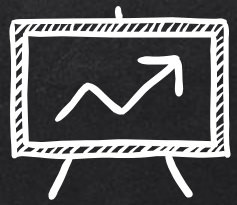


RECOMMENDATION ENGINE

Q1

Recommendation engine

What? Why? How?

What is recommendation engine?

- ✘ Filters the data using different algorithms
- ✘ Captures the past behavior of a customer
- ✘ Recommends the most relevant items to users that likely to buy

1st visit?

- ✘ Recommend best selling products
- ✘ Recommend products which could bring maximum profit

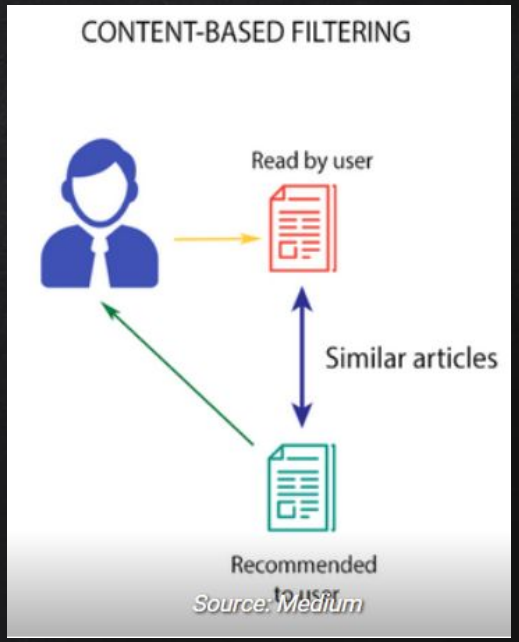
Why recommendation engine?

- ✘ create a positive impact on the user experience
- ✘ Lead to frequent visits
- ✘ Improve the quality and effectiveness of marketing campaign
- ✘ Increase the sales and profit rate

How does recommendation engine works?

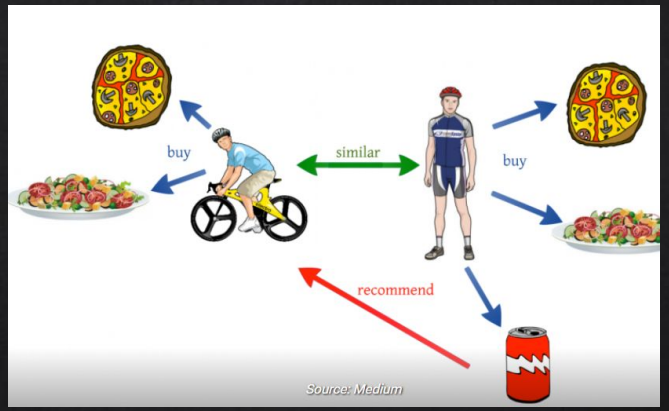
- ✘ Content based filtering
- ✘ Recommends products which are similar to the ones that a user has liked in the past

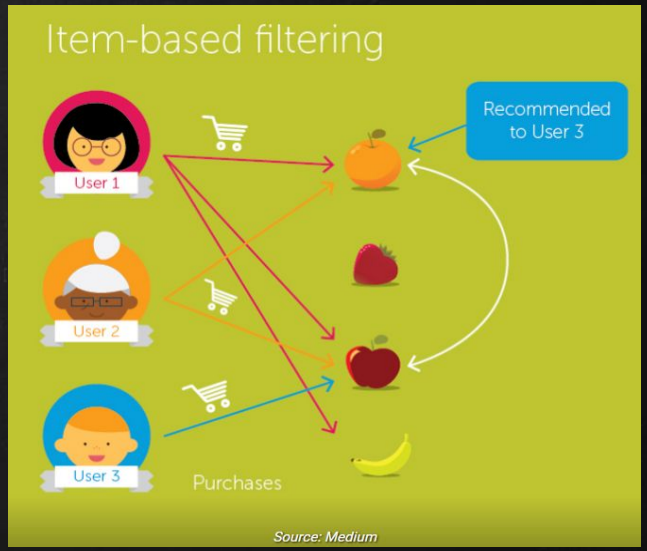
Examples: Netflix



How does recommendation engine works?

- ✘ Collaborative filtering
  - ✘ User – User
  - ✘ Item – Item





Example: Amazon, Social network

Recommendation Engine Approaches

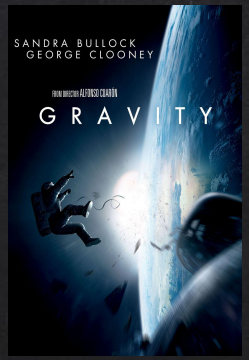
- ✘ Multi-criteria recommender systems
- ✘ Risk-aware recommender systems
- ✘ Mobile recommender systems
- ✘ Hybrid recommender systems

Sally

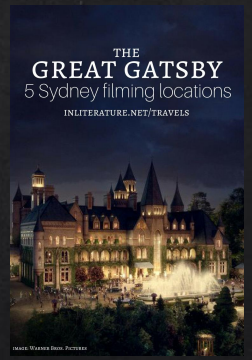
M1

M2

Lucy



M3



M4

M5

Pepper

Linus

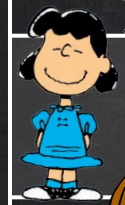
Ratings

5/5 ★★★★★

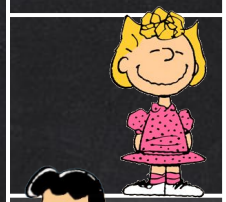
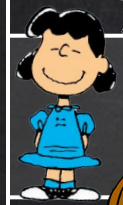
M1

★★★

3/5

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Ratings

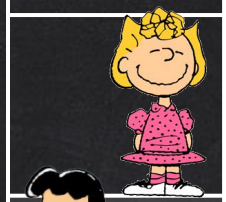
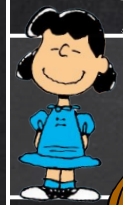
	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Ratings

=

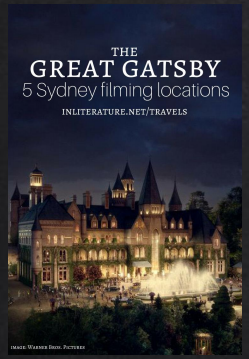
	M1	M2	M3	M4	M5
	3	1	1	3	1
					
	3	1	1	3	1
					

Ratings

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

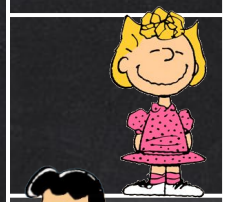
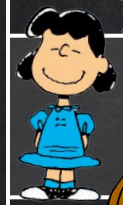
Hidden Dependency

M1 = M4

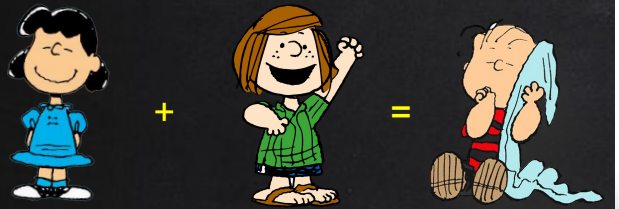


	M1	M2	M3	M4	M5
	3			3	
	1			1	
	3			3	
	4			4	

Ratings

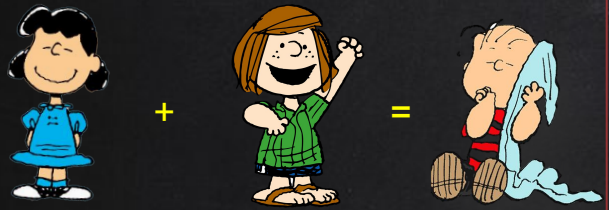
	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

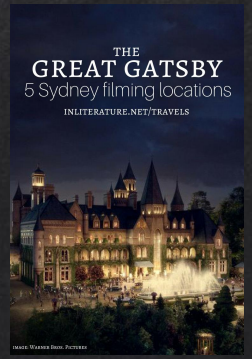
Hidden Dependency



	M1	M2	M3	M4	M5
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Hidden Dependency





M4

M1

Drama

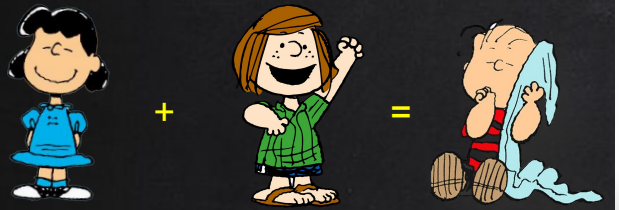
M2

M3

M5

Science Fiction

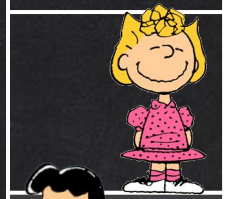
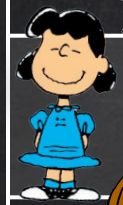
Hidden Dependency



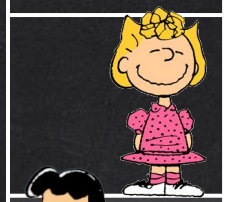
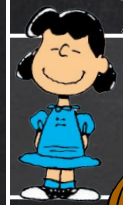
Si-Fi

Drama

Drama & Si-Fi

	M1	M2	M3	M4	M5
					
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Ratings

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Hidden Dependency on rows and columns

M5 = Average(M2, M3)

	M1	M2	M3	M4	M5
		1	1		1
		2	4		3
		1	1		1
		3	5		4

Prediction

M5 = ?

Pepper hasn't watch Movie 5.

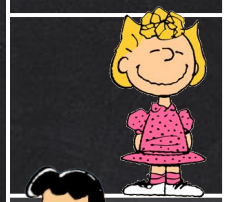
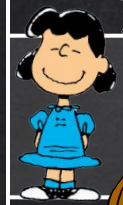
What's his rating on M5?

Dependency

=

M5

1/5 ★

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	
	4	3	5	4	4

Q5

What is matrix factorization?

What is Matrix Factorization?

- ✘ A class of collaborative filtering algorithms
- ✘ A kind of Latent factor model
- ✘ Decomposing the user-item interaction matrix into the product of two lower dimensionality rectangular matrices

$$\tilde{R} = HW$$

Why?

Superior to classic nearest-neighbor techniques:

- ✘ allowing the incorporation of additional information such as implicit feedback, temporal effects, and confidence levels

Solving the problems:

- ✘ The amount of information does not increase linearly with the increase of the vector dimension because of the correlation between the items
- ✘ The matrix elements are sparse.(so the calculation result is unstable, resulting a large difference when change the vector dimensions.)

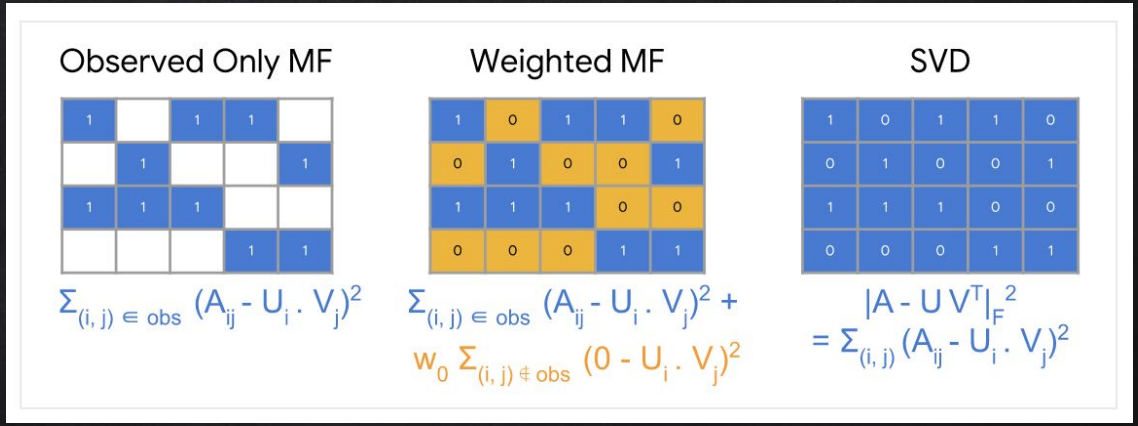
Matrix Factorization

- Matrix Factorization is a simple embedding model. Given the feedback matrix $A \in R^{m \times n}$, where m is the number of users (or queries) and n is the number of items, the model learns:
- A user embedding matrix $U \in R^{m \times d}$, where row i is the embedding for user i.
- An item embedding matrix $V \in R^{n \times d}$, where row j is the embedding for item j.

Matrix Factorization

The embeddings are learned such that the product UVT is a good approximation of the feedback matrix A. Observe that the (i,j) entry of U.VT is simply the dot product ⟨Ui,Vj⟩ of the embeddings of user i and item j, which you want to be close to Ai,j.

Matrix Factorization



$$\min_{U \in \mathbb{R}^{m \times d},\, V \in \mathbb{R}^{n \times d}} \sum_{(i,j) \in \text{obs}} (A_{ij} - \langle U_i, V_j \rangle)^2.$$

Matrix Factorization

$$\min_{U \in \mathbb{R}^{m \times d},\, V \in \mathbb{R}^{n \times d}} \|A - UV^T\|_F^2.$$

$$\min_{U \in \mathbb{R}^{m \times d},\, V \in \mathbb{R}^{n \times d}} \sum_{(i,j) \in \text{obs}} (A_{ij} - \langle U_i, V_j \rangle)^2 + w_0 \sum_{(i,j) \notin \text{obs}} (\langle U_i, V_j \rangle)^2.$$

- ✘ SVD
- ✘ NMF
  - Loss function can be minimized by SGD, ALS

Other Matrix Factorization Models

- ✗ SVD++
- ✗ Asymmetric SVD
- ✗ Hybrid Matrix Factorization
- ✗ Deep-Learning Matrix Factorization