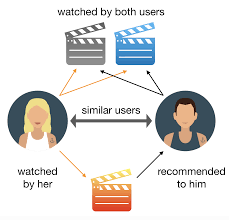
**Movie Recommender Systems**

A movie recommender system predicts the likelihood that a user might prefer to watch a movie not seen before based on past behaviour of same user or other similar users and user’s preferences.

*There are two types of recommender systems:*

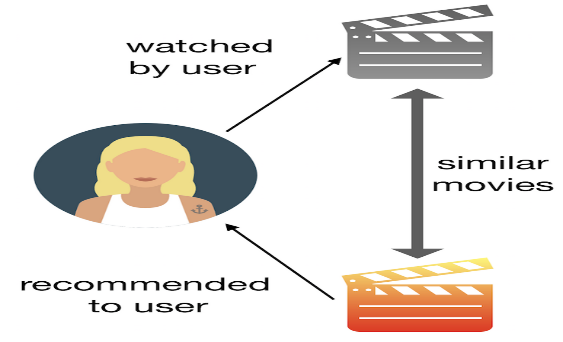
**User based:** It suggests new movies based on both users’ previously watched movies and experiences of other similar users. For example, if two users have watched few of the same set of movies, the one can be suggested unseen movie based on watchlist of other user. Basically, they can be considered neighbours in such case.



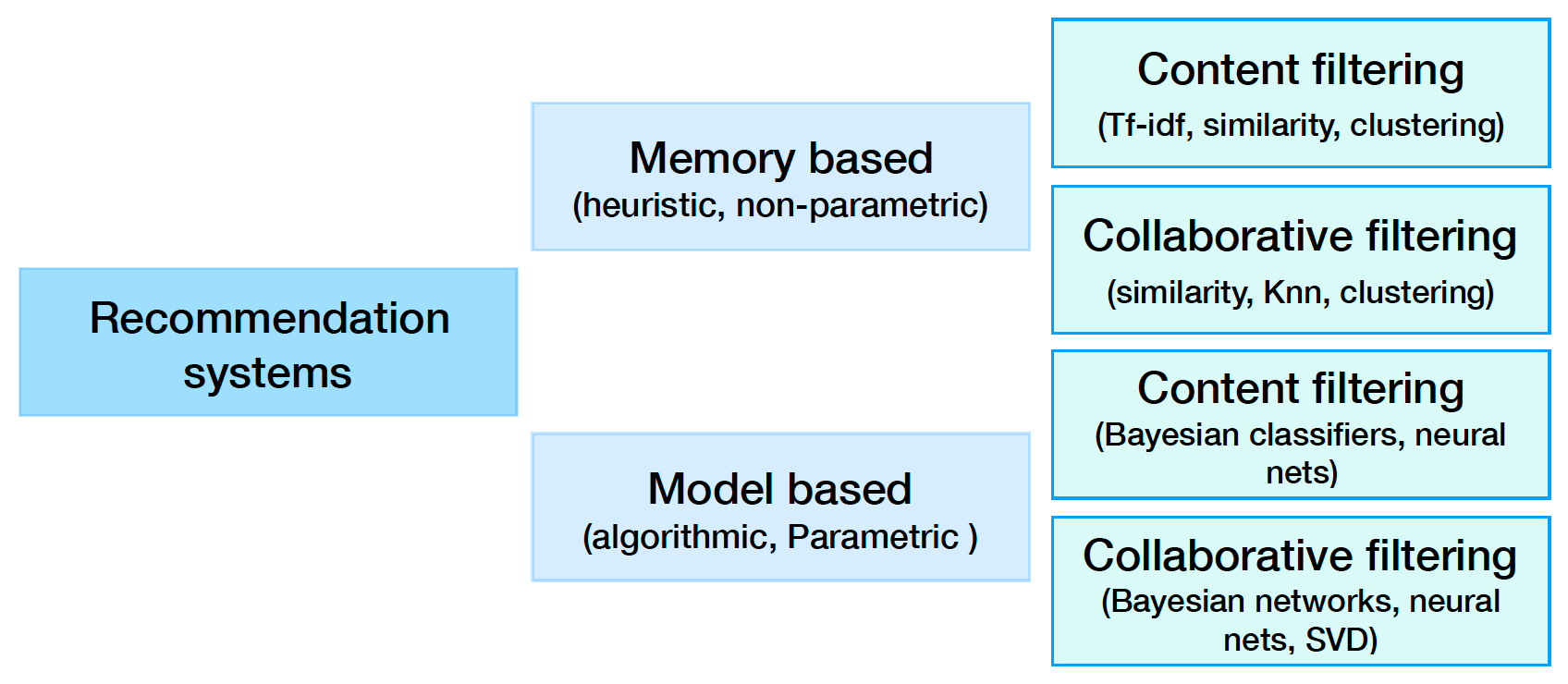
Limitations:

1. As we are comparing one use with all other users, it is computationally burdensome.
2. Also, individual user can change his/her preferences and start behaving differently than its neighbour which is not considered in user-based recommender system.

**Item based:** In this system, instead of finding relationship between users, used items like movies are grouped together or compared with each other. In such cases, even if the habit of user’s changes, this doesn’t impact recommendations. This is also computationally less challenging as the comparison is made between movies than users.



**Different types of recommendation Models**



As mentioned in above diagram, the recommendation systems can be broadly classified into parametric and non-parametric models. Each of these can have content based filtering or Collaborative filtering methods to provide recommendation.

Parametric models: The advantage is that they are simpler to explain and understand.

Non-parametric Models: These are based on matrix factorization to predict user’s rating for all unseen movies and then filter top movies. They are better in handling data sparsity issues.

**ABOUT MOVIELENS DATASET**

**Source:**

Kaggle: https://www.kaggle.com/grouplens/movielens-20m-dataset

**Context**

The datasets describe ratings and free-text tagging activities from MovieLens, a movie recommendation service. It contains 20000263 ratings and 465564 tag applications across 27278 movies. These data were created by 138493 users between January 09, 1995 and March 31, 2015. This dataset was generated on October 17, 2016 by GroupLens.

**Data Gathering process**

Users were selected at random for inclusion. All selected users had rated at least 20 movies.

**Content of data**

The data are contained in six files:

1. tag.csv that contains tags applied to movies by users:
2. rating.csv that contains ratings of movies by users:
3. movie.csv that contains movie information:
4. link.csv that contains identifiers(IMDB,TMDB) that can be used to link to other sources:
5. genome\_scores.csv that contains movie-tag(provided by users) relevance data:
6. genome\_tags.csv that contains tag descriptions

## Acknowledgements

To acknowledge use of the dataset in publications, please cite the following paper:

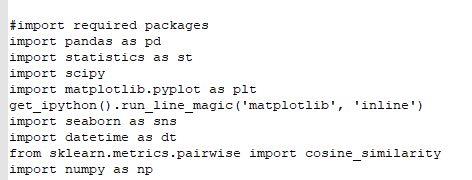
F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=<http://dx.doi.org/10.1145/2827872>

## Limitations in data:

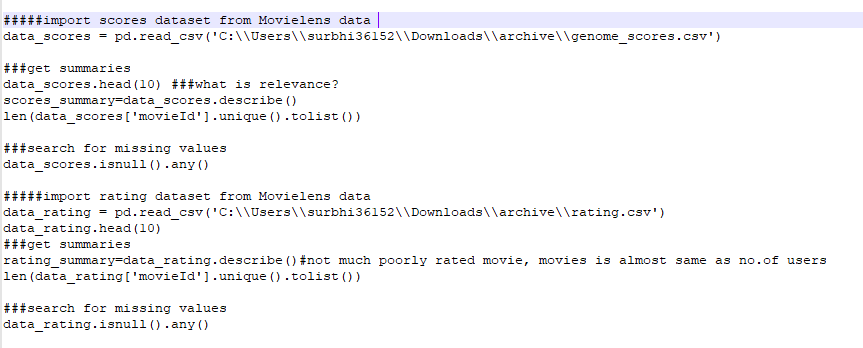
* No user demographic information is provided which can help in suggesting age specific movies.

**Approach taken for recommender system**

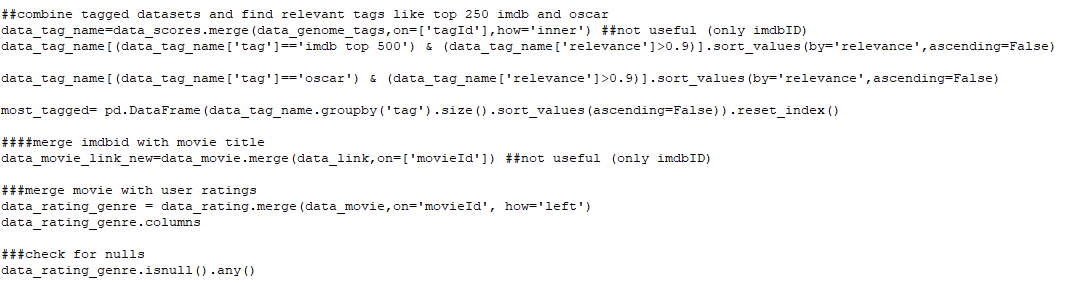
**Step 01: Importing relevant packages**

****

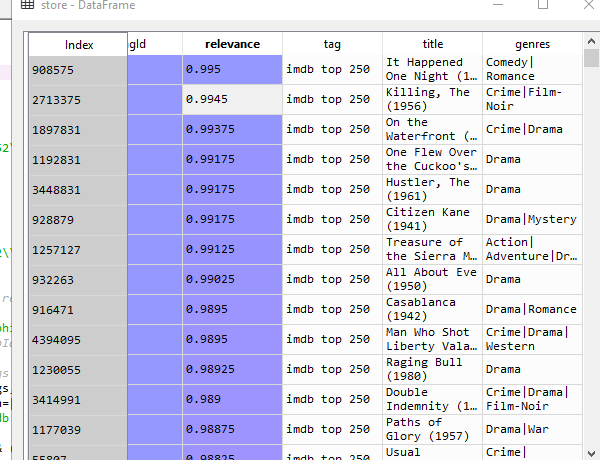
**Step 02: Loading Datasets and check for basic summaries and missing values**



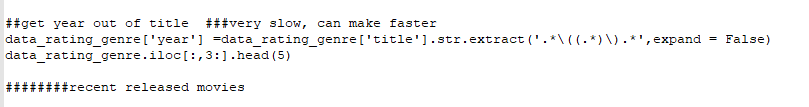
**Step 03: Merge datasets on similar columns to bring information together about movies. Also recommend common tags like imdb top 250 movies and Oscar nominated movies**

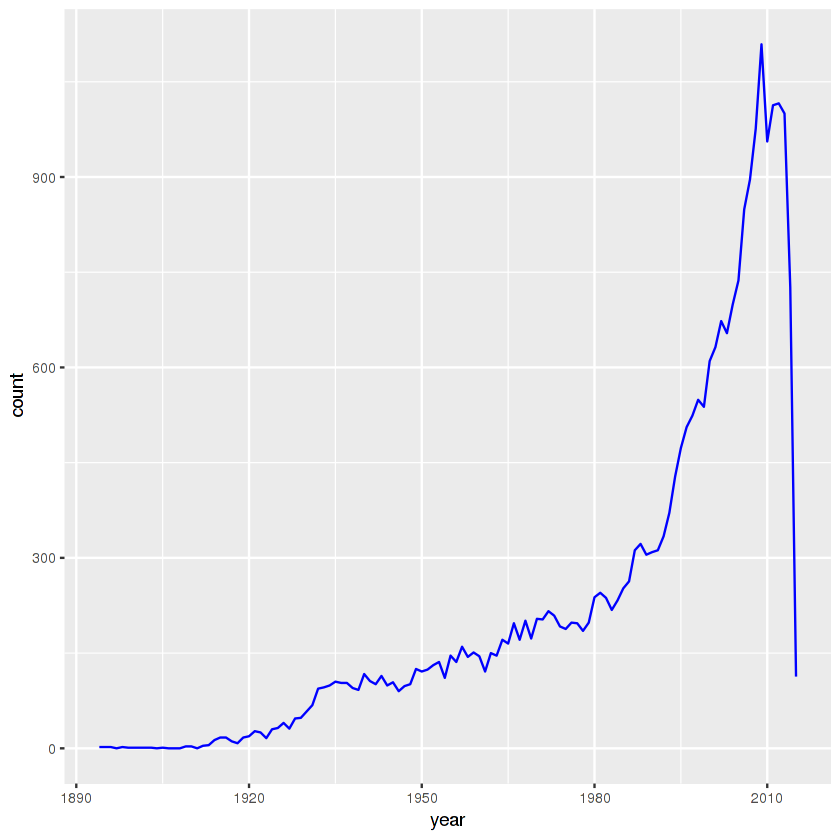


**Output:**

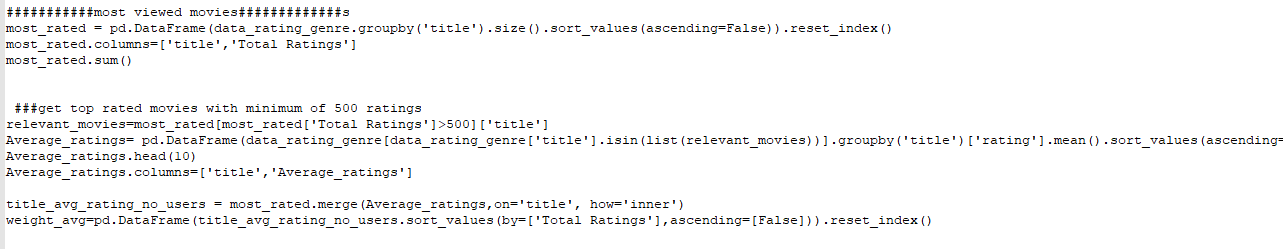


**Step 04: Data feature engineering to extract new value from existing columns.For example, extract year of movie from title to recommend recent released movies**



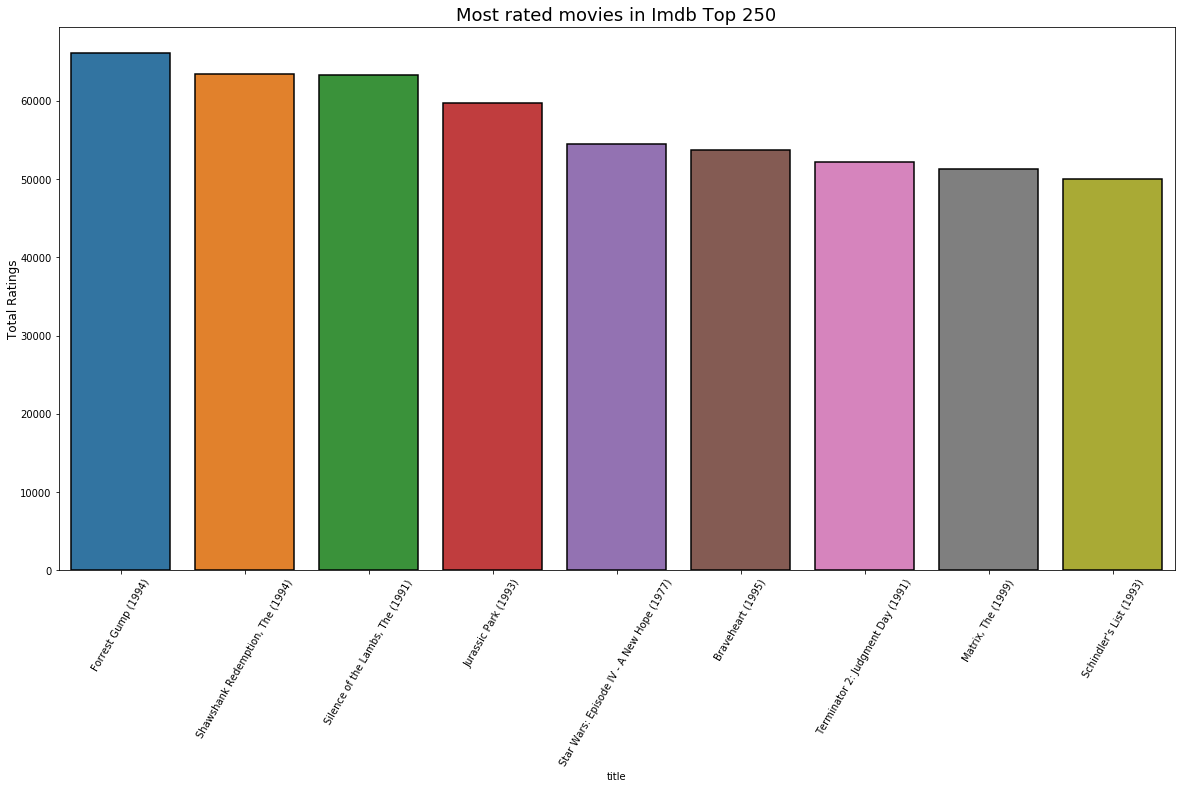


**Step 05: Basic EDA to get most viewed movies and highly rated movies(after removing movies with less than 500 ratings)**

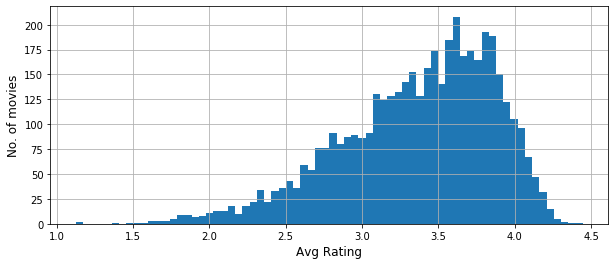


**Output:**

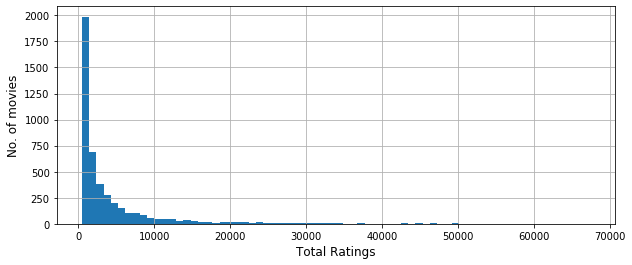
**IMDB top 250-** This gives list of top 10 most rated movies

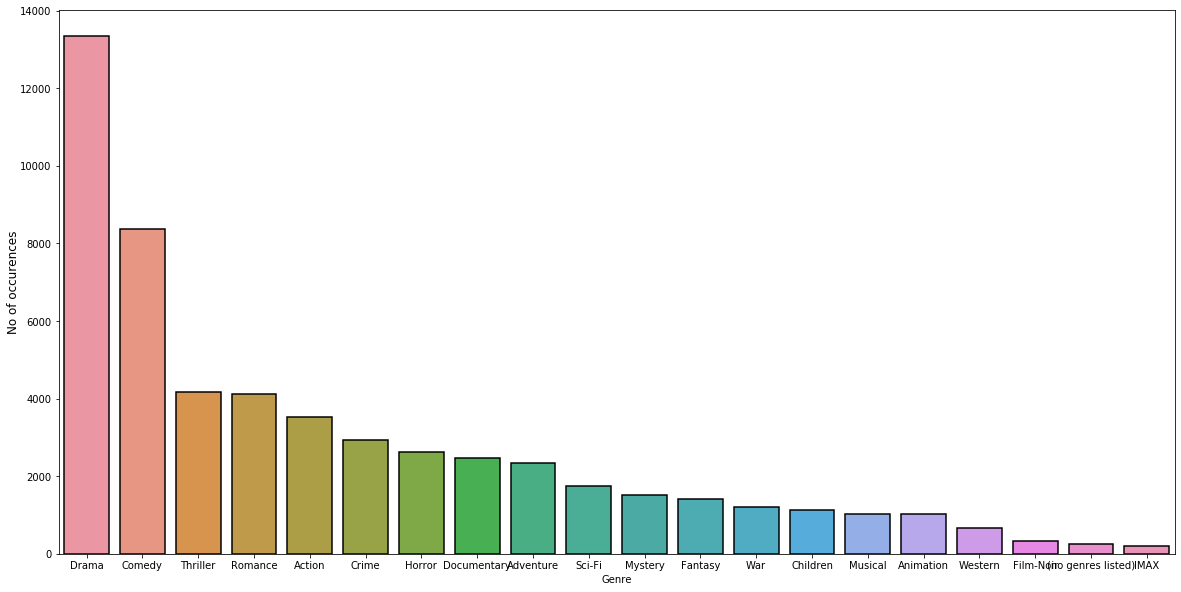


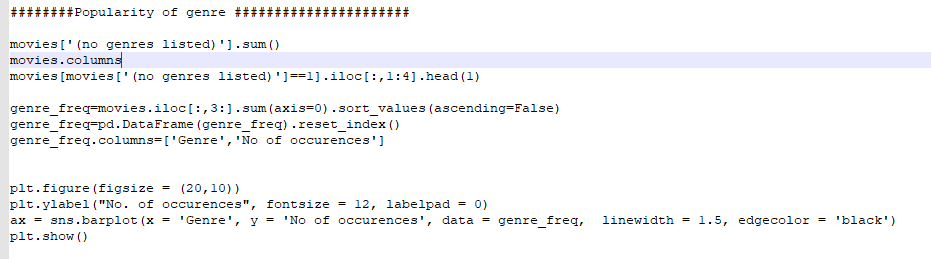
**Distribution of Average Ratings –** Giving idea about average rating as ~3.5 for most of the movies



**Distribution of total ratings for movies-** We can see most of the movies are rated by less than 10,000 users**.**

**Most popular genres:**



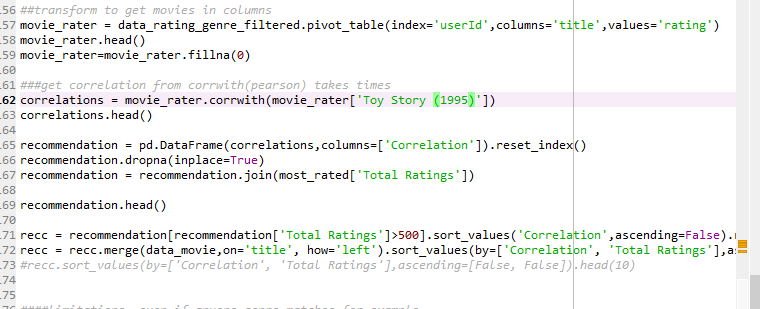


**Output:**

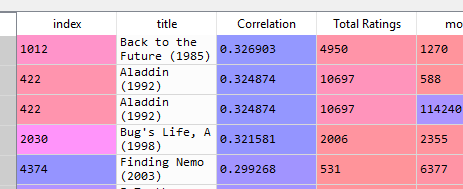
**Step 06:**

1. **First Recommendation is simple and based on ratings(item) correlation**

This method creates a matrix for all movies with ratings as values and finds correlation of all movies with movie already seen by user e.g. Toy Story (1995)



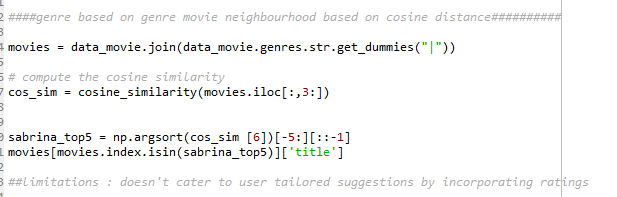
**The Output** comes out as similar rated other movies:



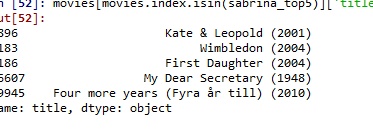
**Drawbacks:** This method doesn’t integrate other items and similar user behaviour.

1. **Another recommendation is again item based(genres) based on cosine similarity**

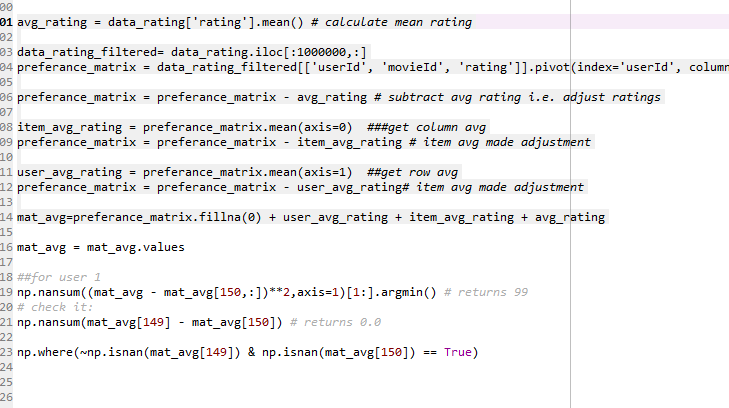
This method first does one hot encoding of genres in data and then calculates cosine distance between each pair of genres and recommends movies like genre seen previously by user. Here example is taken of k=6 (7th movie) i.e. Sabrina (genre comedy/romance)



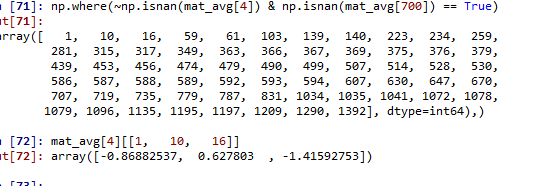
**The Output** recommends similar genre movies



1. **Another recommendation is similar user based collaborative filtering based on ratings.**



**The Output** recommends 11th movie which is yet not seen by user and liked by another user.



**Drawbacks:** It doesn’t consider sudden change in user behaviour.

**Future Scope**

1. Try parametric models by dividing data in train and test as data is more than 20Mn, which is more robust for training model e.g. SVM, other neural nets, matrix factorization
2. We can further explore usage of timestamps in data by exploring more on temporal patterns on user ratings and genres views.