

**LEARNING RESOURCES RECOMMENDATION
FRAMEWORK FOR MOODLE BASED ON ANALYSIS OF
MOSTLY ACCESSED RESOURCES BY GOOD
STUDENTS**

Sivakumar Tharsan

128232U



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.digitallibrary.lk
Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

June 2014

**LEARNING RESOURCES RECOMMENDATION
FRAMEWORK FOR MOODLE BASED ON ANALYSIS OF
MOSTLY ACCESSED RESOURCES BY GOOD
STUDENTS**

Sivakumar Tharsan

128232U



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations

Thesis submitted in partial fulfillment of the requirements for the degree Master of
Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

June 2014

DECLARATION

“I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).”

Signature

Date

(Sivakumar Phansan)  University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk
MSc in Computer Science / 128232U

The above candidate has carried out the research for the Masters thesis under my supervision.

Signature

Date

Dr. Shehan Perera (Supervisor)

Department of Computer Science and Engineering

University of Moratuwa.

Learning resources recommendation framework for Moodle based on analysis of mostly accessed resources by good students

ABSTRACT

We live in an e-learning era, where the fast growth of e-learning around the world is inspiring many educational and business institutions. The rapid growth of e-learning has changed the traditional learning behavior and presented a new situation to both lecturers and students. Lecturers find it harder to guide students to select suitable learning materials due to overdosed learning resources. In the other hand learners spend more time on browsing and filtering learning resources to identify the suitable materials, rather than focusing on learning all the materials. Hence, the learning material recommendation is an essential requirement for e-learning system, which would take the e-learning, to the next level.

Though e-learning brings many benefits in comparison with the conventional learning paradigm, with the rapid increase of learning contents on the web, the e-learning lacks proper feedback or the guidance, which is a key in the traditional teaching process to identify the relevant resources. The students who do not have knowledge to find out the most suitable resources, links and references for their studies and the assignments, may waste most of their time on browsing in search of the relevant material, without any guidance. Some of the “good students” may indirectly act as good guides to their fellow friends. The fellow average learners could follow the methods adapted by good students in learning or accessing relevant learning material. They may refer to the mostly accessed learning materials, by the “good learners”. The lack of feedback in e-learning systems could be overcome by evolving an intelligent feedback system that would recommend the heavily accessed resources by the good students to the average students and that is the aim of this project.

The research collected the Moodle log data of the courses that contain a rich set of electronic course contents from Department of Computer Science and Engineering, University of Moratuwa. This audited data of the collected courses have been used to construct “student classification” and “resource recommendation” models. The former classifies the students as “Good students” and “Average Students” and the latter recommends the “mostly accessed resources” by good students.

The results show that the resources that are mostly accessed by good students are more probable to be recommendable resources. The learning resource recommendation framework helps the average students who fail to choose the relevant materials for their studies from the heaps of learning resources. By making the website interactive, the communication between faculty and students could be made more effective, hence that would promote active learning.

Key words: e-learning, recommendation framework, access pattern, learning analytics, Moodle

ACKNOWLEDGEMENT

I take pleasure to acknowledge the active support and valuable advice provided by my project supervisor Dr. Shehan Perera. The constant guidances and frequent reviews had enabled the successful completion of this research project. His continuous support was available via meetings, email and telephone communications throughout the project duration. Whenever I experienced constraints, this gentleman had the generosity in providing the necessary advice and suggestions that helped me a lot in completing this task.

Also I am glad to express my sincere gratitude to Dr. Shantha Fernando for granting the access to the Moodle backups which were used for data analytics and validate the research models. In addition to that that Mr. Pradeep Manoj, the course assistant was much cordial with regard to obtaining the necessary facts and figures through the course backups and final grades. Last but not least I thank to all who helped me in numerous ways in order to make this project a success.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

TABLE OF CONTENTS

DECLARATION	I
ABSTRACT	II
ACKNOWLEDGEMENT	III
TABLE OF CONTENTS	IV
LIST OF FIGURES	VI
LIST OF TABLES	VIII
LIST OF EQUATIONS	IX
LIST OF ABBREVIATIONS	X
LIST OF APPENDICES	X
1 INTRODUCTION	1
1.1 MOTIVATION FOR THE RESEARCH	1
1.2 RESEARCH OBJECTIVES	1
1.3 RESEARCH DESIGN	2
1.3.1 The scope of the research	2
1.3.2 The research method	2
1.3.3 The data collection and analysis	3
1.4 THESIS OUTLINE	3
2 LITERATURE REVIEW	4
2.1 E-LEARNING SYSTEMS	4
2.2 THE DATA MINING IN E-LEARNING	5
2.3 PREVIOUS RESEARCHES IN LEARNING ANALYTICS	6
2.4 LEARNING RESOURCE RECOMMENDATIONS	7
2.4.1 Recommendation Based on Good Learners' Rating	7
2.4.2 Recommendation Based on Good learners' Access Pattern	10
2.4.3 Recommendation Based on Collaborative Filtering	10
2.4.4 Recommendation Based on Content Filtering	11
3 METHODOLOGICAL FRAMEWORK	14
3.1 COLLECT DATA	16
3.1.1 Remove the user identity from the course data	19
3.1.2 Extraction of moodle data and store it in the database	23
3.1.3 Extraction of moodle resource data	29
3.2 TRANSFORM DATA	31
3.2.1 Dropout students	32
3.2.2 The Computed attributes of student classification model	33
3.2.3 Update the good students in the student table	60

3.2.4	The computed attributes of resource recommendation model	61
3.3	REDUCE DATA.....	69
3.3.1	RapidMiner.....	70
3.3.2	The reason for RapidMiner	70
3.3.3	Data reduction for student classification model	71
3.4	PARTITION DATA.....	74
3.5	BUILDING MODELS	75
3.5.1	Why classification.....	75
3.5.2	Student classification model	79
3.5.3	The resource recommendation model.....	91
3.6	EVALUATE AND CHOOSE MODELS	97
3.6.1	The performance factors used to evaluate the models	98
3.6.2	Data quality issues.....	99
3.6.3	Evaluate student classification model	99
3.6.4	Evaluate resource recommendation model.....	103
4	ANALYSIS OF THE RESULTS	106
4.1	THE ANALYSIS OF STUDENT CLASSIFICATION MODEL	106
4.1.1	The accuracy analysis of student classification model.....	106
4.1.2	RMSE analysis of student classification model.....	111
4.1.3	The best student classification model.....	112
4.2	THE ANALYSIS OF RESOURCE RECOMMENDATION MODEL.....	114
4.2.1	The accuracy of resource recommendation model	114
4.2.2	The RMSE analysis of resource recommendation model.....	117
4.2.3	The best resource recommendation model.....	118
5	CONCLUSION AND RECOMMENDATION	121
5.1	DISCUSSION.....	121
5.1.1	The analytical models.....	121
5.1.2	The student model	123
5.1.3	The resource recommendation model.....	123
5.2	RECOMMENDATION	124
5.3	LIMITATIONS OF THE RESEARCH	124
5.4	FUTURE RESEARCH	125
6	REFERENCES	126
	APPENDIX A: PROGRAMS.....	130
	APPENDIX B: RESULTS.....	132

LIST OF FIGURES

Figure 2-1	Rating based recommendation ([4]).....	9
Figure 2-2	PLRS Framework ([2]).....	12
Figure 3-1	High level design of proposed recommendation system.....	14
Figure 3-2	The modeling process	16
Figure 3-3	Folder structure of an archived course.....	18
Figure 3-4	The intermediate steps of data set preparation	19
Figure 3-5	Anonymize the students' private data to lessen confidential leakage	21
Figure 3-6	Solution for encoding issues in jdom	22
Figure 3-7	Grab data from moodle.xml and transfer it to the database	24
Figure 3-8	Insert the grabbed data into the database.....	25
Figure 3-9	Sets up the connection to the SQL server.....	26
Figure 3-10	Get the connection object and executes the query string.....	26
Figure 3-11	SQL server error in Windows authentication mode.....	27
Figure 3-12	The code snippet for the data extraction of Moodle resources.....	30
Figure 3-13	Distinguishes the dropped students.....	33
Figure 3-14	Eliminates the students with NULL grades	35
Figure 3-15	Computes the Average Ratio of each student.....	36
Figure 3-16	Collects all of the opened discussions of each course into a temp table	39
Figure 3-17	Assess the percentage of open discussions of each student.....	40
Figure 3-18	Picks all of the replies by the student per course.....	41
Figure 3-19	Updates the percentage of open discussion of student.....	42
Figure 3-20	The percentage of wiki entries per student	44
Figure 3-21	Quantify the percentage of assignment of completion of each student.....	46
Figure 3-22	Assess percentage of visit of course	49
Figure 3-23	Accumulates the total number of resources exist in each course	52
Figure 3-24	Quantify the percentage of resource access.....	52
Figure 3-25	The number of late accesses by a student is getting composited.....	54
Figure 3-26	Acquire the percentage of last access of a student.....	55
Figure 3-27	The variation of skewness (source: www.pertrac.com)	57
Figure 3-28	The skewness of access of some of the students selected for this research..	58
Figure 3-29	The selection of StudentResourceLogPivot table	59
Figure 3-30	Update the skewness of student in the Students table	60
Figure 3-31	Classify the students based on the rule learnt by the best model	61
Figure 3-32	The percentage of access of a single resource.....	63
Figure 3-33	Percentage of access of a resource by good students.....	64
Figure 3-34	The position of resource in week	66
Figure 3-35	The position of section in the course	67
Figure 3-36	Percentage of good students accessed a resource.....	68

Figure 3-37	The outliers removed from Student data set.....	72
Figure 3-38	Segregate the data set randomly	81
Figure 3-39	Label good and average students	82
Figure 3-40	Chooses the Id and the label of input data.....	83
Figure 3-41	A classification process using RapidMiner	84
Figure 3-42	The confusion matrix to obtain the accuracy of the model.....	85
Figure 3-43	The classification algorithms, the core of validation operator.....	88
Figure 3-44	Prepare Training data for resource model	92
Figure 3-45	Resource classification process.....	95
Figure 4-1	Accuracy variation of selected algorithms.....	107
Figure 4-2	Accuracy comparison of data splitting – neural networks.....	108
Figure 4-3	Accuracy comparison of data splitting – decision tree	108
Figure 4-4	The accuracy variation over the outlier removal.....	109
Figure 4-5	The accuracy varies with the number of class labels	110
Figure 4-6	The variation of RMSE with algorithms	111
Figure 4-7	The RMSE variation across different data set	112
Figure 4-8	The text view of the selected student classification model.....	113
Figure 4-9	Tree view of selected student classification model.....	114
Figure 4-10	The accuracy of the resource models for different algorithms	115
Figure 4-11	The accuracy varies with different data splitting point	116
Figure 4-12	The accuracy declines with the change of number of class labels	117
Figure 4-13	The RMSE of student model varies with manual and random data	118
Figure 4-14	The text view of selected resource recommendation model	119
Figure 4-15	A tree view of portion of resource recommendation model	120

LIST OF TABLES

Table 3-1	List of Courses selected for the research.....	17
Table 3-2	List of tables in the Moodle database filled by moodle.xml.....	27
Table 3-3	A list of tables which were derived from tables in Table 3.2.....	31
Table 3-4	The mean of the averages obtained in all assignments for each course	36
Table 3-5	Number of open discussions per course.....	39
Table 3-6	The number of replies posted per course	41
Table 3-7	Number of wiki entries to each course	43
Table 3-8	The number of assignments given in every course.....	46
Table 3-9	Some of the students' resource access volume.....	57
Table 3-10	The boundary of each Student label.....	90
Table 3-11	The percentage of access of a resource by good students.....	96
Table 3-12	The accuracies of student models with manual data fragmentation for 2 labels	100
Table 3-13	The accuracies of student models with random data fragmentation for two class labels	102
Table 3-14	The RMSEs of student models with manual data fragmentation for 2 class labels.....	103
Table 3-15	The accuracies of resource models with manual split for two class labels.	104
Table 3-16	The accuracies of resource models with random split for two class labels	104
Table 3-17	The RMSEs of resource models with random data fragmentation for two class labels.....	105
Table 4-1	The number of highly accessed resources at each different splitting point	115



LIST OF EQUATIONS

Equation 2-1	Average rating of learning material.....	7
Equation 2-2	The weight of a document j respective to the term i	8
Equation 3-1	Ratio of average marks (Average ratio).....	35
Equation 3-2	The percentage of open discussion.....	38
Equation 3-3	The percentage of replies to the forum discussion	41
Equation 3-4	The percentage of wiki entries	44
Equation 3-5	The percentage of assignment completion.....	45
Equation 3-6	The percentage of visits of student.....	48
Equation 3-7	The percentage of resource access.....	51
Equation 3-8	The percentage of last minute access.....	53
Equation 3-9	Microsoft Excel's skew function.....	59
Equation 3-10	The percentage of access of a resource	63
Equation 3-11	The percentage of access of a resource by good students.....	65
Equation 3-12	The percentage of access of a resource by average students.....	65
Equation 3-13	The percentage of access by good students.....	68
Equation 3-14	The accuracy of prediction	98



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF ABBREVIATIONS

Abbreviation	Description
MOODLE	Modular Object Oriented Dynamic Learning Environment
LMS	Learning Management System
PLRS	Personalized Learning Recommendation System
PAWS	Perdue Early Warning System
CMS	Course Management System
VLE	Virtual Learning Environment
SVM	Support Vector Machines
K-NN	K-Nearest Neighbour
SQL	Structured Query Language
CSV	Comma Separated Values
XML	Extensive Markup Language



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF APPENDICES

Appendix A: PROGRAMS	130
Appendix B: RESULTS	132

1 INTRODUCTION

E-learning systems are becoming more and more popular at educational institutions. The introduction and the rapid growth of e-learning systems have changed the traditional teaching and learning approaches of both teachers and students. The increased popularity of e-learning has created a huge amount of educational resources and as a result, locating the suitable learning references is a formidable challenge [6]. In a typical University e-learning environment, students are expected to follow all the existing course materials that are available in the e-learning systems like Moodle [37] and Sakai [3]. The time available for a student in semester based educational system is limited and bounded, when it is compared with the materials that they need to refer.

1.1 Motivation for the research

These excess materials lead the students to spend more time on browsing and filtering to identify information that suits their needs better, rather than focusing on actual learning. If the proper guidance is provided to select the right materials it is possible to spend time more effectively. The process of selecting “good resources” may be relatively easier for “good” students. Hence, making the suggestions based on “highly accessed materials” by the “good students” to average students would help them to pay more attention on studies rather than spending time on hunting for “good materials”. This research therefore aims to propose an intelligent recommender system based on the intelligence gained from the access volume of “good learners” to the students who are not capable of identifying relevant learning materials that are suitable for their studies. The idea of learning from best students or good learners is also strongly supported by the Social Learning Theory, that states that people can learn by observing the behavior of others and the outcome of that behavior [1].

1.2 Research objectives

The objective of this research is to provide better guidance to students who face difficulties in selecting suitable materials for their studies by implementing an intelligent learning resource/material recommendation system. Implementation of an intelligent

recommendation system to the learning management systems using learning analytics would take e-learning to the next level. The e-learning process lacks adequate feedback or guidance to the student which is a key in the traditional teaching process. Hence, the ultimate goal of this research is to provide better guidance to the students in Moodle e - learning environment using learning analytics.

1.3 Research design

The sub section research design discusses briefly about the overview of the methods carried out in this research and its scope.

1.3.1 The scope of the research

The research extracted the data from the Moodle instance of Department of Computer Science and Engineering, University of Moratuwa. The analysis has been performed on selected 10 courses that are rich in electronic course contents. This resultant model that was derived through this research can be used as reference for the future researches, but cannot be directly applied in different contexts as they may vary in student populations, academic institutions and the course management systems that are placed in [3] .

1.3.2 The research method

The data mining concept was used in this research to perform data analysis. The weblog of the Moodle system was used as input to the data mining tool to build the models, followed by data preprocessing steps. The constructed models were validated with the random data and their recommendations ratified.

Identifying good students and recommending learning resources are two key problems that were dealt within the methodological framework. The quality of the learning material recommendations has an important effect on a student's future learning behavior [2]. Hence the model should be designed accordingly. Poor recommendations can cause two types of characteristic errors either false negatives or false positives. The false negative means some of the learning materials that are not recommended, though the students need to study them. False positive, states the opposite behavior where some learning materials are getting recommended, though the student actually does not need

them. In an e-learning environment, it is most important to avoid false positives errors, because they will lead to disgruntled students and they will unlikely to revisit the site [2].

In this research, student classification model and resource recommendation models are constructed. The former classifies the students as “good students” and “average students” based on the students’ grades and other activities within the course. The latter model recommends the “mostly accessed” resources by the “good students” to the “average students”. The principle of classification in data mining was used to construct the data models.

1.3.3 The data collection and analysis

This research used the weblog data generated by the Moodle system of Department of Computer Science and Engineering, University of Moratuwa to analyse and derive the models. The Moodle’s log contains a rich set of data that was help to design models for the recommendation systems.

1.4 Thesis outline

Chapter one (introduction) contains the introduction to thesis, including motivation of the research, research objectives and the introduction about research methodology. Chapter two (literature review) contains literature review on e-learning and resource recommendation frameworks. Chapter three (methodological framework) discusses the methodological framework in depth which was used to construct the models. Chapter four (analysis of results) analyses the results which were obtained in chapter three. Chapter five (conclusion) attempts to build some concluding remarks with the recommendations.

2 LITERATURE REVIEW

2.1 E-learning systems

E-learning is not a new concept to the world. It is quite old and has been used for more than a decade but, learning analytics is gaining a considerable attention in higher education [3]. A continuous stream of weblog data being collected, and stored in course management systems can be applied as input to build the predictive models using data mining techniques that can be used to implement data driven decision making practices. Despite this early success, academic analytics remains as an immature field that has yet to be researched broadly across a range of institutional types, student populations and learning technologies [3].

The previous researches done using different e-learning systems cannot be directly applied to an existing problem, rather they can be only used as reference [3]. The generic methodological framework for student predictive models was laid down in [3]. The steps need to be followed, the best practices and the flow of learning analytics such as data collection, preprocessing, attribute selection, model building, testing, validation and data quality issues all of them have been discussed in [3]. The variability among the courses is an issue as it could inaccurately represent differences in behaviors between students [3]. The courses vary in their contents, lecturers and activities. Hence the attributes cannot be used as it is and they have to be normalized and the variability of the course has to be dealt with.

All access events of the learning management system can be tracked and that contains the transactions chronologically ordered, along with the URL and the timestamp [16]. Log files contain the login name of the user who generated the request, the date and time of the request, the method of the request (GET or POST), the name of the requested file, and the result of the request (success, failure, error, etc.), the size of the data sent, the URL of the page, and a bit of data generated by an application and is exchanged between the client and the server. These log entries are not in a format that is directly

usable as the input for mining applications and require reformatting and cleansing in order to extract the necessary information to build up the data models [17].

2.2 The data mining in e-learning

There are existing statistical analysis tools that provide insights of web logs and provide reports on the most popular pages, the most active visitors, etc., in certain spans of time. However, these tools provide very little idea on what is really happening on the website, and they only do the surface analysis [18]. In other words, its ability to help the users to understand the information and use of implicit hidden trends in students' online access behaviors is very limited. Data Mining was originated to tackle problems similar to this. As a field of research, it is almost contemporary to e-learning. Therefore, data mining can be used to extract knowledge from e-learning systems through the analysis of the information available in the form of data generated by its users [18].

Data mining, a discipline combining elements of artificial intelligence and applied statistics, is the process of extracting patterns from large data sets [3]. Data mining is applied successfully in a wide scope of domains ranging from business and scientific settings to law enforcement. Such applications include customer profiling, cross-selling, fraud detection, drug discovery, intrusion detection, and DNA sequencing, among many others.

The e-learning systems' data is increasing exponentially and this vast amount of data can be used for the knowledge extraction using data mining [14]. E-learning data contains a wealth of details compared to traditional learning data [2]. Click-stream which is generated for each and every user action/click on the e-learning site provides information that is required to understand the learning behavior of students, such as the kind of materials they are looking for, and those that may be of interest. The data mining on the click information will provide an overall analysis of student requirements across all learning materials and also will provide interesting relationships or associations, between learning materials [2].

2.3 Previous researches in learning analytics

The Signals project of Purdue University, which applied the principles of business intelligence analytics to academia, facilitates to improve student success, retention, and graduation rates and demonstrate institutional accountability [7]. Through the analytics, the project mines a large data set continuously collected by these tools and applies statistical techniques to predict which students might be falling behind. The Signals System is based on a Purdue-developed student success algorithm (SSA) designed to provide students an early warning (from the second week of the semester) of potential problems in a course by providing near real-time status updates of performance and effort in a course. This early recommendation warning system helps the faculties to identify the students at risk, provide a variety of instructional services, from help desks to tutoring.

The warnings generated by Purdue Early Warning System (PAWS) tended to be general in nature and did not specifically include the resources available for a certain course. These warnings may simply alert the students, but will not provide any learning material recommendation. It would be a good solution if the system was the students who are at risk and recommends some learning material to mitigate risk. This literature review studies the research papers from e-learning recommendations, each was written for different underlying technologies of recommendation frameworks.

Different technologies can be used to build recommender systems. Such technologies include data mining, collaborative filtering, content based filtering, rule based expert systems, artificial networks and fuzzy logic. Different researches have been done by selecting one of the above technologies or any combination of technologies [2] [4] [8]. Each of the above technologies have advantages and disadvantages and based on the analysis of the technologies, the research has selected data mining, as the modeling technology. The data mining is relatively easier and straight forward technology.

2.4 Learning resource recommendations

The resources that need to be recommended can be chosen using different technologies. The following sub sections studies each of the above technologies.

2.4.1 Recommendation Based on Good Learners' Rating

This framework suggests learning materials based on the average ratings given by good learners [1] [4]. The average rating of a learning material is calculated using Equation 2.1.

$$R_{i,j} = \frac{\sum_{i=1}^N r_{i,j}}{N_j}$$

Equation 2-1 Average rating of learning material

$r_{i,j}$. The rating of good learner i on item j .

N_j . The total number of good learners that rated item j .

Note that the calculation for good learners' average rating on a particular item is based solely on good learners' ratings.

Another positive outcome of this paper is that the good learners' rating prediction will be calculated when the good learners' ratings do not exist for a particular item using another formula which is important to avoid the cold start problem [1]. The cold start problem is the one where the recommendation cannot be produced, due to insufficient ratings or absence of ratings that is usually faced by the collaborative filtering technique. The recommendation engine also calculates the document weight for each document, once it is uploaded, to the learning management system database as presented in Equation 2.2 . The keywords are retrieved and based on the frequency of the keyword, the weight of the document is calculated. The resulting weight will be used as the input for the cosine similarity calculation.

$$w_{i,j} = f_{i,j} \frac{f_{i,j}}{\max_z f_{z,j}} * \log\left(\frac{D}{d_i}\right)$$

Equation 2-2 The weight of a document j respective to the term i

$f_{i,j}$ - The frequency of a term i occurs in document j

$\max_z f_{z,j}$ - The maximum frequency among all the z keywords that appear in document j

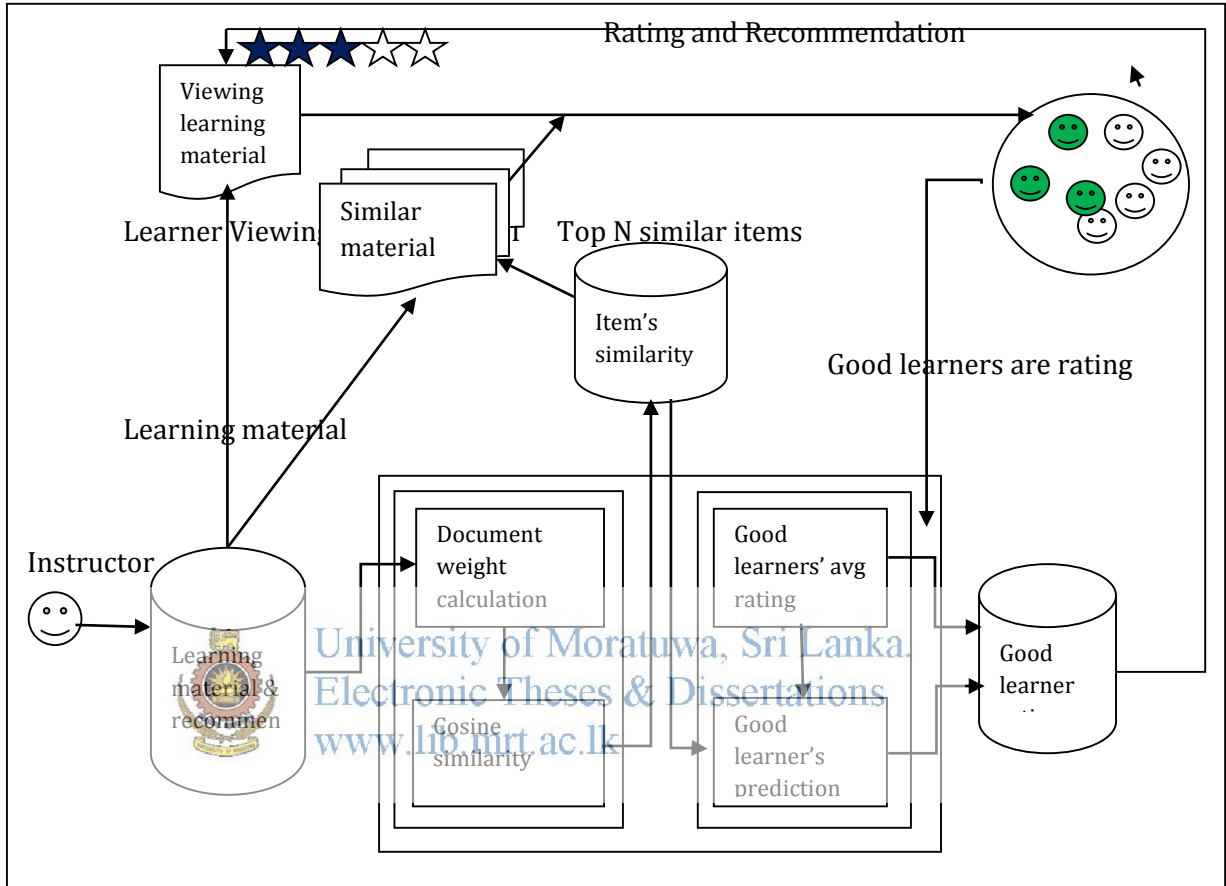
D - Total number of documents that can be recommended for the learners.

d_i - The numbers of documents that contain term i

The normalized frequency ensures that the long documents with high occurrence terms will not have a high impact on the weight, thus it helps to reduce the possibility of keyword spamming. The weight is computed using the above equation is used to calculate the similarity value between two items. The cosine similarity values are calculated by doing production on vector of content based profiles of user and content of documents and divide it by the scalar production of above two vectors.



The Figure 2.1 shows the overall architecture of the recommendation framework which was based on good learner's rating. The items which are rated higher would be shown in the top order.



😊 - Good Learners 😊 - Learners

Figure 2-1 Rating based recommendation ([4])

Though it outperforms in the accuracy of recommendation when it is compared with other frameworks, the drawback of this framework is, it is required to explicitly rate each learning material. Since the Moodle of the University does not have the explicit rating features, the research was unable to follow up the recommendation based on the good learner's rating to build up the models.

2.4.2 Recommendation Based on Good learners' Access Pattern

The pervasiveness of the internet has allowed online distance education to become more than it used to be, and that has happened in a surprisingly short time [18]. E-learning courses that are now offered are plentiful, and many new platforms and e-learning systems have been developed and implemented with varying degrees of success. These systems generate an exponentially increasing amount of data, and much of this information has the potential to become new knowledge to improve all levels of e-learning. Due to the vast amount of data the students face problems in identifying the materials relevant to them. They need to spend more time to browse and filter out the necessary materials within the short semester durations. The “bright students” would manage the situation, but the average students face problems in choosing the relevant materials as a result they score less in their exams.

From a student perspective, it would be very useful if the system could automatically guide the students for their activities and intelligently recommends online resources that support and enhance learning. Automatic recommendation could be more interesting based on browsing patterns of other successful learners [15]. The successful learners' have the capability to discriminate the only relevant materials from the largest collection of resources. Hence, if the access pattern of good students is identified the resources that likely to be used by the good students can be recommended to the average students.

A recommender system suggests possible actions, or web resources based on their understanding of user access [19]. To do this the entries that are in the weblog have to be translated into any of useful information for modeling purposes.

2.4.3 Recommendation Based on Collaborative Filtering

Collaborative filtering is the underlying technology which is used by the most of the recommendation frameworks including e-commerce application recommendation frameworks [2]. That makes sense, because in reality, people make decisions based on the opinions of others, hence the same theory is applicable to the digital world as well. The classification is based on data that is collected automatically (background) and the

data that is introduced by the users (input). Collaborative filtering techniques are based on the ratings that users gave to the products. These ratings are used to find similar users and based on that community of users, products (LMSs or books) are recommended. The demographic approach also finds similar users. The difference is that previous ratings or transactions are not used. Instead the characteristics of the users obtained through a questionnaire are used to group them. Content-based techniques are used to filter data based on a user profile. The user profile is built by finding the habits of the users in the data available [5].

2.4.4 Recommendation Based on Content Filtering

Jie L from University of Technology, Sydney proposed a Personalized Learning Recommendation System in 2004. The personalized learning recommendation system (PLRS) suggests learning materials based on content filtering [2]. The personalized recommendation approaches were first proposed and applied to e-commerce applications by many popular web retailers, including Amazon.com and CDNow.com when the customers buy products. Providing personalized product recommendations would help the customers to find products that they would like to purchase, by producing a list of recommended products for each given customer. Such recommendations are generated by the recommender systems, based on the analysis done, on the past transactions made by each customer [2].

The content-based recommender systems provide recommendations to a customer by automatically matching his/her preferences in the product content, such as recommendation of web pages, news items and video suggestions. In general the customer preferences are predicted by analyzing the relationship between the product ratings and the corresponding product attributes. The content-based recommender has a common problem which requires a large set of key attributes. If the data set is too small, obviously there is insufficient information to learn about the customer profile. Hence, if a customer visits the site, but has not made any purchase and the customer wants to buy

a product which is not frequently purchased. In this case, what products need to be suggested, This is called a cold start problem [1].

This research paper [2] proposes a framework called PLRS which is presented in Figure 2.2 for recommending learning materials to students who may have different backgrounds, learning styles and learning needs. The PLRS tries to identify a student's need, and how to accurately find the learning materials which match the student's need. The framework comprises four components, each with different purposes. 'Getting student information', 'Identifying student requirements', 'Learning material matching analysis and 'Generating recommendations' are the main four components of the PLRS.

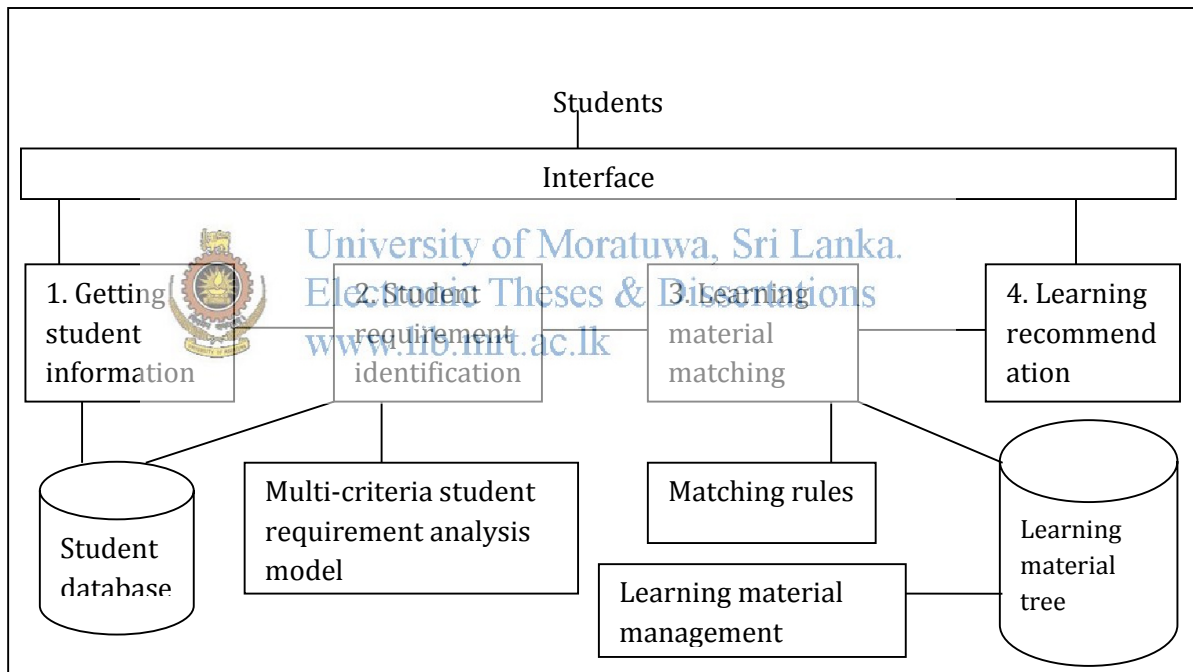


Figure 2-2 PLRS Framework ([2])

A better understanding of how learners accessed the electronic course materials is needed to evaluate the effectiveness of developing and delivering courses. [14]. There is a risk that the adoption of e-learning recommendation can be driven by technology and the need to achieve a return on investment detracts from the evaluation of the benefit to the student of this medium. It has been suggested that learning environments

based on the web have a psychological and physical cost, as it "may affect student performance due to cognitive overload caused by too much information to process multiple resources in a very short time and because of vision problems caused by spending long time to read the materials off computer screens.

Based on the literature review, the research chose the resource recommendation framework based on access behavior of good learners. Though recommendation based on good learners' rating is the ideal solution, but that cannot be implemented at the Moodle right now as this requires getting the rating feedback from the good learners. The collaborative filtering suggests the materials to a student which are accessed by similar students. If an average student requires recommendation, he would often request for suggestions from another average student, who is probably on par with him. The research assumes that it is hard for an average student to recommend the learning resources to another average students who is unable to choose the right learning resources. In addition, the content based filtering would not solve the problem of average learners as it suggests similar learning materials he's interested in based solely on that student's access pattern. By suggesting similar materials to an average student would not enhance his/her learning as the student wants more "good materials" than he/she used to access. Hence the research decided to recommend the resources based on the access pattern of good students. It identifies the good students first and based on their access patterns of resources it recommends the resources which are mostly accessed to the fellow average students.



3 METHODOLOGICAL FRAMEWORK

The principle of data mining was selected to design the model for the recommendation system after the analysis of existing e-learning recommendation systems and their backbone technologies. The classification, a sub division of datamining, is very straight forward and uses simple technologies and also the performance would not be an issue if the data set is not much larger. This chapter discusses the methodological framework that was used to build up the recommendation framework based on most accessed resources by the good learners.

The Figure 3.1 shows the high level design of the overall architecture of the proposed recommender system. The students are classified into either “Good” or “Average” and the materials accessed by the good students are recommended to the average students. These learning materials will be visible when they login to the Moodle for each course module. In addition to all the course materials the students will get suggestions on the right hand side of the course content, main page. The suggestions would vary based on the current location of the student within the Moodle system.



Figure 3-1 High level design of proposed recommendation system

If he logs into the course main page he will get top N documents which were accessed by “good learners” for the whole module. For example, let’s assume the advanced database course module for 4th year students contains around 10 materials per week, 15 week semester duration and altogether 150 course materials for the entire semester. The system would recommend “concurrency control and locking”, “basic data warehouse concepts”, “hashing and indexing techniques”, “data mining and information retrieval” and “query optimization techniques” as the most accessed top 5 course materials by the “good students”. Hence, when average students logged into the Moodle system and when accessing the course module they would get top N materials accessed by “good learners”.

And also when he traverses by the week / section view he would get suggestions for each week. As already stated, each week contains around 10 materials per week and let’s assume the system would suggest the top 3 materials, per week. For example, if the current week is filled up with the course materials that are related to “data mining technologies” in the Moodle system, the recommender system would recommend mostly accessed 3 materials for that week which are mostly related to the data mining. Hence it would be helpful for the students who spend much time on browsing course materials rather than learning materials as they do not need to go through all 10 materials of the week, rather it would be enough for them to learn the materials that are recommended by the system. And when the user traverses a certain location in the Moodle system, the relevant materials will be selected and will be recommended.

The methodological framework used in this research consists of six phases (Refer Figure 3.2). They are Collect data, Transform data, Partition and Reduce data, Build models, and Evaluate and Choose models. The first four phases are used to prepare input data which are later used to construct (train) and evaluate models.

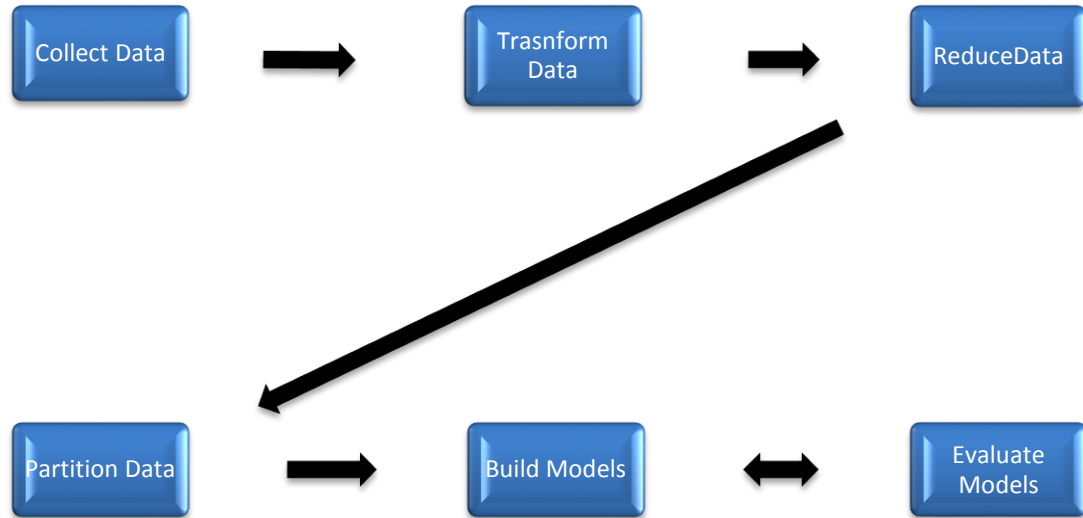


Figure 3-2 The modeling process

The first phase (Collect Data) collects the source data for the learning analytics. It extracts the data for the analytics from different sources such as moodle.xml Moodle e-learning materials. The extracted data has been transformed into useful information in phase two. Several techniques have been applied in the data transformation process. The transformed data are further reduced by removing the missing data, outliers, unknown data and duplicates. In phase four the reduced data has been partitioned into two sets where one set was used to train and build up the models and the balance was used to evaluate the trained model. The partitioned data which resulted from phase four was used to build up the models in phase five using several data mining principles and it was evaluated in phase six. The building and evaluating models were done recursively to choose the best model which produces more accurate and less errors. The detailed description about each phase and the techniques, tools and processes used in this methodological framework will be discussed in upcoming chapters.

3.1 Collect Data

The phase one (Collect Data) comprises a process of extracting data from data sources and an initial preprocessing of data to handle missing values, outliers, strangers (incomplete) records and the calculation of derived data. The data which was used for

the data analytics was extracted from the Moodle learning management system (LMS) of Computer Science and Engineering Department, University of Moratuwa. The backup tool in the Moodle system was used to export the data as archived files. As the research's target was to identify the most accessed learning materials by the good students, which later can be used to suggest to the average students. The courses that were selected for the analysis consist plenty of e-learning materials. Most of those e-learning materials were in the form of pdf, power point slides, word documents, html pages and web links. Higher number of resources would result in more accurate resource prediction model. Based on that the courses shown in Table 3.1 were selected for the data analysis purpose which comprises more resources, more student interactive activities such as forum posts, wiki entries and online quizzes. The courses were selected from the Master of Science (M.Sc.) and Master of Business Administration (MBA) conducted by Computer Science and Engineering department, University of Moratuwa in the last couple of years. The Table 3.1 summarizes the basic information about each selected course.

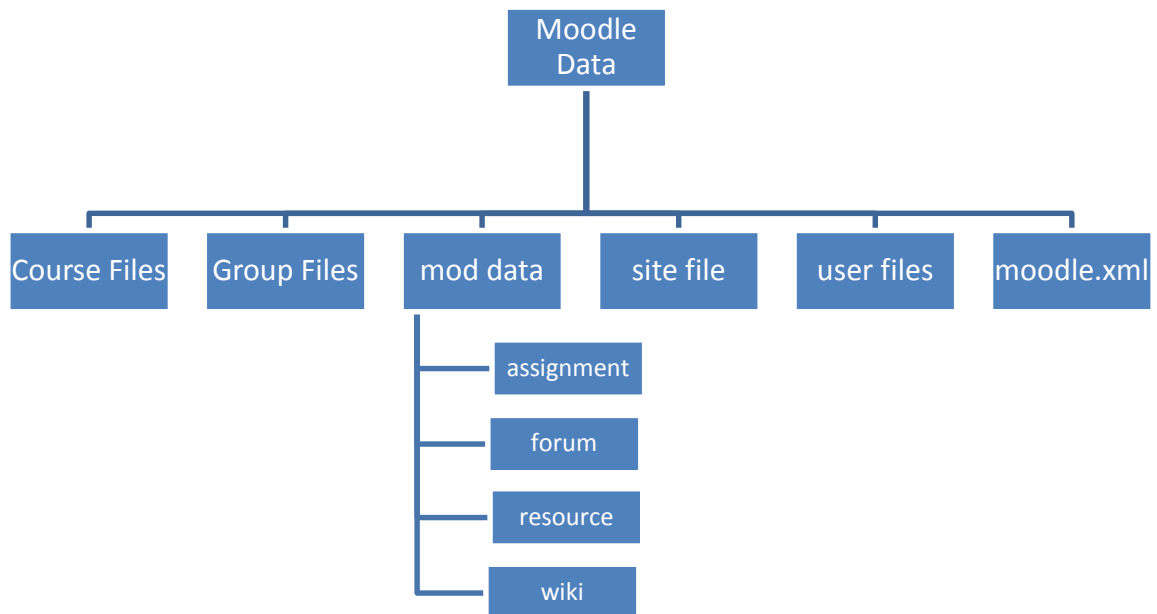


University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Table 3-1 List of Courses selected for the research

Course Name	Category	Year	No of students
Information Security	MBA- IT	2008	42
Information Security	MBA- IT	2009	40
Computer and Network Security	MSc in CS	2010	32
Information Security	MBA- IT	2009	31
System and Network Design	MSc in CS	2011	29
Computer and Network Security	MSc in CS	2011	29
Information Security	MBA- IT	2009	23
Computer and Network Security	MSc in CS	2012	31
Information Security	MBA- IT	2012	24
Computer and Network Security	MSc in CS	2009	33

Each course followed a folder structure that is the same as in the Figure 3.3 when they were extracted from the archived format.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure 3-3 Folder structure of an archived course

The folder “course files” consists of all of the learning materials within itself. And the group files folder comprises of group related data and similarly site files and user files includes the site related data and the user data respectively. The mod data cover the assignments, wiki and forum data. The model.xml holds all of the Moodle related data in xml format. The moodle.xml contains information such as course basic details, user roles, course weekly sections, user information, access logs, groups, events, grades, grade histories and modules. The data collection phase extracts the information from different data sources such as moodle.xml and metadata extraction from Moodle resources. Using these data, the data set is prepared to build up the models. The intermediate steps of data set preparation are demonstrated in Figure 3.4.

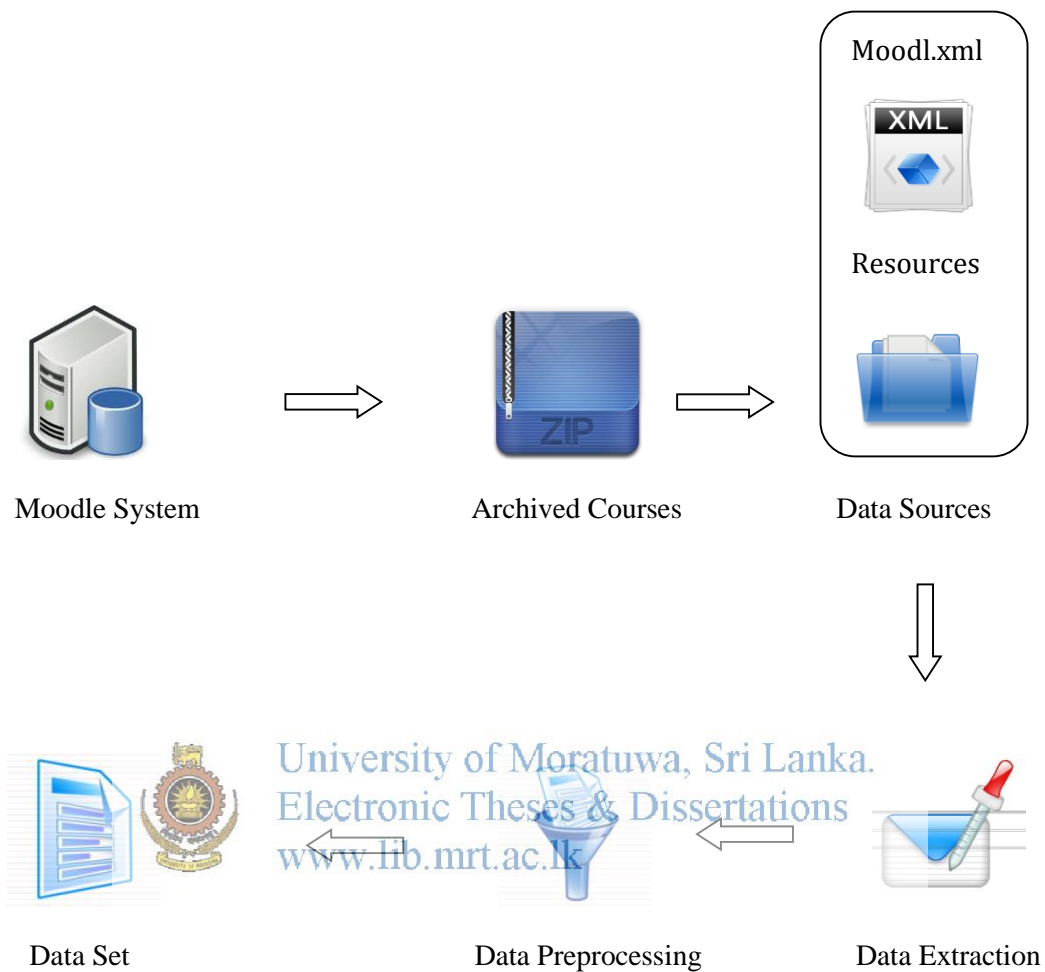


Figure 3-4 The intermediate steps of data set preparation

3.1.1 Remove the user identity from the course data

The anonymization of search logs is an important process that must be performed before the disclosure of such sensitive data. The sensitive data such as employee and customer data or student grade data, medical data of patients should be protected by privacy laws. The reference [22] proposes a solution to this problem to perform anonymization on security monitoring logs before they are sent to the third parties. The logs are the main communication channel between the organization and the third-party, and by applying anonymization to them, it is possible to limit the loss of sensitive information while

permitting the third party to perform safety analysis on the logs. The main challenge will be to balance the level of removal of sensitive information and information required to do the analysis. Anonymization techniques will remove sensitive information, such as the identities of users, students or patients from these logs before sending them to the third parties for the third parties' analysis. A similar anonymization technique was applied to the student data collected from the Moodle prior to submitting them to the researcher. Hence researcher is unaware of the students' grades and other students' privacy data.

The backup data which were collected from the Moodle contains the student's information such as first name, last name, email, index no, their grades for assignments and access logs of each student. Moodle backup data recorded each course events, announcements, resources (e- learning materials), access logs, discussion forums, assignments, in class tests and quizzes. A detailed description of the information collected from Moodle will be discussed in the upcoming chapters. A single student could have taken multiple courses. If the moodle.xml is used for the data analysis process without applying any of the anonymization procedure, it would be highly possible the students' and courses' related confidential data would be exposed to the public. All of the above said data exists in a single moodle.xml file which is a high risk to leak the students' privacy data specially grades to the outsiders. The xml contains the students' data, such as first name, last name, email and an identity number (id). And the grades also exist in the same file under different tag with the student's identity number (Id) and the grades for each assignment.

If anybody had access to the moodle.xml it would be easy for them to get to know the grades of known students. If the student's first name, last name, index no or email is known then that student's identification number can be easily retrieved and using that, the grades obtained by that student can be easily discovered. This would be a big security hole in the University's learning management system. Though the University takes necessary steps to secure the students' information, the security can be leaked

easily to the third parties when it is supplied to the third parties for analysis similar to this. Hence there was a small program was coded in Java to remove or transform the confidential information while preserving the privacy. The Java code would extract the first name, last name, user name/ (index no) and email and convert them to random strings. If the above said attributes are altered in the moodle.xml then it would not be possible for the third parties to identify the students personally and retrieve the student's grades, whoever gets access to the moodle.xml. A small snippet of the PrivacyImpl.java which was implemented to keep the privacy of student's grade data is presented in Figure 3.5.

```

Public class PrivacyImpl
public static void main(String[] args) throws IOException {
    SAXBuilder builder = new SAXBuilder();
    File xmlFile = new File("C:\\ backup-cs5105isec\\moodle.xml");
    File logFile = new File("C:\\backup-ss5105isec\\StringConversion.log");
    if(usersElement != null) {
        List users = usersElement.getChildren("USER");
        if(users.size() > 0) {
            for(int i = 0; i < users.size(); i++ ) {
                Element user = (Element)users.get(i);
                String userName = user.getChildText("USERNAME");
                String email = user.getChildText("EMAIL");
                String newUserName = privacyImpl.randomString(10);
                String newEmail = privacyImpl.randomString(8)+"@uom.lk";
            }
        }
    }
}

```

Figure 3-5 Anonymize the students' private data to lessen confidential leakage

The program gets the backup file's path and it will process that file and convert the student's name and other security vulnerable attributes to randomized strings. And also

the Java code which was coded by the researcher will be used by the authorised department staff to execute against the Moodle backup course data before it is released for analysis. That would update the same moodle.xml file with the generated randomized strings. In addition to that will create another text file called StringConversion.log which keeps the mapping of old attribute value and the new value of the attribute. If the first name, last name, email of a student is Tharsan, Sivakumar and 128232U@uom.lk respectively, then it would be converted to some randomized strings such as mqbzlwixkx, ciyjmkvptd and balzobls@uom.lk. As a result, it would be less possible for any third party, whoever gets the access to the moodle.xml to get the student identity and their associated grades and other related private information.

The program generates randomized strings of 10 characters long for the above student's attributes and for the email, it will generate randomized string of 8 characters and the email will be appended by "@uom.lk" for each student's email. Initially, when the program was run it gave some character encoding errors such as "Invalid byte 1 of 1-byte UTF-8 sequence". The reason was the encoding system used when the course was backed up and the encoding system where PrivacyImpl.java was running were different. The issue was resolved by adding a snippet to the code as shown in Figure 3.6.

```
PrivacyImpl privacyImpl = new PrivacyImpl();  
XMLOutputter xmlOut = new XMLOutputter();  
.....  
user.getChild("FIRSTNAME").setText(newFirstName);  
xmlOut.setFormat(Format.getPrettyFormat());  
xmlOut.output(document, new FileWriter("C:\\ backup-cs5105isec\\moodle.xml"));
```

Figure 3-6 Solution for encoding issues in jdom

And also the program will generate the StringConversion.log file which is in the same location as the moodle.xml resides that will be helpful for the department staff to keep

track the student's attribute values and its generated attribute values. The PrivacyImpl.java was run by one of the department staff and the Moodle course backups were given to the researcher after removing the student identity from the backup file. Hence the researcher has no clue about the grades, access log and other student related data belongs to which student. It completely removes the user identity from the data analysis.

3.1.2 Extraction of moodle data and store it in the database

The research picked 10 courses as shown in Table 3.1 from MBA and MSc of the department of Computer Science and Engineering, University of Moratuwa for its learning analytics research project. Each archived backup of courses has the size in between 30 – 60 MB and each of the moodle.xml contains around 12 – 15 MB of data. In general terms the xml file contains more than 300, 000 lines within it. Hence manually read the XML file and grabbing the data for the analysis would not be good and infeasible solution. As a result a small program was coded in Java to read the xml file line by line and acquire the data which is trapped inside the XML tags and write it back to any database.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

There are several xml text processing libraries available for Java. Initially the “DOM Parser” was selected for the purpose, it was working fine for small test xml file and it threw out of memory exceptions when the file size got larger and the real moodle.xml file was used. Hence, as an alternative “JDOM” parser was selected which is capable of processing large files without any memory related issues to replace the DOM parser. JDOM (Java Document Object Model) is a document object model based on open XML, Java that was specifically designed for the Java platform hence, we can take advantage of their linguistic code. JDOM integrates with Document Object Model (DOM) and Simple API for XML (SAX), supports XPath and XSLT. [21] JDOM is an open source library for Java-optimized XML data manipulations. JDOM was created to be specific Java and therefore take advantage of Java characteristics, including method overloading, collections, reflection and family programming idioms [21].

Since the JDOM has the capability to process the large xml files without much memory associated issues, the JDOM was selected to process the moodle.xml files that consist all

```
import org.jdom.Document;

import org.jdom.Element;

import org.jdom.input.SAXBuilder;

public class ParseXMLUsingJDOM {

public static void main(String[] args) throws SQLException, IOException {

    SAXBuilder builder = new SAXBuilder();

    File xmlFile = new File ("I:\\ Tharsan\\cs5404cns-2010s3"+"\\moodle.xml");

    try {    Document document = (Document)builder.build(xmlFile);

        Element rootNode = document.getRootElement();

        List infoElements = rootNode.getChildren("INFO");

        if (infoElements.size() > 0){

            for (int a = 0; a < infoElements.size(); a++){

                Element node = (Element) infoElements.get(a);

                String name    = node.getChildText("NAME");

                String version = node.getChildText("MOODLE_VERSION");

                String release = node.getChildText("MOODLE_RELEASE");
```

Figure 3-7 Grab data from moodle.xml and transfer it to the database

The ParseXMLUsingJDOM.java was used to process the xml files in the given path and accumulate data from the xml file that are trapped within the XML tags and store them into the database. It imports the JDOM library at the top of the program. Then the absolute path of the moodle.xml is given. The program will generate the MoodleInfoLogger.log and MoodleErrorLogger.log in the same directory where MoodleInfoLogger.log will contain the log information and the MoodleErrorLogger will

log all of the errors or exceptions occurred while running the program with the value of the parameters which caused those errors and exceptions. The program initially builds up the parsed XML tree and then it will retrieve the child elements from the root node. The child elements of xml tag whose tag name is “INFO” can be retrieved using `rootNode.getChildren("INFO")`. This will return the entire child elements from the root element that has the “INFO” tag. And also the text value of xml tag called “NAME” can be extracted using `node.getChildText("NAME")`. These are very straight forward and user friendly built in methods from the JDOM library.

The extracted data from the moodle.xml has to be stored in any form of consistent storage. The relational database system is a good solution for the consistent storage. There are several relational database implementations from different vendors, but the researcher chose SQL Server as the storage system. There is no special reason for selecting this database, other than the familiarity with this database to the researcher. Even other databases get chosen that will not impact the research. The Figure 3.8 having the code snippet to prepare a query string to insert the data into the Course table. Similar to the above snippet there were a lot of query strings which were used to insert the data grabbed from the moodle.xml into different database tables. The Figure 3.8, Figure 3.9 and Figure 3.10 shows a sample insert statement, getting SQL server connection and executing update using the retrieved connection object respectively.

```

queryString = "INSERT INTO Course ([CourseId],[CategoryId] ,[Password],
[FullName],[Summary] " +
.....
"[StartDate] ,[NoOfSections] , [Lang],[Currency] ,[Visible], "
"[NotifyStudents] ,EnrolStartDate] ,[EnrolEndDate]) VALUES (" +
"""+ courseId +""", "" +courseCategoryId+""", "coursePassword +""
"" + courseEnrolStartDate +""", " +courseEnrolEndtDate + ')" "
executeQueryString(queryString);

```

Figure 3-8 Insert the grabbed data into the database

```

public static Connection getConnection() throws Exception {
    Connection connection = null;

    String url = jdbc:sqlserver://localhost;databaseName=Moodle;integratedSecurity=true";
    try {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        connection = DriverManager.getConnection(url);
    }

    return connection;
}

```

Figure 3-9 Sets up the connection to the SQL server

```

public static void executeQueryString(String queryString) {
    Statement stmt;

    try {
        stmt = getConnection().createStatement();
        stmt.executeUpdate(queryString);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

Figure 3-10 Get the connection object and executes the query string

The Figure 3.9 returns the connection object which connects to the Moodle database in the SQL server that is hosted on the local server. The “integratedSecurity=true”; indicates that the connection is authenticated by Windows authentication in SQL server. The connection will use the windows account details to establish the connection to the SQL server. But in the very first instance an error was thrown as shown in the Figure 3.11 when the program was run. This is an authentication error when the connection is established to SQL Server from eclipse.

```
com.microsoft.sqlserver.jdbc.AuthenticationJNI <clinit>
```

WARNING: Failed to load the sqljdbc_auth.dll cause :- no sqljdbc_auth in java.library.path

This driver is not configured for integrated authentication.

Figure 3-11 SQL server error in Windows authentication mode

The solution was to copy the sqljdbc_auth.dll from bin directory of the Java runtime installation (C:\Program Files (x86)\Java\jre1.6.0\bin) to Windows system32 location (Windows\system32)

The same data extraction procedure was used to collect different set of information from the moodle.xml. The Table 3.2 tabularizes the different set of information grabbed from all of the courses and stored in the Moodle database as database tables.

Table 3-2 List of tables in the Moodle database filled by moodle.xml

Table Name	moodle.xml Tag Name	No of attributes	No of records	Description
Blocks	BLOCKS	11	84	Contains the blocks of course index page
Course	COURSE	42	10	Course information
CourseRoleAssignments	ROLES_ASSIGNMENTS	12	360	The role assigned to each user
Events	EVENTS	17	17	The events of courses
Grade	GRADE_ITEMS	22	4552	The grades of all students for all assignments
GradeCategories	GRADE_CATEGORIES	12	73	Assignment grading categories e.g. forum, text, project, wiki
GradeCategoryHistory	GRADE_CATEGORIES_HISTORIES	17	628	History of grade categories
GradeHistory	GRADE_GRADES_HISTORIES	25	17730	History of Grades
GradeItemHistory	GRADE_ITEM_HISTORIES	30	2224	History of Grade Items
GradeItems	GRADE_ITEMS	28	238	All of the grade items belongs to each grade category

GradeLetter	GRADE_LETTERS	4	99	The grading letters for each grade boundary
Groups	GROUPS	9	323	The groups details, group members
Log	LOGS	10	273750	The user activity of all users
ModAssignments	MODULES	32	1192	All of the student assignments data
ModForum	MODULES	21	1717	All of the student forums
ModForumDiscussion	MODULES	31	1303	All of the student forum activities
ModLabels	MODULES	6	191	All of the labels of course
ModResource	MODULES	28	489	All of the resource information uploaded to the Moodle
ModWiki	MODULES	34	1878	All of the student's wiki activities
MoodleDetails	HEADER	8	954	Moodle course structure details
MoodleInfo	INFO	18	10	Basic details of course backup
Roles	ROLES	8	3262	The roles in the system and their capabilities
Users	USERS	43	352	The existing users in the Moodle for given courses
UserPreferences	USER_PREFERENCES	46	1871	Each user's preference settings

There were 23 tables that were created within the Moodle database which were directly fetched from moodle.xml. The data for each table were extracted using the relevant xml tags. The Table 3.2 lists down the tables which were directly populated from the moodle.xml and the parent tag that was used to identify the data in those tables. In addition to that it displays the number of attributes that the relation consists and the number of rows that the tables possess. The last column of Table 3.2 briefly describes each table. The Course and MoodleInfo tables are smaller tables which contain only 10 records where the log table is the largest among the tables which consists around two hundred and seventy five thousand entries. Some of the tables were used to build up the models, and some were transformed into other tables and some used to get few data.

3.1.3 Extraction of moodle resource data

The resource table was used to store the file related data of all of the Moodle resources. Each of the Moodle course backup consists of a folder called course files to store all of the resources as shown in Figure 3.2. The metadata and other useful information about those resources(files) were stored in the resource table. The metadata about those files were extracted using a Java program (Refer Figure 3.12) and inserted into the Resource table in the Moodle database in the SQL Server platform. The FileMetaDataExtraction.java was used to extract the file metadata of the Moodle resources. The program accepts the Moodle backup directory as an argument and it will process all of its files, folders and sub folders and collect all of the information about all of the files resides within the course files directory of backed up courses. It collects information such as file name, file size, file type, whether is it a regular file or symbolic link, file creation time, file last accessed time, file last modified time, file owner and file extension. All of this information about each file was inserted into the Resources table using a SQL query. The Resources table was populated with all together with 607 e-learning material records.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

```
public class FileMetaDataExtraction {  
  
    static String backedUpCourse = "backup-cs5404cns-2010s3-20130518-1701";  
  
    public static void main(String[] args) {  
  
        File folder = new File("I:\\MSc\\Tharsan\\" + backedUpCourse  
+ "\\course_files");  
  
        FileMetaDataExtraction metaData = new FileMetaDataExtraction();  
  
        metaData.listFilesOrFolders(folder); }  
  
    .....  
  
    /* Continue in next page */  
}
```

```

public void listFilesOrFolders(File folder)    {
    FileMetaDataExtraction metaData = new FileMetaDataExtraction();
    for(File fileEntry: folder.listFiles()) {
        if (fileEntry.isDirectory())    {
            listFilesOrFolders(fileEntry); }
        else {    Path path = Paths.get(fileEntry.getAbsolutePath());
            metaData.fileMetaDataExtraction(path); } }
    }

public void fileMetaDataExtraction(Path filePath){
    File file = new File(filePath.toString());

    BasicFileAttributes fileAttr = Files.readAttributes(filePath,
    BasicFileAttributes.class);

    fileSize = basicFileAttributes.size()/1024+" KB";

    FileCreationTime = basicFileAttributes.creationTime().toString();
    LastAccessed = basicFileAttributes.lastAccessTime().toString();
    UserPrincipal owner = Files.getOwner(filePath);

    FileOwner = owner.getName();    int lastDot = fileName.lastIndexOf(".");

    String extension = fileName.substring(lastDot + 1);

    fileExtension = extension;
}

```

Figure 3-12 The code snippet for the data extraction of Moodle resources

The size is calculated in Kilobytes (KB) and also the extension is deduced by chopping off the substring after the last dot in the file name. Hence the first phase involves collecting the data from different sources such as moodle.xml and resource folder and those will be transformed into useful dataset to build up the models after doing the necessary data transformation, reduction and partition in 2nd, 3rd and 4th phases in this chaining process.

3.2 Transform data

The transformation data phase applies a series of rules or functions to the extracted data that are from different sources to derive the dataset to build up the data models. The transformation can be done by using the following techniques.

- ✓ Joining multiple tables to derive a new table
- ✓ Aggregation
- ✓ Pivoting
- ✓ Deriving newly calculated values

By combining the above technologies the transformation can be done and this paragraph illustrates about some of the transformations done during research. In addition to the tables shown in Table 3.2, four new tables were introduced as indicated in Table 3.3, that store the transformed data from either single or combination of source tables.

Table 3-3 A list of tables which were derived from tables in Table 3.2

Table name	Source table(s)	No. of attributes	No. of rows	Description
Student	Students, log, UserGrade, CourseRoleAssignmnt, ModAssignment, ModForum, ModWiki ModForumDiscussions,	84	332	The table which is used for building the student model.
StudentResourceLog Pivot	Students, log	18	332	The number of resources accessed in each week for each student. The week wise access is shown in separate columns
StudentResourceLog	Student, log	4	5312	The number of resources accessed in each week for each student. The week wise access is shown in each row
AccessResourceSkewness	StudentResourceLogPivot, Course, Student	3	319	The skew of number of accesses in each week of all students
DroppedOrNoGradeStudents	Students, Grade, log	2	24	The students who do not have grades/dropped outs.

The student table was used to build up the student classification model. That was populated with different data from various other source tables which were tabled in Table 3.1. It contains 84 attributes and they were derived from different tables. Forty three attributes were directly fetched from Users table without any alteration. But the users who have the roles other than student such as teacher, non-editing teacher were removed while loading the data from the User table to Student table. The rest 41 attributes were computed using several complex stored procedures and functions.

3.2.1 Dropout students

The Student table contains a column called IsDropped to indicate whether student dropped off the course or are still actively participating in the course. The student details will be entered into the User table once they enrolled for a course. But they may drop the course within the add/drop period if they are not interested to continue the course. There is no indication from the moodle.xml whether the student has dropped the course or is still actively following it. As a result this was derived from existing tables. Whether the student has dropped the course or not, is calculated from the log table where a student has access for 1st two weeks and doesn't have any access after that means, he has dropped the course, after the add/drop period and he will not take part the course anymore. The students' attribute IsDropped need to be updated whether he/she dropped or not. The first task is to get all of the dropped students and then, if the a given student is in the dropped student's list, then update that student's IsDropped column as boolean true, otherwise false. The update_add_drop_to_student_table.sql stored procedure (Refer : Figure 3.13) updates the Student table with their IsDropped attribute whether it is dropped or not dropped. It loops through the Student table via a cursor and check whether the student exists in the temp table called "#TmpDroppedStudents" or not. If the student exists in the #TmpDroppedStudents temp table for a given course, then his Is Dropped field will be set to true, otherwise false.

```

OPEN @getStudents
FETCH NEXT
FROM @getStudents INTO @CourseId, @UserId
WHILE @@FETCH_STATUS= 0
BEGIN
DECLARE @IsDropped bit
SET @IsDropped = 'true'
IF EXISTS (SELECT * FROM #TmpDroppedStudents WHERE CourseId = @CourseId AND
UserId = @UserId)
SET @IsDropped = 'true'
ELSE
SET @IsDropped = 'false'
UPDATE [Moodle].[dbo].[Students]
SET [IsDropped] = @IsDropped
WHERE CourseId = @CourseId AND UserId = @UserId

```

Figure 3-13 Distinguishes the dropped students

3.2.2 The Computed attributes of student classification model

The Student classification model was constructed from the Student table that consists 84 attributes. Some attributes were removed as they were identified as those attributes do not impact the model's prediction. As a result, they were wiped out when the input data set was prepared to build up the Student classification model. Forty three attributes were directly fetched from the Users table and 41 were computed using very complex SQLs and stored procedures. The computed attributes performed as the backbone of the Student classification model and some interprets the text data into very useful figures. Some of the important computed attributes which were formalized using very complex logic are discussed in the next subsection.

The average ratio of student

The importance of evaluation was acknowledged in the long history of education. Examinations or assignments have played a vital role to do the evaluation of students. The assignments/ examinations are important to ensure whether the student has correctly understood the subject that they are covering up during the course. The Moodle system also provides assignments and those are varied in type such as online quizzes, online text, offline activity, upload single file, forum discussions and wiki entries [37]. The online quiz provides questions of different types such as multiple choice questions (MCQ), short answer and essay type answer. The “online text” type assignment requires the students to submit text, using the normal Moodle editing tools. Students type directly into the Moodle text editor later the teachers can provide online feedback. The “upload single file type” assignment accepts the assignment outcome as a single file. This could be a word document, spreadsheet or anything in digital format.

A student may submit a file as many times as they want until the deadline. Only the latest file is retained, and this is the one the lecturer marks down. In the assignment type of “offline activity”, the teachers provide a description and due date for an assignment outside of Moodle. A grade & feedback can be recorded in Moodle. Of course the forum discussion and wiki are a subset of offline activity. These types of assignments are useful when the assignment is performed outside of Moodle.

The grades for all types of the assignment can be captured from the moodle.xml and it was stored in the UserGrades table after successfully processed, using the ParseXMLUsingJdom.java. By using it as the source table it was loaded into the Students table with slight data transformations. There was cursor written inside a stored procedure to process all of the user grades and calculate some grade related statistics. Initially the average marks that a student obtained for each course, calculated within that cursor. Afterwards the average grade that was achieved by all students whoever enrolled for that course was calculated. Using the above two calculations, the ratio of average marks that is shown in Equation 3.1 was derived.

Ratio of average marks = average marks of a student for all assignment in a course / average marks of the entire student for all assignment in a course.

Equation 3-1 Ratio of average marks (Average ratio)

The stored procedure `average_marks_of_student_for_a_course.sql` as the code snippet in the 3.14 was used to calculate the `AvgOfStudent`, `AvgOfCourse` and `AvgRatio`.

```
INSERT INTO #TmpCourseAvg(CourseId, CourseGradeAvg)
SELECT CourseId, AVG(CAST(FinalGrade AS NUMERIC(16,4)))
FROM UserGrades
WHERE FinalGrade != '@NULL@'
GROUP BY CourseId
```

Figure 3-14 Eliminates the students with NULL grades

The average of a course is calculated as shown in Figure 3.15. The average marks obtained by all students for each course is calculated by using the mean value of the final grade of each course. The grades which has the Final Grade value NULL are ignored. The average of a course is required to identify the difficulty level of the course or the strictness in grading or the brightness of students. Higher the `AvgRatio` means the students are smarter. The students score good marks for the assignments as a result the `AvgOfCourse` is higher. Sometimes, though students are not that much smarter, the difficulty level of the subject is lesser or the teacher or lecturer may not be so stricter while marking assignments. Hence the average of a course is important to identify the students stand on that course. The Table 3.4 shows the means obtained by all students in each course.

Table 3-4 The mean of the averages obtained in all assignments for each course

Course Name	Category	Mean Grade of Course
CS5105ISec-2008S3-Information Security	MBA-IT	78.32
CS5105ISec-2009S3-Information Security	MBA-IT	79.14
CS5404CNS-2010S3-Computer and Network Security	MSc	75.07
CS5105ISec-2010S4-Information Security	MBA-IT	76.41
0-CS5401SND-2011S2-System and Network Design	MSc	46.95
CS5404CNS-11S3 - Computer and Network Security	MSc	71.85
CS5105ISec-2011S3 - Information Security	MBA-IT	76.19
12S3 CS5404CNS Computer and Network Security	MSc	72.65
0-12S3 CS5105ISec Information Security	MBA-IT	78.59
CS5404CNS-2009S2-Computer and Network Security	MSc	76.90

The AvgOfStudent is calculated from the mean of the marks obtained by a student for all of the assignments for a course excluding NULL. The assignment marks of a student can be queried from the UserGrade table. Rather than getting the average marks scored by a student, the ratio was taken into account in order to eliminate the course impactness introduced in the course. There may be some courses where students can score very high marks, but in some case it would be difficult for them to reach the score. If a student scores higher marks it does not mean that he is always brighter than, one who scores less marks. It depends on the course. Hence, building up the student classification model the course level dependencies have to be removed, as a result the research derives the AvgRatio of a student as shown in the Equation 3.1.

```

SET @StudentAvg = ( SELECT AVG(CAST([FinalGrade] AS NUMERIC))
FROM [Moodle].[dbo].[UserGrades]
WHERE UserId = @UserId AND CourseId = @CourseId AND FinalGrade != '$@NULL@$'
GROUP BY UserId, CourseId )

SET @CourseAvg = (SELECT CourseGradeAvg FROM #TmpCourseAvg WHERE CourseId = @CourseId)

SET @AvgRatio = @StudentAvg / @CourseAvg

```

Figure 3-15 Computes the Average Ratio of each student

Percentage of open discussions

The forum module is an activity where students and teachers can exchange ideas through comments. Forum posts can be graded by the teacher based on the contribution the student has made to exchange the knowledge with the other students. It is useful for mutual support and / or collaborative learning where students are physically remote. A discussion forum allows participants to communicate online with text. Moodle allows tutors to establish and set up online forums for groups or subgroups of students, which may include text and other media. Participants can log into Moodle to view them. In Moodle, messages are organized by thread (all responses to a given forum post), responses indented below their antecedent response [37]. A forum can contribute significantly to successful communication and community building in an online environment. Forums encourage more deliberate, less spontaneous, contributions, so may be more appropriate for commenting on a given topic than for brainstorming.

The selected courses in this research also have too many forum discussions to evaluate the students. The students need to discuss the requirements and come to the conclusion within their group. The communication needs to be done via the forum discussion. The lecturer may request for comment on a given material after reading it or the students need to solve the problems through the forum discussions. Whatever the scenario, the students need to actively participate in the forum discussion and need to use the forum discussion as a collaborative learning environment. Based on the discussions within the forum, as a group they need to conclude their forum discussions and need to share their conclusions and results with the rest of the students. The contribution to this discussion flow is evaluated by the lecturer, by inspecting at the forum discussions. A student can open up any number of new forum discussions and can comment on forum posts published by him/her or other students. Normally the forum posts are visible only to their group members. Some are open to all the students in the course. The research assumes that whoever opened up more forum discussion and posting more comments

are more interested in the course. As a result the research tried to calculate those figures as well.

A stored procedure called `average_no_of_opened_discussions.sql` was written to calculate the percentage of open discussions. The percentage of open discussions are derived by calculating the percentage from the number of open discussions of the student and the number of open discussions in the course. As shown in Equation 3.2 the percentage of open discussions are calculated from the percentage value between the number of open discussions by the student and the course.

$$\text{Percentage of opened discussions of student} = (\text{number of opened discussions of a student} / \text{total number of opened discussions in the course}) * 100$$

Equation 3-2 The percentage of open discussion

Some courses may have more forum discussion type assignments, as a result a student would get more chances to open more discussions than students in other courses. Hence the percentage of open discussions were calculated and used as an attribute of the student classification model. Further some students may have opened more forum posts for some specific forum discussions. They may be more interested in those topics or they may have more knowledge on the discussion topic and in some forum posts they may contribute less. So in order to eradicate the topic level dependencies the percentage of open up discussions, is calculated at the course level. Hence the course level and the topic level dependency may be ignored.

All of the forum discussions are stored in the `ModForumDiscussion` table with all of the forum post details. But the very first forum posts of each discussion can be differentiated from other forum posts using `ParentId` column. The `parentId` would be 0 for the first discussion posts. When the students responded to them, those will be child posts to the first discussion post.

The store procedure average_no_of_opened_discussions.sql has the SQL statements to calculate the percentage of open discussion by a student. The SQL code snippet shown in Figure 3.16 is part of the average_no_of_opened_discussions.sql.

```

INSERT INTO #TmpOpenedDiscussion(CourseId, NoOfOpenedDiscussion)
SELECT CourseId, COUNT(*)
FROM [Moodle].[dbo].[ModForumDiscussion]
WHERE ParentId = 0 -- Parent id = 0 means they voluntarily opened the discussion
GROUP BY CourseId -- may be good students

```

Figure 3-16 Collects all of the opened discussions of each course into a temp table

The snippet calculates the number of open discussions per course. As described above the query fetches only the discussions which have the parentId value equal to zero. The variation of a number of open discussions within the selected course for the research is tabulated in Table 3.5.



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
 www.lib.mrt.ac.lk
 Table 3-5 Number of open discussions per course

Course Name	Category	No of Opened discussion
CS5105ISec-2008S3-Information Security	MBA-IT	99
CS5105ISec-2009S3-Information Security	MBA-IT	7
CS5404CNS-2010S3-Computer and Network Security	MSc	110
CS5105ISec-2010S4-Information Security	MBA-IT	19
0-CS5401SND-2011S2-System and Network Design	MSc	44
CS5404CNS-11S3 - Computer and Network Security	MSc	37
CS5105ISec-2011S3 - Information Security	MBA-IT	12
12S3 CS5404CNS Computer and Network Security	MSc	73
0-12S3 CS5105ISec Information Security	MBA-IT	22
CS5404CNS-2009S2-Computer and Network Security	MSc	133

Based on the Table 3.5, if the number of open discussions is selected instead of percentage of open discussions of a student, then the result would be dependent on the course. Some courses have more opened discussion which would probably more

opened the discussion by student and also some courses have very less opened the discussion, maybe those courses have less forum discussion type assignments. In order to eliminate the course dependency the percentage value of opened discussion is calculated as exposed in Figure 3.17.

```

SET @StudentOpenedDiscussion = (SELECT COUNT(*) FROM ModForumDiscussion
                                WHERE CourseId = @CourseId AND PostedUserId =
                                @UserId AND ParentId = 0
                                GROUP BY PostedUserId, CourseId )

SET @CourseOpenedDiscussion = (SELECT NoOfOpenedDiscussion FROM
                                #TmpOpenedDiscussion WHERE CourseId = @CourseId)

SET @PercentageOfOpenedDiscussion = ( @StudentOpenedDiscussion * 100 ) /
@CourseOpenedDiscussion

```

Figure 3-17 Assess the percentage of open discussions of each student

The stored procedure calculates the number of open discussions by a student via looping the Student table using a cursor. The number of open discussions in a course is calculated from the temp table #TmpOpenedDiscussion. Using these two values the percentage of the opened discussion of a student is deduced and these figures are updated in the student table against the respective columns.

Percentage of replies to forum discussions

Similar to the percentage of open discussion, the percentage of replies also taken into account. A student can reply to his forum discussion which was started by him, or other discussions started by other peer students. This is one of the important features of collaborative learning. The research expects a good student should possess the behavior that he/she reads the other's forum posts and get the knowledge from them and share his or her expertise too with other peer students. The percentage of replies to the forum discussion is also calculated in a very similar manner except with one difference. The replies to a forum discussion will have a parent post, to which they are replying to. The very first forum discussions have the parentId as 0 whereas the replies of forum discussions have the parentId as the first forum discussion post's Id. As shown in Figure

3.18, the number of replies to the forum discussion of a course is stored in a temp table called #TmpReplyToDiscussion which will be used to calculate the percentage of replies of a student that is shown in Equation 3.3.

```

INSERT INTO #TmpReplyToDiscussion(CourseId, NoOfReplies)
SELECT CourseId, COUNT(*)
FROM [Moodle].[dbo].[ModForumDiscussion]
WHERE ParentId != 0 -- Parent id != 0 means these are replies to the 1st forum posts.
GROUP BY CourseId -- we exclude the 1st post as we counted already

```

Figure 3-18 Picks all of the replies by the student per course

Percentage of replies to forum discussion of student = (number of replies to forum discussion of student / number of replies to the forum discussion in the course)*100

Equation 3-3 The percentage of replies to the forum discussion

As discussed in the percentage of open discussions by a student, in order to remove the course dependency percentage is calculated. As some courses may have little room for the forum discussion where some have more. The Table 3.6 exposes the variability of the number of replies to the forums within each course selected for this project.

Table 3-6 The number of replies posted per course

Course Name	Category	No of replies to the discussion
CS5105ISec-2008S3-Information Security	MBA-IT	56
CS5105ISec-2009S3-Information Security	MBA-IT	189
CS5404CNS-2010S3-Computer and Network Security	MSc	71
CS5105ISec-2010S4-Information Security	MBA-IT	97
0-CS5401SND-2011S2-System and Network Design	MSc	34
CS5404CNS-11S3 - Computer and Network Security	MSc	45
CS5105ISec-2011S3 - Information Security	MBA-IT	56
12S3 CS5404CNS Computer and Network Security	MSc	80
0-12S3 CS5105ISec Information Security	MBA-IT	72
CS5404CNS-2009S2-Computer and Network Security	MSc	47

It is very notable the course CS5105ISec-2009S3-Information Security has very less number of opened discussions which has only 7 as per the Table 3.5. But it has the most number of replies to the discussion which indicates the discussions were going on fewer threads rather than opening up new forum discussions. The SQL snippet shown in the Figure 3.19 calculates the percentage of replies.

```

SET @NoOfStudentRepliesToForum = ( SELECT COUNT(*) FROM ModForumDiscussion
                                   WHERE CourseId = @CourseId AND PostedUserId =
                                   @UserId AND ParentId != 0
                                   GROUP BY PostedUserId, CourseId )

SET @NoOfStudentRepliesToForum = CASE WHEN @NoOfStudentRepliesToForum IS NOT
NULL THEN @NoOfStudentRepliesToForum ELSE 0 END

SET @NoOfCourseRepliesToForum = (SELECT NoOfReplies FROM
#TmpReplyToDiscussion WHERE CourseId = @CourseId)

SET @PercentageOfRepliesToForum = ( @NoOfStudentRepliesToForum * 100 ) /
@NoOfCourseRepliesToForum

UPDATE [Moodle].[dbo].[Students]
SET [NoOfRepliesOfStudent] = @NoOfStudentRepliesToForum,
    [NoOfRepliesOfCourse] = @NoOfCourseRepliesToForum,
    [PercentageOfReplies] = @PercentageOfRepliesToForum
WHERE CourseId = @CourseId AND UserId = @UserId

```

Figure 3-19 Updates the percentage of open discussion of student

When calculating the number of replies the logic ignores the first forum post, though it is also a forum discussion. The reason is that it was already taken into account for the percentage of opened discussion calculation and it is not necessary to repeat the same post for this calculation as well.

Percentage of wiki entries

This is also another form of collaborative learning as forum discussion. A wiki is indeed a fast method for creating content as a group. It is a very popular format on the web for creating documents collaboratively. Generally, there is no central editor of a wiki, not a single person who has the final editorial control. Instead, the community edits and develops its own content. Consensus views arising from the work of many people. In Moodle, wikis can be a powerful tool for collaborative work [37]. An entire class can edit a document together, creating a class product, or each student can have their own wiki and work on it to get feedbacks from the lecturer and their classmates. The students are requested to discuss any topics and then based on the discussions on the wiki, they need to conclude the wiki and present the final outcome of the wiki discussion to the other students. The contribution can be seen by looking at the wiki history, which shows the number of wiki entries/ edit by the student. The data related to wiki entries are stored in ModWiki table. Similar to the forum discussion the wiki entry also removes the course dependencies by calculating the percentage of the number of wiki contributions of a student and number of wiki entries in the course. The Table 3.7 shows the number of wiki entries in each course. And also the equation which is shown in Equation 3.4 was used to calculate the percentage of wiki entries.

Table 3-7 **Number of wiki entries to each course**

Course Name	Category	No of wiki entries
0-12S3 CS5105ISec Information Security	MBA-IT	164
CS5404CNS-2010S3-Computer and Network Security	MBA-IT	214
CS5105ISec-2011S3 - Information Security	MSc	252
CS5105ISec-2008S3-Information Security	MBA-IT	127
0-CS5401SND-2011S2-System and Network Design	MSc	81
CS5404CNS-11S3 - Computer and Network Security	MSc	196
CS5105ISec-2011S3 - Information Security	MBA-IT	98
12S3 CS5404CNS Computer and Network Security	MSc	204
0-12S3 CS5105ISec Information Security	MBA-IT	90
CS5404CNS-2009S2-Computer and Network Security	MSc	451

Percentage of wiki entries of student = (number of wiki entries of student / total number of wiki entries in the course)*100

Equation 3-4 The percentage of wiki entries

The percentage of wiki entries of a student is derived by calculating the fractional value between the number of wiki entries of a student and number of wiki entries in the course. The more interested students, in other words, the good students would post more wiki entries to drive the wiki discussion towards a worthy end of the discussion. That is the reason the research chooses the percentage of wiki entries as one of the attributes to build up student classification model. The business logic shown in Figure 3.20 calculates the number of wiki entries of a student within a course and the number of wiki entries of a course and based on those figures, it deduces the percentage of wiki entries and those figures are updated in the Students table with the respective columns against the given student records using a cursor.

```

SET @StudentWikiEntries = (SELECT COUNT(*) FROM [Moodle].[dbo].[ModWiki]
WHERE CourseId = @CourseId AND PageUserId = @UserId
GROUP BY PageUserId, CourseId )

SET @CourseWikiEntries = (SELECT NoOfWikiEntry FROM #TmpWikiEntries WHERE
CourseId = @CourseId)

SET @PercentageOfWikiEntries = ( @StudentWikiEntries * 100 ) /
@CourseWikiEntries

UPDATE [Moodle].[dbo].[Students]

SET [NoOfStudentWikiEntries] = @StudentWikiEntries,
    [NoOfCourseWikiEntries] = @CourseWikiEntries,
    [PercentageOfWikiEntries] = @PercentageOfWikiEntries

WHERE CourseId = @CourseId AND UserId = @UserId

```

Figure 3-20 The percentage of wiki entries per student

Percentage of assignment completion

The assignment refers to tasks assigned to the students by their teachers or lecturers to be completed outside the class. The basic objective of assignments is to increase knowledge and improve the skills of the students. The task can be designed to reinforce what students have already learned, prepare for future lessons, extend what they know by having to apply it to new situations, or to integrate their abilities by applying many different skills to a single task. The assignment improves the student learning by correcting the misunderstandings, and highlighting errors in thinking. Hence assignments play a vital role in the students' assessment.

The Moodle system also provides the assignments to the students to evaluate their positions in the course. It is an important duty to the students to complete the assignments on time and submit it for the purpose of grading. Getting higher average for the assignments is not only important, but also needed to complete more assignments. Some students may complete few assignments which they would feel is easy for them and may leave the harder assignments as unattended. Hence the research takes the percentage of assignment completion also as one of the attribute to build up the student classification model. The percentage of assignment completion is derived from the number of assignments completed by the student within a course and the number of assignments given in a course. The percentage of assignment completion is calculated using the above two factors. The Equation 3.5 demonstrates the percentage of assignment completion in the equation form.

$$\text{Percentage of assignment completion of student} = (\text{number of assignment completed by the student} / \text{number of assignments given in the course}) * 100$$

Equation 3-5 The percentage of assignment completion

The Table 3.8 consists of a number of assignments given in each course which was selected for this research project. Some have more assignments where as some have

less. The number of assignments given in a course has an inverse impact on the percentage of completion of assignments.

Table 3-8 The number of assignments given in every course

Course Name	Category	No of given assignments
0-12S3 CS5105ISec Information Security	MBA-IT	5
CS5404CNS-2010S3-Computer and Network Security	MBA-IT	4
CS5105ISec-2011S3 - Information Security	MSc	5
CS5105ISec-2008S3-Information Security	MBA-IT	4
0-CS5401SND-2011S2-System and Network Design	MSc	2
CS5404CNS-11S3 - Computer and Network Security	MSc	6
CS5105ISec-2011S3 - Information Security	MBA-IT	3
12S3 CS5404CNS Computer and Network Security	MSc	6
0-12S3 CS5105ISec Information Security	MBA-IT	5
CS5404CNS-2009S2-Computer and Network Security	MSc	4

If the number of assignments is higher, then students need to spend more time to complete them which would reduce the percentage of assignment completion. Hence the dependency on the course has to be removed that induced the researcher to consider the percentage of assignment completion as one of the student classification model's attribute, instead of a number of assignments completed. The assignment details are stored in the ModAssignments table which contains all of the information related to the assignments. The stored procedure shown in Figure 3.21 calculates the number of assignments completed by a student in a course, the number of assignments given in a course and the percentage of assignments completed and it is updated in the Student table.

```

SET @StudentCompletedAssign = ( SELECT COUNT(*)
                                FROM [Moodle].[dbo].[ModAssignments] WHERE
                                StudentGrade > 0 and UserId = @UserId AND CourseId = @CourseId
                                GROUP BY CourseId )
SET @CourseAssignment = (SELECT
NoOfCourseAssign FROM #TmpAssignCompletion WHERE CourseId = @CourseId)
SET @PercentageOfAssignCompletion = ( @StudentCompletedAssign * 100 ) /
@CourseAssignment

```

Figure 3-21 Quantify the percentage of assignment of completion of each student

A good student would have more percentage of assignment completion. They may have the capability, interest in the course which would make their percentage of completed assignment higher.

Percentage of visits of student

Moodle is an open source Course Management System (CMS), also known as a Learning Management System (LMS) or a Virtual Learning Environment (VLE). There are many types of activities available in Moodle. Some of them are the assignments, forums, wikis, accessing learning materials, calendars and activities for communication and collaboration among students. The Moodle learning management system creates a virtual learning environment where the system provides enough electronic materials in a variety of forms such as PDFs, word documents, power point presentations, spreadsheets, zip files, and the media files such as image, video, and audio files [37]. The students can access those resources and extract the knowledge from them.

The virtual learning environment allows the student to access the Moodle learning management system from anywhere at any time.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

It overcomes one of the main obstacles in the traditional learning management systems where the learning time and the learning places are restricted. Accessing the learning management system and getting the benefit from it, is depends on the student. Nobody is going to inquire from the student as to whether he/she frequently access / visit the learning management system. The door is opened for every student and it is their sole responsibility to get the benefit out of it. No one is going to blame as well as no one is going to care about their access to the LMS. But all of the activities done on the learning management system can be tracked and audited. This audit log would reveal more useful, unknown information about the students, their activities, their access patterns, interests, and their standpoint in the course to the lecturers or instructors. Using this information the lecturers may take necessary steps to change the learning pattern of the students and that is the ultimate goal of the virtual learning environments.

The students can be monitored and their virtual attendance can be captured to identify their interest in the course. It is not hard to come to the conclusion that a student who visits the learning management system has more interest in the course. Having an interest in the subject is one of the behaviors of a good student. Most of the student activities can be recorded and the same were extracted from the moodle.xml and was stored in the log table which contains all of the activities done by all users in the Moodle learning management system. The Equation 3.6 shows the calculating percentage of visits of a student using the number of visits of the student and number of visits by all of the students in a course.

$$\text{Percentage of visits of student} = (\text{number of visits of the student} / \text{number of visits by all students in the given course}) * 100$$

Equation 3-6 The percentage of visits of student

The stored procedure average_visit_per_week.sql was used to calculate the percentage of visits of the student which is shown in the Equation 3.6. That estimates the total number of visits of students in a given course during the 16 weeks since the commencement of the semester as shown in Figure 3.22. It also calculates the total number of visits by all of the students in a given course. Later it derives the percentage of visits of student from number of visits of students and number of visits of all students.

```

SET @NoOfVisitsOfStudent = ( SELECT COUNT(*)      FROM [Moodle].[dbo].[log]

                               WHERE logUserId = @UserId AND CourseId = @CourseId
                               AND              logModule = 'course' AND logAction
                               = 'view' AND

                               (DATEADD (SECOND, CAST (logTime AS INT), CAST ('1970-
                               01-01 00:00:00.000' AS SMALLDATETIME)))

                               BETWEEN (SELECT CourseStartDate FROM
                               #TmpCoursePeriod WHERE CourseId = @CourseId) AND
                               (SELECT CourseEndDate FROM #TmpCoursePeriod
                               WHERE CourseId = @CourseId)

                               GROUP BY logUserId, CourseId )

SET @NoOfVisitsOfCourse = ( SELECT COUNT(*)      FROM [Moodle].[dbo].[log]

                               WHERE CourseId = @CourseId AND

                               logModule = 'course' AND logAction = 'view' AND
                               (DATEADD(SECOND, CAST(logTime AS INT),

                               CAST('1970-01-01  00:00:00.000' AS SMALLDATETIME)))

                               BETWEEN (SELECT CourseStartDate FROM
                               #TmpCoursePeriod WHERE CourseId = @CourseId) AND
                               (SELECT CourseEndDate FROM #TmpCoursePeriod WHERE
                               CourseId = @CourseId)

                               GROUP BY CourseId )

```

Figure 3-22 Assess percentage of visit of course

The stored procedure shown in Figure 3.22 first calculates the total number of visits of a student in a selected course. Anyway, it is not shown here, initially the store procedure fills the course start date and the end date (Though there is no predefined value for course end date, the research assumes the course ends after 16 weeks from the start of the semester) for each course in the temp table called #TmpCoursePeriod. This temp table later used to join with the log table to obtain the number of visits by a student within the course period (during 16 weeks).

It counts the number of log entries with the logModule value “course” and the logAction value “view” which were logged between the course start date and end date which was picked from the temp table. The logModule value “course” means the

student has logged into the course main page in the Moodle system. But even if a student revisits the course main page after clicking several pages within the course since the first log in, that is also counted as a visit. As there is no way to identify the visits within the same session, it is not possible to distinguish the course visits within a session. Hence all of the visits are taken to the count, irrespective of either they are first course visit or visit after several page clicks within the course since the logged in. And also the research calculates the percentage of visits of the student instead of the number of visits of students, which would help to neglect if any course dependency problem arises. The derived value of percentage of visits of student was updated in the students table.

Percentage of resource accesses

The Moodle system provides lots of electronic learning materials to the students which are used by them to get more insight into the course. The students are expected to access those learning materials and get the benefit out from them. The assignments, forum discussion, wikis may depend on the chapters discussed in these materials. The students may get some insight into the assignment or a forum or wiki when they read those materials prior to starting them. Hence that would be a good indicator to measure the interest shown by the student to access those learning materials. Though it is not possible to measure those students whoever accessed/ downloaded those materials actually referred them or not using the generated logs, the research assumes the students whoever access/ download the materials have the real intention to read or get the benefit out from them.

As a result the research tried to measure the accessing volume of the resources. Each course has a different number of resource materials. Each student may access different number of course materials for each different course. A stored procedure using a cursor was implemented to calculate the resource access of each student for each course. Even though purposely or accidentally the same resource has been accessed more than once, it was considered as a single access. The distinct operator was applied at the SQL,

otherwise this may introduce a biased, unreliable result. Initially the distinct number of times a particular student accessed a particular course related materials was calculated. The number of distinct occurrences of “log module” as “resource” per student, per course was counted from the “log” table. Then the total number of times the course materials accessed by all of the students who are registered for that course in that academic semester was calculated. This was calculated as the number of distinct occurrences of “log module” as “resource” per course from the “log” table. Using the above two factors the percentage of resource access was derived as shown in Equation 3.7.

$$\text{Percentage of resource accesses} = (\text{Total number of resources accessed by a student on a course} / \text{number of existing resources}) * 100$$

Equation 3-7 The percentage of resource access

The result would be normalized as the percentage of resource access is calculated instead of calculating the number of resource access of a student within a course. Some courses may have more learning materials while others may have less. Hence it would be possible for a student who enrolls for a course which has lots of e-learning materials to get more number of access times than those who enroll for a course which has less e-learning materials. This course based favoritism should be eliminated at the student classification modeling. That’s the reason to take the percentage of course material access instead of the number of course material accesses. The stored procedure `percentage_of_resource_at_later.sql` was used to get the percentage of resource access of a student. It calculates the distinct number of accessed by a student and the number of existing recourses within a course and based on these two values it deduces the percentage of resource access as shown in Figure 3.24. At the start of the stored procedure, it populates a temp table called # TmpResourceAccess with the number of resources exist in a course as shown in Figure 3.23.

```

INSERT INTO #TmpResourceAccess(CourseId, NoOfResources)
SELECT CourseId, COUNT(*) FROM (SELECT DISTINCT logCmId, CourseId
FROM [Moodle].[dbo].[log]
WHERE logModule = 'resource' AND logAction = 'view') AS X
GROUP BY CourseId

```

Figure 3-23 Accumulates the total number of resources exist in each course

```

SET @NoOfResourcesAccessedByStudent =
( SELECT COUNT(*)
FROM (SELECT DISTINCT logCmId, CourseId, logUserId
FROM [Moodle].[dbo].[log]
WHERE logUserId = @UserId AND
CourseId = @CourseId AND logModule = 'resource' AND logAction = 'view') AS X
GROUP BY logUserId, CourseId)
SET @NoResourcesInCourse = (SELECT NoOfResources FROM #TmpResourceAccess WHERE
CourseId = @CourseId)
SET @PercentageOfAccessedResources = ( @NoOfResourcesAccessedByStudent * 100 ) /
@NoResourcesInCourse

```

Figure 3-24 Quantify the percentage of resource access

Percentage of last minute accesses

One of the usual noted bad behaviors of University students is last minute preparation for the assignments/ examinations. Good students usually plan and prepare for their examination in advance. That's how they can perform well and keep on maintaining their consistency in the studies. Hence the percentage of last minute access also was taken into account in the process of building up the student classification model. Normally the good students are expected to visit to the course and access the project related materials consistently. So measuring up their consistency would be a good

indicator to the student classification model. The consistency of the access of the course by a student can be measured by looking at the distribution of the access log. The access pattern of the student should be normally distributed or positively skewed.

It was assumed that accessing the learning materials after 75% completion of the semester would be considered as late access. As a semester is structured into 14 weeks duration and the study leave, exam week requires at least two weeks, all together it is assumed a semester comprises 16 weeks together. Hence the accesses after 12th week will be considered as late access. The percentage of last minute access is calculated as the percentage of the number of last minute learning material access divided by the number of total learning material access as shown in the Equation 3.8.

$$\text{Percentage of last minute access} = (\text{No of last accessed resources of the student} / \text{total number of accessed resources by the student}) * 100$$

Equation 3-8 The percentage of last minute access

The same stored procedure percentage of resource at late access, which was used to calculate the percentage of accessed resources was used to calculate the percentage of last minute access. (Refer Figure 3.25) The number of last access is calculated by counting the number of learning material access occurrences in the last quarter of the semester as shown in Figure 3.24. And the total material access of a student is calculated as the count of leaning material access of a student during the course. If a student accesses the material before (Though it is logically not possible) or after the semester duration that will not be taken into account. Though there is no valid reason to reject the learning materials accessed prior to semester commencement, it was ignored. The research assumed students cannot access the course related learning materials via Moodle before the course is started. If the learning materials accessed after the semester is over, the research assumes those materials may not be used by the students for their studies. Hence those log entries were ignored.

```

SET @NoOfLateAccessedResources = ( SELECT COUNT(*)
                                     FROM (SELECT DISTINCT logCmId, CourseId, logUserId
                                             FROM [Moodle].[dbo].[log]
                                             WHERE logUserId = @UserId AND CourseId =
@CourseId AND logModule = 'resource' AND
logAction = 'view' AND
                                             (DATEADD(SECOND, CAST(logTime AS INT),
CAST('1970-01-01 00:00:00.000' AS
SMALLDATETIME))) BETWEEN
                                             (SELECT CourseMidDate FROM
#TmpCourseDuration WHERE CourseId =
@CourseId) AND
                                             (SELECT CourseEndDate FROM
#TmpCourseDuration WHERE CourseId =
@CourseId)) AS Y
                                     GROUP BY logUserId, CourseId )

```



University of Moratuwa, Sri Lanka.

Electronic Theses & Dissertations
www.lib.mrt.ac.lk

As stated in the last paragraph all of the log entries of the learning materials access were taken into consideration while calculating the total number of learning material access. There were instances where student accessed the same learning materials more than once by known or unknown. Due to this the distinct number of accesses of the learning materials calculated. Otherwise, it is hard to reject the possibility for a student to have more number of resource accesses than the total number of existing resources in the system, even he/she may not access all out of them. The Figure 3.26 calculates the percentage of late access. If none of the resources accessed by a student, then it updates the percentage of late access as a higher value which is 100% as research believes percentage of resource access of a poor student should be higher as possible. Since the student doesn't access any of the resources it is less probable for that student to be as a good student.

```

SET @NoOfLateAccessedResources = CASE WHEN @NoOfLateAccessedResources IS NOT
NULL THEN @NoOfLateAccessedResources ELSE 0 END

IF @NoOfResourcesAccessedByStudent != 0

BEGIN

SET @PercentageOfLateAccess = ( @NoOfLateAccessedResources * 100 ) /
@NoOfResourcesAccessedByStudent

END

ELSE

BEGIN

SET @PercentageOfLateAccess = 100

END

```

Figure 3-26 Acquire the percentage of last access of a student

Skew of the access of the student

Some distributions of data, such as the bell curve, are symmetric. This means that the right and the left are perfect mirror images of one another. But not every distribution of data is symmetric. Sets of data that are not symmetric are said to be asymmetric. The measure of how asymmetric a distribution can be called as skewness. The data can be skewed either to the right or to the left. The mean, median and mode are all the measures of the center of a set of data. The skewness of the data can be determined by how these quantities are related to one another. The number of resources accessed by the students also can be analyzed along the semester duration. In each week what's the number of accessed resources. The number of resources accessed per week may vary throughout the semester period. Some students may access all of the resources at the very beginning of the semester where as others access all of them at the last minute of the semester. There are still students who access the resources consistently throughout the semester. Students who have consistent access pattern access nearly same number of resources per week throughout the semester. Hence the research tries to experiment the skewness of resource access also as one of the attribute to build up the student models.

Based on that some of the complex stored procedures implemented to derive the skewness of resource access of a student.

Skewed to the right

Data is shifted to the right has a long tail extending to the right. An alternative way to talk about a data set skewed to the right is to say that it is positively skewed. In this situation, the mean and median are both larger than the mode. As a general rule, most of the time for data skewed to the right, the mean will be greater than the median.

Skewed to the left

The situation reverses when we deal with data skewed to the left. Data that is skewed to the left has a long tail extending to left. An alternative way to talk about a data set skewed to the left is saying that it is negatively skewed. In this situation, the mean and median are both less than the mode. As a general rule, most of the time for data that is shifted to the left, the average being less than the median.

The ideal skewness

The ideal skewness depends on the domain of the data that was collected. For example, the retirement age is negatively skewed, since most people do not tend to retire until their sixties and very few people retire before that. The number of children in a family is an example of data is positively skewed, most of the families have between 0 and 5 children, and there are very few families with six or more children. And some data are normally distributed without much deviation. Hence all depends on the domain of the data which is being analyzed. The expected data distributions can be plotted as shown in the Figure 3.27.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

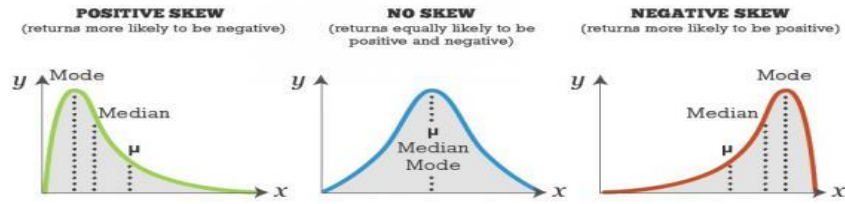


Figure 3-27 The variation of skewness (source: www.pertrac.com)

The data distribution of the number of resources accessed in a semester for all of the weeks can also be plotted. Ideally it is expected that a good student should have resource access pattern of positively skewed access or access pattern without skew also good. But its highly demotivates the negatively skewed access pattern of a student. The research assumes all of the resources related to a course are uploaded to the Moodle system prior to the course officially commences. If a student is a good student, he / she should prepare the resources which are required for his studies at the start of the semester. He has to download all of the necessary materials beforehand and it is not good to access all of the materials at the last minute. That would lead the students to face lack of time to go through all of the materials. Hence the research expects a good student's number of resources accessed in a week should have a similar shape of the left side image of Figure 3.27, even middle one is also acceptable. But most of the students have negatively skewed access pattern. The Table 3.9 shows the number of resources accessed by some students in each week of the semester who were selected for this research and the variation is plotted in the Figure 3.28.

Table 3-9 Some of the students' resource access volume

Week No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
User Id																
442	41	6	0	1	0	3	4	0	1	1	1	1	2	5	7	0
413	5	1	4	8	3	4	2	7	1	5	7	8	4	7	5	21
3951	45	19	4	0	3	4	5	3	0	3	0	6	0	5	0	20

3984	2	0	4	1	0	2	3	1	1	4	0	3	0	1	61	9
4820	0	4	2	1	0	4	2	2	2	0	0	0	0	3	59	37
3957	12	4	11	0	7	6	1	8	1	4	7	12	3	34	3	5

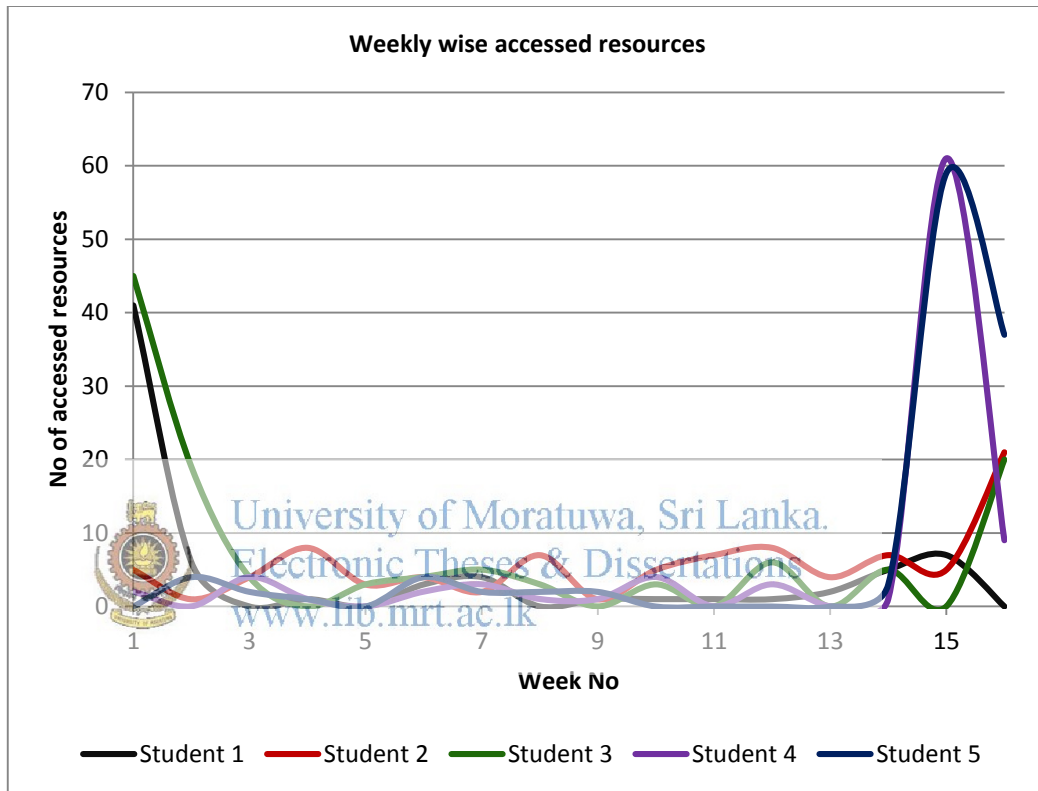


Figure 3-28 The skewness of access of some of the students selected for this research

Measures of skewness

It can be very subjective to decide which data set is more skewed than other data set by looking at the graph of the distribution. This is why there are ways to numerically calculate the measure of bias. There are a number of ways of measuring skewness. The research used the Microsoft Excel’s formula as displayed in Equation 3.9 to calculate the skewness of a number of resource accesses in a week data distribution. The SKEW function uses the following formula to calculate the skewness.

$$\gamma = \frac{n}{(n-1)(n-2)} \sum \left(\frac{x_i - \mu}{\sigma} \right)^3$$

Equation 3-9 Microsoft Excel's skew function

γ – The skewness of the distribution


x_i – A data point in the distribution

μ - Mean of the distribution

σ - Standard deviation of the distribution

n - The number of data points in the distribution

The data from the StudentResourceLogPivot table which is shown in the Figure 3.29 was queried and it was stored in a excel sheet and the skew function was applied. Then that excel sheet was imported into a table called AccessResourceSkewness. A stored procedure could have written to calculate the skewness of the number of resources accessed within a week data distribution, but to ease the workload, Excel was chosen.



CourseId	UserId	NoOfAccessWk1	NoOfAccessWk2	NoOfAccessWk3	NoOfAccessWk4	NoOfAccessWk5	NoOfAccessWk6
11	23	0	4	0	0	3	5
11	24	0	8	0	0	0	0
11	25	0	1	3	0	0	0
11	26	4	5	0	0	0	2
11	27	2	5	0	0	0	0
11	28	3	5	7	0	9	0
11	29	0	8	0	0	0	0
11	32	21	10	2	0	0	0
11	33	18	3	1	3	0	0
11	34	6	8	4	0	0	31

Figure 3-29 The selection of StudentResourceLogPivot table

The table AccessResourceSkewness was populated with the CourseId, UserId and the skewness of number of accessed resources. Then the skewness of a student which is populated in the AccessResourceSkewness table was updated in the Students table with the respective student id. The SQL snippet shown in Figure 3.30 illustrates the stored

procedure that was used to update the skewness of a student which is extracted from the AccessResourceSkewness table.

```
FETCH NEXT FROM @getStudents INTO @CourseId, @UserId
WHILE @@FETCH_STATUS = 0
UPDATE [Moodle].[dbo].[Students]
SET [SkewOfAccessResource] = (SELECT [Skew]
                              FROM [Moodle].[dbo].[AccessResourceSkewness]
                              WHERE CourseId = @CourseId AND UserId = @UserId)
WHERE CourseId = @CourseId AND UserId = @UserId
```

Figure 3-30 Update the skewness of student in the Students table

Hence the skewness of a number of resources accessed in a week of student also taken into the process of building the student classification model as one of the attributes.

3.2.3 Update the good students in the student table

The students are categorized as good and average students based on the model building process described in section 3.5.2.1. The model reveals a rule based on decision tree that was learnt during the training phase as shown in Figure 4.8. The rule was implemented as a SQL procedure to categorize students as good and average students. The Figure 3.31 exhibits part of the stored procedure which does the job.


```
SELECT @AvgRatio = AvgRatio, @PercentOfWiki = PercentageOfWikiEntries,
@PercentOfAssignCompletion = PercentageOfAssignCompletion,
@PercentOfLateAccess = PercentageOfLateAccess,
@PercentOfOpenedDiscussion = PercentageOfOpenedDiscussion,
@PercentOfVisitOfStudent = PercentageOfVisitOfStudent,
@SkewOfAccessedResource = SkewOfAccessResource
FROM [Moodle].[dbo].[Students] WHERE UserId = @UserId AND CourseId = @CourseId
/*Continue in next page*/
```



```

IF (@AvgRatio > 1.034)
  IF( @PercentOfWiki > 5)
    SET @StudentType = 'Good Student'
  ELSE
    IF( @PercentOfAssignCompletion > 70.5)
      IF (@PercentOfWiki > 0.5)
        IF (@AvgRatio > 1.042)
          IF (@PercentOfOpenedDiscussion > 3)
            SET @StudentType = 'Avg Student'
          ELSE
            IF (@PercentOfLateAccess > 42.5)
              IF (@AvgRatio > 1.057)
                IF (@AvgRatio > 1.061)
                  IF (@AvgRatio > 1.080)
                    SET @StudentType = 'Good Student'

```



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

/* Continue */

Figure 3-31 Classify the students based on the rule learnt by the best model

3.2.4 The computed attributes of resource recommendation model

The ultimate goal of this research project is to discover a learning material recommendation model based on the heaviness of their access by the good students. The research highly assumes the good students are familiar with good resources or learning materials and they have the ability to distinguish the good materials and those that are not usable. There are lots of materials uploaded to the learning management systems such as Moodle.

- ✓ Subjective and dense materials
- ✓ Subjective and non-dense materials
- ✓ Subordinate materials

Some of the materials align with the subjects taught in the classes and they are heavy dense towards the course syllabus. If the students study these dense materials it would be more helpful for them towards their final examinations. Also some are less dense, but still they are subjective. They have the contents which are expected in the course syllabus, but the explanation would be more. The students need to spend more time to get the extract out of it. There is another type of learning materials hosted in the learning management system that are subordinate materials. They are helpful to understand the subject, but rarely required in the examination perspective. They facilitate students to get deeper and wider knowledge on the subject. Sometimes they are recommended for reading purposes only.

The course main page is divided into sections in the middle of the page. The section is a set of resources grouped together on the course main page. Sometimes a section may consist of resources of a single week. If so the course may contain the sections that is equal to the number of weeks of the course duration. Sometimes the section may form by grouping the resources based on the subjects. If it is grouped so, a course may have less than the number of weeks of a semester duration.



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The bright students may identify the good materials by looking at the title of the resources based on their knowledge and the experience. Brighter students may have the ability to identify the materials more suited for their studies and what are the subordinate materials. Hence, they may know which resources should be chosen for the studies and what are the most preferred resources. But it is not always possible for the average students. Hence there is a requirement for the recommendation model to suggest the learning materials mostly accessed by the good students to the average students.

Similar to student model, the resource also did not use all of the attributes that were derived and calculated. The ModResource table contained 38 attributes and 13 were directly fetched from the moodle.xml and the resource folder of the courses the rest 25 attributes were calculated from other attributes of the same table as well as other tables.

Some of the important attributes that contribute towards the resource recommendation model are discussed in the sub section below.

Percentage of accesses of a resource

All of the access of each resource can be audited by the Moodle. The total number of accesses of all of the resources of a course by entire students population can be calculated. And also the number of times a single resource accessed by all of the students can also be derived. More number of accesses of a single resource would indicate that as highly accessed resource. Based on the above two figures the percentage of access of a resource is derived as shown in the Equation 3.10.

A stored procedure moodle_resources_percentage_of_accessed_by_good_students.sql was used to do the logic to calculate the percentage of access of a resource is shown in Figure 3.32.

```

SET @TotalNumberOfResourceAccessInCourse =
(SELECT COUNT(*)
FROM (SELECT DISTINCT logUserId, logInfo
FROM [Moodle].[dbo].[log]
WHERE logModule = 'resource' AND logAction = 'view' AND
CourseId = @CourseId) AS X)

SET @numberOfAccessOfSingleResource =
(SELECT COUNT(*)
FROM ( SELECT DISTINCT logUserId, logInfo
FROM [Moodle].[dbo].[log]
WHERE logModule = 'resource' AND logAction = 'view' AND
CourseId = @CourseId AND logInfo = @ResourceId) AS X)

SET @percentageOfAccessOfSingleResource = @numberOfAccessOfSingleResource *
100 / CAST(@TotalNumberOfResourceAccessInCourse AS DECIMAL(16,2))

```

Figure 3-32 The percentage of access of a single resource

$$\text{Percentage of access of a resource} = \frac{\text{number of access of a resource by all students}}{\text{number of access of all resources by all students}}$$

Equation 3-10 The percentage of access of a resource

The percentage of accesses of resource by good students

Similar to the percentage of access of a resource, the percentage of access of a resource by good students is also calculated as per Equation 3.11. The percentage of access of a resource by good students is calculated by the ratio between the number of the access of this resource by good students and the total number of access of all resources by all good students. In other words, first the total number of accesses of a resource by all good students within that course is computed. Later the total count of access of all of the resources by the good students within a given course is computed. Based on these two figures the percentage of access of a resource by good students is derived as shown in Figure 3.33. This will be used later to label the resources. In essence, this attribute will be used as the labeling attribute to categorize the resources as highly accessed and less

```
SET @TotalNumberOfResourceAccessInCourseByGoodStudents =
(SELECT COUNT(*)
FROM ( SELECT DISTINCT logUserId, logInfo
FROM [Moodle].[dbo].[log]
WHERE logModule = 'resource' AND logAction = 'view' AND
CourseId = @CourseId AND logUserId IN
(SELECT DISTINCT UserId
FROM [Moodle].[dbo].[Students]
WHERE CourseId = @CourseId AND CHARINDEX('Good Student',
Student) > 0)) AS X)
SET @numberOfAccessOfSingleResourceByGoodStudents =
(SELECT COUNT(*)
FROM ( SELECT DISTINCT logUserId, logInfo
FROM [Moodle].[dbo].[log]
WHERE logModule = 'resource' AND logAction = 'view' AND
CourseId = @CourseId AND logInfo = @ResourceId AND
logUserId IN (SELECT DISTINCT UserId
FROM [Moodle].[dbo].[Students]
WHERE CourseId = @CourseId AND
CHARINDEX('Good Student', Student) > 0)) AS X)
SET @percentageOfAccessOfSingleResourceByGoodStudents =
@numberOfAccessOfSingleResourceByGoodStudents * 100 /
CAST(@TotalNumberOfResourceAccessInCourseByGoodStudents AS DECIMAL(16,2))
```

Figure 3-33 Percentage of access of a resource by good students

The stored procedure first collects the access of a resource and then within that access, the number of accesses by the good students is determined. Similarly then total access of all resources is calculated.

$$\text{Percentage of access of a resource by good students} = \text{number of access of a resource by good students} / \text{number of access of all resources in a course by good students} * 100$$

Equation 3-11 The percentage of access of a resource by good students

Percentage of accesses of a resource by average students

Similar to the percentage of access of a resource by good students, the percentage of access of a resource by average students is also calculated as presented in Equation 3.12. Instead of good students, the access by average students is separately calculated from the total access. The access by the average is separated from the access by good students and the average students as if the access of all students, it is hard to discriminate the impact of access by good or average students separately. Similar to Figure – 3.33 instead of good students, the average students are computed.

$$\text{Percentage of access of a resource by average students} = \text{number of access of a resource by average students} / \text{number of access of all resources in a course by average students} * 100$$

Equation 3-12 The percentage of access of a resource by average students

Week level relative position

The week level relative position is used to measure the position of the resources within the week. In correct terms, this is the section level relative position. The resources which are at the start of the section / week are having more probable to get access. As they are at the top of the week/ section their number of access would be more. The resources which are at the latter part of the section/ week have less probable to have more access. Not all of the sections having equal number of resources. Hence, if the research considers just the position of the resource in a week, then that would be a biased data set. Hence the research seeks to derive the weekly relative position of the resources

within the week/section (e.g. 3rd resource in the section). The relative position of the week is calculated by getting the position of the resource within the week and count the number of resources within the week. Based on that the weekly relative position is evaluated. Its highest value is 1, when the resource situates at the last position in the week/section. None of the resources' week level relative position would not be zero and all will be in the range of 0 to 1.

Previously, there was a stored procedure moodle_resource_position_in_section.sql was used to derive the number of resources in week, position of the week/section within the course. The Figure 3.34 demonstrates the calculation of the number of resources in a week and the position of the week and based in that it derives the week level position.

```

SET @PositionOfResourceInSection =
(SELECT PositionInWeek
FROM (SELECT Instance
      FROM [Moodle].[dbo].[Sections]
      WHERE CourseId = @CourseId AND Instance = @ResourceId) P INNER JOIN
      (SELECT Instance, COUNT(*) OVER (ORDER BY ModId) AS PositionInWeek
      FROM [Moodle].[dbo].[Sections]
      WHERE CourseId = @CourseId AND ModType = 'resource' AND
      SectionId IN (SELECT DISTINCT SectionId
                   FROM [Moodle].[dbo].[Sections]
                   WHERE Instance = @ResourceId)) Q ON P.Instance = Q.Instance )

SET @NoOfResourcesInSection =
(SELECT MAX(PositionInWeek)
FROM #TmpSectionResources
WHERE CourseId=@CourseId AND
SectionId IN (SELECT DISTINCT SectionId
             FROM [Moodle].[dbo].[Sections]
             WHERE Instance = @ResourceId))

```

Figure 3-34 The position of resource in week

Course level relative position

Similar to the week level relative position the attribute course level relative position, calculates the relative position of the resources within the entire course. The resources

which are at the start of the course have more access probability, whereas the resources on the latter part has less access. The position of the resource within the course impacts the number of accesses. Hence this attribute is evaluated. Similar to the week level relative position, this also has the maximum of 1 and lies between 0 and 1. The Figure 3.35 calculates the position of the section / week in the course and the number of sections/ week in the course.

```

WHILE (@i <= (SELECT COUNT(*) FROM [Moodle].[dbo].[Course]))
BEGIN
SET @Course_Id = (SELECT CourseId
                  FROM (SELECT CourseId, ROW_NUMBER() OVER(ORDER BY
                  (SELECT (1))) AS RowNo
                  FROM [Moodle].[dbo].[Course] ) X
                  WHERE RowNo = @i )
INSERT INTO #TmpSections(CourseId, SectionId, WeekNo)
(SELECT CourseId, SectionId, ROW_NUMBER() OVER(ORDER BY (SELECT (1))) AS
WeekNo
FROM (SELECT DISTINCT SectionId, CourseId
      FROM [Moodle].[dbo].[Sections]
      WHERE CourseId = @Course_Id )
      SET @i = @i + 1
END
SET @PositionOfSectionInCourse =
(SELECT WeekNo
FROM (SELECT SectionId
      FROM [Moodle].[dbo].[Sections]
      WHERE CourseId = @CourseId AND Instance = @ResourceId) X INNER JOIN
(SELECT SectionId, ROW_NUMBER() OVER(ORDER BY (SELECT (1))) AS WeekNo
FROM (SELECT DISTINCT SectionId, CourseId
      FROM [Moodle].[dbo].[Sections]) Y
WHERE CourseId = @CourseId ) Z ON X.SectionId = Z.SectionId )

```

Figure 3-35 The position of section in the course

The percentage of good students accessed this resource

This look similar to the percentage of access of a resource by good students. It calculates the ratio between the number of accesses of a resource by good students and

the number of good students in the course as in Equation 3.13. In simple terms the attribute try to identify, from all of the good students of the course, how many good students accessed this resource. Though it looks like the same as percentage of access of resource by good students, both are different. The dividend number of accesses of a resource by good students is same for both attributes. The divisor is different in each case. Its divisor is the total number of accesses of all resources by good students and in this case the divisor is the number of good students within the course. The Figure 3.36 displays the calculation of percentage of good students accessed this resource.

```

SET @noOfGoodStudentInCourse      =
(SELECT COUNT(*)
FROM [Moodle].[dbo].[Students]
WHERE CourseId = @CourseId AND
CHARINDEX('Good Student', StudentType) > 0)

SET @noOfGoodStudentAccessedThisResource      =
(SELECT COUNT(DISTINCT logUserId
FROM [Moodle].[dbo].[log]
WHERE CourseId= @CourseId AND logInfo = @ResourceId AND logModule = 'resource'
AND logUserId IN (SELECT DISTINCT UserId
FROM [Moodle].[dbo].[Students]
WHERE CourseId = @CourseId AND
CHARINDEX('Good Student', StudentType) > 0))

```

Figure 3-36 Percentage of good students accessed a resource

Percentage of access by good resources = number of good students accessed this resource / number of good students in the course.

Equation 3-13 The percentage of access by good students

File size

The file size of the resources was derived from the resources folder of the course back up and it was updated in the table. But it was appended with “KB” at last. The KB has to

be removed and only numerical figures have to be used in the data mining models. There was a small substring function was used using SQL.

3.3 Reduce Data

The second phase (Transform data) applied data extraction techniques to the data set produced in the first stage to transform it as the data set that requires modeling. Most of the learning processes of the machine learning (in which data mining algorithms are based typically) are characterized by high-dimensional data. Accuracy of consultations and generally, efficiency degrades rapidly with increasing dimension as the number of variables and parameters grow, the number of data samples required to estimate the variables and parameters grow exponentially. This problem is sometimes referred to as the "curse" of dimensionality [15]. Therefore, for a given sample size, there is a maximum number of variables and parameters that can be accurately estimated.

In addition, not all of these dimensions are relevant and some are redundant. There are two main approaches to reduce the dimensionality, feature selection and feature transformation to a low dimensional space. Feature selection is concerned with the selection of a subset of the original characteristics, the original representation of the characteristics of the data set is not changed. In contrast, the feature transformation methods based on modifying the input characteristics.

The research also applied number of data reduction techniques to prepare the data set to build the models. The research aimed to build two models. The student and resource model were derived as per the explanation given in the section 3.5 building models. The data for the student classification model was derived from the Student table where as the ModResource table fed the Resource recommendation model. The Student table had 84 attributes and ModResource populated with 44 properties. All of the 84 attributes of the Student table cannot be used to build up the Student classification model as that would lead to a less accuracy model. And also not all of the attributes would play a key role in model building. Even though they are removed, that may not impact the model. And some of the attributes may be redundant and depend on other attributes. Hence the

attributes which do not impact the model should be removed and the attributes which contribute to the model should be included. The research calculates the weight of each attribute that contributes to the model. And the attributes which have less weight can be neglected from the model building process.

3.3.1 **RapidMiner**

RapidMiner is a software platform that provides an integrated environment for machine learning, data mining, text mining, predictive analytics and business intelligence [33]. It is used for commercial and industrial applications as well as research, education, training, rapid prototyping, and application development and support all steps of the data mining process including results visualization, validation and optimization. RapidMiner provides data mining and machine learning procedures including data loading and transformation (Extract, Transform, Load (ETL), data preprocessing and visualization, predictive analytics and statistical modeling, evaluation and deployment. RapidMiner is written in the Java programming language and it provides a GUI for designing and implementing analytical workflows. These workflows are called "process" and consist of several "operators" and each operator performs a single task in the process and results of each operator forms the input to the next one [33].

3.3.2 **The reason for RapidMiner**

Even though there are lots of data mining tools such as Weka which is one of the oldest data mining tools available in the market, the research chose RapidMiner. The Weka cannot evolve when the size of the data set gets increased. It may throw out of memory exceptions when it cannot handle the larger data set. But the RapidMiner is unquestionably the world-leading open-source system for data mining, which can withstand for larger data sets with higher performance [33]. Thousands of applications of RapidMiner in more than 40 countries give their users a competitive edge. RapidMiner is easily the most powerful and intuitive graphical user interface for the design of analysis processes [33].

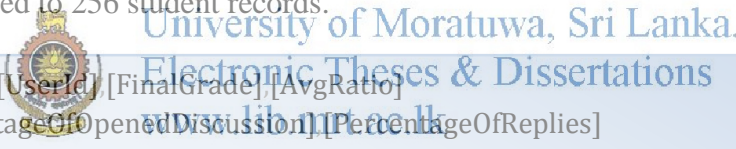
3.3.3 Data reduction for student classification model

The research chose 10 attributes as the building blocks for the Student classification model which were extracted from initial 84 attributes. A process was set up using the RapidMiner tool to identify the impactness of each of the attributes towards building up the Student classification model.

The Student classification model was built using the chosen attributes. At the beginning the research selected the clustering techniques to group the students into either good or average. Clustering can be considered the most important unsupervised learning problem, so, like any other problem of this kind, it deals with finding a structure in a collection of unlabeled data. A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters [28]. Initially the research doesn't know about the students whether they are brighter or not less intelligence prior to do the data analysis. In other words the students cannot be labeled. As a result, it was thought that the clustering could be the suitable data mining principle which can be applied to the student data set. The research also analyzed the data with different clustering algorithms by configuring different parameters of the algorithms. Based on that, it derived the models which supposed to be more accurate in predicting. But when the model was built up using that data set the derived clustering models resulted with less accurate results. The data set failed to output highly accurate results even for good clustering algorithms.

Hence it was decided to get the final grades of the students and based on the final grades label the students and later use the classification algorithms to predict the students. Classification is the process of finding a model for class attribute as a function of the values of other attributes [29]. The classification algorithms classify data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data.

Initially the data set contained 343 of student records and it shrunk to 256 after the removal of drop offs, non-graded for assignments and not having final grade students. As a result the data set needs to be cleaned and some of the outliers and the missing values to be removed to deliver acceptable results by the models. Outlier detection is one of the most important tasks in the data analysis. The approach finds how much the outlier data are away from their average values [30]. The threshold that separates abnormal and normal data numerically often are the basis for important decisions. All of the data points may look that they are away from the centroid of the cluster when the data set is zoomed in. So the threshold value has to be decided by looking at the data and based on the requirement. This research also removed some of the data points which deviated much from the average set of data by over looking at the clusters. A stored procedure student-avg-gt-than-0.5-removal.sql was used to remove the data set which deviates too much from average data set. The Figure 3.37 removes the outliers from the initial data which contained 343 student records. Following that the data set was reduced to 256 student records.



```

SELECT [UserId],[FinalGrade],[AvgRatio]
,[PercentageOfOpenedDiscussion],[PercentageOfReplies]
,[PercentageOfWikiEntries],[PercentageOfAssignCompletion]
,[PercentageOfVisitOfStudent],[PercentageOfAccessedResource]
,[PercentageOfLateAccess],[SkewOfAccessResource]
FROM [Moodle].[dbo].[Students]
WHERE FinalGrade IS NOT NULL AND IsDropped = 0 AND CAST(AvgRatio AS
DECIMAL(14,4))BETWEEN 0.8 AND 1.2 AND
CAST(PercentageOfOpenedDiscussion AS DECIMAL(14,4)) <= 10 AND
CAST(PercentageOfWikiEntries AS DECIMAL(14,4)) < 10 AND
CAST(PercentageOfOpenedDiscussion AS DECIMAL(14,4)) <= 10 AND
CAST(PercentageOfAssignCompletion AS DECIMAL(14,4)) > 40 AND
CAST(PercentageOfVisitOfStudent AS DECIMAL(14,4)) < 10 AND
CAST(PercentageOfAccessedResource AS DECIMAL(14,4)) > 40 AND
CAST(PercentageOfLateAccess AS DECIMAL(14,4)) > 10 AND
CAST(SkewOfAccessResource AS DECIMAL(14,4)) > 1

```

Figure 3-37 The outliers removed from Student data set

After the outlier detection and the removal the research resulted with improved accuracy models. Hence, once again it was decided to remove the outliers from the current data set to improve the accuracy further. As illustrated earlier, outlier detection can be continued until the last data point also considered as outlier. That's why the outliers should be removed carefully by looking at the data rather blindly removing the boundary data. The second time the student data further removed some more outliers and the data set got reduced from 256 to 186 records. The model building process recursively done by slightly changing the data set by outlier removal and other modifications to the data set.

Second time the accuracy improved and that let the research go forward to the third phase of outlier detection and removal. The third outlier detection removal removed 44 records further and the student data set shrunk to 141 of rows of students. But the third phase of outlier detection and removal did not provide good result as expected. The accuracy was less than the accuracy that received in the phase two of outlier detection and removal. Hence the recursive process of outlier detection and removal was halted and the data set which was used in the phase two of outlier detection was chosen as the data set for the student modeling. That was tested with different classification algorithms with a variety of parameter sets to find out the best student classification model.

The students' activity based data is extracted as relation or proportion rather than as an absolute value. The goal is to avoid variability introduced by differences in the criteria for evaluating workload and course imposed different faculty or determined by the characteristics of the course [3]. For example, instead record the number of assignments completed by the student, it was decided to record the percentage of assignments completed by a student, calculated as the ratio between the number of assignments performed by the student and the number of assignments given during the course.

3.4 Partition data

In Phase Four (data partitioning), the input data are divided into two subgroups, the first one is a set of training data, and the second one is the validation data set. The training data set is used to build the models. Once the models are trained from the training data set that needs to be validated with the unseen data. A validation subset is used for this purpose, in which the actual value of a class variable is known and can be used to test the accuracy of the candidate models [3]. The validation data set is often used to adjust the model building process and select between algorithms and architecture competition (different configurations of parameters) in a particular machine learning algorithm.

Partitioning the data into two distinct subgroups is an ideal situation may be limited if there is limited available input data, since a small amount of input data has a negative effect on the accuracy of the model [1]. Different heuristics have been proposed for determining the minimum amount of data required to train accurate models. Shmueli [31] suggests at least ten samples per variable predictor data as a rule of thumb. Others [32] have proposed that for classification tasks, a minimum of $6 \times p \times m$ sample data is required to do the modeling where p is the number of features and m is the number of class values. Limited data partitions must be replaced by other cross-validation methods (for example n -fold). In this research, the volume data obtained is neither too small nor too big, which does not provide problems in the production of sufficient data to train, validate and test the models. The partition data in each model building process will be explained in the Building Model section.

Tracking records and information at a low generic level to make sense of these data, it is necessary to formulate some queries to obtain aggregate results [26] (“percentage of visits”, “average ratio “). This type of information is sometimes called quantitative analysis [27].

3.5 Building Models

In any business, the studies or data analysis is done to get to find patterns of behavior, predict the outcomes and later turning them into profitable opportunities. In the past, this study of the patterns is severely limited by the amount of human effort involved, and the expense of collecting the necessary data. In recent years, the balance changed. More data are readily available to most businesses than ever before. More computing power is available to process these data, and automated techniques that can be used to find patterns in data with limited human intervention has developed and matured. The data mining technology is one of the techniques to build up models. Data Mining is the art and science of discovering and exploiting new, useful, and profitable relationships in data. The most accurate models have to be chosen among several other data models based on the artistic and scientific thinking. The research too has chosen data mining principle to build up the models. The requirement is to build up models to identify and predict the good students and based on the good students' access pattern suggest the materials used by the good students to average students. Hence, it is required to build up two data models, one is to identify the good students and second one is to suggest the learning resources used by the good students to others.



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

3.5.1 Why classification

Data mining algorithms can follow three different learning approaches: supervised, unsupervised or semi - supervised. In supervised learning algorithm works with a set of examples that labels are known. Its labels can be nominal values in the case of the classification task or numeric values in case of regression task. In unsupervised learning, in contrast, the labels of examples in the dataset are unknown, and algorithm usually aims at grouping examples according to the similarity of their attribute values , characterize a grouping task. Finally, half- supervised learning is usually used when a small subset of labeled examples is available, along with a wide range of unlabeled examples.

The task classification can be seen as a guided technique in which each instance belongs to a class, which is indicated by the value of a special target attribute or simply

the class attributes. Target attribute can take on categorical values each of them corresponds to a class. Each entry of data set consists of two parts, namely a set predictor attribute values and a target attribute value. The former is used to predict the value of the latter. During the classification task, the set of entries in the data set are extracted is divided into two mutually exclusive and exhaustive set, called training set and testing set. Classification model is built from the training set, and during the testing, the model is evaluated using the test data set. In the training phase the algorithm has access the values of both predictor attributes and the target attribute for the sample training set, and it uses this information to build a classification model. This model represents the classification knowledge, in essence, a relationship between predictor attribute values and classes that allow prediction of the class of an example given its predictor attributes values. The knowledge discovered by a classification algorithm expressed in many different ways as rules, decision trees, Bayesian networks, etc. Various techniques for classification are explained in the following section. The student data set comprises 84 attributes and the final grade as the class attribute. So Based on that the students can be classified and later the new students can be predicted based on the model that learnt during the student modeling process. The final marks of the students are not available and only the final grades, which were released such as “A+”, “A”,” A-“exist.

The classification algorithms

A brief description of each of the classification algorithms that are used in the student classification model are given below.

Decision tree

A decision tree classification comprises a decision tree generated on the basis of examples. It has two types of nodes [23].

- a) The root and the internal nodes,
- b) The leaf nodes.

The root and the internal nodes are associated with attributes and the leaf nodes are associated with classes. Basically, each non-leaf node has an outgoing branch for each possible value for the attribute associated with the node. To determine the class of a new instance using a decision tree, beginning with the root, subsequent internal nodes are visited until a leaf node is reached. At the root node and each internal node is a test used.

Neural net

An artificial neural network (ANN), usually called neural network (NN), is a mathematical model or computational model that is inspired by the structural and functional aspects of biological neural networks. A network consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation.

Support vector machine (SVM)

The Support Vector Machine constructs a hyper plane or set of hyper planes in a high or infinite-dimensional space which can be used for classification, regression or other tasks. A good separation of hyper planes is achieved by the hyper plane that has the largest distance to the nearest training data points of any class or functional margin, since in general larger margin the lower the generalization error of the classifier.

Naive Bayesian

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes ' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". The advantage of the Naive Bayes classifier is that it requires only a

small number of training data to estimate the means and variances of the variables required for classification.

Rule induction

The rule induction algorithm starts with the less prevalent classes, and then iteratively grows and prunes rules until there are no positive examples left. Rule sets have the advantage that they are easy to understand, representable in first-order logic and having the major drawbacks that they scales poorly with the training set size and has the problems with noisy data.

Linear regression

Regression is a technique used for numerical prediction. Regression is a statistical measure that attempts to determine the strength of the relationship between one dependent variable and the independent variables. Linear regression attempts to model the relationship between a scalar variable and one or more explanatory variables by fitting a linear equation to observed data.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

K-nearest neighbour (k-NN)

K-nearest neighbor algorithm is based on learning by analogy, by comparing a given test example with the training examples that are similar to it. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms. It is classified by a majority of its neighbors, the sample is assigned to the class most common among its k nearest neighbours.

Perceptron

The perceptron is a type of artificial neural network, which is an algorithm for supervised classification of an input in one of the several possible non-binary outputs.

Vector linear regression

The vector linear regression applies the regression on all of the regular attributes upon a vector of labels.

Polynomial regression

Polynomial regression is a form of linear regression, wherein the relationship between the independent variable X and the dependent variable Y is modeled as an n th order polynomial. The polynomial regression models usually fit using the method of least squares whose least-squares method minimizes the variance.

Gaussian process

A Gaussian process is a stochastic process that realizations consist of random values associated with each point in a range of space, so that each such random variable has a normal distribution. It is a powerful non-parametric machine learning technique to construct comprehensive probability models of real world problems.

3.5.2 Student classification model

The student classification model was used to identify the good students whose access pattern are required to suggest the resources to the average students. Hence all the data related to the students collected, extracted, derived from the Moodle backup data. Initially the research received 343 records of students' grade to do the analysis. But later some of them were removed to improve the accuracy of the data models that were deviate much from the mean data set as they were considered as outliers. And some of the data which are noises, duplicates and the incomplete data also removed. The data in the student table were used to build up the student classification model. But the data in the student table were derived from several other source tables. Usually the transaction data spread across several tables in order to support fast insertion, update, deletion, selection operations. However, data mining algorithms require a single table that contains all the information relevant for the analysis and organize data particular level of

detail [26] (e.g. Student). This table is the result of a number of preparations via select, group, rotate or join operations plus a data cleaning and transformation step.

Step by step Student classification model building

The student classification model was built to classify the students as brilliant students and the average students. The data set which was prepared during the data processing phase is used to build up the model. The model building is not just straightforward execution of data mining algorithms fed by the data set. The final model was chosen after applying different classification algorithms, tuning the parameters of the algorithms and removing outliers from the data set recursively. The model which gives a more accurate prediction with the less error rate, less execution time and less resource usage would be selected as the best model. The resulting models were analyzed with different perspectives and the parameters were tuned to enhance the accuracy.

Split the data set

The dataset which was prepared in the data preprocessing step was used to build up the models. The classification models use the training data to build up the classification model and later the test data set will be used to validate the accuracy of the prediction of the classification model. The testing data and training data play a vital role in the model building process. The result would change as per the changes in the training and the testing data set. Hence, it is important to choose the right data set for the model building. Initially the data set was fragmented manually as testing data and training data. The data set can be partitioned using either of the following mechanisms.

- ✓ Manually
- ✓ Randomly

Manual fragmentation means, the data set was divided into training data set and test data with the interference of the researcher. The student data set from the courses Information Security (Course code - “0-12S3 CS5105ISec Information Security”) of the

MBA and the Computer and Network Security (Course code - 12S3 CS5404CNS Computer and Network Security) of M.Sc. were pulled off as the test data by the research. And the rest of the course except Computer Network Security (Course code - CS5404CNS-2009S2-Computer and Network Security) which does not have the final grades were picked out for the training data set.

In the random data set splitting methodology, 75% of the data set was divided as training data and the rest 25% were considered as test data which was later used to validate the model. The random data split the data without any intervention. And also it is less probable for the data set to get fragmented similarly two iterations. In other words, the data in the training set or the test data set would not be in the same set as it was in previously. The Figure 3.38 shows the fragmentation of the data set arbitrarily.

```

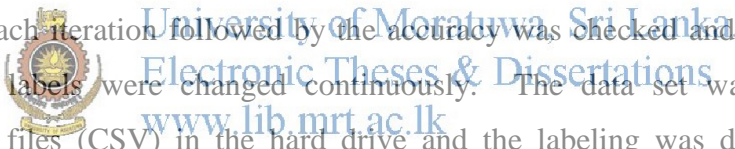
/*
Select 75% of the data for the training
These data is selected from the students whose Final grade is not
null and the not dropped students
*/
INSERT INTO #TmpUserIds(CourseId, UserId)
SELECT TOP 75 percent [CourseId], [UserId]
FROM [Moodle].[dbo].[Students]
WHERE FinalGrade IS NOT NULL AND IsDropped = 0 AND
CAST(AvgRatio AS DECIMAL(14,4)) BETWEEN 0.8 AND 1.2 AND
CAST(PercentageOfOpenedDiscussion AS DECIMAL(14,4)) <= 10 AND
CAST(PercentageOfWikiEntries AS DECIMAL(14,4)) < 10 AND
CAST(PercentageOfOpenedDiscussion AS DECIMAL(14,4)) <= 10 AND
CAST(PercentageOfAssignCompletion AS DECIMAL(14,4)) > 40 AND
CAST(PercentageOfVisitOfStudent AS DECIMAL(14,4)) < 10 AND
CAST(PercentageOfAccessedResource AS DECIMAL(14,4)) > 40 AND
CAST(PercentageOfLateAccess AS DECIMAL(14,4)) > 10 AND
CAST(SkewOfAccessResource AS DECIMAL(14,4)) > 1
ORDER BY NEWID()

```

Figure 3-38 Segregate the data set randomly

Label the students

The training data set is used to train the model in such a way that the model can learn from it and based on the learning it would build up the models. The prediction will be based on the intelligence, it is acquired in the learning phase. The classification algorithms require a class label as a target attribute on each row of data set. Based on the class labels, the algorithms would build up the models which later predict the result. Hence labeling plays vital role in the classification. If the classes are divided wrongly, it is harder to expect higher accurate results from a model. Initially the number of class labels of student classification model was chosen as two and as a result the all set of students was divided into either “Good Students” or “Average Students”. In the first instance, the students whoever got the final grade as “A+”, “A” and “A-“ were considered as “Good Students” and the rest who were not on this label boundary were labeled as “Average Students”. The labeling was not done in one iteration, rather it was a continuous process and the labels’ boundaries were changed slowly, and one change is done in each iteration followed by the accuracy was checked and based on the results the class labels were changed continuously. The data set was stored as comma separated files (CSV) in the hard drive and the labeling was done using Microsoft Excel’s “IF” function. The Figure 3.39 exhibits one of the IF functions of excel application that was used to label the students.



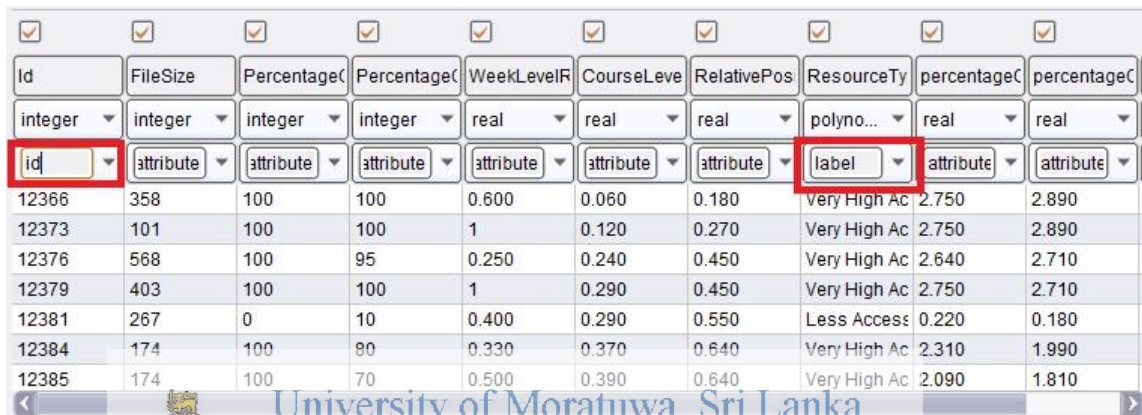
```
=IF (OR ((B2="A+"), (B2="A")), "Very Good", IF ((B2="A-"), "Good", IF (B2= "B+", "Average", IF (OR ((B2="B"), (B2="B-")), "Below Average", "Very Low"))))
```

Figure 3-39 Label good and average students

Prepare the data set

Once the splitting and the labeling was done, the next step was to prepare that data set as the input for the model building process. As stated earlier, the RapidMiner was used to build the models. It expects the input to be stored in the RapidMiner repositories. The data for the RapidMiner input data repository can be imported from a CSV, excel,

access, XML, access database, database table and the binary file. The research used CSV as the input source in most of the times as it was easy when there was a requirement to divide the data set often. While importing the CSV file into the RapidMiner repository, then it is required to choose the attribute which is to be used as the label. Otherwise, it may throw an error saying the “Classification algorithms require a label attribute”. The Figure 3.40 presents a RapidMiner window in the import CSV wizard, which selects an attribute as Id and another as a label and the rest as attribute.



University of Moratuwa, Sri Lanka.

Electronic Theses & Dissertations

www.lib.mrt.ac.lk

Figure 3-40 Chooses the Id and the label of input data

Set up the modeling process

The model building is the salient step in data mining problems. The models have to be built from the training data set and then the validity has to be checked using the test data. The RapidMiner is an easy, the most powerful and intuitive graphical user interface for the design of data analysis processes. An easy-to-use visual environment allows the users to recognize errors, apply quick fixes, and see quick and accurate results with absolutely no programming skills [33]. A RapidMiner classification process builds up the model using the training data set. The training data set should be big enough to build up the model so that the errors would be less. The Figure 3.41 illustrates a sample classification process using RapidMiner.

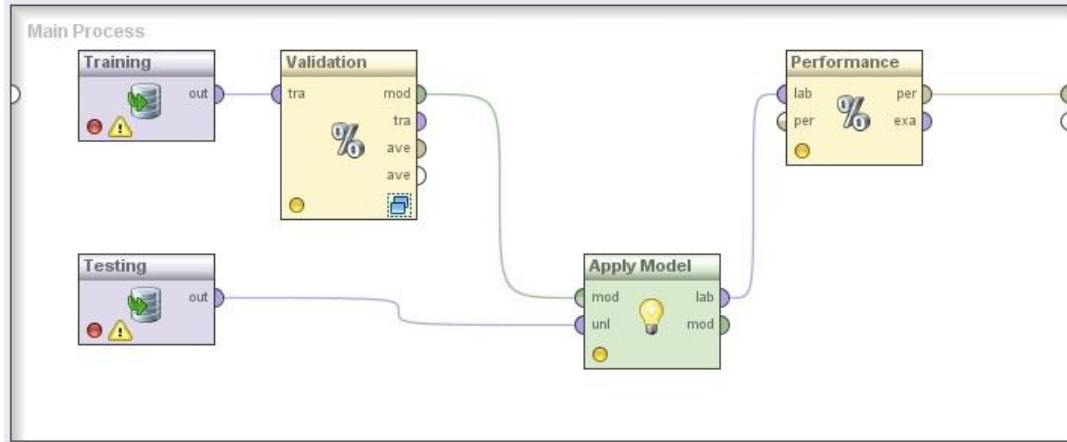


Figure 3-41 A classification process using RapidMiner

The Training and Testing blocks fetch the data from training and testing CSV files into the Validation operator. The validation operator is a nested operator. It has two sub processes within it and they handle training and testing inside the validation operator itself. The input training data set is divided into k subsets of equal size. At a time, only a subset is maintained as a set of test data, and the remaining $k - 1$ sub-subsets are used as a set of training data. Next, the process of cross-validation is repeated k times, with each of the k subsets used exactly once as the test data. The k results from k iterations results can be averaged to produce a single estimate of the model. Sometimes the model's accuracy may vary drastically change. Hence the models are derived by slicing the data set into k subsets and iterative over the k subsets. And the resultant model is a combination of all of the models derived in each of the iterations. This would optimize the model and when the model is tested with some independent set of data, it would give more consistent results than when it is modeled using single iteration. The value of k can be adjusted using the parameter number of validations. Larger the k results more consistent model. But it would also take more time to generate the model. Throughout the research the k was chosen as 10 which is the default value by RapidMiner.

After iterating k times the validation operator outputs a resultant classification model which is used as input to the "Apply Model". The resultant model of Validation operator is applied against the independent testing data set using a Apply Model

operator. This operator applies an already learnt or trained model on a trained data set. Then the already learnt model can be applied generally to other testing data set of prediction. All necessary parameters are stored in the model object at the learning or training phase. It is mandatory that both training data set and the testing data set must have exactly the same number, order and type of role attributes. The result of the Apply Model later passed to the classification performance operator which evaluates the accuracy, root mean squared error (RMSE), classification error and other performance related parameters. The performance operator yields the accuracy as the confusion matrix which outputs the overall accuracy of the model and the individual accuracy of each class label. The confusion matrix reports the number of false positives, false negatives, true positives, and true negatives. Based on this it derives the accuracy on each class label.

Obtain the result

The result of the modeling process would be outputted by the classification performance operator. Through it is capable of producing lists of performance criteria parameters the research was keen on collecting the accuracy which is derived from the confusion matrix and the RMSE. The Figure 3.42 reveals one of the confusion matrix derived during the research.

accuracy: 84.38%			
	true Average	true Good	class precision
pred. Average	22	1	95.65%
pred. Good	4	5	55.56%
class recall	84.62%	83.33%	

Figure 3-42 The confusion matrix to obtain the accuracy of the model

Adjust and tailor-make the model

It is not so easy to get the optimized model at first or second modeling process. The optimized model can be acquired by subsequent iterative modeling processes. In each

time the model designs, parameters and data sets were changed slightly, based on that the resultant models were compared with other results and the best model was chosen among all other models. The model might be picked as best based on the accuracy, execution time and resource usage. In this research several techniques were also used to get the optimized student model. Those are discussed in the following sections.

Determine the data fragmentation

The training data set and the testing data set which were fetched for the modeling process were chopped randomly and manually. In the manual data split the students who enroll for the courses Information Security (Course code - 0-12S3 CS5105ISec Information Security) of MBA 2012 batch and the Computer and Network Security (Course code - 12S3 CS5404CNS Computer and Network Security) of M.Sc. 2012 were considered as testing data set. And the rest of the students who enroll in other courses selected for the research other than Computer Network Security (Course code- CS5404CNS-2009S2-Computer and Network Security) of M.Sc. 2009 batch was included into the training data set. And when the data were fetched into the model it eliminated the students who do not have the final grade or average assignment ratio less than 0.5 as the research assumes the students who got the average less than 0.5 as outliers from the data set. Initially the training data set consists 205 records and the testing comprises 51 student records.

And in another scenario the students were randomly divided into training data set and testing data set. At the beginning data set was freed from the students who got the average ratio below 0.5. And that intermediate data set was randomly divided into training and testing data. The seventy five percentages of the data assigned to the training data set and the rest 25% included in the test data. During the randomized data splitting, the training data set was filled with 192 records and testing data set composed with 64 records. Altogether 256 students have their average ratio above 0.5. In the manual process the training data set and testing data set were made up with 205 and 51 respectively and altogether it was 256 student records.

The models always over fit to the data which was used to train them. In other words the model learns from the training data set and it adjusts its rules and other parameters according to the training data set. In essence, the model depends on the training data. Hence, it is very important to supply the well prepared training and testing data sets. That's the reason the research experiments with the data set which are cropped manually as well as randomly. The results will be analyzed in the "Evaluate and Choose Models" subsection, but they do not exhibit considerable discrimination in the accuracy. The results are nearly equal in both manual and randomly split data where the results derived from the randomly fragmented data is a bit high in accuracy. Though it is possible to argue that it is not always possible to get a more accurate result in randomized split. Yes, it is true that since the randomized split provided more accurate result in this iteration does not mean it would yield more accurate model in the next iteration. It may output less accurate or even more accurate model in the next iteration. But research chooses this randomized split for its model building and progresses to the next step.

Change the algorithms  University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations

In the first instance the models were built using the decision tree as the classification algorithm. But other algorithms also experimented while the data set and other process related parameters remain same. The neural networks (neural net), support vector machines (SVM), k-nearest neighbor (k-NN), Naïve Basis, Perceptron, Rule Induction, Linear Regression, Vector Linear Regression, Polynomial Regression and Gaussian process were used as the classification algorithms in model building phase. The accuracy was measured in each algorithm and whenever the parameters of the algorithms get tuned and the input data set changes. The classification algorithms inserted into the sub process of validation operator as shown in Figure 3.43.

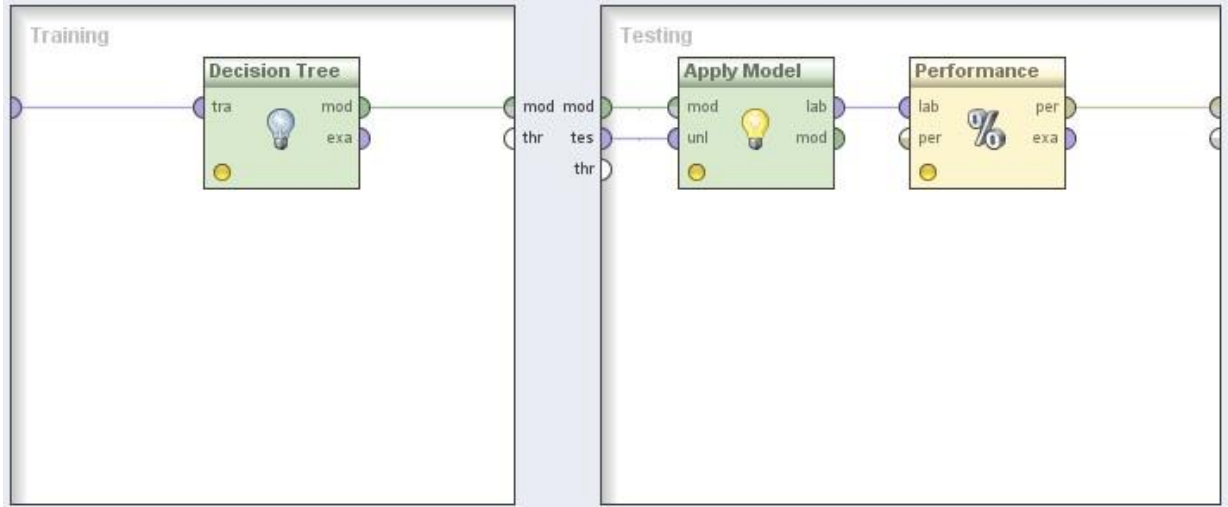


Figure 3-43 The classification algorithms, the core of validation operator

The decision tree operator can be replaced by some other classification algorithms in the next instance to compute the accuracy of that algorithm.

Change the class label boundaries

The final grades were used as the label attribute in the data sets that were used in the classification algorithms. The students were categorized as “Good Students” and “Average Students” based on the final grade when the number of class label was considered as two. When the number of class variables was increased to three the students were classified as “Very Good Students”, “Good Students” and “Average Students”. Similarly the students were grouped as “Very Good Students”, “Good Students”, “Average Students” and “Below Average” when the number of class variables switched to four. At the end the students were labeled as “Very Good Students”, “Good Students”, “Average Students”, “Below Average” and “Very Low” by increasing the class variables transitioned to five.

In all the cases there were no clue to define the boundary of each of the class labels for the researcher. In other words it is hard to determine the boundary of the final grade, which distinct out the “Good Students” and “Average Students” when the class variable

equal to two. The research was initiated, with some assumed boundary value and calculated its accuracy. And later, the boundary values were slightly shifted up and down to verify the boundary value of class label impact the result or not. If the accuracy increases when the value changes, the research assumes its good point. And that was continued till the accuracy goes down. The best splitting point for the label was the one which out-turns more accuracy.

At the start the research assumed boundary of “Good Students” and “Average Students” as A-, B+ border. In essence the students whoever got A-, A and A+ was categorized as Good Students and the students got below A- such as B+, B, B-, C+, C, C-, I, D and F were judged as Average Students. And the modeling process was set and accuracy of the modeling was recorded. In the next phase the research believed the students who obtained A+, A, A- and B+ as good students and the rest were Average Students. When the accuracy was observed this time the accuracy moved down. Hence the research branded A+ and A students as Good Students and the rest were pushed to Average Students. During this phase the accuracy improved and it was the best result among three. Further the research disclosed the boundary of the Good Students and Average Students as A, A-. The students who obtained A and above were identified as Good Students and the Students obtained A- and below in their final examination were recognized as Average Students. This was checked with all classification algorithms and verified the disclosed boundary of the label was true for all of the algorithms. The research had to do three experiments using a single algorithm to determine the boundary. The research employed 11 varieties of classification algorithms for the modeling. Altogether 33 experiments done to determine a single boundary. Similarly the boundary of the other labels such as “Very Good Students”, “Good Students”, “Average Students”, “Below Average” and “Very Low” were determined after the enormous number of iterative experiments. The Table 3.10 lodges boundary of each label of the students determined following by a variety of experiments.

Table 3-10 The boundary of each Student label

Label	Final Grade(Including Lower Boundary)
Very Good Student	A+, A
Good Student	A-
Average Student	B+
Below Average	B, B-
Very Low	C+, C, C-

Note: The research removed the students' grades D, I and F considered them as outliers.

Detecting the exact boundary of the labels is one of the ways of advancing the accuracy. The number of experiments using this technique helped to boost the accuracy of the model. The next technique is removing the outliers to enhance the accuracy.

Remove the outliers

The outliers were removed recursively at certain intervals. Though the research was able to collect the 343 student records at the data collection phase, later it diminished to 256 when the first model was constructed. The first modeling process removed the students who do not have their final grades and whose average ratings less than 0.5. But the research still hunts for accuracy as a result, it was decided to eliminate the outliers, if any, of such exist in the input data set. The outliers were removed by overlooking the distribution of the data set and the data points which deviate far from the mean distribution of the data set. The first phase of outlier detection and removal process removes the data which diverged from mean distribution of the data set.

Change the number of class variables

The research initiated the investigation of the student model with the number of class labels as two. Though it is obvious that the accuracy would drop down when the number of class labels increased, the research continued the experiment by advancing the number of class labels. The class labels were increased in order to observe the variation of the accuracy over the number of class variables. A model which is built with more class labels is more consistent than a model which predicts the data set with

two labels. The research commenced with the number of class labels as two and gradually to five by steps of one at a time. The research examines by switching the input data set splitting technology between randomly fragmented data set and manually segregated data set. The research also changes the classification algorithms, carefully choosing the label boundary, in addition to removing the outliers. Incrementing the number of class variables halted at five as the accuracy of the model drastically declined. It is useless to continue the experiment by incrementing the number of class variables as the prediction accuracy decreases. Irrespective of the algorithms the accuracy of the prediction model downgraded to less than 30% when the number of class label increased to five.

3.5.3 The resource recommendation model

The time for the students in a semester based study is restricted to 14 – 16 weeks. The students are expected to do the assignments, quizzes, forum discussion, wiki entries and other learning activities within this short period. Most of their time is spent to complete their assignment and other related activities in the learning management systems. They are preparing for their final examinations at the last minute. When they start to prepare for their exam, they begin to download all of the learning resources from the Moodle. And they do not have any help to choose which materials need to be studied first or which materials are more supportive for their studies. Their time is wasted when they are switching between the resources here and there to identify the most suitable materials in descending order for their examination preparations.

Gradual process of resource recommendation model

The resource recommendation model was formed from the ModResource table which contains 489 rows of resources and 38 attributes. Out of 38, 14 were directly fetched from Resources table. The rest were computed from several other tables and updated in the Mod Resource table. Unlike Student table, the ModResource does not have any null values in the computed columns. The resource recommendation model suggests the resources to the average students based on the access level of those learning materials

by the good students. Some of the materials are accessed mostly by the good students, whereas some are ignored by them. The research assumes that the materials left out by the good students are less usable for the students' studies. Hence the resource model depends on the results of the student model. The results exposed by the student classification model would be consumed by the resource recommendation model.

Split the data

Training and validating the model are not skippable steps in the data mining modeling process. The data to the model has to be chosen carefully and supplied. The research also took special care in selecting the raw data for the models. As it is done for the student model, the input data set for the resource model was also chosen randomly as well as manually. The resources exist in the Information Security of MBA 2012 batch and the Computer and Network Security of M.Sc. 2012 batch were grouped into test data and resources from the rest of the courses were formed as a training data set. The 489 resources were randomly divided into training and testing data in such a way 80% is allocated for the training and the remaining 20% is allocated for testing data. The training data set comprises 392 records of resources, whereas 97 pushed into test data set randomly. The SQL shown in Figure 3.44 fragments 80 % resource data set as training and remaining will be allocated to the testing data set.

```

IF OBJECT_ID('tempdb..#TmpResourceIds') IS NOT NULL
    DROP TABLE #TmpResourceIds
ELSE
BEGIN
    CREATE TABLE #TmpResourceIds(CourseId VARCHAR(50), Id VARCHAR(10))
END
INSERT INTO #TmpResourceIds(CourseId , Id)
SELECT TOP 80 percent [CourseId], [Id]
FROM [Moodle].[dbo].[ModResource] ORDER BY NEWID()
SELECT S.[Id] ,LEFT([FileSize], CHARINDEX('KB', [FileSize])-1) AS FileSize
FROM [Moodle].[dbo].[ModResource] S INNER JOIN #TmpResourceIds U ON
S.CourseId = U.CourseId AND S.Id = U.Id;

```

Figure 3-44 Prepare Training data for resource model

The SQL randomly divides the data set into 80 and 20 percentages and the 80% of the data set is selected as training data and the remaining data, in other words the data which were not selected as training data set were picked as testing data.

The idea of splitting the data set manually and randomly is to pay intensive care while choosing the best data set for the modeling input. The splitting mechanism which provides more accuracy would be selected as better splitting methodology.

Labeling the resources

Similar to the student data set which was labeled as good students and average student, the resource data set as well as labeled “highly accessed resources” and “less accessed resources” by the good students. The resources which are branded as highly accessed resources are the ones that need to be suggested to the average students. Hence the resources should be labeled correctly. The resources are classified as highly accessed and average access by the good students based on the percentage of access of a good student. But there is no clue what's the cutoff of the percentage of access to a resource by good students, which segregates the highly accessed resources and less accessed resources. Initially the resources that have percentage of access of a resource by good students more than 2.5 were considered as highly accessed resources and the rest were labeled as less access resources.

The accuracy was observed and the cutoff was advanced upwards with the increment of 0.25 in each step. The model was run and the accuracy was studied at 2.75 and 3.00. Though the accuracy too increased the number of highly accessed resources in the data set started to diminish. In other words the percentage of highly accessed resources within the data set started to decrease. This would bias the prediction model towards the less accessed resources. As a ratio of less accessed resources is getting increased the probability to predict the less accessed resources would be high. Based on that it was decided to tune the splitting point downwards and the boundary point gradually decremented by 0.25 from 2.5. The accuracy was noted when the splitting point moved

over 2.25 and 2.00. Based on the experiments the model outputs higher accuracy when the resources were split at 2.25 %.

In the next iteration the research attempted to divide the data set into three categories and labeled the resources accordingly. The splitting point of the resources when the number of class variable increments, determined carefully followed by several iterations of experiments as explained a bit in the previous paragraph as well as in the Student classification model.

Prepare the dataset

The data set was prepared followed by the splitting process. The training and testing data set for the resource recommendation model imported from CSV files. The data set contains 8 attributes and ResourceType was a polynomial type which was used to label the data set and the rest were regular data types such as integer and decimal.

Modeling process

The resource recommendation model was constructed using the prepared training and testing data set. The classification model was trained using the training data set supplied at the input port. It optimized using validation operator. The validation operator of the modeling process divides the training data into 10 sub folds as per the default configuration and treat one subset as the test data and the remaining as the training data and constructs the model and reads the accuracy in a single iteration. Later next data set was elected as the test data and it is keep on the loop till the number of the sub folds value configured at the validation operator. This is how the optimization is done by the validation operator of the RapidMiner to build up a consistent model. The resultant model of the validation operator later used to validate with the real test data using an Apply Model operator. The result of that operator later used by the Performance operator to figure out the accuracy, RMSE and other statistical parameters. The resource recommendation model which was fed by the manually split data set with the



two classification label which was sliced at 2.25 of percentage of access of good resource is shown in Figure 3.45.

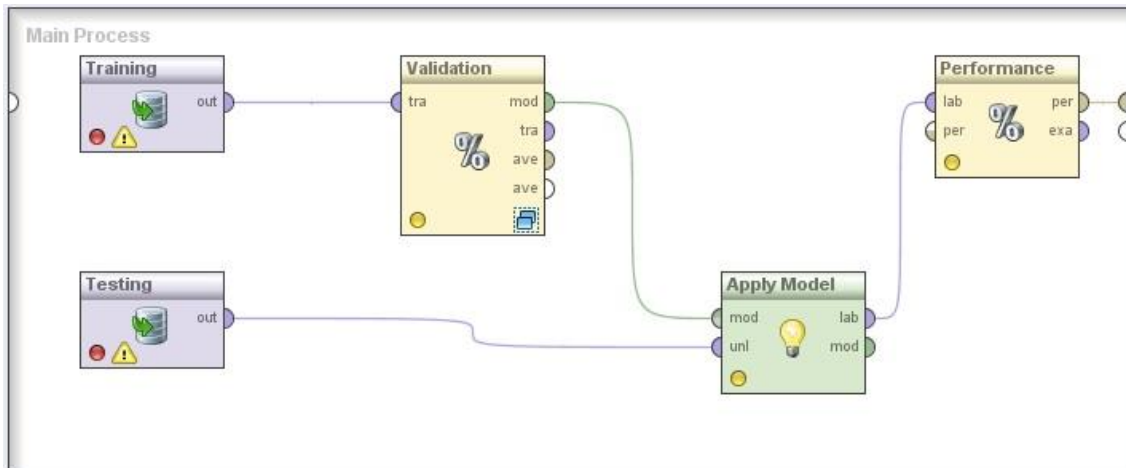


Figure 3-45 Resource classification process

Obtain the result

The accuracy obtained at the above process was 91.07%. Though the overall is accuracy 91.07%, the model predicts the good resources with 77.42% precision. Though it is not possible to compare the accuracy of student model with the resource model, the resource model leads in figure wise. The expected reason for the accuracy dominance is discussed in the conclusion.

Adjust and tailor make the result

Even though relatively the accuracy of the resource recommendation model discloses more accuracy the research did not halt the experimental process. It tried several options to enhance the accuracy, expecting at least a small boost in the accuracy. The classification algorithms swapped, the data set splitting mechanism altered, the number of classification labels increased and the configuration parameters adjusted to get better results. All the applied techniques are briefly discussed in the subsections below.

Determine the data fragmentation

The resource data set which was collected from the moodle.xml and the resources folder was fragmented into training and testing data for the modeling process. Two techniques were used to partition the data set, one was divided manually and the next was split randomly. The manually divided assigns all of the resources except for the courses Information Security of MBA 2012 and Computer and Network Security of M.Sc. 2012 to the training set. And the resources from the above said courses fill the test data set. They occupy 392 and 97 resources respectively in training and testing. The model yields more consistent and accurate results when the data set splitting mechanism was random. It turns out 95.88% of the accuracy when the splitting boundary was 2.25 and the number of class labels equal to two. The decision tree was used as the classification algorithm in the above experiment. 91.07% of the accuracy was observed when the data set changed to manually split the data while the others remain unchanged. The accuracy was observed in all of the manual and randomly split data set while changing one factor at a time and keeping the others intact. The Table 3-11 displays the boundaries of splitting point of the percentage of access of a resource by good students.



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Table 3-11 The percentage of access of a resource by good students

Label	% of access of resources by good students
Very High access	> 2.25
High access	>1.80
Average access	>1.40
Less access	<= 1.40

Change the algorithms

The research used 8 classification algorithms to test the accuracy over the resource recommendation model. All of the classification algorithms that were used in the project were implemented using underlying different structures. The Decision Tree, Neural Networks, Support Vector Machines (SVM), k-Nearest neighbor, Rule Induction, Naïve

basis, Perceptron and Linear Regression were used to testify the model. The accuracy was observed on each adjustment of the classification algorithm in the modeling process. In these circumstances only the classification algorithms adjusted while keeping the other factors untouched.

Change the label boundary

The accuracy of the model depends on the input data. The model is trained using training data which have the label that it is already known. The labeling of a data set has to be done correctly in order to get good results. But in this research the labels were not known in advance. The records have to be labeled based on some conditions. Identifying the exact condition was a challenge in this research and it has to do recursive tests to identify that condition. The label's boundaries were marked based on the conditions. The resource recommendation model was labeled based on the percentage of good students accessed this resource. Initially the research considered the resources which have percentage of access of a resource by good students more than 2.50 as heavily accessed resources and the rest as less accessed resources. The boundary shifted in both directions iteratively by 0.25 till 2.00 and 3.00 respectively. The other facts of the modeling process were untouched when the label boundary was changed.

Change the number of class variables

Even though it is obvious the accuracy would fall down when the number of the class label increased, the research intentionally did the experiment by amplifying the number of class variables to monitor the accuracy changes over the number of class labels and to get a consistent model. The accuracy fell down from 95.88 to 88.6 when the number of class labels enlarged from two to three and it further decreased to 70.1% when the number of class label was further extended to 4.

3.6 Evaluate and Choose Models

There are many models built, both for students and resource in this research but all of them cannot be used for the future predictions. Hence the best has to be chosen in order

to carry out future classifications. While choosing the model several factors have to be taken into account. Some models were best in accuracy under certain conditions only, where some would fail in some other conditions. As a result, it is highly expected that the model which is selected should be consistent over all of the conditions. The model should produce at least nearly equal accuracy for the new conditions. This section discusses in depth about the techniques used to choose the best model. The extra features possessed by the selected models, and the reasons for discarding others are discussed in detail. The performance factors used to evaluate the best model are discussed in the first section. The second subsection discussed about the student classification model and the next subsection analyzes and evaluated the best resource recommendation model.

3.6.1 The performance factors used to evaluate the models

Even though several performance factors packaged with the performance operator of the RapidMiner software the research selected only two out of them. The accuracy and the root mean squared error (RMSE) were the only two performance factors chosen for the evaluating the models. These two performance factors are normally chosen in any of the data mining practices. The accuracy gives the percentage of correctly predicted data set and for a good model, it is expected to have higher accuracy. And the RMSE is used to measure the gap between the expected value and the predicted value. A good model should result the RMSE small as possible.

The accuracy

The accuracy is the proportion of true results (true positives and true negatives) in the population. It can be calculated as shown in the Equation 3.14

$$\text{Accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{false positive} + \text{true negative} + \text{false negative}}$$

Equation 3-14 The accuracy of prediction

Once they the classification exercise has done it is necessary to determine the degree of accuracy in the final product [24].

The root mean squared error (RMSE)

The root mean square error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed values. The average of the error or deviation of prediction is not useful as the model result positive and negative. The best way to avoid this annoyance is to square this and calculate the RMSE [25]

3.6.2 Data quality issues

Similar to the product recommendations on e-commerce, the quality of learning recommendations of the models has an important role in a student's future learning behavior. Incorrect recommendations may result two types of errors, false negatives and false positives [2]. The false negative means the learning materials that are not recommended, though the students need to study them. The false positive stands for learning materials that are recommended, though the students do not need them or they are not suitable for the students. It is very important to avoid false positives in an e-learning environment, because these errors may lead the students to get disturbed and as a result they may not like to revisit the site. There is a risk at the adoption of e-learning environments as it has psychological cost as well. The poor recommendations may impact learning achievement of students [18].

3.6.3 Evaluate student classification model

The research constructed plenty of student classification models in order to determine the best student classification model which later will be used for the prediction of good student and the average student at the Department of Computer Science & Engineering, University of Moratuwa. Hence, it is required to build up the best model with more accuracy as the false result would de-motivate the students and the lecturers. False negative and the false positive should be reduced to the lowest possible level and at the same time true positive and true negative should be increased to the highest possible

level. In other words the model should predict the real good students as good students and label the real average student as average student as accurately as possible. At the same time it should not or predict less the good students as average students and average students as good students. Special attention and careful experiments done in order to produce the higher accurate and less error prone models as explained in the building models section.

Accuracy of the student classification model

The accuracy of the model was calculated based on the percentage of correctly predicted labels. The accuracy of the models were obtained by slightly adjusting the classification algorithms, input data splitting techniques, detection and removal of outliers and change the number of class labels. The accuracy observed in each scenario is discussed in depth in the sub sections to follow. All of the accuracies observed at manual data fragmentation with two class labels for all of the chosen classification algorithms are tabularized below using Table 3.12.

Table 3-12. The accuracies of student models with manual data fragmentation for 2 labels

Used data set Algorithms	Average ratio greater than 0.5	Outliers removed first time	The label boundary changed down	The label boundary changed up	Outliers removed second time
Decision Tree	50.98	53.33	60.00	82.22	75.00
Neural Networks	66.67	60.00	62.22	71.11	78.57
SVM	64.71	64.44	68.89	75.56	78.57
k-NN	54.90	62.22	62.22	62.22	67.86
Naive Basis	70.59	64.44	66.67	80.00	78.57
Rule Induction	68.63	77.78	71.11	77.78	82.14
Perceptorn	50.98	53.33	40.00	75.56	75.00
Linear Regression	70.59	73.33	73.33	75.56	78.57
Polynomial Regression	41.18	53.33	44.44	62.22	64.29
Vector Linear Regression	50.98	46.67	60.00	24.44	21.43
Gaussian Process	56.86	64.44	64.44	68.89	71.43

The used data set shown in each column, applied to different types of data alteration. Initially the data set chose the students who obtained the average ratio of more than 0.5. That data set was initially used and the accuracies read. Later outliers removed from the same set which is previously used as in the 1st column. Some of the outliers removed for both training and testing data using the same SQL. The students who had more A- or more than that in their final exam were labeled as good students in the first and second data set. But in the third data set the boundary of the labeling was shifted upwards and the data set at the third column, labeled the good students as whoever had more than A and A+. The rest were categorized into average students. The classification label boundary brought down in the fourth data set to exactly determine the correct labeling boundary. The students whoever had B+ and more were labeled as good students and the rest as average students. Finally the data set was shrunk further followed by the 2nd outlier detection and removal operation applied. All over the changes of the data set the same data set which was used at the 1st step used with the alterations and the modifications to the data sets were saved as different input data repositories. The 1st outliers removed from the initial data set. The resulted data set from this second step was used to shift the label boundary upwards and downwards as in the 3rd column and 4th column data set. Further outliers removed from the 3rd column data set as that shows higher accuracy than the 4th column data set.

A nearly equal experiment was applied to the data set which was segregated randomly. The accuracy studied while random fragmentation is specified in the Table 3.13.

Table 3-13 The accuracies of student models with random data fragmentation for two class labels

Used data set Algorithms	Average ratio greater than 0.5	Outliers removed first time	The label boundary changed up	The label boundary changed down	Outliers removed second time
Decision Tree	61.90	67.39	56.52	84.38	65.62
Neural Networks	65.08	65.22	63.04	65.22	62.5
SVM	66.67	65.22	65.22	82.61	75.00
k-NN	50.79	54.35	50.00	78.26	65.62
Naive Basis	65.08	67.39	60.87	76.09	75.00
Rule Induction	49.21	71.74	58.07	82.61	78.12
Perceptron	60.32	60.87	56.52	45.65	25.00
Linear Regression	68.25	69.57	60.87	78.26	78.12
Polynomial Regression	49.21	52.17	41.30	43.48	40.62
Vector Linear Regression	61.90	32.61	56.52	82.61	75.00
Gaussian Process	49.21	54.35	50.00	80.43	65.62


 University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The root mean squared error (RMSE) of the student classification model also observed during the model construction phase, in order to determine the best model. The RMSE figures obtained is analyzed with the classification algorithms and the data splitting techniques in following subsections. The table 3.14 exhibits the RMSEs obtained for manually fragmented data with two class labels.

Table 3-14 The RMSEs of student models with manual data fragmentation for 2 class labels

Used data set Algorithms	Average ratio greater than 0.5	Outliers removed first time	The label boundary changed up	The label boundary changed down	Outliers removed second time
Decision Tree	0.51	0.51	0.49	0.38	0.45
Neural Networks	0.51	0.60	0.56	0.5	0.41
SVM	0.46	0.47	0.44	0.43	0.39
k-NN	0.67	0.61	0.61	0.61	0.57
Naive Basis	0.46	0.49	0.47	0.39	0.44
Rule Induction	0.49	0.43	0.49	0.45	0.41
Perceptron	0.70	0.68	0.77	0.49	0.50
Linear Regression	0.47	0.47	0.47	0.46	0.45
Polynomial Regression	23.7	205.60	181.97	135.76	22.82
Vector Linear Regression	0.70	0.73	0.63	0.87	0.87
Gaussian Process	0.66	0.59	0.59	0.55	0.53

3.6.4 Evaluate resource recommendation model

Similar to the student modeling process, several resource recommendation models constructed in order to find out the best resource recommendation model. Those models need to be closely analyzed in order to identify the optimized model that would do the better recommendation


Accuracy of resource recommendation model

The accuracy was observed for the resource recommendation models from the performance operator of RapidMiner. Each time the accuracy was obtained by changing the splitting techniques, switch the classification algorithms, remove outliers and change the class labels. The tables 3.15 and 3.16 presents the accuracies obtained for manual and random split of resource data with two class labels.

Table 3-15 The accuracies of resource models with manual split for two class labels

Used data set	% of access by good students split at 2.00	% of access by good students split at 2.25	% of access by good students split at 2.50	% of access by good students split at 2.75	% of access by good students split at 3.00
Decision Tree	81.19	91.07	93.69	95.54	100.00
Neural Networks	88.29	87.50	91.89	95.54	96.43
SVM	90.09	85.71	92.79	94.64	100.00
k-NN	65.77	67.86	76.58	88.39	90.18
Rule Induction	90.99	85.71	94.50	90.18	100.00
Naive Basis	88.29	88.62	90.09	95.89	90.18
Perceptron	40.54	72.32	78.38	85.71	100.00
Linear Regression	94.59	93.96	96.40	87.50	100.00

Table 3-16 The accuracies of resource models with random split for two class labels



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
 www.lbr.mrt.ac.lk

Used data set	% of access by good students split at 2.00	% of access by good students split at 2.25	% of access by good students split at 2.50	% of access by good students split at 2.75	% of access by good students split at 3.00
Decision Tree	92.78	95.88	92.78	94.85	98.97
Neural Networks	90.72	93.81	91.75	90.72	98.97
SVM	87.63	91.75	92.78	92.78	97.94
k-NN	65.98	71.13	73.12	80.41	89.69
Rule Induction	85.57	91.75	92.78	95.80	94.85
Naive Basis	82.47	85.57	81.51	88.60	88.60
Perceptron	46.39	50.52	69.07	83.51	94.85
Linear Regression	88.66	93.81	95.88	87.63	97.94

RMSEs of resource recommendation models

The RMSE values of each resource recommendation model were observed for the resource recommendation models from the performance operator of RapidMiner. Each time the RMSE was obtained by changing the splitting techniques, switch the classification algorithms, remove outliers and change the class labels. The table 3.17 demonstrated the RMSEs resource recommendation models with random data fragmentation for two lasses obtained for each algorithm.

Table 3-17 The RMSEs of resource models with random data fragmentation for two class labels

Used data set	% of access by good students when split value at 2.00	% of access by good students when split value at 2.25	% of access by good students when split value at 2.50	% of access by good students when split value at 2.75	% of access by good students when split value at 3.00
Algorithms					
Decision Tree	0.24	0.20	0.25	0.23	0.08
Neural Networks	0.26	0.22	0.21	0.28	0.11
SVM	0.30	0.26	0.27	0.25	0.16
k-NN	0.58	0.54	0.54	0.44	0.32
Rule Induction	0.36	0.28	0.25	0.20	0.22
Naive Basis	0.37	0.34	0.34	0.32	0.29
Perceptron	0.73	0.70	0.56	0.41	0.23
Linear Regression	0.44	0.43	0.43	0.47	0.40

4 ANALYSIS OF THE RESULTS

This chapter analyses the results obtained in the section 3.6 for both student classification and resource recommendation models. The results are analyzed based on the performance factors such as accuracy and the RMSE.

4.1 The Analysis of Student classification model

The student classification model was analyzed with the factors, accuracy and RMSE and based on that the best student classification model was deduced.

4.1.1 The accuracy analysis of student classification model

The accuracy of the student classification model was observed by changing the classification algorithms, switching the data set fragmentation technology, removing the outliers and adjusting the class label boundary of the final grade.

The accuracy variation over the classification algorithms

The research employed 11 classification algorithms in order to determine the best classification algorithms to classify the students. Though the difference is small, the overall accuracy of all of the classification algorithms shows more accuracy in manual data splitting than the randomly split data except when the label changed up as in the 3rd column of Table 3.12. Hence the research does more test on the manually fragmented data set. The Figure 4.1 plots the variation of the accuracy of the classification algorithms along the data set changes.

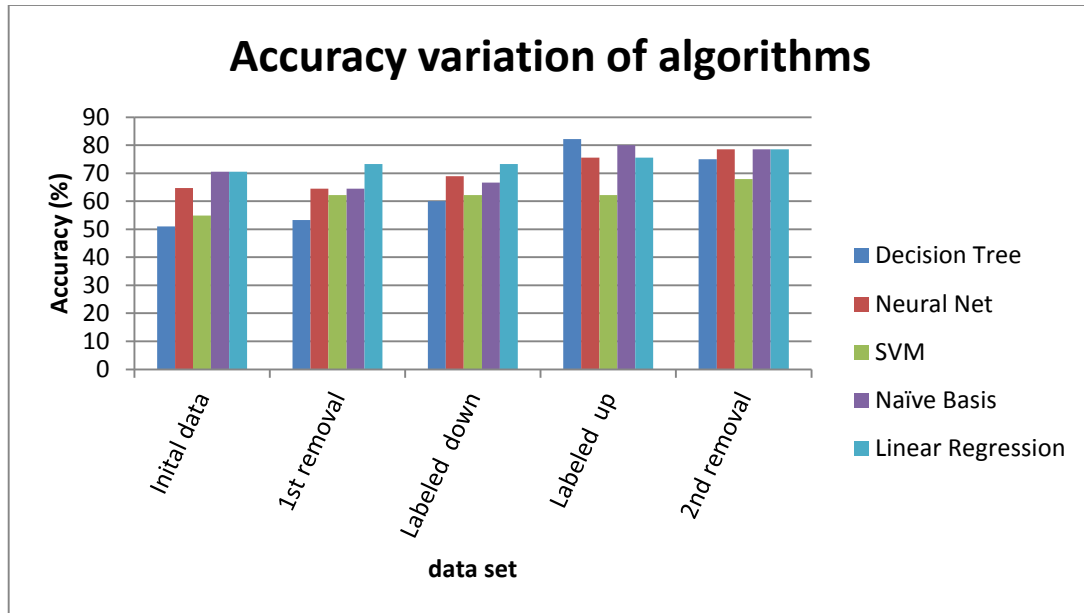


Figure 4-1 Accuracy variation of selected algorithms

Some of the algorithm's accuracies were not plotted here in order to maintain the clarity. The accuracy variation of some of the important classification algorithms exposed in the Figure 4.1. The accuracy is high when the used data set was labeled upwards. In other words, the accuracy would be high if the students who obtained A , A+ categorized as good students. Initially the accuracy starts with a figure and when the 1st outliers were removed the accuracy showed an increase. The accuracy slightly dropped when the labeling boundary shifted downwards and it accelerates when its changed upwards. It is notable that the accuracy drops down in some of the algorithms the outliers removed second time. The reason may be though outliers removed, the data set may not contain enough number of data for the modeling. Removing outliers always will not improve the accuracy.

The accuracy variation over the data set splitting mechanism

The results observed in the Table 3.16 and Table 3.17 shows a slight high accuracy when the data set was fragmented manually. The Figure 4.2 which plots the accuracy of neural network algorithms over a different set of data sets show slightly more accuracy

in manual data than random. But the Figure 4.3 demonstrates a small higher accuracy in random over manual that was done for decision tree classification algorithm. But a tiny excess in randomly split over manual split for the decision tree algorithm. Though some of the algorithms show a bit elevation in the randomly split data, the overall accuracy is slightly higher in the manually split data. Hence the research chose the manually fragmented data for the future experiments. The Figure 4.2 and 4.3 compare the accuracy of two different classification algorithms along with the data split.

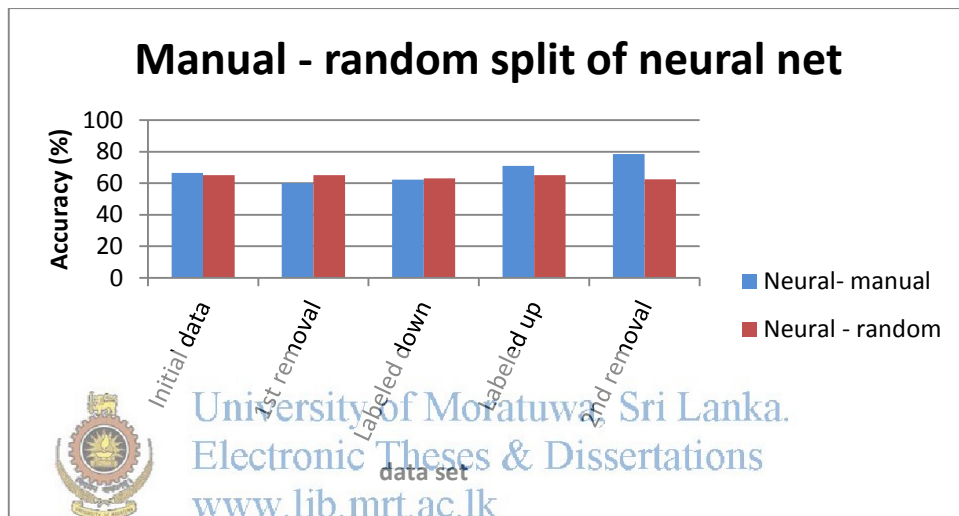


Figure 4-2 Accuracy comparison of data splitting - neural networks

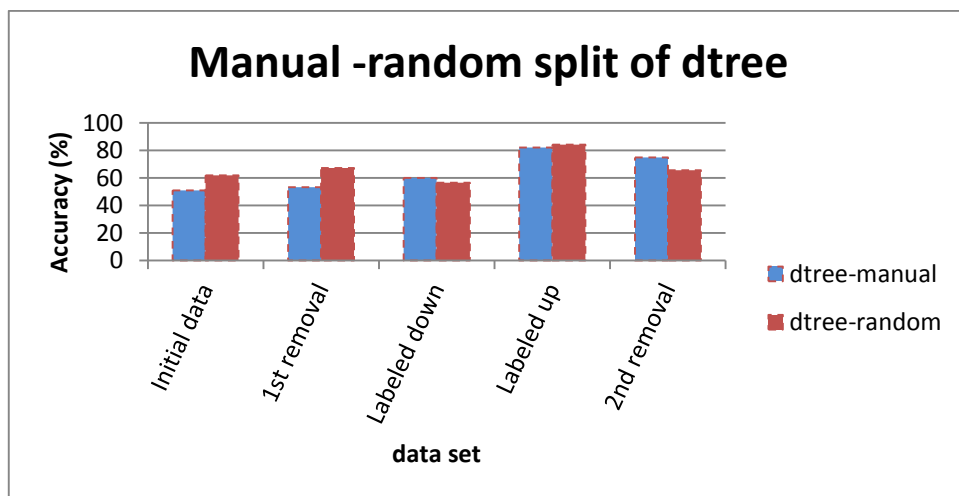


Figure 4-3 Accuracy comparison of data splitting - decision tree

The accuracy variation over the outlier removal

The initial data which was freed from non-null grade was applied with the outlier detection and removal process in two stages. At the first stage it posses altogether 256 student records and later it was reduced to 186 followed by the first outlier removal process. During the second stage of outlier removal process it further got shrunk and the data set ended up with 131 records. The accuracy changes over the outlier removal of five algorithms are given in the Figure 4.4.

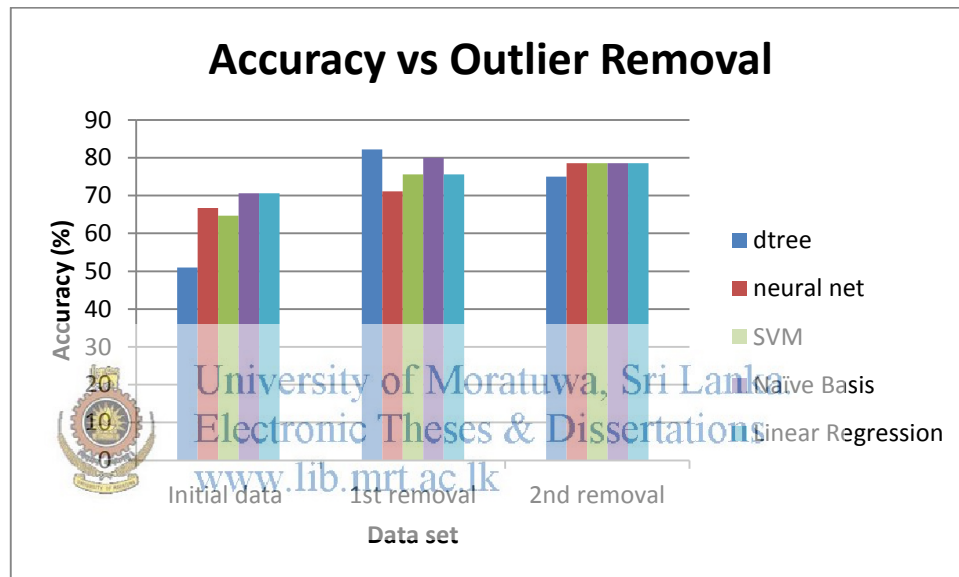


Figure 4-4 The accuracy variation over the outlier removal

The accuracy variation over number of class labels changes

The accuracy of the models declined rapidly when the number of class labels were gradually increased from two to five. It is an obvious and expected behavior as the number of class labels grow, the probability to predict the correct labels would drop exponentially. The accuracy of the prediction is calculated based on the right prediction along the diagonal of the confusion matrix. If the number of class label is two, then the confusion matrix would be 2 x 2 and two cells along the diagonal would result the correct predictions and the rest of the two would reveal the incorrect predictions. In other words, the diagonal cells of the confusion matrix would provide the true positive

and true negative where as the rest two cells of the matrix result the false positive and false negative. But if the number of class labels increased the confusion matrix would be 3 x 3 and out of those cells, 3 along the diagonal were used to calculate the accuracy and the rest 6 would contribute to increase the error. It is obvious that the number of cells that contribute to the error in the confusion matrix increased exponentially from 2 to 6 when the number of labels increased from 2 to 3. It further increases to 12 and 20 when the number of class labels further advances to 4 and 5 respectively. As a result, it is transparent that the accuracy would rapidly drop along the number of class labels. The Figure 4.5 graphically exhibits it clearly.

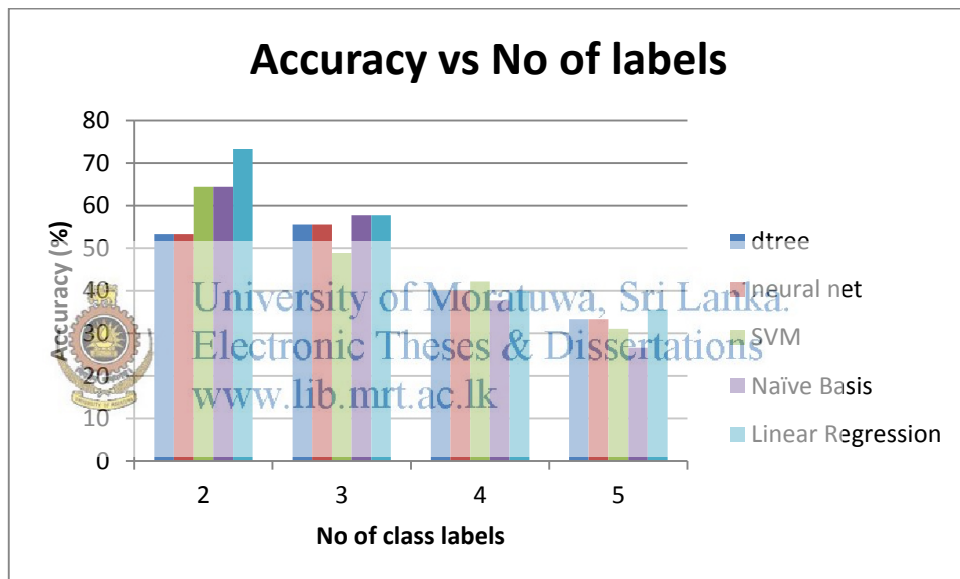


Figure 4-5 The accuracy varies with the number of class labels

The graph plotted with the initial data that was applied different number of class variables for selected algorithms. In order to maintain clarity of the graphs, only the graphs of 5 algorithms plotted. The accuracy of all of the algorithms is given in Appendix B.

4.1.2 RMSE analysis of student classification model

By changing the classification algorithms, switching data set fragmentation technology, removing outliers and adjusting the class label boundary of the final grade the RMSEs of the student classification model was studied and analyzed.

The RMSE variation over the classification algorithms

The RMSE variation with the different set of data set used in the research is charted in Figure 4.6. The RMSE decreases towards the right side except for the 2nd outlier removal. As already discussed, the accuracy increases, in essence the RMSE declines for most of the algorithms with the different data sets.

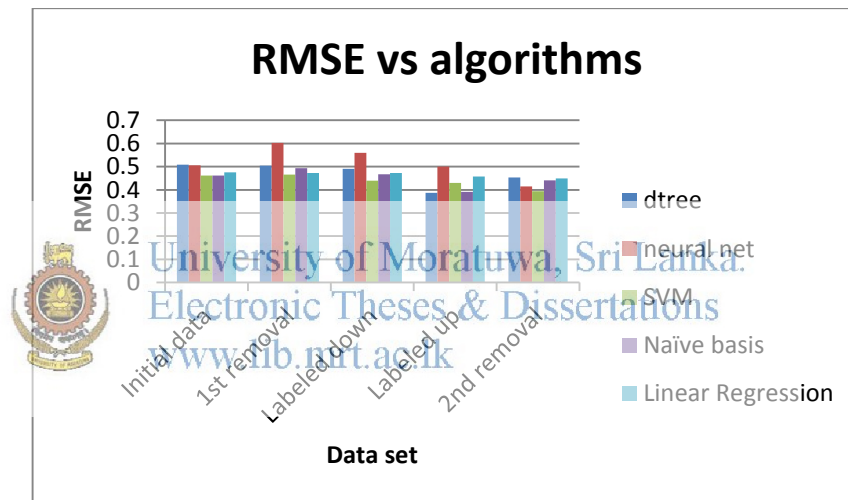


Figure 4-6 The variation of RMSE with algorithms

The RMSE variation over the data set splitting mechanism

The Figure 4.7 graphs the variation of the RMSE of decision tree of manual and random data splitting techniques. The research chose only the decision tree algorithms as that gave more accurate and less error algorithm for the student data set.

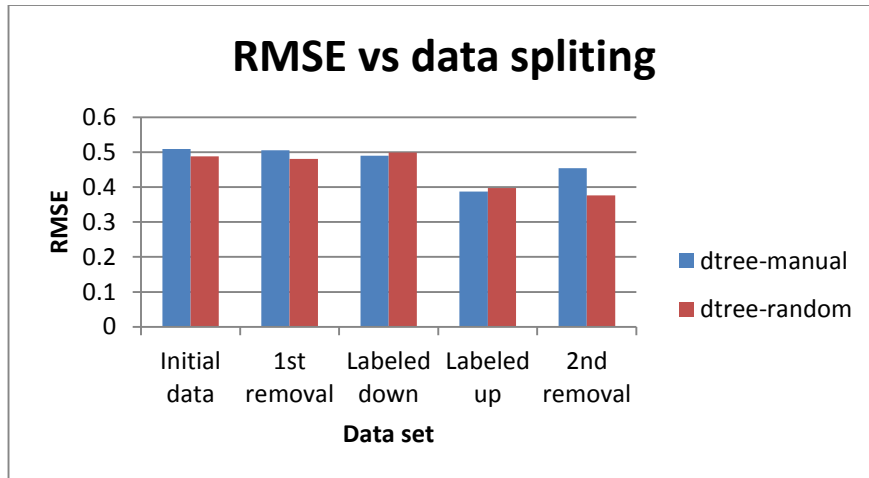


Figure 4-7 The RMSE variation across different data set

4.1.3 The best student classification model

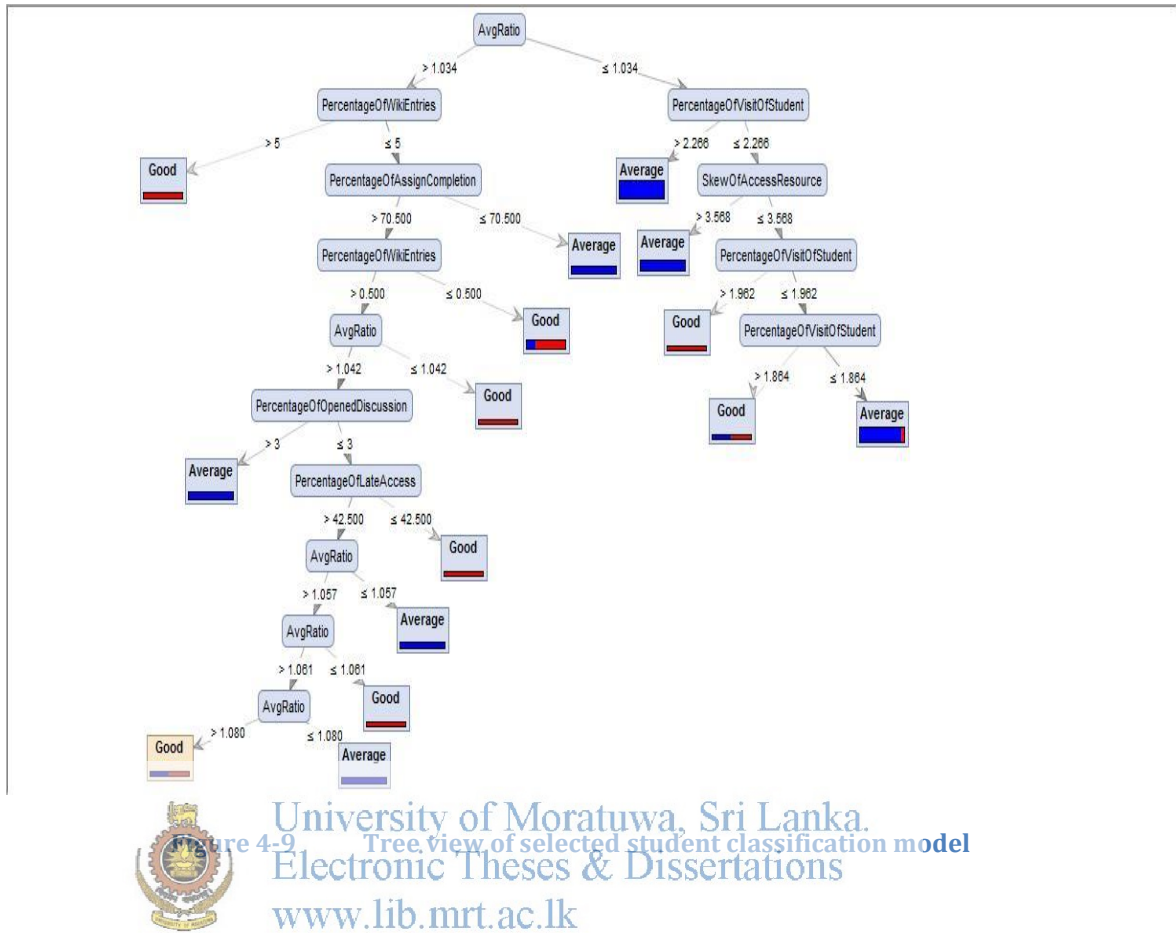
As already explained the research formulated several student classification models to evaluate and choose the best model. Among all of them the ideal model should be higher in prediction accuracy and less in error rate. This section would do the evaluation and choose the best student classification model. Out of all of the algorithms decision tree produces good results. And the more accurate model resulted when the data set that was prepared with labeled up. The decision tree with the two class labels outputs 82.22 and 84.38 percentages of accuracy in manual and random data split respectively. They produce the models with 0.39 and 0.40 errors respectively. Though other algorithms generate slightly better accurate models using manual data split, the decision tree supplies the most accurate result. But when the accuracy is compared the manually split model is the least error model. It has 0.01 errors less than the randomly split model. As it is a tiny error difference the research choose the model which was generated from randomly split data set with two class labels that result 84.38% accurate results as the best student classification model.

The selected model's rule

The Figure 4.8 illustrates the text view of the model with the conditions and within the curly brackets the number of training data sets that satisfies the conditions. A part of the decision tree model is shown in the Figure 4.9.

```
AvgRatio > 1.034
| PercentageOfWikiEntries > 5: Good {Average=0, Good=3}
| PercentageOfWikiEntries ≤ 5
| | PercentageOfAssignCompletion > 70.500
| | | PercentageOfWikiEntries > 0.500
| | | | AvgRatio > 1.042
| | | | | PercentageOfOpenedDiscussion > 3: Average
{Average=7, Good=0}
| | | | | PercentageOfOpenedDiscussion ≤ 3
| | | | | | PercentageOfLateAccess > 42.500
| | | | | | | AvgRatio > 1.057
| | | | | | | | AvgRatio > 1.061
| | | | | | | | | AvgRatio > 1.080: Good {Average=1,
Good=1}
| | | | | | | | | AvgRatio ≤ 1.080: Average
{Average=3, Good=0}
| | | | | | | | | | AvgRatio ≤ 1.061: Good {Average=0,
Good=2}
| | | | | | | | | | AvgRatio ≤ 1.057: Average {Average=5,
Good=0}
| | | | | | | | | | PercentageOfLateAccess ≤ 42.500: Good
{Average=0, Good=2}
| | | | | | | | | | AvgRatio ≤ 1.042: Good {Average=0, Good=2}
| | | | | | | | | | PercentageOfWikiEntries ≤ 0.500: Good {Average=2, Good=6}
| | | | | | | | | | PercentageOfAssignCompletion ≤ 70.500: Average {Average=6,
Good=0}
AvgRatio < 1.034
```

Figure 4-8 The text view of the selected student classification model



4.2 The analysis of resource recommendation model

The resource recommendation model was analyzed with the factors, accuracy and RMSE and based on that the best resource recommendation model was deduced.

4.2.1 The accuracy of resource recommendation model

The accuracy of the resource recommendation model was observed by changing the classification algorithms, altering the data set fragmentation technology, removing the outliers and adjusting the class label boundary of the final grade.

Accuracy variation over classification algorithms

The research used 8 classification algorithms to evaluate the resource recommendation model. The accuracies obtained for each algorithm by switching the algorithms while keeping the environment remain unchanged. The Figure 4.10 demonstrates the accuracy

variation of the resource recommendation models over the classification algorithms with the two class labels and randomly split data.

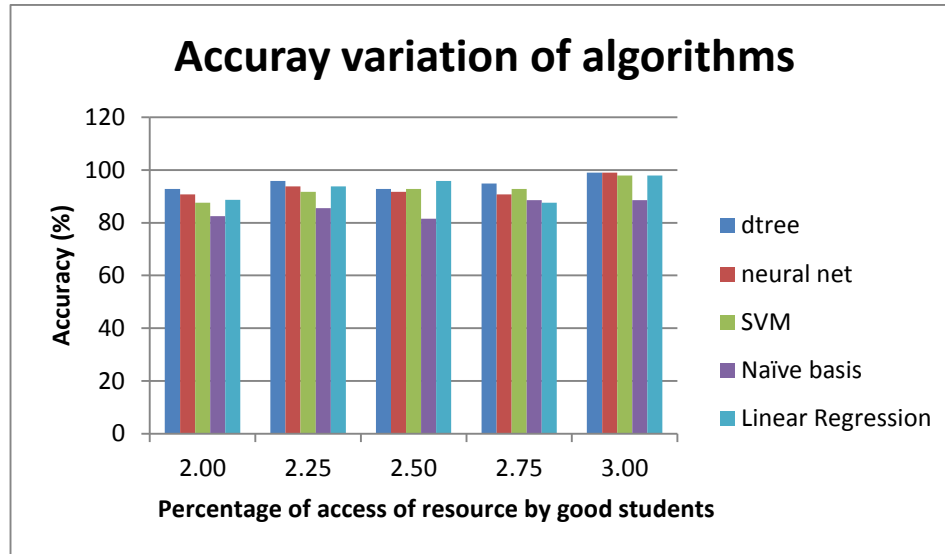


Figure 4-10 The accuracy of the resource models for different algorithms

Most of the algorithms provide more accurate results when the data split at 3.00%. But when the data set is divided as highly accessed resource and less accessed resource at 3.00 the percentage of resources whose access is beyond 3% is less. Only 5 resources are most accessed from the total of 111 resources. Hence the model would result biased result. The Table 4.1 exhibits the number of highly accessed resources at each splitting point.

Table 4-1 The number of highly accessed resources at each different splitting point

Splitting point	Most accessed resources
2.00	46(111)
2.25	31(111)
2.50	24(111)
2.75	15(111)
3.00	5(111)

Based on the accuracy and the number of most accessed resources in the test data it was concluded the 2.25 is best point to split the most accessed resources and the less accessed resources.

Accuracy variation over splitting mechanism

The testing and training data split from the total 489 resources manually as well as randomly. But unlike in the student model, the models which, fed by the randomly split data shows more accurate results. Though overall accuracy is a bit higher on the randomly fragmented data, the research charts the accuracy variation of the decision tree algorithm in the Figure 4.11 for clearer interpretation.

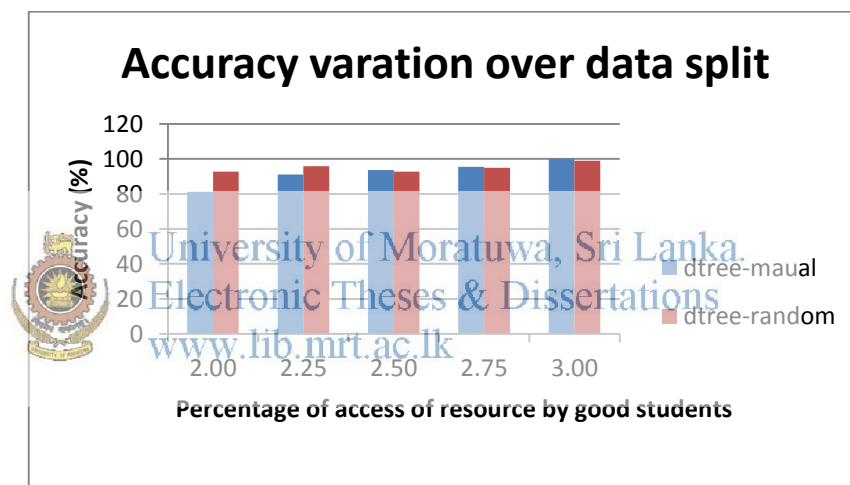


Figure 4-11 The accuracy varies with different data splitting point

A small spike can be observed in the randomly split data in the Figure 4.11.

Accuracy variation over class label changes

As the overall accuracy of randomly split data was slightly higher than the manually split data, the Figure 4.12 analyzes the accuracy variation over the number of class labels at the splitting point 2.25 for selected algorithms.

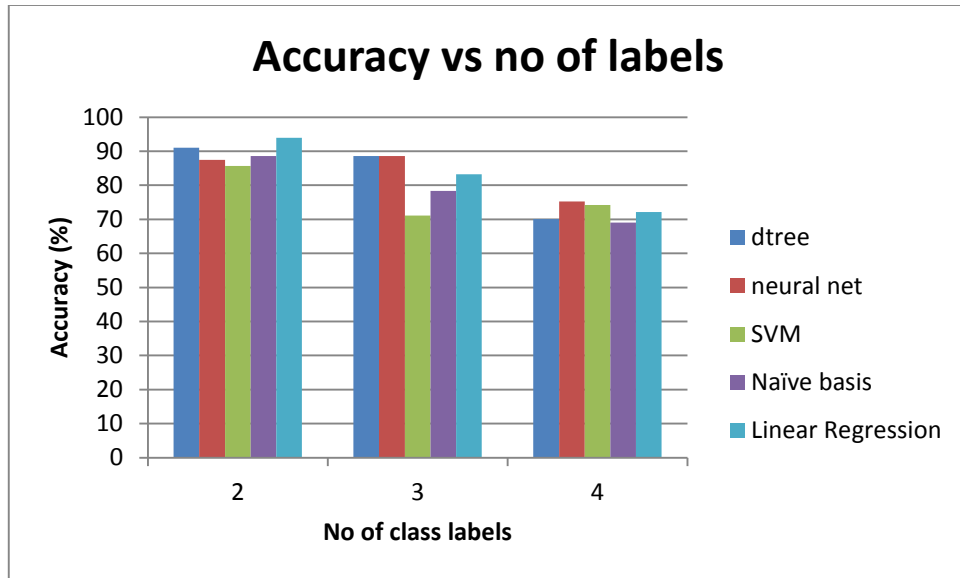


Figure 4-12 The accuracy declines with the change of number of class labels

It is obvious the accuracy would decline when the number of class labels grows, the graphs too behave so.

4.2.2 The RMSE analysis of resource recommendation model

The Figure 4-13 shows the RMSE variation of selected resource recommendation models that are based on decision tree algorithm with different splitting points and having two labels of classes. The models that are split using random split technique shows less error.

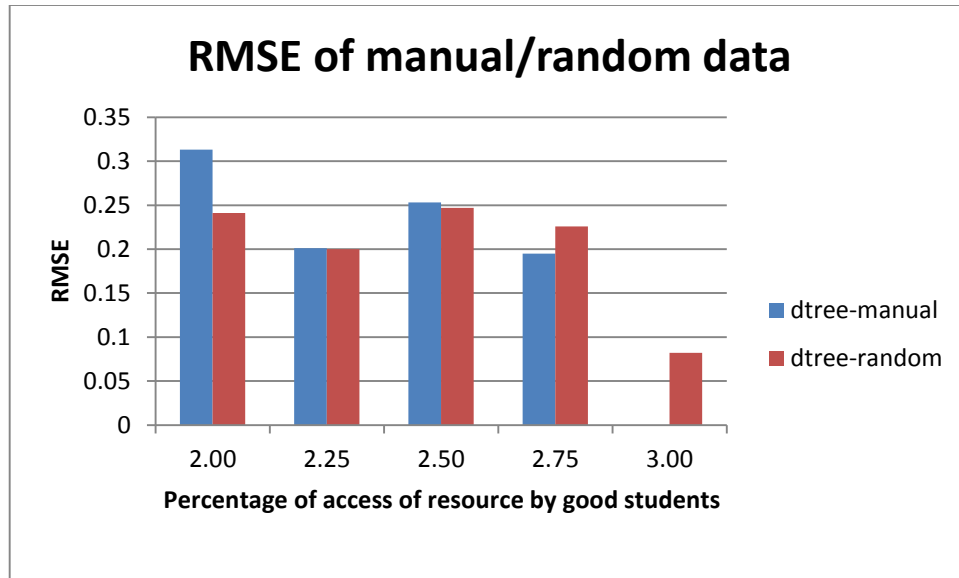


Figure 4-13 The RMSE of student model varies with manual and random data

The Figure 4.13 clearly shows the error is less at the split point 2.25 irrespective of 3.00 which was ignored as it contained biased data set. Also the RMSE is 0 for manual split data as a result, it has not appeared on the graph.

4.2.3 The best resource recommendation model

Based on the results obtained, the next task is to determine the best resource recommendation model. The decision tree at the splitting point 2.25 provides 95.88 and 91.07 percentage accuracy in random and manual data fragmentation. Also the error is tiny less in the randomly split data which has 0.200 and the manual data set has 0.201 errors. Based on the two factors the best resource recommendation model is formed by decision tree and fragmented by randomly with two class label is chosen as the best resource recommendation model.

The best resource recommendation model's rule

The text view of the resource recommendation model obtained by RapidMiner for decision tree with two class labels at the splitting point of 2.25 is shown in Figure 4.14 and the portion of the tree view is shown in Figure 4.15.

```
percentageOfAccessOfSingleResource > 2.260
|  percentageOfAccessOfSingleResource > 2.485: Good Resources {Good
Resources=115, Normal Resources=0}
|  percentageOfAccessOfSingleResource ≤ 2.485
|  |  percentageOfAccessOfSingleResourceByAvgStudents > 2.665: Normal
Resources {Good Resources=0, Normal Resources=4}
|  |  percentageOfAccessOfSingleResourceByAvgStudents ≤ 2.665
|  |  |  percentageOfAccessOfSingleResourceByAvgStudents > 2.275
|  |  |  |  FileSize > 139
|  |  |  |  |  FileSize > 701
|  |  |  |  |  |  FileSize > 1465.500: Good Resources {Good
Resources=2, Normal Resources=0}
|  |  |  |  |  |  FileSize ≤ 1465.500: Normal Resources {Good
Resources=0, Normal Resources=2}
|  |  |  |  |  |  |  FileSize ≤ 701
|  |  |  |  |  |  |  |  FileSize > 196.500: Good Resources {Good
Resources=11, Normal Resources=0}
|  |  |  |  |  |  |  |  FileSize ≤ 196.500
|  |  |  |  |  |  |  |  |  PercentageOfGoodStudents > 0.205: Normal
Resources {Good Resources=0, Normal Resources=2}
|  |  |  |  |  |  |  |  |  PercentageOfGoodStudents ≤ 0.205: Good
Resources {Good Resources=3, Normal Resources=0}
|  |  |  |  |  |  |  |  |  |  FileSize ≤ 139
|  |  |  |  |  |  |  |  |  |  |  PercentageOfAvgStudentAccesedThisResource > 78.500:
```

Figure 4-14 The text view of selected resource recommendation model

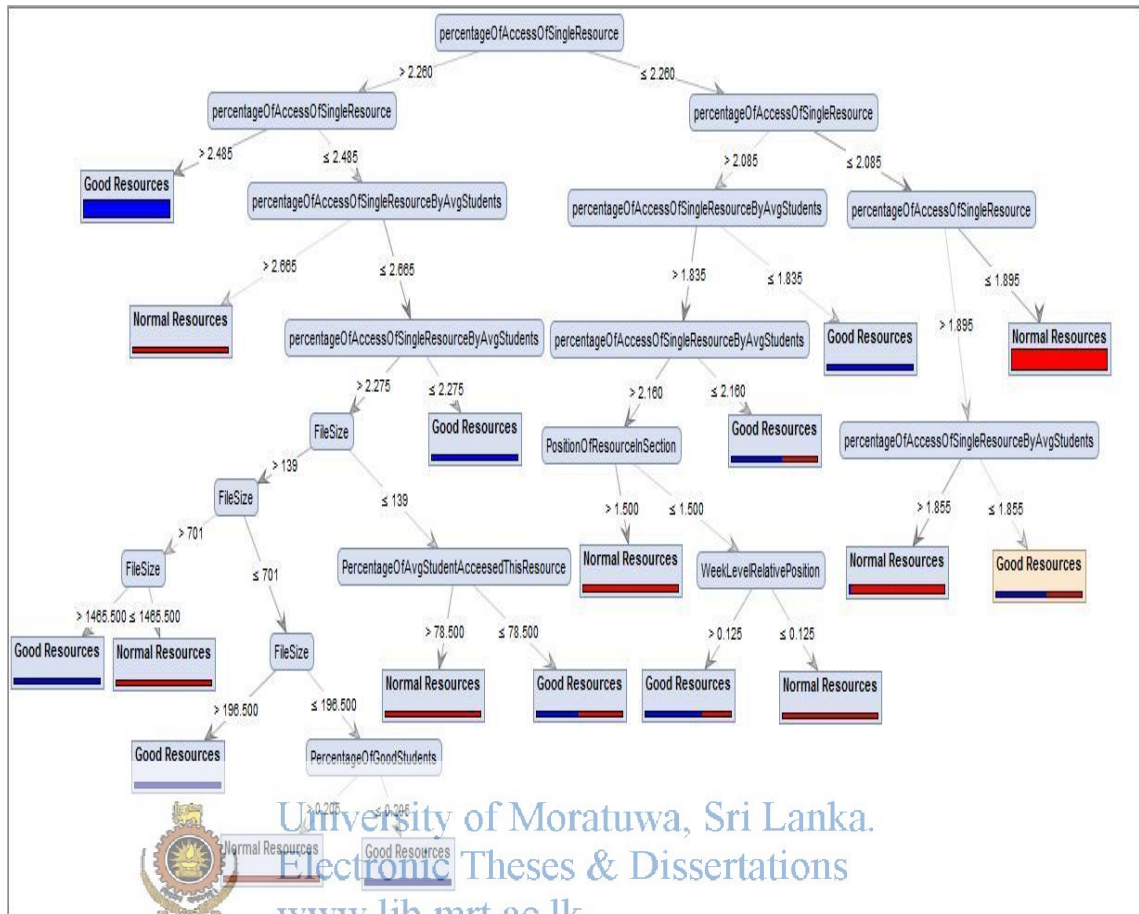


Figure 4-15 A tree view of portion of resource recommendation model

5 CONCLUSION AND RECOMMENDATION

This chapter summarizes the conclusions and the recommendations inferred from the findings of the research and suggestions for future research.

5.1 Discussion

The rapid growth of e-learning has made the learning management systems (LMS) to accommodate large amounts of e-learning material. The students spend a vast time on the assignments and other LMS related activities within their short semester. It is very difficult for students to find time to study all the e-learning materials available in the LMSs. The excess of materials in the LMSs push students to spend a significant time on browsing and filtering learning resources to identify the most suitable materials for their studies. Hence, if the students are guided using a resource recommendation framework that suggests most relevant material, it would be supportive for them. They can spend more time on studying materials if the recommendation framework takes care of selecting the most appropriate materials for their studies.

The good students are smart enough to choose best materials for their studies within a short period, but the average students are not good at this. The social learning theory states that people can learn by observing the behavior of others. The “good students” indirectly act as role models to their fellow friends, where the average learners can follow the methods adapted by the good students for learning or accessing relevant learning material, by referring to the “mostly accessed learning materials” by the “good students”[1].

5.1.1 The analytical models

The research was initiated with the problem of classifying the mostly accessed and least accessed learning materials by good students as stated in section 1.1. In order to solve this problem, the research constructed “student classification” and “resource recommendation” models as in sections 3.5.2 and 3.5.3. The former attempts to classify the students as “good” and “average” students and the latter model suggests the “mostly accessed” materials by the good students.

The research collected Moodle access logs of the courses with a rich and sufficient set of e-learning materials conducted by the Department of Computer Science and Engineering, University of Moratuwa. The data set was preprocessed by the chaining process of data collection, transformation, reduction and partition [3]. The student and the resource models were formulated from the prepared data set. Several techniques were used to determine the best models as in sub sections of 3.5.2 and 3.5.3.

The student classification model was experimented with 11 classification algorithms where the resource recommendation model was tested with 8 algorithms to determine the best models. The decision tree algorithm provides more accurate results compared to the other algorithms in both models as elaborated in section 3.6.2. The input data were split manually and randomly as training and testing data sets, to ensure that there is no discrimination in the input data. However both did not exhibit considerable discrimination in the accuracy as described in a sub section of 3.6.2. The results are nearly equal in both manual and randomly fragmented data for both student and resource models. But the results of the student classification models were slightly higher in manually split data and the resource recommendation model secures a bit high in accuracy for the randomly fragmented data. The resource recommendation model turns out with 95.88 % accuracy when the splitting point of percentage of access by good students was 2.25% using a decision tree algorithm for the randomly divided data. It was observed 91.07 % accuracy when the data set was switched to the manually fragmented data as reported in section 4.2.3. The student model obtains 82.22 % accuracy using manually divided data with two class labels for decision tree algorithm where it produces 84.38 % for randomly fragmented data as per section 4.1.3.

There was no good prior heuristic to define the slicing point of the class variables. But based on the analysis, the students who obtained “A” and “A+” were categorized as “Good Students” and the rest were labeled as “Average Students” as described in one of the sub sections of 3.5.2. Similarly the resources that had more than 2.25% percentage of access by good students were considered as “mostly accessed resources” and the rest

as “less accessed”. These conclusions were not derived by a single iteration. Rather than that they were derived based on a continuous process with necessary adjustments to the experiment process and the parameters. The final models were derived after applying several adjustment techniques such as identifying the correct class label boundaries, choosing correct data splitting mechanisms, removing the outliers and switching the classification algorithms and fine tuning their parameters. The best student classification model delivers 84.38 % accuracy and the best resource recommendation model secures 95.88 % accuracy. Implementing an intelligent feedback system based on the mostly accessed resources by the good students was the aim of the project. The feedback system would be a combination of student classification and resource recommendation models. The statistics figures discussed in this section would help the readers to realize that the aim of this research project has been successfully achieved.

5.1.2 The student model

The decision tree which is shown in the Figure 4.9 reveals that the average ratio (the ratio between the average marks obtained by a student for all assignments within a given course and the average marks obtained by all of the students for all assignments in that course), is the most influencing factor while constructing a model as it was at the top of the decision tree. If the student got more than 1.034 as average ratio, it is more probable to be a good student irrespective of other attributes. As the right hand side branch of the decision tree unveils the higher probability of classifying a student as an average student unless he has the percentage of visit above 1.982 and the skew of the access resource is less than 3.568. The notable rule in the model is the model predicts the students as good students who have the late access less than 42.5% and some other constraints. The research assumes that a good student may not access the resources at the last minute instead they access all of the resources from the beginning and uses consistently throughout the semester.

5.1.3 The resource recommendation model

The percentage of access of a single resource has a significance while discriminating the mostly accessed resources and less accessed resources by the good students. Its more

probable for a resource to be a good resource when the percentage of access of a single resource is more than 2.26. The file size also plays a noteworthy impact on the resource recommendation model. The larger the size, a resource is likely to be in most accessed resources. Moreover the position of the resources within the section/week also determines the accessibility of the resources. The resources at the top of the week section possess more probability to get accessed, when they are below they are less feasible to get accessed by the students.

5.2 Recommendation

The accuracy of the student classification model could have further improved if the final marks had been used instead of final grades to categorize the students. Also instead of choosing the same courses taught in different semesters, different courses could have been selected. The recommendation model of this research is purely based on the most accessed resources by good students. In addition to the recommendation based on access of good students, the research suggests integrating other recommendations such as good learners' rating and the collaborative recommendations.

5.3 Limitations of the research

The research utilized data from only a small set of courses from different semesters. Using this small set of courses it is very hard to derive a sufficiently generalized model. Further in resource recommendation model it uses the percentage of access of a resource by the good students as the classification variable but none of the courses exceed 50 learning resources. Since the Moodle courses contain limited number of resources it is more probable for a resource to get accessed by good students. Hence most of the resources will get accessed by the good students irrespective of whether they are actually "good" or "not good" as the no of resources in a course are less. In addition to that the resources with meaningless names are less probable to get accessed by the good students and resources with meaningful names are more probable to get accessed. Therefore it is possible for a good resource to get ignored by it is meaningless name itself and vice versa. Therefore there is a high possibility of skipping a good resource because of it's less meaningful name.

5.4 Future research

Using the above derived models as reference, a software could be implemented as a Moodle plugin, that would recommend the resources to the students who require assistance to identify the most relevant materials. It is also possible to employ the student classification model at the department as an early warning system to identify the students who are at a risk during the semester period and they can be monitored with close surveillance by adjusting the classification label boundary based on the requirements. Further, if the Moodle is extended to support to get ratings for each learning material, then good learner's rating can be incorporated to recommend learning materials.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

6 REFERENCES

- [1] Ghauth *et al.*, “Measuring learner’s performance in e-learning recommender systems.,” *Australasian Journal of Educational Technology*, vol. 26, no. 6, 2010.
- [2] Jie Lu, “A Personalized e-Learning Material Recommender System,” Faculty of IT, University of Technology, Sydney, Proc. of the Int. Conf. on Information Technology for Application, 2004.
- [3] Eitel J.M. Lauría and Joshua Baron, “Mining Sakai to Measure Student Performance: Opportunities and Challenges in Academic Analytics,” *School of Computer Science and Mathematics Marist College*, 2011.
- [4] Ghauth, Khairil Imran, and Nor Aniza Abdullah. "The Effect of Incorporating Good Learners' Ratings in e-Learning Content-based Recommender System." *Educational Technology & Society* 14.2 (2011): 248-257. Faculty of Computer Science and Information Technology, University of Malaya, Malaysia
- [5] Chellatamilan T and Suresh R. M, “An e-learning recommendation system using association rule mining technique,” *European Journal of Scientific Research*, Dept of Computer Science and Eng Arunai Engineering College, Tiruvannamalai, India, vol. 64, no. 2, pp. 330–339, 2011.
- [6] Kimberly Arnold, “Signals: Applying Academic Analytics,” [Online]. Available: <http://www.educause.edu/ero/article/signals-applying-academic-analytics/>. [Accessed: 13-Nov-2012].
- [7] Riccardo Mazza and Christian MILANI, “Exploring usage analysis in learning systems: Gaining insights from visualizations,” presented at the AIED Workshops (AIED’05), Faculty of Communication Sciences, University of Lugano, Switzerland, 2005.
- [8] Sten Govaerts *et al.*, “The student activity meter for awareness and self-reflection,” in *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, ACM, New York, NY, USA, 2012, pp. 869–884.

[9] Ghauth *et al.*, “Building an E-learning Recommender System Using Vector Space Model and Good Learners Average Rating,” presented at the Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference, Riga, 2009, pp. 194 – 196.

[10] Paul S. Steif and Anna Dollár, “Study of usage patterns and learning gains in a web-based interactive static course,” *Journal of Engineering Education*, vol. 98, no. 4, pp. 321 – 333, Oct. 2009.

[11] Khribi *et al.*, “Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval,” in *Advanced Learning Technologies, 2008. ICALT '08*, Santander, Cantabria, 2008, pp. 241 – 245.

[12] T. Tang and G. McCalla, “Smart recommendation for an evolving e-Learning system: Architecture and experiment,” *Association for the Advancement of Computing in Education (AACE)*, Chesapeake, VA, vol. 4, no. 1, 2005.

[13] Ivon Arroyo *et al.*, “Inferring Unobservable Learning Variables from Students’ Help Seeking Behavior,” in *Intelligent Tutoring Systems*, vol. 3220, *7th International Conference, ITS 2004*, Maceió, Alagoas, Brazil: Springer Berlin Heidelberg, 2004, pp. 782–784.

[14] David Monk, “Using Data Mining for e-Learning Decision Making,” *E-learning Services, University of Glamorgan*, vol. 3, no. 1, pp. 41–54, 2005.

[15] Zaiane O.R., “Building a Recommender Agent for e-Learning Systems,” in *Computers in Education, 2002*, vol. 1, pp. 55 – 59.

[16] Zaiane O.R., “Web Usage Mining for a Better Web-Based Learning Environment,” Department of Computing Science University of Alberta, Edmonton, Alberta, Canada., 2001.

[17] Myra Spiliopoulou, Lukas C. Faulstich, and Karsten Winkler, “A Data Miner analyzing the Navigational Behaviour of Web Users,” in *the Workshop on Machine Learning in User Modelling of the ACAI99*, Faculty of Computer Science, Otto-von-Guericke-Universität Magdeburg, Greece, 1999.

[18] Félix Castro *et al.*, “Applying Data Mining Techniques to e-Learning Problems,” in *Evolution of Teaching and Learning Paradigms in Intelligent Environment*, vol. 62, Springer Berlin Heidelberg, 2007, pp. 183–221.

- [19] Feng-jung Liu and Bai-jiun Shih, “E-Learning Activity-Based Material Recommendation System,” vol. 4, no. 4, pp. 200–207, 2007.
- [20] Cristóbal Romero, “Data Mining Algorithms to Classify Students,” in *1st Int. Conf. on Educational Data Mining (EDM’08)*, p. 187191, 2008. 49 Data Mining 2009, Computer Science Department, Córdoba University, Spain, 2009.
- [21] Jason Hunter, “JDOM and XML Parsing, Part 1,” *Oracle*, Oct-2002, [Online]. Available: <http://www.jdom.org/docs/oracle/jdom-part1.pdf>. [Accessed: 21-June-2013].
- [22] Jianqing Zhang *et al.*, “Outsourcing Security Analysis with Anonymized Logs,” presented at the Securecomm and Workshops, 2006, Dept. of Comput. Sci., Univ. of Illinois at Urbana-Champaign, Urbana, IL, 2006, pp. 1 – 9.
- [23] Sunita Beniwal and Jitender Arora, “Classification and Feature Selection Techniques in Data Mining,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 1, no. 6, Aug. 2012.
- [24] Mathew D. Pistilli and Kimbrly E. Arnold, “Mining real time academic data to enhance student success,” *American College Personal Association and Wiley Peroidocal, Inc*, 2010.
- [25] Peter Brusilovsky *et al.*, “Personalisation in e-learning environments at individual and group level,” presented at the 11th International Conference on User Modeling, Corfu, Greece, 2007.
- [26] Luis Talavera and Elena Gaudioso, “Mining Student Data To Characterize Similar Behavior Groups In Unstructured Collaboration Spaces,” Dept. Inteligencia Artificial, E.T.S.I. Inform´atica, UNED, Juan del Rosal 16, 28040 Madrid, Spain.
- [27] Nikos Manouselis *et al.*, “Simulated Analysis of MAUT Collaborative Filtering for Learning Object Recommendation,” presented at the *Proceedings of the 1st Workshop on Social Information Retrieval for Technology-Enhanced Learning & Exchange*.
- [28] “A Tutorial on Clustering Algorithms.” [Online]. Available: http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/. [Accessed: 21-Sep-2013].
- [29] “Introduction to Data Mining.” [Online]. Available: <http://www-users.cs.umn.edu/~kumar/dmbook/index.php>. [Accessed: 21-Sep-2013].

[30] Oded Maimon, "Data Mining and Knowledge Discovery Handbook," in *Data Mining and Knowledge Discovery Handbook*, Department of Industrial Engineering Tel-Aviv University Ramat-Aviv, Tel-Aviv 69978, Israel.: Springer Science + Business Media, Inc, 2005, pp. 131–142.

[31] Galit Shmueli *et al.*, *Data Mining for Business Intelligence*. John Wiley & Sons, Inc, Hoboken, New Jersey.

[32] Rhonda Delmater and Monte Hancock Jr., *Data Mining Explained: A Manager's Guide to Customer-Centric Business Intelligence*. Butterworth-Heinemann.

[33] Rapid Miner Studio [Online]. Available: <http://rapidminer.com/>. [Accessed: 02-July-2013].

[34] Zhiwen Yu *et al.*, "Ontology-Based Semantic Recommendation for Context-Aware E-Learning," Academic Center for Computing and Media Studies, Kyoto University, Japan, 2007.

[35] Paul De Bra, "Personalized Links Recommendation Based on Data Mining in Adaptive Educational Hypermedia Systems," Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands, 2007.

[36] Ahmad Baylari and Gh.A. Montazer, "Design a personalized e-learning system based on item response theory and artificial neural network approach," IT Engineering Department, School of Engineering, Tarbiat Modares University, Tehran, Iran.

[37] Moodle [Online]. Available: <https://moodle.org/>. [Accessed: 03-June-2013].

APPENDIX A: PROGRAMS

```
public class ParseXMLUsingJDOM {
    public static void main(String[] args) throws SQLException, IOException {
        SAXBuilder builder = new SAXBuilder();
        String filePath = "I:\\MSc\\Tharsan \\backup-cs5404cns-2010s3-
20130518-1701";
        File xmlFile = new File(filePath + "\\moodle - orignal.xml");
        try {
            Document document = (Document)builder.build(xmlFile);
            Element rootNode = document.getRootElement();
            List infoElements = rootNode.getChildren("INFO");
            if(infoElements.size() > 0 ){
                for(int a = 0; a < infoElements.size(); a++){
                    Element node = (Element)infoElements.get(a);
                    String name = node.getChildText("NAME");
                    String moodleVersion = node.getChildText("MOODLE_VERSION");
                    if(users != null){
                        List userList = users.getChildren("USER");
                        if(userList.size() > 0) {
                            for(int l = 0; l < userList.size(); l++){
                                Element userElement = (Element)userList.get(l);
                                String userName = userElement.getChildText("USERNAME");
                                String userFirstName= userElement.getChildText("FIRSTNAME");
                                String userEmail = userElement.getChildText("EMAIL");
                                String userTimeZone = userElement.getChildText("TIMEZONE");
                                String userCountry = userElement.getChildText("COUNTRY")
                            }
                        }
                    }
                }
            }
        }
    }
}
```



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Figure -1 XML Parsing

```

USE Moodle

GO

DECLARE @CourseId INT

DECLARE @ResourceId VARCHAR(250)

DECLARE @TotalNumberOfResourceAccessInCourse INT

DECLARE @numberOfAccessOfSingleResource INT

DECLARE @percentageOfAccessOfSingleResource NUMERIC(16,2)

DECLARE @getMoodleResources CURSOR

SET @getMoodleResources = CURSOR FOR

SELECT [CourseId],[Id]

FROM [Moodle].[dbo].[ModResource]

OPEN @getMoodleResources

FETCH NEXT

FROM @getMoodleResources INTO @CourseId, @ResourceId

WHILE @@FETCH_STATUS = 0

SET @TotalNumberOfResourceAccessInCourse = (SELECT COUNT(*)

FROM (SELECT DISTINCT logUserId, logInfo

FROM [Moodle].[dbo].[log]

WHERE logModule = 'resource' AND logAction = 'view' AND

CourseId = @CourseId) AS X )

SET @numberOfAccessOfSingleResource = (SELECT COUNT(*)

FROM ( SELECT DISTINCT logUserId, logInfo

FROM [Moodle].[dbo].[log]

WHERE logModule = 'resource' AND logAction = 'view' AND

CourseId = @CourseId AND logInfo = @ResourceId) AS Y)

SET @percentageOfAccessOfSingleResource = @numberOfAccessOfSingleResource * 100

/ CAST(@TotalNumberOfResourceAccessInCourse AS DECIMAL(16,2))

```

Figure - 2 Calculate the percentage of access by good students

APPENDIX B: RESULTS

Table 1 Accuracy of student model with manual split with three class labels

Used data set	Average ratio greater than 0.5	Outliers removed first time	The label boundary changed up	The label boundary changed down	Outliers removed second time
Algorithms					
Decision Tree	54.90	55.56	53.33	53.33	53.57
Neural Networks	60.78	46.67	53.33	44.44	57.14
SVM	50.98	48.89	53.33	48.89	53.57
k-NN	33.33	40.00	48.89	42.22	46.43
Naive Basis	45.10	57.78	48.89	48.89	57.14
Rule Induction	45.10	42.22	53.33	35.56	53.57
Perceptron	45.10	17.78	51.11	40.00	50.00
Linear Regression	47.06	57.78	55.56	51.11	50.00
Polynomial Regression	31.97	42.22	24.44	33.33	42.86
Vector Linear Regression	19.61	24.44	24.44	24.44	21.43
Gaussian Process	39.22	46.67	53.33	48.89	50.00

Table 2



University of Moratuwa, Sri Lanka.

Electronic Theses & Dissertations

RMSE of student model with manual split with three class labels

www.lib.mrt.ac.lk

Used data set	Average ratio greater than 0.5	Outliers removed first time	The label boundary changed up	The label boundary changed down	Outliers removed second time
Algorithms					
Decision Tree	0.63	0.62	0.62	0.60	0.62
Neural Networks	0.58	0.70	0.63	0.71	0.61
SVM	0.60	0.62	1.08	0.62	0.60
k-NN	0.82	0.78	0.72	0.76	0.73
Naive Basis	0.61	0.59	0.63	0.62	0.62
Rule Induction	0.72	0.74	0.65	0.62	0.62
Perceptron	0.73	0.88	0.69	0.77	0.71
Linear Regression	0.65	0.55	0.86	0.65	0.64
Vector Linear Regression	0.90	0.87	0.87	0.87	0.89
Gaussian Process	0.78	0.73	0.70	0.71	0.71

Table 3

Accuracy of student model with manual split with four class labels

Used data set	The label boundary changed down	The label boundary changed up	Outliers removed second time	Outliers removed third time
Algorithms				
Decision Tree	33.33	40.00	42.86	36.84
Neural Networks	37.78	31.11	35.71	31.58
SVM	33.33	42.22	53.57	36.80
k-NN	33.33	37.78	39.29	42.11
Naive Basis	37.78	37.78	39.29	31.58
Rule Induction	33.33	33.33	46.43	47.37
Perceptron	15.56	40.00	39.29	36.84
Linear Regression	24.44	40.00	50.00	42.11
Polynomial Regression	33.33	33.33	28.57	26.32
Vector Linear Regression	24.44	24.44	21.43	15.79
Gaussian Process	40.00	44.44	39.29	47.37



University of Moratuwa, Sri Lanka.

Accuracy of student model with manual split with five class labels

Electronic Theses & Dissertations


www.lib.mrt.ac.lk

Used data set	The label boundary changed down	The label boundary changed up	Outliers removed second time	Outliers removed third time
Algorithms				
Decision Tree	33.33	20.00	28.57	26.32
Neural Networks	28.89	28.89	21.43	41.11
SVM	31.11	31.11	28.57	26.32
k-NN	28.89	26.67	25.00	26.32
Naive Basis	26.67	22.22	25.00	21.05
Rule Induction	17.78	26.67	25.00	31.58
Perceptron	11.11	17.98	14.29	21.05
Linear Regression	35.56	28.89	39.29	31.58
Polynomial Regression	31.11	20.00	14.29	21.05
Vector Linear Regression	24.44	24.44	21.43	15.79
Gaussian Process	35.56	33.00	25.00	31.58

Table 5 Accuracy of resource model with random split with three class labels

Used data set	% of access by good students at split borders 2.25 and 1.8	% of access by good students at split borders 2.5 and 1.8	% of access by good students at split borders 2.6 and 1.8	% of access by good students at split borders 2.25 and 1.9	% of access by good students at split borders 2.25 and 1.7
Algorithms					
Decision Tree	88.60	86.60	82.47	87.63	88.60
Neural Networks	88.60	89.69	86.60	87.63	88.60
SVM	71.13	78.35	85.57	74.23	71.13
k-NN	60.82	60.82	63.92	59.79	60.82
Rule Induction	79.38	86.60	82.47	83.51	79.38
Naive Basis	78.35	77.32	82.47	78.35	78.35
Perceptron	43.30	37.11	49.48	37.11	43.30
Linear Regression	83.19	89.69	81.44	71.38	83.19

Table 6 Accuracy of resource model with random split with four class labels



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Used data set	% of access by good students at split borders 2.25-1.8-1.4	% of access by good students at split borders 2.25-1.8-1.4	% of access by good students at split borders 2.25-1.8-1.4
Algorithms			
Decision Tree	70.10	81.44	81.44
Neural Networks	75.26	87.63	85.57
SVM	74.23	64.95	62.89
k-NN	51.55	56.70	56.70
Rule Induction	78.35	79.38	77.32
Naive Basis	69.07	77.32	77.32
Perceptron	45.36	45.36	45.36
Linear Regression	72.16	64.95	63.92