Recommendation System

Notes by Mannan Ul Haq (BDS-4A)

There are so many choices that people often feel trapped, whether they're trying to choose a movie to watch, the right product to buy, or new music to listen to. ***To solve this problem, recommendation systems comes into play that help people find their way through all of these choices by giving them unique ideas based on their likes and dislikes.***

Algorithms in recommendation systems evaluate user data, such as prior purchases, reviews, or browsing history, to find trends and preferences to utilize this information for recommending goods that are likely to interest the user.

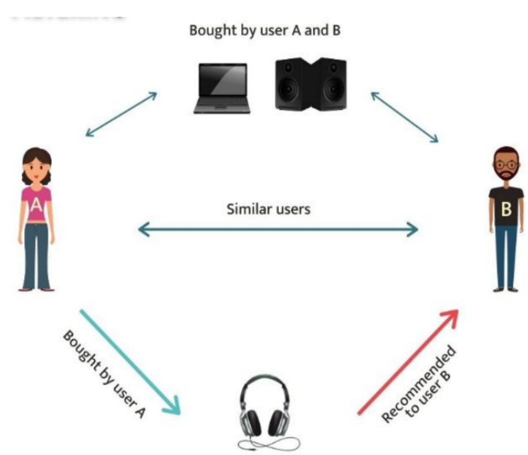
Examples Of Recommendation Systems:

- Online e-commerce model such as Amazon recommend goods based on your browsing and purchase history.
- Music streaming services like Spotify, propose songs and artists based on your listening history.
- Podcast streaming providers such as Netflix recommend movies and TV series based on your watching history.

Types of Recommendation Systems

1. Collaborative filtering:

Which makes recommendations based on the behavior and preferences of similar users. Match **`like-minded'** people. For **example**, if two customers have purchased similar products in the past, the algorithm may suggest similar products to both customers.



Key Concepts:

- **Match Similar Interests**: The system identifies users with similar interests to form the basis of recommendations.
- **User Participation**: A large number of users must participate to increase the likelihood of finding someone with similar interests.
- **Expressing Interests**: There must be an easy way for users to express their interests, such as rating items or making purchases.
- **Efficient Algorithms**: An effective algorithm is needed to match users with similar interests.

Ratings and Interests:

Users rate items, which records their interests. Ratings can be:

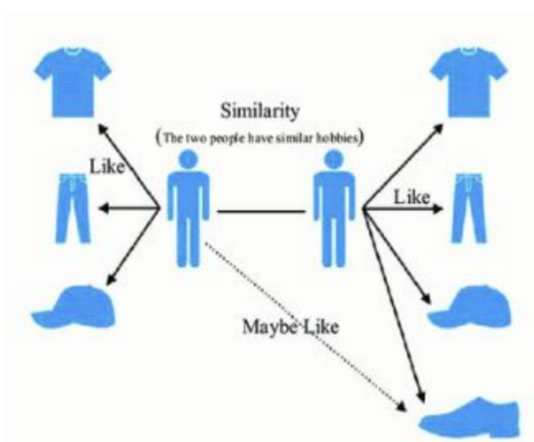
- **Explicit**: Direct actions like buying or rating an item.
- **Implicit**: Indirect actions like browsing time or the number of mouse clicks.

Types of Collaborative Filtering:

There are two main types of collaborative filtering:

User-Based Collaborative Filtering:

- **How it Works**: Finds users with similar past interactions with items and recommends items liked by similar users.
- **Example**: If two users have similar viewing histories on Netflix, the system may recommend the same movie to both users.



Example:

We look for the users who have rated various items in the same way and then find the rating of the missing item with the help of these users.

Calculating Similarity

Using Jaccard Distance:

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

$$Sim(A,B) = |r_A \cap r_B| / |r_A \cup r_B|$$

$$= 1/5$$

$$Sim(A,C) = 2/4$$

∵ Problem: Ignores rating values

Using Cosine Distance:

$$Sim(A,B) = \cos(x,y) = \frac{x.y}{\|x\| \cdot \|y\|}$$

$$= \frac{4 \times 5}{(\sqrt{4^2+5^2+1^2})(\sqrt{5^2+5^2+4^2})} = 0.380$$

$$Sim(A,C) = \frac{5 \times 2 + 1 \times 4}{(\sqrt{4^2+5^2+1^2})(\sqrt{2^2+4^2+5^2})} = 0.322$$

∵ Problem: Treat missing ratings as negative

Using Centered Cosine Similarity:

	M_1	M_2	M_3	M_4
Alice	5	4		3
Bob		5	4	2
Charlie	4		3	5
David	3	2	5	

Compute Avg:

	M_1	M_2	M_3	M_4	Avg
Alice	5	4		3	4
Bob		5	4	2	3.66
Charlie	4		3	5	4
David	3	2	5		3.33

Normalize:

	M_1	M_2	M_3	M_4
Alice	1	0		-1
Bob		1.34	0.34	-1.66
Charlie	0		-1	1
David	-0.33	-1.33	1.67	

$$Sim(A,B) = \cos(A,B) = \frac{\sum A_i B_i}{\sqrt{\sum A_i^2} \times \sqrt{\sum B_i^2}}$$

$$sim(Alice, Bob) = \frac{0 \times 1.34 + (-1) \times -1.66}{\sqrt{1^2+0^2+(-1)^2}\sqrt{1.34^2+0.34^2+(-1.66)^2}}$$

$$= 0.543$$

$sim(Alice, Charlie) = -0.5$

$sim(Alice, David) = -0.108$

$sim(Bob, Charlie) = -0.6546$

$sim(Bob, David) = -0.26$

$sim(Charlie, David) = -0.546$

Find the missing ratings.

Using formula :- $\frac{\sum s.r}{\sum s}$

Where "r" is the corresponding rating

Where "s" is the similarity value

∵ Above formula is only applicable for positive similarity values

For Alice $= \frac{0.55 \times 0.34}{0.55} = 0.34$

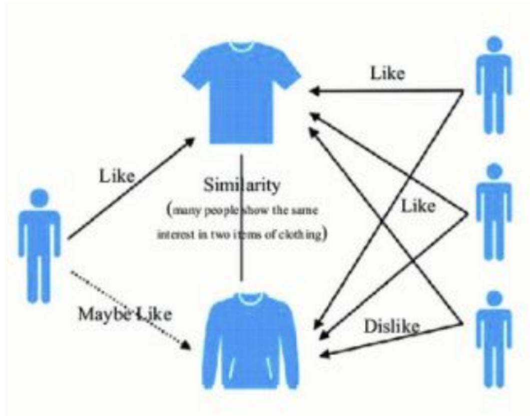
For Bob $= \frac{0.55 \times 1}{0.55} = 1$

For Charlie = We can't compute

For David = We can't compute

Item-Based Collaborative Filtering:

- **How it Works**: Finds items that are similar based on how users have interacted with them and recommends those similar items.
- **Example**: If a user has liked several movies of a particular genre, the system may recommend other movies of that genre to the user.



In centered cosine similarity in item-item collaborative filtering, please do note that that after taking similarity of item "A" with rest of the items, no matter how many positive probabilities are computed, you only apply the formula of rating prediction on the top N probabilities given that they are positive, let's say if N=3 but only a single positive probability was computed, then you'll only use that probability value in the formula because it's the only positive one.

Benefits:

1. **Versatile Recommendations**:

   Can suggest items to users with minimal interaction history by leveraging similar users' data.

2. **Adaptive**:

   Adapts to users' changing preferences over time as they interact with more items.

3. **Diverse Suggestions**:

   Introduces users to new and varied items they might not find on their own.

Drawbacks:

1. **Cold Start Problem**:

   Difficult to recommend for new users or new items due to lack of interaction data.

2. **Small User Base**:

   Struggles to find similar users for reliable recommendations if the user base is too small.

3. **Data Dependency**:

   Requires a large amount of user interaction data to make effective recommendations.

2. Content-based filtering:

Which makes recommendations based on the products' attributes. Use personal preferences to match and filter items. For **example**, the algorithm may suggest other products if a customer has purchased a particular clothing brand.

Concept:

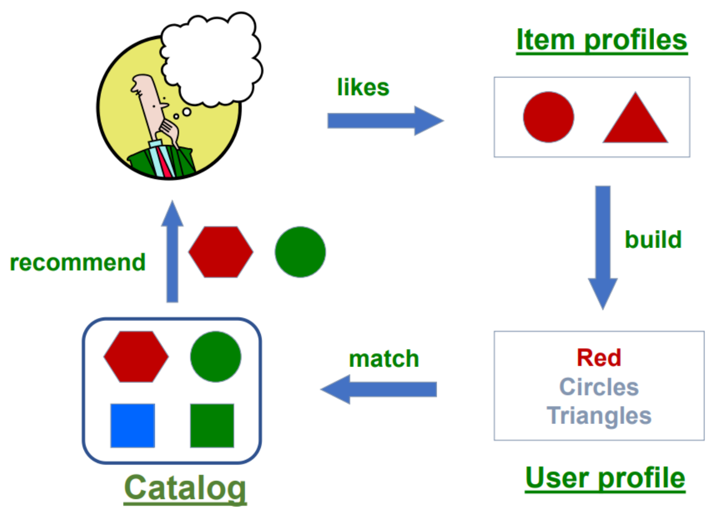
- **Principle**: "Find me things similar to what I liked in the past."
- **Learning Preferences**: The system builds a user profile based on feedback and past interactions.

How It Works:

1. **User Feedback**:
    - **Explicit Feedback**: Users rate items directly.
    - **Implicit Feedback**: The system records user activities, such as browsing history, clicks, and time spent on pages.
2. **Data Classification**:
    - **Clickstream Data**: Activities are categorized (e.g., browsing a product page).
    - **Activity Duration**: Time spent on activities is considered to infer user preferences.

What do we need:

- some information about the available items such as the genre ("content")
- some sort of user profile describing what the user likes (the preferences)



Example:

Numerical Example:

	Feature 1	Feature 2	Feature 3	Feature 4
Product 1	1		1	1
Product 2		1	4	
Product 3	3			1
User Interest	2		1	1

$$User\ Interest\ Level\ = \sum_{i=1}^{d} p_i u_i$$

where

p_i is the product feature value and u_i user interest value in column i

Numerical Example:

	Feature 1	Feature 2	Feature 3	Feature 4
Product 1	1		1	1
Product 2		1	4	
Product 3	3			1
User Interest	2		1	1

$$User\ Interest\ Level\ Product\ 1 = 2 * 1 + 1 * 1 + 1 * 2 = 5$$

$$User\ Interest\ Level\ Product\ 2 = \ 1 * 4 = 4$$

$$User\ Interest\ Level\ Product\ 3 = 2 * 3 + 1 * 1 = 7$$

Pros:

1. **Independence from Other Users' Data**:

   No Cold-Start or Sparsity Problems: Recommendations can be made without needing data on other users.

2. **Personalized Recommendations**:

   Unique Tastes: Can recommend items to users with unique or niche preferences.

3. **New and Unpopular Items**:

No First-Rater Problem: Can recommend new and less popular items since it doesn't rely on other users' ratings.

4. **Explainability**:

   Clear Explanations: Can explain recommendations by listing content features that influenced the recommendation.

Cons:

1. **Feature Identification**:

   Complexity: Finding the right features to base recommendations on can be challenging, especially for items like images, movies, or music.

2. **New Users**:

   Profile Building: Difficulty in recommending items for new users due to lack of interaction data to build a user profile.

3. **Overspecialization**:

   - **Limited Diversity**: Tends to recommend items similar to what the user has already interacted with, potentially ignoring other interests.
   - **Multiple Interests**: May not capture the full range of a user's interests.
   - **Quality Judgments**: Unable to leverage the quality assessments and preferences of other users.