

Wi-Fi LIVE ONLINE TRAINING

Building Recommendation Engines in Python

Topic: Data



github.com/maxhumber/BRE

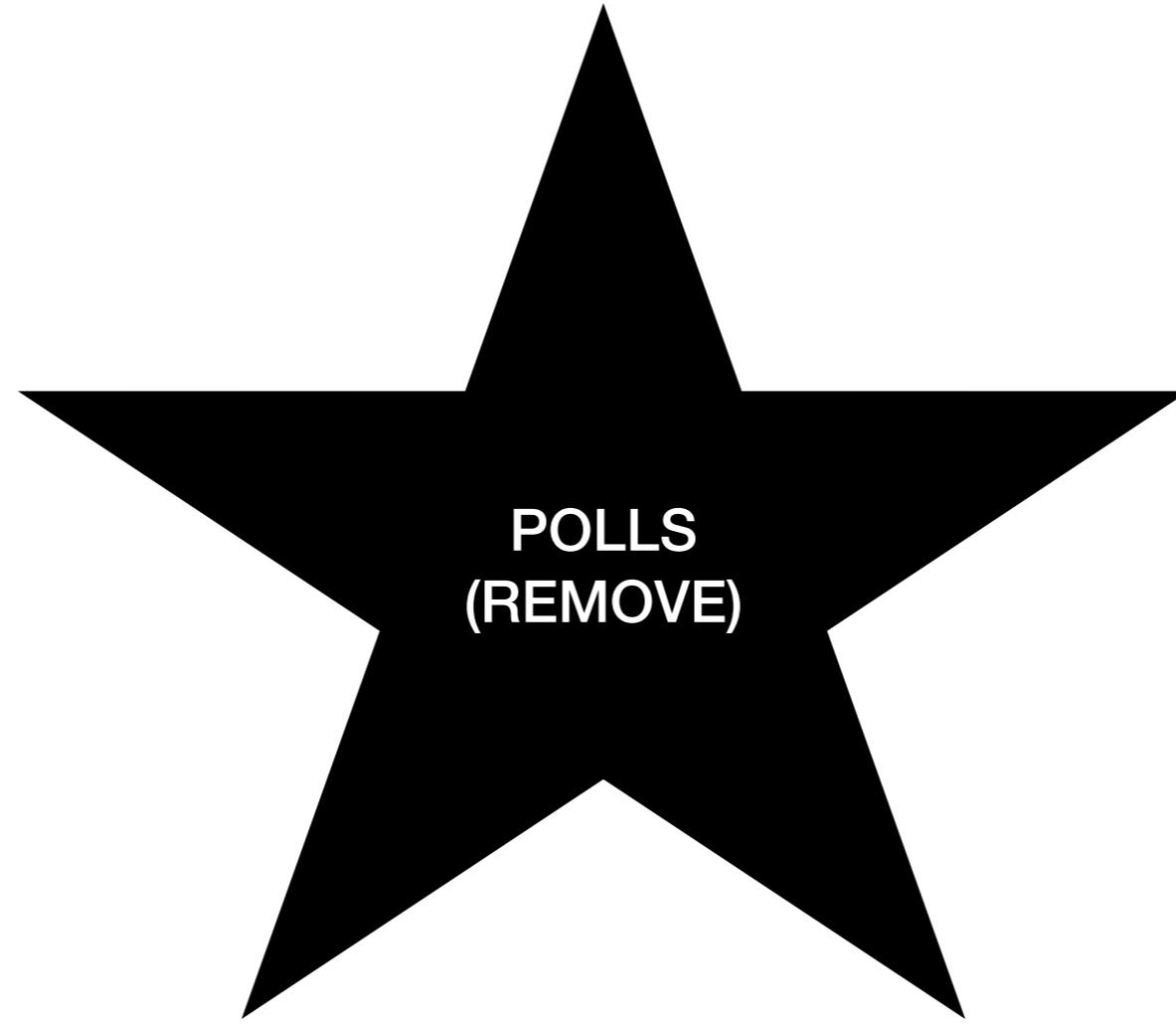
April 9, 2020

10:00am – 12:00pm EDT

1
Theory

2
Practice

3
Something
New



- Poll: Have you ever built a recommendation engine? {Y/N}
- Poll: Is your interest personal or professional? {personal/professional}

NETFLIX



amazon







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

Sheena S. Iyengar
Columbia University

Mark R. Lepper
Stanford University

6 samples



24 samples



VS.

Interest

6 samples



40% of shoppers

24 samples



60% of shoppers

Purchase

6 samples



30% conversion

24 samples



VS.

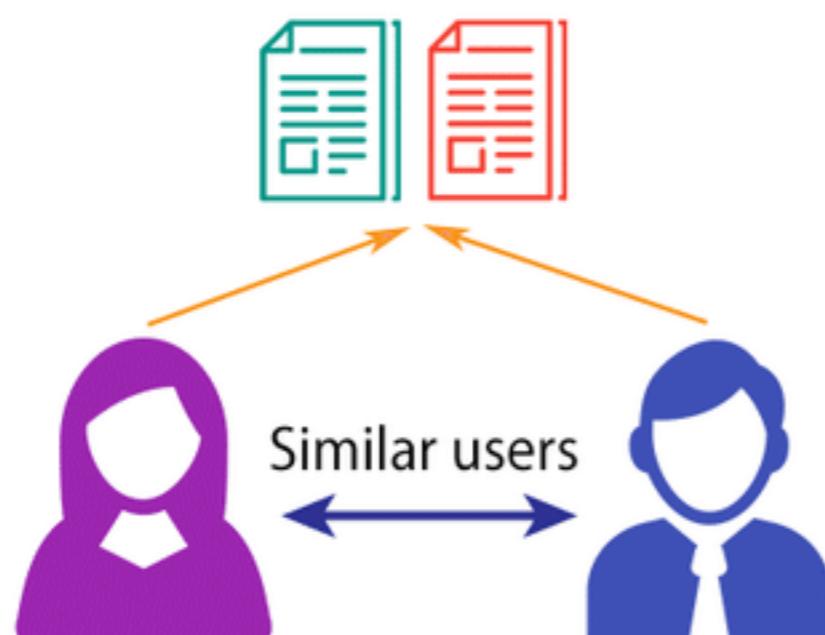
3% conversion



2 hours...

COLLABORATIVE FILTERING

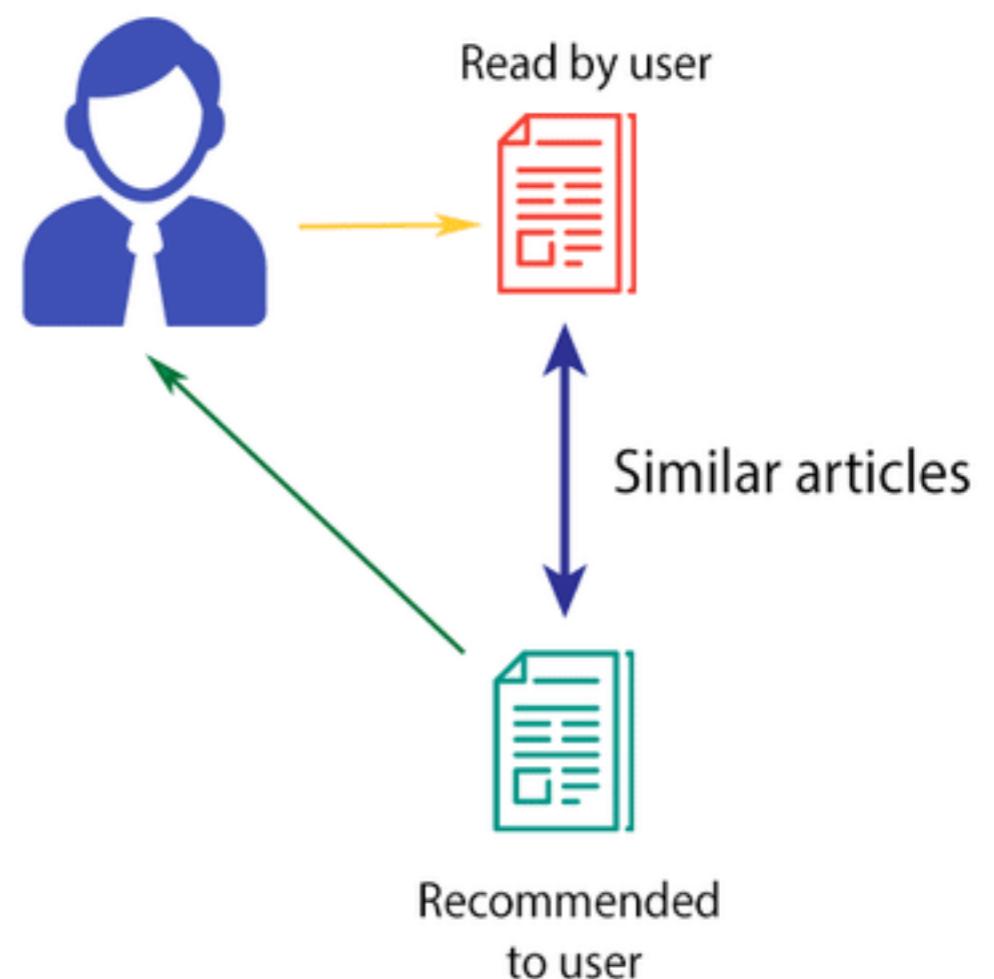
Read by both users



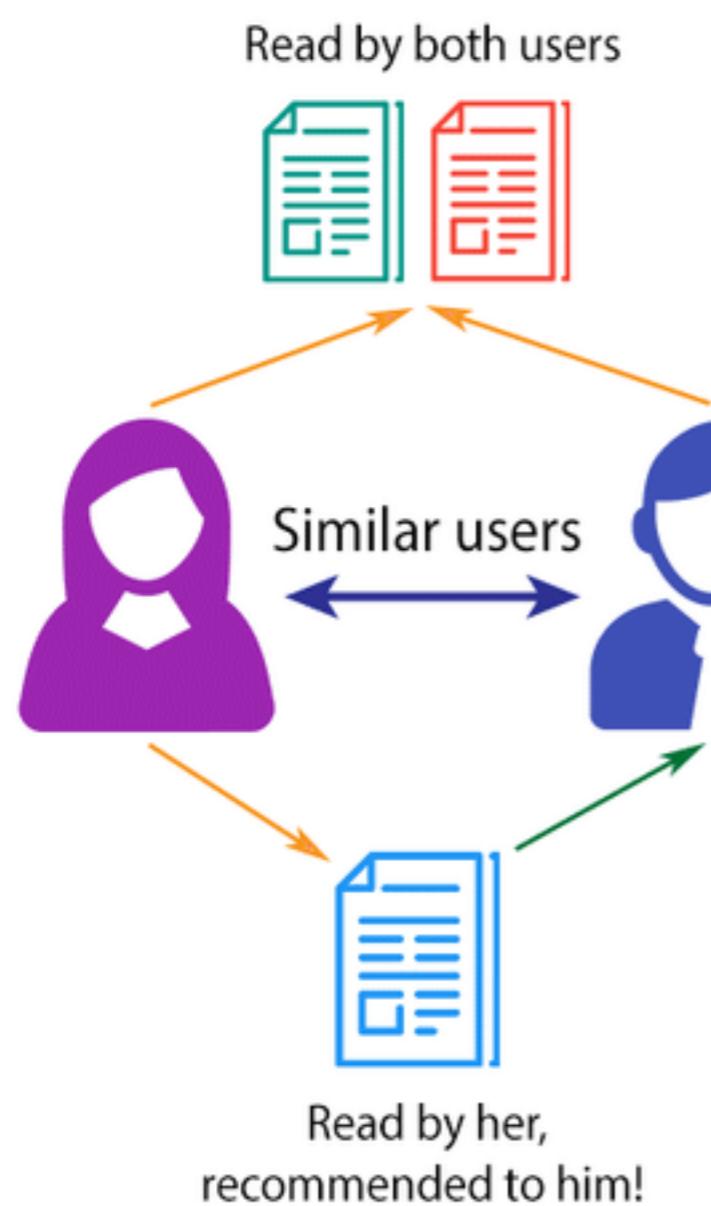
Read by her,
recommended to him!

CONTENT-BASED FILTERING

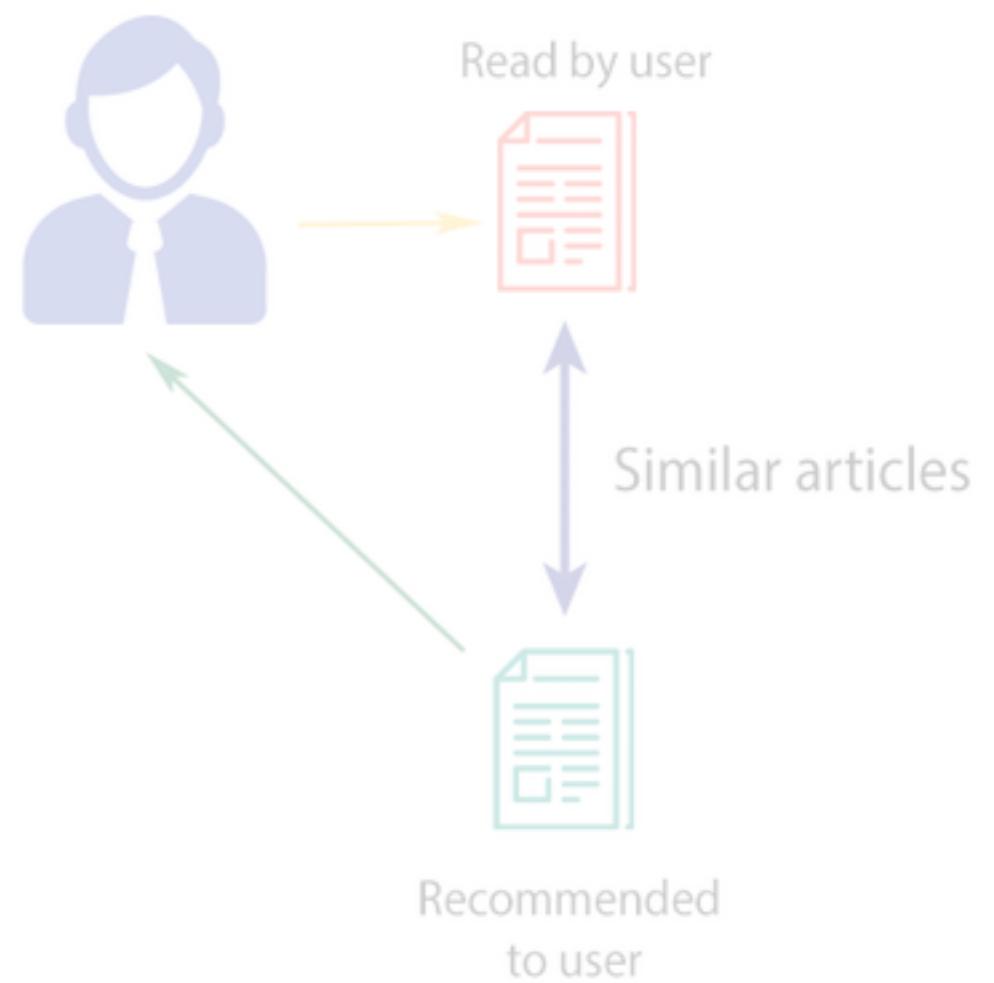
Read by user

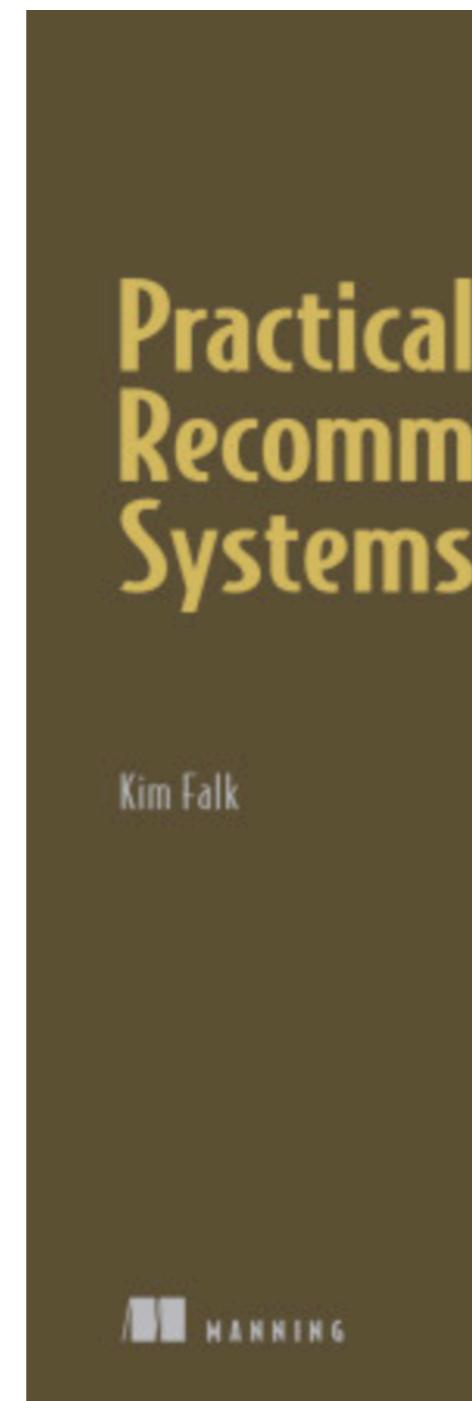


COLLABORATIVE FILTERING



CONTENT-BASED FILTERING





Practical Recommender Systems



Data

top picks

[see more](#)

based on your ratings, MovieLens recommends these movies

[Band of Brothers](#)

 2001 R 705 min 

[Casablanca](#)

 1942 PG 102 min 

[One Flew Over the Cuckoo's Nest](#)

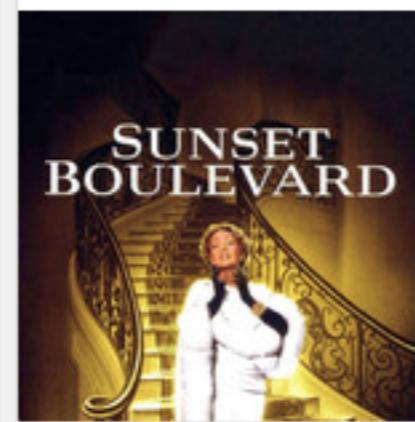
 1975 R 133 min 

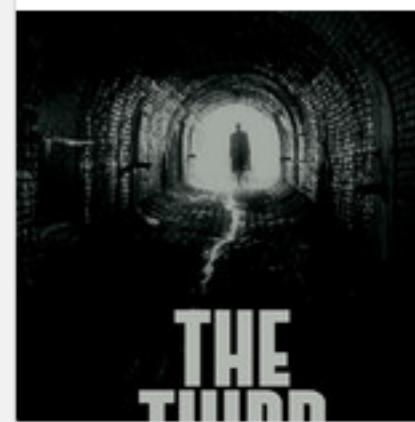
JACK NICHOLSON
ONE FLEW OVER THE CUCKOO'S NEST


[The Lives of Others](#)

 2006 R 137 min 

[Sunset Boulevard](#)

 1950 NR 110 min 

[The Third Man](#)

 1949 NR 104 min 

[Path](#)

 1957 


recent releases

[see more](#)

movies released in last 90 days that you haven't rated

[Cantinflas](#)

 2014 PG 106 min 

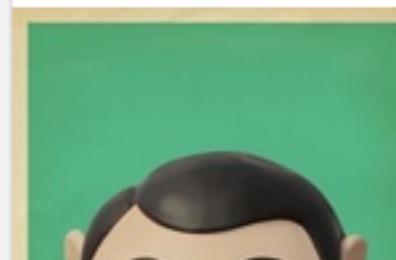
[Felony](#)

 2014 

[What If](#)

 2014 PG-13 102 min 

[Frank](#)

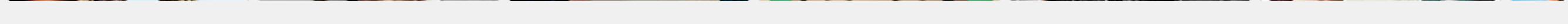
 2014 R 96 min 

[Sin City: A Dame to Kill For](#)

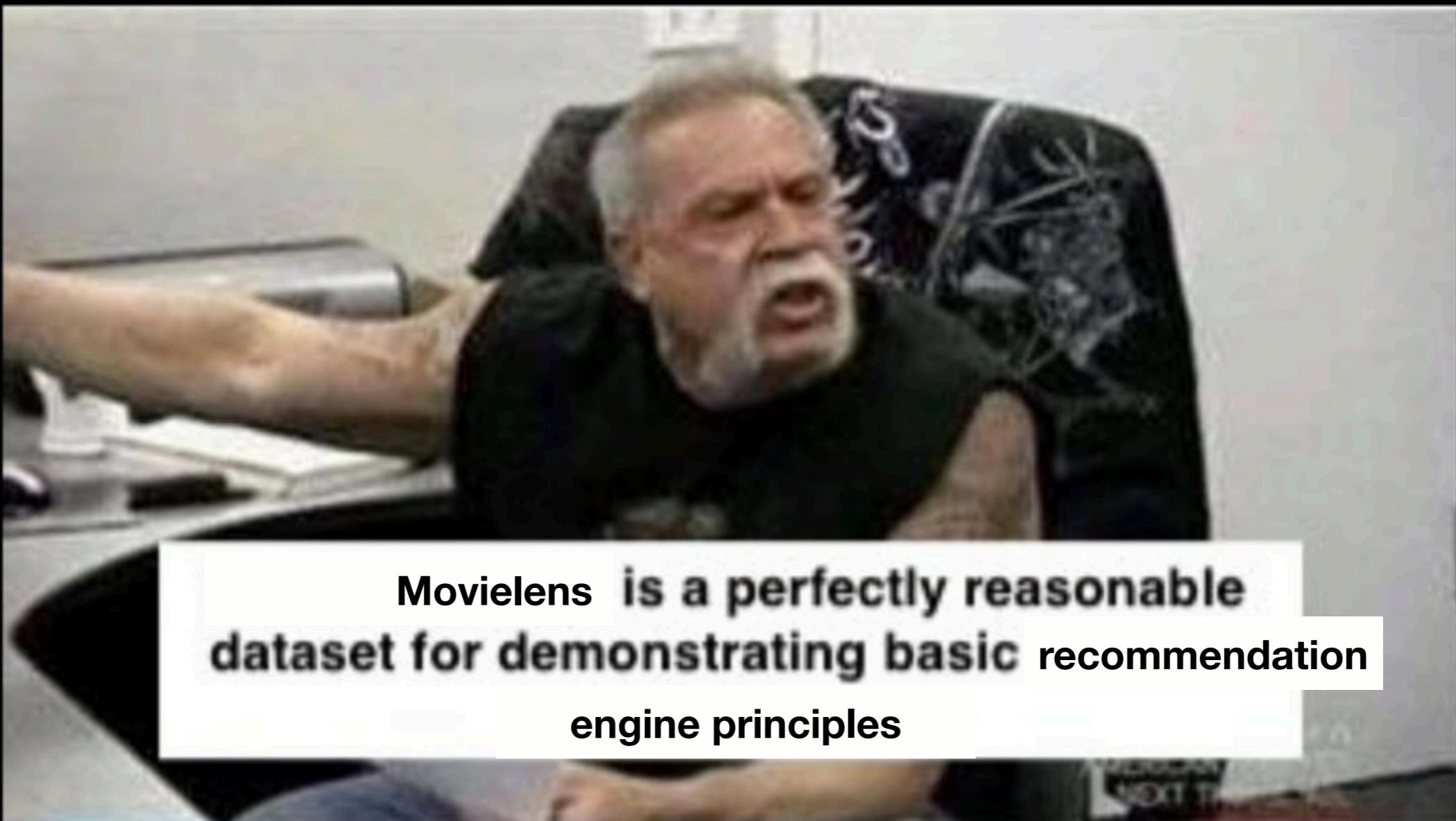
 2014 R 102 min 

[If I Stay](#)

 2014 PG-13 106 min 

[Are](#)

 2014 




**Movielens is a perfectly reasonable
dataset for demonstrating basic recommendation
engine principles**



**It lulls beginners into falsely
thinking working with data is
easy**



**EVERYONE NEEDS TO START
SOMEWHERE**

IT'S THE WRONG
PLACE FOR
BEGINNERS TO
START

AMERICAN DREAM
NEXT THURSDAY



**YOU DON'T
REMEMBER
WHAT IT'S LIKE
TO BE A
BEGINNER**

top picks

[see more](#)

based on your ratings, MovieLens recommends these movies

[Band of Brothers](#)

2001 R 705 min

[Casablanca](#)

1942 PG 102 min

[Catch-22](#)

1970 PG 140 min

[The Lives of Others](#)

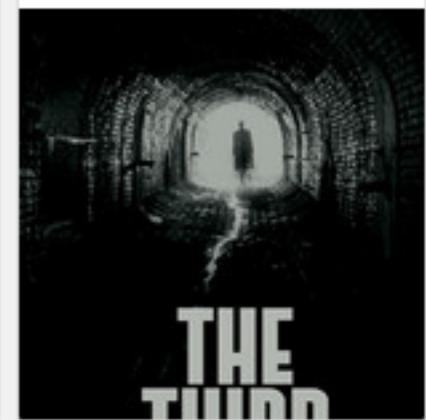
2006 R 137 min

[Sunset Boulevard](#)

1950 R 110 min

[The Third Man](#)

1949 NR 104 min

[Path](#)

1957



recent releases

movies released in last 90 days that you have

[Cantinflas](#)

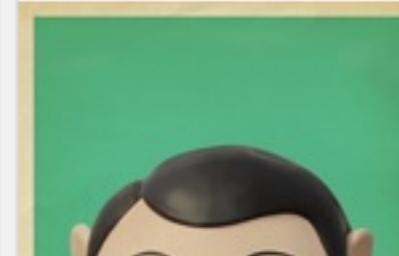
2014 PG 106 min

[Felony](#)

2014

[Frank](#)

2014 R 96 min

[Sin City: A Dame to Kill For](#)

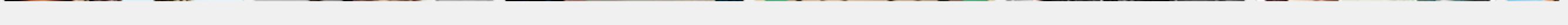
2014 R 102 min

[If I Stay](#)

2014 PG-13 106 min

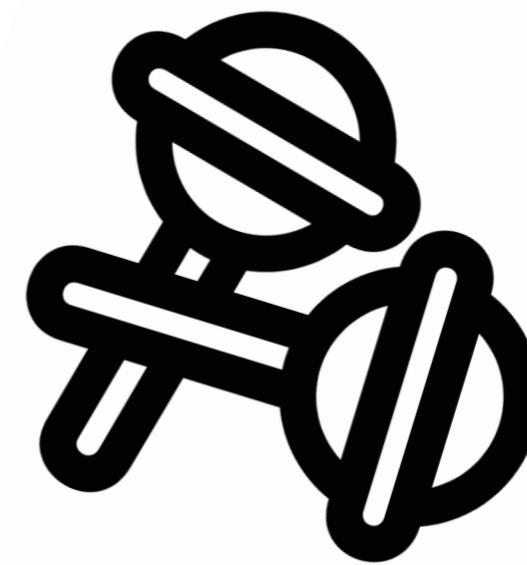
[Are](#)

2014





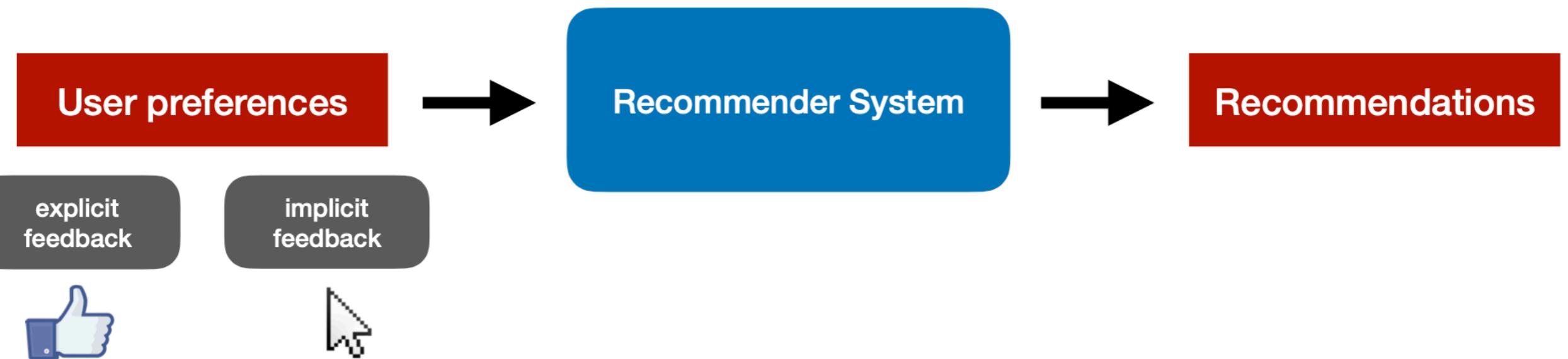
Stars!



Candy!!

1

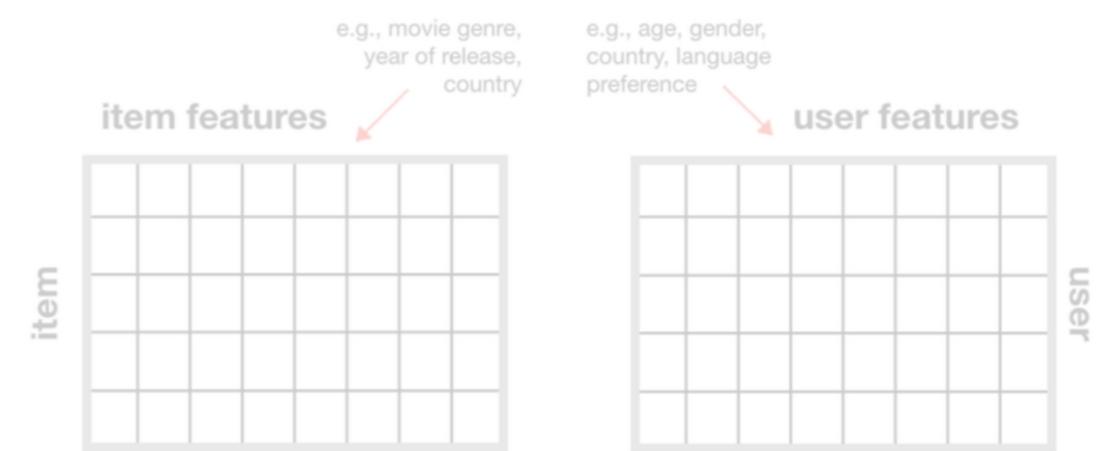
Theory



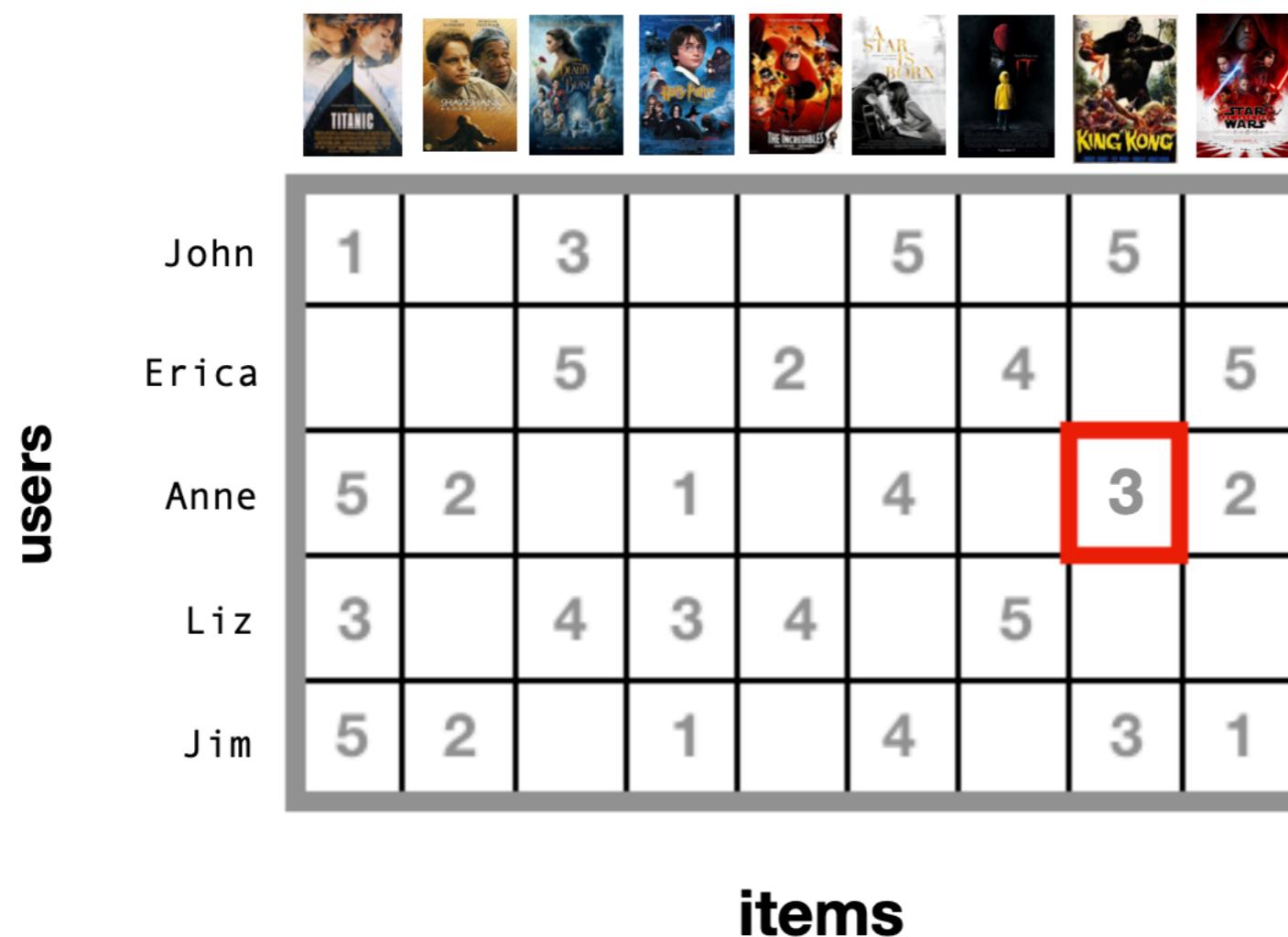
Collaborative filtering

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

Content-based filtering



Similar people like similar things



User-item (“utility”) matrix



**What are we populating
these cells with?**



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

Explicit feedback

Likert-scale rating (1-5)
Liked or not (boolean)

Implicit feedback

Browsing behaviour
Purchased? Read? Watched?

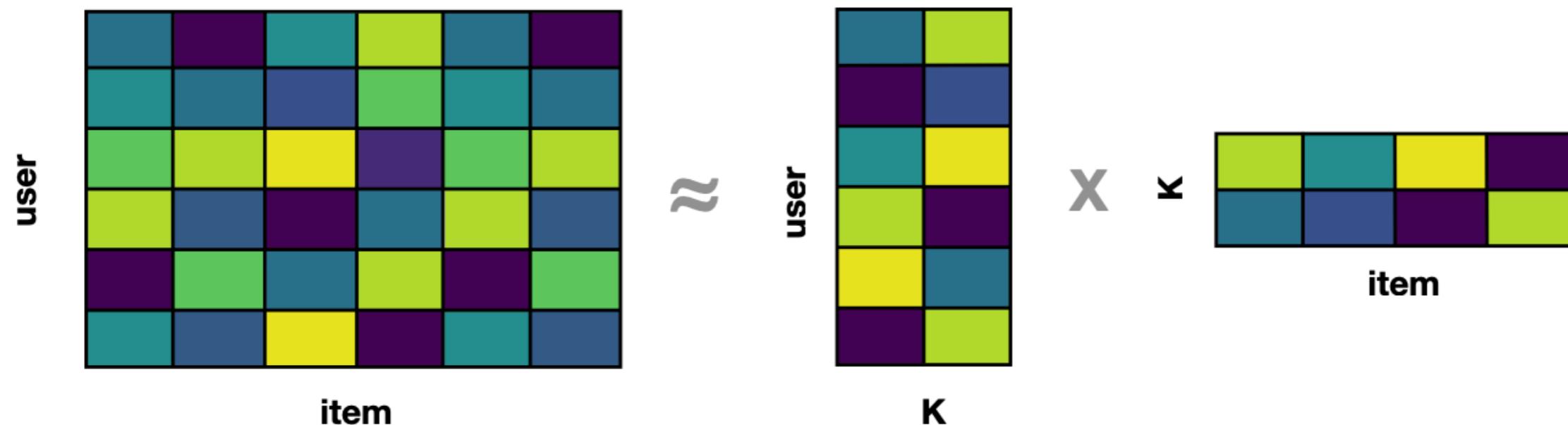
Developing a user feedback score

- Dwell time
- Recent vs. old interactions
- Negative implicit feedback
- What behaviour are you trying to drive?

Matrix Factorization

- Dimensionality reduction
- 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

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



Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches

Maurizio Ferrari Dacrema
Politecnico di Milano, Italy
maurizio.ferrari@polimi.it

Paolo Cremonesi
Politecnico di Milano, Italy
paolo.cremonesi@polimi.it

Dietmar Jannach
University of Klagenfurt, Austria
dietmar.jannach@aau.at

ABSTRACT

Deep learning techniques have become the method of choice for researchers working on algorithmic aspects of recommender systems. With the strongly increased interest in machine learning in general, it has, as a result, become difficult to keep track of what represents the state-of-the-art at the moment, e.g., for top-n recommendation tasks. At the same time, several recent publications point out problems in today's research practice in applied machine learning, e.g., in terms of the reproducibility of the results or the choice of the baselines when proposing new models.

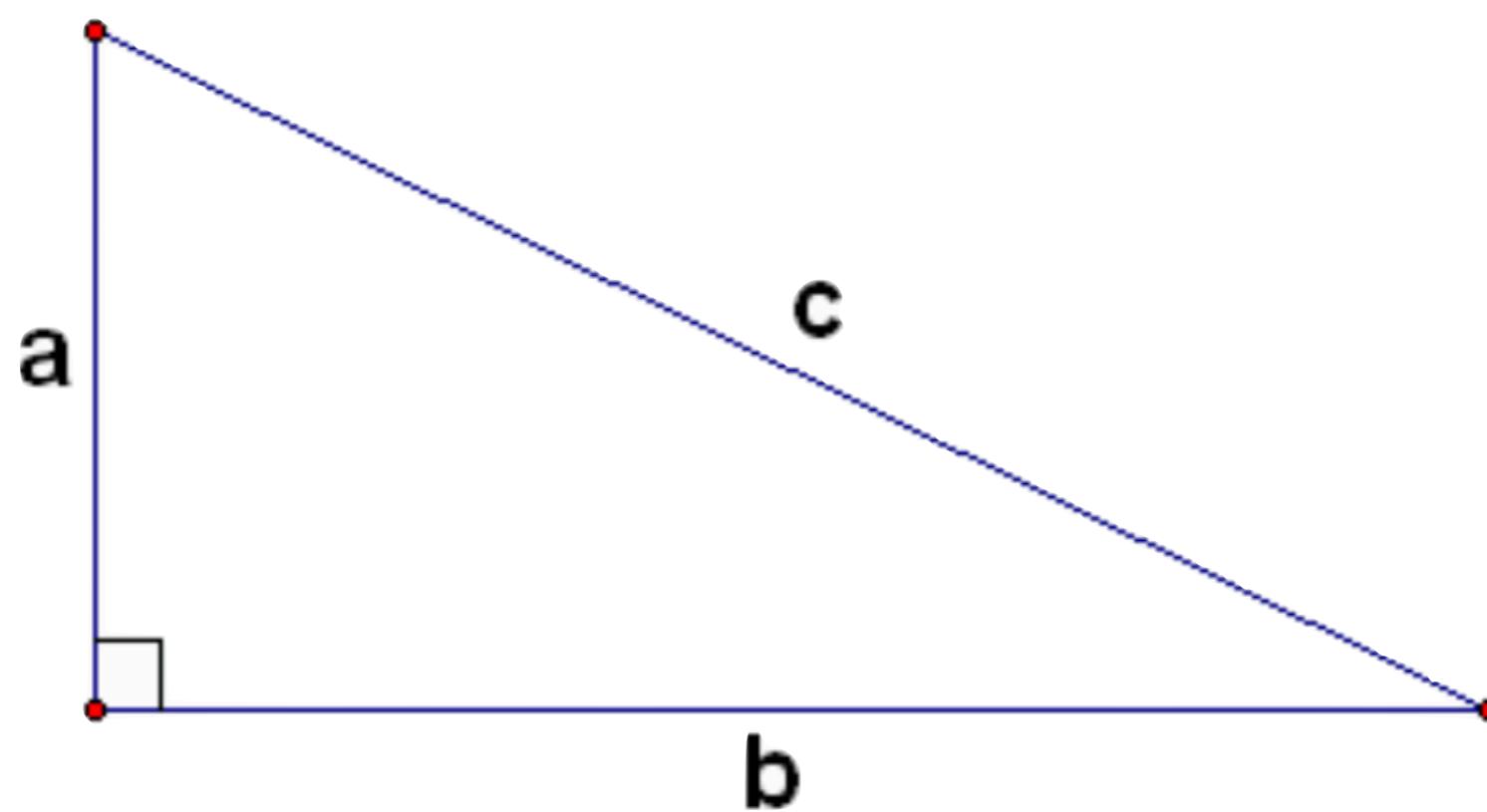
In this work, we report the results of a systematic analysis of algorithmic proposals for top-n recommendation tasks. Specifically, we considered 18 algorithms that were presented at top-level research conferences in the last years. Only 7 of them could be reproduced with reasonable effort. For these methods, it however turned out that 6 of them can often be outperformed with comparably simple heuristic methods, e.g., based on nearest-neighbor or graph-based techniques. The remaining one clearly outperformed the baselines but did not consistently outperform a well-tuned non-neural linear ranking method. Overall, our work sheds light on a number of potential problems in today's machine learning scholarship and calls for improved scientific practices in this area.

1 INTRODUCTION

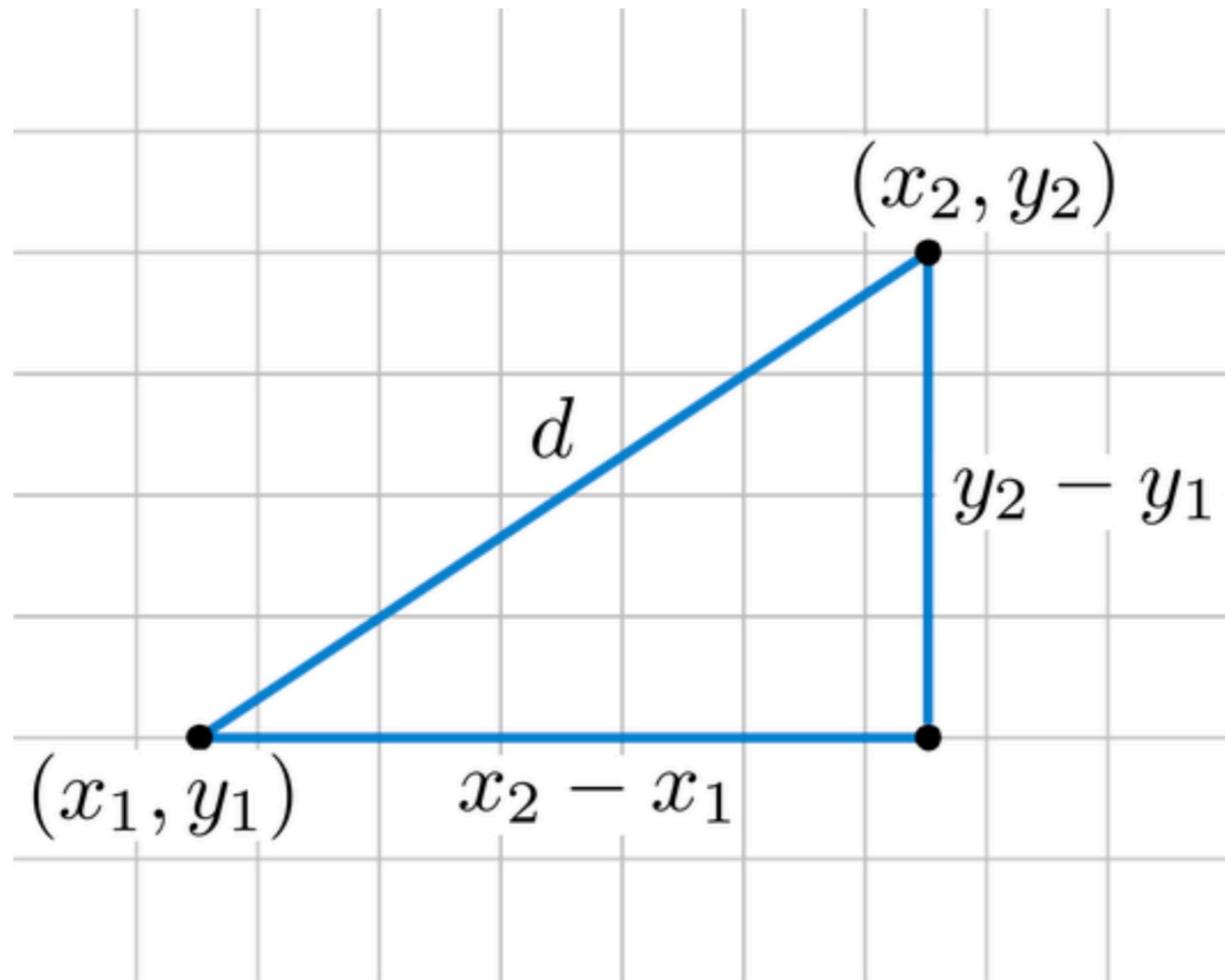
Within only a few years, deep learning techniques have started to dominate the landscape of algorithmic research in recommender systems. Novel methods were proposed for a variety of settings and algorithmic tasks, including top-n recommendation based on long-term preference profiles or for session-based recommendation scenarios [36]. Given the increased interest in machine learning in general, the corresponding number of recent research publications, and the success of deep learning techniques in other fields like vision or language processing, one could expect that substantial progress resulted from these works also in the field of recommender systems. However, indications exist in other application areas of machine learning that the achieved progress—measured in terms of accuracy improvements over existing models—is not always as strong as expected.

Lin [25], for example, discusses two recent neural approaches in the field of information retrieval that were published at top-level conferences. His analysis reveals that the new methods do *not* significantly outperform existing baseline methods when these are carefully tuned. In the context of recommender systems, an in-depth analysis presented in [29] shows that even a very recent neural method for session-based recommendation can, in most cases, be outperformed by very simple methods based, e.g., on nearest-





$$a^2 + b^2 = c^2$$



$$\text{Euclidean distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$



01_distance.ipynb

Q&A

2

Practice



 Message

More...

Maciej Kula · 3rd

London, United Kingdom · 396 connections · [Contact info](#)



Netflix



University of Oxford

Experience



Senior Research Engineer

Netflix

Aug 2016 – Present · 3 yrs 8 mos

Los Gatos, California

Personalization algorithms.



Data Scientist

Lyst

Nov 2012 – Aug 2016 · 3 yrs 10 mos

London, United Kingdom

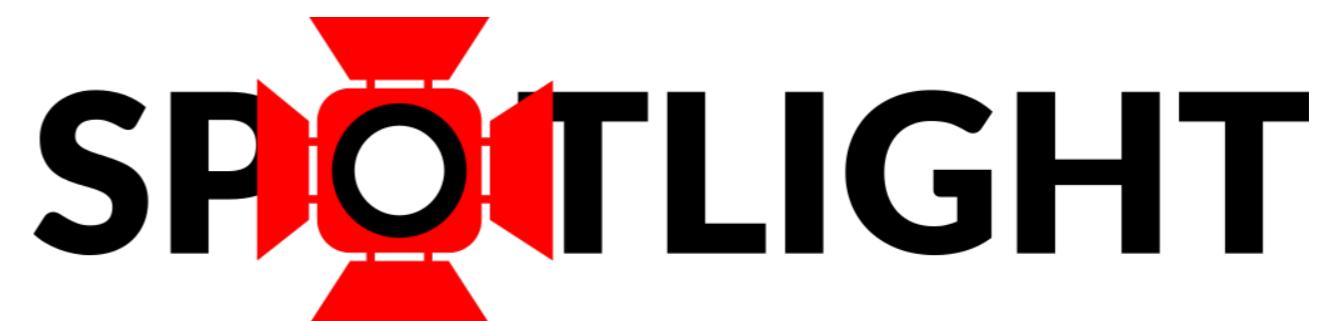
Recommendations and Personalization.



<https://github.com/lyst/lightfm>



<https://github.com/lyst/lightfm>

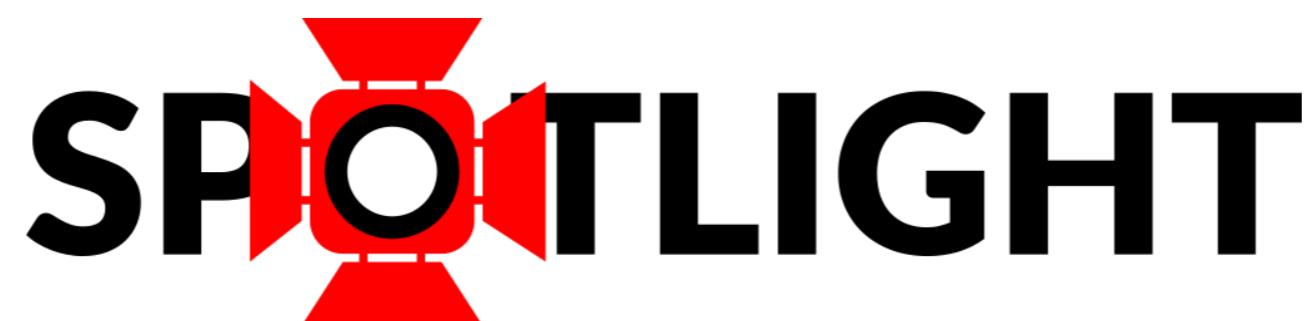


<https://github.com/maciejkula/spotlight>



02_lightfm.ipynb

Q&A



Spotlight uses [PyTorch](#) to build both deep and shallow recommender models. By providing both a slew of building blocks for loss functions (various pointwise and pairwise ranking losses), representations (shallow factorization representations, deep sequence models), and utilities for fetching (or generating) recommendation datasets, it aims to be a tool for rapid exploration and prototyping of new recommender models.

See the full [documentation](#) for details.

Installation

```
conda install -c maciekjula -c pytorch spotlight
```



03A_spotlight_setup.md



03B_spotlight.ipynb

Q&A

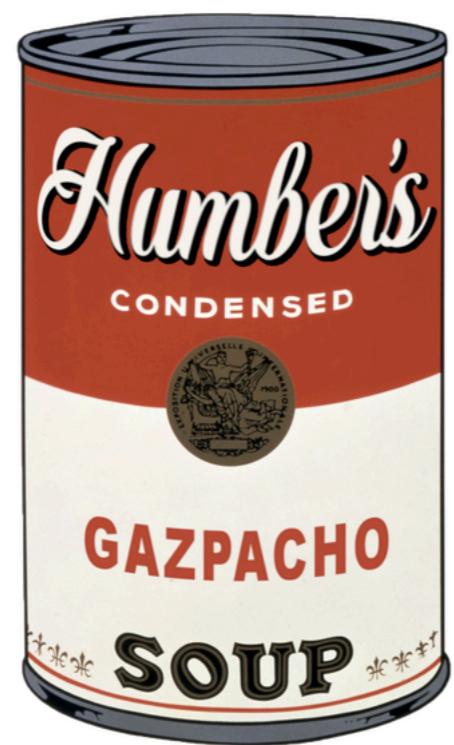


04_XYZ

Q&A

That's all Folks!

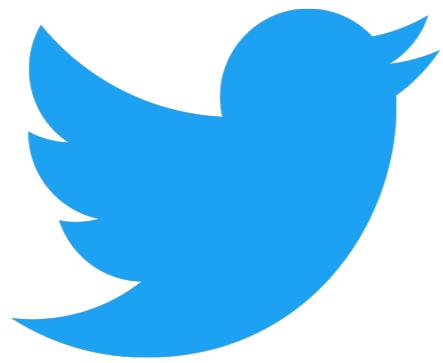




github.com/maxhumber/gazpacho



github.com/maxhumber/gif



twitter.com/maxhumber



www.linkedin.com/in/maxhumber