**Analysis of Movie Recommendations with movielens dataset**

MovieLens:

MovieLens is a web-based [recommender system](https://en.wikipedia.org/wiki/Recommender_system) and [virtual community](https://en.wikipedia.org/wiki/Virtual_community) that recommends movies for its users to watch, based on their film preferences using [collaborative filtering](https://en.wikipedia.org/wiki/Collaborative_filtering) of members' movie ratings and movie reviews. It contains about 11 million ratings for about 8500 movies. MovieLens was created in 1997 by [GroupLens Research](https://en.wikipedia.org/wiki/GroupLens_Research" \o "GroupLens Research), a research lab in the Department of Computer Science and Engineering at the [University of Minnesota](https://en.wikipedia.org/wiki/University_of_Minnesota), in order to gather research data on personalized recommendations.

Recommendation:

MovieLens bases its recommendations on input provided by users of the website, such as movie [ratings](https://en.wikipedia.org/wiki/Star_(classification)). The site uses a variety of recommendation algorithms, including [collaborative filtering](https://en.wikipedia.org/wiki/Collaborative_filtering) algorithms such as [item-item](https://en.wikipedia.org/wiki/Item-item_collaborative_filtering), user-user, and regularized [SVD](https://en.wikipedia.org/wiki/Singular_value_decomposition). In addition, to address the [cold-start](https://en.wikipedia.org/wiki/Cold_start_(recommender_systems)) problem for new users, MovieLens uses [preference elicitation](https://en.wikipedia.org/wiki/Preference_elicitation) methods.

For each user, MovieLens predicts how the user will rate any given movie on the website. Based on these predicted ratings, the system recommends movies that the user is likely to rate highly. The website suggests that users rate as many fully watched films as possible, so that the recommendations given will be more accurate, since the system would then have a better sample of the user's film tastes.

Dataset:

The final data used here consist of 4 rows and 2269 columns.

* Ratings.csv :
  + Related to user ratings.
  + Consisting 100836 rows and 4 columns
  + Columns : userId, movieId, rating, timestamp
* Movies.csv :
  + Related to movies characteristics
  + Consisting 9742 rows and 3 columns
  + Columns : movieId, title, genres

# Collaborative filtering:

Collaborative Filtering is a Machine Learning technique used to identify relationships between pieces of data. Collaborative filtering is also known as social filtering. Collaborative filtering uses algorithms to filter data from user reviews to make personalized recommendations for users with similar preferences.

# Procedure steps:

1. Import required libraries.

import pandas as pd

from scipy import sparse

1. Read the Data using pd.read\_csv

ratings = pd.read\_csv('ratings.csv')

movies = pd.read\_csv('movies.csv')

1. Merge the two datasets into single one as ratings.

ratings = pd.merge(movies,ratings).drop(['genres','timestamp'],axis=1)

1. Drop the unnecessary data and fill the unfilled cells.

userRatings = userRatings.dropna(thresh=10, axis=1).fillna(0,axis=1)

1. Find the correlation matrix.

corrMatrix = userRatings.corr(method='pearson')

1. Apply collaborative filtering.

def get\_similar(movie\_name,rating):

similar\_ratings = corrMatrix[movie\_name]\*(rating-2.5)

similar\_ratings = similar\_ratings.sort\_values(ascending=False)

#print(type(similar\_ratings))

return similar\_ratings

1. Test the Model.

romantic\_lover = [("(500) Days of Summer (2009)",5),("Alice in Wonderland (2010)",3),("Aliens (1986)",1),("2001: A Space Odyssey (1968)",2)]

similar\_movies = pd.DataFrame()

for movie,rating in romantic\_lover:

similar\_movies = similar\_movies.append(get\_similar(movie,rating),ignore\_index = True)

similar\_movies.head(10)

# Conclusion :

The primary goal of movie recommendation system is to filter and predict only those movies that a corresponding user is most likely to want to watch. The [ML algorithms](https://labelyourdata.com/articles/how-to-choose-a-machine-learning-algorithm#5_simple_steps_to_choose_the_best_machine_learning_algorithm_that_fits_your_ai_project_needs) for these recommendation system use the **data** about this user from the system’s database. This data is used to predict the future behavior of the user concerned based on the information from the past.