Recommendation System: A Literature Survey

Mohan Kubendrian, Nishal Pradhan

Abstract*: Recommendation systems aim at identifying the best products and contents that suits the preference of a user. It has become increasingly popular in a number of areas, like recommending books, news articles, movies, music, commercial products, restaurants, web pages, and many more. Retail companies and e-commerce sites take full advantage of recommender systems in order to boost their profit margins by boosting sales or rather leverage the data that are provided to them. To improve performance and accuracy for a more specialized recommendation for each customer, there has been a lot of research on developing hybrid recommendation systems instead of improving collaborative or content-based methods alone. Hybrid systems club together both content-based and collaborative based methods. LightFM, a hybrid model, has been proven to be the most effective when it comes to recommendation systems. This makes it interesting to study the effectiveness of LightFM compared to other existing models. In this study, we provide a literature survey of the existing recommendation systems, with our focus based on LightFM which is used for implicit feedback and user-item cold start problems.

***Index Terms*: Collaborative Filtering, Content-Based Filtering, Explicit Feedback, Hybrid Filtering, Implicit Feedback, LightFM.**

I. INTRODUCTION

A recommender system is a filtering technique which aims at predicting the preference of a user or rather how the user would rate an item and would prefer an item in the future. Let us look at an example from the online movie streaming company called Netflix. Netflix always recommends movies to users based on what he/she has watched or clicked on. Suppose, a user likes action movies, Netflix would recommend that user action movies all the time. In the case of online shopping site called Amazon, when a user views a product, it also recommends other similar products viewed by other users to that user. Likewise, in the case of friend suggestions on Facebook, it always recommends friends based on location, user's details and history, mutual friends, etc.

Recommender system has become increasingly popular in this era which consists of technology and advancement in a number of areas, like recommending books, news articles, movies to a user, music, commercial products, restaurants, web pages, and many more. Retail companies and e-commerce sites take full advantage of recommender systems in order to boost their profit margins by boosting sales or rather leverage the data that are provided to them. It aims to identify a user's interest through this data and predict specific items to them or items that the users might find interesting. A typical case of recommending items would be in the form of ratings. For example, let us take the case of movie ratings that the user provides. Whenever a user provides a high rating to a movie, say 5 is the highest, the recommender system would likely start recommending movies that match the genres that the user just rated 5. If user rates 1, the recommender system would likely stop recommending such genres of movies to that users. So, the case here is that the customer is providing the ratings and recommender systems are recommending items based on the ratings. This is called Explicit Feedback. But there are also cases where the user watches movies but never provides a rating for the movie he watches. These are cases where recommendation system tends to be challenging because now, recommendation system has to constantly track that user's activity, like which genre of movie he is clicking, or the web pages he is landing to, or the number of minutes/hours spent on a particular page and through this information, the system has to recommend products to the customers in order to gain profits. This type of system is called Implicit Feedback. We are going to discuss in detail in the later section on this topic of explicit and implicit feedback systems. So, the basic idea or principle behind recommender systems is that there exists relationship between user and item. Let's take the example of a customer in Netflix who watches only a specific genre of movie. So, if the customer likes watching action movies, it is highly probable that he would prefer watching action movies in the future too.

II. BACKGROUND

The primary aim of recommender system is to gather facts and information on how the users prefer items (e.g., articles, commercial product, movies, friends in the case of social media, etc.). This type of information can be collected from the user either explicitly or implicitly. Explicitly as in, the information can be obtained from user ratings or some other data explicitly provided by the user. Implicitly collecting data means analysing user's behaviour like the type/genre of songs he listens to, the number of visits on a particular item webpage or the genre of movies he clicked on, etc. [1], [2]. Recommender systems also make use of demographic information like tagged location or age or sex. Recommender systems also use a lot of other information like the number of followers, tweets, following etc.

Revised Manuscript Received on May 10 ,2019

Dr.Mohan Kubendiran, Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India.

Nishal Pradhan, PG student, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India.

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

Since the internet of things is the next big thing in the market, recommender systems often use real time information of these products to improve their recommendation system. (e.g., geo-tagged locations or GPS, real-time signals, radio frequencies, etc.). Recommendation systems use various sources of data so that it can deliver users with the right prediction/recommendation of items. It then uses measures like accuracy, mean squared error and other evaluation measures. The most important technique by far has been Collaborative Filtering. Even though it is used with other techniques like knowledge-based, content-based, etc., they have proved to be a major role in recommendation systems. Of late, the implementation of recommendation system in the industry has increased drastically. Netflix, the online movie streaming global giant uses recommendation system in and out in order to leverage their platform. Amazon, the world's largest online retail store, uses it thoroughly for its product recommendation to its customers and likewise there are several other platforms where it is used for news/article recommendation and of course not to forget the famous social networking sites like Twitter, Facebook and Instagram. This evolution of recommendation system shows how important hybrid techniques can be, in recommender systems. These hybrid techniques are basically systems where different recommender system techniques are merged so that we can yield the advantages of each of these individual techniques and provide a better system as a whole. The survey on these hybrid techniques have been discussed and presented in [3]. With the drastic advancement in technology, there is a huge influx of data and information, especially in this decade, people have become overwhelmed with the amount of information that are provided to them in the internet. This in turn creates a huge problem for users when choosing items from, given a large set of data. This is where recommendation system comes into picture. Recommendation system thus helps in identifying the right product for the user from these large volumes of data and hence, makes the life a customer easier and saves time too [4]. Presently with the approach of web-based business sites like Amazon, it turned out to be progressively clear the critical job that recommender systems play. The significance of recommender systems additionally became visible in 2006. Netflix, an online movie streaming giant declared an open challenge in 2006, to predict ratings, based on what the users rated previously. The catch here was, there were no information provided about the films nor the users. The challenge here was, it had to improve Netflix's own algorithm and the team who finally won the challenge achieved about 10% improvement on the algorithm of Netflix [4]. To honour their hard work and feat, they were awarded 1 million US dollars. The way toward creating a recommendation system is based on a mix or say combination of these [5]:

1. The kind of information accessible in the database (e.g., user id, ratings, content for items which will turn into feature at a later stage, user relationships and also geotagged location information).
2. The different filtering techniques/algorithms that is used (e.g., content-based, collaborative, demographic, hybrid).
3. The type of model (e.g., ''memory based,'' where data is directly used or ''model-based'' where data is used to generate a model).
4. The various machine learning techniques used are also to be considered: use of probabilistic methods like Bayesian Networks, Nearest Neighbours Algorithm, Artificial Neural Networks, Genetic Algorithms, Deep Learning, Fuzzy Models, Singular Value Decomposition Techniques for reducing sparsity.
5. Sparsity of the dataset and also the scalability.
6. Performance in terms of computation with respect to time and space.
7. The class is considered (e.g., recommendations of top N predicted items)
8. Quality results through various evaluation measures (e.g., precision, accuracy, etc.).

The research in the recommendation system field requires utilizing a set of open databases so that investigations on these models and algorithms and techniques can be facilitated. It is from these open databases that the scientific community can experiment upon and validate and come up with improved techniques. There are several public databases available already, namely, Movielens, Netflix, Last.Fm, etc.

The filtering algorithms in recommendation system are divided as [6]:

1. Collaborative Filtering
2. Demographic Filtering
3. Content-based Filtering
4. Hybrid-based Filtering

A recommender system consists of:

1. **Users**: These are people who have affinity for items/products [4]. Here, a user will have its own attributes. For e.g., demographic attributes like age, sex, etc. and there are also users from whom a model can be inferred, based on their likings like genre of movies/music.
2. **Items**: These are the products which the recommendation system chooses for recommending it to the user [4]. Every item will have a set of attributes/properties. For example, a movie will have a genre, language etc. Similarly, an article can be a news article or an article on technology, etc.
3. **Preferences:** These basically refer to items that the user would like to prefer over other items. It is basically denoted by a like or a dislike or in most cases by a rating from 1 to 5 [4]. Let's say, a movie can be rated by a user based on a scale of 1 to 5 with 5 being the most liked and 1 being the worst movie for that particular user.

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

The elements discussed above are used by the recommendation algorithm in various ways. So, these algorithms can also be classified into various types as follows:

1. **Demographic filtering** is supported on the rule that people with certain normal individual traits (sex, age, nation, and so forth.) will likewise have regular inclinations [8].
2. **Content-Based filtering**: The user gives rating to items and hence a user preference is created based on what he likes and dislikes. This in turn helps in creating a model of user preferences for items [4]. Let's take an example in the movie domain. Let's say a user likes romantic comedy and action but hates horror. So the algorithm will learn about this trait of a user overtime and marks that the user has a positive score for genres like romantic comedy and action whereas it has a negative score for horror. Also, there might be users who are a fan of some actors and who are not. So, the algorithm will figure about a particular user, say user A, who likes all by Brie Larson but may not be fan of Julia Roberts. Content based filtering utilises this data to map the ratings of the users against product attributes. So the idea here basically is that content based filtering makes recommendation based on our past choices (e.g. in the online retail shopping platform Amazon, if you buy or land on a page on cameras, it will also recommend some other cameras to you as well based on the type of camera you just browsed). One more interesting fact is that content based filtering can also make recommendation according the item content present in the data or list. So with analysis, such as similarity measure, a similar item can be recommended in the future. For example, suppose a content based filtering algorithm recommends an item A to user1. So, through cosine similarity measure, a similar item to that of item A can be found and recommended to user1 in the future [7].

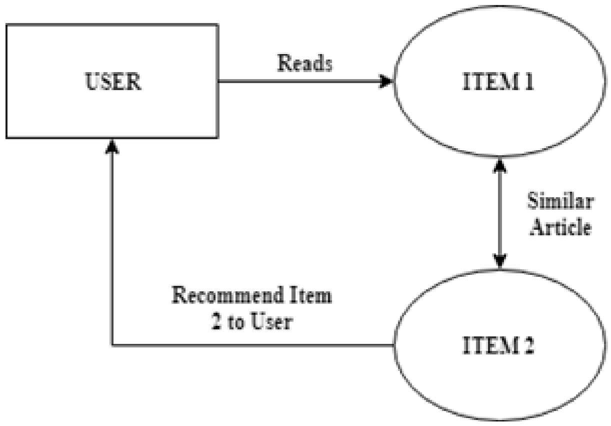

Fig. 1: A flowchart depicting Content-Based Filtering

3. **Collaborative filtering**: More emphasis is given to the ratings of the user than the data attributes for prediction and recommendation [4]. In collaborative filtering, we have user and item model where user has ratings and also item has a set of ratings. When we combine these two models, we get a rating which is basically a sparse matrix rating with some filled cells and some unfilled cells. Now, the task here is to fill the unfilled cells with ratings and in the case of filled cells, it has to be chosen to recommend an item. So, Collaborative Filtering [6], [9], [10], [11] basically allows users to give ratings on items such as news article, products or any content in a way that when we have gathered enough information, "items are recommended based on information which those people have provided with who are considered to have the most in common with them". Collaborative Filtering provides a wide area in terms of research [12], [13]. Also, in CF, we can obtain ratings in an implicit manner as well, like, we take into account the times a user lands on a particular webpage and the time spent on it, activity done on a particular music/article, etc.)
4. **Hybrid filtering** [15], [8]: Hybrid filtering has proven more powerful than ever before. It basically clubs two filtering methods to give an even better result than the individual filtering would have given otherwise. Some common examples include "combination of demographic filtering with collaborative filtering [16] or content based filtering with collaborative filtering [17], [1]. Hybrid Filtering is largely associated with probabilistic methods like Artificial Neural Networks [21], [22], [23], Bayesian Networks [24], Genetic algorithms [18], [19], latent features [26], fuzzy genetic [20] and clustering [25]. This whole idea of recommendation system research started in mid 1990s where the researches started focusing on problems with items that had ratings in it. In the basic problem sense in recommendation system, it's problem boils down to solving or rather estimating the rating of items which the user has not seen it yet. The basic intuition here is the fact that users give ratings to items based on their preferences and also on explicit feedback criteria and we use this information to estimate ratings on other items which the use has not rated yet. Thus, when we are finished with assessing the ratings for the items, the users with the most noteworthy evaluated ratings are recommended with the product/item.

The new evaluations of the items not rated yet can be assessed from multiple points of view utilizing techniques from approximation theory/estimate hypothesis, machine learning and different heuristics.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

III. LITERATURE SURVEY

Recommender Systems are typically characterized by their way to deal with estimation of ratings. Here, we study different types of recommender systems. The definition was expressed in [27], [28] for the first time and has been considered widely. Additionally, recommender systems are normally arranged into these categories, in light of how suggestions are made:

1. **Content-based**: Recommendation of items happens based on how the user favoured items previously.
2. **Collaborative-based**: In this case, we look for users with similar preferences and tastes and based on this, the user will be suggested items;
3. **Hybrid-based**: These are methods which clubs together both "content-based and collaborative-based methods".

Apart from the ideas we discussed above about recommender systems predicting the ratings which a person would give to a not yet seen item, there are

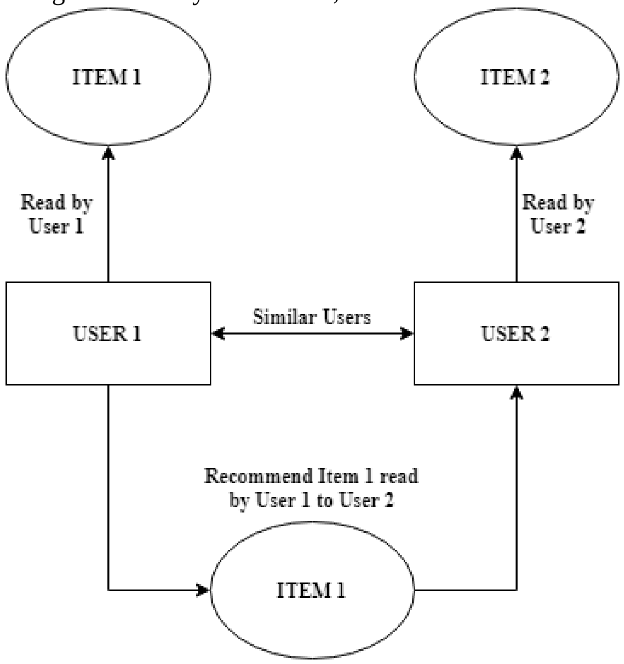

Fig. 2: A flowchart depicting Collaborative-Based Filtering

several other work done in the field of recommender system which we call preference-base filtering. This filtering basically estimates the preference of the users on items [29], [30], [31], [4]. Consider movie recommenders. Here, the preference filtering would help in ranking the movies based on ranking for the customers or users. Nevertheless, this paper has its focus mainly on rating based systems since it is one of the major and famous was to deal with recommender systems.

A generally acknowledged scientific classification arranges the framework into model-based and memory-based:

Memory-based [6], [10], [32], [33]: This method does the following; (a) builds a matrix which consists of ratings from the feedback of the user and (b) utilize the recently updated rating because the results are constantly refreshed. These methods make use of similarity metrics in order to evaluate the distance between items and also between users.

Model-based [6], [11]: The recommendation system in this creates a model based on the information provided to them. "Among the most broadly utilized models we have matrix factorization [40], neural networks [35], latent features [39], fuzzy systems [36], Bayesian classifiers [34], genetic algorithms [37], [38], and among others."

In [41], news recommendation system was displayed as a problem which was contextual bandit. So, a new algorithm was conceived by the authors in order to tackle this issue. It was called LinUCB. This algorithm tackles the bandit problem and can also be applied to other domains as well apart from news. This algorithm greatly increased the effectiveness of the recommendation system as it increased the number of clicks in the system to 12.5% when compared to the system without this algorithm in use.

[42] is a great paper when it comes to news recommendation. The authors have used both memory as well as model based systems here. When it comes to memory based technique, it uses "weighted average of the ratings" that have been provided previously by the customers. Here, weight is relative to the similarity between users. The most common measures used are "Pearson Correlation Coefficient and Cosine Similarity". In "model based technique", the feedback/rating are models based on user's history or past data. After the model based technique, they use a clustering technique called PLSI and MinHash. When it comes to memory based, they make use of item co-visitation. Now, all of these three methods appoint a score. At last, two mixes of three systems (PLSI, co-appearance and MinHash) were utilized for assessment. In one mix, co-appearance was given a higher load than different systems (2.0 rather than 1.0). CVBiased was the name given to this combination and they also experimented with another combination which was between MinHash and PLSI wherein they were given higher weight of 2.0 and thus this combination was named CSBiased. In this experimentation, they used recommendation based on recent popularity which was called Popular and this acted as the baseline for it. It was watched, by and large both CVBiased and CSBiased performed 38% superior to the standard Popular on live traffic. One more important paper is [41], and this work was based on [42]. What they have used here is information filtering. So, information filtering basically expels undesirable data from a data stream. To begin with, they completed an investigation/analysis of a user's interest over a 14-month time span for every class of article (categories as pre-characterized). In order to study the interest of the user, they basically used click distribution of the user. At that point they utilized Bayes guideline to predict a clients' enthusiasm for a specific timeframe. At that point, predictions made for specific period were joined to predict a client's interest for an extensive stretch. The expectations made till now were clients' certified interest.

Be that as it may, a user additionally gets affected by current news patterns, so Bayes rule is utilized to make forecasts utilizing click circulation of a short ongoing period (precedent an hour ago). At long last, users' veritable interest is joined with current news patterns to get IF (article). This score for each article IF (article) is multiplied by collaborative filtering score for the article CF (article), and the last score after augmentation of CF (article) and IF (article) is utilized to rank the articles. The communitarian sifting score CF (article) is gotten from [31]. Utilizing this strategy, navigate/click rates (CTR) were improved by 30.9% after existing technique for example [42].

The previous papers we discussed dealt with news recommendation in particular but [43] does not talk about a particular domain but instead is interested in implicit feedback model in recommendation system. So, they basically use implicit feedback dataset. For finding the number of times a user 'u' did activity on a program 'i' (r_{ui}), they make use of latent factor models. Least square optimization technique was utilized with regularization to abstain from overfitting while finding the parameters to process r_{ui}. This methodology was to prescribe TV programs. The fascinating thing about this methodology was there was just positive criticism

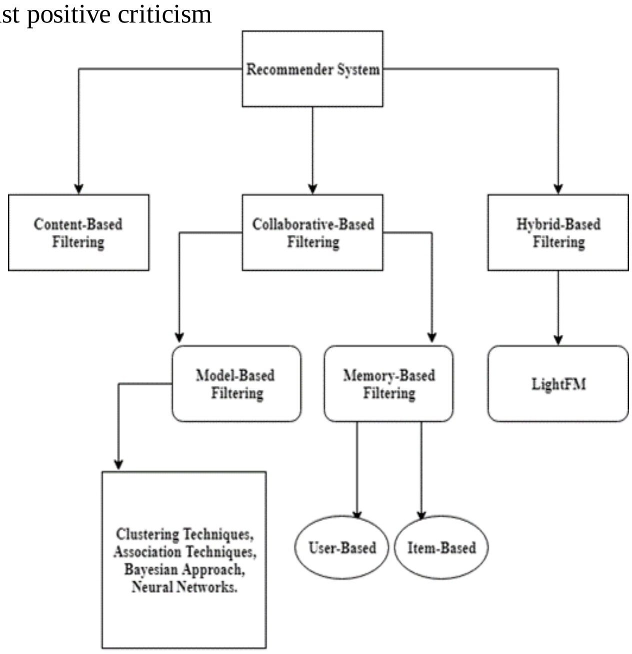

Fig. 3: Recommendation System Flowchart [48]

and no negative input and this made the undertaking of suggestion substantially more troublesome. Assessment was finished by the clients themselves as they were given a rundown of TV programs arranged from most liked to least favored shows. Expected percentile rank was utilized for correlation where latent factor display beat both the models, namely neighborhoods and popularity.

In [44], they have worked on the domain of music recommendation and they have focused on how to map implicit feedback to explicit ratings. They have studied the relationship between implicit feedback and explicit ratings and they have also done an analytic study on the impact of different other variables. To predict ratings, they use a linear model. So, they conducted a study in which they collected data based on users rating the album in a scale of 1 to 5. They had three variables for analysis, wherein implicit feedback was, how many times the customer played a particular song, worldwide popularity which indicated amassed play tally of all users on a given song and lastly, the recently played which indicated the time since the user played a given song last. All these values are then divided into bins – namely, low, medium, high. It was seen that quantized implicit feedback and the way in which ratings were distributed were connected, quantized recentness and the way in which rating were distributed were likewise related yet not as unequivocally as implicit feedback and no noteworthy connection was found between quantized worldwide prevalence and rating distribution. Examination of connection between different factors demonstrated some connection among recentness and implicit feedback just and no other pair. Linear Regression was utilized for the rating prediction utilizing three factors and RMSE (root mean squared error) was utilized for assessment. Four models were utilized for prediction, initial one was just with implicit feedback, in second one recentness was additionally included with implicit feedback, in third worldwide prominence was added to the second and in the fourth and last, a variable which was augmentation of implicit feedback and recentness was added to the third. The standard utilized was rating average. In spite of the fact that execution of all models was generally same, however, they all had normal improvement of 6.5% over the benchmark. There are likenesses and contrasts between the works depicted above The similitude and contrast is in the assessment metric or the evaluation metric, the features used in the model and the machine learning models used. There are papers which have utilized linear regression [44], [41] and logistic regression [44], yet none of them have demonstrated a comparison among them. In assessment, RMSE has been utilized [44] however a large portion of them have favored MAP and different other metrics, some have even tried their frameworks on live traffic.There are certain recommender systems who make use of hybrid approaches. A major part of hybrid technique is the combination of content based and collaborative based techniques. Hybrid technique was built for the fact that it could avoid limitations of both the techniques and at the same time draw the advantages of both of these techniques. However, there are different ways of combining these techniques to get a better hybrid recommender system:

1. build "collaborative and content based" models and then combine them for prediction,
2. some pros of content based or rather advantageous characteristics of it can be incorporated in collaborative system,
3. some pros of collaborative based or rather advantageous characteristics of it can be incorporated in "content-based" system, and

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

4. fuse both content-based and collaborative based techniques [6].

The above methodologies have been experimented upon by recommender frameworks specialists, as portrayed below:

1) Combination of recommenders which are separate: One way to build a hybrid system is to create content based and collaborative frameworks separately. At that point, we can have two unique situations. Initially, we can consolidate the yields (evaluations) acquired from these recommender frameworks into one final recommendation utilizing a linear combination of voting or rating scheme. On the other hand, we can utilize one of the individual recommenders, at some random minute utilizing the better one depending on the metric score. For instance, the DailyLearner framework chooses the recommender framework that can give the suggestion with the more elevated amount of certainty, while one paper picks the one whose proposal is increasingly reliable with past evaluations of the user [6].

2) Content-Based to Collaborative System: A few hybrid recommender frameworks depend on conventional collaborative filtering methods yet in addition keep up the content based profiles for every user. These are then used to find similarity among users. As referenced in [6], this permits to beat some sparsity-related issues of a pure collaborative oriented methodology since, ordinarily, relatively few sets of users will have a critical number of normally rated items. Another advantage of this methodology is that users can be suggested an item not just when this item is evaluated exceptionally by users with comparable profiles, yet additionally straightforwardly, i.e., when this item scores very against the user's profile.

3) Collaborative to Content-Based Systems: This system use dimension reduction techniques. For instance, as per [6], latent semantic indexing is used (LSI) to make a collaborative oriented perspective on an accumulation of user profiles, where users are denoted by term vectors, bringing about an act improvement contrasted with the content based methodology.

4) Fusion of both: Numerous analysts have pursued this methodology as of late. For instance, [6] make use of collaborative and content based filtering (e.g., could be the genre of movie or sex or age) in a single rule-based classifier. Some make use of probabilistic method for combining collaborative and content-based recommendation system. Also, in some cases Bayesian models are used that utilize Markov chain Monte Carlo strategies for parameter estimation and prediction [6].

In light of hybrid systems, let us discuss the latest hybrid system called LightFM. LightFM is a "hybrid model that incorporates both content-based recommendations and the transfer learning of collaborative filtering methods" [49]. It was developed by Lyst, a Fashion e-Commerce site based in London by Maciej Kula, in 2015.

LightFM makes sense of what users like by learning connections that map users and user metadata to the tasks and undertaking metadata that they like. These connections are called embedding.' In a request to construct these inserting, LightFM uses three arrangements of data: the user metadata, the item metadata, and the interaction between them.

For user metadata, we assemble a framework containing the majority of the master's business and expertise labels, just as chosen words from their profile slogan and 'About Me' segment. For instance, on account of movielens100k data, user feature matrix is assembled which essentially is a network of user versus attributes with 0 and 1 as entries.

Correspondingly, for item metadata, we fabricate a matrix containing the venture's business labels, ability labels, chosen words from the task name and description, and spending range. On account of movielens100k information, it is an item feature lattice of movie vs genre with 0 and 1 as entries.

Now we get the interaction matrix from the user and item matrix wherein we have entries of 0 and 1 and 1 represents an interaction and 0 represents no interaction. So, this is basically a matrix factorization of "user and item feature matrix".

LightFM then uses all these three matrices and solves for the embedding so that it can correctly predict the items.

This framework is somewhat a combination of collaborative and content based method. It figures out how gatherings of specialists or experts or users cooperate with different activities or items, similar to a collaborative filter, yet it is additionally learning connections among user and item metadata, similar to a content based framework. By working in the two universes, LightFM gives us the qualities of both, and encourages us coordinate our various tasks to our similarly assorted specialists [15].

A. Phases of Recommendation System

1) *Information Collection:* In this phase, the task is to collect as much data of users as possible which will help in

Table I: Recommendation System Classification [6]

Recommendation Approach	**Recommendation Techniques**	
	Heuristic-Based	Model-Based
Content-Based	Commonly used Techniques: 1. TF-IDF (Information Retrieval) 2. Clustering	Commonly used Techniques: 1. Bayesian Approach 2. Clustering 3. Decision Trees 4. Artificial Neural Networks
Collaborative-Based	Commonly used Techniques: 1. Nearest Neighbor (Cosine, Correlation) 2. Clustering 3. Graph Theory	Commonly used Techniques: 1. Bayesian Approach 2. Clustering 3. Artificial Neural Networks 4. Linear Regression 5. Probabilistic Models

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Hybrid-Based	Fusion of Content-Based and Collaborative-Based Filtering using: 1. Linear combination of predicted ratings 2. Various voting schemes 3. Incorporating one component as a part of the heuristic for the other 4. LightFM	Fusion of Content-Based and Collaborative-Based Filtering by: 1. Incorporating one component as a part of the model for the other 2. Building one unifying model

creating a user profile of important information. The data has to include user's behaviour or explicit feedback as in age, gender, some useful preferences. This we do so that the recommendation system we built in the later stage can function in the most accurate manner [45].

2) Explicit Feedback: The system will ask the customer or the user using the interface to choose their preferences or assess some of the qualities through a questionnaire for example. So, the precision of the recommender system entirely banks on user's input. The only drawback of this system is that users are required to take the assessment and some of them are not always ready to comply with the rule. Despite this, proves to provide trustworthy data and it similarly gives straightforwardness into the proposal technique that results in a to some degree higher suggestion quality and more trust in the proposal [46].

3)Implicit Feedback: The system tracks and keeps record of the customer's behaviour when online, for example, the pages the customer clicks on and spends time on, the activity he/she does on a particular webpage, the number of visits, the kind of items bought, etc. This in a way is effortless on the customer's side as they are not asked to provide data. However, the data collected through this method may not be always accurate [46], [47].

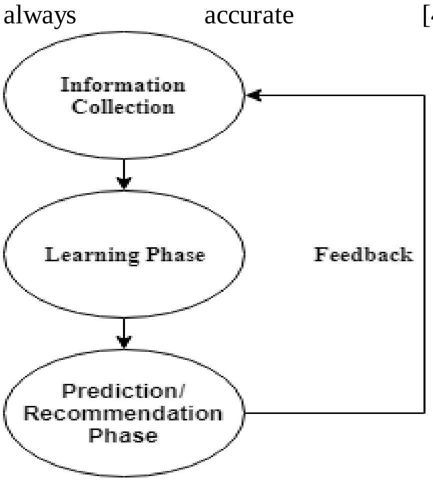

Fig. 4: A flowchart depicting the phases of a recommendation process [48]

4) Hybrid Feedback: Here, the data acquired through explicit feedback system and the data acquired through implicit feedback system is taken into account and combined to build a better hybrid model and also in that wat they overcome their limitations and hence provide a better recommender system.

5) Learning Phases: This is the phase where the training of the data happens. All the information gathered in the information collection phase will be fed to the algorithm so that it learns all the uniqueness and characteristics and features about the user.

6) Prediction/Recommendation Phase: This is the last phase of the recommendation system. Here, with the amount of information collected from the customers, the right products are recommended to the customer, explicitly or implicitly, either by memory based or through model based. [48].

B. Advantages

E-commerce sites and other companies using recommender systems have their main focus on increasing the sales of their products and also build an overall easy and enticing customer experience.

These systems are mainly used for speeding the search results of a customer and help them find products which they would've never found otherwise.

What is more interesting is the fact that these companies are able to send personalized emails to each unique customer on new offers that interests the customers based on the customer's preferences. This in a way helps customers build up a trust and hence not lose customers. Thus, it allows the companies to generate a large revenue out of it.

C. Challenges and Limitations

1) *Data Sparsity:* Due to the massive amount of data, the matrix that is used for prediction can be huge and also very sparse, which will eventually affect the performance of the recommendation system model.

2) *Scalability:* Let's take Netflix as an example. Netflix is the largest online movie streaming company in the world and it has millions of customers and millions of items which means the complexity of the problem of recommending unique items to each and every customer is a huge challenge. Also, these systems need to act in real time to every online movements and requirements. People are shifting to Big Data Technology platforms like Hadoop and Spark in order to meet such requirements.

Cold Start Problem **-** A recommender system requires certain amount of information so that it can rightly predict products to its customers. When there is a lack of such information available, cold-start problems happen. There is so less information or no information at all available about newly joined customers that the system faces difficulty in drawing inferences in order to recommend items to newly joined individuals.

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

IV. CONCLUSION

This study presents some of the techniques used in recommendation systems. Each of these techniques come with its own set of advantages and disadvantages. Hybrid systems by far have been proven to be the most effective in terms of scalability, performance and accuracy. LightFM model performs extremely well as compared to other models mainly because it uses a combination of both collaborative as well as content based filtering and also uses BPR and WARP as loss functions. The model proves very effective in the case of implicit feedback problem and cold-start problem. Hybrid systems were built for the fact that it could avoid limitations of both content-based and collaborative-based techniques and at the same time draw the advantages of both of these techniques. However, there is a continued research in this field for creating an even better model than ever before and will only continue to get better and better.

REFERENCES

1. Choi, K., Yoo, D., Kim, G., & Suh, Y. (2012). A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electronic Commerce Research and Applications*, *11*(4), 309-317.
2. Núñez-Valdéz, E. R., Lovelle, J. M. C., Martínez, O. S., García-Díaz, V., De Pablos, P. O., & Marín, C. E. M. (2012). Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*, *28*(4), 1186-1193.
3. Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4), 331-370.
4. F. Ricci, L. Rokach, B. Shapira and P.B. Kantor, Recommender Systems Handbook, Springer, 2011.
5. Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-based systems, 46, 109-132.
6. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge & Data Engineering, (6), 734-749.
7. Salter, J., & Antonopoulos, N. (2006). CinemaScreen recommender agent: combining collaborative and content-based filtering. *IEEE Intelligent Systems*, *21*(1), 35-41.
8. Porcel, C., Tejeda-Lorente, A., Martínez, M. A., & Herrera-Viedma, E. (2012). A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. *Information Sciences*, *184*(1), 1-19.
9. Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 5-53.
10. Candillier, L., Meyer, F., & Boullé, M. (2007, July). Comparing state-of-the-art collaborative filtering systems. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition* (pp. 548-562). Springer, Berlin, Heidelberg.
11. Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, *2009.*
12. Bobadilla, J., Hernando, A., Ortega, F., & Bernal, J. (2011). A framework for collaborative filtering recommender systems. Expert Systems with Applications, 38(12), 14609-14623.
13. Bobadilla, J., Hernando, A., Ortega, F., & Gutiérrez, A. (2012). Collaborative filtering based on significances. Information Sciences, 185(1), 1-17.
14. Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web* (pp. 291-324). Springer, Berlin, Heidelberg.
15. www.medium.com
16. Vozalis, M. G., & Margaritis, K. G. (2007). Using SVD and demographic data for the enhancement of generalized collaborative filtering. *Information Sciences*, *177*(15), 3017-3037.
17. Barragáns-Martínez, A. B., Costa-Montenegro, E., Burguillo, J. C., Rey-López, M., Mikic-Fonte, F. A., & Peleteiro, A. (2010). A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Information Sciences*, *180*(22), 4290-4311.
18. Gao, L., & Li, C. (2008, October). Hybrid personalized recommended model based on genetic algorithm. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on* (pp. 1-4). IEEE.
19. Ho, Y., Fong, S., & Yan, Z. (2007, July). A Hybrid GA-based Collaborative Filtering Model for Online Recommenders. In *Ice-b* (pp. 200-203).
20. Al-Shamri, M. Y. H., & Bharadwaj, K. K. (2008). Fuzzy-genetic approach to recommender systems based on a novel hybrid user model. *Expert systems with applications*, *35*(3), 1386-1399.
21. Lee, M., Choi, P., & Woo, Y. (2002, May). A hybrid recommender system combining collaborative filtering with neural network. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems* (pp. 531-534). Springer, Berlin, Heidelberg.
22. Christakou, C., Vrettos, S., & Stafylopatis, A. (2007). A hybrid movie recommender system based on neural networks. *International Journal on Artificial Intelligence Tools*, *16*(05), 771-792.
23. Ren, L., He, L., Gu, J., Xia, W., & Wu, F. (2008, December). A hybrid recommender approach based on widrow-hoff learning. In *2008 Second International Conference on Future Generation Communication and Networking* (pp. 40-45). IEEE
24. De Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *International Journal of Approximate Reasoning*, *51*(7), 785-799.
25. Shinde, S. K., & Kulkarni, U. (2012). Hybrid personalized recommender system using centering-bunching based clustering algorithm. *Expert Systems with Applications*, *39*(1), 1381-1387.
26. Maneeroj, S., & Takasu, A. (2009, May). Hybrid recommender system using latent features. In | *2009 International Conference on Advanced Information Networking and Applications Workshops* (pp. 661-666). IEEE.
27. Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2016). paper recommender systems: a literature survey. *International Journal on Digital Libraries*, *17*(4), 305-338
28. Jugovac, M., Jannach, D., & Karimi, M. (2018, September). Streamingrec: a framework for benchmarking stream-based news recommenders. In *Proceedings of the 12th ACM Conference on Recommender Systems* (pp. 269-273). ACM
29. Chaney, A. J., Blei, D. M., & Eliassi-Rad, T. (2015, September). A probabilistic model for using social networks in personalized item recommendation. In *Proceedings of the 9th ACM Conference on Recommender Systems* (pp. 43-50). ACM
30. Guo, G., Zhang, J., & Yorke-Smith, N. (2015, January). TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings. In *AAAI* (Vol. 15, pp. 123-125).
31. Das, D., Sahoo, L., & Datta, S. (2017). A Survey on Recommendation System. International Journal of Computer Applications, 160(7).
32. Sun, X., Kong, F., & Ye, S. (2005, March). A comparison of several algorithms for collaborative filtering in startup stage. In *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE* (pp. 25-28). IEEE.
33. Symeonidis, P., Nanopoulos, A., & Manolopoulos, Y. (2009, October). MoviExplain: a recommender system with explanations. In *Proceedings of the third ACM conference on Recommender systems* (pp. 317-320). ACM.
34. Caizer, S., & Aickelin, U. (2005). A recommender system based on idiotypic artificial immune networks. *Journal of Mathematics, Models and Algorithms*, *4*(2), 181-198.
35. Roh, T. H., Oh, K. J., & Han, I. (2003). The collaborative filtering recommendation based on SOM cluster-indexing CBR. *Expert systems with applications*, *25*(3), 413-423.
36. Yager, R. R. (2003). Fuzzy logic methods in recommender systems. Fuzzy Sets and Systems, 136(2), 133-149.
37. Gao, L., & Li, C. (2008, October). Hybrid personalized recommended model based on genetic algorithm. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on* (pp. 1-4). IEEE.
38. Ho, Y., Fong, S., & Yan, Z. (2007, July). A Hybrid GA-based Collaborative Filtering Model for Online Recommenders. In *Ice-b* (pp. 200-203).
39. Zhong, J., & Li, X. (2010). Unified collaborative filtering model based on combination of latent features. *Expert Systems with Applications*, *37*(8), 5666-5672.

40. Luo, X., Xia, Y., & Zhu, Q. (2012). Incremental collaborative filtering recommender based on regularized matrix factorization. *Knowledge-Based Systems*, *27*, 271-280
41. Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010, April). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web* (pp. 661-670). ACM.
42. Das, A. S., Datar, M., Garg, A., & Rajaram, S. (2007, May). Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web* (pp. 271-280). ACM.
43. Hu, Y., Koren, Y., & Volinsky, C. (2008, December). Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on* (pp. 263-272). Ieee.
44. Parra, D., & Amatriain, X. (2011, July). Walk the talk. In *International Conference on User Modeling, Adaptation, and Personalization* (pp. 255-268). Springer, Berlin, Heidelberg.
45. Oard, D. W., & Kim, J. (1998, July). Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems* (Vol. 83). WoUongong.
46. Buder, J., & Schwind, C. (2012). Learning with personalized recommender systems: A psychological view. *Computers in Human Behavior*, *28*(1), 207-216.
47. Gadanho, S. C., & Lhuillier, N. (2007, October). Addressing uncertainty in implicit preferences. In *Proceedings of the 2007 ACM conference on Recommender systems* (pp. 97-104). ACM.
48. Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), 261-273.
49. Metadata Embeddings for User and Item Cold-start Recommendations Kula, M. (2015). Metadata embeddings for user and item cold-start recommendations. arXiv preprint arXiv:1507.08439

AUTHORS PROFILE

He is working as an Associate Professor in School of Computer Science and Engineering (SCOPE) in Vellore Institute of Technology (VIT), Vellore, India. His area of interest includes Cloud Computing, e-Health Systems, Internet of Things, Big Data Analytics and Information Security.

He is a postgraduate student in School of Computer Science and Engineering (SCOPE) in Vellore Institute of Technology (VIT), Vellore, India. His area of interest includes Virtualization, Recommendation Systems and Security in Cloud.