Recommender Systems Cont...

→ NETFLIX :

Comedy Shows → Comedy Movies (User will like this / User will give good rating)

R.S → 100 predictions

↓ Rank these predictions

Movie 1 ←

Movie 2

Movie 3

⋮

Movie 100

NETFLIX

↳ User to stay on platform (hook on platform)

Recommender System evolved over time:

① Pre-2007 era:

→ Similarity based

→ Content based

→ Collaborative Filtering

② 2007 - 2015 :

NETFLIX → Open Contest / Hackathon
(Improve R.S.)

Winning team → Solution [Matrix Factorization]

③ Post 2015 :

- Deep Learning Algorithms

- How do we formulate a R.S. problem ?

NETFLIX
Suppose, we have 'n' users : billions of users
we have 'm' items : millions of items

Vectors :

User-Vector	U_i	→	$i = [1, n]$
Item-Vector	I_j	→	$j = [1, m]$

Goal :

① To suggest a list of items to a user U_i, which user will like.

② These items should be ranked (preferred)

③ This is based on historical data.

User U_i → I_{10} , I_{12} , I_{17}

[Comedy Genre] → Hera Pheri, Golmaal, Dhamaal

- How can we represent a dataset in Recommender Systems?

- Dataset is represented in the form of a Matrix = 'A'

    - Each row represents a user U_i.
    - Each column represents an item I_j.

A_{ij} ⇒ interaction of user 'i' with 'item' j

↓ can be measured in many ways

① If the video is liked or not (binary values)

② Percentage of songs listened to, SPOTIFY

③ Time spent on product (Amazon, Flipkart)

④ Ratings on Netflix movies (Numerical value)

$A \rightarrow n \times m$ dimensions

$n \simeq 10^9$

$m \simeq 10^8$

Matrix $\rightarrow 10^9 \times 10^8$

$\rightarrow 10^{17}$ very very big matrix

↓ Sparse Matrix

- It is not possible for every user to watch every video/movie, hence, Matrix 'A' is mostly a 'Sparse Matrix'

$$\text{Sparsity} = \frac{\text{No. of non-empty cells}}{\text{Total no. of cells}}$$

① Collaborative Filtering :

(i) Item-Item based

(ii) User-User based

(i) Item-Item based.

	I_1	I_2	I_3	I_4	.....		I_m
U_i →	0	1	0	0	1	0	1

User-Vector

I_j →	0	U_1
	1	U_2
	0	U_3
	1	⋮
	1	
	1	
	0	U_n

Item-Vector

$U_i \rightarrow$ bought items I_{10} and I_{12} in the past

similar similar

I_k

$\rightarrow$ This is like 'nearest neighbours'

Find similarity between different items.

- Cosine Similarity

$$\text{sim}(I_i, I_j) = \frac{I_i \cdot I_j}{||I_i|| \cdot ||I_j||} = \frac{I_i^T \times I_j}{||I_i|| \cdot ||I_j||}$$

Example:

$I_1 = [4, 5, 0, 3]$

$I_2 = [5, 4, 2, 0]$

$\text{sim}(I_1, I_2)$

① $I_1 \times I_2 \Rightarrow 20 + 20 + 0 + 0 = 40$

$$\begin{bmatrix} 4 & 5 & 0 & 3 \end{bmatrix} \begin{bmatrix} 5 \\ 4 \\ 2 \\ 0 \end{bmatrix}$$

② $||I_1|| \rightarrow \sqrt{4^2 + 5^2 + 0^2 + 3^2} = \sqrt{50} = 7.07$

$||I_2|| \rightarrow \sqrt{5^2 + 4^2 + 2^2 + 0^2} = \sqrt{45} = 6.7$

$$\text{sim}(I_1, I_2) = \frac{40}{7.07 \times 6.7} = \frac{40}{47.43} \simeq 0.8$$

0.8 → very close to 1 → highly similar

- How to store similarities:

  - Similarity Matrix → S_i

'm' items

Dimension → $m \times m$

	I_1	I_2	I_3	......	I_m
I_1	1	0.84	0.7	......	0.9
I_2	0.84	1			
I_3			1		
⋮				1	
I_m					1

Item-Item based C.F → Amazon in 1998

(b.) User-User based C.F

Cosine Similarity

$$sim(U_i, U_j) = \frac{U_i^T \times U_j}{\|U_i\| . \|U_j\|}$$

Using C.S → we find similar users

Consider, U_1 has already bought 'I_{10} and I_{18}'

→ 3 similar users to U_1

U_{10} : I_{10}, I_{12}, I_{18}, I_{20}

U_{26} : I_{10}, I_{18}, I_{26}, I_{12}

U_{58} : I_{18}, I_{12}

Using frequency based approach

↳ Recommend I_{12}, making it a good recom.

User-User based C.F.

New User?

→ past browsing history

→ no idea about similar users

↳ not able to provide recommendations

U_i → New User

I_i → New Item

} All values are empty in Matrix 'A'

↓ No similar users

No similar items

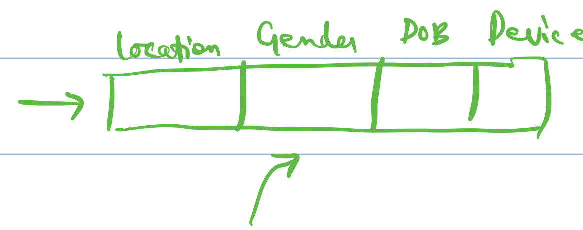
"COLD START PROBLEM"

New User → Trending / Popular / Frequently Bought

→ Even though the user has not interacted with any item, still we have some additional information like –

1. Location
2. Gender
3. Age (DOB)
4. Device

d-dimensional vector



Using above information, a new vector is created.

New Item :

1. Product Description
2. Price
3. Product Category

d′-dimensional vector

Additional information of new user / item → METADATA

To get recommendations

→ Content based Recommendation System

H.W. → Adv. and Dis. of Content Based R.S.

→ Adv. and Disa. of Collaborative Filtering

- Recommendation as a Regression Classification:

Suppose a user, U_i gave a rating of 4 to Item I_j

$$\boxed{A_{ij} = 4}$$

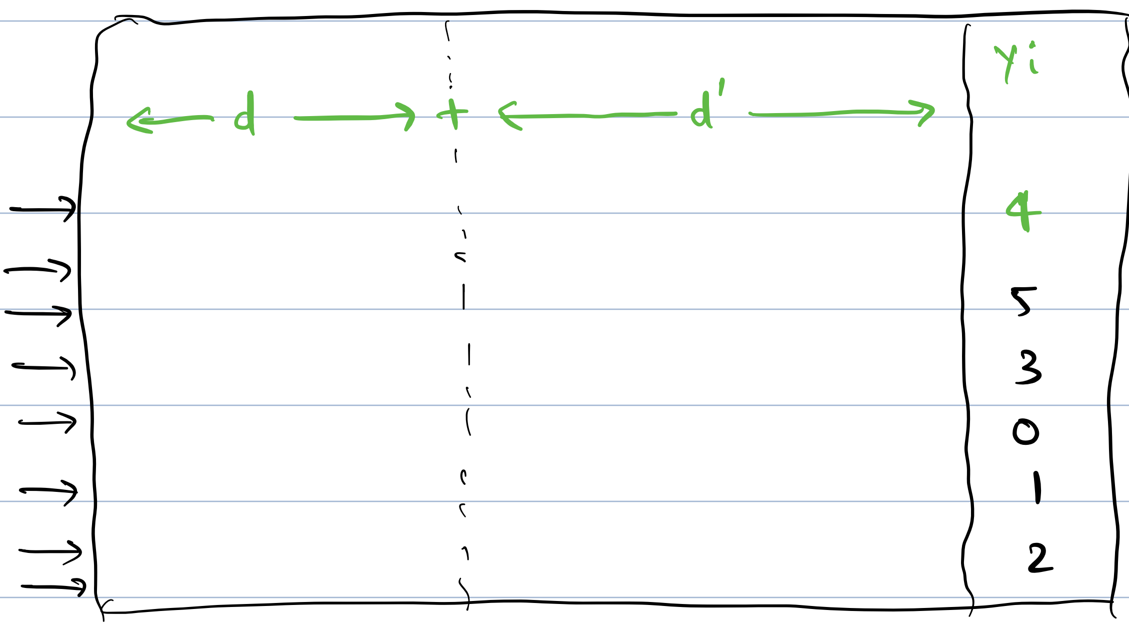
Using metadata,

$U_i \to d$ dimension vector

$I_j \to d'$ dimension vector

Locat' DOB $\underline{d=2}$

Loc. DOB Gender $d=3$



Make a train data from Matrix - A and metadata

→ doing this will be very computationally heav

Rec. Systems → Content-based → uses Metadata

Rec. Systems → Collaborative Filtering → A_{ij} (Similarity)

Rec. Systems → Hybrid → uses Content + A_{ij}

→ "MATRIX FACTORISATION"

4 items

Milk, Bread, Butter, Eggs

$$
\begin{array}{c|cccc}
 & \text{Milk} & \text{Bread} & \text{Butter} & \text{Eggs} \\
\hline
U_1 & 0 & 1 & 0 & 1 \\
U_2 & 1 & 0 & 1 & 1
\end{array}
$$

Milk - Br - $\boxed{0.8}$

Mil - B $\boxed{0.7}$

M - E →

B

Item - Item Based Sim.

$$\text{Cos. sim} = \frac{\text{Milk} \cdot \text{Butter}}{|\text{Milk}|\ |\text{Butter}|}$$

	I_1	I_2	I_3	I_4
I_1	1	0.7	0.6	0.3
I_2	0.8	1	0.1	0.2
I_3			1	
I_4				1

$[I_1, I_2]$ 0.6 0.7 $\rightarrow$

$\hookrightarrow$ I_3, $\underline{I_4}$ 0.3 0.2 $\rightarrow$