STORE

Recommendation Systems Project

Mate Balogh

STORE

Item recommendations for **visitors** in an online store based on *implicit feedback* to provide better user experience and to boost sales.

Methodology

1. Dataset
2. Business needs
3. Requirement details
4. Model planning
5. Model construction
6. Explanation
7. Evaluation

Dataset

E-commerce dataset[1]

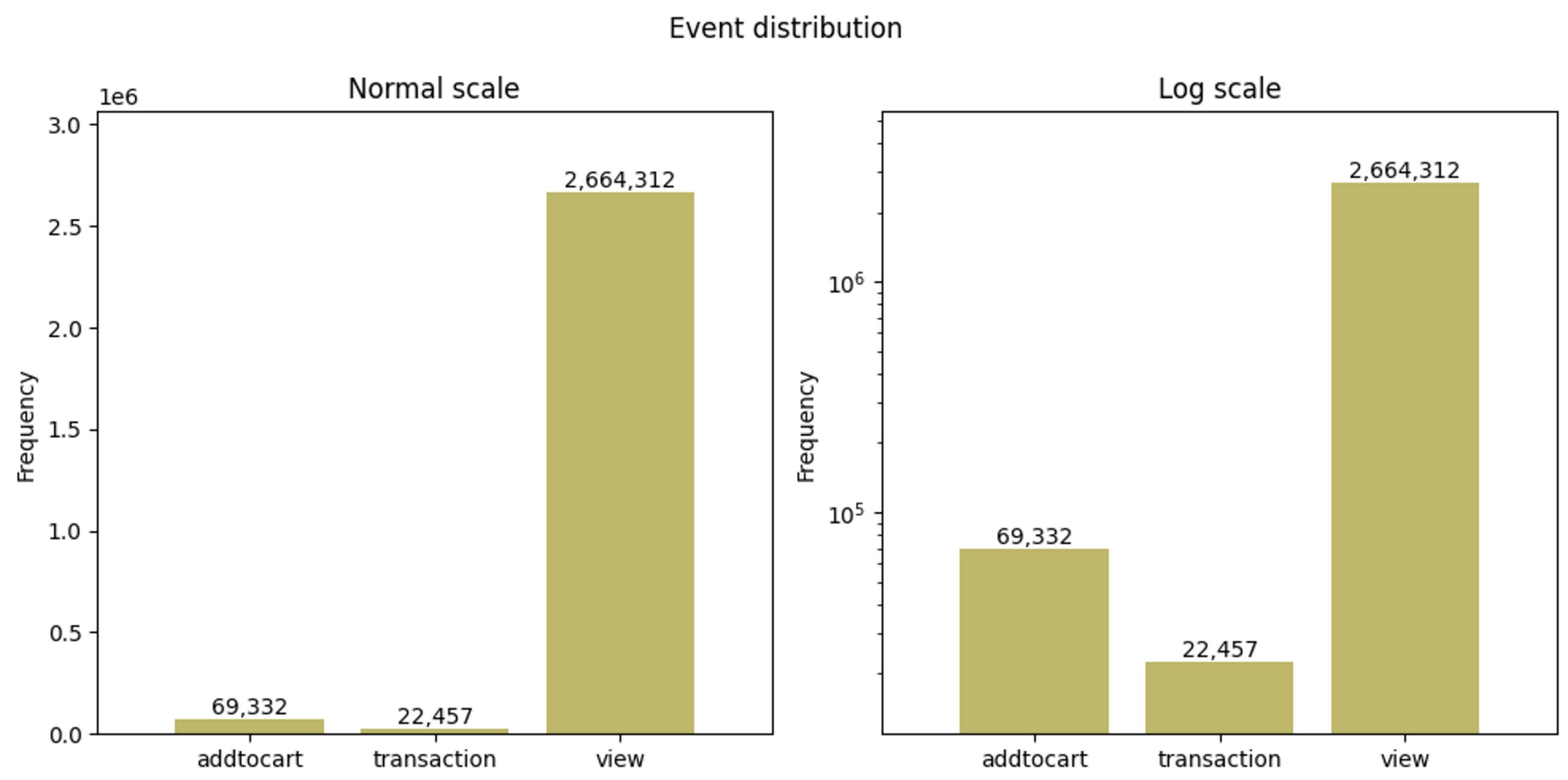
Elements:

- Tree of categories
- **Click stream data of users**
- Item properties

Event types in stream:

- *View*
- *Add to cart*
- ***Transaction***

[1] *Retailrocket recommender system dataset from Kaggle*



Event distribution
Normal scale
1e6
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Frequency
69,332
22,457
2,664,312
addtocart
transaction
view
Log scale
10^6
10^5
Frequency
69,332
22,457
2,664,312
addtocart
transaction
view

Business Needs

- **Home page recommendations**
  Popular items, Items purchased by similar users
- **Newsletter recommendations**
  Discovered patterns

STORE

Home Categories

1

All-time Favorites

See more

Item #461686

Pellentesque gravida magna nec leo malesuada, sit amet dictum dui euismod.

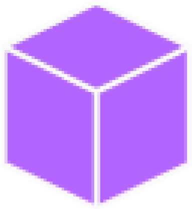
Item #119736

Item #213834

Item #7943

Item #312728

Others are Coming Back for

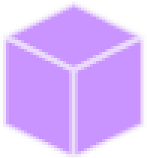


Item #396042

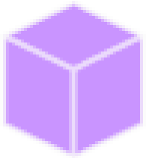
Ut id imperdiet lectus. In blandit scelerisque odio eu fringilla. Maecenas sit amet pretium nunc.

See more

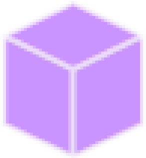
Item #224549



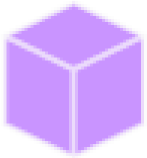
Item #147



Item #218612



Item #347641



We think You will Love these

See more

Item #10572

Ut id imperdiet lectus. In blandit scelerisque odio eu fringilla. Maecenas sit amet pretium nunc.

Item #171878

Item #218794

Item #40630

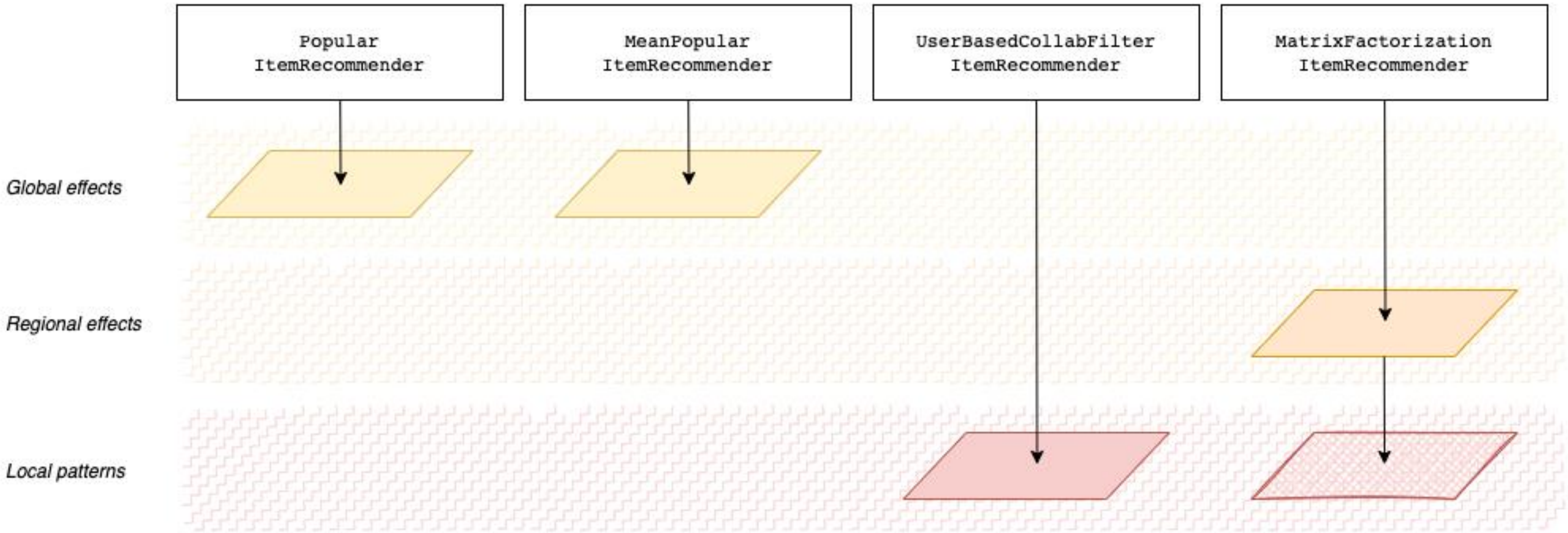
Item #32581

Requirement Details

Business need	**In-app title**	**Expected behavior**	**Model name**
Popular item recommendation	*All-time Favorites*	The N most-frequently purchased items are recommended.	`PopularItemRecommender`
Popular item recommendation	*Others are Coming Back for*	The N items with the largest mean purchase frequency are recommended.	`MeanPopularItemRecommender`
Items purchased by similar users	*We think You will Love these*	Based on the entire purchase history of the given user, similar K users are selected and N items from their purchases (not yet bought by the given user) are recommended.	`UserBasedCollabFilterItemRecommender`
Discover hidden patterns	*Already Picked for You*	Based on the entire purchase history of all the users, hidden factors are discovered via *Matrix Factorization*. N not yet purchased items with the top score implied by the factors are recommended for the given user.	`MatrixFactorizationItemRecommender`

Model Planning

Model name	User elements	Measurement	Similarity	Filtering	Selection
`PopularItemRecommender`	-	Purchase frequency, item-wise	-	-	Top N items
`MeanPopularItemRecommender`	-	Mean purchase frequency, item-wise, after grouping by user	-	-	Top N items
`UserBasedCollabFilterItemRecommender`	Previously purchased items	Binary purchase flag, user-item pairwise	Cosine similarity, user-user pairwise	K-most similar users (threshold: 0.1)	Top N items
`MatrixFactorizationItemRecommender`	Previously purchased items for all users (implicit feedback, weighted)	Discovered latent factors (U, V)	Dot product of latent factors of users and items (U, $V.T$)	-	Top N items


Popular
ItemRecommender
MeanPopular
ItemRecommender
UserBasedCollabFilter
ItemRecommender
MatrixFactorization
ItemRecommender
Global effects
Regional effects
Local patterns

Model Construction*

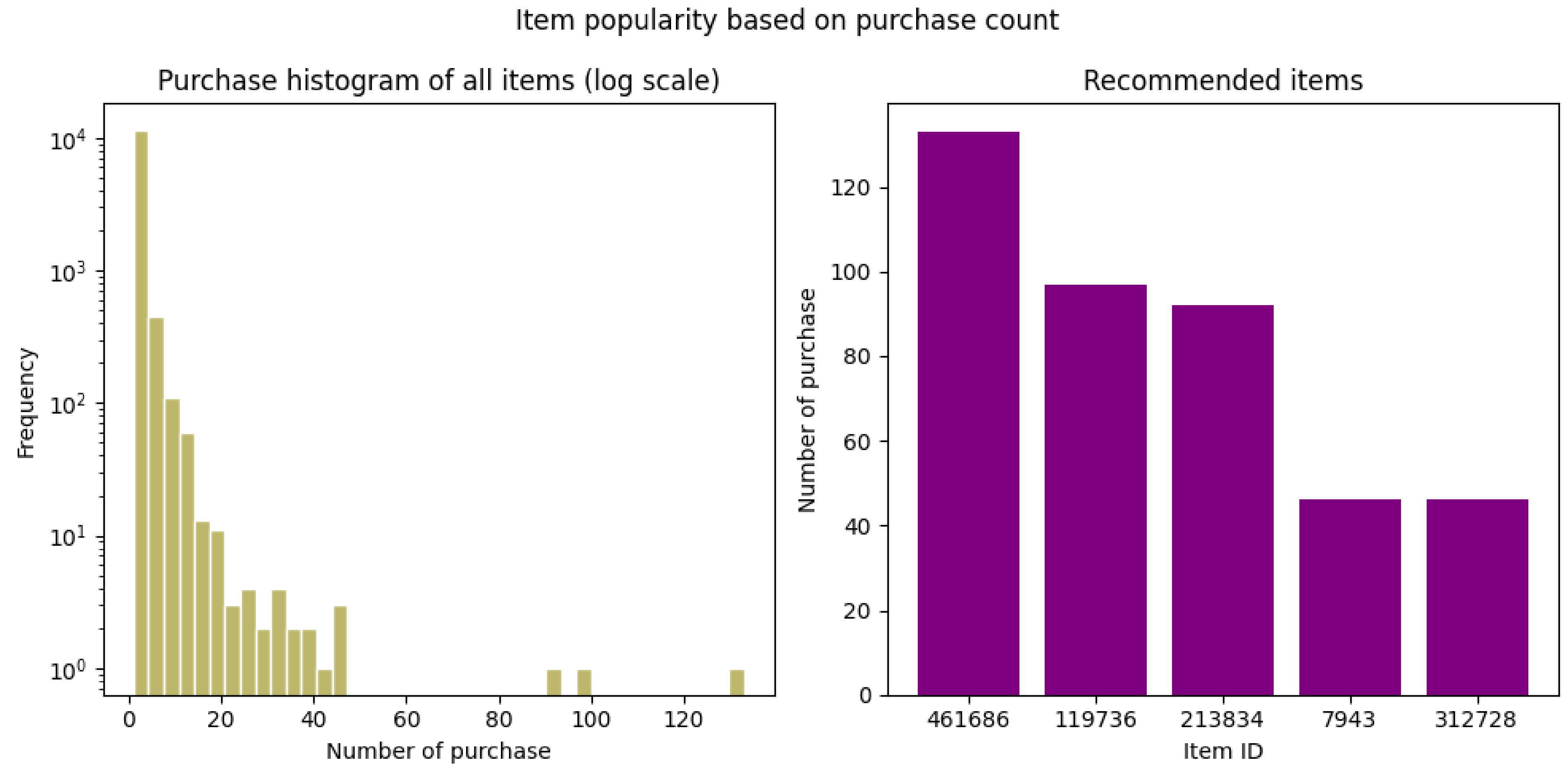
```
items_recommended = (
    PopularItemRecommender(transaction_events, verb=True)
        .recommend(N)
)
```

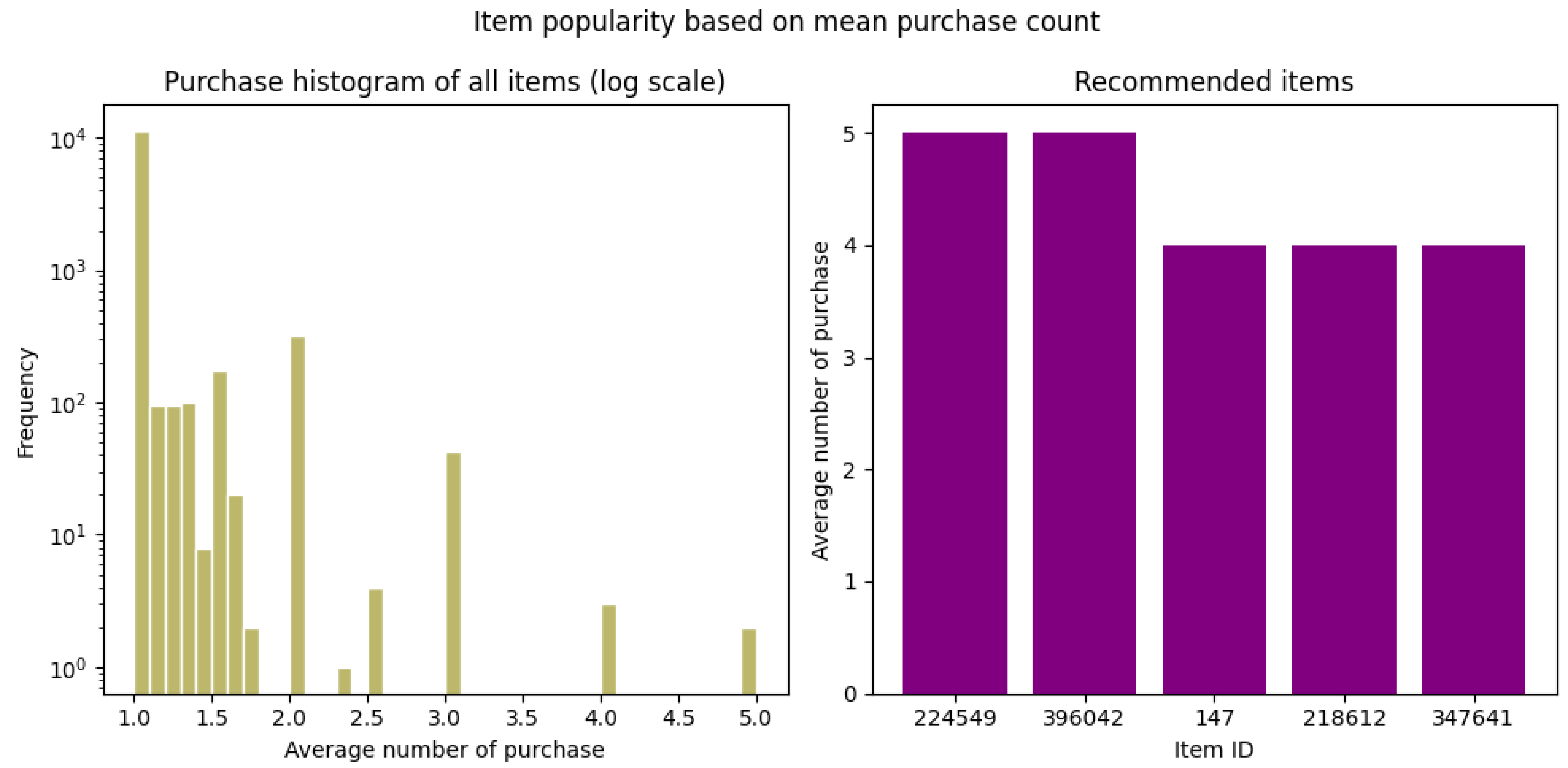
```
items_recommended = (
    MeanPopularItemRecommender(transaction_events, verb=True)
        .recommend(N)
)
```

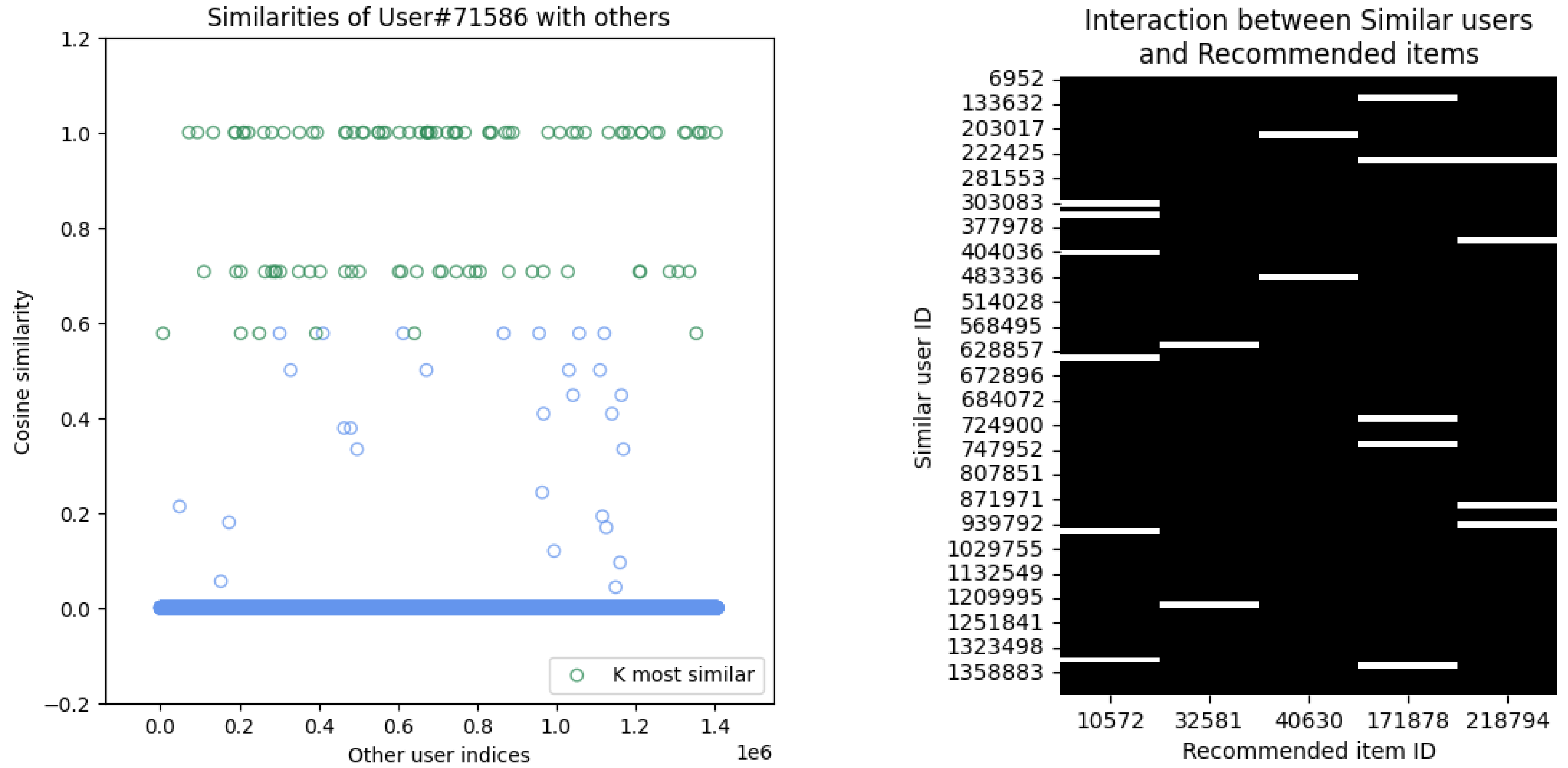
```
items_recommended = (
    UserBasedCollabFilterItemRecommender(visitor_item_pivot, visitor_item_similarity, verb=True)
        .fit(designated_visitor_id)
        .recommend(K, N, sim_threshold=0.1)
)
```

* Model call syntaxes. For the model implementations, visit the Python notebook on GitHub

Explanation

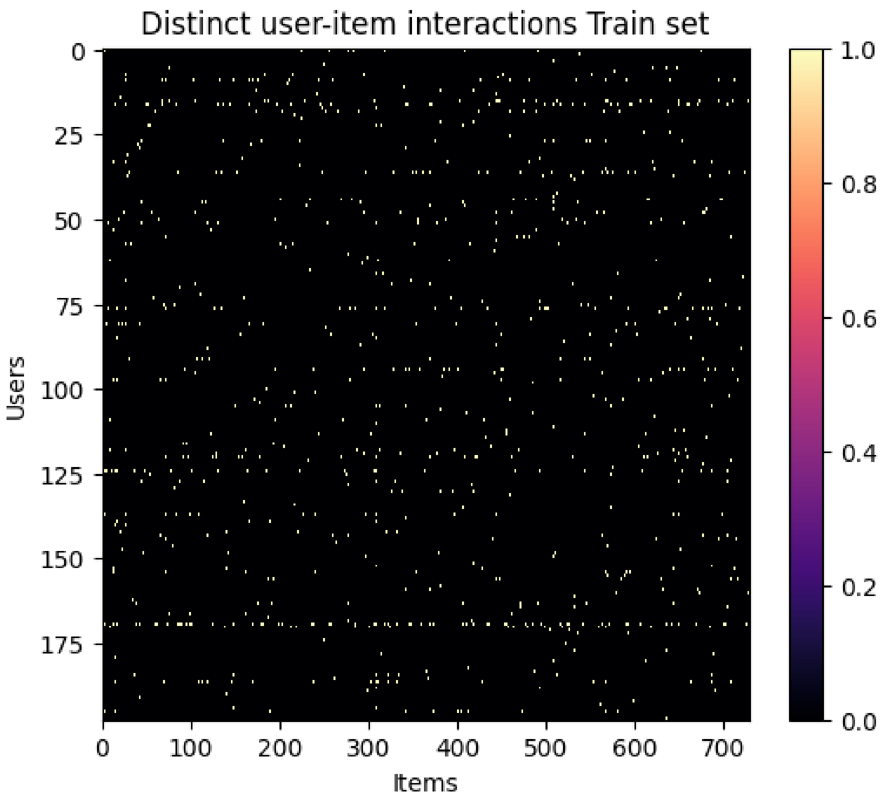

Item popularity based on purchase count
Purchase histogram of all items (log scale)
Frequency
Number of purchase
Recommended items
Number of purchase
Item ID
461686
119736
213834
7943
312728

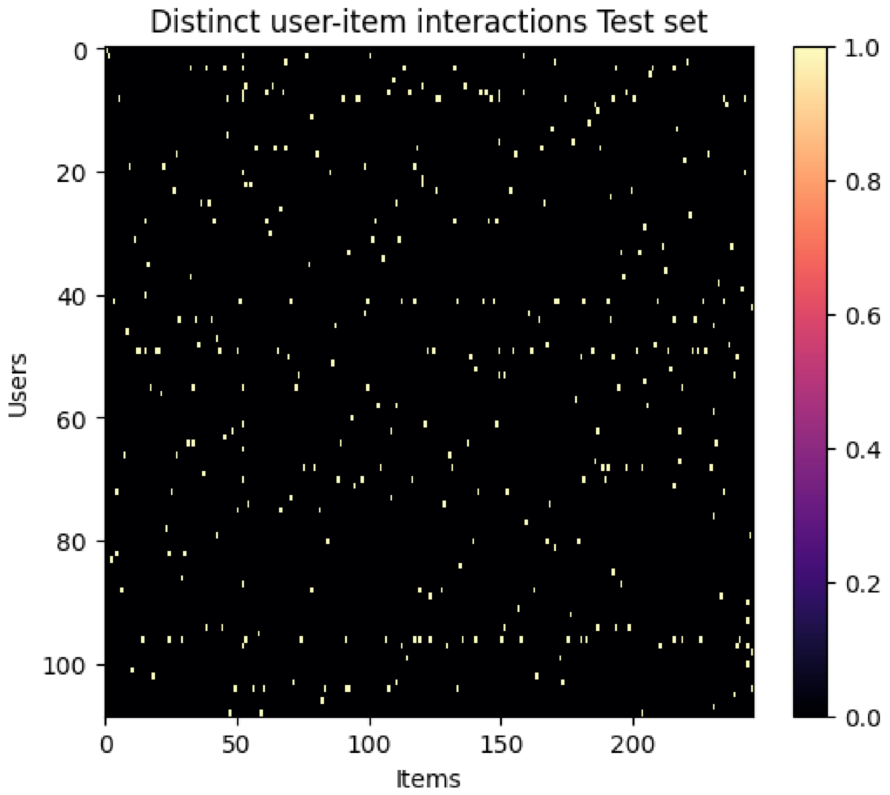

Item popularity based on mean purchase count
Purchase histogram of all items (log scale)
Frequency
Average number of purchase
Recommended items
Average number of purchase
Item ID
224549
396042
147
218612
347641


Similarities of User#71586 with others
Cosine similarity
Other user indices
1e6
K most similar
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Interaction between Similar users and Recommended items
Similar user ID
Recommended item ID
6952
133632
203017
222425
281553
303083
377978
404036
483336
514028
568495
628857
672896
684072
724900
747952
807851
871971
939792
1029755
1132549
1209995
1251841
1323498
1358883
10572
32581
40630
171878
218794

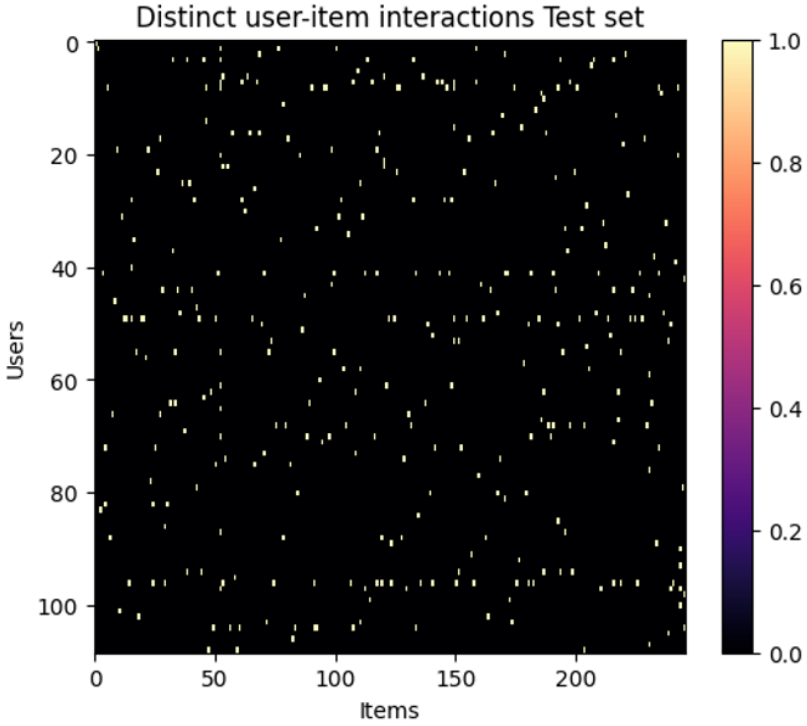
Matrix Factorization (Weighted)

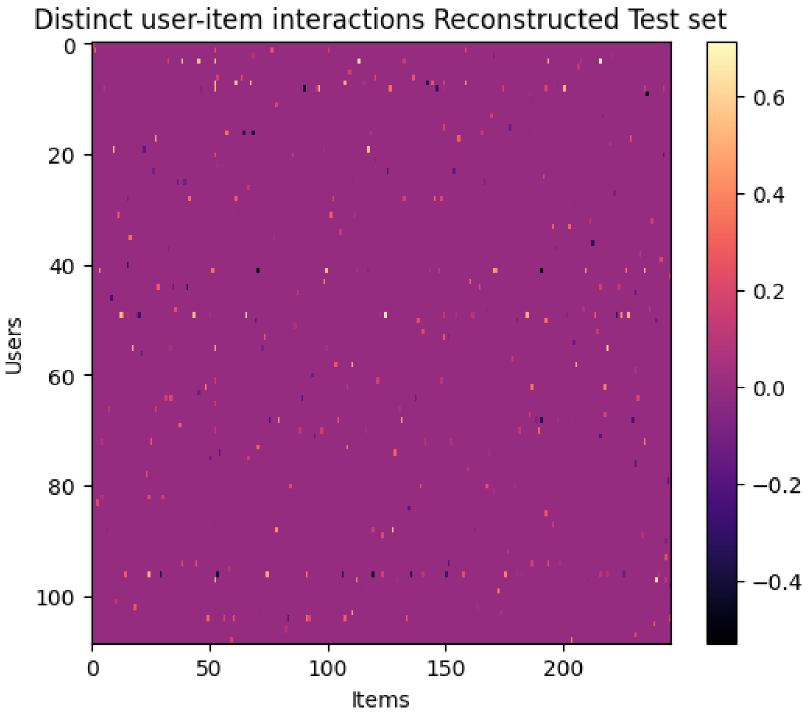
```
model = MatrixFactorizationItemRecommender(verb=True)
model.experiment(ratio_split, metrics, k=2, max_iter=100, learning_rate=0.001)
```

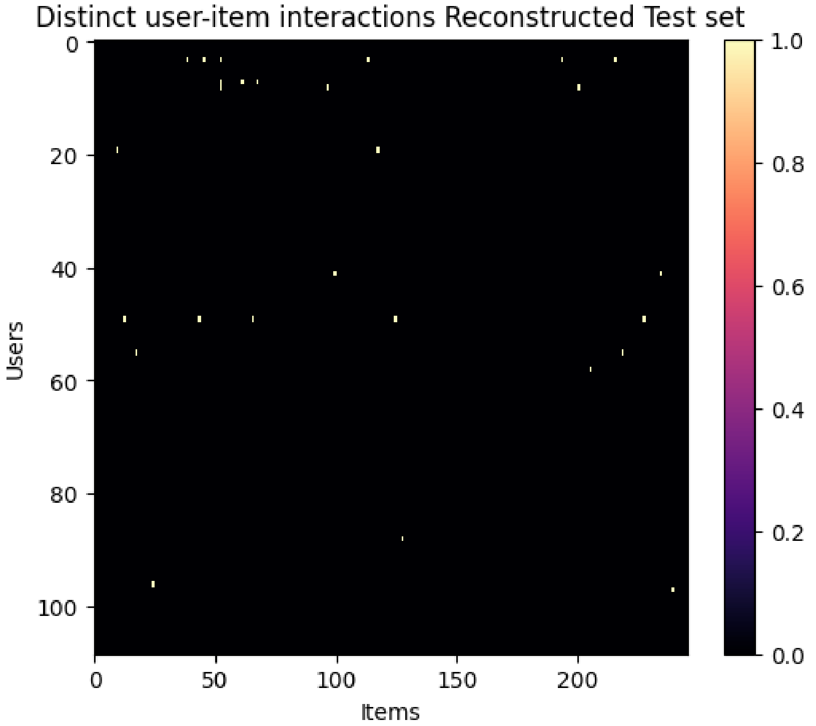




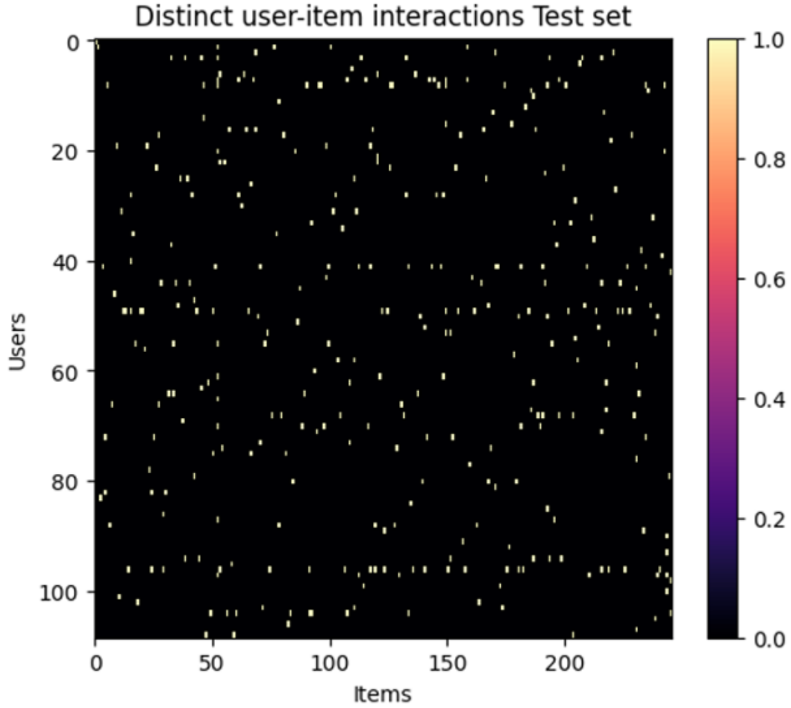
Reconstruction with U and V



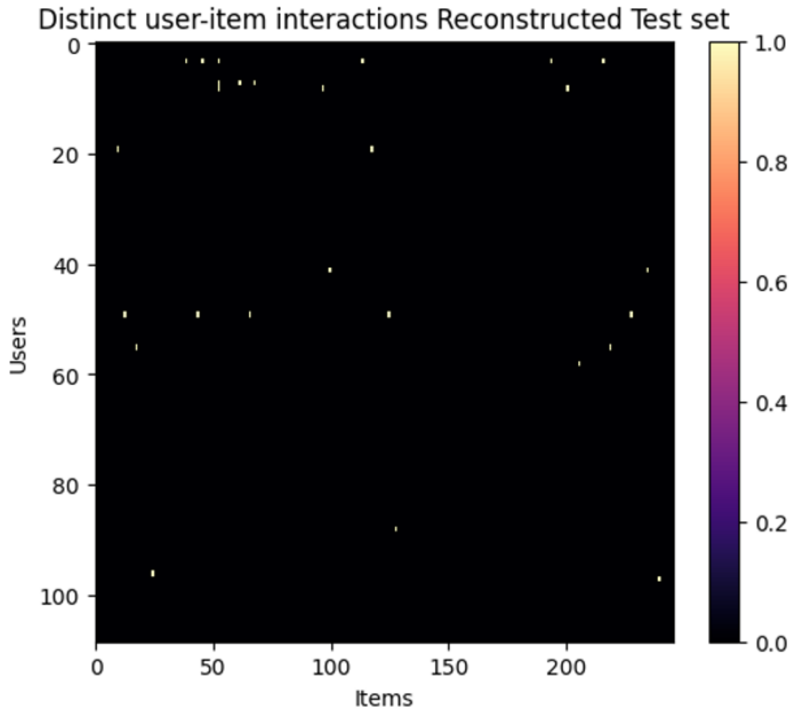




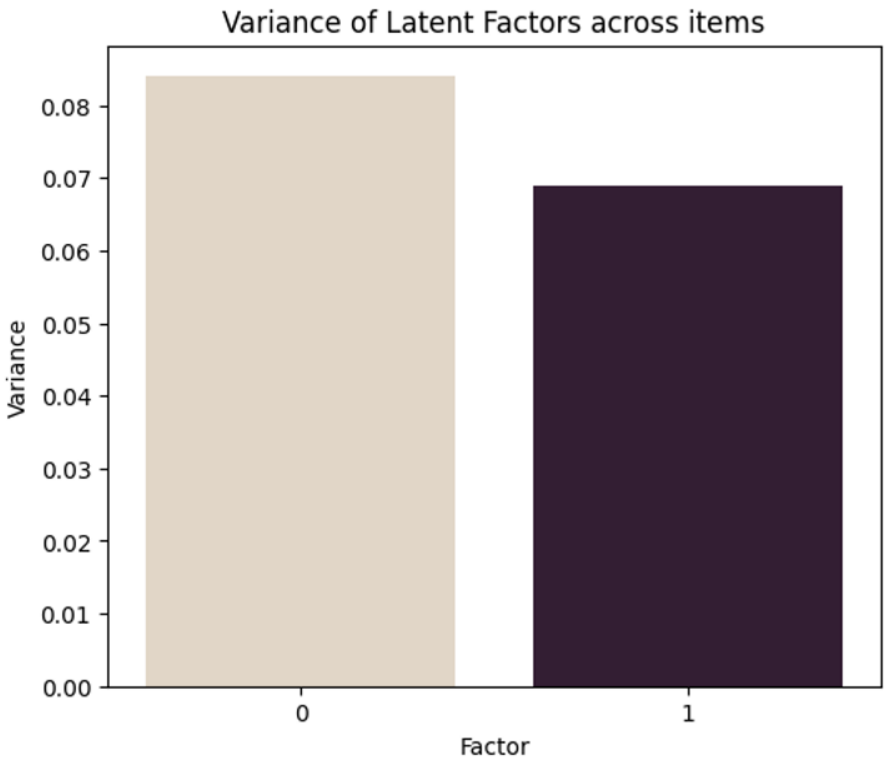
Reconstruction with *U* and *V*

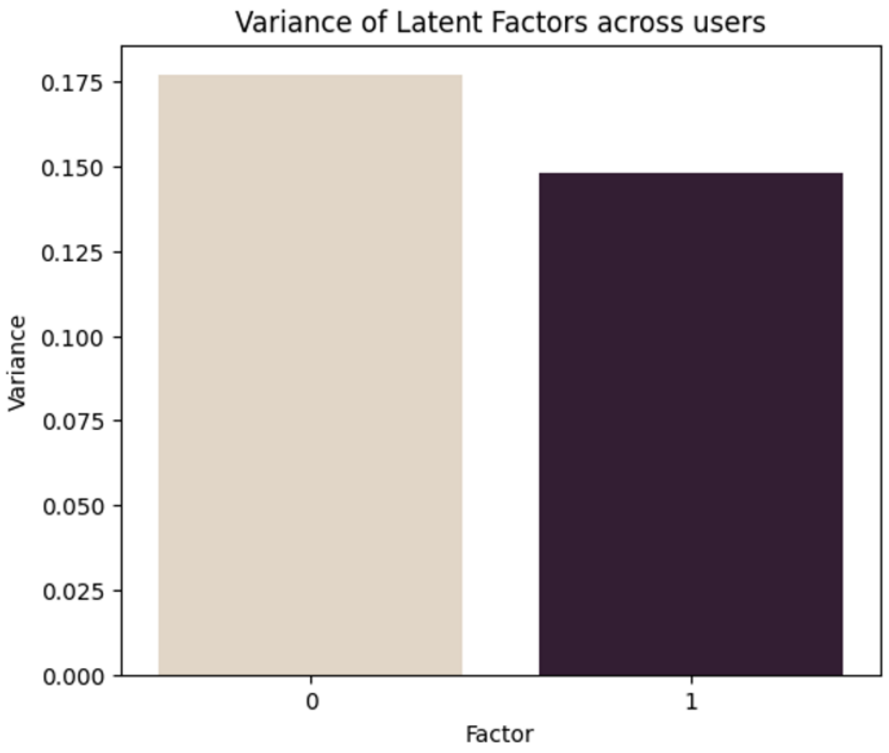


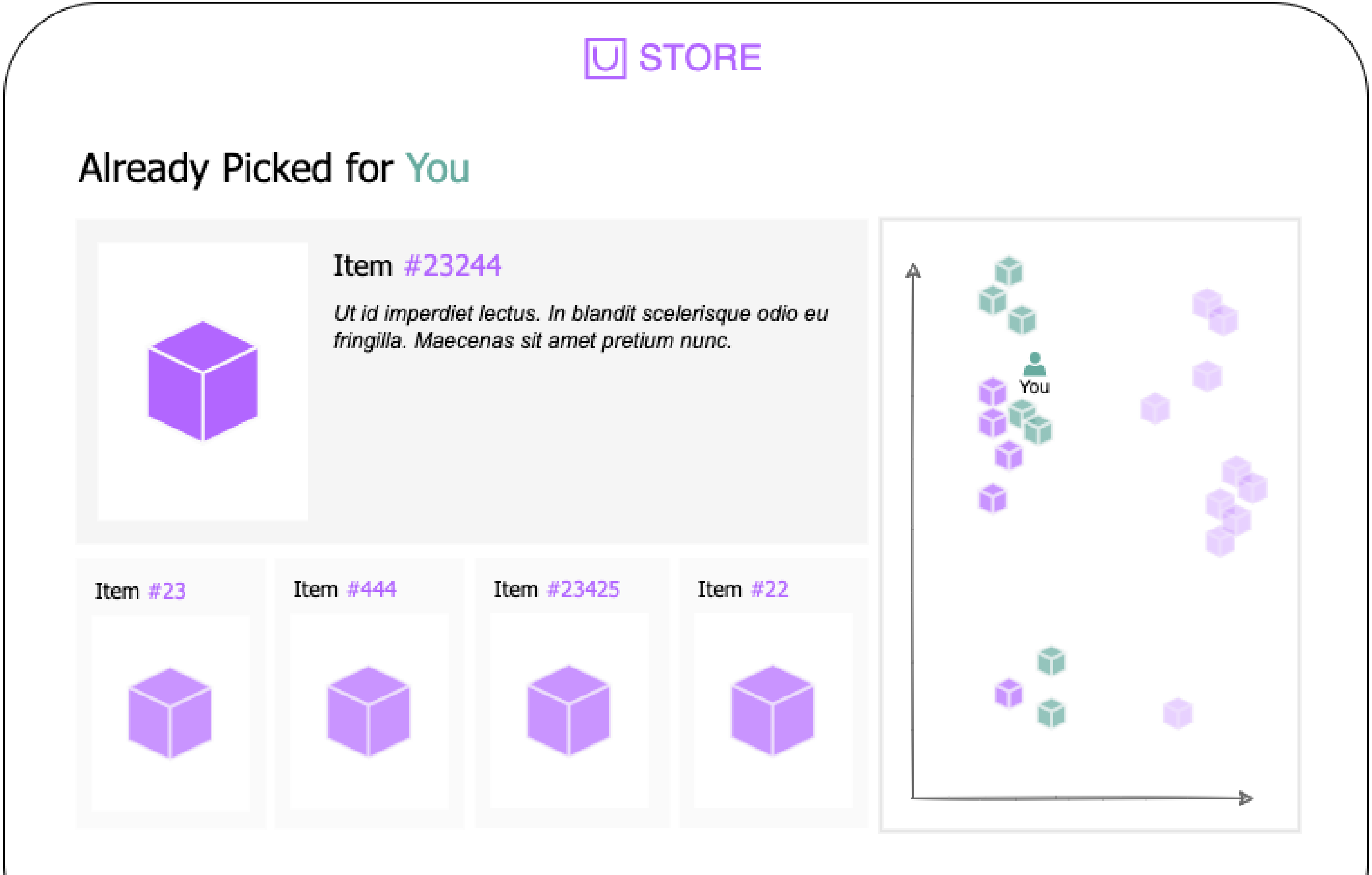
Reconstruction with *U* and *V*



Variance of Latent Factors






STORE
Already Picked for You
Item #23244
Ut id imperdiet lectus. In blandit scelerisque odio eu fringilla. Maecenas sit amet pretium nunc.
Item #23
Item #444
Item #23425
Item #22
You

Future Directions

- Investigate effect of bias(es)
- Complete MF model
- Interpret behavior of MF in details
- Evaluate model performances
- Incorporate other implicit feedbacks
  (view, addtocart, addtocart -> transaction)

Project-based Topics and Questions

1. **Collaborative Filtering Fundamentals and Similarity-Based Methods**
2. **Matrix Factorization and Latent Factor Models**
3. **Handling Implicit Feedback and Data Preprocessing**

1. Collaborative Filtering Fundamentals and Similarity-Based Methods

Source: Recommender Systems Handbook – Chapter 4:
Neighborhood-based Recommendation Methods

Key aspects:

- Core concepts of user-based and item-based collaborative filtering
- Different similarity metrics (e.g., Cosine, Pearson)
- Rating normalization and neighborhood selection strategies

2. Matrix Factorization and Latent Factor Models

Source: ch09-recsys2.pdf
Concepts of SVD, latent factors, and gradient descent optimization

Key aspects:

- Representing users/items as vectors in a shared latent space
- Learning via RMSE minimization using gradient descent
- Comparing factorization with neighborhood methods

3. Handling Implicit Feedback and Data Preprocessing

Source: *Recommender Systems Handbook* – Chapter 2:
Data Mining Methods for Recommender Systems

Key aspects:

- How to interpret implicit feedback (views, cart additions) as signals of user preference
- Denoising, dimensionality reduction, and sampling strategies
- Importance of preprocessing for high-quality recommendations