1. INTRODUCTION
   1. RECOMMENDATION SYSTEM

On the Internet, where the quantity of decisions is overpowering, there is a need to have a filter which can prioritize the information and can efficiently deliver that information to overcome the problem of overloaded information, which creates problems to the Internet users. Recommender system tackles this problem by progressively looking through substantial volume of data to provide its users with more personalized content and services. Recommender System has the ability to predict whether a particular user would prefer the item or not based on user’s profile. It is beneficial for both the users and the organization providing services. It reduces the costs of finding and selecting items in an online shopping environment, and also have proved to improve the decision making process and the quality. For e.g. whenever we go to say YouTube there based on what we had watched in the past, we get the some more videos similar to them. These videos which are similar to the videos watched previously are the recommendations which are done by the recommendation system.

But nowadays making accurate and good recommendations is not that so easy task to do, because there are too much of data present with us which needs to be processed in order to find the good similar results and give them to users in the form of recommendations. This increases the time to process the data and needs powerful machines to do these computations. So in order to solve these types of problem, we had tried to make some algorithms which can efficiently process the data and give us some good and accurate results. Another problem with recommendation system is that as data is increasing exponentially the methods used to process those data and make recommendation are getting old. So here we had tried to make our algorithms scalable enough which can process more quantity of data easily and still yield good results.

* 1. APPROACHES FOR DESIGNING RECOMMENDATION SYSTEM

There are some famous and frequently used approaches for making recommendation systems :-

**COLLABORATIVE FILTERING**

In this approach for making recommendations to the particular user we need to have information about some users also other than to whom we are making these recommendations. This information can be about anything which can give us information about the interest of the users, which can distinguish them. Through this data we can find the users similar to the user to whom we have to make recommendations and give the items liked by those users as recommendation. This approach does not rely on items themselves but also on how the users have reacted towards those items. It is based on one assumption that if a user X has same interest as that of user Y in the past, they are more likely to have same interest in future also.

For e.g.

Let’s say that we have a portal for watching movies online and giving reviews on them, and in that portal we have 4 users (Robert, Tony, Leonard, Ross), who had watched 4 movies (The Conjuring, Captain America : Civil War, The Conjuring 2 and Minions) and also had given some reviews on them. Here “Yes” means that a user has rated that movie high, and “No” means that user has given lower rating to that movie.

Now there is a new user named Michael, who has watched only The Conjuring, and had given a higher rating to it. Now he wants to see other movies also but is confused as to watch which other movie. So his confusion can be solved by the recommendation system by recommending him some movie which he can watch and will like it.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Image result | Related image | Image result | Related image |
| Robert | Yes | Yes | Yes | Yes |
| Tony | No | Yes | No | Yes |
| Leonard | Yes | No | Yes | No |
| Ross | No | No | Yes | Yes |
| Michael | Yes | ? | ? | ? |

Now recommendation will be done by finding the user similar to the Michael, who possesses interest to him, now as Robert and Leonard both has rated The Conjuring higher, and Tony and Ross has not given higher rating to it. So, we find that Robert and Leonard possesses same interest as that of Michael.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Image result | Related image | Image result | Related image |
| Robert | Yes | Yes | Yes | Yes |
| Leonard | Yes | No | Yes | No |

Now we will find the movie which have been given higher ratings by both Robert and Leonard, and that movie is The Conjuring 2, So it would be recommended to Michael.

**TYPES OF COLLABORATIVE FILTERING**

**Memory Based** - It is categorized into two parts, User based, and Item based. With the help of the given U - I rating matrix, a user based CF approach predicts a user’s rating on a target item by aggregating the ratings that the similar users have given previously. In contrast to the user - based CF, item based CF recommends item based on the information about the other items that a user has previously rated.

**Model Based -** This type uses some machine learning algorithms to train the data and make models based on them.

**Hybrid -** This model uses both the characteristics of the Memory Based and Model Based Collaborative Filtering.

**CONTENT BASED FILTERING**

In this approach rather than concentrating on the information of the users we concentrate on the features of the items available to us. Here similarity is determined based on their features only. For e.g. let’s say if a user X has watched Harry Potter which is a fiction movie with magic involved in it. Then other movies like other parts of Harry Potter which have almost same actors, can be recommended to him as they are similar to it, and also movies like Fantastic Beasts and Where to find them, which is of similar type as of fictional and also have magic involved in it can be recommended.

**HYBRID**

This uses the features of both approaches Collaborative and Content Based Filtering for recommendation.

* 1. CROSS DOMAIN

Till now what we have talked about is recommending items belonging to one domain only like recommending movies from the movies dataset, recommending songs from songs dataset etc. Now when we extend these recommendation approaches to different domain, which can help improve the recommendations. This type of recommendations can be useful in e-commerce sites where recommendations of different types are given.

**NOTION OF DOMAIN**

Here domains can be anything of different items sharing some common attributes like, Movies and Books, Movies and Music, etc., domains can be of same items whose information is collected at different sources, like of movies one dataset of movie lens and other of Kaggle.

**LEVEL OF DOMAIN**

**ATTRIBUTE LEVEL (ACTION - FANTASY)**

Same type of items having different values of certain attribute

**TYPE LEVEL (MUSIC - BOOKS)**

Similar types sharing some attributes

**ITEM LEVEL (BOOKS - HOTELS)**

Items of distinct types, differing in most if not all attributes.

**SYSTEM LEVEL (IMDB – ROTTEN TOMATOES)**

Same items whose information is collected at different sources.

Here in this project we had used System level domains.

**DOMAIN A DOMAIN B**

Transfer Of Knowledge

**USER - MOVIE PREFERENCE USER–MUSIC PREFERENCE**

Penny 8 – Mile Matrix Mark Lose Yourself My Heart ill go on

Monica Skyfall Inception Peter What I’ve Done Skyfall

### Joey 8 – Mile Skyfall Kate Lose Yourself Skyfall

**Recommendation**

Penny Titanic

Now let’s take a look at an example of Cross Domain Collaborative Filtering

Here in this example Domain A is of Movie and Domain B is of Music. Let’s say in Domain A, Penny has watched “8 - Mile" and has given it a 5 stars and ”Matrix” and given it 2 stars only, now we would like to recommend some more movies to Penny.

So to do this first we will find the user similar to Penny from Domain A only, as we can see that there is no similar user to Penny in domain A, so will go to the other domain, Domain B and will try to find the similar user from that domain also. In that domain we find that Mark and Kate are similar to the Alice as both has given 5 stars “Lose Yourself” which is the title song of the movie “8 - Mile”, so one of the recommendation that can be given to Penny is “Titanic” because Mark has also given 4 stars to the “My Heart Will Go on” which is the title song of the movie Titanic.

* 1. CHALLENGES TO CROSS DOMAIN

**What to Share -** To decide what information is to be shared between our target domain and source domains.

**How To Share -** To decide how efficiently we can transfer the information between domains.

* 1. DATASET USED

Movie Lens dataset has over 10000 movies listed and ratings from Rotten Tomatoes, where users have given ratings on the scale of 1-5. It also contains a User Dataset which has over 10000+ users where each user has rated at least 20 movies.

The Kaggle Dataset has 5000 movies which has ratings from IMDB, where users have given rating on the scale of 1-10