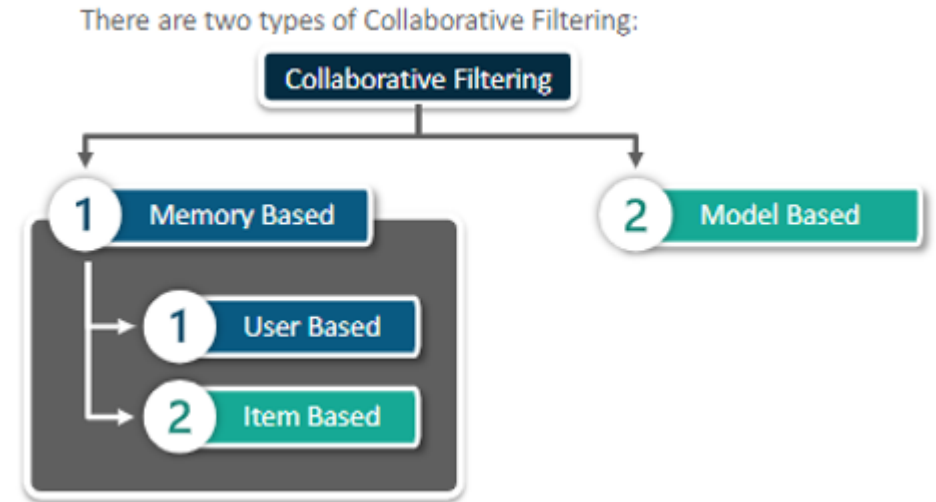
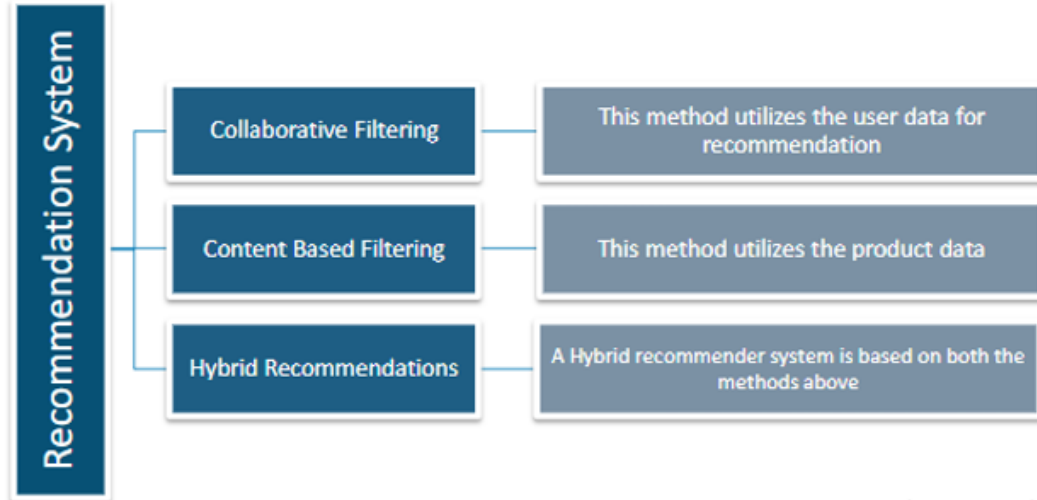


Recommendation System

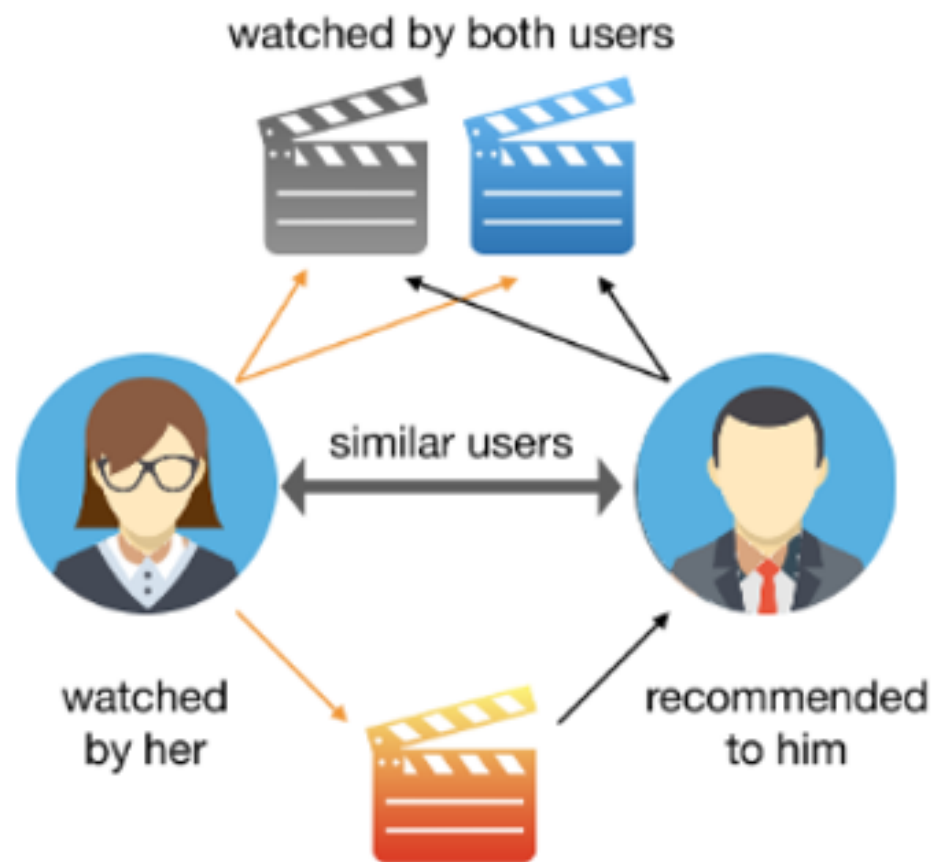
Recommendation Systems



User based Collaborative Filtering

User-Based Collaborative Filtering (UBCF)

- Algorithm searches a large group of users and finds users with similar tastes
- Algorithm looks at different things they like and combines them to create a ranked list of suggestions
- Algorithms used for measuring user or item similarity:
 - ❖ K – nearest neighbor (k-NN)
 - ❖ Pearson Correlation



UBCF: Example

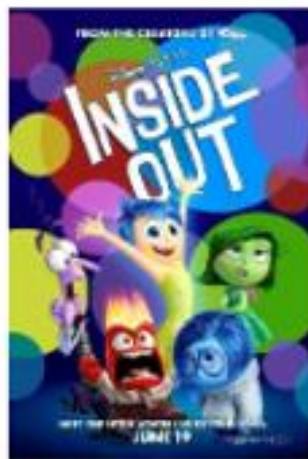


"Sarah just watched the movie *COCO*."

Let's see how the recommendation engine recommends her the movies?

UBCF Working: Step 1

- Generate a list of users who have seen the following movies
- List down the movies watched by Sarah (the user to which movie will be recommended)



John	Yes	Yes	Yes	Yes
Dave	No	Yes	No	No
Stuart	No	Yes	Yes	No
Sam	No	No	No	Yes
Sarah	?	Yes	?	?

UBCF Working: Step 2

- Finding users similar to Sarah:



John	Yes	Yes	Yes	Yes
Dave	No	Yes	No	No
Stuart	No	Yes	Yes	No
Sam	No	No	No	Yes
Sarah	?	Yes	?	?

UBCF Working: Step 3

- Using the data, it's found that the movie *Book of Life* gets more votes, so it gets recommended to Sarah



John	Yes	Yes	Yes	Yes
Dave	No	Yes	No	No
Stuart	No	Yes	Yes	No
Sarah	?	Yes	?	?

1 vote

2 votes



1 vote

Item based Collaborative Filtering

Item based Collaborative Filtering

- The table below contains the ratings the Users have give to Items (movies)
- The rating is explicit and is on a scale of 1 to 5
- In most cases majority of cells are empty as a user rates only for few items

User/Item	I1	I2	I3
User_1	2	–	3
User_2	5	2	–
User_3	3	3	1
User_4	–	2	2

- We need to find similarity between each of I1 & I2, I2 & I3, I3 & I1
- That can be done using Cosine similarity or Correlation
- Now if a new user has watched I1, we can recommend the most similar item from the rest of the items

Model Based Collaborative Filtering

Matrix Factorization

- Matrix factorization is to, obviously, factorize a matrix, i.e. to find out two matrices such that when you multiply them you will get back the original matrix
 - Given matrix R (size $u \times d$), find matrices P & Q such that $R \approx P * Q$
 - P will be of size $(u \times k)$ and Q will be size $(k \times d)$

Matrix Factorization - Intuition

- The intuition is that there should be ***some latent features*** that determine how a user rates a movie
 - For example, a user would give high ratings to a certain movie if he like the actors/actresses of the movie, or if the movie is an action movie, which is a genre preferred by the users
- If we can discover these latent features, we should be able to predict a rating with respect to a certain user and a certain movie

- Given user ratings matrix **R**

	4Ever	Hells Bells	Maria	2Cellos	Rimbaud
David	5	4	0	-	1
Rosie	0	1	-	2	-
Emma	5	5	-	-	-
Jade	0	-	5	2	4

- With matrix factorization, would like to discover latent features
- Each row of **P** would represent the strength of the associations between a user and the features
- Similarly, each row of Q^T would represent the strength of the associations between an item and the features
- Assumption - number of features would be smaller than the number of users and the number of items.

- By matrix factorization we find out matrices **P** & **Q** such that $R \approx P * Q$

P

	Rock	Classical	Jazz
David			
Rosie			
Emma			
Jade			

*

Q

	4Ever	Hells Bells	Maria	2Cellos	Rimbaud
Rock					
Classical					
Jazz					

Latent Features

=

new R

	4Ever	Hells Bells	Maria	2Cellos	Rimbaud
David	5	4	0	3	1
Rosie	0	1	3	2	3
Emma	5	5	3	5	4
Jade	0	1	5	2	4

- Given user ratings

		4Ever	Hells Bells	Maria	2Cellos	Rimbaud
David		5	4	0		1
Rosie		0	1		2	
Emma		5	5			
Jade		0		5	2	4

- By matrix factorization we find out

		Rock	Classical	Jazz
4Ever		100	0	0
Hells Bells		80	20	0
Maria		0	60	40
2Cellos		60	40	0
Rimbaud		20	60	20

 \times


		Rock	Classical	Jazz
David		✓	✗	✗
Rosie		✗	✓	✗
Emma		✓	✓	✗
Jade		✗	✓	✓

 $=$





		4Ever	Hells Bells	Maria	2Cellos	Rimbaud
David		5	4	0	3	1
Rosie		0	1	3	2	3
Emma		5	5	3	5	4
Jade		0	1	5	2	4

How to recommend?

Given

		4Ever	Hell's Bells	Maria	2Cellos	Rimbaud
David		5	4	0		1
Rosie		0	1		2	
Emma		5	5			
Jade		0		5	2	4

Find

		4Ever	Hell's Bells	Maria	2Cellos	Rimbaud
David		5	4	0	3	1
Rosie		0	1	3	2	3
Emma		5	5	3	5	4
Jade		0	1	5	2	4

Since we just filled the values using **Matrix Factorization** and found out that if **Emma** listens to "2Cellos – They don't care about us" she would rate it 5/5, so we recommend her the song.

Content Based Filtering

CBF: Example



“Sarah just watched the movie *Inside Out*.”

Lets see how the recommendation engine recommends her the movies?

CBF Working: Step 1

- Generate a list of features about the movies like:
 - Actors
 - Directors
 - Themes



CBF Working: Step2

- Compare columns of each movies with column of the movie *Inside Out*
- The column with the most match is of *Minions*, so the system will recommend it to watch



Animated	Yes	Yes	No	No
Marvel	No	No	Yes	Yes
Super Villain	No	Yes	Yes	Yes
IMDB rating 8+	Yes	No	Yes	No
Comedy	Yes	Yes	No	Yes

3

1

1

Content v/s Collaborative Filtering

- Content-based filtering, makes recommendations based on user preferences for product features
- Collaborative filtering mimics user-to-user recommendations
 - It predicts users preferences as a linear, weighted combination of other user preferences

