سیستم های

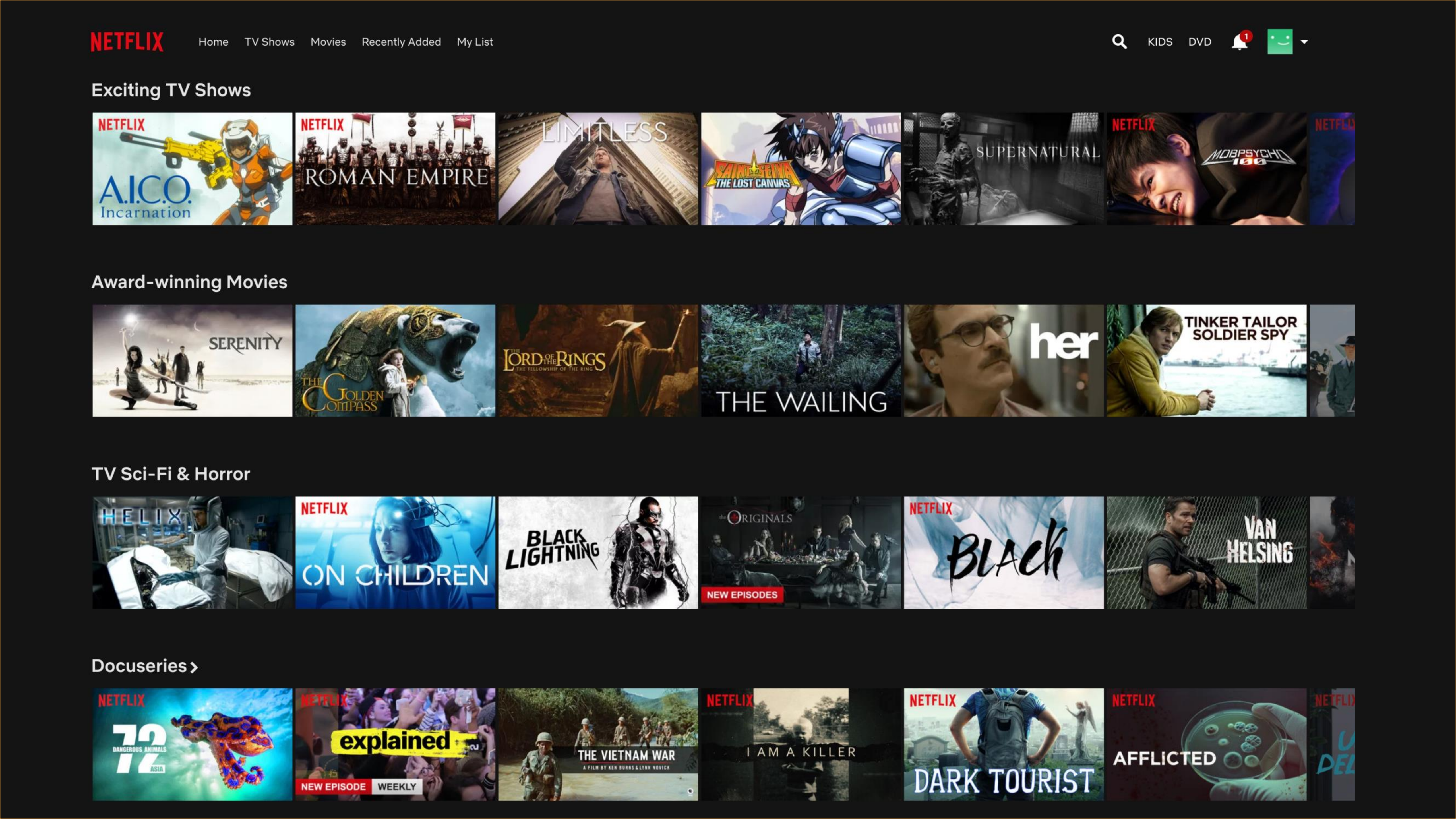
توصیه گر

محمد دهقانی

data-hub

معرفی

۱. مدیر دیتاهاب
۲. لیسانس نرم افزار از دانشگاه اصفهان و ارشدIT تربیت مدرس
۳. کارشناس سابق پردازش متن شرکت لایف وب
۴. سابقه همکاری با شرکت های داده پردازی آرون و توانمند
۵. سابقه تدریس در مرکز علوم شناختی(IPM) و دانشگاه های شریف، اصفهان، امیرکبیر، شهرکرد، علوم پزشکی تهران و کنفرانس وب پژوهی
۶. دارای بیش از ۳ مقاله ژورنالی
۷. مترجم کتاب یادگیری ماشین
۸. نویسنده کتاب تحلیل عواطف با استفاده از تکنیک های یادگیری ماشین


NETFLIX
Home
TV Shows
Movies
Recently Added
My List
KIDS
DVD
1
Exciting TV Shows
NETFLIX
A.I.C.O. Incarnation
NETFLIX
ROMAN EMPIRE
LIMITLESS
SAINT SEIYA THE LOST CANVAS
SUPERNATURAL
NETFLIX
MOBPSYCHO 100
Award-winning Movies
SERENITY
THE GOLDEN COMPASS
THE LORD OF THE RINGS
THE FELLOWSHIP OF THE RING
THE WAILING
her
TINKER TAILOR SOLDIER SPY
TV Sci-Fi & Horror
HELIX
NETFLIX
ON CHILDREN
BLACK LIGHTNING
the ORIGINALS
NEW EPISODES
NETFLIX
BLACK
VAN HELSING
Docuseries
NETFLIX
72 DANGEROUS ANIMALS ASIA
NETFLIX
explained
NEW EPISODE
WEEKLY
THE VIETNAM WAR
A FILM BY KEN BURNS & LYNN NOVICK
NETFLIX
I AM A KILLER
NETFLIX
DARK TOURIST
NETFLIX
AFFLICTED

35% of the purchases on Amazon are the result of their recommender system, according to McKinsey.

Recommendations are responsible for 70% of the time people spend watching videos on YouTube.

75% of what people are watching on Netflix comes from recommendations, according to McKinsey.

Recommendation Engine – Examples

Facebook–"People You May Know"

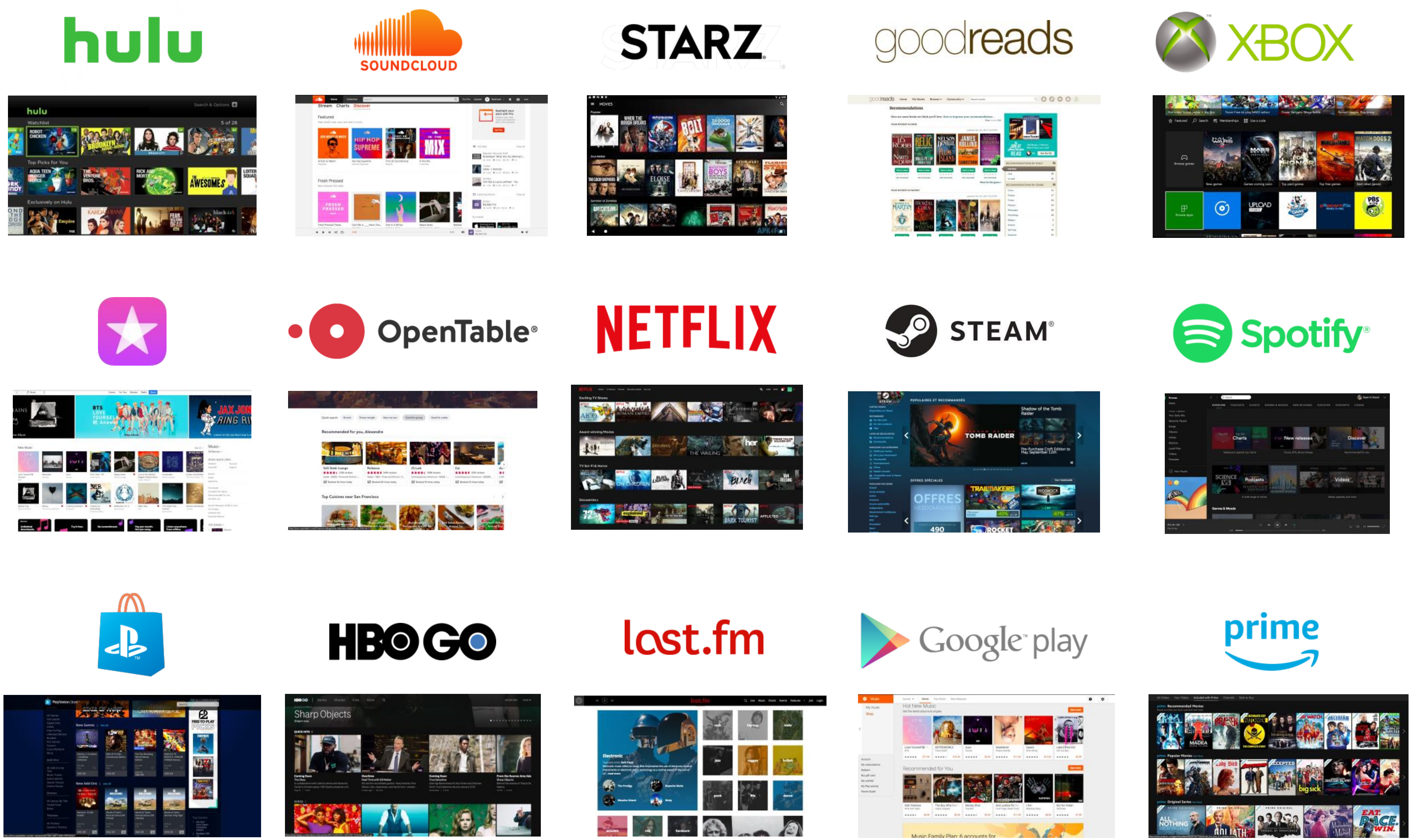
YouTube–"Recommended Videos"

Netflix–"Other Movies You May Enjoy"

Amazon–"Customer who bought this item also bought …"

LinkedIn–"Jobs You May Be Interested In"

Pinterest–"Recommended Images"


hulu
SOUNDCLOUD
STARZ
goodreads
XBOX
OpenTable
NETFLIX
STEAM
Spotify
HBO GO
last.fm
Google play
prime

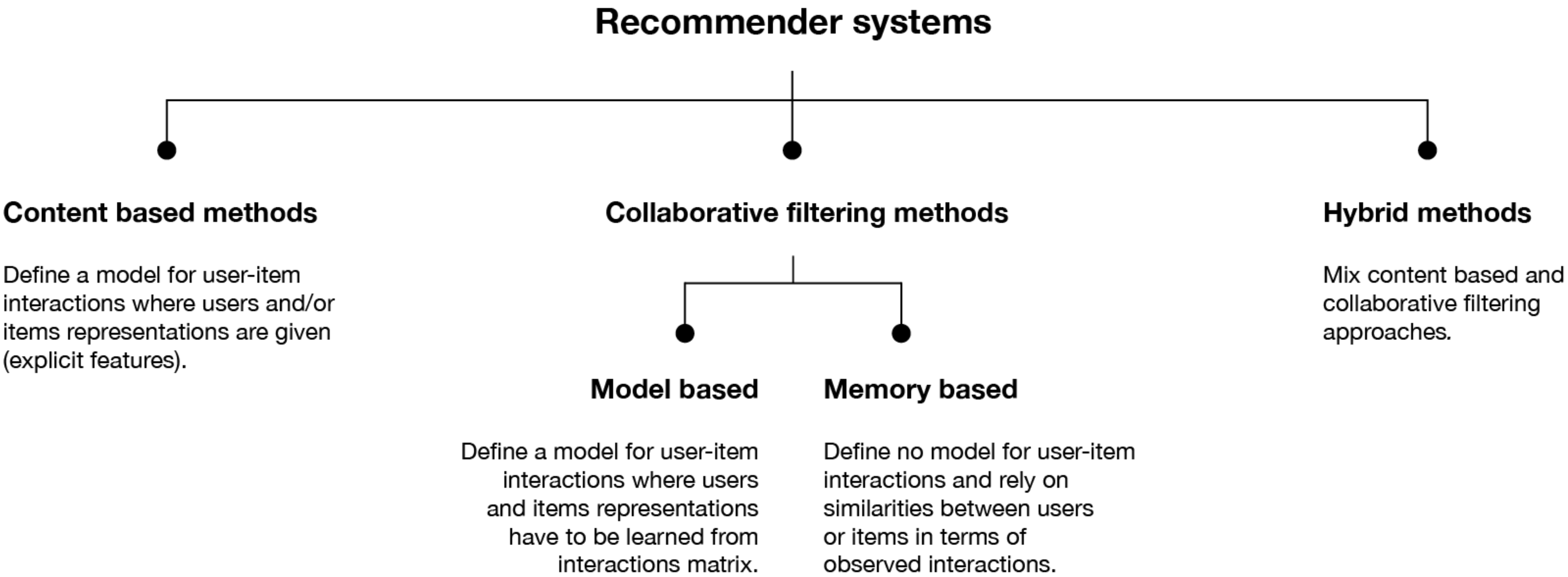
Plan for Today

1. Collaborative Filtering

2. Content-Based

3. In Production

Category


Recommender systems
Content based methods
Define a model for user-item interactions where users and/or items representations are given (explicit features).
Collaborative filtering methods
Model based
Define a model for user-item interactions where users and items representations have to be learned from interactions matrix.
Memory based
Define no model for user-item interactions and rely on similarities between users or items in terms of observed interactions.
Hybrid methods
Mix content based and collaborative filtering approaches.

Most popular items

Most popular items

- This is a blazing fast and dirty approach.
- The things is, there is no personalization involved with this approach.
- Surprisingly, such approach still works in places like news portals.

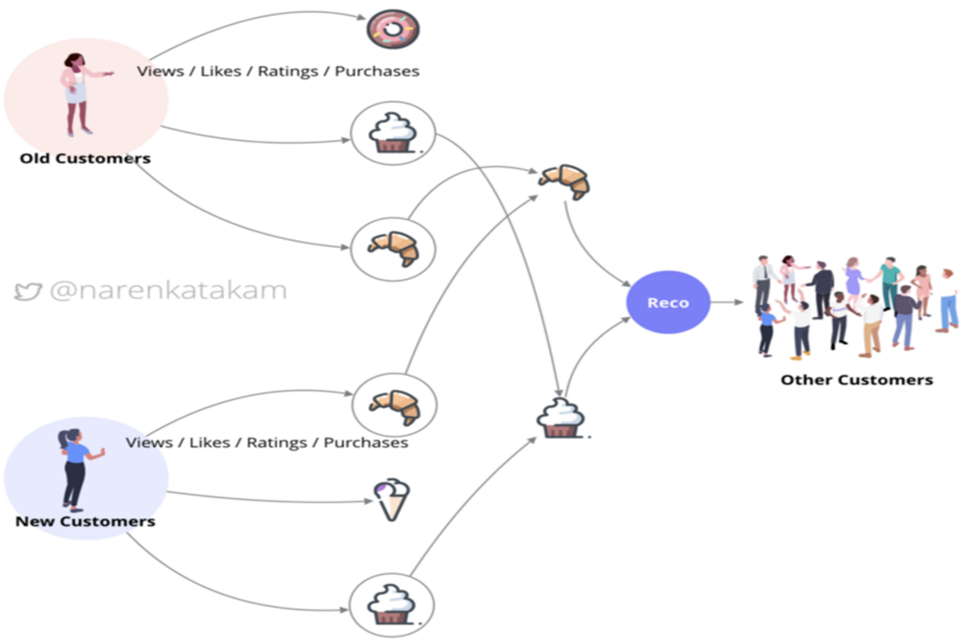


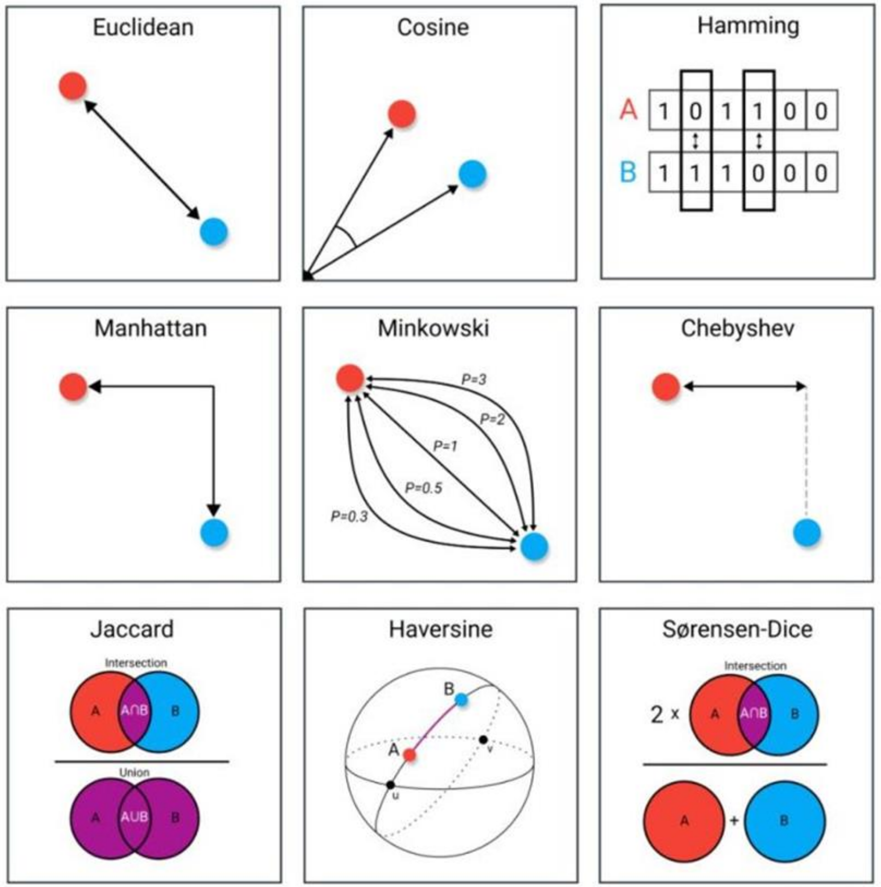
Fig.1: Popularity Filtering Model

Collaborative Filtering

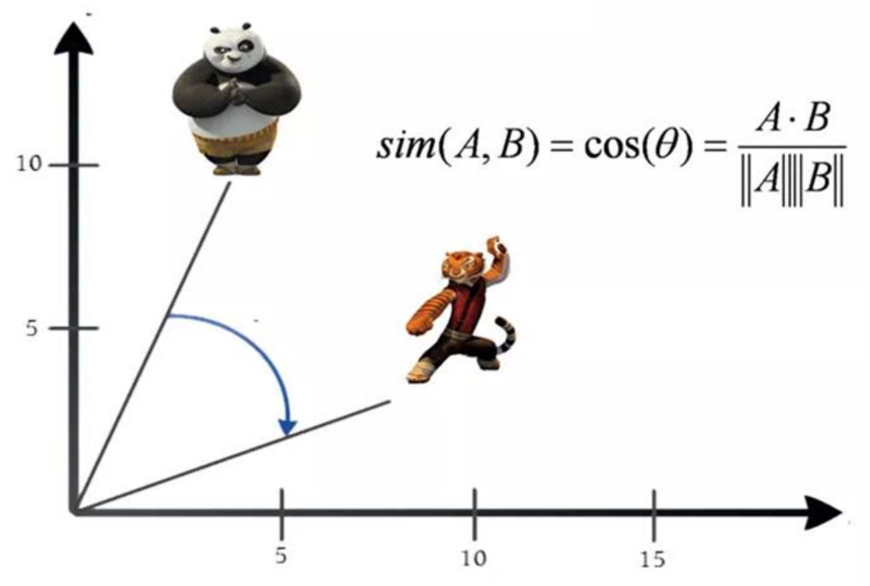
1. Collaborative Filtering – **Similarity Function**

Real function that quantify the similarity between two objects.

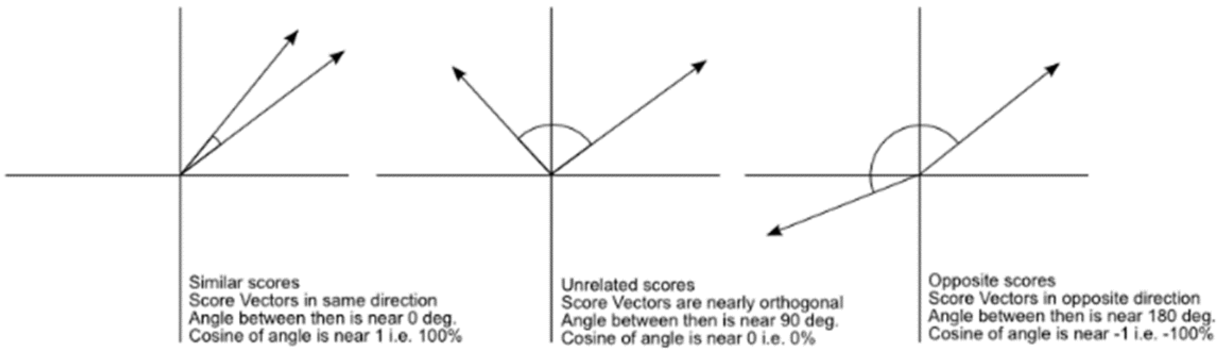
$\text{sim}(a, b) = \dots$



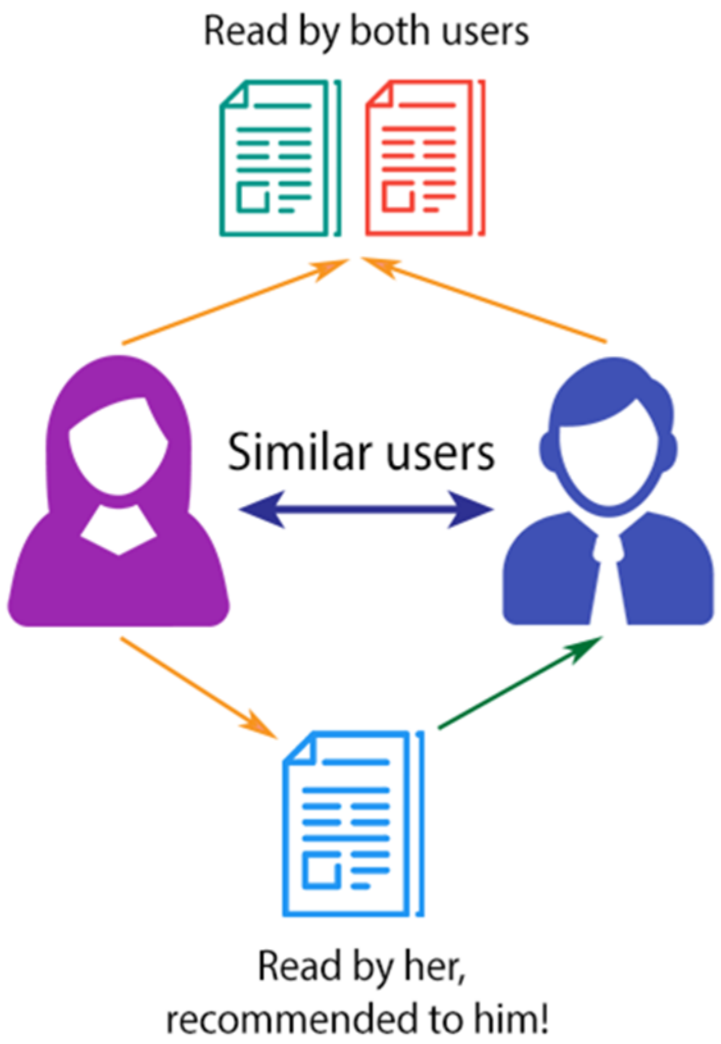
1. Collaborative Filtering – **Similarity Function**



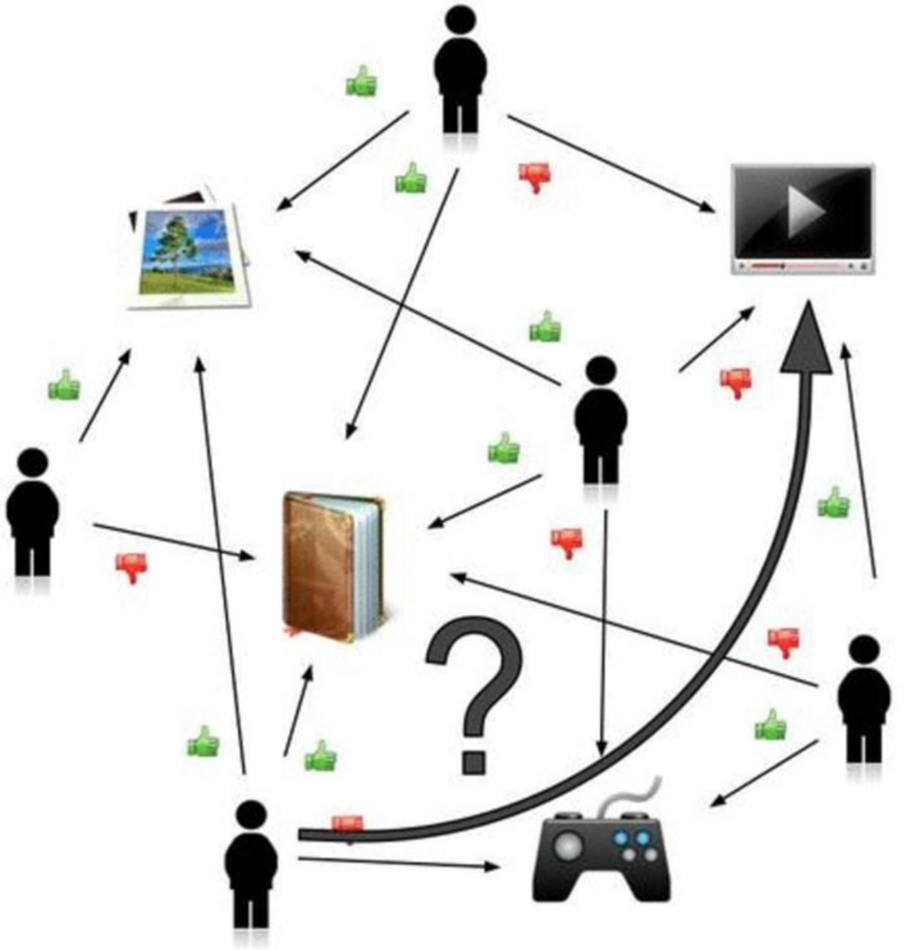
$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

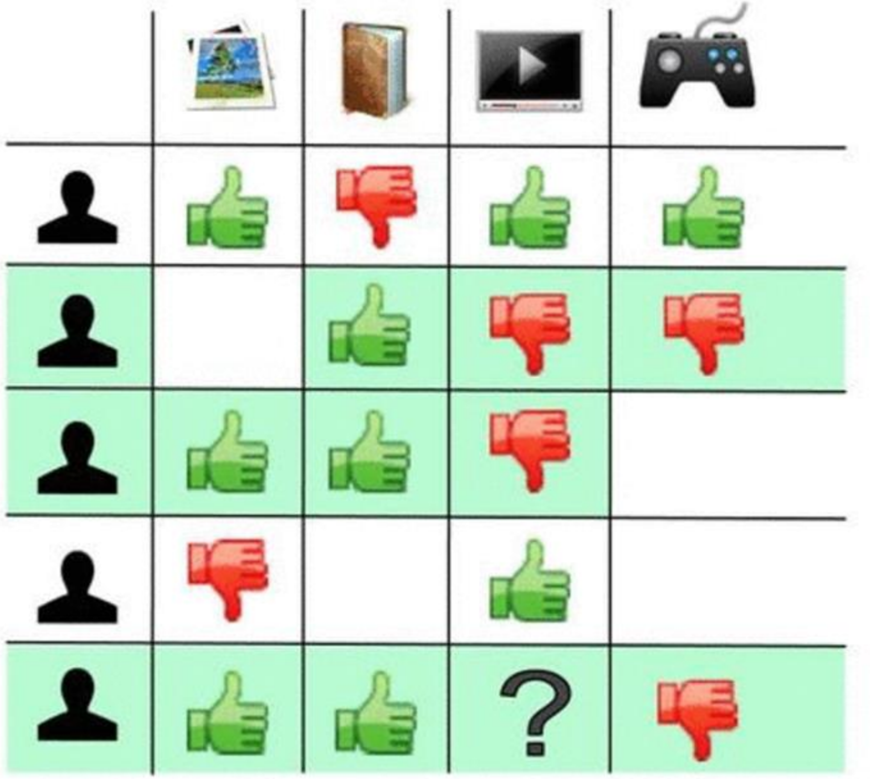


1. Collaborative Filtering



1. Collaborative Filtering





https://en.wikipedia.org/wiki/Collaborative_filtering

1. Collaborative Filtering – **Rating Matrix**

Rating data:

$$D_{(\,\cdot\,,\,\cdot\,)} = 4 \quad \in \{\varnothing, 1, 2, 3, 4, 5\}$$

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

1. Collaborative Filtering – **User-User**

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

	0	1	2	3	4	5
0	1.000000	0.396780	0.530558	0.440867	0.255551	0.264520
1	0.396780	1.000000	0.573068	0.095238	0.621059	0.476190
2	0.530558	0.573068	1.000000	0.000000	0.830455	0.000000
3	0.440867	0.095238	0.000000	1.000000	0.276026	0.238095
4	0.255551	0.621059	0.830455	0.276026	1.000000	0.000000
5	0.264520	0.476190	0.000000	0.238095	0.000000	1.000000

1. Collaborative Filtering – **User-User**

2		2	4	5	
5		4			1
		5		2	
	1		5		4
		4			2
4	5	2.5	1		

Rating prediction: $\hat{y}_{u,i} = \sum \text{sim}(u',u) r_{u'i}$

5 0.264520 0.476190 0.000000 0.238095 0.000000 1.000000

0.264520*2 + 0.476190*4 + 0*5 + 0.238095*0 + 0*4 + 1*0

2.43 ≈ 2.5

1. Collaborative Filtering – User-User **Benefits**

- "People who bought that also bought that"

- Good when #items >> #users

1. Collaborative Filtering – User-User **Challenges**

- Sparsity
- Don’t scale – Nearest Neighbors requires computation that grows with the number of users and items
- Model Too Simplistic – Accuracy of recommendation may be poor

1. Collaborative Filtering – **Item-Item**

2		2	4	5	
5		4			1
		5		2	
2.6	1	2.15	5		4
		4			2
4	5		1		

0.58*1 + 0.28*5 + 0.16*4
2.62 ≈ 2.6

0*1 + 0.16*5 + 0.34*4
2.16 ≈ 2.15

1.00	0.58	0.46	0.28	0.28	0.16
0.58	1.00	0.00	0.30	0.00	0.17
0.46	0.00	1.00	0.16	0.48	0.34
0.28	0.30	0.16	1.00	0.57	0.67
0.28	0.00	0.48	0.57	1.00	0.00
0.16	0.17	0.34	0.67	0.00	1.00

What about mean? 2.62/5

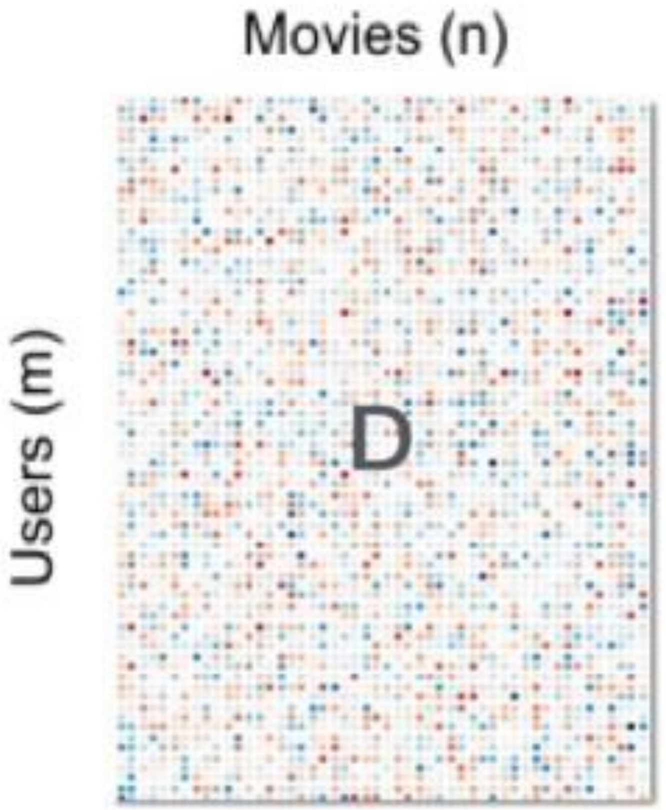
1. Collaborative Filtering – Item-Item **Benefits**

- “If you like this you might also like that”
- Good when #users >> #items
- Very fast after the item-item table has been pre-computed

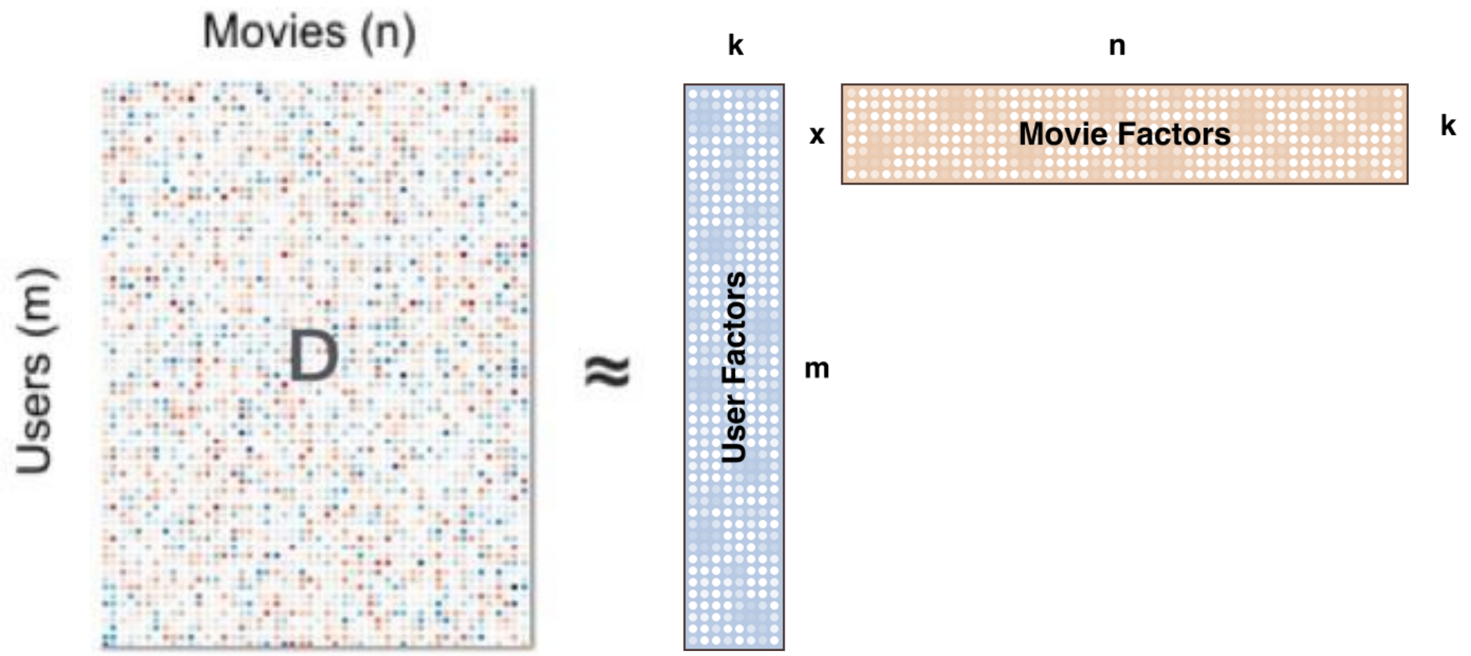
1. Collaborative Filtering – Item-Item **Challenges**

- Bottleneck – similarity computation
- Space complexity – dense item-item similarity matrix
- Model Too Simplistic – Accuracy of recommendation may be poor

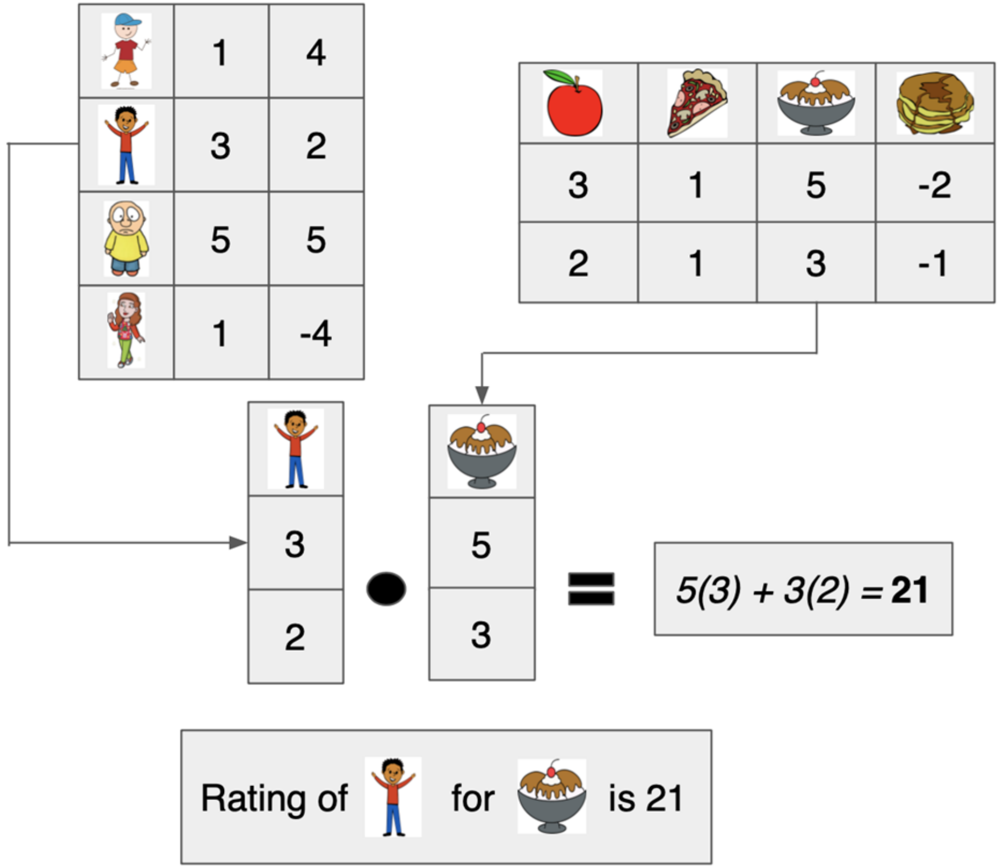
1. Collaborative Filtering – **User-Item**



1. Collaborative Filtering – **User-Item**



1. Collaborative Filtering – **User-Item**



1. Collaborative Filtering – **User-Item**

U*I = U*d   d*I

Item

User	W	X	Y	Z
A		4.5	2.0	
B	4.0		3.5	
C		5.0		2.0
D		3.5	4.0	1.0

Rating Matrix

=

A	*1.2*	*0.8*
B	*1.4*	*0.9*
C	*1.5*	*1.0*
D	*1.2*	*0.8*

4*2

User Matrix

X

W	X	Y	Z
1.5	*1.2*	*1.0*	*0.8*
1.7	*0.6*	*1.1*	*0.4*

2*4

Item Matrix

we’ll feed our model with the vector [1.2, 0.8].[1.2,0.6] and force its output to equal 4.5

1. Collaborative Filtering – **Matrix Factorization**

- SGD – Stochastic Gradient Descent
- SVD – Truncated Singular Value Decomposition
- ALS – Alternating Least Square

1. Collaborative Filtering – User-Item **Benefits**

similar

- Fast after U and I are pre-computed
- Can learn more about users with U
- Can learn more about items with I

1. Collaborative Filtering – User-Item **Challenges**

- Sparsity
- Need to re-learn everything every time a new user or new item or new rating enter the game

1. Collaborative Filtering – Sparsity Example, the **Netflix Prize**

- 17,770 Movies

- 480,189 Users

- 100,480,507 Ratings How

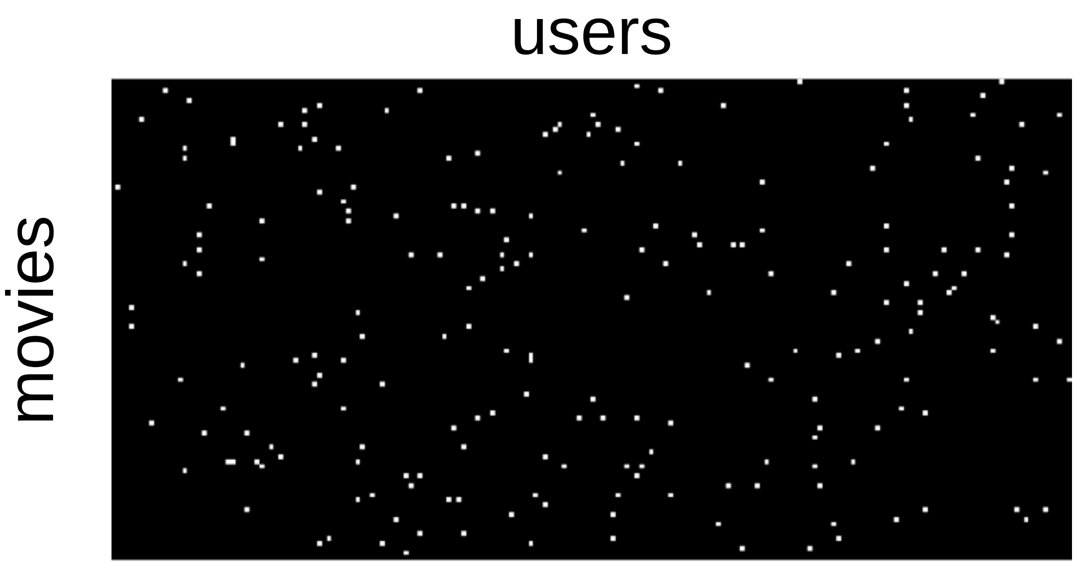
dense is our Matrix?

$$\frac{Ratings}{Movies \times Users} = \frac{100,480,507}{17,770 \times 480,189} \times 100 = 1.18\%$$

1. Collaborative Filtering – Sparsity Example, the **Netflix Prize**

- 17,770 Movies
- 480,189 Users
- 100,480,507 Ratings

How dense is our Matrix ?



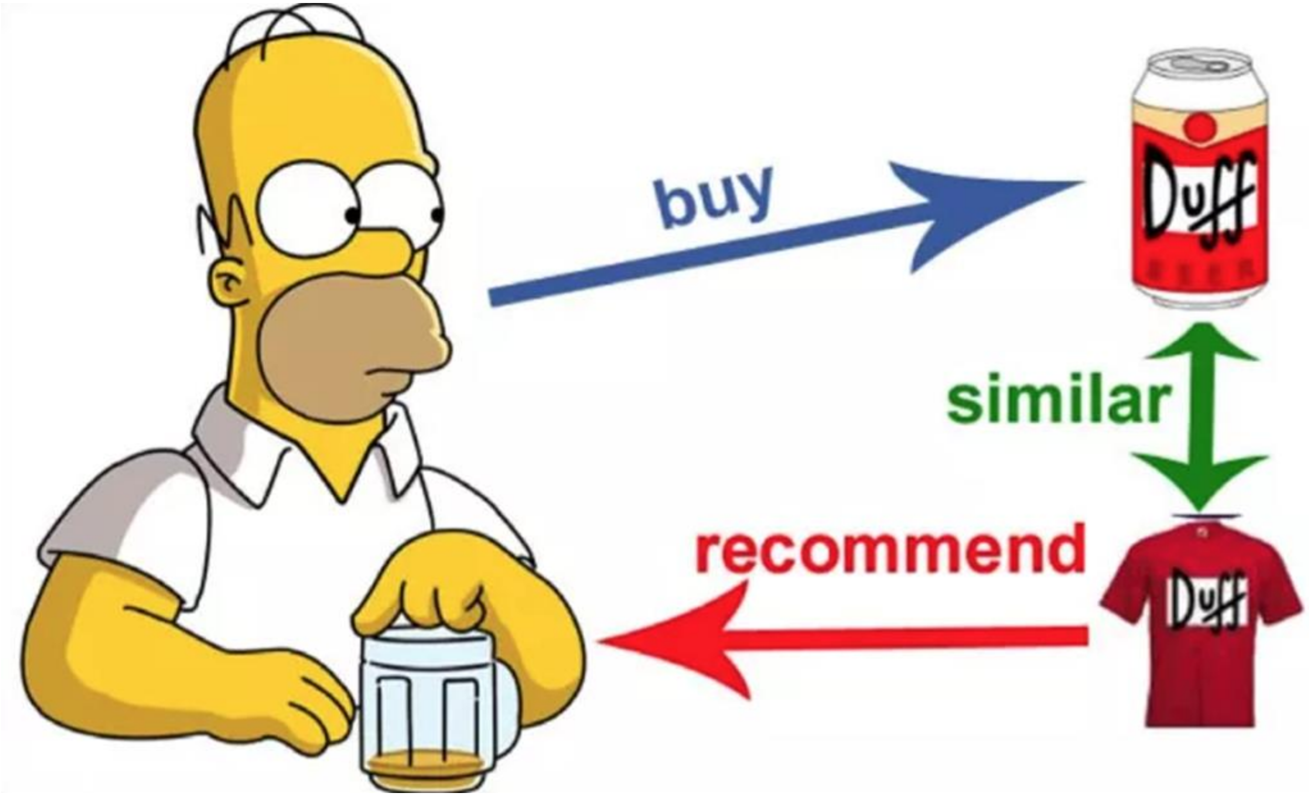
$$\frac{Ratings}{Movies \times Users} = \frac{100,480,507}{17,770 \times 480,189} \times 100 = 1.18\%$$

Content based filtering

2. Content based filtering(Content Extraction)

Based on “what does the user like about an item”:

- Meta-data extraction
- Clustering
- Similarity/distance between objects



2. Content based filtering– **Item-Item Similarity**

- Allow to compute similarities between items

- Does not require rating dataset

- The previous item-item recommendation algorithm still works

- No item cold start

- User attributes mitigate user cold start

In Production

4. In Production – Current Problematics

Data quality –implicit feedback; etc.

Sparsity – increase in size with items / users

Cold start problem – user cold start; item cold start

Recommendation speed – O(#items) algorithms not possible

4. In Production – Solutions

Data quality

Unbiased consumer app where the users enter their tastes(transfer learning)

Sparsity

User interaction: Ask each user to rate the most informative items

Cold start problem

Hybrid models with deep content extraction to recommend new items without ratings

Recommendation speed

Use GPU & TPU ☺

Use ligther method

4. In Production – Tools

LightFM

☺ open source: https://github.com/lyst/lightfm
☺ hybrid: matrix factorization + context

Deep Learning?

😐 way less tools than Computer Vision or NLP
😐 no pre-trained model available – you need large dataset and GPUs
😐 TensorFlow and PyTorch support for sparse data is limited

#DONTFORGETUS

آموزش های
رایگان بیشتر

www.data-hub.ir
www.youtube.com/channel/datahub1
www.linkedin.com/company/data-hub-ir
www.t.me/data_hub_ir
www.github.com/datahub-ir