Computational Communication Science 2
Week 6 - Lecture
»Recommender systems«

Marthe Möller
Anne Kroon

A.M.Moller@uva.nl, @MartheMoller
a.c.kroon@uva.nl, @annekroon

April, 2022

Digital Society Minor, University of Amsterdam

Today

Recommender Systems

Knowledge-based RecSys

Content-based RecSys

Collaborative RecSys

Group assignment

Recommender Systems


You
Tube
in

Recommender Systems in Communication Science

New research questions

1. Political communication and journalism. E.g., to craft personalized news diets. This may have, however, consequences for the diversity of news diets and for democracy (Locherbach & Trilling, 2018; Möller et al., 2018)
2. Organizational and corporate communication. E.g., applications in the field of hiring and recruitment.
3. Persuasive communication. E.g., in the health domain–recommendation algorithms for tailored health interventions (Kim et al., 2019)
4. Entertainment communication. E.g., movie recommenders

)

Recommender Systems in Communication Science

New research questions

1. Political communication and journalism. E.g., to craft personalized news diets. This may have, however, consequences for the diversity of news diets and for democracy (Locherbach & Trilling, 2018; Möller et al., 2018)
2. Organizational and corporate communication. E.g., applications in the field of hiring and recruitment.
3. Persuasive communication. E.g., in the health domain–recommendation algorithms for tailored health interventions (Kim et al., 2019)
4. Entertainment communication. E.g., movie recommenders

)

Recommender Systems in Communication Science

New research questions

1. Political communication and journalism. E.g., to craft personalized news diets. This may have, however, consequences for the diversity of news diets and for democracy (Locherbach & Trilling, 2018; Möller et al., 2018)
2. Organizational and corporate communication. E.g., applications in the field of hiring and recruitment.
3. Persuasive communication. E.g., in the health domain–recommendation algorithms for tailored health interventions (Kim et al., 2019)
4. Entertainment communication. E.g., movie recommenders

)

Recommender Systems in Communication Science

New research questions

1. Political communication and journalism. E.g., to craft personalized news diets. This may have, however, consequences for the diversity of news diets and for democracy (Locherbach & Trilling, 2018; Möller et al., 2018)
2. Organizational and corporate communication. E.g., applications in the field of hiring and recruitment.
3. Persuasive communication. E.g., in the health domain–recommendation algorithms for tailored health interventions (Kim et al., 2019)
4. Entertainment communication. E.g., movie recommenders

)

Recommender Systems in Communication Science

New research questions

1. Political communication and journalism. E.g., to craft personalized news diets. This may have, however, consequences for the diversity of news diets and for democracy (Locherbach & Trilling, 2018; Möller et al., 2018)
2. Organizational and corporate communication. E.g., applications in the field of hiring and recruitment.
3. Persuasive communication. E.g., in the health domain–recommendation algorithms for tailored health interventions (Kim et al., 2019)

)

Recommender Systems

Two central problems within Recommender Systems

1. Predicting problem. Typical problem involves a lot of missing data (e.g., user only rated a small subset of movies/news articles/ etc.) How can we deal with missing data?
2. Ranking problem. Given n items, can we identify the top k items to recommend?

Recommender Systems

Two central problems within Recommender Systems

1. Predicting problem. Typical problem involves a lot of missing data (e.g., user only rated a small subset of movies/news articles/ etc.) How can we deal with missing data?
2. Ranking problem. Given n items, can we identify the top k items to recommend?

Recommender Systems

Two central problems within Recommender Systems

1. Predicting problem. Typical problem involves a lot of missing data (e.g., user only rated a small subset of movies/news articles/ etc.) How can we deal with missing data?
2. Ranking problem. Given n items, can we identify the top k items to recommend?

Recommender Systems

Two central problems within Recommender Systems

1. Predicting problem. Typical problem involves a lot of missing data (e.g., user only rated a small subset of movies/news articles/ etc.) How can we deal with missing data?
2. Ranking problem. Given n items, can we identify the top k items to recommend?

These problems are not isolated; rather, they are connected.

Recommender Systems

How do recommender systems learn?

1. Explicit user preferences. Ratings or responses
2. Implicit user preferences. E.g., clicks or viewing time
3. Content. E.g., based on text similarity techniques

Recommender Systems

How do recommender systems learn?

1. Explicit user preferences. Ratings or responses
2. Implicit user preferences. E.g., clicks or viewing time
3. Content. E.g., based on text similarity techniques

Recommender Systems

How do recommender systems learn?

1. Explicit user preferences. Ratings or responses
2. Implicit user preferences. E.g., clicks or viewing time
3. Content. E.g., based on text similarity techniques

Recommender Systems

Types of recommender systems (Locherbach & Trilling, 2018; Möller et al., 2018; Wieland et al., 2021)

1. 'Basic' knowledge-base recommender systems
2. Content-based recommender systems
3. Collaborative recommenders

Knowledge-based RecSys

Knowledge-based recommender system

When to use?

- To overcome the **cold start problem**; when we do not have ratings of individual users.
- Simple model. It does not rely on user's explicit or implicit ratings, but on specific queries.
- Typical use case: Real-estate. Buying a house is, for most families, a rare/ single event.

Knowledge-based recommender system

When to use?

- To overcome the **cold start problem**; when we do not have ratings of individual users.
- Simple model. It does not rely on user's explicit or implicit ratings, but on specific queries.
- Typical use case: Real-estate. Buying a house is, for most families, a rare/ single event.

Knowledge-based recommender system

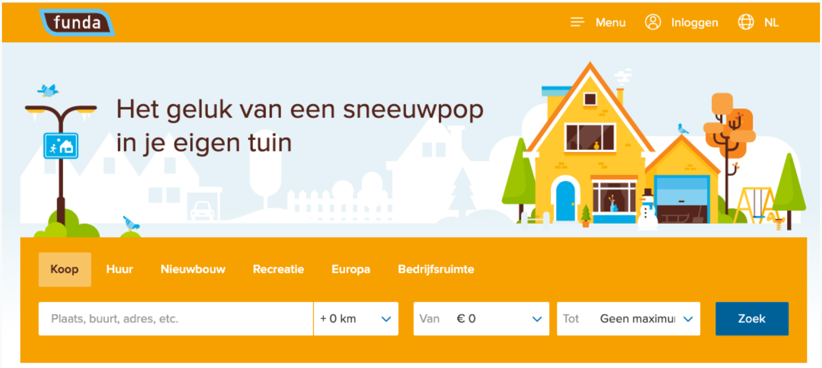
When to use?

- To overcome the **cold start problem**; when we do not have ratings of individual users.
- Simple model. It does not rely on user's explicit or implicit ratings, but on specific queries.
- Typical use case: Real-estate. Buying a house is, for most families, a rare/ single event.

Knowledge-based recommender system

When to use?

- To overcome the **cold start problem**; when we do not have ratings of individual users.
- Simple model. It does not rely on user's explicit or implicit ratings, but on specific queries.
- Typical use case: Real-estate. Buying a house is, for most families, a rare/ single event.


funda
Menu
Inloggen
NL
Het geluk van een sneeuwpop
in je eigen tuin
Koop
Huur
Nieuwbouw
Recreatie
Europa
Bedrijfsruimte
Plaats, buurt, adres, etc.
+ 0 km
Van
€ 0
Tot
Geen maximu
Zoek

Use case: ImDb database

	genres	title	tagline	release_date	vote_average	vote_count
0	[action, adventure, fantasy, science fiction]	Avatar	Enter the World of Pandora.	2009-12-10	7.2	11800
1	[adventure, fantasy, action]	Pirates of the Caribbean: At World's End	At the end of the world, the adventure begins.	2007-05-19	6.9	4500
2	[action, adventure, crime]	Spectre	A Plan No One Escapes	2015-10-26	6.3	4466
3	[action, crime, drama, thriller]	The Dark Knight Rises	The Legend Ends	2012-07-16	7.6	9106
4	[action, adventure, science fiction]	John Carter	Lost in our world, found in another.	2012-03-07	6.1	2124
...						

What are relevant variables to use in a knowledge-based recommender system?

Knowledge-based recommender system

How can we work with user input without a front-end (such as the website of funda?

→ enter python's native input() function.

```
print("What is your favorite movie genre?")
genre = input()
```

```
what is your favorite movie genre?
[...]
```

Knowledge-based recommender system

How can we work with user input without a front-end (such as the website of funda?

$\rightarrow$ enter python's native input() function.

```
print("What is your favorite movie genre?")
genre = input()
```

```
what is your favorite movie genre?
[...]
```

Knowledge-based recommender system

How can we work with user input without a front-end (such as the website of funda?

→ enter python's native input() function.

```
print("What is your favorite movie genre?")
genre = input()
```

```
what is your favorite movie genre?
[...]
```

Knowledge-based recommender system

How can we work with user input without a front-end (such as the website of funda?

→ enter python's native input() function.

```
print("What is your favorite movie genre?")
genre = input()
```

```
what is your favorite movie genre?
[...]
```

Improving knowledge-based recommender system

When to use?

- It is important to think about ways to make the recommendation relevant for individuals
- Do you have more information in your db that make your top-listed recommendations as relevant as possible?

Improving knowledge-based recommender system

When to use?

- It is important to think about ways to make the recommendation relevant for individuals
- Do you have more information in your db that make your top-listed recommendations as relevant as possible?

Improving knowledge-based recommender system

When to use?

- It is important to think about ways to make the recommendation relevant for individuals
- Do you have more information in your db that make your top-listed recommendations as relevant as possible?

Improving knowledge-based recommender system

When to use?

- It is important to think about ways to make the recommendation relevant for individuals
- Do you have more information in your db that make your top-listed recommendations as relevant as possible?

Content-based RecSys

Content-based systems

- Recommends items based on user's profiles.
- Profiles are based on e.g., ratings, and represents user's tasts/ preferences.
  - For example: how often a user has clicked on, or liked, a movie
- Recommendation is based on similarity beween items in the content.
  - Content is here: e.g., genre, tags, plot, authors, directors, location, etc.

Content-based systems

- Recommends items based on user's profiles.
- Profiles are based on e.g., ratings, and represents user's tasts/ preferences.
  - For example, how often a user has clicked on, or liked, a movie.
- Recommendation is based on similarity beween items in the content.
  - Content is here: e.g., genre, tags, plot, authors, directors, location, etc.

Content-based systems

- Recommends items based on user's profiles.
- Profiles are based on e.g., ratings, and represents user's tasts/ preferences.
  - For example, how often a user has clicked on, or liked, a movie.
- Recommendation is based on similarity beween items in the content.
  - Content is here: e.g., genre, tags, plot, authors, directors, location, etc.

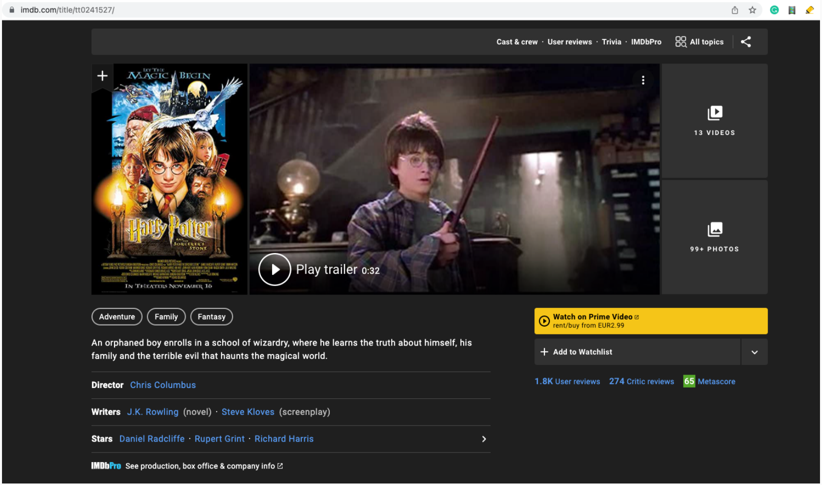
Content-based systems

- Recommends items based on user's profiles.
- Profiles are based on e.g., ratings, and represents user's tasts/ preferences.
  - For example, how often a user has clicked on, or liked, a movie.
- Recommendation is based on similarity beween items in the content.
  - Content is here: e.g., genre, tags, plot, authors, directors, location, etc.

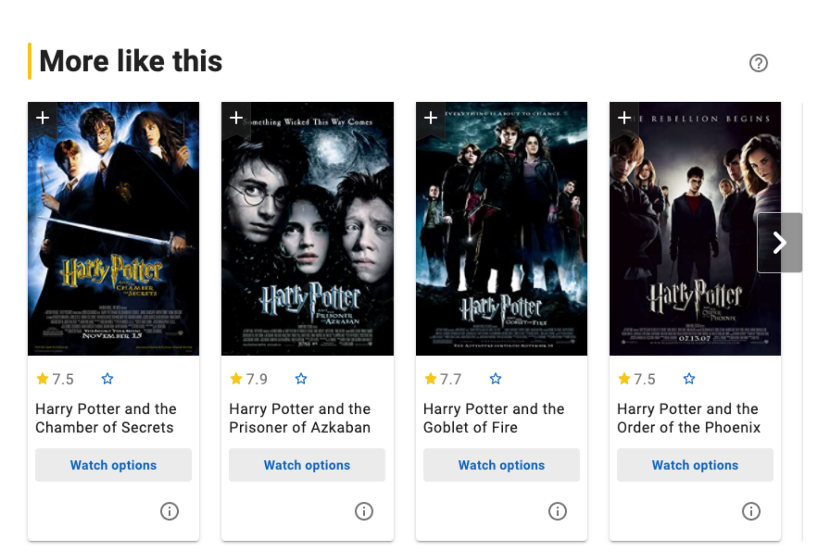
Content-based systems

- Recommends items based on user's profiles.
- Profiles are based on e.g., ratings, and represents user's tasts/ preferences.
  - For example, how often a user has clicked on, or liked, a movie.
- Recommendation is based on similarity beween items in the content.
  - Content is here: e.g., genre, tags, plot, authors, directors, location, etc.

Example of a content-based recsys



Example of a content-based recsys



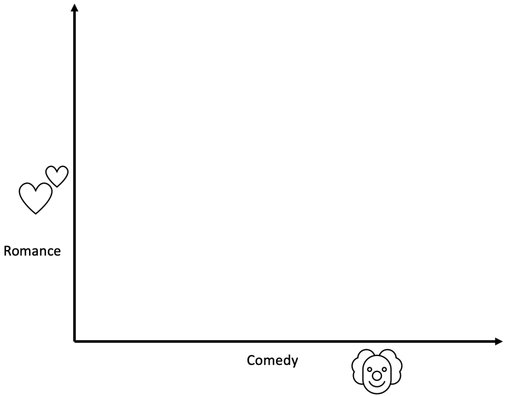
Use case: ImDb database

Let's have a look at our use-case again.

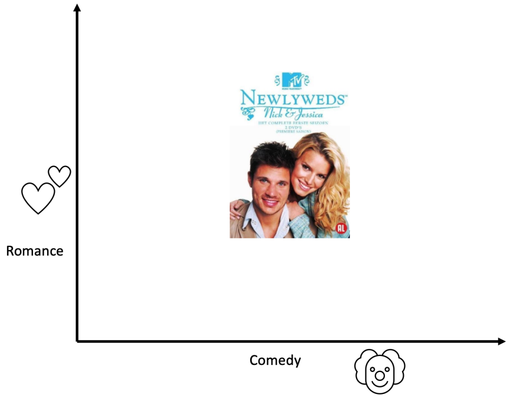
Are there attributes that you could use to estimate similarity in movies?

	genres	title	tagline	release_date	vote_average	vote_count
0	[action, adventure, fantasy, science fiction]	Avatar	Enter the World of Pandora.	2009-12-10	7.2	11800
1	[adventure, fantasy, action]	Pirates of the Caribbean: At World's End	At the end of the world, the adventure begins.	2007-05-19	6.9	4500
2	[action, adventure, crime]	Spectre	A Plan No One Escapes	2015-10-26	6.3	4466
3	[action, crime, drama, thriller]	The Dark Knight Rises	The Legend Ends	2012-07-16	7.6	9106
4	[action, adventure, science fiction]	John Carter	Lost in our world, found in another.	2012-03-07	6.1	2124
...						

Similarity between movies

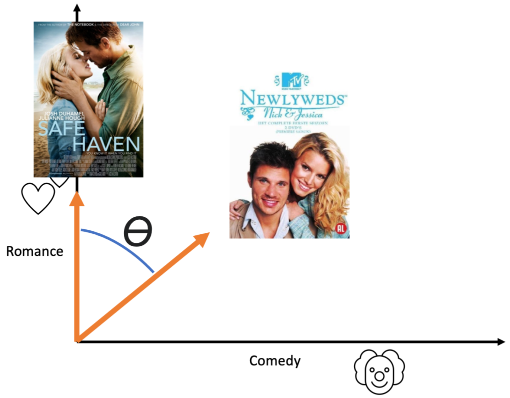


Similarity between movies

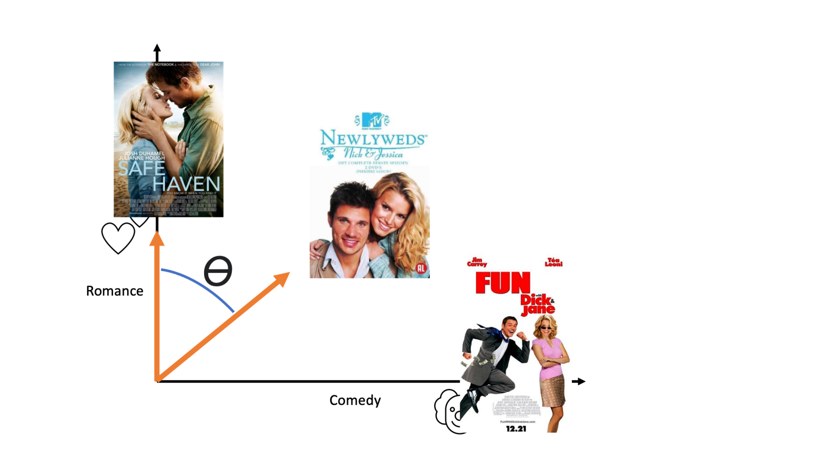


Similarity between movies

Similarity between movies



Similarity between movies



Let's put this in code!

First, create a create a toy dataset.

```
data = data.sample(6)
data[['title', 'genres']]
```

Out[239]:

	title	genres
0	Avatar	action adventure fantasy science fiction
1	Pirates of the Caribbean: At World's End	adventure fantasy action
2	Spectre	action adventure crime
3	The Dark Knight Rises	action crime drama thriller
4	John Carter	action adventure science fiction

Let’s put this in code!

First, create a create a toy dataset.

```
data = data.sample(6)
data[['title', 'genres']]
```

Out[239]:

	title	genres
0	Avatar	action adventure fantasy science fiction
1	Pirates of the Caribbean: At World's End	adventure fantasy action
2	Spectre	action adventure crime
3	The Dark Knight Rises	action crime drama thriller
4	John Carter	action adventure science fiction

Let's put this in code!

First, create a create a toy dataset.

```
data = data.sample(6)
data[['title', 'genres']]
```

Out[239]:

	title	genres
0	Avatar	action adventure fantasy science fiction
1	Pirates of the Caribbean: At World's End	adventure fantasy action
2	Spectre	action adventure crime
3	The Dark Knight Rises	action crime drama thriller
4	John Carter	action adventure science fiction

Let's put this in code!

First, create a create a toy dataset.

```
data = data.sample(6)
data[['title', 'genres']]
```

Out[239]:

	title	genres
0	Avatar	action adventure fantasy science fiction
1	Pirates of the Caribbean: At World's End	adventure fantasy action
2	Spectre	action adventure crime
3	The Dark Knight Rises	action crime drama thriller
4	John Carter	action adventure science fiction

Feature selection and vectorization

Let's assume we want to calculate similiarity based on genres. Therefore, we need to vectorize this column.

```
tfidf = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf.fit_transform(data['genres'])
```

Note that we use tfidf vectorizer here, but we you might opt for a different one.

Feature selection and vectorization

Let's assume we want to calculate similiarity based on genres. Therefore, we need to vectorize this column.

```
tfidf = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf.fit_transform(data['genres'])
```

Note that we use tfidf vectorizer here, but we you might opt for a different one.

Calculate similiarity

Now, let's calculate cosine similarity scores between the genre attributes of the movies

```
from sklearn.metrics.pairwise import cosine_similarity

sim = cosine_similarity(tfidf_matrix)
```

This returns an array of the similiarity scores between each movie and all other movies.

Calculate similiarity

Now, let's calculate cosine similarity scores between the genre attributes of the movies

```
from sklearn.metrics.pairwise import cosine_similarity

sim = cosine_similarity(tfidf_matrix)
```

This returns an array of the similiarity scores between each movie and all other movies.

Cosine Similairity

Let's inspect the output...

```
print(sim)
[[1.         0.34503493 0.         0.         0.40824829 0.
[0.34503493 1.         0.16581288 0.         0.28171984 0.29
[0.         0.16581288 1.         0.25964992 0.         0.56
[0.         0.         0.25964992 1.         0.44115109 0.45
[0.40824829 0.28171984 0.         0.44115109 1.         0.
[0.         0.29130219 0.56921261 0.4561563  0.         1.
```

	title	Avatar	Pirates of the Caribbean: At World's End	Spectre	The Dark Knight Rises	John Carter
	genre	action adventure fantasy science fiction	adventure fantasy action	action adventure crime	action crime drama thriller	action adventure science fiction
title	genres					
Avatar	action adventure fantasy science fiction	1.000000	0.691870	0.315126	0.084696	0.859850
Pirates of the Caribbean: At World's End	adventure fantasy action	0.691870	1.000000	0.455470	0.122417	0.366490
Spectre	action adventure crime	0.315126	0.455470	1.000000	0.473354	0.366490
The Dark Knight Rises	action crime drama thriller	0.084696	0.122417	0.473354	1.000000	0.098501
John Carter	action adventure science fiction	0.859850	0.366490	0.366490	0.098501	1.000000

Based on this overview, you may assume that users that like *Avatar*, may be interested in *John Carter*.

If you want to convert output of cosine_similarity to a df type of object, see here

https://github.com/annekroon/CCS-2/blob/main/week06/cosine_to_df.md

title	genres	Avatar	Pirates of the Caribbean: At World's End	Spectre	The Dark Knight Rises	John Carter
	genre	action adventure fantasy science fiction	adventure fantasy action	action adventure crime	action crime drama thriller	action adventure science fiction
Avatar	action adventure fantasy science fiction	1.000000	0.691870	0.315126	0.084696	0.859850
Pirates of the Caribbean: At World's End	adventure fantasy action	0.691870	1.000000	0.455470	0.122417	0.366490
Spectre	action adventure crime	0.315126	0.455470	1.000000	0.473354	0.366490
The Dark Knight Rises	action crime drama thriller	0.084696	0.122417	0.473354	1.000000	0.098501
John Carter	action adventure science fiction	0.859850	0.366490	0.366490	0.098501	1.000000

Based on this overview, you may assume that users that like *Avatar*, may be interested in *John Carter*.

If you want to convert output of cosine_similarity to a df type of object, see here https://github.com/annekroon/CCS-2/blob/main/week06/cosine_to_df.md

	title	Avatar	Pirates of the Caribbean: At World's End	Spectre	The Dark Knight Rises	John Carter
	genre	action adventure fantasy science fiction	adventure fantasy action	action adventure crime	action crime drama thriller	action adventure science fiction
title	genres					
Avatar	action adventure fantasy science fiction	1.000000	0.691870	0.315126	0.084696	0.859850
Pirates of the Caribbean: At World's End	adventure fantasy action	0.691870	1.000000	0.455470	0.122417	0.366490
Spectre	action adventure crime	0.315126	0.455470	1.000000	0.473354	0.366490
The Dark Knight Rises	action crime drama thriller	0.084696	0.122417	0.473354	1.000000	0.098501
John Carter	action adventure science fiction	0.859850	0.366490	0.366490	0.098501	1.000000

Based on this overview, you may assume that users that like *Avatar*, may be interested in *John Carter*.

If you want to convert output of cosine_similarity to a df type of object, see here
https://github.com/annekroon/CCS-2/blob/main/week06/cosine_to_df.md

Feature selection and engineering

- Feature engineering is very important here. What qualities or attributes do we want to incorporate?
- Think about this critically. Instead of using genres or authors, you could think about crafting features yourself, such as topics.

Feature selection and engineering

- Feature engineering is very important here. What qualities or attributes do we want to incorporate?
- Think about this critically. Instead of using genres or authors, you could think about crafting features yourself, such as topics.

Feature selection and engineering

- Feature engineering is very important here. What qualities or attributes do we want to incorporate?
- Think about this critically. Instead of using genres or authors, you could think about crafting features yourself, such as topics.

Benefits

- Content-based recommender systems can be very efficient...
- They are often part of more complex recommender systems that leverage (deep) supervised learning

Drawbacks

- Features that are not part of the user profile will be neglected; e.g., if the user does not like Super Hero movies, the recommender system wil never recommend this.
- does not use the power community data. Recommendations may be obvious (e.g., *Harry Potter* recommendation when you like *Lord of the Rings*)

Benefits

- Content-based recommender systems can be very efficient...
- They are often part of more complex recommender systems that leverage (deep) supervised learning

Drawbacks

- Features that are not part of the user profile will be neglected; e.g., if the user does not like Super Hero movies, the recommender system wil never recommend this.
- does not use the power community data. Recommendations may be obvious (e.g., *Harry Potter* recommendation when you like *Lord of the Rings*)

Benefits

- Content-based recommender systems can be very efficient...
- They are often part of more complex recommender systems that leverage (deep) supervised learning

Drawbacks

- Features that are not part of the user profile will be neglected; e.g., if the user does not like Super Hero movies, the recommender system wil never recommend this.
- does not use the power community data. Recommendations may be obvious (e.g., *Harry Potter* recommendation when you like *Lord of the Rings*)

Benefits

- Content-based recommender systems can be very efficient...
- They are often part of more complex recommender systems that leverage (deep) supervised learning

Drawbacks

- Features that are not part of the user profile will be neglected; e.g., if the user does not like Super Hero movies, the recommender system wil never recommend this.
- does not use the power community data. Recommendations may be obvious (e.g., *Harry Potter* recommendation when you like *Lord of the Rings*)

Collaborative RecSys

Collaborative recommender systems

- Tries to overcome some of the limitations of content-based systems
- Leverages the power of the community, tries to give relevant, but also suprising recommendations.

Collaborative recommender systems

- Tries to overcome some of the limitations of content-based systems
- Leverages the power of the community, tries to give relevant, but also suprising recommendations.

Group assignment

https://www.kaggle.com/saurabhbagchi/books-dataset

References

References

Kim, H. S., Yang, S., Kim, M., Hemenway, B., Ungar, L., & Cappella, J. N. (2019). An experimental study of recommendation algorithms for tailored health communication. *Computational Communication Research*, *1*(1), 103–129. https://doi.org/10.5117/ccr2019.1.005.sukk

Locherbach, F., & Trilling, D. (2018). 3bij3: A framework for testing effects of recommender systems on news exposure. *Proceedings - IEEE 14th International Conference on eScience, e-Science 2018*, 350–351. https://doi.org/10.1109/eScience.2018.00093

Möller, J., Trilling, D., Helberger, N., & van Es, B. (2018). Do not blame it on the algorithm: an empirical assessment of multiple recommender systems and their impact on content diversity. *Information Communication and Society*, *21*(7), 959–977. https://doi.org/10.1080/1369118X.2018.1444076

Wieland, M., Von Nordheim, G., & Kleinen-Von Königslöw, K. (2021). One recommender fits all? An exploration of user satisfaction with text-based news recommender systems. *Media and Communication*, *9*(4), 208–221. https://doi.org/10.17645/mac.v9i4.4241