MOVIE RECOMMENDATION SYSTEM

A PROJECT REPORT

Submitted by:

KAARTHIKEYAN G M 191401012

in the partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND BUSINESS SYSTEM

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI-602105

ANNA UNIVERSITY: CHENNAI-600025

BONAFIDE CERTIFICATE

Certified that this project report **"MOVIE RECOMMENDATION SYSTEM"** is the bona fide work of **"KAARTHIKEYAN G M (191401012)"** who carried out the project work under my supervision.

SIGNATURE **SIGNATURE**

Dr. DEVAKI K, M.E., Ph.D.,	**Mr. BHUVANESWARAN, M.E.,**
HEAD OF DEPARTMENT	SUPERVISOR
PROFESSOR	ASSISTANT PROFESSOR (SG)
Department of Computer Science and Business System	Department of Computer Science and Engineering
Rajalakshmi Engineering College	Rajalakshmi Engineering College
Chennai – 602 105	Chennai - 602 105

Submitted to project Examination held on ____________________

INTERNALEXAMINER **EXTERNALEXAMINER**

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report.

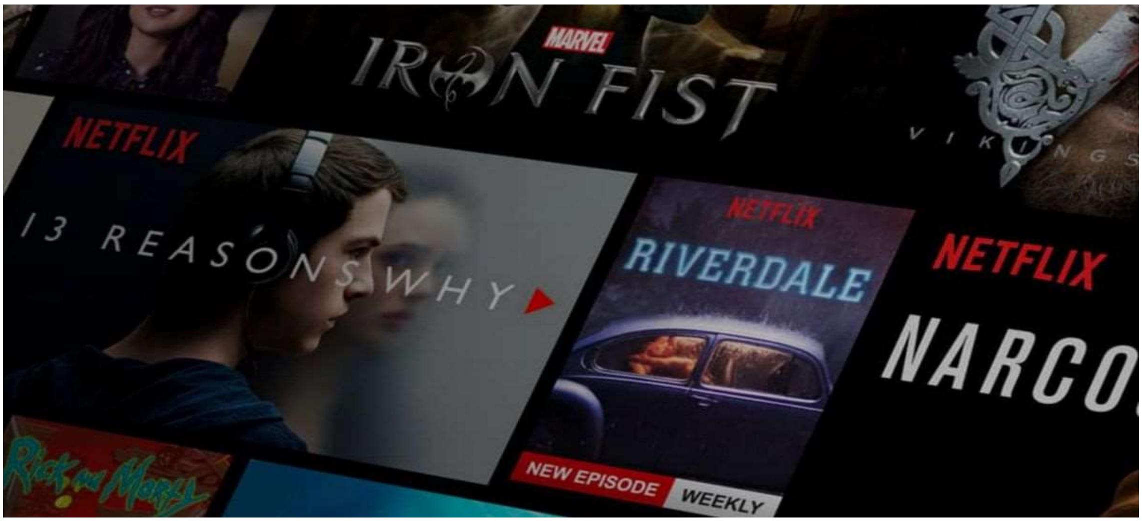
Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E., F.I.E.,** and our respected Chairperson **Dr. (Mrs) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time.

We express our sincere thanks to **Dr. DEVAKI K, M.E., Ph.D.,** Head of the Department of Computer Science and Business System for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Mr. BHUVANESWARAN, M.E.,** Assistant Professor (SG), Department of Computer Science and Engineering, Rajalakshmi Engineering College for her valuable guidance throughout the course of the project.

TABLE OF CONTENT:

1.INTRODUCTION:



A recommendation system is a type of information filtering system which challenges to assume the priorities of a user and make recommendations on the basis of user's priorities. Huge range of applications of recommendation systems are provided to the user. The popularity of recommendations systems has gradually increased and are recently implemented in almost all online platforms that people use.

The content of such system differs from films, podcasts, books and videos, to colleagues and stories on social media, to commodities on e-commerce websites, to people on commercial and dating websites. Often, these systems can retrieve and filter data about a user's preferences and can use this intel to advance their suggestions in the upcoming period.

For an instance, Twitter can analyse your collaboration with several stories on your wall to comprehend what types of stories please you. Many a times, these systems can be improvised on the basis of activities of a large number of people. For example, if Flipkart notices that many users who buy the modern laptop also buy a laptop bag. They can commend the laptop bag to a new customer who has just added a laptop to his cart. Due to the advances in recommender systems, users continuously expect good results. They have a low edge for services that are not able to make suitable recommendations.

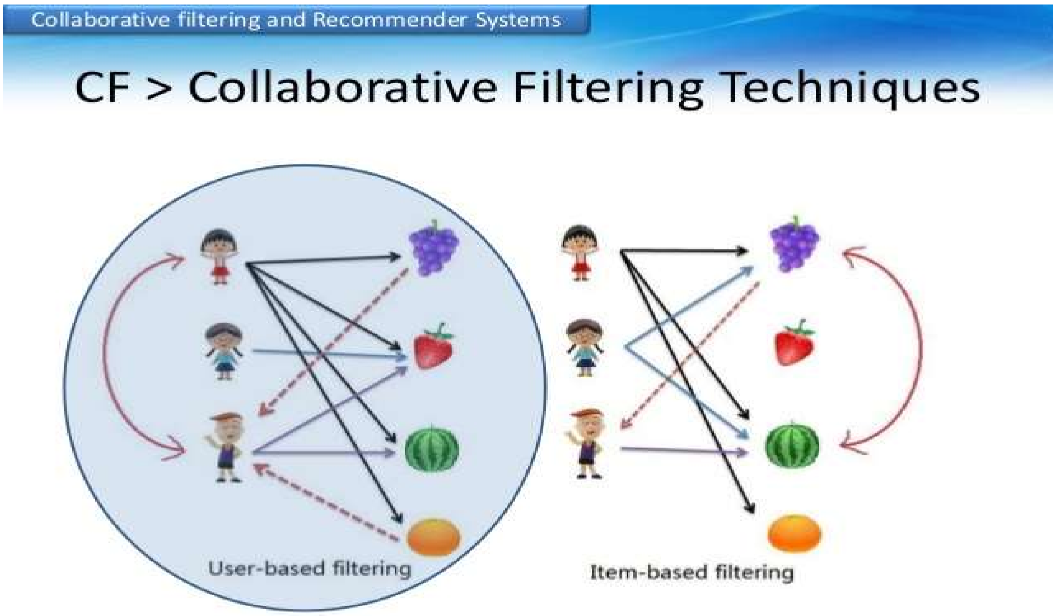
This has led to a high importance by technical corporations on refining their recommendation structures. However, the problem is more complicated than it appears. Every user has different likes and dislikes. In addition, even the taste of a single customer can differ depending on a large number of aspects, such as mood, season, or type of activity the user is performing. For an instance, the type of music one would prefer to listen during exercising varies critically from the type of music he would listen to while preparing dinner.

They must discover new areas to determine more about the customer, whilst still determining almost all of what is already known about of the customer. Two critically important methods are widely used for recommender systems. One is content-based filtering, where we attempt to shape the user's preferences using data retrieved, and suggest items based on that profile. The other is collaborative filtering, where in we try to cluster alike users together and use data about the group to make recommendations to the customer.

Recommender System is a system that seeks to predict or filter preferences according to the user's choices. Recommender systems are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general.

2.Recommender systems produce a list of recommendations in any of the two ways: -

2.1 Introduction to Collaborative Filtering:



The basic methodology of collaborative filtering systems is that these undetermined ratings can be credited since the noticed ratings are often highly linked across several users and items. For an instance, assume two users named Ram and Shamu, who have very comparable tastes. If the ratings, which both have stated, are very similar, then their resemblance can be determined by the fundamental algorithm. In such cases, there is a high probability that the ratings where in just one of them has definite value, are also likely to be similar. This similarity can be used to make interpretations about partly stated values.

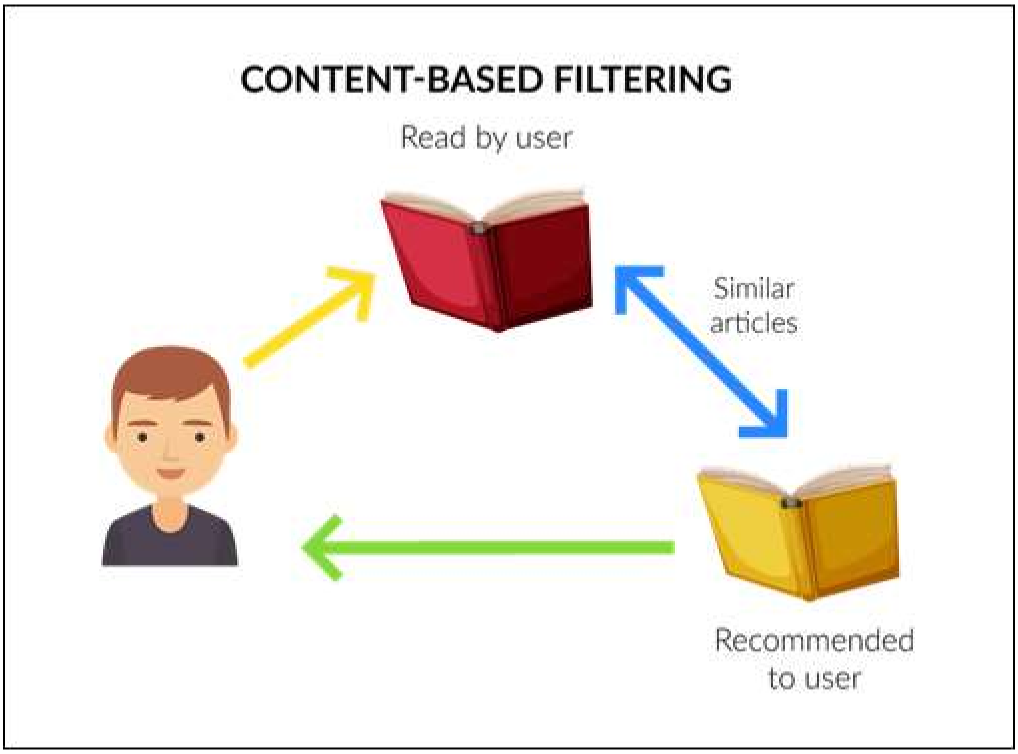
Almost all the projects for collaborative filtering emphasis on leveraging either item associations or user associations for the calculation procedure. Many of the models implement both kinds of correlations.

Additionally, some mock-ups use judiciously designed optimization procedures to generate a training model in much the similar way a classifier generates a training model from the mentioned or specified information.

This model is later used for assigning the absent values in the matrix, in the similar way that a classifier assigns the absent test tags.

There are two types of methods which are frequently implemented in collaborative filtering that are denoted to as memory-dependent procedures and model-dependent procedures.

2.2 Introduction to Content Based Filtering:

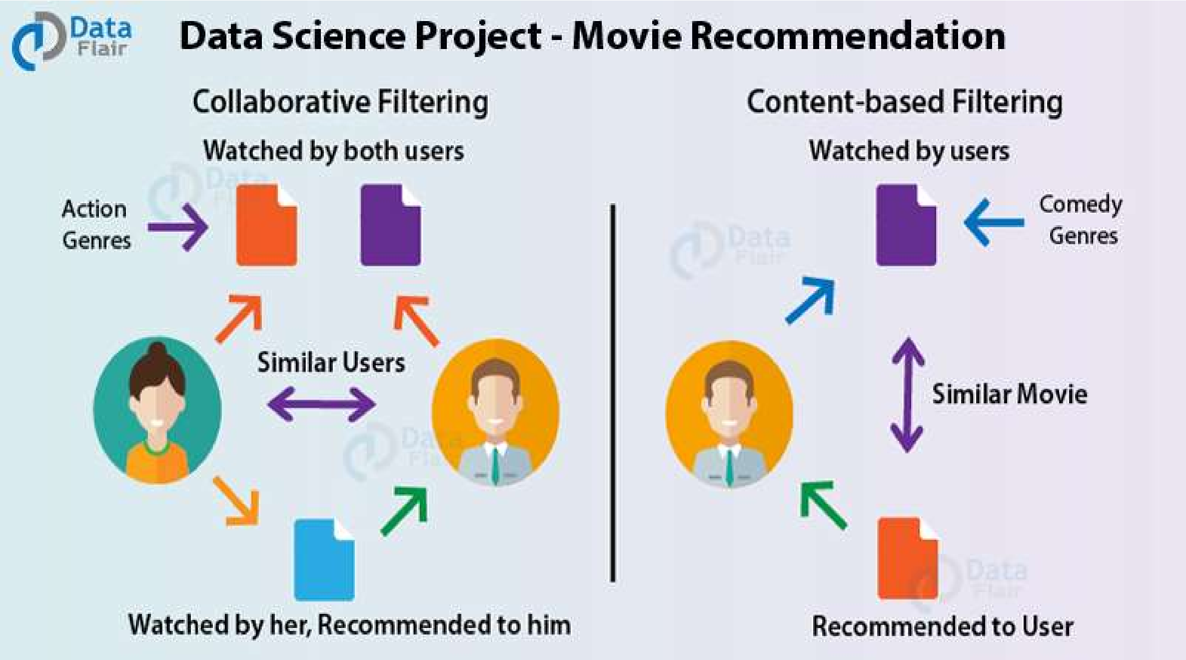


Content Based Recommendation procedure checks for the adores and aversions of the user and creates a User-based Profile. For producing a user profile, we check for the item profiles and their equivalent user rating.

The user profile is the combination of sum of the item profiles where combination being the ratings customer or user has evaluated.

After profile of the user has been generated, we estimate the resemblance of the user profile with all the items in the database, which is considered using cosine resemblance between the user generated profile and item profile.

Benefits of Content oriented procedure is that other user's information or data is not essential, and the recommender system can commend new commodities or anything which are not evaluated presently, nevertheless the recommender system will not recommend the items outside the type of items the user has given ratings of.



3.WORKING:

The movie recommendation system basically works by providing suggestions to the users by using the two renowned algorithms explained above. This movie recommendation system recommends movies to a user or client by evaluating IMDB ratings. The software and language which we have used for designing our interface and front end is Visual Basic Asp.Net. For creating database, we have used SQL Server since it is convenient.

This system collaborates with IMDB ratings and displays a list of movies which are highly rated by a user based on category of the movie.

This approach asks the user to provide 2 inputs –

1. Category of the movie (for e.g. comedy)

2. Year in which the movie is released (for e.g. 2016)

The algorithm segregates the list of movies from the dataset according to the inputs provided by user and finally displays the list of movies. The algorithm compares the inputs with the traits of the dataset and formulates the list. A user may select more than one category according to his fancies.

A bright feature of allowing the user to rate movies has enhanced the beauty of this recommender system. This is achieved by using collaborative filtering approach, wherein the system will provide recommendations to other likeminded users which have the same taste.

Example-User XYZ watches Movie "Bajirao Mastani" of category "Romance" (1 movie may fall in more than one category) and rates 8.5/10 (8.5 is considered to be a good rating). A like-minded person which has the same taste and is searching for the same category, then he may receive "Bajirao Mastani" as one of the suggestions.

The only problem in this prototype is that a user can't upload/view movies online on this website.

Let us develop a basic movie recommendation system using Python and Pandas.

Let us focus on providing a basic recommendation system by suggesting items that are most like a particular item, in this case, movies.

It just tells what movies/items are most similar to user's movie choice.

4.1 Importing Dataset:

import pandas as pd

column_names = ['user_id', 'item_id', 'rating', 'timestamp']

path = 'https://media.geeksforgeeks.org/wp-content/uploads/file.tsv'

df = pd. read_csv(path, sep='\t', names=column_names)

Check the head of the data

df.head()

	user_id	item_id	rating	timestamp
0	0	50	5	881250949
1	0	172	5	881250949
2	0	133	1	881250949
3	196	242	3	881250949
4	186	302	3	891717742

4.2 Check out all the movies and their respective IDs:

movie_titles = pd. read_csv('https://media.geeksforgeeks.org/wp-content/uploads/Movie_Id_Titles.csv')

movie_titles. head ()

	item_id	title
0	1	Toy Story (1995)
1	2	GoldenEye (1995)
2	3	Four Rooms (1995)
3	4	Get Shorty (1995)
4	5	Copycat (1995)

4.3 Merging the Data:

data = pd. merge (df, movie_titles, on='item_id')

data.head()

	user_id	item_id	rating	timestamp	title
0	0	50	5	881250949	Star Wars (1977)
1	290	50	5	880473582	Star Wars (1977)
2	79	50	4	891271545	Star Wars (1977)
3	2	50	5	888552084	Star Wars (1977)
4	8	50	5	879362124	Star Wars (1977)

4.4 Calculate mean rating of all movies:

data.groupby('title')['rating'].mean().sort_values(ascending=False).head()

```
title
Marlene Dietrich: Shadow and Light (1996)     5.0
Prefontaine (1997)                            5.0
Santa with Muscles (1996)                     5.0
Star Kid (1997)                               5.0
Someone Else's America (1995)                 5.0
Name: rating, dtype: float64
```

4.5 Calculate count rating of all movies:

data. groupby('title')['rating'].count().sort_values(ascending=False).head()

```
title
Star Wars (1977)                 584
Contact (1997)                   509
Fargo (1996)                     508
Return of the Jedi (1983)        507
Liar Liar (1997)                 485
Name: rating, dtype: int64
```

4.6 Creating data frame with 'rating' count values:

ratings = pd. Data Frame(data.groupby('title’) ['rating'].mean())

ratings ['num of ratings'] = pd.DataFrame(data.groupby('title')['rating'].count())

ratings. head()

Out[159]:

title	rating	num of ratings
'Til There Was You (1997)	2.333333	9
1-900 (1994)	2.600000	5
101 Dalmatians (1996)	2.908257	109
12 Angry Men (1957)	4.344000	125
187 (1997)	3.024390	41

Visualization imports

5.1 Data visualization:

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

Visualizations such as charts or graphs have been successfully used for hundreds of years in order to make sense of rather abstract data. Visualizations help us leverage the human's ability to see patterns and spot trends or outliers. In this way, the understanding of data can notably be supported.

Meanwhile, the **"visualization zoo"** accommodates a large variety of different visualization techniques. When trying to visualize a data set, however, only a very limited range of visualizations may fit the specific purpose. The selection of suitable visualizations is a crucial task in information visualization.

While the pool of suitable visualization techniques for each information block is limited to those fitting the identified data type, the selection of the most useful and usable technique is hard to measure and a quite subjective estimate.

5.2 Characteristics of graphical display:

Graphical displays are pictorial representations of statistical data that may be produced in a variety of ways (bar graphs, charts etc).

- show the data
- make large data sets coherent
- avoid distorting what the data has to say
- serve a reasonably clear purpose: description, exploration, tabulation or decoration

import matplotlib.pyplot as plt

import seaborn as sns

sns.set_style('white')

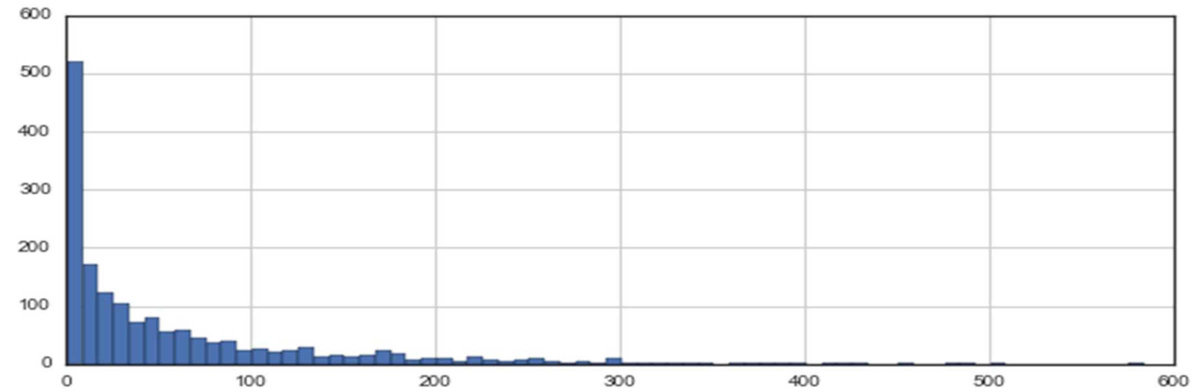
%matplotlib inline

plot graph of 'num of ratings column' :

plt.figure(figsize =(10, 4))

ratings['num of ratings'].hist(bins = 70)

<matplotlib.axes._subplots.AxesSubplot at 0x1258f8780>

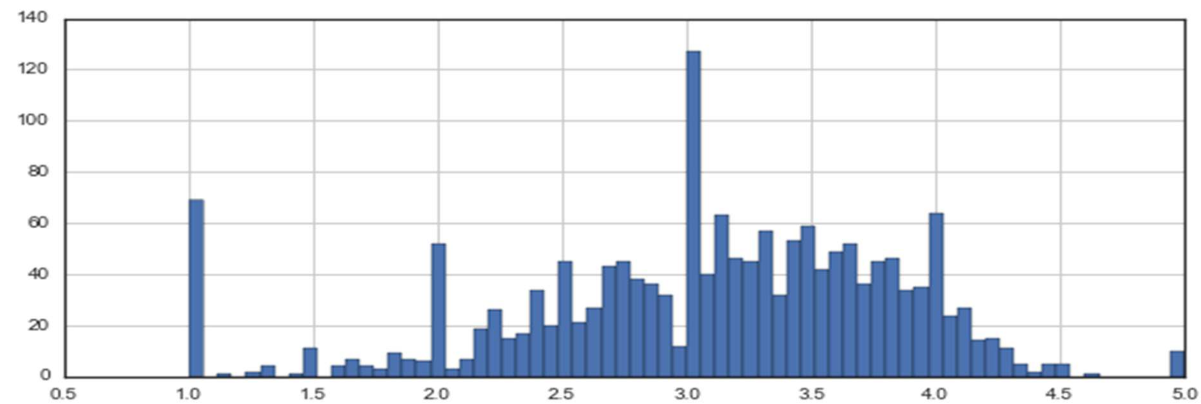


plot graph of 'ratings' column:

plt.figure(figsize =(10, 4))

ratings['rating'].hist(bins = 70)

<matplotlib.axes._subplots.AxesSubplot at 0x125d12908>



User's Rating

- The recommendations are calculated based on user ratings for that to get a complete data frame which consists of user rating for any given movie. attributes namely, user-id, movie-id, and user-ratings.

- These ratings are then defined into five class i.e., bad, ok, average, good, and excellent. Suppose a huge number of users have assigned the same ratings to movies X ... the title of the movie and then calculate the mean of the rating for each movie.

6.1 Sorting values according to the 'number of rating column' :

moviemat = data.pivot_table(index ='user_id',

columns ='title', values ='rating')

moviemat.head()

ratings.sort_values('num of ratings', ascending = False).head(10)

title	rating	num of ratings
Star Wars (1977)	4.359589	584
Contact (1997)	3.803536	509
Fargo (1996)	4.155512	508
Return of the Jedi (1983)	4.007890	507
Liar Liar (1997)	3.156701	485
English Patient, The (1996)	3.656965	481
Scream (1996)	3.441423	478
Toy Story (1995)	3.878319	452
Air Force One (1997)	3.631090	431
Independence Day (ID4) (1996)	3.438228	429

ANALYSATION:-

7.1 analysing correlation with similar movies

starwars_user_ratings = moviemat['Star Wars (1977)']

liarliar_user_ratings = moviemat['Liar Liar (1997)']

starwars_user_ratings.head()

```
user_id
0     5.0
1     5.0
2     5.0
3     NaN
4     5.0
Name: Star Wars (1977), dtype: float64
```

7.2 analysing correlation with similar movies:

similar_to_starwars = moviemat.corrwith(starwars_user_ratings)

similar_to_liarliar = moviemat.corrwith(liarliar_user_ratings)

corr_starwars = pd.DataFrame(similar_to_starwars, columns =['Correlation'])

corr_starwars.dropna(inplace = True)

corr_starwars.head()

title	Correlation
'Til There Was You (1997)	0.872872
1-900 (1994)	-0.645497
101 Dalmatians (1996)	0.211132
12 Angry Men (1957)	0.184289
187 (1997)	0.027398

7.3 Similar movies like starwars:

corr_starwars.sort_values('Correlation', ascending = False).head(10)

corr_starwars = corr_starwars.join(ratings['num of ratings'])

corr_starwars.head()

corr_starwars[corr_starwars['num of ratings']>100].sort_values('Correlation', ascending = False).head()

title	Correlation	num of ratings
Star Wars (1977)	1.000000	584
Empire Strikes Back, The (1980)	0.748353	368
Return of the Jedi (1983)	0.672556	507
Raiders of the Lost Ark (1981)	0.536117	420
Austin Powers: International Man of Mystery (1997)	0.377433	130

Now the same for the comedy Liar Liar:

7.4 Similar movies as of liarliar :

corr_liarliar = pd.DataFrame(similar_to_liarliar, columns =['Correlation'])

corr_liarliar.dropna(inplace = True)

corr_liarliar = corr_liarliar.join(ratings['num of ratings'])

corr_liarliar[corr_liarliar['num of ratings']>100].sort_values('Correlation', ascending = False).head()

title	Correlation	num of ratings
Liar Liar (1997)	1.000000	485
Batman Forever (1995)	0.516968	114
Mask, The (1994)	0.484650	129
Down Periscope (1996)	0.472681	101
Con Air (1997)	0.469828	137

8. CONCLUSION:

Movie recommendation systems which are existing have poor efficiency due to which movies are suggested in view of aspects for example - movie rated & evaluated by the User.

They have almost same viewing tastes, by means of data mining and insisting movies based on juncture of the three methods mentioned above that is - User Based Collaborative filtering, Content-based algorithm & data mining because of which the user will not only be recommended movies but this scheme also delivers the user with additionally advanced and sophisticated endorsements as movies which have a poor rating score in any of the Movie features produced based on data mining will be refined out during the significant allocation platform of the expected three way hybrid movie recommendation system.

9.REFERENCES :

[1] Microsoft SQL Server 2012 Management and Administration Second, Kindle Edition Book by Ross Mistry.

[2] ASP.NET: The Complete Reference Book by Matthew MacDonald (2002).