**Objective**:

The main objective of this project is to recommend top ten movies among the best suitable movies for any user, based on two different approaches, *User Based Collaborative Filtering* and *Item Based Filtering.*

**Dataset**:

I used the ‘MovieLens’ dataset for generating the recommendations for the users.

The dataset contains 6 data files.

(i) udata: This file contains 100000 rating entries by 943 users on a total of 1682 different movies.

The format of this data is –

user id | item id | rating | timestamp

(ii) uinfo: This file contains the number of users, items, and ratings in the u.data set

(iii) uitem: This file contains the information about the movies.

The format of the data is –

movie id | movie title | release date | video release date |

IMDb URL | unknown | Action | Adventure | Animation |

Children's | Comedy | Crime | Documentary | Drama | Fantasy |

Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi |

Thriller | War | Western |

The last entries indicate the genres of the movies. A ‘1’ indicates that a movie belongs to that particular genre and 0 otherwise.

(iv) uuser: This file contains the demographic information about the users.

The format of the data is -

user id | age | gender | occupation | zip code

**How this Works**:

This system generate movie recommendations for user U1 based on two approaches. These two approaches are explained briefly below.

**Collaborative filtering:** Analyze the similarities between user U1 and the other users who had also given high ratings (rating r) to the same movies as U1. Rank the users based on their similarities (age, gender, occupation, zip code) with U1. Extract top x% of the users and explore the other movies (not rated by U1) rated highly by the users with top similarity to U1. Rank the movies based on the overall ranking and display the top 10 movies as the recommendation for U1.

**Item-based filtering:** Analyze the different movies rated highly by U1. Try to find similarities (genre, release date) between them and extract other movies in the database that have similar characteristics. Rank the movies based on the overall rankings and display the top 10 movies.