Recommender Systems

Presented By: Seyed Erfan Nourbakhsh

Master: Dr.Kiani

Fall 2022

Contents

Introduction

Recommender systems capture the pattern of people's behavior and use it to predict what else they might want or like.

Applications

What to buy?
- E-commerce, Books, Movies, Shops, …

Where to eat?

Which job to apply to?

Who you should be friends with?
- LinkedIn, Facebook, …

Personalize your experience on the web
- News platforms, News personalization

Some company's achievements by using recommender systems

- ✓ According to the statistics provided by MacKenzie Scott, **35%** of the purchases made from the **Amazon** retail website were as a result of using the recommender system.
- ✓ Recommendation systems are the main reason for **70%** of people who watch videos on **YouTube**.
- ✓ According to statistics provided by MacKenzie Scott, **75%** of what people watch on **Netflix** is based on the recommendations of the system's recommender.
- ✓ According to the statistics provided by DigiMag, During the last year, the use of the smart recommendation system in **Digikala** has grown by **30%**.
- ✓ ….

Method of providing data

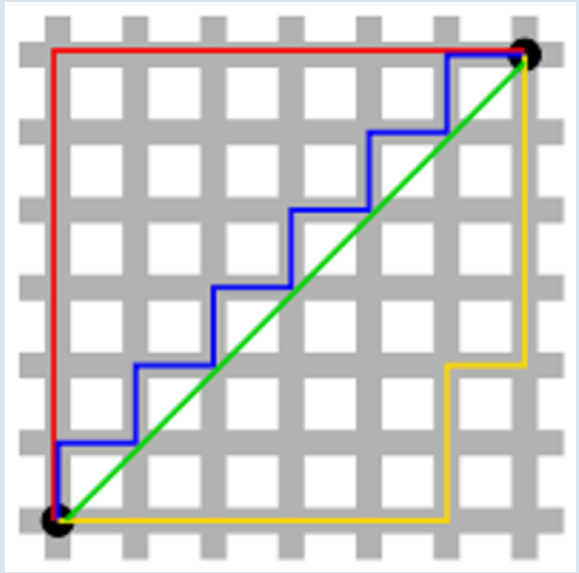
1) **Explicit ranking**
   - ✓ It is done by the users
   - ✓ recommender system extracts the user's opinion directly
   - ✓ Types of ratings:
     - ❖ Star rating
     - ❖ Review
     - ❖ Feedback
     - ❖ Like
     - ❖ Following

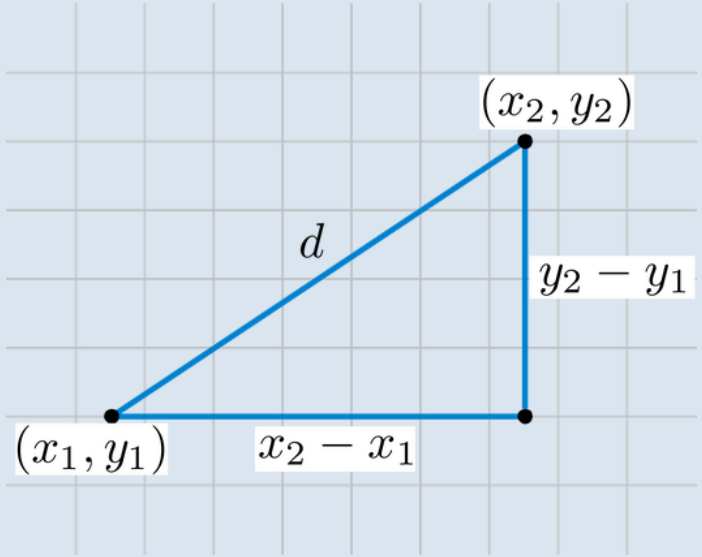
2) **Implicit ranking**
   - ✓ When users interact with items
   - ✓ The recommender system infers user behavior
   - ✓ Types of ratings:
     - ❖ Clicks
     - ❖ Views
     - ❖ Purchases

Some similarity measurements

Similarity measures are **distance** measures. The **closest** points to each other are the **most similar** to each other and the **farthest** points are the **least similar** to each other.

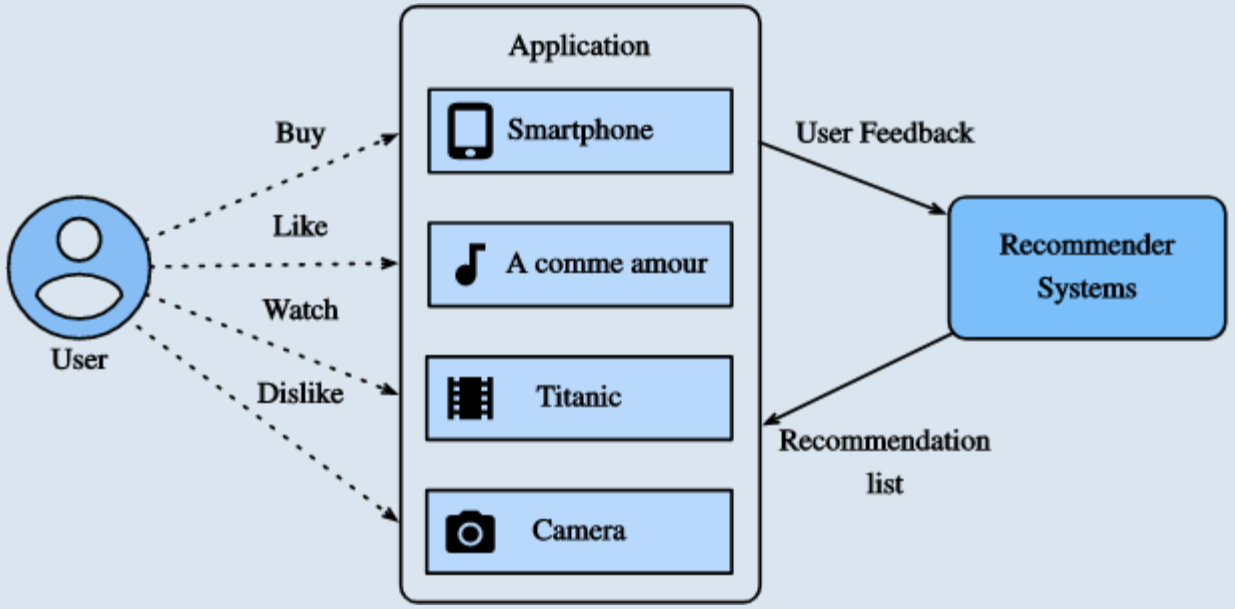
1) Minkowski Distance
2) Manhattan Distance
3) Euclidean Distance
4) Hamming Distance
5) Cosine Similarity
6) Pearson Coefficient
7) Jaccard Index



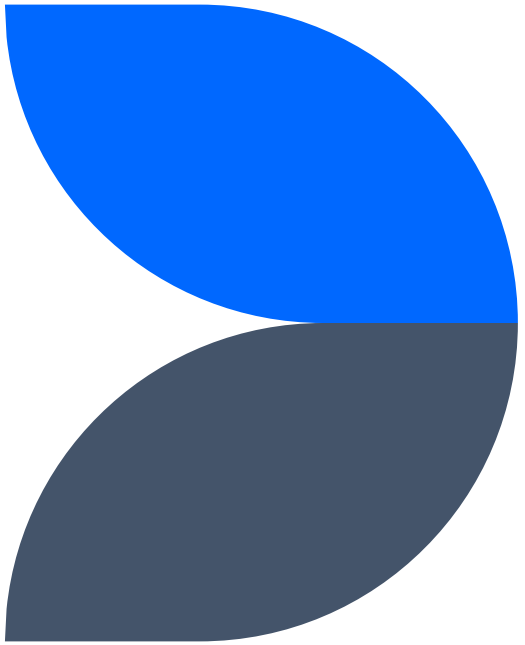


Advantages of recommender systems

- ✓ Broader exposure to many different products that might be interested in
- ✓ Possibility of continual usage or purchase of products
- ✓ Provides better experience (increase potential revenue & better security for customers)
- ✓ Send emails that contain links to suggested items based on the user's preferences
- ✓ Gaining the trust of customers

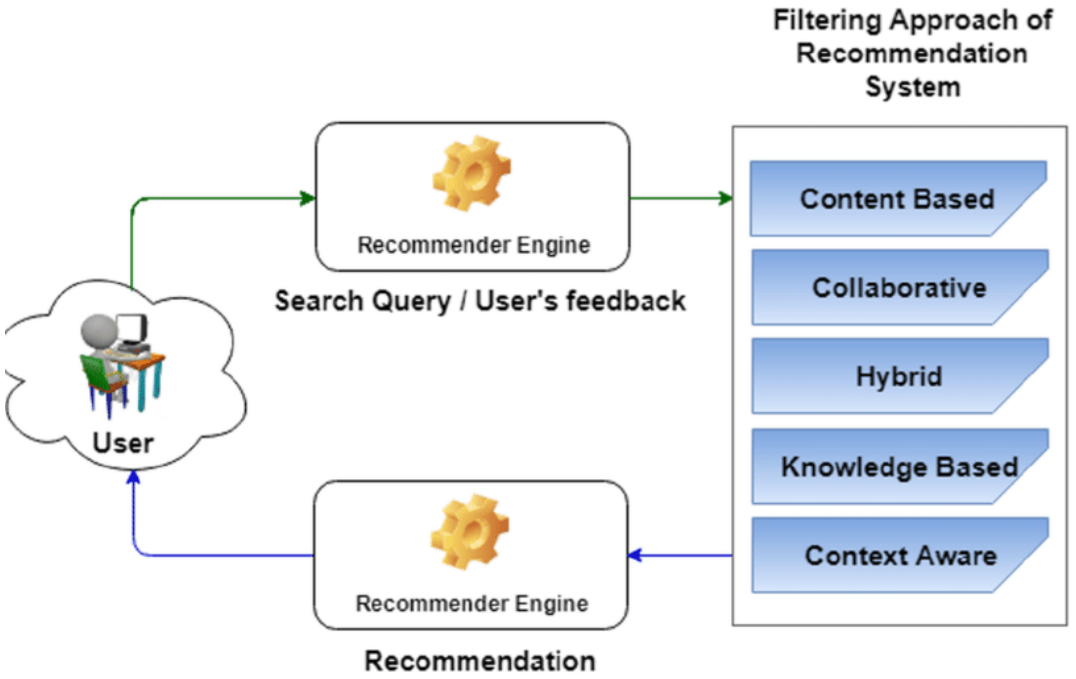


Filtering methods and algorithms



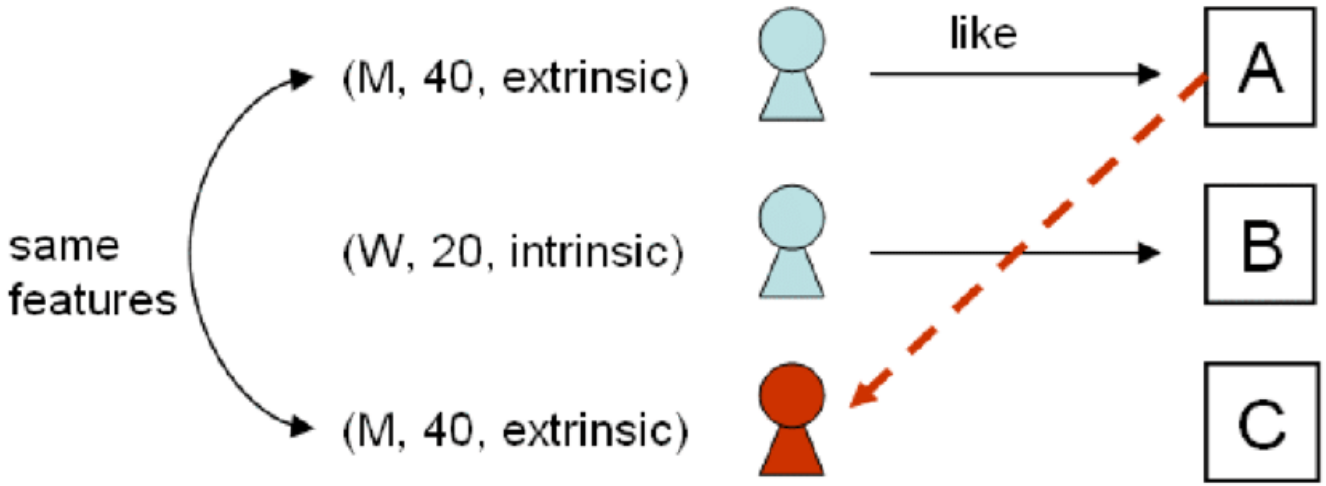
Filtering methods and algorithms

1) Collaborative filtering
2) Demographic filtering
3) Content-based filtering
4) Social-based filtering
5) Context-aware filtering
6) Location-aware recommendation systems
7) Knowledge-based filtering
8) Bio-inspired algorithm



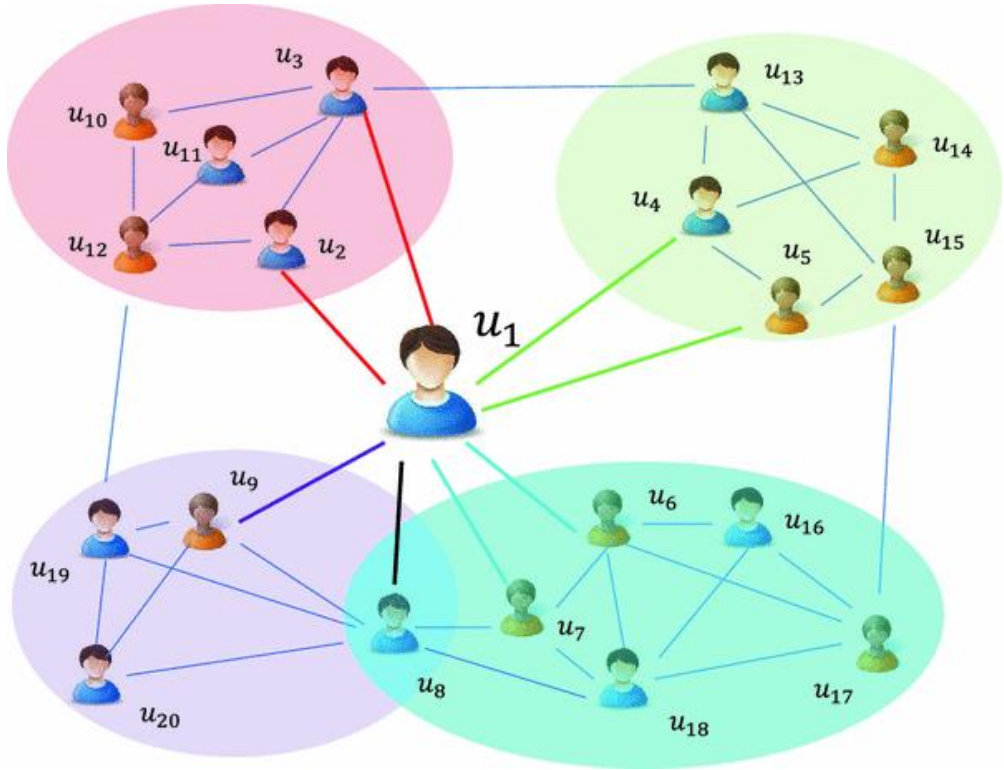
Demographic Filtering

- ✓ Information such as age, gender, nationality, etc. are included in the demographic information group
- ✓ Users who have similar **demographic characteristics** (for example, fall in the same age range) probably have similar **tastes and desires**.



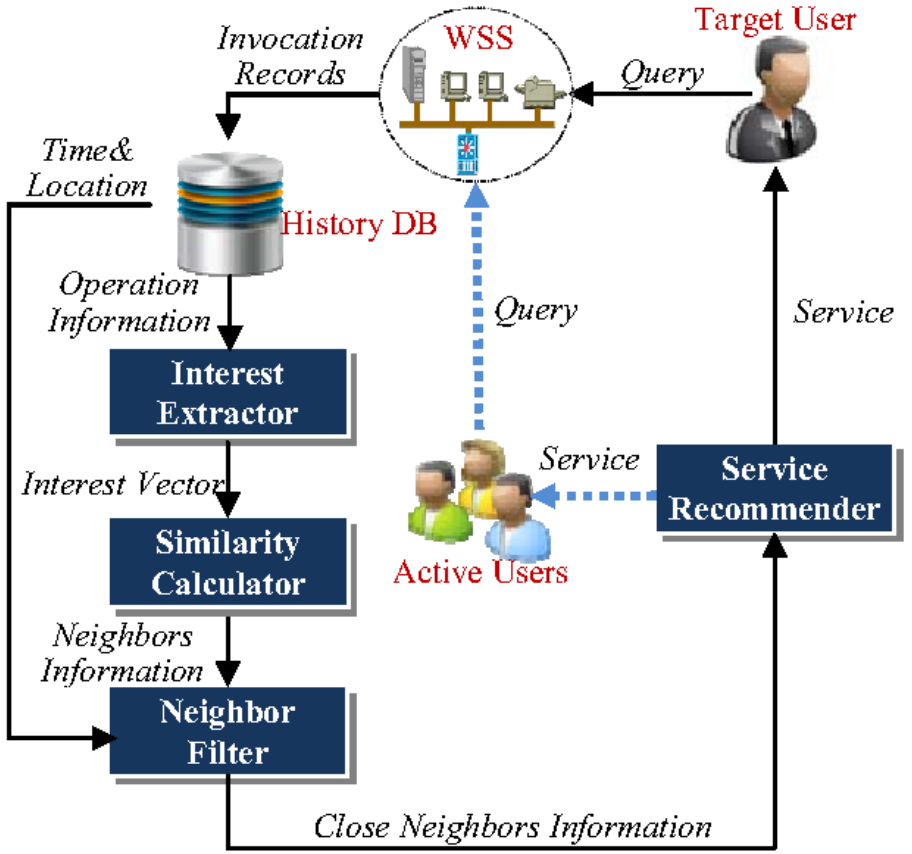
Social-based Filtering

- ✓ Using the information available in these networks (such as trust, followed, followers, friends, comments, blog and tags) in recommender systems
- ✓ Improving the proposed results as well as reducing the sparsity problem
- ✓ Two types of scientific studies and research in social-based filtering:
  - ❖ Improving the efficiency of existing systems
  - ❖ Creating a new recommender system based on Social Filtering (use the potential in such networks to create an independent system)



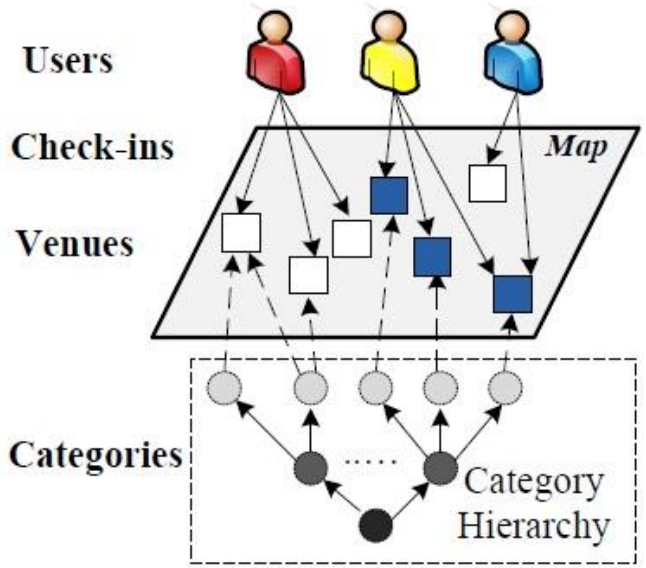
Context-aware Filtering

- ✓ There are various devices and sensors that collect information about the user's context.
- ✓ The emphasis of these systems is on information such as **time**, **location**, information from **security cameras**, **RFIDs** and **wireless sensor networks**, as well as **health parameters**, **shopping** and **eating habits** of a person.



Location-aware recommendation systems

- ✓ Generally, they appear in **mobile phone** applications, based on the **current location** of the user, they give her suggestions in a certain area.
- ✓ When the offer is given to the user, his **geographic information** is also involved in the process of providing the offer

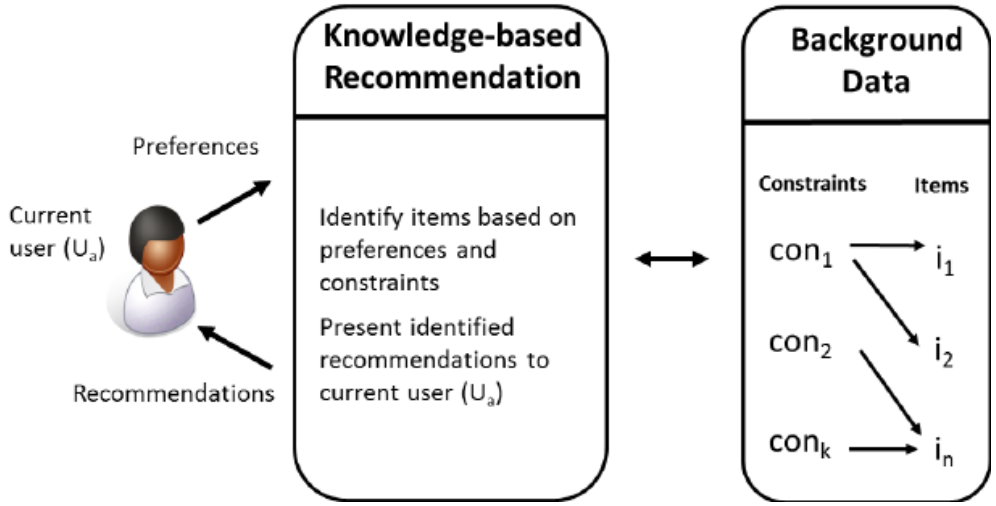


Knowledge-based Filtering

- ✓ Based on **existing knowledge** about users and items
- ✓ They provide their suggestions based on their **interpretation** and **inference** of the user's **tastes and needs**
- ✓ The need for a knowledge-based **platform** and **structure**
  - ❖ Constraint-based reasoning
  - ❖ Case-based reasoning
  - ❖ Social knowledge
  - ❖ Vectors knowledge

Some models in this field:

- ❖ Based on **users-roles-tasks** model (Each of the users perform what roles and what tasks)
- ❖ Based on **peer-to-peer** model(System knowledge about items and users is distributed among peers)



Bio-inspired Algorithm

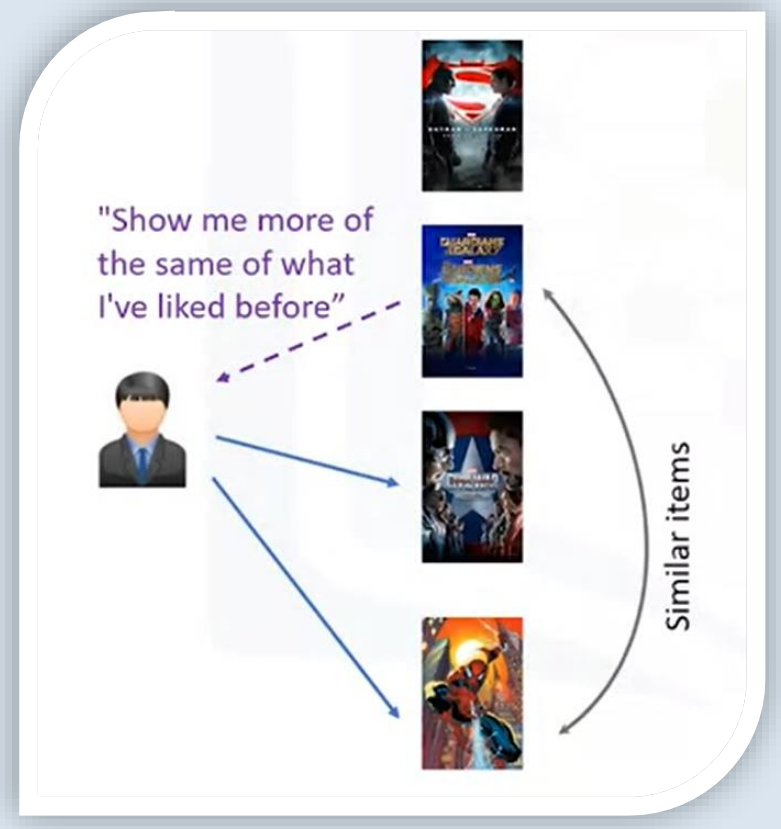
1) **Evolutionary algorithm solutions (Genetic Algorithms)**
   - ✓ Clustering models
     - ❖ Users are classified into **groups** so that similar users are placed in the same groups
     - ❖ Suggestions should be given to the group instead of the individual, thus **reducing the time** and **overhead of calculations**
   - ✓ Hybrid user models
     - ❖ Generally, a combination of **Collaborative Filtering** with methods such as **Demographic Filtering** or **Content-based Filtering**
     - ❖ Genetic algorithm **chromosomes** can be used as **Demographic Information of users**

2) Neural network solutions (NN)

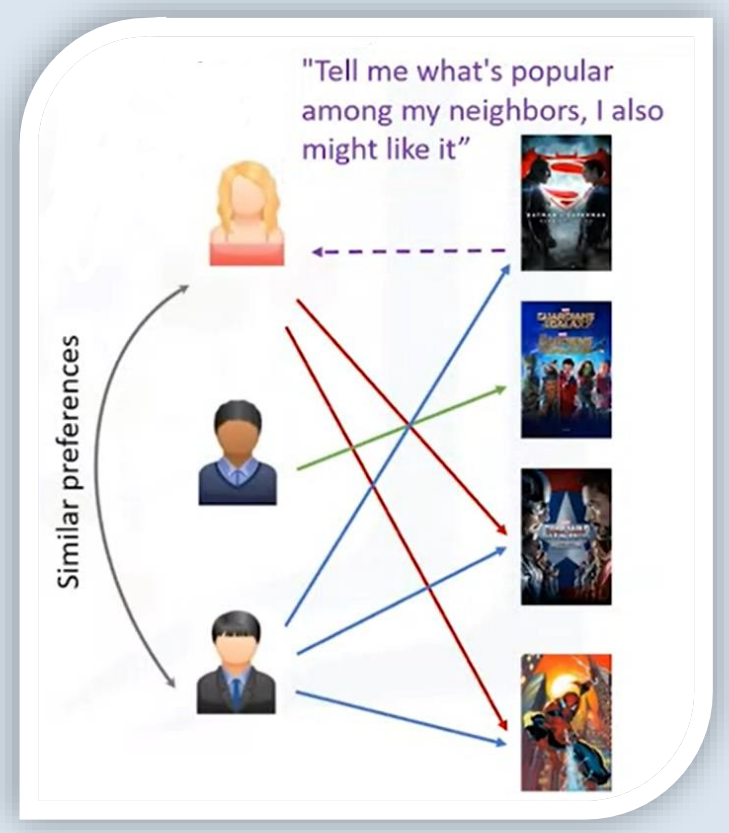
✓ They are formed based on the function and behavior of the **body's neural system**, which provide the power of **learning** based on previous **inputs** and **patterns** for recommender systems.

Two main types of recommender systems

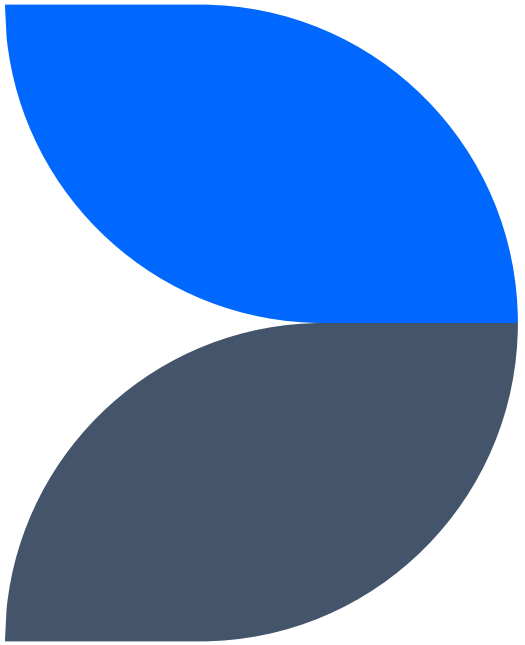
Content-Based



Collaborative Filtering



Approaches of implementing recommender systems

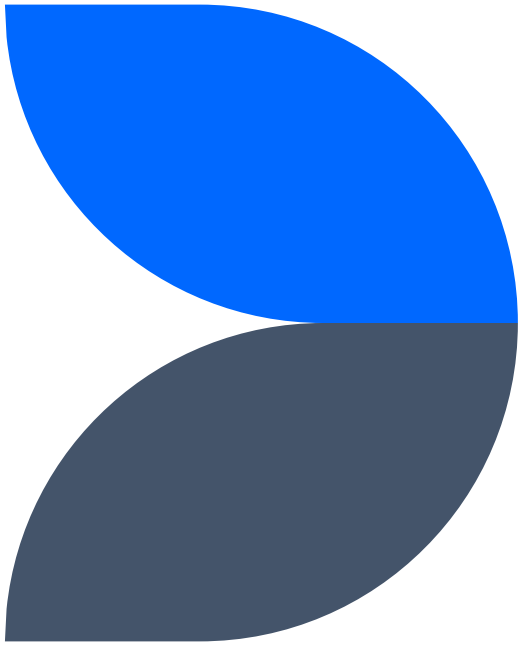


Implementing recommender systems

- ✓ **Memory-based**
  - ❖ Uses the entire user-item dataset to generate a recommendation
  - ❖ Uses statistical techniques to approximate users and items
    e.g., Pearson Correlation, Cosine Similarity, Euclidean Distance, etc.

- ✓ **Model-based**
  - ❖ Develops a model of users in an attempt to learn their preferences
  - ❖ Models can be created using Machine Learning techniques like regression, clustering, classification, etc.

Content-based recommendation systems



Content-based recommendation systems

Items content: Items category, Tag, Genre, etc.

For example: If we have four movies, and if the user likes or rates the first two items, and if Item 3 is similar to Item 1 in terms of their genre, the engine will also recommend Item 3 to the user.

How the engine of content-based recommender systems work?

NETFLIX

User watched & rated 3 movies:

1) Batman v Superman (2/10)
2) Guardians of the Galaxy (10/10)
3) Captain America: Civil War (8/10)

Based on:

- ✓ User's taste
- ✓ The content/feature set items

Task:

Recommend one of the three candidate movies to this user (we want to predict what the user's **possible rating** would be of the three candidate movies if she were to watch them)

Weighing the genres

Input User Rating

2 

10 

8 

One-hot encode

Movies Matrix

Comedy	Adventure	Super-Hero	Sci-Fi
0	1	1	0
1	1	1	1
1	0	1	0

Weighing the genres

Input User Rating * Movies Matrix = **Weighted Genre Matrix**

Comedy	Adventure	Super-Hero	Sci-Fi
0	2	2	0
10	10	10	10
8	0	8	0

Making user profile

Sum comedy weights = 0 + 10 + 8 = 18
Sum adventure weights = 2 + 10 + 0 = 12
Sum super-hero weights = 2 + 10 + 8 = 20
Sum Sci-Fi weights = 0 + 10 + 0 = 10
Total weights = 18 + 12 + 20 + 10 = 60

Normalize the weights:
Comedy: 18 / 60 = 0.3
Adventure: 12 / 60 = 0.2
Super-hero: 20 / 60 = 0.33
Sci-Fi: 10 / 60 = 0.16

User Profile

Comedy	Adventure	Super Hero	Sci-Fi
0.3	0.2	0.33	0.16

Finding a recommendation

User Profile

Comedy	Adventure	Super-Hero	Sci-Fi
0.3	0.2	0.33	0.16

Mul

Weighted Movies Matrix

	Comedy	Adventure	Super-Hero	Sci-Fi
	1	1	0	1
	0	0	1	0
	1	0	1	0

Movies Matrix

Comedy	Adventure	Super-Hero	Sci-Fi
0.3	0.2	0	0.16
0	0	0.33	0
0.3	0	0.33	0

Recommendation Matrix

	Weighted Average
	0.3+0.2+0+0.16 = 0.66
	0+0+0.33+0=0.33
	0.3+0+0.33+0=0.63

Sort & Recommend

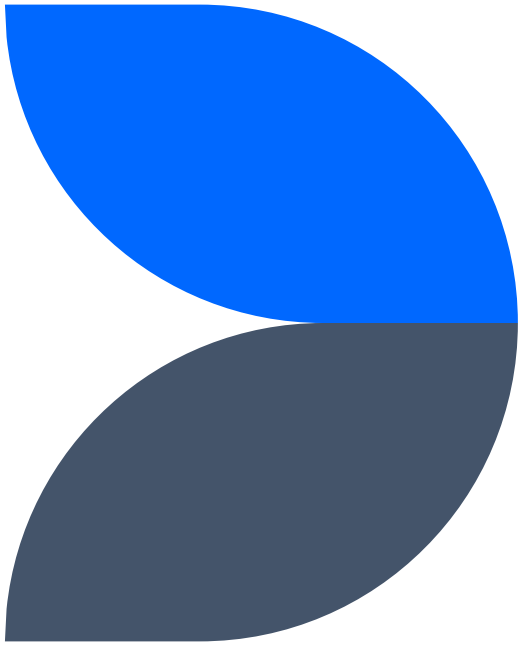
	Weighted Average
	0.66
	0.63
	0.33

Never recommend any movie within other genres

Challenges of content-based filtering

- ✓ The difficulty of extracting data and information about items and users
  - ❖ In some applications(such as music, video weblogs), finding similarities between different items based on their attributes and contents is very complicated and difficult
  - ❖ The need for solutions to extract pages automatically
- ✓ Overspecialization problem
  - ❖ Items that may be liked by the user, but are not similar to the items selected in the past, will never be offered to the user and will remain hidden from her view
- ✓ Inability to get user feedback
  - ❖ In such systems, users usually do not rate the items, which makes it impossible to find out whether the offer given to the user was correct.

Collaborative filtering recommendation systems



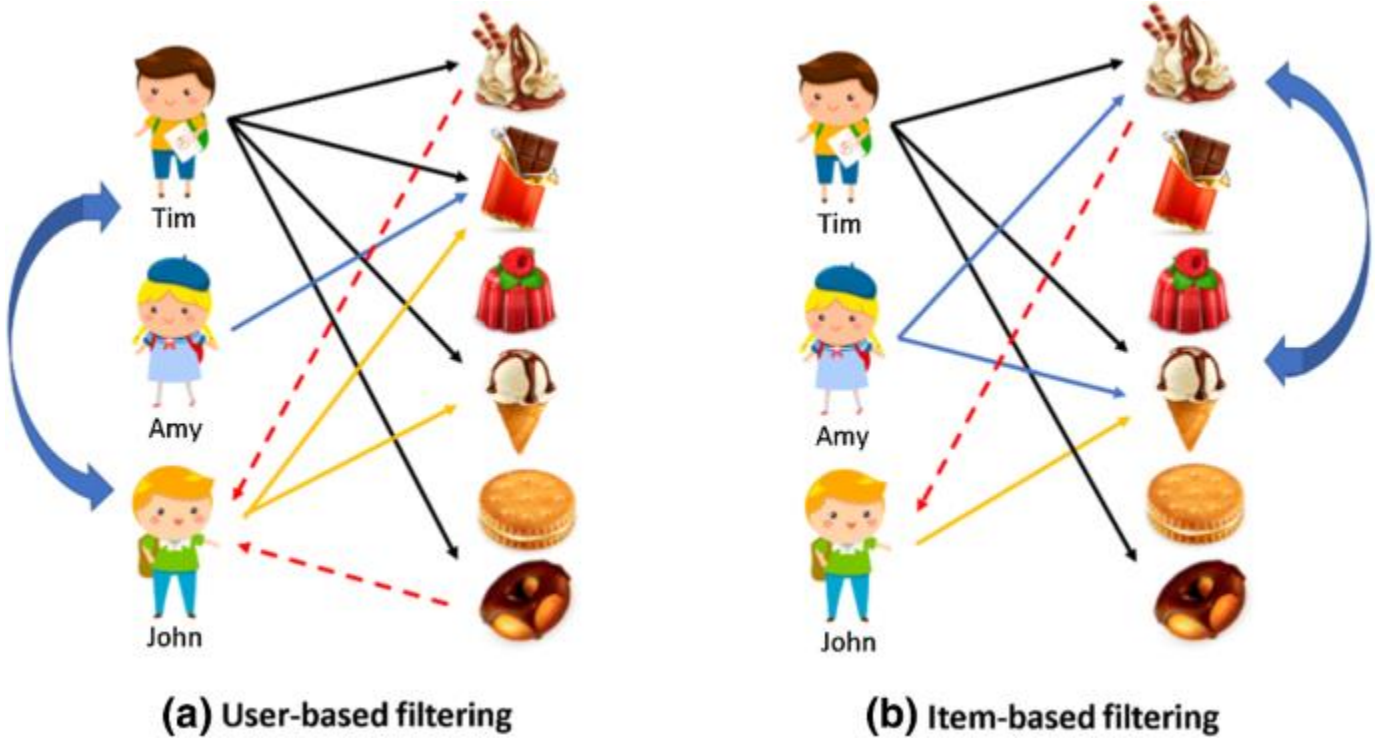
Collaborative filtering recommendation systems

Based on : Relationships exist between products & people's interests

Two approaches:

1) **User-based** collaborative filtering
   - ❖ Based on user's similarities/ user's neighborhood

2) **Item-based** collaborative filtering
   - ❖ Based on item's similarity



(a) User-based filtering

(b) Item-based filtering

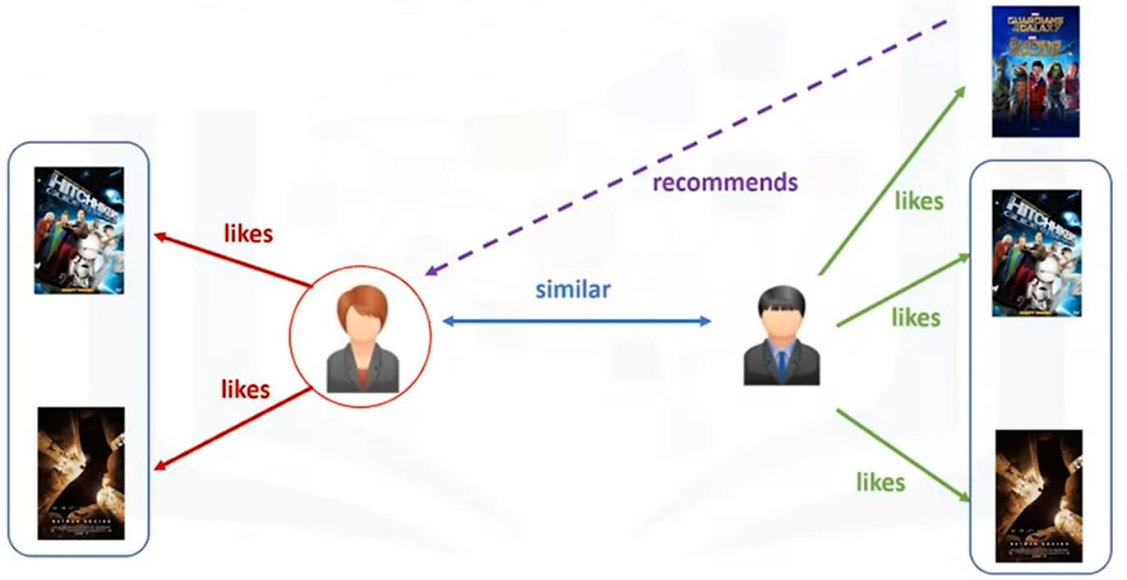
User-based collaborative filtering

Similarity on things like:

- ✓ History
- ✓ Preference
- ✓ Choices

Active user: User for whom the recommendation is aimed

** E.g., movies that similar users have rated highly



User ratings matrix

Find out which of the two movies that our active user hasn't watched should be recommended to her.
These two movies are:

- ✓ Batman v Superman
- ✓ Batman begins




				
9	6	8	4	
2	10	6		8
5	9		10	7
?	10	7	8	?

Active User

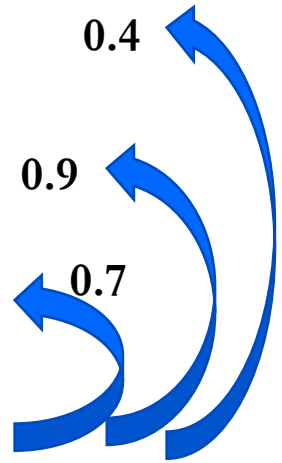
Similarity weights

Several statistical & vectorial techniques (**distance & similarity measurements**)

- ✓ Euclidean Distance
- ✓ Pearson Correlation
- ✓ Cosine Similarity
- ✓ …

Level of similarity between two users:
Use movies that both of users have rated in the past

	9	6	8	4	
	2	10	6		8
	5	9		10	7
	?	10	7	8	?



Rating Matrix

Creating the weighted ratings matrix




9	
2	8
5	7

Ratings Matrix Subset

Similarity Index
0.4
0.9
0.7

Similarity Matrix


3.6	
1.8	7.2
3.5	4.9

Weighted Ratings Matrix

Weighted Ratings Matrix:
Represents the user's neighbors opinion about are two candidate movies for recommendation.

Recommendation matrix


		7.2+4.9= **12.1**
		3.6+1.8+3.5 **=8.9**

Weight Sum
0.9+0.7=1.6
0.4+0.9+0.7=2.0

Similarity Index
0.4
0.9
0.7

Div


Rating & Weight Matrix

8.9/2.0= **4.45**	12.1/1.6= **7.56**

Recommendation Matrix

Sort & Recommend

Similarity
7.56

KNN(K-Nearest-Neighbors) Algorithm

- ✓ The most common algorithm used in the Collaborative filtering method
- ✓ Advantages: Simplicity & Accuracy
- ✓ Disadvantages: Scalability & Sparsity

3 Steps of algorithms that implement based on **user-to-user**:

1) Based on a similarity measurement, a number of **k neighbors** are selected for user a. These neighbors are the ones that are **most similar** to user a.
2) All items in the system are calculated as a quantitative measure to **predict** whether item i will be liked by user a or not. (Using different solutions like average, weighted sum, …)
3) Among all the items, N items that have **more than the predicted value** are suggested to the user.

3 Steps of algorithms that implement based on **item-to-item**:

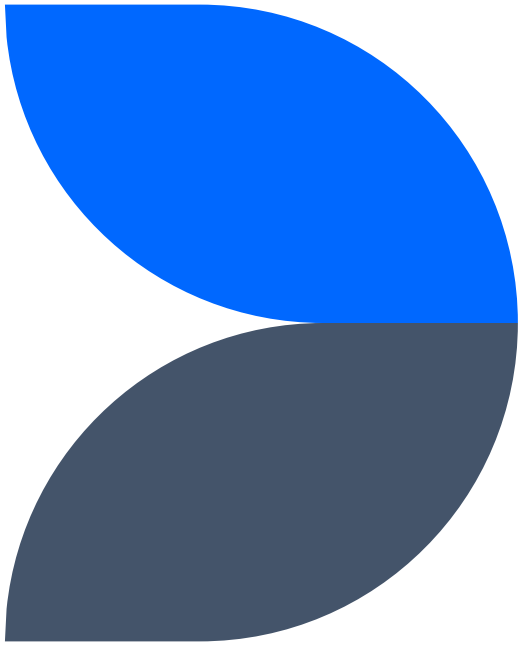
1) Based on a similarity measurement, we determine the number of neighbors q for each item i.
2) If user a has not rated item i so far, we calculate the prediction value based on the ratings this user has given to **neighboring** items i.
3) Based on the values of predictions, we suggest the items that have the highest value of prediction to user a.

Challenges of collaborative filtering

- ✓ Data Sparsity
  - ❖ Users in general rate only a limited number of items
- ✓ Cold Start
  - ❖ Difficulty in recommendation to new users or new items
  - ❖ User profile doesn't exist for users
  - ❖ Items which has not received a rating
- ✓ Scalability
  - ❖ Increase in number of users or items (drop in performance)
  - ❖ Expands amount of data

Use hybrid-based recommendation systems

Evaluation of recommender systems



Evaluation of recommender systems

1) **Online method (A/b Test):**
   - ✓ Ideal for evaluation
   - ✓ User interactions are measured when system recommender suggestions are given to the user
   - ✓ Difficult to implement this way of working (the only way to implement interaction testing is with the system that is currently in production)
   - ✓ Using real customers for tests in recommender systems is much slower than if the data is already available

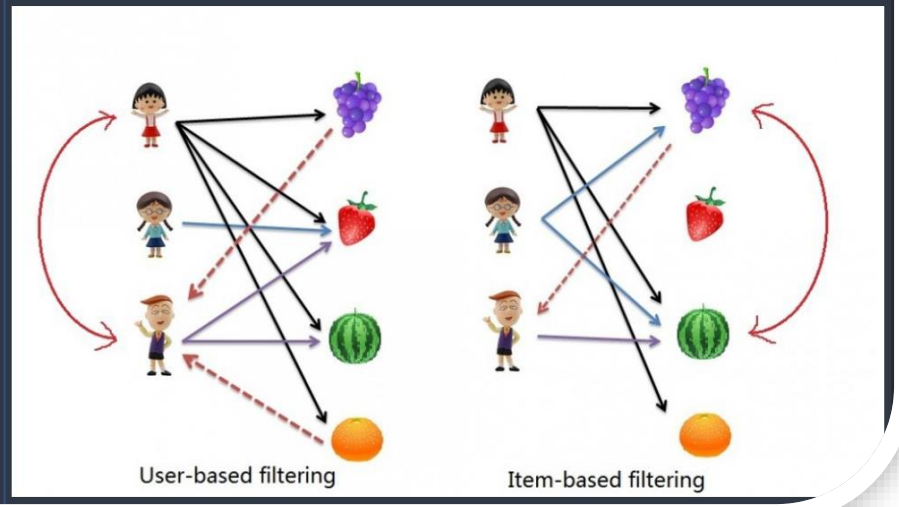
2) **Offline method:**
   - ✓ Ideal for experimental
   - ✓ The user is not directly involved
   - ✓ The system does not need to be deployed to be evaluated
   - ✓ Part of the data will be used to **build** the system(Train) and the other part will be used to **evaluate** it(Validation)

Implementation of the recommender system

Implementation of the recommender systems in Python

Collaborative Filtering

The technique we're going to take a look at is called **Collaborative Filtering**, which is also known as **User-User Filtering**. As hinted by its alternate name, this technique uses other users to recommend items to the input user. It attempts to find users that have similar preferences and opinions as the input and then recommends items that they have liked to the input. There are several methods of finding similar users (Even some making use of Machine Learning), and the one we will be using here is going to be based on the **Pearson Correlation Function**.



Content Based Filtering

Content-based Filtering is a Machine Learning technique that uses similarities in features to make decisions. This technique is often used in recommender systems, which are algorithms designed to advertise or recommend things to users based on knowledge accumulated about the user. This technique attempts to figure out what a users favourite aspects of an item is, and then recommends items that present those aspects. In our case, we're going to try to figure out the input's favorite genres from the movies and ratings given.

Thank you For Your Attention