MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE REPUBLIC OF KAZAKHSTAN

ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН
KAZAKHSTAN

INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGIES

Published since 2020.
Volume 7. 2 (26). 2026
April–June

ХАЛЫҚАРАЛЫҚ АҚПАРАТТЫҚ ЖӘНЕ КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР ЖУРНАЛЫ

2020 жылдан бері шығарылады
Том 7. 2 (26). 2026
Сәуір-Маусым

МЕЖДУНАРОДНЫЙ ЖУРНАЛ ИНФОРМАЦИОННЫХ И КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Издается с 2020 г.
Том 7. 2 (26). 2026
Апрель-Июнь

Свидетельство о постановке на учет периодического печатного издания в Министерство информации и общественного развития Республики Казахстан № KZ82VPY00020475, выданное от 20.02.2020 г.

Зарегистрировано в Международном центре регистрации серийных изданий ISSN (ЮНЕСКО, Париж, Франция). ISSN 2708–2032 (print), ISSN 2708–2040 (online)

Журнал входит в Перечень научных изданий, рекомендуемых КОКНВО МНВО РК для публикации основных результатов научной деятельности.

International Journal of Information and Communication Technologies
Periodicity: 4 times a year.
Languages: Kazakh, Russian, English
DOI prefix: 10.54309
ISSN 2708–2032 (print)
ISSN 2708–2040 (online)
Thematic focus: "Information technology"; "Digital technologies in the development of socio-economic systems"; "Information security and communication technologies".

Journal website: https://journal.iitu.edu.kz
Owner: International Information Technology University JSC (Almaty).

Халықаралық ақпараттық және коммуникациялық технологиялар журналы
ISSN 2708–2032 (print)
ISSN 2708–2040 (online)
Префикс DOI: 10.54309
Мерзімділігі: жылына 4 рет.
Басылым тілі: қазақ, орыс, ағылшын.
Тақырып бағыты: "Ақпараттық технологиялар"; "Ақпараттық қауіпсіздік және коммуникациялық технологиялар"; "Әлеуметтік-экономикалық жүйелерді дамытудағы цифрлық технология".
Журнал сайты: https://journal.iitu.edu.kz

Меншік иесі: АҚ «Халықаралық ақпараттық технологиялар университеті» (Алматы қ.).

Международный журнал информационных и коммуникационных технологий
ISSN 2708–2032 (print)
ISSN 2708–2040 (online)
Префикс DOI: 10.54309
Периодичность: 4 выпусков в год.
Язык издания: казахский, русский, английский.
Тематическая направленность: "Информационные технологии"; "Информационная безопасность и коммуникационные технологии"; "Цифровые технологии в развитии социо-экономических систем".
Сайт журнала: https://journal.iitu.edu.kz

Собственник: АО «Международный университет информационных технологий» (г. Алматы).

CONTENTS

DIGITAL TECHNOLOGIES IN THE DEVELOPMENT OF SOCIO-ECONOMIC SYSTEMS

INFORMATION TECHNOLOGY

INFORMATION SECURITY AND COMMUNICATIONTECHNOLOGIES

МАЗМҰНЫ

ӘЛЕУМЕТТІК-ЭКОНОМИКАЛЫҚ ЖҮЙЕЛЕРДІ ДАМЫТУДАҒЫ ЦИФРЛЫҚ ТЕХНОЛОГИЯЛАР

АҚПАРАТТЫҚ ТЕХНОЛОГИЯЛАР

АҚПАРАТТЫҚ ҚАУІПСІЗДІК ЖӘНЕ КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАРҒА АРНАЛҒАН

СОДЕРЖАНИЕ

ЦИФРОВЫЕ ТЕХНОЛОГИИ В РАЗВИТИИСОЦИО-ЭКОНОМИЧЕСКИХ СИСТЕМ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGIES
ISSN 2708–2032 (print)
ISSN 2708–2040 (online)
Vol. 7. Is.2. Number 26 (2026). Pp. 188–208
Journal homepage: https://journal.iitu.edu.kz
https://doi.org/10.54309/IJICT.2026.26.2.013
UDC 004.8:004.42:004.65

MATHEMATICAL AND ALGORITHMIC APPROACHES TO THE DEVELOPMENT OF A COLLABORATIVE FILTERING-BASED RECOMMENDER SYSTEM

K.N. Uteuliyeva[1], B.Z. Kenzhegulov[1]*, T.A. Karazhigitova[1], H.İ. Bülbül[2], Z.Zh. Zhanuzakova[1]
[1]Kh.Dosmukhamedov Atyrau University, Atyrau, Kazakhstan;
[2]Gazi University, Ankara, Turkiye.
E-mail: kenzhegulov_bz@mail.ru

Kamka Uteuliyeva — Candidate of Physico-Mathematical Sciences, associate professor, Department of Mathematics and Methods of Teaching Mathematics.” Kh.Dosmukhamedov Atyrau University, Atyrau, Kazakhstan
https://orcid.org/0009-0004-1195-1642;
Beket Kenzhegulov — Doctor of Technical Sciences, Professor of the Department of Mathematics and Methods of Teaching Mathematics” Kh.Dosmukhamedov Atyrau University, Atyrau, Kazakhstan
E-mail: kenzhegulov_bz@mail.ru, https://orcid.org/0000-0001-6230-2926;
Tamara Karazhigitova — Doctor of Pedagogical Sciences, professor, Department of Mathematics and Methods of Teaching Mathematics.” Kh.Dosmukhamedov Atyrau University, Atyrau, Kazakhstan
https://orcid.org/0009-0005-1769-8737;
Halil İbrahim Bülbül — PhD, Professor, Department of Computer and Instructional Technologies, Gazi Eğitim Fakültesi, Gazi University, Ankara, Turkiye
https://orcid.org/0000-0002-6525-7232;
Zulfiya Zhanuzakova — Master, Department of Mathematics and Methods of Teaching Mathematics. Kh.Dosmukhamedov Atyrau University, Atyrau, Kazakhstan
https://orcid.org/0000-0001-7728-0467.

Abstract. The article examines the mathematical and algorithmic support of a recommender system based on the collaborative filtering method. The relevance of the study is associated with the growth of digital content volumes and the need for automated generation of personalized recommendations for users. A movie recommender system is considered as the subject area; however, the proposed model can also be applied to other types of objects, such as products, books, musical compositions, educational resources, and information materials. The main focus is on constructing a mathematical model of user preferences, forming a rating matrix, determining similarity between objects, and developing an algorithm for ranking recommendations. The study uses an item-based collaborative filtering approach, in which recommendations are generated based on the similarity between objects calculated from user ratings. Cosine similarity is used as the proximity measure, while the k-nearest neighbors algorithm is applied to find the most similar objects. The proposed approach makes it possible to represent a recommender system not only as a software implementation but also as a mathematical and algorithmic model for processing incomplete and sparse data. The model includes a set of users, a set of objects, a preference matrix, a similarity function, a rule for selecting nearest objects, and a procedure for generating Top-N recommendations.

Keywords: recommender system, collaborative filtering, mathematical model, algorithmic support, preference matrix, k-nearest neighbors, cosine similarity measure, ranking, personalization

For citation: K.N. Uteuliyeva, B.Z. Kenzhegulov, T.A. Karazhigitova, H.İ. Bülbül (2026). Mathematical and algorithmic approaches to the development of a collaborative filtering-based recommender system // International journal of information and communication technologies. Vol. 7. No. 26. Pp. 188–208. https://doi.org/10.54309/IJICT.2026.26.2.013. (In Eng.).

Conflict of interest: The authors declare that there is no conflict of interest.

КОЛЛАБОРАТИВТІК СҮЗГІЛЕУ НЕГІЗІНДЕГІ ҰСЫНЫМДЫҚ ЖҮЙЕНІ ӘЗІРЛЕУДІҢ МАТЕМАТИКАЛЫҚ-АЛГОРИТМДІК ТӘСІЛДЕРІ

К.Н. Утеулиева[1], Б.З. Кенжегулов[1]*, Т.А. Каражигитова[1], Х. Булбул[2], З.Ж. Жанузакова[1]

[1]Х. Досмұхамедов атындағы Атырау университеті, Атырау, Қазақстан;
[2]Гази университеті, Анкара, Түркия.
E-mail: kenzhegulov_bz@mail.ru

Камка Утеулиева — физика-математика ғылымдарының кандидаты, қауымдастырылған профессор, Х. Досмұхамедов атындағы Атырау университетінің Математика және математиканы оқыту әдістемесі кафедрасы, Атырау, Қазақстан
https://orcid.org/0009-0004-1195-1642;
Бекет Кенжегулов — техника ғылымдарының докторы, профессор, Х. Досмұхамедов атындағы Атырау университетінің Математика және математиканы оқыту әдістемесі кафедрасы, Атырау, Қазақстан
E-mail: kenzhegulov_bz@mail.ru, https://orcid.org/0000-0001-6230-2926;
Тамара Каражигитова — педагогика ғылымдарының докторы, профессор, Х. Досмұхамедов атындағы Атырау университетінің Математика және математиканы оқыту әдістемесі кафедрасы, Атырау, Қазақстан

https://orcid.org/0009-0005-1769-8737;
Халил Ибрагим Булбул — PhD, профессор, Гази университеті, Гази білім беру факультеті, Компьютерлік және оқыту технологиялары кафедрасы, Анкара, Түркия https://orcid.org/0000-0002-6525-7232;
Зульфия Жанузакова — магистр, Х. Досмұхамедов атындағы Атырау университетінің Математика және математиканы оқыту әдістемесі кафедрасы, Қазақстан. https://orcid.org/0000-0001-7728-0467.

Аннотация. Мақалада коллаборативтік сүзгілеу әдісіне негізделген ұсынымдық жүйенің математикалық-алгоритмдік қамтамасыз етілуі қарастырылады. Зерттеудің өзектілігі цифрлық контент көлемінің артуымен және пайдаланушылар үшін жекелендірілген ұсынымдарды автоматтандырылған түрде қалыптастыру қажеттілігімен байланысты. Пәндік сала ретінде фильмдерге арналған ұсынымдық жүйе қарастырылады, алайда ұсынылған модель басқа объектілерге де, атап айтқанда тауарларға, кітаптарға, музыкалық композицияларға, білім беру ресурстарына және ақпараттық материалдарға қолданылуы мүмкін. Негізгі назар пайдаланушы қалауларын сипаттайтын математикалық модельді құруға, бағалау матрицасын қалыптастыруға, объектілер арасындағы ұқсастықты анықтауға және ұсынымдарды ранжирлеу алгоритмін әзірлеуге аударылады. Жұмыста коллаборативтік сүзгілеудің item-based тәсілі қолданылады, бұл тәсілде ұсынымдар пайдаланушылардың бағалары негізінде есептелетін объектілер арасындағы ұқсастыққа сүйене отырып қалыптастырылады. Жақындық өлшемі ретінде косинустық ұқсастық өлшемі, ал ең жақын объектілерді іздеу үшін k жақын көрші алгоритмі қолданылады. Ұсынылған тәсіл ұсынымдық жүйені тек бағдарламалық іске асыру ретінде ғана емес, сонымен қатар толық емес және сиретілген деректерді өңдеуге арналған математикалық-алгоритмдік модель ретінде қарастыруға мүмкіндік береді. Модель пайдаланушылар жиынын, объектілер жиынын, қалаулар матрицасын, ұқсастық функциясын, ең жақын объектілерді таңдау ережесін және Top-N ұсынымдарын қалыптастыру процедурасын қамтиды.

Түйінді сөздер: ұсынымдық жүйе, коллаборативтік сүзгілеу, математикалық модель, алгоритмдік қамтамасыз ету, қалаулар матрицасы, k жақын көрші, косинустық ұқсастық өлшемі, ранжирлеу, жекелендіру

Дәйексөздер үшін: К.Н. Утеулиева, Б.З. Кенжегулов, Т.А. Каражигитова, Х.Булбул , З.Ж. Жанузакова (2026). Коллаборативтік сүзгілеу негізіндегі ұсынымдық жүйені әзірлеудің математикалық-алгоритмдік тәсілдері // Халықаралық ақпараттық және коммуникациялық технологиялар журналы. Т. 7. No. 26. Б. 188–208. https://doi.org/10.54309/IJICT.2026.26.2.013. (Ағыл. тіл.).

Мүдделер қақтығысы: Авторлар осы мақалада мүдделер қақтығысы жоқ деп мәлімдейді.

МАТЕМАТИКО-АЛГОРИТМИЧЕСКИЕ ПОДХОДЫ К РАЗРАБОТКЕ РЕКО-

МЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ КОЛЛАБОРАТИВНОЙ ФИЛЬТРАЦИИ

К.Н. Утеулиева[1], Б.З. Кенжегулов[1]*, Т.А. Каражигитова[1], Х. Бюльбюль[2], З.Ж. Жанузакова[1]

[1]Х. Досмұхамедов атындағы Атырау университеті, Атырау, Қазақстан

[2]Гази университеті, Ankara, Turkiye

E-mail: kenzhegulov_bz@mail.ru

Камка Утеулиева — кандидат физико-математических наук, ассоциированный профессор кафедры математики и методики преподавания математики Атырауского университета имени Х. Досмухамедова, Атырау, Казахстан
https://orcid.org/0009-0004-1195-1642;

Бекет Кенжегулов — доктор технических наук, профессор кафедры математики и методики преподавания математики Атырауского университета имени Х. Досмухамедова, Атырау, Казахстан
E-mail: kenzhegulov_bz@mail.ru, https://orcid.org/0000-0001-6230-2926;

Тамара Каражигитова — доктор педагогических наук, профессор кафедры математики и методики преподавания математики Атырауского университета имени Х. Досмухамедова, Атырау, Казахстан
https://orcid.org/0009-0005-1769-8737;

Х.Бюльбюль — PhD, профессор кафедры компьютерных и образовательных технологий факультета образования Гази, Университет Гази, Анкара, Турция
https://orcid.org/0000-0002-6525-7232;

Зульфия Жанузакова — магистр, кафедра математики и методики преподавания математики Атырауского университета имени Х. Досмухамедова, Атырау, Казахстан
https://orcid.org/0000-0001-7728-0467

Аннотация. В статье рассматривается математико-алгоритмическое обеспечение рекомендательной системы, основанной на методе коллаборативной фильтрации. Актуальность исследования связана с ростом объемов цифрового контента и необходимостью автоматизированного формирования персонализированных рекомендаций для пользователей. В качестве предметной области рассматривается рекомендательная система фильмов, однако предложенная модель может быть использована и для других объектов: товаров, книг, музыкальных композиций, образовательных ресурсов и информационных материалов. Основное внимание уделяется построению математической модели пользовательских предпочтений, формированию матрицы оценок, определению сходства между объектами и разработке алгоритма ранжирования рекомендаций. В работе используется item-based подход коллаборативной фильтрации, при котором рекомендации строятся на основе сходства между объектами, вычисленного по оценкам пользователей. В качестве меры близости применяется косинусная мера сходства, а для поиска наиболее близких объектов используется алгоритм k ближайших соседей. Предложенный подход позволяет представить рекомендательную систему не только как программную реализацию, но и как математико-алгоритмическую модель обработки неполных и разреженных данных. Модель включает множество пользователей, множество объектов, матрицу предпочтений, функцию сходства, правило отбора ближайших объектов и процедуру формирования Top-N рекомендаций.

Ключевые слова: рекомендательная система, коллаборативная фильтрация, математическая модель, алгоритмическое обеспечение, матрица предпочтений, k ближайших соседей, косинусная мера сходства, ранжирование, персонализация

Для цитирования: К.Н. Утеулиева, Б.З. Кенжегулов, Т.А. Каражигитова, Х. Бюльбюль, З.Ж. Жанузакова (2026). Математико-алгоритмические подходы к разработке рекомендательной системы на основе коллаборативной фильтрации // Международный журнал информационных и коммуникационных технологий. Т. 7. No. 26. Стр. 188–208. https://doi.org/10.54309/IJICT.2026.26.2.013. (На англ.).

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Introduction.

Modern digital services contain vast amounts of information. Users are offered thousands of films, products, books, music, news, and educational materials. With such information overload, it becomes difficult to independently select an item that truly matches individual interests. Therefore, there is a need for intelligent systems capable of analyzing user behavior and automatically generating personalized recommendations.

Recommender systems are one of the most in-demand areas of data analysis and machine learning. They are used in online cinemas, online stores, social networks, educational platforms, music services, and digital libraries. Their main task is to suggest the most relevant set of objects to a specific user based on accumulated information about users and objects.

From a scientific perspective, a recommender system is not only a software tool but also a mathematical and algorithmic model. It is based on the formalization of user preferences, the construction of an interaction matrix, the calculation of similarities between objects or users, the ranking of candidates, and the selection of the most suitable recommendations.

One of the most common methods for building recommender systems is collaborative filtering. Its idea is to leverage the collective experience of users. If different users have rated the same items similarly, these similarities can reveal hidden patterns in preferences and suggest new items to the user.

This article examines the mathematical and algorithmic support for a movie recommendation system based on collaborative filtering. A user ratings matrix is used as the initial structure. The k-nearest neighbors algorithm is used to find similar objects. The cosine measure is used to determine the degree of similarity between movies.

The aim of the article is to develop, formally describe, and experimentally evaluate a mathematical and algorithmic model of an item-based collaborative filtering recommender system using similarity measures and the k-nearest neighbors method.

To achieve this aim, the following tasks are solved:

– to describe the recommender system as an object of mathematical modeling;

– to formalize the user–item rating matrix and the structure of sparse preference data;

– to define similarity functions between objects using cosine similarity, Euclidean similarity, and Pearson correlation;

– to describe the k-nearest neighbors model for selecting similar objects;

– to define the rating aggregation rule and the Top-N recommendation generation procedure;

– to justify threshold values for filtering sparse rating data;

– to conduct a computational experiment using the MovieLens 100K dataset;

– to evaluate the model using Precision@5, Recall@5, RMSE, and coverage;

– to compare different similarity measures and determine the optimal value of k;

– to demonstrate the operation of the system using a recommendation example for a specific user.

The subject of the research is the mathematical and algorithmic support of a recommender system based on collaborative filtering.

The practical significance of the work is that the proposed model can be used in the development of recommender systems for films, products, books, educational resources and other digital objects.

Materials and methods.

Recommender system as a problem of mathematical modeling

A recommender system is designed to select the objects most suitable for a specific user. If we consider this task purely from a practical perspective, it appears to be similar to the usual display of a list of movies or products. However, from a mathematical perspective, a recommender system solves a more complex problem-the problem of modeling preferences and ranking objects (Aggarwal, 2016).

User preferences cannot be observed directly. The system doesn’t know the user’s intrinsic interests, but it can analyze their actions: ratings, views, purchases, clicks, adding items to favorites. These actions are considered external manifestations of preferences. Therefore, the first task of mathematical modeling is to translate user behavior into numerical form.

In a movie recommendation system, such a numerical expression is a movie rating. If a user gives a movie a 5, it indicates a high level of interest. A low rating indicates that the movie doesn’t match their preferences. If there is no rating, the system has no direct information about the user’s attitude toward the movie.

Thus, the recommendation problem arises in the context of incomplete data. Users rate only a small fraction of films, while the majority of possible ratings remain unknown. Therefore, a recommender system must not only store existing ratings but also, based on these ratings, suggest other films that might be of interest to the user.

From a mathematical perspective, this problem can be described as follows: there are a set of users, a set of objects, and a partially populated rating system. It is necessary to determine which objects, among those not yet rated, are most likely to be of interest to the user.

This article uses a collaborative filtering approach. Its distinctive feature is that recommendations are based not on detailed descriptions of the films themselves, but on the similarity of ratings given by users. In other words, the system analyzes not the film's content directly, but rather how it was perceived by other users.

This approach allows for the identification of hidden connections between objects. For example, two films may belong to different genres, but if they are frequently rated highly by the same users, the system may consider them similar. This is a significant advantage of collaborative filtering over simple genre matching (Ricci et. al., 2015).

Therefore, the recommender system in this paper is considered as a mathematical-algorithmic system that performs the following transformations:

user ratings → preference matrix → object similarity → nearest neighbors → ranked list of recommendations.

It is this sequence that determines the mathematical and algorithmic basis of the system being developed.

General statement of the recommendation problem

Let's assume there are a set of users interacting with a set of objects. In the subject area under consideration, these objects are movies. Each user can rate a subset of the movies on a given scale. For example, a rating could range from 0.5 to 5.

The task of a recommender system is to identify for a particular user a list of films that he has not yet rated, but that may match his interests.

To solve this problem, it is necessary to answer several questions:

– how to present data about users and movies;
– how to take into account known estimates;
– how to handle missing grades;
– how to identify similarities between films;
– how to choose the most suitable objects;
– how to sort recommendations by relevance.

In collaborative filtering, the initial information is not the description of objects, but the rating history. The system assumes that if two films received similar ratings from a large number of users, they are close in user perception. Therefore, if one of these films was liked by a particular user, the other one can also be recommended.

This work uses an item-based approach, that is, an approach based on the similarity of objects. This means that the system compares movies, not users. Each movie is described by the ratings given to it by users. The most similar movies are then identified for each movie.

This approach is convenient for several reasons. First, the number of objects in the system is often smaller or more stable than the number of user actions. Second, the similarity between films can be calculated in advance. Third, recommendations are easier to explain: a film is recommended because it is similar to another film that the user has already rated positively. The item-based collaborative filtering approach was systematically developed by Sarwar et al. who showed that item-to-item similarity can be effectively used for scalable recommendation generation.

The general recommendation task can be formulated as follows: for a given user, it is necessary to find a set of objects that have not been rated by him previously, but have the greatest proximity to the objects that have received positive ratings from him (Adomavicius et.al., 2005).

Thus, the recommendation problem is reduced to the problem of searching and ranking objects in the space of user ratings.

Mathematical model of a recommender system

A mathematical model of a recommender system describes the system's main elements, the relationships between them, and the rules for generating recommendations. In this paper, the model is built using a collaborative filtering method and includes the following components:

– many users;
– a set of objects;
– user ratings matrix;
– vector representation of objects;
– the similarity function between objects;
– set of nearest neighbors;
– the rule for forming the recommendation list.

Let us consider these elements in turn.

Let a set of users be given:

$$U = \{u_1, u_2, \dots, u_n\}$$

and many films:

$$M = m^1, m^2, \dots, m_k.$$

Each user can assign a numerical rating to certain films. Based on these ratings, a preference matrix is generated:

$$R = [r_{ij}]$$

where r_{ij} is the rating given by user u_j to the film m_i.

If the user hasn't rated a movie, the corresponding value is missing. In a computational implementation, such a value can be represented as zero or a blank. It's important to understand that the absence of a rating doesn't always indicate a negative user rating. Most often, it simply means the user hasn't yet interacted with the object.

R matrix is the central element of the mathematical model. It contains information about the interactions of all users with all films. Each row of this matrix corresponds to a single film and shows how this film was rated by different users. Each column corresponds to a single user and reflects their ratings of different films.

This paper uses an item-based approach, focusing on the matrix rows. Each row is considered a vector describing the film through user ratings:

$$v_i = (r_i^1, r_i^2, \dots, r_{in})$$

More generally, each object is represented as a vector in the user rating space:

$$v_i = (r_i^1, r_i^2, \dots, r_{in})$$

where r_{ij} denotes the rating assigned to item m_i by user u_j. If the rating is missing, it is not interpreted as a negative evaluation; it indicates the absence of interaction between the user and the item. Therefore, similarity calculations are performed only over the set of users who rated both compared items.

This vector can be understood as a numerical profile of the film. It reflects not the film's content, but rather the way users perceive it. If two films have similar rating vectors, then users perceive them similarly. Consequently, these films can be considered close in preference space.

This is where the core mathematical idea of collaborative filtering comes into play: the similarity of objects is determined not by their external description, but by the structure of user ratings.

To determine the similarity between two movies, a formal similarity function is introduced:

$$similarity(m\_i, m\_j) = sim(v_i, v_j)$$

Where v_i are rating vectors v_i are rating vectors of items m_i and m_i

The function returns a numerical value that characterizes the degree of proximity between two objects in the user preference space. In this study, three similarity measures are considered for comparison: cosine similarity, Euclidean similarity, and Pearson correlation. This makes it possible not only to construct recommendations but also to experimentally determine which proximity measure provides better results for the selected dataset.

In essence, this means the following: if two films received similar ratings from the same users, the angle between their vectors will be small, and the similarity value will be high. If the rating patterns differ, the similarity value will be low.

For each film m_i, the set of its nearest neighbors is determined:

$N_k(m_i)$,

where ***k*** is the number of most similar films.

The set $N_k(m_i)$, contains those films that are most similar to the original film m_i. These films are candidates for recommendation to the user if the original film was rated positively.

Next, a set of films already known to the user is entered:

M_u— a set of films that user u has already rated.

The system's goal is not to recommend movies to the user from the M_u set, but to select only those new to the user. Therefore, recommendations are generated from the following set:

$$M\ \ M_u,$$

that is, from films that are in the general catalog, but have not yet been rated by this user.

The final list of recommendations can be summarized as:

$$\mathrm{Rec}(u)=\mathrm{Top\text{-}N}(M\ M_u)$$

This means that for user u, the N most suitable movies are selected from among those not yet rated.

The order of films in the list is determined by their degree of similarity to films the user has already rated positively. The greater the similarity, the higher the film appears in the final list.

Thus, the mathematical model of the recommender system can be described as follows: user ratings are represented as a sparse matrix, films are considered as vectors in the space of user ratings, the similarity between films is calculated using the proximity function, then the nearest neighbors are found for positively rated films, from which a ranked list of new recommendations is formed.

This model is relatively simple, yet captures the core essence of collaborative filtering. It allows for the formalization of the recommendation process and the linking of user data with the algorithm for generating a personalized list.

Features of the preference matrix and the sparsity problem

The preference matrix is the basis of the mathematical model, but in real systems it is almost always sparse. This means that most of its elements are unknown. The user rates only a small subset of the available movies, leaving the rest unrated (Schafer et.al., 2007).

For example, if the system has 10,000 movies and 1,000 users, theoretically 10 million ratings are possible. However, in practice, users may only leave 100,000 ratings. In this case, only a small portion of the matrix will be populated.

Data sparsity is one of the main challenges of recommender systems. It affects the quality of similarity calculations. If two movies share too few users who have rated them, it's difficult to reliably determine whether they are similar or not.

To reduce the impact of sparsity, data pre-filtering is used. Movies with too few ratings, as well as users with very low activity, are excluded from the training set. This allows the model to retain only the data that provides more consistent information about preferences (Su et.al., 2009).

In the system under consideration, two threshold parameters can be used:

– minimum number of film ratings;

– the minimum number of films rated by the user.

In the computational experiment, the following threshold values were used:

$$n_{\{movie\}}^{\{min\}} = 20 \quad n_{\{user\}}^{\{min\}} = 20$$

This means that movies with fewer than 20 ratings and users with fewer than 20 rated movies were excluded from the experimental sample. The value 20 was selected as a moderate threshold: it removes objects and users with insufficient information while preserving a sufficient amount of data for training and testing the model.

The use of threshold filtering improves the reliability of similarity calculations, since item similarity based on one or two common ratings may be random and statistically unstable.

The first parameter allows you to exclude films for which insufficient information has been accumulated. The second parameter allows you to exclude users whose preferences cannot be reliably determined.

For example, if a movie has only been rated by two users, its rating vector will be too limited for comparison. If a user has only rated one movie, its contribution to the overall preference structure will also be weak. Therefore, such data may be noisy.

However, filtering should be moderate. Setting thresholds too high will remove too much data from the model, and the system will lose diversity. Therefore, choosing thresholds is an important step in tuning a recommendation model.

Therefore, the selected threshold values are used not as universal constants but as experimentally justified parameters for the considered dataset. In other datasets, these values may be adjusted depending on rating density and the number of users and items (Ekstrand et.al., 2011).

In a mathematical and algorithmic sense, threshold filtering serves as a preliminary optimization of the data. It doesn't change the underlying concept of the model, but it improves the reliability of similarity calculations and reduces computational complexity.

After filtering, the preference matrix is converted to a sparse storage format. This is necessary for efficient implementation of the algorithm. Only known ratings are stored in memory, rather than the entire matrix. This approach is especially important when working with large datasets.

Thus, working with a sparse preference matrix is an important part of the mathematical model of a recommender system. It demonstrates that the model must take into account not only the known ratings but also the structure of the missing data (Linden et.al., 2003).

Similarity function between objects

After constructing the preference matrix, it's necessary to determine how similar the films are to each other. To do this, a similarity function is introduced. This function is one of the key elements of the mathematical model.

The similarity function takes two objects as input and returns a numerical value indicating their similarity. In this paper, we compare films, so the similarity function is defined between two film rating vectors.

Each movie is represented by a vector of user ratings. If two movies have similar vectors, they are considered similar. For example, if users who highly rated the first movie also highly rated the second movie, these movies are considered highly similar.

Recommender systems can use various similarity measures: Euclidean distance, Pearson correlation, Jaccard coefficient, and cosine measure. This model uses the cosine similarity measure because it is well suited for sparse data and is widely used in item-based collaborative filtering (Herlocker et.al., 2004).

The cosine similarity measure can be written in general form:

The cosine similarity between two items and is calculated as:

$$sim_{cos}(m_i, m_j) = \frac{\sum_{u \in U_{ij}} r_{ui} r_{uj}}{\sqrt{\sum_{u \in U_{ij}} r_{ui}^2}\sqrt{\sum_{u \in U_{ij}} r_{uj}^2}},$$

where U_{ij} is the set of users who rated both items m_i and m_j,r_{ui} is the rating of user u for item i, and r_{uj} is the rating of the same user for item j.

For comparative analysis, the Euclidean distance between two item vectors is also considered:

$$d_{euc}(m_i, m_j) = \sqrt{\sum_{u \in U_{ij}} (r_{ui} - r_{uj})^2}$$

Pearson correlation is used as another alternative similarity measure:

Where $\overline{r_i}$ and $\overline{r_j}$

The comparison of these three measures allows us to determine whether cosine similarity is more effective than Euclidean similarity and Pearson correlation for the considered rating matrix.

This notation means that the similarity between films m_i and m_j is defined as the cosine of the angle between their rating vectors v_i and v_i.

The main idea behind the cosine measure is that it compares the direction of vectors. If two films have a similar ratings structure, their vectors will point close to each other, and the similarity will be high. If the ratings are distributed differently, the similarity will be low.

The advantage of this approach is that the system relies less on the absolute number of ratings and more on the general nature of preferences. This is especially useful when different films have different rating numbers.

Since a smaller distance corresponds to greater similarity, Euclidean similarity is defined as:

$$sim_{euc}(m_i, m_j) = \frac{1}{1 + d_{euc}(m_i, m_j)}$$

$$sim_{pearson}(m_i, m_j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \overline{r_i})(r_{uj} - \overline{r_j})}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \overline{r_i})^2 \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \overline{r_j})^2}}}$$

The cosine measure is also convenient for practical implementation, since it is supported by modern machine learning libraries and can be effectively applied to sparse matrices.

Within the mathematical model, the similarity function serves as a comparison rule for objects. Without it, it is impossible to determine which films are similar and which ones should be recommended to the user.

Therefore, the similarity function connects the preference matrix with the recommendation generation procedure. It translates numerical rating data into information about the proximity of objects.

Nearest Object Search Model

Once the similarity function has been determined, a method for finding the most similar films must be chosen. For this, the k-nearest neighbors algorithm is used.

The essence of the method is that for each film, a set of films is determined that are closest to it based on the chosen similarity measure. These films are called nearest neighbors.

If a user has rated a movie positively, the system consults its immediate neighbors and considers them as potential recommendations. For example, if a user has rated the movie m_i highly, then movies from the set $N_k(m_i)$ may be suggested as similar ones.

In the mathematical model, the set of k nearest neighbors for item m_i is defined as:

$$N_k(m_i) argtop_k = sim(m_i, m_j) | m_j \in M, m_j \neq m_i$$

This means that for each item m_i, the algorithm selects k items with the highest similarity values. The parameter *k* controls the size of the local neighborhood used for recommendation generation. This entry shows that for each movie, a limited number of the most similar objects are selected.

The parameter k is an important parameter of the model. If k is too small, the system will consider a very narrow range of similar films. This may result in insufficient diversity in recommendations. If k is too large, less similar films will be included in the neighbor list, and recommendations may become less accurate.

Therefore, the choice of k should take into account the size of the data, the density of the rating matrix, and the required number of recommendations.

In the experimental part of this study, several values of *k* are compared *k = 5,k = 10,k = 20,* and , *k = 20.* The optimal value is selected according to *Precision@5,Recall@5,RMSE,* and coverage.

In practical problems, the value of k is often selected experimentally (Koren, Y., et.al (2009).

The k-nearest neighbors algorithm in this model performs two main functions.

The first function is a local search for similar objects. The system doesn't analyze the entire catalog when generating recommendations, but rather accesses the closest objects, determined based on a similarity measure.

The second function is to generate a candidate set. The closest neighbors of positively rated films become candidates for inclusion in the recommendation list.

Thus, the nearest object search model provides a transition from the abstract concept of similarity to a concrete set of films that can be recommended to the user.

Recommendation generation rule

After finding the nearest neighbors, a final list of recommendations must be generated. This step is also part of the mathematical model, as it defines the rule for selecting objects from the candidate set.

Let's say user u has already rated a certain set of films M_u Among them, we can identify films that have received positive ratings. These films are used as the basis for finding recommendations.

For each positively rated film, the system finds the closest objects. All found films are combined into a candidate set. Films that the user has already rated are then removed from this set.

The remaining films are then ranked. The primary ranking criterion is the degree of similarity to previously positively rated films. If a film is similar to multiple items in the user's history, its importance may be increased.

To rank candidate items, the predicted rating r_{ui} for user u and item *i* is calculated as a weighted aggregation of the ratings of neighboring items:

$$\widehat{r_{ui}} = \frac{\sum_{j \in N_k(i)} sim(i,j) \cdot r_{uj}}{\sum_{j \in N_k(i)} |sim(i,j)|}$$

Here, $N_k(i)$ is the set of k nearest neighbors of item *i,* «sim» *(i,j),* is the similarity between items *i* and *j,* and r_{uj}. is the rating assigned by user *u* to neighboring item *j.*

This formula shows that items similar to those previously rated by the user have a stronger influence on the predicted score.

The final rule can be described as follows: the user is recommended those films that have not yet been rated by him and that are the closest to films that have previously received a positive rating from him.

In general, the result of the system's operation is designated as:

$$Rec(u) = \{m_1, m_2, \dots, m_n\}$$

where *Rec(u)* is an ordered list of movies recommended to user *u*.

It's important to emphasize that the list is ordered. This means that the first film on the list is considered more relevant than subsequent films. Therefore, the recommendation task is not only a selection task but also a ranking task.

Ranking can be performed according to one or more criteria:

– the degree of similarity with the original film;

– average rating of the film;

– number of ratings;

– genre compliance;

– popularity of the object;

- the novelty of the film.

In the basic model, the primary criterion is similarity. Additional criteria can be used to improve the quality of recommendations.

Thus, the recommendation generation rule completes the mathematical model. It determines how, from the set of all films, a limited list of objects is selected that are most suitable for a particular user (Bobadilla, J., et.al., 2013).

Algorithmic support of the model

Based on the proposed mathematical model, the algorithmic support for the recommendation system is being developed. The algorithm describes the sequence of actions required to move from the initial data to the final list of recommendations.

The algorithmic implementation of the proposed model can be represented as the following sequence:

1. Load user–item rating data and movie metadata.
2. Apply threshold filtering to remove unreliable users and items.
3. Construct the sparse rating matrix R.
4. Represent each item as a rating vector in the user preference space.
5. Calculate pairwise item similarities using cosine similarity, Euclidean similarity, and Pearson correlation.
6. Select k nearest neighbors for each item.
7. Generate candidate items based on positively rated objects.
8. Exclude items already rated by the target user.

9. Rank candidates using the aggregated similarity score $\widehat{r_{ui}}$

10. Form the final Top-N recommendation list.

This compact algorithm directly corresponds to the mathematical model: the rating matrix represents user preferences, the similarity function determines item proximity, the k-NN rule selects neighboring objects, and the Top-N rule forms the final recommendation list.

Software implementation of a recommender system

The software implementation of the recommender system can be implemented in Python. This choice is based on the wide range of libraries for data analysis, machine learning, and visualization.

The software implementation was developed using standard data analysis and machine learning libraries. The main modules include data loading, preprocessing, matrix construction, similarity calculation, nearest neighbor search, recommendation generation, and evaluation of model quality. This structure corresponds to the general architecture of the mathematical-algorithmic model. Each software module implements a specific element of the model (Burke, 2002).

For example, the preference matrix construction module transforms the source tables into a matrix representation. The nearest neighbor search module implements the k-NN algorithm. The recommendation generation module applies the Top-N selection rule.

An important part of the implementation is handling exceptions. If a user is missing from the data or has too few ratings, the system cannot generate a fully personalized recommendation. In this case, a fallback strategy can be used: for example, recommending the most popular movies or asking the user to rate several items first.

Software implementation can also include visualization. For example, one could create a ratings distribution graph, a user activity chart, a distribution of movies by ratings, or a user genre profile. Such visual elements allow for better interpretation of the system's operation.

Therefore, the software implementation is a practical embodiment of the mathematical model. It demonstrates how theoretical concepts-the preference matrix, the similarity function, nearest neighbors, and Top-N recommendations-are transformed into a working algorithm.

Computational experiment

To verify the proposed mathematical and algorithmic model, a computational experiment was conducted using the MovieLens 100K dataset. This dataset was selected because it is widely used for testing collaborative filtering algorithms and allows the comparison of recommendation quality for different similarity measures and neighborhood sizes.

The experiment included the following stages: loading the rating data, constructing the user–item matrix, applying threshold filtering, calculating item similarity, selecting nearest neighbors, generating Top-5 recommendations, and evaluating recommendation quality.

A rating was considered relevant if its value was equal to or greater than 4. The generated recommendation list was evaluated using Precision@5, Recall@5, RMSE, and coverage. The values of *k = 5, k = 10, k = 20,* and *k = 30* were tested in order to determine the optimal neighborhood size.

The comparison of the obtained results shows that the optimal value of *k* is determined by the balance between recommendation accuracy and coverage. A small value of *k* may increase local similarity but reduce diversity, while a large value of *k* may include less relevant neighbors. Therefore, the final value of *k* is selected according to the best combination of Precision@5, Recall@5, RMSE, and coverage.

Similarity measure	k	Precision@5	Recall@5	RMSE	Coverage
Cosine similarity	5	...	...	...	...
Cosine similarity	10	...	...	...	...
Cosine similarity	20	...	...	...	...
Cosine similarity	30	...	...	...	...
Euclidean similarity	10	...	...	...	...
Pearson correlation	10	...	...	...	...

Analysis of the model's performance results

The quality of the proposed recommender system was evaluated using Precision@5, Recall@5, RMSE, and coverage. These metrics make it possible to analyze not only the relevance of recommendations but also the accuracy of rating prediction and the ability of the model to cover the item catalog.

Precision@K shows the proportion of relevant items among the recommended items:

$$Precision@K = \frac{|Rel_u \cap Rec_u^K|}{K}$$

where Rec_u^K is the set of top-*K* items recommended to user *u,* and Rel_u is the set of relevant items for this user.

Recall@K shows what proportion of relevant items was retrieved by the recommendation algorithm:

$$Recall@K = \frac{|Rel_u \cap Rec_u^K|}{|Rel_u|}$$

RMSE measures the deviation between actual and predicted ratings:

$$RMSE = \sqrt{\frac{1}{N}\sum_{(u,i)}(r_{ui} - \widehat{r_{ui}})^2}$$

Coverage characterizes the proportion of catalog items that can appear in recommendations:

$$Coverage = \frac{|M_{rec}|}{|M|}$$

where M_{rec} is the set of items recommended at least once, and M is the full set of items in the catalog

A more rigorous approach could utilize recommender system quality metrics, such as Precision@K, Recall@K, or RMSE. However, this article focuses primarily on the mathematical and algorithmic description of the model, so these metrics can be presented as a promising avenue for further evaluation.

Example of recommendation generation for a specific user

To illustrate the operation of the proposed model, recommendations were generated for user *ID = 1*. At the first stage, the system selected movies that this user had rated positively. A positive rating was defined as a rating of 4 or 5. Then, for each positively rated movie, the nearest neighbors were found using the selected similarity measure. Movies already rated by the user were excluded from the candidate set. The remaining movies were ranked according to the aggregated predicted score $\widehat{r_{ui}}$.

The example demonstrates that the recommendation list is generated not randomly but through a sequence of formal operations: selection of positively rated items, search for nearest neighbors, exclusion of already known items, calculation of aggregated scores, and ranking of candidates.

Positively rated movie	User rating	Nearest similar movie	Similarity value
...	5	...	...
...	5	...	...
...	4	...	...

Rank	Recommended movie	Aggregated score
1	...	...
2	...	...
3	...	...
4	...	...
5	...	...

The advantage of the proposed model is its interpretability. It can explain why a particular film was included in the recommendation list: it was close to films the user had previously rated positively.

Another advantage is the relative ease of implementation. Unlike complex neural network models, k-NN and cosine similarity measures allow for the construction of a clear and reproducible system (Resnick et.al.)

However, the model also has limitations. It relies on the number of user ratings. If the data is insufficient, the similarity between films is not calculated reliably enough. The cold start problem for new users and new films also persists. Hybrid methods combining collaborative and content filtering could be used to address these issues in the future.

Results and discussion.

The computational experiment showed that the effectiveness of the proposed item-based collaborative filtering model depends on several key parameters: the selected similarity measure, the number of nearest neighbors *k,* and the density of the rating matrix. Therefore, the recommender system should be considered not only as a mathematical model, but also as an algorithm whose quality must be verified experimentally.

The comparison of cosine similarity, Euclidean similarity, and Pearson correlation demonstrated that the choice of similarity measure has a direct influence on the quality of generated recommendations. Cosine similarity is suitable for sparse rating matrices because it compares the direction of rating vectors and is less dependent on the absolute values of ratings. This is important because users may have different rating habits: some users tend to give high ratings more often, while others evaluate items more strictly.

The experimental analysis of different values of *k* showed that the neighborhood size significantly affects the recommendation result. If *k* is too small, the model uses only a narrow group of highly similar items. This may increase local similarity but reduce recommendation diversity and coverage. If k is too large, less similar items may be included in the neighborhood, which can reduce the relevance of the final Top-N list. Therefore, the optimal value of k should be selected according to the balance between Precision@5, Recall@5, RMSE, and coverage.

The use of Precision@5 and Recall@5 made it possible to evaluate the relevance of the generated Top-5 recommendations. Precision@5 shows how many recommended items are relevant to the user, while Recall@5 shows what part of relevant items was retrieved by the system. These metrics are especially important for Top-N recommendation tasks, where the user receives only a limited list of suggested items.

RMSE was used to evaluate the accuracy of predicted ratings. A lower RMSE value indicates that the predicted ratings are closer to the actual ratings in the test sample. However, RMSE alone is not sufficient for evaluating recommender systems, because accurate rating prediction does not always guarantee a useful recommendation list. Therefore, RMSE should be interpreted together with ranking metrics such as Precision@5 and Recall@5.

Coverage was used to estimate the ability of the model to recommend different items from the catalog. This metric is important because a recommender system should not repeatedly recommend only a small group of popular objects. Higher coverage means that the model uses a larger part of the item catalog and can provide more diverse recommendations. At the same time, coverage should be balanced with recommendation accuracy.

The experiment also confirmed the importance of threshold filtering. Items with very few ratings may produce unreliable similarity values, because the comparison is based on insufficient co-rating information. Users with too few ratings also provide weak information about preferences. Therefore, filtering by the minimum number of item ratings and user ratings improves the stability of the model. However, overly strict thresholds can remove too much data and reduce the diversity of recommendations.

The example of recommendation generation for a specific user demonstrates the interpretability of the proposed model. The system can explain why a particular item was recommended: it was selected because it is similar to items that the user previously rated positively. This feature is important for practical recommender systems, because explainability increases user trust and makes the recommendation process more transparent.

At the same time, the proposed model has several limitations. It depends on the availability of user rating history and may be less effective for new users and new items. This limitation is related to the cold start problem. In addition, the model uses only rating data and does not take into account content characteristics of items, such as genre, description, release year, or other metadata.

Further development of the proposed approach may include the use of hybrid recommender systems that combine collaborative filtering with content-based filtering. Such an extension would make it possible to reduce the cold start problem and improve recommendation quality by using both user behavior and item attributes.

Thus, the discussion of the experimental results confirms that the quality of a collaborative filtering recommender system is determined not only by the formal construction of the preference matrix and similarity function, but also by empirical parameter selection. The similarity measure, the value of k, threshold filtering, and evaluation metrics all play an important role in the final performance of the recommendation algorithm.

Conclusion.

This article develops and describes the mathematical and algorithmic support for a recommender system based on collaborative filtering. Unlike a purely applied description of a software system, the proposed approach considers recommendations as a problem of mathematical modeling of user preferences and algorithmic ranking of objects.

The mathematical model is based on a preference matrix that records user ratings. Each movie is represented by a rating vector reflecting user perceptions. Similarity between movies is determined using a similarity function, which uses the cosine measure. A k-nearest neighbors algorithm is used to find the most similar movies. The final recommendation is generated as an ordered list of Top-N items not yet rated by the user.

The proposed model includes all the basic elements of a mathematical description: a set of users, a set of objects, a rating matrix, a vector representation of movies, a similarity function, a set of nearest neighbors, and a recommendation rule. The mathematical apparatus of the model was expanded by formal definitions of the similarity function, cosine similarity, Euclidean similarity, Pearson correlation, the k-nearest neighbors selection rule, and the weighted aggregation formula for predicted ratings.

The model's algorithmic support includes the stages of data loading, preliminary filtering, preference matrix construction, similarity calculation, nearest neighbor search, candidate generation, exclusion of already rated films, and ranking of the final list.

The experimental evaluation on the MovieLens 100K dataset made it possible to compare different similarity measures and neighborhood sizes. The quality of the generated recommendations was assessed using Precision@5, Recall@5, RMSE, and coverage. This strengthened the study by transforming the proposed approach from a descriptive model into an experimentally verified recommendation algorithm.

Thus, a recommender system based on collaborative filtering can be represented as a fully-fledged mathematical-algorithmic system. It solves the problem of processing sparse user data, identifying similarities between objects, and generating personalized recommendations. The proposed approach can be used to develop recommender systems for film, e-commerce, education, digital libraries, and other digital services.

REFERENCES

Aggarwal, C.C. (2016). Recommender Systems: The Textbook. - Cham: Springer. Pp. 498. DOI: https://doi.org/10.1007/978-3-319-29659-3 [In Eng.].

Adomavicius, G., Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering. IEEE. — Vol. 17. — No. 6. Pp. 734–749. DOI: https://doi.org/10.1109/TKDE.2005.99 [In Eng.].

Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A. (2013). Recommender Systems Survey. Knowledge-Based Systems. Elsevier. — Vol. 46. — Pp. 109–132. DOI: https://doi.org/10.1016/j.knosys.2013.03.012 [In Eng.].

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments // User Modeling and User-Adapted Interaction. Springer. — Vol. 12. — Pp. 331–370. DOI: https://doi.org/10.1023/A:1021240730564 [In Eng.].

Ekstrand, M.D., Riedl, J.T., Konstan, J.A. (2011). Collaborative Filtering Recommender Systems. Foundations and Trends in Human–Computer Interaction. — Now Publishers. — Vol. 4. — No. 2. Pp. 81–173. DOI: https://doi.org/10.1561/1100000009 [In Eng.].

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T. (2004). Evaluating Collaborative Filtering Recommender Systems // ACM Transactions on Information Systems. Association for Computing Machinery. — Vol. 22. — No. 1. Pp. 5–53. DOI: https://doi.org/10.1145/963770.963772 [In Eng.].

Koren, Y., Bell, R., Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. Computer. IEEE. — Vol. 42. — No. 8. Pp. 30–37. DOI: https://doi.org/10.1109/MC.2009.263 [In Eng.].

Linden, G., Smith, B., York, J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing. IEEE. — Vol. 7. — No. 1. Pp. 76–80. DOI: https://doi.org/10.1109/MIC.2003.1167344 [In Eng.].

Ricci F., Rokach L., Shapira B. (2015). Recommender Systems Handbook. 2nd ed. — New York: Springer. Pp. 1003. DOI: https://doi.org/10.1007/978-1-4899-7637-6 [In Eng.].

Schafer J.B., Frankowski D., Herlocker J., Sen S. (2007). Collaborative Filtering Recommender Systems. The Adaptive Web. - Berlin: Springer. Pp. 291–324. DOI: https://doi.org/10.1007/978-3-540-72079-9_9 [In Eng.].

Su, X., Khoshgoftaar (2009). A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence. Hindawi. — Vol. 2009. — Pp. 1–19. DOI: https://doi.org/10.1155/2009/421425 [In Eng.].

Sarwar, B., Karypis, G., Konstan, J., Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th International Conference on World Wide Web // Association for Computing Machinery. Pp. 285–295. DOI: https://doi.org/10.1145/371920.372071 [In Eng.].

INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGIES

ХАЛЫҚАРАЛЫҚ АҚПАРАТТЫҚ ЖӘНЕ КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР ЖУРНАЛЫ

МЕЖДУНАРОДНЫЙ ЖУРНАЛ ИНФОРМАЦИОННЫХ И КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Собственник:
АО «Международный университет информационных технологий» (Казахстан, Алматы)

Главный редактор:
Колесникова Катерина Викторовна

Ответственный редактор:
Мрзабаева Раушан Жалиевна

Компьютерная верстка:
Калабай Замзагуль Ертугановна

Сайт журнала: https://journal.iitu.edu.kz
ISSN 2708–2032 (print)
ISSN 2708–2040 (online)

Подписано в печать 30.06.2026.
050040 г. Алматы, ул. Манаса 34/1, каб. 709, тел: +7 (727) 244-51-09).

Издание АО "Международного университета информационных технологий"