BMS COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

An Technical Seminar Report based on review of Research Publication/Patent

Recommender System

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Ankit Kesar

1BM18CS150

Work carried out at

Internal Guide

Sheetal V A

Assistant Professor

Department of Computer Science and Engineering

BMS College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
2021-2022

BMS COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I,Ankit Kesar (1BM18CS150) student of 7th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this technical seminar entitled "Recommender System" has been carried out under the guidance of **Sheetal VA,** Assistant Professor
, Department of CSE, BMS College of Engineering,Bangalore during the academic semester September - January 2022. I also declare that to the best of our knowledge and belief, the technical seminar report is not from part of any other report by any other students.

Signature of the Candidate

Ankit Kesar (1BM18CS150)

BMS COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Technical Seminar titled "**Recommender System**" has been carried out by Ankit Kesar (1BM18CS150)during the academic year 2021-2022.

Signature of the guide   Signature of external examinar

Sheetal VA
Assistant Professor,
Department of Computer Science and Engineering
BMS College of Engineering, Bangalore

Abstract

The purpose of a recommendation system basically is to search for content that would be interesting to an individual. Moreover, it involves a number of factors to create personalized lists of useful and interesting content specific to each user/individual. Recommendation systems are Artificial Intelligence and machine learning-based algorithms that skim through all possible options and create a customized list of items that are interesting and relevant to an individual. These results are based on their profile, search/browsing history, what other people with similar traits/demographics are watching, and how likely are to watch those movies. This is achieved through predictive modeling and heuristics with the data available.

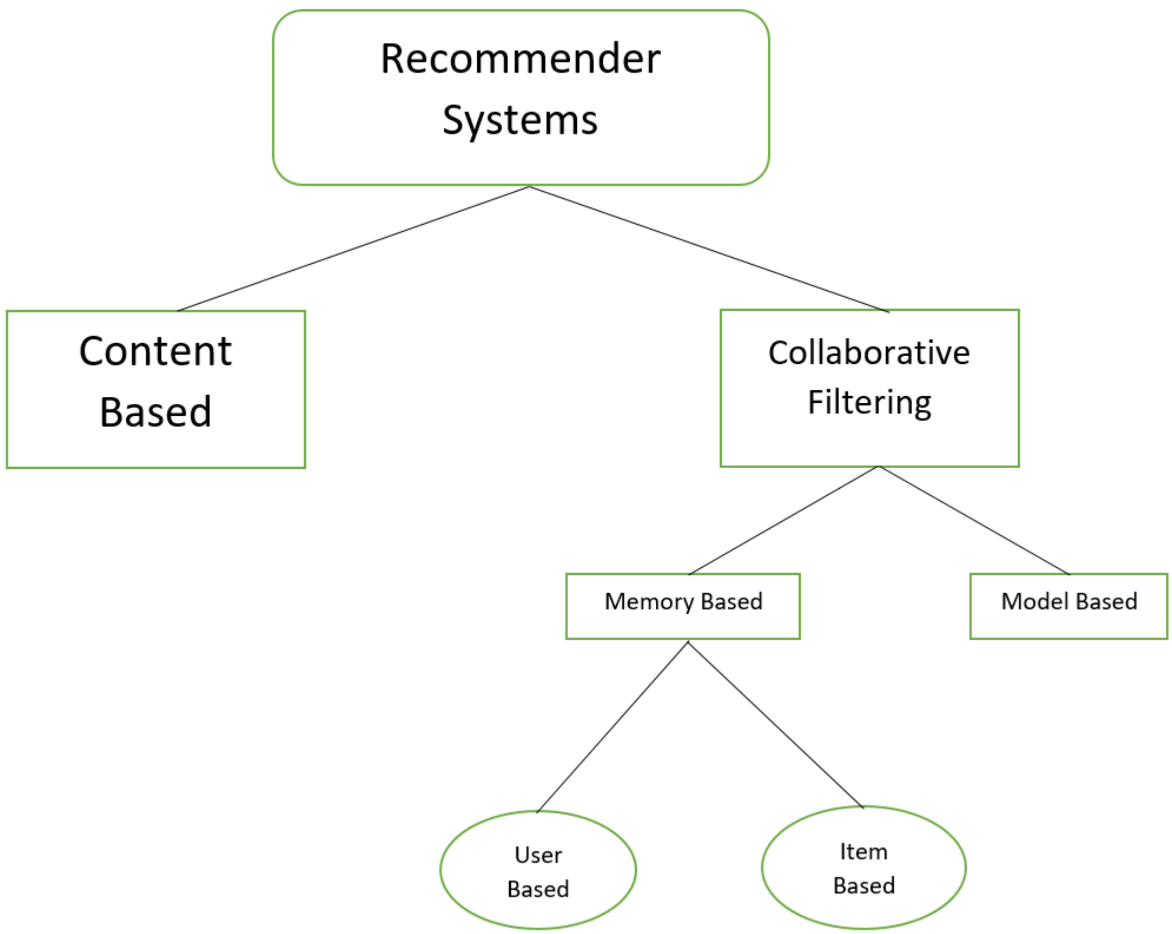
Since there are many techniques and algorithms used for recommender systems, we tried to find the algorithm with the least RMSE value

Chapter 1: Introduction

1.1 Overview

So with this project we propose to come up with way to recommend items, movies according to the user's interests. This becomes very important in this age. This system can be used alongside large datasets which has data like movie names, rating given by users.

We have implemented recommender systems with algorithms listed below:



1.2 Motivation

Recommendation systems are becoming increasingly important in today's extremely busy world. People are always short on time with the myriad tasks they need to accomplish in the limited 24 hours. Therefore, the recommendation systems are important as they help them make the right choices, without having to expend their cognitive resources.

The purpose of a recommendation system basically is to search for content that would be interesting to an individual. Moreover, it involves a number of factors to create personalized lists of useful and interesting content specific to each user/individual. Recommendation systems are Artificial Intelligence and machine learning-based algorithms that skim through all possible options and create a customized list of items that are interesting and relevant to an individual. These results are based on their profile, search/browsing history, what other people with similar traits/demographics are watching, and how likely are you to watch those movies. This is achieved through predictive modeling and heuristics with the data available.

Since there are many techniques and algorithms used for recommender system, we tried to find the algorithm with the least RMSE value.

1.3 Objective

Recommender systems are widely used in several different domains for the recommendation of articles, music, movies, and even people. Portals such as Amazon and Submarino use recommender systems to suggest products to their customers. Meanwhile, social networks such as LinkedIn and Facebook use them to suggest new contacts.
To accomplish that, the most used techniques employed in recommender systems are collaborative filtering and content-based systems. The collaborative filtering does not take into account the type of items, nor their attributes. It takes exclusively into account the expressed opinion about the other items in order to make recommendations. Meanwhile, content-based filtering uses the knowledge it has of the items and their attributes to make recommendations.

These techniques perform well, but they employ cluster[1] solutions to solve the scalability problem and be able to process high chunks of data.
This article looks for a new solution, different from the one normally employed. The objectives of this article are:

From an assessment database, similar to Grouplens' (GroupLens Research 2010) initiative, complemented with the relations among the participants, in such a way that is possible to draw the graph of the social network;
Evaluate the benefits to recommender systems originated from the knowledge provided by the social network. This database has missing values that are treated as non-evaluated items. Recommender systems try to predict the user's evaluation of an item that has not yet been evaluated. Based on the concept that people who are more closely connected have more influence on each other's choice

Chapter 2:

LITERATURE SURVEY

1) Content-Based Recommendation Systems

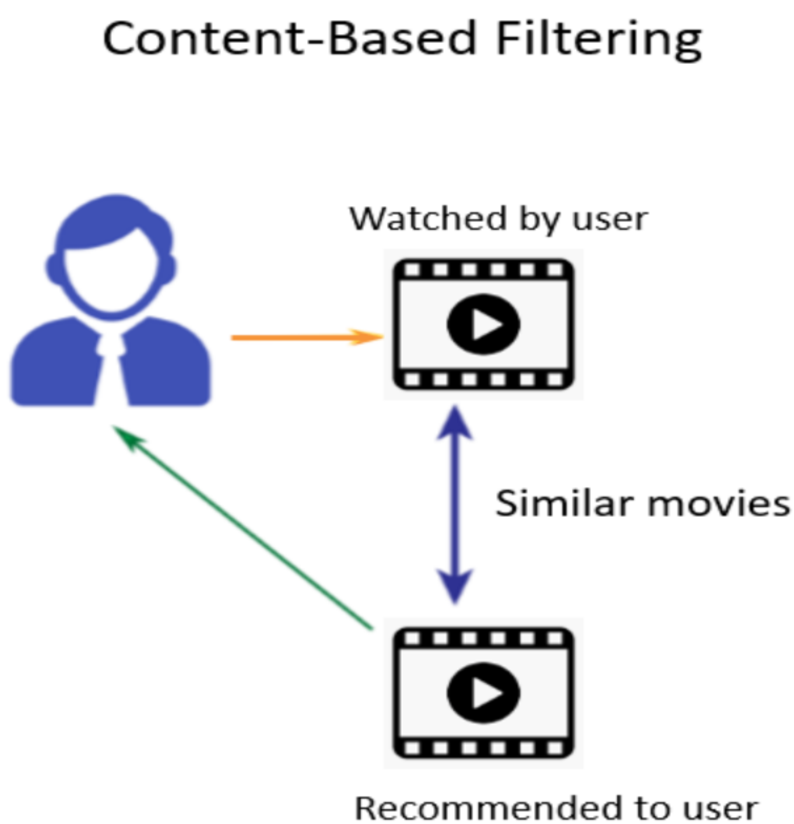
This chapter discusses content-based recommendation systems, i.e., systems that recommend an item to a user based upon a description of the item and a profile of the user's interests. Content-based recommendation systems may be used in a variety of domains ranging from recommending web pages, news articles, restaurants, television programs, and items for sale. Although the details of various systems differ, content-based recommendation systems share in common a means for describing the items that may be recommended, a means for creating a profile of the user that describes the types of items the user likes, and a means of comparing items to the user profile to determine what to recommend. The profile is often created and updated automatically in response to feedback on the desirability of items that have been presented to the user.

2) Item-Based Collaborative Filtering Recommendation Algorithms

Recommender systems are a powerful new technology for extracting additional value for a business from its user databases. These systems help users nd items they want to buy from a business. Recommender systems benet users by enabling them to and items they like. Conversely, they help the business by generating more sales. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web. New technologies are needed that can dramatically improve the scalability of recommender systems. In this paper we presented and experimentally evaluated a new algorithm for CF-based recommender systems. Our results show that item-based techniques hold the promise of allowing CF-based algorithms to scale to large data sets and at the same time produce high-quality recommendations.

Chapter 3: METHODOLOGY/TECHNIQUES OR ALGORITHM USED

Content Based Filtering:



Content-Based recommender system tries to guess the features or behavior of a user given the item's features, he/she reacts positively to.

	Adventure	Animation	Children	Comedy	Fantasy	Romance	Drama	Action	Crime	Thriller
0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
1	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	1.0
3	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
4	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0

Above Image is a boolean matrix based on userInput genre.

Now, given these genres, we can know which users like which genre, as a result, we can obtain features corresponding to that particular user, depending on how he/she reacts to movies of that genre.

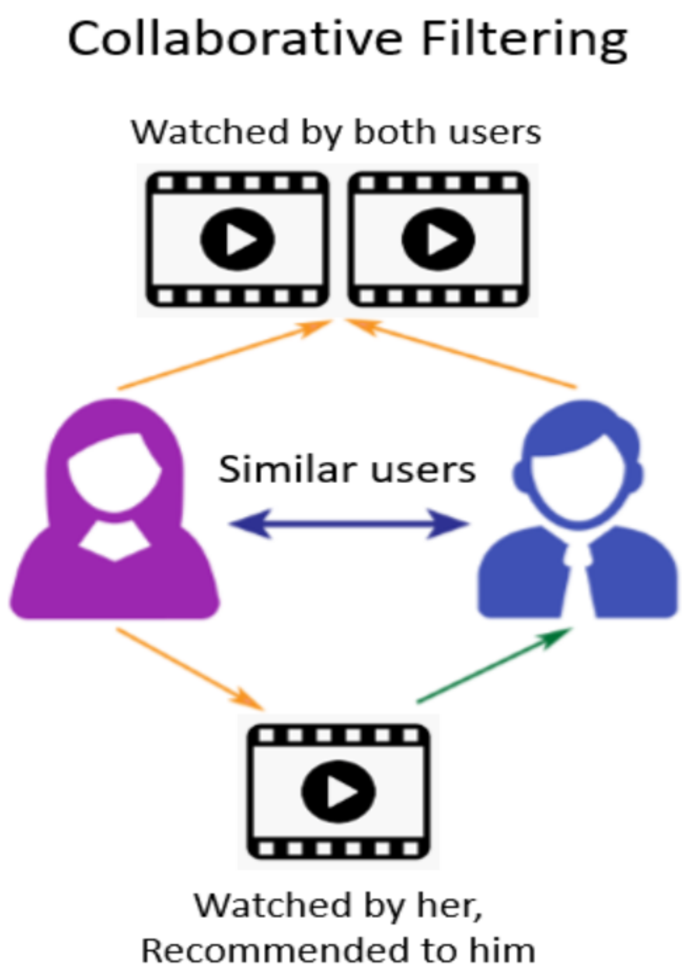
Once, we know the likings of the user we can embed him/her in an embedding space using the feature vector generated and recommend him/her according to his/her choice.

Collaborative Filtering

The technique we're going to take a look at is Collaborative filtering. It is based on the fact that relationships exist between products and people's interests. Many recommendation systems use collaborative filtering to find these relationships and to give an accurate recommendation of a product that the user might like or be interested in. As hinted by its alternate name, this technique uses other users to recommend items to the input user.

Memory Based

- **User Based Collaborative Filtering**



The process for creating a User Based recommendation system is as follows:

- Select a user with the movies the user has watched
- Based on his rating to movies, find the top X neighbours
- Get the watched movie record of the user for each neighbour.
- Calculate a similarity score using some formula
- Recommend the items with the highest score

	movieId	title	rating
0	1	Toy Story	3.894802
1	2	Jumanji	3.221086
2	3	Grumpier Old Men	3.180094
3	4	Waiting to Exhale	2.879727
4	5	Father of the Bride Part II	3.080811

Figure: User Input

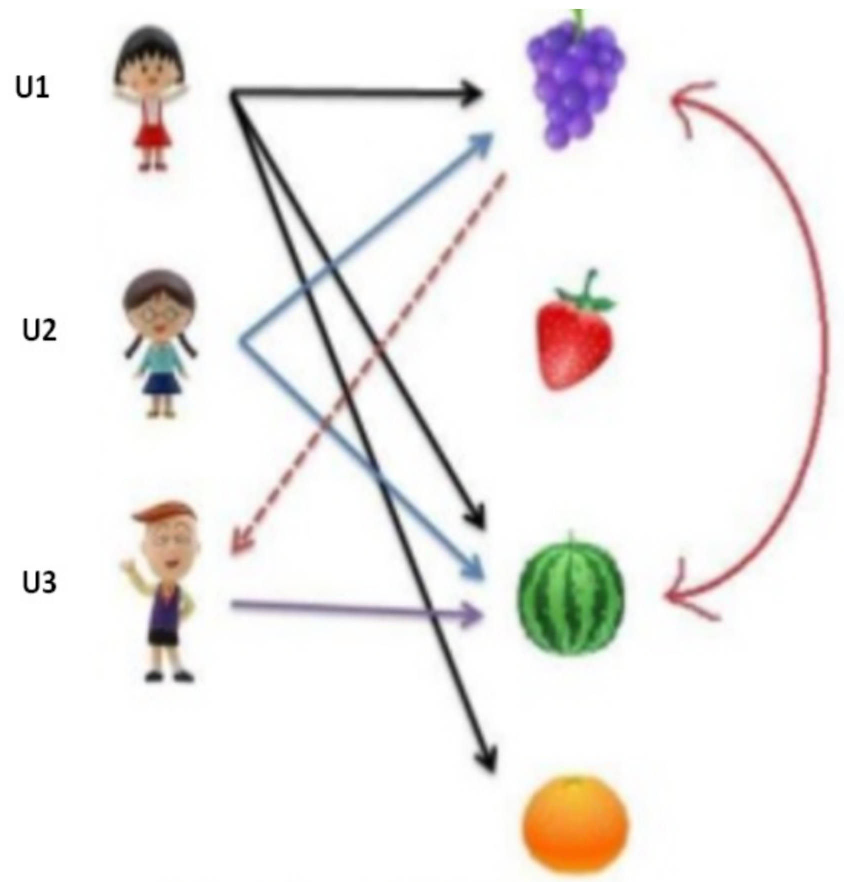
To find the similarity index between the user input and other movies in the dataset, we used the pearson correlation formula:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Pearson correlation is invariant to scaling, i.e. multiplying all elements by a nonzero constant or adding any constant to all elements. For example, if you have two vectors X and Y,then, pearson(X, Y) == pearson(X, 2 * Y + 3). This is a pretty important property in recommendation systems because for example two users might rate two series of items totally different in terms of absolute rates, but they would be similar users (i.e. with similar ideas) with similar rates in various scales. Based on the similarity index score, we recommend the movies which have high similarity score.

	weighted average recommendation score	movieId
movieId		
3814	5.0	3814
8580	5.0	8580
2295	5.0	2295
1999	5.0	1999
71268	5.0	71268
74282	5.0	74282
77846	5.0	77846
2771	5.0	2771
5915	5.0	5915
38304	5.0	38304

- Item Based Collaborative filtering



Item-item collaborative filtering, or item-based, or item-to-item, is a form of collaborative filtering for recommender systems based on the similarity between items calculated using people's ratings of those items.

The Only difference between Item Based and User Based is that we calculate the similarity index between Items in Item based and calculate similarity between users in User based.

The Procedure is similar to User based except that we calculate similarities between the Items here instead of Users.

title title	'burbs, The (1989)	(500) Days of Summer (2009)	10 Cloverfield Lane (2016)	10 Things I Hate About You (1999)	10,000 BC (2008)	101 Dalmatians (1996)	101 Dalmatians (One Hundred and One Dalmatians) (1961)
'burbs, The (1989)	1.000000	0.063117	-0.023768	0.143482	0.011998	0.087931	0.224052
(500) Days of Summer (2009)	0.063117	1.000000	0.142471	0.273989	0.193960	0.148903	0.142141
10 Cloverfield Lane (2016)	-0.023768	0.142471	1.000000	-0.005799	0.112396	0.006139	-0.016835
10 Things I Hate About You (1999)	0.143482	0.273989	-0.005799	1.000000	0.244670	0.223481	0.211473
10,000 BC (2008)	0.011998	0.193960	0.112396	0.244670	1.000000	0.234459	0.119132

Figure: Similarities between Items

Model Based

Remembering the matrix is not required here. From the matrix, we try to learn how a specific user or an item behaves. We compress the large interaction matrix using dimensional Reduction or using clustering algorithms. In this type, We fit machine learning models and try to predict how many ratings will a user give a product.

Methods:

Clustering Algorithms: They normally use simple clustering Algorithms like K-Nearest Neighbours to find the K closest neighbors or embeddings given a user or an item embedding based on the similarity metrics used.

Code Snippet:

```
sim_options = {'name' : 'msd'}

algo = KNNBasic(k=20, sim_options=sim_options )
cross_validate(algo=algo, data=data, measures=['RMSE'], cv=5, verbose=True)
```

Matrix Factorization based algorithms: The Singular Value Decomposition (SVD), a method from linear algebra that has been generally used as a dimensionality reduction technique in machine learning. SVD is a matrix factorisation technique, which reduces the number of features of a dataset by reducing the space dimension from N-dimension to K-dimension (where K<N).

Code Snippet:

```
algo = SVD()
cross_validate(algo=algo, data=data, measures=['RMSE'], cv=5, verbose=True)
```

Chapter 4: DESCRIPTION OF TOOL SELECTED

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

Chapter 5: DETAILED DESCRIPTION OF MODULES IMPLEMENTED and OUTPUT

During the last few decades, with the rise of YouTube, Amazon, Netflix and many other such web services, recommender systems have taken more and more place in our lives. From e-commerce (suggest to buyers articles that could interest them) to online advertisement (suggest to users the right contents, matching their preferences), recommender systems are today unavoidable in our daily online journeys.

In a very general way, recommender systems are algorithms aimed at suggesting relevant items to users (items being movies to watch, text to read, products to buy or anything else depending on industries).

Since there are various techniques and algorithms out there we try find out the best algorithm suitable for recommendation.

The project task or aim is to predict users interests and recommend product items that quite likely are interesting for them.

The task is not only to recommend movies but also to find the best recommender approach or technique using RMSE evaluation technique.

The input for our project is basically a csv file which is obtained from Movie Lens 100k Data Set. The dataset used for our algorithms has fields like User Id, Movie Id, Rating, Timestamp.

This data set consists of:

* 100,000 ratings (1-5) from 943 users on 1682 movies.

* Each user has rated at least 20 movies.

userId	movieId	rating	timestamp
1	169	2.5	1.2E+09
1	2471	3	1.2E+09
1	48516	5	1.2E+09
2	2571	3.5	1.44E+09
2	109487	4	1.44E+09
2	112552	5	1.44E+09
2	112556	4	1.44E+09
3	356	4	9.21E+08
3	2394	4	9.21E+08
3	2431	5	9.21E+08
3	2445	4	9.21E+08
4	16	4	1.04E+09
4	39	4	1.04E+09
4	45	4	1.04E+09
4	47	2	1.04E+09
4	94	5	1.04E+09
4	101	4	1.04E+09
4	246	4	1.04E+09
4	288	2	1.04E+09
4	296	4	1.04E+09
4	345	4	1.04E+09

ratings

Figure showing the ratings.csv which is used as input in our project

The other input used in our project is the dataset which has fields like Movieid, Genres, titles.

movieId	title	genres
1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1	Adventure|Children|Fantasy
3	Grumpier	Comedy|Romance
4	Waiting to	Comedy|Drama|Romance
5	Father of	Comedy
6	Heat (199	Action|Crime|Thriller
7	Sabrina (1	Comedy|Romance
8	Tom and	Adventure|Children
9	Sudden D	Action
10	GoldenEy	Action|Adventure|Thriller
11	American	Comedy|Drama|Romance
12	Dracula: D	Comedy|Horror
13	Balto (199	Adventure|Animation|Children
14	Nixon (19	Drama
15	Cutthroat	Action|Adventure|Romance
16	Casino (1	Crime|Drama
17	Sense and	Drama|Romance
18	Four Roon	Comedy
19	Ace Ventu	Comedy
20	Money Tr	Action|Comedy|Crime|Drama|Thriller
21	Get Shorty	Comedy|Crime|Thriller

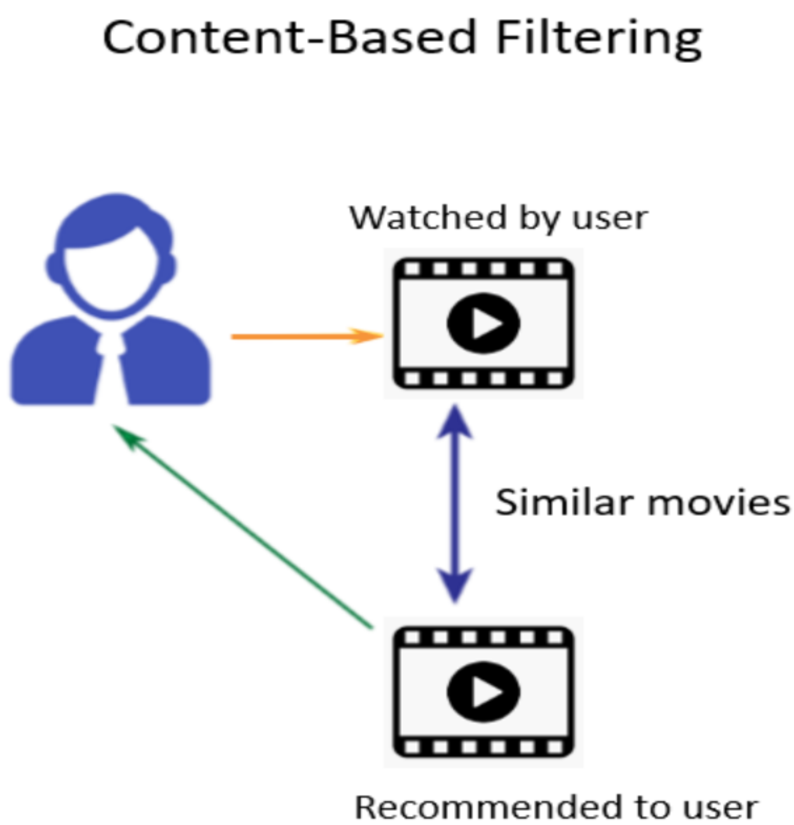
movies

Figure showing the movies.csv which is used as input in our project

The output of our project from Content Based Filtering is that this algorithm recommends movies which are similar to the ones that user liked before. For example, if a person has liked the movie "Inception", then this algorithm will recommend movies that fall under the same genre. In the Item-item collaborative filtering, or item-based, or item-to-item, is a form of collaborative filtering for recommender systems based on the similarity between items calculated using people's ratings of those items and based on the predicted rating from our algorithm recommends a list of movies which have the best rating to the user. The model based Collaborative Filtering calculates the similarity matrix which gives the best score and also recommends the matrix items with the best score to the user.

The main output of our project is that we try to recommend movies using various techniques like content based and collaborative filtering and compare the RMSE values of different techniques.

Content Based Filtering:



Content-Based recommender system tries to guess the features or behavior of a user given the item's features, he/she reacts positively to.

	Adventure	Animation	Children	Comedy	Fantasy	Romance	Drama	Action	Crime	Thriller
0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
1	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	1.0
3	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
4	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0

Above Image is a boolean matrix based on userInput genre.

Now, given these genres, we can know which users like which genre, as a result, we can obtain features corresponding to that particular user, depending on how he/she reacts to movies of that genre.

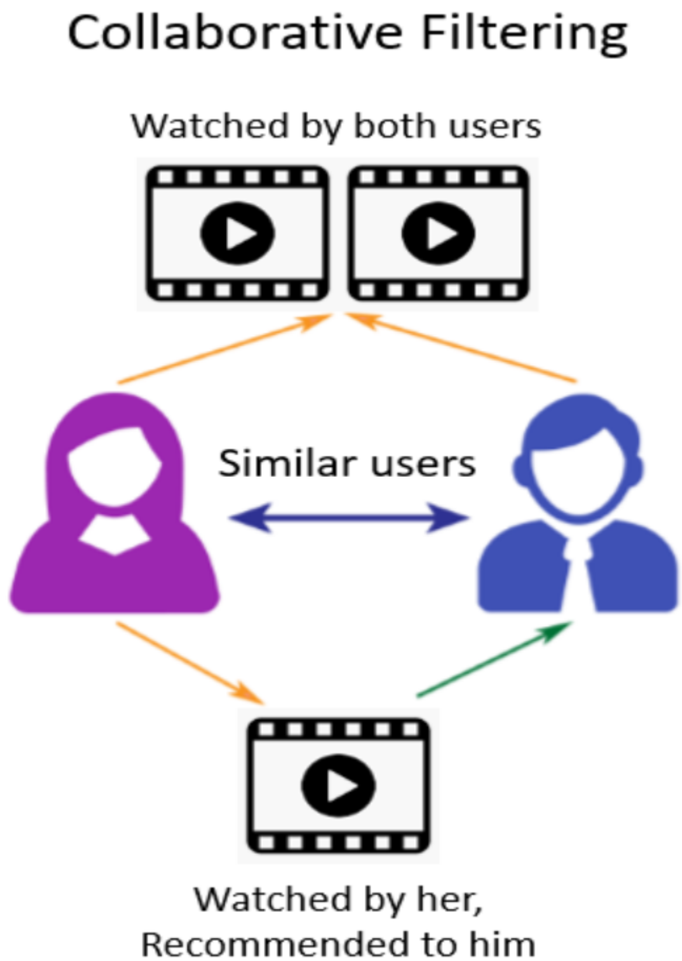
Once, we know the likings of the user we can embed him/her in an embedding space using the feature vector generated and recommend him/her according to his/her choice.

Collaborative Filtering

The technique we're going to take a look at is Collaborative filtering. It is based on the fact that relationships exist between products and people's interests. Many recommendation systems use collaborative filtering to find these relationships and to give an accurate recommendation of a product that the user might like or be interested in. As hinted by its alternate name, this technique uses other users to recommend items to the input user.

Memory Based

- User Based Collaborative Filtering



The process for creating a User Based recommendation system is as follows:

- Select a user with the movies the user has watched
- Based on his rating to movies, find the top X neighbours
- Get the watched movie record of the user for each neighbour.
- Calculate a similarity score using some formula
- Recommend the items with the highest score

	movieId	title	rating
0	1	Toy Story	3.894802
1	2	Jumanji	3.221086
2	3	Grumpier Old Men	3.180094
3	4	Waiting to Exhale	2.879727
4	5	Father of the Bride Part II	3.080811

Figure: User Input

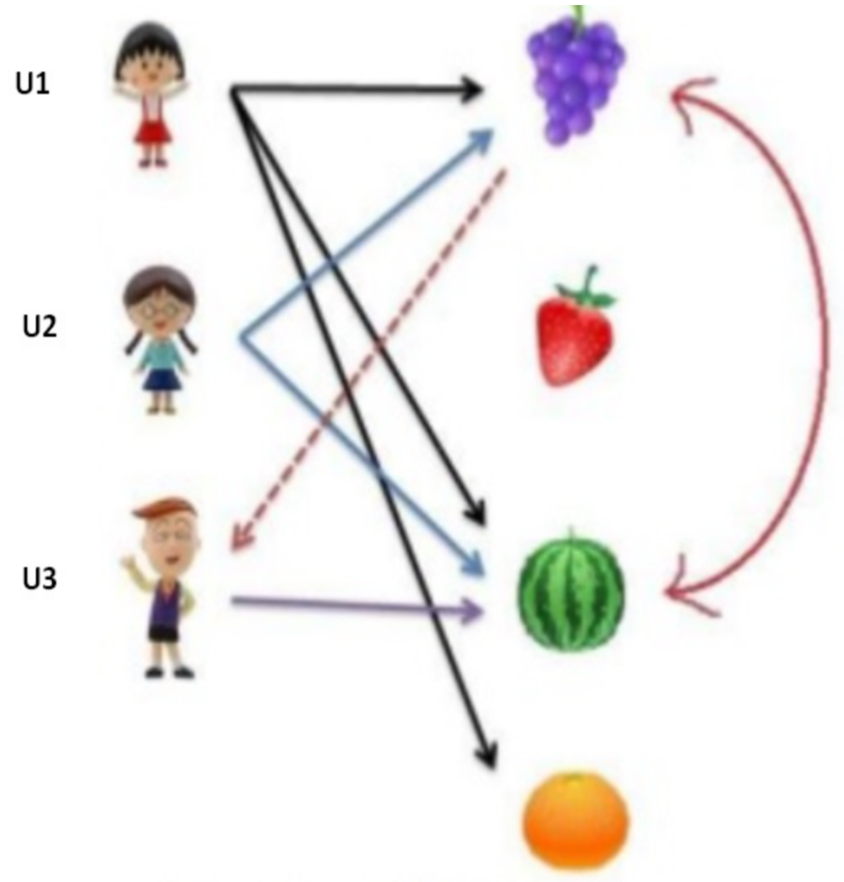
To find the similarity index between the user input and other movies in the dataset, we used the pearson correlation formula:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Pearson correlation is invariant to scaling, i.e. multiplying all elements by a nonzero constant or adding any constant to all elements. For example, if you have two vectors X and Y,then, pearson(X, Y) == pearson(X, 2 * Y + 3). This is a pretty important property in recommendation systems because for example two users might rate two series of items totally different in terms of absolute rates, but they would be similar users (i.e. with similar ideas) with similar rates in various scales. Based on the similarity index score, we recommend the movies which have high similarity score.

	weighted average recommendation score	movieId
movieId		
3814	5.0	3814
8580	5.0	8580
2295	5.0	2295
1999	5.0	1999
71268	5.0	71268
74282	5.0	74282
77846	5.0	77846
2771	5.0	2771
5915	5.0	5915
38304	5.0	38304

Item Based Collaborative filtering



Item-item collaborative filtering, or item-based, or item-to-item, is a form of collaborative filtering for recommender systems based on the similarity between items calculated using people's ratings of those items.

The Only difference between Item Based and User Based is that we calculate the similarity index between Items in Item based and calculate similarity between users in User based.

The Procedure is similar to User based except that we calculate similarities between the Items here instead of Users.

title	'burbs, The (1989)	(500) Days of Summer (2009)	10 Cloverfield Lane (2016)	10 Things I Hate About You (1999)	10,000 BC (2008)	101 Dalmatians (1996)	101 Dalmatians (One Hundred and One Dalmatians) (1961)
title							
'burbs, The (1989)	1.000000	0.063117	-0.023768	0.143482	0.011998	0.087931	0.224052
(500) Days of Summer (2009)	0.063117	1.000000	0.142471	0.273989	0.193960	0.148903	0.142141
10 Cloverfield Lane (2016)	-0.023768	0.142471	1.000000	-0.005799	0.112396	0.006139	-0.016835
10 Things I Hate About You (1999)	0.143482	0.273989	-0.005799	1.000000	0.244670	0.223481	0.211473
10,000 BC (2008)	0.011998	0.193960	0.112396	0.244670	1.000000	0.234459	0.119132

Figure: Similarities between Items

Model Based

Remembering the matrix is not required here. From the matrix, we try to learn how a specific user or an item behaves. We compress the large interaction matrix using dimensional Reduction or using clustering algorithms. In this type, We fit machine learning models and try to predict how many ratings will a user give a product.

Methods:

Clustering Algorithms: They normally use simple clustering Algorithms like K-Nearest Neighbours to find the K closest neighbors or embeddings given a user or an item embedding based on the similarity metrics used.

Code Snippet:

```
sim_options = {'name' : 'msd'}

algo = KNNBasic(k=20, sim_options=sim_options )
cross_validate(algo=algo, data=data, measures=['RMSE'], cv=5, verbose=True)
```

Matrix Factorization based algorithms: The Singular Value Decomposition (SVD), a method from linear algebra that has been generally used as a dimensionality reduction technique in machine learning. SVD is a matrix factorisation technique, which reduces the number of features of a dataset by reducing the space dimension from N-dimension to K-dimension (where K<N).

Code Snippet:

```
algo = SVD()
cross_validate(algo=algo, data=data, measures=['RMSE'], cv=5, verbose=True)
```

Results

Using the rmse/rmsd we have obtained the results of evaluation of each algorithm we have implemented. Lesser the value of rmse obtained from evaluation better the algorithm is.

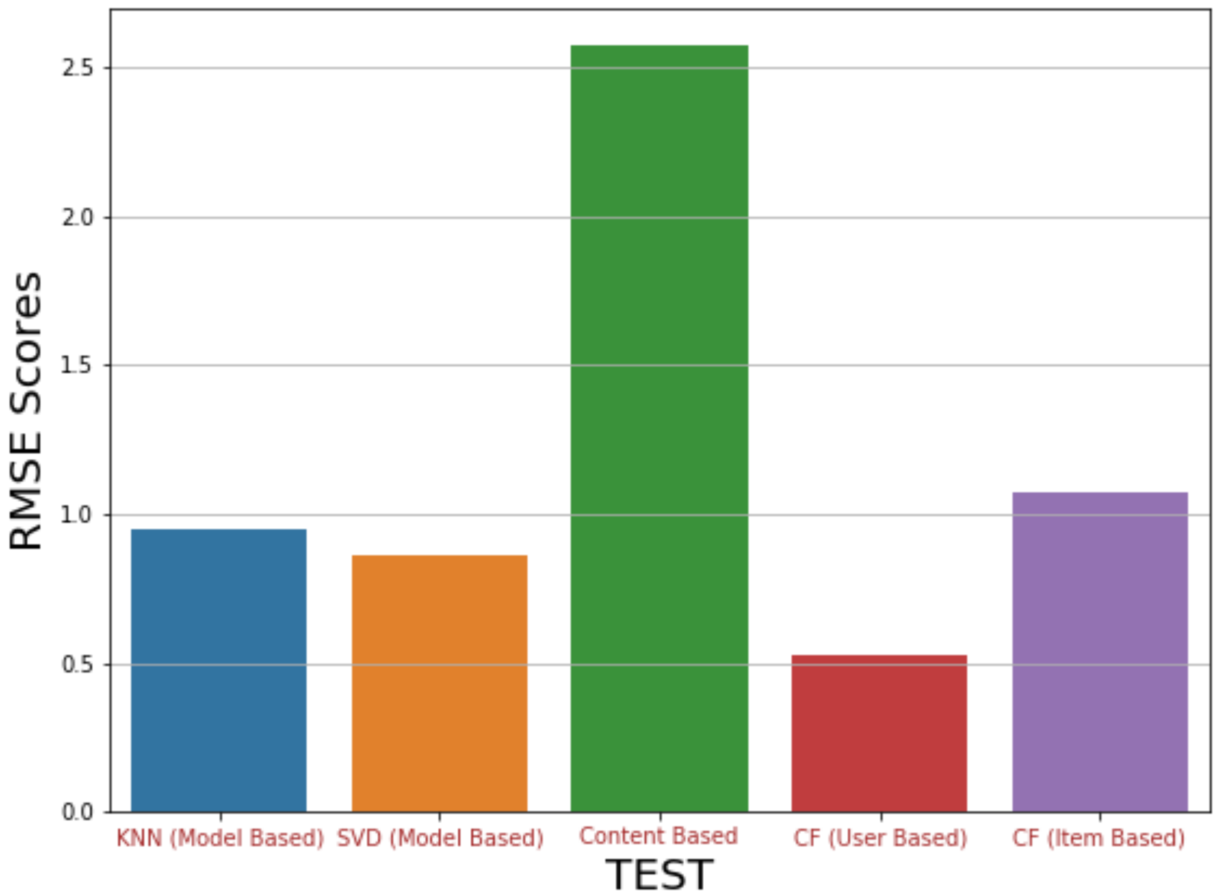
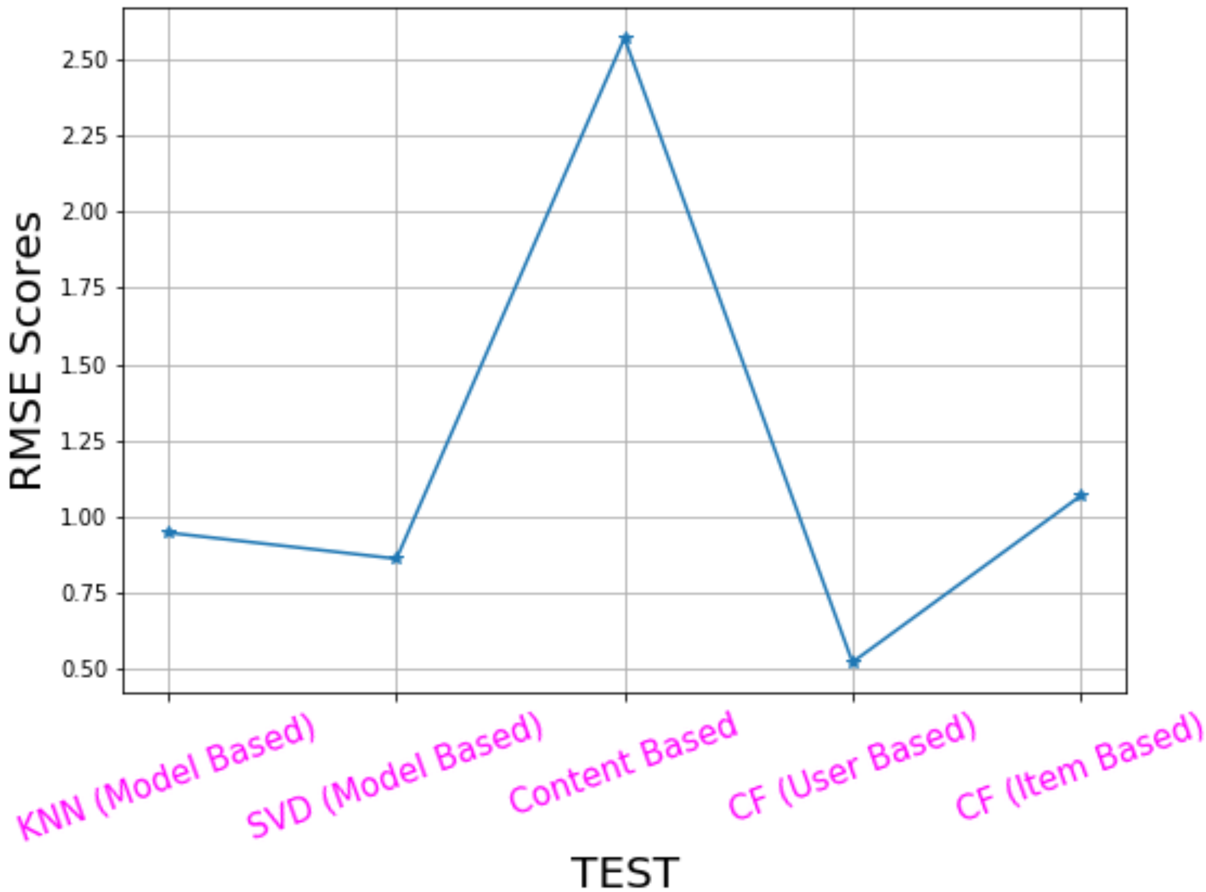


Figure showing rmse evaluation



From the above Inferences, we can conclude that Collaborative Filtering (User-Based) is better algorithm for recommender system using MovieLens 100k Dataset since it has a lower RMSE value compared to others.

Chapter 6: NEW LEARNINGS FROM THE TOPIC

- I learned about how the recommender System works

  and how can we implement recommender Systems in real-time

- It gives insight into the diverse fields in which Machine Learning is used, and how beneficial it has proven to be.

- Got to know that Collaborative Filtering (User-Based) provided us with the best RMSE score which is around 0.42 which is the least among all the other algorithms performed since it takes other users' ratings also into consideration and it adapts to the user's interests which might change over time.

REFERENCES AND ANNEXURES

[1] Aggarwal, C. C., Wolf, J. L., Wu, K., and Yu, P. S. (1999). Horting Hatches an Egg: A New Graph-theoretic Approach to Collaborative Filtering. In Proceedings of the ACM KDD'99 Conference. San Diego, CA. pp. 201-212.
[2] Basu, C., Hirsh, H., and Cohen, W. (1998). Recommendation as Classication: Using Social and Content-based Information in Recommendation. In Recommender System Workshop'98. pp. 11-15.
[3] Berry, M. W., Dumais, S. T., and O'Brian, G. W. (1995). Using Linear Algebra for Intelligent Information Retrieval. SIAM Review, 37(4), pp. 573-595.