

Введение в рекомендательные системы

Елена Кантонистова

План занятия

- Введение в рекомендательные системы
- Простые подходы:
 - ✓ мера Жаккара
 - ✓ Кластеризация
- Коллаборативная фильтрация

Что такое рекомендательная система

"Говорят, что Интернет покидает эпоху поиска и входит в эпоху открытий. В чем разница? Поиск - это когда вы ищете что-то. Открытие - это когда что-то замечательное, о существовании которого вы не знали, находит вас."

- CNN Money.

Что такое рекомендательная система

Рекомендательная система автоматически предсказывает товары/фильмы/музыку, которые могут заинтересовать пользователя на основе:

- прошлого поведения
- похожести на других пользователей
- похожести товаров/фильмов/музыки
- контекста (например: пользователь находится в поисковой выдаче по запросу "ipad")

Что рекомендуют?

Если вам понравился этот фильм, не пропустите



Остров проклятых
Shutter Island



Матрица
The Matrix



Престиж
The Prestige



Исходный код
Source Code



Помни
Memento

🔗 Знаете похожие фильмы? Порекомендуйте их...

⇒ все рекомендации к фильму (32)

Что рекомендуют?

OZON

Каталог

Везде ▾ Искать на Ozon



Войти

Заказы

Избранное

Корзина

Рекомендуем для вас



274 Р 409 Р



319 Р 830 Р



148 Р 229 Р



1175 Р 2849 Р



293 Р 777 Р

Зачем рекомендовать?

Вокруг слишком много информации!

"Люди читают около 10 МБ материалов в день, слушают около 400 МБ в день, и просматривают 1 МБ информации каждую секунду" - The Economist, November 2006.

"В 2015 потребление достигнет 74 ГБ в день" - UCSD Study 2014.

Зачем рекомендовать?

Полезно не только пользователям, но и бизнесу!

- ✓ Netflix: 2/3 фильмов просматриваются из рекомендаций.
- ✓ Google News: рекомендации генерируют на 38% больше просмотров.
- ✓ Amazon: 35% покупок совершаются через рекомендации.

Варианты постановки задачи

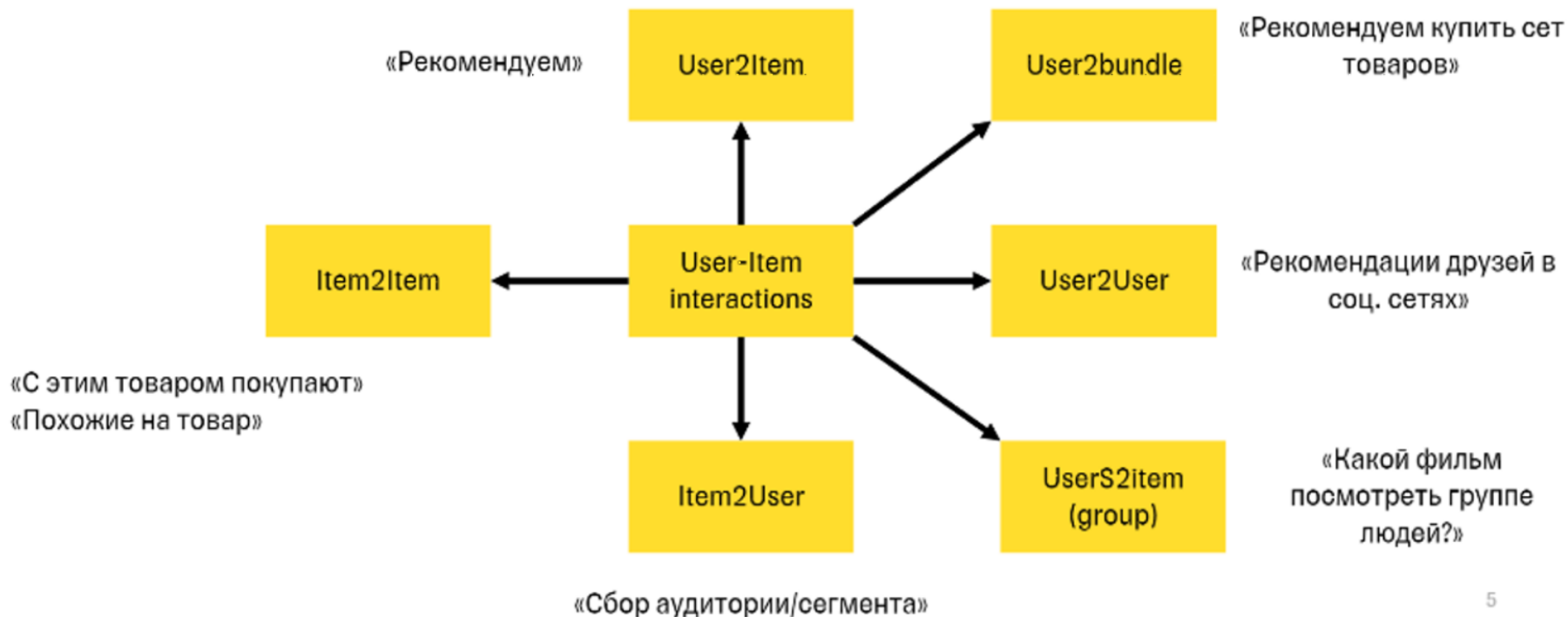
Задачу рекомендации наиболее подходящего (релевантного) товара можно сформулировать по-разному:

- как задачу *регрессии* - каждой паре пользователь-товар: (User, Item) ставить в соответствие меру удовлетворенности пользователя взаимодействием с товаром
- как задачу *классификации* - каждой паре (User, Item) предсказывать класс (чаще, бинарный) взаимодействия пользователя и товара (например, 1 - пользователь купит товар, 0 - пользователь не купит товар)
- как задачу *ранжирования* - каждому пользователю ставить в соответствие ранжированный список наиболее релевантных товаров, в порядке убывания релевантности

Рекомендации как задача ранжирования

1. **Поточечное (*point-wise*) ранжирование:** здесь каждой паре (запрос *Query*, документ *Document*) ставится в соответствие релевантность документа запросу. Поточечное ранжирование можно рассматривать как классификацию или регрессию (в зависимости от целевой переменной).
2. **Попарное (*pair-wise*):** в этом подходе мы пытаемся минимизировать количество пар документов, для которых был предсказан неправильный порядок. Эту задачу можно решать алгоритмами [RankNet](#), [LambdaRank](#), [LambdaMART](#).
3. **Списочное (*list-wise*):** тут мы ранжируем моделью всю выдачу документов разом. Эту задачу можно решать алгоритмами [SoftRank](#), [ListNet](#), [AdaRank](#).

Как можно рекомендовать?



Бейзлайны

- **Случайные рекомендации** - рекомендуем пользователю один или несколько случайных товаров
- **TopPopular** - рекомендуем пользователю один или несколько товаров, являющихся самыми популярными среди всех пользователей (товар больше всего покупают, или фильм чаще всего смотрят, или же рейтинг товара/фильма самый высокий и так далее). Такой подход не только бейзлайн, но и хорошее решение для "холодных" пользователей, то есть для клиентов, которые только что зарегистрировались, и мы пока что не знаем о них ничего
- **Популярное внутри какого-то сегмента** - это развитие подхода TopPopular: рекомендуем самое популярное внутри некоторого сегмента. Например, можем поделить пользователей по возрастным группам, и внутри каждой возрастной группы рекомендовать самое популярное

Простые подходы

Матрица взаимодействий (оценки)












Явное взаимодействие

Пользователи

Товары

Оценка

Понравится?

						
	2		2	4	5	
	5		4			1
			5		2	
		1		5		4
			4			2
	4	5		1		

Другая матрица взаимодействий (просмотры)

Неявное взаимодействие

						
	1	1	0		1	
	0	1	1			1
				1	1	0
		1	1		0	
		1				1

?

Явные и неявные оценки



Явные
(оценка по шкале)



Неявные
(посмотрел фильм)

Как понять что два пользователя похожи?

- Явные: **Схоже** оценивают те же товары
- Неявные: Оценивают **тот же набор** товаров

Подход 1: Мера Жаккара

Рекомендации по неявным взаимодействиям

- Можно посчитать, как часто два товара покупают вместе
- В пару к купленному первому товару рекомендовать самый близкий к нему по мере Жаккара

Частые пары покупок уже полезны

С этим товаром часто заказывают



Смартфон Apple iPhone
XS 4/64GB, серый
космос



Чехол/бампер YoHo для
iPhone X/XS, YCHIXXSC,
прозрачный

~~76 026~~ Р
= 75 685 Р

В корзину

Как посчитать схожесть по неявным оценкам?

	u1	u2	u3	u4
i1	1		1	1
i2		1		
i3	1	1		

$$J(i_1, i_2) = \frac{0}{4} = 0$$

Пример расчета

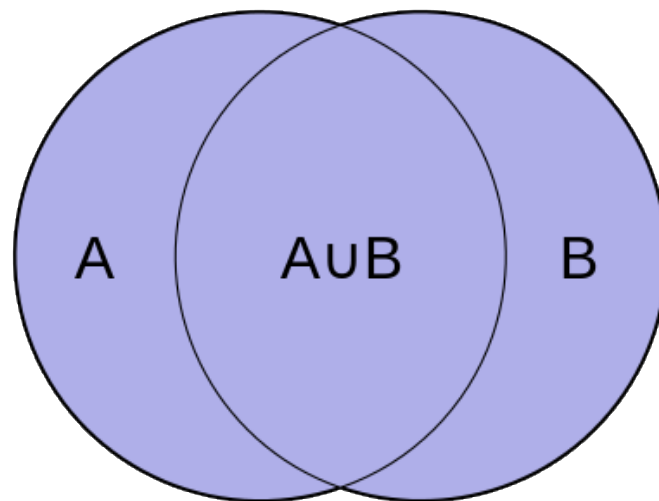
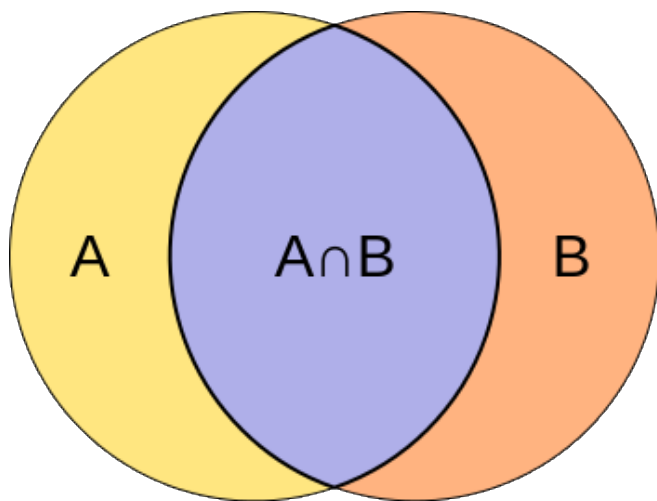
	u1	u2	u3	u4
i1	1		1	1
i2		1		
i3	1	1		

$$J(i_1, i_2) = \frac{0}{4} = 0$$

$$J(i_1, i_3) = \frac{1}{4} = 0.25$$

Мера Жаккара - учет неявных оценок!

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$



Найдем похожих исполнителей!



?



Мера Жаккара
Корреляция

Похожести по мере Жаккара

request	document	distance ↓
the beatles	the beatles	1
the beatles	dylan. bob	0.714
the beatles	led zeppelin.	0.709
the beatles	the rolling stones	0.703
the beatles	pink fluid	0.701
the beatles	divid bowie	0.691
the beatles	simon and garfunkel	0.691
the beatles	who	0.689
the beatles	the flaming lips	0.687
the beatles	queen	0.687

Подход 2: Кластеризация пользователей

Кластеризация

Будем кластеризовать товары/исполнителей. В итоге мы получим товары, которые покупают вместе/исполнителей, которых слушают одни и те же люди.

Далее будем делать рекомендацию так: если пользователь купил некоторый товар, то рекомендуем ему товары из того же кластера, что и купленный товар.

K-Means для музыки (центроиды)

Тяжелый рок

Attribute	cluster_0 ↓
niI	0.031
metallica	0.015
iron maiden	0.011
tool	0.010
Ёdir en grey	0.009
nightwish[0.009
opeth	0.009
judas priest	0.009
in flames	0.009
dream theater	0.008
megadeth	0.008
black sabbath	0.008
rammstein	0.008
the misfits	0.007
johnny clash	0.007
ルートヴィヒ・...	0.007
marilyn manson	0.007

Рок

Attribute	cluster_1 ↓
the beatles	0.073
dylan. bob	0.017
pink fluid	0.015
the rolling stones	0.011
led zeppelin.	0.011
divid bowie	0.011
radiohead	0.009
queen	0.007
who	0.007
the gateful dead	0.007
young, neil	0.006
u2	0.006
the beach boys	0.006
the kinks	0.006
red hot chili pe...	0.006
phish	0.006
iohnny clash	0.006

Рэп

Attribute	cluster_4 ↓
lil' wayne	0.035
kanye west	0.031
jay-z	0.020
nas	0.019
common	0.016
atmosphere	0.014
t.i.	0.013
lupe the gorilla	0.013
outkast	0.012
the notorious b...	0.011
a tribe called q...	0.011
the roots featur...	0.011
eminem	0.010
j dilla	0.010
50 cent	0.009
tupak shakur	0.009
ghostface killah	0.009

Поп

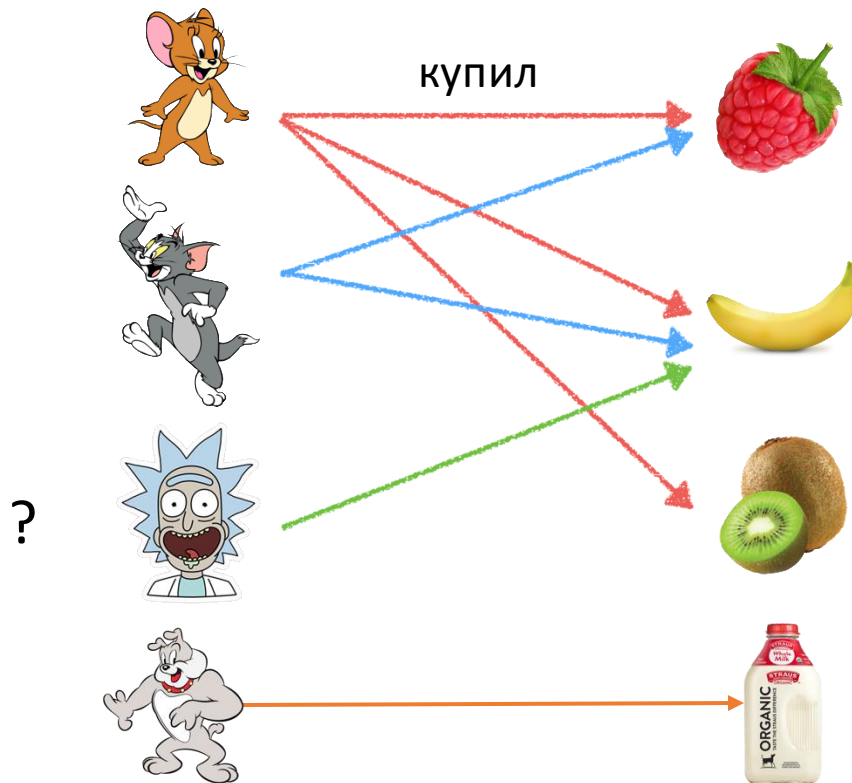
Attribute	cluster_5 ↓
coldplay	0.019
britney spears	0.018
madonna	0.012
johnson jack	0.012
linkin park	0.011
jason mraz	0.010
john mayer	0.010
브아	0.009
muse	0.009
enya	0.009
u2	0.009
evanescence	0.009
relient k	0.007
tori amos	0.007
depeche Mode	0.007
micheal bublè	0.006
kelly clarkson	0.006

Основные подходы к построению рекомендаций

- **Collaborative Filtering:** рекомендуем товары, основываясь на прошлом поведении пользователя и всех остальных пользователей
- **Content-based:** рекомендации, основанные на схожести свойств товаров
- **Matrix Factorization:** рекомендации, основанные на разложении матрицы оценок "пользователь-товар" в произведение матриц меньшей размерности
- **Neural Networks:** рекомендации, полученные с помощью нейросетевых подходов

Коллаборативная фильтрация

Коллаборативная фильтрация



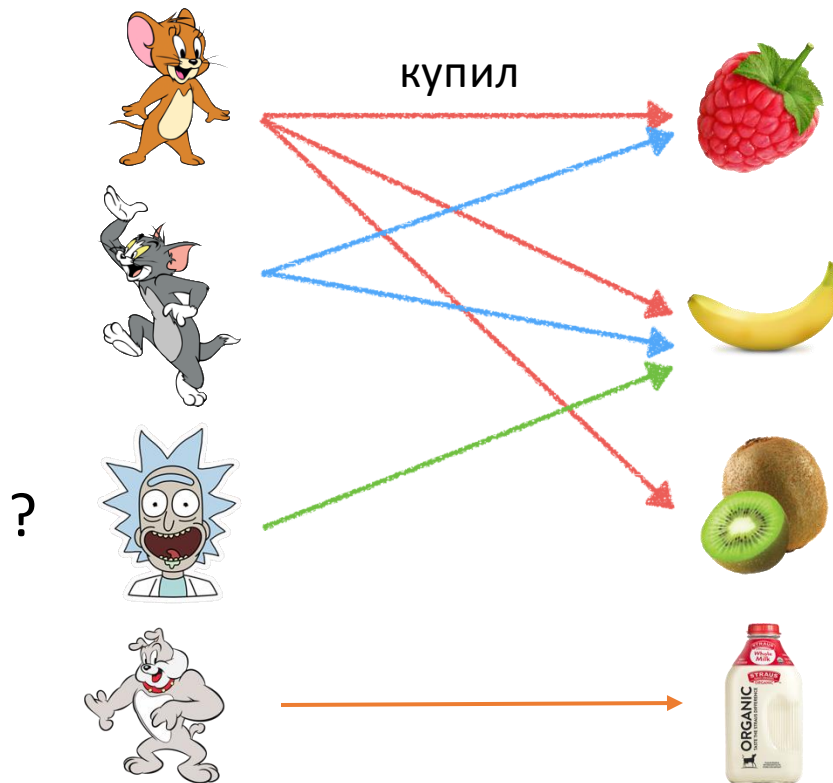
Рекомендации на основе похожести товаров

Как сделать рекомендацию для пользователя user?

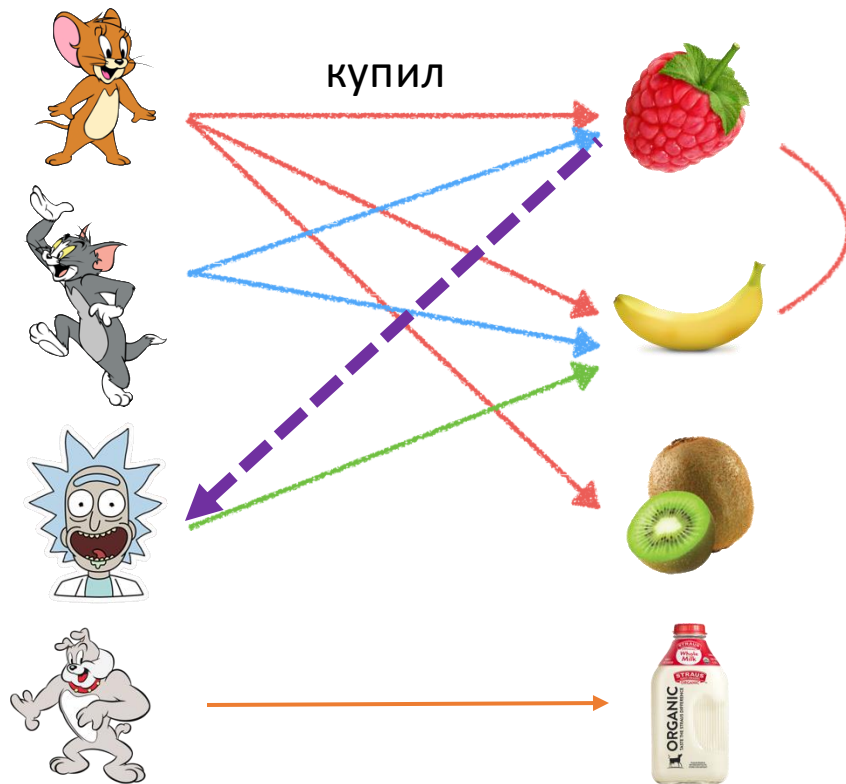
Идея: найдем похожих на user пользователей и порекомендуем ему понравившиеся им товары.

Такой подход называется user-based collaborative filtering.

Коллаборативная фильтрация



Рекомендации на основе похожести товаров



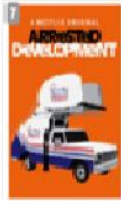
Рекомендации на основе похожести товаров

						
	1	1	0		1	
	0	1	1			1
				1	1	0
		1	1		0	
		1				1

?

Рекомендации на основе похожести товаров

	1	2	3	4	5	6
1	1	1	0		1	
2	0	1	1			1
3				1	1	0
4		1	1		0	
5		1				1



Рекомендации на основе похожести товаров

						
	1	1	0		1	
	0	1	1			1
				1	1	0
		1	1		0	
		1				1

Похожие пользователи

Рекомендации на основе похожести товаров





Коллаборативная фильтрация для матрицы явных взаимодействий

Пользователи

Товары

Понравится?

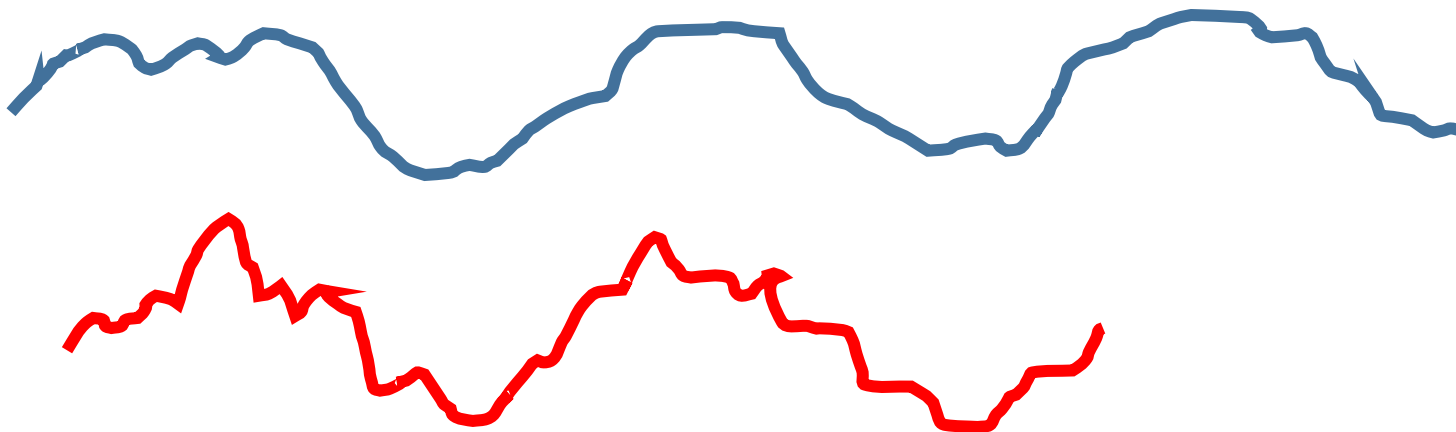
Оценка

						
	2		2	4	5	
	5		4			1
			5		2	
		1		5		4
			4			2
	4	5		1		

★ ★ ☆ ☆ ☆

Явные оценки: одинаково оценивают те же фильмы

Корреляция!



Как считать
похожесть
пользователей?

	1	2	3	4	5	6	
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							
50							
51							
52							
53							
54							
55							
56							
57							
58							
59							
60							
61							
62							
63							
64							
65							
66							
67							
68							
69							
70							
71							
72							
73							
74							
75							
76							
77							
78							
79							
80							
81							
82							
83							
84							
85							
86							
87							
88							
89							
90							
91							
92							
93							
94							
95							
96							
97							
98							
99							
100							

Обозначим через I_{uv} - множество товаров, которые оценили и пользователь u , и пользователь v .

Сходство пользователей u и v будем вычислять как корреляцию Пирсона:

$$sim(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

- здесь \bar{r}_u и \bar{r}_v - средние рейтинги пользователей по множеству товаров I_{uv} .

Пример.

							
				4	5		NA
	5	3	4			1	
	2		1		2		
	4	1		5		4	
	2		4			2	
		5		1			NA

Первый и последний пользователи не имеют общих оценок с целевым пользователем - их не рассматриваем.

Найдем похожесть нашего пользователя u и пользователя два (женщина) v :

- $I_{uv} = \{\text{Sherlock, Avengers, Walking dead}\}$ - общие просмотренные фильмы
- $\bar{r}_v = \frac{5+4+1}{3} = \frac{10}{3}$
- $\bar{r}_u = \frac{2+4+2}{3} = \frac{8}{3}$

Посчитаем w_{uv} :

- Числитель:

$$(5 - \frac{10}{3})(2 - \frac{8}{3}) + (4 - \frac{10}{3})(4 - \frac{8}{3}) + (1 - \frac{10}{3})(2 - \frac{8}{3}) = \frac{4}{3}$$

- Знаменатель:

$$\sqrt{(5 - \frac{10}{3})^2 + (4 - \frac{10}{3})^2 + (1 - \frac{10}{3})^2} \sqrt{(2 - \frac{8}{3})^2 + (4 - \frac{8}{3})^2 + (2 - \frac{8}{3})^2} = \sqrt{\frac{26}{3} \cdot \frac{8}{3}} = \frac{\sqrt{26 \cdot 8}}{3}$$

Получаем

$$\text{sim}(u, v) = \frac{4}{\sqrt{26 \cdot 8}} \approx 0.28$$

Как сделать рекомендацию для целевого пользователя?

Прогноз = средняя оценка пользователя + добавки от похожих пользователей с весами:

$$p_{ui} = \bar{r}_u + \frac{\sum_{v \in nn(u)} sim(u, v)(r_{vi} - \bar{r}_v)}{\sum_{v \in nn(u)} |sim(u, v)|},$$

где

- $nn(u)$ - множество пользователей, похожих на u (то есть тех, с кем мы вычисляли $sim(u, v)$)
- p_{ui} - оценка, которую согласно нашему алгоритму пользователь u поставит товару i .

Пример.

							
				4	5		NA
	5	3	4			1	0.28
	2		1		2		-1
	4	1		5		4	1
	2		4			2	
		5		1			NA

Для этого осталось вычислить корреляцию между целевым пользователем и пользователем 4 (в синей рубашке).
Здесь ситуация посложнее, так как при вычислении по формуле $\text{sim}(u, v)$ мы и в числителе, и в знаменателе получим 0. Однако мы видим, что пользователи одинаково оценивают фильмы относительно своего среднего (4 и 4, и 2 и 2) - поэтому корреляция между этими пользователями равна $\text{sim}(u, v) = 1$.

Посчитаем, какую оценку наш целевой пользователь u поставит фильму i - House of cards:

- $\bar{r}_u = \frac{2+4+2}{3} = \frac{8}{3}$
- $nn(u) = \{2, 3, 4\}$ (номера похожих пользователей v_2, v_3, v_4)
- $\text{sim}(u, v_2) = 0.14, \text{sim}(u, v_3) = -1, \text{sim}(u, v_4) = 1$
- $\bar{r}_{v_1} = \frac{10}{3}, \bar{r}_{v_2} = \frac{3}{2}, \bar{r}_{v_3} = 4$ (вычисленные ранее средние рейтинги по совпадающим фильмам)

Пользователь три (с похожестью -1 на нашего) не смотрел House of Cards, поэтому его мы в формуле не учитываем.

$$p_{ui} = \frac{8}{3} + \frac{0.28(3 - \frac{10}{3}) + 1(1 - 4)}{0.28 + 1} \approx 0.26$$