# CS-4053 Recommender System

Fall 2023

## Lecture 1: Introduction

Course Instructor: Syed Zain Ul Hassan

National University of Computer and Emerging Sciences, Karachi

*Email: zain.hassan@nu.edu.pk*

# What is this course about?

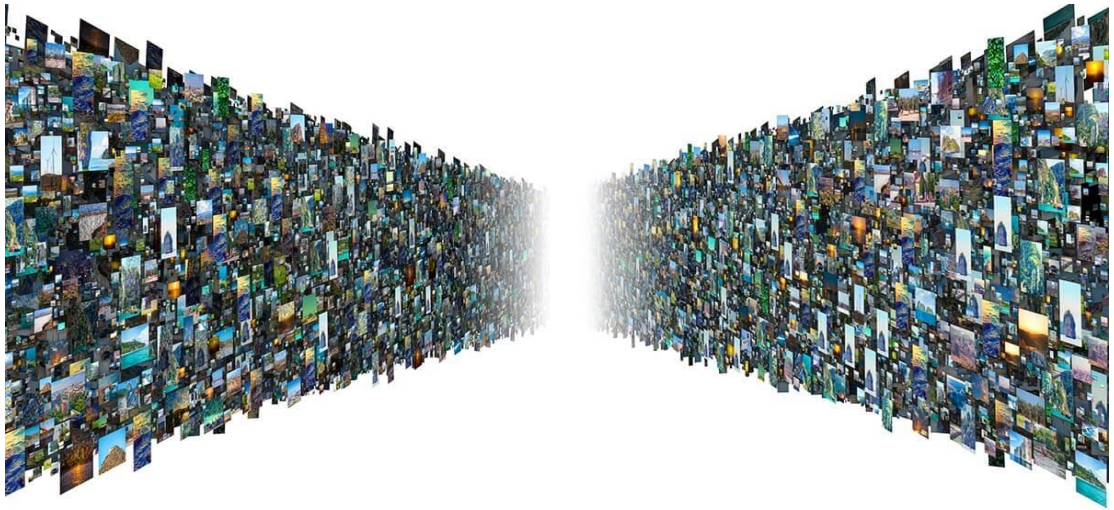
- Understanding ***systems*** that ***recommend*** something
  *(obviously — it's the title of this course!)*

# Too much information

- At times, we can suffer from **information overload**

- *“It’s good to have choices as long as there aren’t too many”*

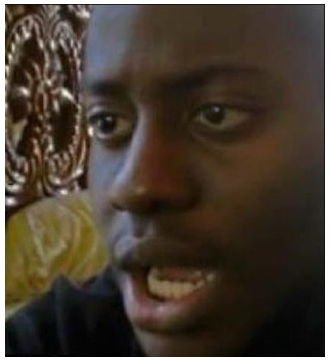
*— Made-up quote*



# Too much information

- Have to try five new flavors of ice cream and tell your favorite

- Now try ten million flavors of ice cream and tell your favorite



# Importance of Recommendation

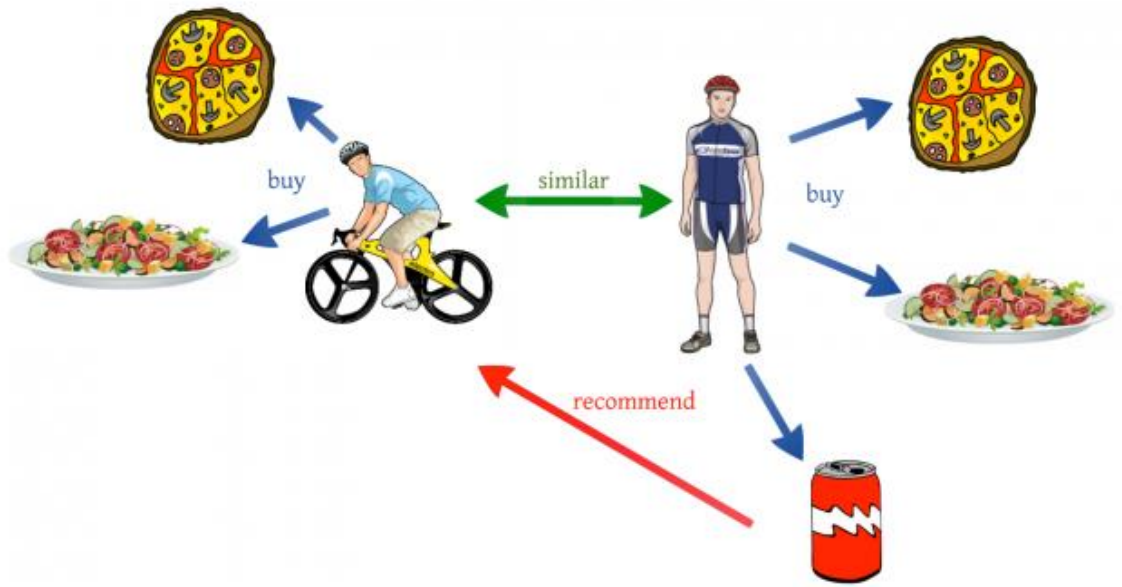
- Why are recommendations so important?
- Value of recommendation
  - *To increase sales*
  - *To improve user experience*
  - *To maximize productivity*
- Both **Spotify** and **Netflix** rely heavily on songs and movies recommendations to improve user experience
- A large part of sales for **Amazon** are recommendation-based

# Recommendation

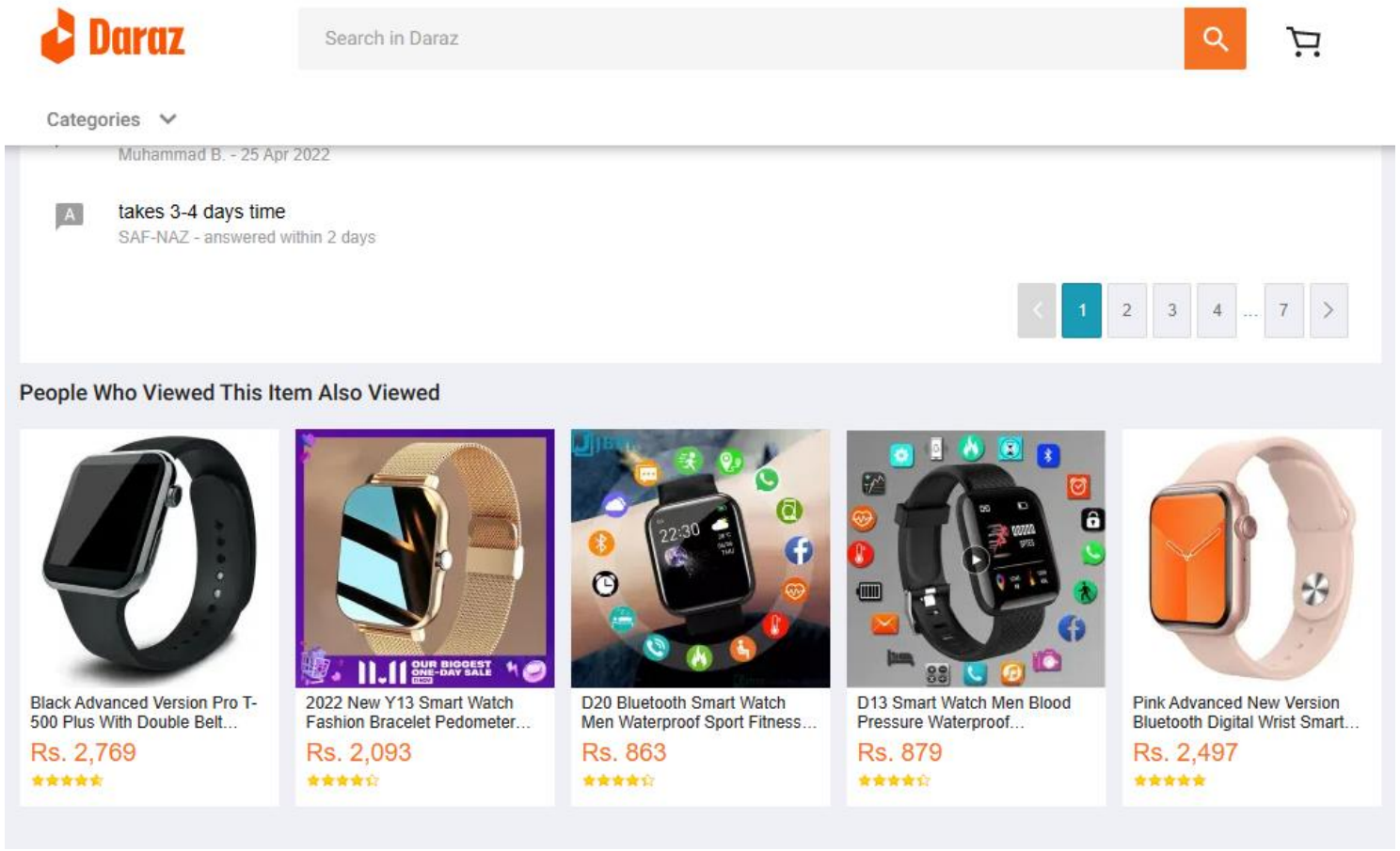
- A suggestion to help user in decision-making

## Example

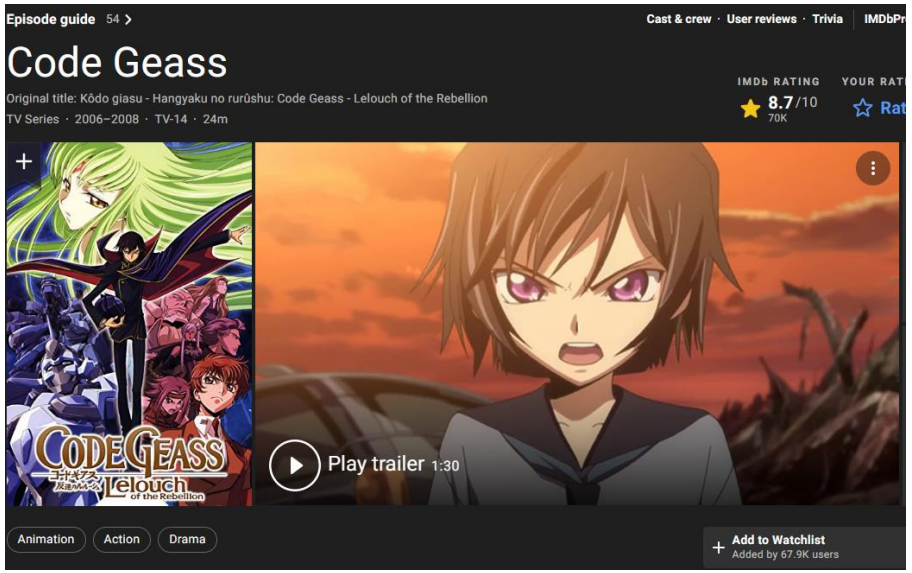
If **cyclist A** always orders *pizza* and *salad* at our food joint, and **cyclist B** eats *pizza* and *salad* as well. Then recommend cola to **cyclist A** if **cyclist B** had been ordering it lately

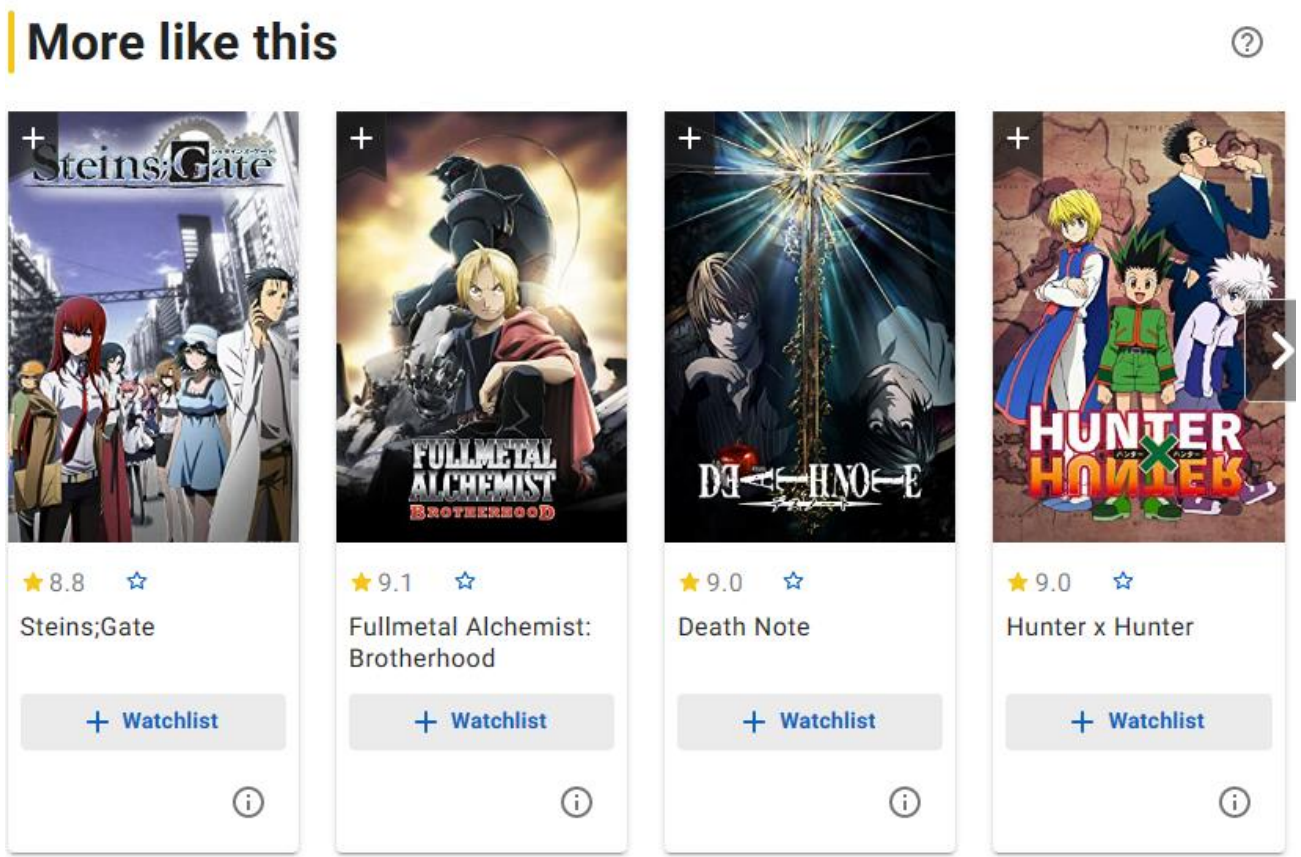


# Recommendation: Example



# Recommendation: Example



# Formulation for Recommender System

- **U** = {*Users*}
- **I** = {*Items*}
- $\boldsymbol{f}$ = utility function measuring the usefulness of item $\boldsymbol{i}$ to user $\boldsymbol{u}$

$$\boldsymbol{f} : \boldsymbol{U} \times \boldsymbol{I} \rightarrow \boldsymbol{R}$$

- $\boldsymbol{R}$ = {*Recommended Items — strictly an ordered list*}

- For each user $\boldsymbol{u}$, what we want to do is to choose the item $\boldsymbol{i}$ that maximizes $\boldsymbol{f}$

$$\boldsymbol{S}_u = \underset{i}{\textbf{argmax}}\, \boldsymbol{f}(\boldsymbol{u},\boldsymbol{i})$$

# Recommender System: Formal Definition

- **A Recommender System (RS)** helps match users with items
  - *It eases information overload*
  - *It helps improve user experience, increase sales and suggest novel items*

# Recommender System: How does it work?

- Recommendations for a user may be based on:
  - *Past behavior (pattern) of that user*
  - *Similarity with other users*
  - *Item similarity*
  - *Context*

*so on...*

# Paradigms of Recommender System

- Some of the approaches to designing a Recommender System are:
  - *Collaborative*
  - *Content-based*
  - *Demography-based*
  - *Social or trust-based*
  - *Hybrid*

# Collaborative Approach

- **Collaborative:** *"Tell me what's popular among my peers“*

- **Collaborative Filtering (CF)** is still one of the most common technique to develop a Recommender System based on collaborative paradigm

- **Collaborative Filtering** can be:
  - **User-based:** Find users *similar* to me and recommend to me what they like
  - **Item-based:** Recommend to me an item *similar* to the ones I normally like

# Collaborative Approach

- **Collaborative:** *"Tell me what's popular among my peers“*

- **Collaborative Filtering (CF)** is still one of the most common technique to develop a Recommender System based on collaborative paradigm

- **Collaborative Filtering** can be:
  - **User-based:** Find users *similar* to me and recommend to me what they like
  - **Item-based:** Recommend to me an item *similar* to the ones I normally like

*Question: How do you measure this “similarity”?*

# Content-based Approach

- **Content-based:** *"Recommend to me an item based on the features of what I just liked or searched"*

- This approach does not require usage history for other users or the user's own usage pattern i.e. minimal collaboration

# Demography-based Approach

- **Demography-based:** *"Recommend to me an item keeping in view either my age, where I live or my education"*

- The main idea behind Knowledge-based Recommendation Systems

# Other Approaches

- Social or trust-based approach involves recommendations based on what people in a user's connections like
  - **Example:** Facebook *groups recommendations*

- Hybrid approaches are usually a combination of two or more of the previously mentioned design strategies

# Serendipity

- Expand the user's taste into neighboring areas
  **Basic Idea:** At times, it's good to recommend something different to the user

# Summary

- Course Introduction
- Recommender System basics
- Common approaches

**References**
*Ladislav Pleska NSWI66 lectures*
*X. Amat's NF Prize presentation*
*Recommender System handbook by Ricci et al*