Recommendation system

Business Analytics –Project presentation

Lalit

Claude

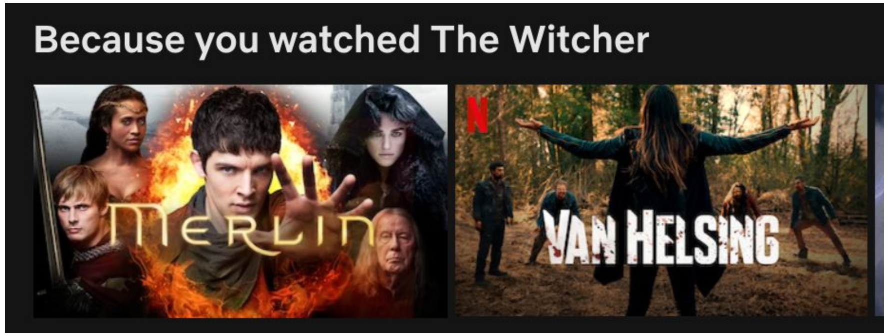
Nishita

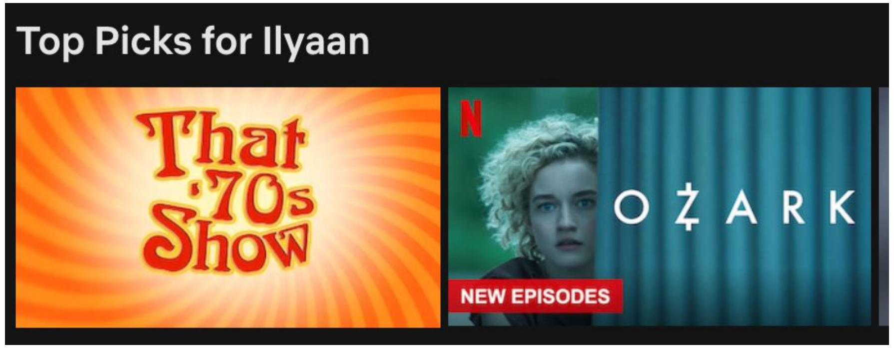
Umesh

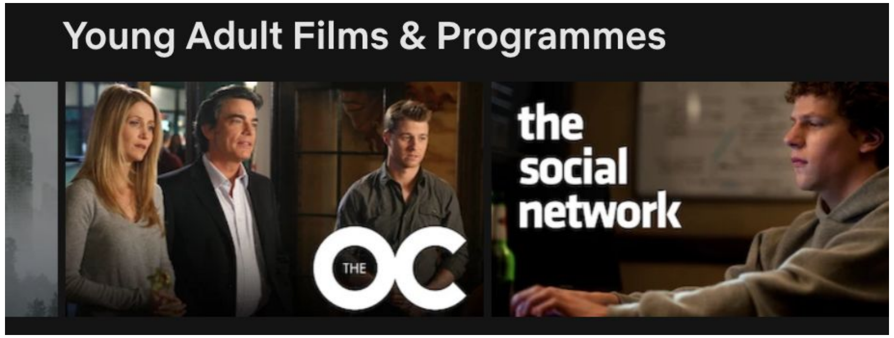
Mehdi

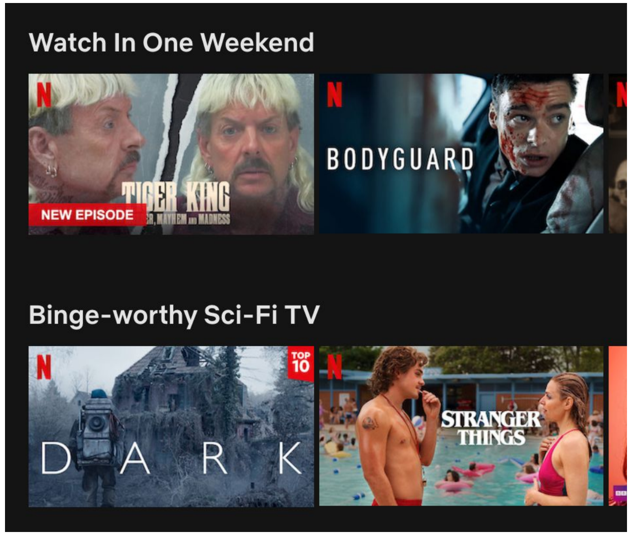
Inaara

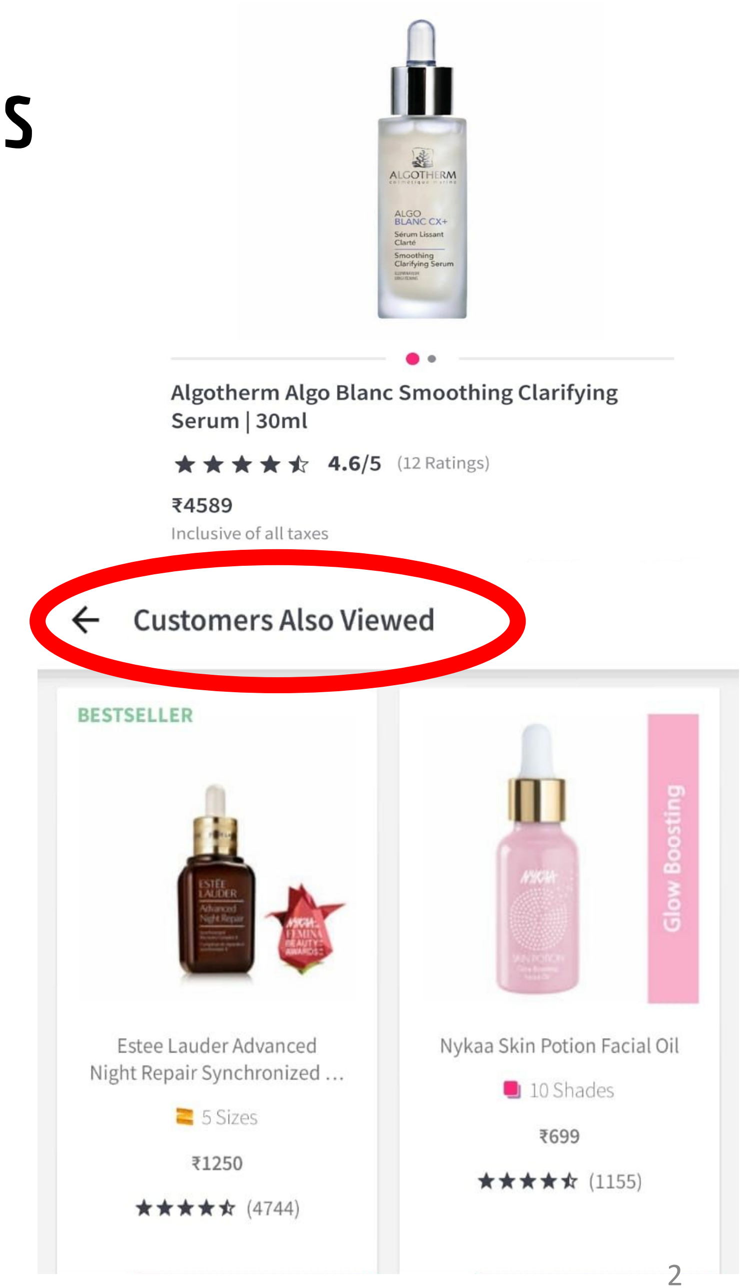
Quick Introduction to Recommendation Systems











Amazon Luxury Beauty: Objective and Dataset overview

- Using a dataset by Amazon for Luxury Beauty Products. It includes:
  - ‘product_id’: A unique name (alphanumeric) that identifies the different products
  - ”user_id”: A unique name (alphanumeric) that identifies the different users
  - ”rating”: Integer values between 1 and 5 given by a user for a product
- 86,771 ratings;
- 19,748 users (users that rated 3+ products)
- 8,308 total products
- Objective: Based on the ratings given by a user for a product, build a recommendation system for Luxury Beauty products on Amazon.com

Quick introduction: Types of Recommendation systems

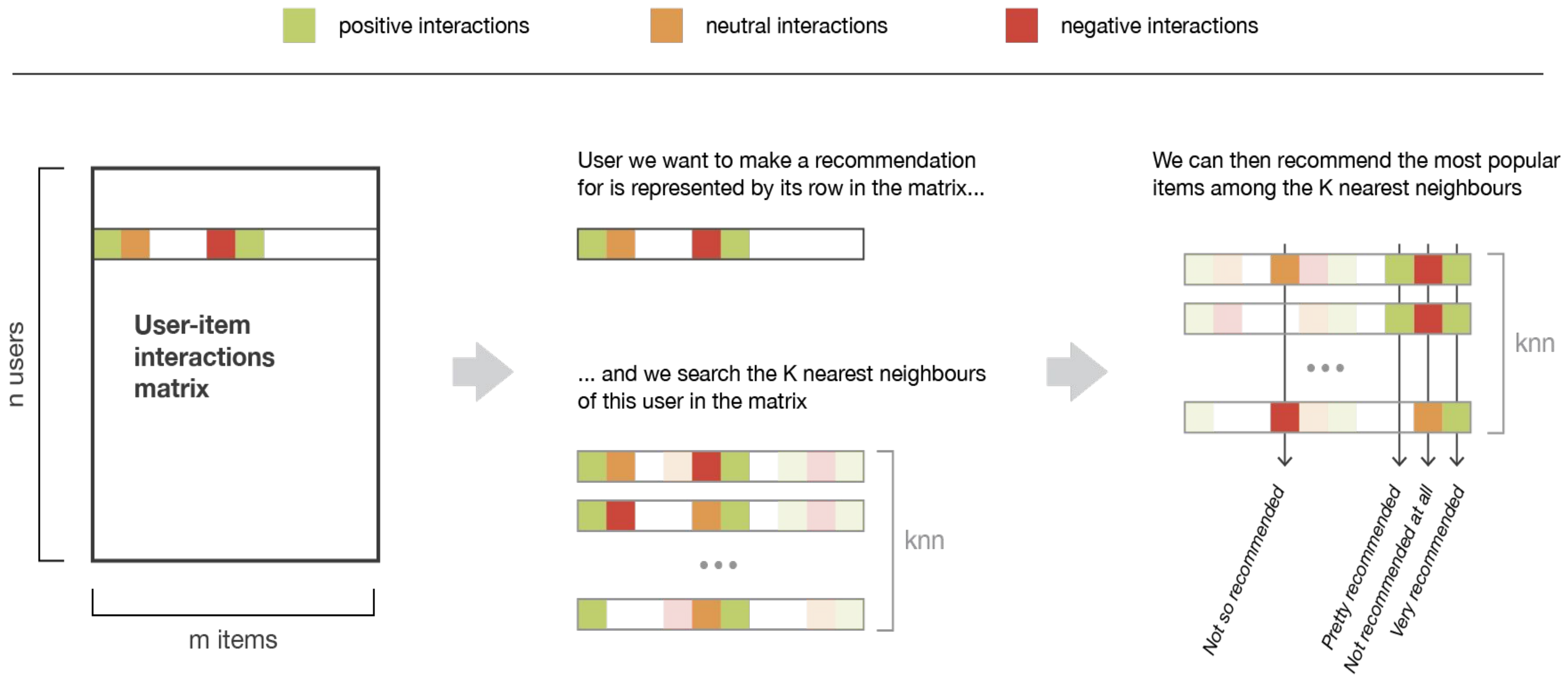
- Content based methods
  - User centred
  - Item centred
- Collaborative Filtering methods
  - Memory based
    - User-User
    - Item-Item
  - Model based
    - Matrix factorisation
- Hybrid methods

Data required:
User data (Demographics, CLV Engagement history, etc) (and/or) **Item data** (Pricing, brand, quantity, form, color, make, etc)

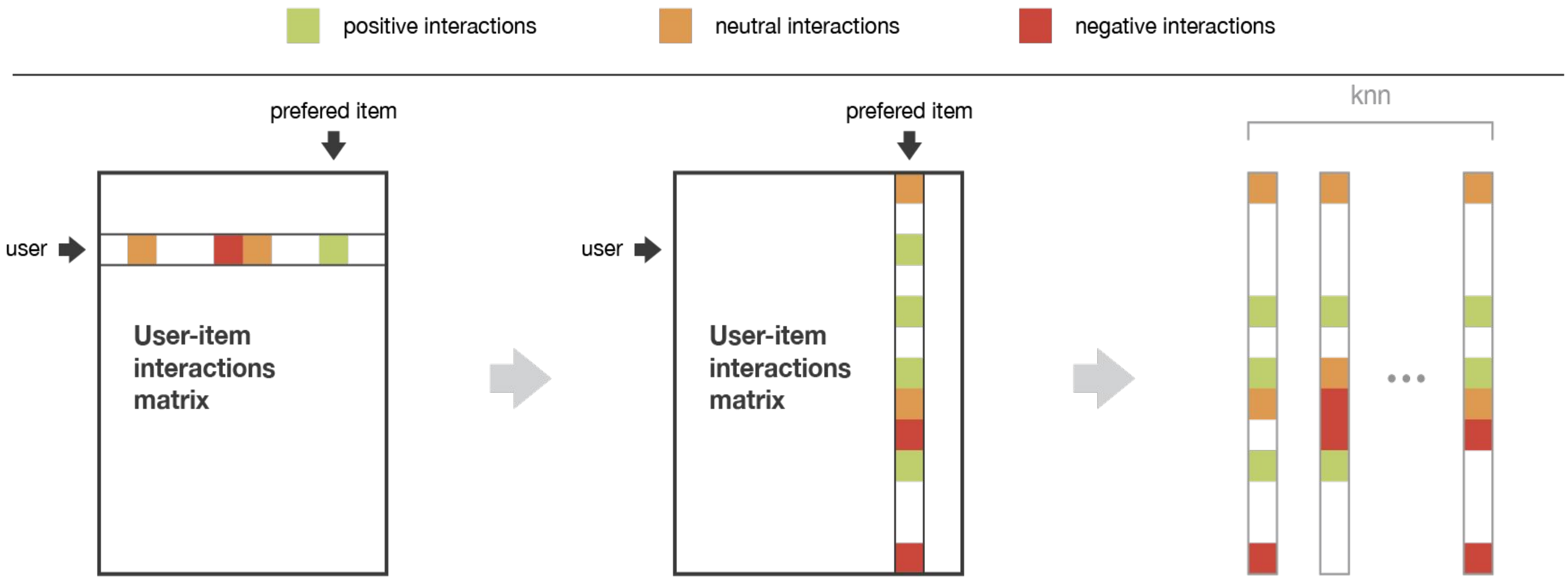
Data required:
User - Item Interaction matrix

Data required:
Both User - Item Interaction matrix as well as User data / Item data

Collaborative Filtering Methods: User-User Interaction



Collaborative Filtering Methods: Item-Item Interaction



We identify the prefered item of user we want to make recommendation for.

The prefered item is represented by its column in the matrix.

We can search and recommend the K nearest items to this “prefered item”

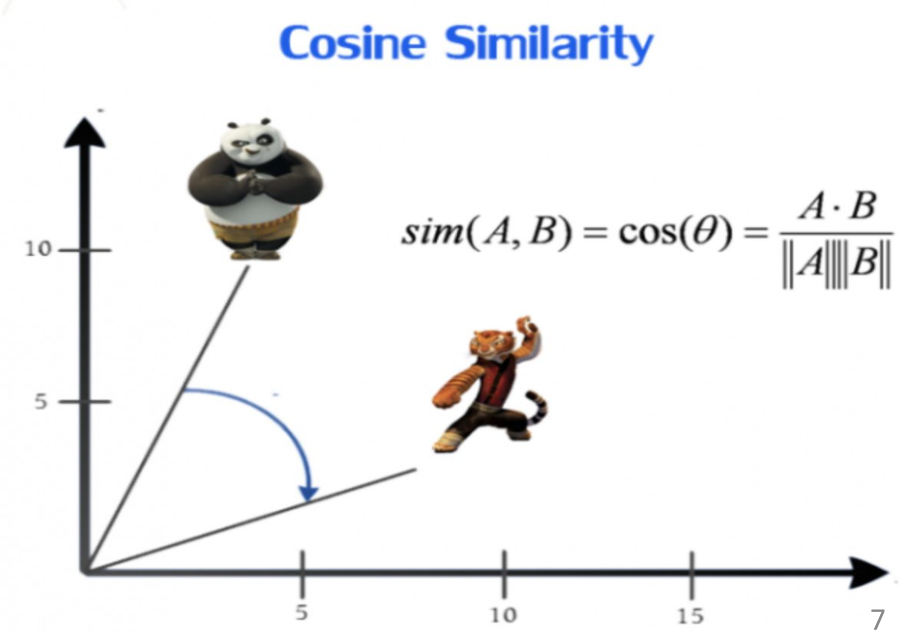
A note on KNN distance metric: Cosine Similarity

⇒ **Similarity between Vectors:** Calculates the cosine of the angle between 2 vectors

⇒ It is generally used as a metric for measuring distance when the magnitude of the vectors does not matter but rather the orientation

⇒ **Low Complexity** especially for sparse vectors

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$



Collaborative Filtering Methods: Pros and Cons

Advantages:

⇒ Domain agnostic: Requires no information about users or items
⇒ As more data becomes available, the recommendations become highly personalized
⇒ Novelty: Users are give unseen, i.e. new recommendations

Disadvantages:

⇒ **"Cold Start Problem"**: Impossible to recommend **item** to **new users** or to recommend a **new item** to **existing users.**
⇒ "**Rich gets richer**": Products that are rated well get recommended

Collaborative Filtering Methods: Comparison

USER - USER METHOD	ITEM - ITEM METHOD	MATRIX FACTORISATION
● Based on the search of similar users in terms of interactions with items. ● **Less Robust**: As every user has only interacted with a few items, the method is sensitive to any recorded interactions. ● **Low Bias:** As the final recommendation is only based on interactions recorded for users similar to our user of interest, we obtain more personalized results. ● Not easily scalable	● Based on the search of similar items in terms of user-item interactions. ● **More Robust**: As a lot of users have interacted with an item, the method is less sensitive to single interactions. ● **High Bias**: Interactions coming from every kind of users are then considered in the recommendation, making the method less personalised. ● Not easily scalable	● Based on the dot product of latent factor matrices of items and users. ● As the factor vectors are learned and not given by us, extracted features taken individually have only a mathematical meaning but **no intuitive interpretation** ● Is much more **scalable** than the other two methods since it deals with denser matrices

So which ~~one~~ do we use?

One size does not fit all!

Choosing a single recommendation system: Some possible issues

- Different models perform better in different circumstances

- Businesses have different expectation out of different Users

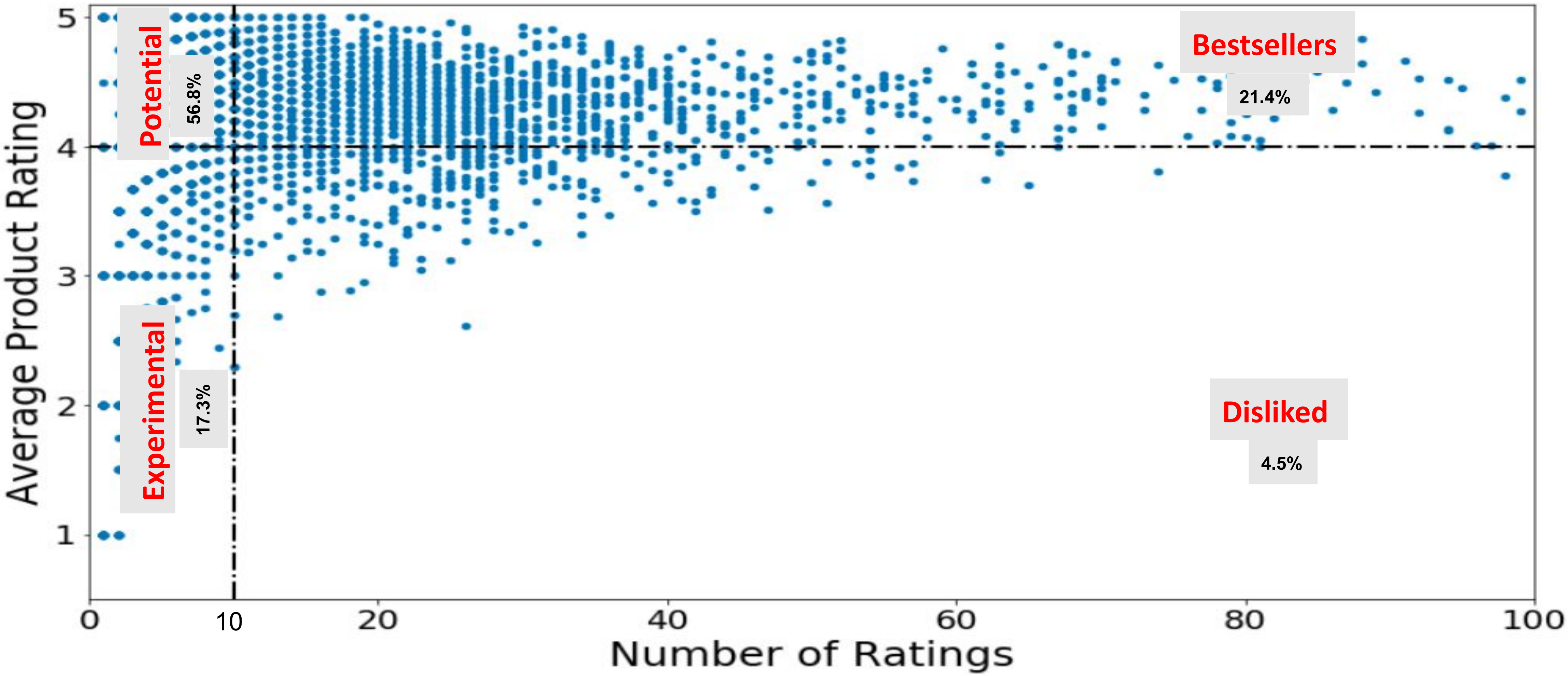
- Some Products are better and some are new (based on rating)

- Also, remember the "Cold start problem"?

Using a single type of recommendation system does not always make sense

Understanding Product and User segmentation

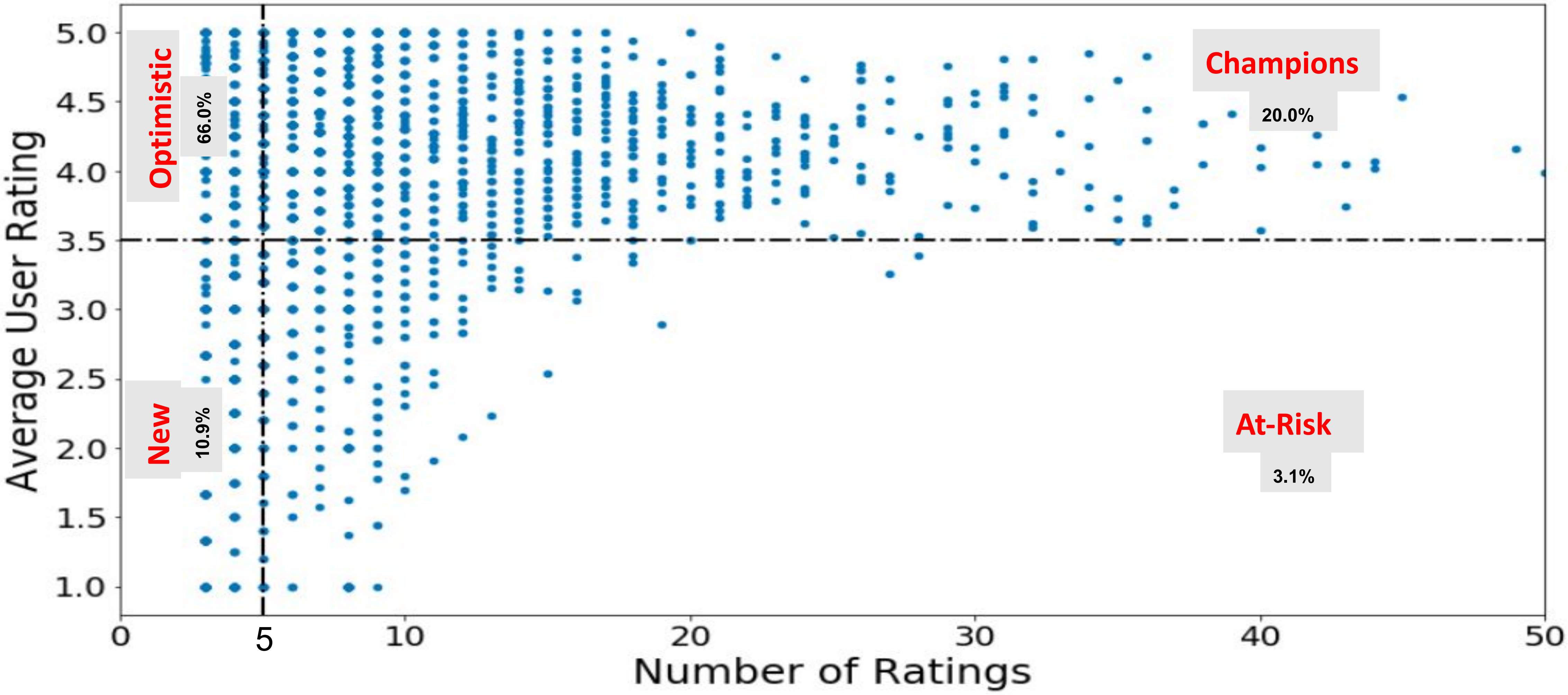
Product Segmentation



Total products: 8,308

* Graph omits products with # ratings > 200 and average rating < 2.5 for brevity

User Segmentation

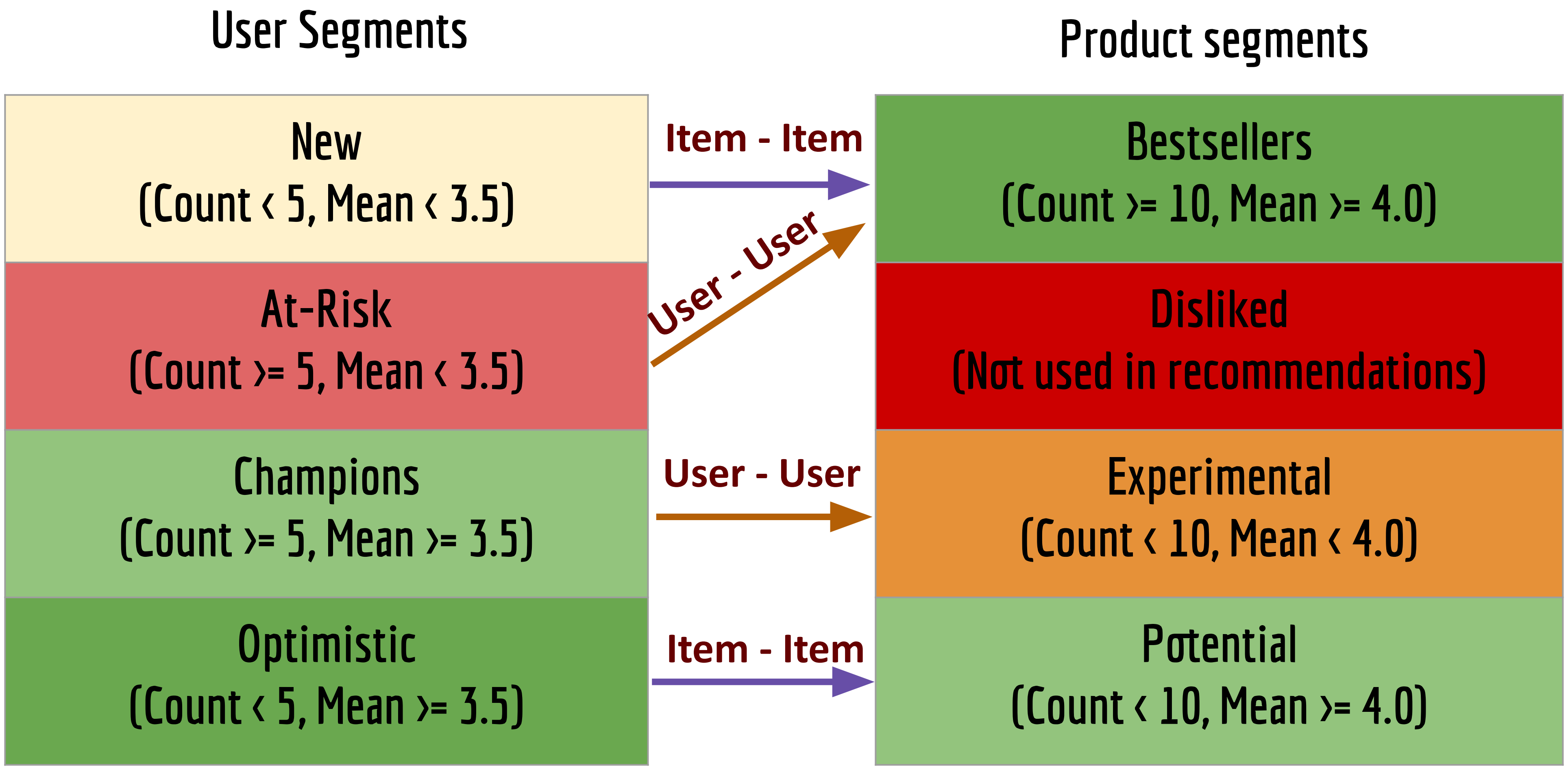


Total users: 19,748

* Graph omits users with # ratings > 40 for brevity

Our Recommendation System
Le’CUMIN

Addressing differences in segments: Targeted recommendations



Recommendations for an At-risk user

Average rating = 3.2 , Number of ratings = 30

An At-risk user's top rated product:

Given Rating: 5.0

Similar to items you have liked

Product average rating: 4.8
Number of ratings: 53

Product average rating: 4.5
Number of ratings: 24

Product average rating:4.9
Number of ratings: 20

Our model is bringing up the Bestselling products for the "At-risk user"

Recommendations for an Optimistic user

Average rating = 4.6 , Number of ratings = 3

An Optimistic user's top rated product:

Given Rating: 5.0

Similar to items you have liked

Product average rating: 4.2
Number of ratings: 7

Product average rating: 4.1
Number of ratings: 2

Product average rating:4.4
Number of ratings: 4

Our model is bringing up potential products for the "Optimistic user"

Recommendations for a Champion user

Average rating = 4.9 , Number of ratings = 14

A Champion user's top rated product:

Given Rating: 5.0

Similar to items you have liked

Product average rating: 3.7
Number of ratings: 4

Product average rating: 3.5
Number of ratings: 2

Product average rating:4.1
Number of ratings: 2

Our model is bringing up the Experimental products for the "Champion user"

Product Coverage: Improvement through our Business rules

> ***Coverage :***
>
> Percentage of items included in the recommendation list over the total number of items.

Simulation of users - products

We create **500 (user - item) possibilities** with the Users following the distribution in our data (New, at-risk, champion, optimistic)

At-Risk

Item-Item recommender

Our recommender

Calculate the coverage results for each product segment

Bestsellers
Disliked
Experimental
Potential

Coverage results for the recommenders

	Simple Item-Item recommender	Our recommender
Bestsellers	29%	21%
Disliked	23%	0%
Experimental	16%	21%
Potential	17%	19%

Possible next steps

- Adapt the rules to different business contexts (increase of sales, profits, user-base…)
- Implement randomization strategy to reduce cold start problem
- Leverage data on the user and the products to build a new model


CUSTOMERS WHO BOUGHT THIS ITEM:
ALSO BOUGHT:
piccolo
demotivateur.fr

Thank you!

Appendix

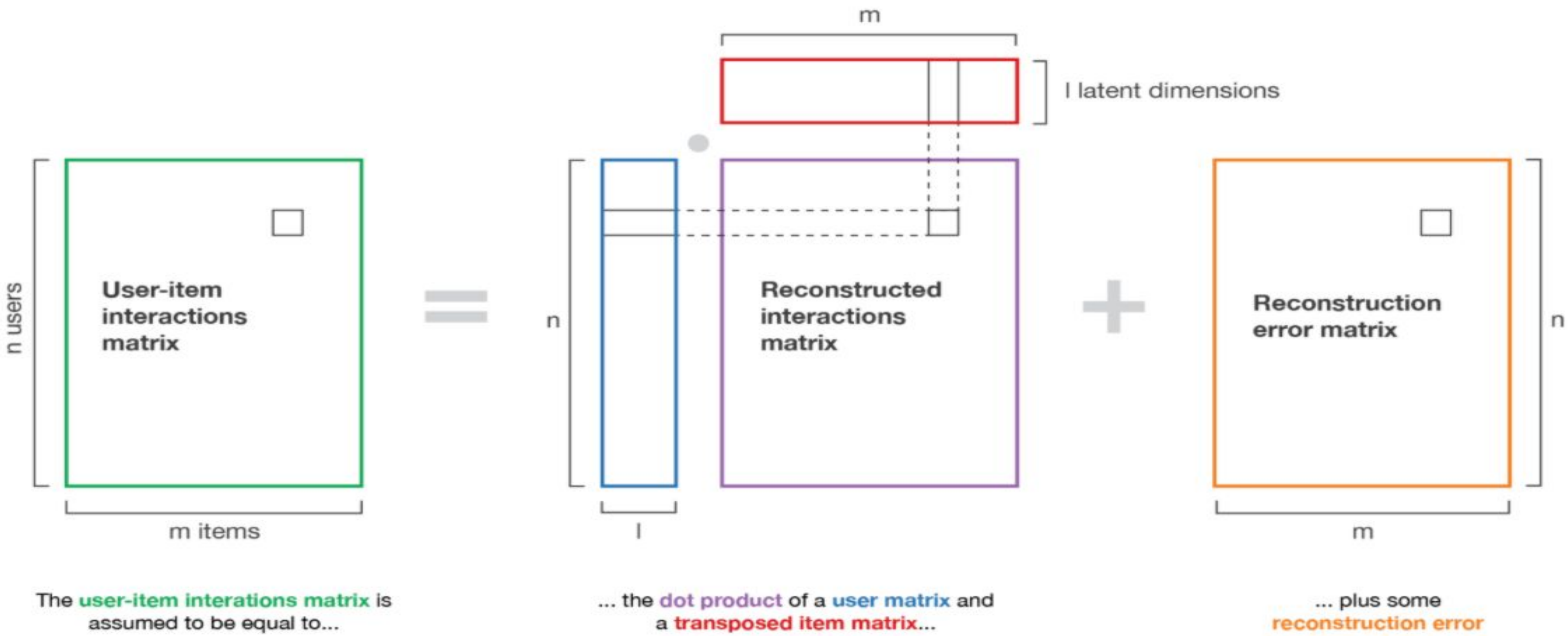
Matrix factorization: Utilizing the sparsity

```
ratings_pivoted[ratings_pivoted["B00004U9V2"]!=0].head(10)
```

product_id	B00004U9V2	B0000532JH	B00005A77F	B00005NDTD	B00005R7ZY	B00005V50B	B00005V50C	B000066SYB	B000068DWY	B00008WFSM
user_id										
A15C2P6JJJQ3YM	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A15FJ6KJUPIX8O	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A1606LA683WZZU	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A177XNPRJ4OQ3R	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A17XGOOKP8RZI5	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A1BA60MYHKMI1Q	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A1CY00CCQC67XX	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A1DNABDMALTASM	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A1J3MG8MQF1MPM	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A1OOSEM3O1H389	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

10 rows × 8308 columns

Collaborative Filtering Methods: Matrix factorization



- The sparse interaction matrix is factorized into 2 dense (user and item) matrices using algorithms like SVD and gradient descent
- Prediction for a user is then given by multiplying the user vector by the items vector in order to estimate the corresponding rating.

Product Coverage: Improvement through our Business rules

Coverage :

Percentage of items included in the recommendation list over the total number of items.

Simulation of users looking at products

We create 500 (user - item) possibilities with the Users following the distribution in our data (New, at-risk, champion, optimistic)

At-Risk

Item-Item recommender

Our recommender

Calculate the coverage results for each product segment

Potential	Bestsell
Experim.	Disliked

Coverage results for the recommenders

Simple Item-Item recommender

Total Covered = 22%

18.9%	29%
15.8%	23%

Our recommender

Total Covered = 17%

19%	21%
21%	0%

Possible next steps

- Tweak the threshold for creating user and product segments for different business contexts
- Leverage matrix factorisation to recreate the user-item interaction matrix
- Build a content based model and then a hybrid model that combines both