

EDN: VEALXR  
УДК 004.031.42

## COMPARATIVE ANALYSIS OF MATHEMATICAL ALGORITHMS FOR BUILDING RECOMMENDATION SYSTEMS

Lev N. Kene\*, Roman I. Kuzmich

Siberian Federal University, Krasnoyarsk, Russian Federation

Received 03.07.2024, accepted 29.07.2024

**Abstract.** Recommendation systems are widely used in online services to provide personalized recommendations to users, which helps to improve the user experience and increase business efficiency. This article provides a comparative analysis of the mathematical algorithms used to build recommendation systems. The article discusses the main classes of algorithms, such as collaborative filtering, content methods, hybrid approaches and algorithms based on matrix decompositions, as well as clustering methods. For each class of algorithms, there is an overview of the basic principles of research, advantages and disadvantages.

A comparative analysis of algorithms based on their accuracy, speed, scalability and ability to work with sparse data is carried out. The issue of resistance of algorithms to cold starts and ways to overcome it are also considered. Based on the analysis, it is concluded that there is no universal algorithm that would be suitable for all types of recommender system problems. The selection of the optimal algorithm depends on the specific task, data characteristics and requirements for the quality of recommendations. Further research in this area can be aimed at developing new hybrid algorithms that take into account the features of modern online services and improve the quality of recommendations, as well as the development of recommender systems built on the basis of these algorithms.

**Keywords:** recommendation systems, collaborative filtering, clustering, estimation prediction, user similarity, subject similarity, Python, SVD decomposition.

---

**Citation:** Kene, L. N., Kuzmich, R. I. (2024). Comparative analysis of mathematical algorithms for building recommendation systems. In: Trade, service, food industry. Vol. 4(4). Pp. 390–401. EDN: VEALXR

---



## СРАВНИТЕЛЬНЫЙ АНАЛИЗ МАТЕМАТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ ПОСТРОЕНИЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ

Лев Николаевич Кене\*, Роман Иванович Кузьмич

Сибирский федеральный университет, Красноярск, Российская Федерация

**Аннотация.** Рекомендательные системы широко применяются в онлайн-сервисах для предоставления персонализированных рекомендаций пользователям, что способствует улучшению пользовательского опыта и повышению эффективности бизнеса. В данной статье проведен сравнительный анализ математических алгоритмов, используемых для построения рекомендательных систем. Рассмотрены основные классы алгоритмов, такие как коллаборативная фильтрация, контентные методы, гибридные подходы и алгоритмы на основе матричных разложений, а также методы кластеризации. Для каждого класса алгоритмов осуществлен обзор основных принципов работы, преимуществ и недостатков. Выявлены общие проблемы

---

© Siberian Federal University. All rights reserved

\*Corresponding author E-mail address: lev\_kene@mail.ru

ORCID ID: 0000-0001-6017-1483 (Kuzmich)

рекомендательных систем, дан сравнительный анализ алгоритмов на основе их точности, скорости работы, масштабируемости и способности работать с разреженными данными. Также рассматривается вопрос устойчивости алгоритмов к холодному старту и способы его преодоления. На основе проведенного анализа сделан вывод о том, что не существует универсального алгоритма, который подходил бы для всех типов задач рекомендательных систем. Подбор оптимального алгоритма зависит от конкретной задачи, особенностей данных и требований к качеству рекомендаций. Дальнейшие исследования в этой области могут быть направлены на разработку новых гибридных алгоритмов, учитывающих особенности современных онлайн-сервисов и повышающих качество рекомендаций, а также на разработку рекомендательных систем, построенных на основе этих алгоритмов.

**Ключевые слова:** рекомендательные системы, коллаборативная фильтрация, кластеризация, прогнозирование оценок, подобие пользователей, подобие предметов, Python, SVD разложение.

---

**Цитирование:** Кене, Л. Н. Сравнительный анализ математических алгоритмов для построения рекомендательных систем / Л. Н. Кене, Р. И. Кузьмич // Торговля, сервис, индустрия питания. – 2024. – № 4(4). – С. 390–401. – EDN: VEALXR

---



**Введение.** Современный мир генерирует огромное количество данных, и их не всегда удается обработать и использовать в полной мере. В условиях меняющегося мира и экспоненциального роста информации рекомендательные системы становятся важным инструментом для предоставления пользователям персонализированных рекомендаций. Уже сейчас мы ежедневно взаимодействуем с рекомендательными системами, когда смотрим фильмы в онлайн-кинотеатрах или просматриваем товары на популярном маркетплейсе. Пользователям становится все труднее ориентироваться в множестве предлагаемых им вариантов. Решить эту проблему призваны рекомендательные системы. Именно они помогают пользователям сделать более взвешенный выбор, при этом экономя время и улучшая пользовательский опыт. Коммерческий интерес предприятий и компаний в повышении конверсии и выручки является также достаточно важным аспектом. Рекомендации, основанные на данных о предпочтениях и поведении пользователей, могут способствовать увеличению продаж, удержанию клиентов и повышению лояльности. Кроме того, не следует забывать о персонализации. Рекомендации, разработанные специально для каждого пользователя, могут повысить удовлетворенность и вовлеченность клиентов.

Рекомендательная система – это информационная система, предназначенная для предсказания и предоставления персонализированных рекомендаций пользователям на основе их предпочтений, поведения и интересов. Основной целью рекомендательных систем служит улучшение опыта пользователя. Достигнуть этого возможно через обеспечение пользователю релевантных и интересных рекомендаций, которые могут помочь ему с выбором продуктов, услуг или контента [1]. Задача рекомендательной системы – предсказать на основе персональных данных в профиле или некоторых статистических сведений, что может заинтересовать конкретного пользователя, какие у него потребности [2]. В теме рекомендательных систем фигурируют следующие понятия:

- пользователь/субъект – человек, использующий систему и получающий рекомендации;
- объект (предмет, товар и т.п.) – то, что рекомендуют пользователю;
- рейтинг – характеристика ценности объекта для пользователя;

– матрица рейтингов – матрица, по строкам которой находится информация о всех субъектах/пользователях, а по столбцам – о всех возможных объектах. Элементом матрицы является, например, информация о рейтинге конкретного объекта для пользователя. Если пользователь еще не оценил объект, то соответствующее значение в матрице пусто [1].

Область применения рекомендательных систем действительно широка: электронная коммерция, социальные сети, онлайн-платформы для потокового видео и музыки, новостные порталы и многое другое [3]. Иногда рекомендательные системы используют в качестве помощника при составлении плана тренировок или программы поездок (область туризма) [4].

В зависимости от задач использования выделяют несколько типов рекомендательных систем. Наиболее распространенными способами их построения признаны следующие четыре вида.

1. Коллаборативная фильтрация. Состоит, в свою очередь, из других двух фильтраций: фильтрации по пользователям и фильтрации по объектам. В коллаборативной фильтрации лежит принцип двойственного поиска: либо искать похожих пользователей – это называется «рекомендации, основанные на пользователях» (user-based collaborative filtering), либо искать похожие продукты – это называется «рекомендации, основанные на продуктах» (item-based collaborative filtering) [5]. В случае фильтрации по пользователям обозревается конкретный субъект и ему сопоставляются пользователи, схожие по вкусам, предпочтениям или поведению. На основе оценок схожих пользователей строится предсказание по поведению рассматриваемого пользователя. В случае фильтрации по объектам составляют рейтинг наиболее предпочтительных объектов – чем больше средняя оценка пользователей, тем популярнее объект. Фильтрации по пользователям и объектам основаны на истории взаимодействия пользователей и объектов, однако выделяют также коллаборативную фильтрацию, основанную на модели. К таким моделям относят кластерную модель, модель скрытых переменных, метод наименьших квадратов, линейную регрессию и байесовскую сеть [5, 6].

2. Фильтрация, основанная на контенте. Данная фильтрация базируется на истории взаимодействия пользователей и объектов, в частности на том, какой контент, какие объекты, потребляет пользователь. Такой тип рекомендательной системы широко используется в стриминговых сервисах: когда люди смотрят фильмы определенного жанра, система предлагает им другие фильмы того же жанра [7].

3. Гибридная фильтрация. Объединяет методы коллаборативной и контентно-ориентированной фильтраций. Таким образом можно избежать большинства недостатков «несмешанных» систем. Например, в онлайн-магазинах одежды в рекомендациях показываются товары, похожие на те, что вы уже видели, а также товары, которые покупали клиенты с общими вкусами [5, 7, 8].

4. Фильтрация, основанная на знаниях. Здесь рекомендации строятся на основе экспертного мнения и знаний в предметной области [8]. Такие системы подразделяют на экспертные и основанные на правилах. Этот метод включает в себя тщательный анализ предпочтений клиента, основанный на его запросах. Например, при продаже недвижимости система учитывает такие факторы, как площадь дома, количество этажей, материал стен и другие, чтобы предложить клиентам наиболее привлекательные варианты [8]. Отличие такого подтипа от вышеперечисленных – отсутствие персонализации.

Каждый из этих видов рекомендательных систем имеет свои преимущества и недостатки и может быть эффективно применен в зависимости от конкретной задачи и контекста использования.

Несмотря на популярность рекомендательных систем и удобство использования для конечного пользователя, существуют определенные проблемы, связанные с их функционированием.

Было проведено исследование проблем рекомендательных систем, где также рассмотрены пути их решения. Наиболее частая – проблема холодного старта. Для прогнозирования доверительного рейтинга необходимо достаточное количество информации о новых пользователях или субъектах [9]. Чтобы решить эту проблему, предложено использовать гибридные рекомендации, которые объединяют данные о пользователе и объекте, или активно собирать информацию о пользователе через различные интерактивные методы (например, через формы обратной связи).

Добавление новых объектов в систему может также быть проблемой: если уже есть старые объекты с оценками, то новый попросту потеряется среди низкорейтинговых. Для этого эксперты советуют искусственно корректировать рейтинг новых объектов при их добавлении [10]. Далее пользователи сами определяют, станет объект популярным или все же будет отвергнут большинством.

После внедрения рекомендательной системы ее эффективность может резко снизиться из-за нескольких факторов:

- 1) генерация излишнего количества рекомендаций;
- 2) смещение в рекомендациях к популярным объектам;
- 3) проблема недостаточной персонализации.

Эти сценарии связаны друг с другом и могут привести к большей проблеме – потере пользователей, которые в целом склонны покинуть систему из-за недовольства или других причин, а проблемы с развитием системы, указанные ранее, могут еще больше оттолкнуть их. Поэтому необходимо следить за изменениями в рекомендательной системе, предоставлять обратную связь о качестве предоставленных рекомендаций и подключать функции персонализации, учитывающие активность пользователя или контент рекомендаций, чтобы избежать потери клиентов.

Поскольку коллаборативная фильтрация имеет достаточно надежную самодостаточную историю взаимодействия, она может показаться более устойчивой к внешним факторам. Однако манипуляции пользователей системы практически делают ее бессильной. Пользователи могут обманывать систему, интересуясь только однотипной темой, игнорируя все рекомендации или намеренно завышая рейтинг объектов. Способность пользователей изменить алгоритмы, как это сделали пользователи Reddit с алгоритмами Google в 2018 году [11], демонстрирует прямую манипуляцию системой. Для решения этой проблемы можно использовать алгоритмы обнаружения мошенничества или контролировать и анализировать данные оценок и отзывов.

Таким образом, при функционировании рекомендательных систем нужны постоянный контроль и ручная настройка, а лучше автоматизация процессов, которые будут решать возникающие проблемы. Крайне важно осуществлять ежедневный мониторинг метрик эффективности системы и своевременно обновлять алгоритмы машинного обучения, фильтровать и нормализовать данные, следить за отзывами.

Матричное разложение является одним из ключевых методов, применяемых в рекомендательных системах для предсказания предпочтений пользователей. Одним из самых популярных алгоритмов матричного разложения является SVD (Singular Value Decomposition). Он разбивает исходную разреженную матрицу  $A$  на три более простые: матрицу  $U$ , матрицу  $\Sigma$  и матрицу  $V$ . Матрица  $U$  содержит левые сингулярные векторы, матрица  $\Sigma$  – диагональную матрицу с сингулярными числами, а матрица  $V$  – правые сингулярные векторы [12]. SVD разложение позволяет снизить размерность исходных данных, сохраняя при этом основные зависимости между пользователями и товарами.

Процесс работы матричного разложения в рекомендательных системах начинается с построения матрицы пользователей и товаров. Каждый пользователь оценивает определенные товары, что приводит к заполнению ячеек матрицы. Однако, матрица может быть разреженной, то есть иметь большое количество пропущенных значений. В таком случае и применяют матричное разложение для заполнения пропущенных значений и предсказания оценок, которые пользователи могли бы поставить товарам.

После разложения матрицы и получения матриц  $U$ ,  $\Sigma$  и  $V$  мы можем использовать их для предсказания рейтингов пользователей по различным товарам. В случае рекомендательных систем получается, что мы представляем каждого пользователя вектором из  $f$  факторов  $U_i$ , а каждый продукт — вектором из  $f$  факторов  $V_j$ . Далее, чтобы предсказать рейтинг пользователя  $i$  товару  $j$ , берем их скалярное произведение  $U_i V_j = U_i^T V_j$  [12].

Матричное разложение позволяет снизить вычислительную сложность алгоритма и улучшить скорость работы системы. Кроме того, матричное разложение позволяет учесть скрытые зависимости между пользователями и товарами, что может повысить точность предсказаний.

Немаловажно отметить, что работа с большими матрицами будет требовать вычислительных мощностей и может быть долгой в обучении. В рекомендательных системах применяется также следующий алгоритм – RNSA (The Refined Neighbor Selection Algorithm) [12, 13]. Этот алгоритм помогает улучшить качество рекомендаций за счет более точного определения ближайших соседей и учета их влияния на предсказания.

Процесс работы RNSA начинается с построения матрицы пользователей и товаров аналогично другим рекомендательным системам. Затем алгоритм применяет методы коллаборативной фильтрации, чтобы найти ближайших соседей для каждого пользователя. Однако, в отличие от классических методов, RNSA использует дополнительные критерии и правила для выбора соседей, что позволяет улучшить точность предсказаний.

В работе [12] представлено пять этапов алгоритма RNSA. На входе принимается тест-клиент  $t$ , входной набор данных  $S$ , а на выходе – соседи. Алгоритм следующий:

- 1) создать  $K$  кластеров из  $S$  методом кластеризации  $k$ -средних;
- 2) найти лучший кластер  $C$  для  $t$ ;
- 3) добавить  $t$  в лучший кластер  $C$  и рассматривать его как  $v$ ;
- 4) добавить  $v$  в список соседей;
- 5) если число соседей достаточно, возвращаем список соседей. В противном случае извлекаем  $v$  из  $C$ , обходим вершины (клиентов) поиском в ширину и возвращаемся к пункту 4 алгоритма.

Оба метода SVD и RNSA могут быть эффективными инструментами в создании персонализированных рекомендаций для пользователей. На практических задачах рекомендуется использовать преимущественно SVD, когда количество входных данных велико.

**Материалы и методы.** В данной работе решается задача кластеризации посредством построения рекомендательных систем методами  $k$ -ближайших соседей, матричным разложением и неотрицательным матричным разложением.

За основу данных был взят датасет MovieLens [14], который содержит в себе 25 миллионов рейтингов фильмов и представляет собой стабильный набор данных для тестирования. 25 миллионов оценок и один миллион тегов были применены к 62 000 фильмам 162 000 пользователями. Опубликован в декабре 2019 года. Так как нет

целевой переменной, а прямые зависимости скрыты, задача относится к классу обучения без учителя и является задачей кластеризации [15].

Общая постановка решаемой задачи кластеризации: по данным оценок пользователей и схожести фильмов составить рейтинг подобных для конкретного фильма.

Решение задачи кластеризации выполнено на языке программирования Python с использованием множества его библиотек. Например, с помощью библиотек NumPy и Pandas можно обобщить и обработать большие данные, для моделирования данных подойдут Scikit-learn, Statsmodels, SciPy, а для визуализации данных – Matplotlib и Seaborn [16].

Проведено сравнение трех методов построения рекомендательных систем.

Один из наиболее распространенных при определении сходства пользователей и объектов – метод k-ближайших соседей (kNN). Он основан на идее, что пользователи, которые оценили объекты одинаковым образом, склонны иметь схожие вкусы. kNN ищет k ближайших соседей пользователя или объекта и использует их оценки для предсказания оценки для неизвестного объекта или пользователя [16]. kNN используется в алгоритме RNSA [12].

Другим методом персонализированных рекомендаций служит метод альтернативных наименьших квадратов, или ALS (Alternating Least Squares), в основе которого лежит матричная факторизация (SVD разложение). Алгоритм ALS работает путем аппроксимации матриц пользователей и объектов, минимизируя среднеквадратичную ошибку между фактическими и предсказанными оценками. Он использует итеративный подход, который чередует оптимизацию по отношению к пользователям и объектам, чтобы находить оптимальные веса и уменьшать ошибку предсказаний [17, 18].

Следующий метод – один из популярных для решения задачи кластеризации – DBSCAN [19]. Метод DBSCAN (Density-Based Spatial Clustering of Applications with Noise) ищет выбросы в плотных областях пространства данных. DBSCAN использует параметры радиуса и минимального количества точек при формировании кластера для поиска областей с высокой плотностью точек, которые считаются кластерами. Принцип работы следующий: алгоритм DBSCAN находит все точки в заданном радиусе, начиная с выбора произвольной нерассмотренной точки. В этой области точка считается кластером, если количество точек достаточно для образования кластера; в противном случае точка считается выбросом. Алгоритм обнаруживает новые кластеры и продолжает расширять их для остальных нерассмотренных точек. Способность обнаруживать количество кластеров на основе данных и не требовать их заранее заданного количества является одним из основных преимуществ DBSCAN. Кроме того, DBSCAN может обрабатывать шум и выбросы [20].

Последний – агломеративная кластеризация [21] – это метод, который начинает с каждой точки данных как отдельного кластера и постепенно объединяет их в более крупные кластеры на основе сходства. На каждом шаге алгоритма два ближайших кластера объединяются, пока не останется один общий кластер. Этот процесс приводит к иерархической структуре кластеров, где каждая точка данных принадлежит определенному кластеру на разных уровнях иерархии [22]. Кроме того, этот метод позволяет обнаружить внутренние структуры данных и выделить различные уровни сходства между объектами. В контексте создания рекомендательной системы, агломеративная кластеризация может помочь группировать пользователей или предметы на основе их характеристик или поведения, что, в свою очередь, может быть использовано для персонализации рекомендаций.

В каждом из этих подходов есть как преимущества, так и недостатки. Выбор методов зависит также и от характеристики данных, цели системы и требования к точности предсказаний. Важно помнить, что сочетание различных подходов и их интерпретаций может привести к лучшим результатам, чем использование одного метода в отдельности.

Для того, чтобы оценивать то, насколько верно, точно и качественно работает система, нужны критерии и методы оценки. Можно оценить модель с точки зрения машинного обучения на основе метрик или же позволить пользователю также проводить оценку качества работы. Так как в данной работе не предполагается проведение опроса среди пользователей тестовой рекомендательной системы, для оценки работы алгоритмов в нашей модели машинного обучения, которая имитирует рекомендательную систему и решает задачу кластеризации, будем использовать следующие метрики:

– Silhouette Score – метрика качества кластеризации. Оценивает, насколько объекты внутри кластеров похожи между собой и насколько они отличаются от объектов в других кластерах. Чем ближе значение Silhouette Score к 1, тем лучше кластеризация [23].

– скорость работы модели – показатель эффективности алгоритма, который влияет на производительность системы и оптимизацию ресурсов. Быстрая скорость работы модели позволяет быстрее обрабатывать данные, делать прогнозы и принимать решения в реальном времени. Учитывая ограничения по времени и вычислительным ресурсам, оптимизация скорости работы модели является важным аспектом при разработке и использовании алгоритмов машинного обучения.

**Полученные результаты.** Рекомендательные системы в крупном бизнесе основываются на нейронных сетях, а в компаниях поменьше достаточно методов машинного обучения. Для создания эффективной рекомендательной системы необходимо учитывать не только выбор методов машинного обучения, таких как kNN, SVD и DBSCAN, но и ряд других факторов. В первую очередь, для успешной работы рекомендательной системы необходимо иметь качественные данные [24]. Это включает в себя информацию о пользователях, их предпочтениях, взаимодействии с товарами или контентом, а также описания самих товаров или текстов. Чем больше данных и чем полнее их структура, тем точнее и эффективнее будут рекомендации.

Далее важным шагом является предобработка данных, что включает в себя очистку данных от лишней информации, заполнение пропущенных значений, кодирование категориальных признаков и нормализацию данных. Чистые и подготовленные данные позволяют избежать искажений и ошибок в работе рекомендательной системы.

После этого необходимо выбрать подходящие методы машинного обучения для построения модели. Именно в методах построения модели и заключается суть создания рекомендательной системы.

В ходе исследования были построены имитационные рекомендательные системы, базирующиеся на четырех методах машинного обучения: k-ближайших соседей (kNN), SVD-разложение, DBSCAN и агломеративная кластеризация (Agglomerative Clustering). Был выбран фильм «Assasins (1995)», по нему строились рекомендуемые фильмы. Результаты полученных в виде рекомендаций фильмов, схожих с «Assasins (1995)», представлены в табл. 1.

Таблица 1. Сравнение топ-10 рекомендованных фильмов  
с рассмотренными алгоритмами  
Table 1. Comparison of the top 10 recommended films by the algorithms considered

№	kNN	SVD	DBSCAN	Agglomerative Clustering
1	Money Train (1995)	Assassins (1995)	Assassins (1995)	Across the Sea of Time (1995)
2	Assassins (1995)	Toy Story (1995)	Babe (1995)	Assassins (1995)
3	Copycat (1995)	City of Lost Children, The (Cité des enfants perdus, La) (1995)	Balto (1995)	Balto (1995)
4	Dangerous Minds (1995)	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)	Carrington (1995)	Carrington (1995)
5	GoldenEye (1995)	Dangerous Minds (1995)	City of Lost Children, The (Cité des enfants perdus, La) (1995)	City of Lost Children, The (Cité des enfants perdus, La) (1995)
6	Heat (1995)	Twelve Monkeys (a.k.a. 12 Monkeys) (1995)	Clueless (1995)	Copycat (1995)
7	Sudden Death (1995)	Wings of Courage (1995)	Copycat (1995)	Cry, the Beloved Country (1995)
8	Cutthroat Island (1995)	Babe (1995)	Dangerous Minds (1995)	Cutthroat Island (1995)
9	Ace Ventura: When Nature Calls (1995)	Carrington (1995)	Dead Presidents (1995)	Dangerous Minds (1995)
10	Casino (1995)	Dead Man Walking (1995)	Four Rooms (1995)	Dead Presidents (1995)

В ходе работы всех алгоритмов были произведены замеры времени вычисления рекомендаций в каждом алгоритме. Замеры происходили на устройстве с процессором Intel Core i7-11800H 2.3 ГГц, 32 Гб ОЗУ, SSD. Операционная система Windows 11, вычисления производились на Python версии 3.11.2 в среде Jupyter Notebook. Получены следующие результаты: kNN – 528 мс; SVD – 688 мс; DBSCAN – 2 мин 16с; Agglomerative Clustering – 157 мс.

Дополнительной метрикой оценки качества работы методов DBSCAN и Agglomerative Clustering является silhouette\_score. По расчетам средняя метрика силуэта для DBSCAN составила -0.3798, и для Agglomerative Clustering – 0.3036.

**Обсуждение.** Исходя из полученных результатов, приведенных в табл. 1, следует, что каждый алгоритм в первую очередь ищет максимально подобный фильм. Основываясь на результатах косинусного расстояния, констатируем, что наилучший результат будет у того фильма, схожесть с которым максимальна – и это сам фильм, для которого составлена рекомендация. Такие рекомендации тоже имеют место, поскольку возможно, что пользователь будет не против снова посмотреть фильм, который он уже видел.

Полученные значения средней метрики силуэта (а точнее ее отрицательное значение) говорят об обнаружении в данных кластеров, которые перекрываются или неоднородны. Это может свидетельствовать о том, что метод DBSCAN не смог эффективно выделить компактные кластеры в данных и имеет тенденцию объединять



различные кластеры в один, значит, этот метод не подходит для решения указанной задачи, ошибка определения будет высокой. Положительное же значение метрики силуэта указывает на то, что кластеры, сформированные методом Agglomerative Clustering, имеют высокую внутрикластерную плотность и хорошую разделимость между кластерами. Таким образом, Agglomerative Clustering показал более высокое качество кластеризации по сравнению с DBSCAN в данном случае.

**Выводы и дискуссионные вопросы.** В работе приведен анализ математических алгоритмов, используемых для построения рекомендательных систем. Проведен обзор способов построения рекомендательных систем, в ходе которого определено понятие и базовые концепции рекомендательных систем, приведена их классификация, а также существующие проблемы в практическом применении. Для указанных проблем приведены рекомендации по их устранению.

В работе подробно изложены идеи матричного разложения на примере метода SVD, который может применяться при работе с большими данными. Однако в ходе практических экспериментов на базе указанного метода выявлена проблема: высокие требования к объему оперативной памяти, который задействуется при вычислениях. При использовании метода k-ближайших соседей такой проблемы не возникает.

В практическом исследовании построены имитационные рекомендательные системы, основанные на методах машинного обучения. В ходе изучения методов kNN, SVD, DBSCAN и Agglomerative Clustering выявлено, что kNN и SVD местами справляются лучше, чем агломеративная кластеризация. Метод kNN в общем дает высокую точность, но плохо работает с большими данными [25], поэтому SVD предпочтительнее в работе с большими объемами данных.

На основании метрики silhouette\_score показано, что агломеративная кластеризация лучше подходит для решаемой задачи, чем DBSCAN.

В дальнейшем планируется разработка рекомендательной системы в виде приложения на базе рассмотренных в работе методов машинного обучения.

#### Библиографический список

1. Цурко, В. В. Рекомендательные системы в здравоохранении [Текст] / В. В. Цурко // Управление большими системами. – 2019. – № 82. – С. 61–73.
2. Рекомендательные системы [Электронный ресурс] // Викиконспекты ИТМО. – 2022. – URL: [https://neerc.ifmo.ru/wiki/index.php?title=Рекомендательные\\_системы](https://neerc.ifmo.ru/wiki/index.php?title=Рекомендательные_системы) (дата обращения: 11.06.2024).
3. Якупов, Д. Р. Рекомендательные системы на основе сессий – модели и задачи [Текст] / Д. Р. Якупов, Д. Е. Намиот // International Journal of Open Information Technologies. – 2022. – № 7(10). – С. 128–152.
4. Применение рекомендательных технологий в системах с пространственной информацией [Текст] / Д. С. Парыгин, А. С. Стрекалова, А. С. Гуртяков [и др.] // Прикаспийский журнал: управление и высокие технологии. – 2019. – № 1(45). – С. 96–109.
5. Филиппов, С. А. Оценка эффективности метода повышения пертинентности информации в рекомендательных системах поддержки жизнеобеспечения на основе неявных данных [Текст] / С. А. Филиппов, В. Н. Захаров // Современные информационные технологии и ИТ-образование. – 2016. – № 2(12). – С. 186–193.
6. Ларионов, В. С. Обзор методов коллаборативной фильтрации [Текст] / В. С. Ларионов, И. В. Дунин // Форум молодых ученых. – 2017. – № 5(9). – С. 1230–1238.
7. Ершов, Т. А. Рекомендательная система для изображений на основе искусственного интеллекта [Текст] / Т. А. Ершов // E-Scio. – 2023. – № 8(83). – С. 1–7.
8. Что такое рекомендательные системы и как они работают [Электронный

ресурс // Skillbox Media. – 2021. – URL: [https://skillbox.ru/media/code/chto\\_takoe\\_rekomendatelnye\\_sistemy\\_i\\_kak\\_oni\\_rabotayut/](https://skillbox.ru/media/code/chto_takoe_rekomendatelnye_sistemy_i_kak_oni_rabotayut/) (дата обращения: 10.06.2024).

9. Рекомендательные системы: что это и как работает алгоритм рекомендаций [Электронный ресурс]. – 2022. – URL: <https://mindbox.ru/journal/education/rekomendatelnye-sistemy/> (дата обращения: 27.05.2024).

10. Рекомендательные системы: как алгоритмы подбирают нам товары и контент [Электронный ресурс]. – 2020. – URL: <https://hub.forklog.com/rekomendatelnye-sistemy-kak-algoritmy-podbirayut-nam-tovary-i-kontent/> (дата обращения: 30.05.2024).

11. Что такое рекомендательные системы? [Электронный ресурс]. – 2022. – URL: <https://forklog.com/criptorium/ai/chto-takoe-rekomendatelnye-sistemy> (дата обращения: 01.06.2024).

12. Королева, Д. Е. Анализ алгоритмов обучения коллаборативных рекомендательных систем [Текст] / Д. Е. Королева, М. В. Филиппов // Инженерный журнал: наука и инновации. – 2013. – № 6(18). – С. 1–8.

13. Kim, T. H. A Refined Neighbor Selection Algorithm for Clustering-Based Collaborative Filtering [Text] / T. H. Kim, S. B. Yang // The KIPS Transactions: PartD. – 2007. – № 14(3). – Pp. 347–354.

14. MovieLens [Электронный ресурс]. – 2024. – URL: <https://grouplens.org/datasets/movielens/> (дата обращения: 30.05.2024).

15. Как работают рекомендательные системы [Электронный ресурс]. – 2019. – URL: <https://neurohive.io/ru/osnovy-data-science/rekomendatelnye-sistemy-modeli-i-ocenka/> (дата обращения: 10.06.2024).

16. 15 библиотек Python для Data Science [Электронный ресурс]. – 2024. – URL: <https://blog.skillfactory.ru/15-bibliotek-python-dlya-data-science/> (дата обращения: 12.06.2024).

17. Hastie, T. Matrix completion and low-rank SVD via fast alternating least squares [Text] / T. Hastie, R. Mazumder, J. D. Lee // The Journal of Machine Learning Research. – 2015. – № 16(1). – P. 3367–3402.

18. Takács, G. Alternating least squares for personalized ranking [Text] / G. Takács, D. Tikk // Proceedings of the 6th ACM conference on Recommender systems (18–23 September 2022), Seattle. – Pp. 83–90.

19. Deng, D. DBSCAN clustering algorithm based on density [Text] / D. Deng // Proceedings of 7th international forum on electrical engineering and automation (25–27 September 2020), China. – Pp. 949–953.

20. Жилов, Р. А. Интеллектуальные методы кластеризации данных [Текст] / Р. А. Жилов // Известия КБНЦ РАН. – 2023. – № 6(116). – С. 152–159.

21. Tokuda, E. K. Revisiting agglomerative clustering [Text] / E. K. Tokuda, C. H. Comin, L. D. F. Costa // Physica A: Statistical mechanics and its applications. – 2022. – № 585. – P. 126433.

22. Булыга, Ф. С. Алгоритмы агломеративной кластеризации применительно к задачам анализа лингвистической экспертной информации [Текст] / Ф. С. Булыга, В. М. Курейчик // Известия ЮФУ. Технические науки. – 2021. – № 6(223). – С. 73–88.

23. Оценка производительности кластеризации в Scikit Learn [Электронный ресурс]. – 2024. – URL: <https://ru.linux-console.net/?p=25753> (дата обращения: 12.06.2024).

24. Таратухина, Ю. В. Машинное обучение модели информационной рекомендательной системы по вопросам индивидуализации образования [Текст] / Ю. В. Таратухина, Т. В. Барт, В. В. Власов // Образовательные ресурсы и технологии. – 2019.

– № 2(27). – С. 7–14.

25. A Survey and Classification on Recommendation Systems [Text] / M. Sharma, R. Mittal, A. Bharati [and et. al.] // Proceedings of 2nd International Conference on Big Data, Machine Learning and Applications (19–20 December 2021), Kurukshetra, 2022. – Pp. 1–16.

### References

1. Tsurko, V.V. (2019). Recommender systems in healthcare. Managing large systems, 82, 61–73.

2. Recommender systems [Electronic source] // ITMO Wikinotes. – 2022. – URL: [https://neerc.ifmo.ru/wiki/index.php?title=%D0%A0%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5\\_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B](https://neerc.ifmo.ru/wiki/index.php?title=%D0%A0%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) (Date of access: 11.06.2024).

3. Yakupov, D. R., Namiot, D. E. (2022). Session-based recommender systems – models and tasks. International Journal of Open Information Technologies, 7(10), 128–152.

4. Parygin, D. S., Strekalova, A. S., Gurtyakov, A. S., et. al. (2019). Application of recommendation technologies in systems with spatial information. Caspian Journal: Management and High Technologies, 1(45), 96–109.

5. Filippov, S. A., Zakharov, V. N. (2016). Assessing the effectiveness of a method for increasing the persistence of information in life support recommendation systems based on implicit data. Modern information technologies and IT education, 2(12), 186–193.

6. Larionov, V. S., Dunin, I. V. (2017). Review of collaborative filtering methods. Forum of young scientists, 5(9), 1230–1238.

7. Ershov, T. A. (2023). Recommendation system for images based on artificial intelligence. E-Scio, 8(83), 1–7.

8. What are recommender systems and how do they work? Skillbox Media. [Electronic source] URL: [https://skillbox.ru/media/code/chto\\_takoe\\_rekomendatelnye\\_sistemy\\_i\\_kak\\_oni\\_rabotayut/](https://skillbox.ru/media/code/chto_takoe_rekomendatelnye_sistemy_i_kak_oni_rabotayut/) (Date of access: 10.06.2024).

9. Recommender systems: what they are and how the recommendation algorithm works [Electronic source] URL: <https://mindbox.ru/journal/education/rekomendatelnye-sistemy/> (Date of access: 27.05.2024).

10. Recommendation systems: how algorithms select products and content for us. [Electronic source]. URL: <https://hub.forklog.com/rekomendatelnye-sistemy-kak-algoritmy-podbirayut-nam-tovary-i-kontent/> (Date of access: 30.05.2024).

11. What are recommender systems? [Electronic source] URL: <https://forklog.com/cryptorium/ai/chto-takoe-rekomendatelnye-sistemy> (Date of access: 01.06.2024).

12. Koroleva, D. E., Filippov, M. V. (2013). Analysis of learning algorithms for collaborative recommender systems. Engineering journal: science and innovation, 6(18), 1–8.

13. Kim, T. H., Yang, S. B. (2007). A Refined Neighbor Selection Algorithm for Clustering-Based Collaborative Filtering. The KIPS Transactions: PartD, 14(3), 347–354.

14. MovieLens. [Electronic source] URL: <https://grouplens.org/datasets/movielens/> (Date of access: 30.05.2024).

15. How recommender systems work. [Electronic source] URL: <https://neurohive.io/ru/osnovy-data-science/rekomendatelnye-sistemy-modeli-i-ocenka/> (Date of access: 10.06.2024).

16. 15 libraries Python for Data Science. [Electronic source] URL: <https://blog.skillfactory.ru/15-bibliotek-python-dlya-data-science/> (Date of access: 12.06.2024).

17. Hastie, T., Mazumder R., Lee, J. D. (2015). Matrix completion and low-rank SVD via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1), 3367–3402.
18. Takács, G., Tikk, D. (2022). Alternating least squares for personalized ranking. *Proceedings of the 6th ACM conference on Recommender systems*, 83–90.
19. Deng, D. (2020). DBSCAN clustering algorithm based on density. *Proceedings of 7th international forum on electrical engineering and automation*, 949–953.
20. Zhilov, R. A. (2023). Intelligent methods of data clustering. *News of the KBSC RAS*, 6(116), 152–159.
21. Tokuda, E. K., Comin, C. H., Costa, L. D. F. (2022). Revisiting agglomerative clustering. *Physica A: Statistical mechanics and its applications*, 585, 126433.
22. Bulyga, F. S., Kureichik, V. M. (2021). Agglomerative clustering algorithms in relation to problems of analysis of linguistic expert information. *Izvestia SFU. Technical science*, 6(223), 73–88.
23. Evaluation of clustering performance in Scikit Learn. [Electronic source] URL: <https://ru.linux-console.net/?p=25753> (Date of access: 12.06.2024).
24. Taratukhina, Yu. V., Bart, T. V., Vlasov, V. V. (2019). Machine learning model of an information recommendation system on issues of individualization of education. *Educational resources and technologies*, 2(27), 7–14.
25. Sharma, M., Mittal, R., Bharati, A., et. al. (2022). A Survey and Classification on Recommendation Systems. *Proceedings of 2nd International Conference on Big Data, Machine Learning and Applications*, 1–16.