MOVIE RECOMMENDATION SYSTEM

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

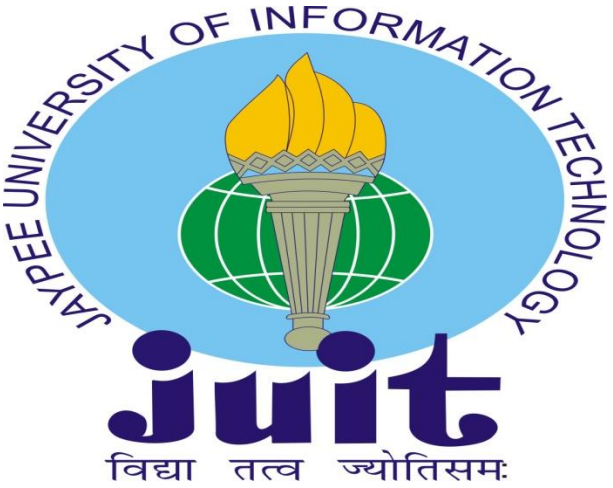
By

Aakash (181241)

Under the supervision of

Dr. Rakesh Kanji
Assistant Professor[SG]

to



Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CERTIFICATE

Candidate's Declaration

I hereby declare that the work presented in this report entitled "MOVIE RECOMMENDATION SYSTEMS" in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Wakanaghat is an authentic record of my own work carried out over a period from January 2022 to May 2022 under the supervision of **Dr Rakesh Kanji** (Assistant Professor(SG)).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Aakash
181241

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr Rakesh Kanji
Assistant Professor
Computer Science & Engineering
Dated: 13/05/2022

ACKNOWLEDGEMENT

I would like to thank and express our gratitude to our Project supervisor Dr. Rakesh Kanji for the opportunity that he provided us with this wonderful project "Movie Recommendation System". The outcome would not be possible without his guidance. This project taught me many new things and helped to strengthen concepts of Machine Learning. Next, I would like to express my special thanks to the Lab Assistant for cordially contacting us and helping us in finishing this project within the specified time. Lastly, I would like to thank my friends and parents for their help and support.

Aakash (181241)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A typical Recommender system engine extracts different types of data by using various algorithms and then giving out the best fit of the relevant items as needed by Looking at the nature of the previous behavior of the user on the basis of which it recommends the items or products by recommending the best items which are currently in great demand . The other major reasons for this may be maximizing the gain. There are basically 3 ways which are used on recommending the items to the users . Demographic filtering which gives a general recommendation which is based on any one of the more popular movie or its genere. Having same demographic matches it recommends the users movies .In general the movies that are popular are possibly liked more by the more number of users .Other technique is content -based filtering that takes into account the interest of the people and then suggesting the movies to the users .Third technique that is used here is that Collaborative method of filtering where same people are put or grouped together which have same characteristic interest or movies pattern .

INTRODUCTION

Recommendation system is basically a filtering system that predicts the users choices and then suggest them the the more accurate results based on the the previous likings of the users . We have a variety of varied applications of this recommendation systems in which we can can be used over the years and now used in various online platforms the basic content of all this platforms are basically different types of movies such as action thriller romantic or maybe your eCommerce website any platform of social media having a professional website such as Linkedin . For example when we use Instagram we can see the previous stories that on the feed of the people we follow so here we can see that the Instagram can monitor our interaction with the various people are our past activities and then it just suggest kind of other related stories of some other accounts that have done some same kind of activity previously or currently. Quite a few time is recommender system also keep improving the activities of a bunch of users based on the activities they have scroll through you attempted. For example on Flipkart when we buy some laptop or any mobile phone then it simply suggests mobile cover tempered glass for mobile or buy USB type C adaptor or type A adaptor for the laptop also. Safed enhancements in the recommender systems users get good recommendation all the time and it keeps on improving as we move forward in the 21st century and they make almost accurate solutions. In case of clash of any e App Music any music platform or any educational then use a simply deny using the app in addition to this the companies have to focus on their recommendation system which is more Complex than it seems. Every user has different preferences and different choices based upon their different type of activities sometime mood also so in case of musics while playing, travelling, running aur after having some fight in relationships etc.

PROBLEM STATEMENT

Recommender systems are tools that aims to get the user's rating and then recommend the movies from a big set of data on the basis of the users matching interest and then classify them into different categories. The sole purpose of the whole system of this recommendation is the search for the content that it would fit into the person's interest for an individual's personal oasis. However it takes into account different factors that would create some different list of content that is specific to different categories of individual/ users . AI based algorithms that recommender systems basically used creates a list of possible different scenarios of devices and then customizing that all the interesting and matching interest/ choices of the individual categories in the end. All the results are basically based on the different activities that they have done previously such as how does the profile look what have gone through the Chrome Browser Opera browser and other Browser which includes their previously browsed history for considering the demographic traits or the possibility how they would like the movie is based on the genre, a set of predictive modelling is constructed through the data(big) which is available and then the movies are protected through the list of 2000 movies set a bunch of few selected movies are recommended using different algorithms different methods different similarity measures.

OBJECTIVE

Movie recommendation system provides the mechanism and classifying the users with the same interest and searches for the content that would be so much interesting belonging to different set of users and then creating different kind of lists and providing interesting recommendations to the individual based on the content the love. The main objective of the recommender system is to used approaches suggest demographic filtering ,content based filtering , collaborative filtering to find the set of movies with every user likes for specific set of users. The movies that have high probability of being liked by the general set of users will be displayed to the user by the recommender in the end and then in another technique we will try to find the users with different interest using the information collected through different activities an Indian in collaborative filtering will test all those users which have same type of interests to get the final set of movies to be recommended to the users individually. So we will use different categories of recommender filtering techniques and then compare in contrast that results obtained in different methods and will try to to improve the results as h dataset for set of movies goes larger and larger above the computational bound of the system which is generally a limitation on the large dataset.

METHODOLOGY

Various types of recommender system which we can classify as below

1. **Demographic Filtering** : This technique of recommendation filtering is based on the popularity basis for the gender specific users. The system simply e recommends the movies to users chaps slightly same demographic matches. Every user is different in this case so it is very simple to applied this approach. Idea is that the movies which are very popular and accepted by a bunch of people are having the highest probability of getting like by the users.

To understand this demographic filtering:

. Create a metric to rate the movie.

. Find the different metric score.

. Shorting the scores and then recommending the movies which are best rated for the users

$$\text{Weighted Ratings (WR)}=\frac{v}{v+m}.R+\frac{m}{v+m}.C$$

- 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

```
#Sort movies based on score calculated above
q_movies = q_movies.sort_values('score', ascending=False)

#Print the top 15 movies
q_movies[['title', 'vote_count', 'vote_average', 'score']].head(10)
```

	title	vote_count	vote_average	score
1881	The Shawshank Redemption	8205	8.5	8.059258
662	Fight Club	9413	8.3	7.939256
65	The Dark Knight	12002	8.2	7.920020
3232	Pulp Fiction	8428	8.3	7.904645
96	Inception	13752	8.1	7.863239
3337	The Godfather	5893	8.4	7.851236
95	Interstellar	10867	8.1	7.809479
809	Forrest Gump	7927	8.2	7.803188
329	The Lord of the Rings: The Return of the King	8064	8.1	7.727243
1990	The Empire Strikes Back	5879	8.2	7.697884

Fig -Demographic Filtering Method

2. **Content Based Filtering system:** In the content based filtering method we compare the different items with the user's interest profile. So basically the user profile holds the content that is is much more matching to use the form of the features. The previous actions or for the feedback is taken into account a generally takes into account the description of the content that has been edited by the users of different choices. Considering that example where a person buys some favourite item 'M' but item has been sold out and as a result he has to buy the item 'N' on the recommendation of some person as and 'N' has same type of matching features that the first one possesses. So this is simply the content based filtering which is demonstrated below

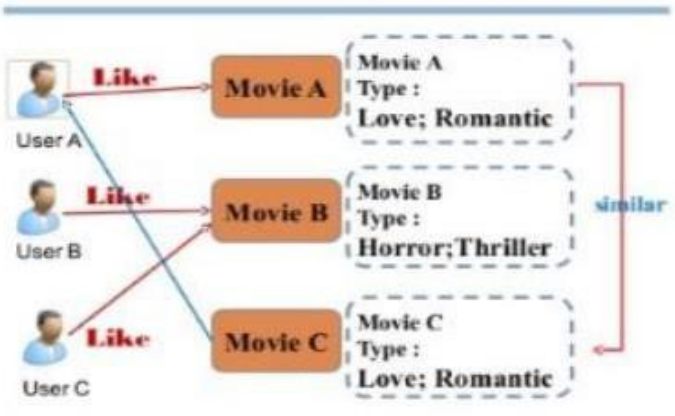


Fig.-Content Based Filtering Method

So here numeric quantity that will be used to calculate the similarity between the two types of movies will be cosine similarity and we will calculate the score it is very very fast to calculate the the magnitude of the score which is obtained through the cosine similarity

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

The steps involved in getting the movie recommendation are as below:

. Having the title find the index of that movie

. Calculate the cosine similarity scores for all the movies

. Arranging the scores in the order of highest priority first that is ascending order

. And then shorting the list based on the similarity scores.

. Getting the first 10 element of the list excluding the first one as it is the movie name in itself.

. Getting the top elements

Repeating above steps we will find the top movies based on the distances which it can get rhe best possible recommendation, the movies that have high probability of being liked by the general set of users will be displayed to the user by the recommender in the end and then in another technique we will try to find the users with different interest using the information collected through different activities an Indian in collaborative filtering will test all those users which have same type of interests to get the final set of movies to be recommended to the users individually. The cosine similarity is the cause of the angle between the two vectors where the vectors are non zero and the inner product space it is described as the dot product of the two vectors divide by by the product of the euclidean magnitude. In most cases cosine similarity is used to get preffered recommendations for users.

This method simply use the cosine distance between the vectors and then it uses similarity to calculate the score and then the preference of the user. For example movie with actors which define number of user likes and only few actors which a group of users don't like so we belie ve plotting a good sign angle between the user and the movie vectors which will generally be a large positive fraction, so angle is almost closely to be zero small distance of cosine will be present between the two vectors. Better metric somehow like the movie and cosine distance is large then the cosine similarity fails in this case we will approach in new method call decisio n tree to refine the recommender system. this method generally contains levels baby can appl y some conditions in a classification approach of refining the recommender system which try to find out if a user wants to what movie or not at all.

The advantages of the content based filtering are:

. Can recommend the unrated items.

. Can recommend the movies based on the ratings of the user

The disadvantages of the content based filtering are:

. Can't work on the new user hasn't red kidney movie act

. It can't make the user likes with the un -likes .

3.Collaborative based Filtering: Content based filtering suffer from various limitations which is only capable of the suggesting movies having only one type of users preferences and then unable to provide recommendations in case of genres . However collaborative filtering based system provides much complexibility in finding the record between the similarity of user and the the likes of the users having similar interest. For measuring the similarity of users views cosine similarity or pearson's correlation. Taking example in the below Matrix every row has a user with column corresponding to the movies having the same similarity it also has the ratings of different movies which the user have given to each movie has a target user.

All the the collaborative filtering in case of user based is simple but it has also drawbacks the biggest challenges that the choices of the users where is with time. Pre computing the Matrix orphan let the problem of lower performance. So we can use the item based collaborative filtering which basically considers the items based on the similarity with the items and that it it find the similar matches with the target users the same similarity coefficients suggest pearson's correlation or Cosine similarity can be used. Item based collaborative filtering is most static in nature . Like blow example only one user which has related both Matrix and Titanic so similarity which stands between them is only one . There may be cases where we have millions of users and the similarity between those two different movies is very high as they have same rank for the user who have rated them both.

In collaborative filtering try to find out the users have which have name interest and similar li kes. In this case we don't use features of the item to recommend it but we use the classificatio n of users into clusters of similar types and then seperate each cluster into the order of the pre ference of the user. we can also use the cosine distance here which takes into account the user s with the similar interest greater the cosine small angle between the two user. Here we simpl y use the utility matrix we can assign the zero value to the sparse columns forming the calcul ations easy. Item based Colaborative filtering is preferred in general because it takes into acc ount the movie instead of the number of users which further only make the classification of th e movies and user much easier. Hence the user based collaborative filtering is not preferred b ecause it's simply only takes the user's into account and ignore the sparse values which create s the issues in bringing out the performance of the recommender system.

	The Avengers	Sherlock	Transformers	Matrix	Titanic	Me Before You	Similarity(i, E)
A	2		2	4	5		NA
B	5		4			1	
C			5		2		
D		1		5		4	
E			4			2	1
F	4	5		1			NA

Since user A and F do not share any movie ratings in common with user E, their similarities with user E are not defined in Pearson Correlation. Therefore, we only need to consider user B, C, and D. Based on Pearson Correlation, we can compute the following similarity.

	The Avengers	Sherlock	Transformers	Matrix	Titanic	Me Before You	Similarity(i, E)
A	2		2	4	5		NA
B	5		4			1	0.87
C			5		2		1
D		1		5		4	-1
E			4			2	1
F	4	5		1			NA

Fig.- User Based Collaborative Filtering

So now we want our recommendation problem to be converted into a Optimisation problem . The most preferred common metric is a root mean square error(RMSE). Better the performance lower will be The RMSE value.

Advantages of collaborative filtering based systems are:

. It is simply content dependent

. It often reads the mind of people having same preferences

. Create real quality assessment of items.

Disadvantages of collaborative filtering based systems are:

. Early rater problem as the most common where the collaborative filtering method fails to provide ratings of the movie which has no user waiting.

. Sparsity problem is more common in this type of welding method where null values are in so much quantity that is difficult to find items which are rated by the majority of the people.

	The Avengers	Sherlock	Transformers	Matrix	Titanic	Me Before You
A	2		2	4	5	2.94*
B	5		4			1
C			5		2	2.48*
D		1		5		4
E			4			2
F	4	5		1		1.12*
Similarity	-1	-1	0.86	1	1	

Fig.-Item based Collaborative Filtering

4.Hybrid Based Filtering: It is simply a mixture of content based filtering and collaborative based filtering methods where we will take the input as the the userid and the title of the movie and the output will be e the similar movies shorted by the particular users based on the expected ratings. Expected ratings are calculated internally where the ideas from content and collaborative filtering are used to build a engine where movies are suggested to the particular user and then estimation of the ratings takes place

In the comparisons section below we will see how movies are determined through the hybrid technique of filtering where we have both used content based method as well as the collaborative based filtering method. It is clear that hybrid filtering method is is good in most of the cases and scenarios where it is difficult to distinguish or get the accuracy which the users can get the recommended movies.

```
def hybrid(userId, title):
    idx = indices[title]
    tmdbId = id_map.loc[title]['id']
    #print(idx)
    movie_id = id_map.loc[title]['movieId']

    sim_scores = list(enumerate(cosine_sim[int(idx)]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:26]
    movie_indices = [i[0] for i in sim_scores]

    movies = smd.iloc[movie_indices][['title', 'vote_count', 'vote_average', 'year', 'id']]
    movies['est'] = movies['id'].apply(lambda x: svd.predict(userId, indices_map.loc[x]['mov
ieId']).est)
    movies = movies.sort_values('est', ascending=False)
    return movies.head(10)
```

Hybrid Filtering Method

Organization

The following is a general outline of the report's structure:
Part 1: Describes the project's overall presentation, issue proclamation venture points, and scope.

Part 2: It provides an overview of current research in the topic. It explains in full all of the research, investigations, theories, and social gatherings that took place throughout the project.

Part 3: Discusses the project's framework and plan in order to forecast the correct outcome.

Part 4: Discusses the results and provides screenshots.

Part 5: Complete the project and submit a proposal for future work.

LITERATURE SURVEY

1.Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey
Authors Dheeraj Bokde , Sheetal Girase, Debajyoti Mukopadhyay

Publisher: Procedia Computer Science,

Year: 2015

This paper attempt present a comprehensive survey of Factorization model like Singular Value Decomposition address the challenges of Collaborative Filtering algorithms, which can be served as a roadmap for research and practice in this area.

2. Latent Factor Models for Web Recommender Systems

Authors: Bee-Chung Chen, Deepak Agarwal, Pradheep Elango, Raghu Ramakrishnan

Publisher: Yahoo! Research & Yahoo! Labs

This paper discuss about the model that

• uses feature-based regression to predict the initial point for online learning, and

• reduces the dimensionality of online learning

Rapidly update online models once new data is received:

• Fast learning: Low dimensional and easily parallelizable

• Online selection for the best dimensionality

[3] Matrix Factorization Techniques for Recommender Systems

3. Authors: Yehuda Koren, Yahoo Research, Robert Bell and Chris Volinsky, AT&T Labs---Research

Publisher: IEEE Computer Society Press Los Alamitos, CA, USA

Year: 2009
Used in Netflix Competitions

This paper discusses about matrix factorization techniques have become a dominant methodology within collaborative filtering recommenders.

Experience with datasets such as the Netflix Prize data has shown that they deliver accuracy superior to classical nearest-neighbour techniques. At the same time, they offer a compact memory-efficient model that systems can learn relatively easily. What makes these techniques even more convenient is that models can integrate naturally many crucial aspects of the data, such as multiple forms of feedback, temporal dynamics, and confidence levels.

4. Related Articles and Research Papers

1. In December 2017, Springer US published a research article titled Recommendation System USING K-MEANS CLUSTERING on "Active Learning in Recommender Systems." Lior Rokach and Bracha Shapira were the authors of the paper.

2. A research article titled "A content-based recommender system for computer science papers" was published in IJCRT in 2018.] are the writers of the articles. R.C. Guan, D.H. Wang, Y.C. Liang, D.Xu, X.Y. Feng

3. A study titled MOVREC utilising Machine Learning Techniques was published in SAIT. Ashrita Kashyap, Sunita. B, Sneha Srivastava, Aishwarya. PH, and Anup Jung Shah are the authors of the article.

4. Lopez-Nores, M., J. Garca-Duque, A. Fernandez- Vilas, R. P. Daz-Redondo, J. Bermejo-Mu noz, “A Flexible Semantic Inference Metho-dology to Reason about User Preferences in Knowledge-based Recommender Systems”, Knowledge-Based Systems, Vol.21, No.4(2016), 305～32

.

5. Gurpreet Singh, International Journal of Advanced Trends in Computer Applications (IJATCA)Volume 3, Number 7, July - 2016, pp. 11-14 ISSN: 2395-3519

SYSTEM DESIGN

Dataset

1) For Content and Collaborative Based Filtering:

- Kaggle provided the data set. The Movie Recommendation System uses it as a standard Dataset.
- We used the movie dataset from 'Movie Lens(Kaggle)' for the project.
- Movies and ratings are taken into account.
- Total of 9743 movies
- Total of 100147 ratings
- MovieLens users were chosen at random.
- A unique id is assigned to each user and movie

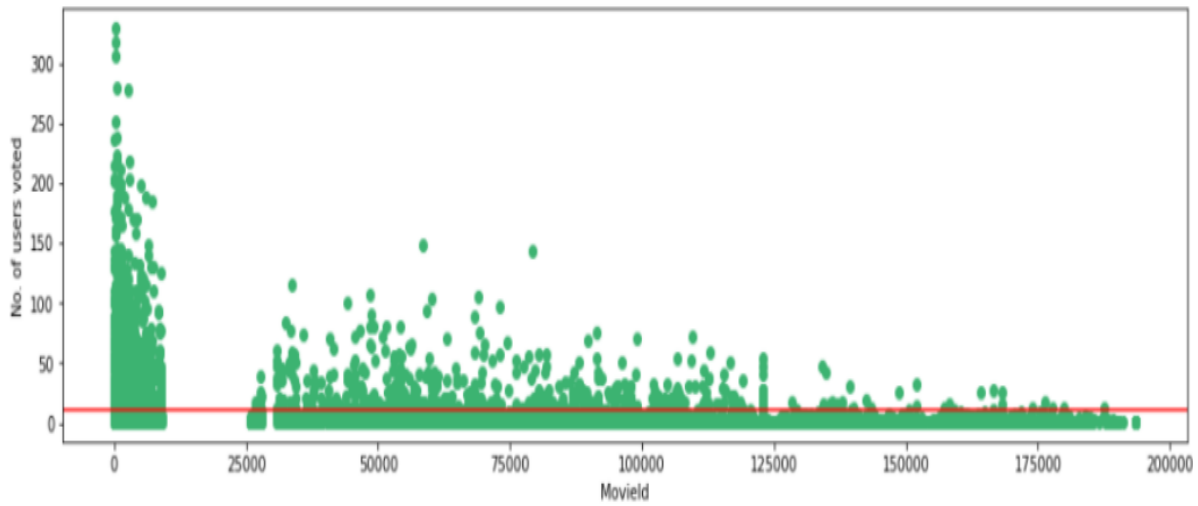
2) For Hybrid filtering method :

Consists of 26,000,000 ratings and 750,000 taag applications applied to 45,000 movies by 270,000 users

Ratings are from 1-5 scale and taken from Group Lens Officially.

Visualizing the no. of users Voted

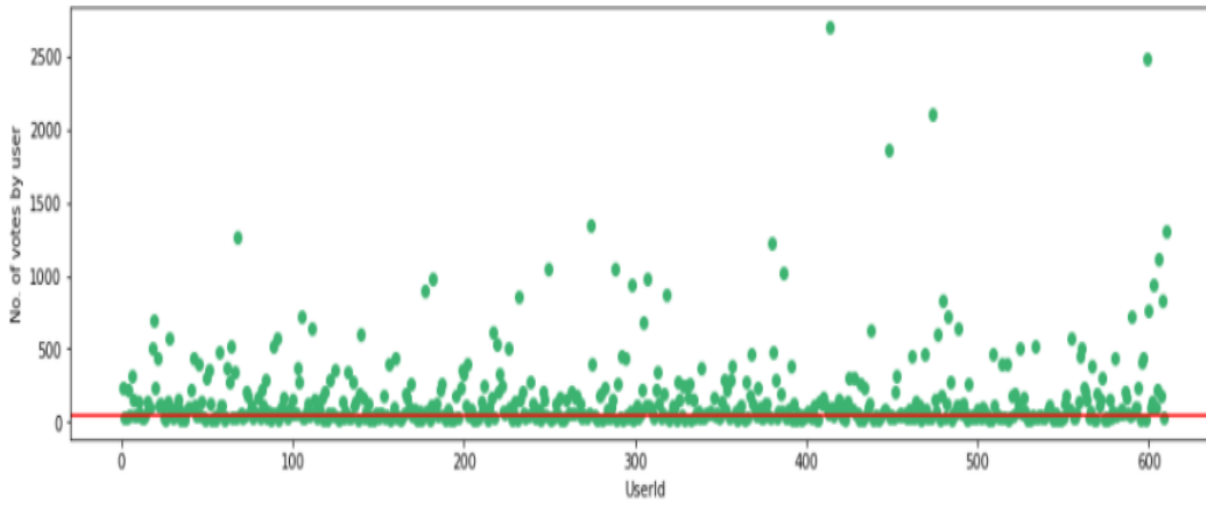
```
f,ax = plt.subplots(1,1,figsize=(16,4))
# ratings['rating'].plot(kind='hist')
plt.scatter(no_user_voted.index,no_user_voted,color='mediumseagreen')
plt.axhline(y=10,color='r')
plt.xlabel('MovieId')
plt.ylabel('No. of users voted')
plt.show()
```



Visualization the no. of Users Voted

Visualizing the no. of Votes by User

```
f,ax = plt.subplots(1,1,figsize=(16,4))
plt.scatter(no_movies_voted.index,no_movies_voted,color='mediumseagreen')
plt.axhline(y=50,color='r')
plt.xlabel('UserId')
plt.ylabel('No. of votes by user')
plt.show()
```

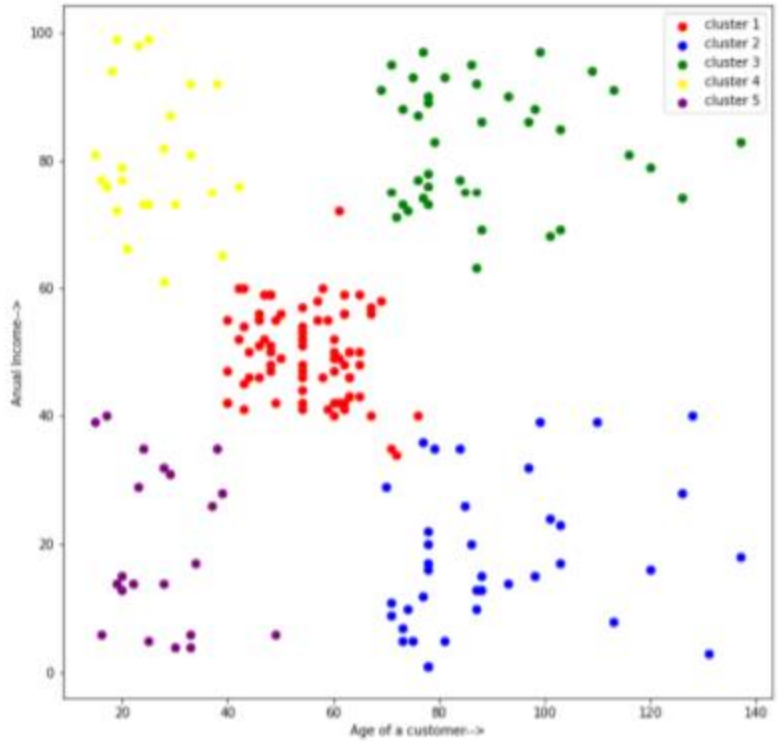


Visualizing the no. of Votes by Users

Algorithms Used

K-Means Algorithm:

K means clustering algorithm just simply create the cluster inside a cluster which have same matching features in between them. The degree of closeness defines the the similarity basis as 2 how 2 points are related to each other. In this algorithm re simplify and centroid and then repeat the the process until optimum centroid is is calculated or found . It simply determines the best value for the K Centre points by iterative process and then assign each data point to the closest nearest centre of K value.The number of clusters found from the data is denoted simply by the notation 'K'. Simple unsupervised ml algorithm categorize the data points into subgroups even from the very less information about the data.



K-Mean algorithm

Cosine Similarity

The cosine similarity is the cause of the angle between the two vectors where the vectors are non zero and the inner product space it is described as the dot product of the two vectors divide by by the product of the euciledian magnitude. Smaller the angle higher the similarity so the cosine similarity is much much more preferable over the equilibrium distance because angle is smaller in case of cosine similarity.

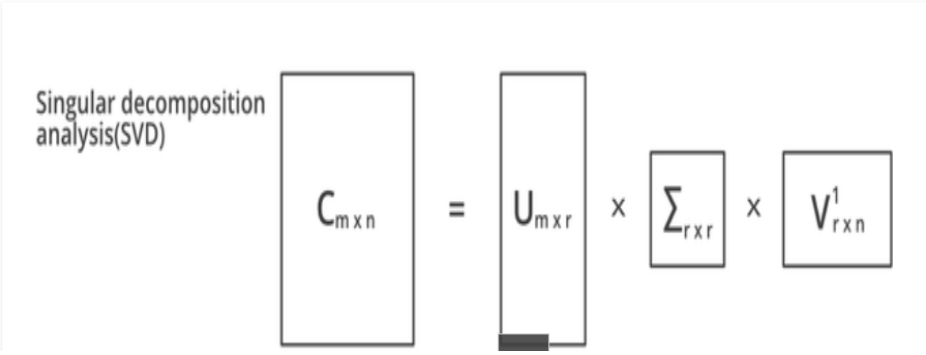
Singular Value Decomposition (SVD)

SVD is basically matrix factorization of a matrix into 3 matrices. It hols properties and convey some geometrical as well as the theoretical outputs in a linear transformation the mathematical way of representing. A SVD of a given Matrix is given by the formula:

A=UWV^T

where:

- U: *mxn* matrix of the orthonormal eigenvectors of AA^T.
- V^T: transpose of a *nxn* matrix containing the orthonormal eigenvectors of A^{T}A.
- W: a *nxn* diagonal matrix of the singular values which are the square roots of the eigenvalues of $A^T A$.



SVD Algorithm

RMSE (Root Mean Square Error)

RMSE is just basically the standard deviation of the predicted errors. Residue which are the measure of the regression where is the data points however it also shows this widespread of the residuals in the data points and also finds out the the best fit in the data .It is also used in forecasting ,regression analysis to get the verified results of the experiments . Better the performance lower will be The RMSE value.

$$\mathbf{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}}$$

$\mathbf{RMSD}$ = root-mean-square deviation

i = variable i

N = number of non-missing data points

x_i = actual observations time series

$\hat{x}_i$ = estimated time series

Hardware and Software Requirements

● 4.2 GB RAM

● MS Window 7 and above Software Requirements

● Jupyter Notebook

● Wamdp Server

● Visual Studio Code

● Sublime Text

● MYSQL

CONCEPTS REQUIREMENTS

• Machine Learning Algorithms

• Data Pre-processing Functions and tools

• scikit-learn

• seaborn

• knowledge of K-Means clustering

.NumPy is a Python programming language.

• Panda bears

• matplotlib (matplotlib)

• Cleaning of data

•64bit processors are required

PERFORMANCE ANALYSIS

Comparisons and Results

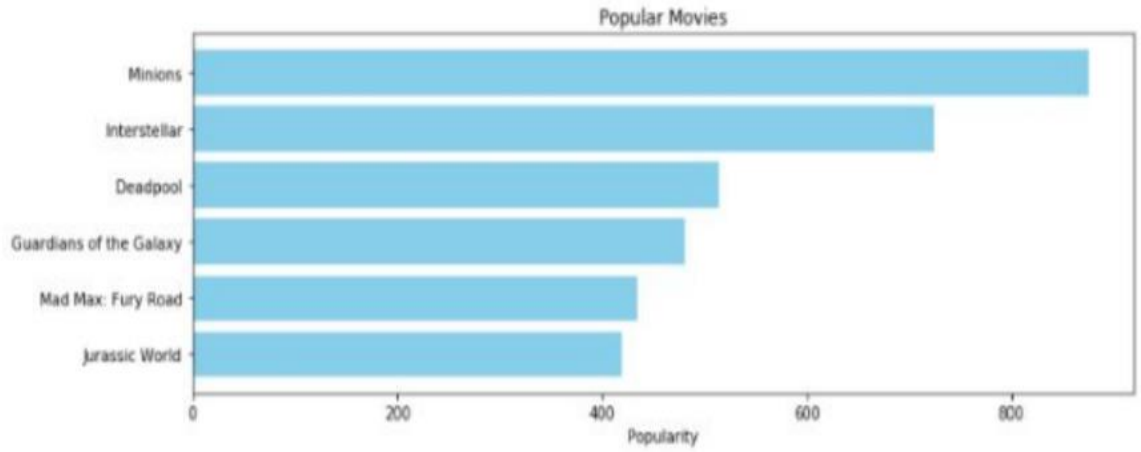
1.Demographic Filtering:

Filtering the cause of short in the movies recommended which are best to the users based on the metric scores and personalized and generalized recommendations are recommended to the every users on the basis of the popularity which are generally like by the average audience.

```
pop= df2.sort_values('popularity', ascending=False)
import matplotlib.pyplot as plt
plt.figure(figsize=(12,4))

plt.barh(pop['title'].head(6),pop['popularity'].head(6), align='center',
        color='skyblue')
plt.gca().invert_yaxis()
plt.xlabel("Popularity")
plt.title("Popular Movies")
```

```
Text(0.5,1,'Popular Movies')
```



Demographic Filtering Output

2 Content Based Filtering:

User profile holds the content that is is much more matching to use the form of the features. The previous actions or for the feedback is taken into account a generally takes into account the description of the content that has been edited by the users of different choices.

```
In [16]: get_recommendations('The Dark Knight Rises')

Out[16]: 65                           The Dark Knight
         299                           Batman Forever
         428                           Batman Returns
         1359                                 Batman
         3854    Batman: The Dark Knight Returns, Part 2
         119                            Batman Begins
         2507                               Slow Burn
         9           Batman v Superman: Dawn of Justice
         1181                                    JFK
         210                           Batman & Robin
         Name: title, dtype: object

In [17]: get_recommendations('The Avengers')

Out[17]: 7                Avengers: Age of Ultron
         3144                            Plastic
         1715                            Timecop
         4124                 This Thing of Ours
         3311              Thank You for Smoking
         3033                     The Corruptor
         588     Wall Street: Money Never Sleeps
         2136          Team America: World Police
         1468                       The Fountain
         1286                        Snowpiercer
         Name: title, dtype: object
```

Content Based Filtering Output

3.) Collaborative based Filtering: In the collaborative filtering behaviour used here item based collaborative filtering where we have taken 3 different types of metrics and varied the results accordingly. Brief comparison of three of the metric used in the collaborative filtering are are shown with the movies recommended from them based on the the bounds set to the number of users and a number of ratings by a user to a movie.

Metric =”Cosine” Cosine similarity, or the cosine kernel, **computes similarity as the normalized dot product of X and Y**: K(X, Y) = <X, Y> / (||X||*||Y||) On L2-normalized data, this function is equivalent to linear_kernel.

```
knn = NearestNeighbors(metric='cosine', algorithm='brute', n_neighbors=20, n_jobs=-1)
knn.fit(csr_data)
```

```
NearestNeighbors(algorithm='brute', metric='cosine', n_jobs=-1, n_neighbors=20)
```

```
def get_movie_recommendation(movie_name):
    n_movies_to_reccomend = 10
    movie_list = movies[movies['title'].str.contains(movie_name)]
    if len(movie_list):
        movie_idx= movie_list.iloc[0]['movieId']
        movie_idx = final_dataset[final_dataset['movieId'] == movie_idx].index[0]
        distances , indices = knn.kneighbors(csr_data[movie_idx],n_neighbors=n_movies_to_reccomend+1)
        rec_movie_indices = sorted(list(zip(indices.squeeze().tolist(),distances.squeeze().tolist())),key=lambda x: x[1])[:0:-1]
        recommend_frame = []
        for val in rec_movie_indices:
            movie_idx = final_dataset.iloc[val[0]]['movieId']
            idx = movies[movies['movieId'] == movie_idx].index
            recommend_frame.append({'Title':movies.iloc[idx]['title'].values[0],'Distance':val[1]})
        df = pd.DataFrame(recommend_frame,index=range(1,n_movies_to_reccomend+1))
        return df
    else:
        return "No movies found. Please check your input"
```

```
get_movie_recommendation('chain')
```

	Title	Distance
1	Social Network, The (2010)	0.365830
2	The Hunger Games (2012)	0.363921
3	Deadpool (2016)	0.354786

Collaborative Based Filtering Method(Metric =Cosine) Output

Metric=”Cityblock”-> This function simply returns the valid pairwise distance metrics. It exists to allow for a description of the mapping for each of the valid strings.

The function for the cityblock is as below

‘cityblock’ =metrics.pairwise.manhattan_distances

```
[28] knn = NearestNeighbors(metric='cityblock', algorithm='brute', n_neighbors=20, n_jobs=-1)
     knn.fit(csr_data)

     NearestNeighbors(algorithm='brute', metric='cityblock', n_jobs=-1,
                      n_neighbors=20)
```

```
[29] def get_movie_recommendation(movie_name):
         n_movies_to_reccomend = 10
         movie_list = movies[movies['title'].str.contains(movie_name)]
         if len(movie_list):
             movie_idx= movie_list.iloc[0]['movieId']
             movie_idx = final_dataset[final_dataset['movieId'] == movie_idx].index[0]
             distances , indices = knn.kneighbors(csr_data[movie_idx],n_neighbors=n_movies_to_reccomend+1)
             rec_movie_indices = sorted(list(zip(indices.squeeze().tolist(),distances.squeeze().tolist())),key=lambda x: x[1])[:0:-1]
             recommend_frame = []
             for val in rec_movie_indices:
                 movie_idx = final_dataset.iloc[val[0]]['movieId']
                 idx = movies[movies['movieId'] == movie_idx].index
                 recommend_frame.append({'Title':movies.iloc[idx]['title'].values[0],'Distance':val[1]})
             df = pd.DataFrame(recommend_frame,index=range(1,n_movies_to_reccomend+1))
             return df
         else:
             return "No movies found. Please check your input"
```

```
get_movie_recommendation(' Games')
```

	Title	Distance
1	Bull Durham (1988)	0.570608
2	Witness (1985)	0.565722

Collaborative Based Filtering Method(Metric =Cityblock) Output

Metric=”Minkowski”-> It is a metric intended for real-valued vector spaces. We can calculate Minkowski distance only in a normed vector space, which means in a space where distances can be represented as a vector that has a length and the lengths cannot be negative.

```
[13]  (0, 2)        3
      (1, 0)        4
      (1, 4)        2
      (2, 4)        1

[14] csr_data = csr_matrix(final_dataset.values)
     final_dataset.reset_index(inplace=True)

[33] knn = NearestNeighbors(metric='minkowski', algorithm='brute', n_neighbors=20, n_jobs=-1)
     knn.fit(csr_data)

     NearestNeighbors(algorithm='brute', n_jobs=-1, n_neighbors=20)

     def get_movie_recommendation(movie_name):
         n_movies_to_reccomend = 10
         movie_list = movies[movies['title'].str.contains(movie_name)]
         if len(movie_list):
             movie_idx= movie_list.iloc[0]['movieId']
             movie_idx = final_dataset[final_dataset['movieId'] == movie_idx].index[0]
             distances , indices = knn.kneighbors(csr_data[movie_idx],n_neighbors=n_movies_to_reccomend+1)
             rec_movie_indices = sorted(list(zip(indices.squeeze().tolist(),distances.squeeze().tolist())),key=lambda x: x[1])[:0:-1]
             recommend_frame = []
             for val in rec_movie_indices:
                 movie_idx = final_dataset.iloc[val[0]]['movieId']
                 idx = movies[movies['movieId'] == movie_idx].index
                 recommend_frame.append({'Title':movies.iloc[idx]['title'].values[0],'Distance':val[1]})
             df = pd.DataFrame(recommend_frame,index=range(1,n_movies_to_reccomend+1))
             return df
         else:
             return "No movies found. Please check your input"
```

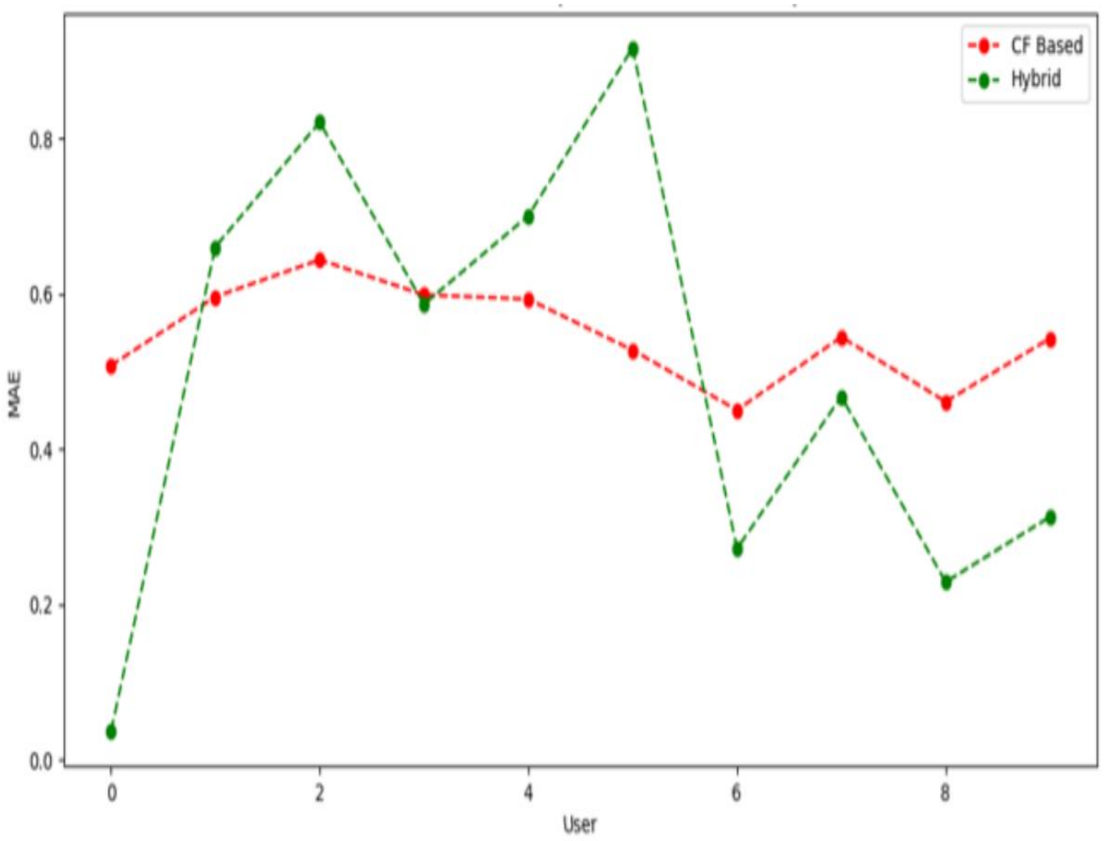
Collaborative Based Filtering Method(Metric =Minkowski) Output

4) Hybrid Based filtering: It is simply a mixture of content based filtering and collaborative based filtering methods where we will take the input as the the userid and the title of the movie and the output will be e the similar movies shorted by the particular users based on the expected ratings. Expected ratings are calculated internally where the ideas from content and collaborative filtering are used to build a engine where movies are suggested to the particular user and then estimation of the ratings takes place.

```
hybrid(500, 'Avatar')
```

	title	vote_count	vote_average	year	id	est
8401	Star Trek Into Darkness	4479.0	7.4	2013	54138	3.238226
974	Aliens	3282.0	7.7	1986	679	3.203066
7265	Dragonball Evolution	475.0	2.9	2009	14164	3.195070
831	Escape to Witch Mountain	60.0	6.5	1975	14821	3.149360
1668	Return from Witch Mountain	38.0	5.6	1978	14822	3.138147
1376	Titanic	7770.0	7.5	1997	597	3.110945
522	Terminator 2: Judgment Day	4274.0	7.7	1991	280	3.067221
8658	X-Men: Days of Future Past	6155.0	7.5	2014	127585	3.043710
1011	The Terminator	4208.0	7.4	1984	218	3.040908
2014	Fantastic Planet	140.0	7.6	1973	16306	3.018178

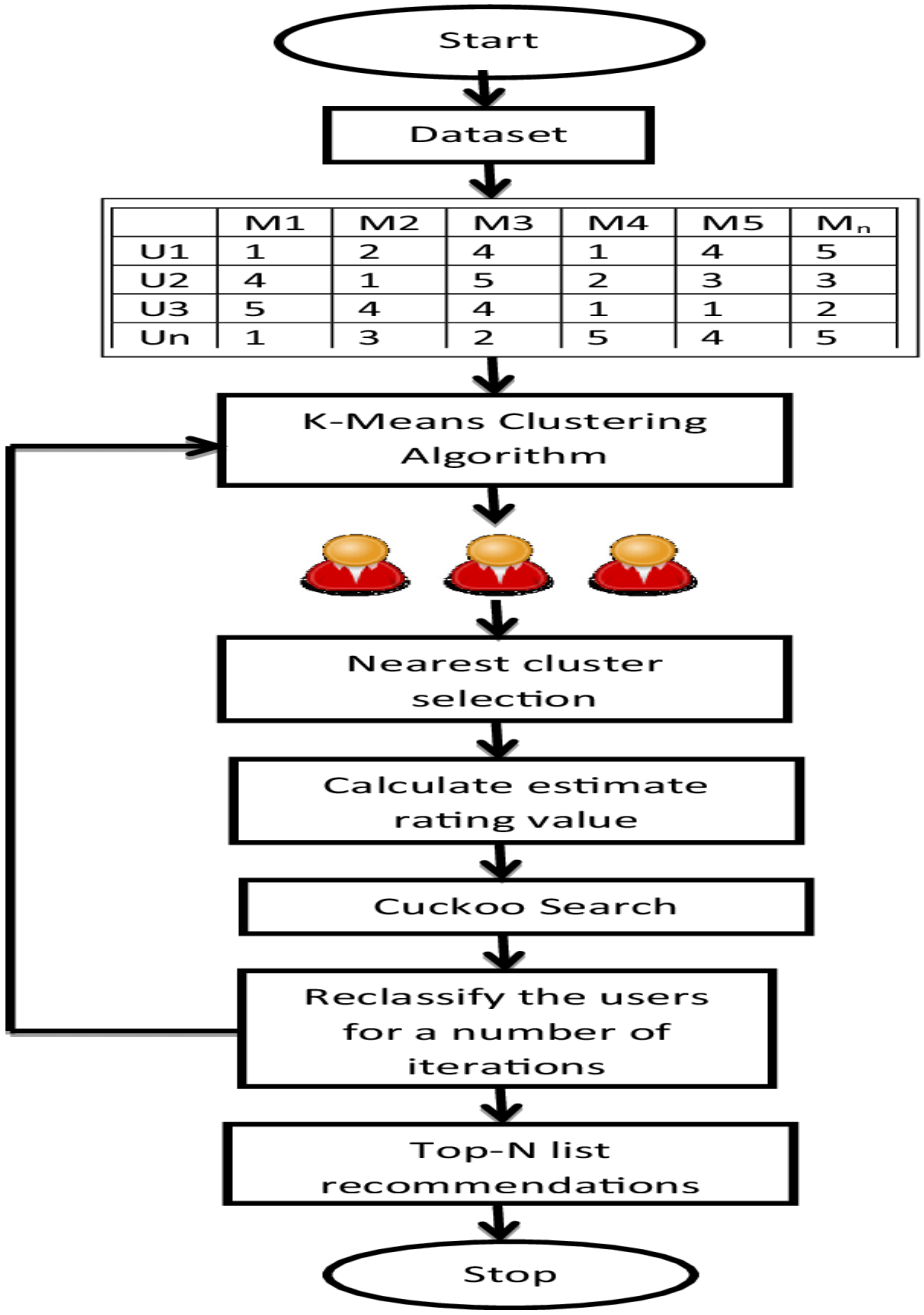
Hybrid Based Filtering Output



Comparitative output

Here we can see that the hybrid filtering technique stands good in in overcoming the the issues faced in the content based filtering technique and the collaborative based filtering method we can generalize from the method of root mean square error that the value for hybrid filtering method is less so performance is higher for hybrid case. While we can say that collaborative filtering technique stands good only in terms of the quality perspective but when it comes to both qualitative and quantitative achievement of the result will prefer hybrid filtering technique where the all flaws. While content based filtering technique only outperform the collaborative in terms of similarity e the collaborative filtering technique can you recommend one item to the other item of the similar interest, the overall flaws can be removed by the hybrid based collaborative filtering with two or more examination techniques are combined to gain the better performance with the less possibilities of drawback of this system. In general in case of hybrid filtering techniques the collaborative filtering technique is combined with some other type of filtering technique to avoid the ramp up problem and thus it outperforms the the major drawbacks of the system in case if we prefer to use single content based or collaborative filtering technique.

So hybrid filtering recommender simply allows the user to select his own choices from a given data which contain some attributes or some set of values which contain user specific values and then recommend then the best movie which is based on the similarities based calculating the the accumulator weight and then applies the algorithm which is in our case K mean algorithm. Expected ratings are calculated internally where the ideas from content and collaborative filtering are used to build a engine where movies are suggested to the particular user and then estimation of the ratings takes place. So in the the process of getting different results from different algorithms and techniques hybrid approach is preferred to be better one between the content and collaborative filtering techniques which simply overcomes the drawbacks of the the single algorithm and then tries to improve the performance of the overall recommender system. Moreover some other techniques like classification clustering can be used to get the best of the recommendations which would simply increase our accuracy for the recommender system. So the the better performance can be achieved in the end by a hybrid based filtering technique which is why it is most preferable over the other two techniques.



System Framework

CONCLUSIONS

So for implementing a hybrid technique for content and collaborative based filtering we take into account the hybrid approach which improves the overall performance of the system and then recommended movies to the users as per the choice in a much better way than the other two system of recommendation lower the mean average error, it further increases the the accuracy of the recommender system and then we can use h system of recommendation for future uses as well in a better way. We also have some system computational bounds or limitations to perform the recommender system on the large dataset here but we have done enough to distinguish between the various recommender system which finally put hybrid system of recommendation on the top of the all. Hence we can conclude that hybrid based filtering helps in getting the system fragmentation much efficient enhance the Precision of the overall system and and no doubt it is the the the mixture of both content in collaborative based filtering methods where even if one method fails The Other takes over and maintains the overall accuracy of the the system and and simply increase the performance overall all around.

Overall flaws can be removed by the hybrid based collaborative filtering with two or more examination techniques are combined to gain the better performance with the less possibilities of drawback of this system. In general in case of hybrid filtering techniques the collaborative filtering technique is combined with some other type of filtering technique to avoid the ramp up problem and thus it outperforms the the major drawbacks of the system in case if we prefer to use single content based or collaborative filtering technique.

While we can say that collaborative filtering technique stands good only in terms of the quality perspective but when it comes to both qualitative and quantitative achievement of the result will prefer hybrid filtering technique where the all flaws.

In the end hybrid system stands alone the better performer for Recommending movies to the users of different taste, choices or similarities.

Future Work

1.) In case of content based filtering method we can look up on the cast and crew also where we have only considered the genre and also we can see at the movies are compatible or not.
2.) Comparison of collaborative filtering based approaches and different kind of similarity measurements would be a good one for the recommender system
3.) We can use matrix factorization for calculating the number of factors involved.
4.) We can also apply deep learning techniques to for the the enhance the recommender system and optimising the efficiency of the system.
5.) We can work on different areas such as video some books aur even recommending some songs to the users of the mobile phones based on the platforms of the different apps available on the Play Store
6.) Various techniques such as clustering classification can be used to get the better version of our recommender system which for the the enhance the accuracy of the overall model**.**

REFERENCES

1] Hirdesh Shivhare, Anshul Gupta and Shalki Sharma, "Recommender system using fuzzy c-means clustering and genetic algorithm based weighted similarity measure" Communication and Control. IEEE International Conference on Computer, 2015.

[2] Manoj Kumar, D.K. Yadav, Ankur Singh and Vijay Kr. Gupta. "A Movie Recommender System: MOVREC, International Journal of Computer Applications (0975-8887) Volume 124-No.3, 2015.

[3] RyuRi Kim, Ye Jeong Kwak, HyeonJeong Mo, Mucheol Kim, Seungmin Rho,Ka Lok Man, Woon Kian Chong "Trustworthy Movie Recommender System with Correct Assessment and Emotion Evaluation". Proceedings of the International MultiConference of Engineers and Computer Scientists Vol II, 2015

[4] Zan Wang, Xue Yu*, Nan Feng, Zhenhua Wang, "An Improved Collaborative Movie Recommendation System using Computational Intelligence" Journal of Visual Languages & Computing. Volume 25, Issue 6, 2014

[5] Debadrita Roy, Arnab Kundu, "Design of Movie Recommendation System by Means of Collaborative Filtering". International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 4, 2013.

[6] E. Amolochitis, Algorithms and applications for academic search, recommendation and quantitative association rule mining. Gistrup, Denmark: River Publishers, 2018.

[7] T. Brodt, Collaborative Filtering. Saarbrücken: VDM, Müller, 2010.

[8] M. Jalali, H. Gholizadeh and S. Hashemi Golpayegani, "An improved hybrid recommender system based on collaborative filtering, content based, and demographic filtering", International Journal of Academic Research, vol. 6, no. 6, pp. 22-28, 2014.

[9] C. Li and K. He, "CBMR: An optimized MapReduce for item-based collaborative filtering recommendation algorithm with empirical analysis", Concurrency and Computation: Practice and Experience, vol. 29, no. 10, p. e4092, 2017.

[10] Y. Ng, "MovRec: a personalized movie recommendation system for children based on online movie features", International Journal of Web Information Systems, vol. 13, no. 4, pp. 445-470, 2017.

[11] A. Roy and S. Ludwig, "Genre based hybrid filtering for movie recommendation engine", Journal of Intelligent Information Systems, vol. 56, no. 3, pp. 485-507, 2021.
N. Shahabi and F. Najian, "A New Strategy in Trust-Based Recommender System using K-Means Clustering", International Journal of Advanced Computer Science and Applications, vol. 8, no. 9, 2017.

[12] S. Agrawal and P. Jain, "About Performance Evaluation of the Movie Recommendation Systems", International Journal of Computer Applications, vol. 158, no. 2, pp. 7-10, 2017.

[13] Y. Patil and D. Karandam, "COLLABORATIVE FILTERING APPROACHES FOR MOVIE RECOMMENDATION SYSTEM USING PROBABILISTIC RELATIONAL MODEL", International Journal of Advance Engineering and Research Development, vol. 2, no. 03, 2015.

[14]P. Sharma and L. Yadav, "MOVIE RECOMMENDATION SYSTEM USING ITEM BASED COLLABORATIVE FILTERING", International Journal of Innovative Research in Computer Science & Technology, vol. 8, no. 4, 2020.

[15]M. Gogri, D. Chheda and V. Solani, "Movie Recommendation Using Deep Learning with Hybrid Approach", Aksh - The Advance Journal, vol. 1, no. 2, pp. 1-4, 2020. Available: 10.51916/aksh.2020.v01i02.001.