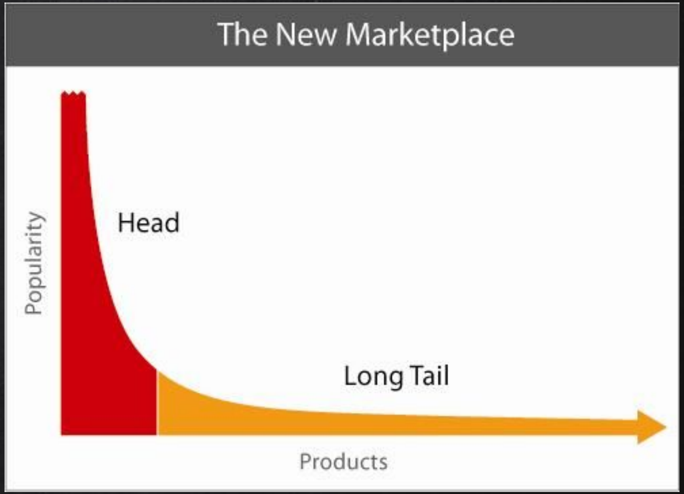
Recommender

Recommendation engine is deeply embedded in our lives

https://medium.com/@libreai/a-glimpse-into-deep-learning-for-recommender-systems-d66ae0681775

Why recommender system?



	Choice overload	**Long tail problem**
Customer	What to choose?	Never exposed to long tail products
Saler	What to promote?	Poor sales on long tail products

The dilemma of the recommender

Recommender underlying assumption:

What customer bought haven't address all their needs.

If the history is the golden standard, there is no need for recommender.

Normal model underlying assumption:

History is going to repeat itself

If history is not going to repeat itself, there is no need for predictive model

An effective recommender needs to cater diverse user needs

	Meaning	History	Good recommender	Bad recommender
Familiarity	“This makes sense. I do by it regularly. It is convenient for me to see it on the recommendation.”			 “What is this?!”
Novelty	“Don’t keep telling me everything I already knew.”			
Diversity	”Just because I bought chips before, doesn’t mean chips are the only thing I will ever need!”			
Serendipity	“This is a match made in heaven.”		 “It’s summer and after all these fast food, I really need some exercise 😊.”	 “What the ??? I am a single man…”

Association rule mining – Market Basket Analysis

Basket 1

Basket 2

Basket 3

Basket 4

$$support(x \to y) = \frac{count(x \cup y)}{totalcount}$$

$$confidence(x \to y) = \frac{count(x \cup y)}{count(x)}$$

$$lift(x \to y) = \frac{confidence(x \to y)}{support(y)} = \frac{support(x \to y)}{support(x) \times support(y)}$$

	Support	Confidence	Lift
{beer} -> {diaper}	50%	67%	1.33
{beer, milk} -> {diaper}	100%	100%	2
{beer} -> {coke}	25%	0.33	0.67

Typically the association is interesting if it satisfy certain support, confidence and have a lift that is bigger than 1

https://pbpython.com/market-basket-analysis.html

Smart Adaptive Recommendation System

Get the similarity by Association Rule Mining

$$lift(x \rightarrow y) = \frac{confidence(x \rightarrow y)}{support(y)} = \frac{support(x \rightarrow y)}{support(x) \times support(y)}$$

Measure the similarity of the item to the items the users already purchased

An oversimplified explanation

	Lift
{beer} -> {chip}	1.5
{beer} -> {pizza}	2
{beer} -> {milk}	0.2
{beer} -> {coke}	0.5

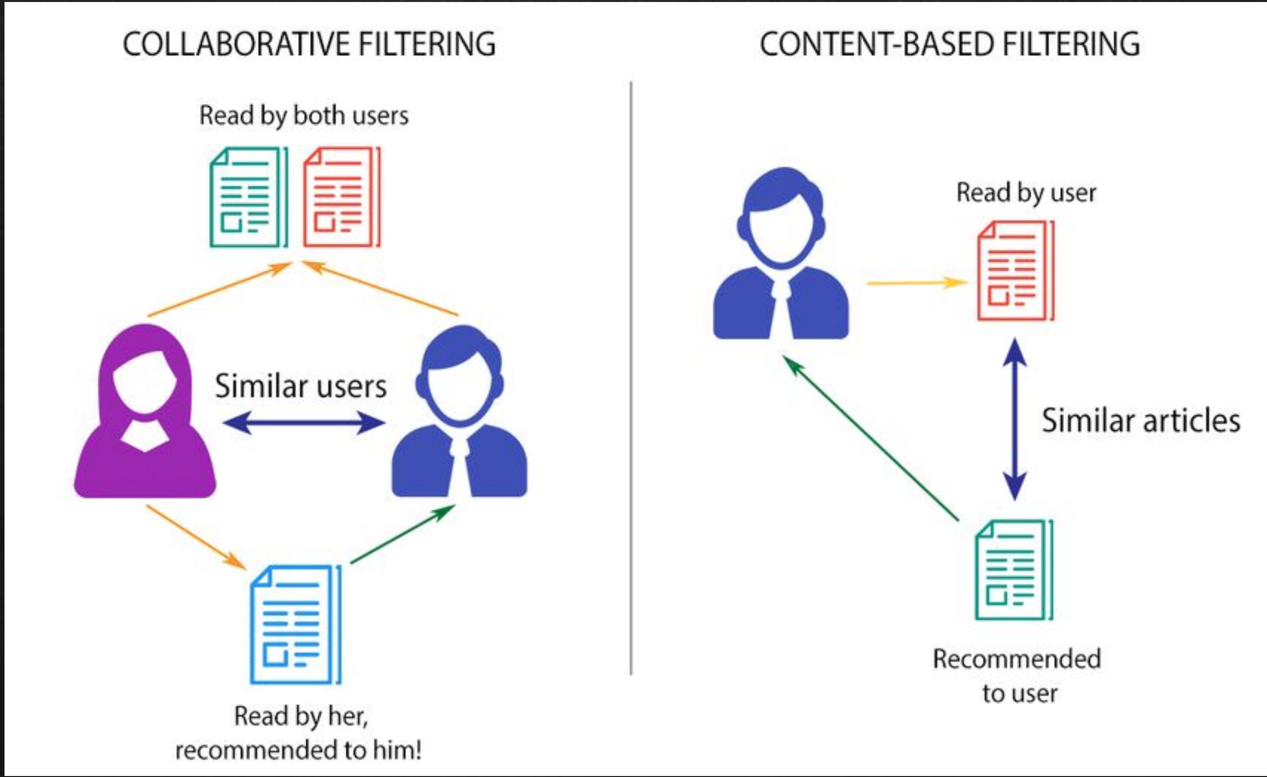
	Past purchase	Beer
	chip & pizza	Recommend
	milk & coke	NOT recommend

https://github.com/Microsoft/Recommenders/blob/master/notebooks/02_model/sar_deep_dive.ipynb

Filtering Methods

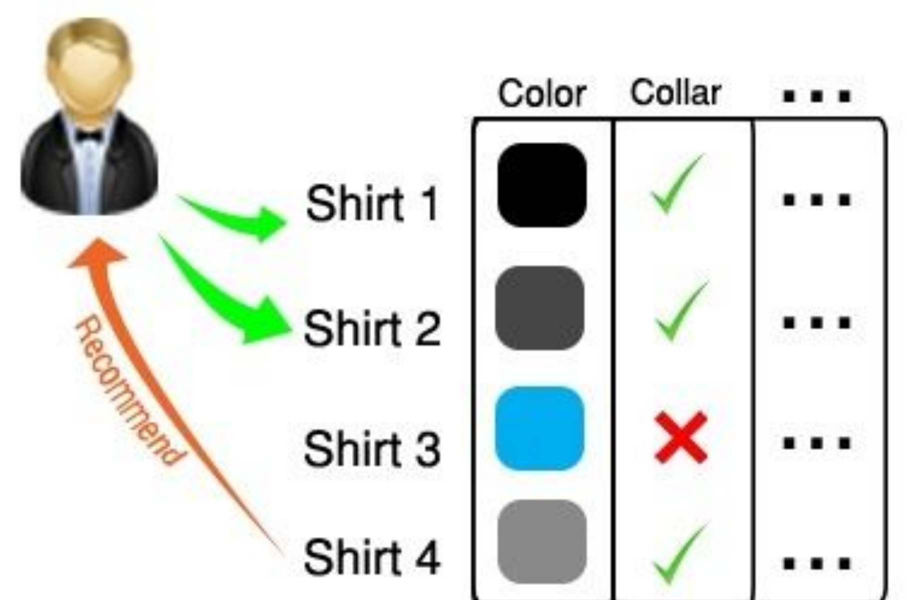
The key is to define and calculate similarity

Collaborative filtering v.s. Content based filtering



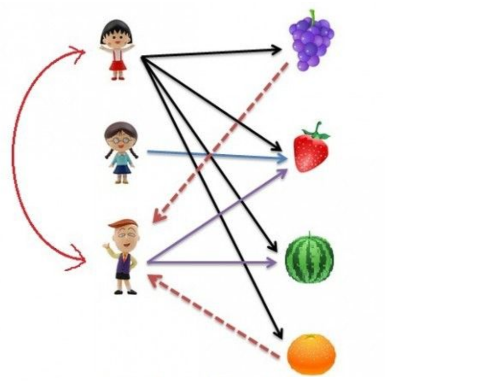
https://www.researchgate.net/publication/318129942_Recommender_System_for_News_Articles_using_Supervised_Learning

Content based filtering

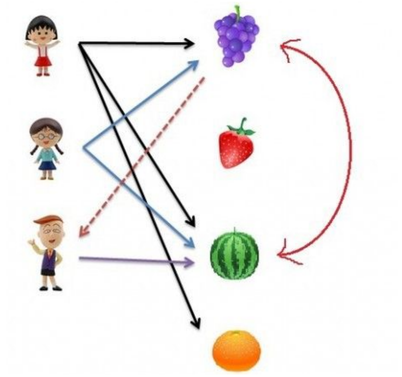


From internet

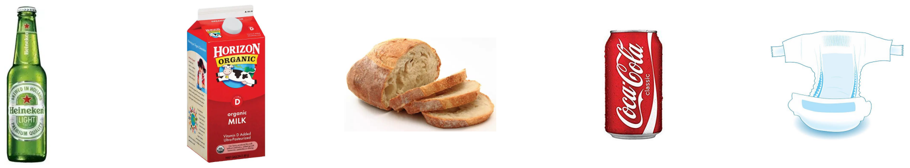
Collaborative Filtering



User-based filtering



Item-based filtering



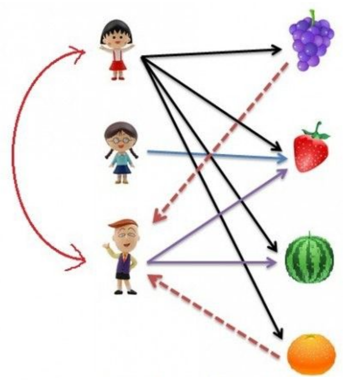
0	2	1	0	1
0	1	2	1	1
3	0	0	4	1
2	0	3	7	0

User similarity

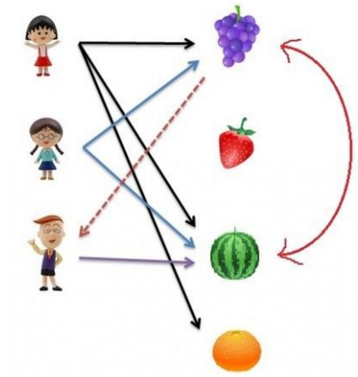
Item similarity

From internet and modified

Collaborative Filtering



User-based filtering



Item-based filtering

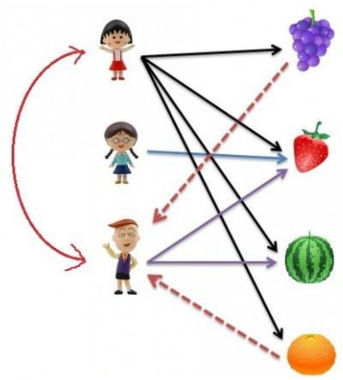
0	2	1	0	1
0	1	2	1	1
3	0	0	4	1
2	0	3	7	0

User similarity

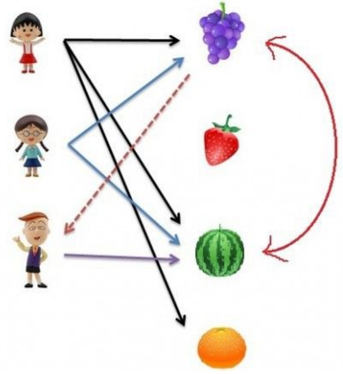
Item similarity

From internet and modified

Collaborative Filtering



User-based filtering



Item-based filtering

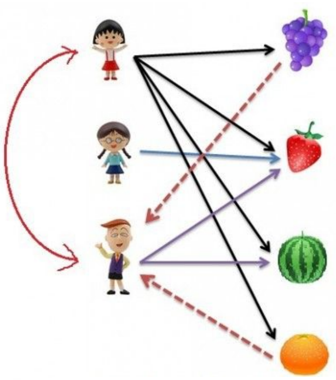
0	2	1	0	1
0	1	2	1	1
3	0	0	4	1
2	0	3	7	0

User similarity

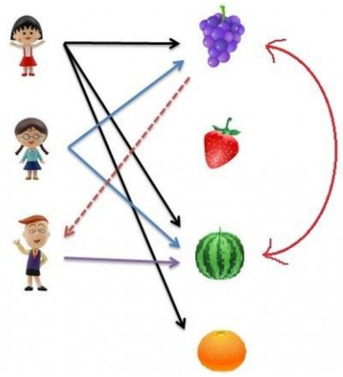
Item similarity

From internet and modified

Collaborative Filtering



User-based filtering



Item-based filtering

0	2	1	0	1
0	1	2	1	1
3	0	0	4	1
2	0	3	7	0

User similarity

Item similarity

From internet and modified

Collaborative Filtering with Matrix Factorizations

Item

User	W	X	Y	Z
A		4.5	2.0	
B	4.0		3.5	
C		5.0		2.0
D		3.5	4.0	1.0

Rating Matrix

Seriousness

Masculin

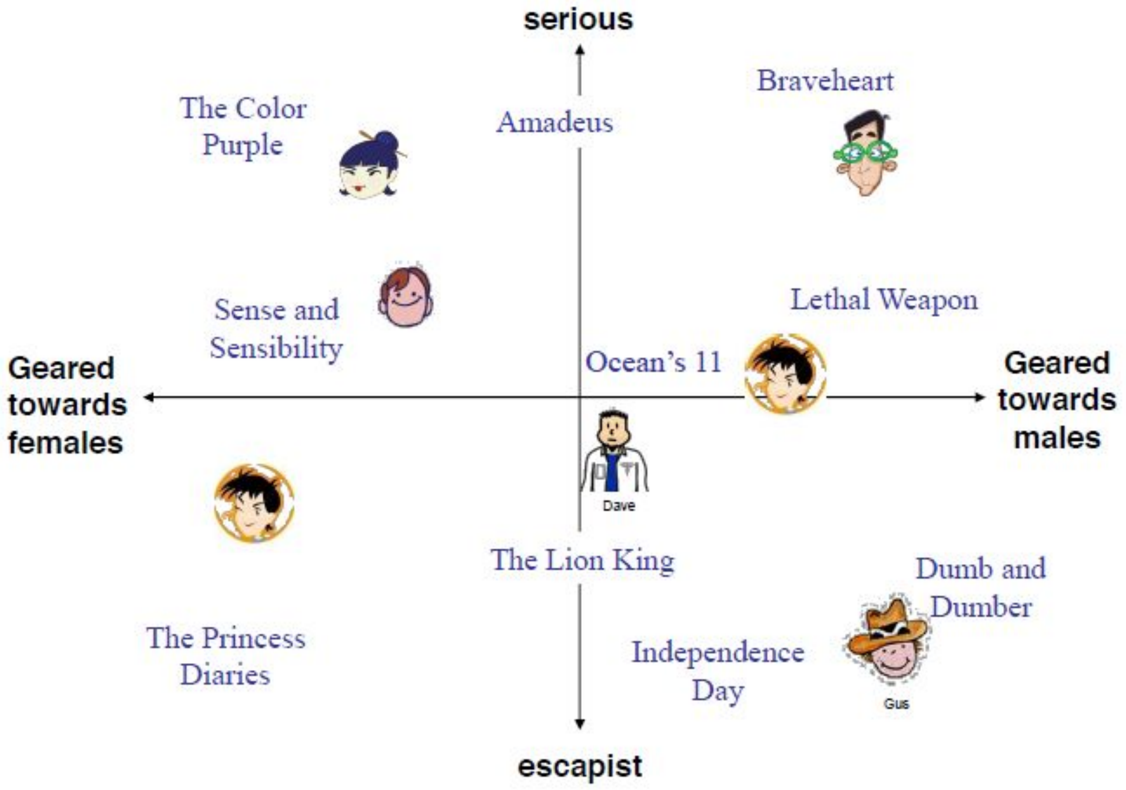
A	1.2	0.8
B	1.4	0.9
C	1.5	1.0
D	1.2	0.8

User Matrix

X

W	X	Y	Z
1.5	1.2	1.0	0.8
1.7	0.6	1.1	0.4

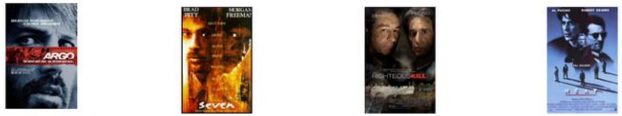
Item Matrix



https://github.com/Microsoft/Recommenders/blob/master/notebooks/02_model/als_deep_dive.ipynb

Bring in additional information to recommender system

Item inforamtion



User inforamtion

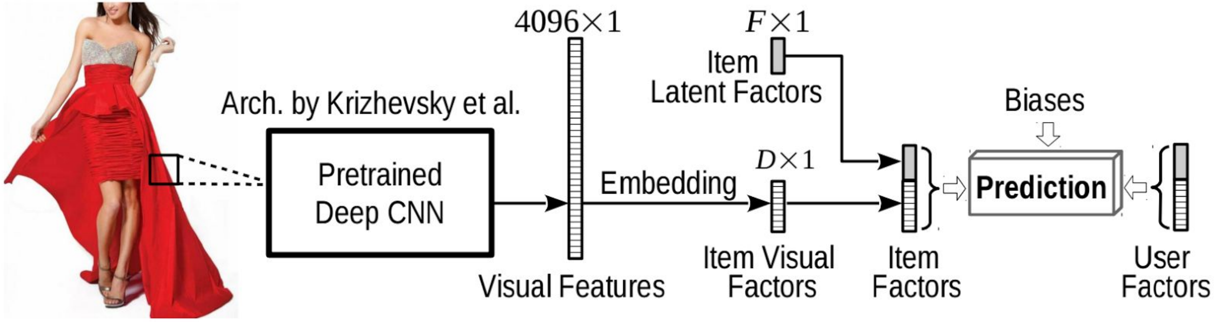
John	5	1	3	5
Tom	?	?	?	2
Alice	4	?	3	?

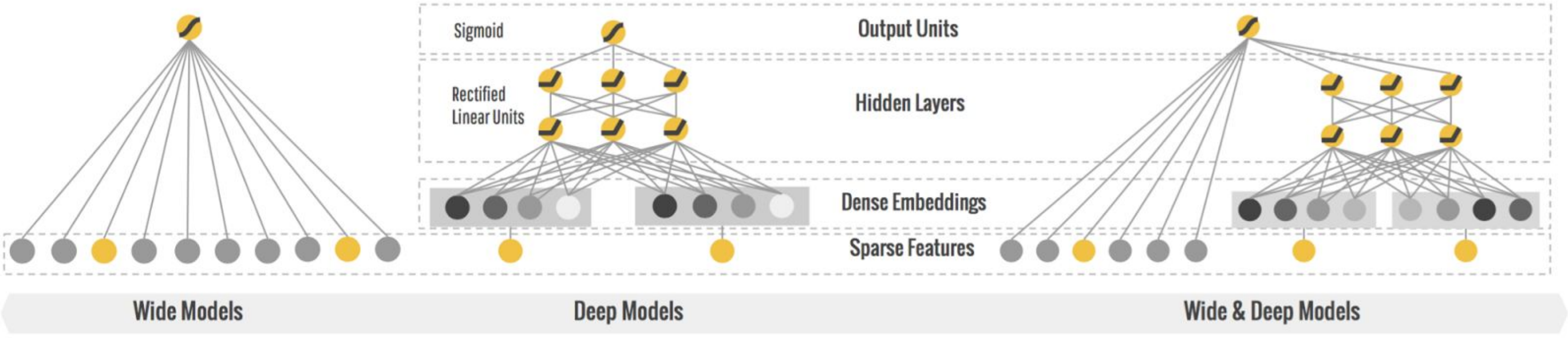
Contextual inforamtion:

Location
Time
Weather
Special event
Holiday
......

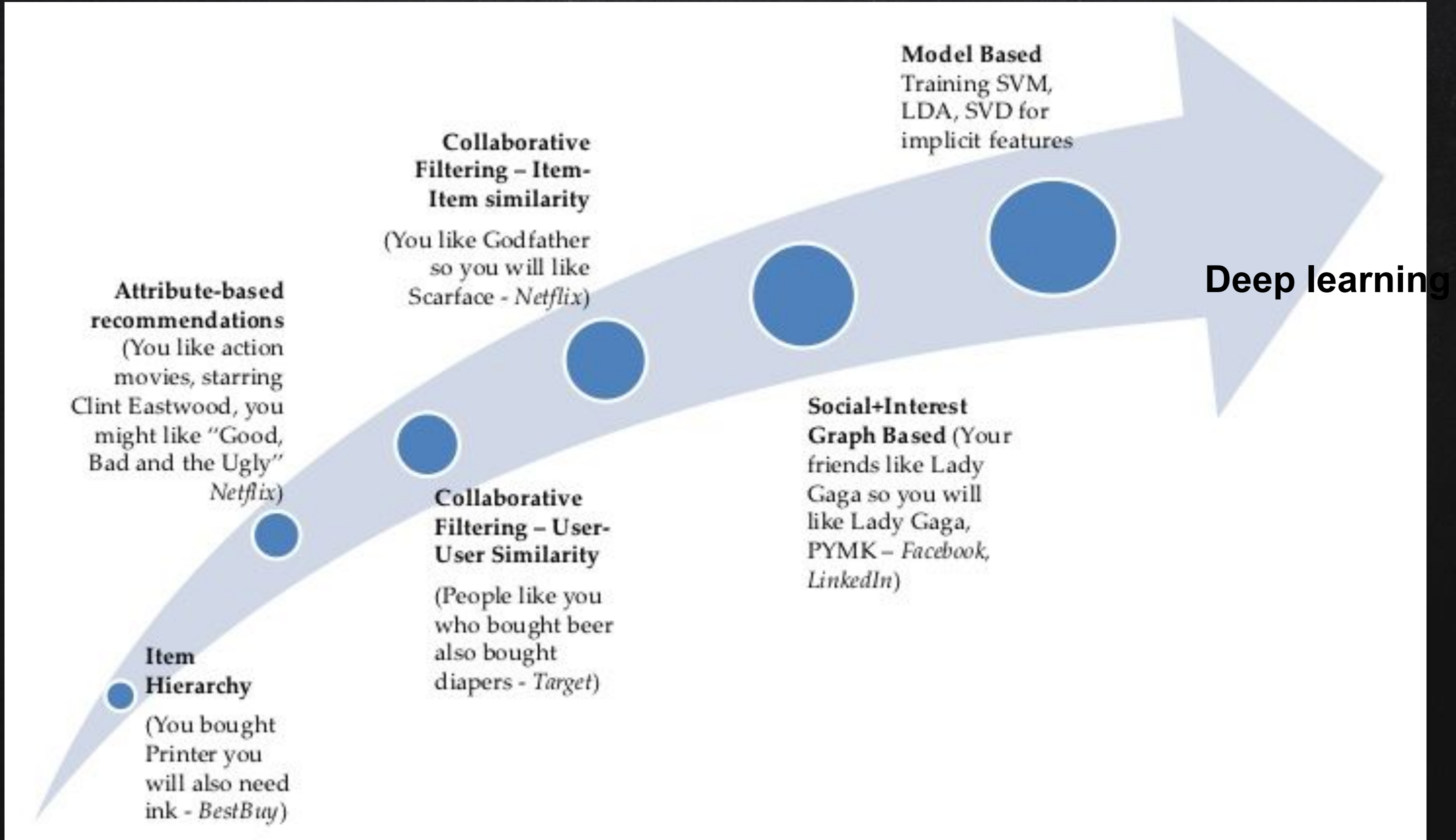
From internet and modified

Deep learning for recommendation system





https://github.com/Tian-Su/Recommenders/blob/tian-test/notebooks/02_model/deepNN_test.ipynb



From internet and modified

Evaluation

P@K

R@K

NDCG

https://en.wikipedia.org/wiki/Discounted_cumulative_gain

The value of the recommendation system

Netflix: 2/3 of the movies watched are recommended

Google news: recommendation generate 38% more click through

Amazon: 35% incremental sales from recommendation

Linkedin: Job matching algorithm improve the performance by 50%