# Big Data

## (Recommendation System)

Instructor: Thanh Binh Nguyen

September 1st, 2019

S3Lab
*Smart Software System Laboratory*

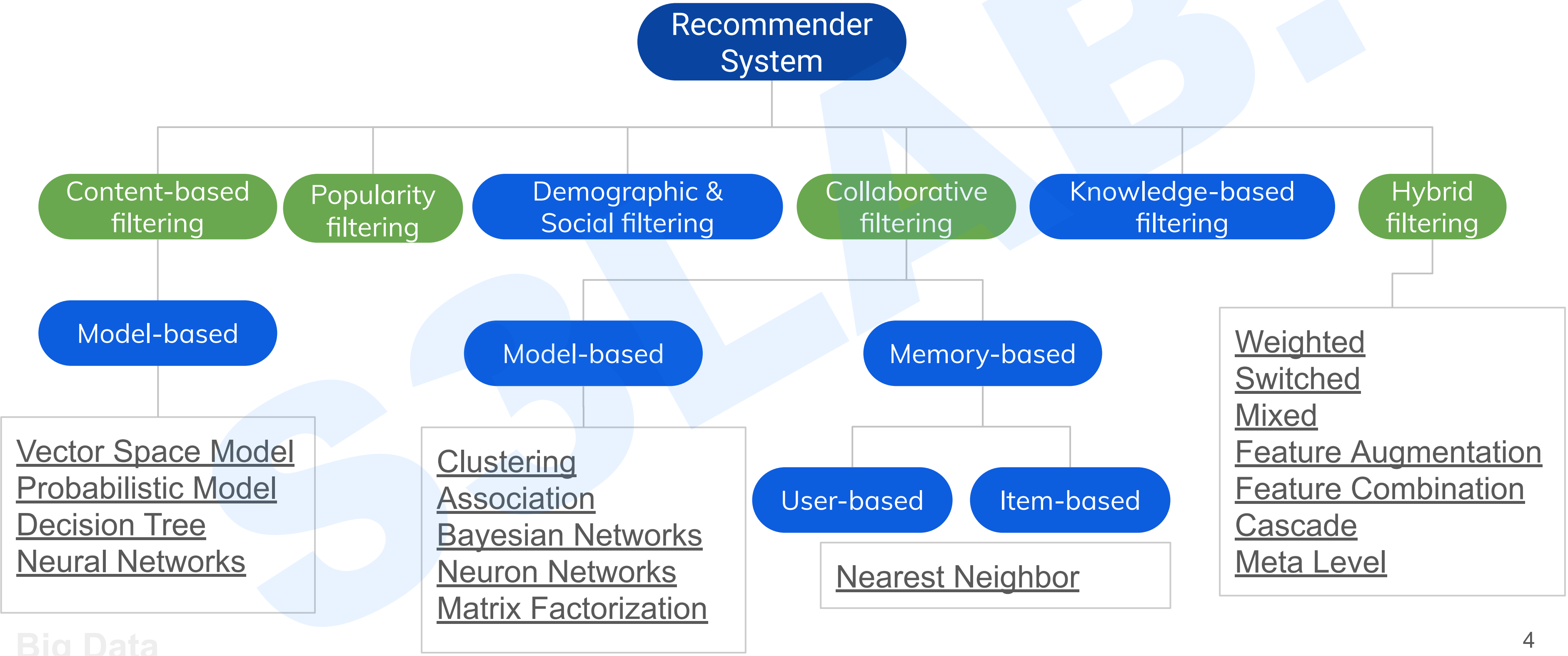
“Big data is at the foundation of all the megatrends that are happening today, from social to mobile to cloud to gaming.”

– Chris Lynch, Vertica Systems

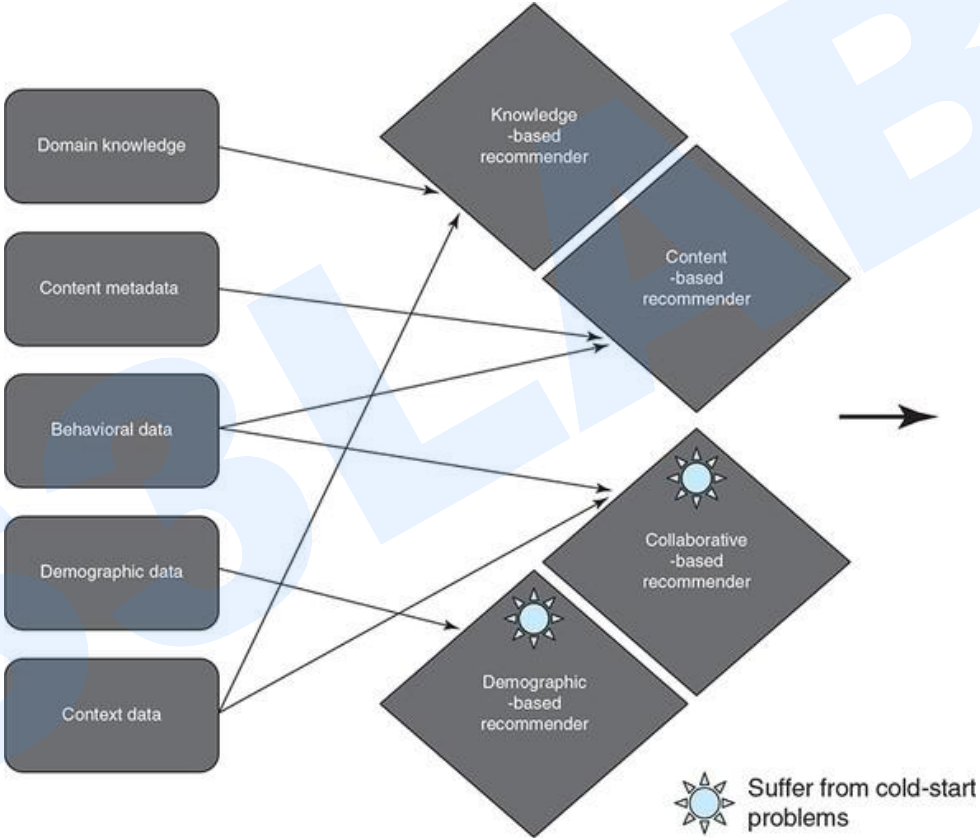
# Applications

- **Facebook**—"People You May Know"
- **Netflix**—"Other Movies You May Enjoy"
- **LinkedIn**—"Jobs You May Be Interested In"
- **Amazon**—"Customer who bought this item also bought …"
- **Google**—"Visually Similar Images"
- **Google adsense**—"The products you may be interested"
- **YouTube**—"Recommended Videos"
- **Waze**—"Best Route"
- **Coursera's** — "Recommended courses…"
- **Spotify** — "Recommended songs…"

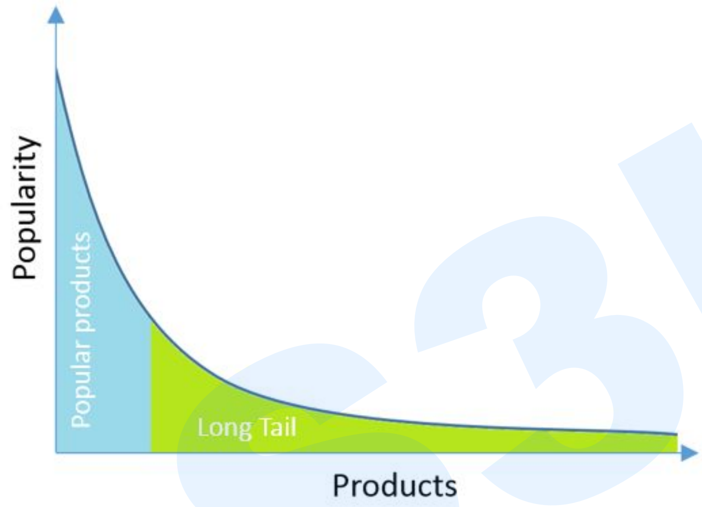
# Traditional Approaches

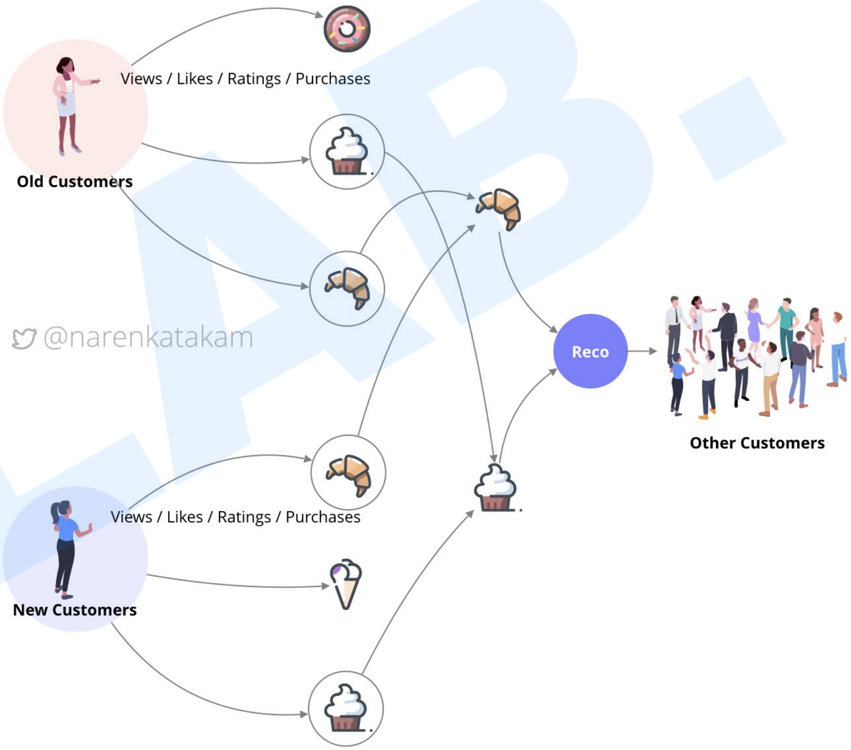


# Traditional Approaches



# Popularity Filtering



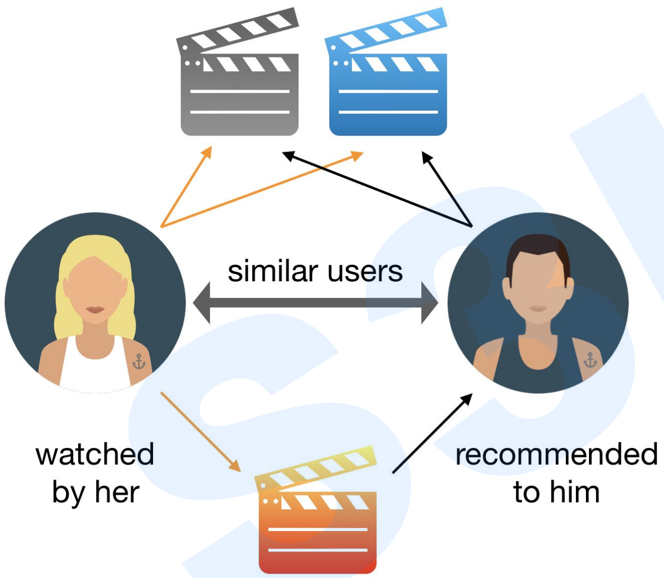

@narenkatakam

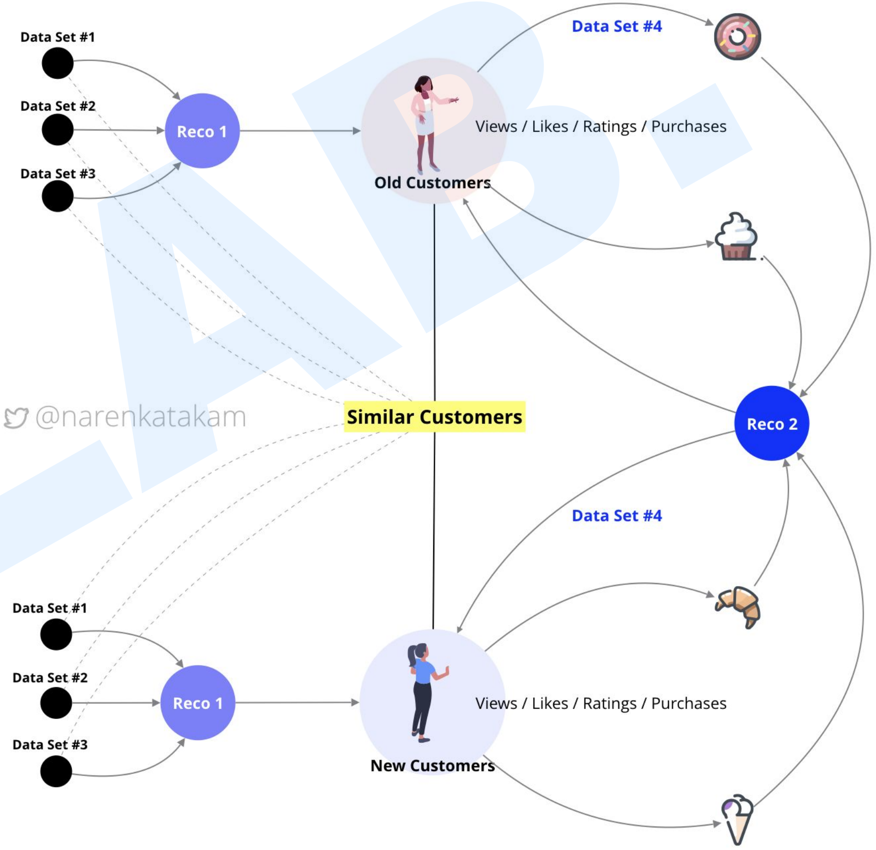
# Popularity Filtering

| | |
|---|---|
| **Popularity** is an approach where you recommend items that are popular (e.g. most downloaded, watched, high impact). | |
| **Input:** uses usage data and item content (e.g. categories). | |
| **Pros** | **Cons** |
| • Relatively easy to implement<br>• Good baseline algorithm<br>• Helps with new user cold start problem | • Needs standardised products<br>• Often needs some type of item categorisation<br>• Won't recommend new items (fewer opportunities to learn)<br>• Recommendation list tends not to change much |

# Collaborative Filtering

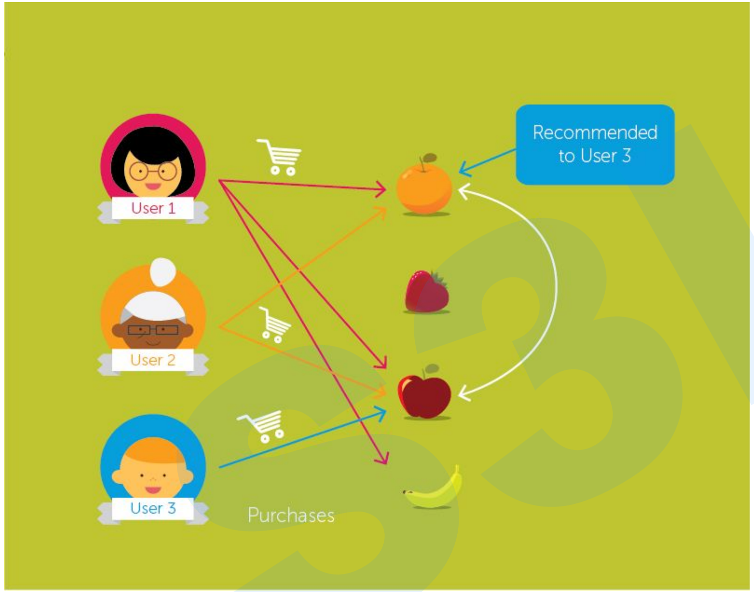
## ***Memory Based:*** *User - User*

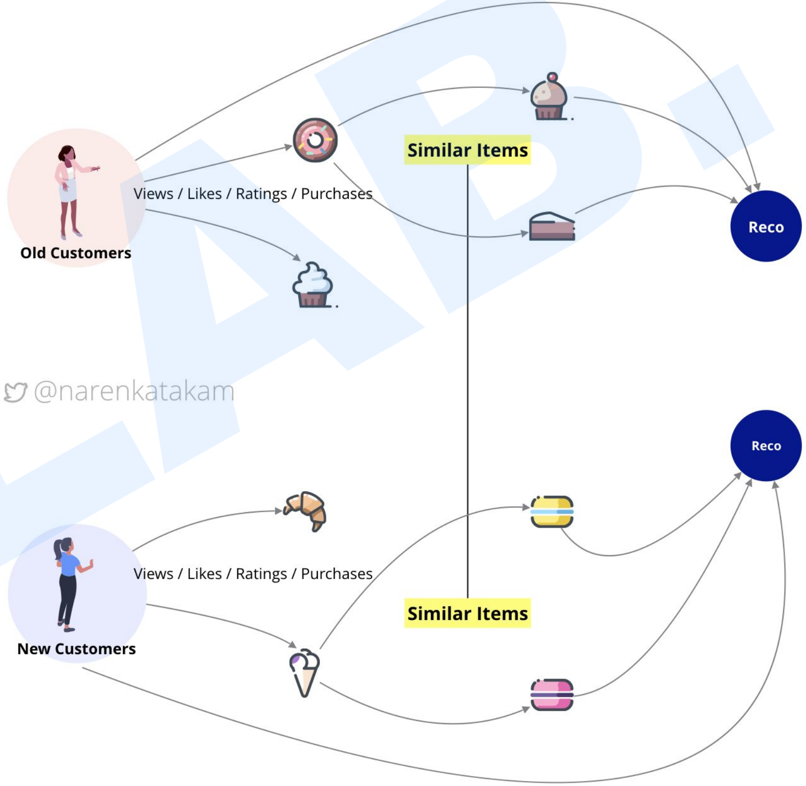


# Collaborative Filtering

## ***Memory Based:*** *Item - Item*

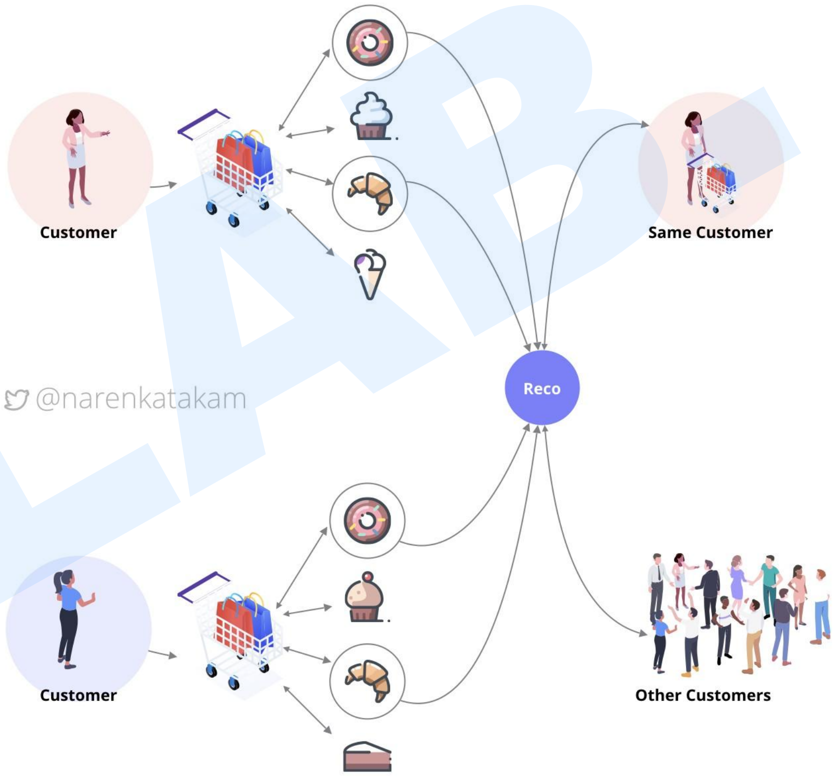


# Collaborative Filtering

*Association rules mining model*

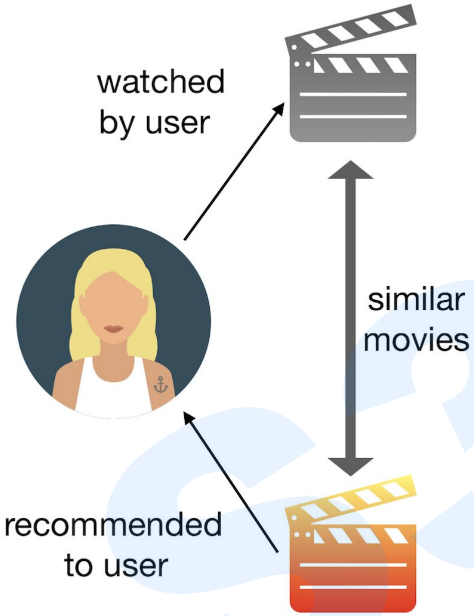
- 80% customer who like A and B also like C, 20% of all users like all three items.
- 70% of all item liked by user X and Y are also liked by user Z, 30% of all items are liked by all three users.

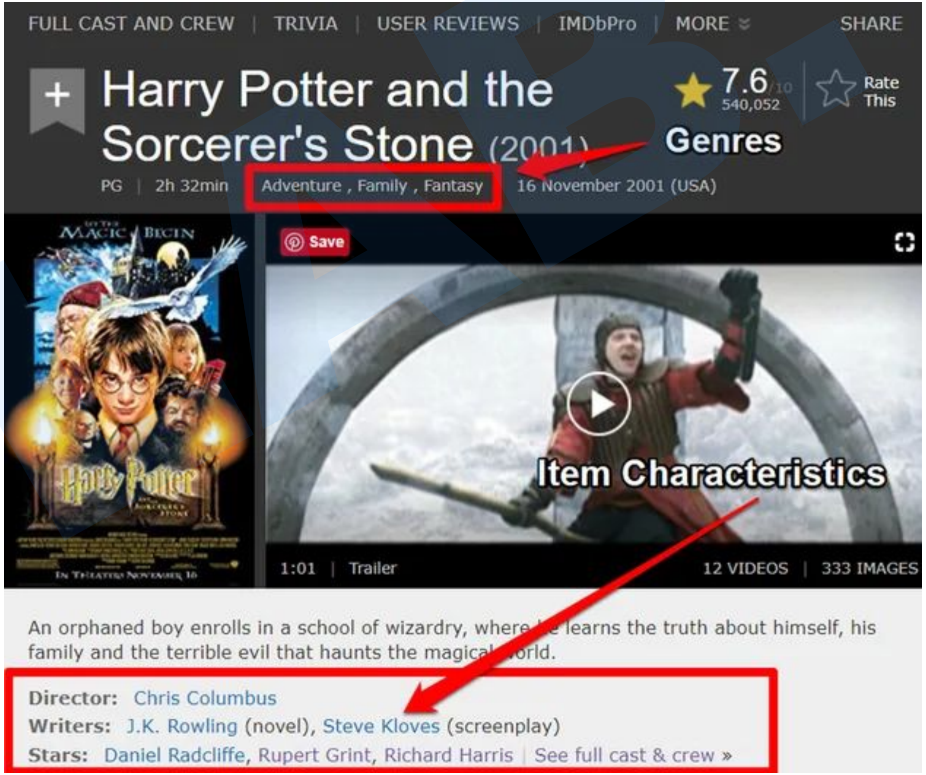


# Collaborative Filtering

| | |
|---|---|
| **Collaborative filtering** looks for patterns in the user activity to produce user specific recommendations. | |
| **Input:** depends only on usage data (e.g. ratings, purchases, downloads, user preferences). | |
| **Types:**<br>• Neighbourhood-based CF (user-based, item-based)<br>• Model-based CF (matrix factorisation, restricted boltzmann machines, bayesian networks, etc) | |
| **Pros** | **Cons** |
| • Minimal domain knowledge required<br>• User and item features are not required<br>• Produces good enough results in most cases | • Cold start problem<br>• Needs standardised products<br>• Requires high user:item ratio (1:10)<br>• Popularity bias (doesn't play well with the long tail)<br>• Can be difficult to provide explanations |

# Content-Based Filtering



# Content-Based Filtering

<table>
<tr><td colspan="2">Content-based filtering recommends items with similar content (e.g. metadata, description, topics) to the items the user has liked in the past.</td></tr>
<tr><td colspan="2">Input: depends only on the content/descriptions of the items and the users (but not usage data).</td></tr>
<tr><td colspan="2">Types:<br>• Information Retrieval (e.g. tf-idf, Okapi BM25)<br>• Machine Learning (e.g. Naive Bayes, support vector machines, decision trees, etc)</td></tr>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>• No cold start problem<br>• No need for usage data<br>• No popularity bias, can recommend items with rare features<br>• Can user content features to provide explanations</td><td>• Item content needs to be machine readable and meaningful<br>• Easy to pigeonhole the user<br>• Difficult to implement serendipity<br>• Difficult to combine multiple item's features together</td></tr>
</table>

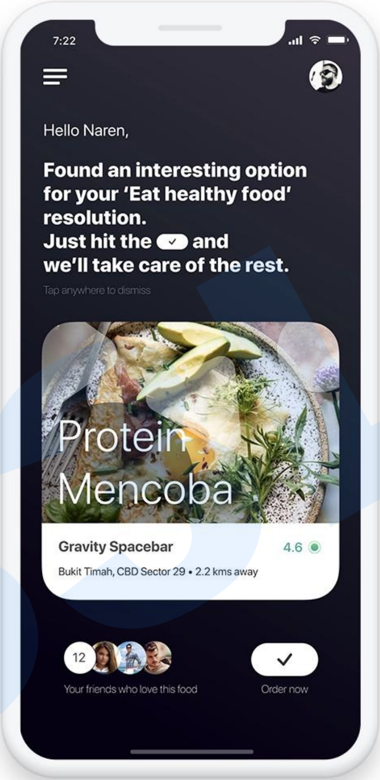
# Demographic & Social Filtering

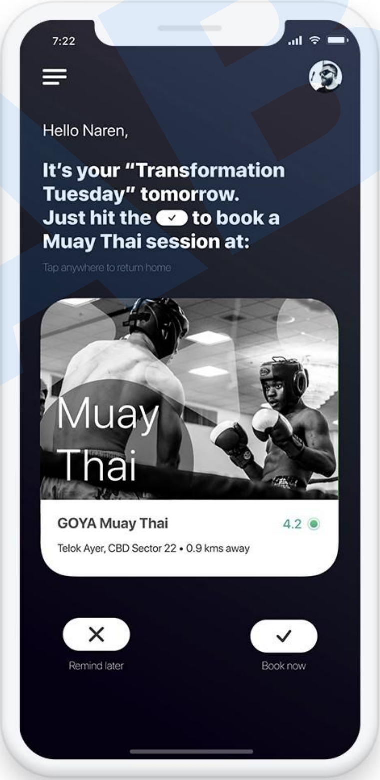
Modal screen to increase the engagement on the screen

Understanding the user and the context to recommend an item which has a high probability of acceptance by the user.

Quick card interface to give relevant information to the user to act or compelling info to make the user tap for more info

Call to action and relevant information to make the sell. In this case to increase the trust quotient

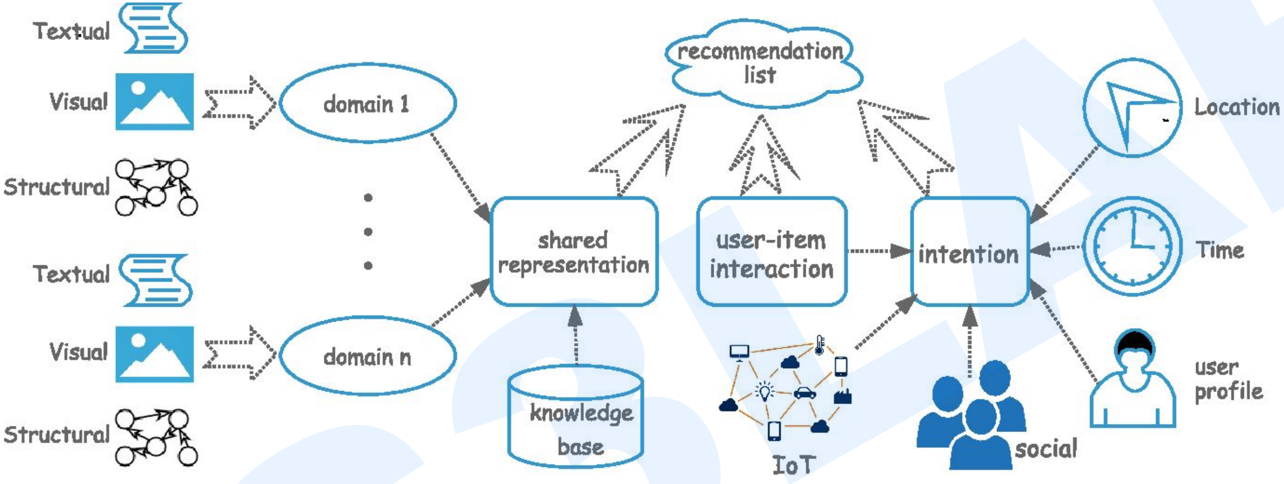


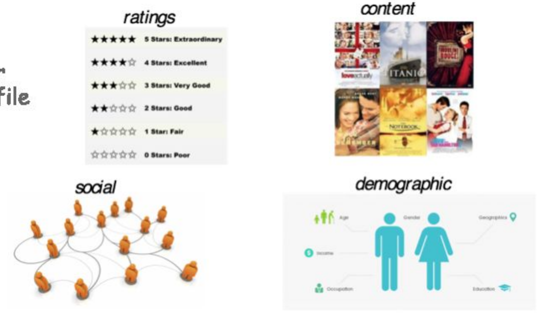




Understanding the user and the context to recommend an item which has a high probability of acceptance by the user.

Quick call to action

# Demographic & Social Filtering



# Knowledge-based Filtering

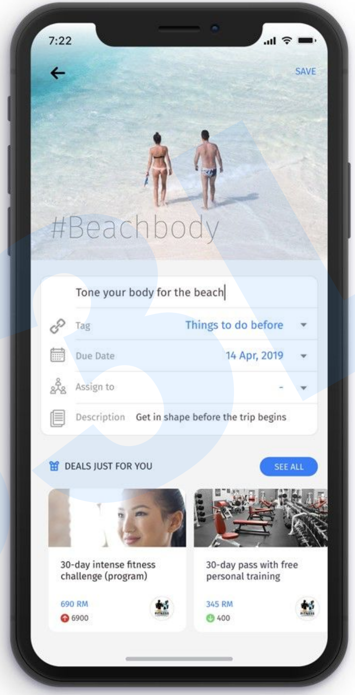
## ***Contextual - Aware*** *Filtering*

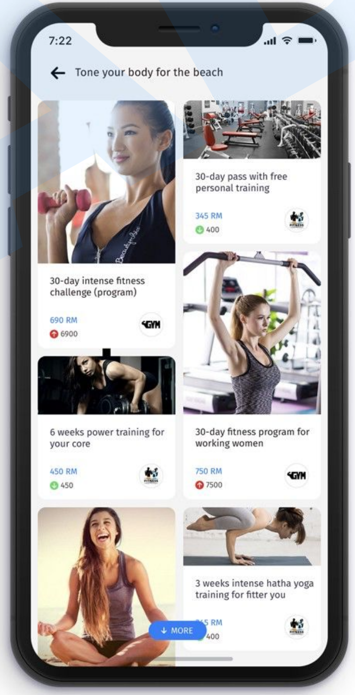
Context sync

Context tag sync

Edit title, tag section, add due date, assign to group member & add description to the task

Contextual & relevant deals



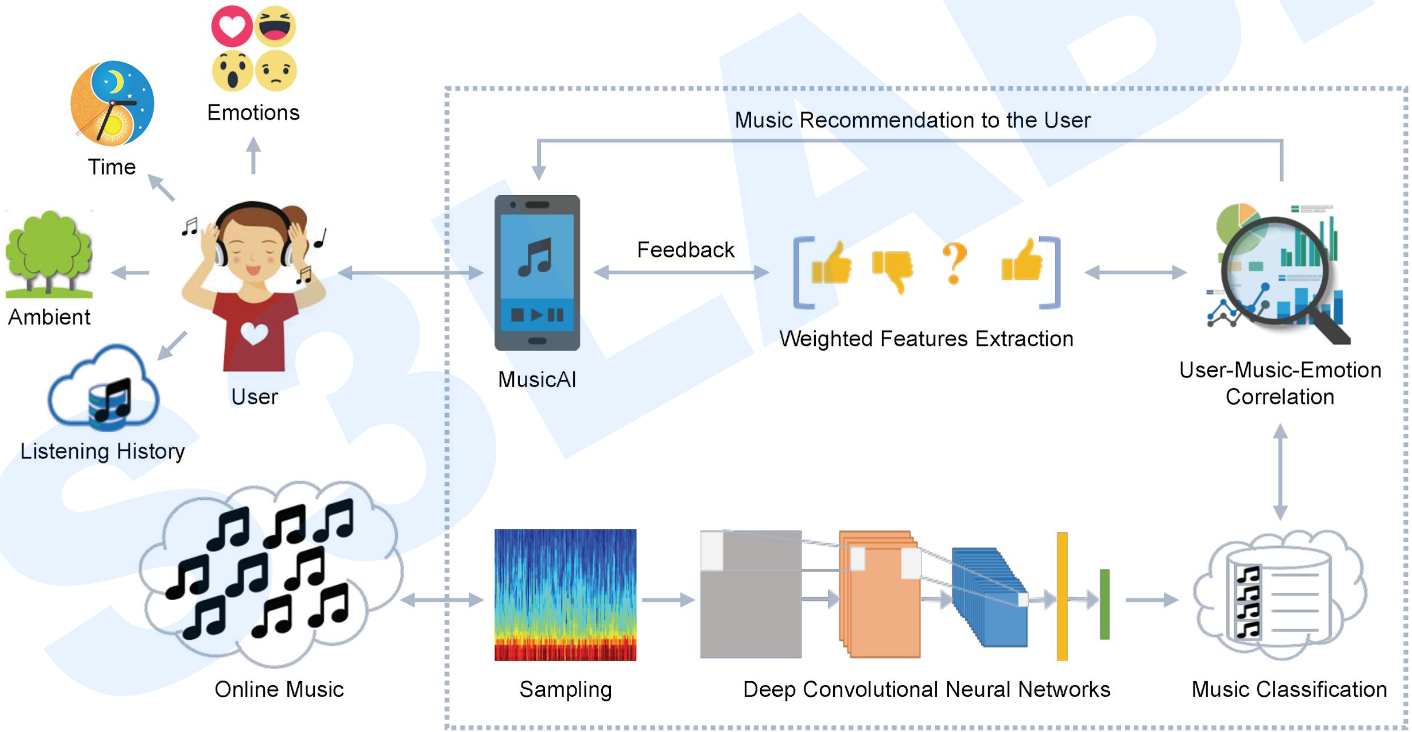




Recommended deals related to the context and user centric suggestions

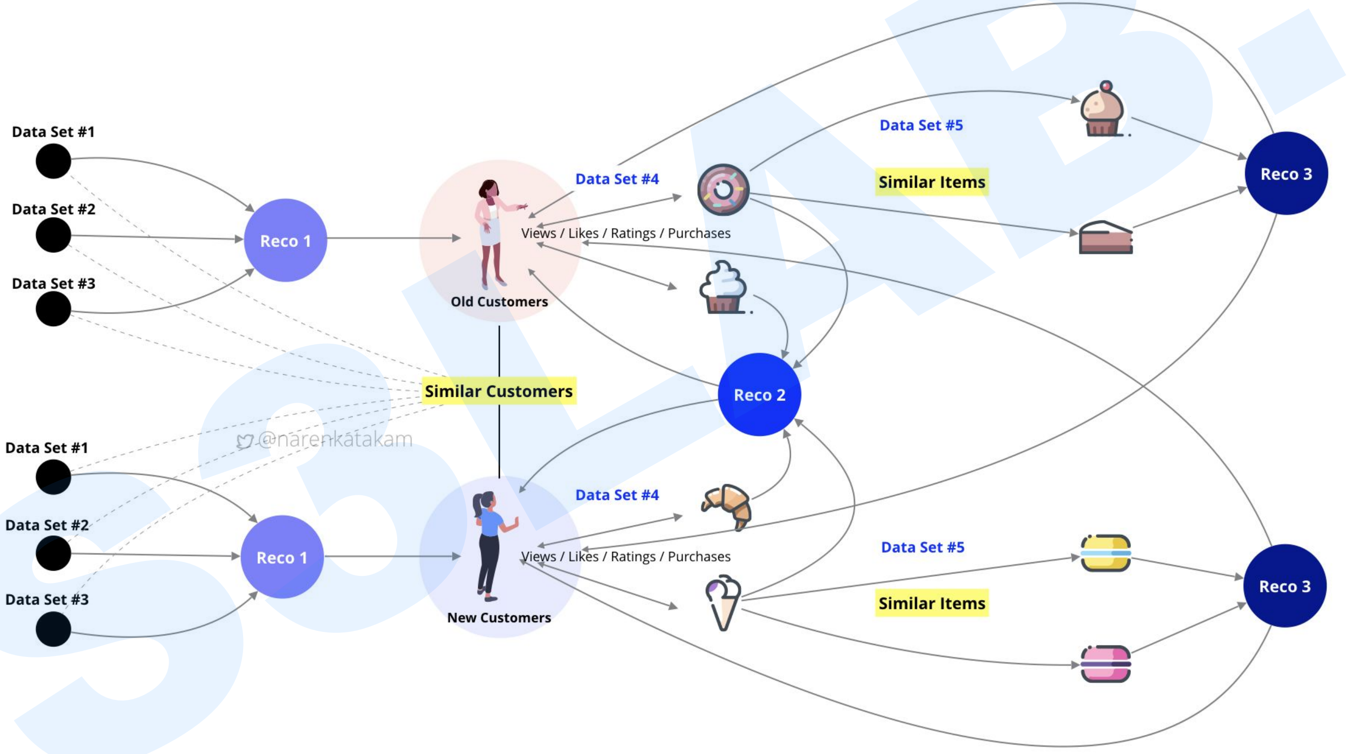
Burn and earn points clearly communicated for quick decision making

# Knowledge-based Filtering

## **Contextual - Aware** *Filtering*

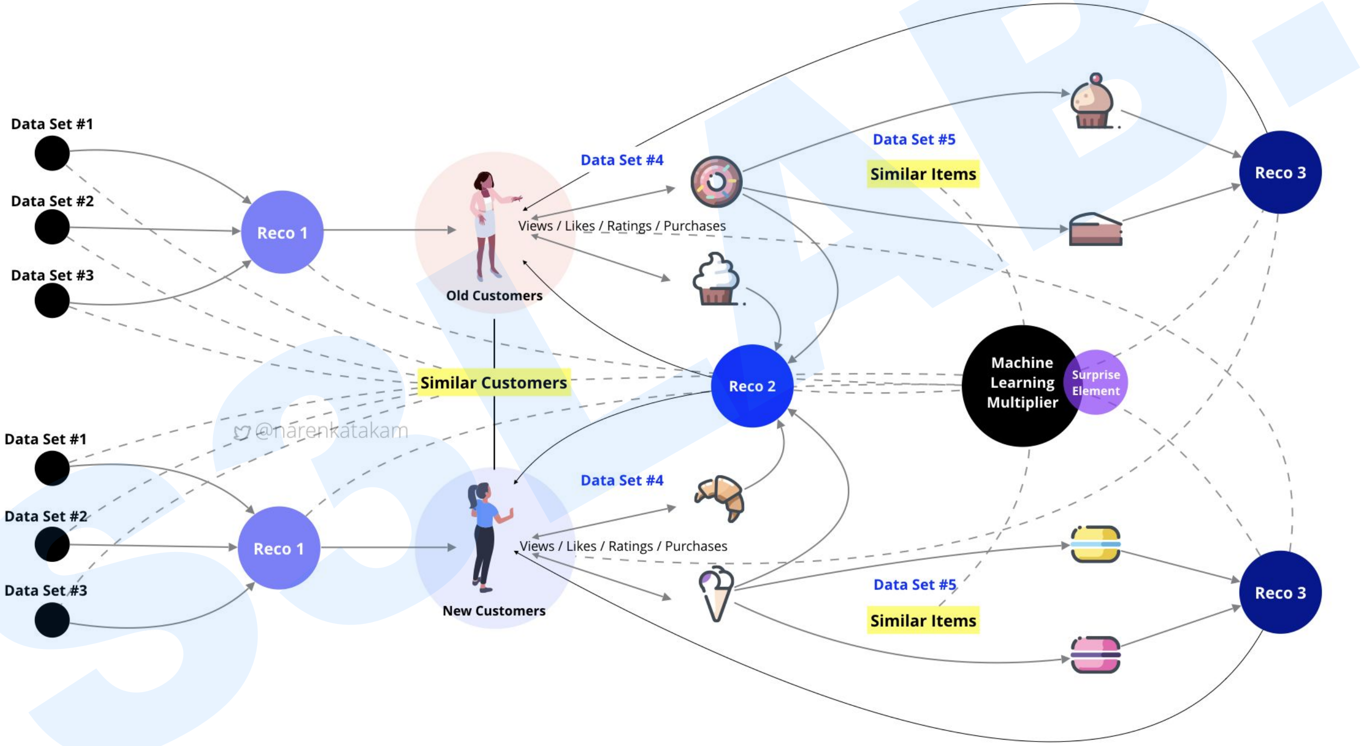


# Hybrid Filtering



# Hybrid Filtering

*With AI*



# Hybrid Filtering

| Method | Description |
|---|---|
| Weighted | Outputs from several techniques (in the form of scores or votes) are combined with different degrees of importance to offer final recommendations |
| Switching | Depending on situation, the system changes from one technique to another |
| Mixed | Recommendations from several techniques are presented at the same time |
| Feature combination | Features from different recommendation sources are combined as input to a single technique |
| Cascade | The output from one technique is used as input of another that refines the result |
| Feature augmentation | The output from one technique is used as input features to another |
| Meta-level | The model learned by one recommender is used as input to another |

# Hybrid Filtering

| | |
|---|---|
| **Hybrid Approaches** combines collaborative filtering and content-based filtering in order to use the pros of one to address the cons of the other. | |
| **Input:** uses user and item content features as well as usage data to benefit from both types of data. | |
| **Types**:<br>• Weighted<br>• Switching<br>• Mixed<br>• Feature Combination<br>• Cascade<br>• Feature augmentation<br>• Meta-level | |
| **Pros** | **Cons** |
| • Often outperforms CF and CB alone<br>• No cold start problem<br>• No popularity bias, can recommend items with rare features<br>• Can implement serendipity, diversity | • Can be a lot of work to get the right balance |

# Example about Book Recommender

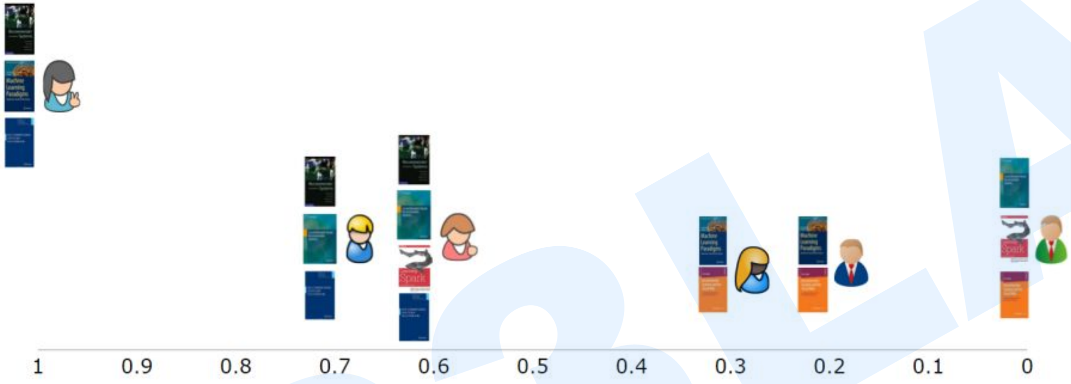
*Given Data*

- User preferences for books. All preferences are on a scale of 1-5, 5 being the most liked.
- Where a cell is empty, the user has not give a preference for the book.


| | | | | | |
|---|---|---|---|---|---|
| 4 | 3 | | | 5 | |
| 5 | | 4 | | 4 | |
| 4 | | 5 | 3 | 4 | |
| | 3 | | | | 5 |
| | 4 | | | | 4 |
| | | 2 | 4 | | 5 |

# Example about Book Recommender

*User-based Collaborative Filtering*




cosine similarity between and the other readers

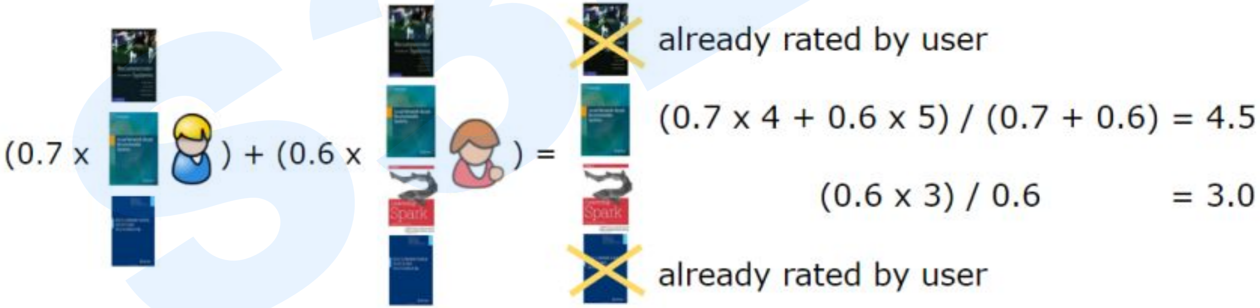
$$\text{Similarity}(p,q) = \cos\theta = \frac{p \cdot q}{\|p\|\|q\|} = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^{\,2}}\sqrt{\sum_{i=1}^{n} q_i^{\,2}}}$$

| | | | | | |
|---|---|---|---|---|---|
| 1.00 | 0.75 | 0.63 | 0.22 | 0.30 | 0.00 |
| 0.75 | 1.00 | 0.91 | 0.00 | 0.00 | 0.16 |
| 0.63 | 0.91 | 1.00 | 0.00 | 0.00 | 0.40 |
| 0.22 | 0.00 | 0.00 | 1.00 | 0.97 | 0.64 |
| 0.30 | 0.00 | 0.00 | 0.97 | 1.00 | 0.53 |
| 0.00 | 0.16 | 0.40 | 0.64 | 0.53 | 1.00 |

# Example about Book Recommender

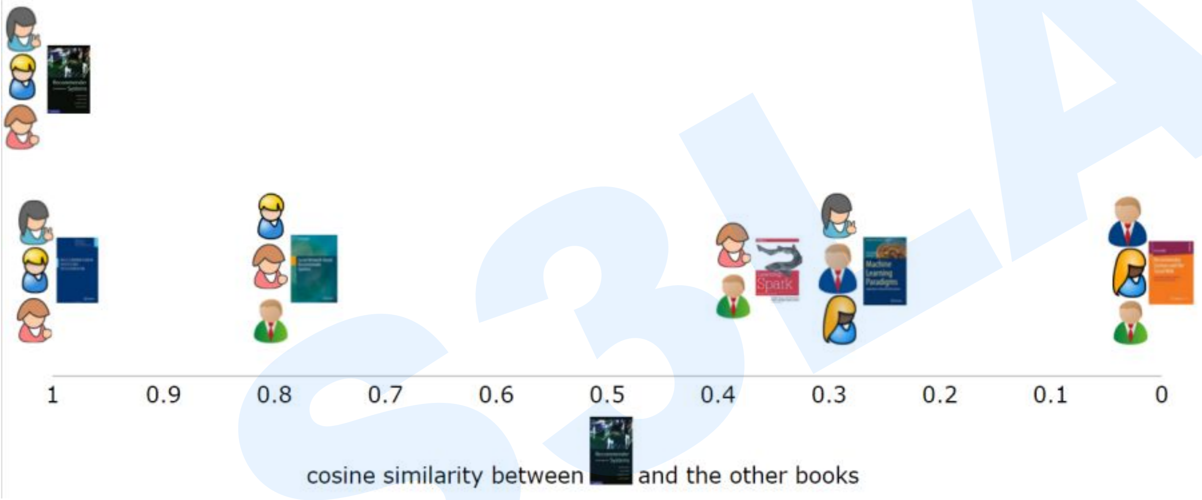
*User-based Collaborative Filtering*

- Generate some recommendation for first User.
  - Find top n users who are most similar to the first user (n=2)
  - Remove books that the user has already given preferences for, weight the books that the most similar users are reading, and sum them together



# Example about Book Recommender

*Item-based Collaborative Filtering*

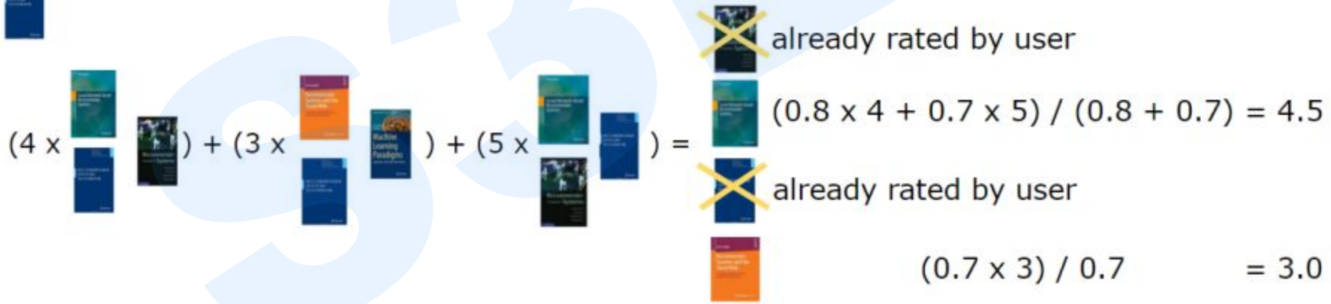



| | | | | | |
|---|---|---|---|---|---|
| 1.00 | 0.27 | 0.79 | 0.32 | 0.98 | 0.00 |
| 0.27 | 1.00 | 0.00 | 0.00 | 0.34 | 0.65 |
| 0.79 | 0.00 | 1.00 | 0.69 | 0.71 | 0.18 |
| 0.32 | 0.00 | 0.69 | 1.00 | 0.32 | 0.49 |
| 0.98 | 0.34 | 0.71 | 0.32 | 1.00 | 0.00 |
| 0.00 | 0.65 | 0.18 | 0.49 | 0.00 | 1.00 |

# Example about Book Recommender

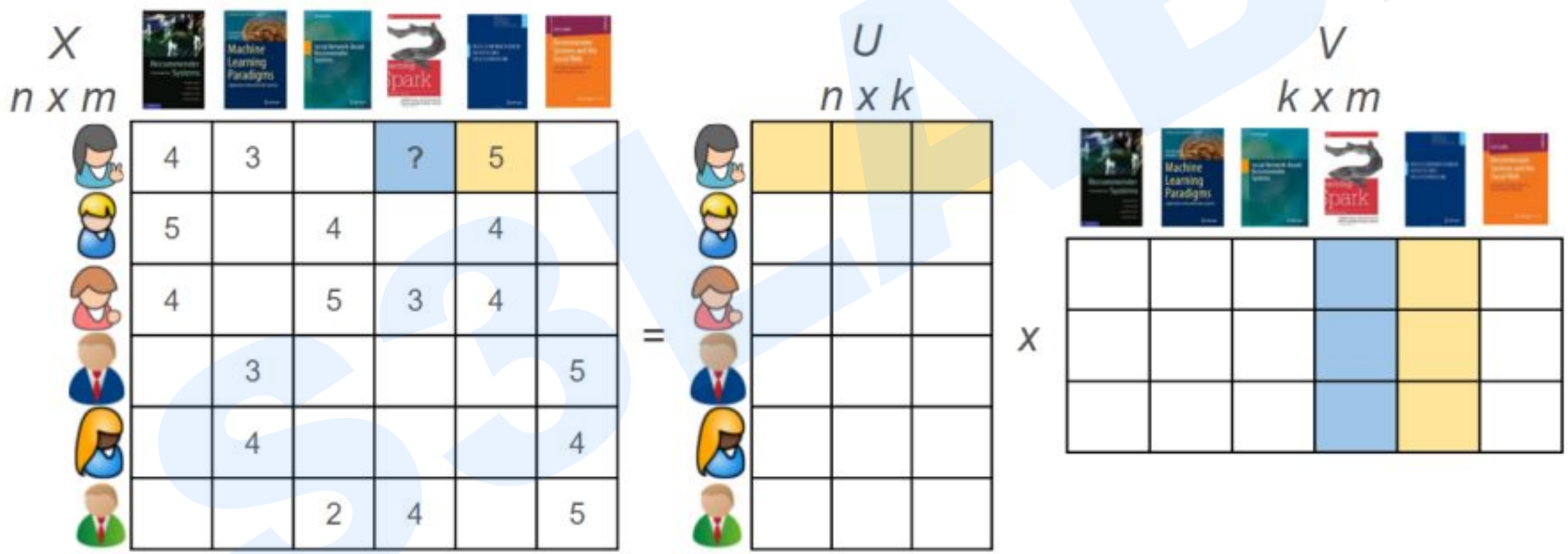
*Item-based Collaborative Filtering*

- Generate some recommendation for first User.
  - Find top n books which are most similar to each book of first User (n=2).
  - Remove books that the user has already given preferences for, weight the books that similar to referenced books, and sum them together



# Example about Book Recommender

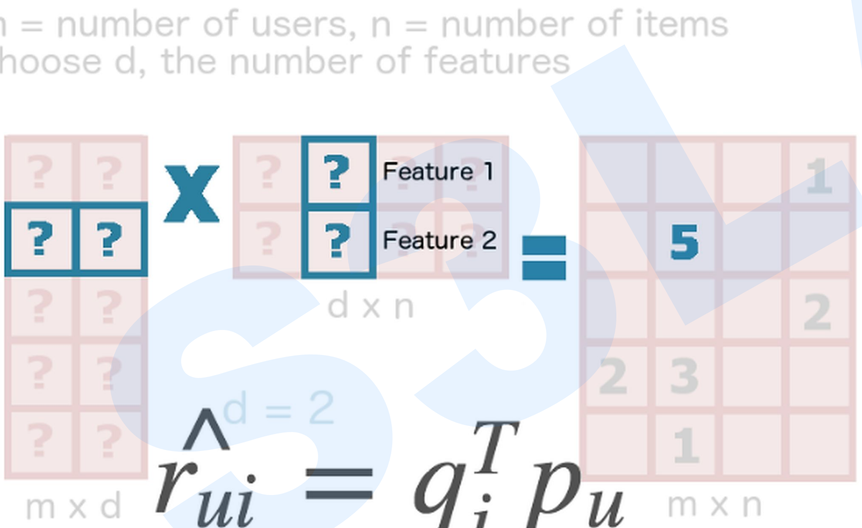
*Model-based Collaborative Filtering - Matrix Factorisation*



# Example about Book Recommender

*Model-based Collaborative Filtering - Matrix Factorisation*

## Matrix Factorization




$$\hat{r}_{ui} = q_i^T p_u$$

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2)$$

$$e_{ui} \overset{def}{=} r_{ui} - q_i^T p_u.$$

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i)$$

$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u)$$

http://nicolas-hug.com/blog/matrix_facto_1

# Example about Book Recommender

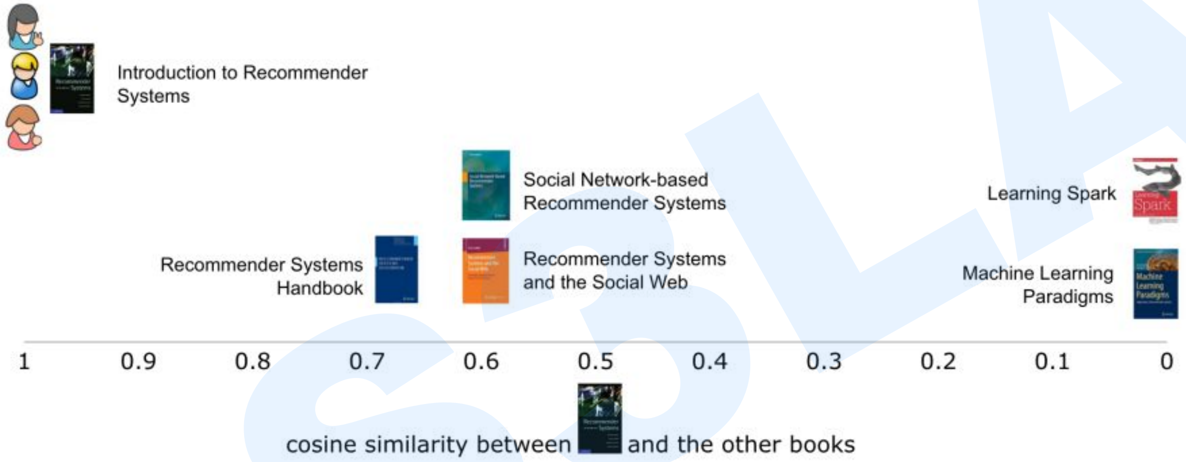
*Content-based*

| |
|---|
| Introduction to Recommender Systems |
| Machine learning Paradigms |
| Social Network-based Recommender Systems |
| Learning Spark |
| Recommender Systems Handbook |
| Recommender Systems and the Social Web |

| | | | | | | |
|---|---|---|---|---|---|---|
| introduction | 1 | | | | | |
| recommender | 1 | | 1 | | 1 | 1 |
| systems | 1 | | 1 | | 1 | 1 |
| machine | | 1 | | | | |
| learning | | 1 | | 1 | | |
| paradigms | | 1 | | | | |
| social | | | 1 | | | 1 |
| network-based | | | 1 | | | |
| spark | | | | 1 | | |
| handbook | | | | | 1 | |
| web | | | | | | 1 |

# Example about Book Recommender

*Content-based*




| | | | | | |
|---|---|---|---|---|---|
| 1.00 | 0.00 | 0.58 | 0.00 | 0.67 | 0.58 |
| 0.00 | 1.00 | 0.00 | 0.41 | 0.00 | 0.00 |
| 0.58 | 0.00 | 1.00 | 0.00 | 0.58 | 0.75 |
| 0.00 | 0.41 | 0.00 | 1.00 | 0.00 | 0.00 |
| 0.67 | 0.00 | 0.58 | 0.00 | 1.00 | 0.58 |
| 0.58 | 0.00 | 0.75 | 0.00 | 0.58 | 1.00 |

# Example about Book Recommender

*Content-based*

- Generate some recommendation for first User.
    - Find top n books which are most similar to each book of first User (n=2).
    - Remove books that the user has already given preferences for, weight the books that similar to referenced books, and sum them together

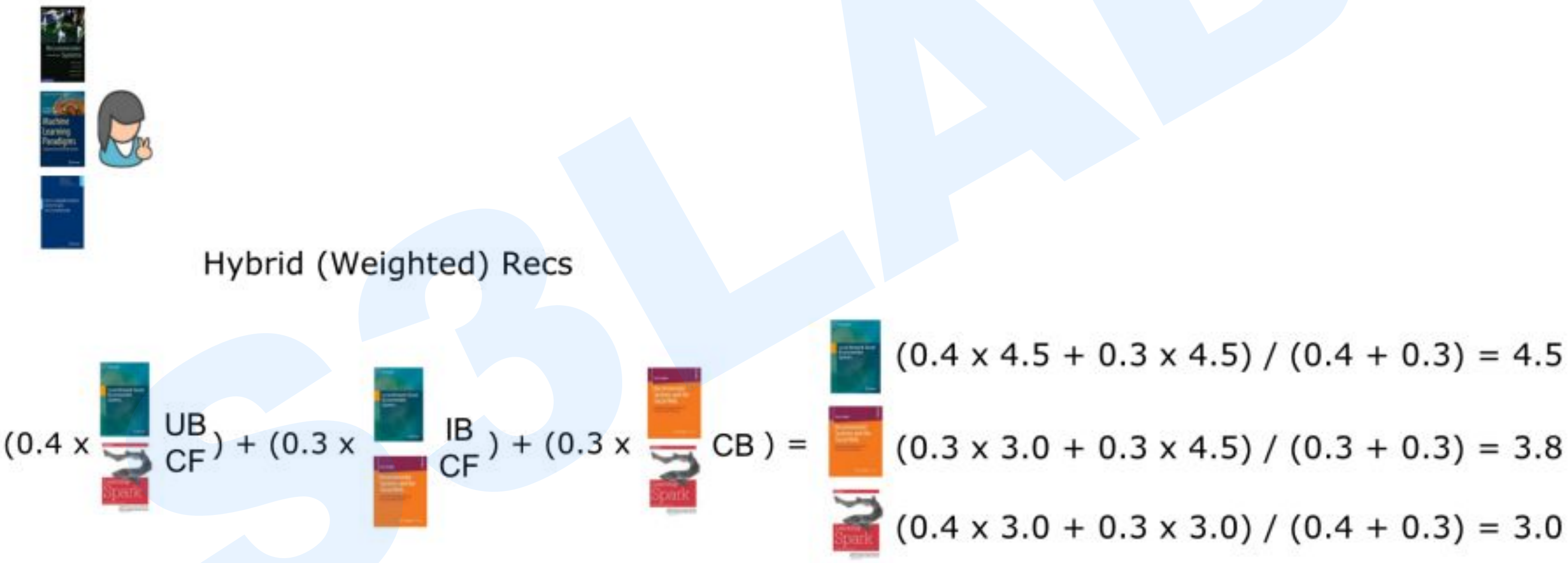
# Example about Book Recommender

*Hybrid Filtering*

- Generate some recommendation for first User.
  - We will use weighted method which combines the output from several techniques.
  - Given: 40% of the weight on user-based CF, 30% on item-based CF and 30% on content-based filtering.

# Example about Book Recommender

*Hybrid Filtering*



# System Model for Retail

# System Model for Retail

*User behaviors*

- These data can be collect manually or automatically:
    - Purchase Data
    - Likert
    - Search
    - Watching
    - Web browsing
    - Comments
    - Editor’s choice
    - ...

# System Model for Retail

*Goods Description*

- The content which system want to delivery to customer / audience.
- It can be:
  - A goods list of the store
  - Goods catalog, meta-data, description, ...
- ...

# System Model for Retail

*Context - Aware, Demographic, Social network data*

- These data can be collected manually or automatically.
    - The data which is collected from outside of core business system:
    - Location
    - Weather
    - Time (holiday, season, day of week, ...)
    - ...
- The others personal data of customer / audience:
    - Biometrics information: Gender, age, face ...
    - Identification information: clothes color, ...
    - ...

# Q & A

## Cảm ơn đã theo dõi

Chúng tôi hy vọng cùng nhau đi đến thành công.