



University of Granada

Doctoral Thesis

Modelling Course Difficulty Indexes to Enhance Students Performance and Course Study Plans

Doctoral Program in Information and Communication Technologies

Author:

Mohammed Al-Twijri

Thesis Directors:

Dr. Sebastián Ventura Soto

Dr. Francisco Herrera Triguero

Granada, May 2022

Editor: Universidad de Granada. Tesis Doctorales
Autor: Mohammed Al-Twijri
ISBN: 978-84-1117-429-9
URI: <http://hdl.handle.net/10481/75974>

Acknowledgments

There are many whom I must thank for, but there are those who I need to specially thank.

I Thank **ALLAH**, the Lord of the Worlds, first and foremost. May Allah accept my humble efforts and make this thesis helpful and usable to many people.

To my wonderful and precious **Father**, and to my wonderful and precious **Mather**, all the love and respect for his guidance, advice, and support throughout my life, for his unconditional love. All success comes to my life because you are with me.

To my supervisor Professor **Francisco Herrera** and special thanks to my supervisor Professor **Sebastián Ventura**, for his superior advice throughout the thesis progress, and for his cooperative effort and patience.

To my beloved brothers, and my sisters, you make life wonderful and always give me hope in tomorrow, thank you because you sustained me during the sad and happy moments in my life, thank you for your direct and indirect support.

To my wife, my sons Ibrahim and Jaser who gave me a lot of sacrifices to create a distinct academic environment, Thank you.

To all my family and my friends, your belief in me, your constant support, encouragement, and cooperation all times, when I am up and when I am down, your pray for me and help guide me to the way of success.

To Dr. Mahmoud Alsehetry for his technical support, advice, help and endless support and his time.

To every person who gave me even the tiniest advice or help, believed in me and remembered me in their prayers. Thank you so much.

Resumen

El objetivo general de esta tesis ha sido abordar la tarea de la planificación de cursos a largo plazo (*Long Term Course Planning – LTCP*), asesorando a los estudiantes para que elijan el plan de aprendizaje que más les convenga, y reduciendo la tasa de abandono. Este objetivo general se desglosa en una serie de subobjetivos tal y como se describe:

- Proponer un enfoque de minería de patrones secuenciales que analice, de forma descriptiva, diferentes planes de estudio para estudiantes similares. Este plan de estudios viene dado por la consideración de los datos históricos de los alumnos con buenas notas en cuanto a cursos y notas medias de la titulación.
- Presentar un índice de dificultad del curso para medir la elegibilidad de un curso específico. Se considera un valor máximo de dificultad para que los estudiantes hagan las elecciones de acuerdo con la métrica proporcionada.
- Proponer una aplicación web para que sea posible obtener el valor del índice de dificultad para diferentes asignaturas de diferentes titulaciones.
- Aplicar los subobjetivos anteriores a un problema real con información de la King Abdulaziz University (KAU), una de las universidades más importantes de Arabia Saudí, ubicada en la ciudad de Jeddah.

Los datos analizados procedían de múltiples fuentes en entornos heterogéneos como el sistema ODUSPLUS que utiliza la base de datos Oracle, el sistema ANJEZ que utiliza la base de datos DB2, NOOR, ENTEMAA y ESTEBANA que almacenan los datos en la base de datos SQL.

En cuanto al desarrollo de la tesis, se ha comenzado con una revisión del estado del arte de las temáticas de interés, a saber: minería de patrones en educación, minería de currículos académicos, sistemas de personalización, etc. Con esta visión en mente se ha planteado la hipótesis general de que las técnicas de extracción de conocimiento pueden ayudar a la obtención de nuevos métodos que ayuden al desarrollo de sistemas de apoyo a la decisión aplicados a LTCP. En este sentido, se han desarrollado dos propuestas para personalización de currículos:

La primera se basa en minería de patrones secuenciales para descubrir qué secuencias de materias son las que conducen a perfiles de éxito (mejores calificaciones en la titulación). Esta contribución ha consistido en el desarrollo un algoritmo evolutivo especialmente diseñado

para analizar secuencias de cursos que diferencien claramente los alumnos con notas excelentes del resto. Estas secuencias permiten establecer unos índices de dificultad para recomendar asignaturas según las asignaturas cursadas previamente.

El enfoque arrojó excelentes resultados en el caso del estudio produciendo secuencias interesantes para cada alumno concreto en función de los cursos anteriores que ya había superado. La recomendación se hizo en función de los cursos ya realizados por estudiantes similares y obteniendo un excelente promedio final. Además, esta metodología fue capaz de proporcionar planes de estudio completos desde las primeras etapas de la carrera, lo cual es importante para los nuevos estudiantes. Además, se ha propuesto una métrica de índice de dificultad de los cursos y se utiliza una aplicación online para describir qué cursos son más difíciles para que los estudiantes puedan elegir diferentes cursos según un valor de dificultad máximo.

La segunda propuesta consistió en el cálculo de un índice de dificultad de cursos, denominado DMDIM, el cual se puede utilizar para llevar a cabo una estimación adecuada de la carga que supone para los estudiantes la elección de un determinado conjunto de asignaturas. La memoria muestra las hipótesis que han conducido a la obtención de este índice, ilustrándolo con varios ejemplos asociados a los datos académicos de la KAU. Los resultados muestran que el índice obtenido es una gran ayuda para aconsejar a los estudiantes sobre la carga de asignaturas a elegir durante un curso académico. Asimismo, sirve para que los responsables de organización académica puedan diseñar los itinerarios para mejorar el rendimiento de los alumnos en general. Esta segunda propuesta se ha implementado en una aplicación web que permite llevar a cabo el cálculo de todos los índices de dificultad de cursos de forma automática, permitiendo a los docentes y/o estudiantes obtener información sobre la carga docente que tiene la elección de un determinado bloque de asignaturas.

Además de los resultados alcanzados en los experimentos planteados cabe indicar que, como resultado general, a la hora de diseñar los planes de estudio de los departamentos, toda institución educativa no debería basarse únicamente en las horas de crédito o en las unidades, sino que también debería tener en cuenta nuevos factor como es el índice de dificultad del curso o la secuencia correcta de actividades.

Abstract

The overall aim of this thesis has been to address the task of Long-Term Course Planning (LTCP), advising students to choose the learning plan that suits them best, and reducing the drop-out rate. This overall objective is broken down into several sub-objectives as described:

- To propose a sequential pattern mining approach that analyses, in a descriptive way, different study plans for similar students. This study plan considers historical data of students with good grades in terms of courses and average grades of the degree.
- To present a course difficulty index to measure the eligibility of a specific course. A maximum difficulty value is considered for students to make choices according to the metric provided.
- To propose a web application to make it possible to obtain the value of the difficulty index for different subjects of different degrees.
- To apply the above sub-objectives to a real problem with data from King Abdulaziz University (KAU), one of the most important universities in Saudi Arabia, located in Jeddah.

The analyzed data came from multiple sources from heterogeneous environments such as the ODUSPLUS system using the Oracle database, the ANJEZ system using the DB2 database, NOOR, ENTEMAA and ESTEBANA storing the data in the SQL database.

As for the development of the thesis, it has started with a review of the state of the art of the topics of interest, namely: pattern mining in education, mining of academic curricula, personalization systems, etc. With this vision in mind, the general hypothesis has been put forward that knowledge extraction techniques can help obtain new methods that aid the development of decision support systems applied to LTCP. In this sense, two proposals for curriculum personalization have been developed:

The first one is based on sequential pattern mining to discover which sequences of subjects are the ones that lead to successful profiles (better grades in the degree). This contribution has consisted of developing an evolutionary algorithm specially designed to analyse sequences of courses that differentiate students with excellent rates from the rest. Furthermore, these sequences establish difficulty indices to recommend subjects according to the previously taken issues.

The approach yielded excellent results in the case study, producing interesting sequences for each student based on the previous courses they had already passed. The recommendation was based on the courses already taken by similar students, and an excellent final average was obtained. In addition, this methodology was able to provide complete study plans from the early stages of the course, which is essential for new students. A course difficulty index metric has been also proposed, and an online application is used to describe which courses are more difficult so that students can choose different courses according to a maximum difficulty value.

The second proposal consisted of calculating a course difficulty index, called DMDIM, which can be used to carry out a proper estimation of the burden for students to choose a given set of subjects. The report shows the hypotheses that have led to the derivation of this index, illustrating it with several examples associated with KAU academic data. The results show that the index obtained is an excellent help in advising students on the load of subjects to choose from during an academic year. It also helps those responsible for educational organization design pathways to improve student performance. This second proposal has been implemented in a web application that automatically calculates all course difficulty indices, enabling teachers and/or students to obtain information on the teaching load of a given block of subjects to be chosen.

In addition to the results achieved in the experiments, it should be noted that, as a general result, when designing departmental curricula, any educational institution should not only rely on credit hours or units but should also consider new factors such as the course difficulty index or the correct sequence of activities.

Index

CHAPTER I. Context	1
1. Long-Term Course Planning.....	2
2. Educational Data Mining.....	5
3. Research Hypothesis And Objectives.....	7
4. Organization of the Document	8
CHAPTER II. Literature Review	9
1. Recommendation Systems in Education	9
1.1. Pattern Mining in Education	10
1.2. Curriculum Mining	12
1.3. Curriculum Personalization Systems.....	13
1.3.1. Goal Setting.....	13
1.3.2. Feedback.....	14
1.3.3. Periodic Formative Assessment	14
1.3.4. Deliberate Practice	15
1.3.5. Peer Tutoring	15
1.4. Course Difficulty Index.....	16
2. Supervised Learning in Education.....	17
3. Literature Review Summary	24
CHAPTER III. Applying the CRISP-DM Methodology to KAU Student Information Data	27
1. Methodology	27
1.1. Research and Business Understanding.....	28
1.2. Data Understanding.....	29
1.3. Data Preparation.....	30
1.4. Modelling.....	32
1.5. Evaluation	32
1.6. Deployment	33
2. Applying CRISP-DM to KAU Data. A Case of Study.....	33
2.1. Data Understanding.....	34
2.2. Data Preparation.....	49

CHAPTER IV. A Sequential Pattern Mining Approach	53
1. Introduction	53
2. An Evolutionary Algorithm for Searching Emerging Sequential Patterns	54
2.1. Data representation	54
2.2. Encoding criterion	57
2.3. Algorithm	57
3. Resulting Set of Solutions	63
4. Experimental Study	65
4.1. Experimental Setup	65
4.2. Analysis of the Proposal	66
4.3. Comparative Analysis	69
4.4. Top-K Sequential Pattern Mining Algorithms	73
5. Study Case	74
5.1. Study Plan Recommendation Based on the Best Ordering of Courses	74
5.2. Course Recommendation Based on the Previous Academic Path	75
CHAPTER V. DMDIM: A Method to Calculate a Course Difficulty Index. Validation on KAU Student Data ...	79
1. Proposed Difficulty Index	80
2. DMDIM Evaluation	81
2.1. Course Difficulty Index Calculation (CDIC)	82
2.2. General Weight for Factor (GWF)	83
2.3. General Weight for Course (GWC)	86
2.4. General Weight for Plan (GWP)	89
3. DMDIM Environment	93
4. Remarks and Conclusions	105
CHAPTER VI. Conclusions and Future Work	107
1. Conclusions	107
2. Future Work	108
References	109

CHAPTER I. Context

Education is the key factor in a growing society. The learning process can be defined as the way in which people acquire knowledge, values, morals, habits, and skills. It is a process of personal development, and the way knowledge is transmitted from one generation to the next. Its importance is so high that education has turned into the initial step for each human action, and schooling is the basis for any developed country [KASJ14]. Schooling empowers people to construct more prosperous and effective lives and social orders to accomplish financial thriving and social government assistance. The number of students receiving formal education and the length of that formal education have increased dramatically over the last 5,000 years[Mour05].

In the last few years, Information Technology (IT) has emerged as a key factor in business, creating, processing, storing, retrieving, and exchanging data and information from the business domain. Admittance to the Internet is basic to the rise of IT in multiple domains, and Education is one of such domains where IT is playing a significantly important role [HMRJ11]. The use of the Internet in Education has opened entryways to an abundance of data, information, and instructive assets, expanding open doors for learning in and past the homeroom. Instructors utilize online materials to get ready illustrations, and understudies to expand their scope of learning [RoVe20]. All of this has given rise to a new Education paradigm usually known as *Blended Learning*, where in-person classes is not the only way of learning [GaKa04]. Now, the learning process is carried out by not only face-to-face classes but also using e-learning tools thanks to learning platforms or course management systems. This new way of learning, where a student does not need to be physically in class, has opened Education to anyone having Internet access. Therefore, blended learning is a type of learning that combines remote and face-to-face teaching with the only objective of bringing together the best of both worlds to achieve more efficient learning [ChCR21]. There are four main modalities of blended learning: students rotate between different learning modes; teachers instruct and support on-demand, mainly through online platforms; the students decide which subjects will be taught in person or online; students require a certain number of hours of face-to-face learning.

Nevertheless, students are not the only actors who have gained something with IT but also educators and administrative staff. Thanks to IT, educators are able to obtain information for each specific students and to provide them with the right assistance. Education has turned into personalized education [FiVä11] where the contents and advises are customized according to each student's strengths, needs, skills, and interests. Each student gets a learning plan that is based on what they know and how they learn best.

1. Long-Term Course Planning

One of the aims of personalized education is to advise each student to choose the right learning plan [ToPa18], which is based on their knowledge and how they learn best. Long-Term Course Planning (LTCP) is an essential task in academic advising [GSOC20], and it aims to help students in proposing a course list of all future semesters so the dropout rate might be reduced. LTCP is particularly thought-provoking due to many reasons such as the number of constraints related to university regulations, the student's abilities, and background knowledge, or simply some personal preferences caused by external factors [NLRV16]. As a result, when building a study plan, a lot of different aspects should be considered to prioritize courses. Some of such aspects quantify the importance of including a specific course in the study plan (students' preferences, expected grades due to easy courses, etc.), whereas others rate the chronology of those courses (complexity of the semester based on the courses).

A good decision support system for academic advising, working as LTCP, is essential at university as an instrument to guarantee the fulfilment of the instructive necessities without confronting pointless deferrals [NLRV16]. This kind of systems are excellent to avoid failing in extreme credits per semester, minimizing troubles in the learning process, increasing the graduation per semester, and many more. LTCP is especially useful at university, where a credit hours system is usually considered, and many courses are opened for selection including of elective courses and mandatory courses. The former is those that can be chosen by the students from several optional courses in a curriculum, while later must be taken by the students. Mandatory courses are essential for an academic degree. Elective courses, on the contrary, to be more specialized. Sometimes, each course either mandatory or elective presents some prerequisite courses or co-requisite courses that should be passed before moving on the following courses. As a result, a course study plan (CSP) is a challenging problem for many several reasons (see Figure 1.1).



Figure 1.1: CSP Problems

One of the main problems when building a good CSP is the existing constraints [RiTs20], which are related to university regulations and students' abilities. Two main types of constraints are considered here. Hard constraints, which are conditions that must never be violated, such as registration rules. Soft constraints, which are conditions that could be fulfilled at various levels of satisfaction, such as course priorities. An additional problem when building a CSP is related to the course priority. Many different criteria could be used to prioritize courses when building a study plan, and these priorities can be fixed according to several factors: Importance (it determines which elective courses to include in the CSP according to the students' interests. No mandatory courses are considered here); Grade (it indicates the minimum expected grade for an unfinished course. Again, this factor is only considered in elective courses); Chronology (it determines when either mandatory or elective courses should be taken). Another important problem to be considered in a CSP is the course dependencies. Three types of dependencies must be considered: precedence, succession, and concurrency. Precedence courses are related to the fact that many advanced courses have prerequisite courses that a student must finish before registering the advanced course. Successive courses are pairs of courses that should be given in two consecutive semesters because the later course is a continuation of, or strongly dependent on, the earlier one. Concurrent courses are those ones that should be assigned to the same semester because they complement each other. Finally, hard constraints are also of vital importance when building a good CSP. These are constraints that must be fulfilled when assigning courses to semesters in the CSP, and they are of many different types:

- Must-fulfil the degree requirements. The student must pass a certain set of courses to fulfil one of the degree requirements. These requirements come in various levels such as University requirements, College requirements and Department requirements. It is important to highlight that all these levels could be different from a university to another according to its program structure. Academic department in each college is responsible for developing its own academic course study plan (Bachelor, Master, or PhD).
- Must take elective courses. There are some obligations in this regard.
- Course availability. Some courses may be offered in specific semesters but not in others.
- Limit of credit hours and number of courses per semester. Some universities place a limit on both the credit-hour load per semester and the number of courses to be taken per semester
- GPA (Grade Point Average) per semester. All students are interested in how to get the highest GPA. Students who have low cumulative GPA might be at risk of dismissal if their unsatisfactory work continues for several successive semesters.

- Budget per semester. There is a limitation in the budget of the students to pay their tuition fees.
- Student leave: Universities usually encourage their students to maintain a continuous registration in an academic program. Therefore, it is necessary to develop techniques that provide appropriate support for both academic advisors and students to reduce uncertainties relevant to estimating the expected grades for students. Instead of relying on students to estimate their future grades (even with the help of their advisors), statistical analysis of students' academic history could improve our expectations of a student's future grades. A technique is also required to provide justification about course assignments. Devising a technique that increases our understanding of the generated course plans, and hence improve the process of stepwise refinement of the input as well as reduce uncertainties related to our choices.

All the above demonstrates that LTCP is a really hard task [ToPa18], and it is important to design and develop approaches that can help both academic mentors and students to improve the expectations of a student's future grades. Statistical analysis of students' academic history and combining it with their advisors' opinions are essential to this aim. The general objective is to enhance the precision and reduce uncertainty relevant to estimate expected grades for students. This should lead to improved advising and better decision-making by faculty and administrators in response to student effort, performance, and instruction. It also enables to provide any justification about course assignments. With all of this in mind, the analysis of educational data [RoVe10] may provide the right methods for exploring any the data and to provide accurate and meaningful study plans for each student according to their requirements and needs.

Finally, it is important to remark the importance of Education and LTCP for many institutions and countries around the world. As an example, the budget of the Ministry of Education for one year in the Kingdom of Saudi Arabia is about 88 billion. The total number of students in higher education is around one million and two hundred thousand. Each student's cost for a bachelor's degree is about \$ 80,000. King Abdulaziz University (KAU), one of the most important Universities in Saudi Arabia, is paying special attention to LTCP. This University, which includes about 347,991 undergraduate students and about 23,723 postgraduate students, presents some major problems with the students' graduation plans. About 46%, 50% of the students exceeded the graduation period respectively in 2018, 2017 as shown in Figure 1.2. About 11%, 15% drops in some subjects in the semester respectively in 2018, 2017 as shown in Figure 1.3. About 91%, 96% of students registered for the summer course respectively in 2018, 2017 as shown in Figure 1.4. About 97%, 98% of students broke the university course plan respectively in 2018, 2017 as shown in Figure 1.5. All these queries show big issues according to some errors in course sequence recommendations and lead to a huge load in the university budget and students' ability.

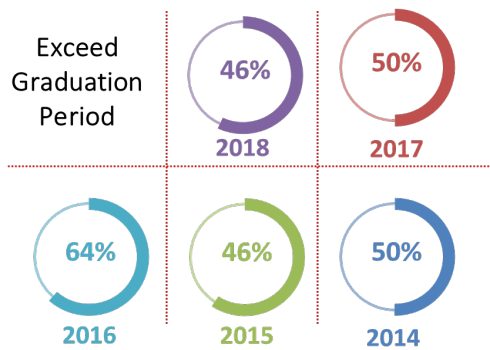


Figure 1.2 Extended Graduation Period

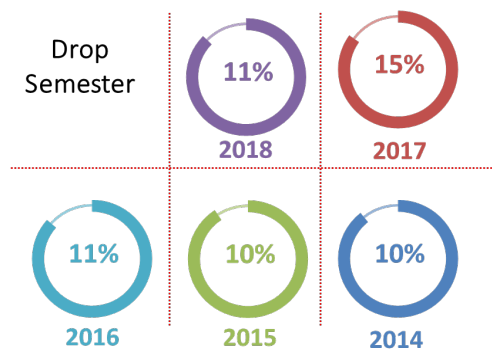


Figure 1.3 Semester Subjects Drop

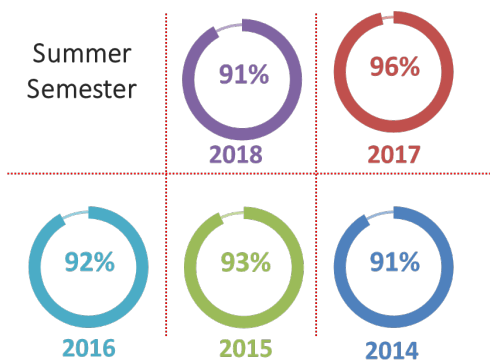


Figure 1.4 Summer Course Registration

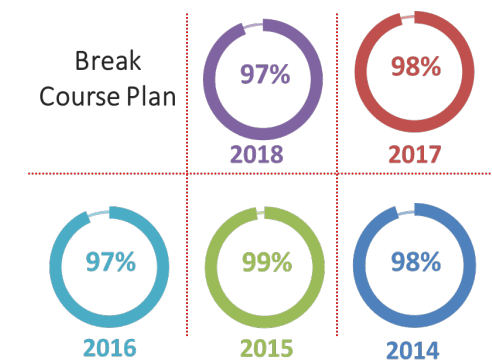


Figure 1.5 Semester Break Course Plan

2. Educational Data Mining

As previously described, IT has played a significantly important role in education for the last years [HMRJ11]. The use of the Internet in Education has given rise to a new Education paradigm known as blended learning, where in-person classes appear together with e-learning tools. In this new type of education, the critical challenge for all educational institutions is improving student performance. It has given rise to an emerging discipline, known as Educational Data Mining (EDM) [RoVe20]. Many different definitions of EDM can be found in the specialized literature, but they all agree in terms of it is an interdisciplinary research area which uses different methods and techniques from machine learning, statistics, data mining and data analysis, to analyze data that come from education settings (see Figure 6). EDM uses these methods to better understand students and improve their learning processes. Educational data come from students' use of interactive learning environments, computer-supported collaborative learning, or administrative data [LFPR22]. EDM assists in analyzing the data gathered and generating intelligence by applying

data analysis strategies. The final aim of EDM is the improvement of any educational process, providing an accurate explanation about educational strategies for better decision making. Methods associate to EDM are useful to improve the study process [NLRV16], increase course completion, assist students in course selection, perform students' profiling and targeting, identify problems that lead to dropout, curriculum development, predict student performance, among others.

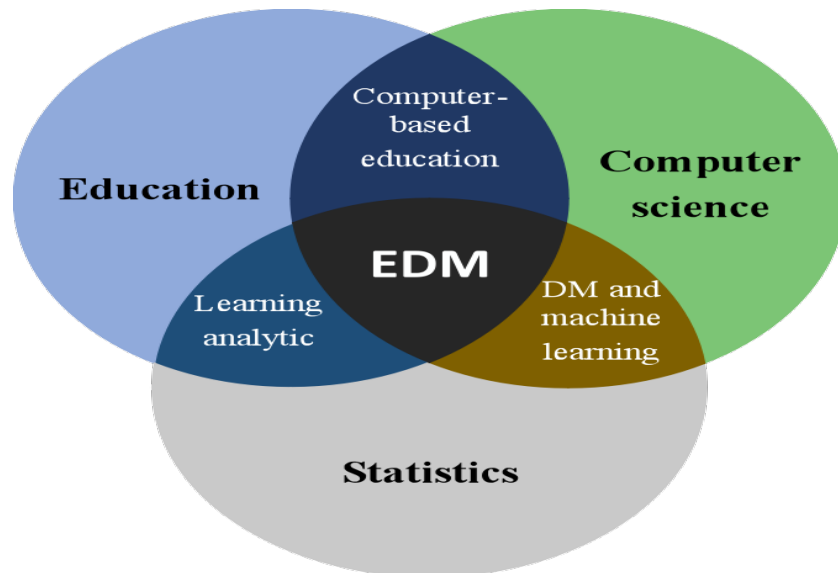


Figure 1.6: Main areas related to Educational Data Mining / Learning Analytics

Nowadays, most education institutions use e-learning systems, and some of them are also considering learning analytics. An example is higher education (HE), which is starting to use this kind of analytics to upgrade the services they provide and to enhance measurable and visible targets that help all the agents involved in the educational process to optimize the performance of studies (improving final grades, reputation, difficulty, and other relevant actors you want to optimize). EDM is solving most of the claims of different education facets. Yu, Shaoying [Yu15], reported that it was important to realize and enhance the performance of HE. Identification of the most important criteria that influence in the quality of academic performance is a very complex and challenging issue [KASJ14]. Also, there is need for transferring the theoretical knowledge to empirical research and practice, which is particularly important for the development of students in their careers [YiBa14]. [MeMa14, RSGS13] said that the evaluation of students' performance is considered an important issue since it has a great impact on the student approach to learning and their outcomes. Also, the correctness of student performance prediction is vital in various contexts in the educational process [DoCh13]. In general, there is a need to upgrade the scholarly exhibition and improve the educational quality [YiBG13].

3. Research Hypothesis And Objectives

The previous sections have illustrated that LTCP is a new task of great interest, which enhances the adaptability of studies so that students can take better advantage of the opportunities offered by new educational technologies. It has also been commented that educational data mining techniques enable the extraction of new knowledge that improves teaching and learning processes in general. These facts lead us to pose our research hypothesis, namely, *the use of data mining techniques, essentially descriptive analysis, may provide useful information to define different learning plans*. Based on historical data, it is possible to find students with similar characteristics so accurate course planning may be provided.

Once this research hypothesis is presented, it is important to highlight that the general objective of this thesis is to deal with the LTCP task, advising students to choose the learning plan that best suits them, and reducing the dropout rate. This general objective is broken down into a series of subobjectives as it is described:

- To conduct a literature review in the lines of research directly related to LTCP, namely: pattern mining, curriculum mining, and curriculum personalisation systems. The idea is to contextualise the work done so far to find out what alternatives we can propose that are interesting and competitive.
- To propose the best possible methodological approach to carry out this research as a knowledge extraction process that will use data from the King Abdulaziz University (KAU) as input, describing the different tasks that comprise the previous operations, to leave ready a set of data that will be used as input for the knowledge discovery algorithms developed in the subsequent parts of this research.
- To propose a sequential pattern mining approach that analyse, in a descriptive way, different study plans for similar students. This course plan is given by considering historical data of students with good GPAs in terms of courses and average marks of the degree.
- To present a course difficulty index to measure the eligibility of a specific course. A maximum difficulty value is considered so students should make elections according to the provided metrics.
- To propose a web application so it is possible to obtain the difficulty index value for different subjects of different degrees.

4. Organization of the Document

This Dissertation consists of six chapters and one appendix section as follows:

Chapter I is this chapter that has presented a brief introduction to the problem to be tackled, the concepts of long-term course planning and educational data mining, and the objectives and work hypothesis of this thesis.

Chapter II is concerned with a literature review of the most important related tasks. It includes tasks applied to Education such as recommendation systems in Education, pattern mining, curriculum mining, curriculum personalization systems and supervised learning.

Chapter III addresses the dataset environment and data modelling. It describes the development of the methodology CRISP to prepare the available information to be used in the research developed in Chapters IV and V. The chapter will finish with a descriptive analysis of the resulting information.

Chapter IV describes a Sequential Pattern Mining approach for academic advising based on student's preferences, complexity of the semester, even background knowledge, etc. Besides presenting the algorithm proposal, the chapter shows the results obtained with the available data from the KAU database.

Chapter V presents a novel Difficulty Index Methodology as a technique to calculate the course difficulty index for improving the student efficiency and performance and recommending each student a proper course plan. A presentation on the five traditional methods that students use to register a course and select the right balance of courses on one semester is presented.

Chapter VI outlines the Dissertations' conclusions and future works.

CHAPTER II. Literature Review

This chapter presents a literature review of the main research areas related to this Thesis. First, it focusses on Recommendation Systems in Education (RSE). Then, it describes the main research works related to Pattern Mining (SPM), Curriculum Mining (CM), Curriculum Personalization Systems (CPS) and Supervised Learning in Education. Finally, it provides a summary and some conclusions about this literature review.

1. Recommendation Systems in Education

Recommendation Systems (RS) have been emerged as a useful technique to guide to the users in different domains where there is a vast amount of information available, such e-commerce, music, or movies [BOHG13]. Within this field, the most common technique is the Collaborative Filtering (CF), based in similar users, follows by Content-based Filtering (CBF), based in similar items. Moreover, hybrid techniques are increasing their use because they take advantages of different models. A very successful application of EDM techniques is the development of Educational RS. One of the firsts applications in the field can be found in [Lu04], that explore the extraction of students' learning requirements and use matching rules to generate personalized recommendations of learning activities in a context of e-learning environments. Since then, these systems have been applied to a broad domain, ranging from automatic suggestions for the assignment of courses timetables and classrooms [MiRR12], to recommendations for creation of a long-term course planning that consider constraints concerning to both student and courses [Moha15]. In this context, RSs have been thoroughly applied to the problem of course recommendation from different approaches. Recently, [IaKF17] present a systematic review of most recent RSs applied to course selection from an experimental perspective encompassed in the Academic Advising Systems discipline.

Nowadays, students have many options when they want to take a course. Usually, it is complicated for students take that decision. In this context, recommendation systems have been proposed as tools that help students to make their choice using CF, CBF and hybrid techniques. Systems based on CF are widely used. [ChLC16] presented a two-stage user-based CF process using an Artificial Immune System (AIS) for the prediction of student grades. In order to address the problem of the amount of feedback required from students to produce recommendations, authors segregated the students' population with demographic information, and they introduced a control mechanism that filters courses whose instructors have a low rating. [Taha12] introduced an XML user-based collaborative system which advises a student to take courses that were taken successfully by students with the same interests and academic performance. The students' categorization is based on course features such as memorization skills or programming skills, among others. [BSZE17] explored the inclusion of a normalized system to describe the

competences that a course provides and the courses that helped to the students to achieve them. [GaLi15] designed a web-based RS that uses K-means algorithm to determinate the similarity of the students. With respect to systems based on CBF, recent and relevant proposals can be found. [MOKR14] presented a case-based reasoning that made recommendations based on matching features associated to courses. Ontology-driven software development [HuCC13] is also explored as CBF system. In this case, modeling various aspects of the academic plan to recommend courses that help students to complete the required credits. Recently, [MWHL17] explored the application of semantic similarity to courses description for providing recommendations. Finally, hybrid RSs that combine several techniques of recommendation are taking more and more importance. [Unel11] explored the combination of CF and CBF through the generation of recommendations generated independently and presented together. That study showed the importance of using an existing and relatively large dataset to test the RS. [DEAA14] presented a hybrid CBF system that combines association rules and case-based reasoning with courses-related information. [AIAI16] combined CF with association rules to predict students' performance. [GuLD18] explored the use of an ontology along with N-gram queries. [WuLZ15] proposed a CF combined with fuzzy trees to represent both student and learning activities information.

1.1. Pattern Mining in Education

The ever-increasing volume of data collected and saved in a variety of domains necessitates the need for data analysis and extraction of meaningful information. Raw data, in general, is uninteresting, and an in-depth analysis is necessary to extract the important knowledge that is normally hidden. This method gave rise to the knowledge discovery in databases (KDD) [AgSh14]. KDD is associated with the design of methods and strategies for extracting useful information from data, and pattern discovery is a key part of this. Patterns are subsequences, substructures, or itemset that express any form of uniformity and consistency in data [AgSh14]. As a result, patterns are basic and important aspects of datasets. and important properties of datasets. Given a set of items $I = \{i_1, i_2, \dots, i_n\}$ in a dataset, a pattern P is formally defined as a subset of I , i.e., $\{P = \{i_j, \dots, i_k\} \subseteq I, j \geq 1, k \leq n\}$ that describes valuable features of data. Given a pattern P , its length or size is expressed as $|P| = j$, representing the number of single items or singletons it contains. Thus, the length of a simple pattern $P = \{i_1, i_2, \dots, i_j\} \subseteq I$ is defined as $|P| = j$ since it is comprised of j singletons. Detecting and analyzing patterns in a database may appear to be a simple process, but it becomes progressively difficult as the importance of the identified patterns increases. Pattern mining [VeLu16], defined as the activity of detecting patterns of high interest in data for a given user goal, is regarded as a key aspect of the KDD process. Thus, the process of properly assessing the interest of identified patterns includes several metrics that are significantly relevant to the objective for which the work is implemented [CNAG13]. On many cases, many purchases are made on the spur of the moment and in-depth examination is

required to collect clues that indicate which specific objects are closely linked. For example, how likely is it that a consumer will purchase bananas and milk all at once? In this regard, it appears simple to deduce that the interest of probability of occurrence of the patterns could be used to quantify them. Using a sample set of customers, we can observe that the pattern $P = \{\text{Banana; Milk}\}$ has a frequency of three out of a total of five customers, implying that there is a 60% chance that a random customer buys both bananas and milk in the same trip. This study may help shops enhance sales by relocating products on the shelf, or it could help managers plan advertising campaigns. The findings of pattern mining can be analyzed in a variety of ways, and the same pattern can be utilized to implement a variety of marketing techniques in the view of few authors, Pattern mining is a high-level and difficult endeavor due to computational challenges. The pattern mining problem becomes a challenging task as the search space expands rapidly with the number of single objects or singletons taken within the application area.

The sequential pattern mining task was introduced by Agrawal and Srikant [AgSr95] to identify useful patterns in a set of sequences. Although the task was originally proposed to mine sequences of patterns, it has been extended to time series of ordered events [NZXW07]. Formally speaking, let $I = \{i_1, i_2, \dots, i_n\}$ be the set of n items contained in a database Ω . Let us also define an itemset X as a set of items from I , that is, $X \subseteq I \in \Omega$. A sequence s is described as an ordered list of itemsets $\langle X_1, X_2, \dots, X_m \rangle$. In a sequence s , an item i_j appears only once in an itemset X_k , but such an item i_j is allowed to appear multiple times in different itemsets belonging to s . As a matter of clarification, let us consider the set of items $I = \{a, b, c, d\}$ and the sequence $s = \langle \{a, c\}, \{b\}, \{a, c, d\} \rangle$. In this example, the item $\{a\} \in I$ appears in the itemsets $X_1 = \{a, c\}$ and $X_3 = \{a, c, d\}$, but it only appears once in each itemset. Additionally, the sequence s is formed by a set of itemsets: $X_1 = \{a, c\}$, $X_2 = \{b\}$ and $X_3 = \{a, c, d\}$. Generally speaking, the meaning of a sequence is that events within an itemset occur at the same time, but itemsets take place one after another (the itemset X_1 does always appear before X_2). Sequential pattern mining aims to extract any sequence that appears in Ω [ReFT09].

A sequential pattern is a restrictive form of association rule [AgIS93] in which the order of the accessed items is considered (the association rule discovers all the relationships without restrictions). Sequential pattern mining was first introduced into the study of customer purchase sequences. Given a set of sequences, where each sequence consists of a list of elements and each element consists of items, and given a user-specified minimum support threshold, sequential pattern mining tries to find all frequent subsequences, i.e., the subsequences whose occurrence frequency in the set of sequences is no less than the minimum support. There are several popular pattern discovery algorithms [HaPY05] such as AprioriAll, GSP, SPADE, PrefixSpan, CloSpan and FreSpan.

SPM techniques have been widely applied to analyze student learning behaviors in web-based educational systems. For example, sequential pattern mining has been developed to

personalize recommendations on learning content based on learning style and web usage habits [ZhLL08]; to study eye movements (of students reading concept maps) in order to detect when focal actions overlap unrelated actions [NXWZ08]; for developing personalized learning scenarios in which the learners are assisted by the system based on patterns and preferred learning styles [BaPA07]; to identify significant sequences of activity indicative of problems/success in order to assist student teams by early recognition of problems [KaMY06]; to generate personalized activities for learners [WWST04]; for personalizing based on itineraries and long-term navigational behavior [MoMi04]; to recommend the most appropriate future links for a student to visit in a web-based adaptive educational system [RVZB09]; to select different learning objects for different learners based on learner profiles and the internal relation of concepts [ShSh04]; for personalizing activity trees according to learning portfolios in a SCORM compliant environment [WWST04]; for recommending lessons (learning objects or concepts) that a student should study next while using an adaptive hypermedia system [Kris05]; to discover LO relationship patterns to recommend related learning objects to learners [OuZh07]; for adapting learning resource sequencing [KaSa05].

1.2. Curriculum Mining

A curriculum is partially designed by an educational institution in order to accomplish certain goals. Curricula normally suggest that students can follow differing paths from start to end due to a liberal approach in selecting courses [WaZa15].

Curriculum Mining (CM) is very related with Educational Process Mining (EPM), an emerging field in Educational Data Mining (EDM) aiming to make unexpressed knowledge explicit and to facilitate better understanding of the educational process [BoCR18]. EPM normally uses log data gathered specifically from educational environments to discover, analyze, and provide a visual representation of the complete educational process. It includes three main kinds of tasks: (i) actual curriculum model discovery, i.e. constructing complete and compact academic curriculum models that are able to reproduce the observed behavior of students, (ii) curriculum model conformance checking, i.e. checking whether the observed behavior of students match their expected behavior as defined by the previously discovered or pre-authored curriculum model, and (iii) curriculum model extension, i.e. projecting information extracted from the observed data onto the model, to make the tacit knowledge explicit, facilitate better understanding of the particular academic processes and enable decision making processes.

Curriculum data is the history of the courses effectively taken by students. It is essentially process centric. Applying process mining on curriculum data provides a means to compare cohorts of students, successful and less successful, and presents an opportunity to adjust the requirements for the curriculum by applying enhancement of process mining [WaZa15]. This can lead to building recommenders for courses to students based on expected outcome.

There are several good examples of application of EPM specifically for curriculum mining or in a more general educational application. For example, a domain driven EPM approach was proposed by [TřPe09] for doing curriculum mining. They proposed a framework which assumes that a set of pattern templates can be predefined to focus the mining on a desired way and make it more effective and efficient. The framework is aimed at helping educators analyze educational processes based on formal modeling. In other related re-search, [WaZa15] discovered a curriculum process model of students taking courses and compared the paths that successful and less successful students tended to take, highlighting discrepancies between them. In other work [SMMS17] presented research into educational process mining and student data analytics in a whole university scale approach with the aim of providing insight into questions raised by degree pathways. Their goal was to uncover statistically significant and meaningful patterns in students' course pathway choices, and to provide student support units, degree and course coordinators with longitudinal indicators that could be used to inform students.

1.3. Curriculum Personalization Systems

Curriculum Personalization Systems (CPS) are designed to offer students unique learning opportunities. The issues of curating (i.e., selecting) and developing individualized curriculum resources have become increasingly prominent as the usage of personalization tactics has grown in popularity. There is no systematic way for instructors to learn how to choose and design curriculum resources that support personalized instruction. This has resulted in widespread misunderstanding about what personalized instruction is and is not, as well as a wide range of approaches used by educators in the label of personalized curriculum [BOHG13].

This section concentrates on personalization approaches that have been shown in the educational research literature to have a favorable impact on student learning outcomes. These strategies are recommended to be included in instruction as well as in the resources chosen or generated to personalized instruction. Goal setting, feedback, periodic formative evaluation, deliberate practice, and peer tutoring are all recommended approaches.

1.3.1. Goal Setting

Setting goals requires identifying and specifying the learning objectives that an individual student should acquire after completing an activity, module, or other part of curriculum. Goals often include not just the accomplishments to be attained, but also the period required to complete the achievement. In the view of [Lock16], goals instruct people "as to what sort or degree of performance is to be reached so that they can direct and assess their activities and efforts correspondingly" (p. 23). Furthermore, Locke and Latham propose that objectives govern action, describe the nature of the relationship between the past and the future, and postulate that human aims are guided by intents. Performance goals must be clear, so that teacher and student have a common knowledge of what is expected, and they should be demanding in comparison

to a student's present repertoire. When framed in terms of "personal best" targets for individual learners, goals are likely to be particularly effective as a personalization method. Some digital resources automatically analyze performance and create goals, which is a characteristic to check for when purchasing items. Assess a student's present performance in relation to mastery, then create an acceptable and attainable, yet demanding, goal that is an adequate "personal best," considering the student's present ability.

1.3.2. Feedback

Feedback is information about a learner's performance that he/she receives. According to [Lock16], feedback enables students to "establish appropriate goals and assess their performance with regard to those goals so that modifications in effort, direction, and even strategy can be made as needed" (p. 23). Feedback should be precise, rapid, and regular, according to the rule of thumb. Specific feedback includes statements like, "You did an outstanding job including numbers today," rather than the more generic and imprecise, "Great work." In the first situation, the learner learns just what he performed well. Finally, feedback should be delivered frequently. Frequent feedback lets learners know how successfully they are moving toward their goals as they progress. This is especially feedback to shape their performance in the direction of mastery not only helps with accuracy, but it also helps keep them motivated. When choosing digital products, test them out to ensure that feedback for correct responses and errors is included. The higher the quality of the educational product, the more feedback there will be. If a digital product does not contain incorporated feedback or if a low-tech product is used, feedback must be provided by the teacher. Most teachers use feedback to personalized instruction for individual learners without any effort, but teachers can also include areas in curriculum materials where participants are allowed to ask for feedback. Individual learners, when compared to one another, perform different things well and make different mistakes, therefore as long as the feedback is particular, it is automatically individualized for the learner. As the instructor observes her learners, she will most likely see that the struggling students require more frequent feedback, both to fix their problems and to keep them motivated by pointing out what they are doing properly. One of the most important aspects of personalizing instruction with feedback is to remember to give feedback to the more successful learners as well, because they are sometimes disregarded in favor of the students who require more support. In generally, the goals are to retain feedback relevant and quick for all learners while also varying feedback frequency based on individual learner needs.

1.3.3. Periodic Formative Assessment

Periodic formative assessment comprises frequent and scheduled reviews on progress toward the student performance targets designed for curriculum. Formative assessment does not affect student grades and is not considered "testing." It is meant, on the other hand, to provide

feedback to the teacher regarding which information the student is mastering and which content the student may be misunderstanding so that corrections can be made as soon as possible. With formative assessment data, the teacher (or digital program) can make changes to the curriculum path, including remediation to address any challenges the individual student may have with the topic, as well as speeding up or slowing it down. As a result, whether provided by a teacher or a software programmed, the student receives individualized interventions as the curriculum path adapts to match his specific needs. One significant advantage of continuous formative evaluation in digital training is that it automatically adjusts difficulty levels based on the learner's performance. This happens on the go, without the student or an adult having to adjust any programmed settings. The key to this is the automatic component, which can be found in online curricula, computer-based software, or mobile apps that run on a device. The curriculum adapts its degree of difficulty on a response-by response basis to what is best appropriate for the specific learner based on the pattern of responses given by that student.

1.3.4. Deliberate Practice

Deliberate practice is the provision of multiple chances for active response throughout a period of education. Unlike "time on task," which includes all time spent in the presence of a task, both active and passive, deliberate practice focuses on active responding and the chances generated to encourage active responding. "Behaviors such as writing, oral reading, academic talk, asking questions, answering questions, and motor behaviors associated in engaging in academic games or activities" are examples of active responding [DGWC86]. Enhancing active response through deliberate practice improves the probability of learners paying attention and focus on the task. Deliberate practice is more than just "drill and practice," as it incorporates feedback and established performance objectives. A student answers interactively, and quick feedback on the appropriateness of the response is provided, allowing the learner to change her next response as needed. The more chances an individual student must answer actively during a period of academic instruction, receive feedback, and reply again while incorporating that feedback, the faster he or she will acquire mastery performance.

1.3.5. Peer Tutoring

Peer tutoring is the partnering of students to work together on projects during their studies. Peer tutoring is frequently used with more experienced students instructing less skilled and struggling students, however it is believed that one of the key reasons peer tutoring works so effectively is that "it is a great approach to encourage students to become their own teachers [Hatt08a]. Class wide peer tutoring (CWPT) is one sort of peer tutoring implementation in which all learners in a classroom are grouped into tutor-learner pairs. CWPT promotes students' opportunity for deliberate practice by ensuring that all students are actively engaged during academic instruction. Peer tutoring can be utilized to personalize the experience of both students who have

been paired; the difficulties and obligations of each student in the duo will be different, based on each learner's skills and abilities. For example, if Katie is partnered with a more talented learner, the level of difficulty will be increased, but she will have a student mentor to assist her in making progress. Some digital items are designed to be used by more than one student at a time. When choosing from these products for use in peer tutoring, make sure to choose collaborative products that allow individual users to collaborate to achieve a goal, rather than products that allow users to compete against one another in real-time play. If you are partnering a more talented student with the less skillful student to use a collaborative digital product, make sure the more skilled student understands how to use the product, what the product's learning goal is, and how to track progress toward that goal before beginning a session with the less skilled learner.

1.4. Course Difficulty Index

It is generally accepted throughout the academic community that differences exist among college courses in the level of difficulty of the course. Some of these differences can be attributed to the fact that courses are designed to be taught at each of the different levels of student classification [Mund91]. Courses designed primarily for freshmen and sophomores (course numbers in the 100-200 range at many colleges and universities) are generally considered less difficult than those designed for juniors and seniors, or for graduate students. Another component of the difficulty level can often be attributed to the course instructor. Two different instructors teaching the same course (often in different terms or even in different years) may design the course differently and/or teach the course in different ways. A third aspect of the difficulty level of courses which may vary from one course to another is the number of credit hours associated with the course. It was originally the case that courses carrying larger numbers of credits were the more difficult courses. It is not at all clear if that is still the case. It is often perceived that the number of credit hours is somehow equated with the amount of work involved in a course but may represent nothing more than the number of hours that a course meets each week. But there is more to the differences among courses than these instructor, classification-level, and credit hour variations. All freshmen- and sophomore-level course or junior- and senior-level courses do not carry the same level of difficulty. Even within a given department, it is not clear that every senior-level course would be rated as having equal levels of difficulty, and when courses are compared on an interdepartmental basis, the difference in difficulty levels is likely to be even greater.

The difficulty of a course can be estimated by considering the average grades awarded, rank correlation coefficient (ρ) – means, scaling analysis and cluster analysis as factors [1, 2]. So, Course Difficulty Index (CDI) estimation relies on factors like course content difficulty, the difficulty of the question paper, feedback of the instructors/students. There are different authors that has used their own Course Difficulty Index in their educational problem as we describe below.

Mundfrom et al. initially propose to record the course difficulty index in the Likert-type scale of 9 points from very easy to very difficult [Mund91]. Then, [PrVi19] consider the Likert-type scale of 5 points for recording the course difficulty as very easy, easy, moderate, difficult, and very difficult from 1-5, respectively. Every course C_i is assumed with a course Difficulty DC_i . Courses are distributed to k semesters such that the average course difficulty of every semester is approximately equal to the Average Difficulty (AD) of all the courses. [Jign21] proposed a methodology that estimates the difficulty level of a course by mapping the Bloom's Taxonomy action words along with Accreditation Board for Engineering and Technology (ABET) criteria and learning outcomes. The estimated difficulty level is validated based on the history of grades secured by the students. *Sonavane et al.* [SRA00] calculated a Difficulty index (CDI) by counting the total number of students who properly answered each question divide the number of students who answered correctly by the total number of students for each item. This tells you what percentage of students answered each question correctly. So, the item becomes easier as the difficulty level rises, and vice versa.

Finally, the Difficulty Index has been proposed for measuring the likelihood that candidates or students will properly answer a test item. The percentage, or P-value, of more complex and difficult is usually smaller [BaSc03].

2. Supervised Learning in Education

Supervised learning is also called supervised machine learning and is part of artificial intelligence and machine learning [Kots07]. Supervised machine learning is the search for algorithms that reason from externally supplied instances to produce general hypotheses, which then make predictions about future instances. In other words, the goal of supervised learning is to build a concise model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown.

Predicting the educational outcomes for students is one of the main application areas of the application of supervised learning in education [RoVe10]. In this area, supervised learning has received a lot of attention for its ability to accurately predict a variety of complex areas. In fact, one of the most important areas for accurately predicting outcomes is educational environments.

In the supervised scenario, one attribute (or occasionally a small collection of attributes) is deemed exceptional, and we are only concerned in discovering correlations between this attribute and the other attributes. While this reduces the number of patterns that can be discovered, it makes the method more specialized and, in many situations, more effective. Look at the context of customer defection (churn), where one wants to find relationships between customer loyalty and other customer characteristics; or consider applications in

cheminformatics, where one wants to find relationships between molecular structures and their activity: in all these cases, a targeted analysis with respect to the indicated target attribute is likely to produce the most valuable results. In terms of new obstacles, supervised pattern mining provides new opportunities, because supervision allows for the application of additional quality measures and pruning depending on the features of constraints depending on such measurements.

There are a lot of published papers about the application of supervised pattern mining in education. In the next paragraph it is described some of the most important previous research works related with this dissertation. In [KASJ14], Kaur proposed that in the last few years, Information technology has a great priority in many businesses especially education institutions. One of the biggest challenges in all educational institutions and schools faces today is improving student performance. The recent years have witnessed period of global growth and technical innovation, learning is taken into consideration as a first step for each human activity. It performs a crucial role in the world, especially individual's well-being and opportunities for better living. In [Yu15], Yu reported that the quality of Higher Education (HE) faced some difficulties so, it is important to realize and enhance the performance of HE. Identifying the most important criteria that influence the excellence of educational achievement is a very difficult and challenging issue. In [YiBa14], there is a need to shift the theoretical information to observed research and practice, which is especially crucial for improving students in their profession. In [DoCh13], pupil overall performance prediction is vital in various exceptional contexts in the instructional system. In [ChDo14] the accuracy of Student Academic Performance Prediction (SAPP) facilitates decision making process and enhances educational services at Higher Education Institutions (HEIs). So, this may especially assist in enhancing the instructional performance and, therefore, enhance the overall academic first-rate. In [Kuch16], on other hand, the increasing competition and the international mobility of students and staff raised the expectations about better education quality. This reflects the need to develop an internal mechanism for quality management as a strategic goal for many of the HEIs. In [BBSS16], Higher Education Institutions (HEIs) want to be subjected to a few benchmarking or performance assessment. Moreover, in [AIBS00] *Al-Turki et al.* reported that the HEIs and governments in need to modify its strategic procedures and established rules for the improvement of educational process and added that Key Performance Indicators (KPIs) provide quality assurance to higher education. In [ACDL16] the achievement of good honors in HEIs are important issues that must be considered, both for pupils and the institutes that host them. The authors in [NaZw14] proposed that HEIs are regularly inquisitive regarding fate of pupils. In [Edin12], *Osmanbegovic and Suljic* defined data mining as a data processing approach that is useful to attain good understanding of data. In [FMSF13], *Mashat et al.* defined data mining as a technique to investigate records to extract hidden patterns. In [Chan06], the authors defined it as the process of selection, discovering, enhancing, demonstrating, and evaluating huge amounts of facts to discover formerly unknown patterns., it

has been used extensively in various regions consisting of technology, engineering, business, banking, and even preventing terrorism for a long term.

Data mining (DM) is a critical part of HEI's information control structures [DeVH16]. It has significance in the business world and allows the educational group to make good choices associated with the pupils' instructional popularity. DM may be used to extract and find out the valuable and meaningful understanding from a big quantity of information [RoVG08]. In [Kiri14], the authors studied the ambiguities of factors that affect predicting Course Study Planning (CSP) to enhance the student performance decision making process are very critical issue. CSP is one of the most significant complications in education systems of HEIs. DM techniques used in different educational applications, such as admission system, enrollment management strategies, predicting student's division, performance prediction and other related areas. [FMSF13] provided an affiliation rule discovery version to analyze and examine King Abdul Aziz University (KAU) admission gadget database. They modeled the machine as a relational database to preprocess data before applying mining algorithms. The model discovered the relation students' records and their utility fame within the college machine. They determined some vital points related to college students' gender and students' classifications.

In [KuBa13], the authors used a simple method based totally on k-means clustering algorithm, as a simple and efficient tool, to investigate the information received from the admission shape filled via admission seekers. The proposed method assists the academic planners to reveal admission details of college students in search of admission in institute over the years. In [ZAAZ12], *Zainudin.et al.* found the presence or absence course difficulty using a in particular designed algorithm to compute the course difficulty level. In [Lagh16], the authors designed and developed a Knowledge Based Course Planning System (KBCPS) to identify the course planning needs of the institution. In [AhIA15], *Ahmad et al.* carried out numerous studies to construct Student Academic Performance prediction model for unique courses or subjects. These studies hire unique styles of student's statistics with a ramification of parameters to pick out and classify their college students. In Malaysia, researchers have performed a look at first-year bachelor college students of pc science from UniSZA. They took a look at completed comparative evaluation amongst 3 selected category algorithms; Decision Tree (DT), Nave Bayes (NB), and Rule Based (RB). The facts set selected for experiments comprises of a duration of 8 years that incorporates 497 reports from July 2006/2007 to July 2013/2014. The record includes numerous elements of college students' files, such as family heritage, previous instructional reports, and other demographic features. RB showed the very best accuracy fee of 71.3%. The version will allow the academics to take early movements to help and assist the poor and common class students to improve their effects. Authors declare that terrible and average result may be identified earlier with the aid of the usage of this version to enhance their destiny overall

performance. The dataset turned into quite small because of incomplete and missing values. The study can be comprehensive by way of including greater data to boom accuracy.

In [MCNY19], *Ma et al.* proposed a Multi-Instance Multi-label (MIML) set of rules by using okay-nearest neighbor strategies. The studies were performed on academic warning device for detection of students who find problem of their prior guides. The paper further confirmed that college publications were correlated consequently, it is far higher to expect them simultaneously. The result is no longer the simplest compared with traditional supervised learning approach but also with quotation KNN. Mostly researchers heavily relied upon on-line gaining knowledge of activities however here consciousness is on conventional face to face getting to know. Student performance changed into predicted prior to the start of each direction. Corse correlation became absolutely utilized. There are many other features that can affect scholar performance, including circle of relatives, fitness, and philological fame. Moreover, the amount of employed dataset used became quite small.

In [KaPN17], the authors discovered that the very last semester marks had been expected from the internal marks of college students. Dataset with 1938 example have been used in the experiments. To growth the accuracy, this tool has added reweight more advantageous boosting set of rules. The results have been in comparison to existing algorithms like Adaptive Boosting (Decision Stump). The Adaptive Boosting (J48) produced a great deal better accuracy. The type strategies have been implemented to the scholar's records. This model has shown development in student overall performance and class imbalance problem changed into advert dressed.

In [AmHA16] presented that prediction version turned into proposed on dataset gathered from LMS (getting to know control gadget). Behavioral capabilities, demographic capabilities, educational lower back-ground functions were taken into consideration. Filter approaches the use of statistics gain based on choice algorithm become used. The results turned into severely analyzed with and with-out behavioral features. Ensemble approach became used to improve accuracy and recall after making use of conventional class techniques specifically: DT, NB and ANN. Boosting method came up with good result. More features could be analyzed by way of the usage of this model. This model executed thoroughly with 480 facts and showed eighty% of accuracy however it must be verified with big dataset.

In [AMAH17], *Asif et al.* analyzed the completion of college pupils in four years bachelor diploma. Looking at the best marks as an input without considering every other function. Naive Bayes done notable with accuracy of 83.65 % observed by using 1-NN and Random Forest. NB, 1-NN and Random wooded area had been now not human understandable, so decision tree became used to derive the effective indicator guides. Typical progression of pupil performance was analyzed through X. Suggest clustering and Euclidean distance. The employed information set become primarily based upon a sample of 210 undergraduate students. The result confirmed that proposed pragmatic coverage was dependable which showed early sign of battle and

opportunity, graduation overall performance of other diploma application will be analyzed. The guides identified as indicator for excessive or low can be investigated for student performance. This prediction device changed into proposed for annual device. Further studies might be conducted for semester system at the equal parameters, to be giving the college any other leverage to enhance academic effect.

In [AICA19], the authors performed a prediction version for comparing student performance through the usage of dataset of four hundred statistics with thirteen capabilities. This observes analyzed correlation and courting of capabilities to their corresponding labels (student performance). Several Machine Learning (ML) techniques had been examined to predict the pupil overall performances that imply how range becomes the usage of those methods and to what quantity they assist in developing the performance. The only classifiers in forecasting the scholar performance were tree-primarily based classifiers related to the other households of classifiers with high accuracy and F-measure values. The experiments have been performed to improve the result of best classifier by using the ensemble technique. The outcomes have revealed a great enhancement using the novel model set of rules. Student might be examined with extra functions consisting of how social media can torment pupil concerning academic overall performance. The concrete set of ML techniques may be used here to enhance the overall performance. More records mining techniques might be implemented including clustering etc. With identical dataset for assessment, it is version that cannot handle range of different courses.

In [DLAA17], *Daud et al.* assessed the consequence of every feature for estimate of overall performance of scholarship retaining college pupils. The dataset is composed of 23 decided on features and 776 student record. New features aside from academic, by and large related to circle of relative's expenditure and student private information were taken into consideration. The capabilities like student's natural gasoline expenses, electricity expenses, self-hired and area characteristics have been maximum important for prediction of college student's academic prediction. The concrete function may be used to reap most accuracy. More records mining techniques can be carried out to boom accuracy.

In [MaDM16], the authors identified scholars at risk via in route performance at some point of the semester. The authors have employed logistic regression, support vector machines (SVMs), choice timber (DTs), ANNs and a Nave Bayes classifier (NBC) for experiments. The look at aimed to preserve false bad mistakes with lowering fake nice errors. This looks at used input functions, including grades, project milestones, group participation, attendance, quizzes, weekly homework, mathematical modeling pastime tasks, and checks. The high-quality fashions for predicting student who exceeded are KNN with ninety-nine.7%, MLP with 96.7 and DT with ninety-six.1 accuracy. These models have low accuracy with admire to identifying failed students. As studies are dealing with scholar at danger, pleasant fashions for identifying such are NBC with 86.2%, SVM with 72.4 and logistic regression with 58.6. Finally, researchers have used ensemble

approach which includes two models with the bottom fake terrible mistakes. The NBC completed better because records set carries most effective 10% of the students. The NBC is an easy model with excessive bias and coffee variance. However, NBC might also perform poorly for different dataset. Pedagogical selections made by means of the direction designers can vary from route to course. Another hassle is locating most efficient time for prediction.

In [ASVG17], *Atherton et al.* analyzed college students' range and how courses are added to fulfill the requirements and effects. This examine concluded that student engagement with era advanced the academic consequences. The on- line course getting to know cloth played a substantial role in the result of students. Accessing on-line studying cloth was a trademark of good overall performance of students. The students enrolled in full-time consuming one semester have been as compared to element-time student lasting semesters in duration. The result confirmed that the scholar getting access to on-line guides frequently had better evaluation and marks. It additionally found that there has been a tremendous relationship between talent and participation level. Moreover, the male college students were dominant in full-time course and girl in half-time route. The authors have shown the comparisons outcomes through graph; however, basic results are not completely elaborated. There turned into a minority group that confirmed excessive stage of get right of entry; however, their development result in examination became worse. Further studies can be conducted on the quantity of time spent on online cloth to improve instructional research. Female and male students' behavior can be studied extra for impact on overall instructional performance.

In [AAAB17], *Al-Shehri et al.* predicted the very last scores of Mathematics so that proper scholar can be selected for the positive obligations with the aid of the usage of methods, ok-nearest neighbor model and SVM prediction version. Feature choice system was carried out by finding correlation among the grade and goal value. The comparative assessment of KNN and SVM model showed that both classifiers carried out thoroughly for this kind of situation but SVM barely outperformed KNN with the correlation coefficient of 0.96 and 0.95 respectively. This work anticipated actual cost which is regression problem this is more complex than classification. Performance of proposed techniques need to be evaluated thinking about type elements. The SVM has not performed nicely with huge database as it calls for long time for training.

In [MaDS15], *Marbouti et al.* looked at become conducted on use of early caution device based totally on route to growth student achievement. Three models were built for week 2, 4 and 9 to predict students at exclusive time. Models ware optimized to discover scholar at chance. Models accomplished accuracy up to seventy-nine% for week 2, ninety% for week 4 and ninety-eight% for week 9 by Analyzing effective time for performance prediction of scholar.

In [FrBa19], the authors proposed some new predictions method was followed based on each classification and clustering methods. This observes achieved the experiments on Learning Management System (LMS) sixteen features. The quality accuracy of 75% was determined while

carried out to Academic, Behavior and Extra features. The result confirmed that the hybrid technique yielded precise outcomes in time of accuracy in prediction of college student's performance. This model can be prolonged for types of features of pupil dataset.

In [BhSB18], *Bharara et al.* used disposition evaluation to understand student association with a path. This research explores meaningful capabilities affecting maximum the scholar's overall performance. Learning Management System (LMS) of 489 records became used as dataset with 16 features. K-Means clustering turned into used on this version to evaluate the effect of pupil's interactional functions and college pupil's parental involvement capabilities on student academic performance. The result showed more potent impact of those features on instructional performance. Clustering performs properly for heterogeneous form of records. More features should be studied along with different clustering techniques.

In [TuSA19], *Sana et al.* conducted the study on on-learning management system and analyzed the capabilities as punctuality of student and participation of parents regarding college students learning sports. This looks at used gain ration as a characteristic choice method which confirmed excessive effect of those capabilities.

In [MAMA19], *Suhaimi et al.* reviewed on prediction of student graduation time. It turned into visible student have been not able to manage to finish their have a look at on time. This paper focused on various factors and approaches used to are expecting commencement time. The result confirmed that of Neural Network and Support Vector Machine carried out well compared to Nave Bayes and Decision Tree. It became indicated that educational evaluation becomes a distinguished issue whilst predicting such students.

In [AKPS19], *Anusha et al.* stored the music of instructional document to make choice whether a pupil desires the academic intervention or no longer. The dataset contains statistics of 2015-19 batch college students of Computer Science by means of considering academic features. They have used regression model rather than classification version. The proposed device has expected to bring about the numeric manner with the aid of using KNN, Decision tree, SVM, Random woodland, Linear Regression and multi linear Regression by way of reading the result. It is seen that multi linear regression is a most beneficial answer.

In [AjAS20], *Ajibade et al.* analyzed the impact of behavioral impact on student performance prediction. Most influential functions have been decided on by using a filter out-primarily based method. To optimize the performance, distinct ensemble approaches were used. The result confirmed that behavioral features played important role in scholar overall performance prediction and DT carried out outclass through reaching ninety-four% accuracy with assembling.

In [VPSP14], *Veeramuthu et al.* proposed approach to know talents of the beginners in educational institute. These observes supplied a tenet to the higher education system in

enhancing choice making. They look at aims to analyze the various factors affecting beginners studying conduct and their performance using okay means clustering set of rules.

In [GaMo17], *Gadal and Mokhtar* proposed a hybrid device getting to know for network intrusion detection. The Consistency Sebest Evel and Genetic seek algorithms have been implemented to select to unique features. Hybrid approach with the aid of aggregate of K method and SMO is applied by way of attaining 97.3% and decreasing the false alarm rate (1.2%).

In [HTCL17], the authors performed that study by using distinct datasets. It is also shown that student's performance can be predicted and better via preprocessing. Here social surroundings and conduct functions were analyzed by way of using 5 special classifiers including Back propagation (BP), Support Vector Regression (SVR), Long- Short Term Memory and Boosting Classifier.

3. Literature Review Summary

Students are the main participants of any institutes. They contribute to the socio-economic growth of a country, leading to creative graduates, innovators, and entrepreneurs. Obviously, the use of learning management system has been increased and as a result, institutes contain extensive dataset storing various aspects related to students' academic performance. Instead of relying on experience, this performance data can be helpful to enhance student success by taking instant steps based on prior feedback. This analysis can help instructors, students, or educational institutes to predict.

The instructional systems are facing the trouble of low performances of college students. Many factors are inflicting educational degrading performance of a student. It has been located that some college students do now not entire their studies and depart during the consultation. Students may be spending a lot of time completing diplomas due to the negative overall performance, as they must repeat the publications as in line with institute requirements [MAMA19]. Therefore, it is essential to perceive the student at danger at the early stage to take appropriate measure to improve their overall achievements in the future.

We have completed in-depth evaluation of literature on pupil performance prediction the use of a hard and fast of varying features. The present predictive models are of widespread nature and cannot address course to path variety, as every path is differently designed by using the trainer. Time of conduction of pupil predictions is also influential in overall performance as it could be taken before the session begins, throughout the semester, give up of look at. The current studies have no longer targeted on time of conduction [AMAH17, MaDM16].

This thesis specializes in performance/applicability of different prediction models of students' overall performance with the goal of identifying students who are on chance so that

appropriate measures could be considered to enhance drawing close instructional overall performance. The proposed study is based on papers [AlCA19] and [FrBa19], that are considered as a baseline. The reasons of these papers being the baseline are as follows:

- These papers are highly recent
- They are published at reputed journals
- They recommend distinctive procedures, using the same records set
- The statistics set used is publicly to be had
- The consequences claimed in the papers for the same statistics set are distinctive.

The look at in [AlCA19] plays student performance prediction using ensemble methods achieving accuracy of 98%. Here dataset is reduced to 13 features and 400 statistics because of inconsistent and lacking values. In [FrBa19] while another technique was applied at the authentic dataset, the pronounced accuracy became quite low. So, record removed for the duration of preprocessing with no reason reasons result very high which is pretty unrealistic. In [35], Feature choice is implemented on classes rather than man or woman characteristics and an authentic dataset is used. Conversely, the proposed observe will attention on individual attribute and use hybrid approach combining both clustering (EMT) and class processes. The consequences can be compared to research first-rate strategies in terms of pupil performance prediction. This thesis makes a specialty of the prediction models by using comparing student performance the use of information set containing 480 records with 17 features.

The proposed look at is predicated on papers [AlCA19] and [FrBa19] taken into consideration as a baseline for this examine as the dataset hired by way of these research is openly available and posted in a reputed journal. The have a look at [AlCA19] predicts students' performance by using ensemble techniques to accomplish accuracy of ninety-eight%. Here dataset is reduced to thirteen features and four hundred records because of inconsistent and lacking values. As the original dataset is to be had, it's far visualized those missing values are very low in numbers and eliminating data can cause imbalance hassle. In [FrBa19] while another approach is implemented on the unique dataset, the model yielded low accuracy. In [FrBa19] feature selection is applied on classes as opposed to person characteristic. The top focus of our proposed look at is identifying smarter feature set and reaching advantages of category and clustering forming a hybrid approach. The effects are compared to investigate first-rate strategies in term of pupil overall performance prediction.

Recommender systems have received a lot of attention from different institutions to improve students' overall satisfaction grade [MDVH11], resulting in a higher number of students to enroll. Course recommendation systems can be categorized into collaborative filtering-based recommendation systems (CFRS), content-based recommendation systems (CRS), knowledge-based recommendation systems (KRS), hybrid approaches and data mining approaches. CFRSs

assume that predictions are done by considering the choices of other students with similar preferences and interests. A major problem of CFRs and CRs is the huge amount of data they require to make recommendations. When these systems begin to be used, the recommendation power decreases significantly due to a lack of information. Knowledge based recommendation systems (KRSs) can overcome that issue since the recommendation is performed by meeting users' requirements and courses. The system relied on several curricular profiles needed by the students to meet the requirements of different jobs. Like KRSs, hybrid course recommendation systems are also commonly used to overcome the problems of CFRs and CRs. Finally, recommendation systems based on data mining approaches have been proposed to help students choose the courses that best fit them. The UniNet method was recently proposed as a recommendation system based on deep learning to help students to take the right decision on the order, combination, and number of courses to take.

Data mining strategies function a remarkable technique for extracting precious styles. The educational institutes are concerned regarding pupil's overall performance as it is very critical ranking of the institutes. It is a serious challenge to reveal and monitor the overall performance by means of manner of everyday strategies. EDM is a growing research domain [RoVe20], however, a lack of art exists on the use of EDM techniques for global evaluation studies. This examines interests to fill the distance within the contemporary literature by using machine learning approaches.

The issue of student self-assurance has also been observed because the first-rate aspect on eighth-grade college students' mathematics success. By classifying students into remarkable groups, academic institutions can make stronger their admission systems and offering stronger their admission systems and higher educational services. Thus, a version which could classify college students based totally on their predicted instructional performance ranges is critical for establishments. There had been various approaches to classifiers. However, growing the classification model accuracy continues to be a subject with exquisite importance for a particular hassle. In this look at, we supplied an NFC model to a group of university college students and performed the comparative assessment against baseline models. The received outcomes established that the NFC version demonstrated superior performance than others. The consequences of the existing have a study also enhance the reality that a comparative evaluation of various processes usually supports selecting a class model with high accuracy. This work can be utilized for selection of better academic services by means of presenting custom designed help to students. A fuzzy true judgment model has been proposed for performance evaluation of university students of an everyday degree direction. The model defined factors including attendance; inner evaluation and external evaluation for fuzzification of the parameters through fuzzy inference policies.

CHAPTER III. Applying the CRISP-DM Methodology to KAU Student Information Data

In any knowledge discovery project, it is essential to establish a working methodology that takes you from raw data to the desired knowledge. In this sense, the objective of this chapter is two-fold. On the one hand, a brief description of the CRISP-DM methodology, which has been used for the preparation of the information and the development of the knowledge extraction methods to achieve the objectives of personalization of teaching set out in this thesis, will be given. On the other hand, the result of the application of this methodology will be shown, showing the resulting structure of the information as well as a first descriptive analysis.

1. Methodology

Data mining methodologies were introduced to provide a more holistic view to the knowledge discovery process, beyond the application of statistical or machine learning algorithms. Several data mining methodologies have been developed in the last two decades. [AzSa08] introduced a comparison between two of the most popular methodologies for the knowledge discovery process in databases: SEMMA (Sample, Explore, Modify, Model, Assess) and CRISP-DM (Cross-Industry Standard Process for Data Mining) and, although he states that both methodologies are very similar, CRISP-DM has two additional steps before and after the knowledge discovery process: Business Understanding and Deployment.

CRISP-DM steps help in the incorporation of the discovered knowledge into the business processes and are vital for project implementation. That is why we select CRISP-DM for this research. Moreover, CRISP-DM has been successfully applied in several similar educational projects [Luan02]. As shown in Figure 3.1, CRISP-DM involves iteratively performing several phases [Chap00]:

- 1) problem/business understanding
- 2) data understanding
- 3) data preparation
- 4) modelling
- 5) evaluation
- 6) deployment.

This section illustrates how we have applied the CRISP-DM steps to carry out the development of the models that have been developed in this research.

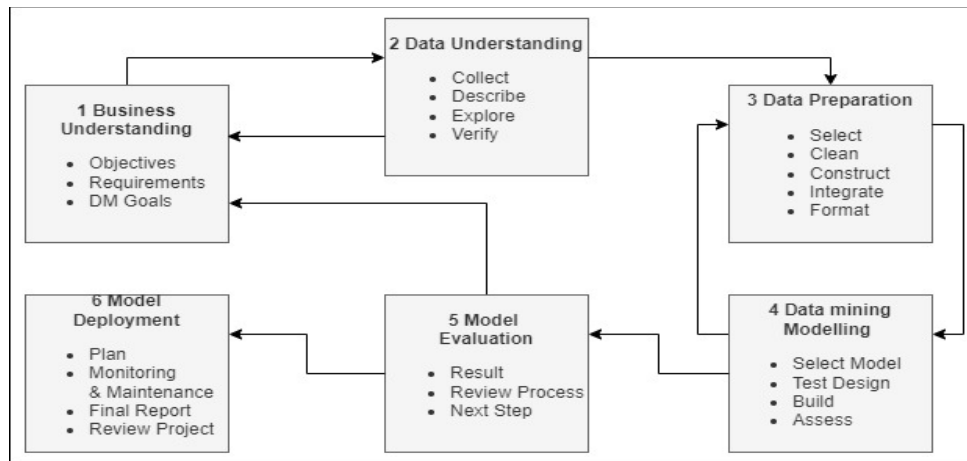


Figure 3.1: CRISP-DM Cross-Industry Standard Process

1.1. Research and Business Understanding

Business understanding is the initial phase that focuses on understanding the research objectives and requirements from a business perspective, then converting this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives [Chap00]. So, the Business Understanding phase can be identified with the development of an understanding of the application domain, the relevant prior knowledge, and the goals of the end-user.

Understanding the relevant domain knowledge is one of the key factors required for a successful data mining research approach. An analysis of the literature related to improving student performance and student course plans provided good information regarding the education area, student performance, and results of previous data mining studies in educational areas.

The research objective was motivated by the ongoing desire to improve the quality of education and improve the student efficiency and performance as well as measuring the course difficulty index to recommend each student a proper course plan, data mining techniques have been increasingly deployed to analyze the vast amounts of historical data being collected at various KAU Colleges that pertain to students' academic performance. This research can be considered from the academic points of view aims to find appropriate MODEL to improve student performance and student course plan using "Data Mining" techniques by analyzing student history, behavior, and other information systems to extract key factors to support credit hours, statistical methods for assessing Course Difficulty Index (CDI) and algorithms for implementing

systems for predicting CDI for the university providing information critical for decision-making. Thus, the research aims to find the answers to the following theoretical and practical questions:

- What is combination or set of courses will be suitable for students in semester?
- Can the students' historical data predict course difficulty?
- Can course difficulty measures be used to improve the prediction of academic performance?

1.2. Data Understanding

[AzSa08, Wirt00] reported that data understanding phase starts with an initial data collection and proceeds with activities to get familiar with the data, to identify data quality problems, to discover first insights into the data, or to detect interesting subsets to form hypotheses for hidden information. [Wirt00] added that there is a close link between Business Understanding and Data Understanding. The formulation of the data mining problem and the research plan require at least some understanding of the available data. The main steps involving data understanding are described below:

1. First, the data were collected from heterogeneous data sources in King Abdulaziz University. The researcher found five different databases in different environments that he needed to connect, retrieve, and collect the data as shown in Table 3.1.

Table 3.1: Data Sources in King Abdulaziz University

Systems	Environment	Database Type
Enterprise Resource Planning (Anjez-ERP)	IBM	DB2
Students Information System (OduPlus-SIS)	Banner	Oracle
Alumni System (Entemaa)	ASP.net	SQL
Questioner System (Estebana)	ASP.net	SQL
Yasser (Noor)	Webservice	SQL

The sample shows the portion of few colleges represented in 13 colleges, 5256 teachers, 95 study concentrates and 69 programs as shown in Table 3.2.

Table 3.2: Sample of proportion of Data in King Abdulaziz University

Colleges	Students	Teachers	Programs
13	13437	5265	69

The data sources contained four types of information stored by KAU University:

College, students, teachers, and programs. And contained the historical information of diverse dimensions such as courses, genders, age, nationality, grades, their combinations, and their collective viewpoints taken from the KAU database systems.

2. Data were described including its format. Evaluate whether the acquired data satisfies the requirements of this research in size and type. Data Size for the proposed Data Mining Model (DMM) dataset Properties:
 - Size statistics for datasets: 1.58 GB.
 - Number of records: 869,993 records.
 - Fields (attributes 83 fields).
3. The researcher describes results of the data exploration and creates the proposed DMDIM including first findings or initial hypothesis and their impact on the remainder of the study. During this stage we will address data mining questions using querying, data visualization and reporting techniques.
4. The researcher used different tools to list the results of the data quality verification.

1.3. Data Preparation

The data preparation phase covers all activities to construct the final dataset from the initial raw data [AzSa08, Wirt00]. So, in this phase data selection should be conducted by defining inclusion and exclusion criteria. Moreover, the bad data quality can be handled by cleaning data. Dependent on the used model (defined in the first phase) derived attributes must be constructed. For all these steps different methods are possible and are model dependent [ScKG21]. Data preparation tasks are likely to be performed multiple times, and not in any prescribed order. Tasks include table, record, and attribute selection, data cleaning, construction of new attributes, and transformation of data for modeling tools [Wirt00].

According to [HaKa12], data quality depends on six attributes, namely: accuracy, completeness, consistency, timeliness, believability, and interpretability. The authors highlighted six steps of data preparation that help to obtain data of high quality: Data Selection, Data Cleaning, Data Construction, Data Integration, Data Formatting, and Dataset Description.

Data selection decides on the data to be used for analysis. Criteria include relevance to the data mining goals, quality and technical constraints such as limits on data volume or data types.

Data cleaning deals with missing values, noisy data, outliers, and inconsistency problems. Missing values can be ignored (if the percentage of missing values is quite small), or filled with constants, global central measures, conditional central measures, or the most probable values. Noisy data can be smoothed, and outliers can be excluded from the analysis, although some potentially useful information

can be lost as a result. Several factors can lead to inconsistencies: unfilled optional fields, errors (human, deliberate, or technical), and outdated information. The information on the inconsistency problems and the solutions applied is called metadata and is captured and stored for future data transformation.

Data Construction task includes constructive data preparation operations such as the production of derived attributes, entire new records or transformed values for existing attributes. In this task the researcher constructs new data he needs in DMDIM dataset and not found it in KAU sources.

- *Derived attributes – There are new attributes that are constructed from one or more existing attributes in the same record such as teacher's experience in teaching the same course, the teacher's age at the time of teaching the specific course.*
- *Generated records – the researcher doesn't need to generate new records.*

Data Integration involves merging useful data from various data sources to create new records or values. In this task the researcher connects different databases as he mentioned in explore data task in data understanding phase. The researcher needed to combine tables and records to create new records and values.

- *Merged data - Merging tables refer to joining together two or more tables that have different information about the same objects such as students' information tables. These tables can be merged into a new table with one record for each store, combining fields from the source tables.*
- *Aggregations - Aggregations refers to operations in which new values are computed by summarizing information from multiple records and/or tables.*

Data Formatting. Formatting transformations refer to primarily syntactic modifications made to the data that do not change its meaning but might be required by the modelling tool to increase the predictive accuracy of the statistical models. Formatting is the final step before build the model, it is helpful to check whether certain techniques require a specific format or order to the data. The researcher used

- *Rearranging attributes.*
- *Reordering records.*
- *Reformatted within-value.*

Dataset Description. After finishing Data Understanding Phase and Data Preparation, all data inserts in new database called DMDIM DB as shown Figure 3.2.

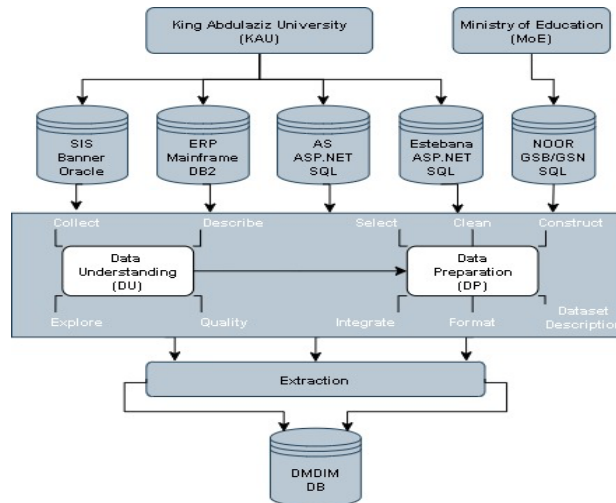


Figure 3.2: King Abdulaziz University Data Sources

1.4. Modelling

According to [AzSa08, PIDM21, ScKG21, Wirt00] data modelling phase consists of selecting the modeling technique, building the test case and the model. All data mining techniques can be used. In general, the choice is depending on the business problem and the data. More important is, how to explain the choice. For building the model, specific parameters must be set. For assessing the model, it is appropriate to evaluate the model against evaluation criteria and select the best ones.

Statistical and machine learning techniques for classification were employed in this phase where [Wirt00] reported that, there are several techniques for the same data mining problem type and some techniques require specific data formats, moreover there is a close link between Data Preparation and Modeling. Often, one realizes data problems while modeling or one gets ideas for constructing new data. In summary, the Modelling phase of CRISP-DM does not cater to needs of developing, improving, and refining models in data mining lifecycle. Furthermore, explicit guidelines how to iterate between phases.

1.5. Evaluation

According to [Wirt00] at the evaluation phase in the research project we have built one or more models that appear to have high quality, from a data analysis perspective. Before proceeding to final deployment of the model, it is important to evaluate the model more thoroughly, and review the steps executed to construct the model, to be certain it properly achieves the business objectives. A key objective is to determine if there is some important business issue that has not been sufficiently considered. At the end of this phase, a decision on the use of the data mining results should be reached.

For the evaluation of our model, a prototype of the Data Difficulty Index Model (DMDIM) for calculate the difficulty index for a course, different factors presented. The DMDIM tool

consisted of the initial hybrid statistical model for calculate the difficulty index for a course, based on four levels of student data (course/student/subject/teacher) incorporated into a software application.

1.6. Deployment

According to [Wirt00] the creation of the model is generally not the end of the project. Usually, the knowledge gained will need to be organized and presented in a way that the customer can use it. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process. In many cases, it will be the user, not the data analyst, who will carry out the deployment steps. In any case, it is important to understand upfront what actions will need to be carried out to actually make use of the created models.

The deployment of our research project started with the prototype of the of the Data Difficulty Index Model (DMDIM) for calculate the difficulty index for a course, different factors. Implementation of the Initial Model included statistical software that could be used for calculating the difficulty index for a course based on student historical data from the previous years. The output of the DMDIM was the list of difficulty weights for the courses for each student with a given threshold or percentile for course difficulty used to determine which courses are suitable for students should be included in his course plan.

All the necessary data for updating the database of the DDIM application will be uploaded by the primary user (a trained administrator from the Deanship of Information Technology at King Abdulaziz University), the main DMDIM maintenance will consist of regular system backups of the database. If any input items are added or deleted, or their sequence or format changes, minor software adjustments by a programmer will be needed.

2. Applying CRISP-DM to KAU Data. A Case of Study

This section illustrates the application of the CRISP methodology to all the available information on the King Abdullaziz University. Table 3.3 shows the scheduling for the application of the preliminary tasks (Data Understanding and Data Preparation), which main objective was the production of a reliable dataset to be used in the different investigations performed on this thesis.

In the following, we will proceed to describe how the different steps that comprise the CRISP methodology were performed, as well as the obtained results for each step.

Table 3.3: CRISP Scheduling

PHASES	February 2020 Weeks				March 2020 Weeks				April 2020 Weeks			
	1	2	3	4	5	6	7	8	9	10	11	12
1- Data Understanding												
Data Collecting	■	■	■									
Data Describing		■	■	■								
Data Exploration		■	■	■	■							
Data Verification				■	■	■						
2- Data Preparation	■	■	■	■	■	■	■	■	■	■	■	■
Select Data.						■	■					
Clean Data.							■	■				
Construct Data.									■	■	■	
Integrate Data.									■	■	■	
Format Data.									■	■	■	
Data Set Description.										■	■	■
3- High Performance Computing AZIZ.	■	■	■	■	■	■	■	■	■	■	■	■
Configuration.	■	■	■									
Create Environments.								■	■	■		
Database Creation.										■	■	■

2.1. Data Understanding

In this task, the researcher lists the available data sources acquired together with their locations, the methods used to acquire them, and any problems encountered. He found five different databases in different environments that he needs to connect, retrieve, and collect the data as shown in detail.

- **OdusPlus:** Student Information System (SIS) at King Abdulaziz University. This system allows the student to follow the entire registration and class schedule, as well as follow the courses and all services from the deanship of admission and registration. KAU management chooses the SIS from Banner with some development on the system to

make it easier and publish more electronic services. OdusPlus database is Oracle it is around 1800 tables.

- **Anjez:** This is Front-End web development for Enterprise Resources Plan (ERP).it was programmed using Lotus Notes-IBM environment. This system used to introduce electronic self-services for the staff in KAU. The Back-End is the IBM Mainframe ERP contains modules Human Resources HR – Payroll – Supplier management – Material Management MM – Warehouse Management WM – Finance and Controlling FICO. All transactions in the Anjez system in the DB2 database.
- **Estebana:** This system was programmed using ASP.net and linked with the SIS system to send a survey for all students before end of the semester to take their opinion in the course, instructors, and lecture building. Estebana read from SIS and write in the SQL database.
- **Entemaa:** This system introduces some facilities to the KAU Alumni such as access to the libraries and the medical services. The Entemaa system was programmed using ASP.net and database is SQL.
- **NOOR:** Is the Ministry of Education system they used to manage all students at all levels and send all data to the universities or the institutions by Government Secure Network GSN and Government Service Bus GSB. They use Web service for these services.

The best way to gather data of different factors would be to collect the data from a university setting, providing different areas of study for different majors. This model contains the historical information of diverse dimensions such as courses, genders involved, age, nationality, grades, their combinations, and their collective viewpoints taken from the KAU databases system. The data were collected from heterogeneous systems as dataset. It consists of 13,437 students enrolled in the university from 2013 to 2017 and they have completed the requirements of the academic program and obtained from the university with the 669,392 records. The sample shows the portion of few colleges represented in 13 colleges, 5256 teachers, 95 study concentrates and 69 programs as shown in Table 3.4.

Table 3.4: Sample of proportion of Data in King Abdulaziz University

Colleges	Students	Teachers	Programs
13	13437	5265	69

The data of teachers along with their qualification, experiences when the subjects were taught, students' personal, demographic, and academic data were included in it. Through exploration, following information has been obtained:

- These 13 colleges are specialized in different programs and diverse numbers of males and females are taught over there such as in arts and humanities (1450 males and 2442 females), in economics and administration (1078 males and 1128 females), Sciences (524 males and 1404 females), Business (636 males and 777 females), Law (718 males and 228 females), Sciences and Arts (209 males and 684 females), Home economics (614 females), Communication and media (325 males and 158 females), engineering (349 males and 16 females), computer and information technology (78 males and 206 females), Design and arts (167 females), Computing and information technology (70 males and 74 females), engineering (102 males) and collective male strength in all colleges is 5539 and female strength is 7898 as shown in Table 3.5 below.

Table 3.5: Information related to colleges

Colleges	Male	Female
Arts and Humanities	1450	2442
Economics and Administration	1078	1128
Sciences	524	1404
Business – Rabigh	636	777
Law	718	228
Sciences and Arts – Rabigh	209	684
Home Economics	-	614
Communication and Media	235	158
Engineering	349	16
Computing and information Technology	78	206
Design and Arts	-	167
Computing and Information Technology – Rabigh	70	74
Engineering – Rabigh	102	-
Total	5539	7898

- Figure 3.3 shows. the distribution of DMDIM data set students based on gender: 58.78% females and 41.22% males.

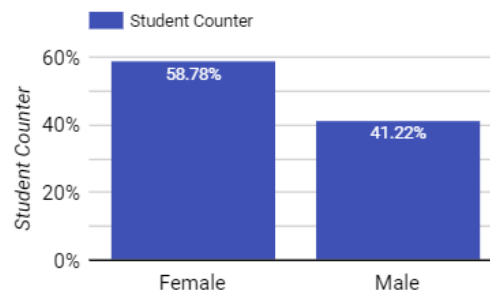


Figure 3.3: Distribution of DMDIM dataset students based on Gender

- The distribution of DMDIM dataset students based on nationality depicts that 97.12% students belong to Saudi Arabia and 2.88% are non-Saudi as shown in Figure 3.4.

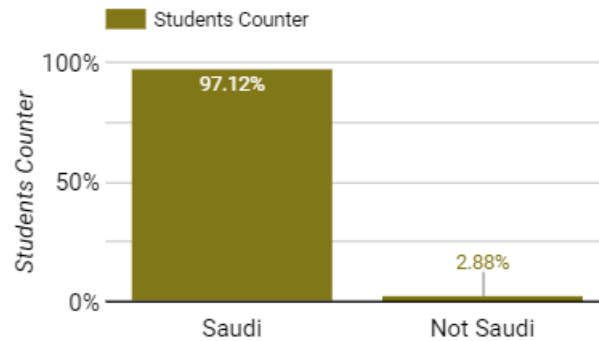


Figure 3.4: Distribution of DMDIM dataset students based on Nationality

- The collective effect of nationality and gender represents that 56.94% females and 40.18% males are non-Saudi. Simultaneously, 1.84% females and 1.94% males are non-Saudi as shown in Figure 3.5.

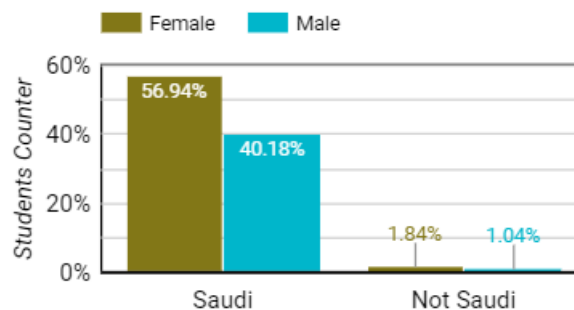


Figure 3.5: Distribution of DMDIM dataset students based on Gender and Nationality

- The age distribution depicts that major age limit is 19-21 of the students. 9726 students are age 19, 2130 belongs to 20 age group, 876 students are of 18 year, 697 students are of 21 years, 7 students belong to age group 17 and just 1 student is of age 16 as shown in Figure 3.6.

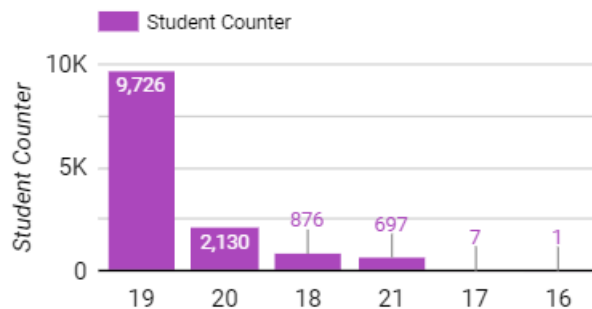


Figure 3.6: Distribution of DMDIM dataset students based on Age

- The distribution of KAU students on bases of their collective effect of age and gender depicts that 5,873 female and 3,853 male students belong to age group 19. Students having the age of 20, among of them 1,111 are female and 1,019 are male,554 female and 322 are male students that belong to age group 18 and just 357(female) and 340(male) belong to age 21 as shown in Figure 3.7.

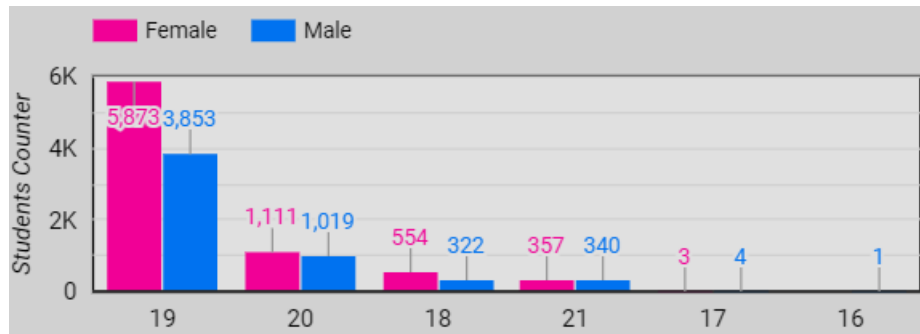


Figure 3.7: Distribution of DMDIM dataset students based on Age and Gender

- The distribution of students based on their grade depicts those 5,548 students have obtained very good grade,4,703 students lie in good category,2,636 have got excellent grades and 550 have just passed out as shown in Figure 3.8.

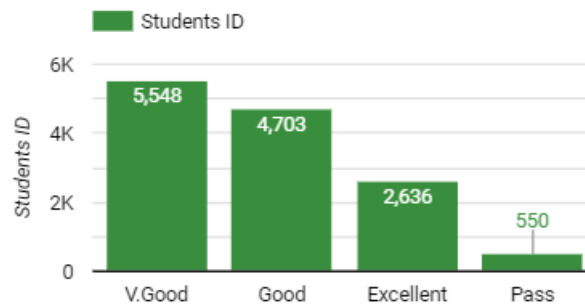


Figure 3.8: Distribution of DMDIM dataset students based on Grade

- The collective distribution of students based on their gender and grades depicts that 3,592 female and 1,956 males have got very good grade,1,833 female and 2,870 males lie in good category,2,256 females and 380 males have got excellent grades and 217 female and 333 males have just passed out as shown in Figure 3.9.



Figure 3.9: Distribution of DMDIM dataset students based on Gender and Grade

As shown in Figure 3.10, it displays the mass distribution of students depending on their faculties distributed by their grades.

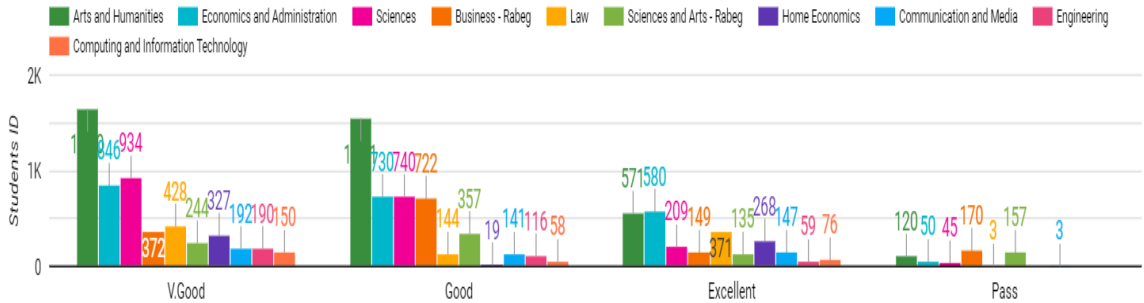


Figure 3.10: Distribution of DMDIM dataset students based on colleges and grade

Data Description

In this task, the researcher describes the collected data including its format. Evaluate whether the acquired data satisfies the requirements of this research in size and type.

- Data Size for DMDIM dataset Properties:
- Size statistics for datasets: 1.58 GB.
- Number of records: 869,993 records.
- Fields (attributes): 83 fields.
- Data Types: Data can be in different formats, such as numeric, categorical (string), or Boolean (true/false). Here are Tables constructing in research after collecting data and its fields, Attributes, and Data Types as shown in Tables 3.6 – 3.24.

Table 3.6: TSTUD_INF. This table contain student information collected from more than resources

Attribute	Type	Description
STUDENT_NO (PK)	TEXT	Student ID as primary key
SCHOOL_TYPE	TEXT	School type in high school
GENDER	TEXT	Male or Female
HOME_CITY	TEXT	City name for school
BIRTH_DATE	DATE/TIME	To calculate age
HIGH_SCHOOL_GPA	NUMBER	Score and grade in high school
UNIVERSITY_GPA	NUMBER	Score in university
SPECIALIZED_GPA	NUMBER	Score in his major
STUDY_PERIOD_YEAR	TEXT	To calculate years
STUDY_PERIO_SEMESTER	TEXT	To calculate summer semester
NATIONALITY	TEXT	Which country he belongs to

Table 3.7: TSTUD_ACADEMY. This table for students' status and how many plans he changed

Attribute	Type	Description
STUDENT_NO	TEXT	Student ID
STATUS	TEXT	Student status ex:
PLAN_CODE	TEXT	Plan code
PLAN_NO	TEXT	Plan number

Table 3.8: EMP_INFORMATION. This table contain information for staff in university

Attribute	Type	Description
EMP_NO (PK)	TEXT	Employee ID
NATIONALITY	TEXT	Nationality for the employee
LANGUAGE	TEXT	Language
BIRTH_DATE	DATE/TIME	To calculate age
CONTRACT_TYPE	TEXT	Temp – Permanent - Hired
GRADUATED_FROM	TEXT	University name
UNIVERSITY_RANK	TEXT	Rank in Times Higher Education

Table 3.9: TEMP_ACADEMY. This table contain academic employee information

Attribute	Type	Description
EMP_N (PK)	TEXT	Employee ID
ACADEMIC_LEVEL	TEXT	Teaching Assistant, Lecturer, Assistant Professor, Associate Professor or Professor
EXP_TEACH_COURSE	TEXT	Teaching experience in same course
EXP_TEACH_MAJOR	TEXT	Teaching experience in same major

Table 3.10: TCOURSE_INFORMATION. This table contain all courses information

Attribute	Type	Description
COURSE_CODE (PK)	TEXT	Course code
COURSE_NO (PK)	TEXT	Course number
COURSE_TITLE	TEXT	Course title ex: Operating Systems
CREDIT_HOURS	TEXT	Hours for courses ex: 3, 4 or 2
COLLEGE_CODE	TEXT	College code ex: IT, ENG or ...
DEPARTMENT_CODE	TEXT	Department code in college
PREREQ_CODE	TEXT	this for if this course has prerequisite or not
PREREQ_NO	TEXT	this for if this course has prerequisite or not
COURSE_TYPE	TEXT	Mandatory – Elective – Free Course – Same Major – Same College – Has Pre-Requisite – University Requirements

Table 3.11: TSECTIONS. This table contains the information for courses sections and dates for it

Attribute	Type	Description
SEMESTER_DATE (PK)	TEXT	What was course date
COURSE_CODE (PK)	TEXT	Course code example (cpit)
COURSE_NO (PK)	TEXT	Course no as (416)
SECTION_NO (PK)	TEXT	Section number or alphabet ex:(AB1)
COURSE_DATE	TEXT	Course date
COURSE_TIME	TEXT	Time for specific section
DAYS	TEXT	Section days for course
EVALUATION_TYPE	TEXT	Exam or assignments or open book ...
STUDENT_COUNTER	TEXT	Real students seats in class

Table 3.12: TPLANS. This table contains all departments and versions for plans

Attribute	Type	Description
PLAN_CODE (PK)	TEXT	Plan code for the department
COURSE_CODE (PK)	TEXT	Course code in the plan
COURSE_NO (PK)	TEXT	Course number in the plan
COURSE_TYPE	TEXT	Mandatory – Elective – Free Course – Same Major – Same College – Has Pre Requisite – University Requirements
AREA	TEXT	Course level or semester number

Table 3.13: TPREREUISTE. This table contains prerequisite for courses

Attribute	Type	Description
PREREQ_CODE (PK)	TEXT	Prerequisite code
PREREQ_NO (PK)	TEXT	Prerequisite number
COURSE_CODE (PK)	TEXT	Course code
COURSE_NO (PK)	TEXT	Course number

Table 3.14: TGRADES. This table contains grades and scores for all student dataset

Attribute	Type	Description
STUDENT_NO (PK)	TEXT	Student ID
SEMESTER_DATE (PK)	TEXT	Date for this semester
COURSE_CODE (PK)	TEXT	Course code
COURSE_NO (PK)	TEXT	Course number
SECTION_NO	TEXT	Section number
GRADE_CHAR	TEXT	Ex: A+, A, B+, B or ...
GRADE	TEXT	Number ex:95, 83 or ...

Table 3.15: CITY_LOC. This table contains the cities name

Attribute	Type	Description
CITY_CODE (PK)	TEXT	City code
CITY_TITLE	TEXT	City name

Table 3.16: COLLEGE_LOC. This table for colleges

Attribute	Type	Description
COLLEGE_CODE (PK)	TEXT	College code
COLLEGE_TITLE	TEXT	College name

Table 3.17: CONTRACT_LOC. This table contains contract type

Attribute	Type	Description
CONTRACT_TYPE (PK)	TEXT	
CONTRACT_TITLE	TEXT	

Table 3.18: DEPARTMENT_LOC. This table contains departments

Attribute	Type	Description
DEPARTMENT_CODE (PK)	TEXT	Department number
DEPARTMENT_TITLE	TEXT	Department name

Table 3.19: EVALUATION_LOC. This table contains evaluation type

Attribute	Type	Description
EVALUATION_TYPE (PK)	TEXT	Evaluation code
EVALUATION_TITLE	TEXT	Evaluation name

Table 3.20: GENDER_LOC. This table contains gender type

Attribute	Type	Description
GENDER (PK)	TEXT	Gender code
GENDER_TITLE	TEXT	Gender name

Table 3.21: LANGUAGE_LOC. This table contains languages

Attribute	Type	Description
LANGUAGE (PK)	TEXT	Language code
LANGUAGE_TITLE	TEXT	Language name

Table 3.22: NATIONALITY_LOC. This table contains nationalities

Attribute	Type	Description
NATIONALITY (PK)	TEXT	Nationality code
NATIONALITY_TITLE	TEXT	Nationality name

Table 3.23: SCHOOL_LOC. This table contains schools' type

Attribute	Type	Description
SCHOOL_TYPE (PK)	TEXT	School type code
SCHOOL_TITLE	TEXT	International – Local - Abroad

Table 3.24: STATUS_LOC. This table contains all status in the system

Attribute	Type	Description
STATUS (PK)	TEXT	Status code
STATUS_TITLE	TEXT	Status title from Start to Graduation

Here are some queries the researcher uses it to retrieve data from the SIS system.

1. This table to determine students.

```

04/01/2020 17:35:20
1  --THIS TABLE TO DETERMINE STUDENTS
2  CREATE TABLE DRMTUPIDMSALL AS
3  SELECT SGBSTN_PIDM,COUNT(*) PROG_COUNT
4  FROM
5  (
6  SELECT DISTINCT SGBSTN_PIDM,SGBSTN_PROGRAM_1 FROM SGBSTN
7  WHERE
8  --SGBSTN_STST_CODE = 'GR'
9  --AND SGBSTN_LEVEL_CODE = 'UG'
10 SGBSTN_PROGRAM_1 IS NOT NULL
11 AND SGBSTN_PIDM IN
12 (
13 SELECT SGBSTN_PIDM FROM SGBSTN A
14 WHERE
15 SGBSTN_TERM_CODE_EFF = (SELECT MAX(SGBSTN_TERM_CODE_EFF) FROM SGBSTN B WHERE B.SGBSTN_PIDM = A.SGBSTN_PIDM)
16 AND SGBSTN_LEVEL_CODE = 'UG'
17 AND SGBSTN_STST_CODE = 'GR'
18 )
19 )
20 GROUP BY SGBSTN_PIDM

```

2. Then we were filter more than one time to reach the faculties of arts, sciences, and computers from 2010 to 2015 graduates only from them and taking into account the stability of the plan and not change it either from the department or that the student was on another department and made the transfer-- check no of change plan for each student.

```

21
22 --THEN WE WAS FILTER MORE THAN ONE TIME TO REACH THE FACULTIES OF ARTS, SCIENCES AND COMPUTERS FROM 2010 TO 2015 GRADUATES
23 --ONLY FROM THEM AND TAKING INTO ACCOUNT THE STABILITY OF THE PLAN AND NOT CHANGE IT EITHER FROM THE DEPARTMENT OR THAT
24 --THE STUDENT WAS ON ANOTHER DEPARTMENT AND MADE THE TRANSFER-- CHECK NO OF CHANGE PLAN FOR EACH STUDENT.
25 SELECT DISTINCT SGBSTN_PROGRAM_1 FROM SGBSTN
26 WHERE
27 SGBSTN_STST_CODE = 'GR'
28 AND SGBSTN_PROGRAM_1 IS NOT NULL
29 AND SGBSTN_PIDM IN
30 (
31 SELECT SGBSTN_PIDM FROM DRMTUPIDMSALL A
32 WHERE
33 PROG_COUNT = '1'
34 )
35 AND SGBSTN_PROGRAM_1 LIKE '%-SC%'

```


06/01/2020 14:00:48

TSTUDENT_INFORMATION

Field Name	Data Type	Description (Optional)
STUDENT_NO	Short Text	STUDENT NUMBER
ST_PIDM	Number	THIS LIKE STUDENT NUMBER ITS A PRIMARY KEY FOR STUDENT (BANNER REGULATION)
COLLEGE_CODE	Short Text	COLLEGE CODE IN LOOKUP
PLAN_CODE	Short Text	PLAN CODE IN LOOKUP
SCHOOL_CODE	Short Text	SCHOOL TYPE IN LOOKUP
GENDER	Short Text	MALE OR FEMALE
QTY_CODE	Short Text	STUDENT GRADUATED QTY
BIRTH_DATE	Long Text	STUDENT BIRTH DATE
HIGH_SCHOOL_GPA	Long Text	STUDENT HIGH SCHOOL GPA
UNIVERSITY_GPA	Number	STUDENT UNIVERSITY GPA
SPECIALIZED_GPA	Number	STUDENT GPA IN HIS MAJOR
ADMYEAR	Short Text	STUDENT ADMISSION YEAR
GRDYEAR	Short Text	MALE OR FEMALE
STUDY_PERIOD_YEAR	Number	HOW MANY YEARS THE STUDENT SPENT TO GRADUATE
LEGAL_WITHDRAW	Number	HOW MANY STUDENT DROP SEMETER
STUDY_PERIOD_SEMESTER	Number	SEMESTER COUNTER
SUMMER_SEMESTERS	Number	HOW MANY TIMES STUDENT TAKE SUMMER SEMESTER
NATIONALITY	Long Text	STUDENT NATIONALITY

6. Filling the PLANS according to the plan code in the TSTUDENT_INFO table with this query, thus completing the table.

```

119  --FILLING THE PLANS ACCORDING TO THE PLAN CODE IN THE TSTUD_INFO TABLE WITH THIS QUERY, THUS COMPLETING THE TABLE
120  CREATE TABLE TPLANS AS
121  SELECT S.MBPGEN_PROGRAM PLAN_CODE, S.MBPAAP_AREA SEMESTER_LEVEL, S.SHRACAA_SFQNO COURSE_SEQ, S.SHRACAA_SUBJ_CODE COURSE_CODE
122  , S.SHRACAA_CRSE_NUMB LOW COURSE_NO, ' ' COURSE_TYPE
123  FROM S.MBPGEN A, S.MBPAAP C, S.SHRACAA E
124  WHERE
125  S.MBPGEN_TERM_CODE_EFF = (SELECT MAX(S.MBPGEN_TERM_CODE_EFF) FROM S.MBPGEN B WHERE B.SMBPGEN_PROGRAM=A.SMBPGEN_PROGRAM)
126  AND S.MBPAAP_TERM_CODE_EFF = (SELECT MAX(S.MBPAAP_TERM_CODE_EFF) FROM S.MBPAAP D WHERE D.SMBPAAP_AREA=C.SMBPAAP_AREA AND D.SMBPAAP_PROGRAM=A.SMBPGEN_PROGRAM)
127  AND S.SHRACAA_TERM_CODE_EFF = (SELECT MAX(S.SHRACAA_TERM_CODE_EFF) FROM S.SHRACAA F WHERE F.SHRACAA_AREA=E.SHRACAA_AREA)
128  AND C.SMBPAAP_PROGRAM=A.SMBPGEN_PROGRAM
129  AND C.SMBPAAP_AREA=E.SHRACAA_AREA
130  AND S.MBPGEN_PROGRAM IN
131  (
132  SELECT PLAN_CODE FROM TSTUD_INFO
133  )
134

```

06/01/2020 14:08:28

TPLANS

Field Name	Data Type	Description (Optional)
PLAN_CODE	Short Text	PLANE CODE
SEMESTER_LEVEL	Short Text	COURSES SET LEVEL
COURSE_SEQ	Number	COURSE SEQUENCE
COURSE_CODE	Short Text	COURSE CODE
COURSE_NO	Short Text	COURSE NUMBER
COURSE_TYPE	Short Text	COURSE TYPE

7. Table of courses the fifth table from the reality of the courses found in the plans in the seventh table through the next query.

```

135  --TABLE OF COURSES THE FIFTH TABLE FROM THE REALITY OF THE COURSES FOUND IN THE PLANS IN THE SEVENTH TABLE THROUGH THE NEXT QUERY
136  CREATE TABLE TCOURSE_INFORMATION AS
137  SELECT DISTINCT COURSE_CODE, COURSE_NO, SCBCRSE_TITLE, SCBCRSE_CREDIT_HR_LOW, SCBCRSE_COLL_CODE, SCBCRSE_DEPT_CODE
138  FROM TPLANS, SCBCRSE A
139  WHERE
140  COURSE_NO IS NOT NULL
141  AND TRIM(COURSE_CODE) = TRIM(SCBCRSE_SUBJ_CODE)
142  AND TRIM(COURSE_NO) = TRIM(SCBCRSE_CRSE_NUMB)
143  AND SCBCRSE_EFF_TERM =
144  (
145  SELECT MAX(SCBCRSE_EFF_TERM)
146  FROM SCBCRSE B
147  WHERE
148  B.SCBCRSE_SUBJ_CODE = A.SCBCRSE_SUBJ_CODE AND B.SCBCRSE_CRSE_NUMB = A.SCBCRSE_CRSE_NUMB
149  )
150

```

06/01/2020 14:13:01

TCOURSE INFORMATION

Field Name	Data Type	Description (Optional)
COURSE_CODE	Short Text	COURSE CODE
COURSE_NO	Short Text	COURSE NUMBER
COURSE_TITLE	Short Text	COURSE NAME
CREDIT_HOURS	Short Text	CREDIT HOURS
COLLEGE_CODE	Short Text	COLLEGE CODE
DEPARTMENT_CODE	Short Text	DEPARTMENT CODE IN COLLEGE

8. Prerequisite table

```

151  --PREREQUISITE TABLE
152  CREATE TABLE TPREREQUISITE AS
153  SELECT SCRTST_SUBJ_CODE COURSE_CODE, SCRTST_CRSE_NUMB COURSE_NO, SCRTST_SUBJ_CODE_PREQ PRE_CODE
154  ,SCRTST_CRSE_NUMB_PREQ PRE_NO, SCRTST_CONNECTOR FLAG
155  FROM SCRTST A
156  WHERE
157  SCRTST_TERM_CODE_EFF =
158  ( SELECT MAX(SCRTST_TERM_CODE_EFF) FROM SCRTST B
159  WHERE B.SCRTST_SUBJ_CODE = A.SCRTST_SUBJ_CODE
160  AND B.SCRTST_CRSE_NUMB = A.SCRTST_CRSE_NUMB
161  )
162  AND TRIM(SCRTST_SUBJ_CODE)||TRIM(SCRTST_CRSE_NUMB) IN
163  (
164  SELECT TRIM(COURSE_CODE)||TRIM(COURSE_NO) FROM TCOURSE_INFORMATION
165  )
166  AND SCRTST_LVL_CODE='00'
167  ORDER BY 1,2
168
169

```

06/01/2020 14:15:04

TPREREQUISITE

Field Name	Data Type	Description (Optional)
COURSE_CODE	Short Text	COURSE CODE
COURSE_NO	Short Text	COURSE NUMBER
PRE_CODE	Short Text	PREREQUISITE COURSE CODE
PRE_NO	Short Text	PREREQUESTE COURSE NUMBER
FLAG	Short Text	FLAG

9. Then the grades table, the ninth table by the next query

```

170  --THEN THE GRADES TABLE, THE NINTH TABLE BY THE NEXT QUERY
171  CREATE TABLE TGRADES AS
172  SELECT SFRSTCR_PIDM_CODE, SSBSECT_CRN, SFRSTCR_PIDM, F_GET_STD_ID(SFRSTCR_PIDM) STUDENT_NO, SSBSECT_TERM_CODE SEMESTER_DATE
173  , SSBSECT_SUBJ_CODE COURSE_CODE, SSBSECT_CRSE_NUMB COURSE_NO, SSBSECT_SEQ_NUMB SECTION_NO, 'XXX' GRADE_CHAR, 'XXX' GRADE
174  FROM SFRSTCR, SSBSECT
175  WHERE
176  SFRSTCR_PIDM IN (SELECT SOBSTIN_PIDM FROM STR.DRNTWALLSTUD3)
177  AND SFRSTCR_RSTS_CODE IN ('RE', 'AR', 'RW')
178  AND SSBSECT_TERM_CODE = SFRSTCR_TERM_CODE
179  AND SFRSTCR_CRN = SSBSECT_CRN
180

```

06/01/2020 14:17:58

TGRADES

Field Name	Data Type	Description (Optional)
STUDENT_NO	Short Text	STUDENT NUMBER
SEMESTER_DATE	Short Text	SEMESTER DATE
COURSE_CODE	Short Text	COURSE CODE
COURSE_NO	Short Text	COURSE NUMBER
SECTION_NO	Short Text	SECTION NUMBER
GRADE_CHAR	Short Text	GRADE CHARACTER
GRADE	Short Text	GRADE NUMBER

10. This table for sections information.

```

181  --THIS TABLE FOR SECTIONS INFORMATION
182  CREATE TABLE TSECTIONS AS
183  SELECT SSBSECT_TERM_CODE SEMESTER_DATE, SSBSECT_SUBJ_CODE COURSE_CODE, SSBSECT_CRSE_NUMB COURSE_NO, SSBSECT_SEQ_NUMB SECTION_NO
184  , SSBSECT_SCHED_CODE SECTION_TYPE, SSBSECT_BEGIN_TIME SECTION_START_TIME, SSBSECT_END_TIME SECTION_END_TIME
185  , INVL(SSBSECT_BGN_DAY, ' ') || INVL(SSBSECT_BGN_DAY, ' ') || INVL(SSBSECT_TUE_DAY, ' ') || INVL(SSBSECT_WED_DAY, ' ')
186  || INVL(SSBSECT_THU_DAY, ' ') || INVL(SSBSECT_FRI_DAY, ' ') || INVL(SSBSECT_SAT_DAY, ' ') DAYS, SSBSECT_EVALUATION_TYPE
187  , SSBSECT_SEQ_NUMB STUDENT_COUNTS, STR_F_GET_TEACHER_NO SSBSECT_TERM_CODE, SSBSECT_CRN TEACHER_NO
188  FROM SSBSECT
189  WHERE
190  SSBSECT_TERM_CODE || SSBSECT_CRN IN
191  (
192  SELECT DISTINCT SEMESTER_DATE || SSBSECT_CRN FROM TGRADES
193  )
194  AND SSBSECT_TERM_CODE = SSBSECT_TERM_CODE
195  AND SSBSECT_CRN = SSBSECT_CRN
196

```

06/01/2020 14:19:15

TSECTIONS

Field Name	Data Type	Description (Optional)
SEMESTER_DATE	Short Text	SEMESTER DATE
COURSE_CODE	Short Text	COURSE CODE
COURSE_NO	Short Text	COURSE NUMBER
SECTION_NO	Short Text	SECTION NUMBER
SECTION_TYPE	Short Text	SECTION TYPE
SECTION_START_TIME	Short Text	START TIME
SECTION_END_TIME	Short Text	END TIME
DAYS	Short Text	DAYS
EVALUATION_CODE	Short Text	EVALUATION CODE
STUDENT_COUNTER	Short Text	STUDENT COUNTER
EMP_NO	Short Text	FACULTY MEMBER WHO TEACH THIS COURSE

Data Exploration

The data were extracted more than once from these sources (Online/Direct access to KAU databases without permission authentication, authorization, security connection and Non-Disclosure Agreement is not allowed). In this task the researcher describes results of the data exploration and creates DMDIM including first findings or initial hypothesis and their impact on the remainder of the project. During this stage we will address data mining questions using querying, data visualization and reporting techniques. Data Mining Difficulty Index Model (DMDIM) database were mapped from the Enterprise Resource Planning (ERP) database, Students Information System (SIS) database, Alumni System (AS) database, Questioner System (Estebana) database and Yasser (Noor) database into tables in SQL server database. Multiple SQL queries were run on the tables to create the relational dataset, as each variable was added to the dataset it was move into data cleansing model.

Data Quality Report

In this task the researcher used different tools to list the results of the data quality verification. The researcher looks for the following types of problems:

- Missing data include values that are blank or coded as a non-response (such as \$null\$,?, or 999).
- Data errors are usually typographical errors made in entering the data.
- Coding inconsistencies typically involve nonstandard units of measurement or value inconsistencies, such as the use of both M and male for gender.
- Bad metadata include mismatches between the apparent meaning of a field and the meaning stated in a field name or definition.

The researcher searches for answers to all these problems and reflects them on the dataset as an example using JUPYTER notebook as shown in Figure 3.11.

```

Handeling Missing Values

In [8]: # From Original File we will do:
# Count missing values in each column
# first, we find the null values by calling (isnull), this will return True/False if the value is null or not
# then, sum these values to get the count of null values in each column (True is equal to 1 and False is equal to 0)
#raw_data.isnull().sum()
#raw_data.isnull().sum().sort_values(ascending=False)

# check now the count of missing values.
print('missing values: ', df1.isnull().sum().sum())
null_columns = df1.isnull().sum()
for key,value in null_columns.iteritems():
    if value > 0:
        print(key," ",value)

missing values: 5370347
STD_AGE_BY_YEAR , 1346
STD_ADYYEAR , 1346
STD_STUDY_PERIOD_YEAR , 1346
CRS_SECTION_NO , 1
CRS_SECTION_NO , 2454
STD_GRADE_CHAR , 30
STD_GRADE , 26265
TCHR_GRADE_CODE , 143010
TCHR_GRADE_TITLE_AR , 143010
TCHR_GRADE_TITLE_ENG , 143010
TCHR_EXP_TEACH_MAJOR , 142996
TCHR_EXP_TEACH_MAJOR , 243688
TCHR_NATIONALITY_CODE , 142996
TCHR_NATIONALITY_TITLE_AR , 142996
TCHR_NATIONALITY_TITLE_ENG , 142996
TCHR_PREV_NATIONALITY_CODE , 166403
TCHR_PREV_NATIONALITY_TITLE_AR , 170372
TCHR_PREV_NATIONALITY_TITLE_ENG , 166403
TCHR_GENDER , 142996
TCHR_GENDER_TITLE_AR , 142996
TCHR_GENDER_TITLE_ENG , 142996
TCHR_BIRTHDATE , 142996
TCHR_AGE_TEACH_COURSE , 142996
TCHR_BIRTHPLACE_AR , 142996
TCHR_BIRTHPLACE_ENG , 142996
TCHR_LANGUAGES_CODE , 142996
TCHR_FRST_WRK_DTE_UNV , 142996
TCHR_RECRUITING_CODE , 142996
TCHR_RECRUITING_TITLE_AR , 142996
TCHR_RECRUITING_TITLE_ENG , 142996
TCHR_JOB_TYPE_CODE , 142996
TCHR_JOB_NAME_AR , 142996
TCHR_JOB_NAME_ENG , 142996
TCHR_CERT_CODE , 144349
TCHR_CERT_TITLE_AR , 144349
TCHR_CERT_TITLE_ENG , 144349
TCHR_CERT_GRADE_CODE , 144386
TCHR_CERT_GRADE_TITLE_AR , 144386
TCHR_CERT_GRADE_TITLE_ENG , 144386
TCHR_CERT_COUNTRY_CODE , 144362
TCHR_COUNTRY_TITLE_AR , 144417
TCHR_COUNTRY_TITLE_ENG , 144417
TCHR_CERT_SITE , 145338

In [8]: # After make some updates in Original File from the KAU sorces we got this Result:
# Count missing values in each column
# first, we find the null values by calling (isnull), this will return True/False if the value is null or not
# then, sum these values to get the count of null values in each column (True is equal to 1 and False is equal to 0)

# check now the count of missing values.
print('missing values: ', df2.isnull().sum().sum())
null_columns = df2.isnull().sum()
for key,value in null_columns.iteritems():
    if value > 0:
        print(key," ",value)

missing values: 189053
CRS_SECTION_NO , 2454
STD_GRADE , 26208
TCHR_PREV_NATIONALITY_TITLE_AR , 3969
TCHR_CERT_CODE , 1353
TCHR_CERT_TITLE_ENG , 1353
TCHR_CERT_GRADE_CODE , 1390
TCHR_CERT_GRADE_TITLE_AR , 144386
TCHR_CERT_GRADE_TITLE_ENG , 1390
TCHR_CERT_COUNTRY_CODE , 1366
TCHR_COUNTRY_TITLE_AR , 1421
TCHR_COUNTRY_TITLE_ENG , 1421
TCHR_CERT_SITE , 2342

```

Figure 3.11: Handling Missing Data

2.2. Data Preparation

Data preparation is one of the most important and highly affects any used data mining algorithm. It is estimated that data preparation usually takes 50-70% of a project's time and effort. Data preparation typically involves the following tasks, merging data sets and/or records, selecting a sample subset of data, aggregating records, deriving new attributes, sorting the data for modeling, Removing or replacing blank or missing values and splitting into training and test data sets.

Data Selection

This is the stage of the project where you decide on the data that you're going to use for analysis. The researcher selects 45 attributes from DMDIM dataset as shown in Figure 3.12.

```
## DMDIM Dataset
This dataset contains the recorded information about the students, courses and faculty member from King Abdulaziz University (KAU). The dataset contains the following variables:
```

Column	DESCRIPTION	Dtype
>> STD_STUDENT_ID	STUDENT IDENTIFICATION	TEXT
>> STD_SCHOOL_TYPE	STUDENT HIGH SCHOOL TYPE	TEXT
>> STD_GENDER	STUDENT GENDER CODE	TEXT
>> STD_GENDER_TITLE	STUDENT GENDER NAME	TEXT
>> STD_HOME_CITY	STUDENT HOME CITY CODE	TEXT
>> STD_CITY_TITLE	STUDENT HOME CITY ARABIC NAME	TEXT
>> STD_CITY_TITLE_ENG	STUDENT HOME CITY ENGLISH NAME	TEXT
>> STD_BIRTH_DATE	STUDENT BIRTH DATE	DATE
>> STD_AGE_BY_YEAR	STUDENT AGE	NUMBER
>> STD_HIGH_SCHOOL_GPA	STUDENT HIGH SCHOOL GPA	FLUAT
>> STD_UNIVERSITY_GPA	STUDENT UNIVERSITY GPA	FLUAT
>> STD_SPECIALIZED_GPA	STUDENT SPECIALIZED GPA	FLUAT
>> STD_ADMYEAR	STUDENT ADMISSION DATE	NUMBER
>> STD_GRYEAR	STUDENT GRADUATION DATE	NUMBER
>> STD_STUDY_PERIOD_YEAR	STUDENT STUDY PERIOD BY YEAR	NUMBER
>> STD_LEGAL_WITHDRAW	STUDENT LEGAL WITHDRAW SEMESTER	NUMBER
>> STD_STUDY_PERIOD_SEMESTER	STUDENT STUDY PERIOD BY SEMESTER	NUMBER
>> STD_SUMMER_SEMESTERS	STUDENT STUDY STUDY SIN SUMMER	NUMBER
>> STD_NATIONALITY_CODE	STUDENT NATIONALITY CODE	TEXT
>> STD_NATIONALITY_NAME_AR	STUDENT NATIONALITY NAME ARABIC	TEXT
>> STD_NATIONALITY_NAME_ENG	STUDENT NATIONALITY NAME ENGLISH	TEXT
>> CFS_COLLEGE_CODE	COLLEGE CODE	TEXT
>> CFS_COLLEGE_NAME	COLLEGE NAME WHICH CONTAIN COURSE	TEXT
>> CFS_PROGRAM	COURSE PROGRAM IN COLLEGE	TEXT
>> CFS_SEMESTER_DATE	COURSE DATE	TEXT
>> CFS_COURSE_CODE	COURSE CODE	TEXT
>> CFS_CSCOURSE_NO	COURSE NUMBER	TEXT
>> CFS_COURSE_TITLE	COURSE TITLE IN ARABIC ENG	TEXT
>> CFS_SECTION_NO	COURSE SECTION NUMBER	TEXT
>> STD_GRADE_CHAR	STUDENT GRADE SYMBOL	TEXT
>> STD_GRADE	STUDENT GRADE AS INTEGER	NUMBER
>> CFS_STUDENT_COUNTER	STUDENT COUNTER IN SECTION	TEXT
>> TOR_ID	TEACHER IDENTIFICATION	TEXT
>> TOR_GRADE_CODE	TEACHER GRADE NUMBER	TEXT
>> TOR_GRADE_TITLE_AR	TEACHER GRADE TITLE IN ARABIC	TEXT
>> TOR_GRADE_TITLE_ENG	TEACHER GRADE TITLE IN ENGLISH	TEXT
>> TOR_EXP_TEACH_MAJOR	TEACHER TEACHING EXPERIENCE MAJOR	TEXT
>> TOR_EXP_TEACH_MINOR	TEACHER TEACHING EXPERIENCE MINOR	TEXT
>> TOR_NATIONALITY_CODE	TEACHER NATIONALITY CODE	TEXT
>> TOR_NATIONALITY_TITLE_AR	TEACHER NATIONALITY TITLE ARABIC	TEXT
>> TOR_NATIONALITY_TITLE_ENG	TEACHER NATIONALITY TITLE ENGLISH	TEXT
>> TOR_PREV_NATIONALITY_CODE	TEACHER PREVIOUS NATIONALITY CODE	TEXT
>> TOR_PREV_NATIONALITY_TITLE_AR	TEACHER PREVIOUS NATIONALITY NAME ARABIC	TEXT
>> TOR_PREV_NATIONALITY_TITLE_ENG	TEACHER PREVIOUS NATIONALITY NAME ENGLISH	TEXT
>> TOR_GENDER	TEACHER GENDER CODE	TEXT
>> TOR_GENDER_TITLE_AR	TEACHER GENDER TITLE ARABIC	TEXT
>> TOR_GENDER_TITLE_ENG	TEACHER GENDER TITLE ENGLISH	TEXT
>> TOR_BIRTH_DATE	TEACHER BIRTHDATE	DATE
>> TOR_AGE_TEACH_COURSE	TEACHER AGE WHEN TEACHING THE COURSE	NUMBER
>> TOR_BIRTHPLACE_AR	TEACHER BIRTHPLACE AR	TEXT
>> TOR_BIRTHPLACE_ENG	TEACHER BIRTHPLACE ENG	TEXT
>> TOR_LANGUAGES_CODE	TEACHER LANGUAGES CODE	TEXT
>> TOR_LANGUAGES_NAME	TEACHER LANGUAGES NAME	TEXT
>> TOR_FIRST_WORK_DATE_BY_KAU	TEACHER FIRST WORK DATE BY KAU	TEXT
>> TOR_RECRUITING_CODE	TEACHER RECRUITING TYPE CODE	TEXT
>> TOR_RECRUITING_TITLE_AR	TEACHER RECRUITING TYPE NAME ARABIC	TEXT
>> TOR_RECRUITING_TITLE_ENG	TEACHER RECRUITING TYPE NAME ENGLISH	TEXT
>> TOR_JOB_TYPE_CODE	TEACHER JOB TYPE CODE	TEXT
>> TOR_JOB_NAME_AR	TEACHER JOB TYPE NAME ARABIC	TEXT
>> TOR_JOB_NAME_ENG	TEACHER JOB TYPE NAME ENGLISH	TEXT
>> TOR_CERT_CODE	TEACHER CERTIFICATION CODE	TEXT
>> TOR_CERT_TITLE_AR	TEACHER CERTIFICATION ARABIC	TEXT
>> TOR_CERT_TITLE_ENG	TEACHER CERTIFICATION ENGLISH	TEXT
>> TOR_CERT_GRADE_CODE	TEACHER CERTIFICATE GRADE CODE	TEXT
>> TOR_CERT_GRADE_TITLE_AR	TEACHER CERTIFICATE GRADE TITLE ARABIC	TEXT
>> TOR_CERT_GRADE_TITLE_ENG	TEACHER CERTIFICATE GRADE TITLE ENGLISH	TEXT
>> TOR_CERT_COUNTRY_CODE	TEACHER COUNTRY CERTIFICATE CODE	TEXT
>> TOR_CERT_COUNTRY_TITLE_AR	TEACHER COUNTRY CERTIFICATE TITLE ARABIC	TEXT
>> TOR_CERT_COUNTRY_TITLE_ENG	TEACHER COUNTRY CERTIFICATE TITLE ENGLISH	TEXT
>> TOR_CERT_SITE	TEACHER GRADUATION FROM	TEXT

```
</div>
<div class="alert alert-block alert-info">
For more information about case study visit: <a href="url">https://www.kau.edu.sa/Home.aspx/a</a>
</div>
```

Figure 3.12: The Selected Data Attributes

Cleaning Data

Data cleaning includes a closer look at problems in the data the researcher has chosen to involve raise the data quality to the level required by the analysis techniques that he selected. the process of detecting and correcting or removing corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate, or irrelevant parts of the data and then replacing, modifying, or deleting the dirty data. The following are the main tasks employed for data validation and quality:

- Missing data required searching in the different data sources, comparing between records, and exchanging values between matching records from both databases, or deriving data from other fields in the same database. For example, remove the student's records that didn't complete and updated his record. So, the number of records has been reduced of about 5000 students ... after the exclusion of students who did not fulfil their data.
- Data inconsistency was a common factor in most of the fields with data contradicting the main domain data type and length for specific fields. Standardizing and unifying data type, length, and format was essential. Below are some examples of inconsistent data along with the solutions used to restore consistency:
 - ❖ Gender was also typed in different formats, either Boolean or string (0, 1, M, F, male, female) and sometimes two genders were typed in the same field. The type was standardized and verified from other databases.
 - ❖ Student' record numbers were identified in three different ways: as string, as number, and as string with different length. All these types were unified as a string with a fixed length to support searching and sorting issues.
 - ❖ Dates. Procedure date, admission date, date of surgery, and so on were typed in different formats (Janury-10-2010, 10-Janury 2010, 10/1/2009, 10-01-210), all these types of formats was standardized to a single type 'DATE' in the basic format "dd-mm-yyyy" or sometimes written in Islamic calendar.
 - ❖ Accordingly, several steps were considered to match the records correctly:
 - Verify dates of procedures because sometimes they were written correctly as a format but with mistaken values, and
 - Remove redundant records.

Figure 3.13 shows the distribution of KAU students' age before the cleansing, while Figure 3.14 shows their distribution after applying the cleansing process.

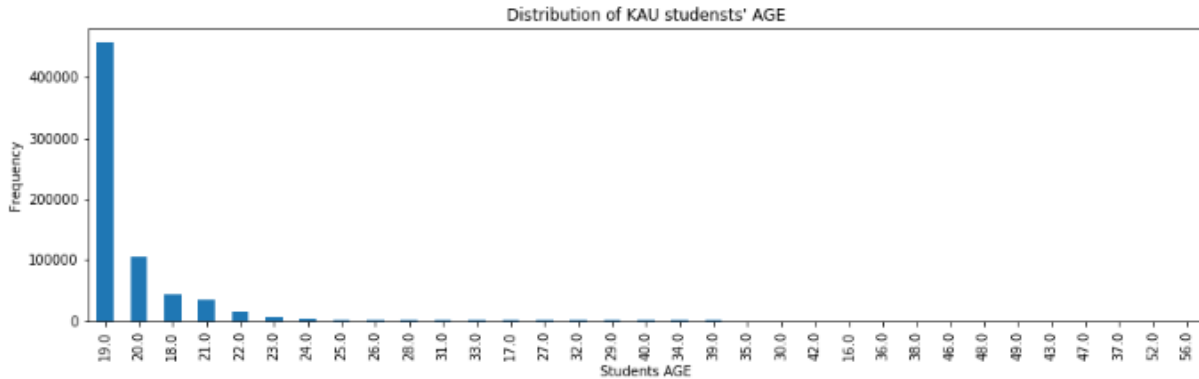


Figure 3.13: Distribution of KAU Students' Age – Before Cleansing

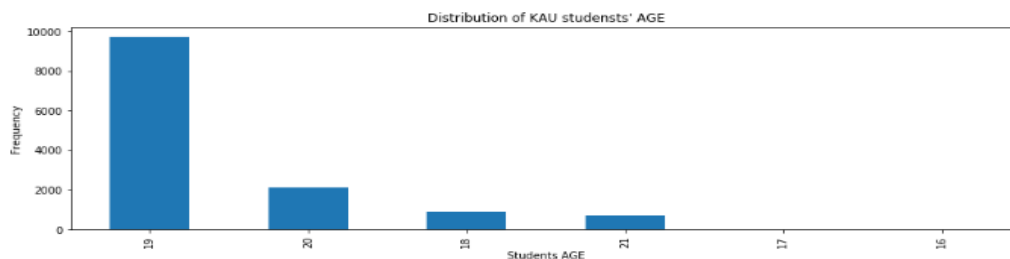


Figure 3.14: Distribution of KAU Students' Age – After Cleansing

Construct Required Data

In this task the researcher constructs new data he needs in DMDIM dataset and not found it in KAU sources.

- **Derived attributes** – There are new attributes that are constructed from one or more existing attributes in the same record such as teacher's experience in teaching the same course, the teacher's age at the time of teaching the specific course.
- **Generated records** – the researcher doesn't need to generate new records.

Integrate Data

In this task the researcher connects different databases as he mentioned in explore data task in data understanding phase. The researcher needed to combine tables and records to create new records and values.

- **Merged data** - Merging tables refer to joining together two or more tables that have different information about the same objects such as students' information tables. These tables can be merged together into a new table with one record for each store, combining fields from the source tables.
- **Aggregations** - Aggregations refers to operations in which new values are computed by summarising information from multiple records and/or tables.

Format Data

This is the final step before build the model, it is helpful to check whether certain techniques require a specific format or order to the data. The researcher used

- Rearranging attributes.
- Reordering records.
- Reformatted within-value.

CHAPTER IV. A Sequential Pattern Mining Approach

1. Introduction

The university period is one of the most decisive periods in one's life. Students often experience the university period as a very stressful time mainly due to the fear of failure [BCSJ17]. From time to time, the lack of success is caused by reasons related to the students themselves, including the freedom to plan their learning processes and the flexibility of the university schedules. According to Eurostat, the statistical office of the European Union (EU), over 3 million young people in EU had been to university but had discontinued their studies at some point in their life. The main reasons for not continuing their education were numerous: a desire to work instead, finding their studies uninteresting or not meeting their needs, family reasons, etc.

As it was described in previous chapters, long-term course planning (LTCP) is an important task in academic advising [ShAb20], and it aims to help students in proposing a course list of all future semesters so the dropout rate might be reduced. Nevertheless, LTCP is especially challenging for several reasons such as the number of constraints related to university regulations, the students' abilities, and background knowledge, or simply some personal preferences caused by external factors [WuHa05]. As a result, when building a study plan, a lot of different aspects are required to prioritize courses. Some of such aspects quantify the importance of including a specific course in the study plan (students' preferences, expected grades due to easy courses, etc.), whereas others rate the chronology of those courses (complexity of the semester based on the courses). LTCP, however, does not consider graduate students to model the course priority, which might be a good starting.

Taking all the above into consideration, chapter aims to provide a course index [GuKH21] based on the sequence of courses a student has taken already, and the grades obtained by graduate students that followed a similar sequence. This analysis not only provides an index for a specific course but also enables general paths (courses grouped by semesters) that should be followed. The performed recommendation does not take any external factor, but the know-how of the system. In other words, it is based on the paths followed by other students and the final grade of such students not only in a specific course but in the degree. To this aim, we propose (ES)²P (Evolutionary Search of Emerging Sequential Patterns), a sequential pattern mining algorithm [AgSr95] to extract general paths (a set of semesters with different courses per semester) that were frequently followed by excellent students, but infrequently or never followed by not so good students. In this regard, (ES)²P gathers two well-known tasks in descriptive analysis: sequential pattern mining [SrAg96] and emerging patterns [Dong99]. This

synergy is key to identify paths that provide a course recommendation for each student.

The proposal also deals with an extra issue: the provided set of solutions. Generally, frequent itemset mining algorithms [LuFV19] require a minimum frequency threshold value to be predefined, which is related to the number of solutions. It was studied [40] that a small change in that threshold value may lead to an extreme variation in the number of solutions as well as a significant increment in the execution time, especially on high-dimensional data [PaLV19]. Hence, to determine the right threshold value is key, and it is not trivial even though the user has a profound background in the application field (it needs to try different thresholds by guessing and re-executing the algorithms once and again until results are good enough). To achieve this goal, the proposed algorithm, which is based on evolutionary algorithms [VeLu16], can extract a reduced set of solutions without considering any frequency threshold. This proposal guides the search process through the growth ratio or difference in the frequency between two groups of students (excellent and not so good ones).

The rest of the chapter is organized as follows. Section 2 describes the proposed methodology, and Section 3 presents some experimental studies to demonstrate the performance of proposal. Finally, Section 4 shows some study cases for a real scenario.

2. An Evolutionary Algorithm for Searching Emerging Sequential Patterns

To understand the proposed algorithm, which is known as (ES)²P (Evolutionary Search of Emerging Sequential Patterns), it is important to describe first the data representation. Thus, the following subsection includes a description about how data is read and stored in memory. Next, a subsection describing the encoding criterion of the solutions is presented. Finally, the algorithm is explained and how the resulting subset of solutions is obtained is also described in detail.

2.1. Data representation

The original database is stored into two different data representations while data are being read, transaction by transaction. The idea behind these data representations is to provide a fast data access, and to avoid useless information to be maintained. First, data are kept in memory through a vertical data representation that creates a list of indices (sequences in data) in which each item appears at least once. To obtain the frequency or support of each single item in data is quite simple since the algorithm just needs to calculate the length of the list associated to the item at hand. Given two or more items, this vertical data representation allows to obtain the set of data records that have at least one common instance of such items. The only operation to be performed is an intersection of the lists associated to each item. Second, a horizontal data representation is performed where a list of indices is stored. Here, instead of index of the sequences in data, it stores the index of the itemsets in which the item appears. This second data

representation is based on a hashing function, so given a key k based on an item, it maps k to the corresponding set of indices on which k is included. Index values are in increasing order from sequence to sequence and it depends on the number of itemsets each sequence has, which is also saved.

Table 4.1 shows a sample sequence (transactional) database, and Figures 4.1, 4.2 and 4.3 illustrate the data representation followed by the proposed approach on the sample transactional dataset. The vertical representation (see Figure 4.1) includes eight different lists of indices, one per item in data. Since each index denotes the data record (sequence) in which the item appears, the length of the list is the frequency of each item in data. Hence, the item $\{a\}$ appears twice in the dataset (first and second sequence). The item $\{b\}$ also appears twice in the dataset (first and second sequence), even when it appears twice in the second sequence, that is, $\langle\{a,b\},\{b\},\{g,h\}\rangle$. Similarly, the horizontal data representation (see Figure 4.2) is responsible for storing the itemsets in which each item appears. The item $\{b\}$ appears in the first itemset (sequence with ID 1) as well as in the first and second itemsets of the sequence with ID 2. In other words, $\{b\}$ appears in the first, fourth and fifth itemsets from Table 4.1. Finally, the proposed data representation needs to store the number of itemsets included in each sequence (see Figure 4.3). The accumulated sum is maintained so the last value corresponds to the number of itemsets in data. This accumulated sum is useful to determine whether the horizontal representation values belong to one or different sequences. In other words, taking the vector of values shown in Figure 4.3, any horizontal data representation value in the range $[1,3]$ belongs to the first sequence; any horizontal data representation value in the range $[4,6]$ belongs to the second sequence; and any value in the range $[7,8]$ belongs to the third sequence.

Table 4.1: Sample Sequence Database

Sequence ID	Sequence
1	$\langle\{a,b\},\{c\},\{d,e,f\}\rangle$
2	$\langle\{a,b\},\{b\},\{g,h\}\rangle$
3	$\langle\{c\},\{d,e}\rangle$

a	b	c	d	e	f	g	h
1	1	1	1	1	1	2	2
2	2	3	3	3			

Figure 4.1: Vertical data representation of the proposed algorithm

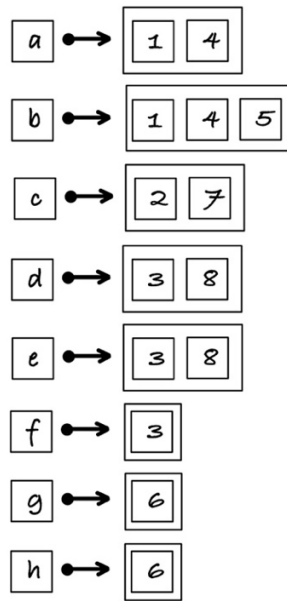


Figure 4.2: Vertical data representation of the proposed algorithm

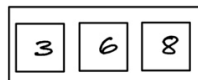


Figure 4.3: Number of itemsets in each sequence

The proposed data representation is used to compute the frequency in a fast way. The frequency of a single item is simply computed as the length of the vertical representation: the frequency of $\{a\}$ is 2, the frequency of $\{b\}$ is 2, etc. Additionally, the frequency of a sequence (including itemsets) is computed through both the vertical and horizontal representations. Let us consider the sequence $s = \langle \{c\}, \{d, e\} \rangle$. For this sequence, the intersection of the vertical data representation is performed for each single item, resulting as indices 1 and 3 (see Figure 4.1). At this point, it is required to check if every itemset in such sequences is also satisfied. The itemset $\{c\}$ appears in indices 2 and 7 according to the horizontal representation (see Figure 4.2). Due to the resulting indices from the vertical representation were 1 and 3, we have to check the first and third ranges of values from Figure 4.3 as follows: $2 \in [1, 3]$ and $7 \in [7, 8]$. As a result, the itemset $\{c\}$ is satisfied in the first and third sequence. Let us do the same for the itemset $\{d, e\}$, which appears in indices 3 and 8 according to the horizontal representation (see Figure 4.2). Again, due to the resulting indices from the vertical representation were 1 and 3, we have to check the first and third ranges of values from Figure 4.3 as follows: $3 \in [1, 3]$ and $8 \in [7, 8]$. As a result, the sequence s appears twice in data: first and third sequences.

2.2. Encoding criterion

The proposed algorithm uses an encoding vector of variable length to represent each individual or solution to the problem. The vector includes the set of items that belong to the represented solution. Such a set of items, in turn, is grouped into subsets that represent the itemsets within a sequence. The proposed encoding criterion includes the only restriction that the same item cannot appear twice in the same sequence. Since the algorithm was proposed for mining sequences of subjects (items) ordered by semesters (itemsets), it does not make sense to include the same subject twice in a sequence. For a matter of clarification, let $I = \{a, b, c, d\}$ be a sample set of items (subjects). A valid sequence is $s = \langle \{a, b\}, \{c\}, \{d\} \rangle$, denoting that a student passed the subjects a and b in a semester. Then, a semester later, the student passed the subject c . Finally, in a following semester, the student passed the subject d . An invalid solution would be $s = \langle \{a, b\}, \{a\} \rangle$ since once the subject a is passed by a student there is no sense in enrolling it again.

The proposed encoding criterion is easily adapted to the data representation since each item has two different pointers, one to the corresponding list of sequences (vertical data representation) and other to the list of itemsets (horizontal data representation). Thus, to determine in which sequences the items appear it only performs an intersection of the lists (vertical data representation) associated to the items in the sequence. Additionally, to obtain which itemsets appear in the sequences, it only has to perform an intersection of the lists (horizontal data representation) associated to the items for each itemset in the sequence. In order to clarify this methodology, let us consider again the valid sequence $s = \langle \{a, b\}, \{c\}, \{d\} \rangle$. The sequences in which the items belonging to s appear are obtained by intersecting the vertical representation of a , b , c and d . Additionally, the intersection of the lists obtained from the horizontal data representation returns the sequences in which the itemsets in s are satisfied. Such itemsets are $\{a, b\}$, $\{c\}$, and $\{d\}$.

2.3. Algorithm

The proposed algorithm comprises three main procedures, which were properly designed to the problem at hand. Descriptions of all these procedures as well as how they are combined to form the whole algorithm can be found below.

- 1) **Initial solutions.** The proposed algorithm creates the initial set of solutions randomly, each solution is a sequence including random itemsets. The number of itemsets that each solution (sequence) may include is limited by a maximum predefined value. Each itemset includes, in turn, a random subset of items from I . A maximum number of items per itemsets is also predefined. It is finally important to highlight that an item $i \in I$ cannot appear more than once in a sequence.

Table 4.2: Sample Database including sequence for good and not so good students

Sequence ID	Ω_1 (good students)
1	$\langle \{a,b\}, \{c\}, \{d,e,f\} \rangle$
2	$\langle \{a,b\}, \{c\}, \{e\} \rangle$
3	$\langle \{a\}, \{b,c,d,e\} \rangle$
4	$\langle \{c,d\}, \{g,h,k\}, \{l,m\} \rangle$
5	$\langle \{a,b,j\}, \{h,l,k\}, \{c\}, \{d,e\} \rangle$

Sequence ID	Ω_2 (not so good students)
1	$\langle \{b\}, \{a,c\}, \{f\} \rangle$
2	$\langle \{b,d\}, \{a,c\}, \{d,e\} \rangle$
3	$\langle \{d\}, \{a,b\}, \{c,e\} \rangle$
4	$\langle \{e\}, \{f,g,h\}, \{m\} \rangle$
5	$\langle \{a,d\}, \{b\}, \{k,l\} \rangle$

- 2) **Evaluation procedure.** This procedure is responsible for assigning a fitness value F to each individual or solution s . The evaluation procedure calculates how close a given solution is to the optimum solution on a dataset Ω . In the proposed approach, F for a solution s is formally defined based on the support of s and the growth ratio (GR) of s on the dataset Ω as shown in Equation 1. At this point, it is important to explain the concept of support and GR. The support, also known as frequency, is defined as the percentage of sequences in a dataset Ω that contains a specific sequence. It is also defined in terms of absolute values as the number of sequences in $S \subseteq \Omega$ that contains the sequence to be evaluated. Given a sequence s_j , the frequency of such a sequence in a database Ω is denoted as $\text{support}(s_j, \Omega)$ and formally defined, in relative terms. As a matter of exemplification, the sequence $\langle \{c\}, \{d\} \rangle$ has a support or frequency of 2 in absolute terms, or a value of 0.66 in relative terms when the dataset shown in Table 4.1 is considered. This sequence is satisfied by the sequences #1 and #3 from a total of 3 sequences included in this dataset. As for GR, it seeks for patterns whose frequency greatly differs from one group or dataset Ω_1 to another dataset Ω_2 . It is therefore defined as $\text{GR} = \text{support}(s_j, \Omega_1) / \text{support}(s_j, \Omega_2)$. Additionally, it is important to remark that F in Equation 1 is defined in the range $[0,1]$ and the best solutions are close to 1. Ω is split into two groups, that is, Ω_1 for good students (those that obtained a high final mark) and Ω_2 for not so good students. The fitness value F is based on the frequency of s in the subset of good students and the normalized growth rate obtained by s . In other words, a solution s is good if it

represents a high percentage of good students and a low percentage of not so good students. Finally, it is important to clarify that, in those situations where the difference in the frequencies between Ω_1 and Ω_2 is maximum, that is, $GR(s, \Omega_1, \Omega_2) = \infty$, only the frequency of s in Ω_1 is considered to compute F .

$$F(s, \Omega_1, \Omega_2) = \frac{\text{support}(s, \Omega_1)}{GR(s, \Omega_1, \Omega_2)} \times (GR(s, \Omega_1, \Omega_2) - 1) \quad (1)$$

For a matter of clarification, let us consider a sample dataset (see Table 4.2) that is divided into Ω_1 and Ω_2 . A sample solution $s_1 = \langle \{a, b\}, \{c\} \rangle$ appears in 60% of the good students (two first sequences as well as the last sequence in Ω_1). It is mathematically represented as the frequency or support(s_1, Ω_1) = 3/5 = 0.6. This solution s_1 appears in 20% of not so good students (third sequence in Ω_2), also denoted as support(s_1, Ω_2) = 1/5 = 0.2. It implies that $GR(s_1, \Omega_1, \Omega_2) = 0.6 = 3/5$ and, therefore, $F(s_1, \Omega_1, \Omega_2) = 0.40$. Let us now consider an additional sample solution $s_2 = \langle \{a, b\}, \{c\}, \{e\} \rangle$. Its support value in Ω_1 is calculated as support(s_2, Ω_1) = 3/5 = 0.6, whereas in Ω_2 it is calculated as support(s_2, Ω_2) = 0. Hence, $GR(s_2, \Omega_1, \Omega_2) = 0.6/0 = \infty$ and, therefore, the fitness value F is only computed as the frequency of s_2 in Ω_1 or $F(s_2, \Omega_1, \Omega_2) = \text{support}(s_2, \Omega_1) = 0.60$.

- 3) **Genetic operators.** The proposal includes two genetic operators: the crossover operator, which focuses on exploiting current individuals by examining their neighbors; the mutation operator, which aims to diversify the search process and to explore new areas in the search space. The crossover genetic operator (see Algorithm 1) combines generic material of two solutions to act as parents (p_1 and p_2) to generate offspring (o_1 and o_2).

This operator works as a two-points crossover by considering the itemsets as feasible points. Hence, it is not possible to split any itemset within p_1 or p_2 . An additional requirement of this operator is that the crossover points cannot produce the whole individual. This two-points crossover operator works differently depending on a probability. On the one hand, it leaves the itemsets of p_1 within the two cut-points unaltered (see lines 5 to 14, Algorithm 1). It adds the itemsets that appear out of the range of the cut-points of p_2 , but removing those items that were already in the itemsets obtained from p_1 . Additionally, it does the same for p_2 , that is, it takes the itemsets within the cut-points and adds those itemsets out of the cut-points from p_1 . Again, those items that were already included due to p_2 are removed. As a result, no item can appear more than once either in o_1 or in o_2 . On the other hand, the genetic operator leaves the itemsets of p_1 and p_2 outside the two cut-points unaltered (see lines 14 to 23, Algorithm 1). It then adds the corresponding itemsets within the two cut-points by removing repeated items. As a matter of clarification, Figure 4.4 illustrates an example of the proposed crossover operator. Let us consider the following individuals taken from a sample dataset (see Table 4.2) to act as parents $p_1 = \langle \{b\}, \{c\}, \{d, e, f\} \rangle$, $p_2 = \langle \{a, c\}, \{j,$

Algorithm 1 Pseudocode for crossover operator

Input Parent individuals p_1 and p_2
Output Offspring individuals o_1 and o_2

```

1:  $m_{p_1} \leftarrow$  number of itemsets in  $p_1$ 
2:  $m_{p_2} \leftarrow$  number of itemsets in  $p_2$ 
3:  $c_1^1, c_1^2 \leftarrow$  random cut-points in  $p_1$ 
4:  $c_2^1, c_2^2 \leftarrow$  random cut-points in  $p_2$ 
5: if  $\text{random}(0, 1) \leq 0.5$  then
6:    $o_1^{(c_1^1, c_1^2)} \leftarrow p_1^{(c_1^1, c_1^2)}$ 
7:    $o_2^{(c_2^1, c_2^2)} \leftarrow p_2^{(c_2^1, c_2^2)}$ 
8:   for each itemset  $X_j \mid j \in [0, c_1^1] \cup (c_1^2, m_{p_1}]$  do
9:      $o_2 \leftarrow o_2 \cup \{i \in X_j \mid i \notin o_2\}$ 
10:  end for
11:  for each itemset  $X_j \mid j \in [0, c_2^1] \cup (c_2^2, m_{p_2}]$  do
12:     $o_1 \leftarrow o_1 \cup \{i \in X_j \mid i \notin o_1\}$ 
13:  end for
14: else
15:    $o_1^{[0, c_1^1] \cup (c_1^2, m_{p_1})} \leftarrow p_1^{[0, c_1^1] \cup (c_1^2, m_{p_1})}$ 
16:    $o_2^{[0, c_2^1] \cup (c_2^2, m_{p_2})} \leftarrow p_2^{[0, c_2^1] \cup (c_2^2, m_{p_2})}$ 
17:   for each itemset  $X_j \mid j \in (c_1^1, c_1^2]$  do
18:      $o_2 \leftarrow o_2 \cup \{i \in X_j \mid i \notin o_2\}$ 
19:   end for
20:   for each itemset  $X_j \mid j \in (c_2^1, c_2^2]$  do
21:      $o_1 \leftarrow o_1 \cup \{i \in X_j \mid i \notin o_1\}$ 
22:   end for
23: end if
24: return  $o_1, o_2$ 
  
```

k }, $\{f, l\}$, $\{d, m\}$ >, and the cut-points marked as dotted lines. Let us also consider the range within the cut-points to be copied unaltered. Thus, o_1 is first initialized as $\langle \{b\}, \{c\} \rangle$. The itemsets outside the cut-points from p_2 are then added to o_1 , that is, $\{a, c\}$ and $\{d, m\}$. Since the item c is already in $\langle \{b\}, \{c\} \rangle$, then it is removed and the resulting offspring is $o_1 = \langle \{a\}, \{b\}, \{c\}, \{d, m\} \rangle$. Finally, o_2 is initialized as $\langle \{j, k\}, \{f, l\} \rangle$ and the itemsets outside the cut-points from p_1 are added: d, e, f . Due to f is already in o_2 it is removed and the offspring is finally formed as $\langle \{j, k\}, \{f, l\}, \{d, e\} \rangle$

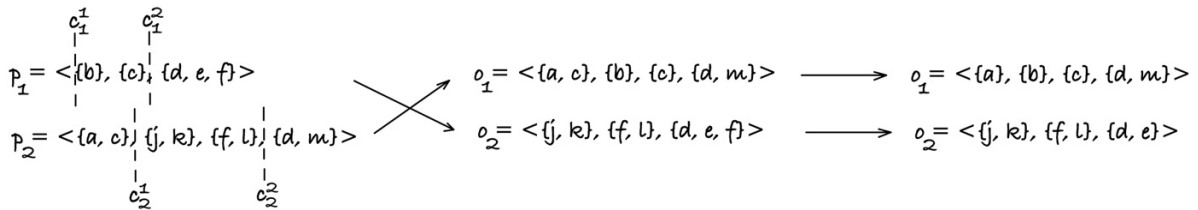


Figure 4.4: Crossover operator example

On the other hand, the mutation genetic operator (see Algorithm 2) has been designed to perform different tasks with a certain probability. It slightly modifies

solutions, looks for near neighbors, and seeks for far un- explored areas of the search space, maintaining part of the information of the original sequence (solution or individual). Given an individual, we first give the same probability to apply a more disruptive operator or a subtler one (see line 2, Algorithm 2). The disruptive option (see lines 2 to 10, Algorithm 2) randomly selects a cut-point within the solution and any itemset before that cut-point (see lines 5 to 7) or after it (see lines 7 to 9) is replaced by a set of itemsets generated randomly. It adds a random number of itemsets, but ensuring that the sequence does not exceed the maximum number of itemsets. Those items that are already included in p are not considered. On the contrary, the less disruptive option (see lines 11 to 24, Algorithm 2) provides three different options: 1) to include a new itemset at a random position of the sequence (see lines 12 to 15). Those items within such an itemset that are already included in p are not considered; 2) to remove a random itemset from p (see lines 16 to 18); 3) to replace a random itemset from p by a new one randomly generated (see lines 19 to 24).

Algorithm 2 Pseudocode for mutation operator

Input Parent individual p
Output New individual o

```

1:  $m_p \leftarrow$  number of itemsets in  $p$ 
2: if  $\text{random}(0, 1) \leq 0.5$  then
3:    $c \leftarrow$  random cut-point in  $p$ 
4:    $X \leftarrow$  generate random itemsets
5:   if  $\text{random}(0, 1) \leq 0.5$  then
6:      $o \leftarrow p^{[0,c]} \cup \{i \in X \mid i \notin p^{[0,c]}\}$ 
7:   else
8:      $o \leftarrow \{i \in X \mid i \notin p^{(c,m_p)}\} \cup p^{(c,m_p)}$ 
9:   end if
10: else
11:    $r \leftarrow \text{random}(0, 1)$ 
12:   if  $r \leq 0.33$  then
13:      $X \leftarrow$  generate a random itemset
14:      $c \leftarrow$  generate a random position within  $p$ 
15:      $o \leftarrow p^{[0,c]} \cup \{i \in X \mid i \notin p\} \cup p^{(c,m_p)}$ 
16:   else if  $r \leq 0.66$  then
17:      $X \leftarrow$  select a random itemset from  $p$ 
18:      $o \leftarrow p \setminus X$ 
19:   else
20:      $X^j \leftarrow$  select a random itemset from  $p$ 
21:      $o \leftarrow p \setminus X^j$ 
22:      $X^j \leftarrow$  generate a random itemset
23:      $o \leftarrow o \cup \{i \in X^j \mid i \notin o\}$   $\triangleright$  same position as  $X^j$  but removing duplicated items
24:   end if
25: end if
26: return  $o$ 

```

Again, those items within such an itemset that are already included in p are not considered. As a matter of clarification, let us consider the following individual to act as a

parent $p = \langle \{b\}, \{c\}, \{d, e, f\} \rangle$, which is a feasible solution from the sample dataset shown in Table 4.2. Considering the disruptive operator (see lines 2 to 10, Algorithm 2), the cut-point between the second and third itemsets, and removing any itemset on the left of such cut-point, the result is a partial new solution comprising just the itemset $\{d, e, f\}$. After generating the random itemset $X = \{a, b\}$, the resulting solution o obtained from p is $o = \langle \{a, b\}, \{d, e, f\} \rangle$. As for the less disruptive operators, let us consider the last one that replaces an itemset by another randomly obtained (see lines 19 to 24, Algorithm 2). A random itemset b is chosen from the individual $p = \langle \{b\}, \{c\}, \{d, e, f\} \rangle$, and that itemset is replaced by a new one randomly generated: $\{a, c, h\}$. Due to the item c was already included in p , it is removed from the new itemset and the resulting individual is $o = \langle \{a, h\}, \{c\}, \{d, e, f\} \rangle$.

- 4) **Algorithm.** Finally, it is important to combine all the procedures described above to produce the final algorithm (see Algorithm 3). The proposed $(ES)^2P$ algorithm maintains a fixed size elite with the best individuals (sequences) produced along the evolutionary process, and this set of best solutions is finally returned. The first step is to split the dataset Ω into two datasets, and then, the algorithm creates the population or the initial set of solutions, which are evaluated according to the fitness function (see lines 2 and 3, Algorithm 3). At this point, the elite is also initialized with the best e individuals from the initial population (see line 4), and the number of generations (iterations of the algorithm) is set to 0. An iterative process starts, and it is performed for a number of generations g (see lines 6 to 21, Algorithm 3). In each iteration, the algorithm performs as follows. First, a set of individuals are selected from to act as parents. This selection procedure is carried out by a tournament selector of size 2. The set P is used to apply genetic operators by considering an α probability for the crossover, and a β probability for the mutation. Such genetics operators were already described in Algorithms 1 and 2. Right after the application of the genetic operators, a restoration operator is performed (see line 9) to check that invalid solutions are not formed so individuals can be evaluated again (see line 11). The population is then updated by replacing the previous population by the set of offspring obtained in the current generation. At this point, the best individual is never lost so if it is not in the new population then it is taken from the previous one (see lines 12 to 15, Algorithm 3). The following step carried out by the proposed algorithm is to update the elite set E with the best e unrepeated solutions found along the evolutionary process (see line 16). Last but not least, it is important to highlight that in the event that the algorithm is stuck (i.e., the elite does not improve after m generations), the current population restarts (lines 20 to 24, Algorithm 3). The population is formed by n random

Algorithm 3 Pseudocode for the proposed (ES)²P algorithm

Input Dataset Ω , population size n , elite size e , max number of generations g , max number of generations without improvement m , probability α for crossover and β for mutation

Output \mathcal{E} elite set

```
1:  $\Omega_1, \Omega_2 \leftarrow$  split the dataset  $\Omega$ 
2:  $\mathcal{P} \leftarrow$  generate  $n$  individuals
3: evaluate  $\mathcal{P}$  according to  $\Omega_1$  and  $\Omega_2$ 
4:  $\mathcal{E} \leftarrow$  take the best  $e$  unrepeated individuals from  $\mathcal{P}$ 
5:  $it \leftarrow 0$ 
6: while  $it < g$  do
7:    $b \leftarrow$  gets the best individual from  $\mathcal{P}$ 
8:    $\mathcal{S} \leftarrow$  applies tournament selector to  $\mathcal{P}$ 
9:    $\mathcal{P} \leftarrow$  applies genetic operators on  $\mathcal{S}$ , using  $\alpha$  and  $\beta$ 
10:   $\mathcal{P} \leftarrow$  applies a restoration operator on  $\mathcal{P}$ 
11:  evaluate  $\mathcal{P}$  according to  $\Omega_1$  and  $\Omega_2$ 
12:  if  $b$  is better than  $\text{best}(\mathcal{P})$  then
13:     $\mathcal{P} \leftarrow \mathcal{P} \setminus \{\text{worst}(\mathcal{P})\}$ 
14:     $\mathcal{P} \leftarrow \mathcal{P} \cup \{b\}$ 
15:  end if
16:   $\mathcal{E} \leftarrow$  best  $e$  unrepeated individuals in  $\{\mathcal{E} \cup \mathcal{P}\}$ 
17:  if Average fitness of  $\mathcal{E}$  does not improve after  $m$  generations then
18:     $\mathcal{P} \leftarrow$  generate  $n$  individuals
19:  end if
20:   $it \leftarrow it + 1$ 
21: end while
22: return  $\mathcal{E}$ 
```

individuals, and the crossover and mutation probabilities are also reset to the default values. Finally, once the maximum number of generations is reached, the elite population is returned (see line 27).

3. Resulting Set of Solutions

The previously described, (ES)²P algorithm is really useful to extract not only paths followed by students during the degree but also to extract paths previous to some specific courses. Hence, the idea would be the same, but the input dataset is properly obtained by considering the full paths or the specific sub paths before the specific course. The resulting set E , which is given by the elite of the algorithm for a specific purpose (for example to extract paths that reach to a specific objective course j) is key to perform course recommendations. Giving a course j , the recommendation is based on the analysis of the sequences that describe the paths followed by excellent students in the course j .

The set E of sequences returned by the algorithm is ordered by the fitness value F_s for each $s \in E$. Additionally, each item i within a sequence s has associated a $F_s = F_s \times c_i$, being c_i the credit hours of the course i . Thus, those courses with a higher number of credit hours are more

important than those with lower credit hours. Once a new student t is analysed by the system, his/her complete path s_t is taken into account, and every sequence $s \in E$ is activated for t if the student passed the courses in the same order denoted by s . A recommendation index d_t^j for the student t to take the course j is calculated based on Equation 2. The numerator sums the highest F_s value for items in the activated sequences. Finally, to provide a recommendation index in the $[0, 1]$ range, the denominator sums the maximum F_s value for items in all sequences in the set. Values of d_j closer to 1 means that the student t is prepared to enroll in course j due to its background. Values of d_t^j closer to 0 means that this course should not be taken at this moment. Applying the proposed recommendation index to a given student t on all the courses returns the set of courses that are more appropriate for t .

$$d_t^j = \frac{\sum_{i \in s | s \subseteq s_t} \max(F_i^s | s \subseteq s_t)}{\sum_{i \in s | s \in \mathcal{E}} \max(F_i^s | s \in \mathcal{E})} \quad (2)$$

For a matter of clarification, let us consider the sample set E shown in Table 4.3, which is ordered by the fitness value F_s . Let us also consider three sample students and the correspondence of credit hours for each of the courses. Considering the course j as objective, the first student $t = 1$ activates the sequences #1 and #3 from E , since s_1 belongs to E and is a subset of s_t , and s_3 is also a subset of s_t . Then for each item i in s_1, s_3 , its maximum F_s is obtained, and added to calculate the recommendation index d_t^j . In the case of student $t = 1$, the numerator of d_t^j would be: $0.8 \times 3 + 0.8 \times 2 + 0.8 \times 3 + 0.8 \times 2 + 0.6 \times 2 + 0.6 \times 1 = 9.8$. Note that from s_3 only two courses f and g have incremented the index, since b and d were already present in s_1 with higher F_s value. Additionally, the denominator is calculated as if all the items in E are satisfied. Hence, according to d_t^j (see Equation 2), the recommendation index for student $t = 1$ would be $d_1^j = 9.8/13.4 = 0.731$. On the other hand, the recommendation index value for the other two students ($t = 2$ and $t = 3$) and the course j is $d_2^j = 0.112$, and $d_3^j = 0$. Note that student $t = 3$ does not match any of the sequences in the resulting set so the algorithm does not recommend him/her to enroll in course j at all.

Table 4.3: Sample set E of sequences returned by the algorithm and their F_s values, paths s_t already followed by some sample students, and credit hours c_i of each course

Sequence ID	E	F_s
1	<{a},{b, c},{d }>	0.8
2	<{a,e},{c}>	0.7
3	<{b},{g,d},{f}>	0.6
4	<{k},{l}>	0.5

Student ID	s_t
------------	-------

1	$\langle \{a,l\}, \{b,c,e\}, \{k\}, \{g,d\}, \{f\} \rangle$
2	$\langle \{l\}, \{a,e\}, \{c,d\}, \{f\} \rangle$
3	$\langle \{k,a\}, \{b,c\} \rangle$

Course	c_i	Course	c_i
a	3	f	1
b	2	g	2
c	3	j	2
d	2	k	2
e	3	l	1

4. Experimental Study

This section presents the experimental study, describing first the experimental set up. This section analyses the performance of the proposed algorithm, which is finally compared to exhaustive search algorithms to demonstrate that the proposed evolutionary process makes sense.

4.1. Experimental Setup

All the exhaustive search algorithms used in this comparison are available in the SPMF library [FGCT14]: Spade [Zaki01], Spam [AFGY02], PrefixSpan [PHMW04], CM-Spade [FGCT14] and CM-Spam [FGCT14]. Additionally, the algorithms TKS [10] and TSP [36] are also considered since they do not require any frequency threshold, and their aim is to extract the top-k most frequent itemsets from data. The experiments are carried out on a set of 13 real datasets (see Table 4.4) taken from King Abdulaziz University including information about different faculties: sequences of courses taken by students. Additionally, all the gathered data were divided into 13 groups or datasets, one per faculty. #Records stands for the number of students, Length is the average number of subjects taken by the students to obtain the degree, and finally, #Courses is the number of different courses that each faculty provides. Last but not least, it is important to highlight that each dataset (faculty) is split into two: Ω_1 includes the 30% of students with best GPA in the degree; Ω_2 includes the rest of students. All the experiments are performed on a machine with 6 Intel Xeon E5-2620 CPUs at 2.10 GHz and 64 GB of RAM. The experiments are run ten times, and the average results are considered to reduce environment variations.

Table 4.4: Datasets and their main characteristics

Faculty	#Records	Length	#Courses
Arts	3,892	49.49	375
Business	1,413	46.80	216
Communication & Media	483	48.10	174
Computing & Inf. Tech.	144	51.01	109
Design & Arts	167	54.67	155
Economics & Admin.	2,206	48.31	387
Engineering	365	58.73	364
Engineering Rabeg	102	62.99	217
Home Economics	614	45.29	264
Information Technology	284	54.70	233
Law	946	49.18	183
Sciences & Arts	893	52.26	283
Sciences	1,928	51.87	467

4.2. Analysis of the Proposal

The goal of this first analysis is to demonstrate how well the proposal behaves on multiple datasets and to determine the best values for the hyperparameters, that is, those that provide best fitness values requiring a lower computational time. Table 4.5 shows the average results obtained by different combinations of values for the population size (n) and the population restarting after m generations without improvement. A hypothesis testing by means of non-parametric statistical tests has been conducted with the aim of determining whether there exist significant differences in the overall performance for the combination of values. The Friedman's test [Stat40] has been used to analyze the general differences, whereas the Shaffer's post-hoc test [Shaf86] has been employed to perform all pairwise comparisons. The Friedman's test detected that there were general statistical differences in the ten combinations of values at a significance level of $\alpha = 0.01$, rejecting the null hypothesis with a p -value smaller than $2.2e-16$. Then, the Shaffer's post-hoc test was performed to detect where these significant differences were located. The results for this post-hoc test, at a significance level of $\alpha = 0.01$, are summarized through the critical difference diagram shown in Figure 4.5, illustrating that the values $n=500$ and $m=100$ produce the best values. However, according to the post-hoc test, no statistical difference is found among n values between 500 and 300, and m values between 50 and 100, being the only exception the combination of $n = 300$ and $m = 50$. At this point, it is interesting to analyse the runtime (see Table 4.6) for these six combinations of parameters that present the same performance, with a statistical significance of 99%. The Friedman's test revealed statistical differences in these combination of values at a significance level $\alpha = 0.01$, thus rejecting the null hypothesis with a p -value smaller than $5.074e-10$. Finally, the Shaffer's post-hoc test at a significance level of $\alpha = 0.01$ (see Figure 4.6) revealed no significant differences for $n = 300$ and any m value, as well as $n = 400$ and $m = 100$. According to the results shown in Table 4.6, the

combi- nation of parameters $m = 100$ and $n = 300$ presents the best runtime. Taking all the above into consideration, we recommend the previous combination of parameters.

Table 4.5: Average fitness values obtained by the proposed (ES)²P algorithm considering different population sizes (n) and number of generations without improvement (m) to reset the population. Best results are in bold type-face.

Faculty	n=100		n=200		n=300		n=400		n=500	
	m =50	m =100	m =50	m =100	m =50	m =100	m =50	m =100	m =50	m =100
Arts	0.201	0.202	0.208	0.207	0.210	0.212	0.215	0.214	0.222	0.219
Business	0.150	0.152	0.173	0.184	0.197	0.209	0.207	0.218	0.219	0.226
Communication & Media	0.335	0.339	0.372	0.424	0.409	0.430	0.433	0.464	0.427	0.487
Computing & Informat. Technology	0.374	0.386	0.386	0.386	0.399	0.410	0.408	0.415	0.414	0.419
Design & Arts	0.250	0.259	0.264	0.273	0.273	0.274	0.275	0.282	0.281	0.285
Economics & Administration	0.417	0.419	0.419	0.420	0.419	0.420	0.420	0.421	0.421	0.422
Engineering	0.243	0.247	0.250	0.251	0.251	0.251	0.251	0.251	0.251	0.252
Engineering Rabeg	0.293	0.294	0.297	0.302	0.298	0.301	0.302	0.307	0.305	0.309
Home Economics	0.315	0.323	0.339	0.338	0.348	0.345	0.351	0.350	0.351	0.351
Information Technology	0.367	0.368	0.379	0.378	0.395	0.399	0.398	0.407	0.403	0.407
Law	0.264	0.267	0.268	0.273	0.275	0.278	0.277	0.282	0.280	0.284
Sciences & Arts	0.157	0.156	0.184	0.173	0.189	0.183	0.199	0.191	0.203	0.201
Sciences	0.160	0.162	0.170	0.173	0.173	0.172	0.174	0.176	0.176	0.179

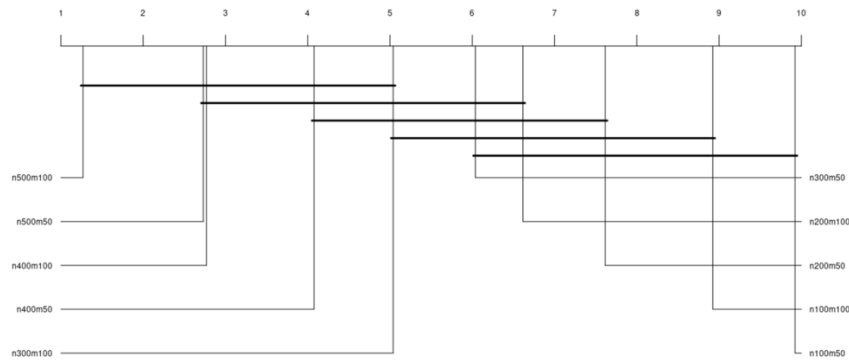


Figure 4.5: Critical difference diagram of the different parameter combinations considered. The comparisons were performed using a Shaffer's test

Let us continue now with the analysis of the convergence of the proposed approach. Figure 4.7 shows how the algorithm behaves on four different datasets: Computing & Information Technology; Design & Art; Economics & Administration; Law. The results on this heterogeneous group of datasets (#Records varies from 144 to 2,206; Length is between 48.3 and 54.67; and #Courses varies from 109 to 387. See Table 4.4) demonstrate that the convergence of the algorithm is high on different scenarios and it is around 1,500 generations for which the algorithm does not widely improve the results. Thus, in order to avoid spending time and computational resources on little fitness improvements, the number of generations is set to 1,500. Last but not least, the crossover and mutation probability values are also fixed to 0.8 and 0.3, respectively. In summary, to obtain the best combination of parameter values, more than 50 parameter configurations were considered, resulting in more than 6,500 executions. Additionally, 10

independent runs were performed for each dataset and parameter configuration to study the algorithm's performance due to its stochastic component.

Table 4.6: Average time in seconds obtained by the proposed (ES)²P algorithm considering different population sizes (n) and number of generations without improvement (m) to reset the population. Best results are in bold type-face.

Faculty	n=300	n=400		n=500	
	<i>m</i> = 100	<i>m</i> = 50	<i>m</i> = 100	<i>m</i> = 50	<i>m</i> = 100
Arts	4.91 × 10 ²	6.55 × 10 ²	6.00 × 10 ²	7.71 × 10 ²	7.41 × 10 ²
Business	3.01 × 10²	3.96 × 10 ²	3.73 × 10 ²	4.44 × 10 ²	4.59 × 10 ²
Communication & Media	1.64 × 10²	2.25 × 10 ²	2.19 × 10 ²	2.76 × 10 ²	2.67 × 10 ²
Computing & Inf. Tech.	1.10 × 10²	1.48 × 10 ²	1.43 × 10 ²	1.86 × 10 ²	1.79 × 10 ²
Design & Arts Economics & Admin.	1.39 × 10²	1.87 × 10 ²	1.81 × 10 ²	2.33 × 10 ²	2.29 × 10 ²
Engineering	3.30 × 10 ²	4.51 × 10 ²	4.32 × 10 ²	5.81 × 10 ²	5.24 × 10 ²
Engineering Rabeg	2.07 × 10²	2.85 × 10 ²	2.72 × 10 ²	3.47 × 10 ²	3.38 × 10 ²
Home Economics	1.45 × 10²	1.97 × 10 ²	1.89 × 10 ²	2.45 × 10 ²	2.38 × 10 ²
Information Technology	1.72 × 10²	2.27 × 10 ²	2.28 × 10 ²	2.82 × 10 ²	2.80 × 10 ²
Law	1.45 × 10²	1.95 × 10 ²	1.91 × 10 ²	2.41 × 10 ²	2.36 × 10 ²
Sciences & Arts	3.15 × 10 ²	4.63 × 10 ²	4.20 × 10 ²	5.54 × 10 ²	5.57 × 10 ²
Sciences	2.04 × 10²	2.74 × 10 ²	2.65 × 10 ²	3.38 × 10 ²	3.28 × 10 ²
	3.01 × 10²	3.99 × 10 ²	4.00 × 10 ²	5.02 × 10 ²	4.85 × 10 ²

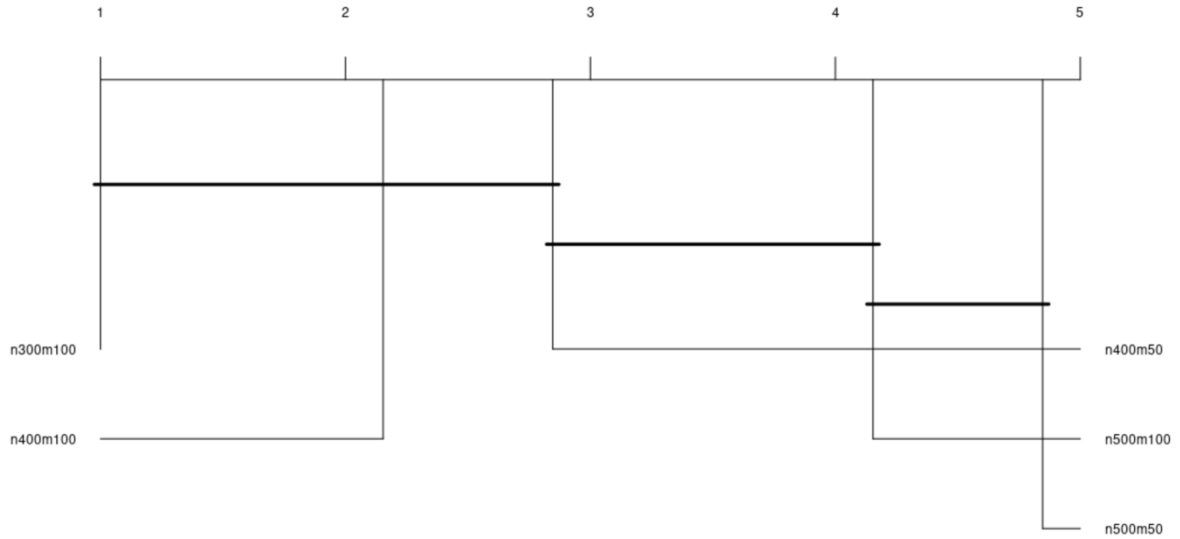
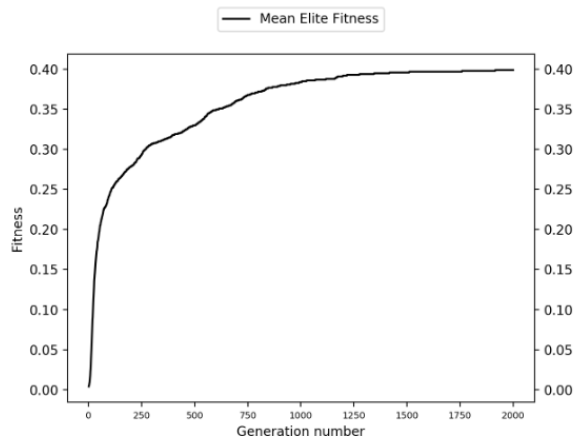
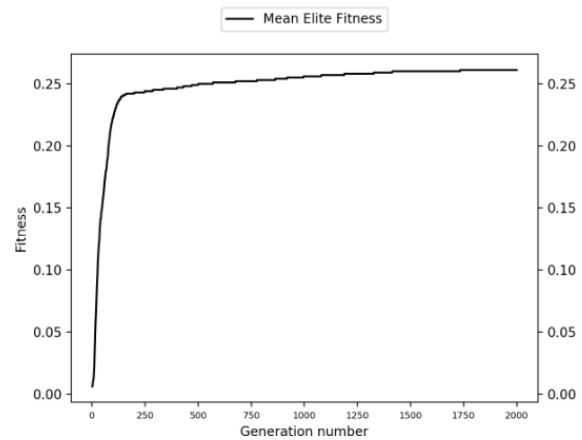


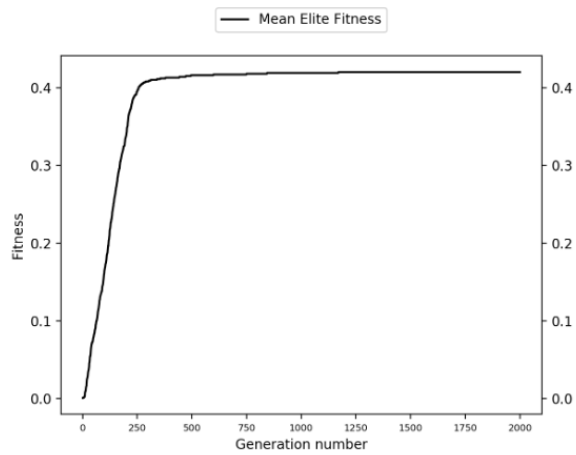
Figure 4.6: Critical difference diagram of the different execution times obtained for the parameter combinations considered. The comparisons were performed using a Shaffer's test



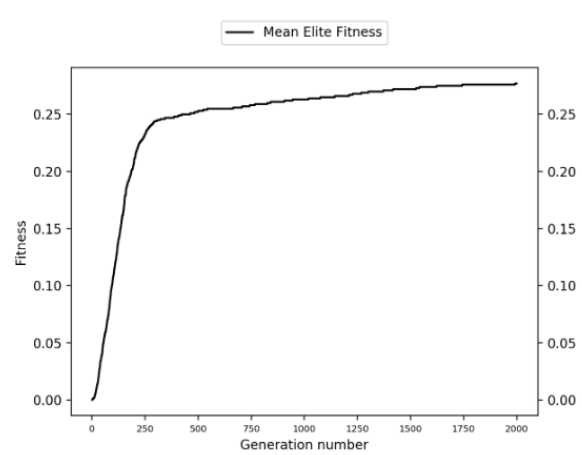
(a) Computing & Information Technology



(b) Design & Art



(c) Economics & Administration



(d) Law

Figure 4.7: Analysis of the convergence of the proposed approach on different datasets

4.3. Comparative Analysis

This second analysis aims to study how well the proposed (ES)²P algorithm behaves when it is compared to existing exhaustive search algorithms. First, we analyse the number of solutions returned by any exhaustive search algorithm (results are the same for any algorithm, as expected) when different frequency thresholds are considered (see Table 4.7). At this point, it is required to highlight that our proposal returns exactly the same for any dataset since it obtains the best 50 solutions found. As it is shown, the number of solutions extracted by exhaustive search algorithms highly vary and it depends on the dataset. This number varies between 59 to 148,461,607 solutions. Second, we analyse the average fitness value obtained by the algorithms on different support threshold values (see Table 4.7). It is important to remark that this is not the average frequency value but the fitness value. Additionally, since the number of solutions is

completely different, we have taken the best 50 solutions to perform a fair comparison with regard to the proposal. As it is expected, exhaustive search algorithms obtained the best results (bold type-face) and, the lower the support threshold value the better results since a wider number of solutions are being analysed. However, for some datasets, our proposal achieved better results in average fitness: Design & Arts; Economics & Administration; Engineering; Home Economics; Information Technology; Law; Sciences & Arts. This interesting behaviour occurs due to fitness function depends on the GR measure, which can be high for small support values. Thus, given a specific support threshold value, extremely good results for the fitness value might be missed. On the contrary, the proposal guides the searching process by the fitness value, achieving better results.

Table 4.7: Number of solutions and average fitness value returned by exhaustive search algorithms (considering different support threshold values) and the proposed (ES)²P approach. *Memory* stands for memory problems when running (out of memory). The average fitness value was calculated by taking the best 50 solutions based on the fitness value.

Faculty	#Solutions				(ES) ² P
	0.5	0.6	0.7	0.8	
Arts	2,642	811	749	583	50
Business	59,336	4,705	1,397	604	50
Communication & Media	184,168	93,667	45,942	14,952	50
Computing & Inf. Tech.	Memory	9,932,208	3,589,091	820,513	50
Design & Arts	4,091	2,413	1,730	751	50
Economics & Admin.	159,850	43,706	8,996	248	50
Engineering	325,611	19,824	10,384	6,764	50
Engineering Rabeg	Memory	8,512,968	1,279,911	180,360	50
Home Economics	Memory	1,661	1,147	790	50
Information Technology	8,027,089	510,522	112,565	38,561	50
Law	484,583	145,763	79,418	43,112	50
Sciences & Arts	1,164	674	65	59	50
Sciences	18,220	1,954	1,397	1,226	50

Faculty	Average Fitness				(ES) ² P
	0.5	0.6	0.7	0.8	
Arts	0.313	0.313	0.310	0.225	0.208
Business	0.303	0.302	0.284	0.202	0.195
Communication & Media	0.659	0.659	0.659	0.640	0.412
Computing & Inf. Tech.	Memory	0.558	0.558	0.558	0.401
Design & Arts	0.295	0.288	0.220	0.158	0.267
Economics & Admin.	0.457	0.457	0.433	0.158	0.420
Engineering	0.266	0.210	0.145	0.126	0.249
Engineering Rabeg	Memory	0.464	0.464	0.464	0.300
Home Economics	Memory	0.437	0.436	0.333	0.340
Information Technology	0.446	0.416	0.323	0.289	0.389

Law	0.326	0.303	0.281	0.261	0.273
Sciences & Arts	0.332	0.323	0.040	0.034	0.172
Sciences	0.309	0.299	0.292	0.290	0.168

Additionally, let us analyse the runtime required by different algorithms on different support threshold values. In this analysis, we consider a set of exhaustive search algorithms that is denoted as the best ones in the specialized literature [LAFA17]: Spade [Zaki01], Spam [AFGY02], PrefixSpan [PHMW04], CM-Spade [FGCT14] and CM-Spam [FGCT14]. Table 4.8 shows the runtime, in seconds, required by the algorithms for a support threshold value of 0.5. At this point, it is important to remind that small differences in the average fitness value were obtained (see Table 4.7): 0.028 in Design & Arts; 0.037 in Economics & Administration; 0.017 in Engineering; 0.057 in Information Technology; 0.053 in Law. Additionally, three datasets cannot be run due to memory problems when using exhaustive search algorithms on such a threshold value (see Tables 4.7 and 4.8). In general terms, our proposal needs lower runtimes and these values do not widely vary from dataset to dataset. Huge differences are found on multiple datasets. For example, in Economics & Administration exhaustive search approaches need more than 2,000 seconds, whereas our proposal only needs 260 seconds. In fact, for this specific dataset the difference in the resulting average fitness value was really low (0.457 in exhaustive search algorithms and 0.420 in our proposal). Among all the results, the maximum differences in runtime are found when the Information Technology dataset is considered, since exhaustive search approaches require more than 16,000 seconds whereas our proposal just 108 seconds. Additionally, for this dataset, the difference in average fitness value was really small (0.446 in exhaustive search algorithms and 0.389 in our proposal). In summary, after analysing all the results for a support threshold value of 0.5, it is possible to assert that the proposed approach is really useful to obtain really good results (according to the average fitness value) in a small quantity of time. Furthermore, this proposal is able to be run on any dataset, whereas exhaustive search approaches fail on some datasets due to memory requirements.

If we continue the analysis for other support threshold values (0.6, 0.7 and 0.8), it is obtained that the higher the threshold value, the lower the runtime required by exhaustive search approaches (see Table 4.8). However, analysing the average fitness value (see Table 4.7), the higher the threshold value, the lower the average fitness value obtained by exhaustive search approaches. In fact, considering a support threshold of 0.8, our proposal obtains better results in seven datasets (see Table 4.7).

Last but not least, it is important to remark that those algorithms that require a minimum support threshold value to be predefined need an extra (previous) process to determine the exact value. This procedure is not trivial, and generally requires a profound background in the application field. Inexpert and many expert users need to try different thresholds by guessing

and re-executing the algorithms once and again until results are good for them [WSTY12]. All of this, together with the large runtimes required on different datasets, and the small differences in the resulting average fitness values, let us to the conclusion that our proposal outperforms exhaustive search algorithms.

Table 4.8: Runtime, in seconds, required by each algorithm on different datasets and considering different support threshold values (0.5, 0.6, 0.7 and 0.8). Our proposal does not require any threshold. *Memory* stands for memory problems when running (out of memory).

Faculty	Support threshold 0.5					
	CM-Spade	CM-Spam	PrefixSpan	Spade	Spam	(ES) ² P
Arts	112	113	111	111	115	372
Business	415	410	446	405	451	208
Communication & Media	619	555	585	587	584	120
Computing & Inf. Tech.	Memory	Memory	Memory	Memory	Memory	82
Design & Arts	10	10	10	10	10	103
Economics & Admin.	2,093	2,201	1,983	2,019	2,299	260
Engineering	723	705	713	728	707	153
Engineering & Rabeg	Memory	Memory	Memory	Memory	Memory	106
Home Economics	Memory	Memory	Memory	Memory	Memory	130
Information Technology	16,025	16,439	16,059	16,614	16,706	108
Law	2,460	2,651	2,774	2,516	2,852	267
Sciences & Arts	23	22	23	22	22	153
Sciences	232	238	220	229	232	233

Faculty	Support threshold 0.6					
	CM-Spade	CM-Spam	PrefixSpan	Spade	Spam	(ES) ² P
Arts	76	74	78	75	72	372
Business	52	56	53	55	52	208
Communication & Media	292	301	306	291	288	120
Computing & Inf. Tech.	12,032	12,075	12,887	12,198	11,578	82
Design & Arts	8	8	8	8	8	103
Economics & Admin.	577	549	561	565	593	260
Engineering	53	51	55	53	51	153
Engineering & Rabeg	8,610	8,055	8,148	8,138	8,209	106
Home Economics	17	17	17	19	18	130
Information Technology	942	947	942	956	892	108
Law	762	806	783	786	791	267
Sciences & Arts	20	20	20	19	19	153
Sciences	52	51	51	52	50	233

Faculty	Support threshold 0.7					
	CM-Spade	CM-Spam	PrefixSpan	Spade	Spam	(ES) ² P
Arts	74	74	75	75	73	372
Business	32	32	32	32	31	208
Communication & Media	156	144	154	160	141	120
Computing & Inf. Tech.	4,270	4,296	4,166	4,528	4,295	82
Design & Arts	7	8	8	7	7	103
Economics & Admin.	142	137	138	142	139	260
Engineering	29	30	29	30	30	153
Engineering & Rabeg	1,175	1,155	1,165	1,163	1,195	106
Home Economics	16	16	16	17	16	130
Information Technology	206	207	208	208	196	108
Law	409	406	421	419	433	267
Sciences & Arts	16	16	17	17	16	153
Sciences	46	46	46	46	45	233

Faculty	Support threshold 0.8					
	CM-Spade	CM-Spam	PrefixSpan	Spade	Spam	(ES) ² P
Arts	72	70	71	70	69	372
Business	26	27	27	26	25	208
Communication & Media	52	56	52	57	54	120
Computing & Inf. Tech.	914	942	887	946	935	82
Design & Arts	7	6	7	6	7	103
Economics & Admin.	37	38	37	38	37	260
Engineering	22	23	22	23	22	153
Engineering & Rabeg	147	148	158	154	152	106
Home Economics	16	15	15	16	17	130

Information Technology	73	72	72	73	74	108
Law	230	242	230	235	231	267
Sciences & Arts	17	16	18	17	17	153
Sciences	44	43	45	44	46	233

4.4. Top-K Sequential Pattern Mining Algorithms

This third analysis aims to study how well the proposed (ES)²P algorithm behaves when it is compared to existing exhaustive search algorithms for mining the top-k solutions. The main advantage of these approaches is that they do not require a previous study to determine a good threshold value. Additionally, they return the same number of solutions regardless the input dataset, which is easier to be managed by experts. However, the main disadvantage of these approaches is related to the fitness values. Existing algorithms for mining top-k sequential patterns were proposed for mining the best results in terms of frequency (support values). Nevertheless, for the problem at hand, the support value cannot establish the importance of the sequence. A sequence can be frequent for both excellent and not so good students and, therefore, the GR value is low. Additionally, a sequence can be infrequent for excellent students and zero for not so good students, providing an excellent GR value. This theoretical behaviour is tested by running two algorithms that determine the state-of-the-art, that is, TKS [FGGM13] and TSP [TzYH03], on different datasets. The results (see Table 4.9) demonstrate that extremely bad results are obtained by these algorithms. The runtime needed by TKS and TSP is much lower than (ES)²P, but the results are useless for the problem at hand (fitness values close to 0).

Table 4.9: Runtime and average fitness value returned by top-k search algorithms and the proposed approach considering different k values (25, 50, 100, 200).

Faculty	k = 25						k = 50					
	Average Fitness			Runtime			Average Fitness			Runtime		
	TKS	TSP	(ES) ² P	TKS	TSP	(ES) ² P	TKS	TSP	(ES) ² P	TKS	TSP	(ES) ² P
Arts	0.004	0.004	0.208	62	62	372	0.004	0.004	0.208	63	62	372
Business	0.015	0.015	0.195	25	22	208	0.025	0.025	0.195	23	23	208
Communication & Technology	0.034	0.249	0.412	11	11	120	0.031	0.242	0.412	11	11	120
Computing & Inf. Tech.	0.250	0.460	0.401	7	10	82	0.235	0.443	0.401	8	10	82
Designs & Arts	0.022	0.022	0.267	6	8	103	0.022	0.022	0.267	5	5	103
Economics	0.001	0.001	0.420	34	34	260	0.002	0.002	0.420	36	34	260
Engineering	0.043	0.049	0.249	11	11	153	0.031	0.035	0.249	12	12	153
Engineering Rabeg	0.145	0.159	0.300	5	5	106	0.127	0.154	0.300	6	5	106
Home Economics	0.003	0.003	0.340	13	12	130	0.003	0.003	0.340	13	13	130
Information Technology	0.148	0.156	0.389	9	10	108	0.147	0.155	0.389	9	10	108
Law	0.031	0.036	0.273	21	28	267	0.028	0.034	0.273	21	27	267
Sciences	0.011	0.011	0.172	17	16	153	0.025	0.025	0.172	17	16	153
Sciences & Arts	0.002	0.005	0.168	33	33	233	0.002	0.003	0.168	34	34	233

Faculty	k = 100						k = 200					
	Average Fitness			Runtime			Average Fitness			Runtime		
	TKS	TSP	(ES) ² P	TKS	TSP	(ES) ² P	TKS	TSP	(ES) ² P	TKS	TSP	(ES) ² P
Arts	0.018	0.018	0.208	63	62	372	0.065	0.077	0.208	64	65	372
Business	0.041	0.041	0.195	23	23	208	0.052	0.052	0.195	24	24	208
Communication & Technology	0.023	0.224	0.412	11	11	120	0.048	0.193	0.412	11	12	120
Computing & Inf. Tech.	0.227	0.418	0.401	7	10	82	0.192	0.386	0.401	8	10	82
Designs & Arts	0.016	0.016	0.267	6	6	103	0.031	0.066	0.267	7	6	103
Economics	0.002	0.002	0.420	35	34	260	0.068	0.067	0.420	36	37	260
Engineering	0.021	0.024	0.249	11	11	153	0.014	0.017	0.249	11	11	153
Engineering Rabeg	0.108	0.139	0.300	5	6	106	0.086	0.120	0.300	5	5	106
Home Economics	0.023	0.023	0.340	14	12	130	0.036	0.033	0.340	13	13	130
Information Technology	0.118	0.152	0.389	9	10	108	0.076	0.124	0.389	9	10	108
Law	0.024	0.031	0.273	21	28	267	0.019	0.027	0.273	21	27	267

Sciences	0.054	0.054	0.172	17	17	153	0.069	0.069	0.172	17	17	153
Sciences & Arts	0.002	0.003	0.168	33	33	233	0.004	0.005	0.168	33	34	233

5. Study Case

In this section, we propose two different methodologies to apply the proposed (ES)²P algorithm for course recommendation. First, we propose a methodology for ordering courses that should be taken by students to success in the degree. This methodology, based on the proposed (ES)²P algorithm, provides full study plans that should be followed by students to reduce the dropout and failure. Second, we propose a methodology for rating courses with the aim of providing the students with advises on which courses should be taken at any specific moment of their degree. The aim is to recommend subjects that best fits to them according to their paths (previous courses). This methodology requires nothing more than the courses (ordered by semesters) already passes by the student that is being advised. Last but not least, it is important to remark that courses are represented by IDs in these cases of study to simplify the results. We consider a real dataset including more than 13,000 students belonging to 13 different faculties from King Abdulaziz University, Saudi Arabia.

5.1. Study Plan Recommendation Based on the Best Ordering of Courses

When no course is given, the algorithm extracts discriminative sequential patterns on complete paths carried out by students. As a study case, we have considered two different faculties: Business and Information Technology (see Table 4.4). As a matter of simplification, we have taken only the top 5 solutions returned by the proposed methodology.

Let us start with the Faculty of Business study case. Table 4.10 shows the 5 best solutions found according to the fitness value. The support on the set of good students and the GR value is also available. The path with the best fitness value denotes that more than 70% of the excellent students have passed course 30 in a semester and courses 11 and 43 together in a subsequent semester. This path is satisfied 1.69 times more often in excellent students than in not so good students. A similar behaviour is denoted by the second ($\langle\{44\}, \{11, 43\}\rangle$) and the third paths ($\langle\{11, 43\}\rangle$). As a result, it is possible to assert that to take subjects with IDs 11 and 43 in the same semester is a synonymous of being an excellent student in the Faculty of Business. Nevertheless, it is fair to say that no excellent result was obtained in terms of courses that heavily denote a difference between excellent and not so good students. It is mainly due to, for this Faculty, there is not good paths to be performed by students and, generally, all the students equally behave.

Even more interesting are the results obtained on the Faculty of Information Technology (see Table 4.11). Analyzing the top 5 solutions, we obtain that the courses with id 35 and 47 appear in any of the paths and, in fact, they are studied in the same semester. In any of the cases, all the returned paths present a behavior that is three times more often for excellent students than for not so good students. For example, focusing on the solution $\langle\{50\}, \{39\}, \{35, 47\}\rangle$, it determines that if a student pass the course with ID 50 in a semester, then in a different semester, such a student pass the course with ID 39, and then, in a different semester, he/she pass courses with IDs 35 and 47 (in the same semester this time), such a student has 3.8 times more probability to be an excellent student and the end of the degree. Hence, this information is really useful to provide study plans and to analyze why such differences among students when they take such courses in that order.

Table 4.10: Top 5 complete paths returned by the proposal on Faculty of Business.

Paths	Fitness	Support	GR
$\langle\{30\}, \{11, 43\}\rangle$	0.295	0.719	1.694
$\langle\{44\}, \{11, 43\}\rangle$	0.294	0.724	1.685
$\langle\{11, 43\}\rangle$	0.286	0.724	1.654
$\langle\{10, 7\}\rangle$	0.280	0.639	1.781
$\langle\{30\}, \{10\}, \{46\}\rangle$	0.246	0.802	1.442

Table 4.11: Top 5 complete paths returned by the proposal on Faculty of Information Technology.

Paths	Fitness	Support	GR
$\langle\{38\}, \{35, 47\}\rangle$	0.409	0.565	3.625
$\langle\{6\}, \{35, 47\}\rangle$	0.409	0.565	3.625
$\langle\{50\}, \{39\}, \{35, 47\}\rangle$	0.401	0.541	3.846
$\langle\{19\}, \{35, 47\}\rangle$	0.399	0.565	3.405
$\langle\{51\}, \{35, 47\}\rangle$	0.399	0.565	3.405

5.2. Course Recommendation Based on the Previous Academic Path

This second study case is related to the recommendation of which courses should be taken by a student in a specific semester according to its path (courses already taken by him/her). The aim is to improve his/her academic success. In this study case, we have considered the same faculties discussed in the previous study case: Business and Information Technology. We have also taken four different students to provide them a recommendation (two of each faculty).

Table 12: Top 5 courses recommended to the student $t_1 = \langle\{12, 22, 30, 44, 45, 75\}, \{1, 9, 14, 15, 17, 33\}, \{3, 11, 34\}\rangle$ belonging to the Faculty of Business.

Course ID	Course name	$d_{t_1}^i$
49	ACC415	1

88	BLA322	1
166	ACC321	1
169	PE120	0.950
96	MRK303	0.885

Let us start with a student t_1 from the Faculty of Business, having the path $t_1 = \langle \{12, 22, 30, 44, 45, 75\}, \{1, 9, 14, 15, 17, 33\}, \{3, 11, 34\} \rangle$. Thus, t_1 has passed 6 subjects in a semester, 6 subjects in a posterior semester, and 3 subjects in a subsequent semester. For this student, our proposal recommends to take the courses shown in Table 4.12 right now. The recommendation index score of the recommended courses is really good (close to the maximum of 1), meaning that all (or almost all) students with the same path have obtained excellent marks in such courses. Analysing what this student really did, we check that one of the recommended courses was taken (course with ID 88). In this course, the student t_1 obtained a GPA that is within the 5.4% of the best GPA obtained for that course among all the students. Thus, it is demonstrated that when a student follows the recommendations, he/she obtains really good GPAs.

Table 4.13: Top 5 courses recommended to the student $t_2 = \langle \{12, 22, 30, 44, 45, 75\}, \{1, 9, 14, 15, 17, 33\}, \{3, 21\}, \{2, 11, 26, 43\}, \{10, 24, 36, 74, 76\}, \{20, 46, 63, 77, 142\}, \{7, 8, 68, 69, 71\}, \{34\} \rangle$ belonging to the Faculty of Business.

Course ID	Course name	$d_{t_2}^j$
193	COM205	0.841
59	ACC411	0.834
121	MRK322	0.832
169	PE120	0.806
96	MRK303	0.799

Let us now consider a second student t_2 from the same faculty and who has followed the path $t_2 = \langle \{12, 22, 30, 44, 45, 75\}, \{1, 9, 14, 15, 17, 33\}, \{3, 21\}, \{2, 11, 26, 43\}, \{10, 24, 36, 74, 76\}, \{20, 46, 63, 77, 142\}, \{7, 8, 68, 69, 71\}, \{34\} \rangle$. This student is close to finish his/her degree since he/she has completed 8 semesters and 34 different subjects. The top 5 courses recommended by the algorithm and the recommendation index scores are summarized in Table 4.13. This time, the student did not take any of the recommended courses and took some courses that were not appropriate at all for him/her. For example, analysing the path finally followed by such a student, he/she took the courses with ID 72 and 88. Such courses presents a recommendation index score of 0.069 and 0.000, respectively. Thus, such courses were not appropriate for the student t_2 as it is finally proved by the GPA obtained for such courses. In course with ID 72, the student obtained a GPA of 87, which is within the 34.89% of the students (ranked by GPA for that course). Similarly, the student obtained a GPA of 85 in the course 88, which is within the 55.94% of the ranking of students. As it is demonstrated, to follow the recommendation is crucial to obtain good GPAs.

The following analysis is carried out on a different Faculty, that is, Information Technology. For this Faculty, we take a student t_3 that has followed the path $t_3 = \{\{5, 8, 9, 18, 25, 45\}, \{4, 12, 13, 23, 28\}, \{1, 10, 15, 21, 51\}, \{3, 6, 14, 27, 74\}, \{26, 29, 75, 78, 84\}\}$. Table 4.14 summarizes the top 5 courses recommended to this student by the proposed methodology together with the recommendation index scores for each course. In this occasion, the student t_3 finally took two of the five best courses recommended to him/her, that is, courses with IDs 64 and 77. To show the adequacy of the proposal and the validity of the recommendations proposed, let us analyse the GPA obtained by t_3 on those courses. t_3 obtained a GPA of 96 in the course with ID 64, being among the 11.11% best students for that course. As for the course with ID 77, he/she obtained a GPA of 95, which corresponds to the top 13.58% of the best students for that course.

Finally, let us consider a student t_4 belonging to the Faculty of Information Technology, which has passed the courses identified by the following path $t_4 = \{\{5, 12, 23, 28, 45\}, \{4, 8, 9, 13, 18, 25\}, \{1, 10, 15, 21, 51\}, \{3, 6, 14, 27, 74\}\}$. Analysing t_4 , he/she is recommended to take 5 courses as the top according to the recommendation index score (see Table 4.15). Analysing what the student finally did, it is obtained that he/she finally took two of such five courses that were recommended. In this way, for the course with ID 26, t_4 obtained a GPA of 85 (top 37.32% in the total GPA ranking for that course). On the other hand, the GPA obtained by t_4 on the course with ID 77 was again 85, being this time in the top 27.16% of the ranking for the given course. However, if we consider the rest of the courses taken by this student, they are present with a very low recommendation index score. For example, in the case of the course with ID 78, the recommendation index score is 0.365 (the student t_4 finally obtained a GPA of 68 in that course, which is within the 80.25% of the best GPAs of that course); whereas for the course with ID 84, the recommendation index value was 0. The student finally took this course and his/her GPA was 87 for that course and this GPA is within the 64.36% of best students. As it is demonstrated, the recommendation index score is low because taking such courses implies not to be in the group of excellent students. Finally, it is important to clarify that the number of credits of each course is taken into account to obtain the recommendation index score, what explains why lower score values may imply the student to be in a higher position of the ranking.

Table 4.14: Top 5 courses recommended to the student $t_3 = \{\{5, 8, 9, 18, 25, 45\}, \{4, 12, 13, 23, 28\}, \{1, 10, 15, 21, 51\}, \{3, 6, 14, 27, 74\}, \{26, 29, 75, 78, 84\}\}$ belonging to the Faculty of Information Technology.

Course ID	Course name	$d_{t_3}^i$
64	CPCS223	1
19	CPIT210	0.957
77	CPCS211	0.946
88	ISLS211	0.920

105	CPIS210	0.883
-----	---------	-------

Table 4.15: Top 5 courses recommended to the student $t_4 = \langle \{5, 12, 23, 28, 45\}, \{4, 8, 9, 13, 18, 25\}, \{1, 10, 15, 21, 51\}, \{3, 6, 14, 27, 74\} \rangle$ belonging to the Faculty of Information Technology.

Course ID	Course name	$d_{t_4}^i$
26	CPCS204	0.940
88	ISLS211	0.920
29	ARAB201	0.901
77	CPCS211	0.874
20	CPIT220	0.851

CHAPTER V. DMDIM: A Method to Calculate a Course Difficulty Index. Validation on KAU Student Data

In this chapter, we introduce the Data Mining Difficulty Index Method (DMDIM) as a technique approach to calculate the course difficulty index for improving the student efficiency and performance and recommending each student a proper course plan. According to the student's performance in the semesters, we find that the student needs to choose a group of study courses to complete the department's study plan and achieve the highest General Point Average (GPA) with high grades, and thus he is selected courses based on five traditional methods,

- 1- Ask the department's academic advisor about the courses that can be selected for this semester to improve the GPA.
- 2- Asking for information of past students about the courses they studied, and which one would be the best to choose as a set of courses taught in a single semester.
- 3- Choose the courses based on the names of the registered instructors who will be teaching them this semester.
- 4- Choosing courses with friends from the same batch, where the study is done as a group to support one another.
- 5- Acceptance of the scientific department's recommended study plan.

The student's traditional methods to choose a group of study courses are shown in Figure 5.1.

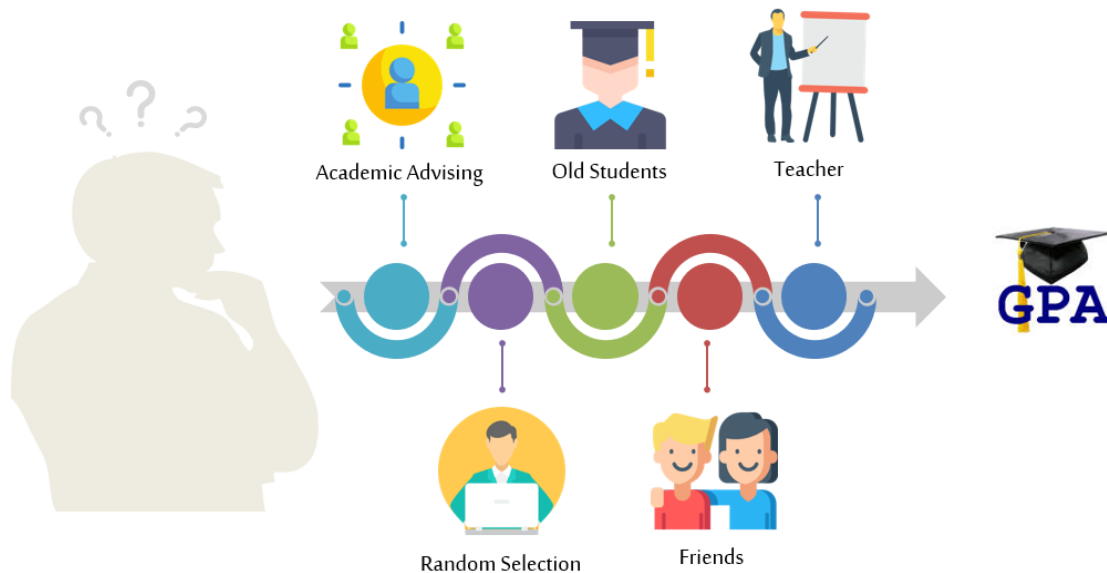


Figure 5.1: The student's traditional methods to choose a group of study courses

We note that the student used these methods to determine the degree of difficulty / simplicity of the courses he will be studying. However, none of the previous methods are based on a clear scientific approach to assist him in selecting the group of study courses that he must register in one semester to achieve the best academic results.

1. Proposed Difficulty Index

This study provides a data-mining model that extracts the major factors from the student's historical data and improves the student efficiency and performance by measuring the course difficulty index to commend each student a proper course plan.

The researcher select KAU as a study case as mentioned in chapter one and some tasks were done to calculate the difficulty index.

- 1- Collecting student and teacher biographical and academic data, course elements data, and student grades for 13 colleges, 69 study programs, 5223 teachers, and 13472 students.
- 2- Collecting the factors likely to affect the course difficulty index from the real data obtained.
- 3- Constructing for all factor's classifications.
- 4- Finding a Relative Degree Points using the cumulative average method.
- 5- Factor selection follows the finding for their mathematical variance.

DMDIM factors as shown in Table 5.1. These all factors are recognized with respect their ability to have the impact on student's grades.

Factors Related Student													
STD_AGE_BY_YEAR	18	19	20	21									
STD_GENDER_CODE	F	M											
STD_HIGH_SCHOOL_GPA	PASS	GOOD	VERYGOOD	EXCELLENT									
STD_HOME_CITY_CODE	IN_REGION	OUT_REGION											
STD_NATIONALITY_CODE	CITIZEN	NOTCITIZEN											
STD_STUDY_PERIOD_SEMESTER	7	8	9	10	11	12	13	14	15	16	17	18	19
STD_STUDY_PERIOD_YEAR	3	4	5	6									
STD_SUMMER_SEMESTERS	0	1	2	3	4	5	6						
STD_UNIVERSITY_GPA	PASS	GOOD	VERYGOOD	EXCELLENT									
Factors Related Course													
CRS Course Type	B	C	L										
CRS_STUDENT_COUNTER	30	60	100	150									
CRS_TIME_EARLYMORNING_WEIGHT	0	1	2	3	4	5	6	7					
CRS_TIMEAFTERNOON_WEIGHT	0	1	2	3	4	5	6	7	8	9	10	11	12
CRS_TIMEEVENING_WEIGHT	0	1	2	3	4	5	6	7	9	10	11		
CRS_TIMEMORNING_WEIGHT	0	1	2	3	4	5	6	7	8	9	10	11	13
CRS_TIMENIGHT_WEIGHT	0	1	2	3	4								
Factors Related Teacher													
TCHR_AGE_TEACH_COURSE	0	30	40	50	60	GT							
TCHR_CERT_COUNTRY_CODE	ABROAD	LOCAL											
TCHR_EXP_TEACH_COURSE	0	10	30	GT									
TCHR_EXP_TEACH_MAJOR	5	10	15	GT									
TCHR_GENDER	F	M											
TCHR_GRADE_CODE	LECTR	ASSIS	ASOC	PROF	NACDM								
TCHR_NATIONALITY_CODE	CITIZEN	NOTCITIZEN											

Table 5.1: Classification of resulting values of factors

2. DMDIM Evaluation

First, let us present some important concepts and definitions:

- There are n factors F_i where $i = \{1, 2, \dots, n\}$
 $\forall F_i$ We have c_i classifications such that $F_{i(r)}$ where $r = \{1, 2, \dots, c_i\}$
 Suppose: The sixth factor is *STD_UNIVERSITY_GPA* hence; $F_6 = \text{STD_UNIVERSITY_GPA}$
 and it has 4 classifications, such that: $F_{6(1)} = \text{PASS}$, $F_{6(2)} = \text{GOOD}$, $F_{6(3)} = \text{VERYGOOD}$
 $F_{6(4)} = \text{EXCELLENT}$ Here, $i = 6$ and $c_6 = 4$
- We get the Cartesian product that is the classification's factor component's matrix to obtain the relationship between two factors that is:

$$F_x \times F_y = \begin{bmatrix} F_{x(1)}F_{y(1)} & F_{x(1)}F_{y(2)} & \dots & F_{x(1)}F_{y(c_y)} \\ F_{x(2)}F_{y(1)} & F_{x(2)}F_{y(2)} & \dots & F_{x(2)}F_{y(c_y)} \\ \vdots & \vdots & \vdots & \vdots \\ F_{x(c_x)}F_{y(1)} & F_{x(c_x)}F_{y(2)} & \dots & F_{x(c_x)}F_{y(c_y)} \end{bmatrix}$$

That is employing the $c_x \times c_y$ components. Let $c_x = 2$ and $c_y = 3$, so; we have 6 components.

Table 5.2 present some examples for Component, classification, and their corresponding weights as follows:

Table 5.2: Component, classification, and their corresponding weights

Component	Classification	Weight
$C_{(1,1)}$	$F_{x(1)}F_{y(1)}$	$w_{(1,1)}$
$C_{(1,2)}$	$F_{x(1)}F_{y(2)}$	$w_{(1,2)}$
$C_{(1,3)}$	$F_{x(1)}F_{y(3)}$	$w_{(1,3)}$
$C_{(2,1)}$	$F_{x(2)}F_{y(1)}$	$w_{(2,1)}$
$C_{(2,2)}$	$F_{x(2)}F_{y(2)}$	$w_{(2,2)}$
$C_{(2,3)}$	$F_{x(2)}F_{y(3)}$	$w_{(2,3)}$

For example: If we have two factors namely STD_GENDER_CODE and STD_UNIVERSITY_GPA. The first factor has 2 classifications {"MALE","FEMALE"} and the second one has 4 classifications {"PASS", "GOOD", "VERYGOOD", "EXCELLENT"}

So; that component's matrix is:

$$\begin{bmatrix} \text{MALE, PASS} & \text{MALE, GOOD} & \text{MALE, VERY GOOD} & \text{MALE, EXCELLENT} \\ \text{FEMALE, PASS} & \text{FEMALE, GOOD} & \text{FEMALE, VERY GOOD} & \text{FEMALE, EXCELLENT} \end{bmatrix}$$

So; if we have 3 Factors, the weights are employed the $c_x \times c_y \times c_z$ weights.

1st weight for the 1st component $C_{(1,1,1)} = F_{x(1)}F_{y(1)}F_{z(1)}$ is $w_{(1,1,1)}$

2nd weight for the 2nd component $C_{(1,1,2)} = F_{x(1)}F_{y(1)}F_{z(2)}$ is $w_{(1,1,2)}$

And so on.

For F_1, F_2, \dots, F_n , we have $c_1 \times c_2 \times \dots \times c_n$ weights.

2.1. Course Difficulty Index Calculation (CDIC)

Course Difficulty Index Calculation (CDIC) passes through two steps. Depending on the benefit derived from them.

1. To identify the best factors the general weight for factor (GWF), will be utilized to measure the factor that affects the student's registration of the course with credit hours.
2. To calculate course weight, we will use the general weight for course (GWC) algorithm. During the weight calculating process, many specific items are considered. The following are some of them:

- Number of students (\mathcal{A}) that contains GWF that is referred to as all students in the university and GWC that is meant to be as particular course that all students studied.
- The set of grades (G) in dataset
where $G = \{A+, A, B+, B, C+, C, D+, D, F, DN, NF, NP, NA\}$.
- Number of students in each grade level (s_g).

$$KAU_{GPA} = \frac{\sum_{j=1}^r chc(j) * GPAP(chc)}{\sum_{j=1}^r chc(j)}, \quad (5.1)$$

where, $chc(j)$ represents the course credit hours and $GPAP(chc)$ GPA Grade Point for course chc .

- The Relative Degree Points (RDP), which is used in the DMDIM, was derived from the way of calculating the General Cumulative Average (GPA) of KAU students, which is based on calculating a number of points in consideration of the grade that was taken by the student. In the table that has been given below that complementary has been gotten from 6 as compared to 5 to avoid the RDP "0" for $A +$ grade as shown in Table 5.3.

Grade Name	Marks	Symbol	GPA Points (KAU)	RDP (6 – GPA Points)
High Excellent	95 – 100	A+	5.00	1.00
Excellent	90 – 94	A	4.75	1.25
High Very Good	85 – 89	B+	4.50	1.50
Very Good	80 – 84	B	4.00	2.00
High Good	75 – 79	C+	3.50	2.50
Good	70 – 74	C	3.00	3.00
High Pass	65 – 69	D+	2.50	3.50
Pass	60 – 64	D	2.00	4.00
Fail	Less than 60	F	1.00	5.00
Denial		DN	1.00	5.00
No Grade Fail		NF	1.00	5.00
No Grade Pass		NP	2.00	4.00
Not Applicable		NA	5.00	1.00

Table 5.3: RDP in calculating the GPA

2.2. General Weight for Factor (GWF)

General Weight for Factor Calculation is the process to rank the factors and determine the best factors. It will be calculated for all of the courses given in the dataset. In this step we'll calculate the General Weight for Factor (GWF) for any of the factor F_i having classifications c_i . Through this step, the benefit of factors on each other will also be examined. The calculation is explained below:

$$GWF = \left[\sum_{r=1}^{c_i} w_r^* \right] / c_i , \quad (5.2)$$

where,

$$w_r^* = \sum_{g=1}^{n(G)} \frac{(s_g)_r}{\mathcal{A}_r} \times (RDP)_g \text{ as a component weight,}$$

$n(G)$ represents the number of elements in the grade set G ,

\mathcal{A}_r represents the number of students in r classification, r

$(s_g)_r$ represents the number of students has grade g and in classification r ,

$(RDP)_g$ represents grade g related Relative Difficulty Points,

The arithmetic mean of all of the component's weights have been gotten to calculate each factor's (GWF) total weight (w^*), for the selection of factor (Factor screen control). The standard

deviation is a measurement of a set of values' variation or dispersion. A low standard deviation implies that the values are close to the set's mean, whereas a high standard deviation suggests that the values are dispersed over a larger range. By calculating the variance(s^2) for each factor, difference between factors has been depicted and it can also be used to select the number of factors that directly influence the DI.

$$s^2 = \left[\sum_{r=1}^{c_i} (w_r^* - GWF)^2 \right] / c_i \quad (5.3)$$

The Algorithm of GWFCA and its flowchart (Figure 5.2) are stated as follows:

General Weight Factor Calculation Algorithm (GWFCA)

```

F ← Get All Factors in Dataset
G ← Get All Gardes in Dataset
P ← Get All RDP in Dataset
Define C as empty Array
Define All_Factors_W as empty Array
Define Class_Comp_S , Comp_S as integer
Define Factor_W, Comp_W, Div1 as double
for each itemset  $F_i \in F$  do
    Factor_W ← 0
    Comp_W ← 0
    C ← GetCassifications( $F_i$ )
    for each itemset  $c_j \in C$  do
        Comp_S ← GetStudentNo( $c_j$ )
        for each itemset  $g_t \in G$  do
            Class_Comp_S ← GetStudentNo( $c_j, g_t$ )
            Div1 =  $\frac{Class\_Comp\_S}{Comp\_S} \times P(g_t)$ 
            Comp_W ← Comp_W + Div1
        end for
    end for
    Factor_W ←  $\frac{Comp\_W}{n(C)}$ 
    All_Factors_W[i] = Factor_W
end for

```

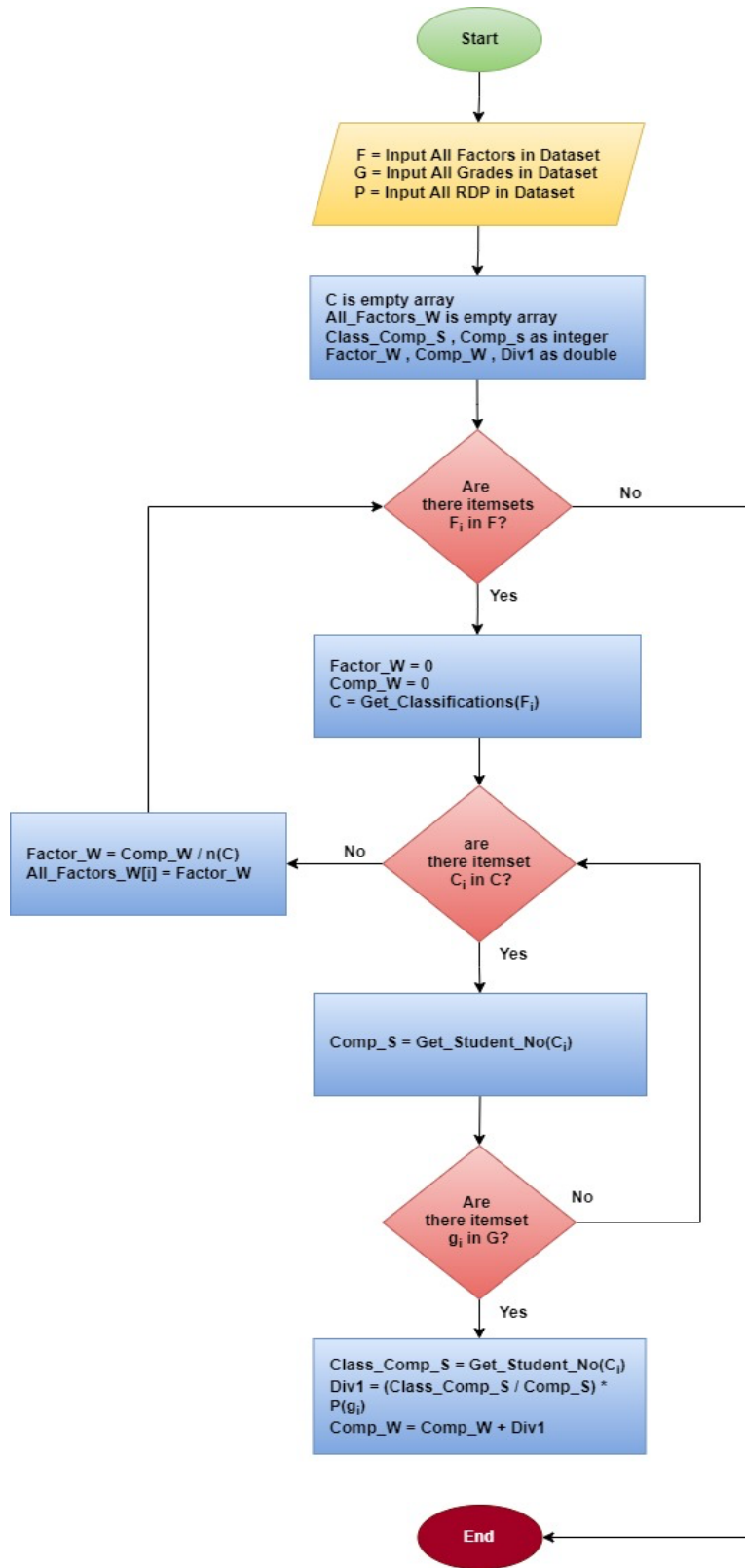


Figure 5.2: Flowchart of GWFCA

2.3. General Weight for Course (GWC)

General Weight for Course (GWC) is measured for specific course, and it affects the student's registration of the course with credit hours. In this step, it will be calculated for all selected factors in the step of GWF. Let's assume that selected factors are n numbers like F_1, F_2, \dots, F_n and each factor has its classification c_i and in the result of overlapping of chosen factor's classification, number of components is generated. Each component has one classification of each factor as

$$C_{(x,y,\dots,z)}$$

where,

C represents Component.

c_i represents the factor number i 's number of classification.

x is referred as any classification number in F_1 so; $x \in \{1,2, \dots c_1\}$,

y is referred as any classification number in F_2 so; $y \in \{1,2, \dots c_2\}$

and

z is referred as any classification number in F_n so; $z \in \{1,2, \dots c_n\}$

Hence, number of factors are equal to the number of classifications in one component and the maximum number of components are $c_1 \times c_2 \times \dots \times c_n$. When for few of the courses, some classifications don't exist in some factors then the perspective component's weight will be 0. To obtain GWC (course D.I) for any course, following equation will be used.

$$GWC = \left[\sum_{\alpha=1}^{c_1} \sum_{\beta=1}^{c_2} \dots \sum_{\gamma=1}^{c_n} w_{(\alpha,\beta,\dots,\gamma)} \right] / [c_1 \times c_2 \times \dots \times c_n] \quad (5.4)$$

where,

$$w_{(\alpha,\beta,\dots,\gamma)} = \sum_{g=1}^{n(G)} \frac{(S_g)_{(\alpha,\beta,\dots,\gamma)}}{\mathcal{A}_{(\alpha,\beta,\dots,\gamma)}} \times (RDP)_g \text{ as a component weight,}$$

$\mathcal{A}_{(\alpha,\beta,\dots,\gamma)}$ represents the number of course's student in component $(\alpha, \beta, \dots, \gamma)$,

$\alpha, \beta, \dots, \gamma$ represents the few classifications in F_1, F_2, \dots, F_n respectively,

$(S_g)_{(\alpha,\beta,\dots,\gamma)}$ represents the number of course's students with component $(\alpha, \beta, \dots, \gamma)$ and grade g .

The Algorithm of GWCCA and its flowchart (Figure 5.3) are stated as follows: -

General Weight for Course Calculation Algorithm (GWCCA)

```
Fr ← Get Ranked Factors in Dataset
CN ← Get Course Name
G ← Get All Gardes in Dataset Related CN
P ← Get All RDP in Dataset
Define MFr = GetCount(Fr)
Define GWC, ALL_Comp_W, Comp_W, Div1 as double
Define CN, x, y, ..., z as String
Define num1, num2, ..., numMFr as integer
ALL_Comp_W = 0
for num1 = 1 to GetClassificationsCount(Fr1) do
  x ← GetClassification(Fr(1,num1))
  for num2 = 1 to GetClassificationsCount(Fr2) do
    y ← GetClassification(Fr(2,num2))
    ...
    for numMFr = 1 to GetClassificationsCount(FrMFr) do
      z ← GetClassification(Fr(FrMFr,numMFr))
      Comp_W ← 0
      Comp_S ← GetStudentNo(CN, x, y, ..., z)
      for each itemset gt ∈ G do
        Class_Comp_S ← GetStudentNo(CN, x, y, ..., z, gt)
         $Div1 = \frac{Class\_Comp\_S}{Comp\_S} \times P(g_t)$ 
        Comp_W ← Comp_W + Div1
      end for
      ALL_Comp_W ← ALL_Comp_W + Comp_W
    end for
  end for
end for
GWC = ALL_Comp_W / [num1 × num2 × ... × numMFr]
```

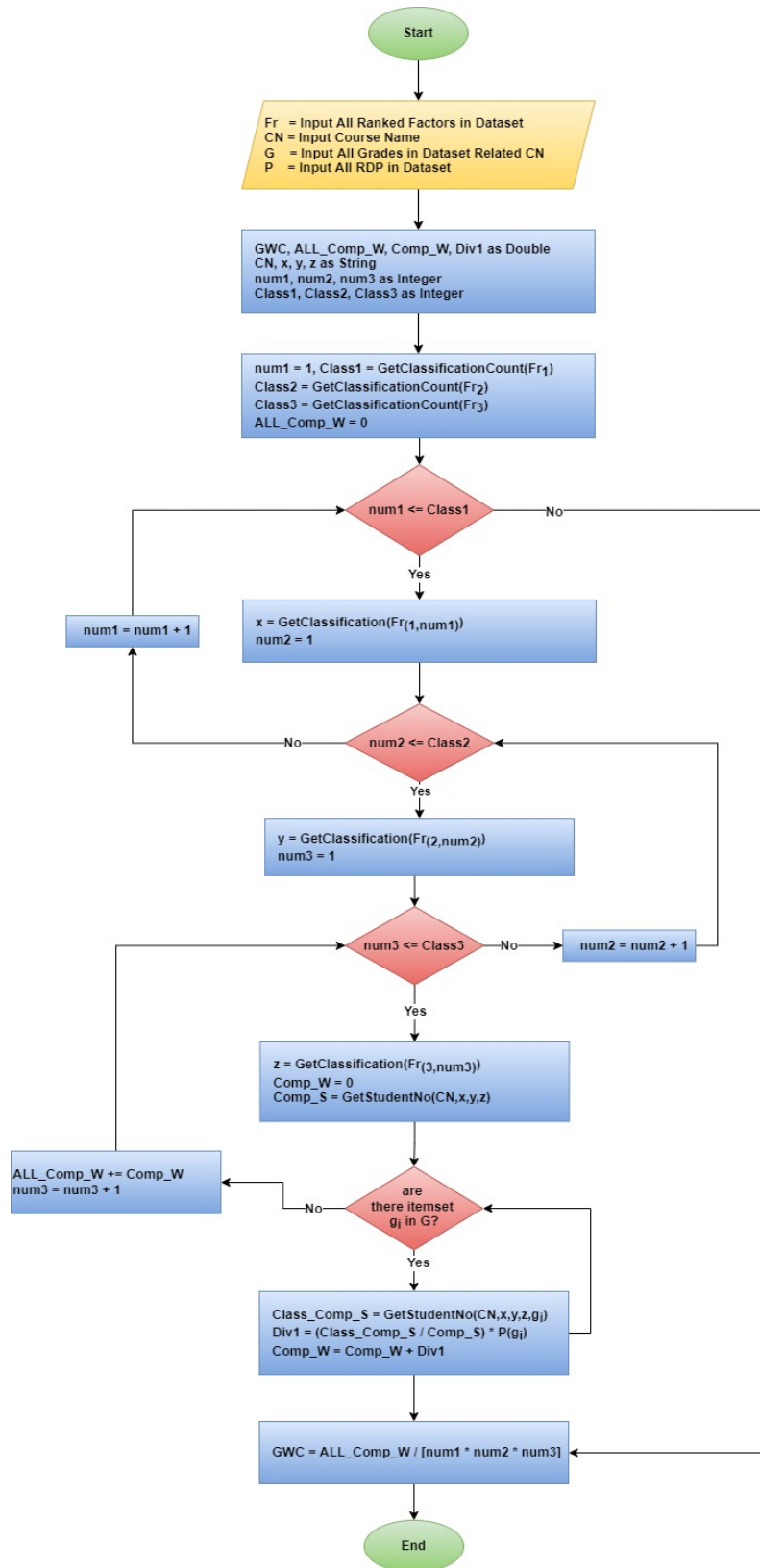


Figure 5.3: Flowchart of GWCCA

Note that: There are two main differences between GWFCA and GWCCA, which are: -

- The implementation of the algorithm in GWF was done on one of the factors only. As for GWC, it is run for all the factors that were selected from GWF step.
- The GWF step was done for all courses in dataset, but GWC will be done on the specific course.

2.4. General Weight for Plan (GWP)

That after the calculation of the difficulty index for each academic course GWC as a measure that affects students' registration for credit-hour courses, it was necessary to calculate the total difficulty index of the study plan, calculate the average academic level of the plan, and then evaluate the average difficulty index of the academic level in order to provide the appropriate recommendation with respect to the combination of courses in one level of the study plan.

The system presents the study plan and the difficulty index produced for each course based on a variety of variables such as college, plan code, and concentrate (if any).

The total difficulty index for each academic level is calculated, and then the overall difficulty index for the study plan is computed, consider the following points: -

- The first and second levels are excluded due to the fact that these are general courses and owing to the requirements of the university.
- The last level (which is an unreal level) is excluded; this level is containing only elective or free courses.

The arithmetic mean of the degree of difficulty of the academic level is found after calculating the overall difficulty index of the study plan and counting the number of levels that represent this index (the academic level difficulty index estimation). However, because this technique limits the estimation to a single value (Point Estimation), it is important that we apply a statistical method that gets us closer to the reality, which we call interval estimation by constructing the Confidence Interval (CI). The CI is a set of estimates for an unknown parameter (in this case, "the population mean") with an associated confidence level. A 95% confidence level is most common called $(1 - \alpha)100\%$. Thus, the significant level (α) is 5% (estimation error).

The purpose of the foregoing is to obtain an upper and a minimum D.I. for the level with a confidence level of 95% for this study plan.

Let the level D.I. estimation

$$\bar{X} = \left[\sum_{i=1}^n x_i \right] / n \quad (5.5)$$

Then the upper and a minimum D.I for the level with a confidence level (C.I.) of 95% is

$$C.I. = \bar{X} \pm t_{\left(\frac{\alpha}{2}, n-1\right)} \times \sqrt{\frac{s^2}{n}}, \quad (5.6)$$

where,

n is the number of calculated levels

x_i is the D.I. of the level number (i)

s^2 is the variance

$$s^2 = \left[\sum_{i=1}^n (x_i - \bar{X})^2 \right] / (n - 1) \quad (5.7)$$

t is t-distribution table with parameter $\left(\frac{0.5}{2}, n - 1\right)$ as in Figure 5.4.

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.90}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587

Figure 5.4: t-distribution table

As shown in Figure (5.4), we find

- The first parameter at $\alpha = 0.5$ then we have the value $\frac{\alpha}{2} = 0.025$ (hence the T distribution has two tails).
- The second parameter we have 8 levels.
- So, we search in T distribution table in the first parameter $(n - 1) = (8 - 1) = 7$, therefore we get the value of $t_{\left(\frac{0.5}{2}, 7\right)} = 2.365$

The Algorithm of GWPCA and its flowchart (Figure 5.5) are stated as follows:

General Weight for Plan Calculation Algorithm (GWPCA)

Define CN, All_Levels_weight as empty Array
Define $Level_weight, Sum_Levels_weight, Level_average$ as double
Define $Sum_squares, Variance, Level_weight_UB, Level_weight_LB$ as double
 $Coll \leftarrow$ input chosen college
 $Pln \leftarrow$ input chosen plan in chosen college
 $Cons \leftarrow$ input chosen concentrate in chosen plan
 $CN \leftarrow$ GetPlanByLevelCourse($Coll, Pln, Cons$)
 $Sum_Levels_weight = 0$
for $i = 3$ to $GetPalnLevelsCount(Coll, Pln, Cons) - 1$ *do*
 $Level_weight = 0$
 for each course in CN_i *do*
 $Level_weight = Level_weight + Get_GWC(course)$
 end for
 $All_Levels_weight_{(i-2)} = Level_weight$
 $Sum_Levels_weight = Sum_Levels_weight + Level_weight$
end for
 $Level_average = Sum_Levels_weight / count(All_Levels_weight)$
for $i = 1$ to $count(All_Levels_weight)$ *do*
 $Sum_Squares = Sum_Squares + (All_Levels_weight_i - Level_average)^2$
end for
 $Variance = Sum_Squares / count(All_Levels_weight)$
 $T_dis \leftarrow$ GetTDistribution($0.25, count(All_Levels_weight) - 1$)
 $Level_weight_UB = Level_average + \left[\frac{T_dis \times Variance}{\sqrt{count(All_Levels_weight)}} \right]$
 $Level_weight_LB = Level_average - \left[\frac{T_dis \times Variance}{\sqrt{count(All_Levels_weight)}} \right]$

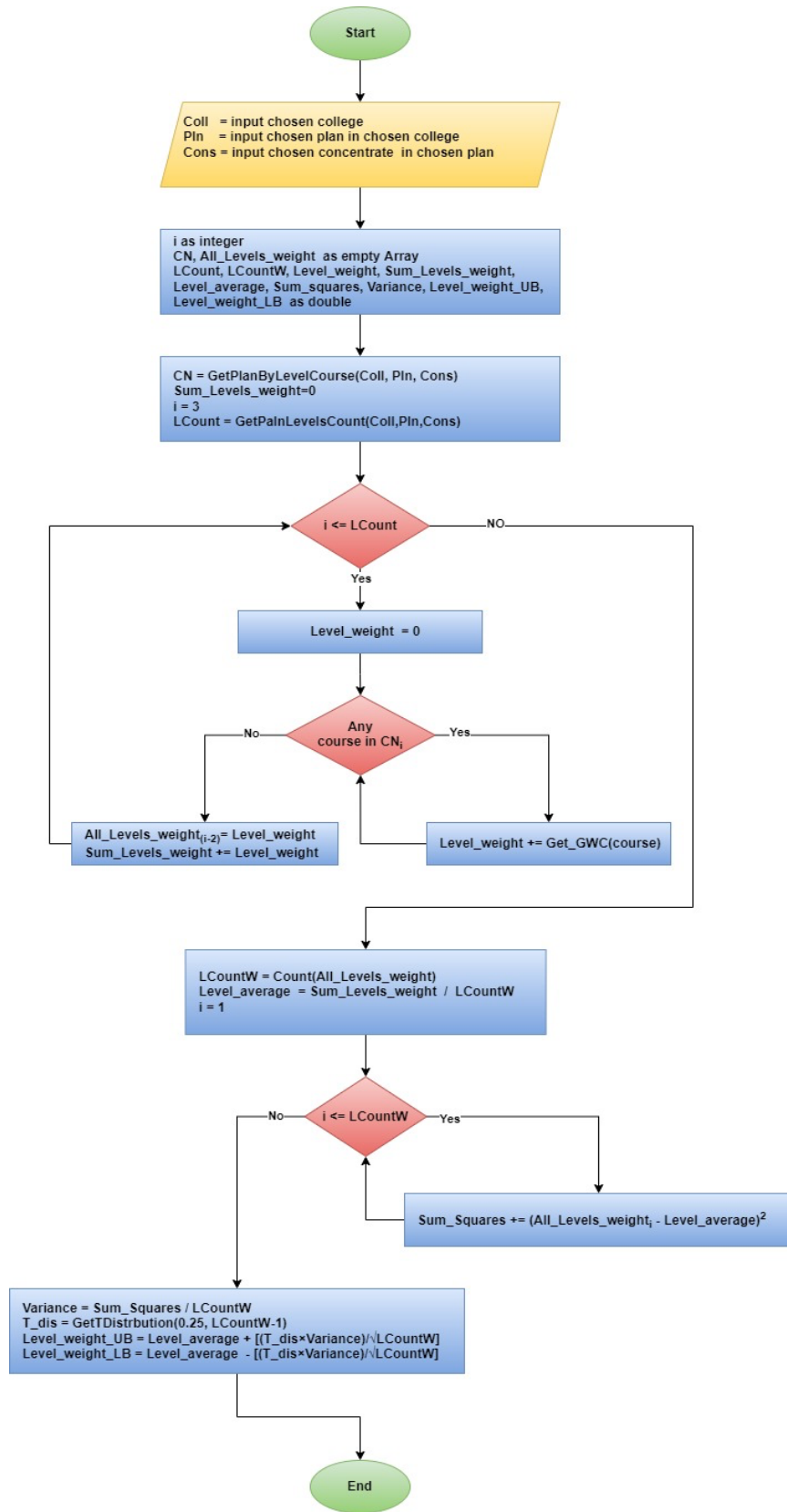


Figure 5.5: Flowchart of GWCCA

3. DMDIM Environment

The DMDIM stands for Data Mining Difficulty Index, designed using asp.net as programming language for the front end, C# is utilized and for the backend, and oracle database to store the data before / result after processing. DMDIM facilitates the students with proper course code, title, credit hours, and prerequisites along with their weights. The following is a brief description of the developed system and their main features.

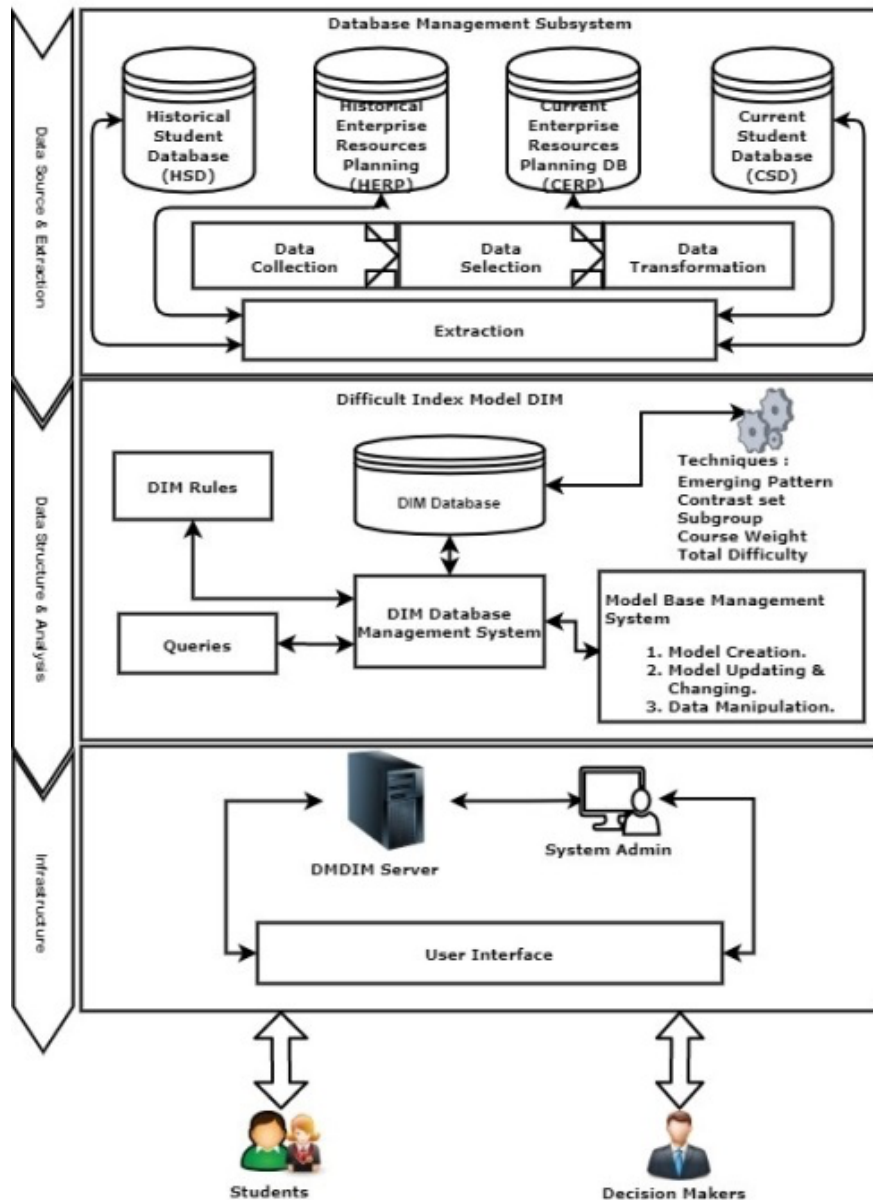
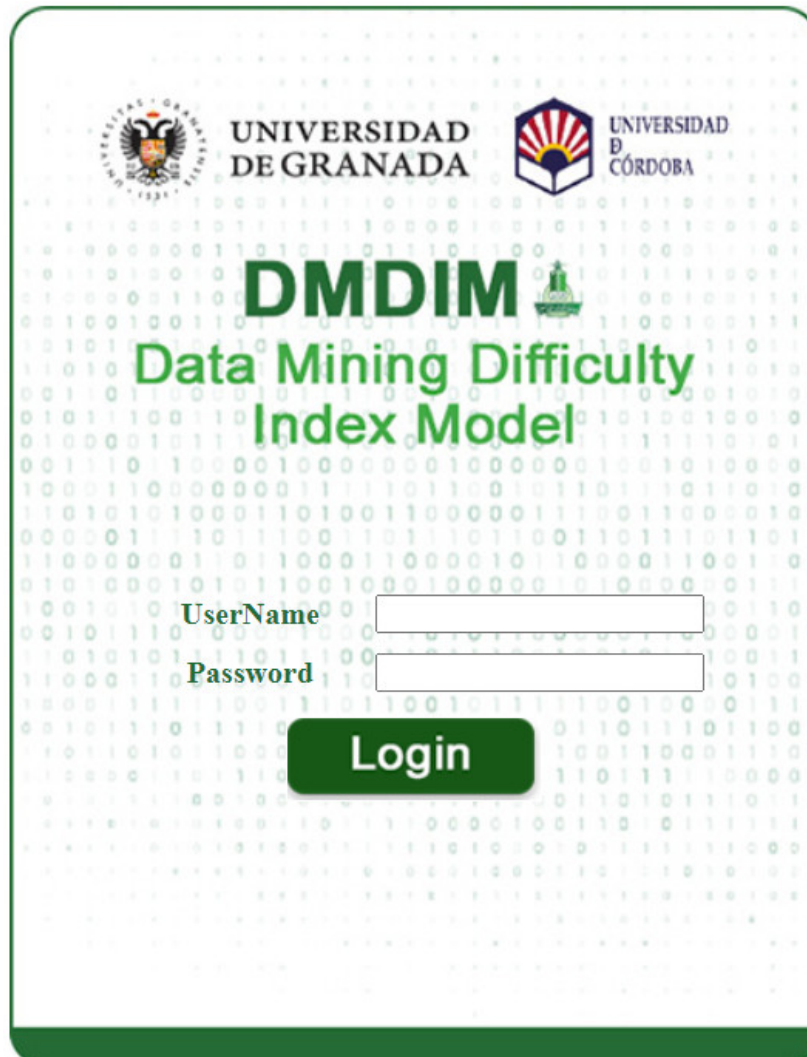


Figure 5.6: The DMDIM components

DMDIM Software Architecture

1. DMDIM starts with the **LOGIN PAGE** as shown in Figure 5.7.



The image shows the login page for the DMDIM system. It includes the logos of the Universidad de Granada and Universidad de Córdoba at the top. The main heading is 'DMDIM' in large green font, with 'Data Mining Difficulty Index Model' below it. There are two input fields for 'UserName' and 'Password', and a green 'Login' button at the bottom.

Figure 5.7: The DMDIM login page

2. The **DATASET REPORT** page which represent the statistical classifications for KAU dataset including:-

- Count of all Students, Teachers, Colleges, Study plans (Programs)
- Distribution of KAU students based on Colleges and Gender
- Distribution of DMDIM dataset students based on Gender
- Distribution of DMDIM dataset students based on Nationality
- Distribution of KAU students based on Nationality and Gender

- Distribution of KAU students based on Age
- Distribution of KAU students based on Nationality and Gender according to Colleges
- Distribution of KAU students based on Age and Gender
- Distribution of Student's grades and their GPA
- Distribution of Students Gender and Grade according to their GPA
- Distribution of Students Colleges and Grades according to GPA

Dataset Report page is shown in Figure 5.8.



Figure 5.8: Dataset Report

3. The **SYSTEM ANALYSIS** page which display
 - The problem definition (Motivation)
 - The objectives of this research (Goal)
 - How the researcher collected the data (Data Collection)
 - All factors which is the researcher depending on (Factors)
 - All factors' values and its classifications (Classifications)
 - Some useful abbreviations and notation that is making system easy for understanding (Some Useful Definitions)
 - The methodology for GWC and what is its usefulness (Calculate General Weight)
 - The process of GWF (Factors' Ranking)
 - The process of GWC (Course's D.I.)
 - The process of PWC (Plan's D.I.)
 - Some algorithms for the last 3 process (Algorithms)
 - The comments and remark employed from this work

System analysis page is shown in Figure 5.9.

DMDIM Data Mining Difficulty Index Model		UNIVERSIDAD DE GRANADA	UNIVERSIDAD DE CORDOBA
DATASET REPORT	SYSTEM ANALYSIS	FACTORS CONTROL	C. D. I. ONLINE
C. D. I. BATCH	COURSES WEIGHTS	PLAN WEIGHT	LOG-OUT
SYSTEM ANALYSIS			
1. Motivation			▼
2. Goal			▼
3. Data Collection			▼
4. Factors			▼
5. Classifications			▼
6. Some Useful Definitions			▼
7. Calculate General Weight			▼
8. GWF Calculation Process (Factor's Ranking)			▼
9. GWC Calculation Process (Course's D.I.)			▼
10. Plan Weight Calculation Process (Plan's D.I.)			▼
11. Algorithms			▼
12. Conclusion			▼
DMDIM MIT			

Figure 5.9: System analysis

4. The **FACTORS CONTROL** page which is state all factors and
 - Let user to put all factor names in a short name
 - Pointer for enable or disable the factor

- Pointer for determining that the factor is classified or not
- Display the general weight for every factor
- The variance for every factor (the dispersion of this factor about its mean)

System analysis page is shown in Figure 5.10.




  															
DATASET REPORT		SYSTEM ANALYSIS		FACTORS CONTROL		C. D. I. ONLINE		C. D. I. BATCH		COURSES WEIGHTS		PLAN WEIGHT		LOG-OUT	
FACTORS CONTROL															
Order	Factor	Short Expression	Flag	Classif.	GWF	Var.									
01	STD_GENDER_CODE	S_GENDER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.19	0.0576									
02	STD_NATIONALITY_CODE	S_NATIONALITY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1.94	0.0507									
03	STD_AGE_BY_YEAR	S_AGE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.19	0.0056									
04	STD_HOME_CITY_CODE	S_CITY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.15	0									
05	STD_HIGH_SCHOOL_GPA	S_SCHOOL_GPA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.43	0.0842									
06	STD_UNIVERSITY_GPA	S_UNIV_CUM_GPA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.34	0.6393									
07	STD_STUDY_PERIOD_YEAR	S_STDY_PERIOD_Y	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.46	0.1242									
08	STD_STUDY_PERIOD_SEMESTER	S_STDY_PERIOD_S	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.53	0.3842									
09	STD_SUMMER_SEMESTERS	S_SUMMER_S	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.53	0.3003									
10	CRS_COURSE_TYPE	C_COURSE_TYPE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.03	0.0368									
11	CRS_TIME_EARLYMORNING_WEIGHT	C_TIME_EARLYMORNING	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.89	0.6958									
12	CRS_TIMEMORNING_WEIGHT	C_TIME_MORNING	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.32	0.109									
13	CRS_TIMEAFTERNOON_WEIGHT	C_TIME_AFTERNOON	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.65	0.1579									
14	CRS_TIMEEVENING_WEIGHT	C_TIME_EVENING	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.79	0.429									
15	CRS_TIMENIGHT_WEIGHT	C_TIMENIGHT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.39	0.1756									
16	CRS_STUDENT_COUNTER	C_STD_COUNTER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.01	0.0666									
17	TCHR_GRADE_CODE	T_GRADE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.26	0.0191									
18	TCHR_EXP_TEACH_MAJOR	T_EXP_MAJOR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.13	0.0033									
19	TCHR_EXP_TEACH_COURSE	T_EXP_COURSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.14	0.0025									
20	TCHR_NATIONALITY_CODE	T_NATIONALITY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.16	0.0225									
21	TCHR_GENDER	T_GENDER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.18	0.0576									
22	TCHR_AGE_TEACH_COURSE	T_AGE_TEACH_COURSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.17	0.0168									
23	TCHR_CERT_COUNTRY_CODE	T_CERT_COUNTRY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2.1	0.0273									
99	CRS_DEPARTMENT_CODE	DEPT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1.99										
99	STD_LEGAL_WITHDRAW	LWITHD	<input type="checkbox"/>	<input checked="" type="checkbox"/>	2.15										
99	STD_PROGRAM	PROG	<input type="checkbox"/>	<input checked="" type="checkbox"/>	2.13										
99	STD_MAJOR_CODE	MAJOR	<input type="checkbox"/>	<input checked="" type="checkbox"/>	2.09										
99	STD_COLLEGE_CODE	COLL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	2.07										
99	TCHR_CERT_SITE_CODE	TCHCSC	<input type="checkbox"/>	<input type="checkbox"/>	2.27										
99	CRS_SEMSTER_LEVEL	LTERM	<input type="checkbox"/>	<input checked="" type="checkbox"/>	2.12										
FACTORS UPDATE															
DMDIM MIT															

Figure 5.10: The DMDIM components

5. The **ONLINE C. D. I.** page to calculation for GWF and GWC

- By selecting the factor with all courses for GWF which is shown in Figures 5.11 and 5.12.

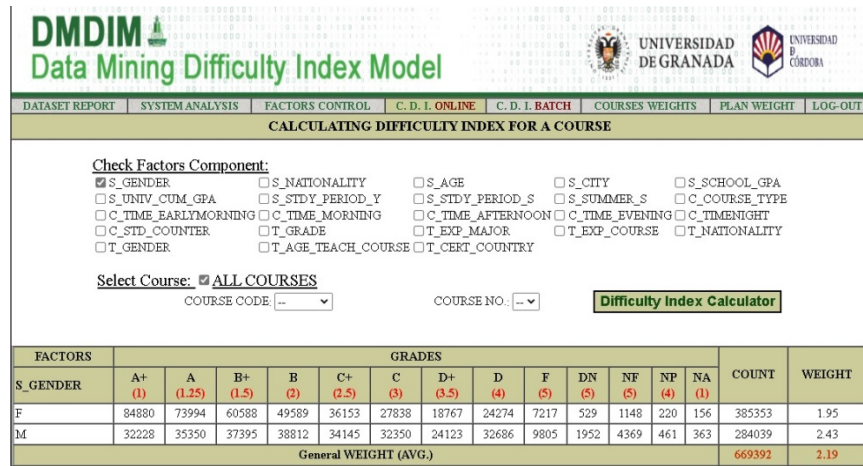


Figure 5.11: GWF calculation for Gender factor

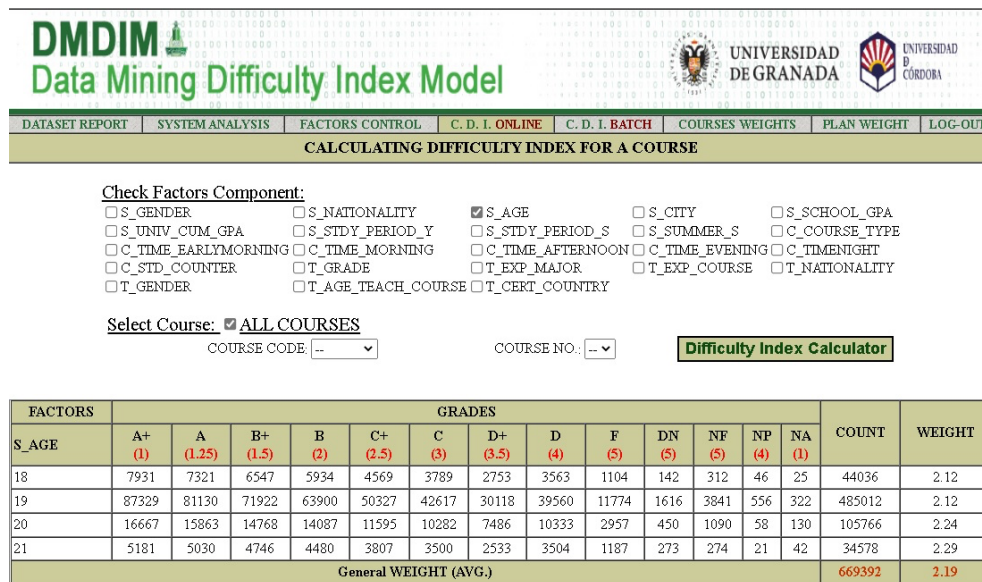


Figure 5.12: GWF calculation for Age factor

- By selecting the course code and course number with one factor (or more) for GWC which is shown in Figures 5.13 and 5.14.

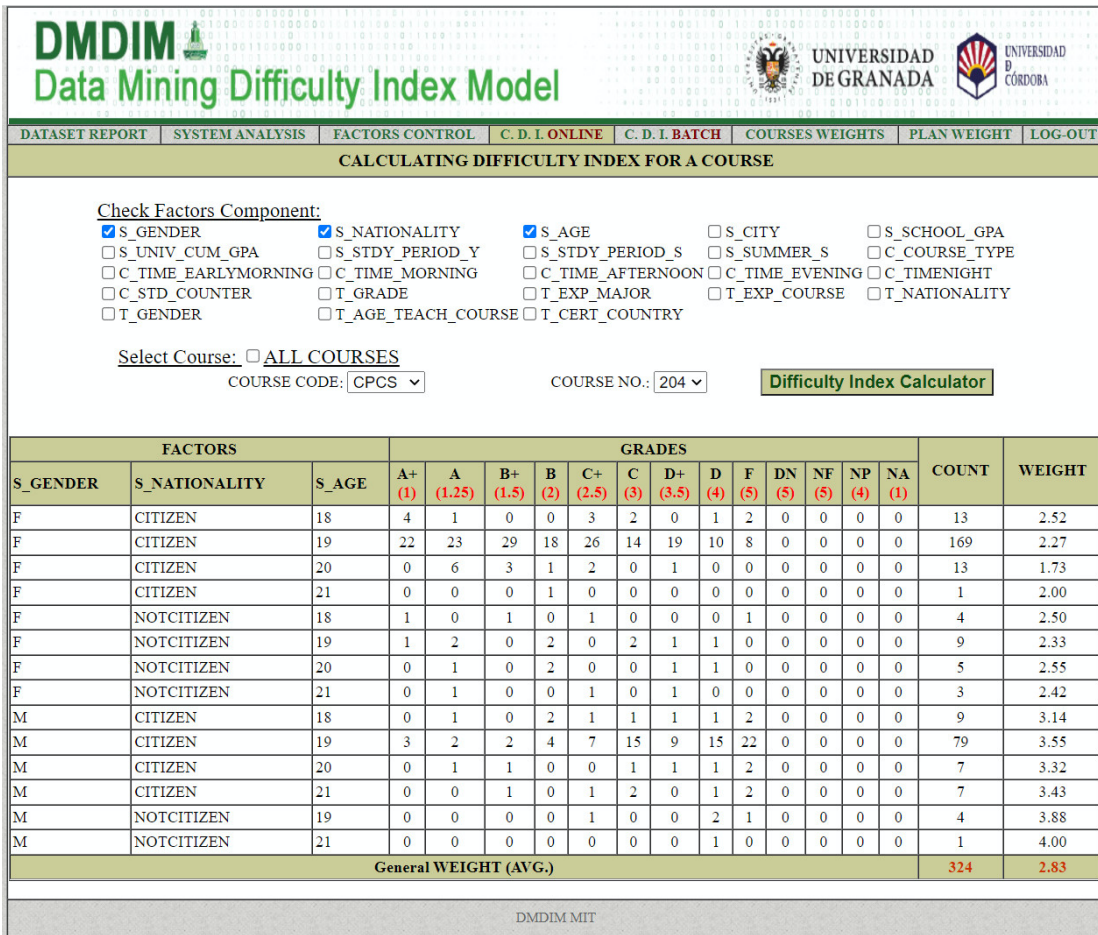


Figure 5.13: GWC calculation for CPCS-204 by three factors (Gender-Nationality-Age)

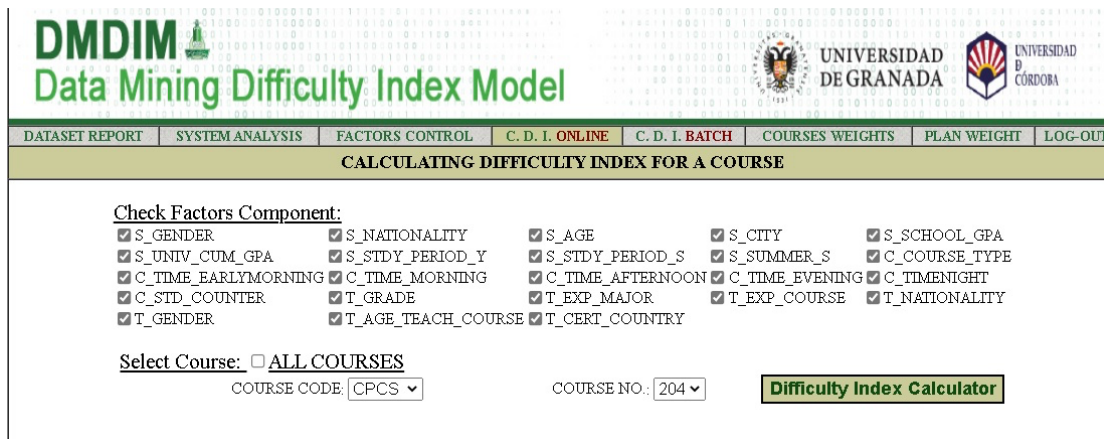


Figure 5.14: GWC calculation for CPCS-204 by all enabled factors

6. The **BATCH C. D. I.** page to GWC calculation for all courses with all factors (see Figure 5.15).

DMDIM
Data Mining Difficulty Index Model

UNIVERSIDAD DE GRANADA | UNIVERSIDAD DE CORDOBA

DATASET REPORT | SYSTEM ANALYSIS | FACTORS CONTROL | C. D. I. ONLINE | **C. D. I. BATCH** | COURSES WEIGHTS | PLAN WEIGHT | LOG-OUT

Check Factors Component:

- Check ALL
- S_GENDER
- S_UNIV_CUM_GPA
- C_TIME_EARLYMORNING
- C_STD_COUNTER
- T_GENDER
- S_NATIONALITY
- S_STDY_PERIOD_Y
- C_TIME_MORNING
- T_GRADE
- T_AGE_TEACH_COURSE
- S_AGE
- S_STDY_PERIOD_S
- C_TIME_AFTERNOON
- T_EXP_MAJOR
- T_CERT_COUNTRY
- S_CITY
- S_SUMMER_S
- C_TIME_EVENING
- T_EXP_COURSE
- S_SCHOOL_GPA
- C_COURSE_TYPE
- C_TIMENIGHT
- T_NATIONALITY

Difficulty Index Calculator

```

Course Serial :23 - Course Name : (ACCT231) Records Inserted ...
Course Serial :24 - Course Name : (ACCT310) Records Inserted ...
Course Serial :25 - Course Name : (ACCT311) Records Inserted ...
Course Serial :26 - Course Name : (ACCT312) Records Inserted ...
Course Serial :27 - Course Name : (ACCT315) Records Inserted ...
Course Serial :28 - Course Name : (ACCT316) Records Inserted ...
Course Serial :29 - Course Name : (ACCT317) Records Inserted ...
Course Serial :30 - Course Name : (ACCT333) Records Inserted ...
Course Serial :31 - Course Name : (ACCT410) Records Inserted ...
Course Serial :32 - Course Name : (ACCT411) Records Inserted ...
Course Serial :33 - Course Name : (ACCT412) Records Inserted ...
Course Serial :34 - Course Name : (ACCT413) Records Inserted ...
Course Serial :35 - Course Name : (ACCT414) Records Inserted ...
Course Serial :36 - Course Name : (ACCT415) Records Inserted ...
Course Serial :37 - Course Name : (ACCT416) Records Inserted ...
    
```

Figure 5.15: GWC calculation for all courses with all factors

7. The **COURSES WEIGHTS** page to the GWC process results for all courses (see Figures 5.16 and 5.17).

Serial	Course Code	Course Number	Course Title	Difficulty Index
1	ACC	251	FUNDAMENTALS OF FINANCIAL ACCO	3.16
2	ACC	252	FUNDAMENTALS OF MANAGERIAL ACC	2.74
3	ACC	253	Accounting Principles (1)	3.67
4	ACC	301	COST ACCOUNTING I	2.54
5	ACC	302	FINANCIAL ACCOUNTING II	2.31
6	ACC	303	FINANCIAL STATEMENTS ANALYSIS	2.41
7	ACC	304	COST ACCOUNTING II	2.34
8	ACC	305	INTERMEDIATE ACCOUNTING I	2
9	ACC	306	COMPUTERIZED ACCOUNTING	2.06
10	ACC	321	GOVERNMENT&NON-PROFIT ACCOUNTI	1.73
11	ACC	322	INTERNATIONAL ACCOUNTING	1.91
12	ACC	411	INTERMEDIATE ACCOUNTING II	2.13
13	ACC	412	AUDITING I	2.54
14	ACC	413	ACCOUNTING INFORMATION SYSTEM	2.08
15	ACC	414	ADVANCE FINANCIAL ACCOUNTING	2.45
16	ACC	415	ACCOUNTING THEORY	2.35
17	ACC	491	CO-OP. TRAINING	1.63
18	ACCT	101	PRINCIPLES OF ACCOUNTING	2.88
19	ACCT	102	PRINCIPLES OF ACCOUNTING II	1.9
20	ACCT	117	Intro.to Financial Accounting	3.31
21	ACCT	210	ACCOUNTING HEALTH COSTS	2.11
22	ACCT	213	Intro.to Managerial&Cost Acco.	3.16

Figure 5.16: GWC calculation results for all courses with all factors

Serial	Course Code	Course Number	Course Title	Difficulty Index
1	CPCS	202	PROGRAMMING I	2.75
2	CPCS	203	PROGRAMMING II	2.58
3	CPCS	204	DATA STRUCTURES (I)	2.85
4	CPCS	206	PRINCIPLES OF PROGRAMMING	2.35
5	CPCS	211	DIGITAL LOGIC DESIGN	2.64
6	CPCS	212	APPLID MATH FOR COMPUTING (I)	2.38
7	CPCS	214	COMPUTER ORGANIZATION & ARCHIT	2.7
8	CPCS	222	DISCRETE STRUCTURES I	2.55
9	CPCS	223	ANALYSIS&DESIGN OF ALGORITHMS	2.61
10	CPCS	241	DATABASE 1	1.97
11	CPCS	301	PROGRAMMING LANGUAGES	2.33
12	CPCS	302	COMPILER CONSTRUCTION	2.07
13	CPCS	323	SUMMER(WORKPLACE)TRAINING	4
14	CPCS	324	ALGORITHMS & DATA STRUCTURES 2	2.34
15	CPCS	331	ARTIFICIAL INTELLIGENCE 1	2.39
16	CPCS	351	SOFTWARE ENGINEERING 1	1.84
17	CPCS	353	SOFTWARE ENG. PRACTICES	1.74
18	CPCS	361	OPERATING SYSTEMS 1	2.16
19	CPCS	371	COMPUTER NETWORKS 1	2.34
20	CPCS	372	COMPUTER NETWORKS 2	2.56
21	CPCS	381	HUMAN-COMPUTER INTERACTION 1	2.3
22	CPCS	391	COMPUTER GRAPHICS 1	1.92
23	CPCS	403	INTERNET APPLICATION PROGRAMM.	1.85
24	CPCS	405	SOFTWARE TECHNOLOGY TOPICS	1.59
25	CPCS	425	INFORMATION SECURITY	2.36
26	CPCS	432	ARTIFICIAL INTELLIGENCE 2	1.36
27	CPCS	433	ARTIFICIAL INTELLIGENCE TOPICS	1.48
28	CPCS	454	OBJECT-ORIENTED ANALYSIS & DES	2.1
29	CPCS	463	COMPUTING SYSTEMS SECURITY	1.58
30	CPCS	466	SYSTEMS PROGRAMMING	1.08
31	CPCS	473	COMPUTER NETWORKS PRACTICE	1.85
32	CPCS	474	TCP/IP & WEB NETWORKING	1.5
33	CPCS	482	MULTIMEDIA&USER INTERFACE DES.	1.39
34	CPCS	494	SPECIAL-SELECTED TOPICS	1.7
35	CPCS	498	SENIOR PROJECT 1	1.55
36	CPCS	499	SENIOR PROJECT 2	1.36

Figure 5.17: GWC calculation results for course code CPCS with all factors

8. The **PLAN WEIGHT** page to the GWP process by using GWC for all plan's courses
- By receiving the require plan results for all courses from user (see Figure 5.18).

DMDIM
Data Mining Difficulty Index Model

DATASET REPORT SYSTEM ANALYSIS FACTORS CONTROL C. D. I. ONLINE C. D. I. BATCH COURSES WEIGHTS PLAN WEIGHT LOG-OUT

PLAN WEIGHT

Select Plan:

College: 7. IT - COMPUTING & INFORMATION TECH. (3) 13 Colleges

Plan: 1. BS-CPCS-IT (1) 69 Plans

Concentrate: 1. GENERAL 95 Concentrates

Figure 5.18: select College, Plan and Concentrate (if any) for displaying the computer science department plan

- The plan is displayed with the GWC for every course. So the cumulative GWC for every level is calculated (see Figure 5.19).

LEVEL : BS-CPCS-01				LEVEL : BS-CPCS-02			
Course	Course Title	Credit Hours	D. I.	Course	Course Title	Credit Hours	D. I.
COMM-101	COMMUNICATION SKILLS	3	1.54	BIO-110	GENERAL BIOLOGY (1)	3	3.21
CPIT-100	COMPUTER SKILLS	3	1.99	CHEM-110	GENERAL CHEMISTRY I	3	2.82
ELL-101	ENGLISH LANGUAGE I	0	2.19	ELI-103	ENGLISH LANGUAGE III	2	3.48
ELI-102	ENGLISH LANGUAGE II	2	2.61	ELI-104	ENGLISH LANGUAGE IV	2	3.42
MATH-110	GENERAL MATHEMATICS (1)	3	3.13	STAT-110	GENERAL STATISTICS (1)	3	2.97
PHYS-110	GENERAL PHYSICS (1)	3	2.89	ELCS-110	ENGLISH LANGUAGE	3	2.72
ELCS-100	ENGLISH LANGUAGE	3	2.63				
Level Sum		14	16.98	Level Sum		13	18.62

LEVEL : BS-CPCS-03				LEVEL : BS-CPCS-04			
Course	Course Title	Credit Hours	D. I.	Course	Course Title	Credit Hours	D. I.
CPCS-202	PROGRAMMING I	3	2.75	ARAB-101	ARABIC LANGUAGE (1)	3	2.3
CPIT-201	INTRODUCTION TO COMPUTING	3	2.52	CPCS-203	PROGRAMMING II	3	2.58
CPIT-221	TECHNICAL WRITING	2	2.33	CPCS-222	DISCRETE STRUCTURES I	3	2.55
ISLS-101	ISLAMIC CULTURE (1)	2	1.94	ISLS-201	ISLAMIC CULTURE (2)	2	1.82
STAT-210	PROBABILITY THEORY	3	2.6	MATH-202	CALCULUS II	3	3.64
Level Sum		13	12.14	Level Sum		15	12.89
Cumulative Sum		(13)	(12.14)	Cumulative Sum		(28)	(25.03)

LEVEL : BS-CPCS-05				LEVEL : BS-CPCS-06			
Course	Course Title	Credit Hours	D. I.	Course	Course Title	Credit Hours	D. I.
CPCS-204	DATA STRUCTURES (1)	3	2.85	CPCS-214	COMPUTER ORGANIZATION & ARCHIT	3	2.7
CPCS-211	DIGITAL LOGIC DESIGN	3	2.64	CPCS-223	ANALYSIS&DESIGN OF ALGORITHMS	3	2.61
CPCS-212	APPLID MATH FOR COMPUTING (1)	4	2.38	CPCS-241	DATABASE 1	3	1.97
PHYS-202	GENERAL PHYSICS II	4*	2.72 *	CPCS-301	PROGRAMMING LANGUAGES	3	2.33
CHEM-202	GENERAL CHEMISTRY (II)	4*	2.8 *	STAT-352	APPLIED PROBABILITY & RANDOM	3	2.52
BIO-202	GENERAL BIOLOGY (2)	4*	2.25 *	Level Sum		15	12.13
BIOC-371	BIOCHEM FOR NON-BIOCHEMISTS	4*	2.35 *	Cumulative Sum		(57)	(47.56)
Level Sum		14	10.4				
Cumulative Sum		(42)	(35.43)				

LEVEL : BS-CPCS-07				LEVEL : BS-CPCS-08			
Course	Course Title	Credit Hours	D. I.	Course	Course Title	Credit Hours	D. I.
CPCS-324	ALGORITHMS & DATA STRUCTES 2	3	2.34	CPCS-302	COMPILER CONSTRUCTION	3	2.07
CPCS-331	ARTIFICIAL INTELLIGENCE 1	3	2.39	CPCS-381	HUMAN-COMPUTER INTERACTION 1	2	2.3
CPCS-351	SOFTWARE ENGINEERING 1	3	1.84	CPCS-391	COMPUTER GRAPHICS 1	3	1.92
CPCS-361	OPERATING SYSTEMS 1	3	2.16	ISLS-301	ISLAMIC CULTURE (3)	2	1.62
CPCS-371	COMPUTER NETWORKS 1	3	2.34	Level Sum		10	7.91
CPIS-334	INTR.TO SOFTWARE PROJECT MANGE	2	2.19	Cumulative Sum		(84)	(68.73)
Level Sum		17	13.26				
Cumulative Sum		(74)	(60.82)				

LEVEL : BS-CPCS-09				LEVEL : BS-CPCS-10			
Course	Course Title	Credit Hours	D. I.	Course	Course Title	Credit Hours	D. I.
ARAB-201	ARABIC LANGUAGE (2)	3	1.87	CPCS-499	SENIOR PROJECT 2	3	1.36
CPCS-323	SUMMER(WORKPLACE)/TRAINING	0	4	CPIS-428	PROFESSIONAL COMPUTING ISSUES	2	1.84
CPCS-498	SENIOR PROJECT 1	1	1.55	ISLS-401	ISLAMIC CULTURE (4)	2	1.62
Level Sum		4	7.42	Level Sum		7	4.82
Cumulative Sum		(88)	(76.15)	Cumulative Sum		(95)	(80.97)

Figure 5.19: The Plan with the cumulative GWC for every level is calculated

- The last level in plan is considered the elective (optional) courses (see Figure 5.20).

LEVEL : BS-CPCS-11			
Course	Course Title	Credit Hours	D. I.
CPCS-353	SOFTWARE ENG. PRACTICES	3*	1.74 *
CPCS-372	COMPUTER NETWORKS 2	3*	2.56 *
CPCS-403	INTERNET APPLICATION PROGRAMM	3*	1.85 *
CPCS-404		3*	0 *
CPCS-405	SOFTWARE TECHNOLOGY TOPICS	3*	1.59 *
CPCS-413		3*	0 *
CPCS-414		3*	0 *
CPCS-424		3*	0 *
CPCS-425	INFORMATION SECURITY	3*	2.36 *
CPCS-432	ARTIFICIAL INTELLIGENCE 2	3*	1.36 *
CPCS-433	ARTIFICIAL INTELLIGENCE TOPICS	3*	1.48 *
CPCS-442		3*	0 *
CPCS-454	OBJECT-ORIENTED ANALYSIS & DES	3*	2.1 *
CPCS-457		3*	0 *
CPCS-462		3*	0 *
CPCS-494	SPECIAL/SELECTED TOPICS	3*	1.7 *
CPCS-474	TCP/IP & WEB NETWORKING	3*	1.5 *
CPCS-473	COMPUTER NETWORKS PRACTICE	3*	1.85 *
CPCS-482	MULTIMEDIA&USER INTERFACE DES.	3*	1.39 *
CPCS-463	COMPUTING SYSTEMS SECURITY	3*	1.58 *
CPCS-464		3*	0 *
CPCS-465		3*	0 *
CPCS-466	SYSTEMS PROGRAMMING	3*	1.08 *
Level Sum		18	24.14

Figure 5.20: Optional courses and difficulty index

- Finally, the last part for calculating the statistical measurements to find the minimum and maximum level weight. Therefore, discovering the pad level(s) in study plan (see Figure 5.21).

MEASUREMENTS	VALUES						
<i>COVERED LEVELS</i> (n)	8						
<i>SET OF D.I(s)</i> (X) = $\{x_1, x_2, \dots, x_n\}$	{ 12.14,12.89,10.4,12.13,13.26,7.91,7.42,4.82 }						
<i>Average</i> (\bar{X}) = $\sum_{i=1}^n x_i / n$	10.12 (Estimation Point)						
<i>Variance</i> (S^2) = $\sum_{i=1}^n (x_i - \bar{X})^2 / (n - 1)$	9.43						
<i>t</i> distribution with parameter (0.025, 7) * $\sqrt{\frac{S^2}{n}}$	2.365 * 1.161 = 2.74						
<i>The 95% confidence Interval (C.I.)</i> = $\left[\bar{X} \pm t_{\left(\frac{\alpha}{2}, n-1\right)} \sqrt{\frac{S^2}{n}} \right]$, where <i>Level of Significance</i> (α) = 0.05 <i>(t)</i> distribution with $\frac{\alpha}{2}$ and $(n - 1)$ degree of freedom	[7.38 , 12.86]						
<table border="1"> <thead> <tr> <th>THE MINMUM D.I. PER LEVEL</th> <th>THE AVERAGE D.I. PER LEVEL</th> <th>THE MAXIMUM D.I. PER LEVEL</th> </tr> </thead> <tbody> <tr> <td>7.38</td> <td>10.12</td> <td>12.86</td> </tr> </tbody> </table>		THE MINMUM D.I. PER LEVEL	THE AVERAGE D.I. PER LEVEL	THE MAXIMUM D.I. PER LEVEL	7.38	10.12	12.86
THE MINMUM D.I. PER LEVEL	THE AVERAGE D.I. PER LEVEL	THE MAXIMUM D.I. PER LEVEL					
7.38	10.12	12.86					

Figure 5.21: Statistical measures

4. REMARKS AND CONCLUSIONS

Using mathematics methods such as variance, arithmetic mean, and the supplement of cumulative average, a weight for the degree of difficulty of the courses was calculated at King Abdulaziz University, which will assist academic departments in faculties in developing study plans that suit students to graduate at the specified time according to the study plan proposed by the department and reduce the dropout phenomenon.

As a result, we reviewed the objectives of the Ph.D., its results, and its impact on the case study, which was the problem of the seventh and eighth semesters of the Department of Computer Science, College of Computing and Information Technology, King Abdulaziz University. As it appeared that the two semesters were not equitable and that there was no appropriate distribution of courses, as shown in Figure 5.22 with the result called Course Difficulty index:

LEVEL: Bachelor's in computer science Department # 7				LEVEL: Bachelor's in computer science Department # 8			
Code	Course Name	C. H	CDI	Code	Course Name	C. H	CDI
CPCS-324	ALGORITHMS & DATA STRUCTURES 2	3	2.3 4	CPCS-302	COMPILER CONSTRUCTION	3	2.0 7
CPCS-331	ARTIFICIAL INTELLIGENCE 1	3	2.3 9	CPCS-381	HUMAN-COMPUTER INTERACTION 1	2	2.3
CPCS-351	SOFTWARE ENGINEERING 1	3	1.8 4	CPCS-391	COMPUTER GRAPHICS 1	3	1.9 2
CPCS-361	OPERATING SYSTEMS 1	3	2.1 6	ISLS-301	ISLAMIC CULTURE (3)	2	1.6 2
CPCS-371	COMPUTER NETWORKS 1	3	2.3 4	Level Sum		10	7.9 1
CPIS-334	INTR.TO SOFTWARE PROJECT MANGE	2	2.1 9				
Level Sum		17	13. 26				

Figure 5.22: the seventh and eighth semesters with the Course Difficulty index

As a result, the study sample chosen from the student community has an impact on the factors that affect the study's selection process, thus the researcher was keen to collect data from a variety of colleges and programs. The following are some of the conclusions that have been reached:

- The effect of the factor on the difficulty of academic courses is based on a scientific approach that verifies or disproves this. For example, contrary to popular belief, the academic program aspect had no impact on the course's difficulty index.
- Some elements have a stronger impact on the student's perception of course difficulty than others. For example, it is obvious that the student's home city has a bigger impact on the course's difficulty than the number of times the teacher teaches the course, and that the latter has a greater impact than the teacher's country, and so on...
- The predicted level of difficulty is as close to the truth as possible due to the obvious great number of factors and their classifications gathered from historical data of past students (the degree of difficulty of the course in the study plan).

CHAPTER VI. Conclusions and Future Work

1. CONCLUSIONS

Nowadays, most education institutions use e-learning systems, and some of them are also considering learning analytics. In higher education, for example, learning analytics tools are being used to improve the educational process in many ways: easing the learning process, improving the final grades, reducing dropout and, therefore, increasing the reputation, etc. Educational data mining is an emerging discipline, considered as an interdisciplinary research area (it includes methods and techniques from machine learning, statistics, data mining and data analysis) to analyze data that come from education settings.

This Dissertation has addressed one of the most complex problems in education, that is, long-term course planning. In this problem, the aim is to provide advises or help decision makers in selecting the most student's suitable courses and courses combination for one semester at educational organizations. The final aim is to improve the student's performance, and then reducing the repetition rate, exceeding the regular period and university dropping out. In this regard, the research work has been divided into two main methods: a sequential pattern mining approach acting as a recommendation of courses, and a course difficulty index metric to measure the eligibility of a specific course. A maximum difficulty value is considered so students should make elections according to the provided metrics. These two research methods have been applied to a real scenario. We took real information gathered from King Abdulaziz University, and these data came from multiple sources in heterogeneous environments such as ODUSPLUS system that use Oracle database, ANJEZ system that use DB2 database, NOOR, ENTEMAA, and ESTEBANA that store data in SQL database.

The sequential pattern mining approach returned excellent results in the study case. This methodology was able to obtain interesting courses for each specific student depending on previous courses they already passed. The recommendation was done according to what courses were already taken by similar students and obtaining an excellent final GPA. Additionally, this methodology was able to provide full study plans from the very early stages of the degree, which is important for new students. Additionally, a course difficulty index metric has been proposed and an online application is also used to describe which courses are more difficult so the students can choose different courses according to a maximum difficulty value.

As a result, when designing departmental study plans, every educational institution should not rely just on credit hours or units but should also consider a new factor such as the course difficulty index. Consequently, university decision-makers do not need to spend extra time on such issues, so this helps the university to conserve resources.

2. FUTURE WORK

The proposed methods have achieved excellent results as it was demonstrated on a real scenario. However, further studies are necessary to be addressed:

- Apply the proposed methods to all levels of education such as institutions, schools, etc.
- Run the proposed approaches with data from other universities to know the consistency and convergence to obtain the best impact factors that affect course difficulty index because there are differences between educational environments.
- Finding the best distribution of study courses plan for each semester based on the credit hours and the proposed course difficulty index, considering the course prerequisites.

References

- [AAAB17] AL-SHEHRI, HUDA ; AL-QARNI, AMANI ; AL-SAATI, LEENA ; BATOAQ, ARWA ; BADUKHEN, HAIFA ; ALRASHED, SALEH ; ALHIYAFI, JAMAL ; OLATUNJI, SUNDAY O.: Student performance prediction using Support Vector Machine and K-Nearest Neighbor. In: *Canadian Conference on Electrical and Computer Engineering* (2017) — ISBN 9781509055388
- [ACDL16] ALHARBI, ZAHYAH ; CORNFORD, JAMES ; DOLDER, LIAM ; DE LA IGLESIA, BEATRIZ: Using data mining techniques to predict students at risk of poor performance. In: *Proceedings of 2016 SAI Computing Conference, SAI 2016* (2016), pp. 523–531 — ISBN 9781467384605
- [AFGY02] AYRES, JAY ; FLANNICK, JASON ; GEHRKE, JOHANNES ; YIU, TOMI: Sequential pattern mining using A bitmap representation. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2002), pp. 429–435
- [AgIS93] AGRAWAL, RAKESH ; IMIELIŃSKI, TOMASZ ; SWAMI, ARUN: Mining Association Rules Between Sets of Items in Large Databases. In: *ACM SIGMOD Record*. vol. 22, 1993, pp. 207–216
- [AgSh14] AGARWAL, KULDEEP ; SHIVPURI, RAJIV: Knowledge discovery in steel bar rolling mills using scheduling data and automated inspection. In: *Journal of Intelligent Manufacturing* vol. 25 (2014), Nr. 6, pp. 1289–1299
- [AgSr95] AGRAWAL, RAKESH ; SRIKANT, RAMAKRISHNAN: Mining sequential patterns. In: *Proceedings - International Conference on Data Engineering* (1995), pp. 3–14
- [AhIA15] AHMAD, FADHILAH ; ISMAIL, NUR HAFIEZA ; AZIZ, AZWA ABDUL: The prediction of students' academic performance using classification data mining techniques. In: *Applied Mathematical Sciences* vol. 9 (2015), Nr. 129, pp. 6415–6426
- [AjAS20] AJIBADE, SAMUEL SOMA M. ; AHMAD, NOR BAHIAH ; SHAMSUDDIN, SITI MARIYAM: A data mining approach to predict academic performance of students using ensemble techniques. In: *Advances in Intelligent Systems and Computing* vol. 940 (2020), pp. 749–760 — ISBN 9783030166564
- [AKPS19] ANUSHA, M. ; KARTHIK, K. ; PADMINI RANI, P. ; SRIKANTH, V.: Prediction of student performance using machine learning. In: *International Journal of Engineering and Advanced Technology* vol. 8 (2019), Nr. 6, pp. 247–255
- [AIAI16] AL-BADARENAH, AMER ; ALSAKRAN, JAMAL: An Automated Recommender System for Course Selection. In: *International Journal of Advanced Computer Science and Applications* vol. 7 (2016), Nr. 3
- [Alba16] AL-BASHIR, ADNAN: Applying Total Quality Management Tools Using QFD at Higher Education Institutions in Gulf Area (Case Study: ALHOSN University). In: *International Journal of Production Management and Engineering* vol. 4 (2016), Nr. 2, p. 87
- [AIBS00] AL-TURKI, YUSUF A ; BILGRAMI, ANWAR L ; SAID, ABOL-ELA: a Case Study of Key Performance Indicators in Scientific Research in a Middle Eastern University. In: *International Journal of Latest Research in Science and Technology ISSN* vol. 428, Nr. 621, pp. 2278–5299

- [AICA19] ALMASRI, AMMAR ; CELEBI, ERBUG ; ALKHAWALDEH, RAMI S.: EMT: Ensemble meta-based tree model for predicting student performance. In: *Scientific Programming* vol. 2019 (2019)
- [AMAH17] ASIF, RAHEELA ; MERCERON, AGATHE ; ALI, SYED ABBAS ; HAIDER, NAJMI GHANI: Analyzing undergraduate students' performance using educational data mining. In: *Computers and Education* vol. 113 (2017), pp. 177–194
- [AmHA16] AMRIEH, ELAF ABU ; HAMTINI, THAIR ; ALJARA, IBRAHIM: Mining Educational Data to Predict Student's academic Performance using Ensemble Methods. In: *International Journal of Database Theory and Application* vol. 9 (2016), Nr. 8, pp. 119–136
- [ASVG17] ATHERTON, MIRELLA ; SHAH, MAHSOOD ; VAZQUEZ, JENNY ; GRIFFITHS, ZOE ; JACKSON, BRIAN ; BURGESS, CATHERINE: Using learning analytics to assess student engagement and academic outcomes in open access enabling programmes. In: *Open Learning* vol. 32 (2017), Nr. 2, pp. 119–136
- [AzSa08] AZEVEDO, ANA ; SANTOS, MANUEL FILIPE: KDD, semma and CRISP-DM: A parallel overview. In: *MCCSIS'08 - IADIS Multi Conference on Computer Science and Information Systems; Proceedings of Informatics 2008 and Data Mining 2008*, 2008 — ISBN 9789728924638, pp. 182–185
- [BaPA07] BA-OMAR, HAFIDH ; PETROUNIAS, ILIAS ; ANWAR, FAHAD: A framework for using web usage mining to personalise e-learning. In: *Proceedings - The 7th IEEE International Conference on Advanced Learning Technologies, ICALT 2007*, 2007 — ISBN 076952916X, pp. 937–938
- [BaSc03] BASSIRI, DINA ; SCHULZ, E. MATTHEW: Constructing a universal scale of high school course difficulty. In: *Journal of Educational Measurement* vol. 40 (2003), Nr. 2, pp. 147–161
- [BBSS16] BASU, APARNA ; BANSHAL, SUMIT KUMAR ; SINGHAL, KHUSHBOO ; SINGH, VIVEK KUMAR: Designing a Composite Index for research performance evaluation at the national or regional level: ranking Central Universities in India. In: *Scientometrics* vol. 107 (2016), Nr. 3, pp. 1171–1193
- [BCSJ17] BHUMICHITR, KIRATIJUTA ; CHANNARUKUL, SONGSAK ; SAEJIEM, NATTACHAI ; JIANTHAPTHAKSIN, RACHSUDA ; NONGPONG, KWANKAMOL: Recommender Systems for university elective course recommendation. In: *Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering, JCSSE 2017* (2017) — ISBN 9781509048342
- [Berk06] BERKHIN, P.: A survey of clustering data mining techniques. In: *Grouping Multidimensional Data: Recent Advances in Clustering* (2006), pp. 25–71 — ISBN 354028348X
- [BhSB18] BHARARA, SANYAM ; SABITHA, SAI ; BANSAL, ABHAY: Application of learning analytics using clustering data Mining for Students' disposition analysis. In: *Education and Information Technologies* vol. 23 (2018), Nr. 2, pp. 957–984
- [BiFM14] BIENKOWSKI, MARIE ; FENG, MINGYU ; MEANS, BARBARA: Enhancing teaching and learning through educational data mining and learning analytics: An issue brief. In: *Educational Improvement Through Data Mining and Analytics* (2014), pp. 1–60 — ISBN 9781633213746
- [BoCR18] BOGARÍN, ALEJANDRO ; CEREZO, REBECA ; ROMERO, CRISTÓBAL: A survey on educational process mining. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* vol. 8 (2018), Nr. 1

- [BOHG13] BOBADILLA, J. ; ORTEGA, F. ; HERNANDO, A. ; GUTIÉRREZ, A.: Recommender systems survey. In: *Knowledge-Based Systems* vol. 46 (2013), pp. 109–132
- [BSZE17] BANKSHINATEGH, BEHDAD ; SPANAKIS, GERASIMOS ; ZAIANE, OSMAR ; ELATIA, SAMIRA: A course recommender system based on graduating attributes. In: *CSEDU 2017 - Proceedings of the 9th International Conference on Computer Supported Education*. vol. 1, 2017 — ISBN 9789897582394, pp. 347–354
- [BZEI18] BAKHSHINATEGH, BEHDAD ; ZAIANE, OSMAR R. ; ELATIA, SAMIRA ; IPPERCIEL, DONALD: Educational data mining applications and tasks: A survey of the last 10 years. In: *Education and Information Technologies* vol. 23 (2018), Nr. 1, pp. 537–553
- [Chan06] CHANG, LIN: Applying data mining to predict college admissions yield: A case study. In: *New Directions for Institutional Research* vol. 2006 (2006), Nr. 131, pp. 53–68
- [Chap00] CHAPMAN, P: CRISP-DM 1.0 Step-by-step data mining guide. SPSS (2000)
- [ChCR21] CHANGO, WILSON ; CEREZO, REBECA ; ROMERO, CRISTÓBAL: Multi-source and multimodal data fusion for predicting academic performance in blended learning university courses. In: *Computers and Electrical Engineering* vol. 89 (2021)
- [ChDo14] CHEN, JENG FUNG ; DO, QUANG HUNG: A cooperative Cuckoo Search-hierarchical adaptive neuro-fuzzy inference system approach for predicting student academic performance. In: *Journal of Intelligent and Fuzzy Systems* vol. 27 (2014), Nr. 5, pp. 2551–2561
- [ChLC16] CHANG, PEI CHANN ; LIN, CHENG HUI ; CHEN, MENG HUI: A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems. In: *Algorithms* vol. 9 (2016), Nr. 3
- [DEAA14] DARAMOLA, OLAWANDE ; EMEBO, ONYEKA ; AFOLABI, IBUKUN ; AYO, CHARLES: Implementation of an Intelligent Course Advisory Expert System. In: *International Journal of Advanced Research in Artificial Intelligence* vol. 3 (2014), Nr. 5
- [Dele10] DELEN, DURSUN: A comparative analysis of machine learning techniques for student retention management. In: *Decision Support Systems* vol. 49 (2010), Nr. 4, pp. 498–506
- [DeVH16] DEVASIA, TISMY ; VINUSHREE, T. P. ; HEGDE, VINAYAK: Prediction of students performance using Educational Data Mining. In: *Proceedings of 2016 International Conference on Data Mining and Advanced Computing, SAPIENCE 2016* (2016), pp. 91–95 — ISBN 9781467385947
- [DGWC86] DELQUADRI, JOE ; GREENWOOD, CHARLES R. ; WHORTON, DEBRA ; CARTA, JUDITH J. ; HALL, R. VANCE: Classwide Peer Tutoring. In: *Exceptional Children* vol. 52 (1986), Nr. 6, pp. 535–542
- [DLAA17] DAUD, ALI ; LYTRAS, MILTIADIS D. ; ALJOHANI, NAIF RADI ; ABBAS, FARHAT ; ABBASI, RABEEH AYAZ ; ALOWIBDI, JALAL S.: Predicting student performance using advanced learning analytics. In: *26th International World Wide Web Conference 2017, WWW 2017 Companion* (2017), pp. 415–421 — ISBN 9781450349147
- [DoCh13] DO, QUANG HUNG ; CHEN, JENG FUNG: A neuro-fuzzy approach in the classification of students' academic performance. In: *Computational Intelligence and Neuroscience* vol. 2013 (2013)

- [Dong99] DONG, GUOZHU: Discovering Trends and Differences. In: *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 43–52
- [Edin12] EDIN, OSMANBEGOVIC: Data Mining Approach for Predicting Student Performance Economic (2012)
- [EsYa00] ESTIVILL-CASTRO, VLADIMIR ; YANG, JIANHUA: Fast and robust general purpose clustering algorithms. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. vol. 1886, 2000, pp. 208–218
- [Farh11] FARHADIEH, FARINAZ: A Statistical Framework for Quantifying Adaptive Behavioural Risk for the Banking Industry (2011)
- [FGCT14] FOURNIER-VIGER, PHILIPPE ; GOMARIZ, ANTONIO ; CAMPOS, MANUEL ; THOMAS, RINCY: Fast vertical mining of sequential patterns using co-occurrence information. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 8443 LNAI (2014), Nr. PART 1, pp. 40–52
- [FGGM13] FOURNIER-VIGER, PHILIPPE ; GOMARIZ, ANTONIO ; GUENICHE, TED ; MWAMIKAZI, ESPÉRANCE ; THOMAS, RINCY: TKS: Efficient mining of top-k sequential patterns. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 8346 LNAI (2013), Nr. PART 1, pp. 109–120 — ISBN 9783642539138
- [FGGS15] FOURNIER-VIGER, PHILIPPE ; GOMARIZ, ANTONIO ; GUENICHE, TED ; SOLTANI, AZADEH ; WU, CHENG WEI ; TSENG, VINCENT S.: SPMF: A java open-source pattern mining library. In: *Journal of Machine Learning Research* vol. 15 (2015), pp. 3389–3393
- [FiVä11] FIEDLER, SEBASTIAN H.D. ; VÄLJATAGA, TERJE: Personal learning environments: Concept or technology? In: *International Journal of Virtual and Personal Learning Environments* vol. 2 (2011), Nr. 4, pp. 1–11
- [FMSF13] FATTAH MASHAT, ABDUL ; M.FOUAD, MOHAMMED ; S. YU, PHILIP ; F. GHARIB, TAREK: Discovery of Association Rules from University Admission System Data. In: *International Journal of Modern Education and Computer Science* vol. 5 (2013), Nr. 4, pp. 1–7
- [FrBa19] FRANCIS, BINDHIA K. ; BABU, SUVANAM SASIDHAR: Predicting Academic Performance of Students Using a Hybrid Data Mining Approach. In: *Journal of Medical Systems* vol. 43 (2019), Nr. 6
- [GaKa04] GARRISON, D. RANDY ; KANUKA, HEATHER: Blended learning: Uncovering its transformative potential in higher education. In: *Internet and Higher Education* vol. 7 (2004), Nr. 2, pp. 95–105
- [GaLi15] GANESHAN, KATHIRAVELU ; LI, XIAOSONG: An intelligent student advising system using collaborative filtering. In: *Proceedings - Frontiers in Education Conference, FIE*. vol. 2015, 2015 — ISBN 9781479984534
- [GaMo17] GADAL, SAAD MOHAMED ALI MOHAMED ; MOKHTAR, RANIA A.: Anomaly detection approach using hybrid algorithm of data mining technique. In: *Proceedings - 2017 International Conference on Communication, Control, Computing and Electronics Engineering, ICCCEE 2017* (2017) — ISBN 9781509018093

- [GoDP17] GORADE, S.M. ; DEO, A. ; PUROHIT, P.: A Study Some Data Mining Classification Techniques. In: *International Journal of Modern Trends in Engineering & Research* vol. 4 (2017), Nr. 1, pp. 210–215
- [GSOC20] GUTIÉRREZ, FRANCISCO ; SEIPP, KARSTEN ; OCHOA, XAVIER ; CHILUIZA, KATHERINE ; DE LAET, TINNE ; VERBERT, KATRIEN: LADA: A learning analytics dashboard for academic advising. In: *Computers in Human Behavior* vol. 107 (2020)
- [GuKH21] GURUGE, DEEPANI B. ; KADEL, RAJAN ; HALDER, SHARLY J.: The state of the art in methodologies of course recommender systems—a review of recent research. In: *Data* vol. 6 (2021), Nr. 2, pp. 1–30
- [GuLD18] GULZAR, ZAMEER ; LEEMA, A. ANNY ; DEEPAK, GERARD: PCRS: Personalized Course Recommender System Based on Hybrid Approach. In: *Procedia Computer Science*. vol. 125, 2018, pp. 518–524
- [HaKa12] HAN, J. ; KAMBER, M.: *Data mining: concepts and techniques*. vol. 49, 2012
- [HaPY05] HAN, J. ; PEI, J. ; YAN, X.: Sequential Pattern Mining by Pattern-Growth: Principles and Extensions* (2005), pp. 183–220
- [Hatt08] HATTIE, JOHN: *Visible learning: A synthesis of over 800 meta-analyses relating to achievement*, 2008 — ISBN 0203887336
- [HMRJ11] HAMIDI, FARIDEH ; MESHKAT, MARYAM ; REZAAE, MARYAM ; JAFAR, MEHDI: Information technology in education. In: *Procedia Computer Science*, 2011, pp. 369–373
- [HTCL17] HAN, MEIMEI ; TONG, MINGWEN ; CHEN, MENGYUAN ; LIU, JIAMIN ; LIU, CHUNMIAO: Application of Ensemble Algorithm in Students' Performance Prediction. In: *Proceedings - 2017 6th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2017* (2017), pp. 735–740 — ISBN 9781538606216
- [HuCC13] HUANG, CHUNG YI ; CHEN, RUNG CHING ; CHEN, LONG SHENG: Course-recommendation system based on ontology. In: *Proceedings - International Conference on Machine Learning and Cybernetics*. vol. 3, 2013 — ISBN 9781479902576, pp. 1168–1173
- [IaKF17] IATRELLIS, OMIROS ; KAMEAS, ACHILLES ; FITSILIS, PANOS: Academic advising systems: A systematic literature review of empirical evidence. In: *Education Sciences* vol. 7 (2017), Nr. 4
- [IsIs17] ISMAIL, SHAHRINAZ ; ISMAIL, SARERUSAENYE: Agent-mediated academic advising system: Nodal approach on knowledge management system towards achieving students' graduate-on-time. In: *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, IMCOM 2017* (2017) — ISBN 9781450348881
- [JaMF99] JAIN, A. K. ; MURTY, M. ; FLYNN, P. J.: Data Clustering: A Review. In: *ACM Computing Surveys* vol. 31 (1999), Nr. 3, pp. 264–323
- [Jign21] JIGNESH CHOWDARY, G.: Course Difficulty Estimation Based on Mapping of Bloom's Taxonomy and ABET Criteria. In: *arXiv preprint* (2021)
- [KaMY06] KAY, JUDY ; MAISONNEUVE, NICOLAS ; YACEF, KALINA: Mining Patterns of Events in Students' Teamwork Data. In: *In Educational Data Mining Workshop, held in conjunction with Intelligent Tutoring Systems (ITS)* (2006), pp. 45--52

- [KaPN17] KALAIVANI, S. ; PRIYADHARSHINI, B. ; NALINI, B. SELVA: Analyzing Student's Academic Performance Based on Data Mining Approach. In: *International Journal of Innovative Research in Computer Science & Technology* vol. 5 (2017), Nr. 1, pp. 194–197
- [KaSa05] KARAMPIPERIS, PYTHAGORAS ; SAMPSON, DEMETRIOS: Adaptive learning resources sequencing in educational hypermedia systems. In: *Educational Technology and Society* vol. 8 (2005), Nr. 4, pp. 128–147
- [KASJ14] KAUR, PARWINDER ; AGRAWAL, PRATEEK ; SINGH, SANJAY KUMAR ; JAIN, LEENA: Fuzzy rule based students' performance analysis expert system. In: *Proceedings of the 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques, ICICT 2014* (2014), pp. 100–105
- [KeLi05] KEOGH, EAMONN ; LIN, JESSICA: Clustering of time-series subsequences is meaningless: Implications for previous and future research. In: *Knowledge and Information Systems* vol. 8 (2005), Nr. 2, pp. 154–177
- [Kiri14] KIRIŞ, ŞAFAK: AHP and multichoice goal programming integration for course planning. In: *International Transactions in Operational Research* vol. 21 (2014), Nr. 5, pp. 819–833
- [Kots07] KOTSIANTIS, S. B.: Supervised machine learning: A review of classification techniques. In: *Informatica (Ljubljana)* vol. 31 (2007), Nr. 3, pp. 249–268
- [Kris05] KRISTOFIC, ANDREJ: Recommender System for Adaptive Hypermedia Applications. In: *IIT. SRC 2005: Student Research Conference* (2005), pp. 229–234
- [KuBa13] KUMAR ARORA, RAKESH ; BADAL, DHARMENDRA: Admission Management through Data Mining using WEKA. In: *International Journal of Advanced Research in Computer Science and Software Engineering* vol. 3 (2013), Nr. 10, pp. 2277–128
- [Kuch16] KUCHAROVA, D.: Internal Quality Assurance Practices in Czech Public Higher Education. In: *20th International conference on Current Trends in Public Sector Research, 2016*
- [Lafa17] LAPTEV, NIKOLAY ; AMIZADEH, SAEED ; FLINT, IAN ; AHMAD, SUBUTAI ; LAVIN, ALEXANDER ; PURDY, SCOTT ; AGHA, ZUHA ; MALHOTRA, PANKAJ ; ET AL.: A survey of sequential pattern mining. In: *Data Science and Pattern Recognition* vol. 34 (2017), Nr. 1, pp. 185–196 — ISBN 153864794X
- [Lagh16] LAGHARI, MOHAMMAD SHAKEEL: Knowledge Based Course Planning System for EE Students at UAE University. In: *Journal of Computers* (2016), pp. 455–462
- [LFPR22] LUNA, J. M. ; FARDOUN, H. M. ; PADILLO, F. ; ROMERO, C. ; VENTURA, S.: Subgroup discovery in MOOCs: a big data application for describing different types of learners. In: *Interactive Learning Environments* vol. 30 (2022), Nr. 1, pp. 127–145
- [Lock16] LOCKE, EDWIN A.: Problems With Goal-Setting Research in Sports—and Their Solution. In: *Journal of Sport and Exercise Psychology* vol. 13 (2016), Nr. 3, pp. 311–316
- [Lu04] LU, JIE: A personalized e-learning material recommender system. In: *Proceedings of the Second International Conference on Information Technology and Applications (ICITA 2004), 2004* — ISBN 0646423134, pp. 23–28

- [Luan02] LUAN, JING: Data Mining and Knowledge Management in Higher Education-Potential Applications. In: *Workshop associate of institutional research international conference, Toronto* (2002), pp. 1–18
- [LuFV19] LUNA, JOSÉ MARÍA ; FOURNIER-VIGER, PHILIPPE ; VENTURA, SEBASTIÁN: Frequent itemset mining: A 25 years review. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* vol. 9 (2019), Nr. 6
- [LXHY09] LI, GUANGYUAN ; XIAO, QIN ; HU, QINBIN ; YUAN, CHANGAN: An efficient algorithm for mining frequent sequences in dynamic environment. In: *2009 IEEE International Conference on Granular Computing, GRC 2009* (2009), pp. 329–333 — ISBN 9781424448319
- [MaDM16] MARBOUTI, FARSHID ; DIEFES-DUX, HEIDI A. ; MADHAVAN, KRISHNA: Models for early prediction of at-risk students in a course using standards-based grading. In: *Computers and Education* vol. 103 (2016), pp. 1–15
- [MaDS15] MARBOUTI, FARSHID ; DIEFES-DUX, HEIDI ; STROBEL, JOHANNES: Building Course-Specific Regression-based Models to Identify At-risk Students (2015), pp. 26.304.1-26.304.11
- [MAMA19] MOHAMMAD SUHAIMI, NURAFIFAH ; ABDUL-RAHMAN, SHUZLINA ; MUTALIB, SOFIANITA ; ABDUL HAMID, NURZEATUL HAMIMAH ; HAMID, ABDUL: Review on Predicting Students' Graduation Time Using Machine Learning Algorithms. In: *International Journal of Modern Education and Computer Science* vol. 11 (2019), Nr. 7, pp. 1–13
- [MCNY19] MA, YULING ; CUI, CHAORAN ; NIE, XIUSHAN ; YANG, GONGPING ; SHAHEED, KASHIF ; YIN, YILONG: Pre-course student performance prediction with multi-instance multi-label learning. In: *Science China Information Sciences* vol. 62 (2019), Nr. 2
- [MDVH11] MANOUSELIS, NIKOS ; DRACHSLER, HENDRIK ; VUORIKARI, RIINA ; HUMMEL, HANS ; KOPER, ROB: Recommender Systems in Technology Enhanced Learning. In: *Recommender Systems Handbook*, 2011, pp. 387–415
- [MeMa14] MEENAKSHI, G. ; MANISHARMA, V.: A rubric based assessment of student performance using fuzzy logic. In: *Advances in Intelligent Systems and Computing* vol. 236 (2014), pp. 557–563 — ISBN 9788132216018
- [MiRR12] MIRANDA, JAIME ; REY, PABLO A. ; ROBLES, JOSÉ M.: UdpSkeduler: A Web architecture based decision support system for course and classroom scheduling. In: *Decision Support Systems* vol. 52 (2012), Nr. 2, pp. 505–513
- [Moha15] MOHAMED, ABDALLAH: A decision support model for long-term course planning. In: *Decision Support Systems* vol. 74 (2015), pp. 33–45
- [MOKR14] MOSTAFA, L. ; OATELY, G. ; KHALIFA, N. ; RABIE, W.: A Case based Reasoning System for Academic Advising in Egyptian Educational Institutions. In: *2nd International Conference on Research in Science, Engineering and Technology (ICRSET)*, 2014, pp. 5–10
- [MoMi04] MOR, ENRIC ; MINGUILLÓN, JULIÀ: E-learning personalization based on Itineraries and long-term navigational behavior. In: *Thirteenth International World Wide Web Conference Proceedings, WWW2004* (2004), pp. 996–997 — ISBN 158113844X

- [Mour05] MOURSUND, DAVID: Introduction to Information and Communication Technology in Education. In: *University of Oregon* (2005), pp. 1–121
- [Mund91] MUNDFROM, DANIEL JAMES: *Estimating course difficulty*, 1991
- [MWHL17] MA, HUALONG ; WANG, XIANDE ; HOU, JIANFENG ; LU, YUNJUN: Course recommendation based on semantic similarity analysis. In: *2017 3rd IEEE International Conference on Control Science and Systems Engineering, ICCSSE 2017, 2017* — ISBN 9781538604847, pp. 638–641
- [NaZw14] NATEK, SREČKO ; ZWILLING, MOTI: Student data mining solution-knowledge management system related to higher education institutions. In: *Expert Systems with Applications* vol. 41 (2014), Nr. 14, pp. 6400–6407
- [NLRV16] NOAMAN, AMIN Y. ; LUNA, JOSÉ MARÍA ; RAGAB, ABDUL H.M. ; VENTURA, SEBASTIÁN: Recommending degree studies according to students' attitudes in high school by means of subgroup discovery. In: *International Journal of Computational Intelligence Systems* vol. 9 (2016), Nr. 6, pp. 1101–1117
- [NXWZ08] NESBIT, J. C. ; XU, Y. ; WINNE, P. H. ; ZHOU, M.: Sequential pattern analysis software for educational event data. In: *Learning and Instruction*. vol. 14, 2008, pp. 307–323
- [NZXW07] NESBIT, JC ; ZHOU, M ; XU, Y ; WINNE, PHILIP H.: Advancing log analysis of student interactions with cognitive tools. In: *12th Biennial Conference of the European Association for Research on Learning and Instruction, 2007*, pp. 1–20
- [OgGO14] OGBULOGO, CHARLES U ; GEORGE, TAYO O ; OLUKANNI, DAVID O: Teaching Aids, Quality Delivery, and Effective Learning Outcomes in a Nigerian Private University. In: *Edulearn14: 6Th International Conference on Education and New Learning Technologies* (2014), Nr. July, pp. 61–68 — ISBN 978-84-617-0557-3
- [OuZh07] OUYANG, YANG ; ZHU, MIAOLIANG: eLORM: Learning object relationship mining based repository. In: *Proceedings - The 9th IEEE International Conference on E-Commerce Technology; The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services, CEC/EEE 2007* (2007), pp. 691–698 — ISBN 0769529135
- [PaLV19] PADILLO, F. ; LUNA, J. M. ; VENTURA, S.: A Grammar-Guided Genetic Programming Algorithm for Associative Classification in Big Data. In: *Cognitive Computation* vol. 11 (2019), Nr. 3, pp. 331–346
- [PaSP14] PATEL, SANSKRUTI ; SAJJA, PRITI ; PATEL, ATUL: Fuzzy Logic based Expert System for Students' Performance Evaluation in Data Grid Environment vol. 5 (2014), Nr. 1, pp. 36–40
- [Peña14] PEÑA-AYALA, ALEJANDRO: Educational data mining: A survey and a data mining-based analysis of recent works. In: *Expert Systems With Applications* vol. 41 (2014), Nr. P1, pp. 1432–1462
- [PHMW04] PEI, JIAN ; HAN, JIAWEI ; MORTAZAVI-ASL, BEHZAD ; WANG, JIANYONG ; PINTO, HELEN ; CHEN, QIMING ; DAYAL, UMESHWAR ; HSU, MEI CHUN: Mining sequential patterns by pattern-growth: The prefixspan approach. In: *IEEE Transactions on Knowledge and Data Engineering* vol. 16 (2004), Nr. 11, pp. 1424–1440
- [Phyu09] PHYU, T. N.: Survey of Classification Techniques in Data Mining. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists, 2009*, pp. 1–8

- [PIDM21] PLOTNIKOVA, VERONIKA ; DUMAS, MARLON ; MILANI, FREDRIK: Adapting the CRISP-DM Data Mining Process: A Case Study in the Financial Services Domain. In: *Lecture Notes in Business Information Processing* vol. 415 LNBI (2021), pp. 55–71 — ISBN 9783030750176
- [PrVi19] PREMALATHA, M. ; VISWANATHAN, V.: Course Sequence Recommendation with Course Difficulty Index Using Subset Sum Approximation Algorithms. In: *Cybernetics and Information Technologies* vol. 19 (2019), Nr. 3, pp. 25–44
- [ReFT09] REIMANN, PETER ; FREREJEAN, JIMMY ; THOMPSON, KATE: Using process mining to identify models of group decision making in chat data. In: *Computer Supported Collaborative Learning Practices, CSCL 2009 Conference Proceedings - 9th International Conference* (2009), pp. 98–107 — ISBN 1615841377
- [RiTs20] RICHARDS, ASHEEN LATEILLA ; TSAY, REN SONG: An Optimal Slack-Based Course Scheduling Algorithm for Personalised Study Plans. In: *ACM International Conference Proceeding Series* (2020), pp. 1–7 — ISBN 9781450375085
- [Roka05] ROKACH, L.: Clustering Methods. In: *Data Mining and Knowledge Discovery Handbook*, 2005, pp. 61–94
- [RoVe10] ROMERO, CRISTÓBAL ; VENTURA, SEBASTIÁN: Educational data mining: A review of the state of the art. In: *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* vol. 40 (2010), Nr. 6, pp. 601–618
- [RoVe20] ROMERO, CRISTÓBAL ; VENTURA, SEBASTIÁN: Educational data mining and learning analytics: An updated survey. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* vol. 10 (2020), Nr. 3
- [RoVG08] ROMERO, CRISTÓBAL ; VENTURA, SEBASTIÁN ; GARCÍA, ENRIQUE: Data mining in course management systems: Moodle case study and tutorial. In: *Computers and Education* vol. 51 (2008), Nr. 1, pp. 368–384
- [RSGS13] RASMANI, KHAIRUL A. ; SHAHARI, NOR A. ; GARIBALDI, JONATHAN M. ; SHEN, QIANG: Practicality Issues in Using Fuzzy Approaches for Aggregating Students' Academic Performance. In: *Procedia - Social and Behavioral Sciences* vol. 83 (2013), pp. 398–402
- [RVZB09] ROMERO, CRISTÓBAL ; VENTURA, SEBASTIÁN ; ZAFRA, AMELIA ; BRA, PAUL DE: Applying Web usage mining for personalizing hyperlinks in Web-based adaptive educational systems. In: *Computers and Education* vol. 53 (2009), Nr. 3, pp. 828–840
- [ScKG21] SCHRÖER, CHRISTOPH ; KRUSE, FELIX ; GÓMEZ, JORGE MARX: A systematic literature review on applying CRISP-DM process model. In: *Procedia Computer Science* vol. 181 (2021), pp. 526–534
- [ScYP17] SCHOLZ, FREDERIKE ; YALCIN, BETUL ; PRIESTLEY, MARK: Internet access for disabled people: Understanding socio-relational factors in Europe. In: *Cyberpsychology* vol. 11 (2017), Nr. 1Special Issue
- [SGBM04] SALAZAR, A. ; GOSÁLBEZ, J. ; BOSCH, I. ; MIRALLES, R. ; VERGARA, L.: A case study of knowledge discovery on academic achievement, student desertion and student retention. In: *ITRE 2004 - 2nd International Conference on Information Technology: Research and Education - Proceedings, 2004* — ISBN 0780386256, pp. 150–154

- [ShAb20] SHAKHSI-NIAEI, MAJID ; ABUEI-MEHRIZI, HOSSEIN: An optimization-based decision support system for students' personalized long-term course planning. In: *Computer Applications in Engineering Education* vol. 28 (2020), Nr. 5, pp. 1247–1264
- [Shaf86] SHAFFER, JULIET POPPER: Modified Sequentially Rejective Multiple Test Procedures. In: *Journal of the American Statistical Association* vol. 81 (1986), Nr. 395, p. 826
- [ShSh04] SHEN, LI PING ; SHEN, RUI MIN: Learning Content Recommendation Service Based-On Simple Sequencing Specification. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 3143 (2004), pp. 363–370 — ISBN 3540225420
- [SMMS17] SCHULTE, JURGEN ; DE MENDONCA, PEDRO FERNANDEZ ; MARTINEZ-MALDONADO, ROBERTO ; SHUM, SIMON BUCKINGHAM: Large scale predictive process mining and analytics of university degree course data. In: *ACM International Conference Proceeding Series* (2017), pp. 538–539 — ISBN 9781450348706
- [SPGB17] SAXENA, AMIT ; PRASAD, MUKESH ; GUPTA, AKSHANSH ; BHARILL, NEHA ; PATEL, OM PRAKASH ; TIWARI, ARUNA ; ER, MENG JOO ; DING, WEIPING ; ET AL.: A review of clustering techniques and developments. In: *Neurocomputing* vol. 267 (2017), pp. 664–681
- [SRA00] SONAVANE, S ; RATHI, M ; AND, S PATIL - INTERNATIONAL JOURNAL OF EDUCATION ; 2018, UNDEFINED: Practice based closure quality loop optimization in teaching learning process: A case study in software engineering design. In: *Iaras.Org*
- [SrAg96] SRIKANT, RAMAKRISHNAN ; AGRAWAL, RAKESH: Mining sequential patterns: Generalizations and performance improvements. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 1057 LNCS (1996), pp. 3–17 — ISBN 354061057X
- [Stat40] STATISTICS, MATHEMATICAL: A Comparison of Alternative Tests of Significance for the Problem of m Rankings. In: *The Annals of Mathematical Statistics* vol. 11 (1940), Nr. 1, pp. 86–92
- [Taha12] TAHA, KAMAL: Automatic academic advisor. In: *CollaborateCom 2012 - Proceedings of the 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2012* — ISBN 9781936968367, pp. 262–268
- [ThNT16] THANH-NHAN, HUYNH LY ; NGUYEN, HUU HOA ; THAI-NGHE, NGUYEN: Methods for building course recommendation systems. In: *Proceedings - 2016 8th International Conference on Knowledge and Systems Engineering, KSE 2016* (2016), pp. 163–168 — ISBN 9781467389297
- [ToPa18] TOMY, SARATH ; PARDEDE, ERIC: Course Map: A Career-Driven Course Planning Tool. In: *Computational Science and Its Applications - (ICCSA) 2018*, 2018, pp. 185–198
- [TřPe09] TŘČKA, NIKOLA ; PECHENIZKIY, MYKOLA: From local patterns to global models: Towards domain driven educational process mining. In: *ISDA 2009 - 9th International Conference on Intelligent Systems Design and Applications* (2009), pp. 1114–1119 — ISBN 9780769538723
- [TuSA19] TUAHA, SANA ; SIDDIQUI, ISMA FARRAH ; ALI ARAIN, QASIM: Analyzing Students' Academic Performance through Educational Data Mining. In: *3C Tecnología_Glosas de innovación aplicadas a la pyme* (2019), pp. 402–421

- [TzYH03] TZVETKOV, PETRE ; YAN, XIFENG ; HAN, JIAWEI: TSP: Mining top-K closed sequential patterns. In: *Proceedings - IEEE International Conference on Data Mining, ICDM (2003)*, pp. 347–354 — ISBN 0769519784
- [Unel11] UNELSRØD, HF: *Design and Evaluation of a Recommender System for Course Selection*, 2011
- [VeLu16] VENTURA, SEBASTIÁN ; LUNA, JOSÉ MARÍA: *Pattern mining with evolutionary algorithms*, 2016 — ISBN 9783319338583
- [VPSP14] VEERAMUTHU, P ; PERIYASAMY, R ; SUGASINI, V ; PATTI-, PUTHANAM: Analysis of Student Result Using Clustering Techniques. In: *International Journal of Computer Science and Information Technologies*, vol. 5 (2014), Nr. 4, pp. 5092–5094
- [WaZa15] WANG, REN ; ZAÏANE, OSMAR R: Discovering Process in Curriculum Data to Provide Recommendation. In: *Proceeding of the 8th International Conference on Educational Data Mining, EDM15 (2015)*, pp. 580–581
- [Wirt00] WIRTH, R., & HIPPE, J.: CRISP-DM : Towards a Standard Process Model for Data Mining. In: *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, 2000, pp. 29–39
- [WSTY12] WU, CHENG WEI ; SHIE, BAI EN ; TSENG, VINCENT S. ; YU, PHILIP S.: Mining top-K high utility itemsets. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2012)*, pp. 78–86 — ISBN 9781450314626
- [WuHa05] WU, KUN ; HAVENS, WILLIAM S.: Modelling an academic curriculum plan as a mixed-initiative constraint satisfaction problem. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 3501 LNAI (2005), pp. 79–90 — ISBN 3540258647
- [WuLZ15] WU, DIANSHUANG ; LU, JIE ; ZHANG, GUANGQUAN: A Fuzzy Tree Matching-Based Personalized E-Learning Recommender System. In: *IEEE Transactions on Fuzzy Systems* vol. 23 (2015), Nr. 6, pp. 2412–2426
- [WWST04] WANG, WEI ; WENG, JUI FENG ; SU, JUN MING ; TSENG, SHIAN SHYONG: Learning portfolio analysis and mining in SCORM compliant environment. In: *Proceedings - Frontiers in Education Conference, FIE* vol. 1 (2004)
- [YiBa14] YILDIZ, ZEHRA ; BABA, A. FEVZI: Evaluation of student performance in laboratory applications using fuzzy decision support system model. In: *IEEE Global Engineering Education Conference, EDUCON (2014)*, pp. 1023–1027 — ISBN 9781479931910
- [YiBG13] YILDIZ, OSMAN ; BAL, ABDULLAH ; GULSECEN, SEVINC: Improved fuzzy modelling to predict the academic performance of distance education students. In: *International Review of Research in Open and Distance Learning* vol. 14 (2013), Nr. 5, pp. 144–165
- [Yu15] YU, SHAOYING: Performance management and evaluation research to university students. In: *Chemical Engineering Transactions* vol. 46 (2015), pp. 631–636 — ISBN 9788895608372

- [ZAAZ12] ZAINUDIN, SUHAILA ; AHMAD, KAMSURIAH ; ALI, NAZLENA MOHAMAD ; ZAINAL, NOOR FARIDATUL AINUN: Determining Course Outcomes Achievement Through Examination Difficulty Index Measurement. In: *Procedia - Social and Behavioral Sciences* vol. 59 (2012), pp. 270–276
- [Zaki01] ZAKI, MOHAMMED J.: SPADE: An efficient algorithm for mining frequent sequences. In: *Machine Learning* vol. 42 (2001), Nr. 1–2, pp. 31–60
- [ZhLL08] ZHANG, LIANG ; LIU, XIUMIN ; LIU, XIUJUAN: Personalized instructing recommendation system based on web mining. In: *Proceedings of the 9th International Conference for Young Computer Scientists, ICYCS 2008* (2008), pp. 2517–2521 — ISBN 9780769533988