The Science Behind Netflix Recommendations

Presented by:
Ben Oren
Data Science Coach, Chicago Campus, Flatiron School
April 22nd, 2020

Part 0: Welcome and Introductions

Introductions

Welcome!

Introductions

Welcome!

Interactive elements

Introductions

Welcome!

Interactive elements

- During: Google doc

Introductions

Welcome!

Interactive elements

- During: Google doc
- After: Github

Introductions

Welcome!

Interactive elements

- During: Google doc
- After: Github

Introductions

Welcome!

Interactive elements

- During: Google doc
- After: Github

Need accounts

Introductions

Introductions

Zoom chat -> Google doc -> Introduce yourself

Introductions

Zoom chat -> Google doc -> Introduce yourself

- Name

Introductions

Zoom chat -> Google doc -> Introduce yourself

- Name
- Quick blurb about you

Introductions

Zoom chat -> Google doc -> Introduce yourself

- Name
- Quick blurb about you
- Linkedin / git / twitter

Introductions

MOD 1
Weeks 1 - 3

Data engineering

Statistics / regression

MOD 2
Weeks 4 - 6

MOD 3
Weeks 7 - 9

Machine learning

Advanced topics

MOD 4
Weeks 10 - 12

MOD 5
Weeks 13 - 15

Capstone

Introductions

MOD 1
Weeks 1 - 3

Data engineering

Statistics / regression

MOD 2
Weeks 4 - 6

MOD 3
Weeks 7 - 9

Machine learning

Advanced topics

MOD 4
Weeks 10 - 12

MOD 5
Weeks 13 - 15

Capstone

Part 1: Intro to Recommendation Engines

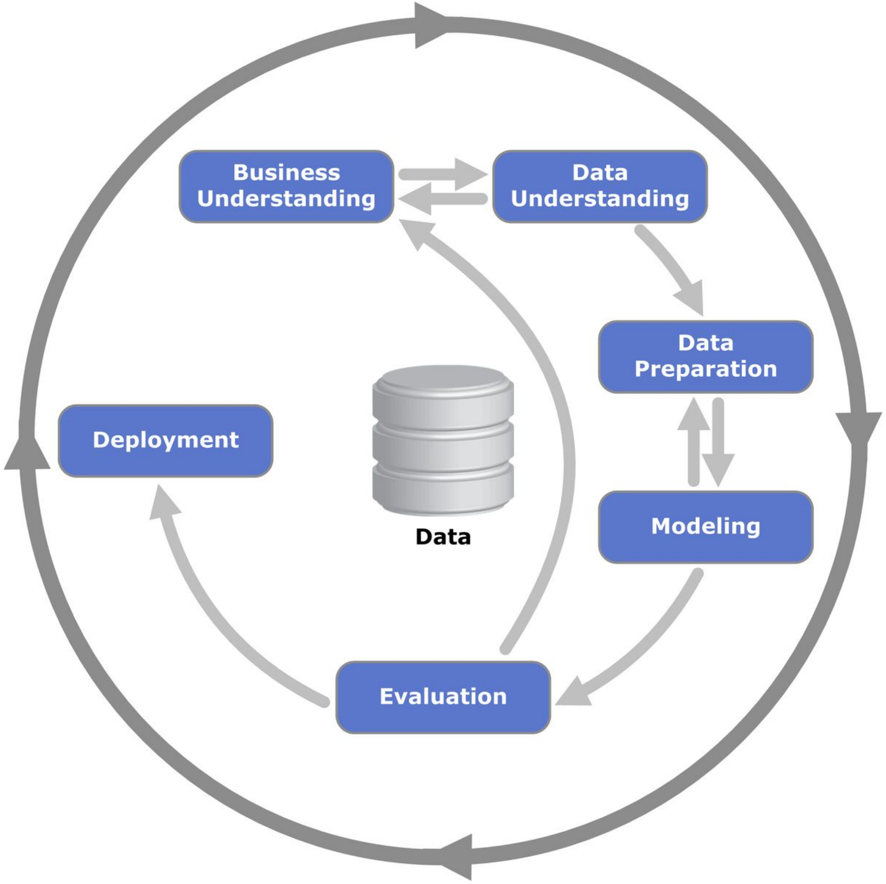
How does Netflix make recommendations??!?!!?!?

Thinking like a data scientist

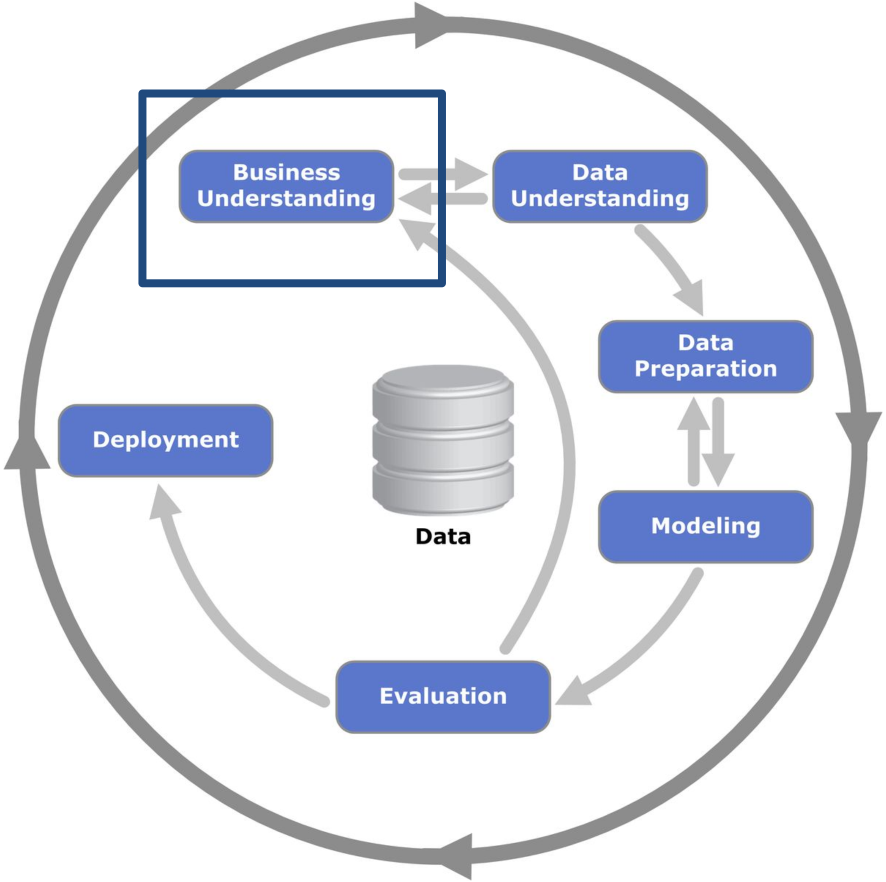
How does Netflix make recommendations??!?!!?!?

Broader data science question: why make recommendations?

Thinking like a data scientist



Thinking like a data scientist



Thinking like a data scientist

Thinking like a data scientist

Start user engagement

Thinking like a data scientist

Start user engagement

Increase user engagement

Thinking like a data scientist

Start user engagement

Increase user engagement

Help discover new content

Thinking like a data scientist

If you had to recommend a movie to me to watch after event, what would you recommend?

Thinking like a data scientist

If you had to recommend a movie to me to watch after event, what would you recommend?

-> Google doc

Background concepts - cold start problem

How did you make your rec?

Currently trending?

Things you’ve heard are good?

Classics?

. . . kind of hard to pick criteria, right?

Background concepts - cold start problem

"Cold start problem" - making recs w/o having any information about the user

Possible solutions

Make generic recs

- Popular
- Trending

Force user to provide data

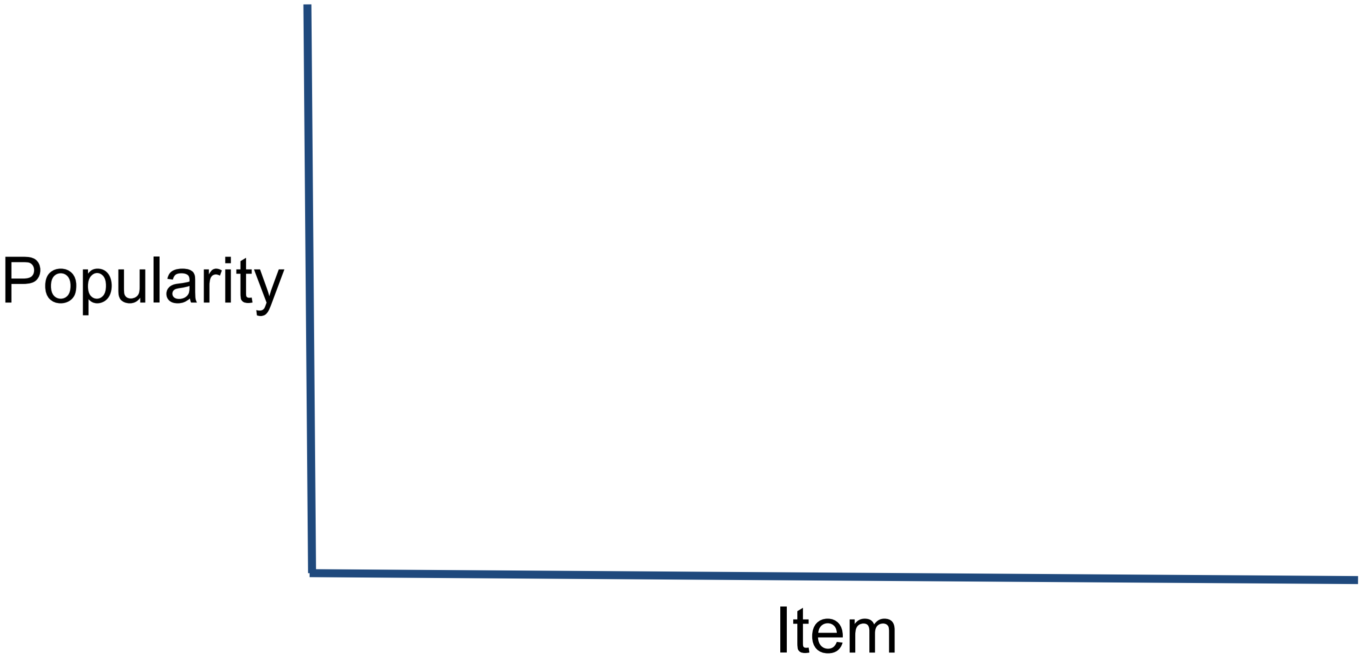
- Integrate w/ social media
- Answer ?s when signing up

Background concepts - long tail

Broadly correspond to “the long tail”

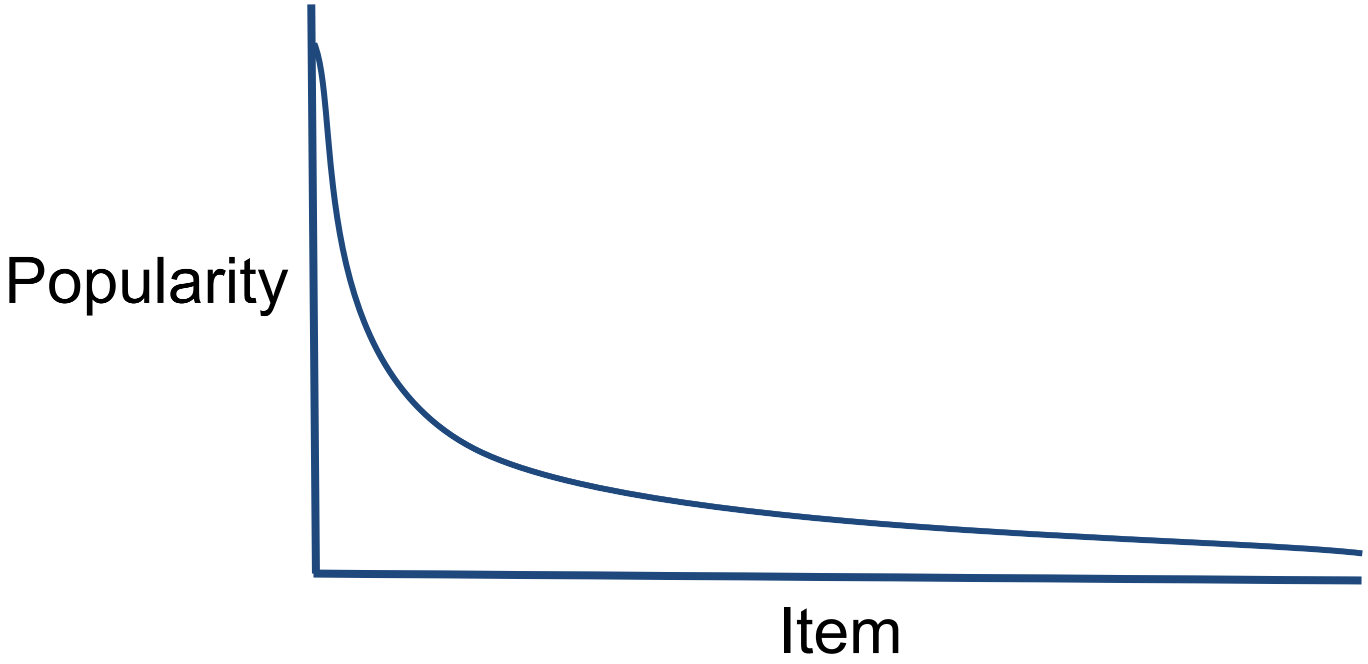
Background concepts - long tail

Broadly correspond to “the long tail”



Background concepts - long tail

Broadly correspond to “the long tail”

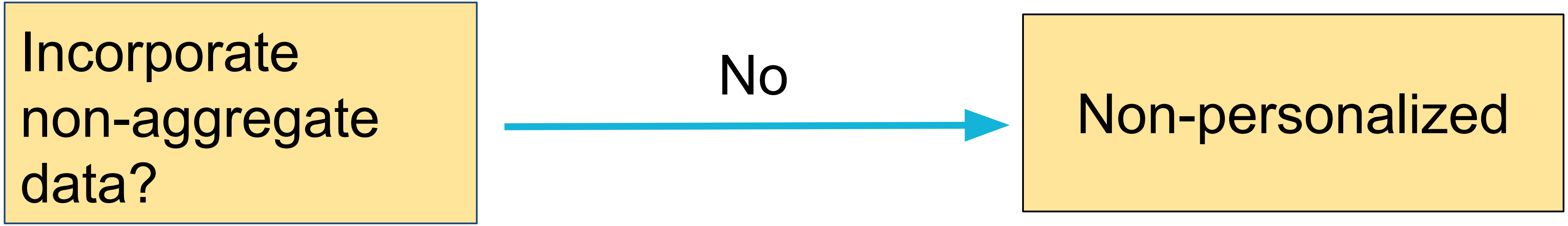


Part 2: Types of Recommendation Engines

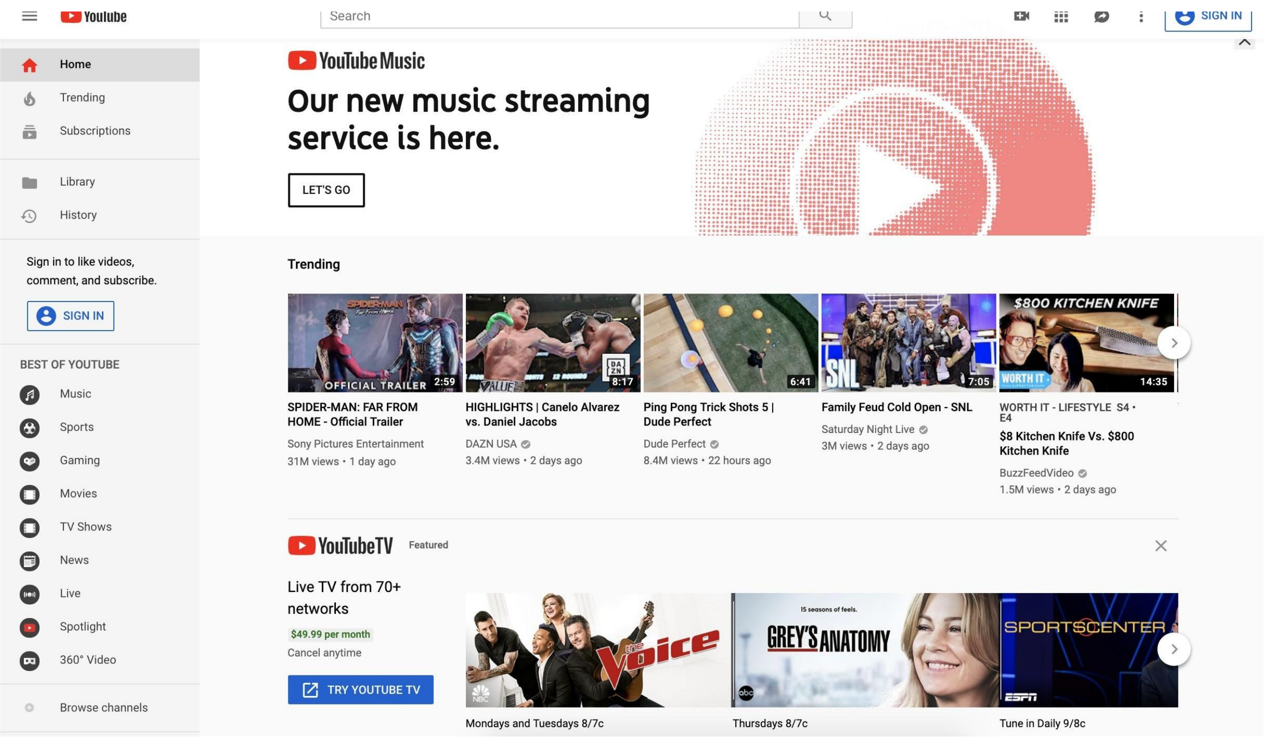
Types of recommendation engines

Incorporate non-aggregate data?

Types of recommendation engines



Non-personalized recommendation engines



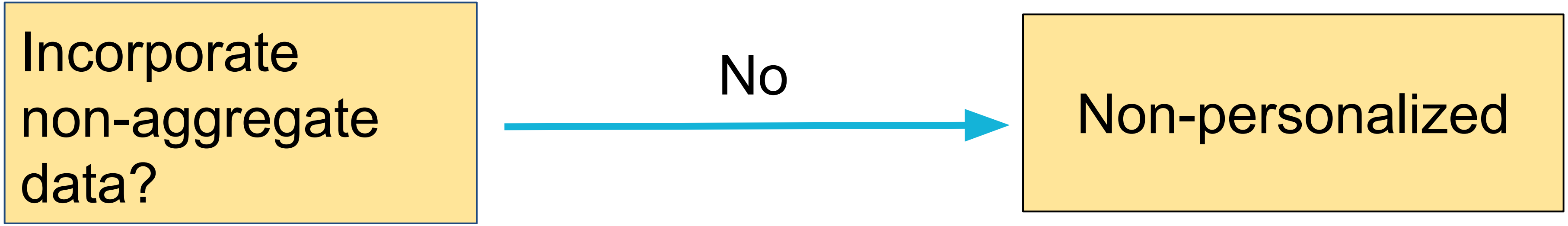
Pros

- Easy
- Cheap computation
- “Cold start”

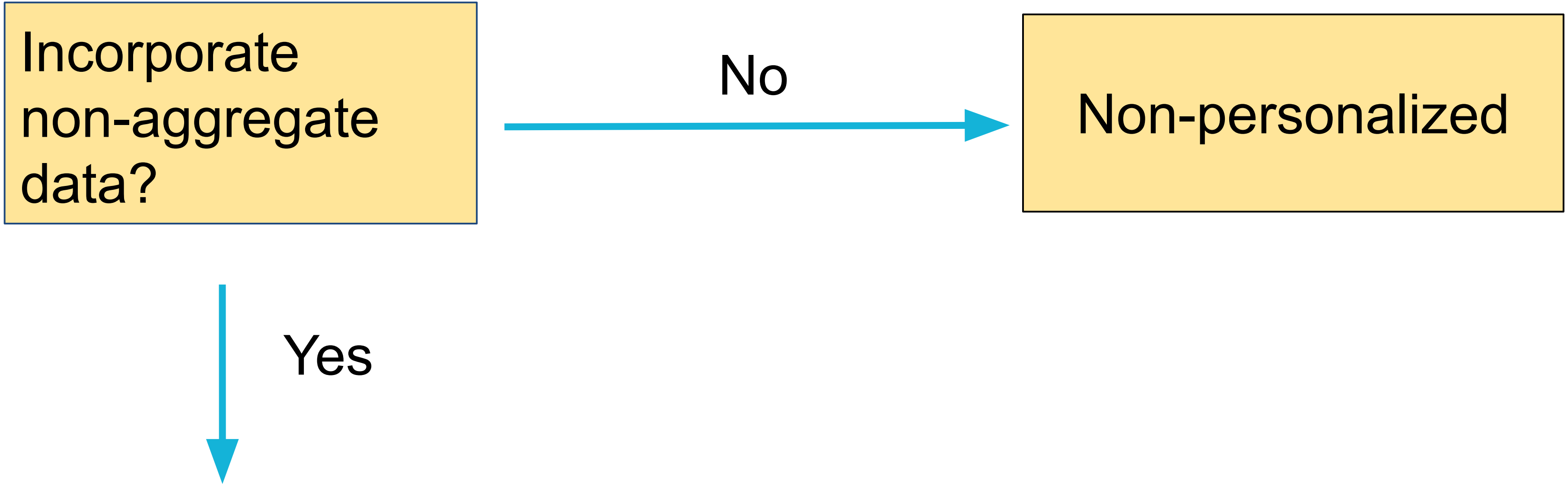
Cons

- Not personalized
- New item traction

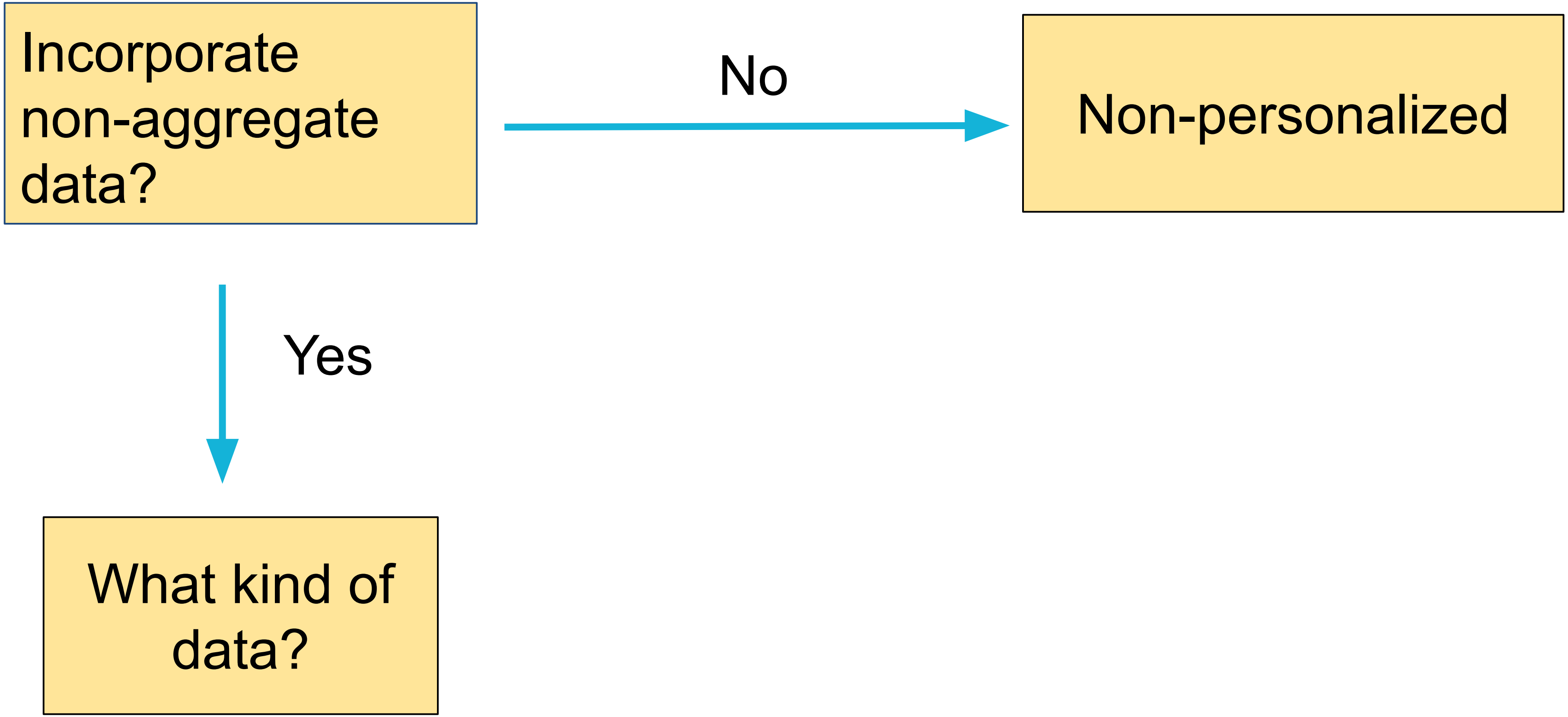
Types of recommendation engines



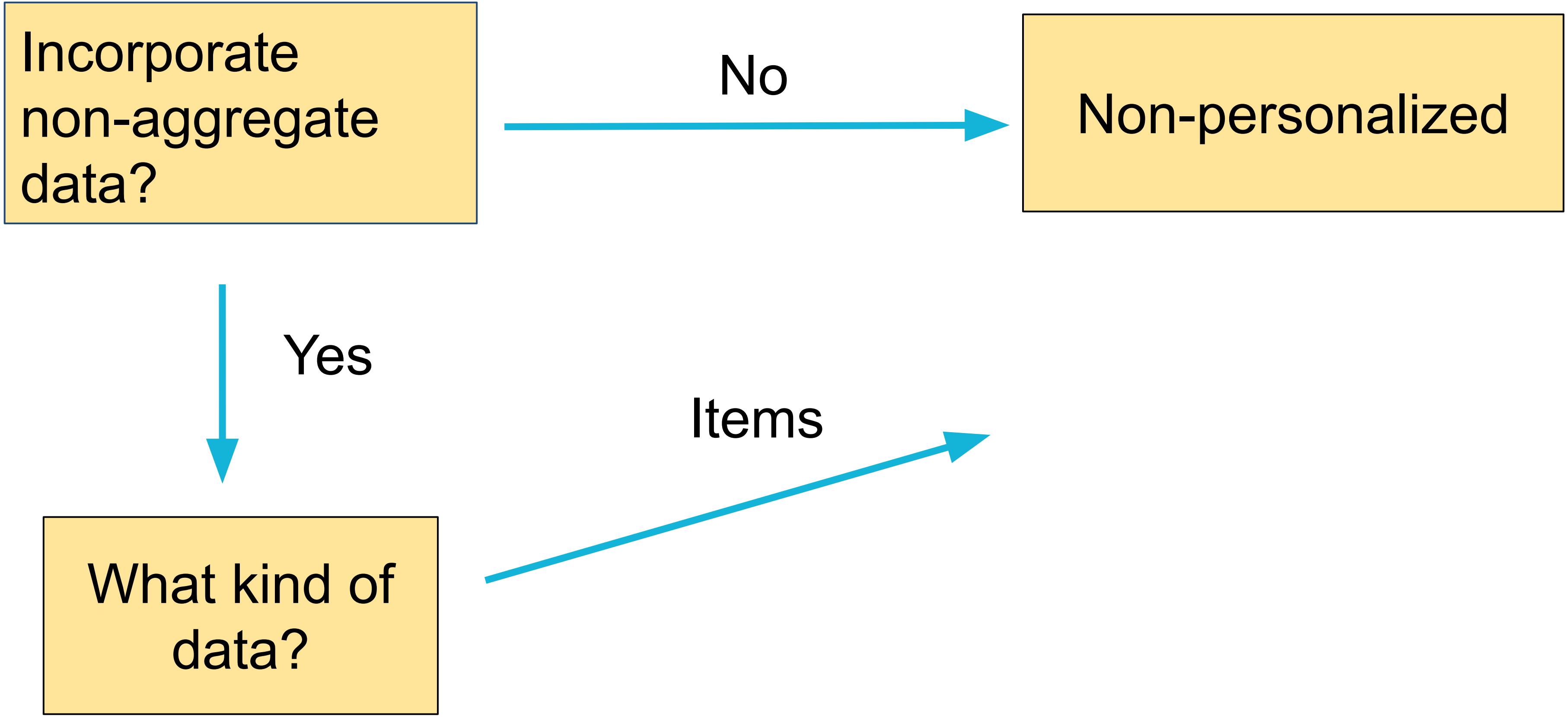
Types of recommendation engines



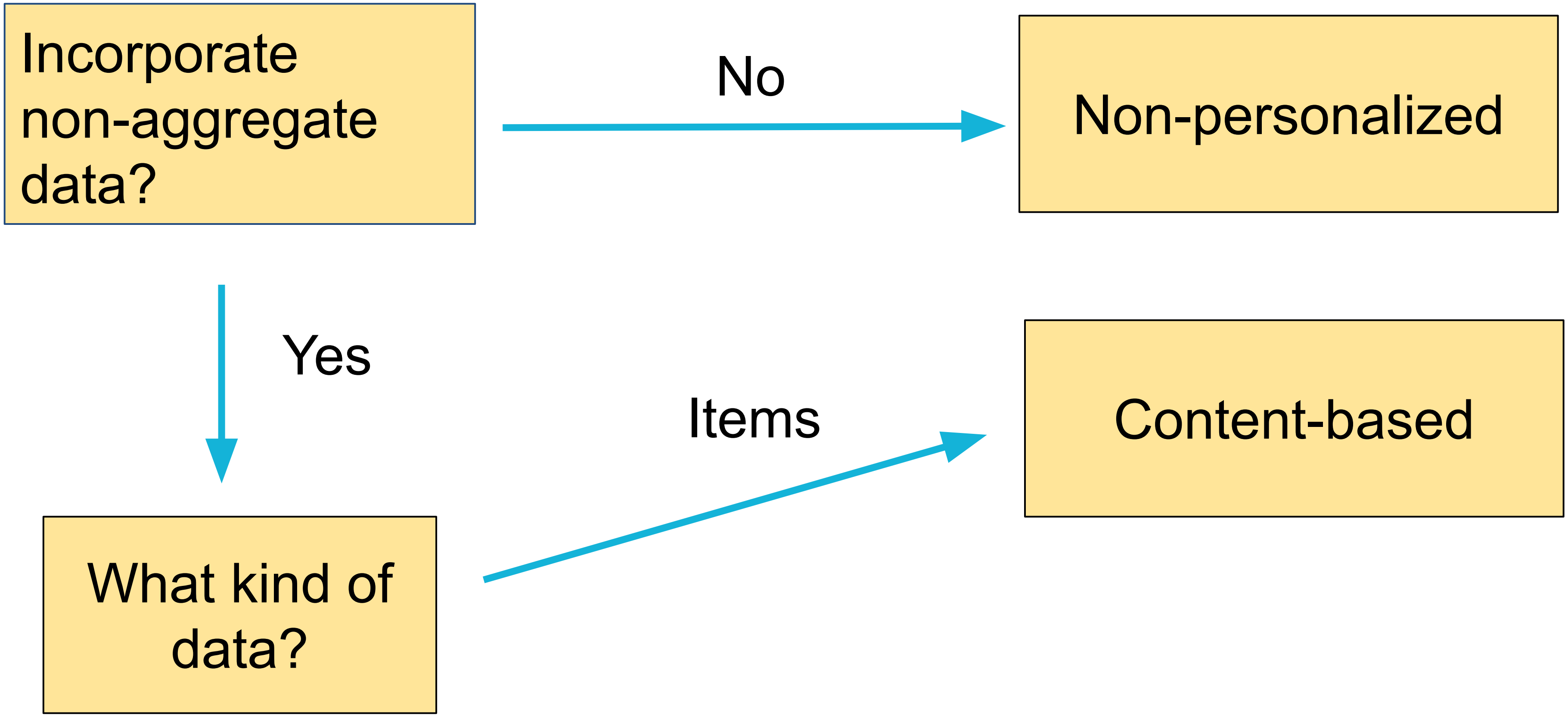
Types of recommendation engines



Types of recommendation engines



Types of recommendation engines



Content systems

Makes recommendations based on an item's **features**

movies	Rotten Tomatoes score	# of views	IMDB Rating	Year	Director	Actor	Genre	Critically Acclaimed?	Classic?	. . .
1										
2										
3										
4										
5										
...										

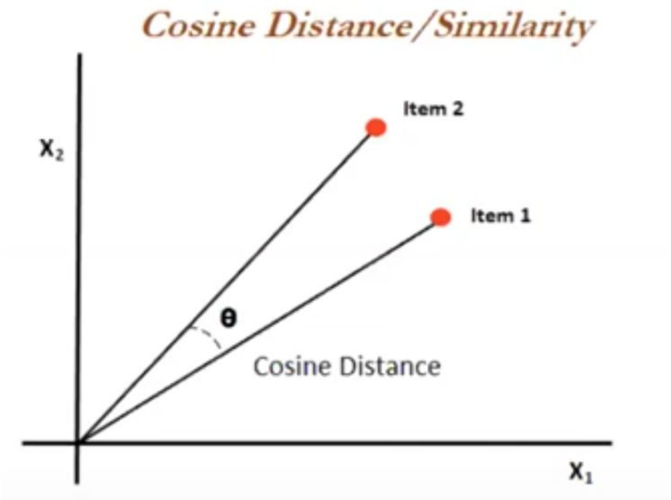
Content systems

Build a content-based recommendation system step-by-step

Content systems

Build a content-based recommendation system step-by-step

- Collect data and perform exploratory data analysis (EDA)
- Calculate (cosine) similarity or other similarity metrics for each pair-wise item
- For a given item, find the top N items that have the closest values
- Similarity metrics: cosine distance, Jaccard distance, Pearson correlation



Content systems

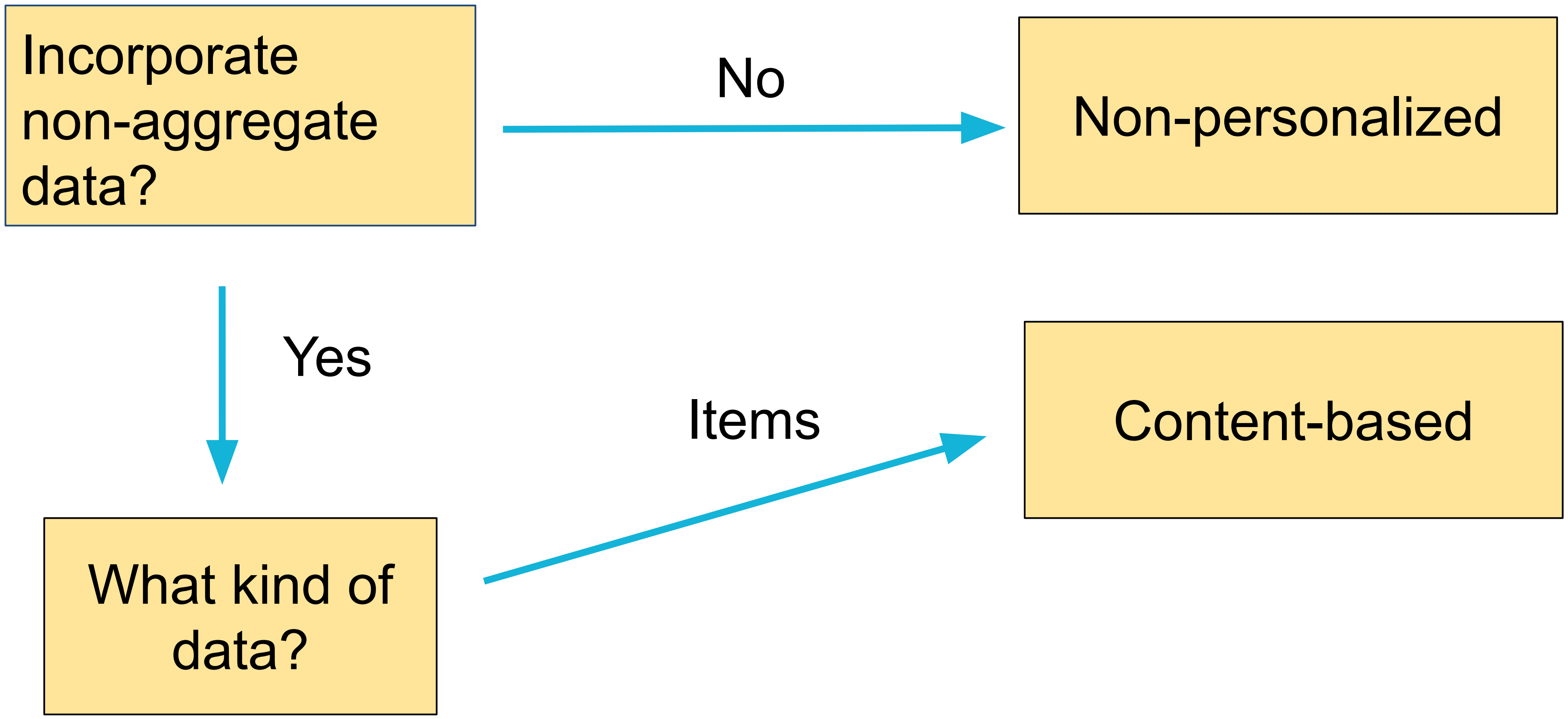
Pros

- Perform calcs immediately (cold start)
- Easy
- Recommendations to unique tastes

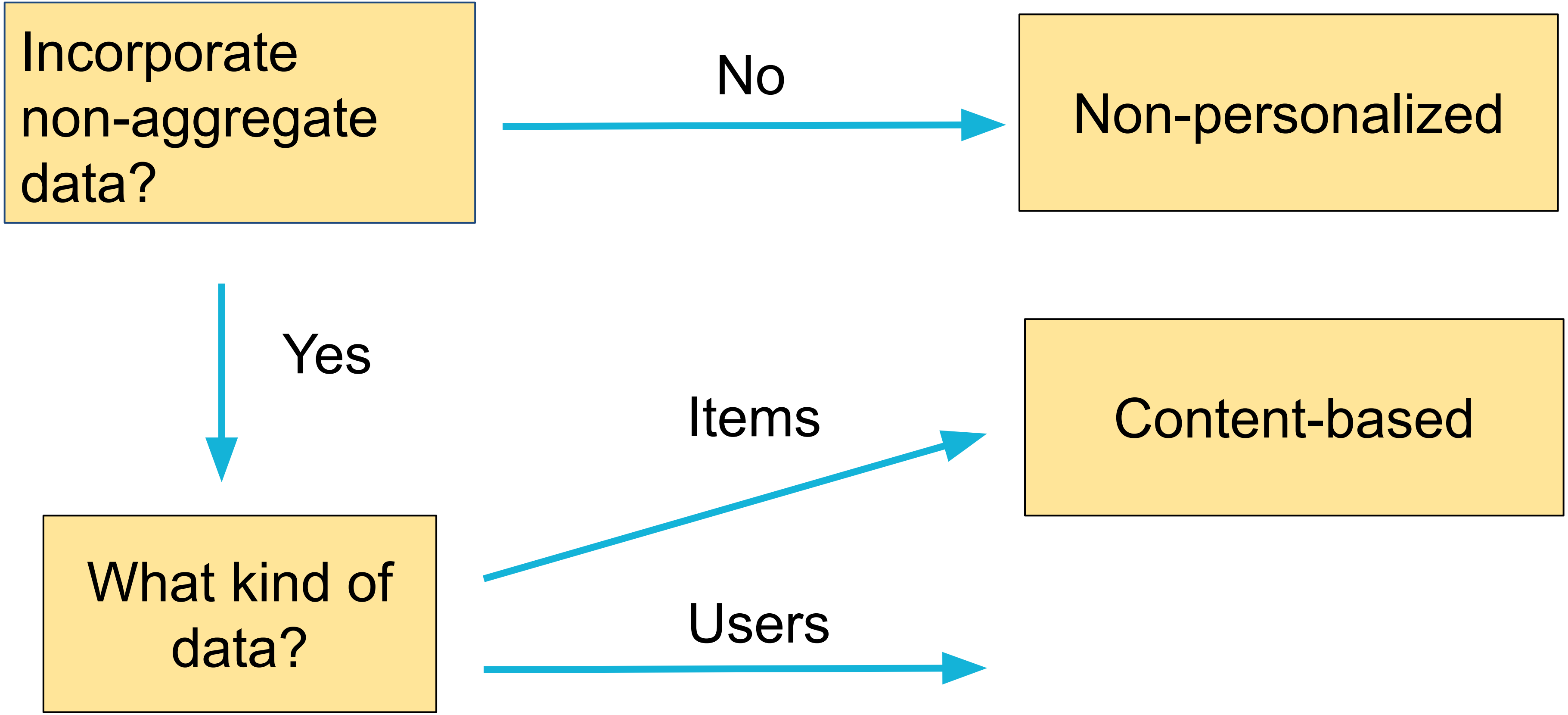
Cons

- Requires tagging
- Over-specialization of specific types of items

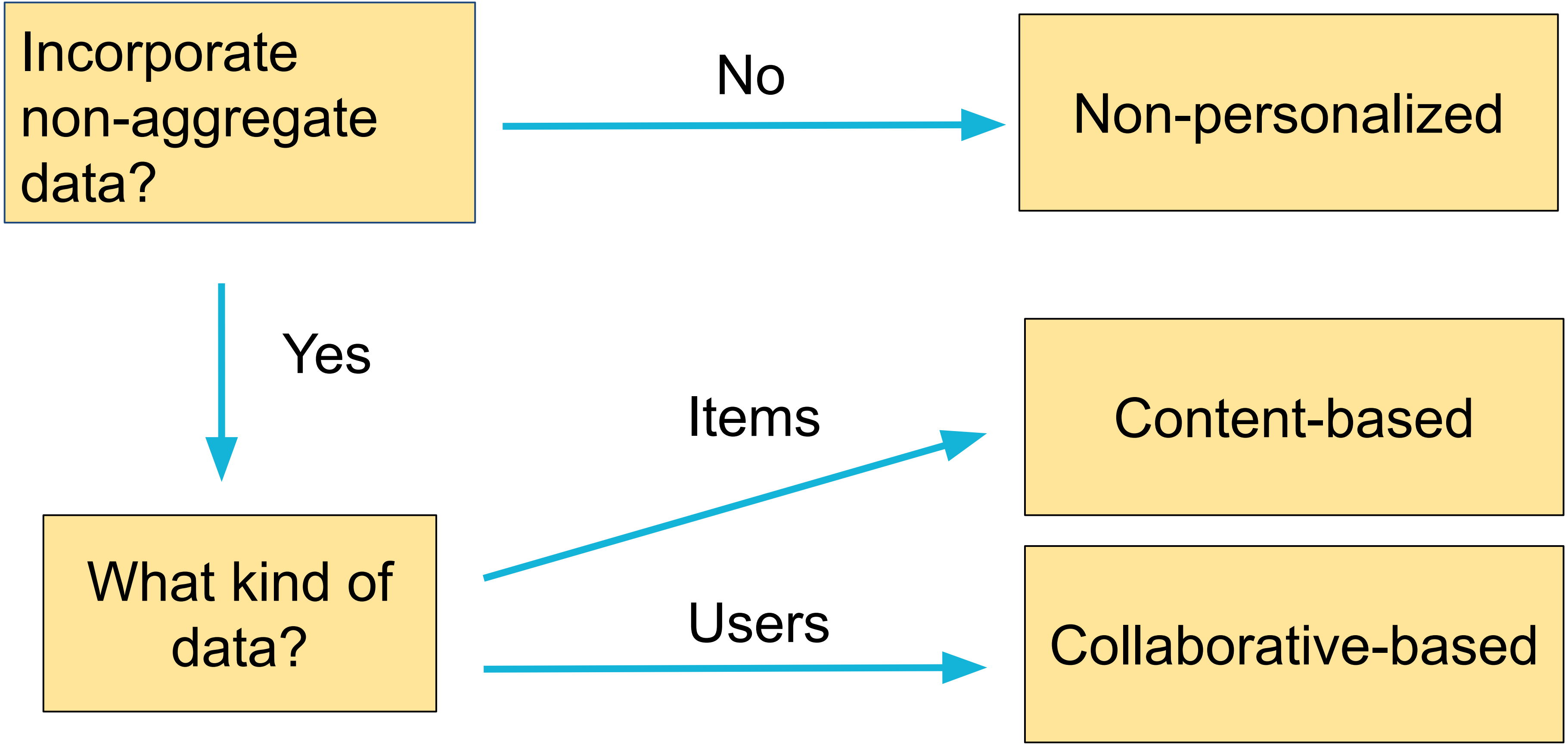
Types of recommendation engines



Types of recommendation engines



Types of recommendation engines



Collaborative systems

Incorporate user preferences (usually, ratings of items)

Problem: most users have few ratings

Solution: find relationships among users and use that info to calculate ratings

Collaborative systems -- the utility matrix

Utility matrix - “sparse”

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7	Movie 8	Movie 9
Ben	2	2	BLANK	3	BLANK	BLANK	4	BLANK	1
Max	BLANK	3	3	BLANK	4	5	BLANK	5	2
Joél	BLANK	BLANK	4	BLANK	3	BLANK	4	BLANK	BLANK
Erin	2	BLANK	BLANK	1	3	2	BLANK	BLANK	3
Greg	BLANK	2	BLANK	2	BLANK	3	BLANK	1	BLANK

Collaborative systems -- the **utility matrix**

Utility matrix - calculated

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7	Movie 8	Movie 9
Ben	2	2	4	3	1	2	4	1	1
Max	3	3	3	4	4	5	4	5	2
Joél	2	4	4	5	3	4	4	5	2
Erin	2	1	2	1	3	2	3	1	3
Greg	5	2	3	2	5	3	5	1	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

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	x	x
I2	x	x
I3	x	x
I4	x	x

·

U1	U2	U3	U4
x	x	x	x
x	x	x	x

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

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

initiating our component matrices with 1s

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

How compare?

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

initiating our component matrices with 1s

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

How compare?

RMSE = 1.75

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

Improve with gradient descent and alternating least squares

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

Improve with gradient descent and alternating least squares

	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

Improve with gradient descent and alternating least squares

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

Minimize RMSE through gradient descent

	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

Improve with gradient descent and alternating least squares

Minimize RMSE through gradient descent

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

	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

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

Improve with gradient descent and alternating least squares

Minimize RMSE through gradient descent

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

	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

minimize: $(3-x)^2 + (1-x)^2 + (2-x)^2 + (4-x)^2$

Improve with gradient descent and alternating least squares

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

Minimize RMSE through gradient descent

	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

minimize: $(3-x)^2 + (1-x)^2 + (2-x)^2 + (4-x)^2$

dy / dx = 2*(3-x)*-1+ 2*(1-x)*-1+ 2*(2-x)*-1+ 2*(4-x)*-1

Improve with gradient descent and alternating least squares

Minimize RMSE through gradient descent

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

	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

minimize: $(3-x)^2 + (1-x)^2 + (2-x)^2 + (4-x)^2$

$0 = 2*(3-x)*-1 + 2*(1-x)*-1 + 2*(2-x)*-1 + 2*(4-x)*-1$

Improve with gradient descent and alternating least squares

Minimize RMSE through gradient descent

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

	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

minimize: $(3-x)^2 + (1-x)^2 + (2-x)^2 + (4-x)^2$

$x = 2.5$

$0 = 2*(3-x)*-1 + 2*(1-x)*-1 + 2*(2-x)*-1 + 2*(4-x)*-1$

Improve with gradient descent and 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

Improve with gradient descent and 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

Improve with gradient descent and alternating least squares

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

	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

Improve with gradient descent and alternating least squares

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!!

	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

~~~machine learning~~~

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

$\cdot$

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

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

·

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

~~~machine learning~~~

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

·

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

~~~machine learning~~~

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

·

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

~~~machine learning~~~

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

·

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

~~~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

Final RMSE: 1.18

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

	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

~~~machine learning~~~

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

Final RMSE: 1.18

Rating for user 1 for item 3: 3.7

	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

Collaborative systems -- pros and cons

Pros:

- Personalized
- Recommends niche items (long tail)

Cons:

- Computationally expensive
- Popularity bias
- Needs ratings (cold start)

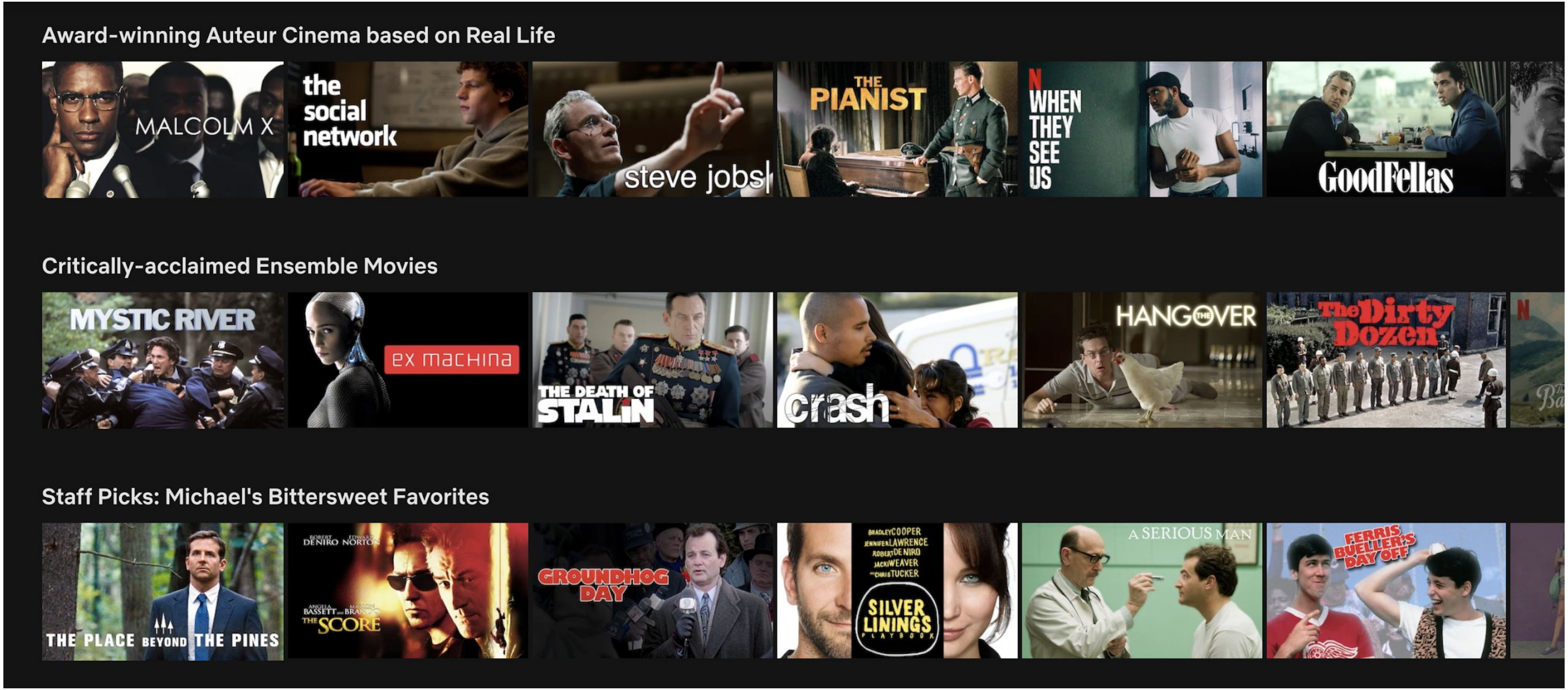
part 3: So . . . how *does* Netflix make recommendations?

The Netflix algorithm

Netflix uses a *hybrid* of content-based and collaborative-based systems:

- Manually tag movies with categories
- After grouping items together, then do collaborative system. Utility matrix uses “groupings” instead of “items”
- User specific behavior: time of day, time of week
- Fun A/B testing

The Netflix algorithm



part 4: Summary and code-along

Stuff we've learned

Data science process (CRISP-DM)

Problems

- Cold start
- Long tail

Types of systems

- Non-personalized
- Content-based
- Collaborative-based

Content-system process and Model-based SVD walk-through

Stuff we've learned

Fun doesn't stop: code along!

github.com/ben-oren/science-of-netflix

Thanks everyone!

ben.oren@flatironschool.com
linked-in.com/ben-oren

Created by **Yish Lim**

Flatiron School Graduate // Flatiron School Data Science Coach // Data scientist and machine learning engineer with a passion for finding the narrative within numbers.

LinkedIn: https://www.linkedin.com/in/yishuen-lim/

Github: github.com/yishuen

limyishuen@gmail.com

Have questions?

Send them in the chat box.

FLATIRON SCHOOL

FLATIRON SCHOOL

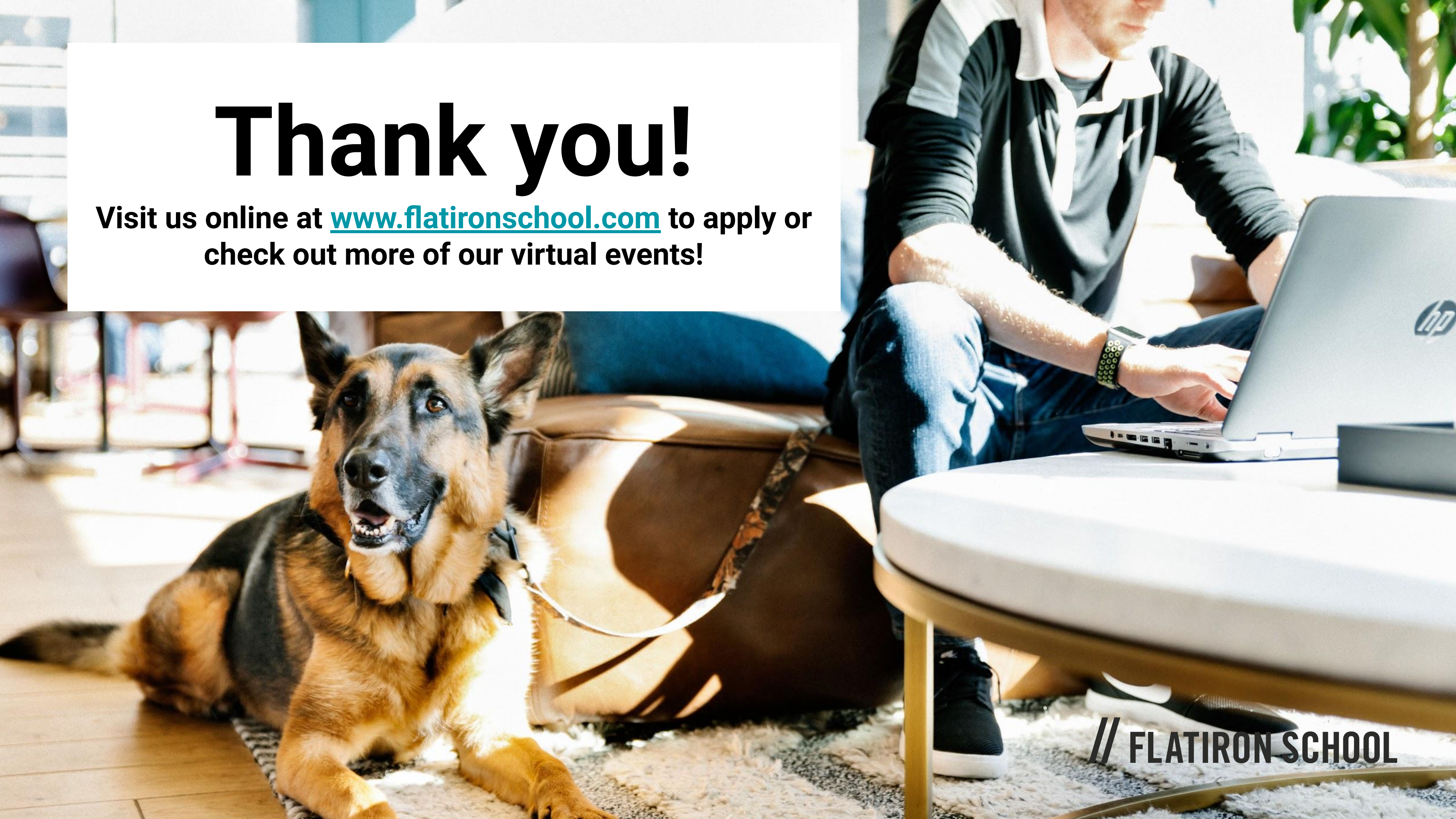
CHANGE

Weeks 0/6

#changethings challenge

Join our Facebook group and share your experience with other people looking to connect and grow while navigating uncharted waters. **Every week for the next six weeks,** we'll provide a roadmap to inspire communication and collaboration. In addition to weekly virtual workshops, AMAs, and community chats, we'll offer inspiration, recommendations, and our free bootcamp prep courses as materials to guide your growth.

Join "Flatiron School #changethings Community" on Facebook



Thank you!

Visit us online at www.flatironschool.com to apply or check out more of our virtual events!

// FLATIRON SCHOOL