**Movie Recommendation System**

What is a Recommendation system?

During the last few decades, with the rise of You-tube, Amazon, Netflix and many other such web services, recommender systems are today unavoidable in our daily online journeys. In a very general way, recommender systems are algorithms aimed at suggesting relevant items to users .It has spread its branches starting from items being movies to watch, text to read, products to buy or anything else depending on industries).Recommender systems are really critical in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors. So in simple language

*“a recommendation system suggest anything relevant based on the used interest”*

Movie Recommender System

Movie Recommendation system adds a whole new dimension to the movie watching experience by providing real-time personalized movie recommendations to users.  Such a system can suggest a set of movies to users based on their interest, or the popularities of the movies. It mines movie databases to collect all the important information, such as, popularity and attractiveness, required for recommendation. It generates movie swarms not only convenient for movie producer to plan a new movie but also useful for movie recommendation. Experimental studies on the real data reveal the efficiency and effectiveness of the proposed system.

Different ways to design a Movie Recommender System

We often rate movies on the different platforms and all the preferences we express and data we share (explicitly or not), are used by recommender systems to generate, in fact, recommendations. The rating websites like IMDB, Rotten Tomatoes etc. maintain a huge database of movie ratings by user which are ultimately being used by the movie recommendation systems. There are different approaches to design such kind of machines.

* Content-Based Approach

The idea behind Content-based (cognitive filtering)recommendation system is to recommend an item based on a comparison between the content of the items and a user profile. User may get recommendation for a movie based on the description of other movies watches already by the user.

Considering different attributes/features of a movie we can recommend similar kind of movies to the user.

Example:

Suppose we are considering rating as a parameter to recommend some movies to the user. For this approach we can collect the database of user ratings given and compare with rating given by the user which he has already watched. We can recommend some movies of a particular genre having similar range of ratings. Similar way we can add move attributes of movies such as Lead Actor,Director,Plot, Emotion, Content etc to personalize the user profile.

* Collaborative Based Recommender System

The idea behind collaborative filtering is to work with collaboration with user and movies. Using such concept, online collaborative movie recommendations make attempts to assist users to access their preferred movies by capturing precisely similar neighbors among users or movies from their historical common ratings.

I t works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user. It looks at the items they like and combines them to create a ranked list of suggestions.

Exp:

Suppose we have a dataset of user and list of ratings of the movies given by him. We can find user-user correlation matrix and find some similar kind of users’ .Then we can suggest some movies to watch as per the watch list of the similar taste users.

Similar way we can apply some other techniques like movie-movie correlation, some popular machine learning and deep learning algorithms like PCA,SVD etc.

Let’s Code

**Content based approach**:

Regarding Dataset:

**100,000 ratings and 3,600 tag applications applied to 9,000 movies by 600 users**

*Merge function is used to concatenate two datasets one consisting of movie name and movie ID dataset to the ratings dataset.*

# In[7]:

df\_result=movies\_df.merge(ratings\_df,on='movieId')

df\_result=df\_result.merge(tags\_df,on='movieId')

*Creating a data frame of the rating mean of each movie. But the demerit doing it is some movies is not watched by many people by have been give rating which does not infer to be popular movie.For this reason we need to look at some other options in subsequent coding part.*

rating=pd.DataFrame(df\_result.groupby('title')['rating'].mean()

*The number of ratings given by the users plays primary role here so that we can guess if the movie has been seen by most number of people*

rating\_count=pd.DataFrame(df\_result.groupby('title')['rating'].count().sort\_values(ascending=False))

rating['number of total ratings']=pd.DataFrame(df\_result.groupby('title')['rating'].count())

rating.sort\_values(by='number of total ratings',ascending=False)

*Creating a pivot table having individual movies and individual used in the columns and rows correspondingly.It will have lot of NaN values because most of the movies are not watched by most of the users*

movie\_matrix=df\_result.pivot\_table(index='userId\_y',columns='title',values='rating')

movie\_name=input("Please put a movie name to see the recommended movies to watch next")

*Suppose we want to find some similar moves related to ‘Days Of Summer’ since it was watched by some user.*

movie\_name='(500) Days of Summer (2009)'

*Collecting the movie ratings in a series*

movie=movie\_matrix[movie\_name]

*Most of the movies will have null values in most of the rows, we need to remove the null values*

movie.dropna(inplace=True)

*Applying the correlation with the rest of the movie columns to get the similar movies*

Sim\_movie=movie\_matrix.corrwith(movie)

Sim\_movie=pd.DataFrame(Sim\_movie,columns=['Correlation'])

*Getting similar movies with most number of ratings in descending order i.e. populat movies having highest ratings*

Sim\_movie=Sim\_movie.join(rating['number of total ratings']).sort\_values('number of total ratings',ascending=False)

*Output got is below*

Correlation number of total ratings

title

Pulp Fiction (1994) NaN 55567

Fight Club (1999) NaN 11772

Star Wars: Episode IV - A New Hope (1977) NaN 6526

Léon: The Professional (a.k.a. The Professional) (Léon) (1994) NaN 4655

2001: A Space Odyssey (1968) NaN 4469

**Collaberative based approach**: