International Journal of Advanced Science and Technology
Vol. 29, No. 5, (2020), pp. 4852 - 4865

Design of Recommender System using Content Based Filtering and Collaborative Filtering Technique: A Comparative Study

Indu Hariyale[1*], M.M. Raghuwanshi[2]
[1]Research Scholar, Yashwantrao Chavan College of Engineering, Nagpur, India
[2] Professor, Computer Engineering, G. H. Raisoni College of Engineering and Management, Pune, India
[1]induhrl@gmail.com, *[2]m_raghuwanshi@rediffmail.com*

Abstract

Recommendation based systems have gained a lot of popularity due to their wide range of applicability. From e-commerce-based product recommendation, to social media-based friend recommendation, these systems can be used for any kind of pattern analysis targeted to recommending data based on interlinked usage statistics. To refine the aspect of such systems, they must have a strong pattern recognition engine, combined with a strong prediction engine. Because, a strong pattern recognition engine will be able to analyze and distinguish different patterns effectively, and the prediction engine will be able to merge these patterns together in order to predict the recommendation for the system. Generally, algorithms like neural network, k-means, kNN and SVM based pattern analyzers are combined with neighborhood-based, context-aware pattern analysis-based and collaborative filtering-based predictors in order to develop a complete recommendation system. Many authors have also combined recommender systems in order to generate a high-performance hybrid recommender. In this paper we have studied different recommender systems and observed their statistical performance under various recommendation conditions; present the design of recommender system using two cutting edge method content based and collaborative filtering. These systems are then ranked as per their performance parameters in order to recommend the best recommender system for a given application under test. Furthermore, recommendations are so as to additionally enhance efficacy of these systems.

Keywords: *Recommender, kNN, neighborhood, pattern, prediction, collaborative, context, content.*

1. Introduction

Recommending items of interest based on usage patterns is tremendously driving social and ecommerce website revenues. This is possible due to the presence of highly interconnected social and ecommerce information that is provided by users free-of-cost via their personal logins on these websites. All this information is represented in the form of interconnected data graphs, which tend to extend from few thousand connections to more than 10 million connections, based on the application under study. But processing such complex data graphs requires implementation of algorithms which can; not only analyze the data patterns but also predict next data patterns with utmost accuracy. Recommender systems use the following steps while performing recommendation tasks for any system-under-test,

A. Cross-domain data acquisition and pre-processing

This is one of the crucial steps in recommendation. In this step, information from various sources is gathered, and pre-refined to remove any duplicates, missing values or redundancies. The data collection process has to be done accurately, because based on the collected data, the system will be able to predict patterns and finally recommend items which are either most-frequently used, or items which need most attention (like a product on an ecommerce website which is not moving due to people tweeting incorrectly about it). Moreover, this step is also decided by the application under test, and decides the overall performance of the system.

B. Data linking and processing for recommendation

Data collected from different sources needs linking with the help of certain keys. For example, social media data about users can be linked with the user's buying patterns from the ecommerce data via the user's unique ID. This process of linking fuses the data from different sources into a single dataset, and makes it easier for processing.

C. Pattern analysis and classification

The fused data is given to a pattern analyser, wherein the data is either clustered into different groups, and similarity between data patterns is evaluated, or the data is used for classification of any new input entry. In either case, patterns obtained from the input dataset are used for developing a trained engine which is used for recommendation.

D. Recommendation based on classified data

The trained engine developed during the pattern analysis phase is given a set of inputs. These inputs are processed by the engine, and an output recommendation is obtained. This recommendation is often found in terms of most probable product which you might buy on ecommerce websites, or most probable user which can be your friend on social media. The output of this step decides the accuracy and effectiveness of the system under test.

E. Post-processing tasks

Once the recommendations are made, then the system might need re-tuning, or the recommended data might be used as an incremental learning entry for the trained system. These post-processing tasks are evaluated in this step, and are not always required for recommendation systems. Figure 1 describes social relationship based movie recommender.

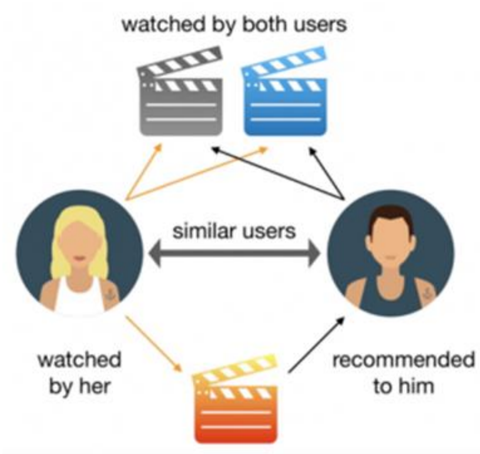


Fig 1.Social relationship-based movie recommender

Based on the given steps, researchers have developed different techniques for recommendation. A sample recommendation system which recommends movies based on social data is shown in the following figure, where in because the users are friends, and their movie patterns match, so the movie seen by one user is recommended to her friend. The next section does a deep-dive into these techniques, and allows for the readers to analyze which techniques can be used for what kind of application. In the later section, we discuss certain performance parameters of these algorithms, followed by some interesting observations.

2. Literature Review

ISSN: 2005-4238 IJAST

Recommendation systems have come a long way, from simple threshold-based recommenders to highly complex deep-learning-based recommenders. Simple recommenders provide one-to-one recommendations, like recommending what to buy and what not to buy based on a particular characteristic of a product. But, as complexity increases, the recommenders are able to predict the users' buying patterns, and recommend things like, what the user can buy based on a combination of previous purchase histories of the user and their friends. Moreover, this system becomes computationally expensive, and the recommendation cost-to-output ratio increases. Various techniques [1] have been mentioned to lessen the cost-to-output ratio. Collaborative Filtering, Content-Based and Hybrid approaches are mostly utilized techniques for recommendation incorporate,. These approaches make sure that the overall recommendation process is effective and that the efficiency of recommendation is improved. Collaborative filtering [1] uses data from different sources. Data can be in the form ratings, or reviews, or any other metric which is useful to describe the item under recommendation. Collaborative filtering approaches include Memory based User to User (MBU2U), Memory based Item to Item (MBI2I), Model Based Clustering (MoC), and Model based Matrix Factorization (MoMF). The MBU2U approach uses user similarity for recommendation, while MBI2I uses item-based similarity in order to recommend entities of interest. Both the models are heavily dependent on strong dataset dependency. If the available datasets have good interconnections, then the algorithm will be able to find the connections of users or items, to showcase strong results. In contrast, the MoC uses clustering approaches in order to make predictions. The generated clustering model will enable data points of similar features to be clubbed together, and thereby assisting in identifying data patterns which are similar to each other. The MoMF model is superior to all the other models, and it utilizes the information provided by all the different models in order to generate a recommendation engine that can predict user behavior, and thereby improving the item recommendation for the user.

Another kind of recommendation process is called as content-based recommendation. It uses either item-based or user profile-based recommendation, and utilizes different kind of similarity metrics like cosine, jaccard, pearson, adjusted cosine, constrained correlation, mean squared differences and silhouette coefficient. All these metrics [1] find the similarity of the user under test with the other users' profiles, and based on this similarity recommendations are made. An in-depth analysis of these techniques is mentioned in [1]. Recommending products on ecommerce websites covers more than 30% of all kinds of recommendations. The study done in [2] takes into consideration the work done by various researchers in the domain of big data-based ecommerce recommendations. There is no statistical analysis done by the authors, but the work gives a brief idea about the concepts used while designing recommenders for big data systems.

Domain specific recommenders provide better accuracy than general purpose recommenders. The system proposed in [3] uses percentage of view as a metric for recommendation of movies to users. Their recommender is based on feedbacks given by users after watching movies. These feedbacks are then accumulated into a percentage of view metric, and finally recommendation is done. Based on their analysis, the Content-based Random Forest, Content-based Linear Regression, Collaborative Filtering-Model-based, User-based Collaborative Filtering along with Item-based Collaborative Filtering approaches are inferior to proposed algorithm after comparison. The error percentage of the proposed algorithm is reduced by 3% when compared to these techniques. While using statistical metrics paves the way for recommendation, some researchers like Bushra Alhijawi [4] and others make the use of semantic information in order to create recommendation systems. The work done in [4] uses collaborative filtering and combines semantic relevancy of items in order to perform recommendation. The system utilizes rating data for a particular item, and combines it with semantic feedback about the item in order to evaluate the rank of the item. Based on this rank, the item is recommended to the users. The obtained results showcase that score normalization works better, and is able to reduce the error by more than 20% when compared to non-normalized techniques like Pearson similarity, Pearson correlation cosine similarity. The proposed technique is also compared with its own non-normalized version, and it is found that normalization improves the accuracy of the proposed technique by 10%.

Another movie recommendation system similar to [3] is proposed in [5]. In the system, the authors have made use of collaborative filtering in order to recommend movies to users. They have used User Neighborhood for user-based recommendation and Log Likelihood Similarity for item-based recommendation in order to develop a hybrid recommender. The paper describes an interesting approach for recommendation, but doesn't provide any statistical analysis for the same. The approach must be revisited before actual implementation. Hybrid recommender systems are the future of recommendation systems, and when combined with machine learning, they further tend to provide highly accurate results. This has been proven by the research done in [6], wherein machine learning is used for recommendation systems. The researchers have evaluated the accuracies of Supervised, Unsupervised, Semi-supervised and Reinforcement learning. Results showcase that reinforcement learning combined with big data can provide high accuracy, and minimum error when compared to other systems.

Having a large dataset does not always guarantee high accuracy. In order to achieve high accuracy, the system developer must be able to identify features which are variant enough that upon using them the system will be able to distinguish between different sets of data effectively, and will be able to classify information correctly. A method to identify such variant information is defined in [7], wherein a collaborative filtering -based recommendation algorithm uses clustering and dimensionality reduction in order to obtain a high accurate recommender. The algorithm uses k-Means combined with singular value decomposition (SVD) which further enhance recommendation accuracy. System is compared with kNN and simple k-Means based system proves that RMSE of the proposed algorithm is reduced by more than 10% when compared to other methods. As previously suggested, that semantic information can be useful for recommendation. The work in [8] uses word2vec, which is a sentiment analysis tool, in order to convert the input reviews into scores, and then based on these scores a clustering algorithm is devised to perform recommendation. The algorithm's error is 20% lower than that of the ICRRS [8] method. It is recommended to further performance evaluation of this approach on different datasets, further contrast it with different algorithms in order to comment on its usability.

Machine learning algorithms can reduce the error of recommendation to a very low level. This can be seen from the work proposed in [9], wherein the researchers have used the Mahout Apache framework in order to implement user preference-based recommendations. The overall error is reduced to less than 10%, and thus the system is able to accurately recommend entities. A survey of such machine learning along with deep learning techniques is given in [10], wherein it is concluded that deep learning and machine learning based recommenders can be useful for social recommendations, ecommerce recommendations, movie recommendations, etc. and can perform at very high accuracies.

A different approach towards recommendation of books in case of a shared-account structure is given in [11]. Here, researchers have used a COVER algorithm for item-based disambiguation. The COVER algorithm improves the performance of top-K rules algorithm, and reduces the error rate using disambiguation. The algorithm is able to improve the accuracy by almost 10%, when compared to top-K rules algorithm, and thus can be used in simple recommendation structures. The algorithm's performance is not evaluated for complex cross-domain problems, and thus that area must be explored by interested readers. Incorporation of COVER with deep learning can be recommended, due to the inherent advantages of deep-learning systems. Some of the advantages are mentioned in [12]. The work in [12] uses a deep belief network (DBN) which enhances recommendation accuracy of Movie Lens dataset. The results are compared with Hybrid Features Selection Algorithm (CHFSA), and it is found that the proposed algorithm reduces the mean absolute error by more than 10%. The algorithm can take into consideration both semantic and non-semantic values, and thus must be used as a recommendation system for any kind of dataset.

Context-aware recommendation systems take into consideration the context of the user under which recommendations is needed. The work done in [13], proposes Declarative Context-Aware Recommender System (D-CARS) which creates user-specific profiles by considering the user's historical usage data. They also propose the use of l User Window Non-Negative Matrix Factorization topic model (UWNMF) for profile generation, and Subspace Ensemble Tree Model (SETM) for analysis of data given by other users. The proposed method is compared with CTT, SVD, kNN, & NMF models, and it is evaluated. The proposed algorithm is able to reduce the error by more than 20%, which is due to its user-specific recommendation capabilities. Moreover, the incremental update phase is added to D-CARS in order to further improve its error performance. The incremental update phase reduces the error by 5% when compared to normal retrained D-CARS method. Use of context awareness with machine learning for education systems is studied in [14], wherein researchers have reviewed more than 40 papers in order to conclude that knowledge-based hybrid context aware systems outperform collaborative filtering, demographic, and domain-based systems. Another review is proposed in [15], wherein researchers have compared the advantages and drawbacks of different recommendation mechanisms. They observe that collaborative filtering is advantageous for recommendation when compared with other methods. They further comment that hybrid collaborative systems outperform other systems in terms of error performance, and must be used for effective recommendations. Similar observations are made in [16], wherein model-based, user-based along with item-based collaborative filtering methods are studied. But here they have mentioned that use of hybrid recommendation systems can further improve the performance of collaborative systems, and must be used in order to improve recommendation accuracy. This is again proven to be true by the work proposed in [17], wherein the researchers have used collaborative filtering along with context-aware user profiles in order to provide highly accurate movie recommendations. A hybrid recommendation system can be seen from figure 2, wherein data from different kinds of sources is combined in order to create a recommendation.

The proposed work in [17] improves the accuracy of recommendation over non-context-aware methods by 10%, which is a good enough value for real-time uses. Hybrid collaborative filtering systems are again compared in [18], wherein it is observed that Recommendation with social trust ensemble, Personal interest, interpersonal interest similarity, and interpersonal influence are fused into a unified personalized recommendation model & subspace clustering solves the problem of data-sparsity, DSMMF and DSTNMF, user and item content information, pLSA, GM, item-based CF improves the recommendation accuracy, while Social Poisson factorization and probabilistic matrix factorization incorporates social network information into recommendation. Different challenges faced by such collaborative systems are mentioned in [19], wherein problems like User Cold-Start problem, Item Cold-Start problem,

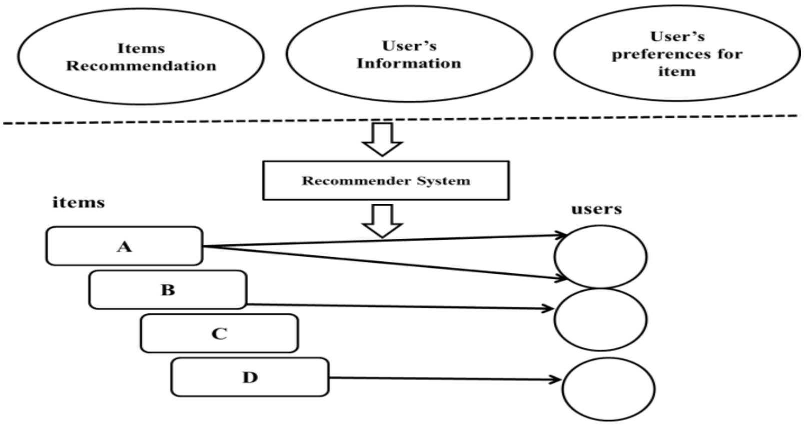
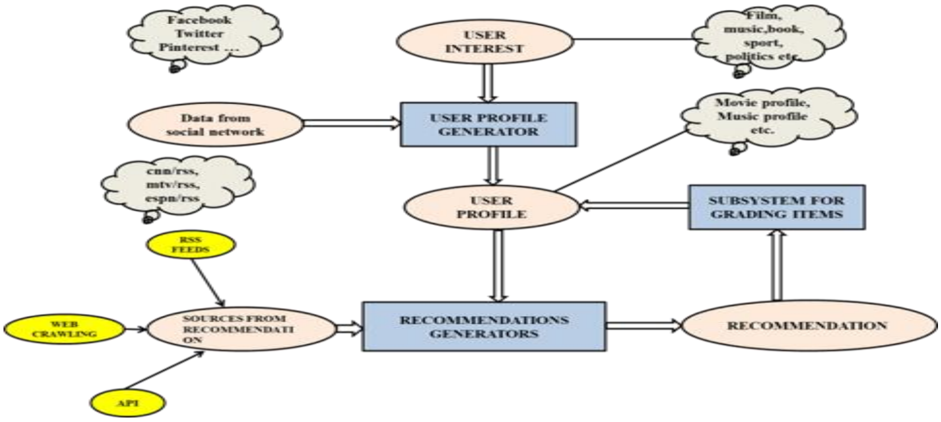


Figure 2. Hybrid recommender

Sparsity, Scalability, Grey Sheep Problem, Ramp-Up Problem, Shared Account Problem, selection of Evaluation Measures and Limited Scope of RSs are discussed in details. A good recommender system must be able to solve these issues in order to reduce the error probability while recommendation.

User and item-based recommendations are again revisited in [20], wherein the researchers have used a hybrid recommendation model for improving the overall accuracy of recommendation. The system is used for movie recommendation, and improves the quality of recommendation by more than 5% when compared to its non-hybrid counterparts. Systems like case-based reasoning are mentioned in [21]. These systems involve using multiple recommenders to develop a user specific recommendation system which has low error with high accuracy of recommendation. The hybrid approaches are then further classified into Mixed hybrid, weighted hybrid, Feature combination hybrid, Switching hybrid, and Feature Augmentations hybrid. Here too, mixture of these hybrid systems forms a multi-hybrid architecture, which is the most accurate system for recommendation. Similar mentioning of techniques is done in [22], wherein different collaborative recommenders are compared. It can be observed that collaborative filtering alone cannot solve issues; it must always be combined with other methods. Cold-start problem of collaborative filtering, work in [23] proposes the use of pre-filled data. This data serves as the initial recommendation data for filtering, which is then iterated over in order to obtain the final filtering results.



Figre 3. High efficiency collaborative filtering

A generalized architecture for filtering data using such a system can be seen in figure 3. Wherein data from different sources is combined together in order to resolve the issues prevalent to collaborative filtering algorithms.

As described previously, use of highly accurate classifiers is the main requirement of a good recommender system. This is proven in [24], where a strong support vector machine (SVM) classifier is incorporated to improve results. This work also uses hybrid approach for recommendation, and thus proves that hybrid systems with machine learning classifiers are paving the way for higher accuracy systems. In order to evaluate any recommender system, we need to have good evaluation metrics. A discussion on these evaluation metrics is proposed in [25], wherein metrics like accuracy, delay, mean-absolute error, and f-measure are discussed. The next section compares the performance of content and collaborative filtering methods based on the experiment executed and some interesting observations about the methods accuracy are made.

3. Recommender Techniques

In collaborative filtering recommendations are based on rating of the user to specific item. The likeness between users based on their inclinations is utilized to prescribe new items for the new user without matching the description of items. If user A like item i and give rating and if user B like item i and give rating which is similar to A, then there is possibility on pattern matching in their profile. CF[2][3][8][26]:Memory-based CF and model-based CF are two categories of Collaborative filtering.

3.1 Memory-based CF technique

Memory based CF works on authentic users rating information which is used further to calculate likeness among users for recommendations, that both user A and B give same rating to item j. CF prescribes all item which are rated by A to user B based This technique performed in 4 steps:

- Find the similarity among the user or item by using the user item matrix
- Find matched items' list or users' list to the active item or user[3] [8]
- Find the neighbor of the active user
- Apply the strategy to join the rating of neighbor user to generate recommendation

User based CF and item based CF are two cultivated types of Memory based CF[2][10][27].

3.1.1. User based CF

User based Collaborative Filtering discovers K alike users as per the items they have purchased. User based CF [2][10][27]technique first figure closeness between the dynamic user and other different users by taking a gander at their ratings on similar items, and identify the neighbor of the dynamic users as expressed by the similarity. In next step it processes the anticipated rating of the items which is given by dynamic user, and then it calculate the average of all the rating given by the users which are similar to the dynamic user.

3.1.2. Item based CF

Item based collaborative filter discovers K identical items on the basis of common users who have bought those items. Item based [10][13]CF calculate predictions by employing the similarity between items. It works in the four steps:

- Construct item similarity model from all item which are rated by dynamic users
- Find the similar item as indicated by the similarity
- Find N matched item and their relating similarity
- Make forecast of unknown ratings on the dynamic items as indicated by the neighbor items

To made prediction it used average of dynamic user rating on matched items.

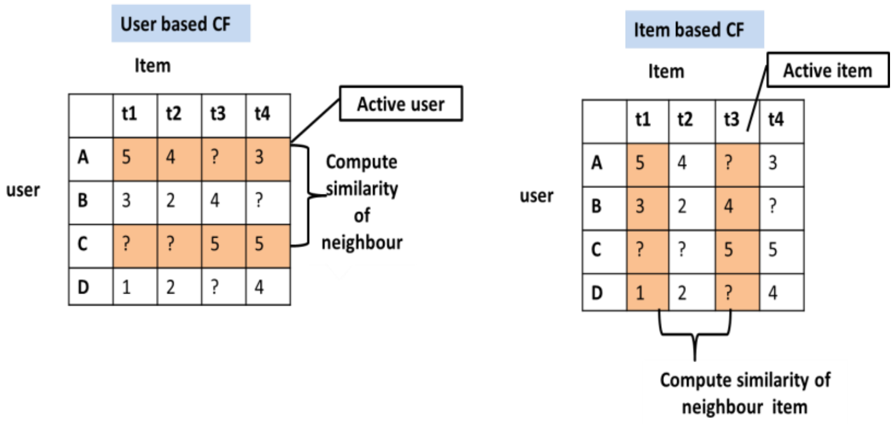


Figure 4. Item based CF and User based CF

In this paper Cosine [4] similarity is used to calculate similarity between users or item.

$$sim(p,q) = cos(p,q) = \frac{\vec{p}.\vec{q}}{\sqrt{|\vec{p}|^2|\vec{q}|^2}} \quad (1)$$

Figure 5 describe the working model of User based CF In pre-processing step normalization of data is performed, in next step similarity is calculated among users or item. Third step find nearest neighbor, and forecast rating for the user, then it rank all item as per their rating , and prescribe item to users. In this paper we implemented user based collaborative filtering with top N recommendation result.

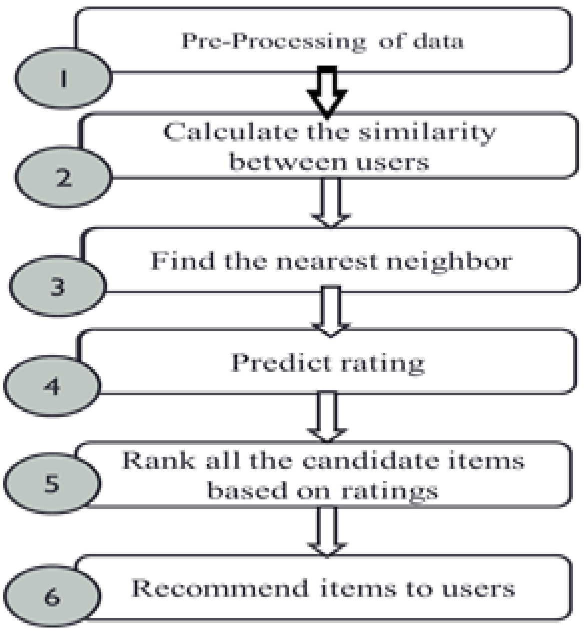


Figure 5. Working model of Memory based CF

3.2. Content based filtering

Content based filtering (CB) uses items' description by taking into account their attributes and qualities for matching of user profiles[4]. It analyzes the group of items that are recently evaluated and construct model for users intrigues which depends on item's features. To compare the new item from the recently preferred item similarity metric is utilized. Further the best matched item is recommended. [4][6]For the most part procedure utilized in CBF model is keyword matching or vector space model and TF/IDF(Term frequency and Inverse document frequency)[27]. In the next section Performance of Text based TF/IDF and [27]Text based kNN[1] is compared with Memory based collaborative filtering.

3.2.1. Text based TF/IDF

Term frequency, quantifies how as often as possible a term presents in a document.

$$\text{TF}=\frac{\text{Number of times term t presents in a docment}}{\text{Total number of terms in the documents}} \qquad (2)$$

IDF, Inverse document frequency which gauges how significant a terms in the document.

$$\text{IDF(x)}=\log\left(\frac{\text{Total number of docments}}{\text{Number of documents with term x in it}}\right) \qquad (3)$$

In Text based TF/IDF technique, a model using the MovieLense data sets. In first step pre-processing and cleaning of data is performed, After pre-processing feature extraction is done using TF/IDF vectorizer which results in feature item matrix. Cosine similarity function is utilized to calculate similar item from feature-item matrix. In next step as per the input given by user top n similar item are ranked and produce the recommended list to the user. Figure 6 describe the working model of Text based TF/IDF.

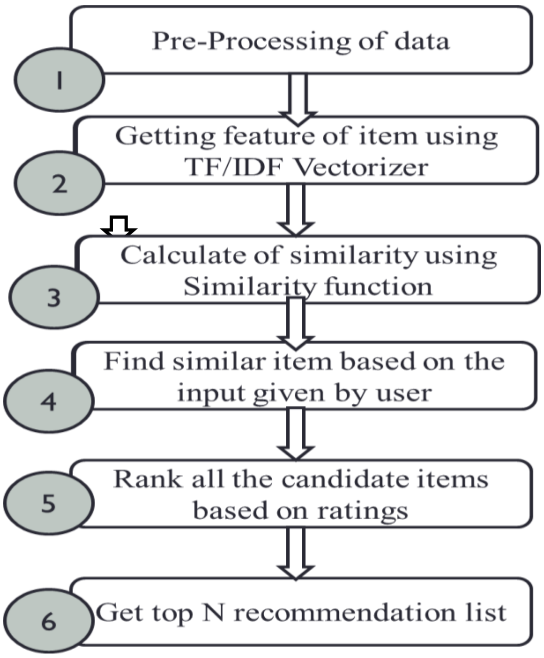


Figure 6. Working model of Text based TF/IDF

3.2.2. Text based kNN

To find nearest neighbor in text based kNN Euclidean distance is computed. Euclidean distance is the radial distance [2] between two observations or records. Two objects are similar when the distance is zero and dissimilar objects have higher distance values.

$$Sim(p,q) = \sqrt{\sum_{j=1}^{n}(p_j - q_j)^2} \tag{4}$$

Here p and q are the two observations with n features. Experiment is performed in amazon and movie lens 100K datasets. This technique first find k similar user who rated item ,then compute mean similarity score, after that predict top K similar item. Model is built on training and test data. Data set is divided into training and test data. Experiments are executed on Text based TF/IDF and Text based kNN and Memory based collaborative filtering. All experiments are implemented in python 3.7. Comparison is done on the basis of RMSE. Recommendation Model trained upon both, first on training data and then on test data.

4. Experimental Study and Results

The outcomes from various recommender frameworks are assessed in this section. RMSE is calculated using formula:

$$RMSE = \sqrt{\frac{1}{M}\sum_{u,i}\left(X_{u,i} - R_{u,i}\right)^2} \tag{5}$$

Xu,i: Calculated rating, Ru,i: Given rating ,M: Count of users in system, Table 1 describes results of various recommendation algorithms.

Table 1. Result of different recommendation algorithms

Algorithm	**Dataset**	**No. of users**	**No. of items**	**No. of rating**	**Sparsity**	**RMSE**
User-User based CF ,Top N Users	Amazon review	5130	1685	37126	0.43%	1.038
User-User based CF, For All users	Amazon review	5130	1685	37126	0.43%	1.12
User based CF	Movie lens 100k	610	9724	100836	6.3%	2.90
Item based CF	Movie lens 100k	610	9724	100836	6.3%	3.40
Top-k User-based CF	Movie lens 100k	610	9724	100836	6.3%	2.55
Top-k Item-based CFs	Movie lens 100k	610	9724	100836	6.3%	2.78
Text based TF/IDF	Sample amazon	-	500	-	7.8%	2.3
Text based KNN	Movie lens 100k	610	9724	100836	6.3%	2.89

For Memory based CF execution is performed on amazon and movie lens data sets. For amazon review data sets:

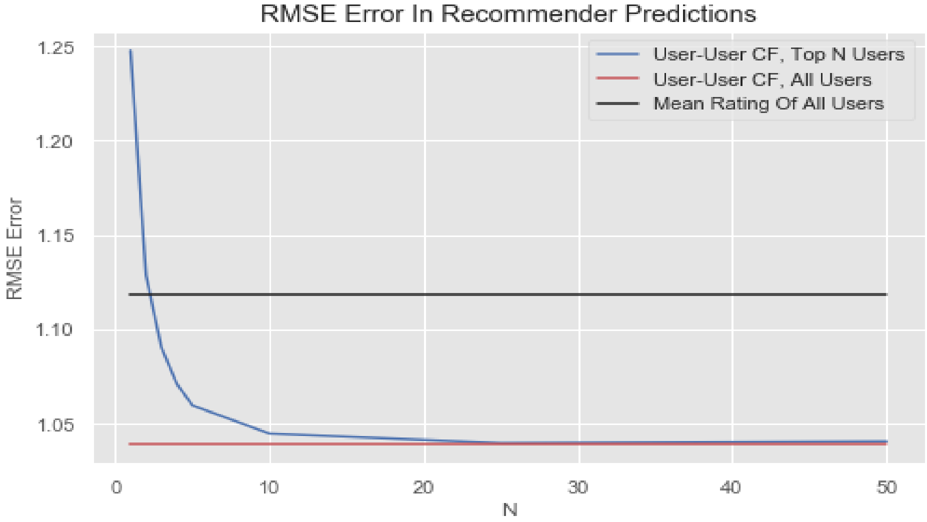


Figure 7. RMSE Value for User-User based CF

Figure 7 describes RMSE result For amazon review datasets for Top N users and All users present in dataset. Figure 8 shows MSE value using kNN classifier for item based CF.

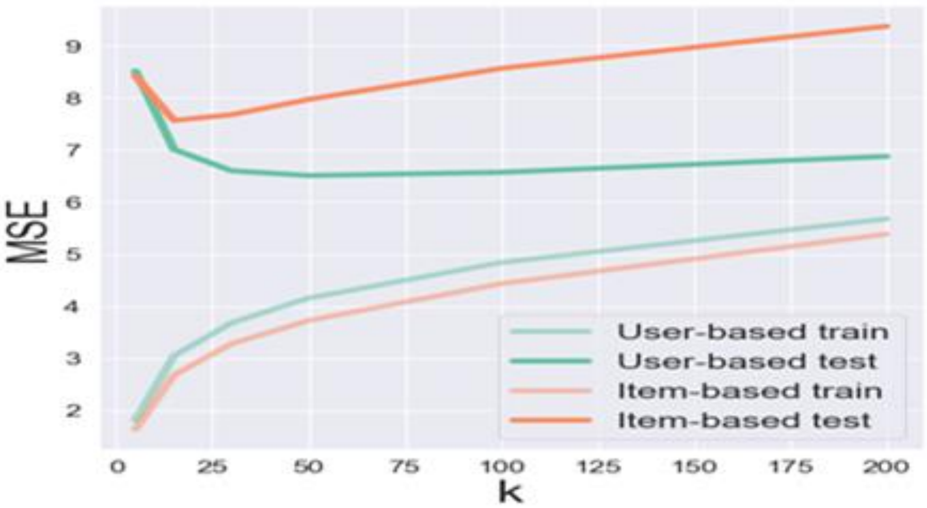


Figure 8. User and item based CF using kNN for Movie lens dataset

Text based kNN

Figure 9 shows results for Text based kNN that produce recommendation for 8 nearest and similar neighbor of the movie iron man. Here Movie lens 100k data set is utilized. Here 8 is threshold value.

```
Results

Results for iron man

You have input movie:iron man
Recommendation system start to make inference
......
Recommendations for {}:iron man
1:Sherlock Holmes (2009) with distance of 26.18682861328125
2:Guardians of the Galaxy (2014) with distance of 26.18205451965332
3:Star Trek (2009) with distance of 26.14861297607422
4:WALL·E (2008) with distance of 25.539186477661133
5:Kung Fu Panda (2008) with distance of 25.22399711608886
6:District 9 (2009) with distance of 25.174392700195312
7:Avatar (2009) with distance of 23.8903751373291
8:Avengers, The (2012) with distance of 22.885585784912111
```

Figure 9. Top k item using Text based kNN where k=8(threshold)

Text based TF/IDF

Text based TF/IDF generate results on the basis of item id. As shown in the result for item id 3 it displays 6 similar items to the item ID 3.The similarity is calculated using Cosine similarity [4].For text based TFIDF item descriptive data sets of 500 samples is utilized for experiment.

```
Results

Results for 3

Recommending 6 products similar to Active sport briefs...
-------
Recommended: Active sport boxer briefs (score:0.4181663992161579)
Recommended: Active boy shorts (score:0.114018481220387 6)
Recommended: Active briefs (score:0.1105372944657289 5)
Recommended: Active briefs (score:0.10917640016582 87)
Recommended: Active mesh bra (score:0.10172320448715239)
Recommended: Barely hipster (score:0.10046865496069653)
```

Figure 10. Recommendation using item ID threshold value=6, item id=3

From the results it is observed that user based and item based algorithm generate better recommendation if data is not sparse. Text based kNN and Text based TF/IDF based on only the description of the data. Both the algorithm suffers Cold start problem and scalability issue. Hybrid techniques. Moreover, machine learning and deep learning algorithms further improve the recommendation accuracy for the system. algorithms like PoV[3] and D-Cars[13] outperform other techniques. Moreover, machine learning and deep-learning algorithms further improve the recommendation accuracy for the system.

5. Conclusion

In order to improve recommendation accuracy of and to get optimum recommender system we would recommend the use of Q-learning and incentive-based learning mechanisms. Moreover, use of blockchain to improve peer-to-peer review based recommendation can be explored by researchers in order to further improve the system performance. Hybrid Deep learning model need to incorporate in content based filtering in order to solve cold start problem.

References

[1] Rahul, Hritik Dahiya, Divyansh Singh, “A Review of Trends and Techniques in Recommender Systems”, 2019 IEEE.

[2] Xuesong Zhao, “A Study on E-commerce Recommender System Based on Big Data”, 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analytics, PP-222-226.

[3] Ramin Ebrahim Nakhli, Hadi Moradi, Mohammad Amin Sadeghi, “Movie Recommender System Based on Percentage of View”, 5th Conference on Knowledge-Based Engineering and Innovation, Iran University of Scienc, e and Technology, Tehran, Iran, 2019 IEEE, PP 656-660.

[4] Bushra Alhijawi, Nadim Obeid, Arafat Awajany, Sara Tedmori, “Improving Collaborative Filtering Recommender Systems Using Semantic Information”, 9th International Conference on Information and Communication Systems (ICICS), 2018 IEEE, PP 127-130.

[5] Ching-Seh (Mike) Wu, “Deepti Garg, Unnathi Bhandary, Movie Recommendation System Using Collaborative Filtering”, 2018 IEEE, PP 11-15.

[6] Agata Nawrocka, Andrzej Kot, Marcin Nawrocki, “Application of machine learning in recommendation systems”, 2018 IEEE, PP 328-331.

[7] Hafed Zarzour, Ziad Al-Sharif , ahmoud Al-Ayyoub2, Yaser Jararweh2, “A New collaborative Filtering Recommendation Algorithm Based on Dimensionality Reduction and Clustering Techniques”, 9th International Conference on Information and Communication Systems (ICICS). 2018 IEEE, PP 102-106.

[8] Pang-Ming Chu and Shie-Jue Lee, “A Novel Recommender System for E-Commerce”, 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI 2017), 2017 IEEE.

[9] Ruchika, Ajay Vikram Singh, Mayank Sharma, “Building an Effective Recommender System Using Machine Learning Based Framework”, International Conference on Infocom Technologies and Unmanned Systems (ICTUS'2017), Dec. 18-20, 2017, ADET, Amity University Dubai, UAE ,IEEE.

[10] Ruihui Mu1, Xiaoqin Zeng1, Lixin Han1, “A Survey of Recommender Systems Based on Deep Learning”, DOI 10.1109/ACCESS.2018.2880197, IEEE Access.

[11] Rizkiyana Prima Putra, Dade Nurjanah, Rita Rismala, “Top-N Recommendation for Shared Account on Book Recommender System”, International Conference on Information Technology Systems and Innovation (ICITSI) Bandung - Padang. October 22-25, 2018 IEEE. PP 60-65.

[12] Brahim Ouhbi, Bouchra Frikh, El moukhtar Zemmouri, “Deep Learning Based Recommender Systems”, 2018 IEEE, PP 161-166.

[13] Rosni Lumbantoruan, Xiangmin Zhou, Yongli Ren, Zhifeng Bao, “D-CARS:A Declarative Context-Aware Recommender System”, IEEE International Conference on Data Mining,DOI 0.1109/ICDM.2018.00151, 2018 IEEE, PP 1152-1157.

[14] Dina Fitria Murad, Yaya Heryadi, Bambang Dwi Wijanarko, “Recommendation System for Smart LMS Using Machine Learning : A Literature Review” , Recommendation System for Smart LMS Using Machine Learning : A Literature Review”, 4th International Conference on Computing, Engineering, and Design (ICCED), DOI 10.1109/ICCED.2018.00031,2018 IEEE, PP 113-118.

[15] Bansari Patel, Palak Desai, Urvi Panchal, “Methods of Recommender System: A Review”, International Conference on Innovations in information Embedded and Communication Systems (ICIIECS), 2017, IEEE.

[16] Farhin Mansur, Vibha Patel, Mihir Patel, “A Review on Recommender Systems”, International Conference on Innovations in information Embedded and Communication Systems (ICIIECS), 2017, IEEE.

[17] “Design and Implementation of Movie Recommender System Based on Graph Database”, 14th Web Information Systems and Applications Conference, DOI 10.1109/WISA.2017.34, 2017 IEEE, PP 132-135.

[18] Rui Chen, Qingyi Hua, Yan-Shuo Chang, Bo Wang, Lei Zhang, and Xiangjie Kong, “A Survey of Collaborative Filtering-Based Recommender Systems: from Traditional Methods to Hybrid

Methods Based on Social Networks", Digital Object Identifier 10.1109/ACCESS.2017. IEEE ACCESS.

[19] Ritu Sharma, Dinesh Gopalani, Yogesh Meena, "Collaborative Filtering–Based Recommender System: Approaches and Research Challenges" 3rd IEEE International Conference on "Computational Intelligence and Communication Technology" (IEEE-CICT 2017), 2017 IEEE PP 1-6.

[20] Elena Shakirova, "Collaborative Filtering for Music Recommender System" 2017 IEEE, 548-550.

[21] Kunal Shah, Akshaykumar Salunke, Saurabh Dongare, Kisandas Antala,"Recommender Systems: An overview of different approaches to recommendations", International Conference on Innovations in information Embedded and Communication Systems (ICIIECS), 2017, IEEE.

[22] NajdtMustafal, AshrafOsman Ibrahim, Ali Ahmed, Afnizanfaizal Abdullah, "Collaborative Filtering: Techniques and Applications", International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), Khartoum,Sudan, 2017, IEEE.

[23] Goran Antolić, Ljiljana Brkić*, "Recommender System Based on the Analysis of Publicly Available Data", MIPRO 2017, May 22- 26, 2017, Opatija, Croatia.

[24] Shreya Agrawal, Pooja Jain, "An Improved Approach for Movie Recommendation System", International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2017).

[25] Kartik Chandra Jena, Sushruta ishra, Soumya Sahoo, Brojo Kishore Mishra, "Principles, Techniques and Evaluation of Recommendation Systems", International Conference on Inventive Systems and Control (ICISC-2017), 2017 IEEE, PP 1-6.

[26] Rui Chen, Qingyi Hua, Yan-Shuo Chang, Bo Wang, Lei Zhang, And Xiangjie Kong, A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks, IEEE Access 2018.

[27] Content-based Recommender Systems: State of the Art and Trends, Pasquale Lops, Marco de Gemmis and Giovanni Semeraro, Recommender Systems Handbook, Springer 2011.