**1.What is a recommendation system? How does it work?**

* A recommender system is an application of machine learning that provides recommendations to users on what they might like based on their historical preferences. It can be further defined as a system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting objects in a larger space of possible options.

Examples:

* Offering news articles readers, based on reader’s interests.
* Offering customers suggestions about what they might like to buy, based on their past history of purchases/ searches.

## How does a Recommendation Engine Work?

One of the crucial components behind the working of a product recommendation engine is the recommender function, which considers specific information about the user and predicts the rating that the user might assign to a product.

* Having the ability to predict user ratings, even before the user has provided one, makes recommender systems a powerful tool.
* It uses specialized algorithms and techniques that can support even the largest of product catalogs. Driven by an orchestration layer, the recommendation engine can intelligently select which filters and algorithms to apply in any given situation for a specific customer. It allows marketers to maximize conversions and also their average order value.

**Collection**

* Data collected here can be either explicit such as data fed by users (ratings and comments on products) or implicit such as page views, order history/return history, and cart events.

**Analyzing**

* The recommender system analyzes and finds items with similar user engagement data by filtering it using different analysis methods such as batch analysis, real-time analysis, or near-real-time system analysis.

**Storing**

* The type of data you use to create recommendations can help you decide the kind of storage you should use, like the NoSQL database, a standard SQL database, or object storage.

**Filtering**

* The last step is to filter the data to get the relevant information required to provide recommendations to the user. And for enabling this, you will need to choose an algorithm suiting the recommendation engine.

**2. What is the difference between Collaborative and Content Based Recommender Systems?**

Here is a list of points that differentiate Collaborative Filtering and Content-Based Filtering from each other :

* The Content-based approach requires a good amount of information about items’ features, rather than using the user’s interactions and feedback. They can be movie attributes such as genre, year, director, actor etc. or textual content of articles that can be extracted by applying Natural Language Processing. Collaborative Filtering, on the other hand, doesn’t need anything else except the user’s historical preference on a set of items to recommend from, and because it is based on historical data, the core assumption made is that the users who have agreed in the past will also tend to agree in the future.
* Domain knowledge in the case of Collaborative Filtering is not necessary because the embeddings are automatically learned, but in the case of a Content-based approach, since the feature representation of the items is hand-engineered to an extent, this technique requires a lot of domain knowledge to be fed with.
* The collaborative filtering model can help users discover new interests and although the ML system might not know the user’s interest in a given item, the model might still recommend it because similar users are interested in that item. On the other hand, A Content-based model can only make recommendations based on the existing interests of the user and the model hence only has limited ability to expand on the users’ existing interests.
* A Content-Based filtering model does not need any data about other users, since the recommendations are specific to a particular user. This makes it easier to scale down the same to a large number of users. A similar cannot be said or done for Collaborative Filtering Methods.
* The collaborative algorithm uses only user behavior for recommending items while for Content-based filtering we have to know the content of both user and item.

**3. What are similarity measures? Explain their types.**

The similarity measure is the measure of how much alike two data objects are. A similarity measure is a [data mining](https://dataaspirant.com/data-mining/) or machine learning context is a distance with dimensions representing features of the objects. If the distance is small, the features are having a high degree of similarity. Whereas a large distance will be a low degree of similarity.

* Similarity measure usage is more in the text related [preprocessing techniques](https://dataaspirant.com/nlp-text-preprocessing-techniques-implementation-python/), Also the similarity concepts used in advanced [word embedding](https://dataaspirant.com/word-embedding-techniques-nlp/) techniques. We can use these concepts in various deep learning applications. Uses the difference between the image for checking the data created with [data augmentation](https://dataaspirant.com/data-augmentation-techniques-deep-learning/) techniques.
* The similarity is subjective and is highly dependent on the domain and application.
* For example, two fruits are similar because of color or size or taste. Special care should be taken when calculating distance across dimensions/features that are unrelated. The relative values of each element must be normalized, or one feature could end up dominating the distance calculation.
* Generally, similarity are measured in the range 0 to 1 [0,1]. In the machine learning world, this score in the range of [0, 1] is called the similarity score.
* Two main consideration of similarity:
* Similarity = 1 if X = Y         (Where X, Y are two objects)
* Similarity = 0 if X ≠ Y

# **1.Cosine Similarity:**

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.

2. Manhattan distance is a metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates. In a simple way of saying it is the total sum of the difference between the x-coordinates and y-coordinates.

**3.** **Euclidean distance:**

The Euclidean distance between two points in either the plane or 3-dimensional space measures the length of a segment connecting the two points. It is the most obvious way of representing distance between two points.

**4.** **Minkowski distance**

Minkowski distance is a generalisation of the Euclidean and Manhattan distances.

**5.Jaccard similarity:**

The **Jaccard index**, also known as **Intersection over Union** and the **Jaccard similarity coefficient** is a statistic used for gauging the similarity and diversity of sample sets. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets.

**Q4. what are challenges involved with recommendation engines?**

### 

### **1.Cold start**

* This problem arises when new users or new items are added to the system, a new item can’t recommend to users initially when it is introduced to the recommendation system without any rating or reviews and hence it is hard to predict the choice or interest of users which leads to less accurate recommendations.
* For example, a newly released movie cannot be recommended to the user until it gets some ratings. A new user or item added based problem is difficult to handle as it is impossible to obtain a similar user without knowing previous interest or preferences.

### **Sparsity**

* It happens many times when most of the users do not give ratings or reviews to the items they purchased and hence the rating model becomes very sparse which could lead to data sparsity problems, it decreases the possibilities of finding a set of users with similar ratings or interest.

### **Synonymy**

* Synonymy arises when a single item is represented with two or more different names or listings of items having similar meanings, in such condition, the recommendation system can’t recognize whether the terms shows various items or the same item.

For example, recommendation systems predict ‘action movie’ or action film’ the same.

### **Privacy**

* Generally, an individual needs to feed his personal information (have an experience with [hyper-personalization](https://www.analyticssteps.com/blogs/what-hyper-personalization-benefits-framework-and-examples) to the recommendation system for more beneficial services but it causes the issues of data privacy and security, many users feel hesitation to feed their personal data into recommendation systems that suffer from data privacy issues.
* The recommendation system is bound to have the personal information of users and use it to the fullest in order to provide personalized recommendation services. To deal with this issue, the recommendation systems must ensure trust among their users.

### **Scalability**

* One biggest issue is the scalability of algorithms having real-world datasets under the recommendation system, a huge changing data is generated by user-item interactions in the form of ratings and reviews and consequently, scalability is a big concern for these datasets.
* Recommendation systems interpret results on large datasets inefficiently, some advanced large-scaled methods are required for this issue.

1. Latency

* We observe many products are added more frequently to the database of recommendation systems, only already existing products are recommended to users as newly added products are not rated yet.
* so an issue of Latency arises. The collaborative filtering method and category-based approach in combination with user-item interaction can be used to deal with this issue.

**Q5. What is a hybrid recommendation system?**

* A hybrid recommendation system is a special type of recommendation system which can be considered as the combination of the content and collaborative filtering method. Combining collaborative and content-based filtering together may help in overcoming the shortcoming we are facing at using them separately and also can be more effective in some cases.
* Hybrid recommender system approaches can be implemented in various ways like by using content and collaborative-based methods to generate predictions separately and then combining the prediction or we can just add the capabilities of collaborative-based methods to a content-based approach (and vice versa).