

ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ/INFORMATICS AND INFORMATION PROCESSES

DOI: https://doi.org/10.60797/IRJ.2026.171.24 EDN: WADHGE

ИСПОЛЬЗОВАНИЕ МЕХАНИЗМА ВНИМАНИЯ В ГРАФОВЫХ НЕЙРОННЫХ СЕТЯХ ДЛЯ ФИЛЬТРАЦИИ ОБУЧАЮЩИХ ДАННЫХ НА ПРИМЕРЕ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ

Научная статья

Гудков В.М.[1, *]

[1] Воронежский государственный университет, Воронеж, Российская Федерация

* Корреспондирующий автор (dfcz.petro[at]yandex.ru)

Предложена: 03.06.2026; Принята: 20.08.2026; Опубликована: 17.09.2026

Аннотация

Современные рекомендательные системы демонстрируют высокую точность, но характеризуются значительной ресурсоемкостью, что затрудняет их практическое внедрение и масштабирование. Статья посвящена методу оптимизации обучения рекомендательных систем путем сокращения объема обучающих данных с помощью механизма внимания в графовых нейронных сетях. Предложенный подход, основанный на фильтрации менее значимых взаимодействий на основе весов внимания, позволяет снизить вычислительные затраты и ускорить процесс обучения без существенной потери качества рекомендаций. Экспериментальная проверка на датасете Amazon Books подтвердила эффективность метода для различных архитектур (LightGCN, BPR, KNN). Результаты работы указывают на перспективность использования механизмов внимания как средства повышения эффективности обучения.

Ключевые слова: рекомендательные системы, графовые нейронные сети, механизм внимания, машинное обучение, дистилляция знаний.

THE USE OF AN ATTENTION MECHANISM IN GRAPH NEURAL NETWORKS FOR FILTERING TRAINING DATA ON THE EXAMPLE OF RECOMMENDATION SYSTEMS

Research article

Gudkov V.M.[1, *]

[1] Voronezh State University, Voronezh, Russian Federation

* Corresponding author (dfcz.petro[at]yandex.ru)

Suggested: 03.06.2026; Accepted: 20.08.2026; Published: 17.09.2026

Abstract

Modern recommendation systems demonstrate high accuracy but are characterised by significant resource consumption, which hinders their practical implementation and scaling. The article presents a method for optimising the training of recommendation systems by reducing the volume of training data using an attention mechanism in graph neural networks. The suggested approach, which involves filtering out less significant interactions based on attention weights, reduces computational costs and speeds up the training process without any significant loss in the quality of recommendations. Experimental validation on the Amazon Books dataset confirmed the method's effectiveness across various architectures (LightGCN, BPR, KNN). The results of this work indicate the potential for using attention mechanisms as a means of improving training efficiency.

Keywords: recommendation systems, graph neural networks, attention mechanisms, machine learning, knowledge distillation.

Введение

Рекомендательные системы стали неотъемлемой частью современных цифровых платформ, обеспечивая персонализированный доступ к информации, товарам и сервисам. Они используются в интернет-магазинах для формирования индивидуальных товарных подборок, в медиасервисах — для ранжирования фильмов, музыки и новостей, в социальных сетях — для рекомендации друзей и так далее. Эффективность таких систем напрямую влияет на удовлетворенность пользователей, время взаимодействия с платформой и коммерческие метрики, что делает область рекомендательных алгоритмов одной из наиболее интенсивно развивающихся в современном машинном обучении.

1.1. Современные подходы и механизм внимания

В последние годы активно развиваются алгоритмы использующие глубокие нейронные сети, в частности нейронные сети, включающие механизм внимания. К наиболее заметным подходам относятся нейро-факторизационные модели, архитектуры последовательных рекомендаций на основе Transformer, вдохновленные достижениями в области языковых моделей, а также графовые модели [1], [2], [3], [4], [5].

1.2. Проблемы использования современных моделей машинного обучения

Современные рекомендательные модели демонстрируют впечатляющие результаты, однако их повсеместное внедрение часто оказывается существенно затруднено. Основная проблема заключается в высокой ресурсоемкости таких моделей, обусловленной несколькими факторами.

Во-первых, большинство актуальных архитектур — многослойные архитектуры, реализующие свое преимущество через масштабируемость за счет увеличения числа параметров.

Во-вторых, реальные рекомендательные системы работают в динамических условиях. Быстро появляются как новые алгоритмы, так и новые области, в которых необходимы точные рекомендации. Это приводит к необходимости частого обучения моделей. Для тяжелых моделей такая частота обновлений становится практически невыполнимой.

В-третьих, часто модели чувствительны к гиперпараметрам. Настройка и оценка требует полного цикла обучения, что в совокупности с ресурсоемкостью данного процесса приводит к невозможности полноценного анализа влияния параметров и в следствии использования эмпирических значений.

Таким образом, несмотря на высокую точность, современные рекомендательные модели оказываются трудными для использования в реальных системах из-за высокой вычислительной стоимости и требования к регулярному переобучению. Это формирует запрос на методы, которые позволяют ускорить обучение, уменьшить объем данных или снизить сложность модели без существенной потери качества.

Методы и принципы исследования

Ключевая идея настоящей работы концептуально опирается на подходы дистилляции знаний, активно развиваемые в области больших языковых моделей (LLM) [6], [7]. В классической дистилляции знаний информация передаётся от сложной модели-учителя к более простой модели-ученику за счет использования внутренних состояний модели-учителя (обычно внутренних активаций). Однако в предлагаемом исследовании используется альтернативное направление дистилляции. Вместо внутренних состояний слоев нейросети источником знаний выступают веса внимания, формируемые крупной рекомендательной моделью. Значения внимания интерпретируются как индикатор важности отдельных взаимодействий пользователя с объектами. Это позволяет выделить наиболее информативные элементы обучающих данных и использовать их. Таким образом, предлагаемая работа переносит идеологию дистилляции из LLM в область рекомендательных систем, но делает это через механизм фильтрации данных, а не через передачу внутренних скрытых представлений.

2.1. Методы

Механизм внимания (attention) реализует предложение о том, что необходимо фокусироваться на наиболее значимых элементах входных данных. В рекомендательных системах это означает способность выявлять наиболее информативные пользовательские взаимодействия — такие, которые вносят наибольший вклад в итоговые рекомендации.

Интуитивно внимание работает как механизм адаптивного взвешивания: алгоритм самостоятельно определяет важные элементы, присваивая им большие коэффициенты значимости.

Формально механизм внимания для рекомендательных систем можно ввести следующим образом. Пусть пользователь $u \in U$ имеет историю взаимодействий $s_1, s_2, ..., s_m$ и потенциального кандидата на рекомендацию i_t $(s, i \in I)$, тогда имеет место мера, которая будет показывать насколько стоит обращать внимание на каждый объект из истории в момент данной рекомендации

$$score(s_k, i_t), k = 1..m. \tag{1}$$

Тогда вес внимания для одного взаимодействия

$$\alpha_{kt} = \frac{e^{\text{score}(s_k, i_t)}}{\sum_{j=1}^{m} e^{\text{score}(s_j, i_t)}}, \qquad k = 1, \dots, m. \tag{2}$$

В подавляющем большинстве методов машинного обучения адаптация модели происходит за счет минимизации функции потерь. В общем виде ее можно представить следующим образом

$$L(\theta) = \sum_{i=0}^{N} l(y_i, \hat{y}_i), \tag{3}$$

где i — номер обучающего примера, N — их общее количество, y_i — предсказанное моделью значение, $\hat{y}_i$ — точное значение, l — функция потерь на одном примере, θ — обучаемые (оптимизируемые) параметры модели.

Заметим, что функция потерь, а следовательно, и оптимизируемое значение напрямую зависят от обучающей выборки, используемой для оптимизации параметров θ . Как известно существуют функции, одинаково преобразующие множество входных аргументов во множество выходных, отличающихся при этом только представлением.

Из этих предпосылок и формируется идея данной статьи – преобразовать функцию потерь в более компактный вид с минимальной потерей показателей использующих ее моделей. Как следствие такой подход должен ускорить как процесс настройки параметров моделей за счет более быстрого процесса обучения, так и процесс внедрения новых моделей за счет возможности использовать оптимизированные наборы данных с произвольными алгоритмами машинного обучения.

В графовых нейронных сетях (архитектурах на основе GAT [8], [9]) рекомендации строятся за счет формирования в процессе обучения латентных (скрытых) представлений вершин и пользователей. Релевантность рекомендации определяет близость полученных представлений, т.е. величиной скалярного произведения

$$y_{ui} = h_u h_i^T, \tag{4}$$

где $h_u \in R^d$ — представление пользователя u , $h_i \in R^d$ — представление объекта i , d — размерность пространства представлений.

В качестве графа рассматривается граф взаимодействий — двудольный граф $G = (U \cup I, E)$, где U — множество пользователей, I — множество объектов, а $E \subseteq U \times I$ — множество рёбер, отражающих взаимодействия между ними (например, факт покупки или оценки). Пример такого графа представлен на рисунке 1.

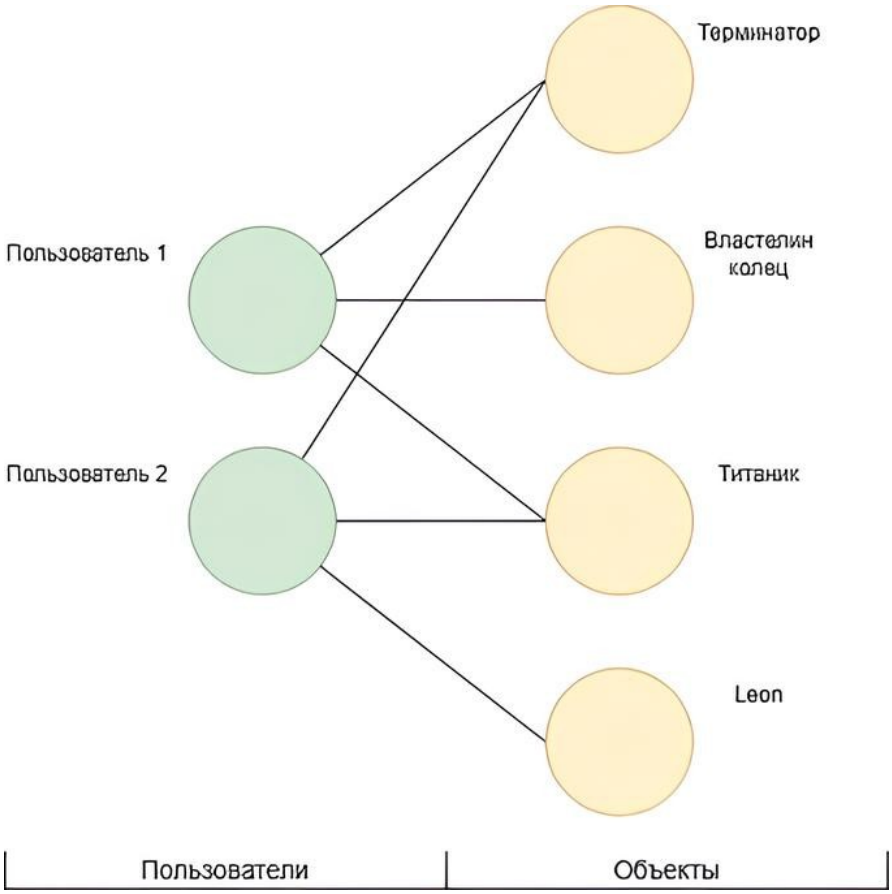

Рисунок 1 - Граф взаимодействий
DOI: https://doi.org/10.60797/IRJ.2026.171.24.1

Представления обновляются итеративно по следующим формулам:

$$h_i' = \alpha_{ii} W_s h_i + \sum_{j \in N(i)} \alpha_{ij} W_t h_j, \tag{5}$$

где $W_s, W_t \in R^{d \times d}$ — обучаемые параметры модели, матрицы преобразования представления при передаче по графу, α_{ij} — коэффициент внимания, $N(i)$ — множество вершин соседей вершины i.

Оценка внимания вводится следующим образом:

$$\text{score}(h_i, h_j) = a^T \text{ LeakyReLU}(W_s h_i + W_t h_j), \tag{6}$$

где $a \in R^d$ — обучаемый параметр, вектор получения внимания.

Подставляя (6) в (2) получаем формулу для вычисления весов внимания:

$$\alpha_{kt} = \frac{\exp(a^T \text{ LeakyReLU}(W_s h_k + W_t h_t))}{\sum_{j \in N(k) \cup i} \exp(a^T \text{ LeakyReLU}(W_s h_k + W_t h_j))}, \quad k = 1 \dots |U \cup I|, \; t \in N(k) \tag{7}$$

Подбор обучаемых параметров W_s, W_t, a осуществляется за счет минимизации функции потерь. В рамках данной архитектуры используется Bayesian personalized ranking loss или BPRLoss [11]

$$L = \sum_{(u,i,j)} -\ln(\sigma(y_{ui} - Y_{uj})), \tag{8}$$

где (u, i, j) — тройки пользователь, верная рекомендация, неверная рекомендация. Идея данной функции ошибки состоит в максимизации вероятности (минимизации отрицательного логарифма вероятности) верного ранжирования, т.е. такого ранжирования, в котором оценка верной рекомендации выше неверной рекомендации.

По окончании процесса оптимизации функции потерь значения α_{ij} сохраняются вместе с соответствующими взаимодействиями.

2.2. Материалы

Amazon Books [15] представляет собой поднабор крупномасштабного корпуса пользовательских отзывов Amazon и содержит данные о взаимодействиях пользователей с товарами категории «книги».

В рамках задачи рекомендаций данные интерпретируются в постановке неявной обратной связи (implicit feedback), где наличие взаимодействия между пользователем и книгой рассматривается как положительный сигнал, а отсутствие взаимодействия — как неявный негативный сигнал.

Разделение на обучающую и тестовую выборку в обоих случаях производилось с использованием стратегии randomsplit, т.е. в качестве тестового набора используется случайная часть фиксированного размера. Размер тестовой выборки 20%.

2.3. Фильтрация

В связи с разреженностью данных могут существовать различия в результатах в зависимости от того, как будут отбираться взаимодействия по весам внимания:

- число k — выбирается фиксированное число взаимодействий на каждого пользователя, независимо от длинны истории, т.е. каждый пользователь ограничивается абсолютным значением;
- доля p — выбирается доля от общего числа, т.е. каждый пользователь ограничивается относительно длинны своей истории взаимодействий.

Результаты моделей сравниваются при каждом различном количестве взаимодействий на пользователя, в одном из трёх сценариев:

- random — для каждого пользователя выбирается k случайных или p процентов взаимодействий;
- top — для каждого пользователя выбирается k или p процентов взаимодействий с наибольшими весами внимания;
- bottom — для каждого пользователя выбирается k или p процентов взаимодействий с наименьшими весами внимания.

При этом полная модель GAT обучается на всем наборе и не участвует в оценке результатов после фильтрации. Параметры модели подбираются также по модели, использующей полный набор данных.

Максимальное значение k — 95 перцентиль длинны истории пользовательских взаимодействий. Для Amazon Books это примерно 100.

Основные результаты

Результаты предлагаемого подхода оценивается в паре с тремя различными вариантами алгоритмов коллаборативной фильтрации LightGCN, BPR и базового алгоритма ближайших соседей item-based KNN.

LightGCN [10] — графовая нейронная сеть основанная на тех же принципах, что и GAT, но использующая механизм обновления весов без обучаемых параметров

$$h_i' = \sum_{j \in N(i)} \frac{1}{\sqrt{\deg(i)\deg(j)}} h_i, \tag{10}$$

где $deg(i)$ — степень вершины i .

Модель BPR напрямую оптимизирует представления пользователей и объектов путем минимизации ошибки в виде (8).

Item-based KNN [12] использует для рекомендаций близость объектов в user-item матрице по корреляционному коэффициенту Пирсона.

Для оценки эффективности были использованы две метрики: Hit Ratio и Mean Reciprocal Rank или же HR@K и MRR [13].

Hit ratio — показывает долю предсказаний по которым в верхних K рекомендованных объектах есть реально оцененный объект

$$\mathrm{HR@}K = \frac{|\{u : \mathrm{rank}_u \le K\}|}{|U|}, \tag{11}$$

где $rank_u$ — первая позиция, на которой был рекомендован релевантный объект для пользователя u . Оценивает возможность рекомендовать правильные объекты в некотором окне.

Mean Reciprocal Rank измеряет среднее значение обратного ранга первого релевантного объекта в рекомендованном списке, отражая способность модели помещать релевантный элемент на максимально высокую позицию.

$$\mathrm{MRR} = \frac{1}{|U|} \sum_{u \in U} \frac{1}{\mathrm{rank}_u} \tag{12}$$

Таким образом, HR@30 оценивает вероятность появления правильной рекомендации в выдаче из 30 объектов, а MRR — близость результата к началу выдачи.

Эксперименты над Amazon Books показывают следующие результаты (рисунки 2 и 3):

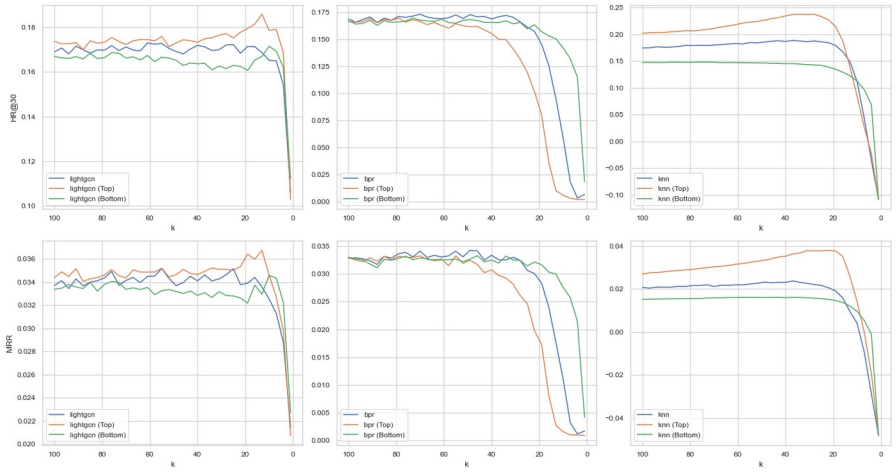

Рисунок 2 - Результаты для k
DOI: https://doi.org/10.60797/IRJ.2026.171.24.2

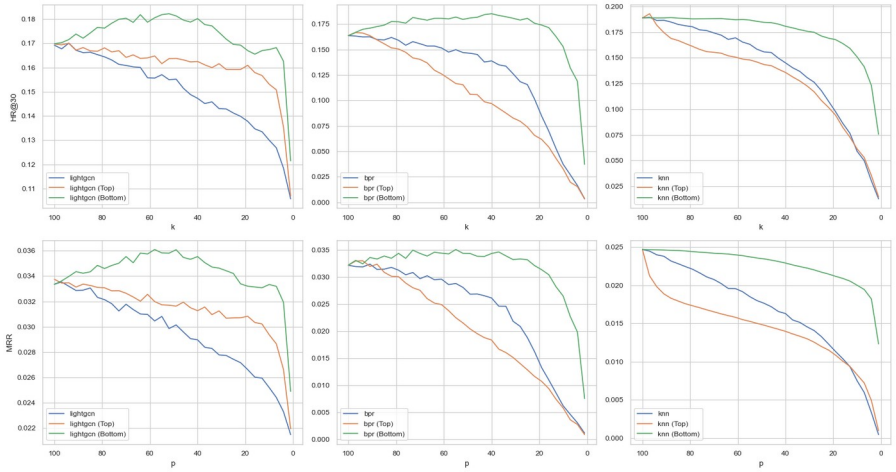

Рисунок 3 - Результаты для p
DOI: https://doi.org/10.60797/IRJ.2026.171.24.3

Обсуждение

Как может быть видно из графиков, результат разнится в зависимости от применяемой модели. При этом непосредственно в рамках одной модели наблюдаемые метрики демонстрируют одинаковую динамику.

Также стоит заметить, что для всех экспериментов свойственно падение метрик к 0 взаимодействий на пользователя. Это говорит как об адекватности моделей (отсутствие пользовательской истории приводит модели к случайным предсказаниям), так и о наличии некоторого оптимального значения длинны пользовательской истории.

Для вариации с k наиболее значимый вывод — результаты с фильтрацией по значению внимания почти полностью совпадают со случайной выборкой. Второй вывод — для отдельных моделей фильтрация позволяет сместить точку падения метрик, но в зависимости от самой модели лучшие результаты показывает фильтрация по максимальному вниманию (KNN), минимальному значению (BPR) или же вовсе не оказывают существенного влияния (LightGCN).

Интереснее выглядит вариация с p. Для всех трех моделей фильтрация показывает возможность снизить объем обучающих данных. При этом для LightGCN и BPR дополнительно наблюдается прирост метрик.

Недостатком же служит тот факт, что различные модели выдают различную динамику изменения метрик, что не дает универсальных рекомендаций к применению представленного подхода и требует проведения разведывательного анализа, что может в ряде сценариев свести к минимуму полезность применения представленного метода.

Заключение

В настоящей работе был исследован подход к сокращению объема обучающих данных рекомендательных систем на основе использования механизма внимания графовых нейронных сетей.

Проведенные эксперименты показали, что переиспользование оценок внимания модели GAT позволяет для представленных моделей сохранять качество при повышении скорости обучения.

Полученные результаты подтверждают перспективность использования механизмов внимания как инструмента оптимизации обучающих данных, особенно в условиях, в которых модели с механизмом внимания уже интегрированы в рекомендательные системы. Предложенный подход открывает возможности для снижения вычислительных затрат при обучении рекомендательных систем, ускорения подбора гиперпараметров и более эффективного применения ресурсоемких моделей в условиях регулярного обновления данных.

Развитием идеи, представленной в данной статье может послужить исследование других моделей с использованием внимания, например, вариантов генеративных и последовательных моделей.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы на английском языке / References in English

1. Zhang Y. Deep learning for recommender systems / Y. Zhang, X. Chen // ACM Computing Surveys. — 2020. — Vol. 52, № 1. — P. 1–38.

2. Zhang K. Explainable recommender systems: A survey and new perspectives / K. Zhang, L. Zheng, X. Wang // ACM Computing Surveys. — 2020. — Vol. 54, № 6. — P. 1–37.

3. Wang X. A survey on knowledge graph-based recommender systems / X. Wang, X. He, M. Wang // IEEE Transactions on Knowledge and Data Engineering. — 2021. — Vol. 34, № 8. — P. 3549–3578.

4. Chang B. Sequential recommendation: A survey of methods, applications, and evaluation / B. Chang, Y. Sun, Y. Zhu // ACM Computing Surveys. — 2021. — Vol. 54, № 8. — P. 1–40.

5. Wu L. Graph neural networks for recommender systems: Challenges, methods, and directions / L. Wu, J. Li, P. Sun // IEEE Transactions on Knowledge and Data Engineering. — 2022. — Vol. 35, № 4. — P. 3077–3104.

6. Gu Y. MiniLLM: Knowledge Distillation of Large Language Models / Y. Gu, L. Dong, F. Wei et al. // arXiv preprint. — 2023. — URL: https://arxiv.org/abs/2306.08543 (accessed: 12.06.26)

7. Yang C. Survey on Knowledge Distillation for Large Language Models: Methods, Evaluation, and Application / C. Yang, L. Wang, Y. Zhu et al. // ACM Computing Surveys. — 2024. — Vol. 57, № 5. — P. 35.

8. Veličković P. Graph attention networks / P. Veličković // arXiv preprint. — 2017. — URL: https://arxiv.org/abs/1710.10903. (accessed: 12.06.26)

9. Brody S. How attentive are graph attention networks? / S. Brody, U. Alon, E. Yahav // arXiv preprint. — 2021. — URL: https://arxiv.org/abs/2105.14491. (accessed: 12.06.26)

10. He X. Lightgcn: Simplifying and powering graph convolution network for recommendation / He X. et al. // Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. — 2020. — P. 639–648.

11. Rendle S. BPR: Bayesian personalized ranking from implicit feedback. / Rendle S. [et al.] // arXiv preprint. — 2012. — URL: https://arxiv.org/abs/1205.2618 (accessed: 12.06.2026).

12. Aggarwal C.C. Recommender systems / C.C. Aggarwal [et al.] — Cham : Springer International Publishing, 2016. — Vol. 1. — DOI 10.1007/978-3-319-29659-3

13. Ricci F. Introduction to recommender systems handbook / F. Ricci, L. Rokach, B. Shapira // Recommender systems handbook. — Boston, MA : Springer US, 2010. — P. 1–35.

14. MovieLens // GroupLens Research. — DOI: 10.1234/movielens.2024

15. Amazon Review Data // Jianmo Ni. — 2018. — URL: https://nijianmo.github.io/amazon/index.html (accessed: 12.06.2026).