SPRINGBOARD - DATA SCIENCE CAREER TRACK

CAPSTONE PROJECT-2

REPORT ON MOVIE RECOMMENDATION SYSTEMS

TABLE OF CONTENTS

INTRODUCTION **4**

PROBLEM STATEMENT **4**

EXPLORATORY DATA ANALYSIS **4**

GENRES 4

MOVIE CHART FOR LISTING TOP MOVIES IN EACH GENRES 5

Viewer behaviour/ Movies as per user's likings. 6

VOTE AVERAGE AND VOTE COUNTS 6

CHOOSING AN INDICATOR FOR GOOD MOVIES 7

MEAN RATINGS(from Movielens dataset) 8

GENERIC RECOMMENDATIONS

RECOMMENDATIONS BASED ON POPULARITY: 8

RECOMMENDATIONS BASED ON WEIGHTED VOTE AVERAGE 10

RECOMMENDATIONS BASED ON CONTENT FILTERING **10**

RECOMMENDATION BASED ON MOVIE DESCRIPTION 10

RECOMMENDATION BASED ON LIKES OF DIRECTOR OR CAST 11

RECOMMENDATIONS BASED ON COLLABORATIVE FILTERING **12**

EVALUATE PREDICTIONS 13

CUSTOMISED RECOMMENDATION FOR THE USER 13

GENRES SPECIFIC RECOMMENDATION FOR THE USER 14

Why is collaborative filtering better ? 16

INTRODUCTION

The purpose of this project is to build a recommendation engine for movies.Recommender System seeks to predict or filter preferences according to the user's choices. Recommender systems are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general.

I have worked on the TMDB 5000 Movies dataset . The link is: https://www.kaggle.com/tmdb/tmdb-movie-metadata. For collaborative filtering, I have used data from movielens dataset of 10000 movies. The link is https://grouplens.org/datasets/movielens/latest/. We will clean data, analyse relevant variables and build the following 3 kinds of recommenders:

1. Simple recommender
2. Recommender based on content-based filtering
3. Recommender based on collaborative filtering

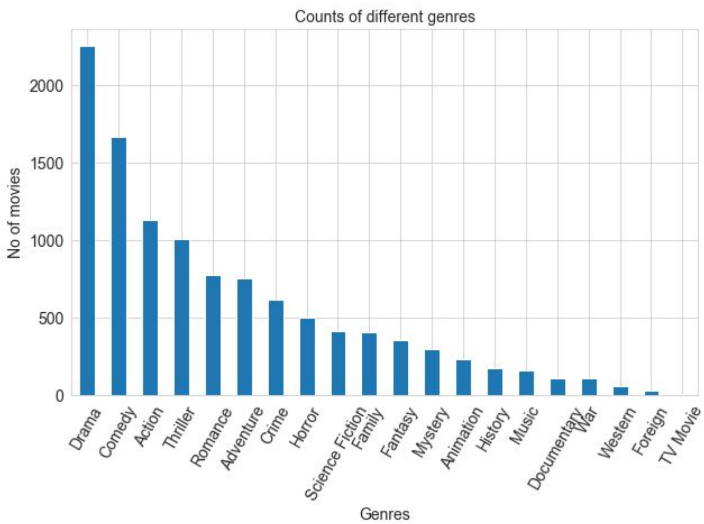
PROBLEM STATEMENT

For the viewer, the problem is of choice as there are thousands of movies to choose from. The viewer experience is better when movies can be recommended according to his tastes and preferences in the best possible manner. The purpose of this project is to build a recommendation engine for movies.Recommender System seeks to predict or filter movies according to the user's choices.

EXPLORATORY DATA ANALYSIS

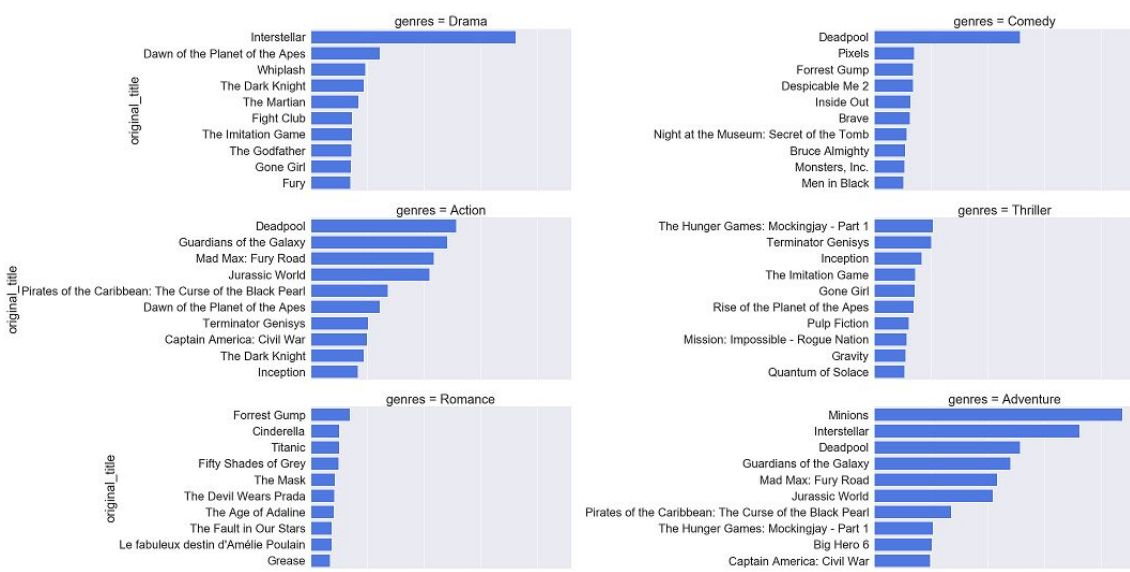
GENRES

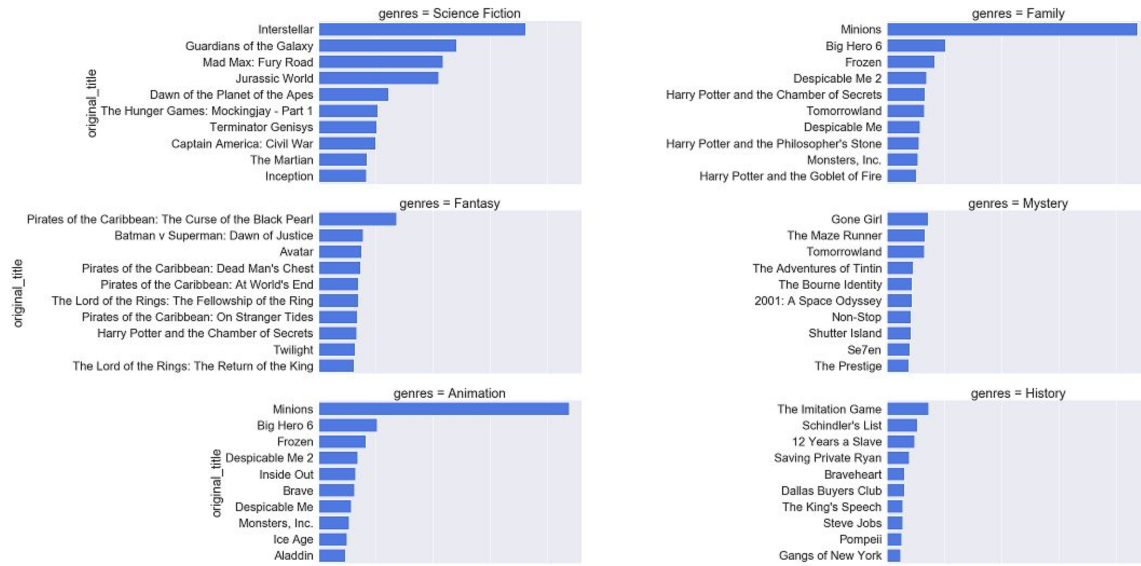
Let’s explore this variable to analyse the counts and the movies that figure in each category.



Most of the movies listed on TMDB are from the Drama, Action and Comedy genres.

MOVIE CHART FOR LISTING TOP MOVIES IN EACH GENRES





This variable describes the movies and differentiates it into distinct groups. Selection of movies based on genres also gives an idea about the tastes of viewers. We can see some trends in genres wise popularity. Most popular movies listed on TMDB are of genres:- Action, Sci-Fi and Adventure.

Viewer behaviour/ Movies as per user's likings.

Some insights about the viewer can be captured from this data. It's clear that the ratings given by the user reflects how he evaluates the movie. So an aggregate of rating of 5/4 for typical kinds of movies reflects user behaviour. This would reveal if the user likes more of comedy or romantic or action kinds of movies.

We can find out the movie genres that the user tends to rate highest most of the time.

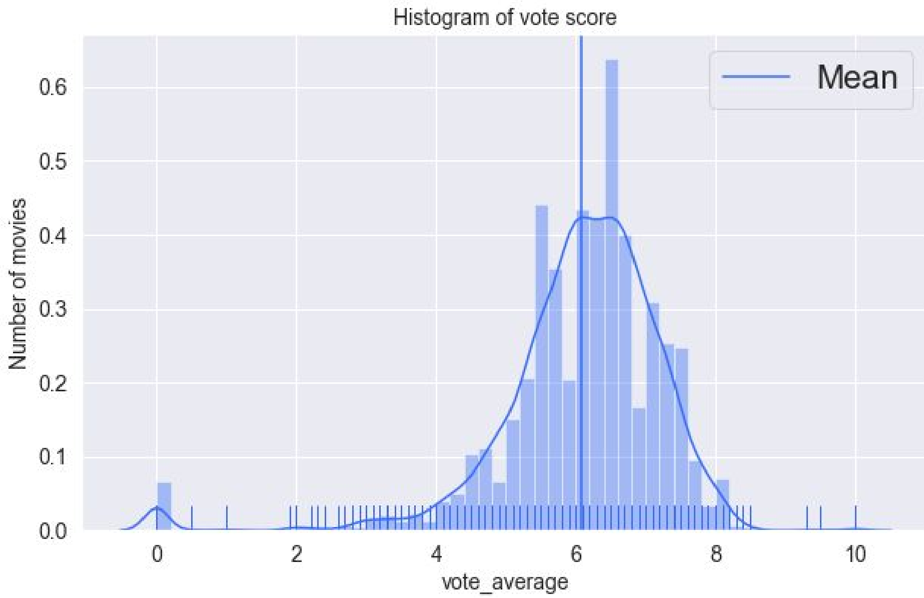
Let's take userId: 1 and analyse the genres this user has rated highest. The viewer with userId 1 mostly gives high ratings to Adventure kinds of movies.We can list down movies in the category most liked by the user.

Title	Rating	Genres
Rocketeer, The (1991)	5	['Action', 'Adventure', 'Sci-Fi']
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)	5	['Action', 'Adventure']
Back to the Future (1985)	5	['Adventure', 'Comedy', 'Sci-Fi']
Highlander (1986)	5	['Action', 'Adventure', 'Fantasy']
Indiana Jones and the Last Crusade (1989)	5	['Action', 'Adventure']
Austin Powers: International Man of Mystery (1997)	5	['Action', 'Adventure', 'Comedy']
Thunderball (1965)	5	['Action', 'Adventure', 'Thriller']
Conan the Barbarian (1982)	5	['Action', 'Adventure', 'Fantasy']
Live and Let Die (1973)	5	['Action', 'Adventure', 'Thriller']
Goonies, The (1985)	5	['Action', 'Adventure', 'Children', 'Comedy', 'Fantasy']

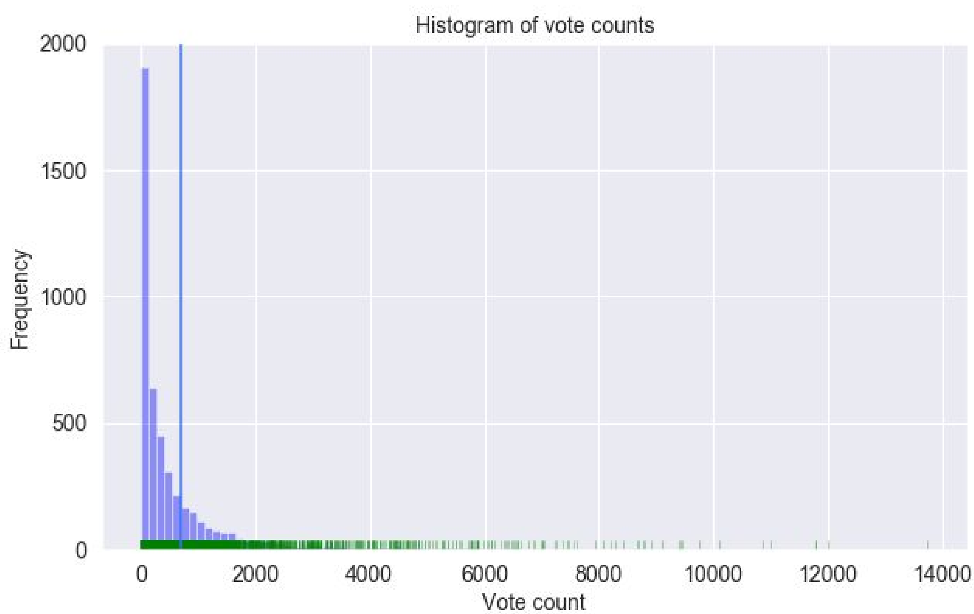
VOTE AVERAGE AND VOTE COUNTS

Minimum vote_average is 1 and maximum is 10. Let's take a look at the histogram.

The mean vote_average is 6.09



This is a bimodal and left- skewed distribution, There are only a handful of movies that have a vote average greater than 8 .



As seen, vote count is exponentially distributed. There are a number of movies with more than 2000 vote_count.

CHOOSING AN INDICATOR FOR GOOD MOVIES

Vote Average as an Indicator

Title	Vote Count	Vote Average
Dancer, Texas Pop. 81	1	10
Little Big Top	1	10
Stiff Upper Lips	1	10
Me You and Five Bucks	2	10
Sardaarji	2	9.5
One Man's Hero	2	9.3
The Shawshank Redemption	8205	8.5
There Goes My Baby	2	8.5
The Godfather	5893	8.4
The Prisoner of Zenda	11	8.4

Vote Average, used alone as an indicator for good movies shows movies watched by single viewers. As these scores are topping the charts, it could be misleading.

A more reliable Indicator

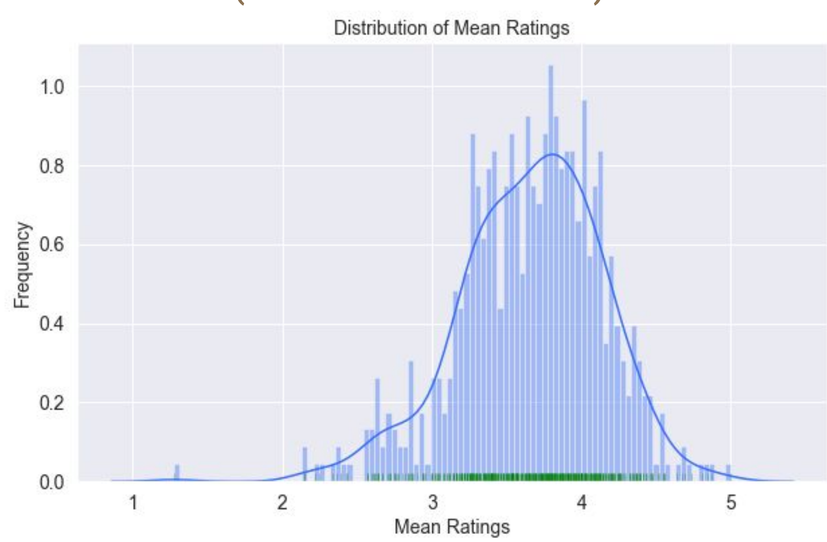
Movie ID	Title	Vote Count	Vote Average
96	Inception	13752	8.1
65	The Dark Knight	12002	8.2
0	Avatar	11800	7.2
16	The Avengers	11776	7.4
788	Deadpool	10995	7.4
95	Interstellar	10867	8.1
287	Django Unchained	10099	7.8
94	Guardians of the Galaxy	9742	7.9
426	The Hunger Games	9455	6.9
127	Mad Max: Fury Road	9427	7.2

If Vote Average along with vote count is chosen as an indicator for good movies, it makes recommendations of movies more reliable.

Highest vote_count is 13700. When we see this chart, it's clear that vote_average is more reliable when vote_counts also figure in.There are many thousands of movies and it is desired that one gets more relevant choices to pick from. 'Most watched movies' relate to their ratings and descriptions which represent a cumulative view of many people. So we can have a benchmark in vote counts.

If we take the 90th percentile as a benchmark, we get the vote_count of 1832 and there are some 480 movies that qualify for it.

MEAN RATINGS(from Movielens dataset)



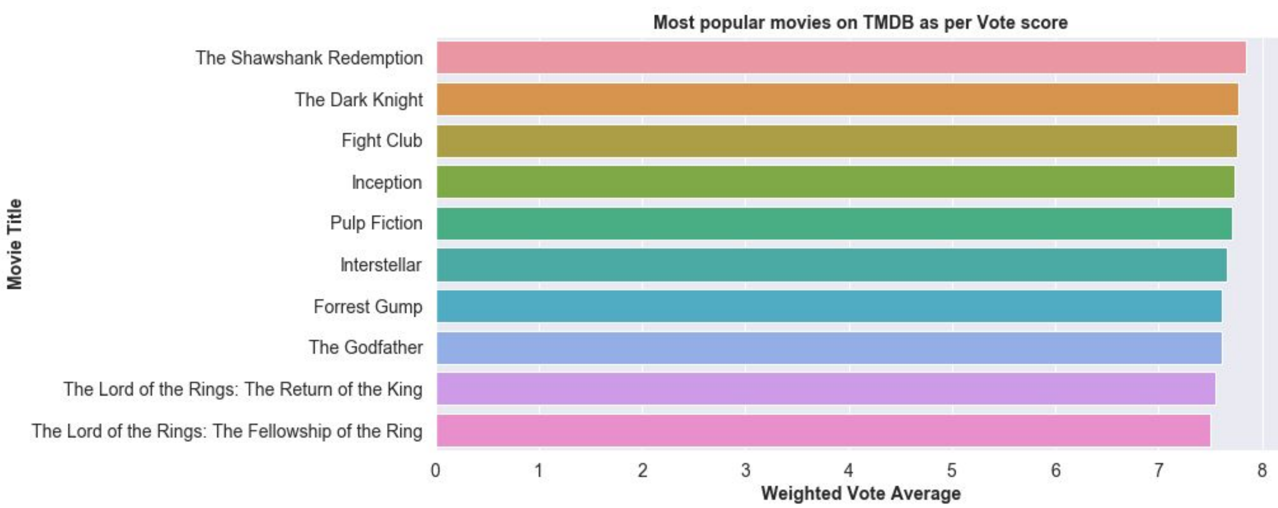
The distribution is close to normal with slight skewness to the left.
The overall-mean of ratings is 3.5 The mean of user-wise ratings is 3.66 The mean ratings are centered in the range 2.5 to 4.5. Ratings of 1,2 and 5 are quite rare

GENERIC RECOMMENDATIONS

Generic recommendations are based on measures such as popularity, vote scores or genres. The movies are ranked as per the scores and Top ranking movies are recommended. These recommendations will be the same for all.

A. RECOMMENDATIONS BASED ON POPULARITY:

Popularity as a measure, fairly indicates how it has been liked by people. Popularity can be a reliable measure for selecting movies to watch. If this single metric is applied, it would generate recommendation for all viewers, irrespective of their individual preferences. And as there are only a handful of movies in popularity range of 200 to 800, its highly probable that these movies would have been already watched.



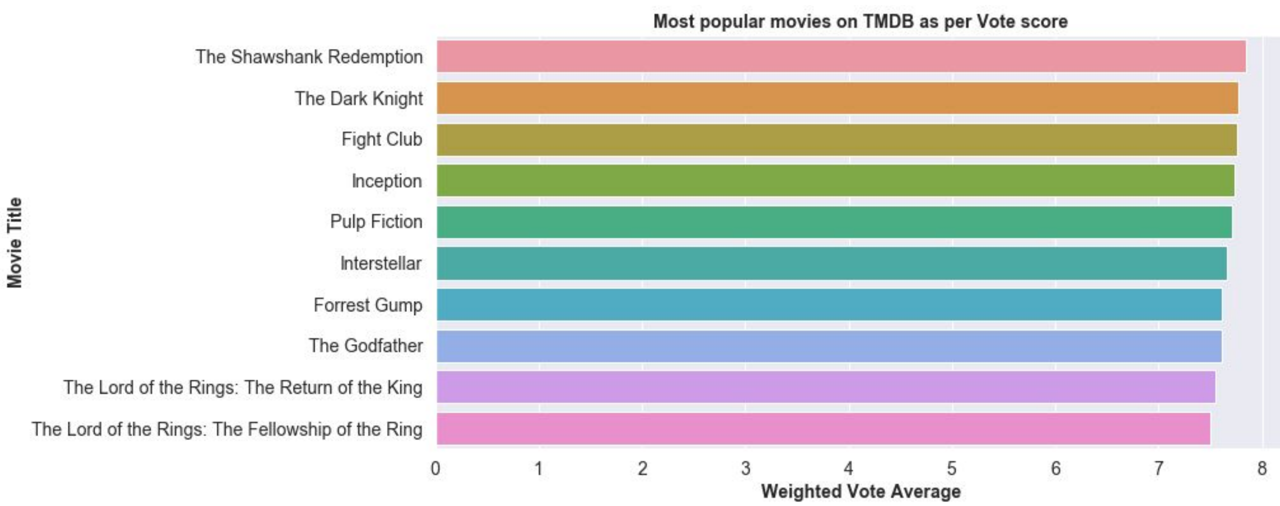
B. RECOMMENDATIONS BASED ON WEIGHTED VOTE AVERAGE:

We can improve the movie recommendation chart by being selective on the scores assigned to a movie by a large number of viewers.

A movie might have a high vote_average but very few might have watched it. So we need to factor the vote_count to get a better idea. Further, we also factor a benchmark for a minimum vote_count to get listed in the Top 250(currently 3000). I will use IMDB's weighted rating formula to construct.

The formula for calculating the Top Rated 250 Titles gives a true Bayesian estimate:
weighted rating (WR) = (v ÷ (v+m)) × R + (m ÷ (v+m)) × C where:
R = average for the movie (mean) = (Rating)
v = number of votes for the movie = (votes)
m = minimum votes required to be listed in the Top 250
C = the mean vote across the whole report (currently 7.0)

For the Top 250, only votes from regular voters are considered.



These movies indicate the best impressions of a larger number of viewers. However, generic recommendations are impersonalised.

RECOMMENDATIONS BASED ON CONTENT FILTERING

Content-based recommenders use item features to recommend other items similar to what the user likes. 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 likes a particular item, he or she will also like an item that is similar to it.

A. RECOMMENDATION BASED ON MOVIE DESCRIPTION

This recommendation is designed to match movies of the same description. The steps are:

- Clean text data of movie descriptions.
- Convert text into Tfidf vectors.
- Find similarities between movies by computing cosine similarities.
- Group movies that are most similar to a selected movie.

Let's say a movie 'Dark Knight Rises' is selected. Let's see movies similar to it.

Title
Batman Forever
The Dark Knight
Batman
Batman Returns
Slow Burn
Batman Begins
Batman: The Dark Knight Returns, Part 2
JFK
Batman & Robin
Batman v Superman: Dawn of Justice

If we look at the movie recommendations, it's more suitable for viewers, who like the same movie plot or are fans of a type of movie like a 'Batman' movie.

But 'Dark Knight Rises' could have been selected for other reasons. The viewer could be looking for movies of the same make like the same Director /cast / genres.

B. RECOMMENDATION BASED ON LIKES OF DIRECTOR OR CAST

This recommendation is designed to match movies of the same make. A text metadata is created including director, genres, keywords and lead actors. The steps are:

- Clean text data of movie descriptions.
- Convert text into word vectors.
- Find similarities between movies by computing cosine similarities.
- Group movies that are most similar to a selected movie.

Now let's take a look at the movies similar to 'Dark Knight Rises':

Title	Weighted Vote Average
The Dark Knight	7.77
Inception	7.74
Interstellar	7.66
The Dark Knight Rises	7.22
The Prestige	7.22
Batman Begins	7.09
Memento	7.24
Kick-Ass	6.70
Hitman	6.05
Insomnia	6.29

The recommendation has improved but it lacks personalization.

RECOMMENDATIONS BASED ON COLLABORATIVE FILTERING

Collaborative filtering methods are based on collecting and analyzing a large amount of information on user behaviors, activities or preferences and predicting what users will like based on their similarity to other users. The fundamental assumption behind collaborative filtering technique is that similar user preferences over the items could be exploited to recommend those items to a user who has not seen or used it before. In simpler terms, we assume that users who agreed in the past (purchased the same product or viewed the same movie) will agree in the future.
We will build recommenders based on collaborative filtering for movies using data from movielens dataset. The link is https://grouplens.org/datasets/movielens/latest/.

Viewer behaviour/ Movies as per user's likings.

Some insights about the viewer can be captured from this data. It's clear that the ratings given by the user reflects how he evaluates the movie. So an aggregate of rating of 5/4 for typical kinds of movies reflects user behaviour. This would reveal if the user likes more of comedy or romantic or action kinds of movies.We can find out the movie genres that the user tends to rate highest most of the time by applying the relevant filters.

Creating a python Class for collaborative filtering:

For getting similar user's data, we will take cosine similarities. We will use KNNBasic Model to learn the data and predict the ratings. The steps are:
We will create a Collab_user_wise Class with 5 functions, which can be called as methods to this class. The 5 functions are:

a. Learn: KNNBasic Model from Surprise Library is used to train the dataset and data specific to a user is extracted.

b. Evaluate: This function inputs userId and returns model metrics ie Root Mean Squared Error.

c. Collaborative_recom: This function returns top 10 recommendations based on predicted rating. It is a user-based collaborative filtering. To ensure that the user is recommended trending movies we have

d. User_liked_genres: This is also a user based filtering. In addition, we have selected movies of genres that the user specifically likes based on past ratings. The list is then filtered based on weighted vote averages.

e. Hybrid-recommendation: This recommendation is based on user-based and item based filtering.

The python Class takes a UserID as an input and generates recommendations specific to that user. By instantiating the Class and using the methods we can perform the following tasks:

A. EVALUATE PREDICTIONS

The root mean squared error for predictions, for UserID 1 is 1.01.

B. CUSTOMISED RECOMMENDATION FOR THE USER

These recommendations initially grab attention as it shows all the possible movies that the viewer is likely to rate high. These movies are grouped together based on collaborative ratings of similar kinds of viewers. KnnBasic Model from Surprise Library has been used to learn and predict from the data by Movielens.

Movie ID	Title	Estimated Ratings	Watched Status	Weighted Vote Average
491	Shawshank Redemption, The (1994)	4.7	unseen	7.85
6710	Dark Knight, The (2008)	4.6	unseen	7.77
7372	Inception (2010)	4.4	unseen	7.74
8376	Interstellar (2014)	4.3	unseen	7.66
847	Godfather, The (1972)	4.5	unseen	7.61
4800	Lord of the Rings: The Return of the King, The (2003)	4.3	unseen	7.55
3639	Lord of the Rings: The Fellowship of the Ring, The (2001)	4.2	unseen	7.51
8475	Guardians of the Galaxy (2014)	4.2	unseen	7.47
4137	Lord of the Rings: The Two Towers, The (2002)	4.3	unseen	7.45
8063	Django Unchained (2012)	4.2	unseen	7.40

C. GENRES SPECIFIC RECOMMENDATION FOR THE USER

Workings:

- Identify the genres that the user likes based on his ratings.
- Apply model's predictions on all unseen movies.
- Select movies of genres liked by user
- Filter movies based on weighted vote Average.

These are the movies of the genres that the user likes.

Title	Estimated Rating	Watched Status	Weighted Vote Average	Genres
Star Wars: Episode V - The Empire Strikes Back (1980)	4.4	seen	7.48	['Action', 'Adventure', 'Sci-Fi']
Star Wars: Episode IV - A New Hope (1977)	4.5	seen	7.47	['Action', 'Adventure', 'Sci-Fi']
Back to the Future (1985)	4.3	seen	7.36	['Adventure', 'Comedy', 'Sci-Fi']
Gladiator (2000)	4.3	seen	7.25	['Action', 'Adventure', 'Drama']
Star Wars: Episode VI - Return of the Jedi (1983)	4.5	seen	7.19	['Action', 'Adventure', 'Sci-Fi']
Toy Story (1995)	4.1	seen	7.11	['Adventure', 'Animation', 'Children', 'Comedy', 'Fantasy']
Jurassic Park (1993)	3.9	seen	7.02	['Action', 'Adventure', 'Sci-Fi', 'Thriller']
Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)	4.5	seen	6.99	['Action', 'Adventure']
Indiana Jones and the Last Crusade (1989)	4.3	seen	6.86	['Action', 'Adventure']
Monty Python and the Holy Grail (1975)	4.2	seen	6.71	['Adventure', 'Comedy', 'Fantasy']

Based on the above learning, we can now select movies of the genres liked by the user from unseen lot, and make recommendations after applying the filter of weighted average vote.

Title	Estimated Rating	Watched Status	Weighted Vote Average	Genres
Lord of the Rings: The Return of the King, The (2003)	4.35	unseen	7.55	['Action', 'Adventure', 'Drama', 'Fantasy']
Lord of the Rings: The Fellowship of the Ring, The (2001)	4.22	unseen	7.51	['Adventure', 'Fantasy']
Guardians of the Galaxy (2014)	4.19	unseen	7.47	['Action', 'Adventure', 'Sci-Fi']
Lord of the Rings: The Two Towers, The (2002)	4.25	unseen	7.45	['Adventure', 'Fantasy']
Inside Out (2015)	3.91	unseen	7.40	['Adventure', 'Animation', 'Children', 'Comedy', 'Drama', 'Fantasy']
Lion King, The (1994)	4.30	unseen	7.31	['Adventure', 'Animation', 'Children', 'Drama', 'Musical', 'IMAX']
WALL·E (2008)	4.21	unseen	7.24	['Adventure', 'Animation', 'Children', 'Romance', 'Sci-Fi']
Dark Knight Rises, The (2012)	4.23	unseen	7.22	['Action', 'Adventure', 'Crime', 'IMAX']
Up (2009)	4.18	unseen	7.21	['Adventure', 'Animation', 'Children', 'Drama']
The Martian (2015)	4.06	unseen	7.16	['Adventure', 'Drama', 'Sci-Fi']

D. HYBRID RECOMMENDATION FOR THE USER

Workings::
- Select a movie from the viewer's liked genres.
- Apply models predictions on unseen movies
- Similar movies based on content are generated.
- Apply filter of weighted vote average

The Hybrid recommender lists the following movies for the user:

Title	Watched Status	Weighted Vote Average	Estimated Rating
Seven Samurai (Shichinin no samurai) (1954)	unseen	6.56	4.27
Blade Runner (1982)	unseen	7.06	4.17
Nebraska (2013)	unseen	6.32	4.12
The Martian (2015)	unseen	7.16	4.06
Shin Godzilla (2016)	unseen	6.11	4.00
Raise the Titanic (1980)	unseen	6.09	4.00
Ip Man 3 (2015)	unseen	6.14	4.00
Gunman, The (2015)	unseen	6.02	4.00
Nicholas Nickleby (2002)	unseen	6.10	4.00
Bourne Supremacy, The (2004)	unseen	6.63	3.98

Why is collaborative filtering better ?

- It’s efficient, provided all relevant user data is available.
- This recommender is able to match users and generate reasonable predictions on ratings.
- User data helps to customise the recommendations.

REFERENCES:

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

https://www.datacamp.com/community/tutorials/recommender-systems-python

https://blog.dominodatalab.com/recommender-systems-collaborative-filtering/