МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра дискретной математики и алгоритмики

ЗАЛЕВСКИЙ Александр Александрович

РАЗРАБОТКА СИСТЕМЫ РЕКОМЕНДАЦИЙ МЕДИА ОБЪЕКТОВ ПО ИСТОРИИ ПРОСМОТРОВ

Магистерская диссертация

Специальность 1-31 80 09 «Прикладная математика и информатика»

Научный руководитель
Мушко Вилена Владимировна
канд. физ.-мат. наук

Допущена к защите
«___» _____ 2024 г.
Заведующий кафедрой дискретной математики и алгоритмики
_____В. М. Котов
доктор физико-математических наук, профессор

Минск, 2024

ОГЛАВЛЕНИЕ

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Магистерская диссертация, 43 с., 12 рис., 11 табл., 13 источников.

РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ, МАШИННОЕ ОБУЧЕНИЕ, КОЛЛАБОРАТИВНАЯ ФИЛЬТРАЦИЯ, ITEM-TO-ITEM, ЛОНГ-ТЕРМИНАЛЬНАЯ ПАМЯТЬ, ПЕРСОНАЛИЗИРОВАННЫЕ РЕКОМЕНДАЦИИ, АЛГОРИТМЫ РАНЖИРОВАНИЯ, ИСТОРИЯ ПОЛЬЗОВАТЕЛЬСКОГО ВЗАИМОДЕЙСТВИЯ

Объект исследования – методы и модели рекомендательных систем, применяемые для рекомендации медиа-объектов на основе истории просмотров пользователей.

Цель работы – изучение различных методов рекомендаций и разработка системы рекомендаций фильмов на основе данных из онлайн-кинотеатра Kion.

В ходе работы рассматриваются следующие подходы рекомендации медиа-объектов на основе истории просмотров пользователей:

1. коллаборативная фильтрация на основе объектов;
2. альтернативные наименьшие квадраты;
3. рекомендации на основе сингулярного разложения;
4. градиентный бустинг.

Результаты работы – реализация системы рекомендаций фильмов на основе методов item-to-item, implicit alternate least squares, singular value decomposition, CatBoostRanker, применение этих методов на предоставленных данных, оценка качества моделей и сравнительный анализ.

АГУЛЬНАЯ ХАРАКТАРЫСТЫКА ПРАЦЫ

Магістарская дысертацыя, 43 с., 12 мал., 11 табл., 13 крыніц.

РЭКАМЕНДАТАРНЫЯ СІСТЭМЫ, МАШЫННАЕ НАВУЧАННЕ, КАЛАБАРАТЫЎНАЯ ФІЛЬТРАЦЫЯ, ITEM-TO-ITEM, ДАЎГАЧАСОВАЯ ПАМЯЦЬ, ПЕРСАНАЛІЗАВАНЫЯ РЭКАМЕНДАЦЫІ, АЛГАРЫТМЫ РАНЖЫРОЎКІ, ГІСТОРЫЯ КАРЫСНІЦКАГА ВЗАІМАДЗЕЙСТВІЯ

Аб'ект даследаванням - метады і мадэлі рэкамендацыйных сістэм, якія выкарыстоўваюцца для рэкамендацыі медыя-аб'ектаў на аснове гісторыі праглядаў карыстальнікаў.

Мэта працы - вывучэнне розных метадаў рэкамендацыі і распрацоўка сістэмы рэкамендацыі фільмаў на аснове дадзеных з онлайн-кінатэатра Kion.

У ходзе працы разглядаюцца наступныя падыходы да рэкамендацыі медыя-аб'ектаў на аснове гісторыі праглядаў карыстальнікаў:

1. калябаратыўная фільтрацыя на аснове аб'ектаў;
2. альтэрнатыўныя найменшыя квадраты;
3. рэкамендацыі на аснове сінгулярнага разбягнення;
4. градыентны бустынг.

Вынікі працы - рэалізацыя сістэмы рэкамендацыі фільмаў на аснове метадаў item-to-item, implicit alternate least squares, singular value decomposition, CatBoostRanker, прымяненне гэтых метадаў на прадастаўленых дадзеных, ацэнка якасці мадэляў і параўнальны аналіз.

ABSTRACT

Master's thesis, 43 p., 12 ill., 11 tab., 13 sources.

RECOMMENDER SYSTEMS, MACHINE LEARNING, COLLABORATIVE FILTERING, ITEM-TO-ITEM, LONG-TERM MEMORY, PERSONALIZED RECOMMENDATIONS, RANKING ALGORITHMS, USER INTERACTION HISTORY

Object of research – methods and models of recommender systems used for recommending media items based on user viewing history.

Purpose of work – overview of recommendation methods and development of a movie recommendation system based on data from the online cinema Kion.

The following approaches to recommending media items based on user viewing history are considered in the work:

1. item-based collaborative filtering;
2. alternate least squares;
3. recommendations based on singular decomposition;
4. gradient boosting.

Results of work – implementation of a movie recommendation system based on the methods of item-to-item, implicit alternate least squares, singular value decomposition, and CatBoostRanker, the application of these methods to the provided data, evaluation of model quality, and comparative analysis.

ВВЕДЕНИЕ

Рекомендательные системы представляют собой значимое направление исследований в области информационных технологий, это обусловлено тем, что интернет технологии постоянно развиваются и всё чаще появляется доступ к самым разным видам мультимедийного контента. В связи с тем, что пользователям в современном информационном обществе приходится всё чаще сталкиваться с большими объёмами информации, из которых им становится необходимо подбирать более подходящие элементы, рекомендательные системы становятся более важным инструментом для улучшения качества жизни пользователя, оптимизации его времени и улучшения качества его взаимодействия с медийным контентом.

Очевидно, что современное информационное пространство всё время наполняется медийным контентом, доступ к этому контенту становится более обширным, это связано с непрерывным развитием информационных технологий. Это развитие, как следствие, создаёт сложности для пользователей в поиске для них контента, который им подходит, который соответствует их интересам и их предпочтениям. Это связано с тем, что большое разнообразие контента создаёт трудности при принятии пользователем решения о выборе наиболее подходящего для себя контента.

Таким образом, рекомендательные системы становятся важным инструментом, который помогает пользователям ориентироваться в этом большом разнообразии контента. Основываясь на поведении пользователя, на его предпочтениях и на его опыте взаимодействия с различными объектами, рекомендательные системы строят для него персонализированные рекомендации. Хорошие рекомендации повышают удовлетворённость пользователя и таким образом повышают его вовлечённость в платформу или сервис.

Кроме того, у рекомендательных систем повышается их точность и эффективность. Развитие технологий машинного обучения и анализа данных позволяет им использовать алгоритмы и модели, которые способны лучше адаптироваться к предпочтениям пользователей, которые изменяются, учитывать их меняющиеся интересы.

Следует отметить, что рекомендательные системы могут применяться в различных сферах жизни. В развлекательной сфере они хорошо подходят для рекомендаций фильмов, музыкальных композиций, литературных произведений. У пользователя отпадает необходимость тратить большое

количество времени на то, чтобы просматривать большое количество фильмов с целью выбрать тот, который ему понравится. Кроме того, путём предложения индивидуальных и привлекательных рекомендаций компании могут убедить пользователей остаться на их платформе вместо того, чтобы перейти на платформу конкурентов, либо воспользоваться их услугами вместо услуг конкурентов. Таким образом, рекомендательные системы могут выступать хорошим инструментом удержания клиентов, сохранения конкурентных позиций, снижая отток пользователей. Пользователи, получающие персональные рекомендации, будут чаще оставаться на платформе, будут в течение более продолжительного периода времени взаимодействовать с контентом, будут более удовлетворены этим контентом. В коммерции рекомендательные системы могут анализировать предпочтения каждого клиента и его историю покупок, предлагать те товары либо услуги, которые с наибольшим шансом вызовут у него интерес. Таким образом, рекомендательные системы могут сыграть важную роль в повышении конверсии и среднего чека, а также улучшить взаимодействие с покупателем, делая его более удовлетворённым.

В сфере образования системы рекомендаций, например, могут оказывать помощь студентам в выборе курсов и учебных материалов. Они могут основываться на индивидуальных предпочтениях, уровне знаний студента. Таким образом они могут оптимизировать процесс самостоятельного обучения, повысить эффективность обучения. Наконец, рекомендательные системы могут найти применение в медицине. Они могут, основываясь на истории заболевания и эффективности назначаемого лечения, помогать врачу в принятии решения о лечении пациента.

Целью данного исследования является изучение различных методов рекомендаций и разработка системы рекомендаций фильмов на основе данных из онлайн-кинотеатра Kion.

Задачи исследования:

1. Изучить различные методы и подходы к построению рекомендательных систем.
2. Изучить и проанализировать предоставленные данные о фильмах и об историях просмотров пользователей онлайн кинотеатра Kion.
3. Изучить методы коллаборативной фильтрации на основе объектов, неявных альтернативных наименьших квадратов, сингулярного разложения, градиентного бустинга.
4. Применить данные методы на предоставленных данных.

5. Сравнить полученные результаты, выбрать наилучшие модели.
6. Определить оптимальный метод или их комбинацию для создания хорошей системы рекомендаций.

ГЛАВА 1
АНАЛИЗ НАУЧНОЙ ЛИТЕРАТУРЫ ПО ТЕМЕ ИССЛЕДОВАНИЯ

1.1 Обзор существующих методов рекомендательных систем

Выделяются три основных типа методов рекомендательных систем. Коллаборативные методы основываются на истории взаимодействия и определяют схожесть объектов или пользователей на основе этой истории взаимодействия. Суть контентных методов заключается в анализе характеристик объектов и профилей пользователей. Гибридные методы объединяют в себе подходы коллаборативных и контентных методов.

1.1.1 Коллаборативные методы

Коллаборативная фильтрация на основе пользователей

Принцип метода коллаборативной фильтрации на основе пользователей (user-based collaborative filtering) состоит в анализе сходства пользователей, поиске похожих пользователей, использовании предпочтений пользователей, чтобы рекомендовать им новый контент. В этом методе предполагается, что к выбору похожего контента склонны пользователи, которые имеют похожие предпочтения, то есть такие пользователи, у которых похожи истории их взаимодействий. Метод использует оценки, которые пользователь поставил различным объектам для того, чтобы найти похожих пользователей и на основании их оценок для какого-то определённого объекта предсказывает оценку этого пользователя для этого объекта.

Преимуществом этого метода является простота его реализации и возможность работы с неполными данными, так как он не использует информацию о характеристиках объектов и о профилях пользователей. С другой стороны, недостатком является то, что этот метод плохо решает проблему холодного старта, то есть плохо рекомендует объекты с малым количеством взаимодействий и плохо строит рекомендации для новых пользователей, у которых малая история взаимодействий.

Коллаборативная фильтрация на основе объектов

В отличие от метода коллаборативной фильтрации на основе пользователей, принцип метода коллаборативной фильтрации на основе

объектов (item-based collaborative filtering) состоит в анализе сходств объектов, поиске похожих объектов, использовании предпочтений пользователей об объектах. В этом методе предполагается, что пользователи склонны к выбору объектов похожих на те, с которым они положительно провзаимодействовали, то есть таких объектов, истории взаимодействий которых похожи на истории взаимодействий тех объектов, которые понравились пользователю. Метод использует оценки, которые различные пользователи поставили объекту для того, чтобы найти похожие объекты, и предсказывает оценку, которую поставил бы пользователь объекту на основании тех оценок, которые этот пользователь поставил похожим объектам.

Матричное разложение

Матричное разложение (matrix factorization) может использоваться для моделирования взаимосвязей пользователями и объектами. Суть его заключается в разложении матрицы пользователей-объектов в произведение двух небольших матриц. Первая матрица, как правило, содержит скрытые представления каждого пользователя, а вторая – скрытые представления каждого объекта. Такие модели называются моделями скрытых представлений (latent factor models). Ниже представлен обзор трёх методов, основанных на матричном разложении – это альтернативные наименьшие квадраты, сингулярное разложение, неотрицательное матричное разложение.

Альтернативные наименьшие квадраты (ALS)

Метод альтернативных наименьших квадратов (alternate least squares) является итеративным методом матричного разложения, который минимизирует сумму квадратов ошибок между исходными оценками пользователей и их приближениями, и состоит в повторении следующих четырёх шагов:

1. зафиксировать матрицу скрытых представлений пользователей;
2. решить задачу регрессии для каждого объекта и найти оптимальную матрицу скрытых представлений объектов;
3. зафиксировать матрицу скрытых представлений объектов;
4. решить задачу регрессии для каждого пользователя и найти оптимальную матрицу скрытых представлений пользователей.

Таким образом, попеременное вычисление точных аналитических решений позволяет получить оптимальные матрицы скрытых представлений пользователей и объектов.

В качестве преимуществ метода альтернативных наименьших квадратов выделяют то, что он легко параллелится, а также масштабируется для работы с большими датасетами. С другой стороны, недостатком является его склонность

к переобучению, особенно если использовать большой размер скрытых представлений пользователей и объектов.

Сингулярное разложение (SVD)

Сингулярное разложение (singular value decomposition) – это такой метод разложения матрицы в произведение левой сингулярной матрицы U, диагональной матрицы Σ и правой сингулярной матрицы V. Можно применить это разложение к матрице рейтингов, тогда матрицу скрытых представлений пользователей можно представить как $U\Sigma^{1/2}$, а матрицу скрытых представлений объектов как $\Sigma^{1/2}V$. Тогда $U\Sigma^{1/2} \cdot \Sigma^{1/2}V = U\Sigma V = \hat{R}$ – матрица оценок рейтингов. Чтобы построить k рекомендаций пользователю, нужно взять в соответствующей строке матрицы номера k самых больших значений.

У SVD есть два существенных недостатка по сравнению с методом альтернативных квадратов. Он не предполагает обновления рекомендаций при добавлении новых пользователей или новых объектов. Это значит, что при добавлении новых пользователей или новых объектов придётся переучивать модель заново. Во-вторых, его труднее распараллелить.

Неотрицательное матричное разложение (NMF)

Неотрицательное матричное разложение (non-negative matrix factorization) – ещё один метод разложения матрицы на две матрицы. Особенность состоит в том, что матрица неотрицательная. Этот метод также может использоваться в контексте рекомендательных систем. С помощью него можно извлекать скрытые представления пользователей и объектов. Он может использоваться в качестве альтернативы методам ALS и SVD, на практике выбор метода определяется конкретной задачей, которая решается на конкретных данных.

1.1.2 Контентные методы

Другим типом рекомендательных моделей являются контентные методы. В отличие от коллаборативных моделей, которые основаны на анализе истории взаимодействий пользователей и объектов, контентные методы основаны на анализе характеристик объектов и профилей пользователей. В качестве характеристик фильмов, например, могут выступать название, жанр, текстовое описание фильма, ключевые слова, страна, в которой был снят фильм. В качестве профиля пользователя могут выступать его пол, возраст, размер дохода либо характеристики тех фильмов, которые пользователь посмотрел и которые ему понравились.

Для анализа названия фильмов и текстового описания фильмов хорошим подходом будет использовать методы обработки естественного языка – natural language processing, NLP. С помощью инструментов NLP можно строить векторные представления текстовых описаний фильмов, классифицировать их, вытаскивать из текстовых описаний ключевые слова, после чего строить векторные представления уже для этих ключевых слов или классифицировать их. Далее можно анализировать векторные представления и информацию о принадлежности фильма к тому или иному классу для того, чтобы искать похожие фильмы или использовать их в качестве признаков для оценки подходящести тех или иных фильмов пользователю. У текстовых статей, например, можно таким же образом использовать текстовый заголовок и само текстовое содержание статьи. У различных видов контента в социальных сетях, например, можно использовать текстовые комментарии.

Помимо текстовых описаний также можно анализировать прочие признаки объектов, если они есть. Например, для фильмов это могут быть жанр, тема, стиль, год выпуска, режиссёр, автор и так далее.

По сравнению с коллаборативными методами, контентные методы лучше справляются с рекомендациями объектов с малым количеством взаимодействий, потому что в отличие от коллаборативных методов они основаны не на анализе взаимодействий, а на анализе содержания.

1.1.3 Гибридные методы

Гибридные методы основаны на объединении коллаборативных и контентных методов. Они могут использовать преимущества одних методов для того, чтобы скомпенсировать недостатки других. Например, они могут использовать преимущества контентных методов в их лучшей по сравнению с коллаборативными способности решать проблему холодного старта, для того чтобы скомпенсировать этот недостаток коллаборативных методов.

Объединение методов может происходить разными способами, ниже рассмотрены некоторые из них.

Гибридизация на мета-уровне

Идея гибридизации на мета-уровне (meta-level) состоит в том, чтобы взять более простую модель, заранее её обучить, и после этого использовать эту предобученную модель в качестве входных данных для другой рекомендательной модели.

Комбинация признаков

Идея комбинации признаков (feature combination) состоит в том, чтобы аналогично гибридизации на мета-уровне использовать предобученную модель, но в отличие от этого подхода, комбинация признаков подразумевает использование именно признаков предобученной модели в качестве части признаков для другой рекомендательной модели.

Смешанная гибридизация

Идея смешанной гибридизации состоит в том, чтобы получить списки рекомендаций различных моделей независимо, затем, объединив их, сформировать один список рекомендаций.

Каскадная гибридизация

Идея каскадной гибридизации (cascade hybridization) подразумевает использование выходных данных одной модели в качестве входных данных для другой. Вторая модель уточняет выходные данные первой модели. В контексте рекомендательных систем, например, эта идея может быть реализована следующим образом. Первая модель выступает в качестве фильтра для отбора кандидатов в итоговый список рекомендаций, а вторая используется для того, чтобы более точно анализировать кандидатов и более точно формировать итоговый список рекомендаций.

Переключаемая гибридизация

Идея переключаемой гибридизации (switching hybridization) подразумевает обучение различных моделей и использование на выбор той или иной в зависимости от типа текущего запроса.

Например, в рекомендательных системах для фильмов отдельно может работать модель, которая строит рекомендации для новых пользователей с большим упором в контент, так как у новых пользователей малая или вовсе отсутствует история просмотров. Другой пример: в зависимости от того, является ли пользователь ребёнком, отдельная модель может заточена под рекомендации фильмов, предназначенных для детей.

Взвешенная гибридизация

Идея взвешенной гибридизации (weighted hybridization) состоит в том, что оценки разных моделей объединяются с разными весами для построения одного итогового списка рекомендаций. Для каждой модели задаётся её вес, который определяет степень важности рекомендаций конкретно этой модели. Эти веса, в зависимости от конкретной задачи, могут быть выбраны заранее или подобраны таким образом, чтобы улучшить значения определённых метрик.

1.2 Метрики оценки качества рекомендаций

Оценки качества рекомендаций используются для того, чтобы понять, насколько хорошими являются рекомендации. Цель использования метрик качества рекомендаций состоит в том, чтобы с разных сторон сравнить различные методы рекомендаций, сравнить их сильные и слабые стороны и выбрать самые подходящие в зависимости от конкретной задачи.

Почти всегда рекомендации будут неидеальными. Разные рекомендательные модели будут ошибаться на разных пользователях и на разных объектах, и в разной степени. В зависимости от того, как это посчитать, рекомендации тех или иных моделей будут лучше или хуже. Поэтому, прежде чем выбирать наилучшие модели на основе значений метрик качества рекомендаций, нужно сначала выбрать подходящие метрики качества, которые наиболее точно подходят под решение конкретной задачи.

1.2.1 Точность

Точность на k (Precision@k) показывает долю релевантных документов среди топ-k рекомендованных. Чем меньше объектов модель ошибочно посчитает релевантными, тем больше будет значение Precision@k.

Стоит отметить, что эта метрика полезна в ситуации, когда есть большое множество объектов, среди которых сравнительно небольшое количество релевантных.

$$\text{Precision@k} = \frac{\text{количество релевантных элементов в топ-k}}{\text{количество элементов в топ} - k}$$

1.2.2 Полнота

Полнота на k (Recall@k) показывает долю рекомендованных документов среди всех релевантных. В сравнении точности и полноты, если значение точности будет тем больше, чем меньше объектов модель ошибочно посчитает релевантными, то значение полноты будет тем больше, чем меньше объектов модель ошибочно посчитает нерелевантными.

$$\text{Recall@k} = \frac{\text{количество релевантных элементов в топ-k}}{\text{общее количество релевантных элементов}}$$

1.2.3 F1-мера

В ситуации, когда важно учитывать одновременно и метрику точности, и метрику полноты, можно использовать метрику F1-меры. F1-мера является средним гармоническим двух упомянутых выше метрик.

$$\mathrm{F}_1 = 2\frac{Precision * Recall}{Precision + Recall}$$

F1-меру может быть предпочтительнее использовать, чем по-отдельности точность и полноту, потому что модели удобно сравнивать, когда их качество выражено одним числом.

1.2.4 Fβ-мера

В то время как F1-мера учитывает одинаковую важность точности и полноты, данная метрика может учитывать важность точности и полноты в заданном соотношении с помощью параметра β.

$$F_\beta = (\beta^2 + 1)\frac{Precision * Recall}{\beta^2 * Precision + Recall}$$

Fβ-мера является обобщением F1-меры. При β=1 точность и полнота учитываются одинаково, при 0<β<1 больший акцент делается на точности, а при β>1 больший акцент делается на полноте.

1.2.5 Метрика MAP@k

Главная проблема метрики точности состоит в том, что она не учитывает позицию релевантного элемента в топе, другими словами, неважно, будет ли релевантный элемент первым в списке рекомендаций или последним – это не повлияет на значение метрики. Метрика MAP@k решает эту проблему. Идея состоит в том, чтобы посчитать сумму Precision@i для всех i из списка рекомендаций для пользователя, где i-тый объект является релевантным, а затем усреднить по всем пользователям.

$$\mathrm{MAP@k} = \frac{1}{|U|}\sum_{u \in U}\frac{1}{|R_u|}\sum_{i \in R_u}\mathrm{Precision@}i \cdot rel_i$$

где:

- U – множество всех пользователей;
- R_u – множество, состоящее из k рекомендованных объектов для пользователя u.
- rel_i – индикатор, равный 1, если i-тый объект релевантный, иначе 0.

1.2.6 Покрытие

Покрытие (coverage) отражает долю количества уникальных рекомендованных объектов среди всех объектов.

$$\text{Coverage} = \frac{\text{Количество уникальных рекомендованных элементов}}{\text{Общее количество уникальных элементов}}$$

Эта метрика позволяет сравнить, насколько разнообразные объекты рекомендует рекомендательная система. Более высокое значение в общем смысле означает более разнообразные рекомендации.

Покрытие может быть полезно использовать на разных интервалах времени, например, можно посчитать покрытие в первый день или в первую неделю работы рекомендательной системы. В зависимости от задачи может быть полезно через определённое время рекомендовать пользователю повторно те объекты, с которыми он уже провзаимодействовал. В рекомендациях текстовых статей или фильмов это может не быть полезным, так как пользователям вряд ли будет интересно пересматривать заново фильм или перечитывать статью, но это может быть полезно, например, в рекомендациях музыки, потому что часто пользователям может быть интересно переслушивать наиболее понравившиеся песни.

Метрика покрытия может использоваться в связке с работой с холодным стартом. Так как новым пользователям обычно будут рекомендоваться самые популярные объекты, чтобы новые объекты показывались чаще, может быть хорошей идеей в течение некоторого времени добавлять их в рекомендации.

1.2.7 Новизна

Идея вычисления новизны (novelty) состоит в том, что объект тем более вероятно будет новым для пользователя, чем менее популярен этот объект.

Следовательно, для каждого объекта вычисляется вероятность P_i того, что он будет порекомендован

$$P_i = \frac{m_i}{|U|}$$

где:

- |U| – количество пользователей;
- m_i – количество показов объекта i.

Для каждого пользователя усредняется значение логарифма этой вероятности по всем рекомендованным ему объектам, полученный результат усредняется по всем пользователям.

$$\text{Novelty@k} = \frac{1}{|U|} \sum_{u \in U} \frac{1}{|R_u|} \sum_{i \in R_u} \log(P_i)$$

где:

- U – множество всех пользователей;
- R_u – множество, состоящее из k рекомендованных объектов для пользователя u.

Эта метрика позволяет сравнить, не просто насколько разнообразные объекты рекомендует рекомендательная система, а насколько разнообразные и насколько новые объекты рекомендуются каждому пользователю.

1.2.8 Подобие внутри списка

Ещё один способ оценить разнообразие рекомендаций – метрика подобия внутри списка (intra list similarity, ILS). Идея состоит в том, чтобы вычислить сумму похожестей по всем парам объектов в списке рекомендаций, затем усреднить это значение по всем пользователям

$$\text{ILS@k} = \frac{1}{|U|} \sum_{u \in U} \sum_{i,j \in R_u} \text{sim}(i,j)$$

где:

- U – множество всех пользователей;

– R_u – множество, состоящее из k рекомендованных объектов для пользователя u;
– $\mathrm{sim}(i,j)$ – похожесть между объектами i и j.

В качестве похожести пары объектов обычно используется косинусная похожесть.

$$\mathrm{sim}(i,j) = \frac{\mathrm{i} \cdot \mathrm{j}}{||i|| \cdot ||\mathrm{j}||}$$

Таким образом, чем ниже значение ILS, тем ниже в среднем по пользователям и по парам рекомендованных объектов их похожесть, и тем выше разнообразие.

1.3 Обратная связь в рекомендательных системах

Обратная связь – это какой-то результат взаимодействия пользователя с тем объектом, с которым он провзаимодействовал. Обратную связь ещё могут называть фидбеком, рейтингом. Например, для фильмов – рейтинг от 1 до 5, для товара – добавление товара в корзину или приобретение, для музыки – добавление в плейлист или лайк/дизлайк, или же доля времени прослушивания от времени всей песни. Тогда для задачи рекомендации можно, используя известные рейтинги, попытаться предсказать те рейтинги, которые пользователь поставил бы тем объектам, с которыми он не взаимодействовал, если бы провзаимодействовал.

Обратную связь разделяют на два типа: явную и неявную.

1.3.1 Явная обратная связь

Явная обратная связь (explicit feedback) – это такой результат взаимодействия пользователя, который явно указывает на то, понравился ли ему объект или нет. Например тот факт, что пользователь поставил лайк на песню, высокую оценку фильму или написал отзыв на приобретённый товар, говорит о том, что объект ему понравился. Свойство такого типа обратной связи состоит в том, что здесь ясно понятно, что пользователю точно понравилось, а что нет. С другой стороны, такой обратной связи очень мало, потому что лишь небольшая часть пользователей обычно ставит лайки, оценки, пишет отзывы, стоит

ожидать, что для большого числа непопулярных товаров явного фидбека может вообще не быть.

1.3.2 Неявная обратная связь

Неявная обратная связь (implicit feedback) – это любая обратная связь, которая не является явной. Такую обратную связь не оставляет пользователь явно, а она специально формируется на основе каких-то фактов, которые характеризуют взаимодействие пользователя и объекта. Например, тот факт, что пользователь несколько раз переслушал одну и ту же песню, досмотрел фильм до конца, провёл много времени на странице с определённым товаром или провёл много времени на странице с определённой статьёй, не говорит о том, что эта песня, фильм, товар или статья ему нравится, но на основании этой информации можно предполагать, что эта песня, фильм, товар или статья понравилась большому количеству пользователей.

У неявной связи есть преимущество. Оно состоит в том, что неявной обратной связи намного больше по сравнению с явной. Но стоит помнить, что такой фидбек на самом деле не говорит прямо о том, что объект пользователю нравится. Это может приводить к тому, что вместо задачи рекомендации тех объектов, которые пользователю понравятся, будет решена задача рекомендации того, что вызовет у пользователя интерес. На первый взгляд может показаться, что разницы в этом нет, но на самом деле есть. Например, тот факт, что человек провёл много времени, чистая статью, может говорить о том, что в этой статье ему наоборот трудно найти необходимую информацию. Если в качестве фидбека учитывать клики по заголовку статьи, они могут быть связаны с тем, что только сам заголовок вызывает интерес. Другой пример, человек может смотреть фильм в онлайн-кинотеатре или короткое видео в социальной сети до конца не потому, что ему оно нравится, а потому что ему интересно, что будет в конце.

1.4 Основные результаты и выводы

В данной главе проведён анализ научной литературы по теме работы. Проведён обзор существующих методов рекомендательных систем. Рассмотрены различные существующие метрики оценки качества рекомендаций. Рассмотрены различные типы обратной связи в рекомендательных системах.

ГЛАВА 2
ОПИСАНИЕ ДАННЫХ

В данн работе были использованы данные, которые были предоставлены в онлайн-соревновании RecSys Course Competition, которое проводилось на платформе Open Data Science. В датасете содержится информация о пользователях, фильмах и взаимодействиях пользователей и фильмов в онлайн-кинотеатре Kion.

2.1 Пользователи

В таблице пользователей содержатся следующие поля.

1. user_id – уникальный идентификатор пользователя
2. age – возрастная группа, к которой принадлежит пользователь
   - 18_24 – возраст от 18 до 24 лет
   - 25_34 – возраст от 25 до 34 лет
   - 35_44 – возраст от 35 до 44 лет
   - 45_54 – возраст от 45 до 54 лет
   - 55_64 – возраст от 55 до 64 лет
   - 65_inf – возраст от 65 лет
3. sex – пол
   - М – мужской
   - Ж – женский
4. income – доход пользователя
   - income_0_20 – доход до 20 у.е.
   - income_20_40 – доход от 20 до 40 у.е.
   - income_40_60 – доход от 40 до 60 у.е.
   - income_60_90 – доход от 60 до 90 у.е.
   - income_90_150 – доход от 90 до 150 у.е.
   - income_150_inf – доход от 150 у.е.
5. kids_flg – наличие ребёнка
   - 1 – есть ребёнок
   - 0 – нет ребёнка

Таблица содержит данные о 840197 пользователях, распределение пользователей по различным признакам приведено на рисунке 2.1.

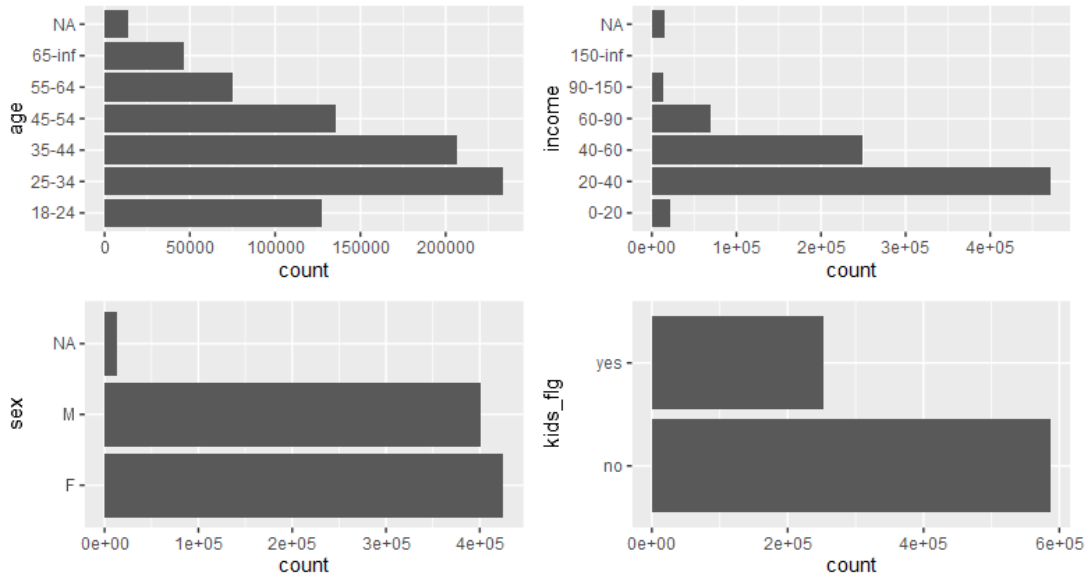

Рисунок 2.1 – Распределения групп возраста, дохода, пола, наличия ребёнка у пользователей.

2.2 Объекты

В таблице объектов содержится следующая информация о фильмах/сериалах.

1. item_id – уникальный идентификатор объекта
2. content_type – тип объекта
   - film – фильм
   - series – сериал
3. title – название на русском
4. title_orig – оригинальное название
5. genres – список жанров
6. countries – список стран
7. for_kids – контент для детей
8. age_rating – возрастной рейтинг
9. studios – студии
10. directors – режиссёры
11. actors – актёры
12. keywords – ключевые слова
13. description – текстовое описание

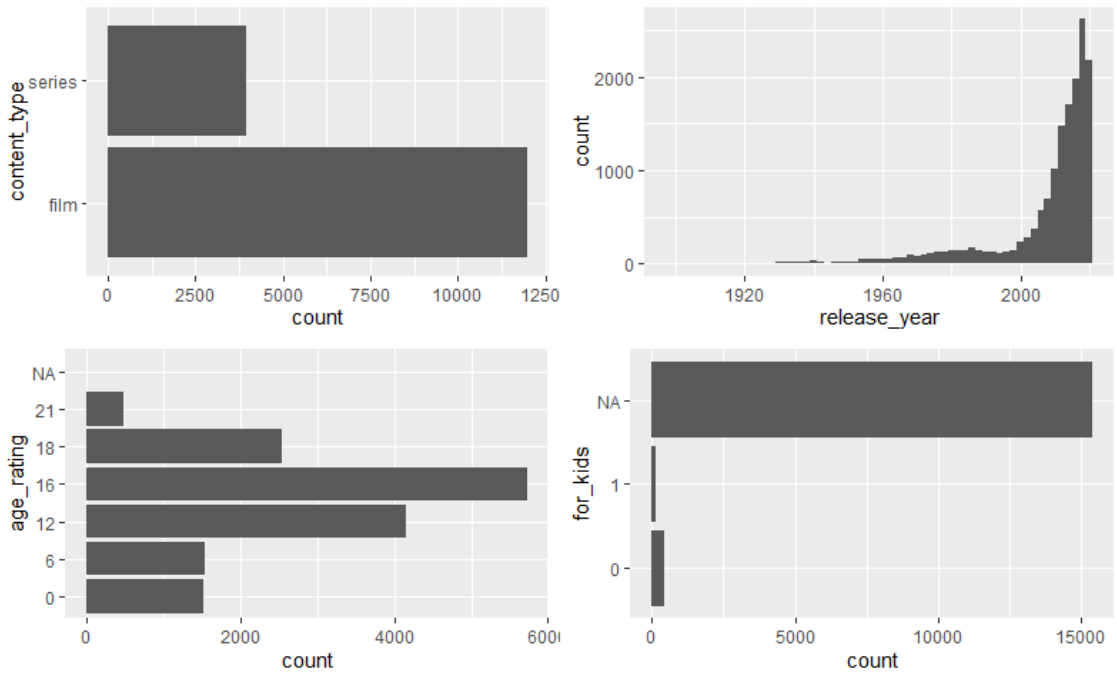

Рисунок 2.2 – Распределения групп типа контента, года выпуска фильмов, возрастного рейтинга и детского контента.

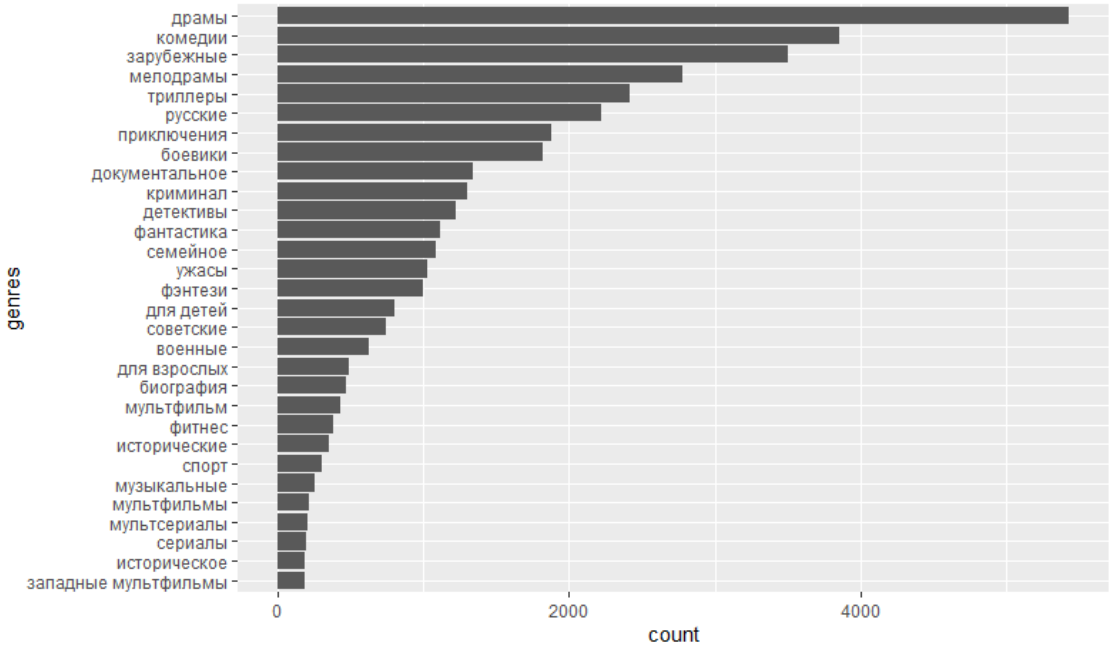

Рисунок 2.3 – Популярность среди фильмов самых популярных жанров.

2.3 Взаимодействия

В таблице взаимодействий содержится следующая информация о просмотрах.

1. user_id – уникальный идентификатор пользователя

2. item_id – уникальный идентификатор объекта
3. last_watch_dt – дата просмотра (если просмотров было несколько, учитывается последний из них)
4. total_dur – продолжительность просмотра в секундах (если просмотров было несколько, учитывается суммарная продолжительность всех просмотров
5. watched_pct – доля времени просмотра от продолжительности фильма/сериала.

На рисунке 2.4 приведено распределение доли просмотра (%) по всем просмотрам пользователей.

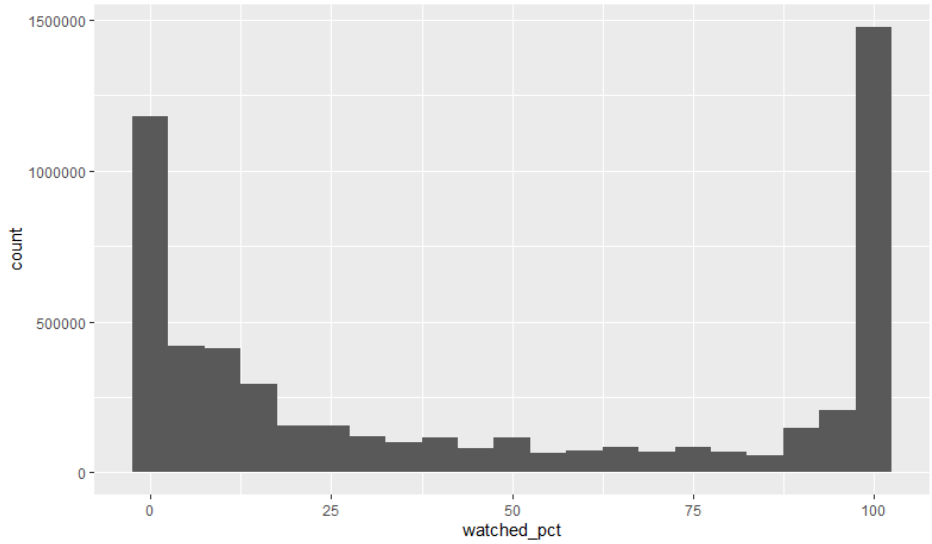

Рисунок 2.4 – Распределение доли просмотра (%) по всем просмотрам пользователей.

На рисунке 2.5 приведено распределение даты просмотра по всем просмотрам пользователей.

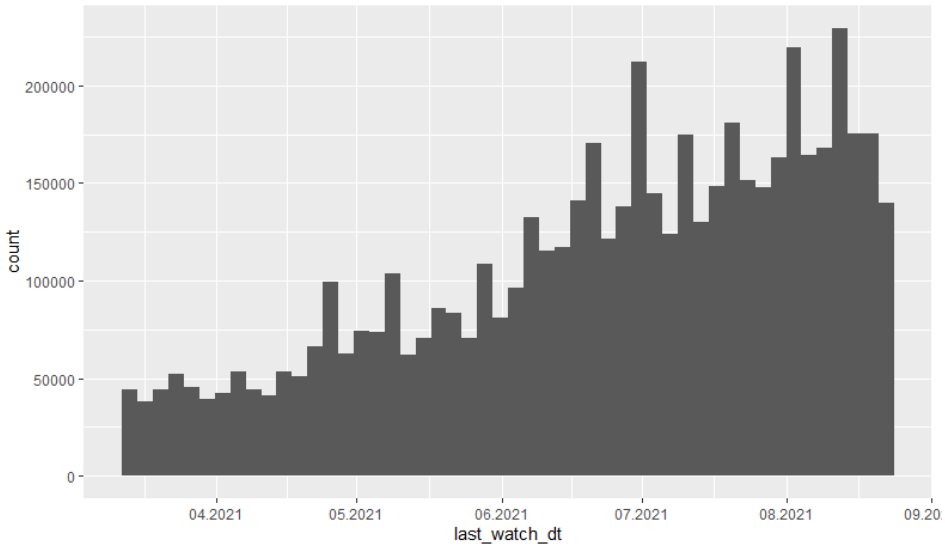

Рисунок 2.5 – Распределение даты просмотра по всем просмотрам пользователей.

На рисунке 2.6 приведено распределение числа пользователей в зависимости от количества просмотренных фильмов.

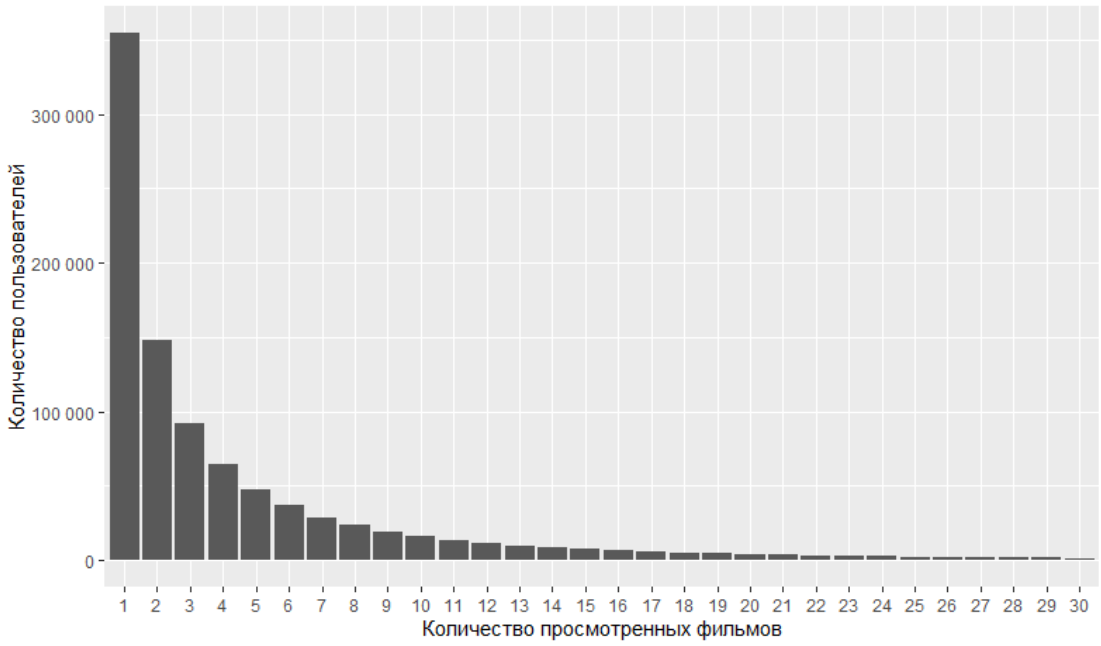

Рисунок 2.6 – Распределение числа пользователей в зависимости от количества просмотренных фильмов.

2.4 Основные результаты и выводы

В данной главе подробно описаны используемые данные о пользователях, фильмах и просмотрах. Перечислены поля в таблице пользователей, а также

распределения групп возраста, дохода, пола, наличия ребёнка у пользователей. Перечислены поля в таблице фильмов, а также распределения групп типа контента, года выпуска фильмов, возрастного рейтинга, детского контента популярностей самых популярных жанров. Перечислены поля таблицы взаимодействий, а также распределение доли просмотра (%) по всем просмотрам пользователей, распределение числа пользователей в зависимости от количества просмотренных фильмов, распределение числа пользователей в зависимости от количества просмотренных фильмов.

ГЛАВА 3
ПОСТРОЕНИЕ РЕКОМЕНДАЦИЙ ОБЪЕКТОВ НА ОСНОВЕ ИСТОРИИ ВЗАИМОДЕЙСТВИЙ

3.1 Реализация методов

3.1.1 Метод item-to-item

Используем матрицу пользователей-объектов R, где элемент R_{ui} задаёт степень неявного взаимодействия пользователя u с объектом i. В рассматриваемой задаче в качестве степени взаимодействия используется параметр watched_pct – доля просмотра фильма пользователем, а также α и β – настраиваемые параметры.

$$R_{ui} = \begin{cases} \alpha \cdot watched\_pct \,/\, 100 \,+\, \beta, \text{если пользователь } u \text{ посмотрел фильм } i \\ 0, \text{иначе} \end{cases}$$

где:

- watched_pct – доля просмотра фильма i пользователем u;
- α и β – настраиваемые параметры.

В качестве представления объекта используется соответствующий столбец матрицы R. В качестве меры похожести для каждой пары объектов используется косинусная похожесть.

$$\text{sim}(i, j) = \frac{R_i \cdot R_j}{||R_i|| \cdot ||R_j||}$$

где:

- $R_i \cdot R_j$ – скалярное произведение векторов R_i и R_j;
- $||R_i||$ – норма вектора R_i.

Для тех объектов, с которыми пользователь не провзамодействовал, предсказывается рейтинг

$$\hat{R}_{ui} = \frac{\sum_{j \in N(i)} \text{sim}(i, j) \cdot R_{uj}}{\sum_{j \in N(i)} |\text{sim}(i, j)|}$$

где:

– $\mathrm{sim}(i, j)$ – похожесть между фильмами i и j;
– N(i) – множество топ-N самых похожих фильмов на фильм i, а N – настраиваемый параметр.

После этого на основании предсказанных рейтингов, для каждого пользователя строятся рекомендации.

3.1.2 Метод IALS

Используем матрицу пользователей-объектов R, где элемент R_{ui} задаёт степень неявного взаимодействия пользователя u с объектом i. В качестве степени взаимодействия используется параметр watched_pct – доля просмотра фильма пользователем.

$$R_{ui} = \begin{cases} watched\_pct \;/\; 100, & \text{если пользователь } u \text{ посмотрел фильм } i \\ 0, & \text{иначе} \end{cases}$$

Зададим матрицу предпочтений P, где

$$P_{ui} = \begin{cases} 1, & \text{если } R_{ui} > 0 \\ 0, & \text{иначе} \end{cases}$$

Зададим матрицу C, которая будет использоваться для того, чтобы учитывать степень уверенности пользователей

$$C_{ui} = \alpha |R_{ui}| + \beta$$

где α и β – настраиваемые параметры.

Задача состоит в том, чтобы оптимизировать следующую функцию.

$$\sum_{\forall u,i} C_{ui}(P_{ui} - U_u \cdot V_i)^2 + \lambda \sum_{\forall u} ||U_u||^2 C_u + \lambda \sum_{\forall i} ||V_i||^2 C_i \rightarrow min$$

Оптимизация происходит последовательным выполнением на каждой итерации следующих двух шагов: сначала вычисляется новая матрица U, затем новая матрица V по следующим формулам.

$$U_u = \left(V^t V + \lambda C_u I + \sum_{\forall i:\, P_{ui} \neq 0} (C_{ui} - 1) V_i V_i^t \right)^{-1} \left(\sum_{\forall i:\, P_{ui} \neq 0} C_{ui}\, P_{ui}\, V_i \right)$$

$$V_i = \left(U^t U + \lambda C_v I + \sum_{\forall u:\, P_{ui} \neq 0} (C_{ui} - 1) U_u U_u^t \right)^{-1} \left(\sum_{\forall u:\, P_{ui} \neq 0} C_{ui}\, P_{ui}\, U_u \right)$$

Рейтинг предсказывается по следующей формуле.

$$\hat{R}_{ui} = U_u^t V_i$$

После этого, пользователю u рекомендуются объекты, соответствующие номерам максимальных значений рейтингов $\hat{R}_u$, которые предсказаны для этого пользователя.

3.1.3 Метод SVD

Зададим матрицу пользователей-объектов R так же, как в подразделе 3.1.1. В рассматриваемой задаче в качестве степени взаимодействия используется параметр watched_pct – доля просмотра фильма пользователем, а также α и β – настраиваемые параметры.

$$R_{ui} = \begin{cases} \alpha \cdot watched\_pct \,/\, 100 + \beta, \text{если пользователь } u \text{ посмотрел фильм } i \\ 0, \text{иначе} \end{cases}$$

где:

- watched_pct – доля просмотра фильма i пользователем u;
- α и β – настраиваемые параметры.

Применяем к матрице R сингулярное разложение

$$R \approx U' \Sigma V'$$

Построим матрицы U и V представлений пользователей и фильмов следующим образом.

$$U = U' \Sigma^{1/2}$$
$$V = V' \Sigma^{1/2}$$

Далее, аналогично описанному в подразделе 3.1.2, перемножив данные матрицы, вычислим предсказанные рейтинги.

$$\hat{R} = \mathrm{UV}$$

После этого, аналогично описанному в п.2.1.2, пользователю u рекомендуются объекты, соответствующие номерам максимальных значений рейтингов $\hat{R}_u$, которые предсказаны для этого пользователя.

3.1.4 Градиентный бустинг

Суть метода градиентного бустинга над деревьями решений заключается в том, что он строит ансамбль простых моделей, в роли которых выступают деревья решений, последовательно, с целью уменьшить ошибку текущего ансамбля. Градиентный бустинг хорошо подходит для работы с неоднородными данными, с большим количеством категориальных признаков (таких как пол, возрастная группа и т.п.) и сложных зависимостей. Поэтому градиентный бустинг применяется в широком спектре задач, в том числе и в рекомендательных системах.

В исследуемом датасете, помимо данных о взаимодействиях пользователей и фильмов, есть данные о самих пользователях и есть данные о фильмах. Более того, среди них есть большое количество категориальных признаков, таких как пол пользователя, возраст пользователя, флаг «наличие ребёнка» у пользователя, доход пользователя, флаг «контент для взрослых» у фильма. Кроме этих данных, есть ещё текстовые, такие как название фильма, описание фильма, ключевые слова и жанр, которые средствами NLP можно предобработать и привести к виду, подходящему к использованию бустингом.

Датасет, для которого применяется градиентный бустинг в задаче ранжирования, представляет из себя таблицу, каждой строке которой соответствует пара пользователь-фильм, в которой данный пользователь посмотрел данный фильм, а каждому столбцу соответствует признак.

В этой работе используется модель CatBoostRanker из библиотеки CatBoost, с функцией потерь PairLogit. Этот метод подразумевает, что задаётся набор пар объектов и разбивается на группы с целью наиболее правильно отранжировать пары объектов в каждой группе. Алгоритм автоматически

разбивает объекты на пары в зависимости от метки: объект с меткой 1 в последовательности должен стоять раньше объекта с меткой 0.

Функция потерь выглядит следующим образом:

$$\frac{-\sum_{p,n \in Pairs} w_{pn} \left(log(\frac{1}{1+e^{-(a_p - a_n)}}) \right)}{\sum_{p,n \in Pairs} w_{pn}}$$

где:

- a_p – предсказанное значение для объекта p
- a_n – предсказанное значение для объекта n
- w_{pn} – заданный вес пары объектов (p, n) – по умолчанию равен 1.

Данные для этой модели включают матрицу признаков, предсказыавемые метки (0 или 1) и метки групп. Метка группы определяет внутри какой группы ранжируется соответствующая пара пользователь-объект.

Сформированы следующие признаки для каждой пары (u, i), где u – пользователь, i – фильм:

- age _18_24 – 1, если возраст пользователя u от 18 до 24 лет, иначе 0;
- age _25_34 – 1, если возраст пользователя u от 25 до 34 лет, иначе 0;
- age _35_44 – 1, если возраст пользователя u от 35 до 44 лет, иначе 0;
- age _45_54 – 1, если возраст пользователя u от 45 до 54 лет, иначе 0;
- age _55_64 – 1, если возраст пользователя u от 55 до 64 лет, иначе 0;
- age _65_inf – 1, если возраст пользователя u от 65 лет, иначе 0;
- age _M – 1, если пол пользователя u мужской, иначе 0;
- age _F – 1, если пол пользователя u женский, иначе 0;
- ials_dot – скалярное произведение векторов U_u и V_i, полученных с помощью метода iALS;
- ials_cos – косинусное расстояние векторов U_u и V_i, полученных с помощью метода iALS;
- svd_dot – скалярное произведение векторов U_u и V_i, полученных с помощью метода SVD;
- svd_cos – косинусное расстояние векторов U_u и V_i, полученных с помощью метода SVD;
- векторное представление, полученное из названия и списка ключевых слов фильма i, полученное с помощью метода word2vec.

Предсказываемая метка формируется как 1, если пользователь посмотрел более 10% фильма, т.е. если watched_pct > 10, и 0 в противном случае.

Метка группы соответствует параметру user_id, так как метод предназначен для ранжирования объектов под каждого пользователя.

3.2 Применение моделей на данных

Для оценки качества рекомендаций выбраны метрики Precision@10, Recall@10 и MAP@10 со значением k=10, потому что это значение типично для рекомендаций, предлагаемых пользователям в реальных рекомендательных системах. Также это позволит сравнить результаты с результатами других исследований.

На рисунках 3.1-3.3 на графиках приведены значения метрик рекомендаций, полученных методом item-to-item с разными значениями параметров α, β и N. Параметры α и β задают степень важности взаимодействия пользователей и фильмов: α регулирует важность длительности просмотра, β регулирует важность факта просмотра. Параметр N задаёт количество самых похожих объектов, которые используются для предсказания рейтинга объекта.

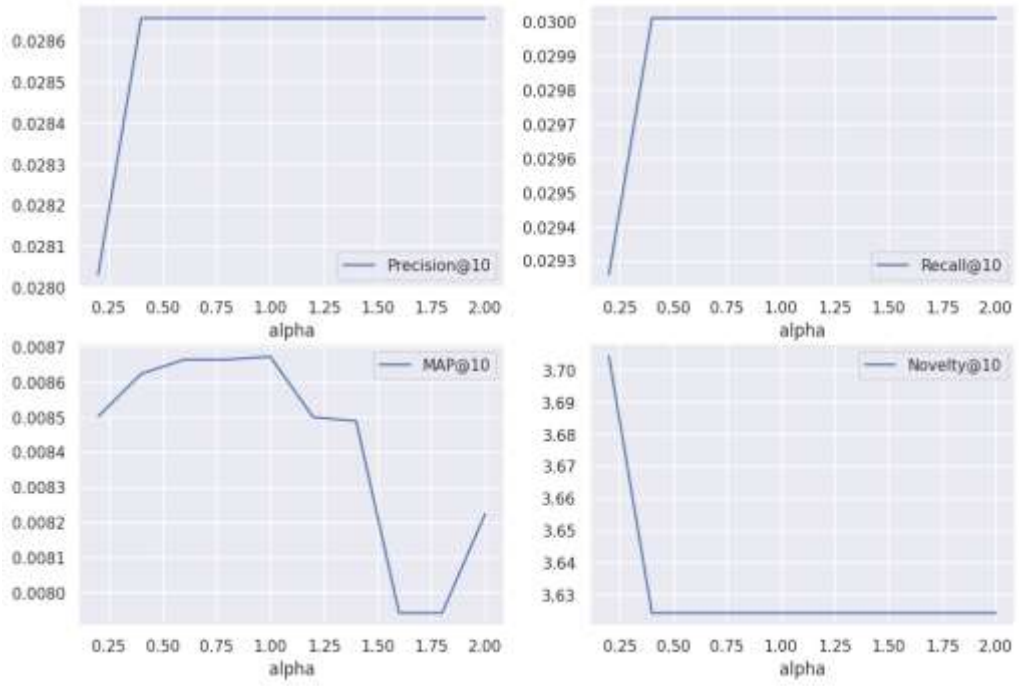

Рисунок 3.1 – Значения метрик, полученных с помощью метода item-to-item, при изменении параметра alpha и фиксированных beta=1 и N=10.

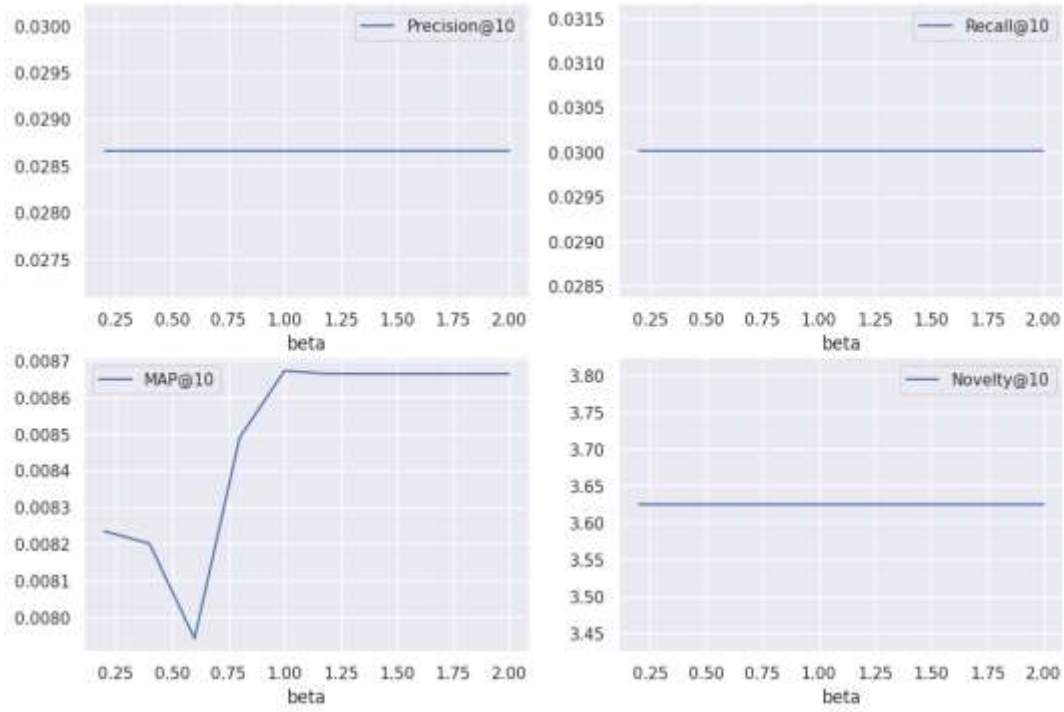

Рисунок 3.2 – Значения метрик, полученных с помощью метода item-to-item, при изменении параметра beta и фиксированных alpha=1 и K=10.

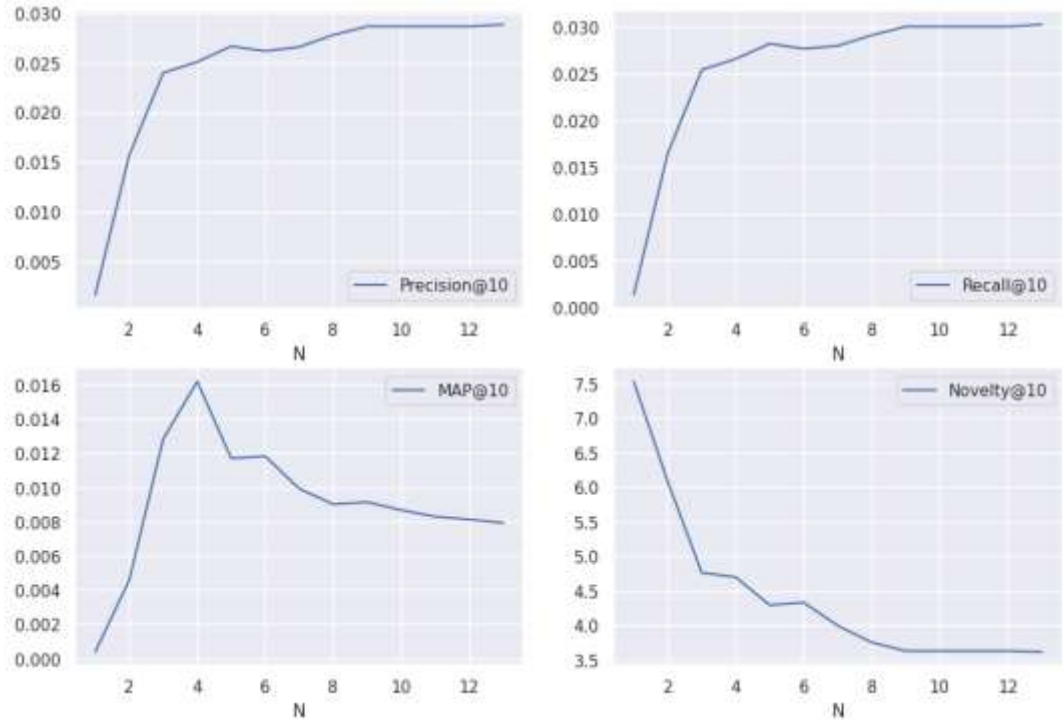

Рисунок 3.3 – Значения метрик, полученные с помощью метода item-to-item, при изменении параметра K и фиксированных beta=1 и alpha=1.

Наилучшие результаты получены при alpha=1, beta=1 и N=4. Значения метрик приведены в таблице 3.1.

Таблица 3.1 – значения метрик, полученные с помощью метода item-to-item при alpha=1, beta=1 и N=4

Precision@10	0.025075
Recall@10	0.026558
MAP@10	0.016188
Novelty	4.692383

На рисунках 3.4-3.5 на графиках приведены значения метрик рекомендаций, полученных методом iALS с разными значениями параметров f и λ. Параметр f определяет размер скрытых представлений пользователей и объектов, параметр λ задаёт степень регуляризации модели с целью предотвратить переобучение.

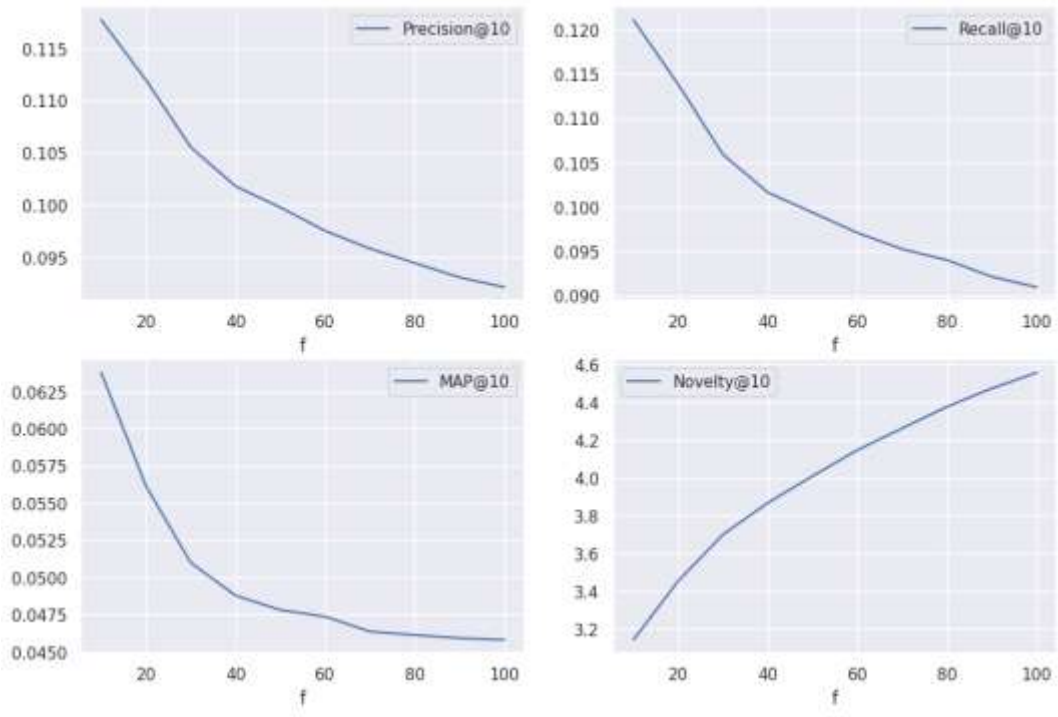

Рисунок 3.4 – Значения метрик рекомендаций, полученных с помощью iALS, при изменении параметра f и фиксированном lambda=0.006.

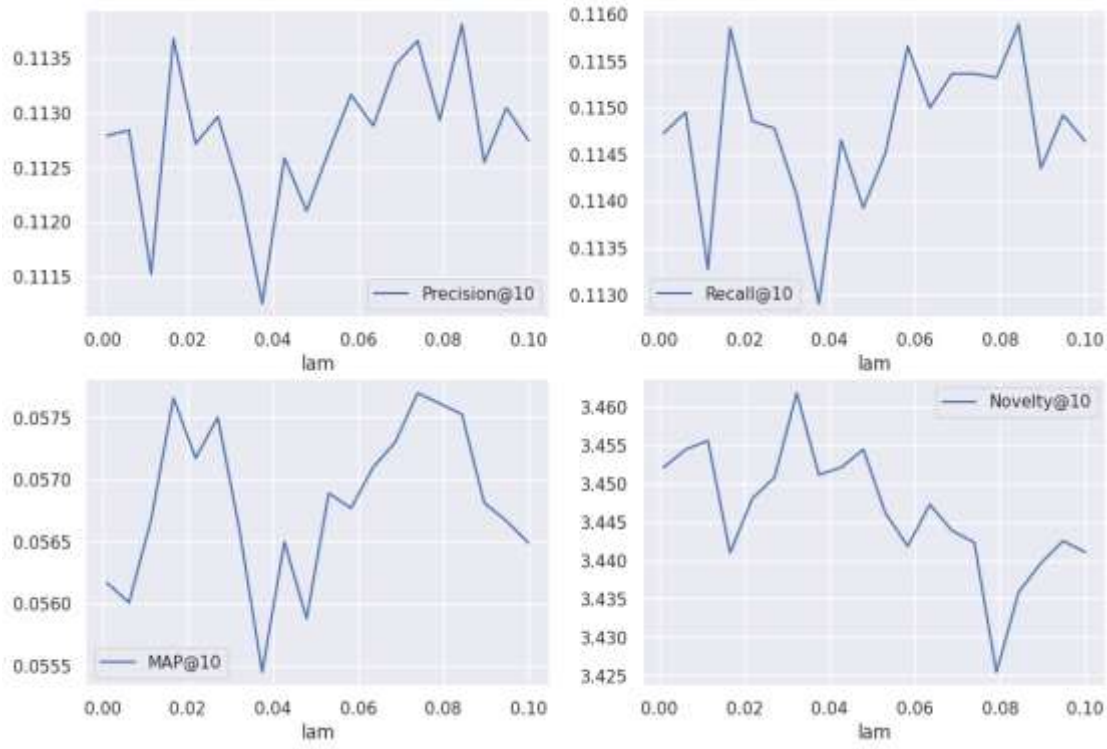

Рисунок 3.5 – Значения метрик рекомендаций, полученных с помощью iALS, при изменении параметра lambda и фиксированном f=20.

Наилучшие значения получены при lambda=0.016, f=20. Значения метрик приведены в таблице 3.2.

Таблица 3.2 – значения метрик, полученные с помощью iALS при lambda=0.016, f=20

Precision@10	0.113683
Recall@10	0.115854
MAP@10	0.057658
Novelty	3.440978

На графиках на рисунке 3.6 приведены значения метрик рекомендаций, полученных методом SVD с разными значениями параметра f. Параметр f определяет размер скрытых представлений пользователей и объектов, аналогично предыдущему методу.

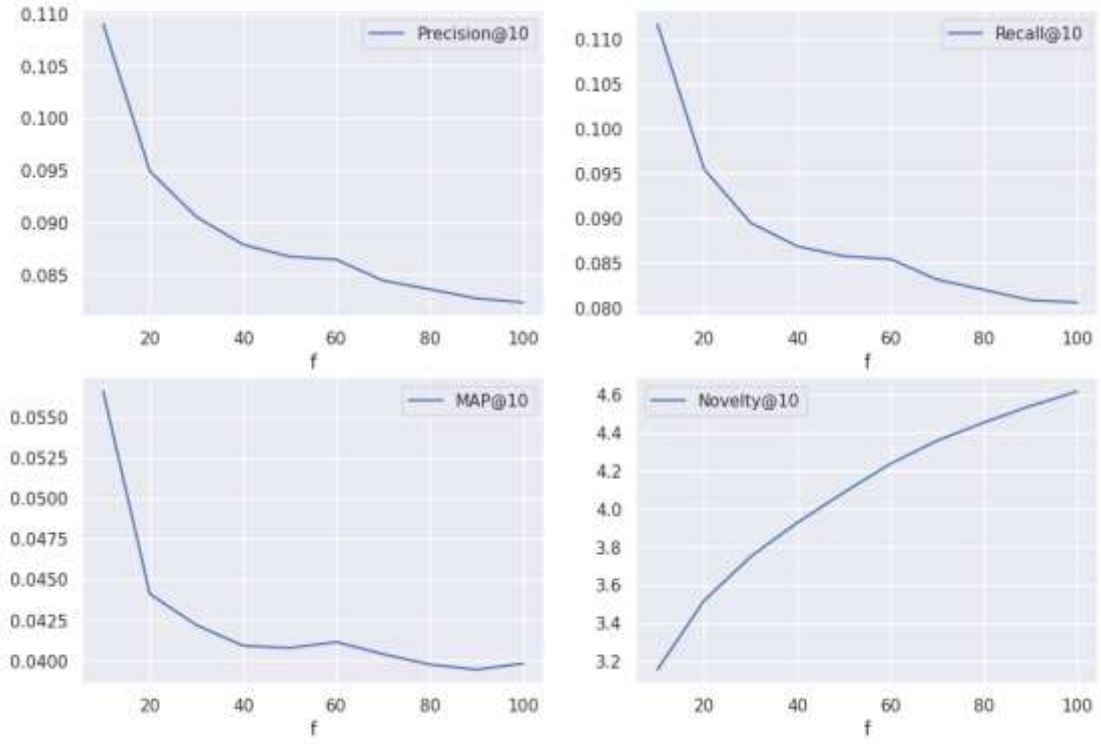

Рисунок 3.6 – Значения метрик рекомендаций, полученных с помощью SVD, при изменении параметра f.

Значения метрик при f=20 приведены в таблице 3.3.

Таблица 3.3 – значения метрик, полученные с помощью SVD при f=20

Precision@10	0.094919
Recall@10	0.095482
MAP@10	0.044104
Novelty	3.517412

Модель бустинга использовалась в качестве второго этапа построения рекомендаций. На первом этапе с помощью модели iALS для каждого пользователя рекомендуется 100 фильмов. Затем для каждой пары (пользователь + рекомендованный ему фильм) с помощью модели бустинга предсказывается скор. В итоговый список рекомендаций (для каждого пользователя) попадает 10 фильмов с наивысшим скором. Полученные результаты приведены в таблице 3.4.

Таблица 3.4 – значения метрик, полученные с помощью модели бустинга

Precision@10	0.094919
Recall@10	0.095482
MAP@10	0.044104
Novelty	3.517412

Итоговые метрики полученных моделей приведены в таблице 3.5.

Таблица 3.5 – значения метрик, полученные с помощью различных моделей

	Precision@10	Recall@10	MAP@10	Novelty
item-to-item	0.025075	0.026558	0.016188	4.692383
iALS	0.113683	0.115854	0.057658	3.440978
SVD	0.094919	0.095482	0.044104	3.517412
CatBoostRanker	0.117846	0.117691	0.060384	3.739019

Как видно из таблицы 3.5, с помощью градиентного бустинга удалось получить наилучшие результаты по сравнению с остальными моделями, хотя и не сильно лучше, чем у модели iALS.

3.3 Примеры

В данном разделе представлены результаты анализа рекомендательных систем, основанных на моделях item-to-item и iALS, а также данные о предпочтениях пользователя в контексте фильмов. Таблица 3.6 содержит список фильмов, предпочтения в которых являются эталоном для сравнения с рекомендациями моделей. В таблице 3.7 приведены рекомендации, полученные с помощью модели item-to-item. В таблице 3.8 приведены рекомендации полученные с помощью модели iALS. В таблице 3.9 приведены рекомендации полученные с помощью модели CatBoostRanker.

Таблица 3.6 – Список любимых фильмов пользователя для иллюстрации примера рекомендаций

Зверополис	мультфильм, детективы, комедии
Зелёная книга	драмы, биография, комедии
Kingsman: Секретная служба	боевики, криминал, комедии
Один дома	семейное, комедии
Пираты карибского моря: Проклятие чёрной жемчу...	боевики, фэнтези, приключения
Волк с Уолл-стрит	драмы, биография, комедии
Холодное сердце	фэнтези, мультфильм, музыкальные
Пираты карибского моря: На краю света	боевики, фэнтези, приключения
Пираты карибского моря: На странных берегах	боевики, фэнтези, приключения

Таблица 3.7 – Фильмы, рекомендованные с использованием модели item-to-item

Хрустальный	триллеры, детективы
Гнев человеческий	боевики, триллеры

Девятаев	драмы, военные, приключения
Мстители: Финал	боевики, драмы, фантастика
Студентка по вызову	драмы, мелодрамы
Тайна Коко	мультфильм, фэнтези, приключения
Суперсемейка 2	фантастика, мультфильм, приключения
Корпорация монстров	мультфильм, фэнтези, приключения, комедии
Тор: Рагнарёк	приключения, фантастика, боевики, фэнтези, ком...
История игрушек 4	мультфильм, фэнтези, комедии
Мстители: Эра Альтрона	боевики, фантастика, приключения
Храбрая сердцем	мультфильм, фэнтези, приключения, комедии
Первый мститель: Противостояние	боевики, фантастика, приключения
Русалочка	мюзиклы, мультфильм, фэнтези, мелодрамы
Три богатыря и Наследница престола	мультфильм, приключения
Три богатыря. Принцесса Египта	мультфильм, фэнтези, приключения, комедии
Три богатыря и Морской царь	мультфильм, приключения, комедии
Три богатыря и Шамаханская царица	мультфильм, фэнтези, приключения, комедии
Снежная королева: зазеркалье	мультфильм, фэнтези, приключения
Легендарное ограбление	историческое, криминал, драмы, триллеры, боевики

Таблица 3.8 – Фильмы, рекомендованные с использованием модели iALS

Клиника счастья	драмы, мелодрамы
Гнев человеческий	боевики, триллеры
Секреты семейной жизни	комедии
Прабабушка легкого поведения	комедии
Подслушано	драмы, триллеры
Афера	комедии
Немцы	боевики, драмы
Восемь сотен	боевики, драмы, военные
Крёстная мама	драмы, комедии
Соседи сверху	комедии
Дублёрша	комедии
Три дня до весны	военные, детективы
Запретная любовь	драмы, военные, мелодрамы
9 месяцев строгого режима	криминал, детективы, комедии
Американский психопат 2: Стопроцентная американка	ужасы, триллеры

Таблица 3.9 – Фильмы, рекомендованные с использованием модели CatBoostRanker

Девятаев	драмы, военные, приключения
Веном	популярное, фантастика, триллеры, боевики
Аладдин	фэнтези, приключения, мелодрамы
Альфа	драмы, популярное, семейное, приключения
Один дома	семейное, комедии
Послезавтра	драмы, фантастика, триллеры, приключения
Пираты Карибского Моря: Сундук Мертвеца	боевики, фэнтези, приключения
Пираты Карибского моря: Мертвецы не ра...	боевики, фэнтези, приключения
Kingsman: Золотое кольцо	криминал, триллеры, боевики, комедии
Принц Персии: Пески времени	боевики, фэнтези, приключения

В таблицах 3.10-3.11 приведены похожие объекты, полученные с помощью метода item-to-item.

Таблица 3.10 – Фильмы, похожие на «Зверополис», полученные с помощью модели item-to-item

Зверополис	**приключения, мультфильм, детективы, комедии**
Гнев человеческий	боевики, триллеры
Девятаев	драмы, военные, приключения
Прабабушка легкого поведения	комедии
100% волк	мультфильм, приключения, семейное, фэнтези, ко...
Монстры на каникулах 3	мультфильм, фэнтези, приключения, комедии
Тайна Коко	мультфильм, фэнтези, приключения
Холодное сердце II	фэнтези, мультфильм, музыкальные
Моана	мультфильм, фэнтези, мюзиклы

Таблица 3.11 – Фильмы, похожие на «Пираты карибского моря: На странных берегах», полученные с помощью модели item-to-item

Пираты карибского моря: На странных берегах	**боевики, фэнтези, приключения**
Гнев человеческий	боевики, триллеры
Девятаев	драмы, военные, приключения
Прабабушка легкого поведения	комедии
Мстители: Финал	боевики, драмы, фантастика
Мстители: Война бесконечности	боевики, фантастика, приключения
Пираты карибского моря: Проклятие чёрной жем...	боевики, фэнтези, приключения
Пираты Карибского Моря: Сундук Мертвеца	боевики, фэнтези, приключения

Пираты Карибского моря: Мертвецы не рассказы...	боевики, фэнтези, приключения
Пираты карибского моря: На краю света	боевики, фэнтези, приключения

3.4 Сравнительный анализ методов

Метод item-to-item основан на вычислении сходства между объектами. Его преимущество состоит в простоте реализации, вычислительной эффективности и интерпретируемости. Обучение этого метода занимает меньше времени по сравнению с другими методами. Этот метод помимо построения рекомендаций позволяет строить списки похожих объектов используя значения из матрицы похожестей.

Метод iALS основан на матричном разложении и неявной обратной связи от пользователей. Метод iALS более сложный в сравнении с item-to-item, содержит большее количество настраиваемых параметров, итеративный, обучение занимает большое количество времени. Однако, в сравнении с другими методами, обладает лучшей точностью. Это в том числе связано с тем, что этот метод лучше приспособлен для работы с неявной обратной связью.

Применение метода SVD для построения рекомендаций также показало себя хорошо, но недостаточно, по сравнению с методом iALS. Этот метод, также как и iALS основан на матричном разложении. Обучение занимает меньше времени, чем обучение iALS, но SVD уступает по качеству рекомендаций. Это может быть связано с различными факторами, в том числе с плохой приспособленностью этого метода к работе с неявной обратной связью.

Модель градиентного бустинга, которая использовалась в качестве второго этапа построения рекомендаций, на первом этапе с помощью модели iALS для каждого пользователя рекомендуется 100 фильмов, показала наилучшие результаты по качеству рекомендаций по сравнению с остальными моделями. Однако обучение этой модели заняло значительно более время, чем для остальных моделей, потому что помимо самого бустинга, нужно также обучить модель альтернативных наименьших квадратов. Построение рекомендаций также занимает значительно больше времени, так как рекомендации формируются в два этапа: сначала для каждого пользователя рекомендовался большой набор фильмов, а затем с помощью модели бустинга из них отбирались топ-10 самых релевантных.

Стоит отметить, что хотя модели iALS и SVD получили более высокие значения метрик качества, это во много может быть связано с тем, что они большинству пользователей рекомендовали самые популярные объекты, об этом

можно судить на основании более низких значений метрики новизны, что в данной ситуации и оказалось хорошей стратегией, но, вообще говоря, может быть не всегда так.

3.5 Основные результаты и выводы

В данной главе реализованы метод коллаборативной фильтрации на основе объектов, метод альтернативных наименьших квадратов, сингулярное разложение и градиентный бустинг. Методы применены на данных, построены рекомендации, посчитаны значения метрик качества рекомендаций, выбраны параметры при которых были получены наилучшие значения метрик. Приведены примеры рекомендаций. Проведён сравнительный анализ методов.

ЗАКЛЮЧЕНИЕ

В данной работе были изучены различные методы и подходы к построению рекомендательных систем. Изучены и проанализированы предоставленные данные о фильмах и об историях просмотров пользователей онлайн кинотеатра Kion. Изучены методы коллаборативной фильтрации на основе объектов, неявных альтернативных наименьших квадратов, сингулярного разложения, градиентного бустинга. Данные методы были применены на предоставленных данных и были вычислены значения метрик качества рекомендаций, полученных данными методами. Проведено сравнение результатов и были выбраны наилучшие модели. С помощью выбранных моделей были построены списки рекомендаций. Была определена наилучшая комбинация методов, которая решает задачу построения рекомендаций фильмов на основе пользовательских просмотров.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Hu, Y., Koren, Y., & Volinsky, C. (2008). "Collaborative Filtering for Implicit Feedback Datasets." AT&T Labs – Research, Florham Park, NJ 07932 (Yifan Hu, Chris Volinsky), Yahoo! Research, Haifa 31905, Israel (Yehuda Koren).
2. Иванов, П. И., & Смирнова, А. А. (2019). "Рекомендательные системы: принципы, технологии, алгоритмы." Москва: БХВ-Петербург.
3. Koren, Y. (2008). Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model. AT&T Labs – Research, 180 Park Ave, Florham Park, NJ 07932.
4. Ge, M., Delgado-Battenfeld, C., & Jannach, D. (2010, September). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In Proceedings of the fourth ACM conference on Recommender systems.
5. Murakami, T., Mori, K., & Orihara, R. (2008). Metrics for evaluating the serendipity of recommendation lists. In New Frontiers in Artificial Intelligence (Vol. 4914). Springer Berlin Heidelberg.
6. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.
7. Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295).
8. Cremonesi, P., Koren, Y., & Turrin, R. (2010, September). Performance of recommender algorithms on top-n recommendation tasks. In Proceedings of the fourth ACM conference on Recommender systems (pp. 39-46).
9. Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In The Adaptive Web (pp. 325-341). Springer, Berlin, Heidelberg.
10. Lops, P., Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Recommender Systems Handbook (pp. 73-105). Springer, Boston, MA.
11. Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331-370.
12. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749.
13. Non-Negative Matrix Factorization for Recommender Systems Utilizing Implicit Feedback from Female Daily: A Study by Hani Nurrahmi, Agung Toto Wibowo, and Selly Meliana from the School of Computing, Telkom University.