

Научная статья  
УДК 004.5:519.81  
<https://doi.org/10.35266/1999-7604-2024-4-4>



## Нейросетевая рекомендательная система по подбору контента для онлайн-кинотеатров

Ксения Евгеньевна Кожихова<sup>1✉</sup>, Дмитрий Викторович Тараканов<sup>2</sup>,  
Иван Вацлавович Чалей<sup>3</sup>

<sup>1,2</sup>Сургутский государственный университет, Сургут, Россия

<sup>3</sup>ПАО «Сургутнефтегаз», Сургут, Россия

<sup>1</sup>[kozikhova\\_ksenia@mail.ru](mailto:kozikhova_ksenia@mail.ru)<sup>✉</sup>, <https://orcid.org/0009-0003-5564-2027>

<sup>2</sup>[sprtdv@mail.ru](mailto:sprtdv@mail.ru), <https://orcid.org/0000-0003-1851-1039>

<sup>3</sup>[ivan@chaley.ru](mailto:ivan@chaley.ru), <https://orcid.org/0000-0001-5274-4296>

**Аннотация.** В статье проведен сравнительный анализ по таким критериям оценки качества, как нагрузка на систему, объем потребляемой оперативной памяти, время, затраченное на обучение, RMSE (корень из среднеквадратичной ошибки), MAE (средняя абсолютная ошибка), FCP (процент прогнозируемого покрытия) и MSE (среднеквадратичная ошибка). С помощью математических методов принятия решений был выбран самый практичный алгоритм – система на архитектуре DLRM, которая продемонстрировала наилучшие результаты по точности и гибкости в обработке большого объема данных, несмотря на высокую ресурсозатратность. Реализация выбранного алгоритма включала разработку самого алгоритма, проектирование базы данных, создание графического интерфейса пользователя и разработку.

**Ключевые слова:** рекомендательные системы, анализ данных, машинное обучение, математические методы принятия решений, нейронные сети, база данных, графический интерфейс, программный интерфейс

**Для цитирования:** Кожихова К. Е., Тараканов Д. В., Чалей И. В. Нейросетевая рекомендательная система по подбору контента для онлайн-кинотеатров // Вестник кибернетики. 2024. Т. 23, № 4. С. 34–52. <https://doi.org/10.35266/1999-7604-2024-4-4>.

Original article

## Neural network-based recommendation system for content selection in online movie theatres

Kseniya E. Kozhikhova<sup>1✉</sup>, Dmitry V. Tarakanov<sup>2</sup>, Ivan V. Chaley<sup>3</sup>

<sup>1,2</sup>Surgut State University, Surgut, Russia

<sup>3</sup>“Surgutneftegas” PJSC, Surgut, Russia

<sup>1</sup>[kozikhova\\_ksenia@mail.ru](mailto:kozikhova_ksenia@mail.ru)<sup>✉</sup>, <https://orcid.org/0009-0003-5564-2027>

<sup>2</sup>[sprtdv@mail.ru](mailto:sprtdv@mail.ru), <https://orcid.org/0000-0003-1851-1039>

<sup>3</sup>[ivan@chaley.ru](mailto:ivan@chaley.ru), <https://orcid.org/0000-0001-5274-4296>

**Abstract.** The article describes a comparative analysis using the following quality assessment criteria: system load, RAM consumption, time spent on training, RMSE (Root Mean Square Error), MAE (Mean Absolute Error), FCP (First Contentful Paint) and MSE (Mean Square Error). Using mathematical decision-making methods, the most practical algorithm was selected, a system based on DLRM (Deep Learning Recommendation Model) architecture, which showed the best results in terms of accuracy and flexibility in processing large amounts of data, despite having high resource intensiveness. The implementation of the selected algorithm included the development of the algorithm itself, database design, creation of a graphical user interface and development.

**Keywords:** recommendation systems, data analysis, machine learning, decision-making mathematical methods, neural networks, database, graphical interface, program interface

**For citation:** Kozhikhova K. E., Tarakanov D. V., Chaley I. V. Neural network-based recommendation system for content selection in online movie theatres. *Proceedings in Cybernetics*. 2024;23(4):34–52. <https://doi.org/10.35266/1999-7604-2024-4-4>.

## ВВЕДЕНИЕ

Разработка алгоритмов рекомендаций, которые учитывают индивидуальные предпочтения и поведенческие особенности пользователей, позволяет значительно повысить удовлетворенность пользователей и увеличить их лояльность к платформе. Данная тема особенно актуальна в контексте конкурентной борьбы между различными онлайн-кинотеатрами, где качественные рекомендации могут стать ключевым преимуществом. Помимо этого, успешная рекомендательная система позволит онлайн-кинотеатрам не только повысить удовлетворенность пользователей, но и улучшить такие бизнес-показатели, как удержание аудитории и время просмотра.

Целью данной работы является разработка наиболее эффективной рекомендательной системы по подбору видеоконтента для онлайн-кинотеатров, основанной на сравнительном анализе различных методов и алгоритмов рекомендаций.

Рассмотрим несколько примеров успешного использования рекомендательных систем в онлайн-сервисах.

1. Netflix – крупнейшая развлекательная платформа, которая реализовала оригинальную версию своей системы рекомендаций в далеком 2000 г. [1]. В настоящее время Netflix использует продвинутую систему персонализированных рекомендаций, которая анализирует историю просмотров, рейтинг и поисковое поведение пользователя, чтобы предложить фильмы и телешоу, которые, скорее всего, ему понравятся. Алгоритм также учитывает жанр, актеров, режиссера и другие факторы, чтобы составить персональные рекомендации для каждого пользователя [2].

По оценкам Netflix, около 80% общего времени просмотра видеоконтента приходится на систему рекомендаций, что является впечатляющим результатом [3].

2. Еще одним примером успешной реализации рекомендательной системы является

Amazon. По оценкам, около 35% доходов Amazon приходится на систему рекомендаций [4].

Система рекомендаций Amazon предлагает продукты на основе истории покупок, поиска и поведения пользователя в приложении или на сайте. Она создает персонализированные рекомендации на основе предыдущих покупок пользователя, просмотренных товаров и товаров, добавленных в корзину [2].

Несмотря на разнообразие и гибкость методов, рекомендательные системы сталкиваются с серьезными проблемами, которые существенно влияют на качество моделей и усложняют процесс их разработки. Рассмотрим несколько из этих проблем [5].

### 1. Холодный старт

Это ситуация, когда система не имеет достаточной информации для генерации точных рекомендаций. Могут возникнуть такие ситуации, когда новый пользователь регистрируется на платформе, система не знает его предпочтений, в результате рекомендации могут быть неточными или общими, или, когда добавляется новый продукт или контент, система не имеет данных о его популярности и релевантности для различных пользователей.

### 2. Масштабируемость

Речь идет о способности системы обрабатывать и анализировать большие объемы данных, так как увеличение количества пользователей и объектов может существенно увеличить время выдачи рекомендаций.

### 3. Разреженность данных

Такая проблема возникает, когда большинство пользователей оценивают лишь небольшой процент доступного контента, что приводит к недостаточности данных и сложности в обучении модели.

### 4. Смещение данных

Данные могут быть смещены или неполными. Например, активные пользователи могут оставлять больше отзывов, что создает

неравномерное представление о предпочтениях всей аудитории.

### 5. Фильтрационный пузырь

Пользователи могут постоянно получать рекомендации на основе своих прошлых предпочтений, что препятствует открытию для них нового контента.

## МАТЕРИАЛЫ И МЕТОДЫ

В рамках данной работы для обучения рекомендательных систем были использованы данные с веб-сайта MovieLens: 25 млн оценок, примененных к 62 тысячам фильмов 162 тысяч пользователей.

Данные представляют собой следующий набор из 4 файлов.

1. Links.csv – файл, в котором представлено соответствие id фильмов из базы MovieLens с их id в крупнейших базах данных о фильмах imdb.com и tmdb.com. Имеет поля:

- movieId – id фильма;
- imdbId – id фильма на imdb.com;
- tmdbId – id фильма на tmdb.com.

2. Users.csv – файл с информацией о пользователях, который содержит поля:

- userId – id пользователя;
- age – возраст;
- sex – пол.

Эти данные позволяют значительно повысить точность и релевантность рекомендаций, предлагаемых пользователям.

3. Movies.csv – файл с фильмами, который содержит в себе следующую информацию:

- movieId – id фильма;
- title – название фильма, в скобках после названия указана дата выхода;
- genres – жанры фильма, указаны через разделитель.

Пример данных представлен в табл. 1.

В файле представлено более 62 тысяч фильмов, которым присвоены 19 уникальных

жанров. Как видно из табл. 1, одному фильму может быть присвоено несколько жанров.

Построим диаграмму, показывающую распределение фильмов по жанрам, которая представлена на рис. 1.

Рис. 1 наглядно иллюстрирует, что ведущими жанрами являются драма и комедия, они встречаются гораздо чаще других жанров. Анализ этих данных также помогает лучше понять аудиторию и настроить алгоритмы рекомендаций для повышения их точности и релевантности.

Отсортируем фильмы по году выпуска и построим диаграмму, которая отражена на рис. 2.

Как видно из рис. 3, преимущественно в базе MovieLens присутствуют фильмы 2000–2010 гг. выпуска. Знание периодов наибольшего производства видеоконтента позволяет лучше учитывать временные предпочтения пользователей и предлагать более релевантные рекомендации.

4. Ratings.csv – файл с оценками фильмов, которые поставили пользователи. Содержит следующие поля:

- userId – id пользователя;
- movieId – id фильма;
- rating – оценка, поставленная пользователем данному фильму;
- timestamp – время, когда была поставлена оценка фильму, представленное в формате UNIX timestamp (количество секунд, прошедших с 01.01.1970).

Пример входных данных представлен в табл. 2.

Диапазон выставяемых оценок варьируется от 1,0 до 5,0 с шагом 0,5.

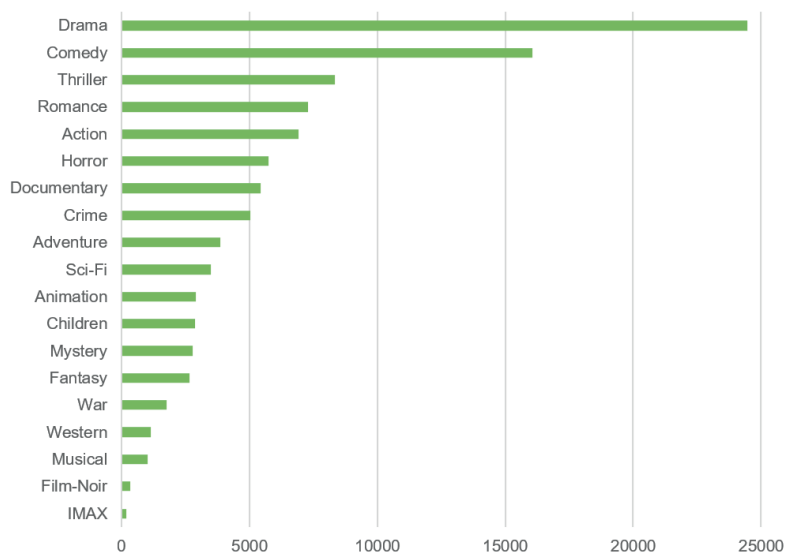
Данный файл является основополагающим в обучении методов и дальнейшей реализации рекомендательной системы, поэтому рассмотрим его подробнее. В ре-

Таблица 1

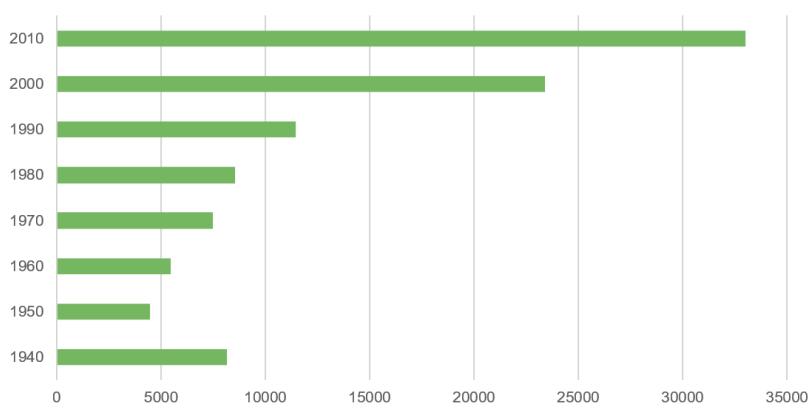
Пример входных данных movies.csv

movieId	title	genres
1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy
8957	Saw (2004)	Horror Mystery Thriller
33615	Madagascar (2005)	Adventure Animation Children Comedy

Примечание: составлено авторами.



**Рис. 1. Диаграмма распределения количества фильмов по жанрам**  
 Примечание: составлено авторами.



**Рис. 2. Диаграмма распределения фильмов по годам выпуска**  
 Примечание: составлено авторами.

зультате анализа получаем следующую информацию:

- минимальное количество оценок, поставленное одним пользователем: 20;
- медианное количество оценок, поставленное одним пользователем: 71,0;
- среднее количество оценок, поставленное одним пользователем: 153,80793153727367;

- максимальное количество оценок, поставленное одним пользователем: 32202;
- более 75% пользователей поставили менее 162 оценок;
- максимальное количество оценок на один фильм: 81491;
- медианное количество оценок на один фильм: 60;

Таблица 2

Пример входных данных rating.cvs

userId	movieId	rating	timestamp
5183	183309	5,0	1530336311
5183	183833	2,5	1546327827
5183	184619	4,0	1523971175

Примечание: составлено авторами.

– количество фильмов с одной оценкой: 10298;

– более 75% фильмов имеют менее 36 оценок.

Данные показатели не являются оптимальными из-за высокой разреженности данных, что может негативно сказаться на точности прогнозирования.

Обучение методов было реализовано на языке программирования Python 3, который является одним из наиболее популярных языков для анализа данных и машинного обучения благодаря своей простоте и богатой экосистеме библиотек [6]. Также для обучения были применены библиотеки:

– numpy – библиотека для работы с многомерными массивами и матрицами, а также для выполнения различных математических операций [7];

– pandas – библиотека для обработки и анализа данных, которая обеспечивает удобные инструменты для работы с таблицами и временными рядами [8];

– sklearn – библиотека машинного обучения, включающая множество алгоритмов для классификации, регрессии, кластеризации и снижения размерности [9];

– keras – высокоуровневый API для нейронных сетей, работающий поверх TensorFlow, который упрощает создание и обучение глубоких нейронных сетей [10];

– surprise – специализированная библиотека для построения и анализа рекомендательных систем [11];

– csv – модуль для работы с CSV-файлами, позволяет читать и записывать данные в формате CSV [12];

– os – модуль для взаимодействия с операционной системой, обеспечивает доступ к файловой системе и позволяет выполнять системные команды [13];

– time – модуль для работы со временем, предоставляет функции для измерения времени, задержек и форматирования времени [14];

– psutil – библиотека для получения информации о системных ресурсах и процессах, используется для мониторинга производительности [15].

Обучение каждого метода можно разделить на три этапа:

1. Импорт библиотек для сбора критериев оценки качества и подготовки директории с файлами для обучения модели.

2. Индивидуальное обучение каждой модели, тестирование модели на тестовом наборе данных для получения значений критериев оценки качества.

3. Сохранение значений критериев, полученных на втором этапе, в csv-файлы для дальнейшего анализа.

Для каждого метода были собраны результаты работы метода:

- на неотсортированном наборе данных;
- на отсортированном наборе данных по возрасту и полу;
- на отсортированном наборе данных по жанру;
- на отсортированном наборе данных по жанру, возрасту и полу.

После получения всех результатов составлена результирующая таблица для каждого алгоритма.

Аппаратная часть, на которой были обучены алгоритмы рекомендательных систем:

процессор: Intel core i5-12400F, 6 ядер, 12 потоков;

оперативная память: ADATA 32GB DDR4.

Рассмотрим общие критерии оценки качества для каждого метода с целью дальнейшего выбора наиболее оптимального алгоритма рекомендательной системы.

Итоговый результат обучения метода Co-Clustering продемонстрирован в табл. 3.

Как видно из результирующей табл. 3, данный метод быстро обучился, при этом не используя большие ресурсы персонального компьютера.

Итоговый результат обучения метода KNNBaseline продемонстрирован в табл. 4.

Как видно из результирующей табл. 4, алгоритм достаточно быстро обучился, но ему потребовалось более 4 Гб оперативной памяти в среднем (17 Гб в максимуме).

Итоговый результат обучения метода KNNBasic продемонстрирован в табл. 5.

Как видно из результирующей табл. 5, алгоритм достаточно быстро обучился, но ему по-

Таблица 3

**Общий результат обучения алгоритма Co-Clustering**

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	15,30	134213632	2156,0000	0,9419	0,7311	0,5899	0,8872
По полу и возрасту	17,00	135294976	164,6500	0,9771	0,7568	0,5976	0,9555
По жанру	20,70	136196096	275,5400	0,9292	0,7152	0,6199	0,8643
По жанру, возрасту и полу	20,00	136228864	235,0182	0,9448	0,7266	0,5992	0,8942
Среднее общее	18,25	135483392	707,8021	0,9483	0,7324	0,6017	0,9003

Примечание: составлено авторами.

Таблица 4

**Общий результат обучения алгоритма KNNBaseline**

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	14,300	17231249415	1661,0000	0,8878	0,6802	0,6196	0,7882
По полу и возрасту	18,100	119967744	124,8583	0,9013	0,6903	0,6217	0,8126
По жанру	18,500	121020416	69,7937	0,8653	0,6520	0,6267	0,7494
По жанру, возрасту и полу	19,600	119853056	57,3000	0,8805	0,6637	0,6044	0,7771
Среднее общее	17,625	4398022658	478,2380	0,8837	0,6716	0,6181	0,7818

Примечание: составлено авторами.

Таблица 5

**Общий результат обучения алгоритма KNNBasic**

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	18,200	21971861513	1559,0000	0,9607	0,7445	0,4494	0,923
По полу и возрасту	18,800	123564032	124,1000	0,9698	0,7492	0,4880	0,9407
По жанру	19,800	123142144	64,8079	0,9937	0,7082	0,5631	0,8688
По жанру, возрасту и полу	36,900	122216448	53,3833	0,9458	0,7181	0,5338	0,8971
Среднее общее	23,425	5585196034	450,3228	0,9675	0,7300	0,5086	0,9074

Примечание: составлено авторами.

требовалось более 20% ЦП и более 5 Гб оперативной памяти в среднем (21 Гб в максимуме).

Итоговый результат обучения метода KNNWithZScore продемонстрирован в табл. 6.

Как видно из результирующей табл. 6, алгоритм достаточно быстро обучился, но ему потребовалось более 5 Гб оперативной памяти в среднем (20 Гб в максимуме).

Итоговый результат обучения метода NMF продемонстрирован в табл. 7.

Как видно из результирующей табл. 7, алгоритм быстро обучился, но ему потребовалось более 20% ЦП и более 4 Гб оперативной памяти в среднем (13 Гб в максимуме).

Итоговый результат обучения метода SlopeOne продемонстрирован в табл. 8.

Как видно из результирующей табл. 8, алгоритму для полного обучения на неотсор-

тированном наборе данных потребовалось 1,5 часа, при этом было затрачено более 10 Гб оперативной памяти (максимум), нагрузка на ЦП была в пределах нормы.

Итоговый результат обучения метода k-Means продемонстрирован в табл. 9.

Из результирующей табл. 9 видно, что, как и предыдущему алгоритму, алгоритму k-Means потребовалось время для полного обучения. В максимуме было использовано около 21 Гб оперативной памяти, но ЦП при этом не был сильно нагружен.

Итоговый результат обучения метода SVD продемонстрирован в табл. 10.

Как видно из результирующей табл. 10, алгоритм достаточно быстро обучился, не затратив больших ресурсов персонального компьютера.

Таблица 6

**Общий результат обучения алгоритма KNNWithZScore**

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	15,800	20887633928	1651,0000	0,9146	0,6956	0,5939	0,8366
По полу и возрасту	19,600	119324672	123,2833	0,9384	0,7133	0,6188	0,8812
По жанру	19,600	118132736	68,8753	0,8740	0,6579	0,6319	0,7650
По жанру, возрасту и полу	22,700	124276736	55,2333	0,9071	0,6832	0,6072	0,8272
Среднее общее	19,425	5312342018	474,5980	0,9085	0,6875	0,6130	0,8275

Примечание: составлено авторами.

Таблица 7

**Общий результат обучения алгоритма NMF**

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	15,0	1304428544	1791,0000	0,9371	0,7211	0,5991	0,8781
По полу и возрасту	17,3	142761984	153,0333	0,9604	0,7378	0,6142	0,9237
По жанру	25,8	142954496	251,5684	0,9143	0,6965	0,6253	0,8370
По жанру, возрасту и полу	26,3	143896576	188,7250	0,9400	0,7181	0,6024	0,8878
Среднее общее	21,1	433510400	596,0817	0,9380	0,7184	0,6103	0,8817

Примечание: составлено авторами.

Таблица 8

**Общий результат обучения алгоритма SlopeOne**

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	15,10	11318329348	5950,000	0,898	0,6899	0,6133	0,8063
По полу и возрасту	16,40	145141760	393,1000	0,9358	0,7161	0,6151	0,8764
По жанру	16,10	149635072	217,0853	0,8782	0,6645	0,6369	0,7720
По жанру, возрасту и полу	23,00	147943424	175,5417	0,9075	0,6881	0,6057	0,8275
Среднее общее	17,65	2940262401	1683,9320	0,9049	0,6897	0,6178	0,8206

Примечание: составлено авторами.

Таблица 9

**Общий результат обучения алгоритма k-Means**

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	14,90	22265462793	6032,0000	0,9181	0,6964	0,6136	0,843
По полу и возрасту	17,00	417189888	125,6083	0,9399	0,7150	0,6203	0,8841
По жанру	18,60	417157120	68,4605	0,8736	0,6580	0,6368	0,7642
По жанру, возрасту и полу	25,30	417652736	54,6083	0,9039	0,6810	0,6095	0,8212
Среднее общее	18,95	5879365634	1570,1690	0,9089	0,6876	0,6201	0,8281

Примечание: составлено авторами.

Таблица 10

**Общий результат обучения алгоритма SVD**

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	14,000	13481541632	1374,0000	0,8712	0,6703	0,6282	0,7591
По полу и возрасту	18,800	107778048	96,2667	0,8932	0,6858	0,6155	0,7982
По жанру	17,900	133238784	143,2632	0,8602	0,6589	0,6272	0,7407
По жанру, возрасту и полу	15,200	144293888	116,6667	0,8673	0,6635	0,6026	0,7539
Среднее общее	16,475	3466713088	432,5492	0,8730	0,6696	0,6184	0,7630

Примечание: составлено авторами.

Итоговый результат обучения метода DLRM продемонстрирован в табл. 11.

Как видно из результирующей табл. 11, алгоритм очень долго обучался (максимальное значение – более 3-х дней), затратив при этом большие ресурсы ЦП, но не используя больше 2 Гб оперативной памяти.

Итоговый результат обучения гибридного метода k-Means + SVD продемонстрирован в табл. 12.

Как видно из результирующей табл. 12, алгоритм долго обучался (максимально – больше суток), затратив в среднем 6 Гб оперативной памяти (22 Гб в максимуме).

Итоговый результат обучения гибридного метода KNNBasic + k-Means продемонстрирован в табл. 13.

Как видно из результирующей табл. 13, алгоритм обучался долго (максимально – больше 1,5 дня), затратив в среднем 6 Гб оперативной памяти (22 Гб в максимуме).

Итоговый результат обучения гибридного метода KNNBasic + SVD продемонстрирован в табл. 14. Как видно из результирующей табл. 14, алгоритм обучался долго (максимально 1,5 дня), используя при этом около 30% ЦП и 6 Гб памяти в среднем (22 Гб в максимуме). Данный метод достаточно требователен по ресурсам.

В табл. 15 приведен общий результат, взятый из строк «Среднее общее» в табл. 3–14, для всех обученных методов выше.

Как видно из табл. 15, алгоритмы все разные, имеют разные значения критериев и нет явного лидера по производительности и результату.

Таблица 11

Общий результат обучения алгоритма DLRM

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	91,70	1616904192	249894,0000	0,9773	0,739	0,5805	0,9551
По полу и возрасту	54,60	1480040448	1457,1250	0,9779	0,7452	0,5744	0,9565
По жанру	58,90	1447968768	2593,2074	0,9296	0,7038	0,5889	0,8655
По жанру, возрасту и полу	82,20	1341693952	1947,9750	0,9472	0,7211	0,5790	0,9009
Среднее общее	71,85	1471651840	63973,0800	0,9580	0,7273	0,5807	0,9195

Примечание: составлено авторами.

Таблица 12

Общий результат обучения алгоритма k-Means + SVD

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	18,7	22271754249	95095,0000	0,8770	0,6742	0,6197	0,7691
По полу и возрасту	18,8	1527545856	857,4000	0,8925	0,6852	0,6282	0,7967
По жанру	19,9	1530687488	1411,6689	0,8455	0,6451	0,6332	0,7154
По жанру, возрасту и полу	19,8	1528791040	965,3833	0,8631	0,6583	0,6046	0,7474
Среднее общее	19,3	6714694658	24582,3600	0,8695	0,6657	0,6214	0,7572

Примечание: составлено авторами.

Таблица 13

Общий результат обучения алгоритма KNNBasic + k-Means

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	17,2	22938648585	139540	0,9098	0,7005	0,5928	0,8277
По полу и возрасту	19,60	403456000	929,3917	0,9218	0,7077	0,6019	0,8500
По жанру	18,20	1521176576	1334,4958	0,8832	0,6689	0,6266	0,7811
По жанру, возрасту и полу	24,80	1527468032	990,3250	0,9020	0,6832	0,5998	0,8159
Среднее общее	19,95	6597687298	35698,5500	0,9042	0,6901	0,6053	0,8187

Примечание: составлено авторами.

Таблица 14

Общий результат обучения алгоритма KNNBasic + SVD

Набор данных	ЦП, %	Память, байт	Время обучения, с	RSME	MAE	FCP	MSE
Неотсортированный	18,700	22271754249	95095,0000	0,8920	0,6896	0,5927	0,7957
По полу и возрасту	26,200	414904320	956,4750	0,9076	0,7005	0,6006	0,8238
По жанру	34,200	419827712	1535,0605	0,8652	0,6630	0,6191	0,7494
По жанру, возрасту и полу	36,000	420458496	1141,7750	0,8762	0,6698	0,5964	0,7693
Среднее общее	28,775	5881736194	24682,0800	0,8853	0,6807	0,6022	0,7846

Примечание: составлено авторами.

Таблица 15

Общие показатели обучения всех методов

Алгоритм рек. системы	Критерии						
	Y1	Y2	Y3	Y4	Y5	Y6	Y7
SVD	<b>16,475</b>	3466713088	<b>432,5492</b>	0,8730	0,6696	0,6184	0,7630
Co-Clustering	18,250	<b>135483392</b>	707,8021	0,9483	<b>0,7324</b>	0,6017	0,9003
KNNBaseline	17,625	4398022658	478,2380	0,8837	0,6716	0,6181	0,7818
KNNBasic	23,425	5585196034	450,3228	<b>0,9675</b>	0,7300	<b>0,5086</b>	0,9074
KNNWithZScore	19,425	5312342018	474,5980	0,9085	0,6875	0,6130	0,8275
NMF	21,100	433510400	596,0817	0,9380	0,7184	0,6103	0,8817
SlopeOne	17,650	2940262401	1683,9320	0,9049	0,6897	0,6178	0,8206
DLRM	<b>71,850</b>	1471651840	<b>63973,0800</b>	0,9580	0,7273	0,5807	<b>0,9195</b>
k-Means	18,950	5879365634	1570,1690	0,9089	0,6876	0,6201	0,8281
KNNBasic + k-Means	19,950	6597687298	35698,5500	0,9042	0,6901	0,6053	0,8187
KNNBasic + SVD	28,775	5881736194	24682,0800	0,8853	0,6807	0,6022	0,7846
k-Means + SVD	19,300	<b>6714694658</b>	24582,3600	<b>0,8695</b>	<b>0,6657</b>	<b>0,6214</b>	<b>0,7572</b>

Примечание: составлено авторами.

Для выбора наиболее оптимального алгоритма рекомендательной системы используем математические методы принятия решений.

Воспользуемся комбинированным критерием для выбора алгоритма. Для установки ограничений будет использован критерий Антиидеальной точки (Y3, Y4, Y5, Y6), а для выбора алгоритма – критерий Идеальной точки (Y1, Y2, Y4).

1. Определение весов критериев.

Так как изначально нет оценок экспертов, попросим независимых экспертов оценить критерии. Оценки отображены в табл. 16.

Нормализуем полученные данные по (1):

$$b_{ij} = 1 - \frac{a_{ij} - 1}{n}, \quad (1)$$

где

$a_{ij}$  – значение из табл. 16,  
 $n$  – количество критериев.

Нормализация оценок продемонстрирована в табл. 17.

Выполняем расчет по (2):

$$c_{ij} = \frac{b_{ij}}{\sum_{j=0}^n b_{ij}}, \quad (2)$$

где

$b_{ij}$  – значение из табл. 17,

$n$  – количество критериев,

$\sum_{j=0}^n b_{ij}$  – сумма рангов для  $i$ -го эксперта.

Вычисляем вес критерия по (3):

$$y_j = \sum_{i=0}^m \frac{c_{ij}}{m}, \quad (3)$$

где

$c_{ij}$  – значение из табл. 18,

$m$  – количество экспертов,

$\sum_{i=0}^m c_{ij}$  – сумма рангов для  $j$ -го критерия.

Все расчеты отображены в табл. 18.

Таблица 16

Оценка независимых экспертов

Эксперт	Критерии							Сумма рангов
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	
Э1	1	3	4	6	6	6	2	28
Э2	2	1	5	7	4	6	3	28
Э3	3	2	4	6	6	6	1	28
Э4	3	4	1	7	6	5	2	28
Э5	2	1	4	6	5	7	3	28
Э6	4	5	1	3	2	6	7	28

Примечание: составлено авторами.

Таблица 17

Нормализация оценок экспертов

Эксперт	Критерии							Сумма рангов
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	
Э1	1,00	0,71	0,86	0,57	0,29	0,29	0,29	4
Э2	0,86	1,00	0,71	0,43	0,57	0,29	0,14	4
Э3	0,71	0,86	1,00	0,57	0,29	0,29	0,29	4
Э4	0,71	0,57	0,86	1,00	0,29	0,43	0,14	4
Э5	0,86	1,00	0,71	0,57	0,43	0,14	0,29	4
Э6	0,57	0,43	0,14	1,00	0,86	0,29	0,71	4

Примечание: составлено авторами.

Таблица 18

Получение весов критериев от среднего значения суммы для каждого критерия

Эксперт	Критерии							Сумма рангов
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	
Э1	0,25	0,18	0,21	0,14	0,07	0,07	0,07	1
Э2	0,21	0,25	0,18	0,11	0,14	0,07	0,04	1
Э3	0,18	0,21	0,25	0,14	0,07	0,07	0,07	1
Э4	0,18	0,14	0,21	0,25	0,07	0,11	0,04	1
Э5	0,21	0,25	0,18	0,14	0,11	0,04	0,07	1
Э6	0,14	0,11	0,04	0,25	0,21	0,07	0,18	1
СУММ	1,18	1,14	1,07	1,04	0,68	0,43	0,46	6
Вес коэф.	0,20	0,19	0,18	0,17	0,11	0,07	0,08	1

Примечание: составлено авторами.

## 2. Нормализация критериев

Для нормализации критериев используем критерий Севиджа. Нормализация критериев критерием Севиджа – это метод, использующийся в многокритериальном анализе, который позволяет привести к одной числовой шкале различные критерии, которые имеют разный масштаб и единицы измерения.

Сначала необходимо определить максимальное значение для каждого критерия. Затем для каждого критерия рассчитывается разность между максимальным значением и реальными значениями.

После этого для каждого критерия рассчитывается минимальное значение разности. Далее выбирается максимальное значение минимальной разности – это и будет максимальным критерием. Используя формулу, каждый показатель нормализуется по максимальному критерию.

Нормализованный показатель для каждого критерия рассчитывается по (4):

$$Z_i = \frac{C_{max} - Z_i}{C_{max} - C_{min}}, \quad (4)$$

где

$Z_i$  – реальное значение  $i$ -го критерия,

$C_{\max}$  – максимальное значение разности между максимальным и реальным значениями,

$C_{\min}$  – минимальное значение разности между максимальным и реальным значениями.

Расчеты приведены в табл. 19.

Для критериев Y1, Y6 и Y7 выполним смену направления. В табл. 20 представлены нормализованные критерии.

Данные в табл. 20 обозначают какую-либо величину. Необходимо перейти к безразмерным шкалам. Для этого используем Относительную нормализацию. Выполняем расчеты по (5):

$$b_{ij} = \frac{a_{ij}}{\text{MAX}(a_j)}, \quad (5)$$

где

$a_{ij}$  – значение из табл. 19,

$\text{MAX}(a_j)$  – значение максимума из табл. 20,

Расчеты внесены в табл. 21.

Применим ограничение в виде критерия Антиидеальной точки для критериев 3, 4, 5 и 6. Ограничения Антиидеальной точки используется для минимизации отклонения от наилучшего возможного решения. Этот метод заключается в том, чтобы найти решение, которое

Таблица 19

Нормализация критерием Севиджа

Алгоритм	Критерии						
	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X1	16,475	3466713088	432,5492	0,873	0,6696	0,6184	0,763
X2	18,25	135483392	707,8021	0,9483	0,7324	0,6017	0,9003
X3	17,625	4398022658	478,238	0,8837	0,6716	0,6181	0,7818
X4	23,425	5585196034	450,3228	0,9675	0,73	0,5086	0,9074
X5	19,425	5312342018	474,598	0,9085	0,6875	0,613	0,8275
X6	21,1	433510400	596,0817	0,938	0,7184	0,6103	0,8817
X7	17,65	2940262401	1683,932	0,9049	0,6897	0,6178	0,8206
X8	71,85	1471651840	63973,08	0,958	0,7273	0,5807	0,9195
X9	18,95	5879365634	1570,169	0,9089	0,6876	0,6201	0,8281
X10	19,95	6597687298	35698,55	0,9042	0,6901	0,6053	0,8187
X11	28,775	5881736194	24682,08	0,8853	0,6807	0,6022	0,7846
X12	19,3	6714694658	24582,36	0,8695	0,6657	0,6214	0,7572
Макс	71,85	6714694658	63973,08	0,9675	0,7324	0,6214	0,9195
Мин	16,475	135483392	432,5492	0,8695	0,6657	0,5086	0,7572

Примечание: составлено авторами.

Таблица 20

Нормализация критерием Севиджа

Алгоритм	Критерии						
	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X1	55,375	3466713088	432,5492	0,873	0,6696	0,003	0,1565
X2	53,6	135483392	707,8021	0,9483	0,7324	0,0197	0,0192
X3	54,225	4398022658	478,238	0,8837	0,6716	0,0033	0,1377
X4	48,425	5585196034	450,3228	0,9675	0,73	0,1128	0,0121
X5	52,425	5312342018	474,598	0,9085	0,6875	0,0084	0,092
X6	50,75	433510400	596,0817	0,938	0,7184	0,0111	0,0378
X7	54,2	2940262401	1683,932	0,9049	0,6897	0,0036	0,0989
X8	0	1471651840	63973,08	0,958	0,7273	0,0407	0
X9	52,9	5879365634	1570,169	0,9089	0,6876	0,0013	0,0914
X10	51,9	6597687298	35698,55	0,9042	0,6901	0,0161	0,1008
X11	43,075	5881736194	24682,08	0,8853	0,6807	0,0192	0,1349
X12	52,55	6714694658	24582,36	0,8695	0,6657	0	0,1623
Макс	55,375	6714694658	63973,08	0,9675	0,7324	0,1128	0,1623

Примечание: составлено авторами.

Таблица 21

Использование относительной нормализации

Алгоритм	Критерии						
	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X1	0,297517	0,51628753	0,006761	0,902326	0,914255	5,48227	4,701171
X2	0,329571	0,02017715	0,011064	0,980155	1	5,33422	5,547135
X3	0,318284	0,65498476	0,007476	0,913385	0,916985	5,47961	4,817006
X4	0,423025	0,83178705	0,007039	1	0,996723	4,508865	5,590881
X5	0,35079	0,79115169	0,007419	0,939018	0,938695	5,434397	5,098583
X6	0,381038	0,06456145	0,009318	0,969509	0,980885	5,410461	5,432532
X7	0,318736	0,43788475	0,026323	0,935297	0,941699	5,47695	5,056069
X8	1,297517	0,21916884	1	0,990181	0,993037	5,14805	5,665434
X9	0,342212	0,87559687	0,024544	0,939432	0,938831	5,49734	5,10228
X10	0,360271	0,98257443	0,558025	0,934574	0,942245	5,366135	5,044362
X11	0,519639	0,87594991	0,38582	0,915039	0,92941	5,338652	4,834258
X12	0,348533	1	0,384261	0,898708	0,90893	5,508865	4,665434
Гамма	0,20	0,19	0,18	0,17	0,11	0,07	0,08
Z	0,30	0,02	0,01	0,90	0,91	4,51	4,67

Примечание: составлено авторами.

максимально удалено от всех нежелательных характеристик (антиидеальных точек) и максимально близко к идеальным характеристикам.

Алгоритм использования ограничения Антиидеальной точки включает в себя следующие шаги.

1. Вычислить расстояние от каждой точки (варианта решения) в матрице оценок до антиидеальной точки по следующей (6):

$$q_i = \sum_{j=1}^m \gamma_j^2 * (z_j - x_{ij})^2, \quad (6)$$

где

$j$  – это индекс критерия,

$\gamma$  – это вес коэффициентов, который был найден на этапе обработки оценок экспертов,

$z$  – это минимальное значение для критерия  $j$  (из всех алгоритмов),

$x$  – это значение критерия конкретного алгоритма.

2. Просуммировать и получить столбец с суммами для каждого метода рекомендательной системы.

3. На основе среднего значения столбца с суммами отсеять те методы рекомендаций, которые ниже среднего.

Расчеты в табл. 22.

Находим среднее значение  $z\_gr$  для последнего столбца в табл. 22, равное 0,008138, и исключаем алгоритмы X1, X2, X3, X4, X5, X6, X7, X9 и X11, т.к. они меньше рассчитанного значения  $z\_gr$ .

Таблица 22

Использование ограничения критерием Антиидеальной точки

Алгоритм	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Сумма
X1			0,000000	0,000000	0,000000	0,004834		0,004835
X2			0,000001	0,000198	0,000106	0,003476		0,003780
X3			0,000000	0,000006	0,000001	0,004808		0,004815
X4			0,000000	0,000306	0,000099	0,000000		0,000404
X5			0,000000	0,000048	0,000011	0,004370		0,004430
X6			0,000000	0,000149	0,000066	0,004147		0,004363
X7			0,000012	0,000040	0,000014	0,004782		0,004847
X8			0,031458	0,000249	0,000090	0,002084		0,033882
X9			0,000010	0,000049	0,000011	0,004985		0,005056
X10			0,009690	0,000038	0,000014	0,003750		0,013492
X11			0,004582	0,000008	0,000005	0,003513		0,008108
X12			0,004544	0,000000	0,000000	0,005102		0,009646

Примечание: составлено авторами.

Применим критерий Идеальной точки для оставшихся критериев. Критерий Идеальной точки широко используется в теории принятия решений при ранжировании альтернатив. Он является одним из наиболее эффективных критериев для принятия решений в группе.

Критерий Идеальной точки предполагает, что при принятии решения индивид выбирает тот вариант, который находится ближе всего к его идеальному представлению о желаемом результате. Основная идея заключается в том, чтобы найти оптимальное решение, которое минимизирует расстояние до всех критериев, максимально приближаясь к идеальному результату по каждому из них.

Алгоритм использования критерия Идеальной точки включает в себя следующие шаги:

1. Вычислить расстояние от каждой точки (варианта решения) в матрице оценок до идеальной точки по следующей (7):

$$q_i = \sum_{j=1}^m \gamma_j^2 * (x_{ij} - z_j)^2, \quad (7)$$

где

$j$  – это индекс критерия,

$\gamma$  – это вес коэффициентов, который был найден на этапе обработки оценок экспертов,

$z$  – это минимальное значение для критерия  $j$  (из всех алгоритмов),

$x$  – это значение критерия конкретного алгоритма.

2. Просуммировать и получить столбец с суммами для каждого метода (алгоритма) рекомендательной системы.

3. Выбрать альтернативу с максимальным значением суммы (учитывая исключения, которые были сделаны ограничением критерия Антиидеальной точки).

Расчеты находятся в табл. 23.

В данном случае лучшей альтернативой является алгоритм X8 – это нейронная сеть на базе архитектуры DLRM. Данный алгоритм показывает достаточно высокий результат точности, а также является одним из немногих алгоритмов, который не нагружает оперативную память, из-за чего имеет большую ресурсоемкость по входным данным для обучения модели и соответственно выдачи рекомендаций.

## РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Реализация рекомендательной системы, построенной на архитектуре DLRM, включает в себя разработку алгоритма, проектирование базы данных, а также графического и программного интерфейса.

Система реализована на языке программирования Python с использованием библиотек numpy, pandas, csv, os, которые ранее применялись и при обучении методов. Помимо данных библиотек потребуются еще следующие:

– flask – легковесный веб-фреймворк для Python, предназначенный для создания веб-приложений и API [16];

– psycopg2 – библиотека для взаимодействия с PostgreSQL базами данных из Python [17];

Таблица 23

Использование критерия Идеальной точки

Алгоритм	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Сумма
X1	0,000000	0,008930					0,000008	0,008937
X2	0,000040	0,000000					0,004655	0,004695
X3	0,000017	0,014621					0,000138	0,014775
X4	0,000608	0,023899					0,005128	0,029635
X5	0,000110	0,021566					0,001123	0,022799
X6	0,000269	0,000071					0,003523	0,003864
X7	0,000017	0,006330					0,000914	0,007261
X8	0,038584	0,001437					0,005988	<b>0,046009</b>
X9	0,000077	0,026548					0,001143	0,027768
X10	0,000152	0,033604					0,000860	0,034616
X11	0,001904	0,026570					0,000171	0,028645
X12	0,000100	0,034832					0,000000	0,034932

Примечание: составлено авторами.

– tkinter – встроенный в Python стандартный модуль для создания графических пользовательских интерфейсов (GUI) [18];  
– pickle – встроенный модуль Python для сериализации и десериализации объектов (конвертация объектов в байтовый поток и обратно) [19].

Алгоритм работы рекомендаций будет зависеть от того, является ли человек зарегистрированным пользователем онлайн-кинотеатра или нет.

Если пользователь зарегистрирован на сервисе онлайн-кинотеатра, то берутся данные, хранящиеся в базе данных сервиса, и формируются в один список из `userId`. Затем этот список передается в функцию для выдачи предсказаний, загружается уже обученная модель алгоритма по рекомендательным системам, после чего выдается `movieId`, который система заранее знает.

Если же пользователь не зарегистрирован, то есть два решения данной ситуации.

1. Предложить незарегистрированному пользователю список самых популярных и высоко оцененных фильмов/сериалов в сервисе, таким образом можно будет как получить в будущем информацию о данном пользователе, так и не испортить его впечатление в качестве посетителя сервиса.

2. Попросить пользователя ввести свои возраст, пол, а также выбрать предпочитаемые жанры фильмов и методом ближайших точек найти пользователей по похожим параметрам и предложить фильмы и сериалы, которые можно будет предложить пользователям с аналогичными данными профиля.

Блок-схема данного алгоритма реализована на рис. 3.

В первую очередь нужно спроектировать базу данных. В качестве СУБД была выбрана PostgreSQL, так как данная технология открыта и хорошо интегрируется с Python. На рис. 4 представлена будущая БД, которую необходимо разработать.

БД включает в себя 5 табл.:

– `users` – таблица, в которой хранится информация о пользователях, а точнее `userId`

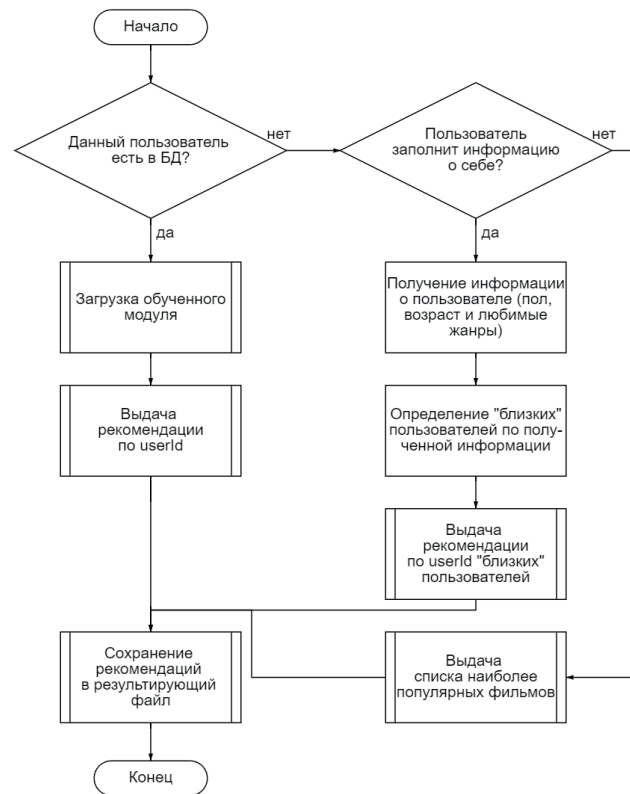


Рис. 3. Блок-схема алгоритма рекомендаций

Примечание: составлено авторами.

(уникальный идентификатор пользователя), `age` (возраст пользователя) и `sex` (пол пользователя);

– `movies` – таблица, в которой хранится информация о видеоконтенте, а точнее `movieId` (уникальный идентификатор), `title` (название видеоконтента), `genres` (жанр видеоконтента);

– `ratings` – таблица, в которой хранится информация об оценках пользователей, а точнее `userId` (уникальный идентификатор пользователя, который оставил оценку), `movieId` (уникальный идентификатор видеоконтента в сервисе), `rating` (оценка пользователя), `timestamp` (время, когда была поставлена оценка пользователем);

– `links` – таблица, в которой хранится информация об уникальном идентификаторе видеоконтента на крупных стриминговых площадках, а точнее `movieId` (уникальный идентификатор видеоконтента в сервисе MovieLens), `imdbId` (уникальный идентификатор видеоконтента в сервисе IMDb), `tmdbId` (уникальный идентификатор

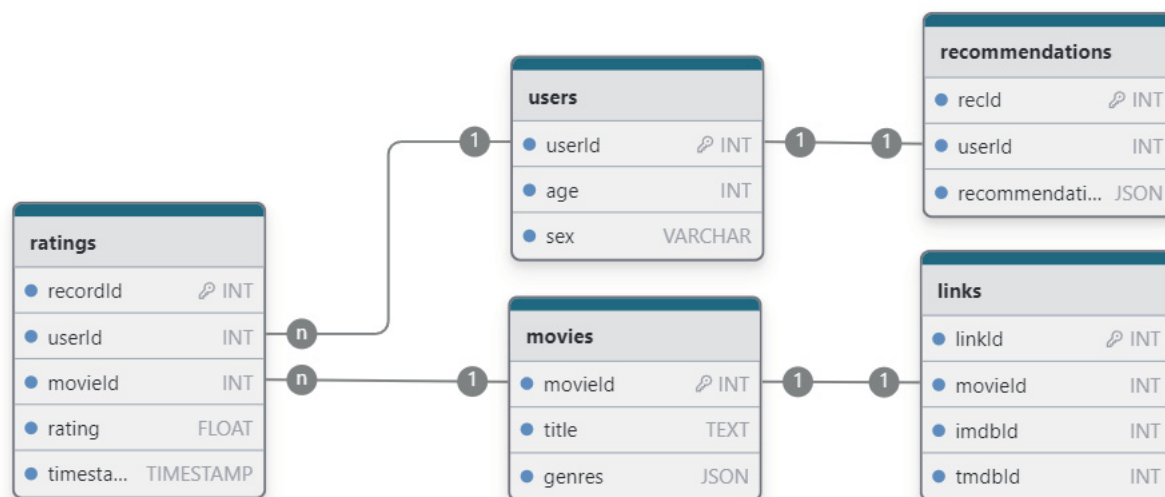


Рис. 4. Схема базы данных  
Примечание: составлено авторами.

видеокартента в сервисе The Movie Database). Это сделано для будущей интеграции сервиса с другими системами, так как movieId предназначен только для внутренней БД, в то время как imdbId и tmdbId являются уникальными идентификаторами крупнейших внешних сервисов;

– recommendations – таблица, в которой хранится информация о рекомендации для пользователей, а точнее userId (уникальный идентификатор пользователя), recommendation (список с movieId, который предлагается для просмотра пользователю).

Связи в БД между таблицами:

– users.userId – ratings.userId (связь один ко многим), так как один человек может оставлять множество оценок на различный видеоконтент;

– movies.movieId – ratings.movieId (связь один ко многим), так как один фильм может иметь множество оценок от пользователей;

– users.userId – recommendations.userId (связь один к одному), так как один человек всегда имеет лишь один список рекомендаций в сервисе;

– movies.movieId – links.movieId (связь один к одному), так как у одного фильма лишь одна привязка на IMDB и TMDB.

В ходе разработки системы по выдаче рекомендаций была реализована административная панель для работы с моделью, а также для

небольших точечных изменений данных в БД. Графический интерфейс, он же GUI, представляет собой способ взаимодействия пользователя с компьютером с использованием графических элементов, например окон, кнопок и меню [20].

Реализованный графический интерфейс представлен на рис. 5.

На интерфейсе расположены кнопки со следующим функционалом.

1. Кнопка «Выдача рекомендаций», представленная на рис. 6, отвечает за формирование списка рекомендаций. Можно выбрать csv-файл с id пользователей либо запросить загрузку актуальных данных из БД.

2. С помощью кнопки «Пользователи», которая изображена на рис. 7, можно добавить или удалить пользователя из БД, сделана для небольших точечных изменений.

3. С помощью кнопки «Фильмы», представленной на рис. 8, можно добавить или удалить фильм из БД, сделана для небольших точечных изменений.

4. Кнопка «Рейтинг», которая изображена на рис. 9, позволяет добавить или удалить рейтинг фильма из БД, сделана для небольших точечных изменений.

5. Кнопка «Ссылки» продемонстрирована на рис. 10, дает возможность добавить или удалить ссылку на фильм из БД, сделана для небольших точечных изменений.

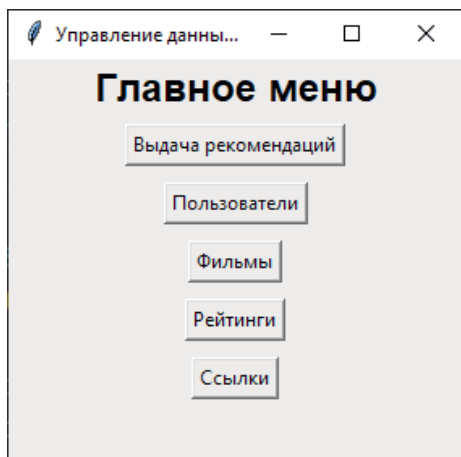


Рис. 5. Главное меню ПО  
Примечание: составлено авторами.

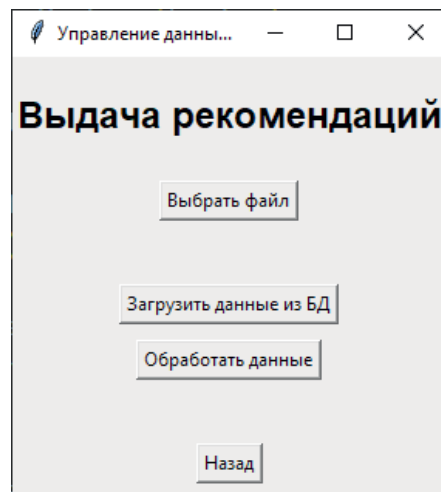


Рис. 6. Выбор действий по кнопке «Выдача рекомендаций»  
Примечание: составлено авторами.

Данное меню было реализовано для небольших изменений данных. Для массовых изменений на стороне базы данных можно воспользоваться платформой pgAdmin или другими профильными инструментами.

Реализуем API рекомендательной системы для возможной будущей интеграции с существующими онлайн-кинотеатрами. Интерфейс реализован на базе библиотеки Flask.

Разработанный программный интерфейс приложения имеет следующие ссылки.

1./recommendations\_client\_service – get запрос для выдачи рекомендаций зарегистрированным пользователям в сервисе онлайн-ки-

нотеатра. Возвращает двухсотый код, а также json объект со списком фильмов (id, название, предполагаемая оценка пользователя), которые можно предложить человеку в качестве рекомендаций. В случае ошибки возвращает пятисотый код.

Имеет следующие параметры:

- user\_id – id пользователя в сервисе онлайн-кинотеатра;
- num\_recommendations (необязательный) – количество рекомендаций на одного пользователя, по умолчанию выдает 5 рекомендаций.

2./recommendations\_no\_client\_service\_by\_model – get запрос для выдачи рекоменда-

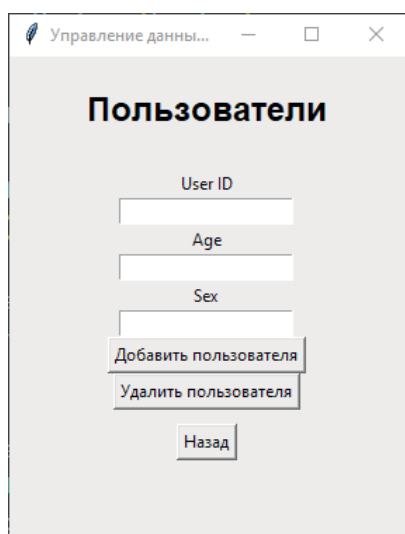


Рис. 7. Выбор действий по кнопке «Пользователи»  
Примечание: составлено авторами.



Рис. 8. Выбор действий по кнопке «Фильмы»  
Примечание: составлено авторами.

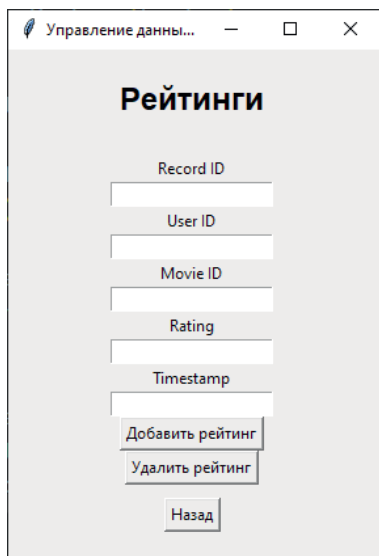


Рис. 9. Выбор действий по кнопке «Рейтинг»  
Примечание: составлено авторами.

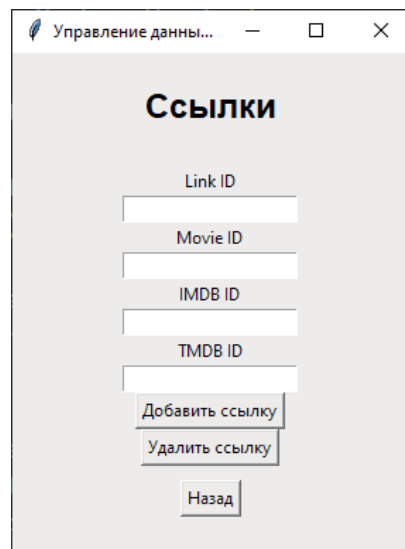


Рис. 10. Выбор действий по кнопке «Ссылки»  
Примечание: составлено авторами.

ций незарегистрированным пользователям, но про которых есть информация об их поле, возрасте и любимых жанрах, полученная с другого сервиса, к которому в дальнейшем с помощью API будет подключена рекомендательная система. Возвращает двухсотый код, а также json объект со списком фильмов (id и название), которые можно предложить человеку в качестве рекомендаций. В случае ошибки возвращает пятисотый код.

Имеет следующие параметры:

- sex – пол пользователя;
- age – возраст пользователя;
- genres – любимые жанры в формате: genre1 |genre2 |...|genreN;
- num\_recommendations (необязательный) – количество рекомендаций на одного пользователя, по умолчанию выдает 25 рекомендаций.

## ЗАКЛЮЧЕНИЕ

В рамках данной магистерской диссертации была проведена разработка рекомендательной системы для онлайн-кинотеатров. Основное внимание уделялось сравнению и анализу трех типов рекомендательных систем: совместной фильтрации, гибридных систем и систем на архитектуре DLRM.

Для оценки качества работы каждого алгоритма использовались различные критерии, включая нагрузку на систему, объем потребляемой оперативной памяти, время,

затраченное на обучение модели, а также метрики качества рекомендаций: RMSE, MAE, FCP и MSE.

Проведенный анализ показал, что каждая из рассмотренных систем имеет свои сильные и слабые стороны. Совместная фильтрация продемонстрировала высокую точность рекомендаций при сравнительно низкой нагрузке на систему, но потребовала значительного времени на обучение модели. Гибридные системы показали хорошее соотношение между точностью и производительностью, комбинируя преимущества различных подходов. Система на архитектуре DLRM оказалась наиболее ресурсозатратной, однако она продемонстрировала наилучшую точность и гибкость в обработке большого объема данных.

На основе проведенного анализа и с использованием математических методов принятия решений был выбран наилучший алгоритм для реализации рекомендательной системы. Лидером оказалась система на архитектуре DLRM, обеспечившая оптимальное соотношение между точностью рекомендаций и производительностью системы.

Реализация выбранного алгоритма включала в себя несколько этапов: разработку самого алгоритма, проектирование базы данных для хранения и обработки данных, создание графического интерфейса пользователя и разработку

API для интеграции рекомендательной системы с другими компонентами онлайн-кинотеатра.

Разработанная система на архитектуре DLRM показала высокие результаты по точности и качеству рекомендаций, что подтверждает возможность ее использования в реальных условиях.

### Список источников

1. Глубокое погружение в рекомендательную систему Netflix. Хабр : офиц. сайт. URL: <https://habr.com/ru/articles/677396/> (дата обращения: 24.06.2024).
2. 10 Remarkable Real-World Examples of Recommender Systems. URL: <https://appstekcorp.com/blog/10-remarkable-real-world-examples-of-recommender-systems/> (дата обращения: 24.06.2024).
3. Deep Dive into Netflix's Recommender System. Medium. URL: <https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48> (дата обращения: 24.06.2024).
4. How retailers can keep up with consumers. URL: <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers> (дата обращения: 24.06.2024).
5. Recommender systems: benefits and practical guidelines for software professionals. URL: <https://swforum.eu/online-sw-forum/software-technology/6/recommender-systems-benefits-and-practical-guidelines> (дата обращения: 24.06.2024).
6. Python: Skillfactory media. Честные истории о карьере в IT. URL: <https://blog.skillfactory.ru/glossary/python/> (дата обращения: 24.06.2024).
7. Библиотека NumPy: всё, что нужно знать новичку. URL: <https://skillbox.ru/media/code/biblioteka-numpy-vsye-chto-nuzhno-znat-novichku/> (дата обращения: 24.06.2024).
8. Работаем с Pandas: основные понятия и реальные данные. URL: <https://skillbox.ru/media/code/rabotaem-s-pandas-osnovnye-ponyatiya-i-realnye-dannye/> (дата обращения: 24.06.2024).
9. Библиотека Scikit-learn: как создать свой первый ML-проект. Skillbox. URL: <https://skillbox.ru/media/code/biblioteka-scikitlearn-kak-sozdat-svoy-pervyy-mlproekt/> (дата обращения: 24.06.2024).
10. Keras: библиотека глубокого обучения на Python. URL: <https://ru-keras.com/home/> (дата обращения: 24.06.2024).
11. Overview. A Python scikit for recommender systems. URL: <https://surpriselib.com/> (дата обращения: 24.06.2024).
12. csv – CSV File Reading and Writing. URL: <https://docs.python.org/3/library/csv.html> (дата обращения: 24.06.2024).
13. Библиотека os. Все о Python. Программирование на Python 3. URL: <https://all-python.ru/osnovy/os.html> (дата обращения: 24.06.2024).

Таким образом, задачи, поставленные в начале исследования, были успешно решены, а цель достигнута. Результаты данной работы могут служить основой для дальнейших исследований в области рекомендательных систем и их применения в различных сферах, связанных с подбором персонализированного контента.

### References

1. Glubokoe pogruzhenie v rekomendatelnuyu sistemu Netflix. URL: <https://habr.com/ru/articles/677396/> (accessed: 24.06.2024). (In Russ.).
2. 10 Remarkable Real-World Examples of Recommender Systems. 2023. URL: <https://appstekcorp.com/blog/10-remarkable-real-world-examples-of-recommender-systems/> (accessed: 24.06.2024).
3. Deep Dive into Netflix's Recommender System: Medium. URL: <https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48> (accessed: 24.06.2024).
4. How retailers can keep up with consumers. URL: <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers> (accessed: 24.06.2024).
5. Recommender systems: benefits and practical guidelines for software professionals. URL: <https://swforum.eu/online-sw-forum/software-technology/6/recommender-systems-benefits-and-practical-guidelines> (accessed: 24.06.2024).
6. Python. Skillfactory media. Chestnyye istorii o karyere v IT. URL: <https://blog.skillfactory.ru/glossary/python/> (accessed: 24.06.2024). (In Russ.).
7. Biblioteka NumRu: vsyo, chto nuzhno znat novichku. URL: <https://skillbox.ru/media/code/biblioteka-numpy-vsye-chto-nuzhno-znat-novichku/> (accessed: 24.06.2024). (In Russ.).
8. Rabotaem s Pandas: osnovnye ponyatiya i realnye dannye. URL: <https://skillbox.ru/media/code/rabotaem-s-pandas-osnovnye-ponyatiya-i-realnye-dannye/> (accessed: 24.06.2024). (In Russ.).
9. Biblioteka Scikit-learn: kak sozdat svoj pervyy ML-proekt. Skillbox. URL: <https://skillbox.ru/media/code/biblioteka-scikitlearn-kak-sozdat-svoy-pervyy-mlproekt/> (accessed: 24.06.2024). (In Russ.).
10. Keras: biblioteka glubokogo obuchenie na Python. URL: <https://ru-keras.com/home/> (accessed: 24.06.2024). (In Russ.).
11. Overview. A Python scikit for recommender systems. URL: <https://surpriselib.com/> (accessed: 24.06.2024).
12. csv – CSV File Reading and Writing. URL: <https://docs.python.org/3/library/csv.html> (accessed: 24.06.2024).
13. Biblioteka os: Vse o Python. Programmirovaniye na Python 3. URL: <https://all-python.ru/osnovy/os.html> (accessed: 24.06.2024). (In Russ.).

14. time – Time access and conversions. URL: <https://docs.python.org/3/library/time.html> (дата обращения: 24.06.2024).
15. psutil documentation. URL: <https://psutil.readthedocs.io/en/latest/index.html> (дата обращения: 24.06.2024).
16. Фреймворк Flask: как он работает и зачем нужен. URL: <https://skillbox.ru/media/code/freymvork-flask-kak-on-rabotaet-i-zachem-nuzhen/> (дата обращения: 24.06.2024).
17. How to Connect and Operate PostgreSQL with Python Using psycopg2 Lib. URL: <https://geekpython.in/integrate-postgresql-database-in-python> (дата обращения: 24.06.2024).
18. tkinter – Python interface to Tcl/Tk. URL: <https://docs.python.org/3/library/tkinter.html> (дата обращения: 24.06.2024).
19. pickle – Python object serialization. URL: <https://docs.python.org/3/library/pickle.html> (дата обращения: 24.06.2024).
20. GUI: Skillfactory media. Честные истории о карьере в IT. URL: <https://blog.skillfactory.ru/glossary/gui/> (дата обращения: 24.06.2024).
14. time – Time access and conversions. URL: <https://docs.python.org/3/library/time.html> (accessed: 24.06.2024).
15. psutil documentation. URL: <https://psutil.readthedocs.io/en/latest/index.html> (accessed: 24.06.2024).
16. Freymvork Flask: kak on rabotaet i zachem nuzhen. URL: <https://skillbox.ru/media/code/freymvork-flask-kak-on-rabotaet-i-zachem-nuzhen/> (accessed: 24.06.2024). (In Russ.).
17. How to Connect and Operate PostgreSQL with Python Using psycopg2 Lib. URL: <https://geekpython.in/integrate-postgresql-database-in-python> (accessed: 24.06.2024).
18. tkinter – Python interface to Tcl/Tk. URL: <https://docs.python.org/3/library/tkinter.html> (accessed: 24.06.2024).
19. pickle – Python object serialization. URL: <https://docs.python.org/3/library/pickle.html> (accessed: 24.06.2024).
20. GUI: Skillfactory media. Chestnyye istorii o karyere v IT. URL: <https://blog.skillfactory.ru/glossary/gui/> (accessed: 24.06.2024). (In Russ.).

### **Информация об авторах**

**К. Е. Кожихова** – аспирант.

**Д. В. Тараканов** – кандидат технических наук, доцент.

**И. В. Чалей** – доктор технических наук, профессор.

### **About the authors**

**K. E. Kozhikhova** – Postgraduate.

**D. V. Tarakanov** – Candidate of Sciences (Engineering), Docent.

**I. V. Chaley** – Doctor of Sciences (Engineering), Professor.