A Collaborative Filtering Recommender System for Github

Surbhi Sharma
Shri Mata Vaishno Devi University, Katra-182320, India

Anuj Mahajan
Shri Mata Vaishno Devi University, Katra-182320, India

Abstract – GitHub, a collaborative source code hosting site is based on Git, version control system. GitHub allows users to fork a repository on which one may be interested in and then one can commit changes via a pull request to the maintainer of that repository. GitHub offers REST APIs to perform data analytics. There are a huge number of repositories on the GitHub and one may find it difficult to choose a repository to contribute to. In this paper, we are proposing a Collaborative Filtering based recommender system for GitHub which may provide recommendations regarding which repositories are relevant to a user. Precise recommendations on targeted repositories may save time browsing the relevant repositories but is restricted to the limitations of the collaborative filtering approach. Collaborative Filtering Approach takes into consideration the preferences of similar users. Various correlation measures can be used to find the similarity between users. We have restricted our experimentation in this paper to only Pearson Correlation Coefficient and Cosine Similarity to find and cross-check similar users and then we have applied a Similarity Threshold to filter off un-similar users.

Index Terms – GitHub, Repositories, Collaborative Filtering, Pearson Correlation Coefficient, Cosine Similarity, R language, Similarity Threshold, Normalization.

1. INTRODUCTION

GitHub being a source code hosting site and the collaborative platform has gained a lot of popularity in the recent years. It has a vast amount of data regarding 'Users' and 'Repositories'. Whenever anybody wants to contribute on GitHub he/she may get confused to which repository they should contribute out of enormous repositories which may lead to a decrease in the possible number of contributors. So, to increase the number of contributors on GitHub it is essential to provide filtered information to users regarding which repositories to contribute to. Recommender systems can be helpful to resolve this ambiguity among contributors by providing the precisely filtered information. Recommender systems these days play a very vital role in every field. Even online shopping websites like Flipkart, Snapdeal etc. recommend new items to its users based on their previous history i.e. items which users have earlier searched or ordered. Even social networking sites like Facebook and Flipkart highly use the concept of recommender systems by suggesting the mutual friends, focused advertisements etc. to users. So, this paper aims to save the time of GitHub users by clubbing the concept of recommender systems on GitHub platform.

We are suggesting:

1. The collaborative filtering approach of recommender systems applied on GitHub to provide targeted information of tentative repositories to users based on our assumption that a user may be interested in contributing to a repository if a similar user (similarity threshold along with correlation analysis are used to find similar users) has contributed to that repository.

2. Two correlation measures i.e. Pearson Correlation Coefficient and Cosine Similarity have been applied on GitHub data to determine and cross check similar users on GitHub.

The rest of the paper is organized as follows: Section II is the related work part, Section III describes the proposed method, Section IV explains the experimental evaluation, Section V explains the Comparative study of results, Section VI is the Visualization of Proposed Recommender System ,Section VII is Conclusion Part and Section VIII is Future Scope of this Paper.

2. RELATED WORK

Recommender systems play an important role in all fields like e-commerce etc. to provide filtered information to the users. It uses various data mining and information filtering techniques to provide targeted information to users rather than exploring whole information and thus saves a lot of time [1].It solves the information overload problem. Recommender systems are further divided into 2 categories-Collaborative Filtering (CF) and Content-based filtering. Collaborative filtering is widely used approach because it is domain free and it is independent of prior description of the item. It builds a model from past experience of a user and based on the preferences of similar users recommendations are provided to the new user [2]. Collaborative filtering further consists of two categories: Memory based CF and Model based CF. Memory based CF employs entire dataset to make predictions and then uses various neighborhood methods to determine similar users [3].

Various commercial sites like Amazon etc. also use memory based CF. It is further divided into the user-oriented and item-oriented approach. In user-based CF, similar users are determined and then based on similar users; suggestions are provided to the targeted user [4]. In item-based collaborative filtering, instead of determining the similar users, here similarity is calculated between items which test user have rated and items which are not yet rated by the active user. So, in item-based collaborative filtering, the profile of item is taken into consideration. Based on the similar items recommendations are provided to target user [5] [6]. Various correlation measures exist to determine the similarity between different users. Pearson Correlation Coefficient, Vector Cosine Similarity, Euclidean Distance Similarity, Spearman Correlation Similarity and Tanimoto Coefficient Similarity are few of the correlation measures [7]. Pearson Correlation Coefficient calculates the similarity in the range of -1 to 1 whereas Cosine Similarity calculates the value in the range of 0 to 1. Euclidean Distance Similarity computes Euclidean Distance between two item's preferences. The similarity is greater if the distance between vectors is shorter and vice versa. Tanimoto Coefficient Similarity is generally used for sparse datasets and it denotes the ratio of intersection for different datasets. Spearman Correlation Coefficient takes into consideration rank of ratings and strength of the relationship between two variables can also be measured using this coefficient. Model based approach firstly develops a model of user ratings and based on it provides item recommendations to users. The probabilistic approach is used in this model building process. Bayesian networks, clustering and rule based approaches are few of the machine learning algorithms used for the model building process. Bayesian networks employ probabilistic approach for collaborative filtering. Rule based approach is based on association rules to determine the association between purchased items. The concept of classification is used to classify purchased and non-purchased items [8]. These authors have discussed recommender system based on content-based filtering. Content-based filtering (CS) approach provides recommendations to users based on the content of items and user's preferences, whereas Collaborative filtering recommends items based on the correlation between people with similar preferences. So in other words, Content based filtering approach recommends items similar to those which user have liked in the past as opposed to Collaborative filtering approach that identifies users with similar preferences as of test user and recommends items they have liked[9]. Different components used in Content-based Recommender systems are Content Analyzer, Profile Learner, and Filtering Component. Content Analyzer Component converts the data coming from different information sources to form data which is suitable for further processing steps. This processed data will be given as input to other components. Profile Learner Component constructs the user profile using various machine learning algorithms based on the user preferences and Filtering Component finally recommends the relevant items to users on the basis of the user profile. Few of the flaws of Content based approach are-

(a) As this approach is dependent on item description so if less information is available then content-based approach cannot provide accurate recommendations.

(b) Another drawback is new user problem i.e. new user should rate few items before it is being recommended.

(c) Serendipity problem also exists as it always provides expected outcomes but it never recommends something interesting which user have never rated. [10]

Due to these shortcomings of this approach, Collaborative filtering approach is considered to be more widely used approach in providing recommendations as Collaborative filtering have following advantages-

(a) Collaborative Filtering can be applied in domains where less information is available about the content of items as it does not depend on the profile of items.

(b) Easy to use.

(c) High performance.

(d) More Scalable

(e) Faster Approach and High Accuracy

(f) More Robust

3. PROPOSED METHODOLOGY

In this section, description of proposed method is discussed. Firstly Collaborative Filtering approach is explained in detail and then different steps of proposed approach are discussed.

Since collaborative filtering approach is widely used and preferred over other approaches of recommender system hence we have applied collaborative filtering to GitHub Data. Collaborative filtering (CF) offers suggestions /recommendations to users based on other users having similar tastes. It takes into account users' feedback in the form of ratings and then based on that similar users are determined using various correlation measures.

Considering an example of Collaborative Filtering (CF) in Table 1-

Suppose user 1 has earlier bought Item 1 and Item 3 and we have to predict the rating for Item 2 of the same user that whether the same user has interest in buying Item 2 or not then using this approach, first task is to determine users who are similar to user 1 based on correlation measures like Pearson Correlation Coefficient (PCC), Cosine Similarity etc. Secondly, we will get the user who is most similar to user 1, as its clear in below table that user 3 is most similar to user 1 then if User 3 has bought item 2 then this approach will recommend

user 1 to buy item 2. This is the basic concept of Collaborative Filtering i.e. recommending items based on the likelihood of other similar users.

Table 1- Example of Collaborative Filtering

Users	Item 1	Item 2	Item 3	Item 4
User1	5	??	4	-
User2	4	5	-	4
User3	5	5	4	3
User4	4	-	-	4

3.1. Steps of Proposed Approach based on Collaborative Filtering-

Step 1-First step is Data Acquisition from GitHub. GitHub's data may be downloaded using a crawler and later refined to do analytics as per the requirements. Few researchers [11] have also tried to collect and analyze GitHub data and they have given an opportunity to other researchers to use their downloaded GitHub data who further want to do any kind of analysis regarding GitHub. So, we have used that GitHub data as the base of our work. They have provided data in the form of two files: One is of Users and other is of Repositories. Users and Repositories have various attributes which are not required for analysis, so authors have provided only those attributes which are of use for further analysis.

Step 2- In 2nd step, we have merged both the files of GitHub data i.e. Users and Repositories dataset. Table 2 illustrates the attributes of Repositories dataset and Table 3 describes the attributes of Users dataset.

Table 2- Attributes of 'Repository' Dataset

Fork, watchers, language, created-at, updated-at, private, full-name, login, owner.id, organization.login, organization.id, forks, id, year

Table 3- Attributes of 'Users' Dataset

Company, longitude, latitude, hireable, followers, location, following, login, type, id

As login is one of the common attributes in both datasets. So, on the basis of login attribute, we have merged both datasets using R language. So, on the completion of this step, we have merged data set with information of millions of users and repositories.

Step 3- In 3rd step, we have determined different users from the merged dataset. As for collaborative filtering, one of the attributes which are of interest to us is Login attribute. Login attribute contains information of user name. So, in this step, we have determined different users present in the merged dataset

Step 4- In 4th step, we have grouped together the information of each user. At the end of this step, contributions of each user are clearly depicted.

Step 5-In 5th step, we have fetched the information of 'login' and 'language' attributes because Collaborative filtering provides recommendations on the basis of similar users so login attribute which illustrates user name is fetched and language attribute here illustrates the language used in a repository. So the outcome of this step describes that user is contributing to a repository which uses that particular language.

Step 6- In 6th step, we have determined the count of repositories of each user on the basis of language used.

Step 7- In 7th step, normalization is performed to normalize the values of the count in the range of 0 to 1 which in turn now indicates the proportion of each language used by a particular user.

Step 8-In 8th step, Data frame is created by all users and all languages where columns consist of all users and rows consist of total languages. As each user has used only a few languages out of all so for rest of the rows 0 values will be displayed. In this way, we have detailed information of all users and the proportion of the corresponding language used by them.

Step 9-In the last step, finally we have applied correlation measures to determine the similarity between different users. First one is Pearson Correlation Coefficient (PCC). PCC calculates the similarity between attributes in the range of -1 to 1. The value of -1 indicates negative correlation i.e. attributes are negatively correlated to each other. The value of 0 indicates no correlation exists. The value of +1 indicates positive correlation [12]. Pearson Correlation Coefficient is calculated by following equation.

$$Sim\ {}^{PCC}_{X,Y} = \frac{\Sigma XY - \frac{(\Sigma X)(\Sigma Y)}{n}}{\sqrt{\left(\Sigma X^2 - \frac{(\Sigma X)^2}{n}\right)(\Sigma Y^2 - \frac{(\Sigma Y)^2}{n})}} \quad (1)$$

Sim [X, Y] - Similarity between users X and Y.

n- Number of values.

After PCC, one more correlation is applied to determine similarity i.e. Cosine Similarity just for the sake of cross checking. In this similarity measure, two users/items are considered as two vectors in m- dimensional space. The similarity between them is measured as the cosine of the angle between two vectors. It is generally used to determine the similarity between two documents and then gradually used in collaborative filtering to determine the similarity between two items/users rather than documents. It is generally used in

positive space so it calculates the similarity in the value of 0 to 1[13].

Cosine Similarity between two Users X and Y is calculated in below Equation -

$$Sim\ {}^{Cosine}_{X,Y} = \frac{\Sigma XY}{\sqrt{\Sigma X^2}\sqrt{\Sigma Y^2}} \quad (2)$$

4. EXPERIMENTAL EVALUATION

In this section, the experimental evaluation of our proposed method is discussed.

4.1. Results of Similarity between Users Based on Pearson Correlation Coefficient (PCC)-

In this section, we have applied Pearson Correlation Coefficient to determine the similarity between different users. Fig 1 is a screenshot of values after applying PCC.As PCC calculates the value in the range of -1 to 1, that's why here few values are negative also.

USERS	USER-1	USER-2	USER-3	USER-4	USER-5	USER-6	USER-7
USER-1	1	0.527275272	0.815467842	-0.01781035	-0.017763911	0.373948029	-0.01590561
USER-2	0.527275272	1	0.350887161	0.237524921	0.626055552	0.557977849	0.563271717
USER-3	0.815467842	0.350887161	1	-0.010836965	-0.010808708	-0.010840629	-0.009677998
USER-4	-0.01781035	0.237524921	-0.010836965	1	0.11934953	-0.015389817	-0.013739297
USER-5	-0.017763911	0.626055552	-0.010808708	0.11934953	1	-0.015349689	0.891177956
USER-6	0.373948029	0.557977849	-0.010840629	-0.015389817	-0.015349689	1	-0.013743943
USER-7	-0.01590561	0.563271717	-0.009677998	-0.013739297	0.891177956	-0.013743943	1
USER-8	-0.012545659	-0.016260755	-0.007633588	-0.010836965	-0.010808708	-0.010840629	-0.009677998
USER-9	0.402839001	0.345690684	0.157268506	0.098382056	0.17313907	0.514158356	0.08801672
USER-10	0.014231876	0.023773105	-0.008664789	-0.012300902	-0.012268828	0.053269476	-3.29E-06
USER-11	-0.01745932	0.283938059	-0.010623376	0.582141857	0.143405094	-0.015086495	-0.013468505
USER-12	-0.012545659	0.04514527	-0.007633588	-0.010836965	-0.010808708	-0.010840629	-0.009677998
USER-13	-0.015241246	0.069452245	-0.009273757	0.160618108	0.032974795	0.218851404	-0.011757434
USER-14	-0.015241246	0.577044275	-0.009273757	-0.013165419	0.911759916	-0.01316987	0.997030402
USER-15	-0.020618557	0.224869164	-0.012545659	0.276060424	0.060201893	-0.017816372	0.115315672
USER-16	-0.012545659	-0.016260755	-0.007633588	-0.010836965	-0.010808708	-0.010840629	-0.009677998
USER-17	0.863801972	0.673397918	0.70440273	-0.015384615	-0.015344501	0.661423907	-0.013739297
USER-18	-0.012545659	0.597799502	-0.007633588	-0.010836965	0.94083322	-0.010840629	0.948443847
USER-19	-0.016879974	0.529202002	-0.010270864	0.305879262	0.83949602	-0.014585885	0.84683269
USER-20	-0.012545659	0.597799502	-0.007633588	-0.010836965	0.94083322	-0.010840629	0.948443847

Fig.1. Similarity between users based on PCC

4.2. Results of Similarity between Users Based on Cosine Similarity-

In this section, we have applied Cosine Similarity to determine the similarity between different users. Fig 2 is a screenshot of values after applying Cosine Similarity. As Cosine Similarity calculates value in the range of 0 to 1, that's why here we have only positive values.

5. COMPARISON OF RESULTS BASED ON PCC AND COSINE SIMILARITY

In this section, we have plotted the above calculated values in the form of a graph.

Firstly, we have plotted the similarity between User 1 and other 40 Users based on PCC. Graph 1 shows the plotted values based on PCC between User 1 and other 40 Users. PCC calculates the value in the range of -1 to 1. In this

USERS	USER-1	USER-2	USER-3	USER-4	USER-5	USER-6	USER-7
USER-1	1	0.539117522	0.816496581	0	0	0.384836029	0
USER-2	0.539117522	1	0.359580009	0.254261463	0.633294537	0.566913517	0.570540812
USER-3	0.816496581	0.359580009	1	0	0	0	0
USER-4	0	0.254261463	0	1	0.132658446	0	0
USER-5	0	0.633294537	0	0.132658446	1	0	0.892576301
USER-6	0.384836029	0.566913517	0	0	0	1	0
USER-7	0	0.570540812	0	0	0.892576301	0	1
USER-8	0	0	0	0	0	0	0
USER-9	0.419645677	0.370583844	0.171399649	0.121197854	0.193689469	0.524711149	0.10855448
USER-10	0.028045647	0.041314885	0	0	0	0.064757853	0.010862032
USER-11	0	0.299188652	0	0.588348405	0.156098771	0	0
USER-12	0	0.060140282	0	0	0	0	0
USER-13	0	0.087210962	0	0.171498585	0.045501472	0.228969438	0
USER-14	1	0.539117522	0.816496581	0	0	0.384836029	0
USER-15	0.539117522	1	0.359580009	0.254261463	0.633294537	0.566913517	0.570540812
USER-16	0.816496581	0.359580009	1	0	0	0	0
USER-17	0	0.254261463	0	1	0.132658446	0	0
USER-18	0	0.633294537	0	0.132658446	1	0	0.892576301
USER-19	0.384836029	0.566913517	0	0	0	1	0
USER-20	0	0.570540812	0	0	0.892576301	0	1

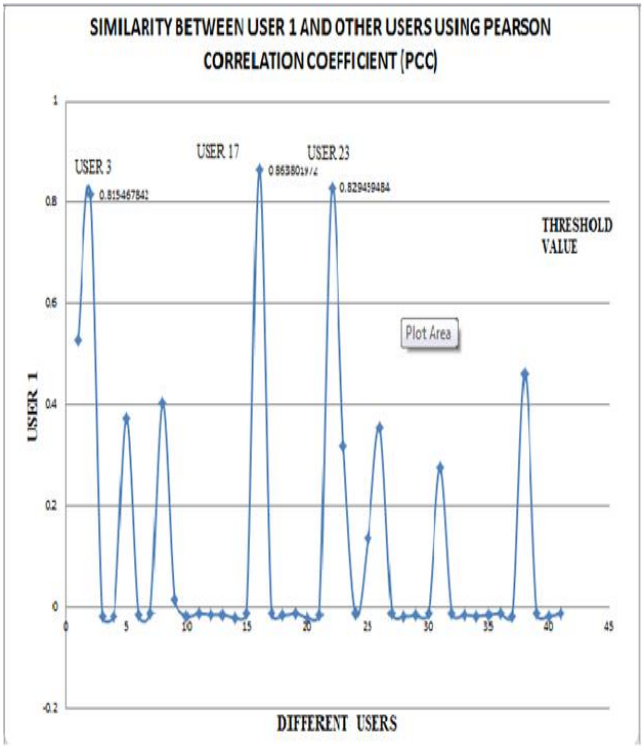
Fig. 2. Similarity between users based on Cosine Similarity.

graph, few users i.e. User 4, User 5, User 7 and User 8 have negative correlation values. Negative values indicate that they are not correlated with each other. In this paper, only positive values are taken into consideration because positive values indicate similar users and this is the main objective. Numerous users might have positive correlative value as data is vast and out of various positive values it will be difficult to determine users who are most similar. To increase the accuracy of recommender systems in determining the most similar users, we have a used a threshold measure for similarity named 'Similarity Threshold' with a default value of 0.8. Only the users having Correlation values above 0.8 are considered as most similar users with User-1. Ultimately, (User 3, User 17 and User 23) are most similar to User 1 because (User1 –User 3) similarity value is 0.8154, (User 1-User 17) similarity value is 0.86380 and (User 1-User 23) similarity value is 0.8294.

Secondly, we have plotted the similarity between User 1 and other 40 Users based on Cosine Similarity. Graph 2 shows the plotted values based on Cosine Similarity between User 1 and

other 40 Users. In this graph all users have positive correlation value because 'Cosine Similarity' calculates the value in the range of 0 to 1. '0' value indicates no correlation. As, (User 4, User 5, User 7 and User 8) are not similar to User-1 that is why they have value '0' based on Cosine Similarity. As 'Similarity Threshold' is 0.8, again User 3, User 17 and User 23 are most similar to User 1 based on the values of Cosine Similarity. Here, (User1-User 3) similarity value is 0.81649, (User 1-User 17) similarity value is 0.8660254 and (User 1-User 23) similarity value is 0.833201.

Once, we have determined that User 3, User 17 and User 23 are most similar to User 1 among 40 Users, then the preferences of these users can be recommended to User 1 i.e. Repositories where User 3, User 17 and User 23 are contributing can be recommended for contribution to User 1.



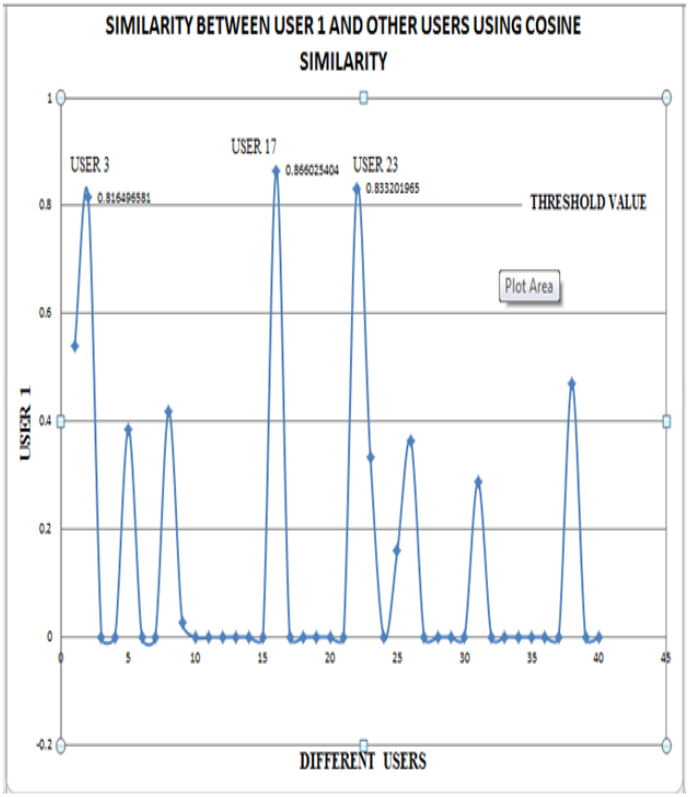
Graph. 1. Similarity values plotted between User 1 and other 40 Users based on PCC

6. VISUALIZATION OF PROPOSED RECOMMENDER SYSTEM

Fig 3. is a representation of how the proposed recommender system when incorporated in GitHub may suggest the contributors to which repositories they may contribute.

7. CONCLUSION

This paper has discussed GitHub which is a collaborative platform for developers to share knowledge and work. This paper has also discussed the need of recommender system for GitHub to suggest contributors to which repositories they should contribute on GitHub to resolve the ambiguity of contributors. Recommender Systems play a vital role now-a-days in every field.



Graph. 2. Similarity values plotted between User 1 and other 40 Users based on Cosine Similarity

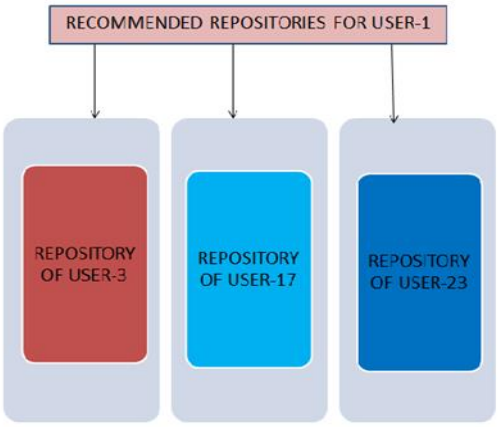


Fig.3. Representation of Proposed Recommender System for Github.

Out of all the approaches of recommender systems, we have used Collaborative Filtering to provide suggestions to GitHub users. To determine the similarity between users, we

experimented with two correlation measures, Pearson Correlation Coefficient (PCC) and Cosine Similarity, and both gave similar results. Based on the preferences of similar users, the repositories of similar users can then be recommended to a user for contribution on GitHub. Using a recommender system with Collaborative Filtering Approach for GitHub may make GitHub more user-friendly and shall save time while searching for repositories. Moreover, providing filtered recommendations while enhancing the user experience may also lead to an increase in contributors on GitHub.

8. FUTURE WORK

As in this paper, we have applied Collaborative Filtering Approach of a Recommender system on Github. In future, we shall make a comparative study of the results of all the recommender system techniques applied on Github.

REFERENCES

[1] Lü, Linyuan, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. "Recommender systems." *Physics Reports* 519, no. 1 (2012): 1-49.
[2] Aberger, Christopher R. "Recommender: An Analysis of Collaborative Filtering Techniques."
[3] Gong, SongJie, HongWu Ye, and HengSong Tan. "Combining memory-based and model-based collaborative filtering in a recommender system." In *Circuits, Communications and Systems, 2009. PACCS'09. Pacific-Asia Conference on*, pp. 690-693. IEEE, 2009.
[4] Zhang, Ruisheng, Qi-dong Liu, and Jia-Xuan Wei. "Collaborative Filtering for Recommender Systems." In Advanced Cloud and Big Data (CBD), 2014 Second International Conference on, pp. 301-308. IEEE, 2014.
[5] Tapucu, Dilek, Seda Kasap, and Fatih Tekbacak. "Performance comparison of combined collaborative filtering algorithms for recommender systems." In *Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual*, pp. 284-289. IEEE, 2012.
[6] Wang, Jun, Arjen P. De Vries, and Marcel JT Reinders. "Unifying user-based and item-based collaborative filtering approaches by similarity fusion." In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 501-508. ACM, 2006.
[7] Su, Xiaoyuan, and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques." *Advances in artificial intelligence* 2009 (2009): 4
[8] Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl. "Item-based collaborative filtering recommendation algorithms." In *Proceedings of the 10th international conference on World Wide Web*, pp. 285-295. ACM, 2001
[9] Van Meteren, Robin, and Maarten Van Someren. "Using content-based filtering for the recommendation." In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, pp. 47-56. 2000.
[10] Lops, Pasquale, Marco De Gemmis, and Giovanni Semeraro. "Content-based recommender systems: State of the art and trends." In Recommender systems handbook, pp. 73-105. Springer US, 2011.
[11] Chatziasimidis, Fragkiskos, and Ioannis Stamelos. "Data collection and analysis of GitHub repositories and users." In Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on, pp. 1-6. IEEE, 2015.
[12] Bamnote, G. R., and S. S. Agrawal. "Evaluating and Implementing Collaborative Filtering Systems Using Apache Mahout." In Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on, pp. 858-862. IEEE, 2015
[13] Breese, John S., David Heckerman, and Carl Kadie "Empirical analysis of predictive algorithms for collaborative filtering." In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 43-52. Morgan Kaufmann Publishers Inc., 1998.

Authors

Surbhi Sharma was a Scholar in the Department of Computer Science & Engineering at Shri Mata Vaishno Devi University, Katra

Anuj Mahajan is working as Assistant Professor in the Department of Computer Science & Engineering at Shri Mata Vaishno Devi University, Katra.