**GIFT UNIVERSITY PROJECT**

**Title:**

**Movie Recommendation System based on Collaborative filtering using K nearest neighbour (ML Project).**

**Team:**

Naveed 191370197

Usman 191370189

Hammad 191370227

Shahzaib 191370191

Nimra 191370230

**Submitted to : DR. Qamar Askari Shah**

**SUMMARY**

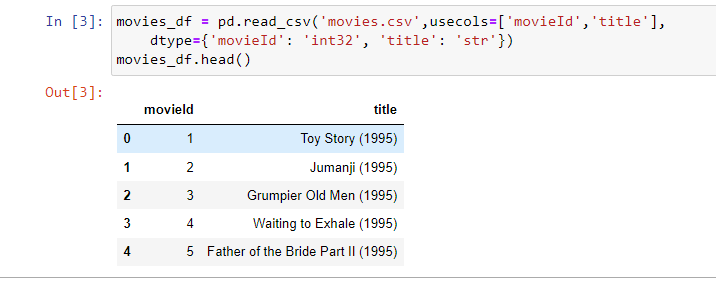
This code implements a movie **Recommendation System** using **collaborative filtering** with the **k-nearest neighbors** algorithm. The dataset used in this code includes movie ratings provided by users. This code performs a collaborative filtering-based recommendation system on a movie rating dataset using the k-Nearest Neighbors (k-NN) algorithm. The dataset consists of two CSV files, one for movie titles and IDs and another for user ratings of movies.

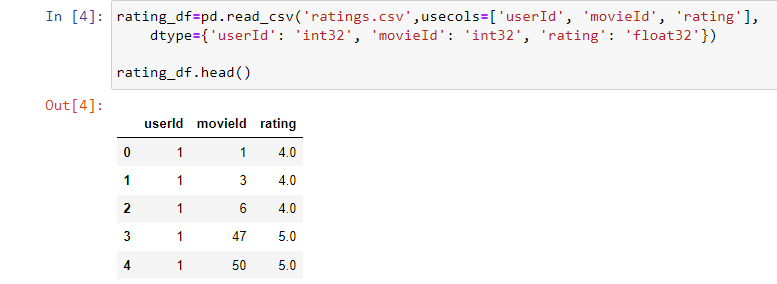
The code first loads the necessary packages and data using Pandas and NumPy libraries. Then, it performs some data preprocessing steps to filter out movies with low popularity ratings (total rating counts) and transform the user ratings into a matrix representation suitable for the k-NN algorithm.

The k-NN algorithm is applied using the cosine similarity metric and a brute force algorithm. Then, for a random movie index in the filtered dataset, the four nearest neighbors are identified based on user rating similarities:

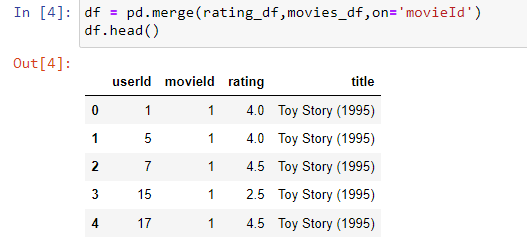
**Stepwise description**

* Read in two CSV files, **movies.csv** and **ratings.csv** containing movie information and ratings information, respectively, using **pandas**.

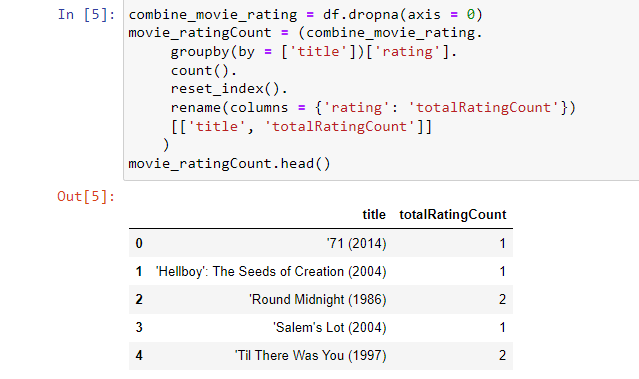




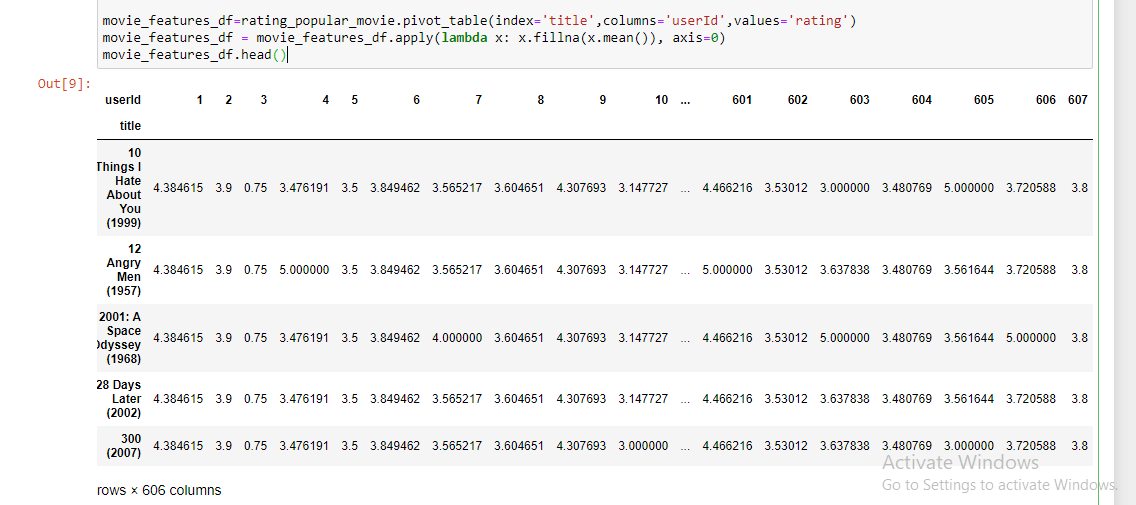
* Merge the two dataframes to create a single dataframe containing movie titles and corresponding ratings given by each user.



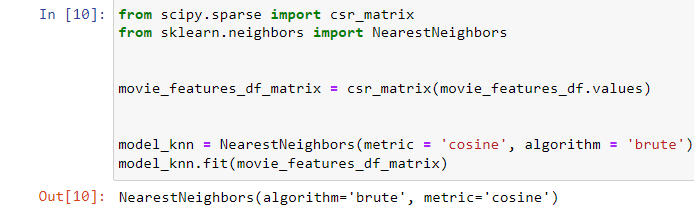
* Remove any rows with missing values and counting the number of ratings each movie has received and filter out movies that have received less than a certain threshold number of ratings.



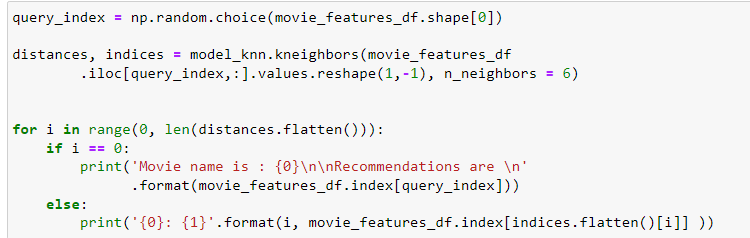
* The pivot\_table method is used to transform the data such that movie titles are listed as rows, user IDs are listed as columns, and the corresponding ratings are the values in the table.
* The second line of code is then filling any missing values in the DataFrame using the mean value of the non-missing values in the same column. This is done using the apply method with the lambda function that fills missing values with the column mean. Finally, the head method is used to display the first few rows of the resulting DataFrame.



* Fit a nearest neighbors model using the **cosine similarity** metric and the brute force algorithm on the sparse matrix.

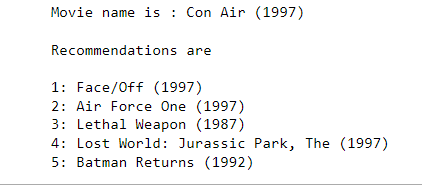


* Randomly choose a movie from the dataset and find the k-nearest neighbors using the fitted model.

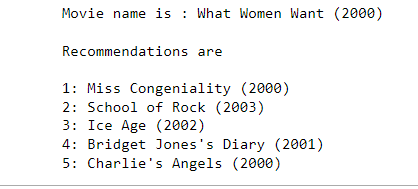


* Print the recommended movies to the user based on their chosen movie.

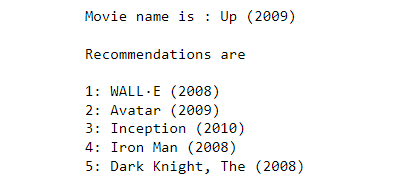
**Result 1 :**



**Result 2 :**



**Result 3 :**



It is important to note that there are no labels or ground truth data to measure the accuracy, precision, recall, and F1 score of this code. These metrics are commonly used in supervised learning tasks where there is a known ground truth to compare the model's predictions against

.

In this case, the recommendation system is unsupervised, and **there is no clear way to measure its performance** using these metrics.