МИНОБРНАУКИ РОССИИ
ФГБОУ ВО «СГУ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

КОЛЛАБОРАТИВНАЯ ФИЛЬТРАЦИЯ В РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМАХ

РЕФЕРАТ

студентки 5 курса 531 группы
направления 100501 — Компьютерная безопасность
факультета КНиИТ
Ивановой Ксении Всладиславовны

Проверил
Доцент _____ И. И. Слеповичев

Саратов 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Одна из интересных тем современного мира, которая стоит на стыке информационных технологий и маркетинга, не дающая уснуть ночью, подкидывая нам интересные видео, и помогающая тратить наши деньги, предлагая товары, которые нам «могут понравиться», тема рекомендательных систем.

Сейчас, в период информационной зависимости, вопрос «что надеть», смещается более актуальным «что посмотреть», а продвижение товара преобразуется из телевизионно-зомбирующей рекламы совершенно ненужных вам вещей в предложения купить товары с оценкой 5 звезд из 5. Умные алгоритмы рекомендаций товаров на основе каких-то действий или признаков покупателя помогают как бизнесу, так и потребителям. Одним из таких методов является коллаборативная фильтрация, о которой и пойдет речь.

1 Метод коллаборативная фильтрации

К коллаборативной (совместной) фильтрации относятся те методы и алгоритмы, которые основываются на данных о предыдущих сеансах работы пользователей с этой же системой.

Общая черта всех методов коллаборативной фильтрации это то, что основой создания рекомендаций является пересечение оценочных мер популярности того или иного объекта. Мерой оценивания объекта можно выбрать не только данные, которые предоставил нам сам пользователь, используя функционал сервиса: отметка звездочкой или оценка объекту, но также можно учитывать данные, которые пользователь предоставляет неосознанно: количество просмотров, время посещения, количество переходов, потраченная сумма.[1]

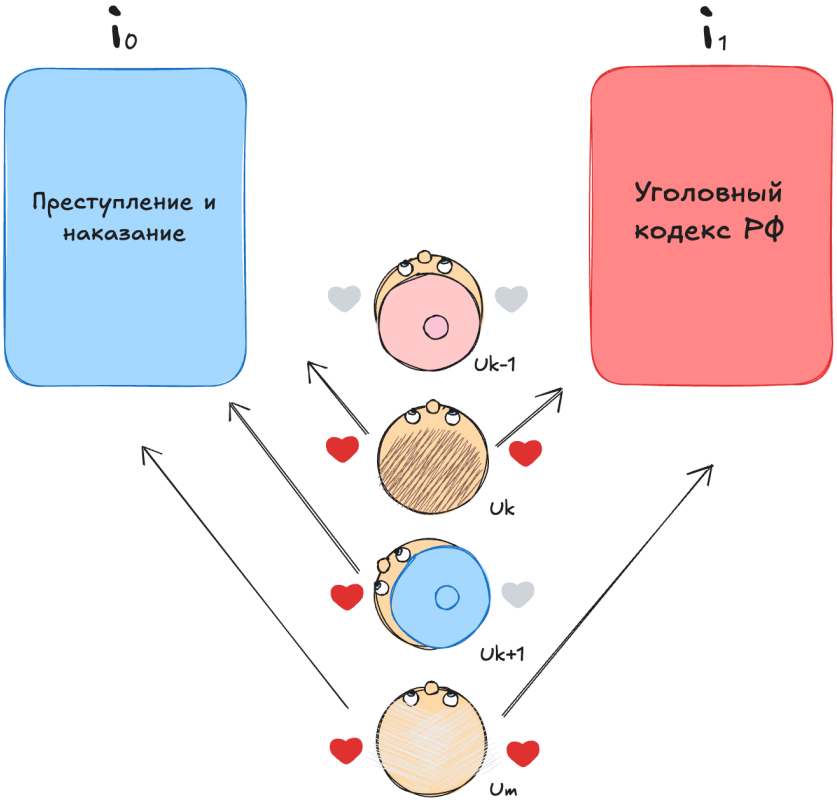

Рисунок 1

Посмотрим пример тривиальной рекомендации для рисунка 1. Формируется матрица предпочтений (user-item matrix), которая отражает взаимодействие пользователей с различными предметами, в нашем случае она отражает понравилась ли пользователю книга. Видим, что пользователем и u_k, u_{k+1}, u_m, понравились одинаковые книги, таких пользователей будем называть «окрестностями» или «коллаборацией», обозначим за $U(i_0)$. Предполагаем, что книга

i_0 похожа на i_1, потому что она понравилась похожим людям, множество таких книг образуется формулой (1), поэтому рекомендуем пользователю u_{k+1} прочитать книгу i_1. Интересный факт, что данный принцип «клиенты купившие предмет А, также купили предмет В», для потребительских манипуляций первым придумали использовать у себя нв страницах Amazon.com, слоган используется до сих пор, хотя алгоритмы подбора рекомендации используют другие принципы.

$$I(i_0) = \{i \in I | sim(i_0, i_1) = |\frac{U(i_0) \cap U(i_1)}{U(i_0) \cup U(i_1)}|\}, \quad (1)$$

где $sim(i_0, i_1)$ - мера оценивания близости двух книг i_0 и i_1.

Несложно оценить, что метод имеет ряд недостатков:

1. Рекомендации предлагают самые популярные товары;
2. Не учитываются интересы конкретного пользователя u_i;
3. Новый товар никому не рекомендуется;
4. Нужно хранить матрицу предпочтений.

Рассмотрим два больших вида подходов в коллаборативной фильтрации, а также их выигрышные и наоборот стороны.[6]

1.1 Корреляционные модели

Корреляционные модели или на английском языке «Memory-Based Collaborative Filtering». В моделях этого вида для построения прогнозов используется выявление разного рода корреляций между пользователями на основе матрицы соответствия оценок пользователей и объектов.

Отличающие особенности:

— Хранение всей исходной матрицы данных;
— Сходство клиентов —- это корреляция строк;
— Сходство объектов —- это корреляция столбцов.

Описанный в предыдущей главе пример также относится к коллаборативной фильтрации, только в самом наивном ее представлении, в основном корреляционные модели делятся на два подхода — «От клиента» и «От предмета».

1.1.1 От клиента

В данном подходе немного меняется формулировка фразы «клиенты купившие предмет A, также купили предмет B» на «клиенты похожие на Вас, также купили предмет B». Если рассматривать матрицу отношений то «От клиента» означает, что корреляцию будем искать между строками матрицы, т.е пользователями.

Для простоты представим, что пользователь оценивает понравилась музыкальная группа или нет, таким образом в матрице, если группа понравилась — на пересечении пользователя и группы ставится красное сердце, если группа известна, но особых чувств не вызвала — пустое сердце, если данный объект еще не попадался пользователю — пустая ячейка. Матрица R показана на рисунке 2.

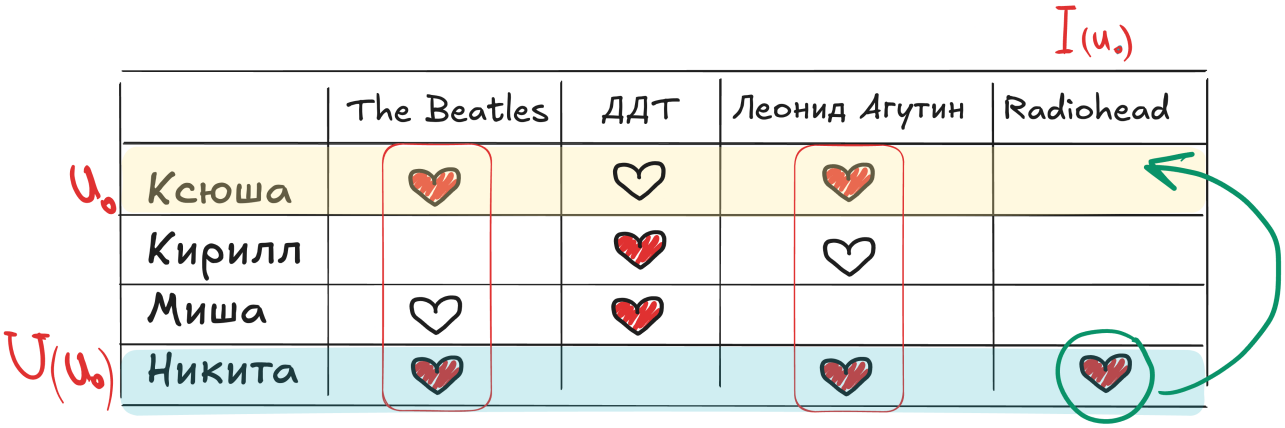

Рисунок 2

Коллаборация пользователей, близких по интересам зададим формулой (2).

$$U(u_0) = \{u \in U | sim(u_0, u) > \alpha\}, \tag{2}$$

где $sim(i, i_0)$ - мера оценивания близости двух пользователей u и u_0.

А множество схожих объектов, которые можно отсортировать по убыванию некоторой функции $B(i)$, а далее предложить пользователю можно выразить формулой (3).

$$I(i_0) = \{i \in I | B(i) = |\frac{U(u_0) \cap U(i)}{U(u_0) \cup U(i)}|\}, \tag{3}$$

где $U(i) = \{u \in U | r_{u,i} \neq \varnothing\}$.

Рассмотрим алгоритм, который предсказывает оценку для объектов, еще неоцененных данным пользователем:

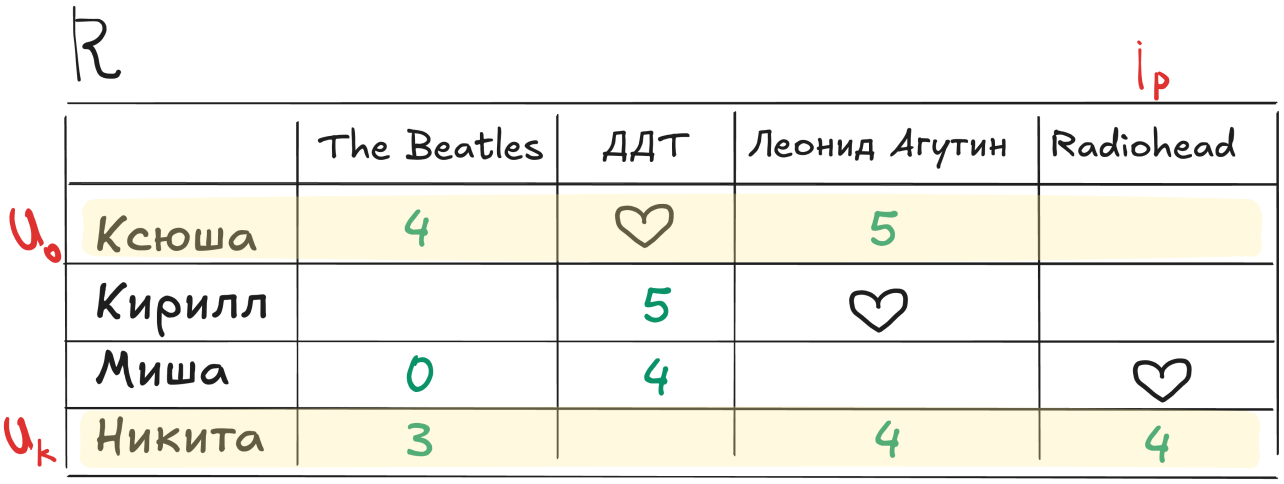

Рисунок 3

Шаг1. Немного изменим предыдущий рисунок, пусть вместо «сердечек», пользователь выставляет числовые оценки, рассмотрим рисунок 3. Вычислим близость строк в матрице R, будем учитывать только тех пользователей, которые оценивали объекты. Есть много видов метрик, которые можно использовать, проще всего будет взять косинусное расстояние формула (4).

$$sim(u_0, u_k) = \frac{\sum_{i=1}^{m} r_{u_0,i} \cdot r_{u_k,i}}{\sqrt{\sum_{i=1}^{m} r_{u_0,i}} \cdot \sqrt{\sum_{i=1}^{m} r_{u_k,i}}}, \tag{4}$$

где $sim(u_0, u_k)$ близость пользователей u_0 и u_k, $r_{u_0,i}, r_{u_k,i}$ – значения матрицы R.

Немного про метрики, пусть у нас есть три вектора оценок от пользователей $a = (5, 5, 2), b = (2, 2, 0), c = (0, 0, 2)$. Если мы возьмем в качестве метрики евклидово расстояние, то наиболее близкими будут вектора b и c т.к их нормы ближе по значению по сравнению с a, в отличие от косинусной метрики, где b и c вообще не имеют общих оценок от пользователей, поэтому ближе будут a и b. Поэтому при выборе метрики нужно учитывать специфику задачи.

Шаг2. После того как мы посчитали близости между пользователями, все пользователи сортируются по убыванию меры близости, нужно выбрать множество, назовем его K, похожих на u_0. Можно выбрать и всех пользователей, но тогда непохожие пользователи будут сильно влиять на точность предска-

зания оценки, также это будет служит дополнительным объемом данных для вычислений на 3 шаге.

Лучше выбрать какое-то пороговое значение, которое будет приблизительно достоверно оценивать «похожесть», обычно выбирается целая константа.

Шаг3. Вычисляем приблизительную оценку, которую пользователь u_0 поставил бы объекту i_p, формула (5).

$$p(u_0, i_p) = \overline{r}_{u_0} + \frac{\sum_{u \in K}(r_{u_0,i_p} - \overline{r}_{u_0}) \times sim(u_0, u_k)}{\sum_{u \in K} |sim(u_0, u_k)|}, \quad (5)$$

где $p(u_0, i_p)$ – предсказываемая оценка, правое слагаемое – среднее отклонение оценки других пользователей из K для объекта i_p от их реальной оценки.

Так как для каждого пользователя понятие «рейтинг» очень субъективное, кто-то поставит хорошему фильму оценку «5», а кто-то оценку «4», ибо есть идеал и он недостижим, чтобы сгладить такие различия между пользователями, в формуле (5) сначала вычитается, а потом добавляется средняя оценка пользователя r_{u_0}. Сама же формула носит название «непараметрическая регрессия Надарайя-Ватсона» или «формула ядерного сглаживания».

Часть недостатков удалось устранить, но все также остаются следующие:

1. ~~Рекомендации предлагают самые популярные товары~~;
2. ~~Не учитываются интересы конкретного пользователя u_i~~;
3. Новый товар никому не рекомендуется;
4. Нужно хранить матрицу предпочтений;
5. Нечего рекомендовать нетипичным пользователям.

1.1.2 От предмета

Данный метод характеризуется фразой: «вместе с объектами которые покупал пользователь U_o, также часто покупают...». Используя этот метод, при рекомендации мы учитываем некоторые действия которые уже совершил пользователь. В отличие от предыдущего метода корреляция ищется между столбцами матрицы R, т.е рассматриваются музыкальные группы похожие на те, что пользователе уже оценил, рисунок 4.

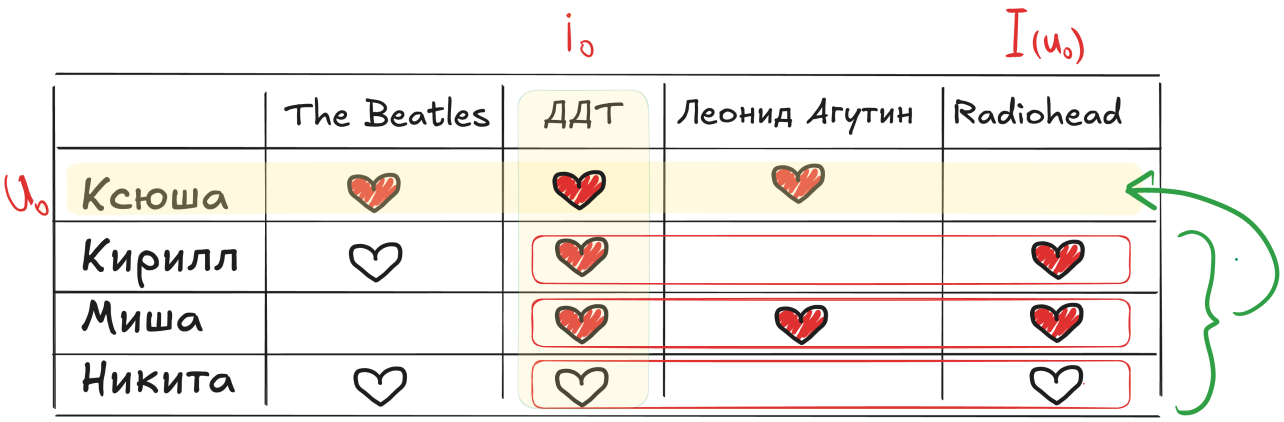

Рисунок 4 – Таблица соответствия пользователей их оценкам для музыкальных групп.

Множество похожих объектов можно выразить формулой (6).

$$I(u_0) = \{i \in I | \exists i_0 : r_{u_0,i_0} \neq \varnothing \wedge B(i) = sim(i, i_o) > \alpha\}, \tag{6}$$

где $sim(i, i_0)$ - мера оценивания близости двух книг i и i_0.
Если рассматривать алгоритм работы, он очень похож на алгоритм «От клиента».
Шаг1. Для каждого объекта i_p вычисляется на сколько он похож на объект i_k для которого предсказывается оценка, формула (7).

$$sim(i_k, i_p) = \frac{\sum_{i=1}^{n} r_{i_k,i} \cdot r_{i_p,i}}{\sqrt{\sum_{i=1}^{m} r_{i_k,i}} \cdot \sqrt{\sum_{i=1}^{n} r_{i_p,i}}}, \tag{7}$$

где $sim(i_k, i_p)$ близость объектов i_k и i_p, $r_{i_k,i}, r_{i_p,i}$ — значения матрицы R.
Шаг2. Выбирается множество объектов наиболее похожих на i_k.
Шаг3. Предсказывается оценка для выбранных на шаге 2 объектов, используя формулу (8).

$$p(u_0, i_k) = \frac{\sum_{i_k \in K} r_{u_0,i_p} \times sim(i_k, i_p)}{\sum_{i_k \in K} |sim(i_k, i_p)|}, \tag{8}$$

где $p(u_0, i_p)$ — предсказываемая оценка, правое слагаемое — среднее отклонение оценки других пользователей из K для объекта i_p от их реальной оценки.

Уже интереснее, но все еще есть некоторые минусы:

1. Рекомендации предлагают самые популярные товары (нет коллаборативности);
2. ~~Косвенно учитываются интересы конкретного пользователя~~;
3. Новый товар никому не рекомендуется;

4. Нужно хранить матрицу предпочтений;
5. ~~Нечего рекомендовать нетипичным пользователям~~.

1.2 Латентные модели

Латентные модели или на английском «Latent Models for Collaborative Filtering». Для каждого пользователя или объекта будем рассматривать не вектор его взаимодействий с соответствующими элементами рекомендательной системы, а некий профиль —— вектор, который описывает объект или пользователя в каком-то пространстве, размерность которого, как правило, много меньше размерностей исходной матрицы R.

Также можем представить это в виде некоторого пространства рисунок 5, где пользователи и объекты размещены в плоскости тегов, исходя из тегов в их профилях, тогда можем видеть, что некоторые субъекты ближе к каким-то объектам и это можно использовать для создания рекомендаций.

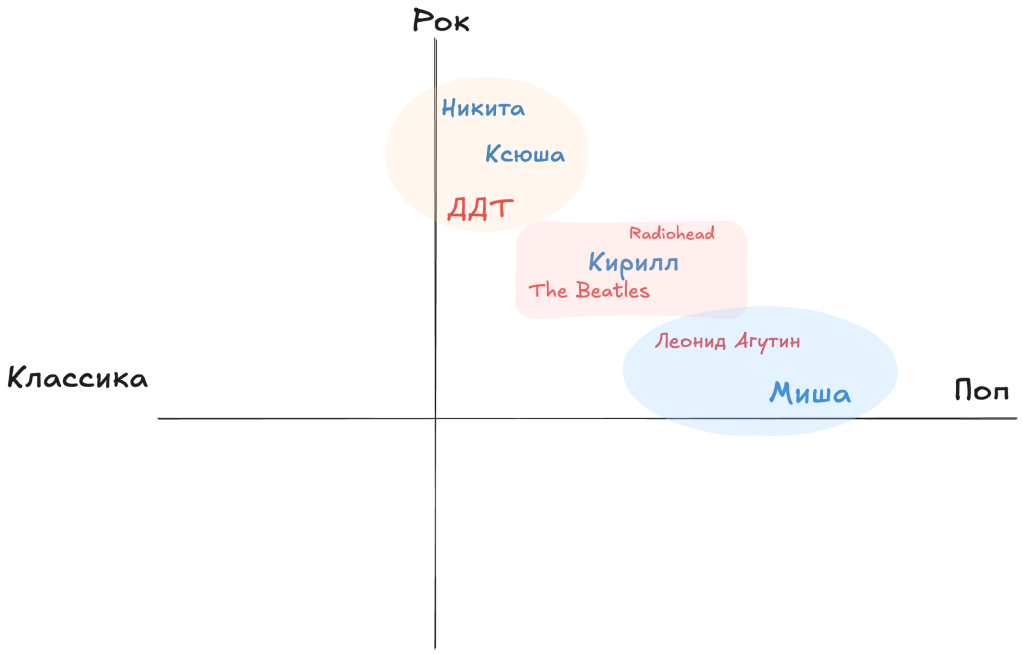

Рисунок 5

Некоторые особенности:

— Профиль —— это вектор скрытых характеристик;
— Хранение профилей вместо хранения матрицы данных;
— Сходство клиентов и объектов — это сходство их профилей.

Рассмотрим некоторые типы идей латентных моделей.

1.2.1 Ко-кластеризация

Идея в том, что нам нужно разделить множество клиентов и объектов по кластерам и найти соответствия для кластеров объектов и пользователей. Бывает жесткая и мягкая кластеризации. Жесткая — приписываем пользователям конкретные кластеры, со степенью принадлежности либо 0 либо 1, например «Никита любит рок». В мягкой кластеризации степень принадлежности величина вещественная, может быть в промежутке значений от 0 до 1, например «Никита относится к року с мерой 0.8»

В качестве примера, можем рассмотреть алгоритм под названием «Латентное размещение Дирихле (LDA)» Латентное размещение Дирихле (LDA) — это вероятностная модель, разработанная для моделирования тем в коллекциях документов. Она основана на предположении, что каждый документ может быть представлен как смесь различных тем, а каждая тема связана с распределением слов.

Алгоритм LDA можно разбить на несколько шагов:

Шаг1. Для применения LDA необходимо заранее определить количество тем, которые вы хотите выделить. Это может быть достаточно сложной задачей, и выбор зависит от конкретных характеристик данных, также нужно выбрать параметры модели.

Шаг2. Подготовка данных. Документы преобразуются в числовое представление, например, с помощью TF-IDF матриц (показывает, сколько раз каждое слово встречается в каждом документе).

Шаг3. Инициализируются распределения тем для документов и распределения слов для тем.

Шаг4. Происходит обучение, итеративно обрабатывается тексты, чтобы определить, какие темы присутствуют в документах и какие слова связаны с каждой темой.

— Для каждого слова в каждом документе вычисляется вероятность принадлежности к каждой теме, используя текущие распределения тем и слов;
— На основе вероятностей слов в темах и вероятностей тем в документах пересчитываются распределения тем и слов;

Алгоритм старается максимизировать вероятность появления слов в каждой теме и вероятность наличия тем в каждом документе. По окончании итераций можно получить распределения тем для каждого документа и распределе-

ния слов для каждой темы.

Некоторые недостатки:

— Определение оптимального количества тем может быть сложной задачей и требует экспериментов;

— Результаты могут сильно зависеть от исходных параметров и инициализации;

— Необходима тщательная предобработка данных, включая удаление стоп-слов и другие шаги.

1.2.2 Матричная факторизация

Была такая известная история про приз от Netflix. В 2006 году компания пообещала выплатить 1 млн разработчикам, которые смогут улучшить эффективность её алгоритма рекомендации фильмов минимум на 10%. В конкурсе, длившемся почти три года, участвовали десятки тысяч команд со всего мира. В итоге выиграла BellKor Pragmatic Chaos, в которую вошли учёные из американской ATT Labs и австрийской Commendo Research Consulting. Она даже немного перевыполнила план —— эффективность алгоритма улучшилась на $10,05\%$. Решение победившей команды использовало матричную факторизацию, т.е поиск матриц, на которые можно разложить матрицу оценок R.

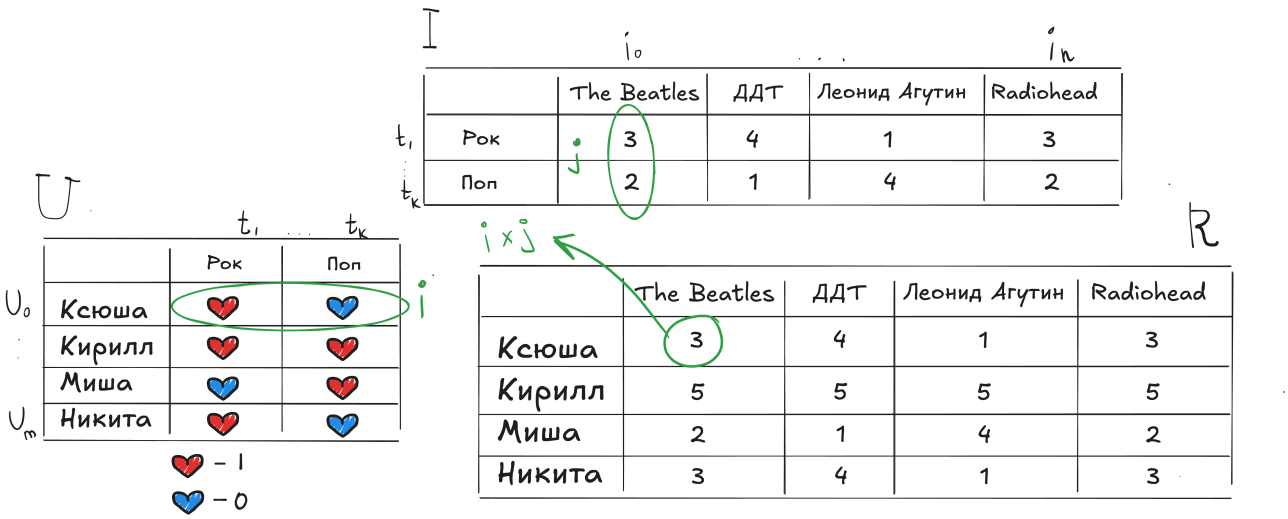

Рисунок 6

Пусть у нас есть наша матрица оценок, оценки складываются из скалярного умножения

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Смоленчук Татьяна Владимировна Метод коллаборативной фильтрации для рекомендательных сервисов // Вестник науки и образования. 2019. №22-1 (76). URL: https://cyberleninka.ru/article/n/metod-kollaborativnoy-filtratsii-dlya-rekomendatelnyh-servisov (дата обращения: 18.12.2024).

2 Ларионов В. С., Дунин И. В. ОБЗОР МЕТОДОВ КОЛЛАБОРАТИВНОЙ ФИЛЬТРАЦИИ // Форум молодых ученых. 2017. №5 (9). URL: https://cyberleninka.ru/article/n/obzor-metodov-kollaborativnoy-filtratsii (дата обращения: 18.12.2024).

3 Гомзин А. Г., Коршунов А. В. Системы рекомендаций: обзор современных подходов // Труды ИСП РАН. 2012. №. URL: https://cyberleninka.ru/article/n/sistemy-rekomendatsiy-obzor-sovremennyh-podhodov (дата обращения: 18.12.2024).

4 https://cts.etu.ru/assets/files/2021/cts21/papers/287-290.pdf

5 https://education.yandex.ru/handbook/ml/article/intro-recsys

6 http://www.machinelearning.ru/wiki/images/archive/9/95/20140413184117!Voron-ML-CF.pdf