**1. Recommender:**

The goal of a recommender system is to generate meaningful recommendations to a collection of users for items or products that might interest them. Suggestions for books on Amazon, or movies on Netflix, are real-world examples of the operation of industry-strength recommender systems. The design of such recommendation engines depends on the domain and the particular characteristics of the data available.

**2. Job Recommender:**

The job recommender system, which is the online recruiting system with personalized recommendation, has been proposed to handle the aforementioned issue for job seekers and enterprises. As a recommender system, the job recommender system is capable of retrieving a list of job positions that satisfy a job seeker’s desire, or a list of talent candidates that meet the requirement of a recruiter by using the recommendation technology.

**3. Types of recommender System**

**3.1 Content-based filtering**:

CBF also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. CBF checks age, gender, skills and his degree preferences.CBF will job recommend seeing this content even will also see his location and provides job in his area. if user belongs to Faisalabad then it will recommend job in Faisalabad.

**3.2 Collaborative filtering**:

collaborative filtering is a family of algorithms where are multiple ways to find similar users or items and multiple ways to calculate rating based on ratings of similar users and similar items. Depending on choices you make, you end up with a type of CF approach.CF recommends jobs according to the previous users criteria. If three users have a same contents then user A have android Application job and user B as well then system will recommend same job android Application to user C. And Item based collaborative filtering Between the rating patterns of items. Since finding similar items is easier than finding similar users, and attributes of items are more stable than users’ preference, item-based methods are suitable for off-line computing.

**3.3 Hybrid Filtering:**

Hybrid Filteringis a collection of both CBF and CF. In which we use both for job recommendation to take outcome as user wants. In which user use both filtering system at a time. In which Hybrid filtering system will apply both filtering system(CBF & CF) based on the user profiles or info etc. And then will show result for the user which will be best and according to the profile of the user.

**Issues in Recommendation System:**

**4.1 Sparsity:**

Sparsity problem creates from the phenomenon that users in general rate only a limited number of items. In recommender system we want to see which user bought which item or which item is rated by which user. In practical scenario users do not rate every item, Mostly users are concentrated on few items in recommender system and since a mostly number of items are untouched by users.

When someone user do not or skip to rate the number of items in recommender then Sparsity problems occurs to recommend to users. i.e. Mostly users in play store do not rating the app that they have used and so here Sparsity problem can be occur to users for recommend that what to recommend

**4.2 Grey Sheep:**

Grey Sheep problem comes in the group of users as collaborative filtering(CF).

Where CF suggesting or recommending items to users who preferences match with other users in the recommendation system. CF is that given user will prefer items which are used by people who have similar preferences as the given user.

Grey Sheep problem creates where interest of user **A** match with user **B,C** but not matched with user **D**. As user **B** and **C** have interested in the Full stack development and user **C** have interested in java programming language. While user **A** interested in the web developer. Here, system will recommend to user **A** of user **B** and **C** not **D** because they different from user **A**. Grey Sheep problem creates here.

When new user interest do not match with other users or few number of items then Grey Sheep problem can be create where user interest do not match with other users using CF(Collaborative Filtering).

**4.3 Coldstart:**

Cold start refers to the difficult the recommendation system for new users or new items. For the new users or new items, in general, they report or receive only a few or no ratings. Both issues will prevent the CF from providing effective recommendations, because users’ preference is hard to extract.

Cold start concerns the issue that the system cannot draw any inferences for [users](https://en.wikipedia.org/wiki/User_(computing)) or items about which it has not yet gathered sufficient information.

There are three cases of cold start:

**New community**:

New community refers to the start-up of the recommender when catalogue of items might exist, almost no users are present and the lack of user interaction makes it very hard to provide reliable recommendations to users.

**New item**:

A new item is added to the system, it might have some content information but no interactions are present. The item cold-start problem refers to when items added to the catalogue have either none or very little interactions. This constitutes a problem mainly for [CF](https://en.wikipedia.org/wiki/Collaborative_filtering) algorithms due to the fact that they rely on the item's interactions to make recommendations. If no interactions are available then a pure collaborative algorithm cannot recommend the item

**New user**:

A new user registers and has not provided any interaction yet, therefore it is not possible to provide personalized recommendations. when new user comes in system then [Collaborative filtering](https://en.wikipedia.org/wiki/Collaborative_filtering) algorithms are the most affected as without interactions no inference can be made about the user's preferences because system do not match content of the user.

**References:**

**1.0**

<https://link.springer.com/10.1007%2F978-0-387-30164-8_705>

**2.0**

<http://ieeexplore.ieee.org/document/6295216>

**3.1**

https://www.ijntr.org/download\_data/IJNTR03040022.pdf

**4.1**

<https://www.quora.com/What-is-the-sparsity-problem-in-a-recommender-system>

**4.1**

<https://www.quora.com/What-is-the-sparsity-problem-in-a-recommender-> system#:~:text=inherently%20suffers%20from%3A-,Data%20sparsity%20arises%20from%20the%20phenomenon%20that%20users%20in%20gener al,a%20limited%20number%20of%20items%3B&text=Both%20issues%20will%20prevent%20th

e,preference%20is%20hard%20to%20extract

**4.2**

<https://dl.acm.org/doi/abs/10.1145/3125649.3125651#:~:text=%22Grey%20Sheep%22%20users%20is%20a,to%20produce%20accurate%20collaborative%20recommendations>.

**4.3**

<https://yuspify.com/blog/cold-start-problem-recommender-systems/>

**4.3**

<https://en.wikipedia.org/wiki/Cold_start_(recommender_systems)>