Recommendation Systems

Swapnali Kurhade

What is Recommendation System?

A recommendation engine filters the data using different algorithms and recommends the most relevant items to users.

NETFLIX	Netflix is a streaming service that allows our members to watch a wide variety of award-winning TV shows, movies, documentaries
YouTube	A video sharing service where users can create their own profile, upload videos, watch, like and comment on other videos.
tinder	Location-based social search mobile app and Web application most often used as a dating service, that allows users to use a swiping motion to like or dislike other users, and allows users to chat if both parties like each other.
amazon	The world's largest online retailer and a prominent cloud services provider.

NETFLIX

- According to a paper written by Netflix executives Carlos A. Gomez-Uribe and Neil Hunt, the video streaming service's AI recommendation system saves the company around $1 billion each year.
- Netflix uses RS personalized diversity to generate Top Ten recommendations for user households, so that it can offer videos that each member of the household may be interested in.
- According to McKinsey, 75 percent of what users watch on Netflix come from product recommendations.

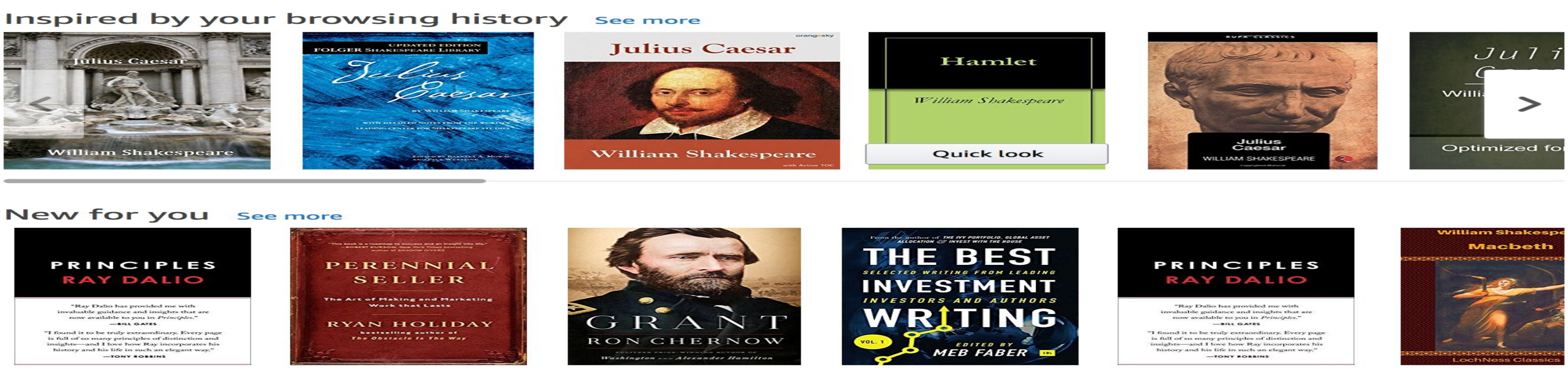
YouTube

- create personalized recommendations so users can quickly and easily find videos that are relevant to their interests.
- Because of the value of keeping users engages, YouTube strives to keep the recommendations updated on a regular basis, to reflect each user's activity on the site and to simultaneously highlight the wide range of available content.

- With 26 million matches per day and more than 20 billion matches made to date, Tinder is the world’s most popular app for meeting new people.

amazon

- Amazon.com uses recommendations as a targeted marketing tool throughout its website.
- When a customer clicks on the “your recommendations” the link leads to another page where recommendations may be filtered even further by subject area, product types, and ratings of previous products and purchases.



Various Areas:

Enhanced experiences

- Recommender systems can also enhance experiences for:
  - **News Websites**
  - **Computer Games**
  - **Knowledge Bases**
  - **Social Media Platforms**
  - **Stock Trading Support Systems**

Advantages

- Advantages of adding a recommender system to your website or software:
  - **Increase in sales thanks to personalized offers.**
  - **Enhanced customer experience.**
  - **More time spent on the platform.**
  - **Customer retention thanks to users feeling understood.**

- A **recent study by Epsilon (2018)** found that **90% of consumers find personalization appealing**. Plus, a further **80% claim they are more likely to do business with a company** when offered personalized experiences.
- The study also found that these **consumers are 10x more likely to become VIP customers**, who make more than 15 purchases per year.

How Recommender Systems Provide Users with Suggestions?

- Using machine learning, recommender systems provide you with suggestions in a few ways:
  - **Collaborative Filtering**
  - **Content-based Filtering**
  - **Hybrid (Combination of Both)**

Content Based Filtering

Content Based Filtering

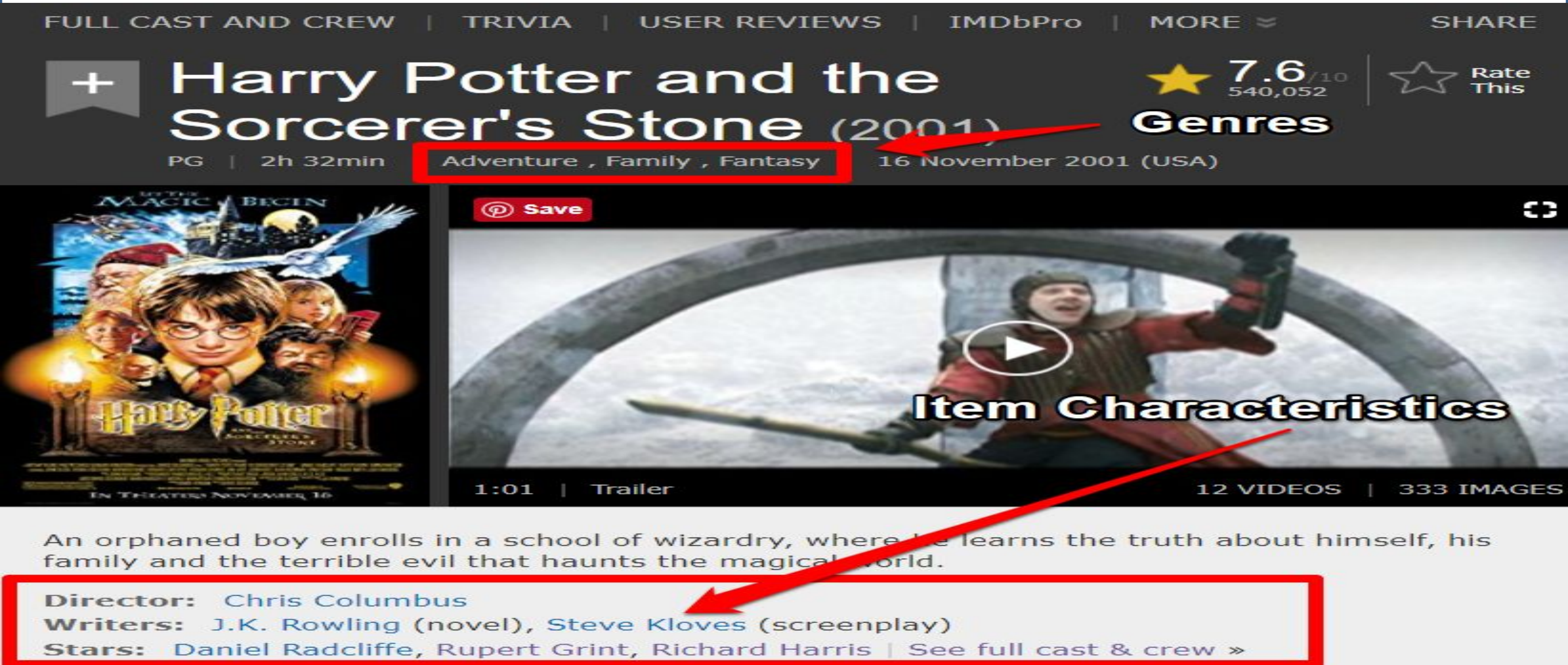
- Content based filtering uses characteristics or properties of an item to serve recommendations. Characteristic information includes:
  - **Characteristics of Items (Keywords and Attributes)**
  - **Characteristics of Users (Profile Information)**

Movie Recommendation System

- Characteristics for the item *Harry Potter and the Sorcerer's Stone* might include:
- **Director Name** – Chris Columbus
- **Genres** – Adventure, Fantasy, Family (IMDB)
- **Stars** – Daniel Radcliffe, Rupert Grint, Emma Watson

Movie Recommendation System

- A content based recommender system can now serve the user:
  - **More Harry Potter Movies**
  - **More Adventure, Family, or Fantasy Movies**
  - **More Chris Columbus Movies**
  - **More Daniel Radcliffe Movies**



How Does it work?

- Save all the information related to each user in a vector form.
- This vector contains the past behavior of the user, i.e. the movies liked/disliked by the user and the ratings given by them.
- This vector is known as the *profile vector*.
- All the information related to movies is stored in another vector called the *item vector*.
- Item vector contains the details of each movie, like genre, cast, director, etc.

content-based filtering

- The content-based filtering algorithm finds the cosine of the angle between the profile vector and item vector, i.e. **cosine similarity**.
- Suppose A is the profile vector and B is the item vector, then the similarity between them can be calculated as:

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

content-based filtering

- Based on the cosine value, which ranges between -1 to 1, the movies are arranged in descending order and one of the two below approaches is used for recommendations:
- **Top-n approach**: where the top n movies are recommended (Here n can be decided by the business)
- **Rating scale approach**: Where a threshold is set and all the movies above that threshold are recommended

Other methods that can be used to calculate the similarity are:

- **Euclidean Distance**: Similar items will lie in close proximity to each other if plotted in n-dimensional space.
- So, we can calculate the distance between items and based on that distance, recommend items to the user.
- The formula for the euclidean distance is given by:

$$\text{Euclidean Distance} = \sqrt{(x_1 - y_1)^2 + \ldots + (x_N - y_N)^2}$$

Pearson's Correlation:

- It tells us how much two items are correlated.
- Higher the correlation, more will be the similarity.
- Pearson's correlation can be calculated using the following formula:

$$sim(u,v) = \frac{\sum(r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum(r_{ui} - \bar{r}_u)^2}\sqrt{\sum(r_{vi} - \bar{r}_v)^2}}$$

Advantages

- The model doesn't need any data about other users, since the recommendations are specific to this user. This makes it easier to scale to a large number of users.
- The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.

Drawbacks:

- A major drawback of this algorithm is that it is limited to recommending items that are of the same type.
- It will never recommend products which the user has not bought or liked in the past.
- So if a user has watched or liked only action movies in the past, the system will recommend only action movies.
- It's a very narrow way of building an engine.

Collaborative Filtering

Collaborative Filtering

- Collaborative filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users.
- It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future.
- for example: A person who wants to see a movie, might ask for recommendations from friends.
- The recommendations of some friends who have similar interests are trusted more than recommendations from others.

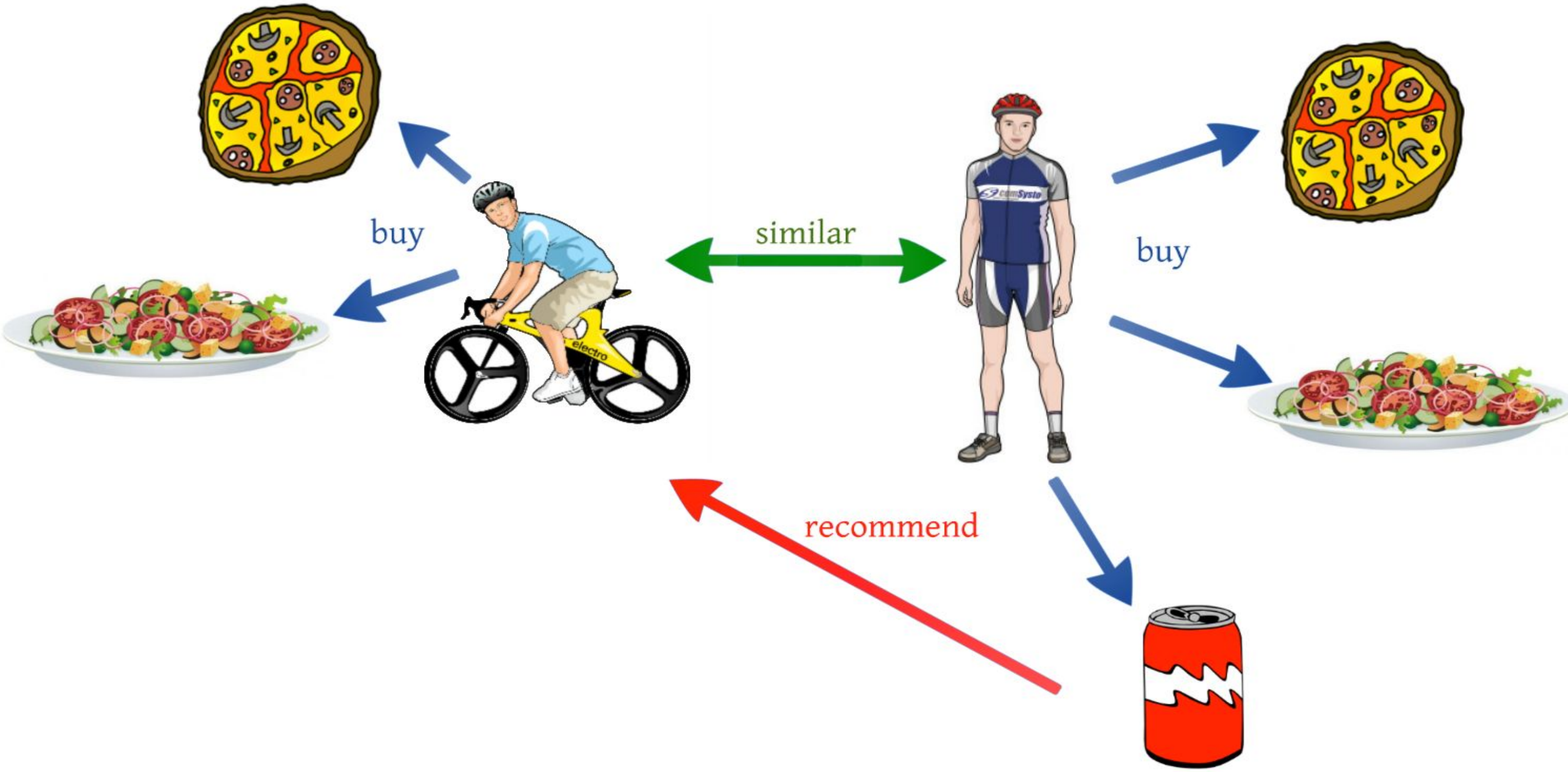
Categories of Collaborative Filtering

There are two categories of CF:

- **User-based**: measure the similarity between target users and other users
- **Item-based**: measure the similarity between the items that target users rates/ interacts with and other items

User Based Collaborative Filtering

User Based Collaborative Filtering



User Based Collaborative Filtering

- You’d have a lot more users than items (ideally anyway).
- You’d expect items to change less frequently than users.
- With more users and less change in the items offered, you can use many more attributes than just purchase history when calculating user similarity.

User Based Collaborative Filtering

- Before the system can make recommendations, it must create a user profile — so it also requires that the user create an account and be logged in (or store session information in the browser via cookies) while viewing a website.

Netflix

- Netflix is an example of quickly building a profile for each customer. Here's the general procedure:
  - Netflix invites its customers to set up queues of the movies they'd like to watch.
  - The chosen movies are analyzed to learn about the customer's tastes in movies.
  - The predictive model recommends more movies for the customer to watch, based on the movies already in the queue.

User Based Collaborative Filtering

Customer	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
A – N1	X	X	X			
B – N1	X	X				
C – N2			X		X	
D – N2			X	X	X	
E – N1		X	X			
F – N1	X	X		X	X	
G – N1	X		X			
H – N3	X					
I – N3						X

For finding similarity distance measures could be used as Euclidean distance, cosine similarity etc

User Based Collaborative Filtering

Customer	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
A – N1	X	X	X			
B – N1	X	X				
C – N2			X		X	
D – N2			X	X	X	
E – N1		X	X			
F – N1	X	X		X	X	
G – N1	X		X			
H – N3	X					
I – N3						X

- There are three user neighborhoods formed: N1, N2, and N3.
- Every user in neighborhoods N1 and N2 has purchased at least 2 items in common with someone else in the same neighborhood.
- N3 are users that have not yet met the criteria and will not receive recommendations until they purchase other items to meet the criteria.

- The system could send marketing ads or make recommendations on the website as follows:
- **Item 3 to Customer B**
- **Item 4 to Customer C**
- **Item 1 to Customer E**
- **Item 3 to Customer F**
- **Item 2 to Customer G**
- **Undetermined item to Customers A and D**

Advantages

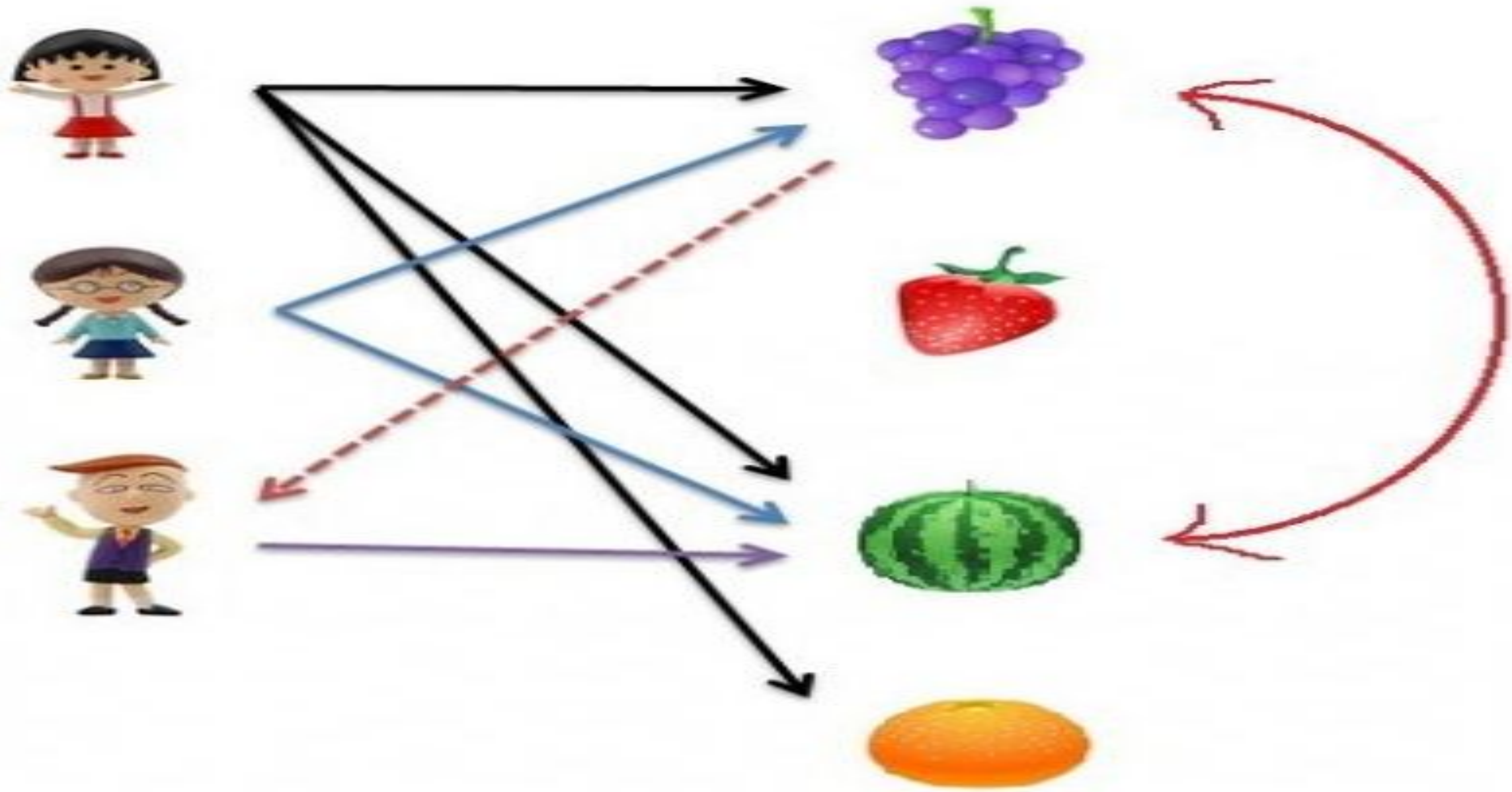
- Easy to implement.
- Context independent.
- Compared to other techniques, such as content-based, it is more accurate.

Drawbacks:

- Sparsity: The percentage of people who rate items is really low.
- Scalability: The more ***K*** neighbors we consider (under a certain threshold), the better my classification should be. Nevertheless, the more users there are in the system, the greater the cost of finding the nearest K neighbors will be.
- Cold-start: New users will have no to little information about them to be compared with other users.
- New item: Just like the last point, new items will lack of ratings to create a solid ranking

Item Based Collaborative Filtering

Item Based Collaborative Filtering



Item-based filtering

Item Based Collaborative Filtering

- Build a product-to-product matrix of similarities by iterating trough all possible pairs
  - *Inefficient because many pairs have no common customers!*
- A better approach for selecting pairs of items for which the similarity can be computed is:
  1. Scan the products, and for all the customers that bought a product, identify the other products bought by those customers
  2. Then compute the similarity only for these pairs

For computing similarity

For each item in product catalog, I_1

For each customer C who purchased I_1

For each item I_2 purchased by customer C

Record that a customer purchased I_1 and I_2

For each item I_2

Compute the similarity between I_1 and I_2

$$sim(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

$$sim(i,j) = \frac{\sum_{u\in U}(R_{u,i}-\bar{R}_i)(R_{u,j}-\bar{R}_j)}{\sqrt{\sum_{u\in U}(R_{u,i}-\bar{R}_i)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\bar{R}_j)^2}}$$

Item Based Collaborative Filtering Example

Suppose we have movie ratings given by different users in a table format as shown below:

User	Movie	Rating
Amy	Pulp Fiction	4
Amy	The GodFather	5
Calvin	Pulp Fiction	5
Robert	Pulp Fiction	3
Calvin	Forrest Gump	2
Robert	Forrest Gump	3
Robert	The GodFather	1
David	Forrest Gump	2
Bradley	Pulp Fiction	1
David	The GodFather	2
Bradley	Forrest Gump	3

Item Based Collaborative Filtering Example

Step 1: We create a sparse matrix where we write user-item ratings in a matrix form

	Amy	Calvin	Robert	David	Bradley
Pulp Fiction	4	5	3	?	1
Forrest Gump	?	2	3	2	3
The GodFather	5	?	1	2	?

- In this matrix user, Amy has already rated and watched movies *Pulp Fiction* and *The GodFather* but hasn't watched the movie, *Forrest Gump*.
- We will be using the above matrix for our example and will try to create an item-item similarity matrix using Cosine Similarity method to determine how similar the movies are to each other.

Item Based Collaborative Filtering Example

Step 2: To calculate the similarity between the movie *Pulp Fiction* (P) and *Forrest Gump* (F), we will first find all the users who have rated both the movies.

In our case, Calvin (C), Robert (R) and Bradley (B) have rated the movies.

We now create two vectors:

v1=5C+3R+1B
v2=2C+3R+3B

$$\cos(v1,v2)=\frac{5*2+3*3+1*3}{\sqrt{5^2+3^2+1}+\sqrt{2^2+3^2+3^2}}$$

=0.79

Similarly, we can calculate the cosine similarity of all the movies and our final similarity matrix will be:

	Pulp Fiction	Forrest Gump	The GodFather
Pulp Fiction	1	0.792	0.902
Forrest Gump	0.792	1	0.8
The GodFather	0.902	0.8	1

Step 3: Now we can predict and fill the ratings for a user for the items he hasn't rated yet. So to calculate the rating of user **Amy** for the movie *Forrest Gump,* we will use the calculated similarity matrix along with the already rated movie by the user.

Therefore, the rating would be:

$$(4*0.792 + 5*0.8) / (0.792+ 0.8) = 4.5$$

Hence, our final matrix would be:

	Pulp Fiction	ForrestGump	The GodFather
Amy	4	4.5	5
Calvin	5	2	4.01
Robert	3	3	1
David	2	2	2
Bradley	1	3	1.94

We will consider the following sample data of preference of four users for three items:

ID	user	item	rating
241	u1	m1	2
222	u1	m3	3
276	u2	m1	5
273	u2	m2	2
200	u3	m1	3
229	u3	m2	3
231	u3	m3	1
239	u4	m2	2
286	u4	m3	2

Step 1: Write the user-item ratings data in a matrix form. The above table gets rewritten as follows:

	m1	m2	m3
u1	2	**?**	3
u2	5	2	**?**
u3	3	3	1
u4	**?**	2	2

Thus, the two item-vectors would be,

```
v1 = 5 u2 + 3 u3
v2 = 3 u2 + 3 u3
```

The cosine similarity between the two vectors, v1 and v2, would then be:

```
cos(v1,v2) = (5*3 + 3*3)/sqrt[(25 + 9)*(9+9)] = 0.76
```

Similarly, to calculate similarity between m1 and m3, we consider only users u1 and u3 who have rated both these items. The two item vectors, v1 for item m1 and v3 for item m3, in the user-space would be as follows:

```
v1 = 2 u1 + 3 u3
v3 = 3 u1 + 1 u3
```

The cosine similarity measure between v1 and v3 is:

```
cos(v1,v3) = (2*3 + 3*1)/sqrt[(4 + 9)*(9+1)] = 0.78
```

We can similarly calculate similarity between items m2 and m3 using ratings given to both by users u3 and u4. The two item-vectors v3 and v4 would be:

```
v2 = 3 u3 + 2 u4
v3 = 1 u3 + 2 u4
```

And cosine similarity between them is:

```
cos(v2,v3) = (3*1 + 2*2)/sqrt[(9 + 4)*(1 + 4)] = 0.86
```

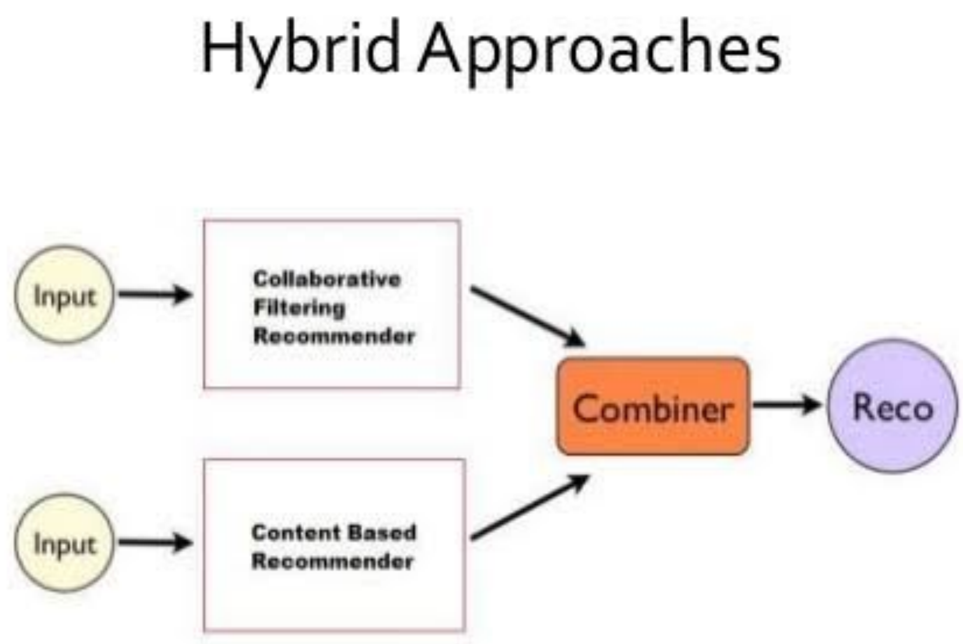
We now have the complete item-to-item similarity matrix as follows:

	m1	m2	m3
m1	1	0.76	0.78
m2	0.76	1	0.86
m3	0.78	0.86	1

Step 3: For each user, we next predict his ratings for items that he had not rated. We will calculate rating for user u1 in the case of item m2 (target item). To calculate this we weigh the just-calculated similarity-measure between the target item and other items that user has already rated. The weighing factor is the ratings given by the user to items already rated by him. We further scale this weighted sum with the sum of similarity-measures so that the calculated rating remains within a predefined limits. Thus, the predicted rating for item m2 for user u1 would be calculated using similarity measures between (m2,m1) and (m2,m3) weighted by the respective ratings for m1 and m3:

```
rating = (2 * 0.76 + 3 * 0.86)/(0.76+0.86) = 2.53
```

Hybrid CF


Hybrid Approaches
Input
Collaborative
Filtering
Recommender
Input
Content Based
Recommender
Combiner
Reco

The next generation of recommendation systems may include:

- More relevant recommendations: By digging deeper into customers' interests and preferences, recommendation systems will be able to present users with more-relevant, predictive recommendations.
- Incorporate item profitability: Instead of having recommendation based solely on a customer's browsing history and past purchases, this would allow businesses to control how much a profit-based recommendation differs from the traditional recommendation and to set a balance so that customer trust would not be compromised.
- Increase product reach: Each retailer has an individual catalogue of products, improved recommendation systems would be able to access a broader range of merchandise in order to include new or niche items in shoppers' recommendations.
- Reach shoppers through multiple channels: Next generation recommendation systems should be able to reach customers across a range of channels including email, social media, on an off-site shopping widgets, mobile apps, and the retail customer service centres.