Integrated Semantic Web Usage Mining with Fuzzy Technique to Improve Accuracy of Recommendation System

THESIS

SUBMITTED TO

BABASAHEB BHIMRAO AMBEDKAR UNIVERSITY

LUCKNOW

BABASAHEB
BHIMRAO
AMBEDKAR
UNIVERSITY

LUCKNOW

प्रज्ञा शील करुणा

ESTABLISHED 1996

FOR THE DEGREE OF

Doctor of Philosophy

IN

COMPUTER SCIENCE

Submitted by

Bhupesh Rawat

Enrolment No. 966/14

Supervisor

Prof. Sanjay Kumar Dwivedi

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL FOR INFORMATION SCIENCE & TECHNOLOGY

BABASAHEB BHIMRAO AMBEDKAR UNIVERSITY

(A CENTRAL UNIVERSITY; NAAC- 'A' GRADE)

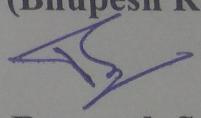
VIDYA VIHAR, RAEBARELI ROAD, LUCKNOW-226 025 (U.P.), INDIA

2018

CANDIDATE'S DECLARATION

I hereby declare that I have completed research work for the full time prescribed and that the thesis embodies the results of my investigation conducted during the period I worked as Ph.D. research scholar. I further declare that to the best of my knowledge the thesis does not contain part of any work submitted for the award of any degree either in this Institute/University or any other Institute/ University. I also declare that the thesis is essentially free from all kinds of plagiarism.

(Bhupesh Rawat)



Research Scholar

CERTIFICATE

This is to certify that the thesis titled "**Integrated Semantic Web Usage Mining with Fuzzy Technique to Improve Accuracy of Recommendation System**" submitted by **Mr. Bhupesh Rawat** is an original research work and has not been previously submitted in part or full for the award of any other degree or diploma to this or any other university.

The thesis submitted to Babasaheb Bhimrao Ambedkar University, Lucknow Satisfies all the requirements as stipulated in the *Doctor of Philosophy (Ph.D.) regulations -1999 as amended in 2013* and it is fit for submission and evaluation for the award of the degree of Doctor of Philosophy of the University.

Date: 28/11/18

Supervisor

Head of Department

Abstract

With the abundance of information available on the web it is becoming increasingly difficult for users to find the items which best match with their preferences. The items to be recommended could be such as 'what course to choose', 'what movie to watch', and 'what book to read' among others. This is commonly referred to as 'information overload' problem. In this context, recommender systems help user to find the most appropriate items based on their preferences. Recommender systems have been used successfully in a wide variety of domains such as e-commerce, e-government and e-resources among others. There are several recommendation techniques which are found to be useful in making recommendations in several domains such as book, movie, and news article among others. The choice of a particular technique depends on the type of data available with the system. For example, if the 'user item rating matrix' is available then the 'collaborative filtering technique' is most appropriate and if the items' features are available then 'content filtering approach' is most suitable. Majority of existing recommender systems are based on content filtering technique, collaborative filtering technique and their combinations. Content filtering approach recommends item (learning objects/learning actions) based on the profile of a user which consists of learners' preferences for a set of items. On the other hand, collaborative filtering systems take into account the opinion of other users in order to find similarity of a group of users with an active user.

In addition to 'information overload' issue, the existing recommender systems are also suffering from the issue of *'one size fits all'* which means that the same course/learning resources are being offered to all the learners without taking into account their differences in terms of their level of knowledge, skill, and interest. This problem could be addressed by building the profile of a learner which consists of learner's preferences. The profile is then used by a recommender system to make recommendations to users.

Furthermore, we have also found that the existing clustering algorithms do not employ '*cluster validation mechanism'* in order to evaluate the quality of clusters produced by these algorithms. This leads to the formation of clusters which do not represent the actual profile of a user and are not consistent with the actual number of clusters present in the given dataset. Moreover, the applications which use these clusters for recommendation of items might end up recommending items which do not match with the learners' profile. In order to deal with this issue, we have suggested cluster evaluation methods which ensure that, the clusters created are of high quality.

In addition, existing recommender systems based on collaborative filtering technique suffer from *sparsity issue*. The sparisty is caused by insufficient ratings in the user item rating matrix which prevent a recommender system from making good quality of recommendations. Furthermore, traditional recommender approaches were not able to fully explore the semantic tools which could be exploited in order to learn more about learners' preferences. With the emergence of semantic web, a large number of semantic tools and techniques have become available which can be exploited in order to elicit additional preferences of learners. In this context, we aim to enrich the user item rating matrix by utilizing resource description framework and Apache Jena rules in order to improve the accuracy of recommendations.

The thesis proposes a course recommender framework which recommends different data mining courses to learners based on their profile. Experiments were conducted on the real world dataset in order to evaluate the performance and accuracy of the proposed framework which shows that the framework is able to improve the accuracy of recommendations significantly. The results are also significant from a learners' point of view as a learner is getting more relevant courses with higher precision and lower error rate. This helps a learner in improving his subject performance and overall academic performance as well.

ACKNOWLEDGMENTS

First and Foremost I would like to Thank **God.** You have given me the power to believe in myself and pursue my dreams. I could never have done this without the faith I have in you, the Almighty.

The award of degree Doctor of Philosophy is one of the hardest deserving achievements. People struggle for it and achievement not easily found. During the entire research works some valuable people conceived their enormous positions in my heart. In this regard, I am grateful to the University and express my deep sense of gratitude to its **Hon'ble Vice-Chancellor** for delivering this great opportunity to me.

I would like to extend my hearty thanks to my supervisor **Prof. Sanjay K. Dwivedi, Department of Computer Science, Babasaheb Bhimrao Ambedkar (A Central University), Lucknow**, for support and mending efforts along with the valuable advice and encouragement through each step, for many lessons on how to do research and write research papers, for being very supportive in my work, for guiding into each part of the research work and life in general. His genuine concern inspired me to give my best and his insights were helpful in looking at the problem from different viewpoints. Specifically, I am thankful for countless hours he spent with me in explaining each part, sharing his experiences on his research. Also, I am thankful for his insightful suggestions that helped me to make the right strategic choices at many crucial decision points along these years. I can never ever forget his contributions in shaping my life. It is all because of his infinite inspiration and contribution, that I am able to present this piece of work in a set tone and style. I am fortunate and feel pride in having his guidance.

I convey my sincere thanks to **Head** and all other faculty members of Department for their motivation and support during the research. I would also like to show my gratitude to the Department of Computer Science, for providing the healthy and pleasant environment required for good quality research. I would like to thank all administrative and supporting staffs of the University for providing the comfortable environment and help.

Specially thank should be given to financial supporting body, University Grants Commission that provided me fellowship for this research work.

I am heartily grateful to my senior research group members Dr. Parul Rastogi, Dr. Ajay Kumar Bharti, Dr. Pramod. Shukdeve, Dr. J.N.Singh, Dr. Rajesh Gautama, Dr. Anand

Kumar, Dr Vaisali Singh and Dr Ganesh Chandra for their great help and time to time constant encouragement throughout the research.

A special thanks to my family, my dearest parents, and sister. Words cannot express how grateful I am to my **PARENTS and Wife** for all of the sacrifices that you've made on my behalf. Your prayer for me was what sustained me so far.

I cannot list the names of all people who I indeed to but thanks to all valuable persons, who have given me enormous support and inspiration directly or indirectly during my research work.

Date: 28-11-18

Place: Lucknow

Bhupesh Rawat

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

With the development of the large number of sophisticated e-learning environments which characterize the huge information, the strong interactivity, the great coverage and no space-time restrictions, personalization has become an important feature in e-learning systems [11][18]. An e-learning system has a large number of learners which have differences in background, need, goal, and level of knowledge among others. Personalized learning occurs when e-learning systems are designed according to the profile of learners which consists of their preferences. Personalization can be achieved by using predefined rules that sequentially offer to learn resources in a specified learning path [119]. It can also be achieved by using heuristic rules, user model, and recommendation techniques [120].

1.1 Recommender Systems

Recommender systems are software tools and techniques providing suggestions for items of interest to the user [14] [180][158]. The suggestions could be such as ‘what item to buy', ‘what movie to watch', ‘what book to read', and ‘what news to read' among others. Here, an item is a general term which may represent what recommender systems recommend to a user. Recommender systems normally focus on a specific type of item such as news, book, and video among others. Based on this, its design, its user interface and recommendation approaches employed produces useful and effective recommendations for that specific type of item. Recommender systems are primarily designed for those users who have not used or experienced an item or set of items presented on a site such as Netflix or Amazon. Some of the widely known sites which have successfully employed recommender systems as part of their business are Amazon (Amazon.com) where one can buy a range of items particularly books online, CDNow (CDNOW.com) which is known for selling music CD online, EBay (Ebay.com) which can be used for buying and selling products, Reel (Reel.com) which sells movie online, Movielens which rents movie online, YouTube, Pandora, Social Networking Sites, Yahoo, Tripadvisor among others. The underlying technology in recommender systems used by Amazon and YouTube can’t be employed directly in ‘e-learning recommender systems’ as the cognitive state of a learner, learning content and context might change over time [121]. Even two learners with the same interest and taste could have

different proficiency, learning objective and context. There are various recommendation techniques for recommending online learning activities to learners based on their profile[96].

Recommender systems in e-learning domain assist learners in discovering relevant learning resources that match with their profile at the right time, in the right context and in the right way, keep them motivated and enable them to complete their learning activities in an

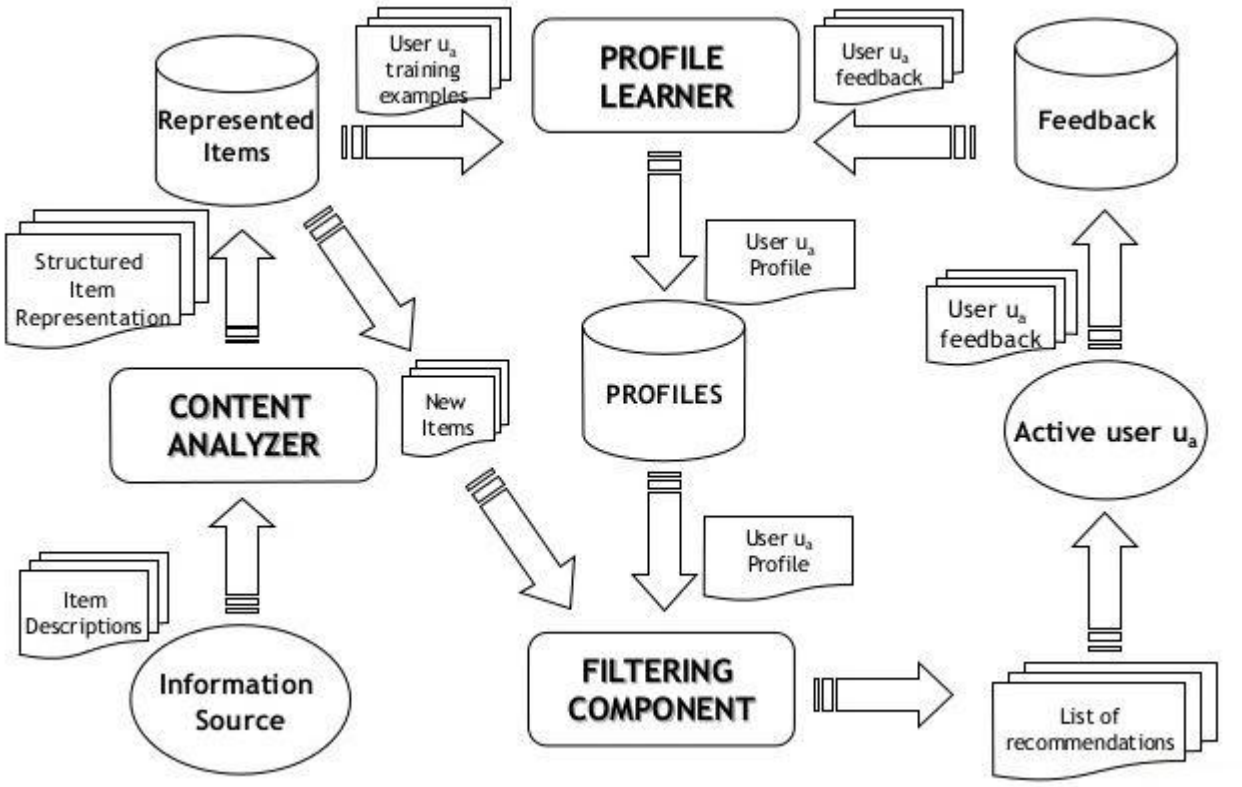


Figure 1.1 Block Diagram of Recommender System [230]

effective and efficient way [122]. E-learning recommender systems have the potential to provide 'learner-centred learning' and are designed based on the learners' needs, abilities, preferences, and styles rather than providing same learning resources without taking into account individual needs and differences[2][3][4].

1.2 History of Recommender Systems

The roots of recommender systems lie in cognitive science [66], information retrieval [155] and approximation theory [156]. Recommender systems emerged in 1990s in order to provide personalized services in e-commerce. Some of the early recommender systems reported in the literature were inspired from the creatures such as ants and cavemen. The first commercial

recommender system based on collaborative filtering was Tapestry [205] which was developed at the 'Xerox Palo Alto Research Centre'. Conventional recommender systems such as collaborative filtering used to perform their function manually which took more time and was also ineffective and inefficient. As a result of this, automotive collaborative algorithms were developed which reduced human involvement and were effective as well. One of the finest application examples of automatic collaborative filtering is 'GroupLense Project' [206] in which users rated items and based on this ratings, users are correlated with others similar users and personal prediction made for unrated items.

One of the issues with traditional recommender systems was that, their database was static, which was ineffective, if the goals and requirements of users changed. This results in seeking information dynamically from users, through 'queries presented in real time'. In this context, TF/IDF (term frequency and inverse document frequency) is a popular technique for discovering the frequency of term in a document and then rank the document based on this frequency.

User models were developed to represent users' information needs which were mostly 'hand created profile'. Later on, with the development of 'machine learning algorithms' such as k-means and Bayesian among others, it became easy to build the profile of a learner using 'machine learning algorithms'. Moreover, the users' profiles were updated using feedback obtained from users over items that were recommended. The profile consisted of only user or item information in order to make recommendation. However, with the emergence of World Wide Web, a large number of social networking sites have emerged which allow user to add comments and annotate resources among others which have been used in many of the existing recommender systems in order to improve the quality of recommendations.

1.3 Features of Recommender Systems

Although features of recommender systems vary from one approach to another, below we discuss some of the common features found in all recommender systems:

1.3.1 Acquisition of Information

A recommender system needs some sort of data about user or item to begin with in order to make recommendations. All this information is stored and represented by the profile of a user. Using this profile a recommender system knows the taste or preferences of a user and is

able to recommend relevant items accordingly. Users' preferences can be collected explicitly or implicitly. In explicit collection of information, users are asked to provide feedback over set of items that they have consumed. The feedback is usually in the form of rating which is measured on a scale of 1 to 5 where, 1 refers to 'least liked item' and 5 indicates 'most liked items'. There are mainly three types of ratings used to collect feedback from a user. The first one already discussed above. The second one is binary ratings which are often seen in e-commerce sites. Such ratings help us to know whether a particular item is purchased or not. If an item is purchased or viewed then an entry of '1' is recorded in the server else '0' is recorded. The third type of rating is 'unary'. One of the widely used examples of this type of ratings is like button in the page of face book. If the button is clicked, then user has liked the item.

Another way to collect feedback from user is through implicit feedback. In this method, direct information is not sought from user rather their actions and behaviour are observed. For example, in a news site, if a user stayed on the page for 1 minute then it can be inferred from this information that the user is interested in the article and similar article could be recommended to the users based on this information.

1.3.2 Processing of Recommendations

System processes user action in batch. User preferences are generated out of user actions which includes 'time spent on a page', clicks and hovers on important DOM elements, click sequence such as Product view→add to cart-> checkout-> Bookmarking->wish listing-> social sharing. Finally system processed user preference data for generating recommendations. In this step a recommendation algorithm or combination of algorithm could be used based on the nature of application and type of data.

1.3.3 Generating and Serving Recommendations

In this step user performs some action on client side, the system sends this information to recommendation algorithm. System decides on the type of algorithm being used based on the information of user. Finally, system generates list of recommendation based on this decision. As not all the generated items may be useful for a user, so the task of ranking of items and finally filtering them is performed. This filtered list of items is displayed to a user.

1.3.4 Feedback and Refinement

The feedback is sought on the items that have been recommended. This feedback is further used to refine the list of items. System analyses type of recommendations that are more likely to be clicked. Finally, system updates this information in order to reflect this analysis.

1.4 Types of Recommender Systems

Traditional recommender systems are based on content based filtering, collaborative filtering and their different combinations. The use of a particular recommender system depends upon the type of data available about user and item. For example, if 'user item rating matrix' is available then it would be better to employ 'collaborative filtering approach' in order to get best recommendations results. On the other hand, if 'item's features' are available in place of user item ratings then it would be better to use 'content filtering systems'. Moreover, some new category of recommender systems are also discussed which are making use of semantic web and machine learning algorithms.

The following section discusses both the traditional and the non traditional recommendation systems with the aim of providing the information on the type of input data that goes into these systems and different techniques that make it possible to generate recommendations:

1.4.1 Content-Based Filtering (CB)

Content-based recommender systems predict items by using their features rather than ratings of other users [30][39][80][230][98]. They consist of "Content analyzer", "Profile learner", and "Filtering component. The inputs to the "content analyzer" are web pages, documents and product description. The content analyzer aims to perform initial pre-processing on unstructured data (i.e text) to prepare it for the next processing step. The "profile analyzer" takes input from the "content analyzer" in the form of "structured item representation" and generates general user profile using machine learning algorithms. This component also requires users' feedback for updating users' profile in order to recommend more relevant items. The "filtering components" takes user profile as input for recommending relevant items to the user by matching their profile with the items to be recommended.

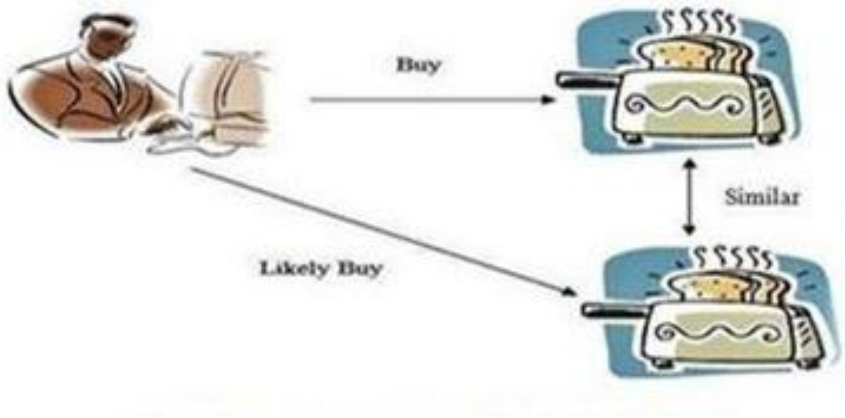


Figure 1.2 Content Based Recommender System

Content-based approaches are unaffected by ‘sparsity’ and ‘gray sheep problems’ due to their independence from “user-item ratings". Moreover, they can be utilized in domains such as movies, news article, web page, and television program. CF approaches are categorized into "case-based reasoning” (CBR) and "attribute-based techniques"(ABT). CBR recommends those learning objects which are strongly correlated to the objects which a user liked in the past. However, it suffers from overspecialization and sparsity problems. Overspecialization occurs due to recommending similar items to a user repeatedly. CBR is more useful in teaching and learning field. For example, Pixed (Project Integrating eXperience in Distance Learning) is an implementation of CBR which is a hypermedia ontology-based system.

Bayesian approach is used due to its reduced complexity in Bayesian calculations. Content filtering systems based on neural network are gaining popularity due to their ability to differentiate between two different classes of items rapidly and accurately.

1.4.2 Collaborative Filtering (CF)

Collaborative filtering (CF) is one of the most widely used approaches in the recommendation system [24][31][32][40][55][152]. They recommend items by matching the preferences of a target user with similar users (also called neighborhood of the target user). The inputs to collaborative filtering systems are a set of all users which is represented by the symbol ‘U’ and a set of all possible items which is denoted by the symbol ‘I’. The goal of CF is to predict the rating of item i by a user u which is denoted by R(u,i). User’s ratings are stored in a two-dimensional "user-item rating matrix". The output of collaborative filtering system is a set of items (i_1, i_2, i_3……i_k) where k<=I which are ranked and top-N items are

suggested to the user. CF-based systems are classified into 'memory based' and 'model-based'.

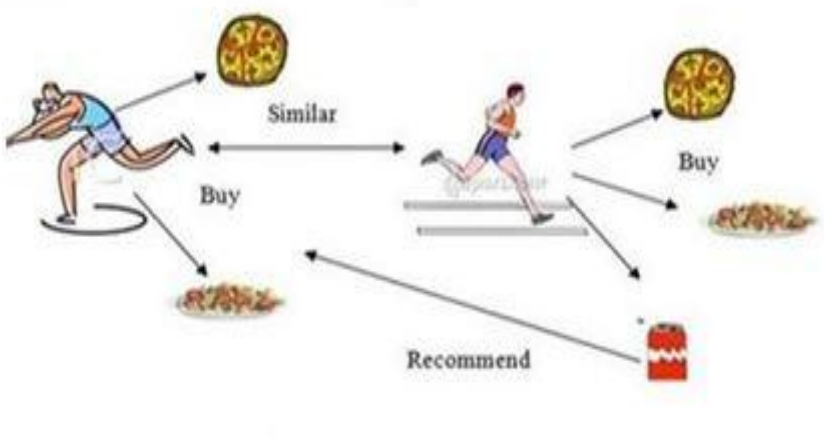


Figure 1.3 User Based Recommender System

1.4.2.1 Memory Based Filtering

The memory-based methods compute the similarity of an active user with similar users without creating a model [135]. In memory-based collaborative technique 'user-item rating matrix' is directly used in order to predict ratings for new items. This can be done in two ways: a) user based recommendation b) item-based recommendation. Some of the most popular recommender systems based on this approach are GroupLens[123], BellcoreVideo [124] and Ringo[125] among others which evaluate the interest of a user *u* for an item *i* based on the ratings for this item by other users, called *neighbours* that are similar to this user *u*. The neighbors of *u* are typically those who have similar ratings as that of *u*. They make use of the Bayesian network, clustering, and rule-based approaches to improve the quality of recommendation [136]. On the other hand, 'item based approaches' predict the ratings of a user *u* for an item *i* based on the ratings of *u* for items similar to *i*. In such approaches, two items are equal if several users of the system have rated these items in a similar fashion. Recommender systems using memory-based collaborative filtering are not fast and scalable which is required in real-time systems hence, the model based approach was developed.

1.4.2.2 Model Based Filtering

In contrast to memory based approaches, model-based approaches use ratings to learn a predictive model. The general idea is to model the 'user-item interactions' with factors representing latent characteristics of user and item in the system. This model is then trained

using the available data and then used to predict ratings of a user for new items. Some of the widely used approaches based on this approach are Bayesian Clustering [126], Latent semantic analysis [127], Support vector machine [128], and Singular value decomposition [129] among others.

1.4.3 Hybrid Filtering

Hybrid filtering based recommender Systems combines two or more recommendation techniques such as collaborative filtering (CF), content-based (CB) or knowledge-based (KB) in order to utilize the strength of one technique in order to compensate the weakness of the other so as to improve the overall quality of recommendations [9][12] [51] [139][173][144]. For example, CF relies on user ratings for predicting the ratings of a new item to be recommended. On the other hand, CB approach does not depend on ratings from a user and based its recommendations on the features of items. Several combinations can be created by combining the basic recommendation techniques [140][133][82].

Although hybrid recommender systems have been successfully employed in wide variety of applications, they have the following issues:

1. The constraint with switching hybrid (one of the forms of hybrid techniques) is to have prior information of confidence value or external criteria.
2. Another issue with hybrid filtering based recommender system is to produce the new rank score based on the individual ranks generated by each component. One of the solutions to this problem are simply adding the individual ranks such as KB_rank (2) + CB_rank (4)->Mixed_rank(6).

Hybrid filtering systems require various inputs such as user profile (which stores users' preferences about item) and contextual parameters such as (time of day, time of year), demographic data (age,location, gender, etc.) or community data, item or product features (title, genre, actors), different knowledge models (such as rule-based, Bayesian network, etc.). The output of HBRS is a list of items (i1, i2, i3…in) with their predicted score such as (0.8, 0.3, 0.5). HBRS are mostly used in the domain of e-government, e-resource, and e-tourism. Among several machine learning algorithms, ANN (Artificial neural networks) has shown good results in combining output from various recommendation models. Another approach used is an ensemble of a classifier in building hybrid recommender system.

1.4.4 Web Usage Mining

Web mining is the application of data mining techniques to the content, structure, and usage of Web resources [35][52][53][56][95]. It is broadly categorized into web content mining, web structure mining and web usage mining. Web content mining concentrates on the content of individual pages, which is contained in the HTML (Hypertext markup language), script and code that generates a page.

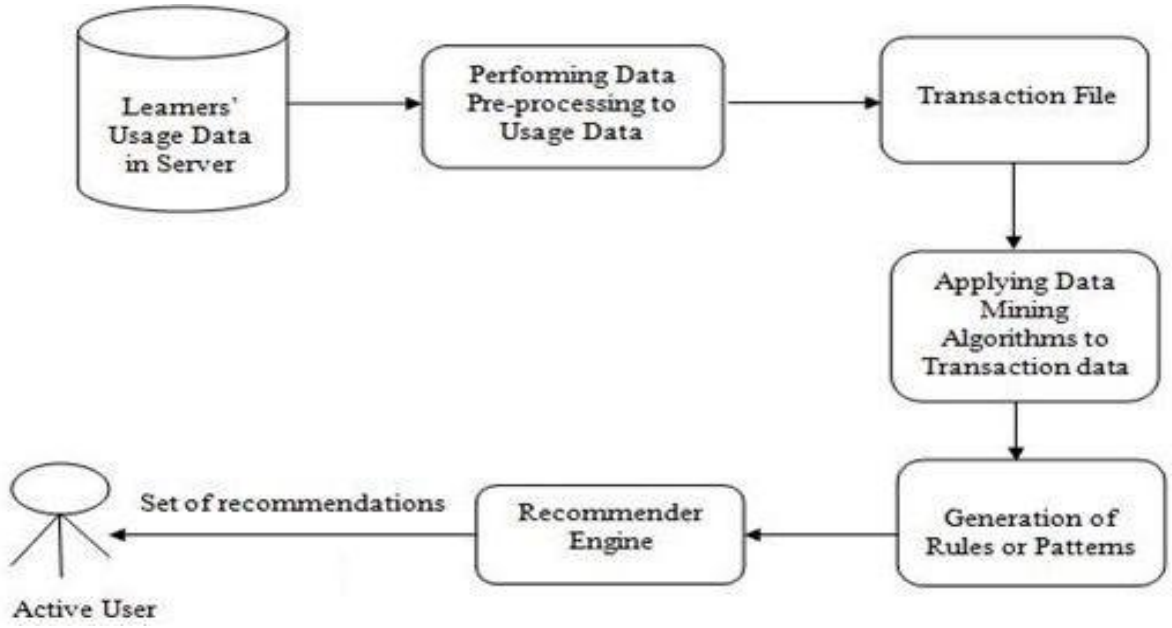


Figure 1.4 Block Diagram of Web Usage Mining

Web usage mining refers to the use of data mining techniques to automatically discover and extract useful information from the web document and services [93][145]. It is becoming more and more popular among users due to easy access to the log file as well as its applicability in CRM (Customer relationship management). The goal of CRM is to provide not just the quality products and services but also make, retain and grow customers. This is possible once we know the needs and preferences of the customers by applying data mining techniques to customers' usage data in order to build the profile of customers. In web usage mining, the primary web resource is a record of the requests made by visitors to a Web site, most often collected in a Web server log[138][151]. The major application areas for WUM are personalization, system improvement, site modification, business intelligence and usage characterization [138].

Majority of recommender systems based on web usage mining employ association rules, sequential patterns [94] and clustering [95]. Moreover, web content or site structure can be

integrated with usage data in order to improve the accuracy of the personalization system[95]. Web usage mining primarily collects its data from the log file of a web server which contains all the users' data visiting the site such as IP address, pages visited and visiting time among others. It collects, analyzes and processes this data to discover user access patterns. The result of this analysis provides us with useful insight into interesting usage information and patterns which can be further used to predict the web pages a user is likely to be interested.

1.4.5 Semantic Web Usage Mining

Semantic web aims to make the web content understandable to humans as well as machines, hence allowing the software agent to search for desired content, share information and knowledge in a format that other software agent can understand [38][41][141][101].

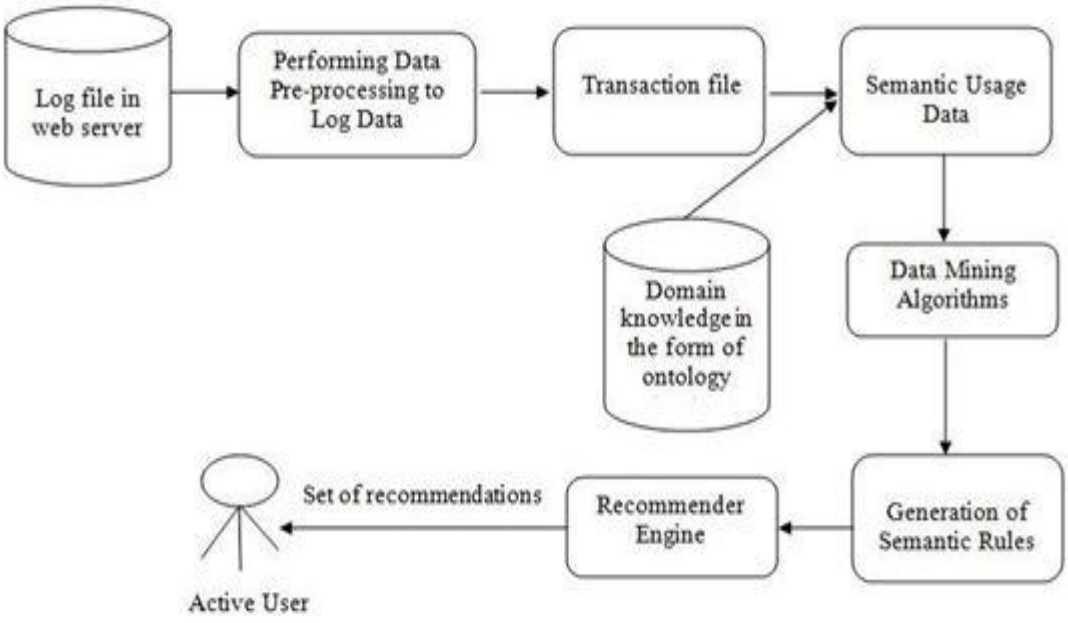


Figure 1.5 Block Diagram of Semantic Web Usage Mining

Recommender systems based on semantic web usage mining require various inputs such as “server log”, “web page” and “domain information”. The server log and web pages are fed to the usage processing step which outputs “semantic user transactions”. Ontology processing is also performed which results in domain ontology and acts as input to the subsequent phases such as pattern discovery and online recommendations. “Semantic user transactions” and “domain ontology” are fed to the pattern discovery phase which outputs “semantic usage patterns” which are finally inputted to online recommendation engine for generating recommendations to the active user.

1.5 Issues in Recommender Systems

Although recommender systems have gained wide popularity due to their widespread applications in a variety of domains such as e-governance, e-learning, and e-resources among others, they are yet to address several issues such as cold start, sparsity, scalability and overspecialization among others. Available literature highlights some critical issues in recommender systems which are discussed below:

1.5.1 Cold Start Problem

Cold start problem occurs when recommender systems don't have enough ratings either of a user or item to make recommendations to a user. This is due to either the user has not rated items or a new item was added to the site recently which makes it difficult to provide high-quality recommendations. The cold start problem is of three types: New user problem, new item problem, and new system problem. The new user problem occurs when a user has not used an item from the repository. The new item problem arises when a new item is added to a site without any ratings from a user on that items and the new system problem arises when the system is new and has just started to operate and does not have any ratings either from the user or the items. The cold start problem has its origin in recommendation system [130][131][118]. One of the most challenging tasks of the recommendation system is to improve the recommendation accuracy for the new user (inactive) or new item (rarely rated item). Cold start problem causes missing ratings in 'user-item rating matrix'. Furthermore, missing ratings could also be filled with their default values such as the middle value of average user or item [126].

1.5.2 Sparsity

Sparsity refers to insufficient information in the "user-item matrix" (R) for making recommendations [20][27][28][149][150]. With the huge number of items and users in many large-scale applications, the rating matrix can still be extremely sparse even when many events have been recorded which leads to the situation where the similarity between two given users is zero, rendering collaborative filtering useless. Due to sparsity, a rule-based recommender system cannot give any recommendations. Sparsity occurs because some users' rate only a very small fraction of the available items, hence making it difficult to find a

sufficient number of common items in multiple user profiles. This problem often occurs in web usage mining and collaborative filtering as compared to other recommendation approaches.

1.5.3 Scalability

With the increasing number of users and items, the existing recommender systems need more resources such as CPU and memory for processing of large information[92]. Most of these resources are used for identifying users with similar taste and descriptions. This problem is solved by considering the combinations of various types of filters and system physical improvement. Scalability is usually computed by increasing the size of the dataset and observing how the speed and resources consumption behaves.

1.5.4 Over Specialization

In over specialization a 'user model' solely relies on tags in his rated items [153][86]. This makes it difficult to recommend new items other than the user has previously rated. For example, if a user is interested in "action movie" but not "comedy movies" then he/she will never get recommendations for the movie which is a mix of action and comedy. Users are restricted to receive recommendations which strictly match with the information in their profile. Overspecialization prevents the user from discovering new items and various available options other than suggested by his profile. However, the diversity of recommendation is an important feature of all recommendation system. The problem is solved using genetic algorithms and provides a wide range of alternatives [154].

1.5.5 Dependency on Usage Data Alone

Recommender systems based on 'web usage mining' suggest items using historical data/transactional data found on the server due to which they miss certain objects recently added to a site which have not been consumed by a user. Moreover, without using the semantic knowledge they cannot recommend certain complex objects using their properties. However, incorporation of semantic information in the different phases of the web usage mining process allows a recommender system to infer new semantic relationships which were not obvious otherwise. For instance, in "pure usage-based e-learning system", a learner could

be recommended "Advance operating system" course based on his profile. However, with the integration of semantic information with usage data we could suggest learner some "introductory level operating system course" in order to enable the learner to acquire some basic knowledge of the course before opting for the "advanced operating system" course or could also suggest him a course instructor. Such type of semantic relationship could only be discovered by enriching usage level transactions with semantic information.

Usage-based recommendation systems can't explain the reason behind a particular set of recommendations. However, the use of the semantic web in the recommendation process has successfully overcome this limitation by providing the domain knowledge representation language such as DAML+OIL [99], SHIP (DL) and description logic [100].

1.6 Motivation and Research Gap

With the emergence of the World Wide Web, academic institutions worldwide have started to use the web in their learning process in the form of e-learning. E-learning is an acronym for electronic learning which makes learning content available for learners to access from anywhere and at any time. It is broadly inclusive of all the forms of education that make use of technology and multimedia. E-learning systems provide huge benefits in comparison to traditional education systems [195]. Some of the major benefits of these systems are the flexibility of time and space which means that a learner could gain knowledge or skill through e-learning irrespective of location and time.

Furthermore, a large number of learners are using e-learning systems to meet their learning goal. However, the resources provided by the existing e-learning systems are overwhelming and learners need assistance in selecting the right kind of learning resources which satisfy their educational goals. This problem is termed as 'information overloading' [54] which is mainly caused by rapid growth of information on the web. In this context, recommender systems have been developed in order to deal with this issue. Although recommender systems have been hugely successful in the domain of e-commerce, they have begun to use in other domains as well such as e-governance, e-resources, e-business, and e-learning [75]. While there are quite a large number of e-learning systems making use of recommender systems for recommending learning resources to learners, a relatively less amount of work is reported in course recommendation. Moreover, the existing course recommender systems make recommendations based on usage data alone and the use of 'semantic knowledge' in the recommendation process is very limited.

Existing literature suggests that there are relatively fewer 'course recommender systems' based on one or more combination of the following techniques: collaborative filtering, matrix and tensor factorization, content-based techniques, association rule mining and clustering among others [15]. While majority of recommender systems use traditional recommendation techniques, a very few of them have used non-traditional techniques such as semantic web, and fuzzy linguistic modelling among others.

One of the major problems with existing e-learning systems is that they provide the same learning resources to all the learners without taking into account the differences in their level of knowledge, skill, interest, and goals. For example, if a learner is lacking the basic understanding of a subject, it would serve no purpose to teach him advanced concepts of the same subject. This situation negatively affects learners' academic performance and their overall learning process. One of the proposed solutions to this problem is to build learner's profile based on their preferences and then recommend courses based on this profile. A learner's profile is a set of information representing a user via user related rules, settings, needs, interests, behaviours, and preferences.

In addition to the above issue, the sparsity issue is commonly found in recommender systems which are based on collaborative filtering technique. This issue occurs as a very small proportion of users rate only few items either because they have not experienced the items or they have experienced them but not bothered to rate. This leads to sparsity which prevents a recommender algorithm to find common users profile which is required in order to make good quality of recommendations. In order to fill this gap, we inferred additional knowledge about learners' preferences from moodle server and represent this knowledge using RDF in order to enrich 'user item rating matrix' which helps improve the accuracy of recommendations.

Another issue is related to the fact that, existing clustering algorithms don't provide 'cluster validation mechanism' which could be used to determine the right number of clusters in the given dataset. Hence, we used some of the cluster validation methods such as 'the elbow' and 'silhouette method' which are found to be useful to evaluate the clusters obtained through clustering algorithms.

This research aims to address the above three issues. The first issue will be addressed by creating different types of clusters namely 'not active', 'average' and 'active' which will be used as the profile of learners and are based on learners' academic performance such as assignment and quizzes and require a learner to use collaborative means such as forum, chats and messaging among others in order to complete their learning assignments.

The second issue will be resolved by employing 'cluster validation methods' in order to make sure that the clusters obtained through k-means algorithm are consistent with the actual clusters present in the dataset.

Finally the issue of sparsity is addressed by enriching the user item rating matrix with resource description framework and Apache Jena rules in order to improve the accuracy of recommendations.

1.7 Objective of Research

Although e-learning systems have many benefits as compared to traditional or face to face learning, they have quite a few issues as well. One of the major issues in the existing e-learning system is information overload due to which users are finding it difficult to choose the item or service according to their need or goal. In this context, personalization has been able to deal with this issue to some extent by using traditional recommendation techniques and few non-traditional recommendation techniques as well. However, in the domain of e-learning particularly in the recommendation of courses relatively less amount of work has been reported in the literature. Majority of existing works on 'course recommendations' uses machine learning algorithms in order to recommend the best combination of courses to learners [97]. However, majority of these 'course recommender systems' are based on usage data alone i.e semantic knowledge about user or item has not been considered while making recommendations.

The thesis aims to achieve the following objectives for recommendation of the most relevant courses to learners based on their profile:

- To improve the accuracy of personalization by integrating fuzzy technique with semantic web usage mining in e-learning domain.
- To assist the user to find relevant hyperlink based on his preference.

The following steps would be taken to achieve the above two objectives:

- Collect learners' data on various academic activities such as quiz and assignments from Moodle's server.
- Perform data pre-processing to this data in order to make it suitable for applying data mining algorithm.
- Generate clusters of learners in order to discover learners' profile.

- Validate the quality of clusters obtained in the previous step by using clusters validation methods.
- Classify a new learner into its appropriate clusters and recommend courses accordingly.
- Evaluate the classifier using 'cross validation' and 'test mode' methods.
- Analyze the data using 'fuzzy c-means algorithm' in order to accurately identify the learners' position in each cluster.
- Recommend and evaluate 'data mining courses' through various evaluation metrics.
- Extract and integrate additional preferences of learners through semantic tools such as RDF in the recommendation process in order to improve the quality of recommendation.

1.8 Scope

The proposed recommender system may be used mainly by learners, tutors, administrators and managements. All the users of this system have different goals in mind. The system will help tutors in improving their teaching process by designing their teaching goals in such a way that are according to the profile of a learner rather than offering the same learning contents to all the learners without considering their skill, level of knowledge, interest among others. On the other hand, learners will enjoy the learning process as well as they are getting learning resources based on their profile. Among many recommendation tasks, two of the important tasks are 'prediction' and 'recommendations' which will be performed by the system.

1.9 Organization of the Thesis

The thesis is divided into seven chapters. While the current chapter has introductory discussion the details of the remaining chapters are as follows:-

Chapter 2 discusses both the traditional and non traditional recommendation techniques and their approaches. Collaborative filtering which is one of the widely used recommendation techniques has been investigated along with its various approaches such as matrix factorization, probabilistic methods and k-nearest neighbour among others. The chapter also focuses on some of the recommender systems making use of web usage mining techniques and semantic knowledge which is extracted by using semantic tools and technologies in order

to improve the quality of recommendations. The chapter also discusses the finding of related work briefly.

Chapter 3 carries out an experimental analysis which involves five clustering algorithms such as expectation maximization, agglomerative clustering algorithms, divisive clustering algorithms, make density based cluster and k-means, in order to find an algorithm which takes least amount of time in building a clustering model.

Chapter 4 discusses data pre-processing which is used in our research in order to transform raw data into a format which is suitable for applying data mining algorithm. As data pre-processing consists of various steps, we require only data selection step, creation of summarization table and data transformation in order to get the required data for our purpose. The chapter also demonstrates the application of *k*-means algorithm to learners' usage data in order to generate clusters which are used as the profile of learners'. The chapter also focuses on experimental evaluation of clusters which help us to determine the optimal number of cluster (k) in the learners' data. Moreover, further analysis of data is performed by applying fuzzy *c*-means algorithm to the learners' usage data which leads to determining the position of learners in each cluster more accurately. In this chapter, we also classify learners into one of the clusters(not active, average and active). The evaluation of classifier is also performed using cross validation and test mode methods.

Chapter 5 proposes a course recommendation framework for suggesting different data mining courses to learners based on their profile. The chapter also discusses an experimental evaluation of various recommendation metrics such as mean average error (MAE), mean square error (MSE), root mean square error (RMSE), normalized mean absolute error (NMAE), precision and recall among others with the objective of identifying the most suitable metrics for evaluation of the proposed recommender framework.

Chapter 6 proposes an enhanced recommender framework which consists of RDF factbase, Apache Jena rules for enriching the user item rating matrix which leads to improvement in the accuracy of recommendations. The chapter also discusses 'moodle information model' which is used for representing different learners' activities which are stored in the log file of 'moodle server' and are then transformed into RDF fact base in the form of RDF triples. The chapter also presents 'Jena inference engine' which is used to implement the RDF 'fact base' and Jena rules in order to find preferences of learners over unrated data mining courses which help us to enrich the 'sparse user item matrix'. A couple of experiments were conducted to measure the effectiveness of the proposed framework. The

results provided by the proposed approach are compared with our own approach and also with other similar approaches.

Chapter 7 presents the conclusion, future work and the summary of the outcomes derived from the thesis, thus depict the worthiness of our proposed work and also motivate others for future work in this area. The chapter also discusses the limitations which can be tackled in the future work in order add new dimensions to the existing recommender system.

1.10 Summary

In this chapter, several recommendation approaches are discussed which are broadly categorized into two categories namely 'traditional' approaches which include collaborative approach, content filtering approach and hybrid approach and 'non traditional' approaches which involve web usage mining, semantic web usage mining and fuzzy modelling among others. The objective of integrating two fast developing field of research namely 'semantic web mining' and 'web usage mining' with 'fuzzy technique' is to improve the quality of recommendations.

The study is motivated by the fact that, a large numbers of 'e-learning systems' such as 'moodle' left a huge amount of learners' data on their server which could be used by 'machine learning algorithms' in order to improve the quality of recommendations. Moreover, the emergence of 'semantic web' has presented a large number of 'semantic tools' and 'technologies' which can be used to extract 'additional preferences' about learners to address the sparsity issue. The chapter introduces the area, the motivation and the objective for taking up the proposed work. The chapter also discusses several issues that are important and need to be catered in order to obtain the proposed objective. Further, the chapter discusses research gaps. Subsequently, the objective of research is discussed with the list of various steps that will be carried out in order to achieve the stated objectives. The chapter also gives an overview on the organization of the thesis. The chapter concludes with discussion on the scope of the proposed work. The next chapter of the thesis explores on the background of evolution of the recommender systems from traditional recommender systems to the present status with a discussion on the literature review performed and the identification of the research gap for taking up the proposed research objective.

Chapter 2

Background and Literature Review

Recommender systems emerged in 1990s in order to provide personalized services in e-commerce. The Netflix competition which was held during 2006 and 2009 helped recommender systems to become popular and also led to the development of a large number of recommendation algorithms. Since then, many algorithms were proposed for the prediction of items and services. Recommender systems have earned a wide-spread acceptance now. This has opened doors for e-commerce companies to increase their sales and earn more revenue. There are several tasks performed by recommender systems. One of them is to predict the rating of an item. In e-learning the task of a recommender system is to assist learners in choosing the most appropriate courses based on their profile. In other words, the recommendation should match with the learning needs of a learner. In the following section, we discuss some of the most significant works on recommendation systems based on traditional and non traditional techniques in the domain of e-learning:

2.1 Recommender System

Majority of recommender systems are making use of one or more recommendation techniques such as content filtering, collaborative filtering and hybrid filtering among others[46]. Many well written surveys on recommender systems are available. The authors have categorized [16] CF algorithms available as of 2006 into content based, collaborative and hybrid and also summarized the possible extension. A recent textbook on recommender systems introduces traditional techniques and explores additional issues such as privacy concerns [14]. In [29] the authors have focused primarily on collaborative filtering methods including ‘memory based’ and ‘model based’. This survey contains majority of state of the art algorithms available as of 2009, including Netflix prize competitions.

Available literature suggests that earlier recommender systems used simple database queries in order to deal with ‘information overload’ problem. The most widely used recommendation technique was ‘nearest neighbour’ [183] which is a collaborative filtering approach. It computes the similarity of a group of users (neighbourhood of target user) and matches it with the similarity of the ‘target user’ and then recommends items preferred by

them. The algorithm is efficient as it considers the most updated information from a database. However, the performance of the algorithm slows down as the size of a database increases. Hence, other filtering algorithms have been used to deal with this issue.

There have been several experimental studies as well among recommendation techniques. The first study [126] compares two widely used memory based methods namely 'Pearson correlation' and 'vector similarity' and two classical model based methods namely 'clustering' and 'Bayesian network' on three different datasets.

In the following subsection, we discuss some of the most important works conducted based on different recommendation approaches:

2.2 Collaborative Filtering (CF)

Collaborative filtering is one of the most widely used techniques in the recommendation system [47][49][50][55] [62] [64] [70] [71]. In collaborative filtering, the preferences of 'active user' are matched with the most similar users of its neighbourhood for recommending an item which has not been experienced by the active user. There are two forms of collaborative filtering namely 'memory based' and 'model based'. The memory based collaborative filtering predicts the rating of item by matching the similarity of 'active user' with training users [126]. Similarity measures used by this approach include 'Pearson correlation' and 'vector cosine similarity' [126]. In addition to these two methods, item based CF [69] and a non parametric probabilistic model based on ranking preference similarity are also used. Model based CF makes use of user and item clustering, Bayesian network [126] and probabilistic latent variable models [204]. Other widely used CF approaches are discussed below:

2.2.1 Matrix Factorization

Recommender systems based on matrix factorization split a 'user-item rating matrix' into a more compact and denser representation that can be utilized to extrapolate the prediction of items the user has not seen. For this purpose, several techniques have been employed as reported in the literature. In one such work [72] the authors have used matrix and tensor factorization in order to predict student performance in e-learning. They have implicitly taken into account latent factors. As the knowledge of a learner has improved overtime, so this factor has been taken care by tensor factorization technique. In a similar work [73] the

authors have proposed 'multi-relational factorization' models for predicting student performance. They have exploited multiple relationships between learner and skills which are required to complete a task. 'Gradient descent' is one of the most effective ways of a decomposing matrix that is computationally efficient and useful for the recommendation but doesn't preserve all the properties of proper singular value decomposition [74][75]. In [3] the authors have proposed a personalized e-learning system based on the item response theory which takes into account both course material difficulty and individual learning paths. In addition to this, they have also proposed a collaborative voting approach for adjusting course material difficulty. The results show that, personalized learning is possible with item response theory and assist learners to learn effectively and efficiently.

2.2.2 Probabilistic Methods

Another commonly used collaborative approach is probabilistic method which infers the rating data in a probabilistic way. Some of these methods are also matrix factorization methods such as probabilistic latent semantic indexing [76], probabilistic matrix factorization [77], and latent dirichlet allocation [78].

Collaborative filtering techniques have been successfully used in many application areas such as movies, articles, and product among others. 'Altavista' (AV) is one of the first collaborative filtering applications used for recommending learning resources [33]. Another popular collaborative filtering based system is 'Web-based Peer Grader' which aims to improve learners' skill by reviewing and evaluating the solutions of their peer [159]. LARGO [160] is another popular collaborative filtering system in the field of legal argumentation to evaluate the verdict (decision rule) of the supreme court of US. Moreover, collaborative filtering systems could be benefited by using the tagging information associated with social networking sites [162]. It helps to enrich the profile of a learner which leads to recommending more relevant items to user. The collaborative filtering systems are more reliable than content filtering as they can capture 'realistic relationships' among items based on similar preferences of identical users.

Existing literature suggests that majority of collaborative filtering systems use *k*-nearest neighbour [163], association rule mining algorithm [164], and matrix factorization algorithm [165] among others for making recommendations. The *k*-NN is more suitable for building collaborative systems due to its similar functionality with collaborative approach [166]. The matrix factorization is used because it allows us to discover latent features underlying the

interaction between user and item [167]. Collaborative filtering system based on association rule mining algorithms is quite effective as compared to other recommendation approaches [168].

2.2.3 k-Nearest Neighbours (k-NN)

There are several works reported in the literature based on *k*-nearest neighbour in a wide variety of domain. In one such work [163] an agent based library recommender system is proposed a recommender system for recommending books to learners. It consists of two modules namely profile agent and library recommender agent. The recommender agent is used to filter and provide recommendations. The profile agent stores the resource requirement of users. The *k*-nearest neighbour is a non-parametric method which stores the entire training dataset in memory and compares the test instance with each of the training instances in order to find its class. Among all 'instance based classifier' such as KStar[89], and IBk[91] instance based classifier is widely used for building collaborative filtering(CF) system due to its functional similarity with CF.

Table 2.1 Summary of Recommender Systems Based on Collaborative Filtering

S.No	Methods and Algorithms	Highlights of Proposed work	Results
1	Agent based recommender system using *k*-nearest neighbour [163].	Providing effective and intelligent use of library resources.	Improvement in the recommendations of resources of library.
2	Combining matrix and tensor factorization methods in order to predict students' performance [72].	Predicting learners' performance considering temporal effect, slip and guess as latent factors.	Improvement in the prediction of student's performance.
3	The proposed approach is based on multi relational matrix factorization methods (MRMF) and weighted MRMF [73].	Exploiting multiple relationships among students, tasks and skills in order to predict student's performance on a given task.	Improved prediction results.
4	Integrating *k*-means, singular value decomposition and *k*-NN[74].	Combining different efficient collaborative filtering techniques in order to obtain a good prediction.	Provided good predictions for the Netflix Prize dataset.
5	The proposed approach is based on probabilistic matrix factorization (PMF) method[77].	Combining PMF with a learnable prior and constrained PMF in order to apply them to a large dataset.	Successfully dealt with large, sparse and imbalance dataset.
6	The proposed work is based on incorporating ontology in	Recommending appropriate courses to learners based on their learning	Improved learners' performance and

	hybrid approach using [78].	requirements and goals.	satisfaction level.
7	Proposed two association rule mining algorithms [164].	Combining the best features of the proposed algorithms into a hybrid algorithm.	The combined algorithm is able to scale linearly with the number of transactions.
8	Several approaches proposed namely positive MF (matrix factorization), momentum based approach, a transducive MF, and an incremental variant of MF [165].	The proposed approaches are experimentally evaluated with the existing approaches.	Improved prediction accuracy reported significantly.
9	The proposed approach is based on relational distance computational approach [88].	To integrate learners and learning items information into a collaborative filtering framework.	Improved accurate recommendations.

There are several works reported in the literature which recommends item based on the specific requirements of a user. In one such work [88] a novel hybrid recommender system also called relational collaborative filtering is proposed which integrates learners and learning items information into a collaborative filtering framework by using relational distance computation approaches.

2.3 Web Usage Mining (WUM)

Web usage mining refers to the use of data mining techniques to 'content', 'structure' or 'usage' data in order to automatically discover and extract useful information from the web document and services [43][45][57][93]. Majority of recommender systems based on web usage mining make use of association rules [8][6][164], sequential patterns [94], and clustering [95] techniques.

Some of the significant works employing web usage mining are reported below: From the study of existing literature we have found that, there is very less number of course recommender system making use of web usage mining techniques. In one such approach [97] the authors have proposed a 'course recommender system' based on the combination of 'k-means' and 'apriori algorithm'. The objective of the proposed system is to recommend the best combination of courses in order to enhance the academic performance of a learner. The use of machine learning algorithms has become prevalent in the domain of e-learning. In [174] the authors have developed a system which predicts the probability of high drop out of students in academic settings. The proposed system is based on naïve based algorithm.

Another system is proposed in [7] which recommend relevant links to an active user. The system is based on 'web usage mining', 'content filtering', and 'collaborative filtering' techniques. Other approaches which have been used widely are 'association rules' [59] and 'ant colony optimization' [60].

The use of association rule mining is reported in several works in the domain of e-learning for improving instructional/learning performance. In one such work [198], the authors have proposed a web based course system making use of association rule mining and collaborative filtering. Association rule mining is used to find interesting information through students' usage data in the form of IF THEN recommendation rules. The score of each rule is measured through collaborative filtering approach.

Furthermore, machine learning algorithms are widely used to learn model for prediction of ratings. In particular, navies bayes algorithm is popular among researchers due to its conceptual and computational simplicity. In one such work[90] the authors have presented a lazy approach in order to learn navies based model which stores the training data and delays learning until classification time.

Several works in e-learning field suggest that, rough sets have proven to be effective in order to improve the quality of results obtained from data mining algorithms. In one such work [200] a novel effective pre-processing algorithm based on rough sets is proposed. The proposed system consists of three steps. The first step builds the relational information system by employing original dataset. In the next step, they make use of attribute reduction theory of rough sets in order to produce the core of information system. In the third and final step, construct indiscernibility matrix using reduced information system and finally, obtain the classification of original dataset. The overall result helps to reduce abundant data in data pre-processing and to reduce amount of computation in data mining process.

Recommender systems utilizing only the 'usage data' miss certain complex objects which have been added to a site recently and haven't been consumed by a user. Furthermore, without the use of 'semantic knowledge' they can't recommend certain 'complex objects' using their properties and attributes. The issues that have been found in 'web usage mining' are discussed in [177].

One of the major issues faced by existing recommender system is of information overloading. From the study of existing literature we have found that, majority of recommender systems are suffering this issue. This is due to the fact that, there are a large number of products and services available through online websites to their customers. The customers have not experienced those items which have been presented to them. Hence, in

order to deal with this situation, the role of personalization has become important which suggests items to users based on their profile. The suggested items are tailored to the specific need of the customer. In this context, several attempts have been made to deal with this issue. In [44] the authors have presented a personalization recommendation methodology in the domain of e-commerce in order to achieve more effectiveness and quality of recommendations.

Table 2.2 Summary of Recommender Systems Based on Web Usage Mining

S.No	Methods and Algorithms	Highlights of Proposed work	Results
1	The proposed work is based on the k-means and apriori association rule mining algorithm [97].	Building intelligent recommender system in order for improving the performance of learners	Best combination of courses recommended.
2	The proposed approach is based on Navies based algorithm [174].	Identifying the most appropriate machine learning algorithms for the prediction of students' dropout.	Successfully recognized students with high probability of drop out.
3	The integration of web usage mining, content filtering and collaborative filtering is employed [7].	Recommending news contents to users by basing a prediction on weighted average of the content based prediction and collaborative prediction.	Recommends relevant links to an active learner.
4	Employing association rules mining with user ratings for course recommendations [59].	To develop a course recommender system this incorporates a data mining process with user ratings in order to infer recommendations from association rules.	Recommended most relevant courses to learners.
5	The proposed recommender system is based on Ant colony optimization (ACO) technique [60].	The problem is represented using graph where each node represents a decision in the problem domain.	Has been able to predict the final grades obtained by a student.
6	Proposed approach is based on association rule mining and collaborative filtering [198].	To find, share and suggest the most appropriate modifications in order to improve the effectiveness of the course.	Has been able to continuously improve e-learning courses.
7	The propose data pre-processing algorithm uses attribute reduction theory of rough sets [200].	Classification of original dataset is constructed by producing core of information system which results in avoiding large computation in data pre-processing phase.	To reduce significant amount of undesirable data.
8	Web usage mining, decision tree, association rule mining and product taxonomy [44].	While web usage mining is used for learning customers' preferences, decision tree is used to select target customers for recommendations of products.	Improved and effective product recommendations.

9	Using ANN enhanced k-means algorithm [94].	The markov model is used to compute dissimilarities between 'web user sessions' which is further used as input of various clustering algorithms.	Discovery of true clusters

The suggested methodology is based on a variety of techniques such as web usage mining, decision tree induction, association rule mining and product taxonomy. Web usage mining is used to learn customers' preferences and product association from click stream data or web usage data. Decision tree induction approach is used to find those customers who are likely to buy recommended products.

2.3.1 Data Pre-Processing

Data pre-processing is an essential part of web usage mining process. It is used to transform the raw data into a form which is free from noise, outliers and unnecessary elements which usually come up in the process of data collection. A significant amount of work is reported in the literature regarding data pre-processing. In this section, we discuss some of the prominent works:

In [142] the authors have proposed several data preparation techniques in order to identify unique user and sessions from the log file of web server. They have also devised a methodology in order to divide user session into semantically meaningful transactions. The discovered transactions are further used to discover association rules from real world data using WEBINAR system.

Several studies have been conducted which are focused on different aspect of data pre-processing. In one such work [176] the authors have presented a detail study on emphasizing on different aspects such as data cleaning, user identification, session identification and path completion and transaction identification.

The importance of data pre-processing can be realized from the fact that, a large number of research papers have been written on this topic. In [172], the authors have presented several data preparation techniques that can be used to improve the performance of data pre-processing in order to identify unique users and session.

Typically data pre-processing involves several tasks. Hence, It would be better if one knows the best algorithms for each task of data pre-processing in order to get the best results for a given dataset. In a similar work [161], the author has presented the best algorithm to be used for each task of data pre-processing.

2.4 Semantic Web Usage Mining (SWUM)

With the emergence of the 'semantic web', many semantic tools and services have appeared which can be used to improve the quality of recommendations [48][178]. Some of the significant works making use of semantic knowledge and usage data are discussed below:

In [80] the authors presented a computational model for developing SWBES (Semantic web-based system) which addresses the problem of how to make the development easier and more useful for both developers and authors. In order to illustrate the features of the proposed model, a case study is presented. This computational model is characterized by low development costs, scalability, extensibility, interoperability, and low maintenance costs. In [157] the authors have demonstrated how 'semantic metadata' can be stored and retrieved to provide better results to the learner along with personalized learning. The proposed work presents a meaningful retrieval of resources based on the user's level of mastery and defined system ontology: Learner's profile modeling ontology and e-learning domain modeling ontology. The concepts of trust and communities have also been leveraged in recommendation systems. In one such work [21] the authors have improved the accuracy of recommendation by combining 'trust communities' and 'collaborative filtering'. They have proposed a SVD (singular value decomposition) sign based community mining method to process the trust relationship matrix in order to discover the trust communities.

The use of ontology has grown immensely with the emergence of semantic web. In one such work [182] the authors have presented an ontology based user modeling strategy in the context of personalized information access. They adopted a hybrid approach by capitalizing on the features of static and dynamic user profiling strategies. Static user profile specifies the user's interest in a much focused manner and dynamic user profiling adds the feature of adaptability into it. The dynamic user profiling strategy make use of the data sources like usage log and mouse operations that are performed by the users during the browsing sessions. Experiments were performed to evaluate the proposed method for user profiling.

In some of the works reported, users' actions are observed and have been translated into semantics. Moreover, the proposed systems have taken into consideration the change in the profile of a learner and recommend learning resources accordingly. In this context, the authors in [183] have suggested a theoretical framework ALMS (Adaptive Learning management System) which focuses on three aspects 1) Extracting the knowledge from the users' interaction, behavior and actions and translate them into semantics which are represented as Ontologies 2) Find the Learner style from the knowledge base and 3) Deriving

and composing the workflow depending upon the learner style. The intelligent agents are used in each module of the framework to perform reasoning and finally the personalized workflow for the e-learner has been recommended.

Semantic web technology is leveraged for building recommender systems. In one such work, [175] the authors have proposed a recommender system based on 'web usage mining' and 'semantic web'. They have used RDF(resource description framework) 'payloads' as 'annotated semantic metadata' containing the topics, social tags, identified entities, facts and events which leads to the creation of RDF graph database which is further used to compute similarity between web pages. In [184] the authors have proposed an intelligent searching mechanism based on predefined semantic concepts and hashing and similarity algorithms. In this study, their objective is to improve a search mechanism based on content hashing in mobile peer devices and address the issue of leveraging semantic web for knowledge representation in the area of education. The author attempts to analyze various techniques applied in adaptive educational systems, in order to find out how such systems can be improved by leveraging the Semantic Web technologies to represent knowledge in different models utilized in these systems. The proposed searching mechanism consists of keyword values taken from the mobile ontology in the network servers and a 'hashing algorithm' working on each mobile node used by the students. However certain issues that were not addressed but highlighted are interoperability among different courses, since they need different ontological structures; database integration in different mobile devices; difficulty in establishing necessary XML syntax; semantic conceptualization for keywords.

Resource description framework has been used by many researchers in a variety of domains for different purposes. In one such work[146] have proposed an approach for recommending learning objects(LO) to learners on the basis of their learning preferences. They have used RDF as a parser in order to map each learning objects to its equivalent RDF schema.

The work presented in [181] makes use of recommendation and adaptive hypermedia techniques. They use XML, RDF and OWL for the representation of ontologies. These standards are further used for standardization and formalization of content and interoperability. The ontological representation using RDF helps in not only to represent meta-data but also for reasoning in order to provide the best recommendation for each individual learner.

In [207] the authors proposed architecture for integration of the products with web log data and generate a list of recommended products by using LCS (longest common sequence).

They use semantic knowledge of the products which has been stored in RDF model provided by JENA framework.

As compared to other domains, the semantic web technologies, particularly RDF have been used relatively less in the domain of e-learning for course recommendations. The most recent work on the use of semantic web is proposed in[81] in which the authors have proposed a semantic web architecture for recommendation of learning objects based on the profile of learner, activities and social interactions. Learning context of a learner has also been taken into account for recommending relevant learning material. The authors have used social ontology that is built from FOAF ontology which is built on the RDF.

There are few works reported in literature which focus on improving the accuracy of intelligent tutoring system. In one such work [185] the authors have presented a novel architecture of agent based simulation of teaching and learning process. This architecture deals with the issue of how to control student's progress in order to improve the efficiency of 'intelligent tutoring system'. The focus of this study is on usage of ontology for agent communication and formal description of learning content and process. Several agent have been used for different purpose such as tutor agent for the purpose of simulating tutor's work, search agent is needed to support student agent with appropriate course material, content manager supports representation of information found by the search agent in a form which is understandable and easy to use, test agent compares student's agent domain's ontology with tutor agent's domain ontology etc. The main emphasis of this study is to describe the role of ontologies and ways how ontology can be used in intelligent tutoring system.

Commercial web search engine have made heavy use of semantic web technologies. In [186] the authors aim to characterize websites in terms of semantics of the queries that lead to them by linking queries to large knowledge bases on the Web. They demonstrate how to exploit such links for more effective pattern mining on query log data. They also show how such patterns can be used to qualitatively describe the differences between competing websites in the same domain and to quantitatively predict website abandonment. They have shown how to use semantic knowledge to aid several types of analyses of queries on a commercial Web search engine.

The two fast emerging techniques namely semantic web and web usage mining have been integrated in several works as reported in the existing literature. In [187] the authors have integrated web usage mining with semantic knowledge in order to obtain semantic profile using which enhanced quality of recommendation can be suggested to a learner.

Sometimes users are not willing to provide feedback which is usually employed by recommendation algorithm particularly collaborative filtering in order to improve the quality of recommendations. This issue has been dealt in several works reported. In one such work [188] the authors describe an automatic personalization approach aiming to provide online automatic recommendations for active learners without requiring their explicit feedback. Recommended learning resources are computed based on the current learner's recent navigation history, as well as exploiting similarities and dissimilarities among learners' preferences and educational content.

Users are often overwhelmed by the huge information available on the web. Recommending most appropriate link containing the required information sought by user is not possible using usage data alone. Hence, usage data is combined with domain knowledge in order to recommend most appropriate link to the user. In one such work [189] the authors have proposed a framework to personalize e-learning services using semantic web mining technologies. In this framework they have distinguished two stages namely offline task and online task. Off line task stage includes data preparation, ontology creation and usage mining and online task consist of generation of recommendations for learners. Data preparation component in off line task results in aggregate structure such as a user transaction file computing meaningful semantic units of user activity to be used in the mining stage. Given this preprocessed data, Apriori algorihtim is applied over it to discover association rules. For effective personalization authors have combined ontology of the content with the knowledge that comes out of the user's navigation paths. Ontology has been used to find the most relevant material for the learner. The online part deals with the generation of recommendations. It keeps track of active user session which contains recent past user choices. According to his current state a recommendation engine recommends him the next more appropriate link. The recommendation engine accepts active user session and also takes into consideration the ontology of the domain and the set of association rules which came from user's transaction during the offline part.

The use of semantic feedback is reported in many works. In [190] the authors have proposed an approach based on semantic web technologies for generating semantic feedback for both teachers and students. Firstly, the knowledge of the course and the exams is modeled using a course ontology written in the web ontology language. Secondly, semantic annotations were provided for any question included in the exam. Feedback component which is an important part of assessment has been incorporated in the OeLE platform for both teacher and students. They have also described the process of feedback generation and how to

provide such feedback to teachers and students. Feedback generation algorithm reuses the semantic component included in the assessment method.

Although e-learning systems offer several advantages, they have few issues as well. One of the crucial issues is lack of face to face communication between a learner and a tutor. Several attempts have been made in order to address this issue. In one such attempt [191] the authors have addressed the issue of lack of face to face communication in E-learning which causes tutor not being able to chase students learning progress. In this approach the two known improvement approaches were suggested namely 'index of learning style' and 'study strategies inventory' and combined them with data mining. Semantic structure is used to classify learners using data mining techniques so that cognitive learning style is used as a dependency factor for the problem. In order to established user learning style index of learning style questionnaire is used as an instrument to assess preferences on four defined dimensions. In order to determine the kind of learning strategy for learners a modified version of LASSI (Learning and study strategies) was used.

In yet another work [192] the authors have proposed a web mining approach for the Semantic Web. The approach uses a search engine and the traditional web as a source of information to produce semantically rich information. In particular, they assess one community and obtain the social network and related information from the Web. As an example, they extract the social network of an academic society and show that extracted information can be incorporated into FOAF representation and utilized to measure the authoritativeness of a member in terms of social trust or individual trust. To demonstrate the Web mining approach in the real application, they show a researcher mining and retrieval system. Finally, they discuss the manner in which the Web mining approach contributes to availability to users of the Semantic Web.

Traditional user based or item based recommender did not perform well when little information about ratings of user or item is available. Hence, in order to deal with this issue semantic knowledge started to be used by the researchers. In a similar work [193] the authors have proposed an approach for semantically enhanced collaborative filtering in which structured semantic knowledge about items, extracted automatically from the web based on domain specific reference ontologies, is used in conjunction with user item mapping in order to create a combined similarity measure and generate predictions. Recent works have shown that, incorporating semantic knowledge in the form of ontology plays an important role in enhancing the quality of recommendations. In a similar work [194] the authors have integrated semantic knowledge in each step of web usage mining process. They have also

used clospan algorithm for sequential pattern mining which results in the generation of semantically enriched patterns which are inputted to web page recommendation model. They also evaluated their proposed approach experimentally, which shows significant improvement in the quality of recommendations.

There are several recommendation approaches in the domain of e-learning which have employed ontology for recommendation of suitable courses to learners. In one of such approaches [196] the authors have proposed an e-learning recommender system based on ontology and web ontology language (OWL) rules. This approach consists of two subsystem namely semantic based system and rule based system. Either of the subsystem consists of observer, learner profile, recommendation storage and user interface. This approach allows a learner to find and choose learning material according to their interest.

In this work [197] the authors have exploited semantic knowledge for e-learning environment of the university. They present a semantic web-based model for e-learning system taking into account the learning environment at universities. The proposed system is mainly based on ontology-based descriptions of content, context and structure of the learning materials. It further provides flexible and personalized access to these learning materials. The framework has been validated by an interview based qualitative method.

Table 2.3 Summary of Recommender Systems Based on Semantic Web Usage Mining

S.No	Methods and Algorithms	Highlights of Proposed work	Results
1	The proposed approach considers the features of both the static and dynamic user profiling strategy [182].	User profiling strategy captures the change in information needs of the user profile which is acquired in two phases.	Construction of adaptable user profile.
2	The proposed recommender system combines trust communities and collaborative filtering [21].	Unlike other approaches which only consider trust relationships in memory based collaborative filtering, the proposed approach also considers distrust relationships as well in order to improve recommendation accuracy.	Improved recommendation accuracy is obtained.
3	The proposed recommender system is based on the new collaborative filtering approach [79].	To recommend learning resources to a group of learners by merging the preferences of different learners and extract a pseudo unified learner profile that represents the preferences	Effective group recommendations.

		of all the learners.	
4	The proposed work uses web usage mining, semantic web and LCS algorithm [175].	Integration of semantic information is performed with web log data and generates a list of recommended products using LCS algorithm.	Shows good performance in terms of precision, recall and F1.
5	Intelligent searching mechanism based on predefined semantic concepts, hashing and similarity algorithms [184].	Leveraging semantic web for knowledge representation.	Ontology working in mobile devices benefits from adapting context and content information.
6	The authors have proposed an approach based on 'web usage mining' and 'information retrieval'[188]	Automatic personalization approach aiming to provide online automatic recommendations for active learners with requiring their explicit feedback.	Enhancing the quality of learning objects for recommendations.
7	The proposed approach integrates 'web usage mining' and 'domain knowledge' [187].	The framework integrates domain ontologies with web usage mining and personalization process at different stages in order to improve the quality of recommendations.	Generating improved recommendations.
8	The authors have proposed an approach based on 'collaborative filtering' and 'semantic knowledge' [193].	Enhancing collaborative filtering semantically by integrating structured semantic knowledge about items with user item mappings to create a combined similarity measures for making predictions.	Improved accuracy, successfully dealt with sparse dataset.
9	Resource description framework (RDF) and ontology language (OWL) is used for building semantic web based e-learning system [197].	The proposed work includes various services such as course registration, uploading course document among others. The OWL is used for developing ontologies. The Protégé tools are used to create the e-learning ontology classes and properties.	Provides many useful services such as semantic search and useful links among others.
10	The proposed system is based on ontology and web ontology language [196].	Semantic recommender system consists of two subsystems namely semantic based system and rule based system.	Learners are able to choose the most appropriate learning material.

2.5 Fuzzy Techniques

A large number of e-learning systems based on fuzzy techniques are reported in literature. In [201] the authors have proposed a fuzzy tree structured learning activity model and a learner profile model in order to describe the complex learning activities and learners' profile. The approach has employed both the knowledge based and collaborative filtering approaches and takes into account both semantic and collaborative similarities. Based on this a prototype of an e-learning recommender system is designed and developed.

A personalized e-learning material recommender system [42] is proposed based on multi attribute evaluation method and fuzzy matching method. The first method is used in order to justify a student' need and the latter one is used in order to find suitable learning materials to best meet each student need.

Table 2.4 Summary of Recommender Systems Based on Fuzzy Approaches

S.No	Methods and Algorithms	Highlights of work Proposed	Results
1	Fuzzy tree matching approach utilizing 'knowledge based approach' and 'collaborative filtering' [201].	A personalized e-learning recommender system is proposed in order to assist learners to select proper activities for their particular requirements in order to meet their learning goals.	Dealt with issues such as vague learning activities, found semantic similarity between learning activities.
2	Recommender system based on integrating 'multi attribute evaluation methods' and 'fuzzy matching' method[42].	The proposed framework uses the multi attribute evaluation method to discover the learners' need and fuzzy matching method is employed in order to find suitable learning materials that best meets each learners need.	Recommending learning materials to learners.
3	Proposed approach based on fuzzy logic and collaborative filtering[143].	Fuzzy logic is used for modelling student clusters. The learning style is considered as an attribute of learner which helps to find its group.	Recommendations are provided to group of learners rather than a single learner.
4	The proposed framework is employing fuzzy cognitive map [147].	The proposed approach uses knowledge representation and its learning and reasoning mechanisms for improving the quality of recommendations.	The authors have been able to improve the quality of recommendations.

5	Fuzzy collaborative and 'content based algorithm based approach is proposed [148].	The proposed approach is used to alleviate the stability vs plasticity problem of technology enhance learning recommender systems.	The proposed technique is feasible and effective.
6	'fuzzy rule system' and 'Gaussian membership function' approach is used [169]	Learning style prediction is performed with fuzz rules to handle uncertainty in learning style prediction and the evaluation is performed with Gaussian membership function.	Improved prediction accuracy of learners' learning style.
7	The authors proposed a conceptual framework based on 'fuzzy logic' and 'collaborative filtering' algorithm[170]	Capturing relationships between users and items thus overcoming sparse or nonexistent rating data.	Able to recommend one and only one item.
8	The proposed approach uses fuzzy set theoretic method for recommending learning resources [171].	Recommending appropriate learning material for learners with different preferences.	Improvement in the value of precision without loss of recall.

Fuzzy logic is used in several works as reported in the literature [83][84]. In one such work [143], the authors have proposed a recommender system for learners belonging to the same cluster having similar characteristics. A new learner, who joins the group, is suggested resources based on the resources already recommended to similar learners in the same group.

An intelligent recommender system is proposed in [147] based on fuzzy cognitive map. The authors have taken the concept of knowledge based system. The proposed system is able to exploits knowledge, infers preferences, and discovers new information among other things. The implementation of the system has been carried out using fuzzy cognitive mapping.

Adaptability is one of the most desirable features of current e-learning systems. It allows a recommender system to suggest learning resources under different context. This feature has been implemented in few existing systems. In a similar work [148], the authors have proposed a recommendation approach that combines fuzzy collaborative approach with content based approach in order to make better recommendations using learner's preferences and importance of knowledge in order to deal with stability vs plasticity problem of technology enhanced learning.

From the study of existing literature, it is found that, the performance of a learner can be improved significantly by suggesting learning contents according to their learning style. In one such work [169] a model is proposed for discovering the learning style of a learner using fuzzy inference engine. The proposed model has been experimentally evaluated which results in the improvement of accuracy in prediction significantly.

Many of the existing works have made use of fuzzy logic for finding relationships between user and item and hence being able to recommend items even in the absence of rating matrix. In one such work [170] the authors have proposed a conceptual framework for recommending one and only one item which uses fuzzy logic in order to capture graded/uncertain information in the domain and to extend the collaborative filtering algorithm thereby overcoming limitations of existing techniques.

In some of the works, fuzzy logic is used to suggest learning resources to different learners with varying preferences. In a similar work [171], the authors have used fuzzy set theoretic methods for recommending most relevant learning material to learners. Both the learning resources and users are represented using fuzzy value. They have been able to achieve an improvement in precision without loss in recall.

2.6 Summary

From the extensive study of existing literature, we have found that, there is a lack of ‘cluster validation mechanisms’ in the existing clustering algorithms, hence, the creation of clusters through these algorithm is not being validated which often results in producing clusters not being representative of the actual clusters present in a dataset.

The chapter also identifies several issues related to recommender system such as ‘cold start’, ‘scalability’, and ‘overspecialization’ among others.

A majority of existing recommender systems are heavily dependent on ‘usage data’ which is stored in the log file of a web server. Consequently, they are not being able to exploit the hidden ‘semantic knowledge’ underlying a site, which could be exploited in order to get more information about learners’ preferences which helps in improving the quality of recommendations.

Another important issue is related to the use of ‘semantic web tools’ such as RDF (resource description framework) for finding additional information about learners and items which could address the issue of ‘sparsity’. These tools could help in enriching the ‘user item

rating matrix' which can be used by recommendation technique such as collaborative filtering in order to enhance the accuracy of recommendations.

The chapter also discusses several fuzzy based approaches particularly 'fuzzy-*c* means' which can be used for the analysis of clusters in order to determine the accurate position of a data object in the given dataset.

The study of literature in the area of 'web usage mining' leads to issues such as 'privacy' which is one of the critical issues and demands a fine balance between users not willing to disclose their personal information while surfing the web and the administrators who may want more personal information of users in order to improve their surfing experience.

Another issue with respect to 'web usage mining' is of 'pre-processing of click stream data' which is one of the essential steps in the process of web usage mining. While data pre-processing involves a number of tasks, data cleaning, normalization, transformation, features extraction and selection are some of the common tasks found in a data pre-processing step of data mining. This step is a prerequisite before applying data mining algorithms to raw data in order to get the best results from machine learning algorithm.

Mapping between 'usage level data' and 'domain level instances' is yet another issue in web usage mining which leads to the creation of 'semantic user profile' and thus enrich a user's profile with semantic knowledge about item which helps in recommending those items which otherwise would not be possible to recommend without the use of semantic knowledge in user's profile.

With the emergence of the 'semantic web' a large number of semantic technologies and tools have come up which have been used in several areas successfully including e-government, e-commerce, e-resources and e-learning among others. However, the semantic web has also presented few issues as well. One of them is, representing the 'domain knowledge' in terms of 'content feature' or 'structured data' which must be made carefully as it directly affects the quality of recommendations.

Although 'semantic web usage mining' has been successfully used in recommendation systems, it also presents several issues. One of them is how to represent domain knowledge as content feature.

For the effective processing of data in domain knowledge, it has to be converted into structured data. This is another crucial issue found in the existing literature as how to represent domain knowledge as structured data.

The 'query string' of URL can also be leveraged for getting more information about users' preferences so that they can be recommended most suitable items. However, the issue of extracting information from 'query string' needs to be addressed.

Among all the issues that we have identified as a result of the study of existing literatures, this research intends to bridge some of these gaps particularly the issue of 'cluster validation mechanism' for validating the clusters obtained through the clustering algorithms and taking advantages of 'semantic web technologies' such as resource description framework (RDF) for representing learners' activities in order to discover their preferences implicitly for enriching 'user item rating matrix'.

The resource description framework is a standard model for representing data in the semantic web [199]. It has been used in many recommender systems in the domain of e-learning for different purposes [202][207][181][146].

While in these works, some authors have used RDF as 'annotated semantic metadata', in other works, it has been used to achieve different objects such as to store semantic knowledge about the users and items, to represent domain knowledge in the ontological representation, for reasoning, as a parser in order to map each learning objects to its equivalent RDF schema. In the most recent work, RDF is used for building social ontology constructed from FOAF ontology based on RDF.

Unlike previous approaches, we used RDF for finding users' preferences implicitly by using RDF factbase based on RDF schema which leads to the enrichment of 'user item rating matrix' and hence helps in improving the quality of recommendations. In the next chapter, we analyze the performance of some of the widely used clustering algorithms with respect to various well known parameters such as normalized dataset, un-normalized dataset and changing the number of clusters in order to see their effect on the time taken by the algorithm to build a model.

Chapter 3

Evaluation of Clustering Algorithms

With the availability of several clustering algorithms, it is often difficult to find the most appropriate one for the problem being solved. In addition, the efficiency of an algorithm plays a key role in the response time of a system particularly recommender system. The time when a query is submitted to the system to the time it gets executed and finally a user gets the response should be minimum. Keeping this in mind, we perform an experimental evaluation of the clustering algorithms such as k-means, expectation maximization, hierarchical clustering and make density based clustering algorithms with respect to various critical parameter such as changing the number of clusters, changing the size of dataset, using different dataset, un-normalized datasets, and normalized datasets. The primary objective of this experimental analysis is to come up with a clustering algorithm which takes least amount of time in building a clustering model. The selected algorithm will be used in order to build the profile of a learner.

3.1 Overview of Clustering Algorithms

Data is vital to any organization for various reasons ranging from strategy formulation in business to decision making. However, for data to be useful to an organization it needs to be converted into useful knowledge. Earlier, conventional tools and techniques were lacking in terms of features, had limited support and were not capable of transforming data into meaningful knowledge. For example, statistical methods help to just quantify data. They are lacking methods as provided by data mining such as classification, clustering, neural networks, association, sequence-based analysis, estimation and visualization. Statistical methods include probability distribution, estimation, hypothesis testing, model scoring, markov chain monte carlo, generalized model classes among others. Hence due to the above limitations of conventional data processing tools and techniques there was a need for more sophisticated methods which could yield unseen knowledge from huge amount of data that earlier methods could not detect. As a result data mining techniques were developed.

The potentially useful and interesting knowledge obtained by data mining techniques can further be analyzed and used for improving marketing strategy in business domain, enhancing student’s learning process in e-learning domain, weather forecasting and discovering

consumer patterns among others. . The primary goal of data mining is to transform raw data into useful and interesting information [107]. It discovers information from huge data which queries and reports are not able to discover [108]. Data mining techniques can be used to perform a variety of tasks such as trend analysis, summarization among others [109]. One of the tasks of data mining is clustering which is widely perform in a variety of domain such as image processing, pattern recognition, text mining, bioinformatics, machine learning, voice mining, web clusters engines, and weather report analysis among others. Clustering is an unsupervised form of data mining [110] which means having no data label in a dataset unlike classification where we already have a dataset along with the various classes. An instance in the dataset could belong to any of the instances in dataset depending upon the values of features. The goal of clustering is descriptive which means if we generate a set of clusters by applying a clustering algorithm to dataset then the characteristics of cluster are descriptive. Moreover, the descriptive algorithms such as clustering transform data into relevant information or in other words one can say that to summarize what has happened in the past. On the other hand the task of classification is predictive [111] which means the algorithm needs to find a class based on the different features of an object. Furthermore, such algorithms are of probabilistic by nature which tells us what will happen in the future.

There are two types of clustering found in literature namely hard clustering and soft clustering respectively. The concept of hard clustering allows a data point to belong to just one cluster at any point of time i.e if an object falls in one cluster then it cannot lie in another cluster at the same time. It is also called binary clustering. On the other hand soft clustering is more flexible by nature because it allows an object to belong to each cluster with certain degree of membership [112]. The data points belonging to multiple clusters may be falling on the boundary or lying close to the centre of one cluster or other.

There are several clustering algorithms which are broadly divided into the following categories: Partition based, Hierarchical based, Density based, Grid based and Model based [113]. These algorithms are discussed in [114] along with their applications in various domains. In [115] the authors reviewed the clustering algorithms and other important issues related to cluster analysis. Furthermore, given a number of clustering algorithms it becomes difficult to decide which one to choose for a particular problem. The decision regarding the suitability of a clustering algorithm in a solving a given problem is paramount. To make this task easier we could compare algorithm with respect to various suitable parameters and analyze their performance. For instance, if efficiency of an algorithm is important for an application then the algorithms must be compared using “time complexity” parameter.

Moreover, if scalability of an algorithm is critical then size of dataset could be considered as parameter. In the following section, we provide a brief overview of each algorithm which is used for comparison:

3.1.1 *K*-Means Algorithm

K-means clustering is used for dividing a dataset into disjoint group of clusters. Each cluster is represented by a data point which is also termed as centroid of a cluster. The main goal of the k-means algorithm is to group the data points in a dataset in such a way that each data point in a cluster is located closest to the centroid. However, in practice the distance between data points in clustering does not characterize the spatial distances. Hence, the only feasible solution is to try all possible starting points. It is to be noted here that the coordinates of a centroid is calculated by making the average of each of the points of samples used in a cluster. The actual data needs to be collected and pre-processed before applying the k-means algorithm. One of the crucial factors in the process of clustering is assign priorities to features which represent data points in a dataset. The value of these features determines a feature vector. While using k-means one has to decide the type of distance metric to be used.

Although there are many distance metrics available such as Manhattan distance, city block, cosine, correlation, hamming among others, we used Euclidean metric which is widely used metric for computing distance between two points in a dataset. K-means algorithm requires two inputs from a user such as number of cluster to be generated and starting points. Once starting points are known then the distance from each data point to the starting points are computed using Euclidean distance metric. After this each data point is placed in the cluster nearest to its starting point. After all data points are placed to their clusters the new cluster centroid is computed. The new centroids are considered as the new initial points. This process is repeated until there is no change in the value of centroid. Some of the distinctive features of k-means are discussed in[103].

The k-means algorithm only supports numeric attribute. However, there are several studies found in literature which shows that this limitation has been overcome now. In one such work [10] the author presented a prototype of the algorithm which removes numeric data limitations while maintaining its efficiency. Parallel techniques for k-means were developed that can accelerate the growth of the algorithm [36]. Although the algorithm has been successfully used in a variety of domain and applications, the algorithm suffers from the limitation of finding the optimal number of cluster the issue with k-means algorithm has been

addressed through an algorithm named ISODATA [37]. The main steps of k-means are briefly described below:

Algorithm:

INPUT: K(where K is the required number of cluster)

Data points (D): $\{d_1, d_2, d_3, d_4, \ldots\ldots d_n)$

OUTPUT: A set of K clusters.

Steps:

1: Initialize K cluster centres.

2: While termination condition is not satisfied do

3: Assign instances to the closest cluster centre.

4: Update cluster centres based on the assignment.

5: end while

3.1.2 Expectation Maximization (EM)

The algorithm is based on clustering model which tries to fit data points to a scientific model. The algorithm aims to discover maximum likely estimates for model parameters with incomplete data [104]. The algorithm is also used in various motion estimation frameworks. The EM algorithm is an expansion of the k-means algorithm which is iterative by nature and looks for maximum likelihood solutions. This algorithm works in two steps: the first step is the expectation step which assigns data points to clusters based on fuzzy clustering or using probabilistic clusters. In the maximization step it looks for new clustering that maximizes the expected likelihood. The method is an iterative one to approximate the maximum likelihood function. The algorithm continues to work well even in the presence of outlier noise and incomplete information. The time taken by this algorithm depends on number of iterations and time to compute E and M steps.

Algorithm:

Step1. Initialize: Set i = 1 and choose an initial θ_1

Step2. Repeat :

(a) Expectation (E): Compute

$Q(\theta, \theta_i) = E\,\theta_i [\ln p_\theta (Z, X \mid X)]$

$= \int \ln (Z, X) p (Z \mid X) dZ.$

(b) Maximization (M): Compute

$\theta_i + 1 = \arg \max Q(\theta, \theta_i).$

(c) $i \leftarrow i + 1$

3.1.3 Hierarchical Clustering (HC)

The main objective of HC is to build clusters of hierarchy. That means data points are combined into clusters and these clusters are further combined into bigger clusters and so on which ultimately creates a hierarchy of clusters. One of the issues with the algorithms is its inability to adjust after merging or splitting the data points as neither undone of the previous operations is allowed nor swapping of data points in clusters. Hence, it becomes very important to take the right decision of merging or splitting as a wrong decision might lead to low quality of clusters. However, in order to improve the quality of clusters hierarchical method can be integrated with other methods. Several advances have been reported in Hierarchical algorithm. HC is broadly categorised into two types [105].

3.1.3.1 Agglomerative Clustering

This is based on 'bottom up approach' where clusters have sub cluster which in turn have sub cluster. One of the properties of this algorithm is that, the clusters generated in the early stages are nested in the clusters created in the later stages. In addition, clusters with different sizes are more important than others. Chameleon HC algorithm is based on k-nearest neighbour graph in which an edge is removed when both vertices are not within the k closest point related to each other. Some of the widely used hierarchical clustering algorithms are CURE (clustering using representatives), BIRCH (Balanced iterative reducing and clustering using hierarchies), ROCK (robust clustering using links), Linkage algorithms, Leader-sub leaders, and Bisecting k-means clustering algorithm among others.

Algorithm:

Step1.Begin:

(a) Assign number of cluster=number of objects.

Step2. Repeat:

When number of cluster = 1 or specify by user

a) Find the minimum inters cluster distance.

b) Merge the minimum inter cluster.

Step3. End.

3.1.3.2 Divisive Clustering (DC)

This approach is based on 'top down approach' which means that this algorithm starts at the top with all data points in single cluster. The clusters are split using a flat clustering algorithm. This process is applied recursively until each data point is in its own cluster. For a cluster with N objects there are 2^{N-1}-1 possible two subset division which is very expensive in computation. The approach is conceptually is more complex than bottom up approach and this is mainly due to the fact that we need another clustering algorithm as subroutine for running DC algorithm. The efficiency of this algorithm shows up when we do not produce a complete hierarchy from top through intermediate node to leaves. The algorithm is found to be linear in nature as more the number of clusters and data points the more time it takes to run. If we compare DC and HC then it is discovered that HC algorithms are quadratic in terms of time complexity. The basic principle of DC was first published as DIANA (Divisive analysis clustering) algorithm [106].

Algorithm:

Step1. Begin:

Assign number of cluster=number of objects.

Step2. Repeat:

When number of cluster = 1 or specify by User.

a) Find the minimum inters cluster distance.

b) Merge the minimum inter cluster.

Step3. End

3.1.4 Density Based Clustering (DBC)

It is one of the widely used clustering algorithms which work well when we have to find non linear clusters based on density in a given dataset. In this context, DBSCAN (Density based spatial clustering of application with noise is most commonly used density based algorithms. DBC is based on these two concepts such as density reach ability and density connectivity[].

3.2 WEKA (Waikato Environment for Knowledge Analysis)

Weka is open source software which is free and developed at the University of Waikato, New Zeeland [179][137]. It is licensed under the GNU general public license. It contains tools for

data analysis and predictive modelling. It consists of a collection of machine learning algorithms in order to perform data mining tasks. Weka provides the option of directly applying these algorithms to a given dataset or they can be called from java source code. It contains an array of algorithms for performing a variety of tasks such as clustering, data pre-processing, classification, regression, association and visualizations. We use Weka in order to analyze the performance of various clustering algorithms such as k-means, expectation maximization, hierarchical clustering algorithm, and make density based cluster.

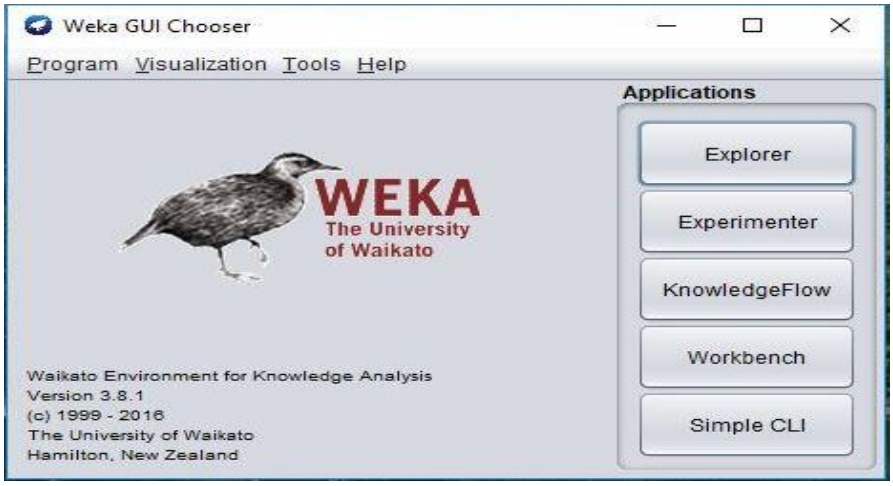


Figure 3.1 Interface of Weka 3.8.1

We have also taken open source datasets that come with Weka software. We have considered four dataset of different types and having varying size. The description of datasets along with the number of instances is shown in Table 3.1.

Table 3.1 Datasets with their Attributes and Instances

Dataset	Number of attributes	Number of instances
Iris	5	150
Glass	10	214
Segment-test	20	810
Vote	17	435

3.3 Experimental Results

First of all we compare the clustering algorithms by varying the value of ‘k’ which denotes the number of cluster. The chart in Figure 3.2 shows that when the value of ‘k’ is set to 2, the

'expectation maximization' algorithm takes highest amount of time than other algorithms in order to build clustering model whereas *k*-means takes least amount of time. The performance of density based clustering algorithm' is next to *k*-means. The value of 'k' is varied from k=2 to k=14 and subsequent observations show that the *k*-means algorithm performs better than other algorithms. Moreover, when the value of k is set to 2 the expectation maximization algorithm has taken significantly more time than the rest of the algorithms. Another interesting point to be noted is that when we increase the number of cluster the time taken by the expectation maximization algorithm has gradually decreased. Moreover, the time taken by the 'hierarchical clustering' algorithm did not vary much and remained almost constant. The results of comparison are shown in Table 3.2 and the same is graphically represented in Figure 3.2.

Table 3.2 Time taken by Algorithms when Number of Clusters are Varied

K(number of cluster)	K-Means	Expectation Maximization	Hierarchical Clustering	Make Density Based Clustering
2	0.1	87.23	2.69	0.02
4	0.01	0.41	2.55	0.023
6	0.1	0.34	2.52	0.02
8	0.03	0.52	2.47	0.02
10	0.2	0.63	2.45	0.2
12	0.3	0.72	3.39	0.1
14	0.3	0.7	2.45	0.3

Consequently, we can say that the *k*-means algorithm outperformed other algorithms when the number of cluster is varied.

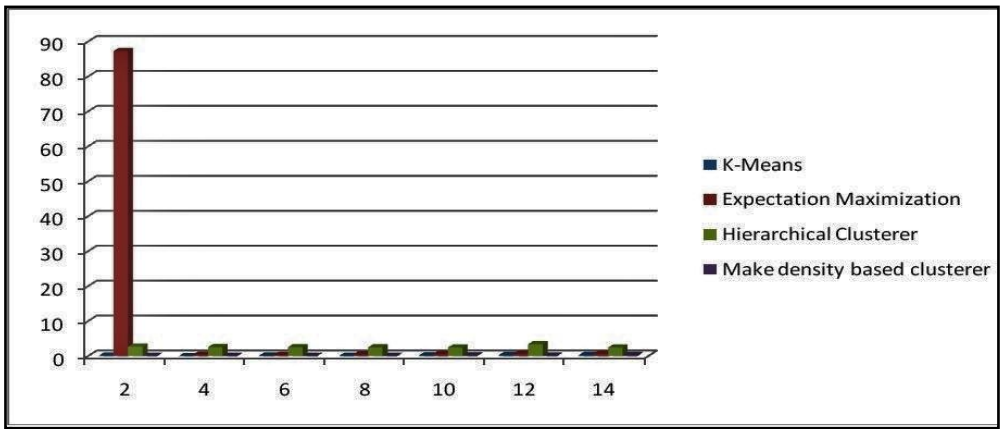


Figure 3.2 Time taken by Different Algorithms when Number of Cluster is varied

We also compared the time complexity of the four algorithms by changing the size of dataset. When the size of dataset is set to 200, the *k*-means algorithm took least amount of time which can be seen both in Table 3.3 and Figure 3.3. The reason for this behaviour by the k-means algorithm is due to its nice terminating behaviour. The likely cluster assignment are $k^{\wedge n}$. If, in

each step we choose the one that is better then, it will terminate after trying out all $k^{\wedge n}$. However, in reality it generally terminates after few dozen steps. Furthermore, expectation maximization takes significantly more time as objects are assigned to all clusters with some degree of probability. Due to this there is no guarantee for termination. Based on this one can say that the run time of expectation maximization algorithm is infinite theoretically. The performance of density based algorithm comes next to the EM algorithm.

Table 3.3 Time Taken by Algorithms when Size of Datasets Varied

Dataset Size	K-Means	Expectation Maximization	Hierarchical Clustering	Make density based clustering
200	0.05	7.67	.06	.06
400	0	3.88	0.58	.02
600	0	4.53	0.83	.03
800	0	6.24	1.66	.02
1000	.03	29.84	3.2	.02
1200	0	84.56	4.92	.01
1400	.03	27.68	5.51	.03

Furthermore, it is also interesting to note that, the size of dataset from 400 to 800 does not have any effect on the running time of *k*-means algorithm, while others algorithms took some time to build a clustering model. The unusual behaviour shown by expectation maximization algorithm is at k=1200 which is significantly higher than the rest of the algorithms. The results of the comparison with respect to change of size of dataset are shown in Table 3.3 and also demonstrated graphically in Figure 3.3.

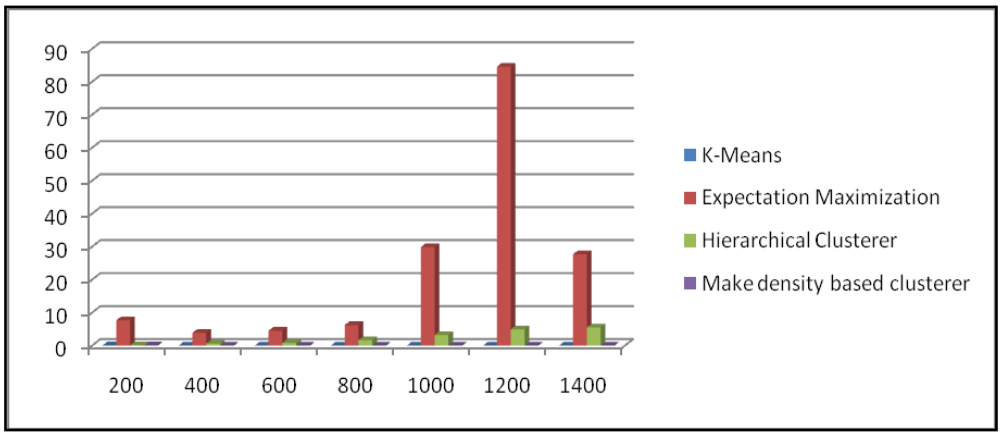


Figure 3.3Time taken by Different Algorithms when Size of Datasets is varied

We also consider different dataset in order to see the performance of different algorithms. Table 3.4 shows that *k*-means outperforms among all the four algorithms. Moreover, the algorithm takes significantly more time when applied to segment test dataset. In addition, the

expectation maximization algorithm took relatively more time to build a model. When we ran the EM algorithm to iris and glass dataset, the time taken by the algorithm is almost similar. Least amount of time taken by the algorithm is in case of vote dataset. It is also interesting to note that, both the algorithm namely *k*-means and make density based cluster took less time as compared to other algorithms. In addition to these two algorithms, it is the hierarchical clustering algorithm which comes next in terms of time taken in building a model.

Table 3.4 Time Taken by Different Algorithms using Different Datasets

Algorithms	Data (Iris)	Data (Glass)	Data (Segment-test)	Data (Vote)
K-means	.01	0	0.02	0.02
Expectation Maximization	0.86	0.88	51.52	7.16
Hierarchical Clustering	.06	0.13	2.52	1.94
Make density based clustering	.02	0.02	0.03	0.03

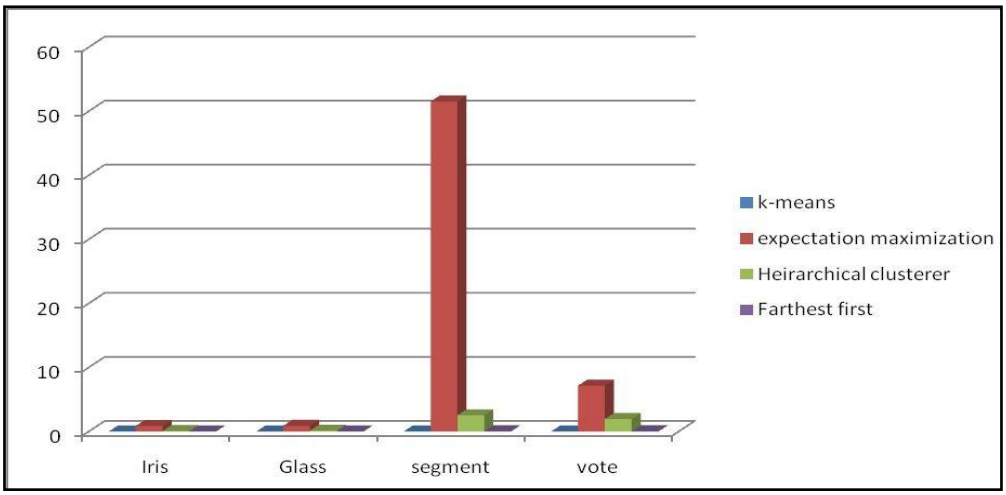


Figure 3.4 Time taken by Different Algorithms using Different Datasets

The results of the comparison in terms of using different datasets are shown in Table 3.4 and in Figure 3.4.

We also analyzed the performance of the four algorithms with respect to Un-normalized data and normalized data. Other parameters were also considered such as number of iteration and time taken by each algorithm which directly associated with the performance of an algorithm. Moreover, number of cluster instances with their percentage also shown in the Table 3.5. Un-normalized data may have some attributes which may be in kilograms and

another may be count. We compared the given algorithms with respect to un-normalized dataset in order to see their performance in terms of running time of building a clustering model. The results of the comparison are demonstrated in Table 3.5 and also shown graphically in Figure 3.5.

Table 3.5 Time Taken by Different Algorithms using Un-normalized Datasets

Algorithm	Number of iterations	SSE	Time	Clustered instances
K-means	7	62.14	0	100 (67%) 50 (33%)
Expectation Maximization	16	NA	1.08	48 (32%) 50 (33%) 29 (19%) 23 (15%)
Filtered Clustering	7	62.14	0.05	100 (67%) 50 (33%)
Make density Based Clustering	7	62.14	0.05	100 (67%) 50 (33%)

One of the interesting observations can be drawn from Table 3.5 which is, while the value of the parameter SSE is same in all the algorithms, the *k*-means algorithm has taken least amount of time as compared to other algorithms in Table 3.5.

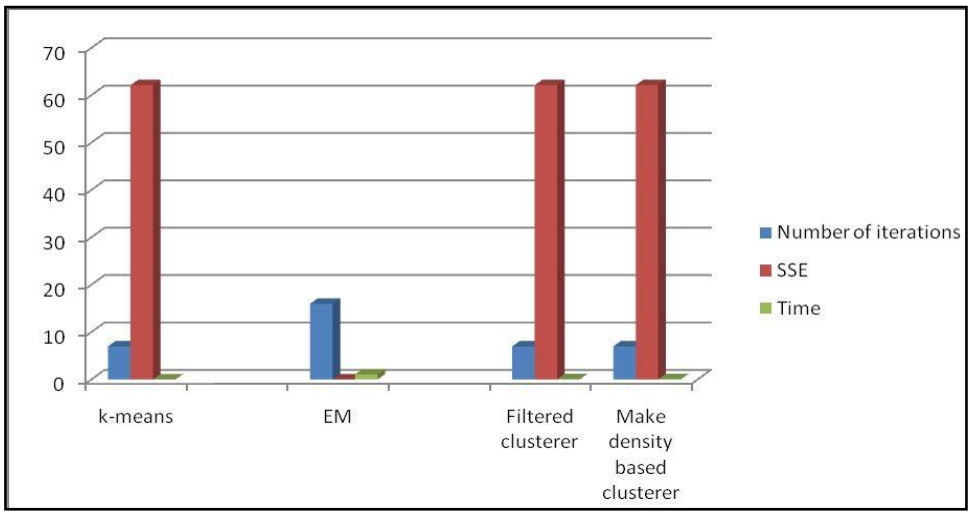


Figure 3.5 Time Taken by Different Algorithms using Un-normalized Datasets

We also compare the algorithms by normalizing the given dataset. The need for normalization of data arises from the fact that, there are several attributes in the data set which are not normalized. Hence, we make them normalized by rescaling one or more attributes to the range of 0 to 1. This means that the largest value for each attribute is 1 and obviously the smallest value is 0. Normalization is also used when we do not know how the

data is distributed or it may also happen that the distribution is not Gaussian (bell curve). Normalization is useful when the data has varying scales and the algorithm being used does not make assumptions about the distribution of your data, such as k-nearest neighbours and artificial neural networks.

Table 3.6 Time taken by Different Algorithms using Normalized Datasets

Algorithm	Number of iterations	SSE	Time	Clustered instances
k-means	7	62.14	0.06	100 (67%) 50 (33%)
Expectation maximization	16	NA	0.92	48 (32%) 50 (33%) 29 (19%) 23 (15%)
Filtered clustering	7	62.14	0.05	100 (67%) 50 (33%)
Make density based clustering	7	62.14	0	100 (67%) 50 (33%)

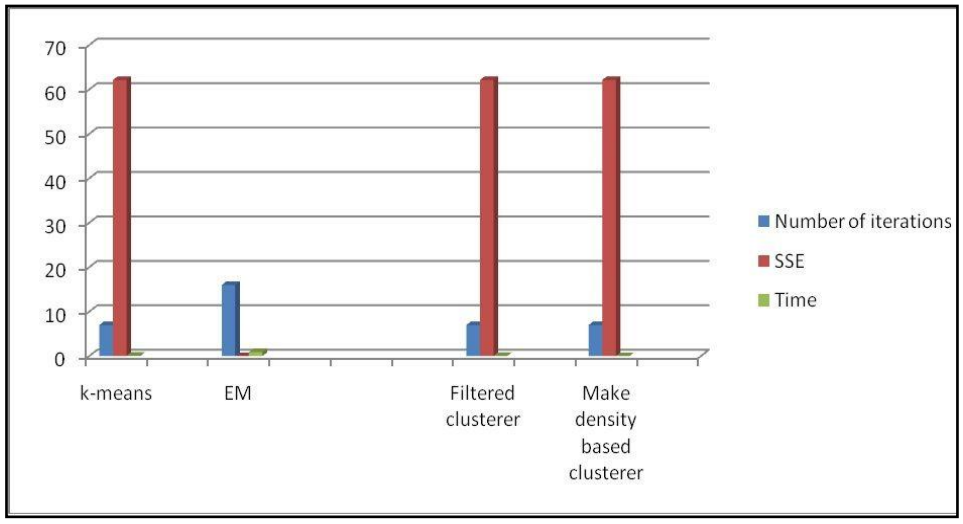


Figure 3.6 Time taken by Different Algorithms using Normalized Datasets

Moreover, the expectation maximization algorithm has taken maximum number of iteration. The observations made from the comparison of various clustering algorithms with respect to different parameters are summarized below:

1. While the k-means algorithm performed well as compared to other algorithms when the initial value of k is 2, the expectation maximization algorithm took significantly more time to build a model. Moreover, it is interesting to note that as the value of k gets increased the time taken by this algorithm reduces.

2. When we make change in the size of dataset it does have minor effect on the time taken by k-means algorithm. When the size of dataset is increased from 200 to 400, the time taken by k-means is insignificant. On the other hand, with the increase in the size of dataset, expectation maximization took highest time.
3. When the four algorithms are applied to different datasets, we found that k-means took the least amount of time to build a model. As expected, the expectation maximization took the highest amount of time. It is interesting to note that when the expectation maximization algorithm is executed using segment test dataset the time taken is as high as 51.52 seconds.
4. The algorithms were also compared using normalized and un-normalized data. The need for normalization arises from the fact that the dataset considered are of varying scale. In order to bring them to a uniform scale we used normalization. Moreover, The number of iterations and sum of squared error are same in both the cases, however k-means algorithm took lesser amount of time using normalized dataset.

3.4 Summary

A large amount of raw data which is generated as a result of user interaction with the learning management systems usually contains undesirable data such as noise, and outlier which needs to be removed before applying data mining algorithms. With the large number of clustering algorithms, it is difficult for a user to select the right algorithm for the problem being solved. Therefore, it becomes necessary to make a comparative analysis of all the available algorithms in order to find the most appropriate for the task at hand. Experimentally, we discovered that it is the k-means which took the least amount of time, expectation maximization took largest amount of time to build a model. We considered five of the most widely used clustering algorithms with five different parameter such as size of dataset, un-normalize dataset among others for making comparative analysis.

Chapter 4

Clusters' Analysis for Learners' Classification

The clusters obtained through *k*-means need to analyze in order to make sure that they represent the learners' profile correctly. Available data mining tools don't provide the option by which clusters can be validated. Hence, one of the objectives of this chapter is to bridge this gap by employing cluster validation methods namely 'elbow' and 'silhouette method' for the analysis of clusters. The validated clusters will be used further for classifying a new learner into its appropriate cluster. Moreover, the evaluation of classifier is equally important as it ensures that a learner falls into its appropriate class which would be used for recommending different data mining courses.

4.1 Data Collection and Pre-processing

Data pre-processing is the pre-requisite in the process of web usage mining which involves transforming raw data into understandable format. Learners' usage data found in often contains outlier, noise and other undesirable elements. These unwanted data need to be removed in order to prepare data for applying data mining algorithms. This section discusses Moodle, a learning management system which has been used as a learning platform for collection of data.

4.1.1 Moodle System

Moodle is an open source learning management system (LMS) which helps educators create effective online learning communities [102][229]. While there are a large number of LMS's, Moodle is one of the widely used collaborative LMS due to its open-source nature and rich functionalities which make it a natural choice for our purpose. Moreover, it can be run on any platform due to its implementation in java. It is provided freely and currently, it is the most popular e-Learning platform with over 85,000 registered sites worldwide, over 8 million courses, almost 76 million students, and over one million teachers. The different functionalities of Moodle are organized nicely into different modules such as the assignment module, quiz module, chat module, messaging module among others. While assignment module allows the instructor to gather, review and grade learners' assignment, quiz module

helps instructors to measure learners' comprehension of the learning material. Each activity performed by a learner in Moodle is recorded such as time spent on the forum, time taken to solve quiz and total time taken in doing assignment etc. Moodle is used by universities, community colleges, business and even individual instructors to add web technology to their course. Moodle is also used by learners for several purposes such as to communicate with their teachers and peers regarding their learning problems, to post a question on the forum, to read messages on the forum, to create a new topic, to answer the question, to visit resources, to access resources among others. Moodle is freely available on the web at (http://www.moodle.org).It allows users to have access to its source code and manipulate it in order to add new functionality to it. Moreover, unlike other learning management system such as *ATutor, Eliademy, Formal LMS, and OLAT* [203], it records each and every user activity in the log file of a web server. A log file collects all the data including quiz, assignment, chat, forum among others which comes from user's interaction with the site. The interaction involves any learning activity performed by the user on the site such as click stream data.

The usage data of undergraduate students on different data mining courses have been collected from the *Department of Computer Science and Engineering, Amrapali group of institute*. Although presently more than 32 subjects are being taught in the department, we have taken data on data mining courses as it uses a high number of Moodle activities such as assignment, quiz, messaging, and forum among others. Each student shares some attributes related to courses and activities performed by a learner on Moodle as shown in Table 4.1.

We use the latest version of moodle which is moodle 3.5 as shown in Figure 4.1 and has been set up using a client server model in which moodle is installed on the server and logins were provided to each learner in order to access the moodle system. The administrator has all the permissions such as creating an account, deleting an account, and changing the role of a user among others. These permissions are automatically assigned to the administrator at the time of moodle installation. However, a learner has been assigned limited permission. The learners also use collaborative means such as the forums, chat and messaging among others in order to complete their tasks. Learners' interaction with moodle results in the generation of data in the log file of the moodle server which was later extracted and pre-processed.

After collecting learners' usage data, the next step is to perform data pre-processing over learners' usage data in order to make it suitable for the next phase which is the application of k-means algorithm to the pre-processed data. Although data pre-processing consists of

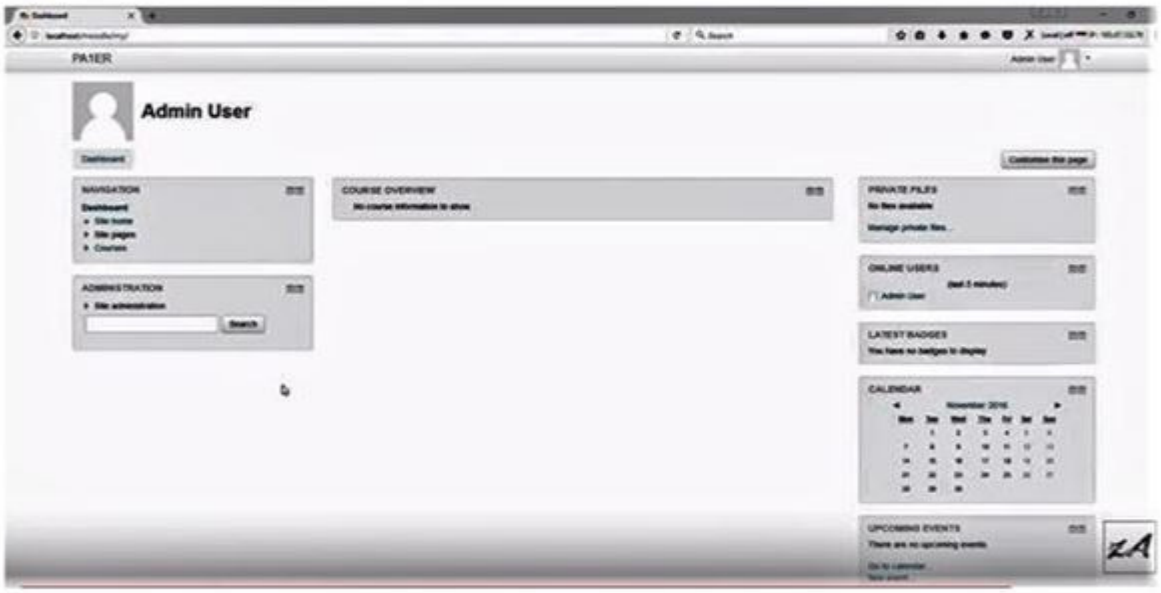


Figure 4.1 An Interface of Moodle 3.5

several steps such as data cleaning, user identification, session identification and path completion among others, all these steps are not required in this research during data pre-processing. In the subsection below, we briefly describe those steps which are required for data pre-processing:

4.1.2 Creating Summarization Table

The summarization table is created with the intent to put together all the required data in one table which is scattered in different tables in moodle database. Although the moodle database contains a lot of information in the form of tables about admin, courses, and users among others, we are not interested in all those information as it does not serve our purpose. The summarization table represents information from database at the required level (student). As usage data is spread over several tables hence a summarization table has been created in order to represent only required information which serve our purpose. The summarization table has a summary per row about all the activities done by each student during the course and the final mark obtained by each student in each course. The summarization table is shown in Table.4.1.

4.1.3 Data Transformation

The data must be exported into the required format which is ARFF (attribute relation file format). ARFF is an ASCII text file that describes a set of instances which share a set of

attributes. The file contains two sections namely header and data. The header section of the ARFF contains name of the relation, set of attributes and their types. The data section contains the actual data in the form of instances. This format is supported by weka and is suitable for data mining algorithms. The data in this research was initially in excel file which needs to be converted into CSV file format. Then we transform to ARFF file format. Although data pre-processing includes several sub-steps such as data cleaning, pattern identification, user identification, session identification and path completion, all these steps are not required in this research because user identification and session identification are decided by providing logging to the user. A user provides authentication details before logging into the system and he/she also logs out once he finished.

Table 4.1 Attributes Shared by Each Learner

S.No	Attribute Name	Description
1	c_id	Course identification
2	no_assign_comp	Number of assignment completed by student
3	no_quiz_comp	Number of quizzes completed
4	no_quiz_passwd	Number of quizzes passed
5	no_quiz_failed	Number of quizzes failed
6	messg_sent_chat	Number of messages sent to chat
7	messg_sent_teacher	Number of messages sent to the teacher
8	messg_sent_forum	Number of messages sent to the forum
9	messg_read_forum	Number of messages read on the forum
10	time_assignment	Total time spent on assignment
11	time_quiz	Total time spent on quiz
12	time_forum	Total time spent on forum
13	marks	Final marks obtained by students

4.2 Application of *k*-Means to Learners' Data

We used weka 3.8.1 version in order to execute K-means algorithm. Although there are other data mining tools both free and commercial, we preferred Weka due to its open-source nature, user-friendliness, platform independence, the availability of wide range of algorithms, better data preparation tools and its support for a very large dataset. Moreover, building models, validating them, excellent visualization tools are some of the unique features of

Weka that help understand the models, unlike other data mining tools. Weka also keeps on incorporating new algorithms as they appear in the research literature.

K-means is one of the widely used algorithms for building group of data objects in such a way that objects belonging to the same group have maximum similarity and the objects falling in different clusters have minimum similarity. The algorithm aims to minimize the sum of the points to centroid distance, summed over all k clusters. Since it is an unsupervised algorithm, the class labels are not known in advance and only the data and its features are available in a dataset. The algorithm generates clusters based on some common characteristics of objects. One of the distinguishing features of the *k*-means algorithm is to determine the optimal number of cluster. The process of reassigning data points to cluster and recomputing cluster centroid is repeated until there is no change in the mean value of cluster.

The dataset at present is of 100 students as shown in Table 4.2. It is important to note that Moodle (Modular object-oriented development learning environment) is not just about text, images or links. The philosophy of Moodle is that learning is only effective when constructing something for others to experience. Moodle was born with collaboration in mind. Following this philosophy, we define a learner to be active if he/she has actively used the collaborative platform such as the forum, chat, messaging among others while doing quizzes and assignments. This is also called active learning. Active learning involves reading (reading messages posted by peers on the forum, reading messages sent by the teacher), writing (creating a topic, replying to a query, discussion (discussing a topic created by some learner) while solving problems. Initially, the learners' usage data is extracted from moodle server and then imported into excel file format (.xls), which was then converted to CSV (comma separated value), a native file format of Weka. The CSV file format is a set of records with a comma between items. Finally, CSV was converted to ARFF (attribute relation file format). This is the standard way to organize dataset consisting of independent, unordered instances which do not have any relationships with them. It is interesting to note that, WEKA prefers to load data in ARFF format. It is an acronym of CSV file format where a header is used to that provides metadata about the data types in the columns. Values in the raw data section that have a question mark symbol (?) indicate an unknown or missing value. The format supports numeric and categorical values but also supports dates and string values. The dataset by default is not in ARFF format, in fact it is most probably in the format CSV. This is a simple format where data is represented in rows and columns and comma is used to separate the values on a row.

Table 4.2 Snapshot of Actual Dataset

Learner _id	Quiz-1	Quiz-2	Quiz-3	Quiz-4	Quiz_ done	N_Quiz _Pass	N_Quiz _fail	Quiz_ marks	Assign_ done
L1	7.71	10	10	6.55	4	4	0	34.26	9
L2	6.95	9.09	8	6.67	4	4	0	30.71	7
L3	5.63	7.5	6	4.55	4	3	1	23.68	9
L4	7.86	8.18	10	5	4	4	0	31.04	10
L5	4	3	2	4	4	0	4	13	1
L6	7.32	9.09	9	7.55	4	4	0	32.96	7
L7	6.65	8.18	10	6.33	4	4	0	31.16	10
L8	8.63	6.5	5.4	6.9	1	1	3	27.43	2
L9	5.68	6.36	7	4.55	4	3	1	23.59	8
L10	3.73	9.55	10	7.88	4	3	1	31.16	9
L11	2.5	1.5	3.5	2.5	0	0	4	10	1
L12	8.16	8.18	6	6.64	4	4	0	28.98	10
L13	2.1	3.1	1.5	2.4	0	0	4	9.1	1
L14	1.5	2.5	3.1	2.9	0	0	4	10	2
L15	6.66	7.27	10	6.36	4	4	0	30.29	9
L16	8.29	9.55	9	6.48	4	4	0	33.32	10
L17	7.57	7.5	9	5.64	4	4	0	29.71	9
L18	4.84	7.27	10	5.45	4	4	0	27.56	9
L19	7.66	7.27	10	4.73	4	3	1	29.66	10
L20	7.8	5.23	7	7.33	4	4	0	27.36	7
L21	4.5	2.1	3.5	2.9	0	0	4	13	1
L22	4.8	7.73	6	6.48	4	3	1	25.01	7
L23	5.68	9.09	9	7.55	4	4	0	31.32	8
L24	3.91	7.27	9	1.61	4	2	2	21.79	6
L25	2.27	5.45	10	4.36	4	2	2	22.08	5
L26	6.66	5.45	9	8.73	4	4	0	29.84	10
L27	1.25	2.1	1.9	2.7	0	0	4	7.95	0

The problem with the *k*-means algorithm is to determine the exact number of clusters present in the dataset. Moreover, it requires a user to specify the desired number of cluster (k). The actual numbers of cluster present in the dataset were validated by inputting different values of k. Hence, we found that we were just getting redundant clusters when k is set to 4. This is one of the ways to validate the clusters. There are other well-known methods for determining the right number of the cluster which will be discussed in the later section.

The default distance measure in Weka is the 'Euclidean distance' in which a centroid is the mean of all the points in a cluster. It is the ordinary straight line distance between two points in Euclidean space. The application of K-means algorithm to the data generated three

clusters namely 'not active'', 'average' and 'active' as shown in Table 4.3. We used weka 3.8.1 version in order to execute K-means algorithm. Although there are other data mining tools both free and commercial, we preferred Weka due to its open-source nature, user-friendliness, platform independence, the availability of wide range of algorithms, better data preparation tools and its support for a very large dataset. Moreover, building models, validating them, excellent visualization tools are some of the unique features of Weka that help understand the models, unlike other data mining tools. Weka also keeps on incorporating new algorithms as they appear in the research literature.

Furthermore, Moodle allows us to assess the performance of a learner using quiz and assignment modules. Hence, we took four quizzes on different topics of data mining such as introduction to data mining, association rules, classification and prediction and clustering. Each quiz carries 10 marks. We also gave ten assignments to learners in order to find their understanding of the subject. We make sure that a learner uses collaborative means such as the forum, chat, and messaging while doing quizzes and assignments. In order to achieve this goal, we set up Moodle in such a way that keeps track whether learners have used them(collaborative means) adequately or not.

Moreover, we also define three clusters namely "Not Active Learner", "Average Learner" and "Active Learner". If a learner scores 40% marks or below in quizzes and complete less than 5 assignments and does not use collaborative means in creating sufficient number of topics on forum, reply sufficient number of posts, and read adequate number of posts while doing quizzes and assignments then he/she falls in "Not Active Learner". On the other hand, all those learners who fall in the cluster "Average Learner" must have scored between 40% and 60% in quizzes and have completed all the assignments and also used resources sufficiently. In addition, for a learner to be in the "Active Cluster" he/she must have scored at least 75% on quizzes and completed all the assignments and have used the sufficient number collaborative resources in order to complete quizzes and assignment.

Furthermore, we have categorized all the attributes into two major type. The first type of attributes includes no_assignment_comp, no_quiz_comp, no_quiz_pasd records, no_quiz_fld and Marks by which we evaluate the performance of a learner in a particular subject. In order to find the activity level of a learner, the attributes such as no_messg_sent_to_chat, no_messg_sent_to_forum, no__messg_read__on_forum, Time_assignment, Time_quiz and Time_forum have been taken as the second type. Both categories of attributes are necessary as they characterize a learner adequately. Although there are many other attributes in the database, we chose only these attributes because they help us to understand the performance

of a learner in a particular subject, as well as whether the work was completed within specified time and using required number of resources or not. The above attributes capture the answer to the following questions:

a. Did they (learners) respond to the query sent by their peers through the forum?
b. Did they use the chat module to communicate with their teacher regarding their learning problems?
c. Did they read the messages posted by their peers?
d. Did they send the sufficient number of messages to their teacher?
e. Did they spend the minimum amount of time in solving quizzes?
f. Did they create a topic on the forum?
g. Did they complete all their assignments?
h. How many assignments did they complete?
i. How many assignments did they not complete?
j. How many quizzes did they pass?
k. How many quizzes did they fail?
l. Did they visit a sufficient number of resources?
m. Did they access adequate resources in order to solve quizzes and assignments?
n. How many resources did they spend their time on(less than 30 seconds)?
o. What was the grade obtained by learners in quizzes?
p. How much they scored in total in each course?
q. How many grades did they obtain in each course?
r. How many messages did they send to their peers?
s. How many messages did they read?

Furthermore, there are a large number of learning styles which a learner can follow, but Moodle's design supports 'social constructionist pedagogy' which suggests that students learn best when they learn through collaboration. The collaborative glossary is one of the most popular terms in Moodle according to which rather than creating a glossary of your own, why not have the students create it as they encounter unfamiliar terms. Each member of the group is supposed to contribute a term, a definition or comments on the submitted definition.

Once the required data is extracted from moodle server, the next step is data pre-processing which is an essential step of data mining process and must be performed in order to make the data suitable for data mining algorithms. Hence, the data stored in Moodle's server was pre-processed as it contains unstructured, noisy and irrelevant data. We transformed raw data

through various data pre-processing steps such as data selection and data summarization [68][132]. Although the Moodle database is huge and contains several pieces of information about learners, all this data is not relevant for our purpose. Hence, we used data selection in order to achieve our goal. Specifically, in this step data relevant to the analysis tasks are retrieved from the database. The second step is 'data summarization' which summarizes evaluation data which consist of both primitive and derived data. Another step is 'data discretization' which is one of the most influential data pre-processing tasks. More specifically, this step enables the algorithm to generate data mining model efficiently. Finally, we perform data transformation steps. The need for performing these steps arises due to the following reasons. The first reason is that the data is scattered in several tables. Hence the data has to be extracted as per our requirements. Another reason is the raw data is not suitable for applying data mining algorithms as it contains missing and noisy data. The data discretization step of data processing is essential as it converts the data into a form which is interpretable and easily understandable by an instructor.

Table 4.3 Results of Applying k-means Algorithm to Learners' Dataset

Attribute	Full Data(100)	Cluster#0 Not Active (9)	Cluster#1 Average (29)	Cluster#2 Active (62)
Quiz1-Intro_topics dwm(10)	5.78	1.5	3.54	7.45
Quiz2-Association Rules(10)	6.23	1.71	3.69	8.07
Quiz3-Classification and Prediction(10)	6.52	2.19	3.98	8.33
Quiz4-Clustering(10)	5.90	1.91	3.65	7.53
Quiz_done	2.86	0.22	3.10	3.12
Quiz_marks(40)	24.44	7.32	14.87	31.40
Assign_done(10)	6.74	1.88	4.51	8.48
Time_assignment(in hrs)	8.01	2.77	4.93	10.20
Msg_sent_teacher	44.34	7	33.31	54.91
Msg_sent_forum	53.53	10.44	29.44	71.04
Msg_read_forum	65.08	16.55	36.13	85.66
Msg_sent_chat	51.51	8.77	28.89	68.29
Time_spent_quiz(in hrs)	10.88	4.44	5.75	14.20
Time on forum(in hrs)	16.69	4.66	6.27	23.20
Access Resources	70.64	51.22	46.79	84.61
Num_Access<30sec	7.60	4.22	8.20	7.80
Resources_Visited	38.09	21.88	39.17	49.29
Quiz_grade(10)	6.18	1.803	3.74	7.95
Course_total	64.69	23.95	49.58	77.67

After data pre-processing, the k-means algorithm was applied to the transformed data which generated three clusters namely 'cluster0', 'cluster1', and 'cluster2' respectively. As it can be seen in Table 4.3, cluster0 represents 'Not Active' learners, which suggests that these learners have got poor marks and have not used collaborative platform enough while solving assignments.

Cluster1 characterizes 'Average' learners whose performance lies between cluster0 and cluster2. Finally, cluster2 shows 'Active' learners as their performance is excellent for quizzes, assignments and also used collaborative platform sufficiently. Moreover, the number of instances and percentage of instances of each cluster are also shown in Table 4.3. The three clusters consist of 9%, 29% and 62% of total instances of dataset respectively. It is also interesting to note that learners of cluster2 have outperformed learners of cluster0 and cluster1. This is simply because, they (cluster2) have not only scored high in quizzes and assignments but also been active in collaborative platforms such as the forum, messaging, chat etc. In addition to that, learners of cluster0 is characterized by low score in quizzes(7.3 out of 40), low assignments done(1.8 out of 10), low number of messages read and few numbers of messages sent on forum(10,16), small number of resources visited and accessed(51,21), poor quiz grade obtained(1.8). On the other hand, cluster2 is characterized by high marks obtained in quizzes(32 out of 40), more than 80% assignment done(8.48 out of 10), high number of messages sent to teacher(54), high amount of resources visited and accessed(84,49), high quiz grade obtained(7.9 out of 10), excellent course total(77 out of 100). Moreover, cluster1 is characterized by moderate score in quizzes (14 out of 40), average number of assignments done (4.5 out of 10), reasonable number of messages sent to teacher (33), average number of messages sent on forum (29), moderate number of messages read on forum (29), considerable amount of time spent on quizzes and forum (5.7,6.2), moderate number of resources accessed and visited (46,19). In the next section, we evaluate the quality of clusters obtained in this section.

4.3 Experimental Evaluation of Clusters

Cluster evaluation refers to determining the quality of clusters obtained through the k-means algorithm. There are several methods for performing these tasks such as The elbow method, X-means clustering, information criteria approach, The silhouette method, cross-validation among others. However, The silhouette method and The elbow method are widely used due to

their wider acceptance in terms of performance. These two methods have been discussed below in order to validate the quality of clusters.

4.3.1 Silhouette Method

This is one of the widely used methods for the evaluation of the quality of clusters. This approach[1] is used to interpret and validate the consistency of an object within a cluster. In the simplest terms, the silhouette plot tells how well separated and well formed are the resulting clusters in a given dataset. In this approach, a Silhouette plot is built using two parameters i.e k (number of clusters) and silhouette values. The silhouette value for the i_{th} point, S_i, is computed using equation (1). Where a(i) is the average distance from the i_{th} point to the other points in its cluster, and b (i,k) is the average distance from the i_{th} point to the points in another cluster k [106]. This value (silhouette) is calculated for each point in a dataset which indicates how well a point lies in its own cluster as compared to neighboring cluster. The silhouette value ranges from -1 to +1. If most points in a cluster have high silhouette (close to 1) value then it suggests that the cluster is compact and well separated from neighboring clusters. Moreover, if some points have a zero value, then it suggests that the points are not distinct in one cluster or another. Furthermore, if some points have negative silhouette value then it indicates that the points are assigned to the wrong clusters.

In Figure 4.2(a), it can be seen that, several points in all the three clusters have a large silhouette value which is greater than 0.8, which further indicates that, these clusters are somewhat separated from their neighbouring clusters. Hence, the clusters in Figure 4.2 suggest three clusters which are consistent with the number of clusters found through the K-means algorithm.

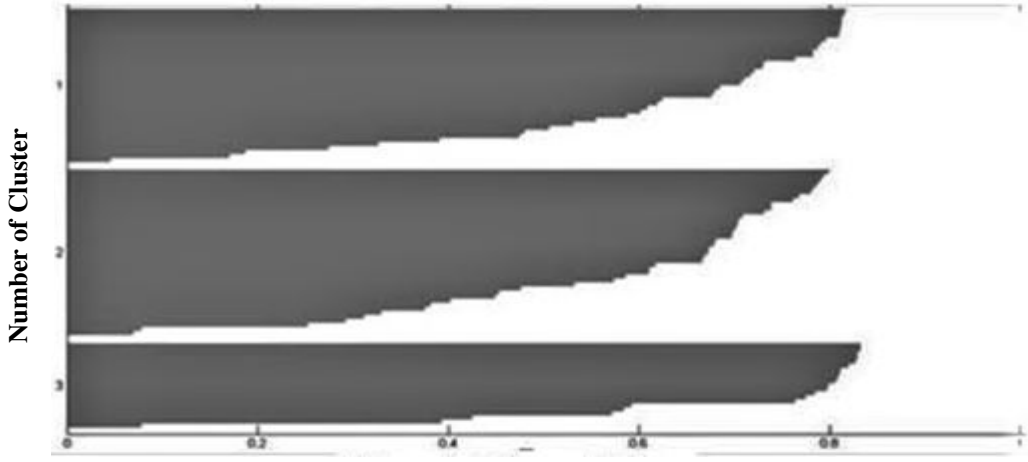


Figure 4.2 Silhouette Value(a)

On the other hand, the value of k is also set to 2 in order to check the possibility of two clusters, which is shown in Figure 4.3 suggesting that the two clusters are not well separated. Furthermore, Figure 4.3 has few points with negative silhouette values, which indicate that these points are assigned to the wrong cluster. These points can be seen on the negative side of the x-axis. Hence, Figure 4.3 does not suggest two clusters.

Another way to compare the clusters shown in Figure 4.2 and Figure 4.3 is to find their average value by using the mean () function. This function is passed two arguments such as silh2 and silh3, where silh2 is the mean silhouette value of the two clusters and silh3 is the mean silhouette value of three clusters as shown below:

Si = (bi-ai)/ max (ai,bi) **…….(1)**

```
mean (silh3)
ans =0.6255

mean (silh2)
ans =0.6207
```

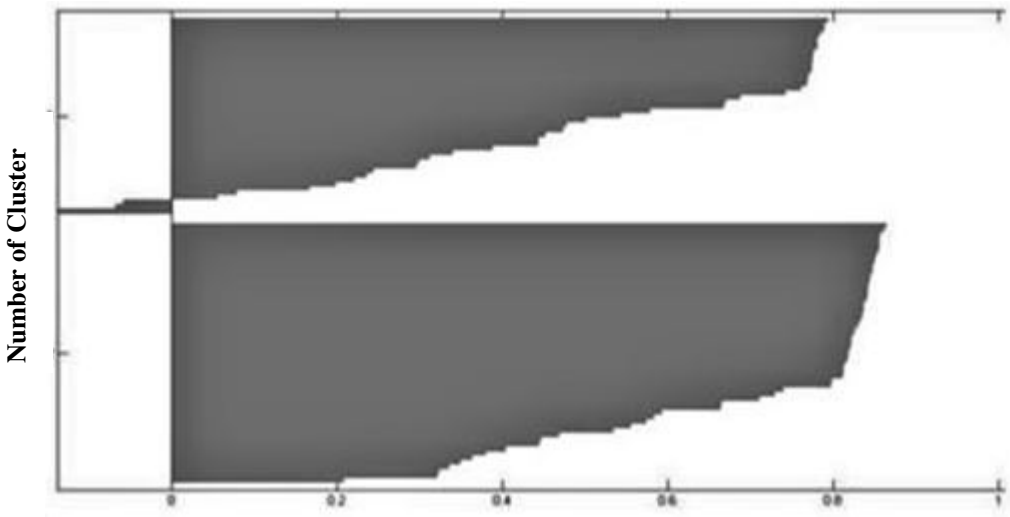


Figure 4.3 Silhouette Value (b)

It is clear from the above two values returned by the mean () function that the mean value of the two clusters is less than the mean value of the three clusters, which further indicates that the clusters in Figure 4.2 are well formed and compact than the clusters in Figure 4.3. This further confirms the results i.e the number of clusters (k=3). The silhouette plot and elbow method have been implemented in version 7.12.0.635 of MATLAB. While there are other similar tools providing same functionality as provided by MATLAB, it lets us to plot our data easily and then change colors, sizes and scales etc by using the graphic interactive tools. Moreover, MATLAB'S functionality can be greatly expanded by the addition of toolbox.

4.3.2 Elbow Method

The elbow is the one of the oldest methods for determining the true value of number of clusters in a given dataset. It is a visual method. This method works by first computing the sum of squared error for different values of k say (2,4,6,8....,.....). The sum of squared error is defined as the sum of the squared distance between each member of the cluster and centroid. Mathematically, it is represented as follows:

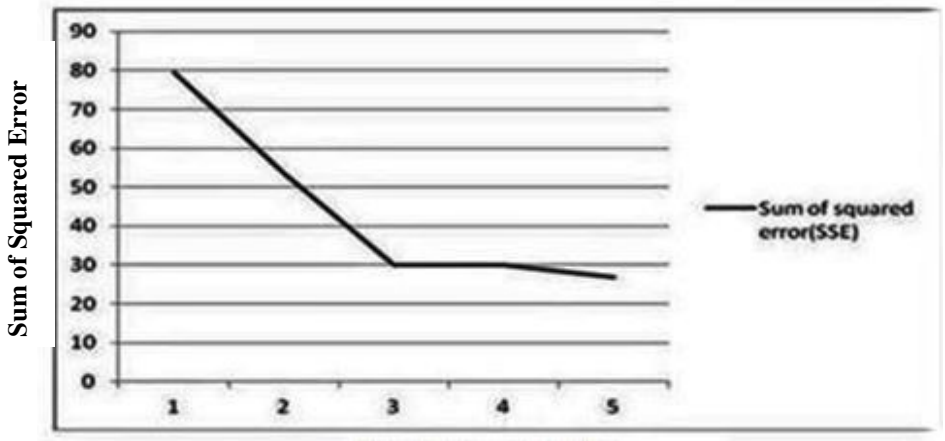
$$SSE = \sum K_i=1 \sum x \in c_i \quad dist(x,c_i)^2 \qquad \text{..... (2)}$$

The elbow method considers the percentage of variance as a function of the number of clusters [116]. The number of clusters should be chosen in such a way that adding another cluster does not add better modeling of data. We start with k=2 and keep on increasing the value of k in a step of 2. At some point of 'k' the value of SSE drops dramatically and if we further increase the value of 'k' then we reach in a state of plateau. This is the desired value of 'k'.

It is clear from Table 4.4 that the change of variance in the first two values of SSE (sum of squared error) is greater than the change of variance in the subsequent two values of SSE. Moreover, this change gets smaller and smaller as we move down the table. Continuing this, we reach a point when any further change in the value of k has no effect on the value of change of variance. This change in the value of k and the corresponding change in SSE are shown in Table 4.4. In Figure 4.3, there are two parameters namely k and SSE. The parameter k' is taken along the x-axis and the parameter SSE is taken along the y-axis. Here, k represents the number of clusters. The value of 'k' is increased from k=1 to k=5 and the corresponding effect on the value of SSE is observed. The value of SSE decreased constantly until the value of SSE reached to 30. When the value of k is increased further, an elbow effect is seen at k=3. The central idea of the elbow method is to choose k at a point where the value of SSE decreases abruptly.

Table 4.4 Number of Clusters and their Corresponding SSE

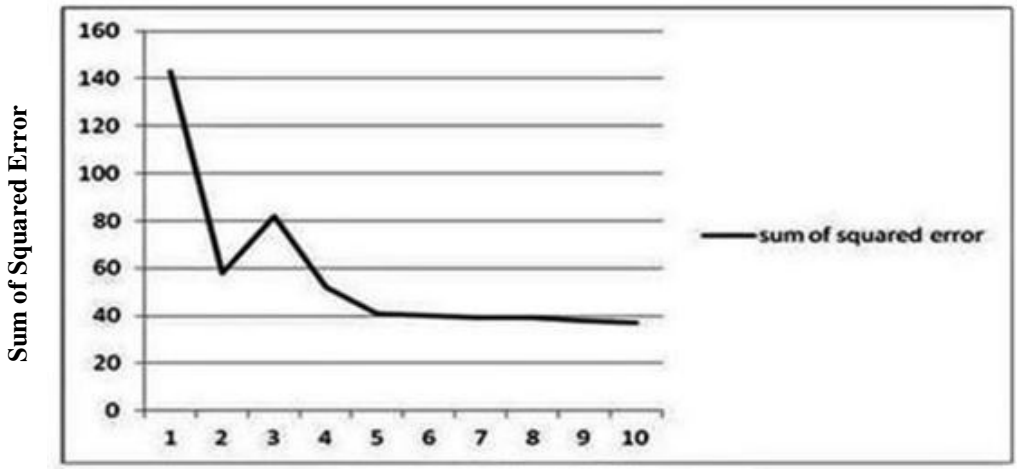
Number of Cluster(k)	Sum of Squared Error(SSE)
2	79.36
4	53.48
6	35.07
8	29.91
10	26.79



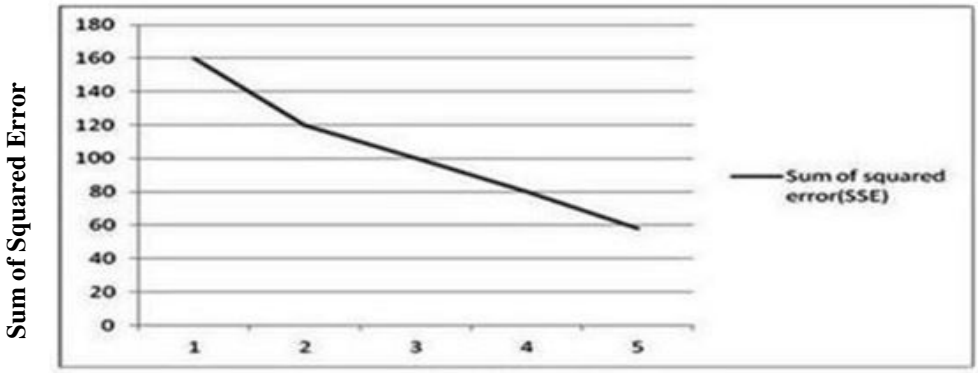
Number of Cluster (k)
Figure 4.4 Single Elbow Point

This abrupt change produces an 'elbow effect' which can be seen in the Figure.4.4, where the value of k can be easily determined right at the elbow point which is 3. Moreover, if we continue to increase the value of k then we may end up at a point where each data point is a cluster in itself and at this point, the change of variance is 0. However, it is not always possible to find unambiguously the exact number of cluster. The chart in Figure 4.5 shows that there are multiple elbow points. The first change could be seen at k=2, the second at k=3 and the third at k=5 which further suggest the presence of three clusters. Moreover, the line in Figure 4.6 suggests the absence of any elbow point. Because the value of SSE constantly decreases until the value of SSE becomes 120.

The change in the value of SSE corresponding to the value of k can be seen at SSE=120 in Figure 4.6. However the change is not abrupt, hence we can conclude that the line in Figure 4.6 does not have any elbow point. Consequently, it is very difficult to determine the number of clusters.



Number of Cluster (k)
Figure 4.5 Multiple Elbow Points



Number of Cluster (k)
Figure 4.6 No Elbow Point

In the previous sections, we found different clusters and validated them. In the following section, we analyze the clusters further obtained from the previous section with fuzzy c-means algorithm in order to find more accurately the position of each learner in each cluster.

4.4 Analysis of Clusters through Fuzzy C-Means

Fuzzy *c-means* is one of the widely used clustering algorithms in which each data point belongs to every cluster with certain value of membership grade which indicates the degree to which each data point belongs to a cluster [37]. The algorithm consists of the following steps:

1. Initialize $U = \left[u_{ij}\right]$ matrix, $U^{(0)}$
2. At k-step: calculate the centers' vectors $C^{(k)} = \left[c_j\right]$ with $U^{(k)}$

$$C_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}$$

3. Update $U^{(K)}$, $U^{(K+1)}$

4. $d_{ij} = \sqrt{\sum_{i=1}^{n} \left(x_i - c_i\right)}$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}}\right)^{2/(m-1)}}$$

4. If PU(k+1)-U(K)P<ε then STOP: otherwise return to step 2.

Here m is any real number greater than 1,
u_{ij} is the degree of membership of x_i in the cluster j,
x_i is the ith of d-dimensional measured data,

Table 4.5 Results of Applying Fuzzy c-Means Algorithm to learners' Dataset

	Fuzzy C-Means Membership distribution					Fuzzy C-Means Membership Distribution			
Id	**Not Active learner (16)**	**Active Learner (59)**	**Average Learner (25)**	**Strong Preferences**	**Id**	**Not Active learner (16)**	**Active Learner (59)**	**Average Learner (25)**	**Strong Preferences**
1	0.072975	0.857376	0.069649	Active	23	0.108869	0.680554	0.210577	Active
2	0.092734	0.570845	0.336421	Active	24	0.083737	0.506632	0.409631	Active
3	0.198386	0.686007	0.115607	Active	25	0.03687	0.272303	0.690827	Average
4	0.514034	0.394008	0.091958	Not Active	26	0.12206	0.761578	0.116362	Active
5	0.013944	0.053813	0.932243	Average	27	0.032065	0.132142	0.835793	Average
6	0.1309	0.562753	0.306347	Active	28	0.031354	0.148373	0.820273	Average
7	0.094419	0.555045	0.350536	Not Active	29	0.02725	0.924701	0.048049	Active
8	0.1054649	0.5654279	0.3291072	Active	30	0.014284	0.963268	0.022448	Active
9	0.056279	0.568117	0.375604	Active	31	0.026095	0.10466	0.869245	Average
10	0.067011	0.746398	0.186591	Active	32	0.031529	0.925015	0.043456	Active
11	0.013812	0.060898	0.92529	Average	33	0.025453	0.919095	0.055452	Active
12	0.199683	0.696318	0.103999	Active	34	0.018457	0.08241	0.899133	Average
13	0.417958	0.170916	0.411126	Not Active	35	0.061137	0.442255	0.496608	Average
14	0.84951	0.120757	0.029733	Not Active	36	0.044145	0.144047	0.811808	Average
15	0.103165	0.833365	0.06347	Active	37	0.030263	0.932106	0.037631	Active
16	0.033265	0.930148	0.036587	Active	38	0.037345	0.312292	0.650363	Average
17	0.021994	0.947874	0.030132	Active	39	0.041908	0.138529	0.819563	Average
18	0.036061	0.873953	0.089986	Active	40	0.065058	0.868959	0.065983	Active
19	0.023372	0.925205	0.051423	Active	41	0.130474	0.787398	0.082128	Active
20	0.124206	0.509967	0.365826	Active	42	0.03815	0.87118	0.090669	Active
21	0.479642	0.106904	0.413455	Not Active	43	0.062684	0.606104	0.331213	Active
22	0.070913	0.684277	0.24481	Active	44	0.134708	0.775227	0.090064	Active

	Fuzzy C-Means Membership distribution					Fuzzy C-Means Membership Distribution			
Id	**Not Active learner (16)**	**Active Learner (59)**	**Average Learner (25)**	**Strong Preferences**	**Id**	**Not Active learner (16)**	**Active Learner (59)**	**Average Learner (25)**	**Strong Preferences**
45	0.067862	0.841946	0.090192	Active	67	0.246703	0.650941	0.102356	Active
46	0.050572	0.163059	0.786369	Average	68	0.244966	0.651574	0.103459	Active
47	0.141365	0.760166	0.09847	Active	69	0.256674	0.600045	0.143281	Active
48	0.051005	0.161346	0.787649	Average	70	0.018211	0.099727	0.882062	Average
49	0.117378	0.762009	0.120612	Active	71	0.009668	0.054457	0.0935875	Average
50	0.653488	0.273256	0.073256	Not Active	72	0.014716	0.089834	0.089545	Average
51	0.246135	0.691872	0.061993	Active	73	0.020353	0.133506	0.846141	Average
52	0.05633	0.924774	0.018895	Active	74	0.023684	0.163107	0.813209	Average
53	0.030127	0.960009	0.009865	Active	75	0.132941	0.730736	0.136323	Active
54	0.012443	0.98346	0.00407	Active	76	0.153053	0.712055	0.134893	Active
55	0.009847	0.986882	0.003271	Active	77	0.157215	0.711964	0.130821	Active
56	0.018535	0.975054	0.00641	Active	78	0.262187	0.582865	0.154948	Active
57	0.017575	0.976408	0.006017	Active	79	0.275091	0.572756	0.152154	Active
58	0.017387	0.976544	0.006069	Active	80	0.01808	0.098025	0.883895	Average
59	0.017816	0.975824	0.006359	Active	81	0.020493	0.115219	0.864289	Average
60	0.016851	0.977167	0.005982	Active	82	0.023237	0.134077	0.842686	Average
61	0.021025	0.971336	0.007639	Active	83	0.026177	0.158527	0.815296	Average
62	0.023565	0.967783	0.008652	Active	84	0.029695	0.181155	0.78915	Average
63	0.023146	0.968336	0.008518	Active	85	0.03439	0.227749	0.737861	Average
64	0.012779	0.983052	0.004169	Active	86	0.037747	0.261043	0.70121	Average
65	0.014611	0.980444	0.004945	Active	87	0.058993	0.045833	0.482674	Average
66	0.015115	0.980203	0.004681	Active	88	0.076604	0.737344	0.186052	Active

In this table, we can see that, each learner in the database belongs to each cluster with certain degree of membership. However, it is important to note that, the final position of a learner is determined by highest value of the membership in a cluster for instance, learner L1 has the highest membership value of 0.85, hence it falls in the active cluster. in other words, we can say that, strong preferences for this learner would be in active cluster.

	Fuzzy C-Means Membership distribution					Fuzzy C-Means Membership Distribution			
Id	**Not Active learner (16)**	**Active Learner (59)**	**Average Learner (25)**	**Strong Preferences**	**Id**	**Not Active learner (16)**	**Active Learner (59)**	**Average Learner (25)**	**Strong Preferences**
89	0.136255	0.743338	0.120406	Active	95	0.20083	0.06984	0.910077	Average
90	0.006525	0.30778	0.962697	Average	96	0.22525	0.076828	0.900647	Average
91	0.010139	0.039691	0.95017	Average	97	0.020758	0.071962	0.90728	Average
92	0.010628	0.041053	0.948319	Average	98	0.032941	0.103787	0.863272	Average
93	0.011435	0.042945	0.945619	Average	99	0.038047	0.115632	0.846321	Average
94	0.021671	0.075789	0.902539	Average	100	0.0352	0.1079	0.8569	Average

Table 4.6 Summary of Results of Applying Fuzzy c-Means Algorithm to Learners' Dataset

Fuzzy c-means Algorithm	**Not active**	**Average**	**Active**
Number of Learners Before Applying Fuzzy c-Means Algorithm	09	29	62
Number of Learners After Applying Fuzzy c-Means Algorithm	06	37	57

4.5 Learners' Classification Using Instance base Classifier (*K*-NN)

There are several classification algorithms reported in the literature [5]. However, we chose IBK (instance-based classifier)[17]. The main reason for choosing IBK for classification is due to the small size of the dataset and runtime of the algorithm is not a major consideration as classification involves negligible amount of time to classify a learner. The easiest way to find which instance or group of instances (neighborhood) in the training dataset is closest to the new instance is by computing the distance from each member of the training dataset to the new instance and then select the smallest. The time taken by the classifier depends on the dataset. There are several distance metrics for measuring the similarity between any two feature vectors (two instances in our case)[19]. The most widely used distance metric is 'Euclidean distance'. This distance metric works well if the data has real value input variable. Since this condition is satisfied by our dataset hence we used it. We consider the default

value of K (number of the neighbourhood) as 1 in IBK. However, it can be set to any other appropriate value.

4.6 Evaluation of Classifier

The evaluation of a classifier is equally important as it directly affects the results of a recommender system. For instance, if out of 100 instances, a classifier is able to classify 50 instances correctly then the accuracy of the classifier would be 50%. We used weka for evaluating the performance of IBK.

4.6.1 Cross Validation

Although weka offers several methods for evaluation such as supplied test mode, cross-validation, and percentage split, we chose cross-validation and test mode [23] for two reasons (i) It gives the best error rate (ii) Due to the limited data set. The tenfold cross-validation method divides the dataset into ten equal partitions. Nine parts of the dataset are used for training a classifier and the one part (test set) is used to evaluate its performance. This process of partition continues until each instance in the dataset is used as a test instance. Figure.6 shows that the IBK classifier has correctly classified 98% of instances. However, it could not classify two instances correctly. Another way to interpret this result is by using "confusion matrix" [25] as shown in Figure 4.7. The confusion matrix is a two-dimensional table with a row and column for each class. Each element of the matrix shows the number of test examples for which the actual class is the row and the predicted class is the column.

The number at the diagonal of the matrix represents correctly classified instances and the number of elements outside diagonal indicates incorrectly classified instances. Figure 4.7 also shows confusion matrix which contains 100 instances of the training dataset out of which 98 instances are predicted correctly which is also the sum of the number at diagonal (13+22+63=98) hence, the percentage of accuracy is 98%. Figure 4.7 also shows 'Kappa statistics' which shows a value of 0.96 which is close to 1.

This suggests the agreement of prediction with the true class. Moreover, mean absolute error (MAE) which provides the average magnitude of errors in a set of instances without taking into account their direction is 0.02. MAE considers the weight of individual differences equally. It is a measure of accuracy for continuous variables. However, this result

was expected as the dataset on which the classifier was trained has also been used for evaluation of the classifier.

```
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          98               98       %
Incorrectly Classified Instances         2                2       %
Kappa statistic                          0.9617
Mean absolute error                      0.0272
Root mean squared error                  0.1146
Relative absolute error                  7.7425 %
Root relative squared error             27.4221 %
Total Number of Instances              100

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   MCC      ROC Area  PRC Area  Class
                 0.929     0.012       0.929     0.929     0.929     0.917     0.958     0.872     cluster1
                 1.000     0.000       1.000     1.000     1.000     1.000     1.000     1.000     cluster2
                 0.984     0.028       0.984     0.984     0.984     0.957     0.978     0.979     cluster3
Weighted Avg.    0.980     0.019       0.980     0.980     0.980     0.961     0.980     0.969

=== Confusion Matrix ===

  a  b  c   <-- classified as
 13  0  1 |  a = cluster1
  0 22  0 |  b = cluster2
  1  0 63 |  c = cluster3
```

Figure 4.7 Evaluation of the Classifier IBK using Cross Validation Method

4.6.2 Test Mode

Another method to evaluate the performance of a classifier is to measure the error rate of the classifier on an independent dataset which is not used for training the classifier. Hence, we input independent 'test data' and the results obtained are shown in Figure 4.8. Various statistics are presented as part of the output as shown in Figure 4.8. The time taken to test model is very small. Moreover, it can be seen that the classifier is able to correctly classify the instance.

```
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.05 seconds

=== Summary ===

Correctly Classified Instances           1               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0129
Root mean squared error                  0.0137
Relative absolute error                  5.2632 %
Root relative squared error              5.2247 %
Total Number of Instances                1

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   MCC      ROC Area  PRC Area  Class
                 0.000    0.000       0.000     0.000     0.000     0.000        ?         ?       cluster1
                 0.000    0.000       0.000     0.000     0.000     0.000        ?         ?       cluster2
                 1.000    0.000       1.000     1.000     1.000     0.000        ?         1.000   cluster3
Weighted Avg.    1.000    0.000       1.000     1.000     1.000     0.000        0.000     1.000

=== Confusion Matrix ===

   a b c   <-- classified as
   0 0 0 | a = cluster1
   0 0 0 | b = cluster2
   0 0 1 | c = cluster3
```

Figure 4.8 Evaluation of the Classifier IBK using Supplied Test Mode

4.7 Summary

This chapter analyzed the clusters obtained by applying *k*-means algorithm to pre-processed data extracted from moodle server. The clusters were further validated by using various 'cluster validation methods' such as 'elbow' and 'silhouette'. Furthermore, in order to find those learners belonging to multiple clusters we also analyzed the obtained clusters using fuzzy c-means algorithm. The algorithm discovered few learners found in multiple clusters. Based on this information, relevant courses would be recommended to those learners. The validated clusters were used by the classifier namely IBK (instance based classifier) in order

to classify a new learner into its appropriate cluster. The accuracy of the classifier was evaluated using two well known methods such as 'cross validation' and 'test mode'. The next chapter recommends various data mining courses to learners based on their profile or the cluster to which they belong. The chapter also experimentally evaluates the quality of recommendations made to the learners using well known evaluation metrics such as precision, recall and F1 measures.

Chapter 5

Recommendation of Courses and their Evaluations

After classifying learners into their appropriate clusters, the next step in the process of recommendation is to recommend different data mining courses to learners based on their profile. Furthermore, the evaluation of recommendations is equally important in order to ensure that the proposed recommender framework is doing well. However, this presents several choices of evaluation metrics which makes it difficult to choose the most appropriate metrics due to the availability of different type of recommender algorithms and a variety of recommender tasks. Hence, one of the objectives of this chapter is to carry out an experimental analysis for finding the most suitable evaluation metrics which best matches with a given recommender algorithm and a recommender task. The outcome of this experimentation provides us few evaluation metrics which will be used for the evaluation of recommendation of data mining courses.

5.1 The Proposed Framework for Course Recommendation

The proposed recommender framework is shown in Figure 5.1 which has eight modules namely data collection, data pre-processing, cluster creation, cluster evaluation, learner's classification, classifier evaluation, course recommendation and evaluation of recommended item.

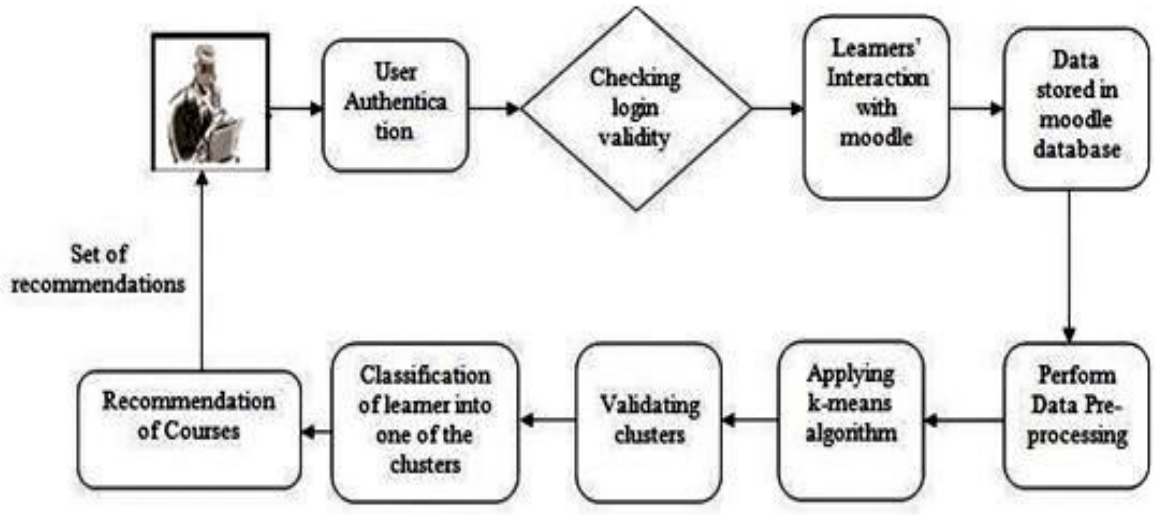


Figure 5.1 A Framework for Recommendation of Courses

While the tasks of data collection and data pre-processing are performed offline, the remaining steps have been performed online. The system aims to recommend different data mining courses to learners based on their profile. The profile is based on the clusters obtained through the application of k-means to learners' usage data. After the creation of clusters, the IBK (instance base classifier) classifier is used to classify a new learner into its appropriate cluster. For instance, if the classifier suggests "not active" cluster for a new learner then the course such as "introduction to data mining" would be recommended from the category "courses for the beginner" to him/her. This course would help the learner in improving the basic understanding of the subject which further leads to improvement in the academic performance of a learner. Moreover, if a classifier suggests that a learner falls in "average cluster" then he/she could be recommended "intermediate level course" and if a classifier discovers that a learner lies in "active category" then he/she would be recommended advanced courses such as "data mining and analysis". For better understanding of the system a flow chart is shown in Figure.5.2.

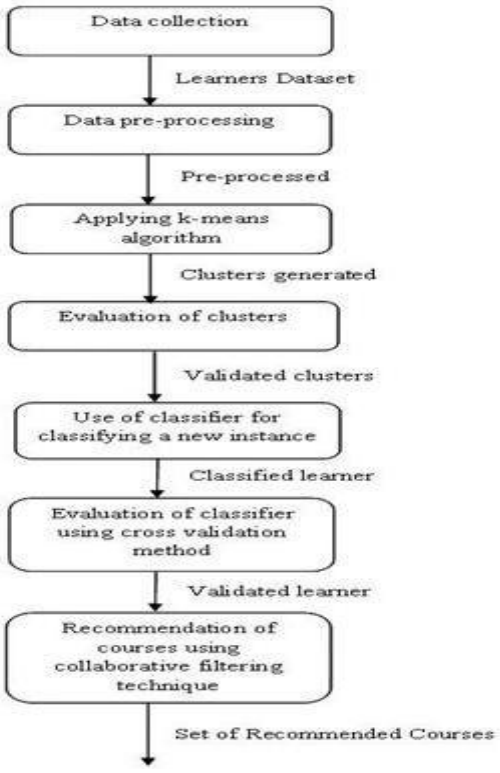


Figure 5.2 Flow Chart of Course Recommender System

5.2 Collaborative Filtering for Recommendations

Although there are several recommendation approaches such as collaborative filtering, content-based filtering and hybrid approach (a combination of collaborative and content

filtering), in this research collaborative filtering technique has been used due to the availability of 'user-item rating matrix'. Collaborative filtering (CF) technique works by matching the preferences of a new user with a group of users [26] [34] [58]. The preferences of learners have been collected on various data mining courses in the form of ratings from the students of undergraduate courses in the Department of Information Technology and Computer Science, Amrapali group of institute,Haldwani(Uttrakhand). The 'user-item ratings' are shown in the Table 5.2, Table 5.3 and Table 5.4 respectively.

There are several scaling methods used for determining user preferences in terms of a number (rating) [61]. The 'numeric scale method' which suits our requirement well has been used for finding ratings from users. A learner has provided ratings on a scale of 1 to 5. A rating of 1 refers to 'least liked' course and a rating of 5 means 'most liked course'. The user's ratings can be obtained either explicitly by interacting with users and asking them to provide ratings on items they have used [63] or implicitly by deducing user preferences through observing user behavior [65].

The database consists of 120 data mining courses which have been taken into consideration for recommendation to learners. Each course is assigned a unique course_id. For example, each course in the category named "courses for the beginner" is assigned a course_id which begins with C1 and ends with C40. Similarly, each course in "courses for intermediate" learner category is assigned a course_id which begins with C41 and ends with C80 and each course in "course for advanced learners" starts with C81 and ends with C120. Some of the courses from each category are shown in Table 5.1. A blank cell in the Table shows that no rating has been provided by a learner to that particular course. For instance, learner1(L1) has not given any rating to course C2. Similarly, learner2 has provided ratings to only course C2.

The ratings provided by learners in "user rating tables" such as Table 5.2, 5.3 and 5.4 have similar preferences to the learners in our dataset. The similarity of preferences of two users can be determined based on the similarity in the rating history of a user by using various similarity measures [67] such as cosine similarity and Pearson's similarity measure among others. However, we have used the "Pearson correlation" measure as it fits in this research well. The value of Pearson correlation is computed using equation (1) which involves the covariance of x and y and their standard deviations which is denoted by σ. The similar learners are computed based on the rating tables 5.2,5.3 and 5.4 as shown below. The collaborative filtering approach has been implemented using Netbeans 8.2 and Apache Mahout.

Table 5.1 A View of Different Categories of Data Mining Courses

Courses for Beginner	Courses for Intermediate Learner	Courses for Advanced Learner
Introduction to Data Mining	Data Mining Techniques	Data Mining and Analysis
Data Mining for Beginner	Data mining: The Tex book	Data Mining with R
Basic Concepts of Data Mining	Frequent Pattern Mining	Mining the Social Web
An Introduction to Data Science	Mining Text Data	Mining of Massive Datasets

Table 5.2 A View of Ratings Collected From Not Active Learner

Cid	Course name	L1 (5)	L2 (5)	L3 (5)	L4 (5)	L5 (5)
C1	Introduction to Data Mining	3	4	3		3
C2	Data mining for Beginner		3	5	5	4
C3	Basic Concepts of Data Mining	4	5		4	1

Table 5.3 A View of Ratings Collected From Average Learner

Cid	Course name	L21 (5)	L22 (5)	L23 (5)	L24 (5)	L25 (5)
C41	Data Mining Techniques	3	3	3	4	4
C42	Data Mining: The Text Book	4	3	5		3
C43	Frequent Pattern Mining	2	4		5	3

Table 5.4 A View of Ratings Collected From Active Learner

Cid	Course name	L41 (5)	L22 (5)	L43 (5)	L44 (5)	L45 (5)
C111	Data Mining with R	4		2		4
C112	Mining the Social Web	4	4	4	4	
C113	Mining of Massive Dataset		3		3	3

$$Pearson(x, y) = \frac{\sum (x, y)}{\sigma x \sigma y} \qquad \text{.....(1)}$$

Table 5.2 and 5.3 and 5.4 show the ratings collected from learners belonging to not active, average and active clusters along with course_id and course_name. After the application of collaborative filtering algorithm to these user item rating tables, the results are shown in Table 5.5 and 5.6 and 5.7 shows that, the recommender algorithm has provided a predicted rating of 5 for learner L4. However, not all these recommendations may be useful for the learner. Therefore, we need to find a way to filter these recommendations so that a learner only gets the best recommendations. In order to deal with this issue, we may select top N recommendations from the list for a learner by sorting those items by ratings and suggest N highest rated items.

Table 5.5 Courses Recommended for Not Active Learner

Learner_id	Course_id	Course_Name	Predicted Rating
L4	C3	Data Mining for Beginner	5.0
L2	C5	Introduction to Data Mining	5.0
L1	C17	An Introduction to Data Science	4.4
L3	C14	Basics Concepts of Data Mining	3.5

Table 5.6 Courses Recommended for Average Learner

Learner_id	Course_id	Course_Name	Predicted Rating
L24	C46	Machine learning: Wikipedia Guide	5.0
L25	C53	Data Mining Techniques	4.0
L26	C56	Mining Text Data	4.4
L27	C46	Data Mining Applications	3.5

Table 5.7 Courses Recommended for Active Learner

Learner_id	Course_id	Course_Name	Predicted Rating
L41	C103	Data Mining and Analysis	5.0
L45	C100	Mining the Social Web	4.6
L44	C86	Applied Data Mining	4.0
L43	C94	Data Clustering: Algorithms and Applications	3.5

Furthermore, the list can also be reduced by specifying a threshold so that only those items with similarity equal to or above the specified threshold will be considered for the recommendations. Another way of filtering the above list is to recommend *k* most similar items. One of the advantages of using top N recommendation based on *k*-NN approach is of displaying a relatively high number of the option without overwhelming the user, preselecting items how well they match the stated preferences of a user and generating relatively high decision accuracy[85].

5.3 Evaluation of Metrics

5.3.1 Recommender Systems' Tasks

It is important to understand the various tasks performed by a recommender system as it helps users in choosing the most appropriate metrics and the best algorithm for a given problem. There are primarily three classes of recommendation tasks namely 'optimizing the utility of an item', 'predicting the ratings of items', and 'recommending good items' among others. Among all these tasks the 'task of prediction' and 'recommending good items' are the most relevant to this research because we have used them task for the prediction of estimated ratings of data mining courses and selecting the most relevant courses from the top-k courses based on the ratings provided by learners over a set of courses. Hence, they are discussed briefly in the following section:

5.3.1.1 Prediction Task

Majority of recommender systems based on collaborative filtering approach recommend items based on the prediction value they generate for a user over an item. Prediction is based on the ratings provided by users over a set of items. The ratings are represented using 'user item rating matrix'. The goal of this task is to make prediction as close as possible to the true rating of item. For example, if the actual rating of an item 'X' is 4 on a scale of 1 through 5 scale where 1 stands for 'least liked item' and 5 represents 'most liked item' and a recommendation system predicts the value of the item say '3', then the prediction can be considered as good. On the other hand, if a recommender system predicts the rating as '1' then this value is far from the true rating of item so the prediction would not be considered good.

5.3.1.2 Recommending Good Items

This is one of the most common tasks of recommender system [26]. For example, in YouTube, when a user selects a particular movie then other movies of similar type are also presented to the user for consideration. Similarly, In Flipkart, when a user buys a book, other similar books are also suggested to the user for consideration. In addition, whether to recommend all the good items or some of the good items to a user requires us to consider factors such as time and resources available to a user. If a recommender system has a large number of good items to be displayed but the 'resources' or 'time' is a constraint, then it would be better to show only a subset of good items. Hence, it is likely that, some of the good items would be missed out from the recommendation list. Here, it is very important not to suggest any item that may be disliked by a user. We have used threshold measure in order to filter out irrelevant courses from the list of recommendations. For instance, a threshold value of 2.5 suggests that, all recommended courses with predicted ratings equal to or above this value would be part of the recommendation list and all those courses having prediction value below this would be removed from the list.

5.4 Dataset

The choice of dataset for evaluating a recommender algorithm plays a crucial role in the quality of recommendations. Hence, if the goal of a recommender system is to recommend a movie then the properties of the dataset should match with the movie domain [26]. In the following subsection, we discuss some of the essential properties of a dataset such as density of a dataset, user-item ratio, synthesized vs real dataset and diversity of a dataset that have been taken into account while selecting the best evaluation metric for a given recommender algorithm and a recommender task.

5.4.1 Properties of Dataset

One of the crucial factors while evaluating a recommendation algorithm is to choose a dataset carefully as it has major impact on the quality of recommendation. Hence, if the goal is to recommend a movie then the properties of the dataset should match with the movie domain [26]. Some of the properties of a dataset are discussed below:

- Density of a dataset means how many cells of a dataset are filled with ratings which increase with the increase in the number of ratings. If a recommender system is designed in such a way that it collects "explicit ratings" from its users then the "user item rating matrix" would be highly sparse. One of the main reasons for this scenario is that, a lot of effort is required from users to provide ratings. On the other hand, in case of "implicit ratings" a "user item rating matrix" is highly likely to have more ratings which mean such matrix would have high density as less effort is required from users [26]. The rating collection method that has been used in this research is 'explicit' in which each cluster of learners is asked to provide ratings over a set of courses on a scale of 1 through 5. However, many of the learners were unwilling to provide ratings which results in the 'sparse ratings item matrix'. 'This factor is especially more important because often a sparse matrix causes low accuracy of recommendations as it doesn't have rating data to find common items between users.

- Another aspect to consider is ratio between learners and courses. The relation between learner and course can also affect the performance of a recommender algorithm. Compute similarity between users and items might achieve unusually high performance. In one of the works reported, it is found that applying item based algorithm to a dataset with more users than items may lead to better results [87]. We have tried to strike this balance between learners and courses so that the results produced may not be unnecessarily better. For instance, 20 learners from each cluster (not active, average and active) provide ratings to 40 courses from each category of data mining courses.

5.4.2 Experimental Settings

In the process of evaluation of a recommender system, it is paramount to take decision regarding the type of experiment (online vs offline) to be performed. While online experiments provide better results than offline experiments in terms of quality of recommendations, offline experiments have their own importance. It is also important to understand that when a particular experiment should be performed. Hence, keeping this in mind, in the following subsection we discuss two major types of evaluation approaches namely, offline experiments and online experiments and also sheds light on when a particular type of experiment would be suitable and under what conditions in a given settings:

5.4.2.1 Offline Experiments

The objective of offline experiment is to select the most appropriate recommender approaches and filter out the irrelevant approaches, which reduces the list of candidate algorithms to be tested in an online experiment. In an offline experiments, no actual users are involved and pre existing dataset is used. This dataset is divided into test and training dataset [26][117][134]. The 'training data' is used for building a model for a specific recommendation task such as prediction and the 'test data' is used in order to evaluate the model. One of the major reasons for performing offline experiments is that they are quick, economical and easy to carry out. However, one of the major limitations of offline experiment is that, we can't evaluate those items for which we don't have their actual ratings by users. Hence, the issue of 'sparsity' limits the set of items that can be evaluated.

	Recommended	Not Recommended
Preferred	true-positive(tp)	false-negative(fn)
Not Preferred	false-positive(fp)	true-negative(tn)

Table 5.8 Confusion Matrix used in Offline Experiment

The results of a classifier can also be represented using the 'confusion matrix'[25]. The matrix is a two dimensional table with a row and column for each class. Each element of the matrix shows the number of test examples. The number at the diagonal of the matrix represents correctly classified instances and the number of elements outside diagonal indicates incorrectly classified instances. For example, true positive value in the above matrix indicates that, an item being preferred by a user has also been recommended. Moreover, another value at the diagonal shows that, an item not preferred by a user has also not been recommended. On the other hand, the value false positive (fp) suggests that, an item which is not preferred by a user has been recommended. Furthermore, the value false-negative indicates that, an item which is not preferred by a user has also not been recommended. The confusion matrix is just another way of representation of results of recommender system. The above confusion matrix can be explained as follows:

Case1: True Positive (It means a relevant item is classified as relevant)
Case2: True Negative (It means a non relevant product is classified as non relevant)
Case3: False Positive (It means Non relevant item is classified as relevant)
Case4: False Negative (It means a relevant item is classified as non relevant)

5.4.2.2 Online Experiments

Online experiments are conducted with real users and actual data which are collected from users who are interacting with the recommender system. They take great amount of user's effort in order to elicit ratings for making recommendations. Data is collected explicitly by getting feedback on a scale of 1-5 where '1' shows least like item and '5' indicates most liked. One of the disadvantages of online experiment is that, it is expensive, and takes more time to complete. However, evaluation of recommendation algorithm through online experiment is more trustworthy than offline experiment and user studies. This is due to the fact that, a set of candidate algorithms can be evaluated using real data coming from real users and a ranking in terms of superiority can be given to each of these algorithm which makes it easy to decide on the best algorithm. Due to this, a majority of real world system make use of online testing system [168]. One of the important consideration in such experiments is to collect data randomly which makes sure that a bias does not introduce in the system which leads to producing recommendation in favour of some items while other more useful items are neglected. Another point to consider is that, a real system may provide irrelevant recommendations to a user which might be seen as a discouragement to use this system further. Hence, keeping all these points in mind, it is always advisable to perform online testing in the last stage after all the offline testing has been completed.

5.5 Evaluation of Metrics

There is a large variety of evaluation metrics available in order to measure the performance of a recommender algorithm. For the convenience of researchers [26][228] the authors have classified them into three broad categories as given below,

5.5.1 Predictive Accuracy Metrics

These are most commonly used metrics by recommender systems. They evaluate how close are the estimated ratings to the actual ratings generated by a recommender system. Some of the widely used example of such metrics are mean absolute error (MAE), mean squared error (MSE), root mean square error (RMSE), and normalized mean absolute error (NMAE) among others.

In the following subsection, we provide an overview of the most popular predictive evaluation metrics that have been reported in the literature on recommender system. We also discuss the properties of each metrics and explain why it is most important for a given recommender task.

5.5.1.1 Root Mean Square Error (RMSE)

This metric is discussed in the context of the 'rating prediction task' as it is used mostly for that task. This task of a recommender system provides a set of items with their predicted ratings. The ratings are evaluated by measuring their accuracy. The task of 'rating prediction' is often performed in classification and regression in the machine learning and statistic literature [208]. Several metrics have been reported in the literature. One of the widely used metric is RMSE (root mean square error) which is used to score an algorithm. For example, If $P_{i,j}$ is the predicted rating for user i over item j, and $V_{i,j}$ is the actual rating and K={i,j} is the set of hidden user item ratings, then RMSE is defined by the following formula:

$$\sqrt{\frac{\sum_{i,j \in K} p_{i,j} - v_{i,j}^{2}}{n}} \qquad \text{.....(2)}$$

Where, n is the total number of items.

RMSE is used when one is dealing with the problem of 'regression' where the predictor variable is a real number, therefore in order to measure the quality of predictor variable from some algorithm 'A', one needs to find some sort of differences between them. It is generally computed by squaring of the error, taking the mean across all objects and finally taking square root. This will provide us a real score, that shows some confidence to how good or bad the given algorithm is performing. The lower the value of RMSE, the better the quality of model. The individual differences being calculated between predicted ratings and actual ratings are called residual and are aggregated into a single value to represent predictive power. One of the properties of RMSE is that, it tends to penalize large errors more severely than the others. For example, if an error of one point increases the sum of error by one, but an error of two point increases the sum by four [26]. RMSE is used widely for measuring the performance of an algorithm due to the following reasons:

a) It is easy to compute the solution.

b) It is symmetric and quadratic which makes it suitable to use for Gaussian noise.

c) Generally, minimizing RMSE provides an approximation for the conditional expected value of the next observation (to be predicted) given the explanatory variables (the past in time series).

d) It is primarily used when we have to show bigger deviations.

e) It is more useful when large errors are particularly undesirable.

RMSE is appropriate for 'prediction task' because it computes inaccuracies on all ratings either negative or positive. However, it is most suitable in those situations where we don't differentiate between errors.

5.5.1.2 Mean Average Error (MAE)

It is another standard statistical evaluation metric which is widely used for measuring the performance of an algorithm. One of the unique features of MAE is that it assigns the same weight to all the errors while RMSE gives higher weight to larger error as compared to smaller errors. It measures the average of a set of prediction without taking into account their direction. It is the average over the test instances of the absolute difference between actual value and the predicted value. It is important to note that the range of MAE varies from '0' to infinity, which means that, it reaches its best value at '0' or in other words we can say that it can have a minimum error of '0' and maximum error could go up to infinity depending upon the rating scale of the application. Furthermore, The value of RMSE is always greater than MAE. The lower the value of MAE the higher the accuracy of a model. If all errors are of the same magnitude then the value of MAE and RMSE is same in other words MAE=RMSE. It is important to note that, MAE is not suitable for some tasks such as finding a small number of objects that are likely to be appreciated by a given user. The computation of mean average error is done with the help of the following formula [26].

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | d_i | - | \hat{d}_i | \qquad .....(3)$$

Where d_i is the actual rating.

$d^{\wedge i}$ is the predicted rating.

and 'n' is the total amount of rating.

5.5.1.3 Normalized Mean Average Error (NMAE)

There are several variations of mean absolute error (MAE) such mean squared error (MSE), root mean square error(RMSE) and normalized MAE(NMAE). NMAE [209] refers to mean absolute error multiplied by a normalization factor α in order to normalize the value to the range of rating values. This normalization is performed in order to allow inter dataset comparisons. It is represented by the following equation:

$$NMAE = \alpha MAE = \frac{1}{r_{max} - r_{min}} MAE \qquad .....(4)$$

Where,

r_{max} = largest possible rating provided by a user

r_{min} = smallest possible rating provided by a user.

In [26] the authors have found that, mean absolute error metric is less appropriate when the granularity of true preferences is small. Error in the predicted rating does not cause any problem as long as an interesting item is not classified as a not interesting item which can lead to user dissatisfaction.

5.5.2 Classification Accuracy Metrics

The second class of evaluation metrics is 'classification accuracy metrics'. These metrics are used when a recommender system needs to make granular decisions about user/item pairs. For example, recommend/do not recommends and Yes/No. Some of the popular examples of these metrics are precision, recall, F-measure, and ROC (receiver operator characteristics) curves among others. The performance of an algorithm can be represented using precision/recall curve. There is a deep connection exist between precision/recall and ROC. Precision refers to set of relevant retrieved documents from a set of retrieved documents. The precision and recall were defined in [210]. Usually, a tradeoff can be seen between precision/recall. This means when we get higher precision, a decline in the value of recall can also be seen and vice versa.

5.5.2.1 Precision

It is one of the widely used measures for measuring the accuracy of prediction in a recommender system. Precision [210]is used when the task of a recommender is to recommend certain item or when the number of recommendations to be shown to a user is predetermined or when we need to find to find precision at N which means only top N results need to be examined to determine if they are relevant or not.

$$\text{precision} = \frac{\left|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}\right|}{\left|\{\text{retrieved docuements}\}\right|} \qquad \text{.....(5)}$$

5.5.2.2 Recall

Recall refers to the ratio of retrieved relevant documents divided by total relevant documents in a database. Usually, the recall doesn't provide a result which can be evaluated in absolute terms. Hence, a recall should be used when the goal is to evaluate one algorithm with respect to another. For example, if an algorithm has a recall value of 0.5 then this doesn't make much sense or it can't be interpretable. On the other hand, if another algorithm B has a recall value of 0.6 then it can be concluded that the performance of algorithm B is better than the performance of algorithm A. The recall is computed using the following formula

$$\text{recall} = \frac{\left|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}\right|}{\left|\{\text{relevant docuements}\}\right|} \qquad \text{.....(6)}$$

5.5.2.3 F1 Measure

In binary classification, F1 score is used to measure a test's accuracy. It takes into account both the precision and recall of a test in order to compute a single score. The best value of F1 is 1 which is obtained when both the precision and recall are equal to 1. The worst value of F1 is 0. The F score is used in the field of machine learning, information retrieval, document classification and query classification [211]. Using the F1 measure, we can get the more realistic view of the performance of a recommender algorithm. F1 measure is the harmonic mean of precision and recall [212] which is computed using the formula as shown below:

$$F1 = 2\left(\frac{precision.recall}{precision + recall}\right) \qquad .....(7)$$

5.5.3 Rank Accuracy Metrics

This class of metrics takes into account the order of items in the list generated by a recommender system. The accuracy of a recommender system is based on the order of items. For example, if there are three items say, X, Y, and Z in the list produced by a recommender system. Let us consider, a user prefers these items in the order of Y, X, and Z. That means, the user prefers item Y over X. Rank accuracy metrics take this into account for ranking the order of items and penalize the recommender system for not producing the list of items as preferred by a user. There are several rank accuracy metrics. Two of the commonly used metrics are Spearman's correlation coefficient and Kendall's tau correlation [26]. The Spearman's correlation coefficient is shown in the following equation:

5.5.3.1 Spearman's Correlation Coefficient

Spearman's correlation is the special case of Pearson product moment correlation coefficient. However, the primary difference is that the data is converted to ranking before computing the coefficient. The following equation shows Spearman's correlation coefficient:

$$\rho = \frac{\sum (x - \overline{x})(y - \overline{y})}{n.stdev(x).stdev(y)} \qquad .....(8)$$

5.5.3.2 Kendall's tau Correlation Coefficient

It is one of the widely used ranking accuracy metrics. It also uses a number of concordant and discordant pairs-pairs. A pair of tuples (x_1,y_1) and (x_2,y_2) is concordant when $sgn(x_2-x_1)=sgn(y_2-y_1)$ and discordant when $sgn(x_2-x_1)= -sgn(y_2-y_1)$, where x_i and y_i are the ranks for the item a_i as ranked by the user and predicted by the recommender system.

$$\tau = \frac{C - D}{\sqrt{(C + D + TR)(C + D + TP)}} \qquad .....(9)$$

Where:

C= Number of concordant pair

D= Number of discordant pair

TR and TP= Number of pair of items that have the same ranking in the true ordering and predicted ordering respectively.

The 'sgn' function is defined as below:

$$sgn(x) = \begin{cases} -1 & x<0 \\ 0 & x=0 \\ 1 & x>0 \end{cases} \qquad .....(10)$$

5.6 Recommendation Methods

Although there are several user based similarity measures such as Euclidean distance, log likelihood, Pearson correlation, Tanimoto coefficient, uncentered cosine and Spearman correlation[213] being used by the existing user based recommender systems, we chose two of the most widely used collaborative recommendation methods namely 'cosine similarity' and 'Pearson correlation' in order to find which one produces the better results with a given 'evaluation metrics' and a given 'recommendation task'. In the following subsection, we provide a brief discussion of these collaborative approaches. We also compare them by evaluating their performance on a given recommender task such as prediction and recommendation and measuring the accuracy of each task in terms of (RMSE).

5.6.1 Cosine Similarity

This is one of the widely used similarity measures of collaborative filtering [155]. It computes the angle between two rating vectors. The formula of cosine similarity is given below:

$$w(a,i) = \sum_{j \in I_{a,i}} \frac{v_{a,j}}{\sqrt{\sum k \in I_a v^2_{a,k}}} \frac{v_{i,j}}{\sqrt{\sum k \in I_i v^2_{i,k}}} \quad \ldots\ldots\ldots\ldots\ldots\ldots (11)$$

It is important to note that, we consider only positive ratings while computing similarity between two rating angle and negative ratings are not taken into account. In the above equation *Ii* refers to the set of items for which user has provided positive ratings and *Ia, i* is the set of items for which both users have rated positively. Furthermore, the predicted value for a user is computed by using the following formula:

$$p_{a,j=K}\sum_{i=1}^{n} w(a,i)v_{i,j} \qquad \text{..... (12)}$$

However, when we are using binary dataset such as the usage dataset, the vector similarity method becomes,

$$w(a,i) = \frac{|I_{a,i}|}{\sqrt{|I_a|}.\sqrt{|I_i|}} \qquad \text{..... (13)}$$

Where I_a is the set of items that a used and $I_{a,i}$ is the set of items that both *a* and *i* used.

5.6.2 Pearson Correlation

Typically a prediction task requires input from users which is represented in the above equation by $v_{i,j}$ where user i has provided a rating to item j. Given such a dataset, we can compute the similarity of each user in a dataset with the active user which is represented by w(a,i). Moreover, $p_{a,j}$ represents predicted rating of 'a' over 'j' and can be computed by [13].

$$p_{a,j} = \bar{v}_a + K\sum_{i=1}^{n} w(a,i)(v_{i,j} - \bar{v}_i) \qquad \text{..... (14)}$$

The value of w (a,i) can be computed using the following formula:

$$w(a,i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - v_i)^2}} \qquad \text{..... (15)}$$

Pearson correlation is specially designed for 'prediction task' as it computes only predicted score for each item of interest. However, many works in the literature have used this method for recommendation task as well which is performed by predicting the score for all items and arrange them in decreasing order of their ratings.

5.7 Experimental Results

In this section, we carry out an experimental evaluation of two of the widely used user-based recommender algorithms namely 'Pearson' and 'cosine' in order to evaluate their performance over a given recommender tasks such as 'prediction' and 'recommendation'. These algorithms are applied to the learners' data that has been extracted from Moodle

server. The dataset contains ratings provided by 60 users over 120 items. We have taken three datasets namely 'dataset1', 'dataset2' and 'dataset3' of size 638, 400 and 400 respectively. Firstly, the ability of each recommender approach is evaluated using one of the most commonly used evaluation metrics i.e RMSE (root mean square error) in order to determine the accuracy of prediction. After applying the algorithms and measuring their accuracy over the tasks the results are shown in Table 5.9. It can be seen that the value of RMSE is lower when Pearson method is used for prediction task. On the other hand, cosine method yields higher RMSE values when the same collaborative approach is applied to the same dataset for prediction. Based on the experimental results, it can be concluded that RMSE yields better results when the recommended approach is Pearson and the task to be performed is 'prediction'. Moreover, when we applied the cosine recommender approach to the same datasets, we obtained higher RMSE values which suggest that the quality of prediction is not as good as in case of Pearson method. The corresponding graph of Table 5.9 is shown in Figure 5.3.

Table 5.9 RMSE Score for Pearson and Cosine Methods

Type of Algorithm	Dataset1	Dataset2	Dataset3
Pearson	0.84	1.25	1.45
Cosine	1.60	2.50	2.60

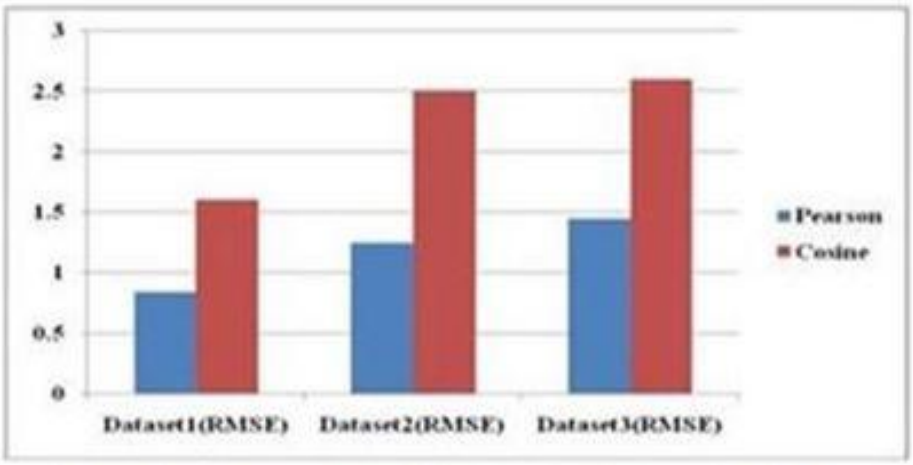


Figure 5.3 Comparison of Pearson and Cosine Methods in terms of Prediction Task

Another important task of a recommender system is recommendations of items or services. We are also interested in evaluating the performance of the two recommender approaches to the task of recommendation and measuring the accuracy of the obtained recommendations using other well-known evaluation metrics such as precision and recall curve. The curve measures the proportion of the recommended items that are actually preferred by the user. Usually, a higher precision is achieved at the cost of lower recall value and vice versa. This means that precision and recall are inversely proportional to each other. Hence, a fine balance needs to be established between these two metrics. This tradeoffs between the two can be observed using the precision-recall curve and an appropriate balance between the two is obtained. Thus, to achieve this, precision-recall curves come in handy. We computed precision and recall values for the different set of recommendations which are shown in the Figure 5.4, where, 'k' represents the number of recommendations. Then, we average precision and recall at each recommendation. Figure 5.4 shows that, if the number of recommendations is set to five then the performance of Pearson correlation yields better RMSE score which is close to 0.4 in our case. Similarly, the approach has performed well for other values of 'k' such as k=10 and k=20. These experimental results suggest that, if the task of a recommender system is recommendation and the two recommender approaches under consideration are Pearson and cosine similarity then it can be concluded that Person collaborative algorithm gives lower RMSE.

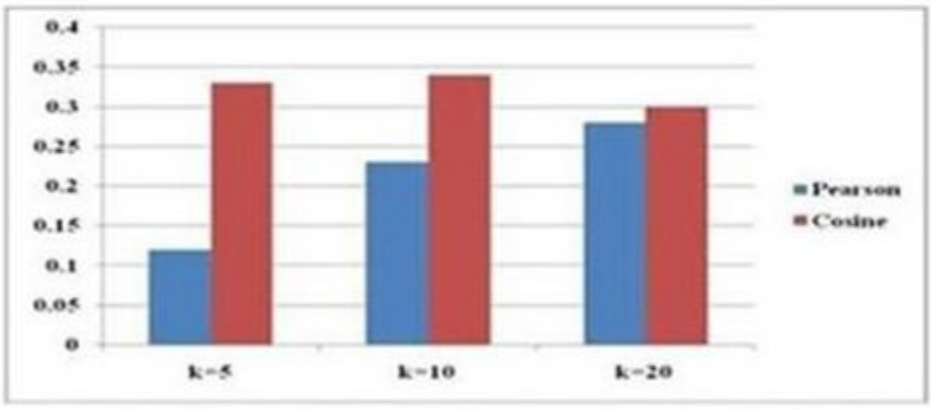


Figure 5.4 Comparisons of Pearson and Cosine Methods over Recommendation Task

This also means that the accuracy of recommendation is higher when Pearson approach is used. It is also interesting to see that, as we increase the value of 'k' the corresponding value of RMSE is decreased. For instance, when k=5(five items are recommended) the value of RMSE is close to 0.4 and when the value of k is increased from 5 to 20 the value of RMSE is increased to 0.3.

We have compared the two of the most widely used collaborative filtering approaches namely 'Pearson correlation' and 'cosine similarity' over two of the most common usage scenario of a recommender system and measured their accuracy using some of the well-known evaluation metrics such as RMSE, precision, and recall.

In addition to these two tasks, the ranking of items is another most common task of the recommender system. In a ranking task, the recommender tries to assign an order to the items often with the objective of creating a top-k list of items. One of the advantages of rank accuracy metrics is that, even if, a recommender system estimates the rating of an item to be lower than the actual ratings provided by the user, it does not matter as long as it presents the correct ranking of items. The concepts of total ordering and partial ordering are important to understand in rank accuracy metrics. The two of the widely used ranking metrics are *Kendall's tau* and *Spearman's correlation coefficient* which have already been discussed in the section rank accuracy metrics. Here, we apply these ranking methods to the ranking dataset provided by users and recommender system which is shown in Table 5.11. In order to find the value of tau(τ) the values of C, D, TR, and TP must be determined first.

The value of *Kendall's coefficient* varies between -1 to +1. All those values closer to +1 indicate a strong correlation between the two variables and those values which are closer to -1 suggest a weak correlation. The value of correlation coefficient is determined by the equation.9 and it is computed as 0.43. This value suggests that there is a weak correlation between the two lists of ranking items. Another widely used method for computing the accuracy of ranking is *Spearman's correlation coefficient* which has been computed using equation.8 and is shown in Table 5.12.

Table 5.10 Correlation Computed using Kendall's Tau Approach

Item	Spearman
T	(1,1)
U	(3,2)
V	(5,5)
W	(6,7)
X	(4,3)
Y	(7,4)
Z	(2,6)

Table 5.11 Ranking List Provided by User and Recommender System

Ranking	User	Recommender System
1	T	T
2	Z	U
3	U	X
4	X	Y
5	V	V
6	W	Z
7	Y	W

Table 5.12 Computation of Spearman's Coefficient Correlation

X Values	Y Values	X_{ra}	X_{ra}-M_x	Y_{ra}	Y_{ra}-M_y	Sumdiffs
1	1	1.00	-3.00	1.00	-3.00	9.00
3	2	3.00	-1.00	2.00	-2.00	2.00
5	5	5.00	1.00	5.00	1.00	1.00
6	7	6.00	2.00	7.00	3.00	6.00
7	3	7.00	3.00	3.00	-1.00	-3.00
4	4	4.00	0.00	4.00	0.00	0.00
2	6	2.00	-2.00	6.00	2.00	-4.00

Where,

X_{ra}= ranks of X values

Y_{ra}= ranks of Y values

X_{ra}-M_x= X_{rank}-Mean of X ranks

Y_{ra}-M_y= Y_{rank}-Mean of Y ranks

Sumdiffs= (X_{ra}-M_x) * (Y_{ra}-M_y)

Result Details

X Ranks
Mean=4
Standard Dev=2.16

Y Ranks
Mean=4
Standard Dev=2.16

Combined covariance=11/6=1.83
R=1.83/(2.16*2.16)=0.393

As we can see from the above computation, that the value of Spearman's correlation coefficient is 0.39 which is far less than 1, hence it can be concluded that the association between the two variables would not be considered statistically significant. The obtained values by *Spearman's correlation coefficient* and *Kendall's coefficient* are 0.43 and 0.39

respectively. This suggests that, with a given choice of these two rank accuracy metrics, *Kendall's coefficient performs* slightly better under the same dataset.

5.8 Evaluation and Discussion

After making the prediction of various data mining courses to learners, it is also important to evaluate the accuracy and performance of the recommender framework in order to measure the quality of recommended courses. Therefore, we used most common statistical measures such as RMSE, Precision, Recall and F1 score [22]. Table 5.13 and its corresponding graph in Figure 5.5 show different values of these metrics for different data mining courses being recommended to learners belonging to one of the three clusters namely 'not active', 'average' and 'active'. Furthermore, the precision that indicates the accuracy of recommendations is also computed for each cluster. Precision refers to the number of documents retrieved that are relevant divided by a total number of documents that are retrieved. The highest precision value obtained for "average cluster" suggests that out of all retrieved courses from database 90% of the recommended courses are relevant to a learner. Similarly, we obtained the highest recall value of 72% for learners belonging to "average cluster" which indicates that the recommender system is able to retrieve 72% of relevant courses from all the relevant courses in the course database. In addition to these two metrics, more balanced view of performance can be obtained by using the F1 metric which provides a single measure based on the combined value of precision and recall. The value of F1 is measured for three classes of learners with the highest value of 79% measured for the average cluster.

We also computed RMSE (Root mean square error) for data mining courses that have been recommended to learners falling in one of the three clusters. It is used to measure the differences between predicted value provided by model and actual observed value. The aim is to reduce the value of RMSE in order to improve the accuracy of recommended courses. Among the three values of RMSE that we obtained, the lowest is 0.39 for the 'active cluster'.

Table 5.13 Evaluation of Recommendations

Type of Cluster	RMSE	Precision	Recall	F1 Score
Not Active	0.73	0.60	0.31	0.40
Average	0.60	0.90	0.72	0.79
Active	0.39	0.70	0.27	0.39

We also compared our results with the results of other similar approaches. In one such approach [214] the author has proposed three recommendation methods namely 'user-based', 'item-based' and 'model-based' and evaluated them experimentally using RMSE and MAE (Mean absolute error). The author obtained an RMSE of 1.88 as compared to 0.39 measured in our approach for courses offered to 'active learners'.

In [215] the author has proposed three algorithms for recommendations of courses and evaluated those algorithms using precision, recall, and F1 metrics. Among the three algorithms the best one produced a precision of 0.81, a recall value of 0.55 and an F1 score of

Table 5.14 Comparison of M1 approach with Model based Approach

Type of Approach	RMSE
Approach based on k-means and K-NN (M1)	0.6
Model based approach	1.88

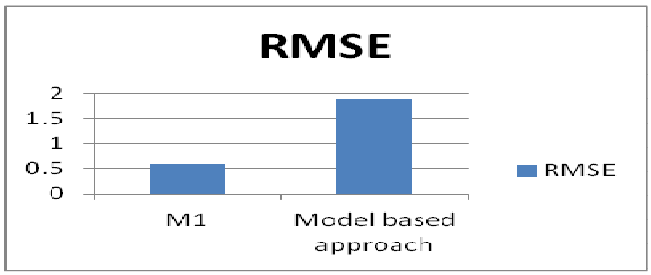


Figure 5.5 Comparison of approaches with respect to RMSE

0.66 as compared to a precision of 0.90, a recall of 0.72 and an F1 score of 0.79 measured by our approach for courses recommended to learners belonging to 'average clusters'.

In [216] the authors have proposed a course recommender system for online enrolment of courses. They used a variation of 'item-based collaborative filtering' algorithm to recommend elective module to students based on the core module that they have selected. The evaluation of collaborative filtering algorithm is performed using 10-fold cross-validation method. They used 'recall' and 'coverage' as the evaluation metrics for top-10 recommendation list and the recall measured by the algorithm is 66% as compared to 72% by our approach for courses being recommended to learners belonging to 'average cluster'.

Furthermore, the recall value obtained for courses offered to other clusters is relatively low due to variation in the sparsity level of 'rating matrix". However, the obtained recall value could further be improved by increasing the number at which precision and recall are being calculated. However, the primary goal of the proposed recommendation

framework is to enhance the accuracy of recommendations with a reasonable recall value by enriching the ‘user-item rating matrix'.

5.9 Summary

The proposed recommendation framework recommends different categories of data mining courses namely ‘courses for beginner’, ‘courses for intermediate learner’ and ‘courses for advanced learners’ to learners belonging to one of the three clusters namely ‘not active’, ‘average’ and ‘active’ based on their profile. The recommendations generated by the algorithm are evaluated using well known metrics. However, we also carried out an experimental analysis of different evaluation metrics is in order to determine the most appropriate evaluation metrics based on the type of recommender algorithm and the type of recommender task. The results of this analysis help us to select a suitable metric which matches a given recommender algorithm and a recommender task and it is further used for the evaluation of recommendation of courses. Furthermore, the results of evaluation of recommendations are compared with other similar works and it is found that, the proposed recommendation approach is able to achieve improved accuracy.

In the next chapter, the semantic knowledge about learners is taken into account in the recommendation process in order to enhance the accuracy of recommendations. In order to achieve this, RDF is being used for enriching ‘user item rating matrix’.

Chapter 6

Improving Recommendations Accuracy by Enriching 'User item rating Matrix'

In the previous chapter, we recommended various data mining course to different learners based on their profile. The accuracy of recommendations was measured using well known evaluation metrics. However, the collaborative filtering approach that we have used for recommendation usually suffers from the 'sparsity' [156] issue in which the rating matrix does not have sufficient ratings to make good quality of recommendations to learners. Keeping this fact in mind, in this chapter, we aim to enrich the 'user item rating matrix' by extracting additional preferences of learners over unrated data mining courses through semantic web techniques such as Apache Jena and resource description framework for improving the accuracy of recommendations.

6.1 Problem Statement

Before we present the recommender framework, it is important to understand the issue of 'sparsity' that the framework is going to deal with. We discuss this issue in the context of e-learning. Let us consider the recommendation problem in e-learning domain where there are a number of courses and a large number of learners enrolled in these courses. Furthermore, learners can express their preferences on different data mining courses in the form of ratings on a scale of 1 to 5. Here, '0' means that, the user has not studied the particular course, '1' refers to least liked courses and '5' implies most liked course. In other words, we can say that, the more a learner preferred a course the higher the rating he/she gives. All the ratings are stored in a data structure called 'user item rating matrix' where each row represents a learner and a column the item. In addition, we can see that, Table 6.1 shows many empty cells. This means, a subset of learners have not rated some courses. This situation leads to the problem of 'sparisty'. The sparsity occurs primarily due to the fact that either the learner has not experienced those courses or they are not getting any sort of incentive from the organization who is seeking their ratings over set of courses. It is

important to note that, Table 6.1 only shows a subset of ratings from the entire 'user item rating matrix'. The actual rating matrix contains 500 learners.

Table 6.1 shows a view of 'sparse user item rating matrix' where we represent all the learners by 'L' and all the courses using 'C'. Each learner shown in Table 6.1 belongs to 'L' and each course belongs to 'C'. Different learners which are a subset of 'L' are represented by L_1,L_2….L_n and the different courses which are subset of 'C' are represented by C_1,C_2…..C_n.

Table 6.1 A View of Sparse 'User Item Rating Matrix'

Courses → Learners ↓	C1	C2	C3
L1	3	?	4
L2	?	3	?
L3	3	?	?
L4	?	5	4
L5	3	?	?
L6	?	4	?
L7	4	?	4

In this context, the objective of this work is to estimate the preferences of learners over unrated courses through moodle server, RDF factbase and Jena rules. Hence, how to improve the accuracy of recommendations by utilizing the above context in e-learning is the main task of this work

6.2 The Proposed Enhanced Recommendation Framework

We now introduce the proposed approach to improve the performance and accuracy of recommender approach. Here it is important to note that, the framework makes use of one of the widely used semantic technology named resource description framework. Firstly, we describe the different components of the framework such as moodle information model, representing classes in moodle information model into RDF representation, RDF factbase and formation of Jena rules as shown in Figure 6.1. It is also important to note that, we have not shown the different categories of data mining courses, all their ratings provided by different type of learners which is already discussed in[224].

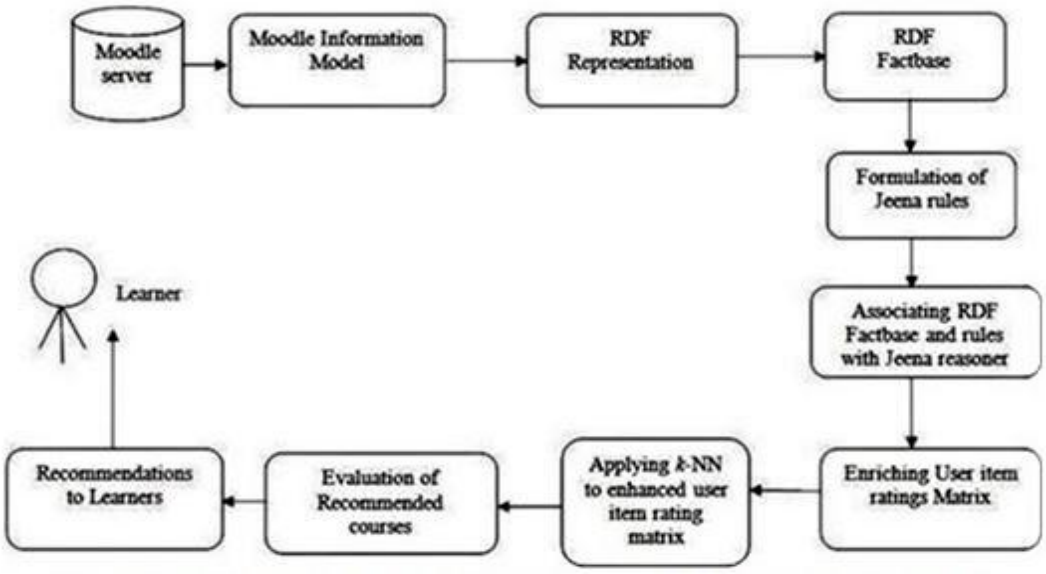


Figure 6.1 Framework for Recommendation of Courses using Enriched User Item Rating Matrix

6.2.1 Moodle Information Model

Firstly, we introduce 'moodle information model' which is derived from 'moodle database schema' which is an open access structure [231]. The schema is quite large containing 200 tables. However, we consider only those tables which are relevant to our work. The schema is used to represent the information about learners' activities such as quiz, assignment and forum among others which are stored in the log file of moodle server [132].

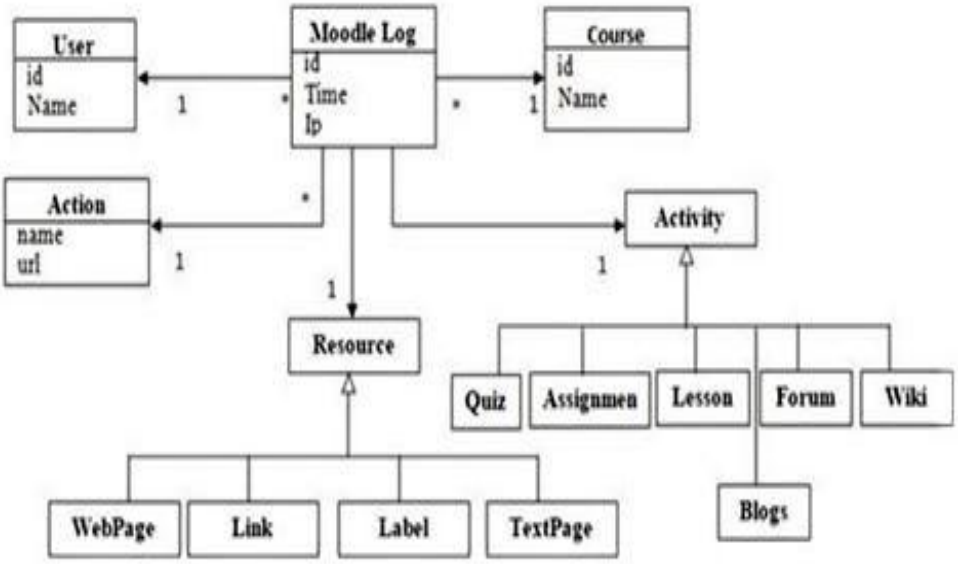


Figure 6.2 A View of Moodle Information Model

The model in Figure 6.2 shows a number of super classes and subclasses and their relationship in the form of generalization. We have used UML (unified modelling language) for modelling notation [217]. The cardinality of the relationship between the classes is also shown. For instance, we can see 1:M cardinality between the two classes named 'course' and 'moodle log' which indicates that, many users are enrolled in a particular course. Moodle log is the core class which stores three users' fields such as 'id','time' and 'ip'. The various activities performed by a user are shown under 'activity' class. Learners also use resources such as text page, link and web page among others in order to carry out their task. The main purpose of moodle information model is not only to graphically represent the log file of a moodle server but it facilitates the process of transformation from UML class diagram to RDF representation.

6.2.2 Resource Description Framework (RDF)

Resource description framework is one of the widely used techniques of semantic web for describing resources on the web [23]. According to W3C recommendations, RDF is a foundation for processing metadata. It provides interoperability between applications that exchange machine understandable information on the web.

We use RDF to represent the various UML classes as shown in Figure 6.2. The reason for this conversion is to make it possible for semantic tools such as 'Apache Jena' to work with RDF dataset. In addition the following tools require data in a form which is structured and meaningful. Apache Jena is one of the commonly used semantic web tools for developing semantic web applications. Although there are several 'semantic web tools such as 'Notation3' [218], 'Semantic web rule language (SWRL)'[219], 'Racer'[220], 'JenaRules'[221] and 'Manarax'[222] among others, we chose particularly JenaRules as it is specially designed for semantic web applications.

6.2.3 Conversion from UML classes to RDF(S)

The conversion process from UML classes in moodle information model to its corresponding RDF is straightforward. Every class in Figure 6.2 corresponds to the RDF schema and every association corresponds to the RDF schema property. There exist several literatures on the rules of converting a given UML class diagram into its RDF(S)[223].We have converted

some of the UML classes in Figure 6.2 namely 'MoodleLog', 'course' and 'user' to their corresponding RDF(S).

*1) Converting **'MoodleLog'** to its corresponding RDF*:

Moodle log is the core UML class where each and every activity of a learner is recorded. Some of these entries are the 'time of logged' into the system, the 'ip' address of the machine from where the user has logged in and the 'id' of the user.

```
<rdfs: Class rdf:ID = "MoodleLog"/>
    <rdf: Property rdf: ID = "id">
        <rdfs:domain rdf:resource = "#MoodleLog"/>
        <rdfs:range rdf:resource = "&xsd:integer"/>
    </rdf:Property>
    <rdf:Property rdf:ID = "ip">
        <rdfs: domain rdf: resource = "MoodleLog"/>
        <rdfs:range rdf:resource="&xsd:integer"/>
  </rdf:Property>
  <rdf:Property rdf:ID = "Time">
      <rdfs: domain rdf: resource = "MoodleLog"/>
     <rdfs:range rdf:resource="&xsd:Time"/>
  </rdf:Property>
```

*2) Converting **'Course'** to its corresponding RDF:*

Another UML class 'course' with its properties 'course_id' and 'name' are represented by the 'id' and 'name' are written in resource description format as below:

```
<rdfs: Class rdf:ID = "course"/>
    <rdf: Property rdf: ID = "id">
        <rdfs: domain rdf:resource = "#course"/>
        <rdfs:range rdf:resource = "&xsd:integer"/>
    </rdf:Property>
    <rdf:Property rdf:ID = "name">
        <rdfs: domain rdf: resource = "course"/>
        <rdfs:range rdf:resource="&xsd:string"/>
  </rdf:Property>
```

*3) Converting UML class **'user'** to its corresponding RDF*:

The class 'user' which is described by two properties namely 'id' and 'name' has also been written in RDF format. The first field 'id' is used to uniquely determine a user on the moodle platform. Another field 'name' stores the learners' name in its log file.

```
<rdfs: Class rdf:ID = "user"/>
    <rdf: Property rdf: ID = "id">
        <rdfs: domain rdf:resource = "#user"/>
        <rdfs:range rdf:resource = "&xsd:integer"/>
    </rdf:Property>
    <rdf:Property rdf:ID = "name">
        <rdfs: domain rdf: resource = "user"/>
        <rdfs:range rdf:resource="&xsd:string"/>
    </rdf:Property>
```

4) Converting Generalization relationship 'Activity' to its corresponding RDF:

As it can be seen in Figure 6.2 that, there are several super and subclasses and they are linked by the generalization relationship. For instance, activity is a super class and all its subclasses are quiz, assignment, lesson and forum among others. This relationship is transformed into RDF as shown below:

```
<rdfs: Class rdf: ID = "Activity"/>
    <rdfs: Class rdf: ID="Lesson">
        <rdfs:subClassOf rdf:resource="#Activity"/>
    </rdfs:Class>
<rdfs: Class rdf: ID = "Activity"/>
    <rdfs: Class rdf: ID="Blogs">
        <rdfs:subClassOf rdf:resource="#Activity"/>
    </rdfs:Class>
<rdfs: Class rdf: ID = "Activity"/>
    <rdfs: Class rdf: ID="Forums">
        <rdfs:subClassOf rdf:resource="#Activity"/>
    </rdfs:Class>
<rdfs: Class rdf: ID = "Activity"/>
    <rdfs: Class rdf: ID="Wiki">
        <rdfs:subClassOf rdf:resource="#Activity"/>
</rdfs:Class>
```

A learner performs those activities which are represented through subclasses in the moodle information model. We have also represented two other activities such as quiz and assignment as shown below:

```
<rdfs: Class rdf: ID = "Activity"/>
    <rdfs: Class rdf: ID="Quiz">
        <rdfs:subClassOf rdf:resource="#Activity"/>
   </rdfs:Class>
<rdfs: Class rdf: ID = "Activity"/>
    <rdfs: Class rdf: ID="Assignment">
       <rdfs:subClassOf rdf:resource="#Activity"/>
</rdfs:Class>
```

6.2.4 RDF Factbase

In this subsection, we aim to convert the RDF representation that we obtained in the previous subsection into its RDF triples which is made up of three elements namely subject, object and predicate. The Jena rules that we have defined in the section- will be executed on the top of this factbase. We have created a database of RDF triples based on the RDF(S) obtained in the previous step. We have also shown some of the RDF triples as shown below:

RDF triple1

```
<moodle:Learner>
  <moodle: id>L1</moodle:id>
     <userdef: clicks_course   rdf:datatype="xs:boolean">rdf: resource =  "Course: Data
       mining for beginner"/>true
     <userdef: reads_forum    rdf:datatype="xs:boolean">true
    <userdef:reads_webpage  rdf:datatype="xs:boolean">true
     <userdef:writes_forum  rdf:datatype="xs:boolean">true
     <userdef:visits_link   rdf:datatype="xs:boolean">true
</moodle:Learner>
```

Here, it is interesting to note that an RDF triple consists of three elements such as subject, predicate and object. These three elements are also known as subject, properties and values. Through these three elements, we can describe any resource on the web.

Moreover, each learner in the RDF Factbase is represented by a triple. For example, we have represented L1 by triple1 as shown above.

RDF triple2

```
<moodle: Learner>
    <moodle: id>L3</moodle:id>
        <userdef: clicks_course rdf:datatype="xs:boolean">rdf: resource="Course:
              Basics Concepts of Data Mining"/ >true
        <userdef:reads_forum  rdf:datatype ="xs:boolean">true
        <userdef:reads_webpage  rdf:datatype ="xs:boolean">true
        <userdef:writes_forum  rdf:datatype ="xs:boolean">true
        <userdef:visits_link  rdf:datatype="xs:boolean">false
       </moodle:Learner>
```

In addition, another learner which is denoted by L3 in the moodle database, is also represented by RDF by triple2. This learner has performed all the activities except visiting link which can be seen in RDF triple2.

RDF triple3

```
<moodle: Learner>
     <moodle: id>L4</moodle:id>
        <userdef: clicks_course rdf:datatype="xs:boolean">rdf: resource="Introduction
            to Data Mining"/ >true
        <userdef:reads_forum  rdf:datatype="xs:boolean">true
        <userdef:reads_webpage  rdf:datatype="xs:boolean">false
        <userdef:writes_forum  rdf:datatype="xs:boolean">false
        <userdef:visits_link  rdf:datatype="xs:boolean">false
</moodle:Learner>
```

Furthermore, in the above triple which is named as RDF triple3, it can be seen that, the learner L4 has only clicked on the course named 'introduction to data mining'.

6.2.5 Formation of Jena Rules

In this subsection, we define certain rules based on the learners' activity which help us to infer the preferences of learners for the unrated courses. The ratings are on the scale of 1 to 5.

This means if a learner has performed all the required tasks then a rating of '5' would be assigned by the learner to the corresponding course.

We have given a single rating to each task performed by a learner. Hence, if a learner has done a single task then a rating of '1' would be provided by the learner to the corresponding course.

Below we present the rules which are implemented in 'Jena inference engine'. The RDF database that we created is made up of these triples. Here, it is important to note that, each triple represents a learner.

1) If a user has *'clicked on a particular course',* then it suggests that the user is interested in the course. Therefore, a rating of **'1'** would be assigned to the corresponding course by the learner.

2) If a user follows the 1st rule and *'reads a forum' pertaining to related concepts* then a rating of **'2'** would be assigned to the corresponding course by the learner.

3) If a learner satisfies the 2nd rule and also spends at least 30 seconds (threshold) reading a web page then a rating of 3 would be assigned to the corresponding course by the learner.

4) If a user follows the 3rd rule and also posts questions or reply answers related to the course he/she is interested then a rating of '4' would be assigned to the corresponding course by the learner.

5) If a user fulfils all the rules specified in the 4th rule and also *visits the relevant link* then a rating of **'5'** would be assigned to the corresponding course by the learner.

If any RDF triples in RDF factbase satisfies any of the above rules then it suggests that, the learner is interested in a particular course. There are two parts of a rule namely *condition* and *conclusion.* The condition part is shown on the right side of the rule and the conclusion part is depicted on the left part of the rule. The two parts are separated by an arrow as can be seen in the first rule written using RDF. If the condition part is satisfied then only the conclusion part will be executed by Jena engine. In addition, the degree of interestingness in a course can be determined based on which of the above rules are satisfied by the learner. The rules are also written in RDF format which is implemented in Apache Jena. The above rules will be specified using apache 'Jena inference engine' as shown below:

```
@prefix moodle: http://www.moodle.org/#
@prefix userdef: http://www.moodle.org/annotations#
@prefix lecture: http://www.moodle.org/lectures#
@prefix xs: http://www.w3.org/2001/XMLSchema

//Rule1
course_rating: (?learner, userdef: rating '1'^^xs:integer)
<-
    (?learner rdf:type moodle: learner)
        (?learner userdef:clicked_course ?clickcourse)
        (?clickcourse, "true" ^^, xs: boolean)
```

For instance, the first triple satisfies all the conditions as specified in the rule5 hence, a rating of '5' would be provided by the learner 'L1' to the course named 'data mining for beginner' which is represented by 'C2' in the course database.

```
//Rule2
course_rating: (?learner, userdef: rating '2'^^xs:integer)
<-
    (?learner rdf:type moodle: learner)
      (?learner userdef:clicked_course ?clickcourse)
        (?clickcourse, "true" ^^, xs: boolean)
     (?learner userdef:reads_forum ?readforum)
     (?readforum, "true" ^^, xs: boolean)
//Rule3
course_rating: (?learner, userdef: rating '3'^^xs:integer)
<-
    (?learner rdf:type moodle: learner)
      (?learner userdef:clicked_course ?clickcourse)
        (?clickcourse, "true" ^^, xs: boolean)
      (?learner userdef:reads_forum ?readforum)
        (?readforum, "true" ^^, xs: boolean)
      (?learner userdef:reads_webpage ?readwebpage)
        (?readwebpage, "true" ^^, xs: boolean)
```

On the other hand, the learner 'L6' would assign a rating of '4' to the course 'introduction to data mining' as the first four conditions are satisfied by the second triple. Similarly, we can determine the ratings of the remaining unrated 'data mining courses' in order to obtain an enriched 'user item rating matrix' which is shown in Table 6.2. The *RDF factbase* and the *Jena rules* have been implemented in *Apache Jena Fuiski*. The 'Jena inference engine' is used in order to get additional information about users' preferences which the users did not provide explicitly in 'user item rating matrix'.

```
@prefix moodle: http://www.moodle.org/#
@prefix userdef: http://www.moodle.org/annotations#
@prefix lecture: http://www.moodle.org/lectures#
@prefix xs: http://www.w3.org/2001/XMLSchema
//Rule4
course_rating: (?learner, userdef: rating '4'^^xs:integer)
<-
(?learner rdf:type moodle: learner)
(?learner userdef:clicked_course ?clickcourse)
(?clickcourse, "true" ^^, xs: boolean)
(?learner userdef:reads_forum ?readforum)
(?readforum, "true" ^^, xs: boolean)
(?learner userdef:reads_webpage ?readwebpage)
(?readwebpage, "true" ^^, xs: boolean)
(?learner userdef:write_forum ?writesforum)
(?writesforum, "true" ^^, xs: boolean)
```

This additional information could be such as the amount of time spent on a resource, the type of courses a user has clicked on, and number of forums read among others.

For example, if learners' usage data in the moodle suggests that, the user has spent 3 minutes reading a web page, or clicked on a relevant link, then this information indicate a learners' interest in a particular course which could be transformed in to ratings for enriching 'user item rating matrix'. This enriched matrix will be used by collaborative filtering algorithm such as '*k*-nearest neighbour' in order to generate a list of recommendation on data mining courses for different type of learners based on their profile. The list of recommended courses will further be evaluated through suitable evaluation metrics such as RMSE, precision and recall in order to measure the accuracy of recommendations. Table 6.1 shows a sparse 'user item rating matrix' which contains the ratings provided by different learners L1 to L7 to three courses C1 to C3.

6.2.6 Jena Inference Engine

In the field of artificial intelligence, inference engine is the part of a system which infers new information by applying logical rules to the available facts in the database. One can find an 'inference module' as part of an expert system which typically consists of 'knowledge base' and 'inference engine'.

In the context of 'semantic web', an inference subsystem has been used successfully in deducing new knowledge from the available facts. The 'Jena inference engine' is used to derive some new information or RDF assertions which are extracted from RDF database. One

of the most important functions of 'Jena inference engine' is to support the use of languages such as RDFS (Resource Description Framework Schema) and OWL (Web Ontology Language).

In Figure 6.3, we can see the overall structure of Jena inference engine[221] in which applications normally access the inference machinery by using the 'ModelFactory' in order to associate a 'data set' with some reasoner for creating a new Model. Queries to the created model will return not only those statements that were present in the original data but also additional statements than can be derived from the RDF database using the rules or other inference mechanisms implemented by the reasoner. For example, if one of the existing rules is A->B, which means, if a user has liked an item 'A' then he would also be interested in item 'B' and there is another rule, B->C which says that, if a user has liked an item 'B' then he/she would also opt the item C. Now, the inference engine takes these two rules and infers another new rule such as A->C which suggests that, if a user has liked an item A, then he would also be interested in item C.

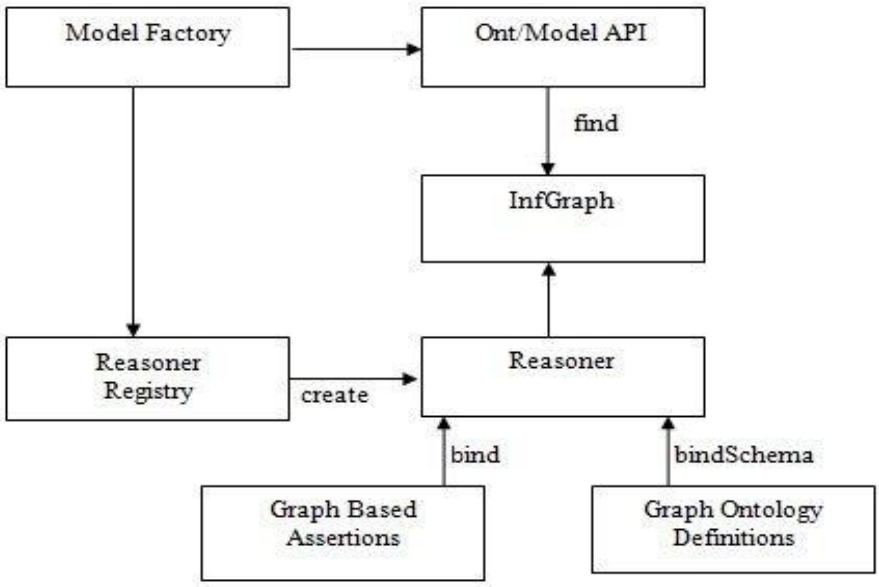


Figure 6.3 Overall Structure of Jena Inference System

We have used 'Apache Jena Fueski' for creating 'RDF database' and implementing rules. It is a SPARQL server which is used as a standalone server in this research. In Figure 6.4, we can see the interface of 'Fueski inference engine'. Fueski is the project of Apache and provides an interface over HTTP for querying RDF factbase. The software can be run as a standalone machine or it can also be used as an operating system service.

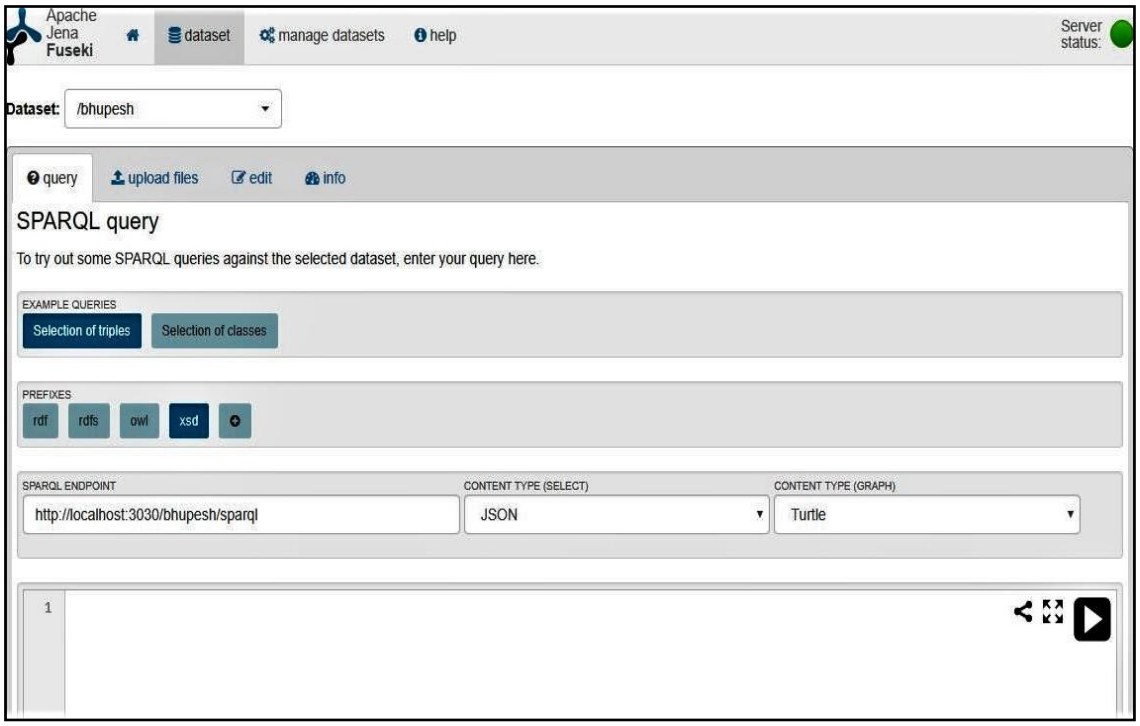


Figure 6.4 A Snapshot of Apache Jena Fueski Inference Engine

The enriched user 'item rating matrix' for active cluster is shown in Table 6.2 which has been enriched with additional ratings obtained through resource description framework. The collaborative filtering approach is applied to the enriched matrix for predicting ratings for unrated courses.

Table 6.2 A View of Enriched User Item Rating Matrix

Courses → **Learners** ↓	C1	C2	C3
L1	3	5	4
L2	3	3	1
L3	3	4	4
L4	2	5	4
L5	3	5	3
L6	5	4	2
L7	4	4	4

Table 6.2 above shows the predicted ratings extracted through RDF on different data mining courses. In the following section, we evaluate the proposed approach by using some of the most widely used evaluation metrics.

6.3 Experimental Evaluation and Discussion

A series of experiments were carried out in order to evaluate the accuracy and performance of the proposed recommender framework. The experimental results measured by 'M2' are compared with our own approach 'M1' in [224] which is based on k-means and K-NN algorithms. The results of the enhanced recommender framework (M2) are also compared with other similar approaches [225] [226] in terms of MAE and precision as these are the only evaluation metrics being used in these approaches. The results of the proposed recommender framework have been evaluated using the most commonly used evaluation metrics such as RMSE, precision, recall and F1 [227].

The proposed recommender framework makes use of collaborative filtering (CF) algorithm for making recommendations. In CF, the size of neighbourhood (k) effects the quality of recommendations. Hence, we also conducted an experiment in order to determine the best value of 'k' which we obtained at k=5 as shown in Figure 6.5.

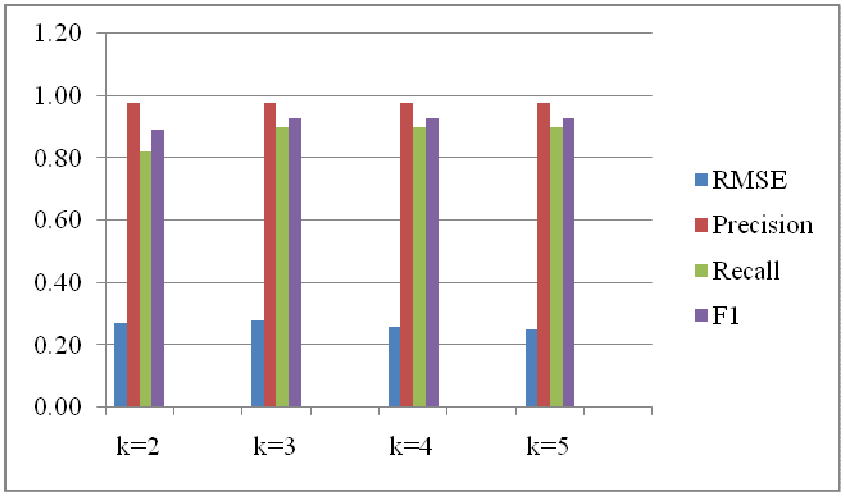


Figure 6.5 Determining the Best Value of 'k' (size of neighbourhood) to Evaluate Recommendations

The accuracy of the proposed framework is evaluated using 'precision' which is the proportion of the relevant documents out of the total retrieved documents. We measured the value of precision using equation (1).

$$\text{Precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \qquad (1)$$

We also evaluated the performance of the proposed framework in terms of 'recall' which refers to the ratio of retrieved relevant items divided by total relevant items in a database. The value of recall is computed using equation (2):

$$\text{Recall} = \frac{|\{\text{relevant document}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|} \quad (2)$$

Another widely used metric is 'F1' which combines both precision and recall and provides us a single score. It gives us more balance view of the performance of a system. We use the equation (3) in order to compute the value of F1.

$$F1 = 2\left(\frac{\text{precision.recall}}{\text{precision} + \text{recall}}\right) \quad (3)$$

We also use another useful metric named 'RMSE (root mean square error) which is used for evaluating the performance of the prediction model. In other words, the value of RMSE indicates how close the predicted ratings are to the actual rating. Its value varies from 1 to 0. The value of '1' suggests that, the produced prediction model has predicted ratings of item which are far away from their actual ratings. On the other hand, the value of '0' suggests that, the model has perfectly predicted the estimated rating. The value of RMSE is computed using equation (4). In the equation, the symbol '$p_{i,j}$' is used to denote the predicted or estimated ratings and '$v_{i,j}$' is the actual ratings provided by learner '*i'* to item '*j'*.

$$RMSE = \sqrt{\frac{\sum_{i,j \in K} p_{i,j} - v_{i,j}^{2}}{n}} \quad (4)$$

Moreover, the set k = {(*i*,*j*)} is the collection of hidden user item ratings. The metric is used to compute the difference between estimated ratings provided by an algorithm and actual ratings. These individual differences are termed as residual. The value of RMSE is always positive. Table 6.3 and Table 6.4 show the accuracy and performance of the two approaches which are represented with 'M1' and 'M2' for ease of understanding. The first approach 'M1' is based on *k*-means algorithm and collaborative filtering approach for recommending different data mining courses to learners falling in one of the three clusters namely not active, average and active [224].The second approach 'M2' which we have proposed in this chapter additionally uses one of the widely used semantic technology named resource description

framework (RDF) in order to infer additional preferences of a learner from the log file of moodle server. We also conducted experiments in order to evaluate the effectiveness of the approach 'M2'.

Table 6.3 Evaluation of Recommendations Based on Recommender Approach (M1)

Type of Cluster	RMSE	Precision	Recall	F1 Score
Not Active	0.73	0.60	0.31	0.40
Average	0.60	0.90	0.72	0.79
Active	0.39	0.70	0.27	0.39

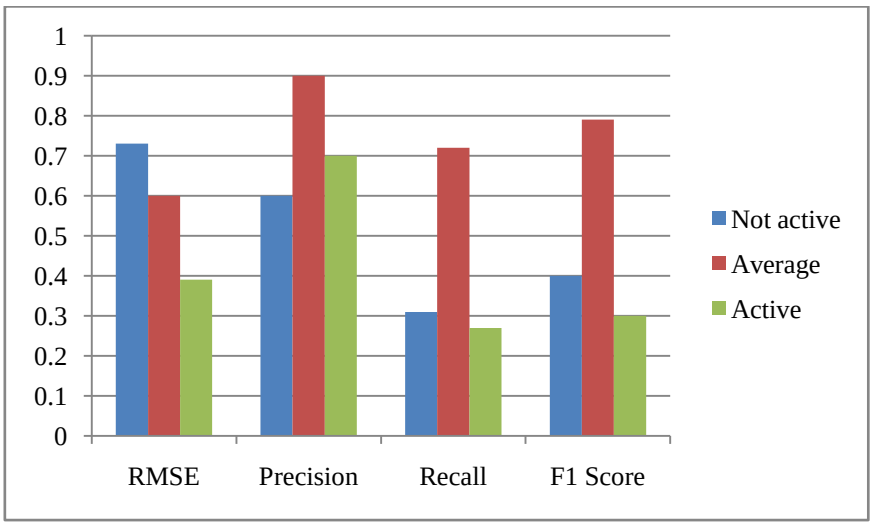


Figure 6.6 Evaluation of Recommendations for Different Clusters Based on Approach (M1)

For ease of understanding of the comparison of these two approaches (M1, M2), we have also shown the results graphically in Figure 6.8. The accuracy of the recommendations of 'M2' is measured using 'precision' which yields a highest value of 0.98 for learners belonging to active cluster as compared to a precision of 0.90 obtained by the approach 'M1'. This is due to the improvement in the density of 'rating matrix' as the less sparse a matrix is the better the quality of recommendations.

Table 6.4 Evaluation of Recommendations Based on Recommender Approach (M2)

Type of Cluster	RMSE	Precision	Recall	F1 Score
Not Active	0.51	0.64	0.27	0.38
Average	0.48	0.97	0.84	0.90
Active	0.25	0.98	0.90	0.93

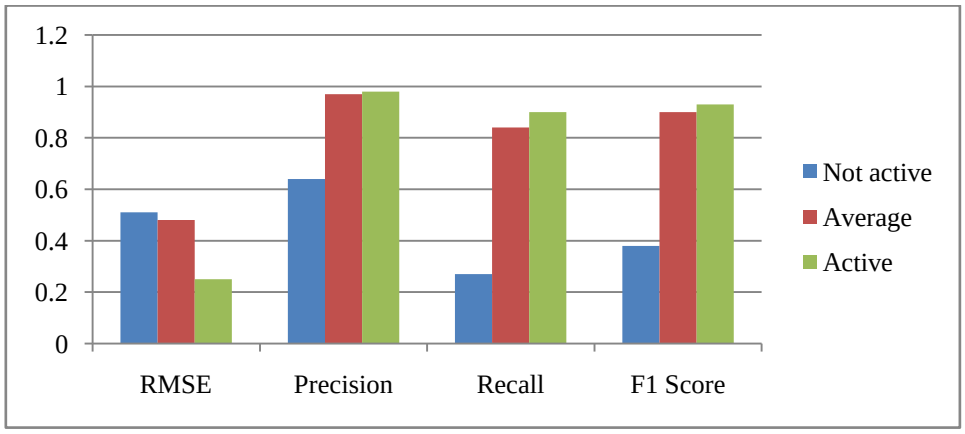


Figure 6.7 Evaluations of Recommendations for Different Clusters Based on Approach (M2)

The recommendations are evaluated @ precision '10' which means, the proportion of the relevant recommendations is being computed out of 10 courses. Therefore, a precision of 0.98 indicates that, the accuracy of the prediction model is 98%. In other words, we can say that, the recommender approach is able to suggest at least 9 courses accurately to learners as compared to 7 courses obtained by approach 'M1' based on learners' profile.

Moreover, if we compare the recall value obtained by 'M1' as 72% to the recall value of 0.90 measured by 'M2', a significant improvement can be seen. The significance of the improvement in this recall value suggests that, while approach M1 was able to retrieve 72% of relevant courses out of the total relevant courses in the course database, the second approach is able to suggest 90% of the relevant data mining courses to learners based on their profile. Therefore, we have been able to achieve an improvement of 18% in terms of recall value. This improvement is attributed to the fact that, the collaborative filtering algorithm is applied to the dense 'user item rating matrix' which is obtained through resource description framework and apache Jena rules.

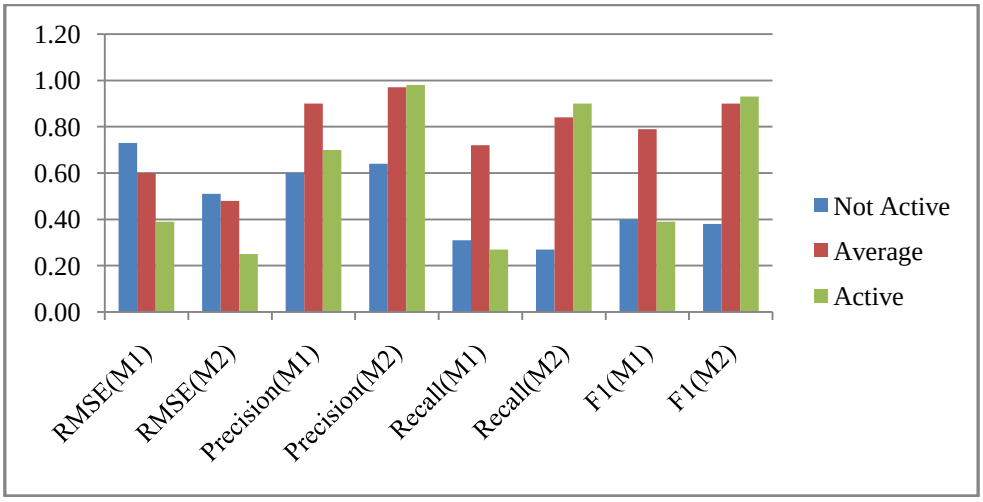


Figure 6.8 Comparisons of Recommender Approaches (M1) and (M2)

In addition, we also compared these two approaches 'M1' and 'M2' with respect to RMSE which is one of the widely used predictive accuracy metrics for measuring the accuracy of predicted rating. The value of RMSE lies between '0' and '1' where '0' indicates a perfect estimated rating and '1' suggest that the predicted score is far away from the actual ratings. Furthermore, Table 6.3 and Table 6.4 show that, the second approach 'M2' managed to achieve a decent value of RMSE which is 0.25 as against 0.39 obtained by 'M1'. This further suggests that, the second approach is able to achieve more accurate prediction than the first approach 'M1'.

Additionally, we also compared the two approaches in terms of 'F1' metric which gives more balance view of the performance and gives equal weight to precision and recall. The value of F1 as measured by 'M1' is 0.79 as against 0.93 computed by 'M2'. It is apparent from Figure 6.8 that, the performance of 'M2' is better than the approach 'M1'.

We also compared the results obtained using 'M2' approach with the results of other similar approaches with respect to accuracy and quality of recommendations in addition to our previous approach. In one such comparison [226] the authors have proposed a recommender system based on *collaborative filtering* and *ontology* (which we have represented by 'M3') for suggesting learning materials to learners based on their learning characteristics which are stored in ontological form. They experimentally evaluated the performance of their approach with respect to accuracy and performance which are measured using MAE (mean absolute error) and F1 evaluation metrics.

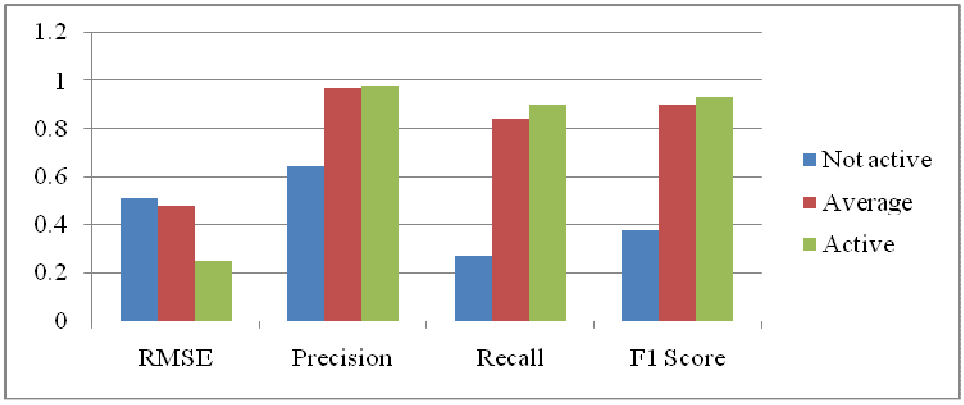


Figure 6.7 Evaluations of Recommendations for Different Clusters Based on Approach 'M2'

The recommendations are evaluated @ precision '10' which means, the proportion of the relevant recommendations is being computed out of 10 courses. Therefore, a precision of 0.98 indicates that, the accuracy of the prediction model is 98%. In other words, we can say that, the recommender approach is able to suggest at least 9 courses accurately to learners as compared to 7 courses obtained by approach 'M1' based on learners' profile.

Moreover, if we compare the recall value obtained by 'M1' as 72% to the recall value of 0.90 measured by 'M2', a significant improvement can be seen. The significance of the improvement in this recall value suggests that, while approach M1 was able to retrieve 72% of relevant courses out of the total relevant courses in the course database, the second approach is able to suggest 90% of the relevant data mining courses to learners based on their profile. Therefore, we have been able to achieve an improvement of 18% in terms of recall value. This improvement is attributed to the fact that, the collaborative filtering algorithm is applied to the dense 'user item rating matrix' which is obtained through resource description framework and apache Jena rules.

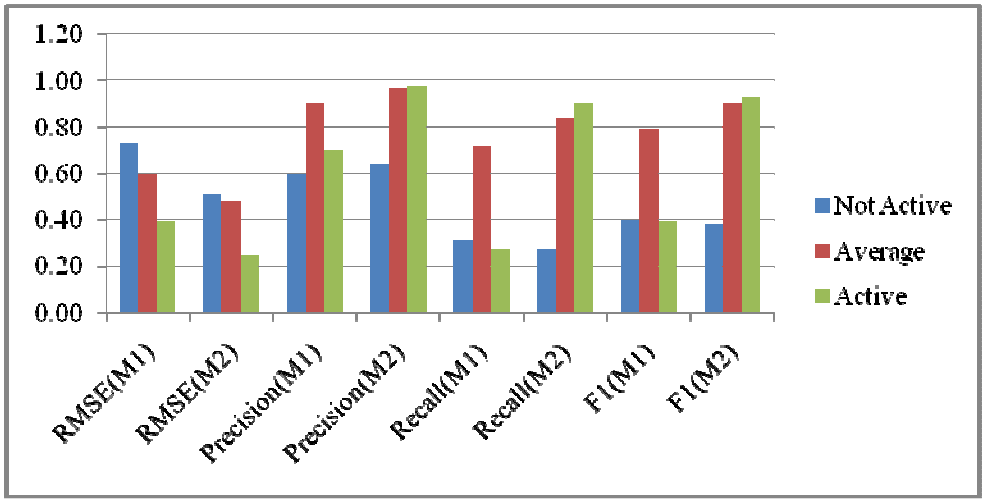


Figure 6.8 Comparisons of Recommender Approaches 'M1' and 'M2'

In addition, we also compared these two approaches 'M1' and 'M2' with respect to RMSE which is one of the widely used predictive accuracy metrics for measuring the accuracy of predicted rating. The value of RMSE lies between '0' and '1' where '0' indicates a perfect estimated rating and '1' suggest that the predicted score is far away from the actual ratings. Furthermore, Table 6.3 and Table 6.4 show that, the second approach 'M2' managed to achieve a decent value of RMSE which is 0.25 as against 0.39 obtained by 'M1'. This further suggests that, the second approach is able to achieve more accurate prediction than the first approach 'M1'.

Additionally, we also compared the two approaches in terms of 'F1' metric which gives more balance view of the performance and gives equal weight to precision and recall. The value of F1 as measured by 'M1' is 0.79 as against 0.93 computed by 'M2'. It is apparent from Figure 6.8 that, the performance of 'M2' is better than the approach 'M1'.

We also compared the results obtained using 'M2' approach with the results of other similar approaches with respect to accuracy and quality of recommendations in addition to our previous approach. In one such comparison [226] the authors have proposed a recommender system based on *collaborative filtering* and *ontology* (which we have represented by 'M3') for suggesting learning materials to learners based on their learning characteristics which are stored in ontological form. They experimentally evaluated the performance of their approach with respect to accuracy and performance which are measured using MAE (mean absolute error) and F1 evaluation metrics.

The authors measured a lowest MAE of 0.55 as compared to 0.25 obtained by our approach (M2). Furthermore, they also measured an F1 score of 0.42 as compared to 0.93 measured through our approach 'M2'. The significant improvement in the value of F1 and RMSE is largely attributed to the 'dense sparse rating matrix' which we have been able to achieve using RDF.

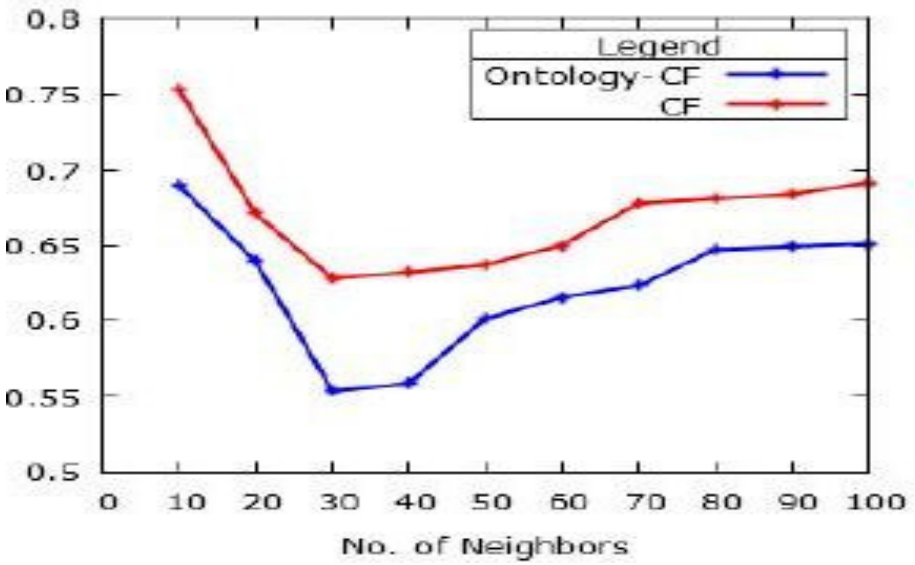


Figure 6.9 Evaluations of Recommendations by other Similar Approach 'M3' [226]

We have also compared our approach with other similar approaches. In [225], the authors proposed a recommender system (which we have denoted by 'M4') based on item's semantic information and user's historical rating data. The evaluation of the proposed approach is carried out using several well known evaluation metrics such as precision, F1 and MAE (mean absolute error). Experimental results show that, the best value of precision is 0.90 by the proposed approach 'M4' as compared to 0.98 achieved by our approach M2.

In addition to this, we have achieved a slight improvement in the value of F1 as 0.93 by M2 as compare to the one obtained by 'M4' as 0.92. It is to be noted that, method 1,2, 3 and 4 are the proposed techniques, the proposed technique with normalizing the Pearson similarity, Pearson correlation based technique and cosine based technique respectively.

Chapter 7

Conclusions

In this thesis, we have proposed a recommender system which recommends different categories of data mining courses to different learners based on their profile. The recommender system makes use of resource description framework which is one of the widely used semantic tools for inferring the additional preferences about learners over unrated courses in 'user item rating matrix' which help to improve the accuracy of recommendations. In this chapter, the thesis work is being concluded with the summary of work done, outcomes derived from the thesis, limitation of our research work, concluding remarks and future scope of the work.

7.1 Thesis Summary

As discussed in the previous chapters, that the majority of existing e-learning recommender systems suffer from the issue of 'one size fits all' in which the recommender system suggests the same learning resources to all the learners without taking into account their differences in terms of level of knowledge, skill, and interest among others. This leads to the drop in the subject performance of a learner and hence overall fall in his/her academic performance. Moreover, we also addressed the issue of not having 'cluster validation mechanism' in the existing machine learning tools such as WEKA due to which it is often difficult to determine the exact number of clusters present in the dataset.

In addition to these issues, many of the e-learning recommender systems are suffering from the issue of 'sparsity' where the recommender systems do not have sufficient ratings in 'user item rating matrix' in order to provide good quality of recommendations. We have dealt with this issue by creating 'RDF factbase' and 'certain rules' based on the learning activities performed by the learners on the moodle platform and provided them to 'apache jena fueski' which is one of the commonly used 'jena inference engine' for enriching the 'user item rating matrix' in order to improve the accuracy of recommendations.

The thesis explored the above issues and challenges along with the requirements to realize the proposed objective. In order to achieve the proposed objectives, we have proposed a recommendation framework which suggests data mining courses to learners belonging to one of the three types of clusters. The thesis also aims to improve the quality of

recommendations by using one of the commonly used semantic tools named ‘apache jena fueski’ in order to elicit additional preferences of learners. The framework proposed in this thesis follows several steps in order to integrate the additional preferences of learners extracted implicitly from moodle server in the recommendation process. In the proposed work, we claim that, the accuracy of a recommender system can be enhanced by incorporating semantic knowledge in the process of recommendation.

In the experimental evaluation, the thesis also explored the performance of the different clustering algorithms. The most appropriate clustering algorithm may be used to generate clusters which are used as the profile of a learner. Out of the four clustering algorithms namely *k*-means, expectation maximization, agglomerative clustering and divisive clustering algorithm taken up for performance evaluation, the k-means is found to be the most appropriate algorithm for building clusters in terms of the time taken to build a model.

In addition, among the many ‘evaluation metrics’ available in the literature, it is crucial to find the most suitable one for the given recommender task. Hence, keeping this in mind, we also carried out an experimental evaluation consisting of several metrics such as precision, recall, and RMSE among other measure in order to find the best combination of a metric with a given recommender task and recommender algorithm.

7.2 Outcome derived from the thesis

We have tried to address some of the critical issues such as ‘one size fits all’, ‘sparsity’, ‘the absence of cluster validation mechanism in the existing machine learning tools’ and ‘not utilizing semantic knowledge’ in the recommendation process by which majority of e-learning recommender systems are suffering. The major outcomes of the thesis are as follows:

- A framework has been designed in order to recommend different categories of data mining courses based on learners’ profile. The proposed framework addresses the issue of ‘one size fits all’, ‘sparsity’ and ‘unavailability of cluster validation mechanism’ in machine learning tools.

- An experimental evaluation is carried out involving several well known clustering algorithms which discovers the most appropriate algorithm and is recommended for its appropriate usage in further experiments.

- The best combination of a 'recommender task' 'recommendation algorithm', and an 'evaluation metric' is found by conducting an experiment which involves the most widely used recommender algorithms, recommender tasks and well known evaluation metrics.

- The existing machine learning tools such as WEKA, and KEEL among others don't' have 'cluster validation mechanisms' hence we used 'elbow method' and 'silhouette method' in order to bridge this gap.

- The 'user item rating matrix' is enriched with the additional learners' preferences which are obtained implicitly by building 'RDF factbase' from learners' activities stored in the log file of the moodle server and 'creating rules'. These two are provided to the 'Apache jena Fueski' in order to predict the preferences of learners for unrated courses.

7.3 Limitations of our Research

In this thesis, we have tried to resolve issues such as 'information overload', 'one size fits all', 'sparsity' and 'use of semantic tools such as RDF' for improving the quality of recommendations by proposing a 'course recommender system' which suggests different 'data mining courses' to 'different type of learners' based on their profile.

Although we have been able to improve the accuracy of recommendations significantly, it can be further explored while considering the following:

- The proposed framework for recommender system has been tested in offline mode, which can be further explored with real time working model.

- In offline mode the quality of recommendations is limited by the density of 'user item rating matrix'. On the other hand, in online mode where we are having actual users and the ratings are generated while the learner is interacting with the system. The more a learner interacts with the system, the more a recommender system learns about the learner.

- One of the most widely and popular machine learning tools WEKA has been used for building and analyzing the clusters. The analysis may be limited by the number of parameters it offers. We used only those features in WEKA that are most relevant to

our research. However, other features can also be explored depending upon the requirement of the e-learning applications.

- In addition to 'WEKA', there are other well known data analysis tools such as Orange, 'RapidMiner', 'KNIME' and 'Neural Designer' among others which can also investigated for analyzing the large usage data stored in the moodle server.

- The programming environment that we have chosen for the implementation of the proposed framework includes 'Netbeans 8.2' and 'Apache Mahout' from 'Apache software foundation'. 'Apache' is mainly used as it provides all the necessary libraries required for the efficient implementation of the recommender system based on collaborative filtering algorithm. Furthermore, 'Netbeans' is one of the widely used integrated development learning environment (IDE) which is used in this research for displaying the list of recommendations along with their estimated ratings produced by collaborative filtering using 'apache mahout libraries'.

- However, in addition to these two programming environment, we can also explore other tools such as ELKI (Environment for Developing KDD-Applications Supported by Index-Structures), and KEEL ((Knowledge Extraction based on Evolutionary Learning among others as an alternative to 'Apache' and 'visual studio code', 'Eclipse', and 'Microsoft Visual Studio' among others as an alternative to 'Netbeans'.

- MATLAB which is one of the widely used open source tools has been used in this research in order to implement 'elbow' and 'silhouette' methods for evaluating the quality of clusters. We couldn't try other similar tools due to time constraints. However, other similar tools such as Scilab, Sage and GNU octave among others can also be investigated in order to analyze the clusters in a more efficient way.

- In offline mode, updating the 'RDF factbase' is a significant additional effort as it requires periodic update of the related RDF datasets. It is important to auto update the factbase as it reflects the current preferences of learner which are used to provide recommendations. Any change in learners' behaviour needs to be reflected in the 'factbase'.
- One of the important issues that need to be catered while using the proposed framework is to evaluate the quality of cluster which is done manually in this research but needs to be done automatically from within the machine learning tools.

- The dataset used in the proposed framework is small, hence the same framework can be explored with larger dataset in order to measure the effectiveness of the framework in terms of scalability.

- The learners' usage data that we have used in this research is small, hence it does not contain much noise, outlier, incompleteness, and inconsistency among others. Therefore, not all steps of data pre-processing are required. However, in case of large dataset, the above anomalies might creep into a dataset and some or all data pre-processing steps can be explored depending upon the nature of dataset.

- Although there exist quite a large number of clustering algorithms, we have used *k*-means algorithm due to its efficiency in terms of running time which is measured experimentally in the second chapter. However, the algorithm provides a quite a large number of parameters all of which have not been explored. We used only the most relevant parameters, the other one can be also be explored.

- Furthermore, in addition to *k*-means, other clustering algorithms such as DBSCAN, Expectation maximization, Heiarchical clustering, and mean shift clustering among others which has quite a large number of features can also be explored with large dataset.

- One of the important issues that need to be addressed while using the proposed framework is the updation of learners' profile with the changes in the learning goals of a learner.

- The framework proposed in this thesis is developed using a stand-alone computer system but can also be explored in a client server setup.

- The proposed recommender system can be explored in order to integrate it in an existing e-learning system so as to recommend courses based on the profile of a learner which leads to the improvement in the quality of recommendations.

- We have considered 'data mining courses' for recommendations, however other popular courses such as 'software engineering', 'operating system' and 'data structure' among others can also be explored in order to strengthen the coverage of the recommender system.

7.4 Concluding Remarks and Future Scope

The huge information generated in the existing ‘ learning management systems’ presents a challenge for users to find the right kind of learning material which best matches with their learning goals, interest and level of knowledge. In this thesis, we have addressed this issue by proposing a recommendation framework which recommends different data mining courses to learners based on their profile. We have also explored and provided the tools and techniques to deal with such issue.

- The large amount of information stored in the ‘learning management system’ such as ‘Moodle’ which is generated as a result of learners’ interaction with the system serves as a knowledge base to which various ‘machine learning algorithms’ can be applied in order to discover the learners’ characteristics for recommending courses according to their profile.
- The learners’ usage data is in the raw form and needs ‘data pre-processing’ in order to prepare the data for applying data mining algorithm. Although data pre-processing consists of several steps, we performed only those steps which are most relevant in our case.
- The proposed approach extracts attributes from the different tables of moodle server which are adequate for our purpose. The extracted attributes are further used for analyzing the various learners and group them into different clusters based on their similar learning patterns.
- The obtained clusters are further validated through ‘elbow’ and ‘silhouette’ methods in order to ensure their quality. *K-NN* is used for the classification of a new learner into one of the clusters and experimental evaluations are carried out in order to check the validity of classifier.
- The proposed approach employs RDF(resource description framework) for representing moodle activites from moodle information model. RDF factbase which consists of triplets is constructed in order to apply jeena inference engine to this database.
- Well known statistical evaluation metrics are used in order to evaluate the quality of recommendations.

- This thesis can be seen as the first step towards representing moodle activities through RDF(resource description framework) for eliciting learners' preferences over set of data mining courses in order to 'enrich user item rating matrix' for improving the quality of recommendations.

- For future work, we propose to incorporate 'clusters validation mechanism' into the existing machine learning tools which will make it possible to automatically validate the quality of clusters obtained through clustering algorithms.

- Another issue that we plan to incorporate in our future work is to automatically update the profile of learners based on the feedback obtained from learners'' activities performed on moodle server which will help us to capture the change in the learning goals and interest of learners and they can be recommended the courses accordingly.

- Although, we have employed the most appropriate classifier for classifying a learner into its appropriate class, other classifiers with additional features such as C4.5, PART (partial decision tree) and random forest algorithms among others can also be explored. Hence, it is also proposed to work in future for the exploration of other classifiers (classification algorithms) in order to further improves the accuracy of the results of learners' classification.

- Although to the best of our knowledge, we have employed two of the most widely used cluster validation mechanisms in order to evaluate the quality of clusters, there are other methods with rich parameters and options such as 'information criterion approach', an 'information theoretic approach', 'cross validation' and 'rule of thumb' among others which can also be explored further for obtaining more accurate clusters.

- In this research, we have used quite a large number of features of moodle such as assignment, messaging, and forum among others. However, due to time constraint we couldn't explore other useful features of the moodle such as 'survey and choice', 'grade and scale', glossaries and lessons among others which if used in an e-learning recommender system can enrich the experience of learners and hence improve their overall academic performance.

- In addition to moodle, there are other learning management systems(LMS) with many features such as BlackBoard, WebCT and Sakay among others which can also be

explored as a learning platform in order to provide learner a better learning experience which meet their learning goals.

- With the emergence of social networking sites, it has become possible to learn about the preferences of learners through other sources such as annotation among others. This area could be explored further in order to learn more about learners' preferences which finally leads to improving the quality of recommendations.
- We have used 'apache jena' as a standalone server for implementing RDF factbase and the set of rules in order to enrich the sparse user item rating matrix. One of the major reasons for using this tool is that, it is mainly meant for developing semantic web applications and it is also open source. We have specifically used 'apache Jena fueski'. However other similar tools such as Microsoft .NET, Apache commons and Apache CFX can also be investigated in order to achieve better performance of execution of query which is fired against the RDF factbase.

References

[1] Rousseeuw,J.P.(1987).Silhouettes:a Graphical Aid to the Interpretation and validation of Cluster Analysis. Computational and Applied Mathematics, 20, 53–65.

[2] Dolog, P., Henze, N., Nejdl, W. and Sintek, M. (2004). Personalization in distributed e-learning environments, In proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, 17–20 May,170–179, New York,NY,USA.

[3] Chen, C.M., Lee, H. M., Chen, Y.H. (2005). Personalized e-learning system using Item Response Theory.Computers & Education, 44(3), 237–255.

[4] Duffy, T.M., & Kirkley,J.R. (2010).Learner-centered Theory and Practice in Distance Education*, Lawrence Erlbaum Associates*, Inc., New Jersey, USA.

[5] Ragab,M. A.,Noaman,A. Y, AL-Ghamdi, S. A., Madbouly,I. A. (2014). A Comparative Analysis of Classification Algorithms for Students College Enrollment Approval Using Data Mining. *In Proc of the workshop on Interaction Design in Educational Environment.*

[6] Zaíane,O.R.(2002).Building a recommender agent for e-learning systems.*In Proceeding of the International Conference on Computers in Education*, 3–6.

[7] Khribi, M.K., Jemni, M., Nasraoui, O. (2009). Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval,*Educational Technology & Society*, 12(4), 30–42.

[8] Tai, D.W., Wu, H.,Li, P. (2008). Effective e-learning recommendation system based on self-organizing maps and association mining.*The Electronic Library Journal*, 26(3), 329–344.

[9] Salehi,M., Pourzaferani,M., Razavi, S. A.(2013). Hybrid Attribute-Based Recommender System for Learning Material Using Genetic Algorithm and a Multidimensional Information Model, *International Journal of Egyptian Informatics*, 14(1), 67-78.

[10] Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values. *In Proc of the First Pacific Asia Knowledge Discovery and Data Mining conference* ,Singapore: World Scientific, 21–34.

[11] Anane,R.,Crowther,S.,Beadle,J et al.(2004).e-Learning content provision.*In: Proceedings of the 15th international workshop on database and expert systems applications*.420-425.

[12] Balabanovi´cM, Shoham, Y. (1997). Fab: content based, collaborative recommendation. *Communication of ACM*, 40,66– 72.

[13] Gong,S., Ye,H., Tan,H. (2009). Combining Memory-Based and Model-Based collaborateve Filtering in Recommender System. *In Proceeding of Pacific Asia conference on circuits,communications and systems*.IEEE.

[14] Ricci,F.,Rokach,L., & Shapira, B. (2011). Introduction to Recommender Systems Hand -book. *In Recommender Systems Handbook*, 1-29. New York: Springer.

[15] Klasnja-Milicevic, A., Ivanovic, M., Nanopoulos, A. (2015).Recommender systems in e- learning environment: a survey of the state of the art and possible extensions. *Artificial Intelligence Review*, Springer , 44(4),571-604.

[16] Adomavicius,G., Tuzhilin, A. (2005).Toward the next generation of recommender system :A survey of the state of the art and possible extensions. *IEEE Transaction on knowledge and Data Engineering*, 17(6),734–749.

[17] Aha, D., Kibler, D., Albert, M. K. (1991), instance- based learning algorithms, *Machine learning*, 6(1), 37-66.

[18] Mallinson,B., Sewry, D. (2004). E-learning at Rhodes University: a case study. *In:Proc of the IEEE international conference on advanced learning technologies* (ICALT'04), 708–711.

[19] Hafner,J., Sawhney,H., Equitz, W., Flickner, M., & Niblack, W. (1995). Efficient Color Histogram Indexing for Quadratic Form Distance Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,17(7), 729-736.

[20] Lin,F.,Zhou,X., Zeng, W. H. (2016). Sparse online learning for collaborative filtering, *International Journal of Computers Communications and Control*,11(2),248-258.

[21] Ma,X.,Lu,H.,Gan,Z. (2014). Improving recommendation a ccuracy by combining trust communities and collaborative filtering, *In proc of the 23rd ACM international conference on information and knowledge management*, 1951-1954.

[22] Gunawardana,A., Shani,G.(2009). A Survey of Accuracy Evaluation Metrics of Recom- mendation Tasks. *The Journal of Machine Learning Research*, 10 (2009), 2935-2962.

[23] Kaly,S.,Kumar,R.,Vassilvitskii,S.(2011).Cross-validation & mean-square stability.*In Proc of the second symposium on innovations in computer science*,Tsinghua University,487-495.

[24] Billsus, D., Pazzani, M.(1998). Learning collaborative information filters .*In Proceeding of international conference on machine learning*,46–54.

[25] Stehman.,Stephen,V.(1997).Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*. 62 (1), 77–89.

[26] Herlocker,H.L., Konstan,J.A., Terveen,L. G., Riedl,J.T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transaction on Information System*, 22 (1),5-53

[27] Wilson,D.,Smyth, B.,Sullivan, D.(2003). Sparsity Reduction in Collaborative Recomme- ndation: A case-based approach. *International Journal of Pattern Recognition*,17(5),863- 884.

[28] Guo, G. (2012). Resolving Data Sparsity and Cold Start in Recommender Systems.In: Masthoff,J.,Mobasher,B.,Desmarais,M.C.,Nkambou,R. (eds). User Modeling, Adaptation and Personalization. UMAP 2012. *Lecture Notes in Computer Science*, 7379.Springer , Berlin, Heidelberg

[29] Su,X.,Khoshgoftaar,T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*.Hindwai,1-19.

[30] Iquinta,L.,Gemmis,M.D.,Lops,P.,Semeraro,G.,Filananino,M.,Molino,P.(2008).Introduci -ng serendipidity in a content based recommender system.*In proceeding of 8th International Conference on Hybrid Intelligent system*.IEEE.

[31] Recker,M.,Walker,A.(2003).Supporting 'word-of-mouth'social networks via collaborative -e information filtering. *Journal of Interactive Learning Research*,14(1),79–98.

[32] Recker,M., Walker, A. Lawless,K.(2003).What do you recommend? Implementation and analyses of collaborative filtering of web resources for education. *Instructional Science*, 31,229–316.

[33] Walker,A., Recker,M., Lawless,K., Wiley, D. (2004). Collaborative information filtering :a review and an educational application. *International Journal of artificial intelligence and education*,14,1–26.

[34] Tan,H.,Guo,J., Li, Y.(2008) E-learning recommendation system. *In: International Confer ence on computer science and software engineering*, csse, 5, 430–433.

[35] Agrawal, R., Imielinski, T., & Swami, A. N. (1993). Mining association rules between sets of items in large databases, *In Proc of the 1993 ACM SIGMOD International Conference on Management of Data*.

[36] Capo,M.,Perez,A., Lozano,A.(2015).An efficient k-means clustering algorithm for massive data.*Journal of Latex Class Files*,14(8),1-14.

[37] Dunn, J. C.(1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*. 3 (3), 32–57.

[38] Saman, S., Seyed, Y. B., Nor, A. M. Z., Shahrul,A. M. N.(2012).Ontological approach in knowledge based recommender system to develop the quality of e-learning system. *Australian Journal of Basic and Applied Sciences*, 6(2), 115-123.

[39] Billsus, D., Pazzani,M.(2007). Content-based recommendation systems: In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *The Adaptive Web. Methods and Strategies of Web Personalization*,325–341. Springer, Berlin.

[40] Dou,Y.,Yang,H.,Deng,Z.(2016). A Survey of Collaborative Filtering Algorithms for Social Recommender Systems.*In Proc of 12th international conference on semantics,knowledge and grid*.IEEE.

[41] Stumme,G., Hotho,A., Berendt,B.(2002).Usage mining for and on the semantic web, *In: National Science Foundation Workshop on Next Generation Data Mining*.

[42] Lu, J. (2004). A Personalized e-learning material recommender system.*In: Proceedings of the 2nd international conference on information technology for application.* England, London.

[43] Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Discovery and evaluation of aggregate usage profiles for web personalization.*Data Mining and Knowledge Discovery* ,6, 61-82.

[44] Cho, Y. H., Kim, J. K., Kim, S. H. (2002). A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications*, 23(3), 329–342.

[45] Srivastava,J., Colley,R., Deshpande,M.,Tan,P.(2000).Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data.*SIGKDD Exploration*,1(2).

[46] Noor,R.,Khan,F.A.(2017).Recommendation Strategies for Personalize Mobile Educational Systems.*International Journal of Computational Linguistics Research*,8(1).

[47] Chen, A., McLeod, D. (2006). Collaborative Filtering for Information Recommendation Systems.*Encyclopedia of E-Commerce, EGovernment, and Mobile Commerce*.118-123.

[48] Stumme, G., Hotho, A., Berendt, B.(2006). Semantic web mining: State of the art and future directions. *Journal of Web Semantics*.Elsevier, 4(2), 124–143.

[49] Takacs,G.,Pilaszy,I.,Nemeth,B.,Tikk,D.(2009). Scalable Collaborative Filtering Approaches for Large Recommender Systems.*Journal of Machine Learning Research*,10,623-656.

[50] Moghaddam,S.G.,Selamat,A.(2011). A scalable collaborative recommender algorithm based on user density-based clustering.*In Proc of 3rd international conference Data mining and intelligent information technology applications*.

[51] Pazzani, M.(1999). A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, 13(5-6), 393-408.

[52] Chen, M., Chiu, A., Chang, H. (2005). Mining Changes in Customer Behavior in Retail Marketing. *Expert Systems with Applications*,28(4), 773-781.

[53] Adomavicius,G., Sankaranarayanan, R., Sen,S., Tuzhilin, A. (2005). Incorporating contextual Information in Recommender Systems Using a Multidimensional Approach. *ACM Transactions on Information Systems*, 23(1), 103-145.

[54] Maes, P.(1994). Agent that Reduces Work and Information Overload. *Communication of the ACM*,37(7), 31-40.

[55] Herlocker,J.L.,Konstan,J.A.,Borchers.,Riedl,J.(1999). An Algorithmic Framework for Performing Collaborative Filtering.In *Proc on 22nd Annual International ACM SIGR conference on Research and Development in Information Retrieval*.

[56] Zhang,Q.,Segall,R.S.(2008). WEB MINING: A SURVEY OF CURRENT RESEARCH, TECHNIQUES, AND SOFTWARE. *International Journal of Information Technology & Decision Making*,7(4),683-720,World Scientific.

[57] Vialardi, C., Chue, J., Peche, J. P., Alvarado, G., Vinatea, B., Estrella, J., & Ortigosa, A. (2011). A data mining approach to guide students through the enrolment process based on academic performance. *User Modeling and User- Adapted Interaction*, 21(1),217-248.

[58] Konstan, A. J.,McNee, S. M ., Ziegler, C. N., Torres,R., Kapoor, N., & Riedl,J.T. (2006). Lessons applying automated recommender systems to information-seeking tasks. *In Proceedings of the Twenty-first National Conference on Artifical Intelligence (AAAI).*

[59] Bendakir, N.,Aimeur,E.(2006).Using Association Rules for Course Recommendation .*In Proceedings of the AAAI Workshop on Educational Data Mining.*

[60] Sobecki, J., Tomczak, J. M. (2010). Student courses recommendation using ant colony optimization. *In Proceedings of the Second international conference on*

Intelligent information and database systems.

[61] Shardanand,U., Maes,P.(1995).Social Information Filtering:Algorithms for Automating Word of Mouth. *In: Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, 1, 210–217.

[62] Ekstrand, M., Riedl, J., & Konstan, J. A. (2010). Collaborative Filtering Recommender Systems. *In: Foundations and Trends® in Human-Computer Interaction*, 4(2), 81–173.

[63] Buder,J.,Schwind,C.(2012).Learning with personalized recommender systems:a psychological view. *Computers in Human Behavior*,28(1): 207−16.

[64] Hofmann,T.(2004). Latent semantic models for collaborative filtering. *In: ACM Transactions on Information Systems*, 22(1),89–115.

[65] Oard, D.W., Kim, J.(1998).Implicit feedback for recommender systems.*In: Proceedings of 5th DELOS workshop on filtering and collaborative filtering*;31−36.

[66] Rich, E. (1979). User Modeling via Stereotypes. *Cognitive Science*, 3(4), 329–354.

[67] Shimodaira, H. (2014). Similarity and Recommender Systems. *School of Informatics*,The university of Edinburgh.

[68] Dwivedi,S.K.,Rawat,B.(2016). A review paper on data processing: A critical phase in web usage mining process, *In Proc of the IEEE conference on Green Computing and Internet of Things(ICGCIoT).*

[69] Sarwar,B.M.,Karypis,G., Konstan,J & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms". *In: ACM WWW* '01, 285–295.

[70] Linden, G., Smith, B., & York,J. (2003). Amazon.com recommendations: item-to - item collaborative filtering. *In: IEEE Internet Computing*, 7(1),76–80.

[71] Deshpande,M., Karypis,G. (2004). Item-based top-N recommendation algorithms". *In:ACM Transactions on Information Systems*, 22(1),143–177.

[72] Thai-Nghe,N., Drumond,L.,Horvath,T.(2011).Matrix and tensor factorization for predicting student performance, *In proceedings of CSEDU*.

[73] Thai-Nghe,N., Drumond,L., Horvath,T.,Schmidt-thieme,L. (2011). Multi-Relational Factorization models for predicting student's performance, *In Proc. of the KDD Workshop on Knowledge Discovery in Educational Data.*

[74] Paterek,A.(2007). Improving regularized singular value decomposition for collaborative filtering*, In Proceedings of KDD Cup and Workshop*.

[75] Lu, J.,Wu,D.,Mao,M.,Wang,W., & Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems*, 74, 12–32.

[76] Hofmann,T.(2004). Latent semantic models for collaborative filtering. *In: ACM Transactions on Information Systems*, 22(1),89–115.

[77] Salakhutdinov,R., & Mnih,A.(2008). Probabilistic Matrix Factorization. *In: Advances in Neural Information Processing Systems,*20,1257–1264.

[78] Xie,W., Dong,Q., & Gao,H.(2014).A Probabilistic Recommendation Method inspired by Latent Dirichlet Allocation Model. *Mathematical Problems in Engineering*.

[79] Dwivedi,P., Bharadwaj, K. K. (2015). e-Learning recommender system for a group of learners based on the unified learner profile approach.*Expert Systems*,Wiley,32(2).

[80] Basu, C., Hirish, H.,& Cohen,W.(1998).Recommendation as Classification: Using Social Social and Content-based Information in Recommendation. *In Proc of the 15th National Conference on Artificial Intelligence*,714-720.

[81] Bouihi, B., Bahaj, M. (2018) A Semantic Web Architecture for Context Recommendation System in E-learning Applications. In: Ben Ahmed M., Boudhir A. (eds) Innovations in Smart Cities and Applications. SCAMS 2017. *Lecture Notes in Networks and Systems*, 37. Springer, Cham.

[82] Burke, R.(2007). Hybrid web recommender systems. *In: The AdaptiveWeb*,377–408

[83] Myszkorowski,K., Zakrzewska, D. (2013). Using Fuzzy Logic for Recommending Groups in E-Learning Systems. In: Bădică C., Nguyen N.T., Brezovan M. (eds) Computational Collective Intelligence. Technologies and Applications. ICCCI 2013. *Lecture Notes in Computer Science*, 8083. Springer, Berlin, Heidelberg.

[84] Dwivedi, P., Bharadwaj, K.K. (2013). A Fuzzy Approach to Multidimensional Context Aware e-Learning Recommender System. In: Prasath R., Kathirvalavakumar T. (eds) Mining Intelligence and Knowledge Exploration. *Lecture Notes in Computer Science,*8284. Springer, Cham.

[85] Pu,P.,Chen,L.(2005).Integrating tradeoff support in product search tools for e-commerce sites.*In Proc of the 6th ACM Conference on Electronic Commerce (EC'05*), ACM,269-278.

[86] Adamopoulos,P., Tuzhilin,A. (2014). On over-specialization and concentration bias of recommendations. *In Proc. of RecSys'14*, RecSys '14, 153–160, New York,USA ,ACM.

[87] Harper, F.M.,Konstan,J.A. (2016). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS,)*,5(4).

[88] Zhang,W.(2008). Relational distance-based collaborative filtering for E-Learning. *In Proc of the 2008 International Symposium on Computational Intelligence and Design.*

[89] Cleary, J. G., Trigg, L. E.(1995). An Instance Based Learner Using an Entropy Distance Measure. *In 12th International Conference on Machine Learning.*

[90] Frank,E.,Hall,M.,& Pfahringer,B.(2003).Locally weighted Naïve Bayes. *In 19th conference in Uncertainty in Artificial Intelligence.*

[91] Aha,D.,& Kibler,D.(1991).Instances based learning algorithms.*Machine learning*,6,37-66.

[92] Kumar A., Sharma, A. (2013). Alleviating Sparsity and Scalability Issues in Collaborative Filtering Based Recommender Systems. In: Satapathy S., Udgata S., Biswal B. (eds) *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications* (FICTA). Advances in Intelligent Systems and Computing,199. Springer, Berlin, Heidelberg.

[93] Facca,F.M., & Lanz,P. I.(2005). Mining interesting knowledge from web logs: a survey.*Journal of Data and Knowledge Engineering*, 53(3),225-241.

[94] Sungjune, P. S., Suresh, N. C.,& Jeong, B. K. (2008). Sequence - based clustering for Web usage mining: A new Experimental framework and ANN-enhanced K-means algorithm. *Data & Knowledge Engineering*,65(3), 512–543.

[95] Liu, B. (2011). Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data(2nd ed.). Berlin: Springer.

[96] Sarwar, B. M., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommender algorithms for e-commerce. *In Proceedings of the 2nd ACM E-Commerce Conference,* Minneapolis, MN.

[97] Aher,S.,Lobo,L.M.R.J.(2013). Combination of machine learning algorithms for Recommmendation of courses in E-learning system based on historical data. *Knowledge-Based System*,51,1-14.

[98] Rawat,B.,Dwivedi,S.K.(2018). State of art Recommendation Approaches their issues and Future Research Direction in E-learning:A Survey.*International Journal of Advanced and Ubiquitous Computing*,10(4).

[99] Horrocks, I. (2002). DAML+OIL: A description logic for the semantic web. *IEEE Data Engineering Bulletin*, 25(1), 4-9.

[100] Horrocks,I., Sattler,U.(2001). Ontology reasoning in the SHOQ (D) description logic.*In17th International Joint Conference on Artificial Intelligence*,17(1),199-

204.

[101] Unadkat,R.(2015). Survey Paper on Semantic Web.*International Journal of Advanced Pervasive and Ubiquitous Computing*.IGI,7(4),13-17.

[102] Cole,J., & Foster.H.(2007). Using Moodle: Teaching with the popular open source course management system, O□Reilly Media.

[103] Dhillon, I., Modha,D. (2001). Concept Decomposition for Large Sparse Text Data Using Clustering. *Machine Learning*.42(2001),143-175.

[104] Dempster,A.P.Laird,N.M.,Rubin,D.B.(1977).Maximum Likelihood from incomplete Data via the EM Algorithm.*Journal of Royal Statistical Society*,39(1),1-38.

[105] Rokach, L., Oded, M.(2005).Clustering methods.*Data mining and knowledge discovery handbook*. Springer.(2005)321-352.

[106] Kaufman, L., Roussew, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. *A Wiley-Science Publication John Wiley & Sons*.

[107] Usama,F.,Gregory,P.,Shapiro.,padhraic,S.(1996).The KDD Process for Extracting useful Knowledge from Volumes of Data. *Communication of the ACM*.39(1996), 27-34.

[108] Edelstein, H.(1996). Mining Data Warehouses. *Information Week*.

[109] Chen,M.S.,Han,J.,Yu, P.S.(1996)Data mining:An overview from a database perspective.*IEEE Transactions on Knowledge and Data Engineering*,8 (1996),866-88.

[110] Han,J.,Kamber,M.,Pei,J.(2006).Data mining,southeast asia edition:Concepts and techniques Morgan kaufmann,

[111] Veyssieres, M.P., Plant, R.E.(1998).Identification of vegetation state and transition domains in California's hardwood rangelands, University of California.

[112] Gosain,A., Dahiya,S.(2016). Performance analysis of various fuzzy clustering algorithms:A Reivew,*In proceeding of 7th international conference on communication ,computing and virtualization*.Elsevier.

[113] Sajana,T.,Rani,S., Narayana,K.V.(2016). A Survey on Clustering Techniques for Big Data mining. *Indian Journal of Science and Technology*.9 (2016).

[114] Mehta,N., Dang,S.(2011) .A Review of Clustering Techniques in various Applications for effective data mining. *International Journal of Research in Engineering & Applied Science*.

[115] Jain,A.K.,Murty, M.N., Flynn,P.J.(1999). Data clustering: A review, *ACM Comput. Survey*,31(1999),264–323.

[116] Andrew,Ng.(2012).Clustering with the K-Means Algorithm, *Machine Learning*.

[117] Hashler,M.(2011).Recommender Lab: A Framework for Developing and Testing Recommendation Algorithms,1-37.

[118] Jazayeriy,H.,Mohamaddi,S.,Shamshirband,S.(2018).A Fast Recommender System for Cold User using Categorized items.*Mathematical and Computer Applications*,23(1).

[119] Koper,R.,Olivier,B.(2004). Representing the learning design of units of learning. *Educational Technology & Society*,7(3),97–111.

[120] Resnick, P., Varian, H. (1997). Recommender systems. *Communication of ACM*, 40,56–58.

[121] Nunez –Valdez, E.R., Cueva Lovelle, J.M., Sanjuan Martinez, O., Garcia Diaz, V., Ordonez de Pablos, P.,Montenegro Marin, C.A. (2012) Implicit feedback techniques on recommender systems applied to electronic books. *Comput Hum Behav*, 28(4):1186–1193.

[122] Tang, T.Y., McCalla, G. (2005). Smart recommendation for an evolving e-learning system: architecture and experiment. *International journal of e-Learning*, 4(1),105–129.

[123] Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.(1997). Group Lens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3), 77–87.

[124] Hill, W., Stead, L., Rosenstein, M., Furnas, G.(1995). Recommending and evaluating choices in a virtual community of use. In: CHI ’95: *Proc. of the SIGCHI Conf. on HumanFactors in Computing Systems*,194–201. ACM Press/Addison-WesleyPublishing Co., New York,NY, USA.

[125] Shardanand,U., Maes, P. (1995).Social information filtering:algorithms for automating “word of mouth”. *In: Proceedings of ACM CHI’95 conference*, 210–217.

[126] Breese, J.S.,Heckerman,D.,Kadie, C.(1998).Empirical analysis of predictive algorithmsfor collaborative filtering,*In: Proc. of the 14th Annual Conference on Uncertainty in Artificial Intelligence*,43–52.

[127] Hofmann, T.(2003).Collaborative filtering via Gaussian probabilistic latent semantic analysis,*In: SIGIR ’03: Proc. of the 26th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, ACM, New York,USA.

[128] Grcar, M., Fortuna, B., Mladenic, D., Grobelnik, M.(2006). k-NN versus SVM in the collaborative filtering framework. *Data Science and Classification*,251–260.

[129] Koren, Y.(2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. *In: KDD'08: Proceeding of the 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, USA.

[130] Iaquinta, L., & Semeraro, G. (2011). Lightweight Approach to the Cold-Start Problem in the Video Lecture Recommendation,*In Proc of ECML-PKDD Discovery Challenge Workshop.*

[131] Zhang, Z., Liu, C., Zhang, Y.-C., & Zhou, T.(2010). Solving the cold start problem in recommender systems with social tags. *EPL*, 92(2).

[132] Romero, C., Ventura, S., & García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1), 368–384.

[133] Degemmis, M., Lops, P., & Semeraro, G.(2007). A content-collaborative recommender that exploits wordnetbased user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3), 217–255.

[134] Sarwar, B., Karypis,G., Konstan,J., & Riedl,J.(2001).Item-based Collaborative Filtering Recommendation Algorithms. *In Proc. of the 10th International WWW Conference*.

[135] Rafter, R. (2010). Evaluation and Conversation in Collaborative Filtering [PhD Thesis]. University College Dublin, College of Engineering Mathematical and Physical Sciences.

[136] Portugal, I., Alencar, P., & Cowan, D. (2015). The use of machine learning algorithms in recommender systems: A systematic review. *In Proceedings of International conference on Hybrid Artificial Intelligence Systems.*

[137] Rangra,K.,Bansal,K.L.(2014).Comparative study of data mining tools.*International Journal of Advanced Research in Computer Science and Software Engineering*,4(6).

[138] Srivastava, J., Cooley, R., Deshpande, M., & Tan, P. N. (2000). Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, 1(2).

[139] Burke, R. (2002). Hybrid recommender systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 2(4), 331–370.

[140] Burke, R. (2007). Hybrid web recommender systems. *In The Adaptive Web*,377–408.

[141] Alomran,H.(2014).Text Mining Based Semantic Web Architecture for e-learning system.*International Journal of Machine Learning and Computing*,4(4), 333-338.

[142] Cooley, R., Mobasher, B., and Srivastava, J. (1999). Data preparation for mining world wide web browsing. *Journal of Knowledge Information Systems*, 1(1), 5-32.

[143] Myszkorowski,K.,Zakrzewska,D.(2013). Using fuzzy logic for recommending groups in e-learning systems, *In Computational Collective Intelligence. Technologies and Applications*,671–680.

[144] Alvarez, S. A., Ruiz, C., Kawato, T., & Kogel, W. (2011). Neural expert networks for faster combined collaborative and content-based recommendation. *Journal of Computational Methods in Sciences and Engineering*, 11(4),161–172.

[145] Hariri, N., Castro-Herrera, C., Mirakhorli, M., Cleland-Huang, J., & Mobasher, B. (2013). Supporting domain analysis through mining and recommending features from online product listings. *IEEE Transactions on Software Engineering*, 39(12), 1736–1752.

[146] Sunil.L.,Saini,D.K.(2013).Design of Recommender System for Web Based Learning. *In Proceedings of the World Congress on Engineering*,1(2013),London,U.K.

[147] Liu.W.,Gao,L.(2014). Recommendation System based on Fuzzy Cognitive Map.*Journal of Multimedia*,9(7).

[148] Maâtallah,M., Seridi,H.(2012).Enhanced collaborative filtering to recommender systems of technology enhanced learning," *In ICWIT*.

[149] Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., & Riedl, J.(1998) .Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system.*In Proceedings of the 1998 ACM conference on computer supported cooperative work.*

[150] Chen, Y., Wu, C., Xie, M., & Guo, X. (2011). Solving the Sparsity Problem in Recom - mender System Using Association Retrieval. *Journal of Computers*, 6(9).

[151] Owoc, M., & Weichbroth, P. (2012). Validation Model for Discovering Web User Navigation Patterns. *In International workshop on Artificial intelligence for knowledge Management.*

[152] Popescul, A., Ungar, L. H., Pennock, D. M., & Lawrence, S. (2001). Probabilistic models for unified collaborative and content - based recommendation in sparse-data environments. *In Proc of the 17th conference on uncertainty in artificial intelligence*, 437-444.

[153] Kim, H. N., Alkhaldi, A., Saddik, A. E., & Jo, G. S. (2011). Collaborative user modeling with user generated tags for social recommender Systems. *Expert Systems with Applications*, 38(7), 8488–8496.

[154] Silva, N. B., Tsang, I. R., Cavalcanti, G. D. C., & Tsang, I. J. (2010). A graph based friend recommendation system using Genetic Algorithm.*In Proc of IEEE Congress on Evolutionary computation.*

[155] Salton, G. (1989). Automatic Text Processing. *Addison-Wesley.*

[156] Powell, M. J. D. (1981). Approximation Theory and Methods. Cambridge University Press.

[157] Shah,K.,Gadhe,J.(2011).Semantic Web Services for E-Learning: Engineering and Technology Domain.*International Journal of Computer Theory and Engineering*,3(6).

[158] Schafer,J.B., Konstan,J., & Riedl.J.(1999). Recommender systems in e-commerce. *In EC '99: Proceedings of the 1st ACM conference onElectronic commerce*, New York, NY, USA.

[159] Gehringer, E. F. (2001). Electronic peer-review and peer grading in computer-science courses. *In Proceedings of the 32nd SIGCSE technical symposium on computer science education.*

[160] Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2006) Toward legal argument instruction with graph grammars and collaborative filtering techniques. *In Proceedings of the 8th international conference on intelligent tutoring systems.*

[161] Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. (2006). Data preprocessing for supervised learning. *International Journal of Computer Science*, 1, 111–117.

[162] Mathes, A. (2004). Folksonomies-cooperative classification and communication through shared metadata. *Journal of Computer Mediated Communication*, 47(10).

[163] Shirude,S.B., Kolhe,S.R.(2016).Machine Learning Using K-Nearest Neighbor for Library Resources classification in Agent-Based Library Recommender System. *In Advances in Computing Applications.*

[164] Aggarwal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. *In Proceedings of the 20th International Conference on Very Large Data Bases.*

[165] Takacs, G., Pilaszy, I., Nemeth, B., & Tikk, D. (2008). Investigation of various matrix factorization. Methods for large recommender systems. *In Proc. of the 2nd KDD Workshop on Large Scale Recommender Systems and the Netflix Prize Competition.*

[166] Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems*, 23(1), 103–145.

[167] Yeung, A. A. (2010). Matrix Factorization: A Simple Tutorial and Implementation in Python [Web log post]. Retrieved January 5, 2018,from http://www.quuxlabs.com/blog /2010/09/matrix-factorization-a-simple-tutoria land-implementation-in-python/.

[168] Kohavi, R., Crook, T & Longbotham,R.(2009). Online Experimentation at Microsoft ,*Third Workshop on Data Mining Case Studies and Practice Prize*. http://expplatform.com/expMicrosoft.aspx.

[169] Deborah,L.J., Sathiyaseelan,R., Audithan,S., and Vijayakumar,P.(2015). Fuzzy-logic based learning style prediction in e-learning using web interface information. In Proc of Engineering Science, 40(2), 379–394.

[170] Cornelis,C., Lu,J., Guo,X., Zhang.G.(2007). One-and-only item recommendation with fuzzy logic techniques. *INFORMATION SCIENCES*,177(2007), 4906-4921.

[171] Ding,L.(2014).E-Learning Resource Recommendation based on Fuzzy Sets. *Applied Mechanics and Materials*,513-517(2014),2186-2189.

[172] Chaofeng,L.(2006). Research and Development of Data Preprocessing in Web Usage Mining. *International Conference on Management Science and Engineering*.

[173] Tiemann, M., & Pauws, S. (2007). Towards ensemble learning for hybrid music recommendation. *In RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems.*

[174] Kotsiantis, S. B., Pierrakeas, C. J., & Pintelas, P. E. (2003). Preventing Student Dropout in Distance Learning Using Machine Learning Techniques. *In Knowledge-Based Intelligent Information and Engineering Systems*, 2774.

[175] Shirgave,S.,Kulkarni,P.,Borges,J.(2014).Semantically Enriched Web Usage Mining for Personalization. *International Journal of Information and Information Engineering.World Academy of Science Engineering and Technology*,8(1).

[176] Peng,S., Cheng,Q.(2009). Research on Data Preprocessing process in the Web Log Mining. *The 1st international conference on Information Science and Engineering (ICISE).*

[177] Rawat,B.,Dwivedi,S.K.(2018). State of the Art Recommendation Approaches: their Issues and Future Research Direction in E-learning A Survey. *International journal of advanced pervasive and ubiquitous computing*,10(1).

[178] Mustapasa, O., Karahoca, D., Karahoca, A., Yucel, A., & Uzunboylu, H. (2010). Implementation of semantic web mining on E-learning. *In Proceedings of social and behavioral science.*

[179] Altalhi,A.H., Luna,J.M.,Ventura,S.(2017). Evaluation and comparison of open source software suites for data mining and knowledge discovery. *WIREs Data Mining Knowl Discovery*,7(3).

[180] Drachsler, H., Hummel, H.G.K., Koper, R. (2008). Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model. *International journal of learning Technology*, 3(4),404–423.

[181] Vesin, B.Ivanovic,M.(2004):Modern educational tools. *In Proceedings of 16th conference on Applied Mathematics, Budva, Montenegro.*

[182] Bhowmick,P.,Sarkar,S.,Basu,A.(2010).Ontology based user modeling for personalized Information access.*International Journal of Computer Science and Applications*,7(1),1-22.

[183] Begam,M.F.,Ganapathy,G.(2013).Adaptive Learning Management System Using Semantic Web Technologies. *International Journal on Soft Computing*,4(1).

[184] Ercan,T.(2011). Benefits of semantic approach in the e-Learning environment. *In proceedings of Social and Behavioral Sciences.*

[185] Graudina,V.,Grundspenkis,J.(2005).The Role of Ontologies in Agent Based simulation Simulation of Intelligent Tutoring System.*In proceeding of the 19th European Conference on Modelling and Simulation.*

[186] Holink,L.,Mika,P.,Blanco,R.(2013).Web Usage Mining with Semantic Analysis. *In Proc of the 22nd International Conference on World Wide Web*. ACM, 561-570.

[187] Dai,H., Mobasher,B.(2005).Integrating semantic knowledge with web usage mining for personalization. *in Web Mining: Applications and Techniques*, A. Scime, Ed. Hershey, PA, USA: IGI Global, 2005, pp. 205–232.

[188] Khirbi,M.,Jemni,M.,Nasraoui,O.(2009). Automatic Recommendation for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval ,*Educational Technology and Society*,12(4),30-42.

[189] Markellou,P.,Mousourouli,I.,Spiros,S.,Tsakalidis,A.(2005).Using semantic web mining technologies for Personalized E-learning Experiences.In Proceedings of the web-based education,461-826.

[190] Frutos-Morales,F.,Sanchez-Vera,M.M.,Castellanos-Nieves,D., Esteban-Gil,A., Cruz-Corona,C., Prendes-Espinosa,M.P., Fernández-Breis,J.T.(2010).An Extension of OELE Platform for Generating Semantic Feedback for Students and Teachers. *In proceedings of Social and Behavioral Sciences.*

[191] Mustapasa,O., Karahoca,A., Karahoca,D., Uzunboylu, H.(2011). "Hello World", Web Mining for E-learning. *In proceedings of computer science*.

[192] Mori,J.,Matsuo,Y.,Hashida,K.,Ishizuka,M.(2015).Web Mining Approach for a User – Centered Semantic Web. *In: Proc. Of International Workshop on User Aspects on the Semantic Web in 2nd European Semantic Web Conf. (ESWC 2005*), Heraklion, Greek, 177–187.

[193] Mobasher, B., Jin, X., and Zhou, Y.(2004). Semantically Enhanced Collaborative Filtering on the Web'. *In B. Berendt, et al. (eds.): Web Mining: From Web to Semantic Web*. LNAI,3209, Springer.

[194] Ramesh,C.,Rao,K.V.C.(2017).Ontology Based Web Usage Mining Model. *International Conference on Inventive Communication and Computational Technology*.IEEE.

[195] Zhang, D., Zhao, J. L., Zhou, L., & Nunamaker, J. (2004). Can e-learning replace traditional classroom learning— evidence and implication of the evolving e-learning technology. *Communications of the ACM*, 47(5), 75–79.

[196] Shishehchi,S.,Banihashem,S.Y.,Zin,N.A.M.(2010).A Proposed semantic recommendation system for e-learning:A Rule and Ontology based e-learning recommendation system. *Information Technology (ITSim), 2010 International Symposium*.IEEE.

[197] Ghaleb,F.,Daoud,S.,Hasana,A., ALJa'am,J.M., El-Seoud,S.A., El-Sofany,H.(2006). E-learning model based on semantic web technology. *International journal of computing and information sciences*,4(2),63-71.

[198] Garcia,E., Romero,C., Ventura,S., & Castro, C.D. (2009). An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering, User Model. User-Adapted Interaction: *Journal of Personalization Recommmendation*., 19,99–132.

[199] http://www.w3.org/RDF/

[200] Xiang-Wei,L., Yian-Fang,Q.(2010). A data preprocessing algorithm for classification model based on rough sets, *International conference on solid state devices and material science.*

[201] Wu,D., Lu,J., & Zhang,G.(2015). A Fuzzy Tree Matching Based Personalized E-Learning Recommender System. *IEEE Transactions on Fuzzy Systems*, 23(6), 2412–2426.

[202] Jevsikova,T.,Berniukevicius,A.,Kurilovas.(2017).Application of Resource Description Framework to Personalized Learning:Systematic Review and Methodology.*Informatics and Education*,16(1),61-82.

[203] Aydin,C.C.,Tirkes,G.(2010). Open source learning management system in distance learning.*The Turkish online journal of educational technology*,9(2).

[204] Langseth,H., Nielsen, T.D.(2015). Scalable learning of probabilistic latent models for collaborative filtering, *Decision Support Systems*, 74 (2015), 1-11.

[205] Goldberg,D., Nichols,D., Oki,B., & Terry,D.(1992). Using collaborative filtering to weave an information tapestry. *Communications of the Association of Computing Machinery*, 35(12),61–70.

[206] Konstan, J.A., Riedl,J.,Borchers,AI.,Herlocker,J.L.(1998).Recommender Systems: A GroupLens Perspective. *In Recommender Systems. Papers from 1998 Workshop. Technical Report WS-98-08*. AAAI Press.

[207] Sneha,Y.S.,Mahadevan,G.,Prakash,M.(2011).An online recommendation system based on web usage mining and semantic web using LCS algorithm. *In proceedings of 3rd International conference on Electronics Computer Technology*, 223-226.

[208] Stone, M.(1974). Cross-Validatory Choice and Assessment of Statistical Predictions (with Discussion). Journal *of the Royal Statistical Society*,36(2),111-147.

[209] Goldberg,K., Roeder,T., Gupta,D., & Perkins,C.(2001).Eigentaste: A constant time collaborative filtering algorithm. *Information. Retrieval*. 4(2),133-151.

[210] Perry, J.W., Kent, A., Berry,M.M.(1955). Machine literature searching X. Machine; language factors underlying its design and development.*American Documentation*,6(4).

[211] Beitzel., Steven,M. (2006). On Understanding and Classifying Web Queries (Ph.D. thesis).IIT. CiteSeerX 10.1.1.127.634.

[212] Blair,D.C.(1979).Information retrieval, 2nd ed. c.j. van rijsbergen. london. *Journal of the American Society for Information Science*,30(6),374−375.

[213] Arsan,T., Koksal,E.,Bozkus,Z.(2016).comparison of collaborative filtering algorithms with various similarity measures for movie recommendation. *International Journal of computer science engineering and application*, 6(3).

[214] Alluhaidan, A. (2013). Recommender System Using Collaborative Filtering Algorithm, *Technical Library: School of Computing and Information Systems*.

[215] Bydzovska, H.(2006). Course Enrolment Recommender System. *In Proceedings of the 9th International Conference on Educational Data Mining*,312-317.

[216] OMahony, M. P., Smyth, B. (2007). A Recommender System for On-line Course Enrolment: An Initial Study, *ACM Conference on Recommender System*, 973–978.

[217] Miler,E.(1998).D-LIB MAGAZINE. An Introduction to the Resource Description Framework. United States: *D-lib Magazine*. Maio.

[218] http://www.w3.org/DesignIssues/Notation3.

[219] http://www.w3.org/Submission/SWRL/.

[220] http://www.racer-systems.com/products/racerpro/index.phtml.

[221] http://jena.sourceforge.net/inference.

[222] http://mandarax.sourceforge.net/.

[223] Tong,Q.,Zhang,F.,Cheng,J.(2014).Construction of RDF(S) from UML Class Diagram.*Journal of Computing and Information Technology*,237-250.

[224] Rawat, B., Dwivedi,S.K(2018). Discovering Learners' characteristics through cluster analysis for recommendation of course in E-learning Environment. *International Journal of Information and Communication Technologies Education*.IGI,15(4).

[225] Alhijawi,B.,Obeid,N.,Awajan,A.,Tedmori,S.(2018).Improving Collaborative Filtering Recommender Systems Using Semantic Information.*In Proc of 9th International Conference on Information and Communication Systems*.IEEE,127-132.

[226] Tarus,J.,Niu,Z.,Khadidja,B.(2017).E-Learning Recommender System Based on Collaborative Filtering and Ontology.*International Journal of Computer and Information Engineering*.World Academy of Science Engineering and Technology,11(2).

[227] Gurawardana,A.,Shani,G.(2009).A Survey of Evaluation Metrics of Recommendation Tasks.*Journal of Machine Learning Research*,10(2009).

[228] Shani, G., Gunawardana, A. (2011). Evaluating recommendation systems.Recommender. System. Handbook, 257–298.

[229] Bizonova, Z, Ranc, D.(2007). Courseware Material Reuse via Model Driven LMS Platform Integration, *CBLIS conference*.

[230] Lops, P., Gemmis, M., & Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. In Recommender Systems Handbook. New York: Springer.

[231] http://www.examulator.com/er/

Appendix I

List of Publications

- Rawat,B.,Dwivedi,S.K.(2018).Discovering Learners' characteristics through cluster analysis for Recommendations of Courses in E-learning Environment. *International Journal of Information Communication and Technologies Education.*IGI,15(1). ESCI Indexing).
- Rawat,B.,Dwivedi,S.K(2018).State of the Art Recommendation Approaches their issues and Future Research Direction in E-Learning: A Survey. *International Journal of Advanced and Ubiquitous computing*.IGI,10(1).
- Rawat,B., Dwivedi,S.K.(2018). Selecting Appropriate Metrics for Evaluations of Recommender Systems. *International Journal of Information Technology and Computer Science*, MECS (Accepted).
- Rawat,B.,Dwivedi,S.K.(2018). Enriching User Item Rating Matrix with Resource Description Framework for Improving the Accuracy of Recommendation in E-Learning Environment. *International Journal on Semantic Web and Information System*,IGI.(Communicated)[SCIE Indexing].
- Dwivedi,K.S., Rawat,B.(2018).Analyzing the Performance of Various Clustering Algorithms. *International Journal of Modern Education and Computer Science*(IJMECS),MECS.
- Dwivedi, K.S., Rawat, B.(2015).A Review Paper on Data Preprocessing: A critical phase in web usage mining process. *In proceedings of international conference on Green computing and internet of things*. IEEE Xplore,506-510.
- Dwivedi,K.S., Rawat,B.(2016).A Review on Improving Recommendation Quality by using Relevant Contextual Information. *In proceedings of international conference on International Conference on Computing for Sustainable Global Development*. IEEE Xplore,244-248.
- Dwivedi,K.S., Rawat,B.(2016).Issues and Requirements for Successful Integration of Semantic Knowledge in Web Usage Mining for Effective Personalization. *In Proceedings of International Conference on SmartCom, Vol.628*.Springer, 97-103.

- Rawat,B.,Dwivedi,K.S.(2017).An Architecture for Recommendation of Courses in E-learning System.*International Journal of Information Technology and Computer Science*,(IJITCS). MECS, 9(4):39-47.

Appendix II

Learners' Usage Dataset

ID	Q1	Q2	Q3	Q4	QD	NP	NF	QM	AD	TH	MT	MF	MR	MC	TQ	TF	AR	NA	RV	QG	CT
L1	7.71	10	10	6.55	4	4	0	34.26	9	8	40	60	80	70	21	21	71	18	50	9.24	85.6
L2	6.95	9.09	8	6.67	4	4	0	30.71	7	7	30	50	70	65	20	20	29	1	20	8.01	76.8
L3	5.63	7.5	6	4.55	4	3	1	23.68	9	8	50	70	100	85	15	15	65	16	50	6.38	59.2
L4	7.86	8.18	10	5	4	4	0	31.04	10	15	60	70	120	115	21	21	95	19	50	8.68	77.6
L5	4	3	2	4	4	0	4	13	1	1	5	4	1	4	4	6	8	1	10	3	30.1
L6	7.32	9.09	9	7.55	4	4	0	32.96	7	10	50	60	80	65	18	18	22	5	12	8.47	79.3
L7	6.65	8.18	10	6.33	4	4	0	31.16	10	18	40	50	70	60	17	17	27	5	15	8.28	77.9
L8	8.63	6.5	5.4	6.9	1	1	3	27.43	2	4	10	25	40	35	7	7	104	11	60	6.84	86.3
L9	5.68	6.36	7	4.55	4	3	1	23.59	8	12	30	45	60	55	15	15	46	2	20	6.35	59
L10	3.73	9.55	10	7.88	4	3	1	31.16	9	16	35	50	70	65	14	14	50	24	30	7.76	72.3
L11	2.5	1.5	3.5	2.5	0	0	4	10	1	1	2	3	4	5	1	3	28	2	15	2.5	25.3
L12	8.16	8.18	6	6.64	4	4	0	28.98	10	14	50	60	80	55	22	22	112	8	80	7.45	72.5
L13	2.1	3.1	1.5	2.4	0	0	4	9.1	1	1	2	3	5	4	2	3	20	8	25	2.23	14.5
L14	1.5	2.5	3.1	2.9	0	0	4	10	2	1	4	10	40	30	2	3	34	4	20	2.37	13.5
L15	6.66	7.27	10	6.36	4	4	0	30.29	9	10	40	60	80	75	25	25	97	5	60	7.98	75.7
L16	8.29	9.55	9	6.48	4	4	0	33.32	10	12	45	50	70	65	22	22	87	4	50	8.95	83.3
L17	7.57	7.5	9	5.64	4	4	0	29.71	9	10	42	57	80	60	23	23	77	5	31	8.02	74.3
L18	4.84	7.27	10	5.45	4	4	0	27.56	9	13	41	50	70	60	24	24	69	8	19	7.37	65.1
L19	7.66	7.27	10	4.73	4	3	1	29.66	10	14	35	50	70	55	21	21	80	7	23	8.31	74.1
L20	7.8	5.23	7	7.33	4	4	0	27.36	7	10	45	60	80	60	22	22	15	3	10	6.68	68.4
L21	4.5	2.1	3.5	2.9	0	0	4	13	1	2	6	5	8	5	4	6	40	4	10	3.37	20.9
L22	4.8	7.73	6	6.48	4	3	1	25.01	7	10	35	50	70	65	18	18	48	25	20	6.18	60
L23	5.68	9.09	9	7.55	4	4	0	31.32	8	12	45	59	80	70	22	22	38	9	20	7.92	78.3
L24	3.91	7.27	9	1.61	4	2	2	21.79	6	8	30	50	70	65	14	14	32	5	10	6.73	54.5
L25	2.27	5.45	10	4.36	4	2	2	22.08	5	6	25	30	50	45	15	15	40	3	20	5.91	55.2
L26	6.66	5.45	9	8.73	4	4	0	29.84	10	15	40	50	60	55	16	16	121	14	65	7.04	74.6
L27	1.25	2.1	1.9	2.7	0	0	4	7.95	0	0	0	0	0	0	0	0	61	4	20	1.75	21.8
L28	3.1	2.5	3.5	2.5	0	0	4	11.6	2	1	6	5	7	9	2	2	35	22	20	3.03	25.9
L29	4.69	6.36	7	8.82	4	3	1	26.87	8	12	35	50	80	65	14	14	76	8	30	6.02	67.2
L30	6.57	8.18	8	7.03	4	4	0	29.78	10	12	45	54	70	60	22	22	76	6	35	7.58	79.6
L31	2.89	1.5	2.1	3.1	0	0	4	9.59	1	1	5	6	4	9	1	6	32	4	20	2.16	19.5
L32	7.21	7.5	6	5.91	4	4	0	26.62	8	14	45	60	80	60	23	23	76	19	30	6.9	66.5
L33	5.35	9.32	8	6.64	4	4	0	29.31	8	13	40	50	70	55	24	24	76	1	24	7.56	73.3
L34	2.98	3.1	2.8	4.33	3	2	2	13.21	5	6	30	50	60	50	10	14	32	18	10	2.96	30.2
L35	7.93	9.09	10	4.91	4	3	1	31.93	7	9	30	40	55	40	18	18	36	19	20	9.01	79.8
L36	3.1	2.73	2.64	2.8	0	0	4	11.27	3	1	2	5	5	2	2	5	4	1	1	2.82	30.2

L37	6.38	7.27	9	7.52	4	4	0	30.17	9	14	42	50	70	65	22	22	100	7	30	7.55	75.4
L38	4.98	5.45	8	-	3	2	2	#REF!	6	10	28	30	50	40	15	15	45	6	20	6.14	61.4
L39	3.1	2.1	2.8	2.1	0	0	4	10.1	3	1	5	10	12	12	2	5	10	12	5	2.67	25.8
L40	5.82	8.64	9	7.76	4	4	0	31.22	9	15	46	50	70	55	25	25	112	4	40	7.82	78
L41	8.02	8.64	6	6.64	4	4	0	29.3	7	12	47	58	80	90	26	26	98	26	40	7.55	73.2
L42	7.93	7.5	9	4.18	4	3	1	28.61	6	9	30	50	80	50	16	16	70	2	30	8.14	71.5
L43	4.06	6.14	8	2.3	4	2	2	20.5	5	6	25	40	70	60	14	14	50	10	25	6.07	51
L44	6.34	5.91	9	5.52	4	4	0	26.77	10	16	46	70	75	50	24	24	120	8	50	7.08	66.9
L45	7.31	9.55	8	9.15	4	4	0	34.01	9	12	40	40	60	60	23	23	110	4	40	8.29	85
L46	4.2	2.8	3.1	2.5	0	0	0	12.6	2	1	10	20	12	20	2	10	12	7	5	3.37	19.5
L47	8.58	8.64	10	8	4	4	0	35.22	9	14	41	50	70	65	21	21	120	6	60	9.07	88
L48	2.7	2.8	2.9	4.1	4	0	4	12.5	4	2	10	20	19	20	4	5	5	5	2	2.8	34.1
L49	6.69	8.18	5	7.91	4	4	0	27.78	10	14	42	50	70	45	22	22	130	10	40	6.62	68.9
L50	4.43	9.55	10	8.42	4	3	1	32.4	8	13	35	50	80	60	16	16	80	7	30	7.99	81
L51	5.55	0.91	8	6.27	4	3	1	20.73	7	12	32	40	60	55	18	18	70	5	35	4.82	51.8
L52	3.1	2.5	2.11	2.3	4	0	4	10.01	3	2	10	20	14	30	2	20	20	4	10	2.57	25.9
L53	3.8	3.1	2.1	2.7	4	0	4	11.7	3	2	40	10	20	30	3	3	20	26	9	3	20.1
L54	5.28	9.32	9	7.91	4	4	0	31.51	10	14	42	60	80	60	21	21	130	8	50	7.87	78.8
L55	7.19	6.36	8	8.61	4	4	0	30.16	9	12	48	50	70	80	24	24	90	4	40	7.18	75.4
L56	7.89	7.5	8	6.64	4	4	0	30.03	10	14	41	70	80	90	23	23	130	1	60	7.8	75.1
L57	7.16	8.41	9	3.61	4	3	1	28.18	7	9	39	40	60	60	18	18	90	13	40	8.19	70.4
L58	5.32	7.73	8	7.73	4	4	0	28.78	9	12	45	50	70	60	22	22	80	2	30	7.02	72
L59	2.5	2.8	5.1	3.6	4	0	4	14	4	3	10	20	30	40	4	6	20	22	10	3.47	34.1
L60	4.63	1.1	2.2	3.5	4	0	4	11.43	3	6	10	30	50	60	5	4	40	7	20	2.64	31.9
L61	6.08	7.27	9	6.85	4	4	0	29.2	9	12	41	50	70	55	23	23	80	6	40	7.45	73
L62	6.39	8.18	9	6.91	4	4	0	30.48	10	14	46	70	80	80	24	24	130	3	50	7.86	76.2
L63	3.76	2.78	6.1	1.8	4	0	4	14.44	4	2	50	20	40	30	4	20	19	19	10	4.21	32.1
L64	5.41	8.18	9	8.45	4	4	0	31.04	9	15	48	60	80	70	26	26	130	1	60	7.53	77.6
L65	3.5	2.1	4.1	2.2	4	0	4	11.9	2	1	50	20	20	30	4	5	20	17	10	3.23	32.9
L66	8.92	5.91	10	7.06	4	4	0	31.89	8	10	47	50	75	60	23	23	90	20	40	8.28	79.7
L67	5.65	5.45	9	8.73	4	4	0	28.83	10	15	42	40	60	55	26	26	135	1	60	6.7	72.1
68	0.66	8.18	10	4.52	4	2	2	23.36	6	10	26	30	50	60	15	15	50	6	30	6.28	60.1
L69	4.4	6.59	5	1.82	4	2	2	17.81	6	8	25	21	40	55	12	12	65	7	35	5.33	44.5
L70	4.56	2.1	2.8	3.33	1	0	4	12.79	2	4	10	30	50	45	0	5	20	14	10	3.15	31.9
L71	7.62	9.55	9	7.58	4	4	0	33.75	10	15	49	70	80	55	21	21	120	2	50	8.72	84.4
L72	6.55	8.41	7	5.97	4	4	0	27.93	9	12	47	60	80	70	23	23	130	26	60	7.32	69.8
L73	4.19	8.64	9	8.79	4	3	1	30.62	7	10	30	50	60	55	21	21	90	2	50	7.28	76.5
L74	5.93	8.41	8	8.97	4	4	0	31.31	10	12	25	40	55	40	24	24	140	9	60	7.45	78.3
L75	3.1	2.5	4.2	1.2	4	0	4	11	5	2	40	20	21	30	5	6	30	10	20	3.27	26
L76	2.22	3.1	2.5	2.1	4	0	4	9.92	5	4	10	20	30	10	5	10	20	4	10	2.61	20.2
L77	2.8	1.97	3.15	2.87	4	0	4	10.79	3	5	10	15	40	20	3	4	40	6	30	2.64	27

L78	7.5	6.36	7	5.97	4	4	0	26.83	10	12	41	50	80	70	20	20	130	7	60	6.95	67.1
L79	3.96	2.1	2.73	2.9	4	2	2	11.69	6	8	32	40	60	40	14	14	60	5	30	2.93	30.2
L80	4.39	6.59	8	2.91	4	2	2	21.89	6	9	25	30	50	50	12	12	70	8	50	6.33	54.7
L81	9.01	9.32	7	6.64	4	4	0	31.97	9	14	45	50	85	80	21	21	130	7	90	8.44	79.9
L82	7.06	7.5	7	6.64	4	4	0	28.2	10	15	49	70	90	70	21	21	140	3	60	7.19	70.5
L83	5.88	4.55	5	8.79	4	4	0	24.22	9	14	47	60	90	85	23	23	130	4	50	5.14	60.5
L84	2.5	1.7	3.33	2.01	0	0	4	9.54	2	1	50	10	30	20	5	5	30	26	15	2.51	30.7
L85	7.27	8.18	8	5.36	4	4	0	28.81	9	14	45	50	70	60	21	21	130	7	60	7.82	72
L86	5.03	7.27	6	5.21	4	4	0	23.51	9	16	42	40	60	50	22	22	30	4	10	6.1	58.8
L87	6.32	4.55	10	6.55	4	3	1	27.42	7	10	40	60	80	60	18	18	90	1	50	6.96	68.5
L88	3.87	3.71	3.1	3.2	4	0	4	13.88	6	4	40	50	20	10	5	4	20	13	10	3.56	20.3
L89	7.89	8.41	9	6.33	4	0	0	31.63	0	0	0	0	0	0	19	0	30	2	15	8.43	79.1
L90	6.53	9.55	9	6.88	4	4	0	31.96	9	14	40	60	80	75	21	21	130	22	60	8.36	79.9
L91	7.63	5.45	6	6.67	4	4	0	25.75	9	15	43	50	65	50	22	22	120	7	50	6.36	64.4
L92	7.32	6.59	10	8.42	4	4	0	32.33	10	15	47	40	60	55	20	20	140	5	80	7.97	80.8
L93	7.35	9.55	9	5.76	4	4	0	31.66	8	13	49	70	85	70	20	20	140	4	70	8.63	79.1
L94	2.29	2.87	2.1	2.3	4	0	4	9.56	5	4	20	50	40	20	4	5	40	19	20	2.42	33.5
L85	5.81	5.45	5	4.73	4	3	1	20.99	8	14	35	60	80	70	23	23	120	1	90	5.42	52.5
L96	6.41	8.64	9	8.42	4	4	0	32.47	10	15	47	50	65	50	20	20	140	17	60	8.02	78.3
L97	4.79	6.14	5	4.48	4	2	2	20.41	5	8	24	30	70	60	18	18	30	20	20	5.31	51
L98	5.37	7.27	9	7.27	4	4	0	28.91	9	14	46	50	70	65	20	20	90	1	60	7.21	72.3
L99	2.5	3.5	2.98	2.78	4	0	4	11.76	4	4	40	20	30	40	2	4	30	6	15	2.99	20.8
L100	8.38	5.45	10	4.55	4	3	1	28.38	8	14	38	60	80	70	18	18	60	5	40	7.94	71

Symbolic Notations

Q1	Quiz-1
Q2	Quiz-2
Q3	Quiz-3
Q4	Quiz-4
Q_D	Number of Quizzes Done
N_P	Number of Quizzes Pass
N_F	Number of Quizzes Fail
Q_M	Quiz Marks(10)
A_D	Number of Assignment Done
T_H	Time Assignment in Hours
M_T	Message sent to Teachers

M_F	Messages sent to Forum
M_R	Messages Read on Forum
M_C	Messages sent on Chat
T_Q	Time Spent on Quizzes in Hrs
N_A	Number of Resources Accessed less than 30 seconds
R_V	Number of Resources Visited
Q_G	Quiz Grade Obtained
C_T	Course Total
T_F	Time spent On Forum
A_R	Access Resources

Appendix III

Learners' Rating Dataset

Learner_id	Course_id	Rating
L41	81	5
L41	82	4
L41	83	5
L41	84	4
L41	85	
L41	86	5
L41	87	4
L41	88	5
L41	89	
L41	90	5
L41	91	4
L41	92	3
L41	93	5
L41	94	
L41	95	3
L41	96	5
L41	97	4
L41	98	
L41	99	5
L41	100	4
L41	101	3
L41	102	5
L41	103	
L41	104	3
L41	105	
L41	106	4
L41	107	3
L41	108	
L41	109	4
L41	110	3
L41	111	
L41	112	4
L41	113	3
L41	114	
L41	115	4

Learner_id	Course_id	Rating
L41	116	3
L41	117	5
L41	118	
L41	119	
L41	120	5
L42	81	4
L42	82	3
L42	83	
L42	84	4
L42	85	3
L42	86	5
L42	87	
L42	88	3
L42	89	5
L42	90	
L42	91	
L42	92	5
L42	93	4
L42	94	3
L42	95	
L42	96	4
L42	97	3
L42	98	
L42	99	4
L42	100	3
L42	101	
L42	102	4
L42	103	3
L42	104	5
L42	105	4
L42	106	
L42	107	5
L42	108	4
L42	109	3
L42	110	5

Learner_id	Course_id	Rating
L21	41	4
L21	42	5
L21	43	4
L21	44	5
L21	45	4
L21	46	
L21	47	4
L21	48	5
L22	41	4
L22	42	5
L22	43	
L22	44	5
L22	45	4
L22	46	5
L22	47	4
L22	48	
L23	41	
L23	42	5
L23	43	4
L23	44	
L23	45	4
L23	46	
L23	47	5
L23	48	
L24	61	
L24	62	4
L24	63	5
L24	64	4
L24	65	4
L24	66	4
L24	67	5
L24	68	5

Learner_id	Course_id	Rating
L25	41	4
L25	42	4
L25	43	5
L25	44	
L25	45	
L25	46	4
L25	47	
L25	48	
L26	41	4
L26	42	5
L26	43	
L26	44	5
L26	45	4
L26	46	
L26	47	4
L27	41	5
L27	42	5
L27	43	
L27	44	4
L27	45	5
L27	46	
L27	47	4
L27	48	
L27	49	5
L28	41	
L28	42	4
L28	43	
L28	44	5
L28	45	
L28	46	
L28	47	5
L28	48	

Learner_id	Course_id	Rating
L1	6	5
L1	7	4
L1	8	5
L1	9	4
L1	10	5
L1	11	
L1	12	5
L1	13	4
L1	14	
L1	15	4
L1	16	5
L2	1	4
L2	2	
L2	3	4
L2	4	4
L2	5	4
L2	6	
L2	7	4
L2	8	4
L2	9	4
L2	10	
L2	11	4
L2	12	5
L2	13	
L2	14	4
L3	31	4
L3	32	5
L3	33	
L3	34	5
L3	35	4
L3	36	
L3	37	
L3	38	
L3	39	4
L3	40	5
L3	21	

Learner_id	Course_id	Rating
L3	22	5
L4	1	5
L4	2	
L4	3	
L4	4	
L4	5	5
L4	6	4
L4	7	5
L4	8	4
L4	9	5
L4	10	
L4	11	
L4	12	
L5	10	5
L5	11	4
L5	12	5
L5	13	4
L5	14	
L5	15	4
L5	16	5
L5	17	
L5	18	
L5	19	4
L5	20	5
L5	21	4
L5	22	5
L6	25	5
L6	26	
L6	27	
L6	28	
L6	29	5
L6	30	4
L6	31	5
L6	32	5
L6	33	5

Appendix IV

List of Abbreviations

Acronyms	Abbreviations
AHP	Analytical Hierarchy Process
ABT	Attribute Based Techniques
ARFF	Attribute Relation File Format
ANN	Artificial Neural Network
CF	Collaborative Filtering
CBF	Content Based Filtering
CSV	Comma Separate Value
HBRS	Hybrid Based Recommender System
K-NN	K-Nearest Neighbor
KBRS	Knowledge Based Recommender System
KDD	Knowledge Discovery in Database
RDF	Resource Description Framework
RDF(S)	Resource Description Framework Schema
SPARQL	SPARQL Protocol and RDF Query Language
SWUM	Semantic Web Usage Mining
SVD	Single Value Decomposition
SWM	Semantic Web Mining
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WUM	Web Usage Mining
WEKA	Waikato Environment for Knowledge Analysis
XML	Extensible Markup Language
EM	Expectation Maximization
HC	Hierarchical Clustering
DC	Divisive Clustering
IBK	Instance Based Classifier
PIM	Platform Independent Model
PSM	Platform Specific Model