**Collaborating filtering recommeded system**

**Why use recommender system?**

A recommender system helps user to quickly find the most relevant options when searching for items in e-commerce sites such as Amazon or contents in entertainment providers such as YouTube or Netflix. Having a good recommender system in place can significantly affect the success of these platforms by improving customer satisfaction and engagement.

Imagine trying to decide which movie to watch without being shown a list of recommended movies or shows, it would be a much less pleasant experience. Some users might even just opt to not watch anything at all due to the lack of idea. The fact is, sometimes the users don’t even know what exactly are they looking for until it pops up as a recommended items. On top of that, great product recommendations can further encourage impulse buying, which in turns will increase sales and revenue.

**Collaborative Filtering**

Recommender system can be either personalized or non-personalized. Non-personalized system can be simpler but personalized system tends to work better as it caters to the needs of each individual user. Collaborative filtering is a common method of personalized recommender system which filters information such as interactions data from other similar users. Since it works by predicting user ratings, it is considered as performing regression task. There are two general types of collaborative filtering:

*(i). User-based:* which measures the similarity between target users and other users.

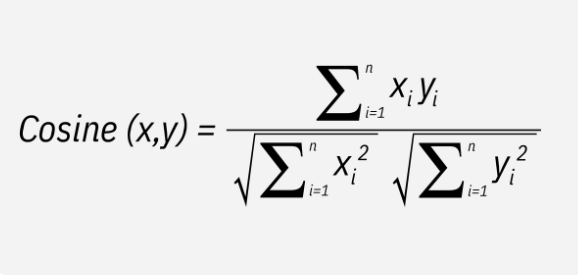
(ii). *Item-based:* which measures the similarity between the items that target users rate or interact with and other items.

**How collaborative filtering works!**

Collaborative filtering uses a matrix to map user behavior for each item in its system. The system then draws values from this matrix to plot as data points in a vector space. Various metrics then measure the distance between points as a means of calculating user-user and item-item similarity.

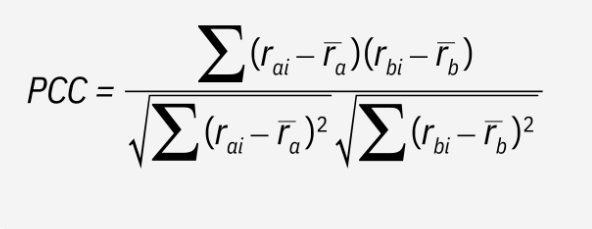
How does a collaborative recommendation algorithm determine similarity between various users? As mentioned, proximity in vector space is a primary method. But the specific metrics used to determine that proximity may vary. Two such metrics are cosine similarity and Pearson correlation coefficient.

1. **Cosine similarity**

Cosine similarity signifies the measurement of the angle between two vectors. Compared vectors comprise a subset of ratings for given user or item. The cosine similarity score can be any value between -1 and 1. The higher the cosine score, the more alike two items are considered. Some sources recommend this metric for high-dimensional feature spaces. In collaborative filtering, vector points are pulled directly from the user-item matrix. Cosine similarity is represented by this formula, where x and y signify two vectors in vector space:

1. **Pearson correlation coefficient (PCC)**

PCC helps measure similarity between items or users by computing the correlation between two users’ or items’ respective ratings. PCC ranges between -1 and 1, which signify negative to identical correlation. Unlike cosine similarity, PCC uses all the ratings for a given user or item. For example, if calculating PCC between two users, we use this formula, in which a and b are different users, and r*ai* and r*bi* are that user's rating for item *i*:

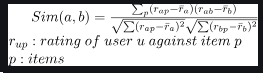


**(i)User-Based Collaborative Filtering**

User-Based Collaborative Filtering is a technique used to predict the items that a user might like on the basis of ratings given to that item by other users who have similar taste with that of the target user. Many websites use collaborative filtering for building their recommendation system.

**Steps for User-Based Collaborative Filtering:**

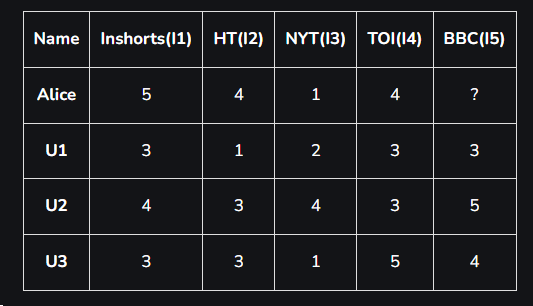
***Step 1:*** *Finding the similarity of users to the target user U.*Similarity for any two users ‘a’ and ‘b’ can be calculated from the given formula,



***Step 2:*** *Prediction of missing rating of an item:*Now, the target user might be very similar to some users and may not be much similar to others. Hence, the ratings given to a particular item by the more similar users should be given more weightage than those given by less similar users and so on. This problem can be solved by using a weighted average approach. In this approach, you multiply the rating of each user with a similarity factor calculated using the above mention formula. The missing rating can be calculated as,



**Example:** Consider a matrix that shows four users Alice, U1, U2 and U3 rating on different news apps. The rating range is from 1 to 5 on the basis of users’ likability of the news app. The ‘?’ indicates that the user has not rated the app.



**Step 1:** Calculating the similarity between Alice and all the other users At first we calculate the averages of the ratings of all the user excluding I5 as it is not rated by,

Alice:

Average rating = (5 + 4 + 1 + 4) / 4 = 14 / 4 = **3.5**

U1:

Average rating = (3 + 1 + 2 + 3 ) / 4 = 9 / 4 = **2.25**

U2:

Average rating = (4 + 3 + 4 + 3 ) / 4 = 14 / 4 = **3.5**

U3:

Average rating = (3 + 3 + 1 + 5) / 4 = 12 / 4 = **3.0**

**Step 2:** Calculate Mean-Centered Ratings.

Alice:

Inshorts: 5−3.5=1.5

HT: 4−3.5=0.5

NYT: 1−3.5=−2.5

TOI: 4−3.5=0.5

U1:

Inshorts: 3−2.25=0.75

HT: 1−2.25=−1.25

NYT: 2−2.25=−0.25

TOI: 3−2.25=0.75

U2:

Inshorts: 4−3.5=0.5

HT: 3−3.5=−0.5

NYT: 4−3.5=0.5

TOI: 3−3.5=−0.5

U3:

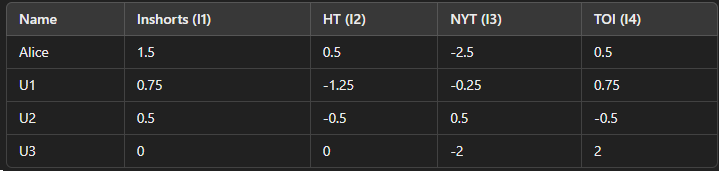
Inshorts: 3−3.0=0

HT: 3−3.0=0

NYT: 1−3.0=−2

TOI: 5−3.0=2

**Mean-Centered Ratings Matrix**



**Step 3: Calculate Pearson Correlation Coefficient**

**By using the pcc formulae above:**

*Pearson Correlation between Alice and U1*

r=0.30

​

*Pearson Correlation between Alice and U2*

*r=-0.33*

*Pearson Correlation between Alice and U3*

*r=0.71*

These values indicate how similar Alice's ratings are to the ratings of each other user, where higher positive values indicate more similarity.

**2. Item-to-Item Based Collaborative Filtering**

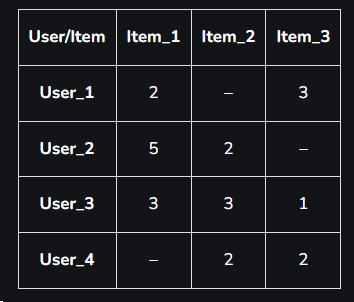
Collaborative Filtering is a technique or a method to predict a user’s taste and find the items that a user might prefer on the basis of information collected from various other users having similar tastes or preferences. It takes into consideration the basic fact that if person X and person Y have a certain reaction for some items then they might have the same opinion for other items too.

 It was first invented and used by Amazon in 1998. Rather than matching the user to similar customers, item-to-item collaborative filtering matches each of the user’s purchased and rated items to similar items, then combines those similar items into a recommendation list. Now, let us discuss how it works.

Item to Item Similarity: The very first step is to build the model by finding similarity between all the item pairs. The similarity between item pairs can be found in different ways. One of the most common methods is to use cosine similarity which I will be using in the example bellow.

**Prediction Computation:** The second stage involves executing a recommendation system. It uses the items (already rated by the user) that are most similar to the missing item to generate rating. We hence try to generate predictions based on the ratings of similar products. We compute this using a formula which computes rating for a particular item using weighted sum of the ratings of the other similar products.

**Example:**   
Let us consider one example. Given below is a set table that contains some items and the user who have rated those items. The rating is explicit and is on a scale of 1 to 5. Each entry in the table denotes the rating given by a *ith* User to a *jth* Item. In most cases majority of cells are empty as a user rates only for few items. Here, we have taken 4 users and 3 items. We need to find the missing ratings for the respective user.



**Step 1: Finding similarities of all the item pairs.**

From the item pairs. For example in this example the item pairs are (Item\_1, Item\_2), (Item\_1, Item\_3), and (Item\_2, Item\_3). Select each item to pair one by one. After this, we find all the users who have rated for both the items in the item pair. Form a vector for each item and calculate the similarity between the two items using the cosine formula stated above.

Sim(Item1, Item2)

In the table, we can see only User\_2 and User\_3 have rated for both items 1 and 2.

Thus, let I1 be vector for Item\_1 and I2 be for Item\_2. Then,

I1 = 5U2 + 3U3 and,

I2 = 2U2 + 3U3



Sim(Item2, Item3)

In the table we can see only User\_3 and User\_4 have rated for both the items 1 and 2.

Thus, let I2 be vector for Item\_2 and I3 be for Item\_3. Then,

I2 = 3U3 + 2U4 and,

I3 = 1U3 + 2U4



Sim(Item1, Item3)

In the table we can see only User\_1 and User\_3 have rated for both the items 1 and 2.

Thus, let I1 be vector for Item\_1 and I3 be for Item\_3. Then,

I1 = 2U1 + 3U3 and,

I3 = 3U1 + 1U3



**Step 2: Generating the missing ratings in the table**

Now, in this step we calculate the ratings that are missing in the table.

*Rating of Item\_2 for User\_1*



*Rating of Item\_3 for User\_2*



*Rating of Item\_1 for User\_4*



# **Conclusion & Recommendations**

Collaborative filtering is a method to build a recommender system that utilizes data from other similar users or items to predict how users will rate items that they have not purchased or viewed yet. This rating prediction will then be used to generate a list of possible top items to be recommended. This is an effective strategy to increase user engagement and sales as the recommendations are highly personalized to fit the preference of each individual user.