



Introduction to Recommendation Systems

Jin Yao CHIN

Nanyang Technological University
Singapore

Outline

- ▷ **WHERE** can we find recommendation systems?
- ▷ **WHY** do we need recommendation systems?
- ▷ **WHAT** is recommendation?
- ▷ **HOW** do we make recommendations?

- ▷ Different **types** of recommendation systems
- ▷ **Challenges** encountered by recommendation systems



1.

WHERE can we find
recommendation systems?

Recommendation Systems – WHERE?



YouTube



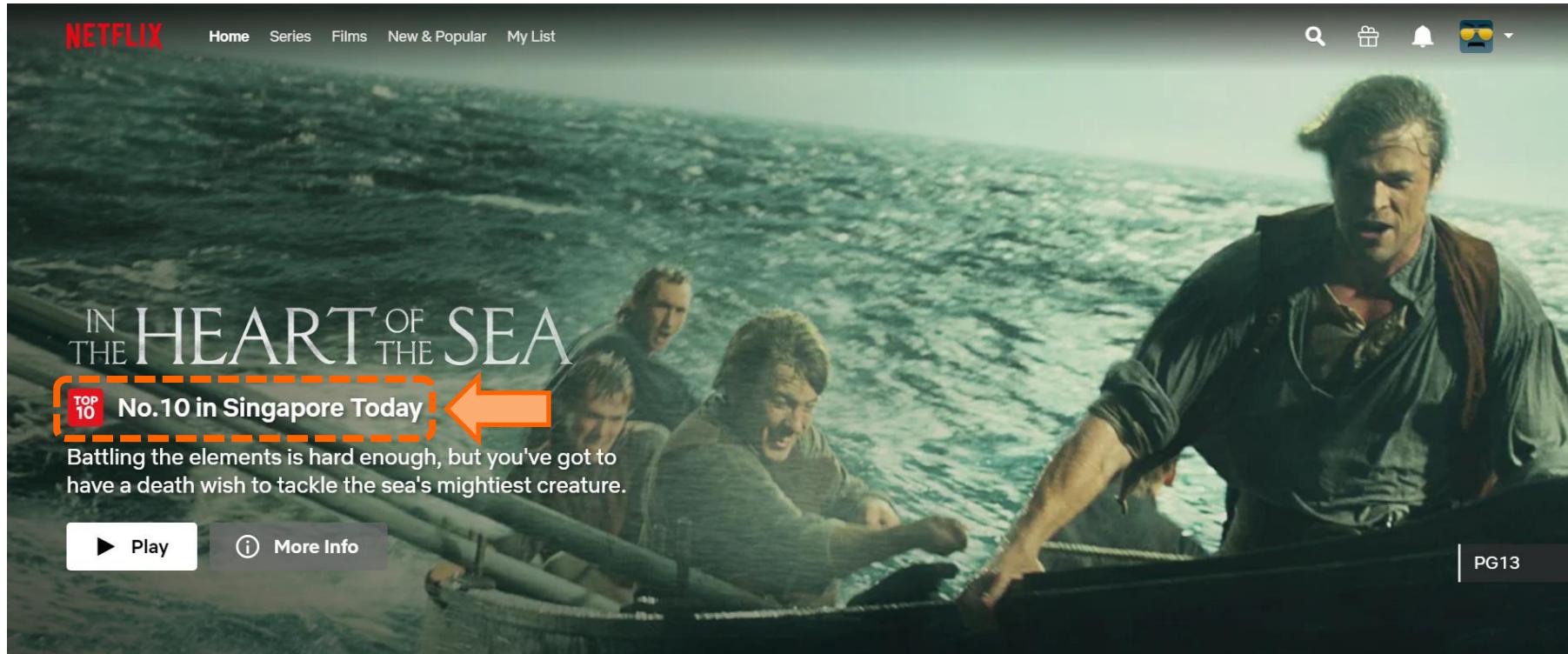
FOURSQUARE

淘宝网
Taobao.com



Google Maps

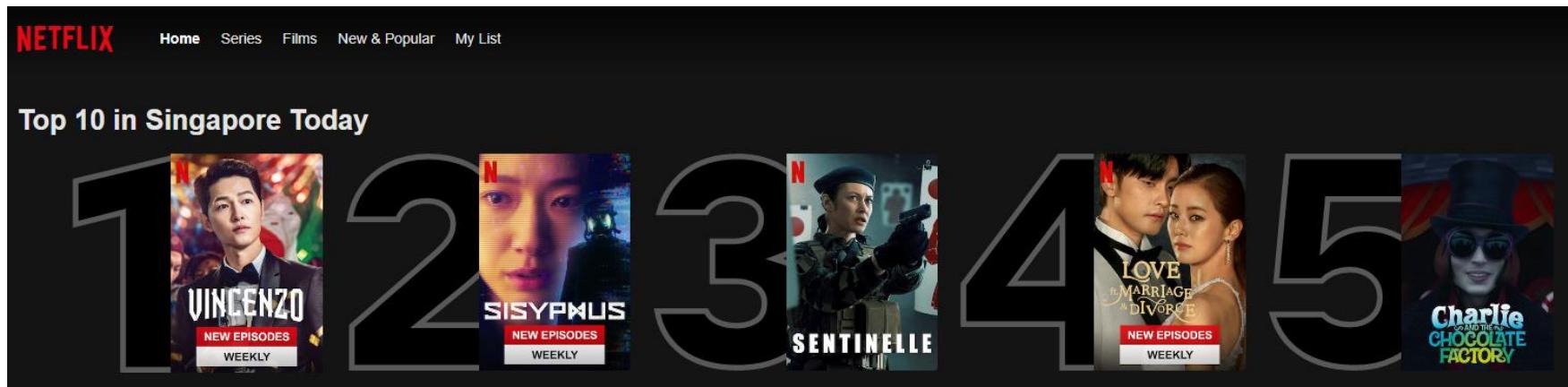
Recommendation Systems – WHERE?



▷ E.g., Netflix

- Netflix might try to recommend a show which is very popular based on your current *location* (E.g., No. 10 in Singapore Today)

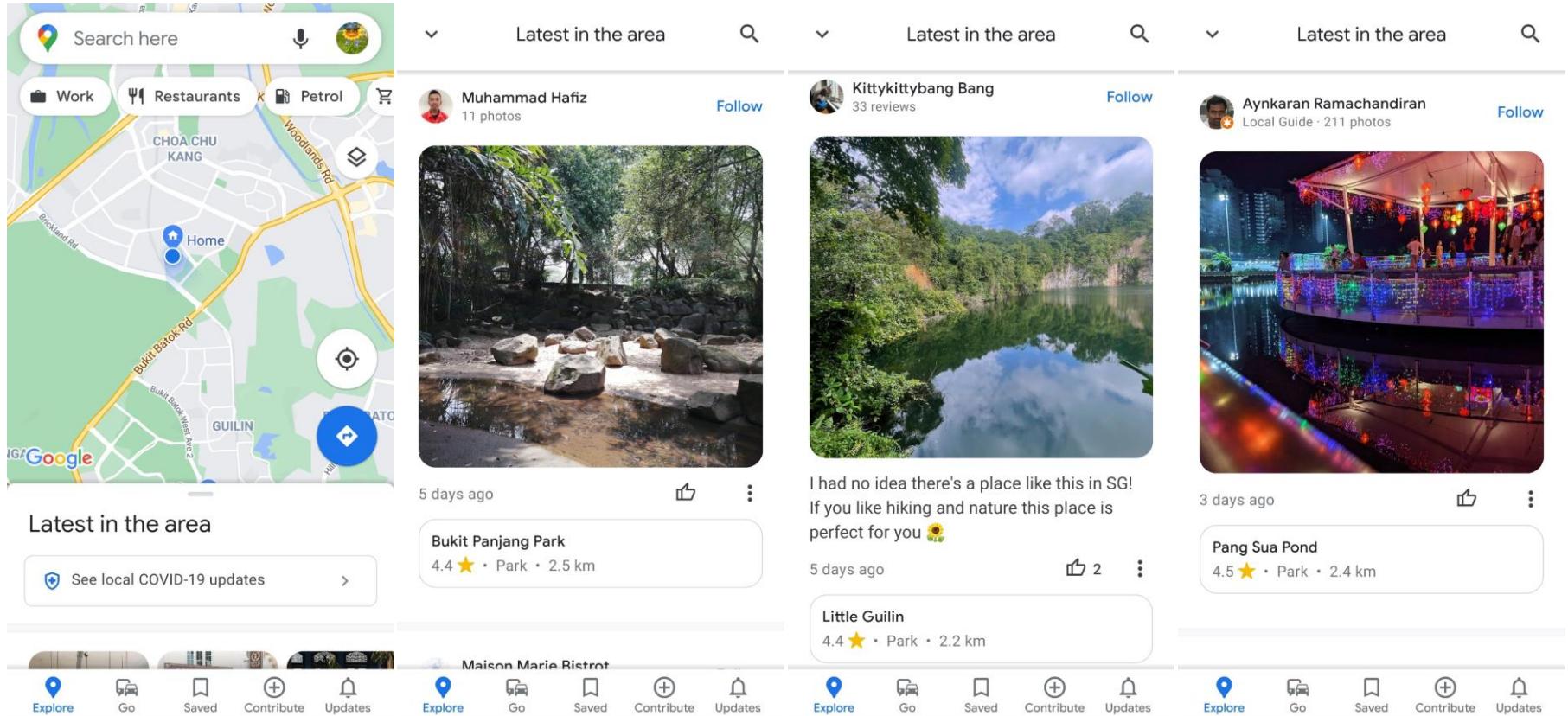
Recommendation Systems – WHERE?



▷ E.g., Netflix

- Netflix might try to recommend the Top 10 shows based on your current *location*
- However, note that these are *not* personalized recommendations
 - Based on other users located in *Singapore*
 - Everyone in *Singapore* gets the same recommendations

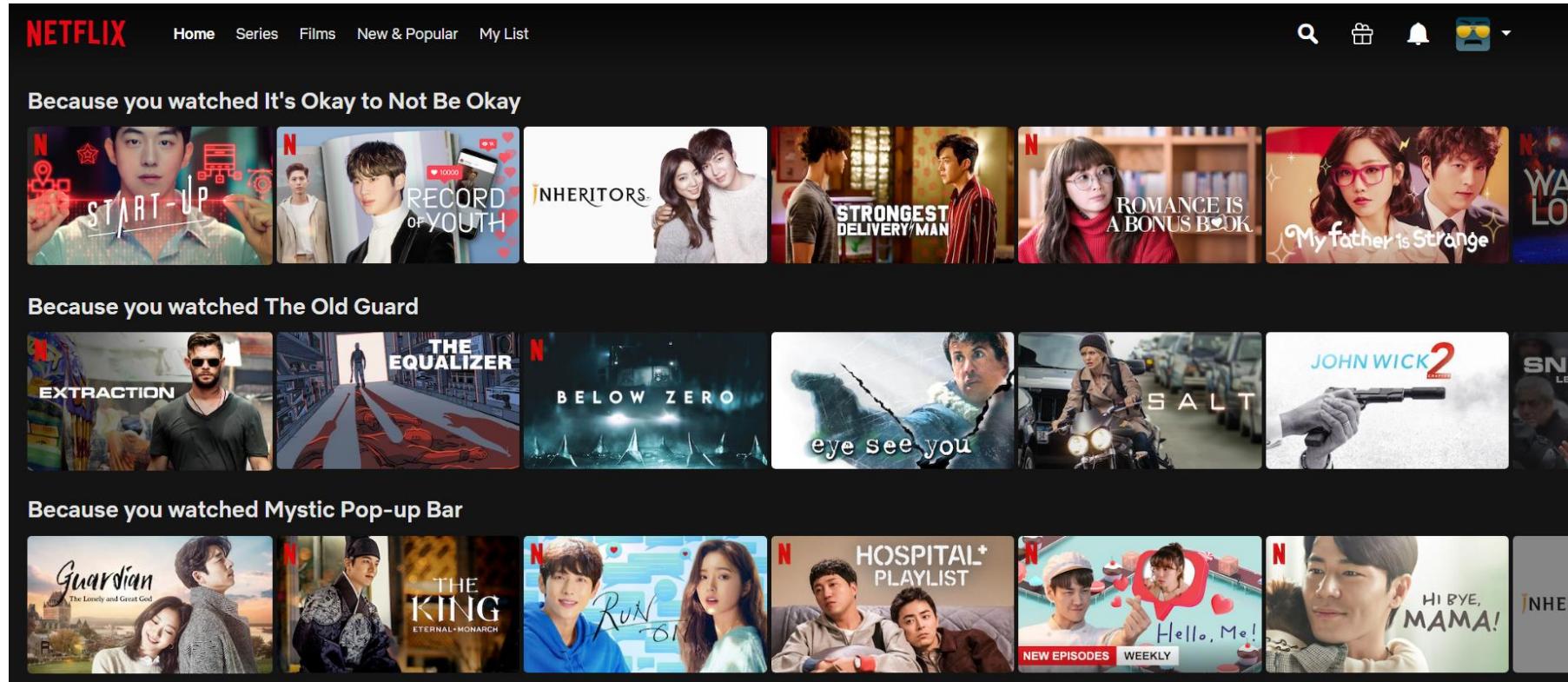
Recommendation Systems – WHERE?



▷ E.g., Google Maps

- Given your current *location*, Google Maps might recommend popular locations *near you*

Recommendation Systems – WHERE?



▷ E.g., Netflix

- Shows are often recommended based on your viewing *history* (i.e., shows that you have watched previously)

Recommendation Systems – WHERE?

The screenshot shows the Amazon Singapore website. At the top, there's a navigation bar with links for All, Best Sellers, Today's Deals, Home, Prime, Customer Service, Home Improvement, Electronics, New Releases, Books, Gift Ideas, and Computers. Below this is a secondary navigation bar for Books, with categories like Arts & Photography, Business & Investing, Children's Books, Fiction, History, Mystery & Suspense, School Books, and Travel & Holiday. The main content area displays the product page for 'Artificial Intelligence: A Modern Approach'. The book cover features a grid of various AI-related images, including chess pieces, a skeleton, a landscape, and a portrait of a man. The title 'Artificial Intelligence' is overlaid on the bottom left of the cover. The book is listed as a Hardcover, priced at S\$255.41. It has a rating of 5 stars based on 1 rating. The page also mentions that it is an International product from outside Singapore.

Amazon.sg Hello Select your address All

All Books Arts & Photography Business & Investing Children's Books Fiction History Mystery & Suspense School Books Travel & Holiday

Books > Computing & Internet > Computer Science

Artificial Intelligence: A Modern Approach Hardcover – 10 November 2020

by [Stuart Russell](#) (Author), [Peter Norvig](#) (Author)

★★★★★ 1 rating

[See all formats and editions](#)

Hardcover
S\$255.41

1 New from S\$255.41

International product from outside Singapore [Learn More](#).

The most comprehensive, up-to-date introduction to the theory and practice of artificial intelligence. The long-anticipated revision of Artificial Intelligence: A Modern Approach explores the full breadth and depth of the field of artificial intelligence (AI). The 4th Edition brings readers up to date on the latest technologies, presents concepts in a more unified manner, and offers new or expanded coverage of machine learning, deep learning, transfer learning, multiagent systems, robotics, natural language processing, causality, probabilistic programming, privacy, fairness, and safe AI.

▷ E.g., Amazon

- Let's assume that we are viewing some book in Amazon (e.g., *Artificial Intelligence: A Modern Approach*)

Recommendation Systems – WHERE?

The screenshot shows a product page for 'Artificial Intelligence: A Modern Approach' on Amazon.sg. At the top, there's a navigation bar with categories like All, Best Sellers, Today's Deals, Home, Prime, Customer Service, Home Improvement, Electronics, New Releases, Books, Gift Ideas, and Computers. Below the navigation, a breadcrumb trail shows 'Books > Computing & Internet > Computer Science'. The main product image is a collage of various AI-related icons and figures. To the right of the product details, a large orange arrow points down to a section titled 'Frequently bought together'.

Frequently bought together

Three items are recommended:

- A book cover for 'Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow' by Aurélien Géron.
- A book cover for 'Deep Learning' by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
- A small image of a lizard.

Total Price: S\$389.48

Add all three to Cart

i These items are shipped from and sold by different sellers. Show details

This item: Artificial Intelligence: A Modern Approach by Stuart Russell Hardcover S\$255.41

Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques... by Aurélien Géron Paperback S\$52.52

Deep Learning by Ian Goodfellow Hardcover S\$81.55

▷ E.g., Amazon

- On the same page, Amazon would try to recommend other items (i.e., books) that are *frequently bought together*

Recommendation Systems – WHERE?

The screenshot shows an Amazon product page for the book "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig. The page includes a navigation bar with categories like All, Best Sellers, Today's Deals, Home, Prime, Customer Service, Home Improvement, Electronics, New Releases, Books, Gift Ideas, and Computers. Below the navigation is a breadcrumb trail: Books > Computing & Internet > Computer Science. The main product image is a collage of various AI-related icons and figures. The product title is "Artificial Intelligence: A Modern Approach" Hardcover – 10 November 2020, by Stuart Russell (Author), Peter Norvig (Author). It has a 4.5-star rating from 1 rating. A large orange arrow points to a section titled "Customers who bought this item also bought" which is highlighted with a red dashed border. This section lists five other books: "Reinforcement Learning: An Introduction" by Richard S. Sutton, "Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow" by Aurélien Géron, "Deep Learning" by Ian Goodfellow, "Introduction to Algorithms" by Thomas H. Cormen, and "Deep Learning with Python" by François Fleuret. Each book entry includes its title, author, price, and a small image.

▷ E.g., Amazon

- On the same page, Amazon would try to recommend other items (i.e., books) that were *bought* by ‘similar’ customers

Recommendation Systems – WHERE?

- ▷ Recommendation systems can be found everywhere!
 - Movies, Books, Music, Points of Interests, ...
- ▷ (Literally) an indispensable part of our daily lives



2.

WHY do we need
recommendation systems?

Recommendation Systems – WHY?

1. *Large catalogue*
 - w/ up to millions of items
2. *Beneficial to both the end users & the businesses*



amazon



yelp*

NETFLIX



FOURSQUARE



Google Maps



淘宝网
Taobao.com

Large Catalogue 😞

- ▷ “Amazon has an inventory of about **12 million items** across all its categories and services” [1]
- ▷ Let’s say... I would like to get a new chair (*but I am not entirely sure what kind of a chair would be good*)
 - There could be thousands of chairs being sold
 - Different types of chairs: Office, Dining, ...
 - Different colours, sizes, prices, ...
 - It could be very challenging and tedious to find something ideal!
 - Recommender systems can help to narrow down the options effectively 😊



Large (and expanding) Catalogue 😞

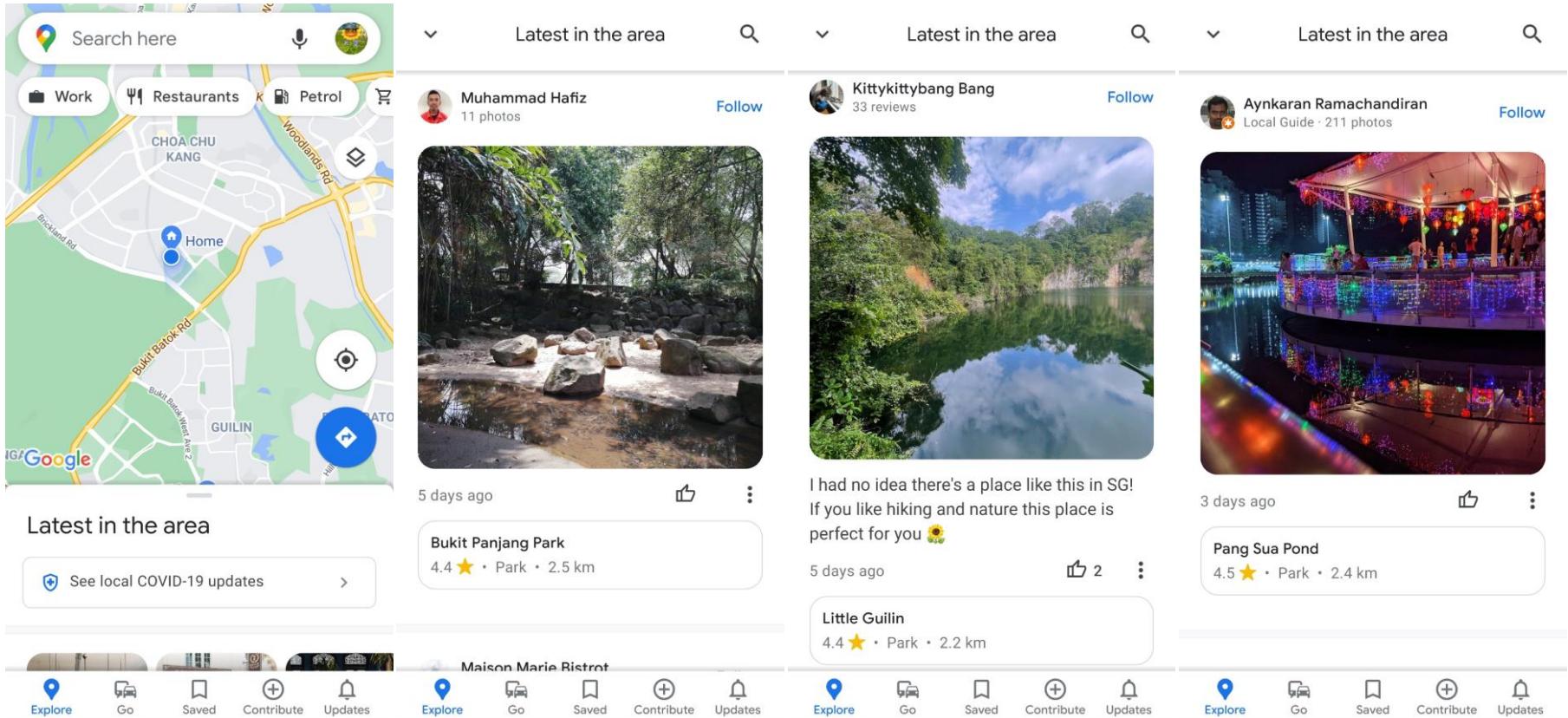
- ▷ Platforms such as YouTube, TikTok, etc. are based on User Generated Content (**UGC**)
- ▷ In contrast to platforms such as Netflix, the catalogue is *expanding rapidly!*



[1] <https://sg.oberlo.com/blog/youtube-statistics> (Retrieved on 08/03/2021)

[2] <https://www.tubefilter.com/2019/05/07/number-hours-video-uploaded-to-youtube-per-minute/> (Retrieved on 08/03/2021)

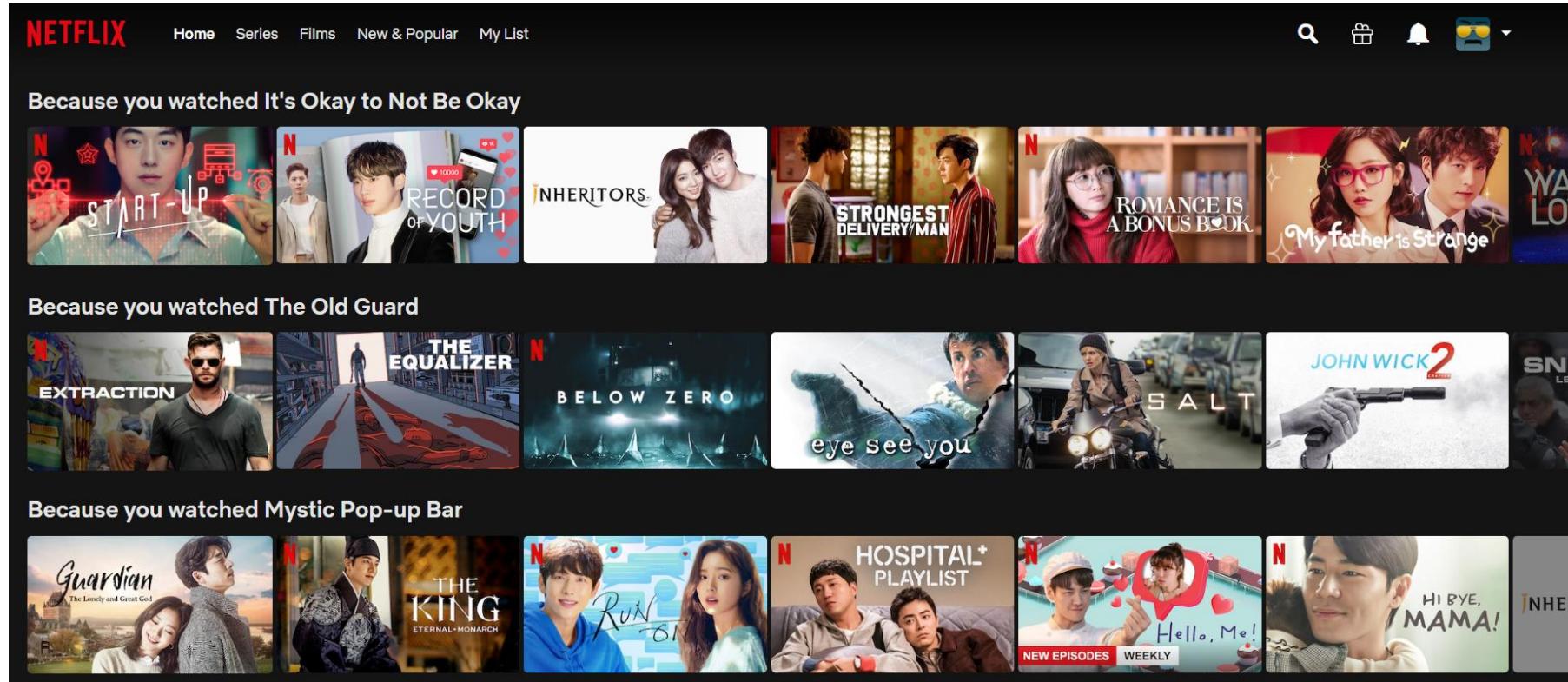
Beneficial to End Users 😊



▷ E.g., Google Maps

- The 'explore' feature assists users in finding *nearby points of interests*
- Users might not be aware of these locations before this!

Beneficial to End Users 😊



▷ E.g., Netflix

- By leveraging your **viewing history**, Netflix could recommend other shows which you are most likely to watch...
- Similarly, you might not be aware of such shows prior to this!

Beneficial to End Users ☺

- ▷ A good recommender system would be akin to having a friend who knows you even better than yourself
 - Assists users in finding *items of interest*
 - Introduces new items to users (*Novelty*)
 - Presents unexpected items to users (*Serendipity*)



Beneficial to Businesses 😊

- ▷ Helps item providers (e.g., sellers, content creators, etc.) deliver their products to its intended audience
- ▷ Improves customer satisfaction
 - Less time spent on browsing / searching
 - Caters to the preferences of different users
- ▷ The underlying recommendation system could be the **key difference** between...
 - 2 platforms having the same catalogue
 - 2 platforms with the same type of products



3. *WHAT* is recommendation?

Recommendation Systems – WHAT?

▷ Where?

- Everywhere!
- Part and parcel of our daily lives

▷ Why?

- Large (and perhaps expanding) catalogues
- Good for both end users & businesses

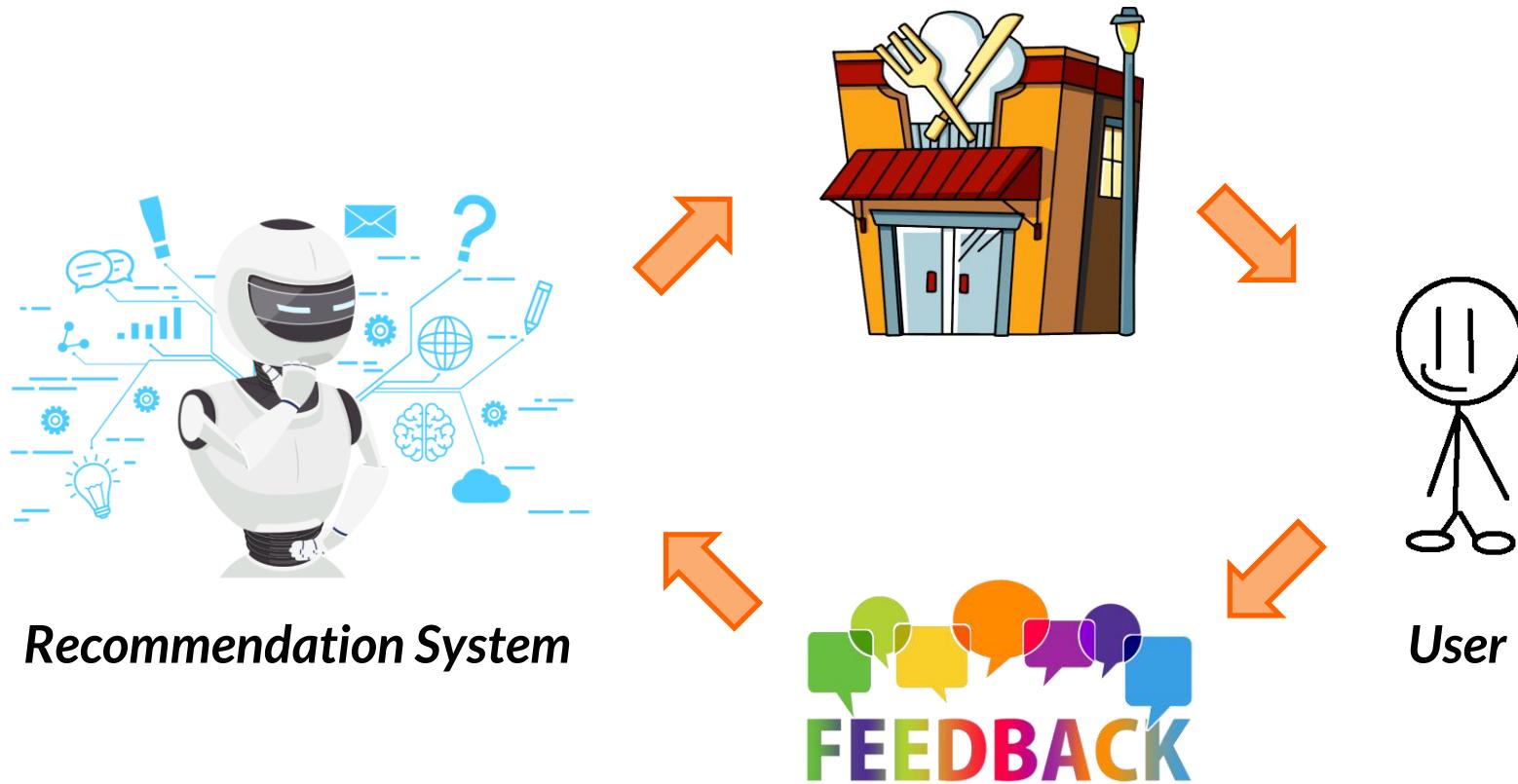
▷ What? → *Formalizing the recommendation problem*

- What are the inputs? Outputs?
- Are we making good (or bad) recommendations?



A (Virtuous) Cycle

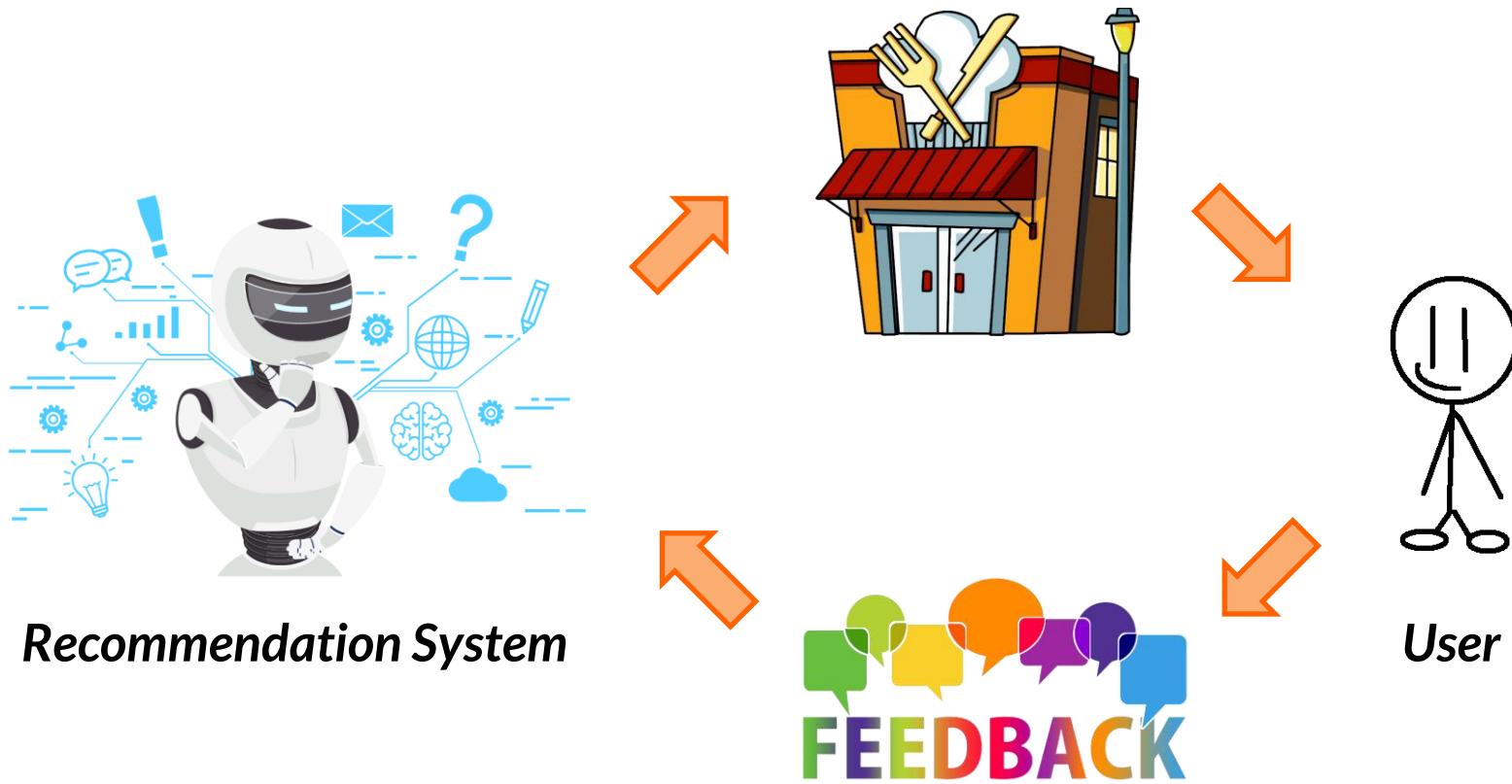
- ▷ Recommendation systems are driven by user interactions



- More interactions → Better awareness of user preferences
- Better awareness of user preferences → Better recommendations
- Better recommendations → More interactions

Recommendations & Feedback

- ▷ Recommendation systems
 - **Outputs:** Items of interest (e.g., Movies, Books, etc.)
 - **Inputs:** User feedback (*Explicit & Implicit*)



Explicit Feedback

- ▷ For example, users can provide a *rating* for an item



calvinnme



17 December 2020 - Published on Amazon.com



Thomas J Ballatore



26 May 2020 - Published on Amazon.com
Verified Purchase



Adam Zhang



2 November 2020 - Published on Amazon.com
Verified Purchase



ZZ



17 August 2020 - Published on Amazon.com
Verified Purchase



by S*** ✓ Verified Purchase



vie*** (SG 🇸🇬) Feb 26, 2021



by Bhaskar V. ✓ Verified Purchase



sui*** (SG 🇸🇬) Feb 25, 2021

Explicit Feedback

- ▷ For example, users can provide a rating for an item
 - Furthermore, some platforms allow users to provide a *review*

 **Solid MI**

By [Zimmer](#) on September 2, 2018

Format: Blu-ray

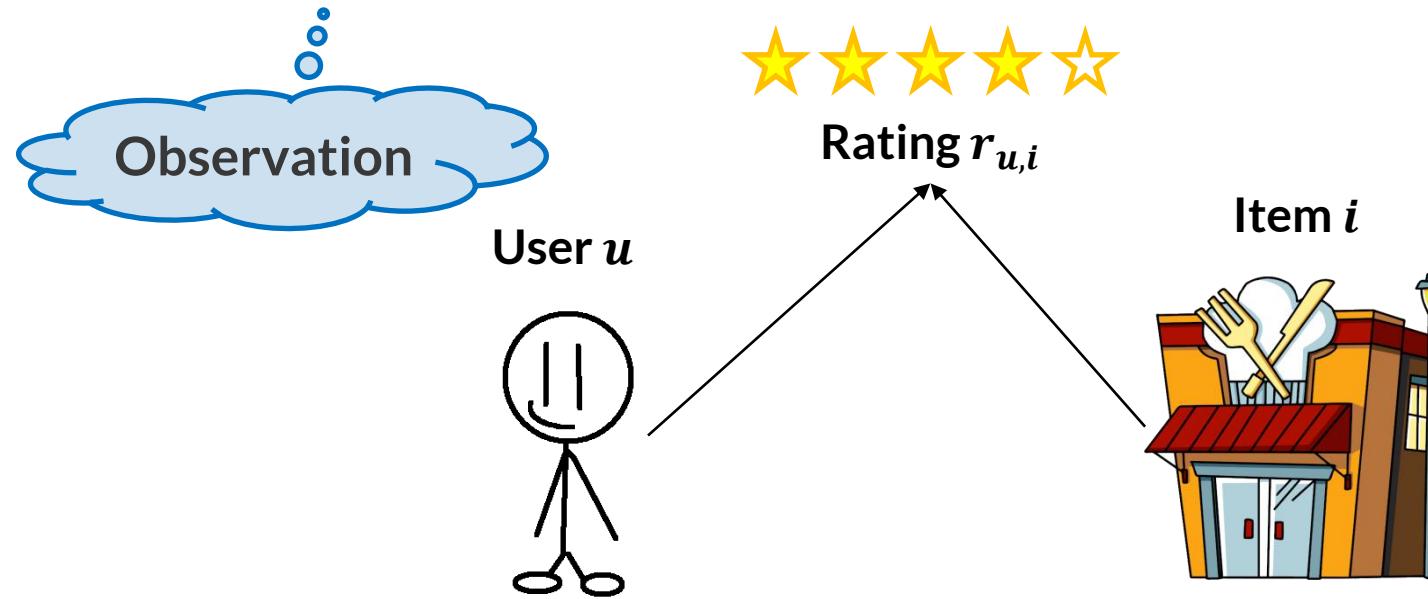
No doubt one of if not the best movie released this year and, just my IMO, in the top 3 Mission films. However im not sure it is quite deserving of the high RT rating it received. It does drag a bit in the second act when Solomon Lane is introduced again. The film needed a truly great scene stealing villain IMO to compete with the great action and Cruise's stunts, and Lane just isnt that interesting. Much has been said of Cavill and his amazing moustache and he's decent but a bit wooden. Great physical presence though. Cruise is solid as usual. Really really enjoyed the first act and the action scenes toward the end were great. The score by Lorne Balfe might just be the best MI score yet. Should have cut the running time a bit tho

Explicit Feedback – Rating Prediction

For each user u , we would like to estimate the rating $\hat{r}_{u,i}$ for any new item i

▷ (Input) Matrix: $R \in \mathbb{R}^{N \times M}$

- N users, M items
- $r_{u,i} = \{1, \dots, 5\}$ if user u has interacted with item i , 0 otherwise



▷ Recommend new items that the user would rate highly

Explicit Feedback – Rating Prediction

- ▷ Given the *observed entries*, recover the *missing entries*
 - $\{r_{ui}\}$: The set of *observed entries*



	5		4
		3	1
	4	2	
			3 5

Explicit Feedback – Rating Prediction

- ▷ Given the *observed entries*, recover the *missing entries*
 - $\{r_{ui}\}$: The set of *observed entries*
 - $\{\hat{r}_{ui}\}$: The set of *missing entries (to be recovered)*



	5	?	4	?
	?	3	?	1
	4	2	?	?
	?	?	3	5



Rating Prediction – Evaluation

- ▷ Are we making good or bad recommendations?
 - Compare *predicted ratings* against *actual ratings*
- ▷ Evaluation (for rating prediction) are usually done using *pointwise metrics*
 - Mean Absolute Error (**MAE**)
 - Root Mean Squared Error (**RMSE**)



A Simple Example



	Avengers: Endgame	Doctor Strange	Benedict Cumberbatch	Thor: Ragnarok
Person 1 (Red)	5	?	4	3
Person 2 (Green)	2	3	?	1
Person 3 (Blue)	4	2	2	4
Person 4 (Yellow)	?	4	3	5

Missing Entries
(in red)



	Avengers: Endgame	Doctor Strange	Benedict Cumberbatch	Thor: Ragnarok
Person 1 (Red)	5	2	4	3
Person 2 (Green)	2	3	5	1
Person 3 (Blue)	4	2	2	4
Person 4 (Yellow)	3	4	3	5



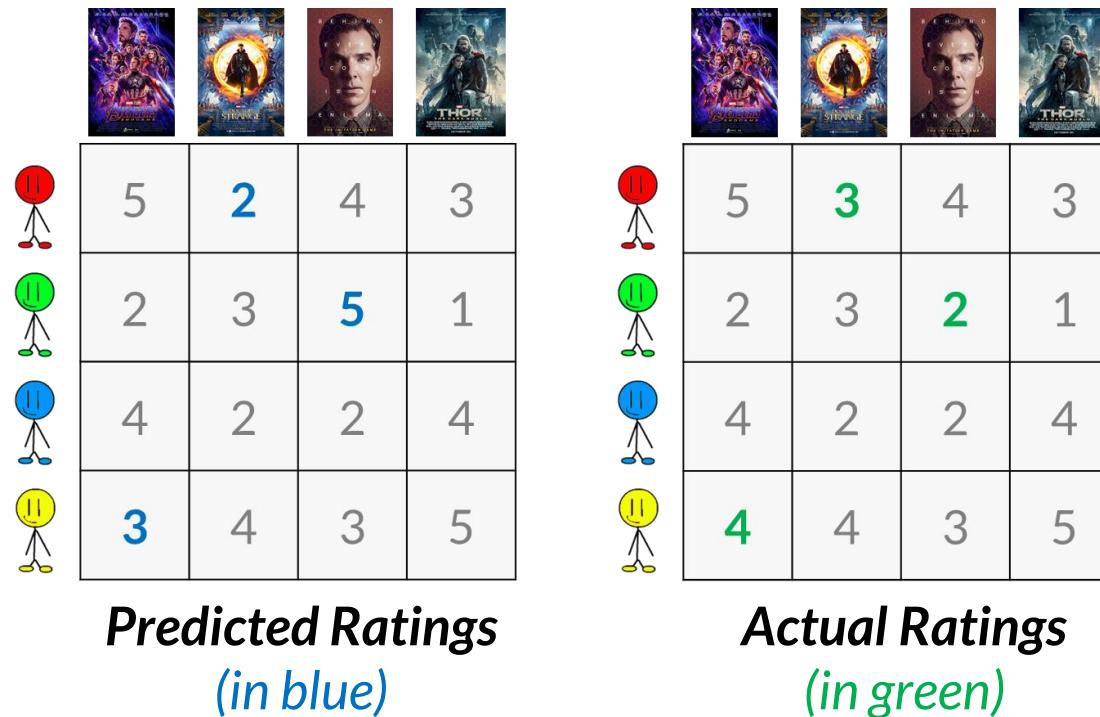
	Avengers: Endgame	Doctor Strange	Benedict Cumberbatch	Thor: Ragnarok
Person 1 (Red)	5	3	4	3
Person 2 (Green)	2	3	2	1
Person 3 (Blue)	4	2	2	4
Person 4 (Yellow)	4	4	3	5



Predicted Ratings
(in blue)

Actual Ratings
(in green)

A Simple Example – MAE & RMSE



$$\triangleright MAE = (|2 - 3| + |5 - 2| + |3 - 4|)/3 \approx 1.66$$

$$\triangleright RMSE = \sqrt{((2 - 3)^2 + (5 - 2)^2 + (3 - 4)^2)/3} \approx 1.91$$

- As compared to MAE, RMSE penalizes *large prediction errors* more (*Errors are squared*)

Implicit Feedback

- ▷ Explicit Feedback (e.g. ratings, reviews, ...)
 - Users need to provide them consciously
 - Might not always be readily available



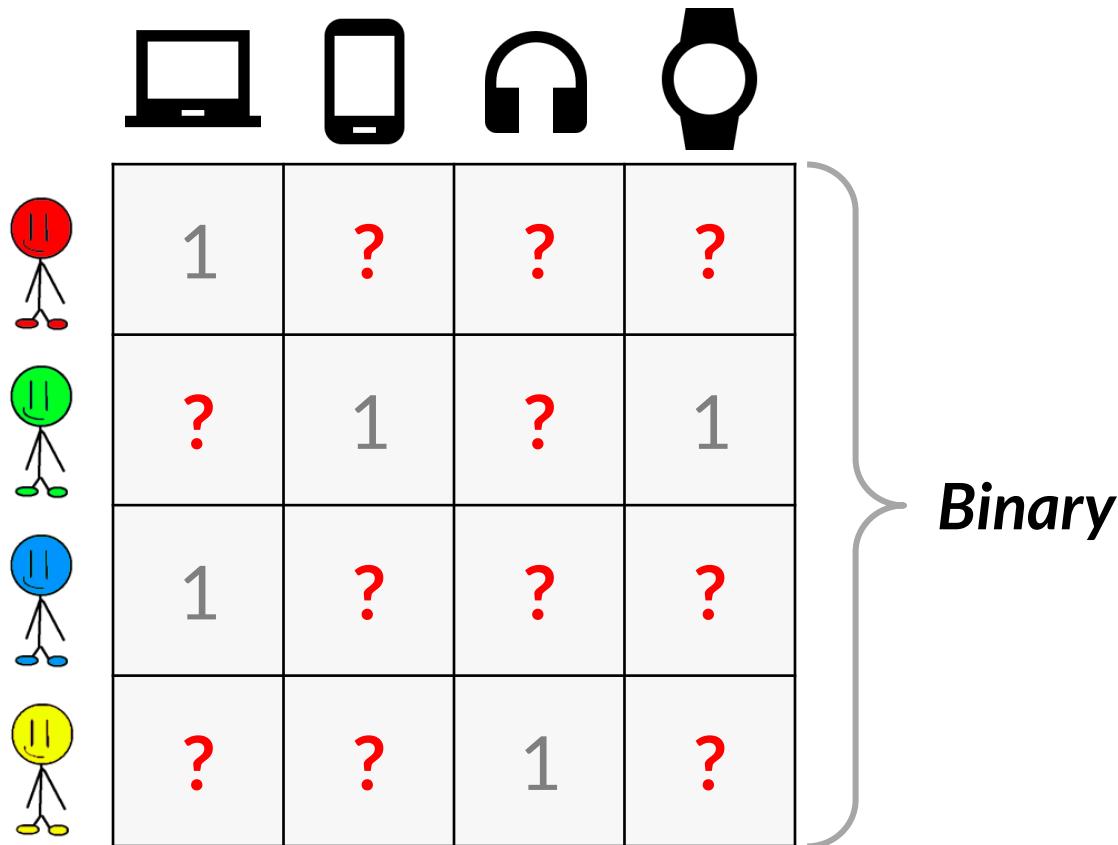
- ▷ Alternative: Implicit Feedback
 - Collected in the background
 - Examples:
 - User has *clicked* on a link
 - User has *watched* a movie
 - User has *viewed* a product
 - User has *purchased* a product



Implicit Feedback – Top-K Recommendation

▷ Given the *observed entries*, predict the likelihoods for the *missing entries*

- $\{r_{ui}\}$: The set of *observed entries*
- $\{\hat{r}_{ui}\}$: The set of *missing entries (to be predicted)*



A Simple Example

	Laptop	Smartphone	Headphones	Watch
User 1	1	?	?	?
User 2	?	1	?	1
User 3	1	?	?	?
User 4	?	?	1	?

*Missing Observations
(in red)*



	Laptop	Smartphone	Headphones	Watch
User 1	1	0.8	0.6	0.1
User 2	0.2	1	0.7	1
User 3	1	0.5	0.3	0.4
User 4	0.8	0.6	1	0.9

*Predicted Likelihoods
(in blue)*



Top-2 Recommendations for

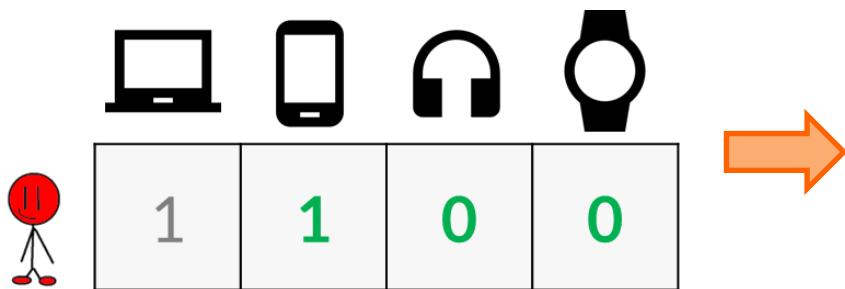


A Simple Example

Top-2 Recommendations for

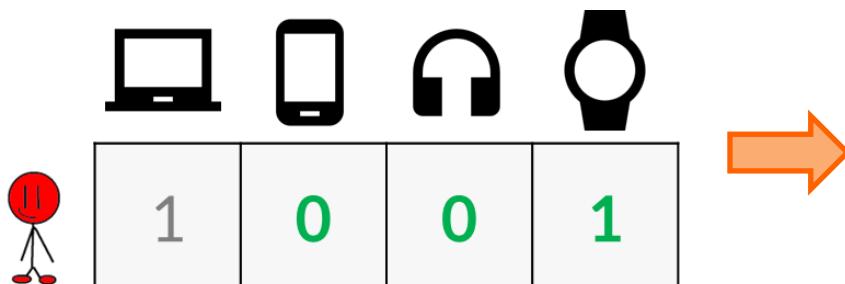


Scenario 1:



- User purchased a phone
- Recommendation is **good!** 😊

Scenario 2:



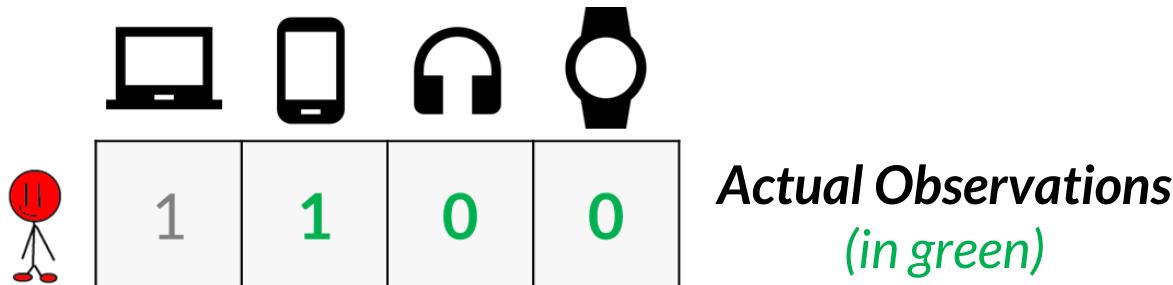
- User purchased a watch
- Recommendation is **bad!** 😞

Top-K Recommendation – Evaluation

- ▷ Are we making good or bad recommendations?
 - Compare *actual observations* against the list of top-K recommended items
- ▷ Evaluation are usually done using *classification metrics* and/or **ranking-based metrics**
 - **Classification:** Precision, Recall, ...
 - **Ranking-based:**
 - Normalized Discounted Cumulative Gain (**nDCG**)
 - Mean Reciprocal Rank (**MRR**)
 - ...



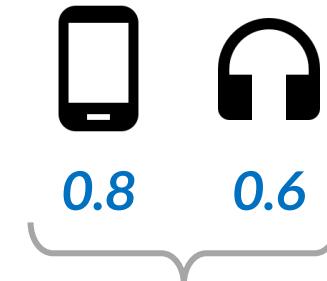
Classification vs Ranking-based



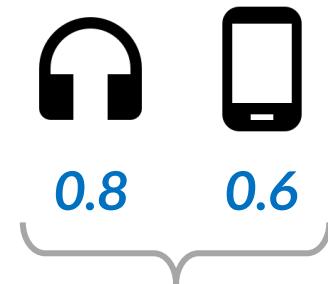
Top-2 Recommendations for



:



or



- For classification-based metrics, Options 1 & 2 are equally good...
 - The phone (*actual purchase*) is in both top-2 lists
 - *Order does not matter*
- For ranking-based metrics, Option 1 is better than Option 2!
 - The phone (*actual purchase*) is ranked above the headphones for Option 1 ☺

Implicit Feedback – Caveat

- ▷ E.g., User did not *watch* a movie
 - Possibility 1: User dislikes that movie
 - Possibility 2: User is *not aware* of such a movie
- ▷ E.g., User has *watched* a movie
 - Does the user like or dislike the movie?
- ▷ E.g., User *purchased* a product
 - Does the user like the product?
 - What if it was bought as a gift for someone else?
- ▷ For *implicit feedback*, observations (or the lack thereof) do not necessarily reflect preferences



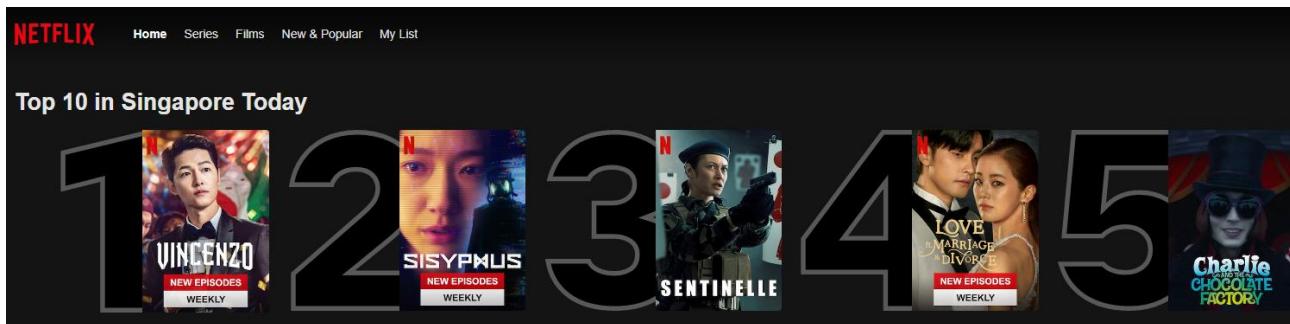
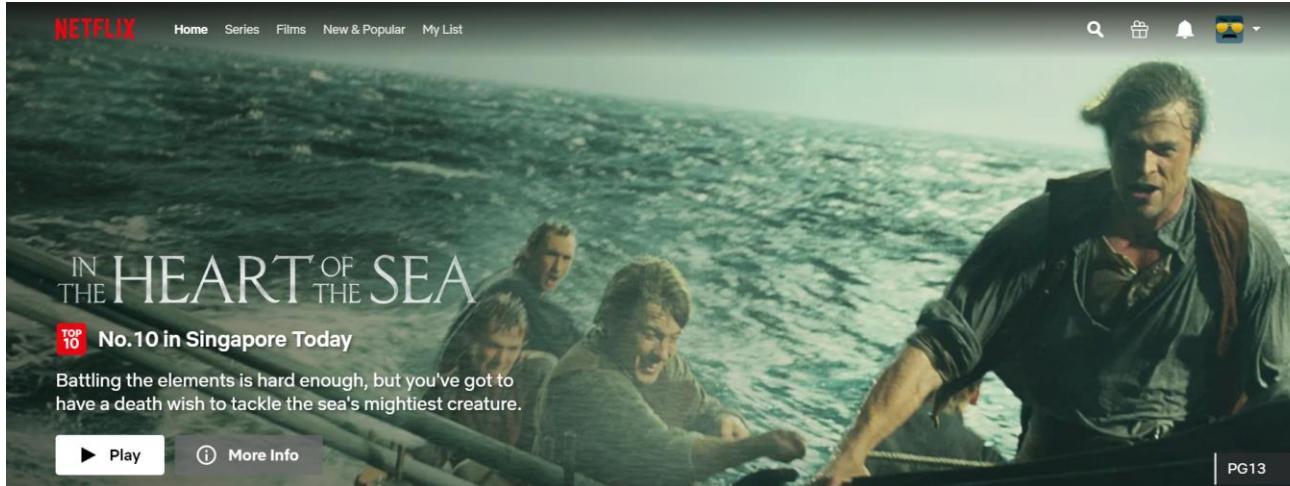
4.

HOW do we make
recommendations?

Making Recommendations

▷ Non-personalized

- Random
- Most Popular (e.g., Top 10 movies)
- Most Recent (e.g., Latest uploads on YouTube)



Making Recommendations

▷ Personalized

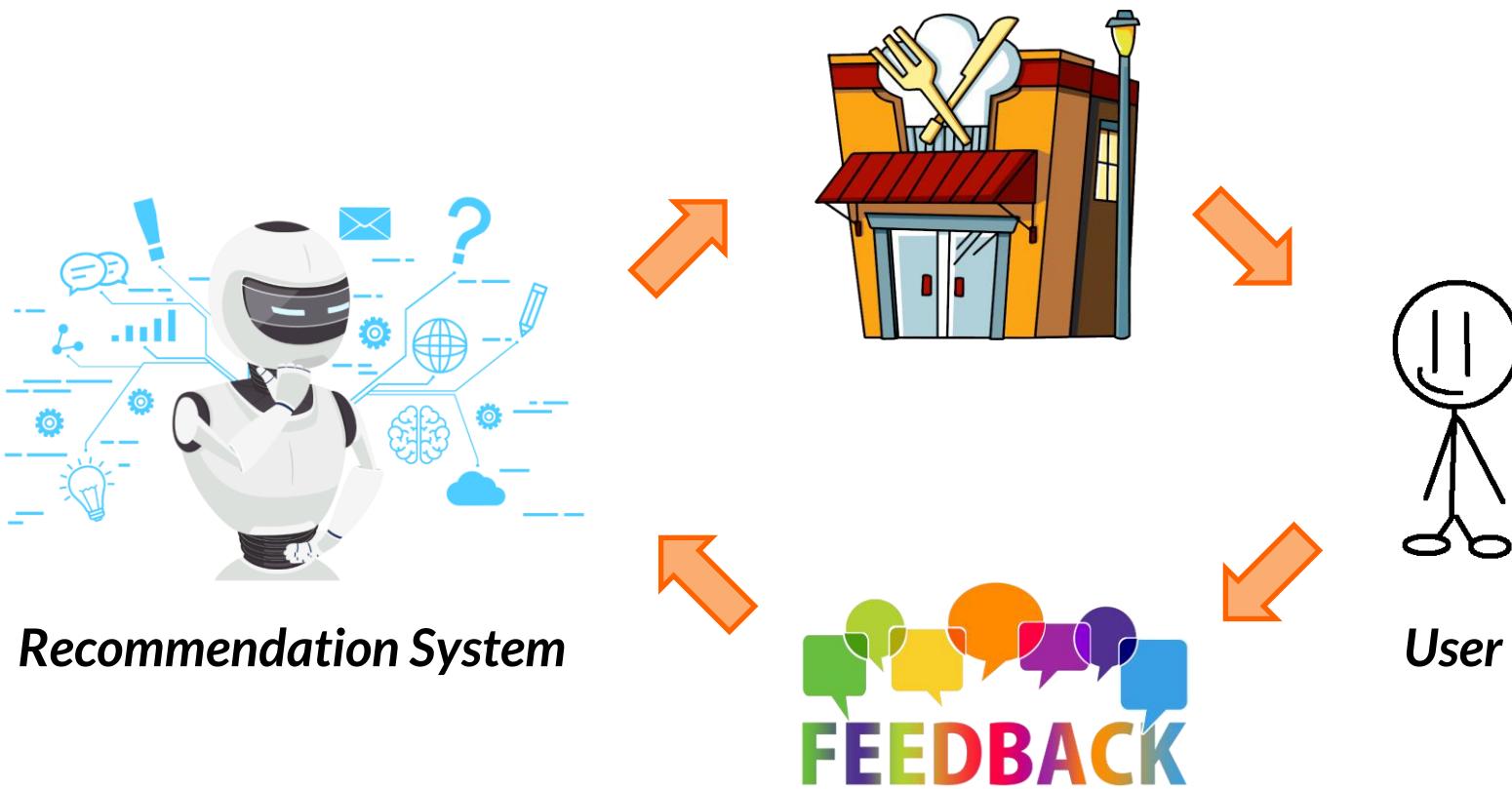
- Adapts to the preferences of each user
- In most cases, provides much better recommendations than non-personalized methods

The screenshot shows the Netflix homepage with a dark theme. At the top, there's a navigation bar with the Netflix logo, a search icon, a gift icon, a bell icon, and a profile icon. Below the navigation, there are three main sections of recommendations:

- Because you watched It's Okay to Not Be Okay**: This section features six movie and TV show thumbnails. From left to right: "START-UP", "RECORD OF YOUTH", "INHERITORS", "STRONGEST DELIVERY MAN", "ROMANCE IS A BONUS BOOK", and "My father is Strange".
- Because you watched The Old Guard**: This section features six movie thumbnails. From left to right: "EXTRACTION", "THE EQUALIZER", "BELOW ZERO", "eye see you", "SALT", and "JOHN WICK 2".
- Because you watched Mystic Pop-up Bar**: This section features six movie and TV show thumbnails. From left to right: "Guardian: The Lonely and Great God", "THE KING ETERNAL MONARCH", "Run On", "HOSPITAL PLAYLIST", "Hello, Me!", and "HI BYE, MAMA!".

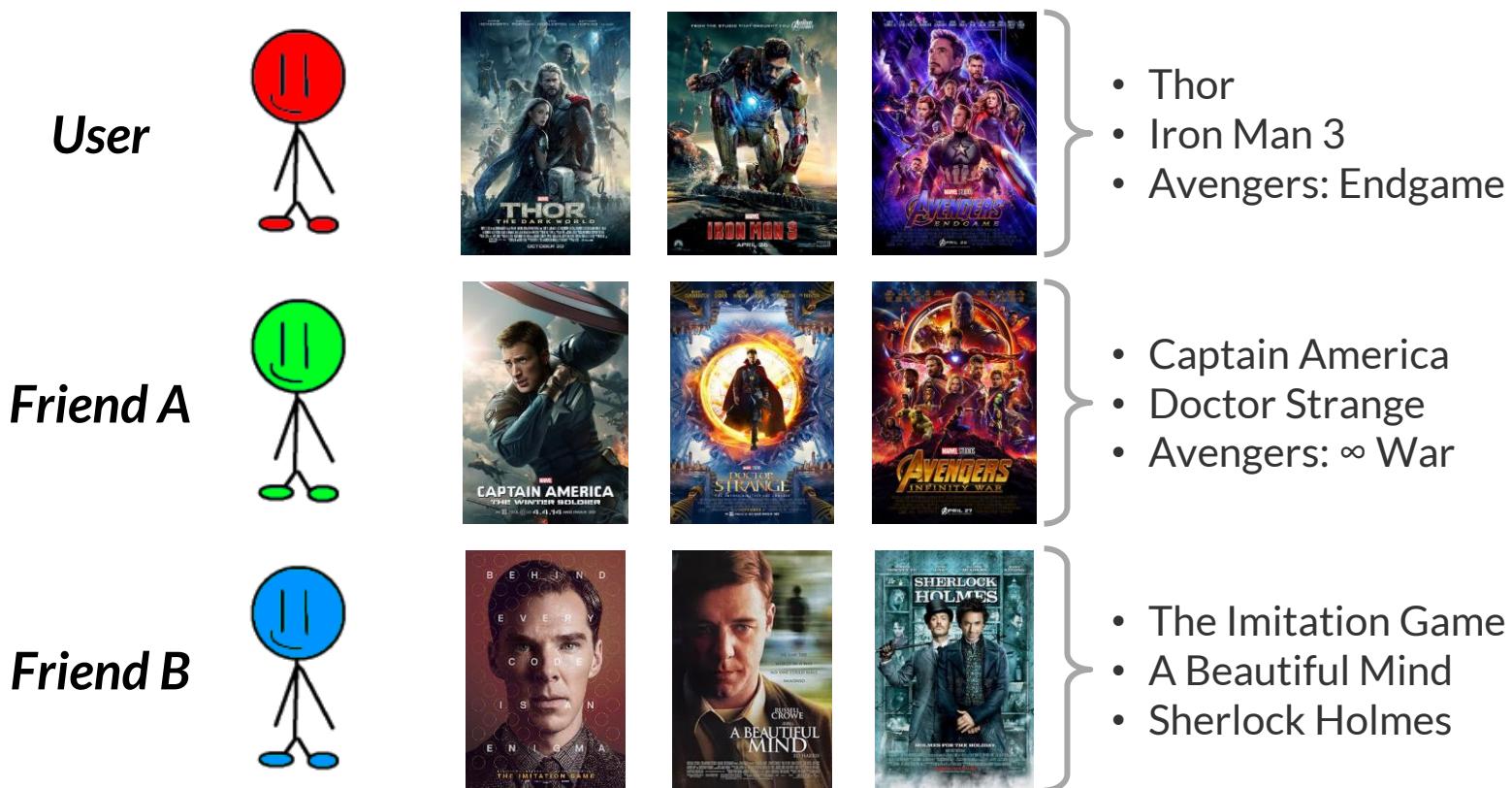
Personalized Recommendations

- ▷ Collaborative Filtering (Concept)
- ▷ Memory-based Approach (Methodology)
- ▷ Model-based Approach (Methodology)



What is Collaborative Filtering?

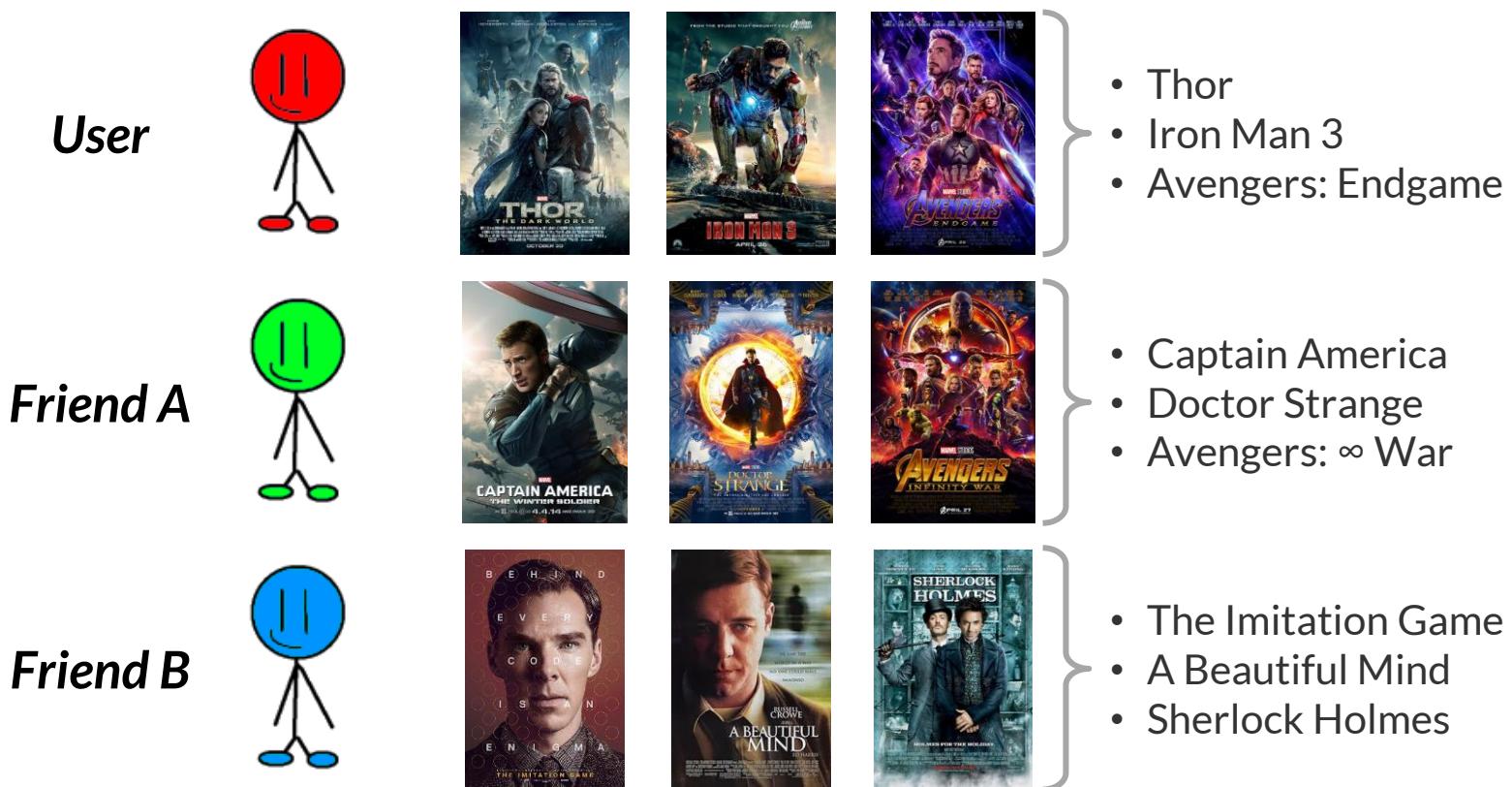
- ▷ Key Idea: People often get the best recommendations from someone with *similar tastes* as themselves!



- Friend A has similar preferences as the User
- Intuitively, Friend A's recommendations should be better than Friend B

What is Collaborative Filtering?

- ▷ Key Idea: People often get the best recommendations from someone with *similar tastes* as themselves!



- The **filtering** (recommending a subset of items) is performed with the help of other similar users (i.e., in a **collaborative** manner)

What is Collaborative Filtering?

- ▷ Key Idea: People often get the best recommendations from someone with *similar tastes* as themselves!
- ▷ Relies (mostly) on the historical interactions
 - Of the *target user*
 - As well as those of other similar users!
- ▷ Not something new
 - Has been proposed a long time ago
 - However, it's still a very powerful tool
 - Most (if not all) recommendation systems are still based on this concept!

Memory-based Approaches

▷ Memory-based Approach (*Methodology*)

- Based on similarities between users (or items)
- Leverages upon the *historical interactions* of the most similar users (or items) to perform recommendations



Memory-based Approaches

- ▷ Memory-based Approach (*Methodology*)
 - Based on similarities between users (or items)
 - Leverages upon the *historical interactions* of the most similar users (or items) to perform recommendations
- ▷ Similarity Measures
 - Jaccard Similarity
 - Cosine Similarity
 - Pearson Correlation
 - ...
- ▷ Examples
 - User-based k-Nearest Neighbour (**UserKNN**)
 - Item-based k-Nearest Neighbour (**ItemKNN**)

Memory-based Approaches

▷ Benefits

- Intuitive & Explainable
- *“We are recommending Movie X to you because other similar users (who have also watched Movies Y and Z) liked Movie X as well”*

▷ Drawbacks

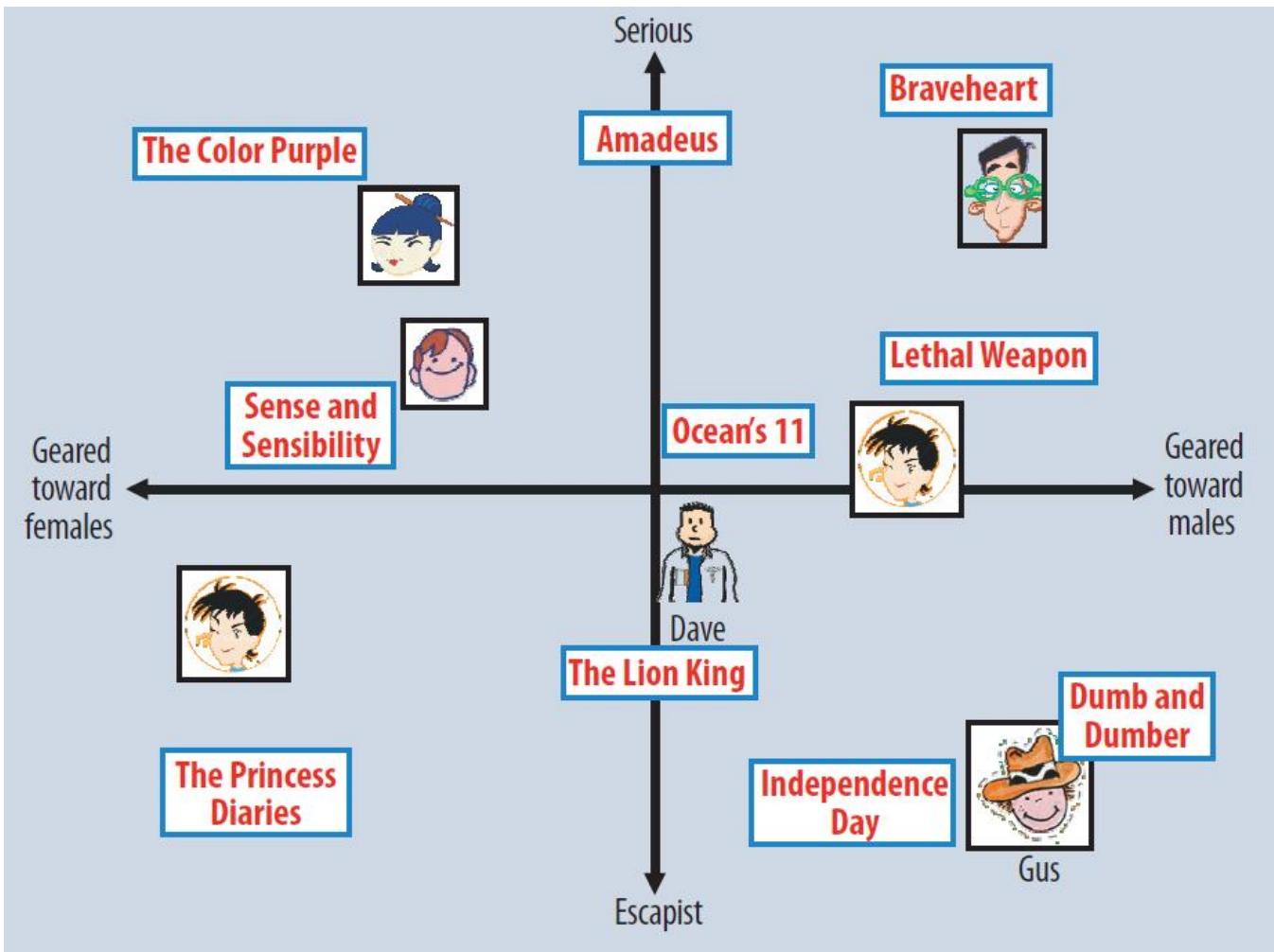
- Cannot handle sparse data well
 - E.g., If most users only interact with small number of items, it would be challenging (or maybe impossible) to find similar users
- Scalability
 - Less computational efficient when the number of users and items grow

Model-based Approaches

- ▷ Model-based Approach (Methodology)
 - Models are built using different data mining (and more recently, machine / deep learning) techniques
- ▷ Examples
 - Bayesian Networks
 - Clustering-based Models
 - *Latent Factor Models (LFMs)* 
 - Matrix Factorization (MF)
 - ...
 - ...



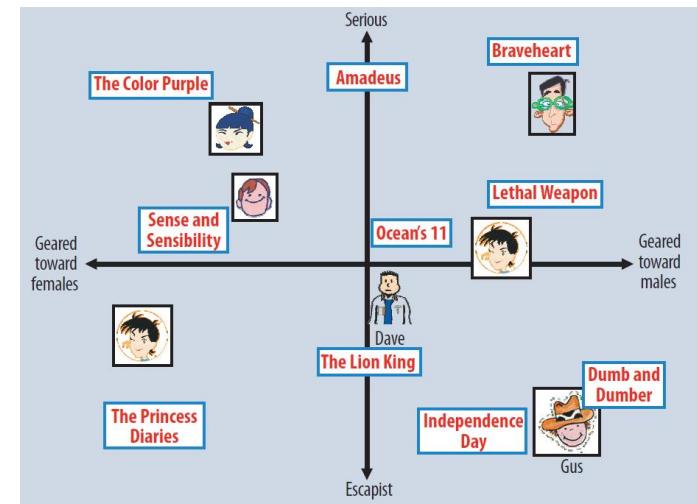
Latent Factor Models



Users and items embedded in a 2-dimensional latent space (From [1])

Latent Factor Models

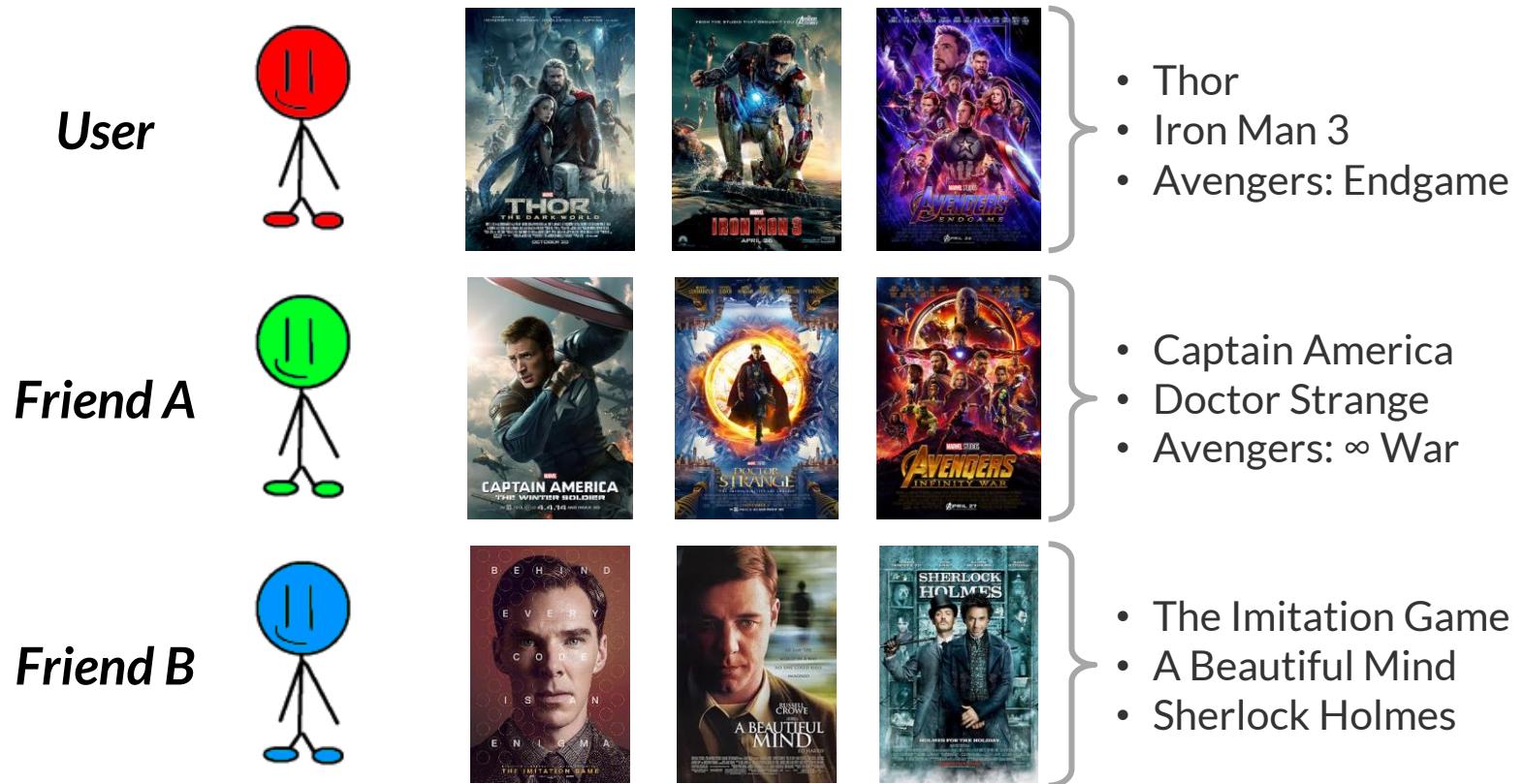
- ▷ Shared Latent Space
 - Each user is represented by a vector (of latent factors)
 - Each item is represented by a vector (of latent factors)
- ▷ Latent Factors
 - Learned automatically from historical interactions
 - These could be *movie genres, item attributes, ...*
 - But they are latent, so we do not really know their true meaning



* Latent: hidden or concealed

Latent Factor Models

- ▷ Similar users are embedded close to each other
- ▷ Similar items are embedded close to each other



$$\bullet \quad Distance(\text{User}, \text{Friend A}) < Distance(\text{User}, \text{Friend B})$$

Latent Factor Models

- ▷ How to compute the user-item rating (or likelihood) using latent factors?
 - **Simple method:** Dot product

$$a \cdot b = \sum_{i=1}^n a_i b_i$$

a = 1st vector

b = 2nd vector

n = dimension of the vector space

a_i = component of vector a

b_i = component of vector b

- **Alternatives:** Learning the user-item interaction function using *neural networks* (e.g. [1]), ...

Model-based Approaches

▷ Benefits

- Works well even with sparse data
- Efficient and scalable
- *# of latent factors <<< # of users & # of items*
 - *User-item matrix is compressed into a low-dimensional latent space*
 - Dot product between two vectors is very fast

▷ Drawbacks

- Interpretability
 - Black box
 - What is the true meaning of the latent factors? ☹

Hybrid Models

- ▷ E.g., Combination of memory-based & model-based
 - Overcomes the limitations of individual approaches
 - Could end up with increased complexity
- ▷ E.g., Training multiple models and combining their predictions



5.

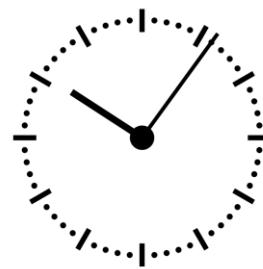
Different types of recommendation systems

Different Types?

- ▷ Due to the different item types
 - E.g., Movies, News, Points of Interests (**POIs**), ...



- ▷ Due to the available information
 - **Content:** Text, Images, User demographics, Item attributes, ...
 - **Context:** Time, Location, ...



POI Recommendation Systems

- ▷ In contrast to movies, books, etc., POIs are ‘physical items’
- ▷ When users interact with POIs, they are limited by *physical constraints*
 - Distance
 - Recommendations cannot be too far away
 - Time
 - Travel time required to reach recommended location
- ▷ As a result, bad recommendations could be somewhat more ‘costly’



POI Recommendation Systems

▷ Context-sensitive

- Distance
 - Current location (e.g., Home vs Workplace)
- Time
 - Weekday vs Weekend
 - Time of the day (Morning/Afternoon/Night)

▷ E.g., Workplace + Friday Evening → Restaurant / Bar



▷ E.g., Home + Saturday Evening → Cinema



▷ E.g., Home + Sunday Morning → Park



POI Recommendation Systems

▷ Locality

- People's activities in urban environments are usually concentrated in certain geographical regions
- E.g., near our home or workplace

▷ Challenges for POI recommendation

- Very sparse data [1]
 - Density for **Netflix (Movies)**: 1.2%
 - Density for **Yelp & Foursquare (POIs)**: 0.1%
- Incorporating the rich context information adequately



Group Recommendation Systems

- ▷ Recommending items to a group of users
 - A group of co-workers
 - A group of friends
 - A couple
 - A family
 - ...
- ▷ Users in a group tend to have...
 - Different preferences
 - Different levels of influence (Leaders vs Followers)
 - ...
- ▷ To be covered later... ☺



6.

*Challenges encountered by
recommendation systems*

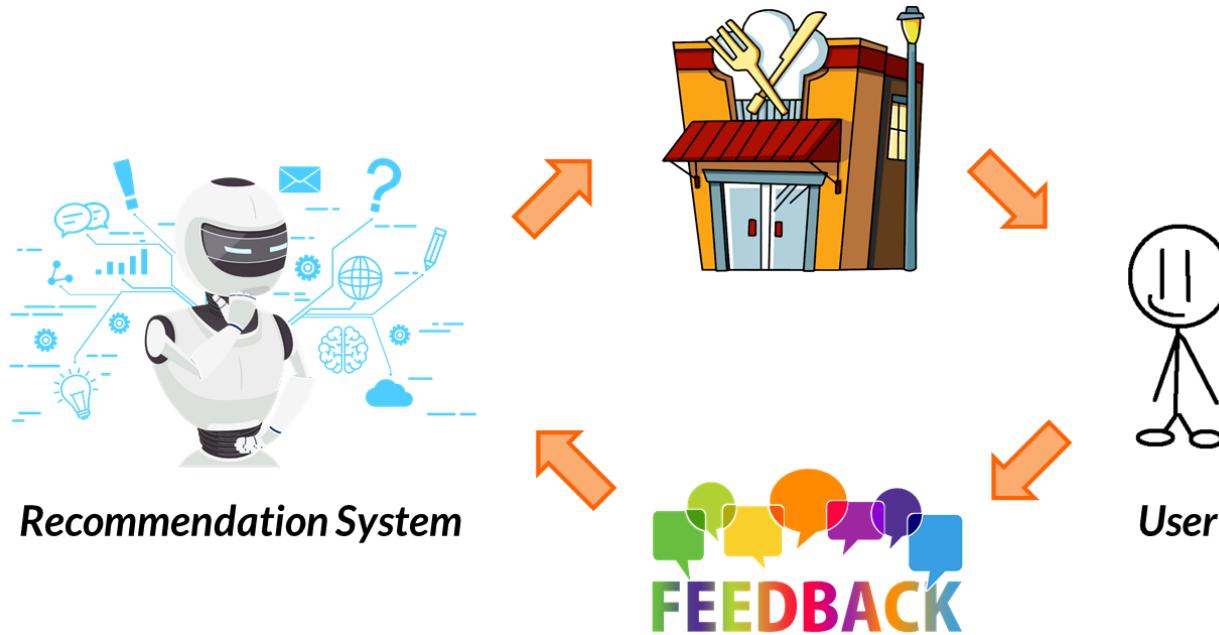
Challenges

- ▷ Cold-start Problem
- ▷ Popularity Bias
- ▷ Shilling Attacks
- ▷ ...



Cold-start Problem

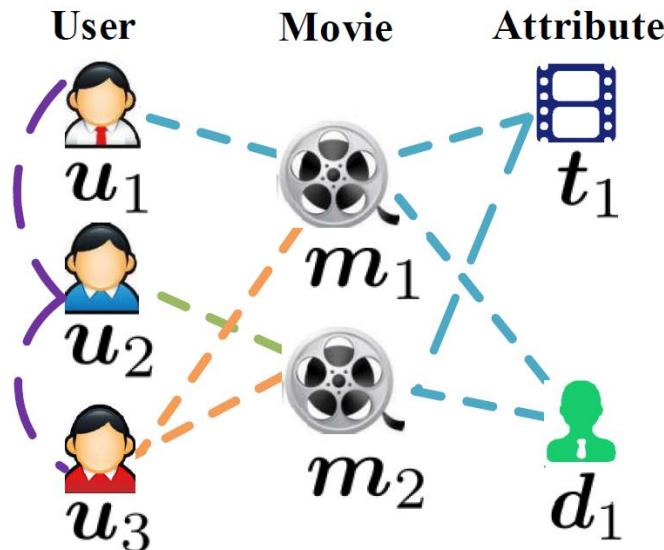
- ▷ Recommendation systems are driven by user interactions



- How about *new users or items* w/ no historical interactions?
- How about users or items w/ little historical interactions?
 - Difficult to *calculate similarity* and/or *derive a good representation*

Cold-start Problem

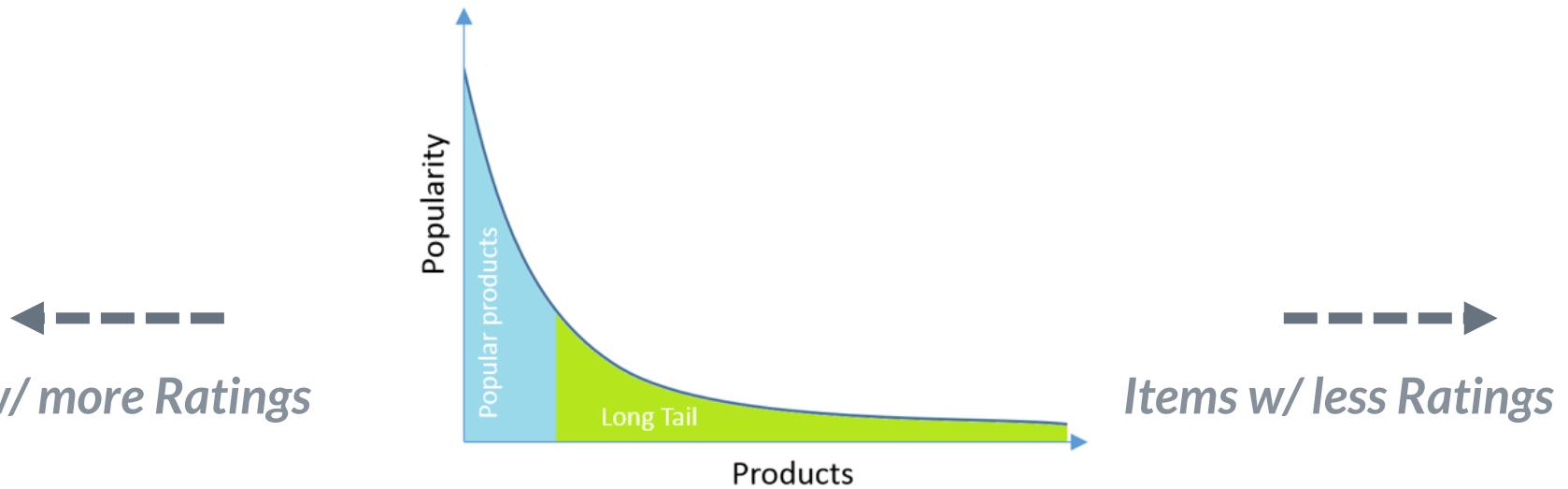
- ▷ In some cases, we could utilize the content information
 - E.g., **User demographics** (Age, Gender, Occupation, ...)
 - E.g., **Item attributes** (Genre, Year, Actors, ...)



- ▷ Possible solutions
 - Content-based Filtering
 - Hybrid Model (Collaborative Filtering + Content)

Popularity Bias

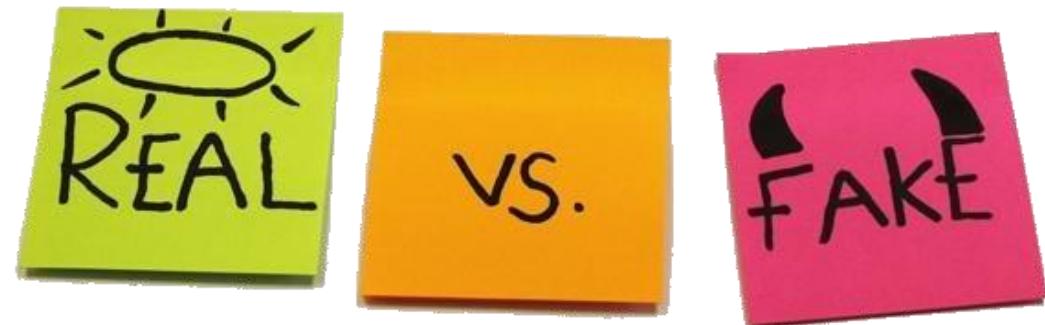
- ▷ Most recommendation systems suffer from *popularity bias*
 - Popular items tend to get recommended over and over again
 - Less popular (or niche) products are less likely to be discovered



Shilling Attacks

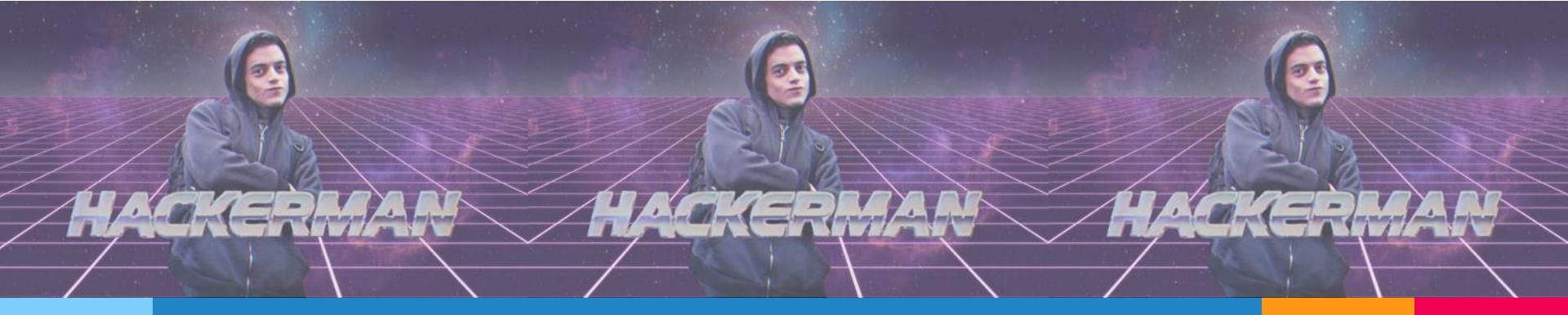
▷ Collaborative Filtering

- “Users who shared similar preferences in the past will likely agree in the future as well”
- Based on “*wisdom of the crowd*”
 - i.e., what others think of an item or a group of items to compute recommendations
- Shortcoming: Unable to distinguish *genuine user profiles* from *fake ones*



Shilling Attacks

- ▷ Characterised by...
 - Several *fake user profiles*
 - Often by an *adversarial party*
 - Harvest recommendation outcomes towards a *malicious desire*
- ▷ “*Malicious Desire*”
 - *Personal gain, market penetration, causing mischief, etc.*



Robustness Against Shilling Attacks

- ▷ Existing work tends to cover *3 main directions*
 1. Attack Designs
 2. Detection Algorithms
 3. Defence Strategies



Thanks!
Any questions?

Email:
S160005@e.ntu.edu.sg