

## Item-Based Collaborative filtering

- ↳ Items tends to more permanent nature than people.
- but peoples opinion will change time to time.
- ↳ So, focusing on similarities b/w unchanged object can produce better results than looking similarities b/w people.

→ ~~Items~~ To find similarity, usually we have less Items than people

{ Note:- but for user-based, we deal with people & people are very high than Items.

→ Computational time is also less.

→ Item Similarities also makes better experience for new user.

Ex:-

		Aneesh	Bobby	Lucky
(1,0,0)	BB 1	4		
(1,0,1)	BB 2	5		5
(0,0,1)	RRR	.		5
(0,0,1)	Eaga			5
(0,1,0)	Raja		1	

Cosine Similarity

↓ Item Similarity matrix

	BB1	BB2	RRR	Eaga	Raja
BB1	1	1	0	0	0
BB2	1	1	1	1	0
RRR	1	1	1	1	0
Eaga	0	1	1	1	0
Raja	0	0	0	0	1

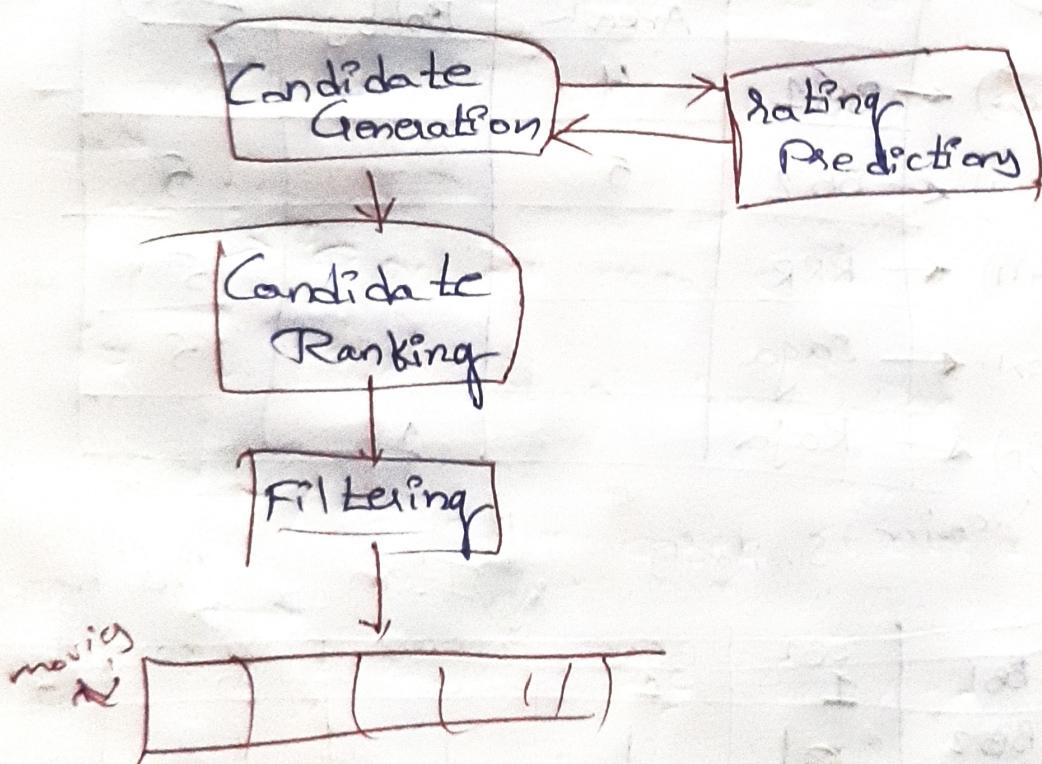
# Evaluating Collaborative filtering!

↳ as, we can't measure accuracy for  
Item based / user-based Collaborative filter  
because they don't make rating prediction

↳ we will measure HIT Rate

## KNN Recommenders! →

### Item-based Recommendation Architecture



we generate recommendation Candidates  
by predicting the ratings of everything

a user hasn't already rated.

### Candidate Ranking

↳ Selecting top K highest predicted

~~Filtering~~  
ratings

↳ we will eliminate the movies ~~which~~

the user has already seen the movies

### Item/User based KNN approach

Item

find the k most  
similar items rated  
by this user

Compute mean  
sim score  
weighted by  
ratings

Rating prediction

User

find the k most  
similar users who  
rated this item

Compute mean sim  
Score weighted  
by ratings

Rating prediction

These  
are  
k-nearest  
neighbors

Weighted  
average

Item & user approaches are same ; just we

replace item with user & user with item