**MOVIE RECOMDATION**

\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Table content**

1. **Introduction**
2. **Aim**
3. **Importance**
4. **Code with Explanation**
5. **What we achieve**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1.AIM**

## The purpose of a recommendation system basically is to search for content that would be interesting to an individual.

## . Recommendation systems are Artificial Intelligence based algorithms that skim through all possible options and create a customized list of items that are

## interesting and relevant to an individual. These results are based on their profile, search/browsing

## history, what other people with similar traits/demographics are watching, and how likely you are

## to watch those movies.

## Your aim will be to recommend similar movies if a type of movie is given.

**2.INTRODUCTION**

* This report presents a movie recommendation model that is designed to suggest personalized movie recommendations to users based on their past movie preferences. The model is developed using a collaborative filtering technique that utilizes user-movie interaction data to identify patterns and make movie recommendations. The main objective of this model is to provide users with a personalized and engaging movie watching experience that matches their interests and preferences.
* The report begins with an overview of the movie recommendation system and its significance in the entertainment industry. It then discusses the data preprocessing and feature engineering techniques used to prepare the data for the model. The collaborative filtering approach is then explained in detail, including the similarity metrics and algorithms used for the model.
* The model evaluation process is also discussed, highlighting the metrics used to assess the performance of the model. The report then concludes with a summary of the findings, limitations, and future directions for the model.
* Overall, this report provides a comprehensive understanding of the movie recommendation model and its effectiveness in providing personalized movie recommendations to users.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3.IMPORTANCE**

* The movie recommendation system plays a crucial role in the entertainment industry, particularly in the digital era, where streaming platforms dominate the market. With a vast amount of movie content available, users often struggle to find the movies that match their interests and preferences. This leads to frustration and decreased user engagement with the platform.
* Movie recommendation systems address this problem by providing personalized movie recommendations to users. By analyzing the user's past movie preferences, the system can suggest movies that are likely to match their interests. This improves the user experience, increases user engagement, and promotes user loyalty to the platform.
* Moreover, movie recommendation systems have become a key driver of business growth for streaming platforms. By providing personalized recommendations, streaming platforms can increase user retention, attract new users, and generate more revenue through increased movie views.
* In conclusion, movie recommendation models are critical for providing users with a personalized and engaging movie watching experience while also driving business growth for streaming platforms.

**4.CODE WITH EXPLANATION**

Our code starts by importing all the necessary files and libraries on the top so we can keep the track of what we imported and can use directly.

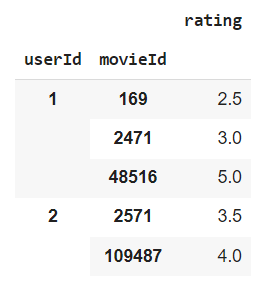
**Data Preprocessing:**

The first and most important part of any machine learning algorithm is to preprocess the data to remove all the unwanted, missing or outlier data points which will not affect out further performance of the system we build.

For pre processing we first remove timestamp from tags.csv as timestamp is not necessary for our movie recommendation. We only want which movies the user has watched that does depends on when he/she has watched those movies.

**Data analyses:**

Now we analyze the data before moving ahead. In analyses of the data we firstly group the userid with the movieIds he/she has watched including the ratings the user gave in order to get a better idea of the data. This analyses helps us to analyze which movies the user watches and likes based on the ratings he/she gives.

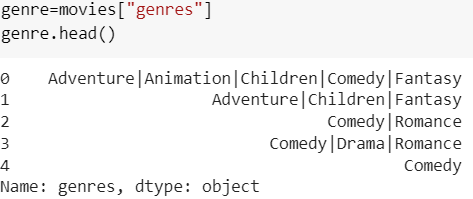


Moving ahead we find out how many unique movies are there in the dataset we have and also how many unique users are there watching those movies. We did this part to get to know how many users and movies are we dealing with and how we going to classify them based on size of data.



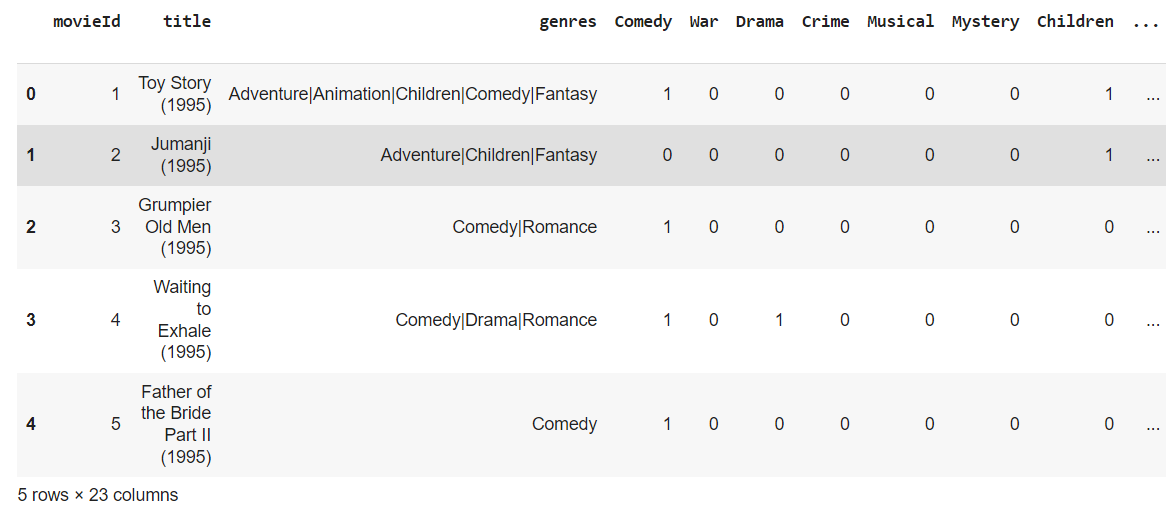
After user analyses we thought of analyzing the data based on the genre given.

Genre based analyses is most important as we want to find out the interest of the user and recommend further movies based on those interests only. For that we first find out how many main genres are there:

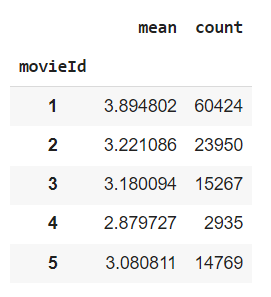


Then we list all the genres present in the dataset to know all the available options of genres from which the user is going to watch movies. We also print the count of all the genres in the list which comes out to be 20.

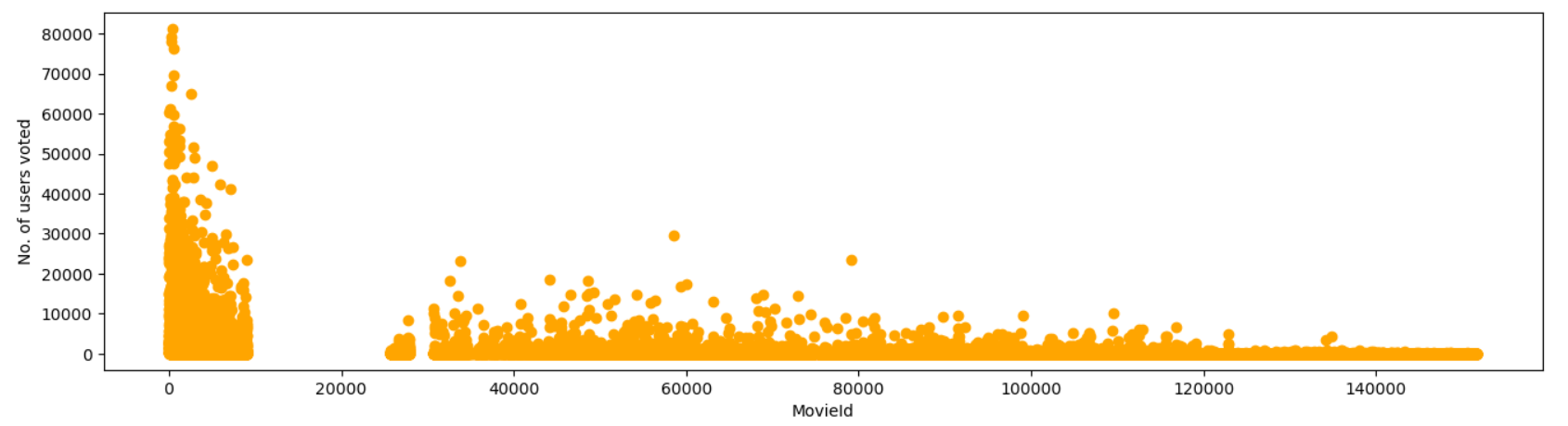
Now for better analyses we created a matrix data frame which finds out individual genre for each of the movie. This helps a lot in further analyses of the data and recommend better movies based on the genre. This matrix data set first states all the genres the movie is based on and then for each genre it increases the count like if the movie is a horror comedy, then it gives horror and comedy as 1 and every else genre as 0.



After all this genre-based classification, we thought that we should also include the factor of rating so we can recommend better based on the user ratings. More rated movies have more probabilities to be liked by the user we recommending the movie. So, for this we first find out the avg rating of each movie and also the no. of users giving ratings for that particular movie.

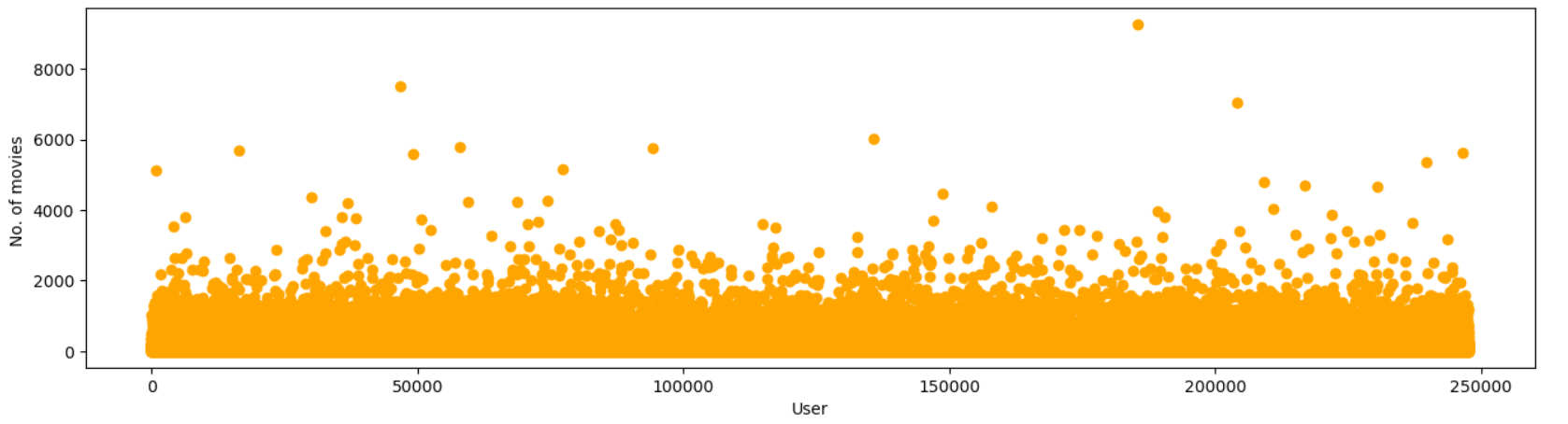
****

Now for better visualization we also create a scatter plot between the movieId and no of users voting for that movie including the ids of the users.

****

This plot helps us the analyze that which movie ids are not rated by any of the users or are less rated by the users so we can recommend more of the famous movies instead of recommending non rated movies

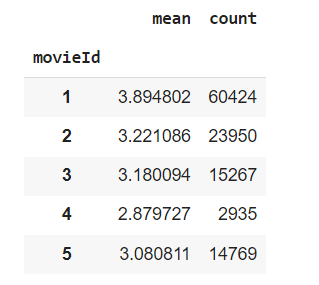
With this movie plot we thought of reversing the plot like plot between userid and movies rated with movie id. So, if we can find out users which don’t rate the movies at all and we can recommend movies to them in a certain different way rather than just based on their past ratings.



Then we thought of finding out avg no for how many users rate the famous movies. For that we set the data it gives the no of ratings at least 65% of the movies have and this number comes out to be 43. Which means on an avg each movie gets rated around 40 times by the users watching it.

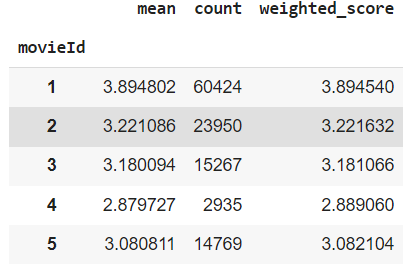
We also find out what is the avg rating of all the movies being watched by the users to get a better hold on the ratings given by the users. This avg rating comes out to be 3.526.

Now we again set a dataframe which holds the movie id with their mean ratings and the count of users rating that movie.



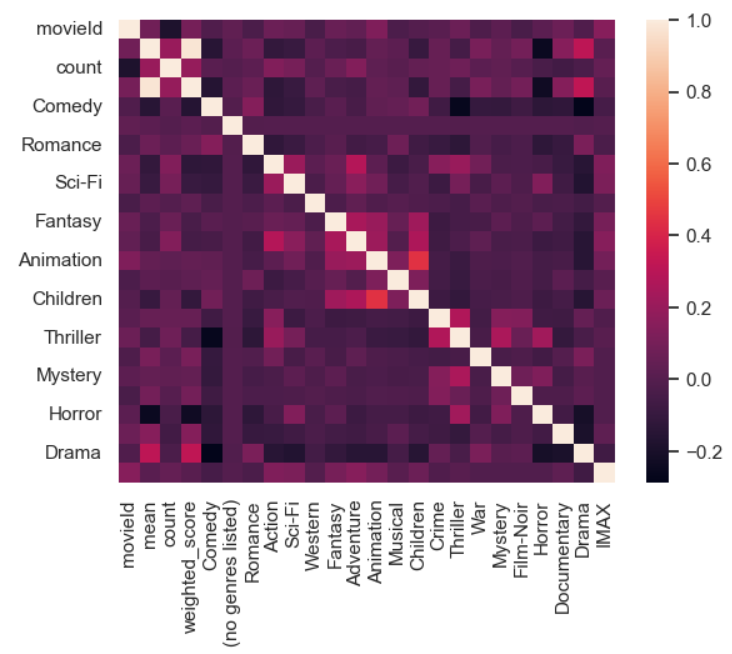
**Working on solution:**

Now when all the analyses are done for the data, we move on to find a solution to recommend the best movies to the users. For this we first create a data frame with weighted ratings. This means that the movies with more count ratings will have more weight with respect to those having less count of ratings. We also use the formulae used by IMDB (one of the famous movie ranking platforms) to calculate the weighted avg for the movies. This weighted score will keep the more user famous movies as the same which removing least famous movies from the list which will help us to recommend movies in a much better way.

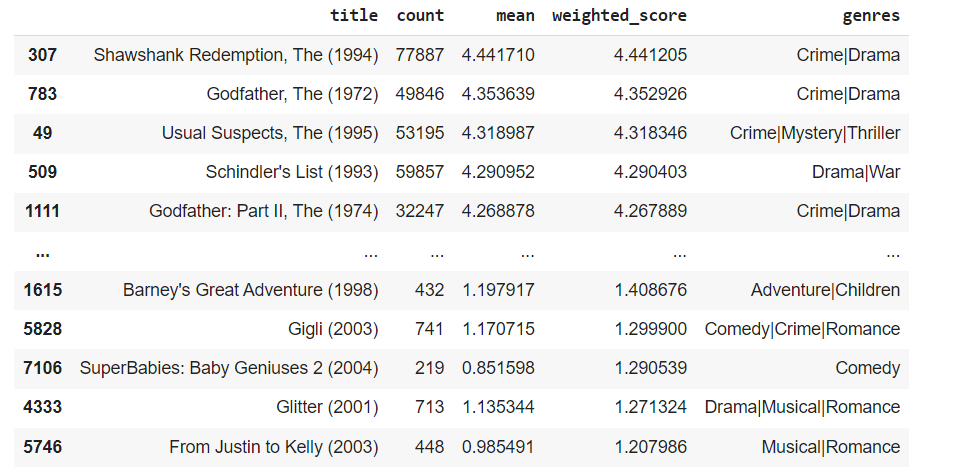


Now we merge these weighted scores for each movie with the dataframe we initially created with genre of each movie so that we can have everything of the analyses in one dataframe.

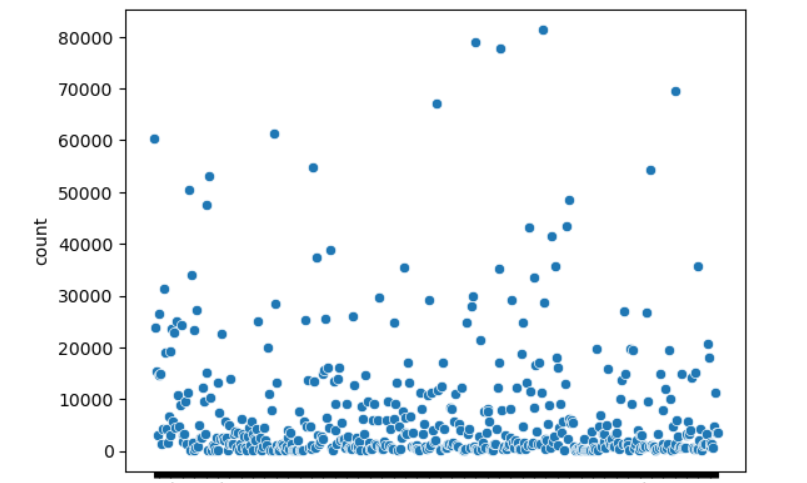
For better representation of this data, we also create heat map of the data frame.



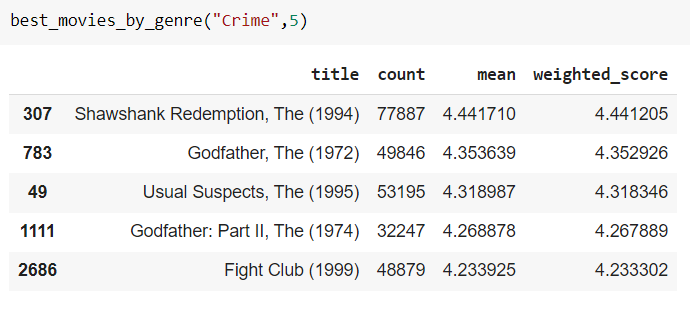
Now we sort the data based on the weighted score these movies get. This part helps us to sort the data in the best fashion for recommendation, the most user famous movies on the top and the least famous movies on the bottom. And we also do this in the common data frame so that we can also incorporate the genre that the user like or want to watch movie from that genre at any times for better results.



We also create a scatter plot for this.



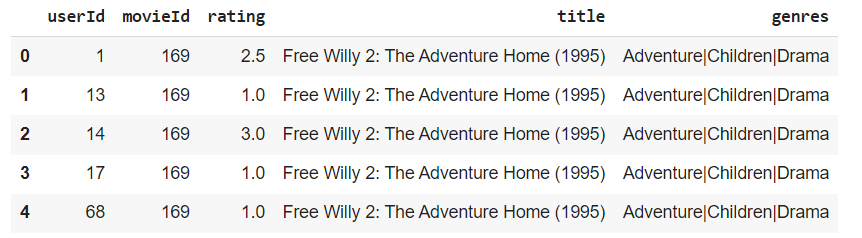
Now as our data frame is already set up sorted by most rated and genre based collection, we can simply search for a particular genre and find out the top movies based on that genre from the data we have created. By this a part of out recommendation system is already complete, we can classify movies based on genre and recommend best movies based on that genre.



**#First Recommendation done**

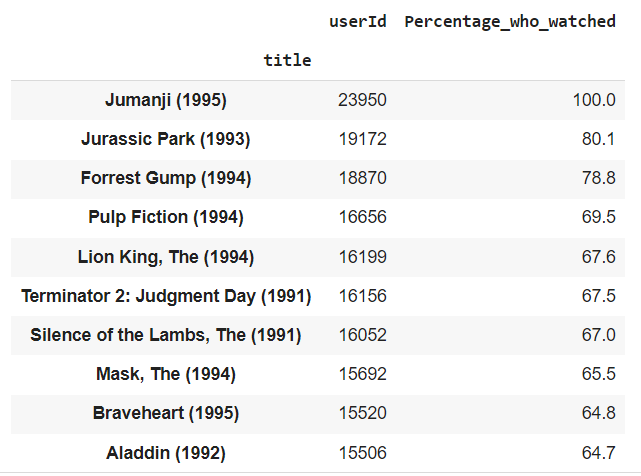
But this was not our only task to do, there could also be users which don’t want to watch movies based on the same genre or they don’t want movie classification based on only genre. So we further started finding solution to recommend movies based on the other viewer choice like if a user watches a movie and moves to some other movies and also finds that good (gives good ratings) then that new movie can also be good for a new user watching the old movie.

Now for working on this we again start by merging the rating and user id and movie id.



Now we first work on to find out the top 10 movies of the users which saw a particular movie in order to recommend the movie which is most viewed by other users. This was all done by using only pandas without any algorithm as there was only user id and ratings which can be easily classified based on more rating or less rating. So till this point we were able to recommend movie in a clean simple way based on what type of genre the user likes or what other users view based on a particular movie the user views. Here we showed percentage of people who watch the particular movie. Like let say the movie is x so for that x movie the percentage of people who watch that movie is gonna be 100, now as we go ahead we find out a movie which has the max percentage of watching so we will recommend that movie next similarly we find out next 10 movies to recommend which can be good for the user watching the movie x.



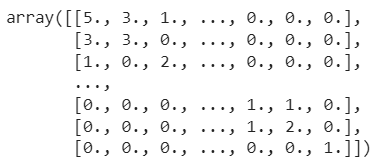


**#Movie Recommendation 2 Done**

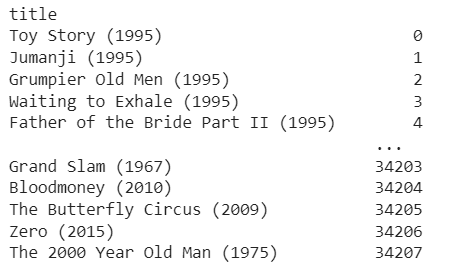
Now we also thought this is not going to be enough. We can also classify movies based on the content the movie has. This was not going to be simple as we had to analyze each movie content based on all the genre stated for that movie so that we can recommend movies in a much better way based on the content or you can say advanced genre based.

For this we started by getting the list with only genres like any X genre and other such columns.

Now we calculate the cosine similartity matrix for this particular movie recommendation method. We find out the similarity matrix for movies based on the content



Now we create a series of all the movie titles with their respective movie ids

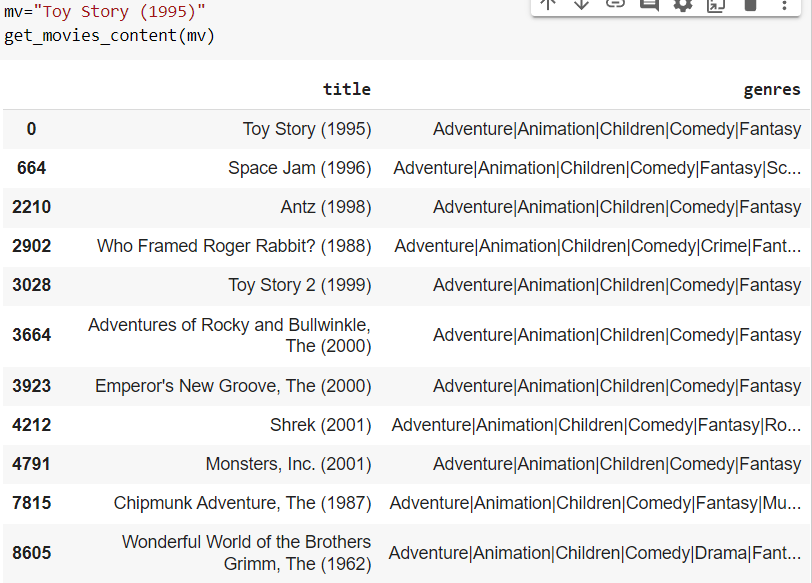


Now as we created the cosine similarity matrix we start finding out the most similar movie.

Cosine simlarity is a matrix that contains simlarity scores between all the pairs of the itmes in a dataset. Each row and column in the matrix represents an item and the cell at the intersection of a row and column represents the cosine similarity score between corresponding pair of items.

Now for this we first enter any movie name we want to find the similarity of, we look for that particular movie in the title list and gather the movie id from it. Then we go to the cosine similarity matrix and we already have that sorted in descending order based on the similarity scores we are getting. So we just have to find out to top similar movies to it. From that matrix we find out 10 most similar movie ids and then from title dictionary we story their movie titles for further display.

The main resason of using the cosine simlarity matric is as this is the most common method to find out similarity between 2 vectors and in our case these vectors are the genre vectors of the movies.

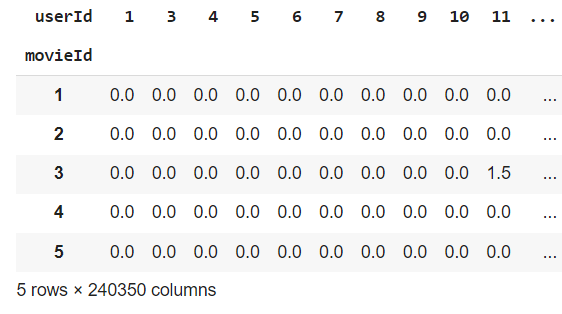


**#Movie Recommendation way 3 done**

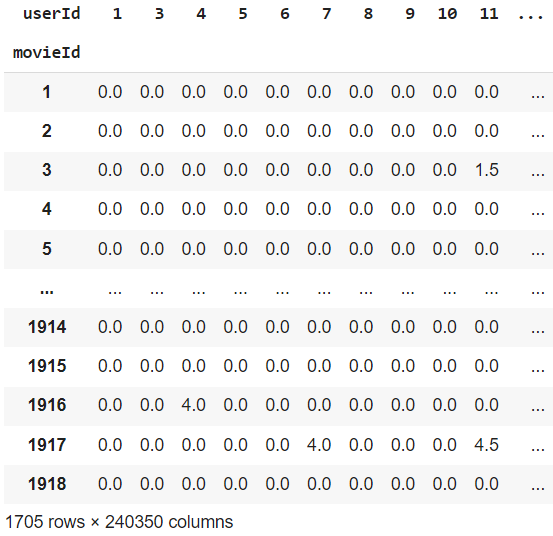
As in all the movie recommendation methods we done earlier we only used pandas dataframe to sort data in a way that it gives the best movie recommendation. So we thought of using something different for movie recommendation now. For this we thought of applying Nearest Neighbor algorithm.

Now for this we first reduced the data size. For reducing the data size we set a min count parameter for the rating count. So all the movies having the no of users rating those movies less than a particular min count those movies are removed from the data set.

Then we create a matrix table with movieids on the rows and userids in the columns and replaced NAN values with 0 if there are any.

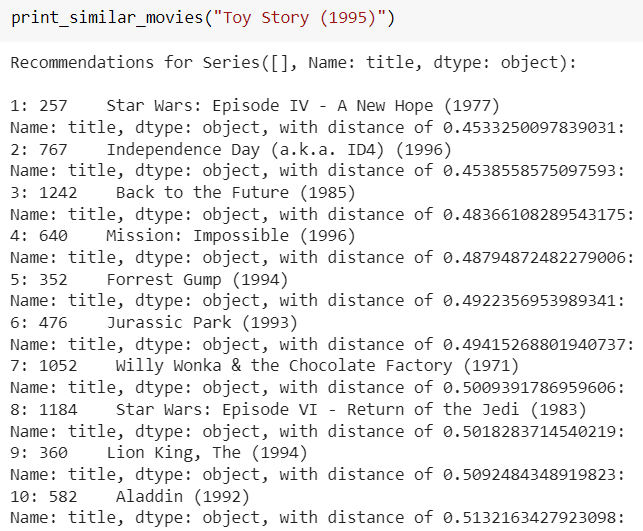


Now we set up with nearest neighbors algo with matric as cosine and brute method. We fit the updated shortened data set and then we find out the view of the fitted data set



Now we write a function to get the 10 movies as suggested by the nearest neighbours algorithm.

For this we put in the title, use the specific movieId to put in a query for nearest neighbours, find out the closest distance movies based on the neaerst neighbour classification and cosine factor and then we show out those closest movies. The result we get is as follows:



Here the distances are also shown for a better understanding of how it recommends the movies based on the least distanve from the selected movie.

**#Movie recommendation by nearest neighbours done**

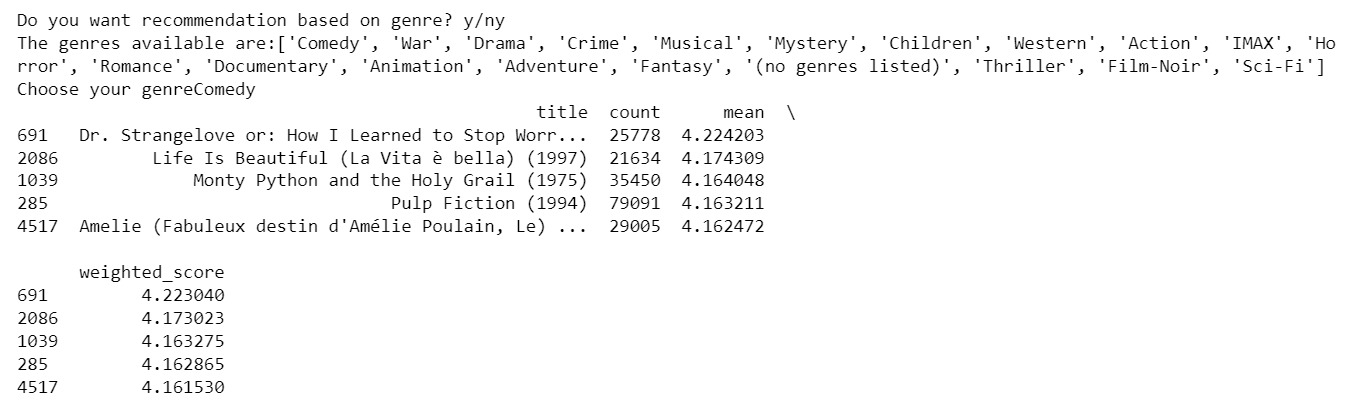
**5.What We Achieve**

Now after completing all these classification techniques and methods we tried to compile them together in 1 place so that the user can be directly given option to select from either genre and get most famous movie recommendation from that genre or he can also ask for a particular movie and get all the recommendations based on the techniques we have used above that are based on other viewers viewing, genre, simple rating method and also the nearest neighbour algorithm we used.

From overall everything we obtain a function which can simply give movie recommendations based on your watch history, the closest content based movie you want to watch, famous genre based classified movies and even content based movies that the user asks.

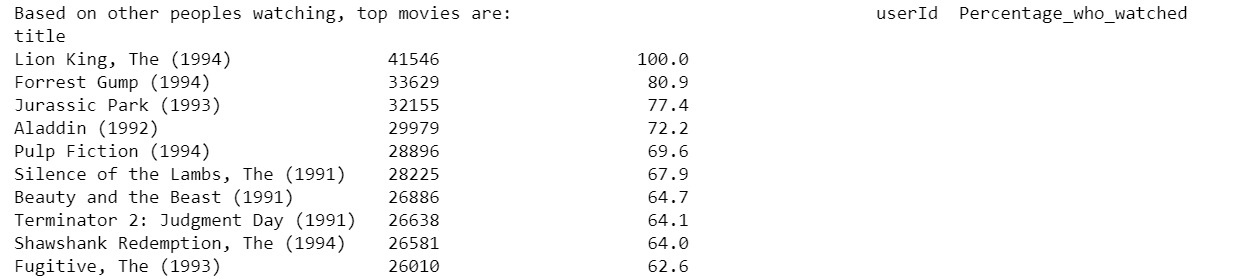
Here are some output snippets of the function we formed:

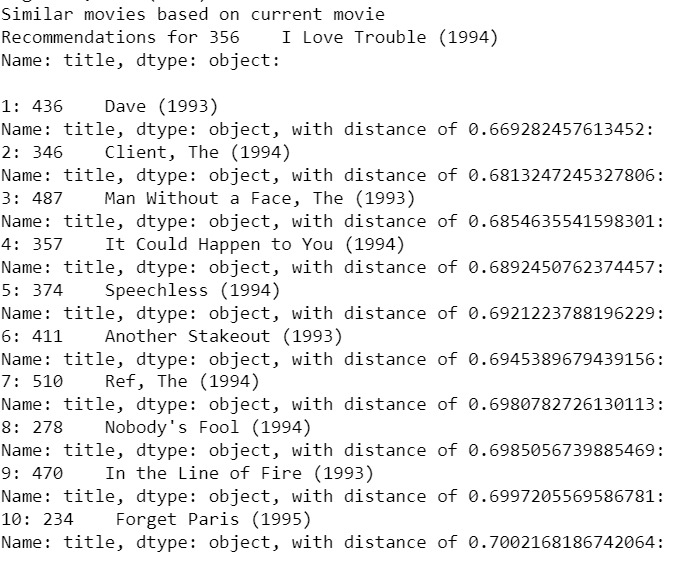
First genre based classification:



Now non genre based classification:







All these recomomendations are given together rather that asking for different type of classifications. This gives the user all types of options to choose from for the next movie he/she wants to watch and enjoy.