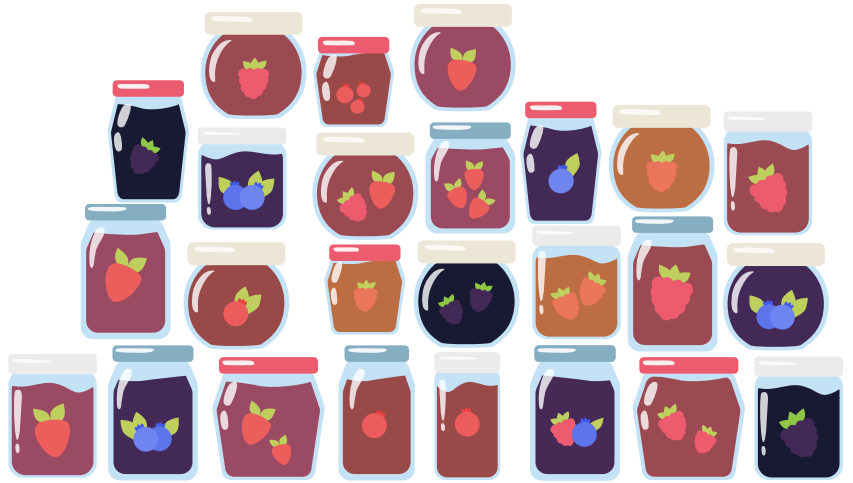
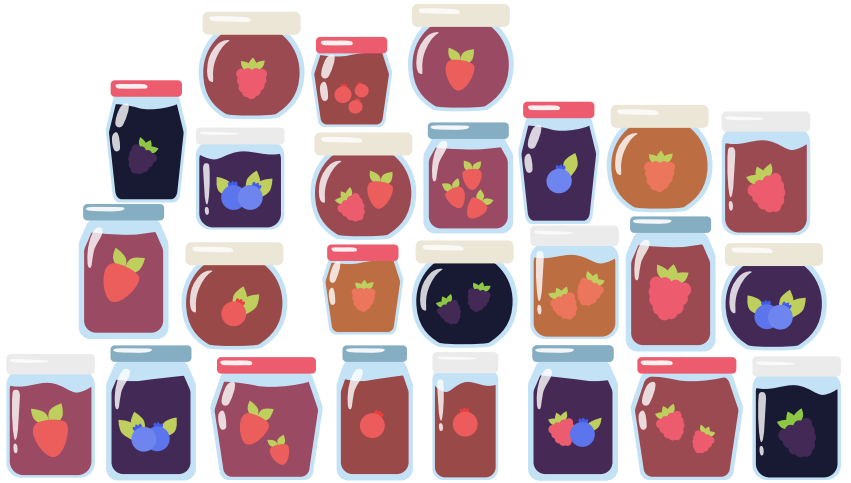
Booth A



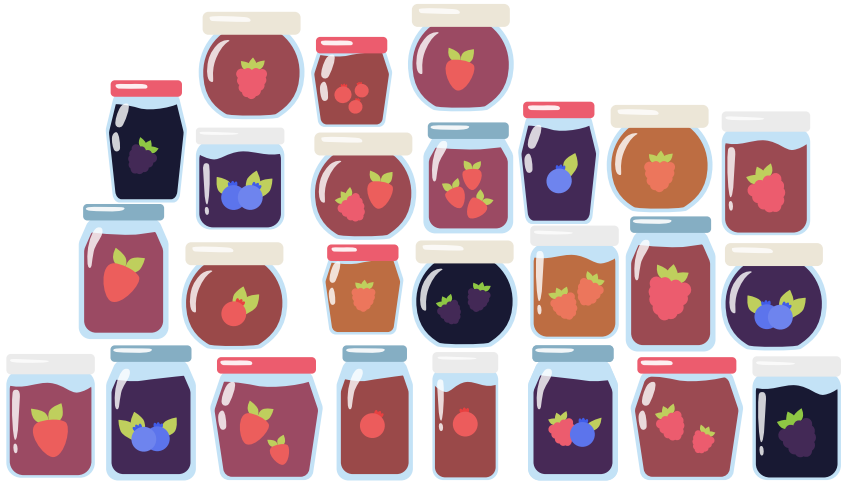
Booth B



Booth A

Booth B

Booth A		Booth B
242	Walked by	260
104 (40 %)	Stopped to taste	145 (60 %)



Booth A

Booth B

Booth A		Booth B
242	Walked by	260
104 (40 %)	Stopped to taste	145 (60 %)
31 (30%)	Bought jam	4 (3 %)


B
MOVIE
SISTER
CHRIS FARLEY
HADERSFIELD
SHEA
Hawaii
HIROSHIMA
Open
Road
MILK
PEOPLE
HOLE
KM.O

Recommendation Systems

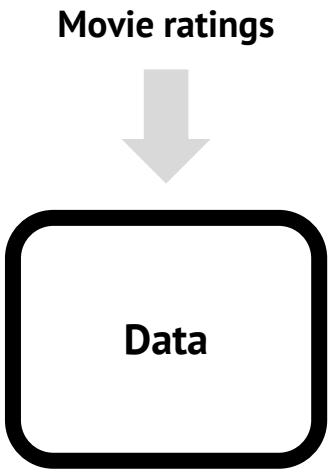
An Overview

Marie Roald

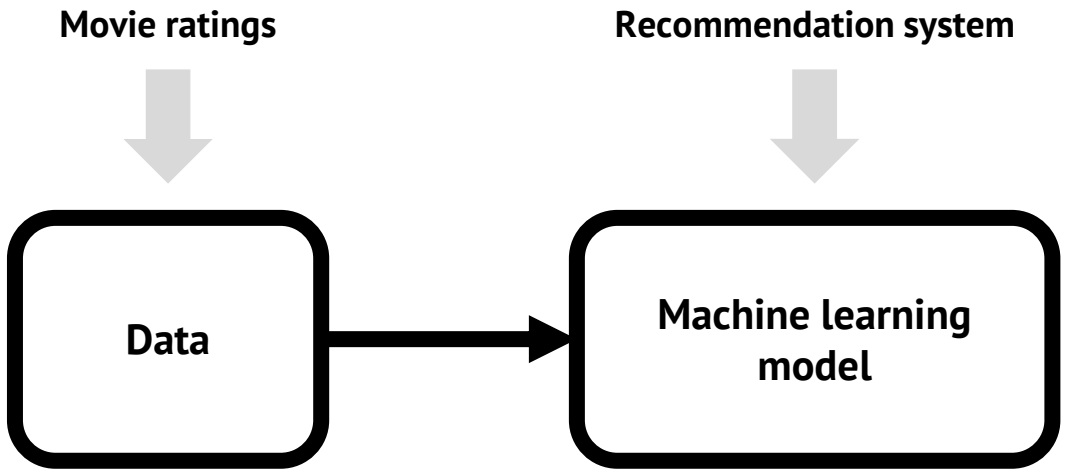
PhD Defence Trial lecture
2023-09-04

OSLOMET simulamet

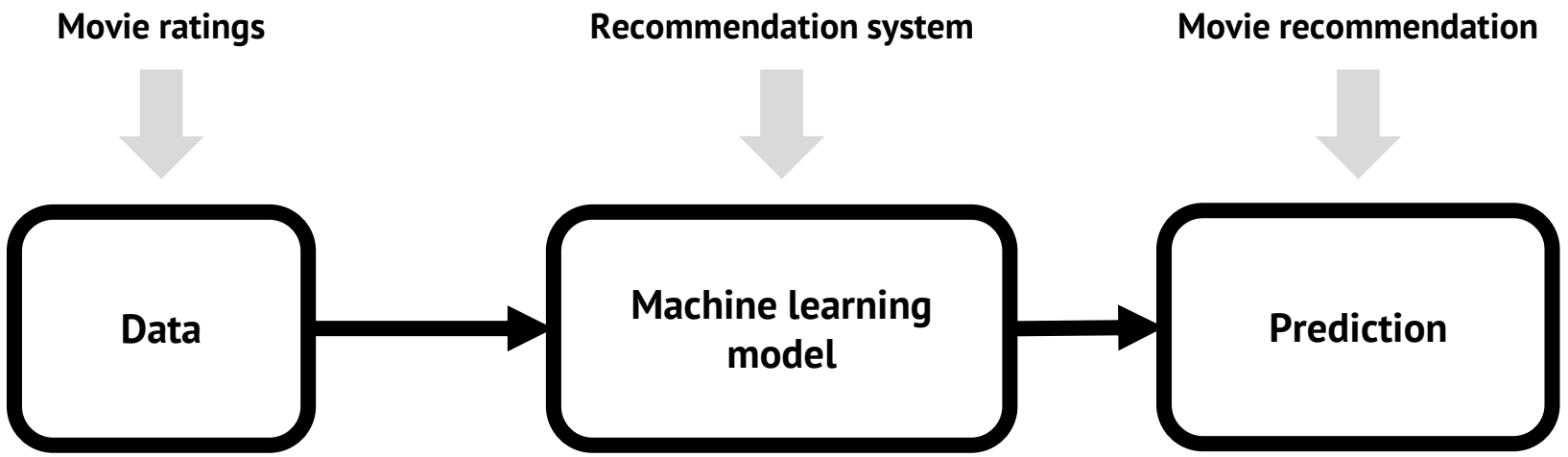
Recommendation engines are machine learning models that predict preferences based on data



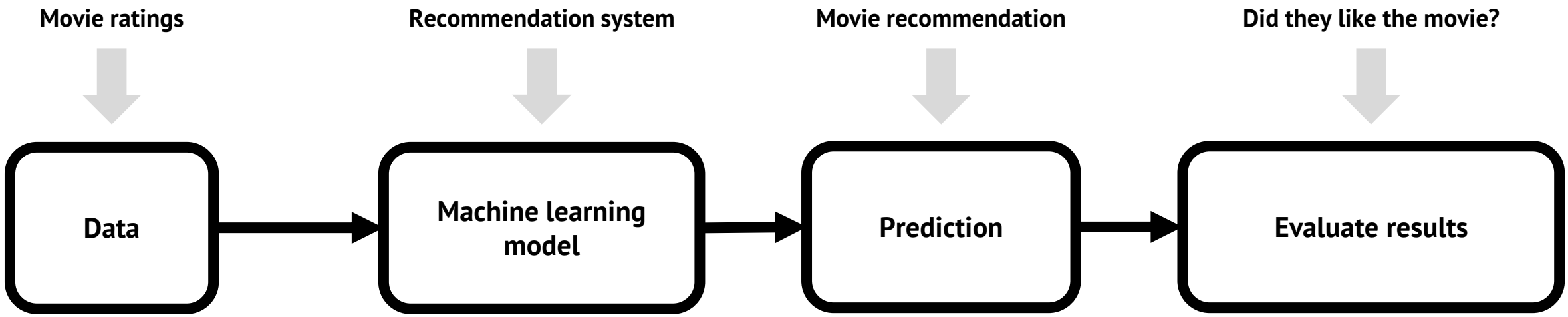
Recommendation engines are machine learning models that predict preferences based on data



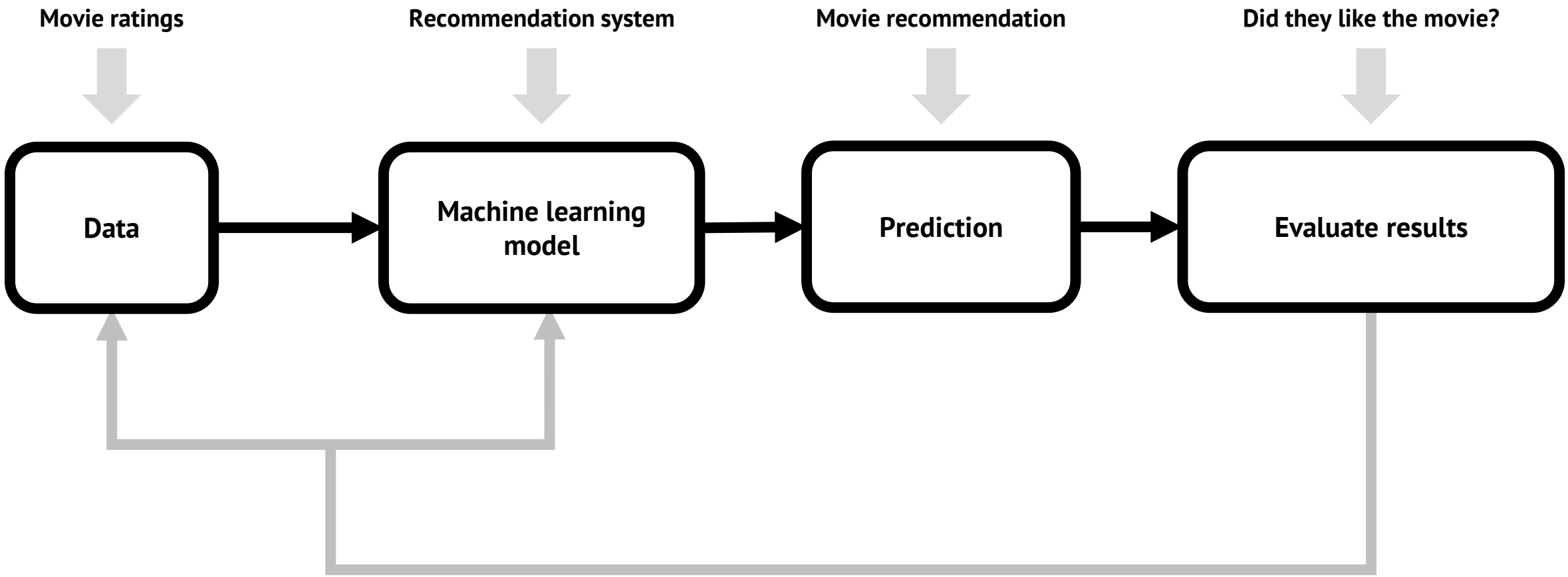
Recommendation engines are machine learning models that predict preferences based on data



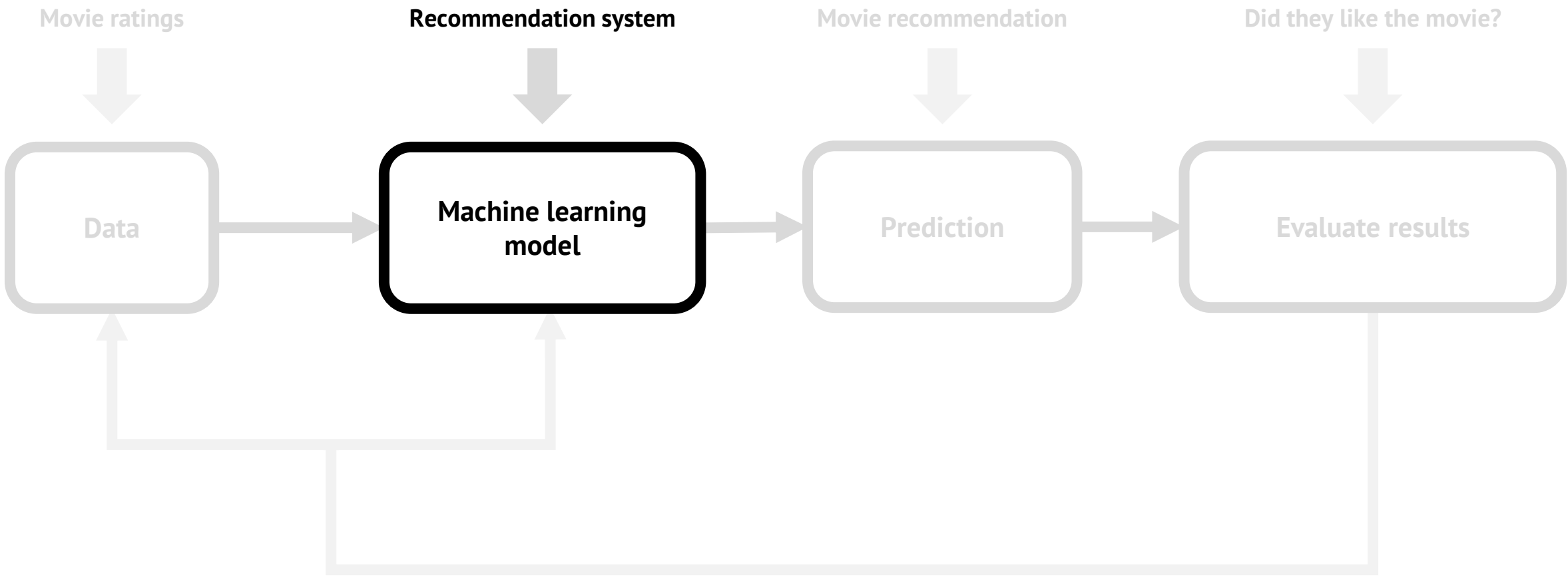
Recommendation engines are machine learning models that predict preferences based on data

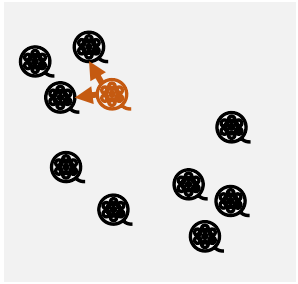


Recommendation engines are machine learning models that predict preferences based on data



Recommendation engines are machine learning models that predict preferences based on data



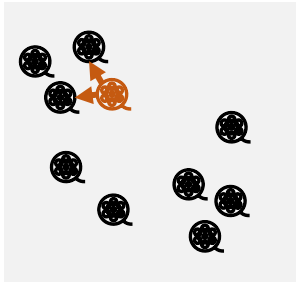


Content-based filtering

5	6	6	1	5	5
7	8	8	2	7	7
3	3	4	2	5	6
7	8	?	3	10	10
2	2	3	2	5	6

Collaborative filtering

Session-based filtering



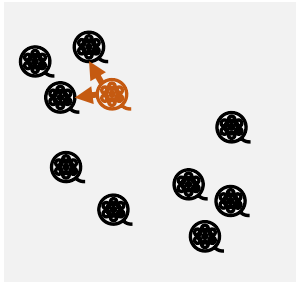
Content-based filtering

5	6	6	1	5	5
7	8	8	2	7	7
3	3	4	2	5	6
7	8	?	3	10	10
2	2	3	2	5	6

Collaborative filtering

Session-based filtering

Concluding remarks



Content-based filtering

5	6	6	1	5	5
7	8	8	2	7	7
3	3	4	2	5	6
7	8	?	3	10	10
2	2	3	2	5	6

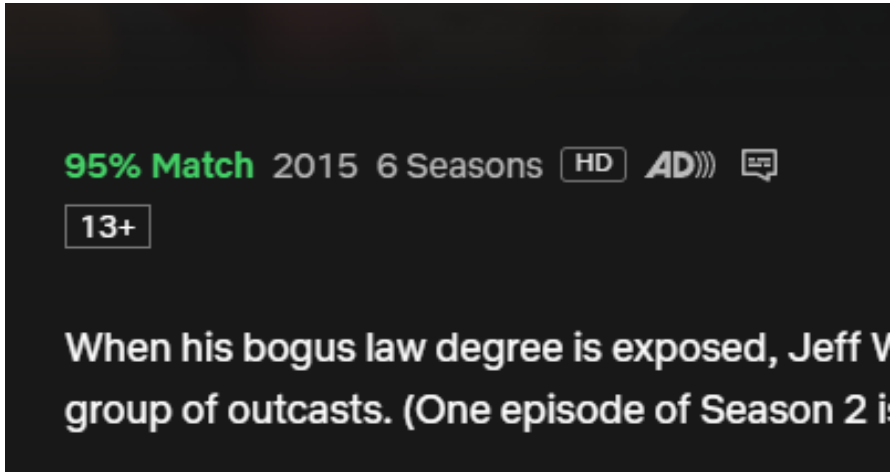
Collaborative filtering

Session-based filtering

Concluding remarks

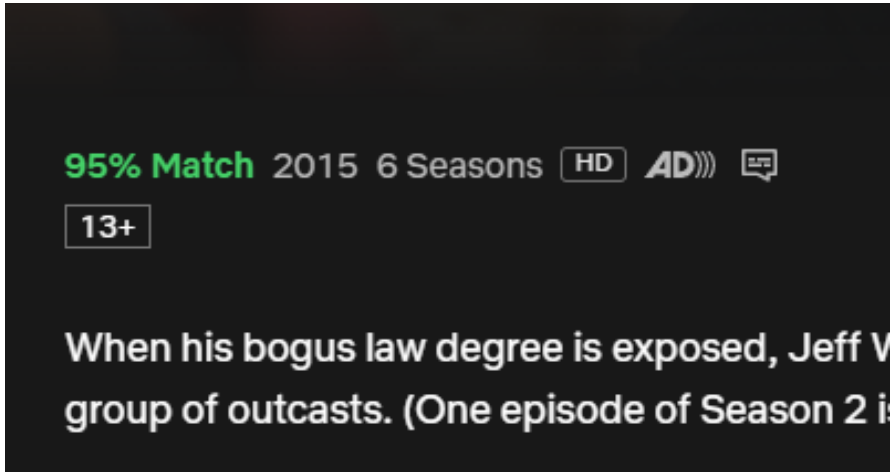
Recommendation systems can for example predict how well you will like something

Predicting ratings

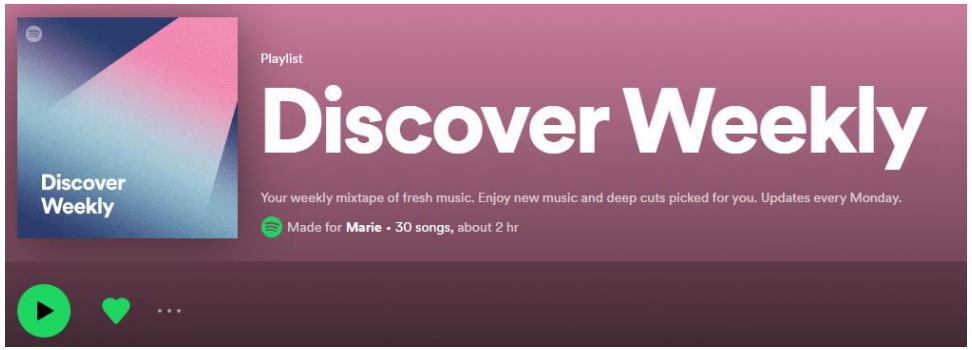


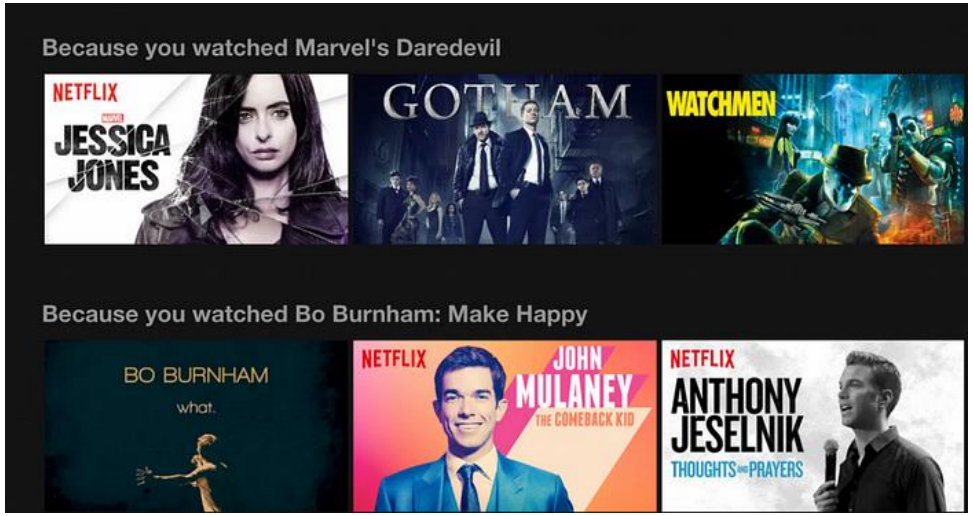
Recommendation systems can for example predict how well you will like something, or give you a list of suggestions for things you might like

Predicting ratings

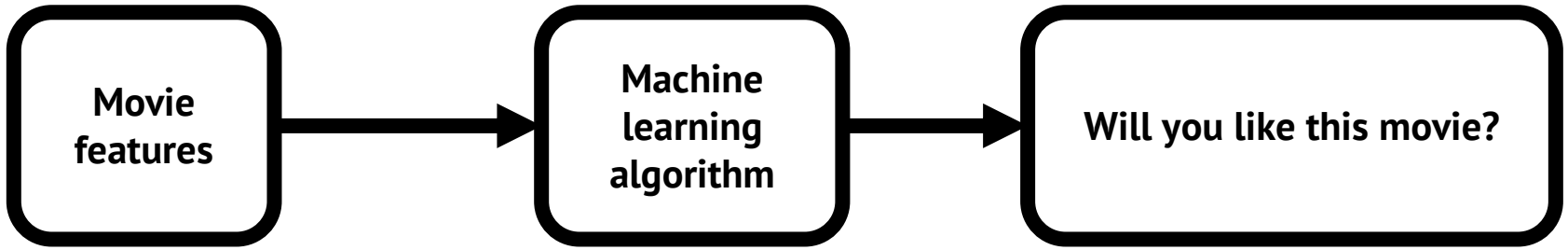


Providing suggestions



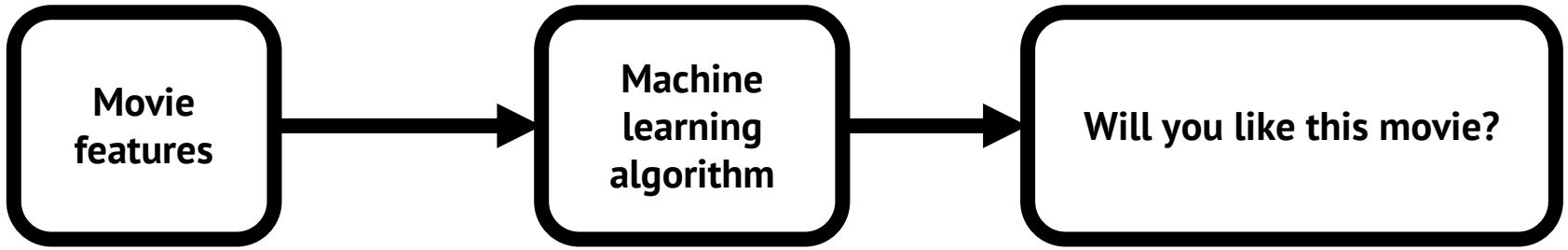


Content-based recommender systems makes recommendations based on features of the content



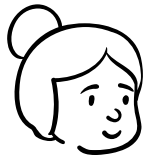
Content-based recommender systems makes recommendations based on features of the content

title
genre
actors
length
...



User features can be combined with item features to provide more personalized recommendations

title
genre
actors
length
...



age,
favourite
movie,
interests,
...

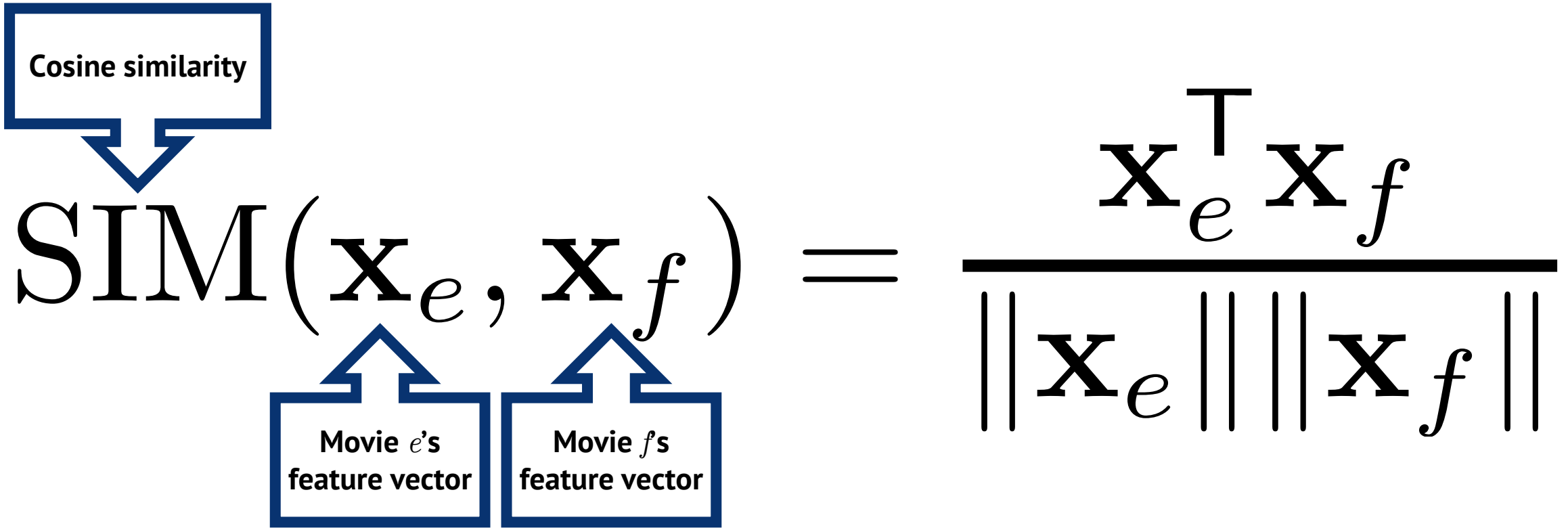
Movie features

User features

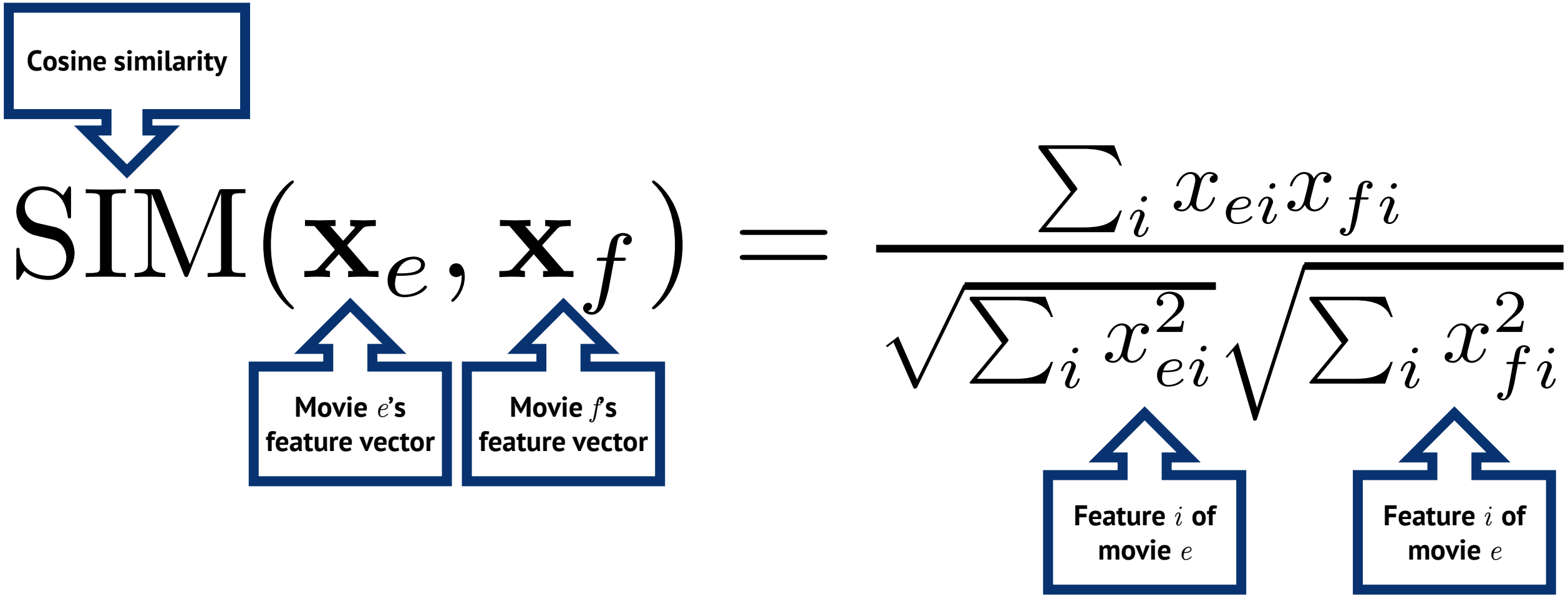
Machine learning algorithm

Will you like this movie?

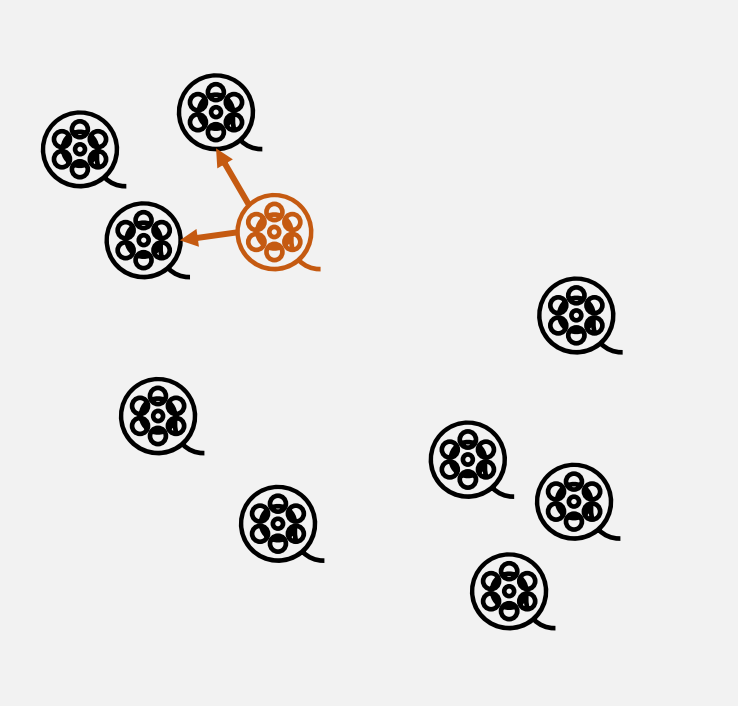
To find similar movies, we can use the *cosine similarity*



To find similar movies, we can use the *cosine similarity*

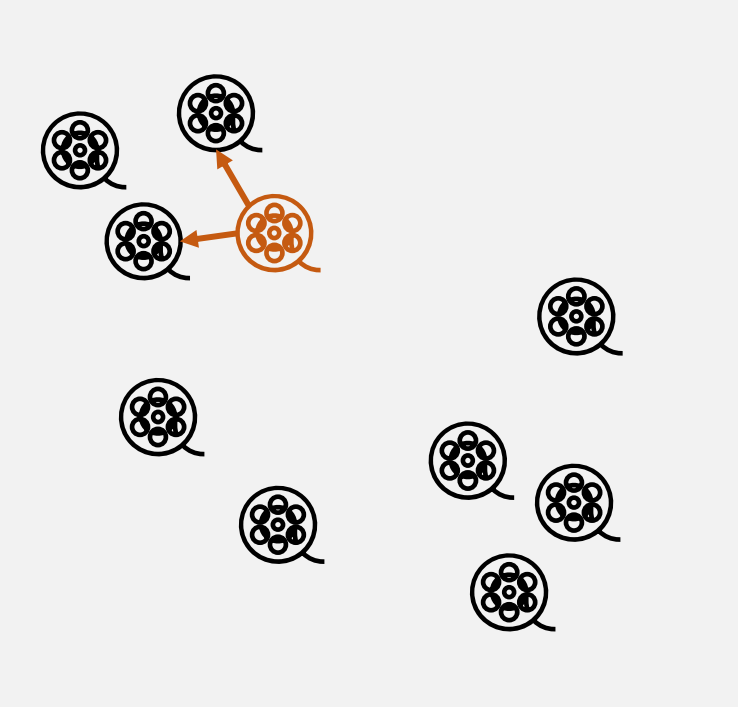


A common algorithm for content based recommendation is KNN



2-nearest neighbours (2NN)

Approximate KNN algorithms is often used when the dataset is very large



2-nearest neighbours (2NN)

≈

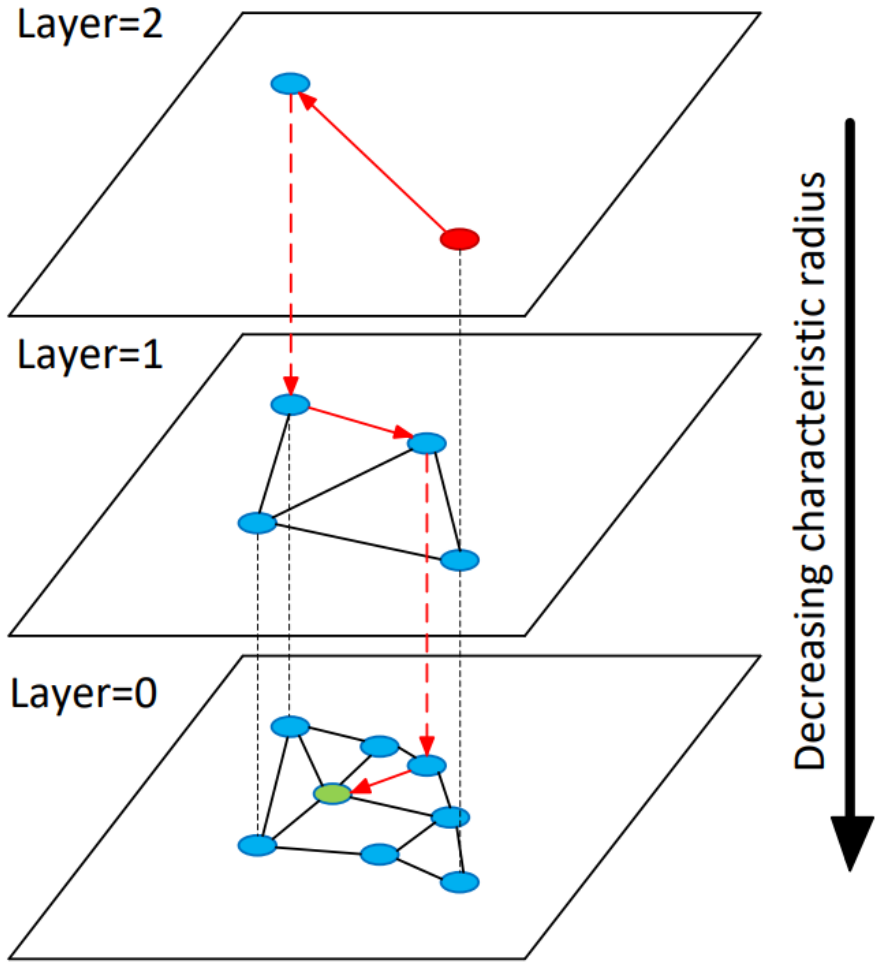
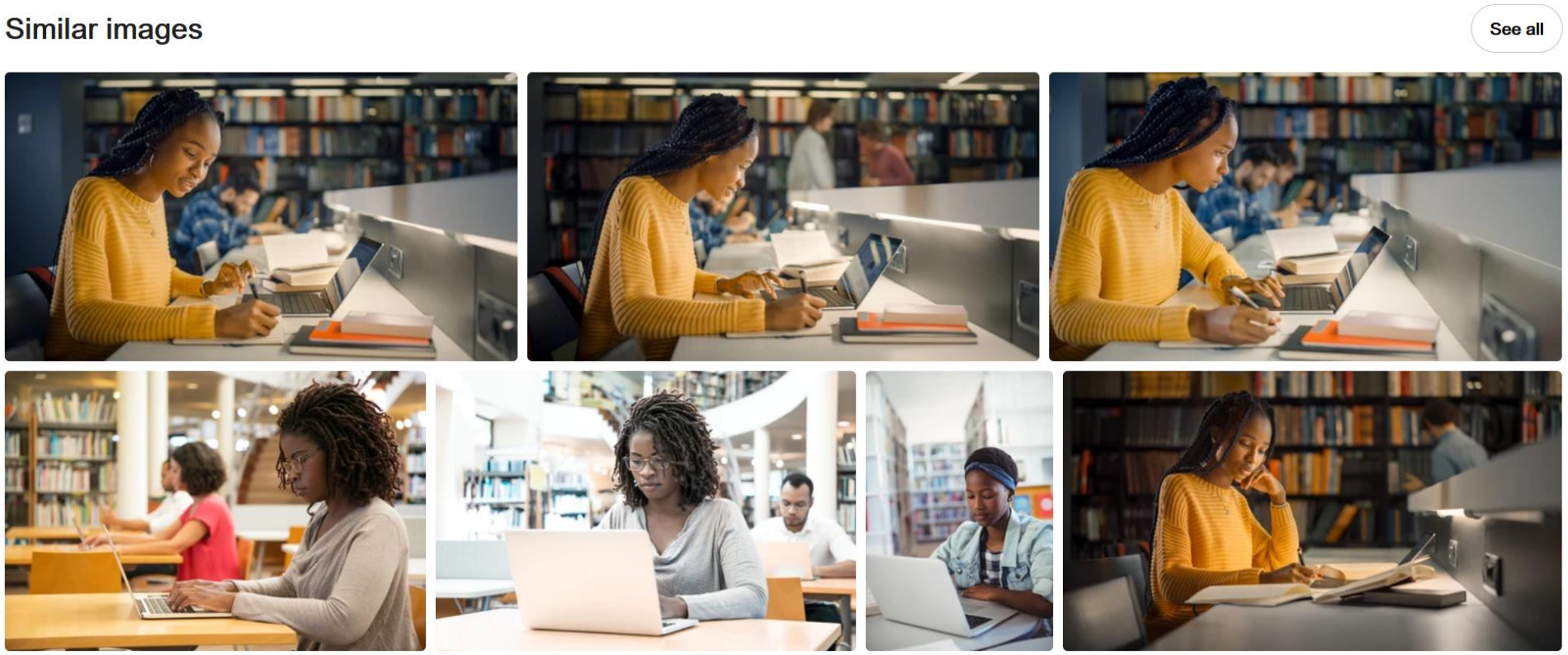


Figure from [Yu. A. Malkov, D. A. Yashunin. IEEE PAMI (2018)]

A key question of content filtering approaches is how to extract features



shutterstock®

[https://www.shutterstock.com/blog/data-science-building-content-personalization]

A key question of content filtering approaches is how to extract features

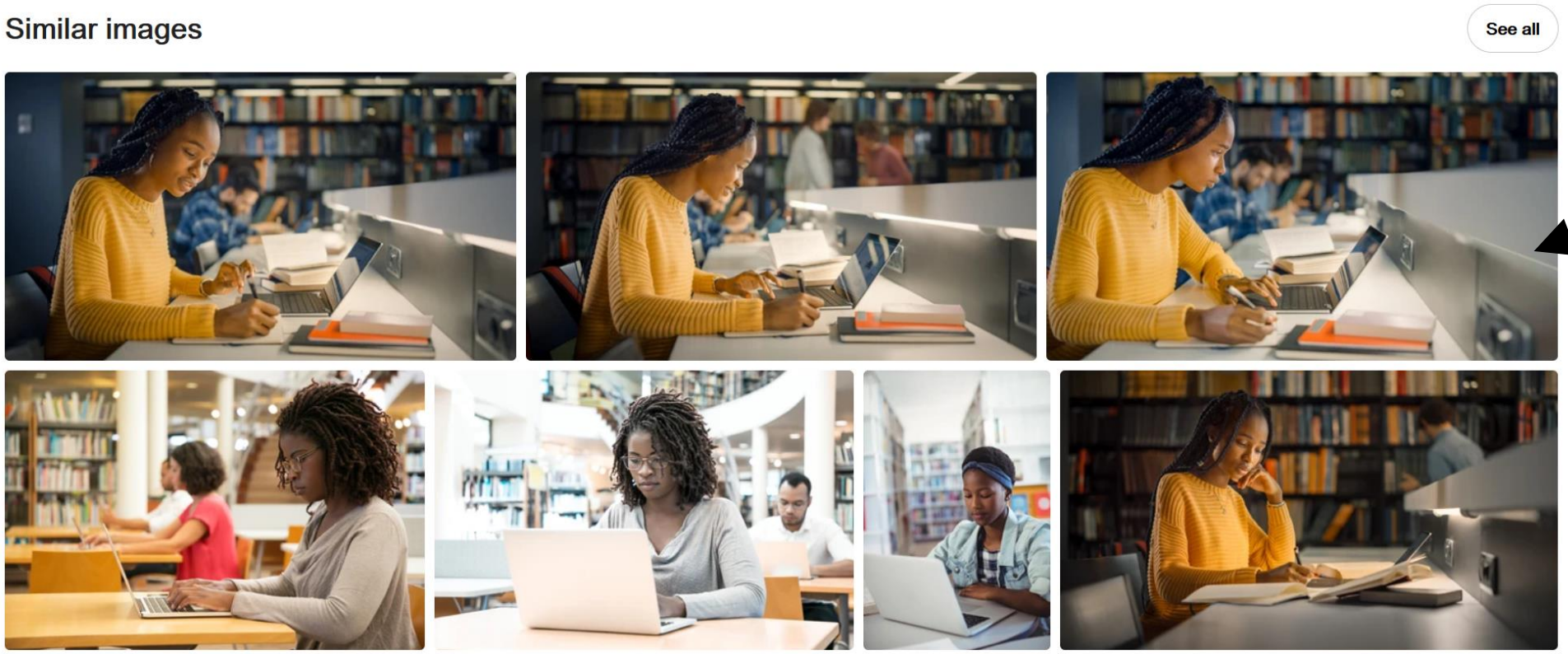


Image embeddings

shutterstock

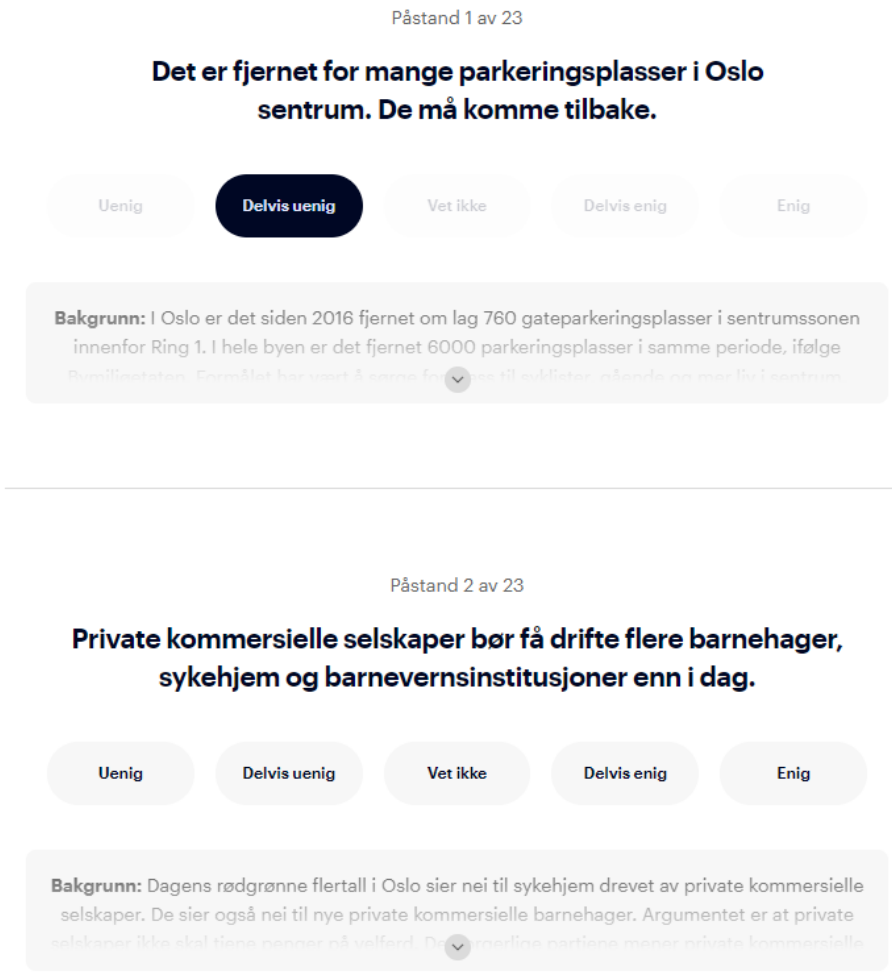
[https://www.shutterstock.com/blog/data-science-building-content-personalization]

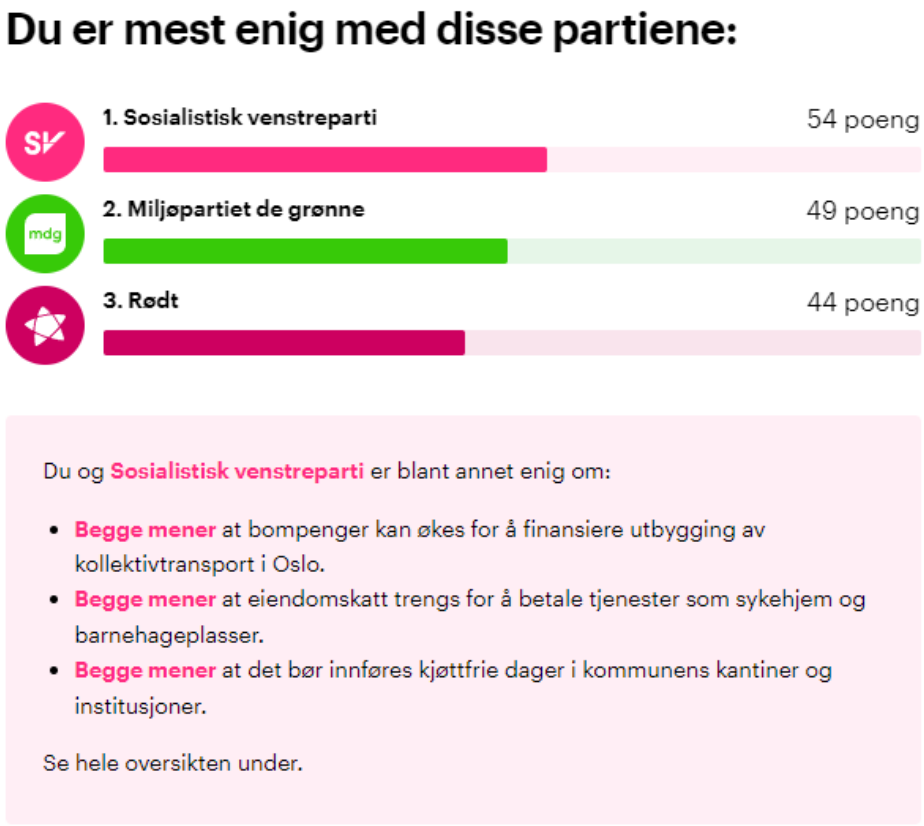
A key question of content filtering approaches is how to extract features

Image and document embeddings

Nasjonalbiblioteket
National Library of Norway

For some applications simple and interpretable features are ideal





Content-based filtering

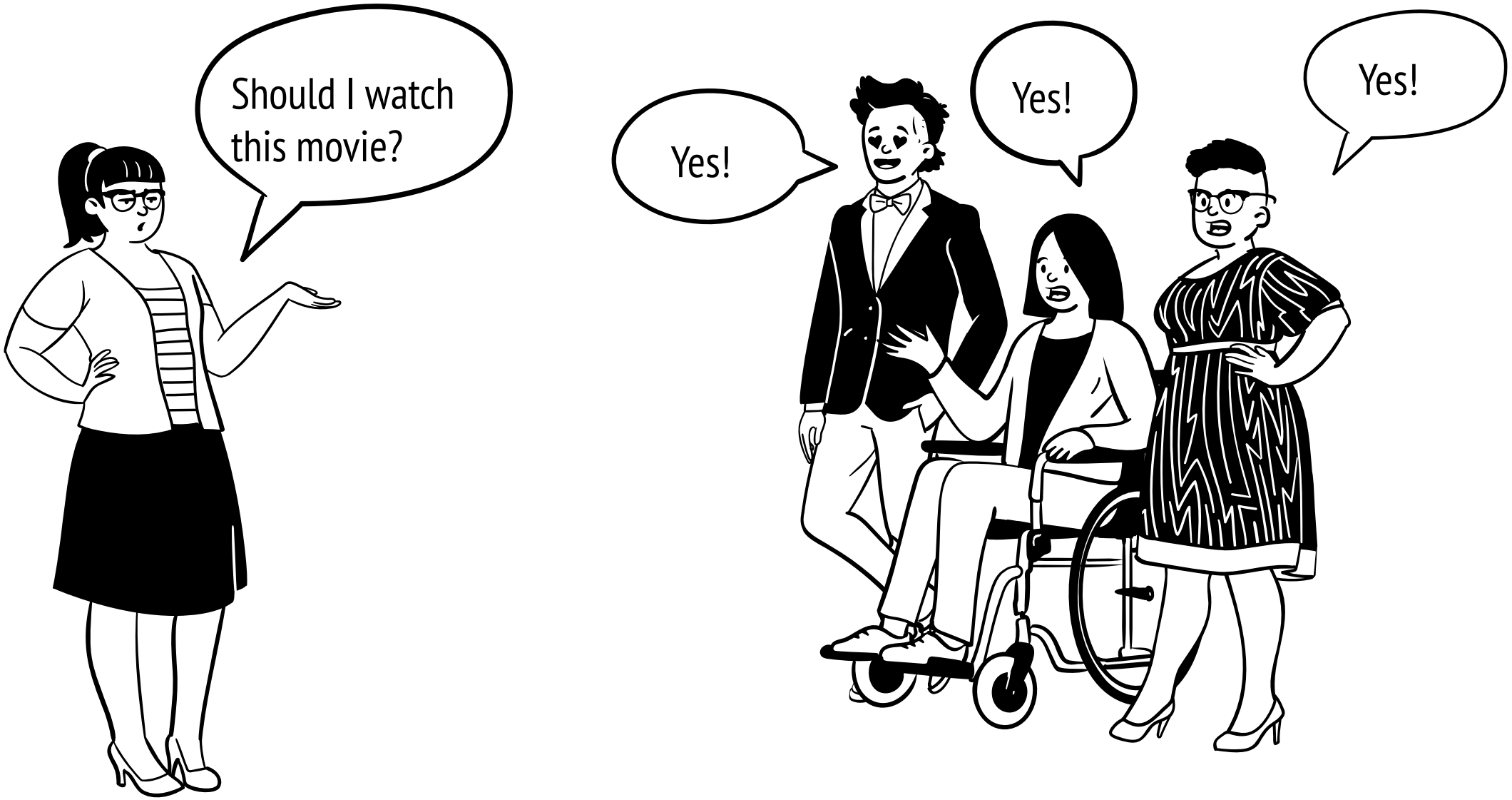
5	6	6	1	5	5
7	8	8	2	7	7
3	3	4	2	5	6
7	8	?	3	10	10
2	2	3	2	5	6

Collaborative filtering

Session-based filtering

Concluding remarks

Collaborative filtering uses similar users' ratings to predict if you will enjoy a movie




Raise your hand if you have seen
“Kill Bill” and **really liked it!**


Raise your hand if you have seen
“Kill Bill” and **didn’t like it!**

I can use the average rating as an estimate for how much I will like a movie

Estimate of my rating of the movie

User u's rating of the movie

$$\hat{x}_{ei} = \frac{1}{U} \sum_{u=1}^{U} x_{eu}$$

I can weight other users' ratings based on how similar their taste is to mine

Estimate of my rating of the movie

User u's rating of the movie

$$\hat{x}_{ei} = \frac{1}{N} \sum_{u=1}^{U} x_{eu} \times \mathrm{SIM}(\mathbf{x}_{:,v}, \mathbf{x}_{:,u})$$

I can weight other users' ratings based on how similar their taste is to mine

Estimate of my rating of the movie

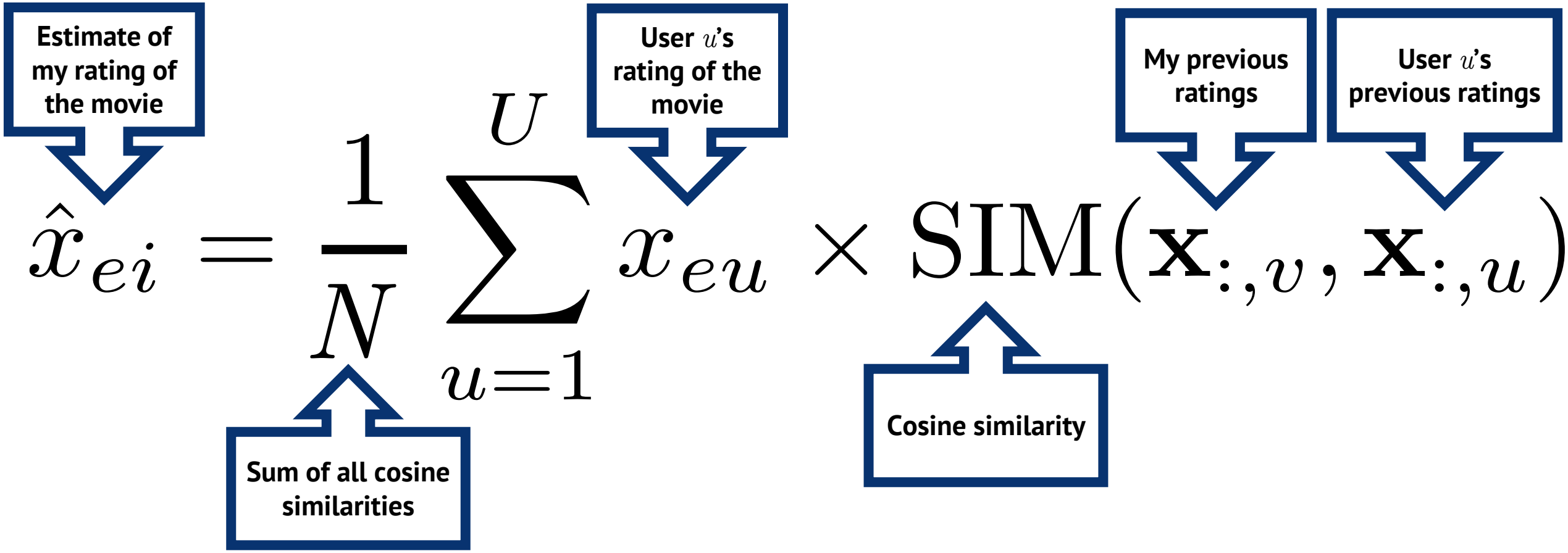
User u's rating of the movie

My previous ratings

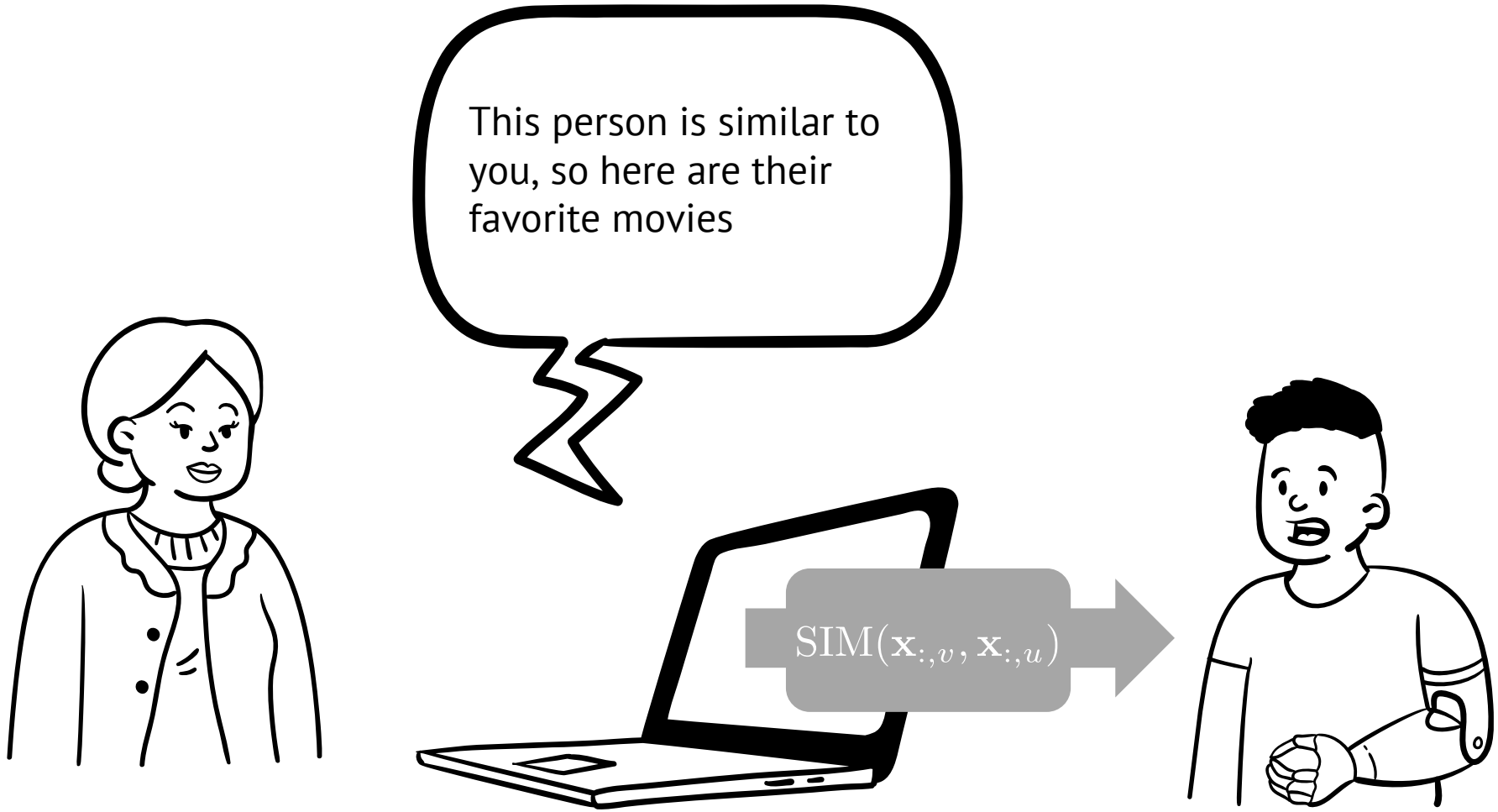
User u's previous ratings

$$\hat{x}_{ei} = \frac{1}{N} \sum_{u=1}^{U} x_{eu} \times \mathrm{SIM}(\mathbf{x}_{:,v}, \mathbf{x}_{:,u})$$

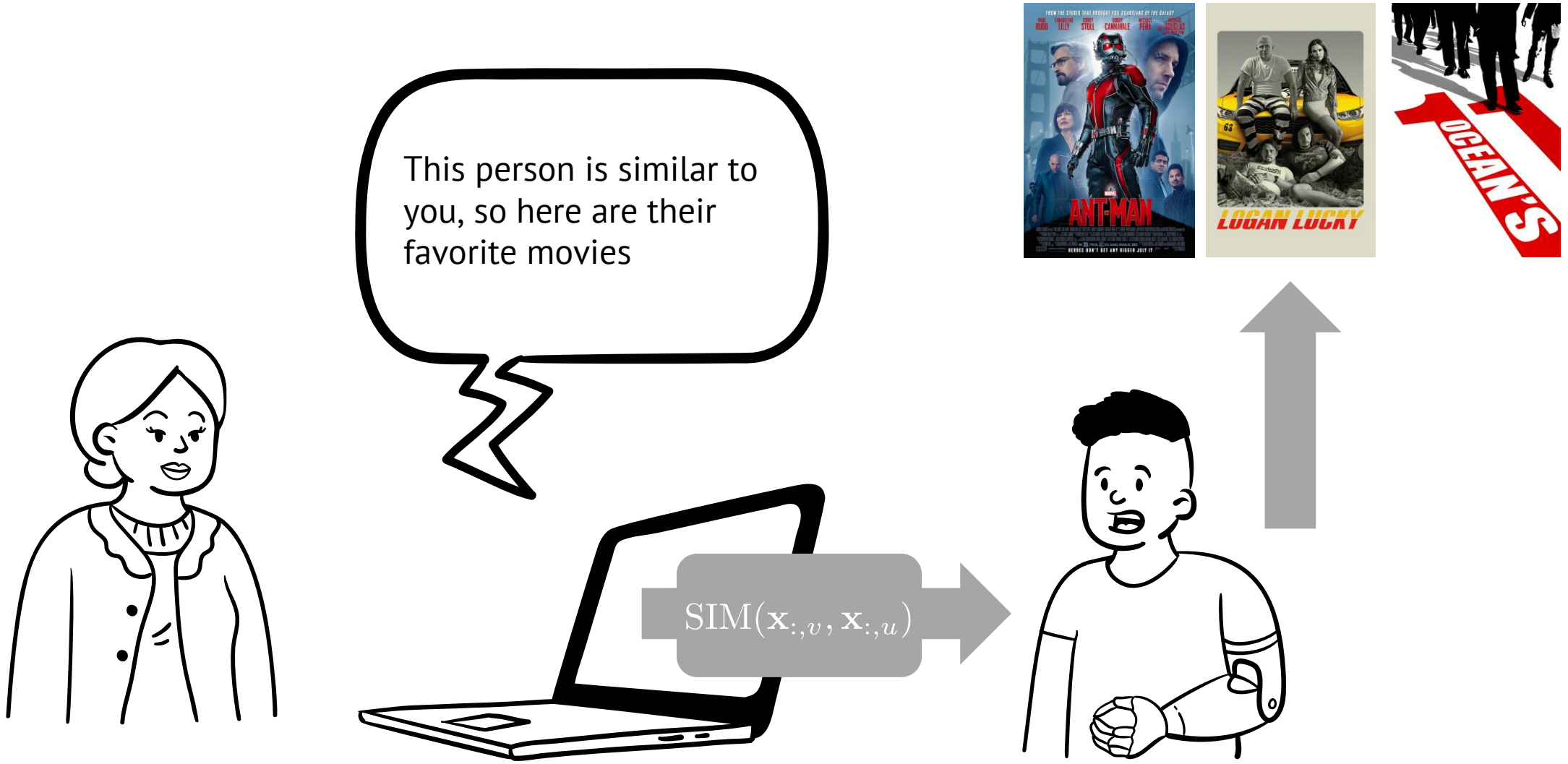
I can weight other users' ratings based on how similar their taste is to mine



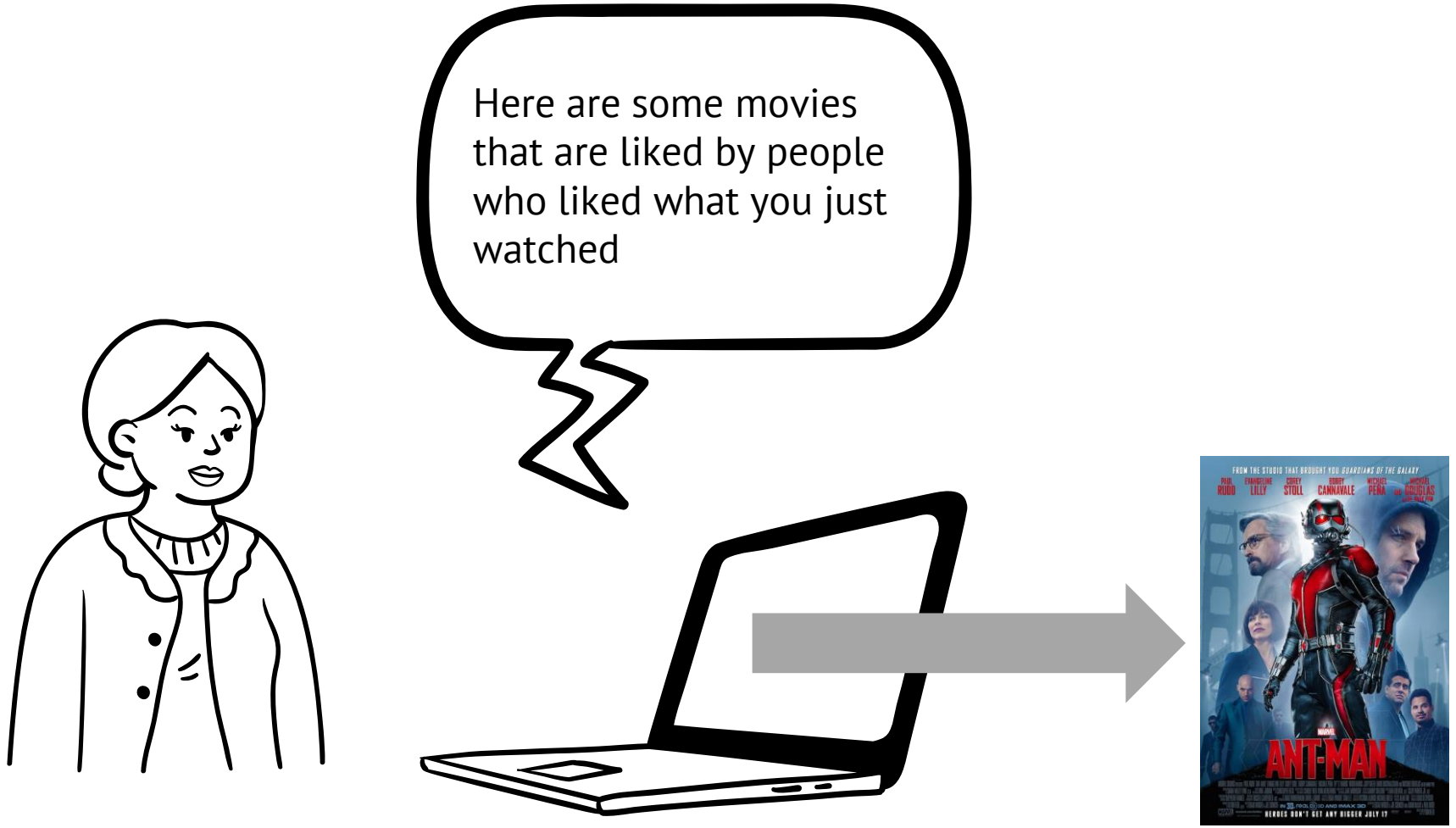
We can also use similarity recommend movies liked by users with similar taste as me



We can also use similarity recommend movies liked by users with similar taste as me



Or use similarity to recommend movies liked by people who also like what you just watched



Or use similarity to recommend movies liked by people who also like what you just watched

Model based collaborative filtering finds a mathematical model that describe the movie ratings

Matrix factorisations can find underlying factors that describe movie ratings

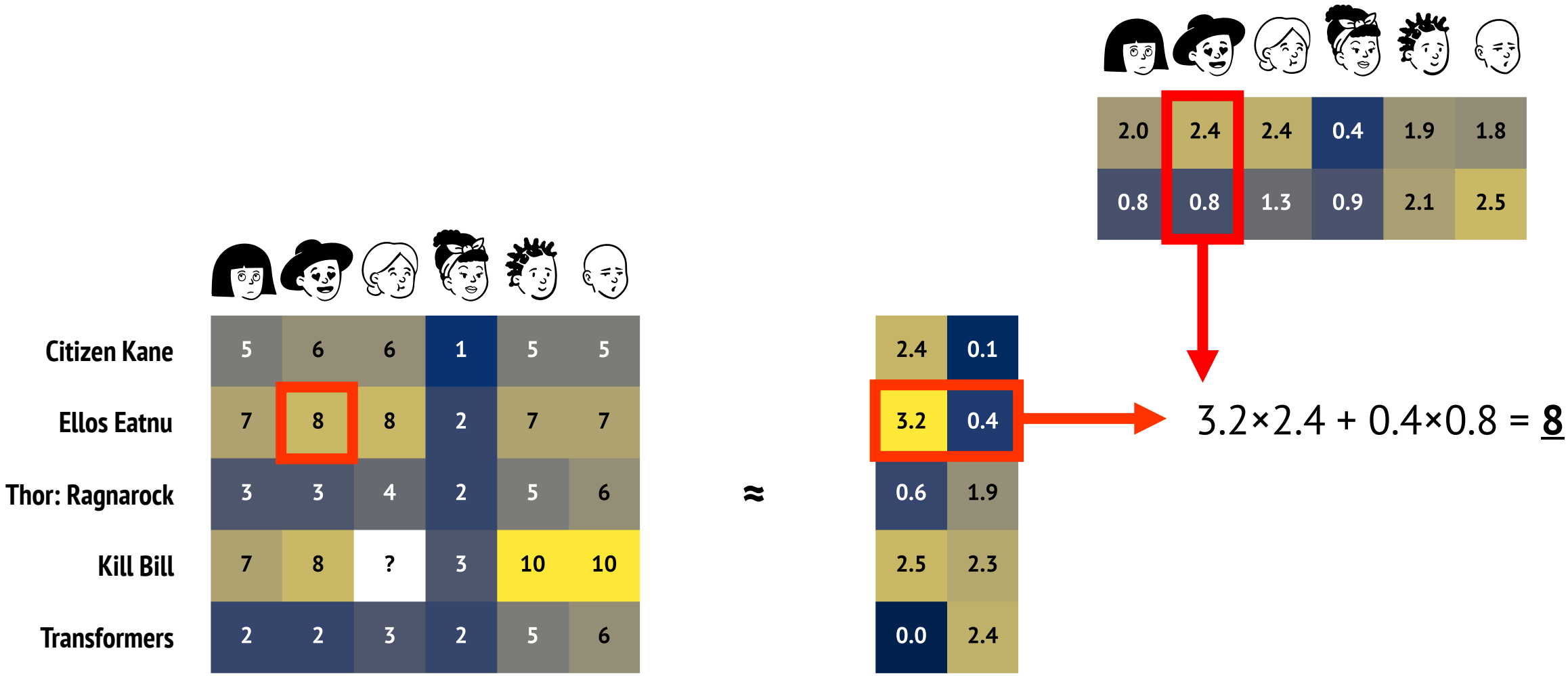
Citizen Kane	5	6	6	1	5	5
Ellos Eatnu	7	8	8	2	7	7
Thor: Ragnarock	3	3	4	2	5	6
Kill Bill	7	8	?	3	10	10
Transformers	2	2	3	2	5	6

Matrix factorisations can find underlying factors that describe movie ratings

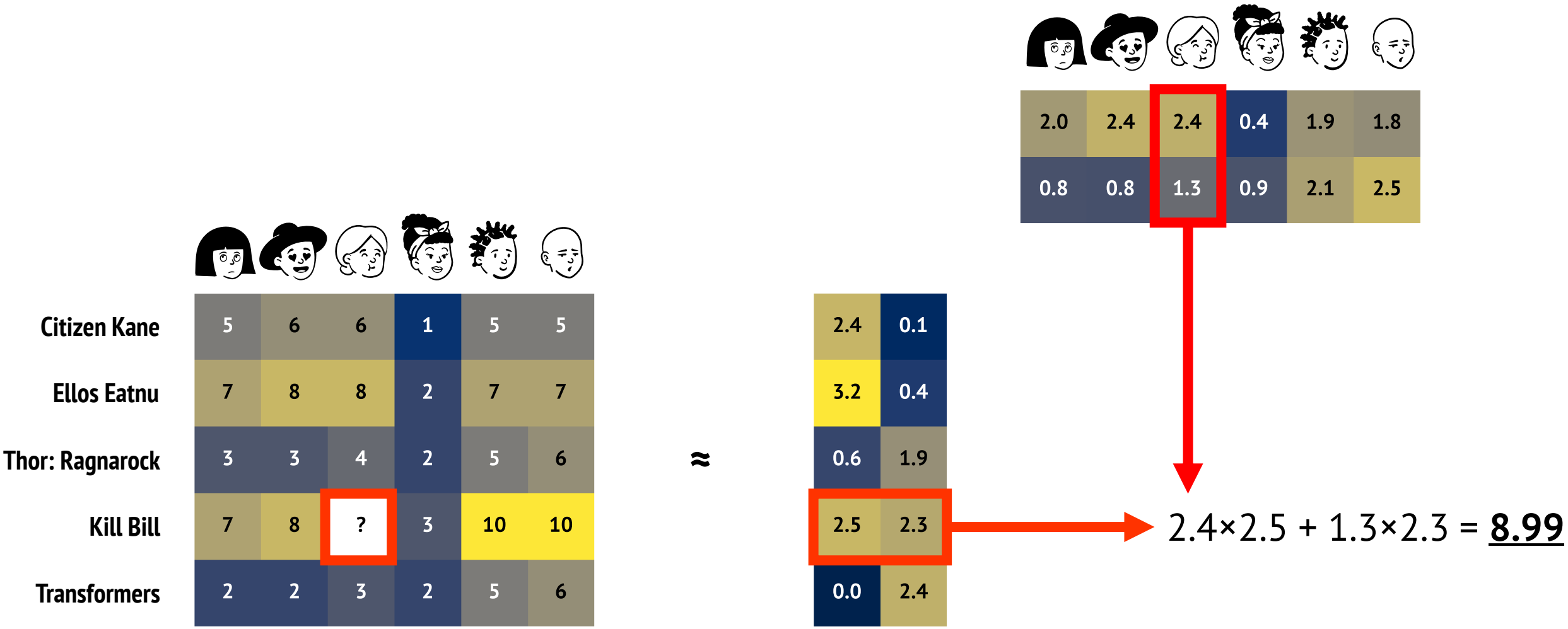
	2.0	2.4	2.4	0.4	1.9	1.8
	0.8	0.8	1.3	0.9	2.1	2.5

Citizen Kane	5	6	6	1	5	5		2.4	0.1
Ellos Eatnu	7	8	8	2	7	7		3.2	0.4
Thor: Ragnarock	3	3	4	2	5	6	≈	0.6	1.9
Kill Bill	7	8	?	3	10	10		2.5	2.3
Transformers	2	2	3	2	5	6		0.0	2.4

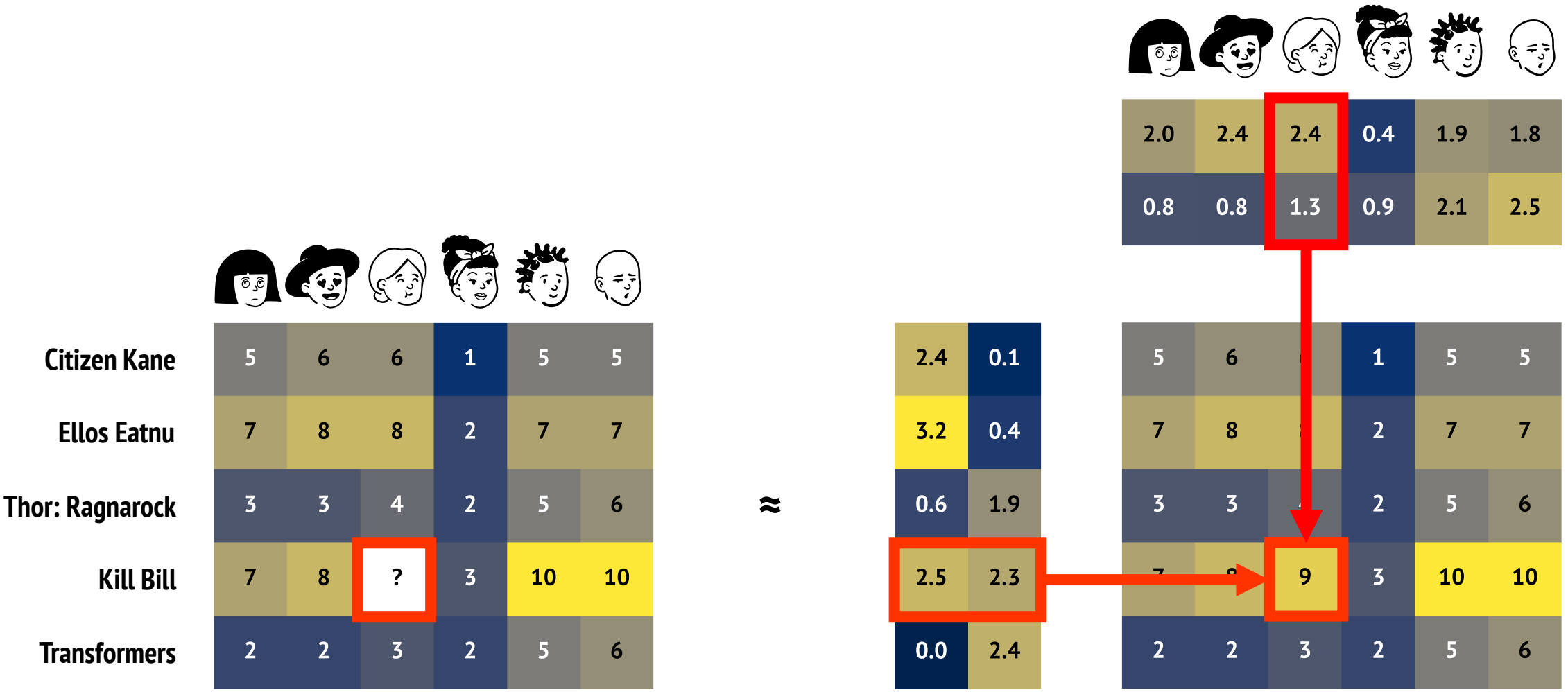
Matrix factorisations can find underlying factors that describe movie ratings



Matrix factorisations can find underlying factors that describe movie ratings



Matrix factorisations can find underlying factors that describe movie ratings



$$\mathbf{X} \approx \mathbf{E}\mathbf{U}^{\top}$$

To find the embeddings, we minimise the squared reconstruction error

$$\min_{\mathbf{E},\mathbf{U}} \sum_{e,u} (x_{eu} - \mathbf{e}_{e,:}^{\mathsf{T}} \mathbf{u}_{u,:})^2$$

We often add regularisation to reduce the chance of overfitting

Minimise error for known movie scores

Product of embedding vectors

Regularisation to avoid overfitting

Regularisation to avoid overfitting

$$\min_{\mathbf{E},\mathbf{U}} \sum_{e,u} (x_{eu} - \mathbf{e}_{e,:}^{\mathsf{T}} \mathbf{u}_{u,:})^2 + \gamma_e \|\mathbf{E}\|_F^2 + \gamma_u \|\mathbf{U}\|_F^2$$

We often add regularisation to reduce the chance of overfitting

Minimise error for known movie scores

Product of embedding vectors

Regularisation to avoid overfitting

Regularisation to avoid overfitting

$$\min_{\mathbf{E},\mathbf{U}} \sum_{e,u} (x_{eu} - \mathbf{e}_{e,:}^{\mathsf{T}} \mathbf{u}_{u,:})^2 + \gamma_e \|\mathbf{E}\|_F^2 + \gamma_u \|\mathbf{U}\|_F^2$$

$$\text{s.t. } \mathbf{E},\mathbf{U} \geq 0 \text{ elementwise}$$

Nonnegativity to reduce overfitting and improve interpretability

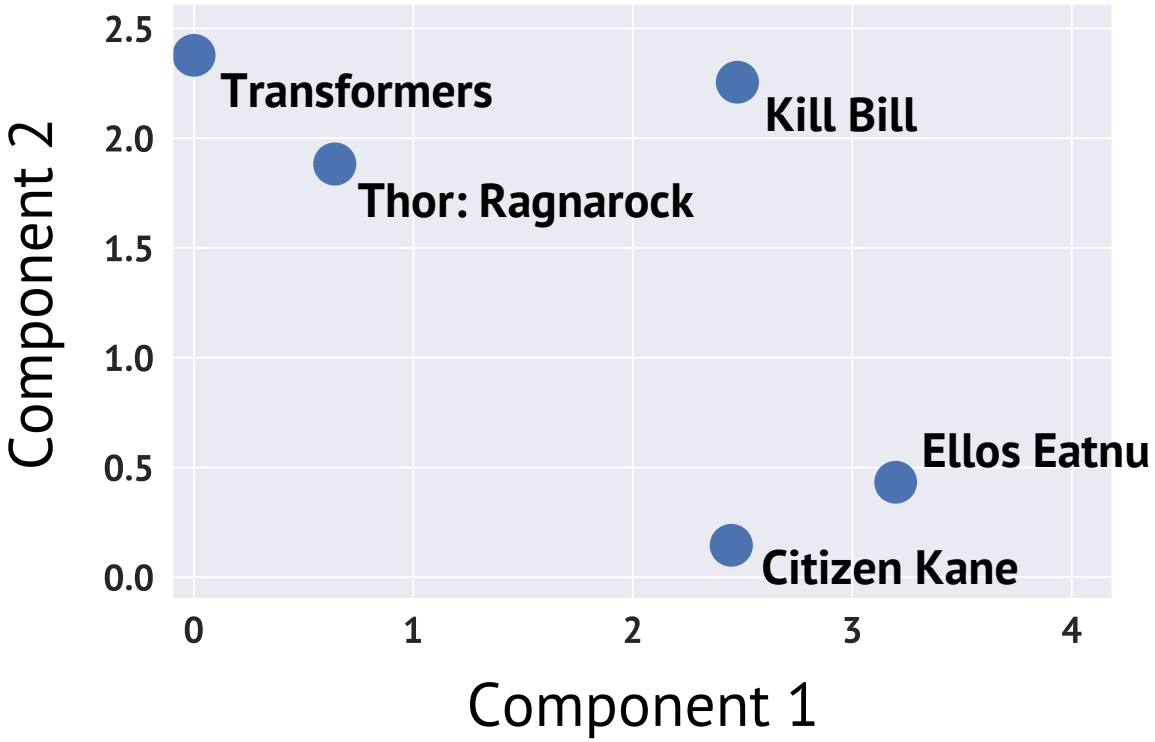
We can also use nonlinear embeddings, e.g., by fitting neural networks to reconstruct our data

Parameters of the learned function

$$\min_{\mathbf{E},\mathbf{U},\mathcal{W}} \sum_{e,u} (x_{eu} - f(\mathbf{e}_{e,:}, \mathbf{u}_{u,:}; \mathcal{W}))^2 + \gamma_e \|\mathbf{E}\|_F^2 + \gamma_u \|\mathbf{U}\|_F^2$$

Learned function to reconstruct the movie scores from embeddings

The latent factors embed movies in a space where similar movies are close



The latent factors embed movies in a space where similar movies are close

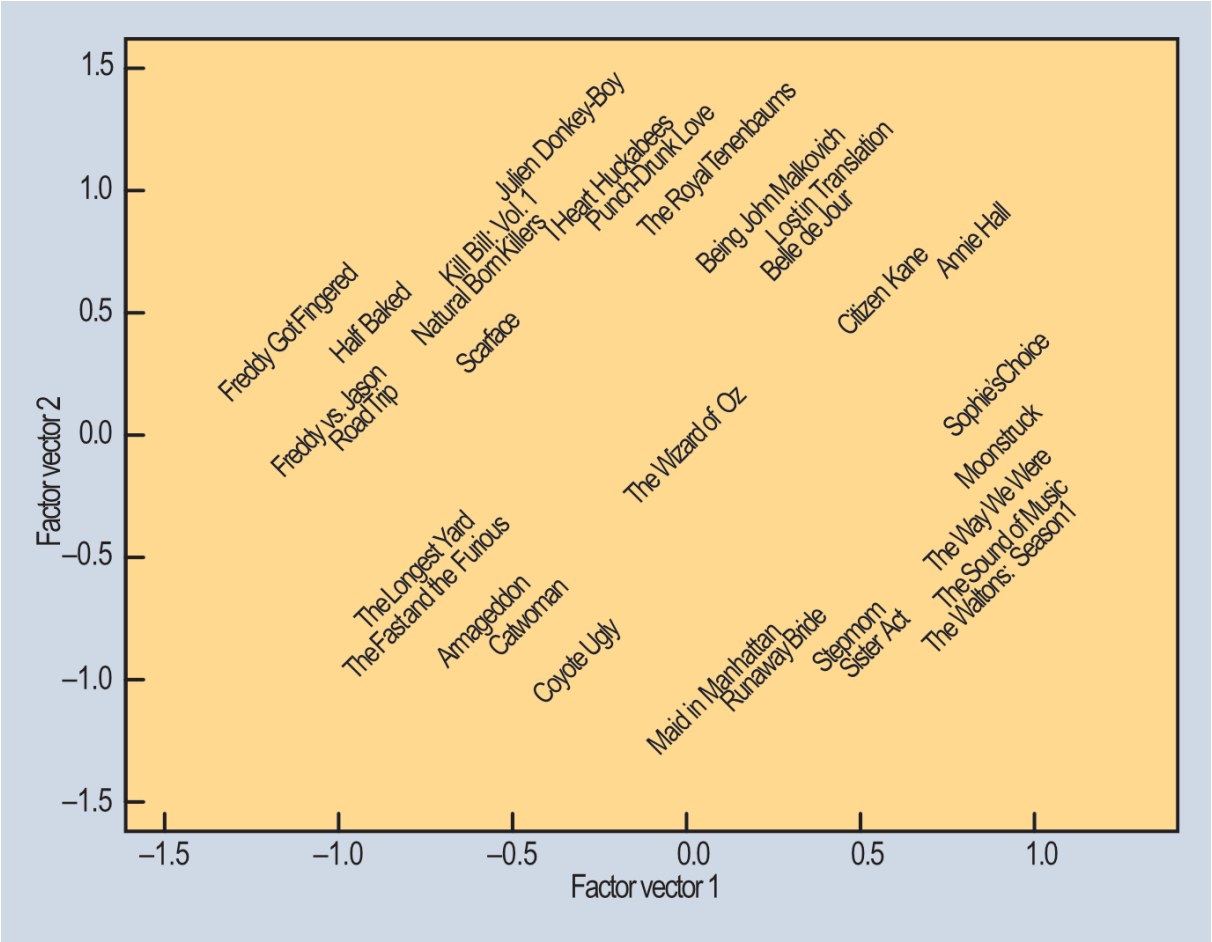


Figure from [Koren, Bell & Volinsky. IEEE Computer (2009)]

The latent factors embed movies in a space where similar movies are close

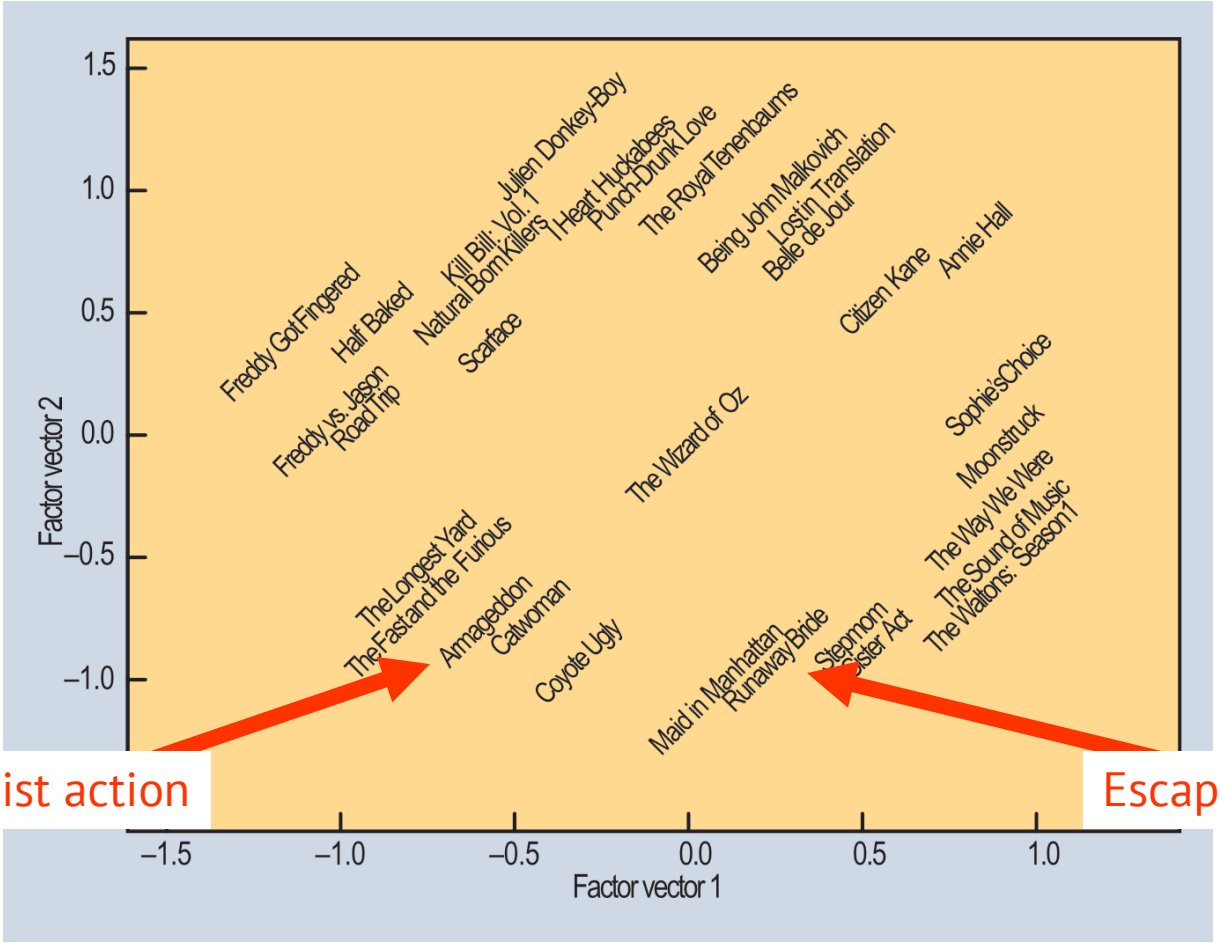
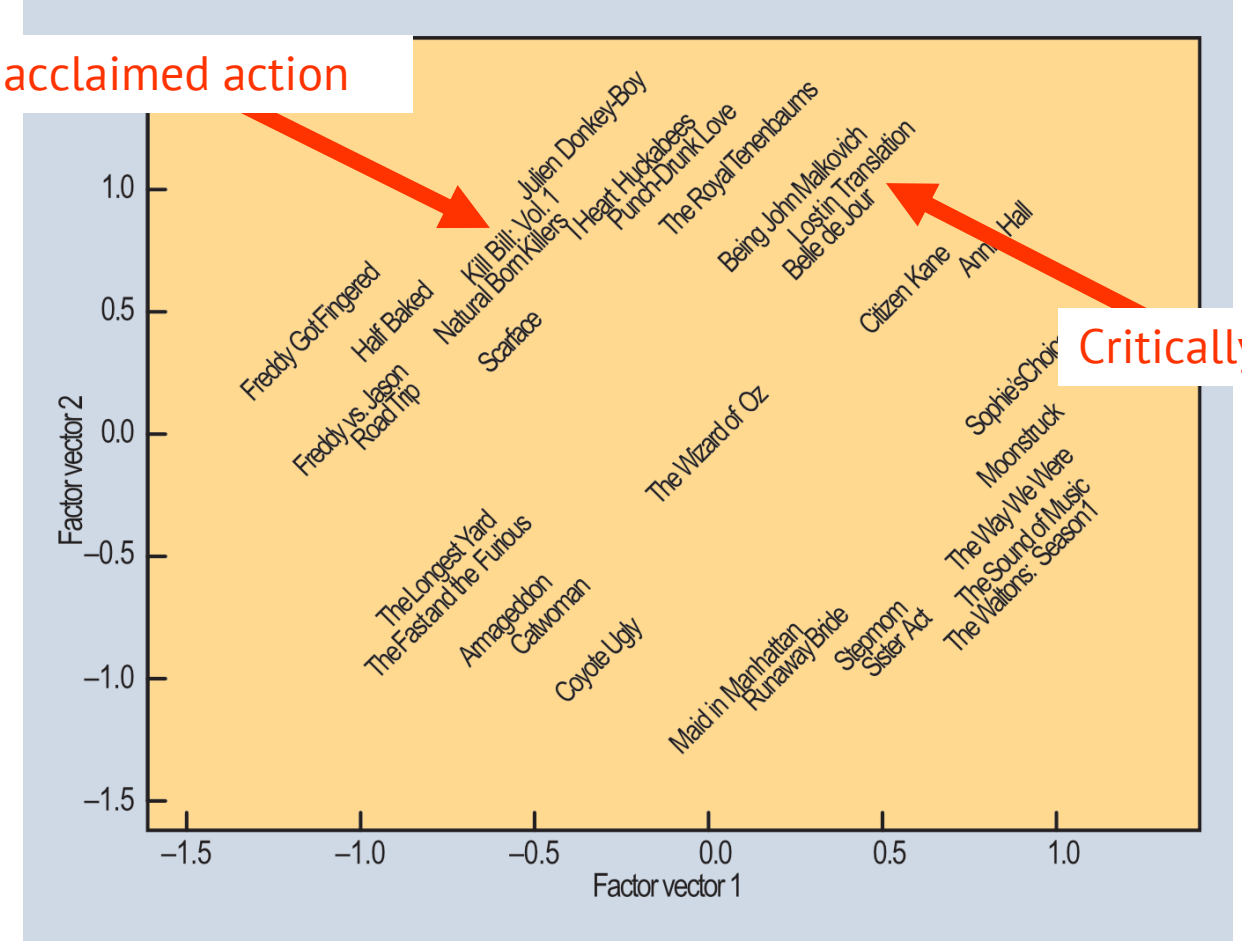


Figure from [Koren, Bell & Volinsky. IEEE Computer (2009)]

The latent factors embed movies in a space where similar movies are close

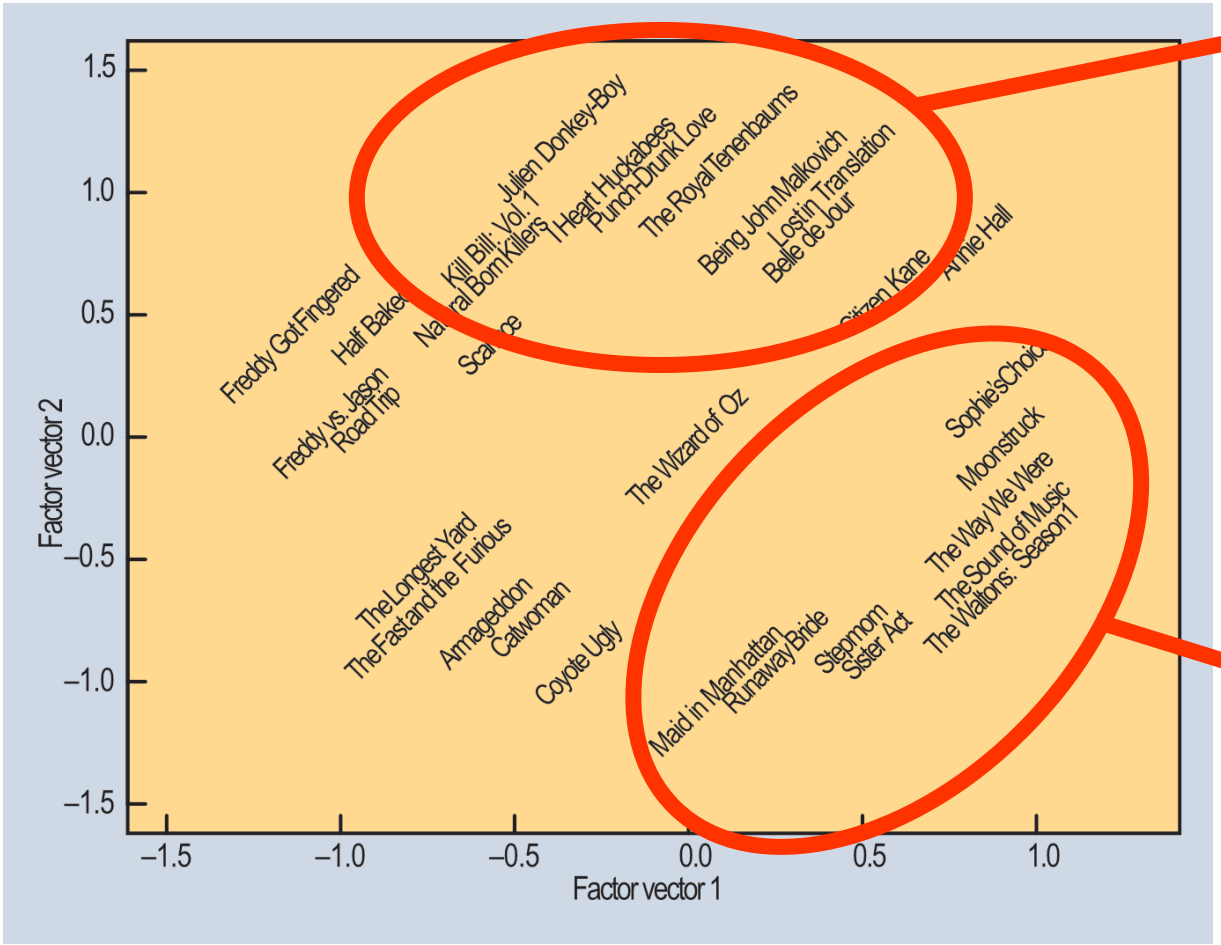
Critically acclaimed action

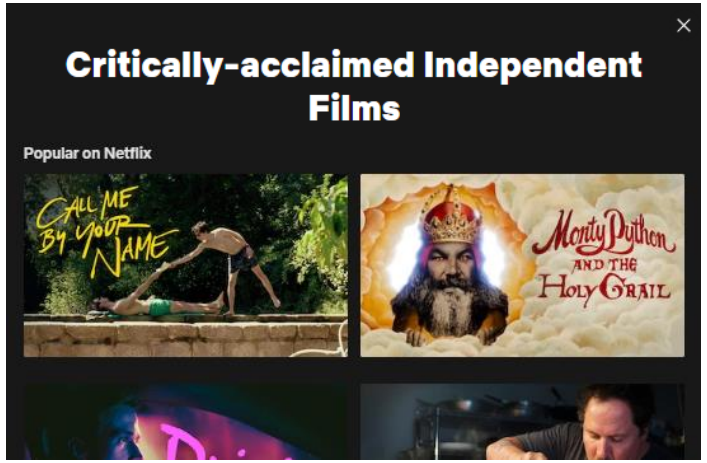


Critically acclaimed comedy/romance

Figure from [Koren, Bell & Volinsky. IEEE Computer (2009)]

The latent factors embed movies in a space where similar movies are close





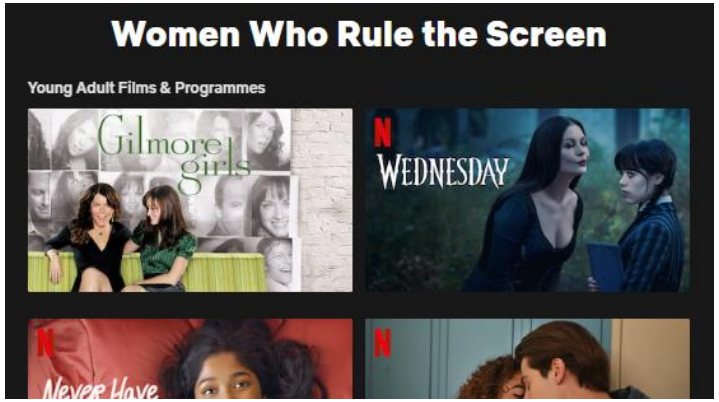
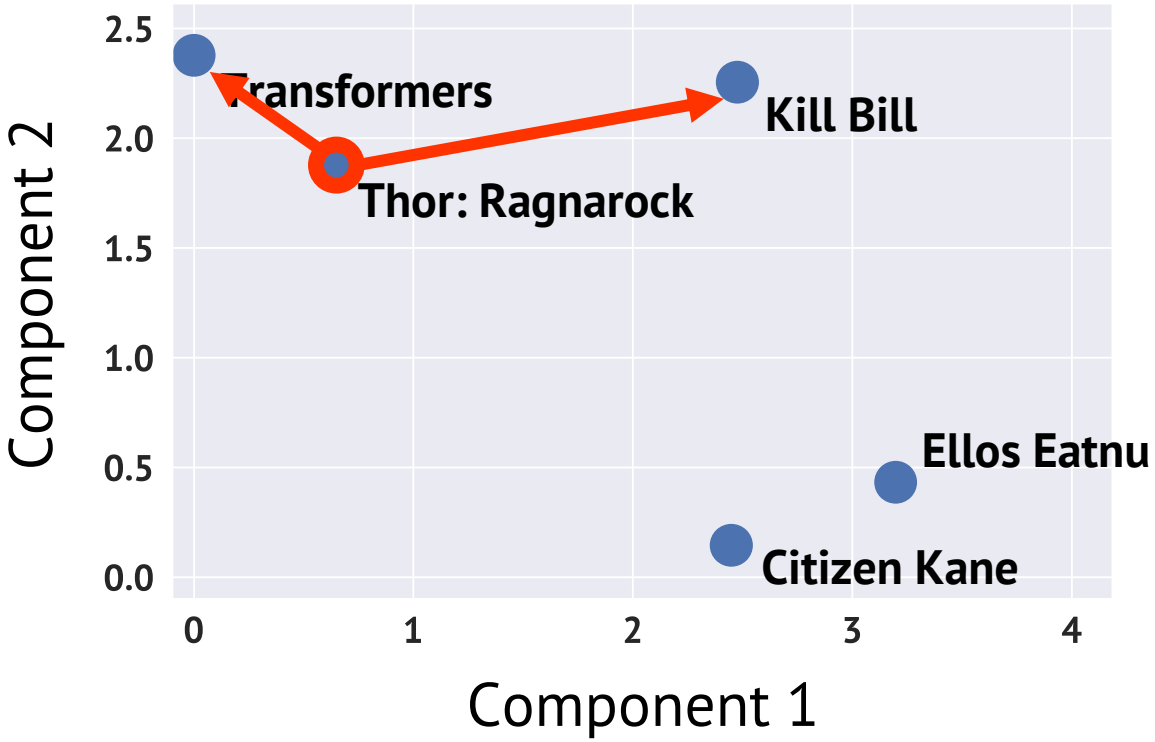


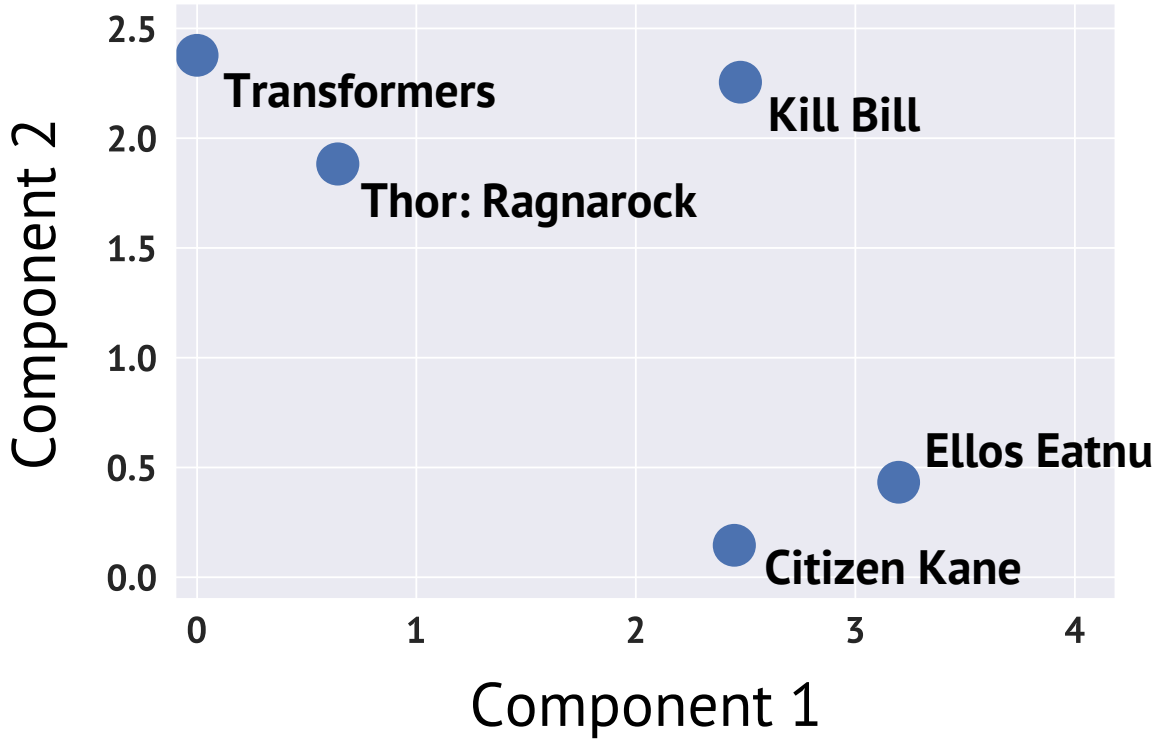
Figure from [Koren, Bell & Volinsky. IEEE Computer (2009)]

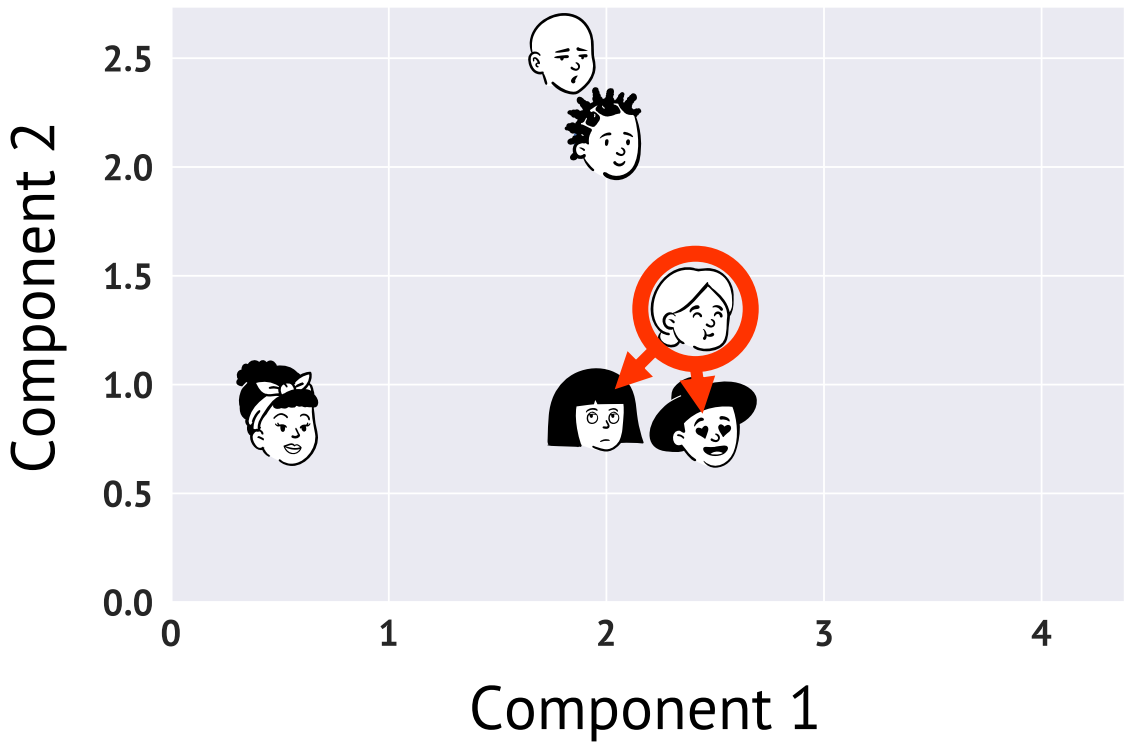
The latent factors embed movies in a space where similar movies are close



The embeddings can be used to recommend similar movies

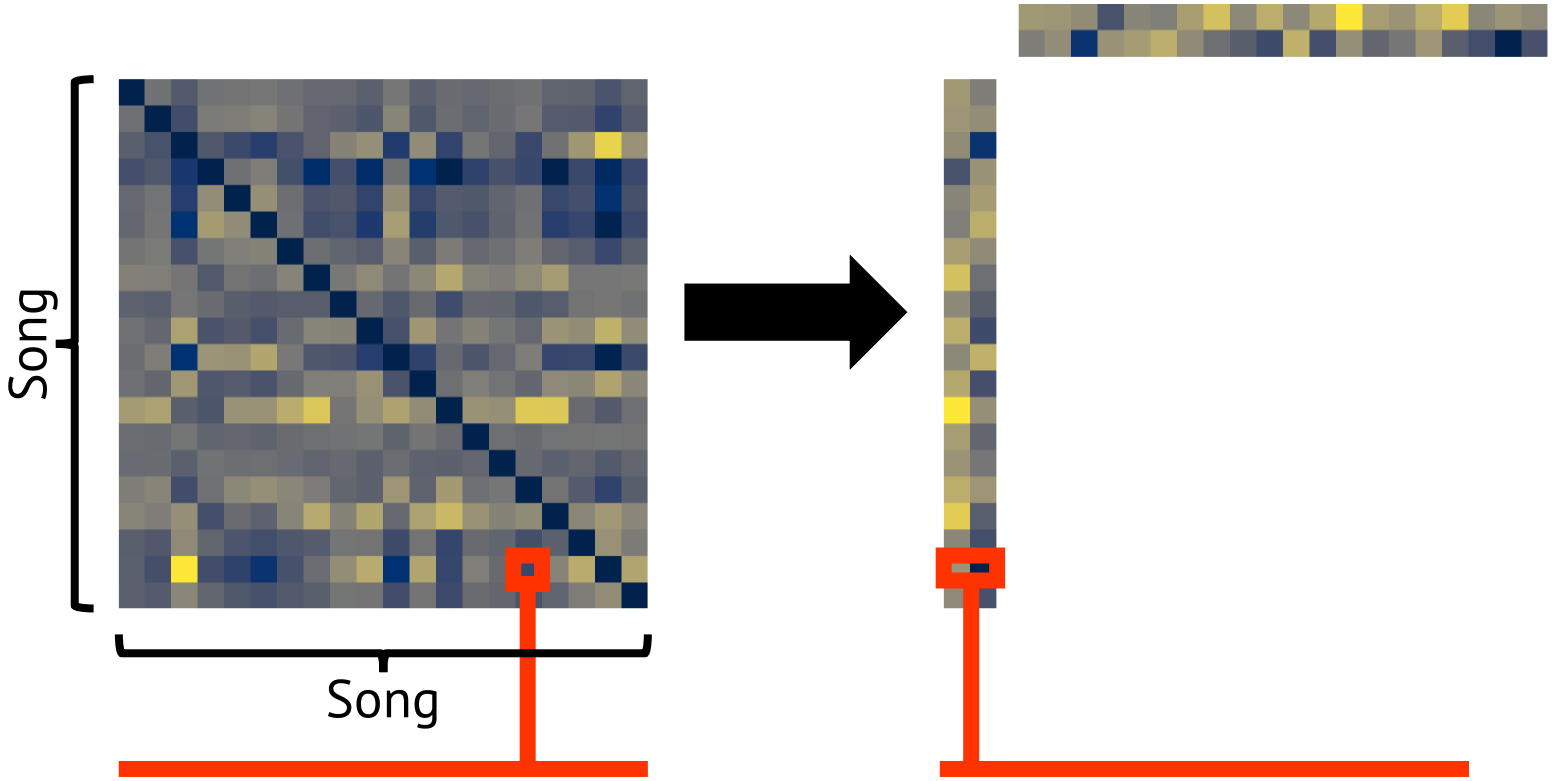
The latent factors embed movies in a space where similar movies are close and where similar users are close





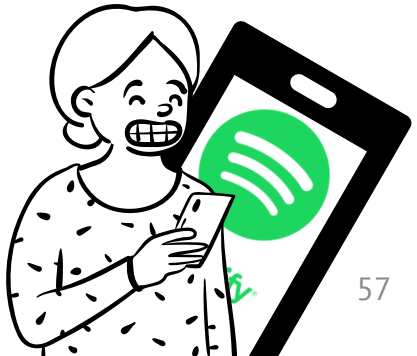
The embeddings can be used to recommend movies liked by similar users

Spotify used nonlinear song embeddings to find similar songs



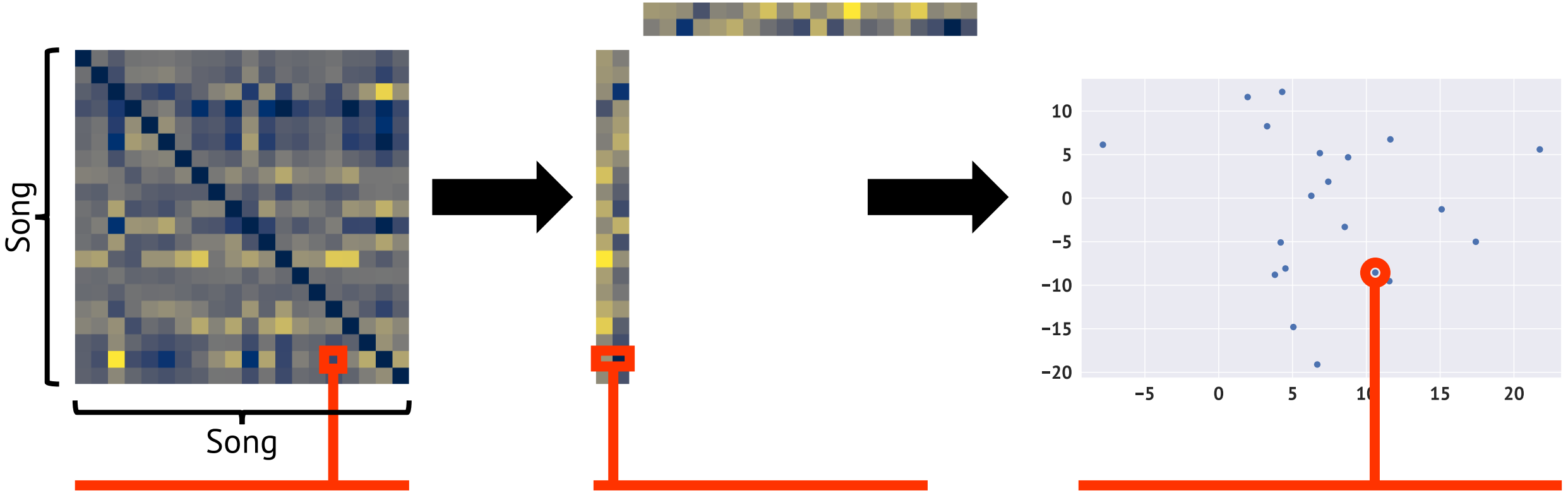
How often does song *i* and *j* occur together

Word2Vec finds embedding for song *i*



[Hansen et al. ACM RecSys (2020)]

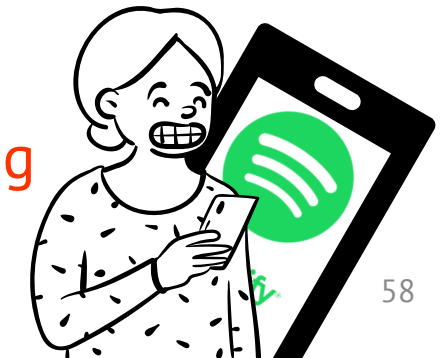
Spotify used nonlinear song embeddings to find similar songs



How often does song *i* and *j* occur together

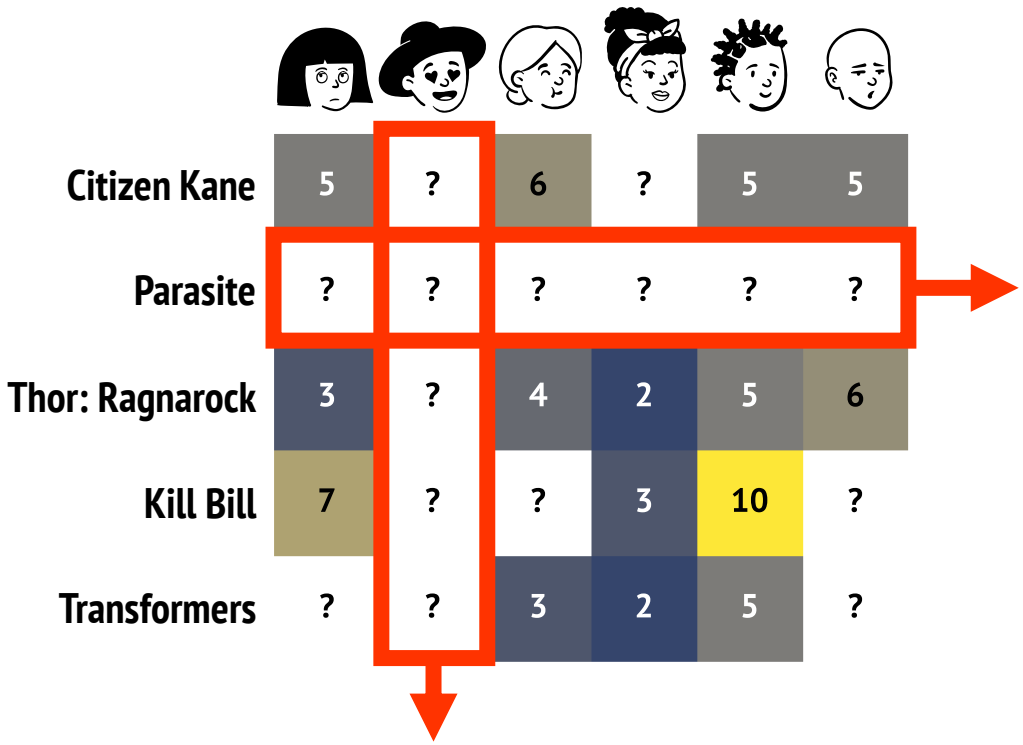
Word2Vec finds embedding for song *i*

Similar songs are close in the embedding space



[Hansen et al. ACM RecSys (2020)]

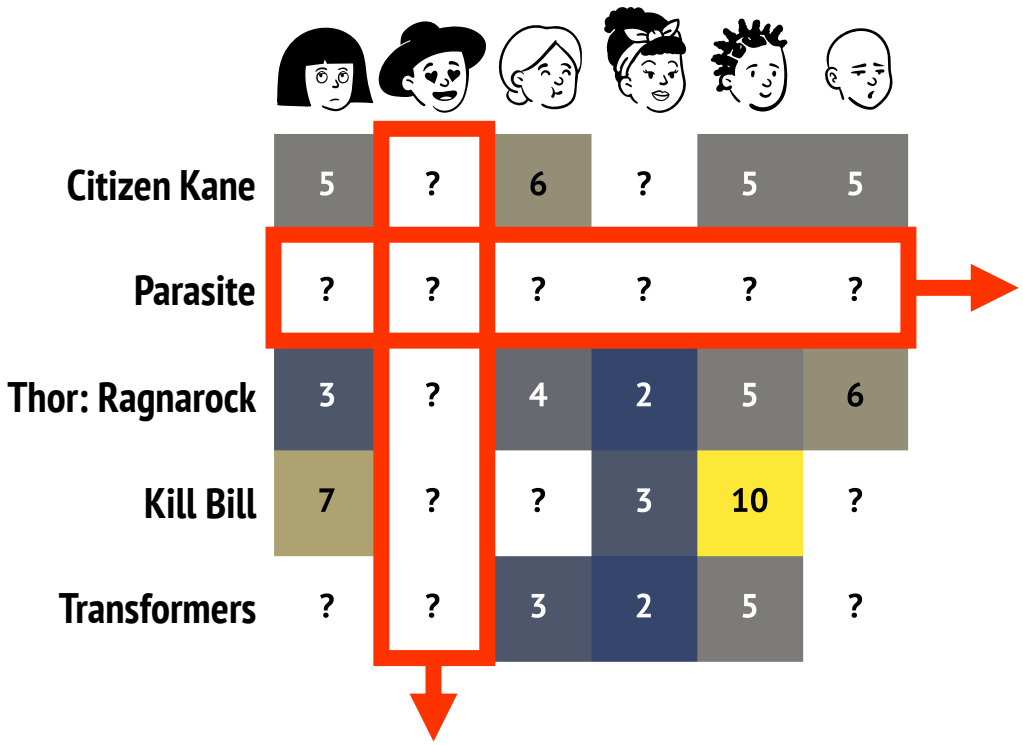
Collaborative filtering struggles with the *cold start problem*: it's difficult to provide recommendations for new movies and users



A **new movie** has no ratings to compare with

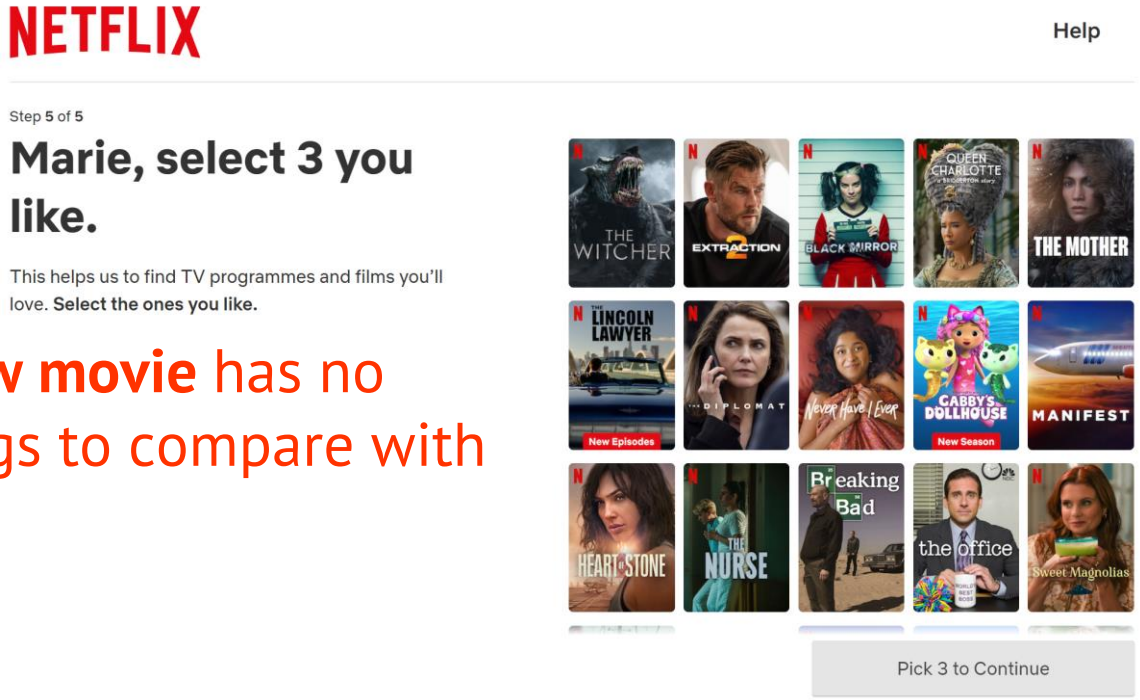
A **new user** has no ratings to compare with

Collaborative filtering struggles with the *cold start problem*: it's difficult to provide recommendations for new movies and users



A **new user** has no ratings to compare with

A **new movie** has no ratings to compare with



Collaborative filtering can be more intrusive than other types of recommender systems

Content-based filtering

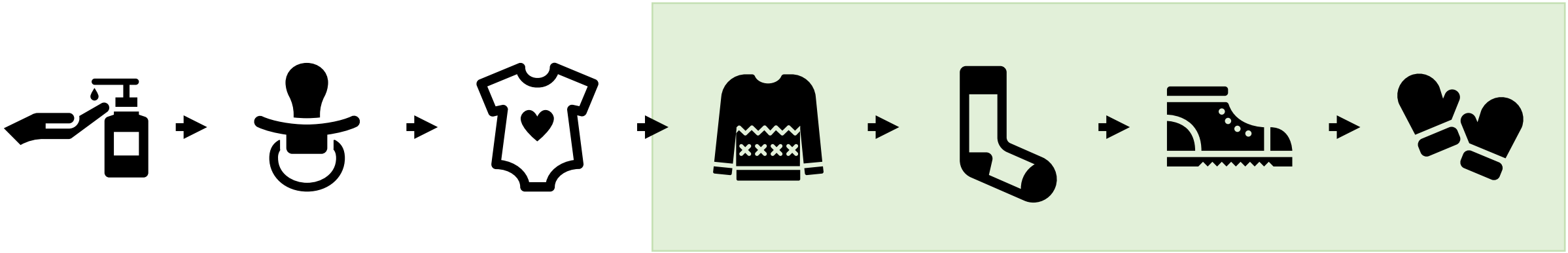
5	6	6	1	5	5
7	8	8	2	7	7
3	3	4	2	5	6
7	8	?	3	10	10
2	2	3	2	5	6

Collaborative filtering

Session-based filtering

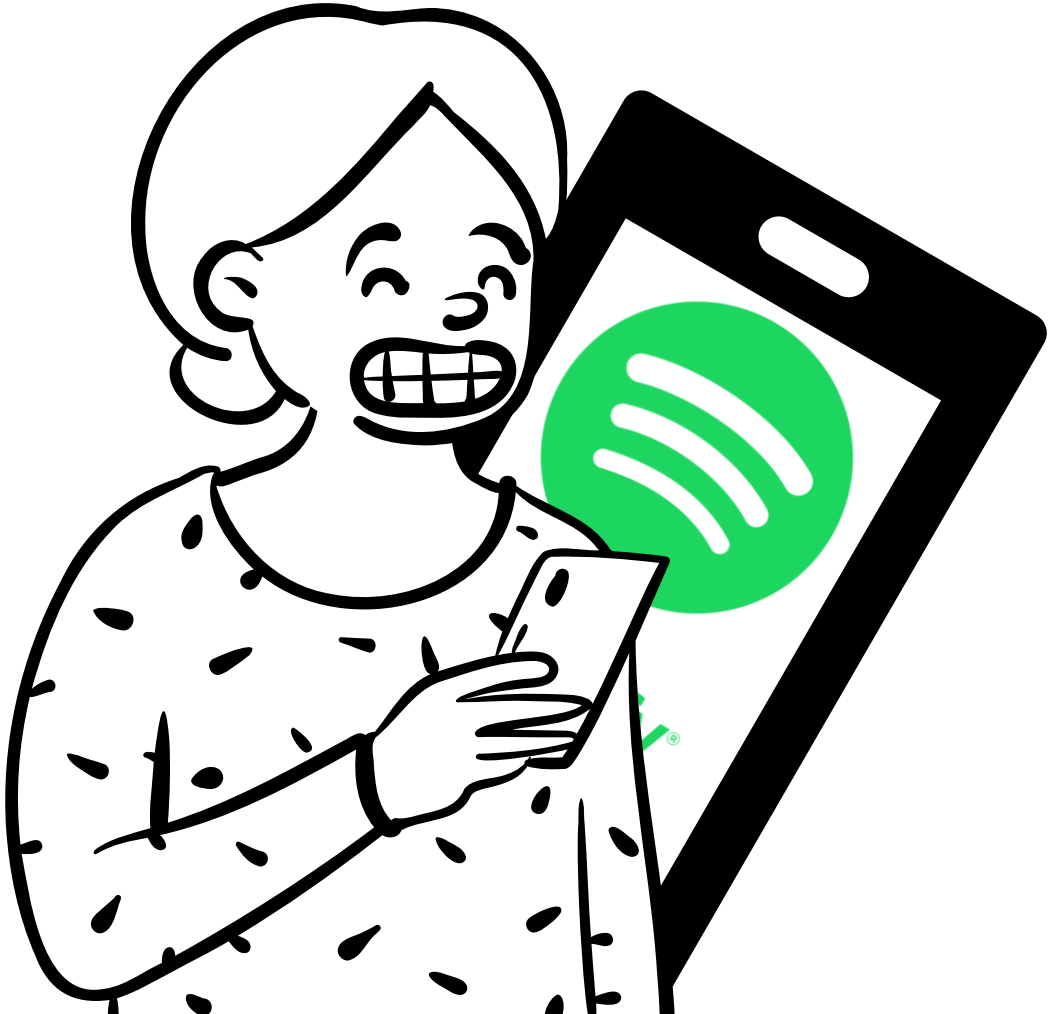
Concluding remarks

Sometimes, we don't want to consider more than the current session when we recommend new items



https://session-based-recommenders.fastforwardlabs.com/

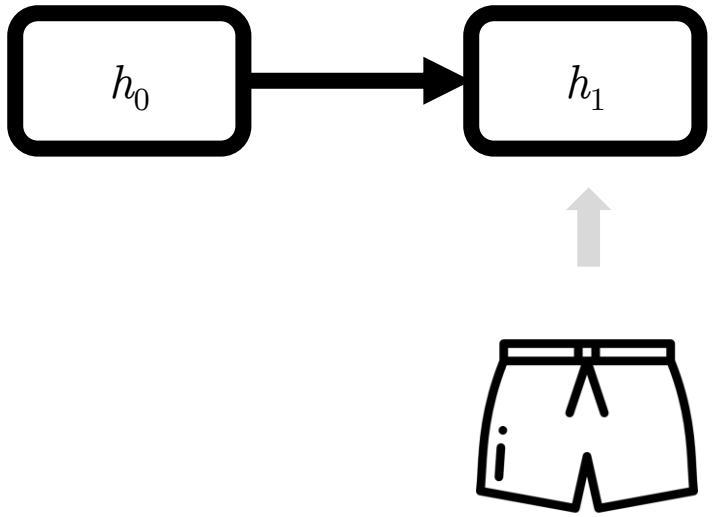
Session-based filtering is beneficial if user behaviour can vary greatly between sessions



Any collaborative filtering algorithm can be used for session-based filtering

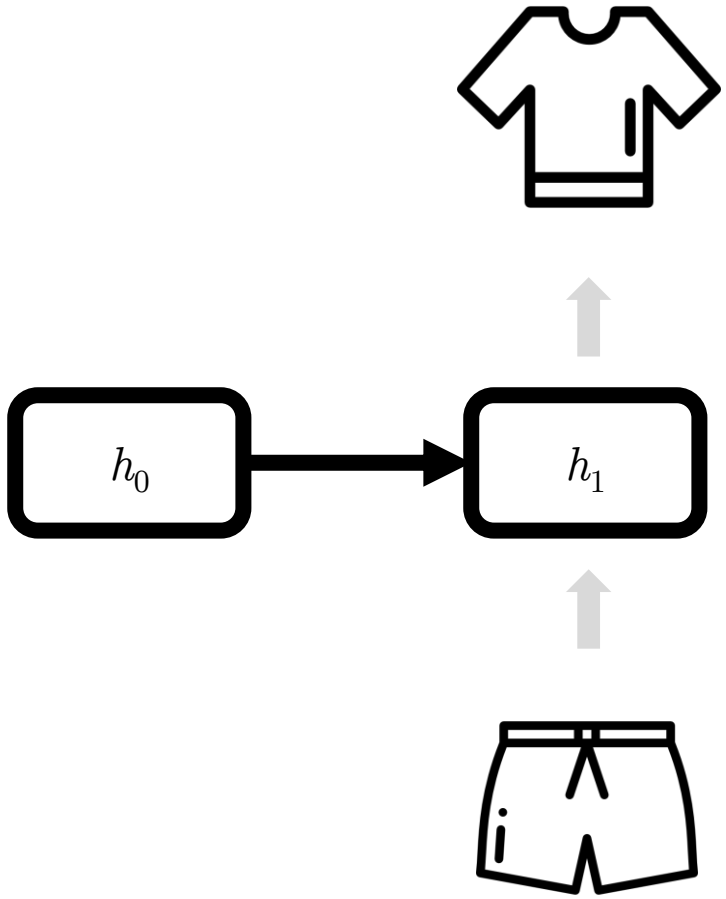
	Session 1	Session 2	Session 3	Session 4	Session 5	Session 6
Citizen Kane	5	6	6	1	5	5
Ellos Eatnu	7	8	8	2	7	7
Thor: Ragnarock	3	3	4	2	5	6
Kill Bill	7	8	9	3	10	10
Transformers	2	2	3	2	5	6

Session-based recommendation system makes them suitable for recurrent neural networks



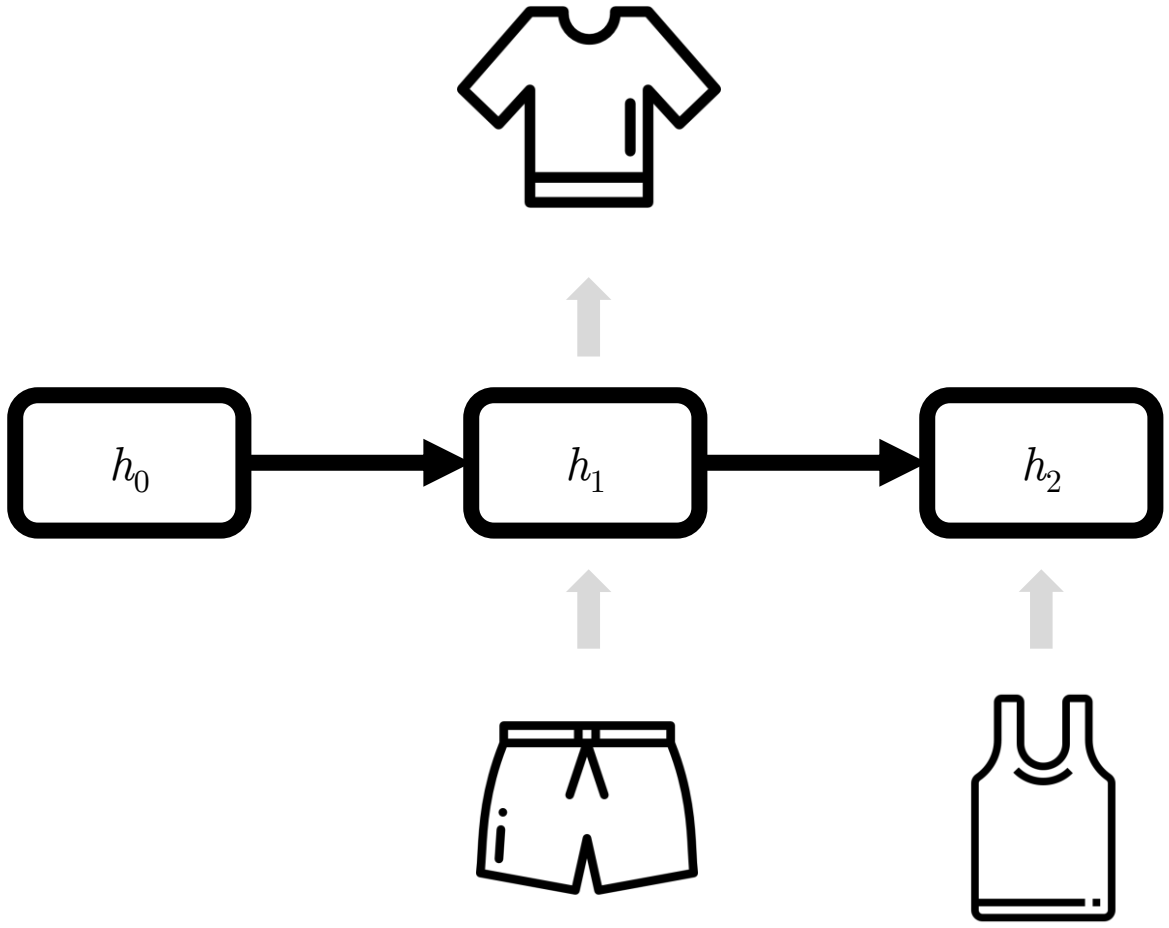
[https://research.atspotify.com/2021/04/contextual-and-sequential-user-embeddings-for-music-recommendation/]

Session-based recommendation system makes them suitable for recurrent neural networks



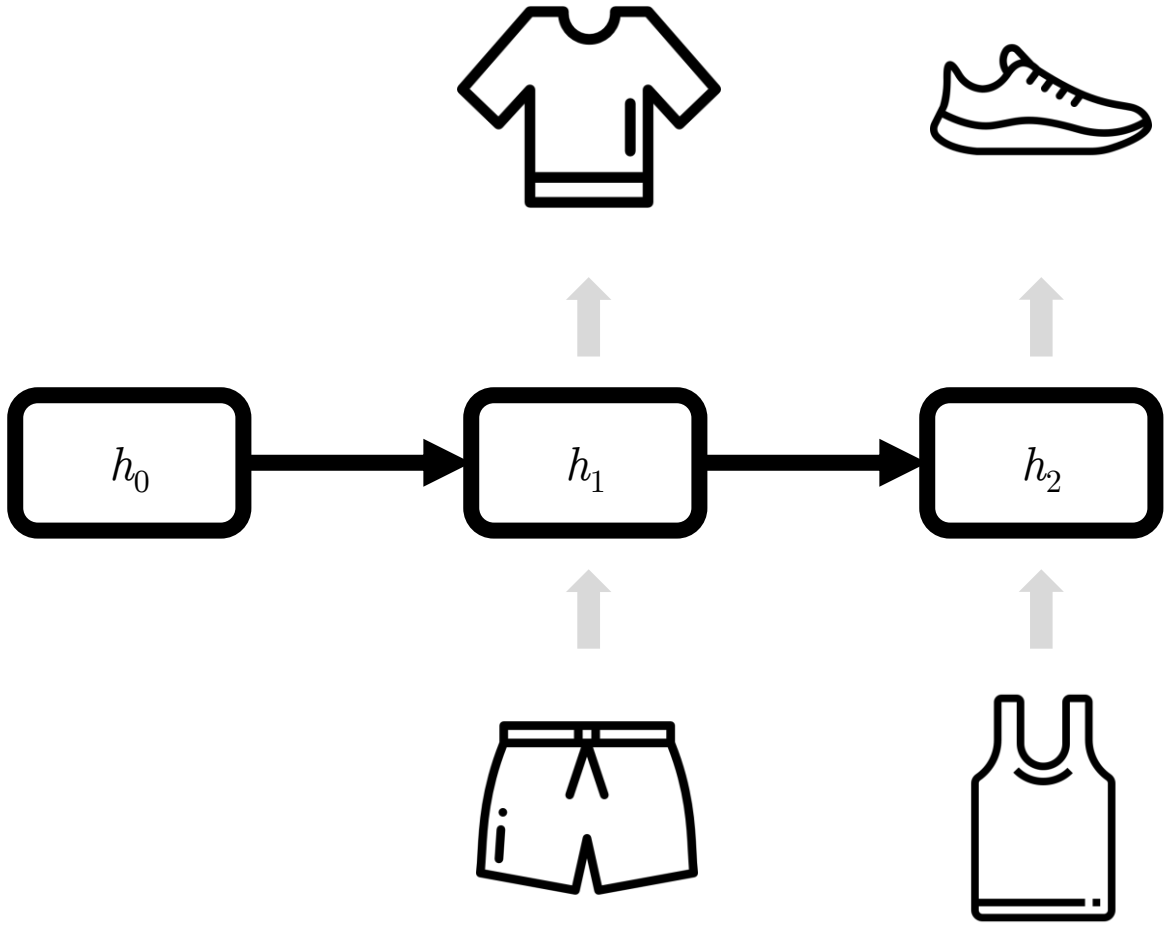
[https://research.atspotify.com/2021/04/contextual-and-sequential-user-embeddings-for-music-recommendation/]

Session-based recommendation system makes them suitable for recurrent neural networks



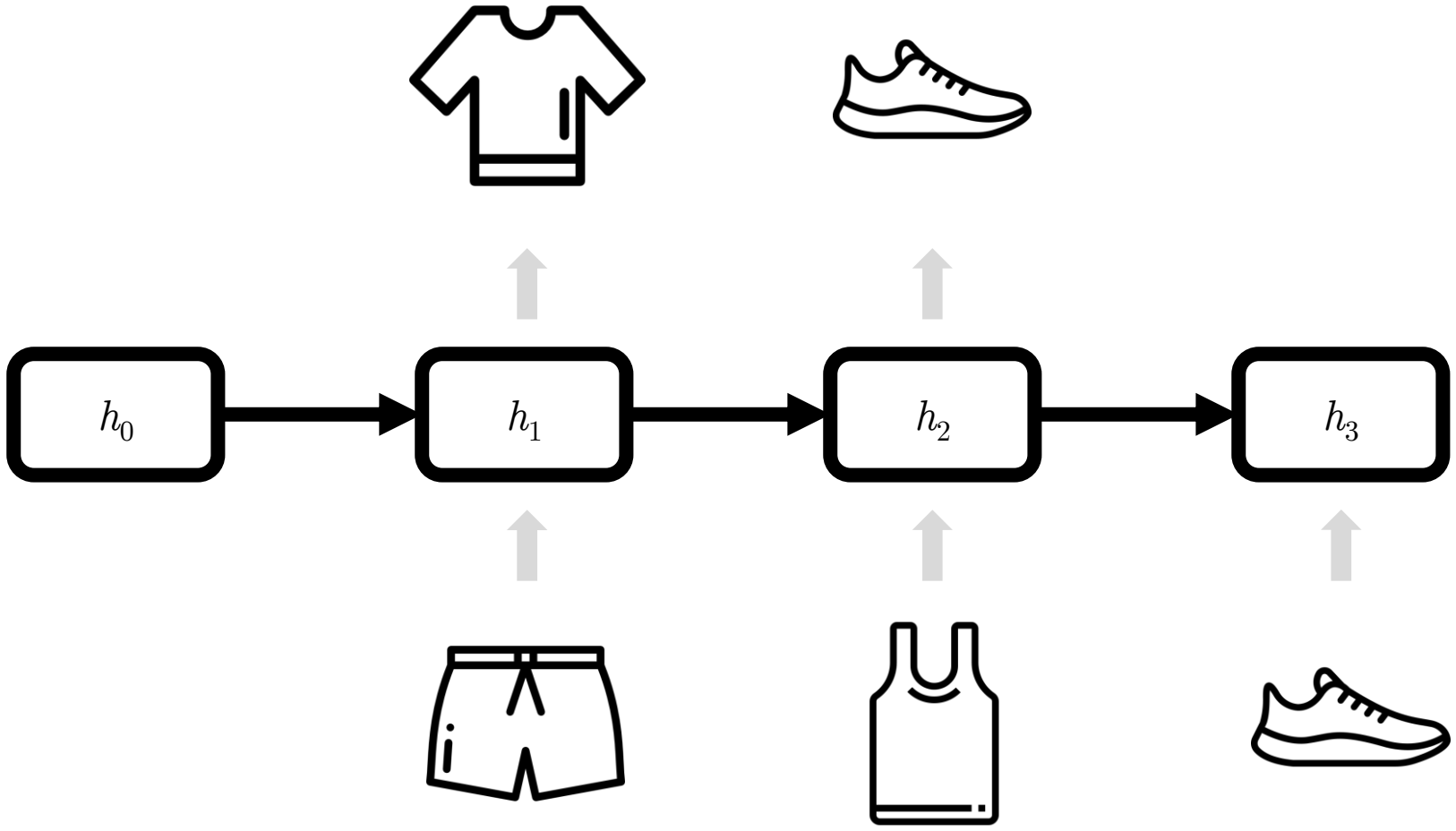
[https://research.atspotify.com/2021/04/contextual-and-sequential-user-embeddings-for-music-recommendation/]

Session-based recommendation system makes them suitable for recurrent neural networks



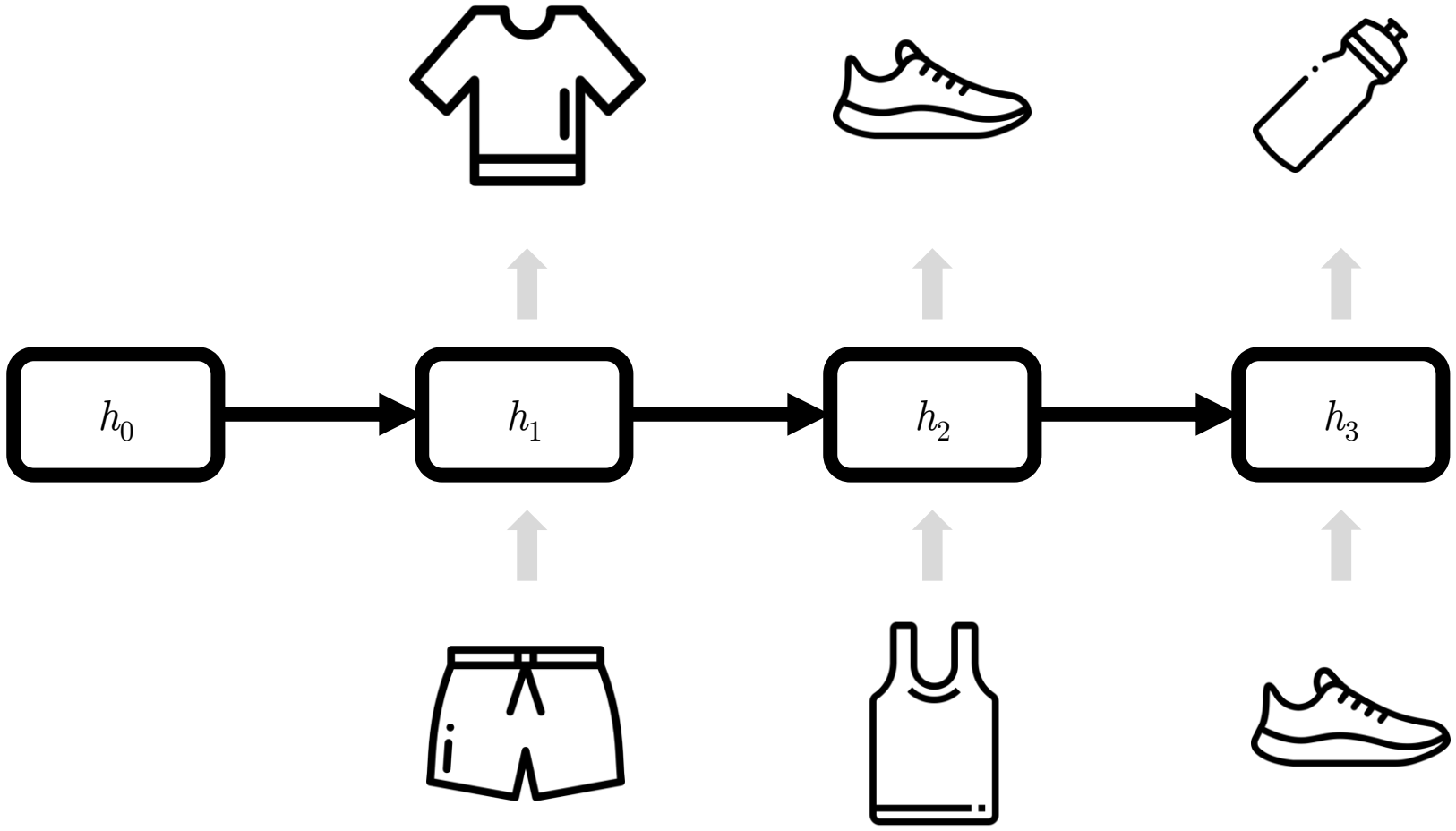
[https://research.atspotify.com/2021/04/contextual-and-sequential-user-embeddings-for-music-recommendation/]

Session-based recommendation system makes them suitable for recurrent neural networks



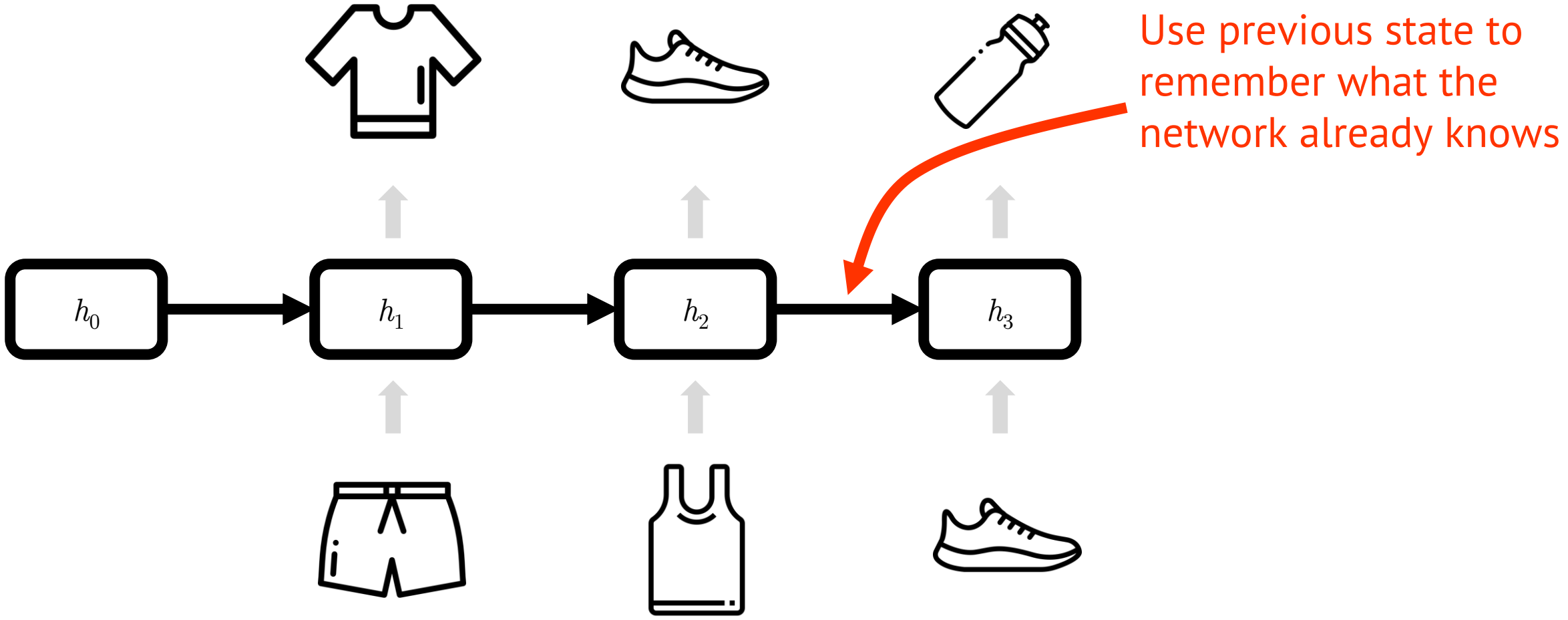
[https://research.atspotify.com/2021/04/contextual-and-sequential-user-embeddings-for-music-recommendation/]

Session-based recommendation system makes them suitable for recurrent neural networks



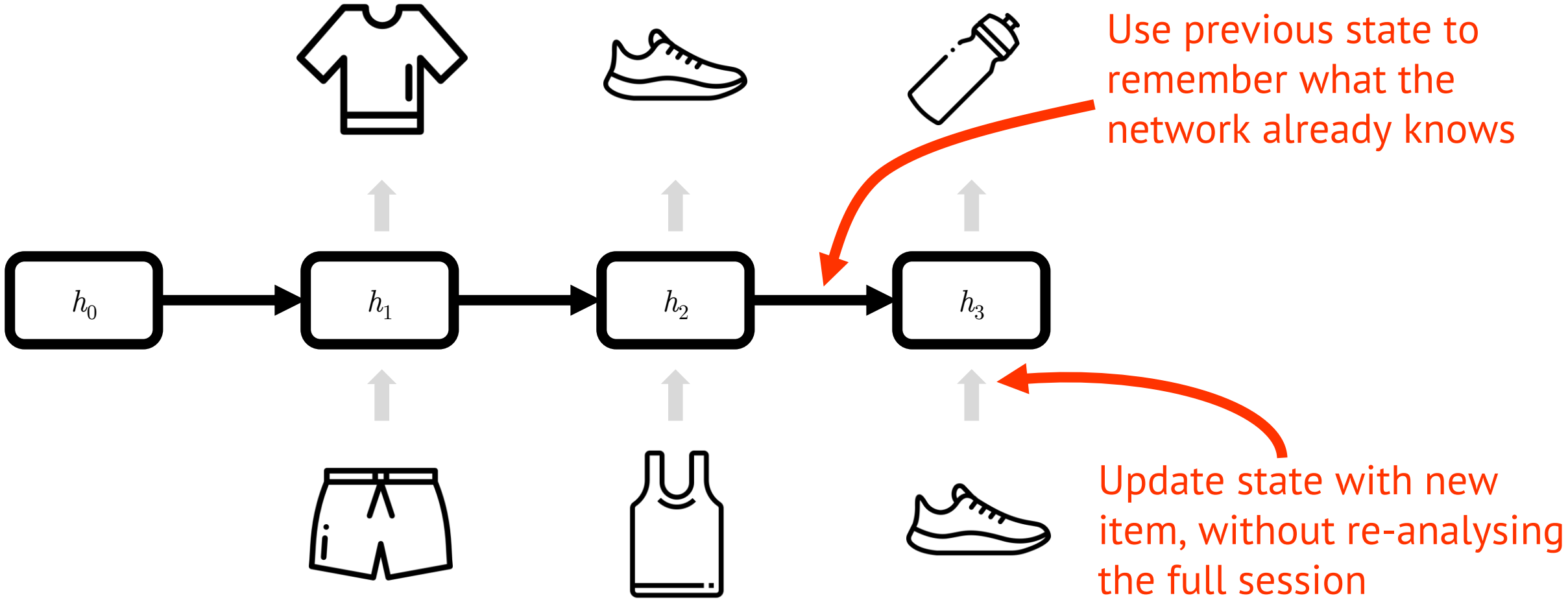
[https://research.atspotify.com/2021/04/contextual-and-sequential-user-embeddings-for-music-recommendation/]

Session-based recommendation system makes them suitable for recurrent neural networks



[https://research.atspotify.com/2021/04/contextual-and-sequential-user-embeddings-for-music-recommendation/]

Session-based recommendation system makes them suitable for recurrent neural networks



[https://research.atspotify.com/2021/04/contextual-and-sequential-user-embeddings-for-music-recommendation/]

Spotify used a recurrent neural network with additional contextual information to give just-in-time recommendations during a session

Contextual and Sequential User Embeddings for Large-Scale Music Recommendation

Casper Hansen* University of Copenhagen c.hansen@di.ku.dk

Christian Hansen* University of Copenhagen chrh@di.ku.dk

Lucas Maystre Spotify lucasm@spotify.com

Rishabh Mehrotra Spotify rishabhm@spotify.com

Brian Brost Spotify brianbrost@spotify.com

Federico Tomasi Spotify federicot@spotify.com

Mounia Lalmas Spotify mounia@acm.org

ABSTRACT

Recommender systems play an important role in providing an engaging experience on online music streaming services. However, the musical domain presents distinctive challenges to recommender systems: tracks are short, listened to multiple times, typically consumed in sessions with other tracks, and relevance is highly context-dependent. In this paper, we argue that modeling users' preferences at the beginning of a session is a practical and effective way to address these challenges. Using a dataset from Spotify, a popular music streaming service, we observe that *a*) consumption from the recent past and *b*) session-level contextual variables (such as the time of the day or the type of device used) are indeed predictive of the tracks a user will stream—much more so than static, average preferences. Driven by these findings, we propose CoSeRNN, a neural network architecture that models users' preferences as a sequence of embeddings, one for each session. CoSeRNN predicts, at the beginning of a session, a preference vector, based on past consumption history and current context. This preference vector can then be used in downstream tasks to generate contextually relevant just-in-time recommendations efficiently, by using approximate nearest-neighbour search algorithms. We evaluate CoSeRNN on session and track ranking tasks, and find that it outperforms the current state of the art by upwards of 10% on different ranking metrics. Dissecting the performance of our approach, we find that sequential and contextual information are both crucial.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Music retrieval**.

KEYWORDS

Music Recommendation, User Embeddings, Context, Sequence

*This work was done while the first two authors were at Spotify.

ACM Reference Format:

1 INTRODUCTION

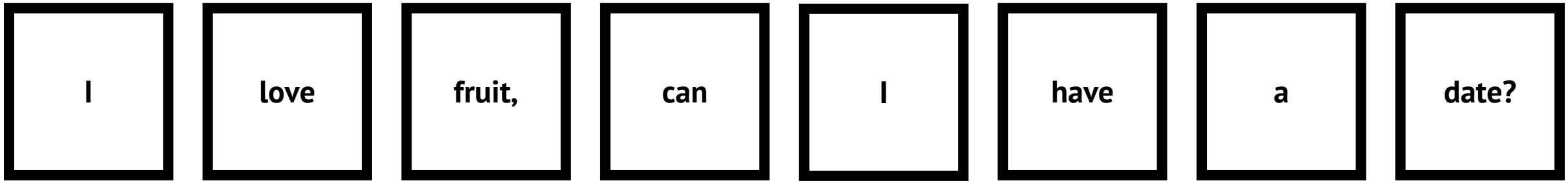
Recommender systems are essential for providing an engaging experience and for helping users navigating the vast amounts of content available in online services. Successful recommender systems have to accurately model each user's individual preferences, such that the most relevant content can be presented to the user. In this work, we consider online music streaming services, which have become increasingly popular in the past decade. By letting users access millions of tracks at the click of a button, they are contributing to democratizing access to music. However, in contrast to other well-studied domains (such as recommending books, movies or clothes), music recommender systems face distinctive challenges [30]. Tracks are short, and therefore often consumed together with other tracks we refer to such a set of tracks listened to in short succession as a session. A given session often contains tracks from the user's recent consumption history [1], suggesting that the sequence of sessions captures essential information about users' changing preferences. Additionally, the relevance of tracks is highly contextual, and preferences depend, among others, on the time of the day and the current season [24]. We seek to embrace these distinctive characteristics to produce a better, more accurate model of user preferences. We focus on the following problem: for a given user, we are interested in predicting, at the beginning of a session, which tracks the user will listen to during the session. We assume that we have access to the user's past consumption and to information about the *current* context. This formulation of the problem enables generating recommendations that are not only matching the user's global tastes, but are also tailored to the specific context and situation they currently find themselves in. While generic context-aware recommender systems have been studied in the past [28], little work has been focused on music recommendation. We aim to address this gap.

We begin our investigation by exploring a dataset from an online music streaming service, containing detailed information about the tracks streamed during a two months period for a sample of 200,000 users. We define context as the time of the day (morning, afternoon,

53

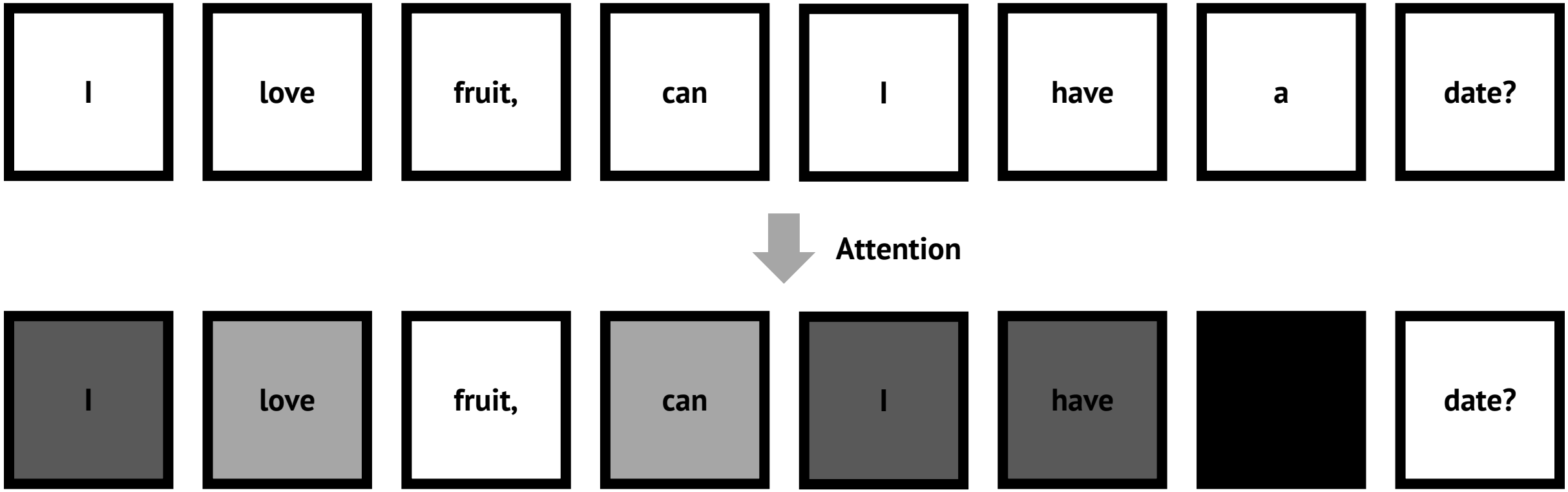
[https://research.atspotify.com/2021/04/contextual-and-sequential-user-embeddings-for-music-recommendation/]

Transformer-based architectures, similar to those used by large language models are also very suitable for session-based recommendation



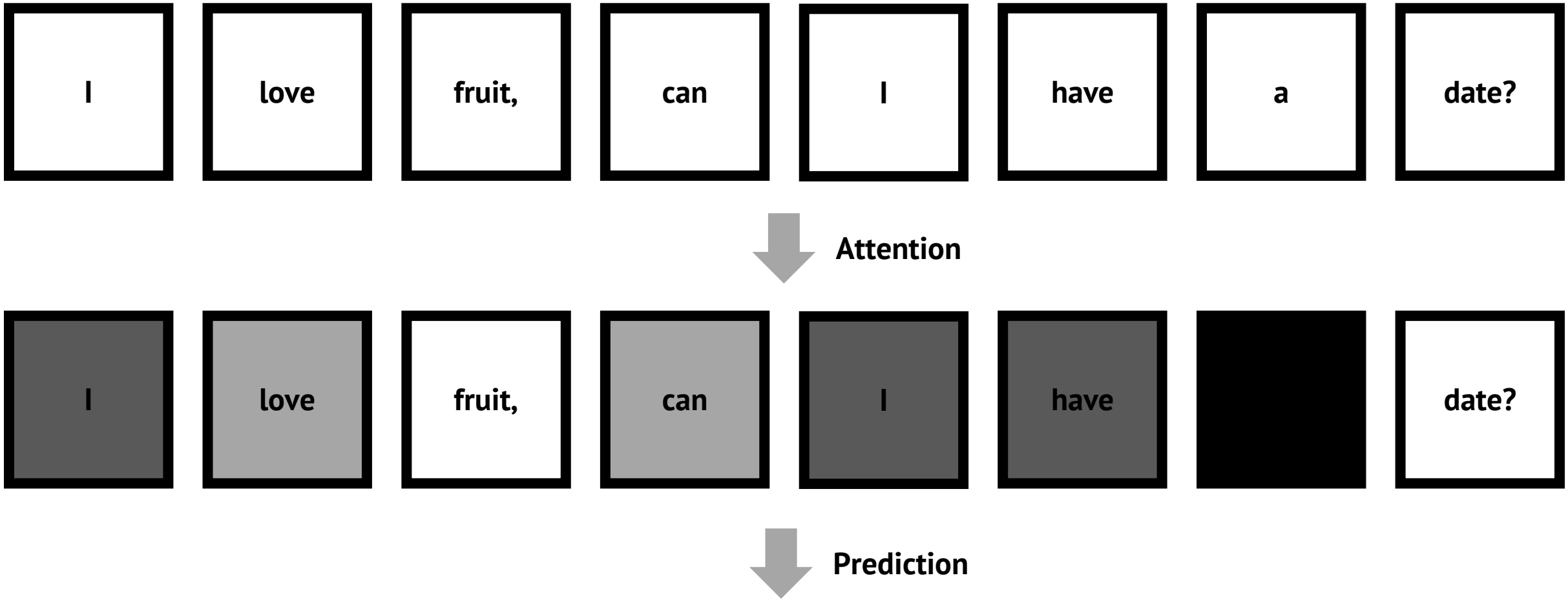
[Vaswan et al. NeurIPS (2017)]

Transformer-based architectures, similar to those used by large language models are also very suitable for session-based recommendation



[Vaswan et al. NeurIPS (2017)]

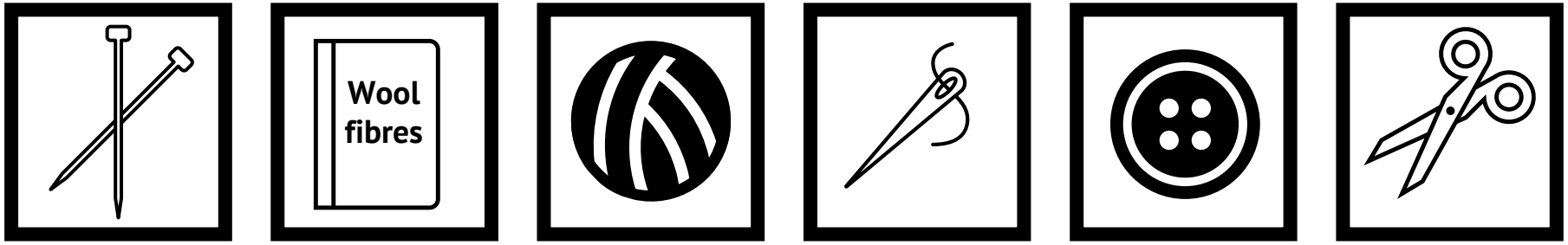
Transformer-based architectures, similar to those used by large language models are also very suitable for session-based recommendation



Of course, enjoying fruit is a great way to satisfy your cravings for something sweet and healthy!

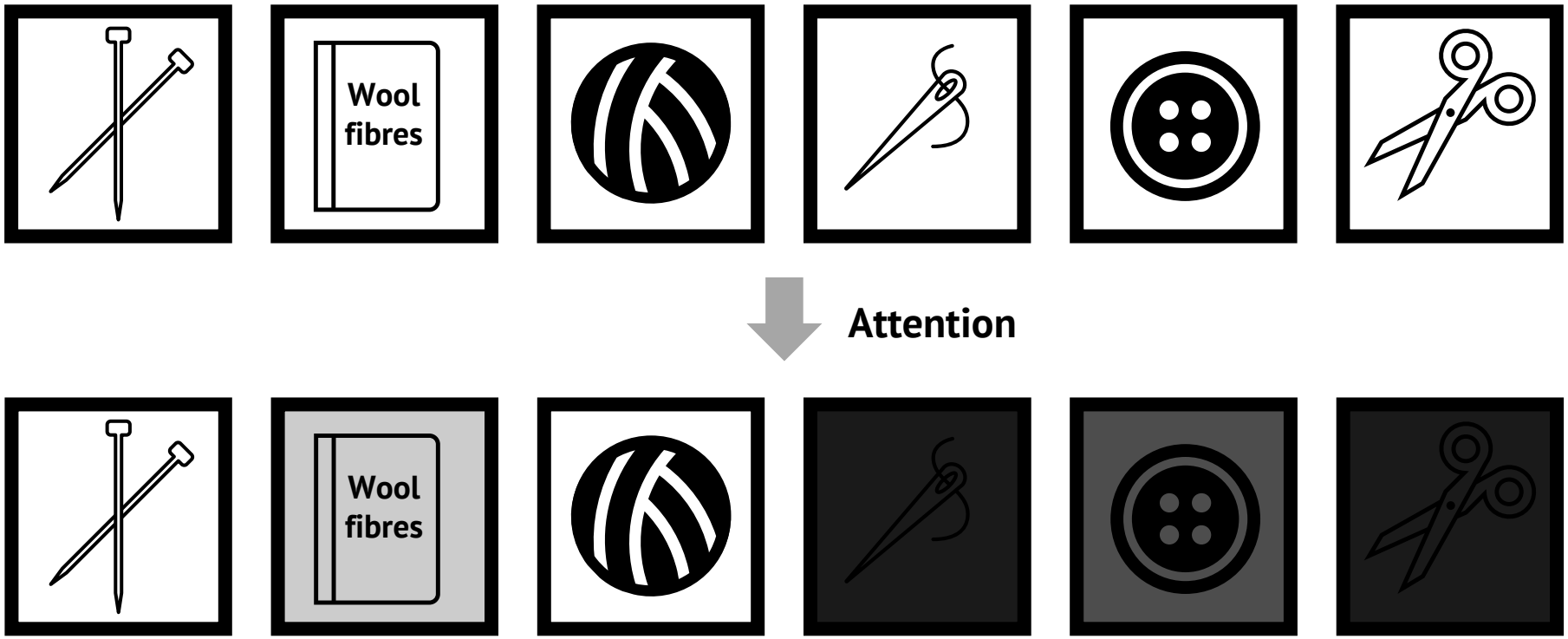
[Vaswan et al. NeurIPS (2017)]

Transformer-based architectures, similar to those used by large language models are also very suitable for session-based recommendation



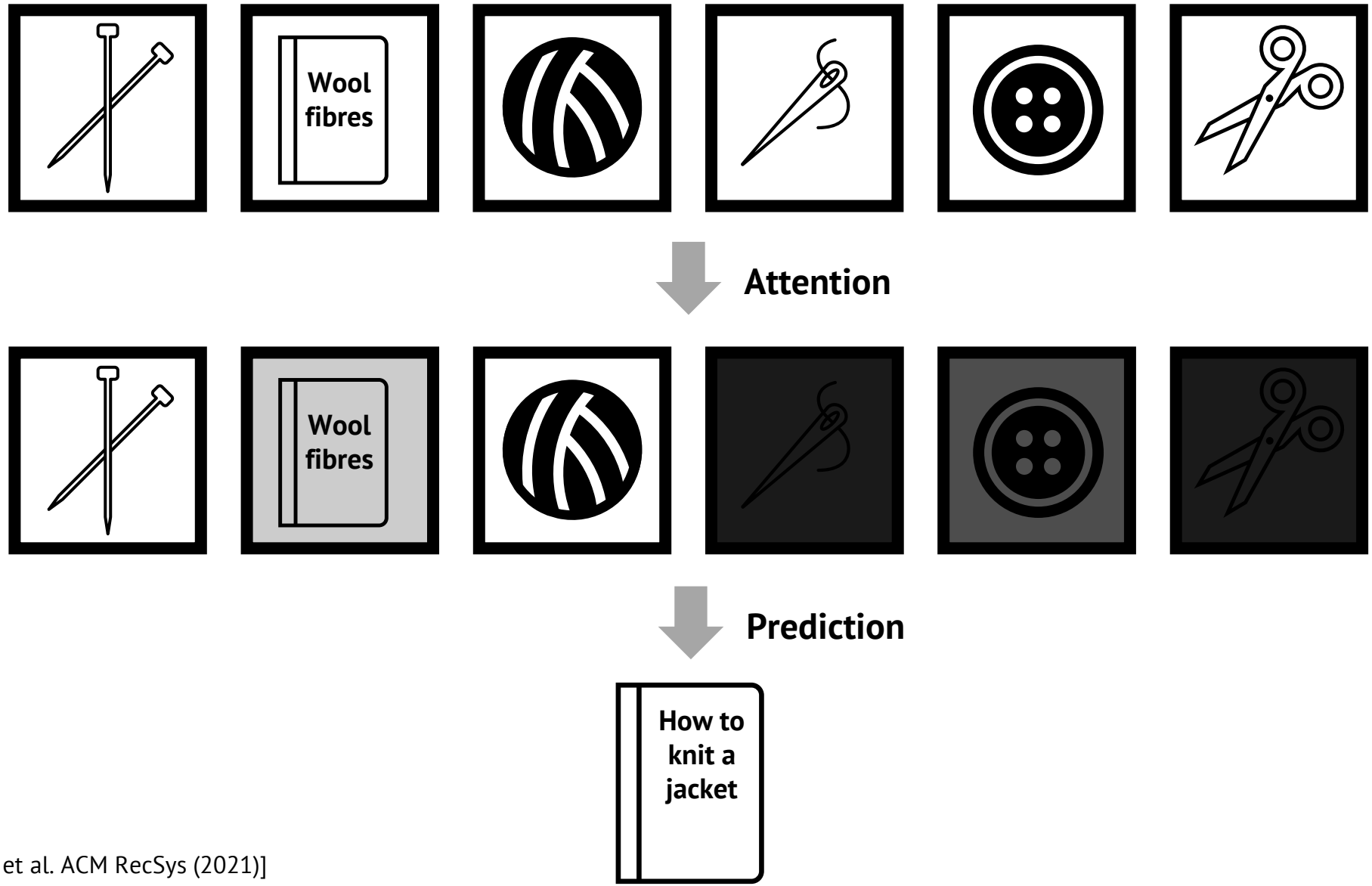
[Moreira et al. ACM RecSys (2021)]

Transformer-based architectures, similar to those used by large language models are also very suitable for session-based recommendation



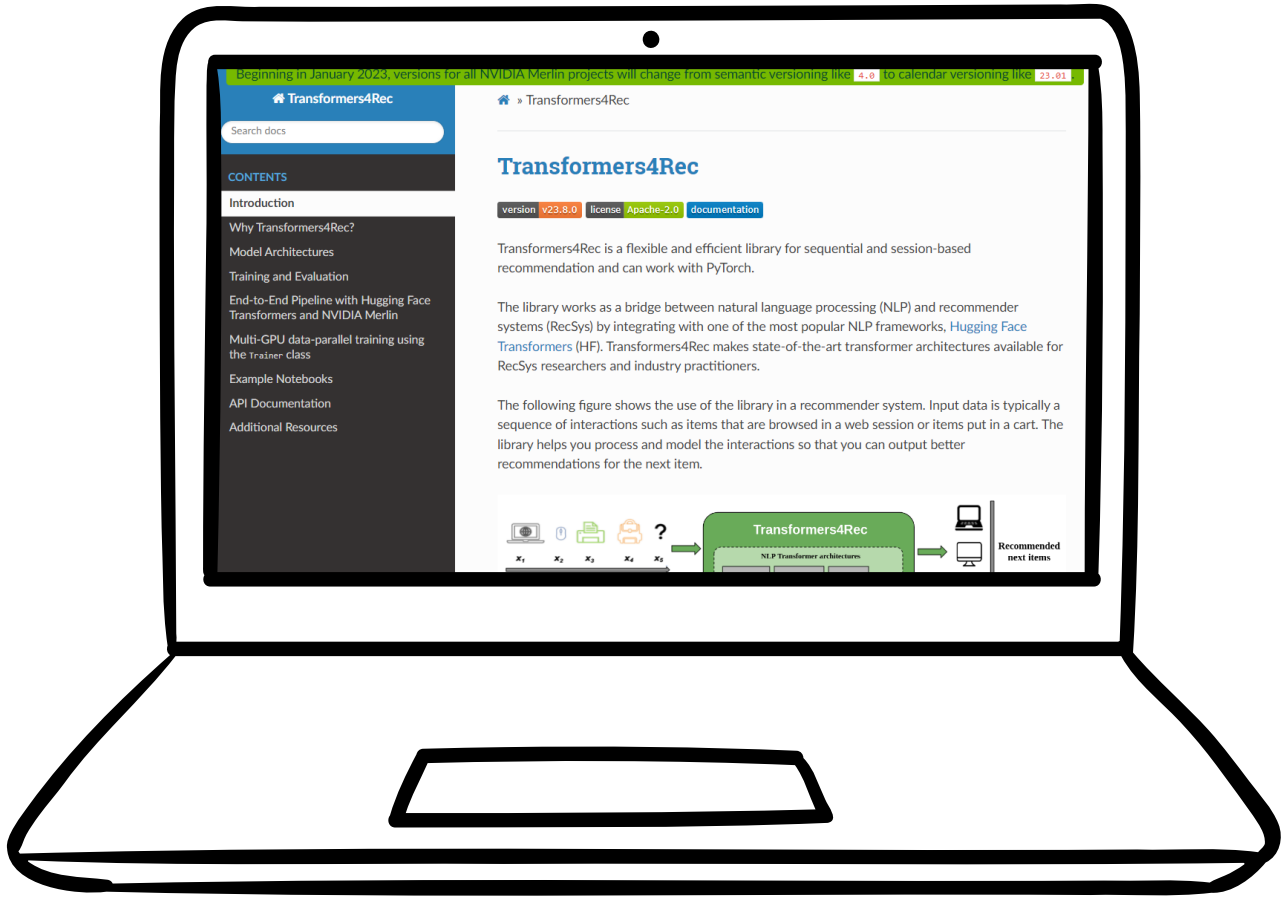
[Moreira et al. ACM RecSys (2021)]

Transformer-based architectures, similar to those used by large language models are also very suitable for session-based recommendation



[Moreira et al. ACM RecSys (2021)]

Transformers4Rec by Nvidia makes it easy to use transformers-based NLP algorithms for session-based recommendation systems




Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation

Gabriel de Souza Pereira Moreira, gmoreira@nvidia.com, NVIDIA, São Paulo, Brazil
Sara Rabhi, srabhi@nvidia.com, NVIDIA, Ontario, Canada
Jeong Min Lee*, jeongmin@fb.com, Facebook AI, California, United States
Ronay Ak, ronaya@nvidia.com, NVIDIA, Florida, United States
Even Oldridge, eoldridge@nvidia.com, NVIDIA, British Columbia, Canada

ABSTRACT

1 INTRODUCTION

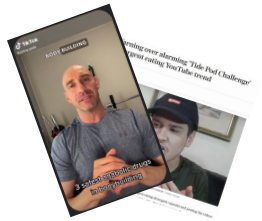
[Moreira et al. ACM RecSys (2021)]

Content-based filtering

5	6	6	1	5	5
7	8	8	2	7	7
3	3	4	2	5	6
7	8	?	3	10	10
2	2	3	2	5	6

Collaborative filtering

Session-based filtering



Concluding remarks

It’s common to combine these paradigms in a hybrid approach

Session embedding

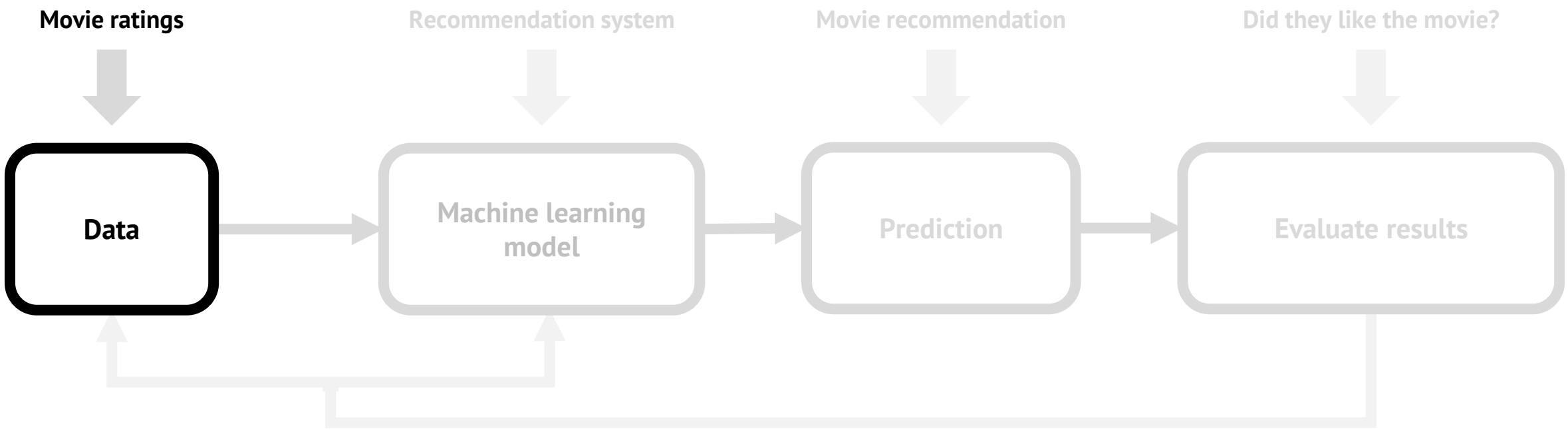
User embedding

Song embedding

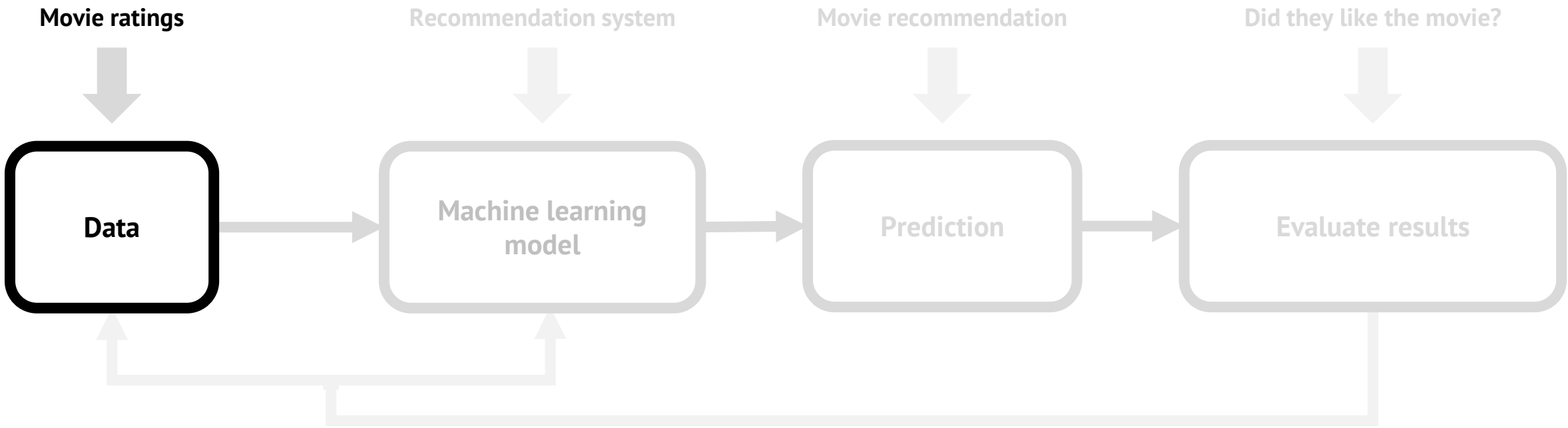
Contextual features

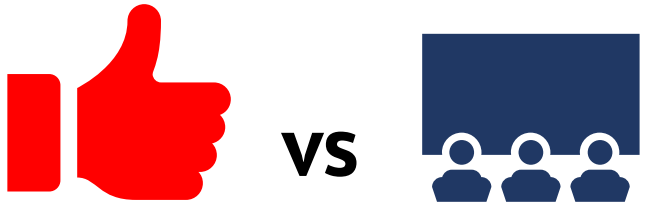
Song recommendation

What data we collect shapes what type of recommendation we perform



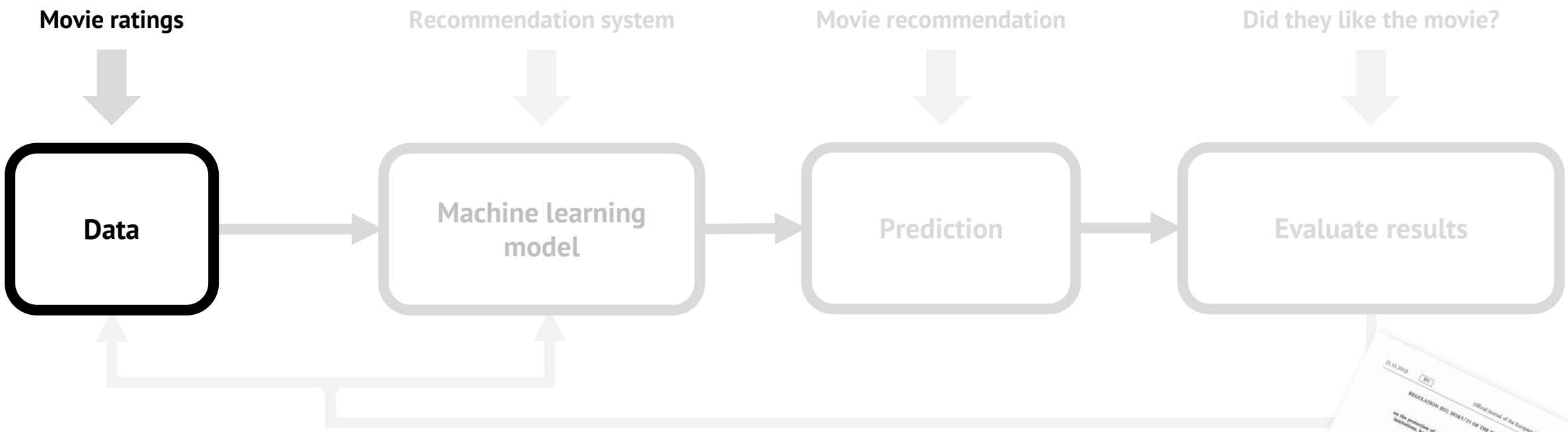
What data we collect shapes what type of recommendation we perform





Explicit vs Implicit

What data we collect shapes what type of recommendation we perform

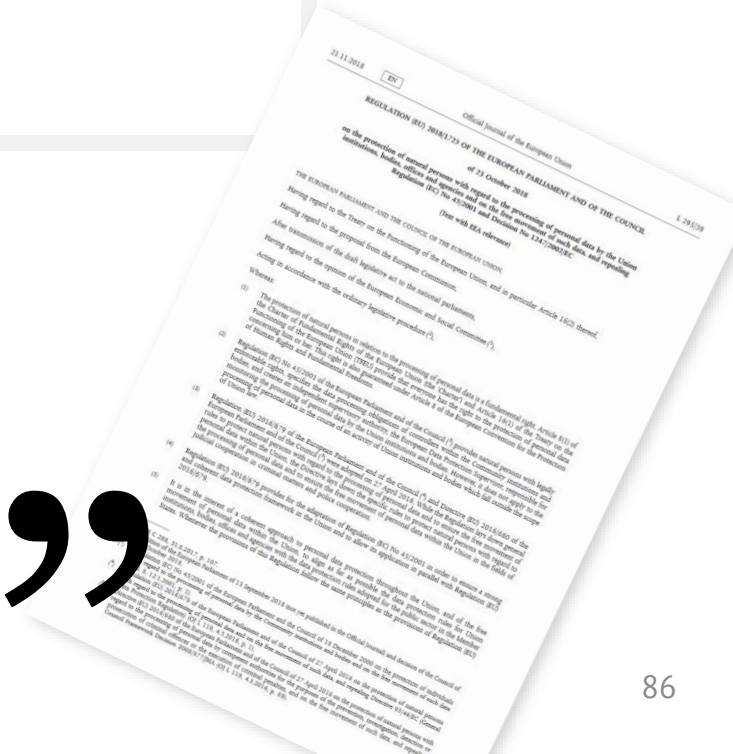


“

Article 4

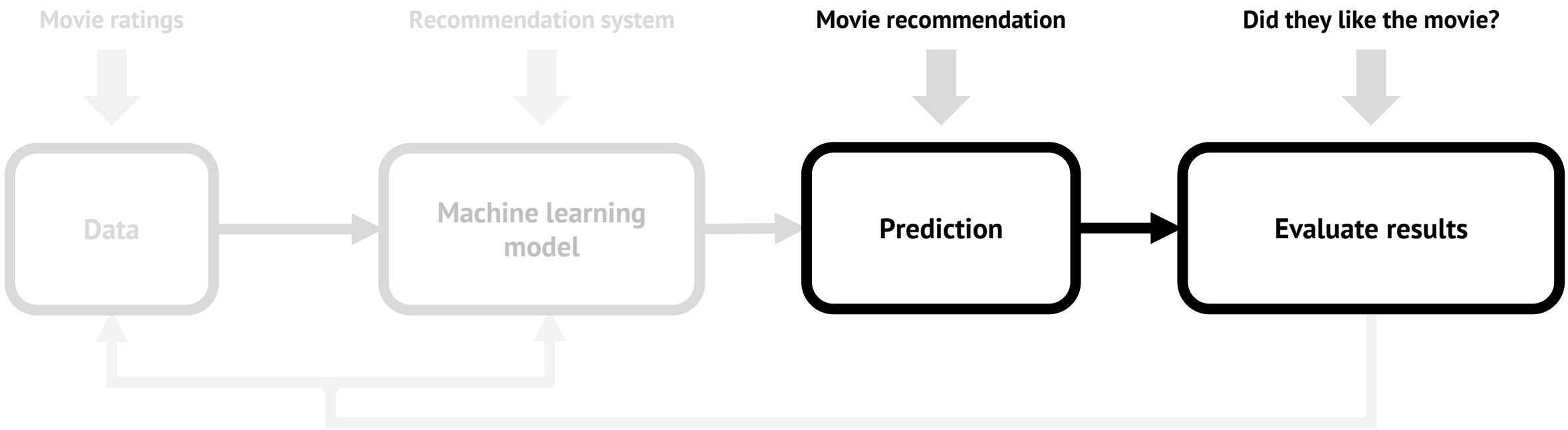
1. Personal data shall be: […]
   c) adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed (‘data minimisation’);

”



REGULATION (EU) 2018/1725 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL (GDPR)

How you evaluate your recommendations affects which model you use



How you evaluate your recommendations affects which model you use

Movie ratings

Recommendation system

Movie recommendation

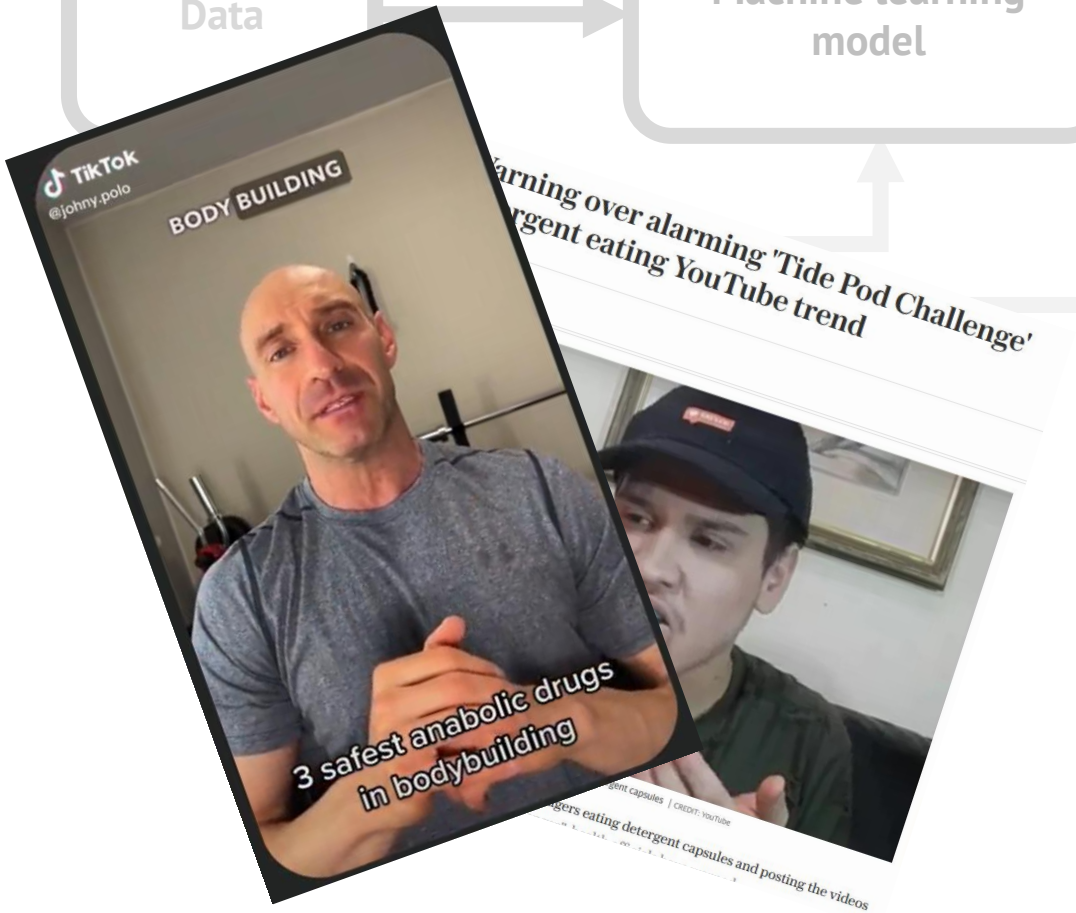
Did they like the movie?

Data

Machine learning model

Prediction

Evaluate results



53 % of children in Norway aged 13-18 years old see adverts for muscle growth supplements

[The Norwegian Media Authority: *Barn og medier 2022: Reklame influensere og sosiale medier*]
[NRK (2023): *TikTok's Muscle Power*]

Further reading about recommendation systems:

Websites and blog posts

- Cloudera (2021). ***Session-based Recommender Systems***
  - https://session-based-recommenders.fastforwardlabs.com/
- Nvidia (2021). ***Winning the SIGIR eCommerce Challenge on session-based recommendation with Transformers***
  - https://medium.com/nvidia-merlin/winning-the-sigir-ecommerce-challenge-on-session-based-recommendation-with-transformers-v2-793f6fac2994
- Spotify (2021). **Contextual and Sequential User Embeddings for Music Recommendation**
  - https://research.atspotify.com/2021/04/contextual-and-sequential-user-embeddings-for-music-recommendation/
- Shutterstock Data Science (2022). **Building Content Personalization**
  - https://www.shutterstock.com/blog/data-science-building-content-personalization

Papers

- Johnson, Douze & Jégou. IEEE Trans Big Data (2019). ***Billion-Scale Similarity Search with GPUs***
- Vaswan et al. NeurIPS (2017). ***Attention is all you need***
- Hansen et al. ACM RecSys (2020). ***Contextual and Sequential User Embeddings for Large-Scale Music Recommendation***
- Moreira et al. ACM RecSys (2021). ***Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation***
- Koren, Bell & Volinsky. IEEE Computer (2009). ***Matrix Factorization Techniques for Recommender Systems***

Books

- Falk. Manning Publications (2019). ***Practical Recommender Systems***

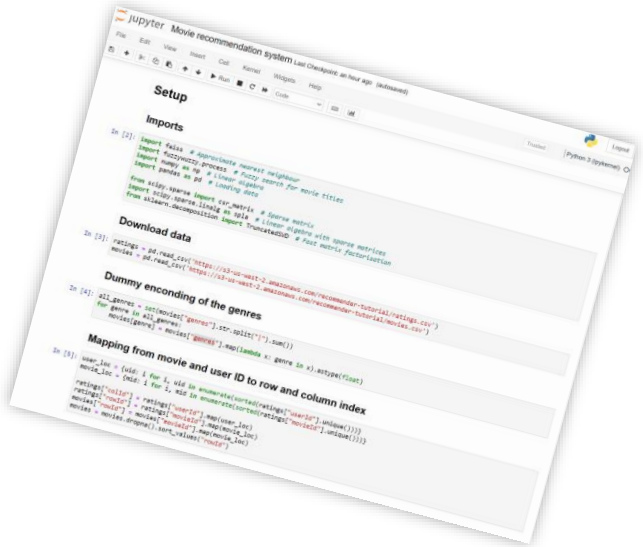
Software resources for making recommendation systems in Python

Software packages

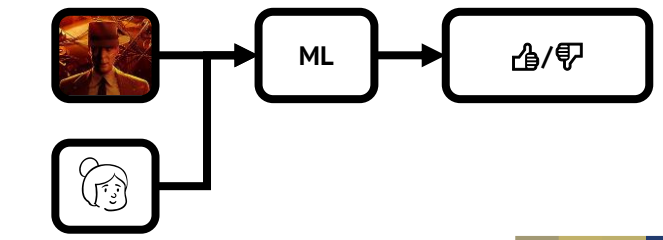
- **Surprise**: https://surpriselib.com/
  - Simple but effective methods for collaborative filtering:
  - Builtin support for model evaluation
- **Implicit**: https://benfred.github.io/implicit/
  - Collaborative filtering for implicit ratings
- **Gensim**: https://radimrehurek.com/gensim/
  - Word2vec
- **Transformers4Rec**: https://nvidia-merlin.github.io/Transformers4Rec/main/
  - Transformers for recommendation systems
- **FAISS**: https://faiss.ai/
  - Fast approximate KNN searches supporting several metrics

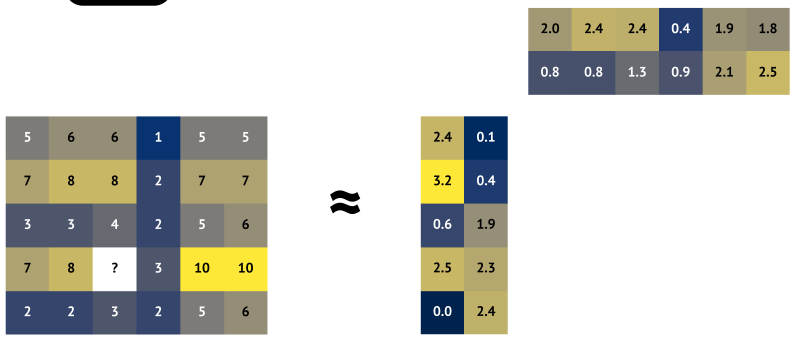
Code example

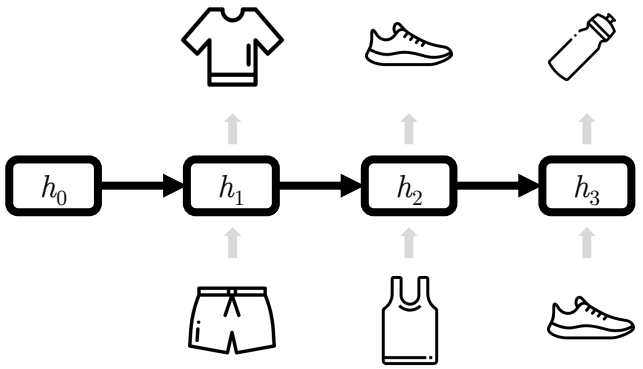
- **Example notebook**: https://github.com/MarieRoald/phd-defence-trial-lecture
  - Covering content-based, memory-based and model-based collaborative filtering with matrix factorisation and approximate nearest neighbour searches
- **Workshop from TMLS 2020**: https://github.com/topspinj/tmls-2020-recommender-workshop
  - More in-depth notebook showing explorative analysis and how to create various types of recommender systems



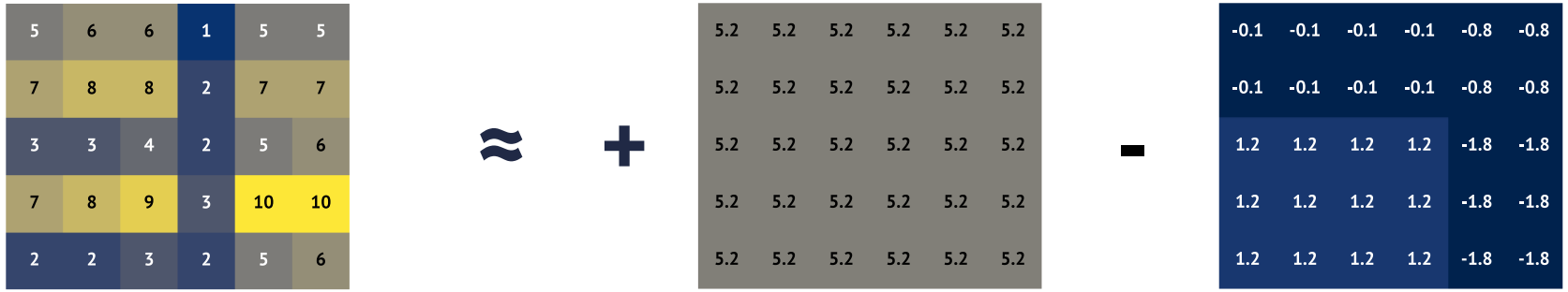
User features can be combined with item features to provide more personalized recommendations





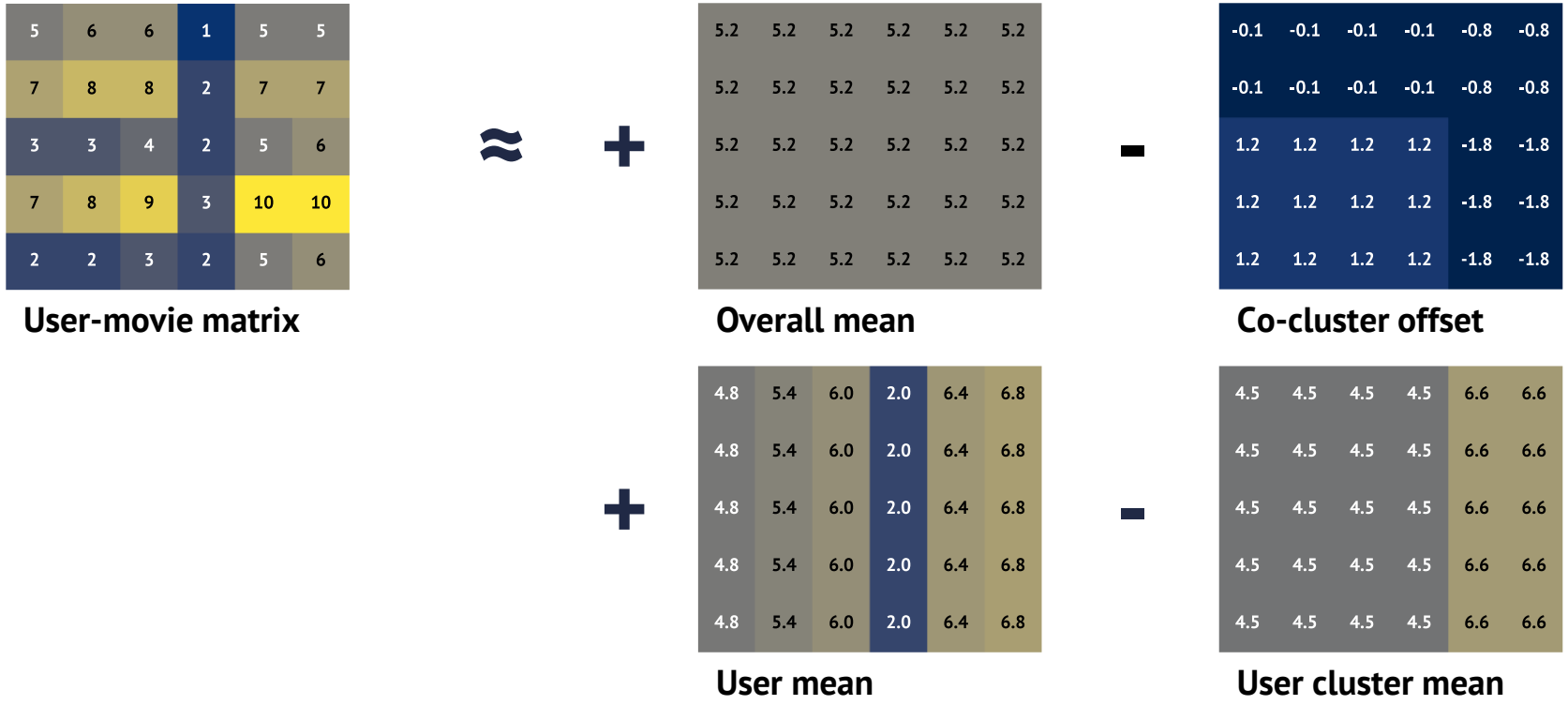


Co-clustering utilise clusters of similar scores among both users and movies simultaneously



User-movie matrix | Overall mean | Co-cluster offset

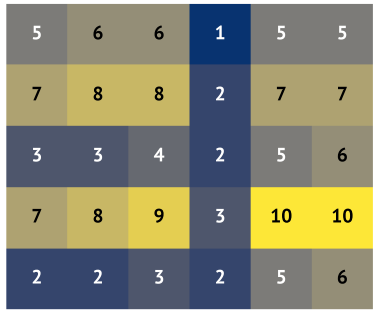
Co-clustering utilise clusters of similar scores among both users and movies simultaneously



User-movie matrix | Overall mean | Co-cluster offset

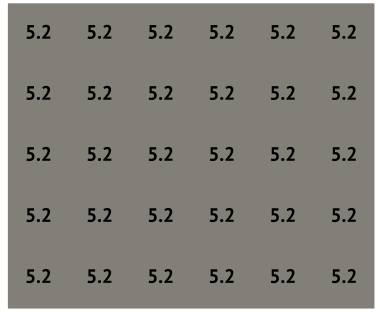
User mean | User cluster mean

Co-clustering utilise clusters of similar scores among both users and movies simultaneously



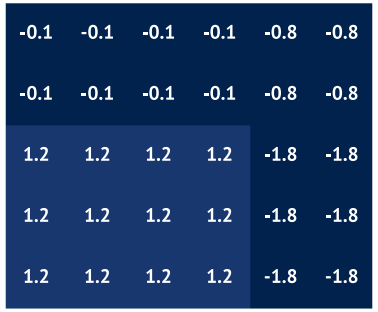
User-movie matrix

≈



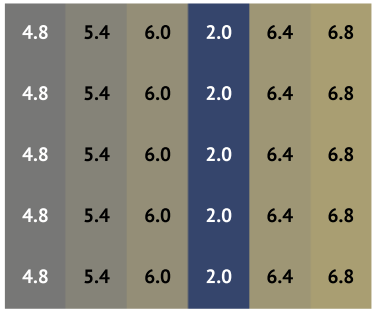
Overall mean

\+



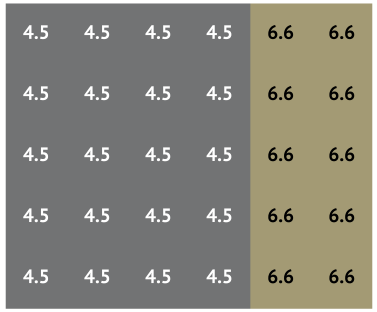
Co-cluster offset

\+



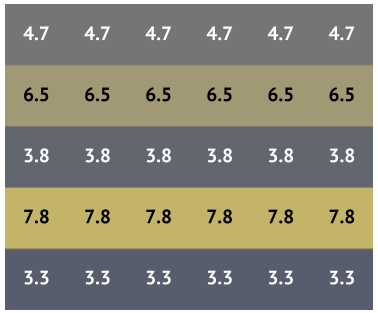
User mean

\-



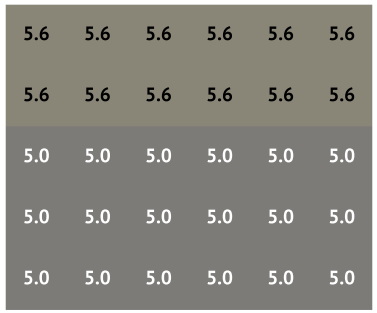
User cluster mean

\+



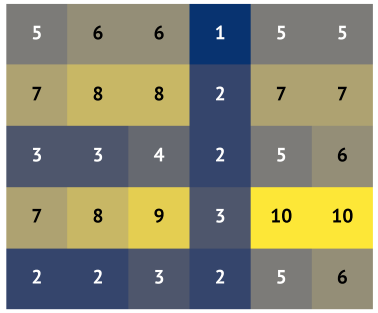
Movie mean

\-



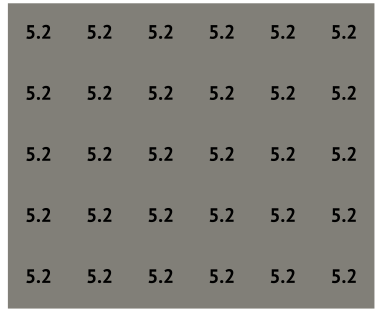
Movie cluster mean

Co-clustering utilise clusters of similar scores among both users and movies simultaneously



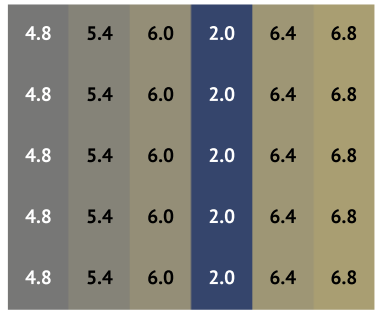
User-movie matrix

≈



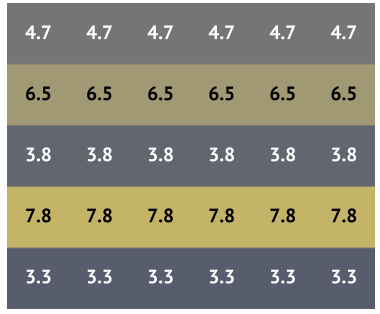
Overall mean

\+



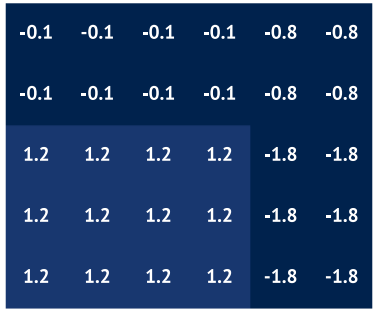
User mean

\+



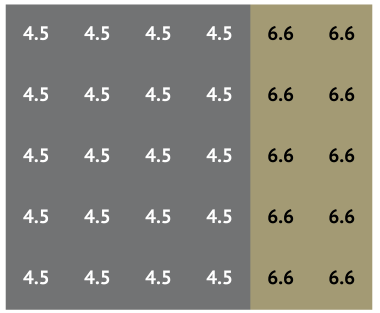
Movie mean

\-



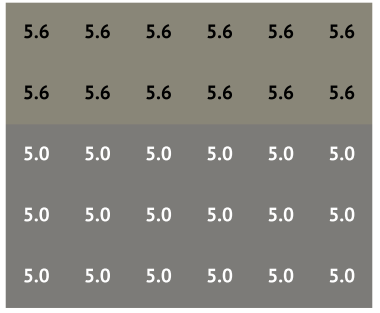
Co-cluster offset

\-



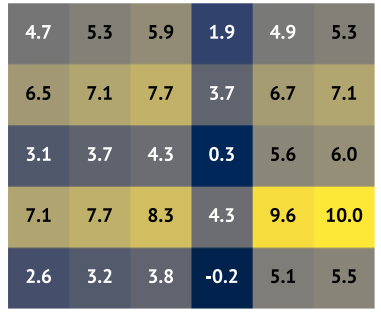
User cluster mean

\-



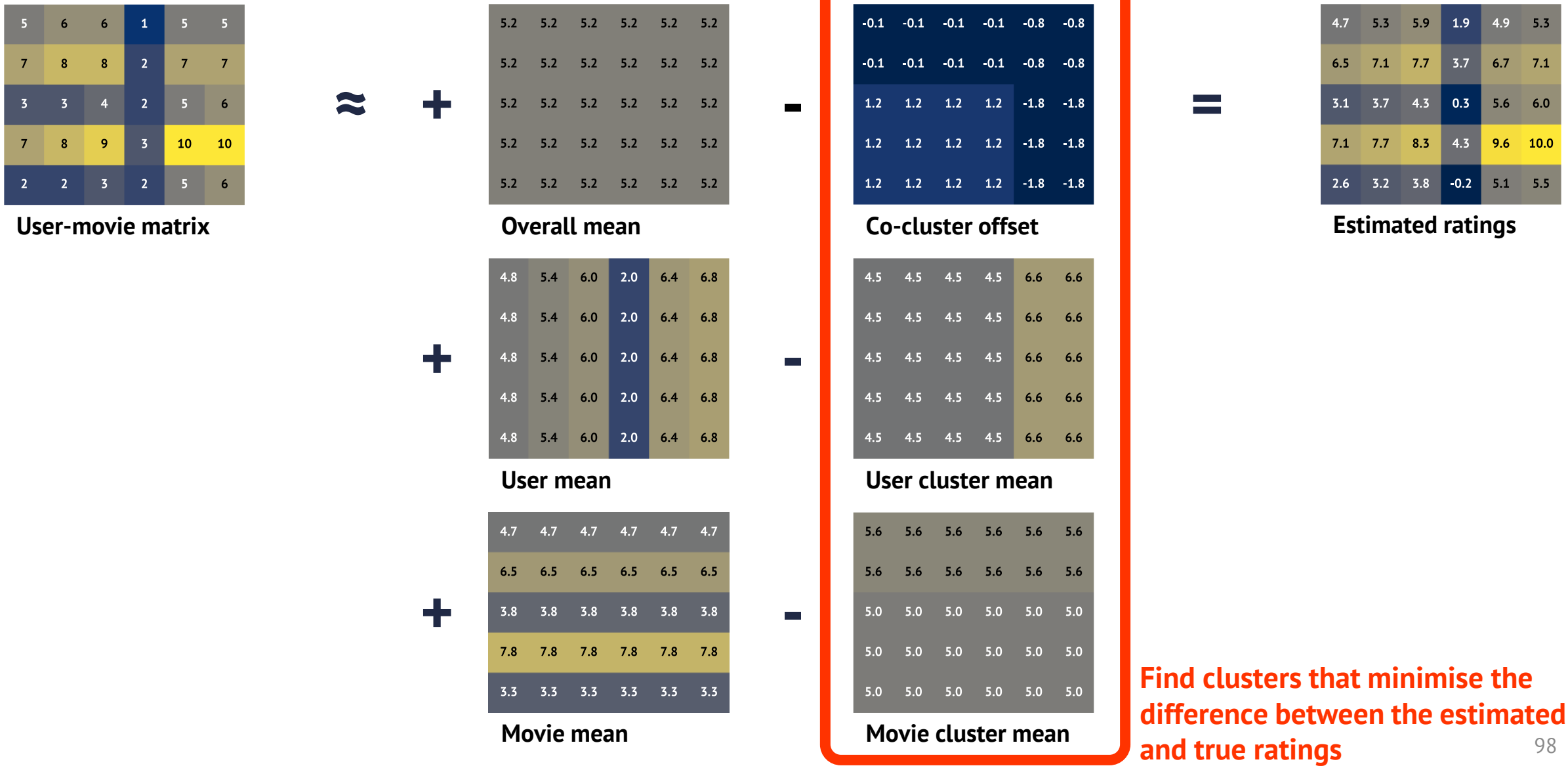
Movie cluster mean

=

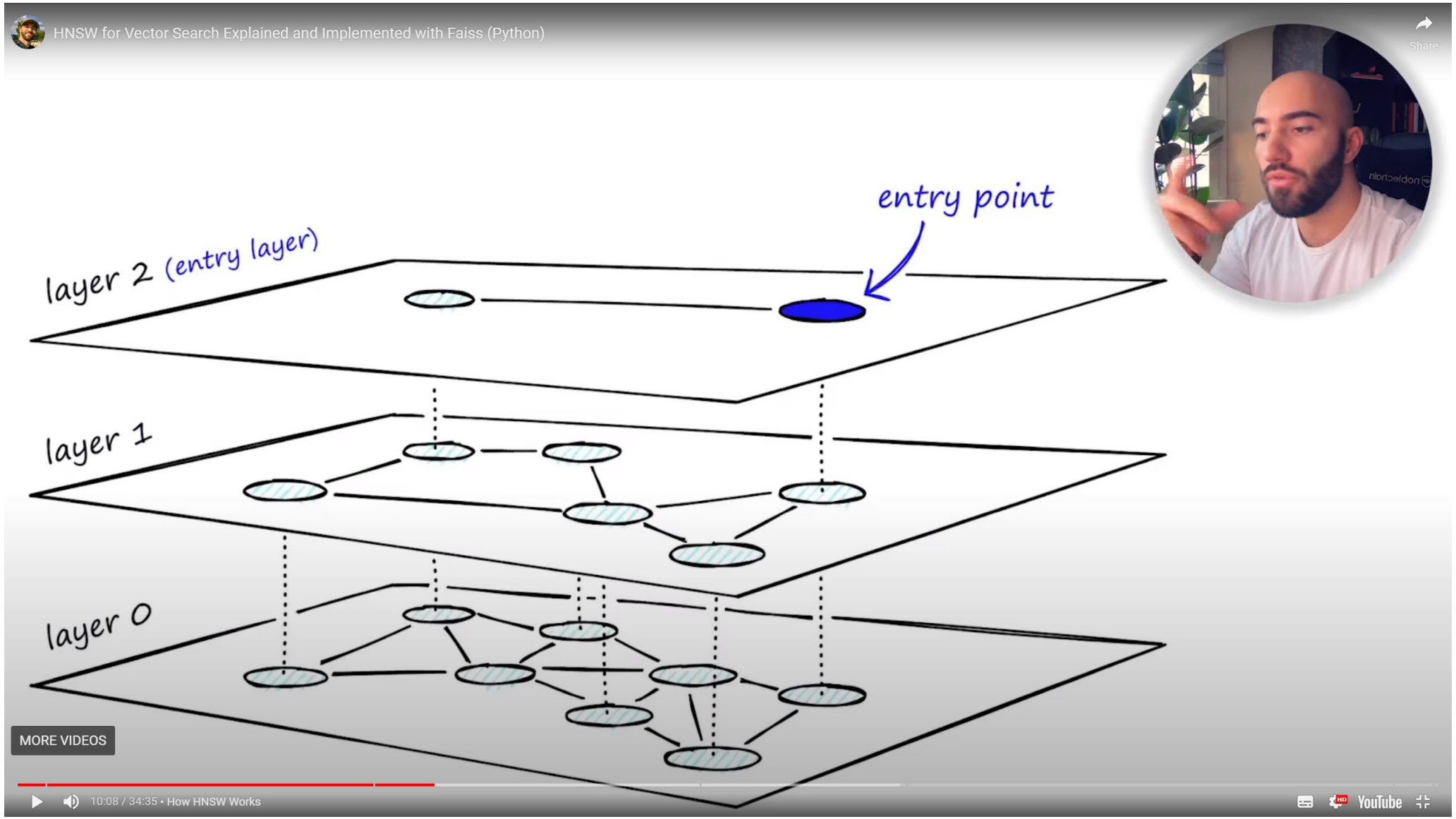


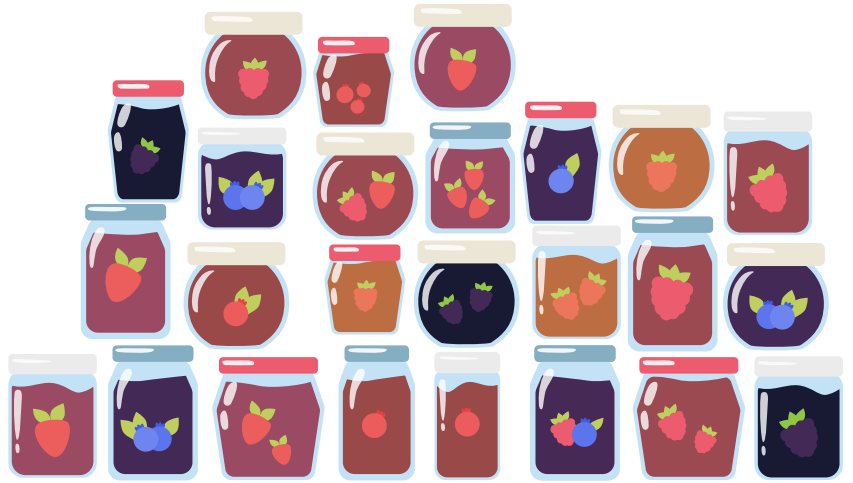
Estimated ratings

Co-clustering utilise clusters of similar scores among both users and movies simultaneously



Find clusters that minimise the difference between the estimated and true ratings

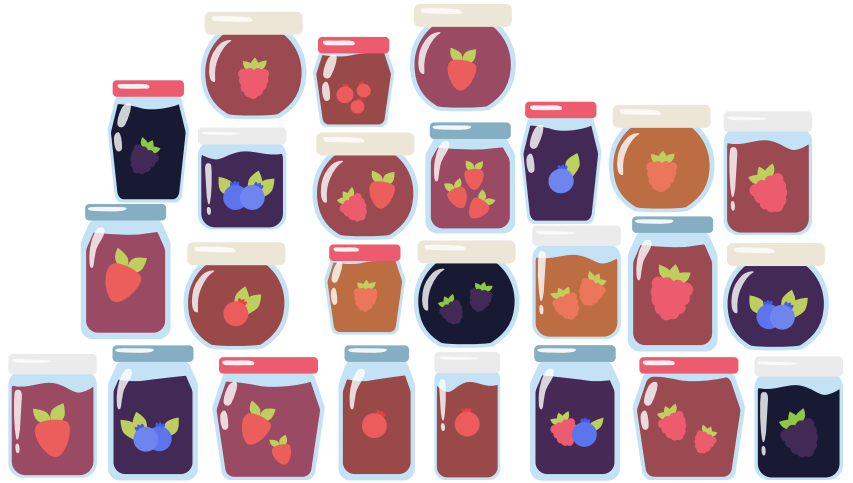

HNSW for Vector Search Explained and Implemented with Faiss (Python)
Share
entry point
layer 2 (entry layer)
layer 1
layer 0
MORE VIDEOS
10:08 / 34:35 • How HNSW Works
YouTube



Booth A

Booth B

	Booth A		Booth B
Walked by	242		260
Stopped to taste	104 (40 %)		145 (60 %)



Booth A

Booth B

Booth A		Booth B
242	Walked by	260
(40 %) 104	Stopped to taste	145 (60 %)
(30 %) 31	Bought jam	4 (3 %)