RECOMMENDER SYSTEM IN BIG DATA ENVIRONMENT

Manisha Khileri[1], Prof.Riya Gohil[2]

[1] *Masters of engineering, Computer Engineering, LDRP ITR Institute Gandhinagar, Gujarat, India*
[2] *Professor, Computer Engineering, LDRP ITR Institute Gandhinagar, Gujarat, India*

ABSTRACT

Recommendation systems use one of the most intelligent technology for user when we surf on internet using recommendation system user can easily find valuable product from supplier. So here in these paper work on recommendation system for user an try to design. Artificially intelligent system work on time complexity and accuracy.

Keyword: *Nearest neighbour, Cluster, GFCM, Membership algorithm etc.*

1. INTRODUCTION

Data is growing at an enormous speed creating it tough to handle such large amount of data. The main problem in handling such large amount of data is as a result of that the result of that the amount is increasing quickly as compared to the computing resources. The Big data term that is getting used currently a days is quite name because it points out solely the dimensions of the data not putting an excessive amount of attention to its different existing properties [1].

Big Data management stands out as a challenge for IT corporations. the answer to such a challenge is shifting more and more from providing hardware to provisioning more manageable package solutions .Big Data additionally brings new opportunities and significant challenges to trade and domain. Similar to most big data applications, the large data tendency conjointly poses significant impacts on service recommender systems. With the growing range of different services, effectively recommending services that users most well-liked has become a vital analysis issue. Service recommender systems are shown as valuable tools to assist users modify services over load and supply acceptable recommendations to them. Examples of such sensible applications include CDs, book, web content and numerous alternative product currently use recommender systems. Over the last decade, there has been a lot of analysis done each in business and world on developing new approaches for service recommender systems. Many major e-commerce Websites are used recommendation systems to produce relevant suggestions to their customers. The recommendations may be supported numerous parameters, like item popular on the company's Website; user characteristics like geographical location or different demographic information; or past shopping for behavior of prime customers [1].

Big data can be categorized into volume, varity, velocity. Volume means it can process terabytes to zettabytes data. Varity may be in the form of structure, unstructured, semi structure data. Velocity means streaming of data. Recommendation system depicts additional information to the active user such as learning material Recommendation system, Recommendation system in real e-commerce, Music recommendation system in Apple iTunes, Amazon.com recommendation system, many more. The main role of the Recommendation system is information filtering. Recommendation system can be classified into two types, they are knowledge based recommendation and collaborative filtering. The widely used technique is collaborative filtering. The collaborative

filter only focuses on the user preference of active user. Knowledge based recommendation system means asking a user about their own preference of the item and find out what active user required [2].

What is "Big Data"?

Big Data refers to datasets whose size is beyond the ability of current technology, method and theory to capture, manage, and process the data within a tolerable elapsed time. "Big data" refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze. This definition is intentionally subjective and incorporates a moving definition of how big a dataset needs to be in order to be considered big data—i.e., big data size is not defined in terms of being larger than a certain number of terabytes (thousands of gigabytes)[4].

1.1 OBJECTIVES

- The time frame increases as the clusters keep increasing.
- And when one then more core is used, GFCM is faster.
- Reduce time complexity.
- Recommendation with high accuracy.

1.2 MOTIVATION

There has been a lot of analysis done both in business and world on developing new approaches for service recommender systems .a lot of firms capture large scale data regarding their customers, providers and operations. The ascent of the amount of consumers, services and different on-line data yields service recommender systems in "Big Data" setting, which poses crucial challenges for service recommender systems. Moreover, in most existing service recommender systems such as hotel reservation systems and Restaurant place guides the ratings of services and therefore the service recommendation lists presented to users are an equivalent. they need not thought-about users totally different preferences, while not meeting users personalized necessities and plenty of times such recommendations aren't helpful for the user.then there is a requirement of personalized recommendation system which will facilitate the user with the choice of product.

2. RELATED WORK

In this section we briefly present some of the research literature related to collaborative filtering, recommender systems, data mining and personalization. Tapestry is one of the earliest implementations of collaborative filtering-based recommender systems. This system relied on the explicit opinions of people from a close-knit community, such as an office workgroup. However, recommender system for large communities cannot depend on each person knowing the others. Later, several ratings-based automated recommender systems were developed. The GroupLens research system [3, 4] provides a pseudonymous collaborative filtering solution for Usenet news and movies. Ringo [5] and Video Recommender [6] are email and web-based systems that generate recommendations on music and movies respectively. A special issue of Communications of the ACM presents a number of different recommender systems other technologies have also been applied to recommender systems, including Bayesian networks, clustering, and Horting. Bayesian networks create a model based on a training set with a decision tree at each node and edges representing user information. The model can be built off- line over a matter of hours or days. The resulting model is very small, very fast, and essentially as accurate as nearest neighbor methods [7]. Bayesian networks [8] may prove practical for environments in which knowledge of user preferences changes slowly with respect to the time needed to build the model but are not suitable for environments in which user preference models must be updated rapidly or frequently. Clustering techniques work by identifying groups of users who appear to have similar preferences. Once the clusters are created, predictions for an individual can be made by averaging the opinions of the other users in that cluster. Some clustering techniques represent each user with partial participation in several clusters. The prediction is then an average across the clusters, weighted by degree of participation. Clustering techniques usually produce less-personal recommendations than other methods, and in some cases, the clusters have worse accuracy than nearest neighbor algorithms [7]. Once the clustering is complete, however, performance can be very good, since the size of the group that must be analyzed is much smaller. Clustering techniques can also be applied as a "first step" for" for shrinking the candidate set in a nearest neighbor algorithm or

for distributing nearest-neighbor computation across several recommender engines. While dividing the population into clusters may hurt the accuracy or recommendations to users near the fringes of their assigned cluster, pre-clustering may be a worthwhile trade-off between accuracy and throughput. Horting is a graph- based technique in which nodes are users, and edges between nodes indicate degree of similarity [9] between two users. Predictions are produced by walking the graph to nearby nodes and combining the opinions of the nearby users. Horting differs from nearest neighbor as the graph may be walked through other users who have not rated the item in question, thus exploring transitive relationships that nearest neighbor algorithms do not consider. In one study using synthetic data, Horting produced better predictions than a nearest neighbor algorithm. Schafer et al., present a detailed taxonomy and examples of recommender systems used in E-commerce and how they can provide one-to-one personalization and at the same can capture customer loyalty [10]. Although these systems have been successful in the past, their widespread use has exposed some of their limitations such as the problems of sparsity in the data set, problems associated with high dimensionality and so on. Sparsity problem in recommender system has been addressed in. The problems associated with high dimensionality in recommender systems have been discussed in, and application of dimensionality reduction techniques to address these issues has been invest. However, with the discovery of data mining tools and knowledge [11] which inherently is associated with databases more robust approaches can be used to analyze large datasets and make recommendation more quick and easy. Our work explores the extent to which item-based recommenders, a new class of recommender algorithms, are able to solve these problems.

Our work explores the extent to which item-based recommenders, a new class of recommender algorithms, are able to solve these problems.

LITERATURE 1:

Title	**Recommender System in Big Data Environment.**
Author	Udeh Tochukwu Livinus , Rachid Chelouah and Houcine Senoussi
Publication	IJCSI , 2016
Technique/Method	k-Means
Conclusion	Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web 2.New technologies are needed that can dramatically improve the scalability of recommender systems. As one of the most successful approaches to building recommender systems, collaborative filtering (CF) uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users.

LITERATURE 2:

Title	**Towards Keyword Based Recommendation System.**
Author	**Vinaya B. Savadekar, Pramod B. Gosavi**
Publication	**IJSR,2016**
Technique/Method	CF algorithm

Conclusion	This system is more efficient in terms of complexity. And the system gives more accurate results or recommendations to the users. This system is being developed for products based on amazon data set.

LITERATURE 3:

Title	**Recommender System in Big Data Environment.**
Author	Udeh Tochukwu Livinus , Rachid Chelouah and Houcine Senoussi
Publication	IJCSI , 2016
Technique/Method	k-Means
Conclusion	Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web 2.New technologies are needed that can dramatically improve the scalability of recommender systems. As one of the most successful approaches to building recommender systems, collaborative filtering (CF) uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users.

LITERATURE 4:

Title	**Towards Keyword Based Recommendation System.**
Author	**Vinaya B. Savadekar, Pramod B. Gosavi**
Publication	**IJSR,2016**
Technique/Method	CF algorithm
Conclusion	This system is more efficient in terms of complexity. And the system gives more accurate results or recommendations to the users. This system is being developed for products based on amazon data set.

3. STATE OF WORK

3.1.1 Recommendation System (RS)

In big software organizations successful products is one of the most important aspects of organization achievement. Team formation directly affects the performance of the team and the product development quality. Staffing is an important issue to be analyzed when software development is undertaken as a value-driven business. Creating competence-based competitiveness is a great challenge because competence identification and consequently, its management, helps in innovation, decision support, faster process and product quality improvement, and constitute an important input to the creation of the organizational knowledge. The finding a value driven developer-to-project assignment is a complex task. Predicting the compatibility of one with other team members is much more complex.

It just considers a match between these competences with job competence requirements. The existing system does not support the fitness of this with team members in terms of interpersonal compatibility, which is equally important while finding a team member for a particular project and we are shown Figure [9].

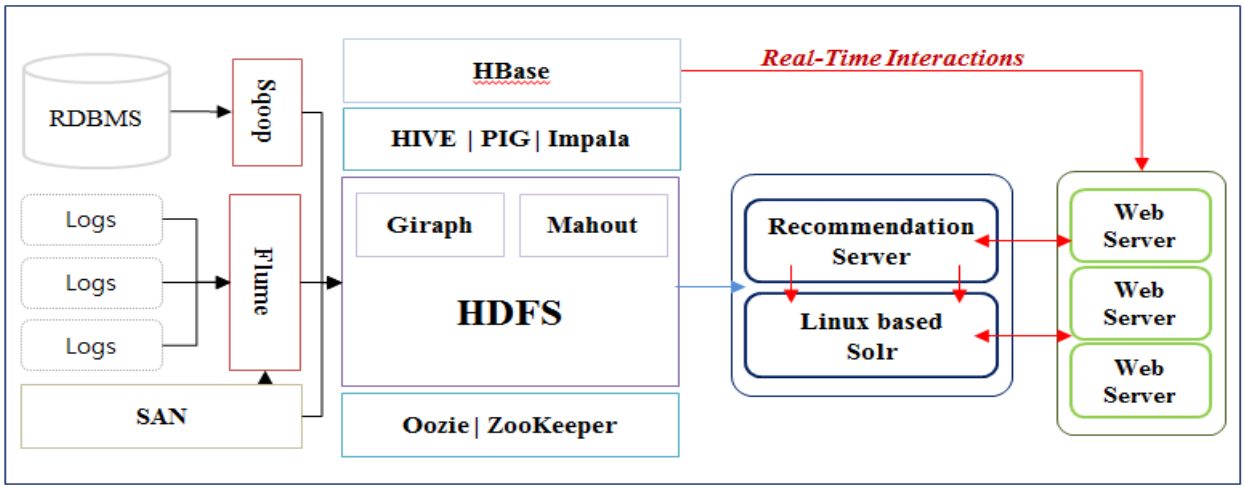

Figure. : Recommendation System Architecture[9]

To develop such a smart, scalable recommendation and search system, a cloud cluster of four servers was configured. Hadoop framework has been chosen to provide the scalability and efficient data mining. HBase distributed column-oriented database on top of Hadoop was chosen for storing data. The recommendation was based on K-N-N algorithm to find the replacement of a person leaving from the project. The competence data and project data were two input files for processing. These two data files were loaded to HDFS and saved in CSV file format. The lists of competences were grouped. Grouping was done on the basis of target use *i.e.*, . where competences can be used. Map Reduce implements this logic of clustering the competence lists into groups of competence [9].

3.1.2 Recommendation System Development

As we mention that recommendable search algorithm and architecture system, in big software organizations successful products is one of the most important aspects of organization achievement. Team formation directly affects the performance of the team and the product development quality. Staffing is an important issue to be analyzed when software development is undertaken as a value-driven business. Creating competence-based competitiveness is a great challenge because competence identification and consequently, its management, helps in innovation, decision support, faster process and product quality improvement, and constitute an important input to the creation of the firm's organizational knowledge. The finding a value driven developer-to-project assignment is a complex task. Predicting the compatibility of one with other team members is much more complex. To develop such a smart, scalable recommendation and search system, a cloud cluster of four servers was configured. Hadoop framework has been chosen to provide the scalability and efficient data mining. HBase distributed column-oriented database on top of Hadoop was chosen for storing data. Hadoop engine was deployed over the 4 node cluster and Java runtime was setup to use the common Hadoop configuration, as specified by the NameNode (master node) in the cluster. The recommendation was based on K-N-N algorithm to find the replacement of a person leaving from the project. The competence data and project data were two input files for processing. These two data files were loaded to HDFS and saved in CSV file format. The lists of competences were grouped. Grouping was done on the basis of target use *i.e.*, where competences can be used. MapReduce implements this logic of clustering the competence lists into groups of competence. MapReduce function reads input files from HDFS, processes them and creates table in HBase to store the output data in table format. Data mining involves the finding the distance between on their competence. REST services provide access to output results stored in HBase tables. It also supports basic database operations to the HBase table. New profiles of competences can be created. Existing profiles can be updated and deleted. All profiles and a profile at a time can be retrieved. Competence data and project data were saved into HBase tables by using MapReduce. As shown in Figure , competence data and project data files were loaded as CSV files in Hadoop distributed file system (HDFS). MapReduce function takes these files from HDFS as input for mapper class and maps to an intermediate <key,value> pairs. Then the reducer combines these intermediate pairs based on similar key to produce the reduced output [9].

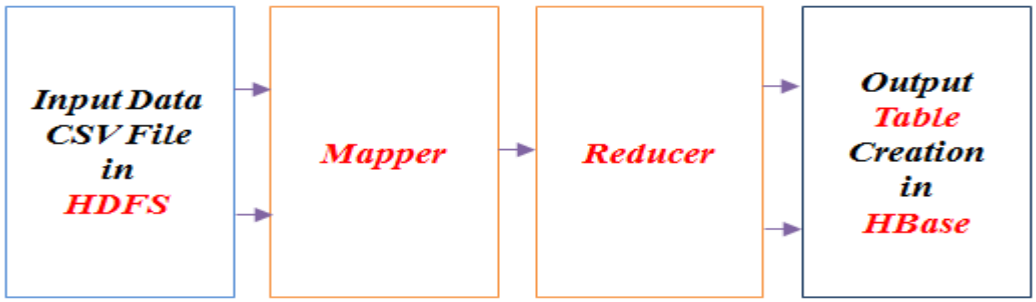

Figure 2. Competence Data Load by MapReduce

3.1.2.1 Recommendation Methods

Recommendation methods can be usually classified into three main categories [11]:

- Content-based,
- Collaborative, and
- Hybrid recommendation

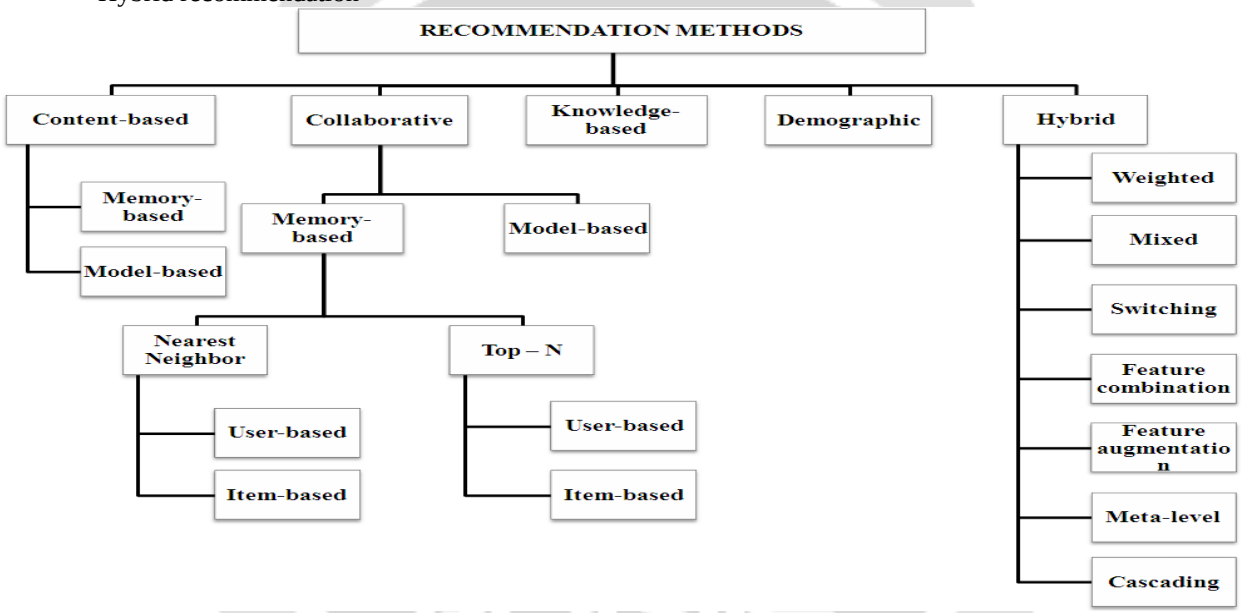

Figure :Classification of recommendation methods[13]

Content-based filtering

Content-based recommender systems work with profiles of users that are created at the beginning. A profile has information about a user and his taste. Taste is based on how the user rated items. Generally, when creating a profile, recommender systems make a survey, to get initial information about a user in order to avoid the new-user problem. The content-based approach to recommendation has its roots in information retrieval and information filtering systems. Because of the significant and early advancements made by the information retrieval and filtering communities and because of the importance of several text-based applications, many current content-based systems focus on recommending items or services containing textual information, such as documents, Web sites (URLs), and Usenet news messages. The profiling information can be elicited from users explicitly, e.g., through questionnaires, or implicitly—learned from their transactional behavior over time [11].It can either be:

- Memory/Heuristic based - uses frequency, inverse document frequency (TF-IDF) text retrieval method [13].
- Model based - uses Decision trees, neural networks, Bayesian classifiers, Clustering or vector-based representation [13].

Collaborative filtering

Collaborative filtering became one of the most researched techniques of recommender The idea of collaborative filtering is finding users in a community that share appreciations. If two users have same or almost same rated items in common, then they have similar tastes. Such users build a group or a so called neighborhood. A user gets recommendations to those items that he/she hasn't rated before, but that were already positively rated by users in his/her neighborhood. The taste is considered to be constant or at least change slowly [11].In CF based systems, users receive recommendations based on people who have similar tastes and preferences, which can be further classified into item-based CF and user-based CF. In the user-based approach the items that were already rated by the user before, play an important role in searching a group that shares appreciations with him. In item-based systems, the predicted rating depends on the ratings of other similar items by the same user [11].

Collaborative recommendations can be classified into following types [13]:

- Item-based CF - The predicted rating depends on the ratings of other similar items by the same user.
- User-based CF - The prediction of the rating of an item for a user depends upon the ratings of the same item rated by similar users.
- Memory-based (Heuristic) Recommendation Systems - They make predictions by operating on data (users, items and ratings) stored in memory.

They can be classified as [13]:

- Nearest Neighbor algorithms - Users similar to the current user with respect to preferences are called as neighbors. This algorithm deals with discovering the past users who had historically the similar taste as that of the new user. Items that the neighbors like are then recommended to the new user, as he will probably also like them. User-based nearest neighbor algorithms generate predictions for a given user based on ratings provided by users in the neighborhood. Item based nearest neighbor algorithms generate predictions based on similarities between items.
- Top-N recommendation algorithms – It recommends a set of N top-ranked items that will be of interest to a certain user. The user-item matrix is analyzed to correlate different users or items and use them to compute there commendations. User-based Top-N Collaborative Filtering Algorithms identify the k most similar users (nearest neighbors) to the active user. Item-Based Top-N Collaborative Filtering Algorithms compute the k most similar items for each item according to the similarities between them.

Hybrid recommendation approaches

Many of the existing recommender systems utilize only a handful of basic recommendation techniques like content-based, collaborative, demographic, utility-based and knowledge-based.. Using hybrid approaches we can avoid some limitations and problems of pure recommender systems, like the cold-start problem. The recommendation methods described above have performed well in several applications [11].

Various hybrids have been explored, including the hybridization techniques namely weighted, mixed, switching, feature combination, feature augmentation, and meta-level. They are described as follows [13]:

- Weighted: Implementing Collaborative and Content-based methods separately and then combining their predictions.
- Switching: A certain switching criterion is used by the system to interchange between two recommendation systems operating on the same object.
- Feature Combination: Features from different recommendation systems' data sources are put into a single recommendation algorithm.
- Cascading: For this category, one recommendation system refines the results given by another.
- Meta Level: In this case, a feature such as a model learned by one recommendation is used as input to another.
- Feature Augmentation: The output of one system is used as an input feature to another; for example, using the model generated by one to generate features that are used by another.
- Mixed: Incorporates two or more techniques at the same time; for example: Contentbased and Collaborative Filtering.

Knowledge-based Recommendations

Knowledge-based Recommender Systems suggest to users items of their interest, on the basis of some understanding of both items' characteristics and users' profiles. It utilizes the knowledge about users and products and reasons out what products meet the users' requirements. In order to properly work, this kind of recommender systems need a thorough modeling of items and users; the usual barrier to their development is, therefore, the availability of the necessary knowledge and its maintenance over time. The advantage with this is that it doesn't have the "ramp-up" problem since its recommendations don't depend

on any database of user ratings. Moreover they need a minimal amount of users, i.e. they do not require a huge amount of data to compute recommendations, differently from other classes of recommender system. The drawback with this system is that it requires an engineered knowledge database to make useful recommendations. This knowledge base has to be updated to keep up with the ever changing consumer ratings and preferences [13].

Demographic Recommendations

Demographical data such as age, gender, social class, education, location, etc. can be used and obtained explicitly and implicitly. Based on the demographical data a user profile can be created. With a demographic profile, certain matching items can be recommended. For instance teenagers prefer different products than the elderly and the rich may want different products than middle and lower classes and are willing to pay more. Therefore demographic recommendation categorizes the user based on personal attributes and makes recommendations based on demographic classes. The advantage of a demographic approach is that the user-item ratings are not used, so new users can get recommendations before they have rated any item. And knowledge about the items and their features is not needed; therefore the technique is domain independent. The disadvantage of the demographic approach is that gathering the required demographic data leads to privacy issues. Demographic classification is also too crude for highly personalized recommendations [8].

Comparison of Table [11]

Recommendation approaches	**Techniques used**	**Disadvantage**
Content-based filtering	Information Retrieval and Filtering	Limited content analysis ,new user problem, Overspecialization
Collaborative filtering	Nearest neighbour	New user and new item problem, Sparsity
Hybrid recommendation	Combine conent based and collaborative	Sometimes quality and accuracy gets affected

Table . : Comparison of Table

3.1.2.2 Big Data Platform (BDP)

Hadoop comes with a distributed filesystem called HDFS (Hadoop Distributed Filesystem). HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. The Hadoop filesystem is designed for storing petabytes of a file with streaming data access using the idea that most efficient data processing pattern is a write-once, readmany-times pattern. HDFS stores metadata on a dedicated server, called NameNode. Application data are stored on other servers called DataNodes. All the servers are fully connected and communicate with each other using TCP-based protocols [9].

1) **Architecture of BDP**

HDFS is based on master/slave architecture. To read a file, HDFS client first contacts NameNode. It returns list of addresses of the DataNodes that have a copy of the blocks of the file. Then client connects to the closest DataNodes directly for each block and requests the transfer of the desired block. For writing to a file, HDFS client first creates an empty file without any blocks. File creation is only possible when the client has writing permission and a new file does not exist in the system. NameNode records new file creation and allocates data blocks to list of suitable DataNodes to host replicas of the first block of the file. Replication of data makes DataNodes in pipeline. When the first block is filled, new DataNodes are requested to host replicas of the next block. A new pipeline is organized, and the client sends the further bytes of the file. Each choice of DataNodes is likely to be different. A HDFS cluster consists of a single NameNode (as master) and a number of DataNodes (as slaves). The NameNode and DataNodes are pieces of software designed to run on commodity machines. These machines typically run a GNU/Linux operating system (OS). The usage of the highly portable Java language means that HDFS can be deployed on a wide

range of machines. A typical deployment has a dedicated machine that runs only the NameNode software. Each of the other machines in the cluster runs one instance of the DataNodes software. The architecture does not preclude running multiple DataNodes on the same machine but in a real deployment one machine usually runs one DataNode.Affiliations are centered, italicized, not bold. Include e-mail addresses if possible. The existence of a single NameNode in a cluster greatly simplifies the architecture of the system. The NameNode is the arbitrator and repository for all HDFS metadata. The system is designed in such a way that user data never flows through the NameNode. 1) CheckpointNode - The Checkpoint node periodically downloads the latest fsimage and edits from the active NameNode to create checkpoints by merging them locally and then to upload new checkpoints back to the active NameNode. This requires the same memory space as that of NameNode and so checkpoint needs to be run on separate machine. Namespace information lost if either the checkpoint or the journal is missing, so it is highly recommended to configure HDFS to store the checkpoint and journal in multiple storage directories. The Checkpoint node uses parameter fs.checkpoint.period to check the interval between two consecutive checkpoints. The Interval time is in seconds (default is 3600 second). The Edit log file size is specified by parameter fs.checkpoint.size (default size 64MB) and a checkpoint triggers if size exceeds. Multiple checkpoint nodes may be specified in the cluster configuration file. 2) BackupNode - The Backup node has the same functionality as the Checkpoint node. In addition, it maintains an in-memory, up-to-date copy of the file system namespace that is always synchronized with the active NameNode state. Along with accepting a journal stream of the filesystem edits from the NameNode and persisting this to disk, the Backup node also applies those edits into its own copy of the namespace in memory, thus creating a backup of the namespace. DataNodes and NameNode connections are established by handshake where namespace ID and the software version of the DataNodes are verified. The namespace ID is assigned to the file system instance when it is formatted. The namespace ID is stored persistently on all nodes of the cluster. A different namespace ID node cannot join the cluster. A new DataNode without any namespace ID can join the cluster and receive the cluster's namespace ID and DataNode registers with the NameNode with storage ID. A DataNode identifies block replicas in its possession to the NameNode by sending a block report. A block report contains the block id, the generation stamp and the length for each block replica the server hosts. The first block report is sent immediately after the DataNodes registrations. Subsequent block reports are sent every hour and provide the NameNode with an up-to date view of where block replicas are located on the cluster [9].

2) Data Model of BDP

HBase uses a data model which is similar to Google BigTable's data model. The applications keep the rows of data in labeled tables. Each row has a sorting key and an arbitrary number of columns. Table row keys are byte arrays. Sorting of table rows can be done by row key and the table's primary key. Rows of one table can have a variable number of columns. A column name is of the form "<family>:<label>" with an arbitrary string of bytes. A table is created with a <family> set, known as "column families". All column family have same prefix. The family name must be string whereas the label can be of any arbitrary bytes. Column can be updated by adding new column family members as per the update request by client. A new <label> can be used in any writing operation, without any previous specification. HBase stores "column families" physically grouped on the disk, so the items in a certain column have the same particular read/write characteristics and contain similar data. By default only single row may be locked at a time. Row writes are always atomic, but single row may be locked to perform both read and write operations on that row atomically. Recent versions allow blocking several rows, if the option has been explicitly activated. The HBase data is modeled as a multidimensional map in which values (the table cells) are indexed by four keys: value = Map(TableName, RowKey, ColumnKey, Timestamp), where: i) TableName is a string; ii) RowKey and ColumnKey are binary values (Java byte[]); iii) Timestamp is a 64-bit integer (Java long); iv) value is an uninterrupted array of bytes (Java byte[]).Binary data is encoded in Base64 for transmission over the wire. The row key is the primary key of the table and is typically a string. Rows are sorted by row key in lexicographic order [9].

3.1.2.3 Recommendable Search Algorithms

The recommendation algorithm is for analyzing data for a particular problem to find the items a user is looking for and to produce a predicted likeliness score or a list of top N recommended items for a given user. Different algorithms can be used based on each use case. K-nearest neighbor has been used for finding a similar replacement of leaving a project. User-based Collaborative Filtering Algorithm is used for predicting missing competences [9].

1) **K-Nearest Neighbor Algorithm (KNN Algorithm)**
The K-nearest neighbor algorithm (KNN) is part of supervised learning that has been used in many applications in the field of data mining, statistical pattern recognition and many others. KNN is a method for classifying objects based on the closest training examples in the feature space. An object is classified by a majority vote of its neighbors. K is always a positive integer. The neighbors are taken from a set of objects for which the correct classification is known. It is usual to use the Euclidean distance, though other distance measures can also be used instead. One of the advantages of the KNN is that it is well suited for multi-modal classes as its classification decision is based on a small neighborhood of similar objects . So, even if the target class is multi-modal (*i.e.*, consists of objects whose independent variables have different characteristics for different subsets), it can still lead to good accuracy. A major drawback of the similarity measure used in the KNN is that it uses all features equally in computing similarities. This can lead to poor similarity measures and classification errors, when only a small subset of the features is useful for classification [9].

2) User-based Collaborative Filtering Algorithm
A user-based collaborative filtering algorithm produces a recommendation list for the object user according to the view of other users. The assumption is that users with similar preferences will rate products similarly. Thus missing ratings for a user can be predicted by first finding a neighborhood of similar users and then aggregating the ratings of these users to form a prediction. User rating data can be a matrix *A(m,n)*, where m represents the number of users, n represents the number of items [19]. Element *R* (at the *ith* line and *jth* column) represents the rating of the item j rated by user *i*. A cosine similarity algorithm can be used to measure the similarity between user *i* and *j*. Similarity between user *i* and user *j* is sim *(i, j)*: $sim(i, j) = cos(i, j) = i * j / ||i|| * ||j||$, Rating of item *i* rated by user *u* can be predicted by rating of nearest neighbors set *NBSu* rated by user *u*. *NBSu* is to the nearest neighbor set of user *u*;, *Pu,i* have predicted rating by user *u* on item *i*;, and *sim(u,n)* is to the semblance between user *u* and *n*. According to the rating of items, select N items that have the highest rating to compose a recommendation set and recommend them to the object user [9].

3) REST Services
REST is an architectural style which is based on Web standards and the HTTP protocol. In REST-based architecture everything is a resource. A resource is accessed via a common interface based on the HTTP standard methods. In REST architecture there is a REST server which provides access to the resources and a REST client which accesses and modifies the REST resources. every resource should support the HTTP common operations. Resources are identified by global IDs (which are typically URIs). REST allows that resources have different representations, *e.g.*, text, XML, JSON *etc* [20]. The client can ask for specific representation via the HTTP protocol (Content Negotiation). The HTTP standard methods which are typically used in REST are PUT, GET, POST and DELETE [9].

4. CONCLUSIONS AND FUTURE WORK

Recommender systems are a powerful new technology for extracting additional value for a business from its user databases. These systems help users find items they want to buy from a business. Recommender systems benefit users by enabling them to find items they like. Conversely, they help the business by generating more sales. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web 2. New technologies are needed that can dramatically improve the scalability of recommender systems. As one of the most successful approaches to building recommender systems, collaborative filtering (CF) uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users. In this paper we explored CF-based recommender systems and compared it with other algorithms such as MAE, RMSE, and Correlation. Our results show that
item-based techniques hold the promise of allowing CFbased algorithms to scale to large data sets and at the same time produce high- quality recommendations. This is similar to RMSE which performed better than MAE and correlation. Notwithstanding, CF uses RMSE method in making recommendation too.
However, for future works on large datasets, we recommend that SparkR be used since it is a distributedsystem. Spark framework is very easy to use more than Hadoop and the code is not complex, easy to read. The high speed and scalability of algorithms built on this system is good because it is embedded on Spark memory.SparkR can work faster for large datasets projects that require parallel solution but the scalability can be sublinear,and very important characteristics are readability, easy deployment on many platforms because it is a distributed system and

maintainability. It is also good for creating proofs of concepts, because creating a parallel program with Spark is nearly as easy as writing a sequential one.

5. REFERENCES

[1] Mr.Nilesh Sherla, Prof.Kiran Joshi, Prof. Sowmiya Raksha," Recommendation System using Keyword Base Approach", IARJSET,ISSN (Print) 2394-1588, Vol. 2, Issue 3, March 2015.
[2] J. Amaithi Singam and S. Srinivasan," Optimal Keyword Search For Recommender System in Big Data Application", ARPN.ISSN 1819-6608, VOL. 10, NO. 7, APRIL 2015.
[3] Udeh Tochukwu Livinus1, Rachid Chelouah2 and Houcine Senoussi, "Recommender System in Big Data Environment", IJCSI, Volume 13, Issue 5, September 2016.
[4] Vinaya B. Savadekar , Pramod B. Gosavi ," Towards Keyword Based Recommendation System", IJSR, ISSN (Online): 2319-7064, Volume 3 Issue 11, November 2014.
[5] Nita Stefania Loredana," On Recommendation Systems Applied in Big Data",IEEE,2016.
[6] David C. Anastasiu, Evangelia Christakopoulou, Shaden Smith, Mohit Sharma and George Karypis," Big Data and Recommender Systems",ACM, 2015.
[7] Kim, Jinhong;Hwang,Sung-Tae," A Study of Recommendation System for Big Data Environment ", American Scientific Publishers .
[8] Kim, Jinhong;Cho,Leesang ," Toward of Conceptual Recommender Service for Big Data Platform ", Advanced science letters ,2016 .
[9] Jinhong Kim and Sung-Tae Hwang, " Big Data Platform of a System Recommendation in Cloud Environment", IJSEIA, ISSN: 1738-9984,2015.
[10] Ajit Gaddam," Securing Your Big Data Environment".
[11] Shakhy.P.S, Swapna.H ," Improved Keyword Aware Service Recommendation System for Big Data Applications", ISSN (Print): 2320-9798, IJIRCCE, Vol. 3, Issue 8, August 2015.
[12] Zeynab S. Kalantarian, Rada Mashayekhi, Ali Abdashahi," GFCM: A Gossip-Based Fuzzy C-means Algorithm",IEEE, 2014.
[13] Venkatesh P," A Survey on Algorithms Used in Recommender Systems", ISSN (Print) : 2320 – 9798, IJIRCC, Vol. 4, Issue 6, June 2016.
[14] Allan Vidotti Prando, Solange Nice Alves de Souza," Modular Architecture for Recommender Systems Applied in a Brazilian e-Commerce", Journal of Software ,March 9, 2016.