Advanced Data Science

Applications

Session 8

Kiran Waghmare
Program Manager
C-DAC Mumbai

Agenda

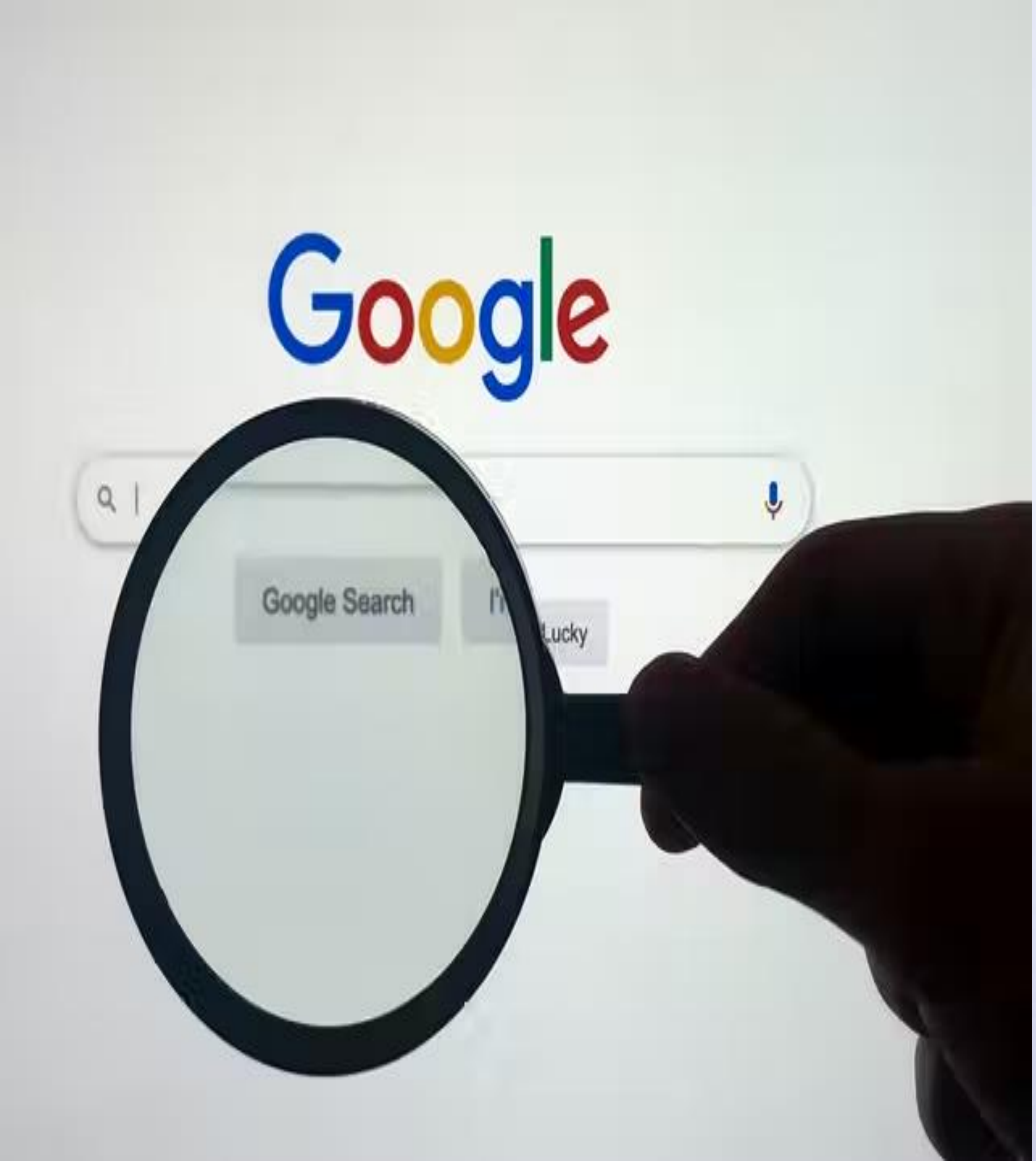
Applications in Machine Learning;

Applications of NumPy in Data Analysis

Case study : Titanic case study

Time Series

Recommended systems;


Google
Google Search


Google
AUSTRALIA NEWS
australia news - Google Search
australia news today
australia news now
australia news latest
australia news covid
australia news code


NETFLIX
Category Codes 2022
NETFLIX
CROWN
how i met your mother
Breaking Bad
SPECTRAL
LE REGOLE DELLA TRUFFA


EECH RECOGNITION


Transit
1:30min 120m
Hotel
40m
Grocery Store
30m
Restaurant
25m


Website
Network
Mobile
Social
Media
Email
Marketing
SEO
Strategy
Forum
Online
Marketing
Content
Marketing
Blog
Marketing


DIGITAL
MARKETING
AD
MOBILE
SEO
E-MAIL
VIDEO
SOCIAL
NETWORK

Case study : Titanic case study

Topics Covered in Today's Training

01 Why Time Series Analysis?

02 What is Time Series?

03 Components of Time Series

04 When not to use Time Series?

05 What is Stationarity?

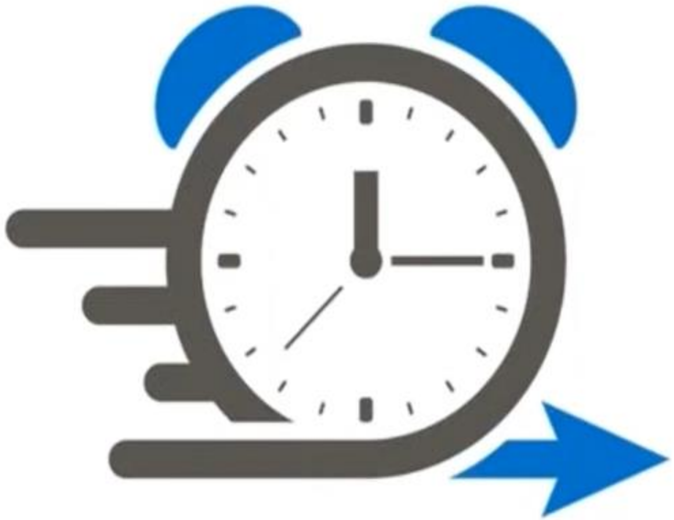
06 ARIMA model

07 Demo: Forecast future

Why Time Series Analysis?

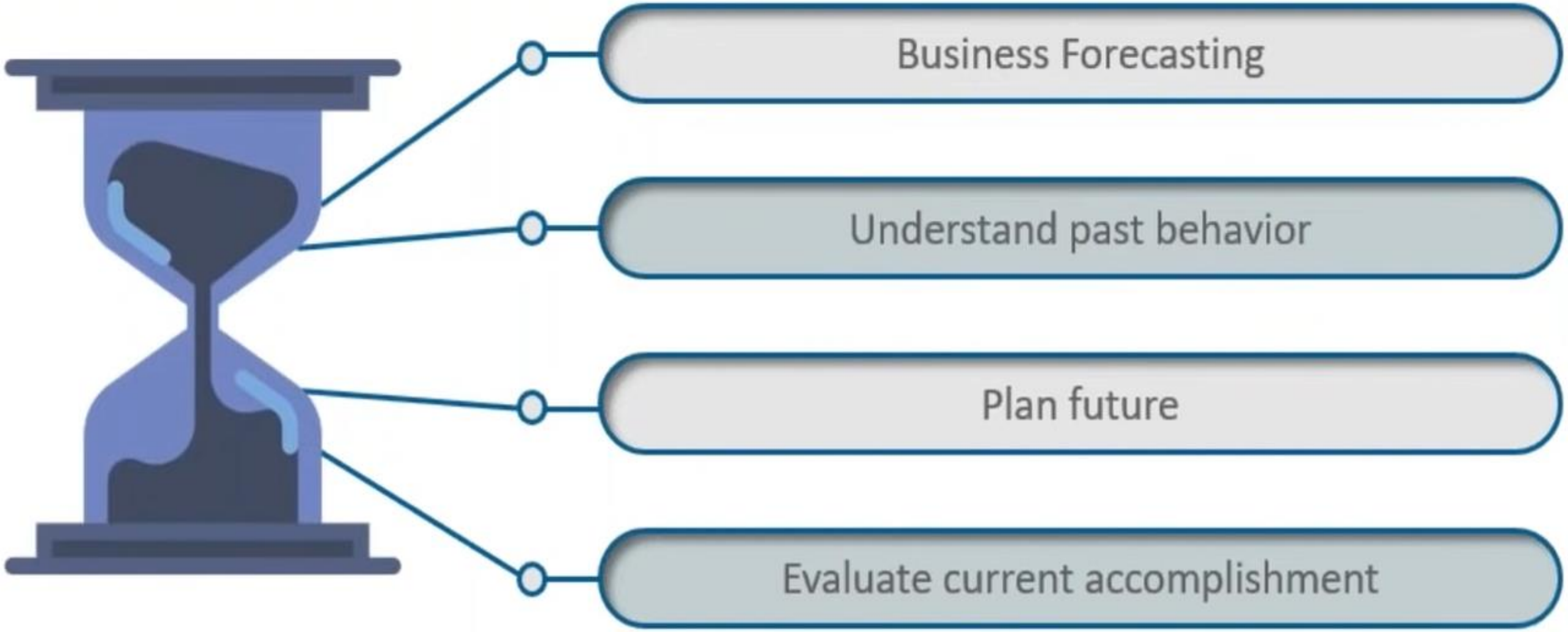
In this analysis, you just have one variable – **TIME**

You can analyse this time series data in order to extract meaningful statistics and other characteristics

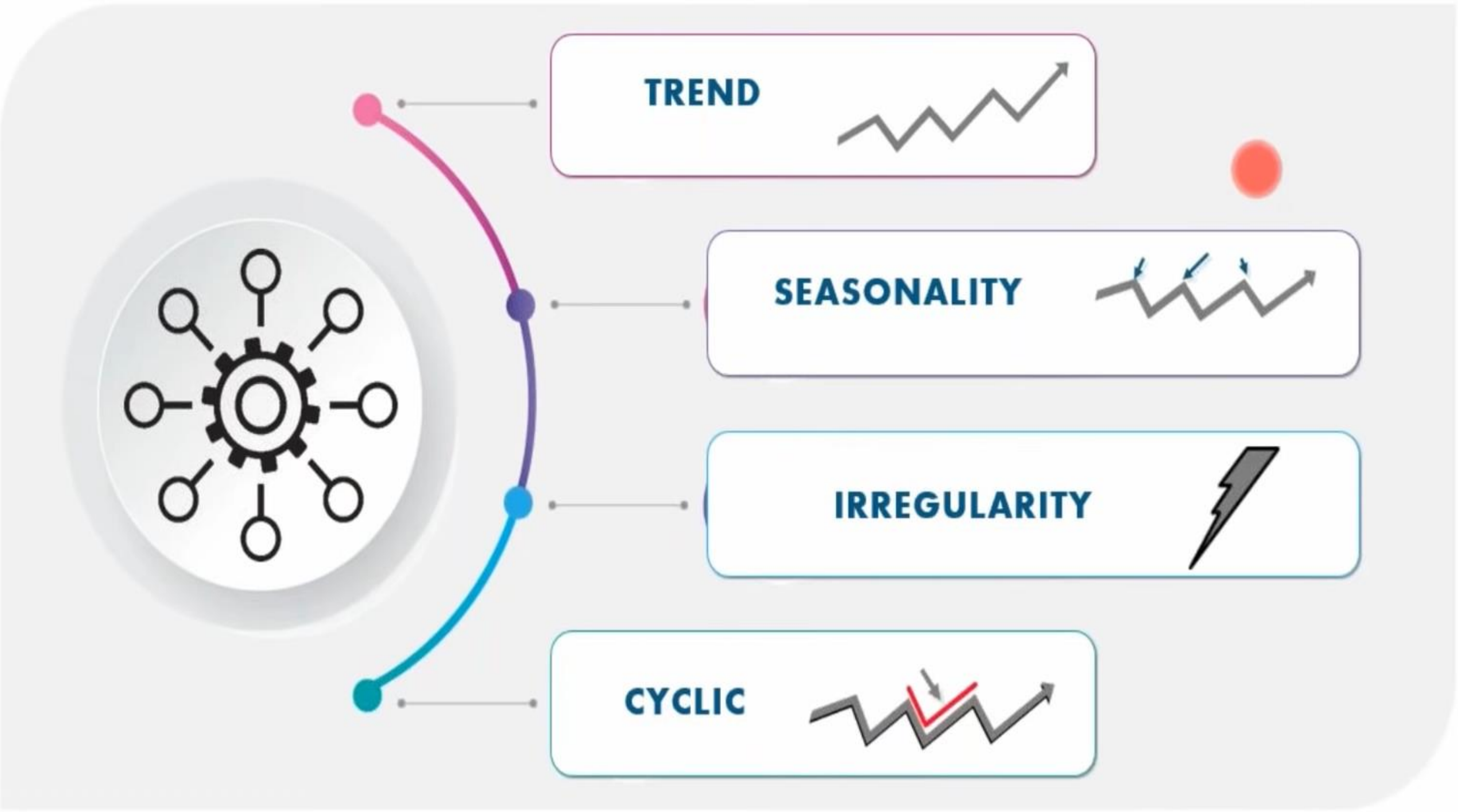


What Is Time Series?

- A time series is a set of observation taken at specified times usually at equal intervals
- It is used to predict the future values based on the previous observed values



Components Of Time Series



Time-Series Components

Time Series

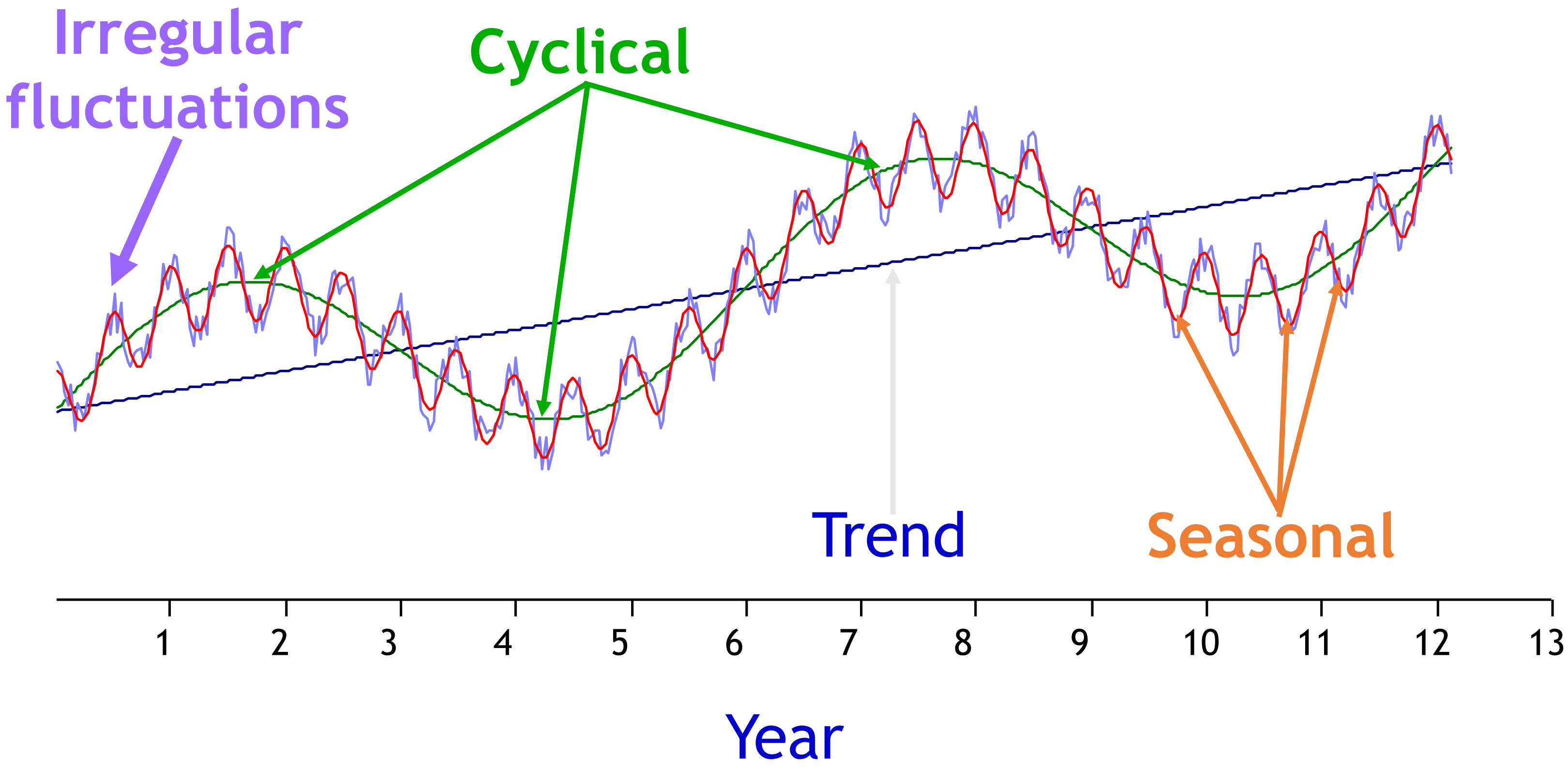
Trend Component	Seasonal Component	Cyclical Component	Irregular /Random Component
Overall, persistent, long-term movement	Regular periodic fluctuations, usually within a 12-month period	Repeating swings or movements over more than one year	Erratic or residual fluctuations

Time series components

Time series data can be broken into these four components:

1. Secular trend
2. Seasonal variation
3. Cyclical variation
4. Irregular variation

Components of Time-Series Data

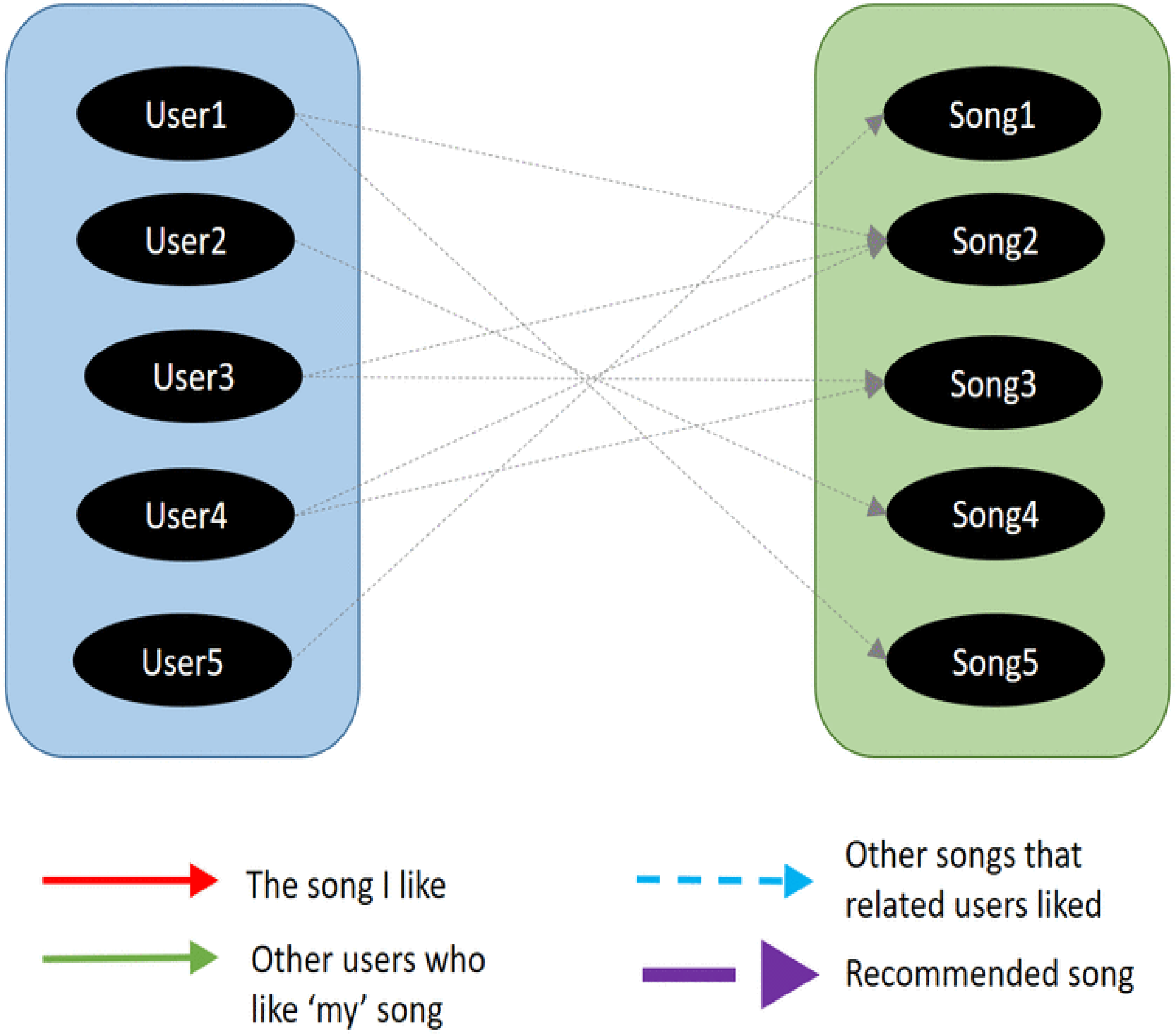


Predicting long term trends without smoothing?
What could go wrong?
Where do you commence your prediction from the bottom of a variation going up or the peak of a variation going down………..

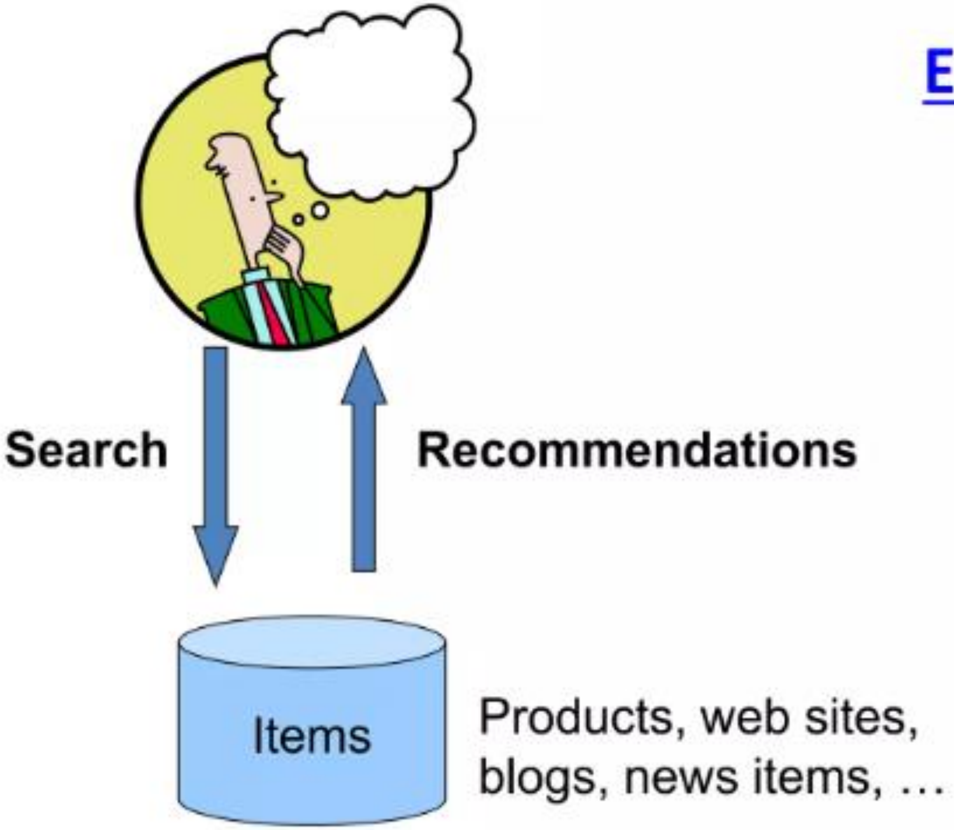
Agenda

- Recommender System
- Content-based Management
- Collaborative Filtering


Spotify R&D
MOSTRA
SAT
Exposure
Discovery
The Winner Takes It All - From "...
Meryl Streep
Amnesia
5 Seconds of Summer
tear myself apart
Tate McRae
Superhero
Lauv
In My Head
Peter Manos
I'll Still Have Me
Cyn
in my head
Ariana Grande
Like Me (feat. BJ The Chicago Kid)
Joey Bada$$
How Do You Sleep?
Sam Smith
Like Me (feat. BJ The Chicago Kid)
Joey Bada$$


User1
User2
User3
User4
User5
Song1
Song2
Song3
Song4
Song5
The song I like
Other users who like 'my' song
Other songs that related users liked
Recommended song

Recommendations



Examples:

amazon.com

PANDORA

StumbleUpon

del.icio.us

NETFLIX

movielens
helping you find the *right* movies

last.fm
the social music revolution

Google News

YouTube

XBOX LIVE

Formal Model

- ***X*** = set of **Customers**
- ***S*** = set of **Items**

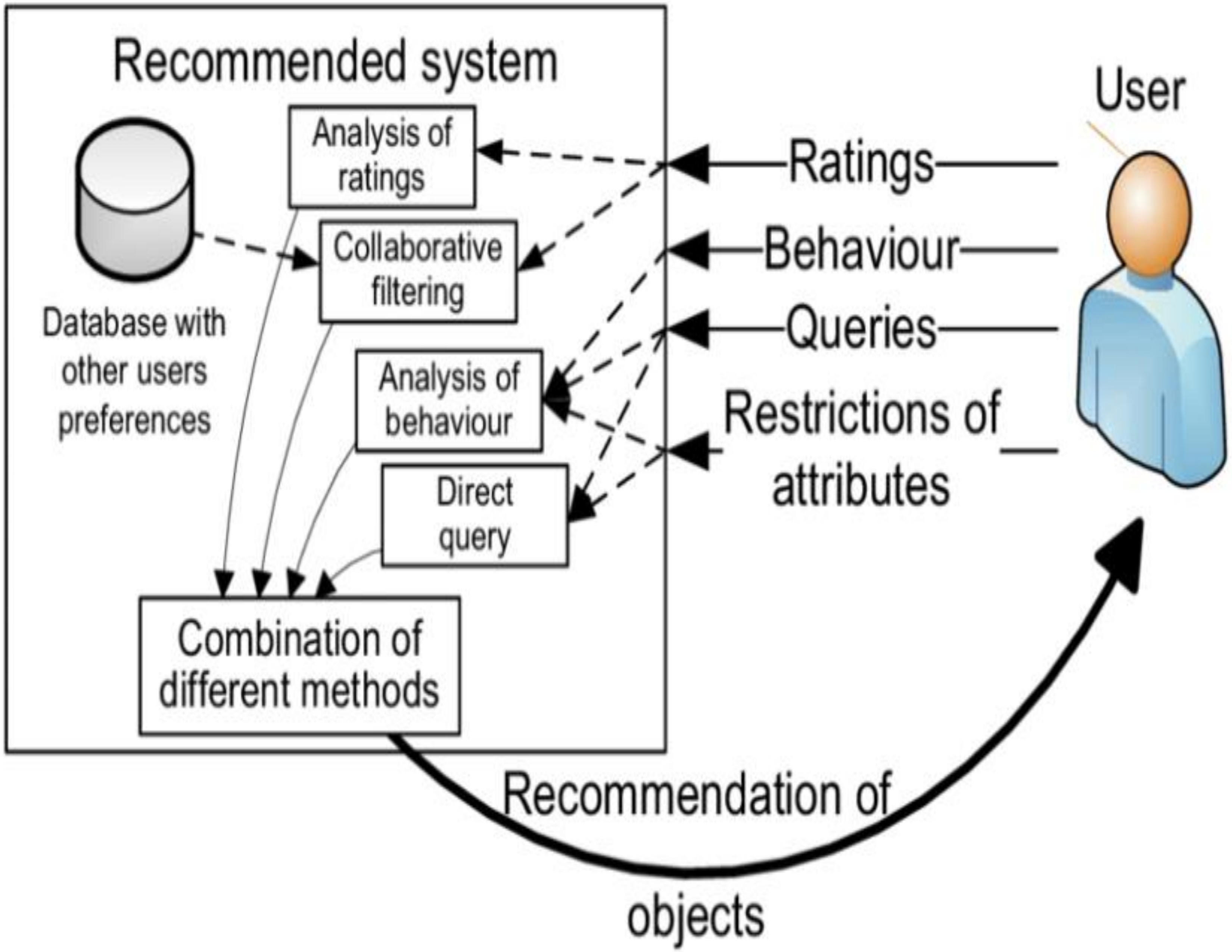
- **Utility function** $u: X \times S \rightarrow R$
  - ***R*** = set of ratings
  - ***R*** is a totally ordered set
  - e.g., **0-5** stars, real number in **[0,1]**

Utility Matrix

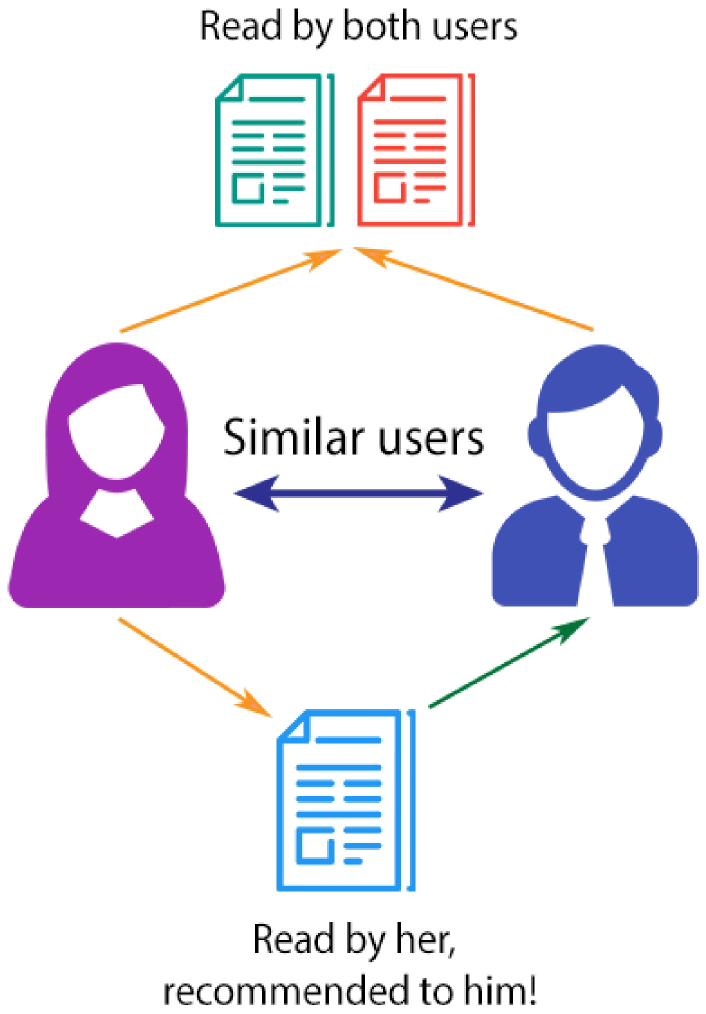
	Avatar	LOTR	Matrix	Pirates
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

(1) Gathering Ratings

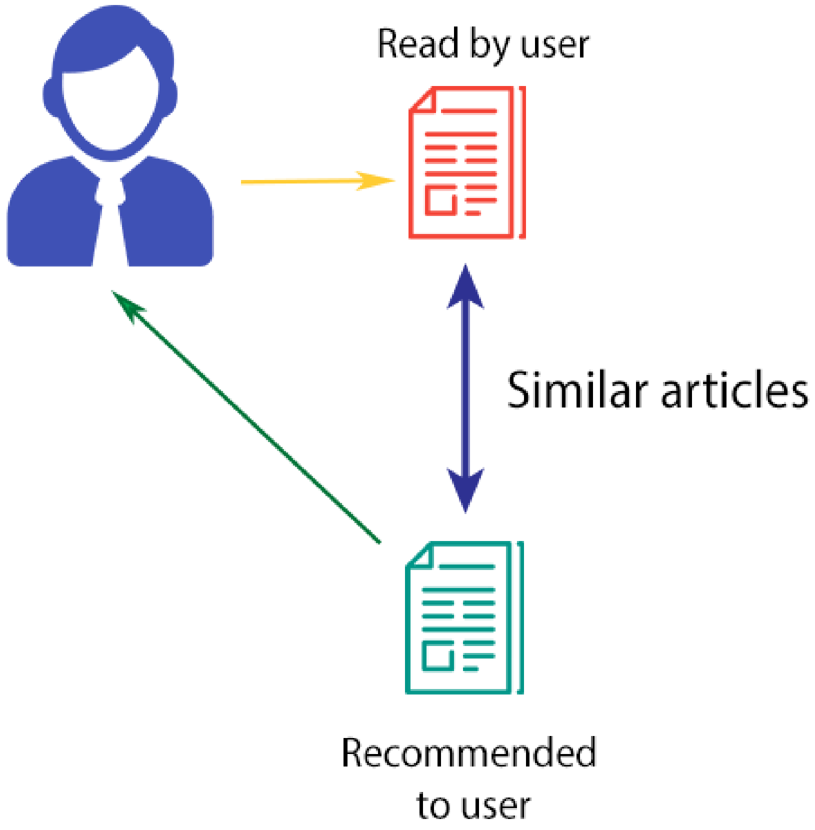
- **Explicit**
  - Ask people to rate items
  - Doesn't work well in practice – people can't be bothered
- **Implicit**
  - Learn ratings from user actions
    - E.g., purchase implies high rating
  - What about low ratings?

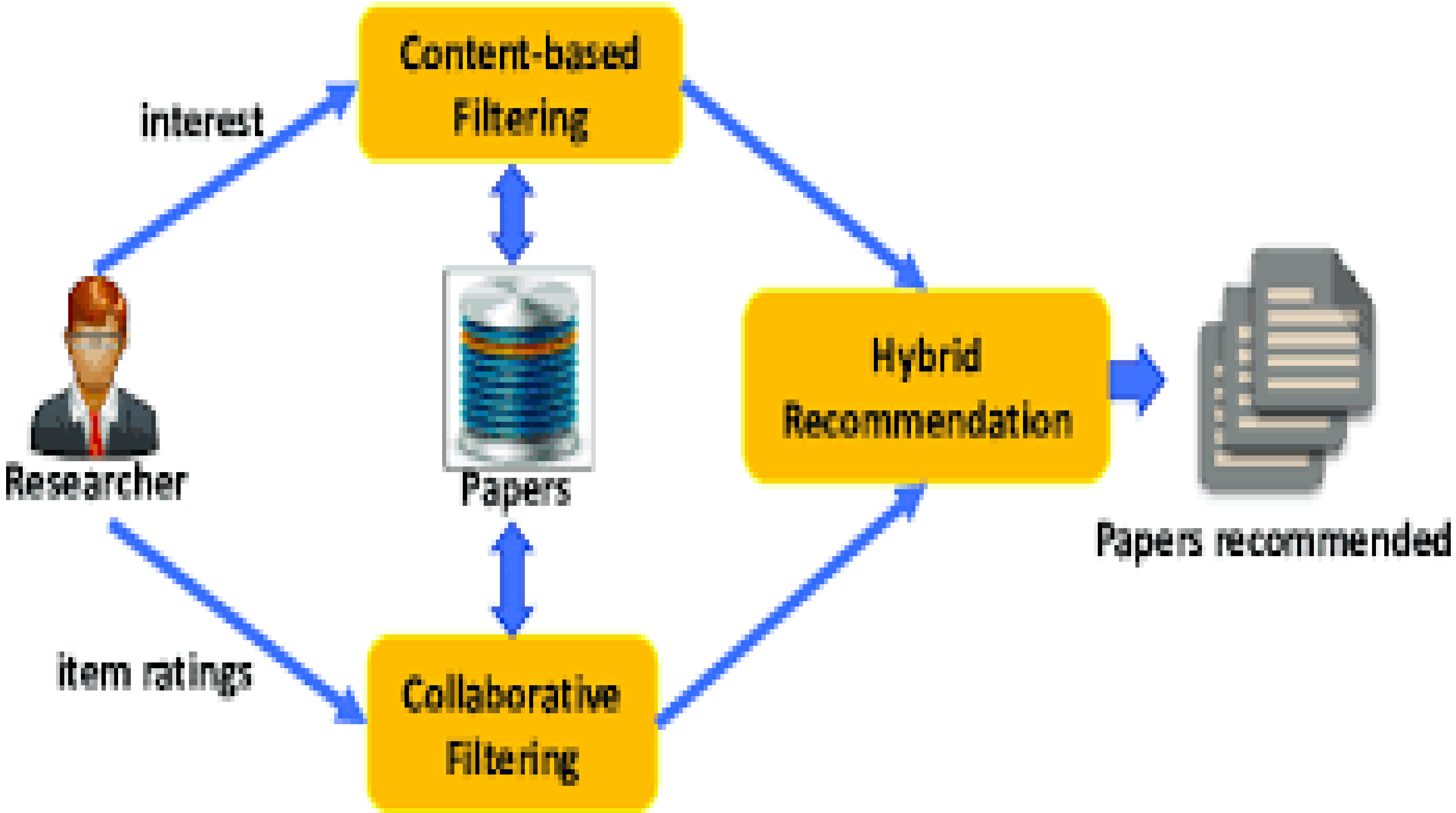

Recommended system
Analysis of ratings
Collaborative filtering
Database with other users preferences
Analysis of behaviour
Direct query
Combination of different methods
User
Ratings
Behaviour
Queries
Restrictions of attributes
Recommendation of objects

COLLABORATIVE FILTERING



CONTENT-BASED FILTERING




Content-based Filtering
interest
Researcher
Papers
Hybrid Recommendation
Papers recommended
item ratings
Collaborative Filtering

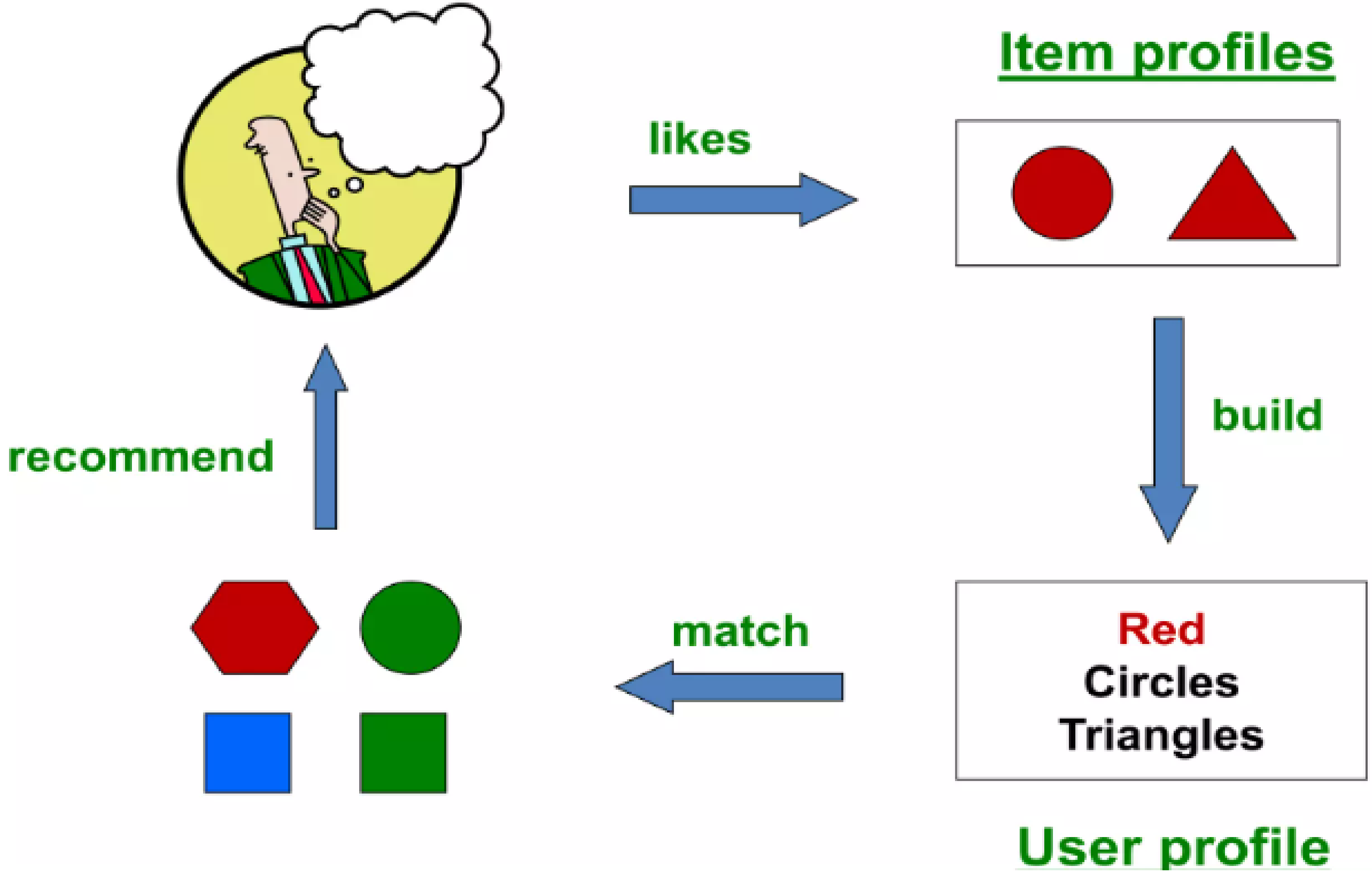
Content-based Recommendations

- **Main idea:** Recommend items to customer ***x*** similar to previous items rated highly by ***x***

Example:

- **Movie recommendations**
  - Recommend movies with same actor(s), director, genre, …
- **Websites, blogs, news**
  - Recommend other sites with "similar" content

Plan of Action



User Profiles and Prediction

- **User profile possibilities:**
  - Weighted average of rated item profiles
  - **Variation:** weight by difference from average rating for item
  - …
- **Prediction heuristic:**
  - Given user profile ***x*** and item profile ***i***, estimate $u(\boldsymbol{x}, \boldsymbol{i}) = \cos(\boldsymbol{x}, \boldsymbol{i}) = \frac{\boldsymbol{x} \cdot \boldsymbol{i}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{i}||}$

Pros: Content-based Approach

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

Cons: Content-based Approach

- **–: Finding the appropriate features is hard**
  - E.g., images, movies, music
- **–: Recommendations for new users**
  - **How to build a user profile?**
- **–: Overspecialization**
  - Never recommends items outside user's content profile
  - People might have multiple interests
  - **Unable to exploit quality judgments of other users**

Collaborative Filtering

Harnessing quality judgments of other users

Collaborative Filtering

- Consider user ***x***
- Find set ***N*** of other users whose ratings are "**similar**" to ***x***'s ratings
- Estimate ***x***'s ratings based on ratings of users in ***N***

Finding "Similar" Users

r_x = [*, _, _, *, ***]
r_y = [*, _, **, **, _]

- Let r_x be the vector of user x's ratings
- **Jaccard similarity measure**
  - **Problem:** Ignores the value of the rating
- **Cosine similarity measure**

$$\text{sim}(\boldsymbol{x}, \boldsymbol{y}) = \arccos(\boldsymbol{r_x}, \boldsymbol{r_y}) = \frac{r_x \cdot r_y}{||r_x|| \cdot ||r_y||}$$

  - **Problem:** Treats missing ratings as "negative"
- **Pearson correlation coefficient**
  - S_{xy} = items rated by both users x and y

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \overline{r_x})(r_{ys} - \overline{r_y})}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \overline{r_x})^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \overline{r_y})^2}}$$

r_x, r_y as sets:
r_x = {1, 4, 5}
r_y = {1, 3, 4}

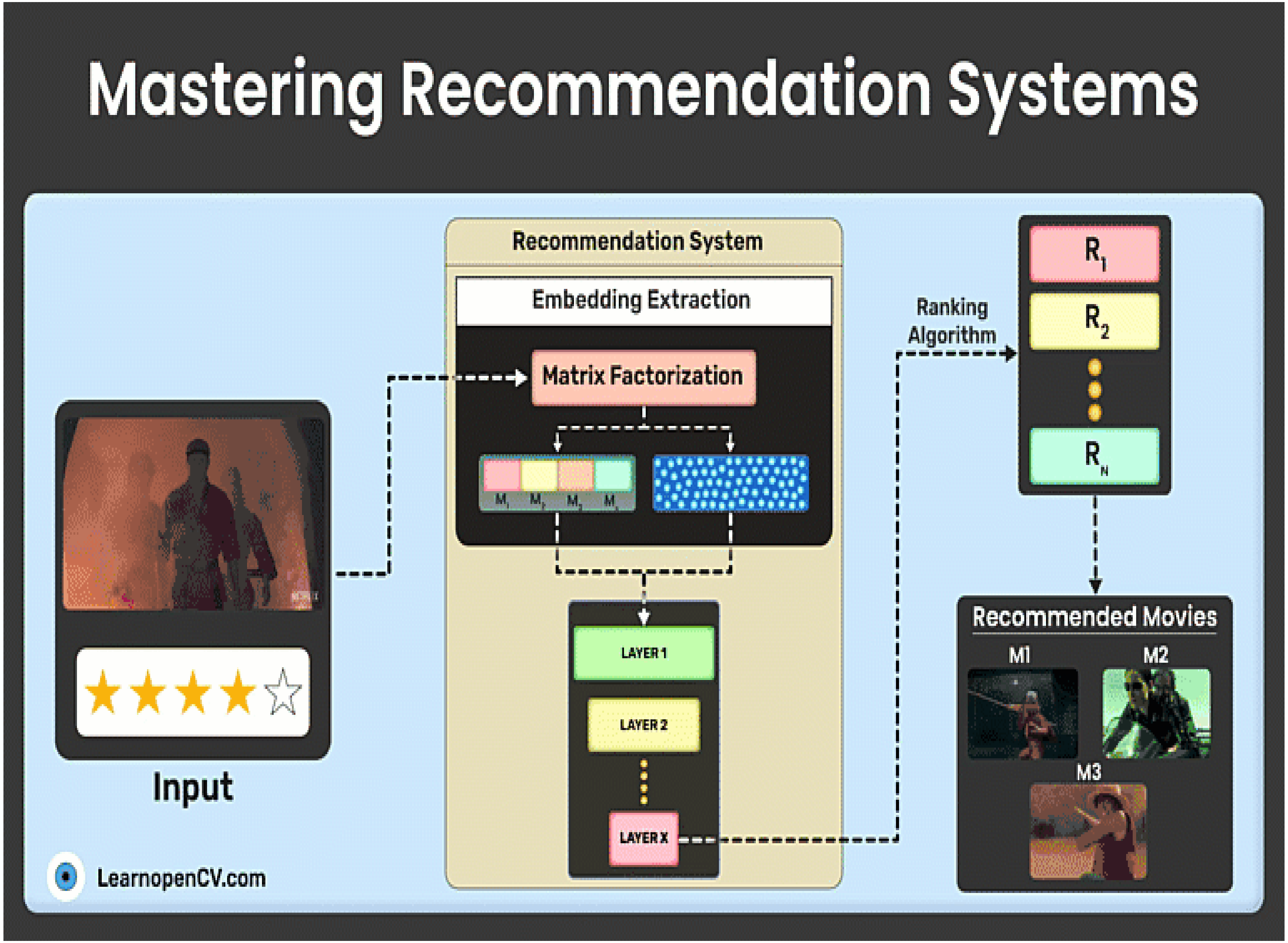
r_x, r_y as points:
r_x = {1, 0, 0, 1, 3}
r_y = {1, 0, 2, 2, 0}

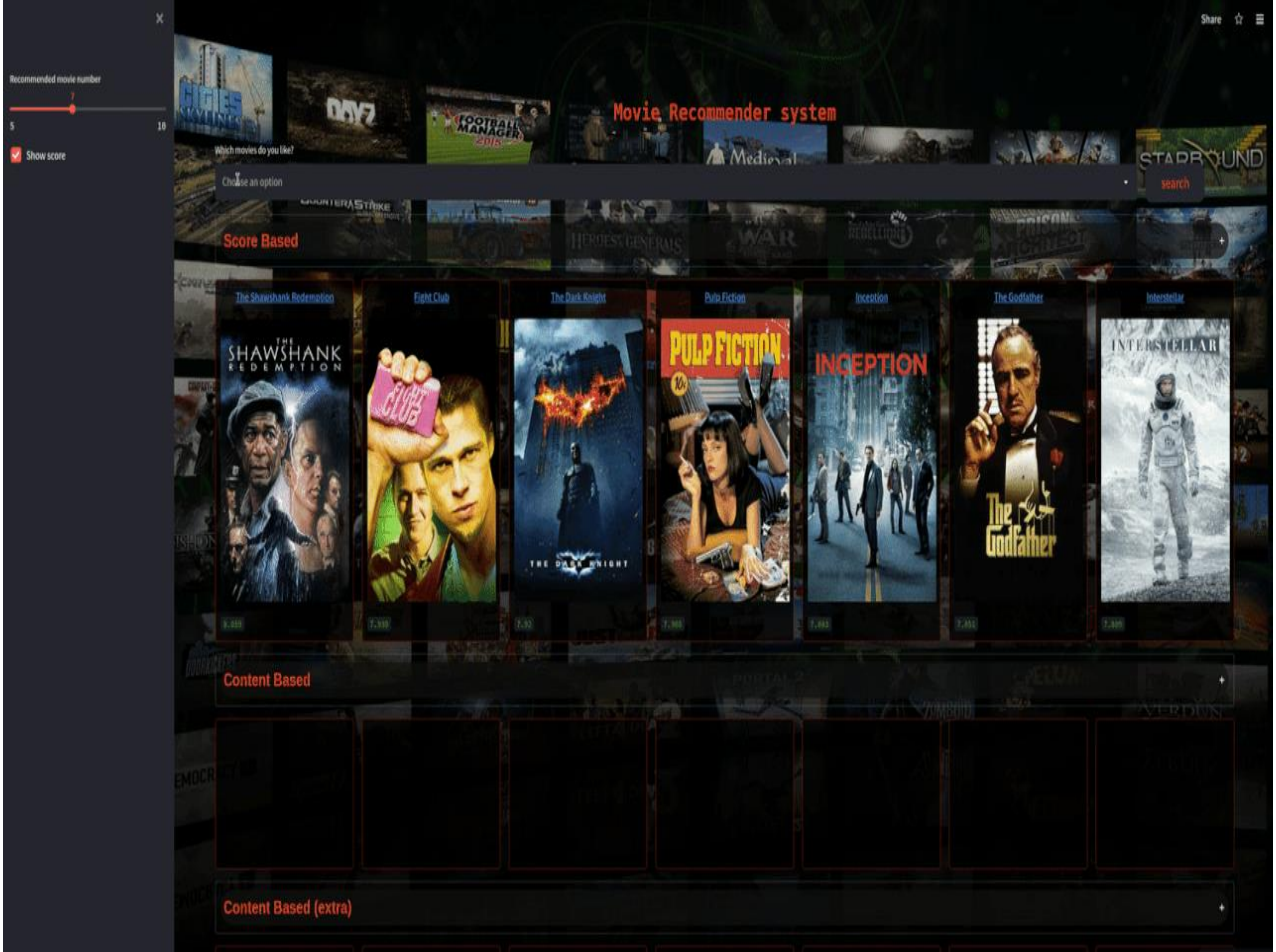
$\overline{r}_x$, $\overline{r}_y$ … avg. rating of **x**, **y**

Pros/Cons of Collaborative Filterin

- **+ Works for any kind of item**
  - No feature selection needed
- **- Cold Start:**
  - Need enough users in the system to find a match
- **- Sparsity:**
  - The user/ratings matrix is sparse
  - Hard to find users that have rated the same items
- **- First rater:**
  - Cannot recommend an item that has not been previously rated
  - New items, Esoteric items
- **- Popularity bias:**
  - Cannot recommend items to someone with unique taste
  - Tends to recommend popular items





Mastering Recommendation Systems
Recommendation System
Embedding Extraction
Matrix Factorization
LAYER 1
LAYER 2
LAYER X
Input
Ranking Algorithm
R1
R2
RN
Recommended Movies
M1
M2
M3
LearnopenCV.com


Share
Recommended movie number
7
5
10
Show score
Movie Recommender system
Which movies do you like?
Choose an option
search
Score Based
The Shawshank Redemption
Fight Club
The Dark Knight
Pulp Fiction
Inception
The Godfather
Interstellar
Content Based
Content Based (extra)