EXPLORE AI

Recommender Systems Pt 1

Content-based filtering

27 Feb 2024

Objectives

By the end of this session you should be able to...

- Describe and differentiate between content and collaborative based filtering at a high level
- Explain cosine similarity and a cosine similarity matrix
- Explain how a content-based filtering model generates recommendations or rating predictions

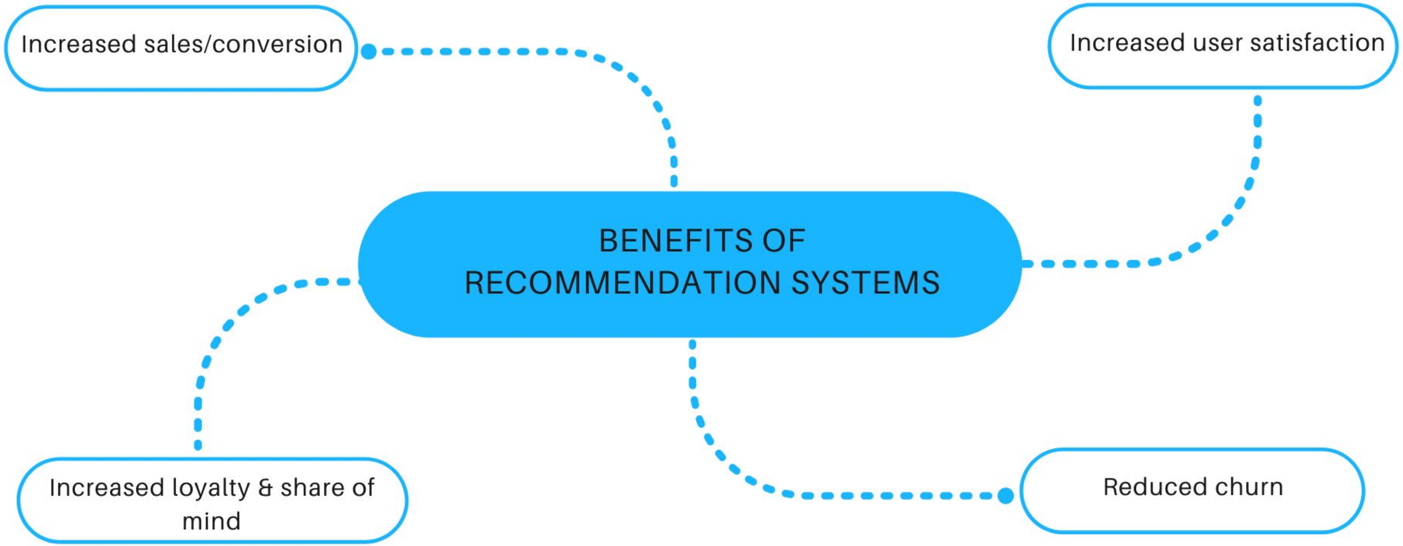
Recommender systems

- Information filtering system that gives suggestions to users based on different factors
- Such as Netflix, Spotify, Youtube, Amazon, etc.
- Incredibly valuable to business and customers

Content-based filtering

Collaborative filtering

Recommender systems



Recommender systems

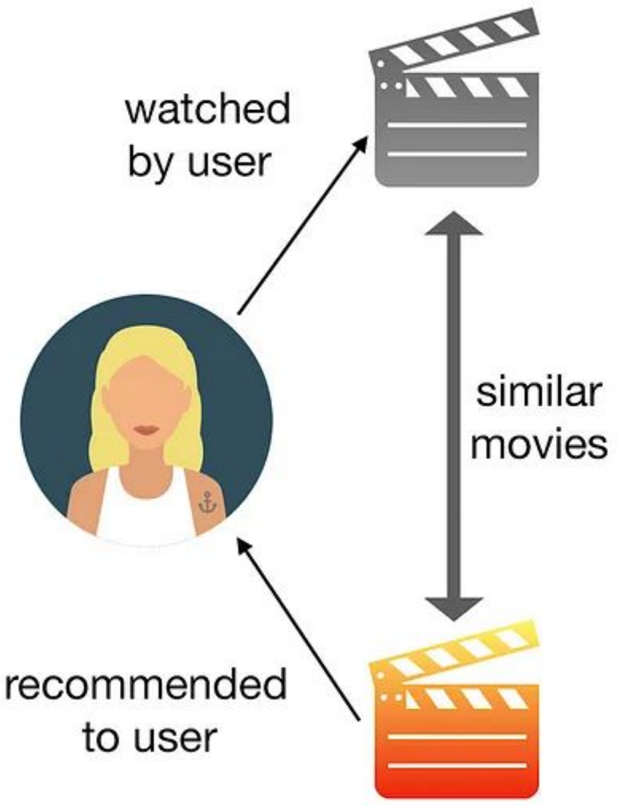
Common terminology

- **Item:** something which is consumed
  - Movies, books, songs, products, etc.
  - Properties do not change
- **User**: a consumer, interact with the **items** in our system
- **Ratings**: created by **users** for specific **items** in our system
  - Explicit or implicit

Recommender systems...briefly!

Content-based recommender system

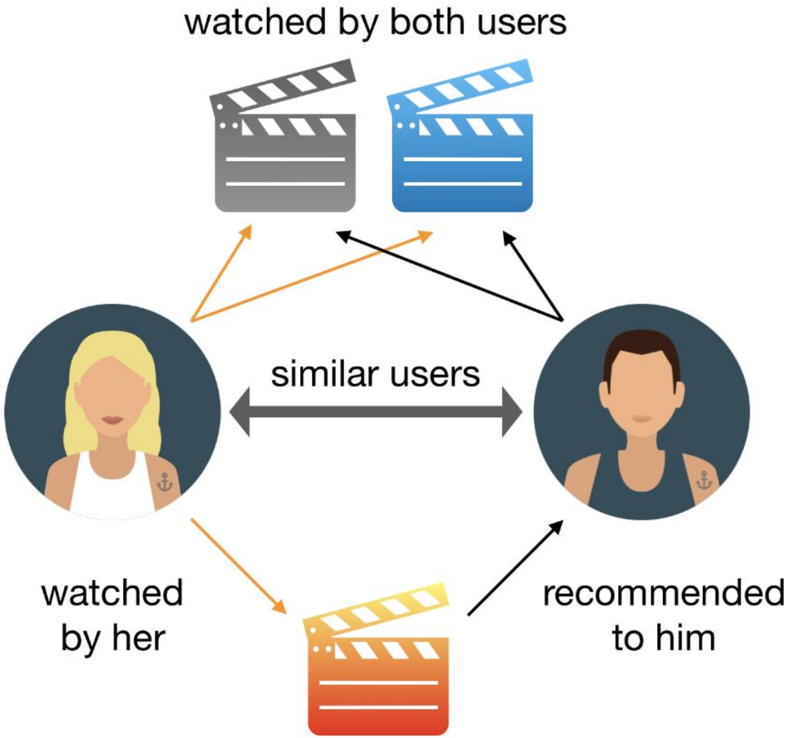
- Aims to identify what a user may like based on their history
- Considers information about the item the user interacted with or rated highly, and will suggest other items with similar characteristics
- Using **cosine similarity**, we can compare items, and predict how our user would rate that item



Recommender systems...briefly!

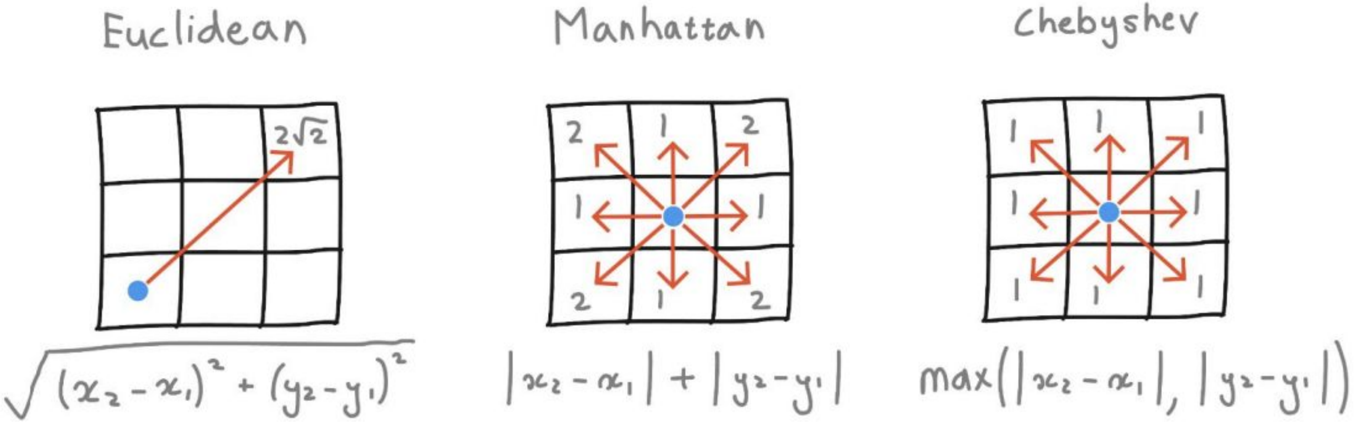
Collaborative-based recommender system

- Movies are recommended to a user based on similar user interests
- Again, uses **cosine similarity** to compare users
- We can calculate how similar other users are to a particular user (User A)
- Based on how other users rated an item, we can predict how User A might rate that item
- If we think they would rate it well, we can recommend it!



Similarity

Previously...



Similarity

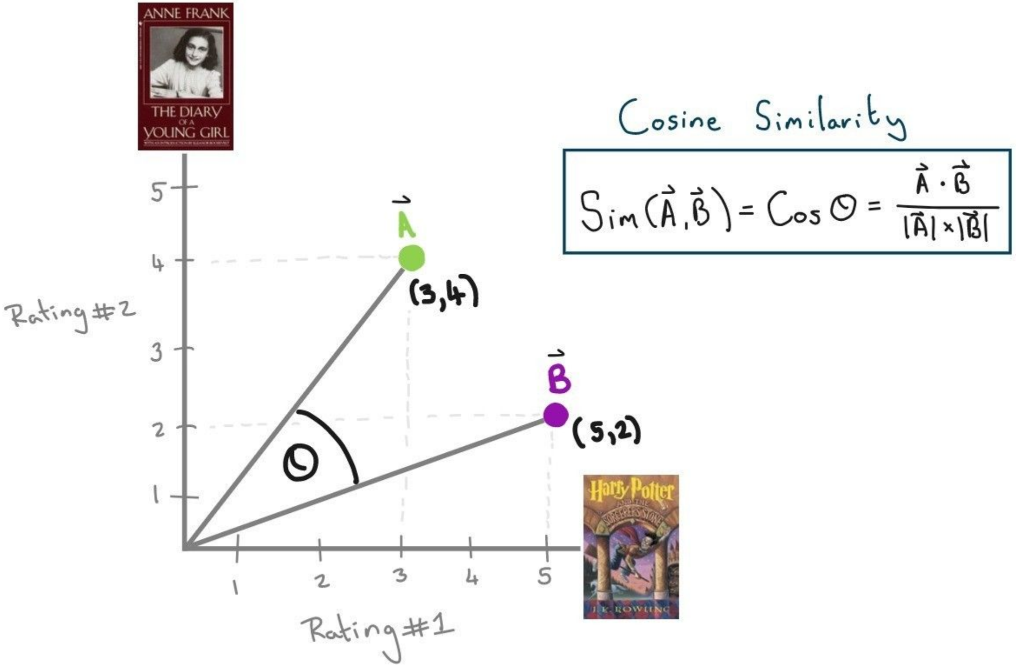
Cosine similarity

- Similarity between vectors is measured by the cosine of the angle between the vectors
- Smaller angle = more similar vectors

$$Sim(A,B) = \frac{A \cdot B}{||A|| \times ||B||}$$

$$= \frac{(3 \times 5) + (4 \times 2)}{\sqrt{9+16} \times \sqrt{25+4}}$$

$$= 0.854$$



The Cosine Similarity Matrix

- A matrix of each of our items mapped against each other, with a value for each pairing indicating how similar they are.
- A cosine similarity of 1 would mean the items are completely similar - this occurs on the diagonals of our matrix, where each movie items itself.

	Item A	Item B	Item C	Item D
Item A	1	0.85	0.23	0.54
Item B	0.85	1	0.65	0.44
Item C	0.23	0.65	1	0.14
Item D	0.54	0.44	0.14	1

Content-based model

Generating **top-N recommendations**

1. Select an initial item to generate recommendations from.
2. Extract all the similarity values between the initial item and each other item in the similarity matrix.
3. Sort the resulting values in descending order (most similar to least similar).
4. Select the top N similarity values, and return the corresponding item details to the user

	Item A	Item B	Item C	Item D
Item A	1	0.85	0.23	0.54
Item B	0.85	1	0.65	0.44
Item C	0.23	0.65	1	0.14
Item D	0.54	0.44	0.14	1

Content-based model

Generating **top-N recommendations**

1. Select an initial item to generate recommendations from.
2. Extract all the similarity values between the initial item and each other item in the similarity matrix.
3. Sort the resulting values in descending order (most similar to least similar).
4. Select the top N similarity values, and return the corresponding item details to the user

Top 2 recommendations...

	Item A	Item B	Item C	Item D
Item A	1	0.85	0.23	0.54
Item C	0.23	0.65	1	0.14
Item D	0.54	0.44	0.14	1

Content-based model

Generating **rating predictions**

1. Select a reference user from the database and a reference item they have not rated.
2. For the user, gather the similarity values between the reference item and each item the user has rated.
3. Sort the gathered similarity values in descending order (most similar to least similar).
4. Select the k highest similarity values which are above a given threshold value, creating a collection K
5. Compute a weighted average rating from these values, which is the sum of the similarity values of each item multiplied by its assigned user-rating, divided by the sum of the similarity values.

	Item A	Item B	Item C	Item D
User 1	3.0	4.0	1.0	?

Content-based model

Generating **rating predictions**

	Item A	Item B	Item C	Item D
User 1	3.0	4.0	1.0	?

	Item A	Item B	Item C	Item D
Item A	1	0.85	0.23	0.54
Item B	0.85	1	0.65	0.44
Item C	0.23	0.65	1	0.14
Item D	0.54	0.44	0.14	1

Content-based model

Generating **rating predictions**

	Item A	Item B	Item C	Item D
User 1	3.0	4.0	1.0	?

Threshold = 0.3

	Item A	Item B	Item C	Item D
Item A	1	0.85	0.23	0.54
Item B	0.85	1	0.65	0.44
Item C	0.23	0.65	1	0.14

Content-based model

Generating **rating predictions**

$$weighted\ average = \frac{(sim_A * rating_A) + (sim_B * rating_B)}{sim_A + sim_B}$$

	Item A	Item B	Item C	Item D
User 1	3.0	4.0	1.0	?

	Item A	Item B	Item C	Item D
Item A	1	0.85	0.23	0.54
Item B	0.85	1	0.65	0.44
Item C	0.23	0.65	1	0.14

$$\frac{(0.54 * 3.0) + (0.44 * 4.0)}{0.54 + 0.44} \approx 3.45$$

To the notebook!

Objectives

By the end of this session you should be able to...

- Describe and differentiate between content and collaborative based filtering at a high level
- Explain cosine similarity and a cosine similarity matrix
- Explain how a content-based filtering model generates recommendations or rating predictions

Recommender Systems Pt 2

Collaborative filtering

29 Feb 2024

1 / **Reminders…**

2 / **Collaborative-based filtering**

3 / **Memory-based approach**

4 / **Model-based approach**

Objectives

By the end of this session you should be able to...

- Describe and differentiate between content and collaborative based filtering
- Explain a utility matrix and how it is created
- Explain how a collaborative-based filtering approach generates recommendations or rating predictions

Reminders...

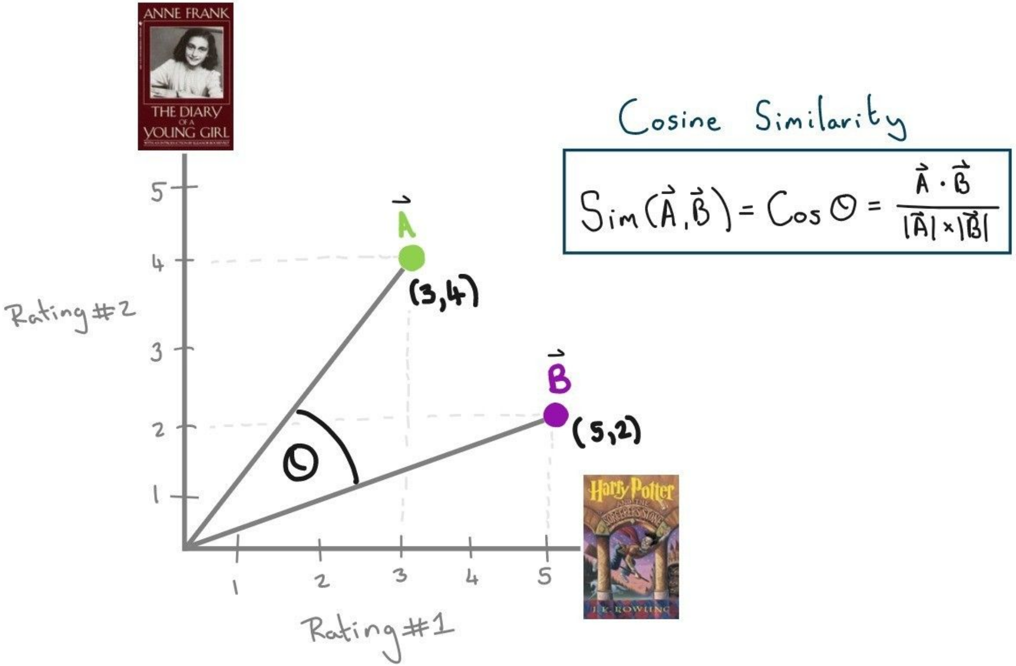
Cosine similarity

- Similarity between vectors is measured by the cosine of the angle between the vectors
- Smaller angle = more similar vectors

$$Sim(A,B) = \frac{A \cdot B}{||A|| \times ||B||}$$

$$= \frac{(3 \times 5) + (4 \times 2)}{\sqrt{9+16} \times \sqrt{25+4}}$$

$$= 0.854$$



Reminders...

Cosine similarity matrix

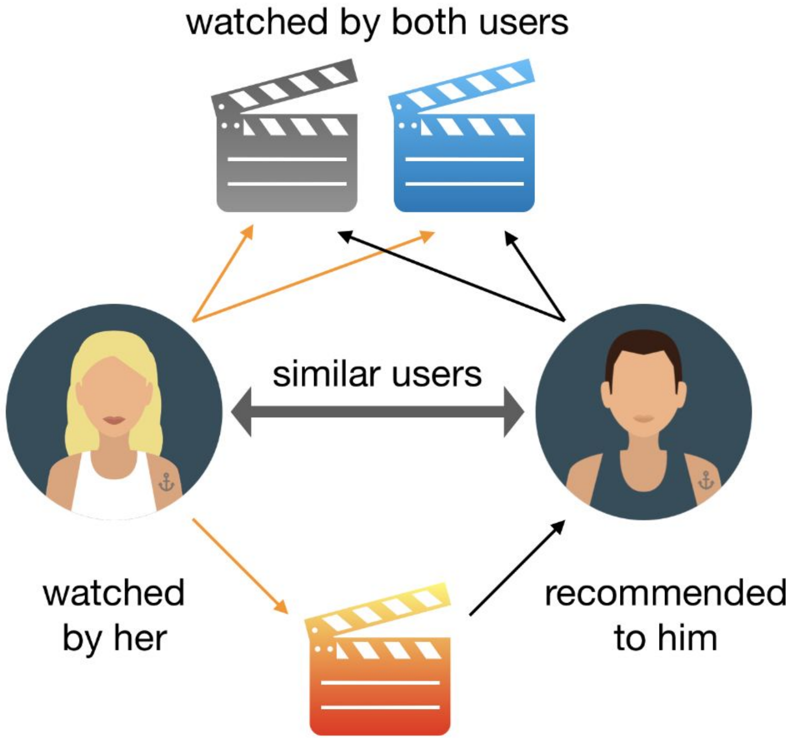
- A matrix of each of our items mapped against each other, with a value for each pairing indicating how similar they are.
- A cosine similarity of 1 would mean the items are completely similar - this occurs on the diagonals of our matrix, where each movie items itself.

	Item A	Item B	Item C	Item D
Item A	1	0.85	0.23	0.54
Item B	0.85	1	0.65	0.44
Item C	0.23	0.65	1	0.14
Item D	0.54	0.44	0.14	1

Reminders...

Collaborative-based recommender system

- Items are recommended to a user based on similar user interests
- We can calculate how similar other users are to a particular user (User A) - cosine similarity
- Similarity calculated on the basis of the rating a user gives to a movie.
  - Any other data about users or items, like the age of users or the genre of the movie are disregarded



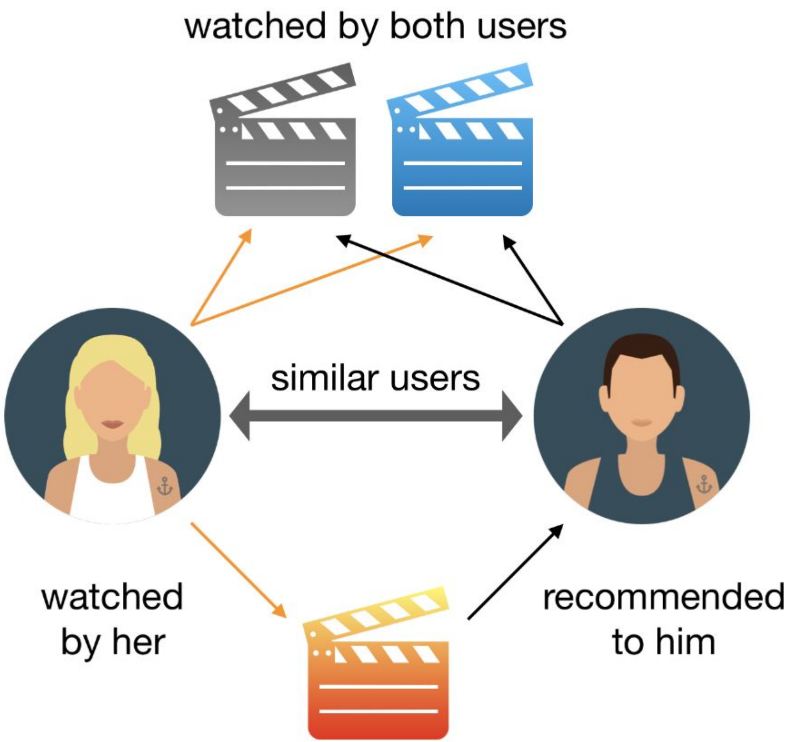
Collaborative-based filtering

Collaborative-based recommender system

There are **two main steps** in this approach:

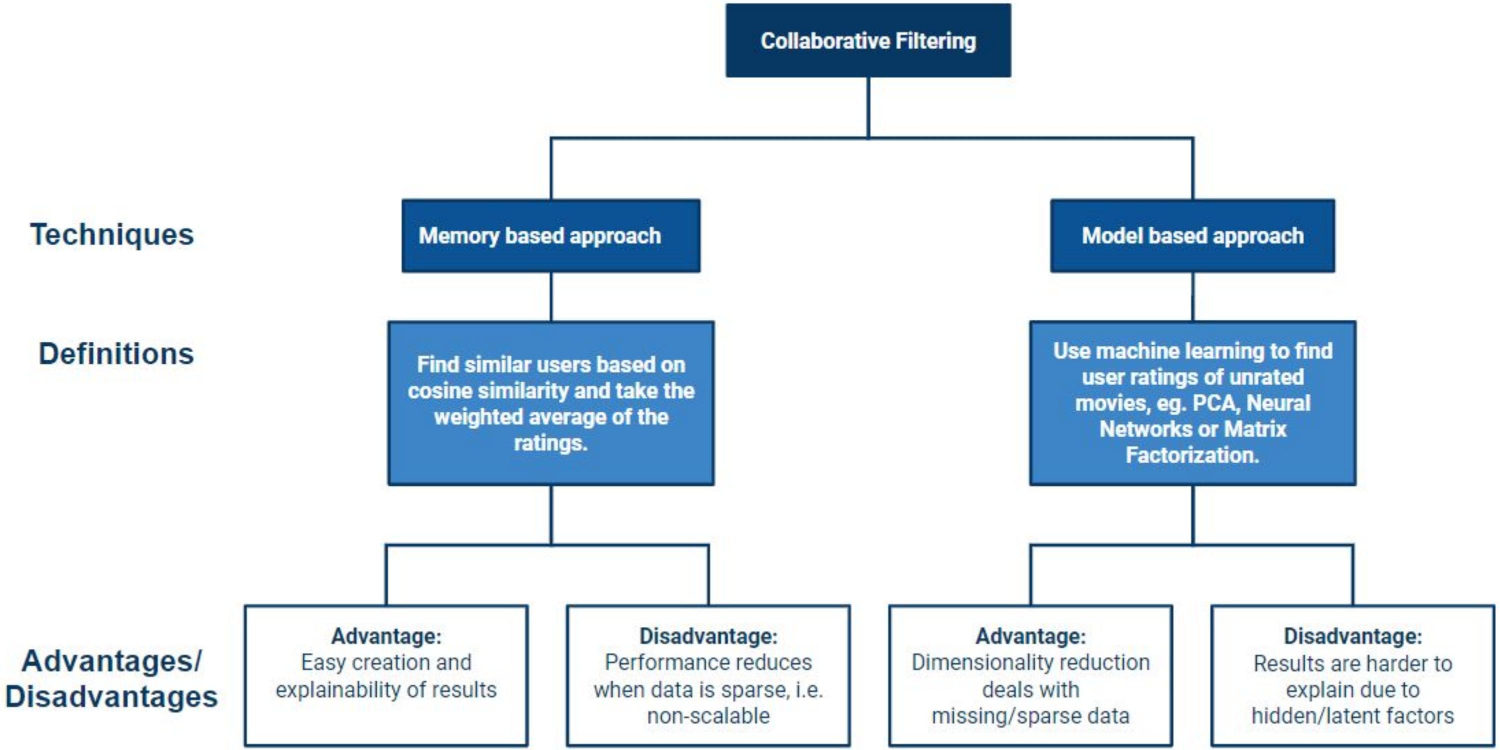
- 1. Find similar users
- 2. Predict the ratings users would give movies that they have not yet rated

We can then use these predictions to recommend movies that we think the user would give the highest ratings.



Collaborative-based filtering

Collaborative-based recommender system



Memory-based approach

Collaborative-based recommender systems

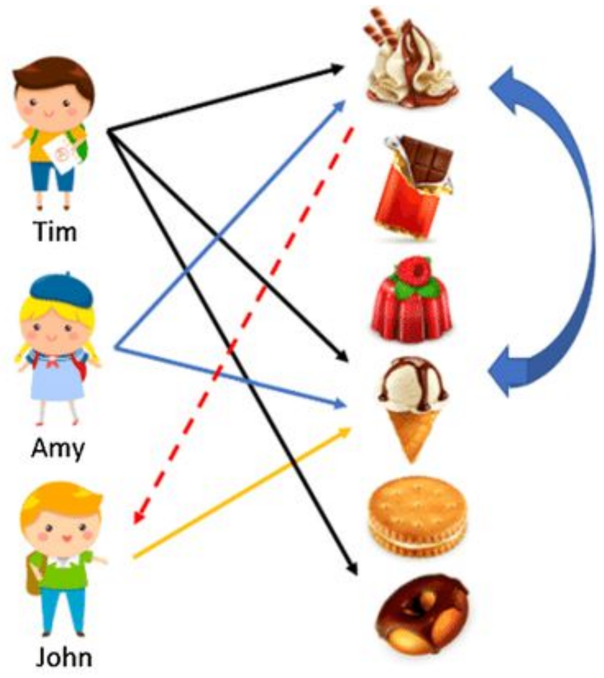
Memory based approaches for collaborative filtering can be divided into two types:

- **Item based** filtering (also known as **item-item** filtering)
- **User based** filtering (also known as **user-item** filtering)

Memory-based approach - item based filtering

Collaborative-based recommender systems

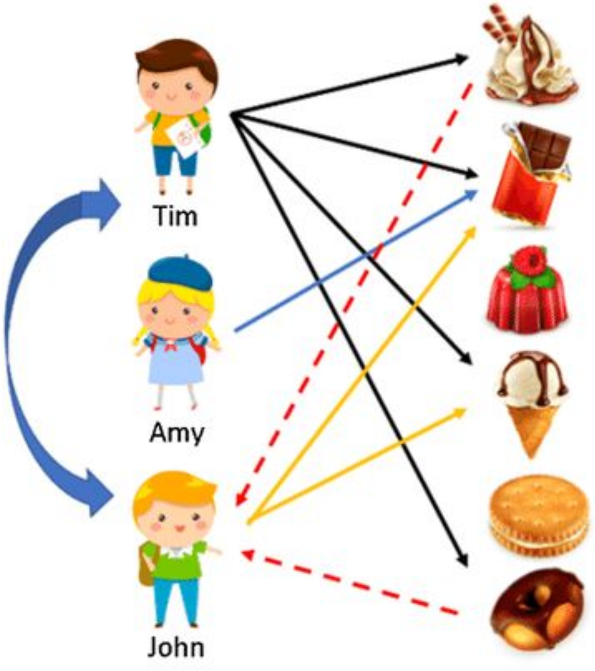
- Looks for items that are 'similar' to the items that a user has already rated
- **NB**: not the same as how we look at items in content based filtering.
    - not looking at their content or attributes (such as genre), but rather how users treat the two items the same.
- ***"Users who viewed this item also viewed..."***



Memory-based approach - user based filtering

Collaborative-based recommender systems

- Looks at the similarity of users
- ***"Users who are similar to you also liked A, B and C"***



The cold-start problem

Collaborative-based recommender systems

- What if no users are similar to our user?
- Or our user hasn't rated anything yet?
- When we cannot draw any inferences for users or items about which we have not yet gathered sufficient information
- We could:
    - recommend the **most popular** items on the platform
    - recommend them items with **similar attributes** if they have rated at least one

Utility Matrix

Collaborative-based recommender systems

User ID	Item ID	Rating
1	A	1
1	B	2
2	A	3
2	C	2
3	C	5

	Item A	Item B	Item C
User 1	1	2	NaN
User 2	3	NaN	2
User 3	NaN	NaN	5

Utility Matrix

Collaborative-based recommender systems

- Formal structure
- Inherently sparse - lots of missing data
  - Not every user has rated every item
  - Some items may not have been rated at all

	Item A	Item B	Item C
User 1	1	2	NaN
User 2	3	NaN	2
User 3	NaN	NaN	5

Memory based approach

- Construct cosine similarity matrix from utility matrix
- User-user similarity matrix

	User 1	User 2	User 3
User 1	1	0.85	0.26
User 2	0.85	1	0.55
User 3	0.26	0.55	1

Memory based approach

- **Top-N recommendations** based on similar users
- **Predicting ratings** for items using similar users' ratings

	User 1	User 2	User 3
User 1	1	0.85	0.26
User 2	0.85	1	0.55
User 3	0.26	0.55	1

Model based approach

- Based on **matrix factorization** - uses dimensionality reduction which makes it more computationally efficient
- Easier to use when data is large and sparse
- Tries to learn the features that describe the characteristics of ratings

- **surprise** package - designed specifically for recommender systems, and has several models within it that can be used in the model based approach.
- Check out the documentation to see how it works!

To the notebook!

Objectives

By the end of this session you should be able to...

- Describe and differentiate between content and collaborative based filtering
- Explain a utility matrix and how it is created
- Explain how a collaborative-based filtering approach generates recommendations or rating predictions