

## ASSIGNMENT 7: RECOMMENDER SYSTEM

Ibrahim Mohamed

### Process:

After reading the data from the csv files:

- First, we start we exploring the data and define the attributes we will be working on.
- Second, we drop the unnecessary attributes.
- Get the first 200 user only out of 610
- Match the first 200 user ratings to movies with movie\_id attribute, now we have a 1 data frame that has the user ratings to their movies.
- Filter the movies according to the count of rating each movie has and exclude any movie that has less than 20 rating
- Due to filtering our users are 198 and movies are 287
- We created a pivot table with shape 198 (Users) x 287 (no of movies) that contains each user rating to each movie, of course there is no user who has rated all movies, so NaN values are present.
- NaN Values are replaced with 0. This has a problem that we considered unseen movies as bad movies, thus we can search for solution about this point.
- Compute similarity score matrix for movies with cosine similarity using pivot table.
- Compute similarity score matrix for with cosine similarity using the transpose of the pivot table.
- Use this similarity score matrix in the next 2 functions to make the recommendation system

### Function 1: to recommend similar movies based on a movie

```
def topSimilarMovies(movie_id, number_of_similar_movies)
```

This Function takes the movie ID I want to find similar movies to and number of similar movies I want and return a list of these movies. It first checks if the movie is presented in the selected ones as we have only 287 movies from 9724. If it exists it get the top similar movies from the movie's similarity matrix and sort them in descending order. Then remove the first item as it will be the same movie due to the high similarity (1).

Output for the Dark Knight Rises (2012):

```
1 # get the movie titles of movie_id = 91529
2 my_movie_id = 91529
3 testing_movie = movies_ratings[movies_ratings['movieId'] == my_movie_id]['title'].tolist()[0]
4 num_of_movies = 10
5
6 # print the top similar movies
7 print(f'The selected movie is {testing_movie} and the top {num_of_movies} similar movies are: ')
8 for item in topSimilarMovies(my_movie_id, num_of_movies):
9     print(item)
```

✓ The selected movie is Dark Knight Rises, The (2012) and the top 10 similar movies are:  
Dark Knight, The (2008)  
Up (2009)  
Inception (2010)  
Django Unchained (2012)  
WALL·E (2008)  
Inglourious Basterds (2009)  
Interstellar (2014)  
Iron Man (2008)  
Batman Begins (2005)  
Hangover, The (2009)

## Function 2: to recommend movies to user

```
def recommendMovieToUser(user_id, number_of_similar_movies):
```

This Function takes the user ID I want to recommend movies to and number of similar movies I want and return a list of these movies.

It starts by getting the movies that the user has watched then get the similarity score between the user and all users to get users who are like him. We get the most 10 similar user and then get the movies they have watched. We need to filter them to see what they have watched and our target user hasn't. then get the titles of the movies

Output for user 200:

```
1 # get the top 10 recommended movies for user_id = 200
2 user_id = 200
3 num_of_movies = 5
4 print(f'The top {num_of_movies} recommended movies for user_id = {user_id} are: ')
5 for item in recommendMovieToUser(user_id, num_of_movies):
6     print(item)
```

✓ The top 5 recommended movies for user\_id = 200 are:  
Jumanji (1995)  
Hangover, The (2009)  
Heat (1995)  
Sabrina (1995)  
Robin Hood: Men in Tights (1993)