

How to Design and Build a Recommendation Pipeline in Python

Jill Cates
November 10th, 2018
PyCon Canada

Spotify



Discover Weekly

MADE FOR JILL

Discover Weekly

Your weekly mixtape of fresh music. Enjoy new discoveries and deep cuts chosen just for you. Updated every Monday, so save your...

Made for Jill Cts by Spotify • 30 songs, 1 hr 47 min

[PLAY](#) [FOLLOWING](#) [...](#)

Filter [Download](#)

TITLE	ARTIST	DATE
+ The Weekend - Funk Wav Remix	SZA, Calvin H...	3 days ago
+ You Say	Ehrling	3 days ago
+ Grow Up	Bolier	3 days ago

Netflix



“Because you watched
this TV show...”

Because you watched Bloodline



Because you watched Orange Is the New Black



Because you watched House of Cards



Amazon

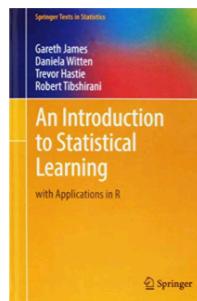


“Frequently bought together”

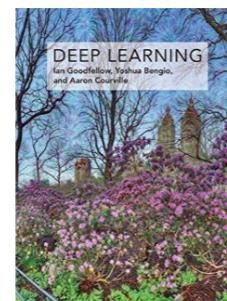
“Customers who bought this item also bought”

Customers who bought this item also bought

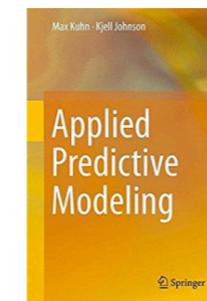
Page 1 of 17



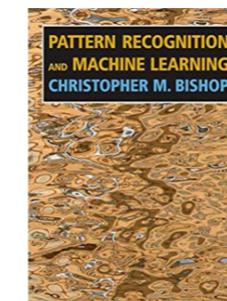
An Introduction to Statistical Learning: with Applications in R
Gareth James
★★★★★ 13
Hardcover
CDN\$ 77.35 ✓prime



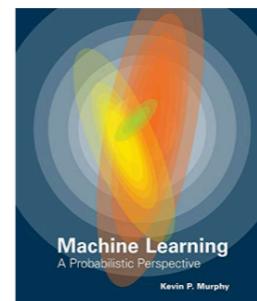
Deep Learning
Ian Goodfellow
★★★★★ 26
Hardcover
CDN\$ 81.36



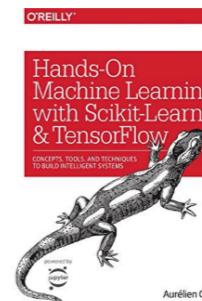
Applied Predictive Modeling
Max Kuhn
★★★★★ 8
Hardcover
CDN\$ 97.10 ✓prime



Pattern Recognition and Machine Learning
Christopher M. Bishop
★★★★★ 8
Hardcover
CDN\$ 86.14 ✓prime



Machine Learning: A Probabilistic Perspective
Kevin P. Murphy
★★★★★ 6
Hardcover
CDN\$ 120.49 ✓prime



Hands-On Machine Learning with Scikit-Learn and TensorFlow:...
Aurélien Géron
★★★★★ 27
Paperback
CDN\$ 45.06 ✓prime



OkCupid



“Finding your best match”

The image shows a mobile application interface for OkCupid's quiz feature. At the top, there is a navigation bar with three circular icons: the first is blue with a checkmark, the second is white with the number '2', and the third is white with the number '3'. To the right of these icons is the text "Answer 7 questions to calculate your best matches." and a "Skip >" button. Below the navigation bar is a large white rectangular box containing the text "1 of 7" at the top center. Inside this box is a question: "Are you a morning person?". At the bottom of this box is a "⟳ Next Question" button. Below the white box is a blue background area with two large buttons: an orange "No" button on the left and a green "Yes" button on the right.

1 of 7

Are you a morning person?

⟳ Next Question

No Yes

Recommender Systems in the Wild



Spotify

Discover Weekly



Amazon

Customers who bought
this item also bought



Netflix

Because you
watched this show...



LinkedIn

Jobs recommended for you



OkCupid

Finding your best match



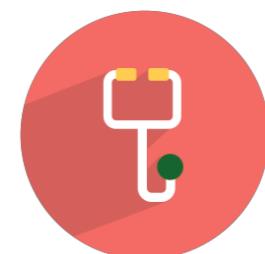
New York Times

Recommended
Articles for You



GitHub

Repos “based on
your interest”



Medicine

Facilitating clinical
decision making

Before e-commerce

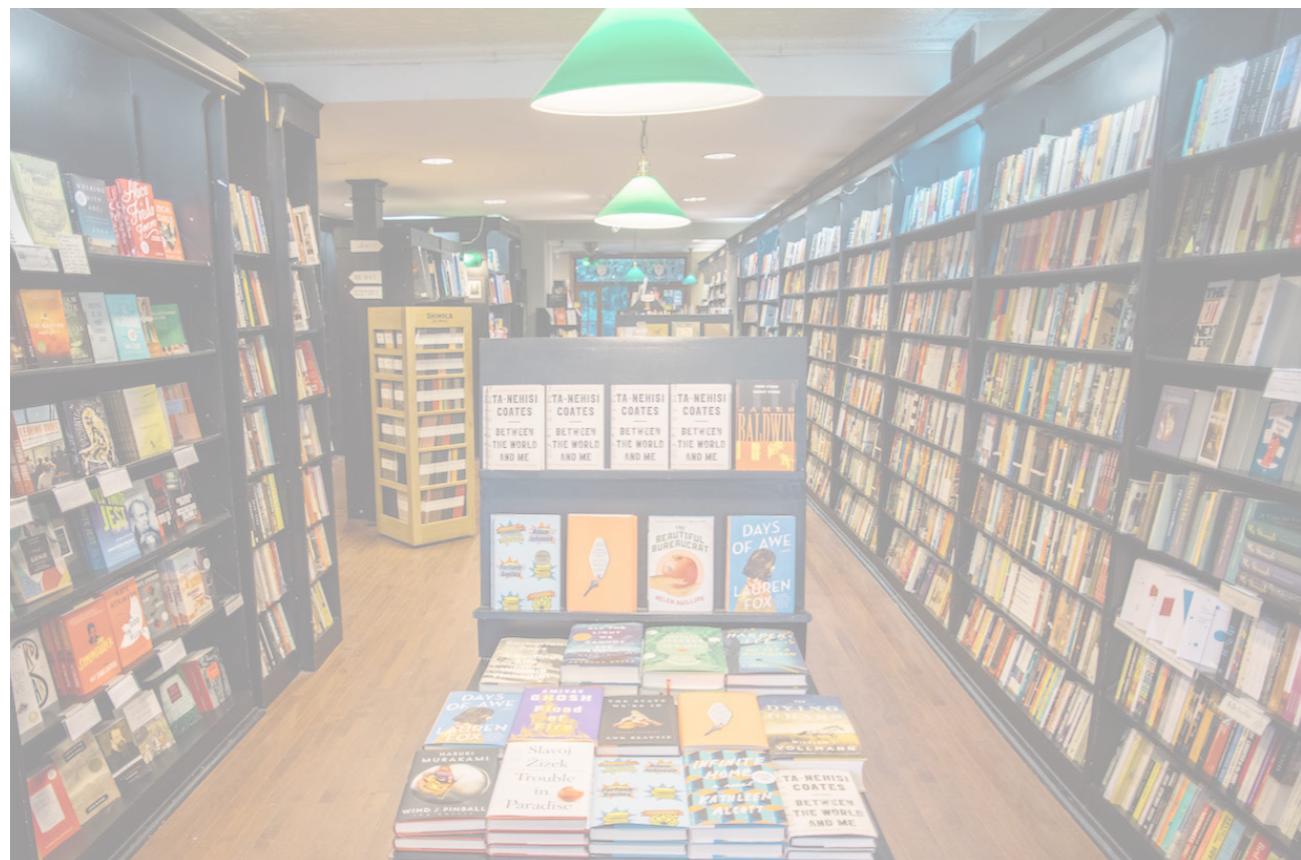
Things were sold exclusively in
brick-and-mortar stores...



limited inventory

mainstream products

Before e-commerce



Things were sold exclusively in
brick-and-mortar stores...

limited inventory

mainstream products

E-commerce

Books Advanced Search Today's Deals New Releases Amazon Charts Best Sellers & More The Globe & Mail Best Sellers New York Times Best Sellers Best Books of the Month Children's Books

1-60 of over 5,000 results for Books : "python"

Sort by | Featured

Books Programming Languages Textbooks Programming Computer Science & Information Systems Textbooks Computer Programming for Beginners Computers & Technology Object-Oriented Software Design Textbooks Artificial Intelligence Textbooks Graphics & Visualization Textbooks Game Programming See more

Bestseller

Get FREE One-Day Delivery on qualifying orders over CDNS 25

Show results for Any Category Books Programming Languages Textbooks Programming Computer Science & Information Systems Textbooks Computer Programming for Beginners Computers & Technology Object-Oriented Software Design Textbooks Artificial Intelligence Textbooks Graphics & Visualization Textbooks Game Programming

Refine by

Amazon Prime

Delivery Date

Author

Avg. Customer Review

New Releases

Availability

Books Programming Languages Textbooks Programming Computer Science & Information Systems Textbooks Computer Programming for Beginners Computers & Technology Object-Oriented Software Design Textbooks Artificial Intelligence Textbooks Graphics & Visualization Textbooks Game Programming

Bestseller

Get FREE One-Day Delivery on qualifying orders over CDNS 25

Python (2nd Edition): Learn Python in One Day and Learn It Well. Python for Beginners with Hands-on Project. (Learn Coding Fast with Project) by Mark Lutz

CDNS 39.49 - CDNS 52.30

prime | FREE One-Day Paperback, Kindle Edition

★★★★★ 24

Python Crash Course: A Hands-On, Project-Based Introduction to Programming by Eric Matthes

CDNS 27.99 - CDNS 44.23

prime | FREE One-Day Paperback, Kindle Edition

★★★★★ 2

Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython by Wes McKinney

CDNS 42.74 - CDNS 52.93

prime | FREE One-Day Paperback, Kindle Edition

★★★★★ 4

Automate the Boring Stuff with Python: Practical Programming for Total Beginners by Al Sweigart

CDNS 15.37 - CDNS 34.60

prime | FREE One-Day Paperback, Kindle Edition

★★★★★ 19

Bestseller

Get FREE One-Day Delivery on qualifying orders over CDNS 25

Python in One Day and Learn It Well by Mark Lutz

The only book you need to start coding in Python immediately

CDNS 0.00 - CDNS 15.07

prime | FREE One-Day Paperback, Kindle Edition

★★★★★ 12

Fluent Python by Luciano Ramalho

Python (2nd Edition): Learn Python in One Day and Learn It Well. Python for Beginners with Hands-on Project. (Learn Coding Fast with Project) by Mark Lutz

CDNS 39.49 - CDNS 52.30

prime | FREE One-Day Paperback, Kindle Edition

★★★★★ 5

Python Cookbook by Luciano Ramalho

Python Cookbook: Recipes for Mastering Python 3 by Brian K. Jones

CDNS 31.97 - CDNS 42.12

prime | FREE One-Day Paperback, Kindle Edition

★★★★★ 3

Deep Learning with Python by Ian Goodfellow

Deep Learning with Python by Ian Goodfellow

CDNS 21.88 - CDNS 49.99

prime | FREE One-Day Paperback, Audio Download

★★★★★ 2

Bestseller

Get FREE One-Day Delivery on qualifying orders over CDNS 25

Learn PYTHON 3 the HARD WAY: A Simple Introduction to the Terrifyingly Beautiful World of Computers and Code by Zed Shaw

CDNS 17.27 - CDNS 41.34

prime | FREE One-Day Paperback, Kindle Edition

★★★★★ 5

Python for Finance: Mastering Data-Driven Finance by Yves Hilpisch

Python for Finance: Mastering Data-Driven Finance by Yves Hilpisch

CDNS 65.68

prime | FREE One-Day Paperback, Kindle Edition

★★★★★ 5

Head First Python by Paul Barry

Head First Python: A Brain-Friendly Guide by Paul Barry

CDNS 37.00 - CDNS 38.95

prime | FREE One-Day Paperback, Kindle Edition

★★★★★ 1

Impractical Python Projects: Playful Programming Activities to Make You Smarter by Matt Bernier

Impractical Python Projects: Playful Programming Activities to Make You Smarter by Matt Bernier

CDNS 17.99 - CDNS 35.01

prime | FREE One-Day Paperback, Kindle Edition

★★★★★ 1

unlimited inventory

niche products

The Tasting Booth Experiment

When Choice is Demotivating: Can One Desire Too Much
of a Good Thing?

Sheena S. Iyengar
Columbia University

Mark R. Lepper
Stanford University

6 jam samples



VS.

24 jam samples



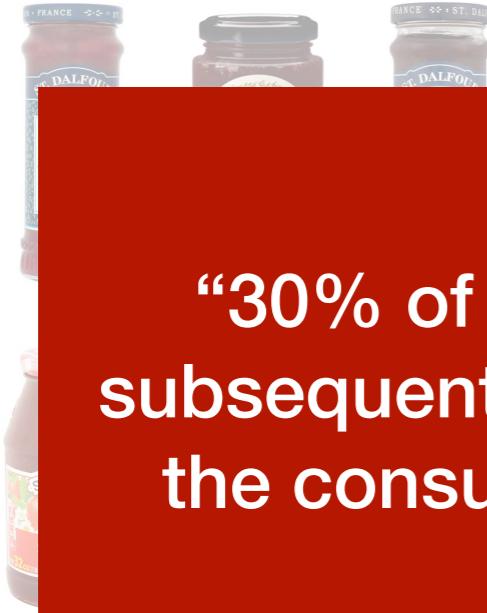
The Tasting Booth Experiment

When Choice is Demotivating: Can One Desire Too Much
of a Good Thing?

Sheena S. Iyengar
Columbia University

Mark R. Lepper
Stanford University

6 jam samples



24 jam samples



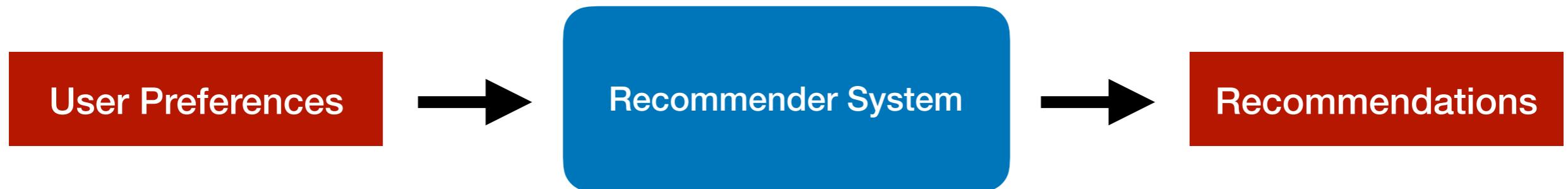
“30% of the consumers in the limited-choice condition subsequently purchased a jar of jam; in contrast, only 3% of the consumers in the extensive-choice condition did so”



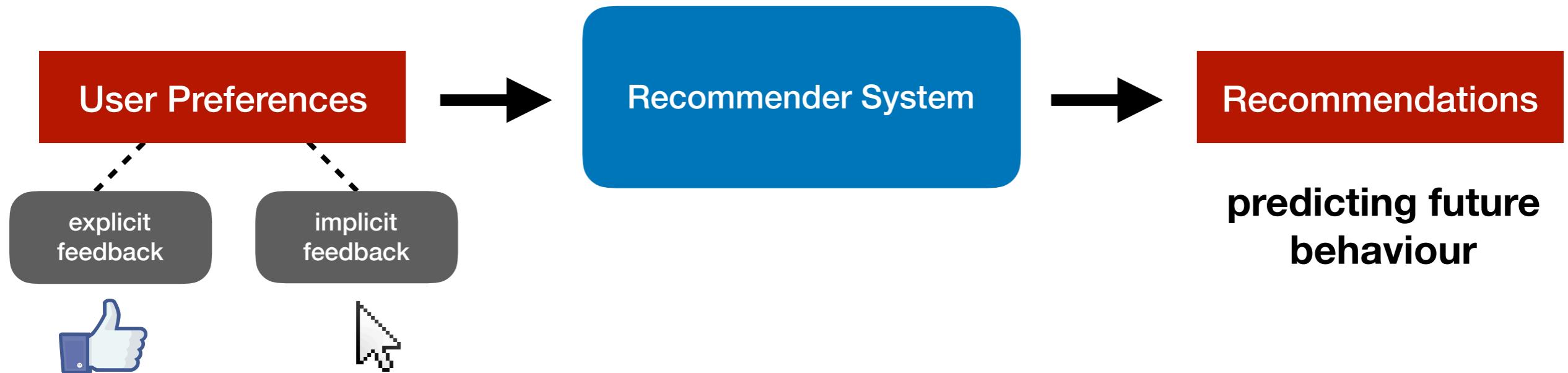
Recommender Crash Course



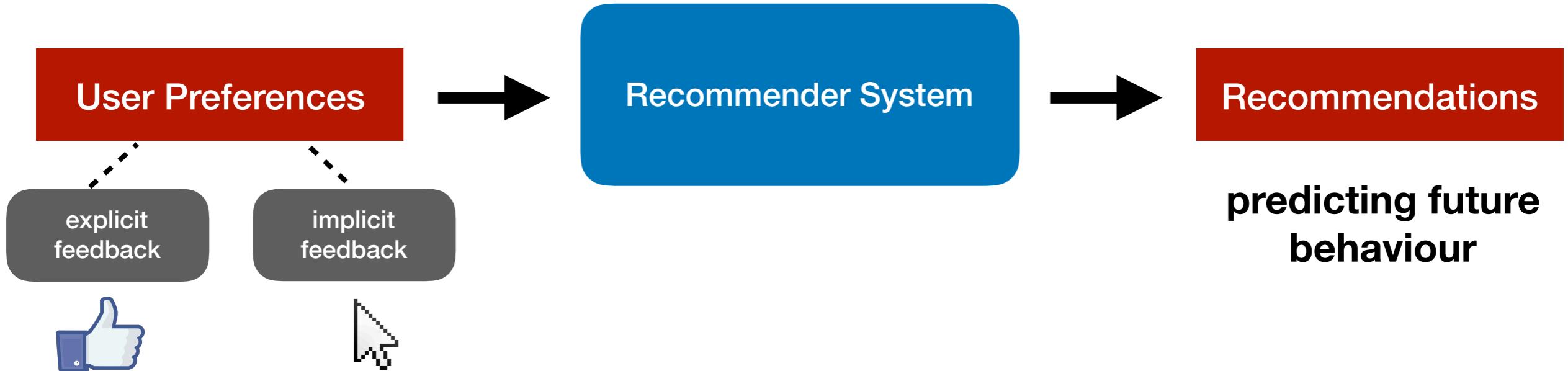
Recommender Crash Course



Recommender Crash Course



Recommender Crash Course

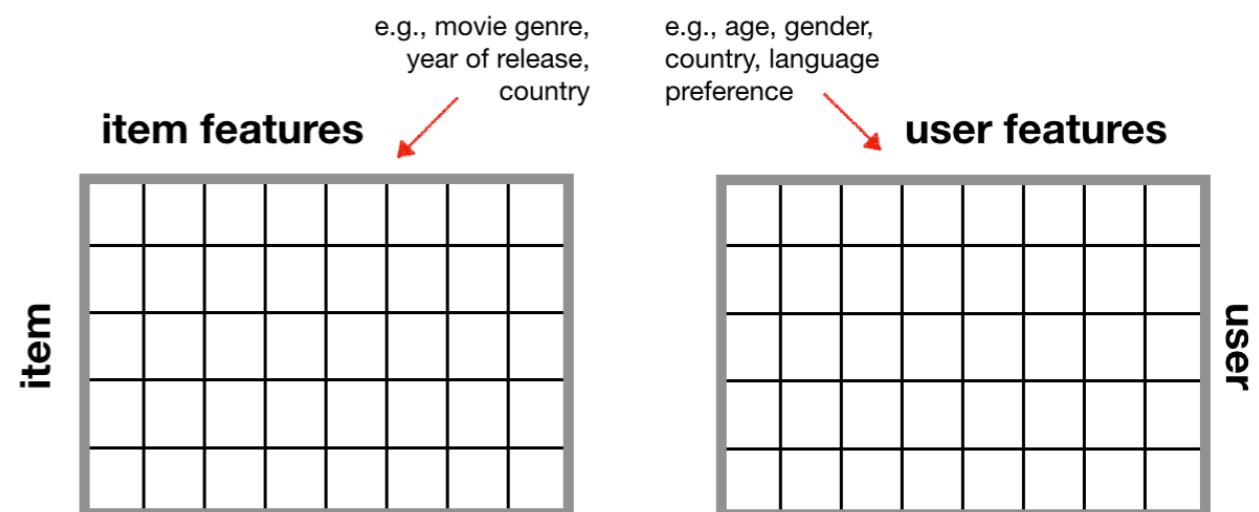


Collaborative filtering

user	John	Erica	Anne	Liz	Jim	
John	1	3		5	5	
Erica		5	2	4	4	5
Anne	5	2	1	4		2
Liz	3	4	3	4	5	
Jim	5	2	1	4	4	3

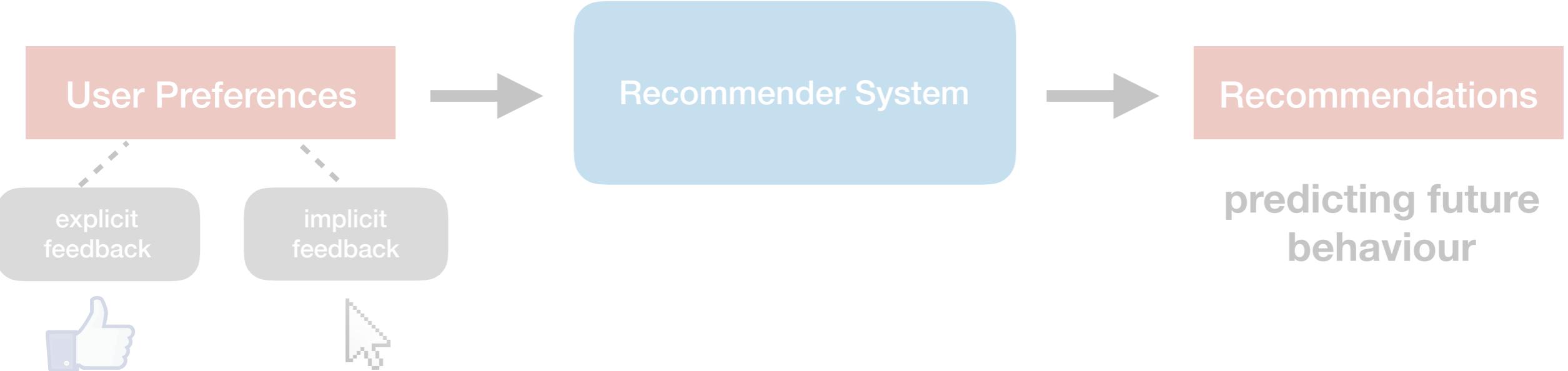
similar users like similar things

Content-based filtering

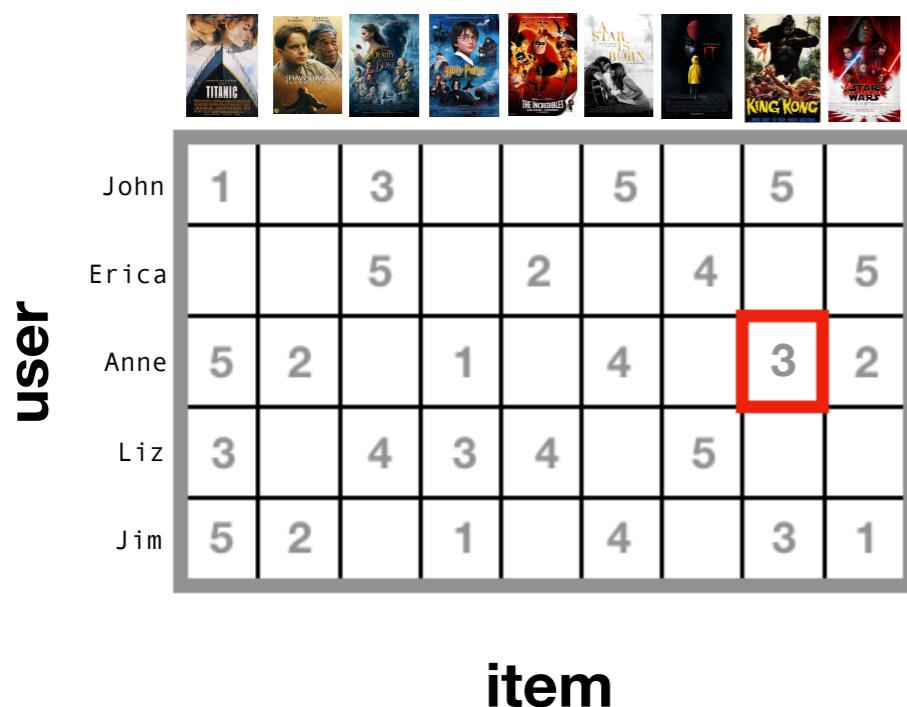


considers items/users features

Recommender Crash Course



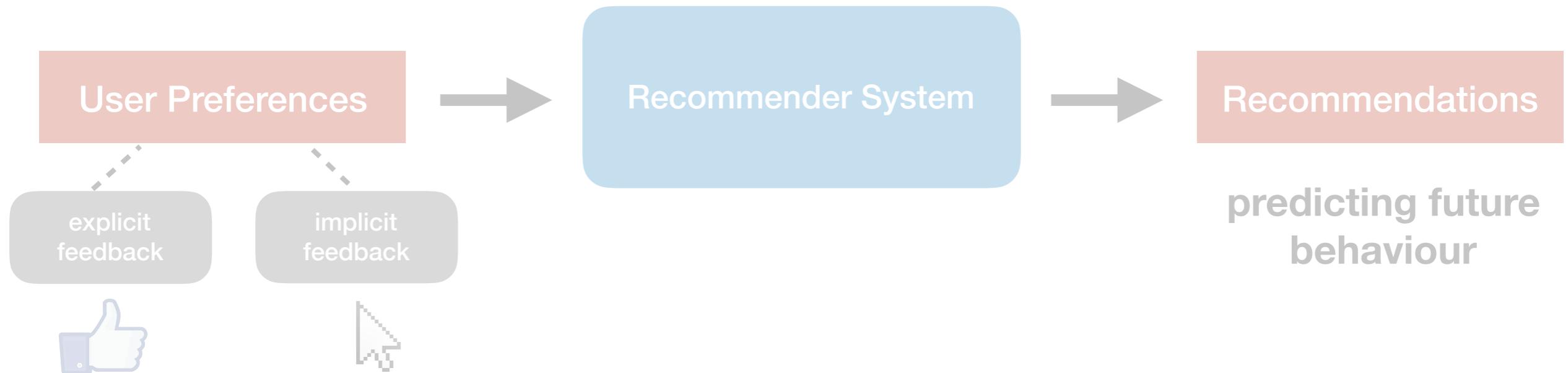
Collaborative filtering



similar users like
similar things

- “Because you watched Movie X”
- “Customers who bought this item also bought”

Recommender Crash Course



Content-based filtering

users	age	gender	country	lang	kids?	religion	items
	scary	funny	family	anime	drama	indie	
John							
Erica							
Anne							
Liz							
Jim							

user and item features

- user features: age, gender, spoken language
- item features: movie genre, year of release, cast

Overview of the Recommender Pipeline

1. Pre-processing
2. Hyperparameter Tuning
3. Model Training and Prediction
4. Post-processing
5. Evaluation

Step 1: Data Pre-processing

user_id	movie_id	rating
2	439	4.0
10	368	4.5
14	114	5.0
19	371	1.0
2	371	3.0
19	114	4.5
3	439	3.5
54	421	2.0
32	114	3.0
10	369	1.0

Pre-processing

Hyperparameter
Tuning

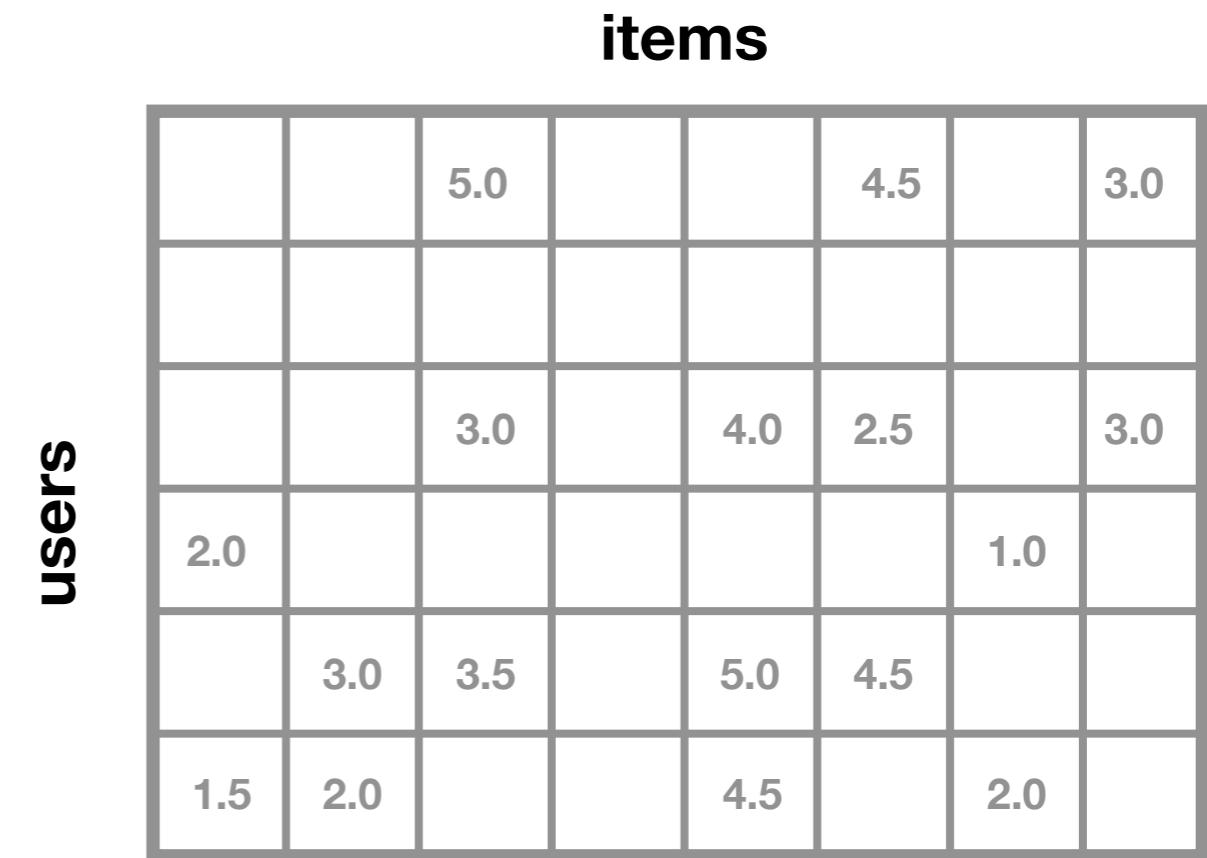
Model Training

Post-processing

Evaluation

Step 1: Data Pre-processing

user_id	movie_id	rating
2	439	4.0
10	368	4.5
14	114	5.0
19	371	1.0
2	371	3.0
19	114	4.5
3	439	3.5
54	421	2.0
32	114	3.0
10	369	1.0



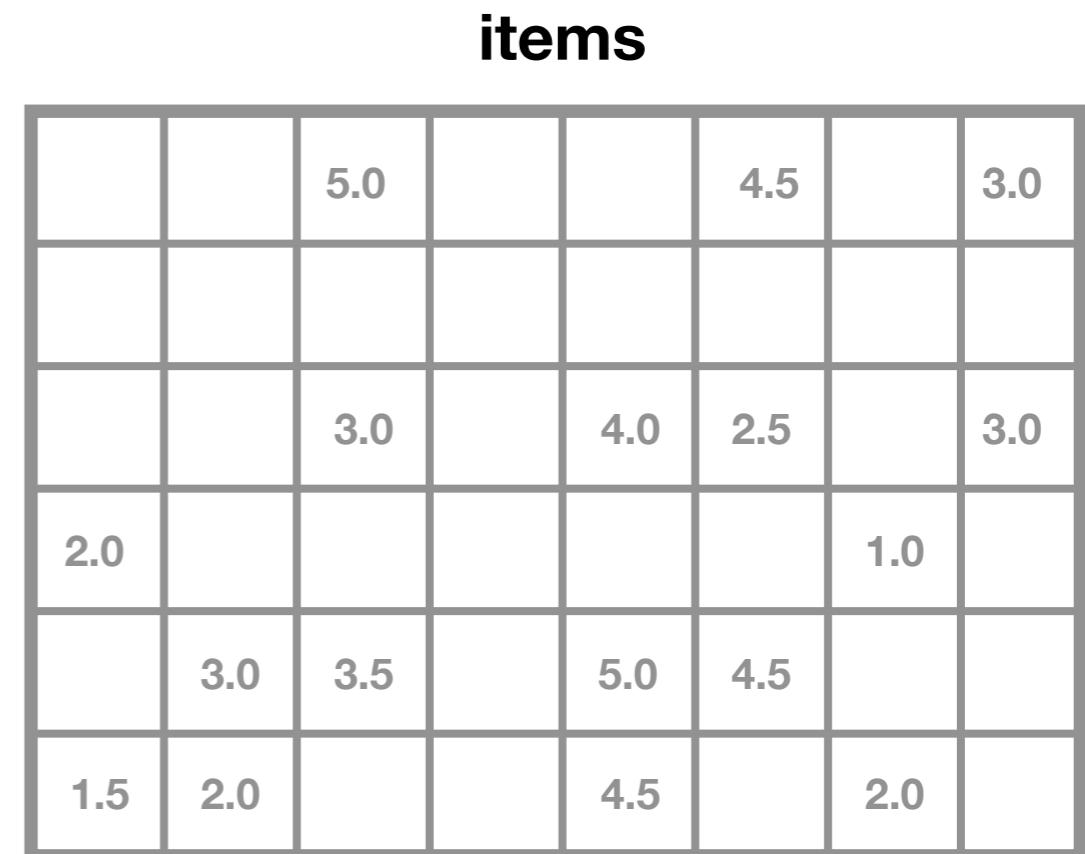
Transform original data to user-item (utility) matrix

Step 1: Data Pre-processing

user_id	movie_id	rating
2	439	4.0
10	368	4.5
14	114	5.0
19	371	1.0
2	371	3.0
19	114	4.5
3	439	3.5
54	421	2.0
32	114	3.0
10	369	1.0



Users

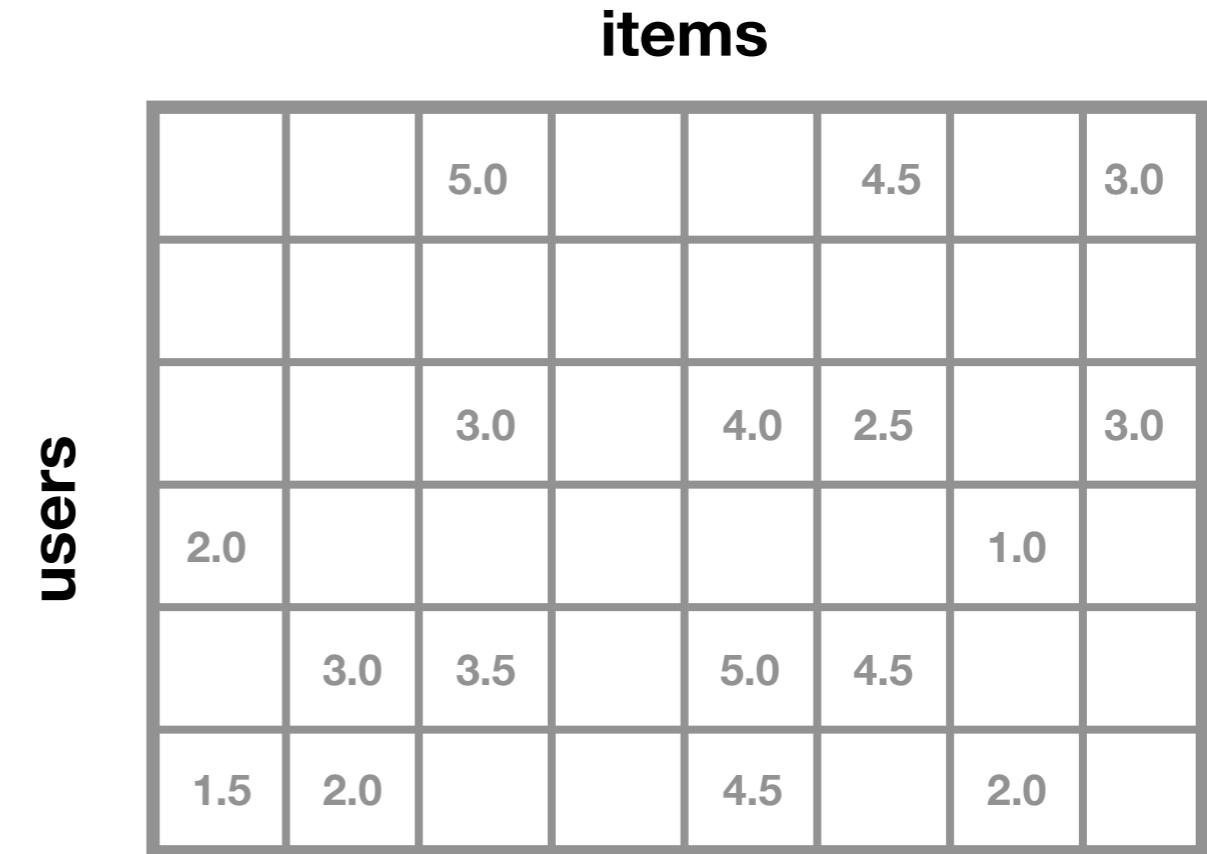


scipy.sparse.csr_matrix



Step 1: Data Pre-processing

user_id	movie_id	rating
2	439	4.0
10	368	4.5
14	114	5.0
19	371	1.0
2	371	3.0
19	114	4.5
3	439	3.5
54	421	2.0
32	114	3.0
10	369	1.0



Calculate Matrix Sparsity

$$\text{sparsity} = \frac{\# \text{ ratings}}{\text{total } \# \text{ elements}}$$

Step 1: Data Pre-processing

Normalization

- Optimists → rate everything 4 or 5
- Pessimists → rate everything 1 or 2
- Need to normalize ratings by accounting for user and item bias
- Mean normalization
 - subtract b_i from each user's rating for given item i

$$b_{ui} = \mu + b_i + b_u$$

Annotations for the equation:

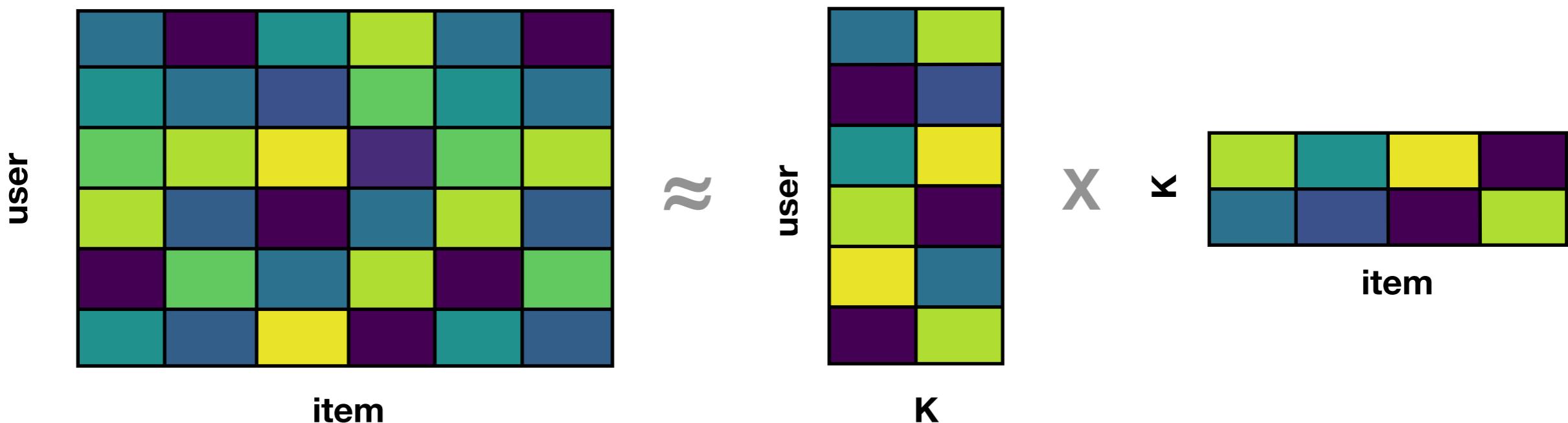
- A red arrow points to b_{ui} with the label "user-item rating bias".
- A red arrow points to μ with the label "global avg".
- A red arrow points to b_i with the label "item's avg rating".
- A red arrow points to b_u with the label "user's avg rating".

Pick a Model

Matrix Factorization

- factorize the user-item matrix to get 2 latent factor matrices:
 - user-factor matrix
 - item-factor matrix
- missing ratings are predicted from the inner product of these two factor matrices

$$X_{mn} \approx P_{mk} \times Q_{nk}^T = \hat{X}$$

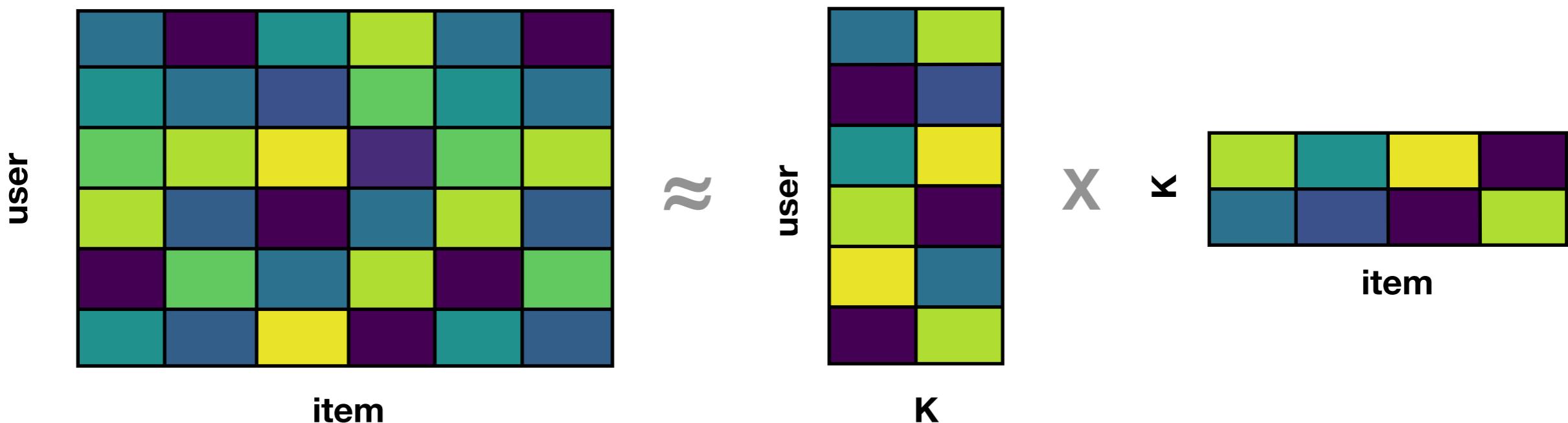


Pick a Model

Matrix Factorization

- Algorithms that perform matrix factorization:
 - Alternating Least Squares (ALS)
 - Stochastic Gradient Descent (SGD)
 - Singular Value Decomposition (SVD)

$$X_{mn} \approx P_{mk} \times Q_{nk}^T = \hat{X}$$

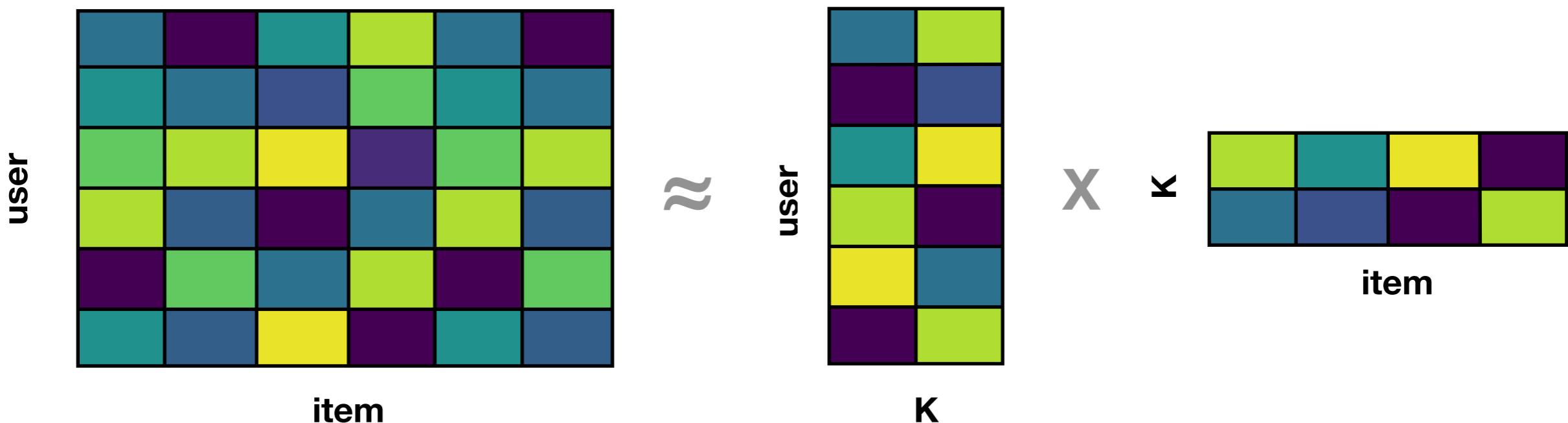


Pick a Model

Matrix Factorization

- Algorithms that perform matrix factorization:
 - Alternating Least Squares (ALS)
 - Stochastic Gradient Descent (SGD)
 - Singular Value Decomposition (SVD)

$$X_{mn} \approx P_{mk} \times Q_{nk}^T = \hat{X}$$



Pick an Evaluation Metric

Precision@K

- Of the top K recommendations, what proportion are relevant to the user?

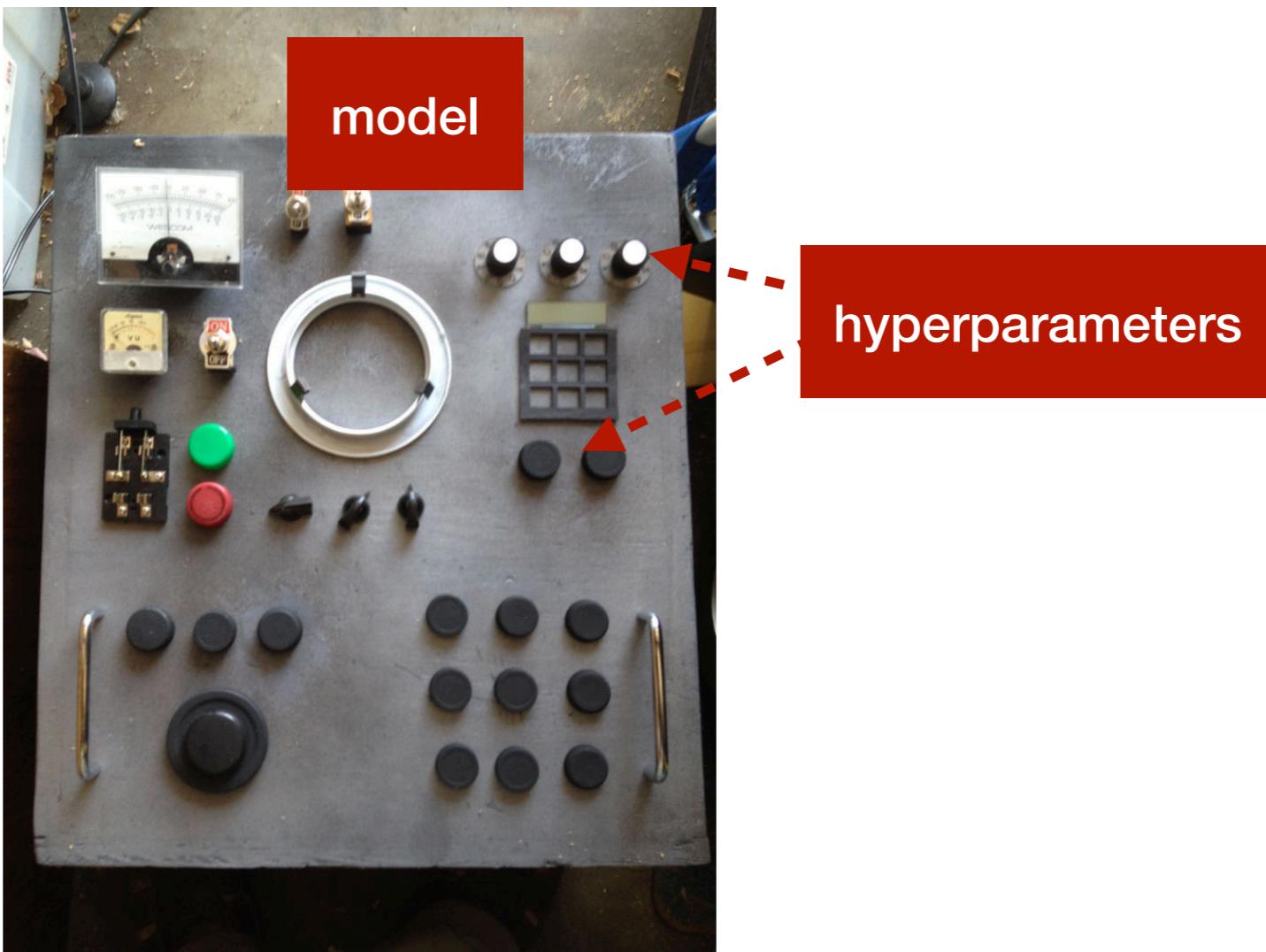
Pick an Evaluation Metric

Precision@10

- Of the top 10 recommendations, what proportion are relevant to the user?

Step 2: Hyperparameter Tuning

What is a hyperparameter?



configuration that is external to the model

Pre-processing

Hyperparameter
Tuning

Model Training

Post-processing

Evaluation

Step 2: Hyperparameter Tuning

Alternating Least Square's Hyperparameters

- k (# of factors)
- λ (regularization parameter)

Goal: find the hyperparameters that give the best precision@10

* (or any other evaluation metric that you want to optimize)

Pre-processing

Hyperparameter
Tuning

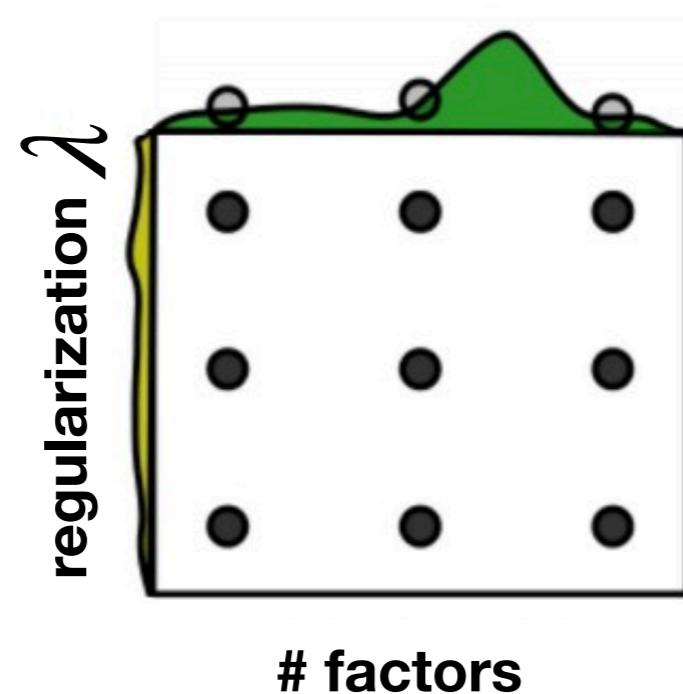
Model Training

Post-processing

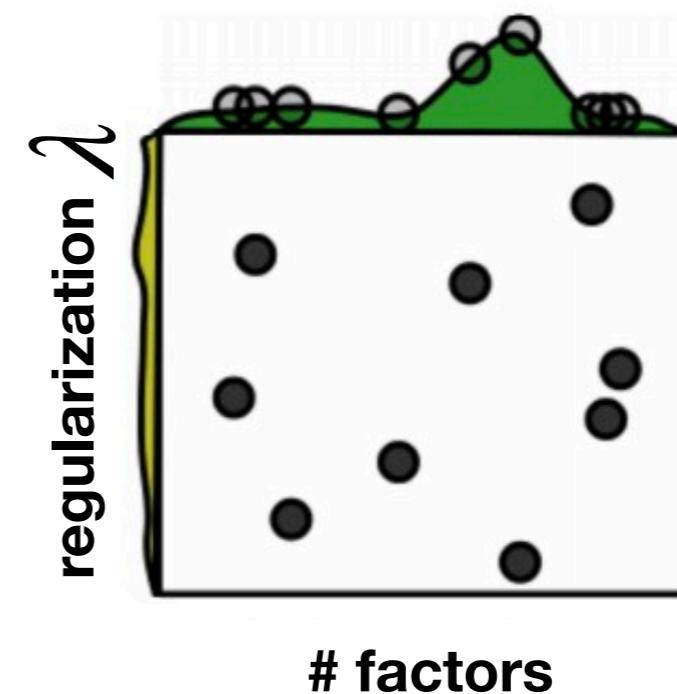
Evaluation

Step 2: Hyperparameter Tuning

Grid Search



Random Search

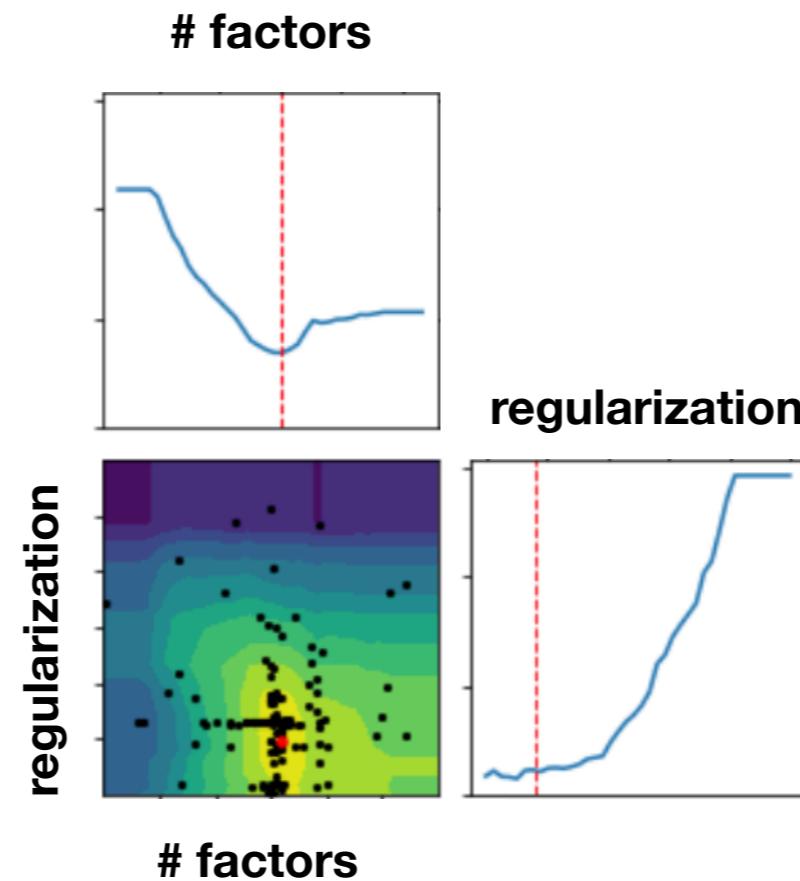


source: blog.kaggle.com

```
sklearn.model_selection.GridSearchCV  
sklearn.model_selection.RandomizedSearchCV
```

Step 2: Hyperparameter Tuning

Sequential Model-Based Optimization



scikit-optimize (skopt)
hyperopt
Metric Optimization Engine (MOE)

Pre-processing

Hyperparameter
Tuning

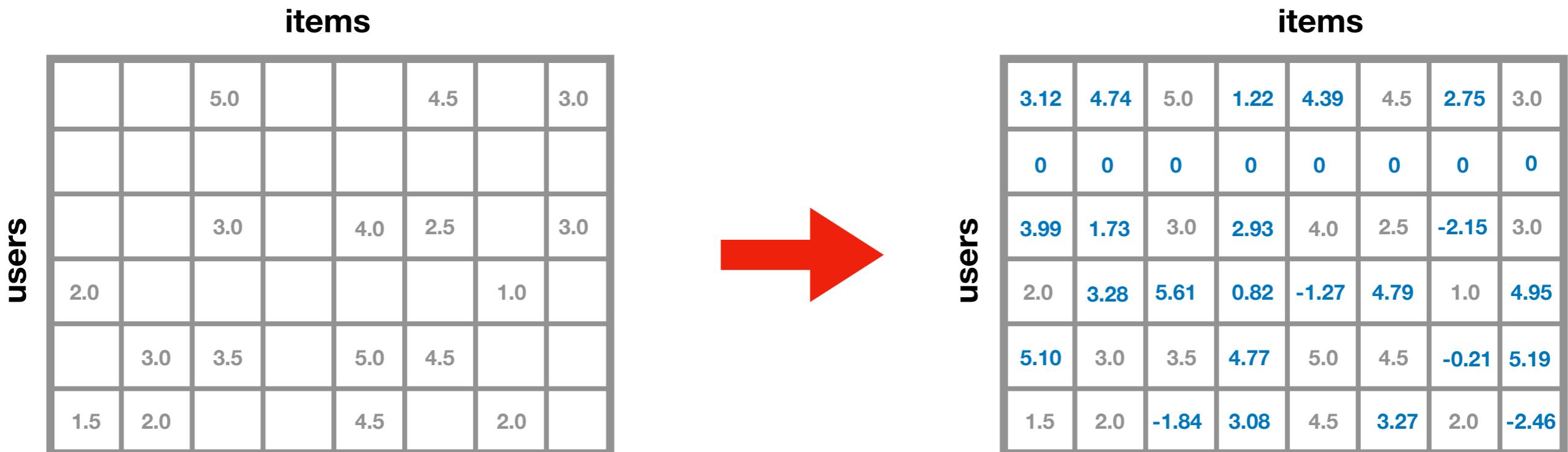
Model Training

Post-processing

Evaluation

Step 3: Model Training

AlternatingLeastSquares(k=8,
regularization=0.001)



Step 4: Post-processing

- Sort recommendations and get top N
- Filter out items that a user has already purchased, watched, interacted with
- Increase diversity of recommendations
- Item-item recommendations
 - Use a similarity metric (e.g., cosine similarity)
 - “Because you watched Movie X”

Pre-processing

Hyperparameter
Tuning

Model Training

Post-processing

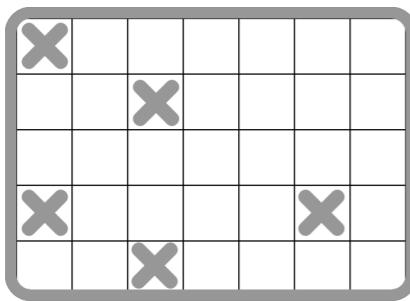
Evaluation

Step 5: Evaluation

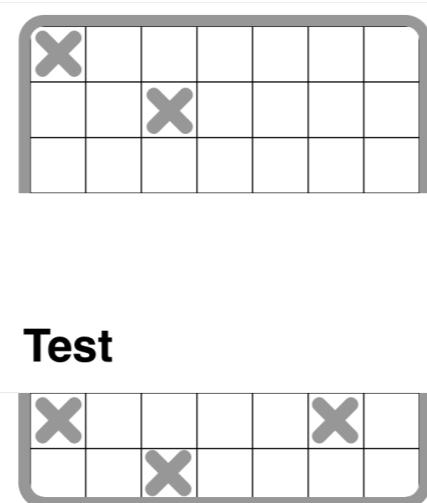
How do we evaluate recommendations?

Traditional ML

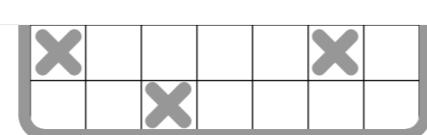
Original



Train

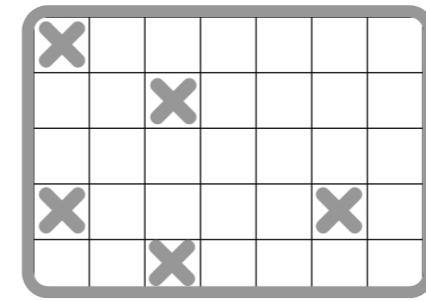


Test

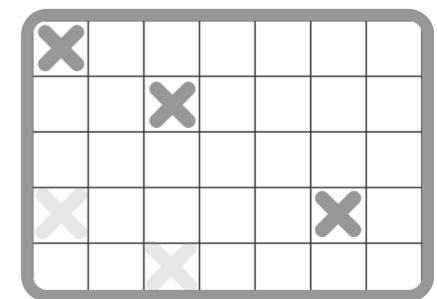


Recommendation Systems

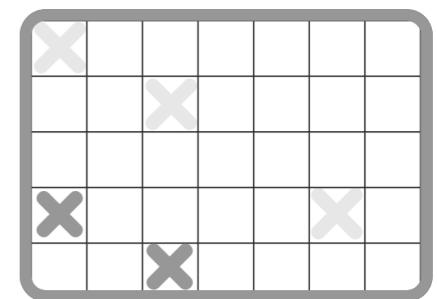
Original



Train



Test



Pre-processing

Hyperparameter
Tuning

Model Training

Post-processing

Evaluation

Step 5: Evaluation

Metrics

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y - \hat{y})^2}{N}}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Pre-processing

Hyperparameter
Tuning

Model Training

Post-processing

Evaluation

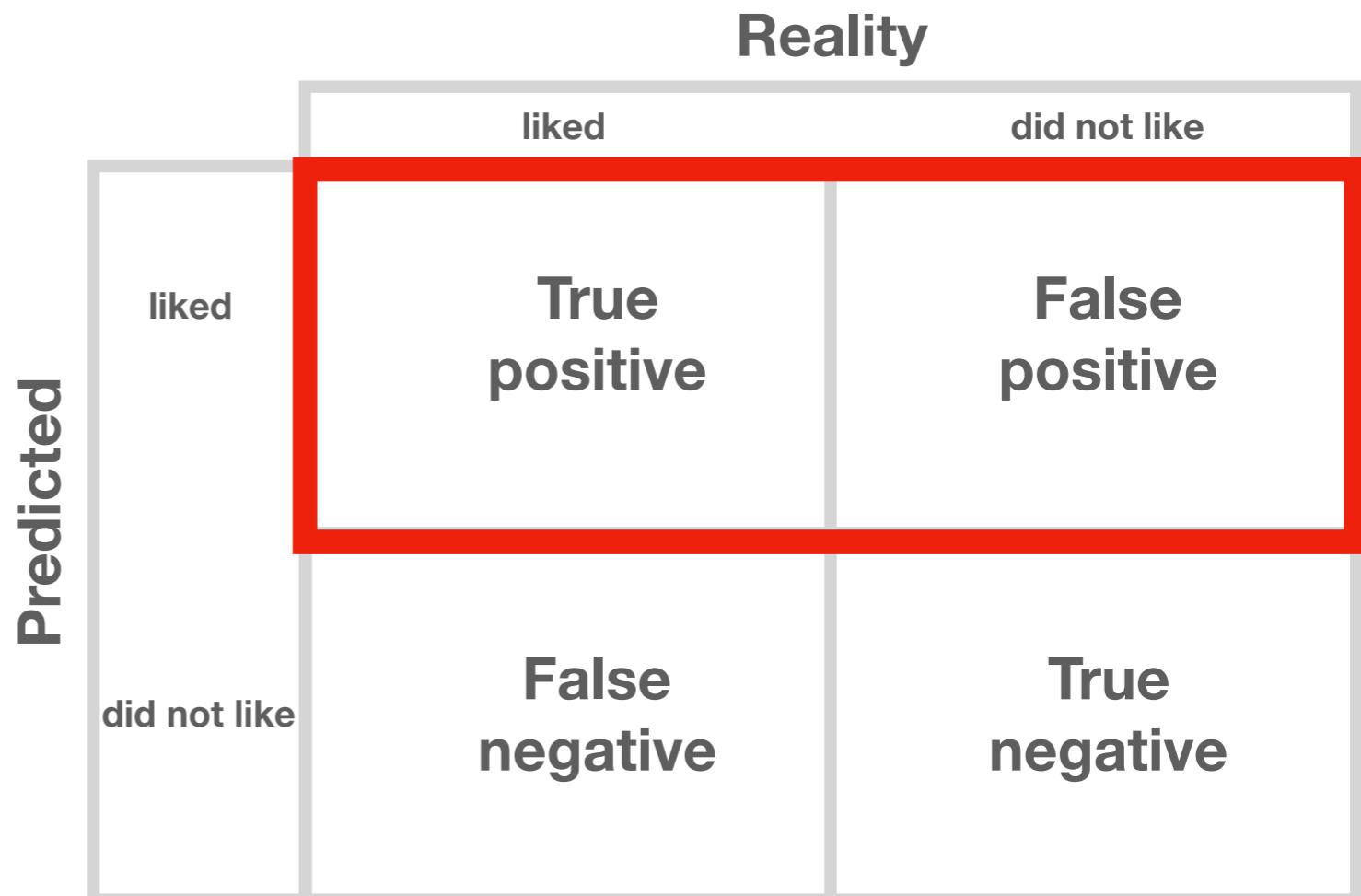
Step 5: Evaluation

Precision@K

Of the top k recommendations, what proportion are actually “relevant”?

Recall@K

Proportion of items that were found in the top k recommendations.



$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

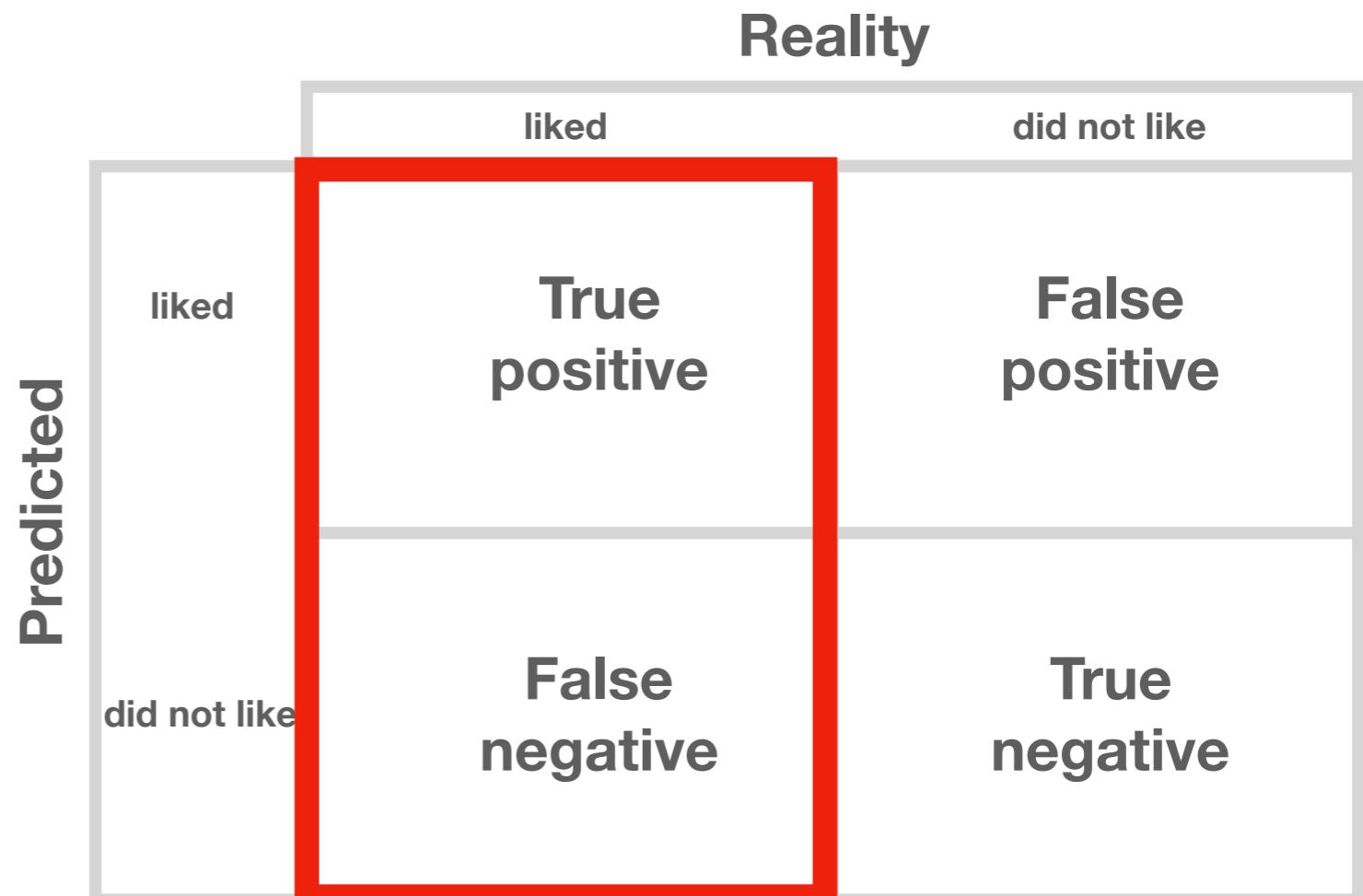
Step 5: Evaluation

Precision@K

Of the top k recommendations, what proportion are actually “relevant”?

Recall@K

Proportion of items that were found in the top k recommendations.



$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

~~$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$~~

Important Considerations

- Interpretability
- Efficiency and scalability
- Diversity
- Serendipity

Python Tools

- import surprise (@NicolasHug)
- import implicit (@benfred)
- import LightFM (@lyst)
- import pyspark.mllib.recommendation

Thank you!



Jill Cates
Data Scientist at BioSymetrics
github: [@topspinj](https://github.com/topspinj)
cates.jill@gmail.com

Cross-validation

- Prevent overfitting with k-fold cross-validation

