ДОКЛАДЫ РОССИЙСКОЙ АКАДЕМИИ НАУК. МАТЕМАТИКА, ИНФОРМАТИКА, ПРОЦЕССЫ УПРАВЛЕНИЯ, 2023, том 514, № 2, с. 333–342

"MTC Kion": НАБОР ДАННЫХ ДЛЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ НА ОСНОВЕ НЕЯВНОГО ОТКЛИКА, КОНТЕКСТНЫХ ПРИЗНАКОВ И АНАЛИЗА ПОСЛЕДОВАТЕЛЬНОСТЕЙ

© 2023 г. И. Сафило[1,*], Д. Тихонович[2,], А. В. Петров[3,***], Д. И. Игнатов[4]**

Представлено академиком РАН А.И. Аветисяном
Поступило 30.05.2023 г.
После доработки 15.10.2023 г.
Принято к публикации 20.10.2023 г.

Мы представляем новый датасет для построения рекомендаций фильмов и сериалов, собранный на основе данных реальных пользователей платформы потокового вещания фильмов и сериалов MTC Kion. В отличие от других популярных датасетов для рекомендаций фильмов, таких как MovieLens или Netflix, наш датасет основан на неявных взаимодействиях, полученных во время просмотров контента пользователями, а не на явных оценках, поставленных фильмам. Мы также предоставляем обширную дополнительную информацию, включая характеристики взаимодействий (такие как дата взаимодействия, продолжительность просмотра и процент просмотра), демографические данные пользователей и подробную мета-информацию о фильмах. Кроме того, мы описываем MTS Kion Challenge – онлайн-соревнование по построению рекомендательных систем на основе датасета Kion, и предоставляем обзор лучших решений от победителей соревнования.

Ключевые слова: рекомендательные системы, нейронные сети, неявный отклик, последовательные рекомендации, коллаборативная фильтрация

DOI: 10.31857/S2686954323600428, **EDN:** CXTWPU

1. ВВЕДЕНИЕ

В современном мире рекомендательные системы помогают нам взаимодействовать с торговыми площадками, такими как Amazon или eBay, выбирать наиболее релевантный контент на платформах потокового воспроизведения, таких как Netflix или Spotify, и получать персонализированную ленту рекомендаций в социальных сетях, таких как Twitter или Facebook. *Рекомендация фильмов* – это особый вид проблемы персонализации, где алгоритмы выбирают следующие видео/фильмы/сериалы для просмотра. Современные системы рекомендаций основаны на моделях машинного обучения (например, на основе матричной факторизации [17] или на основе глубокого обучения [10, 21, 22])[1], и эти модели нужно обучать на данных о взаимодействии пользователей с товарами. Некоторые из моделей [13, 19] также могут использовать дополнительную контекстную и сопутствующую информацию, такую как описания товаров, характеристики пользователей или характеристики взаимодействия пользователя с товаром (например, день недели). Многие модели также основывают обучение на последовательностях действий пользователей [9, 10, 14, 16, 18, 21].

К сожалению, популярные наборы данных для рекомендаций фильмов, такие как MovieLens [8] и Netflix [2], не содержат необходимой таким моделям контекстной информации, а также предоставляют только явный отклик от пользователей. Поскольку мета-информация, предоставленная этими наборами данных, ограничена, они плохо работают с новыми пользователями и объектами. Еще одна проблема этих датасетов заключается в том, что они в некотором роде "искусственные" – данные содержат не реальные просмотры фильмов пользователями, а события проставления оценок. Таким образом, последовательная при-

[1]*MTS, Национальный исследовательский университет "Высшая школа экономики", Москва, Россия*
[2]*MTC, Москва, Россия*
[3]*University of Glasgow Glasgow, United Kingdom*
[4]*Национальный исследовательский университет "Высшая школа экономики" Москва Россия*
**E-mail: irsafilo@gmail.com*
***E-mail: daria.m.tikhonovich@gmail.com*
****E-mail: a.petrov.1@research.gla.ac.uk*

[1] Workshop on Context Aware Recommender Systems (CARS) In conjunction with 16th ACM Conference on Recommender Systems 23d September 2022 © Copyright held by the authors

рода потребления контента может нарушаться: назначение рейтингов может быть выполнено в другом порядке по сравнению с порядком реального просмотра фильмов. Часто пользователи смотрят фильмы, не ставя им рейтинг вообще. Поэтому последовательность оценок пользователя не всегда отражает эволюцию его интересов. Как утверждают Петров и Макдональд [14], с наборами данных, основанными на оценках пользователей, задача лучше описывается как *следующий фильм для оценки*, а не как *следующий фильм для просмотра*.

Мы представляем новый набор данных для задачи рекомендаций фильмов, который не имеет описанных выше ограничений[2]. Мы собрали данные от пользователей платформы потокового воспроизведения видео МТС Kion с 13.03.2021 по 22.08.2021. Набор данных включает 5.476.251 взаимодействий 962.179 пользователей с 15.706 объектами. Поскольку набор данных собран на основе просмотров из реального приложения онлайн кинотеатра, он отражает типичные особенности системы потокового вещания в реальном мире. Мы провели только минимальную предварительную обработку данных с целью сохранения конфиденциальности информации пользователей[3]. Например, в этом наборе данных можно наблюдать явные периодические зависимости и отчетливые временные тренды . Набор данных включает как “холодных”, так и “теплых” пользователей, а также как “холодные”, так и “теплые” объекты, как это часто бывает в приложениях реального мира. Мы предоставляем подробности об этих особенностях датасета в Разделе 2.3.

Благодаря разнообразной мета-информации, датасет позволяет проводить исследования качества построенных рекомендаций, выходящие за рамки простого измерения точности предсказаний. Например, мета-информация позволяет измерять такие метрики, как разнообразие, новизна и “serendipity” [1]. В отличие от других популярных наборов данных для рекомендаций фильмов, датасет Kion включает в себя демографические характеристики пользователей, такие как возраст, пол и уровень дохода, а также содержит разнообразные характеристики фильмов, такие как текстовые описания сюжета и списки режиссеров, актеров и стран производства. Эти мета-данные могут помочь анализировать глубокие отношения между пользователями и объектами. Например, мы обнаружили, что доля женщин среди зрителей фильма является сильным признаком для рекомендательной модели.

Мы использовали предлагаемый набор данных для соревнования по построению рекомендательных систем MTS Kion Challenge, которое проводилось с ноября 2021 г. по январь 2022 г. За время проведения соревнования 46 участников представили 844 решения. Мы представляем обзор победивших рекомендательных моделей в Разделе 3. Значимость мета-информации, представленной в наборе данных Kion, была подтверждена во время соревнования в каждом из победивших решений.

В целом набор данных Kion позволяет исследователям создавать более сложные модели и проводить более глубокие и интересные анализы по сравнению с другими существующими наборами данных.

Подводя итог, наш вклад можно описать следующим образом:

(1) Мы представляем исследователям новый набор данных для построения рекомендаций фильмов, объединяющий в себе неявный отклик от пользователей, реальные последовательности действий пользователей и мета-информацию как о самих пользователях, так и об объектах, с которыми они взаимодействовали.

(2) Мы анализируем ключевые особенности набора данных.

(3) Мы рассматриваем решения, которые были реализованы победителями соревнования по рекомендательным системам, которое мы проводили с использованием этого набора данных.

(4) Мы также описываем ключевые алгоритмы, использованные победителями соревнования.

В оставшейся части статьи мы предоставляем подробную информацию о наборе данных и соревновании Kion. В частности, Раздел 2 дает обзор ключевых характеристик датасета, включая разведочный анализ данных. Раздел 3 рассматривает соревнование MTS Kion Challenge и описывает решения выигравших участников. Раздел 4 содержит описание основных рекомендательных алгоритмов, которые были использованы в решениях. Раздел 5 содержит заключительные замечания.

2. НАБОР ДАННЫХ KION

2.1. Платформа потокового вещания фильмов и сериалов Kion

Для того чтобы помочь исследователям лучше понять природу нашего набора данных, мы кратко рассмотрим платформу Kion, которая предоставила нам данные.

[2] Датасет доступен для скачивания на Github: https://github.com/MobileTeleSystems/RecTools/tree/main/datasets/KIONhttps://github.com/MobileTeleSystems/RecTools/tree/main/datasets/KION

[3] Единственная обработка, которую мы проводим, это анонимизация и добавление небольшого количества шума для сохранения конфиденциальности пользователей Kion

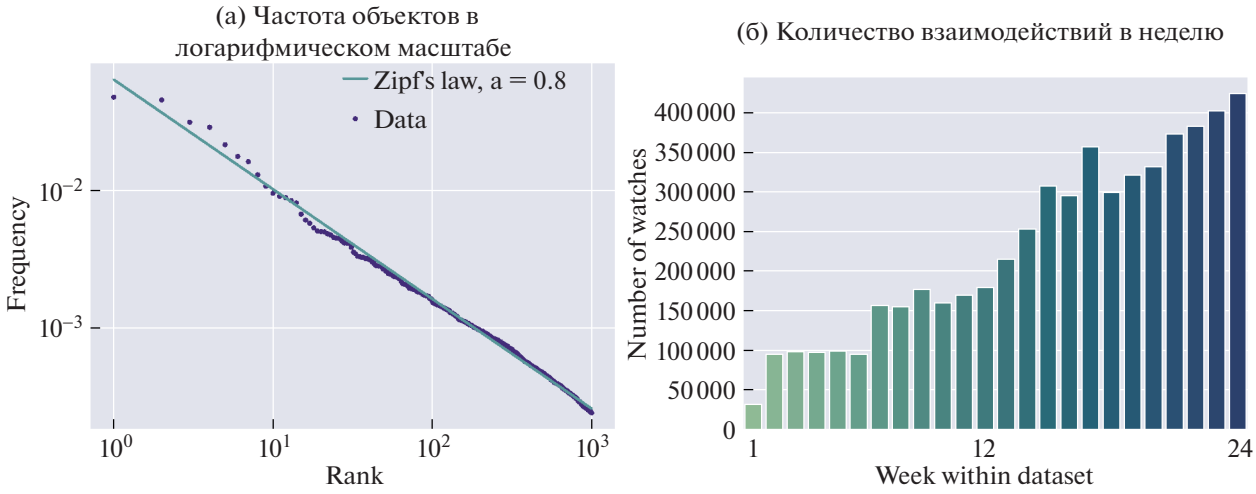

Рис. 1. Распределение взаимодействий.

Kion – одна из крупнейших платформ потокового видео в Восточной Европе с более чем 3.5 миллионами подписчиков. В то же время Kion – это относительно новая платформа, основанная холдингом МТС в 2021 г.; как результат, датасет содержит в данных тенденцию быстрого роста количества взаимодействий с момента основания сервиса до 1 миллиона пользователей. Теперь мы предоставим больше деталей о наборе данных.

2.2. Описание набора данных

Набор данных Kion – это датасет с информацией о взаимодействиях пользователей и объектов, построенный на неявном отклике от пользователей и включающий реальные последовательности их действий в приложении. В данных отражены взаимодействия за 5 мес, начиная с 13.03.2021 и заканчивая 22.08.2021. Датасет содержит 5.476.251 взаимодействий 962.179 пользователей с 15.706 объектами. Каждое взаимодействие описано датой, общей продолжительностью просмотра и процентом контента, просмотренного пользователем. В случае нескольких взаимодействий с одним и тем же объектом в набор данных включено только последнее по дате взаимодействие, однако общая продолжительность и процент агрегируются по всем просмотрам.

Мы также предоставляем дополнительную информацию как для пользователей, так и для объектов. Пользователи описаны демографической информацией, такой как возраст, пол, доход и пометкой “есть дети”. У объектов есть такие характеристики, как тип контента (фильм или сериал), название, название на языке оригинала, год выпуска, жанры, страны, возрастной рейтинг, студии, режиссеры, актеры, ключевые слова и описание.

2.3. Разведочный анализ данных

Наборы данных, основанные на взаимодействиях пользователей с фильмами, обычно имеют распределение просмотров по степенному закону, при котором несколько объектов доминируют в предпочтениях пользователей [20]. Рисунок 1а иллюстрирует этот факт для набора данных Kion, показывая частоты объектов в логарифмическом масштабе. Как показывает рисунок, данные можно довольно хорошо аппроксимировать распределением по закону Ципфа с параметром степени $\alpha = 0.8$.

Также важно подчеркнуть явные временные тренды популярности объектов. Рисунок 1b показывает растущее количество взаимодействий каждую неделю в наборе данных Kion. Рисунок 2 иллюстрирует индивидуальные скачки популярности топ-100 фильмов (за исключением сериалов) по неделям. Эти особенности распределения популярности во времени и среди объектов могут иметь заметные последствия для создания правильной схемы валидации и успешного решения задачи рекомендаций фильмов на таких данных.

Характеристики пользователей и объектов – еще один важный аспект для систем рекомендаций. Подавляющее большинство объектов в датасете имеют и характеристики, и взаимодействия, в то время как у 30 процентов пользователей есть только одно из вышеуказанных. Рисунок 3 иллюстрирует доступность данных как для пользователей, так и для объектов.

В целом набор данных Kion отражает специфические проблемы реальных рекомендательных систем, включая: (i) неявный отклик, (ii) распределение популярности объектов по закону Ципфа, (iii) временные всплески популярности отдельных объектов, (iv) проблему холодного

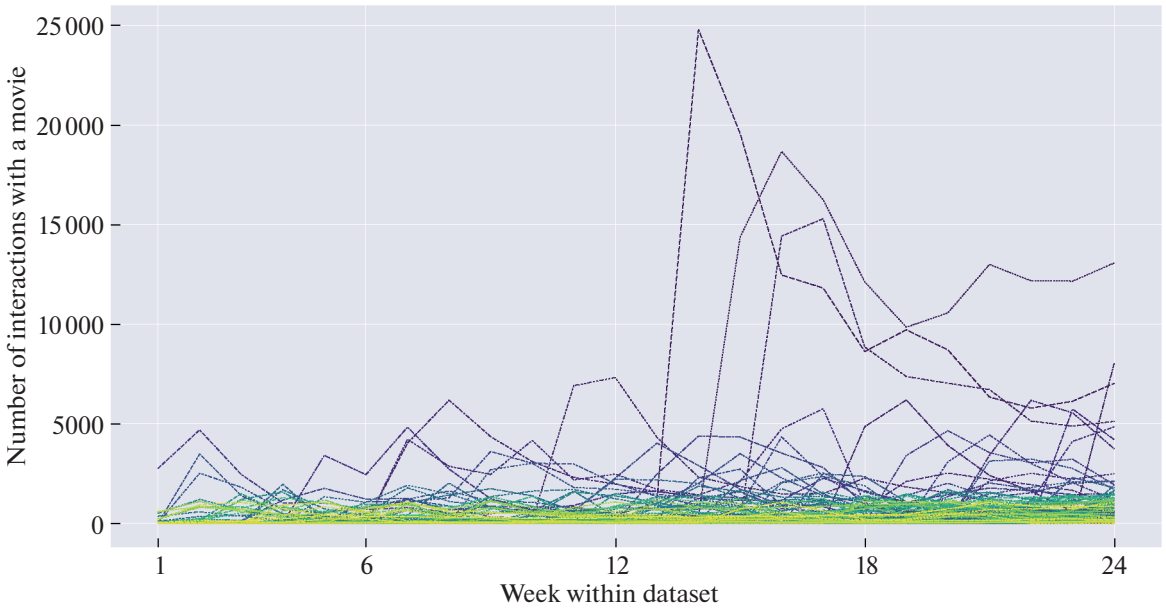

Рис. 2. Топ-100 фильмов: количество взаимодействий в неделю.

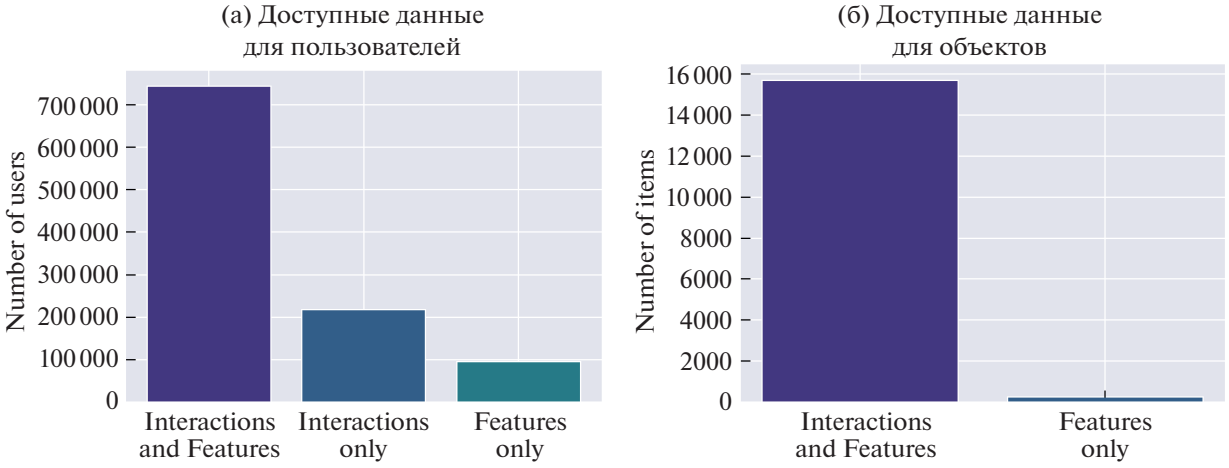

Рис. 3. Доступность данных о характеристиках и взаимодействиях.

старта как для пользователей, так и для объектов, (v) разнообразную, но неполную мета-информацию.

2.4. Сравнение с существующими наборами данных для рекомендаций фильмов

В этом разделе мы сравниваем набор данных Kion с другими наборами данных для рекомендаций фильмов, доступными до нашей работы. В частности, мы сравниваем набор данных Kion с MovieLens-25M, самым крупным набором данных из семейства MovieLens [8], и с набором данных Netflix, который был выпущен для известного соревнования Netflix Prize [2].

Сначала мы сравниваем количественные характеристики наборов данных, которые представлены в табл. 1. Как видно из таблицы, число пользователей в наборе данных Kion значительно больше по сравнению с Netflix (+100%) и Movielens (+493%). Однако при этом число взаимодействий в Kion меньше, чем в других двух наборах данных. Эти два факта приводят к значительно более коротким последовательностям взаимодействий: в Kion средняя длина последовательности составляет всего 5.69, по сравнению с 153.80 в MovieLens и 209.25 в Netflix. Мы утверждаем, что

Таблица 1. Количественные характеристики наборов данных

Набор данных	Пользователей	Объектов	Взаимодействий	Ср. длина последовательности	Разреженность
Netflix	480.189	17.770	100.480.507	209.25	98.82%
Movielens-25M	162.541	59.047	25.000.095	153.80	99.73%
Kion	**962.179**	**15.706**	**5.476.251**	**5.69**	**99.9%**

Таблица 2. Качественные характеристики наборов данных

Набор данных	Netflix	Movielens-25M	Kion
Тип отклика	Явный (Рейтинги)	Явный (Рейтинги)	Неявный (Взаимодействия)
Время добавления отклика	После просмотра	После просмотра	Во время просмотра
Данные о взаимодействиях	Дата, Рейтинг	Дата, Рейтинг	Дата, Продолжительность, Процент просмотра
Данные о пользователях	Нет	Нет	Пол, Возраст, Доход, Наличие детей
Данные об объектах	Год релиза, Название	Год релиза, Название, Жанры, Теги	Тип контента, Название, Название на языке оригинала, Год релиза, Жанры, Страны, Для детей, Возрастное ограничение, Студии, Режиссеры, Актеры, Описание, Ключевые слова

эти короткие последовательности и большое количество новых пользователей характерны для реальных рекомендательных систем, особенно на этапе быстрого роста сервиса.

Теперь сравним качественные характеристики наборов данных, которые представлены в табл. 2. Как видно, Netflix и MovieLens имеют довольно похожие характеристики: оба они представляют собой датасеты с явными оценками, собранными через некоторое время после просмотра фильма. Оба предоставляют только ограниченную контекстную информацию о фильмах и вообще не предоставляют данных о пользователях. Напротив, Kion является датасетом с неявным откликом, с событиями, зарегистрированными во время просмотра пользователями видео на платформе. Он включает в себя богатую мета-информацию как для пользователей, так и для объектов. Также он включает дополнительную информацию о событиях, такую как длительность и процент просмотра.

В целом, как можно увидеть из обеих табл. 1 и 2, набор данных Kion имеет значительные отличия в качественных и количественных характеристиках по сравнению с существующими наборами данных и может способствовать новым видам исследований в области рекомендаций фильмов, которые были невозможны с наборами данных MovieLens или Netflix.

3. СОРЕВНОВАНИЕ ПО РЕКОМЕНДАТЕЛЬНЫМ СИСТЕМАМ MTC KION CHALLENGE

3.1. Детали соревнования

Мы использовали набор данных Kion для онлайн-соревнования, которое проводилось с ноября 2021 г. по декабрь 2022 г. Мы организовали соревнование на платформе ODS.ai [4]. Основная цель соревнования заключалась в прогнозировании фильмов, которые пользователи посмотрят в течение одной недели после периода, охваченного набором данных. Мы хранили эти взаимодействия в *приватной* части набора данных, которая не была доступна участникам.

3.1.1. Метрика конкурса. В качестве основной метрики производительности мы использовали *Mean Average Precision* с порогом K ($MAP@K$), которую измеряли на приватном наборе данных. Одна треть пользователей из приватного набора данных была представлена "холодными" пользователями (т.е. эти пользователи не имели взаимодействий во время периода, доступного для обучения моделей). Рисунок 4 показывает распределение пользователей по количеству взаимодействий в публичном наборе данных.

Average Precision – это метрика, основанная на точности. Рассмотрим список рекомендаций $i_1, i_2, \ldots, i_n$ для пользователя с бинарными метками ре-

[4] https://ods.ai/competitions/competition-recsys-21https://ods.ai/competitions/competition-recsys-21

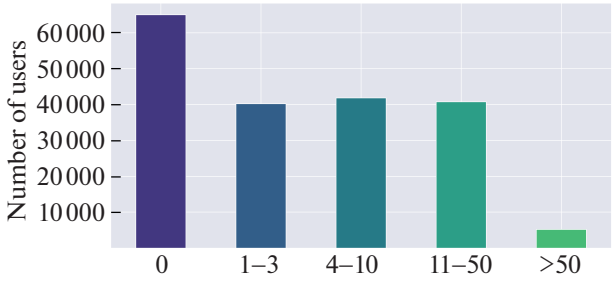

Рис. 4. Количество взаимодействий на пользователя в тестовом наборе данных.

левантности $r_1, r_2, \ldots, r_n$. Тогда *Precision* с порогом K ($P@K$) определяется как:

$$P@K = \frac{\sum_{i=1}^{k} r_i}{K} \quad (1)$$

Average Precision с порогом K ($AP@K$) для пользователя определяется путем усреднения Точностей и их нормализации:

$$AP@K(user) = \frac{1}{C_{user}} \sum_{i=1}^{K} P@i \cdot r_i \quad (2)$$

где C_{user} – общее количество релевантных элементов для пользователя в приватном наборе данных. Чтобы получить *Mean Average Precision* с порогом K, мы усредняем метрику $AP@K$ по всему набору данных.

$$MAP@K = \sum_{i=1}^{N} AP@K(user_i) \quad (3)$$

где N – это количество пользователей в приватном наборе данных. Для конкурса мы выбрали порог 10, поэтому основной метрикой конкурса была $MAP@10$.

3.2. Песочница конкурса

Мы не публиковали приватный набор данных, однако песочница конкурса оставалась открытой[5], так что другие исследователи могли отправить свои решения и проверить свои результаты $MAP@10$ на приватной части набора данных.

В следующем разделе мы предоставляем обзор победивших решений, представленных участниками соревнования.

[5] https://ods.ai/competitions/competition-recsys-21/leaderboard/public_sandbox.

3.3. Лучшие решения

Таблица 3 содержит результаты пяти лучших участников соревнования, измеренные на приватном наборе данных. Все победители использовали двухэтапный подход, генерируя кандидатов для рекомендаций с помощью одной или нескольких базовых моделей на первом этапе и реранжировали их с помощью градиентного бустинга над решающими деревьями на втором этапе, используя выходы моделей первого этапа вместе с дополнительными вручную сконструированными признаками.

3.3.1. Ансамбли с нейронными сетями. Участники Олег Лашигин и Александр Петров, занявшие первое и второе места в итоговом рейтинге, использовали двухэтапные ансамбли с несколькими моделями рекомендаций на первом этапе. В ансамбли входили мощные модели, основанные на глубоком обучении, такие как BERT4Rec [21], SASRec [10] и Caser [22], и традиционные методы рекомендаций, такие как Байесовское персонализированное ранжирование [17]. Олег использовал реализации нейронных моделей из библиотеки RecBole [23] и реализации некоторых моделей, опубликованных Dacrema et al. [4], тогда как Александр использовал реализации, описанные в статье о воспроизводимости [15].

Для объединения моделей первого этапа участники использовали градиентный бустинг над решающими деревьями из библиотеки LightGBM [11]. Олег использовал обучение с целью классификации, а Александр использовал ранжирование по LambdaRank [3].

Оба победивших участника дополнили результаты моделей первого этапа дополнительными признаками, созданными вручную, такими как время с момента последнего взаимодействия пользователя или доля мужчин среди зрителей фильма-кандидата в рекомендации.

3.3.2. Ансамбли без нейронных сетей. Участник Степан Зимин использовал тот же подход, что и два лучших победителя, создавая ансамбль моделей первого этапа, но не включил в него модели глубокого обучения. Его решение основывалось на выводах пяти различных базовых моделей, в том числе гибридной модели матричной факторизации LightFM [12], использующей признаки пользователей и объектов в процессе матричного разложения. Он обогатил результаты моделей разнообразной статистикой взаимодействий как для пользователей, так и для объектов, и дополнительно учитывал жанры объектов при создании признаков для ансамбля. Для финальных прогнозов он использовал градиентный бустинг над решающими деревьями из библиотеки CatBoost [6]

Таблица 3. Решения и результаты победителей соревнования

Позиция	Имя	MAP@10	Тип решения
1	Олег Лашинин	0.1221	Ансамбль с нейронными сетями
2	Александр Петров	0.1214	Ансамбль с нейронными сетями
3	Степан Зимин	0.1213	Ансамбль без нейронных сетей
4	Дарья Тихонович	0.1148	Градиентный бустинг над решающими деревьями
5	Ольга	0.1135	Градиентный бустинг над решающими деревьями
Популярный бейзлайн		**0.0910**	

с целью YetiRank [7] и достиг показателя MAP@10, который почти идентичен двум лучшим решениям.

Участники Дарья Тихонович и Ольга, занявшие четвертое и пятое места в соревновании, достигли конкурентоспособных показателей MAP@10 с единственной моделью на первом этапе рекомендательной системы. Дарья использовала модель ближайших соседей (Item KNN) по расстояниям между объектами в матрице взаимодействий пользователей с объектами [5], а Ольга использовала простую популярную модель. Оба участника использовали градиентный бустинг над решающими деревьями из библиотеки CatBoost с учебной целью классификации на втором этапе. Дарья добавила вручную сконструированные признаки на основе трендов популярности объектов (например, количество взаимодействий за последню неделю, наклон тренда популярности и расстояние в днях до 0.95 квантиля дат распределения взаимодействий с объектом). Ольга использовала идентификаторы объектов из истории взаимодействий пользователя в качестве признаков. Оба участника также добавили демографические данные пользователей.

Подводя итог, пять лучших решений соревнования на наборе данных Kion включили в себя широкий спектр подходов, включая последовательные модели рекомендаций (BERT4Rec, SASRec, Caser), контекстуализированные модели (LightFM) и классические подходы к рекомендациям (матричная факторизация, коллаборативная фильтрация). Все победители обогатили результаты работы модели разнообразными признаками, используя метаданные пользователей и объектов, а также статистику взаимодействий. Это широкое разнообразие успешных подходов к задаче рекомендаций фильмов в соревновании показывает ценность набора данных Kion для разработчиков рекомендательных систем и всеобъемлющую природу его данных. Далее мы опишем основные алгоритмы, которые были использованы победителями соревнования.

4. ОСНОВНЫЕ АЛГОРИТМЫ

4.1. LambdaRank: Ранжирование с использованием градиентного бустинга над решающими деревьями

LambdaRank [3] представляет собой метод ранжирования, который оптимизирует стандартные метрики качества ранжирования, такие как NDCG. Этот метод базируется на идее преобразования проблемы ранжирования в задачу классификации или регрессии.

Основная идея. Вместо оптимизации функции потерь, которая напрямую связана с метрикой качества ранжирования, LambdaRank использует подход, при котором оптимизируется гладкая аппроксимация этой метрики.

Процесс обучения. Во время обучения для каждой пары объектов в обучающем наборе вычисляется значение “lambda” (отсюда и название метода). Это значение представляет собой приближенный градиент ошибки ранжирования по отношению к текущим предсказаниям модели. После этого стандартные методы (например, градиентный бустинг) могут использоваться для оптимизации модели на основе вычисленных значений “lambda”.

Формализация

Для пары объектов i и j, если i имеет более высокий рейтинг, чем j, то определяется следующая функция потерь:

$$\mathcal{L}(i, j) = -\frac{1}{1 + \exp(\sigma(s_i - s_j))} \tag{4}$$

где s_i и s_j – это оценки релевантности объектов i и j, а σ – параметр масштабирования.

“Lambda”, которая вычисляется для пары объектов, представляет собой приближенное изменение в метрике ранжирования, если бы порядок этой пары изменился. Это значение затем используется как градиент для оптимизации деревьев решений.

4.2. LightFM: Гибридная модель матричной факторизации

LightFM [12] представляет собой гибридную модель, которая может использовать как явные, так и неявные отклики и совместно сочетать матричную факторизацию с характеристиками пользователей и объектов.

Математическая модель:

Пусть u обозначает пользователя, а i обозначает товар. LightFM представляет каждого пользователя u и каждый товар i в виде векторов латентных признаков p_u и q_i соответственно. Предсказание модели для пары пользователь-товар определяется скалярным произведением их латентных векторов:

$$\hat{y}_{ui} = p_u^{\top} q_i \tag{5}$$

В отличие от стандартных моделей матричной факторизации, LightFM интегрирует признаки пользователей и объектов. Пусть X_u и Y_i обозначают векторы признаков для пользователя u и объекта i соответственно. Тогда латентные представления определяются как:

$$p_u = \sum_{j \in X_u} \alpha_j e_j \tag{6}$$

$$q_i = \sum_{k \in Y_i} \beta_k f_k \tag{7}$$

где e_j и f_k являются векторами латентных признаков для характеристик j пользователя и k объекта, а α_j и β_k – это веса, связанные с этими характеристиками.

Функция потерь:

Для обучения модели используется функция потерь на основе логистической регрессии:

$$L = -\sum_{(u,i) \in D} y_{ui} \log(\sigma(\hat{y}_{ui})) + (1 - y_{ui}) \log(1 - \sigma(\hat{y}_{ui})) \tag{8}$$

где σ обозначает сигмоидную функцию. **Оптимизация**:

Для оптимизации параметров модели LightFM часто использует стохастический градиентный спуск (SGD). Он позволяет модели масштабироваться для больших наборов данных, делая обновления на основе случайных мини-пакетов данных.

Холодный старт:

Одним из наиболее значимых преимуществ LightFM является его способность справляться с проблемой холодного старта, когда новые пользователи или товары добавляются в систему без какой-либо предыдущей истории взаимодействий. Благодаря включению признаков пользователей и товаров LightFM может предсказывать рейтинги для новых пользователей или товаров на основе их признаков, даже если у модели нет истории взаимодействий с ними.

LightFM предлагает уникальное сочетание матричной факторизации и признаков пользователей/товаров, что делает его мощным инструментом для рекомендательных систем, особенно в сценариях с холодным стартом.

4.3. BERT4Rec: BERT для последовательных рекомендаций

BERT4Rec [21] представляет собой адаптацию архитектуры BERT, настроенную для задач последовательных рекомендаций.

Концептуальные основы. В области рекомендаций история взаимодействия пользователя может быть воспринята как последовательность. BERT4Rec обучается предсказывать следующий товар в последовательности взаимодействий пользователя, используя возможности архитектуры Transformer для нахождения сложных зависимостей в этих последовательностях.

Архитектура модели. BERT4Rec использует архитектуру Transformer, состоящую из нескольких слоев с механизмом внимания. Данной последовательности взаимодействий $s = (i_1, i_2, \ldots, i_t)$, модель создает контекстные вложения для каждого товара в последовательности. Целью во время обучения является предсказание следующего товара, i_{t+1}, на основе текущего контекста.

Для выполнения задачи предсказания BERT4Rec в процессе обучения “маскирует” некоторые объекты в последовательности и пытается их предсказать, аналогично цели маскированной языковой модели в оригинальном BERT.

Для данной последовательности некоторый процент объектов “маскируется”. Цель модели – правильно предсказать эти “маскированные” объекты на основе их контекста. Формально, для последовательности s и “маскированных” позиций M, целью является:

$$\mathcal{L} = -\sum_{i \in M} \log P(i \mid s_{\neg i}; \theta) \tag{9}$$

где $s_{\neg i}$ обозначает последовательность с удаленным элементом i, и θ представляет параметры модели.

4.4. SASRec: Последовательная модель рекомендаций на основе механизма внимания

SASRec [10] – это модель для последовательных рекомендаций, основанная на механизме внимания из архитектуры Transformer.

Концептуальные основы. В задачах последовательных рекомендаций важно учитывать порядок взаимодействий пользователя с объектами. SASRec использует механизм внимания для модели-

рования зависимостей между различными объектами в последовательности без учета предварительно заданных временных шкал.

Архитектура модели. SASRec состоит из нескольких блоков Transformer. Для входной последовательности взаимодействий $s = (i_1, i_2, \ldots, i_t)$, модель создает контекстные вложения для каждого объекта последовательности.

Механизм внимания. Механизм внимания позволяет модели учитывать различные объекты в последовательности с разной степенью внимания. Формально, вес внимания между объектами i и j в последовательности рассчитывается как:

$$\alpha_{ij} = \frac{\exp(\text{score}(i, j))}{\sum_{k=1}^{t} \exp(\text{score}(i, k))} \quad (10)$$

где score(i, j) – это скалярное произведение вложений объектов i и j.

Целью в процессе обучения является предсказание следующего объекта в последовательности. Для элемента i_t в последовательности s, целью является максимизация вероятности следующего объекта, i_{t+1}:

$$\mathcal{L} = -\log P(i_{t+1} | s_1, s_2, \ldots, s_t; \theta) \quad (11)$$

где θ представляет параметры модели.

4.5. Caser: Персонализированные рекомендации на основе сверточных сетей

Caser [22] представляет собой модель для последовательных рекомендаций, которая использует сверточные нейронные сети для учета порядка товаров в истории пользователя.

Основная идея. Традиционные последовательные рекомендательные модели часто не учитывают геометрическую или временную структуру данных. Caser был разработан для учета последовательности и порядка товаров в истории пользователя.

Архитектура модели. Caser использует два ключевых компонента: горизонтальные и вертикальные свертки. Горизонтальные свертки призваны улавливать соседние товары, тогда как вертикальные свертки улавливают долгосрочные зависимости.

Предположим, что у нас есть последовательность товаров $s = (i_1, i_2, \ldots, i_t)$. Вначале каждый товар в последовательности представлен его векторным вложением, формируя матрицу вложений.

Сверточные слои. Горизонтальные свертки применяются к каждому вложению, чтобы уловить локальные зависимости между соседними товарами. Вертикальные свертки применяются ко всей матрице вложений, чтобы уловить глобальные зависимости по всей истории пользователя.

Как и в других последовательных рекомендательных моделях, целью обучения Caser является предсказание следующего товара на основе истории пользователя. Для товара i_t в последовательности s цель состоит в максимизации вероятности следующего элемента, i_{t+1}:

$$\mathcal{L} = -\log P(i_{t+1} | s_1, s_2, \ldots, s_t; \theta) \quad (12)$$

где θ представляет параметры модели.

5. ЗАКЛЮЧЕНИЕ

Подводя итог, мы представляем новый набор данных для разработки рекомендательных систем от потоковой платформы МТС Kion. Мы сравнили его с наборами данных MovieLens-25M и Netflix и пришли к выводу, что наш набор данных лучше отражает особенности данных в реальных приложениях: (i) неявный отклик от пользователей, (ii) распределение популярности объектов по закону Ципфа, (iii) временные пики популярности отдельных объектов, (iv) проблема холодного старта как для пользователей, так и для объектов, (v) разнообразные, но неполные метаданные. Мы также описали лучшие решения соревнования MTS Kion Challenge, которые были построены на нашем наборе данных, и описали использованные победителями алгоритмы. Мы надеемся, что наша статья станет стимулом для дальнейших исследований в области текстуализированных и последовательных рекомендательных систем.

БЛАГОДАРНОСТИ

Мы хотели бы выразить благодарность участникам MTS Kion Challenge Олегу Лашинину, Степану Зимину и Ольге за предоставление описаний их решений в MTS Kion Challenge, холдингу МТС за предоставление набора данных Kion, международной платформе ODS.ai за организацию соревнования. Вклад Ильдара Сафило и Дмитрия И. Игнатова в статью был сделан в рамках Программы фундаментальных исследований НИУ ВШЭ.

СПИСОК ЛИТЕРАТУРЫ

1. *Sivan Yogev Asela Gunawardana, Guy Shani*. Evaluating Recommendation Systems. 2022. P. 547–601. https://doi.org/10.1007/978-0-387-85820-3_8
2. *James Bennett and Stan Lanning.* The Netflix Prize. In Proc. KDD. 2007.
3. *Christopher JC Burges*. 2010. From RankNet to LambdaRank to LambdaMart: An overview. Learning. 2010. V. 11. P. 23–581, 81.
4. *Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach*. Are we really making much progress?

A worrying analysis of recent neural recommendation approaches. In Proc. RecSys. 2019. P. 101–109.

5. *Mukund Deshpande and George Karypis*. 2004. Item-based top-n recommendation algorithms. ACMTransactions on Information Systems (TOIS). 2004. V. 22. № 1. P. 143–177.
6. *Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin*. 2018. CatBoost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363 (2018).
7. *Andrey Gulin, Igor Kuralenok, and Dimitry Pavlov*. Winning the transfer learning track of yahoo!'s learning to rank challenge with yetirank. In Proc. Learning to Rank Challenge. PMLR, 2011. P. 63–76.
8. *F Maxwell Harper and Joseph A* Konstan. 2015. The MovieLens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TIIS). 2015. V. 5. № 4. P. 1–19.
9. *Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk*. Session-based Recommendations with Recurrent Neural Networks. In Proc. ICLR. 2016.
10. *Wang-Cheng Kang and Julian McAuley*. Self-attentive sequential recommendation. In Proc. ICDM. 2018. P. 197–206.
11. *Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu*. LightGBM: A highly efficient gradient boosting decision tree. In Proc. NeurIPS. 2017. P. 3146–3154.
12. *Maciej Kula*. Metadata embeddings for user and item cold-start recommendations. arXiv preprint arXiv:1507.08439. 2015.
13. *Jiacheng Li, Yujie Wang, and Julian McAuley*. Time interval aware selfattention for sequential recommendation. In Proceedings of the 13th international conference on web search and data mining. 2020. P. 322–330.
14. *Aleksandr Petrov and Craig Macdonald*. Effective and Efficient Training for Sequential Recommendation using Recency Sampling. Proc. RecSys. 2022.
15. *Aleksandr Petrov and Craig Macdonald*. A Systematic Review and Replicability Study of BERT4Rec for Sequential Recommendation. In Proc. RecSys. 2022.
16. *Aleksandr Petrov and Yuriy Makarov*. Attention-based neural re-ranking approach for next city in trip recommendations. In Proc. WSDM WebTour. 2021. P. 41–45.
17. *Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme*. BPR: Bayesian personalized ranking from implicit feedback. In Proc. CUAI. 2009. P. 452–461.
18. *Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme*. Factorizing personalized markov chains for next-basket recommendation. In Proc. WWW. 2010. P. 811–820.
19. *Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme*. Fast context-aware recommendations with factorization machines. In Proc. SIGIR. 2011. P. 635–644.
20. *Harald Steck, Linas Baltrunas, Ehtsham Elahi, Dawen Liang, Yves Raimond, and Justin Basilico*. Deep learning for recommender systems: A Netflix case study. AI Magazine. 2021. V. 42. № 3. P. 7–18.
21. *Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang*. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proc. CIKM. 2019. P. 1441–1450.
22. *Jiaxi Tang and Ke Wang*. Personalized top-n sequential recommendation via convolutional sequence embedding. In Proc. WSDM. 2018. P. 565–573.
23. *Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al*. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In Proc. CIKM. 2021. P. 4653–4664.

MTS KION IMPLICIT CONTEXTUALISED SEQUENTIAL DATASET FOR MOVIE RECOMMENDATION

I. Safilo[a], D. Tikhonovich[b], A. V. Petrov [c], and D. I. Ignatov[d]

[a]*MTS, HSE University, Moscow, Russian Federation*

[b]*MTS, Moscow, Russian Federation*

[c]*University of Glasgow, Glasgow, United Kingdom*

[d]*HSE University, Moscow, Russian Federation*

Presented by Academician of the RAS A.I. Avetisyan

We present a new movie and TV show recommendation dataset collected from the real users of MTS Kion video-on-demand platform. In contrast to other popular movie recommendation datasets, such as MovieLens or Netflix, our dataset is based on the implicit interactions registered at the watching time, rather than on explicit ratings. We also provide rich contextual and side information including interactions characteristics (such as temporal information, watch duration and watch percentage), user demographics and rich movies meta-information. In addition, we describe the MTS Kion Challenge – an online recommender systems challenge that was based on this dataset – and provide an overview of the best performing solutions of the winners. We keep the competition sandbox open, so the researchers are welcome to try their own recommendation algorithms and measure the quality on the private part of the dataset.