



Introduction to Intelligent User Interfaces

Recommender Systems

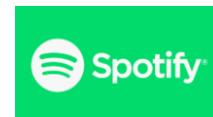
Basics, Algorithms (content-based, collaborative filtering), Applications

Recommender Systems

Definition

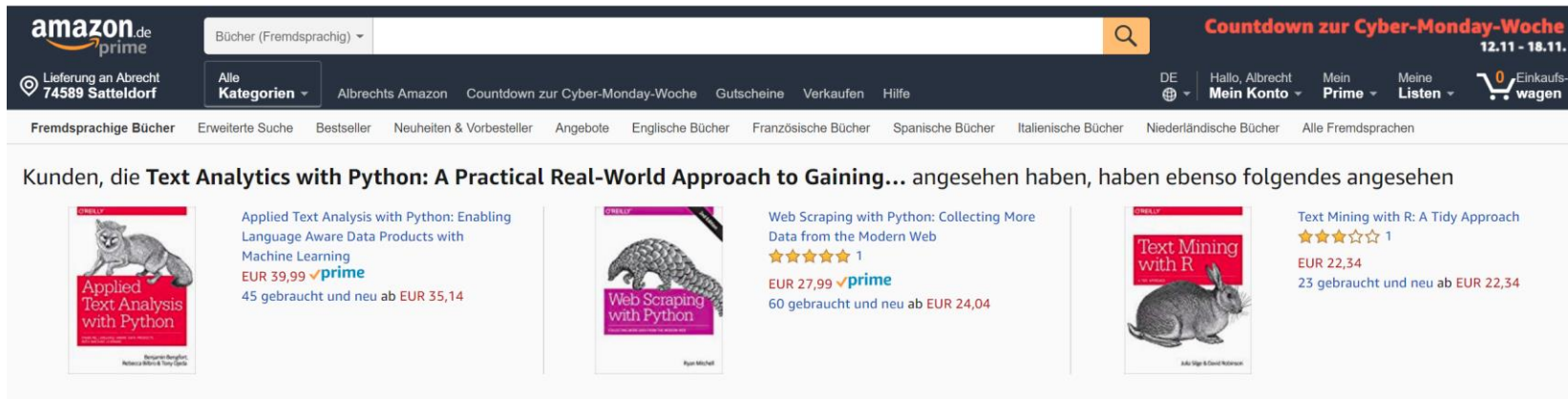
“Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user. [...] The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read.” [1, p1]

[1] Francesco Ricci, Lior Rokach and Bracha Shapira. Introduction to Recommender Systems Handbook. In F. Ricci et al. (eds.), Recommender Systems Handbook. Springer Science+Business Media 2011.

The Netflix logo, consisting of the word "NETFLIX" in a bold, red, sans-serif font.The Yelp logo, featuring the word "yelp" in a lowercase, black, sans-serif font, followed by a red starburst icon.The YouTube logo, featuring the word "You" in a black, sans-serif font, followed by the word "Tube" in a white, sans-serif font inside a red rounded rectangle.The Amazon.com logo, featuring the word "amazon.com" in a black, sans-serif font, with a curved orange arrow underneath the word "amazon".The Spotify logo, featuring a green square with a white circular icon containing three horizontal lines, followed by the word "Spotify" in a white, sans-serif font.The eBay logo, featuring the word "ebay" in a lowercase, sans-serif font, with each letter in a different color (e: red, b: blue, a: yellow, y: green).

Recommender Systems – Examples?

- How are recommender systems used in these services?
- How do recommender systems impact the user experience?
- What are user interface patterns used with recommender systems?



NETFLIX

yelp

You Tube

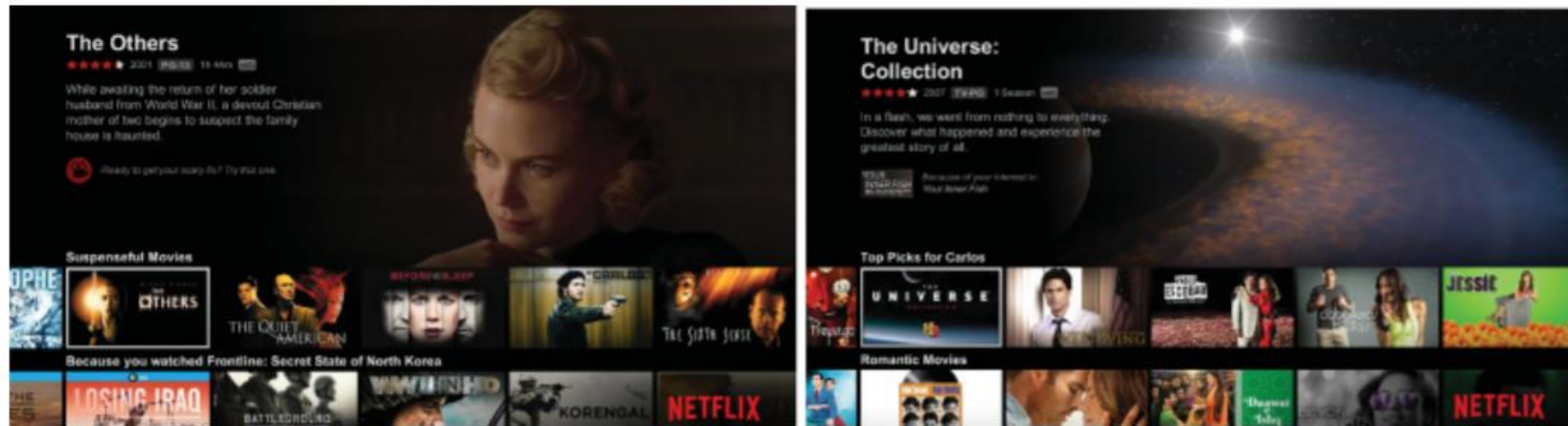
amazon.com

Spotify

ebay

Recommender Systems

- Why are recommender systems used?
- What is the main function?
- What data do recommender systems require?
- How do recommender systems make a user interface intelligent?



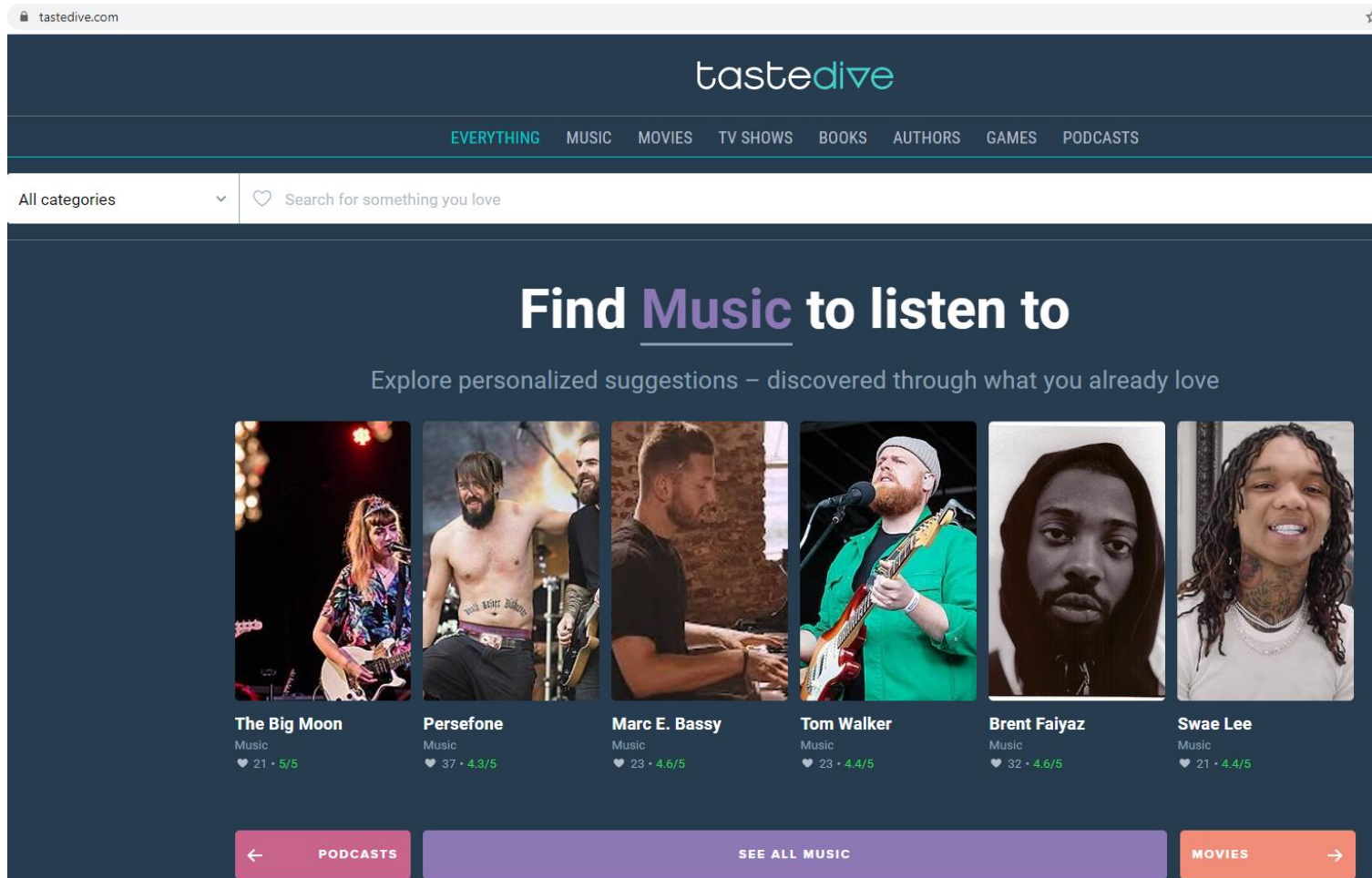
Carlos A. Gomez-Urbe and Neil Hunt. 2015. The Netflix Recommender System: Algorithms, Business Value, and Innovation. ACM Trans. Manage. Inf. Syst. 6, 4, Article 13 (December 2015). DOI: <https://doi.org/10.1145/2843948>



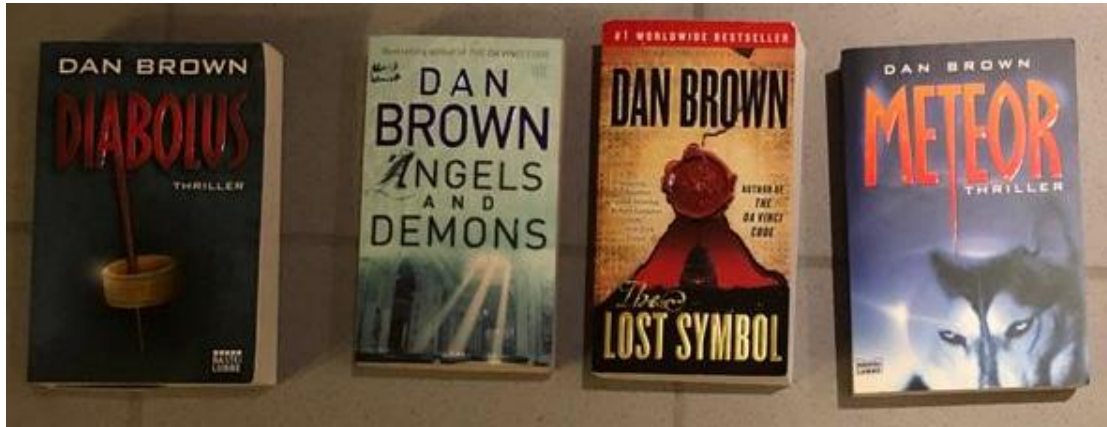
What Book do you Recommend to me?

Example: <https://tastedive.com/>

How does this work?

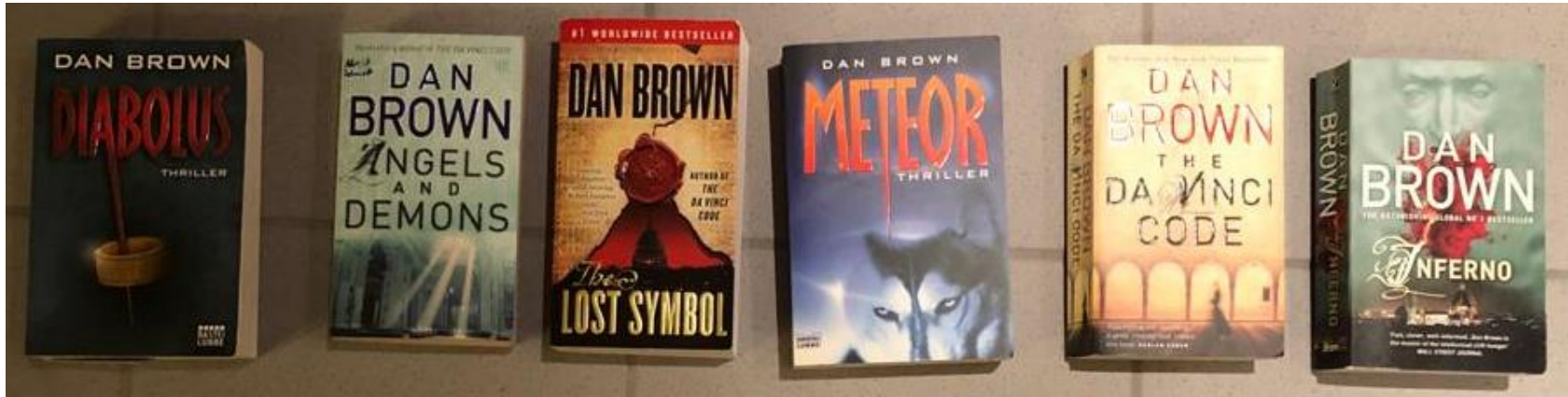


What Book do you Recommend to me?

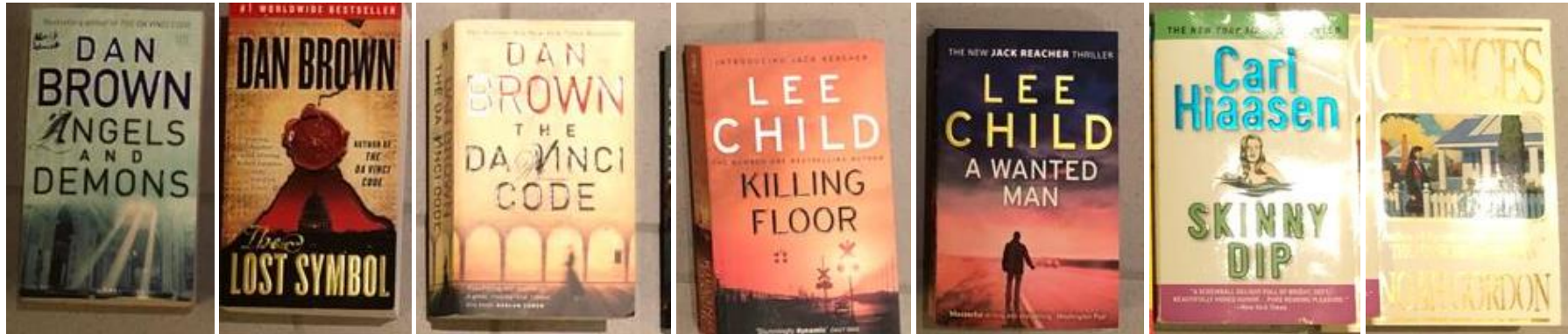


? ?

What Book do you Recommend to me?



This is what I read...



Which of these books do you suggest?



Approaches to Recommender Systems

- **Collaborative Filtering**

“people who liked what you liked also liked X, hence I suggest X to you”

- **Demographics**

“many people of your age, income, family status, and education like X, hence I suggest X to you”

- **Social Relationships**

“people you hang out with, your friend, or friend of your friends like X, hence I suggest X to you”

- **Content-based Filtering**

“X is similar (based on a similarity measure for a domain) to what you liked before, hence I suggest X to you”

- **Contextual**

“many people in your current situation/context/location like X, as you are in now this context, I suggest X to you”

Content Based Filtering

Basic Approach



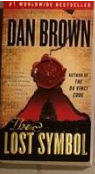
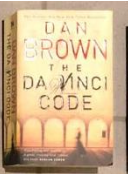


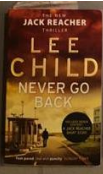

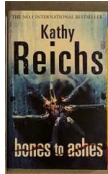

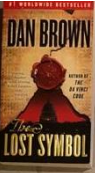



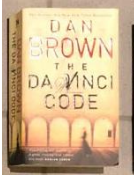


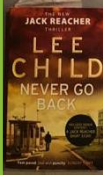
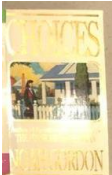




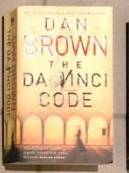

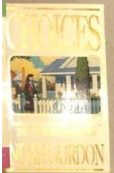
- What properties / factors / dimensions are describing and discriminating the products?
→ taxonomy, list of dimensions or set of criteria
- Each item can be categorized based on the criteria/dimensions/taxonomy
→ for each item represent how much it matches these criteria in a vector
- How much does the user like products based on the criteria/dimensions/taxonomy?
→ vector for the user that represents their relationship to these criteria
- Identify and recommend items that fit the user's criteria
→ calculate the similarity between item vectors and user vectors


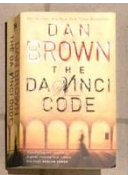








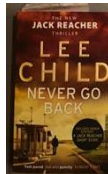
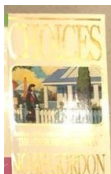

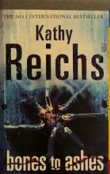


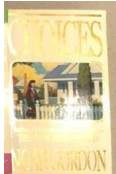




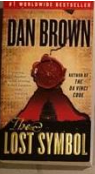
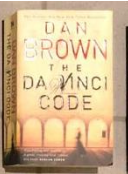


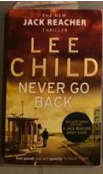

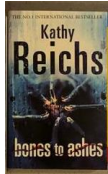

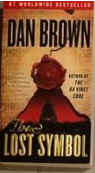



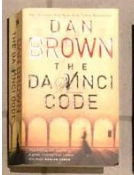

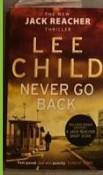
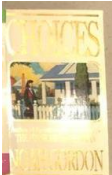




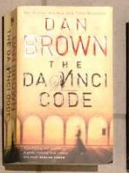

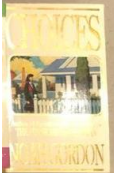
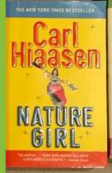
*Language, genre, time set,
main characters, location/setting,
theme, length, time written,
number of parallel plots,
language complexity,...*

Idea of User Based Collaborative Filtering



												
	8	7	9	4	9		10			4		7
		2				6	9					
	4		6	10	8		9		6	7	9	8
	4	?	6	?	?	?	?	9	5	?	10	?

	Item1	item2	item3	item4	Item5	item6	item7	item8	Item9	item10	item11	item12
u1	8	7	9	4	9		10			4		7
u2		2				6	9					
u3	4		6	10	8		9		6	7	9	8
u4	4	?	6	?	?	?	?	9	5	?	10	?

Breakout Sessions 1

How to get item ratings?

Explicitly and Implicitly

- Scenario 1: Web based reading platform/library
- Scenario 2: Public transport application
- Scenario 3: Online platform for travel recommendations

What do you get ratings for? What is the item?

What relevant questions can you ask to get item ratings?

How do you get implicit ratings from “automated observation”?

Discussion: Getting user input?

Where does Ethics come in?

Recommender Systems

Getting user data

What are ways to get data from the user for recommender systems?

- Explicit:
 - Questionnaires at setup time
 - Ratings of items
 - Feedback questions
- Implicit
 - Using items (e.g. watching a movie, listen to music, putting something on a watch list, buying a book)
 - Sharing items (e.g. recommending an article, retweets)
 - Removing items (e.g. deleting a playlist, skipping a suggested song)

Collaborative Filtering

Basic Approach

- Tables:
 - User, Item, Rating
 - Item, Item description
- Approach
 - Users rate items (implicitly, explicitly), e.g. like a movie, buy a book, recommend an hotel, ...
 - Table: user, item, rating
 - Compute similarity between users (or between items)
 - set of users (or set of items) that is similar to the user (the item) we create a recommendation for
 - Predict items (new to the target user) based on the information of similar users
 - weighted list of recommendations

Calculating the similarity

How similar are user to each other?

- Calculating the difference between rating vectors, e.g. k-nearest neighbor, Euclidian distance, Pearson correlation, Cosine distance

	Item A	Item B	Item C	Item D	Cosine Similarity(U_i, U_x)
User 1	7	10	3	4	0,938088
User 2	7	8	5	3	0,948847
User 3	8	2	5	3	0,708337
User 4	2	9	5	2	0,986921
User 5	1	2	5	2	0,828970
User X	3	8	5	3	1,000000

Heuristic required
for tables with
missing values!

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

https://en.wikipedia.org/wiki/Cosine_similarity

Example based on:

<https://buddingdatascientist.wordpress.com/2017/03/22/how-to-calculate-cosine-similarity-in-excel/>

Assessing Similarity?

How could Machine Learning play a role here?

												
	8	7	9	4	9		10			4		7
		2				6	9					
	4		6	10	8		9		6	7	9	8
	4	?	6	?	?	?	?	9	5	?	10	?

	Item A	Item B	Item C	Item D	Cosine Similarity(U_i, U_x)
User 1	7	10	3	4	0,938088
User 2	7	8	5	3	0,948847
User 3	8	2	5	3	0,708337
User 4	2	9	5	2	0,986921
User 5	1	2	5	2	0,828970
User X	3	8	5	3	1,000000





Predicting the right item?

Example Similarity Matrix

Similarities are NOT calculated – for illustration only

												
	8	7	9	4	9		10			4		7
		2				6	9					
	4		6	10	8		9		6	7	9	8
	4	?	6	?	?	?	?	9	5	?	10	?

Similarity Matrix

	0.4
	0.7
	0.9
	1

Based on <https://medium.com/swlh/how-to-build-simple-recommender-systems-in-python-647e5bcd78bd>

Weighted Rating Matrix

Rating Matrix
(part)





		
	7	4
	2	
		10
	?	?

*

*

*

Similarity Matrix

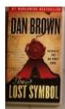





	0.4
	0.7
	0.9
	1

=

=

=

Weighted Rating Matrix
(part)

		
	2.8	1.6
	1.4	
		9.0
	?	?

Based on <https://medium.com/swlh/how-to-build-simple-recommender-systems-in-python-647e5bcd78bd>

Recommendation

Similarity Matrix

	0.4
	0.7
	0.9

Weighted Rating Matrix
(part)

		
	2.8	1.6
	1.4	
		9.0

Summing up weights

		4.2	$0.4 + 0.7$
		10.6	$0.4 + 0.9$

Summing up weighted ratings

		
	3.8	8.2



Based on <https://medium.com/swlh/how-to-build-simple-recommender-systems-in-python-647e5bcd78bd>

Collaborative Filtering

User Based

- Calculate similarity between users
- Suggest items that similar users liked

Item Based

- Calculate similarity between items based on user ratings – no semantics are required!
- Suggest item that is similar to item the user already likes

Cold-Start Problem

What is n

- **Basic problem:** as a system is created data is missing. The algorithms need initial information (e.g. to calculate similarities).
- **What could be new? What is the problem?**
 - **User:** no information yet, no recorded interaction nor know preferences
 - **Item:** the item has not been “like” or viewed by anyone yet, no knowledge who may like it or how its rating are similar
 - **Community:** new system is created, lack of information about users as well as item
- **Solutions?**
 - Hybrid approaches (e.g. using content based filtering to get started and then move more towards collaborative filtering)
 - Use social, demographics, context, content to get started
 - Require initial interaction (e.g. questionnaire, ask for examples, ...)

[1] Bobadilla, Jesús; Ortega, Fernando; Hernando, Antonio; Bernal, Jesús (February 2012). "A collaborative filtering approach to mitigate the new user cold start problem". Knowledge-Based Systems. doi:10.1016/j.knosys.2011.07.021.

[2] [https://en.wikipedia.org/wiki/Cold_start_\(recommender_systems\)](https://en.wikipedia.org/wiki/Cold_start_(recommender_systems))

Pinterest

Registration Process and start-up

Demographics

App Store 23:17

Was ist dein Geschlecht?

Damit können wir relevantere Inhalte für dich finden.

Weiblich

Männlich

Andere Angabe

Contextual

App Store 23:17

Wähle dein Land/deine Region aus

Wir möchten die besten Ideen für dich finden.

Deutschland (Deutschland)

Weiter

Explicit Interest

App Store 23:17

Willkommen bei Pinterest
Wähle 5 oder mehr Themen

Grillen

Diy Möbel

Technologie

Fingerfood

Handwerk Und Basteln

Herrenstile

Fleisch

Reise

Naturfotografie

Badezimmer

Cocktail Rezepte

Fotografie

Weiter

App Store 23:18

Für dich Folge ich

Parmesan Knoblauch Kartoffelecken

Startseite Suchen Benachrichtigungen Gemerkt

How to include new item?

Scenario: News Feed

- You create a recommender system for a news feed in social media (e.g. twitter style).
- How do you add new articles?
- Assuming you get a lot of articles....

Sparseness of Ratings

- E.g. Amazon (low estimates for illustration only)
 - Over 10 Million products
 - Over 100 Million customer
- What is the problem?

What Book do you Recommend to me?

Why does the UI matter?



Breakout Sessions 2

Recommendation and User Interface

- How can you measure the quality of the user experience of a recommender system?
- How does the screen size impact the performance of a recommender system?

Recommender Systems

Summary

Content-based

- Based on the similarity of items
- How similar is a new item to an item already liked/watched/bought by the user
- Advantage
 - If similarity is known or can be calculated, no ratings/actions from the user are required
- Difficulty:
 - Information/meta-data/algorithms for calculating similarity are required

Collaborative Filtering

- suggestions are made based on users that had similar interests/actions
- How has a similar user liked this item?
- Advantage:
 - Know knowledge about the item is required, no meta data or similarity calculation of items required
- Difficulty:
 - Data about other users is required
 - “cold-start” problem

Example data for experiments

<https://grouplens.org/datasets/movielens/>

grouplens about datasets publications blog

MovieLens

GroupLens Research has collected and made available rating data sets from the MovieLens web site (<http://movielens.org>). The data sets were collected over various periods of time, depending on the size of the set. Before using these data sets, please review their README files for the usage licenses and other details.

Help our research lab: Please [take a short survey](#) about the MovieLens datasets

recommended for new research

MovieLens 20M Dataset

Stable benchmark dataset. 20 million ratings and 465,000 tag applications applied to 27,000 movies by 138,000 users. Includes tag genome data with 12 million relevance scores across 1,100 tags. Released 4/2015; updated 10/2016 to update links.csv and add tag genome data.

- [README.html](#)
- [ml-20m.zip](#) (size: 190 MB, [checksum](#))

Also see the [MovieLens 20M YouTube Trailers Dataset](#) for links between MovieLens movies and movie trailers hosted on YouTube.

Permalink: <http://grouplens.org/datasets/movielens/20m/>

Datasets

- MovieLens
- Wikidata
- Book-Crossing
- Jester
- EachMovie
- HetRec 2011
- Serendipity 2018

<https://grouplens.org/datasets/movielens/>

	A	B	C	D	E
1	userId	movieId	rating	timestamp	
2	1	1	4	964982703	
3	1	3	4	964981247	
4	1	6	4	964982224	
5	1	47	5	964983815	
6	1	50	5	964982931	
7	1	70	3	964982400	
8	1	101	5	964980868	
9	1	110	4	964982176	
10	1	151	5	964984041	
11	1	157	5	964984100	
12	1	163	5	964983650	
13	1	216	5	964981208	
14	1	223	3	964980985	
15	1	231	5	964981179	
16	1	235	4	964980908	
17	1	260	5	964981680	
18	1	296	3	964982967	
19	1	316	3	964982310	
20	1	333	5	964981179	
21	1	349	4	964982563	
22	1	356	4	964980962	
23	1	362	5	964982588	
24	1	367	4	964981710	
25	1	423	3	964982363	

Additional reading material: <https://hub.packtpub.com/recommending-movies-scale-python/>

Live coding example

IUI - Recommendation Systems - Collaborative Filtering

by Sven Mayer

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Download and extract dataset

Source <https://grouplens.org/datasets/movielens/>

```
In [ ]: # Download
import urllib.request
url = 'http://files.grouplens.org/datasets/movielens/ml-latest-small.zip'
urllib.request.urlretrieve(url, './archive.zip')

# Extract
import zipfile
with zipfile.ZipFile("./archive.zip", "r") as zip_ref:
    zip_ref.extractall("")

import os
os.rename("ml-latest-small", "archive")
```

```
In [ ]: ! ls -la -h archive
```

```
In [ ]: dfRatings = pd.read_csv("./archive/ratings.csv")
dfRatings.head()
```

```
In [ ]: print("%i Ratings" % len(dfRatings))
```

Breakout Sessions 3

Recommender System

- Design a systems that suggest educational videos on YouTube that fit your lectures in this term

Algorithms for Recommender System

- Not core to Intelligent User Interfaces
- Efficient implementations in libraries
- Important to understand the algorithms to get the parameters right
- Many online resources, .e.g.
 - <https://www.youtube.com/watch?v=Eeg1DEeWUjA>
(introductory video)
 - <https://www.youtube.com/watch?v=Gf4HZpZAIDA>
<https://www.youtube.com/watch?v=b8YyIVulszQ>
(details on Collaborative Filtering Algorithms)

Examples for recommender systems in Python

An introduction to Collaborative filtering in Python and an overview of Surprise.



<https://www.youtube.com/watch?v=z0dx-YckFko>

- <https://kerpanic.wordpress.com/2018/03/26/a-gentle-guide-to-recommender-systems-with-surprise/>
- <https://www.analyticsvidhya.com/blog/2016/06/quick-guide-build-recommendation-engine-python/>
- <https://medium.com/@connectwithghosh/recommender-system-on-the-movielens-using-an-autoencoder-using-tensorflow-in-python-f13d3e8d600d>