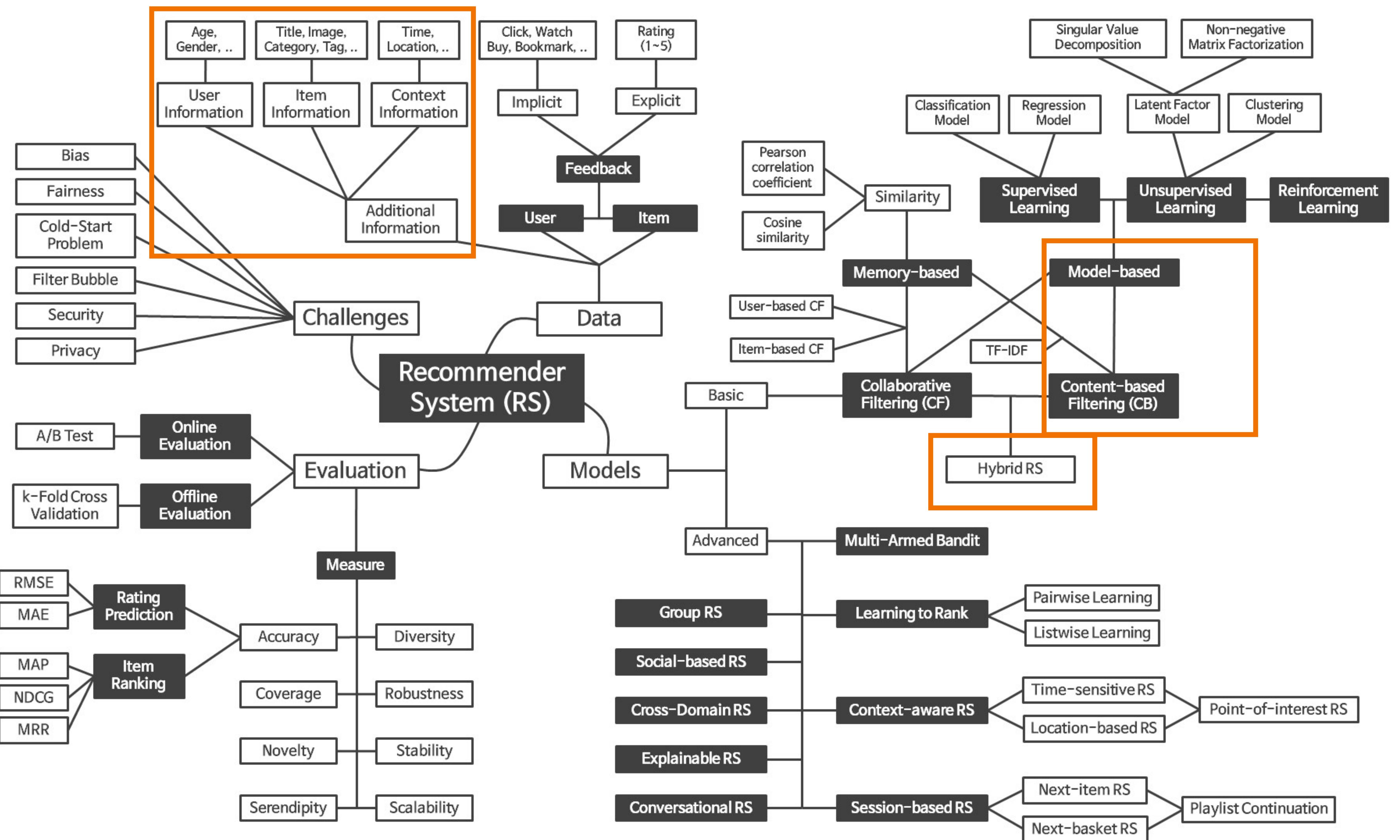


# **Контентные и гибридные подходы**

**Осиновсков Илья, 16.03.2025. AI masters.**

# **Введение**



# Вспомним прошлую лекцию

## Матричная факторизация

- Позволяет строить фактурные представления пользователей и объектов в одном общем пространстве, чтобы предсказывать их взаимодействия.
- Классические подходы плохо работают с дополнительными данными о пользователе и объекте.
- Если данных мало (холодный старт), то матричная факторизация становится неэффективной, так как мало взаимодействий.

# **Контент и контекст**

# Контент и контекст

- Цель: дать полезную информацию по взаимодействиям пользователей с помощью дополнительных описаний объекта и/или пользователя и/или взаимодействия

# Контент и контекст

- Цель: дать полезную информацию по взаимодействиям пользователей с помощью дополнительных описаний объекта и/или пользователя и/или взаимодействия
- Контент - описательное свойство объекта/пользователя, которое обычно не меняется во времени, либо делает это редко

# Контент и контекст

- Цель: дать полезную информацию по взаимодействиям пользователей с помощью дополнительных описаний объекта и/или пользователя и/или взаимодействия
- Контент - описательное свойство объекта/пользователя, которое обычно не меняется во времени, либо делает это редко
- Контекст - внешние условия или быстро меняющиеся признаки

# **Виды контента**

# Виды контента

- Объект
  - Название
  - Описание
  - Категория
  - Атрибуты
  - Картишка
  - ...

# Виды контента

- Объект
  - Название
  - Описание
  - Категория
  - Атрибуты
  - Картишка
  - ...
- Пользователь
  - Пол
  - Возраст
  - Имя
  - Фото
  - ...

# **Виды контекста**

# Виды контекста

- Время
- Геолокация
- Устройство
- Поисковый запрос
- Страница
- ...

# **Контентная рекомендательная система**

# Контентная рекомендательная система

- Система, которая анализирует содержимое объектов и пользователя, чтобы предложить ему то, что наиболее соответствует его предпочтениям.

# Контентная рекомендательная система

- Система, которая анализирует содержимое объектов и пользователя, чтобы предложить ему то, что наиболее соответствует его предпочтениям.
- Контентная система работает по следующей схеме:
  - Анализирует профиль пользователя, то есть определяет, что пользователю нравится, на основе его действий (например, он поставил высокие оценки ряду объектов).
  - Сравнивает профиль пользователя с контентными характеристиками новых объектов.
  - Ранжирует объекты по степени схожести с предпочтениями пользователя и рекомендует те, которые соответствуют его вкусам.

# Пример

Предположим, вы смотрите фильмы. Если вы посмотрели и высоко оценили такие фильмы, как:

- "Матрица" (жанры: научная фантастика, боевик),
- "Терминатор" (жанры: научная фантастика, боевик),

то система выведет, что вы любите **научную фантастику и боевики**. Она будет искать фильмы с похожими характеристиками и предложит, например:

- "Трон" (жанры: научная фантастика, боевик),
- "Интерстеллар" (жанры: научная фантастика, драма).

## COLLABORATIVE FILTERING

Оба пользователя прочитали



Она прочитала,  
рекомендуем ему

## CONTENT-BASED FILTERING

Этот пользователь прочитал

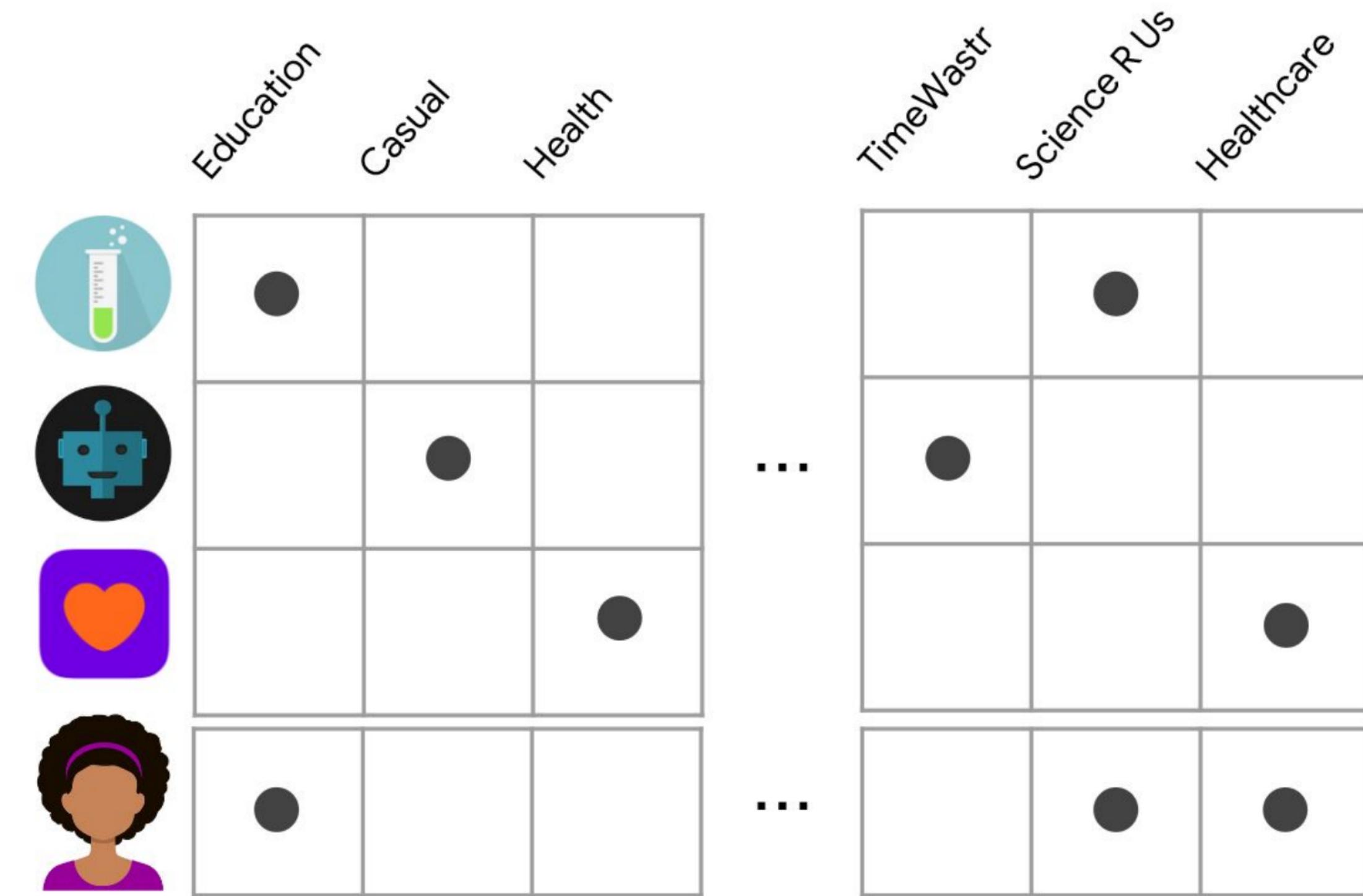


Рекомендуем ему

# **Контентная рекомендательная система**

# Content based similarity models

- $r_{ui} = \frac{\sum_{j \in I_u} sim(i, j) r_{uj}}{\sum_{j \in I_u} |sim(i, j)|}$
- $r_{ui} = \max_{j \in I_u, r_{uj} \geq \alpha} sim(i, j) r_{uj}$



# Функции похожести

Смотрите прошлую лекцию

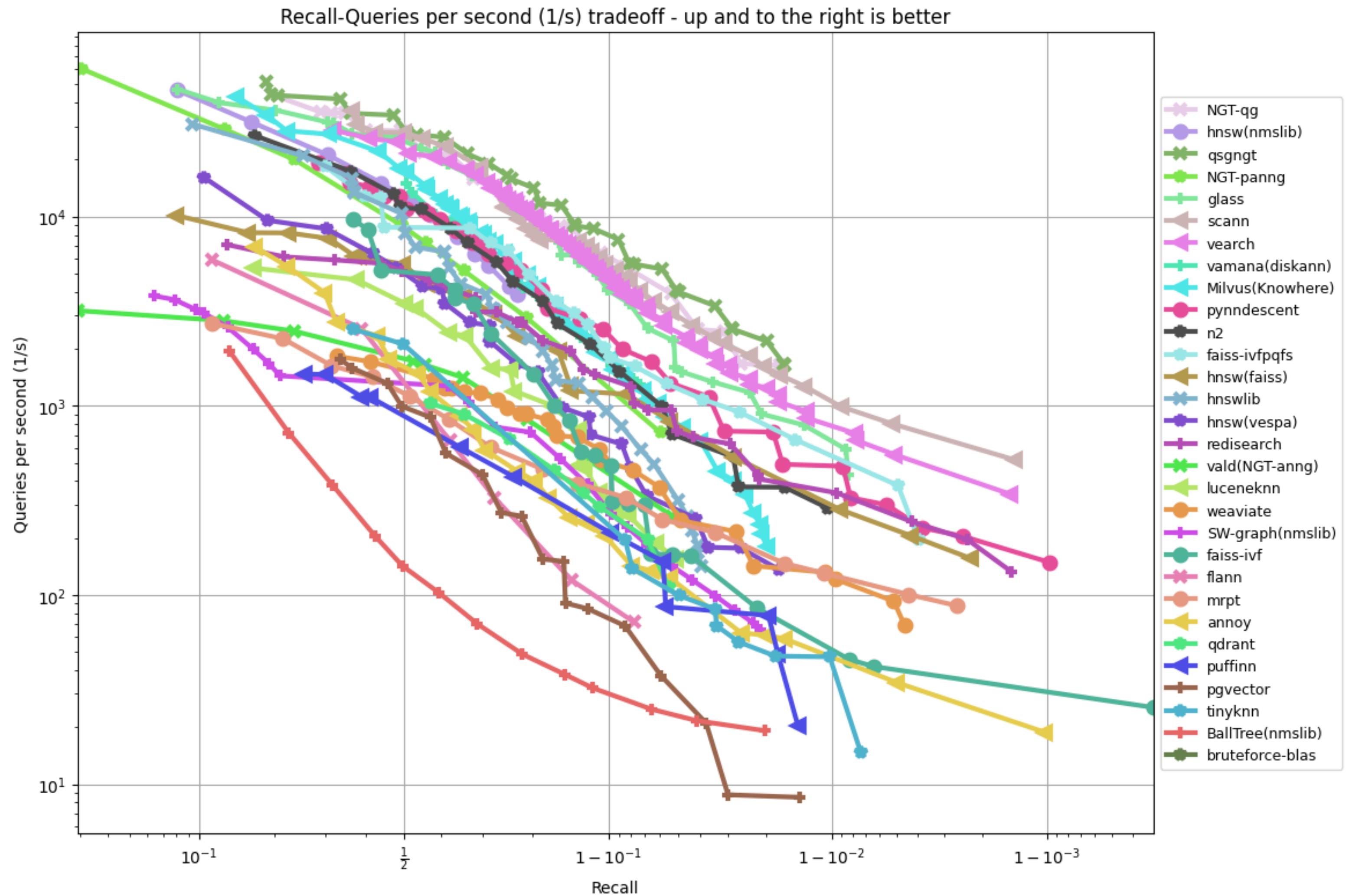
- Корреляция Пирсона
- Косинусная похожесть
- Мера Жаккара
- etc (76+ вариантов похожести)

# Примеры представления контента

- Численные
  - Бинаризация
  - Нормализация
- Категориальные
  - One-hot / multi label
- Текст
  - BOW
  - TFIDF/BM25
  - Предобученные эмбеддинги
- Картишка
  - Предобученные эмбеддинги

# Методы поиска похожих

- KNN
- ANN
- hnsw
- faiss
- annoy



# Content based similarity models

- Преимущества
  - Модель не нуждается в данных о других пользователях, поскольку рекомендации специфичны для этого пользователя. Это упрощает масштабирование для большого числа пользователей.
  - Модель может отражать конкретные интересы пользователя и рекомендовать нишевые товары, которые интересуют очень немногих других пользователей.
- Недостатки
  - Поскольку представление функций элементов в некоторой степени разрабатывается вручную, этот метод требует глубоких знаний предметной области. Таким образом, модель может быть настолько хороша, насколько хороши ее функции, спроектированные вручную.
  - Модель может давать рекомендации только на основе существующих интересов пользователя.

# Демографические модели

- Проблема холодного старта
- Разным соцдем группам требуются разные рекомендации
- Развитие:
  - Топ популярных
  - Соцдем топ
  - Персональный топ

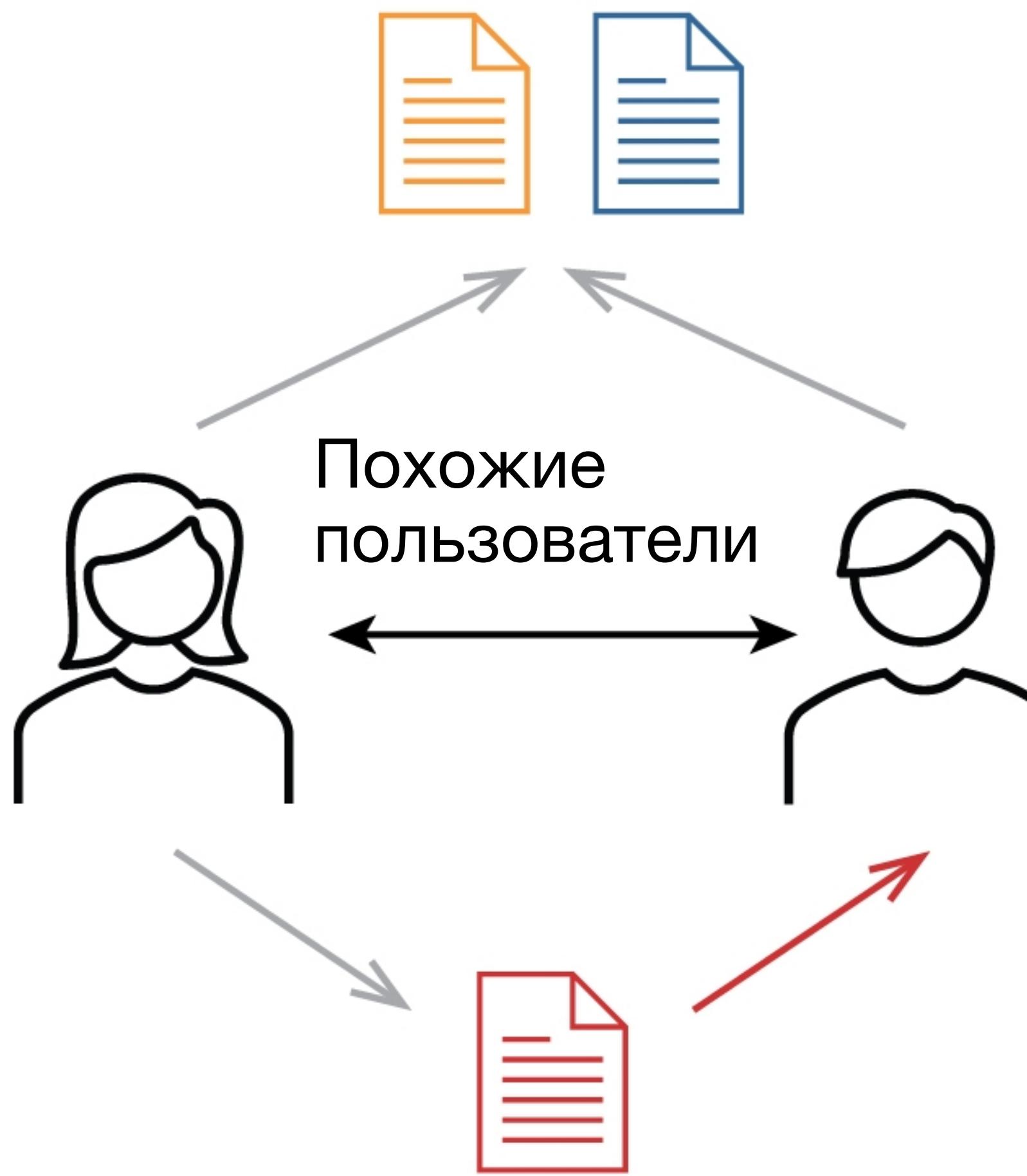
# Демографические модели

## Выводы

- Плюсы
  - Решаем проблему холодного старта
- Минусы
  - Не используем важный сигнал о взаимодействиях

## COLLABORATIVE FILTERING

Оба пользователя прочитали



Она прочитала,  
рекомендуем ему

## CONTENT-BASED FILTERING

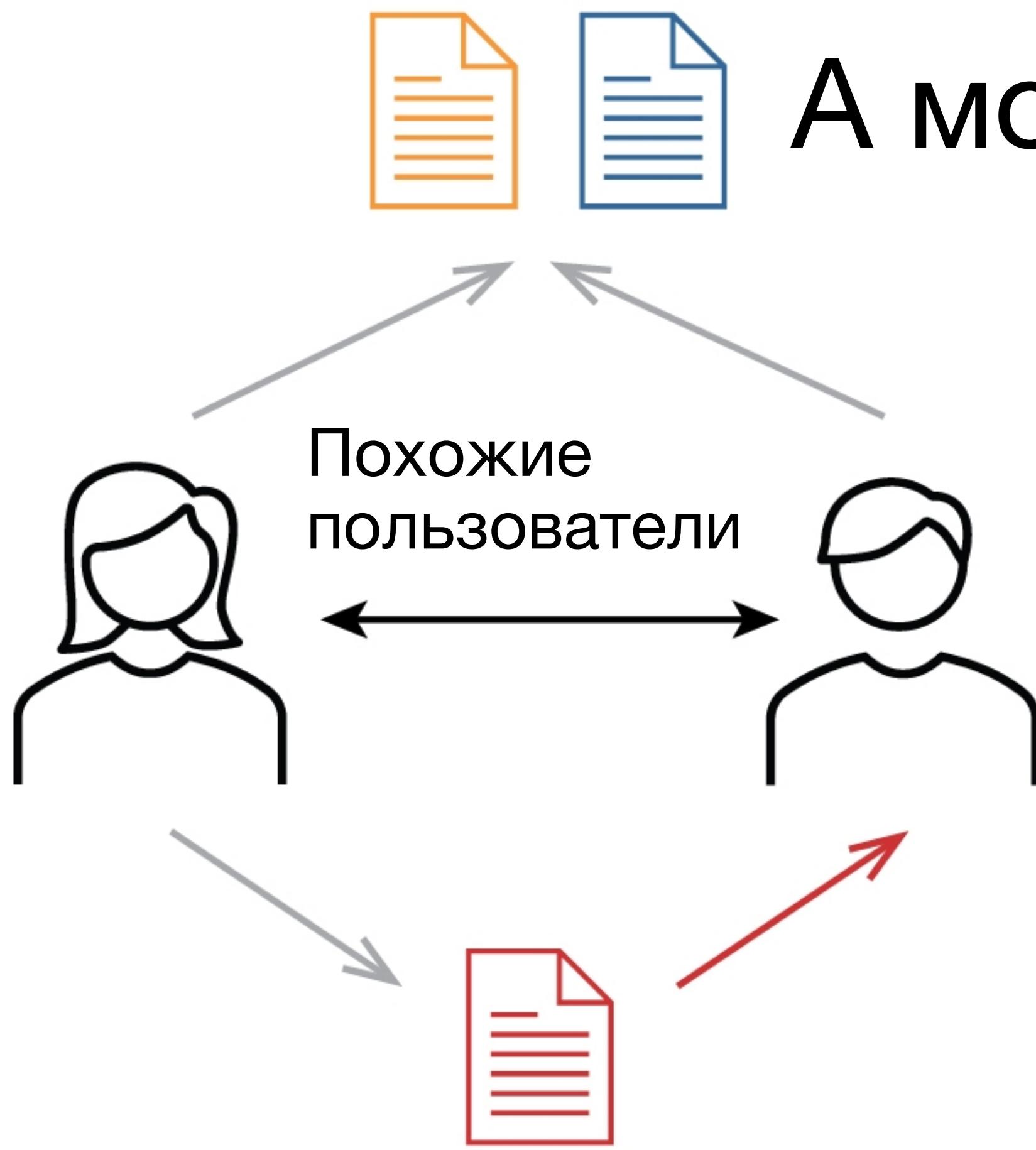
Этот пользователь прочитал



Рекомендуем ему

## COLLABORATIVE FILTERING

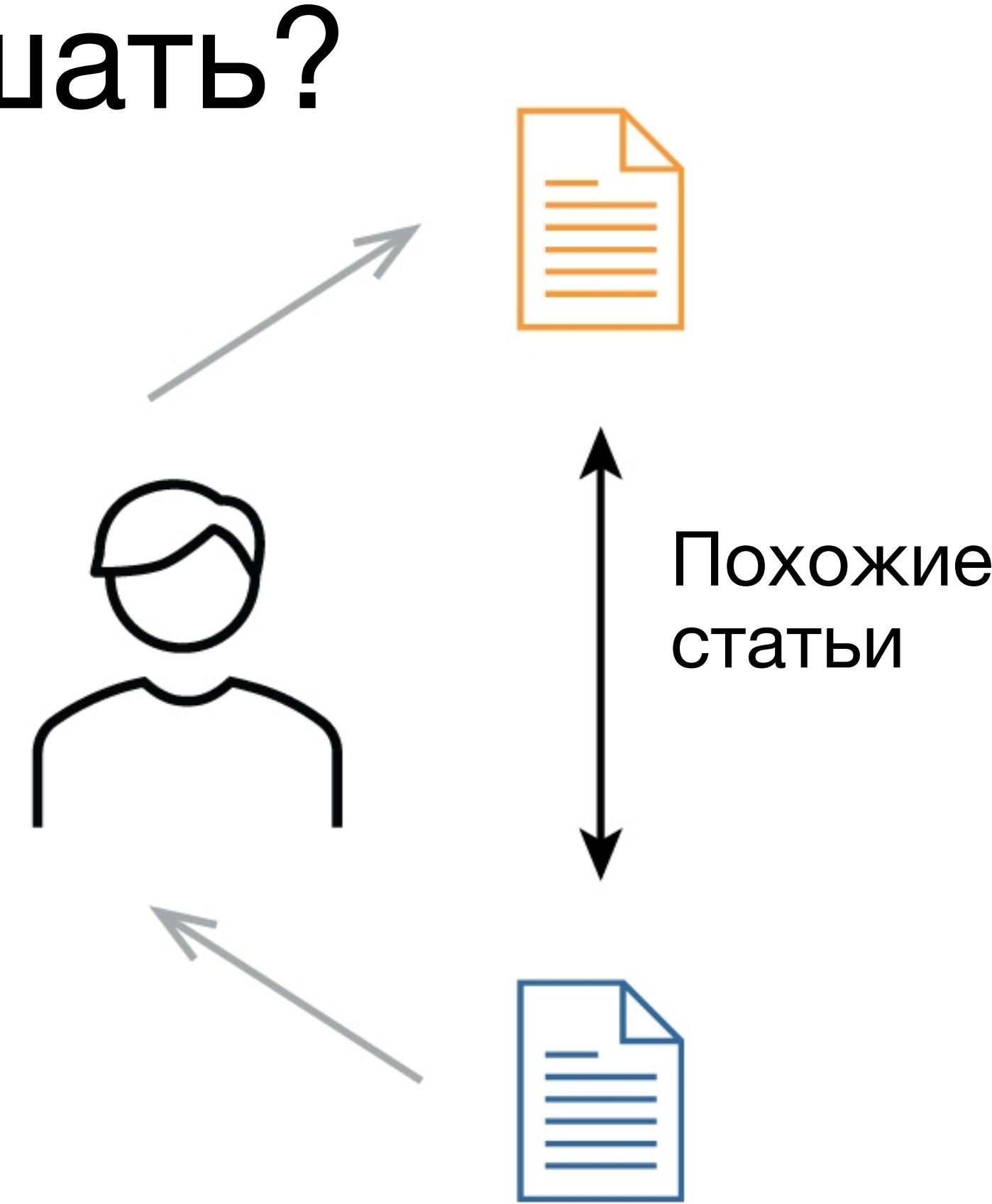
Оба пользователя прочитали



Она прочитала,  
рекомендуем ему

## CONTENT-BASED FILTERING

Этот пользователь прочитал



Рекомендуем ему

# Гибридные рекомендательные системы

# Гибридные рекомендательные системы

## Как добавить контент или контекст в коллаборативную фильтрацию?

User_Id	Item_Id	Date
5	4	2024-02-24
6	1	2024-02-24
5	3	2024-02-22
1	2	2024-02-13
5	1	2024-02-10
4	2	2024-02-08
2	3	2024-02-08
2	1	2024-02-07
....	....	....
1	4	2023-12-31



U/I	item 1	item2	item 3	item 4
User 1	0	1	0	1
User 2	1	0	1	0
User 3	0	1	0	1
User 4	0	1	0	0
User 5	1	0	1	1
User 6	1	1	0	1
User 7	0	1	1	0
User 8	1	0	0	1

# Гибридные рекомендательные системы

## Как добавить контент или контекст в коллаборативную фильтрацию?

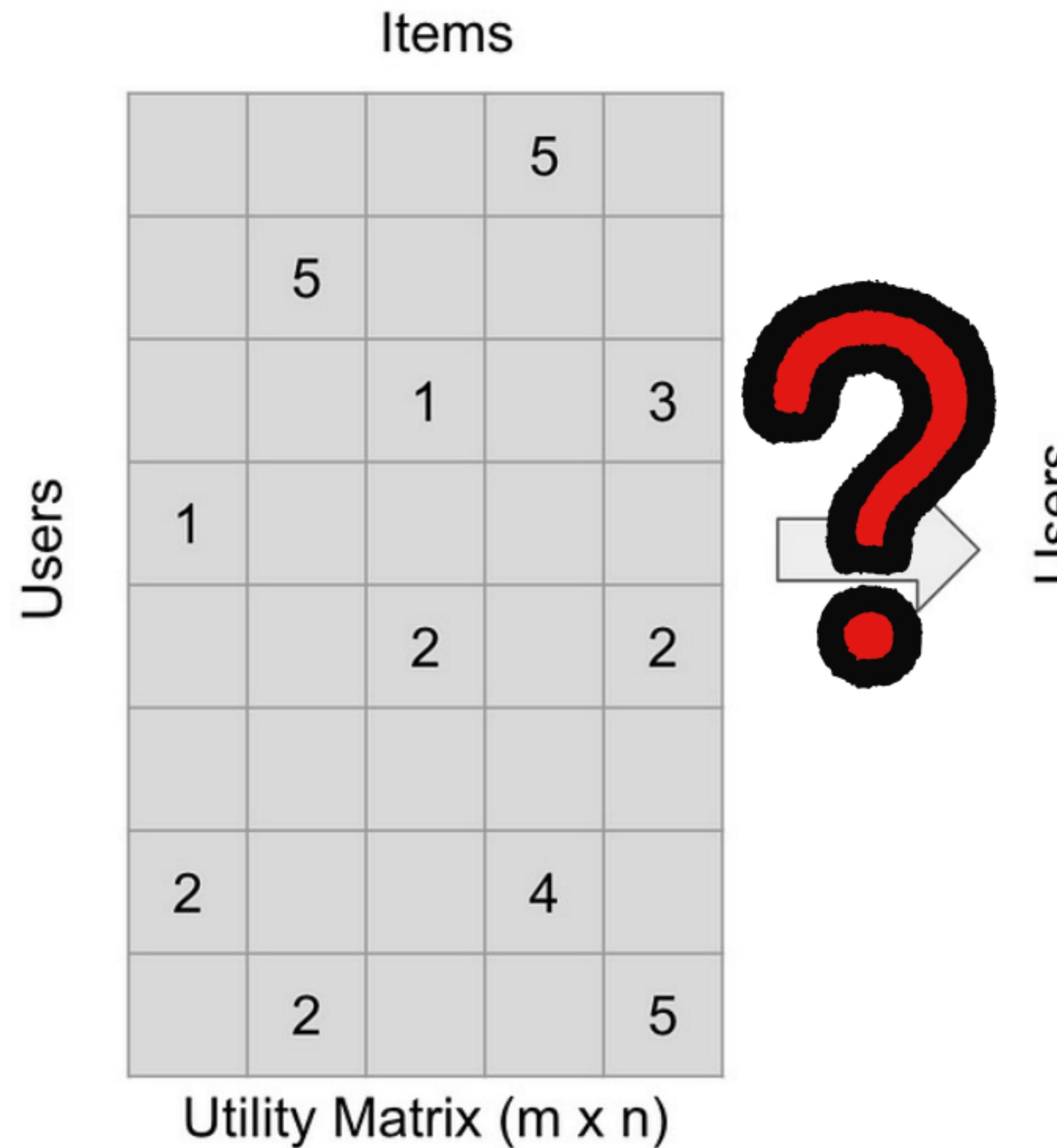
User_id	Item_id	Date
5	4	2024-02-24
6	1	2024-02-24
5	3	2024-02-22
1	2	2024-02-13
5	1	2024-02-10
4	2	2024-02-08
2	3	2024-02-08
2	1	2024-02-07
....	....	....
1	4	2023-12-31



U/I	item 1	item2	item 3	item 4
User 1	0	1	0	0.5
User 2	0.5	0	1	0
User 3	0	1	0	1
User 4	0	1	0	0
User 5	0.33	0	0.5	1
User 6	1	0.33	0	0.5
User 7	0	0.5	1	0
User 8	0.5	0	0	1

# Гибридные рекомендательные системы

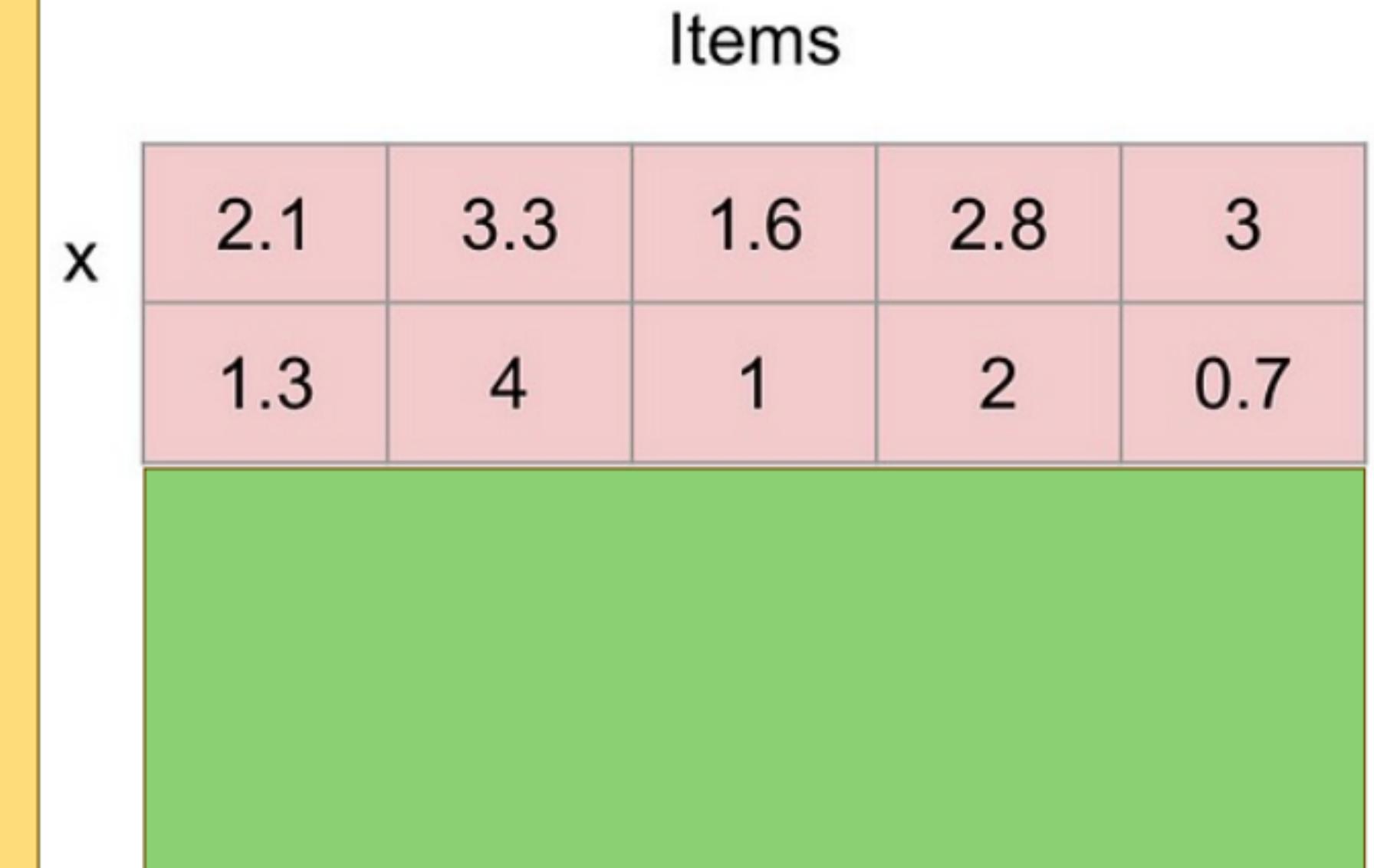
## Как добавить контент или контекст в коллаборативную фильтрацию?



Users

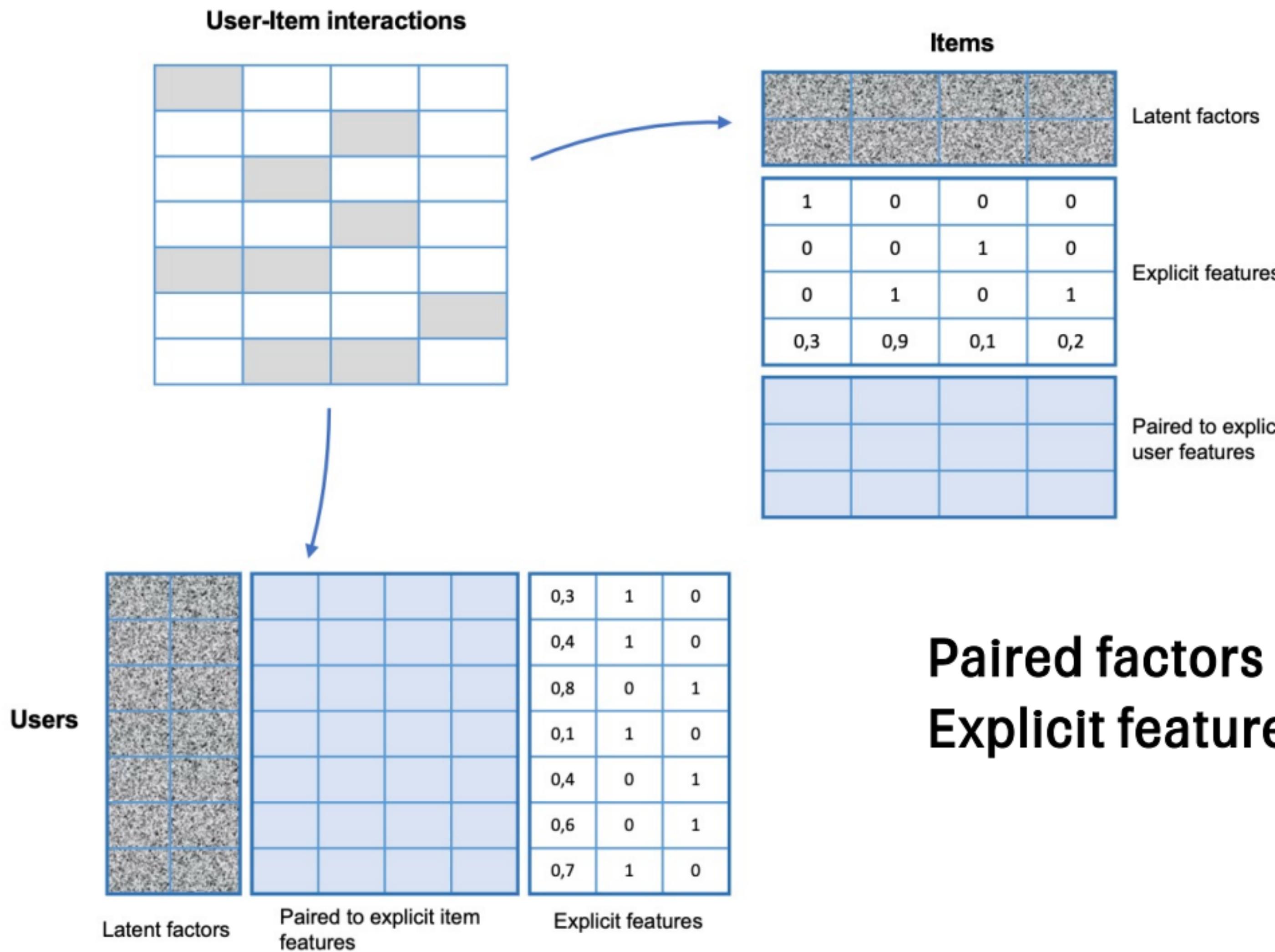
1.5	0.75
3	1.25
4	1.2
3.6	4.1
3.6	1.2
1.1	0.8
0.9	1.4
3.6	5.1

$m \times k$



# Гибридные рекомендательные системы

## Как добавить контент или контекст в коллаборативную фильтрацию?



**Paired factors – обучаемые.**  
**Explicit features – заморожены.**

# **Факторизационные машины**

# Факторизационные машины

Основная идея факторизационных машин заключается в том, чтобы моделировать взаимодействия между признаками в компактной факторизованной форме.

Идём от простого к сложному:

- В линейной регрессии у нас есть формула:  $y = w_0 + \sum_{i=1}^n w_i x_i$

# Факторизационные машины

Основная идея факторизационных машин заключается в том, чтобы моделировать взаимодействия между признаками в компактной факторизованной форме.

Идём от простого к сложному:

- В линейной регрессии у нас есть формула:  $y = w_0 + \sum_{i=1}^n w_i x_i$
- Если мы хотим учесть взаимодействия между признаками, традиционный подход будет добавлять попарные произведения признаков:

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n w_{ij} x_i x_j$$

# Факторизационные машины

Основная идея факторизационных машин заключается в том, чтобы моделировать взаимодействия между признаками в компактной факторизованной форме.

Идём от простого к сложному:

- В линейной регрессии у нас есть формула:  $y = w_0 + \sum_{i=1}^n w_i x_i$
- Если мы хотим учесть взаимодействия между признаками, традиционный подход будет добавлять попарные произведения признаков:
$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n w_{ij} x_i x_j$$
- Однако число взаимодействий  $w_{ij}$  быстро растёт, особенно при большом количестве признаков.

# Факторизационные машины

- Факторизационные машины решают эту проблему с помощью разложения взаимодействий на вектора:

$$\sum_{i=1}^n \sum_{j=i+1}^n w_{ij} x_i x_j \rightarrow \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

где

$\mathbf{v}_i \in \mathbb{R}^k$  - вектор для признака  $x_i$

$k$  - размер скрытого пространства

- Теперь мы учим только векторы  $\mathbf{v}_i$ , а не огромное количество  $w_{ij}$

# Факторизационные машины

## Математическая формулировка

- Формула для факторизационной машины второго порядка выглядит следующим образом:

$$y = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

где

$w_0$  – это смещение (bias),

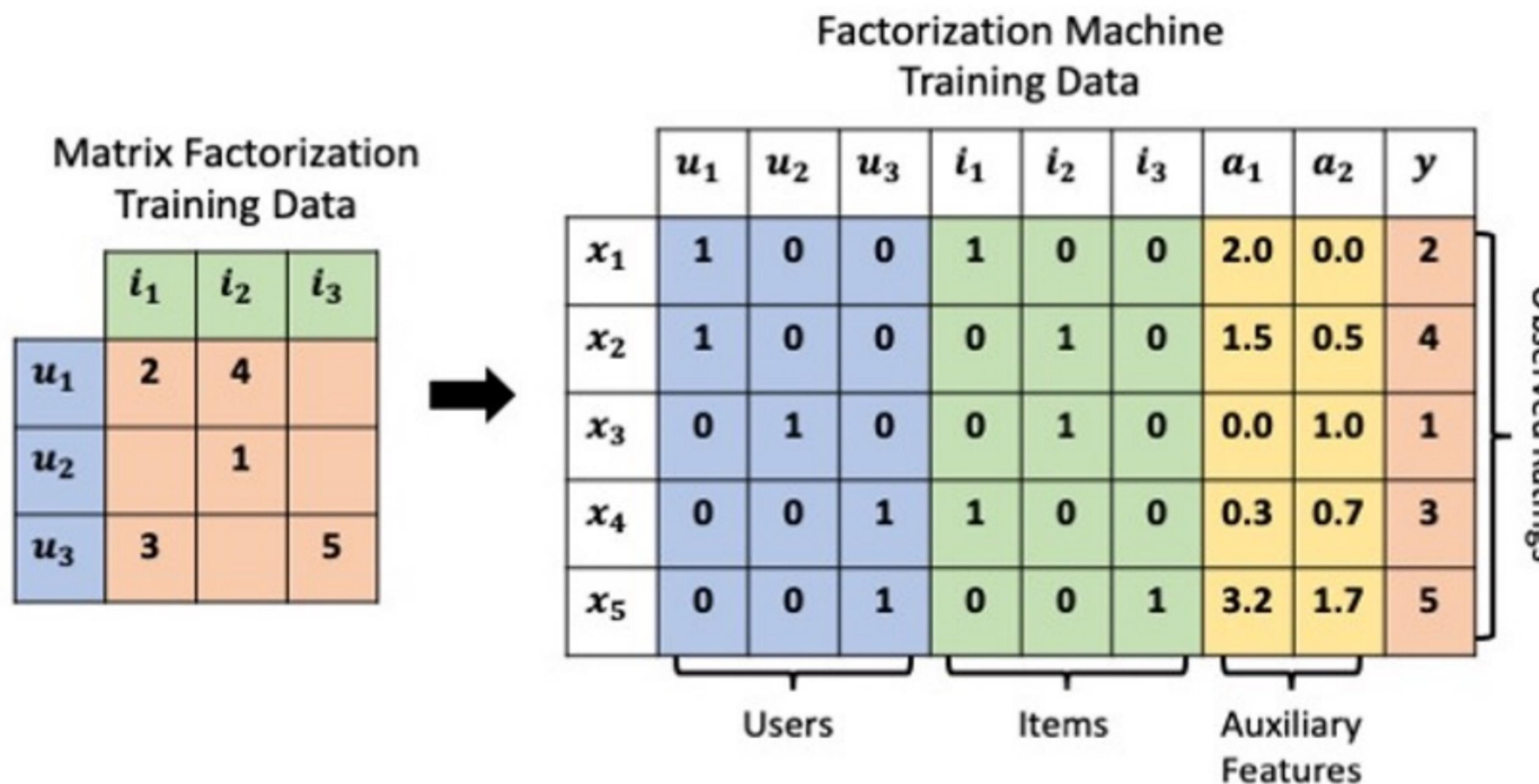
$w_i$  – веса для линейных признаков,

$\mathbf{v}_i$  – вектор факторизации для  $i$ -ого признака,

$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \sum_{f=1}^k v_{i,f} v_{j,f}$  – скалярное произведение векторов.

# Факторизационные машины

## Схема данных



# Факторизационные машины

## Схема данных

	Feature vector $\mathbf{x}$														Target $y$							
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
	User				Movie					Other Movies rated					Last Movie rated							

# Факторизационные машины

## Преимущества

- **Моделирование взаимодействий:** Мы эффективно моделируем взаимодействия между признаками, используя компактное факторное представление.
- **Генерализация:** Даже на разреженных данных (например, пользователь поставил один рейтинг) модель может обобщать за счёт факторизации.
- **Гибкость:** Факторизационные машины могут использовать как числовые, так и категориальные данные.

# LightFM

## Metadata Embeddings for User and Item Cold-start Recommendations

### 2.1 Motivation

The structure of the LightFM model is motivated by two considerations.

1. The model must be able to learn user and item representations from interaction data: if items described as ‘ball gown’ and ‘pencil skirt’ are consistently all liked by users, the model must learn that ball gowns are similar to pencil skirts.
2. The model must be able to compute recommendations for new items and users.

# LightFM

## Metadata Embeddings for User and Item Cold-start Recommendations

The latent representation of user  $u$  is given by the sum of its features' latent vectors:

$$\mathbf{q}_u = \sum_{j \in f_u} \mathbf{e}_j^U$$

The same holds for item  $i$ :

$$\mathbf{p}_i = \sum_{j \in f_i} \mathbf{e}_j^I$$

The bias term for user  $u$  is given by the sum of the features' biases:

$$b_u = \sum_{j \in f_u} b_j^U$$

The same holds for item  $i$ :

$$b_i = \sum_{j \in f_i} b_j^I$$

The model's prediction for user  $u$  and item  $i$  is then given by the dot product of user and item representations, adjusted by user and item feature biases:

$$\hat{r}_{ui} = f(\mathbf{q}_u \cdot \mathbf{p}_i + b_u + b_i) \quad (1)$$

The model's prediction for user  $u$  and item  $i$  is then given by the dot product of user and item representations, adjusted by user and item feature biases:

$$\hat{r}_{ui} = f(\mathbf{q}_u \cdot \mathbf{p}_i + b_u + b_i) \quad (1)$$

There is a number of functions suitable for  $f(\cdot)$ . An identity function would work well for predicting ratings; in this paper, I am interested in predicting binary data, and so after Rendle *et al.* [16] I choose the sigmoid function

$$f(x) = \frac{1}{1 + \exp(-x)}.$$

The optimisation objective for the model consists in maximising the likelihood of the data conditional on the parameters. The likelihood is given by

$$L(\mathbf{e}^U, \mathbf{e}^I, \mathbf{b}^U, \mathbf{b}^I) = \prod_{(u,i) \in S^+} \hat{r}_{ui} \times \prod_{(u,i) \in S^-} (1 - \hat{r}_{ui}) \quad (2)$$

# LightFM

## Metadata Embeddings for User and Item Cold-start Recommendations

**LightFM (tags)**: the LightFM model using only tag features.

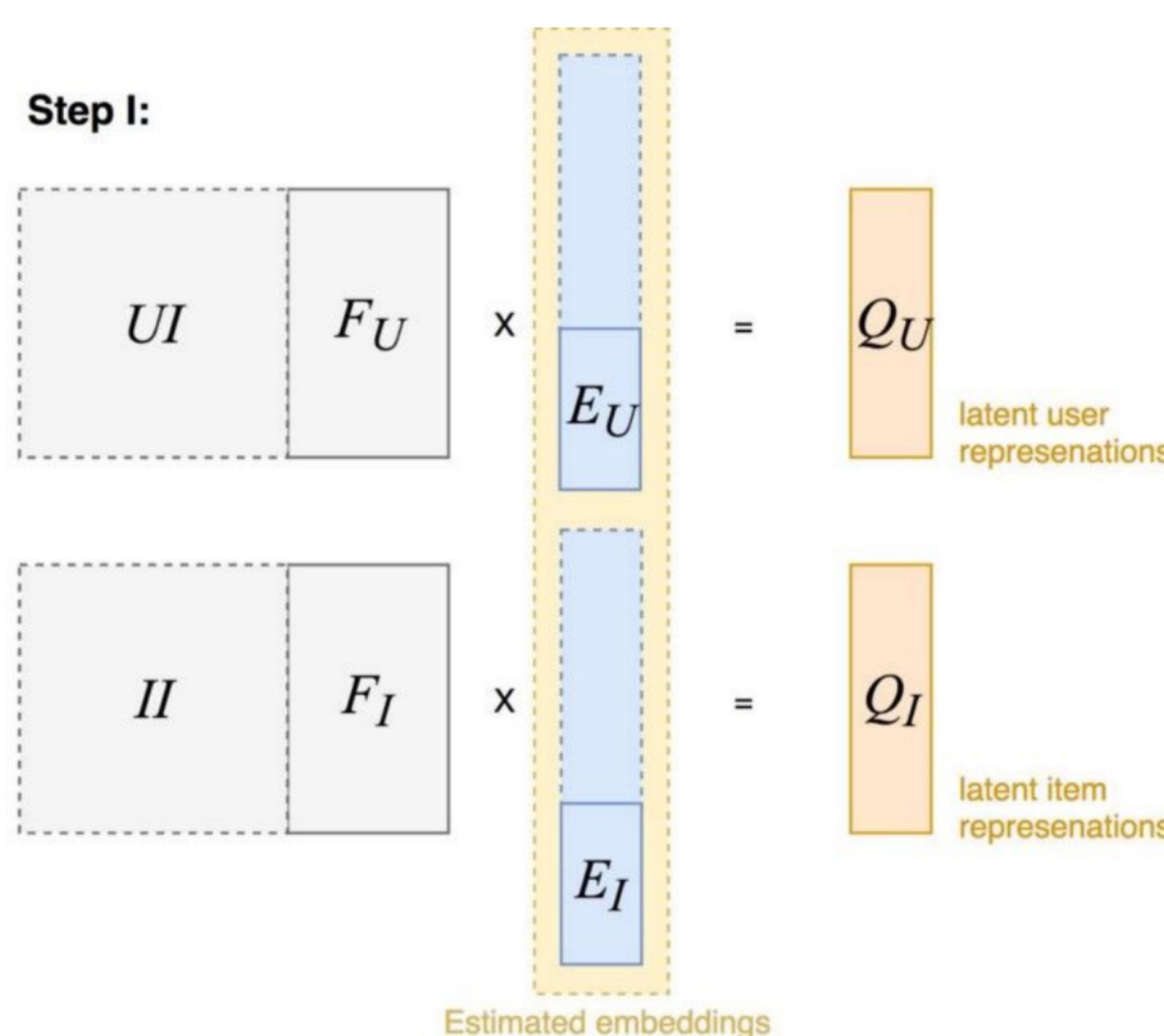
**LightFM (tags + ids)**: the LightFM model using both tag and item indicator features.

**LightFM (tags + about)**: the LightFM model using both item and user features. User features are available only for the CrossValidated dataset. I construct them by converting the ‘About Me’ sections of users’ profiles to a bag-of-words representation. I first strip them of all HTML tags and non-alphabetical characters, then convert the resulting string to lowercase and tokenise on spaces.

Table 1: Results

	CrossValidated		MovieLens	
	Warm	Cold	Warm	Cold
LSI-LR	0.662	0.660	0.686	0.690
LSI-UP	0.636	0.637	0.687	0.681
MF	0.541	0.508	0.762	0.500
LightFM (tags)	0.675	0.675	0.744	0.707
LightFM (tags + ids)	0.682	0.674	<b>0.763</b>	<b>0.716</b>
LightFM (tags + about)	<b>0.695</b>	<b>0.696</b>		

# LightFM



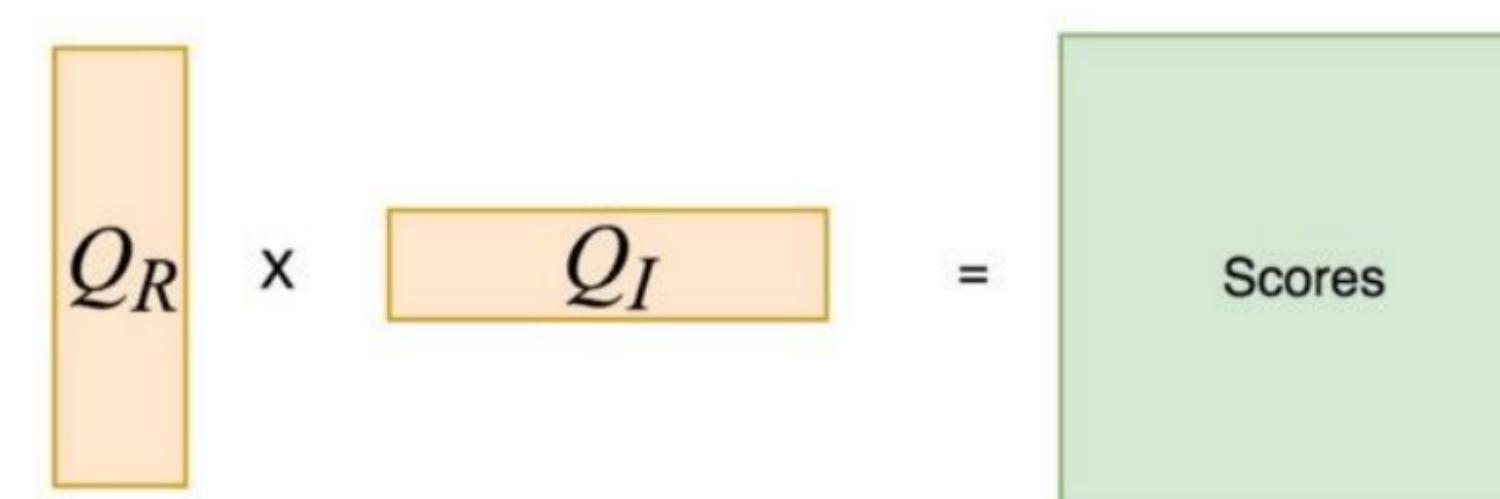
$$r_{ui} = \langle q_u, p_i \rangle + b_u + b_i$$

$$q_u = \sum_{j \in f_u} e_j^U$$

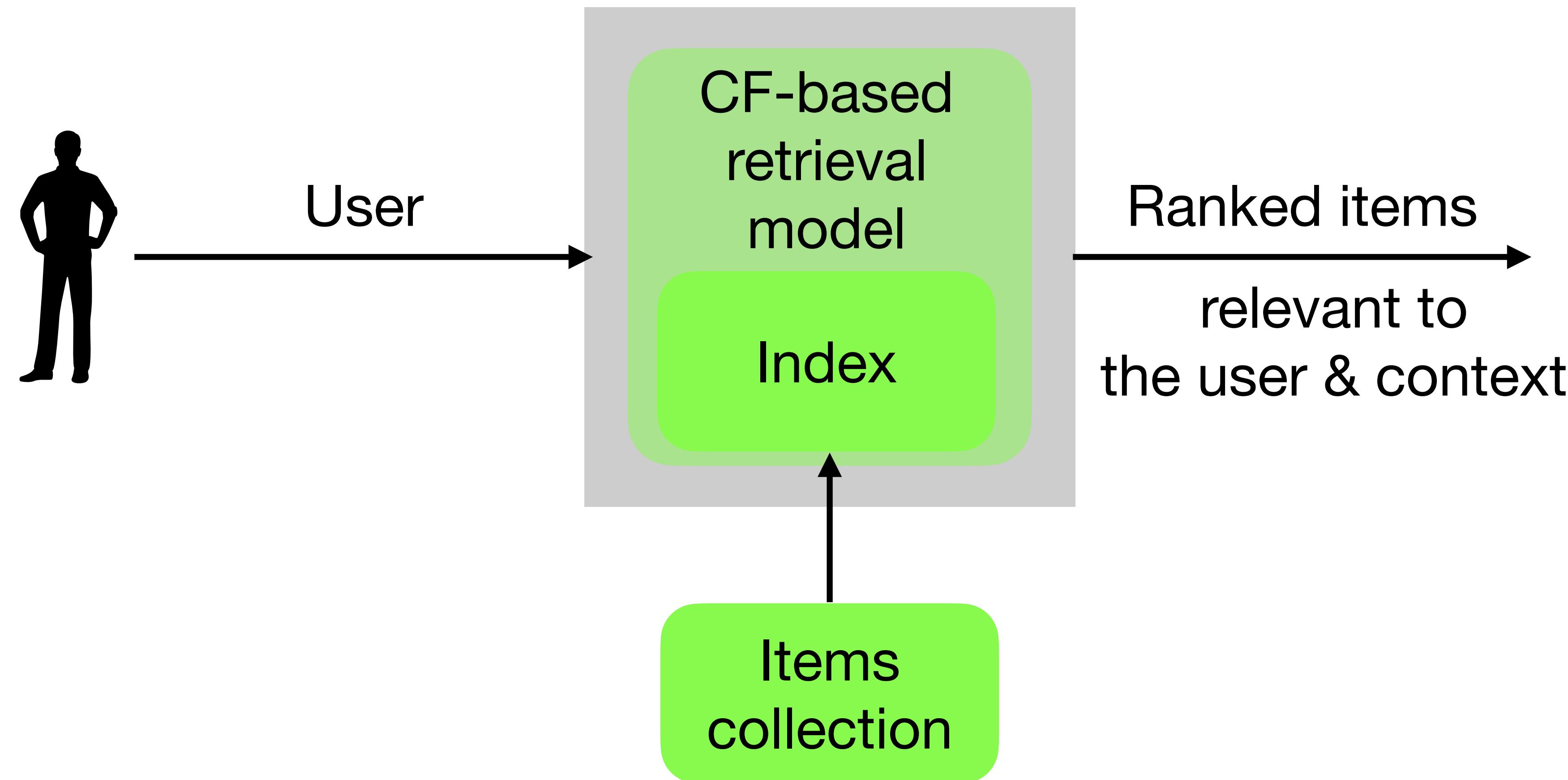
$$e_j^U = w_j^U e_j$$

$$b_u = \sum_{j \in f_u} b_j^U$$

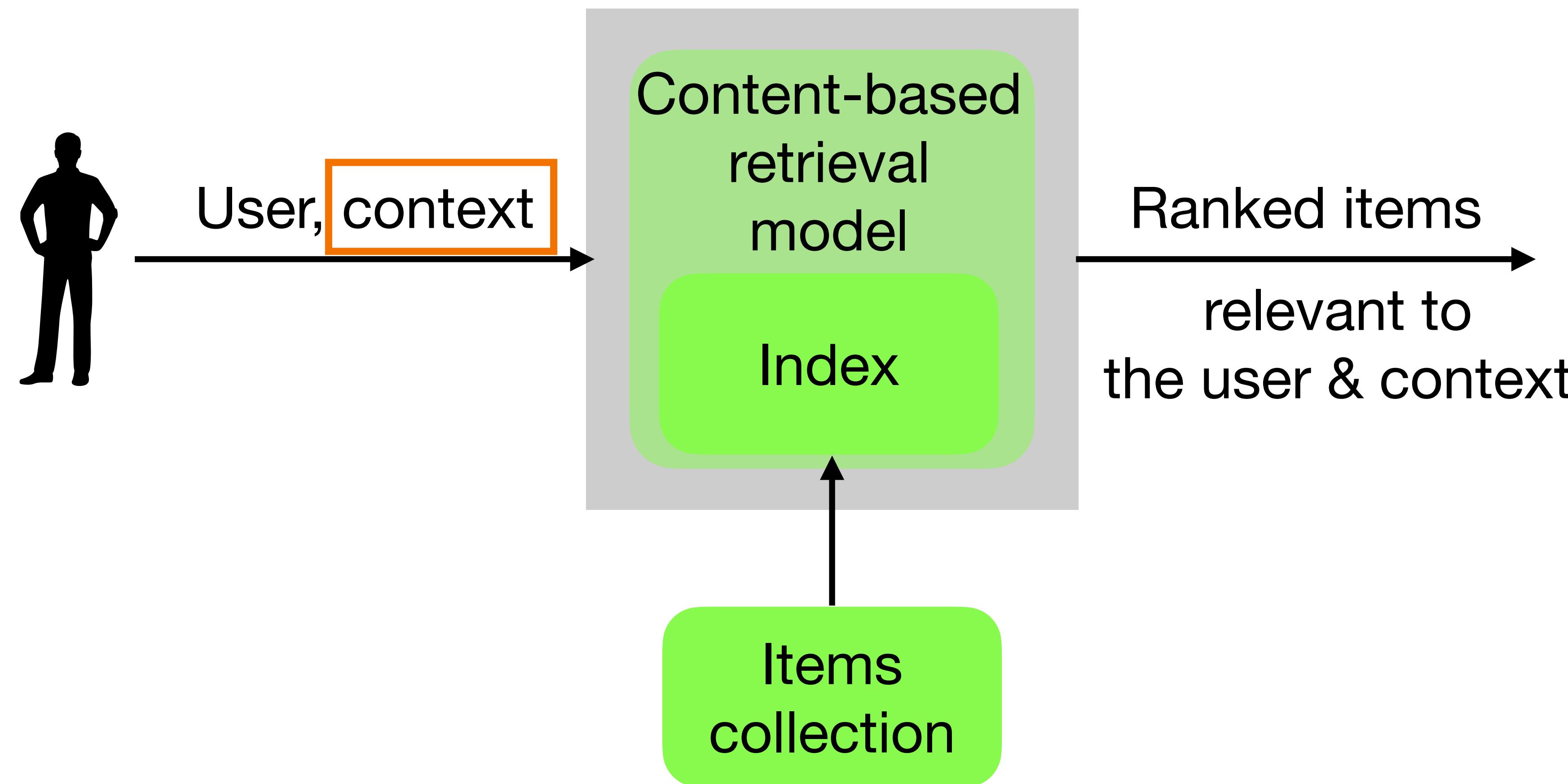
$$b_j^U = w_j^U b_j$$



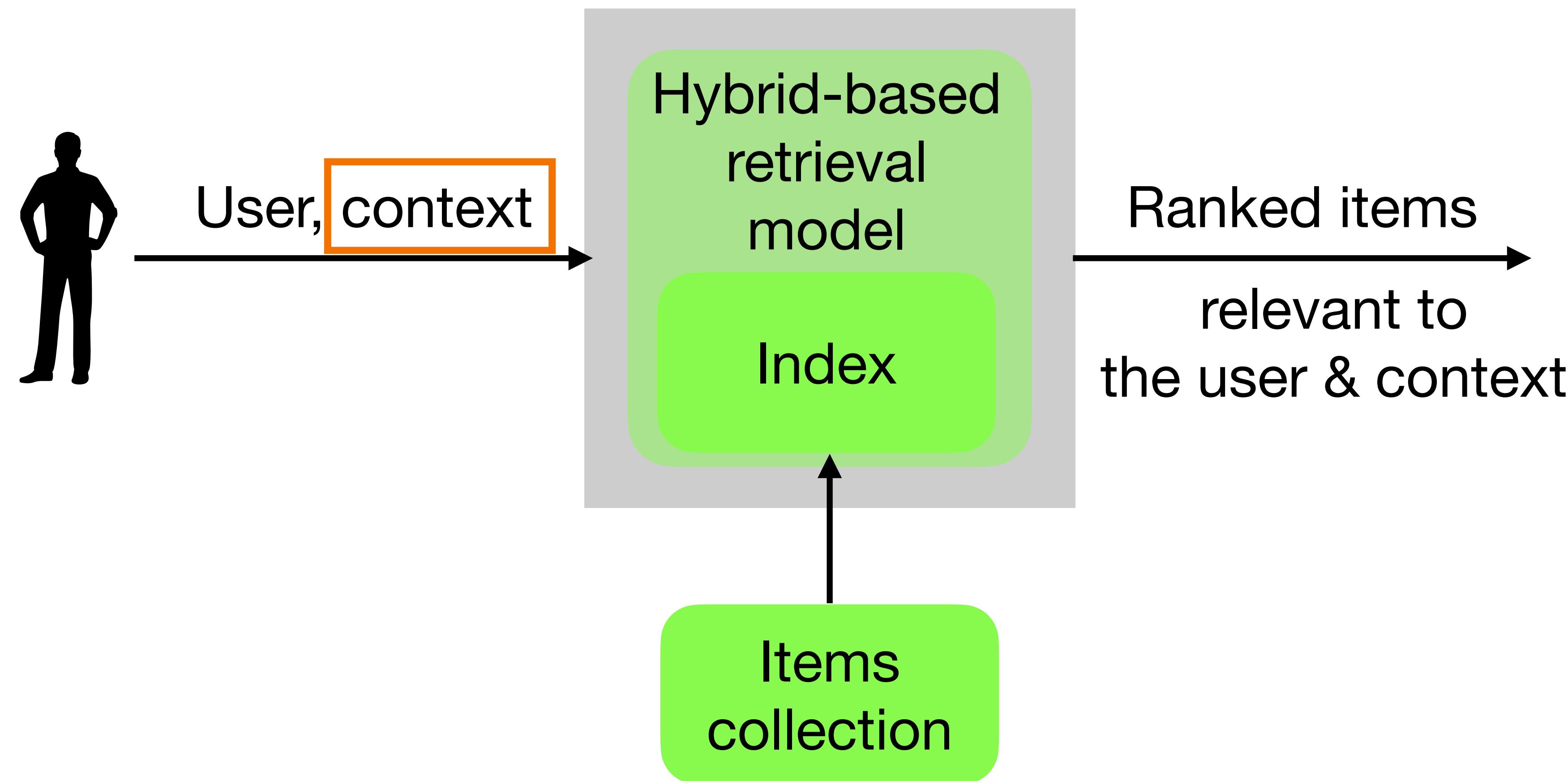
# Рекомендательная система



# Рекомендательная система



# Рекомендательная система



# **Вопросы**