Data Mining for Business Analytics
MSBA 511

Collaborative Filtering

What is Collaborative Filtering?

Collaborative Filtering is a technique used in data mining and machine learning to make predictions or recommendations by leveraging the preferences, behaviors, or interactions of a group of users. It operates on the assumption that individuals who have shown similar preferences in the past are likely to share preferences in the future.

Real-world application are **recommendation systems**:

- e-Commerce platforms
- Streaming services
- Social networks

User Based

Item Based

What's the Difference?

The difference between Association Rules and Collaborative Filtering boils down to the purpose and application.

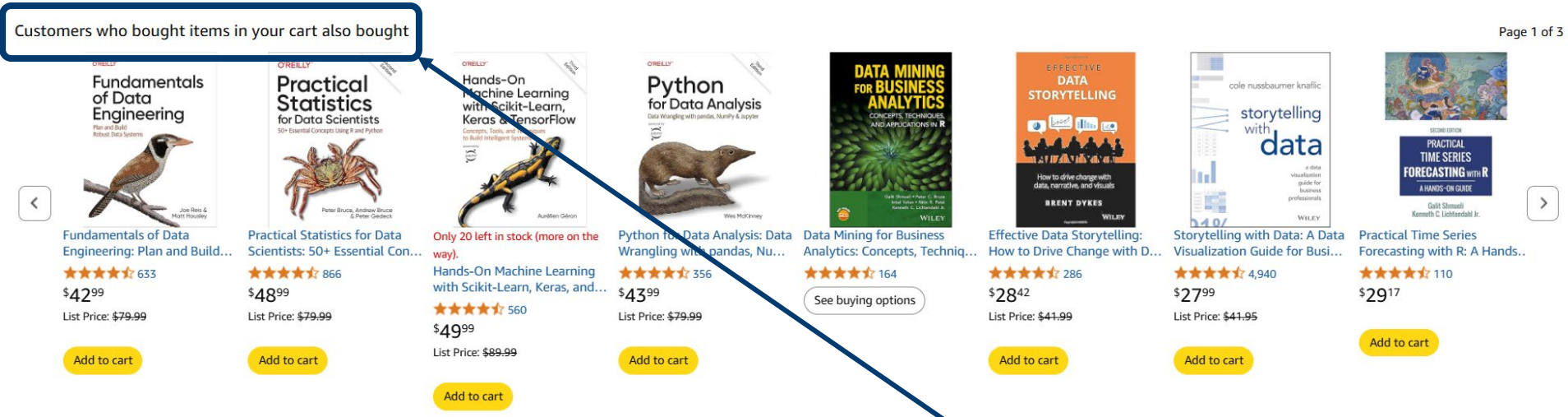
Association Rules Mining: Focuses on discovering relationships or patterns between items in **transactional** data.

Collaborative Filtering: Aims to provide **personalized** recommendations by leveraging **user** interactions and similarities.

Apple TV+ is attempting to predict my user preferences and help me discover new content.

User-Based "Collaboration"

Generate personalized recommendations for a user by finding users with similar preferences (neighbors) and recommend items that the user has not yet purchased that are most preferred by the user's neighbors.

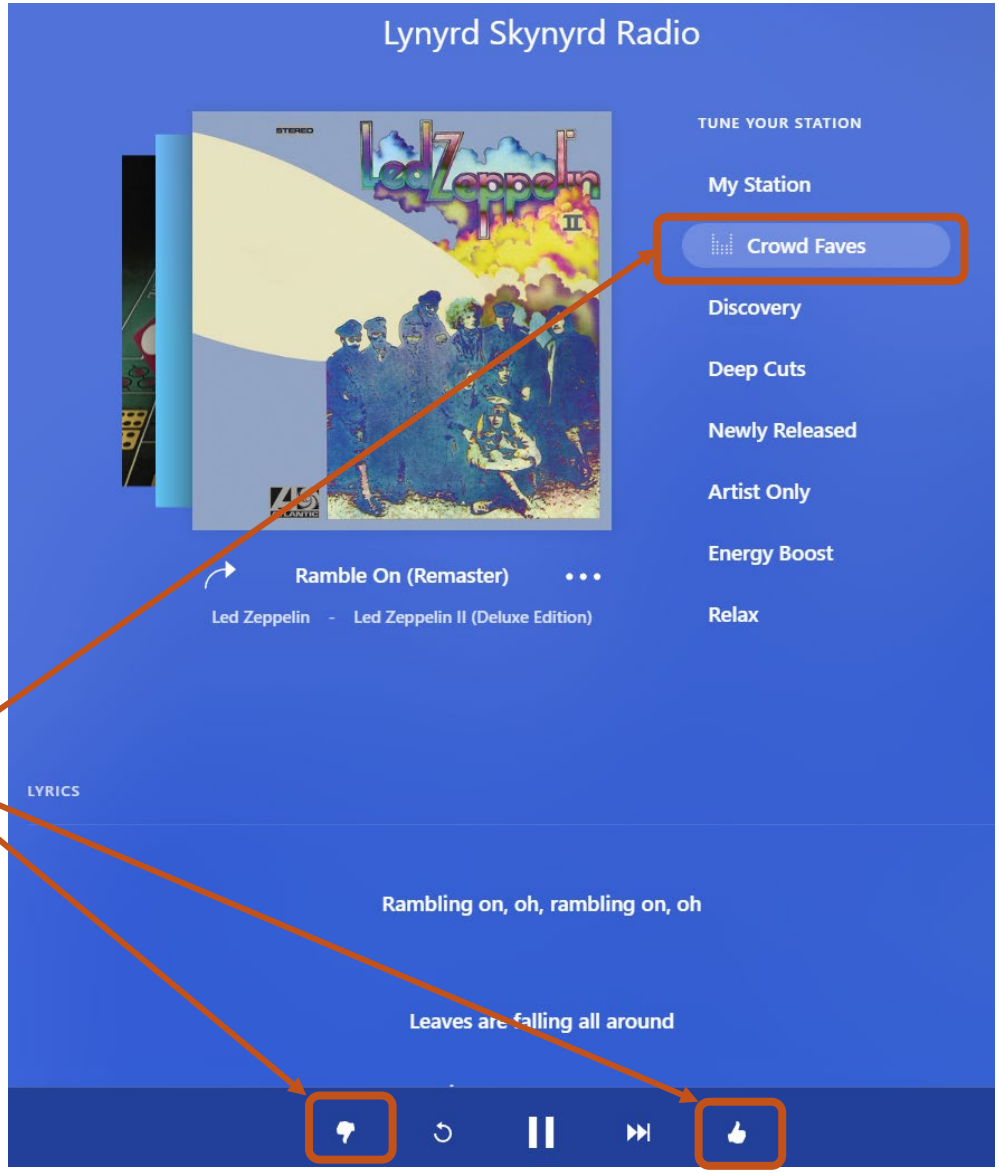


Amazon letting me know what other customers bought after adding Data Mining for Business Analytics to my cart.

Why Personalized Recommendations?

Help people discover new content based on user feedback and behaviors from other users with similar tastes.

Pandora recommends songs in your station based on listening behavior of other users that have similar tastes and even gives you quick access to all songs with the most thumbs up from all other listeners on that station.



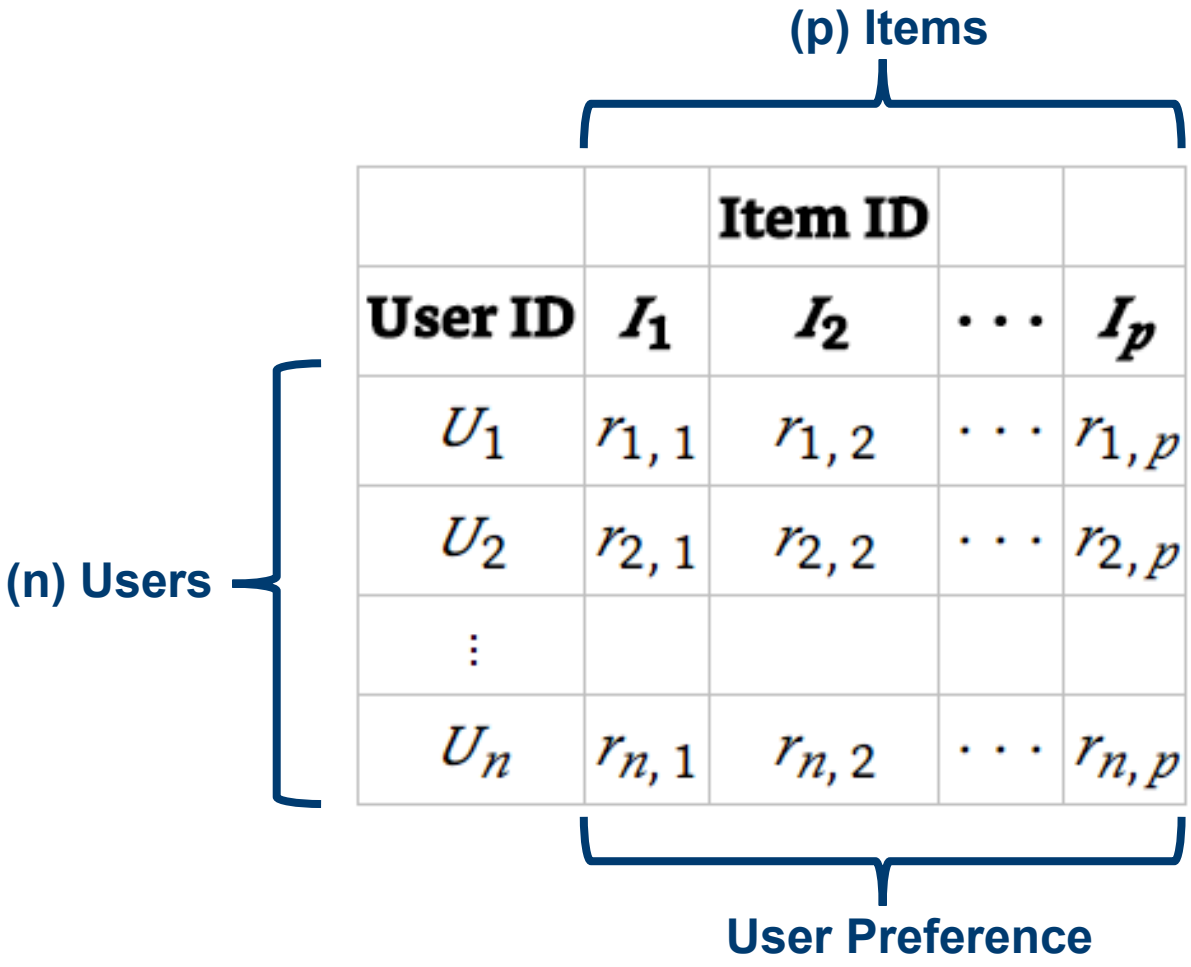
Why Personalized Recommendations?

According to multiple sources, including McKinsey & Company, Amazon generates 35% of its revenue from its recommendation engine, which heavily relies on user-generated data through collaborative filtering.

Source: [How retailers can keep up with consumers]

How is the Data Formatted?

User preference for an item can be either a numerical rating or a binary event such as purchase, "like", or a click.



Measuring Similarity

A popular proximity measure between two users is the Pearson **correlation** measure:

$$Corr(U_1, U_2) = \frac{\sum(r_{1,i} - \bar{r}_1)(r_{2,i} - \bar{r}_2)}{\sqrt{\sum(r_{1,i} - \bar{r}_1)^2}\sqrt{\sum(r_{2,i} - \bar{r}_2)^2}}$$

Ranges from -1 (perfect negative) to 1 (perfect positive)

Another popular measure is a variant of the Pearson correlation called the **cosine similarity** that does not adjust for user rating styles (e.g., always rates high vs. low) by subtracting the mean.

Ranges from 0 (no similarity) to 1 (perfect match)

If you are working with binary matrix instead of ratings, cosine similarity must be calculated over all items that either user purchased and not just overlapping items between users.

Netflix Prize Example

In 2006, Netflix, announced a $1 million contest for the purpose of improving its recommendation system. Below is a tiny sample of records from the contest data:

					Movie ID				
Customer ID	**1**	**5**	**8**	**17**	**18**	**28**	**30**	**44**	**48**
30878	4	1			3	3	4	5	
124105	4								
822109	5								
823519	3		1	4		4	5		
885013	4	5							
893988	3						4	4	
1248029	3					2	4		3
1503895	4								
1842128	4						3		
2238063	3								

Calculate Mean

$$\bar{r}_{30878} = \frac{(4+1+3+3+4+5)}{6} = 3.33$$

$$\bar{r}_{823519} = \frac{(3+1+4+4+5)}{5} = 3.40$$

The average is computed over all the movies that a customer has rated.

Netflix Prize Example

The calculation for correlation involve departures from the average, but only for the items that they co-rated.

					Movie ID				
Customer ID	**1**	**5**	**8**	**17**	**18**	**28**	**30**	**44**	**48**
30878	4	1			3	3	4	5	
124105	4								
822109	5								
823519	3		1	4		4	5		
885013	4	5							
893988	3						4	4	
1248029	3					2	4		3
1503895	4								
1842128	4						3		
2238063	3								

Netflix Prize Example

Calculate correlation and cosine similarity for both users.

Calculate Corr $Corr(U_{30878}, U_{823519})$

$$= \frac{(4-3.33)(3-3.4)+(3-3.33)(4-3.4)+(4-3.33)(5-3.4)}{\sqrt{(4-3.33)^2+(3-3.33)^2+(4-3.33)^2}\sqrt{(3-3.4)^2+(4-3.4)^2+(5-3.4)^2}}$$

$$= \frac{0.606}{1.761} = 0.3441$$

Calculate Cos Sim $Cos\ Sim(U_{30878}, U_{823519})$

$$= \frac{(4\ x\ 3)+(3\ x\ 4)+(4\ x\ 5)}{\sqrt{4^2+3^2+4^2}\sqrt{3^2+4^2+5^2}}$$

$$= \frac{44}{45.277} = 0.9718$$

Collaborative filtering suffers from cold start as it cannot be used to create recommendations for new users or new items.

Making Recommendations

For a user of interest, identify k-nearest users based on specified similarity measure and between all the **other** items they rated/purchased.

We make recommendations for the **best** items to the user:

- Most purchased
- Highest rated
- Most rated

Clustering Alternative

Finding k-nearest neighbors can be computationally expensive when dealing with many users. To address this, clustering techniques can be used to group users into similar, homogeneous clusters based on their preferences. Once clustered, similarity is measured only within each cluster, significantly reducing the computational load.

Advantage:

- Move large computations offline
- Faster and cheaper

Disadvantage:

- Accuracy in recommendations

Item-Based Alternative

When the number of users is much larger than the number of items, it is computationally cheaper to find similar items than similar users.

Find the items that were co-rated or co-purchased **by k-nearest neighbor user** or **any user** with the item of interest and recommend the most popular items among the similar items.

Netflix Prize Example Revisited

Calculate correlation for Movie 1 with a $\bar{r}_1 = 3.7$ and movie 5 with a $\bar{r}_5 = 3$.

					Movie ID				
Customer ID	**1**	**5**	**8**	**17**	**18**	**28**	**30**	**44**	**48**
30878	4	1			3	3	4	5	
124105	4								
822109	5								
823519	3		1	4		4	5		
885013	4	5							
893988	3						4	4	
1248029	3					2	4		3
1503895	4								
1842128	4						3		
2238063	3								

$$Corr(I_1, I_2) = \frac{\sum(r_{1,i} - \bar{r}_1)(r_{2,i} - \bar{r}_2)}{\sqrt{\sum(r_{1,i} - \bar{r}_1)^2}\sqrt{\sum(r_{2,i} - \bar{r}_2)^2}}$$

$$= \frac{(4 - 3.7)(1 - 3) + (4 - 3.7)(5 - 3)}{\sqrt{(4 - 3.7)^2 + (4 - 3.7)^2}\sqrt{(1 - 3)^2 + (5 - 3)^2}}$$

$$= \frac{0}{1.2} = \mathbf{0}$$

There is no relationship between these 2 movies because of the two opposite ratings of movie 5.

Netflix Prize Example Revisited

Calculate cosine for movie 1 and movie 5.

					Movie ID				
Customer ID	**1**	**5**	**8**	**17**	**18**	**28**	**30**	**44**	**48**
30878	4	1			3	3	4	5	
124105	4								
822109	5								
823519	3		1	4		4	5		
885013	4	5							
893988	3						4	4	
1248029	3					2	4		3
1503895	4								
1842128	4						3		
2238063	3								

$$Cos(I_1, I_2) = \frac{\sum(r_{1,i})(r_{2,i})}{\sqrt{\sum(r_{1,i})^2}\sqrt{\sum(r_{2,i})^2}}$$

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

$$= \frac{24}{28.84} = .83205$$

Let's Practice: Item-Based Correlation

Calculate the correlation and cosine between movies 28 and 30.

					Movie ID				
Customer ID	**1**	**5**	**8**	**17**	**18**	**28**	**30**	**44**	**48**
30878	4	1			3	3	4	5	
124105	4								
822109	5								
823519	3		1	4		4	5		
885013	4	5							
893988	3						4	4	
1248029	3					2	4		3
1503895	4								
1842128	4						3		
2238063	3								

$$Corr(I_1, I_2) = \frac{\sum(r_{1,i} - \bar{r}_1)(r_{2,i} - \bar{r}_2)}{\sqrt{\sum(r_{1,i} - \bar{r}_1)^2}\sqrt{\sum(r_{2,i} - \bar{r}_2)^2}}$$

$$\boldsymbol{Corr(28, 30) = .7071}$$

$$Cos(I_1, I_2) = \frac{\sum(r_{1,i})(r_{2,i})}{\sqrt{\sum(r_{1,i})^2}\sqrt{\sum(r_{2,i})^2}}$$

$$\boldsymbol{Cos(28, 30) = .9838}$$

Summary

User-Based: Recommends items by identifying users with similar preferences. This approach requires real-time computation of user-user similarities, which can be slow, especially for new users with little data.

Item-Based: Finds and recommends items like those a user has interacted with. Since item-item relationships can be precomputed, this method is generally more efficient and scalable.

Both approaches rely on similarity measures (e.g., cosine similarity, Pearson correlation) and suffer from challenges like data sparsity and the cold-start problem.

Collaborative filtering is widely applied in e-commerce, streaming services, and social networks to make personalized recommendations.