Movie Recommendation System

Subject: Statistics and Machine Learning

Group Number: 4

Members:

Avadhoot Pawaskar

Kaustubh Joshi

Mayuri More

Mihir Jasmin Rathod

Parvathi Khodanpur

Rutukesh Gavas

Abstract

A movie recommendation is important in our social life due to its strength in providing enhanced entertainment. Such a system can suggest a set of movies to users based on their interest, or the popularities of the movies. Although, a set of movie recommendation systems have been proposed, most of these either cannot recommend a movie to the existing users efficiently or to a new user by any means. The rapid growth of data collection has led to a new era of information. Data is being used to create more efficient systems and this is where Recommendation Systems come into play. Recommendation Systems are a type of information filtering systems as they improve the quality of search results and provides items that are more relevant to the search item or are related to the search history of the user. They are used to predict the rating or preference that a user would give to an item. Almost every major tech company has applied them in some form or the other: Amazon uses it to suggest products to customers, YouTube uses it to decide which video to play next on auto play, and Facebook uses it to recommend pages to like and people to follow. Moreover, companies like Netflix and Spotify depend highly on the effectiveness of their recommendation engines for their business and success.

In this project, we propose a movie recommendation system that has the ability to recommend movies to a new user as well as the others. It mines movie databases to collect all the important information, such as, popularity and attractiveness, required for recommendation. Experimental studies on the real data reveal the efficiency and effectiveness of the proposed system. We propose our project by using three filtering methods which help us recommend movies to the user in an efficient manner.

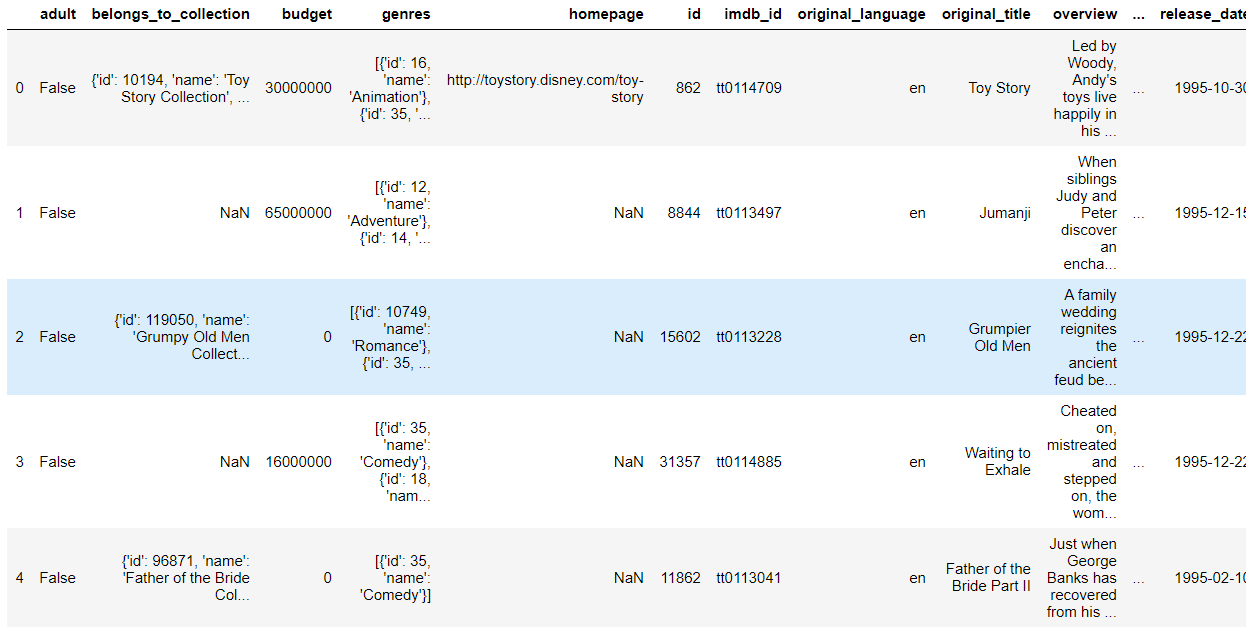
**MATERIALS AND METHODS**

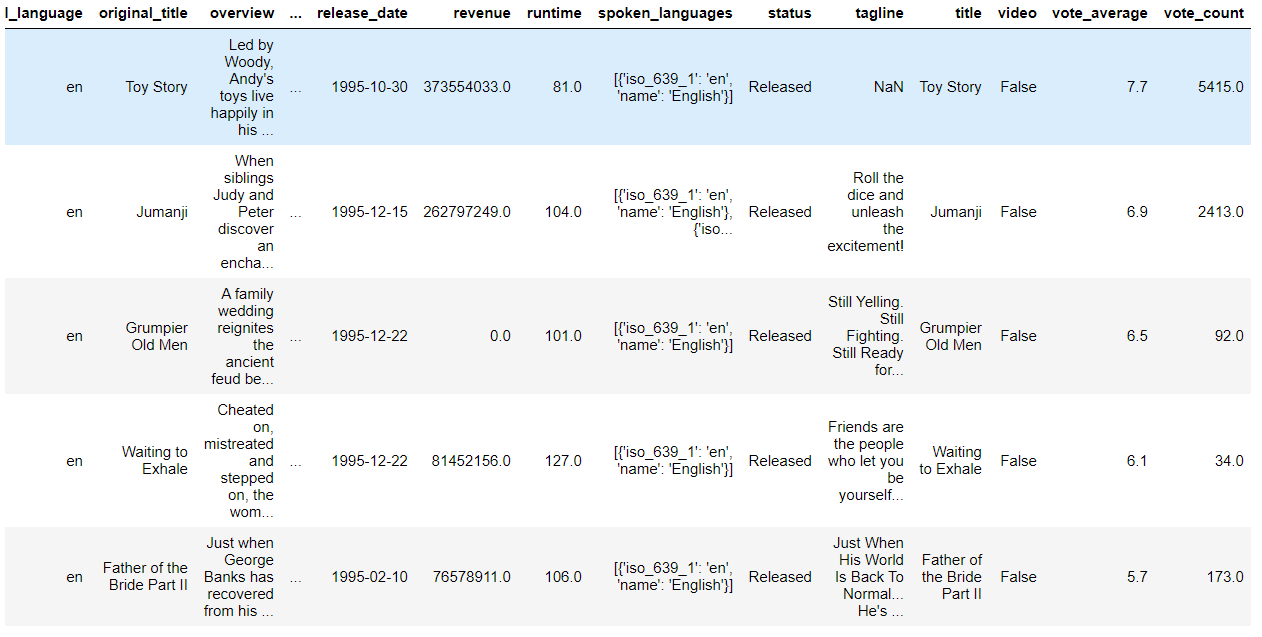
**Dataset Description:**

The dataset has been taken from Kaggle, which comprises of:

Dataset has the following features: -

* adult – the type of audience.
* budget - The budget in which the movie was made.
* genre - The genre of the movie, Action, Comedy, Thriller etc.
* homepage - A link to the homepage of the movie.
* id - the movie\_id as in the first dataset.
* imdb\_id - imdb\_id from the imdb website.
* original\_language - The language in which the movie was made.
* original\_title - The title of the movie before translation or adaptation.
* overview - A brief description of the movie.
* popularity - A numeric quantity specifying the movie popularity.
* production\_companies - The production house of the movie.
* production\_countries - The country in which it was produced.
* release\_date - The date on which it was released.
* revenue - The worldwide revenue generated by the movie.
* runtime - The running time of the movie in minutes.
* spoken\_languages - the languages in which the movie is being released.
* vote\_average – average of the total votes.
* vote\_count – total count for a movie.





**Tools and Techniques:**

* Jupyter Notebooks is used to executed the following project. The language used to execute the code is Python.
* Below are the filtering methods that we have used to recommend movies:

1. Demographic Filtering- They offer generalized recommendations to every user, based on movie popularity and/or genre. The System recommends the same movies to users with similar demographic features. Since each user is different, this approach is considered to be too simple. The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience.

2. Content Based Filtering- They suggest similar items based on a particular item. This system uses item metadata, such as genre, director, description, actors, etc. for movies, to make these recommendations. The general idea behind these recommender systems is that if a person liked a particular item, he or she will also like an item that is similar to it.

3. Collaborative Filtering- This system matches persons with similar interests and provides recommendations based on this matching. Collaborative filters do not require item metadata like its content-based counterparts.

Simple Recommender:

The System recommends the same movies to users with similar demographic features. Since each user is different, this approach is considered to be too simple. The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience. This model does not give personalized recommendations based on the user.

The implementation of this model is extremely trivial. All we have to do is sort our movies based on ratings and popularity and display the top movies of our list. As an added step, we can pass in a genre argument to get the top movies of a particular genre.

Before getting started with this -

* we need a metric to score or rate movie
* Calculate the score for every movie
* Sort the scores and recommend the best rated movie to the users.

We can use the average ratings of the movie as the score but using this won't be fair enough since a movie with 8.9 average rating and only 3 votes cannot be considered better than the movie with 7.8 as average rating but 40 votes. So, I'll be using IMDB's weighted rating (wr) which is given as: -



where,

v is the number of votes for the movie;

m is the minimum votes required to be listed in the chart;

R is the average rating of the movie; And

C is the mean vote across the whole report

To find an appropriate value for m, the minimum votes required to be listed in the chart. We will use 95th percentile as our cut-off. In other words, for a movie to feature in the charts, it must have more votes than at least 95% of the movies in the list.

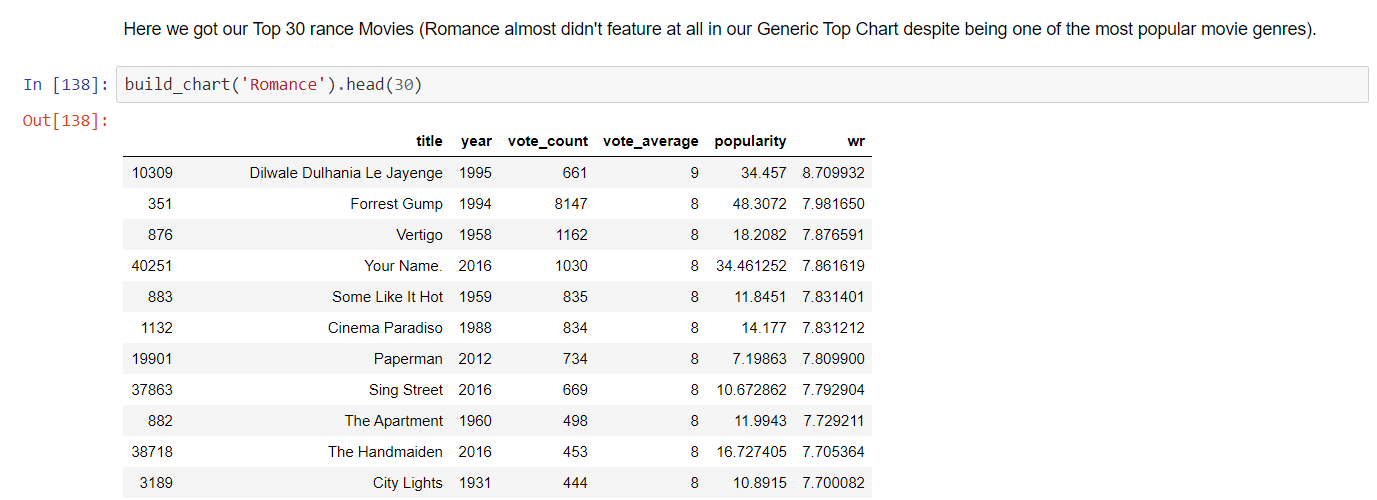
Output that we achieve is this:

A screenshot of a social media post

Description automatically generated

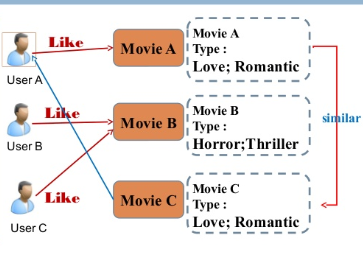
We got Inception, The Dark Knight and Interstellar at the very top of our chart.

We then constructed our function that builds charts for particular genres. For this, we will use relax our default conditions to the 80th percentile instead of 95.



**Content Based Filtering**

In this recommender system the content of the movie (overview, cast, crew, keyword, tagline etc) is used to find its similarity with other movies. Then the movies that are most likely to be similar are recommended.



This system uses item metadata, such as genre, director, description, actors, etc. for movies, to make these recommendations. The general idea behind these recommender systems is that if a person liked item, he or she will also like an item that is similar to it.

We built two Content Based Recommenders based on:

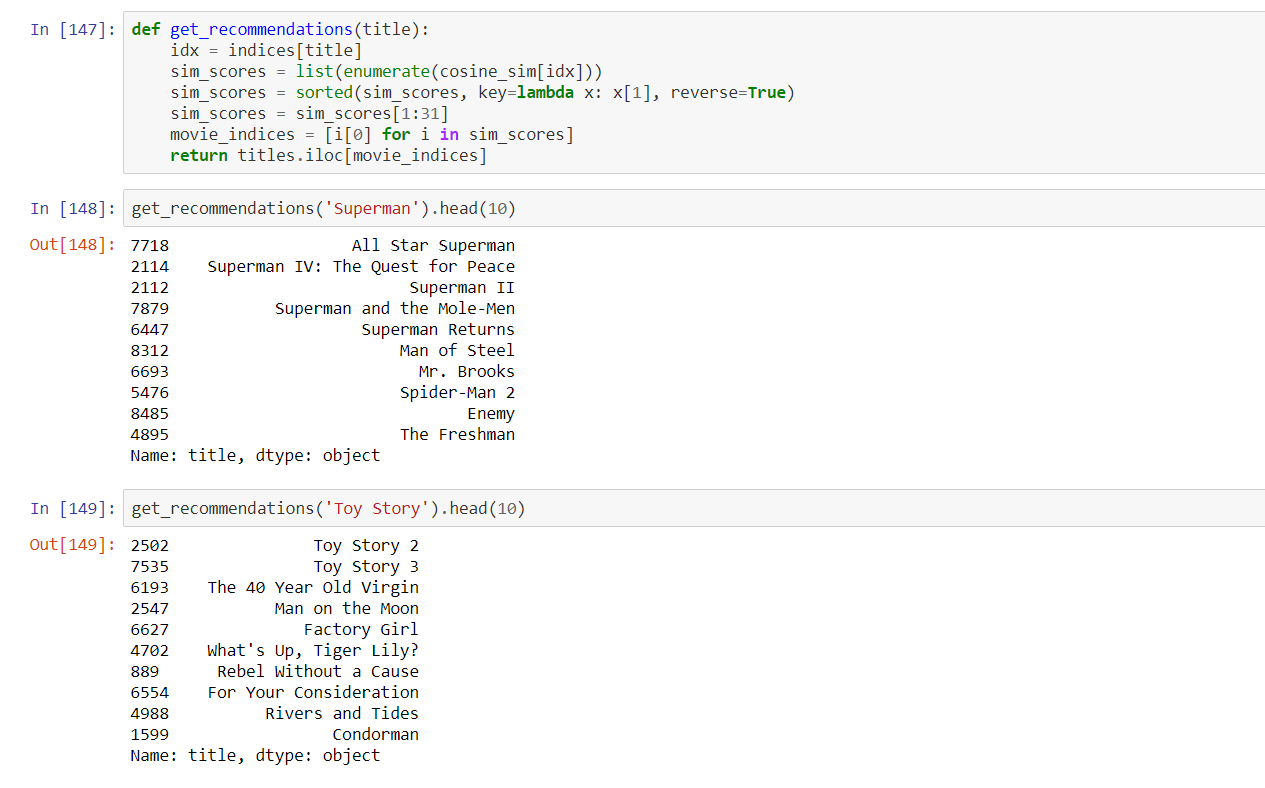
1.Movie Overviews and Taglines

2.Movie Cast, Crew, Keywords and Genre

We used the Cosine Similarity to calculate a numeric quantity that denotes the similarity between two movies. Mathematically, it is defined as follows: cosine(x,y)=(x.y⊺)/(||x||.||y||)

Movie Description Based Recommender

We tried to build a recommender using movie descriptions and taglines.



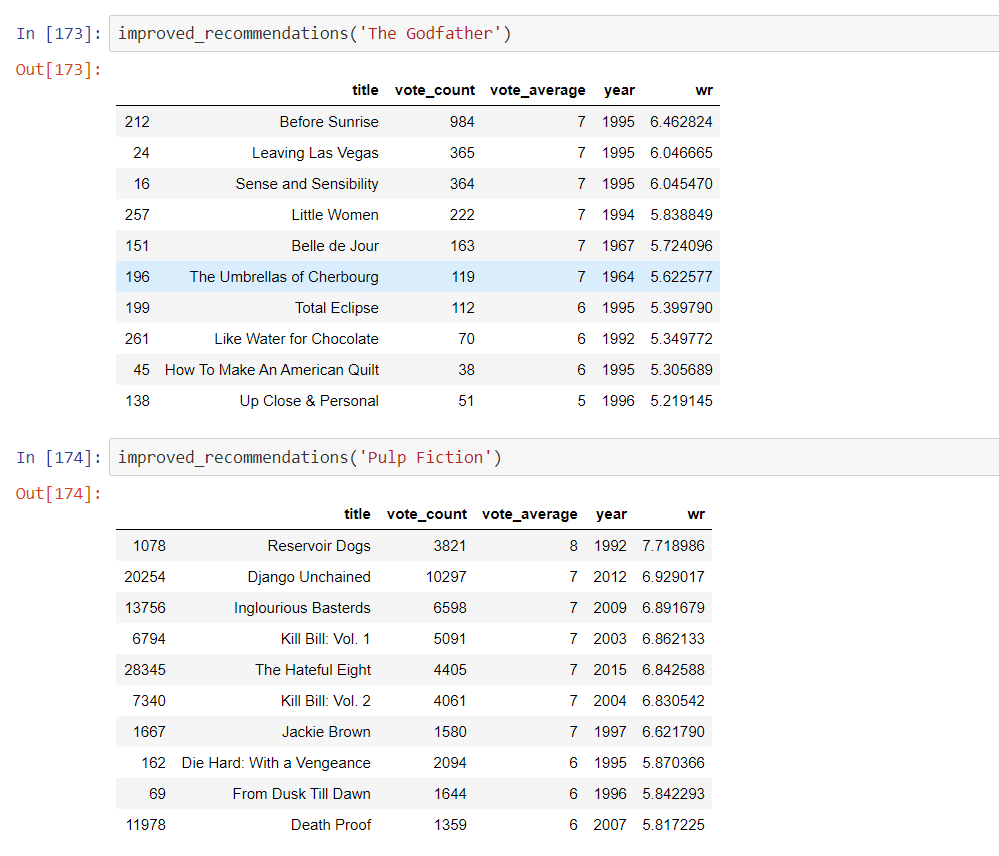
We see that for Superman, our system is able to identify it as a Superman film and subsequently recommend other Superman films as its top recommendations. But unfortunately, that is all this system can do at the moment. This is not of much use to most people as it doesn't take into considerations very important features such as cast, crew, director and genre, which determine the rating and the popularity of a movie.

We would be using much more suggestive metadata than Overview and Tagline. Next, we would build a more sophisticated recommender that takes genre, keywords, cast and crew into consideration.

Metadata Based Recommender:

Popularity and Ratings

It recommends movies regardless of ratings and popularity. Therefore, we will add a mechanism to remove bad movies and return movies which are popular and have had a good critical response.

We will take the top 25 movies based on similarity scores and calculate the vote of the 60th percentile movie. Then, we will calculate the weighted rating of each movie using IMDB's formula.

Hybrid Recommender:

We tried to build a simple hybrid recommender that brings together:

Input: User ID and the Title of a Movie



Conclusion/ Results:

**Results and Discussion:**

Accuracy achieved: 95%

* We created recommenders using demographic, content- based and collaborative filtering.
* While demographic filtering is very elementary and cannot be used practically,
* Hybrid Systems can take advantage of content-based and collaborative filtering as the two approaches are proved to be almost complimentary.
* This model was very baseline and only provides a fundamental framework to start with.

References:

<https://hackernoon.com/introduction-to-recommender-system-part-1-collaborative-filtering-singular-value-decomposition-44c9659c5e75>

<https://www.kaggle.com/rounakbanik/movie-recommender-systems>

<http://trouvus.com/wp-content/uploads/2016/03/A-hybrid-movie-recommender-system-based-on-neural-networks.pdf>