ISSN2515-8260 Volume 10, Issue 02, 2023

Mechanism For Recommending Web Series

Dr.M.Rajaiah,Dean Academics & HOD, Dept of CSE, Audisankara College of Engineering and Technology, Gudur.

Mr.A.Venkateswarlu,Assistant Professor ,Dept of CSE, Audisankara College of Engineering and Technology, Gudur.

Mr.T.Sai Sudeep,UG Scholar, Dept of CSE, Audisankara College of Engineering and Technology, Gudur.

Mr.T.Harish,UG Scholar, Dept of CSE, Audisankara College of Engineering and Technology, Gudur.

Mr.S.Sumanth,UG Scholar, Dept of CSE, Audisankara College of Engineering and Technology, Gudur.

Mr.Sk.Wahed Ali, UG Scholar, Dept of CSE, Audisankara College of Engineering and Technology, Gudur.

ABSTRACT:

Many businesses today aim to provide useful product suggestions to online users in order to increase their consumption on websites. People usually choose or buy a new product based on the recommendations of friends, comparisons of similar products, or feedback from other users. A recommender system must be implemented in order for all of these tasks to be completed automatically. Recommender systems are tools that provide suggestions that best suit the client's needs, even if the client is unaware of it.Personalized content offers are based on past behaviour, and they entice customers to return to the website. A web series recommendation mechanism for Netflix/Prime/Disney plus Hotstar will be built in this paper. The dataset used in this study contains over 5 K web series and 500 K+ customers. Popularity, Collaborative Filtering, Content-based Filtering, and Hybrid Approaches are the four main types of recommender algorithms. This paper will introduce all of them. We will choose the algorithms that best fit the data, implement them, and compare them

Keywords: Content Based Filtering, Popularity Based filtering, Hybrid Approaches, Collaborative Filtering

1.INTRODUCTION:

Online Platforms like Amazon Prime, Netflix handles a big collection of webseries and web series by streaming them at any time through computer , Tv , Mobile Phones. This firms are profitable because the users pay the some amount of money every month to access them.

Users can cancel their subscription at any time.

The popularity of recommendation systems among service providers is growing because they help to increase the number of items sold, offer a diverse selection of items, user satisfaction increases, as does user fidelity to the company, and they are quite helpful in having a better understanding of what the user wants.

The recommender systems consider not only information about the users, but also the items they consume, comparisons with other products, and so on. Nonetheless, there are numerous algorithms available for use in a recommendation system. For example, I Popularity, which recommends only the most popular items; and (ii) Collaborative Filtering, which searches for patterns in user activity to produce user-specific recommendations. (iii) Content-based Filtering, the recommendation of items containing similar information to what the user has previously liked or used (description, topic, among others) (iv) Hybrid Approaches, which combine the two algorithms mentioned previously.

Choosing the algorithm that best fits the analysis is a difficult task, and neither expands the user's taste into adjacent areas by improving the obvious. As a result, the main types of recommender algorithms will be introduced in this paper, along with the pros and cons of each algorithm to provide a better understanding of how they work. As a result, several algorithms will be tested in the end to determine which one works best for Netflix users.

This study is based on real data from Netflix users and the ratings they gave to the web series they watched. The information contains 17,770 files, one for each movie, and each movie has a customer rating on a five-star scale from 1 to 5. The movie file also includes the year of release and the title of the film.

2.PROPOSED SYSTEM:

The proposed system uses a set of different filtration strategies and algorithms to help users find the most relevant web series. The most popular categories of the Machine learning algorithms used for web series recommendation system includes content - based filtering and collaborative filtering systems.

3. LITERATURE SURVEY:

We currently live in an era of information. We are surrounded by a plethora of data in the form of reviews, blogs, papers and comments on various websites. The number of people around the world who use the internet has witnessed an increase of approximately 40% since 1995 and reached a count of 3.2 billion. The increased information flow has opened more avenues, but it has also led to added confusion for the user. Amidst this huge amount of data, the task of making certain decisions becomes difficult. It is rightly said that one should make an informed decision, but too much information can also hinder the decision-making process. Thus, in order to save a user from this confusion and make the experience of surfing the internet a pleasurable one, recommender systems were introduced. Francesco Ricci, LiorRokach and BrachaShapira define the recommender systems as software tools that make relevant suggestions to a user [1], [2]. Depending upon the user profile and the product profile, which are formed using various techniques and algorithms, suggestions are made. More than 32% of consumers rate a product online, over 33% writes reviews and nearly 88% trust online reviews [14]. Thus, reviews play an essential role in affecting the sales of a commodity or a service. Each review posted on the web consists of the user's sentiments (positive or negative) and preferences. Sentiment analysis helps in determining the attitude of the writer by computationally dividing opinions in a piece of text into positive, negative or neutral [11].

Data Analysis

Data exploration

The data file was divided into four documents, each containing the Movie ID, Customer ID, Rating with values ranging from 1 to 5, and the date the ratings were given. The four documents were then combined, yielding a total of 17,770 web series, 480,189 users, and 100,498,277 ratings. This means that not all of the web series have been rated by users. And the data is distributed as shown in Figure.

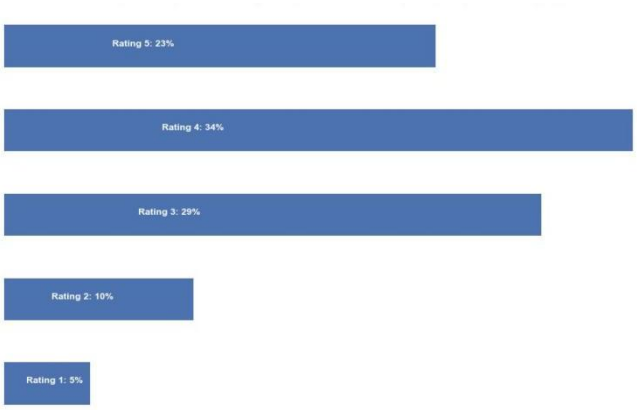


Fig:Rates distribution

Figure shows that only 15% of the webseries ratings are negative (1 or 2), with the remaining 75% providing relatively positive feedback. This could happen because if a user is watching a movie that he does not like, he will simply leave without rating it. However, low ratings indicate that the film is not particularly good. We can also see that the most common value is 4. Because a rating of 0 represents a missing value, it is not displayed in the analysis. as illustrated in Figure

We also obtained another data file containing movie information, which includes the Movie Id, the title of the movie, and the year of release. However, the title information is incomplete because when a movie's title contains more than 49 characters, the title ends there. Because the movie information was insufficient, it was only used for descriptive purposes. This also means that none of the content-based or hybrid filtering approaches can be used because we lack information about the users' profiles and the movie titles are insufficient.The below figure depicts the data set's number of web series per year, which includes 17,770 films. This data set contains web series from 1896 to 2005, with nearly 40% of them released between the years 2000 and 2004.

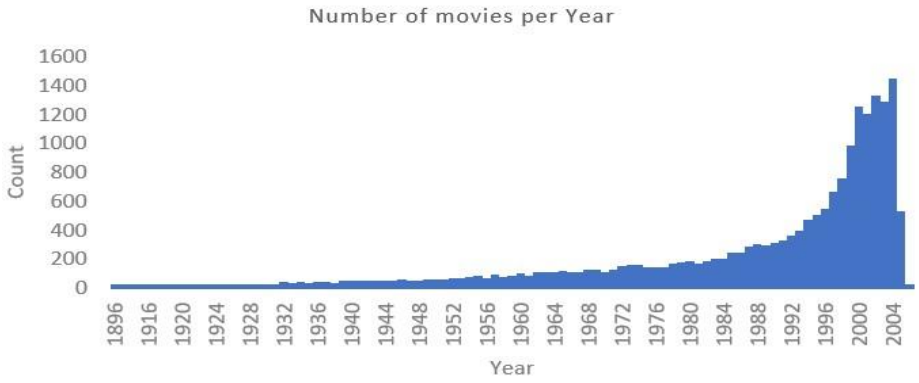


FIGURE : Number of web series per year of release

We can delve deeper into the rate distribution analysis and compute the average rating per film. The graph depicts the distribution of the average movie rating. The distribution shows that the highest value is around 3, with a small number of films having an average rate of 1 or 5. This data set is very large and has a lot of zero values, which means that there are several web series that have only been rated a few times or users who have only rated a few web series, so those users should not be considered.

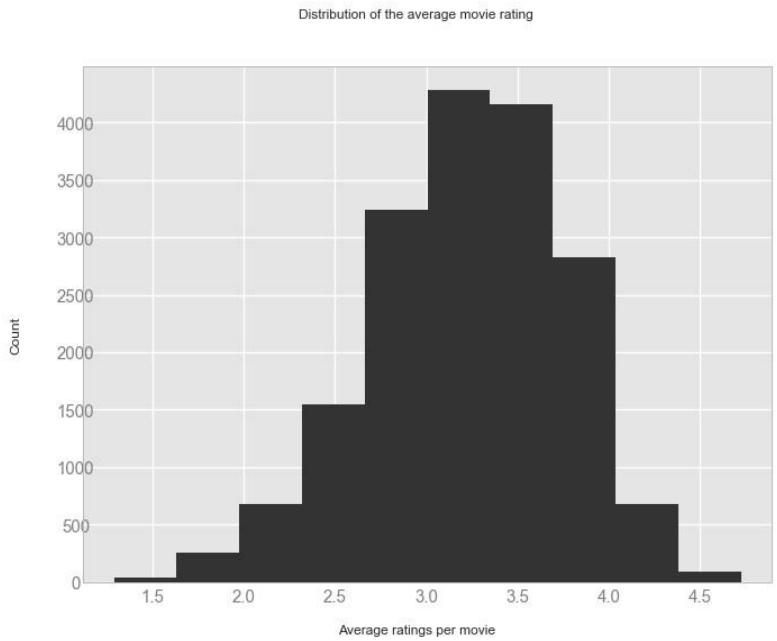


FIGURE : Average Rates distribution

In addition, we can notice in Table a that 80% of the web series have been rated less than 4,040 times, while the most watched movie counts with 232,944 ratings, then those web series are not too popular.

The average rate of the web series that have the largest number of ratings is 4, while the less rated web series have an average of 3, the most rated movie has an average rate of 5.Table b displays the Distribution of the times of review per user, where we can notice that there is a group of users who are relatively less active than the rest, for instance the 80% of the users have review maximum 322 web series, which implies that those users have rated less than 1% of the web series.

Similar to the table above, the average rating of the web series that have been rated for several users is around 4, and the users who have rated less number of web series have an average rating between 3 and 4.

TABLE : Distribution of the times of review

(A) Per movie (B) Per user

web series	Times of review	average rate
10 %	117	3
	161	3
20 %	228	3
	350	3
30 %	561	3
	1006	3
40 %	1948	4
	4040	4
50 %	12304	4
	23294	5
60 %		
70 %		
80 %		
90 %		
100 %		

Users	Times of review	average rate
10 %	19	3
	31	3
20 %	46	3
	66	4
30 %	96	4
	142	4
40 %	211	4
	322	4
50 %	541	4
	17653	5
60 %		
70 %		
80 %		
90 %		
100 %		

Data preparation

It was noted in the previous section that there is a group of web series that have been rated by a few users, implying that their ratings may be biassed. Furthermore, there is a group of users who have only rated a few web series, so their ratings may be biassed as well. Given the lack of information in both cases, this information must be excluded from the analysis.

Based on the information described above, and in order to prepare the data for use in recommender models. It is critical to I select the relevant data, which means reducing data volume by improving data quality, and (ii) normalise the data, which means removing some extreme values in user ratings.

Having above benchmark will help us to improve not only the quality of the data but also the efficiency. As a result, we decide to work with web series that have been rated more than 4,040 times and users who have rated more than 322 web series. After reducing the data, we get 56,222,526 ratings. This means that the data set was reduced by nearly half its original size.

After removing the web series with fewer than 5,000 views, we can see that the distribution of the average rate has shifted (Figure 3.4), and the majority of the ranks are now between 3,5 and 4. The extreme values were removed, as expected, but the highest values remained nearly unchanged. The number of web series has also decreased; in Figure 3.1, the count ranged from 0 to over 4,000, and it now ranges from 1 to nearly 1,000. Table shows a significant shift in the distribution of review times per movie and per user.

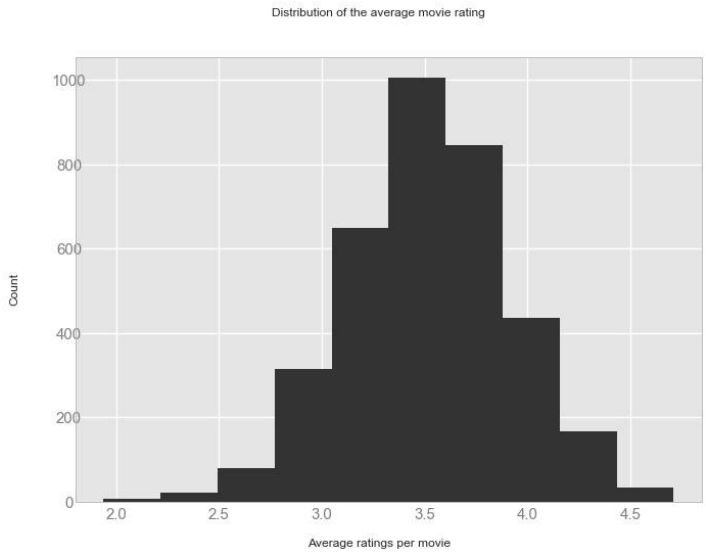


FIGURE: Average Rates distribution after data cleaning

TABLE: Distribution of the times of review after data cleaning

(A) Per web series (B) Per user

web series	Times of review	average rate

Users	Times of review	average rate

10 %	3636	3
20 %	4451	3
30 %	5516	3
40 %	7057	4
50 %	9202	4
60 %	12478	4
70 %	17243	5
80 %	24885	
90 %	40801	
100 %	83640	

10 %	325	3
20 %	358	3
30 %	396	3
40 %	441	4
50 %	494	4
60 %	560	4
70 %	645	4
80 %	768	4
90 %	974	4
100 %	3534	5

The second step in this data preparation is normalizing the data, this step is also important because there are some users who have given low or high values to the web series and this might lead to bias in the results. This problem is easily solved by normalizing the data in order to obtain and average rate of 0 per user. The final step is to create the user-item matrix necessary to implement the recommender systems approach. The dimensions of the matrix are 96,290 × 3,554 . Which indicates our clean data set counts with 92290 users and 3554 web series.

Implementation

As previously stated, the implementation of memory-based techniques is computationally expensive. As a result, we will work with a sample by reducing the number of users and web series. Because the number of users may cause a problem with model accuracy, it is preferable to reduce the number of users on a larger scale than the number of web series, so we used 25% of the users and 60% of the web series. The matrix of ratings is now 24,072 2,132, for a total of 9,272,642 ratings.

We can calculate the average number of neighbours and the average number of ratings again using the formulas from Table 3.3 and the sample data. The results are shown in

Table 4.1, and while the average number of potential neighbours for the User-based CF is now 24,071, the number of potential ratings is still very low 69. The accuracy obtained from User-based CF will then be subpar and will remain computationally expensive in comparison to Item-based CF.

TABLE : Calculation of the average number of neighbors and average number of ratings for the sample

	Avg. Neighbors	Avg. Ratings
User-based	24,071	69
Item-based	2,131	785

Consequently, for Memory-based, just Item-based CF will be implemented using as similarity measure the cosine and Pearson correlation. For Model-based techniques, the SVD approach will be executed. The results from both techniques will be compared.

Now, in order to identify the most suitable model, we are going to build, evaluate and compare the following filtering

Popularity: Most popular items will be displayed.

IBCF_cos: Item-based collaborative filtering, using the cosine as the distance function.

IBCF_cor: Item-based collaborative filtering, using the Pearson correlation as the distance function.

SVD: Singular Value Decomposition

Random: Random recommendations in order to have a baseline.

Popularity

The popularity approach was explained, in which we mention that we can recommend the most viewed and higher-rated web series.

The top ten most matched web series are determined by counting the number of users who have rated each film, and the average rating of each film is calculated for the top ten better-rated films.

Both results are shown in Tables a and b, respectively. We can see that the top ten for both approaches recommend different web series. As previously stated, it is not the best solution because it lacks variety, but it is very useful and simple to implement.

TABLE a: Top most watched web series

position	Web Series_Id	Name	Year
1	5317	Breaking Bad	2,000
2	15124	Salvation	1,996
3	14313	Money Heist	2,000
4	15205	Game of Thrones	2,004
5	1905	Vikings	2,003
6	6287	Locke and Key	1,990
7	11283	Stranger Things	1,994
8	16377	The Witcher	1,999
9	16242	The lost Kingdom	1,997
10	12470	Manifest	1,996

TABLE b: Top better rated web series

position	web series_Id	Name	Year	Rating
1	14961	Breaking Bad	2,003	4.72

2	7230	Money Heist	2,001	4.72
3	7057	Sex Education	2,002	4.70
4	3456	Squid game	2,004	4.67
5	9864	Cobra Guy	2,004	4.64
6	15538	Glow up	2,004	4.61
7	8964	Blue Print	2,003	4.60
8	14791	Big Mouth	2,003	4.60
9	10464	Mom	1,995	4.60
10	14550	House of Cards	1,994	4.59

5.2 Evaluating the ratings

The other four models will now be evaluated. In order to properly evaluate the models, the training and test sets must be created, as previously explained, where the ratings in the test set are those that are not in the train set, but the user and the item are in both sets.

Table shows the RMSE and MAE for each algorithm. The SVD is followed by item-based CF using Pearson correlation, which has a smaller standard deviation of the difference between the real and predicted ratings. Nonetheless, all of the recommenders outperform a random suggestion, demonstrating the value of implementing any of these methodologies.

	RMSE	MAE
IBCF_cor	0.6675	0.5163
SVD	0.7098	0.5526
IBCF_cos	0.8769	0.6831
Random	1.4259	1.144

Table : Accuracy measures

From the results in above Table we noticed that ICBF_cor has a smaller RMSE and MAE than SVD. Nevertheless, we desire to execute a more detailed inspection between the difference of the predictions for the algorithm SVD and the IBCF_cor. For instance, in Table a are displayed some of the predictions from the IBCF_cor when SVD has an error larger than b, which shows that the IBCF_cor does not do it much better.

TABLE : IBCF_cor predictions when the SVD has a huge error

Cust Id	Series Id	Rating	Estimated Rating	Error
727242	3743	5	2.089	2.911
727242	6910	5	1.965	3.035
727242	11771	5	1.596	3.404
727242	14042	5	1.599	3.401
727242	16459	5	1.970	3.030
291503	3624	1	4.437	3.437
1452708	7767	1	4.419	3.419
873713	10928	1	3.718	2.718
2606799	9886	1	4.092	3.092
1697754	15296	1	3.857	2.857

In order to visualize how different are the predictions from both algorithms. The number of predictions for each rating value was calculated, and its distribution is displayed in figure 4.1

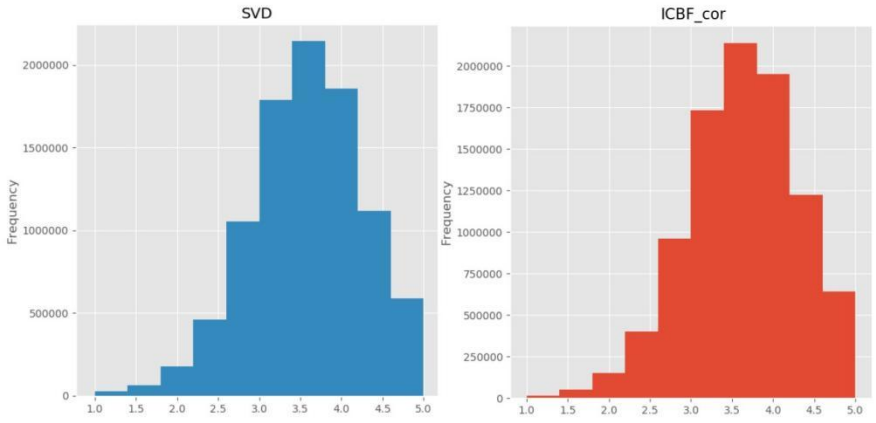
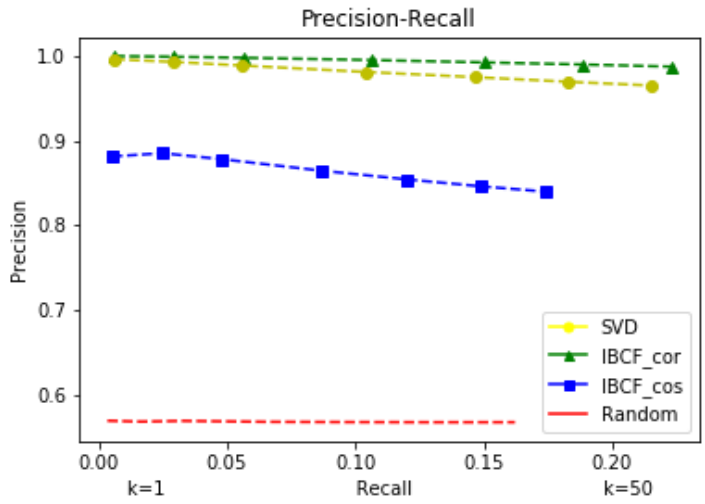


FIGURE : Number of predictions for each rating value

It is well known that when a user has rated only a small number of items, the predictions from these algorithms are not very accurate. So, when the user rated less than 100 web series, we calculated the mean error per algorithm, which for the IBCF cor was 0.48 and for the SVD was 0.52. The top model is still the ICBF with Pearson correlation distance.

Evaluating the recommendations

On the other hand, we can measure the accuracies of the algorithms by comparing the recommendations with the purchases, as was explained in Formulas 2.11 and 2.12. With a rating threshold of 4 for positive ratings, and a number k of the highest predicted ratings k = (1, 5, 10, 20, 30, 50).



In Figure the Precision and Recall are displayed, where we can see that for few recommendations like 1 or 5, IBCF_cor and SVD have a high precision but really low recall. Once the number of recommendations increases (k=50), the recall increases as well, and the performance of ICBF with Pearson correlation distance has a small decrease, however IBCF_corstills the one with the highest precision. Having a large precision implies over all items that have been recommended, the ones that the system is recommending are relevant. But the low value of the recall indicates a low proportion of all relevant items are being recommended. Depending on what we want to achieve, we can set an appropriate number of items to recommend

4.CONCLUSION

We covered the theory of the most popular recommendation system algorithms, including Popularity, Collaborative Filtering, Content-based Filtering, and Hybrid Approaches, in this paper. Based on this discussion, only Popularity and Collaborative Filtering were implemented, while Memory-based CF and Model-based CF were used for CF. The issue with Popularity is that all of the recommendations are the same for every single user, so we didn't pay attention to these results. Memory-based models rely on the similarity of users or items.

Based on the results, we can conclude that Item-Based CF with Pearson correlation as a similarity measure outperformed all other algorithms. With an RMSE of 0.6675, MAE of 0.5163, and precision and recall of 0.9959 and 0.006 for 1 recommendation, 0.9649 and 0.2148 for 50 recommendations, respectively. Outperforms the SVD, especially as the number of recommendations increases. Nonetheless, all of the algorithms outperformed the random recommendation, indicating that we can make good recommendations from a data set of ratings using Collaborative filtering that is not only memory-based (neighbourhood models) but also model-based (matrix factorization models).

Theoretically, SVD should have outperformed the Item-based approach because Low-dimensional recommenders are attempting to capture the taste and preferences of the users, and it is well known that SVD is a good approach for providing recommendations based on people's preferences. However, because of the approximation of SVD with gradient descent, this methodology achieves better and more accurate results in large datasets. Because we only used a sample of the data set, this could explain why it performed worse than the Item-based method. Further research will be interested in comparing the models without reducing the data set; this will be more computationally expensive.

References:

[1] Francesco Ricci, LiorRokach, BrachaShapira and Paul B. Kantor - Recommender Systems Handbook; First Edition; Springer-Verlag New York, Inc. New York, NY, USA, 2010.

[2] Tariq Mahmood and Francesco Ricci," Improving recommender systems with adaptive conversational strategies", 20th ACM conference on Hypertext and Hypermedia, pp. 73–82, ACM, July 2009.

[3] Tariq Mahmood, Francesco Ricci, Adriano Venturini and Wolfram Höpken, "Adaptive recommender systems for travel planning", Information and Communication Technologies in Tourism 2008, Proceedings of the International Conference,Innsbruck Austria, pp. 1 - 11, 2008.

[4] X. Su and T. Khoshgoftaar, "A survey of collaborative filtering techniques," Advances in Artificial Intelligence, vol. 2009, pp. 19, August 2009.

[5] Yongfeng Zhang, Min Zhang and Yiqun Liu, "Incorporating Phrase-level Sentiment Analysis on Textual Reviews for Personalized Recommendation",Eighth ACM International Conference on Web Search and Data Mining, pp. 435 – 440, February 2015.

[6] P. Resnick and H. R. Varian, "Recommender systems," Communications of the ACM, vol. 40, no. 3, pp. 56–58, 1997.

[7] J. Bennett and S. Lanning, "The Netflix Prize", ACM SIGKDD Explorations Newsletter - Special issue on visual analytics, Vol. 9 Issue 2, pp. 51 – 52, December 2007.

[8] Titov and R. McDonald, "A joint model of text and aspect ratings for sentiment summarization", Annual Meeting of the Association for Computational Linguistics, pp. 308 – 316, June 2008.

[9] P. Lops, M. de Gemmis and G. Semeraro, "Content-based recommender systems: State of the art and trends", Recommender Systems Handbook, pp. 73 - 105, 2011.

[10] X. Ding, B. Liu, and P. S. Yu, "A Holistic Lexicon-Based Approach to Opinion Mining", Web Search and Data Mining, pp. 231 - 239, February 2008.

[11] A. Kennedy and D. Inkpen, "Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters", Computational Intelligence, Vol. 22, No. 2, pp. 110 – 125, May 2006.

[12] A. Andreevskaia, S. Bergler and M. Urseanu "All Blogs Are Not Made Equal: Exploring Genre Differences in Sentiment Tagging of Blogs", International Conference on Weblogs and Social Media (ICWSM-2007), Boulder, CO, March 2007.

[13] AntonisKoukourikos, GiannisStoitsis and Pythagoras Karampiperis, "Sentiment Analysis: A tool for Rating Attribution to Content in Recommender Systems", 7 th European Conference on Technology Enhanced Learning, September 2012.

[14] Bo Pang and Lillian Lee, "Opinion Mining and Sentiment Analysis", Foundations and Trends in Information Retrieval, Vol. 2, Issue 1 – 2, pp. 1 – 135, January 2008.

[15] Umberto Panniello, Alexander Tuzhilin, Michele Gorgoglione, CosimoPalmisano and AntoPedone, " Experimental Comparison of Pre- vs. Post-Filtering approaches in Context-Aware Recommender Systems", ACM Conference on Recommender Systems, October 2009.

[16] Fan Yang and Zhi-Meiwang, "A Mobile Location-based Information Recommendation System Based on GPS and WEB2.0 Services", WSEAS Transactions on Computers, Vol. 8, Issue 4, pp. 725 – 734, April 2009.

Author Profiles

Dr.M.Rajaiah, Currently working as an Dean Academics & HOD in the department of CSE at ASCET (Autonomous), Gudur, Tirupathi(DT).He has published more than 35 papers in Web of Science,Scopus,UGC Journals.

Mr.A.Venkateswarlu, Currently working as an Assistant professer in the department of CSE at ASCET Autonomous),Gudur, Tirupati(DT).

Mr.T.Sai Sudeep, B.Tech student in the department of CSE at Audisankara College of Engineering and Technology, Gudur. He has pursuing in computer science and engineering.

Mr.S.Sumanth,B.Tech student in the department of CSE at Audisankara College of Engineering and Technology, Gudur. He has pursuing in computer science and engineering.

Mr.Sk.Wahed Ali,B.Tech student in the department of CSE at Audisankara College of Engineering and Technology, Gudur. He has pursuing in computer science and engineering.

Mr.T.Harish,B.Tech student in the department of CSE at Audisankara College of Engineering and Technology, Gudur. He has pursuing in computer science and engineering.