Ministry of Education and Science of Ukraine

Odessa National Academy of Food Technologies

International Competition of Student Scientific Works

BLACK SEA SCIENCE 2021

Information Technology, Automation and Robotics

Proceedings

Odessa, ONAFT 2021

UDC 004.01/08

Editorial board:

Black Sea Science 2021: Proceedings of the International Competition of Student Scientific Works. Information Technology, Automation and Robotics. / Odessa National Academy of Food Technologies; B.Yegorov, M. Mardar, S.Kotlyk (editors-in-chief.) [*et al.*]. – Odessa: ONAFT, 2021. – 526 p.

These materials of International Competition of Student Scientific Works «Black Sea Science 2021» contain the works of the contest participants in the section «Information technologies, automation and robotics» (not winners).

The author of the work is responsible for the accuracy of the information.

Odessa National Academy of Food Technologies, 2021

Organizing committee:

Prof. Bogdan Iegorov, D.Sc., Rector of Odessa National Academy of Food Technologies, Head of the Committee

Prof. Maryna Mardar, D.Sc., Vice-Rector for Scientific and Pedagogical Work and International Relations of Odessa National Academy of Food Technologies, Deputy Head of the Committee

Prof. Stefan Dragoev, D.Sc., Vice-Rector for Scientific Work and Business Partnerships of University of Food Technologies (Bulgaria)

Prof. Baurzhan Nurakhmetov, D.Sc., First Vice-Rector of Almaty Technological University (Kazakhstan)

Prof. Mircea Bernic, Dr. habil., Vice-Rector for Scientific Work of Technical University of Moldova (Moldova)

Prof. Jacek Wrobel, Dr. habil., Rector of West Pomeranian University of Technology (Poland)

Prof. Michael Zinigrad, D.Sc., Rector of Ariel University (Israel)

Dr. Mei Lehe, Ph.D., Vice-President of Ningbo Institute of Technology, Zhejiang University (China)

Prof. Plamen Kangalov, Ph.D., Vice-Rector for Academic Affairs of “Angel Kanchev” University of Ruse (Bulgaria)

Dr. Alexander Sychev, Ph.D., Assoc. Professor of Sukhoi State Technical University of Gomel (Belarus)

Dr. Hanna Lilishentseva, Ph.D., Assoc. Professor, Head of the Department of Merchandise of Foodstuff of Belarus State Economic University (Belarus)

Prof. Heinz Leuenberger, Ph.D., Professor of the Institute of Ecopreneurship of University of Applied Sciences and Arts (Switzerland)

Prof. Edward Pospiech, Dr. habil., Professor of the Institute of Meat Technology of Poznan University of Life Sciences (Poland)

Prof. Lali Elanidze, Ph.D., Professor of the Faculty of Agrarian Sciences of Iakob Gogebashvili Telavi State University (Georgia)

Dr. V. Kozhevnikova, Ph.D., Senior Lecturer of the Department of Hotel and Catering Business of Odessa National Academy of Food Technologies, Secretary of the Committee

The jury for the section «Information technologies, automation and robotics»

Head of the jury:

Sergii Kotlyk – Ph.D., Associate Professor, Director of the P.M. Platonov Educational-Scientific Institute of Computer Systems and Technologies "Industry 4.0" of Odessa National Academy of Food Technologies (Ukraine)

Members of the jury:

Piotr Artiemjew - Dr hab., Associate Professor in Decision Systems of the Faculty of Mathematics and Computer Science,University of Warmia and Mazury in Olsztyn (Poland)

Francisco Antonio Augusto – Dr., International Relations Manager of Higher Institute of Information and Communication Technologies (Angola)

Andrey Kuprijanov – Ph.D., Associate Professor of the Department of Software for Computers and Automated Systems of Belarusian National Technical University (Belarus)

Simon Milbert – Vice-President of Xtra Information Management, Inc. (USA)

Ivan Palov – D.Sc., Professor of University of Ruse "Angel Kanchev" (Bulgaria)

Degla Gérard Hugues – Communications and Training Manager of "MAPCOM solutions informatiques" company group (Benin)

Nugzar Kereselidze - Academic Doctor of Informatics (Computer Science), Associate Professor of the Department of Natural Sciences, Mathematics, Technology and Pharmacy, Sukhumi State University (Georgia)

Etibar Seyidzade - Associate Professor of the Department of Computer and Information Technolohies, Baku Engineering University (Azerbaijan)

Vladimir Golenkov, D.Sc., Professor of the Department of Intelligent Information Technologies, Belarusian State University of Informatics and Radio Electronics (Belarus)

Zhanar Omirbekova - Ph.D., Associate Professor of the Department of Automation and Management, Satbayev University (Kazakhstan)

Ivan Palov - D.Sc., Professor of the Department of Power Supply and Electrical Equipment, University of Ruse "Angel Kanchev" (Bulgaria)

Siarhei Palavenia - Ph.D., Associate Professor, Head of the Department of Telecommunication Systems, Belarusian State Academy of Communications (Belarus)

Alexander Goloskokov - Ph.D., Professor of the Department of Software Engineering and Information Technology Management, National Technical University "Kharkiv Polytechnic Institute" (Ukraine)

Peter Nikolyuk - D.Sc., Professor of the Department of Computer Technology, Vasyl Stus Donetsk National University (Ukraine)

Vladimir Palagin - D.Sc., Professor, Head of the Department of Radio Engineering, Telecommunications and Robotics Systems, Cherkasy State Technological University (Ukraine)

Viktor Khobin – D.Sc., Professor, Head of the Department of Technological Processes Automation and Robotic Systems of Odessa National Academy of Food Technologies (Ukraine)

Valeriy Plotnikov – D.Sc., Professor, Head of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies (Ukraine)

Sergii Artemenko – D.Sc., Professor, Head of the Department of Computer Engineering of Odessa National Academy of Food Technologies (Ukraine)

Fedir Trishyn - Ph.D., Associate Professor, Vice-Rector on Scientific and Educational Work, Odessa National Academy of Food Technologies (Ukraine)

Valerii Levinskyi – Ph.D., Associate Professor of the Department of Technological Processes Automation and Robotic Systems of Odessa National Academy of Food Technologies (Ukraine)

Viktor Yehorov – Ph.D., Supervisor of the Laboratory of Mechatronics and Robotics of Odessa National Academy of Food Technologies (Ukraine)

Pavlo Lomovtsev – Ph.D., Associate Professor of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies (Ukraine)

Yurii Kornienko – Ph.D., Associate Professor of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies (Ukraine)

Serhii Shestopalov – Ph.D., Associate Professor of the Department of Computer Engineering of Odessa National Academy of Food Technologies (Ukraine)

Anatoly Galiulin - Ph.D., Associate Professor, Acting Head of the Department of Electromechanics and Mechatronics, Odessa National Academy of Food Technologies (Ukraine)

Secretary of the jury:

Oksana Sokolova – Senior Lecturer of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies (Ukraine)

DEVELOPMENT OF A RECOMMENDATION SYSTEM

Author: ***Valeryia Runets***
Advisor: ***Vadzim Sakovich***
Belarusian State University (Belarus)

Abstract. *The existing algorithms for developing recommendation systems were studied, the shortcomings in various approaches were examined, and a solution was proposed.*

Keywords: *recommendation systems, search, object, metrics, matrix factorization.*

I. INTRODUCTION

In today's rapidly developing world, where there is a wide variety of different types of products, goods, services, the value of recommendation systems is growing. It is impossible to underestimate the role of recommendation systems in the life of a modern person. Every day, a large number of people are busy searching the Internet for certain goods and services. Recently, recommendation systems have been used to improve the search on various Internet resources.

The purpose of recommendation systems is to inform the user about an object that may interest him. Recommendation systems have significantly improved user interaction with Internet resources. They are also an excellent tool for businesses, helping to successfully promote and develop their product, service or web service.

Nowadays, it is quite easy for the user to find the object of interest, the search is greatly simplified, since the user is not forced to spend most of the time on the Internet in order to find the product or service that he needs. Now you can download the second part of the book you like in just a few clicks, order an additional service when purchasing a product, or watch a movie with your favorite actor. And all this is available thanks to recommendation services.

The development of recommendation services is not limited to using a small number of algorithms. Nowadays, neural networks are often used for development, and algorithms are being improved very quickly. However, there is still no universal method for predicting a person's liking.

II. LITERATURE ANALYSIS

2.1. Definition of recommendation systems

Recommendation systems are programs that tell the user which entities (movies, music, books, news, websites, products) will be of interest to him and how much he will like them. The work of the recommendation system is to select objects based on his personal preferences and tastes. Each user receives his personal recommendation based on various types of information about the user [1]. These predictions can be based both on information received from the user earlier and saved in his profile, and on the basis of information about other users. The process of work of the recommendation system is clearly shown in Fig. 1.

Fig. 1. The process of the recommendation system

Recommendation systems provide a personalized approach to each user.

Recommendation systems are used on websites with different content. For example, recommendations for films, books, music, goods, various promotions. Each of these cases has its own approaches and solutions. Let's take a look at the differences between some of the recommendation services:

1. Recommendations for films: in this example, it is logical for the user to recommend new films that may be of interest to him;

2. Promotions recommendations: in this case, it makes sense for the user to offer the most profitable promotions on products that might be of interest to him.

The two main strategies for creating recommendation systems are content methods and methods of collaborative filtering, the principle and differences in their work are presented in Fig. 2. A combination of the two strategies is often used.

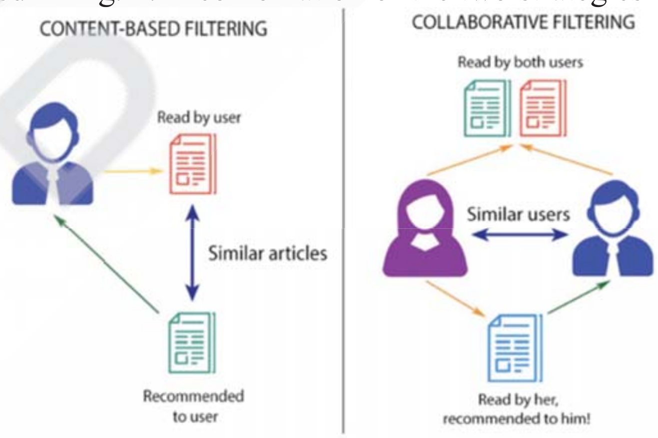

Fig. 2. Recommended systems operation strategies

Recommendation services are used in various fields. They can be seen in such sections as: "Similar books", "They also buy with this", "You may be interested" and others. There are even specialized websites dedicated to recommending certain products. Some directories work on a recommendation basis.

The main question in developing a recommendation system is "What is a similar object?"

Different approaches can be used to solve this problem:

• similar objects are objects that are similar in their characteristics (content-oriented methods);

• similar items are items that are often used together ("customers who bought i also bought j");

• similar objects are recommendations to the user who liked the given object;

• similar objects are simply guidelines in which the given object acts as a context.

Recommendation systems are a good business solution these days. High-quality recommendations reduce the time users need to search for products and services, and significantly increase the likelihood of getting into the field of view of other objects that may be of interest to him. This can increase the number of sales of even not the best-selling products. The result is increased user satisfaction with web services. As a result, the user will visit this web service more often, and the chance that he will return increases. In addition, the user interacts with those products that he would not have paid attention to if it were not for the recommendation service, which leads to an increase in consumption and an increase in profits. In addition, newsletters, personalized advertisements and push notifications encourage users to come back, increase the frequency of visits by repeat users, and reduce customer churn [3].

Depending on the purpose and place of use, recommendation systems can be both the basis of a web resource and an auxiliary service that allows you to attract customers, increase the percentage of sales, or simplify site navigation for users.

If we take recommendation systems in online business, they usually have two goals:

1. Inform the user about an interesting product;

2. Encourage him to make a purchase (by mailing, making a personal offer, etc.).

2.2. Approaches to the development of recommendation systems

The essence of content-oriented techniques is to map the user to objects based on the products they have purchased or the pages they viewed.

In content-oriented methods, a profile with information about them is required for each user and object. The user profile should contain the characteristics of the products that he likes. This data can be obtained in various ways:

• polling the user or filling out a questionnaire to get information about his favorite properties of the product;

• analysis of user actions and, as a consequence, obtaining characteristics of objects that he likes.

The profile of objects is filled in when adding an object to the database. Characteristics can be different, depending on the category of the object [2].

Content-oriented methods involve searching for similarities between products by objective characteristics: color, size, genre, etc. The more characteristics, the more accurately the object is described, however, it should be understood that some characteristics may not say anything about the user's preferences. These characteristics can be:

• color of the book cover;

• the year the film was released.

Most recommendation systems based on content-oriented methods offer the same type of recommendation when viewing a product.

Collaborative filtering uses information about the activity of all users on the network:

• visiting certain websites;

• reviews about objects;

• assessment of objects.

In this case, the subjective properties of objects are taken into account, which are difficult to obtain without statistics of certain activity from the user. The advantage of this filtering is that it does not matter what types of objects you are working with, but implicit characteristics can be taken into account that would be difficult to take into account when using object-oriented methods.

In the process, recommendation systems collect data about users using a combination of explicit and implicit methods.

Implicit data collection can be such actions as:

• tracking and storing data on user behavior online;

• tracking the contents of the user's computer;

• tracking objects that the user views on the network.

Examples of explicit data collection:

• selection of the most preferable of the two objects;

• use of a differentiable scale for evaluating objects;

• placing a group of objects in descending or increasing order of the user's interest in the object;

• compiling a list of objects that characterize a specific user.

Recommendation systems compute a list of recommendations for a specific user by comparing information received from all users of the system. Since the user can find objects with the help of the recommendation system that could not be found by the search engine, the recommendation systems are a good alternative. However, it is interesting that often search engines are used by recommendation [2].

There are also two different approaches to collaborative filtering:

1. Filtering based on user similarity. In this approach, the system, according to certain algorithms and criteria, finds a user with similar preferences and offers a set of objects that were interesting to him;

2. Filtering based on the similarity of objects. In this approach, the system searches for objects that are liked by the same user group, such objects are considered similar and are offered to users who liked objects from this group.

The principle of operation and the differences between these approaches are clearly shown in Fig. 3.

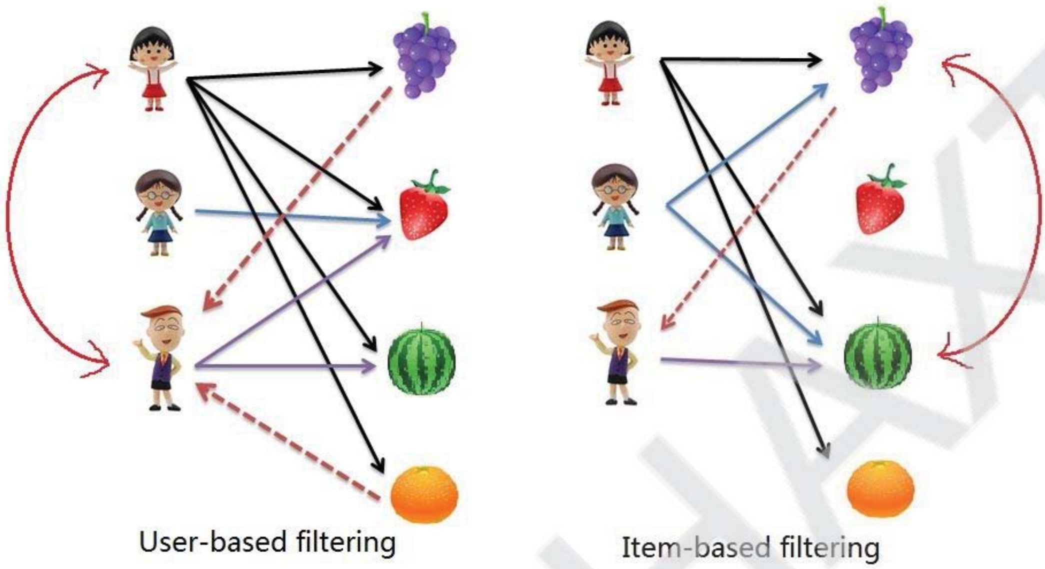

Fig. 3. Collaborative filtering approaches

Advantages of filtering based on similarity of objects over filtering based on similarity of users:

• When the dimension of the problem is large, i.e. when there are a lot of users, the problem of finding the nearest neighbor becomes difficult to compute. Computational complexity is reduced from O (N ^ 2 * n) to O (n ^ 2 * N) when using filtering based on object similarity.

• Filtering based on user similarity usually results in a sparse matrix; there are a lot of goods, but there are few estimates On the one hand, this helps to optimize the calculation - we work only with those elements that intersect. But on the other hand, the list of recommended products is very small.

• Estimating the proximity of goods is usually much more accurate than estimating the proximity of users. Consequently, the error in calculating the correlation of goods there is significantly less. This is a direct consequence of the fact that there are usually many more users than products.

• Product descriptions are much more consistent than user descriptions because they are user preferences can change over time [6].

2.3. Algorithms

The goal of content-oriented techniques is to find similarities. Similarities can be looked for in two ways:

• between the object and the user;

• between two objects.

Various methods can be used to find similarities between entities. Next, 2 methods will be considered:

1. Cosine similarity. In this method, the coefficient of similarity between two objects is calculated using the formula (1).

$$cosine(x,y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \quad (1)$$

where x, y are comparison objects, and xi, yi are characteristics of these objects

This method is useful for data with high-dimensional features. This method will find similar objects to those that the user liked, and these objects will be recommended to him.

1. Jaccard similarity (also called intersection over union).

Used to find a property between two objects. Calculated by the formula (2).

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \quad (2)$$

However, Jaccard's similarity only applies when the characteristics have binary values. If the characteristics are a rating, a score, or other properties that take more than 2 values, then the Jaccard similarity is not applicable.

The collaborative filtering approach is to find users who are similar to a given person, in order to offer him those objects that will potentially be of interest to him. Compared to the content-oriented approach, objects and users have no properties or attributes. All calculations are performed on the basis of a utility matrix, an example of which is presented in Table 1.1.

Table 1. Utility matrix

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

The column lists the users and the row lists the objects. Matrix elements are ratings that users have assigned to certain objects. There are 2 methods that can be used for collaborative filtering.

Let's look at a memory-based implementation. In this method, recommendations are composed using a utility matrix function. This function returns the estimated score for the current user.

First, the average rating is calculated, which is formed for user i on the basis of all objects that he has ever rated, according to formula (3).

$$\bar{y}_i = \frac{1}{|I_i|}\sum_{j \in I_i} y_{ij} \quad (3)$$

Using this, we estimate the rating of object k for user i as shown in Fig. 4.

Similarity between users a and i

$$\hat{y}_{ik} = \bar{y}_i + \frac{1}{\sum_{a \in U_k} |w_{ia}|} \sum_{a \in U_k} w_{ia}(y_{ak} - \bar{y}_a)$$

a's rating of k – a's average ratings

All users that have rated k

Fig. 4. Calculating the estimated value of an object for a specific user

The similarity between users a and i can be calculated using the methods described above, for example, cosine similarity, Jaccard similarity, Pearson's correlation coefficient, or any other. Thus, it is very easy to obtain results as long as the data does not become too sparse, in which case performance is degraded.

For the model-driven approach, matrix factorization is most commonly used. In this case, we create user and item views from the utility matrix. The principle of matrix factorization is shown in Fig. 5.

$$\begin{bmatrix} 5 & 1 & 4 & 5 & 1 \\ & 5 & 2 & 1 & 4 \\ 1 & 4 & 1 & 1 & 2 \\ 4 & 1 & 5 & 5 & 4 \\ 5 & 3 & 3 & & 4 \\ 1 & 5 & 1 & 1 & 1 \\ 5 & 1 & 5 & 5 & 4 \end{bmatrix} \approx \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1K} \\ u_{21} & u_{22} & \cdots & u_{2K} \\ u_{31} & u_{32} & \cdots & u_{3K} \\ u_{41} & u_{42} & \cdots & u_{4K} \\ u_{51} & u_{52} & \cdots & u_{5K} \\ u_{61} & u_{62} & \cdots & u_{6K} \\ u_{71} & u_{72} & \cdots & u_{7K} \end{bmatrix} \times \begin{bmatrix} v_{11} & v_{21} & v_{31} & v_{41} & v_{61} \\ v_{12} & v_{22} & v_{32} & v_{42} & v_{62} \\ \vdots & & & & \\ v_{1K} & v_{2K} & v_{3K} & v_{4K} & v_{6K} \end{bmatrix} \approx \begin{bmatrix} 0.2 & 3.4 \\ 3.6 & 1.0 \\ 2.6 & 0.6 \\ 0.9 & 3.7 \\ 2.0 & 3.4 \\ 2.9 & 0.5 \\ 0.8 & 3.9 \end{bmatrix} \times \begin{bmatrix} 0.0 & 1.5 & 0.1 & 0.0 & 0.7 \\ 1.3 & 0.0 & 1.2 & 1.4 & 0.7 \end{bmatrix}$$

Fig. 5. Matrix factorization

Thus, our utility matrix decomposes into matrices U and V, where U represents users and V represents objects in low-dimensional space. To do this, you can use various methods such as SVD (Singular Value Decomposition), PCA (Principal Component Analysis) or others.

Further, the estimated rating is calculated by the formula (4), where i is the user, and j is the object for which it is necessary to calculate the rating.

$$y_{ij} = u_i v_i (4)$$

where u_i, v_i are matrix elements that we got as a result of matrix factorization.

After calculations, you can recommend the properties with the highest predicted rating. This method helps to reduce the dimension of the problem, so it is convenient to use this approach with large amounts of data with high sparsity. But it is worth noting one of the disadvantages of this method - we do not know what the elements of the resulting matrices mean.

If you look at the algorithm in a different way, then the essence of matrix factorization is to create a set of factors that unite all users into some specific clusters. The number of characteristics is directly proportional to the found relationships. Each characteristic has its own meaning, since there is a certain number of users whose interests include it, and such users can be combined into one group or cluster [5].

In practice, more than one approach is almost always used; they are often combined. The two main goals of combining models are to increase forecast accuracy and avoid various problems associated with small user groups. The main disadvantages are the laborious implementation and poor interpretability of the algorithm.

Examples of such combination strategies:

• switching - application of different algorithms for different products and users;

• stacking - predictions of individual models are the inputs of another classifier, which learns to correctly weigh intermediate estimates;

• mixing - calculating recommendations for different algorithms and combining them into a common table;

• weighting - the predicted score is calculated based on several characteristics.

Thus, each algorithm has its own advantages and disadvantages, you can get the maximum benefit by combining them [8].

III. OBJECT, SUBJECT, AND METHODS OF RESEARCH

The main problem with recommendation systems is that they recommend objects that the user has already bought earlier. To solve this problem, you need to choose the right metrics for optimization.

All quality metrics can be roughly divided into 3 categories:

• Prediction Accuracy - assessment of the predicted rating accuracy (applied in case of scale ratings);

• Decision support - assessment of the relevance of recommendations (works only with binary data);

• Rank Accuracy - an assessment of the ranking quality of the issued recommendations, ie. how correct the order of the recommended objects is when the list is sorted.

Metrics are used to assess the accuracy of a method. For this, the projected estimates are compared directly with the actual estimate given by the user [4].

Consider the metrics that assess the accuracy of the predicted rating. Typical statistical metrics are mean absolute error (MAE), root mean square error (RMSE), and correlation. The most popular is the MAE metric. MAE is calculated using formula (5) and RMSE is calculated using formula (6).

$$MAE = \frac{1}{N}\sum |predicted - actual| \quad (5)$$

$$RMSE = \sqrt{\frac{1}{N}\sum (predicted - actual)^2} \quad (6)$$

Recommendations forecast - inversely proportional to the values of the MAE and RMSE metrics, the more accurate the method used to forecast recommendations. These metrics come in handy when recommendations are based on predicting rating or number of transactions. Metrics can be used to assess the quality of the recommendation system [3].

Decision Support class metrics work with binary data (0 and 1, yes and no). If ratings are initially plotted on a continuous scale in a task, they can be converted to a binary format, but in this case the results will be less accurate.

The popular metrics in this class are Precision and Recall. These metrics are necessary so that users can choose the most similar from the available set of objects. The calculation of these metrics is performed using formulas (7) and (8).

$$\text{recommender system precision:} \qquad P = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of items we recommended}} \tag{7}$$

$$\text{recommender system recall:} \qquad r = \frac{\text{\# of our recommendations that are relevant}}{\text{\# of all the possible relevant items}} \tag{8}$$

As a rule, the recommended objects are displayed in a sorted list of several positions (first the best, then in descending order of priority). Rank Accuracy metrics are used to measure the correctness of the order in which recommendations are displayed.

Thus, evaluating metrics are needed to improve the recommendation system, to understand how close the forecast is to real data, which helps to correct future forecasts [4].

But, in addition to the accuracy of the prediction, we may be interested in other things:

• coverage - the share of products that is present in the list of recommendations;

• personalization - how much the recommendations are personalized for each individual user;

• diversity - variety of objects in the list of recommendations.

These characteristics are also important in assessing the quality of the recommendations received.

IV. RESULTS

4.1. Application architecture and operation

The purpose of the developed recommendation system is to recommend the user of films that he might like. The architecture of this application is built on 2 entities:

• user;

• film.

Each user has the opportunity to add information about the films they have watched: leave a review, rate. Thus, the characteristics of the films viewed are not binary (i.e., they can take on more than two values). In this application, the user assigns a score from 1 to 10.

Collaborative filtering is used to find any recommendations.

In the developed application, data about users, movies and views is stored in a MySQL database, the data is retrieved using the JDBC standard.

The algorithm for providing the calculation of recommended grades is developed in the Java language. A user with information about his views is fed to him, at the output we get a list of films with a predicted rating.

4.2. Algorithm operation

Having information about the films viewed by users, a matrix is compiled, presented in table 2.

Table 2. Assessment matrix

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	5	4	5			
User 2	4		5			
User 3		3	5		4	
User 4				3	4	
User 5			4	2	4	
User 6	3					5

In this matrix, the column contains the users, and the row contains the movies. The corresponding mark is indicated at the intersection of the i-th user with the j-th film.

Further, groups are formed, called clusters, where similar users are united. This grouping is done by calculating the average difference between user ratings. Those who have such a difference in the range from 0 to 2.5 are considered similar users.

The user's rating for an object will be predicted as the average cluster rating for this object according to formula (9), and the user is recommended films with the highest conjectural ratings.

$$\hat{r}_{ui} = \frac{1}{|F(u)|} \sum_{v \in F(u)} r_{vi} \quad (9)$$

where r_{ui} is the rating of user u for film i, and F (u) is the cluster distribution function.

The system recommends films with an estimated rating> 5. The list of films is sorted in descending order of estimated ratings and presented to the user.

This method has the problem of recommending films to new users. Since nothing is known about the user, if the user has less than 3 watched movies, the system offers to rate popular movies at each login, hoping that the user has watched them.

Another problem is that recently added movies are not recommended to anyone. In this system, the problem is solved by the fact that each user is recommended a group of films, one of which is a film that has never been watched. Such a movie is selected at random.

4.3. User part

The user part is developed in Java using the Swing library. The interface is represented by a windowed application, which clearly shows the difference between the real user assessment and the predicted by the algorithm.

Let's take a look at how the application works:

Step 1 (shown in Fig. 6). You must enter the id of the user for whom you want to create a list of recommendations.

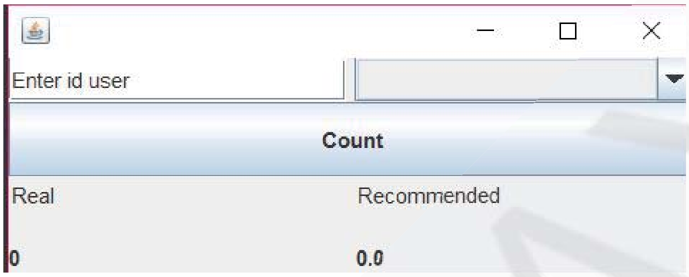

Fig. 6. The first step of the application

Step 2 (shown in Fig. 7). After entering id, the drop-down list displays the movies that this user has already watched and rated. You must select a movie for which the predicted score will be calculated.

Fig. 7. The second step of the application

Step 3 (shown in Fig. 8). When you press the Count button, the selected user is input to the algorithm. The resulting list of recommendations contains the selected movie and its predicted score. And as a result of the application's work, the screen displays the user's real assessment and the predicted one.

Fig. 8. The third step of the application

4.3. Results

The result of the application is a list of movies recommended to this user. Only those films are recommended with an estimated score of> 5 (otherwise the film will not be of interest to the user). The list is displayed in descending order of the estimated estimate.

To assess the work of the developed algorithm, data on the real assessments of some users from the site kinopoisk.ru were taken. They are presented in table 3.

Table 3. User ratings

Film\User	Mistery_girl_	zenitos_rostov	alextokarev1701
«Once upon a time in hollywood»	8	8	10
«Joker»	7	9	10
«Factory»	5	9	8
«Butterfly Effect»	9	8	10
«La-La Land»	7	6	3

For this test, we took users who, under certain conditions, according to the developed algorithm, will be included in one cluster.

The predicted score was calculated for the first user for each of the films. The test was carried out as follows: for the user Mistery_girl_, the data was entered as if one of the films had not been watched, and the score was predicted from it. The predicted and real estimates for the above user are presented in table 4.

Table 4. Algorithm results

Rating type/Film	«Once upon a time in hollywood»	«Joker»	«Factory»	«Butterfly Effect»	«La-La Land»
Predicted	8.0	9.5	8.5	8.0	4.5
Real	8	7	5	9	7

Let's estimate the RMSE (Root Mean Square Error) by the formula (10)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(real(i) - predicted(i))^2} \quad (10)$$

For this sample, an RMSE of 5.5 was obtained. For comparison, Netflix's own algorithm in 2006 predicted user ratings with quality 0.9514 on the RMSE metric. Now this recommendation system is the most technologically advanced in the world and is a combination of 27 recommendation algorithms.

In fact, this value can be interpreted as the distance between two points on the plane, i.e. we get a rather large deviation from the real result.

Since this result turned out to be far from correct, the sample was expanded by 3 times. As a result of the work of the developed algorithm, based on the provided data, RMSE = 1.997 was obtained. This result shows that with an increase in the number of users, the algorithm works more correctly. However, this cannot be considered a regularity, since it is impossible to predict what will happen when the sample is increased by 5, 10, 100 times.

At the next stage of the study, a sample was taken with the number of users = 1000 and the number of films = 11876, and as a result, an RMSE = 1.22 was obtained, which indicates rather accurate predicted estimates during the operation of this algorithm. However, it must be said that this metric shows the average value of the overall work of the algorithm. In some situations, the algorithm does not work correctly and requires further improvements.

Thus, during the operation of the application according to the developed algorithm, some problems were identified: a rather high complexity of this algorithm and some inaccuracies in individual situations.

The results obtained can be saved and used in the future for evaluating metrics. Optimization of this algorithm will lead to a decrease in operating time and an increase in the accuracy of the results.

V. CONCLUSIONS

During the implementation of this project, the following results were obtained:

• requirements for recommendation systems were identified;

• studied and analyzed the main algorithms for the development of recommendation systems;

• developed and implemented an algorithm for solving the problem;

• Possible problems and ways to solve them were identified.

As a result of the work carried out, an algorithm was built that works with small data and gives a result. This system can be optimized to handle large input data correctly. It is necessary to improve the quality of the provided recommendations through the use of assessment metrics.

The constructed algorithm is the basis for further optimization and study of this problem.

VI. REFERENCES

1. Avkhadeev, BR Development of a recommendation system based on data from the profile of the social network "VKontakte" / BR Avkhadeev, LI Voronova, EP Okhapkina // CyberLeninka [Electronic resource]. - 2014. - Access mode: https://cyberleninka.ru/article/n/razrabotka-rekomendatelnoy-sistemy-na-osnove-dannyh-iz-profilya-sotsialnoy-seti-vkontakte. - Date of access: 08.11.2019.

2. Botov, DS Development of a recommendation system for music service based on methods of content and collaborative filtering / DS Botov, NV Menshikova, IV Portnov // Information technologies and systems: tr. Sixth Int. scientific. Conf., Bannoe, March 1-5, 2017 / Feder. issled. Center "Informatics and Management" Ros. acad. Sciences, Chelyab. state un-t; otv. ed .: Yu. S. Popkov, A. V. Melnikov. - Chelyabinsk, 2017 .-- 37 p.

3. Karaush, AS Recommendation systems in public libraries / AS Karaush // Bibliosphere. - 2009. - No. 1. - 41 p.

4. Klemenkov, P. A. Building a real-time news recommendation service using a NoSQL DBMS / P. A. Klemenkov // Informatics and its applications. - 2013. - T. 7, No. 3. - 14 p.

5. Kvyatkovskaya, I. Yu. Information and recommendation system adapted to the cloud-based advertising service / I. Yu. Kvyatkovskaya, PN Savchenko // Vestn. Astrakh. state tech. un-that. Ser .: Management, vychisl. technology and informatics. - 2015. - No. 2. - 88 p.

6. Nefedov, Yu. V. Recommendation systems of Internet services: design and development

stages / Yu. V. Nefedov, IA Galperina // Scientific and technical. inform. Ser. 2, Inform. processes and systems. - 2018. - No. 10. - 29 p.

7. Taratukhina, Yu. V. General principles of designing a recommendatory web service for modeling the individual educational trajectory of students / Yu. V. Taratukhina, MS Markaryan // Open and distance learning. Education [Electronic resource]. - 2016. - No. 2. - 77 p. - Access mode: http://journals.tsu.ru/ou/&journal_page=archive&id=1413&article_id=28949. - Date of access: 01.12.2019.

8. Tregubov, AS Development of methods for adapting user interfaces for USSD services / AS Tregubov // Electronic journals of the Notabene publishing house [Electronic resource]. - 2016. - Access mode: http://e-notabene.ru/kp/article_19497.html. - Date of access: 29.11.2019.

GUITAR TUNER FOR ANDROID OS

Author: ***Andrii Andriichuk***
Advisor: ***Vasyl Lazoryk***
Yuri Fedkovych National University (Ukraine)

Abstract. *The software product was created according to the research. The main function of the program is to process the digital signal and display the corresponding data on the screen of the device, which enables the user to tune the strings on the guitar.*

Keywords: *fast Fourier transformation; sound signal processing; guitar tuner, guitar metronome.*

I. INTRODUCTION

The software product includes a wide range of functionality that will be useful to large users. The main functionality of the program is the analysis of the sound flow, then its processing using the algorithm "Fast Fourier Transform". As a result, the device screen displays the frequency in Hertz, which corresponds to the sound stream, the number and display of strings in the guitar, which is responsible for the frequency and distance between frequency streams and the frequency of the required strings, converted into musical tones.

The program also implements an additional module to display the main guitar chords. The user can view them in various combinations with a specially created graphic element and listen to the sound with the speakers.

Another equally important module is the guitar metronome. A metronome is a device or program that plays sounds at a set speed and is created to help musicians at the right tempo. While playing any musical instrument, it is very easy to lose your sense of rhythm or apply something at the wrong speed. The user, if he arranges money on a guitar or other musical instrument, can set the tempo when it is convenient to play a particular melody. The metronome created in the program programs the rhythm, imitating the game of a real drum beat. Most modern musical works are created on the basis of 4 main acad. Each chord corresponds to a note on the bass guitar. The then module allows you to assign responses to the sounds of the

TABLE OF CONTENTS

International Competition of Student Scientific Works

BLACK SEA SCIENCE 2021

Information Technology, Automation and Robotics

Proceedings

Odessa National Academy of Food Technologies

The collection includes student works of the participants of the competition, which were not included in the number of prize-winners. The texts of the competitive works are published in the form in which they were submitted by the authors. The authors of the articles are responsible for the content and form of submission of the material.

Responsible for the issue: Sergii Kotlyk

Computer typesetting and layout: Oksana Sokolova

Odessa 2021