***Movie Recommendation System***

Steps we might follow to build our movie recommendation system:

**Step 1:** First of all, we will import all the necessary libraries and packages

**Step 2:** As we discuss we will use movies.csv and ratings.csv as our dataset, we will retrieve our data from movies.csv and ratings.csv into dataframes. Here we can also display information about movies if we want. (we have lots of function to overview the data e.g. summary(), str(), head() )

**Step 3:** In the movie dataframe, we will have 3 types of values which are movieId, title, genres. We will convert the genres present in the movie dataframe into a more usable format by the users. To do so, we will first create a [**one-hot encoding**](../Users/Taufiqul/Desktop/IRMANA%20R%20project/One-hot%20encoding%20in%20R_%20three%20simple%20methods%20_%20Data%20Tricks.html) to create a matrix that comprises of corresponding genres for each of the films.

**Step 4:** After that, we will create a ‘search matrix’ that will allow us to perform an easy search of the films by specifying the genre present in our list. There could be movies that fall into several genres like one movie can be comedy, action, and fantasy.

**Step 5:** For our movie recommendation system to make sense of our ratings through recommenderlabs (it is a keyword of R), we have to convert our matrix into a [**sparse matrix one**](../Users/Taufiqul/Desktop/IRMANA%20R%20project/Sparse%20Matrix%20Construction%20And%20Use%20In%20R%20-%20GormAnalysis.html).

**Step 6:** We will implement a single model in our R project – [**Item Based Collaborative Filtering**](../Users/Taufiqul/Desktop/IRMANA%20R%20project/Item-based%20collaborative%20filtering%20recommendation%20algorithms%20_%20Proceedings%20of%20the%2010th%20international%20conference%20on%20World%20Wide%20Web.html).

**Step 7:** Now we will need to explore similar data. So here we will observe suggesting movies to the users that are based on collecting preferences from many other users. For example, if user A likes to watch action films and so does user B, then the movies that the user B will watch in the future will be recommended to A and vice-versa. Therefore, recommending movies is dependent on creating a relationship of similarity between the two users. With the help of recommenderlab, we can compute similarities using various operators like cosine, Pearson as well as jacquard. (These are functions of R)

**Step 8:** From the matrix, each row and column will represent a user and each cell in this matrix represents the similarity that is shared between the two users.

**Step 9:** Then, we delineate the similarity that is shared between the films.

**Step 10:** After that, we will create a table of ratings that will display the most unique ratings.

**Step 11:** Now we will **analyze the most viewed movie visualization** from our dataset. We will first count the number of views in a film and then organize them in a table.

**Step 12:** Then, we will visualize a **bar plot** for the total number of views of the top films. We will carry this out using ggplot2.

**Step 13:** After we can visualize a [**heatmap**](../Users/Taufiqul/Desktop/IRMANA%20R%20project/Heatmap%20in%20R_%20Static%20and%20Interactive%20Visualization%20-%20Datanovia.html) of the movie ratings.

**Step 14:** After that, we will visualize the **distribution of the average ratings per user**.

**Step 15:** We will conduct data preparation in the following three steps – Selecting useful data, Normalizing data, Binarizing the data.

**Step 16:** We will now explore the various parameters of our Item Based Collaborative Filter. These parameters are default in nature.

**Step 17:** Using the getModel() function, we will retrieve the recommendation model. We will then find the class and dimensions of our similarity matrix that is contained within model info.

**Step 18:** Finally, we will generate a heatmap, that will contain the top items and visualize the similarity shared between them.

**Step 19:** In the next step of project, we will carry out the sum of rows and columns with the similarity of the objects. We will visualize the sum of columns through a distribution

**Step 20:** We will create a variable which will be initialized to 10, specifying the number of films to each user. We will then use the predict () function that will identify similar items and will rank them appropriately. Here, each rating is used as a weight. Each weight is multiplied with related similarities. Finally, everything is added in the end.