The Right Wine for You

Finding the best RecSys model
for data sparsity improvement

Team wine T-stem
11th 한은결
12th 김민규
13th 김선기 박세현 정주은

TABLE OF CONTENTS

CHAPTER.1

Intro : Topic Selection

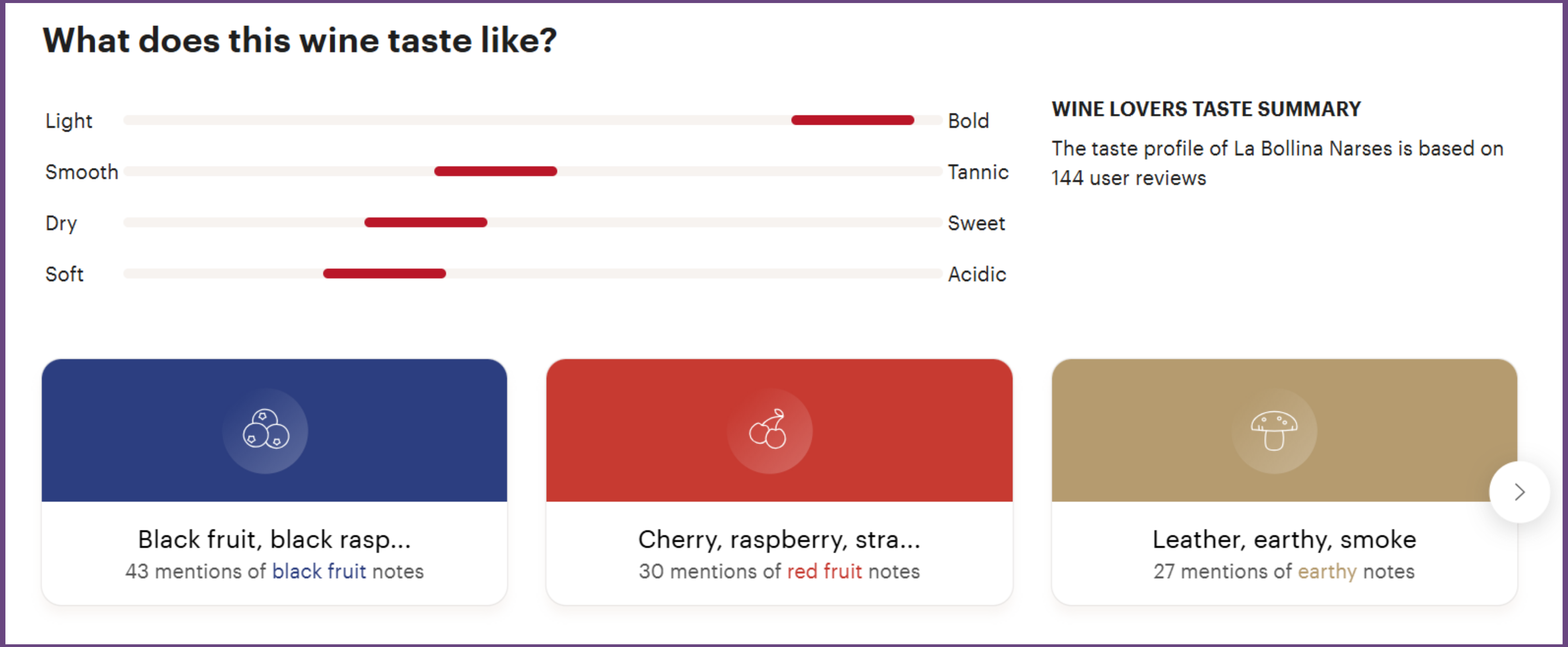
Why did we choose this topic ?

TOPIC SELECTION RATIONALE

Why the Wine Recommendation System?

wine : Flavor, Grapes, Region, Winery ...

-> too many conditions to choose



TOPIC SELECTION RATIONALE

💡 Optimal topic for RecSys based on dataset

CHAPTER.2

Dataset Introduction

What is our dataset?

Global Wine Site : Vivino (https://www.vivino.com)

1. Crawling top 500 reviewers' ratings and reviews in US

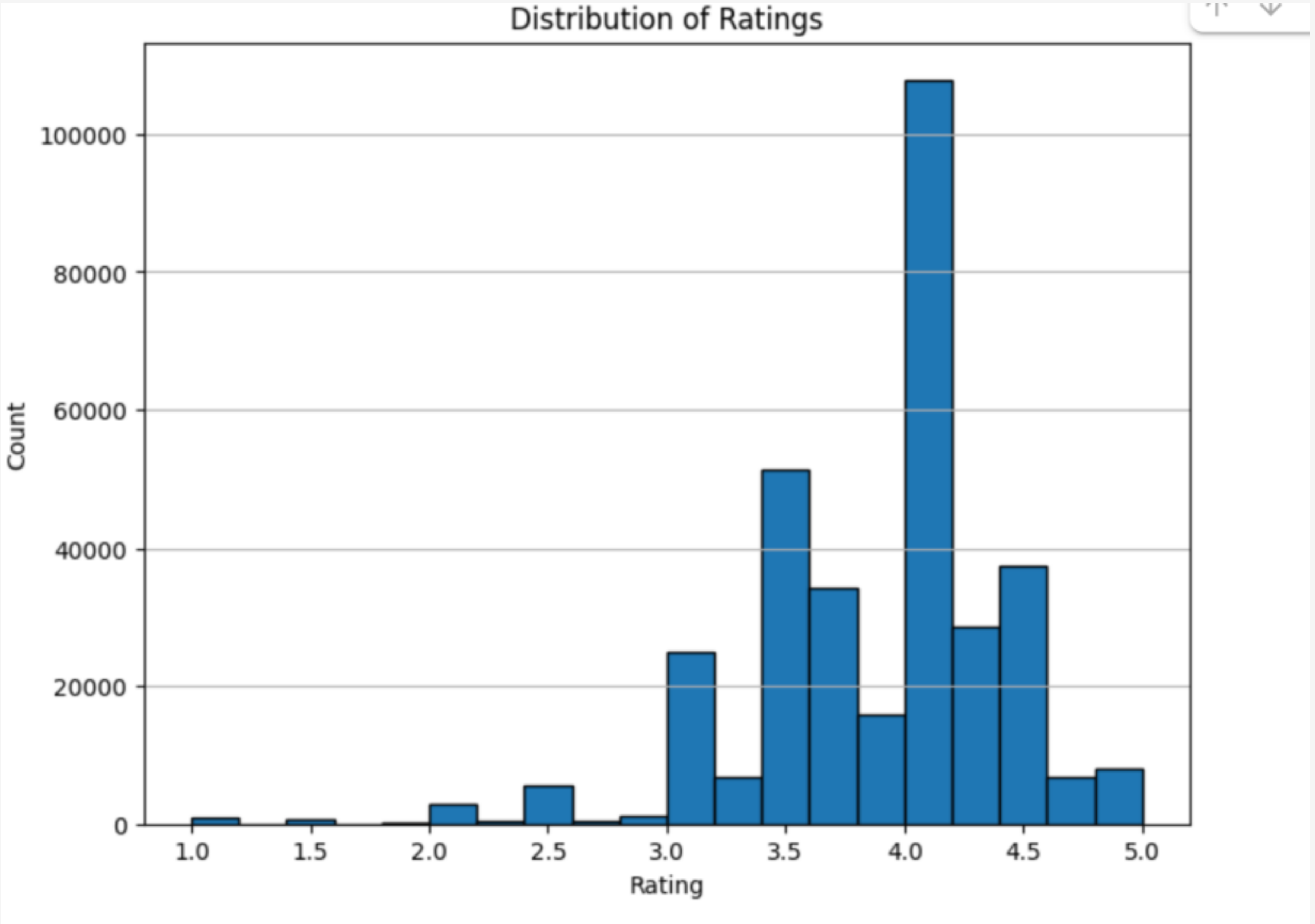
2. And 299,646 Wine items

user	The Holy Trinity Red Blend 2018	Pinot Noir 2012	Pomerol 2019	Brut Rosé Champagne N.V.	Topography 2014
James Pilachowski		4.0		4.5	
Alexander Ross		2.0	3.5	3.5	
-"Paul Neira"-			4.5	4.5	
Tom Colby					
Cs Runner				3.5	
Ming				4.0	4.1

CHAPTER.3

Problems of the Dataset

PREPROCESS



Log Transformation

*10 scaling

Problem : Sparsity

user

466

Too few interaction

wine

300K

Sparsity!

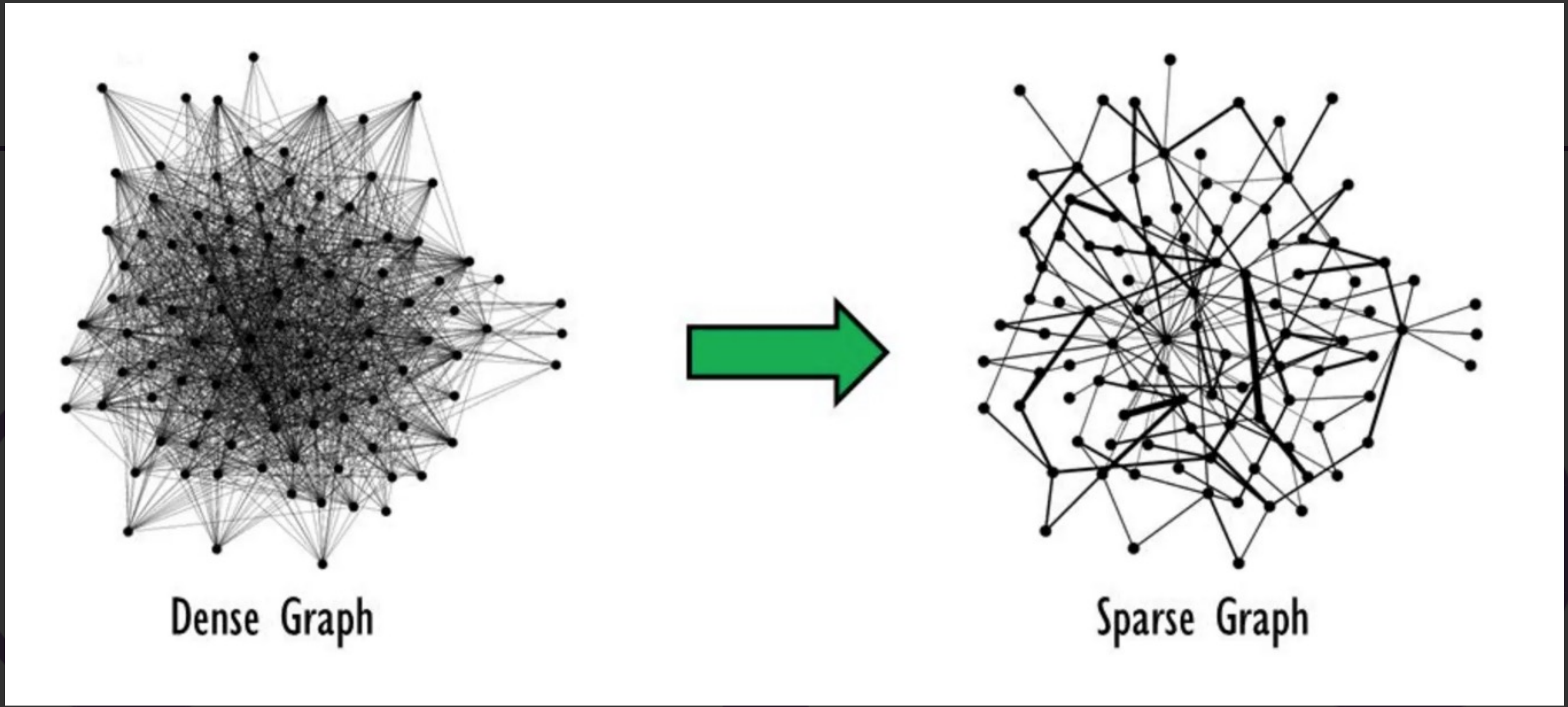


Table 2: Statistics of the experimented data.

Dataset	User #	Item #	Interaction #	Density
Gowalla	29, 858	40, 981	1, 027, 370	0.00084
Yelp2018	31, 668	38, 048	1, 561, 406	0.00130
Amazon-Book	52, 643	91, 599	2, 984, 108	0.00062

Exclude minor wines with very few reviews

Exclude wines with ≤1 review	After exclude
≤2	36140
≤3	19190
≤4	11192

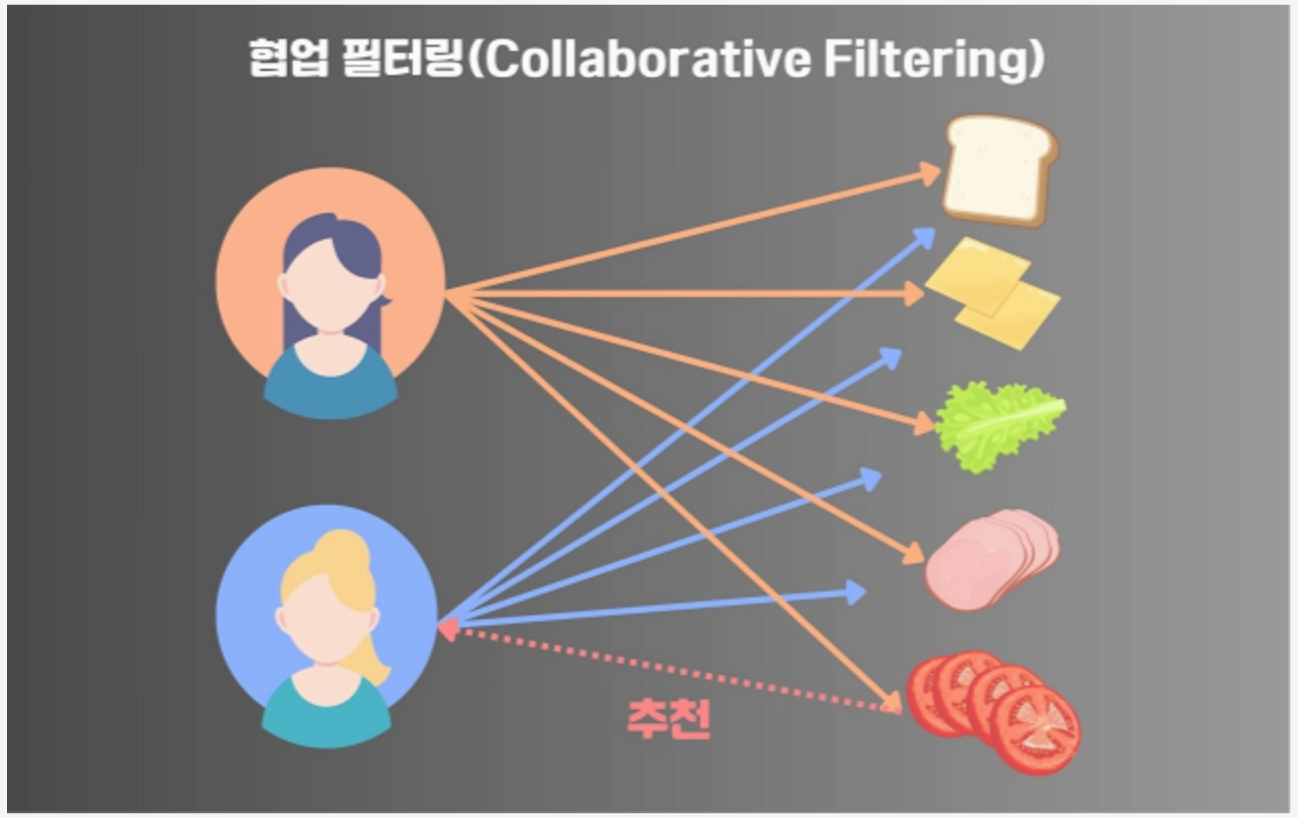
CHAPTER.4

Modeling

How did we solve our dataset's problem?

To Make Wine More Accessible for Everyone

1. Collaborative Filtering for existing wine reviewers

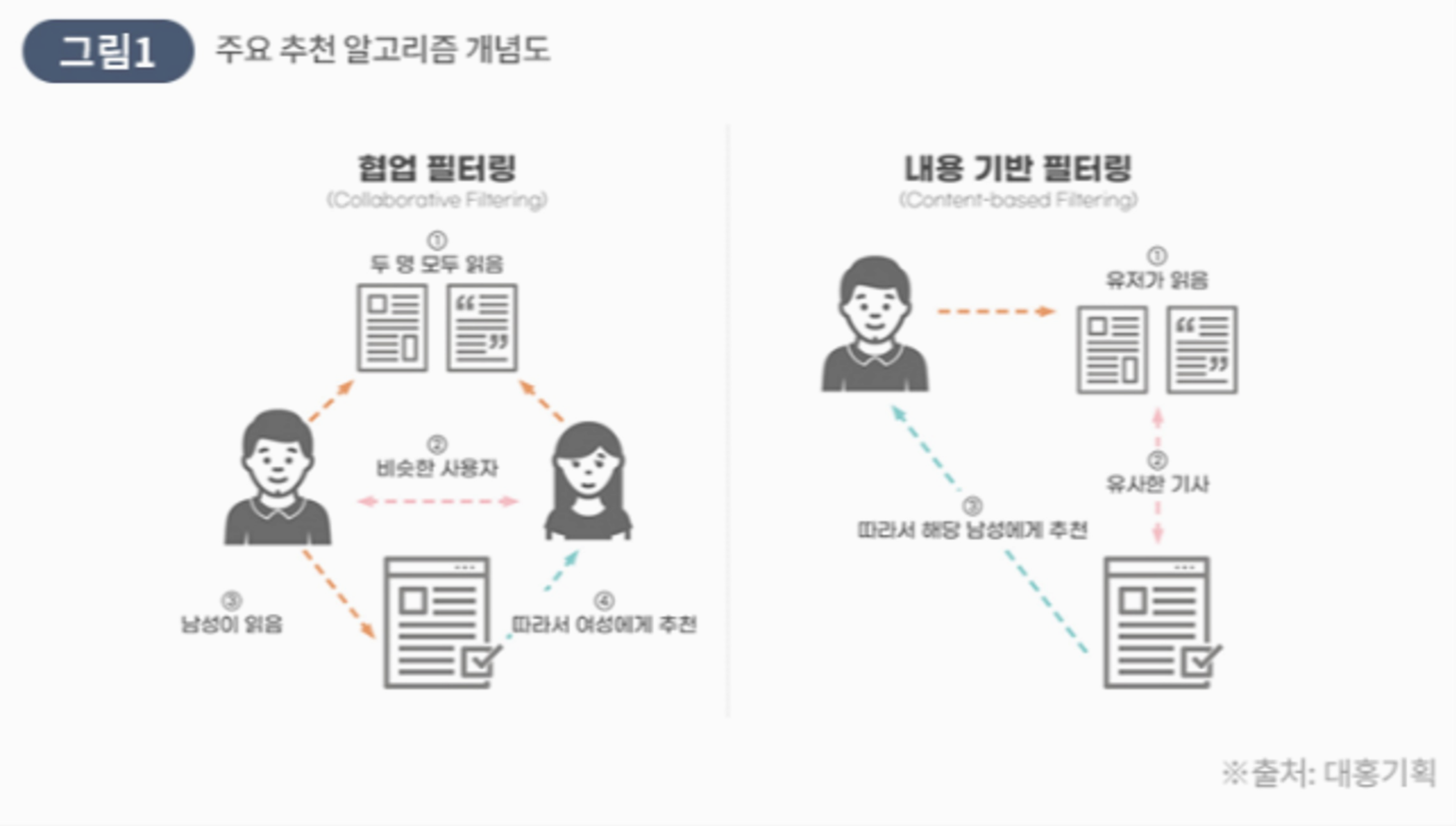


- Problem: Wine has diverse attributes and a complex classification system, making it difficult for beginners to access.
- How to Solve: We aim to provide personalized wine recommendations by learning individual preferences based on user rating data, using a collaborative filtering–based recommendation system.
- Users can select wines suited to their tastes without having to interpret complex information, thereby lowering the entry barrier to wine consumption.

To Make Wine More Accessible for Everyone

2. Hybrid RecSys for existing and "NEW" wine customers

: Hybrid = Collaborative filtering + Content-based filtering



- Instead of recommending wines based on simple popularity, we use **collaborative filtering** to provide **personalized recommendations**.
- By leveraging user rating data on wines, we recommend wines that are preferred by users with similar tastes.

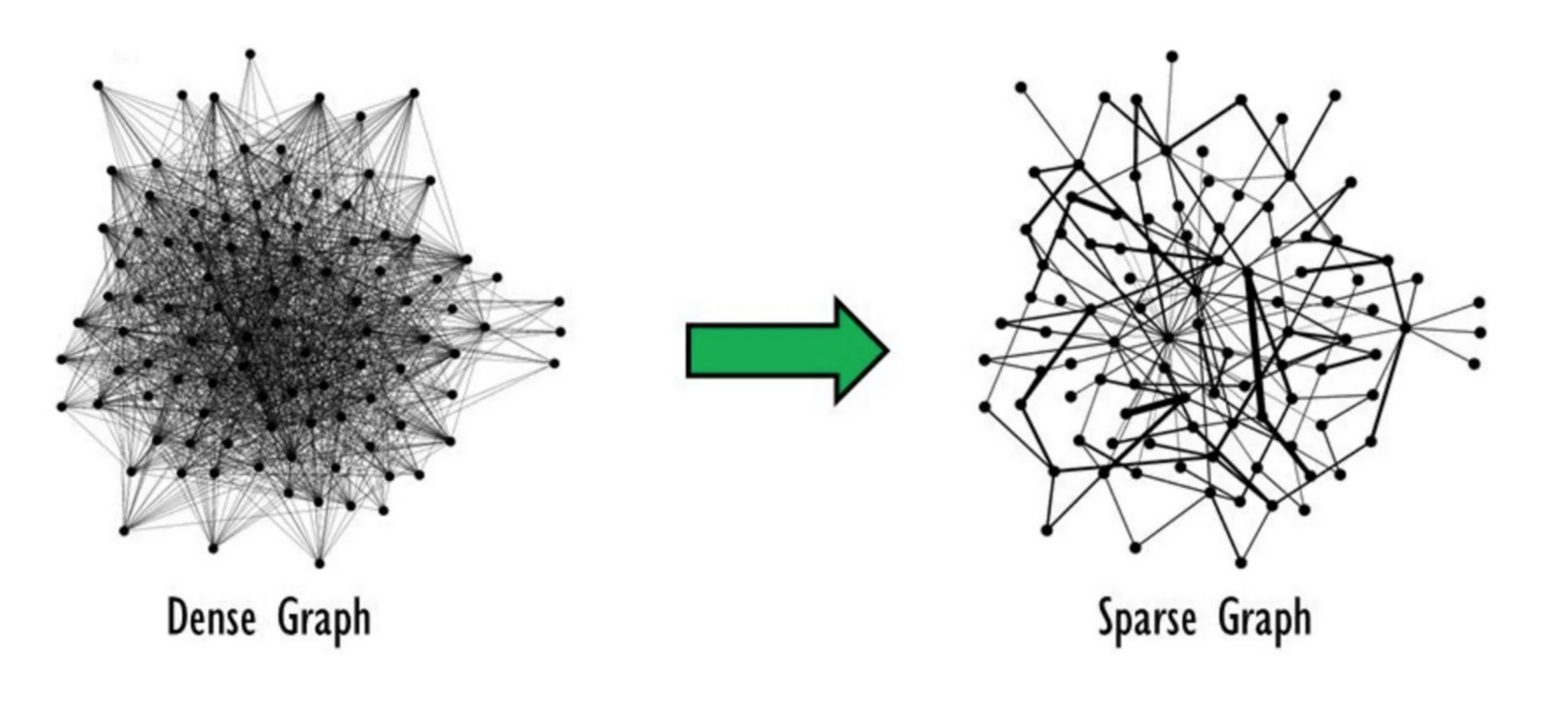
*Content-based filtering: Recommends based on the attributes of the wine itself

*Collaborative filtering: A user-centered approach that becomes more accurate as more data accumulates

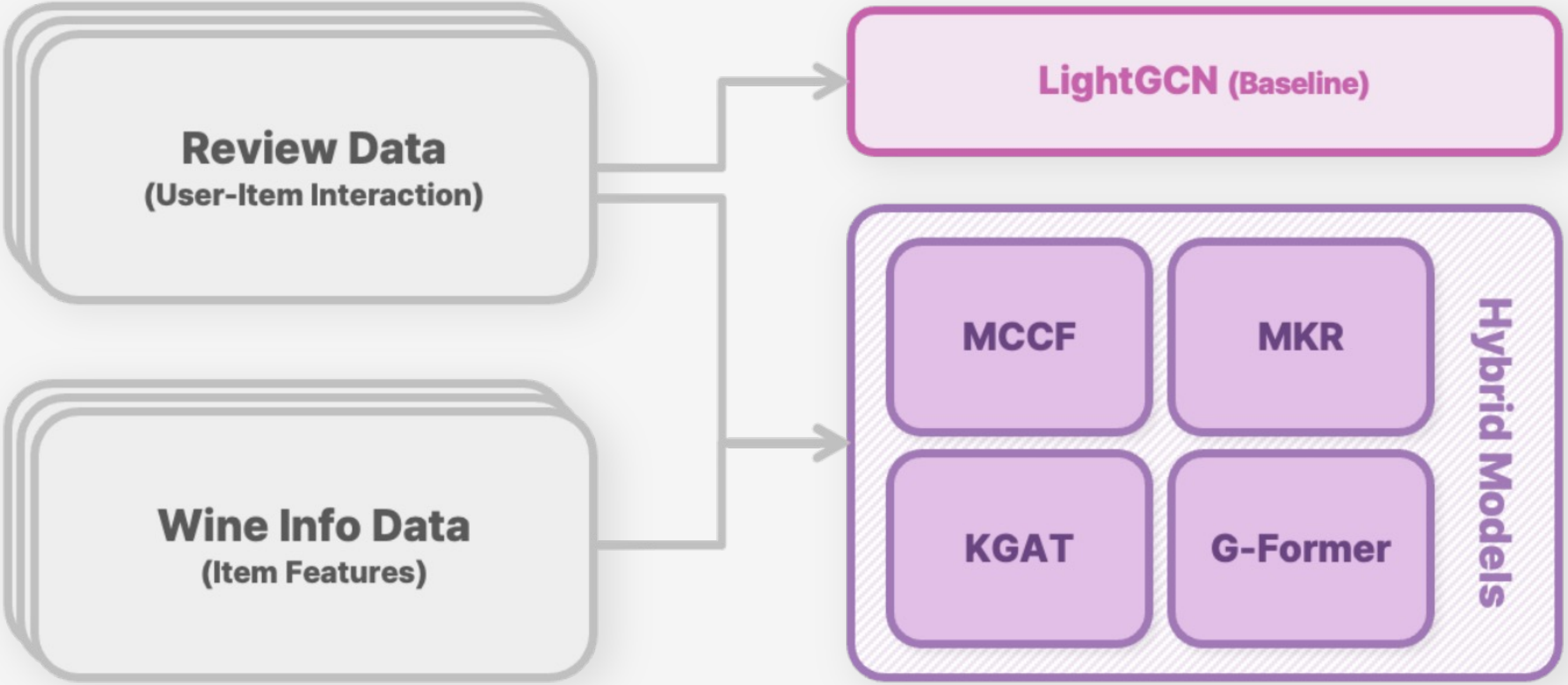
OUR GOAL

1. Collaborative Filtering for existing wine reviewers
2. Hybrid RecSys for existing and "NEW" wine customers
3. **Solve our dataset problem = Sparsity**

: Achieve high performance on 1, 2 goals , even under our sparse data

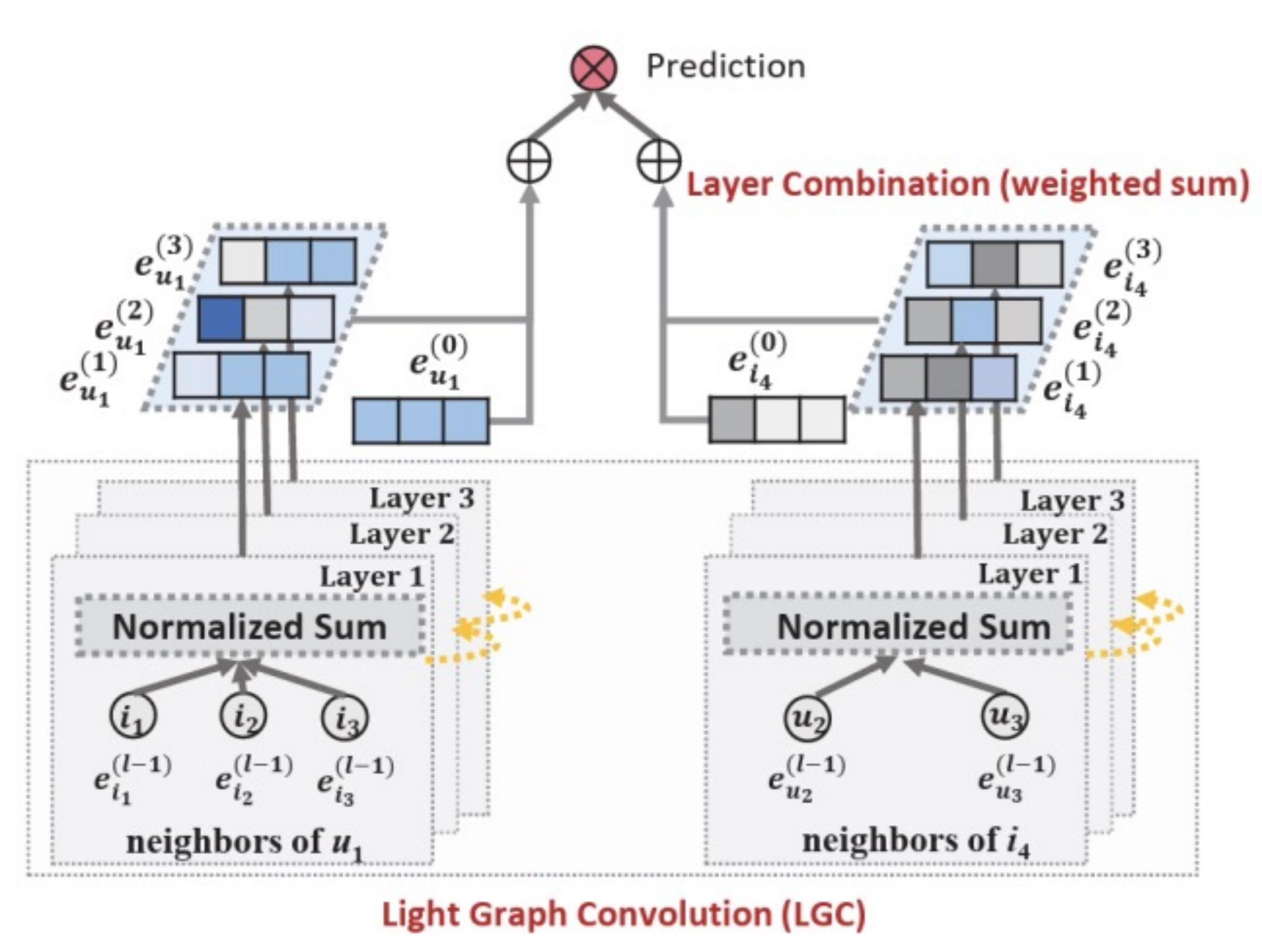


PIPELINE


Review Data
(User-Item Interaction)
Wine Info Data
(Item Features)
LightGCN (Baseline)
MCCF
MKR
KGAT
G-Former
Hybrid Models

LightGCN, SIGIR 2020

1. Only adapts neighbor aggregation from vanilla GCN, while others (feature transformation, nonlinear activation) do not.
2. The only learnable parameters are the embeddings of 0th layer.



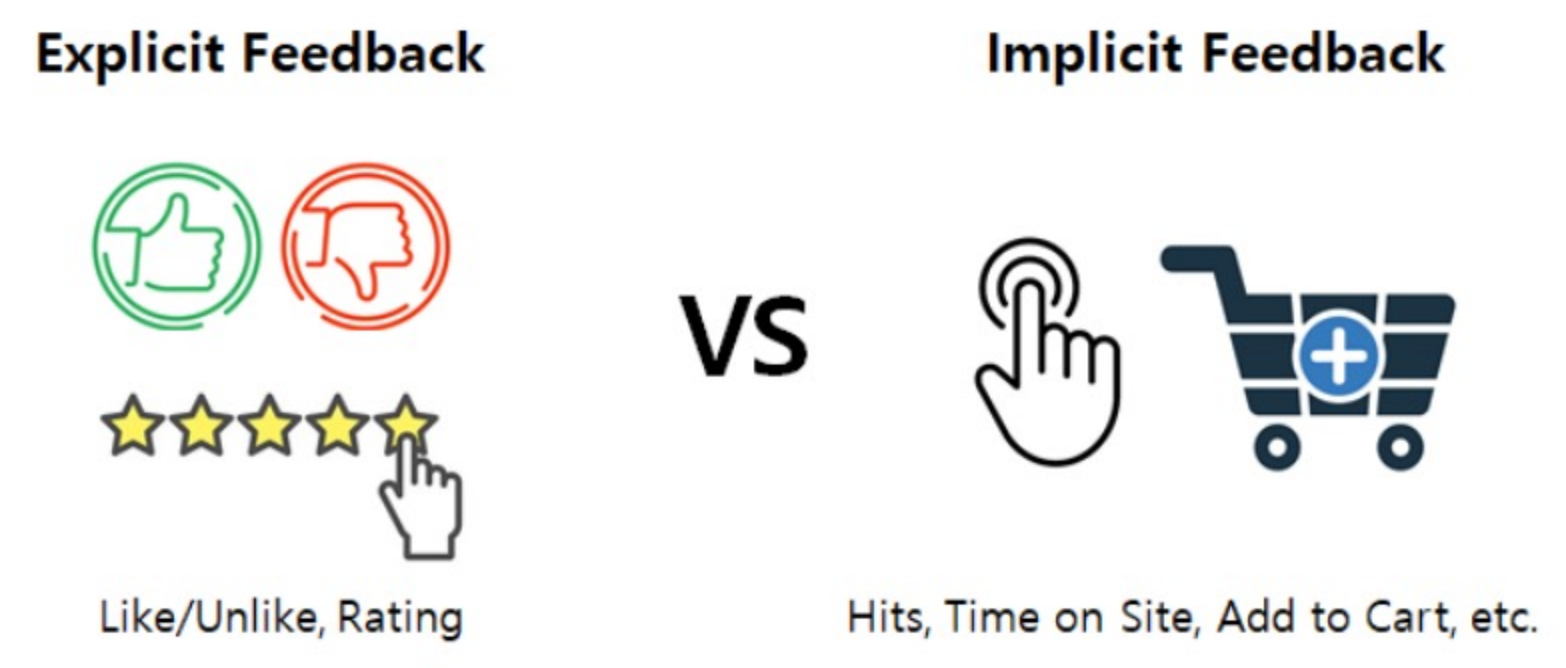
LightGCN, SIGIR 2020

Limitation & Trials of LightGCN

: *We focus on the implicit feedback setting (e.g., a user interacted with an item or not), where the user-item matrix is binary.*

- Thresholding by range: [0, 3] = 0, [4, 5] = 1
- Duplicating (user, item) pairs (e.g 5 stars = 5 pairs)
- Transforming into Weighted graph (matrix)

Q) Can we modify this problem while enhancing the model's performance?



MCCF (Multi-Channel Collaborative Filtering), AAAI 2020

1. consider multiple interaction types
2. use dataset as list, not matrix

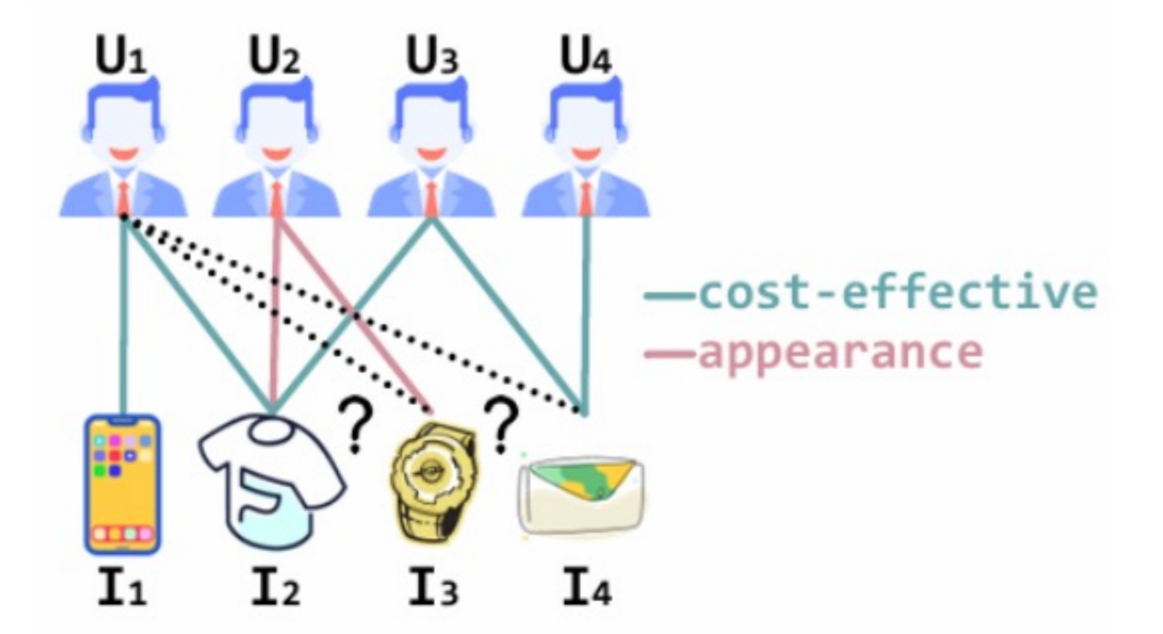


Figure 1: A toy example of purchasing relationships records with different purchasing motivations.

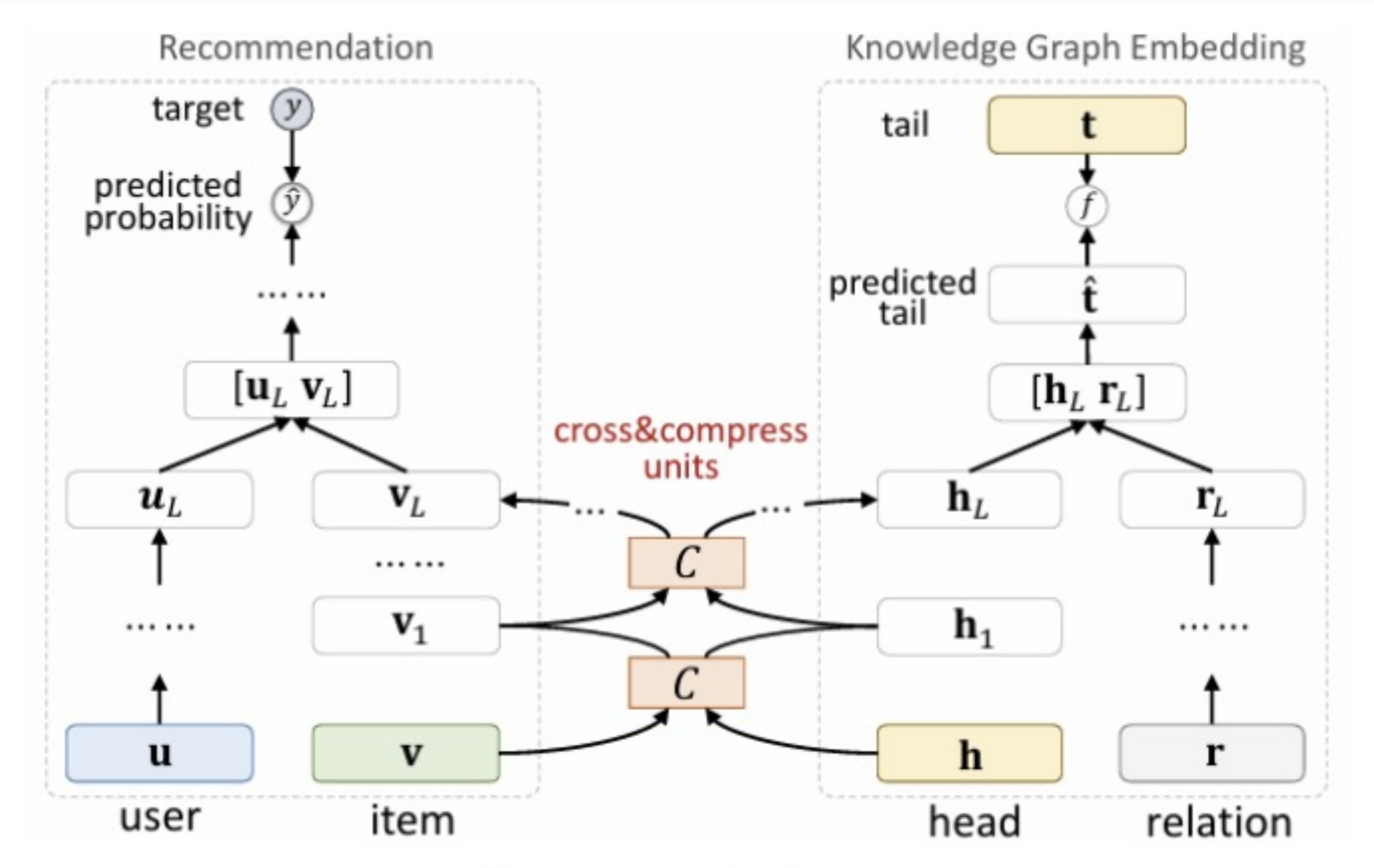
	Item1	Item2	Item3	Item4
User1	5.0	NaN	NaN	NaN
User2	NaN	4.0	NaN	NaN
User3	NaN	NaN	3.0	NaN
User4	NaN	NaN	NaN	2.0

User	Item	Rating
User1	Item1	5
User2	Item2	4
User3	Item3	3
User4	Item4	2

MKR (Multi-Task Knowledge Graph Reasoning) , WWW 2019

1. add Knowledge Graph Embedding

2. connected through a gate-based transfer mechanism



(a) Framework of MKR

KGAT (Knowledge Graph Attention neTwork) , KDD 2019

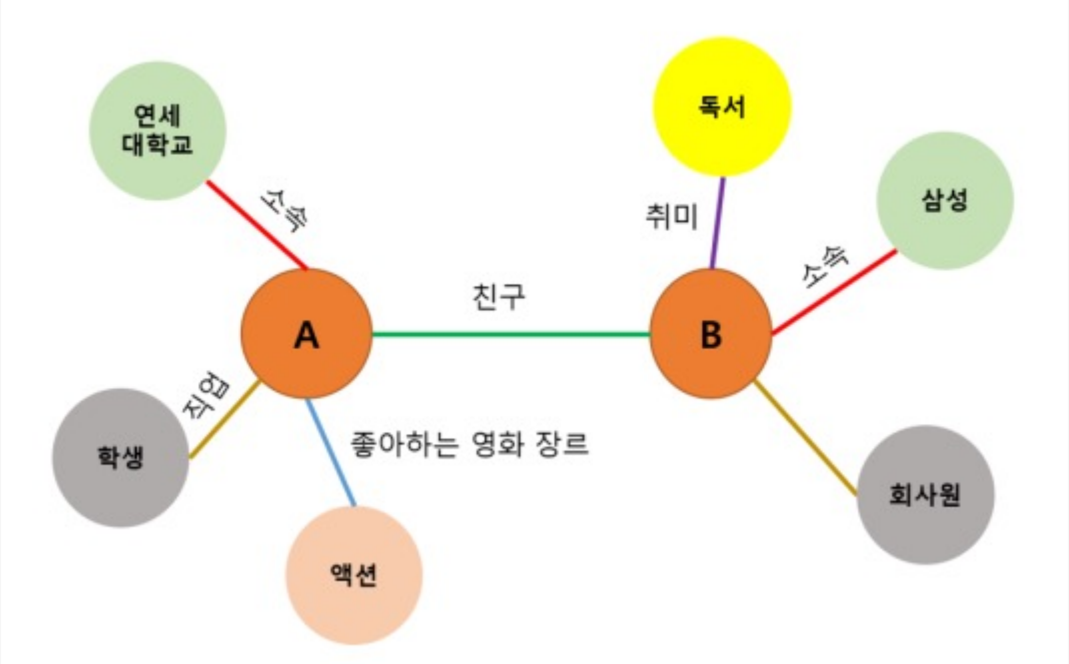
A model that goes beyond simple rating-prediction by utilizing KG (Knowledge graphs) and attention mechanism, and predicts scores for user–item pairs to determine their ranking.

Feature 1. Use of Knowledge Graph

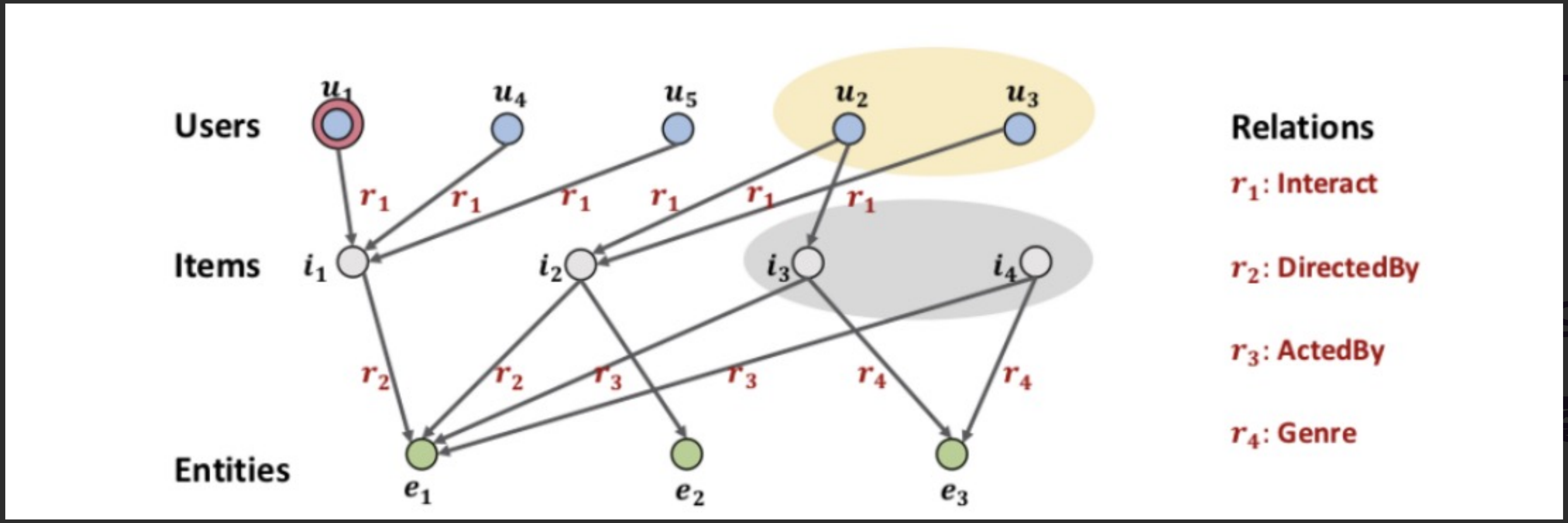
- Picks up not only user–item interactions, but also various item-related attributes (e.g., wine style) through KG

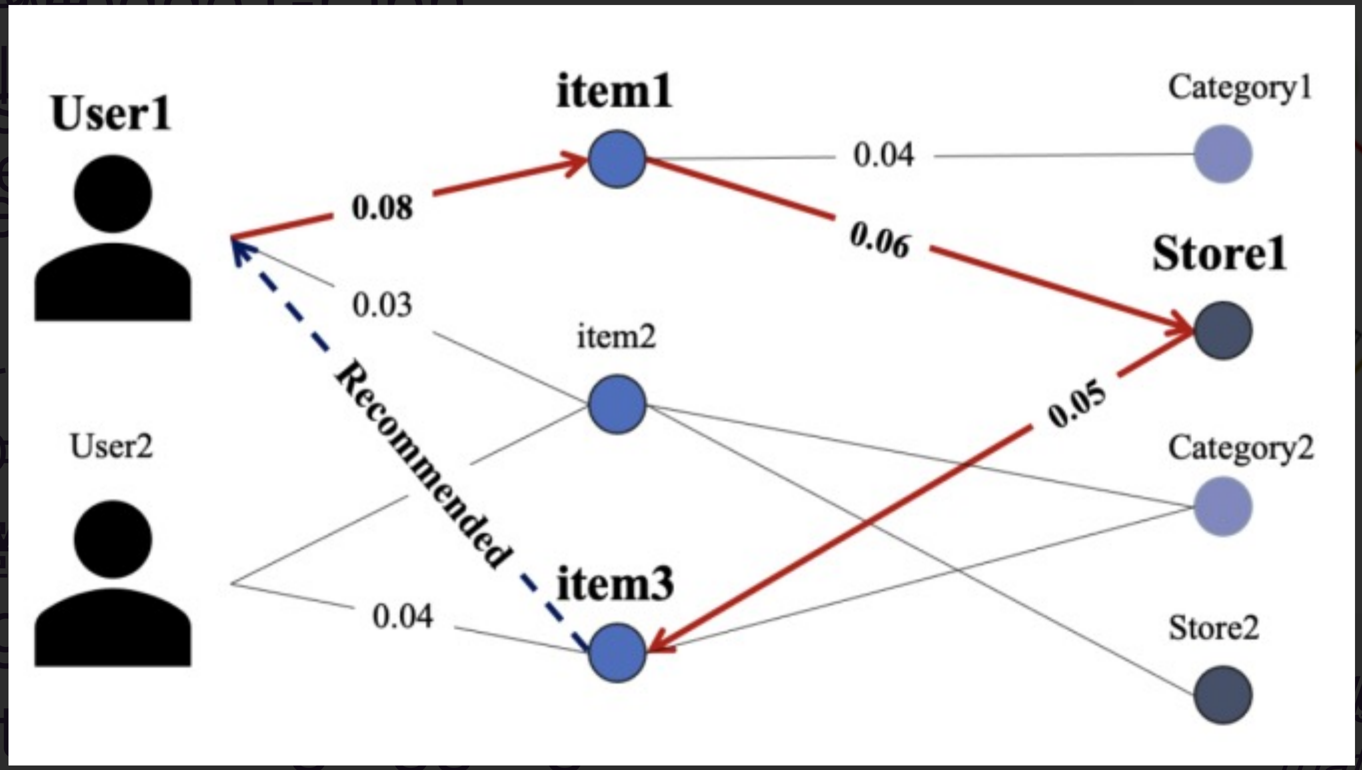
Feature 2. Attention Mechanism

- Assigns higher weights to more relevant neighboring attributes during aggregation.



**Knowledge Graph : Graph composed of edges that represent various types of attributes*





Allows yellow, gray areas to be recommended to user 1(u1), by exploring high-order relations

*High-order relations?

G-former (Graph-Transformer) , SIGIR 2023

A model that combine two GNN architecture-**GCN** and **TransformerConv**-to learn the relationships between users and wines in **a user-item interaction graph**

Function 1. G-former architecture based on collaborative filtering
(GCN + TransformerConv)

Function 2. Integration of wine-related side information
(e.g. characteristics, flavor, food pairing)

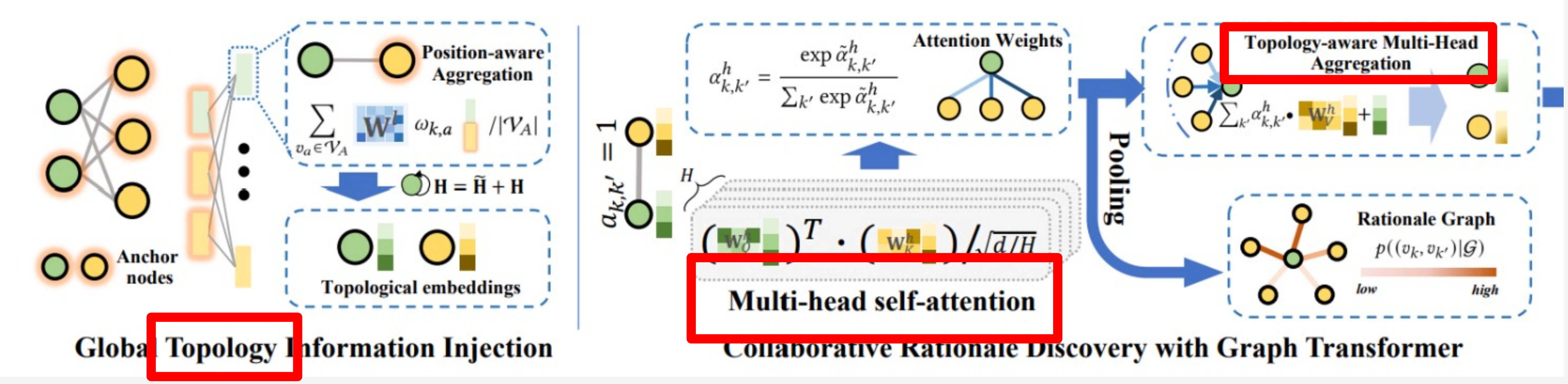
Function 3. Edge augmentation and self-supervised learning

1. G-Former Architecture Based on Collaborative Filtering

1. GCNConv: Aggregates neighbor information to capture collaborative patterns

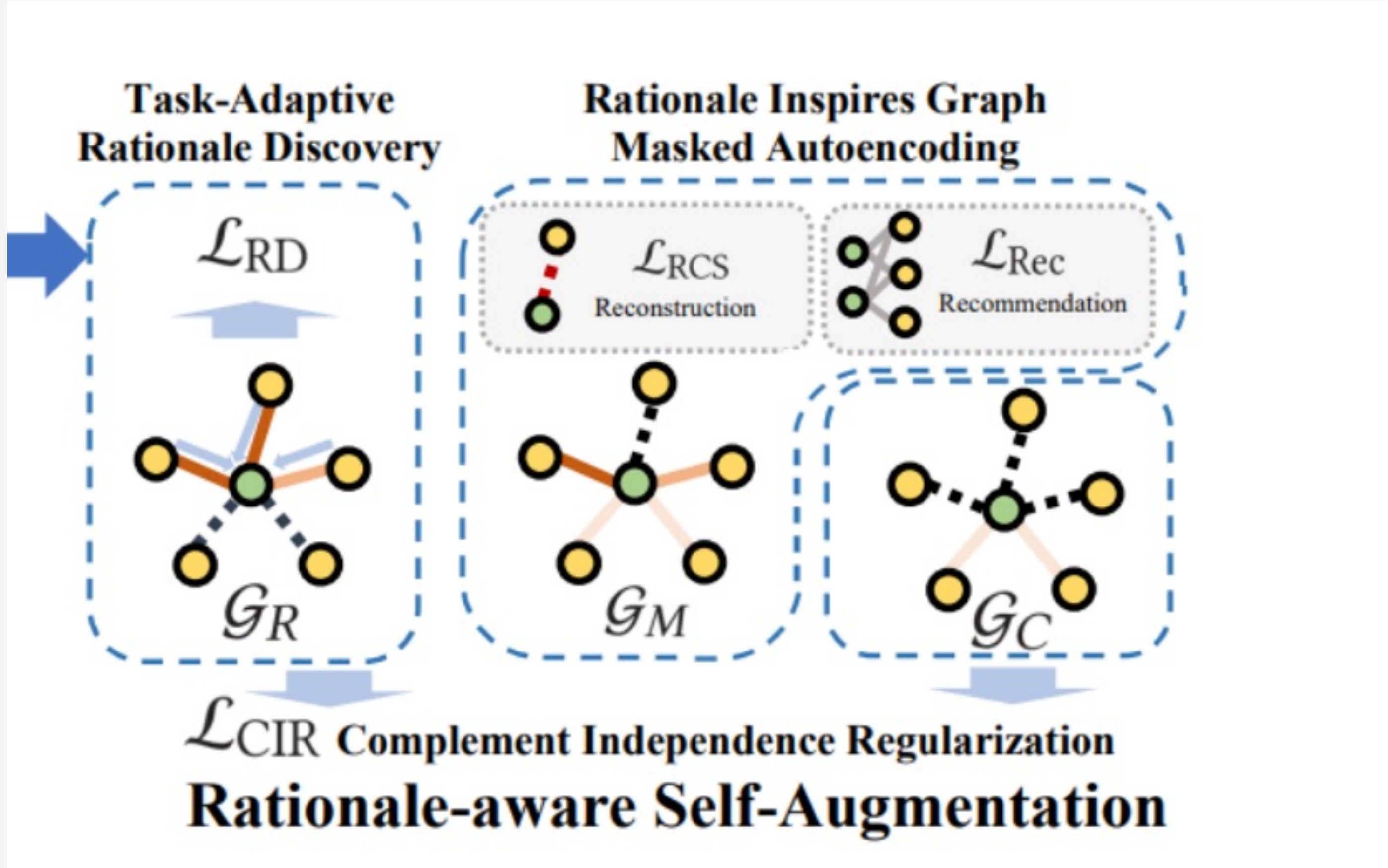
2. TransformerConv: Applies attention to weigh neighbors differently for richer relationship modeling.

3. G-Former: Learns personalized user–wine embeddings from the interaction graph for recommendation.



2. Edge Augmentation and Self-Supervised Learning

- **Edge Augmentation:** Randomly drops user–wine edges during training to improve robustness and generalization.
- **Edge prediction:** Learns to infer whether an edge exists, enhancing understanding of latent relationships.
- **Node Autoencoding:** Masks and reconstructs node embeddings to enrich their representations, even in sparse data.



CHAPTER.5

Evaluation Results

So, which is the best?

EVALUATION RESULTS

MSEloss

Squared loss between prediction and actual
In this case, review score (1~5)
is the main interest.

Cacluate focusing on the deviation
between these values

MSE = $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

Mean Squared Error loss

Recall@20

Metric based on the **consistency between the actual interacted items and top-20 model-recommended items.**

Mainly focus on the true positive counts among
all positive-predictions.

Recall@K = (# of items in top−K) / (total # of items)

True Positive among Top−K

NDCG@20

Simultaneously consider the items and their ranks by calculating the ration between
IDCG (Ideal DCG) and DCG.

Useful for the case when evaluating the
ability of ranking prediction additional to
accuracy of the recommendation.

NDCG@K = DCG@K / IDCG@K

Normalized Discounted Cumulative Gain

EVALUATION RESULTS

	RMSE	Recall@20	NDCG@20	Precision@20
LightGCN	–	0.0490	0.1923	0.1618
KGAT	–	0.0380	0.6736	0.6565
MCCF	1.7732	0.0245	0.0491	0.0478
MKR		0.0438	0.2436	0.2275
Gformer (Augemented)	0.5896	0.1617	0.8834	0.8654

CHAPTER.6

Limitations & Contributions

Limitations

1. The sparsity in our dataset is significantly higher than the level typically addressed in the literature.
2. While MCCF is designed to capture various types of interactions in a multi-dimensional manner, our data only contains a single type of edge (ratings)
3. Computational burden in handling large-scale knowledge graphs
4. Low embedding quality in related wine information

Contributions

1. We constructed our own benchmark by collecting wine-related data.
2. We evaluated which model is most effective in addressing the sparsity problem commonly observed in graph-based recommendation systems.


Q&A


THANK YOU