Building a recommender system from scratch

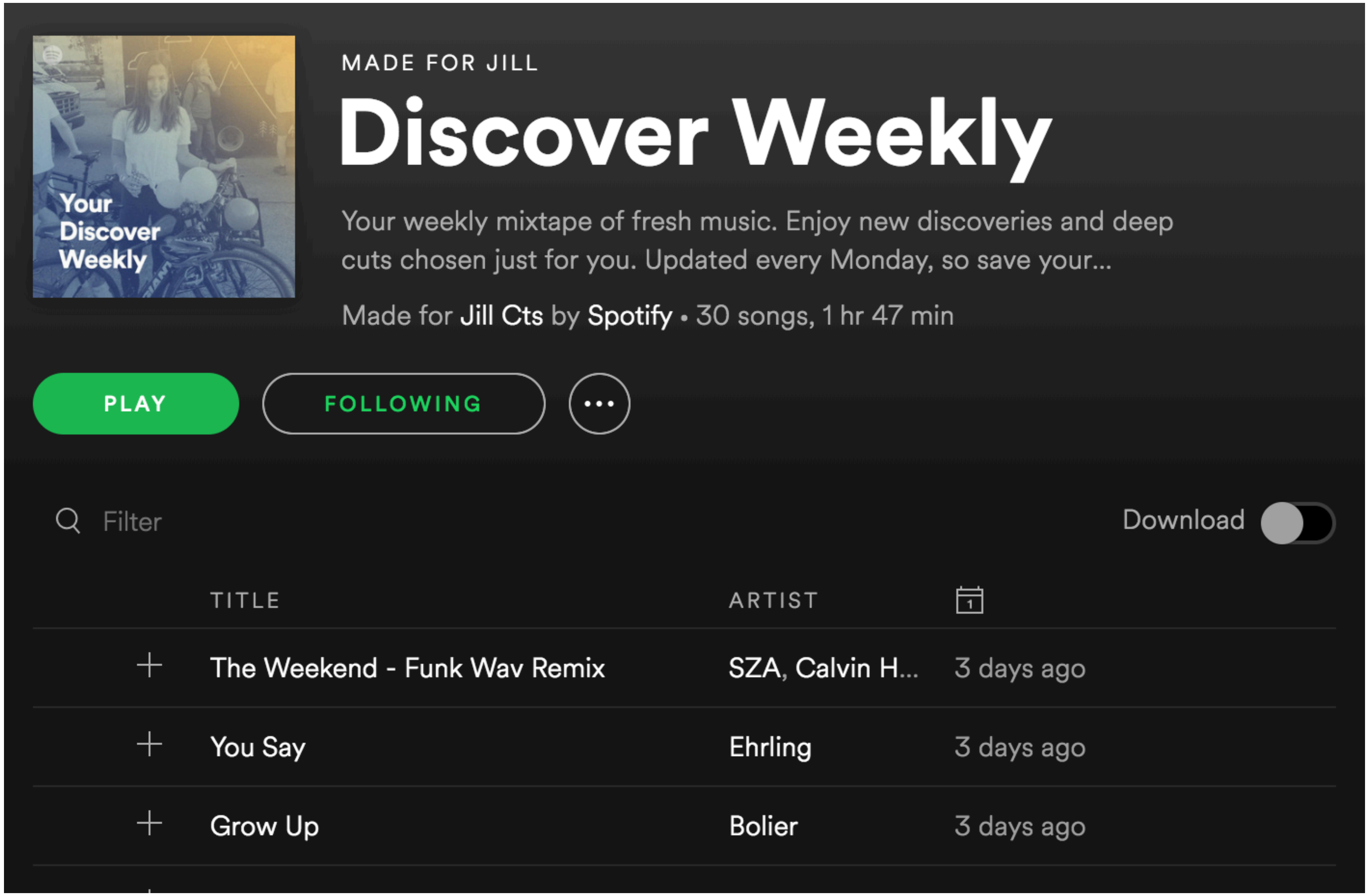
Objective

1. **Build a item-item recommender**
    - “Because you watched Movie X…”

2. **Build a top-N recommender**
    - “Your Top Recommendations”

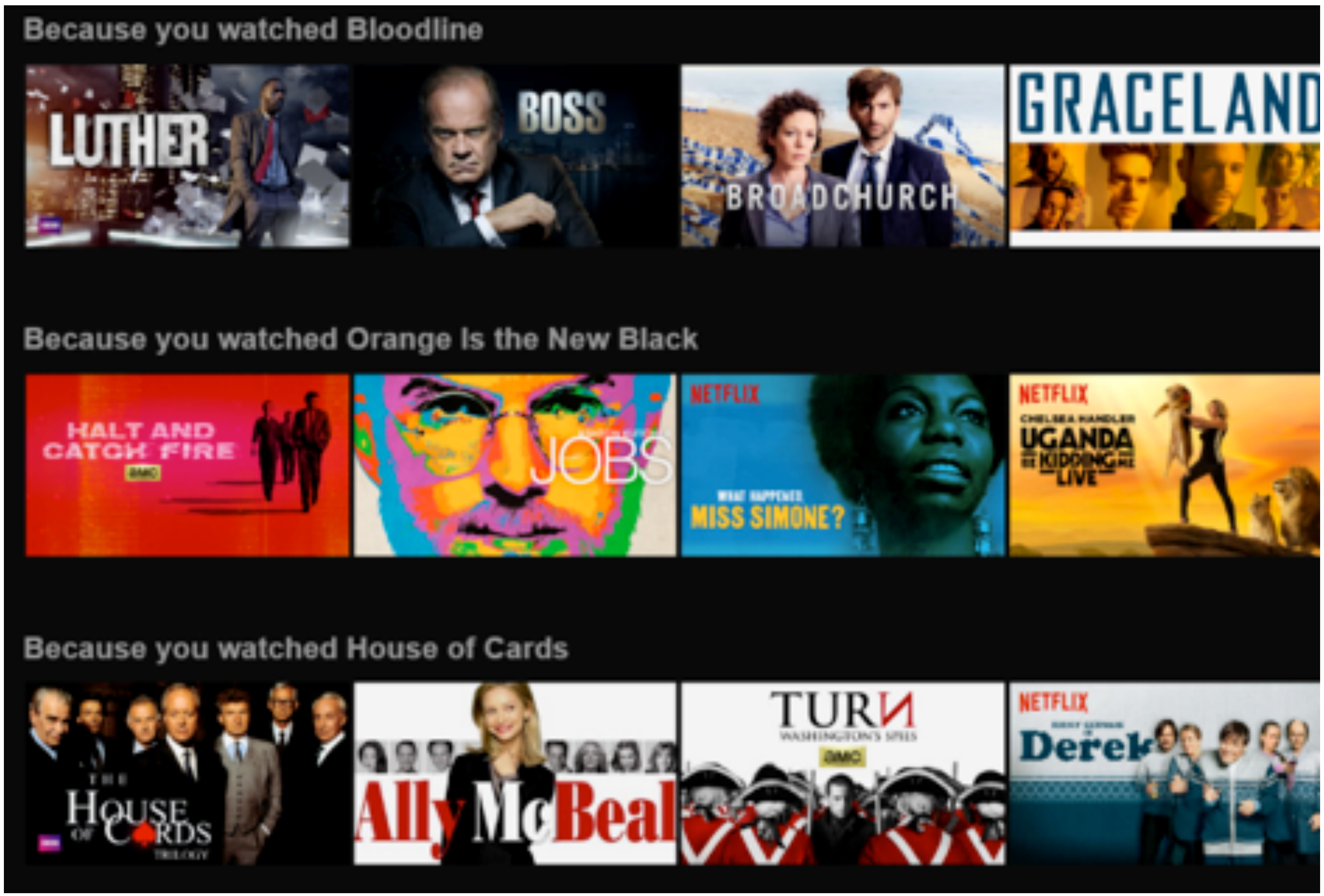
Spotify

Discover Weekly



Netflix

"Because you watched this TV show…"



Recommender Systems in the Wild

Spotify
Discover Weekly

Amazon
Customers who bought this item also bought

Netflix
Because you watched this show…

LinkedIn
Jobs recommended for you

OkCupid
Finding your best match

New York Times
Recommended Articles for You

GitHub
Repos "based on your interest"

Medicine
Facilitating clinical decision making

Before e-commerce

Things were sold exclusively in brick-and-mortar stores…

limited inventory

mainstream products

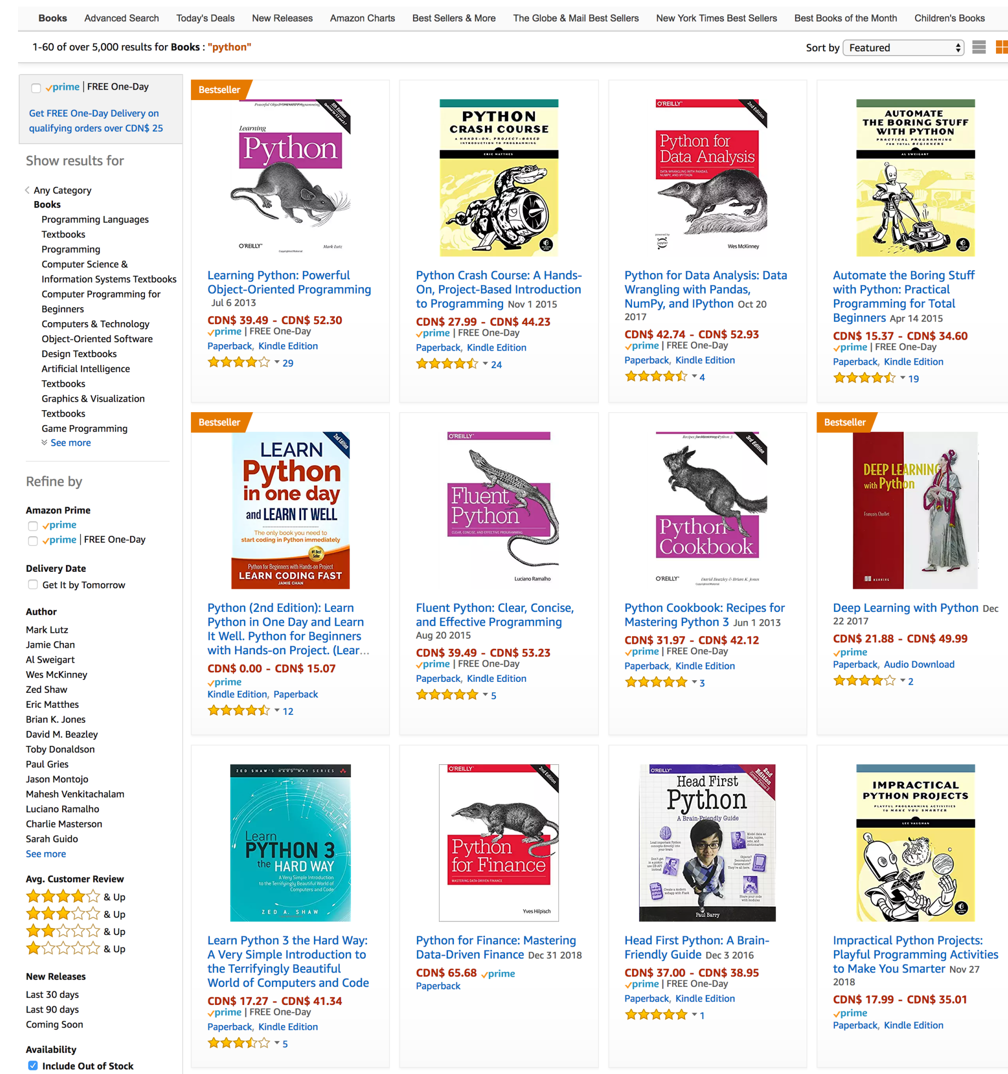
Before e-commerce

Things were sold exclusively in brick-and-mortar stores…

limited inventory

mainstream products

E-commerce



unlimited inventory

niche products

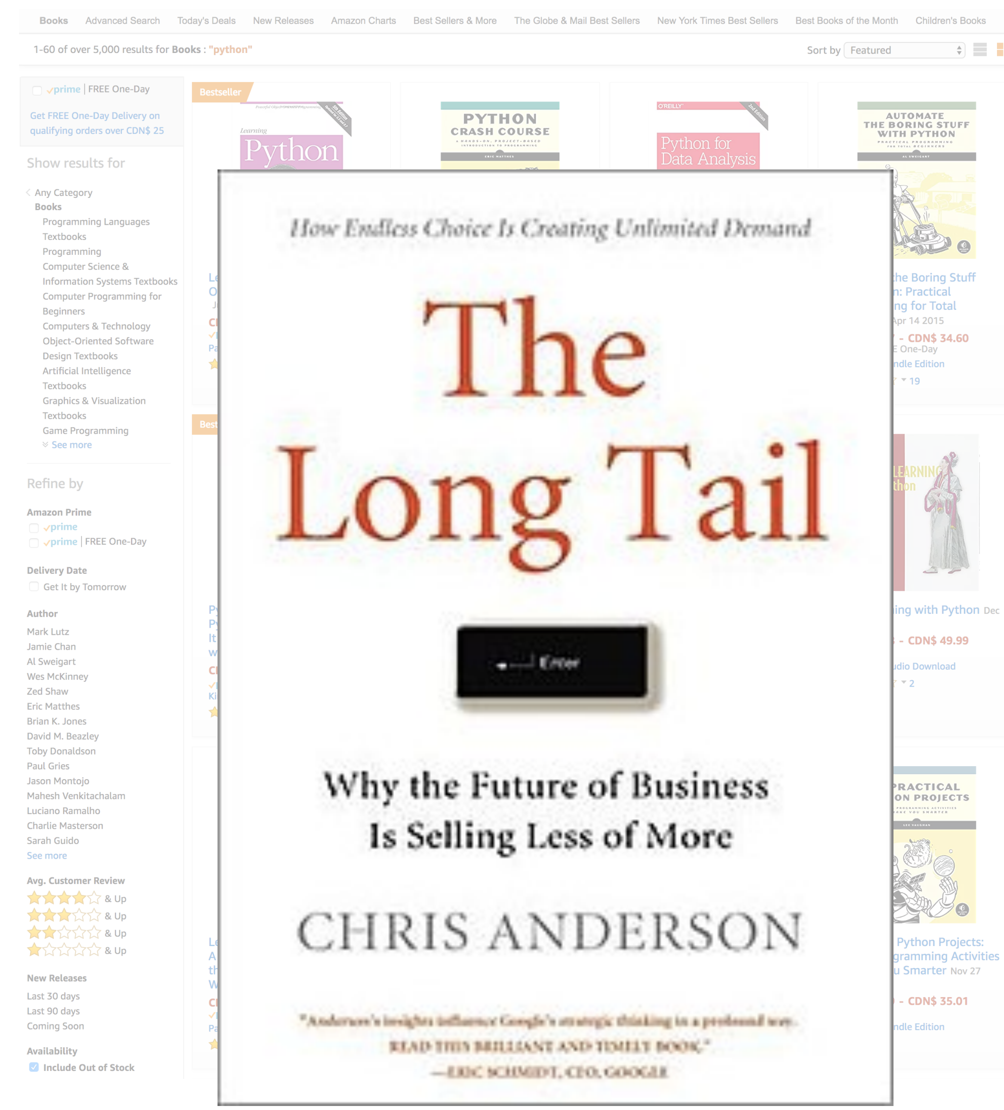
Before e-commerce

Things were sold exclusively in brick-and-mortar stores…

limited inventory

mainstream products

E-commerce



unlimited inventory

niche products

The Tasting Booth Experiment

When Choice is Demotivating: Can One Desire Too Much of a Good Thing?

Sheena S. Iyengar
Columbia University

Mark R. Lepper
Stanford University

6 jam samples

VS.

24 jam samples

The Tasting Booth Experiment

Initial Interest

6 jam samples

VS.

24 jam samples

40% of customers stopped at the limited-choice booth

60% of customers stopped at the extensive-choice booth

The Tasting Booth Experiment

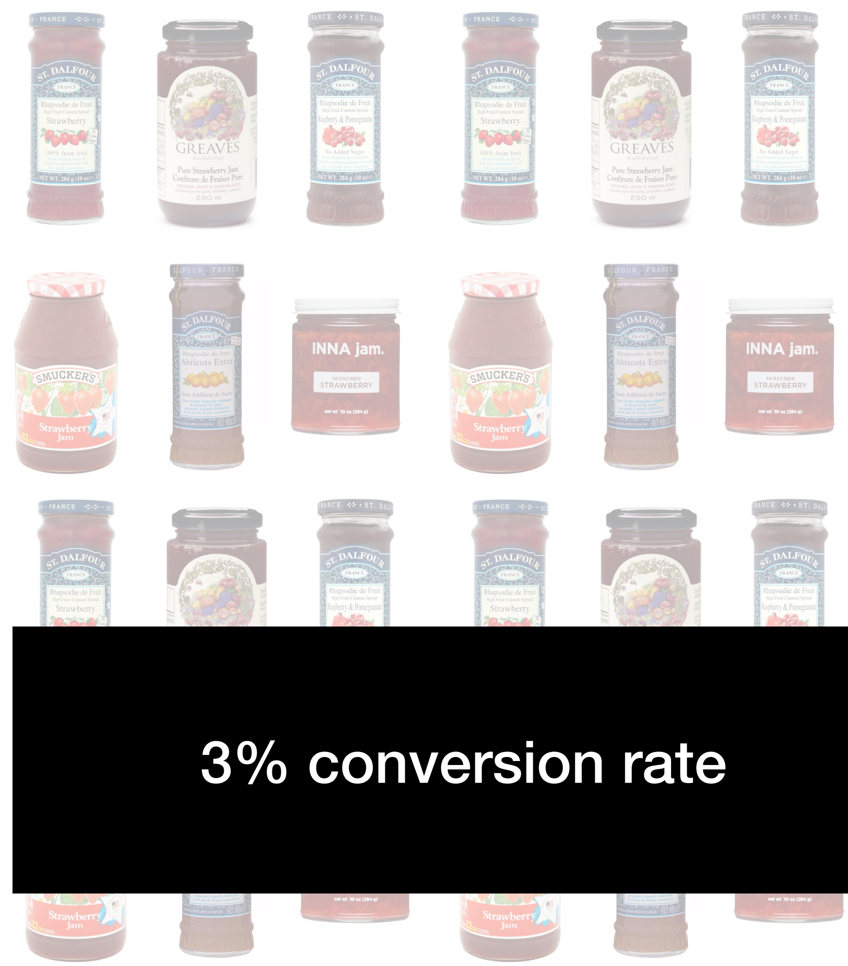
Subsequent Purchase

6 jam samples

30% conversion rate

vs.

24 jam samples



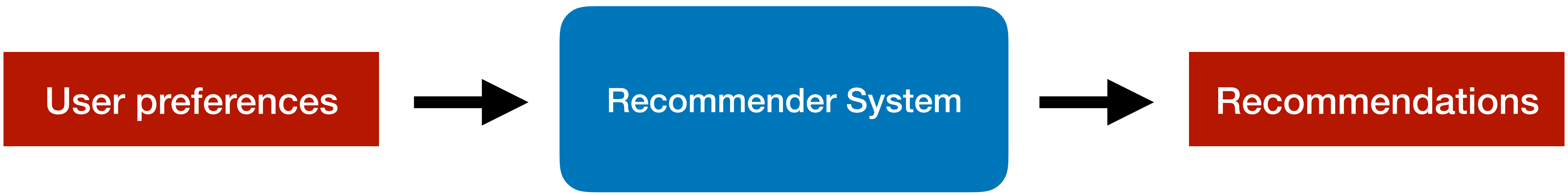
3% conversion rate

What is a recommender system?

An application of machine learning

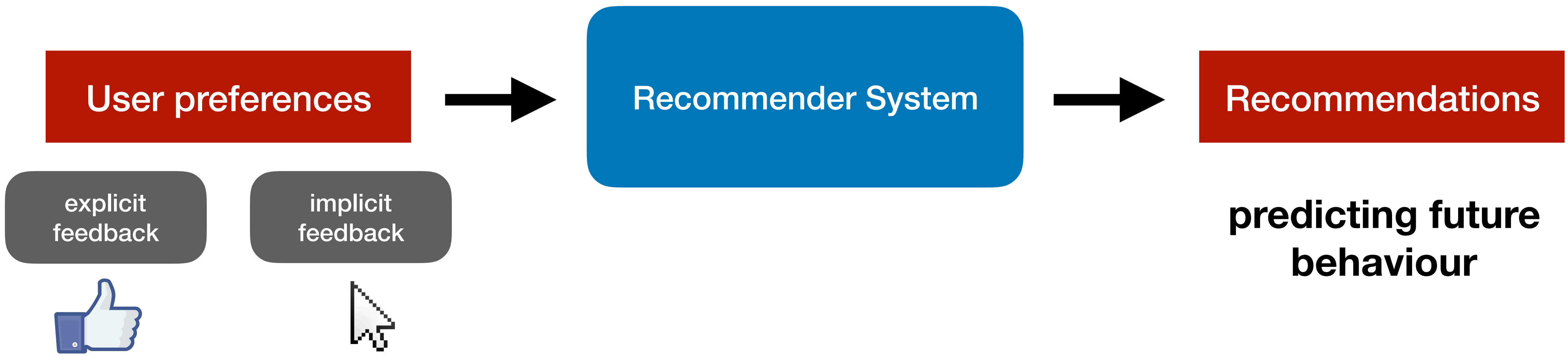
What is a recommender system?

An application of machine learning



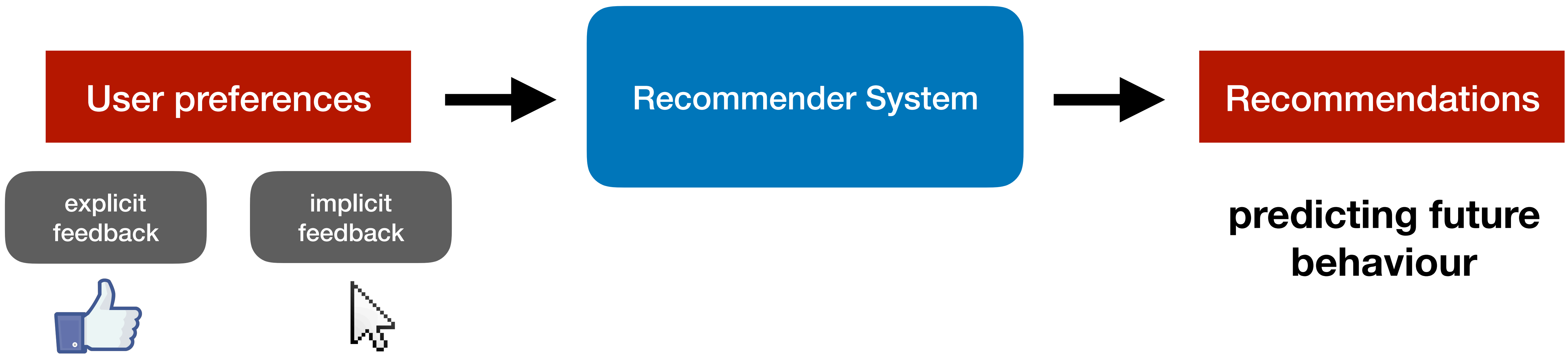
What is a recommender system?

An application of machine learning



What is a recommender system?

An application of machine learning



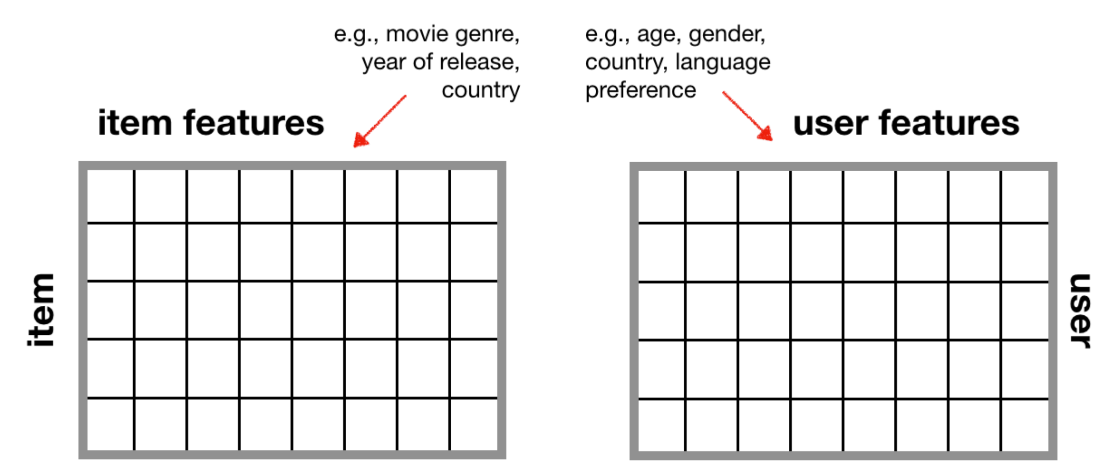
Collaborative filtering

user

John	1		3			5		5	
Erica			5		2		4		5
Anne	5	2		1		4		$\boxed{}$	2
Liz	3		4	3	4		5		
Jim	5	2		1		4		3	1

item

Content-based filtering



Collaborative Filtering

Similar people like similar things

users									
John	1		3			5		5	
Erica			5		2		4		5
Anne	5	2		1		4		$\boxed{3}$	2
Liz	3		4	3	4		5		
Jim	5	2		1		4		3	1

items

User-item (“utility”) matrix

Content-based Filtering

Looks at user and item features

users

	age	gender	country	lang	family?	horror?
John	24	M	CA	EN	N	Y
Erica	63	F	US	EN	N	Y
Anne	10	F	CA	FR	Y	N
Liz	38	F	IT	IT	Y	N
Jim	45	M	UK	EN	Y	Y

scary	funny	family	anime	drama	romance
N	N	Y	N	Y	Y
N	Y	N	N	Y	N
N	N	Y	N	N	Y
Y	N	Y	N	N	N
N	Y	Y	Y	N	N

items

- **User features:** age, gender, spoken language
- **Item features:** movie genre, year of release, cast

User Feedback

What are we populating these cells with?

user

John	1		3			5		5	
Erica			5		2		4		5
Anne	5	2		1		4			2
Liz	3		4	3	4		5		
Jim	5	2		1		4		3	1

item

Explicit feedback

Likert-scale rating (1-5)
Liked or not (boolean)

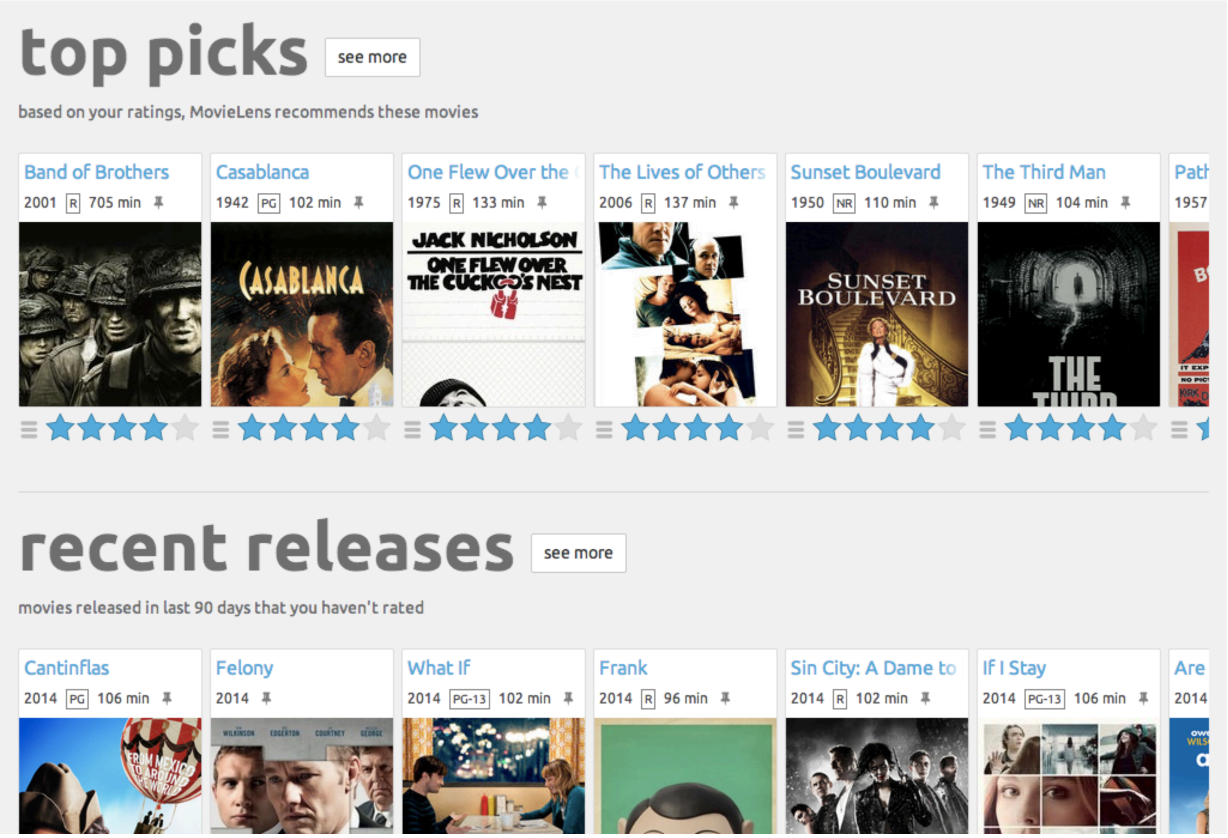
Implicit feedback

Browsing behaviour
Purchased? Read? Watched?

Developing a user feedback score

- Dwell time
- Recent vs. old interactions
- Negative implicit feedback
- What behaviour are you trying to drive?

MovieLens



- Created by GroupLens research group at the University of Minnesota
- Titanic dataset of recommenders

MovieLens

4.1.1. Dataset

In this experiment we used MovieLens 10 M dataset (approx. 10 M ratings from 71k users on 10 M movies on the 1–5 points scale obtained from real MovieLens recommender service), which is in recommender systems domain used usually (Baltrunas et al., 2010; Burke, 2000; Kagita et al., 2013; Lin et al., 2011). Experiment was performed with a sample of 20k users. The items features as genres, directors, keywords and actors were obtained from the Internet Movie Database (IMDb)[2] database by matching the movie name and year. In order to generate groups as real as possible, we generated groups at various levels of homogeneity. For this purpose the users' similarity was examined as the pairwise weighted cosine similarity between all users (users' user model were compared).

4. EXPERIMENTAL SETUP

In this section, we explain the experimental settings used for validating the *Clustered Tail* (*CT*) method, including an overview of the data used, selected variables, data mining methods, performance measurements and statistical tests.

Data. We used two popular datasets in our study MovieLens [5] and BookCrossing [6]. The MovieLens dataset contains 100,000 ratings on the scale of 1 to 5 from 943 customers on 1682 movies. The BookCrossing dataset contains 1,149,780 ratings on the scale of 1 to 10 from 278,858 customers on 271,379 books.

3. EXPERIMENTS

3.1 Methodology and Metrics

Applying our hybrid approach to the movie domain, we use the data set supplied by MovieLens Group [5] with 6040 users, 3952 films and over 1 million ratings. Ten percent of the users are randomly selected to be the test users, which follows the methodology of Breese, Heckerman and Kadie [3]. The others join the training data set. All profiles of training users are selected for the training data set. To test users, we randomly select twenty five percent of their profiles to be the test profiles. Applying this method for three times, we get three sets of training and testing data.

Abstract: Recent research has shown the significant vulnerabilities of collaborative recommender systems in the face of profile injection attacks, in which malicious users insert fake profiles into the rating database in order to bias the system's output. To reduce this risk, a number of approaches have been proposed to detect such attacks. Although the existing detection approaches can detect the standard type of these attacks effectively, they perform badly when detecting the recently proposed obfuscated type of these attacks, for example, average over popular items (AoP) attack. With this problem in mind, in this study the author propose a supervised approach to detect such attack. First, he uses the theory of term frequency inverse document frequency (TFIDF) to extract the features of AoP attack. Second, he uses the training set to train support vector machine (SVM) to generate a SVM-based classifier. Finally, he uses the generated classifier to detect the AoP attack. The experimental results on MovieLens dataset show that the proposed approach can detect AoP attack with high recall and precision.

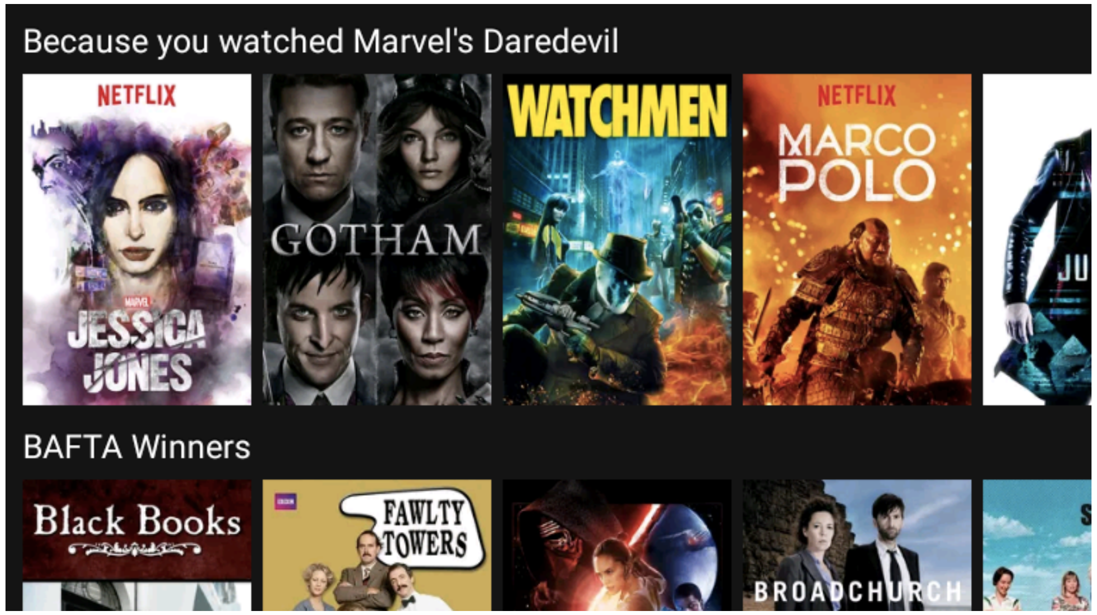
4. Empirical tests performed: experiment design

Due to the lack of any well-known data base for e-learning, publicly accessible for research and which contains information about the scores of the users, we used a known RS database from a field that is different from e-learning; in order to test our approach of CF adapted to e-learning we took the first five items of the MovieLens database [32] as five scores which have been evaluated by each user, in such a way that in Eq. (4) T has the value 5 and we are able to obtain the mean score for each user. Previously a 0 is inserted for those items that have not been rated, therefore indicating that the knowledge of a user in a test not performed is nil. The remainder of the items is used to discover the similarity between pairs of users.

In all the experiments carried out, for each item that each user has rated, the average value of the ratios given by their k-neighborhoods for that item has been calculated and the prediction has been compared with the value rated by the user (6) weighted with its estimated value (5), thus obtaining the calculation of the mean absolute error (MAE).

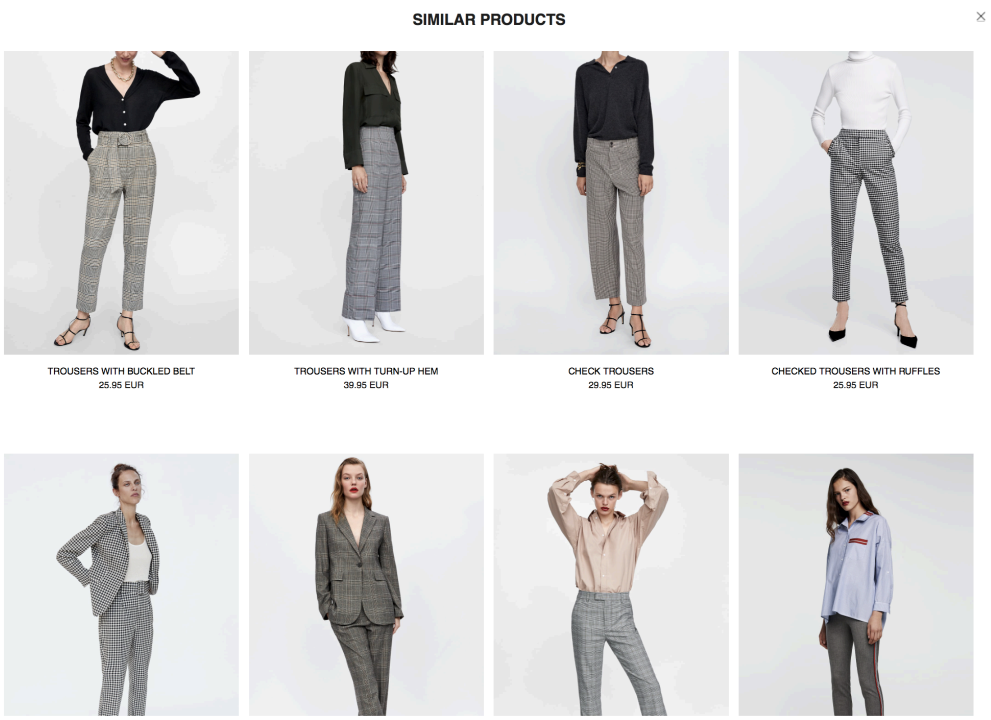
Item-Item Recommender

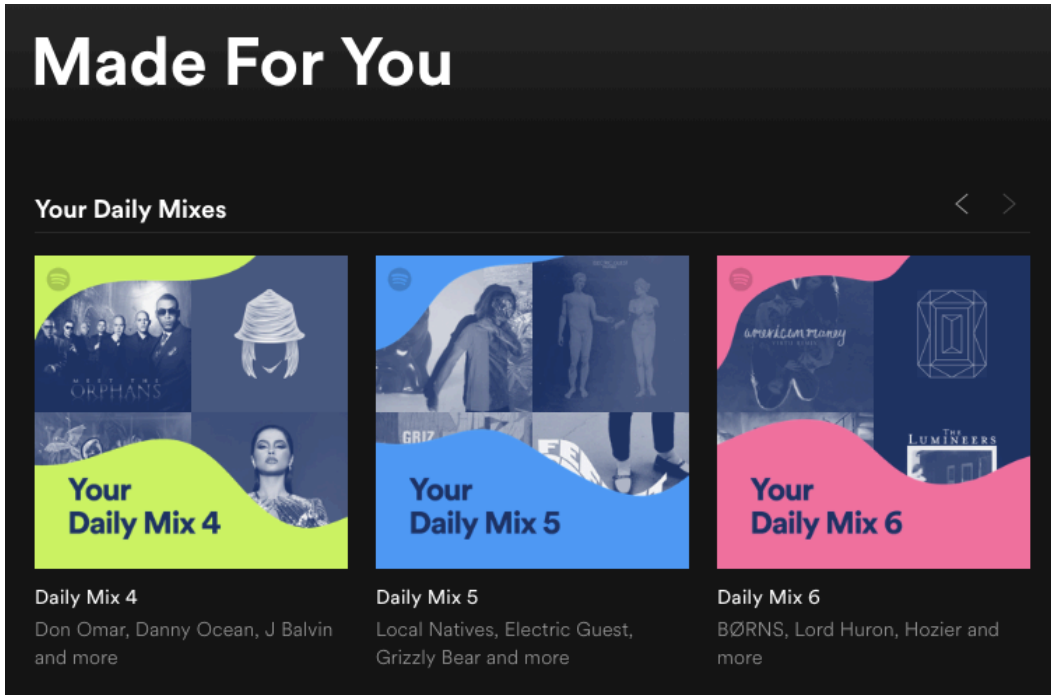
Examples


Because you watched Marvel's Daredevil
NETFLIX
JESSICA JONES
GOTHAM
WATCHMEN
NETFLIX
MARCO POLO
BAFTA Winners
Black Books
FAWLTY TOWERS
BROADCHURCH


Customers Who Bought This Item Also Bought
Predictive Analytics For Dummies
› Anasse Bari
29
Paperback
$17.72 Prime
Predictive Analytics: The Power to Predict Who...
› Eric Siegel
229
#1 Best Seller in Econometrics
Hardcover
$16.88 Prime
Quantifying the User Experience: Practical...
› Jeff Sauro
8
Paperback
$40.63 Prime
Marketing Analytics: Strategic Models and...
› Stephan Sorger
29
Paperback
$50.52 Prime


Related Coverage
Dec. 18, 2017
Hospital Giants Vie for Patients in Effort to Fend Off New Rivals
April 7, 2018
The Disappearing Doctor: How Mega-Mergers Are Changing the Business of Medical Care


SIMILAR PRODUCTS
TROUSERS WITH BUCKLED BELT
25.95 EUR
TROUSERS WITH TURN-UP HEM
39.95 EUR
CHECK TROUSERS
29.95 EUR
CHECKED TROUSERS WITH RUFFLES
25.95 EUR


Made For You
Your Daily Mixes
Your Daily Mix 4
Your Daily Mix 5
Your Daily Mix 6
Daily Mix 4
Don Omar, Danny Ocean, J Balvin and more
Daily Mix 5
Local Natives, Electric Guest, Grizzly Bear and more
Daily Mix 6
BØRNS, Lord Huron, Hozier and more

Pre-processing

user_id	movie_id	rating
2	439	4.0
10	368	4.5
14	114	5.0
19	371	1.0
2	371	3.0
19	114	4.5
3	439	3.5
54	421	2.0
32	114	3.0
10	369	1.0

items

users

		5.0			4.5		3.0
		3.0		4.0	2.5		3.0
2.0						1.0	
	3.0	3.5		5.0	4.5		
1.5	2.0			4.5		2.0	

Transform original data to user-item (utility) matrix

Mean Normalization

- Optimists → rate everything 4 or 5
- Pessimists → rate everything 1 or 2
- Need to normalize ratings by accounting for user and item bias
- Mean normalization
  - subtract b_i from each rating for given item i
  - subtract b_u from each rating for given user u

$$b_{ui} = \mu + b_i + b_u$$

user-item rating bias

global avg

item's avg rating

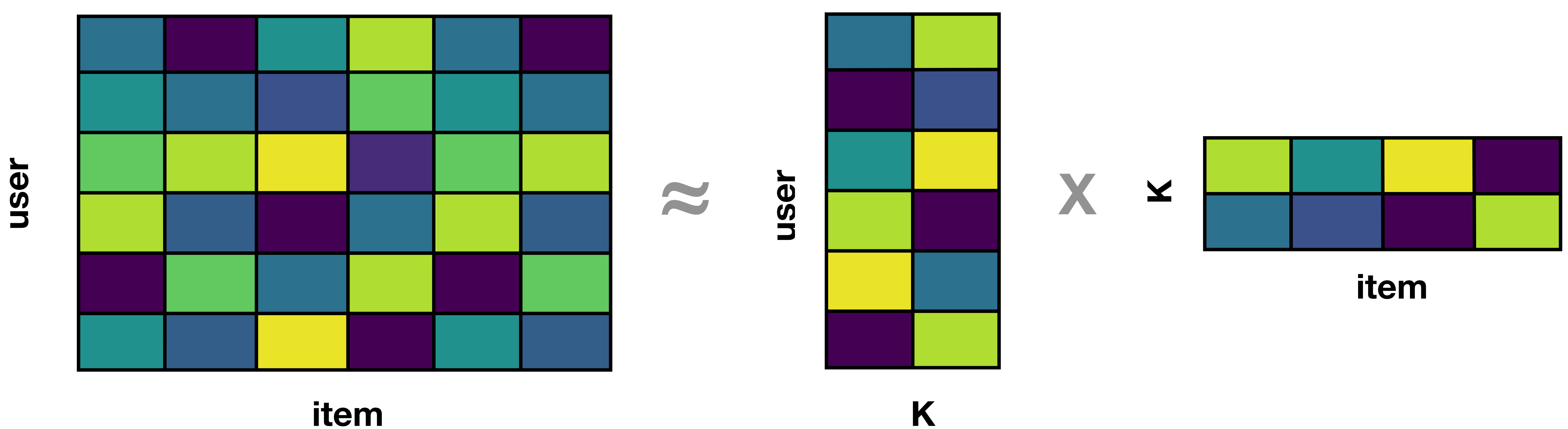
user's avg rating

Top N Recommender

Matrix Factorization

- Dimensionality reduction
- Factorize the user-item matrix to get 2 latent factor matrices:
  - User-factor matrix
  - Item-factor matrix
- Missing ratings are predicted from the inner product of these two factor matrices

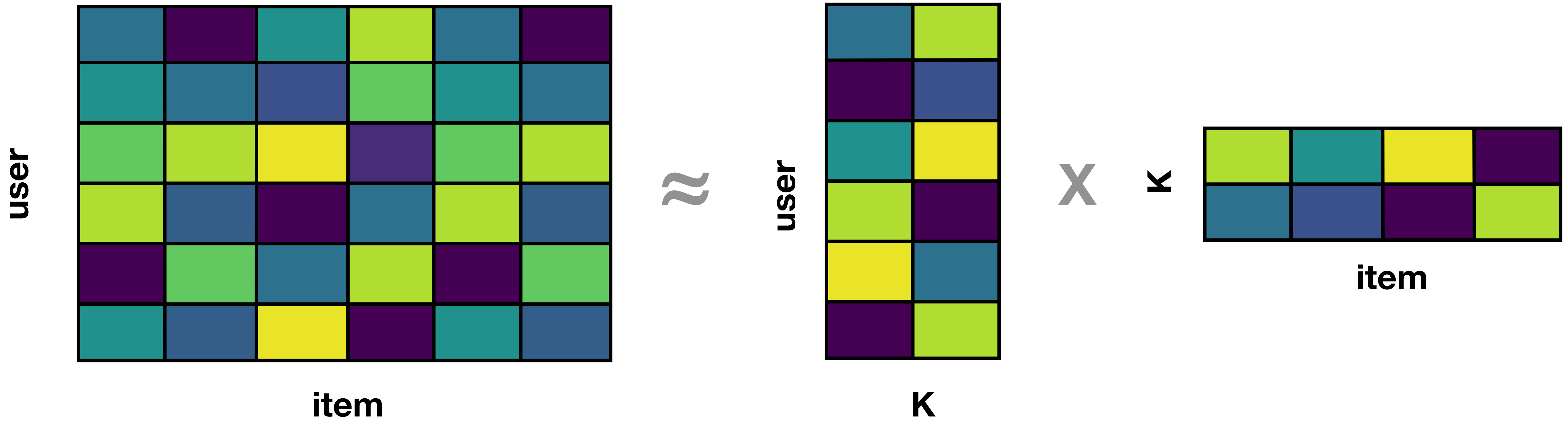
$$X_{mn} \approx P_{mk} \times Q_{nk}^{T} = \hat{X}$$



Matrix Factorization

- Algorithms that perform matrix factorization:
  - Alternating Least Squares (ALS)
  - Stochastic Gradient Descent (SGD)
  - Singular Value Decomposition (SVD)

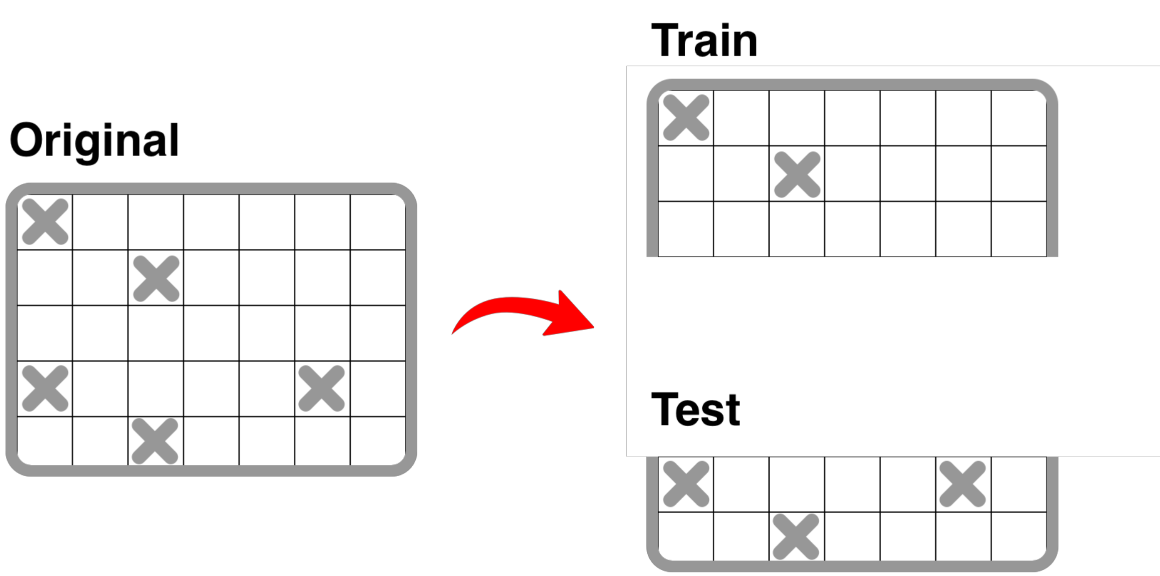
$$X_{mn} \approx P_{mk} \times Q_{nk}^T = \hat{X}$$



Evaluation

How do we evaluate recommendations?

Traditional ML



Recommendation Systems