International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

Book Recommendation using Collaborative Filtering

Prasanta Kumar Sahoo*, S. Dhanish, Venkat Ramana, A. Nikith Kumar
Professor, Dept. of Computer Science and Engineering Sreenidhi Institute of Science and Technolgy, Hyderabad.
B. Tech CSE students, Dept. of CSE, Sreenidhi Institute of Science and Technolgy, Hyderabad-501301.

Abstract: It is becoming a very difficult task for the users to select the appropriate books for a specific topic as there were a lot of choices available. There is a need for a system which takes user preferences into consideration while searching and recommending online books to the user. So the objective of this research work is to design an application that recommends books based on users ratings. The system being proposed uses machine learning algorithm like collaborative filtering [CF] that first construct the user-item interaction matrix, then construct vector matrix using cosine similarity measure from user-item interaction matrix and then find the similarity between the books using vector matrix and recommend the top n books similar to the book given by the user as input to the algorithm. The results indicate that recommendation performance is better when both average ratings and cosine vector similarity measure is used as compared to the existing systems.

Keywords: Collaborative Filtering Technique [CF], Cosine Similarity, Euclidean distance similarity, RMSE

1. INTRODUCTION

Recommendation system used to suggest items to users based on criteria like past purchases, search history, and other factors. Recommendation system finds items based on user preference and solve the problem of information overloading. It enables the user by providing personalized services and user based content and saves a lot of time and money. In 2009 Netflix launch a competition to improve the accuracy of its movie recommender system by 10%. The recommendation system is very much important in increasing the revenue generation of a company. As per the statistics 35 percent of consumers purchase on Amazon and more than a half of what they watch on Netflix come from recommendations systems. Usually, suggestions from friends and family are always useful to read books or watch anything new but even after looking through all suggestions we may not find something of our preferences, Hence there is a need for a system that takes our preferences into consideration before recommending anything. We do not want to spend time searching for books that we prefer so, we create recommendation tool using collaborative filtering where users can give the name of the book as input and items like the input item are suggested. We implemented two methods of recommendation a popularity-based recommendation using total rating and average ratings of users and a collaborative filtering algorithm by applying cosine vector similarity which measure dot product of two vectors (Book data). Cosine similarity is used to measure similarities in book dataset and find books with high similarities with given book. This recommendation can also be used to recommend movies, articles, music, and news.

2. RELATED WORK

Recommender systems have become extremely common and are utilized in a variety of areas: some popular applications include movies, books, research articles, and social tags. There are three basic categories of recommendation algorithms: collaborative filtering, content-based filtering, and hybrid recommendation.

Collaborative filtering methods are based on collecting and analyzing a large amount of information on users' behaviors, activities or preferences and predicting what users will like based on their similarity to other users. The proposed algorithm does not need professional knowledge, and the recommendation effect will become better and better with the interest of the user, but there are data sparsity and other problems. There are two types of Collaborative Filtering techniques: Item-based and User-based collaborative filtering [CF].Similarity between items or users can be calculated using different similarity metrics like cosine, Euclidean, Jaccard.

Content-based filtering methods are based on a description of the item and a profile of the user's preference. These algorithms try to recommend items that are similar to those that a user liked in the past. Hybrid recommendation is combining collaborative filtering and content-based filtering. These methods can also be used to overcome some of the common problems in recommender systems such as cold start and the sparsity problem.

3. EXITING SYSTEM

Okon et.al. (2018) [1] proposed a model that generates recommendations to buyers, through an enhanced CF algorithm, a quick sort algorithm and Object-Oriented Analysis and Design Methodology (OOADM). Scalability was ensured through the implementation of Firebase SQL. This system performed well on the evaluation metrics.

Kurmashov et.al. (2015) [2] used Pearson correlation coefficient-based CF to provide internet based recommendations to book readers and evaluated the system through an online survey.

Mathew et.al. (2016) [3] proposed a system that saves details of books purchased by the user. From these Book contents and ratings, a hybrid algorithm using collaborative filtering, content-based filtering and association rule generates book recommendations. Rather than Apriori, they recommended the use of Equivalence class Clustering and bottom-up Lattice Transversal (ECLAT) as this algorithm

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

is faster due to the fact that it examines the entire dataset only once.

Parvatikar et.al. (2015) [4] proposed item-based collaborative filtering and association rule mining to give recommendations. Similarity between different users was computed through Adjusted Cosine Vector Similarity function. Better recommendations were obtained as through this method data sparsity problem was removed.

Ayub et.al. (2018) [5] proposed a similarity function like Jaccard Similarity to locate alike items and users for the enquiring item and user in nearest neighbor based collaborative filtering. They proposed that absolute value of ratings should be taken as against the ratio of co-rated items taken in Jaccard Similarity. They also compared performance of their method with other similarity measures.

Gogna and Majumdar [6] suggested the use of buyer's demographic and item category to overcome data sparsity and cold start problems in their movie recommendation system. Latent Factor Model (LFM) was used. They developed a matrix to match the buyer and user information to get a dense user and dense item matrix. Label Consistency map, the outcome of this system was used to suggest unrated and other items to new buyers.

Chatti et.al. (2013) [7] suggested tag-based and rating based CF recommendation in technology enhanced learning (TEL) to resolve the data sparsity problem and extract relevant information from the rating database. Memory and model oriented 16 varied tag-based Collaborative filtering algorithms were evaluated for buyer satisfaction and accuracy of recommendations in Personal Learning Environments

Choi et.al. (2010) [8] proposed RS based on HYRED, a hybrid algorithm using both content and collaborative filtering on a compact dataset (by reducing user interest items) and neighbor data. HYRED used altered Pearson Coefficient based Collaborative filtering and distance-to-boundary (DTB) Content filtering. This would result in better and faster recommendation for large amount of data.

Liu et.al. (2012) [9] added the dimension of user-interest. They proposed iExpand, a 3-tier model i.e. user, user-interest and item. This helped in overcoming the issues of overspecialization and cold start as well as reducing computation costs.

Feng et.al. (2018) [10] proposed a RS for movies based on a similarity model constituted of factors S1 (similarity between users), S2 (ratio of co-rated items) and S3 (user's rating choice weight). This RS was particularly useful for sparse datasets.

According to Aggarwal [11], C.C., Collaborative Filtering method is used in recommendation systems to develop recommendations based on ratings provided by the other users of the system. If buyer's ratings of items match, it is likely that their ratings of other items will also match, this is the basic assumption of CF. Computers cannot gauge qualitative factors such as taste or quality, therefore recommendations based on the ratings of humans who can rate on the basis of qualitative factors, i.e., collaboration, will give a better outcome.

Gattu Vijaya Kumar, Prasanta Kumar Sahoo, K.Eswaran [12] explain the main reason we need a recommendation system in the current generation is because humans have extremely many alternatives to utilize required information which is popular from the Internet. It implements five classification algorithms such as Support-Vector Machines, Logistic Regression, Multinomial Naive Bayes, Multilayer Perceptron and K-Nearest Neighbors .It was observed that from the comparison of all the algorithms Support-Vector Machine gives 75.13% accuracy.

Prasanta kumar sahoo, Kodaty Sri Sai Chaitany and N. Archana [13] suggested most of the e-commerce retailers are suffering, in displaying the targeted and relevant results for a search keyword given by customers. In this scenario analyzing their past interests and recent behavior of the customers is one of the important aspects to confirm the user relevant search results. The online customer behavior has been analyzed by using Homophily Detection Algorithm in this research work. They implements Behavior analysis, Trend analysis and Personalization on customer data and displaying relevant search results when a customer search for a particular product which leads to greater customer satisfaction.

Literature survey suggested that recommendation systems are being used by large number of online marketers to increase their sales by offering products to customers which match their tastes.

4. PROPOSED SYSTEM

This research work mainly focuses on popularity based method and Collaborative Filtering Technique for the recommendation system. Collaborative filtering [CF] technique primarily focuses on the relationship between users and items. It is a technique that can filter out items that a user might like on the basis of ratings given by the other users and recommends the top-n similar items to the user. The similarity of items is determined by the similarity of the ratings of those items by the users who have rated both items. Item-based collaborative filtering is being used by Amazon for customers. In an existing system where there are more users than items, item-based filtering is faster and more stable than user-based. This algorithm uses a similarity measure to find similarity between items.

5. METHODOLOGY

1. Import all necessary libraries like pandas, numpy, seaborn etc

2. Data Preprocessing, in this step the dataset is checked for missing values and null values in the data and remove or fill them.

3. The design of the recommendation system based on two methods as given below:

a. Popularity based:

To implement popularity based recommendation, the data set is selected by merging the two data frames. One is data frames for no of ratings and the other one is average ratings and merge them. Popular books are those whose average rating is more than 15 ratings are selected as per the criteria and top 5000 results are printed.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

b. collaborative filtering:

To implement recommendation based on collaborative filtering, cosine similarity is being used to find the similarity of user preferences and recommend books based on cosine similarity of book.

6. OVERVIEW OF DATASET

6.1 Importing Dataset

Datasets are critical component of AI development because they provide the training data that is used to train and test machine learning models. In this project Datasets such as Books.csv, Ratings.csv was collected from Kaggle for the implementation purpose.

6.2 Merging Datasets

The data set is taken by merging Books.csv and Rating.csv to form bookrec.csv. The data set Bookrec.csv contains seven attributes and in the process unnecessary attributes are removed. The attributes of Bookrec.csv are: User-ID, ISBN, Book-Rating, Book-Title, Book-Author, Year-Of-Publication.

User-ID	ISBN	Book-Rating	Books.Book-Title	Books.Book-Author	Books.Year-Of-Publication
276725	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002
2313	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002
6543	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002
8680	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002
10314	034545104X	9	Flesh Tones: A Novel	M. J. Rose	2002
23768	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002
28266	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002
28523	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002
39002	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002
50403	034545104X	9	Flesh Tones: A Novel	M. J. Rose	2002
56157	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002
59102	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002
59287	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002
2	195153448	0	Classical Mythology	Mark P. O. Morford	2002
276726	155061224	5	Rites of Passage	Judith Rae	2001
8	2005018	5	Clara Callan	Richard Bruce Wright	2001
11400	2005018	0	Clara Callan	Richard Bruce Wright	2001
11676	2005018	8	Clara Callan	Richard Bruce Wright	2001
41385	2005018	0	Clara Callan	Richard Bruce Wright	2001
276727	446520802	0	The Notebook	Nicholas Sparks	1996
278418	446520802	0	The Notebook	Nicholas Sparks	1996
638	446520802	0	The Notebook	Nicholas Sparks	1996
3363	446520802	0	The Notebook	Nicholas Sparks	1996
7158	446520802	10	The Notebook	Nicholas Sparks	1996

6.2.1 Information of the dataset

Df.head

```
In [4]: df.head()
```

Out[4]:

	User-ID	ISBN	Book-Rating	Books.Book-Title	Books.Book-Author	Books.Year-Of-Publication
0	276725	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002.0
1	2313	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002.0
2	6543	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002.0
3	8680	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002.0
4	10314	034545104X	9	Flesh Tones: A Novel	M. J. Rose	2002.0

6.3 Data cleaning

In data cleaning step the dataset is analyzed and checked for null values, missing values, and duplicate values The goal of data cleaning is to ensure that the data is accurate, consistent, and free of errors. It is important to clean data as leaving them can negatively impact ML models. This can be done using the pandas.

```
In [5]: print('shape:',df.shape)

        shape: (1048575, 6)

In [6]: df.isnull().sum()

Out[6]: User-ID                     0
        ISBN                        0
        Book-Rating                 0
        Books.Book-Title       107937
        Books.Book-Author      107938
        Books.Year-Of-Publication 107941
        dtype: int64
```

```
In [7]: #REMOVE ALL NULL ROWS
        df=df.dropna()
        df
```

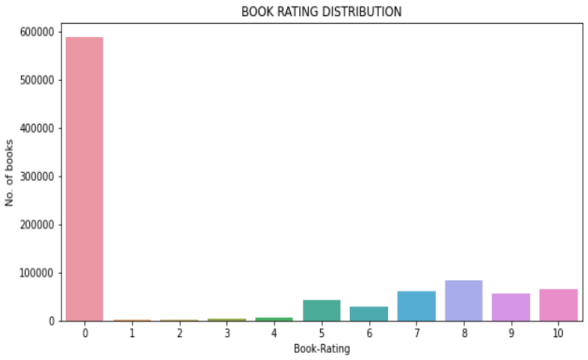
Out[7]:

	User-ID	ISBN	Book-Rating	Books.Book-Title	Books.Book-Author	Books.Year-Of-Publication
0	276725	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002.0
1	2313	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002.0
2	6543	034545104X	0	Flesh Tones: A Novel	M. J. Rose	2002.0
3	8680	034545104X	5	Flesh Tones: A Novel	M. J. Rose	2002.0
4	10314	034545104X	9	Flesh Tones: A Novel	M. J. Rose	2002.0
...	...	...	...	...	...	...
1048570	250764	0451410777	0	Sleep Tight (Onyx Book)	Anne Frasier	2003.0
1048571	250764	0452264464	8	Beloved (Plume Contemporary Fiction)	Toni Morrison	1994.0
1048572	250764	048623715X	0	Glamorous Movie Stars of the Thirties: Paper D...	Tom Tierney	1982.0
1048573	250764	0486256588	0	Schiaparelli Fashion Review: Paper Dolls in Fu...	Tom Tierney	1988.0
1048574	250764	0515069434	0	Lady Laughing Eyes (To Have and to Hold)	Lee Damon	1984.0

940633 rows × 6 columns

6.4 data visualization

```
fig = plt.figure(figsize = (10, 5))
sns.countplot(df['Book-Rating'])
 plt.xlabel("Book-Rating")
plt.ylabel("No. of books")
plt.title("BOOK RATINGDISTRIBUTION")
plt.show()
```



6.5 popularity-based recommendation.

```
num_rating_df=df.groupby ('Books.Book-Title').count()['Book-Rating'].reset_index()
num_rating_df.rename(columns= {'Book-Rating':'Num_ratings'} ,inplace=True)
avg_rating_df=df.groupby ('Books.Book-Title').mean()['Book-Rating'].reset_index()
avg_rating_df.rename(columns= ,inplace=True)
```

```
In [15]: popular_df = num_rating_df.merge(avg_rating_df, on='Books.Book-Title')
         popular_df = popular_df[popular_df['Num_ratings']>=20].sort_values('Avg_ratings', ascending=False)
         popular_df = popular_df.merge(df, on='Books.Book-Title').drop_duplicates('Books.Book-Title')[['Books.Book-Title',
         'Books.Book-Author','Num_ratings','Avg_ratings']]
         popular_df
```

Out[15]:

	Books.Book-Title	Books.Book-Author	Num_ratings	Avg_ratings
0	Free	Paul Vincent	51	8.039216
51	Chobits (Chobits)	Clamp	25	7.920000
76	El Hobbit	J. R. R. Tolkien	25	7.600000
101	Johnny Got His Gun	Dalton Trumbo	30	7.133333
131	Where the Sidewalk Ends : Poems and Drawings	Shel Silverstein	33	7.121212
...	...	...	...	...
388738	The Plan	Stephen J. Cannell	31	0.322581
388769	Go Eat Worms! (Goosebumps, No 21)	R. L. Stine	24	0.291667
388793	Stolen Blessings	Lawrence Sanders	25	0.240000
388818	Et the Extra Terrestrial	William Kotzwinkle	25	0.200000
388843	Hot Flashes	Barbara Raskin	25	0.000000

6836 rows × 4 columns

6.6 Cosine vector similarity (CVS):

Cosine similarity is measured by the cosine of angle between vectors and finds, if they point same direction. It measures cosine angle by using dot product of vectors. It

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

measures similarity between two vectors. It is represented by:

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Case study 1
Here, A=First vector
B=Second vector
Ai=User rating of book(A) by i
Bi=User rating of book(B) by i
Example:
B1= Harry potter: Goblet of fire
B2=Harry potter: Deathly hallows
After creating a word table from the Books, Books can be represented by the following vectors:

Books	Harry	Potter	Goblet	Of	Fire	Deathly	Hallows
B1	1	1	1	1	1	0	0
B2	1	1	0	0	0	1	1

B1= {1,1,1,1,1,0,0}
B2= {1,1,0,0,0,1,1}
Using these two vectors we can calculate cosine similarity. First, we calculate the dot product of the vectors:
B1. B2= 1.1+1.1+1.0+1.0+1.0+0.1+0.1 =2
Magnitude of vectors:
||B1||= √5
||B2||= √4
Cosine similarity (B1, B2)
=**2**/ √5. √**4** =0.44721

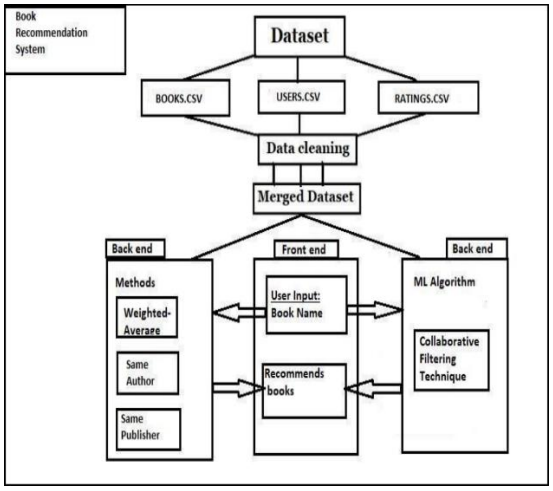
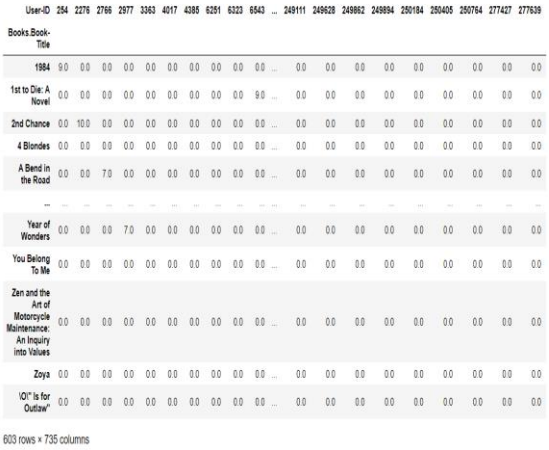


Fig 1: shows the System Architecture

Cosine similarity implementation
1. Create a matrix showing similarities between books in dataset.
2. Import cosine similarity from sklearn library and create cosine matrix using similarities in dataset.
3. Find books with high similarity score with the book given by user.
4. Recommend books with high similarities with the book given by user.



User-ID	254	2276	2766	2977	3363	4017	4385	6251	6323	6543	...	249111	249628	249862	249894	250184	250405	250764	277427	277639
Books.Book-Title																				
1984	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1st to Die: A Novel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	9.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2nd Chance	0.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4 Blondes	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A Bend in the Road	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Year of Wonders	0.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
You Belong To Me	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Zen and the Art of Motorcycle Maintenance: An Inquiry into Values	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Zoya	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
'Night Is for Outlaw'	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

603 rows × 735 columns

Data similarity matrix

6.7 Cosine similarity importing and implementation.

- Importing cosine similarity

```
from sklearn.metrics.pairwise import cosine_similarity
similarity_scores = cosine_similarity(pt)
similarity_scores
```

```
array([[1.        , 0.11013674, 0.0127091 , ..., 0.12447747, 0.07552261,
        0.04645965],
       [0.11013674, 1.        , 0.22459428, ..., 0.07780233, 0.1752651 ,
        0.12821608],
       [0.0127091 , 0.22459428, 1.        , ..., 0.04617038, 0.05001715,
        0.11450939],
       ...,
       [0.12447747, 0.07780233, 0.04617038, ..., 1.        , 0.07085128,
        0.02054493],
       [0.07552261, 0.1752651 , 0.05001715, ..., 0.07085128, 1.        ,
        0.11104108],
       [0.04645965, 0.12821608, 0.11450939, ..., 0.02054493, 0.11104108,
        1.        ]])
```

- Recommending books using similarity scores

```
In [23]: recommend('Outlander')

Out[23]: ['Drums of Autumn',
          'DIANA GABALDON',
          1997.0,
          'Voyager',
          'DIANA GABALDON',
          1994.0,
          'Dragonfly in Amber',
          'DIANA GABALDON',
          1993.0,
          'The Bourne Identity',
          'Robert Ludlum',
          1984.0,
          'The Search',
          'Iris Johansen',
          2000.0]
```

7. CONCLUSION

Recommendation systems are very popular in e-commerce applications such as online book store and can significantly rise the company's revenue generation. Recommendation system helps us to lighten the information overloading problem where the user will have a lot of choices and not able to understand perfectly what to buy or what to see. Recommendation system solves this problem and provides users with personalized content after searching a large volume of information. The proposed work to design a

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

recommendation system based on popularity method and collaborative filtering is much more accurate than the existing systems.

8. REFERENCES

[1] Okon, E.U., Eke, B.O. and Asagba, P.O. (2018). An improved online book recommender system using collaborative filtering algorithm. International Journal of Computer Applications (0975-8887) Volume 179-No.46, June 2018

[2] Kurmashov, N., Konstantin, L., Nussipbekov, A. (2015). Online book recommendation System. Proceedings of Twelve International Conference on Electronics Computer and Computation (ICECC)

[3] Mathew, P., Kuriakose, B. And Hegde, V. (2016). Book Recommendation System through content based and collaborative filtering method. Proceedings of International Conference on Data Mining and Advanced Computing (SAPIENCE)

[4] Parvitikar, S. and Dr. Joshi, B. (2015). Online book recommendation system by using collaborative filtering and association mining. Proceedings of IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) International Conference on Physics and Photonics Processes in Nano Sciences Journal of Physics: Conference Series 1362 (2019) 012130 IOP Publishing doi:10.1088/1742-6596/1362/1/012130 8

[5] Ayub, M., Ghazanfar, M.A., Maqsood, M. and Saleem, A. (2018). A Jaccard base similarity measure to improve performance of CF based recommendation system. Proceedings of International Conference on Information Networking (ICOIN)

[6] Gogna, A., Majumdar, A. (2015). A Comprehensive Recommender System Model: Improving Accuracy for Both Warm and Cold Start Users. IEEE Access Vol. 3, 2803-2813, 2015

[7] Chatti, M.A., Dakova, S., Thus, H. and Schroeder, U. (2013). Tag-Based Collaborative Filtering Recommendation in Personal Learning Environments. IEEE Transactions on Learning Technologies, Vol. 6, No. 4, October-December 2013

[8] Choi, S.H., Jeong, Y.S. and Jeong, M.K. (2010). A Hybrid Recommendation Method with Reduced Data for Large-Scale Application. IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews, Vol. 40, No.5, September 2010.

[9] Liu, Q., Chen, E., Xiong, H., Ding, C.H.Q. and Chen, J. (2012). Enhancing Collaborative Filtering by User Interest Expansion via Personalised Ranking. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, Vol. 42, No.1, February 2012.

[10] Feng, J., Fengs, X., Zhang, N. and Peng, J. (2018). An improved collaborative filtering method based on similarity. PLOS ONE 13 (9): e0204003, September 2018.

[11] Aggarwal, C.C. (2016). Recommendation System: The Textbook. XXI, 29 p. ISBN 978-3-319- 29657-9.

[12] Gattu Vijaya Kumar, Prasanta Kumar Sahoo, K.Eswaran, "A Recommendation System & Their Performance Metrics using several ML Algorithms", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-3, February 2020.

[13] Prasanta kumar sahoo, Kodaty Sri Sai Chaitany and N. Archana, " ANALYZING CUSTOMER BEHAVIOR IN E-COMMERCE USING HOMOPHILY DETECTION ALGORITHM", Advances and Applications in Mathematical Sciences Volume 20, Issue 12, Pages 3235-3255.2021.

Dr. Prasanta Kumar Sahoo

Professor in the Department of Computer Science & Engineering, Sreenidhi Institute of Science & Technology affiliated to JNTUH. He has completed his Ph.D. from Fakir Mohan University, Odisha in Computer Science Engineering. He has 19 years of teaching, research and administrative experience. He has earlier worked as Head of the Dept. in both CSE and IT dept. in various reputed Engineering Colleges. His Research interest includes Cyber Security, Information Security, Data Science and Machine Learning. He has published around 60 research papers in various reputed journals both at national and International level. Many times Dr. Prasanta Kumar Sahoo won the best teacher award in various colleges for his contribution to the teaching and learning process. He is Certified Professional from BalaBit, completed Electronic Contextual Security Intelligence exam Intermediate Level (ECSI). He has guided more than 50 projects both at UG and PG level. He has delivered more than 15 guest lecturers. He has organized three national conference and nine faculty development program with immense success.

S. Dhanish, Venkat Ramana and A. Nikith Kumar are B. Tech IV year students in Computer Science and Engineering at Sreenidhi Institute of Science & Technology affiliated to JNTUH.