Comparative Analysis of Basic Approaches to Implementing Model-Based Recommendation Systems Based on Implicit Economic Information

Yurii Kryvenchuk, Viktoriia Lakiza, Yuliia Bidak and Iryna Myskiv

Lviv Polytechnic National University, Profesorska Street 1, Lviv, 79013, Ukraine

Abstract

The paper considers ways to solve the problem of Internet congestion.Analogs of recommendation systems of different researchers are also given. The main algorithms in recommendation systems are analyzed: Content based, demographic based, Coloborative filter. Two types of data are considered, which help to form an overall assessment in the recommendation system. The main problems that shape the work with recommendation systems are considered.The tasks of recommendation systems are analyzed in detail. The paper provides a step-by-step creation of a recommendation system and identifies the main requirements that it must meet.The study presents a similarity matrix, which is calculated from the entire recommendation vector. The personalization of the recommendation is also calculated.The matrix factorization method is analyzed (Matrixfactorization). The evaluation that follows from the user profile is considered.In the work, to get results on the proposed models, offers its own web service for finding movies, where the user can search for movies, as well as view detailed information about them or the movie rating. Recommendations in this system are based on implicit feedback, and it is possible to receive information about the user's id to make personalized recommendations.The implemented methods of recommendations are also analyzed:Linear Regression Prediction, Content-Based Prediction, Collaborative-Filtering Prediction User-Based, Collaborative-Filtering Prediction Item – Based.

Keywords 1

Machine Learning, Artificial Intelligence, neural network, intelligent technologies, sum of error squares, recommendation system, content-based approach.

1. Introduction

Today, the problem of Internet congestion remains open. The amount of information on the Internet is growing exponentially every day [1, 5, 18]. Recommendation systems are a relatively young field. It all started in 2006 when Netflix launched the Netflix Prize data analysis competition. Around the same time, the annual RecSys conference on referral systems began, which is still held today [3, 7].

The study aims to describe models where the components for translating the characteristics of user behaviour are his assessments, which are used for his content recommendations [6, 2]. If the problem is attributed to the difficulties of classification or regression, the list of required algorithms is quite comprehensive. Therefore, the study should pay attention to the work of several algorithms based on accurate data [4, 8].

Before starting the study, a list of characteristics that can describe any recommendation system is given.

Information Technology and Implementation (IT&I-2021), December 01–03, 2021, Kyiv, Ukraine
EMAIL: yurkokryvenchuk@gmail.com (Yu. Kryvenchuk); viktoriia.v.lakiza@lpnu.ua (V. Lakiza); yuliia.bidak.knm.2019@lpnu.ua (Yu.Bidak); myskiviryna@i.ua (I. Myskiv); inem.news@gmail.com (Yu. Malynovskyy)
ORCID: 0000-0002-2504-5833 (Yu. Kryvenchuk); 0000-0002-6764-8536 (V. Lakiza); 0000-0002-9780-1546 (Yu.Bidak); 0000-0002-3761-2276 (I. Myskiv); 0000-0002-7139-5623 (Yu. Malynovskyy)

- Subject of recommendations - what is recommended. It can be anything: movies, music, products, news, articles, books, products, videos, people and more.
- Purpose of recommendations - the navigator is recommended. They are gathering, providing information, training, meeting new people.
- Recommendation context - what the user is doing at the moment. You are browsing products, listening to music, communicating with people.
- Source of recommendation - who recommends. Audience-like users, experts.
- Degree of personalization. Non-personal recommendations - when you are recommended all the same as other users. They allow targeting by region or time but do not consider their preferences. Additional enhancements include the number of recommendations for your current session. You have reviewed several products and recommended similar products for you. Personal recommendations contain all available information about customers, including the history of their purchases.
- Transparency. People trust recommendations more when they understand what they are based on. So there is less chance of coming across a system that recommends offering goods or services.
- Recommendation format. This can be included in a window, a sorted list found in certain parts of the site, a bar that opens the screen, or something else.
- Algorithms. Despite many available algorithms, they all come down to a few basic approaches. The most classic is:
  - Summary (non-personal);
  - Based on content (models based on the product description);
  - Collaborative (collaborative filtering);
  - Matrix factorization (methods based on matrix schedules).

To define recommendations, standard filtering systems must correlate two fundamentally different objects: elements and users. Therefore, the aim of this study is to compare two main approaches, which are the two main methods of joint filtering: the neighborhood approach and the model of hidden factors. Neighbourhood methods focus on relationships between objects or between users.

The relevance of this study is the process of modeling user preferences based on assessments of similar aspects of the same user [9, 12].

Hidden factor models, such as matrix factorization (SVD), contain an alternative approach, turning both elements and users into the same confidential factor space. Latent space explains ratings by characterizing products and users by factors that automatically follow from user feedback.

Matrix decomposition methods [8, 10] combine ease of implementation with relatively high accuracy. This made them the best technique for solving the most extensive public data set - Netflix data. Hidden factor models (LFMs) are suitable for co-filtering with the holistic purpose of identifying latent features that explain the observed estimates; examples include pLSA, neural networks, latent Dirichlet distribution, and models induced by factoring the evaluation matrix of user elements (also known as SVD models) [11, 15]. Recently, models based on matrix extensions have gained popularity due to their attractive accuracy and scalability.

2. Materials and methods

When searching for information, matrix decomposition methods are used to identify hidden semantic factors. However, its application to precise estimates in co-filtering is complex due to the large proportion of missing values [13, 16]. The usual matrix decomposition method is not determined when knowledge of the matrix is incomplete. Moreover, the careless attitude towards only a few well-known records is prone to excessive placement. Previous work has relied on imputation, filling in the gaps and making the rating matrix dense [14, 17]. However, the hint can be very expensive, as it significantly increases the amount of data. In addition, data can be distorted considerably due to false imputations. Thus, newer works suggest directly modelling only the observed ratings, avoiding adjustable model branches.

2.1. Tasks of the recommendation system

The task of the recommendation system is to inform the user about the product that may interest him most at a particular time. The customer receives recommendations about the product he needs, and the service earns, depending on the business model, recommendation systems can be profitable in different ways [7, 12]. The first option is the direct sale of goods. The following can affect the number of users and in turn the revenue from advertising and so on.

In the previous section, the main principles, problems and objectives of recommendation systems were discussed. This should focus on preparing for practical implementation [10, 16]. The first step is to define the requirements that the recommendation system must meet.

1. Coverage. Coverage is the percentage of test items that a test set recommendation system may recommend..
2. Personalization. Personalization shows how many identical things the recommendation system shows to different users. Personalization is calculated in Table 1.

Table 1
Requirement for the recommendation system - personalization

	A	B	C	D	X	Z
0	1	1	1	1	0	0
1	1	1	1	0	1	0
2	1	1	1	0	0	1

Binary variables define two states (1 – the subject was recommended to the user. 0 – was not). The next step is to calculate the similarity matrix for users in Table 2.

Table 2
Similarity matrix for users

	0	1	2
0	1	0,75	0,75
1	0,75	1	0,75
2	0,75	0,75	1

The similarity matrix is calculated from the whole recommendation vector. Personalization= 1 – 0,75 = 0,25. The next step is to calculate the average of the upper triangle and subtract from the unit. A high score means that the model provides highly personalized recommendations.

3. Estimation of similarity

The similarity assessment determines how much similar items are advised to the user. This uses feature features (such as genres in movies) to calculate similarity. Let's look at an example of Figure 1.

```
example_predictions = [
    [3, 7, 5, 9],
    [9, 6, 12, 623],
    [7, 894, 6, 623]
]
```

Figure 1: Example of defining movie id

In the Table 3 defined features about the object - the film, which are determined by the user using the recommendation system. So, in Table 3 genres for recommended movies for the first user. In Figure 2 shows an assessment of similar films received by the user of the recommendation system.

The higher the rating, the more similar movies the user will receive. Therefore, the metrics that determine the quality of the recommendation system should be considered. Recall and Precision at *k*. This metric was commonly used in binary classification algorithms. Now this is one of the effective

ways to determine the quality of the recommendation system. In this case, it is necessary to say whether the recommendation interested the user or not. A rating of 1-5 is usually used for this.

Table 3
Representation of features on the film using the recommendation system

movieId	Action	Comedy	Romance
3	0	1	0
7	0	1	0
5	0	1	0
9	1	0	0

```
recmetrics.intra_list_similarity(example_predictions, feature_df)
```

```
0.27777777777777773
```

Figure 2: Score for films offered by the recommendation system

To translate the rating into the binary system, suffice it to say that all values above a certain level should be considered positive. For example, take the value of 3.5 (these can be absolute values depending on the problem). The next step is to determine the 'k'. Since recommendation systems usually return a list of recommended products, only the first 'k' should be considered..

This metric shows the percentage of recommendations from the top 'k' items that were correct and relevant to the user.

3. Experiments

The main part of working with recommendation systems is data. To review the algorithms, use the Deskdrop dataset, which includes 12-month records from CI & T's Internal Communication platform (DeskDrop). It includes information about 73 thousand users who interacted with 3000 articles distributed on the platform and includes 2 files: shared_articles.csv; users_interactions.csv.

Their structure should be considered for analysis. For the Shared_articles.csv file, which contains information about common files on the platform, where each article has its original url, title, content as plain text, language, and information about the user who published the article. Also, each time stamp has two possible events: Content is distributed and available to users; Content has been deleted and is not available to users. In the Table 4 shows data with timestamp, type of interaction, movie ID, user ID, user session ID.

Table 4
Representation of file features Shared_articles.csv

	timestamp	eventType	contentId	authorPersonId	authorSessionId	Author UserAgent
1	1459411468	ContentShared	-4011547382	38732923901	243872438932	Nan
2	1459411469	ContentShared	-3834093833	37239832892	894173187267	Nan
3	1459411470	ContentShared	-3736267384	56712348938	-21378327824	Nan
4	1459411471	ContentShared	-3284737777	23923802332	327632872398	Nan
5	1459411471	ContentShared	-5671839300	23983298320	23932893232	Nan

The users_interaction.csvfile stores information about user interaction with articles. This dataset includes the following types of interactions: Views, Preferences, Comments, Tracking (user will be notified of new comments on this article), Saved (the user saved the article to return to it in the future). In the Table 5 users_interaction.csva dataset with a timestamp, type of interaction, movie ID, user ID, user session ID.

The next stage is the transformation of data, where for each type of interaction is given a certain weight (Figure 3), which will reflect the user's interest in a particular article.

Table 5
Representation of file featuresusers_interaction.csv

	timestamp	eventType	PersonId	SessionId	contentId	userAgent
1	12782187	View	-2383298233	1872414232	3223898921	Nan
2	14789898	Follow	83893722323	3213313132	2392841894	Mozilla
3	12873891	View	31839212323	2298283933	8023974873	Nan

```
event_type_strength = {
   'VIEW': 1.0,
   'LIKE': 2.0,
   'BOOKMARK': 2.5,
   'FOLLOW': 3.0,
   'COMMENT CREATED': 4.0,
}

interactions_df['eventStrength'] = interactions_df['eventType'].apply(lambda x: event_type_stre
ngth[x])
```

Figure 3: Giving certain weights for interaction

Also a common problem in referral systems is the cold start problem, so you should only work with users who have 5 or more interactions.. On the DeskDrop platform, the user can view articles several times and interact with them each time, which is why you should create a new column that will reflect the user's interaction with this article by summing up all types of interactions.

In the Table 6 presents data on user interaction with the recommendation system. Let's see what the columns with which the user interacts will look like.

Table 6
Representation of interaction between users and certain content

	personId	ContentId	EventStrength
0	-231789239	-89762372	1.00000
1	-998327887	-83478183	1.00000
2	-932834343	-23873277	3.16943

The following is a list of the most popular algorithms.

3.1. Overview of basic alorithms and models in recommendation systems

1. Model by popularity

The most common model because of its simplicity. This model is not personalized at all. It simply recommends to the user the most popular (with the highest rating) items or content. In general, it offers good recommendations that are liked and will be interesting to most.

It shows in the metric Recall @ 5, where the figures are about 24%, which means that 24 percent with which the user interacted, the system was able to predict the ranking in the top 5. And with Recall @ 10, the figures generally reach 37% (Figure 4).

2. Content based filtering model.

This model uses content attributes that can be recommended to the user of the article, similar to those with which he has already interacted. TF-IDF, a popular technique in search engines, is commonly used to work with text. This technique converts unstructured text into a vector, where each word is represented by a word and the position of that word in the vector. To prepare a user profile, take all the articles he interacted with and display the main words in them and multiply them by the weight of each article relative to the user (The more the user interacted with the article, the more important the keywords in it will be).

```
Evaluating Popularity recommendation model...
1139 users processed

Global metrics:
{'modelName': 'Popularity', 'recall@5': 0.2417540271030427, 'recall@10': 0.3729225262081309
3}
```

Figure 4: Metrics Recall @ 5 for the model by popularity

This method received a score of Recall @ 5 = 0.162 ~ 16.2 percent. Recall @ 10 = 0.261 ~ 26 percent (Figure 5). As you can see in Figure 6, this model, despite the fact that it is more difficult to implement showed worse results than a simpler model in popularity.

```
Evaluating Content-Based Filtering model...
1139 users processed

Global metrics:
{'modelName': 'Content-Based', 'recall@5': 0.16287394528253643, 'recall@10': 0.2614420864229
097}
```

Figure 5: Metrics met Recall @ 5 at Content based filtering model

3. Collaborative model.

This model is divided into two types:

- Memory-based - this model uses previous user interactions with articles to find a user with similar preferences and use it for recommendations in the future.
- Model baseduses different methods and models of machine learning (neural networks, Bayesian networks) to cluster users and find common preferences between them.

Next, you need to evaluate a system based on the Matrixfactorization model. In this case, in Figure 6 ratings for Recall @ 5 (33%) and for Recall @ 10 (46%).

```
Evaluating Collaborative Filtering (SVD Matrix Factorization) model...
1139 users processed

Global metrics:
{'modelName': 'Collaborative Filtering', 'recall@5': 0.33405778573254924, 'recall@10': 0.468
16670928151366}
```

Figure 6: Metric indicators Recall@5 при colloborative model

4. Hybrid model

The last and most progressive model, which combines the two previous models (colloborative and content-basedfiltering). This model showed the best results, namely Recall @ 5 = 34.2%, Recall @ 10 = 47.9% (Figure 7).

```
Evaluating Hybrid model...
1139 users processed

Global metrics:
{'modelName': 'Hybrid', 'recall@5': 0.34275121452313984, 'recall@10': 0.4796727179749425}
```

Figure 7: Metrics Recall @ 5 in the hybrid model

In the Table 7 shows the results of comparison of the main models in the recommendation systems.

Table 7
Comparison of basic models in recommendation systems

	recall@5	recall@10
Conten-based	0,16	0,26
Popularity	0,24	0,37
Colloborative filtering	0,33	0,46

After the results given in Table 7, it can be concluded that for further development of the system it is necessary to use a hybrid model for the best results. We should also give an example of a more modern method that has gained popularity, namely the factorization of the matrix (Matrixfactorization). To begin with, let's learn what factorization is. Factorization is the decomposition of a matrix into principal components. Take for example a table where the columns correspond to the names of the films, and the rows of user ratings for these films (Table 8).

Table 8
User ratings for specific movies

	Avenger	Thor	DeadPool	Avatar	Rocky	Titanic
Pumba	4	5	3	3	1	-
Henry	5	-	3	2	-	4
Jerry	1	2	2	-	4	2
Tom	3	4	-	2	4	1
Timon	4	2	3	5	3	-

If there is a dash at the place of evaluation, it means that the user has not watched this movie and the task is to predict his impressions after watching.

Accordingly, in Figure 9 the initial matrix is marked in blue, let's call it V, and the next two matrices, on which the initial matrix should be decomposed, are called W and H. Thus it is possible to deduce the general kind of expression: $V(m*n) = W(m*k)*H(k*n)$, where k – count of components. This means that when multiplying matrices W and H we obtain an approximate matrix V in which empty values will take on a certain meaning that will correspond to the predicted estimates of users for certain products. There are three main methods of decomposing matrices and their comparison is shown in Figure 8.

As can be seen from Figure 8 SVD and NNMF methods work best. The choice between them depends only on the data set, but they have one significant difference. When SVD works with a range of numbers from minus infinity to plus infinity, the result of the method can give the same range of numbers. And in the analysis, the NNMF method works only with positive numbers.

4. Work results

To get results on the proposed models, the work created a web service for finding movies, where the user can search for movies, as well as view detailed information about them, as well as the movie rating. Based on this data about the user's interaction with the site, you can create recommendations. MovieLensDataset was used to build the service. Recommendations should be based on implicit feedback. To do this, the client side collects information about user clicks, while recording information in the object, which consists of the name of the movie, the number of clicks on this movie, as well as its evaluation. After the user has watched several movies, the information is sent to the server where the object was used as test data (Figure 9). As can be seen from Figure 10, the server also receives user id information to make personalized recommendations. As initial data on object the client with 10 films which can be interesting to the user is sent and we receive the list of recommendations. In general, the system implements several methods of recommendations, so you need to call a certain, of your choice, to get results. The following methods of recommendations are implemented in the proposed system: Linear Regression Prediction, Content-Based Prediction, Collaborative-Filtering Prediction User-Based, Collaborative-Filtering Prediction Item –Based.

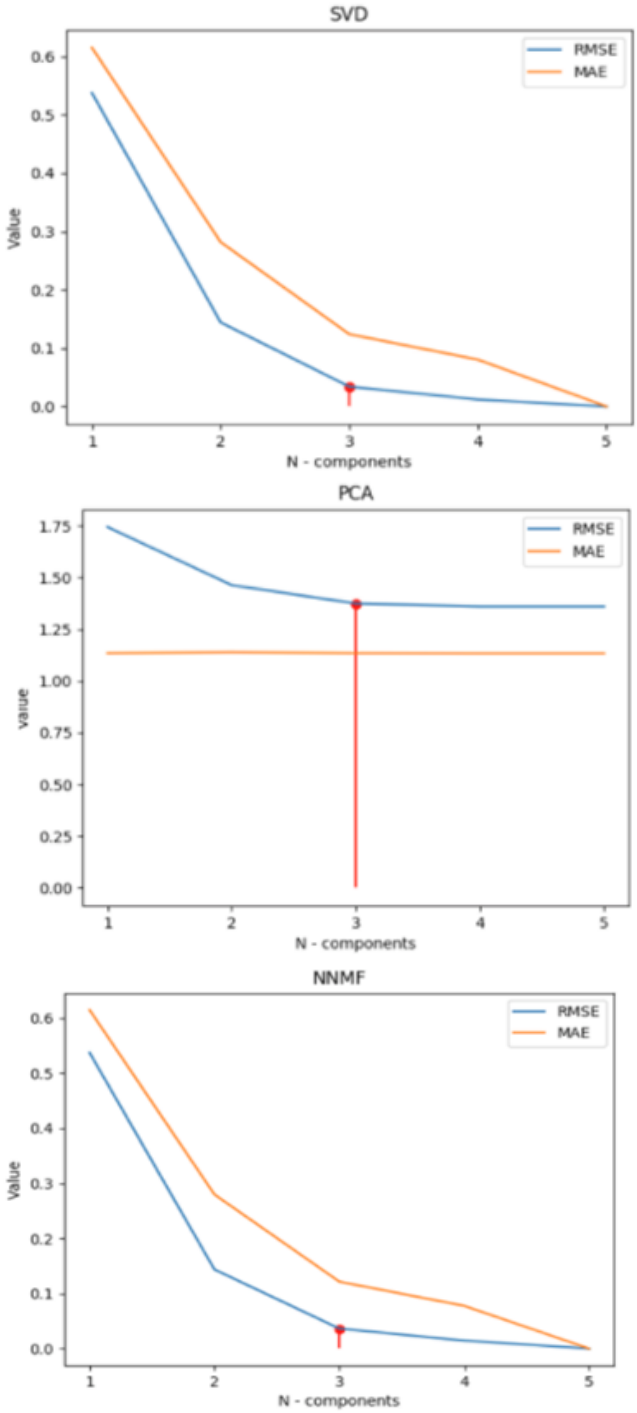

Figure 8: Comparison of basic decomposition methods

```
let ME_USER_RATINGS = [
    addUserRating(ME_USER_ID,  searchTitle: 'Terminator 3: Rise of the Machines',  rating: '5.0', MOVIES_IN_LIST),
    addUserRating(ME_USER_ID,  searchTitle: 'Jarhead',  rating: '4.0', MOVIES_IN_LIST),
    addUserRating(ME_USER_ID,  searchTitle: 'Back to the Future Part II',  rating: '3.0', MOVIES_IN_LIST),
    addUserRating(ME_USER_ID,  searchTitle: 'Jurassic Park',  rating: '4.0', MOVIES_IN_LIST),
    addUserRating(ME_USER_ID,  searchTitle: 'Reservoir Dogs',  rating: '3.0', MOVIES_IN_LIST),
    addUserRating(ME_USER_ID,  searchTitle: 'Men in Black II',  rating: '3.0', MOVIES_IN_LIST),
    addUserRating(ME_USER_ID,  searchTitle: 'Bad Boys II',  rating: '5.0', MOVIES_IN_LIST),
    addUserRating(ME_USER_ID,  searchTitle: 'Sissi',  rating: '1.0', MOVIES_IN_LIST),
    addUserRating(ME_USER_ID,  searchTitle: 'Titanic',  rating: '1.0', MOVIES_IN_LIST),
];
```

Figure 9: Customer feedback information, where each line corresponds to the addition of a new review for a new movie**.**

```
[ { title: 'Batman Begins', score: 1.000000000000002 },
  { title: 'The Dark Knight', score: 0.4640114093054991 },
  { title: 'The Dark Knight Rises', score: 0.4452538047955756 },
  { title: 'Batman: Under the Red Hood',
    score: 0.40475515773768195 },
  { title: 'Lovecraft: Fear of the Unknown',
    score: 0.385344351940051 },
  { title: 'Sherlock Holmes: A Game of Shadows',
    score: 0.377177700012597 },
  { title: 'Batman & Robin', score: 0.3749605328127467 },
  { title: 'Iron Man', score: 0.370344713231993 },
  { title: 'Helter Skelter', score: 0.36784956658401313 },
  { title: 'She\'s Lost Control', score: 0.3656927531464462 } ]
```

Figure 10: Output of movies that are offered to the user through personalized analysis

5. Discussion of results

In an information-saturated world, referral systems play an essential role in the user's interaction only with potentially exciting information. In this paper, a comparative analysis of the main approaches to implementing procedures of this kind. Several basic methods were compared during the study. The most popular are basic, subject-basic, hybrid-basic and matrix factorization. Figure 11 shows the results of this study. On the results shown in Figure 11, it can be seen that models give the best accuracy based on the hybrid approach and matrix factorization. If there are opportunities and necessary personalized recommendations, then the best ones are the ones that are different from neural networks and other approaches. After all, they remain transparent and easy to implement. If personalization is not required, using a popularity system is sufficient for most tasks. They also significantly simplify the procedure.

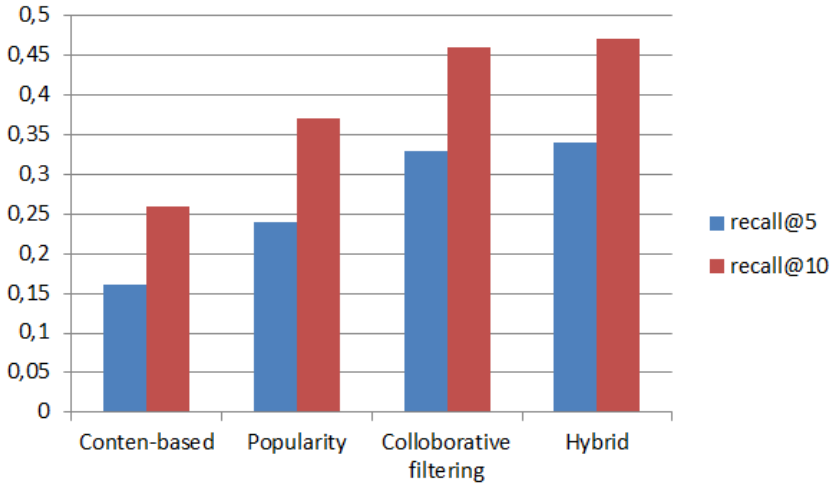

Figure 11: Evaluation of the results of the use of methods in the construction of recommendation systems

6. Conclusions

Today, during a pandemic, many businesses have their recommendation pages. For example, we can name such giants as Amazon, Google, Linkedin. In this paper, much attention was paid to methods based on matrix expansions for recommendation systems, namely for the reconstruction of the rating table. Based on these methods, data analysis for the selected dataset was performed. Each of the studied methods has its characteristics and is worth noting because it is helpful for a specific range of goals set by the developer of recommendation systems. Thus, the proposed models allow us to focus on the characteristics of the object, which determine the rating of the product or service it is

looking for. The application of the proposed algorithms allowed you to choose the best option for creating your recommendation system, which offers the user a behaviour model.

7. References

[1] S. Bo, Ch. Haiyan, A Survey of k Nearest Neighbor Algorithms for Solving the Class Imbalanced Problem, Wireless Communications and Mobile Computing, Vol. 2021, 2021. https://doi.org/10.1155/2021/5520990.

[2] A. Trabelsi, Z. Elouedi, and E. Lefevre, Decision tree classifiers for evidential attribute values and class labels, Fuzzy Sets and Systems, vol. 366, 2019, pp. 46–62.

[3] M. T. Jones, Recommender systems, Introduction to approaches and algorithms. Retrieved November 25, 2017.https://www.ibm.com/developerworks/library/os-recommender1/

[4] N. Shakhovska, N. Boyko, P. Pukach, The information model of cloud data warehouses. Advances in Intelligent Systems and Computing (AISC), 871, 2019, pp. 182–191

[5] N. Kunanets, O. Vasiuta, N. Boiko, Advanced Technologies of Big Data Research in Distributed Information Systems, in: Proceedings of the 14th International conference "Computer sciences and Information technologies", 2019, pp. 71-76. DOI: 10.1109/STC-CSIT.2019.8929756

[6] A. Tejeda-Lorente, C. Porcel, E. Peis, R. Sanz, E. Herrera-Viedma, A quality-based recommender system to disseminate information ina university digital library, Information Sciences, 266, 2014, pp. 52 - 69.

[7] C. Aggarwal, Recommender Systems. Springer, 2016, p. 498. doi: https://doi.org/10.1007/978-3-319-29659-3

[8] B. Hallinan, T. Striphas, Recommended for you: The Netf lix Prize and the production of algorithmic culture. New Media & Society, 18 (1), 2014, pp. 117–137. doi: https://doi.org/10.1177/1461444814538646

[9] G. Adomavicius, J. Bockstedt, S. Curley, J. Zhang, De-Biasing User Preference Ratings in Recommender Systems, in: Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems co-located with ACM Conference on Recommender Systems, 2–9. Available at: http://ceur-ws.org/Vol-1253/paper1.pdf

[10] I. Gunes, C. Kaleli, A. Bilge, H. Polat, Shilling attacks against recommender systems: a comprehensive survey. Arti-ficial Intelligence Review, 42 (4), 2014, pp. 767–799. doi: https://doi.org/10.1007/s10462-012-9364-9

[11] Y. Wang, L. Qian, F. Li, L. Zhang, A Comparative Study on Shilling Detection Methods for Trustworthy Recommen-dations. Journal of Systems Science and Systems Engineering, 27 (4), 2018, pp. 458–478. doi: https://doi.org/10.1007/s11518-018-5374-8

[12] K. Patel, A. Thakkar, C. Shah, K. Makvana, A State of Art Survey on Shilling Attack in Collaborative Filtering Based Recommendation System. Smart Innovation, Systems and Technologies, 2018, pp. 377–385. doi: https://doi.org/10.1007/978-3-319-30933-0_38

[13] W. Zhou, J. Wen, M. Gao, H. Ren, P. Li, Abnormal Profiles Detection Based on Time Series and Target Item Analysis for Recommender Systems. Mathematical Problems in Engineering, 2015, pp. 1–9. doi: https://doi.org/10.1155/2015/490261

[14] M. Gao, Q. Yuan, B. Ling, Q. Xiong, Detection of Abnormal Item Based on Time Intervals for Recommender Sys-tems. The Scientific World Journal, 2014, pp. 1–8. doi: https://doi.org/10.1155/2014/845897

[15] M. Gao, R. Tian, J. Wen, Q. Xiong, B. Ling, L. Yang, Item Anomaly Detection Based on Dynamic Partition for Time Series in Recommender Systems. PLOS ONE, 10 (8), 2015, pp.135-155. doi: https://doi.org/10.1371/journal.pone.0135155

[16] O. Chala, L. Novikova, L. Chernyshova, Method for detecting shilling attacks in e-commerce systems using weighted temporal rules. EUREKA: Physics and Engineering, 5, 2017, pp. 29–36. doi: https://doi.org/10.21303/2461-4262.2019.00983

[17] V. Levykin, O. Chala, Method of determining weights of temporal rules in Markov logic network for building knowledge base in information control systems. EUREKA: Physics and Engineering, 5, 2018, pp. 3–10. doi: https://doi.org/10.21303/2461-4262.2018.00713

[18] S. Chalyi, V. Leshchynskyi, I. Leshchynska, Method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system. EUREKA: Physics and Engineering, 4, 2019, pp. 34–40. doi: https://doi.org/10.21303/2461-4262.2019.00952