JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Cinematic Synergy: Optimizing Movie Recommendations through Collaborative Filtering

Manju Arora*[1] *Department of Information Technology
Jagan Institute of Management Studies,

Rohini, Delhi, India

***Harsh Chauhan*[2]**
Department of Information Technology
Jagan Nath University,
Bahadurgarh, Haryana, India

***Shivam Dimri*[3]**
Department of Information Technology
Jagan Nath University,
Bahadurgarh, Haryana, India

Abstract

This research investigates the enhancement of movie recommendations using collaborative filtering, emphasizing the application of user preferences for improved accuracy. "Cinematic Synergy" analyzes existing recommendation systems, critiques their methodologies, and thoroughly examines the functionalities of collaborative filtering algorithms. By assessing its performance in the context of movie recommendations, this study identifies challenges and proposes innovative solutions to refine the user experience. The findings of this research offer valuable insights for developers and industry professionals, aiming to elevate collaborative filtering for more effective and user-centric movie recommendations. This project introduces a novel feature of suggesting movies to users based on their selected genres. Additionally, the project displays trending popular movies below the recommendations provided, facilitating easy access for users to explore the world of movies.

Keywords- *Clustering, collaborative filtering, Vectorization, Cosine similarity recommendation*

Introduction

Today, personalized movie recommendations have become an indispensable aspect of the dynamic digital entertainment landscape. This study explores the potential of collaborative filtering, a technique that harnesses collective user preferences to optimize movie suggestions. With the vast array of cinematic content now available, the challenge lies in delivering tailored recommendations that align with individual tastes.

"Cinematic Synergy" aims to delve into the intricacies of collaborative filtering, examining its potential to transform the movie recommendation experience. This introduction acknowledges the increasing relevance of recommendation systems and highlights the unique advantages of collaborative filtering.

This research investigates collaborative filtering algorithms to assess their capacity for capturing diverse user preferences. By addressing challenges such as scalability and diversity, this study proposes innovative strategies to

refine and enhance collaborative filtering, ensuring recommendations cater to individual tastes and contribute to a well-rounded cinematic experience.

In summary, "Cinematic Synergy" aims to contribute valuable insights to the discourse on movie recommendation systems, with a particular focus on collaborative filtering. By exploring its complexities and proposing practical solutions, this research seeks to usher in a new era of refined, accurate, and user-friendly movie recommendations, enhancing the cinematic experience for audiences worldwide.

Literature Review

Recommender systems have their roots in the early 1990s with the introduction of the first system, Tapestry, which was an e-commerce recommender. Tapestry was established by Nisha Sharma and Mala Dutta, and the term "recommender system" was coined by computer-based librarian, Grundy, in 1979. Since then, various recommendation systems utilizing different technologies have been developed. Today, there is a wide range of recommendation systems available in different fields.

According to Sang-Min Choi, et. al. [1], collaborative filtering, a popular approach in recommendation systems, has its limitations, such as the sparsity problem or the cold-start problem. To address this issue, the authors proposed a solution that leverages category information. They introduced a movie recommendation system that takes into account correlations within genres. The authors stated that category information is present for new content, allowing the recommendation system to suggest even recently released movies without relying on a significant number of ratings or views. This approach remains unbiased towards both highly rated, extensively viewed content and lesser-known new content.

George Lekakos and colleagues proposed a movie recommendation system that combines both Content-Based Filtering and Collaborative Filtering to overcome their individual limitations and to create a solution suitable for various scenarios [2]. In their system, called "MoRe," they address the problem of users assigning the same rating to movies by excluding such users from the formula. To implement content-based recommendations, they use cosine similarity to consider factors such as movie writers, cast, directors, producers, and genre. They have also developed a hybrid approach with two variations, "substitute" and "switching," which use collaborative filtering to generate recommendations and content-based filtering when specific criteria are met. The primary focus of their approach is on collaborative filtering.

Debashis Das and colleagues [3] examined various types of recommendation systems and provided a comprehensive overview. Their paper served as a survey on recommendation systems, covering both personalized and non-personalized approaches. The authors delved into explanations of user-based and item-based collaborative filtering, offering a clear illustration. Additionally, they highlighted the advantages and disadvantages associated with different recommendation systems.

Jiang Zhang and colleagues [4] introduced a collaborative filtering method for movie recommendations, dubbing their approach 'Weighted KM-Slope-VU.' The authors divided users into clusters of similar individuals using K-means clustering, then identified a virtual opinion leader for each cluster, representing all users within that particular cluster. Instead of handling the entire user-item rating matrix, the authors focused on a more compact virtual opinion leader-item matrix. They applied their unique algorithm to process this smaller matrix, significantly reducing the time required to generate recommendations.

S. Rajarajeswari and colleagues [5] explored the Simple Recommender System, Content-based Recommender System, Collaborative Filtering-based Recommender System, and proposed a solution incorporating a Hybrid Recommendation System. Their approach incorporates cosine similarity and Singular Value Decomposition (SVD). Their system generates 30 movie recommendations using cosine similarity, then filters these movies based on SVD and user ratings. The system considers only the most recently watched movie by the user, as the authors have presented a solution that takes only a single movie as input.

Muyeed Ahmed, et. al. [6] proposed a solution using K-means clustering algorithm. The authors separated similar users into clusters. Subsequently, the authors developed a neural network dedicated to recommendation purposes for each cluster. The proposed system encompasses stages, such as Data Preprocessing, Principal Component Analysis, Clustering, Data Preprocessing specific to the Neural Network, and construction of the Neural Network. User ratings, user preferences, and user consumption ratios were considered. Following the clustering phase, the authors employed

a neural network to predict the ratings that users might assign to movies that they have not yet watched. Finally, recommendations were made with the help of the predicted high ratings.
Gaurav Arora, et. al. [7] have proposed a solution of movie recommendation which is based on users' similarity. The research paper is general in the sense that the authors did not mention the internal working details. In the Methodology section, the authors discussed City Block Distance and Euclidean Distance without including details about cosine similarity or other techniques. The authors stated that the recommendation system Vol-7 Issue-4 2021 IJARIIE-ISSN(O)-2395-4396 14954 www.ijariie.com 635 is based on hybrid approach using context based filtering and collaborative filtering but neither they have stated about the parameters used, They have not provided information regarding the internal workings.
V. Subramaniyaswamy, et. al. [8]have prefer a answer of customised movie proposal which uses collaborative filtering technique. The Euclidean distance metric is employed to identify the most similar users. The user with the smallest Euclidean distance value was determined. Finally, the movie recommendations are based on what particular user has the best rating. The authors even claimed that the recommendations vary with time, so that the system performs better with the changing taste of the user over time.
Harper, et. al. [9] mentioned the details about the Movie Lens Dataset in their research paper This dataset is extensively utilized, particularly for movie recommendation purposes. Various versions of the dataset are accessible, including the Movie Lens 100 K, 1M, 10M, 20M, 25M, and 1 B datasets. The dataset encompasses features such as the user ID, item ID or movie ID, rating, timestamp, movie title, IMDb URL, and release date. along with movie genre information.
According to R. Lavanya, et. Al. [10], in sequence to gear the detail explosion problem, proposal structure are helpful. The authors addressed challenges such as data sparsity, the cold start problem, and scalability. They conducted a comprehensive literature review of nearly 15 research papers related to movie recommendation systems. In this review, the authors noted a predominant preference for collaborative filtering over content-based filtering among researchers. Additionally, they observed the widespread adoption of hybrid approaches in the reviewed papers. Despite the considerable research conducted in the field of recommendation systems, the authors identified ongoing opportunities to address existing limitations and further enhance the system performance.
Ms. Neeharika Immaneni, et. al. [11] prefer a hybrid proposal method which takes focus on both content-based filtering approach and collaborative filtering approach in a hierarchical manner to provide personalized movie recommendations to users. A distinctive aspect of this research lies in the authors' innovative approach to making movie recommendations through a well-sequenced set of images that effectively portray the movie's storyline. This unique method enhances the visual experience of the users. This study delves into various recommendation system approaches, including graph-based recommendations, content-based methods, hybrid recommender systems, collaborative filtering systems, and genre correlation-based recommender systems. The proposed algorithm comprises four key phases. Initially, user interests were gauged using social networking websites such as Facebook. Subsequently, movie reviews were analyzed, leading to the generation of recommendations. Finally, a story plot was created to enhance the visual appeal of recommendations.
Md. Akter Hossain, et. al. [12] proposed NERS which is an acronym for neural engine-based recommender system The authors have meticulously performed a successful integration of two datasets. Furthermore, they asserted that the superior performance of their system compared to existing ones can be attributed to the incorporation of both a general dataset and a behavior-based dataset in their approach. To assess their system against existing ones, the authors employed three distinct estimators. Recommendation systems employ diverse techniques, including collaborative, content-based, and hybrid filtering. Content-Based Filtering tailor recommendations based on a user's past preferences. It overcomes the new user problem, but may lack diversity. Hybrid techniques blend both methods to address issues, such as cold start, data sparsity, and scalability.
Introduced by Goldberg et al. (1991), collaborative filtering has undergone a significant evolution. While early systems like Tapestry had limitations, contemporary applications, such as Grouplens on platforms like Amazon and Moviefinder, showcase its widespread use. In today's data-rich landscape, collaborative filtering is essential for efficiently delivering pertinent information to the vast amount of data available.
This technique, which is renowned for recommending items, hinges on establishing user similarity through their ratings in a user-item matrix. The collaborative filtering process involves identifying similar users, predicting items based on their choices, and generating desired results.
Tianqi Zhou et al. implemented an item- based collaborative filtering recommendation algorithm using the Hadoop programming model and the Movielens-10M dataset.
Collaborative Filtering is further categorized into memory-based and model-based methods, each of which offers distinct advantages and applications.

Algorithm

Input: Given a set of movies (m)
Output: the goal is to determine the optimal number of clusters (K).

Step 1: Choose a subset of movies (n) from the total collection, where n is less than m.
Step 2: If n is greater than 20, select the top 20 movies based on ratings from the subset; otherwise, display the output movies sorted by rating.
Step 3: In the case of equal ratings for movies x and y ($R_x = R_y$), prioritize those with a higher number of user votes.
Step 4: Set the assumed number of clusters, K, to 4.
Step 5: Iteratively perform the following steps (6 and 7):
Step 6: Select initial centroids C1, C2, C3, and C4.
Step 7: Calculate the Euclidean distance of all data points with respect to C1, C2, C3, and C4, then recompute the centroid of each cluster.
Step 8: Repeat steps 6 and 7 until the centroids no longer change.

TF-IDF

Inverse document frequency (IDF) assesses the prevalence or rarity of a word within a collection of documents. The IDF calculation involves the term (word) of interest, denoted as 'n,' and the total number of documents (N) in the corpus (M). The denominator corresponds to the count of documents where the term 'n' is present.

$$idf(n, M) = \log\left(\frac{N}{count(d \in D : t \in d}\right) \quad \text{--- equation (1)}$$

Note: There is a possibility that a term may not be present in the corpus, leading to a potential divide-by-zero error. To address this, one approach is to augment the existing count by 1, resulting in a denominator of (1 + count).

Scikit-Learn

- $IDF(n) = log\ \frac{1+t}{1+df(n)} + 1$ --- equation (2)

Standard Notation

- $IDF(n) = log\ \frac{t}{df(n)}$ --- equation (3)

The purpose of employing IDF is to address the influence of common words such as "of," "as," "the," etc., which are prevalent in an English corpus. Through inverse document frequency, we aim to reduce the significance of frequently occurring terms while amplifying the impact of less common terms. Additionally, IDFs can be derived from either a background corpus to mitigate sampling bias, or from the specific dataset used in the experiment.

Putting it together: TF-IDF

To summarize, the key intuition motivating TF-IDF is that the importance of a term is inversely related to its frequency across documents gives us information on how often a term visible in a paper, and IDF gives us details about the wonder of a expression in the group of paper By multiplying these values, we can obtain the final TF-IDF value.
The higher the TF-IDF score, the more important or relevant the term becomes, and its TF-IDF score will approach 0.

Cosine similarity

Co-sine similarity is a benefit bound by a control range which varies of 0 and 1. The closer the value is to 0, the more orthogonal or perpendicular the two vectors are to each other. When the value is closer to one, the angle is smaller, and the images are more similar.
As the cosine similarity measurement approaches 1, the angle between vectors A and B is smaller. The images below depict this more clearly.

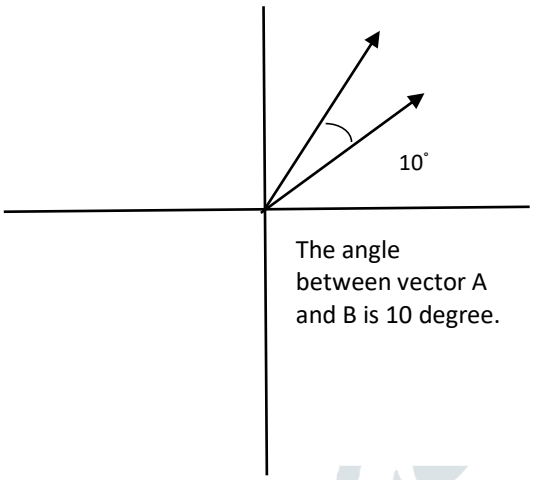

Figure 1 The angle between vector A and vector B

Document Similarity

When there is a need to evaluate the similarity between pairs of documents, employing cosine similarity is a suitable method for gauging the likeness between two entities. To measure document similarity, it is crucial to convert words or phrases in a document or sentence into vectorized representations. Utilizing these vectorized representations in the cosine similarity formula allows quantification of the level of similarity. In this scenario, a cosine similarity of 1 implies identical documents, while a cosine similarity of 0 indicates no similarities between these two documents.

Dataset

This study leverages a comprehensive dataset extracted from The Movie Database (TMDb) to conduct an in-depth analysis of various aspects related to movies. The dataset encompasses key information, including movie ID, title, Overview, genres, keywords, cast, and crew. To enhance data consistency, load data functions from a previous kernel have been employed, and specific adjustments have been made to address variations in fields such as the runtime across the different versions of the dataset.

Data Source Transfer Details

The transfer of data involves extracting essential movie-related details, with a focus on maintaining accuracy and relevance. Notably, the dataset now incorporates separate files containing full credits for both the cast and the crew. This separation enhances the granularity of the dataset and provides a more nuanced understanding of movie-related information.

Data Preprocessing Steps

The preparation of the dataset involved a series of preprocessing steps to refine and structure the information for analysis. These steps include converting JSON strings to lists, extracting relevant cast and crew members, and cleaning and restructuring the data fields. The creation of a 'tags' column consolidates information from the movie overview, genres, keywords, cast, and crew, providing a unified representation of each movie's attributes.

Experimentation and Results

Here is some graphical representation of the data:

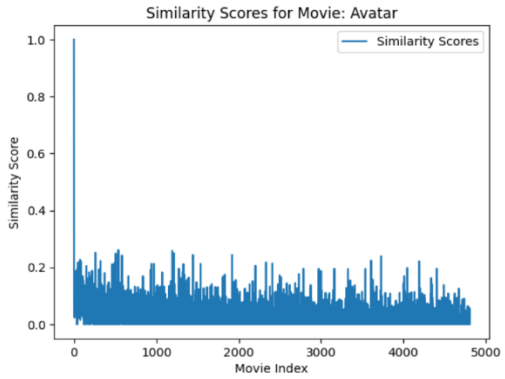

Figure 2 Lineplot between similarity scores and movie index

This is a lineplot showing the similarity scores for the specified movie ('Avatar' in this case) with respect to other movies in the dataset.

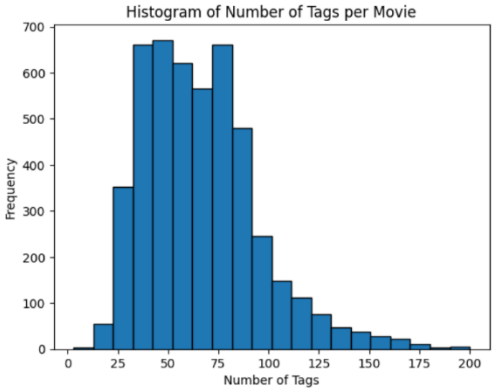

Figure 3 Histplot between no of tags and frequency

This code calculates the number of tags for each movie and creates a histogram to visualize the distribution of the number of tags.

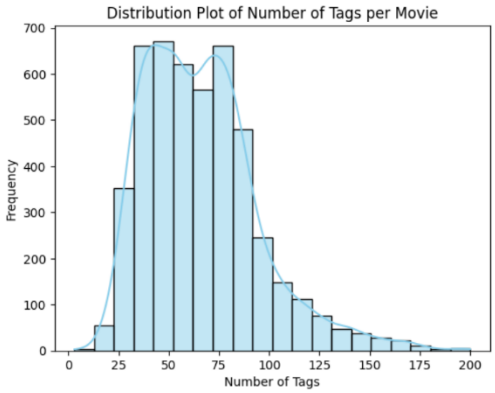

Figure 4 Histplot between no of tags and frequency

In this Distplot, tags_count is calculated as the number of tags for each movie by counting the words in the 'tags' column.

Selected Movie Recommendations

Upon selecting a specific movie from the dropdown menu and clicking the "Show Recommendation" button, the system provides a set of movie recommendations based on the similarity scores. The recommendations include the movie titles and corresponding posters, allowing users to explore related films.

Genre-Based Movie Recommendations

Users have the option to refine their recommendations by selecting specific genres using the multi-select feature. The system then filters the recommendations based on both the selected movie and genres, enhancing the personalization of the suggested movies.

Popular Movie Recommendations

The system also includes a section showcasing popular movies. These recommendations were fetched directly from the TMDb API and displayed with their titles and posters.

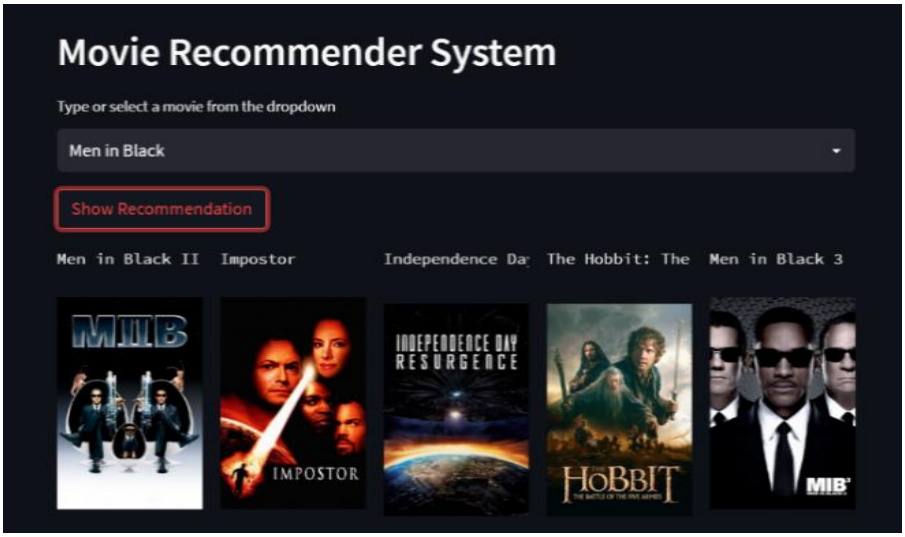

Figure 5 Searching a Movie

This image shows recommended movies related to the movie searched by the user. Cinematic Synergy shows movie recommendations according to the choices made by the user.

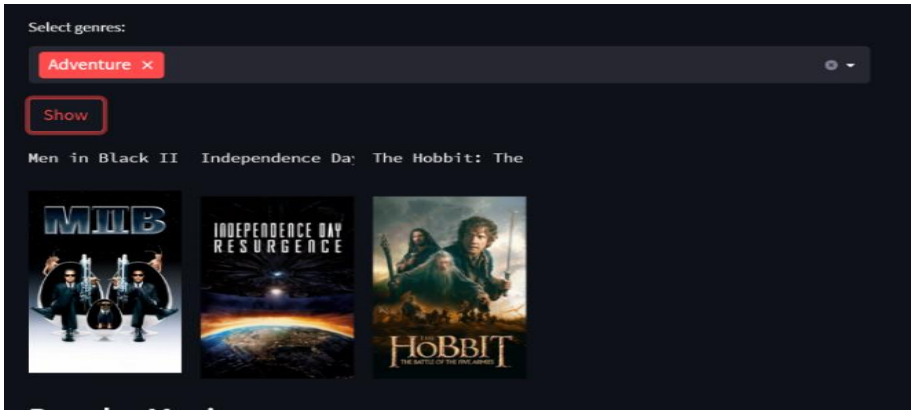

Figure 6 Selecting Genres

This image depicts the movie recommendation by Cinematic Synergy according to the genre selected by the user. In the above image, the user has choices of various genres, and when the user selects the genre of his preference, the Cinematic Synergy recommender shows the results of the movies having that genre.

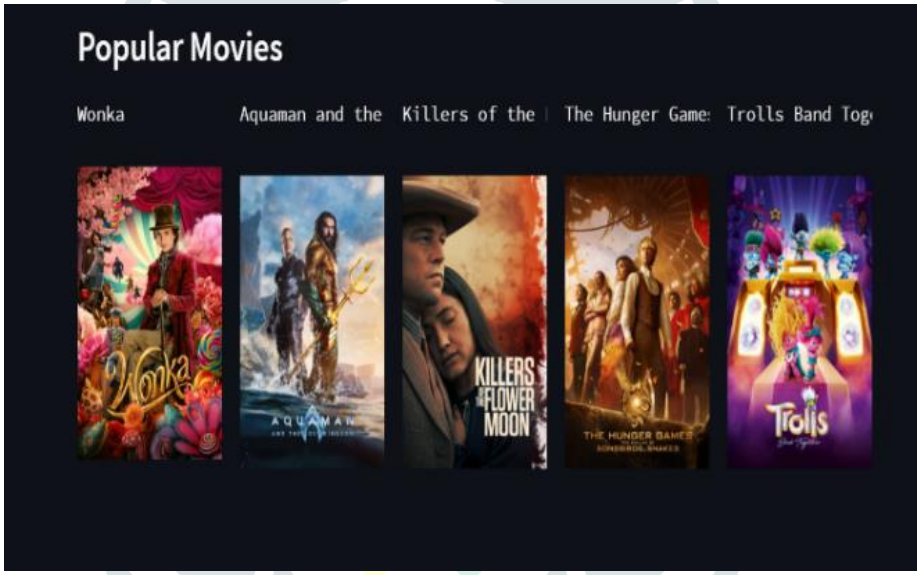

Figure 7 Showing Popular Movies

This image shows popular movies to the users for ease. Herein, all trending movies are updated on the cover page and shown to the users so that they can get to know what is worth watching and what is not. Popular movies were filtered according to their popularity, user ratings in the dataset, and year of release. This makes it easier for users to obtain suggestions for trending movies in the outer world.

```
array([[1.        , 0.08964215, 0.06071767, ..., 0.02519763, 0.0277885 ,
        0.        ],
       [0.08964215, 1.        , 0.06350006, ..., 0.02635231, 0.        ,
        0.        ],
       [0.06071767, 0.06350006, 1.        , ..., 0.02677398, 0.        ,
        0.        ],
       ...,
       [0.02519763, 0.02635231, 0.02677398, ..., 1.        , 0.07352146,
        0.04774099],
       [0.0277885 , 0.        , 0.        , ..., 0.07352146, 1.        ,
        0.05264981],
       [0.        , 0.        , 0.        , ..., 0.04774099, 0.05264981,
        1.        ]])
```

Figure 8 Similarity matrix after using Cosine Similarity which tells the similarity scores of all movies
This is a numerical representation of movie tags using CountVectorizer and then computes the cosine similarity between movies, providing a measure of similarity based on their tag descriptions. This similarity matrix can be used for content-based recommendation systems where movies with higher cosine similarity are considered more alike in terms of their tags.

Conclusion

This project Cinematic Synergy: Optimizing Movie Recommendations through Collaborative Filtering demonstrates the creation of a movie recommendation system with a user-friendly interface using Streamlit. The system leverages a content-based approach, genres, keywords, cast, and crew to provide personalized recommendations. The integration of the TMDb API enhances the application by fetching real-time data and movie posters. Users can input their favorite movie, receive tailored suggestions, and even filter recommendations based on preferred genres. The system also offers a glimpse of currently popular movies. Overall, this project combines data processing, machine learning, and web development to deliver an engaging and dynamic movie recommendation experience.

References

[1] Han J., Kamber M., “Data Mining: Concepts andTechniques”, Morgan Kaufmann (Elsevier), 2006.
[2] Ricci and F. Del Missier, “Supporting Travel Decision making Through Personalized Recommendation,” Design Personalized User Experience for e-commerce,pp. 221-251, 2004.
[3] Steinbach M., P Tan, Kumar V., “Introduction to Data Mining.” Pearson, 2007.
[4] Jha N K, Kumar M, Kumar A, Gupta V K “Customer classification in retail marketing by datamining” International Journal of Scientific & Engineering Research, Volume 5, Issue 4, April-2014 ISSN 2229-5518 International Journal of Computer Applications (0975 – 8887) Volume 124 – No.3, August 201511
[5] Giles C.L., Bollacker K.D., and Lawrence S., “CiteSeer: An automatic citation indexing system,” in Proceedings of the third ACM conference on Digital libraries, 1998, pp. 89–98.
[6] Beel J., Langer S., Genzmehr M., and Nürnberger A., “Introducing Docear’s ResearchPaperRecommender System,” in Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL’13), 2013, pp. 459–460.
[7] Bethard S and Jurafsky D, “Who should I cite: learning literature search models fromcitation behavior,” in Proceedings of the 19th ACM international conferenceon Information and knowledge management, 2010, pp. 609–618.
[8] Bollacker K. D., Lawrence S., and Giles C. L., “CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications,” inProceedings of the 2nd international conference on Autonomous agents, 1998, pp. 116–123.
[9] Erosheva E., Fienberg S., and Lafferty J., “Mixedmembership models of scientific publications,” in Proceedings of the National Academy of Sciences of the United States of America, 2004, vol. 101, no. Suppl 1, pp. 5220–5227.

[10] Ferrara F., Pudota N., and Tasso C., "A KeyphraseBased Paper Recommender System," in Proceedings of the IRCDL'11, 2011, pp. 14–25.
[11] Jiang Y., Jia A., Feng Y., and Zhao D., "Recommending academic papers via users' reading purposes," in Proceedings of the sixth ACM conference on Recommender systems, 2012, pp. 241–244.
[12] McNee S. M., Kapoor N., and Konstan J. A., "Don't look stupid: avoiding pitfalls when recommending research papers," in Proceedings of the 20th anniversary conference on Computer supported cooperative work, 2006, pp. 171–180.
[13] Middleton S. E., De Roure D. C., and Shadbolt N. R.,"Capturing knowledge of user preferences: ontologies in recommender systems," in Proceedings of the 1st international conference on Knowledge capture, 2001, pp. 100–107.
[14] Zarrinkalam F. and Kahani M., "SemCiR - A citation recommendation system based on a novel semantic distance measure," Program: electronic library and information systems, vol. 47, no. 1, pp. 92–112, 2013.
[15] Schafer J. B., Frankowski D., Herlocker J., and Sen S., "Collaborative filtering recommender systems," Lecture Notes In Computer Science, vol. 4321, p. 291, 2007.
[16] Seroussi Y., "Utilising user texts to improve recommendations," User Modeling, Adaptation, and Personalization, pp. 403–406, 2010.
[17] Buttler D., "A short survey of document structure similarity algorithms," in Proceedings of the 5th International Conference on Internet Computing, 2004.
[18] Goldberg D., Nichols D., Oki B. M., and Terry D., "[Using collaborative filtering to weave an information Tapestry]," Communications of the ACM, vol. 35, no. 12, pp. 61–70, 1992.
[19] Beel J., Langer S., and Genzmehr M., "Mind-Map based User Modelling and Research Paper Recommendations," in work in progress, 2014.
[20] MacQueen J.. Some methods for classification and analysis of multivariate observations. In Proc. Of the 5th Berkeley Symp. On Mathematical Statistics and Probability, pages 281-297. University of California Press, 1967.
[21] Ball G. and Hall D.. A Clustering Technique for Summarizing Multivariate Data. Behavior Science, 12:153-155, March 1967. Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems.