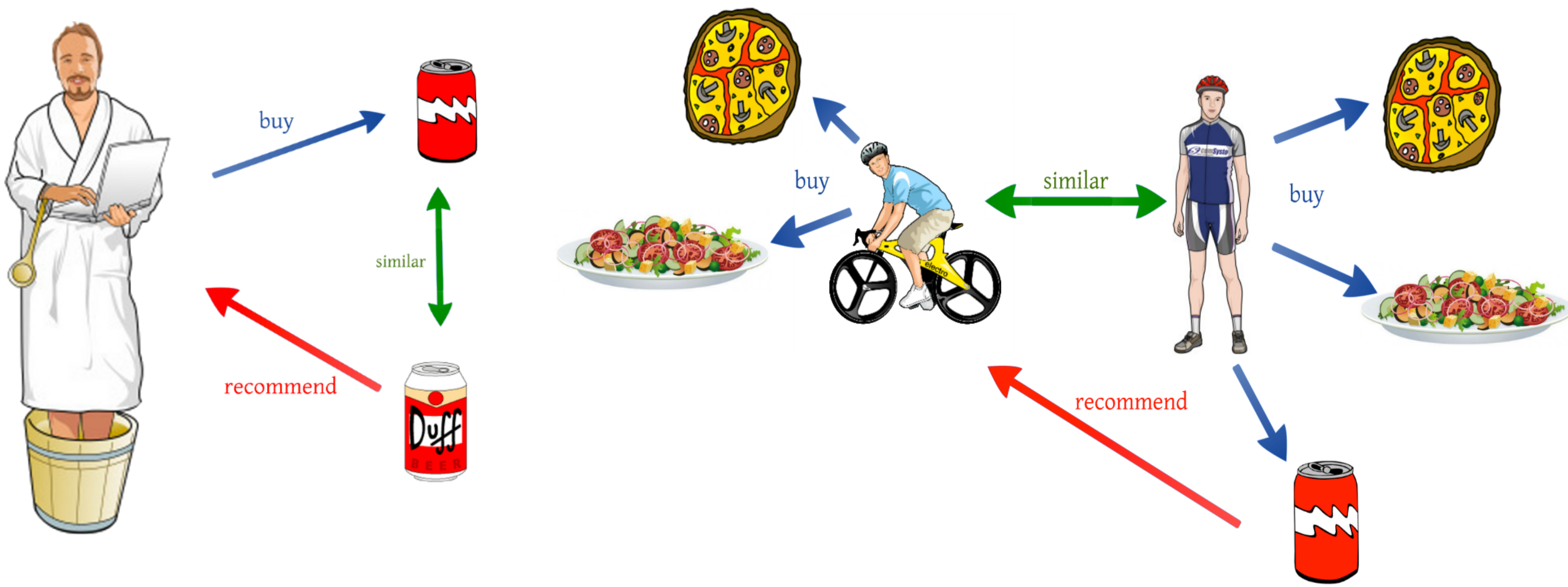
Recommender Systems

DSLA COURSE

ROHIT PADEBETTU

Recommender Systems

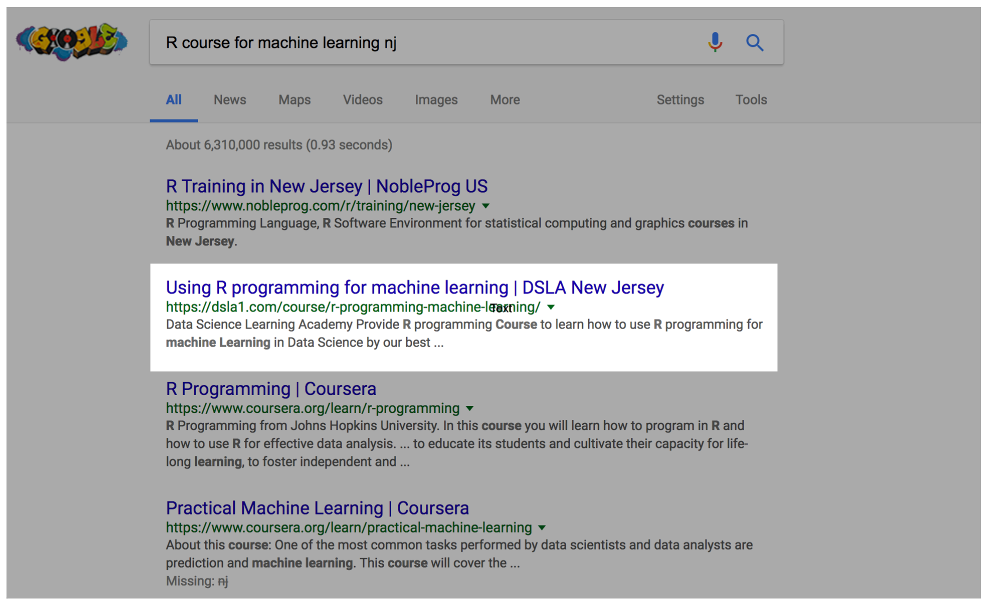


Applications

Traditionally people around us were recommendation engines

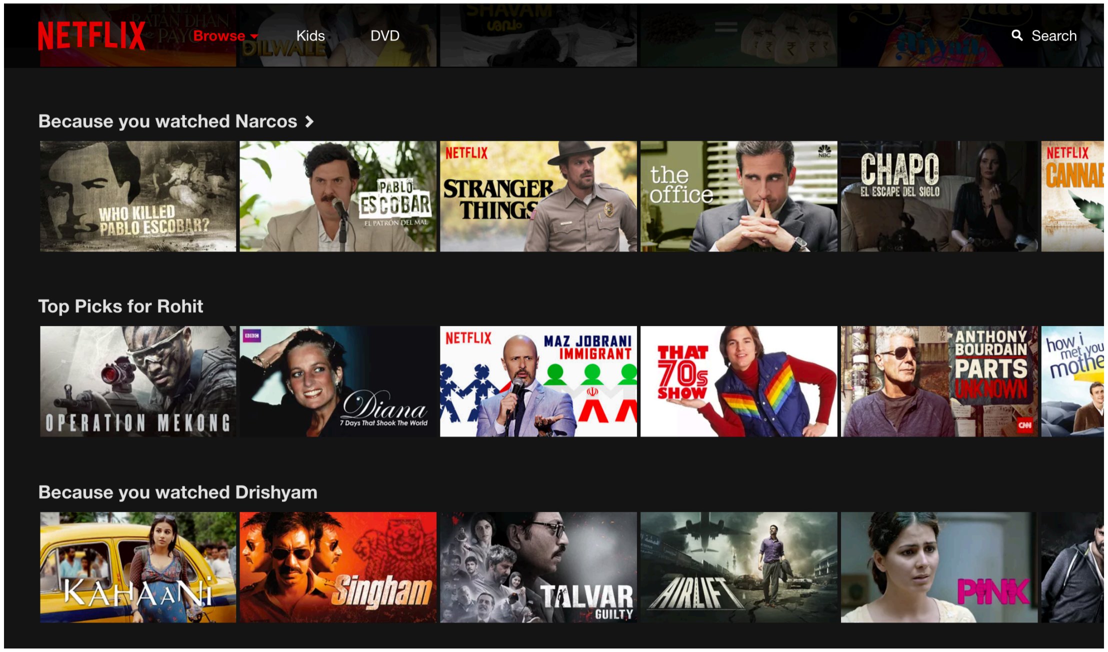
Applications

Google replaced everyone and recommends everything you ask for



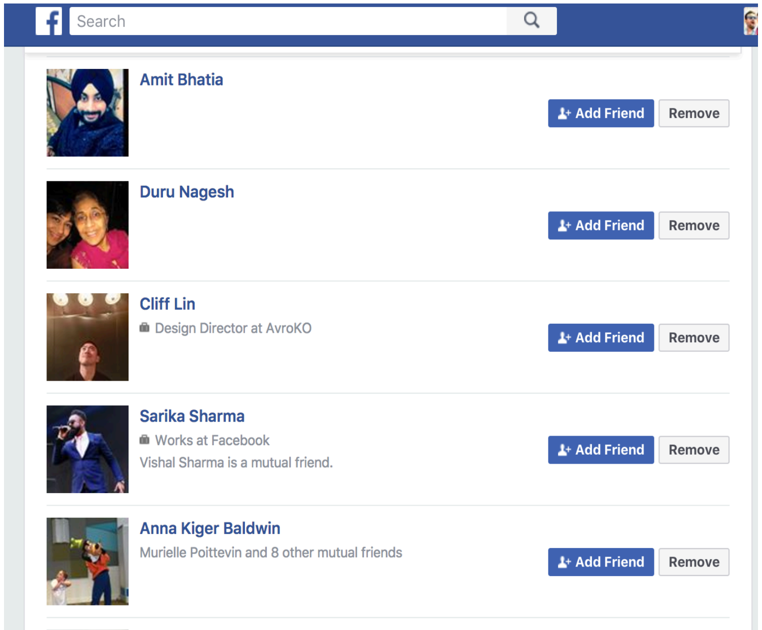
Applications

Netflix movies you may enjoy: Replaces friends you might have otherwise asked



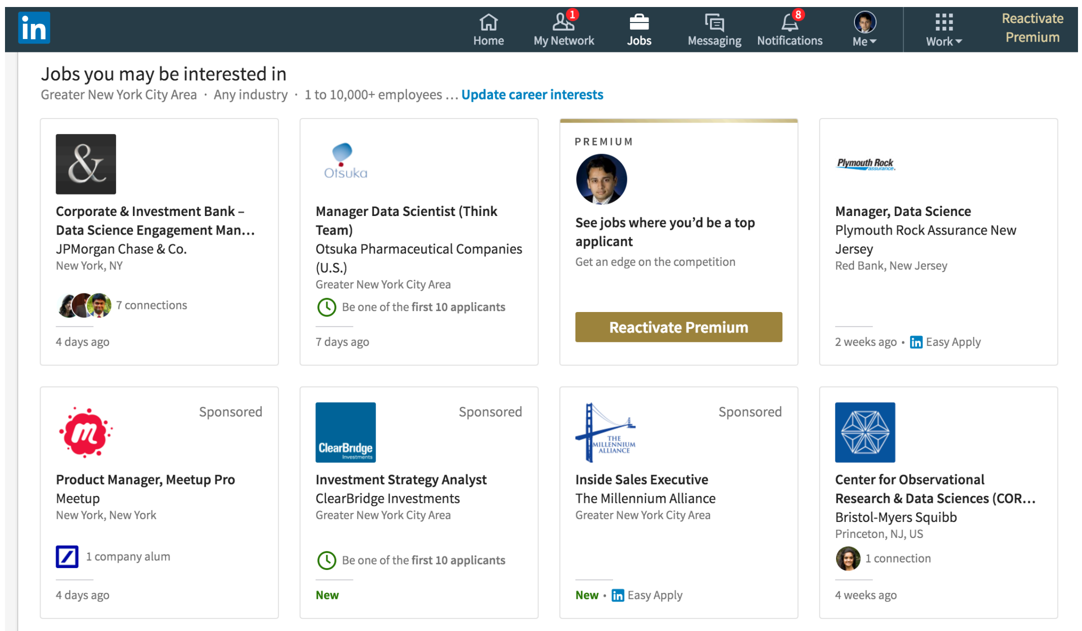
Applications

Facebook: Recommends new Friends you may make



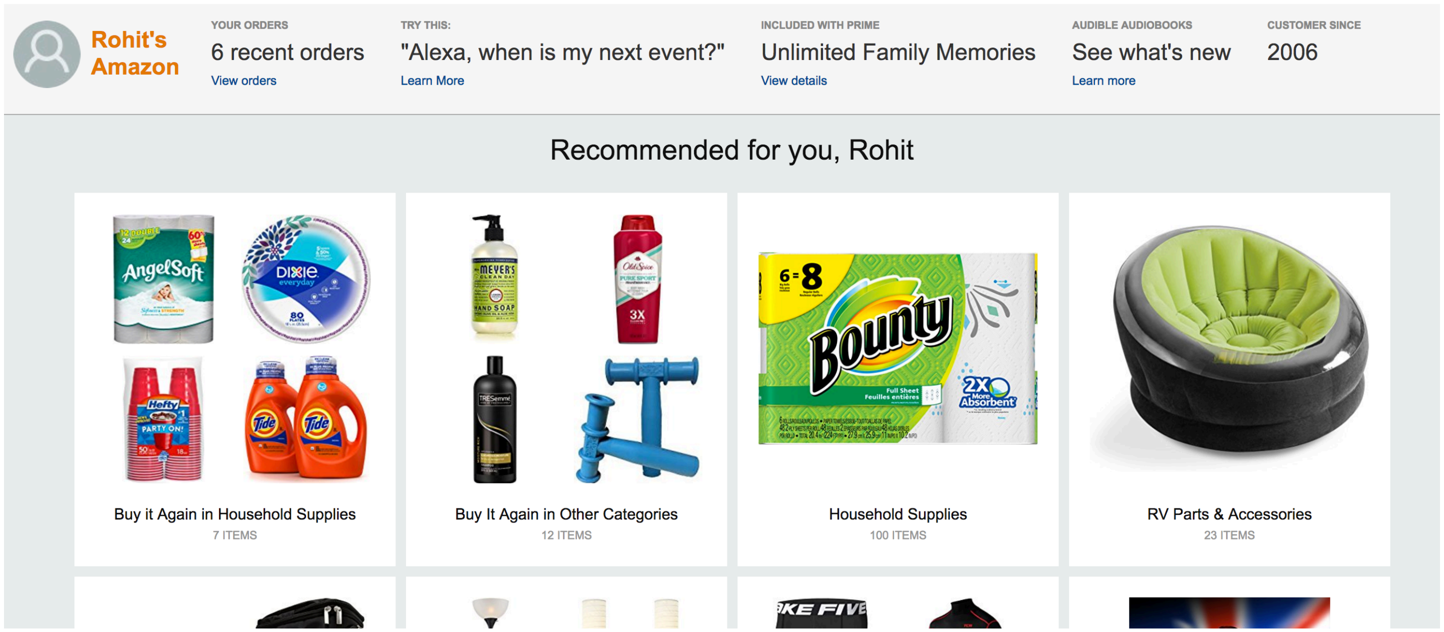
Applications

LinkedIn recommends jobs you maybe interested in



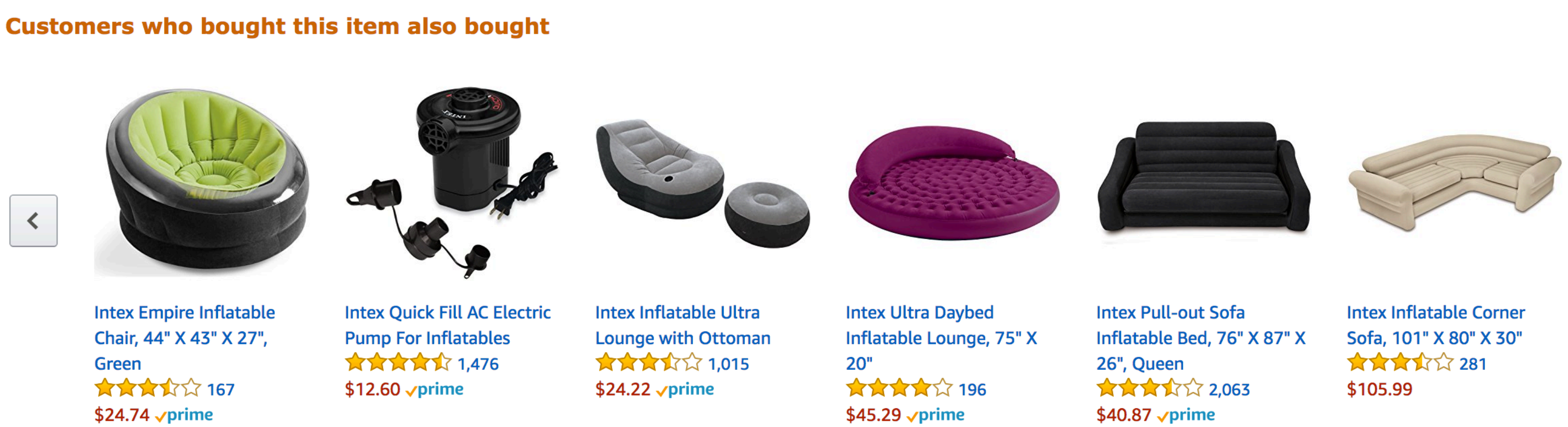
Applications

Amazon recommends stuff you can buy based on your buying habits

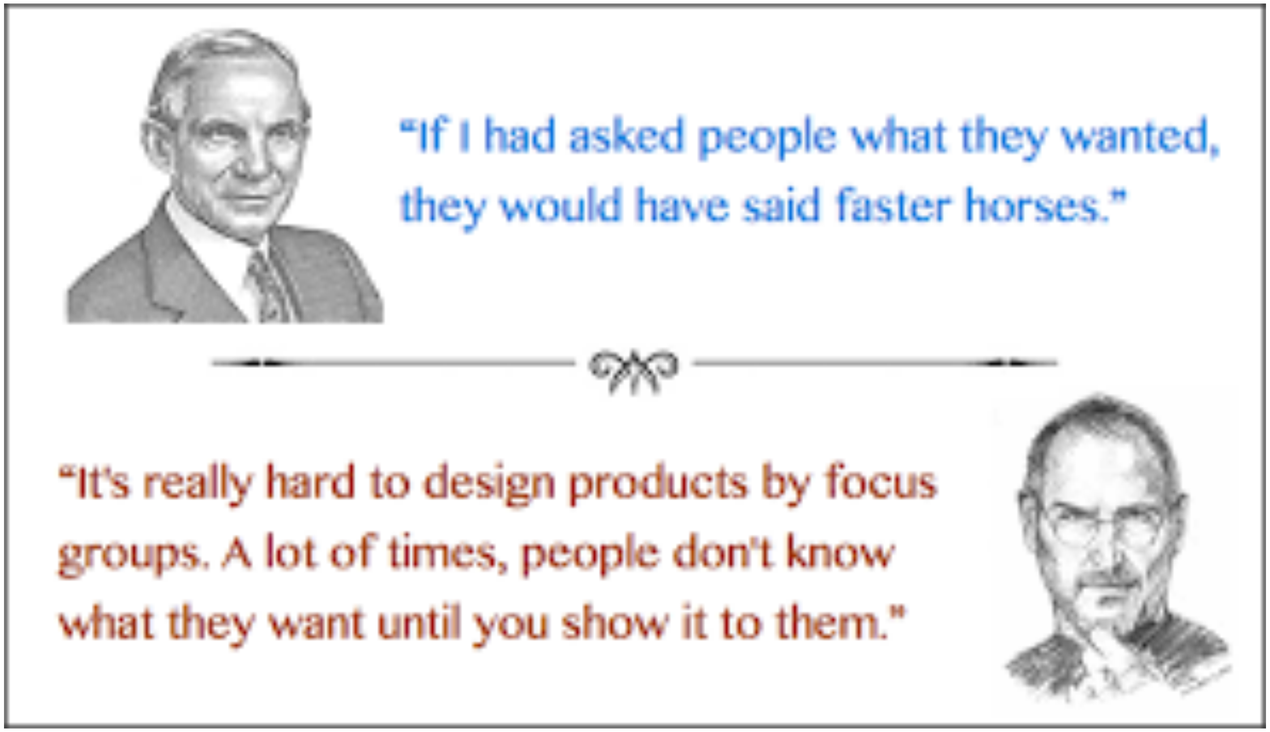


Applications

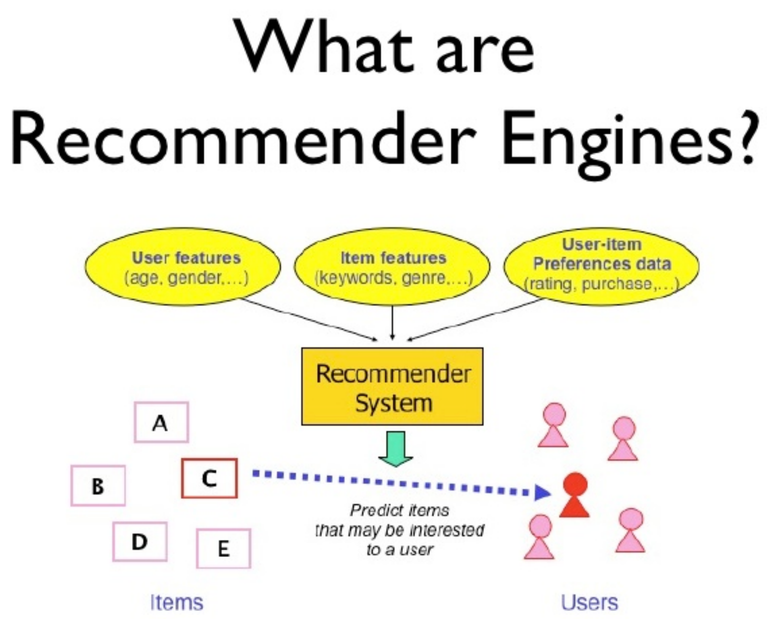
Amazon also recommends stuff you can buy based on others habits



Why Recommendation Engines ?

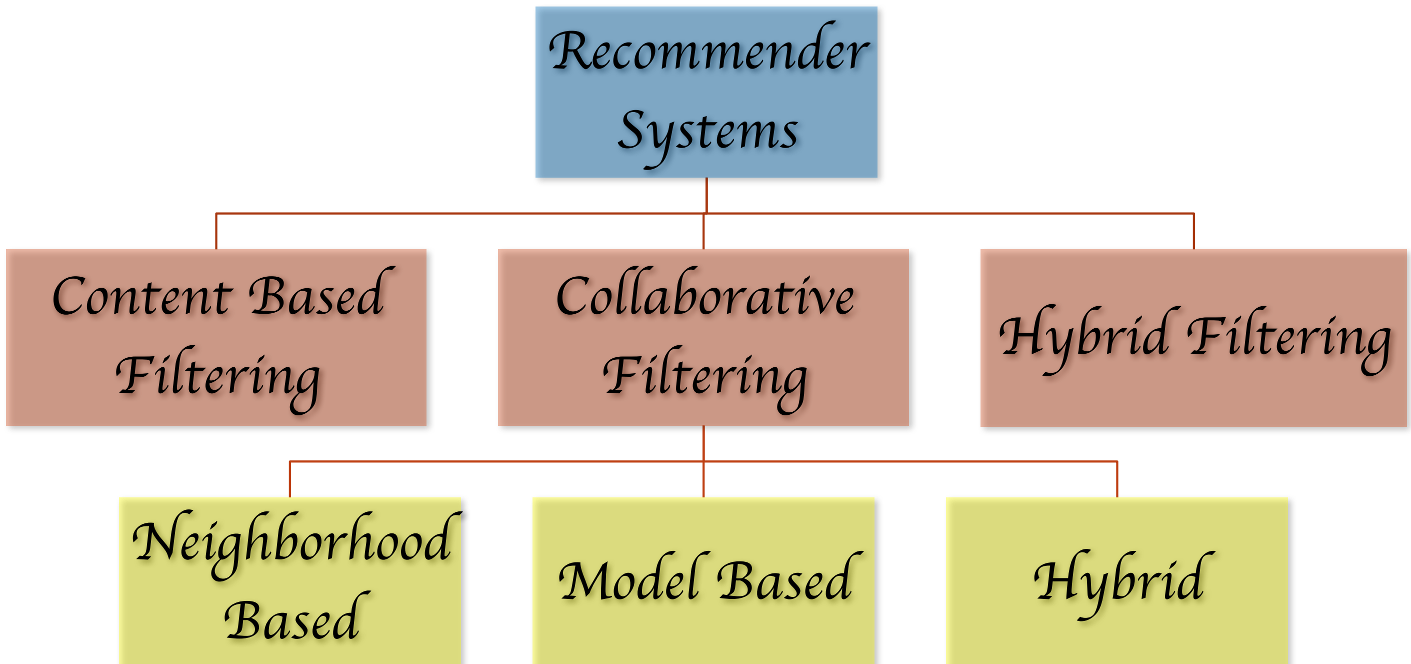


Recommender Systems

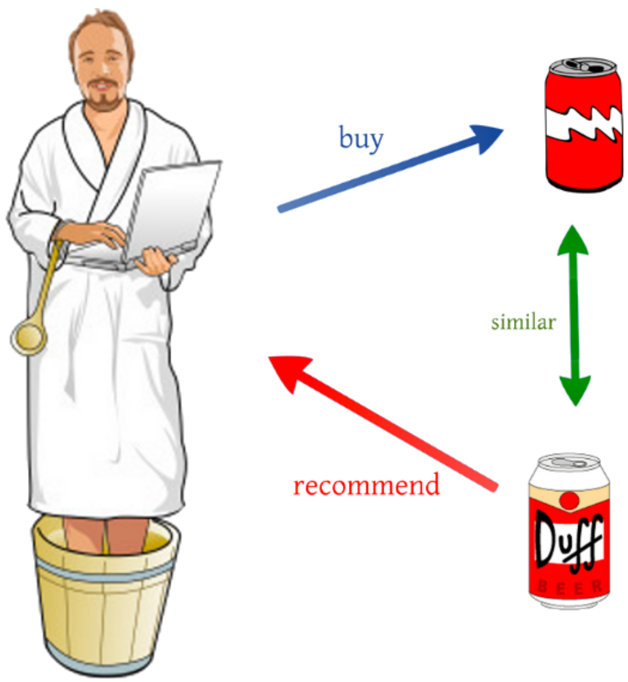


*A **recommender system** or*
*a **recommendation system** is a subclass of*
information filtering system that seeks to
***predict the "rating" or "preference"** that a*
user would give to an item
- Wikipedia

Types of Recommender Systems



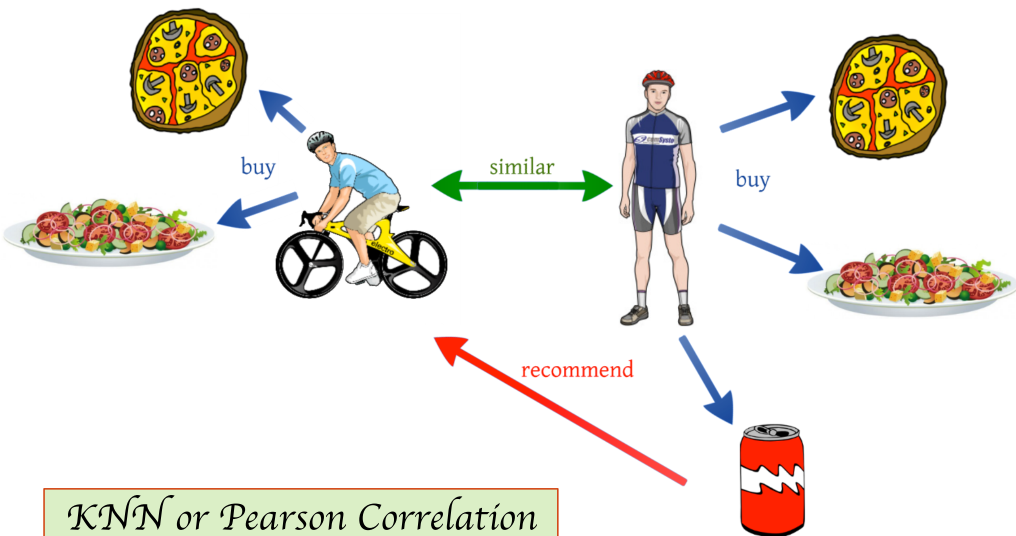
Content Based Filtering



Are based on item features and user preferences

The system tries to predict based on user's past behavior

Collaborative Filtering

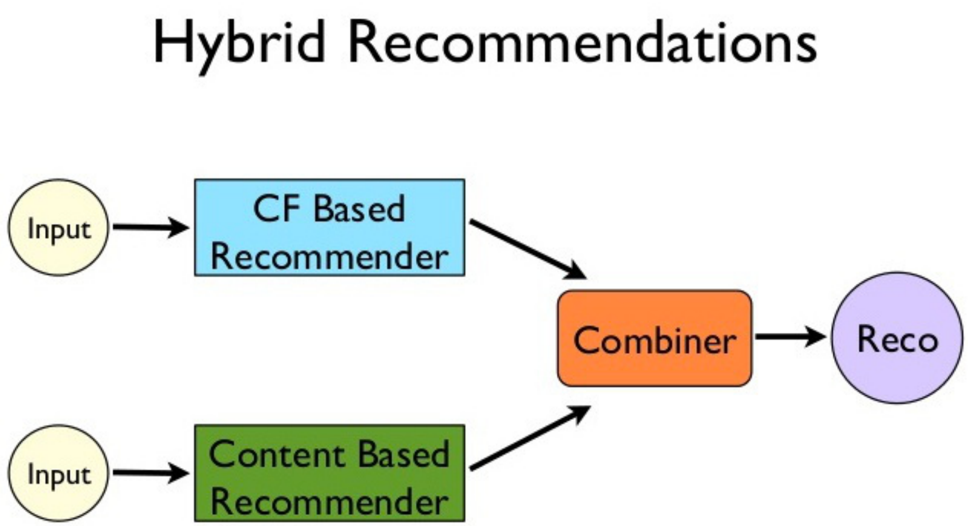


KNN or Pearson Correlation
is used for similarity

Are based on large amount of data collected on many users preferences

The system tries to predict based on
user similarity
and/or
item similarity

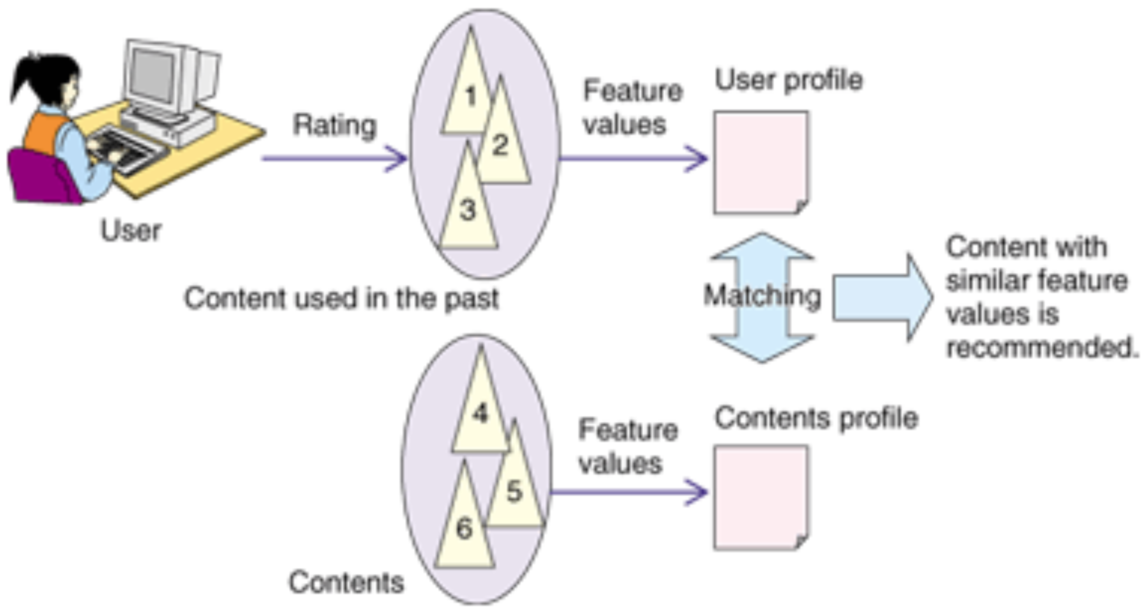
Hybrid Filtering



Are based on combination of recommendations from both content based and collaborative filtering systems

Performs better
Overcomes problems such as cold start or data sparsity

Content Based Filtering



A Feature Profile
&
A User Profile

Are required to make predictions

CBF - Example

Step1 : Track the Users Engagement with the articles

Articles	Big Data	R	Python	Machine Learning	Learning Paths	Total attributes	User 1	User 2
Article 1	1	0	1	0	1	3	1	-1
Article 2	0	1	1	1	0	3	-1	1
Article 3	0	0	0	1	1	2		
Article 4	0	0	1	1	0	2		1
Article 5	0	1	0	0	0	1		
Article 6	1	0	0	1	0	2	1	
DF	2	2	3	4	2			

CBF - Example

Step2 : Normalize features

1/SQRT(3)

Articles	Big Data	R	Python	Machine Learning	Learning Paths	Total attributes
Article 1	0.577350269	0	0.577350269	0	0.577350269	3
Article 2	0	0.577350269	0.577350269	0.577350269	0	3
Article 3	0	0	0	0.707106781	0.707106781	2
Article 4	0	0	0.707106781	0.707106781	0	2
Article 5	0	1	0	0	0	1
Article 6	0.707106781	0	0	0.707106781	0	2

CBF - Example

Step3 : Create User Profiles for each feature

Articles	Big Data	R	Python	Machine Learning	Learning Paths	Total attributes	User 1	User 2
Article 1	0.577350269	0	0.577350269	0	0.577350269	3	1	-1
Article 2	0	0.577350269	0.577350269	0.577350269	0	3	-1	1
Article 3	0	0	0	0.707106781	0.707106781	2		
Article 4	0	0	0.707106781	0.707106781	0	2		1
Article 5	0	1	0	0	0	1		
Article 6	0.707106781	0	0	0.707106781	0	2	1	

User Profiles					
User1	1.28445705	-0.577350269	0	0.129756512	0.577350269
User2	-0.577350269	0.577350269	0.707106781	1.28445705	=SUMPRODUCT(F16:F21,$K$16:$K21)

SUMPRODUCT(array1, [array2], [array3], [array4], ...)

CBF - Example

Step4 : Calculate Inverse Document Frequency (IDF)

Articles	big data	R	python	machine learning	learning paths
Article 1	0.577350269	0	0.577350269	0	0.577350269
Article 2	0	0.577350269	0.577350269	0.577350269	0
Article 3	0	0	0	0.707106781	0.707106781
Article 4	0	0	0.707106781	0.707106781	0
Article 5	0	1	0	0	0
Article 6	0.707106781	0	0	0.707106781	0
User Profiles					
User1	1.28445705	-0.577350269	0	0.129756512	0.577350269
User2	-0.577350269	0.577350269	0.707106781	1.28445705	-0.577350269
DF	2	2	3	4	2
IDF	0.698970004	0.698970004	0.52287875	0.397940009	0.698970004

Log(10/4) = 0.3979

Here we assume there are 10 such articles

CBF - Example

Step5 : Use user profiles & item profiles to create predictions

Articles	big data	R	python	machine learning	learning paths	User 1	User 2	Pred User1	PredUser2
Article 1	0.577350269	0	0.577350269	0	0.577350269	1	-1	0.751333312	
Article 2	0	0.577350269	0.577350269	0.577350269	0	-1	1	-0.20317834	
Article 3	0	0	0	0.707106781	0.707106781			0.321864985	
Article 4	0	0	0.707106781	0.707106781	0		1	0.036511676	
Article 5	0	1	0	0	0			-0.40355052	
Article 6	0.707106781	0	0	0.707106781	0	1	=SUMPRODUCT(B36:F36,$B$39:$F$39,$B$43:$F$43)		

SUMPRODUCT(**array1**, [array2], [array3], [array4], [array5], ...)

User Profiles					
User1	1.28445705	-0.577350269	0	0.129756512	0.577350269
User2	-0.577350269	0.577350269	0.707106781	1.28445705	-0.577350269

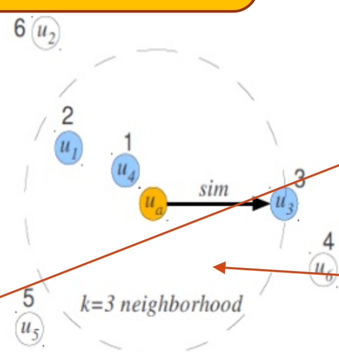
DF	2	2	3	4	2
IDF	0.698970004	0.698970004	0.52287875	0.397940009	0.698970004

Multiply Item profile with IDF to get weighted article score vector.
Multiply that with user profile to get score for user per article

User Based Collaborative Filtering

Neighborhood Based Approach

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	?	4.0	4.0	2.0	1.0	2.0	?	?
u_2	3.0	?	?	?	5.0	1.0	?	?
u_3	3.0	?	?	3.0	2.0	2.0	?	3.0
u_4	4.0	?	?	2.0	1.0	1.0	2.0	4.0
u_5	1.0	1.0	?	?	?	?	?	1.0
u_6	?	1.0	?	?	1.0	1.0	?	1.0
u_a	?	?	4.0	3.0	?	1.0	?	5.0
r_a	3.5	4.0			1.3		2.0	



$$\hat{r}_{aj} = \frac{1}{\sum_{i \in \mathcal{N}(a)} s_{ai}} \sum_{i \in \mathcal{N}(a)} s_{ai} r_{ij}$$

$$\mathrm{sim}_{\mathrm{Cosine}}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

Reference: Recommenderlab vignette, http://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Assume 6 users and 8 item matrix with cells representing ratings

Active User U_a has rated item 3,4,6 & 8. We need to predict ratings for item 1,2,5 & 7

*We calculate **user similarity** to get 3 most similar users U_1, U_3, U_4*
Cosine Similarity or Pearson Correlation of given ratings can be used

Missing Ratings for active user U_a are Average of existing ratings of similar users

Item Based Collaborative Filtering

Model Based Approach

S	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	$\hat{r}_a$
i_1	-	0.1	0	**0.3**	**0.2**	**0.4**	0	0.1	-
i_2	0.1	-	**0.8**	**0.9**	0	**0.2**	0.1	0	0.0
i_3	0	**0.8**	-	0	**0.4**	0.1	0.3	**0.5**	4.6
i_4	**0.3**	**0.9**	0	-	0	0.1	0	**0.2**	3.2
i_5	**0.2**	0	**0.4**	0	-	0.1	**0.2**	0.1	-
i_6	**0.4**	**0.2**	0.1	**0.3**	0.1	-	0	0.1	2.0
i_7	0	**0.1**	**0.3**	0	**0.2**	0	-	0	4.0
i_8	**0.1**	0	**0.5**	**0.2**	0.1	0.1	0	-	-
u_a	2	?	?	?	4	?	?	5	

k=3

Assume 6 users and 8 item matrix with cells representing ratings

Active User U_a has rated item 1,5 & 8. We need to predict ratings for item 2,3,4,6 & 7

*We calculate **item similarity** from the 3 items rated by active user U_a against all other items*
Cosine Similarity or Pearson Correlation can be used again

Missing Ratings are wtd average of similar item ratings filtered by a threshold
*(2*0 + 4*0.4+5*0.5)/(0+0.4+0.5) = 4.6*

Recommendation Systems

Demo

Recommendation Systems

Have good rest of weekend!