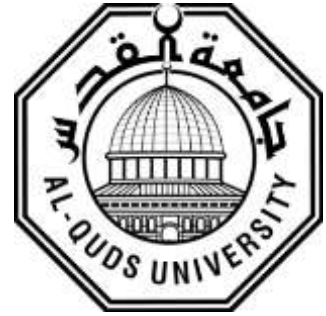


**Deanship of Graduate Studies  
Al-Quds University**



**E-Learner Recommendation Model Based on Level of  
Learning Outcomes Achievement**

**Abeer Hasan Abdelrahim Mousa**

**M.Sc. Thesis**

**Jerusalem-Palestine**

**1439 / 2018**

# **E-Learner Recommendation Model Based on Level of Learning Outcomes Achievement**

**Prepared By:**  
**Abeer Hasan Mousa**

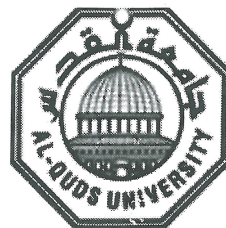
**B.Sc. Computer Engineering from - Birzeit University  
- Palestine.**

**Supervisor: Dr. Badie Sartawi**

**A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science /Department of Computer Science Faculty of Science &Technology /Deanship of Graduate Studies /Al-Quds University.**

**1439/2018**

**Deanship of Graduate Studies**  
**Al-Quds University**  
**Computer Science**



### **Thesis Approval**

This thesis is approved for recommendation to the Graduate Council.

**Prepared by:** Abeer Hasan Mousa

**Student ID No:** 21312128

**Supervisors:** Dr.Badie Sartaw

**Master thesis submission and acceptance date:** 03/06/2018

**The names and signatures of examining committee members are as follows:**

1.Head of committee: Dr.Badie Sartaw

Signature.....*Badie Sartaw*

2. Internal Examiner: Dr. Kamel Hashem

Signature.....*Kamel Hashem*

3. External Examiner: Dr. Ahmad Ewais

Signature.....*Ahmad Ewais*

Jerusalem-Palestine

1439/2018

## **Dedication**

I dedicate my thesis to my family. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears. I also dedicate this dissertation to my beloved husband and my wonderful kids, and I will always appreciate their patience, encouragement and all they have done for me. I also dedicate this dissertation to anyone who have supported me throughout the process.

Thank you all

**Abeer Hasan Mousa**

**Declaration**

I certify that this thesis submitted for the degree of Master is the result of my own research, except where otherwise acknowledged, and that this thesis or any part of the same has not been submitted for a higher degree to any other university or institution.

Signed \_\_\_\_\_

## **Acknowledgement**

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

I would like to express my deep and sincere gratitude to my research supervisor, Dr. Badie Sartawi who inspired and taught me since the days I began working on this thesis. I appreciate all his contributions of time, ideas and support to bring this work to the life.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Ahmad Ewais and Dr. Kamel Hashem for their encouragement, insightful comments, and questions.

My sincere thanks also go to Dr. Rashid Jayousi for his continues support, encouragement and bright ideas and for Dr. Jihad Najjar for his guidance and observations to work in the first days.

I would like to acknowledge my father Dr. Hasan Yousef for his continues help and support. And my mother whose love and guidance are with me in whatever I pursue. Also, I owe a debt of gratitude to my loving and supportive husband Dr. Rami Mousa and my three wonderful children Omar, Mumen and Lamar.

## Abstract

Students in any learning environment differ in their level of knowledge, achieved learning outcomes, learning style, preferences, misunderstand and attempts in solving and addressing problems when their expectations are not met.

When a student searches the web as an attempt to solve a problem, he suffers from the large number of resources which are, in most cases, not related to his “needs”, or may be related but complex and advance. The result of his search might make him more confused, scattered, depressed and finally result in wasting his time which – in some cases -may have negative effects on his achievements.

From here comes the need for an intelligent learning system that can guide students based on their needs. This research attempts to design and build an educational recommender system for a web-based learning environment in order to generate meaningful recommendations of the most interested and relevant learning materials that suit students’ needs based on their profiles<sup>1</sup>. This can be achieved by accessing students’ history, exploring their learning navigation patterns and making use of similar students’ experiences and their success stories.

The study proposed a design for a hybrid recommender system architecture which consists of two recommendation approaches: the content and collaborative filtering.

The study concentrates on the collaborative recommender engine which will recommend learning materials based on students’ level of knowledge, looking at active students' profiles, and achievements in both learning outcomes and learning outcomes levels making use of similar students’ success stories and reflecting their good experience on active student who are in the same level of knowledge.

The design of the collaborative recommender engine includes the “learning” module from which the engine learns past students’ access pattern and the “advising” module from which the engine reflects the experience of similar success stories on active students.

The content base recommender engine with its suggested stages is considered as future work, the research used the k-mean cluster algorithm to find out similar students where five distance function are used: Euclidean, Correlation. Jaccard,

---

<sup>1</sup> A student profile reflects his active courses, achieved learning outcomes and his level of knowledge.

cosine and Manhattan. The cosine function shows to be the most accurate distance function with the minimum

SSE but the highest processing time that doesn't differ a lot when compared the rest functions. The best number of clusters for the selected dataset was determined using three methods Elbow, Gap-statistic and average Silhouette approach where the best number of cluster shows to be three. The research used the two result rating matrices of similar good and good students with Learnings material in order to calculate learning material weights and rank them based on highest weights which results in a final recommendation list.

## **Keywords**

Recommender system, digital libraries, learning outcomes, collaborative filtering, Ranking, adaptive e-learning systems



## Table of Content

Dedication .....	i
Acknowledgement.....	ii
Abstract .....	iii
List of Tables.....	vii
List of Figures .....	vii
Terms and Definitions .....	xii
Introduction .....	1
<b>1.1 Research Overview</b> .....	2
<b>1.2 Motivation</b> .....	2
<b>1.3 Research Questions</b> .....	3
<b>1.4 Research Goals</b> .....	3
<b>1.5 Research Contributions</b> .....	4
<b>1.6 Literature Reviews</b> .....	4
<b>1.7 Thesis Outline</b> .....	8
Educational Theory .....	9
<b>2.1 Learning Approaches</b> .....	10
<b>2.2 Learning Taxonomy</b> .....	10
<b>2.2.1 Bloom's taxonomy theory</b> .....	11
<b>2.2.2 Bologna Declaration</b> .....	14
<b>2.3 E-Learning and adaptive e-Learning systems</b> .....	15
<b>2.4 Intelligent Tutoring System</b> .....	16
Knowledge Discovery .....	18
<b>Introduction</b> .....	19
<b>3.1 Data Mining in Recommender Systems</b> .....	19
<b>3.2 Sampling</b> .....	20
<b>3.3 Data Distribution Models</b> .....	22
<b>3.4 Distance and Similarity Measures</b> .....	27
<b>3.4.1 Euclidean distance</b> .....	27
<b>3.4.2 Correlation Distance (or Pearson correlation distance)</b> .....	28
<b>3.4.3 Jaccard Similarity Coefficient (Index)</b> .....	30

3.4.4 Cosine Similarity .....	31
3.4.5 Manhattan (or City-Block) Distance .....	32
3.5 Analysis Process.....	33
3.5.1 Cluster Analysis.....	34
3.5.2 K-means Algorithm.....	34
Recommender System.....	51
Introduction .....	52
4.1 Recommender system.....	52
4.2 Recommendation in learning management systems .....	53
4.3 Recommendation Approaches.....	54
4.3.1 Content-based recommender system.....	54
4.3.2 Collaborative recommendation approaches .....	55
4.3.3 Hybrid Recommendation Approaches .....	56
4.3.4 Challenges and Issues.....	56
4.4 Collaborative Recommender system algorithms.....	57
4.4.1 Memory-based algorithms.....	57
4.4.2 Model-based algorithms.....	58
Methodology .....	59
Introduction .....	60
5.1 Recommender Engine Architecture .....	60
5.2 Collaborative Recommender Approach.....	64
5.3 Data preparing – Creating the Dataset .....	66
5.4 Data Model.....	69
5.5 Matrix types .....	71
5.6 Students’ responses indicators .....	73
5.7 Recommender Engine Setup – Inputs and configuration.....	74
Results and Discussions .....	77
6.1 Generated Data Statistics .....	78
6.1.1 Students’ achievements in a learning outcome .....	78
6.1.2 Students’ achievements in a Course .....	79
6.1.3 Students’ trends in levels of learning outcomes.....	80
6.2 Students’ Similarity.....	81
6.2.1 Building the Matrix.....	82
6.2.2 Choosing the number of clusters “K”.....	87
6.2.3 Applying k-means method using five different distance methods .....	93
6.2.4 Similarity Results .....	96

6.2.5 Applying K-means method using R .....	105
6.3 Learning Material Recommendation .....	108
Conclusion.....	112
7.1 Conclusion.....	113
7.2 Limitations .....	114
7.3 Future Work .....	114
References .....	115
المخلص.....	140

## List of Tables

Table 2. 1 Bloom’s taxonomy vs Anderson/Krathwohl levels, definitions and verbs	13
Table 5. 1 Database tables classified into three layers.....	70
Table 6. 1 Students’ trends in the levels of learning outcome in course.....	81
Table 6. 2 Learning outcomes and learning outcomes levels for each course, learning outcomes levels. ....	84
Table 6. 3 Students achievements in levels of learning outcomes.....	85
Table 6. 4 Active to Similar Student Absolute Average Difference.....	101
Table 6. 5 Rating matrix of best students.....	108
Table 6. 6 Rating matrix of best students.....	109
Table 6. 7 Two recommendation list is generated for good similar students and good students for an active course .....	109
Table 6. 8 Recommendation Lists based on good students and similar good students weight.....	111

## List of Figures

Figure 2. 1: Original Bloom Taxonomy primary level in cognition domain (1956)..	12
---	----

Figure 2. 2: Revised Taxonomy (2000-2001), Anderson and Krathwohl primary level in cognition domain. ....	12
Figure 2. 3: Adaptive e-learning Architecture [24].....	16
Figure 3. 1: Main steps and methods of a data mining problem .....	20
Figure 3. 2: Normal distribution for student marks based on random generated data set .....	23
Figure 3. 3: Poisson distribution .....	24
Figure 3. 4: Binomial distribution.....	25
Figure 3. 5: Probability mass function .....	26
Figure 3. 6: The Euclidean Distance between 2 variables in 3-dimensional space ....	28
Figure 3. 7: Student's achievement in 30 learning outcome.....	29
Figure 3. 8: Correlation coefficients between students using data analysis in excel ..	29
Figure 3. 9: Correlation distance between students using R where correlation distance is complement for Correlation coefficient .....	29
Figure 3. 10: Learning outcome achievement summary per student .....	30
Figure 3. 11: Similarity between students based on Jaccard distance .....	31
Figure 3. 12: Similarity based on Jaccard distance .....	31
Figure 3. 13: Two-dimensional illustration of Euclidean distances between two points .....	32
Figure 3. 14: Cosine similarity between a set of four students .....	32
Figure 3. 15: Manhattan distance between four students using R .....	33
Figure 3. 16: Two-dimensional illustration of city-block distances between two points .....	33
Figure 3. 17: Basic k-means algorithm steps .....	35
Figure 3. 18: 200 hundred student marks in 30 learning outcomes represented in R. ....	36
Figure 3. 19: Scatterplot matrix of the first five learning outcomes for 200 students .....	37
Figure 3. 20: k-mean cluster for the first ten learning outcome of 200 student using R .....	38
Figure 3. 21: Number of Students in each cluster .....	38
Figure 3. 22: Four cluster plot for the first ten learning outcomes of 200 student .....	39
Figure 3. 23: Plot of k-means of four-clusters solution for student achievement in learning outcome.....	40

Figure 3. 25: Applying Elbow method on all learning outcome for 200-students data set .....	41
Figure 3. 24: Applying Elbow method on the first 10 learning outcome for 200 students data set .....	41
Figure 3. 26:Applying Silhouette method on all learning outcomes for 200 students	42
Figure 3. 27: Applying Silhouette method on the first 10 learning outcomes for 200 students.....	42
Figure 3. 28: Applying Gap Statistic method on all learning outcomes for 200 students.....	43
Figure 3. 29:Applying Gap Statistic method on the first 10 learning outcomes for 200 students.....	43
Figure 3. 30: Applying NbClust package on student data set.....	44
Figure 3. 31:D index to determine the number of index .....	45
Figure 3. 32: Cluster plot for the first ten learning outcomes of 200 students where $k = 2$ .....	46
Figure 3. 33: Cluster cohesion .....	47
Figure 3. 34: Cluster separation .....	47
Figure 3. 35:Plot of within-groups sum of squares error against number of clusters .	48
Figure 3. 36: SSE against the number of tested clusters for both the actual and 250 randomized matrices .....	49
Figure 3. 37: Absolute difference between the actual and random SSE against the cluster solutions.....	50
 Figure 4. 1: matrix of users' rates on nine items.....	 55
 Figure 5. 1: Overall architecture for an E-Learning recommender engine .....	 62
Figure 5. 2: Block diagram for the proposed model of Educational Recommender System – Collaborative Approach .....	66
Figure 5. 3: Normal distribution of students' marks .....	68
Figure 5. 4:Data model in recommender engine .....	70
 Figure 6. 1: Filtering options for measuring students' achievements .....	 78

Figure 6. 2: Marks distribution for a learning outcome .....	79
Figure 6. 3: Students achievements in “Mathematical Expression and Reasoning for Computer Science” in 2007 .....	80
Figure 6. 4: Generated sparse matrix for student learning outcomes where the student’s Id is printed at the beginning .....	83
Figure 6. 5: building Student 17012006 spars Matrix for Learning Outcomes .....	83
Figure 6. 6: The spars matrix of learning outcomes levels for student 17012006.....	86
Figure 6. 7: Log file while generating the sparse matrix for student's learning outcomes levels .....	86
Figure 6. 8: Elbow method for sparse matrix of learning outcome Levels.....	88
Figure 6. 9:Elbow method for sparse matrix of learning outcome .....	88
Figure 6. 10: Elbow method for dense matrix of levels of learning outcome.....	89
Figure 6. 11: Elbow method for dense .....	89
Figure 6. 12: Silhouette approach on sparse matrix of learning outcomes Levels .....	90
Figure 6. 13: Silhouette approach on sparse .....	90
Figure 6. 14:Applying average Silhouette approach on dense matrix of levels of learning outcomes .....	91
Figure 6. 15:Applying average Silhouette approach on dense matrix of learning outcomes .....	91
Figure 6. 16: Gap statistic method on sparse matrix of learning outcomes .....	92
Figure 6. 17: Parameters for clustering process .....	93
Figure 6. 18:Similarity Results based on clustering process .....	94
Figure 6. 19: Statistics dashboard for clustering process.....	94
Figure 6. 20: General statistics on different distance function .....	95
Figure 6. 21: Average time and memory usage indicators .....	95
Figure 6. 22: Required to actual number of cluster indicators.....	96
Figure 6. 23: Clustering process page showing the results of clustering operation....	97
Figure 6. 24: Abdelrahman and Haleema marks in course “CSC495H1” .....	98
Figure 6. 25: a Hussein and Haleema marks in course “CSC495H1” .....	98
Figure 6. 26: Ayoub and Haleema marks in course “CSC495H1” .....	99
Figure 6. 27: Sana and Haleema marks in course “CSC495H1” .....	100
Figure 6. 28: Active to similar students’ absolute average using correlation vs cosine distance.....	102

Figure 6. 29: Active to similar students' absolute average using cosine distance when k=6 .....	102
Figure 6. 30: Active to similar students' absolute average using cosine distance when k=4 .....	102
Figure 6. 31: Gap between junior active & similar student marks - batch 421 .....	104
Figure 6. 32: Gap between senior active & similar student marks .....	105
Figure 6. 33: Convergence of similar students' marks .....	105
Figure 6. 34: K-mean analysis for senior and junior students .....	106
Figure 6. 35: using the clusplot command to show clustering results in R.....	106
Figure 6. 36: Cluster plot for junior students' analysis.....	107
Figure 6. 37:Cluster plot for senior students' analysis.....	107
Figure 6. 38: Recommender engine setup page .....	110

## List of Appendices

Appendix A .....	118
<b>SSE R Script</b> .....	119
<b>Generating Data Set Scripts</b> .....	123
Appendix B .....	134

## Terms and Definitions

<b>Active Student:</b>	Is an under graduate student who register for the current academic semester for at least one course and is waiting a recommendation list of learning material
<b>Active Course:</b>	Is the course which is taken by an active student in the current academic semester.
<b>Similar Student:</b>	Are those students who are clause in their achievement to the active student.
<b>Good Students:</b>	Those students who achieved high marks in an active course.
<b>Senior Student</b>	Students in his fourth year in university.
<b>Junior Student</b>	Student in his third year in university.
<b>High Marks:</b>	Are marks which are higher than the mark configured in the recommender engine setup page; refer to section 5.7 for more details.
<b>Learning Material:</b>	Are those helping material such as papers, presentations, summaries, videos and any other helping material which helps the student to enhance his achievement
<b>Strongly Recommended Materials</b>	materials are those materials which appear to have a strong relationship with a student's better significant results in a certain course.
<b>Users:</b>	Are those who use any type of recommender system, users could be customers, students, employees ...etc.
<b>Active users</b>	Is the user who will be recommended with a set of items that seems to be useful for him based on the recommender system approach.
<b>Student current status</b>	Is defined by the set of learning outcomes accomplished by his achieved marks.
<b>Items</b>	General term for the set of output recommendation, Items Could be products, learning material, CVs and others. This depends on the environment in which the recommender engine works.



<b>tf-idf representation</b>	Term frequency–inverse document frequency, a numerical statistic that reflect how important a word is to a document
<b>Efficiency</b>	Is the computational complexity of the algorithm
<b>ILT</b>	Is the driver of research and development around Learning Technologies and the relevant learning platforms, standards and practices at Al-Quds University
<b>QLearn</b>	platform will be developed to address the pitfalls in current online learning platforms like (Learning (Course) Management Systems). Qlearn is an outcome-based system and will enable effective mapping between learning objectives, learning objects and assessment using keywords mapping between learning objects and learning outcomes

## Chapter 1

### **Introduction**

## **1.1 Research Overview**

Instructors and academic staff in the educational institutes have a large amount of learning materials depending on their courses. These learning materials are a perfect match for course learning outcomes and students' different needs. At the same time, students are always searching for learning material related to their courses in order to increase their understanding and achievement level. But when students search the web, they really suffer from the large number of resources, which in most cases are not related to their needs and make them more confused, scattered and depressed. All of this result in wasting time and may have negative effects on student achievements.

An e-Learning Recommendation System (LRS) is a solution for this problem. The LRS will match between learning materials and students' needs based on their academic profile, achievement level and learning outcomes required by their active courses.

LRS will build its experience and decide on a "Strongly Recommended Learning Material" based on previous similar students' profile, their achievements in certain courses and related Learning materials that they used.

## **1.2 Motivation**

Most of educational recommender system focuses on the accuracy of predicting learning materials and how much these learning materials match a student's active course. The main target in any adaptive e-learning system is to minimize the gap between student needs and the knowledge provided in order to make the learning process easier and more interesting.

This research aims to design an LRS that focuses on the accuracy of predicting learning materials based on student's needs and gap of knowledge between him and the provided course, making use of the learning patterns of similar students who were in the same level of knowledge as the active student but succeed to achieve high marks in the active course.

Students' knowledge was determined based on their academic profile and achievement level of learning outcomes. The LRS will suggest learning materials based on students' needs and will result in a "strongly recommended learning material" based on students' weaknesses, which results in building a coherent knowledge and a deeper understand for their course.

Students' needs and weaknesses are discovered through their academic achievement in learning outcomes and in learning outcome levels for each taken course and its related courses. So, a student who is weak in the "understanding" level needs more learning material that help him advance in this level.

The suggested LRS will build its experience significantly based on better results and achievement of previous students who took the same course and were in the same knowledge level as the active student but differ in the high achievement they gain in the active course. This can be fulfilled by discovering the relation between the high achievement of those students and their interaction with learning materials so the LRS can reflect their good experience on active students.

### **1.3 Research Questions**

1. How can we build an effective LRS that matches between course learning outcomes, learning materials and students' needs in order to gain higher educational achievement levels?
2. How does LRS measure the student needs? In other word, how can the LRS measure student achievements level in each learning outcome related to the course, and decide learning material based on that?
3. How can the LRS build its experience in order to enhance its recommendation for students?
4. A group of students may all register in the same course and may all have completed the same previous courses, but of course their achievement level differs. So, the gained knowledge will differ from one student to another.

Can the suggested LRS take this point into consideration and give its recommendation to each student based on their Knowledge (not just on achieved courses)? How can these learning materials being ranked, fit students' needs and draw the road map of correct Knowledge construction?

### **1.4 Research Goals**

1. To solve the overload problem and huge number of learning materials when using digital learning libraries which may cause students to become confused, scattered and depressed.

2. To build an effective LRS that matches between student knowledge (which is based on his academic profile and achievements) and his active courses' learning outcomes, to find out learning materials which are suitable for the students' needs in order to gain higher educational achievements level.
3. To rank the recommended results according to students' needs, starting from the easiest learning material to the hardest. In other words, starting from the strongly recommended materials
4. To find out an effective mechanism in which the suggested LRS will build its experience in order to enhance its recommendation for students.

## **1.5 Research Contributions**

1. Assessing student knowledge level based on their achieved learning outcomes and outcomes levels.
2. Find similar students with the same level of knowledge.
3. Make use of similar students' success stories and reflect their experience in using learning material on active students in order to enhance their achievements and help them gain higher marks.
4. The design of the suggested recommender engine guarantees that the engine will overcome the cold start problem due to the content base approach which will work as primary recommender in the absence of history information on either course or student profile.
5. Recommending learning materials based on student knowledge, making use of student profile and achievement in learning outcomes and learning outcome levels.

## **1.6 Literature Reviews**

The aim of this research is to build an LRS that evaluates student knowledge based on their academic profile and achievement level, then suggests learning material based on student profile. Researchers in this area have provided different approaches.

Jamil Itmazi (2010) presents a new proposal of recommendation algorithm in learning management system which could automatically recommend suitable learning objects from a big list of digital libraries based on an integration between the digital library and the learning management system. The algorithm is considered as a hybrid recommendation system which consists of some RS approaches; content-based system, collaborative filtering, rule-based filtering and demographic-based system.

The study depends on content-based system as a **primary approach** to detect similarities among learning items of current course to retrieve a list of related learning objects. The retrieved list will be subjected to teacher recommendation in order to classify the heights priority objects. The list will then pass into a collaborative filter, which acts as a complementary approach to organize the priorities of the recommendations in which all similar students with same profiles (department and school) are found to calculate their average rating for learning objects.

After that, the list of learning objects will pass through a demographic base filter which is related to student profile such as student specialization, study year level, faculty and department.

Finally, the list will pass through a rule-based filtering which will filter the incoming recommended digital objects upon a set of rules which were put by the system administrator or the students themselves.

In the same context, Khairil Ghauth & Nor Abdullah (2011) propose a new e-learning recommender system framework that uses content-based filtering and good learners' ratings to recommend learning materials. The research depends on the student profile and achievements in addition to the strategy of good readers to recommend learning objects for the students.

Good readers are those students who completed a course with a mark over 80% by using other learning material and rating them. The researcher depends on those rating to recommend learning materials to other students. The results show a significantly positive impact on the learning outcome of the students by at least 13.8%. The proposed recommender system is prone to the 'cold start' problem, in which the system is not able to calculate or predict the good learners' rating for the items if the good learners' ratings are unavailable.

Tiffany Ya TANG (2003) introduced smart recommendation for an evolving E-Learning system. The system has the ability to find relevant content on the web and then personalize and adapt them based on the system observations of its learners and their accumulated rating.

The system can crawl the web to get new papers and connect these papers with system courses using a “paper maintenance model” which crawl citeseer for new papers. The system cluster learners according to their browsing activities to find out similarities between them. The paper maintenance model add papers when crawling the web and deleting papers according to learners assessment.

Two major techniques were adopted: collaborative filtering and data clustering. There are two kinds of collaboration in the system, one is the collaboration between the system and users; and the other is the collaboration between the system and the open Web.

John Tarus, Niu and Khadidja (2017) proposed a recommendation technique which take into consideration the learner characteristic such as learning style, study level and skill level which can influence the learner’s preferences learning. The recommendation technique combines a collaborative filtering and ontology to recommend personalized learning materials to online learners.

The recommender system used the learner ontology in order to incorporate the characteristic of learners in the recommendation process to achieve better personalization and accuracy in e-learning recommendations. Also, the ontological knowledge is used by the recommender system at the initial stages in the absence of ratings to alleviate the cold-start problem. So, both ratings and ontological knowledge are used in computing similarities and generating recommendations for the learner.

Sunita B Aher and Lobo L.M.R.J. (2012) proposed a course recommender system that uses a combination of machine learning algorithms to identifying the behavior of students interested in a particular set of courses. Different combinations of data mining algorithm like (1)classification & association rule algorithm, (2)clustering & association rule algorithm, (3)association rule mining in classified & clustered data and (4)combining clustering & classification algorithm in association rule algorithms or simply the association rule algorithm.

The study looked mainly at the number of students interested in each course in the e-learning system and found out that the combination clustering, classification & association rule algorithm is the best combination.

Maria Gogaa & others (2014) designed a framework of intelligent recommender system which can predict first year student performance and recommend necessary actions for improvement. The study believes that various predictors at various time and different locations contribute to the outcome of students and evidence that students' background information contribute immensely to the early prediction of student success.

Pensri Amornsirilaphachai (2013) synthesize a learning model using the Student Teams - Achievement Divisions (STAD) technique with a suggestion system according to learners' capability to decrease learners' weakness[2]. the research results in a learning model comprises of 5 modules that are (1) test module, (2) evaluation module, (3) suggestion module, (4) community module and (5) knowledge bank module.

The results derived from experts' evaluation are disclosed that the model is appropriate to 3 aspects that are (1) learning content, (2) design based on theories and (3) media and technology. Moreover, the experts accept the usability of the model in a high level.

In fact, the architecture of the suggested recommender system in this study is a hybrid of two recommender approaches: Content-based and collaborative filtering. Each approach works separately and gives its own recommended list of learning materials and learning materials weight, the final stage of the recommender system is the ranking model which results in a final recommender list with the items and their final weights ranked from the highest to lowest.

This architecture guarantees that the LRS will keep working even if no history for a course exists (cold start problem), in this case the recommendation will depend mainly on the content base filter whereas the collaborative filtering approach will gain experience with time. On the other hand, the collaborative filtering approach is considered as a main recommender engine whenever a history exists and can give more accurate results and gain more experience with time. The collaborative filtering works mainly on the similarity among students based on their achievement



in previous learning outcomes, it suggests best match learning materials - which match active courses- based on previous success stories for similar students.

## **1.7 Thesis Outline**

The thesis in chapter one gives an overview on the research and declares the research motivation, questions, goals, contribution and also gives fast reviews for some studies in the same context.

Chapter two gives a small review on the new trends in education and learning theory which concentrates on learning outcomes and their importance in measuring students' knowledge level.

In chapter three some pattern and methodologies in knowledge discovery are highlighted, as they are used in the research, whereas in chapter four a fast review on recommender approaches and recommender engines in e-learning is given.

Chapter five describes the architecture and methodology of the suggested recommender engine and chapter six discusses and analyzes the results. Chapter seven summarizes the research and the research conclusion.

## Chapter 2

# **Educational Theory**

## **2.1 Learning Approaches**

Designing learning modules and programs have two approaches: “Teacher-Centered” and “student-centered”. The “Teacher-Centered” approach is considered the traditional way of teaching where teachers decide on the content they intend to teach on the program, and plan how to teach the content in the learning period [16]. Kathly drown (2003) mentioned new challenges facing classroom teachers such as legislative mandates for school renewal, diverse student needs and technological advices, which makes this approach not working for a growing number of diverse, student population.

The “student-centered” approach, which is also referred to as “Outcome Based”, is considered as the new international trends in education. According to the constructive learning theory, learning is defined as “active process in which learners are active sense makers who seek to build coherence and organized Knowledge”. This constructive learning theory acts as the source of developing this new trend of learning [50]. “student-centered” approach focuses on what the students are expected to be able to do at the end of the program or learning period. [16]

Fan Yang and Zheng-hong Dong (2017) in their learning theory of constructivism considered each student as a unique individual with personalized needs, learning styles, learning preferences, knowledge levels, and knowledge backgrounds. Under their learning theory, teaching approaches are designed according to learning outcomes and does not focus on the teacher-centered learning environment. It puts more emphasis on self-paced learning by providing access to education at any time, any place, and taking into account students’ differences.

## **2.2 Learning Taxonomy**

A taxonomy is a classification system which is categorized as shared language that orders things in some way.

Learning taxonomy is defined as a tool which “provides the criteria of assessing student learning performance to see if students can achieve their learning outcomes” [17].

According to ECTS Users’ Guide, learning outcomes are defined as “statements of what the individual knows, understands and is able to do on completion of a learning process” [48]

(American Association of Law Libraries) defines learning outcome as “statements that specify what learners will know or be able to do as a result of a learning activity. Outcomes are usually expressed as knowledge, skills or attitudes”.

(University of New South Wales, Australia) defines learning outcomes as “explicit statements of what we want our students to know, understand or be able to do as a result of completing our courses.”

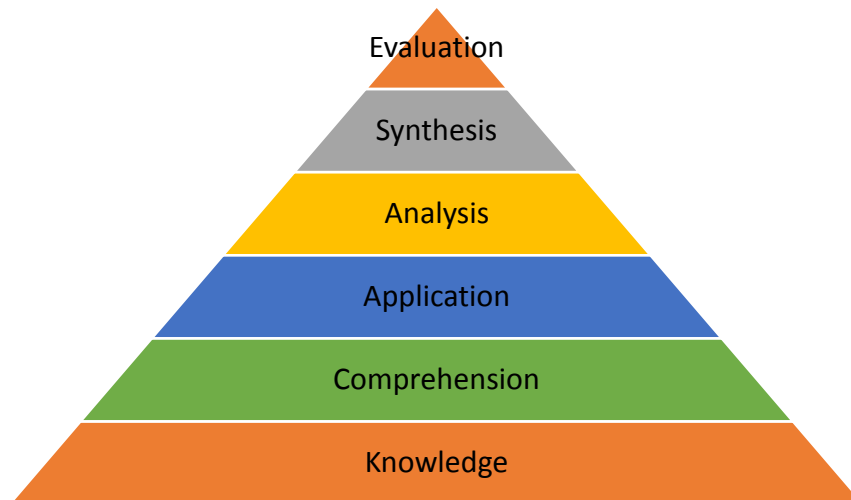
The University of Toronto in its “Developing Learning Outcomes” guide defines learning outcomes as “statements that describe the knowledge or skills students should acquire by the end of a particular assignment, class, course, or program, and help students understand why that knowledge and those skills will be useful to them. They focus on the context and potential applications of knowledge and skills, help students connect learning in various contexts, and help guide assessment and evaluation.”

Learning taxonomy is categorized in three domains: cognitive (thinking), affective (Emption), and psychomotor (kinesthetic). Each domain has a taxonomy associated with it and is divided into several levels, each learning outcome is evaluated by one of these levels. The most common learning taxonomy in cognition domain is Bloom’s taxonomy.

### **2.2.1 Bloom's taxonomy theory**

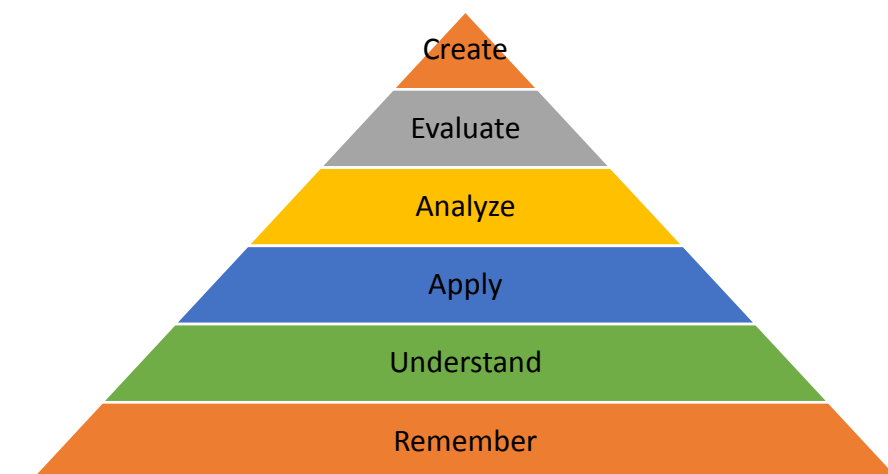
Bloom’s Taxonomy which is also referred as original Taxonomy “is a classification of the different objectives and skills that educators set for their students (learning objectives)”. It is being increasingly widely used in the design and assessment of learning outcomes, it is a set of three hierarchical models used to classify educational learning objectives into levels of complexity and specificity.

The original levels by Bloom et al. (1956) were ordered as follows: knowledge, comprehension, application, analysis, synthesis, and evaluation as shown in Fig. 2.1.



**Figure 2. 1: Original Bloom Taxonomy primary level in cognition domain (1956).**

Original Bloom Taxonomy proposed that our thinking can be divided into six increasingly complex levels from the simple recall of facts at the lowest level to evaluation at the highest level. The revision of original Taxonomy was developed in much the same manner in 2000-2001 by Anderson and Krathwohl. One of the major changes that occurred between the old and the newer updated version is that the two highest forms of cognition have been reversed. In the older version the listing from simple to most complex functions was ordered as knowledge, comprehension, application, analysis, synthesis, and evaluation. In the newer version the steps change to verbs and are arranged as knowing, understanding, applying, analyzing, evaluating, and the last and highest function, creating.



**Figure 2. 2: Revised Taxonomy (2000-2001), Anderson and Krathwohl primary level in cognition domain.**

Table (2.1) presents Bloom’s taxonomy levels (1956) and Anderson/Krathwohl levels (2001) with definitions and sample verbs:

**Table 2. 1 Bloom’s taxonomy vs Anderson/Krathwohl levels, definitions and verbs**

Bloom’s taxonomy Levels – 1956		
Level	Definition	Sample Verbs
Knowledge	This level includes behaviors which emphasize remembering either by recognition or recall of ideas, material or phenomena.	Define, write, name, and list.
Comprehension	This level includes the ability to translate, comprehend or interpret information	Summarize, describe and explain
Application	“To apply something requires "Comprehension" of the method, theory, principle, or abstraction applied.” (1956)[21]	Compute, solve and apply
Analysis	“Analysis emphasizes the breakdown of the material into its constituent parts and detection of the relationships of the parts and of the way they are organized.”(1956)[21]	Analyze, compare
Synthesis	This level involves a “recombination of parts of previous experiences with new materials, reconstructed into a new and more or less well-integrated whole”.(1956)[21]	Design, create and develop
Evaluation	“Evaluation is defined as the making of judgments about the value -for some purpose- of ideas, works, solutions, methods, material, etc.” It involves using criteria and standards. The judgments may be either quantitative or qualitative. .(1956)[21]	Recommend, Judge
Anderson/Krathwohl levels		
Level	Definition	Sample Verbs
Remembering	Recognizing or recalling knowledge from memory.	Recognizing, Recalling
Understanding	Constructing meaning from different types of functions including oral, written, and graphic communication.	Interpreting Comparing
Applying	Executing or implementing procedures in a given situation.	Executing Implementing
Analyze	Breaking material into its constituent parts and detecting how the parts relate to one another and to an overall structure or purpose. [49]	Organizing
Evaluating	“Making judgments based on criteria and standards”. [49]	Checking
Creating	“Putting elements together to form a novel, coherent whole or make an original product” [49]	Generating Producing

### **Example of Learning Outcomes for “Algorithms and Data Structure” course:**

By completion of this course, student should be able to:

1. Define basic static and dynamic data structures and relevant standard algorithms for them: stack, queue, dynamically linked lists, trees, graphs, heap, priority queue, hash tables, sorting algorithms and min-max algorithm.
2. Demonstrate advantages and disadvantages of specific algorithms and data structures.
3. Select basic data structures and algorithms for autonomous realization of simple programs or program parts.
4. Determine and demonstrate bugs in programs and recognize needed basic operations with data structures.
5. Formulate new solutions for programming problems or improve existing code using learned algorithms and data structures.
6. Evaluate algorithms and data structures in terms of time and memory complexity of basic operations.

[Learning outcome for “Data Structures and Algorithms” course designed by lecturer Toma Rončević at university of split

[https://www.oss.unist.hr/sites/default/files/dokumenti/courses/information\\_technology/enDIP\\_SIT019\\_Data\\_Structures\\_and\\_Algorithms.pdf](https://www.oss.unist.hr/sites/default/files/dokumenti/courses/information_technology/enDIP_SIT019_Data_Structures_and_Algorithms.pdf) / page 3

### **2.2.2 Bologna Declaration**

The Bologna Declaration was adopted by ministers of education of 29 European countries in Bologna, Italy to formulate the Bologna agreement leading to the setting up of a common European Higher Education Area (EHEA).

The overall aim of the Bologna process is to improve the efficiency and effectiveness of higher education in Europe as well as promote student and staff mobility throughout the EHEA and beyond, which can guarantee the freely movement of students and graduates between countries by using a supplement which describes the qualification the student has received in a standard format that is easy to understand and compare [16] [14].

Six main objectives were defined by Bologna declaration:

1. “Adoption of a system of easily readable and comparable degrees” which means the using of learning outcomes as a common language which is clear for all institutes, employers and evaluating qualification. [16]
2. Adopt a system with two main cycles (undergraduate/graduate)
3. Establish a system of credits “European Credit Transfer System” (ECTS)

This depends mainly on evaluating the learning outcomes as the user guide declared: “Credits in ECTS can only be obtained after successful completion of the work required and appropriate assessment of the learning outcomes achieved” [48]

4. Promote mobility by overcoming legal recognition and administrative obstacles
5. Promote European cooperation in quality assurance
6. Promote a European dimension in higher education.

## **2.3 E-Learning and adaptive e-Learning systems**

What does e-learning mean? How does it differ from the traditional classroom-based learning? and what benefits it comes with? e-learning is the learning process using electronic technologies or devices (computers, tablets or phones) to access online educational curriculum (course, program or degree) outside of a traditional classroom-based learning. E-learning courses can use variety of techniques such as video, presentation, quizzes, games... etc. [25]

The main benefit of e-learning is that learning becomes accessible for all users around the world as they can select their courses and start to learn at any time during the day with no time restrictions.

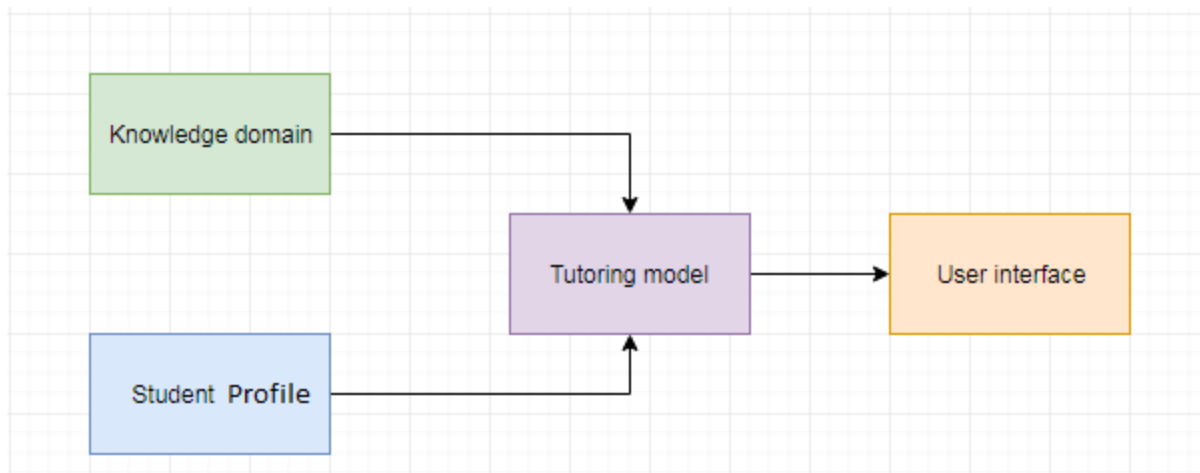
In an adaptive e-learning environment, the learning system respond differently based on learner’s needs, style and context. This type of learning is based on the principle that each student is unique and have different background, knowledge level, learning needs, misunderstand and learning outcomes than others.

The architecture of an adaptive e-learning system consists of four main blocks as follow:



1. Knowledge domain: presents the set of knowledge which will be learned to students
2. Student model: presents the student profile which contains information about the student's learning outcomes, knowledge level, preference, learning styles...etc.
3. Tutoring model: presents the intelligence which matches between the students' needs based on their unique background and the appropriate content in the knowledge domain to minimize the gap between the student and the knowledge and making the process of learning easier.
4. User interface: presents the interaction gate between students and the system. [24]

The following diagram shows an adaptive e-learning architecture with its five blocks.



**Figure 2. 3: Adaptive e-learning Architecture [24]**

## 2.4 Intelligent Tutoring System

“Intelligent Tutoring Systems (ITS) are computer programs that model learners' psychological states to provide individualized instructions”. These customized instructions are given without any intervention from teachers. [27]

The first ITS program was SCHOLAR, which was designed by the computer scientist Jaime Carbonell in 1969, SCHOLAR was a man-to-machine tutorial system which uses templates and keyword recognition. Its job is to teach students about Latin American geography through inquiries and answers on random topics selected by students.

Another early example of ITS is BIP (1976). BIP was a basic instructional program and interactive problem-solving laboratory. Its main job is to assign programming tasks to students based on student learning needs and competencies.

Wenting Ma, et al. (2014) proposed in their paper, the main tasks of an ITS:

1. ITS is a computer system that performs tutoring functions by answering questions, assigning tasks and offering feedback.
2. Compute student inference and based on that either construct a new multidimensional model for students or allocate them within one of the existing models.
3. Use the student model function to adapt the appropriate tutoring functions.

### **Eight principles of ITS design and development**

Anderson et al. (1987) identified a set of eight principle for designing intelligent computer tutors which can be consulted for successful application of such rules.

1. Identify the goal structure of problem space.
2. Provide instructions in the problem-solving context.
3. Provide immediate feedback on errors.
4. Minimize working memory load.
5. Represent student competence as a production set.
6. Adjust the grain size of instruction according to learning principles.
7. Enable the student to approach the target skill by successive approximation.
8. Promote the use of general problem-solving rules over analogy.

## Chapter 3

# **Knowledge Discovery**

## **Introduction**

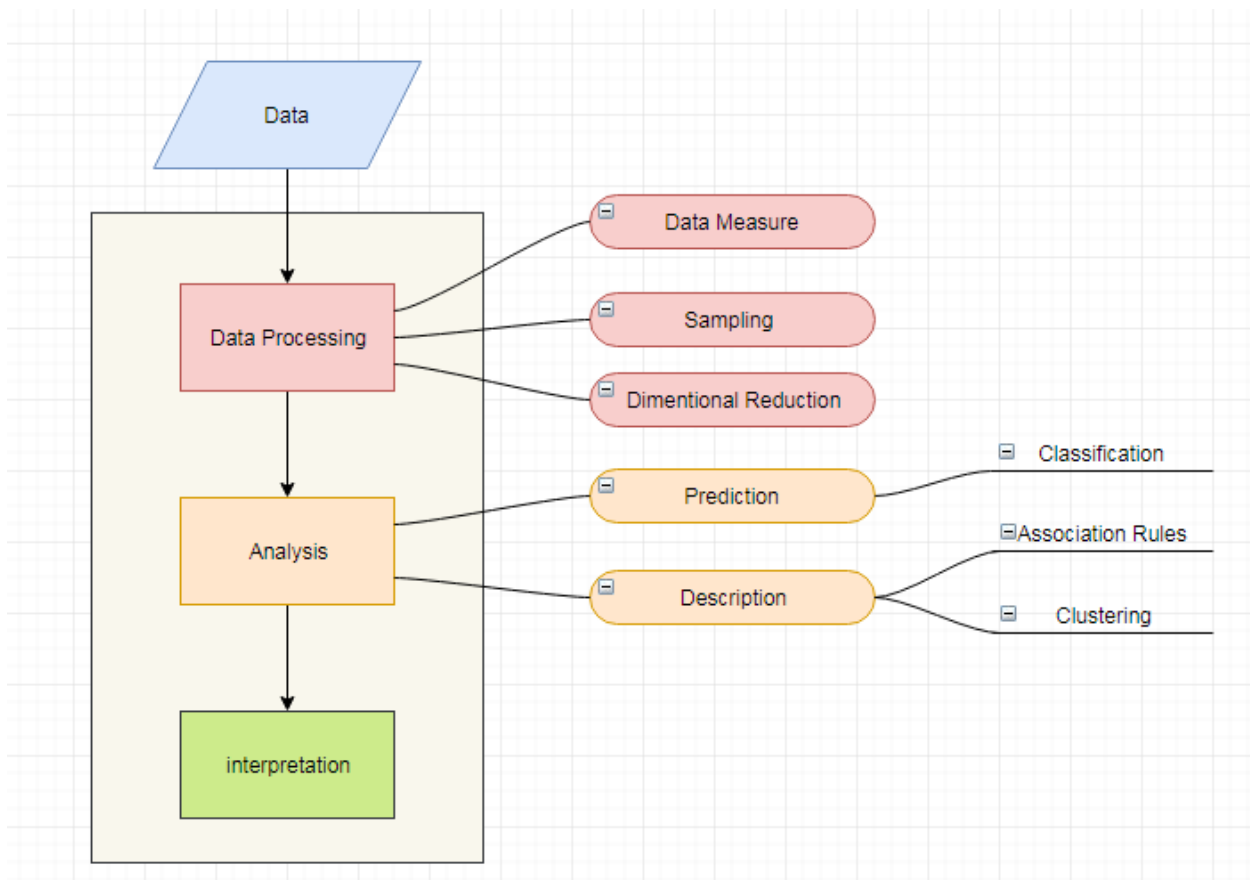
Knowledge discovery in database “KDD” refers to the process of discovering useful knowledge from data using the high-level application of data mining. Data mining is considered as the core part of knowledge discovery where mathematical analysis is used to drive patterns and trends that exist in data [47].

This chapter introduces a sequential steps and procedures of extracting Knowledge form large sets of structured data, concentrating on those steps and procedures which are used in the methodology in “Chapter Five”. The chapter explain how data mining is used in recommender systems and illustrates different types of sampling followed by choosing one of those sampling methods to be used in the study. It also introduces some data distribution models and show how the e-Learning environment matchs the normal distribution model. Finally, the chapter concentrate on clustering analysis process and five main distance functions as they are used in the methodology, highlighting important methods of finding the best number of clusters and measuring the quality of the resulting clusters.

### **3.1 Data Mining in Recommender Systems**

The process of data mining typically consists of three main steps: data preprocessing, data analysis and result interpretation. The data mining methods that are most commonly used in recommender systems are: classification, clustering and association rule discovery.

Figure (3.1) summarizes the main steps in a data mining problem which starts with data processing, analysis and interpretation for results. Data processing include data measures, sampling and dimensional reduction. Analysis could include prediction for future results, making use of current history, supervision data or descriptive and looking for patterns in unsupervised large data set.



**Figure 3. 1: Main steps and methods of a data mining problem**

## Data Processing

Data processing is the stage of cleaning, filtering and transforming data to be prepared for the next step of analysis.

### 3.2 Sampling

Sampling is one of the main techniques in data mining, it is a statistical analysis technique used to select a subset of relevant data from a large dataset in order to identify patterns and trends in the larger dataset. This section discusses some sampling methods in order to choose the appropriate sampling method for the learning environment.

Also, sampling can be done by taking a training dataset which is used for learning and building the analysis model and a testing set to evaluate the created model and its accuracy.

An important consideration is the size of the sample set. Sometimes small datasets can tell all about the data, in other cases, increasing the size of dataset can increase the accuracy of the analysis [7]. Sampling method can be classified into two categories: probability sampling and non-probability sampling. In probability sampling there are many methods such as:

1. Simple Random Sampling (SRS)
2. Stratified Sampling
3. Cluster Sampling
4. Systematic Sampling
5. Multistage Sampling

SRS is a statistical model for the selection of a sample contains an  $n$  number of sampling units out of the population which have  $N$  number of sampling units. In this sampling method, every possible sample of the same size is equally likely to be chosen.

A stratified random sample is obtained by separating the population into mutually exclusive sets where every element in the population is assigned to only one set, or strata, where no elements could be excluded and then drawing simple random samples from each stratum, where samples is taken from all stratum.

Cluster sampling is often used in marketing research in which the total population is divided into groups known as clusters and a simple random sampling is applied on each cluster. The main aim of this type of sampling is to reduce cost and increase efficiency.

Systematic sampling is a statistical model in which the first step is to determine the number of samples “ $n$ ” to be chosen from the whole population “ $N$ ”, then every  $k^{th}$  element is selected from the ordered sampling frame where  $k = N/n$ .

Multistage Sampling is a complex form of cluster sampling, this type of sampling involves dividing the whole population into clusters and then choosing one or more clusters randomly where each selected cluster is then sampled. For example, dividing a study area into districts followed by choosing random districts, then dividing each district into blocks followed by

choosing random number of blocks and finally choosing random samples from the selected blocks.

This research is dealing with students and students' achievements in different learning outcome levels, it is trying to divide students into groups where each student belongs to one group, the members of each group are similar to each other. The target is to find out those students who are similar to the active student and then study their behavior, learning patterns and feedback on learning materials in order to reflect their success learning experience on active students.

As a conclusion, the clusters which includes the active students, are the only clusters which are taken from the whole sets of data as it contains the similar students, based on this the cluster sampling approach is chosen.

### **3.3 Data Distribution Models**

“Things are random” this is a fact about our world. “A random variable is a numerical description of the outcome of an experiment whose value depends on chance” [30]. To design and analyze any experiment, data collection about the phenomena is needed. Good data collection practice involves randomly selecting individuals from the population, or randomly assigning treatments in a controlled experiment.

Probability theory is essential in analyzing human activities which involves quantitative analysis data. It explains how to compute the chance that events will occur based on assumptions about things like the probabilities of the elementary outcomes in the sample space [31]. This section describes briefly some data distribution models for by which to use the appropriate data model for this research which will be discussed in later chapters:

1. Random normal distribution.
2. Poisson distribution.
3. Binomial distribution.
4. Discrete uniformed distribution.

### Random Normal Distribution:

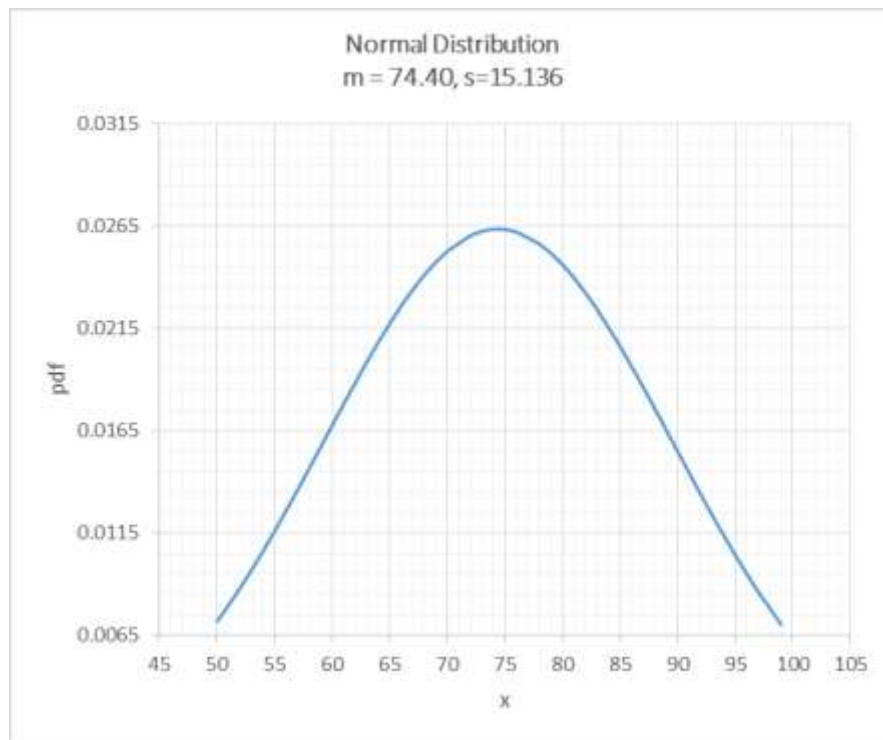
The normal or Gaussian distribution, or the “bell curve”, is based on the assumption that a distribution of values generally clusters around an average. Within the distribution, very high and very low values are still possible, but are less frequent than the ones closer to the average.

The probability density function of the normal distribution is:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\mu\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Where  $\mu$  is the mean distribution,  $\sigma$  is the standard deviation  $\sigma^2$  is the variance.

Figure (3.2) shows the normal distribution of student's marks with an average of 74.40 and a standard deviation of 15.136:



**Figure 3. 2: Normal distribution for student marks based on random generated data set**



## Poisson distribution

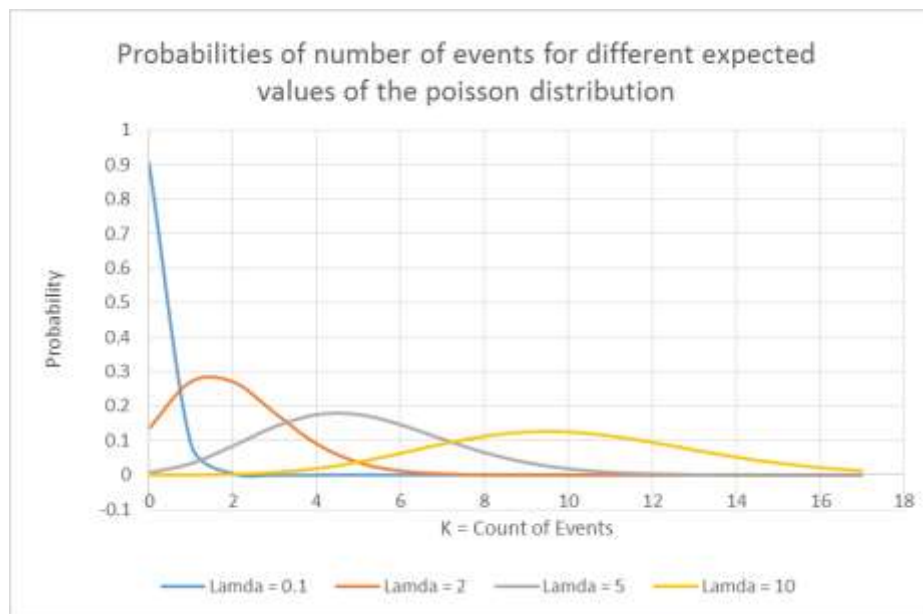
Poisson distribution describes the probability of a given number of events “K” occurring in a fixed interval (e.g. time, distance, area or volume). It is an appropriate model when the number of times an event occurs “k” take values: 0, 1, 2, .... and the occurrence of one event does not affect the probability that a second event will occur, where exactly just one event could happen in an instance of time.

The probability of observing k events in an interval is given by the following equation:

$$P(K \text{ event in interval of time}) = e^{-\lambda} \frac{\lambda^k}{k!}$$

Where  $\lambda$ : average number of events per interval,  $k$ : the number of times an event occurs in an interval.

Figure (3.3) shows Poisson distribution for a set of events with different values for  $\lambda$



**Figure 3. 3:Poisson distribution**

## Binomial Distribution

The binomial distribution is used to model a certain number of successes “r” in an “N” independent trials drawn with replacement from a population of size N’.

The probability of one possible way the event can occur is calculated by the equation below:

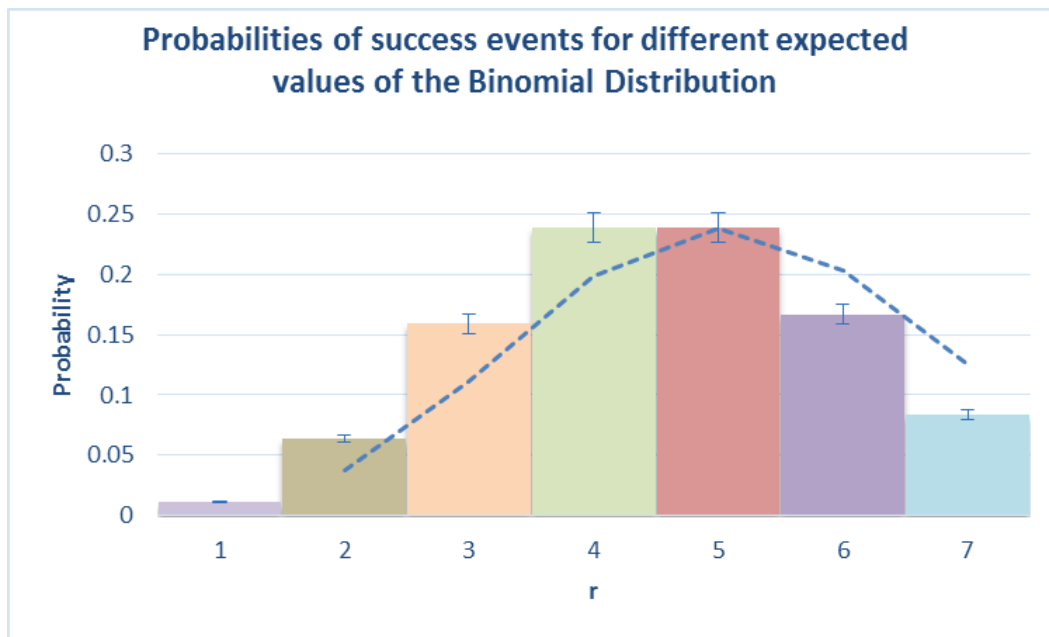
$P(\text{Event}) = (\text{Number of ways event can occur}) * P(\text{One occurrence})$ .

The total number of ways of selecting r distinct combinations of N objects, irrespective of order, is:

$$\binom{N}{r} = \binom{N}{N-r} = \frac{N!}{r!(N-r)!}$$

The probability of getting exactly r successes in N trials is given by the probability mass function:

$$\Pr(r;n,p) = \binom{N}{r} p^r (1-p)^{N-r}$$



**Figure 3. 4:Binomial distribution**

## Discrete Uniformed distribution

Discrete uniform distribution explains finite number of outcomes which are equally likely to happen, it gives its values the same probability to occur. Mathematically this means that the probability density function is identical for a finite set of evenly spaced points.

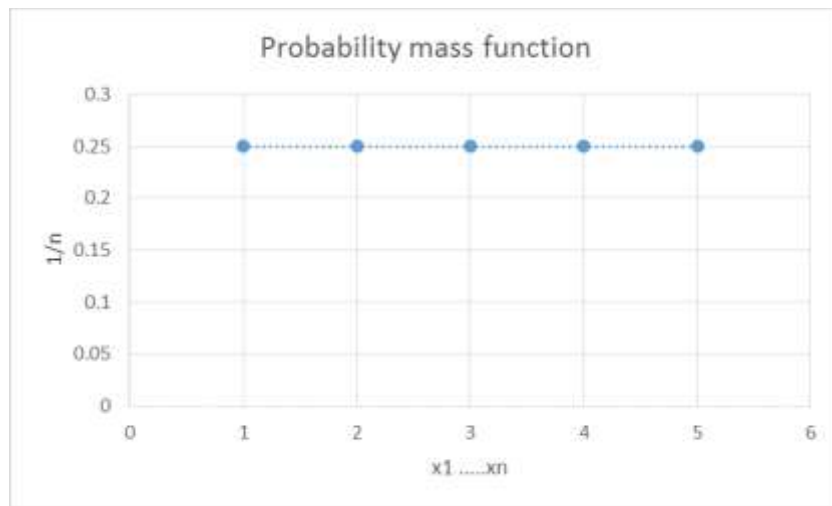
If there exists  $n$  events, each of which have the same probability  $P(X=x)=1/n$ ; the random variable  $X$  follows a discrete uniform distribution and its probability function is:

$$P(X=x) = \begin{cases} 1/n & \text{if } x = x_1, x_2, \dots, x_n \\ 0 & \text{if not} \end{cases}$$

The cumulative distribution function (CDF) of the discrete uniform distribution can be expressed, for any  $k \in [a,b]$ , as

$$F(k; a,b) = \frac{\lfloor k \rfloor - a + 1}{b - a + 1}$$

Figure (3.5) shows the probability mass function.



**Figure 3. 5:Probability mass function**

## **Conclusion:**

The data distribution model used in this research in order to generate student achievements on the level of learning outcome is the random normal distribution which mimics student's results.

### **3.4 Distance and Similarity Measures**

In this section, a brief overview on various distance similarity measures is discussed as they are used in the research methodology, mainly when applying the k-means on student's achievements in order to find out similar students. So the first question is: What is similarity and how can it be measured?

Similarity is the measurement that quantifies the dependency between two sequences X and Y where  $X = \{x_1, x_2, x_3, \dots, x_n\}$  and  $Y = \{y_1, y_2, y_3, \dots, y_n\}$ , both X and Y are measurements from two objects or phenomena. [34]

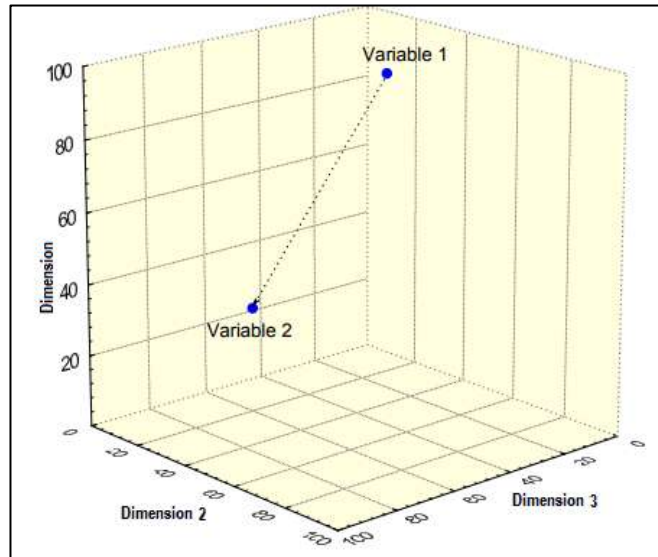
A distance function is “a function defined over pairs of data points. The function produces a real (and possibly bounded) value, which measures the distance between the pair of points.”[35]

Distance and similarity measures are very essential in knowledge discovery and recognizing different patterns in data such as in clustering and classification.

#### **3.4.1 Euclidean distance**

The Euclidean distance or metric is defined to be the straight-line distance between two points in the Euclidean space. In general, for an n-dimensional space, the distance is

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$



**Figure 3. 6: The Euclidean Distance between 2 variables in 3-dimensional space**

A norm is a function which assign a positive value to vector in the vector space which measures the distance between this it and the zero vector.

An Euclidean norm is the length of the vector X in the n dimensional Euclidean space and measured by:

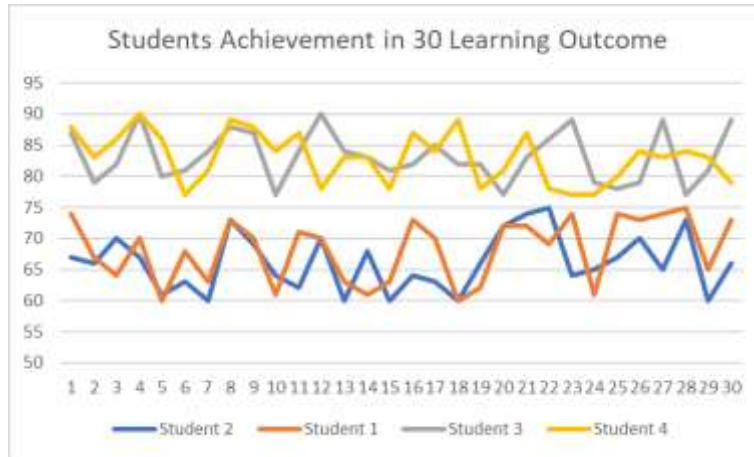
$$\|X\| = \sqrt{\sum_{i=1}^n (x_i)^2}$$

### 3.4.2 Correlation Distance (or Pearson correlation distance)

Correlation coefficients are used to measure the strength of relationship in statistics, it measures how strong a relationship between two variables is. Pearson's correlation is a linear correlation coefficient that returns a value between -1 and 1 where +1 means there are a strong positive relationship between two items and -1 means there exists a strong negative relationship and zero denotes that there is no relationship. The following formula calculates the correlation coefficient between two vectors X and Y.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}}$$

As an example, figure (3.7) shows four students' achievements in 30 learning outcomes:



**Figure 3. 7: Student's achievement in 30 learning outcome**

From the figure above, it is obvious that student1 and student2 are more similar in their achievements when compared with student3 and student4. when correlation coefficient analysis was applied using excel data analysis, the following results appear as in figure (3.8).

student1	student2	student3	student4	
			1	student4
		1	0.31289	student3
	1	0.054541	-0.05885	student2
1	-0.47234	0.039279	-0.02925	student1

**Figure 3. 8: Correlation coefficients between students using data analysis in excel**

The figure above shows that there exists a negative correlation between student 1 and 2 with a correlation of 0.472 and positive correlation of 0.312 between student 3 and 4.

when applying the distance correlation analysis using R the following results appear:

```
> dist(students,students, method = "correlation")
      [,1]      [,2]      [,3]      [,4]
[1,] 0.0000000 0.4992678 0.6754280 0.8932932
[2,] 0.4992678 0.0000000 0.9286079 0.9220904
[3,] 0.6754280 0.9286079 0.0000000 0.8909627
[4,] 0.8932932 0.9220904 0.8909627 0.0000000
```

**Figure 3. 9:Correlation distance between students using R where correlation distance is compliment for Correlation coefficient**

Figure (3.9) shows that “student 1” is more similar to “student 2” where the correlation distance between them is 0.499 which is the minimum among other students and so it is the shortest distance, in the same time the minimum distance is found between “students 4” and “student 3” and equals to 0.8909 which also denotes that they are both more similar.

### 3.4.3 Jaccard Similarity Coefficient (Index)

Jaccard coefficient (Index) measures the number of shared members in two vectors and gives a result that has a range between 0 and 100% for which the higher the percentage is, the more the items are similar. Jaccard distance is the complement of Jaccard index where it measures the value of dissimilarity between two vectors.

Equation calculates jaccard Index between two vectors X and Y.

$$J(X,Y)=|X\cap Y|/|X\cup Y|$$

The following example shows the analysis for the same four students’ marks in the previous section using Jaccard distance. Student’s achievement in each learning outcome is classified as A,B,C,D to make the probability of intersection between marks higher and so giving more accurate similarity calculation.

The example uses the proxy library in R to summarize student’s achievement and find Jaccard distance.

```
> summary(mydata)
Student.1 Student.2 Student.3 Student.4
B:13      B:13      A:16      B: 8
C:17      D:17      B:14      C:22
~ |
```

**Figure 3. 10: Learning outcome achievement summary per student**

```

> dist(mydata, mydata, method = "eJaccard")
      [,1]      [,2]      [,3]      [,4]
[1,] 0.0000000 0.1011905 0.1650485 0.0726257
[2,] 0.1011905 0.0000000 0.3773585 0.2289157
[3,] 0.1650485 0.3773585 0.0000000 0.1993355
[4,] 0.0726257 0.2289157 0.1993355 0.0000000

```

**Figure 3. 11: Similarity between students based on Jaccard distance**

Figure 3.10 shows student achievement summary, counting student achievement in each level, where A indicates that the mark is above 90, B indicates that the mark is between 80 and 90, C indicates that the mark is between 70 and 80 and finally D indicates that the mark is between 60 and 70.

Figure 3.11 calculate the Jaccard distance between students based on the mentioned levels where higher opportunity for intersection is found.

Jaccard distance is recalculated on the same set of students but this time on their achievement as a numerical number from 50 to 100.

```

> dist(students,students, method = "eJaccard")
      [,1]      [,2]      [,3]      [,4]
[1,] 0.000000000 0.005916986 0.042737878 0.043750809
[2,] 0.005916986 0.000000000 0.055548561 0.055009688
[3,] 0.042737878 0.055548561 0.000000000 0.004136142
[4,] 0.043750809 0.055009688 0.004136142 0.000000000

```

**Figure 3. 12: Similarity based on Jaccard distance**

Figure 3.12 also shows that the distance between student1 and student2 is the minimum which indicate that they are more similar to each other's, as well as student3 and student4.

### 3.4.4 Cosine Similarity

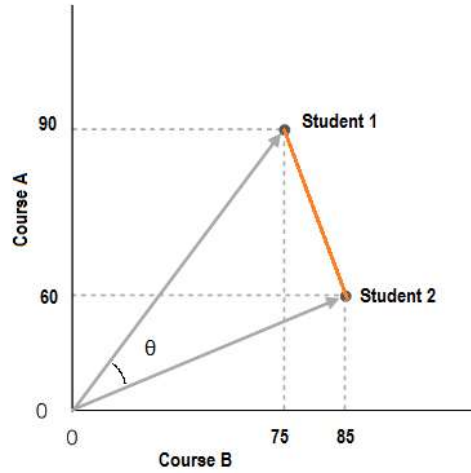
Cosine similarity is widely used in data mining, recommendation systems and information retrieval. The cosine of 0 is one, so whenever an angle between two vectors is zero this means they are the same, the smaller the angle between two vectors the more similar they are. Cosine similarity between two non-zero vectors X and Y is represented as:

$$\text{Similarity} = \cos(\theta) = \frac{X.Y}{\|A\|_2 \|B\|_2} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$



The red line in figure (3.13) represents Euclidean distance between the two vectors which represent two students student 1 and student 2, the distance d is equals to

$$d = \sqrt{(x_i - y_i)^2 + (x_i - y_i)^2}$$



**Figure 3. 13:Two-dimensional illustration of Euclidean distances between two points**

Figure (3.14) shows the cosine similarity between the same set of students, the analysis also shows that student 1 is more similar to student 2, whereas Student 3 is more similar to student 4, where the distance between them were shown to be the shortest.

```
> dist(students,students, method = "cosine")
      [,1]      [,2]      [,3]      [,4]
[1,] 0.000000e+00  2.496009e-03  2.729234e-03  3.502087e-03
[2,] 2.496009e-03 -2.220446e-16  3.191933e-03  3.168365e-03
[3,] 2.729234e-03  3.191933e-03  1.110223e-16  2.071631e-03
[4,] 3.502087e-03  3.168365e-03  2.071631e-03 -2.220446e-16
```

**Figure 3. 14: Cosine similarity between a set of four students**

### 3.4.5 Manhattan (or City-Block) Distance

Manhattan distance measures the shortest distance between two points  $x_i$  and  $x_j$  that one would be required to walk if a city is laid out in square blocks ("city blocks").

This distance is defined by:

$$D_{Manhattan} = \sum_{k=1}^d |x_{ik} - x_{jk}|$$

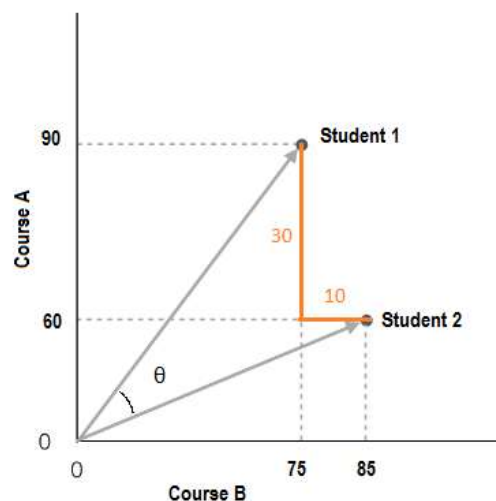
The matrix bellow shows the Manhattan distance between the four students' achievements where the shortest distance is found between student1 and student2 which indicates that they are more similar when compared with the two others, and between student4 and student3 which also indicates that they are more similar.

```
> dist(students,students, method <- "manhattan")
      [,1] [,2] [,3] [,4]
[1,]    0  127  450  447
[2,]  127    0  511  508
[3,]  450  511    0  129
[4,]  447  508  129    0
```

**Figure 3. 15: Manhattan distance between four students using R**

From the matrix above, it's obvious that student 1 &2 is more similar to each other when comparing them with the two other students as Manhattan distance between them is the shortest with a distance of 132. For the same reason, student 3 is more similar to student 4 where the distance is 165.

Figure (3.16) illustrates the city-block distances between two dimensional points where the Manhattan distance is measured as  $|90-60|+|85-75| = 40$ .



**Figure 3. 16:Two-dimensional illustration of city-block distances between two points**

### 3.5 Analysis Process

### 3.5.1 Cluster Analysis

Clustering analysis is an unsupervised learning method, in which each item (vector) is assigned to a group. Items in the same group are more similar than items in other groups. Similarity between items is measured using distance, where the goal of clustering algorithm is to minimize the distance in each cluster and to increase the distance between clusters. Clustering analysis is best fit when speaking about big dimensionality of features. [7]

Clusters could be distinguished by their various type: (1) they could be partitioned or hierarchized (nested), (2) made exclusive where each item is assigned to a single cluster or fuzzy where each item belong to a cluster with a weighted membership. (3) Partial or complete, where in complete clusters, each item in the population must belong to a cluster. [32]

### 3.5.2 K-means Algorithm

“The aim of the K-means algorithm is to divide M points in N dimensions into K clusters so that the within-cluster sum of squares is minimized.” [36]

K-means clustering is a type of unsupervised learning where the methods goal is to group observations into a specific number of disjoint clusters “k”, where “k” refers to the number of clusters specified. The results of applying K-means clustering on a data set is: (1) a set of cluster centroids where each centroid is a collection of features that defines the cluster. (2) Each observation in the dataset is labeled to a cluster.

There are various distance measures used to determine to which cluster each observation will be appended, where the cluster algorithm aims to minimize the distance between the centroid and the observation.

Figure (3.17) shows the basic k-means algorithm steps, which starts with initial estimates for the K centroids either by random selection from the data or randomly generated. The algorithm then iterates between two steps shown in lines three and four in which: (1) each observation is assigned to the cluster of the nearest centroid (2) centroids are recomputed by taking the mean for all points in the same cluster. Finally, the algorithm exits when centroids don't change.

---

**Algorithm 1** Basic K-means Algorithm.

---

- 1: Select  $K$  points as the initial centroids.
  - 2: **repeat**
  - 3:   Form  $K$  clusters by assigning all points to the closest centroid.
  - 4:   Recompute the centroid of each cluster.
  - 5: **until** The centroids don't change
- 

**Figure 3. 17: Basic k-means algorithm steps**

The most commonly used implementation of k-means clustering is the one which tries to find the partition of the  $n$  individuals into  $k$  groups that minimizes the within-group sum of squares (WGSS) over all variables, it is computed as:

$$\text{WGSS} = \sum_{j=1}^q \sum_{l=1}^k \sum_{i \in G_l} (x_{ij} - \overline{x_j^{(l)}})^2$$

*“Where  $G_l$  denotes the set of  $n_l$  individuals in the  $l$ th group and where  $\overline{x_j^{(l)}} = \frac{1}{n_l} \sum_{i \in G_l} x_{ij}$  is the mean of the individuals in group  $G_l$  on variable  $j$ ” [37]*

Figure 3.18 represents a matrix of two hundred students' marks achieved in 30 learning outcomes ( $v_1..v_{30}$ ), each row in the matrix represents one student, each column represents a learning outcome and each entry represents the student's mark in the learning outcome.

```

> mydata
  V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21 V22 V23 V24 V25 V26 V27 V28 V29 V30
1  72 80 75 80 78 73 71 80 75 80 76 71 70 79 79 70 79 75 77 79 72 78 71 80 70 72 80 74 77 75
2  72 80 75 80 78 73 71 80 75 80 76 71 70 79 79 70 79 75 77 79 72 78 71 80 70 72 80 74 77 75
3  80 71 72 72 75 75 73 72 73 81 75 76 71 75 80 76 82 75 83 73 76 73 77 74 80 73 78 83 71 71
4  93 81 92 80 88 95 92 85 85 93 86 95 82 94 82 93 90 82 92 92 80 80 83 85 84 83 91 86 86 88
5  90 93 88 89 92 86 87 94 87 85 86 85 88 92 88 93 88 95 91 85 92 95 87 88 88 86 94 83 85 89
6  86 97 90 94 90 86 98 91 87 88 97 91 92 89 98 97 96 85 98 94 99 89 86 86 88 95 96 98 94 99
7  87 93 87 88 85 90 90 91 95 94 91 89 94 92 85 86 87 93 85 85 95 85 86 91 95 92 95 88 88 85
8  89 89 85 94 99 91 97 90 95 95 94 91 94 85 98 96 88 92 93 97 88 86 90 94 98 90 93 85 87 94
9  99 93 86 87 89 93 97 90 90 88 87 85 99 90 92 97 91 92 98 89 94 93 95 95 92 98 96 99 99 87
10 86 95 93 86 95 91 87 86 93 93 95 99 95 85 88 90 90 87 91 91 89 85 95 86 88 89 88 94 91 93
11 76 78 79 66 76 72 67 74 76 76 79 72 65 77 71 79 67 79 67 77 74 76 76 66 70 74 71 67 65 65
12 79 75 74 70 69 68 68 72 68 73 77 71 77 79 70 66 71 71 71 72 68 66 77 76 79 69 70 78 79 77
13 73 66 78 67 79 67 69 65 79 77 73 79 79 78 68 77 75 74 75 71 69 69 72 76 79 74 77 72 79 73
14 89 88 92 89 99 95 87 98 86 93 98 99 95 97 88 90 86 89 99 98 97 96 92 90 90 91 94 93 92 87
15 75 74 75 76 79 76 70 71 66 76 72 66 75 75 66 77 74 78 69 70 75 69 74 74 67 67 67 69 74 73
16 77 77 79 74 72 72 67 77 73 66 65 77 67 78 76 74 78 66 73 77 75 65 70 72 74 79 66 78 75 67
17 68 75 78 65 77 69 75 69 69 76 67 73 72 72 69 73 77 75 66 72 71 66 73 79 71 72 77 66 71 66
18 79 71 68 66 72 78 70 71 65 76 66 67 72 79 76 73 79 68 75 69 70 76 71 78 72 67 75 70 68 75
19 78 70 65 77 72 74 72 73 77 74 72 67 74 70 76 70 73 70 75 72 65 67 68 67 70 78 76 70 74 67
20 73 79 66 66 75 70 74 66 69 76 77 68 78 77 73 75 69 68 70 71 72 77 79 75 73 78 65 73 72 69
21 68 66 70 75 70 77 75 76 70 75 71 70 69 67 74 78 66 66 67 76 73 72 70 65 74 79 69 65 79 79
22 65 73 77 66 75 67 70 73 76 78 76 72 75 78 69 72 76 71 79 77 71 71 67 74 79 78 68 67 76 69
23 70 69 71 68 75 68 65 77 70 73 75 72 65 77 73 68 67 79 66 70 74 77 74 77 70 66 68 65 68 65
24 79 69 71 73 69 71 67 72 70 75 79 67 70 77 73 65 70 76 73 71 66 74 74 71 74 74 76 76 66 75
25 67 71 68 72 72 71 69 67 77 65 68 79 79 66 76 68 75 79 78 66 71 79 76 76 73 65 76 65 71 66
26 70 66 75 78 83 85 86 80 80 76 78 70 85 81 79 79 79 79 79 79 79 79 79 79 79 79 79 79 79 79

```

**Figure 3. 18: 200 hundred student marks in 30 learning outcomes represented in R**

Figure (3.19) represents the scatterplot matrix for student marks in the first five learning outcomes. Scatterplot matrix contains all the pairwise scatter plots of first five learning outcomes, the plot contains 5 X 5 cells in which each cell represents the plot of  $X_i$  “learning outcome” versus  $X_j$  “learning outcome”.

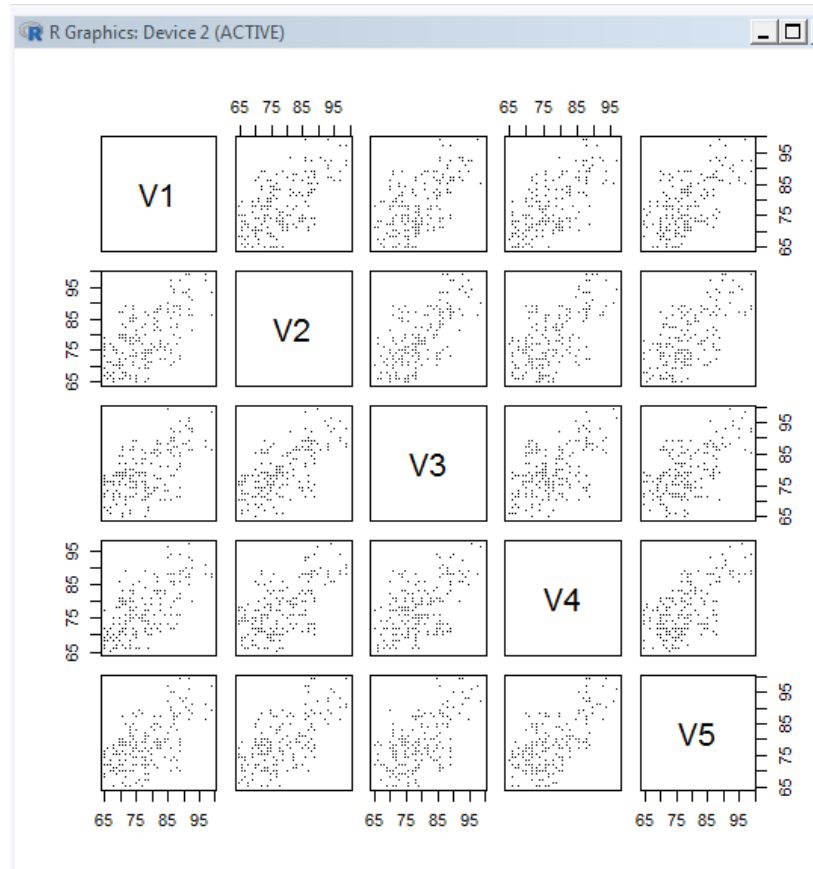


Figure 3. 19: Scatterplot matrix of the first five learning outcomes for 200 students

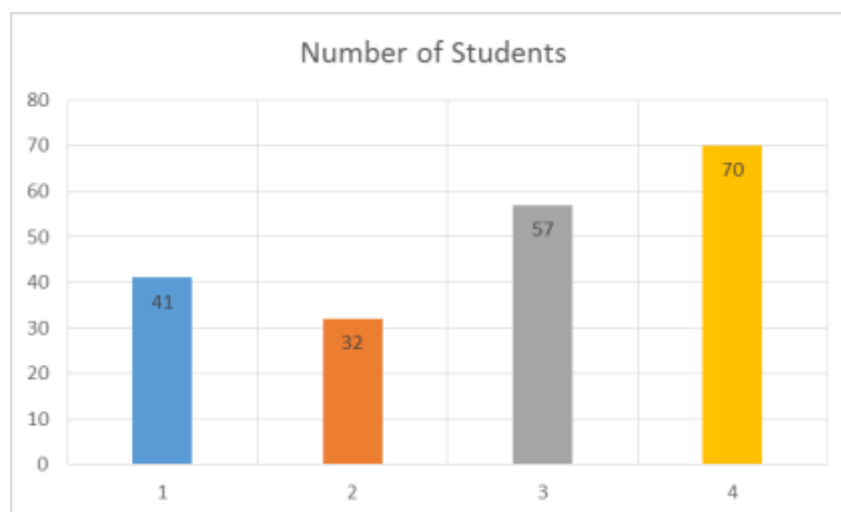
Figure 3.30 shows the results of k-means clustering process when applied on the previous set of 200 students for the first 10 learning outcome, K was set to four centroid points, for where the first cluster centroid point is around learning outcomes achievement (74,78,79,77,77,77,77,80,74,79) ordered from the first learning outcome to the 10th one. Whereas the second centroid is around (90,90,90,88,91,91,91,92,90,89), the third centroid is (81,80,79,79,80,81,79,78,82,79) and finally the last centroid is around (72,71,72,71,72,72,70,72,72,73).

```
> kmeans(sdata, centers = 4)
K-means clustering with 4 clusters of sizes 41, 32, 57, 70

Cluster means:
      V1      V2      V3      V4      V5      V6      V7      V8      V9      V10
1 74.17073 78.80488 79.65854 77.63415 77.60976 77.39024 77.26829 80.68293 74.97561 79.85366
2 90.84375 90.81250 90.53125 88.43750 91.43750 91.59375 91.00000 92.03125 90.90625 89.93750
3 81.59649 80.54386 79.98246 79.35088 80.29825 81.68421 79.54386 78.94737 82.64912 79.61404
4 72.25714 71.24286 72.95714 71.78571 72.30000 72.45714 70.97143 72.54286 72.67143 73.25714
```

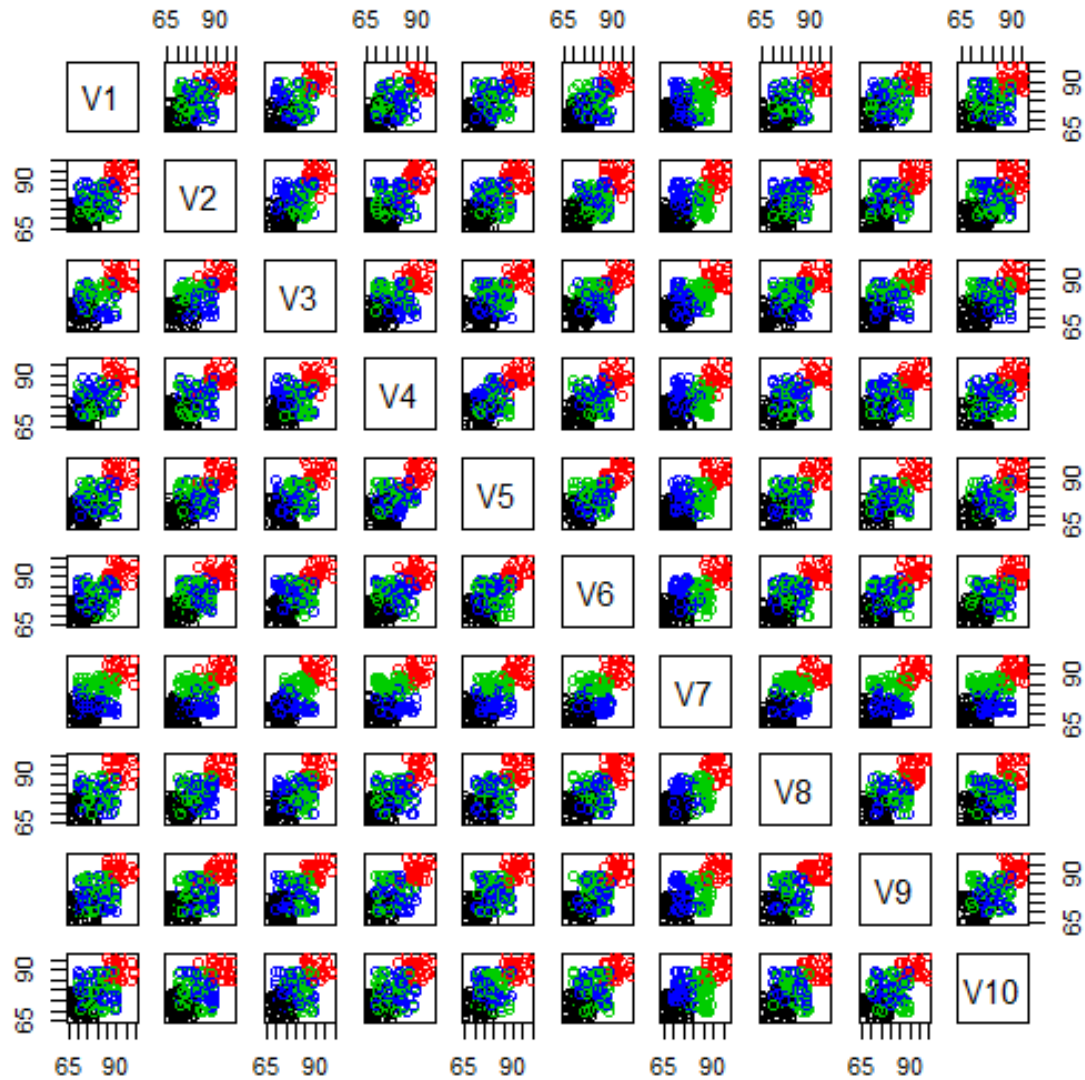
**Figure 3. 20: k-mean cluster for the first ten learning outcome of 200 student using R**

The cluster method - for the first ten learning outcomes - results in 41 students in the first cluster, 32 students in the second cluster, 57 students in the third cluster and 70 students in the last cluster. The following graph shows the distribution of students on clusters.



**Figure 3. 21: Number of Students in each cluster**

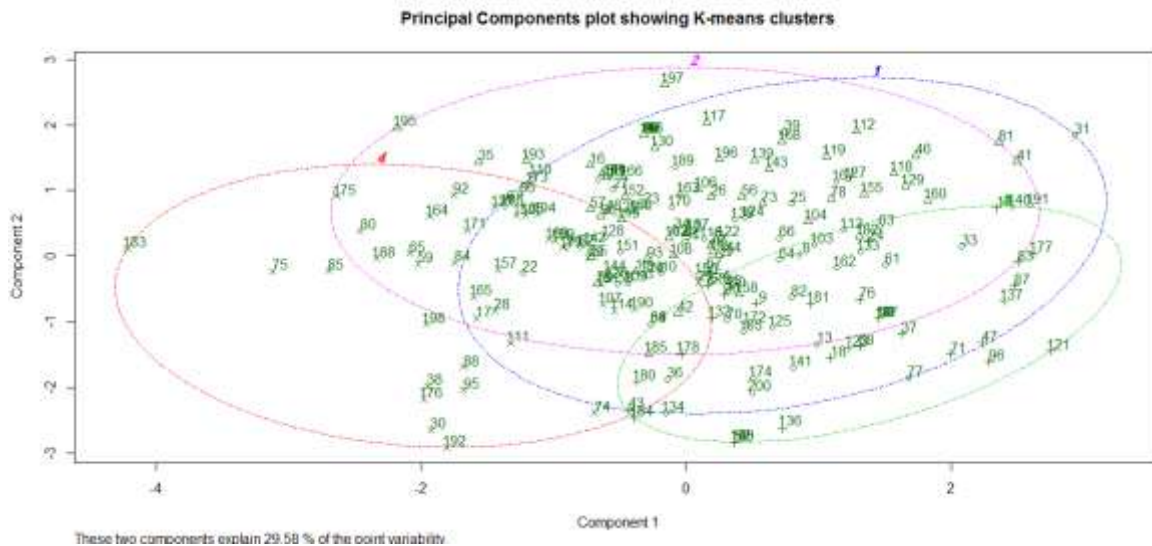
Figure (3.22) shows the 4-clustered scatterplot matrix for the 200 students in the first 10 learning outcomes. The figure shows the pairwise scatter plots of the first ten learning outcomes on a single view of a matrix format. It contains 10 rows and 10 columns where each row and column represent one dimension, and each cell plots a scatterplot of two dimensions.



**Figure 3. 22: Four cluster plot for the first ten learning outcomes of 200 student**

The figure below shows the PCA plot of 4 K-means cluster. In general, the stronger the clusters are, the more variance they have, they should not be overlapping in the distributions. It is worth to be mentioned that PCA plots are not useful in case of high dimensional data such as in our case.





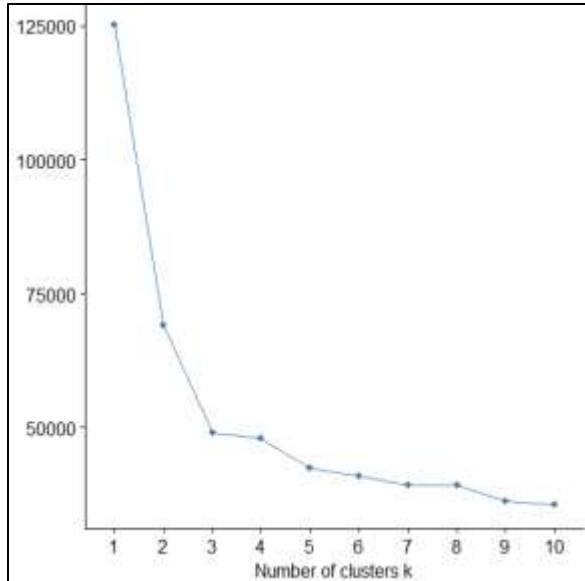
**Figure 3. 23: Plot of k-means of four-clusters solution for student achievement in learning outcome**

### How to determine the best number of clusters when using k-means?

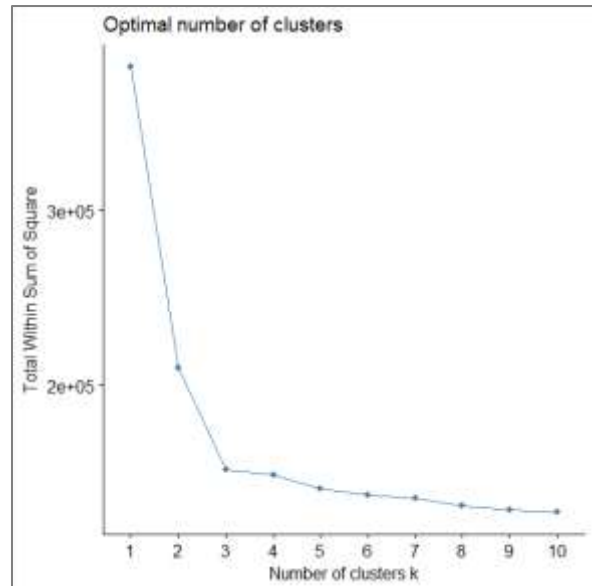
There are different methods for determining the optimal number of clusters for k-means:

1. Elbow method: The Elbow method looks at the total WGSS as a function of number of clusters: One should choose a number of clusters so that adding another cluster doesn't improve the total WGSS. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

The following figures 3.24 and 3.25 reflect the implementation of elbow method on 200-student dataset mentioned before. The left figure shows the elbow result on the first 10 learning outcomes showing the best number of clusters to be ~3, whereas the left figure shows the elbow result on 30 learning outcomes where also the best number of clusters is three.

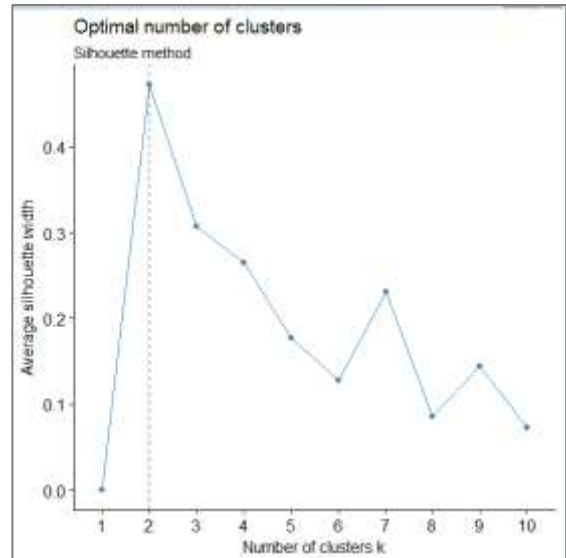


**Figure 3. 25: Applying Elbow method on the first 10 learning outcome for 200 students data set**

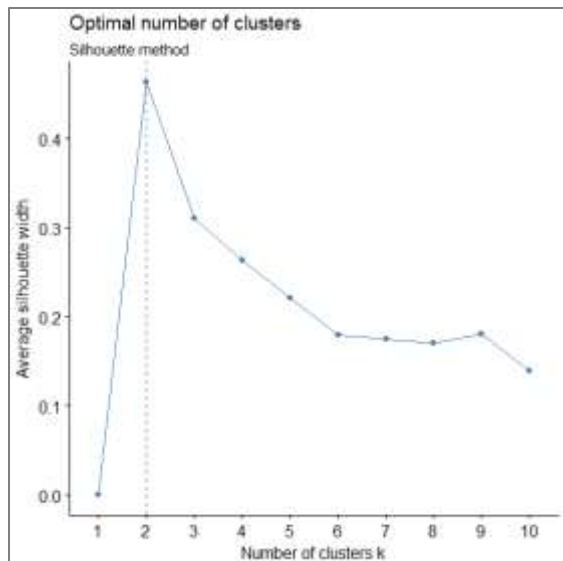


**Figure 3. 24: Applying Elbow method on all learning outcome for 200-students data set**

2. **The average silhouette approach** measures the quality of a clustering. Average silhouette method computes the average silhouette of observations for different values of  $k$ . The optimal number of clusters  $k$  is the one that maximizes the average silhouette over a range of possible values for  $k$ .



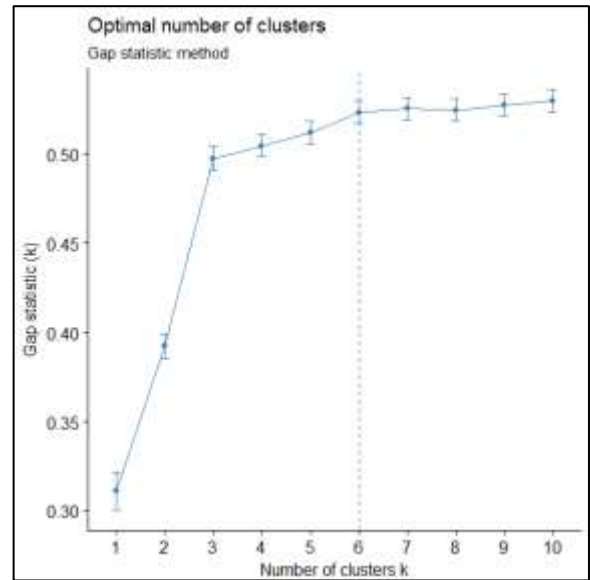
**Figure 3. 26:Applying Silhouette method on all learning outcomes for 200 students**



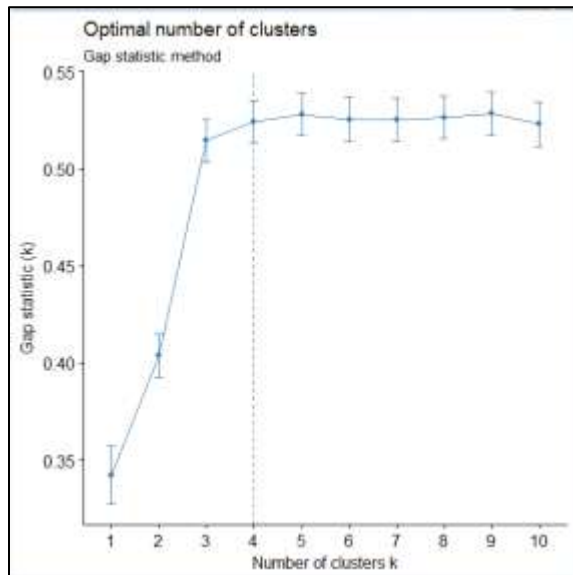
**Figure 3. 27: Applying Silhouette method on the first 10 learning outcomes for 200 students**

Figure 3.26 and 3.27 illustrate the Silhouette approach on the same 200-student data set, the left figure shows the result of the first 10 learning outcomes while right one shows the result on the complete dataset. The best number of clusters in both figures is two clusters.

3. **The gap statistic** calculates a goodness of clustering measure. The estimate of the optimal clusters will be a value that maximizes the gap statistic which means that the clustering structure is far away from the random uniform distribution of points.



**Figure 3. 28: Applying Gap Statistic method on all learning outcomes for 200 students**



**Figure 3. 29:Applying Gap Statistic method on the first 10 learning outcomes for 200 students**

Figure 3.29 and 3.28 illustrate the Gap static approach for 200 students' data set, figure 3.29 shows the result of the first 10 learning outcomes while figure 3.28 shows the result on the complete dataset.

4. **NbClust R function:** “NbClust package provides 30 indices for determining the relevant number of clusters and proposes to use the best clustering scheme from the different results obtained by varying all combinations of number of clusters, distance measures, and clustering methods.” (R Documentation)

Figure 3.21 and 3.22 show out the output of implementing the NbClust method on students’ data set. Figure 3.21 shows the conclusion of the best number of clusters to be two clusters based on the majority rule.

```
> nb <- NbClust(sdata, distance = "euclidean", min.nc = 2,
+ max.nc = 10, method = "kmeans")
*** : The Hubert index is a graphical method of determining the number of clust$
      In the plot of Hubert index, we seek a significant knee that co$
      significant increase of the value of the measure i.e the signif$
      index second differences plot.

*** : The D index is a graphical method of determining the number of clusters.
      In the plot of D index, we seek a significant knee (the signifi$
      second differences plot) that corresponds to a significant incr$
      the measure.

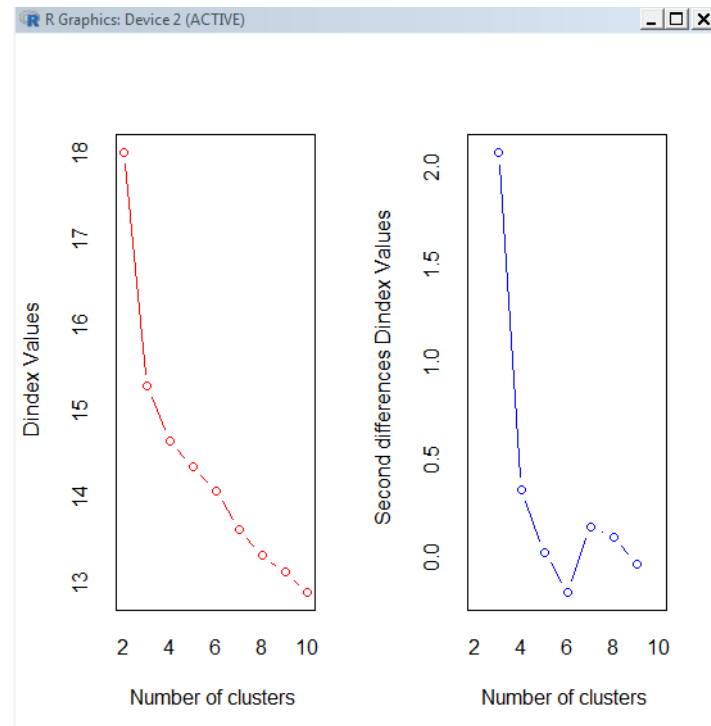
*****
* Among all indices:
* 12 proposed 2 as the best number of clusters
* 7 proposed 3 as the best number of clusters
* 1 proposed 4 as the best number of clusters
* 1 proposed 5 as the best number of clusters
* 1 proposed 7 as the best number of clusters
* 1 proposed 8 as the best number of clusters
* 1 proposed 10 as the best number of clusters

      ***** Conclusion *****

* According to the majority rule, the best number of clusters is 2

*****
,
```

**Figure 3. 30: Applying NbClust package on student data set**



**Figure 3. 31:D index to determine the number of index**

The figure below shows the cluster plot when applying two clusters on the first ten learning outcomes for 200 students.



**Figure 3.32: Cluster plot for the first ten learning outcomes of 200 students where  $k = 2$**

## Evaluating k-means Cluster

The Sum of Squared Error, or SSE, is one of the common measurements of error. For each point, the error is the distance to the nearest cluster, SSE is the sum of square for these distances. One way of reducing SSE is to increase the number  $K$ . The following formula illustrates SSE:

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(m_i, x)$$

Where  $x$  is a data point in cluster  $C_i$  and  $m_i$  is the center point for cluster  $C_i$

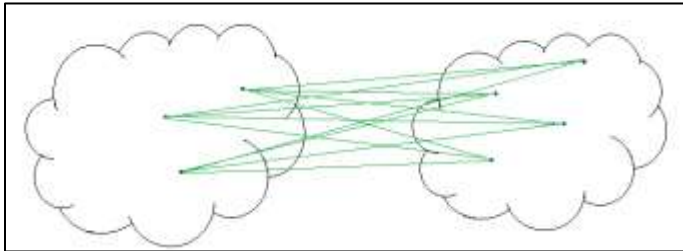
Cluster cohesion and cluster separation are also two important measurements in the quality of clusters. Cluster cohesion shown in figure (3.33) is the sum of the weights of all links within a cluster. It is denoted by WSS “within cluster sum of squares (SSE)”, the following is used to calculate the WSS:

$$WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

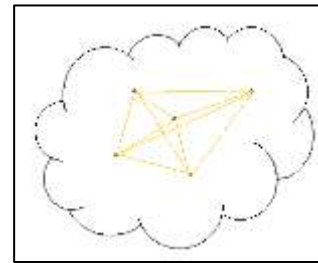
Cluster separation is the sum of the weights between observations in one cluster and another as illustrated in figure (3.34), it is denoted as BSS “between cluster sums of squares”. The following is used to calculate the BSS:

$$BSS = \sum |C_i|(m - m_i)^2$$

Where  $|C_i|$  is the size of cluster  $i$



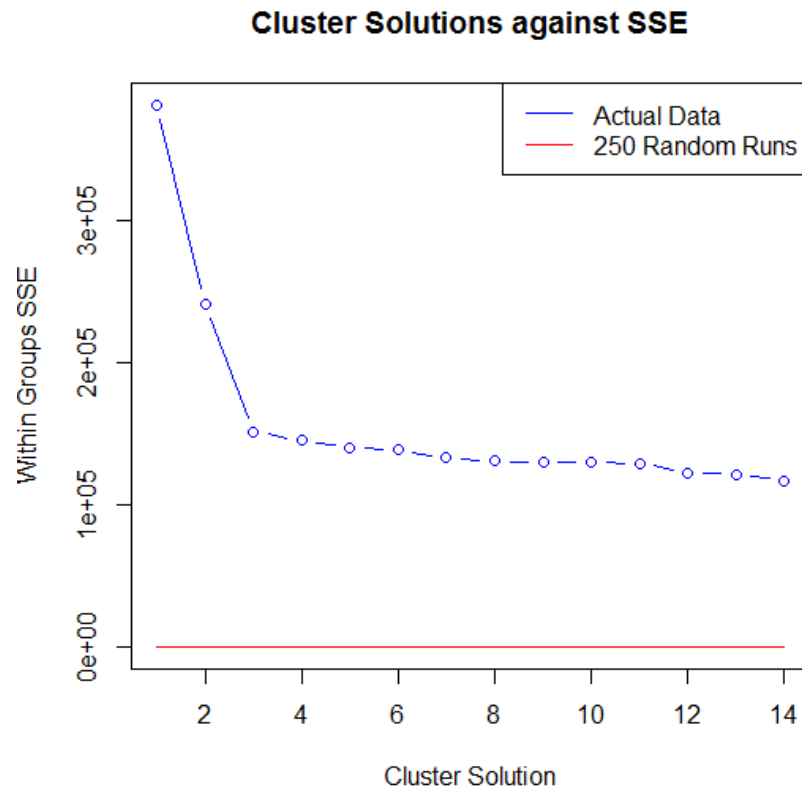
**Figure 3. 34: Cluster separation**



**Figure 3. 33: Cluster cohesion**

The “Cluster solution against SSE” shown in figure (3.35) is generated using the R script to measure the k-means cluster performance. The script represents an iterative process of re-evaluating individuals based on cluster centroid points; refer to appendix A for more details.

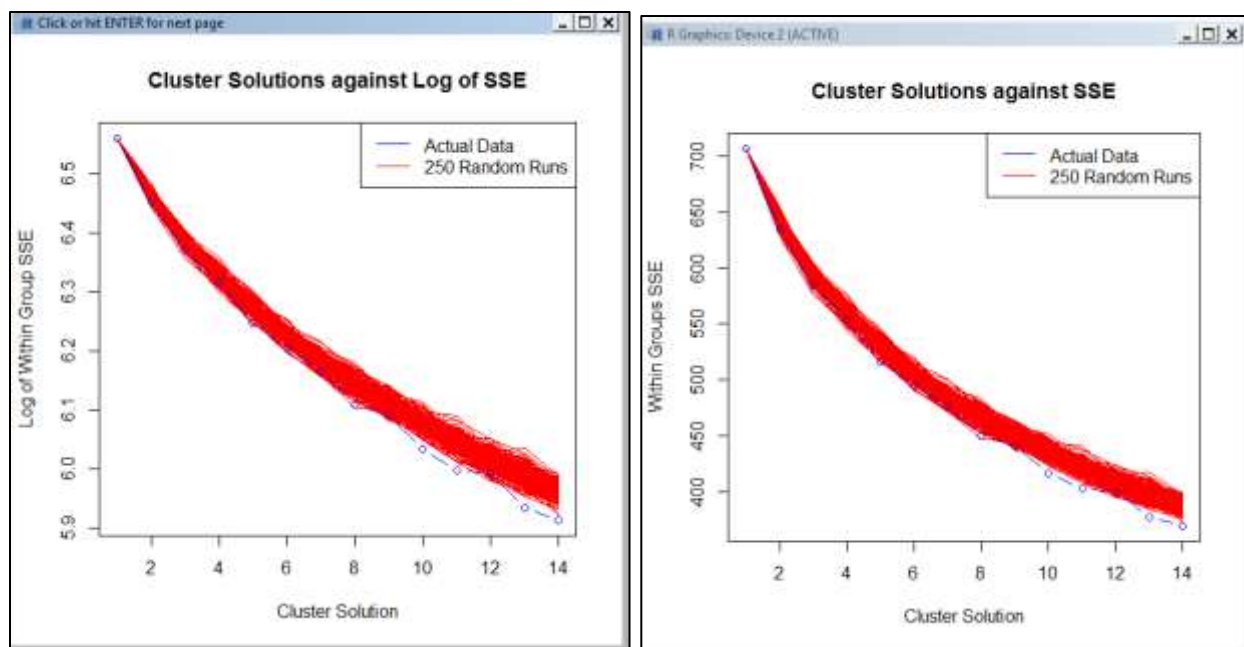




**Figure 3. 35:Plot of within-groups sum of squares error against number of clusters**

Figure (3.35) shows obviously an "elbow" at the 3 clusters solution suggesting that solutions  $>3$  do not have a substantial impact on the total SSE.

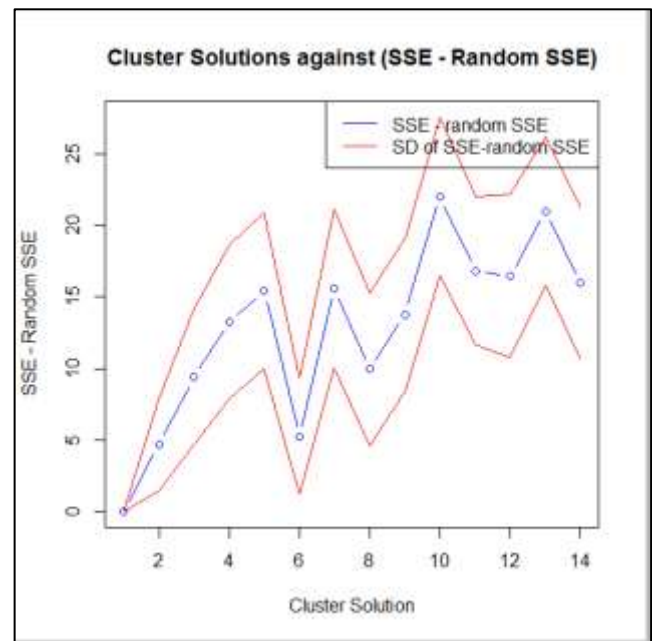
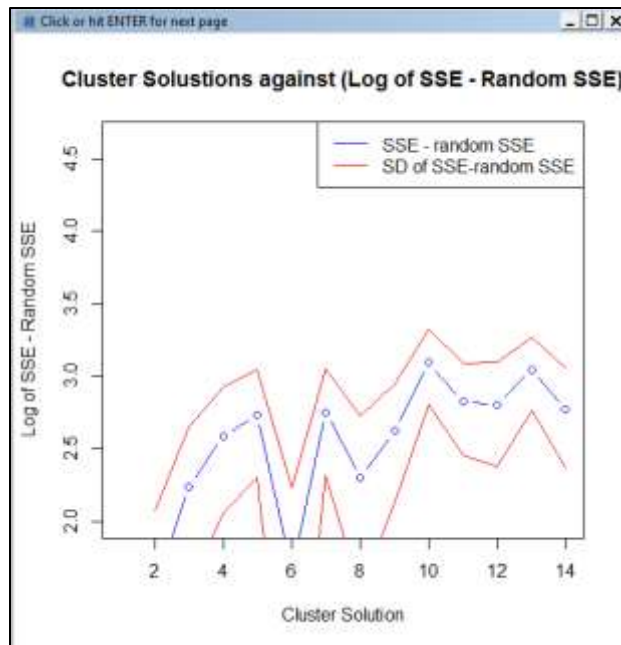
The k-means.R script provides more analysis to evaluate cluster solutions where the script calculates SSE against cluster solutions for a 200 randomized data. If a dataset has strong clusters, the SSE of the actual data should decrease more quickly than the random data as cluster level goes up. When plotting SSE against the number of tested clusters for both the actual and 250 randomized matrices for students' marks the results are shown in the figure below:



**Figure 3. 36: SSE against the number of tested clusters for both the actual and 250 randomized matrices**

Figure (3.36) shows that the SSE for the actual data does decrease faster than the 250 randomized datasets. This suggests that the dataset has structure and clusters are present.

Another way to evaluate the appropriate cluster solution is to examine the absolute difference between the actual and random SSE against the tested cluster solutions as described by k-means.R script. “An appropriate cluster solution could be defined as the solution at which the actual SSE differs the most from the mean of the random SSE”[45]. The k-means.R script displays the absolute difference between the actual and random (mean of all runs) SSE against the cluster solutions. Plots below are shown on both a log scale (left) and on a normal scale (right)



**Figure 3. 37: Absolute difference between the actual and random SSE against the cluster solutions**

## Chapter 4

# **Recommender System**

.

## **Introduction**

This chapter gives a general overview on recommender systems RS, concentrating on some of their approaches which are used in the methodology in chapter five such as content, collaborative and hybrid approaches. The chapter also highlighted some of the main challenges that faces the mentioned approaches and shows how the hybrid solution can overcome some of these challenges. The chapter also shows how RS in e-Learning environment differs from others that works in different environment such as in ecommerce; where the objectives and goals differ. The chapter is considered as an introduction for the next chapter in which the architecture of the suggested RS is presented.

### **4.1 Recommender system**

Because of the explosive growth of information, the overwhelming number of choices and information overload available on the Web, selection becomes very hard, causing a potential problem for the web user. From here the need of a recommender system (RS) appears and becomes a powerful tool in different domains from e-commerce (amazon, e-pay and Netflix) to digital libraries (e.g. ACM Digital Library) and knowledge management.

RSs are defined as software tools and techniques or agents that “intelligently” tries to recommend suggestions for items that are most likely of interest to a particular user. They are primarily directed towards individuals, personalized recommendations and offers a ranked list of items based on active users’ preferences and constraints [7] [6]. Personalization is defined as “the ways in which information and services can be tailored to match the unique and specific needs of an individual or a community”. [11]

“The recommendation problem can be defined as estimating the response of a user for new items, based on historical information stored in the system, and suggesting to this user novel and original items for which the predicted response is high.” [7]

Recommender systems are considered as a data science problem that requires the intersection between software engineering, machine learning, and statistics in order to build a successful recommender system. [10]

But what are the differences between searching the web for something which results in a ranked list of items and recommending a list of items for which also results are shown in a ranked list? Jeffrey M. O'Brien gave a beautiful answer on that: "Search is what you do when you're looking for something. Discovery is when something wonderful that you didn't know existed, or didn't know how to ask for, finds you". [8]

## **4.2 Recommendation in learning management systems**

Recommendation in the domain of learning management system differs from other domains such as e-commerce. In e-commerce the obvious goal of recommendations is to increase the profit. Profit is measured by money and so this goal can be achieved by attracting the users to the items that they may be interested with, so the number of users purchasing will increase and the organization profit will increase.

Whereas in e-learning the goal and the way in which this goal is measured is totally different, the goal here is improving learning, to enhance students' achievement and upraise their knowledge level.

Osmar R. Zaïane (2014) defines an e-learning recommender: "a recommendation system that would recommend a learning task to a learner based on the tasks already done by him and his successes and based on tasks made by other "similar" learners."

Digital Libraries are collections of information in different field such as science, business or personal data and it can be presented as digital text, image, audio, video or other media. [11]

In this research, the recommender system is mimicking a formal setting of learning which has a curriculum framework- such as universities and schools - and has a digital library, the source of learning material from which the engine recommends. In formal learning environment "there are usually well- structured formal relationships like predefined learning plans (curriculum) with

locations, student/teacher profiles, and accreditation procedures” [7]. This well-structured data can help in recommending courses through the adaptation of the learning materials to the students.

## **4.3 Recommendation Approaches**

### **4.3.1 Content-based recommender system**

The content-based filtering, also referred to as cognitive filtering, uses keywords to describe both resources and a user profile and then recommend items through the adaptation of items content and those preferred items in a user’s profile.

“Research on content-based recommender systems takes place at the intersection of many computer science topics, especially information retrieval and artificial intelligence” [7]. Most content-based recommender systems use text documents as the information source where documents can be represented as vectors in a multi-dimensional space using different methods such as the vector space model “tf-idf representation” and latent semantic indexing.

Many techniques and learning methods are used for learning a user’s profile such as relevance feedback, genetic algorithms, neural networks, nearest neighbor and the Bayesian classifier.

When choosing a learning method, many aspects are taken into consideration such as efficiency, accuracy and storage. For example, some learning methods such as genetic algorithms and neural networks are very complex and slower when compared with other learning methods such as Bayesian classifier. Some learning methods needs many training instances before they are able to make accurate predictions, this may not be appropriate in some environment where users’ interests “or profiles” change in short periods. The Bayesian classifier does not do well here whereas relevance feedback method and a nearest neighbor method can make a suggestion with one training instance.

New approach in content-based filtering is the semantic analysis by using ontologies. Ontology is a representation vocabulary, often specialized to some domain or subject matter, it’s sometimes used to refer to a body of knowledge describing some domain [38]. In this approach

more accurate learning is performed because of looking for the domain of knowledge to which key words belong. But still the cultural and linguistic background knowledge may affect ontology. As the knowledge domain of the same word may differ in two different cultures or even in different contexts.

### 4.3.2 Collaborative recommendation approaches

The collaborative recommender approach focuses on the similarity of user rating. Users are similar if their vectors are close according to some distance measure such as Jaccard, cosine distance or others distance methods mentioned in chapter 3.

Collaborative approach is the process of identifying similar users and recommending what similar users like base on their profiles.

The approach represents the entire users  $u$  and items  $I$  as a rating matrix  $A$  of  $u \times i$ , rows in the rating matrix represent users and columns represent items and each entry  $A_{ui}$  represents the rating or the  $u^{th}$  user at the  $i^{th}$  item.

Figure (4.1) shows a matrix of 14 users and nine items where the blue cells are items which are not evaluated by user, orange cells are rates to be predicted and the green column is used to predict “user9” rate for “item4”.

users \ items	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9
user1	3	3	5	2	3	4	5	2	4
user2	4	2	1	1	3	2	5	2	1
user3	1	5	3	2	3	2	4	3	3
user4	2	1	3	4	5	3	3	4	5
user5	5	0	5	3	3	1	2	2	3
user6	2	3	5	4	0	4	3	5	5
user7	5	2	1	5	5	1	5	3	2
user8	1	4	2	1	3	5	1	2	5
user9	1		4		3	4	0	3	3
user10	1	4	5	1	5	4	4	2	1
user11	4	5	3	5	1	4	1	2	5
user12	5	4	4	3	3	5	3	3	1
user13	3	2	1	5	5	3	0	2	3
user14	1	2	2	1	4	4	4	5	5

Figure 4. 1: matrix of users’ rates on nine items



### 4.3.3 Hybrid Recommendation Approaches

A hybrid recommender system attempts to combine different techniques such as collaborative filtering and content-based filtering in order to predict a more accurate recommendation.

Most hybrid methods applied user profiles and descriptions of items to find users who have similar interests, then used collaborative filtering to make predictions. Hybrid recommendation can be implemented in many ways: (1) different approaches working asynchronously and results are combined in order to give final results, e.g. making content-based and collaborative-based working separately and then combining both prediction results; (2) different approaches working synchronously by applying one approach on the result of another one; (3) or by unifying the approaches into one model.

Several studies compare hybrid with the pure collaborative and content-based methods and find out that hybrid approach can provide more accurate recommendation and overcome some recommendation methods problems such as the “cold-start” and sparsity problems.

### 4.3.4 Challenges and Issues

1. **Cold-start:** it is also known as the “cold start problem” and occurs when it is difficult to give recommendation because of “new cases” which have no history or empty profile, for example: new user or new items. In the case of learning environment, this will occur when new students (junior Students) or new courses is defined.  
This problem could be solved by using hybrid recommender systems or by using a survey when creating a new profile.
2. **Trust:** all users rating is taken into consideration and treated the same weight regardless of the type of user profile which may be a rich profile “experienced user” or a poor user.  
This problem could be solved by distribution of priorities to the users.
3. **Scalability:** in recommender systems and mainly in collaborative methods, history is very important to find out similarities, this leads to a rapid growth of data (users, items and profiles) so systems will need more resources for processing information and performing recommendations.

This problem can be overcome by various types of filters and physical improvement. Also, parts of numerous computations can be implemented offline in order to accelerate issuance of online recommendations.

4. **Sparsity:** sparsity is the problem of lack of information where the rating matrix of similar users have null values, these null values indicate that a user didn't rate an item.
5. **Privacy:** this problem becomes the most important problem if the access to the user profile is limited due to reliability, security and confidentiality reasons.

## 4.4 Collaborative Recommender system algorithms

There are different types of algorithms to build recommender systems. Some of them are explained below:

### 4.4.1 Memory-based algorithms

The memory-based algorithm is a collaborative recommender approach which uses the entire data set in order to find similar users to the active user. Similarity between users can be measured using different similarity measures such as correlation distance and cosine mentioned in chapter three, sections 3.4.2 and 3.4.4.

To predict the rating for an item  $x$  for an active user  $y$ , the weighted sum or regression method could be used where both approaches are trying to capture how the active user rates the similar items.

The weighted sum approach predicts the rate by directly using the ratings of similar items and how much these items are similar. The prediction can be calculated using the following formula:

$$P_{u,i} = \frac{\sum_{all\ similar\ items, N} (s_{i,N} * R_{u,N})}{\sum_{all\ similar\ items, N} (|s_{i,N}|)}$$

Where  $s$  is the set of similar items,  $R$  is the rate of  $u$  user for the similar items

In the regression approach, rate is weighted based on a linear regression model expressed by:

$$\overline{R}'_N = \alpha \overline{R}_i + \beta + \epsilon$$

Where  $\alpha$  and  $\beta$  parameters are determined by going over rating vectors.  $\epsilon$  is the error of the regression model. [40]

#### **4.4.2 Model-based algorithms**

In this approach, collaborative filtering algorithms provide recommendation by developing modules using data mining and machine learning algorithms such as Bayesian networks, clustering models, singular value decomposition and Markov decision process which can provide high scalability because of not using all data. On the other hand, the quality of prediction may be affected and this depends on the way models are built.

## Chapter 5

### **Methodology**

## **Introduction**

In this chapter the overall architecture of the suggested recommender engine is illustrated in both sections 5.1 and 5.2. The recommender engine followed the hybrid solution of parallel content and collaborative approach that works simultaneously. Each approach results in a recommendation list which are inserted to a final stage that calculate the final weights of learning material and ranks them in a final recommendation list. The chapter concentrates on the collaborative approach leaving the content approach for future studies as mentioned before.

In section 5.3, the chapter explain how the dataset is generated and in section 5.4 it illustrated the data model and database design.

In order to find out the similarity between students, two types of matrices: sparse and dense matrix are explained in section 5.5. whereas students learning patterns and feedback are measured using one of two student response indicators mentioned in section 5.6.

Finally, the suggested recommender engine is designed to work in a dynamic environment which enable the admin user to configure a set of parameters explained in section 5.7.

### **5.1 Recommender Engine Architecture**

The proposed architecture is mimicking a curriculum learning environment where courses are fixed for all learners and do not adapt to individuals. The course content and its delivery are static while the organization of digital library is dynamic.

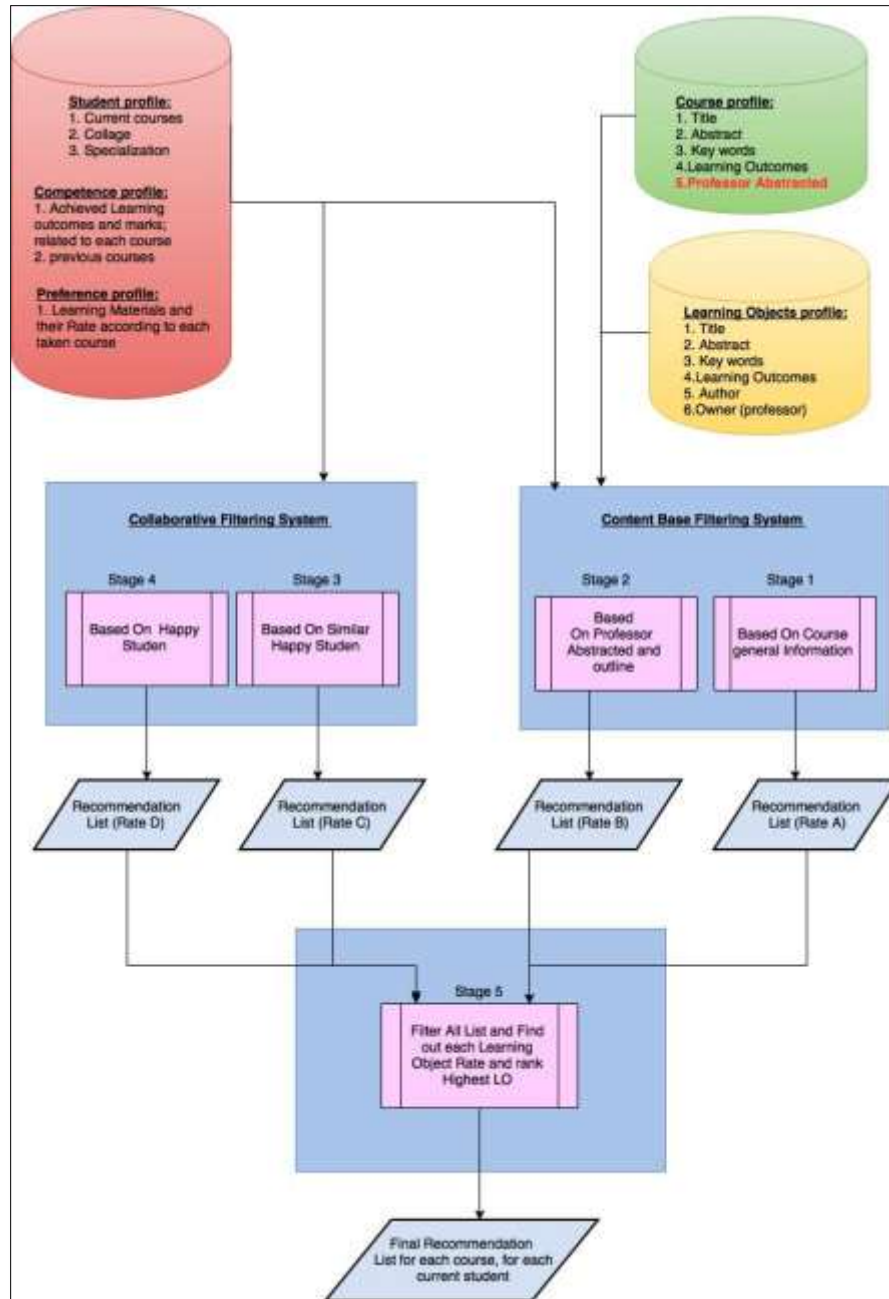
On the other hand, it is widely recognized that students have different preferred learning styles and knowledge background. Very few course management systems accommodate any dynamic component that can follow learners' progress, build intelligent profiles and provide contextual individual help.

This section explains the architecture of a suggested hybrid recommender engine with its two approaches: content and collaborative approaches. The suggested collaborative approach attempts to measure students' knowledge based on their performance in previous courses looking at their achievements at the different levels of learning outcomes and total achievement in

learning outcome levels; this solution considers the availability of the information in students' achievements profiles and the digital library as a part of the ILTS and QLearn project [43][44]

The student's achievement profile consists of: (1) achieved courses and courses' learning outcomes with the achieved mark on the level of learning outcome, (2) the recommended learning materials - which were recommended for a student at the time he took the course, (3) and student's rating for each learning material which records his feedback on how much the learning material was useful for him based on his knowledge background at that time. On the other hand, each learning material is linked with a set of learning outcomes which were predefined whenever the learning material was uploaded to the learning library based on ILT. [43]

As mentioned earlier, the suggested recommender engine is a hybrid recommender engine with its two approaches, content and collaborative approaches. The two approaches work in parallel behavior, each approach involved two imbedded stages and results in its own recommender list, the final stage of the recommender system – which is the fifth stage - is responsible for combining the recommendation lists and results in a final ranked list. Figure (5.1) shows the overall suggested architecture for the e-learning recommender engine.



**Figure 5. 1: Overall architecture for an E-Learning recommender engine**

This architecture will give a recommended learning material based on students' level of knowledge making use of other students' experience whenever history data exists, and so the engine will build its experience with time and will be able to give better recommendation. But when no "history data" exists (cold start problem) then the content recommender engine will

take the lead and give a recommendation list based on the similarity between the course profile and learning material profile.

By this architecture, the engine will not worry about the cold-start Problem where new courses pose a significant challenge. Also, the recommendation will be more accurate as two approaches are used (section 4.3.3).

As shown in figure (5.1), the e-learning recommender system is based on a hybrid filtering approach implemented in five stages, four of these stages can work in parallel while the final stage will wait for the output of all stages to start working. In the final stage, the learning objects scores will be recalculated making use of the recommender engine setup (section 5.7) and will result in a final recommendation list ranked based on final scores which will be illustrated in section 6.3.

The suggested recommender engine consists of:

1. Content based filtering system: which contains two stages; based on the source from which it is reading, each stage will give its own recommendation list; but the priority of each list will differ according to the stage which performs this list, this will be used in calculating the learning objects scores and their ranking in the fifth stage.

The content-based filtering system is very important in suggesting learning materials, it will guarantee that the engine will work at any time and gives a recommendation list, especially when speaking about cases where the system has no experience such as “cold start problem” such as in following examples:

- New courses which no students have taken it before; for example, the computer science department at al-Quds university decided to give “genetic algorithm” course for the first time. So, there is no experience at any of the students about this course and thus the system collaborative filtering will not work in an appropriate way while content-based system will pick out all recommended material based on the professor’s recommendations, course abstract, learning material information and also based on linked learning outcomes to both the course and learning material
- New specialization: for example, the computer department in al-Quds University decided to open a new specialization such as software engineering and game theory. All students registered in this new specialization - in their first semester - may suffer



from the lack of similarity between them and other students. Here the role of the content-based filtering system will appear again.

*Note: This approach with its two stages are considered as future study.*

2. Collaborative based filtering system: this system will work beside the content-based filtering system; it guarantees that the suggested recommender engine will be intelligent and will increase its accuracy and experience by time. The Collaborative based filtering system also contains two stages. The first stage depends on the similarity between current students in a special course and all other previous students who passed the active course in a very good mark. Based on this similarity, all those similar and excellent students' "interactive profile" will be measured and evaluated related to the active course. And this stage will give a recommended list which will have the highest rate between all other recommendation lists of other stages.

The second stage in collaborative based filtering system will look at the rest of excellent students in the current course (students not included in the first stage) and will give recommendation list based on their "interactive profile".

3. The final stage (stage 5) will take the output of the first four stages and will calculate the number of replicas of each learning object in the four lists, knowing that the score of each learning object will differ even if it appears in the four lists (this is determined based on the priority of the information source in each stage).

This stage will give a final recommended list ranked according to the highest scores which will reflect the priority and importance of a learning object.

## 5.2 Collaborative Recommender Approach

Figure (5.2) shows the proposed model of the collaborative recommender engine for learning materials.

The figure shows the four steps of the suggested model:

**Step one:** in this step the source of information to which the recommender system will be linked is defined, the source of information could be any learning environment such as schools, universities or even a training center.

Supposing that the source of the information is a university, the university must provide the recommender engine with data such as the list of courses, course learning outcomes, list of learning outcomes and their levels, student achievement in each learning outcome, current student courses, list of students' learning material and student learning material rates.

**Step Two:** in this step the recommender engine will start looking through the whole history of the university to find out the set of students who are similar to the active student with his current status. The student's current status means the set of learning outcomes which the student passes since joined the university and achievement level in each learning outcomes.

Result of step two will be: (1) the set of similar students to the active student, the result will be saved in staging tables which the recommender engine will refer to during its recommendation process. From the set of similar students, the engine could find the best student, depending on the best student definition configured in the engine setup. (2) Also, the engine finds out the set of best students in the student's active course regardless of the similarity of those students to the current student; in order to study the general learning behavior of these students.

**Step three:** based on the set of best similar students, the engine starts looking in the learning material these students used, and their rating for these learning materials and the number of hits recorded. Based on that, the recommender engine builds a list of learning materials in which each learning material was given a weight.

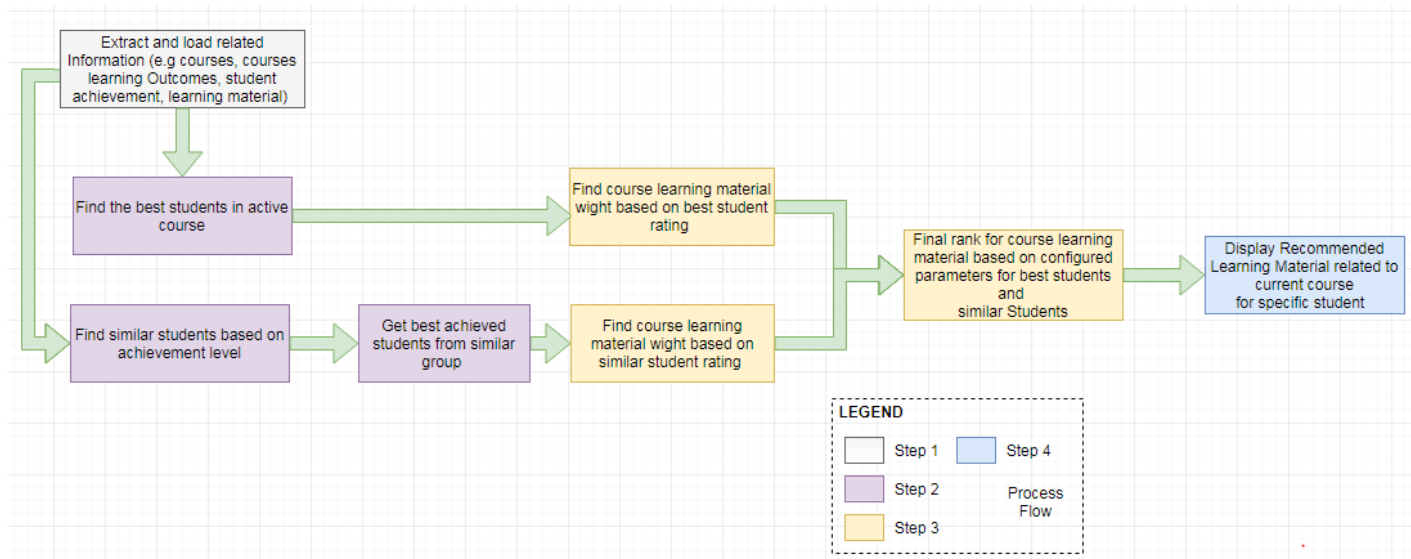
The result of step three will classify the learning materials list in two categories:

1. List of learning materials based on best similar students.
2. List of learning materials based on best students only.

The engine will refer to the configuration setup, to find out the configured weight for each category, and so will find out the final weight for each material which will be declared in section 6.3.

**Step four:** Based on the results of step three, each student will be given a list of learning materials which are ranked based on learning materials' weights, higher weights will be ranked first.

The engine will give feedback to the source; e.g. a university in our case, with the list of active students, their current courses and the list of recommended learning materials for each course.



**Figure 5. 2: Block diagram for the proposed model of Educational Recommender System – Collaborative Approach**

### 5.3 Data preparing – Creating the Dataset

The ideal way to simulate the student environment and test the results of the suggested recommender system, would be in finding real data of universities, schools...etc., and use the data to build a recommender system. But this is not feasible for many reasons, the main one is because of the surrounded learning organization doesn't record students' achievement to the level of learning outcomes. On the other hand, the target of this study is to build a general and scalable recommender engine which can fit any institute, university or school regardless of its database structure.

To overcome the problem of data availability, a random dataset is generated to define students' achievement in each course which is detailed to the level of learning outcomes and learning outcomes levels, also students' learning materials and rates are auto generated randomly.

The generated dataset consists of:

1. List of computer science students registered to the department for seventeen academic years.

2. List of courses related to computer science specialization which includes mandatory and elective courses according to the university of Toronto – computer science specialist<sup>2</sup>
3. List of learning outcomes, each learning outcome is mapped to revised bloom taxonomy level and each course was given a random number of learning outcomes ranged between five to eight learning outcomes. The study considered four taxonomy levels (Remember, understand, apply and analyze) where each learning outcome is mapped to only one level.
4. List of student courses where each graduated student must finalize all mandatory courses and a random set of elective courses resulting in a full 120 hours; the number of hours required for a computer science student to graduate.
5. List of students' learning outcomes, each student was given a set of learning outcomes related to their achieved and active courses.
6. Students' achievement on the level of learning outcome which are generated randomly to simulate the actual reality of students.
7. List of learning materials, each learning material was mapped to a random set of learning outcomes.
8. Students' feedback on learning materials which were recommended previously –by the engine, each recommended learning material has (1) a rate which reflects the student's feedback and (2) number of online hits which reflects students' accesses time to that learning material.

### **Generating Data**

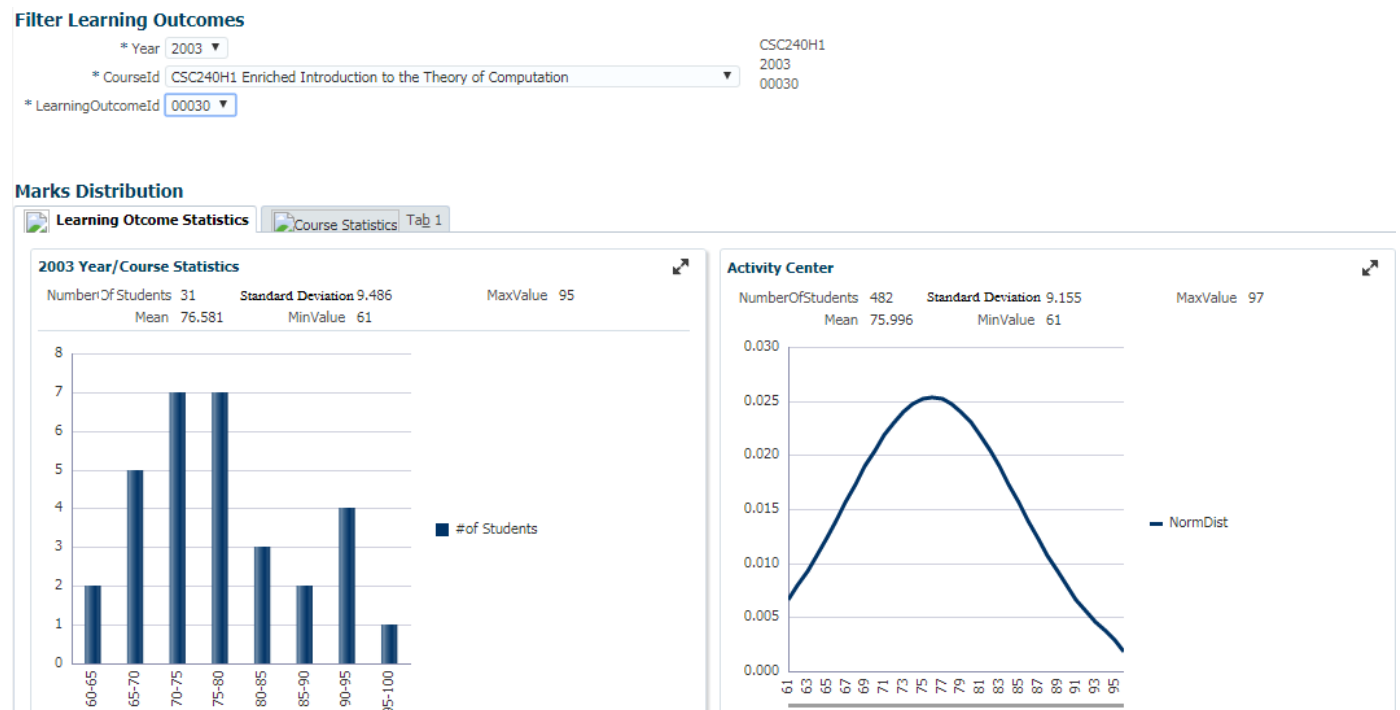
Before generating the dataset, the following points were taken into consideration:

1. A random set of two to five students from each academic year was considered to be the excellent students whose marks were always good thorough their four years studies at the university.
2. On the other hand, a random set of five to ten students was considered to be weak students whose marks' average was between 60 and 72 thorough their four years studies at the university.
3. The rest of students in each academic year were given a random mark average between 60 and 90 in their learning outcomes.

---

<sup>2</sup> [http://calendar.artsci.utoronto.ca/crs\\_csc.htm#ASSPE1689](http://calendar.artsci.utoronto.ca/crs_csc.htm#ASSPE1689)

Figures (5.4) show the normal distribution of generated data which emphasis that the data is mimicking the reality of students' environment.



**Figure 5. 3: Normal distribution of students' marks**

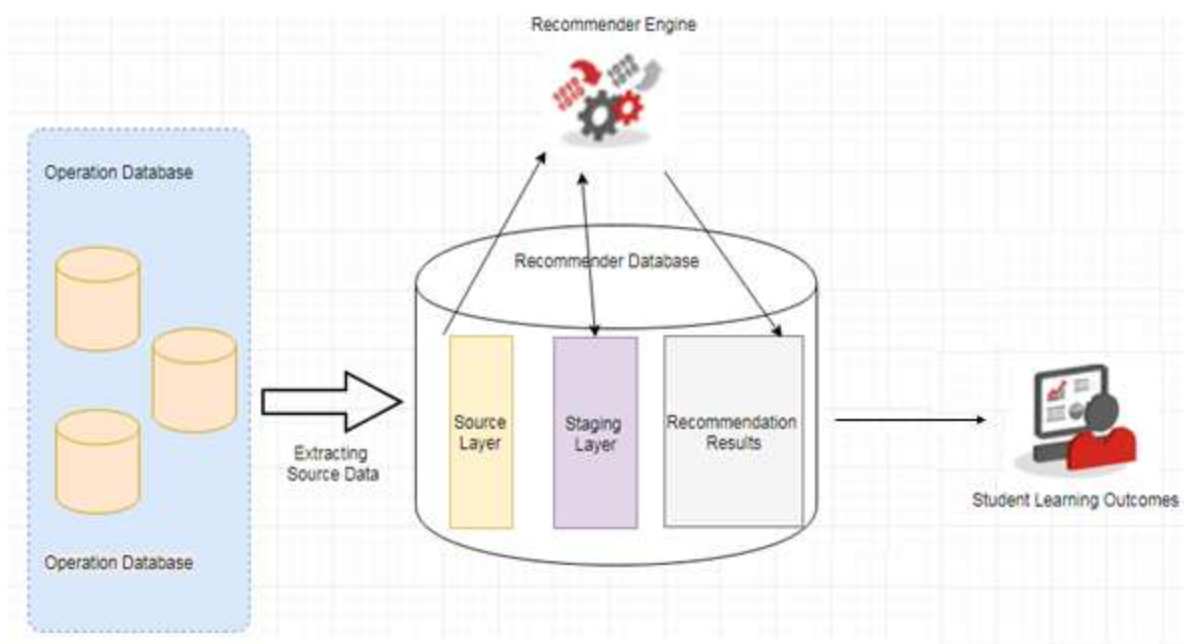
Figure 5.4 above reflects a sample from the generated data, the bar chart shows students' achievements in learning outcome id "00030" of course "Enriched Introduction to the Theory of Computation" in the academic year of 2003. The figure shows that the number of student who took that learning outcome in that year is 31 students, maximum mark achieved is 95 and the minimum mark is 61, the mean of student achievements is 76.58 and the standard deviation is 9.48.

The right line chart shows the normal distribution of students' achievements in that learning outcome all over the academic years where the total number of students who took the learning outcome is 482 students, maximum mark achieved is 97 and the minimum mark is 61, the mean of student achievements is 75.99 and the standard deviation is 9.15.

## 5.4 Data Model

The data model of the recommender engine is designed for query and analysis rather than for transaction processing. It contains historical data derived from a transactional data exists in a university, school or any other learning system database. So, the integration between the recommender engine and any other learning system is close to be subject orientation where the recommender engine requires information related to students, courses, learning outcomes, digital libraries (learning material), student's achievement in each learning outcome, and student's feedback on each learning material..

The database is divided into three layers: (1) source layer which contains the requires information extracted from the source transactional database (learning management system e.g. university, school ...etc.) and loaded into the source layer; (2) staging layer: data in this layer is stored temporary and holds data which the engine will refer to while the recommendation process is taking place; (3) results Layer holds the results of the recommendation process which contains all active students with their courses and the list of recommender materials for each course. This layer will be deleted at the binging of each recommendation process



**Figure 5. 4:Data model in recommender engine**

The following table summarizes the database tables where each table is prefixed with its layer type (e.g. SR: source, ST: stage, RS: result):

**Table 5. 1 Database tables classified into three layers**

Stage	Table	Remarks
Source	SR_COURSES	
	SR_LEARNING_OUTCOMES	
	SR_COURSE_LEARNING_OUTCOMES	
	SR_LEARNING_OBJECTS	
	SR_LEARNING_OBJECT_OUTCOMES	
	SR_STUDENTS	
	SR_STUDENT_LEARNING_OUTCOMES	
	SR_STUDENTS_LEARNING_OBJECTS	
Stage	ST_BATCH_PROCESS	
	ST_BATCH_PROCESS_DTL	
	ST_STUDENT_SIMILARITY	
Result	RS_RECOMENDED_LEARNING_OBJECTS	

For more details on database schema, data model and ERD refer to appendix B.





```

query2 ="SELECT c.student_id,\n" +
"      c.learning_outcome_id,\n" +
"      c.learning_outcome_mark,\n" +
"      c.course_id\n" +
"    FROM student_lo_training c\n" +
"   WHERE      EXISTS\n" +
"              (SELECT '*' \n" +
"                FROM student_lo_training d\n" +
"               WHERE      d.course_id = '"+courseId+"' \n" +
"                          AND d.student_id = c.student_id)\n" +
"              AND EXISTS\n" +
"              (SELECT a.learning_outcome_id\n" +
"                FROM student_lo_training a \n" +
"               WHERE      a.student_id = '"+StudentId+"' \n" +
"                          AND a.learning_outcome_mark IS NOT NULL\n" +
"                          AND a.learning_outcome_id = c.learning_outcome_id\n" +
"              )\n" +
" ORDER BY student_id";

```

**Figure 5.8: Sparse Matrix Query**

This research also compares the recommendation results when the engine builds a dense matrix of students' marks  $R$  of  $S \times L$  where  $R$  represents the list of students who took exactly the same learning outcomes such as the active student  $X$  and  $L$  is the list of learning outcomes which was achieved by student  $X$ . In this case the options will be very limited as some universities programs give the student a freedom to choose courses in their third and fourth year. The following figures show the dense matrix and dense matrix query:

```

65.0,61.0,66.0,65.0,64.0,61.0,62.0,61.0,66.0,65.0,63.0,63.0,62.0,63.0,64.0,62.0,62.0,62.0,66.0,62.0,62
78.0,76.0,78.0,75.0,78.0,75.0,76.0,76.0,77.0,77.0,79.0,78.0,78.0,79.0,79.0,75.0,76.0,78.0,76.0,76.0,74
72.0,74.0,74.0,71.0,75.0,72.0,70.0,72.0,74.0,71.0,75.0,72.0,73.0,74.0,73.0,71.0,72.0,74.0,73.0,73.0,70
68.0,69.0,66.0,66.0,65.0,69.0,67.0,68.0,68.0,67.0,67.0,69.0,67.0,66.0,67.0,65.0,68.0,67.0,67.0,67.0,67
75.0,77.0,77.0,74.0,75.0,74.0,77.0,74.0,78.0,77.0,76.0,77.0,76.0,74.0,76.0,73.0,75.0,76.0,77.0,74.0,75
72.0,69.0,71.0,69.0,71.0,71.0,72.0,69.0,69.0,72.0,71.0,69.0,71.0,68.0,71.0,69.0,69.0,70.0,71.0,72.0,70
80.0,79.0,80.0,79.0,82.0,78.0,80.0,82.0,81.0,83.0,78.0,80.0,80.0,80.0,83.0,81.0,78.0,80.0,82.0,78.0,80
,80.0,80.0,80.0,77.0,81.0,81.0,77.0,79.0,77.0,78.0,76.0,77.0,81.0,78.0,79.0,80.0,78.0,77.0,76.0,79.0,7
,67.0,67.0,69.0,68.0,67.0,66.0,65.0,66.0,67.0,69.0,65.0,65.0,66.0,67.0,68.0,66.0,67.0,70.0,69.0,67.0,7
,69.0,67.0,70.0,68.0,66.0,70.0,67.0,71.0,67.0,69.0,67.0,69.0,68.0,67.0,66.0,67.0,66.0,68.0,70.0,6
,74.0,74.0,70.0,74.0,73.0,75.0,73.0,74.0,75.0,70.0,74.0,73.0,75.0,73.0,74.0,71.0,75.0,73.0,72.0,72.0,7
,76.0,76.0,74.0,79.0,77.0,79.0,78.0,76.0,75.0,77.0,75.0,78.0,78.0,78.0,77.0,77.0,77.0,76.0,76.0,77.0,7
,75.0,78.0,75.0,78.0,77.0,77.0,76.0,77.0,78.0,75.0,77.0,74.0,75.0,77.0,75.0,78.0,73.0,74.0,76.0,76.0,7
,75.0,74.0,76.0,78.0,76.0,77.0,73.0,77.0,73.0,77.0,75.0,75.0,76.0,77.0,77.0,76.0,77.0,75.0,76.0,77.0,7
,87.0,87.0,85.0,86.0,84.0,84.0,85.0,84.0,85.0,87.0,86.0,83.0,86.0,84.0,86.0,87.0,84.0,86.0,87.0,87.0,8
,96.0,91.0,91.0,93.0,90.0,92.0,96.0,93.0,92.0,96.0,93.0,92.0,95.0,94.0,94.0,93.0,96.0,93.0,96.0,93.0,9
,74.0,78.0,73.0,80.0,72.0,79.0,70.0,74.0,71.0,73.0,66.0,77.0,78.0,80.0,67.0,78.0,70.0,78.0,69.0,79.0,7
,67.0,69.0,72.0,68.0,70.0,81.0,69.0,73.0,74.0,69.0,68.0,69.0,66.0,76.0,68.0,80.0,70.0,69.0,78.0,6
,66.0,65.0,66.0,65.0,67.0,67.0,68.0,64.0,66.0,66.0,66.0,64.0,67.0,67.0,68.0,68.0,64.0,65.0,66.0,68.0,6
,96.0,95.0,92.0,94.0,95.0,94.0,97.0,91.0,95.0,94.0,96.0,93.0,94.0,96.0,96.0,96.0,96.0,92.0,97.0,93.0,9
,66.0,79.0,81.0,71.0,68.0,81.0,67.0,76.0,74.0,66.0,69.0,75.0,77.0,75.0,76.0,79.0,68.0,79.0,71.0,79.0,6
,65.0,68.0,67.0,66.0,66.0,67.0,67.0,66.0,64.0,67.0,65.0,64.0,67.0,63.0,63.0,67.0,65.0,68.0,67.0,65.0,6
,69.0,67.0,67.0,64.0,67.0,64.0,68.0,68.0,65.0,68.0,66.0,66.0,67.0,65.0,66.0,68.0,66.0,69.0,67.0,65.0,6
,91.0,92.0,93.0,92.0,97.0,98.0,93.0,94.0,97.0,92.0,93.0,92.0,91.0,93.0,92.0,96.0,95.0,92.0,97.0,93.0,9
,97.0,92.0,95.0,94.0,95.0,94.0,92.0,95.0,93.0,93.0,91.0,96.0,96.0,94.0,93.0,96.0,96.0,94.0,94.0,94.0,9
,74.0,77.0,75.0,78.0,74.0,76.0,77.0,76.0,73.0,77.0,76.0,74.0,76.0,76.0,73.0,73.0,77.0,74.0,74.0,73.0,7
,68.0,68.0,72.0,67.0,70.0,68.0,70.0,69.0,71.0,68.0,69.0,69.0,71.0,68.0,68.0,68.0,70.0,69.0,70.0,70.0,7
,64.0,61.0,64.0,62.0,61.0,62.0,63.0,64.0,63.0,60.0,65.0,65.0,62.0,62.0,62.0,64.0,63.0,61.0,62.0,61.0,64.0,6

```

**Figure 5.9: Dense matrix of student's marks**

```

// student_recommendation

query2 ="SELECT student_id,\n" +
"      learning_outcome_id,\n" +
"      learning_outcome_mark,\n" +
"      course_id\n" +
"    FROM student_lo_training c\n" +
"   WHERE NOT EXISTS\n" +
"      (SELECT learning_outcome_id\n" +
"        FROM sr_student_learning_outcomes\n" +
"       WHERE student_id = '"+StudentId+"'\n" +
"       AND learning_outcome_mark IS NOT NULL\n" +
"      MINUS\n" +
"      SELECT learning_outcome_id\n" +
"        FROM sr_student_learning_outcomes\n" +
"       WHERE student_id = c.student_id\n" +
"       AND learning_outcome_mark IS NOT NULL )\n" +
"      AND learning_outcome_id in\n" +
"      (SELECT learning_outcome_id\n" +
"        FROM sr_student_learning_outcomes\n" +
"       WHERE student_id = '"+StudentId+"'\n" +
"       AND learning_outcome_mark IS NOT NULL)\n" +
"      ORDER BY student_id";
}

```

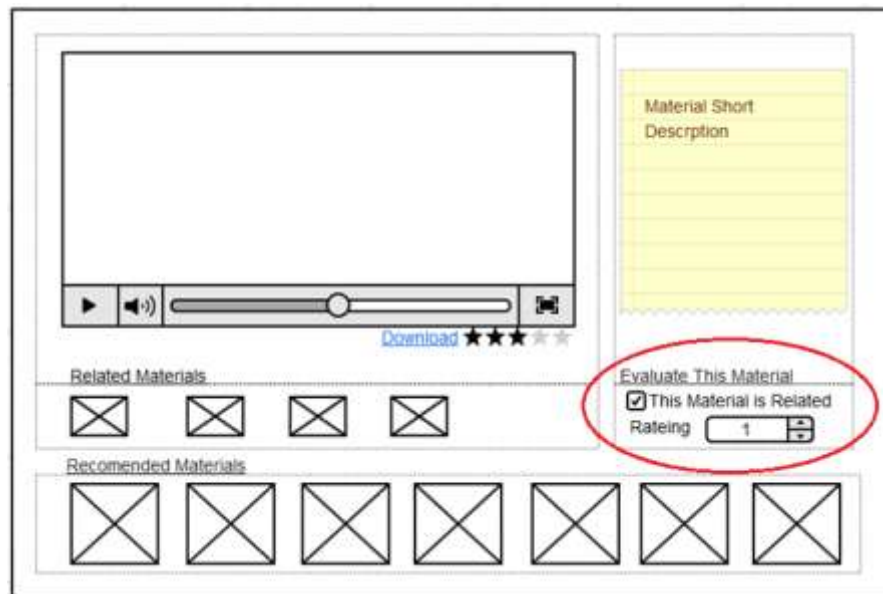
**Figure 5.10: Dense Matrix Query**

Type of matrix is configured before generating students' recommendation. No matrix factorization was done and null values mean that students have not taken the learning outcome yet and so their marks are considered as zeros, this makes their vectors further from the active student.

## 5.6 Students' responses indicators

In order to study students learning behavior and students' interaction with learning materials, the study considered two ways in which students' responses are obtained, an explicit feedback in which students can enter rates explicitly after reading a learning material and so giving their opinion on it. Or implicit feedback from students accesses patterns, exactly the indicator of their number of hits for a learning material within a semester which indicates how many times the students is referring to a learning material within a semester when compared with other learning materials related to the same course.

This research considered that no more than one rating can be made by a student for a particular learning material where rates are given out of five. The following figure shows a prototype for learning material rating:



**Figure 5.6: Prototype for evaluating learning material**

## **5.7 Recommender Engine Setup – Inputs and configuration**

The research provides a solution that enables the admin user to control the recommendation engine based on configurable inputs in the “Recommendation Engine Setup” page as shown in the following screenshot:

**Recommender Engine Setup**

**Student Similarity Parameters**

Similarity Measure

Algorithm

Matrix Type

Achievement level

**Recomender Types**

Collabrative Recomennder weight

Content Recomennder weight

**Collaborative Recomennder Parameters**

Good Marks

Similar Student weight

Excellent Student weight

**Figure 5.11: “Recommender Engine Setup” page**

In the “Student Similarity Parameters” section, the admin can:

1. Configure the number of clusters “K” for the k-means algorithm.
2. Choose one of the five similarity measures: distance Euclidian, distance Manathan, distance Cosine, distance Correlation and distance Jaccard (refer to section 3.4 for more details).
3. Determine the matrix type sparse or dense (refer to section 5.7 for more details)
4. Determine achievement level to which the matrix will be built: learning outcomes or learning outcomes levels.

In the “Recommender Type” section, the admin configures the weight of the resulted recommendation lists from both collaborative and content-based approaches, this will affect the results of the final stage in the recommendation process while building the final recommendation list.

The final section “Collaborative Recommender Parameters” allows the admin to configure the mark of good student and the weight of the good student in the similar group “Stage 3 in Recommender Engine Architected, section 5.1” and the weight of good student in general “Stage 4 in Recommender Engine Architected, section 5.1”.

## Chapter 6

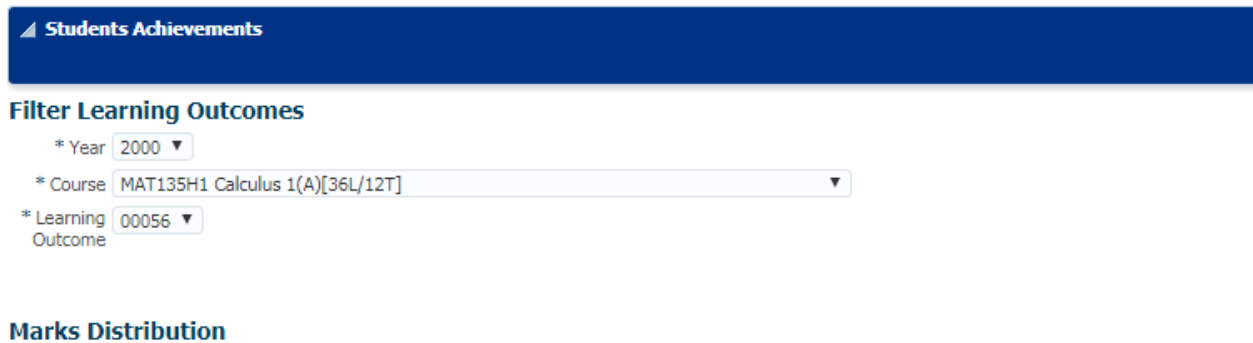
### **Results and Discussions**

## 6.1 Generated Data Statistics

Before starting the analysis of the clustering results, a general data statistic was done on the generated data to make sure that it is mimicking a real and actual learning environment.

Statistics on students' achievements were done to figure out the behavior of students' achievements in each course and each learning outcome in a semester or in the whole life of the course or learning outcome.

The general data statistics can be shown by navigating to the “Student Achievement” screen where the end user can view different statistics by choosing semester, course, learning outcome or level of learning outcome.



The screenshot displays the 'Students Achievements' section of a software interface. It features a dark blue header with the title 'Students Achievements' and a small upward-pointing triangle icon. Below the header, the section is titled 'Filter Learning Outcomes'. There are three filter fields: '\* Year' with a dropdown menu showing '2000', '\* Course' with a dropdown menu showing 'MAT135H1 Calculus 1(A)[36L/12T]', and '\* Learning Outcome' with a dropdown menu showing '00056'. Below these filters, the section is titled 'Marks Distribution'.

**Figure 6. 1: Filtering options for measuring students' achievements**

### 6.1.1 Students' achievements in a learning outcome

Figure (6.2) shows students' achievements in 2014 academic year, for “Data Structure and Analysis” course and exactly for the learning outcome # “00074”. The “screen shot” shows a column chart that reflect the number of students who took the learning outcome in that year which is equals to 34 students, the maximum mark for that learning outcome which is 95, the minimum mark of 64, the mean of students' achievements of 77.23 and the standard deviation of 9.24.

Whereas the line chart shows the normal distribution of students' achievements for the selected learning outcome all over the academic years. The screenshot also shows the number of students who took the learning outcome which is equal to 482 students, the maximum mark all over the

years in that learning outcome which is 96, the minimum mark of 60, the students' achievements mean - all over the years - in the learning outcome is 76.03 and the standard deviation is 9.09.

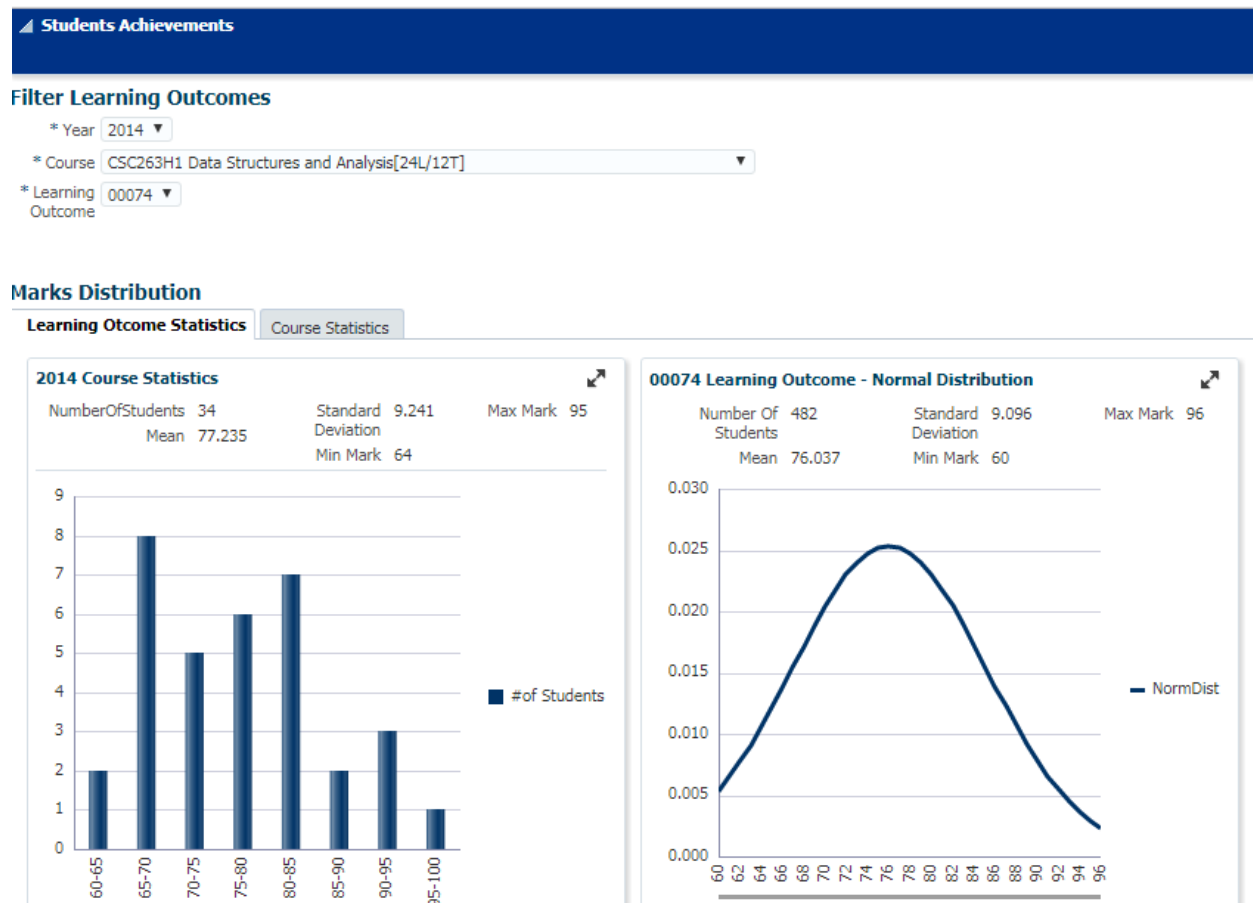


Figure 6. 2: Marks distribution for a learning outcome

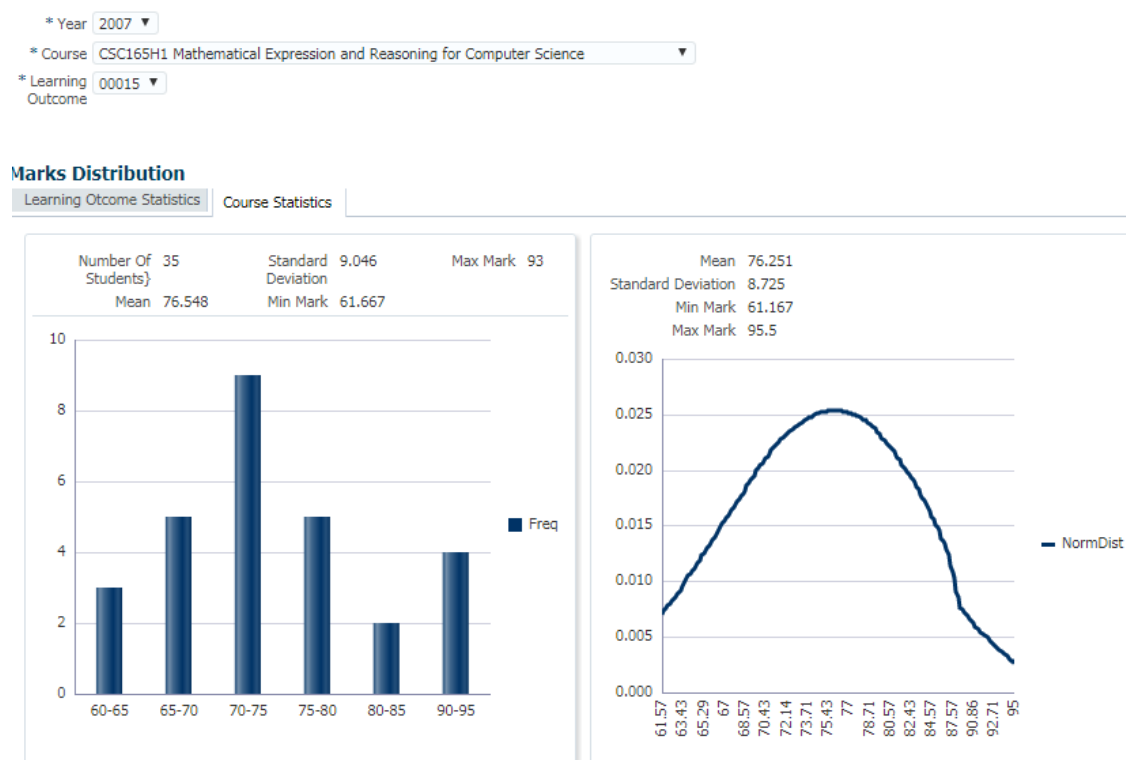
### 6.1.2 Students' achievements in a Course

In the same manner the solution enables the end user to analyze students' achievements on course level.



The following figure (6.3) shows students' achievements in the academic year 2014 for the course of “Data Structure and Analysis”. The “screen shot” shows in the column chart the number of students who took the course in that year, the maximum mark, the minimum mark, the mean of students' achievements and the standard deviation.

Whereas the line chart shows the normal distribution of students' achievements for the selected course all over the academic years. The screenshot also shows the number of students who took the learning outcome, maximum, minimum marks and other statistic all over the years.



**Figure 6. 3: Students achievements in “Mathematical Expression and Reasoning for Computer Science” in 2007**

### 6.1.3 Students' trends in levels of learning outcomes

Students' trends in the levels of learning outcome in course “CSC165H1” in different years where the numbers reflect the mean of students' achievements in the learning

outcome level in each year. By this analysis, instructors could find out their students' trends in the learning outcome levels (understand, apply, analyze and evaluate) .

## 6.2 Students' Similarity

**Table 6. 1 Students' trends in the levels of learning outcome in course**

levels	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016		
1	75.75	74.88	77.47	76.84	75.83	75.04	73.46	75.02	76.87	74.98	77.18	76.05	75.10		
2	76.16	74.92	77.69	76.13	75.28	75.30	74.12	75.25	76.83	74.46	76.99	76.14	74.68		
3	75.42	74.72	76.84	77.20	75.17	75.59	73.76	75.39	76.93	75.30	76.99	76.02	75.22		
4	75.25	76.00	77.74	76.74	75.22	75.04	73.44	74.50	76.54	75.00	76.88	75.50	74.32		

In order to find out the similarity between students, K-means algorithm is used. Five different distance functions are used in order to find out the best matching results, the five distance functions are discussed in section 3.4:

1. Euclidean distance
2. Correlation distance
3. Jaccard similarity coefficient
4. Cosine similarity
5. Manhattan distance

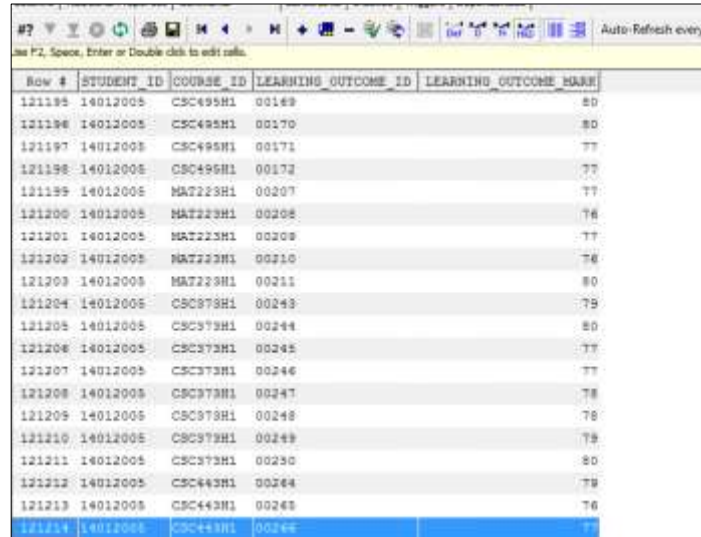
In order to find out the best “K” for the k-means method on the generated data, both methods discussed in section 3.5.2 are used:

1. Elbow method
2. The average silhouette approaches

All the above is applied on the two types of matrices discussed in section 5.7:

1. Dense matrix
2. Sparse matrix

The matrices are built from data existing mainly in the source table of students' marks “SR\_STUDENT\_LEARNING\_OUTCOMES” which contain 121,214 rows that mimic computer science students from 2000 to 2017.



Row #	STUDENT_ID	COURSE_ID	LEARNING_OUTCOME_ID	LEARNING_OUTCOME MARK
121195	14012005	CSC495M1	00169	80
121196	14012005	CSC495M1	00170	80
121197	14012005	CSC495M1	00171	77
121198	14012005	CSC495M1	00172	77
121199	14012005	MAT223M1	00207	77
121200	14012005	MAT223M1	00208	76
121201	14012005	MAT223M1	00209	77
121202	14012005	MAT223M1	00210	76
121203	14012005	MAT223M1	00211	80
121204	14012005	CSC373M1	00243	79
121205	14012005	CSC373M1	00244	80
121206	14012005	CSC373M1	00245	77
121207	14012005	CSC373M1	00246	77
121208	14012005	CSC373M1	00247	78
121209	14012005	CSC373M1	00248	78
121210	14012005	CSC373M1	00249	79
121211	14012005	CSC373M1	00250	80
121212	14012005	CSC443M1	00264	79
121213	14012005	CSC443M1	00265	76
121214	14012005	CSC443M1	00266	77

**Figure 6.1.3:1: Data in "SR\_STUDENT\_LEARNING\_OUTCOMES" table**

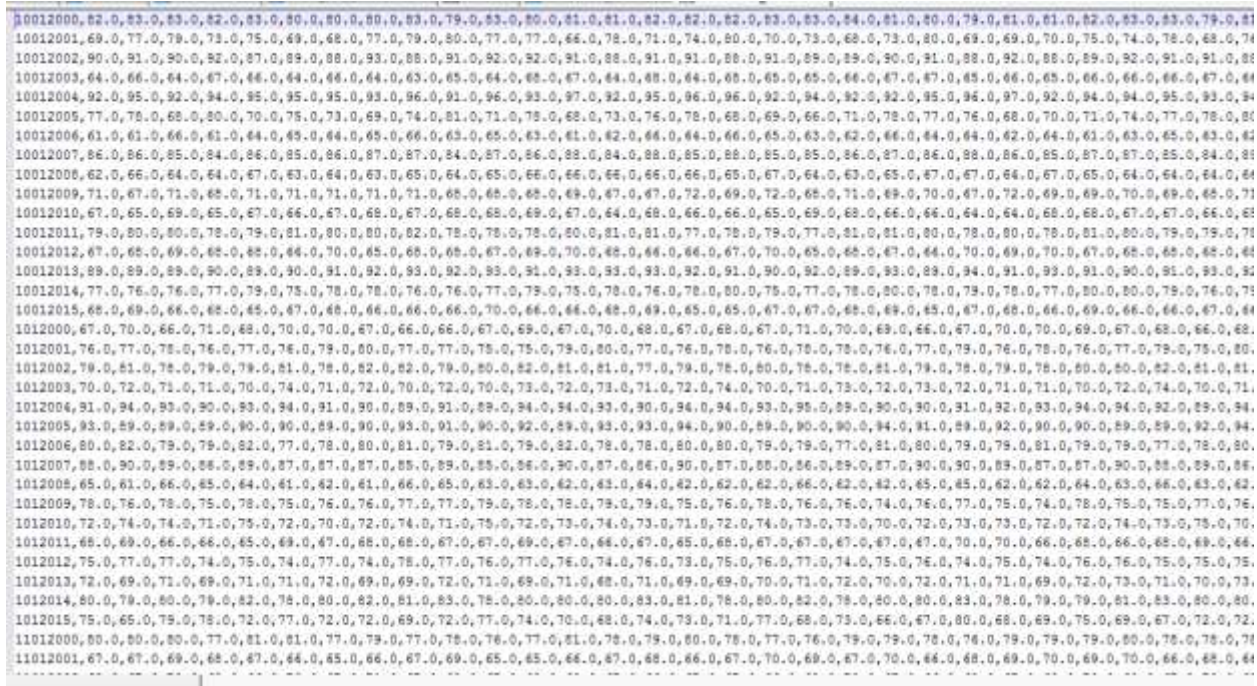
The figure above shows the source table of student achieved marks in learning outcomes “SR\_STUDENT\_LEARNING\_OUTCOMES” where the table contains the student id, the course id, learning outcome id, and the achieved mark.

## 6.2.1 Building the Matrix

As discussed in section 5.7, the matrix  $R$  of  $S \times L$  is built, where  $S$  represents students of the same specialization as the active student, whereas  $L$  represent one of the followings:

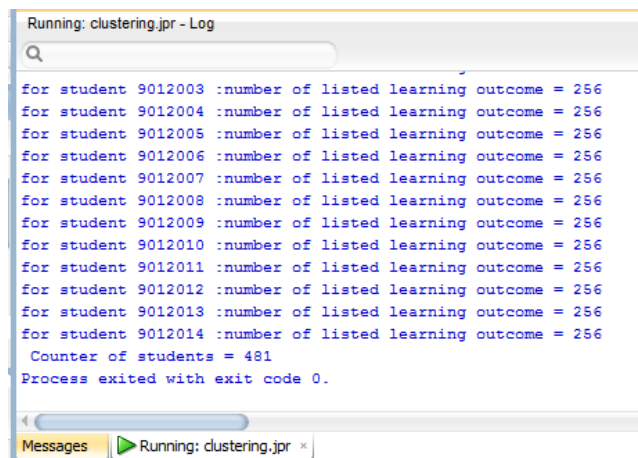
1. The list of learning outcomes achieved by the active students
2. The levels of learning outcomes achieved by the active user.

Figure (6.4) shows the matrix  $R(481 \times 256)$  for the active student “17012006” who is taking course “MAT237Y1”, the matrix shows that 481 student took the same course as the active student. The active student has achieved 256 learning outcomes, which does not mean in necessary, that each student in the same matrix had taken. Whenever a student isn’t taking a learning outcome as the active student, the achievement of the learning outcome for that student will be considered as zero.



**Figure 6. 4: Generated sparse matrix for student learning outcomes where the student’s Id is printed at the beginning**

Figure 6.5 shows the log file while generating the matrix for student “17012006”, where each student is represented by 256 learning outcomes and the total number of students are 481.



**Figure 6. 5: building Student 17012006 spars Matrix for Learning Outcomes**

When building the matrix based on the levels of learning outcomes achieved by students, the matrix  $R$  of  $S \times L$  represents students who took the active course ( $S$ ), the levels of learning

outcome for each course achieved by the active student (L), where each entry in the matrix reflects the students' average achievement in the level of learning outcome.

The matrix is built from two source tables “SR\_LEARNING\_OUTCOMES” in which the levels of learning outcomes are defined, and the previous mentioned table ”SR\_STUDENT\_LEARNING\_OUTCOMES” in which student marks are stored.

The following table shows a sample set of courses and number of learning outcomes on each learning outcomes level. Learning outcome levels are represented by their codes where level 1,2,3,4 denote Remember, Understand, Apply and Analyze.

**Table 6. 2 Learning outcomes and learning outcomes levels for each course, learning outcomes levels.**

Course Name	Learning Outcomes Level	Number of Learning Outcomes
Algorithm Design, Analysis & Complexity[36L/12T]	1	2
Algorithm Design, Analysis & Complexity[36L/12T]	2	2
Algorithm Design, Analysis & Complexity[36L/12T]	3	2
Algorithm Design, Analysis & Complexity[36L/12T]	4	2
Analysis II[72L/48T]	1	2
Analysis II[72L/48T]	2	2
Analysis II[72L/48T]	3	2
Analysis II[72L/48T]	4	1
Analysis I[72L/48T]	1	2
Analysis I[72L/48T]	2	2
Analysis I[72L/48T]	3	2
Analysis I[72L/48T]	4	1
Applied Bioinformatics[24L]	1	2
Applied Bioinformatics[24L]	2	2
Applied Bioinformatics[24L]	3	2
Applied Linear Algebra[36L/12T]	1	2
Applied Linear Algebra[36L/12T]	2	2
Applied Linear Algebra[36L/12T]	3	2
Applied Linear Algebra[36L/12T]	4	2
Compilers II[24L/36P]	1	2
Compilers II[24L/36P]	2	2
Compilers II[24L/36P]	3	2
Computability and Logic[24L/12T]	1	2
Computability and Logic[24L/12T]	2	2
Computability and Logic[24L/12T]	3	2
Computability and Logic[24L/12T]	4	1

The following table shows a sample set of students' achievements in different courses based on learning outcome levels. The highlighted rows show the average achievements of the student “10012000” in the level of learning outcome of course “CSC165H1”, where this course has a set of seven learning outcomes classified in four levels 1,2,3 and 4.

Average Achievement calculation can be found referring to reference [44]

**Table 6. 3 Students achievements in levels of learning outcomes**

STUDENT ID	COURSE ID	LEARNING OUTCOME LEVEL	Average Achievement
10012000	CSC148H1	1	82.5
10012000	CSC148H1	2	82.5
10012000	CSC148H1	3	81.5
10012000	CSC148H1	4	80
10012000	CSC165H1	1	82
10012000	CSC165H1	2	82.5
10012000	CSC165H1	3	83.5
10012000	CSC165H1	4	81
10012000	CSC207H1	1	81
10012000	CSC207H1	2	81.5
10012000	CSC207H1	3	81
10012000	CSC236H1	1	79.5
10012000	CSC236H1	2	81
10012000	CSC236H1	3	82.5
10012000	CSC240H1	1	81
10012000	CSC240H1	2	80.5
10012000	CSC240H1	3	80.5
10012000	CSC258H1	1	82
10012000	CSC258H1	2	81.5
10012000	CSC258H1	3	82
10012000	CSC263H1	1	82.5
10012000	CSC263H1	2	81
10012000	CSC263H1	3	83.5
10012000	CSC263H1	4	82.5
10012000	CSC265H1	1	81.5



Figure (6.6) shows the matrix R of  $481 \times 136$  for the student id “17012006”, where 136 represents the levels of learning outcomes in all achieved courses by the active student. Figure 23 shows the log file when generating the matrix where the log shows that each student is represented by 136 learning levels.

```

455 0012004,77.5,77.5,78.5,80.0,77.5,79.0,77.0,77.0,78.0,77.5,79.0,76.5,78.0,78.5,78.0,0.0,0.0,0.0,78.0,77.0,7
456 0012005,75.0,76.0,74.5,74.0,74.5,75.0,77.0,75.0,74.5,73.5,73.0,74.5,75.0,77.5,75.0,76.5,74.0,75.5,76.0,75.
457 0012006,69.5,64.0,66.0,66.0,66.0,68.5,88.5,86.0,85.5,68.0,68.0,65.5,67.5,68.0,66.0,66.0,67.5,66.5,66.5,67.
458 0012007,94.5,94.0,93.0,91.0,94.5,92.5,94.0,91.0,94.5,92.0,96.0,95.5,94.5,96.0,95.0,0.0,0.0,0.0,94.0,93.5,9
459 0012008,73.0,72.0,74.5,71.0,73.0,73.5,72.0,73.0,73.0,73.0,73.0,75.0,72.5,72.0,74.0,71.0,78.5,74.0,72.0,71.0,73.
460 0012009,63.5,64.5,64.0,67.0,62.5,66.0,65.0,64.0,66.0,69.0,63.0,66.0,69.0,65.5,67.0,64.0,64.0,64.0,67.0,64.
461 0012010,78.5,80.5,81.0,79.0,81.0,80.0,79.5,78.0,78.5,79.0,80.0,80.0,81.5,79.5,79.0,81.0,80.0,78.0,79.0,78.
462 0012011,91.5,94.0,94.0,93.0,92.5,92.5,95.0,92.0,93.0,93.5,94.0,94.0,94.0,95.5,93.0,0.0,0.0,0.0,95.0,94.0,9
463 0012012,72.5,71.5,72.5,73.0,70.5,72.5,71.0,74.0,69.5,71.5,70.0,71.5,73.5,71.5,73.0,71.0,72.0,72.5,70.5,70.
464 0012013,67.5,69.0,65.5,65.0,69.5,66.0,68.5,69.0,69.5,66.5,66.0,66.5,66.5,69.5,65.0,69.0,65.5,69.0,60.0,69.
465 0012014,82.5,81.0,84.0,91.0,82.0,94.0,92.5,91.0,83.5,83.0,91.0,81.0,82.5,93.0,95.0,0.0,0.0,0.0,94.0,95.0,9
466 0012015,61.5,62.5,62.0,63.0,62.5,61.5,61.0,63.0,61.0,64.5,64.0,63.0,63.0,61.5,64.0,0.0,0.0,0.0,61.5,63.0,6
467 0012016,71.0,71.5,70.0,72.0,71.0,72.0,72.0,74.0,71.5,72.0,70.0,71.0,72.0,71.5,73.0,0.0,0.0,0.0,69.5,73.5,7
468 0012017,64.0,66.5,66.5,66.0,66.5,64.5,64.5,64.0,64.5,64.0,63.0,64.0,67.0,65.0,64.0,64.0,63.5,64.0,63.5,63.
469 0012018,73.0,74.5,76.0,77.0,73.5,74.0,74.0,73.0,75.5,75.5,75.0,75.5,75.5,75.5,77.0,0.0,0.0,0.0,76.0,74.5,7
470 0012019,79.0,77.0,78.0,76.0,77.0,79.0,77.5,80.0,78.0,80.5,77.0,78.5,77.5,80.5,76.0,80.0,78.5,78.0,79.5,78.
471 0012020,82.5,81.0,83.5,83.0,83.5,82.0,80.5,83.0,82.0,83.0,85.0,83.5,83.5,83.5,83.5,82.0,82.0,82.0,82.5,84.0,83.
472 0012021,92.0,90.5,90.5,94.0,94.0,92.5,93.5,95.0,94.5,93.5,95.0,92.5,93.5,94.5,92.0,91.5,95.5,93.0,93.5,93.
473 0012022,71.0,73.5,74.0,80.0,71.0,75.5,88.0,71.0,74.5,78.5,71.0,74.0,71.0,76.0,69.0,0.0,0.0,0.0,71.0,70.0,7
474 0012023,80.5,82.0,82.0,83.0,82.0,80.0,82.5,79.0,80.5,81.5,81.0,81.0,81.5,81.0,82.0,0.0,0.0,0.0,80.0,82.5,8
475 0012024,64.5,64.0,65.5,66.0,65.5,65.0,65.5,64.0,65.5,65.5,65.0,64.0,63.5,67.5,66.0,66.0,66.5,66.5,66.0,64.
476 0012025,90.5,94.0,93.0,91.0,91.5,95.5,91.5,92.0,95.0,93.5,92.0,91.5,94.5,93.5,91.0,91.5,93.0,95.0,92.5,92.
477 0012026,77.0,77.5,77.5,77.0,77.0,75.0,76.0,74.0,76.0,75.5,74.0,77.0,76.5,75.5,75.0,0.0,0.0,0.0,75.5,74.0,7
478 0012027,70.5,72.0,74.5,75.0,72.5,71.5,70.5,74.0,72.0,71.5,71.0,71.0,70.5,75.0,75.0,75.0,73.0,73.0,71.5,71.5,73.
479 0012028,83.0,84.5,83.5,84.0,85.0,83.0,84.0,84.0,82.0,83.0,84.0,83.5,83.0,85.5,83.0,82.5,83.5,85.5,82.5,84.
480 0012029,81.5,80.5,80.0,79.0,81.0,80.0,79.5,79.0,81.0,80.5,78.0,79.0,79.0,80.0,0.0,0.0,0.0,80.5,79.0,8

```

Figure 6. 6: The sparse matrix of learning outcomes levels for student 17012006

```

Running: clustering.jpr - Log
for student 9012003 :number of listed learning outcome = 136
for student 9012004 :number of listed learning outcome = 136
for student 9012005 :number of listed learning outcome = 136
for student 9012006 :number of listed learning outcome = 136
for student 9012007 :number of listed learning outcome = 136
for student 9012008 :number of listed learning outcome = 136
for student 9012009 :number of listed learning outcome = 136
for student 9012010 :number of listed learning outcome = 136
for student 9012011 :number of listed learning outcome = 136
for student 9012012 :number of listed learning outcome = 136
for student 9012013 :number of listed learning outcome = 136
for student 9012014 :number of listed learning outcome = 136
Counter of students = 481
Process exited with exit code 0.

```

Figure 6. 7: Log file while generating the sparse matrix for student's learning outcomes levels

When building the dense matrix for the same student on the same course for the achievements of the learning outcomes, the result shows a matrix of  $0 \times 255$ .

Which means that no student in the 17 years of the university took the exact number of 255 learning outcomes. The same was found when generating the dense matrix for the same student on the levels of learning outcomes.

This result of sparse matrix will be better for junior students as almost all achieved learning outcomes are related to mandatory courses which all students took as these students still didn't start with the elective courses.

### **Conclusion:**

1. Whereas there exists an opportunity in the sparse matrix to find similar students to the active ones among a set of students who took the same courses, this opportunity is almost lost when choosing the matrix to be dense and the probability of not finding any student arise when talking about senior students.
2. We can find out that as much as the student took learning objects, the opportunity to find out student who took the same set of learning materials will decrease.

## **6.2.2 Choosing the number of clusters “K”**

In this section the following methods will be applied in order to find out the best number of clusters. The matrix is chosen to be for the junior student “15012016” –as an example- who has achieved 72 learning outcomes, so the sparse matrix R for learning outcomes is of  $481 \times 72$  and the dense matrix R is also  $481 \times 72$  where 481 is the number of students who took the active course “STA257H1”. The sparse matrix R for levels of learning outcomes is  $481 \times 40$  and the dense matrix R is also of  $481 \times 40$  where 40 is the total number of levels for the 72 achieved learning outcomes.

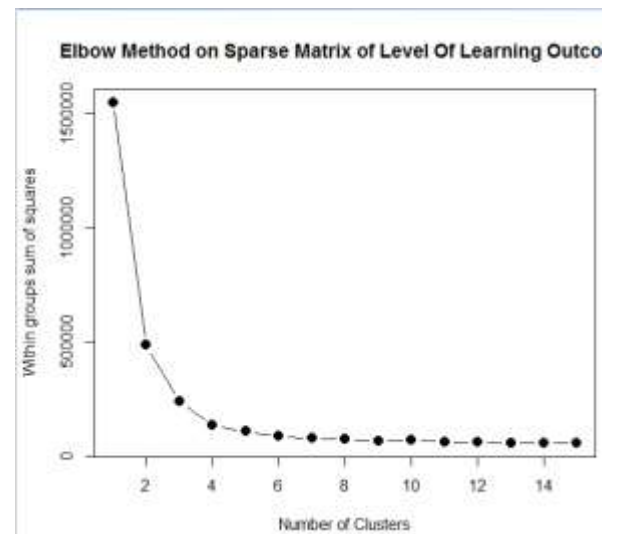
As mentioned, here are the listed methods:

1. Applying Elbow method on sparse matrix
2. Applying Elbow method on dense matrix
3. Applying average silhouette approach on sparse matrix
4. Applying average silhouette approach on dense matrix

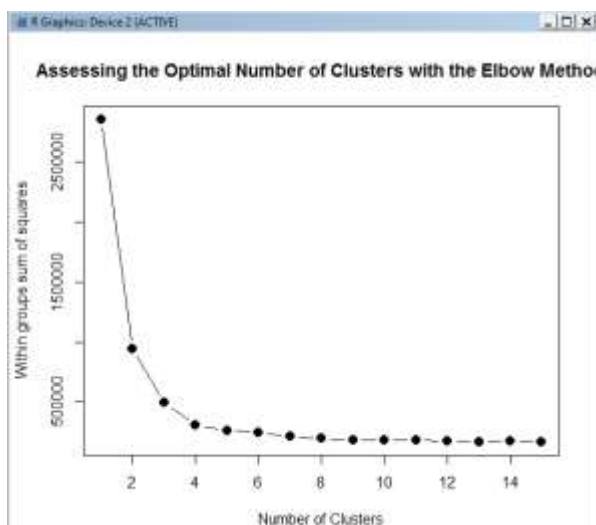


## Applying Elbow method on sparse matrix (Learning outcomes and levels of learning outcomes)

The following figures (6.8) and (6.9) show the results of applying the Elbow method on two sparse matrices: (1) students' achievements in learning outcome "LO" and (2) students' achievements in the levels of learning outcomes. Where the best number of clusters "K" is 3.



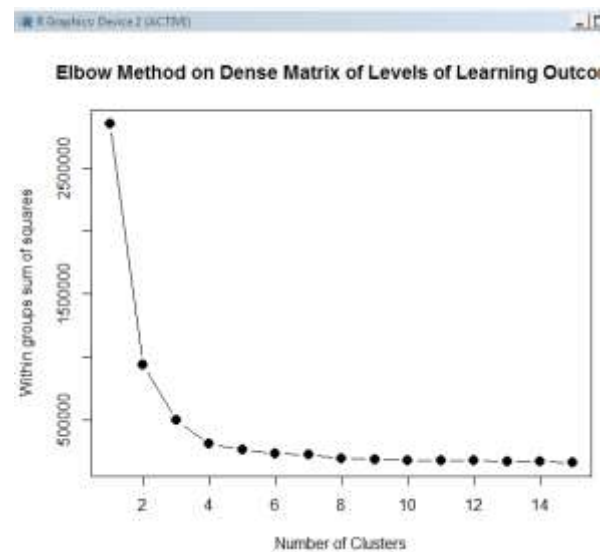
**Figure 6. 8: Elbow method for sparse matrix of learning outcome Levels**



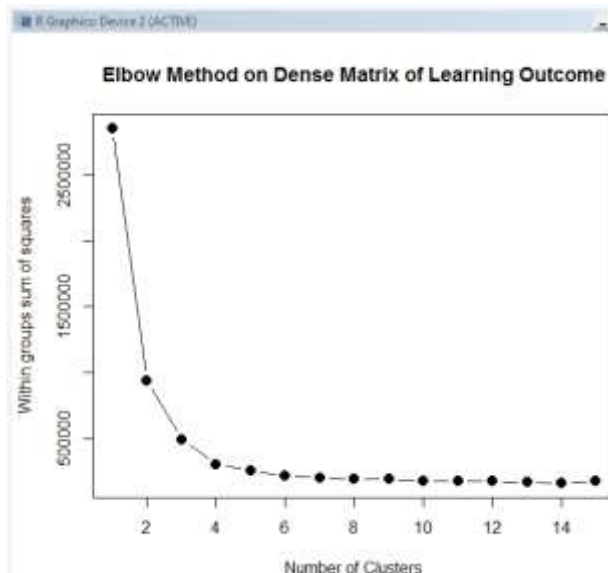
**Figure 6. 9: Elbow method for sparse matrix of learning outcome**

## Applying Elbow method on dense matrix (Learning outcomes and levels of learning outcomes)

In the following figures Elbow method is also applied on two dense matrices of students' achievements in learning outcome "LO" and students' achievements in the levels of learning outcomes. Where the best number of clusters "K" is also 3.



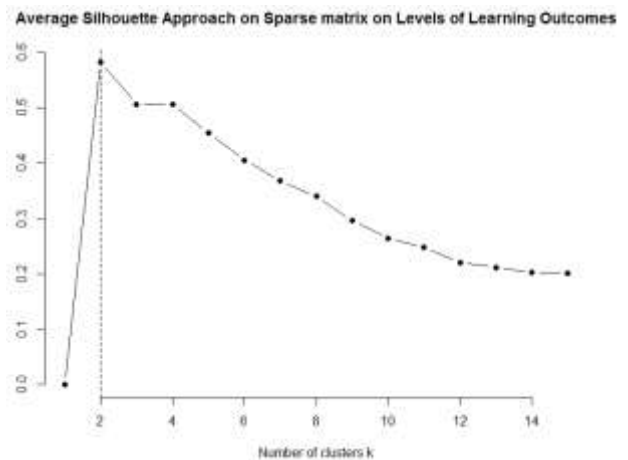
**Figure 6. 10: Elbow method for dense matrix of levels of learning outcome**



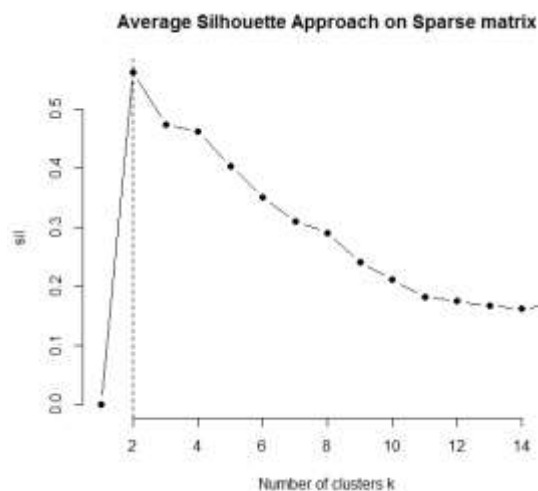
**Figure 6. 11: Elbow method for dense matrix of learning outcome**

### Applying average Silhouette approach on sparse matrix (Learning outcomes and levels of learning outcomes)

When applying the Silhouette approach on the same mentioned sparse matrix, the best number of clusters appears to be 2 as shown in the figures below.



**Figure 6. 12: Silhouette approach on sparse matrix of learning outcomes Levels**

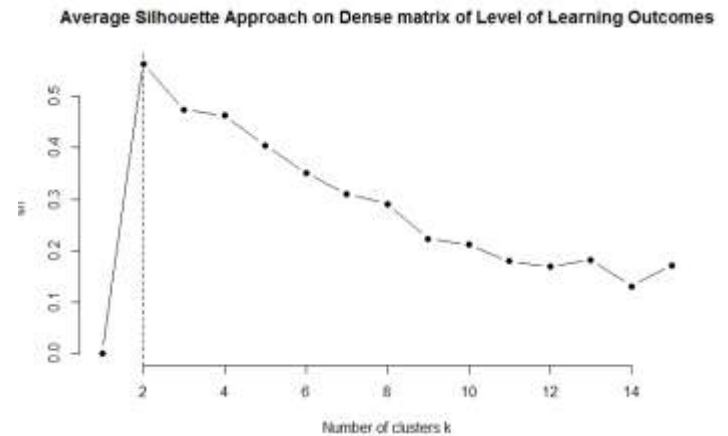


**Figure 6. 13: Silhouette approach on sparse matrix of learning outcomes**

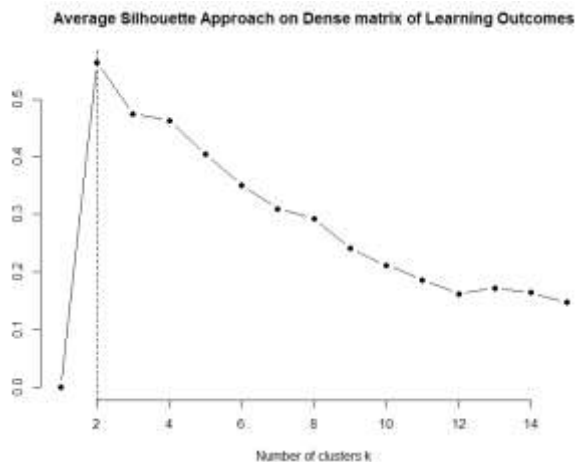
### Applying average Silhouette approach on dense matrix (Learning outcomes and levels of learning outcomes)

The same number of best clusters appears when applying the Silhouette approach on the two mentioned dense matrices, where the best number of clusters is equal to two as shown in the figures below.

Because the two methods Elbow and average Silhouette approach give different results for the optimal number of clusters, the gap-statistic method is also applied on sparse matrix to find out the optimal number of clusters.



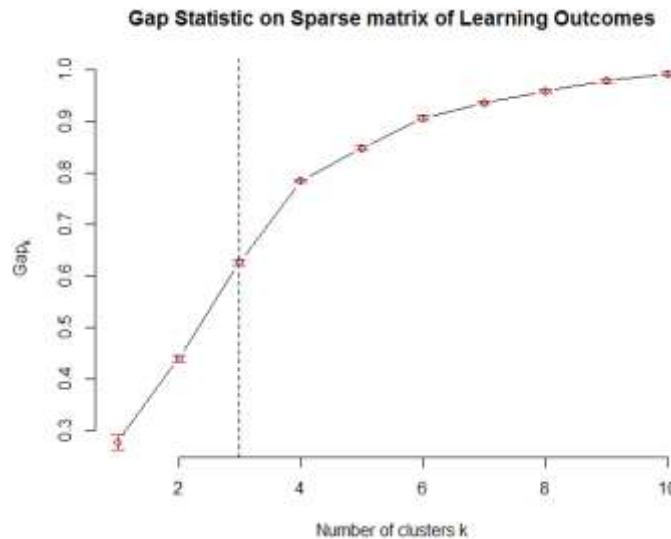
**Figure 6. 14:Applying average Silhouette approach on dense matrix of levels of learning outcomes**



**Figure 6. 15:Applying average Silhouette approach on dense matrix of learning outcomes**

### Applying Gap-statistic method on sparse matrix

When applying gap statistic method on the two-mentioned sparse matrix, the best number of clusters appears to be three as shown in the figure below.



**Figure 6. 16: Gap statistic method on sparse matrix of learning outcomes**

### Conclusion:

- Three cluster solutions are suggested as the optimal number of clusters when using the elbow method on both sparse and dense matrices on learning outcomes and levels of learning outcomes achievements.
- The average silhouette method gives two clusters as the optimal number of clusters on both sparse and dense matrices on learning outcomes and levels of learning outcomes achievements.
- Also, three clusters solutions are suggested using gap-statistic method on sparse matrix of learning outcomes.

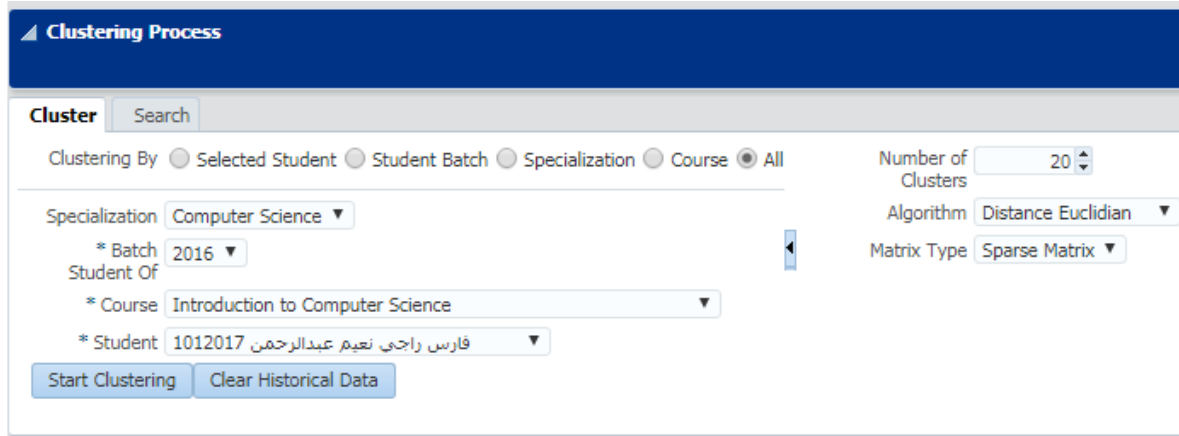
According to these observations, it's possible to define  $k = 3$  as the optimal number of clusters in the data.

### 6.2.3 Applying k-means method using five different distance methods

In this part the k-means clustering method will be applied on 25 computer science senior students who registered in 2014 for two courses using five different distance methods: Correlation, Similarity Coefficient, Cosine, Euclidean and Manhattan distance. The sparse matrix option and the optimal number of  $k = 3$  will be used.

The result was 5 batches of clusters process, each one includes 50 result sets (25 for each course). From this result set, the best distance method for k-means clustering in a learning environment could be found.

To apply the clustering process, an interface is built, clustering process could be chosen to be executed for one student or batch of active students depending on the registration year, also other options are given such as clustering based on specialization or selected course. The designed interface also enables the end user to insert the number of clusters “k” and choose the distance method and matrix type.



The screenshot shows a web interface titled "Clustering Process". It features a "Cluster" tab and a "Search" button. Below these, there are radio buttons for "Clustering By": "Selected Student", "Student Batch", "Specialization", "Course", and "All" (which is selected). To the right of these is a "Number of Clusters" input field set to 20. Below the radio buttons, there are dropdown menus for "Specialization" (set to "Computer Science"), "\* Batch" (set to "2016"), "\* Student Of" (set to "Introduction to Computer Science"), and "\* Student" (set to "1012017 فارس راجي نعيم عبدالرحمن"). To the right of these dropdowns is an "Algorithm" dropdown set to "Distance Euclidian" and a "Matrix Type" dropdown set to "Sparse Matrix". At the bottom, there are two buttons: "Start Clustering" and "Clear Historical Data".

**Figure 6. 17: Parameters for clustering process**

After applying the cluster process based on the selected criteria, the result of similarity was displayed as shown in the screenshot bellow, the end user could select any student of the 25 sets to find out all his similarities among the 17 years of the university history.

Cluster									
Search									
Clustering By: <input type="radio"/> Selected Student <input checked="" type="radio"/> Student Batch <input type="radio"/> Specialization <input type="radio"/> Course <input type="radio"/> All									
Number of Clusters: 3									
Specialization: Computer Science									
Batch: 2014									
Student Of:									
Start Clustering Clear Historical Data									
Algorithm: Distance Cosine									
Matrix Type: Sparse Matrix									
View: [Icon] [Icon] [Icon] [Icon] [Icon] [Icon] [Icon] [Icon] [Icon] [Icon]									
Similar Student	BatchId	StudentId	CourseId	DistanceFunction	MatrixType	MsTime	Sse	MaxMemory	K
[Icon]	341	نا صلاح محمد شويخ		cosine	Sparse Matrix	1924	0.314	425.75756072998	3
[Icon]	341	نا صلاح محمد شويخ		cosine	Sparse Matrix	1400	0.314	425.75756072998	3
[Icon]	341	كفا علي بنسافر ربحي		cosine	Sparse Matrix	327	0.301	425.75756072998	3
[Icon]	341	كفا علي بنسافر ربحي		cosine	Sparse Matrix	1102	0.294	425.75756072998	3
[Icon]	341	نا صلاح محمد شويخ		cosine	Sparse Matrix	372	0.263	425.75756072998	3
[Icon]	341	نا صلاح محمد شويخ		cosine	Sparse Matrix	373	0.256	425.75756072998	3



Figure 6. 18: Similarity Results based on clustering process

Let's now navigate to the clustering statistic page, where the performance of each distance function will be shown. The following dashboard displays general statistics for the average SSE, average processing time, average memory usage and the actual number of clusters to the required one for the five distance functions.



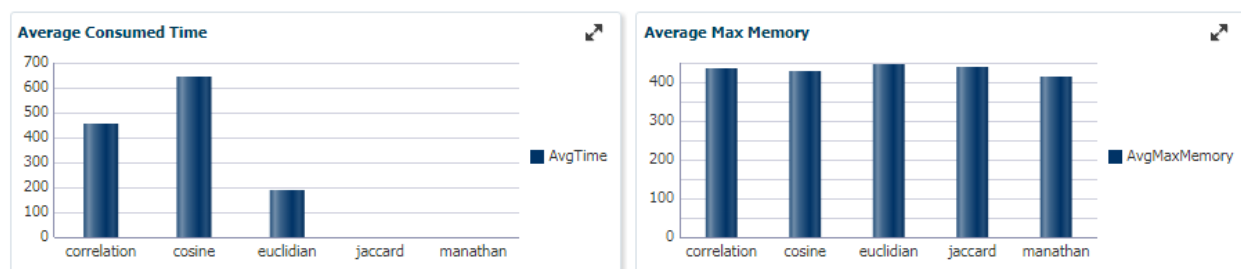
Figure 6. 19: Statistics dashboard for clustering process

From the figure below, the cosine distance records have the less amount of SSE of 0.277 whereas SSE in Jaccard distance is zero with only one cluster as a result so its SSE will be ignored. The best distance function after cosine distance is the correlation, as its average SSE is 15.618. The average maximum memory usage for all distance functions are almost the same but the average processing time for the cosine distance is the maximum with 646.4 milliseconds.

General Statistics					
View   Detach					
DistanceFunction	AvgK	AvgSse	AvgMaxMemory	AvgTime	AvgActualClusterNo
correlation	3	15.618	434.396	453.92	3
cosine	3	0.277	426.796	646.4	3
euclidian	3	37736234.923	446.726	191.24	3
jaccard	3	0	440.607	1.36	1
manathan	3	4542422247.456	415.673	1.26	3

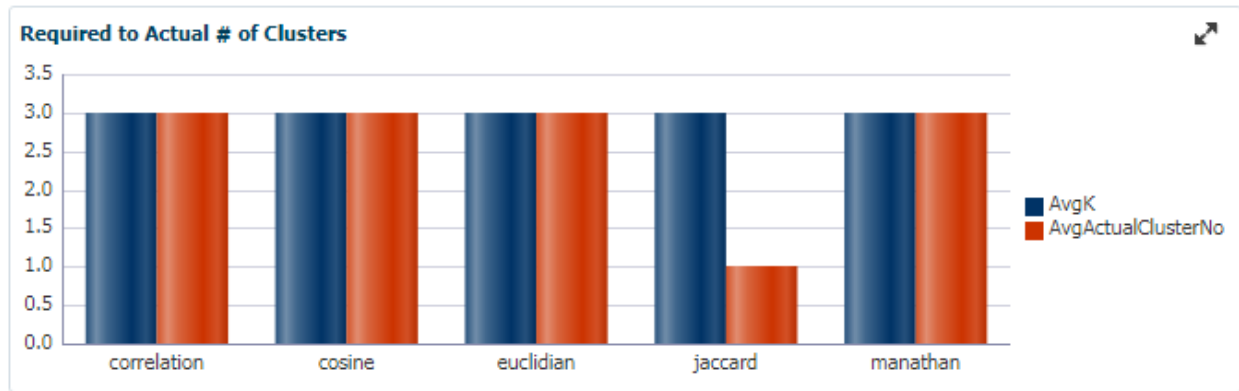
**Figure 6. 20: General statistics on different distance function**

The following figure reflects some other indicators such as average consumed time, average maximum memory usage and actual required number of clusters.



**Figure 6. 21: Average time and memory usage indicators**





**Figure 6. 22: Required to actual number of cluster indicators**

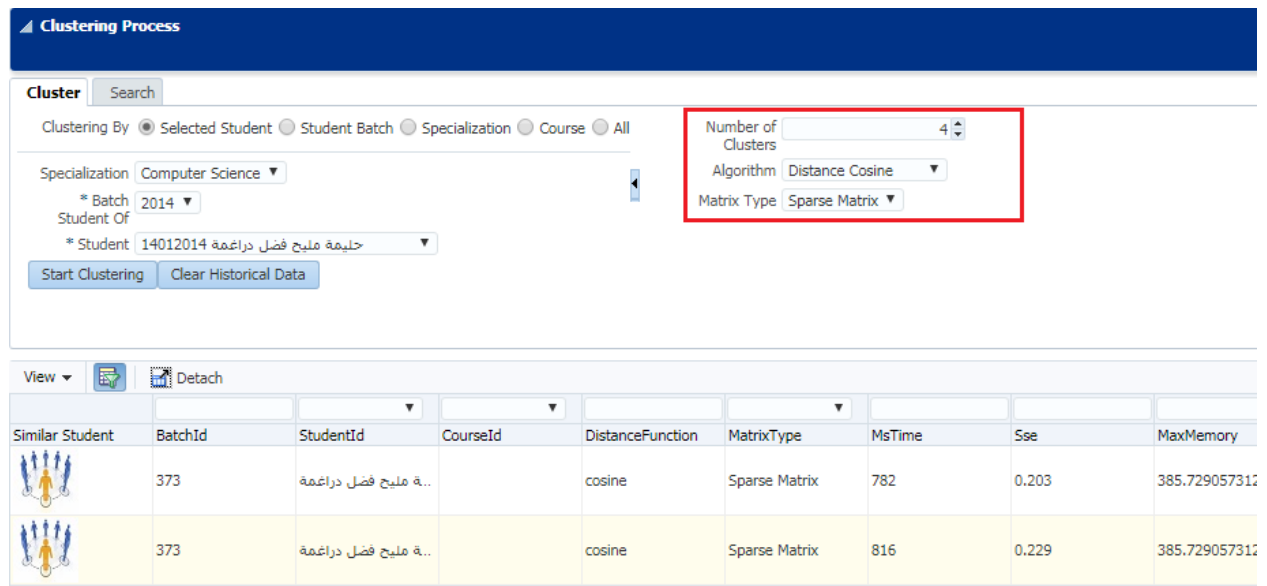
### Conclusion:

- Cosine distance has the lowest SSE of 0.277, followed by the correlation distance of 15.618.
- Manathan distance could not be used, as the result clusters is always one regardless of the requested “K”.
- Cosine distance has the largest –but still reasonable- average processing time of 646 milliseconds followed by correlation distance of 453.
- Cosine distance has the lowest average memory usage –after ignoring Manthan distance.

According to these observations, it’s possible to define Cosine distance as the best distance function for k-means clustering in this data.

### 6.2.4 Similarity Results

Once applying the clustering process, the engine displays all active student in each active course and shows: the distance function used, processing time, SSE, maximum memory usage and actual number of cluster results. Similar student details are shown once clicking on the similar student image link in the left column in figure (6.23).



**Figure 6. 23: Clustering process page showing the results of clustering operation**

The following screen shots show samples of the similarity among students when applying k-means clustering using cosine distance on sparse matrix. Taking Haleema as an example of an active student who the engine is looking for a similar student as her, the engine shows some of the students– who appear in the same cluster as Haleema’s - some of those students are shown in figure (6.24), (6.25) and (6.26) (Abdel Raheem, Ayoub and Hussein).

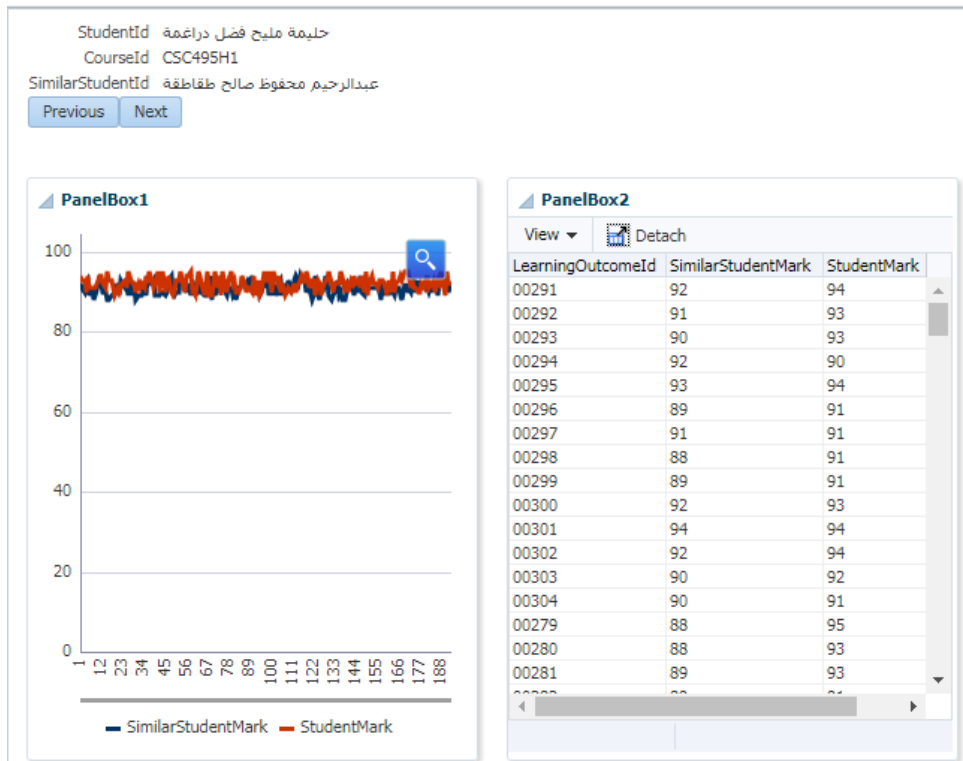


Figure 6. 24: Abdelrahman and Haleema marks in course “CSC495H1”

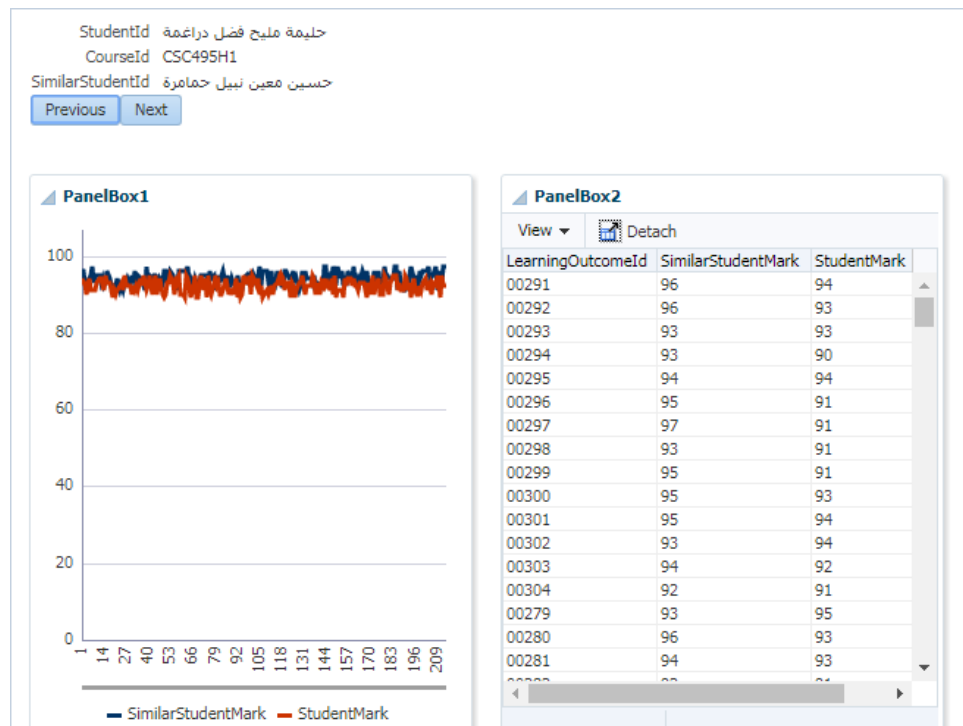
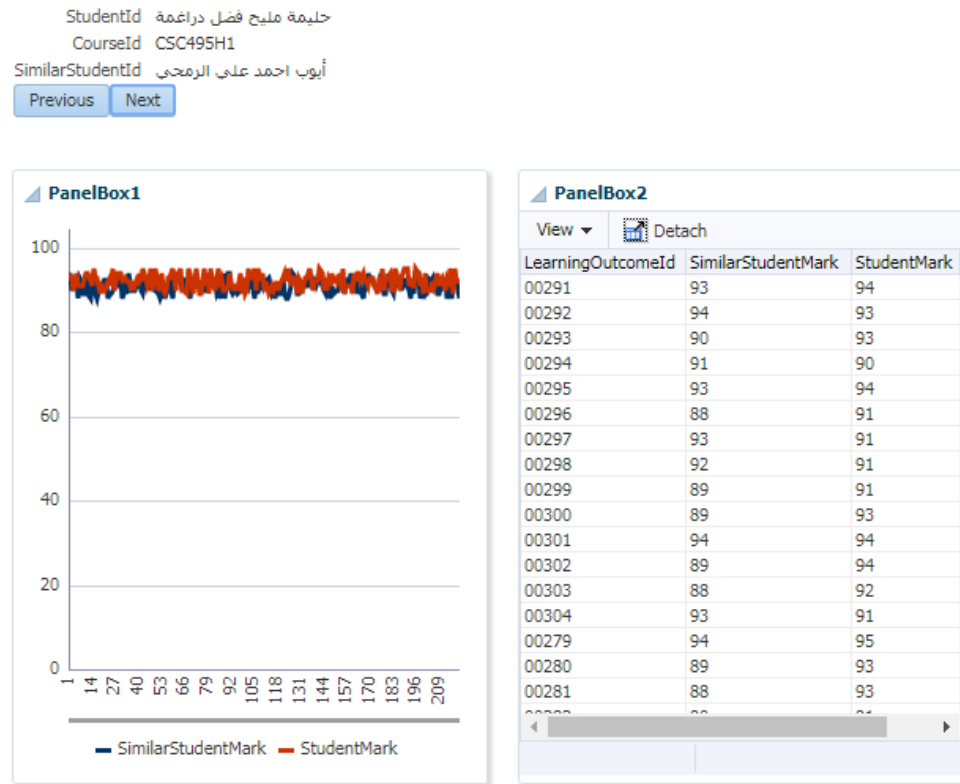


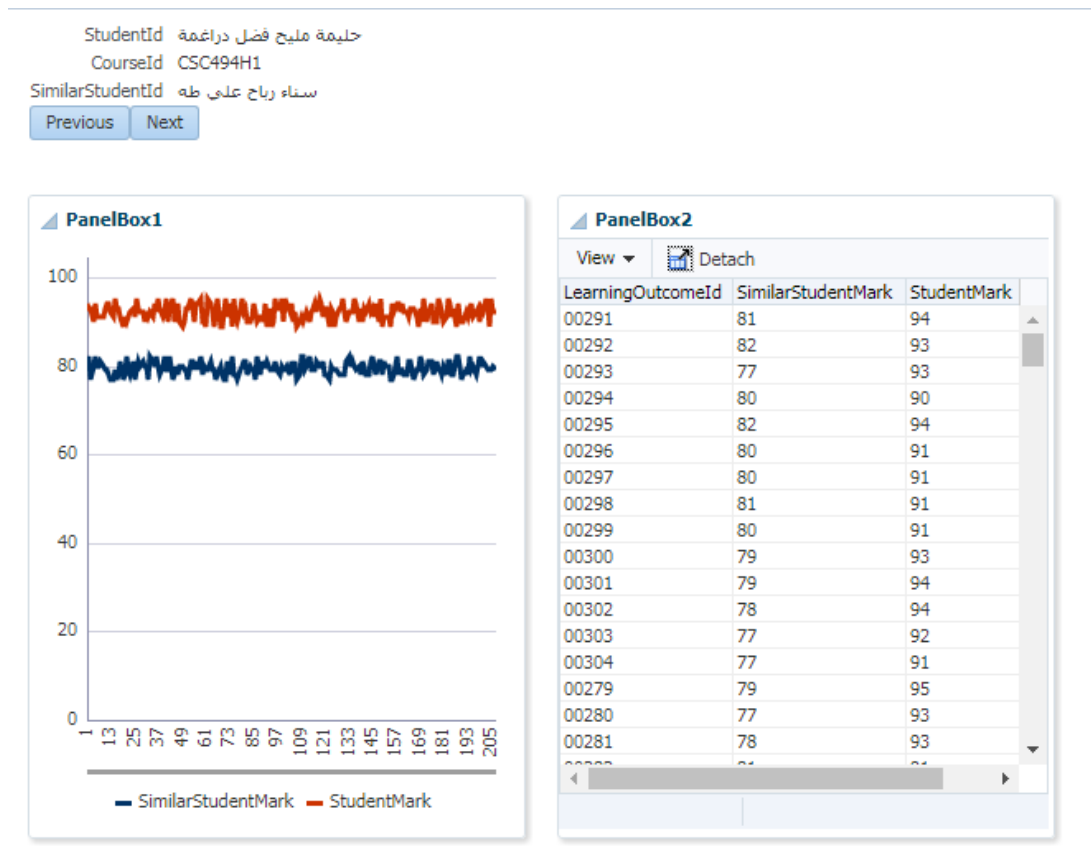
Figure 6. 25: a Hussein and Haleema marks in course “CSC495H1”



**Figure 6. 26: Ayoub and Haleema marks in course “CSC495H1”**

Figures above shows a close achievement between Haleema and Ayoub, Hussain and Abdelrahman who appear in the same cluster. Whereas figure (6.27) shows that the student Sana appears to be in the same cluster as Haleema even though her achievement appears to be far away from Haleema’s. This case led to the following questions:

1. Why does this type of students appear in the clustering result?
2. What is the percentage of this type of students out of the total result?
3. How to enhance the clustering results to fulfill the similarity target in learning environment?



**Figure 6. 27: Sana and Haleema marks in course “CSC495H1”**

To answer the questions above, the average difference “AD” between the active student and each similar student is computed, where “AD” is equal to the absolute value of the active student average in his accomplished learning outcomes minus the similar student average in only the shared learning outcome with active student, as they are only taken into consideration when building the matrix. “AD” gap is compared when increasing the number of clusters using cosine and correlation distance in k-means clustering

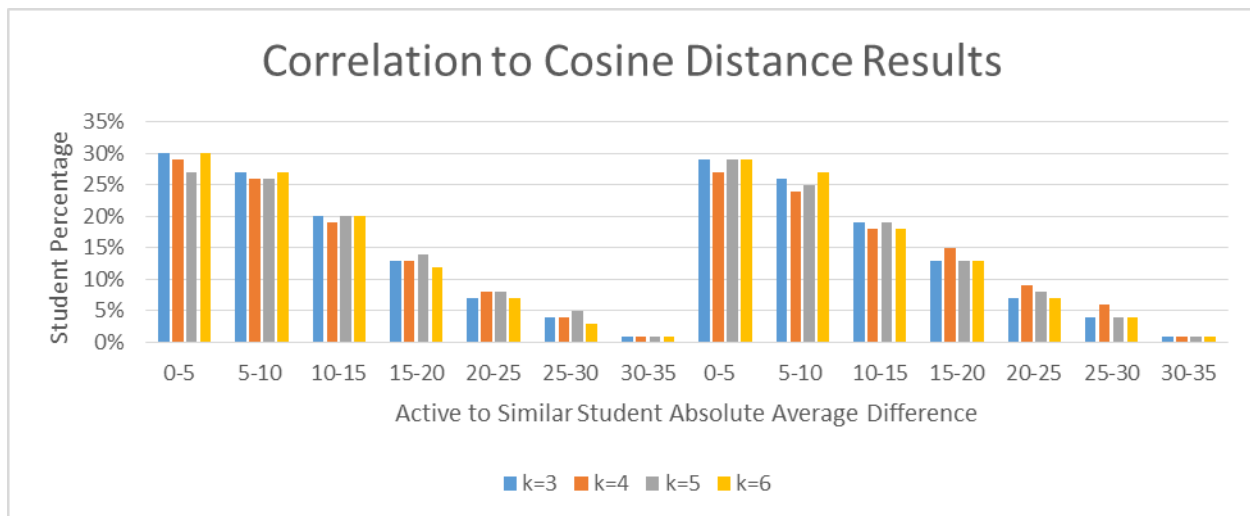
In order to do the above analysis, the clustering process was repeated four times, in each time the number of clusters is increasing by one for both correlation and cosine distance for sparse matrix. The following pivot table and bar chart summarize the result.

**Table 6. 4 Active to Similar Student Absolute Average Difference**

Distance Function	Active to Similar Student Absolute Average Difference	Number of clusters k			
		k=3	k=4	k=5	k=6
Correlation	0-5	30%	29%	27%	30%
	5-10	27%	26%	26%	27%
	10-15	20%	19%	20%	20%
	15-20	13%	13%	14%	12%
	20-25	7%	8%	8%	7%
	25-30	4%	4%	5%	3%
	30-35	1%	1%	1%	1%
cosine	0-5	29%	27%	29%	29%
	5-10	26%	24%	25%	27%
	10-15	19%	18%	19%	18%
	15-20	13%	15%	13%	13%
	20-25	7%	9%	8%	7%
	25-30	4%	6%	4%	4%
	30-35	1%	1%	1%	1%

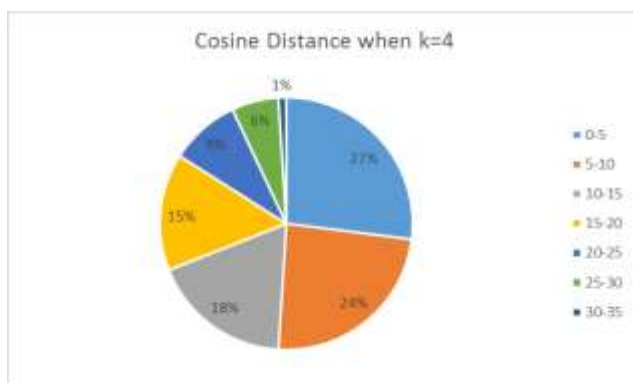
The pivot table above classifies “AD” into 7 slices labeled with “Active to Similar Student Absolute Average Difference”: 0-5, 5-10, 10-15, 15-20, 20-25, 25-30, 30-35, columns “k=3”, “k=4”, “k=5”, “k=6”, reflect the percentage of students in each slice when the number of clusters is equal to 3,4,5 and 6 for both correlation and cosine functions.

The following bar chart reflects the results of the above pivot table where the left set of bars represents the results of k-means using correlation distance whereas the right set of bars represent the results of k-means using cosine distance.

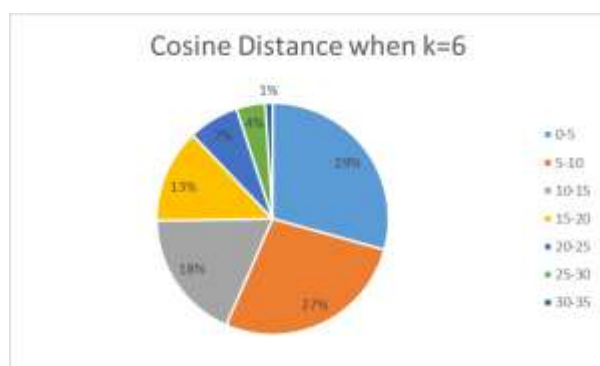


**Figure 6. 28: Active to similar students' absolute average using correlation vs cosine distance**

The following pie charts compare the k-means clustering results using cosine distance when k=4 and k=6 for k-means using cosine distance.



**Figure 6. 30: Active to similar students' absolute average using cosine distance when k=4**



**Figure 6. 29: Active to similar students' absolute average using cosine distance when k=6**

The results appear to be better when increasing the number of clusters to k=6 where the differences between students averages in both slices 0-5 and 5-10 is 56%, whereas it is equal to 51% when k=4.

The next question is: does the output of clusters differ according to the student level? In other words, Does the results of clustering differ for junior students comparing them with senior students?

According to junior students, almost all achieved learning outcomes are related to mandatory courses who almost all students took, whereas when speaking on senior students, elective courses appear which differ from one student to another so the matrix will be more sparsely.

Let's go into deeper analysis and figure out the results of applying one batch on junior students "2016" and then make the same analysis on senior students "2014".

## Junior Students Analysis

When the k-means clustering process was applied using cosine distance of 4 clusters on sparse matrix on junior students for all active courses, the results was 57,164 records in the similar student staging table, the following screen shows the output of the process.

Clustering Process

Cluster Search

Clustering By
☐ Selected Student
☒ Student Batch
☐ Specialization
☐ Course
☐ All

Specialization Computer Science

Batch 2016

Student Of

Start Clustering
Clear Historical Data

Number of Clusters 4

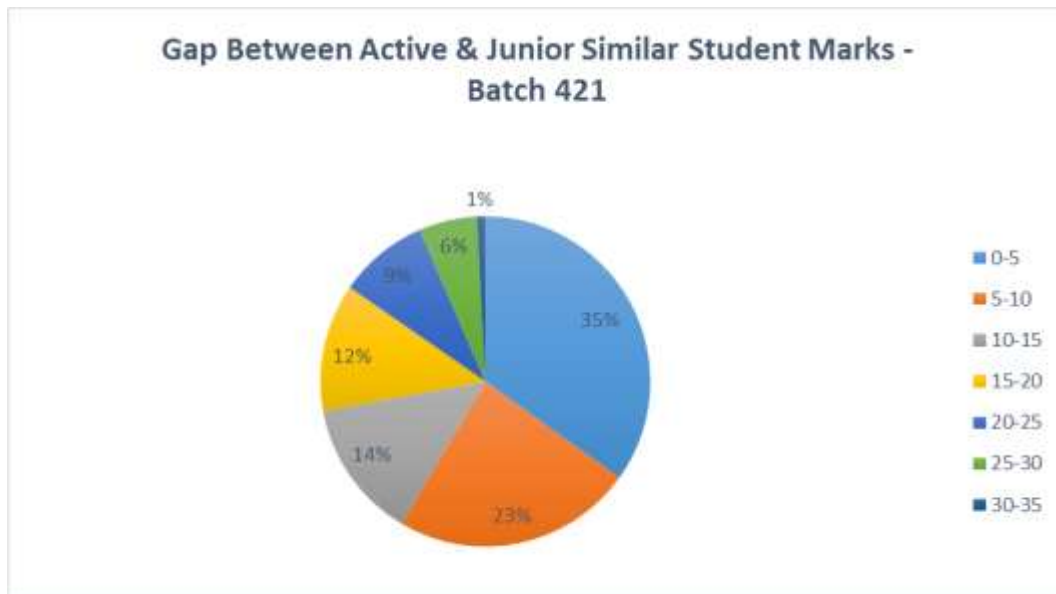
Algorithm Distance Cosine

Matrix Type Sparse Matrix

Similar Student	BatchId	StudentId	CourseId	DistanceFunction	MatrixType	MsTime	Sse	MaxMemory
	421	احمد فايز داود ابوغيث		cosine	Sparse Matrix	97	0	396.0587692260742
	421	احمد فايز داود ابوغيث		cosine	Sparse Matrix	136	0	396.0587692260742
	421	احمد فايز داود ابوغيث		cosine	Sparse Matrix	220	0	396.0587692260742
	421	احمد فايز داود ابوغيث		cosine	Sparse Matrix	307	0	396.0587692260742
	421	سجود عمر فايز بطاط		cosine	Sparse Matrix	96	0	396.0587692260742
	421	سجود عمر فايز بطاط		cosine	Sparse Matrix	372	0	396.0587692260742

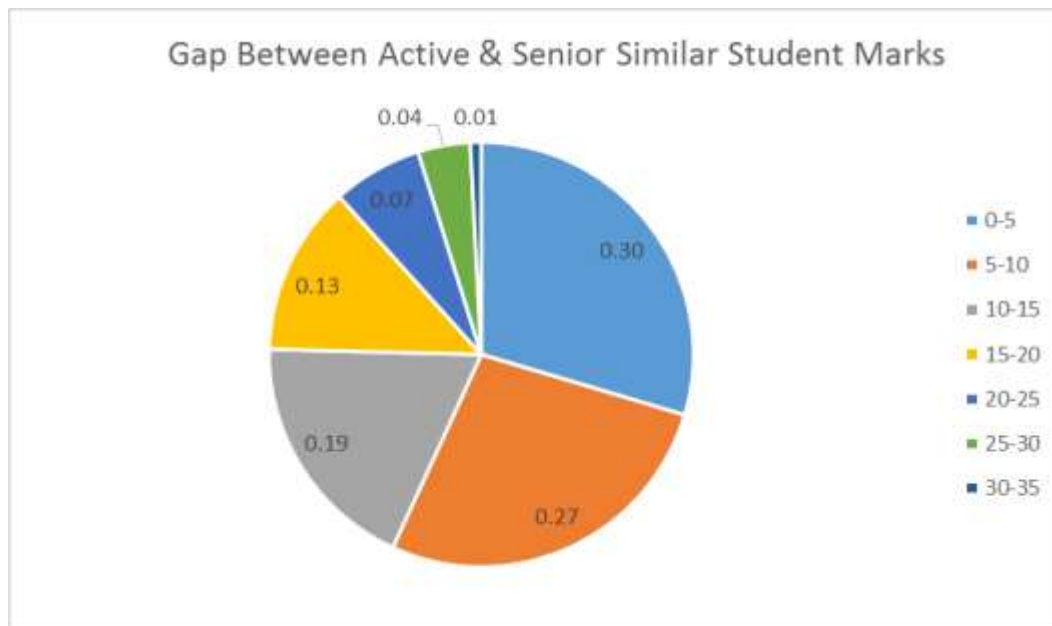


When looking at batch# 421 results, “similar students average gap” analysis shows 35% of students having a gap average between 0-5 marks far from the active student, 23% having 5-10 gap, 14% having 10-15 and 28% more than 15-mark gap as shown in figure (6.31).

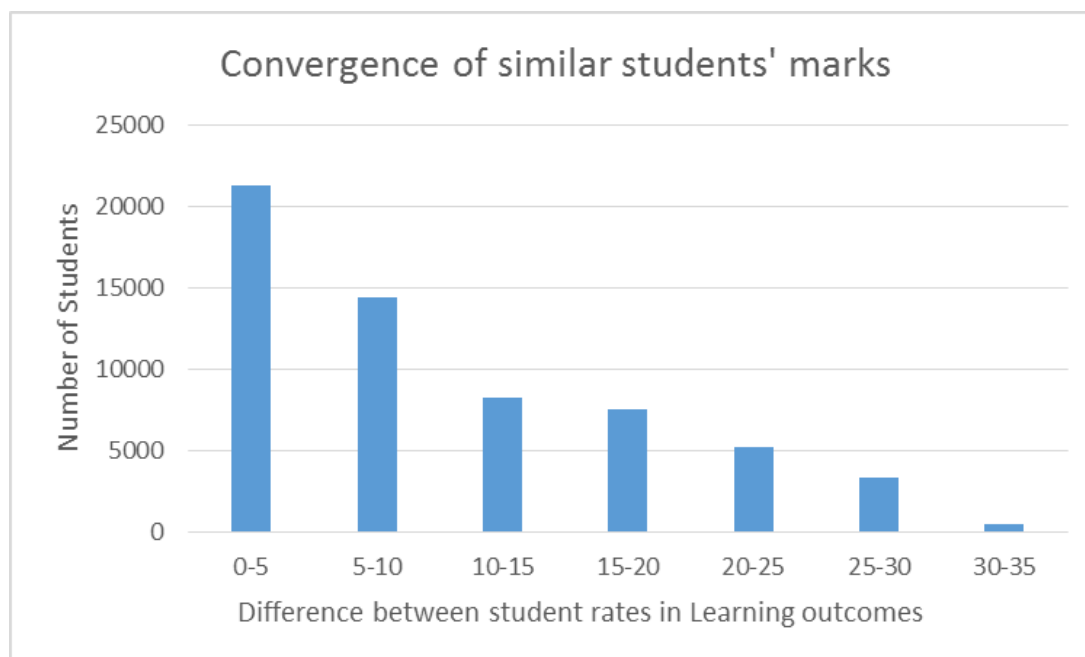


**Figure 6. 31: Gap between junior active & similar student marks - batch 421**

When looking at the same analysis for senior students, results shows 30% of students having a gap average between 0-5 marks far from the active student, 27% having 5-10 gap, 19% having 10-15 and 25% more than 15-mark gap as shown in figure (6.32).



**Figure 6. 32: Gap between senior active & similar student marks**



**Figure 6. 33: Convergence of similar students' marks**

### 6.2.5 Applying K-means method using R

The following shows a partitioning cluster analysis for both junior and senior students. The first step is preparing data for clustering by estimating missing data and rescaling variables for

comparability using omit and scale commands in R. the second step is applying the most popular partitioning cluster method k-means after determining the appropriate number of clusters.

```
[Previously saved workspace restored]
```

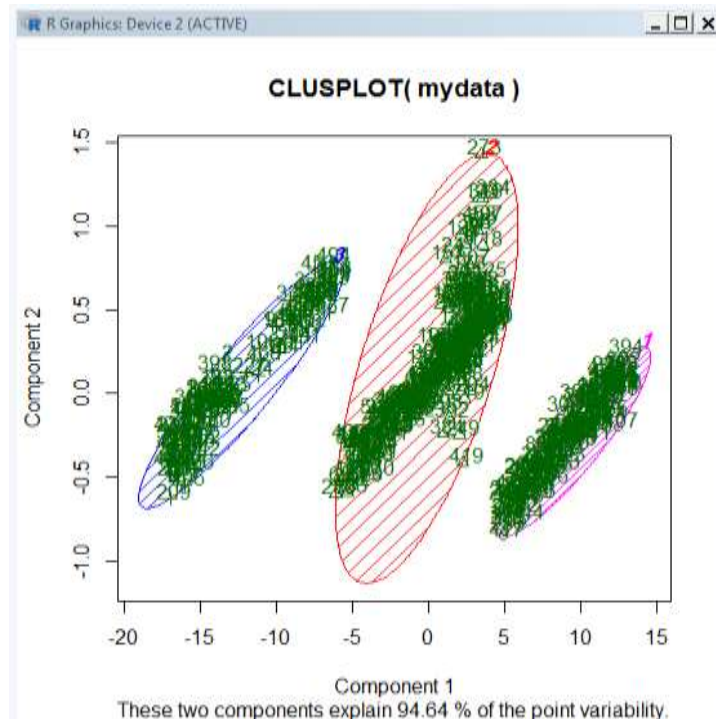
```
> mydata <- read.csv("D:/thesis/MySourceCode/Recomender/clustering/353122017_133$
> mydata <- na.omit(mydata)
> mydata <- scale(mydata)
> wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
> for (i in 2:15) wss[i] <- sum(kmeans(mydata,
+   centers=i)$withinss)
> plot(1:15, wss, type="b", xlab="Number of Clusters",
+   ylab="Within groups sum of squares")
>
> fit <- kmeans(mydata, 3)
> aggregate(mydata,by=list(fit$cluster),FUN=mean)
  Group.1      X82.0      X83.0      X83.0.1      X82.0.1      X83.0.2
1      1 -1.01567885 -1.04590657 -1.04223884 -1.02731023 -1.00954036
2      2 -0.07155703 -0.04291309 -0.03957884 -0.05665539 -0.07382744
3      3  1.51820241  1.49802602  1.48601084  1.50219866  1.51476743
      X80.0      X80.0.1      X80.0.2      X82.0.2      X82.0.3      X82.0.4
```

**Figure 6. 34: K-mean analysis for senior and junior students**

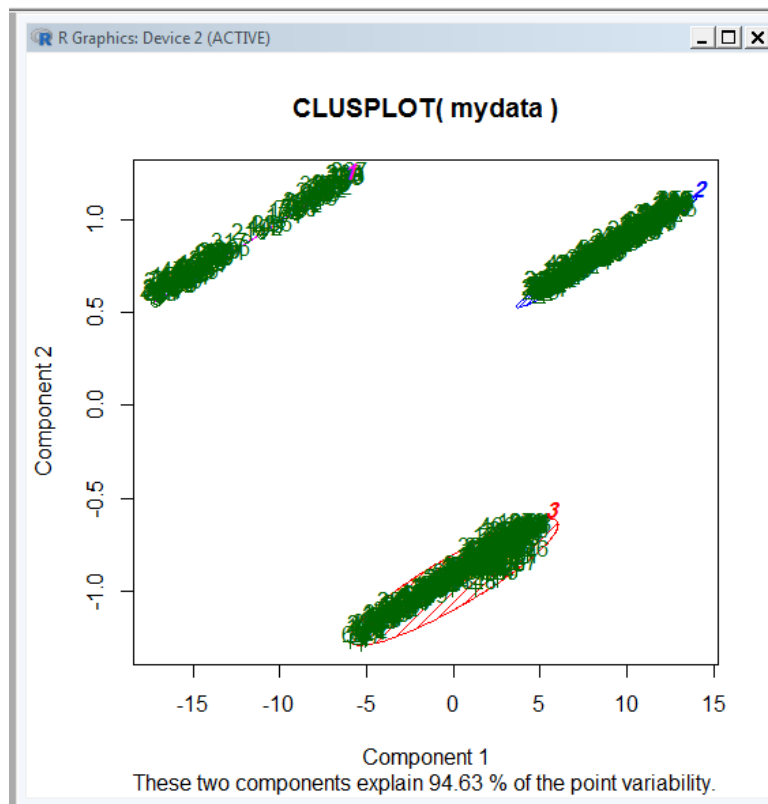
The following figures shows the plot of clustering results for both senior and junior students where three clusters are shown:

```
> mydata <- data.frame(mydata, fit$cluster)
> library(cluster)
> clusplot(mydata, fit$cluster, color=TRUE, shade=TRUE,
+   labels=2, lines=0)
> |
```

**Figure 6. 35: using the clusplot command to show clustering results in R**



**Figure 6. 36: Cluster plot for junior students' analysis**



**Figure 6. 37:Cluster plot for senior students' analysis**

### 6.3 Learning Material Recommendation

Once the set of similar students was found out, the engine starts learning the behavior of best students according to the set of learning materials linked with the active course. Students' behavior could be evaluated (1) based on their rates or (2) based on the percentage of hits on a learning material compared to the total number of hits on all learning materials recommended for each student at that time.

The engine lists a matrix of the best students and learning materials with their rating or hit percentage to find out the best learning material that fits the active student as shown in the matrix below:

**Rate matrix:**

**Table 6. 5 Rating matrix of best students**

Student/L M	LM1	LM2	LM3	LM4	LM5	LM6	LM <i>n</i>
Student 1	Rates	Rates		Rates	Rates	Rates	Rates
Student 2	Rates	Rates	Rates	Rates	Rates		Rates
Student 3	Rates		Rates	Rates		Rates	Rates
Student <i>n</i>	Rates	Rates	Rates		Rates	Rates	Rates
	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$

### Percentage of hit matrix:

**Table 6. 6 Rating matrix of best students**

Student/L M	LM1	LM2	LM3	LM4	LM5	LM6	LM <i>n</i>
Student 1	# of hits%	# of hits%		# of hits%	# of hits%		
Student 2	# of hits%	# of hits%	# of hits%	# of hits%	# of hits%	# of hits%	# of hits%
Student 3	# of hits%	# of hits%		# of hits%	# of hits%	# of hits%	# of hits%
Student <i>n</i>	# of hits%		# of hits%	# of hits%	# of hits%		
	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$	$\frac{\sum hit\%}{\# of Students}$

The empty cells represent learning materials for which the student has not taken or has no rating on, the value of the empty cells is considered as zero.

After building the matrix, the engine finds out the vertical summation for each learning material which is considered as the learning material score. Finally, the engine ranks the learning material based on its scores to generate the first recommender draft.

The second recommendation draft is generated by building the same matrix for good students in active course regardless of their similarity with the active student. So, two recommendation lists will be provided as shown:

**Table 6. 7 Two recommendation list is generated for good similar students and good students for an active course**

Good Similar Student in Active Course X	
LM1	4.9
LM2	4.15
LM3	4.05
LM4	3.9
LM5	3.86
LM6	3.1
LM <sub><i>n</i></sub>	2.9

Good Students in Active Course X	
LM1	4.97
LM2	4.55
LM3	4.45
LM4	3.91
LM5	3.76
LM6	3.163
LM <sub><i>n</i></sub>	2.8

In the final step, the engine will refer to the “Recommender engine setup page” to figure out the weight of good similar students “Similar Student Weight” and good students “Excellent Students Weights” to be reflected on each generated list mentioned above, as shown in the figure below:

The screenshot displays the 'Recommender Engine Setup' interface. It is divided into three main sections: 'Student Similarity Parameters', 'Recommender Types', and 'Collaborative Recommender Parameters'. The 'Student Similarity Parameters' section includes dropdown menus for 'Similarity Measure' (set to 4), 'Algorithm' (Distance Euclidian), 'Matrix Type' (Sparse Matrix), and 'Achievement level' (Learning Outcomes). The 'Recommender Types' section features sliders for 'Collaborative Recommender weight' (80%) and 'Content Recommender weight' (20%). The 'Collaborative Recommender Parameters' section includes a 'Good Marks' field (80) and two weighted fields: 'Similar Student weight' (75%) and 'Excellent Student weight' (25%), which are highlighted with a red rectangular border.

Section	Parameter	Value
Student Similarity Parameters	Similarity Measure	4
	Algorithm	Distance Euclidian
	Matrix Type	Sparse Matrix
	Achievement level	Learning Outcomes
Recommender Types	Collaborative Recommender weight	80%
	Content Recommender weight	20%
Collaborative Recommender Parameters	Good Marks	80
	Similar Student weight	75%
	Excellent Student weight	25%

**Figure 6. 38: Recommender engine setup page**

Figure (6.38) shows that rating weight of similar student is set to 75% which means that it is more valuable than excellent (and not similar) students rates which is set to 25%. The weights are multiplied with the values of each learning material in the two recommended lists as below.

**Table 6. 8 Recommendation Lists based on good students and similar good students weight**

Good Similar Student in Active Course X	
LM1	$4.9 \times 0.75$
LM2	$4.15 \times 0.75$
LM3	$4.05 \times 0.75$
LM4	$3.9 \times 0.75$
LM5	$3.86 \times 0.75$
LM6	$3.1 \times 0.75$
LM <sub>n</sub>	$2.9 \times 0.75$

Good Students in Active Course X	
LM1	$4.97 \times 0.25$
LM2	$4.55 \times 0.25$
LM3	$4.45 \times 0.25$
LM4	$3.91 \times 0.25$
LM5	$3.76 \times 0.25$
LM6	$3.163 \times 0.25$
LM <sub>n</sub>	$2.8 \times 0.25$

Finally, the two list is merged and based on the final score for each learning material, a final recommender list will be provided for the students ranked based on the final weights.



## Chapter 7

### **Conclusion**

## 7.1 Conclusion

Because each student is a unique individual with personalized needs, learning styles, learning preferences, knowledge levels, and knowledge backgrounds. This research attempt to recommend learning material for students based on their knowledge level minimizing the gap between them and knowledge provided making learning process easier and more interesting. This could be achieved by learning students' behavior in using learning materiel making use of those students who was similar to an active student in their knowledge level and achieve excellent or good marks.

To find out useful learning materials, a hybrid recommender engine of two approaches collaborative and content-base was designed to work simultaneously. The collaborative recommender engine looks for similar good students among the university history and then makes use of their success experience in using learning material; similar good students are those students who gain high marks in the active courses and are similar to an active student in his level of knowledge at the time he took the course, so the engine could reflect their good experience in using learning materials on current similar student.

Five different distance algorithms are used to find out the similarity between students, K-means cluster algorithm using cosine distance is used in order to find out similar students. Students behavior toward learning material is measured using students' rates or number of hits on learning material. The engine setup screen is also used in order to configure the weight of good similar students to good students in order to give final scores for learning material and finally rank them based on their scores.

## 7.2 Limitations

Sampling and Dataset:

The dataset is one of the main limitation in the research as there is no learning environment records student's achievement on the level of learning outcomes. Although the study generates a dataset which mimic a learning environment, but in real life students differ in their achievements based on specialization, some are interested in math, algorithms, graph theory ... etc. where others are interested in web development, human interaction and front-end design as an example.

Similarity between students:

When applying k-mean clustering on space matrices using the cosine function; that shows to have the lowest SSE, a set of students of ~ 25% appears in the same cluster as the active student, although the average differences between their achievements and the active student achievement is between 15 and 30. This may refer to the zeroes in the sparse matrix which denote that the student didn't take the learning outcome.

## 7.3 Future Work

As mentioned at the beginning of the research, this study provides a suggested hybrid recommender system where the content base and collaborative approaches works simultaneously. The research concentrates on the collaborative recommender approach leaving the content-based for future study.

The content based is expected to result in a recommendation list based on two embedded stages, the first stage searches all learning materials which have learning outcomes that best match the learning outcomes linked with the active course. whereas the second stage compare the content of the learning material with the content of the active course resulting in a final content recommendation list.

## References

- [1] Itmazi, Jamil (2010) A Suggested Algorithm of Recommender System to Recommend Learning Objects from Digital Library to Learning Management System , *Asian Journal of Information Technology*, Volume: 9 | Issue: 2 | Page No.: 37-44.
- [2] Amornsinlaphachai, Pensri ( 2014) Designing a learning model using the STAD technique with a suggestion system to decrease learners' weakness, *5th World Conference on Educational Sciences - WCES*, [Volume 116](#), 21 Pages 431-435.
- [3] Amran K, Ghauth & Nor Aniza, Abdullah (2011) The Effect of Incorporating Good Learners' Ratings in e-Learning Contentbased Recommender System, *Educational Technology & Society*, v14 n2 p248-257.
- [4] Tarus, John. Niu, Zhendong. Khadidja, Bakhti,( 2017) E-Learning Recommender System Based on Collaborative Filtering and Ontology, World Academy of Science, Engineering and Technology, *International Journal of Computer and Information Engineering*, Vol,11, No2.
- [5] Aher, Sunita & Lobo L.M.R.J, (2012), Best Combination of Machine Learning Algorithms for Course Recommendation System in E-learning , *International Journal of Computer Applications*, Volume 41– No.6.
- [6] Zaiane, R Osmar., (2003), Building a recommender agent for e-learning systems.
- [7] Ricci, Francesco., Rokach, Lior Shapira, Bracha, (2010) Recommender Systems Handbook *Springer*, New York Dordrecht Heidelberg London.
- [8] O'Brien M Jeffrey. ( 2006) The race to create a 'smart' Google, [Internet Dating Excellence Association](#).
- [9] HERLOCKER L JONATHAN, KONSTAN A JOSEPH, TERVEEN, G LOREN. and RIEDL, T JOHN. . (2004) Evaluating Collaborative Filtering Recommender Systems, *ACM*, Vol. 22, No. 1, Pages 5–53.
- [10] Seroussi Yanir(2015) THE WONDERFUL WORLD OF RECOMMENDER SYSTEMS, <https://yanirseroussi.com/2015/10/02/the-wonderful-world-of-recommender-systems/>
- [11] Smeaton, F, Alan.. , Callan, Jamie , (2005). Personalisation and recommender systems in digital libraries, *Digital Object Identifier*, V 5, p299–308 .

- [12] Isinkaye F.O. Folajimi Y.O. , Ojokoh B.A., (2015). Recommendation systems Principles, methods and evaluation, *Egyptian Informatics Journal* 16, 261–273.
- [13] Huang Zan, Chung Wingyan. (2002). Thian-Huat Ong, Hsinchun Chen, A Graph-based Recommender System for Digital Library, Conference Paper · July 13-17, 2002, Portland, Oregon, USA.
- [14] Clark, Nick. (2014) Towards a European Higher Education Area 15 Years of Bologna, *World Education News & Reviews*, June 3, 2014
- [15] European Commission (EACEA)Eurydice,(2015) The European Higher Education Area in 2015: Bologna Process Implementation Report. *Luxembourg*: Publications Office of the European Union.
- [16] KENNEDY, D (2006) Writing and using learning outcomes: a practical guide, Cork, *University College Cork*.
- [17] Fan Yang., Zhenghong, Dong ( 2017) Learning Path Construction in e-Learning, chapter 2, *Springer Singapore*, Academy of Equipment , Beijing, China.
- [18] Fuller, Ursula 8et al. ( 2007) Developing a Computer Science-specific Learning Taxonomy, *ACM*.
- [19] Shabatura, Jessica ( 2013) Using Bloom’s Taxonomy to Write Effective Learning Objectives,September 27, 2013
- [20] Huitt, W. (2011). Bloom et al.'s taxonomy of the cognitive domain,Educational Psychology Interactive. Valdosta, GA: Valdosta State University
- [21] B e n j a m i n S , B l o o m . Max, D. Engelhart. E d w a r d J , F u r s t H i l l . H W a l k e r . (1956). TAXONOMY OF EDUCATIONAL OBJECTIVES, HANDBOOK1, LONGMANS, <https://www.scribd.com/document/145157500/Bloom-Taxonomy-of-Educational-Objectives>
- [22] Centre for Teaching Support & Innovation, University of Toronto, (2008). Developing Learning Outcomes A Guide for University of Toronto Faculty. <http://teaching.utoronto.ca/wp-content/uploads/2015/08/Developing-Learning-Outcomes-Guide-Aug-2014.pdf>,

- [23] Sudhana, Madhu Kalla. Raj V. Cyril. Ravi, T. ( 2013) An Architectural-model for Context aware Adaptive Delivery of Learning Material, *International Journal of Advanced Computer Science and Applications*, Vol. 4, No. 10, 2013
- [24] Omedes, Jose. ( 2016) What is adaptive e-Learning, Learning Analytics & Adaptive Learning 29,sep,2016
- [25] K-12 eLearning(2018)[http://www.elearningnc.gov/about\\_elearning](http://www.elearningnc.gov/about_elearning).
- [26] Boyle, J. Anderson. , Farrell, C. Reiser, B. (1987)Cognitive principles in the design of computer tutors. In P. Morris (Ed.), *Modeling cognition*. NY: John Wiley.
- [27] Wenting Ma. Adesope O. Olusola. . Nesbit, C. John and Liu, Qing ( 2014), Intelligent Tutoring Systems and Learning Outcomes: A Meta-Analysis, American Psychological Association, *Journal of Educational Psychology*, 2014, Vol. 106, No. 4, 901–918.
- [28] Dominic, Maria., Francis, Sagayaraj. (2015) An Adaptable E-Learning Architecture Based on Learners’ Profiling, Modern Education and Computer Science,
- [29] MOORE, S.DAVID. McCABE, P. GEORGE. CRAIG, A.BRUCE (2009) Introduction to the Practice of Statistics, First printing, W. H. Freeman and Company, USA. [http://eunacal.org/metodakerkimi/wp-content/uploads/spss/Introduction\\_to\\_the\\_Practice\\_of\\_Statistics\\_6th.pdf](http://eunacal.org/metodakerkimi/wp-content/uploads/spss/Introduction_to_the_Practice_of_Statistics_6th.pdf)
- [30] T. Agami Reddy(2011) Probability Concepts and Probability Distributions, Applied Data Analysis and Modeling for Energy Engineers and Scientists,DOI 10.1007/978-1-4419-9613-8\_2, © Springer Science&Business Media, LLC 2011
- [31] Seltman J.Howard (2015) Experimental Design and Analysis. <http://www.stat.cmu.edu/~hseltman/309/Book/Book.pdf>
- [32] Tan, Pang-Ning, Steinbach, Michael, Kumar, Vipin(2006) Introduction to Data Mining. <https://www-users.cs.umn.edu/~kumar001/dmbook/sol.pdf>
- [33] , Podani, János (2000) Introduction to the Exploration of Multivariate Biological Data, *Backhuys Publishers*.
- [34] Goshtasby A. Ardeshir, (2012) Images Registration Principles, Tools and Methods, Springer, London Dordrecht Heidelberg New York. [ftp://nozdr.ru/biblio/kolxo3/Cs/CsIp/Goshtasby%20A.%20Image%20registration.%20Principles,%20tools%20and%20methods%20\(Springer,%202012\)\(ISBN%209781447124573\)\(O\)\(460s\)\\_CsIp\\_.pdf](ftp://nozdr.ru/biblio/kolxo3/Cs/CsIp/Goshtasby%20A.%20Image%20registration.%20Principles,%20tools%20and%20methods%20(Springer,%202012)(ISBN%209781447124573)(O)(460s)_CsIp_.pdf)

- [35] Hertz, Tomer(2006), Learning Distance Functions: Algorithms and Applications, THE HEBREW UNIVERSITY OF JERUSALEM.
- [36] HARTIGAN y J. A. and WONG, M. A. , (1979) ,A K-Means Clustering Algorithm”, *Journal of the Royal Statistical Society Series C (Applied Statistics)*, Vol. 28, No. 1
- [37] Everitt, Brian Hothorn, Torsten, (2011) An Introduction to Applied Multivariate Analysis with R, *Springer*
- [38] Chandrasekaran, B. and John R Josephson. (1999) What Are Ontologies, and Why Do We Need Them?, *IEEE INTELLIGENT SYSTEMS*.
- [39] Daniar Asanov,Berlin Institute of Technology, Berlin, Germany,(2004) Algorithms and Methods in Recommender Systems , Berlin Institute of Technology. [https://www.snet.tu-berlin.de/fileadmin/fg220/courses/SS11/snet-project/recommender-systems\\_asanov.pdf](https://www.snet.tu-berlin.de/fileadmin/fg220/courses/SS11/snet-project/recommender-systems_asanov.pdf)
- [40] Badrul Sarwar, Karypis, George. Konstan,Joseph and Riedl, John.,(2001), “Item-based Collaborative Filtering Recommendation Algorithms”, *Army HPC Research Center*. <http://wwwconference.org/proceedings/www10/papers/pdf/p519.pdf>
- [41] Daniel Lew(2007) Recommender systems, Computer Science Comprehensive Exercise, *Carleton College*.
- [42] Shih, Ya-Yueh , Liu, Duen-Ren (2005) Hybrid recommendation approaches: collaborative filtering via valuable content information, *Hawaii International Conference on System Sciences*.
- [43] Ahmad A. Shukr , (2016) Standard Based Exchange Of Learning Objects: Towards Outcome Based Learning, *University of Al-Quds*.
- [44] Baha Thabit (2017) Dynamic Learning Profile based on Achieved Learning Outcomes, *University of Al-Quds*
- [45] Matthew A. Peeples, (2011) R Script for K-Means Cluster Analysis. [online]. Available: <http://www.mattpeeples.net/kmeans.html>. ( March 28, 2018 )
- [46] TANG, Ya. Tiffany. , MCCALLA, Gordon (2013) “Smart Recommendation for an Evolving E-Learning System” , *University of Saskatchewan*
- [47] Priyadharsini.C , Thanamani Antony Selvadoss (2014), “An Overview of Knowledge Discovery Database and Data mining Techniques”, *ICGICT'14*.  
[http://www.ijircce.com/upload/2014/icgict14/251\\_1041.pdf](http://www.ijircce.com/upload/2014/icgict14/251_1041.pdf)

[48] ECTS Users' Guide, European Communities, European Union, (2015)

[49] David R. Krathwohl, (2002) A Revision of Bloom's Taxonomy: An Overview, *EBSCO publishing*

<https://www.depauw.edu/files/resources/krathwohl.pdf>

[50] Marlies Baeten, Eva Kyndt, Katrien Struyven, Filip Dochy (2010) Using student-centred learning environments to stimulate deep approaches to learning: Factors encouraging or discouraging their effectiveness, *Elsevier*

[51] F Brown, Kathy Laboard (2003) From Teacher-Centered To Learner-Centered Curriculum: Improving Learning In Diverse Classrooms, *EBSCO*



## Appendix A

## SSE R Script

# Script by Matt Peeples <http://www.mattpeeples.net/kmeans.html>

```
# initialize all necessary libraries
library(cluster)
library(psych)
```

```
# read CSV file - (kmeans_data.csv) - convert to a matrix
data1 <- read.table(file='kmeans_data.csv', sep=',', header=T, row.names=1)
data.p <- as.matrix(data1)
```

```
# Ask for user input - convert raw counts to percents?
choose.per <- function(){readline("Covert data to percents? 1=yes, 2=no : ")}
per <- as.integer(choose.per())
```

```
# If user selects yes, convert data from counts to percents
if (per == 1) {
  data.p <- prop.table(data.p,1)*100}
```

```
# Ask for user input - Z-score standardize data?
choose.stand <- function(){readline("Z-score standardize data? 1=yes, 2=no : ")}
stand <- as.integer(choose.stand())
```

```
# If user selects yes, Z-score standardize data
kdata <- na.omit(data.p)
if (stand == 1) {
  kdata <- scale(kdata)}
```

```
# Ask for user input - determine the number of cluster solutions to test (must between 2 and the number of
rows in the database)
choose.level <- function(){readline("How many clustering solutions to test (> row numbers)? ")}
n.lev <- as.integer(choose.level())
```

```
# Calculate the within groups sum of squared error (SSE) for the number of cluster solutions selected by the
user
wss <- rnorm(10)
while (prod(wss==sort(wss,decreasing=T))==0) {
  wss <- (nrow(kdata)-1)*sum(apply(kdata,2,var))
  for (i in 2:n.lev) wss[i] <- sum(kmeans(kdata, centers=i)$withinss)}
```

```
# Calculate the within groups SSE for 250 randomized data sets (based on the original input data)
k.rand <- function(x){
  km.rand <- apply(x,2,sample)
  rand.wss <- as.matrix(dim(x)[1]-1)*sum(apply(km.rand,2,var))
  for (i in 2:n.lev) rand.wss[i] <- sum(kmeans(km.rand, centers=i)$withinss)
  rand.wss <- as.matrix(rand.wss)
  return(rand.wss)}
rand.mat <- matrix(0,n.lev,250)
k.1 <- function(x) {
  for (i in 1:250) {
    r.mat <- as.matrix(suppressWarnings(k.rand(kdata)))
```

```

rand.mat[,i] <- r.mat}
return(rand.mat)}

# Same function as above for data with < 3 column variables
k.2.rand <- function(x){
  rand.mat <- matrix(0,n.lev,250)
  km.rand <- matrix(sample(x),dim(x)[1],dim(x)[2])
  rand.wss <- as.matrix(dim(x)[1]-1)*sum(apply(km.rand,2,var))
  for (i in 2:n.lev) rand.wss[i] <- sum(kmeans(km.rand, centers=i)$withinss)
  rand.wss <- as.matrix(rand.wss)
  return(rand.wss)}
k.2 <- function(x){
  for (i in 1:250) {
    r.1 <- k.2.rand(kdata)
    rand.mat[,i] <- r.1}
  return(rand.mat)}

# Determine if the data table has > or < 3 variables and call appropriate function above
if (dim(kdata)[2] == 2) { rand.mat <- k.2(kdata) } else { rand.mat <- k.1(kdata) }

# Plot within groups SSE against all tested cluster solutions for actual and randomized data - 1st: Log scale,
# 2nd: Normal scale
par(ask=TRUE)
xrange <- range(1:n.lev)
yrange <- range(log(rand.mat),log(wss))
plot(xrange,yrange, type='n', xlab='Cluster Solution', ylab='Log of Within Group SSE', main='Cluster
Solutions against Log of SSE')
for (i in 1:250) lines(log(rand.mat[,i]),type='l',col='red')
lines(log(wss), type='b', col='blue')
legend('topright',c('Actual Data', '250 Random Runs'), col=c('blue', 'red'), lty=1)
par(ask=TRUE)
yrange <- range(rand.mat,wss)
plot(xrange,yrange, type='n', xlab='Cluster Solution', ylab='Within Groups SSE', main='Cluster
Solutions against SSE')
for (i in 1:250) lines(rand.mat[,i],type='l',col='red')
lines(1:n.lev, wss, type='b', col='blue')
legend('topright',c('Actual Data', '250 Random Runs'), col=c('blue', 'red'), lty=1)

# Calculate the mean and standard deviation of difference between SSE of actual data and SSE of 250
# randomized datasets
r.sse <- matrix(0,dim(rand.mat)[1],dim(rand.mat)[2])
wss.1 <- as.matrix(wss)
for (i in 1:dim(r.sse)[2]) {
  r.temp <- abs(rand.mat[,i]-wss.1[,1])
  r.sse[,i] <- r.temp}
r.sse.m <- apply(r.sse,1,mean)
r.sse.sd <- apply(r.sse,1,sd)
r.sse.plus <- r.sse.m + r.sse.sd
r.sse.min <- r.sse.m - r.sse.sd

# Plot difference between actual SSE mean SSE from 250 randomized datasets - 1st: Log scale, 2nd: Normal
# scale
par(ask=TRUE)
xrange <- range(1:n.lev)
yrange <- range(log(r.sse.plus),log(r.sse.min))

```

```

plot(xrange,yrange, type='n',xlab='Cluster Solution', ylab='Log of SSE - Random SSE', main='Cluster
Solutions against (Log of SSE - Random SSE)')
lines(log(r.sse.m), type="b", col='blue')
lines(log(r.sse.plus), type='l', col='red')
lines(log(r.sse.min), type='l', col='red')
legend('topright',c('SSE - random SSE', 'SD of SSE-random SSE'), col=c('blue', 'red'), lty=1)
par(ask=TRUE)
xrange <- range(1:n.lev)
yrange <- range(r.sse.plus,r.sse.min)
plot(xrange,yrange, type='n',xlab='Cluster Solution', ylab='SSE - Random SSE', main='Cluster Solutions
against (SSE - Random SSE)')
lines(r.sse.m, type="b", col='blue')
lines(r.sse.plus, type='l', col='red')
lines(r.sse.min, type='l', col='red')
legend('topright',c('SSE - random SSE', 'SD of SSE-random SSE'), col=c('blue', 'red'), lty=1)

# Ask for user input - Select the appropriate number of clusters
choose.clust <- function(){readline("What clustering solution would you like to use? ")}
clust.level <- as.integer(choose.clust())

# Apply K-means cluster solutions - append clusters to CSV file
fit <- kmeans(kdata, clust.level)
aggregate(kdata, by=list(fit$cluster), FUN=mean)
clust.out <- fit$cluster
kclust <- as.matrix(clust.out)
kclust.out <- cbind(kclust, data1)
write.table(kclust.out, file="kmeans_out.csv", sep=",")

# Display Principal Components plot of data with clusters identified
par(ask=TRUE)
clusplot(kdata, fit$cluster, shade=F, labels=2, lines=0, color=T, lty=4, main='Principal Components plot
showing K-means clusters')

# Send output to files
kclust.out.p <- prop.table(as.matrix(kclust.out),1)*100
out <- capture.output(describe.by(kclust.out.p,kclust))
cat(out,file='Kmeans_out.txt', sep='\n', append=F)
pdf(file="kmeans_out.pdf")
xrange <- range(1:n.lev)
yrange <- range(log(rand.mat),log(wss))
plot(xrange,yrange, type='n', xlab='Cluster Solution', ylab='Log of Within Group SSE', main='Cluster
Solutions against Log of SSE')
for (i in 1:250) lines(log(rand.mat[,i]),type='l',col='red')
lines(log(wss), type="b", col='blue')
legend('topright',c('Actual Data', '250 Random Runs'), col=c('blue', 'red'), lty=1)
yrange <- range(rand.mat,wss)
plot(xrange,yrange, type='n', xlab="Cluster Solution", ylab="Within Groups SSE", main="Cluster
Solutions against SSE")
for (i in 1:250) lines(rand.mat[,i],type='l',col='red')
lines(1:n.lev, wss, type="b", col='blue')
legend('topright',c('Actual Data', '250 Random Runs'), col=c('blue', 'red'), lty=1)
xrange <- range(1:n.lev)
yrange <- range(log(r.sse.plus),log(r.sse.min))
plot(xrange,yrange, type='n',xlab='Cluster Solution', ylab='Log of SSE - Random SSE', main='Cluster
Solutions against (Log of SSE - Random SSE)')

```

```

lines(log(r.sse.m), type="b", col='blue')
lines(log(r.sse.plus), type='l', col='red')
lines(log(r.sse.min), type='l', col='red')
legend('topright',c('SSE - random SSE', 'SD of SSE-random SSE'), col=c('blue', 'red'), lty=1)
xrange <- range(1:n.lev)
yrange <- range(r.sse.plus,r.sse.min)
plot(xrange,yrange, type='n',xlab='Cluster Solution', ylab='SSE - Random SSE', main='Cluster Solutions
against (SSE - Random SSE)')
lines(r.sse.m, type="b", col='blue')
lines(r.sse.plus, type='l', col='red')
lines(r.sse.min, type='l', col='red')
legend('topright',c('SSE - random SSE', 'SD of SSE-random SSE'), col=c('blue', 'red'), lty=1)
clusplot(kdata, fit$cluster, shade=F, labels=2, lines=0, color=T, lty=4, main='Principal Components plot
showing K-means clusters')
dev.off()

# end of script

```

## Generating Data Set Scripts

*/\* Formatted on 17-Nov-2017 20:24:20 By Abeer Mousa(QP5 v5.206) \*/*

```
DECLARE
  lo_no  NUMBER;
  i      NUMBER;
  counter NUMBER := 0;
BEGIN
  FOR rec IN (SELECT * FROM sr_courses)
  LOOP
    lo_no :=
      DBMS_RANDOM.VALUE (5,
                          8);

    FOR i IN 1 .. lo_no
    LOOP
      counter := counter + 1;

      INSERT INTO sr_learning_outcomes
        VALUES (LPAD (counter,
                        5,
                        0),
                'LO' || i || '_' || rec.course_id);

      INSERT INTO sr_course_learning_outcomes
        VALUES (LPAD (counter,
                        5,
                        0),
                rec.course_id,
                i);
    END LOOP;
  END LOOP;
END;
```

-----<>

```
DELETE FROM sr_student_learning_outcomes;
```

*--Script for student learning outcomes:*

```
BEGIN
  --Graduated students
  FOR rec IN (SELECT *
              FROM sr_students
              WHERE reg_year <= 2013)
  LOOP
    --give each student all required courses
    FOR rec_lo
    IN (SELECT a.course_id, b.learning_outcome_id
        FROM sr_courses a, sr_course_learning_outcomes b
        WHERE a.course_id = b.course_id AND level_type = 'RS')
    LOOP
      INSERT INTO sr_student_learning_outcomes
        VALUES (rec.student_id, rec_lo.course_id, rec_lo.learning_outcome_id
```

```

        , NULL);
END LOOP;

COMMIT;

--give each student random set of optional courses

FOR rec_courses IN (SELECT course_id
                    FROM ( SELECT course_id, DBMS_RANDOM.VALUE () rnd
                        FROM sr_courses
                        WHERE level_type = 'OS'
                        ORDER BY rnd)
                    WHERE ROWNUM < 15)
LOOP
    FOR rec_lo IN (SELECT *
                  FROM sr_course_learning_outcomes c
                  WHERE c.course_id = rec_courses.course_id)
    LOOP
        INSERT INTO sr_student_learning_outcomes
            VALUES (rec.student_id, rec_lo.course_id, rec_lo.learning_outcome_id
                , NULL);
    END LOOP;
END LOOP;
END LOOP;
END;

-----<>
--Testing Student Courses total hours
--Results must be maximum 120 Credit hours

SELECT student_id, SUM (credit_hours)
FROM sr_courses a, (SELECT DISTINCT student_id, course_id FROM sr_student_learning_outcomes) b
WHERE a.course_id = b.course_id
GROUP BY student_id
-----<>
-----<>
-----<>

--insert LO for student 2016

begin
FOR rec IN (SELECT *
            FROM sr_students
            WHERE reg_year = 2016)
LOOP
FOR rec_lo
IN (SELECT a.course_id, b.learning_outcome_id
    FROM sr_courses a, sr_course_learning_outcomes b
    WHERE a.course_id = b.course_id AND level_type = 'RS'
    and acadimic_year = 1)
LOOP
    INSERT INTO sr_student_learning_outcomes
        VALUES (rec.student_id,
            rec_lo.course_id,

```

```

        rec_lo.learning_outcome_id,
        NULL);
END LOOP;

```

```

END LOOP;
END;

```

```

-----<>
--testing for 2016 student credit hours

```

```

--Testing Student Courses total hours
--Results must be maximum 33 Credit hours

```

```

SELECT b.student_id, SUM (credit_hours)
FROM sr_courses a, (SELECT DISTINCT student_id, course_id FROM sr_student_learning_outcomes) b,
sr_students s
WHERE a.course_id = b.course_id
AND b.student_id = s.student_id
AND s.reg_year = '2016'
GROUP BY b.student_id

```

```

-----<>
--insert LO for student 2016

```

```

begin
FOR rec IN (SELECT *
            FROM sr_students
            WHERE reg_year = 2015)
LOOP
FOR rec_lo
IN (SELECT a.course_id, b.learning_outcome_id
    FROM sr_courses a, sr_course_learning_outcomes b
    WHERE a.course_id = b.course_id AND level_type = 'RS' and acadimic_year in( 1,2))
LOOP
INSERT INTO sr_student_learning_outcomes
VALUES (rec.student_id,
        rec_lo.course_id,
        rec_lo.learning_outcome_id,
        NULL);
END LOOP;

```

```

END LOOP;
END;

```

```

-----<>
--testing for 2015 student credit hours

```

```

--Testing Student Courses total hours
--Results must be maximum 120 Credit hours

```

```

SELECT b.student_id, SUM (credit_hours)
FROM sr_courses a, (SELECT DISTINCT student_id, course_id
FROM sr_student_learning_outcomes) b, sr_students s
WHERE a.course_id = b.course_id

```



```

        AND b.student_id = s.student_id
        AND s.reg_year = '2015'
GROUP BY b.student_id
-----<>

```

```

BEGIN
--Graduated students
FOR rec IN (SELECT *
            FROM sr_students
            WHERE reg_year = 2014)
LOOP
--give each student all required courses
FOR rec_lo
IN (SELECT a.course_id, b.learning_outcome_id
    FROM sr_courses a, sr_course_learning_outcomes b
    WHERE a.course_id = b.course_id AND level_type = 'RS')
LOOP
INSERT INTO sr_student_learning_outcomes
VALUES (rec.student_id,
        rec_lo.course_id,
        rec_lo.learning_outcome_id,
        NULL);
END LOOP;

```

```

COMMIT;

--give each student random set of optional courses
--given from acadimic year 3 and less than 9 courses because
--there is 3 required courses

```

```

FOR rec_courses IN (SELECT course_id
                    FROM ( SELECT course_id,
                                DBMS_RANDOM.VALUE () rnd
                            FROM sr_courses
                            WHERE level_type = 'OS'
                                AND acadimic_year = 3
                            ORDER BY rnd)
                    WHERE ROWNUM < 9)
LOOP
FOR rec_lo IN (SELECT *
               FROM sr_course_learning_outcomes C
               WHERE c.course_id = rec_courses.course_id)
LOOP
INSERT INTO sr_student_learning_outcomes
VALUES (rec.student_id, rec_lo.course_id, rec_lo.learning_outcome_id,
        NULL);

END LOOP;
END LOOP;
END LOOP;
END;

```

```

-----<>
--testing for 2014 student credit hours

```

--Testing Student Courses total hours  
 --Results must be maximum 102 Credit hours

```
SELECT b.student_id, SUM (credit_hours)
FROM sr_courses a, (SELECT DISTINCT student_id, course_id
FROM sr_student_learning_outcomes) b, sr_students s
WHERE a.course_id = b.course_id
AND b.student_id = s.student_id
AND s.reg_year = '2014'
--and a.academic_year = 3
GROUP BY b.student_id
```

-----<>  
 /\*All the above scripts deal with courses which student finalized and got their  
 marks on them  
 The below scripts deal with students who currently are taking the set of courses \*/  
 -----<>  
 -----<>  
 -----<>  
 --insert LO for student 2017

```
begin
FOR rec IN (SELECT *
FROM sr_students
WHERE reg_year = 2017)
LOOP
FOR rec_lo
IN (SELECT a.course_id, b.learning_outcome_id
FROM sr_courses a, sr_course_learning_outcomes b
WHERE a.course_id = b.course_id AND level_type = 'RS'
and academic_year in( 1,2))
LOOP
INSERT INTO sr_student_learning_outcomes
VALUES (rec.student_id,
rec_lo.course_id,
rec_lo.learning_outcome_id,
NULL);
END LOOP;

END LOOP;
END;
```

-----<>  
 --testing for 2017 student credit hours

--Testing Student Courses total hours  
 --Results must be maximum 33 Credit hours

```
SELECT b.student_id, SUM (credit_hours)
FROM sr_courses a, (SELECT DISTINCT student_id, course_id
FROM sr_student_learning_outcomes) b, sr_students s
WHERE a.course_id = b.course_id
AND b.student_id = s.student_id
AND s.reg_year = '2015'
```

**GROUP BY** b.student\_id

-----<>

*/\* Formatted on 17-Nov-2017 20:31:32 By Abeer Mousa \*/*  
*/\**  
*Marks for students*  
*Senario One: Marks for random 2 to 8 students - each year- will be excellent in*  
*all topics*  
*Marks for random 2 to 8 students will be so bad- each year*  
*All Other students will take random marks in all subjects*

*\*/*

-----<>

*-- Excelent students each year*

**UPDATE** sr\_student\_learning\_outcomes oc  
**SET** learning\_outcome\_mark = **NULL**;

**DECLARE**  
**CURSOR** students (  
    p\_reg\_year **VARCHAR2**)  
**IS**  
    **SELECT** reg\_year, student\_id, DBMS\_RANDOM.VALUE () rnd  
    **FROM** sr\_students b  
    **WHERE** reg\_year = p\_reg\_year  
    **AND NOT EXISTS**  
        (**SELECT** '\*'  
            **FROM** sr\_student\_learning\_outcomes a  
            **WHERE** learning\_outcome\_mark =  
                learning\_outcome\_mark  
            **AND** a.student\_id = b.student\_id)  
    **ORDER BY** reg\_year, rnd;

**CURSOR** mid\_students (  
    p\_reg\_year **VARCHAR2**)  
**IS**  
    **SELECT** reg\_year, student\_id  
    **FROM** sr\_students b  
    **WHERE** reg\_year = p\_reg\_year  
    **AND NOT EXISTS**  
        (**SELECT** '\*'  
            **FROM** sr\_student\_learning\_outcomes a  
            **WHERE** learning\_outcome\_mark =  
                learning\_outcome\_mark  
            **AND** a.student\_id = b.student\_id);

student\_no **NUMBER**;  
counter **NUMBER** := 0;  
mark **NUMBER**;  
**LOWER** **NUMBER**;

```

higher    NUMBER;
BEGIN
--marks for random 2 to 5 students - each year- will be excellent in all topics
FOR rec IN (SELECT DISTINCT reg_year FROM sr_students)
LOOP
  counter := 0;
  student_no :=
    DBMS_RANDOM.VALUE (2,
                        5);
  LOWER := 91;
  higher := 97;

  FOR rec1 IN students (rec.reg_year)
  LOOP
    FOR rec2 IN (SELECT *
                 FROM sr_student_learning_outcomes
                 WHERE student_id = rec1.student_id)
    LOOP
      mark :=
        ROUND (DBMS_RANDOM.VALUE (LOWER,
                                   higher),
              2);

      UPDATE sr_student_learning_outcomes oc
      SET learning_outcome_mark = mark
      WHERE   oc.student_id = rec2.student_id
      AND oc.course_id = rec2.course_id
      AND oc.learning_outcome_id = rec2.learning_outcome_id;
    END LOOP;

    LOWER := LOWER - 1;
    higher := higher - 1;
    counter := counter + 1;
    EXIT WHEN counter > student_no;
  END LOOP;
END LOOP;

```

-----<>

```

--Marks for random 2 to 5 students will be so bad- each year
FOR rec IN (SELECT DISTINCT reg_year FROM sr_students)
LOOP
  counter := 0;
  student_no :=
    DBMS_RANDOM.VALUE (2,
                        5);
  LOWER := 65;
  higher := 80;

  FOR rec1 IN students (rec.reg_year)
  LOOP
    FOR rec2 IN (SELECT *
                 FROM sr_student_learning_outcomes
                 WHERE student_id = rec1.student_id)
    LOOP
      mark :=

```

```

        ROUND (DBMS_RANDOM.VALUE (LOWER,
                                   higher),
        2);

    UPDATE sr_student_learning_outcomes oc
    SET learning_outcome_mark = mark
    WHERE   oc.student_id = rec2.student_id
            AND oc.course_id = rec2.course_id
            AND oc.learning_outcome_id = rec2.learning_outcome_id;
END LOOP;

    LOWER := LOWER + 0.5;
    higher := higher + 0.5;
    counter := counter + 1;
    EXIT WHEN counter > student_no;
END LOOP;
END LOOP;

--All Other students will take random marks in all subjects
FOR rec IN (SELECT DISTINCT reg_year FROM sr_students)
LOOP
    LOWER := 60;
    higher := 65;

    FOR rec1 IN mid_students (rec.reg_year)
    LOOP
        FOR rec2 IN (SELECT *
                     FROM sr_student_learning_outcomes
                     WHERE student_id = rec1.student_id)
        LOOP
            mark :=
                ROUND (DBMS_RANDOM.VALUE (LOWER,
                                           higher),
                2);

            UPDATE sr_student_learning_outcomes oc
            SET learning_outcome_mark = mark
            WHERE   oc.student_id = rec2.student_id
                    AND oc.course_id = rec2.course_id
                    AND oc.learning_outcome_id = rec2.learning_outcome_id;
            END LOOP;

            LOWER := LOWER + 1;
            higher := higher + 1;
        END LOOP;
    END LOOP;
END;

```

*--Students avreges*

```

SELECT reg_year, student_id, AVG (course_mark)
FROM ( SELECT b.reg_year, a.student_id, a.course_id,
             AVG (a.learning_outcome_mark) course_mark

```

```

        FROM sr_student_learning_outcomes a, sr_students b
        WHERE learning_outcome_mark = learning_outcome_mark
        AND a.student_id = b.student_id
        GROUP BY b.reg_year, a.student_id, a.course_id)
GROUP BY reg_year, student_id
ORDER BY reg_year, "AVG(COURSE_MARK)" DESC;

```

-----<>

--pivot table

```

WITH t
AS (SELECT learning_outcome_name
      FROM (SELECT a.student_id, a.course_id, a.learning_outcome_id,
                  b.learning_outcome_name, a.learning_outcome_mark
              FROM sr_student_learning_outcomes a, sr_learning_outcomes b
              WHERE a.learning_outcome_id = b.learning_outcome_id))
SELECT *
FROM t PIVOT (COUNT (*)
              FOR (learning_outcome_name)
              IN (SELECT DISTINCT d.learning_outcome_name
                  FROM sr_course_learning_outcomes c, sr_learning_outcomes d
                  WHERE c.course_id = 'CSC207H1'
                  AND c.learning_outcome_id = d.learning_outcome_id));

--IN ('LO1_CSC207H1', 'LO2_CSC207H1', 'LO3_CSC207H1', 'LO4_CSC207H1'));

/* Formatted on 7/18/2017 11:33:29 AM (QP5 v5.206) */
-- add random number of learning material from 10 -30 learning material for each
-- learning outcome for each course
-- link a new learning material with one course; with all learning outcomes
--of that course

```

```

DECLARE
    lm_no    NUMBER;
    i        NUMBER;
    counter  NUMBER := 0;
BEGIN
    FOR rec IN (SELECT * FROM sr_courses)
    LOOP
        lm_no := DBMS_RANDOM.VALUE (20, 30);

        FOR i IN 1 .. lm_no
        LOOP
            counter := counter + 1;

            INSERT INTO sr_learning_objects
            VALUES (
                LPAD (counter, 5, 0),
                LPAD (counter, 5, 0)
                || '/'
                || rec.course_id
                || '/'
                || rec.course_name,

```

```

        ' ',
        NULL);

FOR rec2 IN (SELECT learning_outcome_id
              FROM sr_course_learning_outcomes
              WHERE course_id = rec.course_id)
LOOP
    INSERT INTO sr_learning_object_outcomes
        VALUES (rec2.learning_outcome_id, LPAD (counter, 5, 0));
END LOOP;
END LOOP;
END LOOP;
END;

```

-----<>

*-- add random number of learning material from 5-8 learning material  
-- for each random learning outcomes  
--\*\* repeat this script two or three times*

```

DECLARE
    lm_no  NUMBER;
    i      NUMBER;
    counter NUMBER := 0;
BEGIN
    SELECT COUNT (*) INTO counter FROM sr_learning_objects;

    FOR i IN 1 .. 10
    LOOP
        FOR rec IN (SELECT * FROM sr_courses)
        LOOP
            lm_no := DBMS_RANDOM.VALUE (5, 8);

            FOR i IN 1 .. lm_no
            LOOP
                counter := counter + 1;

                INSERT INTO sr_learning_objects
                    VALUES (
                        LPAD (counter, 5, 0),
                        LPAD (counter, 5, 0)
                        || '/'
                        || rec.course_id
                        || '/'
                        || rec.course_name,
                        ' ',
                        NULL);

                --select random 2 numbers of learning outcomes for each course
                FOR rec2 IN ( SELECT *
                              FROM (SELECT learning_outcome_id,
                                    DBMS_RANDOM.random () random
                              FROM sr_course_learning_outcomes

```

```

        WHERE course_id = rec.course_id)
        WHERE ROWNUM < 3
        ORDER BY random)
    LOOP
        INSERT INTO sr_learning_object_outcomes
        VALUES (
            rec2.learning_outcome_id,
            LPAD (counter, 5, 0));
    END LOOP;
END LOOP;
END LOOP;
END;

```

-----<>  
*--A script to give the student learning objects*

```

DECLARE
BEGIN
    FOR rec IN (SELECT * FROM sr_student_learning_outcomes)
    LOOP
        FOR rec2 IN ( SELECT *
            FROM (SELECT b.learning_outcome_id,
                b.learning_object_id,
                DBMS_RANDOM.random () random
            FROM sr_learning_object_outcomes b
            WHERE b.learning_outcome_id =
                rec.learning_outcome_id)
            WHERE ROWNUM < 4
            ORDER BY random)
        LOOP
            BEGIN
                INSERT
                INTO sr_students_learning_objects (student_id,
                    course_id,
                    learning_outcome_id,
                    learning_object_id,
                    rating,
                    hits)
                VALUES (rec.student_id,
                    rec.course_id,
                    rec.learning_outcome_id,
                    rec2.learning_object_id,
                    DBMS_RANDOM.VALUE (1, 5),
                    DBMS_RANDOM.VALUE (0, 20));
            EXCEPTION
                WHEN OTHERS
                THEN
                    NULL;
            END;
        END LOOP;
    END LOOP;
END;

```

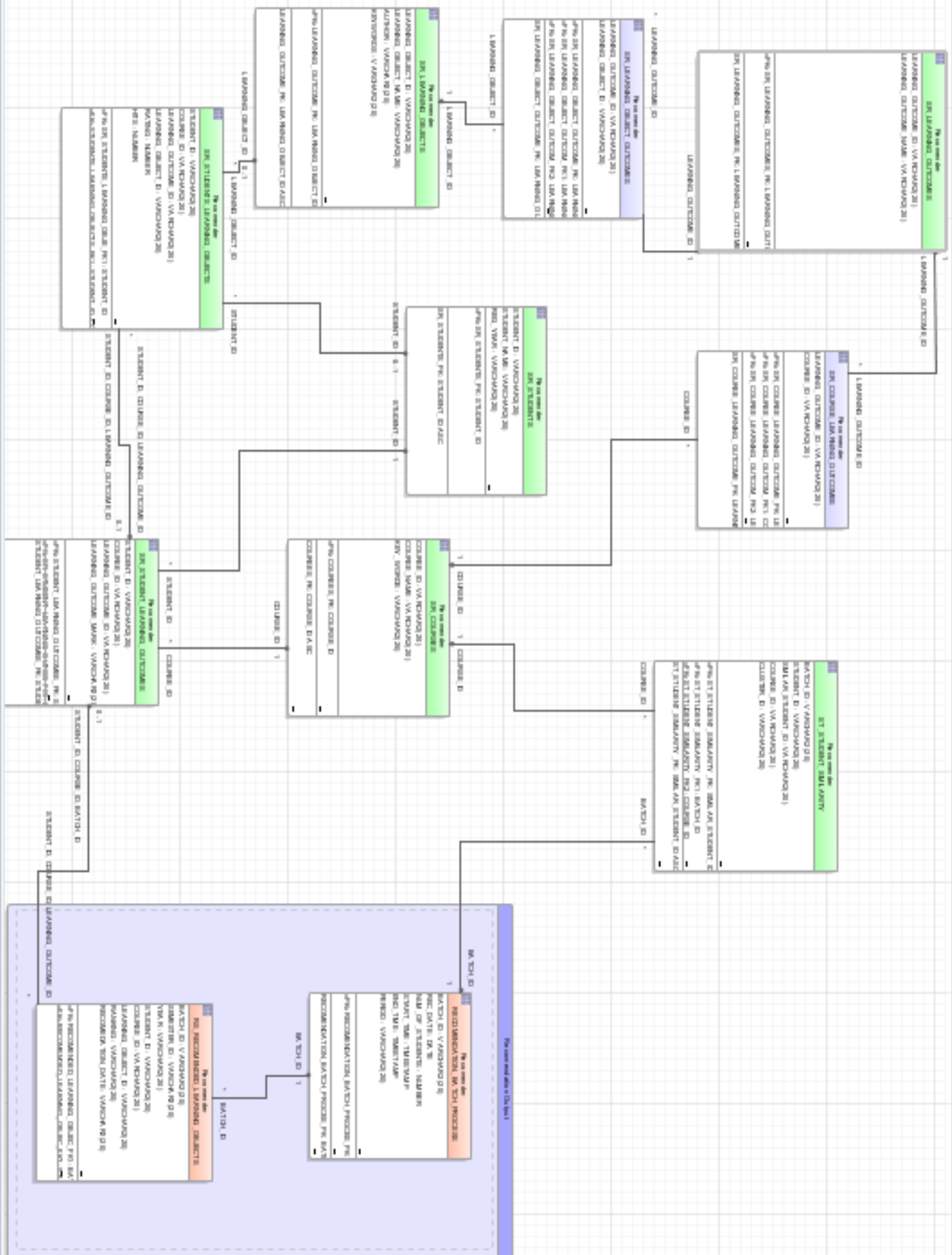


## **Appendix B**

### **Database Design**

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH
RS_RECOMENDED_LEARNING_OBJECTS	BATCH_ID	NUMBER	22
RS_RECOMENDED_LEARNING_OBJECTS	SEMESTER_ID	VARCHAR2	20
RS_RECOMENDED_LEARNING_OBJECTS	YEAR	VARCHAR2	20
RS_RECOMENDED_LEARNING_OBJECTS	STUDENT_ID	VARCHAR2	20
RS_RECOMENDED_LEARNING_OBJECTS	COURSE_ID	VARCHAR2	20
RS_RECOMENDED_LEARNING_OBJECTS	LEARNING_OBJECT_ID	VARCHAR2	20
RS_RECOMENDED_LEARNING_OBJECTS	RANKING	VARCHAR2	20
RS_RECOMENDED_LEARNING_OBJECTS	RECOMEDATION_DATE	VARCHAR2	20
SR_COURSES	COURSE_NAME	VARCHAR2	200
SR_COURSES	COURSE_ID	VARCHAR2	20
SR_COURSES	KEY_WORDS	VARCHAR2	1000
SR_COURSES	LEVEL_TYPE	VARCHAR2	2
SR_COURSES	ACADIMIC_YEAR	VARCHAR2	2
SR_COURSES	CREDIT_HOURS	NUMBER	22
SR_COURSES	SEQ_ID	NUMBER	22
SR_COURSE_LEARNING_OUTCOMES	COURSE_ID	VARCHAR2	20
SR_COURSE_LEARNING_OUTCOMES	LO_COURSE_ORDER	NUMBER	22
SR_COURSE_LEARNING_OUTCOMES	LEARNING_OUTCOME_ID	VARCHAR2	20
SR_LEARNING_OBJECTS	LEARNING_OBJECT_NAME	VARCHAR2	200
SR_LEARNING_OBJECTS	LEARNING_OBJECT_ID	VARCHAR2	20
SR_LEARNING_OBJECTS	AUTHOR	VARCHAR2	20
SR_LEARNING_OBJECTS	KEYWORDS	VARCHAR2	20
SR_LEARNING_OBJECT_OUTCOMES	LEARNING_OBJECT_ID	VARCHAR2	20
SR_LEARNING_OBJECT_OUTCOMES	LEARNING_OUTCOME_ID	VARCHAR2	20
SR_LEARNING_OUTCOMES	LEARNING_OUTCOME_NAME	VARCHAR2	20
SR_LEARNING_OUTCOMES	LEARNING_OUTCOME_ID	VARCHAR2	20
SR_LEARNING_OUTCOMES	LEARNING_OUTCOME_LEVEL	NUMBER	22
SR_STUDENTS	STUDENT_ID	VARCHAR2	20
SR_STUDENTS	STUDENT_NAME	VARCHAR2	200
SR_STUDENTS	REG_YEAR	VARCHAR2	20
SR_STUDENTS	GRADUATED_STATUS	VARCHAR2	2
SR_STUDENTS_LEARNING_OBJECTS	STUDENT_ID	VARCHAR2	20
SR_STUDENTS_LEARNING_OBJECTS	LEARNING_OUTCOME_ID	VARCHAR2	20
SR_STUDENTS_LEARNING_OBJECTS	COURSE_ID	VARCHAR2	20
SR_STUDENTS_LEARNING_OBJECTS	HITS	NUMBER	22
SR_STUDENTS_LEARNING_OBJECTS	RATING	NUMBER	22
SR_STUDENTS_LEARNING_OBJECTS	LEARNING_OBJECT_ID	VARCHAR2	20
SR_STUDENT_LEARNING_OUTCOMES	LEARNING_OUTCOME_ID	VARCHAR2	20
SR_STUDENT_LEARNING_OUTCOMES	COURSE_ID	VARCHAR2	20
SR_STUDENT_LEARNING_OUTCOMES	STUDENT_ID	VARCHAR2	20
SR_STUDENT_LEARNING_OUTCOMES	LEARNING_OUTCOME_MARK	NUMBER	22
ST_BATCH_PROCESS	END_TIME	TIMESTAMP(6)	11
ST_BATCH_PROCESS	START_TIME	TIMESTAMP(6)	11
ST_BATCH_PROCESS	BATCH_ID	NUMBER	22
ST_BATCH_PROCESS	NUM_OF_STUDENTS	NUMBER	22
ST_BATCH_PROCESS	REC_DATE	DATE	7
ST_BATCH_PROCESS	PERIOD	VARCHAR2	20

ST_BATCH_PROCESS_DTL	COURSE_ID	VARCHAR2	20
ST_BATCH_PROCESS_DTL	K	NUMBER	22
ST_BATCH_PROCESS_DTL	MATRIX_TYPE	VARCHAR2	50
ST_BATCH_PROCESS_DTL	MAX_MEMORY	VARCHAR2	50
ST_BATCH_PROCESS_DTL	SSE	NUMBER	22
ST_BATCH_PROCESS_DTL	MS_TIME	NUMBER	22
ST_BATCH_PROCESS_DTL	DISTANCE_FUNCTION	VARCHAR2	50
ST_BATCH_PROCESS_DTL	STUDENT_ID	VARCHAR2	20
ST_BATCH_PROCESS_DTL	BATCH_ID	NUMBER	22
ST_BATCH_PROCESS_DTL	PROCESS	VARCHAR2	20
ST_BATCH_PROCESS_VW	DETAILS	NUMBER	22
ST_BATCH_PROCESS_VW	MATRIX_TYPE	VARCHAR2	13
ST_BATCH_PROCESS_VW	K	NUMBER	22
ST_BATCH_PROCESS_VW	DISTANCE_FUNCTION	VARCHAR2	50
ST_BATCH_PROCESS_VW	BATCH_ID	NUMBER	22
ST_STUDENT_SIMILARITY	CLUSTER_ID	VARCHAR2	20
ST_STUDENT_SIMILARITY	COURSE_ID	VARCHAR2	20
ST_STUDENT_SIMILARITY	SIMILAR_STUDENT_ID	VARCHAR2	20
ST_STUDENT_SIMILARITY	STUDENT_ID	VARCHAR2	20
ST_STUDENT_SIMILARITY	BATCH_ID	NUMBER	22



إعداد نظام توصية لبيئة تعليمية اعتماداً على تحصيل الطلاب في مخرجات عملية التعلم

إعداد: عبير حسن عبد الرحيم موسى

إشراف: د. بديع سرطاوي

## الملخص

تتفاوت المستويات المعرفية بشكل عام للطلاب في أي بيئة تعليمية، ويتجلى ذلك التفاوت من خلال نتائج التعلم المحققة ونمط التعلم، كما تختلف طرقهم في مواجهة الصعوبات وحل المشكلات ومعالجتها، في حال لم تلبي النتائج توقعاتهم، فعندما يستخدم الطالب الشبكة العنكبوتية كمحاولة للبحث عن حل لمشكلة ما؛ فإنه يعاني من وجود كم هائل من الموارد التي لا تحقق احتياجاته في كثير من الأحيان، أو قد تكون ذات صلة ولكنها معقدة ومتقدمة أو بسيطة جداً. مما يؤدي الى تشتته واضاعة وقته وقد يكون لها آثار سلبية على إنجازاته في بعض الحالات، حيث ينحرف مسار البحث إلى جوانب هامشية وقليلة الفائدة؛ وانطلاقاً من تلك المعطيات تأتي الحاجة إلى نظام تعلم ذكي يقوم بتوجيه الطلاب بناءً على احتياجاتهم.

يهدف هذا البحث إلى تصميم وبناء نظام توصية لبيئة تعليمية يوصي بمواد تعليمية تلئم احتياجات الطلاب وفقاً لمستواهم المعرفي وتحصيلاتهم، مرتبة من الأسهل إلى الأكثر صعوبة، ويمكن تحقيق ذلك بواسطة استكشاف نمط التعلم لدى الطلاب الآخرين والاستفادة من تجارب الطلاب المتماثلة في المستوى المعرفي و الحاصلين على علامات متميزة، وعليه فقد اقترحت الدراسة تصميمًا لنظام توصية هجين يتكون من منهجين للتوصية: منهج قائم على المحتوى العام (Content base) فيما المنهج الآخر قام على الترشيح التعاوني (Collaborative approach).

وتركز الدراسة على منهج التوصية التعاوني الذي يوصي بمواد تعليمية اعتماداً على المادة التعليمية وعلى المستوى المعرفي للطلاب وتقييم تحصيلاتهم في مخرجات التعلم ومستويات مخرجات التعلم وذلك بالاستفادة من تجارب طلبة سابقين مشابهين لهم في المستوى المعرفي و حاصلين على علامات مميزة في المادة التعليمية، وقد استخدم في البحث خوارزمية k-mean لمعرفة الطلاب المتشابهين، كما و تم استخدام خمسة وظائف للقياس التشابه تتمثل بـ (المسافة الاقليدية والارتباط والجاكارد إضافة لجيب التمام ومانهاتن) واقد أظهرت النتائج ان جيب التمام اعطى أدق القياسات للمسافة بأقل نسبة خطأ، ولكن أعلى زمن للمعالجة والذي لا يختلف كثيراً عند مقارنته مع الوظائف الاخرى، تم أيضاً تحديد أفضل عدد

من المجموعات الإحصائية (Clusters) للدراسة باستخدام ثلاث أساليب تتمثل بـ (Elbow, Gap- statistic and average Silhouette) حيث تبين أن أفضل عدد من المجموعات يكون ثلاث، وقد استخدم البحث مصفوفتي لدراسة نمط التعلم عند الطلاب: مصفوفة الطلبة المتميزين بشكل عام، و مصفوفة الطلبة المتميزين والمتشابهين للطلبة الحاليين؛ من أجل حساب أوزان المواد التعليمية وترتيبها بناءً على أعلى الأوزان، التي تؤدي إلى التوصيات النهائية المطلوبة والتي يعرضها محرك النظام لتحسين نتائج الطالب.

