

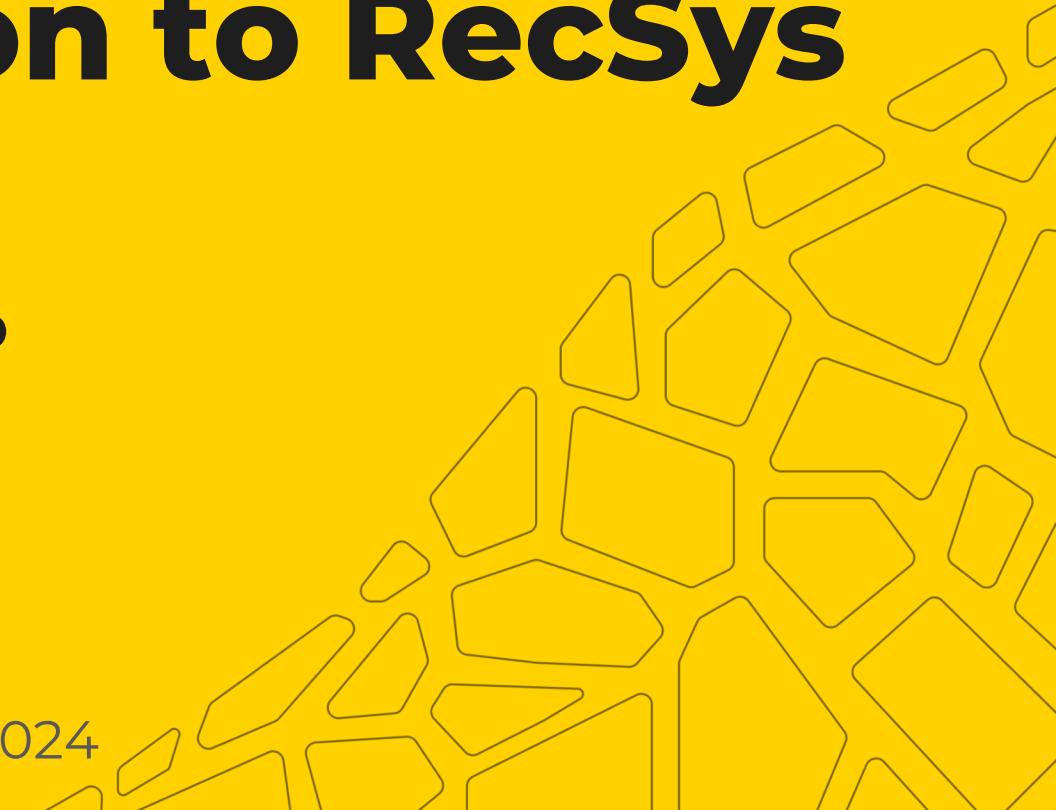
Introduction to RecSys

Vladislav Goncharenko

Senior researcher



MSU
spring 2024



Vladislav Goncharenko

- Author of machine learning courses and Masters program at MIPT
- ML researcher (MIPT)
- Team lead of video ranking team at Dzen (yandex.ru)
- Ex-team lead of perception team at self-driving trucks
- Open source fan



girafe
ai



Дзен

Science based learning techniques

- <https://www.youtube.com/watch?v=ddq8JIMhz7c>
- <https://www.coursera.org/learn/learning-how-to-learn>

The screenshot shows the Coursera platform. At the top, there's a navigation bar with the Coursera logo, an 'Explore' dropdown, a search bar containing 'What do you want to learn?', and a magnifying glass icon. Below the navigation, the breadcrumb trail shows the user is on the 'Personal Development' page under 'Browse'. The main content area features the logo for 'DEEP TEACHING SOLUTIONS' (dTS). The course title 'Learning How to Learn: Powerful mental tools to help you master tough subjects' is prominently displayed in large, bold, black text. Below the title, a note says 'Taught in English | 22 languages available | Some content may not be translated'. A blue button at the bottom left says 'Enroll for Free' and 'Starts Sep 13'. To the right of the course title, it says 'Financial aid available'. At the very bottom, a statistic reads '3,849,549 already enrolled'.

The image shows the front cover of a book. The title 'HOW TO STUDY & LEARN' is written in large, white, sans-serif capital letters across the center of the cover. Below the title, the author's name 'HUBERMAN LAB' is printed in a smaller white box. To the right of the title is a black and white portrait of a man with dark hair and a beard, identified as Adam Huberman. The background of the cover is solid black.



Communication

1. Course channel
2. Course chat
3. Support account @girafeai_support

Teacher will probably not answer to direct messages
due to large number of students!



Grading

Для МФК:

- 100% - тесты на занятиях

Для полноценного курса:

- 30% - тесты на занятиях
- 70% - лабораторные работы

Лабораторные работы:

3 шт в течение семестра в виде jupyter notebook-ов

Why recommendation systems are needed

**girafe
ai**

01



Examples of recommender systems

video recommendations

The image shows a tablet screen displaying the YouTube homepage. The interface includes a navigation bar with 'Home', 'Explore', 'Shorts', 'Subscriptions', 'Library', and 'History'. On the right side of the screen, there is a 'SIGN IN' button. Below the navigation bar, there is a search bar and a series of category filters: 'Live', 'Melodramas', 'House Music', 'Television comedy', 'Kino', 'Music', 'Terrestrial animals', 'Thrillers', 'Tourist destinations', 'Gaming', 'Universe', 'Dance music', and 'Cats'. The main content area displays a grid of video thumbnails. The top row includes a thumbnail for 'RECORD RUSSIAN MIX 24/7' (Record), a thumbnail for 'Vlad & Nika ve bebek Chris ile 1-10 arası sayılan öğrenin' (Vlad ve Nilita), a thumbnail for 'ДАЖЕ НУБ ПРОЙДЕТ ЭТОТ ПАРКУР ИЗ ЛАВЫ МАЙНКРАФТ! НУБ ПРОТИ...' (DaPlay), and a thumbnail for 'Medway chips and jelly Medved Valera' (Познаватель). The bottom row includes a thumbnail for 'Bob Marley Greatest Hits Reggae Song 2022 Top 20 Best Song Bob Marley' (Bob Marley), a thumbnail for 'Земфира – Как Живет Рок-Легенда и Самая Таинственная Певица России' (МедиШоу), a thumbnail for 'Тайны космоса или Земля до человека [Обзор]' (Это Интересно!), and a thumbnail for 'Mega Hits 2022 - The Best Of Vocal Deep House Music Mix 2022' (Hello Deep). Each thumbnail includes a play button icon and a timestamp.



Examples of recommender systems

goods recommendations

The screenshot shows a tablet displaying an e-commerce platform. At the top, there's a header with product details: "Изучаем Java | Сьерра Кэти, Бэйтс Берт", a rating of 5 stars, 17 reviews, a question and answer button, a compare button, and a "Добавить в корзину" (Add to cart) button with a price of 1407 ₽. Below the header, the text "Покупают вместе" (Buy together) is displayed. This section lists six books recommended based on purchase history:

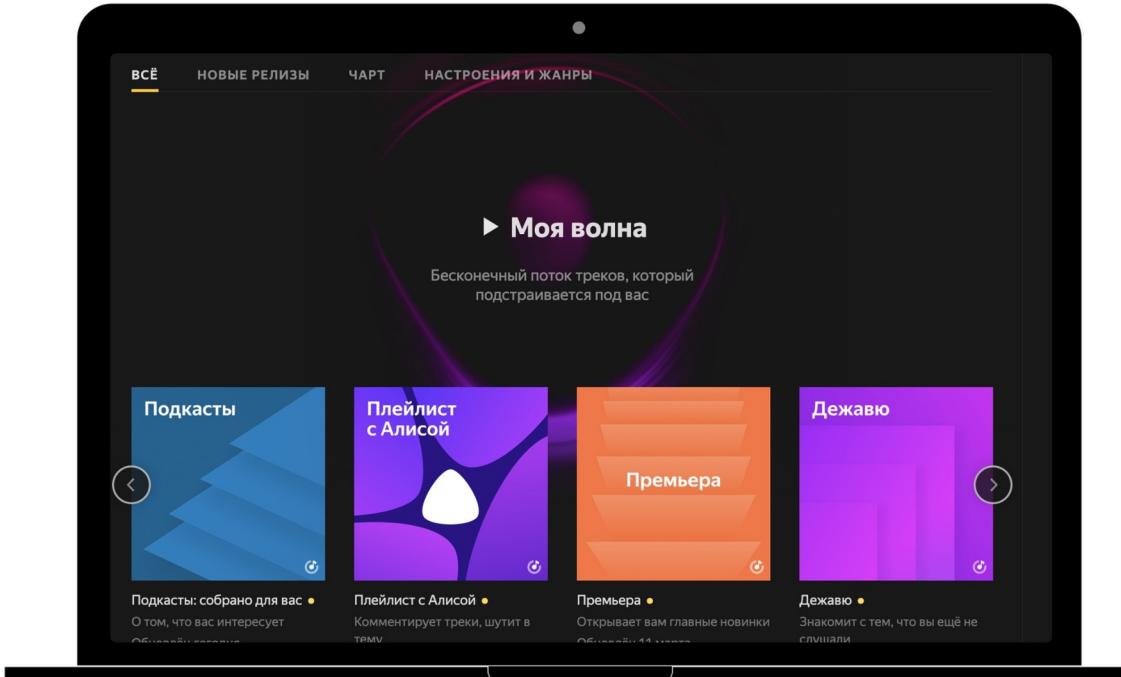
- 1 687 ₽ 2-447-Р. Код: тайный язык информатики [Твердая обл]. ★★★★★
- 765 ₽ 1-109-Р. Бестселлер. Нажми Reset. Как игровая индустрия рушит карьеры и дает второй шанс | Шреипер... ★★★★★
- 690 ₽ 863-Р. Работа с данными в любой сфере. Как выйти на новый уровень, используя аналитику... ★★★★★
- 3 454 ₽ 4-317-Р. Как вытащить из данных максимум. Навыки аналитики для неспециалистов | Джорда... ★★★★★
- 1 839 ₽ 3-189-Р. Распознавание образов и машинное обучение | Джорда... ★★★★★
- 20 марта доставят Ozon. NoSQL. Методология разработки нереляционных баз данных | Фуллер Мартин... ★★★★★

Each book listing includes a small image, a price, a discount percentage (e.g., -31%, -20%), a "В корзину" (Add to cart) button, and a note indicating delivery by March 20th via Ozon.



Examples of recommender systems

music recommendations



Goals



- Provide best user experience
- Provide business value



Examples of recommender systems

How to convince a person to stay on the video service?

- Suggest watching something else
- The proposed videos should be of interest to the user
- There should be variety in the proposals, so that they are not intrusive
- Etc

This is what recommendation systems do.

Netflix Prize

girafe
ai

02



Netflix Prize

On October 2, 2006, a competition began to recommend films from Netflix, at that time a popular DVD rental company



- predict the ratings of films from 1 to 5
- RMSE error
- 17700 films
- ~ 480 thousand customers
- ~ 100 million ratings
- prize is \$1,000,000
- lasted almost 3 years
- 5100 teams participated



Netflix Prize

★ Results on the train

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	The Ensemble	0.8553	10.10	2009-07-26 18:38:22
2	BellKor's Pragmatic Chaos	0.8554	10.09	2009-07-26 18:18:28

★ Results on the test

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22

The competition gave an impetus to the development of the field of recommendation systems and largely determined the vector of development

Problem statement

**girafe
ai**

03

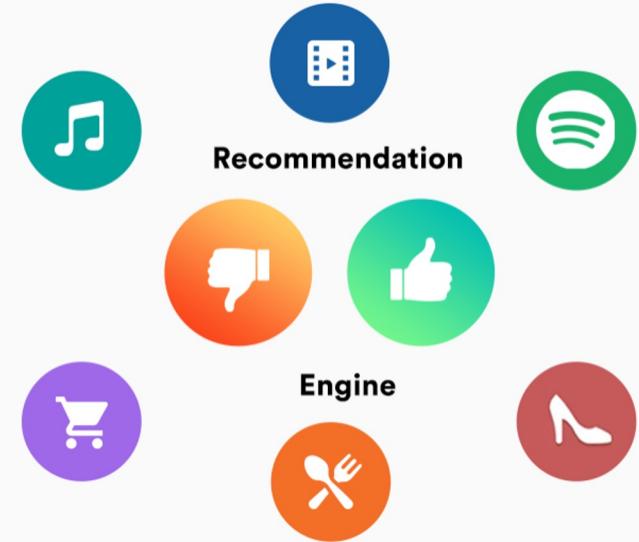


Definition

A **recommender system** is a subclass of information filtering system that provide suggestions for items that are most pertinent to a particular user.

Recommender systems are particularly useful when an individual needs to choose an item from a potentially overwhelming number of items that a service may offer.

[Common knowledge site](#)





Information retrieval subproblems

- Recommender systems
- Search engines
- Advertising



Formal problem statement

- ★ Specified U - set of users, I - set of objects (items), that we recommend
- ★ For each user $u \in U$ his interaction history: he interacted with the items $I_u \subset I$, to which I gave ratings.

$$R_u = (r_{ui})_{i \in I_u}$$

- ★ Such user ratings are called feedback

The goal is to offer the user new items that would be interesting to the user



How to understand that an item is interesting to the user?



What interesting item is?

We can assess this by the quality of interaction.

For example:

- For the product that it was put in the basket
- For the music that it was listened to to the end
- For the article that it was liked
- For videos that have watched at least half of it
- Etc

Feedback types

girafe
ai

04



Explicit feedback

Explicit feedback is actions that express the user's explicit attitude to the product.



rating of the film
from 1 to 5



like/dislike the track



product
review



Implicit feedback

Implicit feedback is any other information, user actions on the site/in the app.

It does not reflect its explicit relationship to the product, but can act as a proxy to explicit feedback.



video viewing
time



viewing an
article



purchase of
goods



Features of explicit feedback

- Usually such data is quite small and more difficult to obtain (1-10% of implicit usually)
- Dislike of an article does not mean that such articles should not be recommended to the user. Perhaps he is interested in the subject, but did not like this particular article
- Not all users leave explicit feedback, thus it is highly biased to this subset of users (usually about 10% of users)

There is a funnel of actions from implicit to the most complex ones



Features of implicit feedback

- There is more such data
- As a rule, implicit data can be trusted less, since they express only implicit signals, and not explicit user ratings. For example, a user could buy a product as a gift, but he does not like the product himself
- Since the implicit signal has a different nature, techniques other than explicit feedback are usually used in the recommendation system to optimize it



Collaborative matrix sparsity

Suppose we have 10^7 users and 10^6 items.

User rate 10 items on average. *What is the percentage of filled cells?*

$$10 * 10^6 / (10^7 * 10^6) = 10^{-6} = \mathbf{0.0001\%}$$

	Item										
User											
Peter											
Mary											
Vasia											
Kate											



Types of user representations

In general, we want to recommend items (music/movie/article/product) to the user so that the user is happy (we work on long-term metrics)

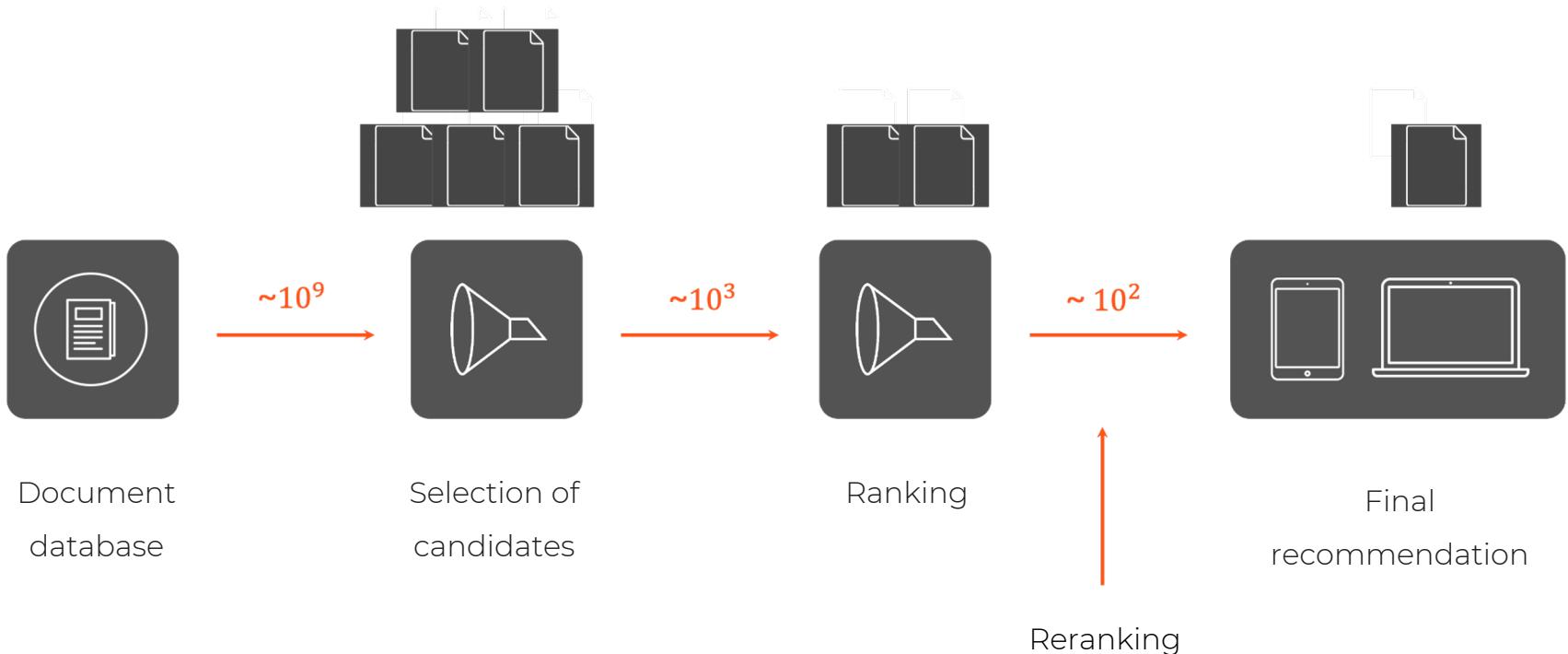
A typical simplification of a user's happiness is whether a recommendation is liked at the moment. Or even a bigger simplification —do you like the item as a whole or any item rating for the user

The user himself can also be represented in different ways

1. As an unordered set of items that the user interacted with
2. As an unordered set of items with ratings
3. As a sequence of items with ratings
4. As a sequence of interactions taking into account the context



General scheme





Course plan

In total, there are 6 lectures in the course:

1. Introduction, main concepts, formal statement
2. Matrix factorizations
3. SLIM, neural networks for recommendations
4. Metrics in RecSys. Learning to rank
5. RecSys in production
6. Case study



Lecture plan

1. Setting the task of recommendation systems

- Basic concepts
- Types of feedback

2. Collaborative filtering

3. Other types of recommendations

- Content models
- Hybrid models

4. Recommendation systems in practice

- Ranking model
- Selection of candidates
- Reranking

5. Properties of recommender systems

Collaborative filtering

**girafe
ai**

05



Example

Consider an example. Green cells are likes, red cells are dislikes.

User \ Item	1	2	3	4	5	6	7	8	9	10
Peter	Green	Red	Green	Red	Red	Red	Red	Green	Red	Red
Mary	Red	Green	Red	Red	Red	Red	Red	Red	Red	Red
Vasia	Red	Red	Red	Red	Red	Red	Green	Red	Red	Green
Kate	Green	Red	Red	Green	Red	Red	Red	Green	Red	Red

What should I recommend to Kate?

Peter and Kate have similar interactions, so we recommend 3

And what is better not to show to Peter?

It's 10 because it was disliked by Kate



Collaborative filtering

Idea

We want to recommend products to the user that users like him liked

You can act differently. Namely, to recommend items similar to the ones the user liked. Similarities between atoms can be identified by user interaction

Collaborative filtering

Generally it is a family of recommendation methods based on similarities in the history of interaction between users and products



User2User recommendations

- ★ Consider some measure of similarity $s(u, v)$ between users. Then we can consider as neighbors user u and users

$$N(u) = \{v \in U \setminus \{u\} \mid s(u, v) > \alpha\}$$

- ★ We want to offer the user u items that users like him liked. Let's evaluate the user's rating by the rating of neighbors.
- ★ After evaluating the ratings of unseen items, recommend a certain number of items with the maximum rating
- ★ Denote:

I_u — a lot of items that appreciated

\bar{r}_u — average user rating



User2User formulas

$$\hat{r}_{ui} = \frac{\sum_{v \in N(u)} s(u, v) r_{vi}}{\sum_{v \in N(u)} |s(u, v)|}$$

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N(u)} s(u, v) (r_{vi} - \bar{r}_v)}{\sum_{v \in N(u)} |s(u, v)|}$$

$$\hat{r}_{ui} = \bar{r}_u + \sigma_u \frac{\sum_{v \in N(u)} s(u, v) (r_{vi} - \bar{r}_v) / \sigma_v}{\sum_{v \in N(u)} |s(u, v)|}$$

$$\sigma_u = \sqrt{\frac{1}{|I_u|} \sum_{i \in I_u} (r_{ui} - \bar{r}_u)^2}$$



User2User recommendations

- Rating score as the average of the neighbors with weights
- Problem: different users put ratings on different scales, so we will make an adjustment for the average user rating
- In addition, estimates from different users may be on a different scale, so we can relate them



User2User similarity

How to choose $s(u, v)$?

- The normalized number of common goods, a measure of Jaccard
- Scalar product of vectors of general ratings
- Pearson correlation between vectors of general ratings
- Pearson correlation between vectors of general ratings adjusted for a small number of general ratings



Similarity formulas

How to choose $s(u, v)$?

$$s(u, v) = \sum_{i \in I_u \cap I_v} r_{ui} r_{vi}$$

$$s(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}}$$

$$s(u, v) = \frac{|I_u \cap I_v|}{|I_u \cup I_v|}$$

$$s(u, v) = \min\left(\frac{|I_u \cap I_v|}{50}, 1\right) \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}}$$



Item2Item recommendations

- Idea: to the products evaluated by the user, we will find the most similar to them and recommend
- The similarity between a item is defined as a cosine measure adjusted for the average user rating, where U_i — users who rated the a item i
- Further, using the similarities and the history of user interaction, we similarly evaluate the ratings of unappreciated items



Adjusted pearson correlation

$$s(i, j) = \frac{\sum_{u \in U_i \cap U_j} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in U_i \cap U_j} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{u \in U_i \cap U_j} (r_{uj} - \bar{r}_u)^2}}$$



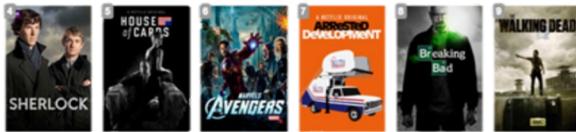
Features of collaborative filtering

- ★ It does not rely on additional information about users and products, collaborative filtering methods are self-sufficient
- ★ They are not applicable for cold users and products, that is, for those with a small history of interactions
- ★ Recommendations are based on the history of interactions in the past, which depends on past recommendations to the user
- ★ More advanced methods will be discussed in Lecture 2



Idea

User2User



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

Item2Item



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



Choosing between Item2Item and User2User

It strongly depends on the number of ratings per user and per item – the more, the more reliable the similarity.

This also leads to the possibility of pre-calculation: if there are a lot of estimates, then adding a pair of estimates will not change anything. Then you can update once in a while.

The similarities themselves can be used for other tasks (candidate generation, contextual recommendations).

<https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf>

Content-based recommendations

**girafe
ai**

06



Content-based recommendations

Let's go back to the example and look at the items themselves

User \ Item	Cat 1	Flowers 1	Cat 2	Car 1	Flowers 2	Cat 3	Flowers 3	Car 2	Cake
Peter	Green	Black	Green	Orange	Orange	Black	Green	Black	Black
Mary	Black	Green	Black	Orange	Black	Black	Black	Black	Black
Vasia	Orange	Black	Orange	Black	Orange	Black	Green	Black	Green
Kate	Green	Black	Black	Green	Orange	Black	Green	Black	Orange

What should I recommend to Peter?

Peter has a pronounced love for cats, you can show an unproven cat

And what to do with Kate?

Kate has only one interaction with cars, this is not enough,

so we show only cats with whom there were 2 likes



Content approach

they consist in recommendations to users of such items that are similar in content to the ones they liked



Example of content-based recommendations

- Let each item i correspond to an embedding e_i obtained from its content
- Then the user u can be recommended items as the proximity ρ between the vectors
As ρ can be
 - Scalar product
 - Cosine distance
 - Etc

Various methods will be discussed in Lecture 3



Hybrid models

Hybrid models are understood as models of recommendation systems that use both collaborative and content information to build user recommendations at the same time



Ranking model

1

Collaborative and content models have their advantages and disadvantages

2

When making recommendations, I want to use contextual information. For example, user preferences may vary on different days of the week

3

Information about the product would be useful: brand, price, etc.

In practice, as a rule, a ranking model is used, which combines various features



Ranking model

Signs of such a model can be

1

Predictions of basic
collaborative, content models

2

User attributes: gender, age,
characteristics of the
interaction history, etc.

3

Features of the object: price,
weight, genre, characteristics
of the history of interactions,
etc.

4

Contextual information: day of
the week, weather, location,
etc.



Ranking Model: Model approaches

Tasks for which such a model can be trained

1

Binary classification.
The model predicts the presence or absence of a target positive interaction

2

Regression. For example, the model predicts the duration of watching a video, which determines the positivity of the interaction

3

Ranking. There are pairwise and listwise approaches, in which the model learns to correctly arrange a set of objects for the user

As a rule, variations of gradient boosting are used as a model, since at the moment they are better able to cope with tasks on tabular data.

Selection of candidates

**girafe
ai**

07



Candidates selection

Problems:

- ★ In a real recommendation system, there can be a huge number of items, about hundreds of millions or billions
- ★ There are complex models that cannot be applied to millions of items
- ★ With each user request for new recommendations, it is necessary to conduct all the items through the entire system
- ★ Solution:
at the initial stage, select a relatively small number of candidates and apply the next steps only to them.
Selection of candidates is the first step in the pipeline of recommendations



Candidates selection

When selecting candidates, it is important not to spoil the completeness of suitable products, so that there is nothing to choose from at all.

Possible approaches:

- Heuristic: the most popular products, the most popular among residents of the same city, etc.
- Collaborative: we consider item2item or user2user to be similar
- Content similarities: looking for close embedding; can be quickly identified using data structures HNSW, FAISS, etc.
- Approaches that take into account business logic: fresh, new

Reranking

girafe
ai

08



Reranking

In addition, it is often necessary in practice to take into account different business logic in the recommended objects

For example:

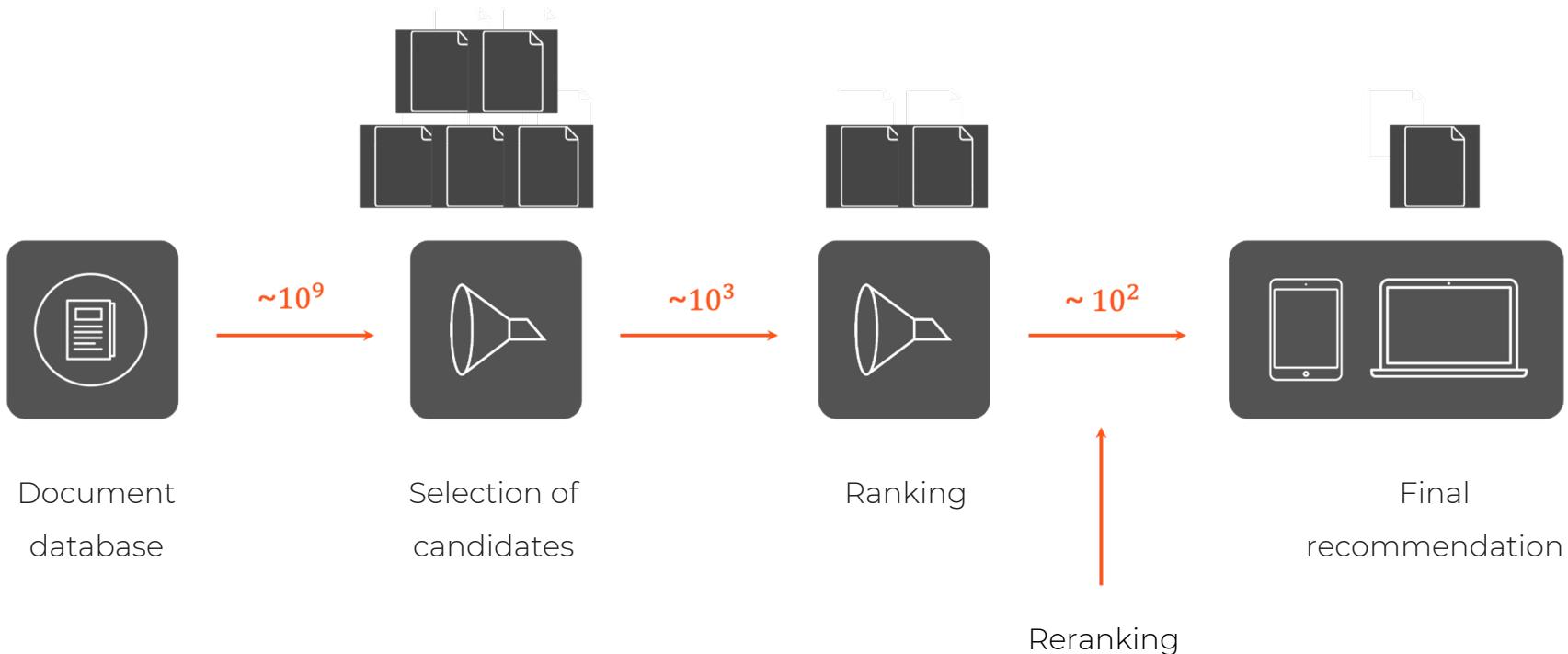
- Limit the number of old or too long videos
- Provide diversity

To do this, after applying the ranking model, a re-ranking mechanism is used.

The objects (top) arranged by the model are rearranged taking into account the required conditions



The final scheme



The cold start problem

girafe
ai

09



The cold start problem

1

Problem: how can I recommend something to a new user? How do I know who to show the new product to? How do I start recommending anything at all?

2

New user: try to find out as much information about him as possible, use technical approaches or extend registration, onboarding

3

New product:
Recommend using only content models

Feedback loop

girafe
ai

10



Feedback loop

- The recommendation system learns from a lot of examples that it recommended itself.
Because of this, we can get "stuck" in a local optimum, from which it is difficult to get out
- There are also risks that the recommendation system will adapt to a lot of users or products that are currently watching the most
- Simple solutions
 - Recommend fixed percent of random items to each user
 - Stratify training set by user groups or item groups

Thanks for attention!

Questions?

girafe
ai

