Recommender Systems Synthesis and Analysis for Goods and Services Sale with Cold Start Problem

Mykola Baran
Information Systems and Networks Department
Lviv Polytechnic National University
Lviv, Ukraine
mykola.baran.mnitm.2021@lpnu.ua

Victoria Vysotska[1,2]
[1]*Lviv Polytechnic National University,*
Lviv, Ukraine
[2]*Osnabrück University,*
Osnabrück, Germany
victoria.a.vysotska@lpnu.ua

Vasyl Andrunyk
Information Systems and Networks Department
Lviv Polytechnic National University
Lviv, Ukraine
Vasyl.A.Andrunyk@lpnu.ua

Lyubomyr Chyrun
Applied Mathematics Department
Ivan Franko National University of Lviv
Lviv, Ukraine
Lyubomyr.Chyrun@lnu.edu.ua

Valentyna Panasyuk
Accounting and Taxation Department
West Ukrainian National University
Ternopil, Ukraine
v.panasiuk@edu.wunu.ua

Tetiana Nazarova
Department of Accounting and Finance
National Technical University "Kharkiv Polytechnic Institute"
Kharkiv, Ukraine
taniya2017@ukr.net

Abstract—**The article develops a general approach to solving the problem of cold start with the possibility of flexible operation of the algorithm. An analysis of the features of the construction of a recommender system was carried out, as a result of which it was found that today there are a number of problems, the implementation of which requires taking into account many parameters depending on the specifics of the subject area, which served as an incentive for further analysis. It is noted that the process of collecting user data is a rather complex and time-consuming process that has several implementation principles. This is due to the fact that not every user provides sufficient information for further work, which in fact creates further complications due to the insufficient amount of information. One of the ways to solve this problem is to apply intelligent methods to construction, namely, machine learning methods. The main aspects of algorithms and methods of their improvement are briefly described. The hybrid method implementation for a system construction, as well as its performance testing in comparison with the classical k-means algorithm, is carried out. Scalability and ways to improve are also taken into account during implementation.**

Keywords— recommender system, Neural collaborative filtering, k-means, cold start, machine learning

I. INTRODUCTION

Very often, when building recommendation systems of certain content (goods and services), they are built on the basis of a simplified approach, such as, for example, recommending products only based on the user's previous preferences, or recommending based on the preferences of other users who may have completely different interests than the target user. It is logical that very often with this approach, the quality of recommendations is extremely low, since, for example, the user may not be interested in recommending a similar product again, or with a large list of goods or services in different categories, what one type of user likes may to be completely inappropriate for the second type. In turn, this approach not only reduces the efficiency of the system, but also reduces the interest of users in the system, as a result, there is a decrease in the audience and a loss of the reputation of the system and, in some cases, of the company. there is a need to apply new approaches for more efficient operation of the system and, accordingly, to meet the needs of the target audience.

To solve this task, recommender systems are increasingly used, based on recommendations, which build a rating list of objects according to certain criteria, and with the help of which each user can give an appropriate preference to a product or service. The advantage of these systems is the involvement of artificial intelligence and machine learning tools, which allow for a much more efficient selection of both targeted goods or services for users, as well as related to them [1, 2]. At the moment, two clear types of intelligent systems for providing information based on recommendations can be generally distinguished. The first principle is that the system is based on collaborative filtering, that is, in order to make a certain recommendation to the user, the system reviews the available information on how other users, whose characteristics are similar to the target, evaluated the potential object. After that, having received all the necessary information, the system can predict how highly the target user will rate a specific object (goods or service) and offer it.

Another type of data system are content filter systems. For these systems, a necessary condition is the presence of a database, which should contain metrics for all available objects, and accordingly update them as needed. After several actions of the user on the site, the system is already able to determine which approximate type, or according to which feature the objects are liked by the user. Accordingly, in the presence of pre-existing metrics, the system can select new objects for recommendation to the user, which will be similar to the previously viewed ones. It is worth noting that a significant drawback of such systems is that there is a need to build a fairly large database with metrics, without which the new system will not be able to immediately start providing recommendations to the user, since there will not be enough initial information [3]. In addition, the very process of building the metric can also become a problem. Based on all of the above, we gradually came to the realization that despite certain advantages of these construction approaches, the system may also have a number of disadvantages that should be taken into account when planning its implementation [4].

The problem of "cold start". One of the classic problems. It occurs when not enough data has been accumulated for the algorithm to work. This is a fairly typical situation for a new or unpopular object, or a site in general, which has been evaluated by a small number of users, or, conversely, for an extraordinary user whose preferences are very different from the average user [5]. In such cases, unless other algorithms are used, the ratings have to be adjusted artificially. For example, the rating for an object is calculated not as an average by position, but as a smoothed average. That is, with a small number of ratings, the rating will approach some "safe

2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT) | 979-8-3503-6046-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/CSIT61576.2023.10324089

average", but when enough real ratings are collected, artificial averaging is disabled.

The problem of "bias". Another classic problem is that of bias. It occurs when algorithms are not precisely configured, embedded in some stereotypes, and even user actions can affect the current analysis and ranking of information. In addition, very often the problem may arise from the user's side, in the case when the algorithm does not take into account situations in which the user limits himself in categories, but because of this he may miss certain information that is significant for him, since the algorithm clearly follows his implicit data (up to for example, demographic) and its explicit instructions (for example, filters) [6].

II. Related Works

There are quite a lot of works devoted to the development and research of recommender systems based on recommendations, methods of construction and their problems both in the international space and in the national segment. In the studies [1-3], the main types of recommender systems on the Internet, based on content and collaborative filtering methods, as well as their problems, among which the most common problems of recommender systems can be attributed to the problem of cold start and filter bubbles [1-3]. This problem is not new, but it has many solutions. As the authors of the study write, one of the ways to increase the accuracy of recommender systems and solve the problems of cold start and filter bubbles is to use the context in which user preferences are fixed and the context in which recommendations are formed [4-6]. In this case, the context means the use of the date and time of the recommendations, the devices from which the user visited the website, his demographic data, geolocation, etc. Based on the development of progress, as well as the new functions brought to us by ES6+ and HTML5, getting the necessary data will not be a big problem. However, the only obstacle that may arise is the privacy policy of browsers, and therefore the user may not provide some of the data we need, or this option may be disabled by default. Another landmark study is the articles [3-9], in which it is emphasized that among the algorithms, the algorithms of collaborative filtering and content-based filtering (CF and CBF) remain the most common), as well as their combination. Mostly, these algorithms work on the basis of meta-information about news. However, the work notes their insufficient quality in relation directly to news recommendations. Recently, researchers and developers have increasingly paid attention to "surrounding context" and deep learning algorithms to improve the quality of recommendations. The authors include the use of the following algorithms as "new" approaches in the development of NRS: Matrix factorization and Integral matrix factorization; Tensor factorization; Probabilistic matrix factorization; Bayesian personalized ranking;General linear modeling. As well as their modifications based on neural networks using deep learning, which is especially gaining popularity in recent years.

In the paper [6-12] the authors reviewed previous research on tag-aware recommender systems to improve the traditional recommender methods performance. In the course of the study, it was determined that user-specified tags usually suffer from many problems, such as ambiguity, redundancy, and sparseness. To solve these problems, the authors proposed one of the new recommendation approaches based on deep neural networks. Experimental results demonstrate the usefulness of the proposed approach and demonstrate its better performance compared to classical algorithms, which confirms the effectiveness of the combination of classical construction approaches and approaches based on neural networks and deep learning. In all the above-mentioned works, the needs to solve the problems of cold start and filter bubbles stand out the most. In addition, it is worth adding another common problem that occurs both during the system start-up and in its further operation, namely the problem of engaging the context and obtaining user data without his knowledge, which may in turn be a false representation of him. Accordingly, the relevance of developing a system that will take into account all existing problems and is built on the principle of hybridity, which in turn provides scalability and reliability for the system, appears in all works.

The purpose of the study is to determine the main methods of constructing algorithms with a solution to the cold start problem. The conducted research will provide means for building a recommender system taking into account the problem of cold start. To achieve the goal, it is necessary to solve the following main tasks: to determine the main approaches to solving the problem, to choose the most appropriate implementation solution based on them, to verify the correctness of the statements through a detailed analysis, and to check the feasibility of further improvement of the algorithm.

III. Experiments, Results and Discussions

In order to successfully develop and implement an information system to solve the task, it is advisable to carefully formulate the tasks and their formulation. Also, it is worth conducting an analysis and predicting possible difficulties in implementation. Accordingly, in order to effectively design the system taking into account all requirements and minimum costs, a set of appropriate diagrams was constructed. Fig. 1 shows a representation of a class diagram consisting of seven main classes:

- the "LogForm" class describes a module that provides an opportunity to register or authorize a user;
- the "PersonalUserDataForm" class describes a module that provides the ability to edit user data, as well as perform manipulations to remove or add new user layouts;
- the "TrackingManager" class describes a module that scans and analyzes user actions, system status, and collects data for further processing, structuring and transmission;
- the "DataBase" class describes the database, which contains all data about users, their activities, as well as about the list of products, and information about interaction with them;
- the "UserList" class describes a local database containing the latest user analysis, which will be transferred to the main database in the future;
- the "AI" class describes the main logical module of the system, which performs calculations for predicting the feasibility of product recommendations, their filtering, sorting, and is also responsible for user clustering.

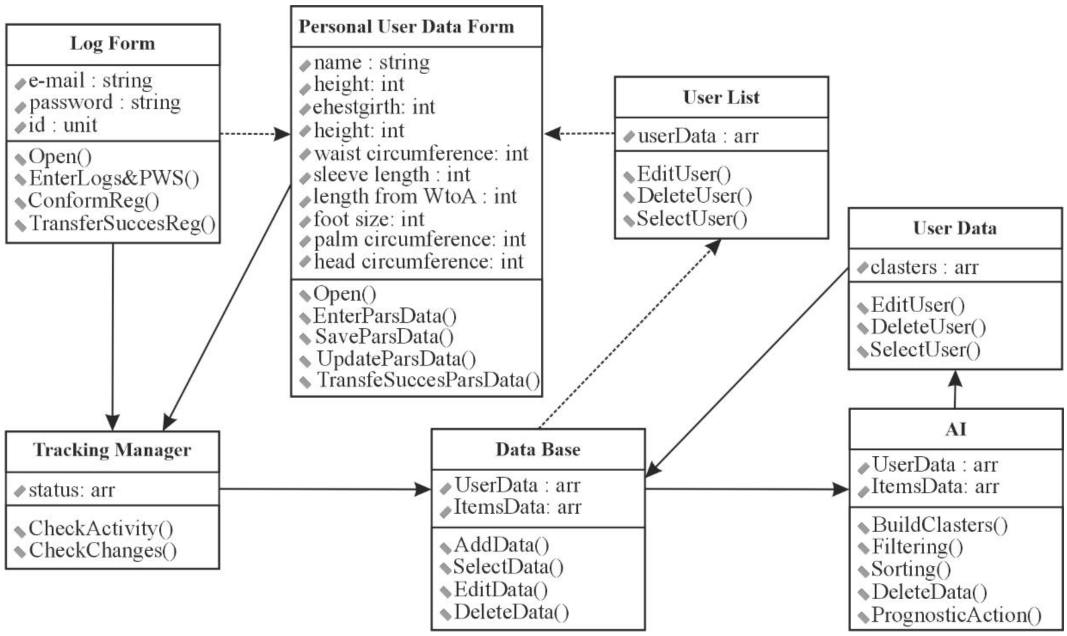

Fig. 1. Class diagram of the recommender systems of goods and services sale with cold start problems

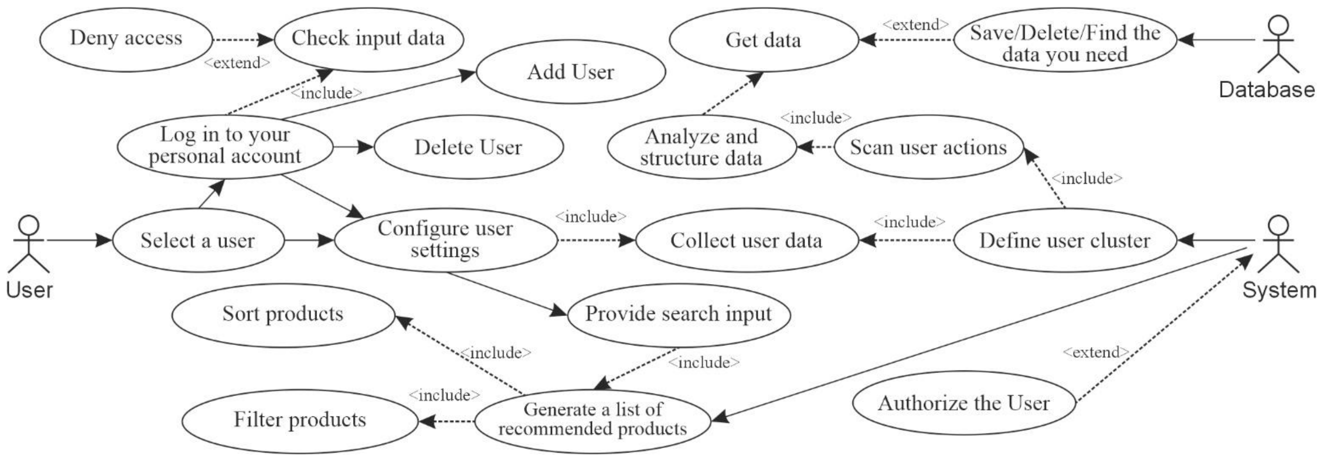

Fig. 2. Diagram of use cases of the recommender systems of goods and services sale with cold start problems

Fig. 2 shows a diagram of use cases for an intelligent product selection information system. This diagram shows three actors, that is, the main entities:

- A user who wants to choose one or more products and needs a specific recommendation.
- The system that authorizes the user collects available data, determines the user's cluster, checks the received data, scans, analyzes and displays the received results in the form of recommendations.

A database that contains all available data, and which has the ability to perform data manipulations, such as: Deleting, finding and editing data.

The user of the system has the following usage options: "Select a user", "Log in to the personal account", "Add a user", "Delete a user", "Configure user parameters", "Provide search input data". Accordingly, the use case "Log in to the personal account" is related by the <<include>> relation to the use case "Check input data", which in turn is related to the use case "Deny access" by the <<extend>> relation. The Configure User Settings use case is linked by an <<include>> relationship to the Collect User Data use case. In turn, the System includes such options of use as: "Generate a list of recommended products", "Define a user cluster", as well as "Authorize a user", which is connected by the relation <<extend>>. The use case "Generate list of recommended products" is related to the use cases "Sort products" and "Filter products" using <<include>> relationships. The "Determine User Cluster" use case is related to the "Collect User Data" and "Scan User Actions" use cases through <<include>> relationships. In turn, these use cases are linked by the <<include>> relation to the "Analyze and structure data" use case. Use case "Analyze and structure data" is related by <<include>> to use case "Get data", which in turn is related to use case "Save/Delete/Find required data" by <<extend>> . The entity "Database" has only one use case available, namely "Save/Delete/Find required data".

Fig. 3 shows the sequence diagram for the developed recommender system. When the user's system starts, the main service loads the GUI and the authorization form. Once the user enters the credentials for authorization, it will be sent to the server and checked against the database, after which access will be allowed or denied. In the next stage, the user will be given the choice of sub-users, their data editing, as well as the creation of a new sub-user.

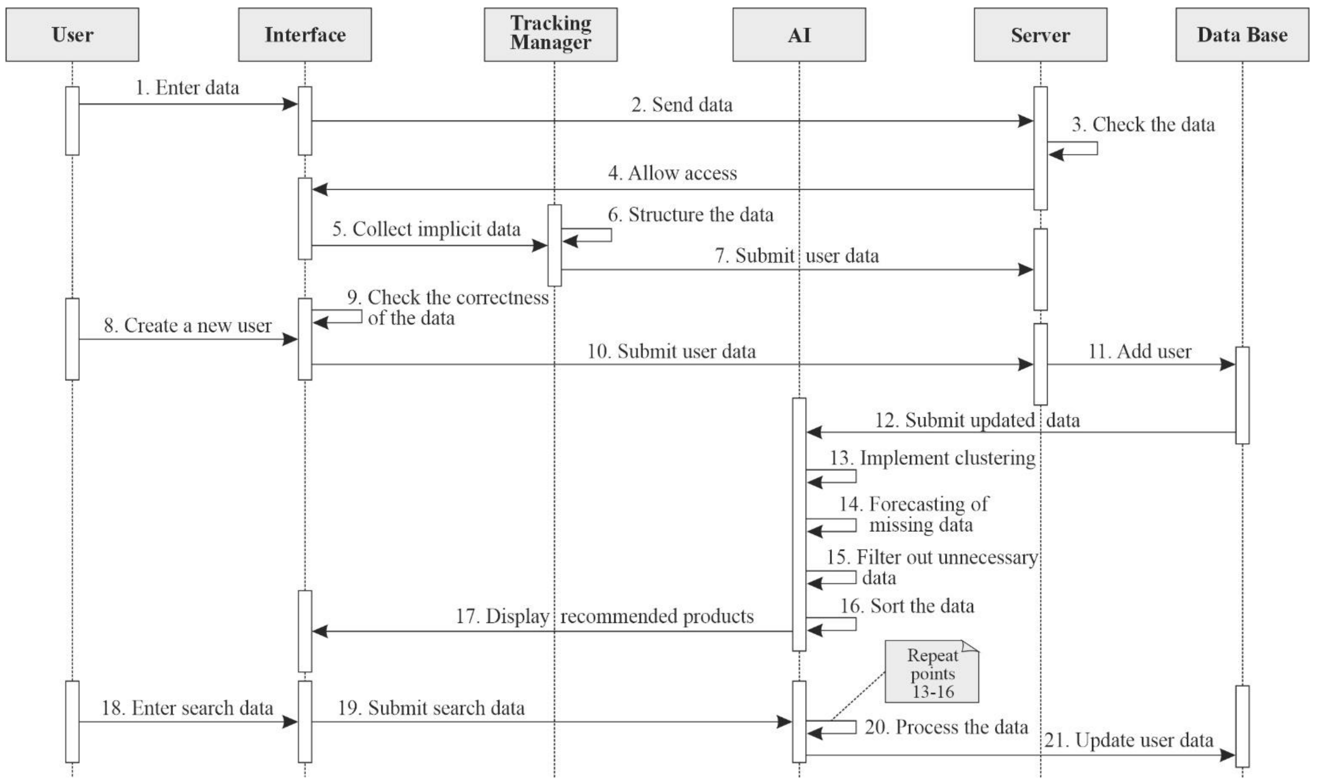

Fig. 3. Sequence diagram of the recommender systems of goods and services sale with cold start problems

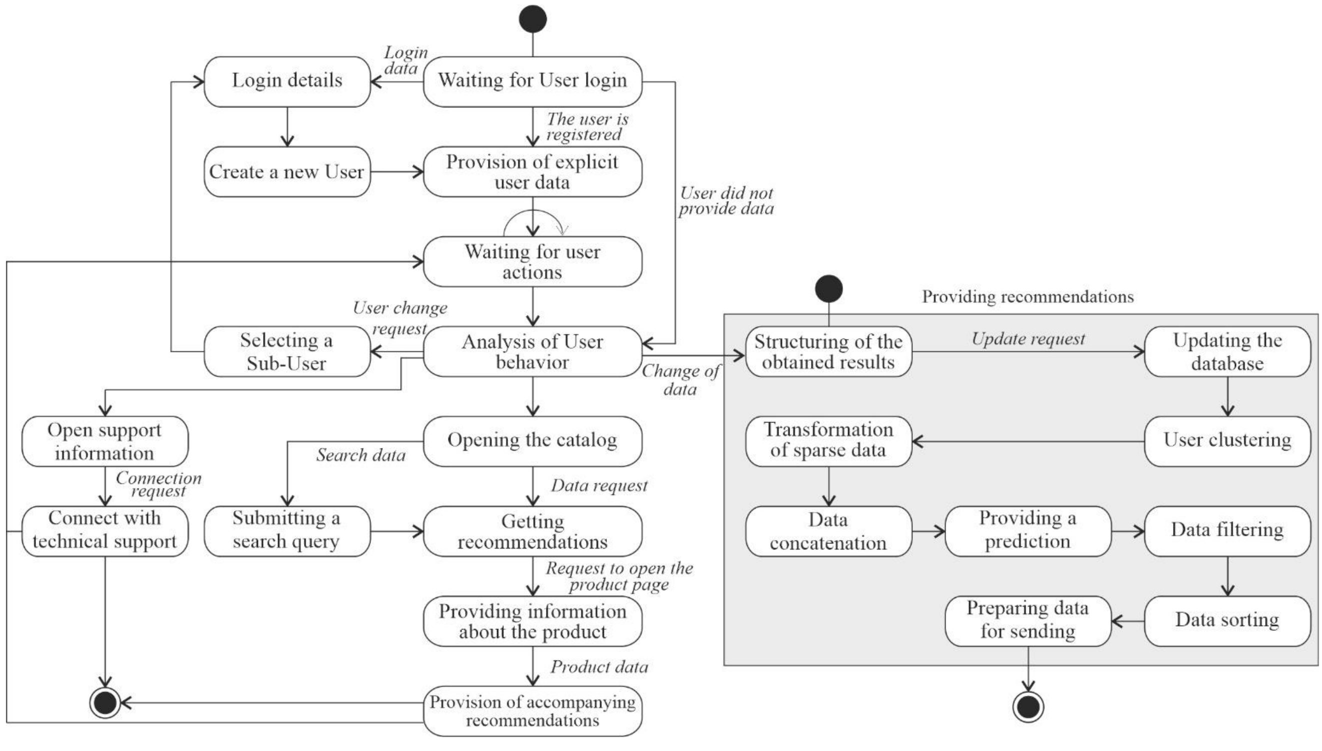

Fig. 4. Diagram of state transitions of the recommender systems of goods and services sale with cold start problems

When creating, the user enters all the necessary data, which is checked on the client side, after which it is sent to the server, where it is further loaded into the database. After that, the system will analyze the received explicit and implicit data, and form clusters of users taking into account the latest received data. When all the necessary information about the user is generated, the system will select recommendations and display them in the interface in a user-friendly form. If additional data is added, such as a search query, new activity or editing of explicit user data, the clustering and recommendation prediction process will be repeated. In fig. 4, in accordance with the tasks of system analysis, shows the implemented diagram of state transitions for our developing recommender system for the selection of equipment for power structures. In order to better understand the operation of the system, the process of providing recommendations was detailed by describing the existing conditions that occur when it is applied in the system. In fig.

Figure 5 presents an activity diagram showing the main states of the system. The diagram itself includes the relationship between the various components of the system and the processes performed in the various threads. The initial state is the "Waiting for user login" action. In addition, the diagram contains several alternative transitions in which all possible cases are described. The final actions of the system are the display of data on the graphical interface and in a user-friendly form. Fig. 6 provides a diagram of the components for the information system for the selection of equipment for power structures. It shows the main program and its user interface, the database, as well as the main components and modules of the system.

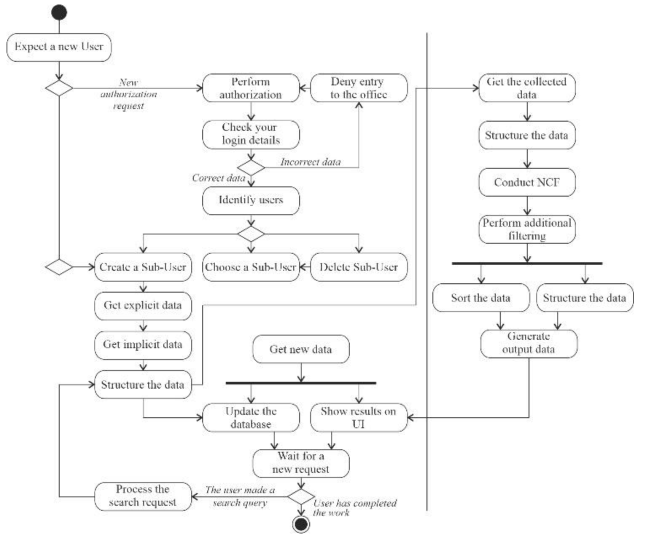

Fig. 5. Activity diagram of the recommender systems of goods and services sale with cold start problems

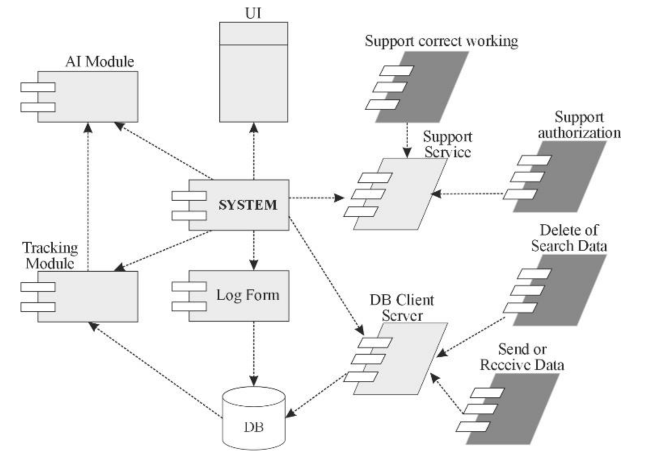

Fig. 6. Diagram of components of the recommender systems of goods and services sale with cold start problems

In the context of this problem, the implementation of a recommender system taking into account algorithms that can overcome the problem of a cold start is a rather important process. In general, the basic solution to the problem of a cold start for users is solved based on the minimum data about the user that can somehow be obtained. The simplest data among these are demographic. Ideally, in order to receive such data, the user is offered to go through registration, during which he can specify certain information about himself, however, in addition to explicitly receiving data about the user, it is also possible to receive it using the API of social networks, where most users indicate their age, social status, level education and others. Other data for a certain class of recommender systems are the user's physical data, such as height, weight, etc. There are two main approaches to determine the recommendations of certain categories of users based on the received data:

- *expertly*- is carried out by determining certain generally accepted patterns by involving a certain expert in this area. In this way, the expert himself determines what to show to each of the user categories at the beginning of the system's operation when it is in a cold start. An obvious disadvantage of this approach is the need for the work of an expert and his certain subjectivity.
- *automatically*- categories are determined by applying algorithms that can identify clusters of users with similar interests, after which recommendations are built on the basis of ratings given by users from a common category.

Since we are solving the cold start problem for an recommender system, the second approach is the most appropriate for further analysis. To build categories of users, it is most expedient to use clustering methods, where users will act as clustering objects, and features will be the necessary data about the user. The most appropriate among these methods is the k-means method, because in this case each cluster is determined by its center point, and thus is well interpreted [7]. Since when providing recommendations, the system must take into account not only primary data about the user, but also his already existing ratings, or ratings of other users, which may also be missing, the Neural collaborative filtering (NCF) algorithm, which is based on collaborative data, comes in handy implicit feedback [8, 9]. Input data can be different, but initially only collaborative data encoded according to the principle of one-hot encoding is used. A significant advantage of this approach is that in the case of a cold start, user and content metadata can be submitted to the input. Behind the input layer is a fully connected one, which is responsible for transforming a sparse one-hot representation into dense embeddings. After receiving, the attachments are concatenated, and the subsequent architecture actually has a name NCF and is responsible for converting nestings of user-object pairs directly into predictions. In this case, the standard rms error may not correspond to the kind of real implicit feedback data represented by the values 0 and 1, so a probabilistic approach to NCF training is used. That is, if we consider the predicted number as the probability of a value of 1 (the relevance of the object to the user) and limit the output of the network to the range [0, 1], then, accordingly, we can use the loss function as on Fig. 7.

$$
\begin{aligned}
L &= -\sum_{(u,i)\in\mathcal{Y}} \log \hat{y}_{ui} - \sum_{(u,j)\in\mathcal{Y}^-} \log(1-\hat{y}_{uj}) \\
&= -\sum_{(u,i)\in\mathcal{Y}\cup\mathcal{Y}^-} y_{ui}\log \hat{y}_{ui} + (1-y_{ui})\log(1-\hat{y}_{ui}).
\end{aligned}
$$

where Y represents a set of negative examples, as a certain subset of interactions that are not observed (user-object pairs for which there is no data). Negative examples are sampled uniformly, but approaches that consider popularity can be used to improve performance. Visually, this geometric meaning of singular expansion can be represented as follows as on Fig. 7. In this way (Fig 7), the algorithm will work based on both the similarity of the primary data about the user, and taking into account explicit and predicted user ratings [10]. After comparing the efficiency when applying the k-means method with and without the NCF algorithm, we can observe the following results on Fig. 8.

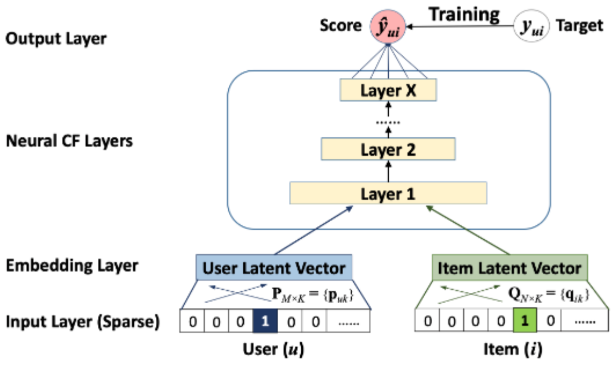

Fig. 7. Principle of operationNeural collaborative filtering (NCF)

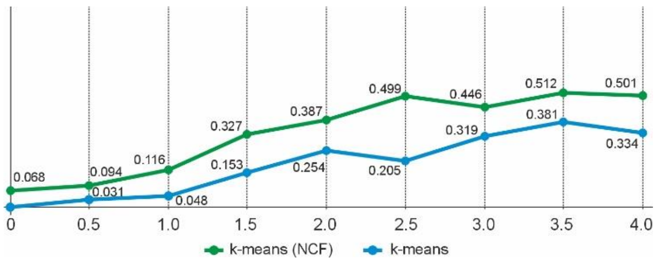

Fig. 8. Comparison of the k-means method with and without the NCF algorithm in the recommender systems of goods and services sale with cold start problems

These metrics reflect recommendations to clothing users with the ratio of the number of product transitions to the number of impressions. It is worth noting that at the same time the system was in the conditions of a cold start, in which all objects and users were considered new, and before the start of the algorithms, no information about the history of previous displays was used. Of course, improving the recommendations in this case can already be done in many ways. The main nuance here is the maximum optimization of the algorithm, and the elimination of gaps and incompatibilities. Since in our case we were based on the formation of clusters from users, in this case it will be appropriate to improve the recommendations by applying group recommendations, the name of which speaks for itself: we select for a new user such recommendations that are liked by the majority of users of his demographic category. There are a number of different strategies for aggregating ratings from different users into a group recommendation. The advantage of this approach is the simplicity of implementation and the absence of the need to change existing algorithms. It is worth noting that filter bots and group recommendations do not conflict with each other, and accordingly, this factor makes it possible to use them together. In this case, the initial ratings of the filter bots are taken as group ratings [11-12]. Taking into account all of the above, the final version of the algorithm concept will have the following form on Fig. 9. The principle of these recommendations is to select for a new user such recommendations that are liked by the majority of users of his category (cluster). It is worth noting that this approach to some extent resembles the user-based principle, which is based on comparing users and takes into account the similarity of a given user to other users involved in the system.

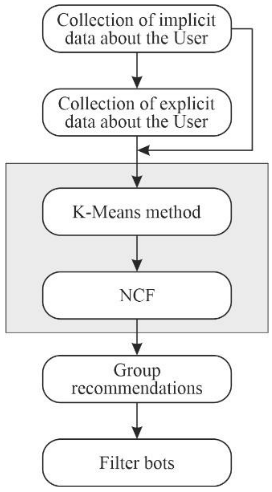

Fig. 9. Concept of the algorithm fot recommendations formation

IV. Conclusion

An alternative to this approach is the use of filterbots, which generate initial ratings for a new user. That is, during registration, filter bots automatically generate several basic ratings for the user based on his data that were collected explicitly or implicitly, and which then use already existing collaborative filtering algorithms on a cold start. Further research will be aimed at improving the algorithm and creating a recommender system with the implementation of approaches obtained as a result of the research.

References

[1] S. Natarajan, et al. "Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data." Expert Systems with Applications 149 (2020): 113248.

[2] K. Pliakos, et al. "Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems." Computers & Education 137 (2019): 91-103.

[3] Y. Ma, et al. "A deep neural network with multiplex interactions for cold-start service recommendation." IEEE Transactions on Engineering Management 68.1 (2020): 105-119.

[4] J. Bobadilla, et al. "A Collaborative Filtering Approach to Mitigate the New User Cold Start Problem". Knowledge-based systems. 26 (2012): 225-238. doi : 10.1016/j.knosys.2011.07.021

[5] B. Lika, et al. "Solution of the cold start problem in recommender systems." Expert systems with applications. 41 (2014): 2065-2073.

[6] Y. Deldjoo, et al. "Movie genome: alleviating new item cold start in movie recommendation." User Modeling and User-Adapted Interaction 29 (2019): 291-343.

[7] Q. Zhang, et al. "Artificial intelligence in recommender systems." Complex & Intelligent Systems 7 (2021): 439-457.

[8] L. Huang, et al. "A deep reinforcement learning based long-term recommender system"Knowledge-Based Systems 213(2021):106706.

[9] J. Herce-Zelaya, et al. "New technique to alleviate cold start problem in recommender systems using information from social media and random decision forests." Information Sciences 536 (2020):156-170.

[10] Z. Ye, et al. "Cold start to improve market thickness on online advertising platforms: Data-driven algorithms and field experiments." Management Science 69.7 (2023): 3838-3860.

[11] I. Balush, V. Vysotska, S. Albota, “Recommendation System Development Based on Intelligent Search, NLP and Machine Learning Methods,” CEUR Workshop Proceedings, 2917, 2021, pp. 584-617.

[12] S. Khanal, et al. "A systematic review: machine learning based recommendation systems for e-learning." Education and Information Technologies 25 (2020): 2635-2664.