ML based Movie Recommendation System

Peteti Harsha, Sidharth K, Mohammed Fazil K

Abstract— e In this fast-paced world, we all need entertainment tokeep our spirits up and our energy levels up. We regain our confidence in work as a result of entertainment, and we may work more enthusiastically as a result. We might rejuvenate ourselves by listening to music or watching movies of our choice. We may use movie recommendation systems to find goodmovies to watch online which are more dependable, because finding favored movies will take an increasing amount of time Itcan't be afforded to be squandered.to improve the quality of a movie in this document a hybrid approach combining content-based filtering and recommendation system Support Vector Machine as a classifier and a genetic algorithm are used in collaborative filtering the recommended technique hasbeen described, and comparison results have been displayed.

Index Terms—Collaborative filtering, Cosine similarity, Feature Vectors, Movie, OTT, Similarity Score, TF-ID vectorizer

1 Introduction

A recommendation system, sometimes known as a recommendation engine, is a model for information filtering that attempts to predict a user's interests and deliver recommendations based on those choices.Based on your tastes, we've come up with some suggestions. These systems have grown in popularity. Nowadays, they are extensively employed in fields such as cinema, music, and books. Videos, apparel, restaurants, food, locations, and other services are available.These systems gather data. information about a user's preferences and behavior, and then utilize that information to customize the user's experience. In the future, they should improve their suggestions.
Movies are an inextricable aspect of life. There are various types offilms, such as some for children. Some are animated movies for children's amusement, while others are for educational purposes. Some are aimed towards youngsters, while others are horror or actionpictures. Movies are simple to make. They are distinguished by their genres, such as comedy.

We Indians enjoy watching movies. We devote a significant amountof time and money to the amusement that movies give. The popularity of movies has only grown since the introduction of OTT platforms. The revenue produced by OTT platforms in 2017 was Rs.2019 crore The revenue is expected to increase to 6000 crores by2022. It is now up to the OTT owners to make this happen a better experience One method to achieve this is to recommend films to them. The goal of this project is to create an Artificial Intelligence-based movie recommendation system and build a functional website This website majorly has two features. Recommending movies to the users based on the movie they have searched. Classify Consumer reviews as positive and negative in the website for the movie they have searched

- *Peteti Harsha is currently pursuing bachelor degree program in electronics and communication engineering in Vellore Institute of Technology, India. E-mail: petetiharsha2002@gmail.com.*
- *Sidharth K is currently pursuing bachelor degree program in electronics and communication engineering in Vellore Institute of Technology, India E-mail: sidhusidharth1021@gmail.com*
- *Mohammed Fazil K is currently pursuing bachelor degree program in electronics and communication engineering in Vellore Institute of Technology, India E-mail: siddiqmohammed697@gmail.com*

Recommendation systems are being used by a growing number of businesses to boost user participation and enrich the buying experience. Customer happiness and income are two of the most crucial advantages of recommendation systems. The movie recommendation system is a very essential and strong mechanism. However, due tothe drawbacks of a purely collaborative approach, and movie recommendations Poor recommendation quality and scalability difficulties plague the system.

2 Literature Review

To handle such problems, we introduced a model combining both content-based and collaborative approach. It will give progressively explicit outcomes compared to different systems that are based on content-based approach. Content-based recommendation systems are constrained to people [1].There are filtering systems for information, known as the recommendation system or recommendation engine, considered here in paper, that help a person in identifying significant and possible services or products of interest based on the preferences given by him/her. This results in searching through lots of results to find the one that the user actually needs [2] the recommendation system has been built on the type of genres that the user might prefer to watch. The approach adopted to do so is content-based filtering using genre correlation. The dataset used for the system is Movie Lens dataset [3] This paper develops a Movie Recommendation System to recommend movies based on different parameters. The principal objective of the project is to construct a movie recommendation framework to prescribe pictures to users. There are many algorithmsthat help to build a recommendation system. Here, the Content-basedalgorithm has been employed to recommend movies based on the similarity with other films by analyzing the content of the movie [4].

This paper presents Filtering approaches including Content-based, which recommends items (movies) to the user (viewer) based on theirprevious history/ preferences and Collaborative-based which uses opinions and actions of other similar users (viewers) to recommend items (movies). In Collaborative filtering, User-based, Item based, SVD, and SVD++ algorithms have been implemented and the performance evaluated [5] research work has implemented a movie recommendation system by considering the latent factor. This system is coded in Java programming language. The implemented application can be commercially used since it helps the viewers to select the movies of their own choice [6] Recommender system (RS) are a type of suggestion to the information overload problem suffered by user of websites that allow the rating of particular item. The movie RS are one of the most efficient, useful, and widespread applications for individual to watch movie with minimum decision time. Many attempts made by the researchers to solve these issues like watching movie, purchasing book etc., through RS, whereas most of the study fails to address cold start problem, data sparsity and malicious attacks. [7] A recommendation system or recommendation engine is a model used for information filtering where it tries to predict the preferences of a user and provide suggests based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places and other utilities. These systems collect information about a user's preferences and behavior, and then use this information to improve their suggestions in the future.[8]

3 Proposed Model Description

We Indians enjoy watching movies. We devote a significant amount of time and money to the amusement that movies give. The popularity of movies has only grown since the introduction of OTT platforms. The revenue produced by OTT platforms in 2017 was Rs.2019 crore Therevenue is expected to increase to 6000 crores by 2022. It isnow up to the OTT owners to make this happen. A better experience One method to achieve this is to recommend films to them. The goal of this project is to create an Artificial Intelligence-based movie recommendation system and build a functional website for movie lovers to use. This website majorly has two features.

1) Recommending movies to the users based on the movie they have searched.

2) Classify Consumer reviews as positive and negative in the website for the movie they have searched.

1. Steps to be followed Collecting the data sets: Collecting all the required data set from Kaggle web site.in these projects we require movie. csv, ratings. csv, users. csv.

2. Data Analysis: make sure that that the collected data sets are correct and analyzing thedata in the csv files. i.e., checking whether all the column Felds are present in the data sets. 3.

3. Algorithms: in ourproject we have only algorithm cosine similarity is used to build the machine learning recommendation model.

4. Training and testing the model: once the implementation of algorithm is completed. we have to train the model to getthe result. We have tested it several times the model is recommend different set of movies to different users.

Based on similarity with other users, collaborative filtering systems analyze the user's behavior and preferences and forecast what they would like. User-based recommender and item-based recommender systems are the two types of collaborative filtering systems recommender.

1. User-based filtering: In the field, user-based preferences are fairly popular for creating custom-made systems This strategy is dependent on the user's preferences, likings. The process begins with people ranking movies (1-5) on a scale. We gather implicit information in these instances. A grading based on their actions As an example, It implies a positive preference when it occurs more than once. In terms of cinema systems, we can say that can imply that a user has some likeability if he or she sees the entire movie relating to it It's worth noting that there are no hard and fast standards for determining implicit evaluations. Next, we find a set number of nearest neighbors for each user first. We Pearson Correlation is used to calculatethe correlation between user ratings algorithm. If two users' evaluations are highly connected, the premise is that these two people must like similar things, and products are recommended to them. users with stuff
2. Item-based filtering: Unlike user-based filtering, item-based filtering focuses on the similarity between the item's users' likes rather than the item's popularity, users who are themselves The elements that are the most comparable are computed ahead of time. then for recommendation, the items that are most similar to the target item are recommended to the user.

Cosine Similarity Algorithm

Cosine similarity is a machine learning method that is used to create recommender systems. It's a method for identifying similarities between two documents. In this article, I'll give you an overview of Cosine Similarity in Machine Learning and how to usePython to construct it.

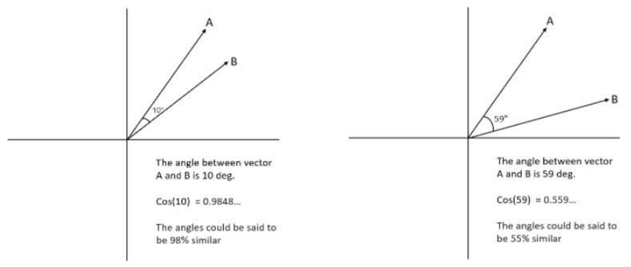
To detect commonalities between the two documents, cosine similarity is employed. It accomplishes this by calculating the similarity score between the vectors, which is accomplished by determining their angles. The number of similarities ranges from 0 to 1. When the similarity score between two vectors is 1, it suggests the two vectors are more similar. If the similarity score between two vectors is 0, however, there is no resemblance between the two vectors. When the similarity score is one, the angle between two vectors is 0 degrees, and when the similarity score is zero, the angle is 90 degrees. This technique is mostly usedin machine learning applications in recommendation systems to detect similarities between product descriptions so that we may offer the most similar product to the user for a better user experience.

Cosine Similarity is a metric that measures how similar two or more vectors are. The cosine of the angle between vectors is the cosine similarity. The vectors are usually non-zero and belong to an inner product space.

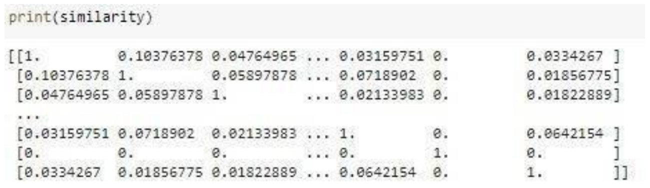
The divide between the dot product of vectors and the product of the Euclidean norms or magnitude of each vector is known as cosine similarity.

$$cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{j=1}^{k} A_j B_j}{\sqrt{\sum_{j=1}^{k} A_j^2}\sqrt{\sum_{j=1}^{k} B_j^2}}$$

Cosine similarity is a value enclosed in a limited range of 0s and 1s. Similarity is a measurement of the sine and cosine of the angle between two non-zero vectors A and B. Suppose the angle between the two vectors is 90 degrees. In this case, the cosine similarity value is 0. This means that the two vectors are orthogonal or perpendicular to each other. The closer the cosine similarity is to 1, the smaller the angle between the two vectors A and B. The image below shows this more clearly.

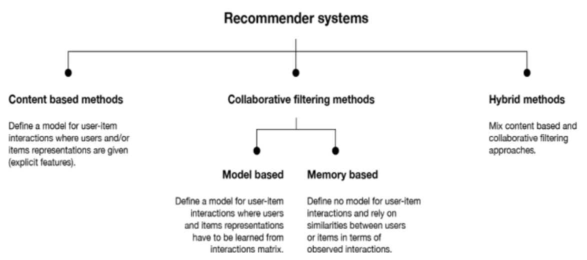

A
B
10°
The angle between vector A and B is 10 deg.
Cos(10) = 0.9848...
The angles could be said to be 98% similar
A
B
59°
The angle between vector A and B is 59 deg.
Cos(59) = 0.559...
The angles could be said to be 55% similar

Similarity scores It is a numerical value that runs from 0 to 1, used to assess how much two items resemble one another on a scale from 0 to 1. By comparing the text details of the two items' details, this similarity score is calculated. Therefore, the similarity score is a metric used to compare two things' given textual descriptions. Cosine similarity isone method for doing this**.**



```
print(similarity)

[[1.         0.10376378 0.04764965 ... 0.03159751 0.         0.0334267 ]
 [0.10376378 1.         0.05897878 ... 0.0718902  0.         0.01856775]
 [0.04764965 0.05897878 1.         ... 0.02133983 0.         0.01822889]
 ...
 [0.03159751 0.0718902  0.02133983 ... 1.         0.         0.0642154 ]
 [0.         0.         0.         ... 0.         1.         0.        ]
 [0.0334267  0.01856775 0.01822889 ... 0.0642154  0.         1.        ]]
```

Tf-Id vectorizer A word's TF-ID is determined by multiplying two distinct metrics for that word in a document: • The number of times a word appears in a document. The simplest method of determining this frequency is to simply count the number of times a word appears in a document. The length of a document or the frequency of the word that appears the most frequently in a document are other ways to modify frequency. • The word's overall inverse document frequency throughout a collection of documents. This refers to how prevalent or uncommon a word is throughout all documents. A word is more common the nearer it is to 0. To calculate this measure, multiply the overall number of documents by counting the instances of a term in documents and computing the logarithm. This number will therefore be close to zero if the word is widely used and appears in numerous papers. If not, it will go close to 1. The result of multiplying these two figures is the word's TF-IDF score in a document. The more relevant a word is in a given document, the higher the score.


Recommender systems
Content based methods
Define a model for user-item interactions where users and/or items representations are given (explicit features).
Collaborative filtering methods
Model based
Define a model for user-item interactions where users and items representations have to be learned from interactions matrix.
Memory based
Define no model for user-item interactions and rely on similarities between users or items in terms of observed interactions.
Hybrid methods
Mix content based and collaborative filtering approaches.

System Requirement Specification:

Hardware:
1) A PC with Windows/Linux OS
2) Processor with 1.7-2.4gHz speed
3) Minimum of 8gbRAM
4) 2gb Graphic card Software: Python libraries Google collab.

Python libraries:

Numpy:

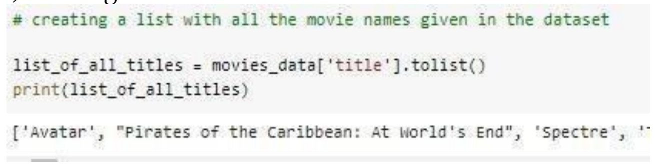
Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

Pandas:

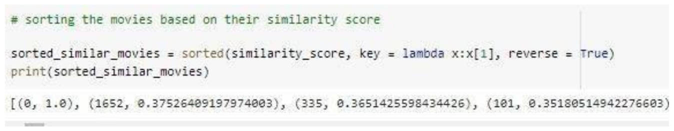
Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools.

STEPS

1) Importing the dependencies.
2) Data collection and Pre-Processing.
3) Selecting the relevant features for recommendation.
4) Replacing the null values with null string.
5) Combining text data with feature vectors.
6) Getting the similarity score with cosine similarity algorithm.
7) Getting the movie name from the user.
8) Creating a list with all movie names.
9) Finding the close match for the movie name.
10) Getting list of similar movies.
11) sorting the movies based on their movie name.
12) Printing the similar movie names.

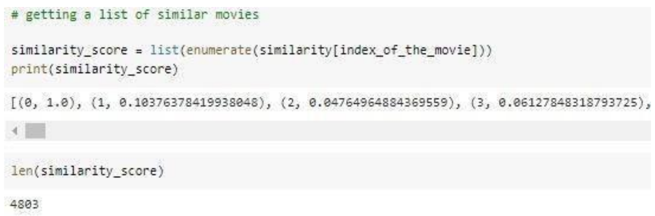


```
# creating a list with all the movie names given in the dataset

list_of_all_titles = movies_data['title'].tolist()
print(list_of_all_titles)

['Avatar', "Pirates of the Caribbean: At World's End", 'Spectre', '
```



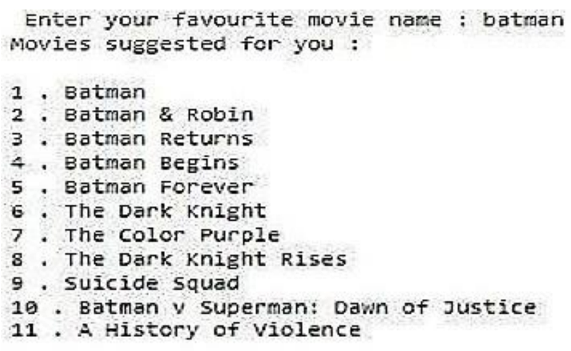
```
# sorting the movies based on their similarity score

sorted_similar_movies = sorted(similarity_score, key = lambda x:x[1], reverse = True)
print(sorted_similar_movies)

[(0, 1.0), (1652, 0.37526409197974003), (335, 0.3651425598434426), (101, 0.35180514942276603)
```

Similarity scores:



```
# getting a list of similar movies

similarity_score = list(enumerate(similarity[index_of_the_movie]))
print(similarity_score)

[(0, 1.0), (1, 0.10376378419938048), (2, 0.04764964884369559), (3, 0.06127848318793725),

len(similarity_score)

4803
```

Finding Similar movies:



```
 Enter your favourite movie name : batman
Movies suggested for you :

1 . Batman
2 . Batman & Robin
3 . Batman Returns
4 . Batman Begins
5 . Batman Forever
6 . The Dark Knight
7 . The Color Purple
8 . The Dark Knight Rises
9 . Suicide Squad
10 . Batman v Superman: Dawn of Justice
11 . A History of Violence
```

Collaborative Filtering:

1) Collaborative filtering doesn't recommend based on the features of the movie. The recommendation is based on the likes and dislikes or ratings of the neighbors or other users.

2) Also called User-User Filtering, it uses other users to recommend items to the input user. It attempts to find users that have similar preferences and opinions as the input and then recommends items that they have liked to the input. There are several methods of finding similar users and the one we will be using here is going to be based on the Pearson Correlation Function.

THE PROCESS FOR CREATING A USER BASED RECOMMENDATIONSYSTEM IS AS FOLLOWS:

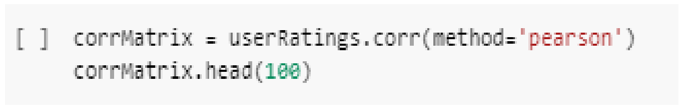
1) Select a user with the movies the user has watched
2) Add movie IDs to the movies watched by the user for easy recommendation.
3) Based on his rating to movies, find the top X neighbors.
4) Get the watched movie record of the user for each neighbor.
5) Calculate a similarity score using some formulaRecommend the items with the highest score.

Pearson Correlation:

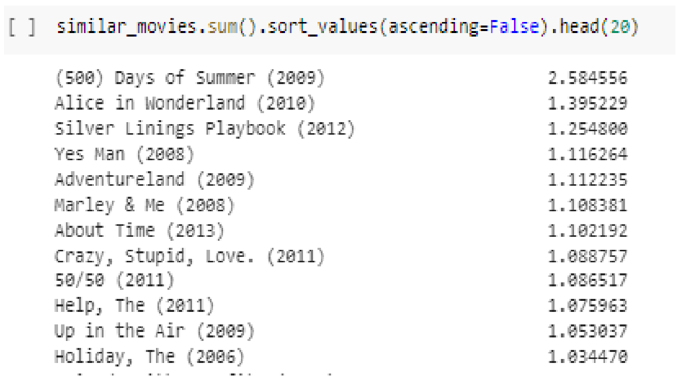
It is invariant to scaling. For example, if you have two vectors X and Y, then pearson (X,Y) == pearson (X, 2*Y+3). This is important because two users might rate two series of totally different in terms of absolute rates, but they would be similar users (i.e., with similar ideas) with similar rates in various scales. The value varies from r = -1 to r = 1, where 1 means that two users have similar tastes while a -1 means the opposite.

$$r = \Sigma(X - X')(Y - Y')/\sqrt{\Sigma(X - X')^2}\sqrt{(Y - Y')^2}$$

Where X' – mean of X variable
Y' – mean of Y variable



```
[ ]  corrMatrix = userRatings.corr(method='pearson')
     corrMatrix.head(100)
```

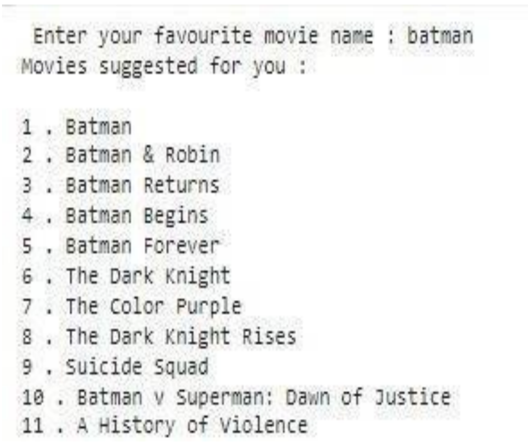


```
[ ]  similar_movies.sum().sort_values(ascending=False).head(20)

     (500) Days of Summer (2009)               2.584556
     Alice in Wonderland (2010)                1.395229
     Silver Linings Playbook (2012)            1.254800
     Yes Man (2008)                            1.116264
     Adventureland (2009)                      1.112235
     Marley & Me (2008)                        1.108381
     About Time (2013)                         1.102192
     Crazy, Stupid, Love. (2011)               1.088757
     50/50 (2011)                              1.086517
     Help, The (2011)                          1.075963
     Up in the Air (2009)                      1.053037
     Holiday, The (2006)                       1.034470
```

The purpose of this article was to provide an intuitive understanding and implementation of the basic methods of recommender systems (Collaborative filtering, content-based hybrid). Collaborative filtering is the process of predicting user interests by identifying preferences and information from many users, while content-based systems generate recommendations based on user preferences and profiles. Hybrid systems are often a combination of many recommender systems.

4 RESULTS

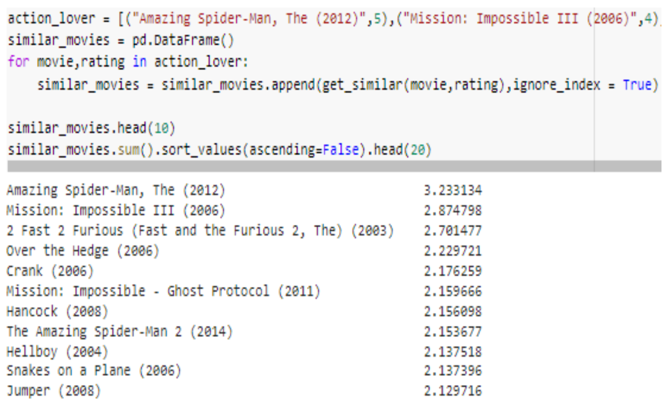
Content based recommending system:



```
 Enter your favourite movie name : batman
Movies suggested for you :

1 . Batman
2 . Batman & Robin
3 . Batman Returns
4 . Batman Begins
5 . Batman Forever
6 . The Dark Knight
7 . The Color Purple
8 . The Dark Knight Rises
9 . Suicide Squad
10 . Batman v Superman: Dawn of Justice
11 . A History of Violence
```

Collaborative recommending system

Our project is a movie recommendation system, so you can develop movies Recommender systems using either content-based or collaborative filtering Combine both. In our project we have developed a hybrid approach, a combination of both contents. Collaborative filtering. Both approaches have their strengths and weaknesses .. Only such types of content- based



```
action_lover = [("Amazing Spider-Man, The (2012)",5),("Mission: Impossible III (2006)",4)
similar_movies = pd.DataFrame()
for movie,rating in action_lover:
    similar_movies = similar_movies.append(get_similar(movie,rating),ignore_index = True)

similar_movies.head(10)
similar_movies.sum().sort_values(ascending=False).head(20)
```

```
Amazing Spider-Man, The (2012)                        3.233134
Mission: Impossible III (2006)                        2.874798
2 Fast 2 Furious (Fast and the Furious 2, The) (2003) 2.701477
Over the Hedge (2006)                                 2.229721
Crank (2006)                                          2.176259
Mission: Impossible - Ghost Protocol (2011)           2.159666
Hancock (2008)                                        2.156098
The Amazing Spider-Man 2 (2014)                       2.153677
Hellboy (2004)                                        2.137518
Snakes on a Plane (2006)                              2.137396
Jumper (2008)                                         2.129716
```

filtering based on user ratings and user preferences The film is recommended for users. Pros: Easy to design and reduce calculation time Disadvantages: The model can only provide recommendations based on existing recommendations User interests. In other words, the model can only be extended to a limited extent User's existing interests.

5 Conclusion

Watching movies is one of the most popular pastimes in modern society, and today people can watch movies anytime, anywhere, at work, at home, or in the car.

However, according to the normal supply and demand curve, 7,547 of the most popular English movies were released in 2019. You can use this movie recommendation system to save time and effort to find a good movie that suits our tastes.

There are no 100% curated recommendations or predictions, but we use machine learning algorithms to generate fairly accurate recommendations.

6 References

[1] IRJET- Movie Recommendation System using Machine Learning Algorithms April-2020

[2] Movie Recommendation System Using Semi-Supervised Learning.IEEE 18-20 Oct. 2019

[3] Analyzing the Behaviour of Java–Based Movie Recommendation System using Machine Learning IEEE-2019

[4] Hybrid Movie Recommendation System Using Machine Learning IEEE-2020

[5] Movie recommendation system using enhanced content-based filtering algorithm based on user demographic data IEEE-2021

[6] A. V. Dev, A. Mohan, "Recommendation system for bigdata applications based on set similarity of user preferences", 2016 International Conference on Next Generation Intelligent Systems (ICNGIS), pp. 1-6, 2016.

[7] Koen Verstrepen, Bart Goethals, "Unifying nearest neighbors collaborative filtering", Proceedings of the 8th ACM Conference on Recommender systems, October 0610,2014.

[8] Seroussi Y., "Utilising user texts to improve recommendations," User Modeling, Adaptation, and Personalization, pp. 403–406, 2010.

[9] Beel J., Langer S., and Genzmehr M., "Mind-Map based User Modelling and Research Paper Recommendations," in work in progress, 2014.

[10]A. Jain, S. K. Vishwakarma, "Collaborative Filtering for Movie Recommendation using RapidMiner", International Journal of Computer Applications, vol. 169,no. 6, pp. 0975- 8887, July 2017.