Dining Place Recommendation System

Team 18

[OPTION 1] RS, Final Presentation

Sojung Noh 20223223 - Junho Kim 20180159 - Taeyoung Yun 20180424 - Min Kim 20190085

CONTENTS

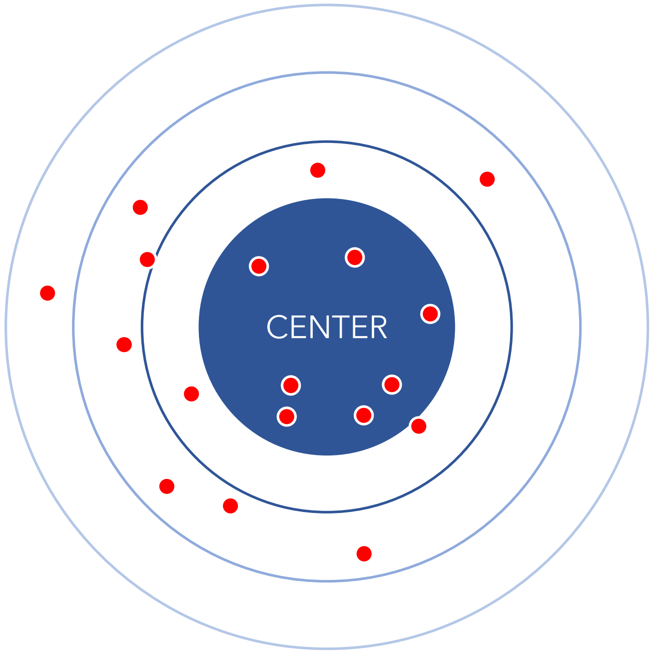
01 **Executive Summary of the Baseline Paper**

[OPTION 1] RS

~~**Project Provided by Class :**~~
~~Music Recommendation System using Metadata~~

Our Own Project Proposal :
Hot Place Recommendation System using Review Data

02 **Introduction**



Motivations

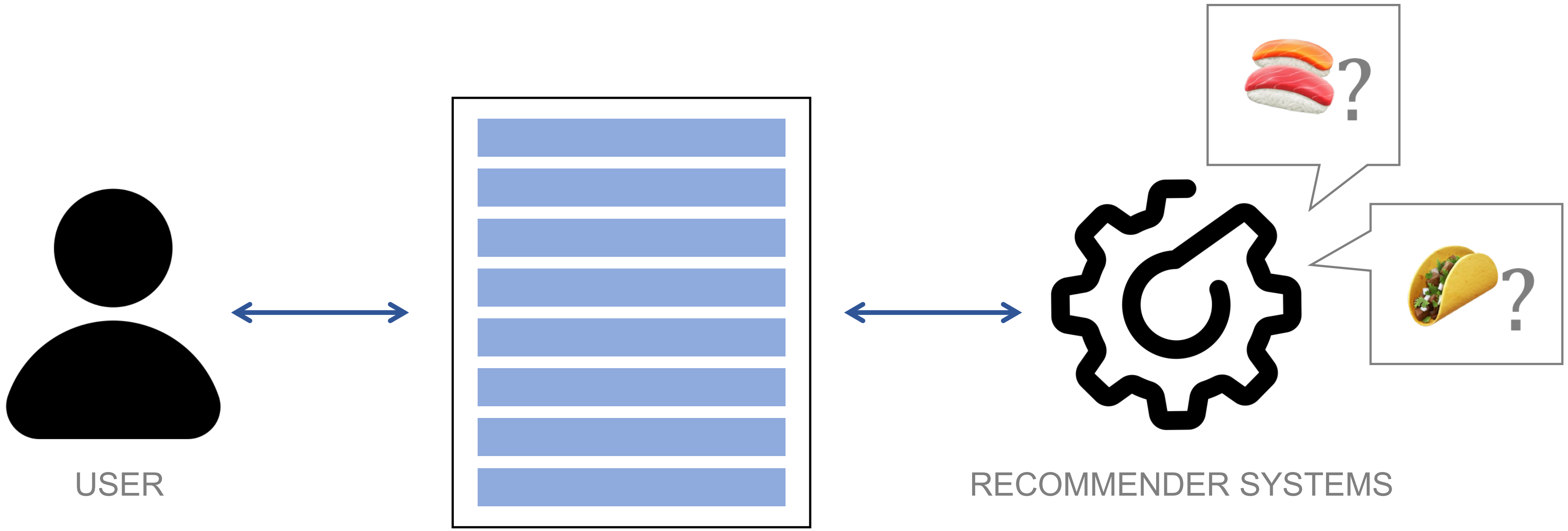
- Dajeon is notorious for its **monotonous-ness.** A.K.A. Nojam City (Which means no fun in Korean)
- Dajeon **lacks intriguing dining places**, which **millennials and Z generations crave for**.
- These places seldomly exist, and it's **difficult to find the right one that fits one's taste**.

02 **Introduction**

Objective

Build a **dining place recommendation system**,

optimized within the dataset (Daejeon Restaurants & Reviews)



03 **Preliminaries**

(i) Cosine Similarity

Cosine Similarity is an useful metric to compare similarity between vectors.

Cosine Similarity between two vectors can be calculated as below :

$$\text{cosine similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{||\mathbf{u}|| \cdot ||\mathbf{v}||}$$

After we represent items into vectors, we can calculate similarity between items using cosine similarity

It is already implemented in *sklearn* package:

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine-similarity.html

03 **Preliminaries**

(ii) Alternating Least Squares

Alternating Least Squares(ALS) is one of the most popular algorithm for Collaborative Filtering proposed by *Collaborative Filtering for Implicit Feedback Datasets*

ALS tries to minimize following objective:

$$\min_{x^*,y^*} \sum_{u,i} c_{ui}(p_{ui} - x_u^\intercal y_i)^2 + \lambda(\|x_i\|^2 + \|y_u\|^2)$$

where $$p_{ui} = \begin{cases} 0, & r_{ui} > 0 \\ 1, & r_{ui} = 0 \end{cases}, \quad c_{ui} = 1 + \alpha r_{ui}$$

Hu, Yifan, Yehuda Koren, and Chris Volinsky. "Collaborative filtering for implicit feedback datasets." *2008 Eighth IEEE international conference on data mining*. Ieee, 2008.

03 **Preliminaries**

(ii) Alternating Least Squares (Computational Perspective)

When we use ALS method, we need to update user(or item) vectors as below:

$$\mathbf{x}_u = (\mathbf{Y}^T \mathbf{C}^u \mathbf{Y} + \lambda \mathbf{I})^{-1} \mathbf{Y}^T \mathbf{C}^u p(u)$$

Unfortunately, calculating $\mathbf{Y}^T \mathbf{C}^u \mathbf{Y}$ is computationally expensive.

Therefore, the paper proposed a following trick to speed up:

$$\underbrace{\boxed{\mathbf{Y}^T \mathbf{C}^u \mathbf{Y}}}_{\text{(a)}} = \underbrace{\boxed{\mathbf{Y}^T \mathbf{Y}}}_{\text{(b)}} + \underbrace{\boxed{\mathbf{Y}^T (\mathbf{C}^u - I) \mathbf{Y}}}_{\text{(c)}}$$

Can be precomputed (independent of user)

Most of values in center matrix are zero (requires less computation)

It is already implemented in *Implicit* package: https://implicit.readthedocs.io/en/latest/index.html#

03 **Preliminaries**

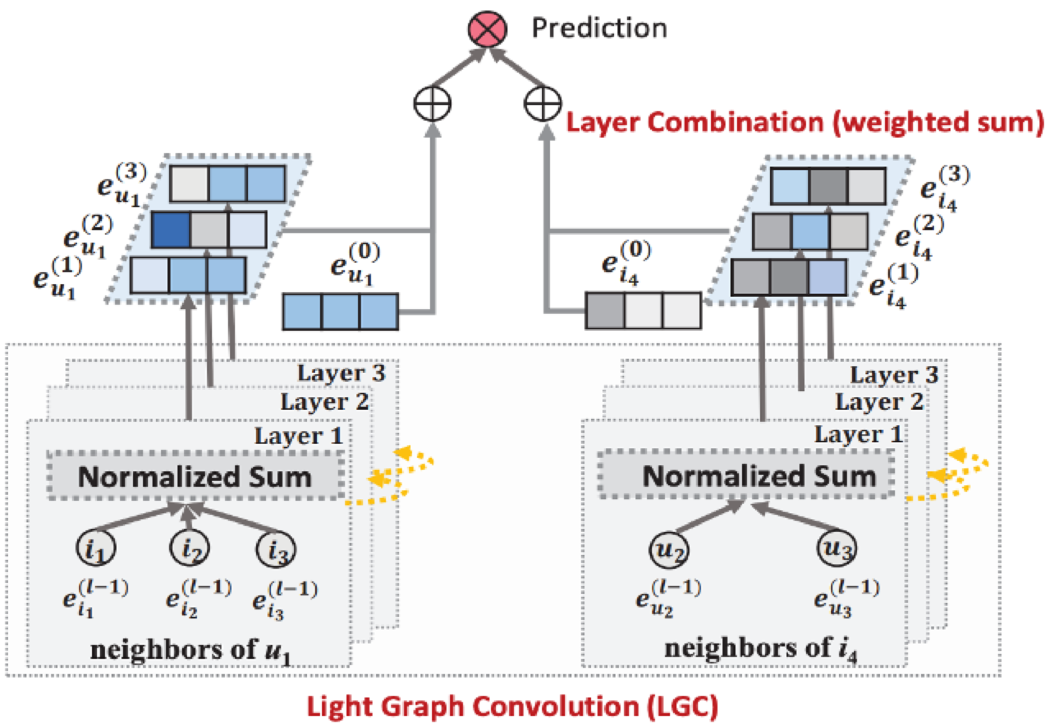
(iii) LightGCN

LightGCN is a powerful graph-based recommendation algorithm proposed by *Simplifying and Powering Graph Convolution Network for Recommendation*

LightGCN tries to minimize the following objective:

$$\mathcal{L} = -\sum_{u=1}^{M}\sum_{i\in\mathcal{N}_u}\sum_{j\notin\mathcal{N}_u} \ln\sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda\|\mathbf{E}\|^2$$

where $\hat{y}_{ui}$ is a predicted preference of user u on item i



[LightGCN model structure]

He, Xiangnan, et al. "Lightgcn: Simplifying and powering graph convolution network f or recommendation." *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 2020.

03 **Preliminaries**

(iii) LightGCN

LightGCN tries to minimize the following objective:

$$\mathcal{L} = -\sum_{u=1}^{M} \sum_{i \in \mathcal{N}_u} \sum_{j \notin \mathcal{N}_u} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \ + \ \lambda ||\mathbf{E}||^2$$

where $\hat{y}_{ui}$ is a predicted preference of user u on item i

LightGCN model is already implemented in pytorch-geomteric:

https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#torch-geometric.nn.models.LightGCN

04 **Solution**

Methods

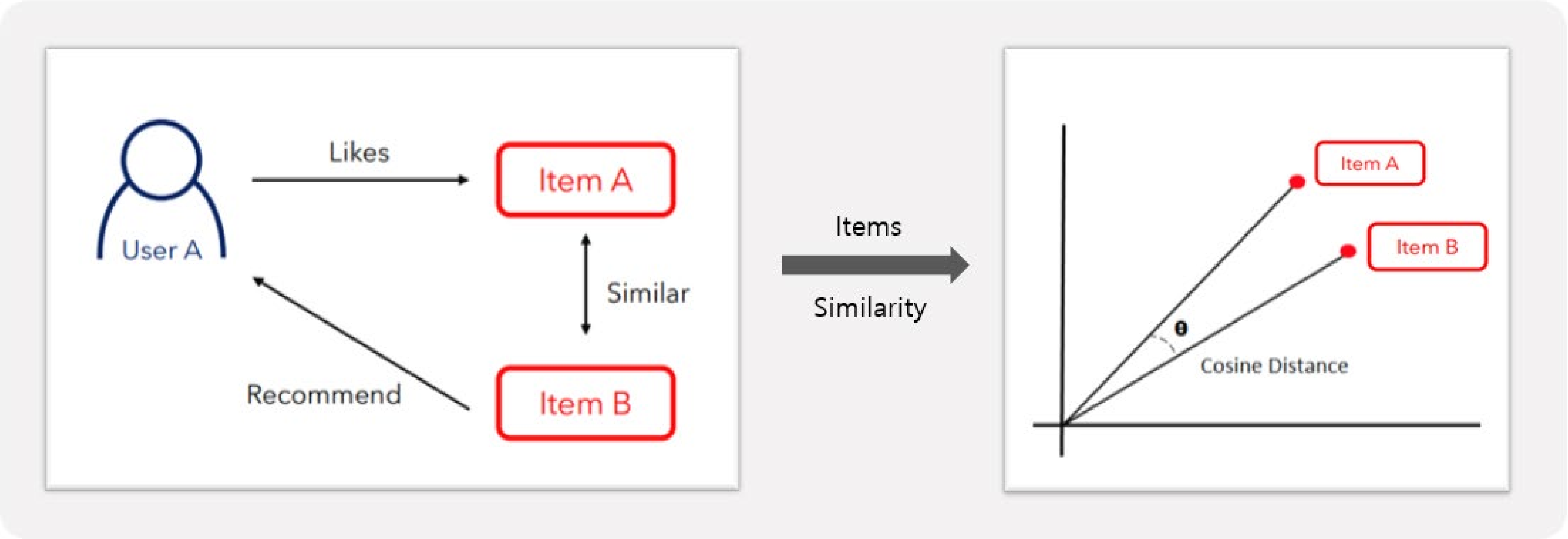
(i) Content-Based Filtering

(ii) Collaborative Filtering

(iii) Graph-Based Recommendation

04 **Solution**

(i) Content-Based Filtering



- **Recommend items that has similarity with user's likes**

The metadata of each item (dining place) is characterized and made into a item vector!

➡ Using cosine similarity, extract items with high similarity to user's favorite items from the item list.

04 **Solution**

(ii) Collaborative Filtering

	Item 1	Item 2	...	Item n
User 1	2	3	...	?
User 2	?	?	...	5
...	...	...	...	...
User n	1	?	...	4

? Alternating Least Squares(ALS) Algorithm!

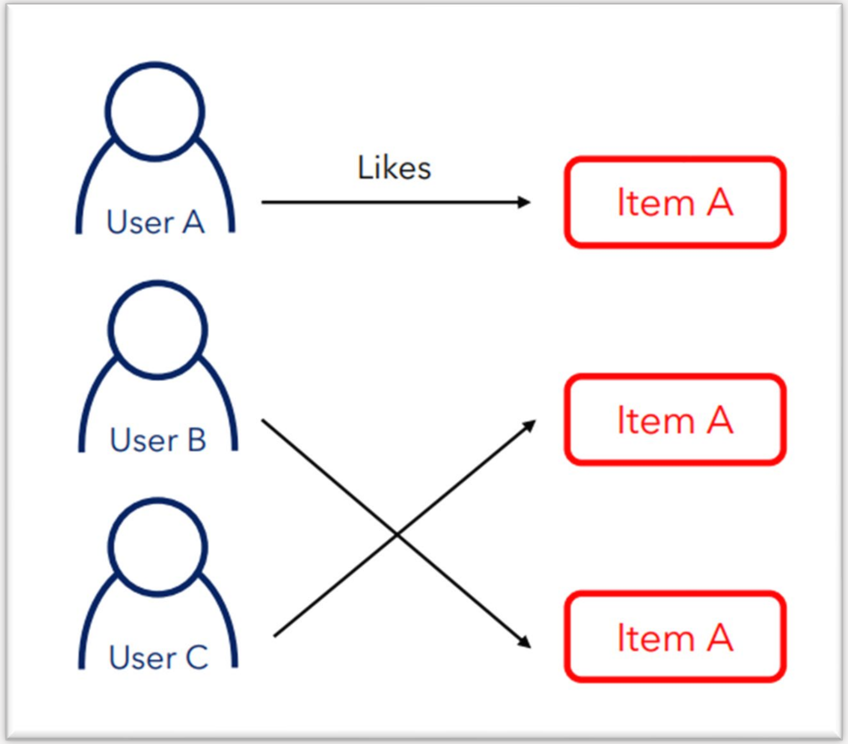
- **Recommend items that other similar users like**

Recommned items with high ratings by predicting ratings for unseen items!

➡ Using ALS, optimize the objective function that minimizes the difference between the predicted and actual.

04 **Solution**

(iii) Graph-Based Recommendation



- **Represent User-Item Interaction by Graph**

  - Predict user's preferences on unseen items using generated embedding for users and items.
  - Using light GCN, optimize the objective function of predicted preference.

05 **Experiment** Project Flow

Data gathering

Data crawling

Review, rating …

Data Processing

Data clustering

extract and organize (One-hot Encoding, Normalization) meaningful data.

Modeling

Implement **content based**, **collaborative filtering**, **graph based model** for diverse situation.

Food Recommendation

Final result

Input

User Information

Output

Recommended restaurant and cafe

Evaluation

User survey

Evaluate performance by conducting a satisfaction survey on the recommended result for each user.

05 **Experiment** Dataset

In this project, we work on top 500 restaurants of Daejeon in Kakao Map.

- **Content-Based Filtering** (Size ≈ 500)

Res_name	Rating	Rev_cnt	Addr	Category

  - **Rev_cnt:** Number of review (can compare restaurants with same rating)
  - **Addr:** Location of restaurant (five districts of Daejeon) → One Hot Encoding
  - **Category:** Sort of main menu

- **Collaborative Filtering** (Size ≈ 7500)

Res_name	User_name	Rating

- **Graph-Based Recommendation**

  Use the above two datasets

05 **Experiment** Dataset

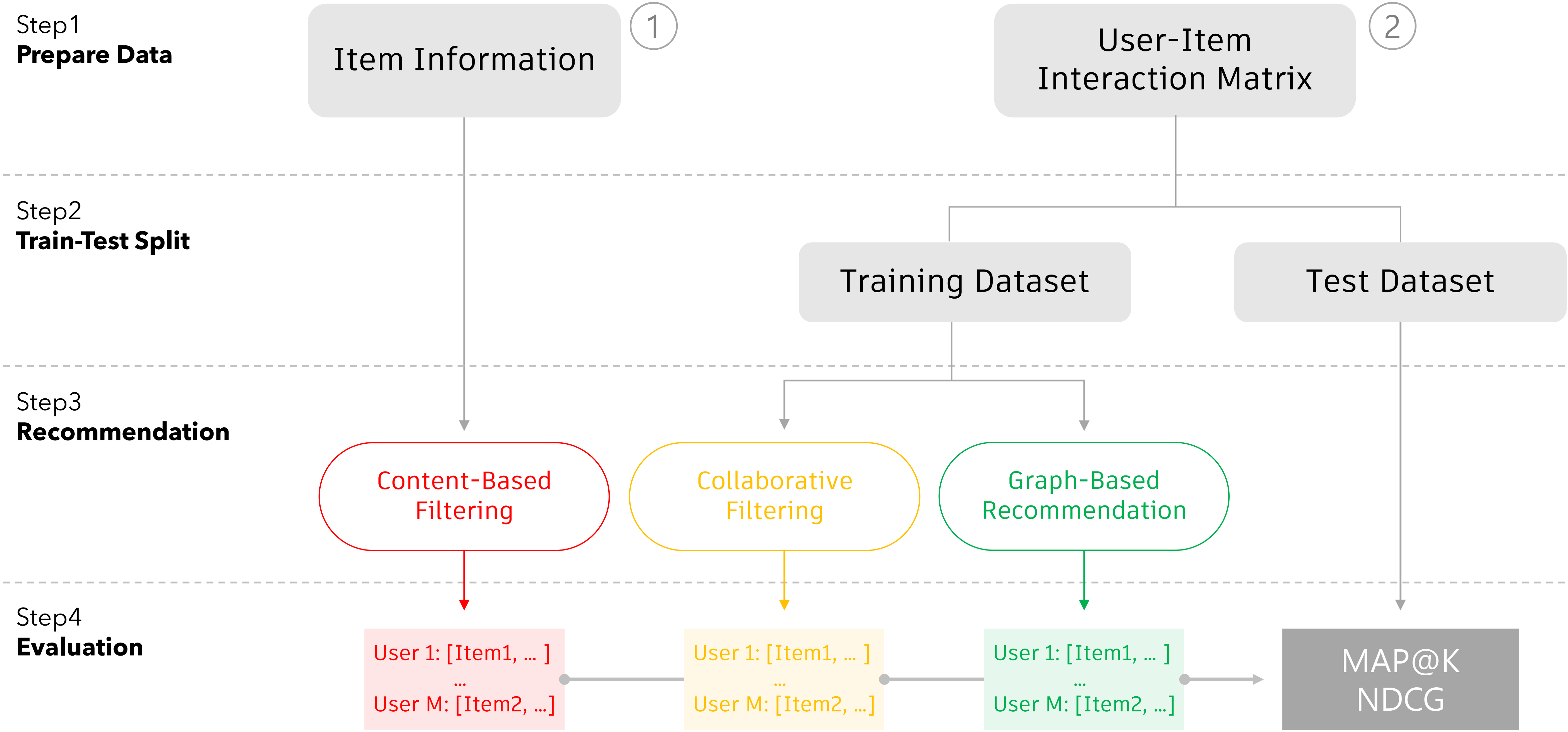
Content-Based Filtering (Size ≈ 500)

res_name	rating	rev_cnt	addr	category
성심당 본점	4.3	428	동구	제과,베이커리
성심당 DCC점	4.3	77	유성구	제과,베이커리
원조태평소국밥	4	120	유성구	국밥
팡시온카페	3.1	62	유성구	카페
오씨칼국수	3.9	109	유성구	국수
상무초밥 유성점	3.5	71	유성구	초밥,롤
광천식당	3.1	113	동구	불고기,두루치기
성심당 대전역점	3.9	65	유성구	제과,베이커리
커피인터뷰 궁동점	3.6	54	유성구	카페
더리스	3	44	유성구	양식
원조태평소국밥 둔산점	3.8	53	서구	국밥
온천손칼국수쭈꾸미	3.7	63	유성구	국수
치앙마이방콕	3.3	64	유성구	태국음식
오문창순대국밥	3.8	123	유성구	순대
사리원 본점	3.4	46	서구	냉면
진로집	3.1	71	동구	불고기,두루치기
대선칼국수	3.4	70	서구	국수
베스타뷔페	4.1	57	서구	뷔페
요우란	4.3	40	동구	일식
오씨칼국수	3.7	49	유성구	국수
청송한우타운	4.7	146	유성구	육류,고기
태화장	3	43	유성구	중화요리
비래키키	3.2	13	유성구	카페

Collaborative Filtering (Size ≈ 7500)

res_name	user_name	rating
진로집	눈치챙겨	5점
진로집	이형철	5점
진로집	그래요	5점
진로집	맛좋다	5점
진로집	소영	5점
진로집	김정민	5점
진로집	J	4점
진로집	.	5점
진로집	.	5점
진로집	김영춘	4점
진로집	김정민	5점
진로집	J	4점
진로집	.	5점
진로집	.	5점
진로집	김영춘	4점
진로집	가스&연어덕	5점
진로집	젤리발바닥	1점
진로집	_ni	5점
진로집	JD	5점
진로집	HS KIM	1점
진로집	블랙	4점
진로집	콩	5점
진로집	[illegible]	5점

05 **Experiment** Outline



06 **Results** Result analysis & Evaluation

MAP@K

Evaluation metric by calculating *precision* of the recommendation.

$$Precision@K = \frac{True\ Positive}{Predictied\ Positive}$$

$$AP@K = \frac{1}{m}\sum_{i=1}^{k} Precision@i$$

$$\rightarrow MAP@K = \frac{1}{|U|}\sum_{u=1}^{|U|}(AP@K)_u$$

NDCG

Evaluation metric by calculating *relevance* of the recommendation.

$$CG_k = \sum_{i=1}^{K} rel_i$$

$$DCG_k = \sum_{i=1}^{K} \frac{rel_i}{\log(i+1)}$$

$$IDCG_k = \sum_{i=1}^{K} \frac{rel_i^{opt}}{\log(i+1)}$$

$$\rightarrow NDCG_k = \frac{DCG}{NDCG}$$

06 **Results**

For random user A :

RECOMMENDATION MODELS

Rank	True Like
1	파운드
2	무이
3	그린베이커리
4	동방명주 동천홍2호점
5	예담추어정
6	설천순대국밥 유성직영점
7	수정삼겹살
8	성심당 DCC점
9	카페시은우
10	무공돈까스 대전둔산점

Content Based	Collaborative Filtering	Graph-Based
화이트무스	복수분식 본점	오씨칼국수 도룡점
텀즈업브로	구들마루	디블루메
착한참치 본점	개천식당	도레미아구찜
연탄구이	복사꽃피는집 대전점	성심당 대전역점
대선칼국수	스바라시라멘 본점	사리원 본점
놀부네집	수통골감나무집 본점	오씨칼국수
맥도날드 대전유성DT점	화이트무스	구름식당
수통골장수오리	수통골장수오리	우사미 대전본점
드르쿰다 나인스테이	하레하레 도안점	꽁뚜식당
설해돈 둔산본점	더함뜰	임프레션커피컴퍼니

06 **Results**

Model Performance with MAP@K & NDCG

Any better than random?

Evaluation	Content-Based	Collaborative	Graph-Based	Random
MAP@K	0.0031	0.0184	0.0151	0.0047
NDCG	0.0012	0.0061	0.0056	0.0023

Yes for *Collaborative* & *Graph-based*

07 **Discussion**

Major problem of our experiment : *The Sparsity of the Dataset*

Our Dataset		**Movielens-1M**
Only 5% of users rate at least 5 items	<	All users rate to at least 20 items
Data Sparsity 0.3%		Data Sparsity 4.4%

07 **Discussion**

- **Limitation of data crawling from 4 websites**

  When the matrix of res_name and user_name is made, the matrix is too sparse to progress learning.

  Reason of choosing Kakaomap

  Most of people in Korea use Kakao-related applications.

  Thus, Kakaomap is considered as the most popular one in this project.

- **Difficulty of presenting price**

  The price info should be gathered manually for 500 restaurants. Thus, price info is excluded in this project.

Conclusion

Not fully successful, but meaningful project:

- Even in a very low rate, Collaborative filtering & Graph-based recommendations exhibit far better performance than Content based.
- Recognized the importance of mass data collection to overcome sparsity.
- In order to construct targeted recommendation for KAIST students, ratings survey is necessary.

😀

Thank you!