MOVIE RECOMMENDATION SYSTEM USING COLLABORATIVE FILTERING

Rutuja Dinde[1], Saloni Zarkar[2], Yashasvini Varma[3]

[1-3]*Department of Information Technology, Padmabhushan Vasantdada Patil College of Engineering, Mumbai*

---***---

Abstract - *Advance technology and rising use of Internet in today's world stands on the fundamental rule i.e. user friendly, it has been achieving by trying to give higher user satisfaction Majority of applications consist of UI this leads to huge collection of information and sufficient resources to make datasets. Recommendation system helps in predicting personalized suggestions by using filtering methods on clustered data.*

Key Words: **collaborative Filtering, Cosine similarity, Pearson Correlation, Singular Value Decomposition, K- Nearest Neighbour, Graphical User Interface**

1. INTRODUCTION

Recommendation System tries to achieve the most appropriate suggestion on the user's interest from the product item. Recommendation system has boost sales on the digital platforms and is one of the most effective marketing strategy.

Dataset for recommendation system is formed by collecting information of user's ratings after they have watched the movie. In the digital era Recommendation System helps industries like Books, Music, movie etc. to give suitable suggestion to the users. In this paper we conduct a comparison study between the Similarity Models and Collaborative Filtering technique in order to find out which method gives the desired Movie recommendations, which are closest to the user's interest.

Two types of filtering methods in recommendation systems:-

1. Content-Based Filtering:

In content based filtering, recommendations are based on the specific content or its feature. It relies on the features and attributes of the item itself. For example if you want to recommend movie to the user then what you could do is look movies that they have liked in the past and find similar movies based on the director of the movie, the actor of the movie, its description or genre of the movie.

2. Collaborative Filtering:

Collaborative filtering approach relies on the information of the other users that have rated, watched or bought these items before. For example in a movie recommendation engine, we could use ratings provided by other users.

2. LITERATURE SURVEY

In paper [1], Due to economic growth in e-commerce business globally there is an increasing rise in use of recommendation system. The system provides recommend on products like movies, music, books and etc. to the uses of these business. For generating these recommendation one of the most favorable techniques is collaborative filtering. In the working of CF it's acquires data from a user which is target who shows similar taste of preferences and predicts the recommendation. For carrying on collaborative filtering task "Group lens" which is one of the representations, CF method based on algorithm named neighborhood. The word neighborhood is vast but here in this case. It can be defined as the user which as more similar and alike are grouped together. The study involves determining the neighborhood. As grouping of similar user is necessary for processing thus, clustering can take place as it involves task of finding groups of alike user. On comparing fuzzy clustering algorithms the result of experiment showed for all datasets has achieved higher recommendation accuracy. Thus, all these fuzzy clustering algorithms determines adequately then conventionally determined neighborhood.

In paper [2], an increasing popularity of internet and technology to obtain users valuable information in numerous data the recommended system plays a huge part. The recommendation system gives a personalized suggestions which is an advantage of this technology. The user have alike interest as these interest are very same in nature. We can co-ordinate the information on wide range with many user and filter recommendation on the targeted user. It works in a manner where user's information by the project and user of by the project's score a vector. User information by the project and all, there is a matrix of data is the project scone of user. The item collected includes all possible recommendation for targeted users. These combination of algorithm is the advantage of collaborative filtering algorithm and clustering algorithm.

In paper [3], On the basis of users preferences primary aim of the recommendation system is to give prediction on different items in which the user would be interested. With adequate data collaborative filtering are able to give accurate prediction. But as internet is expanding tremendously which lead to complexity and size of huge data in websites. Thus searching for suitable data becomes difficult and time consuming. Yet collaborative filtering technique is very pro because of the effectiveness. In this study we use item based technique of collaborative filtering for recommending items. These systems are able to upgrade the sites for every single user by adding a hyperlinks. In given study we work on movie recommendation system with the positive effects of item based collaborative filtering as we are using only ratings given by the user to the movie to make the system speedy and effective for a real time analysis. The recommender model is able to regenerate fresh and new recommendation as on when there are any updates done by the subscriber.

In paper [4], we have a comparison of performance between FCM clustering algorithm and subtractive clustering algorithm to model a set of non-lines system. The optimal modelling results are achieved when the validity indices are on their optimal values. In generally, the model generated by subtractive clustering are more accurate the FCM algorithm. Subtractive clustering does not need training algorithm whereas. It's needed in FCM. By using Sugeno training routine an important is achieved the system modelling error from resulting model FCM whereas turning the radii parameter for the subtractive clustering number and consequently the validity index value and finally modelling LSE.

In paper [5], they tries to explain the movie recommend system via two collaborative filtering algorithm. The tendency of a customer to look at the recommendation provided based on previous transaction or feedback is slightly high which leads to making these system fine tune for user's needs. Which helps the Provides as customer will use the application again. Which leads to revenue if customer uses the application frequently. Our primary uses aims to design a movie recommendation system which considers the past movies rating given by user. This system is implemented by using a collaborative filtering algorithm and apache mahout framework. Secondly, to compare two recommendation system based on performance and efficient. The two recommendation system can be approaches based on the infrastructure as the user based requires cache data storage and item based requires a dedicated processing server.

3. METHODOLOGY

Two major collaborative filtering techniques:

1. Memory Based:

In this approach we save all the results in the matrix and then use that matrix to predict it. So when we deal with memory based we have rating R that a user U would give to an item I.

Step1: Find similar users to that item which other users have not rated.

Step2: Calculate rating based on similar users. Hence, they deal with the similarity models.

3.1 SIMILARITY MODELS

a) Cosine Similarity:

Cosine similarity is widely used in recommendation system. In order to find out similarity between two points, then find out the angle between them. Cosine similarity is represented as Cos (0). 0 is the angle between two points. It ranges between -1 and 1. If cosine distance decreases then similarity increases and if cosine distance increases, similarity decreases.

For example, if both the points lie on the same plane then its angle is 0. Therefore, Cos (0) = 1. 1 represents that the two point are similar to each other.

Cosine similarity calculated as follows,

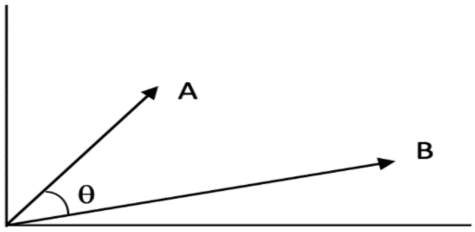


Fig -1: Angle between Two Vectors

Calculate the angle between A and B

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

Fig -2: Cosine Similarity Formula

b) Pearson Correlation:

Pearson correlation measures the powerful relationship between two variables. Pearson r lies between -1 and 1.

For example, your r could be 1 when you have a perfect positive linear relationship that means as x increases y increases with it in exactly the same way. You have R of negative 1 when the opposite is true when as x increases y decreases and the opposite as y decreases x increases.

Correlation is very important, how one movie is related to the other. So if it is very close to 1, the more that certain movie will be recommended. Pearson Correlation Formula as follow:

$$r = \frac{\sum (x - m_x)(y - m_y)}{\sqrt{\sum (x - m_x)^2 \sum (y - m_y)^2}}$$

Fig -3: Pearson Correlation Formula

2. Model Based:

This approach tries to fit the machine learning model into the data. In the recent years, recommendation algorithms have actually been invented a lot like clustering, deep learning, k- nearest neighbor, matrix factorization and so on.

3.2 ALGORITHMS

1. MATRIX FACTORIZATION:

Singular Value Decomposition

In real world scenario, the data is very huge. The number of users and items are in lakhs. It is difficult to deal with such huge data. In such cases we use the concept of dimensionality reduction where the date can be saved but somehow reduce the dimension. SVD is a matrix factorization technique to reduce the dataset by reducing the dimension. SVD works very well for sparse matrix.

Take a matrix and divide it into two X & Y.

X = m x p

Y = p x n

For example if we consider number 12. We can write it into 6 x 2 and 4 x 3. So m is 6 and n is 3

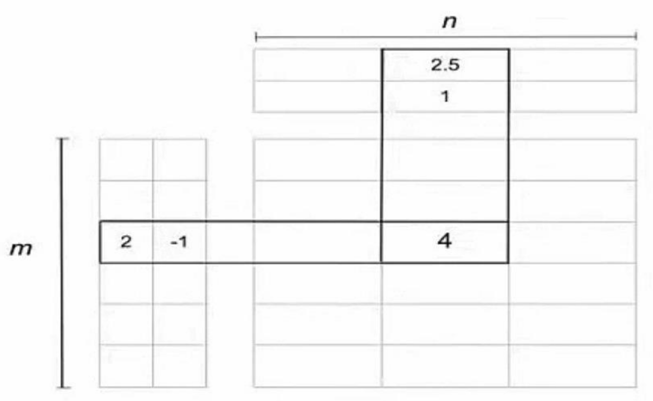


Fig -4: Matrix Factorization

On the rows we have users, on the columns we have items. So a rating of 4 corresponds to two vectors 2 and -1 and item vector 2.5 and 1.

So now let's say, that there is a horror movie where user has rated 2 and an action movie that the user has rated -1. So he/she does not like action movie at all. For a new movie which has 2.5 of the horror content and 1% of action content. For such movies, we want to know what the user will think of such movie. So we multiply the vectors here (2x2.5) + (-1x4) = 4.

Here we are trying to basically get the hidden features i.e. the latent features by breaking the matrix.

2. K-NEAREST NEIGHBORS ALGORITHM:

KNN means K nearest neighbor it is a very simple algorithm and give N training vector

KNN is a classification algorithm. Classification is to determine what groups to form, to predict future group normally in which group the data would cluster.

In order for KNN to work we have to predict future groups. i.e what group the future data is in, we normally refer it to as reference data that means we need some reference data for KNN algorithm

For the data record that needs to be classified it compute distance between data record and all of the reference data record then it looks at the K close data record is reference data.

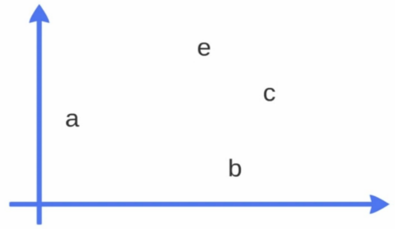
Suppose we have these "a" and "o" as training vectors in this bi-dimensional feature space, the KNN algorithm identifies the K nearest neighbor of "c". "C" is another feature vector that we went to estimate it is class in this case it identifies the nearest neighbor regardless of labels so suppose this element we have K equal to

K= 3

Classes "a" and "o"

Find class for "c"

When k= 1, each training vector defines a region in space, defining an voronoi partition of space



$$R_i = \{x : d(x, x_i) < d(x, x_j), i \neq j\}$$

Fig -5: Graph and Formula

We have a distance between each element x and x_i, that have to be smaller than the same distance for each other element. In this case it will define a voronoi partition of the space, and can be defined, for example this element 'c' and these elements 'b' and 'a' will define these regions, very specific regions.

The property of the KNN algorithm when K is equal to 1. We define them as region 1, 2, 3 and 4 based on the nearest neighbor rule.

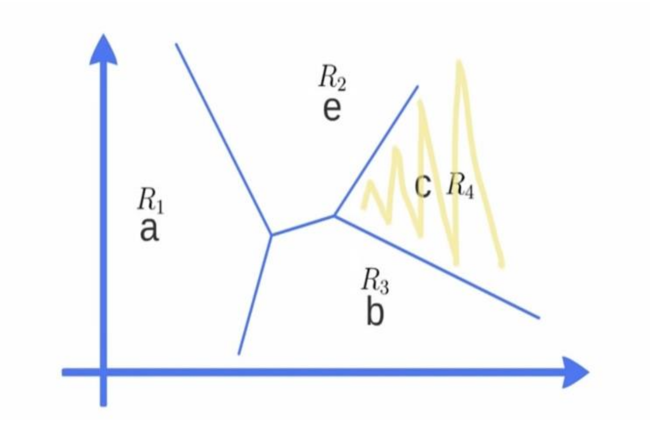


Fig -6: KNN Classification Graph

Each element that is inside this area will be classified as 'a' as well as each element inside this area will be classified as 'c' and the same for classes 'e' and 'b' as well.

4. CREATE A COMPLETE INTERFACE AND EXCEPTION HANDLING IN PYTHON

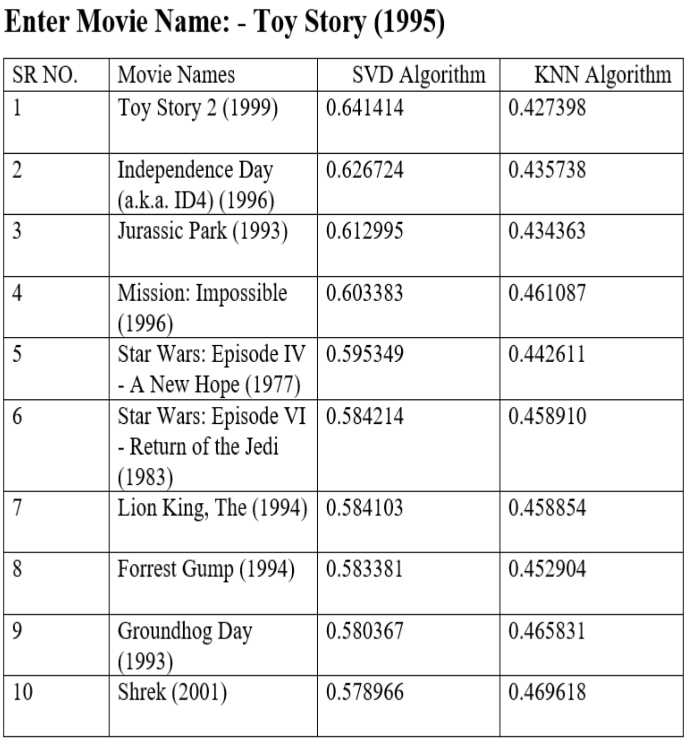
The graphical user interface (GUI) design focuses on expecting what users might need to know and assuring that elements that are easy to access, understand and use to facilitate those actions. It brings together different concepts from visual design, interaction design and its architecture. We have created a pattern where the interface gives you the recommendations of top 10 movies with respect to the movie which is entered by the user.

Now we have two options:

1. If the input movie name is not in the dataset then, the exception is shown on the screen.

2. If a correct movie name is entered and present in the dataset, then show the top 10 movie recommendations on the screen.

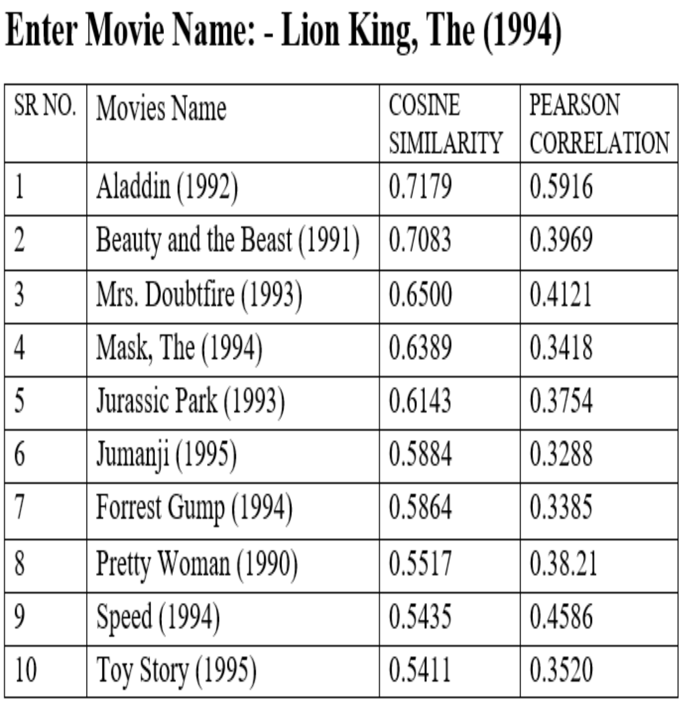
5. RESULTS



Enter Movie Name: - Toy Story (1995)

SR NO.	Movie Names	SVD Algorithm	KNN Algorithm
1	Toy Story 2 (1999)	0.641414	0.427398
2	Independence Day (a.k.a. ID4) (1996)	0.626724	0.435738
3	Jurassic Park (1993)	0.612995	0.434363
4	Mission: Impossible (1996)	0.603383	0.461087
5	Star Wars: Episode IV - A New Hope (1977)	0.595349	0.442611
6	Star Wars: Episode VI - Return of the Jedi (1983)	0.584214	0.458910
7	Lion King, The (1994)	0.584103	0.458854
8	Forrest Gump (1994)	0.583381	0.452904
9	Groundhog Day (1993)	0.580367	0.465831
10	Shrek (2001)	0.578966	0.469618

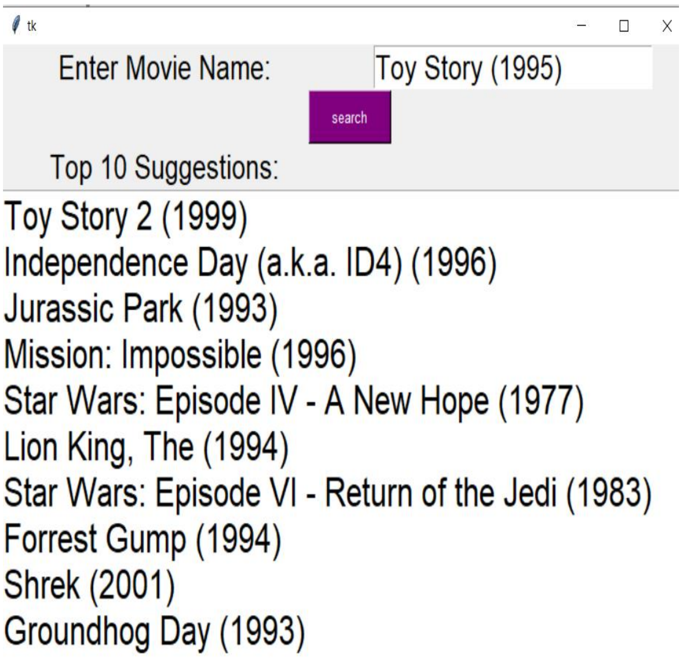
Fig -7: Comparison between Cosine Similarity and Pearson Correlation



Enter Movie Name: - Lion King, The (1994)

SR NO.	Movies Name	COSINE SIMILARITY	PEARSON CORRELATION
1	Aladdin (1992)	0.7179	0.5916
2	Beauty and the Beast (1991)	0.7083	0.3969
3	Mrs. Doubtfire (1993)	0.6500	0.4121
4	Mask, The (1994)	0.6389	0.3418
5	Jurassic Park (1993)	0.6143	0.3754
6	Jumanji (1995)	0.5884	0.3288
7	Forrest Gump (1994)	0.5864	0.3385
8	Pretty Woman (1990)	0.5517	0.38.21
9	Speed (1994)	0.5435	0.4586
10	Toy Story (1995)	0.5411	0.3520

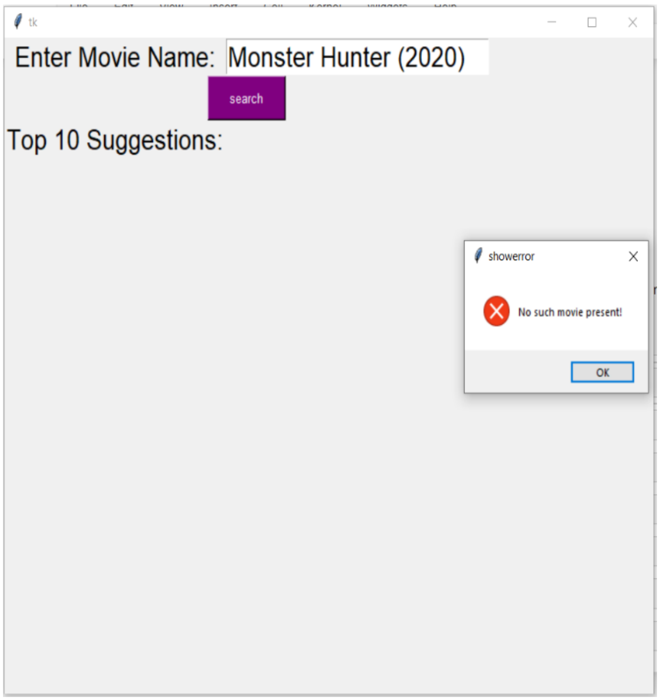
Fig -8: Comparison between SVD and KNN



Toy Story (1995)

Toy Story 2 (1999)
Independence Day (a.k.a. ID4) (1996)
Jurassic Park (1993)
Mission: Impossible (1996)
Star Wars: Episode IV - A New Hope (1977)
Lion King, The (1994)
Star Wars: Episode VI - Return of the Jedi (1983)
Forrest Gump (1994)
Shrek (2001)
Groundhog Day (1993)

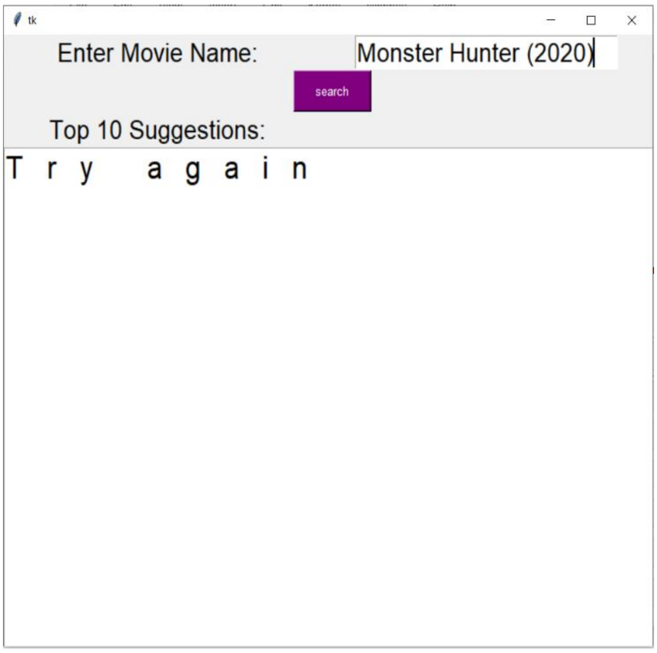
Fig -9: User Interface Design



Monster Hunter (2020)

No such movie present!

Fig -10: Show Error when movie is not present in the dataset



Monster Hunter (2020)

Try again

Fig -11: show message after error Message box

6. CONCLUSION

From the above results we conclude that the asset's returns have exactly the same variability, which is measured by Pearson correlation, but they are not exactly similar which is measured by cosine similarity. So we can think of Pearson correlation coefficient as the demeaned version of cosine similarity. Also in comparison with the algorithms SVD is helpful in reducing the matrix to a rank of the most relevance. In most of the cases, KNN is more likely to be used in data sets with low missing data proportion. SVD is better in huge data sets. It gives more accurate predictions as compared to KNN. When SVD is applied to pre-processed data, it gives better results by reducing the dimensions. Applying SVD helps to reduce the required calculation time for KNN.

REFERENCES

[1] Tadafumi Kondo and Yuchi Kanzawa, "performance comparison of collaborative filtering using fuzzy clustering for spherical data," Shibaura Institute of technology Tokyo, 2018

[2] Xingyuan Li" Collaborative filtering recommendation algorithm based on cluster," Ningbo Dahongying university, 2011

[3] Mukesh Kharita, Atul Kumar and Pardeep Singh" Item based collaborative filtering in movie recommendation in real time," National institute of technology India, 2018

[4] K.M.Bataineh, M.Naji and M.Saqer"A comparison study between various fuzzy clustering algorithms," Jordan university, 2011

[5] Ching-She Wu, Deepti Garg and Unnathi Bhandry" Movie recommendation system using collaborative filtering," San Jose state university USA 2018