**Movie Recommendation System**



**Capstone Project interim report**

15 November 2020

Version 1.0

Vivek Jain

https://github.com/vivek2500

Subhash Bhandari

https://github.com/iasubhash

1. **Overview**

Nowadays, we need recommender systems almost everywhere in our lives. Therefore, retailers are become more interested in recommender systems to analyze patterns of user interest in products and provide personalized recommendations. The first goal of this project is understanding, analyzing, and correlating the trend in average rating movies of different genres. The second goal is building recommender engines to provide recommendations to different users and build different machine learning models to predict the rating of each movie.

1. **Problem Definition**

A recommendation system is a system that provides suggestions to users for certain resources like books, movies, songs, etc., based on some data set. Movie recommendation systems usually predict what movies a user will like based on the attributes present in previously liked movies. Such recommendation systems are beneficial for organizations that collect data from large amounts of customers and wish to effectively provide the best suggestions possible. A lot of factors can be considered while designing a movie recommendation system like the genre, tag associating movies, etc.

Organizations can use such recommendation systems to acquire and retain customers. Such a system can help organizations to increase their customer base and revenue.

**2. Proposed Solution**

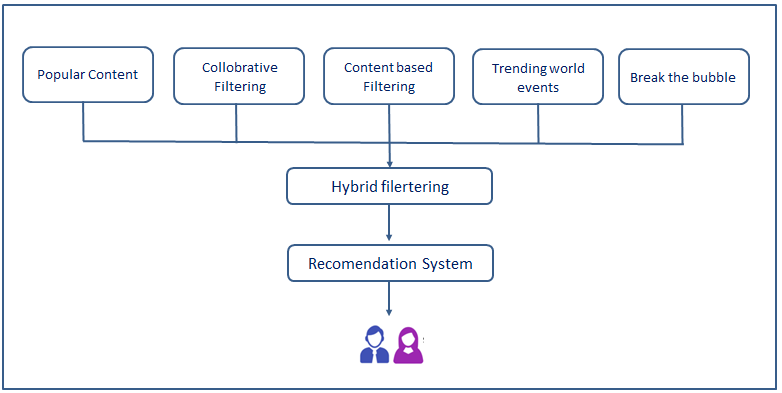
Movie recommendation are old problems and lot of work has been done with respect to optimization the solution, Organizations like Netflix, Amazon Prime invest heavily in understanding user behavior and try to recommend movies which they think the user will like, The idea is to increase the engagement which in turn will increase their revenue and loyalty of their customer towards them

We propose a hybrid solution that takes multiple aspects into consideration, like user preference, content similarity, and events happening around him/her. We plan to take a mix bag approach in order to recommend users movies which s/he will intent to like, We will be having a mix of :

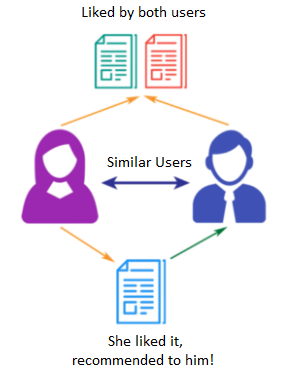
1. Popular Movie Choices
2. Collaborative Filtering Movie recommendation
3. Content-Based Movie Recommendation
4. Movie Recommendation based on trending world events
5. Movie recommendation based opposite to what genre user likes

The mix bag approach will help us in engaging the user with the platform and stay longer. We plan to take future data and tweak the proportion of each recommendation in the future.

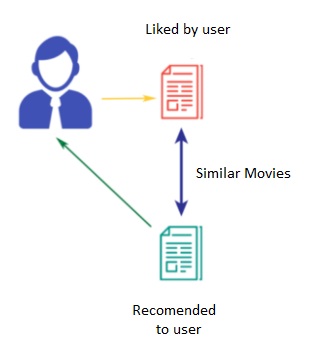
Figure below decapitating our mixed bag hybrid recommendation techniques



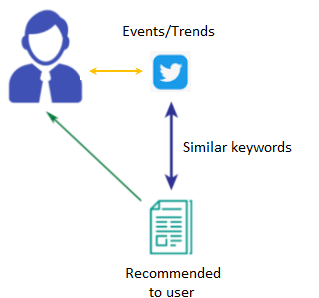
2.2 Collaborative Filtering Movie recommendation



2.3 Content-Based Movie Recommendation



2.4 Movie Recommendation based on trending world events



1. **Dataset**

We will use two datasets for this project as:

1. Dataset is collected from (https://www.kaggle.com/karrrimba/movie-metadatacsv) Kaggle[9], The movie dataset. It contains more than 45000+ movie’s metadata. Along with Credit, Keyword, and Rating files, in csv format.

The **Credit** dataset contains the following features:-

* movie\_id - A unique identifier for each movie.
* cast - The name of lead and supporting actors.
* crew - The name of the Director, Editor, Composer, Writer etc.

The **Movie Metadata** dataset has 45000 records with the following features:-

* 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 - A unique identifier for each movie.
* keywords - The keywords or tags related to the movie.
* 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’s 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.
* status - "Released" or "Rumored".
* tagline - Movie's tagline.
* title - Title of the movie.
* vote\_average - average ratings the movie received.
* vote\_count - the count of votes received.

The **Keyword** dataset contains the following features:-

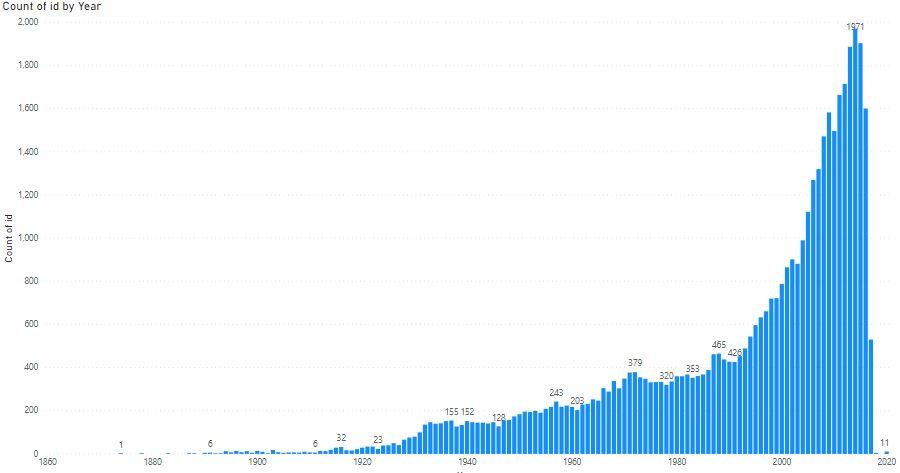
* movie\_id - A unique identifier for each movie.
* Keyword - Keywords related to the movie and its plot.

We have small files ( 100,000 ratings from 700 users on 9,000 movies) for the dataset as we want to reduce the burden on our machines.

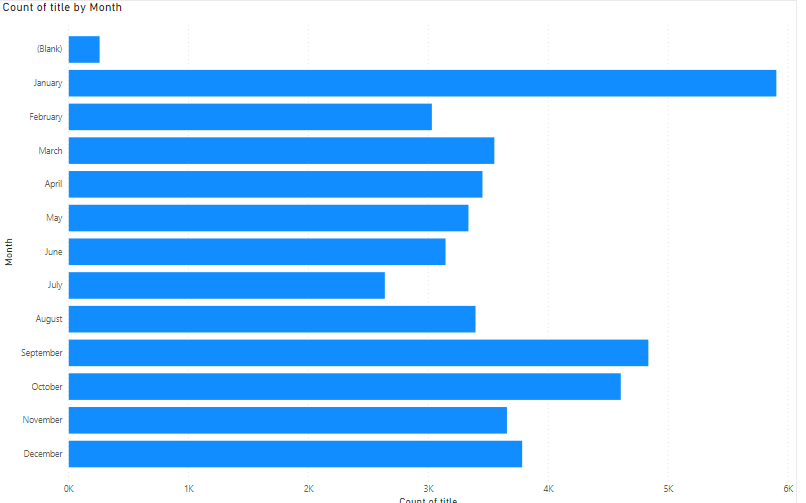
1. The second data set is top trending hashtags on the user location from twitter, we would use from twitter api’s to scrap this tags.

EDA

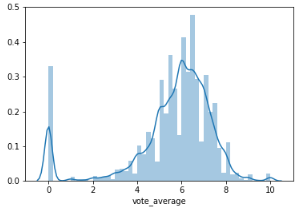
We see that Most movies are released in recent years.



Most movies are released at the start of the year or at the end of the year, This could be a good season to launch new movies.



Average vote of the audience shows that we have a normalized curve , as most ratings are between 4-8..



EDAs are not so helpful when you want to do recommender model problem , we will do more analysis to see if we can find some hidden patterns

**4. Evaluation Metric**

We intend to focus on a few of the well known techniques to gauge the accuracy of the model.

* **RMSE (Root Mean Square Error)** : RMSE is a good way to see the accuracy of the model



* **Mean Absolute error (MAE) :** We also intent to use Mean absolute error to gauge the accuracy

For content based recommendation we will use cosine distance or pearson distance evaluation metrics.

For trending events we still do not have evaluation mechanism as we are doing exact match of hashtags with keywords. In future we can expand this to do “context match” with keywords, at that time we can implement distance based metrics for evaluation.

**5. Initial Observations**

We got a good start with a popularity based approach of finding and recommending most popular movies to users, we relied on weighted average to find the most popular movies , rather than using average votes directly. It gives us a good start for cold start problems as well.

For content based recommendation, which is nothing but identifying similar movies to what users have liked and showing him/her similar movies, we have used cosine similarity matrix to find similar movies and recommended it to users.

For Collaborative Filtering method, we used SVD model to find the movie based on similarity of users, to find the accuracy we would have used RMSE and MAE matrix and we are getting values **below 1** which is a good accuracy to have.

For trending based movies, we plan to scrap top trends from twitter and match the hashtags with our movie keywords to identify movies, which may spike interest of users to proceed with the recommendation.

In order to add a little bit of twist and take some sort of risk, we plan to add a recommendation opposite to what the user is watching lately, it may or may not catch the interest of the user , that we have to see and amend are approach going forward but this can be a good counter intuitive approach.

**6. Challenges and Remedies**

One challenge we see is, to consume all the data to predict the result , we don't have that level of hardware so we plan to take the mini set of data, which will help us in reducing the load on our system and predict the set of movies which we need.

Till now we have just compared similarity with movie plot and which can be a good attribute but we may go forward and include more attributes like cast and crew of the movie, as movies by the same set of people may produce similar levels of content which users like. We can also increase the weightage of director and top actors while trying to find similar movies/content , as director and lead actor has a higher level of involvement in producing the movie/content.

For identifying the trend, we are just relying on global trends which may not be suitable for everyone so we can have an option of customizing it more if we can capture preference of users, we need to find a way to do that.

**7. Literature survey**

Most recommendation systems use content based and collaborative approach as base and then put an extra layer to improve the recommendations. Dhoha and Ghadeer [1] has also used users demography to understand the user profile and cluster the similar users for increasing the efficiency. They have also used Weka Software to find out the movie category. Ronak and Ketan [2] have exploited Tag features along with the genre of the movie to enhance the model, They have used weighted average and used user-movie-tag, user-movie matrix to improve the overall accuracy.

Huizhi [3] has done detailed research on the effectiveness of tags to find the best user item combination, she uses social tags. Huizhi argues that we should use tag based approaches for mapping users with items. Suvir [4] has taken a different approach by using NLP and getting his results verified against movie experts' opinion. He has used top user reviews to find various semantics and has used Cosine Similarity and Hellinger Distance to find similarity between reviews. Has also considered genre, mood, overlapping of cast, similarities of the movie.

Yibi and team [5] has used sentiment analysis on user review, along with content based and collaborative filtering methods. They have used SPARK to enhance the performance of the system.The work of Rahul and Prakash [6] has taken an innovative approach for the problem at hand and used cuckoo search along with K- means to build an efficient recommendation system. Further [7] team has added a layer or item popularity to enhance the model. [8] has used k-cliques to create clusters and introduce movies in similar groups using the collaborative filtering method.

Leidy Esperanza MOLINA FERNÁNDEZ proposed the paper that covered the theory of the most popular recommendation system algorithms Popularity, Collaborative Filtering, Content based Filtering and Hybrid Approaches. The aim of this research was to understand the pros and cons of all the algorithms, and then be able to decide which one was the one that fits better

the dataset. Based on this discussion, just Popularity and Collaborative Filtering were

implemented, for CF both Memory-based CF and Model-based CF were used. The problem with Popularity is that all the recommendations are the same for every single user, hence they did not focus on these results. Item-based collaborative filtering was implemented using the cosine and the Pearson correlation as the distance function. In addition, Model-based CF is based on matrix factorization, then they decided to make use of SVD.

Phonexay Vilakone and team [4] In order to achieve more accuracy than collaborative filtering methods; the maximal clique method used in social network analysis introduced in this paper is the first time that is used in a movie recommendation system and the output of this method is very effective. For performance evaluation, they evaluated the collaborative filtering method using a k nearest neighbor, maximal clique method, k-clique method and improved k-clique methods. The results showed that the improved k-clique method improved the precision of the movie recommendation system more than the other methods used in this.

**8. References**

All research papers available at this link.:

[1] A Survey Paper on Recommender Systems by Dhoha and Ghadeer

[2] ENHANCING MOVIE RECOMMENDER SYSTEM by Ronak Patel

[3] A hybrid recommender systems based on weighted tags by Huizhi Liang [4] Efficient Features for MovieRecommendation Systems by Suvir Bhargav

[5] A Sentiment-Enhanced Hybrid Recommender System for Movie Recommendation: A Big Data Analytics Framework by Yibo Wang

[6] An effective collaborative movie recommender system with cuckoo search by Rahul Katarya

[9] https://www.kaggle.com/rounakbanik/the-movies-dataset