© July 2021| IJIRT | Volume 8 Issue 2 | ISSN: 2349-6002

Movie Recommendation System Using Machine Learning

Ayush Pandey[1], Ananya Sharan[2], Vibhanshu Mishra[3], Ms Richa Gupta[4], Ms Charu Tyagi[5]
[1,2,3,4,5]*Department of Electronics & Communication Engineering, RKGIT Ghaziabad*

***Abstract* - This research paper represents the techniques and approaches which are used in the movie recommendation system. As we are very well aware about the fact that extracting meaningful data from the homogenous amount of raw data is a challenging problem and recommendation systems helps us in this situation. Recommendation system plays a very important role in our day-to-day life as it provides suggestions based on some data sets to users for certain resources such as movies, books, songs, etc. Recommendation systems are very fruitful for various organizations as large amount of data is being collected from various customers and after extracting the data's it provides best suggestions. Movie recommendation systems is helping peoples who are fond of watching movies by providing suggestions for what movie to go with without going through the large set of movies data. To reduce the human efforts by providing suggestions of movies based on the user interest is our main moto. Recommendation system is based on three approaches: first one is Collaborative Filtering; second one is Content based and third one is hybrid-based Approach.**

***Index Terms* - System, Filtering, Approach, Memory based, Content based approach, Hybrid approach.**

I.INTRODUCTION

A recommendation system is a system which provides the suggestion of things such as books, movies, music and list of items to buy on shopping sites, etc. according to the taste of users. These systems have become progressively fashionable today and are used at enormous rate in various sectors like movies, music, books, videos, clothing, restaurants, food, places, and many more. we have a tendency to enforce this system exploitation cooperative filtering algorithms and Apache framework. Matching the performance and potential of user-based recommender system and item-based recommender system is our main intention. Although there are several approaches developed till now but still search is done because of its uses in several applications. It has been developed in various areas like music, movies, news, and shopping sites. Majority of organizations are implementing recommendation systems for fulfilling the demands of their client. LinkedIn, Amazon, and Netflix are one of them. Because of this, user does not have to go through the lots of searches manually. Amazon recommendation systems are suggesting the users that what they should buy based on their past shopping history. For example, if a client is shopping for books on Amazon, Then Amazon provides suggestions of books related to the taste of client. Similarly, Netflix does for his clients if clients watch a show on his account, then Netflix recommendation system sorts the shows that a client watches or according to the taste of client and provides suggestions. Recommendation systems are going to be generally classified into following categories—Content-based, cooperative, and Hybrid approach. Users past behavior and patterns of search are taken into account in Content Based Filtering Approach. While users previous experiences and ratings are taken into account and corelated with alternative user in Cooperative Filtering Approach. Each content-based-filtering approach and collaborative-based filtering approach together forms the Hybrid Approach which we are following in this project. However, both the approaches have their own limitations, so Hybrid Approach is taken into the account. Movie lens dataset is used in this project which consist of more than thousands of ratings from more than twenty different users.

II. INDENTATIONS AND EQUATIONS

(I) Recommendation System:
1.First approach is Content Based: In content-based approach single parameter is passed to the function that is movie searched by the user. f(movie)
2.Second approach is Collaborative: In collaborative approach two parameters are passed to the function that is movie and user rating. f (movies, user)

(II) Dataset Usage:

We have used MovieLens Dataset by GroupLens for this Project this knowledge set consists of:

- More than thousands of rating movies.
- Minimum of thirty movies had been rated by each users.
- We had designed the hybrid recommendation system which contains the demographic knowledge of the users that is (age, gender, occupation) and dataset of IMDB is also taken into the account.

(III) cooperative Filtering:

- In cooperative Filtering system knowledge of the different users is being maintained with respect to their ratings on different of things.
- It basically maintains the user knowledge, besides content- item-information.
- This approach is being used by the majority of existing recommenders' engines for e.g. Netflix, Amazon, Facebook, LinkedIn, etc.

(IV) Basic plan behind cooperative Filtering:

(V) Utility Matrix:

Different users have different approaches while sorting the things so the information collected from different users is represented as a utility matrix. The basic idea behind creating utility matrix is predicting the blanks so that these information can be taken into the account for better and best recommendations .

(VI) Similarity Measures:

In Similarity Measures PCS is taken into the account which stands for Pearson Correlation Similarity which uses the technique of rows and columns of the Utility Matrix.

Advantages of the PCS:

- It is easy to interpret.
- Provides higher results with respect to all different measures.
- The ratings are being Normalized.

Different Similarity Measures are geometer Distance, cos Similarity, etc.

PCS Measure:

Considering the two users x and y have rated, then the Pearson Correlation Similarity relation between the two users is given by:

$$pcs(x,y) = \frac{\sum_{i \in I}(r_{x,i}-\overline{r_x})(r_{y,i}-\overline{r_y})}{\sqrt{\sum_{i \in I}(r_{x,i}-\overline{r_x})^2}\sqrt{\sum_{i \in I}(r_{y,i}-\overline{r_y})^2}}$$

where $\overline{r_x}$ denotes the average rating given by user x to all items. To calculate $\overline{r_x}$ we only consider items that were rated by the user.

III. FIGURES AND TABLES

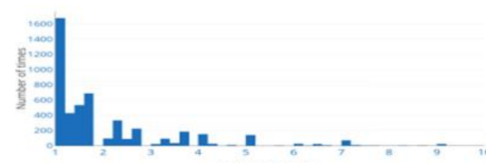

FIGURE 1. Frequency distribution of the dataset cardinalities. *X*-axis: most representative cardinality values from the *cardinality* range [1..160]. *Y*-axis: frequency of each cardinality (number of times this cardinality appears in the dataset).

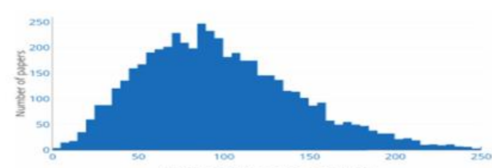

FIGURE 2. Frequency distribution of the papers with a fixed number of assigned topics. X-axis: number of assigned topics. Y-axis: frequency of papers. e.g.: there are approximately 250 papers containing from 85 to 90 topics each one of them.

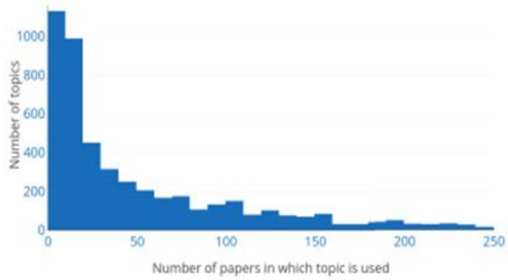

FIGURE 3. **Frequency distribution of the topics contained in the papers. *X*-axis: number of papers. *Y*-axis: frequency of topics. e.g.: there are approximately 200 topics that have been assigned to more than 50 and less than 60 papers each one of them, but only a very small number of topics have been popular enough to be assigned to 250 papers.**

TABLE 1. **Data mined artificial intelligence JCR journals. Years 2016, 2017 and the first half of 2018. Number of added papers to the database and impact factor of each journal.**

Journal	#papers	Impact factor
ACM Transactions on Intelligent Systems and Technology	157	3.196
Applied Soft Computing Journal	1608	3.541
Artificial Intelligence	202	4.797
Cognitive Computation	200	3.441
Data Mining and Knowledge Discovery	144	3.16
Decision Support Systems	286	3.222
Engineering Applications of Artificial Intelligence	523	2.894
Evolutionary Computation	53	3.826
Expert Systems with Applications	1623	3.928
IEEE Computational Intelligence Magazine	10	6.343
IEEE Transactions on Affective Computing	159	3.149
IEEE Transactions on Cybernetics	936	7.384
IEEE Transactions on Evolutionary Computation	131	10.629
IEEE Transactions on Fuzzy Systems	506	7.671
IEEE Transactions on Image Processing	1097	4.828
IEEE Transactions on Knowledge and Data Engineering	499	3.438
IEEE Transactions on Neural Networks and Learning Systems	727	6.108
IEEE Transactions on Pattern Analysis and Machine Intelligence	507	8.329
Information Fusion	161	5.667
Integrated Computer-Aided Engineering	54	5.264
International Journal of Computer Vision	174	8.222
International Journal of Intelligent Systems	34	2.929
International Journal of Neural Systems	95	6.333
Journal of Intelligent Manufacturing	400	3.035
Knowledge-Based Systems	726	4.529
Medical Image Analysis	209	4.188
Neural Networks	200	5.287
Neurocomputing	1860	3.317
Pattern Recognition	678	4.582
Semantic Web	44	2.889
Swarm and Evolutionary Computation	125	3.893
Swarm Intelligence	18	3.115
Total	14143	

TABLE 2. **Data mined information from each paper.**

Data mined information	Database field type
TITLE	varchar(240)
AUTHOR	varchar(400)
NUM_CITATIONS	smallint(5)
JOURNAL_ID	int(11)
YEAR	varchar(6)
PAGES	varchar(20)
DOI	varchar(35)
URL	varchar(200)
ABSTRACT	varchar(4000)
INDEX_KEYWORDS	varchar(1000)
AUTHOR_KEYWORDS	varchar(1000)
REFERENCES_FIELD	text
COUNTRIES	varchar(500)
ADDRESS	varchar(400)
PUBLISHER	varchar(100)
LANGUAGE_ARTICLE	varchar(50)
DOCUMENT_TYPE	varchar(25)
SOURCE	varchar(25)

TABLE 3. **Journals information.**

Data mined information	Database field type
JOURNAL_ID	int(11)
JOURNAL	varchar(250)
IMPACT_LEVEL	decimal(10,5)
POSITION_RANK	int(11)

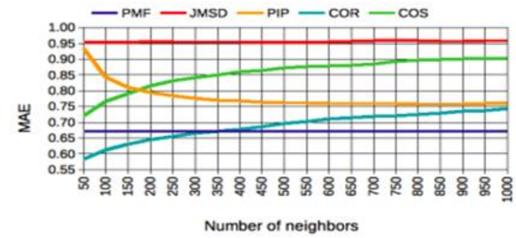

FIGURE 4. ***Mean Absolute Error* values obtained using the proposed *SD4AI* dataset. *X*-Axis: number of neighbors of each KNN run. *Y*-Axis: Prediction quality achieved; since we are facing an error measure, the best results are the lowest in the graph.**

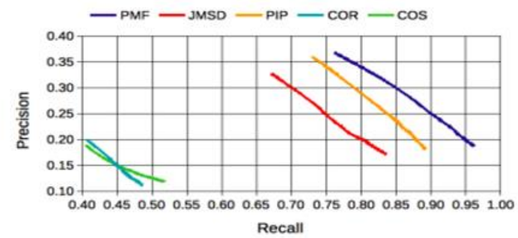

FIGURE 5. ***Precision* and *Recall* values obtained using the proposed *SD4AI* dataset. *X*-Axis: *Recall* results. *Y*-Axis: *Precision* results. Recommendation threshold: percentile 85 (rating 3.75). Best results are the highest in the graph: top-right corner.**

IV. CONCLUSION

Recommendation System is developed using various approaches and collects the ratings and review in data format and provides the suggestions for movies. If a user gives rating or search for a movie of a specific genre, then movie recommendation system is going to recommend him a list of movies according to his taste of movies. These systems are widely taking into the consideration in today's world for checking out important information.

Our intention behind this project was to build a recommendation engine suggesting the list of movies to the users according to users taste. We had worked on implementing Machine learning approaches and skills in a real-life project and how recommendation system is playing its role in our digital life. The present recommendation system somehow needs modifications for present and future requirements of higher recommendations.

REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, ''Recommender systems survey,'' Knowl.-Based Syst., vol. 46, pp. 109–132, 2013.

[2] F. M. Harper and J. A. Konstan, ''The MovieLens datasets: History and context,'' in Proc. ACM

Trans. Interact. Intell. Syst., vol. 5, Dec. 2015, pp. 19:1–19:19.

[3] J. Golbeck and J. Hendler, ‘‘FilmTrust: Movie recommendations using trust in Web-based social networks,’’ in Proc. 3rd IEEE Consum. Commun. Netw. Conf. (CCNC), vol. 1, Jan. 2006, pp. 282–286.

[4] J. Bennett and S. Lanning, ‘‘The Netflix prize,’’ in Proc. KDD Cup Workshop, New York, NY, USA, 2007, p. 35.

[5] “courser”, machine learning “Andrew”.