International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-6, March 2020

Implementation of Popular Techniques for Movie Recommendations

G. Naga Sujini, D. Gyana Deepika

Abstract: ***In today's world where there is a plethora of movies to choose from in any medium, recommendation systems play a crucial role in reducing the search time for the movie watchers by recommending them the most relevant movies that they would most probably like. There are a wide range of approaches and techniques which are used for recommending movies. While some techniques reflect the current trend and popularity of movies, others are able to capture and analyze the past behavior of the viewer and make recommendations accordingly. Recommendation systems are an integral part of companies such as Netflix, Amazon Prime, Hulu and various others to ensure that their customers have a pleasant experience which in turn would boost the company's profits. This study discusses and analyses the various approaches and techniques used for the recommendation of movies.***

Keywords: ***Collaborative filtering, Content Filtering, Recommendation systems***

I. INTRODUCTION

Recommendation systems help the user or customer to narrow down the items such as movies, books etc. to a few items from a very large item database which are well suited to the preferences of the user. These systems have become exceedingly crucial to companies as their main focus is to draw as many customers as possible and make them consume their content. In this context, recommendation systems are able to improve the experience of customers by matching them with the content they are most likely to prefer and enjoy to watch.

Recommendation systems use prediction algorithms and techniques to suggest the best items suitable to the user. Machine Learning techniques and Artificial Intelligence methods such as neural nets are being used in systems to increase the efficiency and accuracy of the recommendations produced. The dynamic nature of the database and the ever increasing number of users do pose a challenge to the recommender systems in all fields. New approaches and research are needed to cope with the increasing size of the database and effectively provide with high quality and accurate recommendations.

Manuscript received on February 10, 2020.
Revised Manuscript received on February 20, 2020.
Manuscript published on March 30, 2020.
* Correspondence Author
G. Naga Sujini*, Assistant Professor, Department of Computer Science & Engineering, Mahatma Gandhi Institute of Technology, affiliated to JNTU-H. Hyderabad, India.
D. Gyana Deepika, Department of Computer Science Engineering from Mahatma Gandhi Institute of Technology affiliated to JNTU-H., Hyderabad, India.

Many statistical, mathematical and artificial intelligence based techniques are being developed and employed in many fields such as e-commerce, media and various others. Commonly used techniques include content filtering, collaborative filtering and popularity based filtering. Algorithms such as matrix factorization are being used by companies like Netflix to make appropriate recommendations to its users or customers.

II. RELATED WORK

In the study written by BasiliyosTilahunBetru [1], recommendation system techniques like content based, collaborative filtering, hybrid based methods and deep learning approaches for recommendations are discussed. Memory and model based Collaborative filtering is explained and the various algorithms that can be used for each type are discussed. Some works related to using deep learning such as Restricted Bottzmann Machines, Collaborative deep learning systems and deep content-based music recommendation system are analysed and summarized. It presents the idea that deep learning will enable a model to learn different features of users which would lead to an improvement of accuracy.

DaniarAsanov [2] discusses traditional approaches like content based, collaborative (user based and item based) and hybrid based approaches. It explains about new modern approaches like context aware approach, semantic based approach, Cross domain based approach. It also discusses and explains the challenges such as cold start problem, sparsity problem, trust, scalability, privacy problems faced during the development of recommendation systems.

In this study by Gediminas Adomavicius1 and Alexander Tuzhilin [3] describes about the current generation of recommender systems techniques. This study discusses various recommendation approaches and describes about various limitations and extensions possible to the recommendation systems. It discusses about New User Problem in which the system must provide correct recommendation by using previous ratings, clicks, likes, dislikes and feedback to the customer who is new to online marketing database.

In the study by J. Ben Schafer [4], Collaborative filtering based algorithms which produces explainable and relevant results in various domains are discussed. Data Mining techniques are implemented in the hybrid systems with which the accuracy and relevancy of the recommenders are improved according to the user's needs. The study mostly focuses on the impact of data mining techniques on recommenders and their use in improving the marketing strategy.

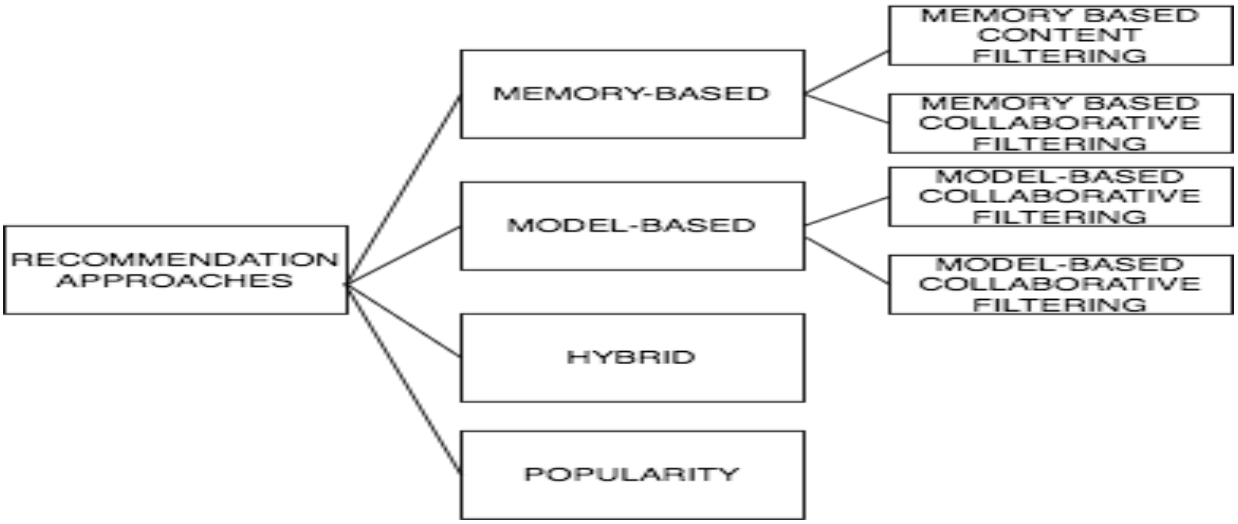

Figure 1: Overview of recommendation approaches

In the study presented by Greg Linden [5], in which they present an algorithm that produces recommendations in real-time, scales to massive data sets, and generates high quality recommendations. They discuss approaches such as traditional collaborative filtering, cluster models and search based models. The implementation of these algorithms can be used for amazon.com is briefly explained. It has been said that item-to-item collaborative filtering is able to react to the changes in user's data and make compelling recommendations to all users. They also discuss how these various models fare with respect to scalability.

III. VARIOUS APPROACHES USED FOR RECOMMENDATIONS

A. Popularity Based Approach

This approach works by analyzing and recommending the movies based upon certain metrics that relate to the popularity of a movie. Metrics or features such as the number of views and positive ratings, likes, Box Office collections, number of positive social media mentions etc. can be used to determine the popularity of a movie.This approach can be easily implemented and is based on the assumption that the most viewed or most popular movies are usually liked by everyone. Popularity based recommenders are a good first step towards recommendation of movies.

As the recommendations are dependent on the popular opinion, this approach does have a few disadvantages. One such disadvantage is the non-personalization of the recommendations. The same set of movies are recommended to everyone irrespective of their likes and dislikes. Due to the generalization of the recommendations, this approach will not cater specifically to individual preferences.

B. Memory Based Content Filtering

The essence of Content based approach is recommending the movies that are similar to the ones the user has liked previously. The similarity between two movies can be measured by various factors such as the genre, star cast, director, production house etc.. There can be various ways in which the similarity of two movies can be measured such as cosine similarity, pearson correlation. Cosine similarity finds how two vectors are related to each other using measuring cosine angle between these vectors.

C. Memory Based Collaborative Filtering

In this approach we find similar users based upon similarity measures such as cosine similarity, Pearson correlation and take weighted average of the ratings. The advantage of this approach is the easy creation and explanation of results. The drawback of this approach is that the performance reduces when data is sparse that is It becomes non-scalable.

D. Model Based Collaborative Filtering

In this approach we use machine learning techniques to find user ratings of unrated items. Such techniques include PCA, SVD, Neural Nets, Matrix Factorization and others. The advantage of this approach is that dimensionality reduction deals with the missing/sparse data. The drawback of this approach is that the inference is intractable because of the hidden/latent factors.

E. Hybrid Filtering Approach

This approach combines the results of other approaches and gives recommendations respectively. One of the way is to compute the average of the similarity scores computed using collaborative filtering and content based filtering and recommend accordingly.

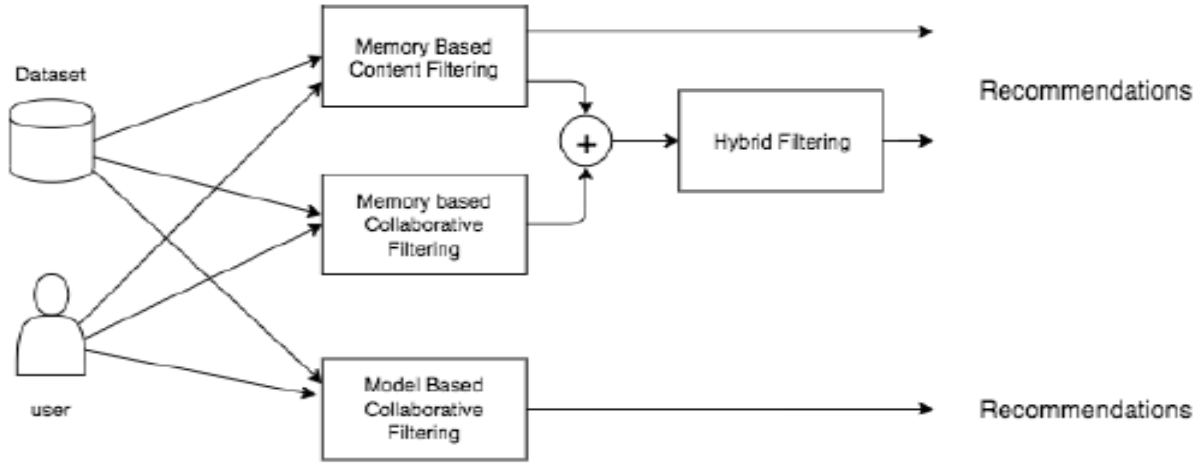

Figure 2: Design of the recommendation system

IV. SYSTEM DESIGN

The design of the system involves four modules as shown in Figure 4 :-

i) Memory based Content Filtering
ii) Memory based Collaborative filtering
iii) Hybrid Filtering
iv) Model based Collaborative filtering

For memory based content and collaborative filtering a matrix will be computed whose values can be used to for recommendations. The hybrid approach uses the average of content based and collaborative based filtering for recommendations. Each of these modules will output a list of recommendations based on the internal logic used by each of the module.

V. IMPLEMENTATION

The dataset used for this purpose is the MovieLens 20M Dataset created by the GroupLens Research Group. This dataset contains 20000263 ratings and 465564 tag applications across 27278 movies. The data is contained in six files -genome-scores.csv, genome tags.csv, links.csv, movies.csv, ratings.csv and tags.csv.

For item content filtering, the data sets used are movies.csv and tags. csv. Using these two sets, we compile all the tags given to movies by various users, genres, descriptions into a column called metadata. We then apply Tf-idf (term frequency- inverse document frequency) transformer on this column to obtain a vector of features. To reduce the size of the feature vectors, truncated singular value decomposition is applied such that 50% of the variance is retained. Then cosine similarity measure is used to compute the similarity scores.

For memory based collaborative filtering, the dataset ratings.csv is used to form items as vectors of input rates. The same measures as stated above are used to reduce the dimensionality of the matrix.

For hybrid filtering, the average of the content based filtering and collaborative filtering is used to make the recommendations.

In Figure 3, we are able to get the recommendations based according to memory based collaborative filtering, content based filtering and hybrid filtering for the input movie 'Strada, La (1954)'.

For model based collaborative filtering, we implement the Singular Value Decomposition algorithm that is present in Surprise library in python. We then calculate the RMSE (Root mean square error) to find out the accuracy.

VI. RESULTS

```
from sklearn.metrics.pairwise import cosine_similarity
# take the latent vectors for a selected movie from both content
# and collaborative matrixes
a_1 = np.array(latent_matrix_1_df.loc['Strada, La (1954)']).reshape(1, -1)
a_2 = np.array(latent_matrix_2_df.loc['Strada, La (1954)']).reshape(1, -1)

# calculate the similartity of this movie with the others in the list
score_1 = cosine_similarity(latent_matrix_1_df, a_1).reshape(-1)
score_2 = cosine_similarity(latent_matrix_2_df, a_2).reshape(-1)

# an average measure of both content and collaborative
hybrid = ((score_1 + score_2)/2.0)

# form a data frame of similar movies
dictDf = {'content': score_1 , 'collaborative': score_2, 'hybrid': hybrid}
similar = pd.DataFrame(dictDf, index = latent_matrix_1_df.index )

#sort it on the basis of either: content, collaborative or hybrid,
# here : content
similar.sort_values('content', ascending=False, inplace=True)

similar[1:].head(11)
```

	content	collaborative	hybrid
Burnt by the Sun (Utomlyonnye solntsem) (1994)	0.783687	0.347234	0.565460
Babette's Feast (Babettes gæstebud) (1987)	0.745094	0.610579	0.677836
Purple Noon (Plein soleil) (1960)	0.730353	0.528732	0.629543
Mon Oncle (My Uncle) (1958)	0.662756	0.946326	0.804541
Sundays and Cybele (Les dimanches de Ville d'Avray) (1962)	0.640997	0.592128	0.616563
Kolya (Kolja) (1996)	0.621453	0.326573	0.474013
Nights of Cabiria (Notti di Cabiria, Le) (1957)	0.612939	0.834067	0.723503
Moscow Does Not Believe in Tears (Moskva slezam ne verit) (1979)	0.611397	0.695872	0.653635
Journey of Hope (Reise der Hoffnung) (1990)	0.608204	0.597546	0.602875
Hamlet (1948)	0.592787	0.496560	0.544674
Country Girl, The (1954)	0.590681	0.453769	0.522225

Figure 3: Code Snippet and results for input 'Strada, La (1954)'

	hybrid
Black Swan (2010)	0.739713
Mr. Nobody (2009)	0.723242
Drive (2011)	0.717629
Shutter Island (2010)	0.711278
Eternal Sunshine of the Spotless Mind (2004)	0.701873
Tron: Legacy (2010)	0.700740
Pan's Labyrinth (Laberinto del fauno, El) (2006)	0.700417
Her (2013)	0.698461
Imaginarium of Doctor Parnassus, The (2009)	0.695913
Moon (2009)	0.694557
Ink (2009)	0.693056

Figure 4: Results for movie 'Inception (2010)'

In the above Figure 4, we obtain the recommendations for the input movie 'Inception (2010)' according to the decreasing order of score obtained from the hybrid filtering approach.

```
from surprise import Dataset, Reader, SVD, accuracy
from surprise.model_selection import train_test_split

# instantiate a reader and read in our rating data
reader = Reader(rating_scale=(1, 5))
data = Dataset.load_from_df(ratings_f[['userId','movieId','rating']], rea

# train SVD on 75% of known rates
trainset, testset = train_test_split(data, test_size=.25)
algorithm = SVD()
algorithm.fit(trainset)
predictions = algorithm.test(testset)

# check the accuracy using Root Mean Square Error
accuracy.rmse(predictions)
RMSE: 0.7724

# check the preferences of a particular user
user_id = 7010
predicted_ratings = pred_user_rating(user_id)
pdf = pd.DataFrame(predicted_ratings, columns = ['movies','ratings'])
pdf.sort_values('ratings', ascending=False, inplace=True)
pdf.set_index('movies', inplace=True)
pdf.head(10)
```

Figure 5: Code snippet for Model-Collaborative Filtering

```
In [18]: user_id = 7020
         pred_user_rating(user_id)
```

Out[18]:

movies	ratings
Bleak House (2005)	4.850939
Prime Suspect (1991)	4.795222
Bill Hicks: Revelations (1993)	4.789093
Children of Heaven, The (Bacheha-Ye Aseman) (1997)	4.723910
Heimat - A Chronicle of Germany (Heimat - Eine deutsche Chronik) (1984)	4.718742
Song of the Little Road (Pather Panchali) (1955)	4.693297
Betrayal (1983)	4.684712
Wings of Desire (Himmel über Berlin, Der) (1987)	4.683682
Witness for the Prosecution (1957)	4.674544
Connections (1978)	4.674173

Figure 6: Results for Model Based Collaborative Filtering

In the above Figure 6, we have obtained the results for model based collaborative filtering for the user id= 7020

```
# check the accuracy using Root Mean Square Error
accuracy.rmse(predictions)

RMSE: 0.7728

0.7727642384605172
```

Figure 7: Results for calculating the RMSE value

In Figure 7, we have achieved a RMSE value= 0.7724 for model based collaborative filtering technique using SVD which is better compared to the memory based approaches.

VII. CONCLUSION

We have seen that there are various approaches by which we can recommend movies to users. We are able to recommend the movies based on content similarity by utilizing the metadata of the movies and also recommend based upon similarity between the users. Hybrid approaches can also be employed to predict results and help in overcoming the challenges of individual approaches. In future works, new approaches and techniques can be used to increase the accuracy and scalability of recommendations.

VIII. FUTURE SCOPE

In recommendation systems, incomplete data pose a major challenge in making the recommendations. Further research can be carried out to manage the problem of sparsity of data and increase the accuracy of the recommendations. In future works, new approaches and techniques can be used to increase the efficiency and scalability of recommendations and handle the dynamic nature of movie database.

REFERENCES

1. Basiliyos Tilahun Betru, Charles Awono Onana, Bernabe Batchakui,"Deep Learning Methods on Recommender System: A Survey of State-of-the-art", International Journal of Computer Applications (0975 – 8887)(2017).
2. Daniar Asanov," Algorithms and Methods in Recommender Systems", Berlin Institute of Technology Berlin, Germany.
3. Gediminas Adomavicius1 and Alexander Tuzhilin," Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", International Journal of Computer Applications (0975 – 8887) (2005).
4. J. Ben Schafer," The Application of Data-Mining to Recommender Systems", University of Northern Iowa (2007).
5. Greg Linden, Brent Smith, and Jeremy York, "Amazon Recommendations Item-to-Item Collaborative Filtering",IEEE Internet Computing Volume 7 Issue, January 2003 page 76-80,(2003).
6. G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
7. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, GroupLens: an open architecture for collaborative filtering of netnews, in: Proceeding of the ACM Conference on Computer Supported Cooperative Work, 1994, pp. 175–186.
8. Laurent Candillier, Frank Meyer, and Françoise Fessant. Designing Specific Weighted Similarity Measures to Improve Collaborative Filtering Systems. ICDM, pp. 242-255, 2008.
9. Similarity Measures used in Recommender Systems: A Study, International Journal of Engineering Technology Science and Research IJETSR www.ijetsr.com ISSN 2394 – 3386 Volume 4, Issue 6 June 2017[10] Dr. P.V.R.D. Prasada Rao, Vineetha B, Ch. Priyanka, V. Satish, A Comparative Study on Recommendation System Using Hybrid Approach, *International Journal of Mechanical Engineering and Technology* 9(1), 2018, pp. 75–82. http://www.iaeme.com/IJMET/issues.asp?JType=IJMET&VType=9&IType=1
10. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 17, NO. 6, JUNE 2005
11. A Survey Paper on Recommender Systems, https://arxiv.org/abs/1006.527

AUTHORS PROFILE

G. Naga Sujini, received B.Tech degree in Computer Science & Engineering from JNT University, Hyderabad, Andhra Pradesh, India and M.Tech Degree in CS from JNT University, Hyderabad, Telangana, India. She is Assistant Professor of Computer Science & Engineering Department of Mahatma Gandhi Institute of Technology Hyderabad affiliated to JNTU-H. She is interested in Data Mining, Machine Learning and Deep learning also. Published several papers in various International Journals/ Conferences. She would like to pursue further studies and research in computer science and in specific areas like Data Mining and Machine learning.

D. Gyana Deepika, currently in final year, pursuing Computer Science Engineering from Mahatma Gandhi Institute of Technology, Hyderabad affiliated to JNTU-H. She is interested in Machine Learning and Artificial Intelligence and has done a few projects in these areas. She is keen towards building solutions for problems which involves working with and analyzing data. She has interned at companies and has contributed to real time projects involving the use of various machine learning algorithms and statistical approaches such as time series analysis. She would like to pursue further studies and research in computer science and in specific areas like machine learning.