VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

PROJECT WORK-4 REPORT
on

PROJECT TITLE

Submitted by

M K Gagan Roshan(1BM18CS049)
P Sai Deekshith(1BM18CS148)
Parmod D Y (1BM19CS405)
Ankit Kesar (1BM18CS150)

Under the Guidance of
Latha NR
Assistant Professor, BMSCE

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B. M. S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

Mar-2021 to Jun-2021

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work entitled "**Recommender system**" carried out by **M K Gagan Roshan (1BM18CS049), P Sai Deekshith(1BM18CS148), Ankit Kesar (1BM18CS150)** and **Pramod D Y (1BM19CS405)** who are bonafide students of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visveswaraiah Technological University, Belgaum during the year 2021. The project report has been approved as it satisfies the academic requirements in respect of **Project Work-4 (20CS6PWPW4)** work prescribed for the said degree.

Signature of the Guide	Signature of the HOD
Latha NR	Dr. Umadevi V
Assistant Professor	Associate Prof. & Head, Dept. of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

External Viva

Name of the Examiner	Signature with date
1.______________________________	_________________________
2. ______________________________	__________________________

B. M. S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECALARATION

We, M K Gagan Roshan (1BM18CS049), P Sai Deekshith (1BM18CS148), Ankit Kesar (1BM18CS150), Pramod D Y (1BM19CS405), students of 5th Semester, B.E, Department of Computer Science and Engineering, B. M. S. College of Engineering, Bangalore, here by declare that, this Project Work-1entitled "Recommender system" has been carried out by us under the guidance of Latha N R, Assistant Professor, Department of CSE, B. M. S. College of Engineering, Bangalore during the academic semester Mar-2021-Jun-2021

We also declare that to the best of our knowledge and belief, the development reported here is not from part of any other report by any other student.

Signature

M K Gagan Roshan (1BM18CS049)

P Sai Deekshith (1BM18CS148)

Ankit Kesar (1BM18CS150)

Pramod D Y (1BM19CS405)

1.Introduction

1.1 Motivation for the Project :

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.

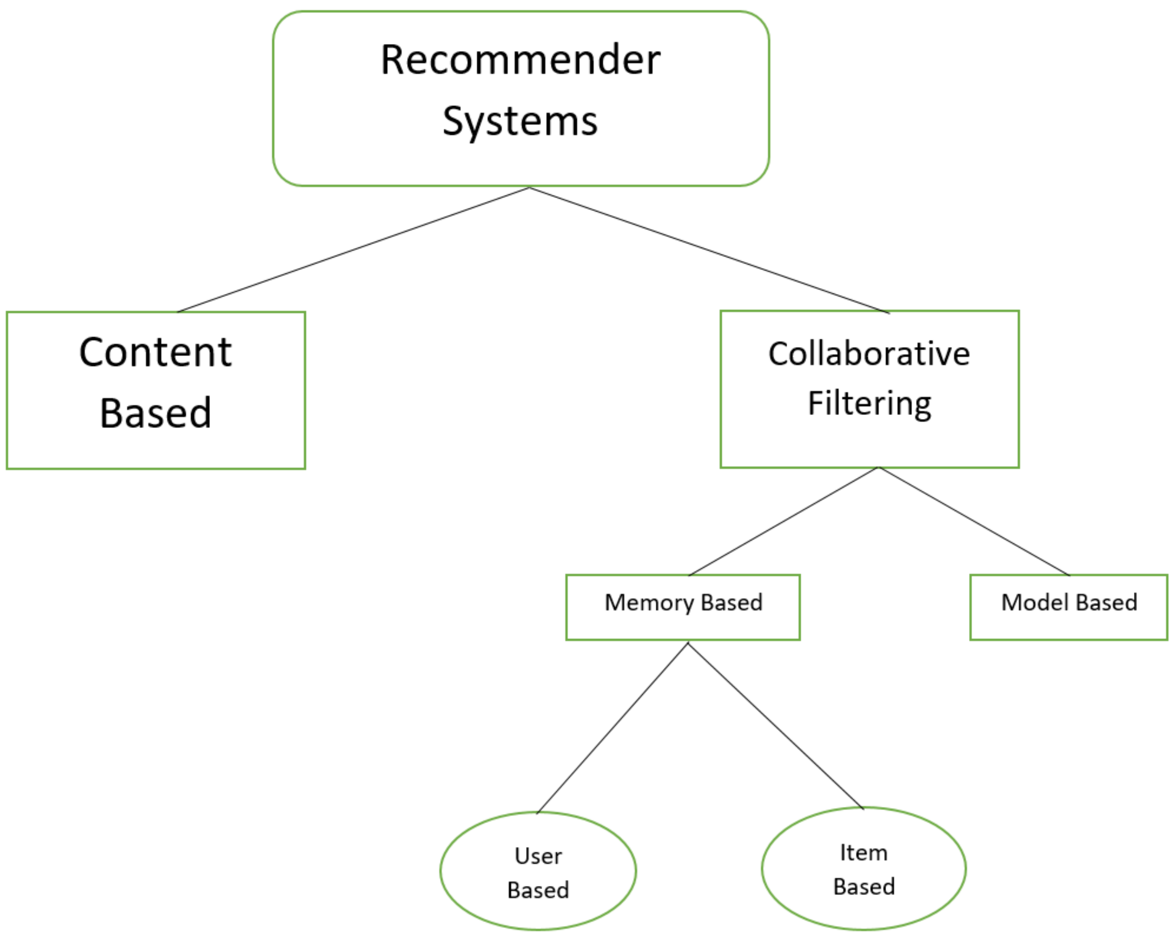
The purpose of a recommendation system basically is to search for content that would be interesting to an individual. Moreover, it involves a number of factors to create personalized lists of useful and interesting content specific to each user/individual. Recommendation systems are Artificial Intelligence 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 least RMSE value.

1.2 Brief Summary of the Project

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) Collaborative Filtering:

The technique we have implemented 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.

- Memory based
  - User based
  - Item based
- Model based
  - SVD
  - KNN

2) Content Based Filtering

This algorithm recommends product 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.

2. Problem Definition and Algorithm:

2.1 Task Definition

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	userId	movieId	rating	timestamp
2	1	169	2.5	1.2E+09
3	1	2471	3	1.2E+09
4	1	48516	5	1.2E+09
5	2	2571	3.5	1.44E+09
6	2	109487	4	1.44E+09
7	2	112552	5	1.44E+09
8	2	112556	4	1.44E+09
9	3	356	4	9.21E+08
10	3	2394	4	9.21E+08
11	3	2431	5	9.21E+08
12	3	2445	4	9.21E+08
13	4	16	4	1.04E+09
14	4	39	4	1.04E+09
15	4	45	4	1.04E+09
16	4	47	2	1.04E+09
17	4	94	5	1.04E+09
18	4	101	4	1.04E+09
19	4	246	4	1.04E+09
20	4	288	2	1.04E+09
21	4	296	4	1.04E+09
22	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.

1	movieId	title	genres
2	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy
3	2	Jumanji (1	Adventure|Children|Fantasy
4	3	Grumpier	Comedy|Romance
5	4	Waiting to	Comedy|Drama|Romance
6	5	Father of t	Comedy
7	6	Heat (199!	Action|Crime|Thriller
8	7	Sabrina (1	Comedy|Romance
9	8	Tom and H	Adventure|Children
10	9	Sudden De	Action
11	10	GoldenEye	Action|Adventure|Thriller
12	11	American	Comedy|Drama|Romance
13	12	Dracula: D	Comedy|Horror
14	13	Balto (199	Adventure|Animation|Children
15	14	Nixon (19!	Drama
16	15	Cutthroat	Action|Adventure|Romance
17	16	Casino (19	Crime|Drama
18	17	Sense and	Drama|Romance
19	18	Four Roon	Comedy
20	19	Ace Ventu	Comedy
21	20	Money Tra	Action|Comedy|Crime|Drama|Thriller
22	21	Get Shorty	Comedy|Crime|Thriller

movies

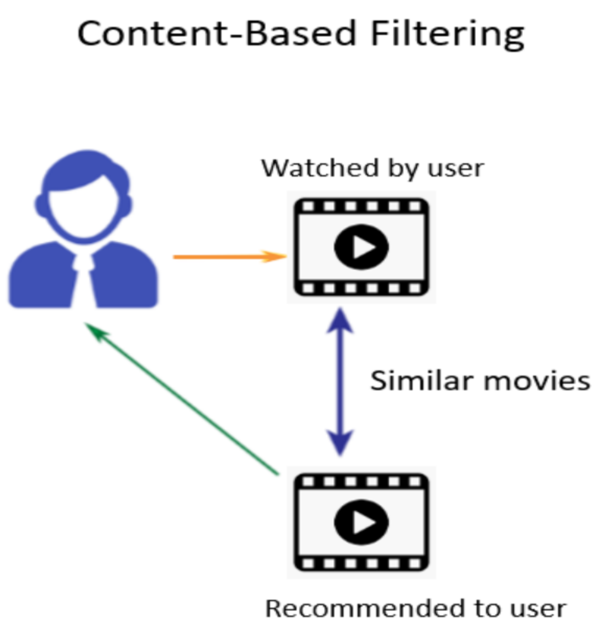
Figure showing the movies.csv which is used as input in our project

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.

2.2 Algorithm Definition

2.2.1 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.

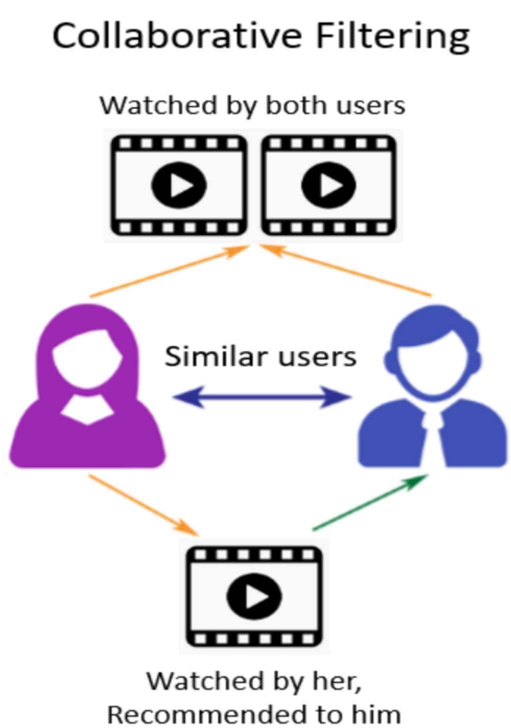
Once, we know the likings of the user we can embed him/her in an embedding space using the feature vector generated and recommend him/her according to his/her choice.

2.2.2 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.

2.2.2.1 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

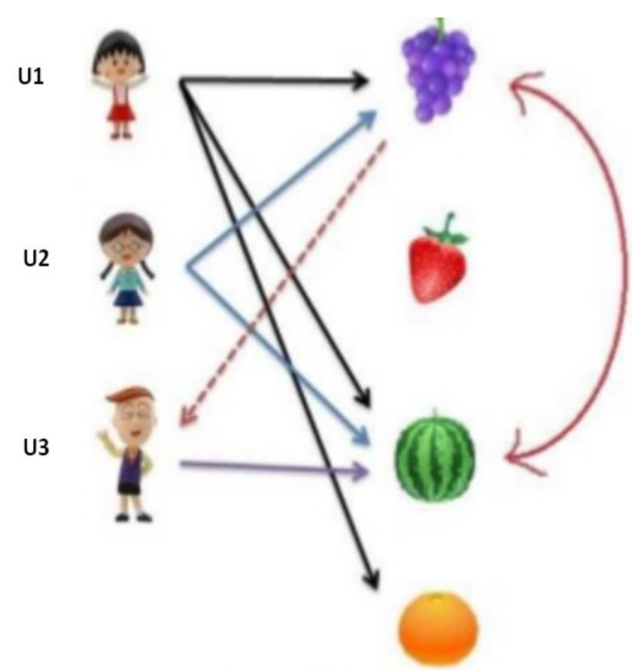
To find the similarity index between the user input and other movies in the dataset, we used the pearson correlation formula:

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

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

2.2.2.2 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)
```

3. Experimental Evaluation

3.1 Methodology

Since recommender systems does not come into the category of supervised algorithms, we do not have many evaluation metrics to find the best algorithm.

The evaluation technique we use here is RMSE (Root Mean Square Error).

Root Mean Square Error (RMSE) is a standard way to measure the error of a model in predicting quantitative data. Lower values of RMSE indicate better fit. RMSE is a good measure of how accurately the model predicts the response, and it is the most important criterion for fit if the main purpose of the model is prediction. Lower values of RMSE indicate better fit. RMSE is a good measure of how accurately the model predicts the response, and it is the most important criterion for fit if the main purpose of the model is prediction.

Formally it is defined as follows:

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} \left(x_i - \hat{x}_i\right)^2}{N}}$$

The above formula is used to calculate the error value of the algorithm by finding the differnece between the true ratings and the predicted ratings and squaring them.

	Test	RMSE
0	KNN (Model Based)	0.948328
1	SVD (Model Based)	0.863072
2	Content Based	2.568551
3	CF (User Based)	0.524199
4	CF (Item Based)	1.068792

Figure: RMSE values of different techniques

3.2 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.

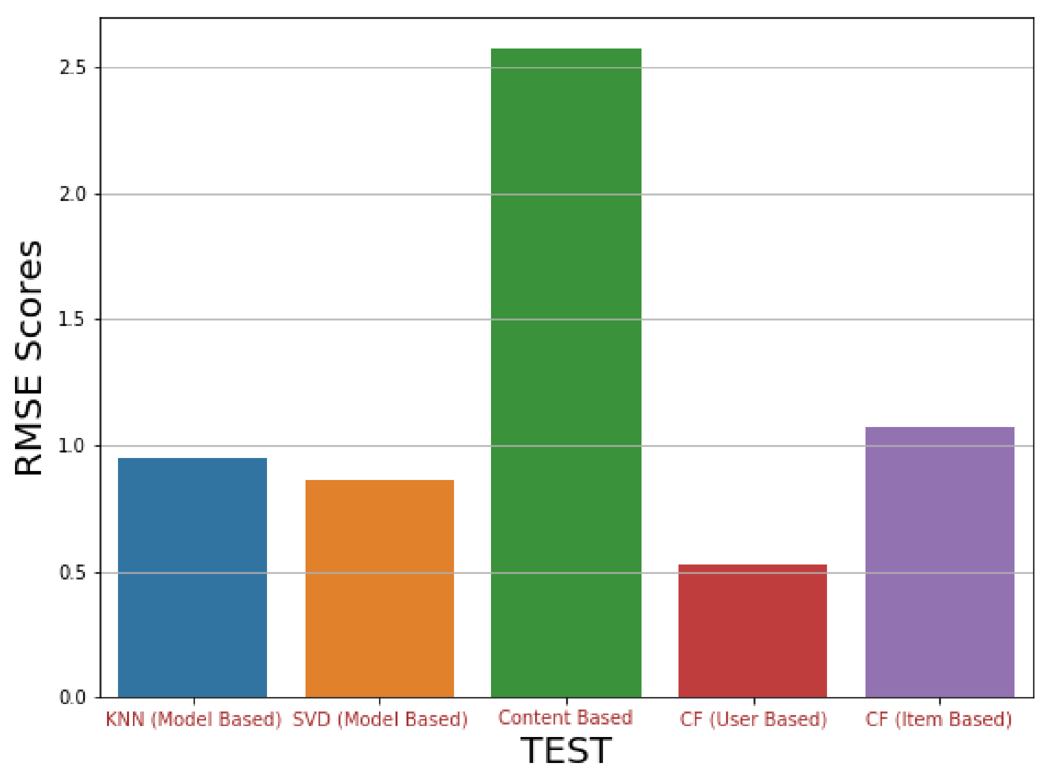
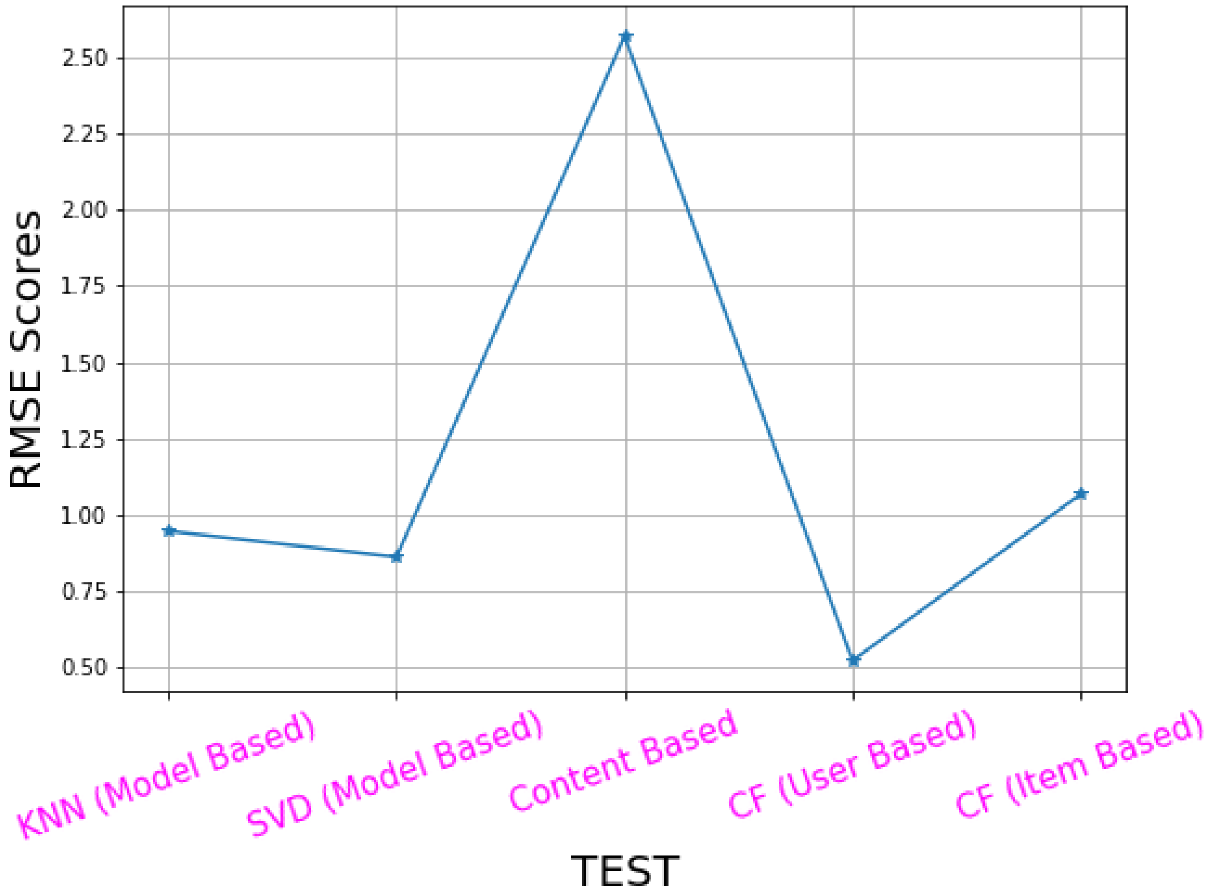


Figure showing rmse evaluation



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.

3.3 Discussion

The result screenshots that are obtained our project provide us with the following analysis:

- As expected, the Content Based Filtering gives us the least performance with a RMSE value of 2.5 approximately since doesn’t take into account what other user think of the movie, so low quality recommendations might happen.
- All other algorithms gave a good score of RMSE value compare to content based recommender system.
- Model Based using SVD and KNN also gave a good RMSE score ~ 1.
- Collaborative Filtering (User Based) provided us with the best RMSE score which is around 0.42 which is least among all the other algorithms performed since it takes other user’s ratings also into consideration and it adapts to the user;s interests which might change over time.

4. Future Works

Our future work includes improving the features of our implemented system. We have decided to extend our system to run against a list of book items and also recommending electronic items such as those present in e commerce sites like amazon.

Improving the advantages of our system is also the agenda of our future work. Our system is already better than some of the related recommender systems.

Further work includes implementing our algorithm such that it runs against real time and recommend items at a very faster rate than the related works of recommender systems.

Mitigating the disadvantages of our system also is part of our agenda of our future work. The predicted rating from the approximation function needs improving such that rating predicted is very near to the one used in the dataset.

Our Future works also include in making a hybrid recommender system which is a combination of content based and collaborative filtering technique.

5. Conclusion

All in all, recommender systems can be a powerful tool for any Movie Recommendation, and rapid future developments in the field will increase their business value even further.

We could recommend movies using various algorithmic techniques and also evaluate based on their RMSE values.

In this project we tried to recommend movies based on Movie Lens data set On successful evaluation of all our algorithms using **RMSE** we can concluded that Collaborative Filtering (User Based) is best algorithm for recommendation system.

The best RMSE score we could get for User Based algorithm is ~ 0.52.

6. Bibliography

https://marutitech.com/recommendation-engine-benefits/.

https://towardsdatascience.com/how-to-build-a-model-based-recommendation-system-using-python-surprise-2df3b77ab3e5

https://towardsdatascience.com/recommendation-systems-models-and-evaluation-84944a84fb8e

https://realpython.com/build-recommendation-engine-collaborative-filtering/#how-to-find-similar-users-on-the-basis-of-ratings