Recommender System: A bibliometric analysis

Sheetal Sharma¹, Kamali Gupta¹ *, Deepali Gupta¹

¹Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India

sheetal.sharma.6@gmail.com , deepali.gupta@chitkara.edu.in

* Corresponding Author: Kamali Gupta, email: kamali.singla@chitkara.edu

Abstract. The exponential growth in the online share of businesses has to lead to a gigantic wave of options available to the active user. Recommender systems, therefore assist the users to go through the tailored list of products to match their preferences. A range of recommender systems is available to serve the purpose. This article will navigate through the basic of recommender systems, and its classifications types viz. collaborative filtering, content-based filtering, demographic, hybrid, and knowledge-based recommender system. It aims to analyze publications of the Scopus database using biblioshiny tool of RStudio software. A bibliometric analysis is conducted on 556 papers to analyze the recent research trends in recommendation systems. Further, challenges have also been discussed that need to be dealt with the recommender system.

Keywords: Recommender system, Collaborative Filtering, Content-based filtering, Hybrid filtering, Analysis.

1. Introduction
With the advent of the Internet, there is a colossal stream of information in the digital forum. To identify and analyze a preferred range of products, it has become a very daunting task for all end-users and e-business players. At this point, the recommender system comes into play. A recommender system is an automated system to filter some entities, that can be products, ads, movies, videos, songs, or peoples from users e-interactions and seeks to foretell the “predilection” a user would extend to an item by using some algorithms or automated techniques [1][2]. Commercial applications in particular use recommender systems. It doesn’t only work for which item to recommend but also in what order the recommendations are being ranked.

The recommender system helps choose similar things, whenever you pick something online. For example, Netflix will advise other movies that one might like to view. Or YouTube will propose different videos or tracks one might wish to view or listen to. Flipkart / Amazon will suggest what types of additional products one might be looking to buy. Similarly, Facebook will recommend a list of additional people that one might want to befriend with. It is a decision support system that assists users, with a list of recommended products or items they can prefer or go for. And on the other side, can increase the profit share of companies by a substantial amount of business. Recommender systems are a productive substitute for search algorithms as they assist active users, discover products/items they might not have located otherwise.

2. Classification of Recommender system
Depending upon the type of filtering technique being used to generate the recommendations, Recommender System can be typically categorized into Collaborative Filtering, Content-Based Filtering, Demographic Filtering[3][4], Knowledge-Based Filtering[5] and hybrid can be considered as the fifth one[6][7] as shown in figure 1.
2.1. Collaborative Filtering (CF) Recommender System

Collaborative Filtering is the most admired recommendation method, which generates the recommendation for the Active User-centred on the users’ records of interactions with the item as well as comparable decisions made by other users for similar items irrespective of the item’s features. It follows a consistent approach for mining items of potential interest (i.e., items not yet rated by the active user and which have been rated by other users) and predicting the rating that the active users would give to them [6][7] figure 2.

A “user-item interaction matrix (table 1)” is used to stock these interactions [8]. An empty row is inserted into the matrix, as soon as a fresh user registers to the site. Likewise, an empty column will be inserted when a neoteric item is added in the catalog. The concept behind is, if two consumers have similarly rated n items, there will be a high possibility that they will rate other items likely [9][10][5]. Therefore, the performance of CF will have a significant influence on the correct selection of
neighboring users, which is done by comparing the pattern of their e-behaviors over time with that of the active user [10].

Table 1: An example of a user-item interaction matrix [9]

| Users/ Movie Series | Harry Potter | Frozen | Jurassic Park | Jumanji | Avengers |
|---------------------|-------------|--------|---------------|---------|----------|
| Hrehaan             | Like        | Dislike| Like          | Like    | Like     |
| Sanskriti           | Like        | Dislike| Like          | Like    | Like     |
| Sharanya            | Like        | Dislike| Like          | Like    | Like     |
| Aarav               | Dislike     | Dislike| ?             | Like    | Like     |
| Manasvi             | Like        | Like   | ?             | Like    | Like     |

Collaborative Filtering generally grounds on explicit ratings, as E-Systems explicitly traces ratings/votes/reviews/likes that users pass to certain items, into a user-item interaction matrix. Individually, all cells of that matrix represent the rating for a specific user-item duo if provided, or can be valued against the rating scale otherwise. Generally, Explicit feedback leads to a Sparse Matrix due to the reason that not all the users tend to provide feedback for all the items they interacted with.

On the contrary, with Implicit rating, the information is extracted from user’s activities, interactions, and e-behaviors (such as browsing history, search patterns, purchases history, songs heard, movies/trailers watched, applications downloaded, web sites visited, and even mouse clicks). Implicit feedbacks are a thickly loaded matrix with feedbacks that is usually present or absence of an event, that can be recorded as Boolean data. [11][12][13].

Collaborative Filtering employs two diverse methodologies represented by figure 4, to autogenerate the recommendations [4][2][6][7].

Memory-based CF employs the entire “user-item interaction matrix” to trace the K-similar users of the Active User. The past ratings about these similar users are put-to-use to forecast the recommendation. Memory-based algorithms can be user-based or item-based as shown in figure 3. It is also known as the neighbor-based CF algorithm. Owing to the absence of any latent model, memory-based CF has a low bias but a high variance [8].

![Memory-based collaborative approaches](image)

Figure 3: Methodologies of Memory-based CF

user-user: From the “user-item interaction matrix” users with most similar profiles are identified (nearest neighbors), extremely popular items among these selected neighbors are then shortlisted to make suggestions. Users are represented based on their interactions with the item, therefore called “user-centered” memory-based CF and it assesses the distance between the users. Steps (1. Find similar users. 2. Compute the ratings. 3. Recommend if not interacted before)

item-item: Filters similar items, for which most of the users have interacted analogously. Items are represented based on how users had interacted with them; this is said to be “item-centered” memory-based CF and it evaluates distance between those items. Steps (1. find similar items rated by the user. 2. Compute the ratings 3. Recommend)

Model-Based CF utilizes a subset of an entire “user-item interaction matrix” and directs a model by employing some machine learning and mining techniques. The model formulated, can absorb to identify intricate patterns constructed on the training data and later can predict intelligently for real-world data [9]. Model-based algorithms train models like singular value decomposition models, Bayesian models,
optimization algorithms, clustering models, and decision trees [7]. Academically has a higher bias but a lower variance [8].

**Figure 4:** Two types of Collaborative Filtering Algorithm

### 2.1.1. Comparison techniques of CF

Collaborative Filtering primarily follows two techniques to facilitate comparison:

1. The K nearest neighbor algorithm (kNN) [11][14][13].
2. Latent factor or matrix factorization algorithms [1][12][14][13].

Neighborhood methods: Emphasis is on relationships between users or, between items to identify neighbors’ set of “like-minded” users/items. The minute, this set has been acknowledged, items decidedly rated by this group are suggested to the current user. Figuring the similarity of users or items is a significant part of the CF recommendation system [13]. The range of similarity will lie between 1 and -1 (1 means similar and -1 means opposite). The kNN algorithm principally employs conventional statistical similarity metrics [11]. Pearson correlation, cosine, adjusted cosine, constrained correlation, and Mean Squared Differences are commonly used metrics.

- Pearson Correlation Similarity: For the user-user based CF algorithm, assuming that the products set rated by users $a$ and $b$ is $P_{(a,b)}$, the similarity $sim(a,b)$ can be computed by Pearson correlation as [12][9][14][15][16][17][18].

$$sim(a,b) = \frac{\sum peP_{(a,b)} (R_{(a,p)} - \bar{R}_a) (R_{(b,p)} - \bar{R}_b)}{\sqrt{\sum peP_{(a,b)} (R_{(a,p)} - \bar{R}_a)^2} \sqrt{\sum peP_{(a,b)} (R_{(b,p)} - \bar{R}_b)^2}}$$

Where,

i. $peP_{(a,b)}$ are summations of the products/items that both the users $a$ and $b$ have rated.
ii. $R_{(a,p)}$ and $R_{(b,p)}$ be the rating given by user $a$ and $b$ on product $p$.
iii. $\bar{R}_a$ and $\bar{R}_b$ is the mean of ratings by the user $a$ and $b$ on all items.

Once the $k$ similar users are identified, prediction can be made as [17]:
prediction(a, p) = \overline{R(a)} + \frac{\sum_{b \in NN} sim(a, b) \times (R_{(b,p)} - \overline{R(b)})}{\sum_{b \in NN} sim(a, b)}

Where,

i. \( NN \) set of nearest neighbors of user \( a \).
   \( R_{(b,p)} \) be the rating of user \( b \) on product \( p \).
ii. \( \overline{R(a)} \) and \( \overline{R(b)} \) is the mean of ratings by the user \( a \) and \( b \) on all items.

For the item-item based CF algorithm, similarity Pearson correlation \( sim(m, n) \) can be calculated as [9][17]

\[ sim(m, n) = \frac{\sum_{u \in U} \sum_{p \in \{m, n\}} (R_{(u,m)} - \overline{R(m)})(R_{(u,n)} - \overline{R(n)})}{\sqrt{\sum_{u \in U} \sum_{p \in \{m, n\}} (R_{(u,m)} - \overline{R(m)})^2 \sum_{u \in U} \sum_{p \in \{m, n\}} (R_{(u,n)} - \overline{R(n)})^2}} \]

Where,

i. the \( u \in U \) summations of users, who rated both the products \( m \) and \( n \).
ii. \( R_{(u,m)} \) and \( R_{(u,n)} \) be the ratings given by users \( u \) to product \( m \) and \( n \) respectively.
iii. \( \overline{R(m)} \) and \( \overline{R(n)} \) is the mean rating of products \( m \) and \( n \) by a set of those users.

Once the \( k \) similar items are identified, prediction can be made as[17]:

\[ prediction(a, m) = \frac{\sum_{b \in NN} sim(m, n) \times (R_{(b,n)})}{\sum_{b \in NN} sim(m, n)} \]

Where,

i. \( NN \) set of nearest neighbors of user \( a \).
   \( R_{(b,p)} \) be the rating given by users \( a \) and \( b \) on product \( p \).
ii. \( \overline{R(a)} \) and \( \overline{R(b)} \) is the mean of ratings by the user \( a \) and \( b \) on all items.

- Cosine-based Similarity: In the Cosine method, two items \( m \) and \( n \) are considered as two vectors in \( x \) dimensional user space “user-item interaction matrix” \((x \times y)\) figure 5. The cosine of the angle between these two vectors is calculated as a similarity. [1][9][19]

\[ Sim_{(m,n)} = \cos(\vec{m}, \vec{n}) = \frac{\vec{m} \cdot \vec{n}}{||\vec{m}|| * ||\vec{n}||} \]

Where,

i. ‘\( \cdot \)’ Denotes dot product of vector \( m \) and \( n \).
ii. If value tending towards 1 directs similarity. Alternatively, a value close to -1 specifies dissimilarity[20].

Correspondingly, Cosine of similarity between two users \( a \) and \( b \) can be calculated [1][18] as:

\[ sim(a, b)^{cos} = \frac{\sum_{p \in P(a,b)} (R_{(a,p)}) \cdot (R_{(b,p)})}{\sqrt{\sum_{p \in P(a,b)} (R_{(a,p)})^2} \sqrt{\sum_{p \in P(a,b)} (R_{(b,p)})^2}} \]

Where,

i. \( p \in P(a,b) \) summations of ratings of products/items that both the users \( a \) and \( b \) have rated.
i. \( R(a, p) \) and \( R(b, p) \) be the rating given by users \( a \) and \( b \) on product \( p \).

ii. '. ' Denotes dot product of vectors.

iii. ' ' Denotes dot product of vectors.

**Figure 5:** Co-rated items by different users [19]

Note: Accuracy of Cosine similarity is not consistent, as practically the rating scale and pattern of different may users vary. Some users lean towards high ratings; whereas some users tend to rate low [9][16][19]. To deal with this downside, adjusted cosine is used.

- Adjusted cosine similarity: The Adjusted cosine fixes the issue in cosine similarity by deducting the respective user mean from each co-rated pair. It has a formula similar to the Pearson correlation. Through Pearson correlation, we may sometimes get negative values because of some type of normalization of the rating pattern of users but it is not there in case of cosine similarity[9][16][19].

\[
sim(a, b) = \frac{\sum_{p \in P(a, b)} (R(a, p) - \bar{R}_a)(R(b, p) - \bar{R}_b)}{\sqrt{\sum_{p \in P(a, b)} (R(a, p) - \bar{R}_a)^2} \sqrt{\sum_{p \in P(a, b)} (R(b, p) - \bar{R}_b)^2}}
\]

- Constrained Pearson Correlation: It is a variant of the Pearson Correlation method; the solitary difference is that Pearson uses mean, while Constrained Pearson employs median for normalization.[18][20]

\[
sim(a, b) = \frac{\sum_{p \in P(a, b)} (R(a, p) - R_m)(R(b, p) - R_m)}{\sqrt{\sum_{p \in P(a, b)} (R(a, p) - R_m)^2} \sqrt{\sum_{p \in P(a, b)} (R(b, p) - R_m)^2}}
\]

Where

i. \( R(a, p) \) and \( R(b, p) \) be the rating given by users \( a \) and \( b \) on product \( p \).

ii. the \( peP(a, b) \) summations of the ratings of products/items that both the users \( a \) and \( b \) have rated.

iii. \( R_m \) represents the median value in the rating scale.

- Mean Squared Difference: Inverse of the average squared difference between the ratings given by \( a \) and \( b \) on the same items is calculated as the similarity between them.[1]

\[
MSD(a, b) = \frac{|P(a, b)|}{\sum_{p \in P(a, b)} (R(a, p) - R(b, p))^2}
\]

Where

i. \( R(a, p) \) and \( R(b, p) \) be the rating given by users \( a \) and \( b \) on product \( p \).
ii. the \( p \epsilon P_{(a,b)} \) summations of ratings given of the products/items that both the users \( a \) and \( b \) have rated.

Benefits[1] [4] [11][7] [18][20]

i. Neighborhood methods are relatively simple and easy to implement and are reasonably accurate.

ii. They are quite popular and widely used.

iii. Able to generate serendipitous suggestions for varied products/people like songs, movies, gadgets, and friends, etc without knowing about the product itself and thus are completely independent of any machine- analyzable content regarding the product.

iv. The neighborhood framework provides a fairly sound explanation for the recommendations it makes, based on the user’s interaction with the product in the past. It recuperates users to interact better and aid fixing wrong impressions and hence enhances the recommendation accuracy to a noticeable extent.

v. Item base neighborhood models can give recommendations even to new users. This is because the similarity between items proves to be much stabler than the similarity between users.

vi. The accuracy of recommendations keeps on improving with every user interaction.

vii. Only Implicit feedback sufficient.

Concerns[1] [4] [11] [21][18][7]

i. Cold Start: Referred to as a state in which a user is quite new to the system and it does not have enough user interaction history to make reliable recommendations. Cold start can be sub-typed into 3 types [11].

   a. New item: Item is new in the Recommender system, thus no rating history. Hence, do not get recommended and remain unnoticed.

   b. New user: Since, a new user has not yet rated any product, so difficult to make valued personalized suggestions.

   c. New community: When the Recommender system itself, is new, its user-item interaction matrix will be empty, leading to almost no history to make recommendation decisions.

Cold start can be addressed with the following strategies[2]:

   a. Random: By suggesting new products to random users or random products to new users.

   b. Maximum expectation: Suggesting new products to keen users or most admired products to new users.

   c. Exploratory: Suggesting a group of varied products to new users or a new product to a group of new users.

   d. Hybrid: Applying hybrid methods of recommendation in the initial stages of a new product or new user.

ii. Sparsity: It is one of the major pitfalls in CF methods. Out of the huge quantity of items sold on any site, only undersized subset items have been rated by active users, resulting in a sparse the user-item interaction matrix. So small numbers of ratings available to evaluate recommendations.

iii. Low scalability: With increasing users and items on the system the size of the user-item interaction matrix grows substantially. High computation power and appropriate algorithms are required to process a huge scale of data in real-time to compute a single recommendation.

iv. Gray Sheep: It's hard to find neighbors of a user with odd tastes.
v. “Rich-get-richer” effect: Popular item gets largely recommended, confining customers’ information area and sometimes cause starvation for the least popular item.

vi. Portfolio effect: It is a situation when the recommendation list consists of items nearly identical, or very analogous to the items already seen by the current user. As a result, user sometime feels trapped within the same kind of products.

Latent factor model: Factorization model like Matrix factorization (alias, SVD), is an alternative method which transforms both items/products and users to a latent factor space. This space attempts to describe ratings by characterizing both products and users on factors automatically inferred from user feedback. Each record in the space vector indicates how meticulously a product/item or a user acquires a specific latent characteristic. For example, when the items are books, the factors might signifies categories, i.e., whether a book belongs to fiction or reality, literature, current events or history, or even other dimensions.[14][13].

Matrix factorization aims to condense the dimensionality of the user-item interaction matrix. Consequently, the factorization algorithm splits the original user rating matrix into two separate matrices, such that when their dot product is calculated, it approaches the original matrix without losing any information and forecasts the absent ratings in the matrix figure 6 [1][14][13][22].

$$R \approx Q \cdot P^T$$

Where

i. R is the actual user-item interaction matrix.

ii. Q is |U|x|F matrix,

iii. P is |I|x|F matrix.

iv. F is the number of factors.

v. ‘.’ Denotes dot product of vectors.

![Figure 6: Matrix Factorization](image)

Now again, similar users or items can be utilized to estimate missing data in the matrix by performing the dot product of only that row and column of the missing cell.[22] figure 7. Each item i is associated with a vector $Q_i \in R^F$ and value at $Q_i$ denotes the level to which item i acquire factors F, positively or negatively. Likewise, each user u is related t vector $P_u \in R^F$ and at $P_u$ signifies, how much user u is appealed by item i possessing factor F both positively or negatively[1][22].
SVD (Singular Value Decomposition): It is a popular approach for recognizing latent semantic factors in information retrieval. SVD decomposes a user-item rating matrix $A(m \times n)$ into 3 matrices as $A = U\Sigma V^T$.

Where

i. $A$ Matrix we want to decompose.
ii. $U$ is the user factor matrix.
iii. $V^T$ is the item factor matrix and $^T$ specifies a particular rated item.
iv. $U$ and $V$ are orthogonal matrices with orthonormal columns such that
   a. $U^T U = I$ (identity matrix)
   b. $V^T V = I$ (identity matrix)
v. $\Sigma$ is the weighted diagonal matrix containing all non-zero singular value entries sorted in decreasing order which specifies the strength of any particular factor/feature and how much dimensions should be reduced.
vi. $U, \Sigma, V$ are unique matrices.

2.2. Content-based Recommender system

The content-based filtering (CBF) method makes suggestions by mining the content information (descriptions) of the items and users. Unlike collaborative filtering, rather than comparing active user’s interaction history with others, it focuses on users’ likings for the attributes of the products, extracted from item profiles [1][4][15][24][25].

Content based filtering involves algorithms to perform following [1][4][24][25]

- Item Profile: A profile is created for each item based on various attributes or keywords, that describes the item. A model is constructed and trained. Also stated as Content Analyzer [1][24] or Feature extraction [25]. Characteristic or features extracted are highly domain-centric. For instance, a movie can have attributes like a thriller, horror, artist, director, romantic, etc. In many systems, a keyword-based matrix is formed[1][25][4].

- User profile: A user-oriented learning model is trained to build a user profile for each user based on the user’s past interactions or ratings for any items that represent training data for the model. This model is executed to suggest the users’ likings for the attributes of the items via their preference for them [25]. Alternatively, the user profile can barely be the listing of items user bought or given a rating to [24]. It is also referred to as Profile Learner [1].

- Filtering recommendations: This component filters the items to be recommended by assessing both the item profile and user profile constructed in previous steps. Users' preferences as modeled in user profiles against items attributes as prototyped in item profiles are taken as a basis to filter a list of items with similar features and are not yet visited by the user but may like to go for it. [1][24][25]. The ensuing list can be binary with like (1) or dislike (0) or continues where products with weighted preferences are ranked consequently[1].
Points favoring content-based recommendation are [7][4]

i. No need to acquire domain knowledge to make recommendations.
ii. Recommendation accuracy improves with time.
iii. Suggestions can be based only on implicit feedback.

Drawbacks[7][4]

i. A major drawback of content-based filtering that the recommendation gets restricted to the features identified by item profile. Whereas, collaborative filtering methods, where a wide variety can be traversed concerning neighbors users and thus can be more accurate.
ii. Cold start problem for new users.
iii. Quality directly proportional to the magnitude of the history dataset.
iv. It also suffers from a portfolio effect.

2.3. Demographic Recommender system

User preferences are categorized concerning the users’ demographic traits such as gender, age, country, career, and education, etc. The idea behind this approach is if an item is liked by a user of attribute class x, then it may be preferred by other users of the same class who has not seen it yet. Thus can be recommended[1][4][25]. The demographics data of the users can be mined through explicit reviews or from implicit users’ past interactions [4].

Benefits[4][7]

i. Recommendations can blend across varied genres.
ii. Prior domain-specific knowledge not necessary.
iii. The quality of suggestions increases with time.

Limitations

i. Demographic information must be gathered.
ii. New users confront the cold start issue.
iii. The volume of the dataset is significantly important in prediction correctness.
iv. Gray Sheep problem.

2.4. Knowledge-based Recommender system

A KBRS maps the users' explicit requirements with the solutions available in the system. The user might guide through the guidelines or attributes or preferences or maybe an example of a product, he is looking for, afterward systems database of items is traversed to fetch the list of items matching to the description provided by the user. Here, suggestions are not the result of users’ rating history. [3][7][23][25][26]. It performs significantly well for products that are not purchased frequently like houses, cars, insurance policies, etc. Quite a limited number of ratings are available, as these products are rarely bought. Also, Set of common attributes between all such objects are complex to identify and does not completely describe the product. KBRS is functional in two types[3][5][25].

- Case-based: When the hooking keys provided by the user points towards a particular item, i.e. user exemplifies his preferences. For eg., Users want a house similar to a house in society x.
- Constraint-based: When the user lays down his requirements and constraints to the attributes of the products and Recommender system filters the database with the items that best match, the users’ specifications. For eg. User want a house with 4 bedrooms, a kitchen, and a balcony within a range of cost.

Benefits[5][23]

i. Working solution to a ramp-up and Sparsity issue.
ii. More user satisfaction as recommendations mostly tends towards their preferences.

iii. Gray sheep concern resolved.

iv. The user has control to modify requirements if desire.

v. Non-product attributes can also be considered.

Limitations[7][23]

i. To build a database and an interface spanning over constraints and rules expertly defending all products’ attributes is often challenging.

ii. They lack in persistent personalization.

iii. Sometimes the suggestions become quite discernible as grounded only on product attributes not on peer ratings.

iv. Knowledge acquisition is a fundamental part, that must be done efficiently. Critiquing Knowledge in three aspects is crucial
   a. Catalog Knowledge – Product features
   b. Functional Knowledge – Products functionality mapped to users needs
   c. User Knowledge – Demographics of user

2.5. Hybrid Recommender system

A hybrid recommender system blends the techniques of more than one above mentioned recommendation methods. This way it can leverage the positives of a method and overcome its limitations, thus gives better performance with the best of all methods. For example, collaborative filtering may be united with knowledge-based to avoid cold start problem[7][23][25]. Hybridization can be performed in three basic ways[23][25]:

i. Ensemble hybridization: Resultant recommendations is the combination of results of various run-of-the-mill algorithms.

ii. Monolithic hybridization: A basic algorithm is modified to accommodate the positives of other algorithms to make a more robust algorithm. There is tight integration between various components.

iii. Mixed hybridization: Like the ensemble hybridization, it also utilizes various algorithms as black-boxes. The difference lies, as the resultant lists from various algorithms are presented side by side.

The prime forte of the hybrid recommendation system is in its potential to incorporate the strength of various systems and overcome their weakness to form a state-of-art robust system.

3. Related Work

Aljunid and Manjaiah [27] proposed a state-of-art technique named Deep Learning method of a Collaborative Recommender System (DLCRS). They evaluated DLCRS on two eminent datasets: 1M Movielens and 100K. They examined its performance by comparing it with other existing methods. The evaluation was done by applying the Root Mean Square Error (RMSE) measurement metric by inputting several parameters as the number of users, items, ratings, rating sparsity, etc. They concluded that the performance measure of MovieLens 1M is better than the MovieLens 100k dataset and the proposed method achieves lesser RMSE than other methods.

Alhijawi [10], presented a novel recommender system based on an optimization technique called OptNibors, to improve prediction accuracy. This technique comprises two stages viz. preprocessing and optimization phase. Two experiments were conducted on a set of 30, 50, and 70 selected neighbors. The superiority of OptNibors in recommendation accuracy and quality is evaluated using MAE (Mean Absolute Error) and using F1- measures. The results establish improvement by 31.1% and 7.7% respectively, concerning other prevailing methods like SimGen, PRC, and COS.
Bogdan and Vladimir [23], introduces a monolithic hybrid RS with a name Predictory, which blends the SVD based collaborative system, a content-based system, besides a fuzzy expert system. The proposed structure is developed to acclaim preferred movies to an active user. The system considers various parameters before making suggestions like rating history, average movie rating, favorite and unpopular genre, and the number of ratings. The MovieLens dataset was undertaken using standard metrics like precision, F1-measures, and recall. Significant results were realized, precision is 81%, recall is 83%, and F1-measures is 82%.

Sujoy, Abhijeet, and Manoj [18] acquaint with a new cost-effective approach of an Integrated RS for improved correctness and personalization. They proposed a Significant nearest neighbor instead of nearest neighbor and the timestamp of targeted rating is supplied as an input to improve recommendation accuracy. four kinds of prediction models were then applied to this SNN. F1 measure and aggregate diversity measures were used to prove the efficiency of the new system in contrast to old methods. And claims to increase profits by recommending a diverse range of products. However, it can perform notably low when ratings are in binary format or highly dense.

Mehdi, Francesco, and Neil[14] survey various active learning strategies, that had been proposed by classifying them into two genres namely personalization and hybridization. Whereby, personalization is the accuracy of the user-tailored list of recommendations. And hybridization is a variety of active learning criteria. They summarised the survey with findings that there is a substantial reduction in Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) when the evaluation is done offline. The highest and binary prediction-based strategies are the best performers when precision is concerned. Ranking quality is finest when Representative-based or Voting strategies are used. The popularity of items can play a key role in acquiring a large number of ratings. The survey done was limited to collaborative type filtering only.

4. Result and Discussion
From the past few years, the Recommender system proves to be an exploring area for research. An enormous number of authors from all over the world have contributed to the topic. So a bibliometric analysis is conducted out using different metrics. A quantitative analysis was done using parameters of academic literature such as authors, keywords, sources, and affiliations is known as bibliometric analysis. A total of 556 publications were analyzed, comprising of 346 conference papers followed by 175 articles and 13 book chapters from different sources as shown in figure 8.

![Figure 8: Documents by types](image)

25.7% of papers are from China shadowed by 18.7% of papers from India as depicted in figure 9.
25.18% of total work published on the recommender system in the year 2019 whereas 21.76% was published in the year 2018 represented by figure 10.

Out of 2580 total keywords in 556 papers, a major share of 462 occurrences are of recommender systems trailed by 348 occurrences of collaborative filtering as shown in figure 11.
From a total of 556 documents, Nanjing University has contributed over 9 papers, 8 are from Mohammed V University and 5 papers are from the University of Delhi, as rolled out by figure 12.

Figure 12: Documents by Affiliations

Figure 13 displays the three-fields plot by County, Author and Keywords, which shows a major sweep is done by China with keyword recommender systems.

Figure 13: Three-Fields Plot by County, Author, and Keywords

National Natural Science Foundation of China has majorly funded the publications in the field followed by Fundamental Research Funds for the Central Universities. Most of the publications are sourced from ACM International Conference Proceeding series and Lecture Notes in Computer
Science. It can be concluded from the analysis done above that among the total publications were done in the last five years, the recommender system and collaborative filtering are the topmost keyword preferences of the authors and widely trending in the year 2018 and 2019. D. Jannach has given maximum contribution to the topic.

5. Conclusion
With the plethora of items/products available online, it's quite a cumbersome task for the user to pick or traverse the right product one is looking for. Recommender systems not only make this task simple for the user but also an added advantage for the businesses online to shoot up their profits. They can offer a vast display of items to their customer, without shelving them physically. A recommender system is a computerized program that offers the active user a list of items as recommendations grounded on users' interaction with the system. Broadly, a Recommender system can be classified into Collaborative, Content-based, Demographic, knowledge-based, and Hybrid systems.

It can be established through the analysis done by biblioshiny tool of RStudio, that much of the work has been done on the recommender system in the year 2018 and 2019 and it's still going strong with collaborative and hybrid filtering as a top preference by authors. A lot of research work has to be done to deal with various challenges in the field of recommendation systems. The accuracy of predictions in the rating is a major problem in recommender systems. Sparsity, due to lack of rating history especially in the collaborative method is an important issue, which leads to another severe challenge called Cold Start. Challenge for new user and new item to find its place in the CF system, though, it gets somewhat resolved in the Content-based system. Low scalability in real-time can deteriorate the performance and poses a challenge to be administered.

References
[1] Ricci F et al 2015 Recommender Systems Handbook (Boston: S. Springer US) pp 1-34
[2] Wikipedia, Recommender system https://en.wikipedia.org/wiki/Recommender_system
[3] Sielis GA, Tzanavari A and Papadopoulos GA 2015 Recommender Systems Review of Types Techniques, and Applications Encyclopedia of Information Science and Technology 10 7260-7270
[4] Ryngkai I and Chameikhlo L 2014 Recommender Systems: Types of Filtering International Journal of Engineering Research & Technology 3 pp 251–254
[5] Khan M M 2020 A Review on Filtering Techniques Used for Classification of Recommender Systems SSRN Electron J pp 1–5
[6] Alhijawi B and Kilani Y 2020 A collaborative filtering recommender system using genetic algorithm Information Processing & Management 57(6) p 102310
[7] Vekariya V and Kulkarni G R 2012 Hybrid recommender systems: Survey and experiments 2nd Int. Conf. Digit. Inf. Commun. Technol. its Appl DICTAP pp 469–473
[8] Introduction to recommender systems - Towards Data Science Towards Data Science https://towardsdatascience.com/introduction-to-recommender-systems-6c66c615ada
[9] Su X and Khoshgoftaar T M 2009 A Survey of Collaborative Filtering Techniques Advances in artificial intelligence pp 1-19
[10] Alhijawi B 2019 Improving Collaborative Filtering Recommender System Results using Optimization Technique Proc. of the 2019 3rd Int. Conf. on Advances in Artificial Intelligence pp 183-187
[11] Bobadilla J, Ortega F, Hernando A, and Gutiérrez A, 2013 Recommender systems survey Knowledge-Based Syst 46 pp 109–132
[12] Bellogín A, Cantador I, Diez F, Castells P and Chavarriga E An empirical comparison of social, collaborative filtering, and hybrid recommenders 2013 ACM Transactions on Intelligent Systems and Technology (TIST) 4(1) pp 1-29
[13] Koren Y, Bell R and Volinsky C 2009 Matrix factorization techniques for recommender systems Computer 42(8) pp 30-37
[14] Elahi M, Ricci F and Rubens N 2016 A survey of active learning in collaborative filtering recommender systems Computer Science Review 20 pp 29–50
[15] Pazzani M J 1999 Framework for collaborative, content-based and demographic filtering Artificial intelligence review 13(5) pp 393–408
[16] Zhang Z, Peng T and Shen K 2020 Overview of Collaborative Filtering Recommendation Algorithms IOP Conf. Series: Earth and Environmental Science 440(2)
[17] Robillard M P et al 2014 Recommendation Systems in Software Engineering (Heidelberg: Springer) chapter 2 pp 301-327
[18] Bag S, Ghadge A and Tiwari M K 2019 An integrated recommender system for improved accuracy and aggregate diversity Computers & Industrial Engineering 130 pp 187-197
[19] Sarwar B, Karypis G, Konstan J and Riedl J 2001 Item-based collaborative filtering recommendation algorithms Proc. 10th Int. Conf. World Wide Web 1 pp 285–295
[20] Cacheda F, Carneiro V, Fernández D and Formoso V 2011 Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems ACM Transactions on the Web (TWEB) 5(1) pp 1-33
[21] Batmaz Z, Yurekli A, Bilge A and Kaleli C 2019 A review on deep learning for recommender systems: challenges and remedies Artificial Intelligence Review 52 pp 1–37
[22] Torstensson E 2019 Comparing neighborhood and matrix factorization models in recommendation systems Saving the user some clicks
[23] Walek B and Fojtik V 2020 A hybrid recommender system for recommending relevant movies using an expert system Expert Systems with Applications 158 p 113452
[24] Falk K 2019 Practical Recommender Systems (Shelter Island : Manning Publications Co)
[25] Aggarwal C C 2016 Recommender systems (NY: Communications of the ACM)
[26] Knowledge-Based Recommender Systems: An Overview https://medium.com/@jwu2/knowledge-based-recommender-systems-an-overview-536b63721dba
[27] Aljunid M F and Dh M 2019 An Efficient Deep Learning Approach for Collaborative Filtering Recommender System Procedia Computer Science 171