Data Schools

Module 5:
Introduction to Machine Learning

Recommender Systems

Agenda Topics

1. Overview: What is Machine learning
   - *Building Classification Model – Lab Exercises*
2. **Recommender Systems**
   - ***Building a Recommender Engine – Lab Exercises***
3. From ML to Deep Learning
   - The Rise of Gen AI – Discussion Topic

Learning Objectives

Upon successful completion of this topic, you will be able to:

- Define machine learning
- Describe the categories of machine learning
- Decide when to leverage Machine learning
- Build a simple classifier model
- Discuss approaches to ML application development
- Differentiate between the ML approaches and motivations
- Build a simple recommender engine
- Good insight to Deep Learning & Gen AI

Recommender Systems

Overview

Objectives

Objectives

- What is a Recommender System
- What is the difference between content based and collaborative filtering Recommender systems
- Which limitations recommender systems frequently encounter
- How collaborative filtering can identify similar users and items

Outline

- What is a recommender system?
- Types of collaborative filtering
- Limitations of recommender systems
- Fundamental concepts
- Essential points
- Conclusion
- Hands-On Exercise: Implementing a Basic Recommender

Definition and Purpose of Recommendation Systems

Data Analysis and Machine Learning
Recommendation systems use data analysis and machine learning techniques to learn about a user's behavior and preferences, and to suggest items that the user is more likely to engage with.

Personalized Content
The purpose of recommendation systems is to provide personalized, relevant content that users are more likely to engage with, based on their behavior and preferences.

Examples of Platforms Using Recommendation Systems

Netflix
Netflix uses recommendation systems to suggest movies and TV shows to users based on their viewing history and preferences.

Spotify
Spotify uses recommendation systems to suggest music to users based on their listening history and preferences.

Amazon
Amazon uses recommendation systems to suggest products to users based on their purchase history and browsing habits.

YouTube
YouTube uses recommendation systems to suggest videos to users based on their viewing history and preferences.

Types of Recommendations

1. **Content-based (CB)**

Analyze attributes of items for building user profiles

2. **Collaborative filtering (CF)**

Inspect rating patterns to find similar users/items

In general, CF performs **better** than CB

- CF fail to provide accurate predictions with insufficient ratings
- CB can alleviate the sparsity problem

Content-Based Recommendations

1. **Focus:**
   - Content-based systems recommend items to users based on the **attributes** and **characteristics** of the items themselves and the user's historical preferences for those attributes.
2. **User Profile:**
   - These systems create a user profile by analyzing the content or features of items the user has interacted with. The user profile captures the user's preferences for different attributes.
3. **Similarity Calculation:**
   - Recommendations are made by calculating the similarity between the user's profile and the attributes of different items.

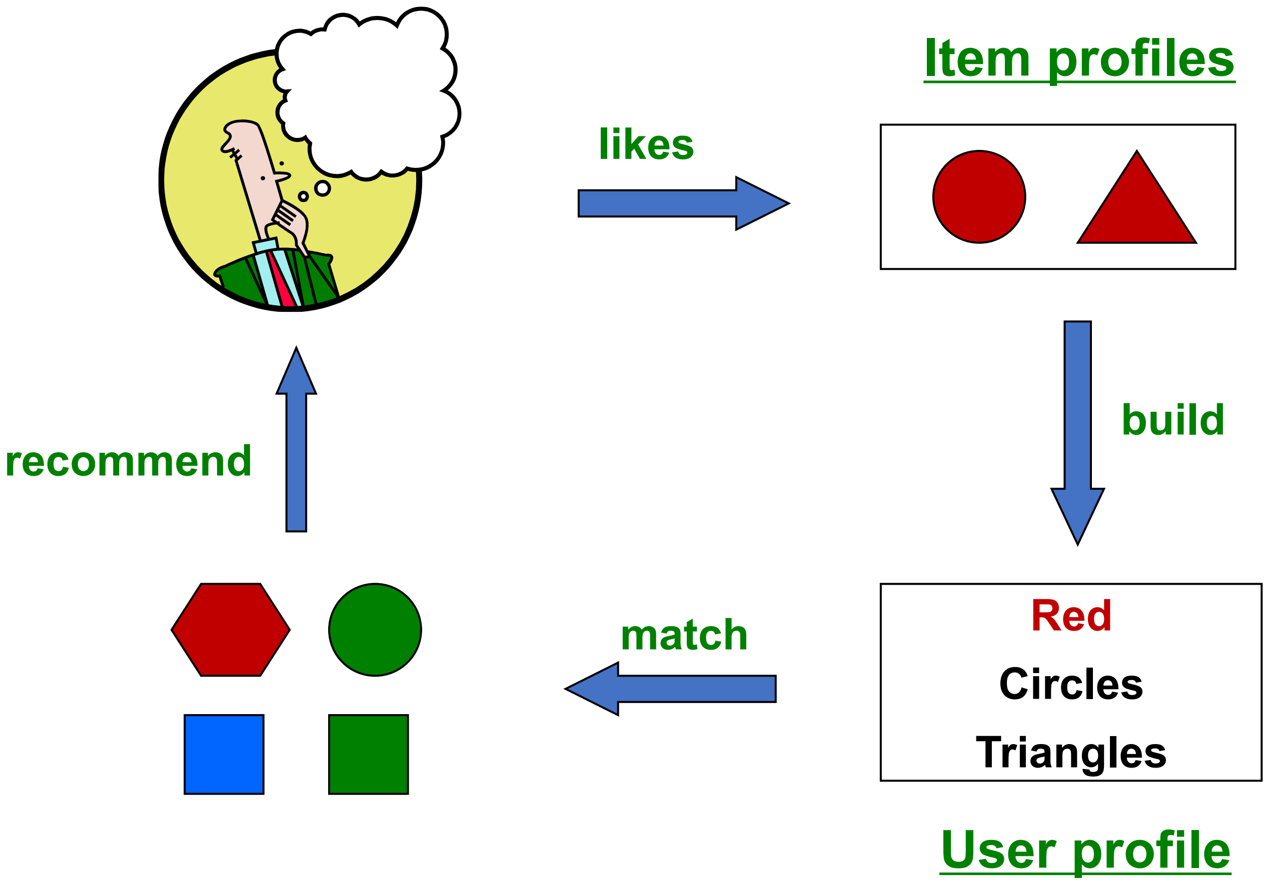
Content-Based Recommendations

- **Main idea:** Recommend items to customer ***x*** like 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



Items Profile

For each item, create an **item profile**

Profile is a set (vector) of features

- **Movies:** author, title, actor, director,…
- **Text:** Set of “important” words in document

How to pick important features?

- Usual heuristic from text mining is **TF-IDF** (Term frequency * Inverse Doc Frequency)
  - **Term … Feature**
  - **Document … Item**

Sidenote: TF-IDF

f_{ij} = frequency of term (feature) i in doc (item) j

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$$

Note: we normalize TF to discount for "longer" documents

n_i = number of docs that mention term i

N = total number of docs

$$IDF_i = \log \frac{N}{n_i}$$

TF-IDF score: $w_{ij} = TF_{ij} \times IDF_i$

Doc profile = set of words with highest **TF-IDF** scores, together with their scores

User Profiles and Prediction

- **User profile possibilities:**
  - Weighted average of rated item profiles
  - **Variation:** weight by difference from average rating for item
- **Prediction heuristic: Cosine similarity of user and item profiles)**
  - Given user profile $\boldsymbol{x}$ and item profile $\boldsymbol{i}$, estimate $u(\boldsymbol{x}, \boldsymbol{i}) = \cos(\boldsymbol{x}, \boldsymbol{i}) = \frac{\boldsymbol{x} \cdot \boldsymbol{i}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{i}||}$
- **How do you quickly find items closest to x?**
  - Job for LSH!

Pros & Cons: Content-based Approach

1. **Advantages**:
    1. Can provide personalized recommendations even for new or less popular items, if their attributes are known.
    2. Less reliant on large user interaction data.
    3. Can handle the cold-start problem for new users.
2. **Limitations**:
    1. Limited to the features available for item descriptions.
    2. May not capture changes in a user's preferences over time.
    3. Tends to produce recommendations that are like past interactions.

Collaborative Filtering

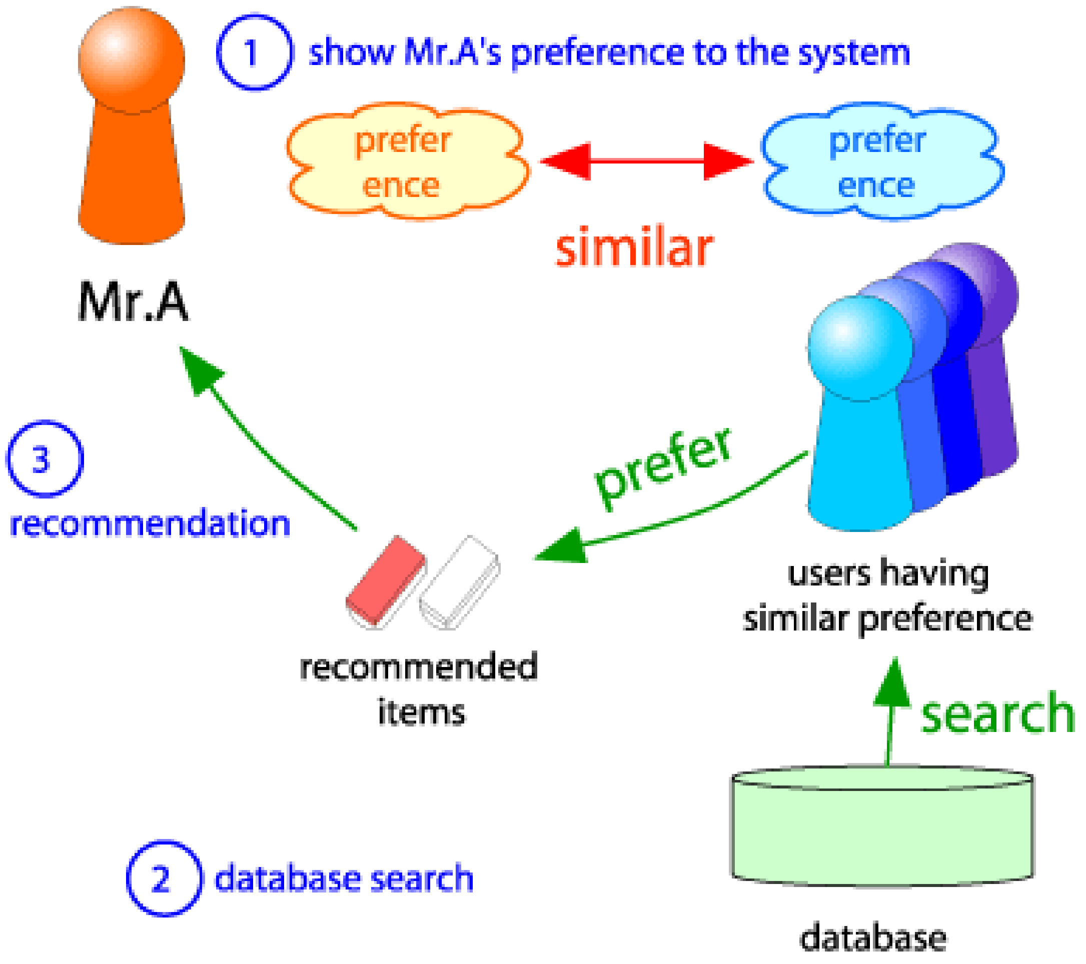
- **Principle**: Collaborative filtering recommends items to users based on the preferences and behaviors of other users. It assumes that users who agree in the past will agree in the future.
- **User-Item Matrix**: It creates a user-item interaction matrix where each entry represents the user's preference for an item (e.g., ratings, likes, purchase history).
- **User-Based vs. Item-Based**:
  - User-Based Collaborative Filtering: Recommends items to a user based on the preferences of similar users.
  - Item-Based Collaborative Filtering: Recommends items to a user based on the preferences of other items they have interacted with.

Types of Collaborative Filtering

- **Collaborative filtering can be subdivided into two main types**
- **User-based: "What do users similar to you like?"**
  - For a given user, find other people who have similar tastes
  - Then, recommend items based on past behavior of those users
- **Item-based: "What is similar to other items you like?"**
  - Given items that a user likes, determine which items are similar
  - Make recommendations to the user based on those items

User-Based 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

- Let $\boldsymbol{r_x}$ be the vector of user $\boldsymbol{x}$'s ratings
- **Jaccard similarity measure**
  - **Problem:** Ignores the value of the rating
- **Cosine similarity measure**
  - sim($\boldsymbol{x}$, $\boldsymbol{y}$) = cos($\boldsymbol{r_x}$, $\boldsymbol{r_y}$) = $\frac{r_x \cdot r_y}{||r_x|| \cdot ||r_y||}$
  - **Problem:** Treats some missing ratings as "negative"
- **Pearson correlation coefficient**
  - $\boldsymbol{S_{xy}}$ = items rated by both users $\boldsymbol{x}$ and $\boldsymbol{y}$

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

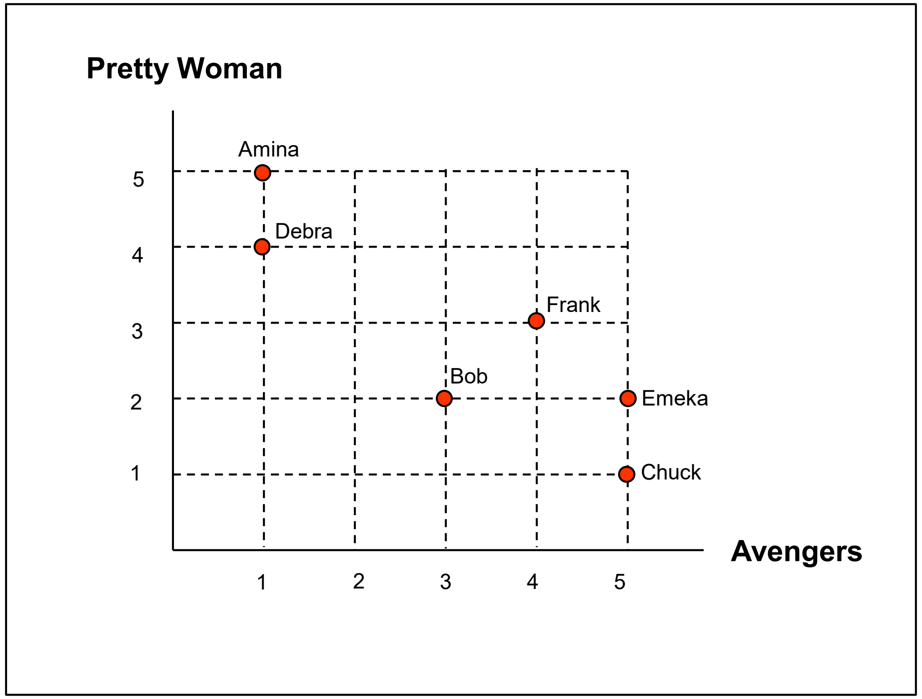
Rating Predictions

From similarity metric to recommendations:

- Let $\boldsymbol{r_x}$ be the vector of user $\boldsymbol{x}$'s ratings
- Let $\boldsymbol{N}$ be the set of $\boldsymbol{k}$ users most similar to $\boldsymbol{x}$ who have rated item $\boldsymbol{i}$
- **Prediction for item *i* of *user x*:**
  - $r_{xi} = \frac{1}{k} \sum_{y \in N} r_{yi}$
  - Or even better: $r_{xi} = \frac{\sum_{y \in N} s_{xy} \cdot r_{yi}}{\sum_{y \in N} s_{xy}}$
- **Many other tricks possible…**

User-Based Collaborative Filtering

- **User-based collaborative filtering is social**
  - It takes a “people first” approach, based on common interests
- **In this example, Amina and Debra have similar tastes**
  - Each is likely to enjoy a movie that the other rated highly



Item-Based Collaborative Filtering

- **So far: User-based collaborative filtering**
- **Another view: Item-based**
  - For item ***i***, find other similar items
  - Estimate rating for item ***i*** based on ratings for similar items
  - Can use same similarity metrics and prediction functions as in user-user model

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

s_{ij}… similarity of items ***i*** and ***j***
r_{xj}…rating of user ***x*** on item ***j***
N(i;x)… set items rated by ***x*** similar to ***i***

Item-Based Collaborative Filtering

users

movies	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3			5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

- unknown rating

- rating between 1 to 5

Item-Based Collaborative Filtering

users

movies	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

- estimate rating of movie **1** by user **5**

Item-Based Collaborative Filtering

Data Schools

users

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		<u>0.41</u>
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		<u>0.59</u>

Neighbor selection:
Identify movies similar to movie **1**, **rated by user 5**

Here we use Pearson correlation as similarity:

1) Subtract mean rating m_i from each movie i

$m_1 = (1+3+5+5+4)/5 = 3.6$

row 1: [-2.6, 0, -0.6, 0, 0, 1.4, 0, 0, 1.4, 0, 0.4, 0]

2) Compute cosine similarities between rows

Item-Based Collaborative Filtering

movies \ users	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		<u>0.41</u>
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		<u>0.59</u>

Compute similarity weights:

$s_{1,3}$=0.41, $s_{1,6}$=0.59

Item-Based Collaborative Filtering

users

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		0.41
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		0.59

Compute similarity weights:

$s_{1,3}=0.41$, $s_{1,6}=0.59$

Pros & Cons: Collaborative Filtering

1. **Advantages**:
   1. Captures complex user behaviors and preferences.
   2. Can discover hidden patterns in user interactions.
2. **Limitations**:
   1. Can suffer from the cold-start problem for new items or users.
   2. Sensitive to sparsity in the user-item interaction matrix.

Summary

Recommendation systems use several different technologies. We can classify these systems into two broad groups.

- Content-based systems examine properties of the items recommended.
- Collaborative filtering systems recommend items based on similarity measures between users and/or items.