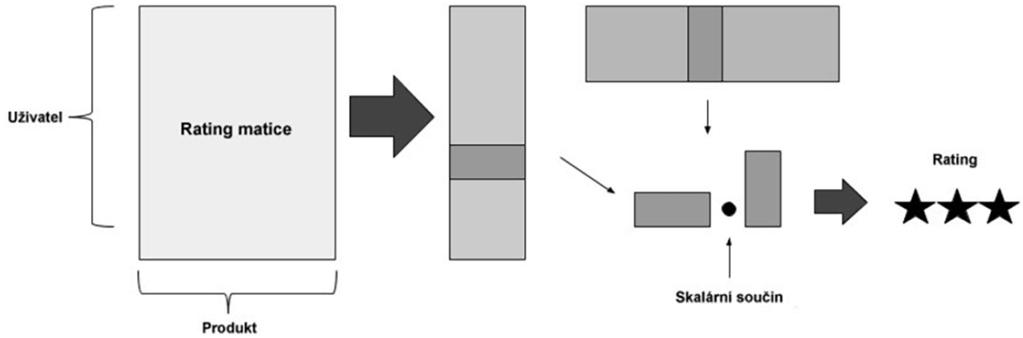

Fig. 4. Principle of the SVD algorithm in a collaborative-filtering system.

Table 3
List of products rated by user ID 4.

Product	Rating	Product category
Pineapple	5	Fruit
Organic Bulgur	4	Durable foods
Plum	5	Fruit
Bozkov Republica	4	Alcoholic beverages
Bulgur	5	Durable foods
Sunflower bulgur	4	Bakery products
Caleo Primitivo Salento	5	Alcoholic beverages
Celery	5	Vegetables
Filet with skin	2	Fish
Grapefruit	4	Fruit
Gyros	1	Spices
Halibut fillet	2	Fish
Strawberries	4	Fruits
Cereal bun	5	Pastries
Curry	2	Spices
Kofola	4	Beverages
Cheesecake	5	Pastries
Grilled chicken	1	Spices
Marjoram	2	Spices
Cake with custard filling	3	Pastries
Cucumber	3	Vegetables
White pepper	3	Vegetables
Pavolin palava	5	Alcoholic beverages
Pellegrino Marsala	5	Alcoholic beverages
Orange	4	Fruit
Peach fruit puree	1	Baby food
Apple fruit puree	2	Baby food
Quinoa red	4	Durable foods
Tomato	4	Vegetables
Sprite	4	Beverages
Cod fillet	1	Fish
China	1	Spices
Beetroot	2	Vegetables
Vanilla snail	4	Pastries
Spaghetti	2	Spices

Table 4
Part of the user-item matrix containing only user ratings of products.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
U1	5	5	4	1	4	2	2	4	4	2
U2	4				4				4	
U3	2		4		5					
U4										
U5	5		1							
U6	5			2						
U7										
U8	5								1	
U9	5		4							
U10										

U6 = user ID 88.
U7 = user ID 89.
U8 = user ID 99.
U9 = user ID 104.
U10 = user ID 107.
P1 = pineapple.
P2 = plum.
P3 = grapefruit.
P4 = gyros.
P5 = kofola.
P6 = marjoram.
P7 = cucumber.
P8 = orage.
P9 = sprite.
P10 = beetroot.

Looking at Table 4, it is clear that the user-item matrix contains only a few ratings for each product. Products P2, *P6*, P7, P8 and P10 were then rated by only one user in this section of the user-item matrix. Since most of the ratings are missing in the user-item matrix, the missing elements of the matrix need to be automatically filled in for the SVD algorithm to work properly. Since we do not know the specific value of these elements, we add the value 0 to the missing elements of the matrix. The user-item matrix completed and the zero values of the missing ratings is shown in Table 5.

A matrix containing the predicted ratings for the selected part of the user-item matrix is shown in Table 6.

3.2.3. Content-based system – Purchased products

The last subsystem of the recommendation module is a content-based system applied to purchased products. The input of this subsystem is the products that the user has purchased from the online store. The output is then the most similar products to the purchased products.

The basic assumption is that similar products to the purchased products will be relevant to the user who purchases certain products. A modified TF-IDF algorithm will be used to calculate similar products.

An active user makes a purchase of several products at a given time.

Table 5
Part of the user-item matrix containing the user ratings of the products plus zero values.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
U1	5	5	4	1	4	2	2	4	4	2
U2	4	0	0	0	4	0	0	0	4	0
U3	2	0	4	0	5	0	0	0	0	0
U4	0	0	0	0	0	0	0	0	0	0
U5	5	0	1	0	0	0	0	0	0	0
U6	5	0	0	2	0	0	0	0	0	0
U7	0	0	0	0	0	0	0	0	0	0
U8	5	0	0	0	0	0	0	0	1	0
U9	5	0	4	0	0	0	0	0	0	0
U10	0	0	0	0	0	0	0	0	0	0

Table 6
Part of the user-item matrix containing predicted ratings based on the SVD algorithm calculation.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
U1	0.166	1.000	0.305	0.000	0.170	0.283	0.283	0.761	0.392	0.283
U2	0.265	0.110	0.201	0.000	0.521	0.204	0.147	0.324	1.000	0.137
U3	0.042	0.255	0.503	0.0186	1.000	0.015	0.000	0.016	0.477	0.006
U4	0.012	0.183	0.007	0.009	0.000	1.000	0.197	0.444	0.008	0.386
U5	1.000	0.907	0.186	0.655	0.000	0.453	0.248	0.481	0.558	0.125
U6	0.515	0.053	0.186	1.000	0.000	0.014	0.014	0.040	0.188	0.014
U7	0.126	0.168	0.007	0.009	0.000	0.112	1.000	0.136	0.008	0.640
U8	1.000	0.907	0.429	0.600	0.059	0.605	0.275	0.579	0.000	0.218
U9	0.391	0.253	1.000	0.000	0.492	0.091	0.083	0.170	0.231	0.097
U10	0.014	0.019	0.008	0.010	0.000	0.131	0.542	0.919	0.010	1.000

The system records whether the user has already evaluated the purchased products or not. Based on the rating of these products, each of the purchased products is assigned a coefficient that will affect the calculation of the TF-IDF algorithm. If the user has not rated the product, a coefficient of 0.5 is automatically assigned. The other products are given a coefficient according to their user rating:

- Product rating 5 - coefficient 1
- Product rating 4 - coefficient 0.8
- Product rating 3 - coefficient 0.6
- Product rating 2 - coefficient 0.4
- Product rating 1 - coefficient 0.2
- Unrated product - coefficient 0.5

```
Input:
P – list of purchased products of an active user
U – active user
H – product rating (Pi)
Output:
T – list of top products listed according to their similarity
foreach (Pi)
{
  if (U rated Pi)
  {
    if H == 5 then product coefficient = 1;
    elseif H == 4 then product coefficient = 0.8;
    elseif H == 3 then product coefficient = 0.6;
  elseif H == 2 then product coefficient = 0.4;
  elseif H == 1 then product coefficient = 0.2;
  }
  else
  {
    product coefficient = 0.5
  }
}
Calculation of the top products list = application of the TF-IDF algorithm on products
  with top products list ordering coefficients according to the similarity
```

Algorithm 1: Principle of the content-based system of purchased products.

Visually, the architecture of the content-based system of purchased products is also shown in Fig. 5:

Another part of this approach is the calculation of the product purchase time coefficient. The purpose of this coefficient is to favour products that have been purchased in the recent past. An example would be the regular purchase of similar foodstuffs. If a user buys groceries every day, then it is appropriate to suggest similar products to him based on all that he has purchased in the past, but to discount products that he has purchased recently. For example, if a user bought rolls and milk today, then tomorrow or in a few days, he is likely to want the same or similar products again. In our proposed system, products purchased on the current day get the highest coefficient, while products purchased 3 months ago get the lowest coefficient.

The value of the coefficient of the product purchase time:

- Products purchased today - coefficient 1
- Products purchased within 7 days - coefficient 0.7
- Products purchased within 30 days - coefficient 0.5
- Products purchased within 90 days - coefficient 0.2
- Products purchased more than 90 days - do not affect the final calculation

Another part is the product category coefficient. If the recommended products are from the same category as products the user has already purchased in the past, then they get an extra 0.5 coefficient. If the user has purchased several products from the Bakery category, then the recommended products from the Bakery category get an extra coefficient of 0.5.

To illustrate the whole approach, here is a list of products purchased by user ID4. The list is shown in Table 7.

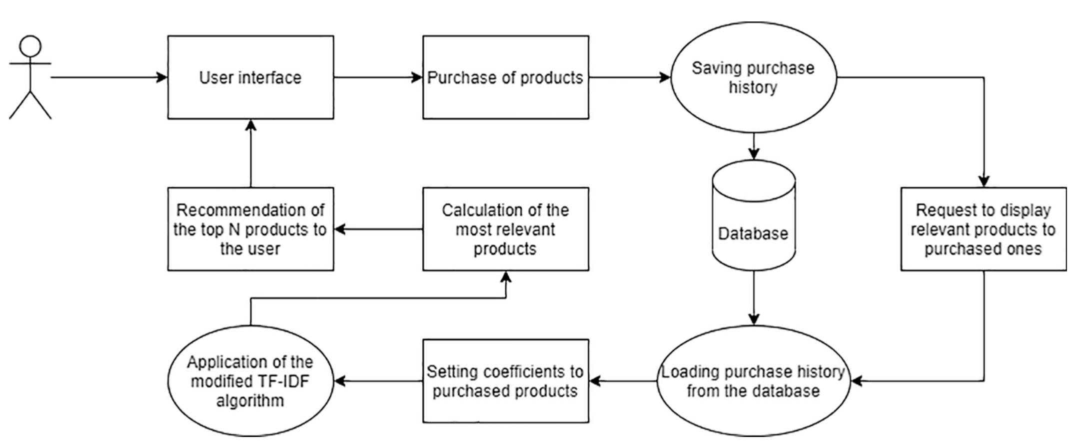

Fig. 5. Architecture of the content-based system of purchased products.

Table 7
List of purchased products by user ID 4.

Product	Purchasing time coefficient	Product category
Jack Daniels Honey	1	Alcoholic beverages
Mackerel	1	Fish
Carp portions	1	Fish
Tangerine	1	Fruit
Watermelon	1	Fruit
Becherovka	1	Alcoholic beverages
Plum	1	Fruit
Pork tenderloin	0.7	Meat
Chicken pieces	0.7	Meat
Chicken	0.7	Meat
Beef burger	0.7	Meat
Milk fresh	0.7	Dairy products
Cottage cheese	0.7	Dairy products
Surface-ripened cheese	0.7	Dairy products
Curry	0.5	Spices
American potatoes	0.5	Spices
Chicken breast	0.5	Meat
Surface-ripened cheese	0.5	Dairy products
Red peppers	0.5	Vegetables
Pavolán Pálava	0.5	Alcoholic beverages
Cheese chunks	0.5	Dairy products

It is clear from the table that the first seven products were purchased on the day in question, the next seven were purchased within 7 days before the appointed day, and the next seven were purchased within 30 days before the appointed day.

Table 8 lists the top similar products to the purchased products for user ID 4.

Similar products to the purchased products are sorted according to the calculated product similarity. The next column shows the coefficient of the purchase time. The fourth column shows the discount for products that are in categories where the user has made a purchase in the past. The product coefficient is then the sum of the purchase time coefficient and this bonus.

3.3. Expert system

Another module of the proposed recommender system is the expert system, which is used for the final ranking of the recommended products. The aim is to evaluate the importance of the recommended products and rank them appropriately by the users. The ranking of products using the expert system will be done based on the relevant information that can be retrieved for each recommended product. This information includes:

- The degree of similarity to the rated products - the output of the collaborative-filtering system
- Coefficient of the purchased product - the coefficient is obtained from the purchase history of the product
- Average rating of the product

We chose a fuzzy expert system due to its ability to model vague concepts using fuzzy sets. Also important is the ability to easily modify the definition of linguistic variables as well as the knowledge base composed of IF-THEN rules. The IF-THEN rules of an expert system are written in a manner similar to the linguistic formulation of rules in everyday life. We use the software tool Linguistic Fuzzy Logic Controller (Habiballa, Novák, Dvořák, & Pavliska, 2003) to create and modify the fuzzy expert system.

Based on this information, the following input linguistic variables of the knowledge base of the expert system were created:

- INP1 - degree of similarity to already evaluated products, possible values from the interval $< 0.1 >$.
- INP2 - product coefficient from purchase history, the coefficient determines the importance of the product in the recommended list of products according to the user's purchase history, possible values from the interval $< 0, 1.5>$
- INP3 - average product rating, possible values from the interval $< 0, 5>$

The output language variable is:

- IMPORTANCE - indicates the level of importance of the product for the final display to the user, possible values are from the interval $< 0,1>$

The output variable of the expert system is a standard language variable with the use of modifiers. The values used have a range of extremely low to extremely high. The basic language values are: low, medium and high. These basic language values are then supplemented with the modifiers extremely, significantly, very, more or less, roughly, quite roughly, very roughly, rather, very very roughly. These modifiers make it possible to model individual situations more accurately and to

Table 8
List of the most similar products to the purchased products of user ID 4.

Product	Similarity	Purchasing time coefficient	Bonus for products from product categories related to the purchase	Product coefficient	Product category
Pear	1	1	0.5	1.5	Fruit
Chicken lower leg	0.917	0.7	0.5	1.2	Meat
Yellow peppers	0.820	0.7	0.5	1.2	Vegetables
Kiwifruit	0.741	1	0.5	1.5	Fruit
White pepper	0.710	0.5	0.5	1	Vegetables
Whole milk long-life	0.706	0.7	0.5	1.2	Dairy products
Beef slices	0.704	0.7	0.5	1.2	Meat
Beef chunks	0.699	0.7	0.5	1.2	Meat
Minced beef	0.665	0.7	0.5	1.2	Meat
Grapefruit	0.658	1	0.5	1.5	Fruit
Chinese seasoning	0.629	1	0.5	1.5	Spices
Black pepper	0.626	0.5	0.5	1	Spices
Spaghetti	0.605	1	0.5	1.5	Spices
Grilled chicken	0.602	1	0.5	1.5	Spices
Straw wine	0.595	1	0.5	1.5	Alcoholic beverages
Strawberries	0.588	1	0.5	1.5	Fruits
Beef cubes	0.583	0.7	0.5	1.2	Meat
Minced pork	0.579	1	0.5	1.5	Meat
Sweet pepper	0.564	1	0.5	1.5	Vegetables
Sheep cheese	0.561	0.7	0.5	1.2	Dairy products

determine the value of the output variable. The output variable indicates the degree of importance of the product for the final recommendation to the user.

Fig. 6 shows the membership functions for the output language variable IMPORTANCE. The language expression more or less small is marked in red, other expressions are extremely low, significantly low, very low, more or less low, roughly low, quite roughly low a very roughly low, rather low, very very roughly low, medium, more or less medium, roughly medium, quite roughly medium, very roughly medium, rather medium, typically medium, very very roughly medium, very very roughly high, rather, very roughly, quite roughly, roughly, more or less high, high, very high, significantly high, extremely high.

The expert system was developed in the Linguistic Fuzzy Logic Controller (LFLC) tool (Habiballa et al., 2003). The LFLC tool allows defining input and output variables and the shape of their fuzzy sets. Furthermore, the tool allows to create a knowledge base structure of the expert system and populate this knowledge base with IF-THEN rules. The tool also includes the possibility to choose an inference mechanism and a defuzzification method to calculate the final value of the output language variable. Examples of IF-THEN rules are shown below:

`IF (INP1 is very low) and (INP2 is low) and (INP3 is low) THEN (IMPORTANCE is extremely low).`

`IF (INP1 is low) and (INP2 is high) and (INP3 is low) THEN (IMPORTANCE is low).`

`IF (INP1 is low) and (INP2 is medium) and (INP3 is medium) THEN (IMPORTANCE is quite roughly low).`

`IF (INP1 is medium) and (INP2 is very high) and (INP3 is medium) THEN (IMPORTANCE is quite roughly high).`

`IF (INP1 is high) and (INP2 is med) and (INP3 is high) THEN (IMPORTANCE is very roughly high).`

`IF (INP1 is very high) and (INP2 is very high) and (INP3 is high) THEN (IMPORTANCE is extremely high).`

Table 9 lists a few selected IF-THEN rules for illustration. Each column contains the language values of the input language variables and the output language variables.

The complete knowledge base of the expert system contains a total of 60 IF-THEN rules. A list of all rules is given in the Appendix (Supplementary Material).

The inference and defuzzification process then computes the resulting crisp number, which represents the resulting numerical value of EXS IMPORTANCE - see Table 10. As part of the testing and debugging of the expert system, we tested various inference and defuzzification methods.

The conjuctive normal form (CNF) or fuzzy approximation with conjunctions was chosen as the inference method for the proposed expert system. In testing different inference methods with defuzzification methods, this inference method had the tiniest differences in the resulting values. The MCOG (modified centre of gravity) or modified centre of gravity method was chosen as the diffusive method. The COG and MCOG methods were considered in the testing, but the chosen method better matched the expected results.

Several test outputs of the expert system are shown in Table 10. Different combinations were tested for a given product. The test results for product 1 and 5 came out the same, with product 1 having average similarity and average rating values and the product coefficient being 1.3, high. Product 5 has low similarity to the other products but a high average rating and a high product coefficient, similar to product 1. On the other hand, products 6 and 7 got a very low value of EXS IMPORTANCE variable, although both products have average similarity, the product coefficient is very low and the average product rating is low to low.

Table 9
Selected IF-THEN rules of the expert system.

Rule	INP1	INP2	INP3	EXS IMPORTANCE
1	Very low	Low	Low	Extremely low
7	Low	High	Low	Low
26	Low	Medium	Medium	Quite roughly low
32	Medium	Very high	Medium	Quite roughly high
48	High	Medium	High	Very roughly high
60	Very high	Very high	High	Extremely high

Table 10
Expert system testing.

Product	INP1	INP2	INP3	EXS IMPORTANCE
1	0.5	1.3	3.1	0.75
2	0.7	0.2	4	0.51
3	0	1.5	4.2	0.5
4	0.8	1	4.5	0.9
5	0.2	1.2	4.7	0.75
6	0.5	0	3	0.25
7	0.5	0.2	1.6	0.23

The average resulting value of EXS IMPORTANCE is possessed by the tested products 2 and 3, both of which have a relatively high average rating, product 2 has a high similarity to other products, but a low coefficient and exactly the opposite is true for product 3, where its coefficient is the highest, but its similarity to other products is zero. The best value of the variable EXS IMPORTANCE is the tested product 4, which has a high similarity to other products, a high average rating and a higher coefficient.

The list of recommended products for user ID 4 is therefore supplemented with the appropriate input variables and also the EXS IMPORTANCE value. The list is then sorted by EXS IMPORTANCE. This list is displayed in Table 11:

Based on the experimental results, we found that if the final ranking of the recommended products was done only on the basis of EXS IMPORTANCE, then the value of product similarity would be suppressed in some cases. We also do not consider whether or not the recommended product belongs to one of the user's favourite product categories. For this reason, we have calculated the so-called final rating based on EXS IMPORTANCE and product similarity and possible popularity of the product category according to the following formula:

FinalRating = EXS IMPORTANCE * (1 + Product Similarity) + FavouriteCategoryCoefficient.

The recommendation value is obtained as the product of the resulting variable rank of each product and the similarity obtained by the recommender system, to which is added the value 1. The products are ranked according to the recommended value and the top N most suitable products are delivered to the user.

The final list of recommended products for user ID 4 - is shown in Table 12:

As shown in Table 12, by calculating the final rating, the recommender system improved the rating of some products and their importance (position) in the final list of recommended products for users.

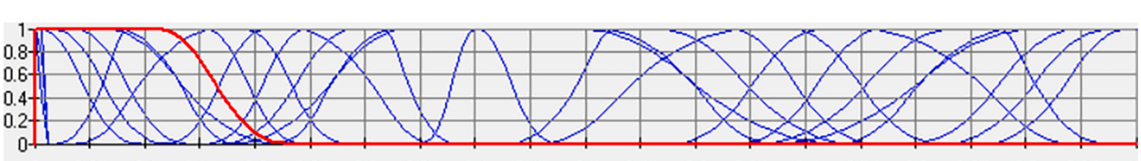

Fig. 6. Membership functions for output linguistic variable IMPORTANCE.

Table 11
List of recommended products with the EXS IMPORTANCE using an expert system.

Product	Similarity	Product coefficient	Product rating	EXS IMPORTANCE
Grilled chicken	0.885	1	4	0.967
Kiwifruit	0.895	0.7	4.2	0.938
Fanta	0.708	0.2	4.2	0.927
Oregano	0.435	0	4	0.873
Pear	0.649	1	4	0.867
Bun	0.577	1.5	3.7	0.863
Cake with custard filling	0.391	0	4	0.785
Barley groats	0.395	0	4.1	0.763
Jack Daniels Honey	0.408	0	4.2	0.502
Bun dark	0.452	0	3.8	0.500
Moet & Chandon Imperial	0.434	0	4.1	0.500
Kofola	0.420	0	4.1	0.500
Poppyseed cake	0.417	0	3.9	0.500
Curry	0.411	0	3.8	0.500
Organic eggs	0.392	0	4.1	0.500
Coca-Cola	0.389	0.2	3.7	0.500
Cheese snail	1.000	1.5	4	0.500
Doughnut	0.55	1.5	3.9	0.500
Becherovka	0.387	0	4	0.500
Lemon	0.520	0.7	3.6	0.500
Energy drink Semtex	0.509	0	3.3	0.500
Spaghetti	0.482	1	3.5	0.500
Organic wheat	0.473	0	3.5	0.500
Lime	0.459	0.7	3.9	0.500
Quinoa	0.455	0	4	0.500

Our proposed system was fully implemented as a web-based recommender system with an online store. Fig. 7 shows part of the main page of the recommendation system.

4. Results

This part describes the verification of our proposed system on a group of real users. To evaluate the quality of the recommendations made in our system, the following standard metrics were used: precision, recall and F1-measure.

First, a general definition of the Precision and Recall metrics will be given:

- Precision is the ratio of RL to N.
- Recall is the ratio of RL to R,

where N denotes the size of the recommendation list L, RL denotes the number of relevant items that are included in L, and R denotes the total number of relevant items (Aggarwal, 2016; Falk, 2019; Symeonidis, Nanopoulos, & Manolopoulos, 2009).

Precision determines the ability of a given system to recommend content that is truly relevant to the user. It is the ratio of relevant recommendations with respect to all recommendations to the user. Precision can be calculated using the formula:

$$\text{Precision} = \text{Correctly recommended content}/\text{Total recommended content},$$

where Correctly recommended content is the number of relevant recommendations marked by the user as “correctly recommended” and Total recommended content is the number of total recommendations offered to the user.

Recall determines the ability of the system to offer relevant content to the user. It is the number of correct recommendations in the set of relevant recommendations, i.e. the TOP recommendations that the system suggests to the user. Recall can be calculated using the following formula:

$$\text{Recall} = \text{Correctly recommended content}/\text{Relevant content},$$

where Correctly recommended content is the number of recommendations marked as “correctly recommended” and Relevant content is the set of top recommendations based on user recommendations.

F1-Measure is then the harmonic mean between precision and recall: F1-measure = 2 * precision * recall/precision + recall.

Since the determination of correct (relevant) or incorrect (non-relevant) items in the list of recommended items is highly subjective, it is not possible to automate this process. In this case, we need to test the proposed system on a group of real users.

A group of 17 users participated in the testing and tested our proposed system by marking relevant and irrelevant products. They are

Table 12
Final list of recommended products.

Product	Similarity	Product coefficient	Product rating	Favourite Category Coefficient	EXS IMPORTANCE	FinalRating
Cheese snail	1.000	1.5	4	0.2	0.500	2.135
Kiwifruit	0.895	0.7	4.2	0	0.938	1.878
Grilled chicken	0.885	1	4	0.2	0.967	1.842
Fanta	0.708	0.2	4.2	0.2	0.927	1.819
Pear	0.649	1	4	0	0.867	1.762
Bun	0.577	1.5	3.7	0.2	0.863	1.639
Doughnut	0.55	1.5	3.9	0.2	0.500	1.638
Lemon	0.520	0.7	3.6	0	0.500	1.531
Energy drink Semtex	0.509	0	3.3	0.2	0.500	1.513
Spaghetti	0.482	1	3.5	0.2	0.500	1.478
Organic wheat	0.473	0	3.5	0.2	0.500	1.467
Lime	0.459	0.7	3.9	0	0.500	1.357
Quinoa	0.455	0	4	0.2	0.500	1.132
Bun dark	0.452	0	3.8	0.2	0.500	1.123
Oregano	0.435	0	4	0.2	0.873	1.105
Moet & Chandon Imperial	0.434	0	4.1	0.2	0.500	1.103
Kofola	0.420	0	4.1	0.2	0.500	1.102
Poppyseed cake	0.417	0	3.9	0.2	0.500	1.093
Curry	0.411	0	3.8	0.2	0.500	1.089
Jack Daniels Honey	0.408	0	4.2	0.2	0.502	1.077
Barley groats	0.395	0	4.1	0.2	0.763	1.069
Organic buckwheat	0.392	0	4.1	0.2	0.500	1.063
Cake with custard filling	0.391	0	4	0.2	0.785	1.057
Coca-Cola	0.389	0.2	3.7	0.2	0.500	1.053
Becherovka	0.387	0	4	0.2	0.500	1.042

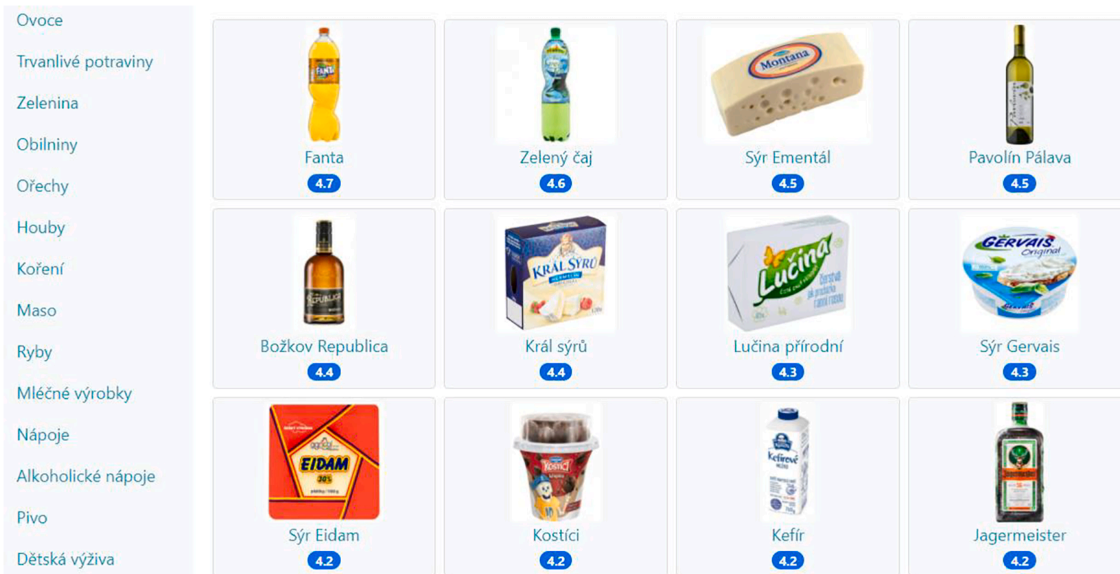

Fig. 7. Main page of the recommender system Eshop recommender.

people aged between 18 and 40 years old. Testing on a group of real users is one of the options that has also been used in other researches (Barragáns-Martínez, Costa-Montenegro, & Juncal-Martínez, 2015; Carrer-Neto et al., 2012; Colombo-Mendoza, Valencia-García, Rodríguez-González, Alor-Hernández, & Samper-Zapater, 2015; Ho, Menezes, & Tagmouti, 2006; Kumar, Yadav, Singh, & Gupta, 2015; Li & Yamada, 2004).

The online store database includes:

- 260 products
- 16 product categories
- 98 users
- 7169 product ratings by users

Users who tested our approach initially made settings for their user preferences by entering the following information:

- Selecting 5 favourite product categories
- Viewing 40 products
- Rating 40 products (the same or other than those viewed in the previous step) using a rating of 1 star – 5 stars
- Making 4 purchases on the store on different days

Then, for each user, a recommendation of suitable products was calculated based on our proposed hybrid recommendation system using an expert system. The system offered 40 products to each user, and for each product, users indicated whether it was “correctly” or “incorrectly” recommended - i.e., whether it was a relevant product based on user preferences or not.

The results are shown in Table 13. Precision indicates the ratio of films marked as relevant to all recommended products. Recall indicates the ratio of products marked as Relevant relative to the list of the top 15 recommended products.

The results are also depicted in Fig. 8.

The results are promising. The system recommended a high number of relevant products in all cases, with a maximum of 40 products in eight cases. Precision, therefore, reaches a slightly lower value in one case (93 %), while in the other cases, the values are high (95 %-100 %). The Recall metric indicates the proportion of products marked as Relevant in the list of the top 15 recommended products. In all cases, Recall was 100 % because none of the products marked as non-relevant appeared in the TOP 15 recommended products list. The chart further shows that on

Table 13
Precision, Recall and F1-measure metrics assessment.

User	Relevant	Irrelevant	Precision	Recall	F1-measure
1	40	0	100 %	100 %	100 %
2	40	0	100 %	100 %	100 %
3	40	0	100 %	100 %	100 %
4	40	0	100 %	100 %	100 %
5	40	0	100 %	100 %	100 %
6	40	0	100 %	100 %	100 %
7	39	1	98 %	100 %	99 %
8	38	2	95 %	100 %	97 %
9	38	2	95 %	100 %	97 %
10	38	2	95 %	100 %	97 %
11	38	2	95 %	100 %	97 %
12	38	2	95 %	100 %	97 %
13	37	3	93 %	100 %	96 %
14	40	0	100 %	100 %	100 %
15	39	1	98 %	100 %	99 %
16	40	0	100 %	100 %	100 %
17	39	1	98 %	100 %	99 %
Total	39	1	98 %	100 %	99 %

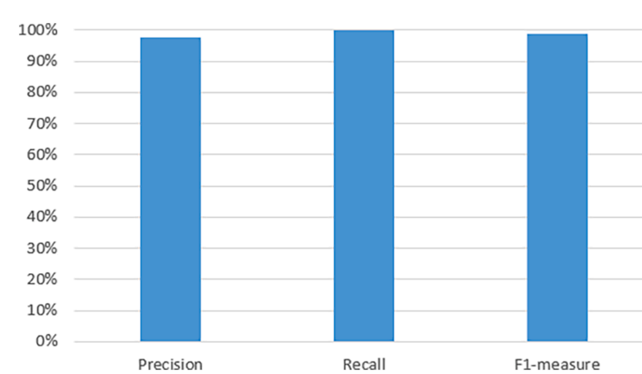

Fig. 8. Precision, Recall and F1-measure metrics assessment.

average, Precision, Recall and F1-measure values reached 98 %, 100 % and 99 % respectively, during the testing, which are promising values.

4.1. Comparison of the proposed approach with other standard approaches in recommender systems

Furthermore, we compared our system with other standard approaches in the field of recommender systems. The reason for this comparison is to see what results our proposed system achieves compared to other standard approaches in terms of the proportion of relevant products tagged by users. The comparison was made with the following systems:

- Content-based filtering system 1 - the system uses the TF-IDF algorithm and recommends products based on the history of purchased products. The system also includes the calculation of a coefficient based on the user's rating of the product. In addition, recommended products receive a coefficient based on the number of products in the user's purchase history.
- Content-based filtering system 2 - the system uses the TF-IDF algorithm and recommends products based on the history of products purchased. The system also calculates a coefficient based on the user's rating of the product. The recommended products get an extra coefficient if they are from the same category as the products the user has purchased. For example, if the user has purchased several products from the Bakery category, then the recommended products from the Bakery category get an extra coefficient of 0.5.
- Eshop recommender - Hybrid system with an expert system - our proposed recommender system

The same group of users was chosen to test these 3 systems. Each of the 3 systems suggested 40 recommended products to the user. The users were then asked to mark relevant and irrelevant products in each of the tested systems.

The proportion of relevant products marked in all systems is shown in Table 14.

The summary results are shown in the graph in Fig. 9.

The results of the comparison are also promising. Content-based filtering system 1 recommended 96 % of relevant products. Content-based filtering system 2 recommended 97 % of relevant products. Our proposed system achieved the best results. The system recommended 98 % of the products that were identified by users as relevant.

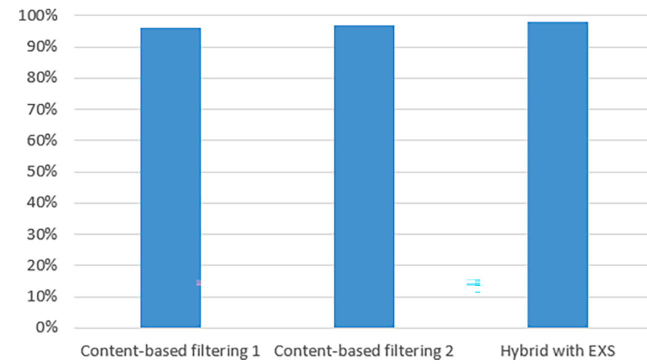

Fig. 9. Comparison of the proposed approach with other standard approaches.

4.2. Comparison of the proposed recommender system with the e-shop Kosik.cz

Next, we compared our proposed recommendation system with one of the largest online grocery stores in the Czech Republic. This is the online store Kosik.cz.

This online shop contains the same products as our proposed recommender system. The Kosik.cz online store also includes a Recommended for You section, which is displayed to the user before completing the purchase to allow them to purchase additional products and then complete the purchase.

The same group of users was chosen to test this online shop. The users were asked to make a purchase (adding to the basket) of pre-selected products. They were then asked to mark relevant and non-relevant products, which were then displayed in the Recommended for You section. In our proposed system, users made a purchase of the same products and then had to mark the relevant and non-relevant products that the system suggested to them based on their purchase. The results of the testing are shown in Table 15:

The results are very promising. The online shop Kosik.cz recommended 59 % of relevant products. Our proposed system then recommended 100 % of the products that were marked as relevant by users. However, one important fact should be noted here. Our proposed system recommends products based on user preference and behaviour. The online store Kosik.cz can also recommend non-personalised content as part of its own recommendation algorithm in order to offer a larger range to the active user and thus make more profits, which could be the reason for a lower ratio of relevant products during testing. Therefore, this possibility should be mentioned here.

Table 14
Comparison of the proposed approach with other standard approaches.

User	Content-based filtering 1	Content-based filtering 2	Hybrid with EXS – Eshop recommender
1	99 %	98 %	100 %
2	100 %	100 %	100 %
3	99 %	100 %	100 %
4	91 %	90 %	100 %
5	100 %	100 %	100 %
6	89 %	98 %	100 %
7	91 %	93 %	98 %
8	99 %	98 %	95 %
9	87 %	95 %	95 %
10	94 %	92 %	95 %
11	91 %	92 %	95 %
12	90 %	97 %	95 %
13	98 %	100 %	93 %
14	100 %	100 %	100 %
15	100 %	100 %	98 %
16	100 %	100 %	100 %
17	96 %	95 %	98 %
Total	96 %	97 %	98 %

Table 15
Comparison of the proposed recommender system with the e-shop Kosik.cz.

User	Kosik.cz	Hybrid with EXS – Eshop recommender
1	61 %	100 %
2	56 %	100 %
3	56 %	100 %
4	58 %	100 %
5	61 %	100 %
6	64 %	100 %
7	64 %	100 %
8	61 %	100 %
9	44 %	100 %
10	58 %	100 %
11	64 %	100 %
12	61 %	100 %
13	61 %	100 %
14	64 %	100 %
15	58 %	100 %
16	56 %	100 %
17	50 %	100 %
Total	59 %	100 %

4.3. Results of the feedback to the system

Part of the testing was also filling in feedback on the system in the form of a short questionnaire. Users answered the following questions:

1. Did the proposed products match your preferences?
2. Were the recommended products relevant to your activity in the recommender system?
3. Were the products in the Kosik.cz online shop relevant to your shopping cart?
4. Were the products in the recommender system more relevant than in the Kosik.cz online shop?

The results are shown in Figs. 10–13.

The results of the survey among respondents are promising. 76 % (13 of 17) answered "Yes" to the first question and 24 % (4 of 17) answered "Rather Yes". No respondents answered "Rather No" or "No". For the second question, 24 % (13 of 17) answered "Yes" and 24 % (4 of 17) answered "Rather Yes". None of the respondents answered "Rather No" or "No". The responses to the first question are important because they relate to the ability of the system to recommend relevant products to the user based on the user's preferences. The answers to the second question are also very important because they relate to the ability of the system to recommend relevant products to the user in the course of working with the online shop (browsing products, rating products, buying products).

The answers to the third question concerned the ability of the Kosik.cz online store (which was used for comparison with our system) to recommend relevant products based on the products in the user's shopping cart. Over 6 % of respondents (1 out of 17) answered "Yes" to this question, while 47 % (8 out of 17) answered "Rather Yes". Over 35 % (6 of 17) of respondents answered "Rather No" and 12 % (2 of 17) answered "No". These results show that more than half of the users think that the compared online shop Kosik.cz is able to recommend relevant products based on the shopping cart. 47 % of the respondents think that the online shop Kosik.cz does not recommend relevant products based on the shopping cart.

The answers to the fourth question were related to the ability of our proposed store to suggest relevant products better than the online store Kosik.cz. <12 % of respondents (2 out of 17) answered "Yes" to this question, while 88 % (15 out of 17) answered "Rather Yes". This shows that respondents think that our proposed system is able to suggest more relevant products based on the shopping cart than the compared online store Kosik.cz.

However, it should be emphasised that the respondents' answers to both questions are subjective, so this evaluation on this set of test users is rather indicative.

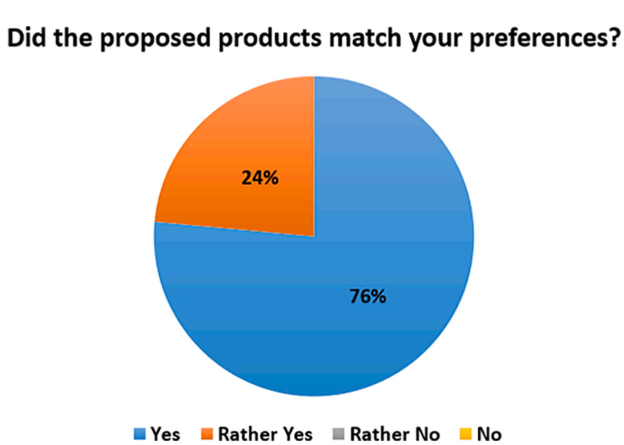

Fig. 10. Results of the product relevancy to user preferences.

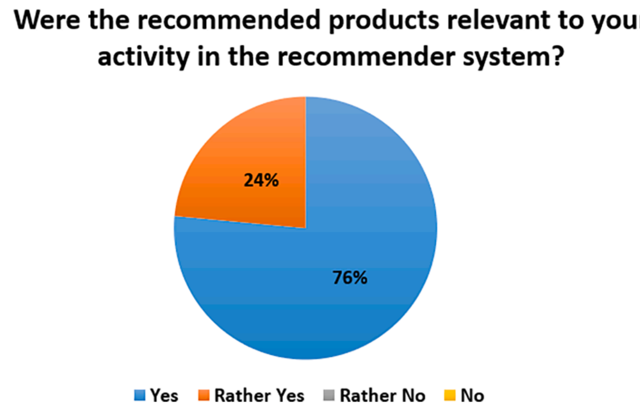

Fig. 11. Results of the product relevancy to the activity in the e-shop.

Fig. 12. Results of the product relevancy in the e-shop Kosik.cz to the shopping cart.

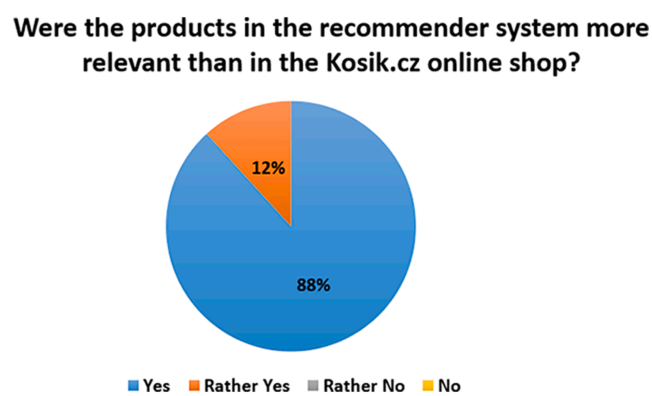

Fig. 13. Results of the product relevancy in the proposed recommendation system to the product relevancy in the e-shop Kosik.cz.

5. Conclusion

The results presented in the Results chapter show that our proposed recommender system achieves promising results. Users who tested the recommender system marked most of the recommended products as relevant. Also, compared to other traditional recommender system approaches, our proposed system recommended the most products marked as relevant.

In the case of comparison with the Czech online shop Kosik.cz, the aim was to compare the recommended products based on the products placed in the shopping cart. Our proposed system recommended more products marked as relevant by users than the online shop Kosik.cz. Also, the user questionnaire showed that the proposed recommender system was able to suggest relevant products based on user preferences and ongoing activity in the online store (viewing and rating products, purchasing products).

This article proposes a hybrid Eshop recommender system, which combines a recommender module composed of three subsystems and an expert system. The first subsystem recommends relevant products with respect to the viewed products and uses a content-based approach. The second subsystem recommends relevant products relative to the products evaluated and uses a collaborative-filtering approach. The third subsystem recommends relevant products given a user's purchase history and uses a content-based approach. The main role of the expert system is to evaluate the relevance of the recommended products and rank them appropriately in the resulting list of recommended products for the user. The ranking of products by the expert system is created based on the similarity measure with the rated products, the coefficient of the purchased product and the average rating of the product.

The results presented in this article have several practical implications:

- The recommender system works with user preferences (favourite categories, products viewed, products rated, purchases made).
- The recommender module combines recommendations based on products viewed, products rated and given the user's purchase history.
- The use of a fuzzy expert system that evaluates the importance of recommended products based on various parameters (similarity rate with rated products, coefficient of purchased product and average product rating). IF-THEN expert system rules can be modified and adjusted, the expert system can also be extended with additional parameters
- Combination of collaborative-filtering approach, content-based approach and fuzzy expert system to calculate the final list of recommended movies
- Based on the experimental validation results, promising values of the standard metrics Precision – 98 %, Recall – 100 % and F1-measure – 99 % were achieved. Also, when compared to other standard approaches, our approach achieved the highest ratio of relevant products that users flagged during the system testing process
- The proposed hybrid system is fully implemented in the form of an Eshop recommender web application. The system is fully available at the URL: https://eshop.osu-vyuka.cz/. Thus, the system functionalities can be tested.

Our proposed recommeder system differs from other hybrid approaches in its interconnection with an expert system, which evaluates the importance of recommended products based on various parameters. This expert system can be modified and extended with other parameters.

5.1. Future work

In future work, we would like to focus on several areas.

The first area is working with user preferences. Within user preferences, it would be useful to be able to mark not only favourite product categories but also unpopular product categories whose products would be penalised for the final recommendation. It would also be useful to be able to mark specific favourites and dislikes. User preferences can change over time, so it is also important that the system continuously assesses the popularity and unpopularity of individual categories and products and adjusts this data dynamically. In practice, it happens that the popularity of some categories marked during registration to the system may change, and therefore, it is necessary to react to the change of user preferences.

Another area is the continuous evaluation of purchases made in the online shop. It would be useful to evaluate the purchases of all users and calculate the similarity of products purchased between users. This would make it possible to find out which users purchase similar products to the current user. And based on the statistical analysis and collaborative-filtering approach, it would then be possible to suggest relevant products to the user that are purchased by users with similar product purchases.

Next, we would like to focus on analysing individual user's purchases and identify which products he/she buys most frequently. Based on this data, it would then be possible to dynamically suggest to the user the most appropriate products (most frequently purchased products) relative to those that the user puts in the online store shopping cart. The result would be a faster shopping experience for the user who repeatedly purchases the same products.

Furthermore, we would like to validate the proposed system on another type of online store - e.g. an online store with home accessories.

CRediT authorship contribution statement

Bogdan Walek: Conceptualization, Formal analysis, Methodology, Project administration, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Petr Fajmon:** Methodology, Software, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

The research was supported by a Student Grant SGS20/PřF-MF/2021 with student participation, supported by the Czech Ministry of Education, Youth and Sports.

References

Abbasi, F., Khadivar, A., & Yazdinejad, M. (2019). A grouping hotel recommender system based on deep learning and sentiment analysis. *Journal of Information Technology Management, 11*(2), 59–78.
Aggarwal, C. C. (2016). *Recommender systems: 1*. Cham: Springer International Publishing.
Anwar, T., & Uma, V. (2021). Comparative study of recommender system approaches and movie recommendation using collaborative filtering. *International Journal of System Assurance Engineering and Management, 12*(3), 426–436.
Barragáns-Martínez, B., Costa-Montenegro, E., & Juncal-Martínez, J. (2015). Developing a recommender system in a consumer electronic device. *Expert Systems with Applications, 42*(9), 4216–4228.
Biswas, P. K., & Liu, S. (2022). A Hybrid Recommender System for Recommending Smartphones to Prospective Customers. *Expert Systems with Applications,* 118058.
Burke, R. (1999, July). The wasabi personal shopper: A case-based recommender system. In AAAI/IAAI (pp. 844-849).
Cao, Y., & Li, Y. (2007). An intelligent fuzzy-based recommendation system for consumer electronic products. *Expert Systems with Applications, 33*(1), 230–240.
Carrer-Neto, W., Hernández-Alcaraz, M. L., Valencia-García, R., & García-Sánchez, F. (2012). Social knowledge-based recommender system. Application to the movies domain. *Expert Systems with applications, 39*(12), 10990–11000.
Cheung, K. W., Kwok, J. T., Law, M. H., & Tsui, K. C. (2003). Mining customer product ratings for personalized marketing. *Decision Support Systems, 35*(2), 231–243.
Colombo-Mendoza, L. O., Valencia-García, R., Rodríguez-González, A., Alor-Hernández, G., & Samper-Zapater, J. J. (2015). RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Systems with Applications, 42*(3), 1202–1222.
Falk, K. (2019). Practical recommender systems (1st ed.). Shelter Island: Manning ISBN 1-61729-270-2 .
Fan, H., Wu, K., Parvin, H., Beigi, A., & Pho, K. H. (2021). A hybrid recommender system using knn and clustering. *International Journal of Information Technology & Decision Making, 20*(02), 553–596.
Garfinkel, R., Gopal, R., Tripathi, A., & Yin, F. (2006). Design of a shopbot and recommender system for bundle purchases. *Decision Support Systems, 42*(3), 1974–1986.

Ghani, R., & Fano, A. (2002). May). Building recommender systems using a knowledge base of product semantics. In *In Proceedings of the Workshop on Recommendation and Personalization in ECommerce at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web based Systems* (pp. 27–29).

Habiballa, H., Novák, V., Dvořák, A., & Pavliska, V. (2003). Using software package LFLC 2000. In *In 2nd International Conference Aplimat* (pp. 355–358).

Ho, A. T., Menezes, I. L., & Tagmouti, Y. (2006). In *E-mrs: Emotion-based movie recommender system* (pp. 1–8). USA: University of Washington Both-ell.

Huang, T. C. K., Chen, Y. L., & Chen, M. C. (2016). A novel recommendation model with Google similarity. *Decision Support Systems, 89*, 17–27.

Huang, Z., Chung, W., & Chen, H. (2004). A graph model for E-commerce recommender systems. *Journal of the American Society for information science and technology, 55*(3), 259–274.

Kaminskas, M., Bridge, D., Foping, F., & Roche, D. (2017). Product-seeded and basket-seeded recommendations for small-scale retailers. *Journal on Data Semantics, 6*(1), 3–14.

Kiran, R., Kumar, P., & Bhasker, B. (2020). DNNRec: A novel deep learning based hybrid recommender system. *Expert Systems with Applications, 144*, Article 113054.

Kumar, M., Yadav, D. K., Singh, A., & Gupta, V. K. (2015). A movie recommender system: Movrec. *International Journal of Computer Applications, 124*(3).

Lavanya, R., Khokle, T., & Maity, A. (2021). Review on hybrid recommender system for mobile devices. In *Artificial Intelligence Techniques for Advanced Computing Applications* (pp. 477–486). Singapore: Springer.

Lawrence, R. D., Almasi, G. S., Kotlyar, V., Viveros, M., & Duri, S. S. (2001). *Personalization of supermarket product recommendations. In Applications of data mining to electronic commerce* (pp. 11–32). Boston, MA: Springer.

Li, Y. M., Chou, C. L., & Lin, L. F. (2014). A social recommender mechanism for location-based group commerce. *Information Sciences, 274*, 125–142.

Li, P., & Yamada, S. (2004, December). A movie recommender system based on inductive learning. In IEEE Conference on Cybernetics and Intelligent Systems, 2004. (Vol. 1, pp. 318-323). IEEE.

Logesh, R., Subramaniyaswamy, V., & Vijayakumar, V. (2018). A personalised travel recommender system utilising social network profile and accurate GPS data. *Electronic Government, an International Journal, 14*(1), 90–113.

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support System, 74*, 12–32.

McCarthy, K., Reilly, J., McGinty, L., & Smyth, B. (2004, August). Thinking positively-explanatory feedback for conversational recommender systems. In Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04) Explanation Workshop (pp. 115-124).

Mooney, R. J., & Roy, L. (2000). June). Content-based book recommending using learning for text categorization. In *In Proceedings of the fifth ACM conference on Digital libraries* (pp. 195–204).

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web* (pp. 325–341). Berlin, Heidelberg: Springer.

Pradeep, I. K., & JayaBhaskar, M. (2018). Comparative analysis of recommender systems and its enhancements. International Journal of. *Engineering & Technology, 7*(3.29), 304–310.

Ravi, L., & Vairavasundaram, S. (2016). A collaborative location based travel recommendation system through enhanced rating prediction for the group of users. *Computational Intelligence and Neuroscience, 2016*, 7.

Rebelo, M.Â., Coelho, D., Pereira, I., & Fernandes, F. (2021). In *December). A New Cascade-Hybrid Recommender System Approach for the Retail Market* (pp. 371–380). Cham: Springer.

Rodríguez-García, M.Á., Colombo-Mendoza, L. O., Valencia-García, R., Lopez-Lorca, A. A., & Beydoun, G. (2015). In *Ontology-based music recommender system* (pp. 39–46). Cham: Springer.

Schafer, J. B., Konstan, J., & Riedl, J. (1999). November). Recommender systems in e-commerce. In *In Proceedings of the 1st ACM conference on Electronic commerce* (pp. 158–166).

Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data mining and knowledge discovery, 5*(1), 115–153.

Seth, R., & Sharaff, A. (2022). A Comparative Overview of Hybrid Recommender Systems: Review, Challenges, and Prospects. *Data Mining and Machine Learning Applications*, 57–98.

Stanley, L., Lorenzi, F., Saldaña, R., & Licthnow, D. (2003). A tourism recommender system based on collaboration and text analysis. *Information Technology and Tourism, 6*(3), 157–165.

Symeonidis, P., Nanopoulos, A., & Manolopoulos, Y. (2009). October). MoviExplain: A recommender system with explanations. In *In Proceedings of the third ACM conference on Recommender systems* (pp. 317–320).

Tareq, S. U., Noor, M. H., & Bepery, C. (2020). Framework of dynamic recommendation system for e-shopping. *International Journal of Information Technology, 12*(1), 135–140.

Vahidi Farashah, M., Etebarian, A., Azmi, R., & Ebrahimzadeh Dastjerdi, R. (2021). A hybrid recommender system based-on link prediction for movie baskets analysis. *Journal of Big Data, 8*(1), 1–24.

Véras, D., Prota, T., Bispo, A., Prudêncio, R., & Ferraz, C. (2015). A literature review of recommender systems in the television domain. *Expert Systems with Applications, 42*(22), 9046–9076.

Walek, B., & Fojtik, V. (2020). A hybrid recommender system for recommending relevant movies using an expert system. *Expert Systems with Applications, 158*, Article 113452.