

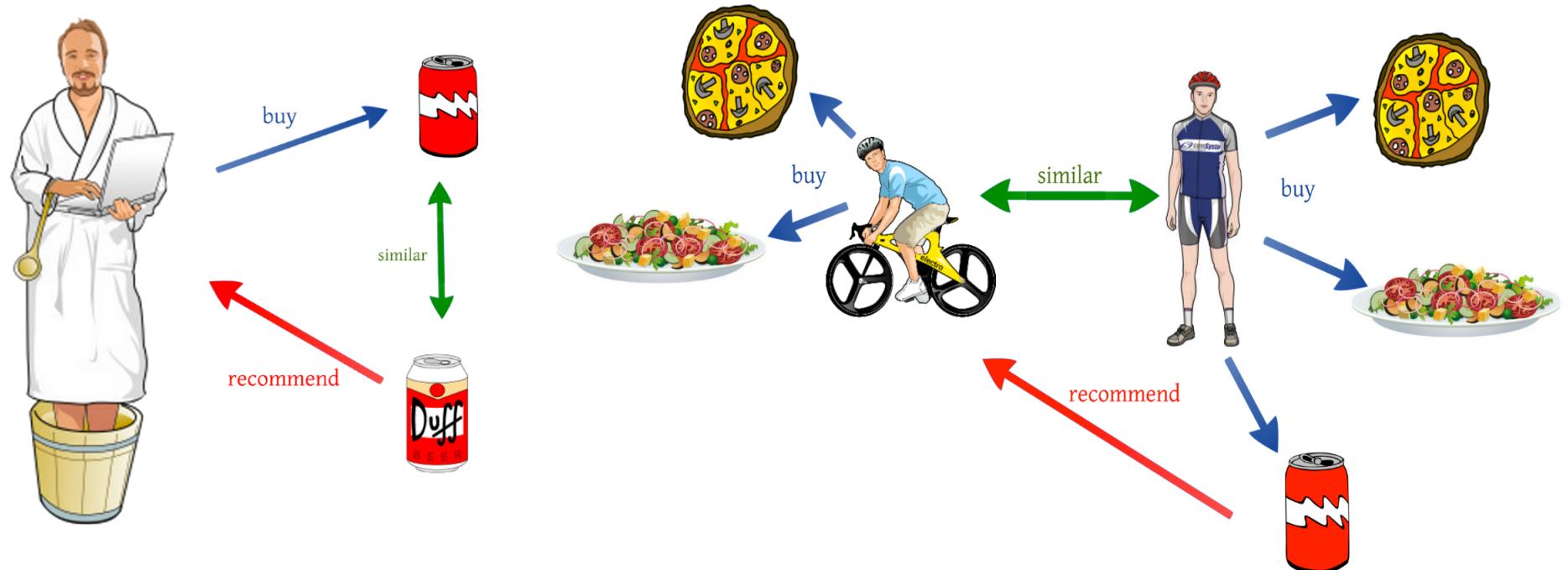


Recommender Systems

DSLA COURSE

ROHIT PADEBETTU

Recommender Systems



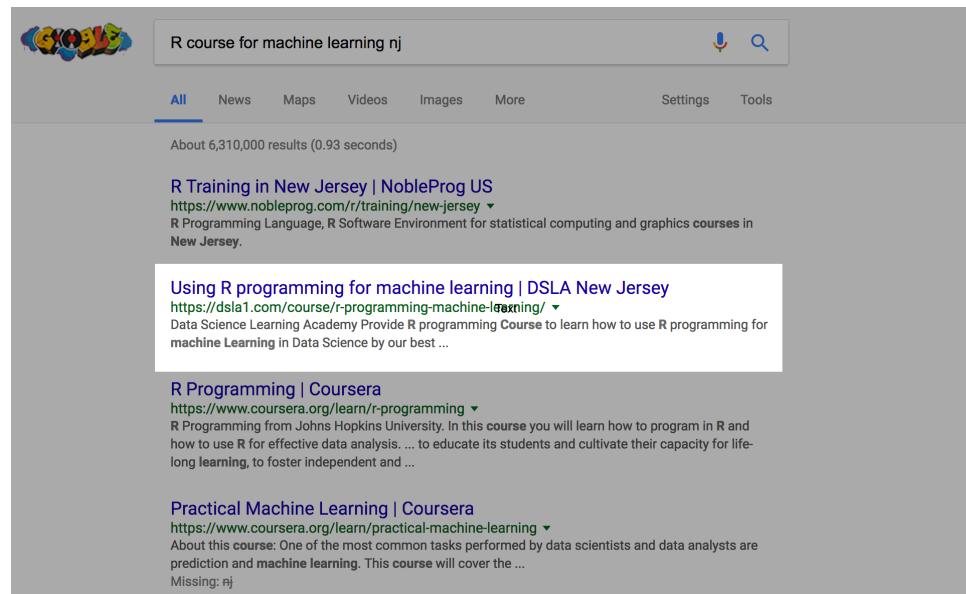
Applications

Traditionally people around us were recommendation engines



Applications

Google replaced everyone and recommends everything you ask for



R course for machine learning nj

All News Maps Videos Images More Settings Tools

About 6,310,000 results (0.93 seconds)

R Training in New Jersey | NobleProg US
<https://www.nobleprog.com/r/training/new-jersey> ▾
R Programming Language, R Software Environment for statistical computing and graphics courses in New Jersey.

Using R programming for machine learning | DSLA New Jersey
<https://dsla1.com/course/r-programming-machine-learning/> ▾
Data Science Learning Academy Provide R programming Course to learn how to use R programming for machine Learning in Data Science by our best ...

R Programming | Coursera
<https://www.coursera.org/learn/r-programming> ▾
R Programming from Johns Hopkins University. In this course you will learn how to program in R and how to use R for effective data analysis... to educate its students and cultivate their capacity for life-long learning, to foster independent and ...

Practical Machine Learning | Coursera
<https://www.coursera.org/learn/practical-machine-learning> ▾
About this course: One of the most common tasks performed by data scientists and data analysts are prediction and machine learning. This course will cover the ...
Missing: nj

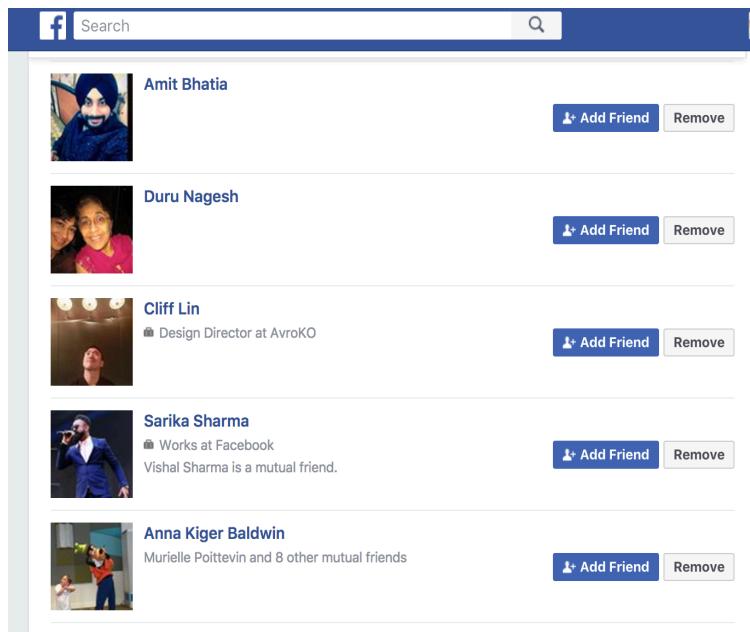
Applications

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



Applications

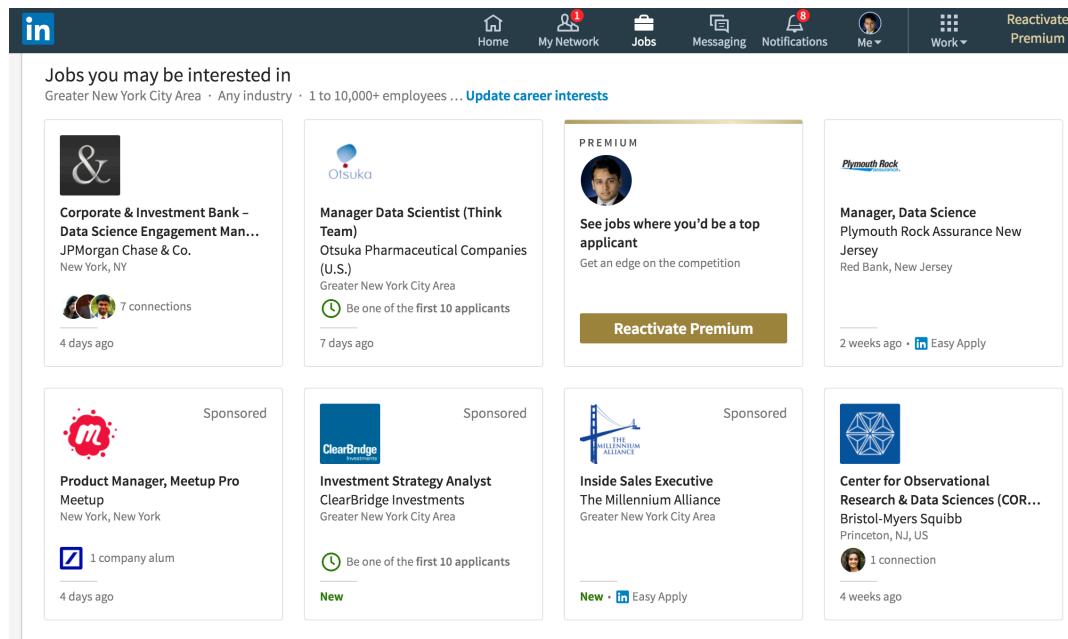
Facebook: Recommends new Friends you may make





Applications

LinkedIn recommends jobs you maybe interested in



The screenshot shows the LinkedIn interface displaying job recommendations. At the top, there's a navigation bar with icons for Home, My Network (1), Jobs, Messaging (8), Notifications (Me), Work, and Reactivate Premium. The main heading is 'Jobs you may be interested in' with a subtitle 'Greater New York City Area · Any industry · 1 to 10,000+ employees ... Update career interests'. Below this, there are eight job listing cards arranged in two columns of four:

- Corporate & Investment Bank – Data Science Engagement Manager** (JPMorgan Chase & Co., New York, NY)
7 connections · 4 days ago
- Manager Data Scientist (Think Team)** (Otsuka Pharmaceutical Companies (U.S.), Greater New York City Area)
Be one of the first 10 applicants · 7 days ago
- Manager, Data Science** (Plymouth Rock Assurance New Jersey, Red Bank, New Jersey)
Reactivate Premium · 2 weeks ago · Easy Apply
- Product Manager, Meetup Pro** (Meetup, New York, New York)
1 company alum · 4 days ago
- Sponsored**
Investment Strategy Analyst (ClearBridge Investments, Greater New York City Area)
Be one of the first 10 applicants · New
- Sponsored**
Inside Sales Executive (The Millennium Alliance, Greater New York City Area)
New · Easy Apply
- Sponsored**
Center for Observational Research & Data Sciences (COR...) (Bristol-Myers Squibb, Princeton, NJ, US)
1 connection · 4 weeks ago

Applications

Amazon recommends stuff you can buy based on your buying habits

YOUR ORDERS
6 recent orders
[View orders](#)

TRY THIS:
"Alexa, when is my next event?"
[Learn More](#)

INCLUDED WITH PRIME
Unlimited Family Memories
[View details](#)

AUDIBLE AUDIOBOOKS
See what's new
[Learn more](#)

CUSTOMER SINCE
2006

Recommended for you, Rohit

 Buy it Again in Household Supplies <small>7 ITEMS</small>	 Buy It Again in Other Categories <small>12 ITEMS</small>	 Household Supplies <small>100 ITEMS</small>	 RV Parts & Accessories <small>23 ITEMS</small>
			

Applications

Amazon also recommends stuff you can buy based on others habits

Customers who bought this item also bought



Intex Empire Inflatable Chair, 44" X 43" X 27",
Green
★★★★★ 167
\$24.74 



Intex Quick Fill AC Electric Pump For Inflatables
★★★★★ 1,476
\$12.60 



Intex Inflatable Ultra Lounge with Ottoman
★★★★★ 1,015
\$24.22 



Intex Ultra Daybed Inflatable Lounge, 75" X 20"
★★★★★ 196
\$45.29 



Intex Pull-out Sofa Inflatable Bed, 76" X 87" X 26", Queen
★★★★★ 2,063
\$40.87 



Intex Inflatable Corner Sofa, 101" X 80" X 30"
★★★★★ 281
\$105.99

Why Recommendation Engines ?



**"If I had asked people what they wanted,
they would have said faster horses."**

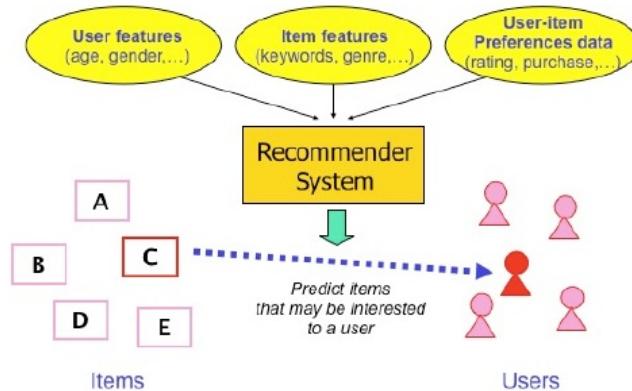


**"It's really hard to design products by focus
groups. A lot of times, people don't know
what they want until you show it to them."**



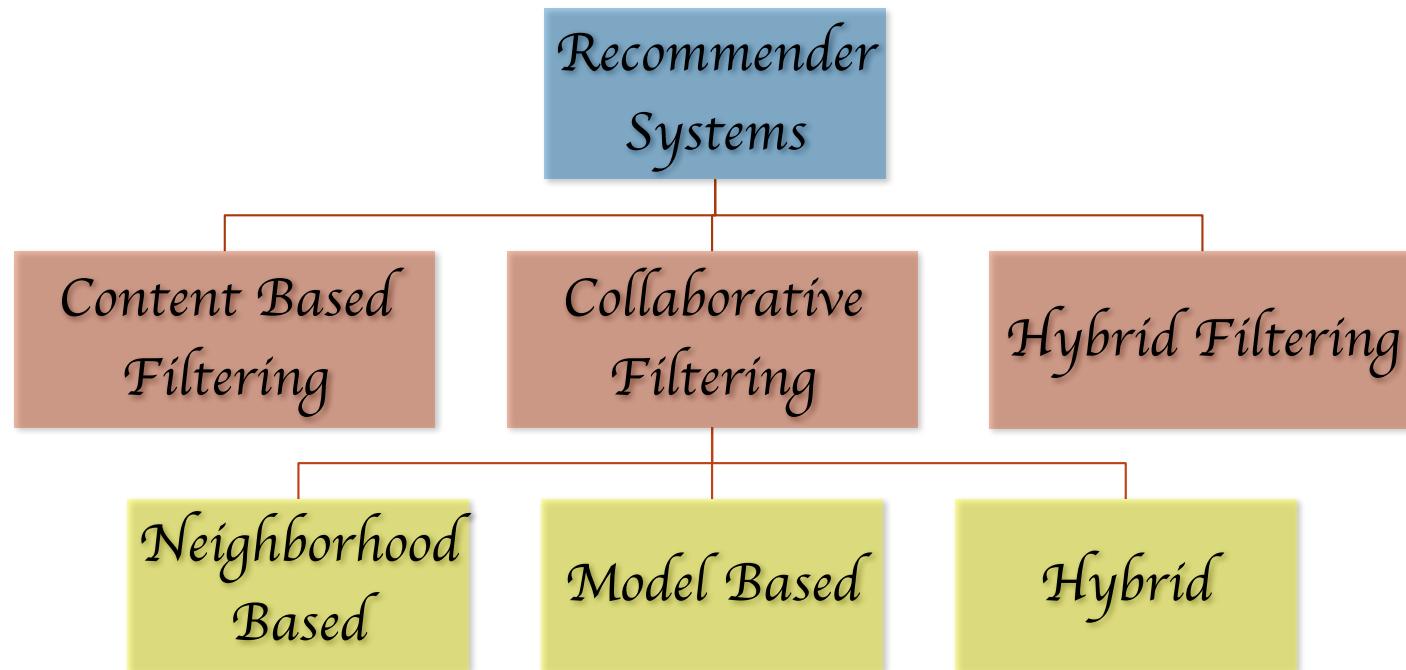
Recommender Systems

What are Recommender Engines?

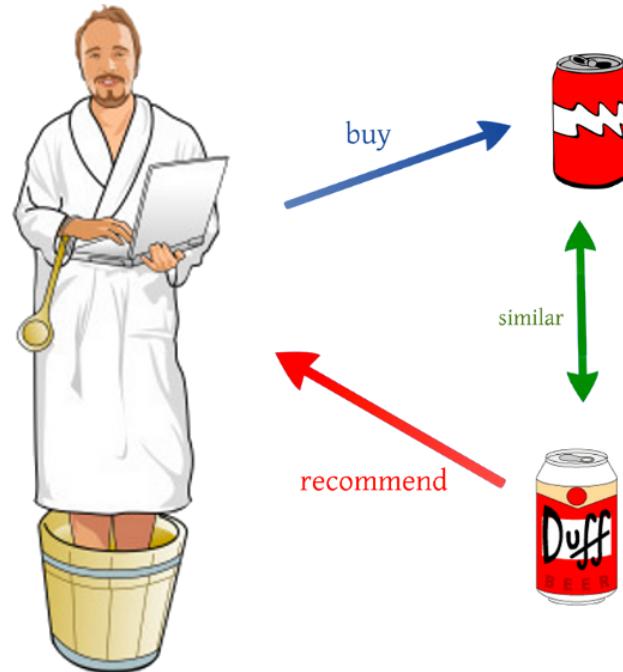


A recommender system or a recommendation system is a subclass of information filtering system that seeks to predict the "rating" or "preference" that a user would give to an item
 - Wikipedia

Types of Recommender Systems



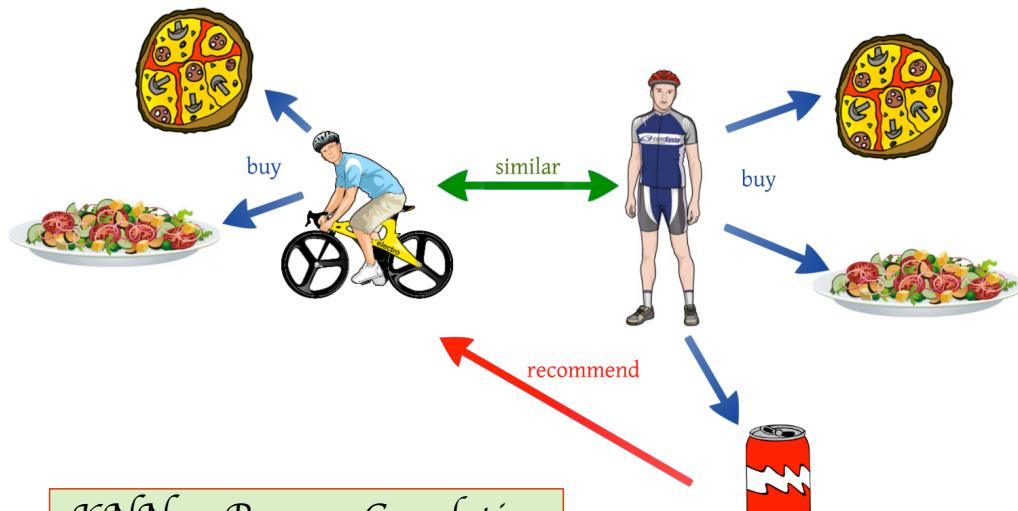
Content Based Filtering



Are based on item features and user preferences

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

Collaborative Filtering



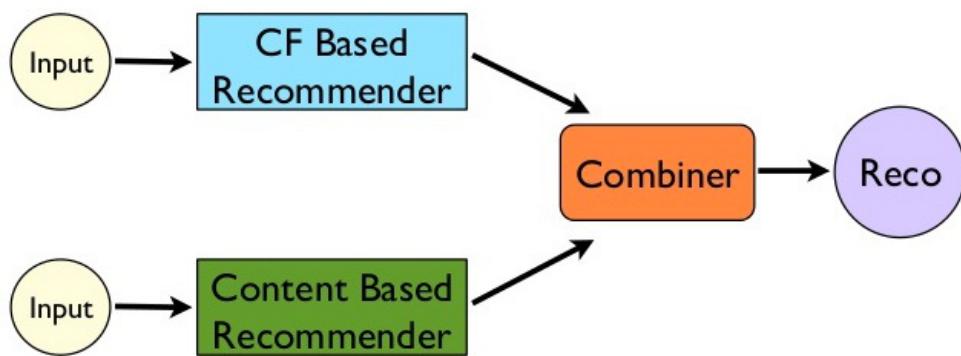
KNN or Pearson Correlation
is used for similarity

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

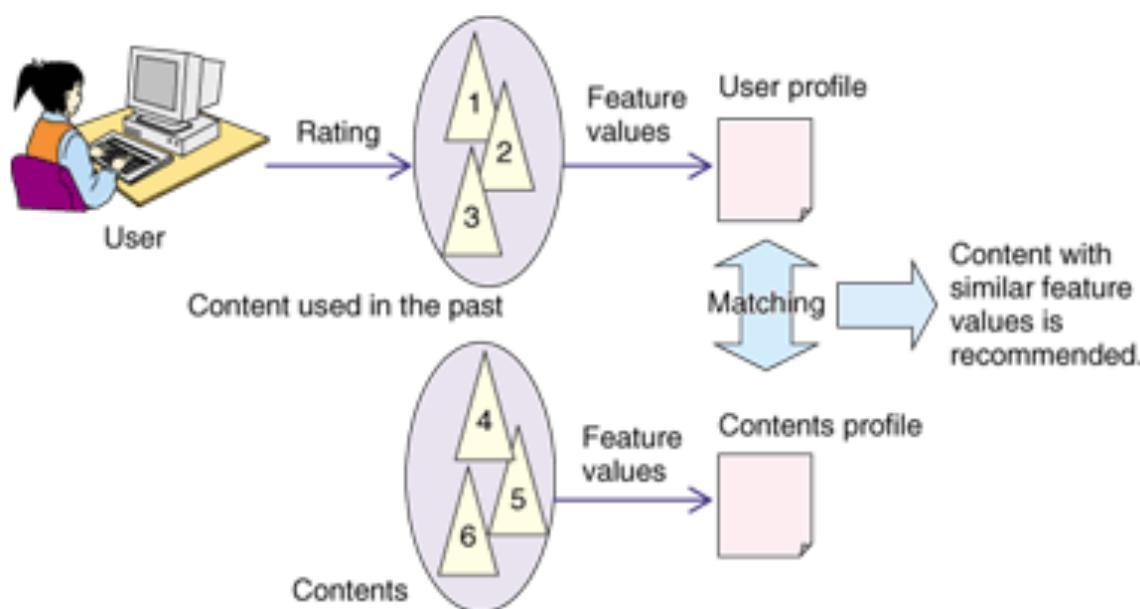
Hybrid Recommendations



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
<u>Article 1</u>	1	0	1	0	1	3
<u>Article 2</u>	0	1	1	1	0	3
<u>Article 3</u>	0	0	0	1	1	2
<u>Article 4</u>	0	0	1	1	0	2
<u>Article 5</u>	0	1	0	0	0	1
<u>Article 6</u>	1	0	0	1	0	2
DF	2	2	3	4	2	

User 1	User 2
1	-1
-1	1
	1
1	

CBF - Example

Step2 : Normalize features

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

$$1/\text{SQRT}(3)$$

CBF - Example

Step3 : Create User Profiles for each feature

Articles	Big Data	R	Python	Machine Learning	Learning Paths	Total attributes	User 1	User 2
<u>Article 1</u>	0.577350269	0	0.577350269	0	0.577350269	3	1	-1
<u>Article 2</u>	0	0.577350269	0.577350269	0.577350269	0	3	-1	1
<u>Article 3</u>	0	0	0	0.707106781	0.707106781	2		
<u>Article 4</u>	0	0	0.707106781	0.707106781	0	2		1
<u>Article 5</u>	0	1	0	0	0	1		
<u>Article 6</u>	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
<u>Article 1</u>	0.577350269	0	0.577350269	0	0.577350269
<u>Article 2</u>	0	0.577350269	0.577350269	0.577350269	0
<u>Article 3</u>	0	0	0	0.707106781	0.707106781
<u>Article 4</u>	0	0	0.707106781	0.707106781	0
<u>Article 5</u>	0	1	0	0	0
<u>Article 6</u>	0.707106781	0	0	0.707106781	0
<hr/>					
User Profiles					
User1	1.28445705	-0.577350269	0	0.129756512	0.577350269
User2	-0.577350269	0.577350269	0.707106781	1.28445705	-0.577350269
<hr/>					
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\$10:\$E\$29,\$B\$42:\$F\$43)	
User Profiles								SUMPRODUCT(array1, [array2], [array3], [array4], [array5], ...)	
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

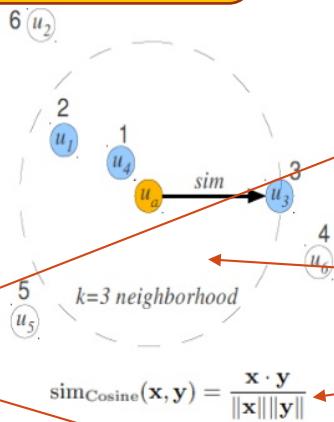
User Based Collaborative Filtering

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		

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



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

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

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	-	-

$k=3$

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!