iJRASET
International Journal For Research in
Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 9 **Issue:** VI **Month of publication:** June 2021

**DOI:** https://doi.org/10.22214/ijraset.2021.35929

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

# Movie Recommendation System

Dinesh Shishodia[1], Shailesh Sharma[2], Rajat Chaudhary[3] and Shailesh Tomar[4]
[1,2,3,4]Department of Computer Science and Engineering, Raj Kumar Goel Institute of Technology, AKTU

***Abstract: This paper represents the overview of Approaches and techniques used in Movie Recommendation system. Recommendation system is used by many companies like Netflix, Amazon, Flipkart etc. It makes the user experience better and decrease the user efforts. It plays a very vital role in our day-to-day life. It is used in recommending Movies, Articles, News, Books, Music, Videos, People (Online Dating) etc. It learns from the user past behavior and based on that behavior it recommends item to the user. Likewise, in Movie Recommendation system movie is recommended to the user on the basis of movies watched, liked, rated by the user. In year 2020, approximate 10,000 movie were launched according to IDMB data. It saves a lot of times and efforts of the user by suggesting movies according to user taste and user don't have to select a movie from a large set of movies.***

***Index Terms: Machine Learning, Recommendation, Testing, Training***

## I. INTRODUCTION

A recommendation system filters out the unwanted items and show relevant item to the user. So how do recommender systems work? It works by understanding every user who interact with it. It stores the user past behavior information and train itself on that information. After training it knows what user would like and what user dislike.

One way to collect data is through explicit feedback. For instance, requesting clients to rate item on a scale from one to five stars or rating content with a like or dislike. You are asking user that you liked this content or not. The problem with explicit ratings is that user need to do some extra work and not every user would leave their feedback. So, this collected data will be too sparce and it will lead to low quality of recommendation. Another problem with explicit ratings is that everyone has different ways to rate an item, so the meaning of a four-star review might be different between two different people. Some people might just be more critical than others.

Another way to collect data is through your implicit behavior. Interpreting user behavior as an indication of interest or disinterest. For example, if you click on a link on a webpage, that can be considered as implicit positive rating for the content. Or if you click on an ad, it might show other ads similar to that ad. Click data is great because there is so much data that you do not have to deal with problems like data sparsity, but clicks aren't always a reliable indication of interest.

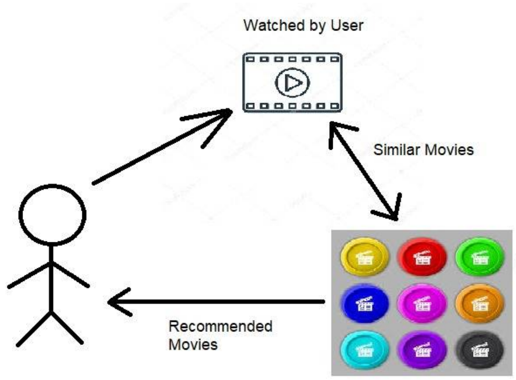
Using purchases as implicit positive ratings is also great because it's very resistant to fraud. If Someone trying to manipulate a recommender system based on purchase behaviour. Then it is going to very expensive because they have to buy a lot of stuff to affect recommender system results. That's why Amazon recommendations are so good because they have so much purchase data to work with. For example, YouTube can look at how many minutes you spend watching a video as an indication of how much you liked it. Watching videos doesn't require consumption of your money like purchase data does, but it does require your time, so it's also a pretty reliable indicator of interest compared to click data. That's why YouTube uses minutes watched heavily in its recommendation. When working on a recommender system, quality data is very important because even the best recommender system cannot produce good results without good data.

## II. OVERVIEW

Recommendation system is classified into three categories
Content based Filtering system
Collaborative Filtering system
Hybrid system

### *A. Content based Filtering System*

In content-based Filtering system, item similar to the user's past behaviour are recommended. For example, if a user liked Action-Adventure genre in the past then content-based Filtering system recommends movies from Action-Adventure genre. The problem with this approach is that it wouldn't recommend movie other than this genre and limits the user choices. That's why every item does not get the equal exposure.

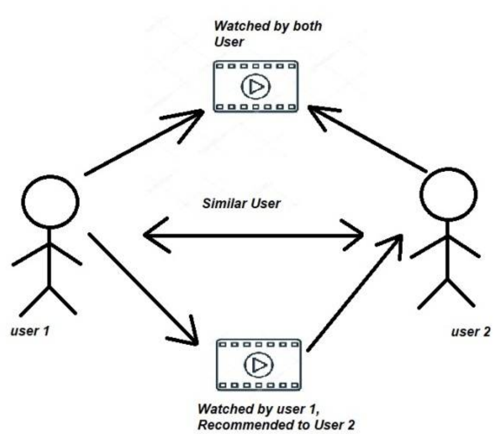



*B. Collaborative Filtering System*

In collaborative Filtering system, find similar user and based on their taste and choices. It recommends movie one liked and other haven't watched yet. For example, user A and B both liked a movie "Movie1", user A also like "Movie2" then we can recommend "Movie2" to the user B.

Collaborative Filtering System is classified into:

1) *User-based:* In user-based Collaborative Filtering, every user interacts with each item in the database and the selected user's taste and preferences are matched with other users, which are similar to the selected user basis of their taste and preferences.
2) *Item-based:* In Item-based Collaborative Filtering, it finds similar movie or item instead of finding similar user. Because movies or item are limited which save computation and storage but if userbase is large then it finding similar user and storing that data can be costly.




*C. Hybrid System*

Hybrid Recommendation system is made by combining two or more recommendation technique. It provides better performance with least demerits. We can combine Content-Based filtering system with Collaborative filtering system to get best from both of them.

## III. SIMILARITY MEASURES

*A. Pearson Correlation Coefficient*

It is used to find the linear correlation between two vectors. It gives value between -1 and +1. Where -1 represent the negative co-relation and +1 represents the positive co-relation.

$$Sim(u,v) = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_{u,I})(r_{v,i} - \bar{r}_{v,I})}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_{u,I})^2 + \sum_{i \in I}(r_{v,i} - \bar{r}_{v,I})^2}}$$

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

### *B. Constrained Pearson correlation*

Constrained Pearson correlation use median value instead of average of rating rated by both users.

$$sim(u,v)\frac{\sum_{i\in I}(r_{u,i}-r_{med})(r_{v,i}-r_{med})}{\sqrt{\sum_{i\in I}(r_{u,i}-r_{med})^2+\sum_{i\in I}(r_{v,i}-r_{med})^2}}$$

### *C. Cosine Similarity*

Cosine similarity finds how two vectors are related to each other using measuring cosine angle between these vectors.

$$\mathrm{Sim(u,v)}=\frac{\vec{R}_u\bullet\vec{R}_v}{|\vec{R}_u|\cdot|\vec{R}_v|}$$

### *D. Adjusted Cosine Similarity*

Cosine similarity measure does not consider the scenario in which different users use different rating scale. Adjusted cosine similarity solves it by subtracting the average rating provided by the user u.

$$Sim=\frac{\sum_{i\in I}(r_{u,i}-\bar{r}_u)(r_{v,i}-\bar{r}_v)}{\sqrt{\sum_{i\in I}(r_{u,i}-\bar{r}_u)^2+\sum_{i\in I}(r_{v,i}-\bar{r}_v)^2}}$$

### *E. Jaccard Similarity*

Jaccard similarity takes number of preferences common between two users into account. This does not consider the absolute ratings rather it considers number of items rated. Two users will be more similar, when two users have more common rated items. Jaccard produces limited number of values which makes the task of user distinction difficult.

$$sim(u,v)=\frac{|I_u\cap I_v|}{|I_u\cup I_v|}$$

### *F. Spearman Rank Correlation*

Spearman Rank Correlation uses ranks instead of ratings for calculating similarity

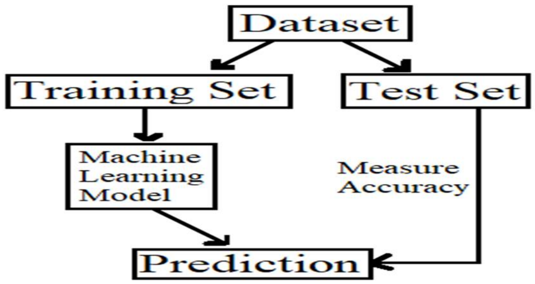
$$sim(u,v)=1-\frac{6\sum_{h_0}^{n_i}d_h^{\,2}}{n_i(n_i^{\,2}-1)}$$

## IV. TRAIN/TEST RECOMMENDER SYSTEM

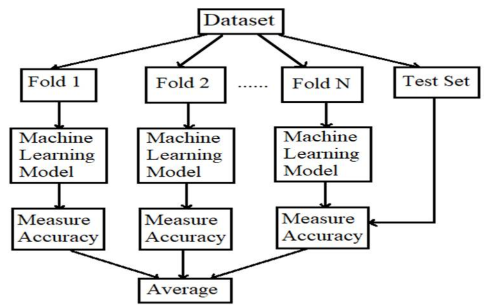
A big part of why recommender systems are as much art as they are science, is that it's difficult to measure how good they are. It's hard to say whether a user considers the recommendation to be good or not. Especially if you're developing your algorithms offline.

Recommender System is train on user past behavior, and then use it to make predictions about items, new users might like. So, on paper at least, you can evaluate a recommender system just like any other machine learning system. To measure recommender system's ability to predict item to the user. Start by splitting up your data into a training set and a testing set. Usually the training set is bigger, say 80% or 90% of all data and remaining for the testing set.

Once recommender system trained, it can make predictions about how a new user might rate some item they've never seen before. So, to measure how well it does, we take the data we reserved for testing. These are ratings that our recommender system has never seen before. So that keeps it from cheating. For example, one rating in test set user rated the movie five-star. We just ask the recommender system how it thinks this user would rate without telling it the answer. And then we can measure how close it came to the real rating.




It is possible to improve our recommender system on a single train/test split by using a technique called k-fold cross validation. It's the same idea as train/test but instead of a single training set, we create many randomly assigned training sets. Each individual training set, or fold, is used to train your recommender system independently, and then we measure the accuracy of the resulting systems against test set. We end up with a score of how accurately each fold ends up predicting user ratings, and average them together. It takes a lot more computing power, but the advantage of k-fold cross validation is that it prevents recommender system from over-fitting on a single training set.




## V. PERFORMANCE METRICS

### *A. Mean Absolute Error*

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

$y_i$ is predicted value
$x_i$ is actual value
n is the total number of observations
Example:

| Predicted Value | Actual Value | Error |
|---|---|---|
| 5 | 3 | 2 |
| 4 | 2 | 2 |
| 5 | 1 | 4 |
| 5 | 5 | 0 |

MAE = (2+2+4+0)/4
MAE = 2

### *B. Root Mean Square Error (RMSE)*

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - x_i)^2}{n}}$$

$y_i$ is predicted value
$x_i$ is actual value
n is the total number of observations

Example:

| Predicted Value | Actual Value | Error |
|---|---|---|
| 5 | 3 | 2 |
| 4 | 2 | 2 |
| 5 | 1 | 4 |
| 5 | 5 | 0 |

RMSE = sqrt $((2+2+4+0)^2/4)$
RMSE = sqrt (64/4)
RMSE = sqrt (16)
RMSE = 4

## VI. ALGORITHM ACCURACY

Surprise is a Python scikit for building and analysing recommender systems that deal with explicit rating data. According to the Surprise library, "here are the average RMSE, MAE of various algorithms (with their default parameters) on a 5-fold cross-validation procedure. The datasets are the Movielens 100k and 1M datasets. The folds are the same for all the algorithms."

| | **Movielens 100k** | **RMSE** | **MAE** |
|---|---|---|---|
| **0** | SVD | 0.934 | 0.737 |
| **1** | SVD++ | 0.920 | 0.722 |
| **2** | NMF | 0.963 | 0.758 |
| **3** | Slope One | 0.946 | 0.743 |
| **4** | k-NN | 0.980 | 0.774 |
| **5** | Centered k-NN | 0.951 | 0.749 |
| **6** | k-NN Baseline | 0.931 | 0.733 |
| **7** | Co-Clustering | 0.963 | 0.753 |
| **8** | Baseline | 0.944 | 0.748 |
| **9** | Random | 1.514 | 1.215 |

| | **Movielens 1M** | **RMSE** | **MAE** |
|---|---|---|---|
| **0** | SVD | 0.873 | 0.686 |
| **1** | SVD++ | 0.862 | 0.673 |
| **2** | NMF | 0.916 | 0.724 |
| **3** | Slope One | 0.907 | 0.715 |
| **4** | k-NN | 0.923 | 0.727 |
| **5** | Centered k-NN | 0.929 | 0.738 |
| **6** | k-NN Baseline | 0.895 | 0.706 |
| **7** | Co-Clustering | 0.915 | 0.717 |
| **8** | Baseline | 0.909 | 0.719 |
| **9** | Random | 1.504 | 1.206 |

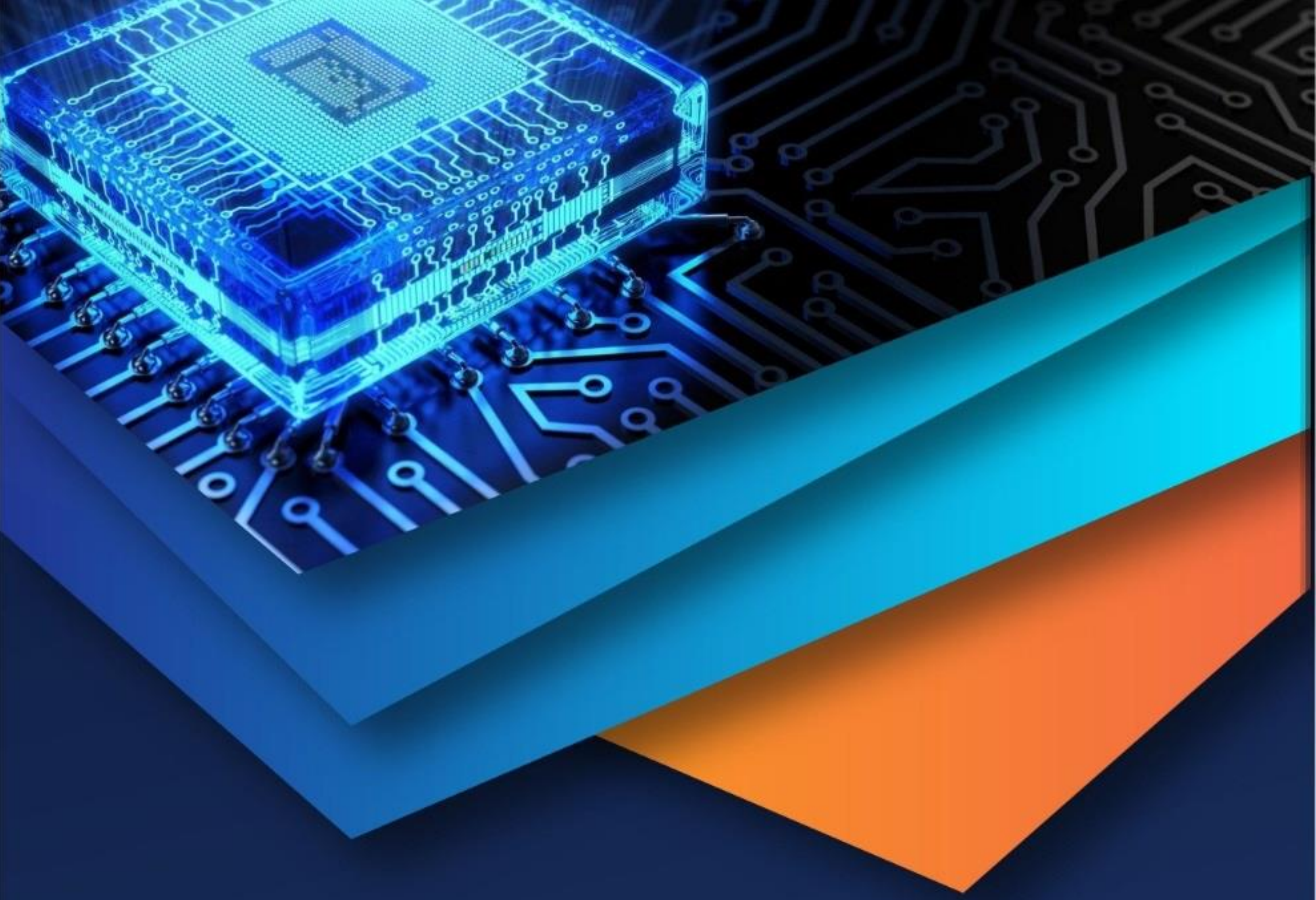
According to Surprise Library benchmark there is significant improvement in each algorithm, if recommender system is trained on a large dataset.

## VII. CONCLUSION

It's hard to say whether a recommender system is good or not just on the basis of performance metrics. The recommender system considered as good if it is able to satisfy real people not just the developer. But if you don't have good quality and good amount of data then it does not matter much, which algorithm you are using or how good your algorithm is.

## REFERENCES

[1] M. Chenna Keshava, S. Srinivasulu, P. Narendra Reddy, B. Dinesh Naik "Machine Learning Model for Movie Recommendation System" April 2020 International Journal of Engineering Research & Technology (IJERT).
[2] Manoj Kumar, D.K. Yadav, Ankur Singh, Vijay Kr. Gupta "A Movie Recommender System: MOVREC" International Journal of Computer Applications (0975 – 8887) Volume 124 – No.3, August 2015.
[3] F. Furtado∗ , A, Singh "Movie Recommendation System Using Machine Learning" Department of Master of Application, Jain University, Knowledge Campus, Bangalore, Karnataka, India.
[4] Ananya Agarwal, S. Srinivasan "Movie Recommendation System" International Research Journal of Engineering and Technology (IRJET) Volume: 07 Issue: 07 | July 2020.
[5] B. Sarwar, G. Karypis, 1. Konstan, 1. Riedl, "Item-based collaborative filtering recommendation algorithms", Proceedings of the 10th international conference on World Wide Web, pp. 285-295, 2001
[6] Bhumika Bhatt, Prof. Premal J Patel, Prof. Hetal Gaudani "A Review Paper on Machine Learning Based Recommendation System".
[7] Ashrita Kashyap, Sunita. B, Sneh Srivastava, Aishwarya. PH, Anup Jung Shah "A Movie Recommender System: MOVREC using Machine Learning Techniques" International Journal of Engineering Science and Computing volume 10 Issue No.6.
[8] Mr. Sridhar Dilip Sondur, Mr. Amit P Chigadani, Dr. Shantharam Nayak "Similarity Measures for Recommender Systems: A Comparative Study" Journal for Research | Volume 02| Issue 03 | May 2016 ISSN: 2395-7549.




10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429


# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY