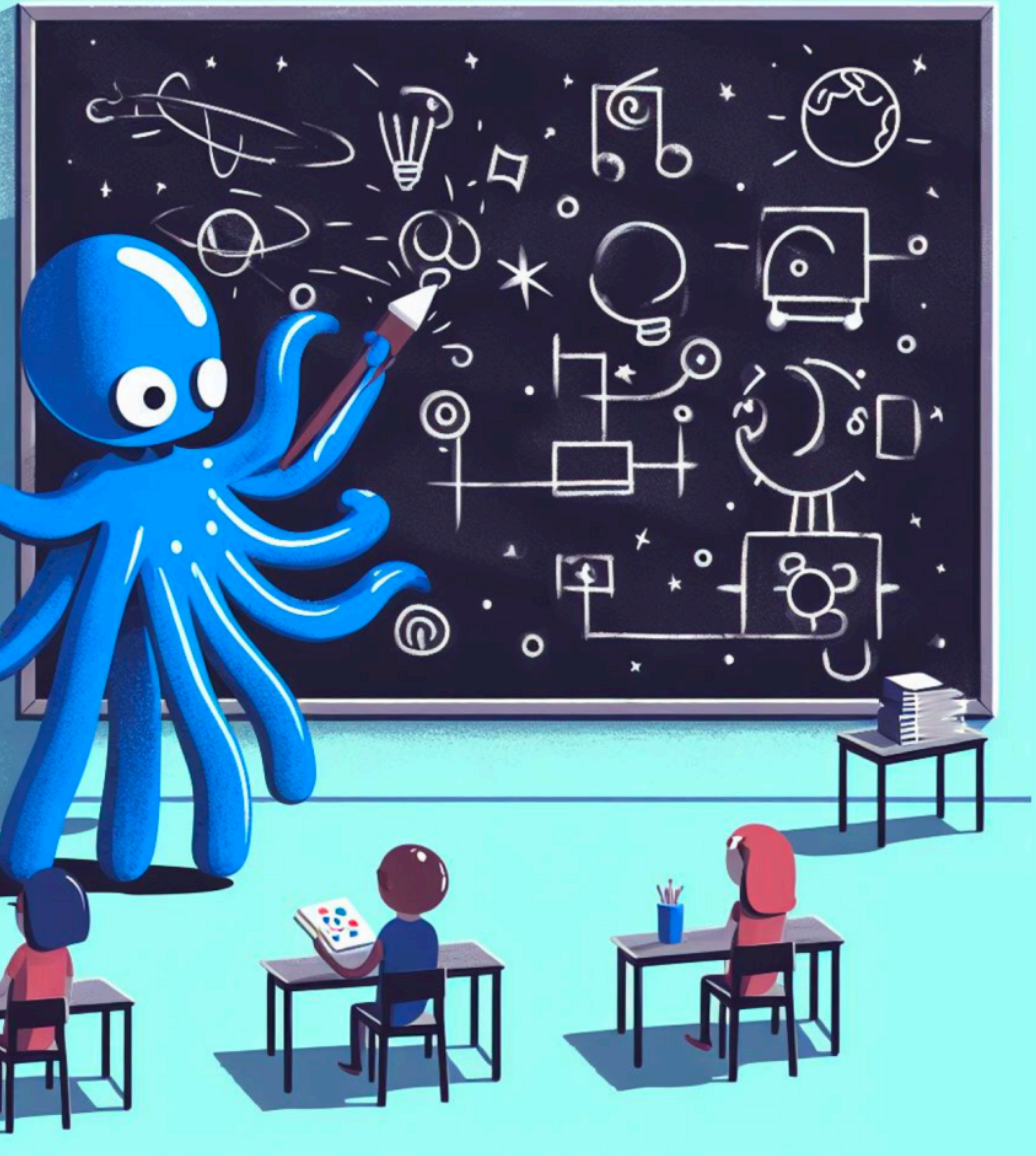


Machine Learning Fortnight 2023

Introduction to Recommender Systems

📍 LB 5173.0055
🗓️ 20 / 11 / 2023
⌚ 15:00 - 17:00

Fully Connected Graph

Fully Connected Graph

Machine Learning Fortnight

Introduction to Recommender Systems

Who are we?

ML Month

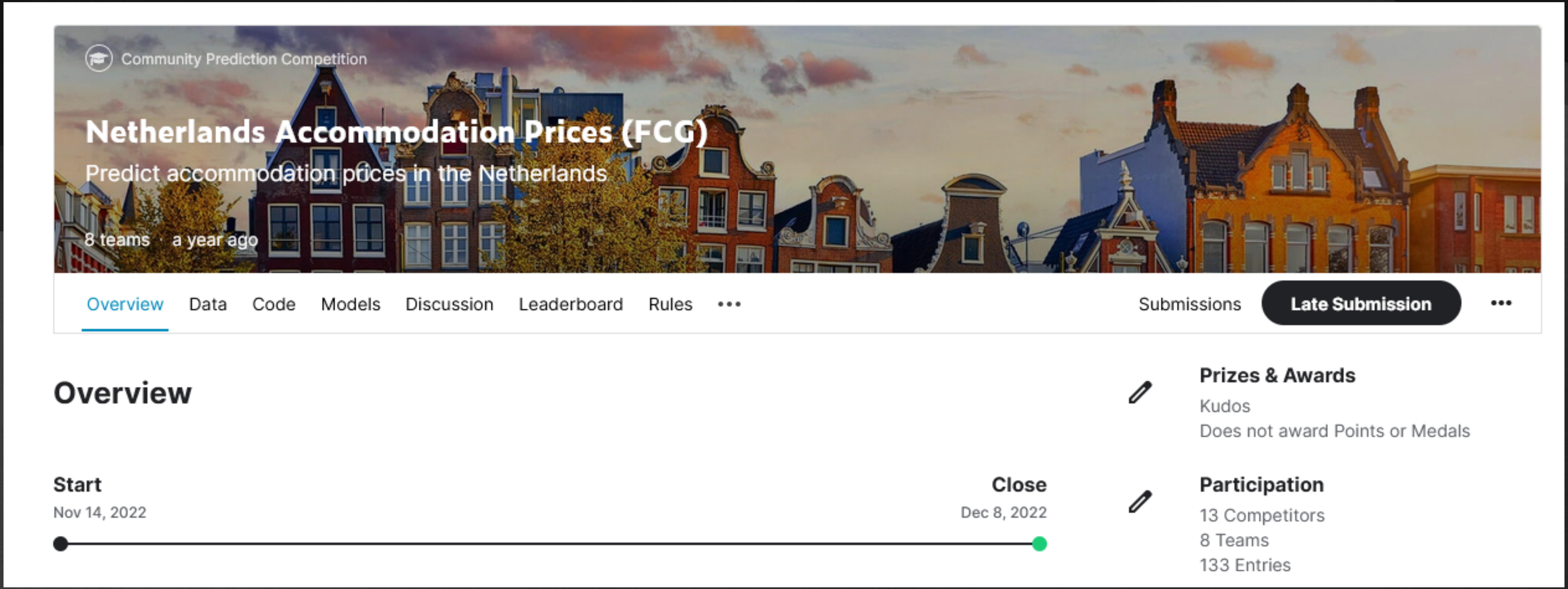
GAPC

Our Wonderful Team

Fully Connected Graph

Last year…

Predicting housing prices in the Netherlands



This year… video games are amazing!

But which one should you play?

Fortnight overview

Introduction to Recommender Systems

20 Nov

NLP in RecSys

24 Nov

Weaviate Vector Databases

27 Nov

Finale & Award Ceremony

1 Dec

Contents

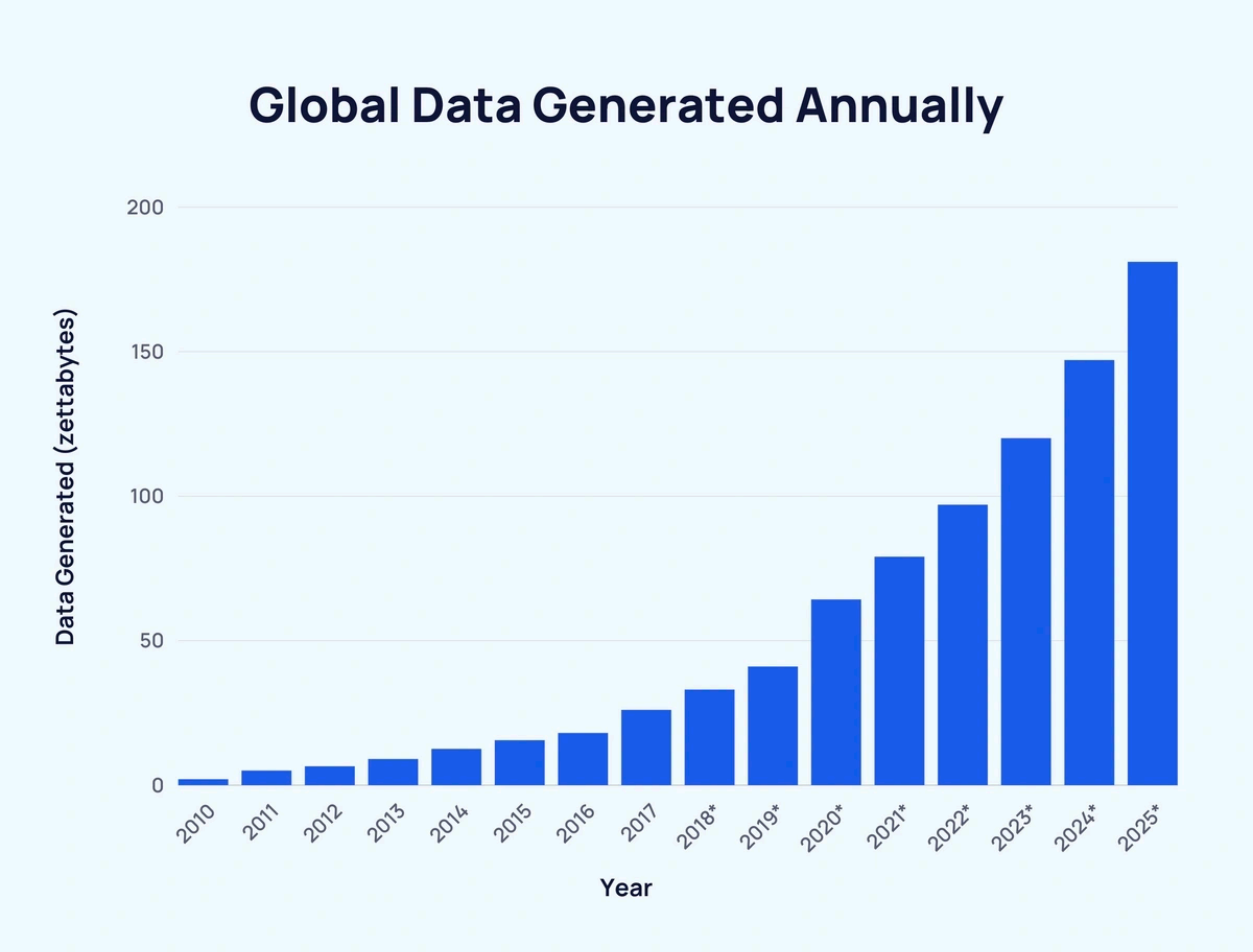
- What Are Recommender Systems?
- Types of Recommender Systems
  - Content filtering
  - Collaborative filtering
  - Hybrid filtering
- Recommendation Algorithms
- **Practical Session**

Prerequisites

Familiar with

- Machine Learning
- Linear Algebra (inner product, matrix vector product)
- Some experience with Tensorflow and pandas

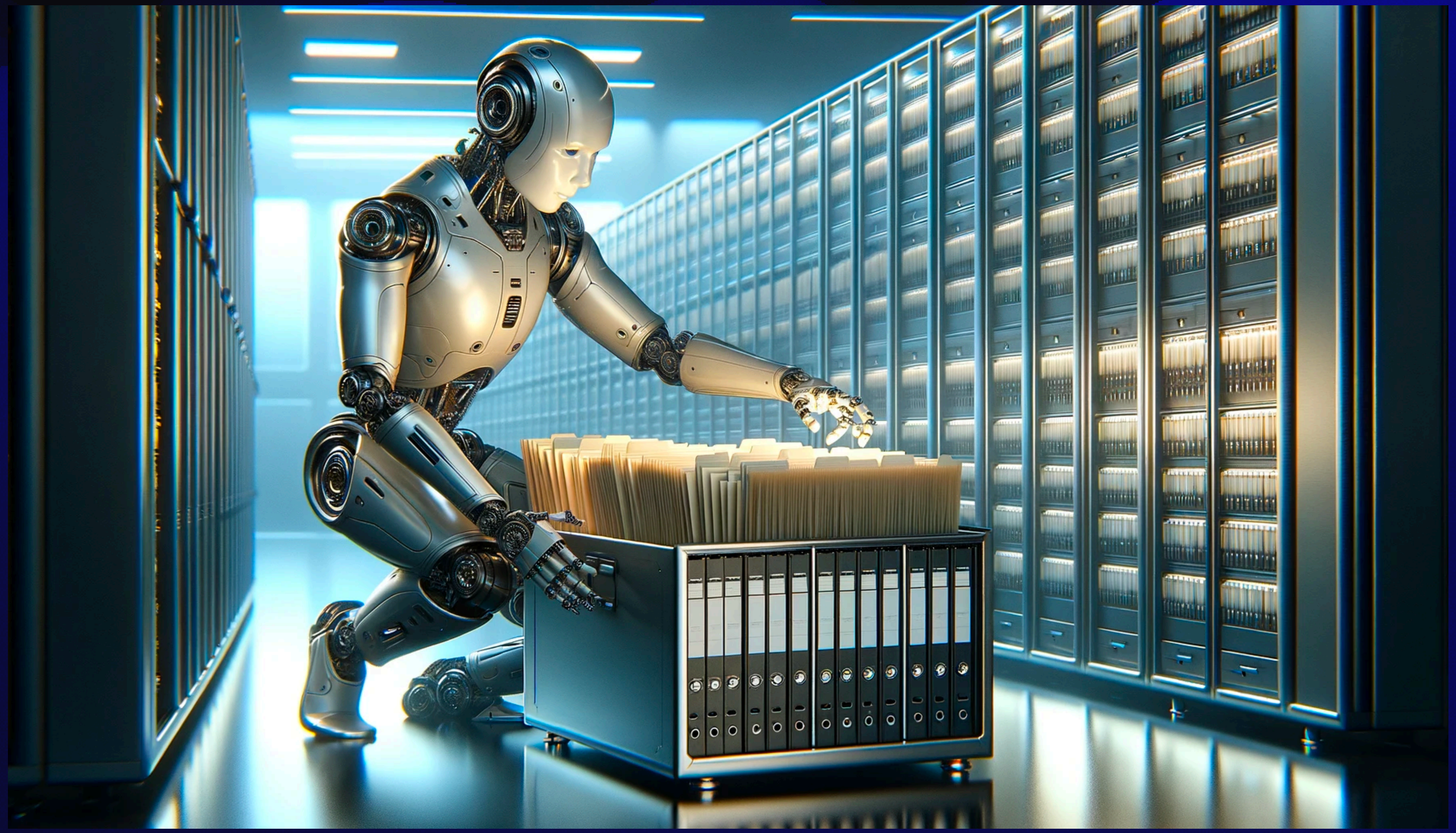
The world generates **2.5 quintillion bytes** per day. That's 1,000 petabytes!



(Source:Statista.com)

Recommender Systems are methods to suggest relevant content to users based on their preferences and behavior.

They *filter* the information

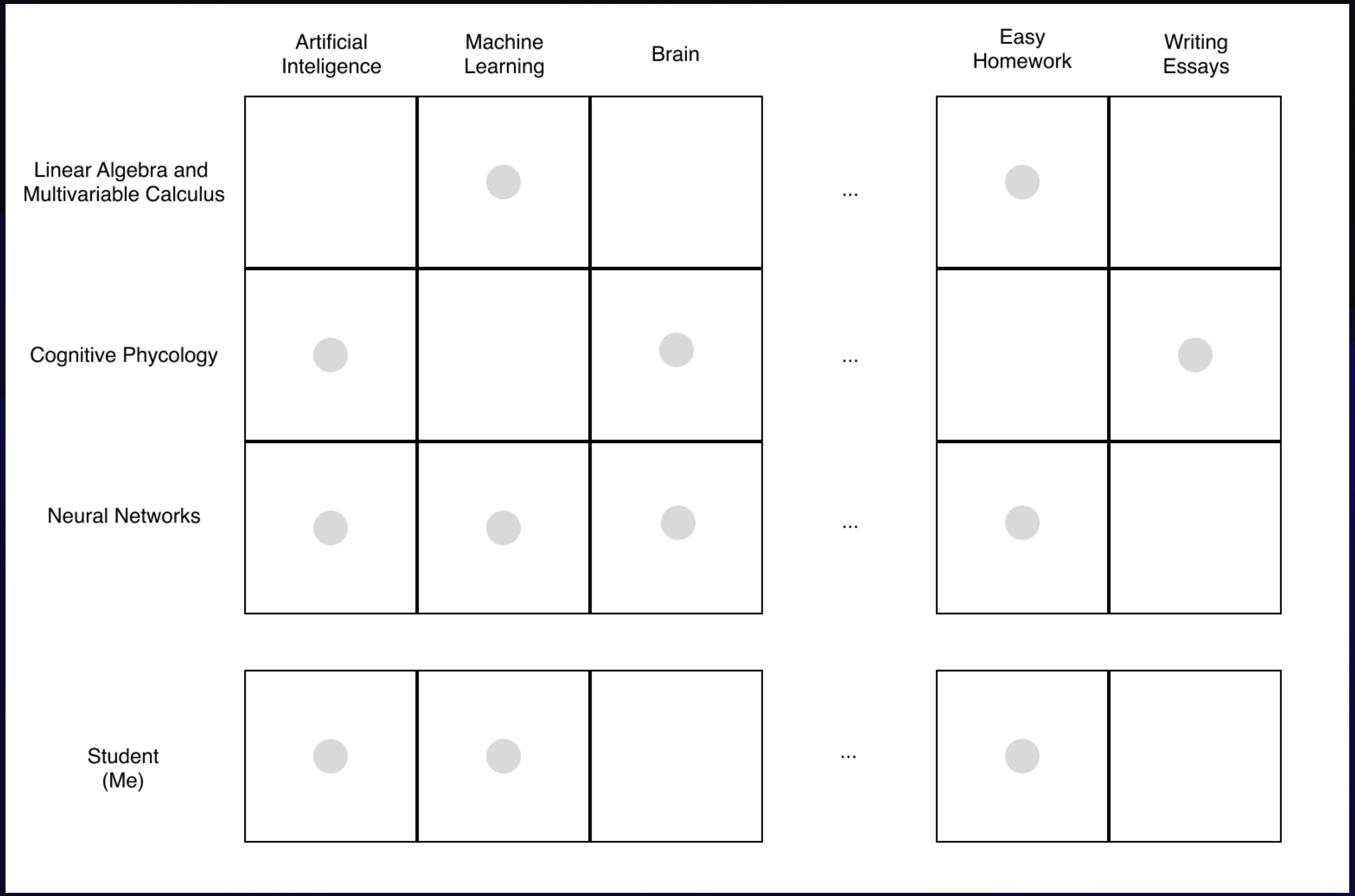


Recommender systems are everywhere…

Content Based Filtering

Content-based filtering recommends items similar to those the user has shown interest in based on item features.

Content Based Filtering



$$\langle x, y \rangle = \sum_{i=1}^{d} x_i y_i$$

```python
import numpy as np

courses = np.array([
    [0, 1, 0, 1, 0],
    [1, 0, 1, 0, 1],
    [1, 1, 1, 1, 0]
])

student = np.array([
    [1, 1, 0, 1, 0]
])

print(np.dot(student, courses.T))

# Output: [[2 1 3]]
```

Content Based Filtering

Advantages:

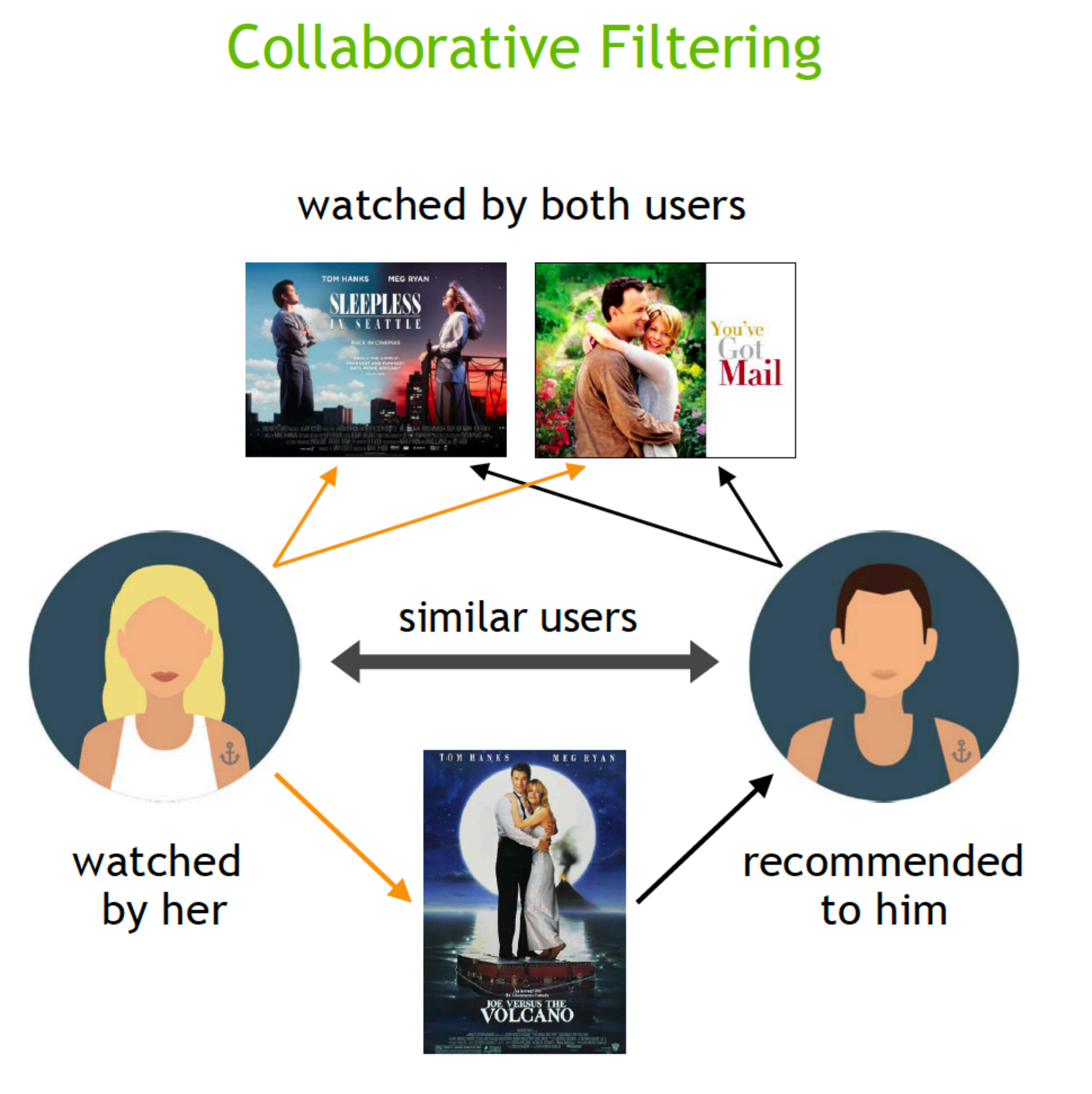
- No need for data about other users; recommendations are user-specific.
- Scalable to a large number of users.
- Can recommend niche items tailored to individual preferences.

Disadvantages:

- Requires domain knowledge for hand-engineering item features.
- Limited ability to expand on users' existing interests; recommendations are based on current preferences.

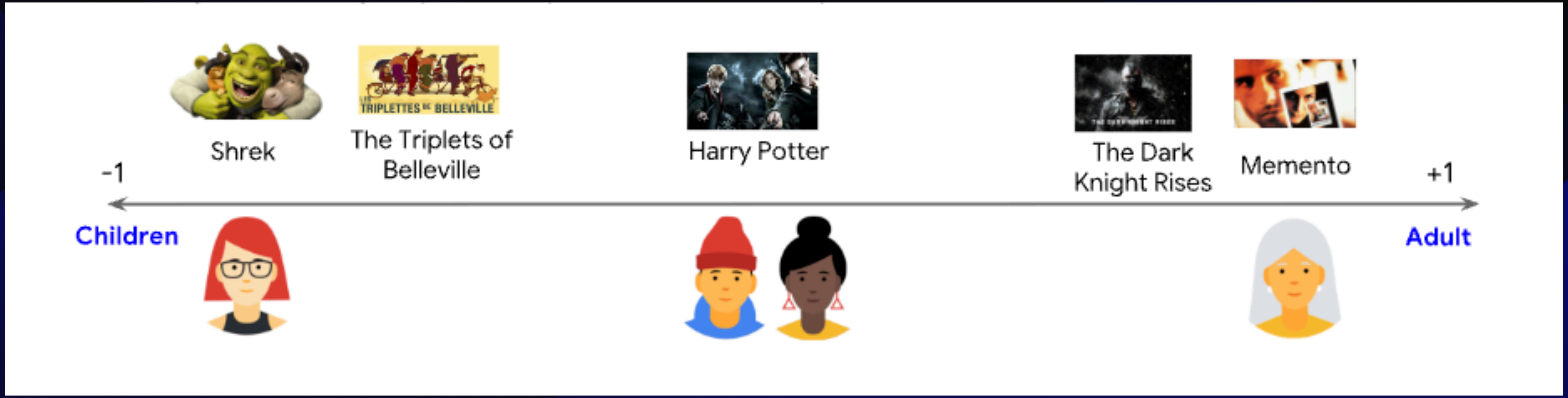
Collaborative Filtering

Collaborative filtering uses past similar decisions to predict future selections based on shared preferences.



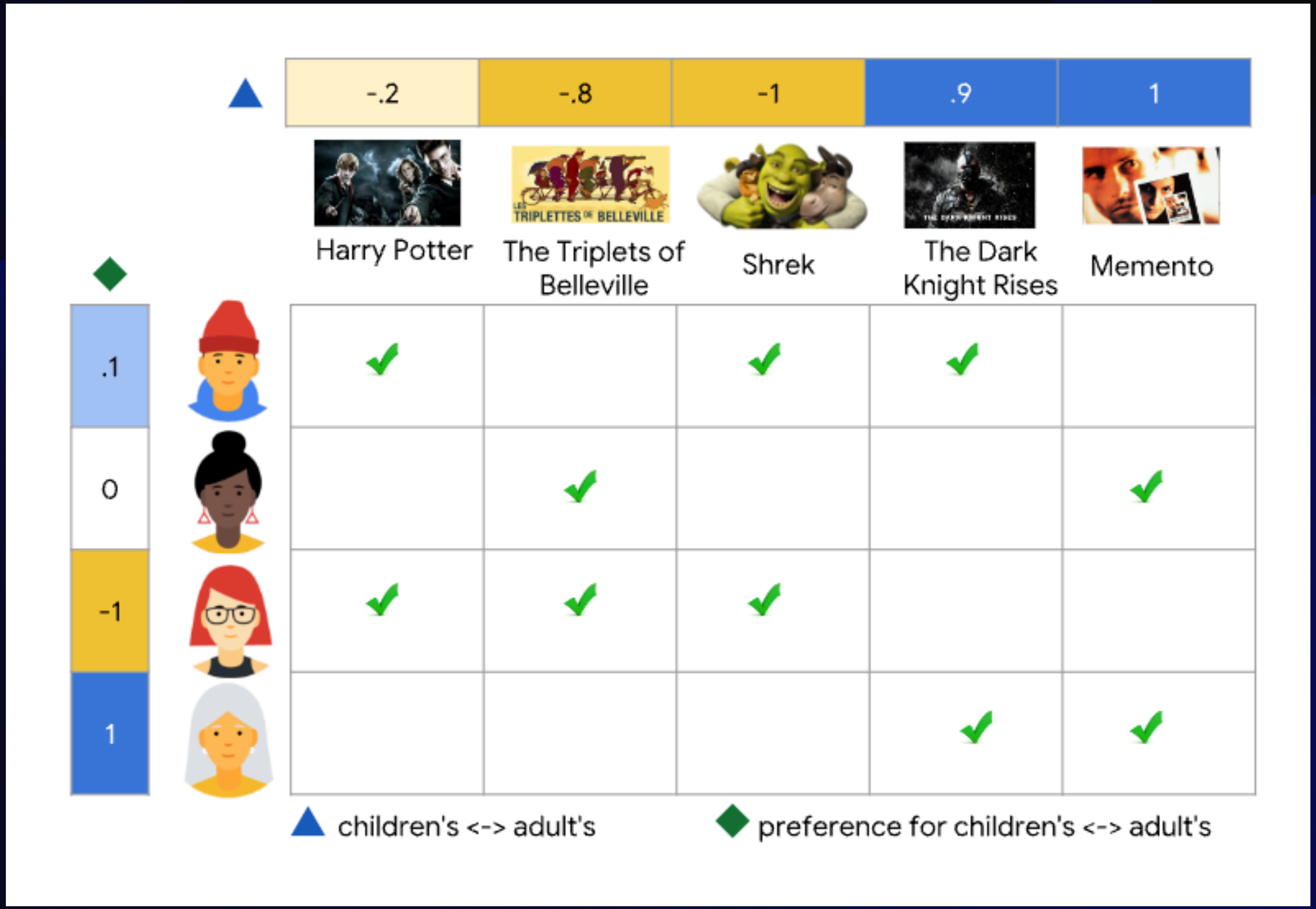
Collaborative Filtering

1D embedding



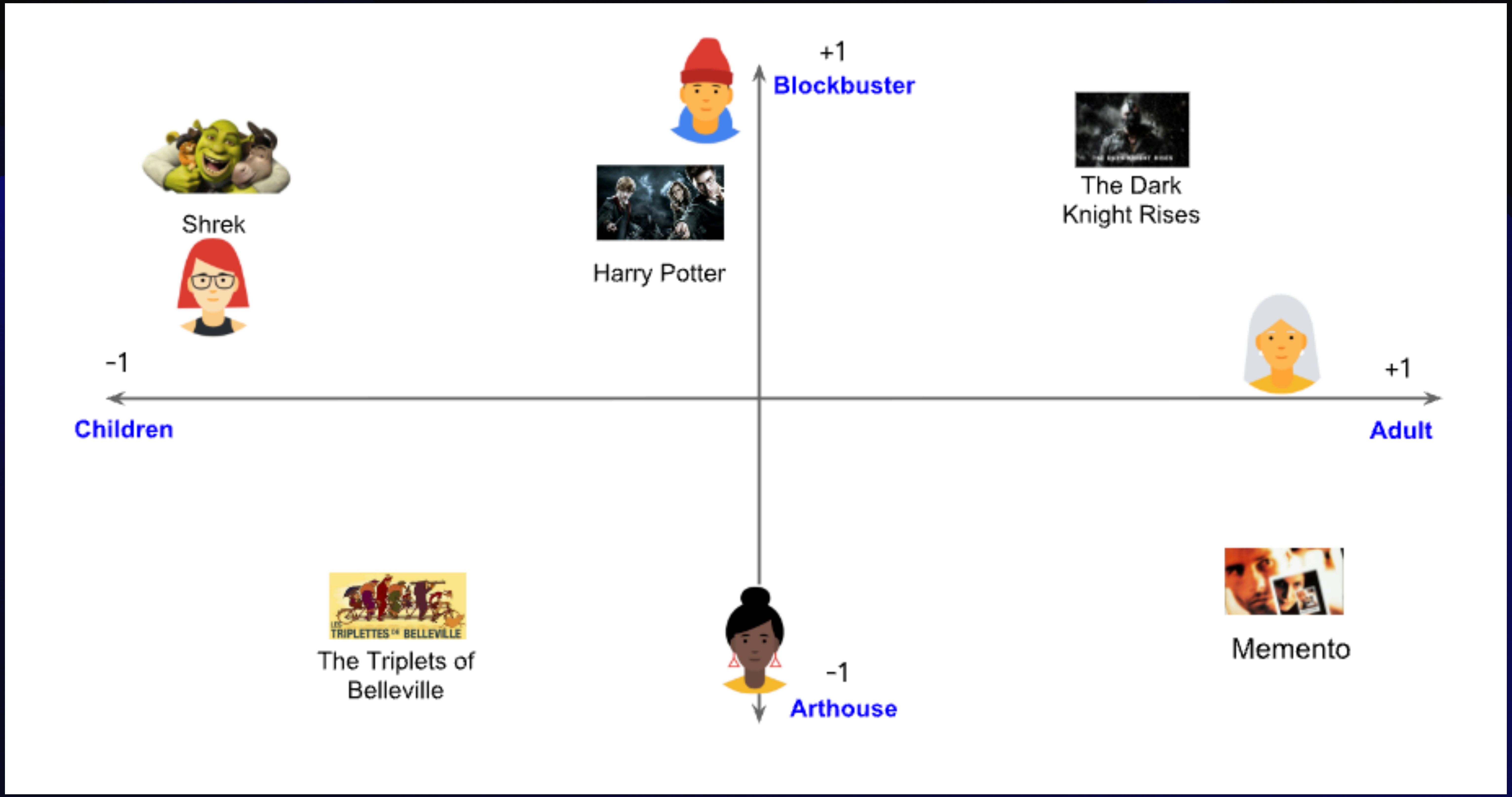
Collaborative Filtering

1D embedding



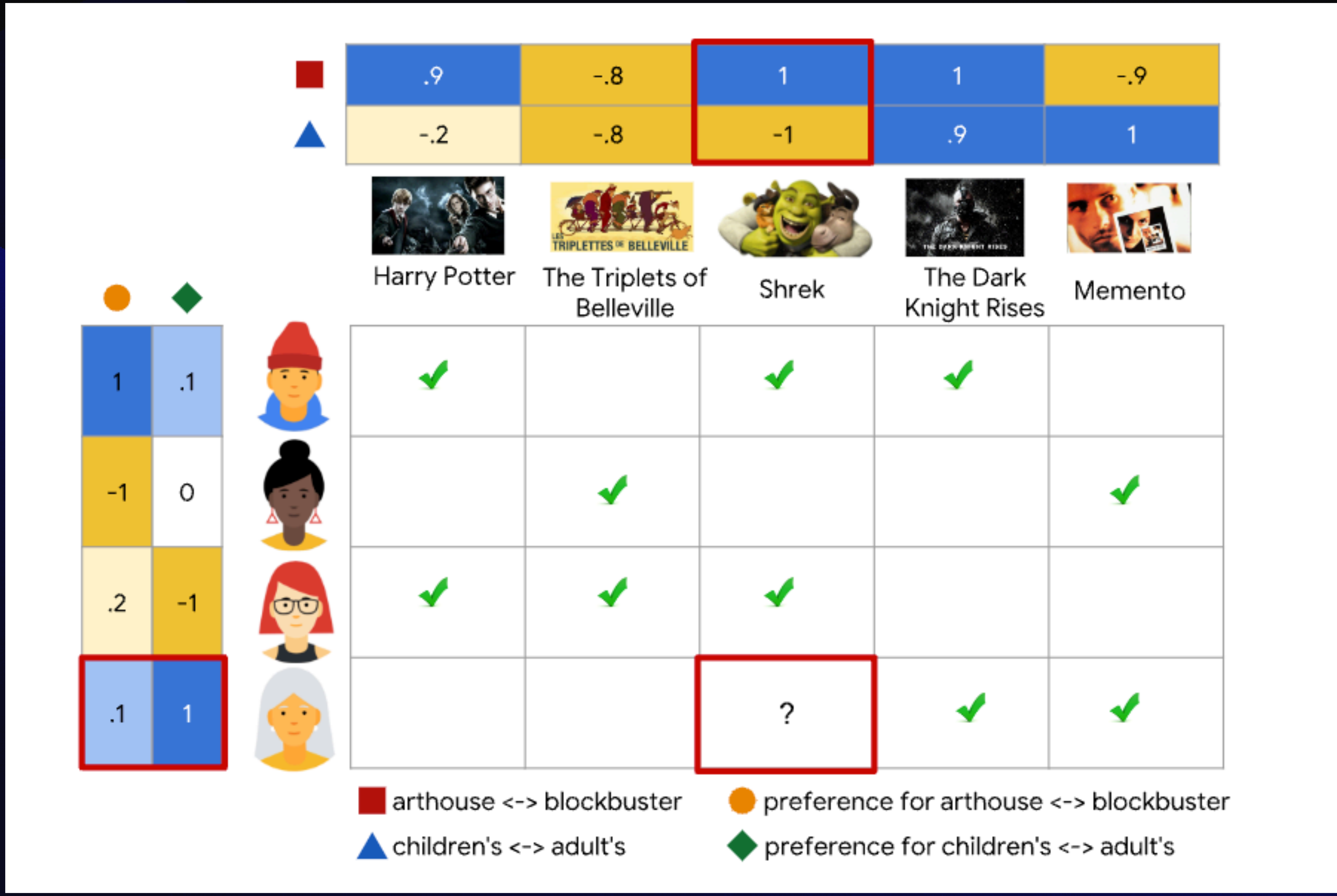
Collaborative Filtering

2D embeddings



Collaborative Filtering

2D embeddings



Collaborative Filtering

Matrix Factorization

Matrix Factorization

Matrix factorization is a simple embedding model. Given the feedback matrix A $\in R^{m \times n}$, where m is the number of users (or queries) and n is the number of items, the model learns:

- A user embedding matrix $U \in \mathbb{R}^{m \times d}$, where row i is the embedding for user i.
- An item embedding matrix $V \in \mathbb{R}^{n \times d}$, where row j is the embedding for item j.

Collaborative Filtering

Matrix Factorization

Matrix Factorization

Matrix factorization is a simple embedding model. Given the feedback matrix A $\in R^{m \times n}$, where m is the number of users (or queries) and n is the number of items, the model learns:

- A user embedding matrix $U \in \mathbb{R}^{m \times d}$, where row i is the embedding for user i.
- An item embedding matrix $V \in \mathbb{R}^{n \times d}$, where row j is the embedding for item j.

$$\min_{U \in \mathbb{R}^{m \times d},\ V \in \mathbb{R}^{n \times d}} \sum_{(i,j) \in \text{obs}} (A_{ij} - \langle U_i, V_j \rangle)^2.$$

SVD

1	0	1	1	0
0	1	0	0	1
1	1	1	0	0
0	0	0	1	1

$$|A - U V^T|_F^2$$
$$= \Sigma_{(i,\, j)} (A_{ij} - U_i \,.\, V_j)^2$$

The task is similar to ...

However, it is not efficient enough :(

Collaborative Filtering

Matrix Factorization

Matrix Factorization

Matrix factorization is a simple embedding model. Given the feedback matrix A $\in R^{m \times n}$, where m is the number of users (or queries) and n is the number of items, the model learns:

- A user embedding matrix $U \in \mathbb{R}^{m \times d}$, where row i is the embedding for user i.
- An item embedding matrix $V \in \mathbb{R}^{n \times d}$, where row j is the embedding for item j.

$$\min_{U \in \mathbb{R}^{m \times d},\ V \in \mathbb{R}^{n \times d}} \sum_{(i,j) \in \text{obs}} (A_{ij} - \langle U_i, V_j \rangle)^2.$$

Weighted Matrix Factorization:

$$\min_{U \in \mathbb{R}^{m \times d},\ V \in \mathbb{R}^{n \times d}} \sum_{(i,j) \in \text{obs}} (A_{ij} - \langle U_i, V_j \rangle)^2 + w_0 \sum_{(i,j) \notin \text{obs}} (\langle U_i, V_j \rangle)^2.$$

Collaborative Filtering

Matrix Factorization (Optimization)

Stochastic Gradient Descent (SGD)

generic algorithm to minimize loss functions.

Optimize *U* and *V* matrices together

Alternating Least Squares (ALS)

specialized to this particular objective.

- Fix U, optimize V
- Fix V, optimize U

Collaborative Filtering

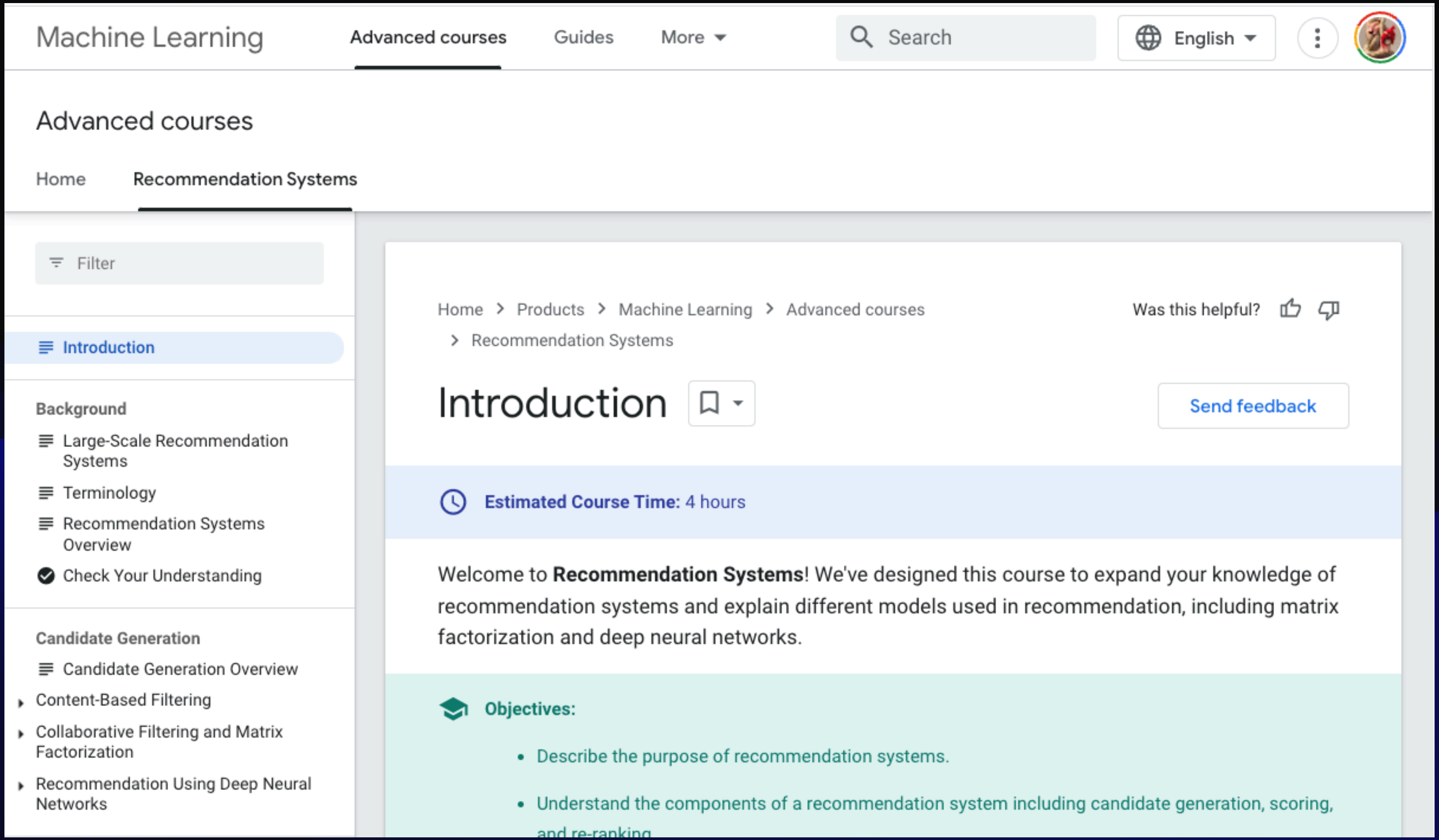
Matrix Factorization (SGD vs ALS)

SGD

- 👍 Very flexible—can use other loss functions.
- 👍 Can be parallelized.
- 👎 Slower—does not converge as quickly.
- 👎 Harder to handle the unobserved entries (need to use negative sampling or gravity).

ALS

- 👎 Reliant on Loss Squares only.
- 👍 Can be parallelized.
- 👍 Converges faster than SGD.
- 👍 Easier to handle unobserved entries.



https://developers.google.com/machine-learning/recommendation

Practical Session :D

Break till …

ML Fortnight 2023

- Register yourself
- Read the Kaggle page
- Build a recommender system
- Evaluate your model
- Submit your results
- Iterate and improve!