1. **Beginner: If you are a beginner I would recommend trying to a) Build/Find a Recommendation Dataset, b) Creating a few simple models using both collaborative filtering and content-based methods, c) Trying to improve this as much as possible by reading new papers and techniques.**

**Solution:** Recommendation datasets are collections of data used to develop Recommendation systems like Movie/music/books/social-media content. These are based on personalized choices. These datasets have specific features:

1. **User data**(age, preferences, demographics etc.)

For eg. A content creator can address people in the age-group of 18-22, is focused on sports and can live in the US.

1. **Items:** (what is that, he provides):

For eg. He provides info, highlights of everyday Basketball games

1. **Interactions:** (How he interacts with his fan-base):

**For eg.** Youtube, Twitch,Instagram,Twitter

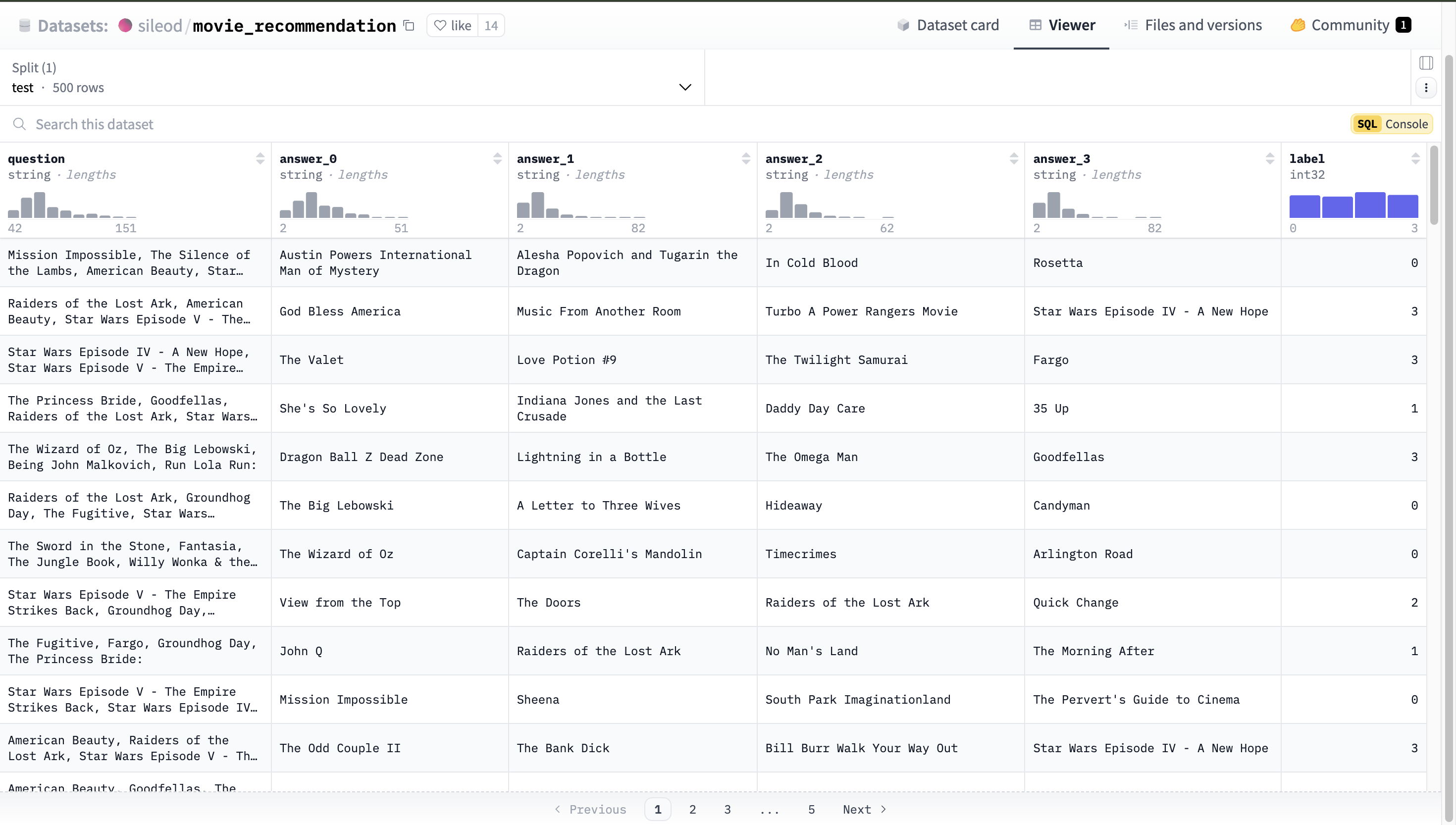
1. **Frequency: (**How often his interactions ae with his fan-base):

**For eg.** How often does his provide updates to his fan-base

And there might be several other factors

Some of the Recommendation datasets that I found include:

<https://huggingface.co/datasets/sileod/movie_recommendation>



Explanation of the Dataset:

The question column contains a list of 4 movies. The next 4 columns contain the number of people opting for each movie among the list of the movies available. The label column gives the index of the question (0,1,2,3) telling the movie with highest watch count.

**Collaborative Filtering:**

**Intuition:**

This is a method used in user-item recommendation systems which are based on the idea “Users with similar interests will have common preferences”.

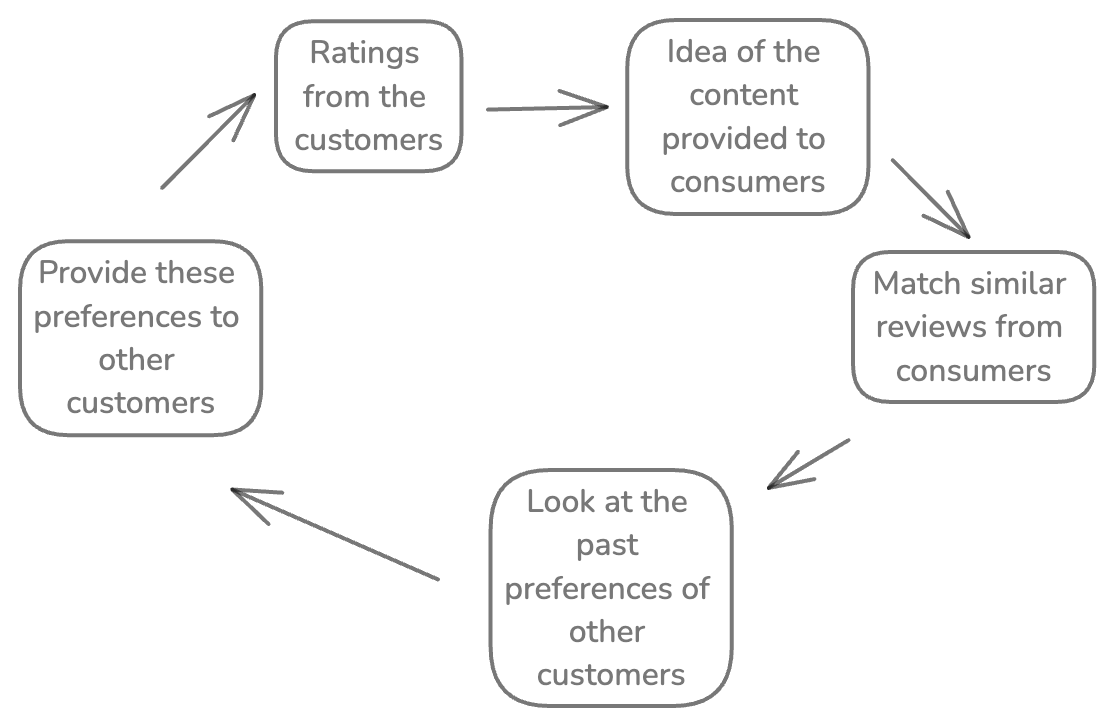
Basically, “Similar past preferences can provide insights about future predictions”.

To compute the similarity in preferences we have multiple methods:

1. Cosine Similarity
2. Pearson Correlation coefficient
3. Euclidean Distance
4. Jaccard Distance
5. Manhattan Distance
6. Spearman Rank Correlation
7. Hamming Distance
8. Tanimoto Coefficient

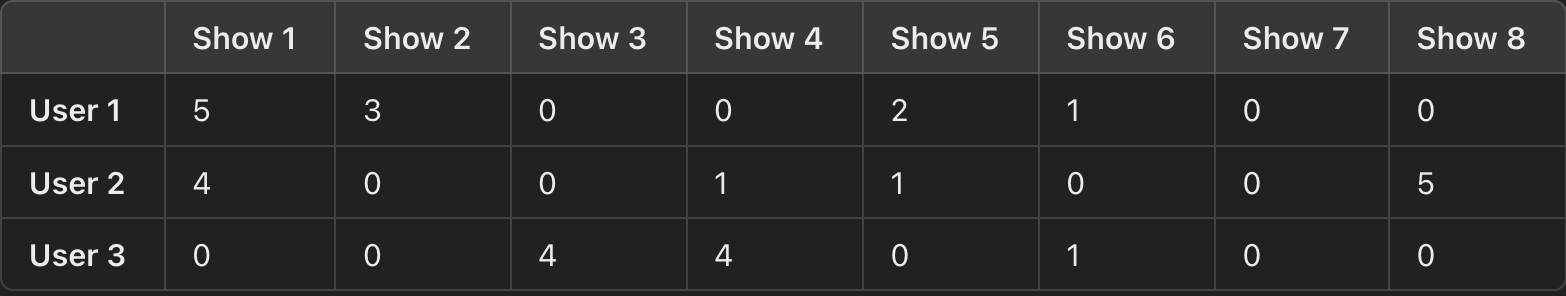
Cosine Similarity is what that is primarily used for the following reasons:

1. Scale Invariance
2. Simplicity and Efficiency

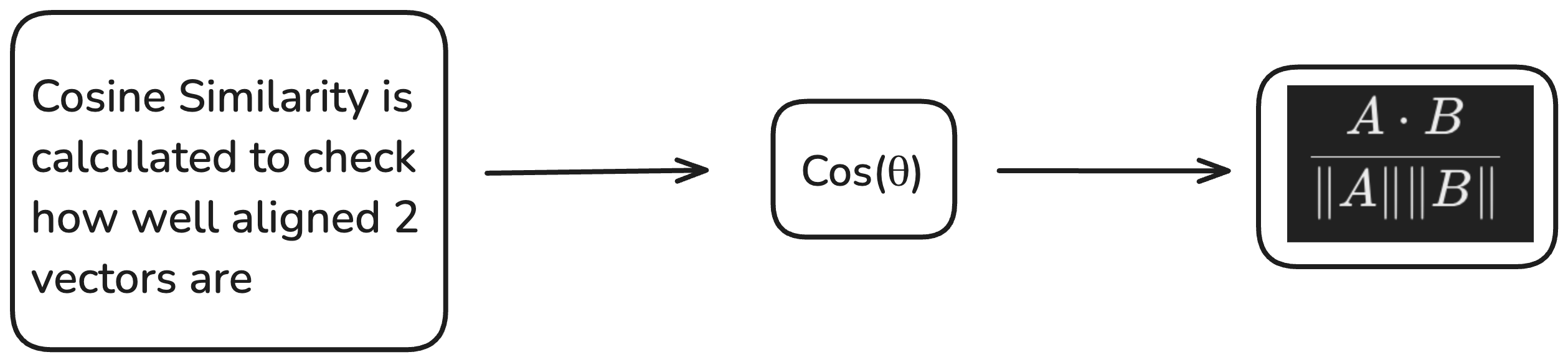


**(source: Me)**

**How Cosine Similarity is Calculated:**

****

This is an example matrix of the ratings of 8 different shows by 3 different users and we will predict the ratings of unrated shows of user1 based on the rankings of user2 and user3.



**Example:**

Let’s denote the similarity of user1 with user2 as u12 and u13 for the other pair. Let’s suppose we have the rating for a particular movie by both user1 and user2. To find the rating of that movie by user3 we take the weighted mean of both the users where the weights are the cosine similarities.

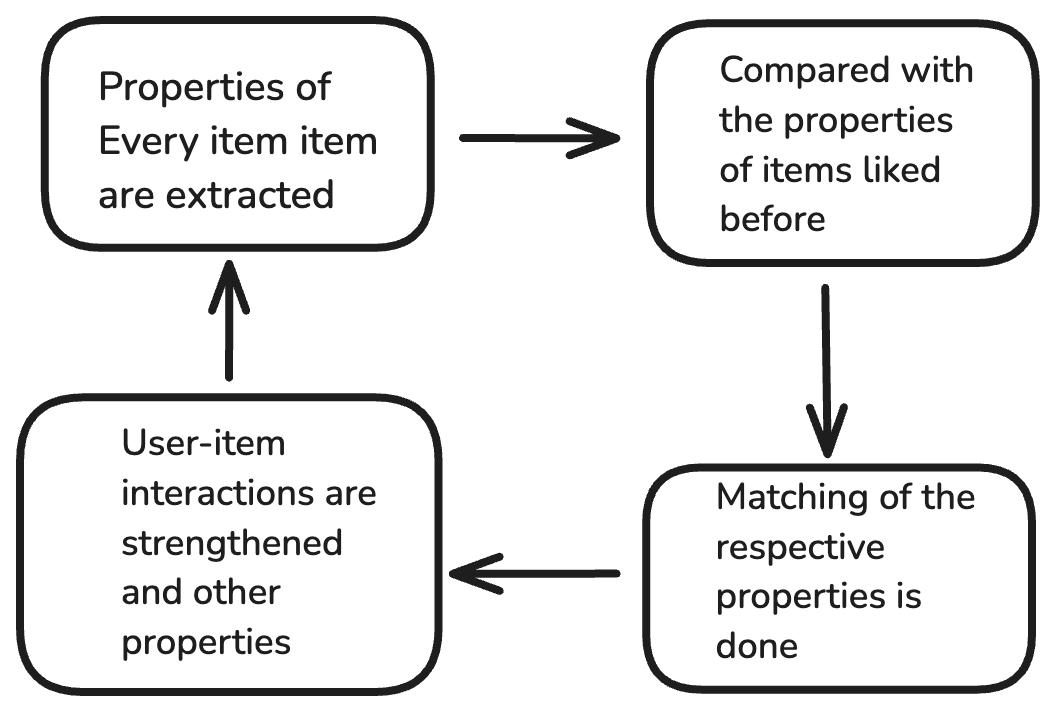
**Barriers in Collab Filtering**

* Cold Start problem: For initial users and items, there isn’t initial data to make predictions based on. This creates a problem of suggestions to new users and/or new items
* Sparsity: Most of today’s users don’t rank items. They make use of them and do not give reviews which creates a very sparse item-user interaction matrix
* Gray-Sheep problem: Several users have very unpredictable preferences and do not align with any particular group of users. They cannot be clustered with any group leading to poor recommendations for them

**CONTENT-BASED FILTERING:**

Content-based filtering solves a lot of these problems faced as it is not user-specific. It focusses on the item data. Works on the data that is taken from items (either explicitly i.e. rating) or (implicitly i.e. clicks).

This method depends more on the item’s features and not on the user preferences or behaviors of other people.



**Content-based filtering solves the Cold-start problem since it doesn’t need the user’s interactions to create the recommendation matrix**

**It also solves the problem of niche-interest since it doesn’t focus on any item in a niche and rather focusses on the properties of the item of that niche which can be matched with properties of items in other niches as well**

**References:**

1. [**https://youtu.be/Fmtorg\_dmM0?si=csfbJJKGPqCgfvZi**](https://youtu.be/Fmtorg_dmM0?si=csfbJJKGPqCgfvZi)
2. [**https://www.geeksforgeeks.org/collaborative-filtering-ml/**](https://www.geeksforgeeks.org/collaborative-filtering-ml/)
3. [**https://developers.google.com/machine-learning/recommendation/collaborative/basics**](https://developers.google.com/machine-learning/recommendation/collaborative/basics)
4. [**https://www.kaggle.com/code/ibtesama/getting-started-with-a-movie-recommendation-system**](https://www.kaggle.com/code/ibtesama/getting-started-with-a-movie-recommendation-system) **: very informative**
5. [**https://www.geeksforgeeks.org/ml-content-based-recommender-system/**](https://www.geeksforgeeks.org/ml-content-based-recommender-system/)
6. [**https://www.kaggle.com/code/omeroruccelik/content-based-recommendation-systems**](https://www.kaggle.com/code/omeroruccelik/content-based-recommendation-systems) **: very informative**