KAIST

KSE801 – Recommender System and Machine Learning on Graph

Lecture 1: Introduction to Recommender System

Prof. Chanyoung Park

Department of Industrial & Systems Engineering
KAIST
cy.park@kaist.ac.kr

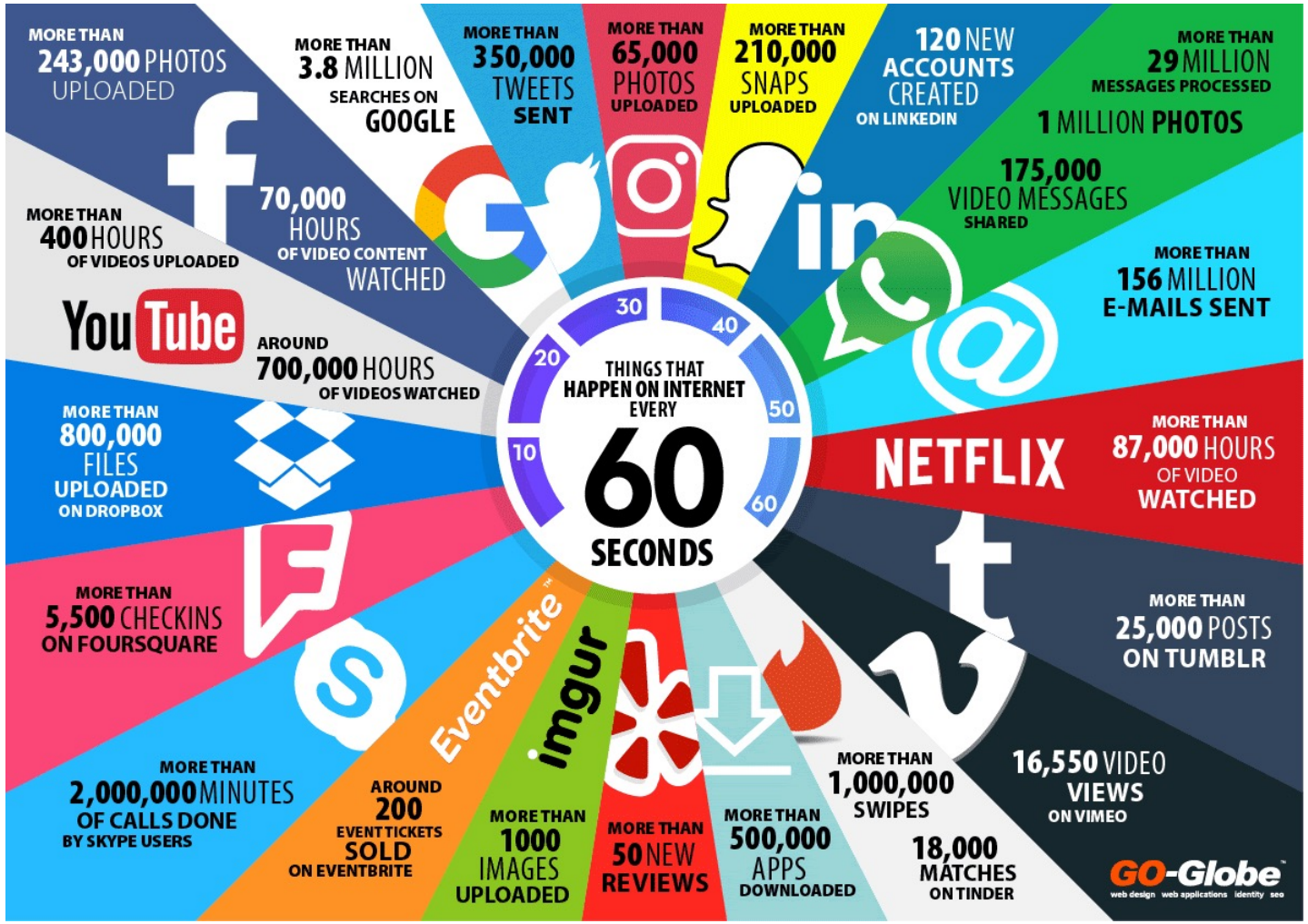
OUTLINE

- Why recommendation?
- Problem Formulation of RecSys
- Business-centric Goal of RecSys
- Basic Models of RecSys
  - Content-based RecSys

OUTLINE

- Why recommendation?
- Problem Formulation of RecSys
- Business-centric Goal of RecSys
- Basic Models of RecSys
  - Content-based RecSys

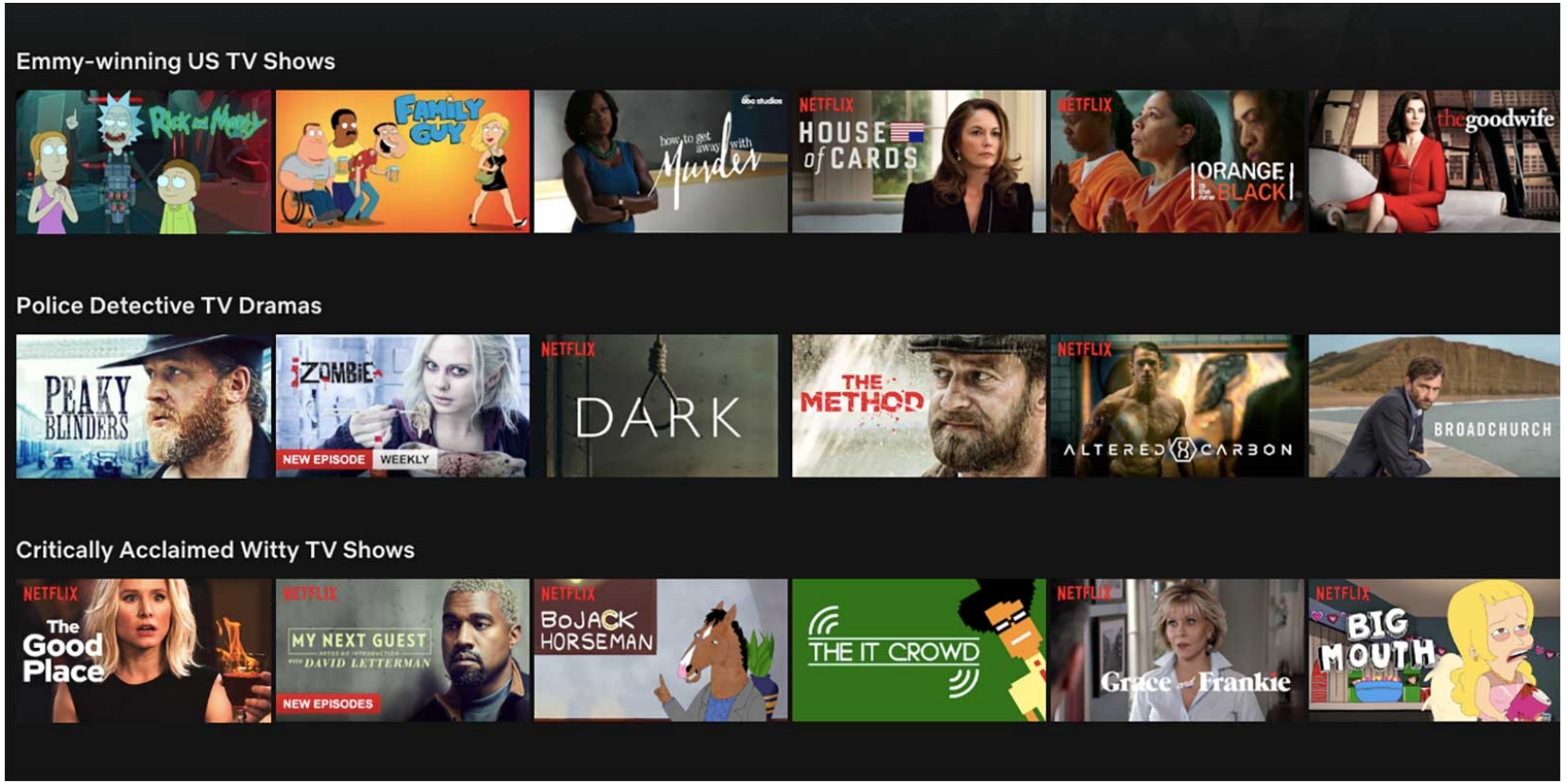
WHY RECOMMENDATION?



[Infographic by GO-Globe Web Design Company]

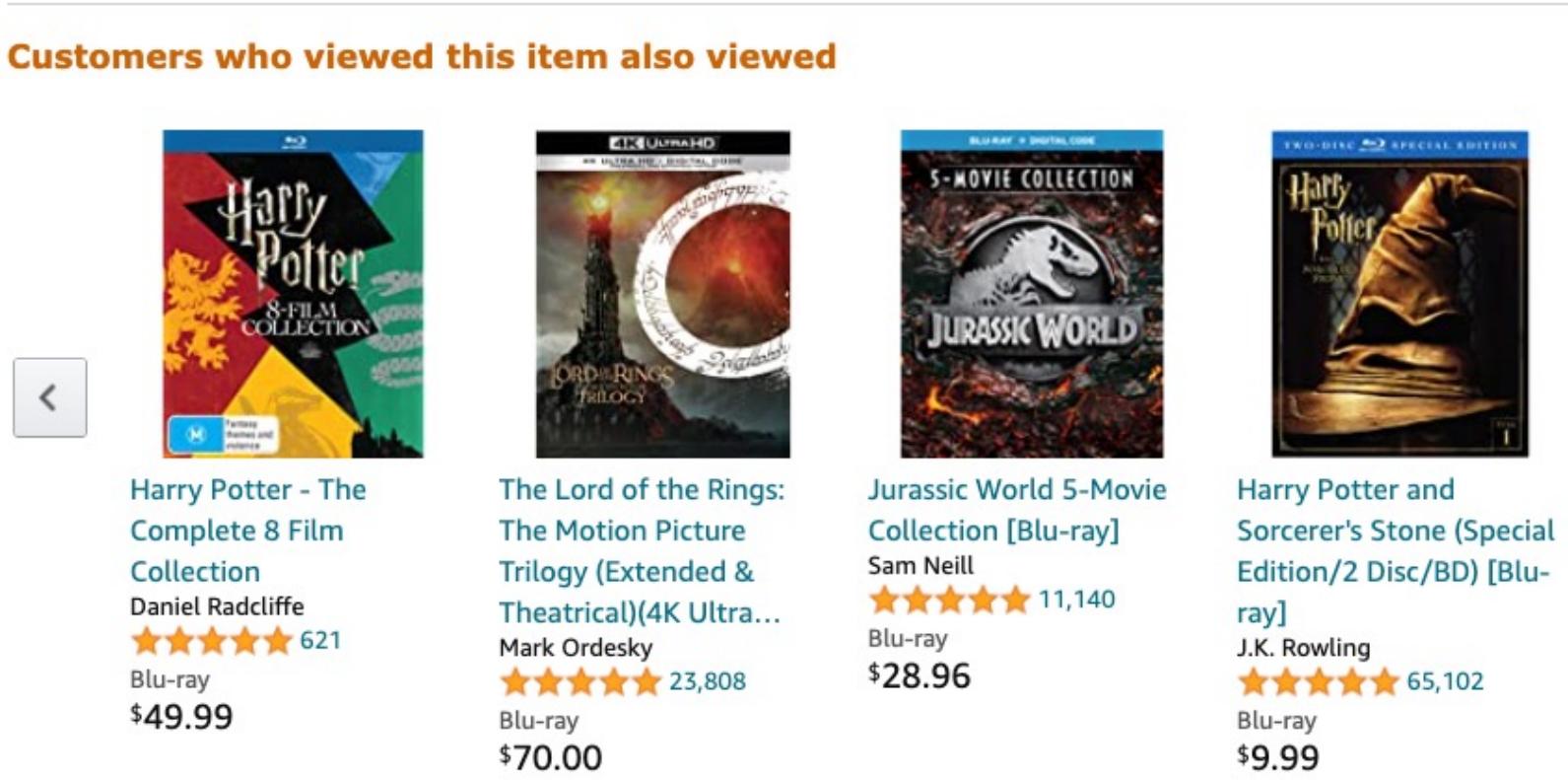
RECOMMENDER SYSTEM: GOAL

- To identify things that **we might like**
  - E.g., recommendation in Netflix



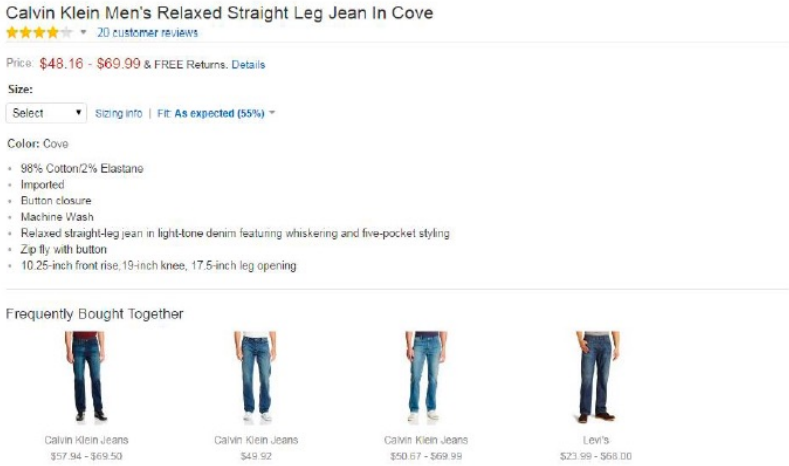
RECOMMENDER SYSTEM: GOAL

- To help people **discover new content**



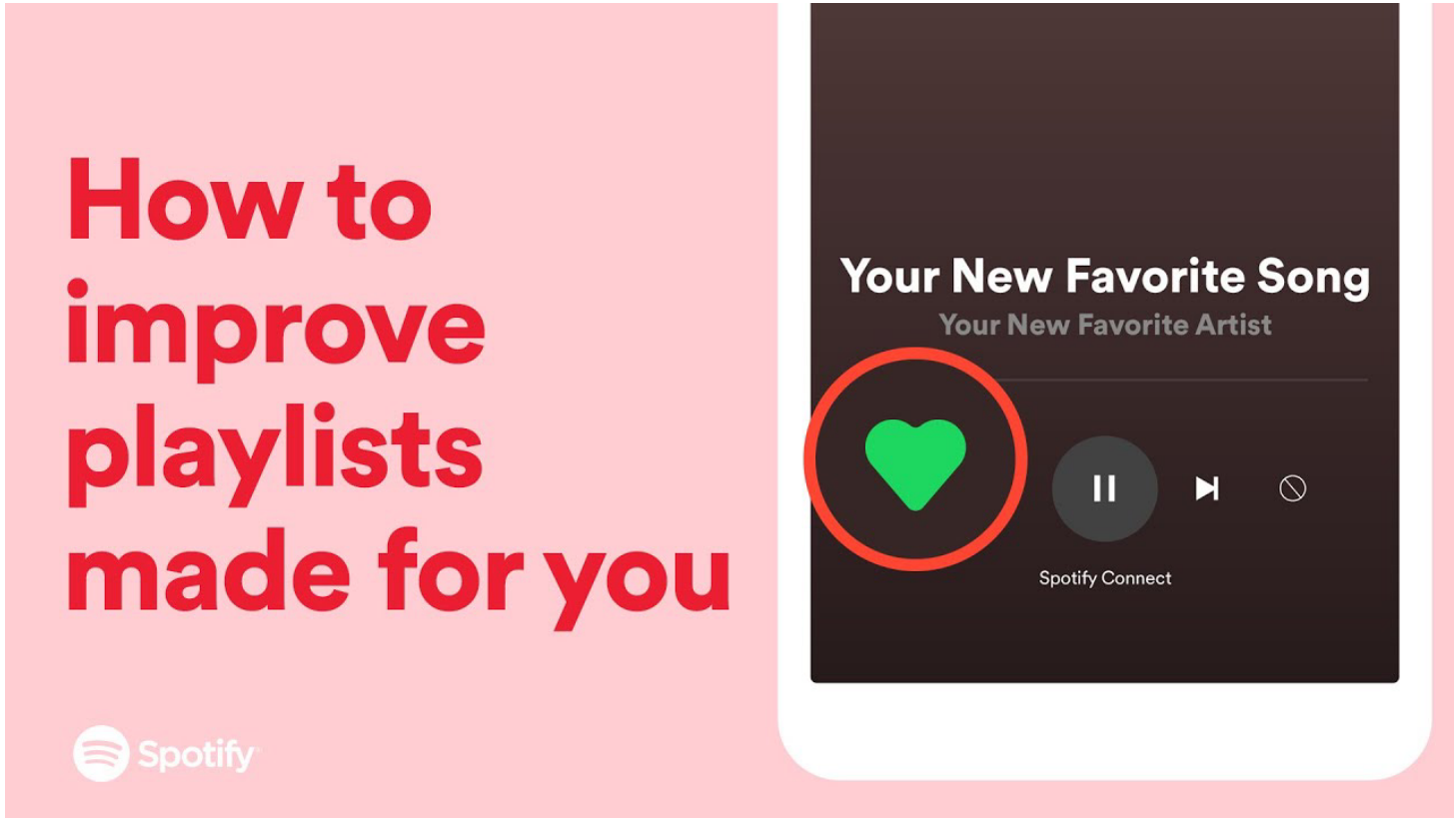
RECOMMENDER SYSTEM: GOAL

- To **discover which things go together**
  - E.g., Fashion recommender system



RECOMMENDER SYSTEM: GOAL

- To **personalize user experiences** in response to user feedback



WHY RECOMMENDATION? SUMMARY

- To identify things that **we might like**
- To help people **discover new content**
- To **discover which things go together**
- To **personalize user experiences** in response to user feedback

To model people's preferences, opinions and behaviors

IMPACT OF RECOMMENDER SYSTEM

*"… **35 percent of what consumers purchase on Amazon** and **75 percent of what they watch on Netflix** come from product recommendations …"*

https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers

How Netflix's AI Recommendation Engine Helps It Save $1 Billion A Year

On Wednesday, Aug 7 2019, by Vishnu Subramanian

Over the last decade, Netflix has slowly grown into the world's most popular subscription-based video streaming service, offering a wide selection of films and TV series including several "Netflix Originals" produced by the company themselves in-house. Netflix has over 150 million subscribers worldwide, a testament to the company's cross-cultural popularity and market dominance in several countries around the world. While this popularity can be attributed to Netflix's pioneering model, an affordable subscription fee and top-notch content/programming, Netflix is also known for using techniques from Artificial Intelligence to maintain its market dominance. Chief among these is the Netflix Recommendation Engine, a tool that is reportedly worth over $1 Billion per year to the company in indirect cost savings.

https://artelliq.com/blog/how-netflix-s-ai-recommendation-engine-helps-it-save-1-billion-a-year/

Deep Learning State of the Art (2020)
Lex Fridman
MIT
Date/Time: Mon, Jan 6, 3-4:30pm
Room: E54-100

https://www.youtube.com/watch?v=0VH1Lim8gL8

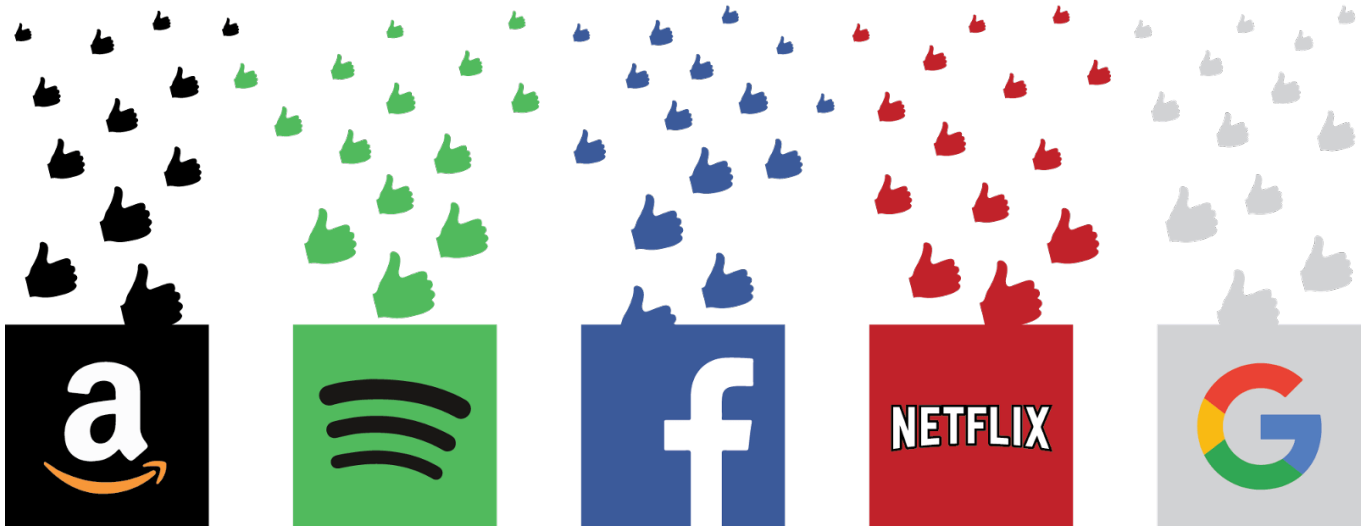
Recommendation System

"Recommendation system is the **most important** in terms of **impact part of AI systems**..."

"...the **most powerful AI space for the next a couple of decades** is recommendation systems. They are going to have **the biggest impact on our society** because they **affect the information we see, how we learn, what we think, how we communicate**. **These algorithms are controlling us**..."

A BRIEF HISTORY

- 1990s – First systems (e.g., GroupLens), basic algorithms
- 2000 ~ 2005 – Research explosion, mainstream applications (Amazon [1])
- **2006 – The Netflix prize**
- 2007 – The first Recommender Systems conference
  - The ACM conference on Recommender Systems (RecSys) https://recsys.acm.org/
- Present – very active research, many applications



(Figure credit): https://thedatascientist.com/right-way-recommender-system-startup/

[1] Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering." *IEEE Internet computing* 7.1 (2003): 76-80.

THE NETFLIX PRIZE

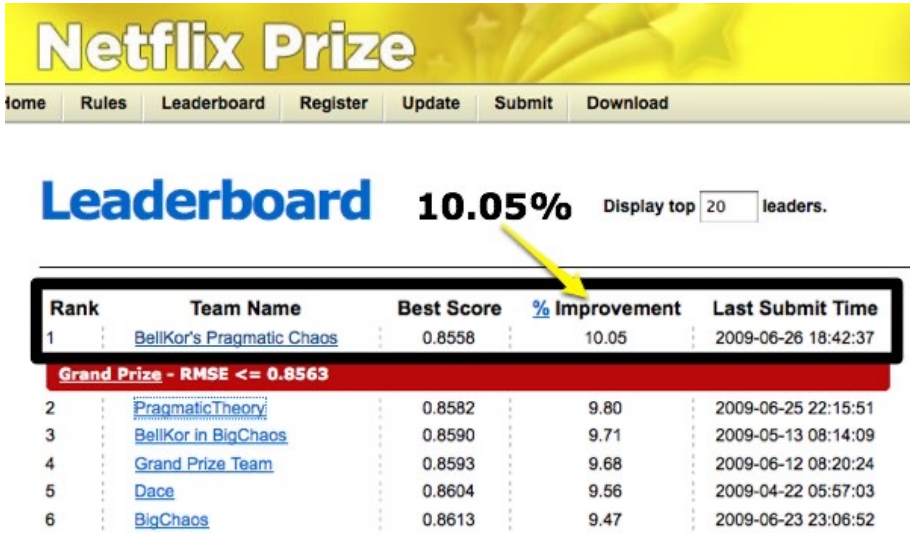
- In 2006, Netflix created a dataset of **100,000,000** movie ratings
- The dataset looked like:

(UserID, ItemID, time, rating)

- The goal was to reduce the RMSE at predicting ratings:

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

- **Prize:** Whoever first manages to reduce the RMSE by **10%** versus Netflix's solution wins **$1,000,000**
  - 2700+ teams
- This led to **a lot** of research on rating prediction by minimizing the RMSE
- Later, we will learn about a few of the main approaches



OUTLINE

- Why recommendation?
- Problem Formulation of RecSys
- Business-centric Goal of RecSys
- Basic Models of RecSys
  - Content-based RecSys

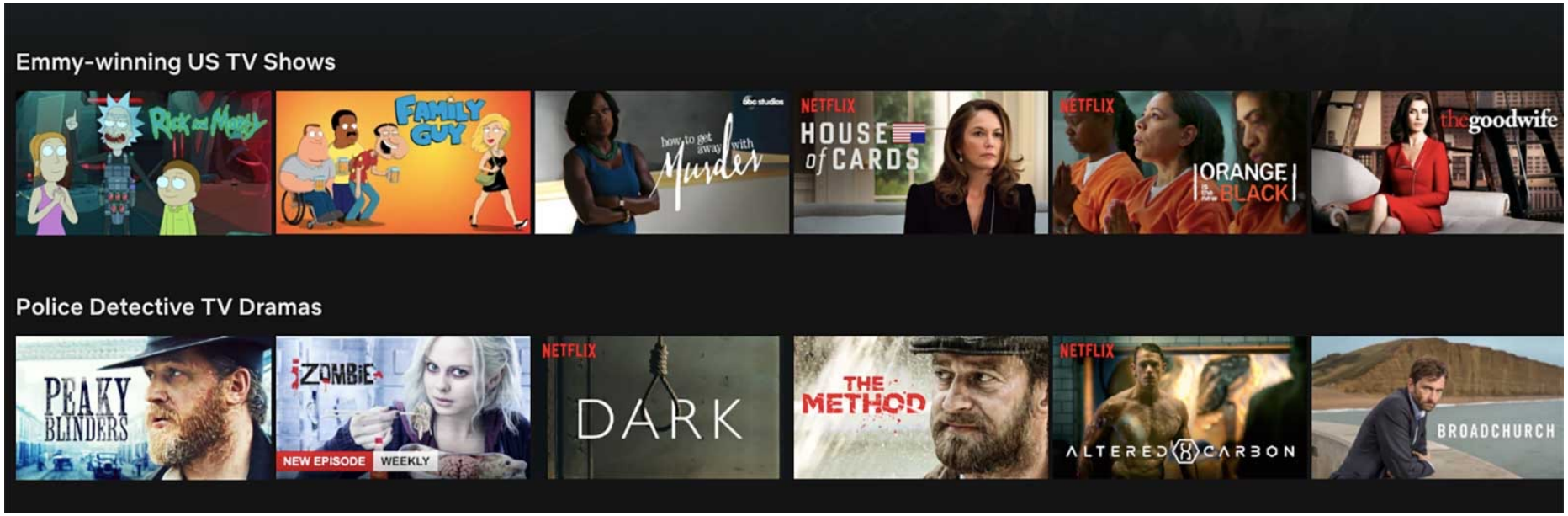
PROBLEM FORMULATION OF RECOMMENDER SYSTEM

- **1) Prediction**
  - *Rating prediction*
  - To predict the missing rating value in an incomplete user-item rating matrix
  - To predict unobserved values given observed values
  - Also known as the **matrix completion problem**

	GLADITOR	GODFTHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			5		2
U_2		5			4	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

PROBLEM FORMULATION OF RECOMMENDER SYSTEM

- **2) Ranking**
  - *Top-k recommendation problem*
  - In practice, we do not need to predict the “ratings” of users in order to make recommendations to users
  - Instead, we need **top-k items** for a particular user
    - or top-k users for a particular item (uncommon)
  - The absolute values of the predicted ratings are not important



TYPES OF RATINGS

- The ratings are often specified on a scale that indicates the specific level of like or dislike of the item
  - **Continuous value**, e.g., [-10,10], etc.
  - **Interval-based**, e.g., {-2,-1,0,1,2}, {1,2,3,4,5}, etc.
  - **Binary**, e.g., Like or Dislike
  - **Unary**

Explicit feedback

Implicit feedback

Interval-based

Binary

Unary

TYPES OF RATINGS: EXPLICIT FEEDBACK

- Probably the most precise rating
- Most commonly used
- **Main Issues**
  - Users are not always willing to rate many items
  - Number of available ratings could be too small → sparse rating matrices → poor recommendation quality

- **How to stimulate users to rate more items?**

TYPES OF RATINGS: IMPLICIT FEEDBACK

- Typically collected by the web shop or application in which the recommender system is embedded
- When a customer buys an item, many recommender systems interpret this behavior as a positive rating
  - e.g., Click through rate, buying an item, visiting a page, viewing a video, dwell time
- Implicit feedback can be collected constantly and do not require additional efforts from user

- **Main Issues**
  - Easy to collect, but less precise
  - There is a mechanism for a user to specify a liking for an item but no mechanism to specify a dislike
  - For example, a user might not like all the books he has bought; the user also might have bought a book for someone else

EXAMPLES OF EXPLICIT AND IMPLICIT RATINGS

	GLADITOR	GODFTHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			5		2
U_2		5			4	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

(a) Ordered ratings

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			1		1
U_2		1			1	
U_3	1	1		1		
U_4			1			1
U_5				1	1	
U_6	1		1			

(b) Unary ratings

Observations

- Two matrices have the **same set of observed entries** but provide **very different insights**
  - U_1 vs. U_3
    - (a): Very different
    - (b): Very similar
- Nonzero values in (b) could be arbitrary positive values
  - e.g., number of clicks, number of listens
- How to treat the **missing values**?

MISSING VALUES

- What about substituting the missing values with 0s?
- **Case 1: Implicit feedback**
  - Often **recommended** to treat the missing entries as 0s to train an algorithm
    - If the missing entries are not substituted with 0s, *significant overfitting* is possible
  - This is because there is often not a sufficient level of discrimination between the observed values

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			1		1
U_2		1			1	
U_3	1	1		1		
U_4			1			1
U_5				1	1	
U_6	1		1			

- **Case 2: Explicit feedback**
  - Pre-substitution of missing ratings is **not recommended** in explicit ratings matrices
  - The substitution of missing entries with any value (e.g., 0 or the row/column/data mean) always leads to a *significant amount of bias* in the analysis
  - Such bias also exists in implicit feedback dataset, but not as severe (also reduces overfitting)
    - e.g., “1” doesn’t explicitly mean “like”

	GLADITOR	GODFTHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			5		2
U_2		5			4	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

OUTLINE

- Why recommendation?
- Problem Formulation of RecSys
- Business-centric Goal of RecSys
- Basic Models of RecSys
  - Content-based RecSys

BUSINESS-CENTRIC GOAL OF RECOMMENDER SYSTEM

- **Primary goal:** Increasing product sales
- However, what items should be recommended to increase product sales?

- **Business-centric goals** of recommender system
  - 1. Relevance
  - 2. Novelty
  - 3. Serendipity
  - 4. Diversity

BUSINESS-CENTRIC GOAL: 1) RELEVANCE, 2) NOVELTY

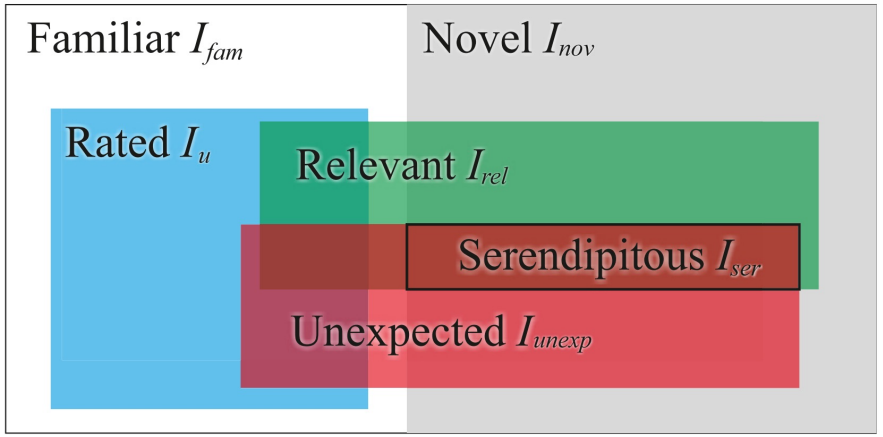
- **1) Relevance**
  - The most obvious goal of a recommender system is to *recommend items that are relevant to the user*
  - Although relevance is the primary goal, it is not sufficient in isolation
    - Users are more likely to consume items they find **interesting**

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}} \qquad MAE = \frac{1}{N}\sum_{i=1}^{N}|Predicted_i - Actual_i|$$

- **2) Novelty**
  - Recommending items that the *user has not seen in the past*
    - e.g., Popular movies of a preferred genre would rarely be novel to the user
  - Repeated recommendation of popular items can also lead to reduction in sales diversity

BUSINESS-CENTRIC GOAL: 3) SERENDIPITY

- Recommending items that are **truly surprising to the user**, rather than simply something they did not know about before (Related to novelty)
  - E.g., Recommending a new Indian restaurant to a user who mostly eats Indian food → Novel but not serendipitous
    - Recommending the same user a Korean restaurant → Serendipitous!
- **Pros**: Long-term and strategic benefits to the merchant
  - Offers the possibility of discovering entirely new areas of interest for a user
- **Cons**: Irrelevant items can be recommended
- The longer term and strategic benefits of serendipitous methods outweigh these short-term disadvantages



Kotkov, Denis, Shuaiqiang Wang, and Jari Veijalainen. "A survey of serendipity in recommender systems." *Knowledge-Based Systems* 111 (2016): 180-192.

BUSINESS-CENTRIC GOAL: 4) DIVERSITY

- Recommender systems typically suggest a list of top-k items
- However, *when all these recommended items are very similar*, it increases the risk that the user might not like any of these items
- On the other hand, when the recommended list contains items of different types, there is a greater chance that the user might like at least one of these items
- Diversity has the benefit of ensuring that the user does not get bored by repeated recommendation of similar items

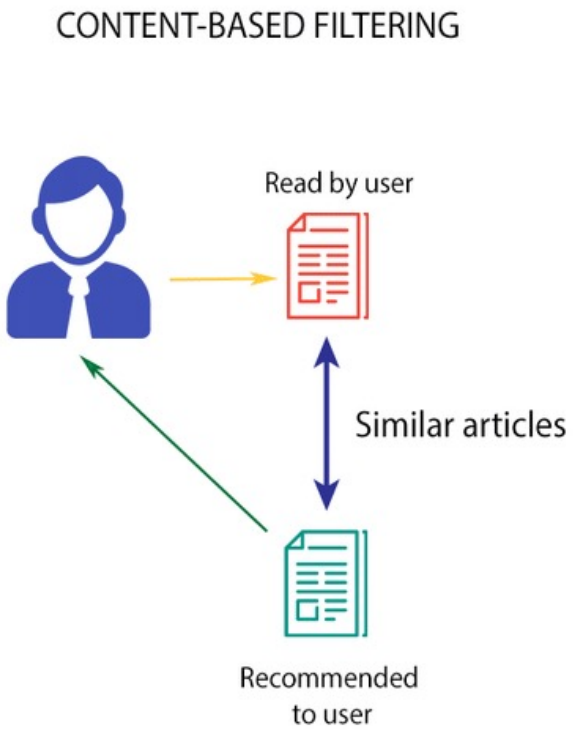
OUTLINE

- Why recommendation?
- Problem Formulation of RecSys
- Business-centric Goal of RecSys
- Basic Models of RecSys
  - Content-based RecSys

OVERVIEW: BASIC MODELS OF RECOMMENDATION

- **Collaborative filtering (CF)** – **Only ratings**
  - Memory-based approach (Neighborhood-based CF) **(Week 2)**
  - Model-based approach **(Week 3 ~ 4)**
- **Side information-based Recommendation (Week 5 ~ 6)**
  - Content-based recommendation – **Only contents (Today)**
  - Content-based CF – **Rating + Contents**
    - Text, image, social network, etc
- **Advanced topics**
  - Sequential Recommendation & Graph-based Recommendation **(Week 7)**

CONTENT-BASED RECOMMENDATION



- **Goal:** Recommend items **similar** to those the user liked
  - Recommendations based on content of items rather than on other users' opinions/interactions
- **Example**
  - **Movie recommendations**
    - Recommend movies with same actor(s), director, genre, …
  - **Websites, blogs, news**
    - Recommend other sites with "similar" content
- **When is it useful?**
  - **Useful when ratings of other users are not available**
  - Example: John has rated the movie "*Terminator*" highly, but we do not have access to the ratings of other users.
    - Therefore, we cannot use CF (No ratings)
    - However, the item description of "*Terminator*" contains similar genre keywords as other science fiction movies, such as "*Matrix*"
    - Therefore, *"Matrix"* can be recommended to John.

How can we find similar items?

https://www.themarketingtechnologist.co/

HOW TO DEFINE SIMILARITY?

- For each item, create an item profile (vector)
- **Profile is vector of features**
  - **Movies**: genre, director, actors, year…

	Actor A	Actor B	…	Johnny Depp	Comic Genre	Spy Genre	Pirate Genre	Avg Rating
Movie X	0	1	1	0	1	1	0	1	3
Movie Y	1	1	0	1	0	1	1	0	4

  - **Text**: Set of "important" words in document

- **How to pick important features?**
  - TF-IDF (Term frequency × Inverse Doc Frequency)

Computing similarity

Cosine similarity between two vectors

Given item profile $\boldsymbol{x}$ and $\boldsymbol{y}$,

$$sim(x, y) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\| \cdot \|\boldsymbol{y}\|}$$

TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)

- **Bag-of-Words:** Each document is represented by a binary vector
  - Each word/term becomes a feature

		Document					
		Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Words	Antony	1	1	0	0	0	1
	Brutus	1	1	0	1	0	0
	Caesar	1	1	0	1	1	1
	Calpurnia	0	1	0	0	0	0
	Cleopatra	1	0	0	0	0	0
	mercy	1	0	1	1	1	1
	worser	1	0	1	1	1	0

  - (-) Order of words/terms is ignored
    - *"John is quicker than Mary"* and *"Mary is quicker than John"* have the same vectors
    - Recently, word embedding methods are used e.g., word2vec
- Which word is more important?
  - We can consider the number of occurrences of a term (i.e., Term frequency)

TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)

- **Term-document count matrix**

		Document					
		Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Words	Antony	157	73	0	0	0	0
	Brutus	4	157	0	1	0	0
	Caesar	232	227	0	2	1	1
	Calpurnia	0	10	0	0	0	0
	Cleopatra	57	0	0	0	0	0
	mercy	2	0	3	5	5	1
	worser	2	0	1	1	1	0

- Does raw term frequency correctly represent the importance of a term?
  - We should also consider the ***document frequency***

TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)

- Document frequency (*df*): the number of documents in the collection that the term occurs in
- **Why df?**
  - Rare terms are more informative than frequent terms (small df → more informative)
- ***Inverse document frequency (idf)***
  - N: Num. documents

$$idf_t = \log \frac{N}{df_t}$$

- idf_t is a measure of the informativeness of the term

- Term frequency−inverse document frequency (TF-IDF)
  - Used to evaluate how important a word is to a corpus of documents.

$$tf - idf_t = (1 + \log tf_{t,d}) \log \frac{N}{df_t}$$

CONTENT-BASED APPROACH: PROS/CONS

- **Pros**
  - No need for data on other users: No cold-start or sparsity
  - Able to recommend to users with unique tastes
  - Able to recommend new and unpopular items
  - Able to provide explanations
    - Can provide explanations of recommended items by listing content-features that caused an item to be recommended

- **Cons**
  - Requires content that can be encoded as meaningful features (difficult in some domains/catalogs)
  - Difficult to implement serendipity (Obvious recommendations)
  - Easy to overfit (e.g. for a user with few data points)
  - Effective for providing recommendations for *new items*, but not for *new users*

- **Pure content-based systems are rarely found in commercial environments**

RATINGS ARE STILL VERY MUCH VALUABLE

Recommending New Movies: Even a Few Ratings Are More Valuable Than Metadata

István Pilászy *
Dept. of Measurement and Information Systems
Budapest University of Technology and Economics
Magyar Tudósok krt. 2.
Budapest, Hungary
pila@mit.bme.hu

Domonkos Tikk *,†
Dept. of Telecom. and Media Informatics
Budapest University of Technology and Economics
Magyar Tudósok krt. 2.
Budapest, Hungary
tikk@tmit.bme.hu

ABSTRACT

The Netflix Prize (NP) competition gave much attention to collaborative filtering (CF) approaches. Matrix factorization (MF) based CF approaches assign low dimensional feature vectors to users and items. We link CF and content-based filtering (CBF) by finding a linear transformation that transforms user or item descriptions so that they are as close as possible to the feature vectors generated by MF for CF.

We propose methods for explicit feedback that are able to handle 140 000 features when feature vectors are very sparse. With movie metadata collected for the NP movies we show that the prediction performance of the methods is comparable to that of CF, and can be used to predict user preferences on new movies.

We also investigate the value of movie metadata compared to movie ratings in regards of predictive power. We compare our solely CBF approach with a simple baseline rating-based predictor. We show that even 10 ratings of a new movie are more valuable than its metadata for predicting user ratings.

1. INTRODUCTION

The goal of recommender systems is to give personalized recommendation on items to users. Typically the recommendation is based on the former and current activity of the users, and metadata about users and items, if available.

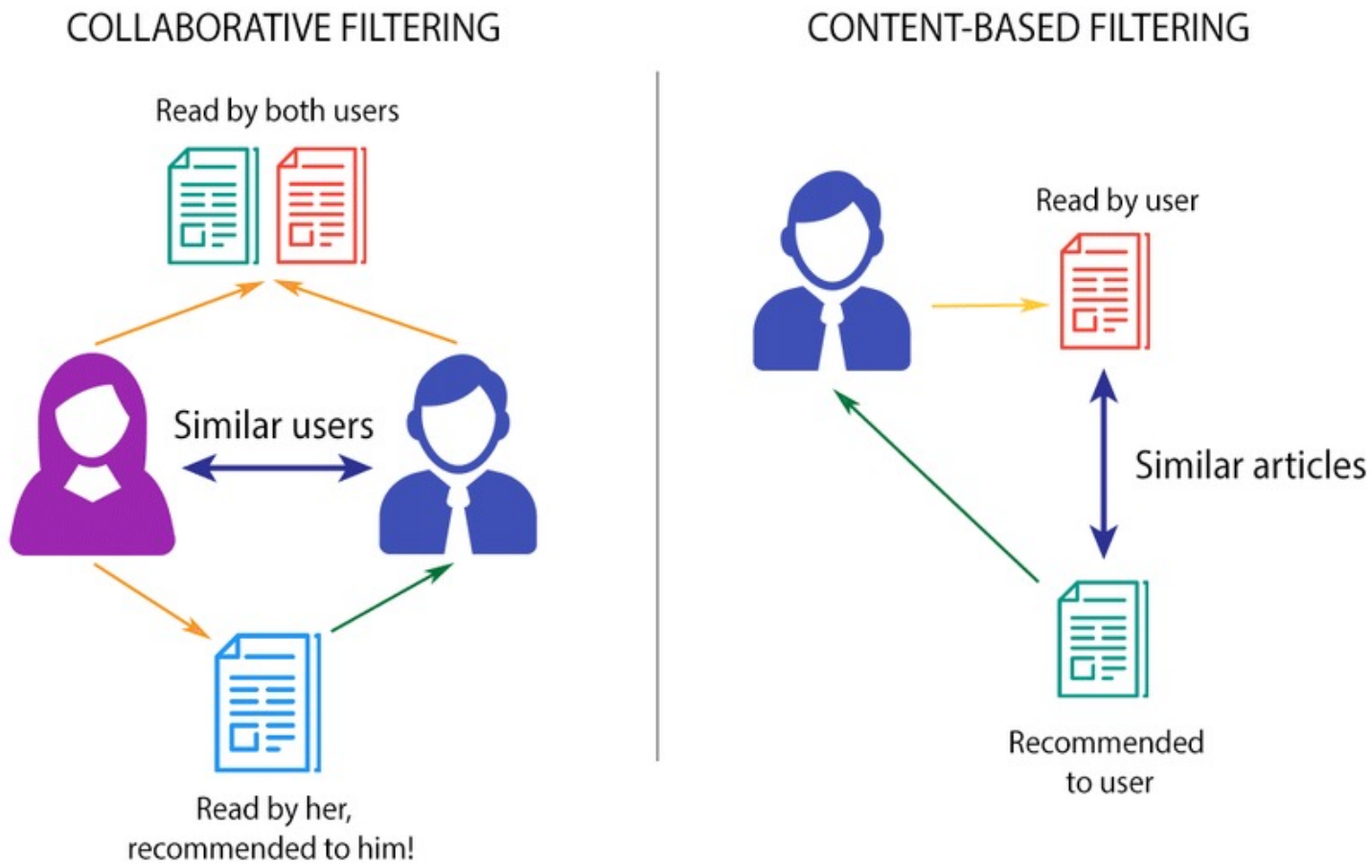
There are two basic strategies that can be applied when generating recommendations. Collaborative filtering (CF) methods are based only on the activity of users, while content-based filtering (CBF) methods use only metadata. In this paper we propose hybrid methods, which try to benefit from both information sources.

The two most important families of CF methods are matrix factorization (MF) and neighbor-based approaches. Usually, the goal of MF is to find a low dimensional representation for both users and movies, i.e. each user and movie is associated with a feature vector. Movie metadata (which are mostly textual) can also be represented as a vector, using the usual vector-space model of text mining.

Our approach to connect CF and CBF methods works via the analog representation form. We aim to find a linear

Pilászy, István, and Domonkos Tikk. "Recommending new movies: even a few ratings are more valuable than metadata." *Proceedings of the third ACM conference on Recommender systems*. 2009.

CONTENT-BASED FILTERING VS. COLLABORATIVE FILTERING



https://www.themarketingtechnologist.co/

CONCLUSION

- Motivation of recommender system
- The Netflix Challenge
- Problem Formulation of recommender system
  - Prediction vs. Ranking
- Types of ratings
  - Explicit feedback vs. Implicit feedback
  - Handling missing values
- Business-centric goal of recommender system
  - 1. Relevance / 2. Novelty / 3. Serendipity / 4. Diversity
- Content-based Recommendation
  - TF-IDF

Coming up next:
Neighborhood-based CF