the science behind Netflix recommendations

agenda

1. an introduction to recommendation engines (three types!)
2. how does Netflix make recommendations?
3. how can we code a recommendation engine?

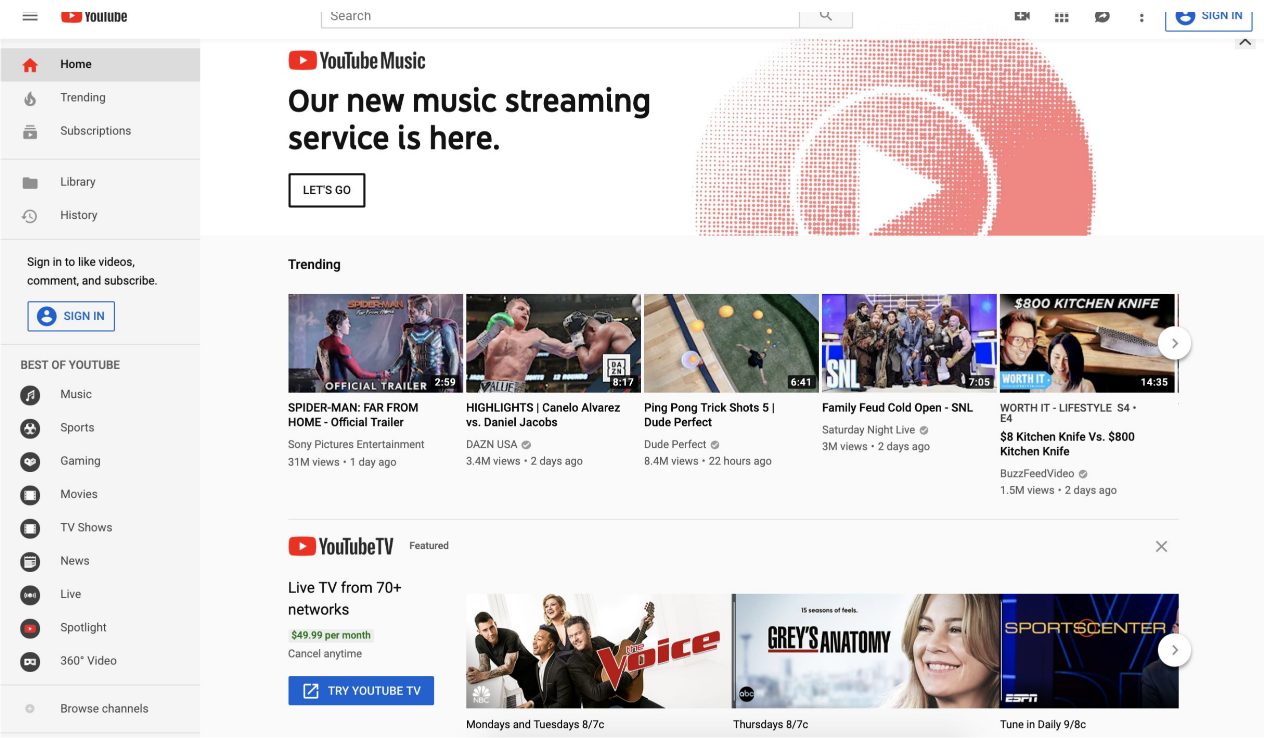
part 1: an introduction to recommendation engines

types of recommendation engines, and **why**???

- why do we need recommendation engines + what are some examples?

- three main types of recommendation engines:
  a. non-personalized
  b. content-based
  c. collaborative filtering

non-personalized recommendation engines



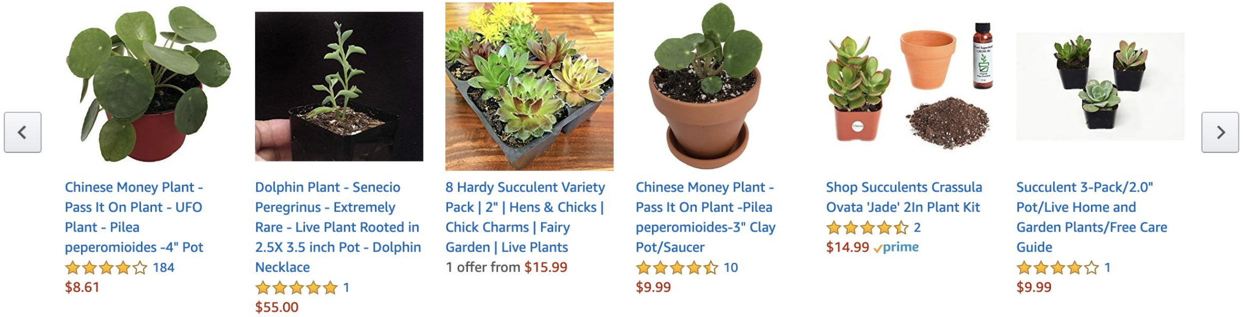
content-based recommendation engines

- makes recommendations based on an item's **features**

movies	Genre	Actor	Director	Year	IMDB	Rotten Tomatoes	...
1							
2							
3							
4							
5							
...							

content-based recommendation engines

- what are some pros and some pitfalls of content-based recommendations?



collaborative filtering

- recommends items **based on ratings of other users**

- different ways to do collaborative filtering:
    - model-based: Singular Value Decomposition (SVD)

collaborative filtering -- the **utility matrix**

- a utility matrix shows user ratings of different items

- the idea is to fill in the blanks

	Movie 1	Movie 2	Movie ...	Movie N
User 1	1	BLANK	BLANK	3
User 2	BLANK	5	BLANK	3
User 3	BLANK	BLANK	1	BLANK
User 4	2	3	BLANK	BLANK
User 5	BLANK	BLANK	1	BLANK
User 6	4	BLANK	5	BLANK
User 7	BLANK	4	BLANK	BLANK
User ...	BLANK	3	BLANK	BLANK
User m	BLANK	BLANK	BLANK	4

	Movie 1	Movie 2	Movie ...	Movie N
User 1	1	4	2	3
User 2	1	5	3	3
User 3	2.5	2.8	1	3.5
User 4	2	3	2	3.5
User 5	2.5	2.8	1	3.1
User 6	4	1.2	5	1.4
User 7	1	4	2.5	3
User ...	2	3	2	3
User m	1	4	2	4

modified **SVD** for filling in utility matrices

	U1	U2	U3	U4
I1	4	2	3	5
I2	3	2	4	2
I3	?	4	5	4
I4	3	2	4	4

≈

I1	×	×
I2	×	×
I3	×	×
I4	×	×

·

U1	U2	U3	U4
×	×	×	×
×	×	×	×

initiating our component matrices with 1s

I1	1	1
I2	1	1
I3	1	1
I4	1	1

•

U1	U2	U3	U4
1	1	1	1
1	1	1	1

=

	U1	U2	U3	U4
I1	2	2	2	2
I2	2	2	2	2
I3	2	2	2	2
I4	2	2	2	2

OG	U1	U2	U3	U4
I1	4	2	3	5
I2	3	2	4	2
I3	?	4	5	4
I4	3	2	4	4

RMSE = 1.75

gradient descent by reducing elementwise RMSE

I1	x	1
I2	1	1
I3	1	1
I4	1	1

•

U1	U2	U3	U4
1	1	1	1
1	1	1	1

=

	U1	U2	U3	U4
I1	x+1	x+1	x+1	x+1
I2	2	2	2	2
I3	2	2	2	2
I4	2	2	2	2

OG	U1	U2	U3	U4
I1	4	2	3	5
I2	3	2	4	2
I3	?	4	5	4
I4	3	2	4	4

Finding **x** to minimize:

$(4-(x+1))^2 + (2-(x+1))^2 + (3-(x+1))^2 + (5-(x+1))^2$

setting d/dx = 0, x = **2.5**

gradient descent by Alternating Least Squares

I1	2.5	1
I2	1	1
I3	1	1
I4	1	1

•

U1	U2	U3	U4
1	1	1	1
1	1	1	1

=

	U1	U2	U3	U4
I1	3.5	3.5	3.5	3.5
I2	2	2	2	2
I3	2	2	2	2
I4	2	2	2	2

OG	U1	U2	U3	U4
I1	4	2	3	5
I2	3	2	4	2
I3	?	4	5	4
I4	3	2	4	4

RMSE = 1.58!!

~~~machine learning~~~

I1	?	?
I2	?	?
I3	?	?
I4	?	?

•

U1	U2	U3	U4
?	?	?	?
?	?	?	?

=

	U1	U2	U3	U4
I1	3.55	2.91	3.68	3.85
I2	3.25	2.66	3.37	3.08
I3	3.7	3.42	4	3.85
I4	3.32	2.86	3.6	3.49

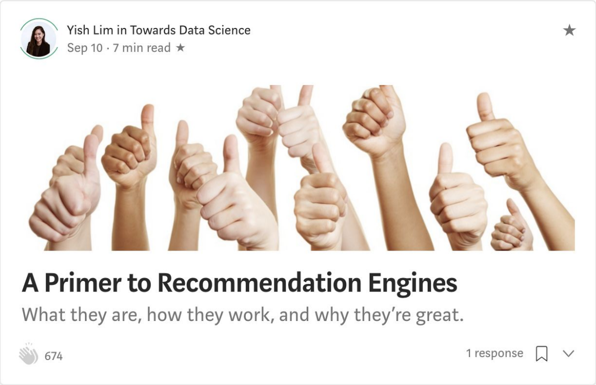
OG	U1	U2	U3	U4
I1	4	2	3	5
I2	3	2	4	2
I3	?	4	5	4
I4	3	2	4	4

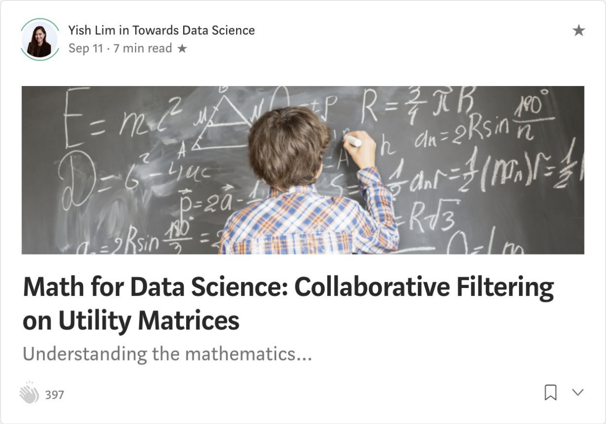
RMSE = 1.15

collaborative filtering -- pros and cons

- personalized for each user!

- computationally heavy
- popularity bias
- the **cold start** problem



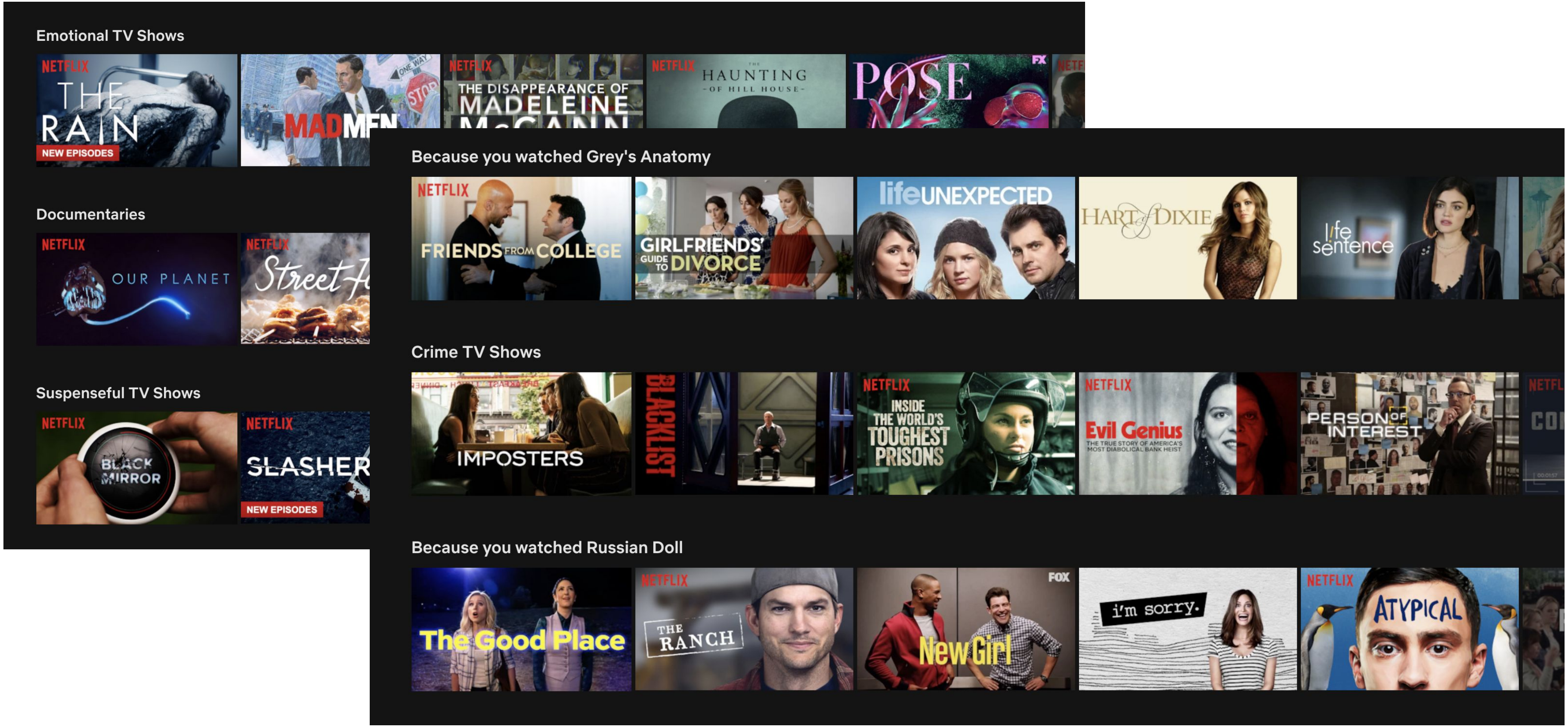


// part 2: how does Netflix make recommendations?

the Netflix algorithm

- Netflix uses a **hybrid** of content-based and collaborative filtering
- content based: tagging
- collaborative filtering: user patterns, user similarities

the Netflix algorithm



part 3: coding our own recommendation engine!

stuff we've learned

recommendation engines!!!

1. non-personalized
2. content-based
3. model-based collaborative filtering with SVD

code-along: https://github.com/yishuen/meetup-movie-recommender